BIG-IP® Local Traffic Manager™: Implementations

Version 13.1

Table of Contents

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.

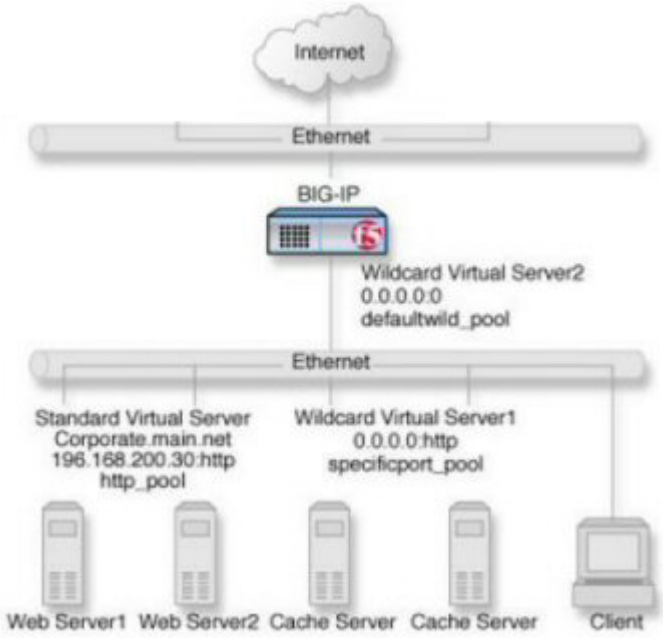


Figure 1: Non-intranet connections

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

You can create pool of servers that you group together to receive and process traffic, to enable the BIG-IP system to efficiently distribute the load on servers.

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, 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. For a host, in the **Destination Address** field, type an IPv4 or IPv6 address in CIDR format to allow all traffic to be translated.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `0.0.0.0/0`, and an IPv6 address/prefix is `::/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 the relevant 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

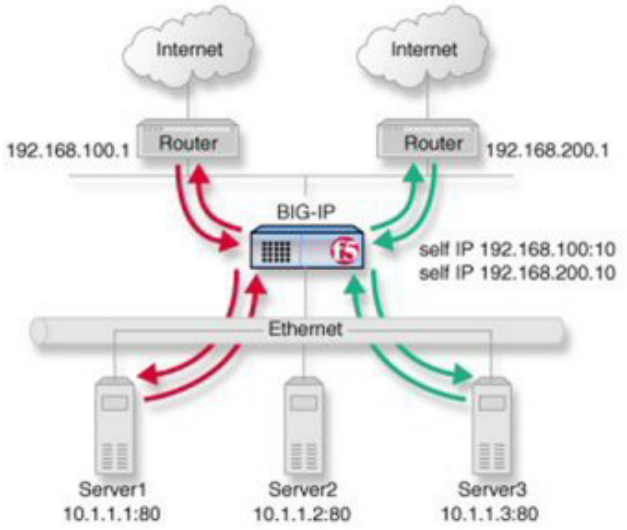


Figure 2: ISP load balancing

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 a create 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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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 the relevant 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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 the relevant 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 network mask for the specified IP address.

   For example, you can type `255.255.255.0`.
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** list, select **external** and **internal**, and using the Move button, transfer the VLANs to the **Selected** list.
6. Click the **Finished** button.

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 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 name, such as `XML_CBR_iRule`.
   The full path name of the iRule cannot exceed 255 characters.
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
`$XML_count`	Shows the number of matching queries.
`$XML_queries`	Contains an array of the matching query names.
`$XML_values`	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.

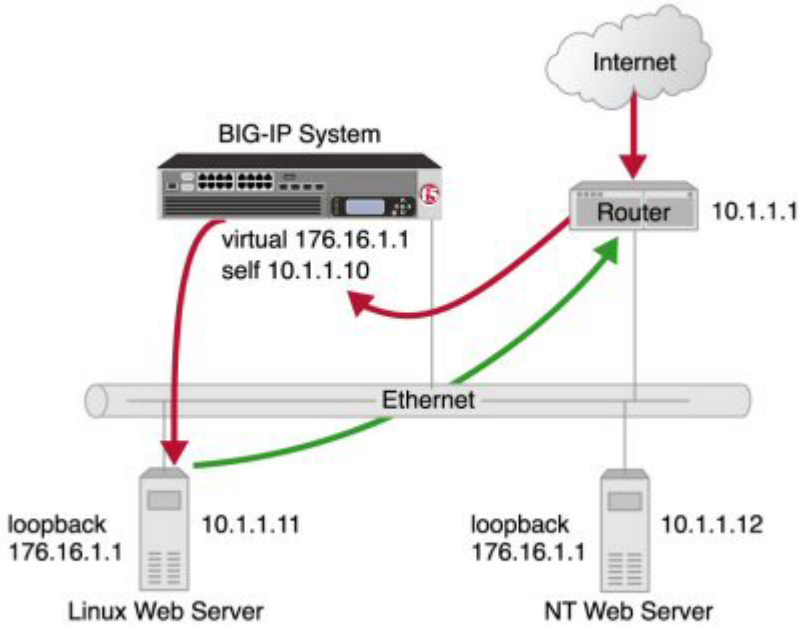


Figure 3: Layer 2 nPath routing

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 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 only for a one-arm virtual server configuration.
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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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 the relevant 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 IPIP and GRE.

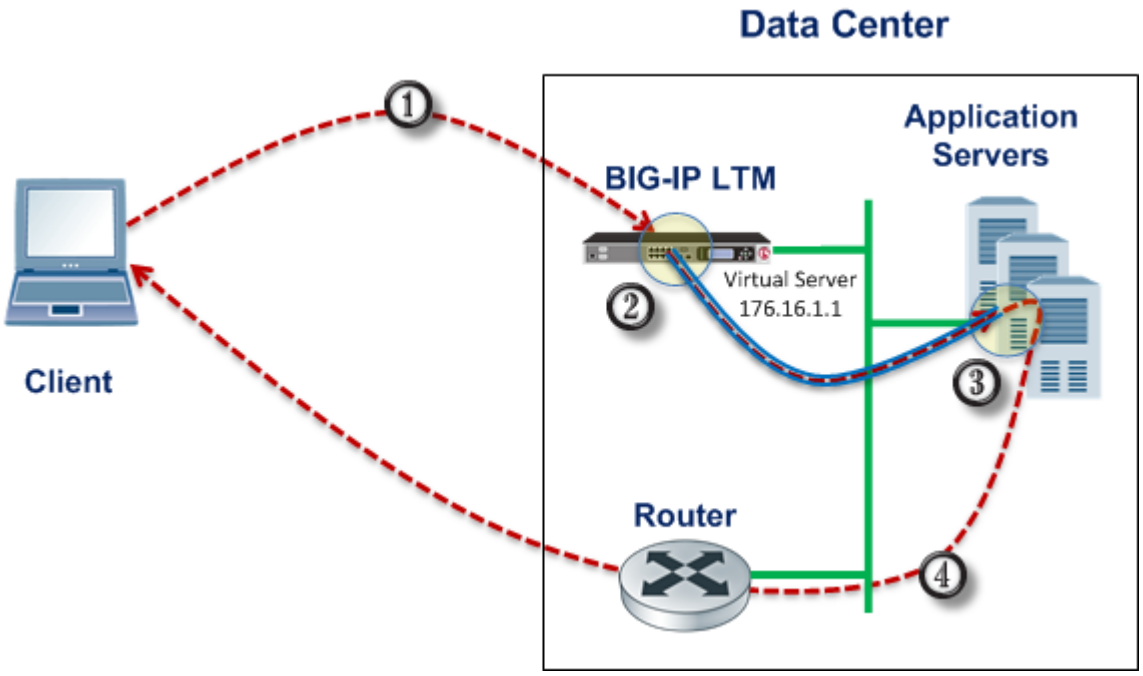


Figure 4: Example of a Layer 3 routing configuration

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.

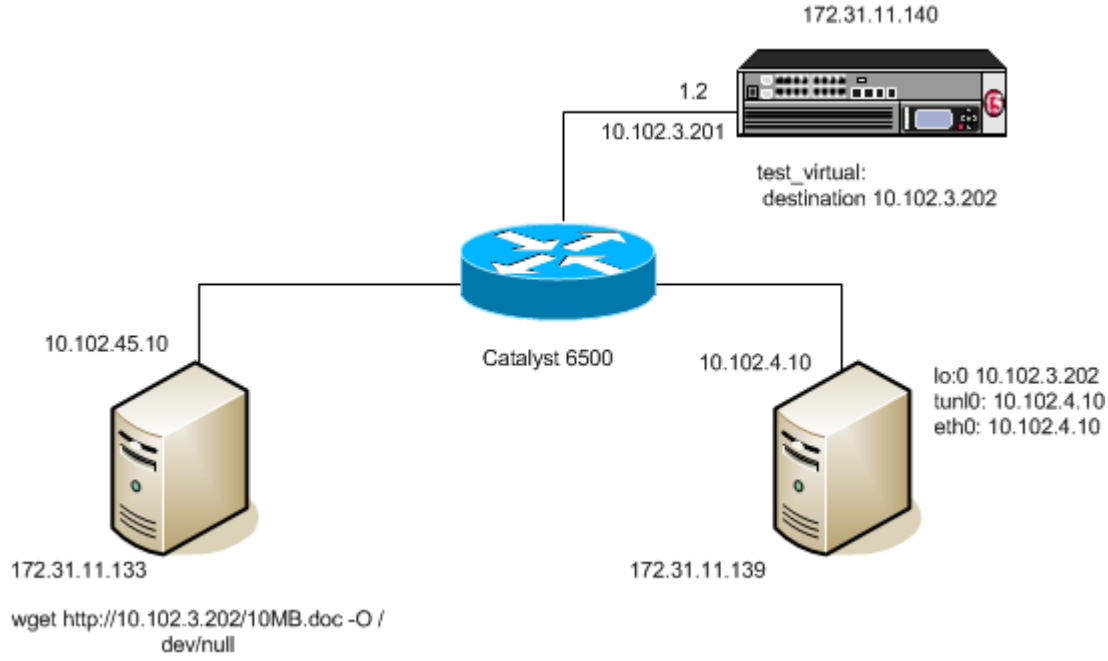


Figure 5: Example of a Layer 3 routing configuration

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

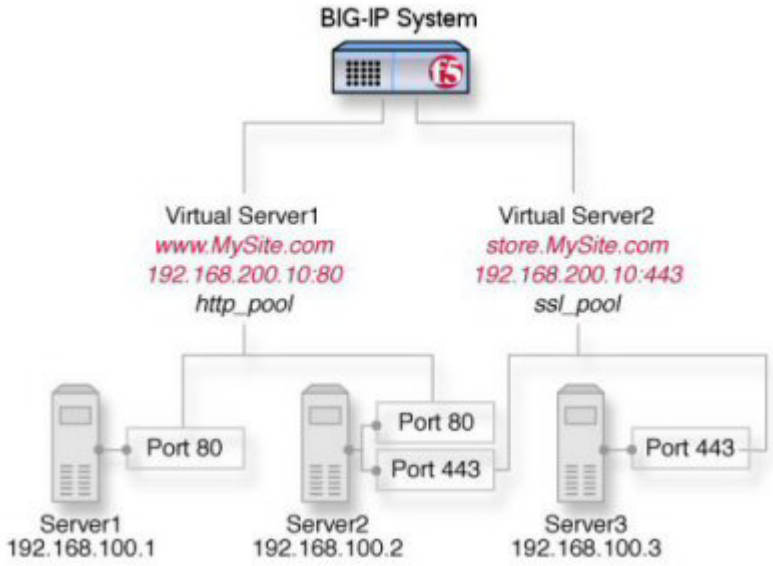


Figure 6: Basic web site and eCommerce configuration

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 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. For the **Health Monitors** setting, assign **https** or **https_443** by moving it from the **Available** list 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. Use the **New Members** setting to add each resource that you want to include in the pool:
   a) In the **Address** field, type an IP address.

b) In the **Service Port** field type `443` , 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 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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. In the Resources area of the screen, from the **Default Pool** list, select the relevant pool name.
10. 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. In the **Destination Address** field, type the IP address in CIDR format.

The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

Note: *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.

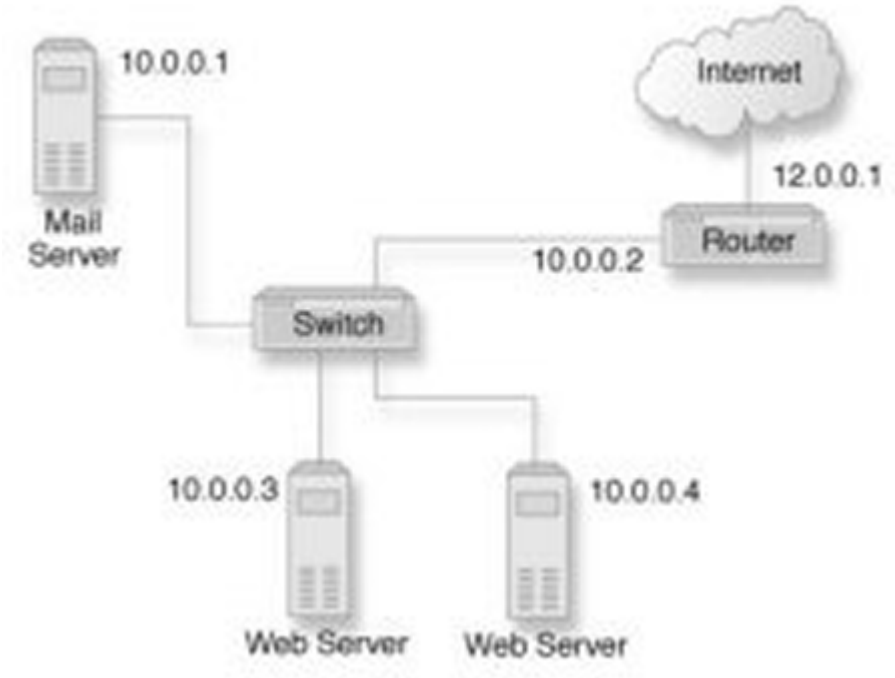


Figure 7: Data center example before adding a BIG-IP system

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.

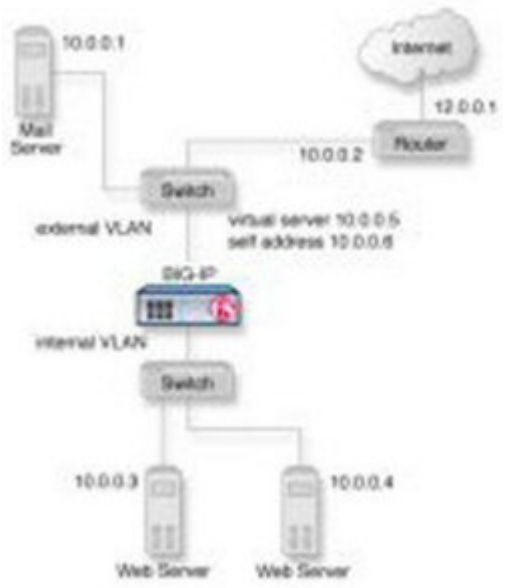


Figure 8: Data center example after adding a BIG-IP system

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

Before you create a self IP address, ensure that you have created at least one VLAN or 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 network mask for the specified IP address.

   For example, you can type `255.255.255.0`.
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 a create 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 Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *The IP address for this field needs to be on the same subnet as the external self-IP address.*
5. From the **Service Port** list, select ***All Ports**.
6. In the Resources area of the screen, from the **Default Pool** list, select the relevant 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

Downloading the IP intelligence database
Creating an iRule to log IP 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

Downloading the IP intelligence database

The requirements for using IP 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 Intelligence is enabled by default if you have a license for it. You only need to enable it if it was previously disabled.*

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

1. Log in to the command line for the BIG-IP® system.
2. To determine whether IP intelligence auto-update 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. No further steps are necessary.
3. If disabled, 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 Intelligence feature remains enabled unless you disable it with the command `tmsh modify sys db iprep.autoupdate value disable`.

Creating an iRule to log IP intelligence information

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

You use iRules® to log IP 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, such as `my_irule`.
   The full path name of the iRule cannot exceed 255 characters.
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 Intelligence categories, type the following iRule:

```
when CLIENT_ACCEPTED {
            log local0. "IP 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 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 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, such as `my_irule`.

   The full path name of the iRule cannot exceed 255 characters.
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 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 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 spam messages, launch various attacks, or cause target systems to behave in other unpredictable ways.
Cloud Service Providers	IP addresses and networks that belong to cloud providers, which offer services hosted on their servers via the Internet.
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.
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.
Mobile Threats	IP addresses of malicious and unwanted mobile applications.
Phishing Proxies	IP addresses that host phishing sites, and other kinds of fraud activities, such as ad click fraud or gaming fraud.
Proxy	IP addresses that are associated with web proxies that shield the originator's IP address (such as proxy and anonymization services).
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.

Category Name	Description
Tor Proxies	IP addresses acting as exit nodes for the Tor Network. Exit nodes are the last point along the proxy chain and make a direct connection to the originator's intended destination.
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.

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.

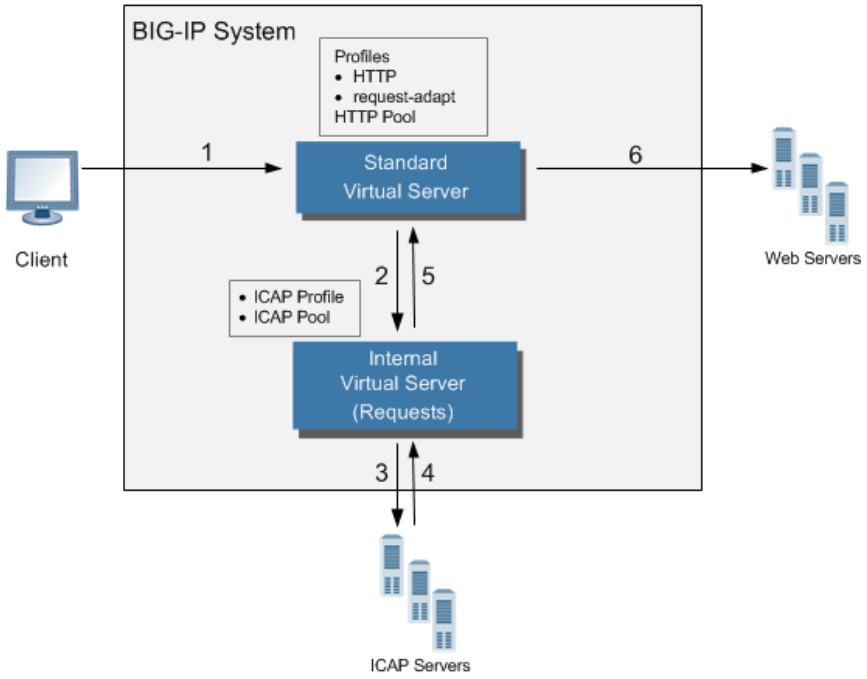


Figure 9: Content adaptation configuration for modifying HTTP requests

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 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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) 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 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. In the **Destination 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 supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

Note: *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.

Note: *If traffic is on a secure internal network, you can use `80`/**HTTP**.*

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. There is support for secure connectivity for ICAP between a BIG-IP system internal virtual server and a pool of ICAP servers.

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.

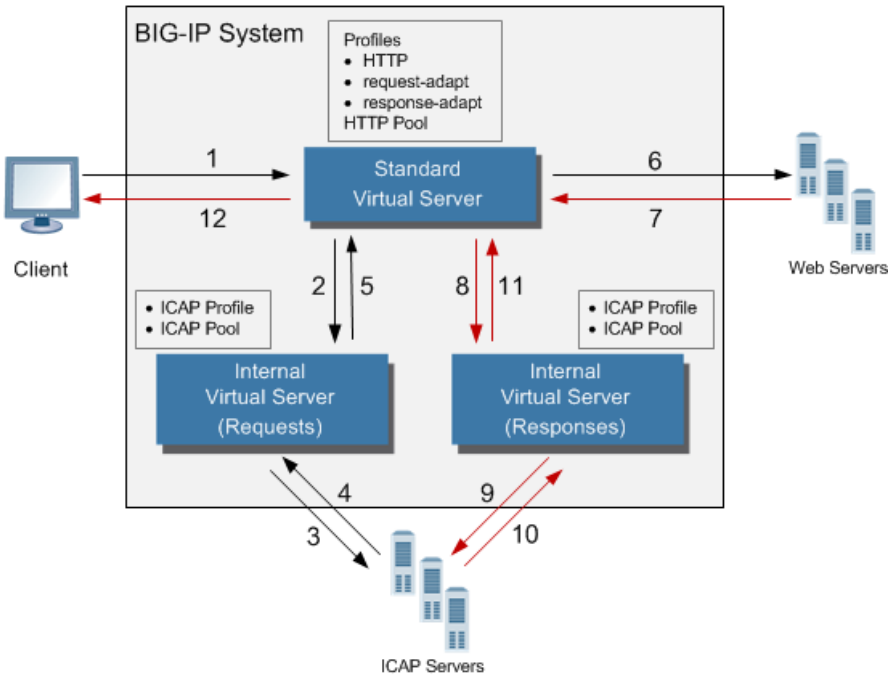


Figure 10: Content adaptation configuration for modifying HTTP requests and responses

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 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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) 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 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. There is

support for secure connectivity for ICAP between a BIG-IP system internal virtual server and a pool of ICAP 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. In the **Destination 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 supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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

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 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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 the relevant 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. From the Main tab, click **Local Traffic** > **Profiles** > **Persistence**.
2. Click **Create**.
3. In the **Name** field, type a name for the profile.
4. From the **Persistence Type** list, select **Cookie**.
5. From the **Parent Profile** list, select **cookie**.
6. On the right side of the screen, select the Custom check box.
7. From the **Cookie Method** list, select **HTTP Cookie Insert**.
8. If you want the BIG-IP system to encrypt the pool name specified in the BigIPServer default cookie, select the **Default Cookie Encrypt Pool Name** check box.
9. Retain or change all other profile settings.
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 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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 the relevant 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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 the relevant 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.

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.

Task list

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:

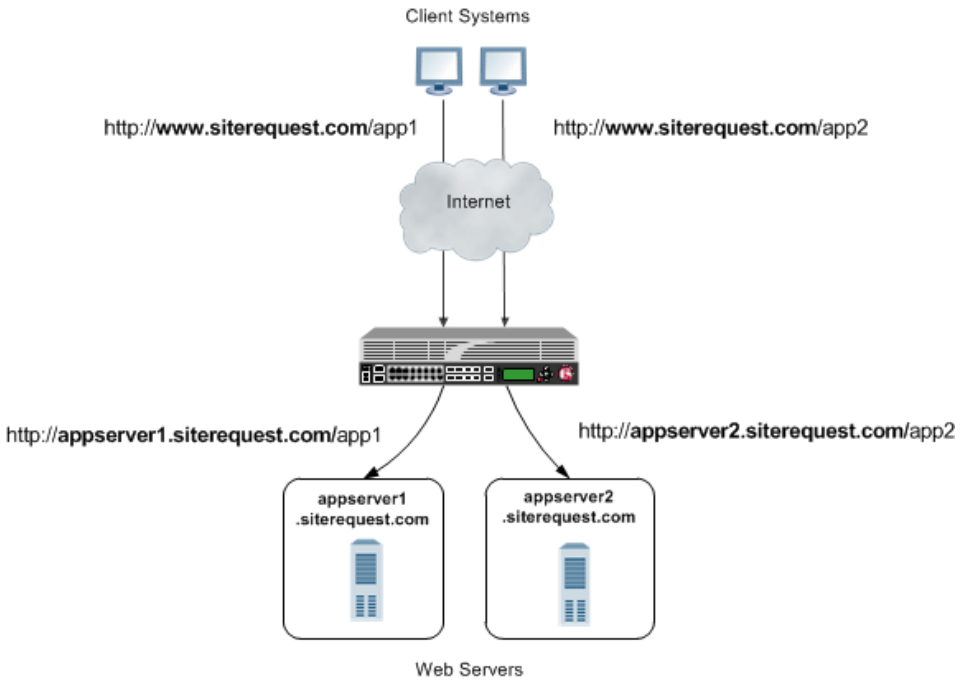


Figure 11: The BIG-IP system as a reverse proxy server for URI translation

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.

Note: *The Rewrite profile supports HTML and CSS content types only. To specify MIME types for HTML content, you can either create an HTML profile or accept the default values that the Rewrite profile uses,* `text/html` *and* `text/xhtml`*. For CSS content, only the* `text/css` *MIME type is supported.*

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.

Note: *The Rewrite profile supports HTML and CSS content types only. To specify MIME types for HTML content, you can either create an HTML profile or accept the default values that the Rewrite profile uses,* `text/html` *and* `text/xhtml`*. For CSS content, only the* `text/css` *MIME type is supported.*

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 pop-up 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** field, type a client path, such as `/marketing/`.
15. In the **Server URI** field, 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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) 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 Page 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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, select the local traffic policy you created from the **Available** list 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.

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.

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 **Local Traffic** > **Monitors**.
   The Monitors List screen opens.
2. Click **Create**.
   The New Monitor screen opens.
3. In the **Name** field, type a name for the monitor.
4. From the **Type** list, select **FTP**.
   The screen refreshes, and displays the configuration options for the **FTP** monitor type.
5. From the **Import Monitor** list, select an existing monitor.

   The new monitor inherits initial configuration values from the existing monitor.
6. In the **Interval** field, type a number that indicates, in seconds, how frequently the system issues the monitor check. The default is `10` seconds.
7. In the **Timeout** field, type a number 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 name in the **User Name** field.

9. Type a password in the **Password** field.
10. In the **Path/Filename** field, type the full path and file name of the file that the system attempts to download.

    The health check is successful if the system can download the file.
11. 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.

12. From the **Configuration** list, select **Advanced**.

    This selection makes it possible for you to modify additional default settings.
13. For the **Up Interval** setting, specify whether to use the up interval:
    - If you do not want to use the up interval, Retain the default, **Disabled**.
    - 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.
14. In the **Time Until Up** field, type a number 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.
15. For **Manual Resume**, specify whether the system automatically enables the monitored resource when the monitor check is successful.

    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.

16. For the **Alias Address** setting, specify an alias IP address:
    - Retain 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`.
17. For the **Alias Service Port** setting, specify an alias port or service for the monitor to check:
    - 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.
18. 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.

19. 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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; for the **Related iRules** setting, from the **Available** list, select the name of the iRule that you want to assign and 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 the relevant 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 Monitors List screen opens.
2. Click **Create**.
   The New Monitor screen opens.
3. In the **Name** field, type a name for the monitor.
4. From the **Type** list, select **FTP**.
   The screen refreshes, and displays the configuration options for the **FTP** monitor type.
5. From the **Import Monitor** list, select an existing monitor.

   The new monitor inherits initial configuration values from the existing monitor.
6. In the **Interval** field, type a number that indicates, in seconds, how frequently the system issues the monitor check. The default is `10` seconds.
7. In the **Timeout** field, type a number 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 name in the **User Name** field.
9. Type a password in the **Password** field.
10. In the **Path/Filename** field, type the full path and file name of the file that the system attempts to download.

    The health check is successful if the system can download the file.
11. 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.

12. From the **Configuration** list, select **Advanced**.

    This selection makes it possible for you to modify additional default settings.
13. For the **Up Interval** setting, specify whether to use the up interval:
    - If you do not want to use the up interval, Retain the default, **Disabled**.
    - 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.
14. In the **Time Until Up** field, type a number 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.
15. For **Manual Resume**, specify whether the system automatically enables the monitored resource when the monitor check is successful.

    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.

Option	Description
No	The system automatically re-enables the monitored resource after the next successful monitor check.

16. For the **Alias Address** setting, specify an alias IP address:
    - Retain 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`.
17. For the **Alias Service Port** setting, specify an alias port or service for the monitor to check:
    - 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.
18. 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.

19. 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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; for the **Related iRules** setting, from the **Available** list, select the name of the iRule that you want to assign and 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 the relevant 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.

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) 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.

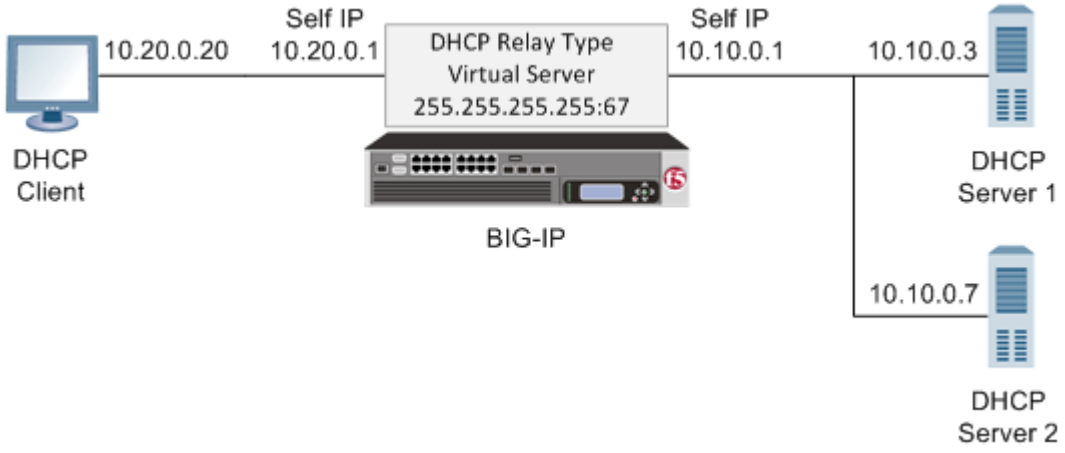


Figure 12: A sample DHCP relay agent configuration

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 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 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 type of 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. (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 type virtual server

A DHCP type of 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**.
6. Select one of the following to configure a **Destination Address** type.

Destination	Steps to configure
255.255.255.255 (IPv4 Default)	None.
ff02::1:2 (IPv6 Default)	None.
Other	For a host or network, in the **Destination Address** field, type an IPv4 address/prefix or an IPv6 address/prefix.

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 type of 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 type of 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.

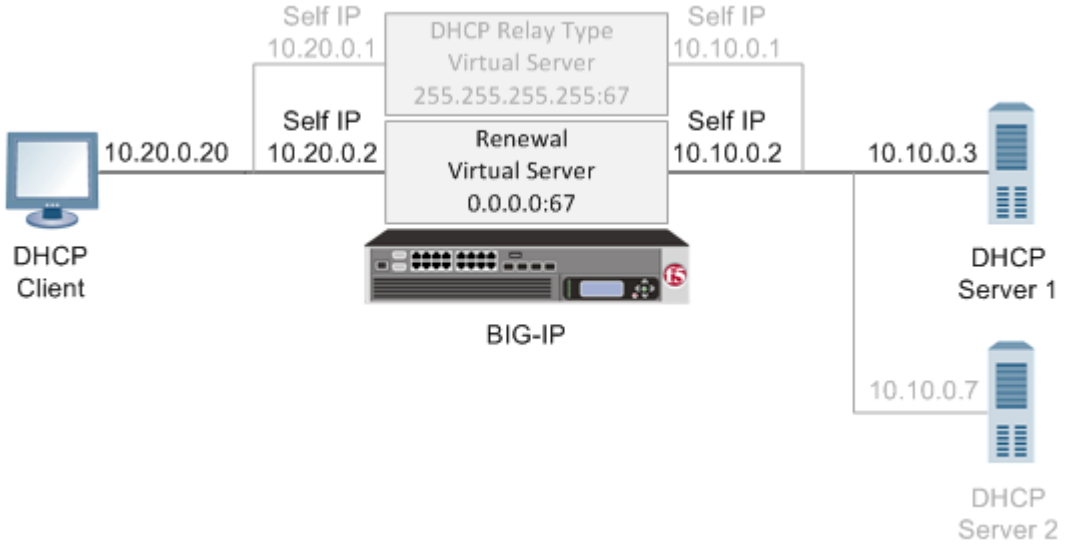


Figure 13: A sample DHCP renewal system configuration

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 host, in the **Destination Address** field, type an IPv4 or IPv6 address in CIDR format to allow all traffic to be translated.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `0.0.0.0/0`, and an IPv6 address/prefix is `::/0`.
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

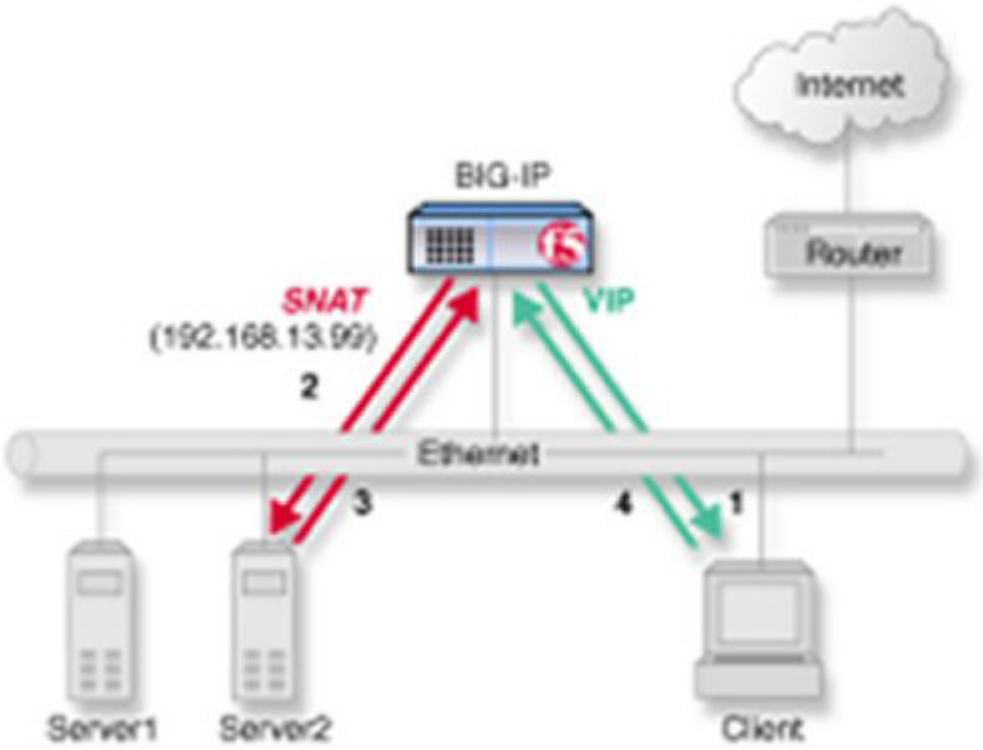


Figure 14: One-IP network topology for the BIG-IP system

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
Configuring optional ephemeral port exhaustion

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 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. In the **Destination Address** field, type the IP address in CIDR format.

The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

Note: *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 the relevant 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. 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) In the **Address** field., type a client IP address.
   b) 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 the **Finished** button.

The BIG-IP system is configured to handle connections originating from the client

Configuring optional ephemeral port exhaustion

You must configure a client SNAT before you can configure ephemeral port exhaustion functionality for that SNAT.

You can configure the BIG-IP® system to accumulate real-time ephemeral-port statistics, and when usage exceeds a specified threshold level, to log an error and provide a Simple Network Management Protocol (SNMP) alert notification. Thus you can assess an approaching exhaustion of ephemeral ports, and respond accordingly.

1. Log on to the command line of the system using the `root` account.
2. Type `tmsh` to access the Traffic Management Shell.
3. Type the following command to enable ephemeral port-exhaustion threshold warning functionality. The default value is `enabled`.

   ```
   modify ltm global-settings traffic-control port-find-threshold-warning
   [enabled_or_disabled]
   ```

4. Type the following command to specify the number of random attempts to find an unused outbound port for a connection. Values can range from `1` through `12`. The default value is `8`.

   ```
   modify ltm global-settings traffic-control port-find-threshold-trigger
   [threshold_level]
   ```

5. Type the following command to specify the timeout period, in seconds, from one threshold trigger until a subsequent threshold trigger, which if exceeded, resets and causes the threshold warning to expire. Values can range from `0` through `300` seconds. The default value is `30`.

   ```
   modify ltm global-settings traffic-control port-find-threshold-timeout
   [timeout_period]
   ```

The BIG-IP system is configured to accumulate real-time ephemeral-port statistics, and to provide a trigger when usage exceeds a specified threshold level.

You need to configure logging functionality, for example, high-speed remote logging, to log any error messages. Additionally, you will want to manage any alert notifications by using SNMP.

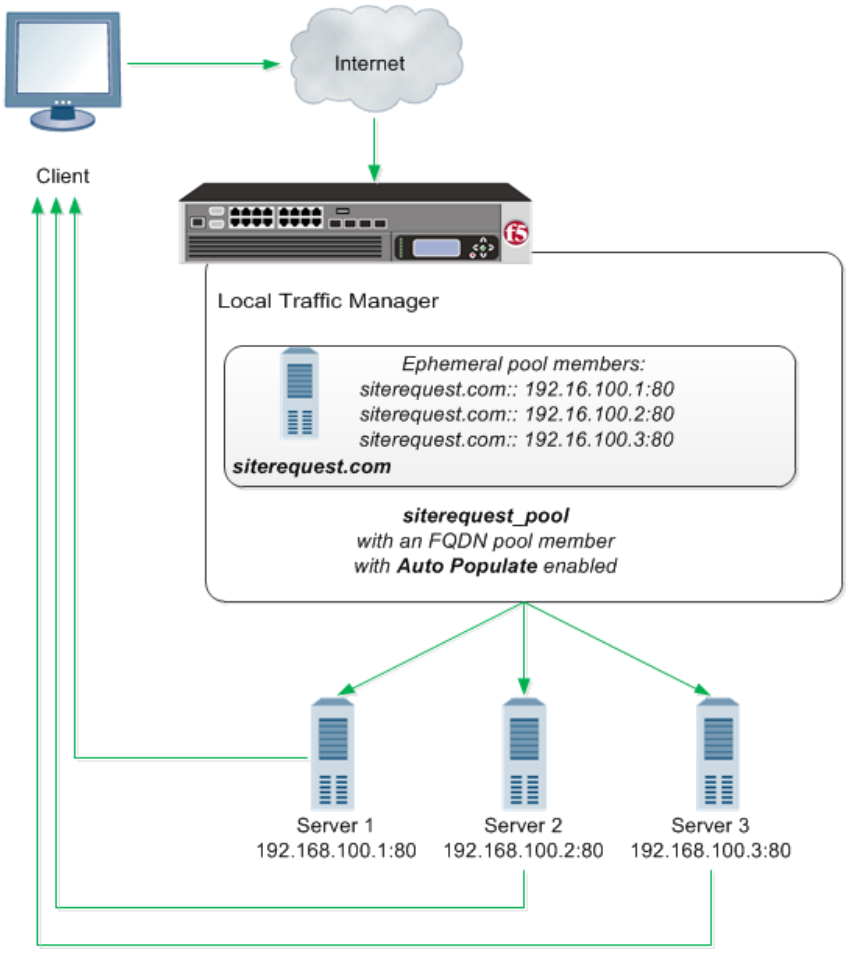
Configuring the BIG-IP System to Auto-Populate Pools

Overview: Using host names to identify pool members and nodes

You create nodes on the BIG-IP® system to represent the backend servers on your network. In turn, you create pool members to represent the backend servers on your network when you create a pool and want to load balance traffic to multiple backend servers.

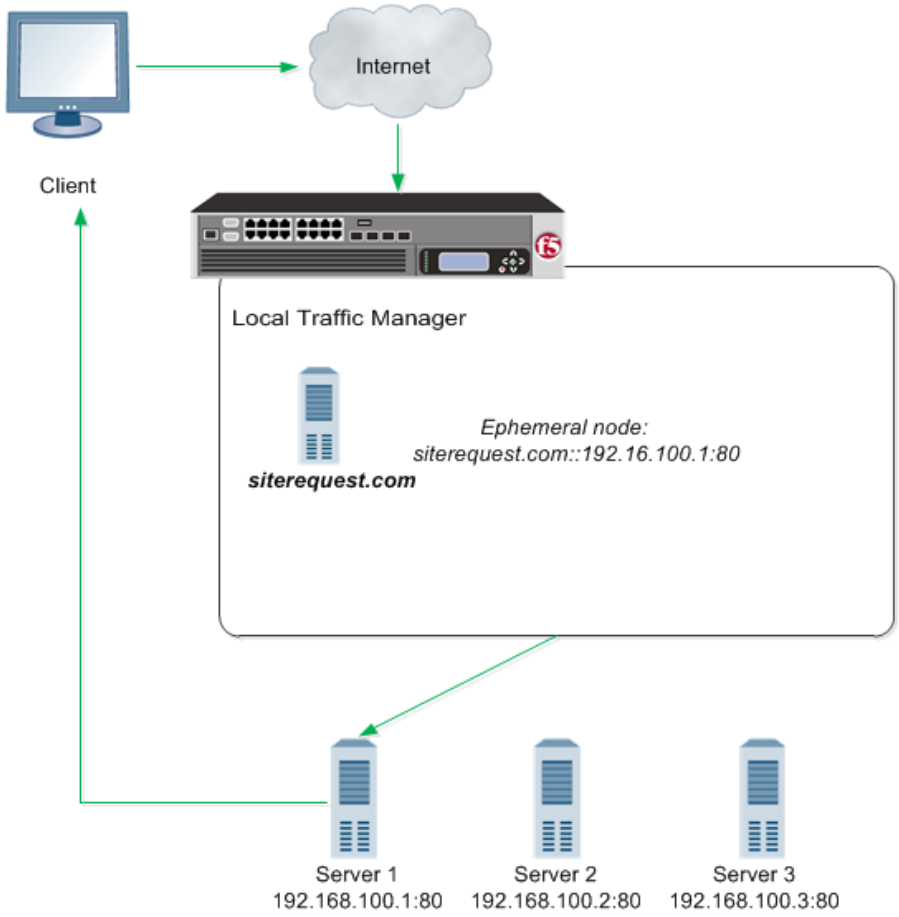
You can configure a BIG-IP system with nodes and pool members that are identified with fully-qualified domain names (FQDNs). When you configure pool members with FQDN, addresses will dynamically follow DNS changes. Fully dynamic DNS-managed pools may even be created. In the following illustration, the BIG-IP Local Traffic Manager™ creates an ephemeral pool member for each IP address returned in the DNS response.

Figure 15: BIG-IP system auto-populating a pool and routing traffic to the pool members



This next illustration shows another option. With this configuration, the system sends a DNS query for the FQDN, and then creates only one ephemeral node or pool member using the first IP address returned in the DNS response. An advantage to this configuration is that you can change the IP addresses of the backend servers that host the domain without reconfiguring the BIG-IP system. However, if your DNS servers are configured to round robin DNS responses, this feature is not recommended.

Figure 16: BIG-IP system routing traffic to a node identified by a host name



About modes of failure and related nodes or pool members

If a node or pool member that is identified by a fully-qualified domain name (FQDN) is down for a specified amount of time, the BIG-IP® system marks the node or pool member down. Failure to resolve a FQDN will not cause the marking down of nodes or pool members currently in service. While the status of the FQDN node or pool member for DNS is reflected in the status of the FQDN node, since the FQDN node or pool member does not itself monitor any servers, its status does not contribute to the status of the pool in any way.

Failure of a monitored ephemeral to respond to monitor probes results in the marking down of a specific node. Niether the FQDN or any of the related ephemeals are directly affected. Because ephemeral objects monitor servers, the status of the epemeral node or pool member affects the pool status in the same way as any other pool member or node.

Task summary

Perform these tasks to configure the BIG-IP® system to auto-populate pools.

Creating a default gateway pool
Configuring the BIG-IP system to handle DNS lookups
Creating nodes using host names
Creating a pool using host names

Creating a default gateway pool

Create a default gateway pool for the system to use to forward 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. For the **Health Monitors** setting, from the **Available** list, select the **gateway_icmp** monitor and move the monitor to the **Active** list.
5. Using the **New Members** setting, add each router that you want to include in the default gateway pool:
   a) Type the IP address of a router in the **Address** field.
   b) Type an asterisk (*) in the **Service Port** field, or select ***All Services** from the list.
   c) Click **Add**.
6. Click **Finished**.

Configuring the BIG-IP system to handle DNS lookups

Configure how the BIG-IP® system handles DNS lookups when you want to use fully-qualified domain names (FQDNs) to identify nodes and pool members.

1. On the Main tab, click **System** > **Configuration** > **Device** > **DNS**.
   The DNS Device configuration screen opens.
2. In the DNS Lookup Server List area, in the **Address** field, type the IP address of the DNS server(s) you want to add.

   The system uses these DNS servers to validate DNS lookups and resolve host names. Then, click **Add**.

   Note: *If you did not disable DHCP before the first boot of the system, and if the DHCP server provides the information about your local DNS servers, then this field is automatically populated.*

   ---
3. Click **Update**.

Creating nodes using host names

Determine the fully-qualified domain name (FQDN) that you want to use to identify a node.

You can create nodes identified by FQDNs and then create a pool and add pool members from a list of nodes.

1. On the Main tab, expand **Local Traffic**, and click **Nodes**.
   The Node List screen opens.
2. Click the **Create** button.
   The New Node screen opens.
3. In the **Name** field, type a descriptive label for the node.

   Names are case-sensitive.
4. For the **Address** setting, select **FQDN**, and then type the host name in the field.
5. In the Configuration area, from the **Health Monitors** list, select the way that you want the system to apply monitors to the node.

The default setting is **Node Default**.

Option	Description
Node Default	Specifies that the system uses the defined default monitors for nodes. The default monitors are defined on the Default Monitors screen of the BIG-IP Configuration utility.
Node Specific	Specifies that the system monitors this node with the monitors that you configure in the **Select Monitors** setting. ***Note:*** *When you select the* ***Node Specific*** *option, the screen refreshes to display the* ***Select Monitors*** *setting.*
None	Specifies that the system does not monitor this node.

6. In the **Ratio** field, type a number for the ratio weight of the node.
7. In the **Connection Limit** field, type a number for the maximum established connection limit for the node.
8. In the **Connection Rate Limit** field, type a number that specifies the number of new connections accepted per second for the node.
9. From the **Address Type** list, select whether the node resolves to an IPv4 or IPv6 address. The default is **IPv4**.
10. From the **Auto Populate** list, select **Enabled**. The options are:

Option	Description
Enabled	The system automatically creates ephemeral nodes using the IP addresses returned by the resolution of a DNS query for the FQDN, that is, for each DNS entry of the resolved FQDN.
Disabled	The system automatically creates a node that corresponds to the IP address of only the first DNS entry of the resolved FQDN.

11. In the **Interval** field, type the number of seconds before the system creates new ephemeral nodes or deletes expired ephemeral nodes based on the IP addresses returned in response to a DNS query for the FQDN of the node. The default is the TTL of the IP address in the DNS response.
12. In the **Down Interval** field, type the number of seconds the system waits to mark an FQDN node down following a DNS query failure.
13. Click **Finished**.
    The screen refreshes, and the new node appears in the node list.

Creating a pool using host names

Before creating a pool, determine the servers that you want to add to the pool using a fully-qualified domain name (FQDN).

Ensure that your DNS servers are not configured for round robin DNS resolutions; instead, ensure that your DNS servers return all available IP addresses in a DNS resolution.

When you want the BIG-IP® system to automatically update pool members as you make changes to the IP addresses of servers in your network, you can create a pool of servers that are identified by FQDNs.

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 a monitor and move the monitor to the **Active** list.

   ***Note:** A pool containing nodes represented by FQDNs cannot be monitored by* `inband` *or* `sasp` *monitors.*

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

   The default is **Round Robin**.

6. In the **New Members** setting, add at least one node with a static IP address. This node serves as a fallback if a DNS query returns no records for the nodes identified by FQDNs.
   a) Select **Node List**.
   b) From the **Address** list, select a node identified by a static IP address.
   c) From the **Service Port** list, make a selection.
   d) Click **Add**.

7. In the **New Members** setting, add the members that you want to include in the pool using FQDNs.
   a) Select **FQDN Node List**.
   b) From the **Address** list, select a node identified by an FQDN.
   c) Make a selection from the **Service Port** list.

8. In the **New Members** setting, select **Enabled**. The options are:

Option	Description
Enabled	The system generates an ephemeral node for each IP address returned in response to a DNS query for the FQDN of the node. Additionally, when a DNS response indicates the IP address of an ephemeral node no longer exists, the system deletes the ephemeral node.
Disabled	The system selects the first address and generates an ephemeral for that address.

9. Click **Add**.
10. Repeat steps 7-9 to add additional members to the pool.
11. Click **Finished**.
    The screen refreshes, and you see the new pool in the Pool list.

About modifying nodes and pool members identified by host names

When you change the configuration of a fully-qualified domain name (FQDN) pool member or node, any ephemeral pool members or nodes that the BIG-IP® system created based on the IP addresses returned in a DNS response for that FQDN are automatically modified, as well. For example, if you change the monitor on an FQDN node, the system automatically changes the monitor assigned to the ephemeral nodes associated with that node.

When you want to modify an FQDN pool member or node, but you want persistent and active connections to be completed before the BIG-IP system marks the pool member or node as down, disable the pool member or node first, and then make modifications.

Task summary

Disabling a node
Disabling a pool member

Disabling a node

Determine the node that you want to disable.

You can disable a node when you want to make changes to your network, but you want persistent and active connections to be completed before the BIG-IP® system marks the node as down.

1. On the Main tab, click **Local Traffic** > **Nodes**.
   The Node List screen opens.
2. In the Name column, click a node name.
3. In the State area, click **Disabled (Only persistent or active connections allowed)**.

   Note: *You can only disable the parent FQDN node or pool member. After disabling, the ephemeral dependents are then disabled, but you cannot directly disable an ephemeral node.*

4. Click **Update**.
   The screen refreshes, and the status in the Availability area changes.

Disabling a pool member

Determine the pool member that you want to disable. You can only disable a parent fully-qualified domain name (FQDN) node or pool member. The ephemeral dependents are then disabled. You cannot directly disable the ephemerals.

Disable a pool member when you want to make changes to your network, but you want persistent and active connections to be completed before the BIG-IP® system marks the pool member as down.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click a pool name in the Pool List.
3. On the menu bar, click **Members**.
4. In the **Member** list, select the relevant pool member.
5. In the State area, click **Disabled (Only persistent or active connections allowed)**.
6. Click **Update**.
   The screen refreshes, and the status in the Availability area changes.

About pool member and node statistics

You can view statistics about pool members and nodes identified by host names.

Task summary

Viewing statistics for a specific node
Viewing statistics for ephemeral pool members

Viewing statistics for a specific node

Ensure that at least one LTM® node exists on the BIG-IP® system.

You can view statistics for an LTM node when you want to analyze BIG-IP system traffic.

1. On the Main tab, click **Statistics** > **Module Statistics** > **Local Traffic**.
   The Local Traffic statistics screen opens.
2. From the **Statistics Type** list, select **Nodes**.
   Information displays about the node.

Viewing statistics for ephemeral pool members

Ensure that at least one LTM® node exists on the BIG-IP® system.

When you want to analyze how the BIG-IP system is handling traffic, you can view statistics for pools and pool members, including the ephemeral pools created when the pool member is identified by a fully-qualified domain name (FQDN) and **Auto Populate** is enabled for the pool member.

1. On the Main tab, click **Statistics** > **Module Statistics** > **Local Traffic**.
   The Local Traffic statistics screen opens.
2. From the **Statistics Type** list, select **Pools**.
   Information displays about the pools configured on the BIG-IP system. The ephemeral pool members are shown indented below their parent pool member and with two dashes preceding the pool member name.

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 Monitors List screen opens.
2. Click **Create**.
   The New Monitor screen opens.
3. In the **Name** field, type a name for the monitor.
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 Monitor** 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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
b) In the **Address** field, type an IP address.
c) In the **Service Port** field, type a port number, or select a service name from the list.
d) (Optional) In the **Priority** field, type a priority number.
e) 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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 relevant pool name.

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.

Enabling TCP enhanced loss recovery

Although this feature is disabled by default, you can enable Enhanced Loss Recovery using TCP profiles. This will increase the TCP performance for networks with low RTT and high packet loss. Enhanced loss recovery retransmits lost segments multiple times during a loss recovery period to prevent a timeout recovery.

Important: *This feature can be used only when the Selective ACKs feature is enabled for the TCP connection.*

1. On the Main tab, click **Local Traffic** > **Profiles** > **Protocol** > **TCP**.
2. Click an existing profile or click **Create**.
3. Scroll down to the Loss Detection and Recovery area, and select the **Custom** check box at the right.
4. For the **Enhanced Loss Recovery** setting, select the check box.

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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.

c) In the **Service Port** field, type a port number, or select a service name from the list.
d) (Optional) In the **Priority** field, type a priority number.
e) 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. In the **Destination Address** field, type the IPv6 address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv6 address/prefix is `64:ff9b::/64` or `2001:ed8:77b5:2::/64`.
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 *SYN flood* 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 *ICMP flood*, 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. 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.	You do not need to make any changes to the BIG-IP system configuration for this type of attack.
UDP flood	The *UDP flood* 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.

Attack type	Description	Mitigation
UDP fragment	The *UDP fragment* 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 *Ping of Death* 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 *Land* 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 *Teardrop* 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.
WinNuke	The *WinNuke* 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 *Sub 7* 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.

Attack type	Description	Mitigation
Back Orifice	A *Back Orifice* 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:

```
Connection Limit = 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)

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`my_crldp_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_crldp** 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)

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`my_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)

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.

About RADIUS profiles

The BIG-IP® system includes a profile type that you can use to load balance Remote Authentication Dial-In User Service (RADIUS) traffic.

When you configure a RADIUS type of profile, the BIG-IP system can send client-initiated RADIUS messages to load balancing servers. The BIG-IP system can also ensure that those messages are persisted on the servers.

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`my_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`my_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)

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`my_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 OCSP 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)

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.

When implementing an SSL OCSP authentication module, you must also create a third type of object. This object is referred to as an OCSP responder.

Task list

Creating an SSL OCSP responder object for authenticating application traffic remotely
Creating an SSL OCSP configuration object for authenticating application traffic remotely
Creating a custom SSL OCSP profile
Modifying a virtual server for SSL OCSP authentication

Creating an SSL OCSP responder object for authenticating application traffic remotely

An *SSL OCSP responder object* is an object that you create that includes a URL for an external SSL OCSP responder. You must create a separate SSL OCSP responder object for each external SSL OCSP 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 as `my_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 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 as `my_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 OCSP authentication

The final task in the process of implementing SSL OCSP authentication is to assign the custom SSL OCSP 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 OCSP profile. Then, using the **Move** button, move the custom SSL OCSP profile to the **Selected** field.
5. Click **Update** to save the changes.

The virtual server is assigned the custom SSL OCSP 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)

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 `my_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 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. 1. In the **Sender** field, type a sender ID that represents the identity of the firm sending the message. Example: `client1` 2. In the **Data Group** field, type a tag substitution data group. Example: `FIX_tag_map` 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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) 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. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *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 the relevant 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 FIX features with low latency

The PVA hardware does not examine the FIX packets that stream through it, so FIX-profile features such as parsing and tag substitution are not supported with low-latency.

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 must specify the TCP protocol setting.

About using low-latency electronic trading with HSRP or VRRP

You can use low-latency electronic trading in a Hot Standby Router Protocol (HSRP) or Virtual Router Redundancy Protocol (VRRP) environment, with a last-hop pool configured with a single pool member to maintain acceleration flows. When using low-latency electronic trading in an HSRP or VRRP environment, you must set the db variable `tmlhpnomemberaction` to `2`, enabling the BIG-IP® system to only route the client traffic back through a pool member defined in the last hop pool. Additionally, in this configuration, the system can respond to client traffic that originates from an address other than an address defined in the last hop pool.

Example

For example, consider the following configuration.

- Router 1 has an IP address of `10.1.1.251`.
- Router 2 has an IP address of `10.1.1.252`.
- Last-hop pool member has a virtual IP address of `10.1.1.254`.

In this example, you create a last-hop pool with a single pool member that is assigned with a virtual IP address of `10.1.1.254`. You can then use the following tmsh command to set the db variable `tmlhpnomemberaction` to `2`.

```
tmsh modify /sys db tm.lhpnomemberaction value 2
```

Note: *Typically, you will want to use a transparent monitor on the last-hop pool.*

Task summary

There are several tasks you can perform to implement low-latency electronic trading.

Task list

Licensing low-latency electronic trading functionality
Creating a custom Fast L4 profile for FIX
Creating a pool
Creating a virtual server for low-latency electronic trading

Licensing 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 specific license. The license must enable both of the following features:

- Advanced LTM® Protocols
- FIX Low Latency

Please contact your F5® Networks support representative to acquire the necessary license.

Creating a custom Fast L4 profile for FIX

You can create a custom Fast L4 profile to manage Layer 4 traffic for FIX.

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 **PVA Acceleration** field to **Guaranteed**.
6. Select the **Loose Close** check box only for a one-arm virtual server configuration.
7. Set the **TCP Close Timeout** setting, according to the type of traffic that the virtual server will process.
8. 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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.

c) In the **Service Port** field, type a port number, or select a service name from the list.
d) (Optional) In the **Priority** field, type a priority number.
e) Click **Add**.

5. Click **Finished**.
6. Repeat these steps for each pool you want to create.

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. From the **Type** list, select **Performance (Layer 4)**.
5. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

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

6. From the **Configuration** list, select **Advanced**.
7. From the **Protocol** list, select **TCP**.
8. From the **Protocol Profile (Client)** list, select the custom Fast L4 profile you defined for low-latency FIX trading.
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 the relevant pool name.
12. Click **Finished**.

The virtual server is configured to use the specified Fast L4 profile and pool. If a client initiates a FIX connection with this virtual server, the connection uses the Fast L4 (ePVA) hardware.

Implementation result

This implementation configures a BIG-IP® system to manage low-latency electronic trading functionality, optimizing the system for predictable latency and jitter. Clients who send FIX packets to the virtual server's Destination address all receive this low-latency service.

Implementing Low-Latency Electronic Trading with FIX load balancing

Overview: Configuring low-latency electronic trading with FIX load balancing

You can configure the BIG-IP® system to manage electronic trading traffic for both low-latency and intelligent load balancing. The BIG-IP system supports Financial Information eXchange (FIX) protocol connections for electronic trading between financial institutions. When you acquire a special license, you can use the FastL4 profile to optimize the FIX connections, and use the embedded Packet Velocity® ASIC (ePVA) to minimize the latency. You can then use an iRule to implement intelligent load balancing: you do this by enabling the Late Binding feature in the FastL4 profile, then creating an iRule that parses each FIX header to choose a back-end server pool.

About Late Binding

With the Late Binding feature enabled, an iRule can examine the FIX logon packet, the one that establishes the connection, and choose a server pool based on the packet's contents. The iRule finishes by sending the connection down to the ePVA hardware, which processes the stream at high speed.

The only TCP options available to the client and server are MSS, accept Selective ACK, and Time Stamp. The BIG-IP system ignores all other options because it must enable SYN cookies on the client-side interface, and because the ePVA hardware does not slow down for any of those options. For example, the BIG-IP system ignores the Window Scaling option as soon as the flow has been released to the ePVA hardware.

Note: *Secure Sockets Layer (SSL) is not supported by a virtual server that uses Late Binding.*

About FIX features with low latency

After the iRule selects a server, the ePVA hardware manages the FIX stream for the rest of its existence. The ePVA does not examine the individual FIX packets that pass through it, so FIX-profile features such as tag substitution are not supported.

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 ePVA 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 must specify the TCP protocol setting.

Task summary

There are several tasks you perform to implement low-latency electronic trading.

Task list

Licensing low-latency electronic trading functionality

Licensing 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 specific license. The license must enable both of the following features:

- Advanced LTM® Protocols
- FIX Low Latency

Please contact your F5® Networks support representative to acquire the necessary license.

Creating a custom Fast L4 profile for FIX

You can create a custom Fast L4 profile to manage Layer-4 traffic for FIX.

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 **PVA Acceleration** field to **Guaranteed**.
6. If you plan to use **Late Binding** and either of the **Loose Initiation** and **Loose Close** check boxes are enabled, clear them both.

   The **Late Binding** feature examines the first few packets in the FIX stream, and the **Loose Initiation** feature makes it possible to skip those packets without any examination.
7. Set the **TCP Close Timeout** setting, according to the type of traffic that the virtual server will process.
8. Disable the **Hardware SYN Cookie Protection** feature by clearing the check box.
9. Enable the **Software SYN Cookie Protection** feature by selecting the check box.
10. The **Late Binding** feature makes it possible to choose a server pool based on data in the FIX header. An iRule in the virtual server parses the FIX header and selects the server pool. Select the check box to enable **Late Binding**.
    a) You can allow the iRule to explicitly determine when the flow is released from Layer 7 down to Layer 4. The iRule code can then perform additional computation before binding the connection to Layer 4. Enable this by selecting the **Explicit Flow Migration** check box. When this feature is enabled, the flow is not released to Layer 4 until the iRule invokes the `BIGTCP::release_flow` command.

       By default, this is disabled and the flow drops down to Layer 4 immediately after the connection to the server is established.
    b) Use the **Client Timeout** field to determine how much time to allow for any client to send the first 2144 bytes of Layer 7information. In normal cases, this amount of data arrives immediately.
    c) From the **Timeout Recovery** list, select an action that the profile should take in case of timeout. Select **Disconnect** to drop the connection summarily, or select **Fallback** to process the packet without parsing the Layer 7 fields. The fallback option sends any timed-out connection to the Virtual Server's default pool.
11. Click **Finished**.

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

Creating a FIX profile for low-latency electronic trading

A virtual server with Late Binding enabled can choose a server pool based on the contents of the FIX connection's initial packet. The Late Binding feature makes it possible to combine this load balancing with low latency.

Note: *This is a simplified FIX profile. The low-latency path goes through the ePVA hardware, which does not examine the contents of each FIX packet. The only packet that the BIG-IP software examines is the logon packet, which the BIG-IP® system uses to choose a server pool. Therefore, most of the features in the FIX-profile screen (such as tag substitution) are ignored for low-latency trading.*

---

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. Click **Finished**.

The FIX profile is configured for low-latency electronic trading with FIX load balancing.

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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) Click **Add**.
5. Click **Finished**.
6. Repeat these steps for each pool you want to create.

The new pool appears in the Pools list.

Creating an iRule for load-balancing Layer-7 (FIX) traffic

Creating an iRule for load-balancing Layer-7 (FIX) traffic requires that the Late Binding feature is enabled in the Fast L4 profile. A virtual server with Late Binding enabled can choose a server pool based on the contents of the FIX connection's initial logon packet(s).

You can create an iRule that reads the FIX logon fields and directs each TCP stream to a different server pool based on the field settings.

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 match FIX fields and choose a server pool based on their settings.

   Use the FIX_HEADER iRule event to select the first five fields in a FIX packet:

   - BeginString
   - BodyLength
   - MsgType
   - SenderCompID
   - TargetCompID

   The total length of a FIX message is unbounded, so this ensures that you capture all of the relevant data to choose a back-end server pool without waiting to collect all of the FIX message.

   For example, this iRule sends messages from each of three senders to a specific server pool. Messages from any other senders revert to the default pool in a virtual server that uses this iRule. The iRule also logs a message to indicate that a new FIX stream has opened:

```
when FIX_HEADER {
    set MsgType [FIX::tag get 35]
    if { $MsgType eq "A" } {  # an A message is a logon message
        # record the sender and the target
        set SenderCompID [FIX::tag get 49]
        set TargetCompID [FIX::tag get 56]

        # log the event locally - a new FIX stream is being created
        log "FIX header: Sender $SenderCompID, Target $TargetCompID"

        # log the event with High Speed Logging (HSL), too
        set hsl [HSL::open -proto UDP -pool syslog_server_pool]
        HSL::send $hsl "[IP::client_addr]: Sender $SenderCompID, Target $TargetCompID\n"

        # choose a server pool based on the name of the sender
        switch $SenderCompID {
            "Fred's Bank"   { pool FIX1 }
            "Wilma's Bank"  { pool FIX2 }
            "Barney's Bank" { pool FIX3 }
        }
    }
}
```

The iRule may be able to explicitly send the flow down to the ePVA, rather than doing it automatically. This explicit control is only possible if you set it in the Fast L4 profile. In the following example, the rule does not release the flow unless it encounters a FIX packet from a sender named `"Mr. Slate's Bank"`. You must release the flow on both the client side (with the CLIENT_ACCEPTED event) and the server side (in the SERVER_CONNECTED event):

```
when CLIENT_ACCEPTED {
    # prepare for releasing the flow down to the ePVA
    BIGTCP::release_flow
}

when FIX_HEADER {
    # (same as above example, with an additional sender)
```

```
    set MsgType [FIX::tag get 35]
    if { $MsgType eq "A" } {  # an A message is a logon message
        # record the sender and the target
        set SenderCompID [FIX::tag get 49]
        set TargetCompID [FIX::tag get 56]

        # log the event - a new FIX stream is being created
        log "FIX header: Sender $SenderCompID, Target $TargetCompID"

        # choose a server pool based on the name of the sender
        switch $SenderCompID {
            "Fred's Bank"      { pool FIX1 }
            "Wilma's Bank"     { pool FIX2 }
            "Barney's Bank"    { pool FIX3 }
            "Mr. Slate's Bank" { pool FIX4 }
        }
    }
}

when SERVER_CONNECTED {
    if {$SenderCompID eq "Mr. Slate's Bank"}{
        # Mr. Slate's Bank sent this, so lower the latency
        log local0. "Detected $SenderCompID - releasing flow to ePVA"
        BIGTCP::release_flow
    }
}
```

The previous code sends all FIX streams through standard FIX-profile processing except the one(s) from `"Mr. Slate's Bank"`, which goes through the ePVA.

5. Click **Finished**.

The iRule is now available. You can use this iRule in a virtual server that also offers a FIX profile and the low latency of Fast L4.

Creating a virtual server for low-latency electronic trading

After you create a server pool, profile(s), and (optionally) iRule, you need to create a virtual server that references those components.

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. From the **Type** list, select **Performance (Layer 4)**.
5. In the **Destination Address** field, type the IP address in CIDR format. This is the address to which the FIX clients send their FIX transmissions.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

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

6. From the **Configuration** list, select **Advanced**.
7. From the **Protocol** list, select **TCP**.
8. From the **Protocol Profile (Client)** list, select the custom Fast L4 profile you defined for low-latency FIX trading.
9. Go to the **FIX Profile** list and select the custom FIX profile you defined for low-latency trading.

10. (Optional) For the **Address Translation** setting, clear the **Enabled** check box to implement direct server return (DSR) functionality.
11. (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.*

12. In the Resources area of the screen, from the **Default Pool** list, select the pool name for FIX streams.

    This pool is for streams that do not match your iRule(s).
13. For the **iRules** setting, from the **Available** list, select the name of the iRule that you created for the Late Binding feature and move it to the **Enabled** list.

    The iRule enables load balancing based on the Layer-7 (FIX) fields at the head of each stream.
14. Click **Finished**.

The virtual server is configured to use the specified Fast L4 profile and pool. If a client initiates a FIX connection with this virtual server, the connection uses the Fast L4 (ePVA) hardware.

Implementation result

This implementation configures a BIG-IP® system to manage low-latency electronic trading functionality, optimizing the system for predictable latency and jitter. Clients who send FIX streams to the virtual server's Destination address all receive this low-latency service. The virtual server intelligently distributes the streams to different server pools based on information in each stream's FIX logon packet.

Implementing Hardware-optimized FIX Low Latency (FIX LL) Electronic Trading

About configuring BIG-IP systems for hardware-optimized FIX LL

You can configure BIG-IP® 10000 and 12000 Series systems that are running BIG-IP software version 12.1.0 to use a customized firmware to manage traffic for hardware-optimized, low-latency electronic trading. This FIX low latency (FIX LL) solution provides enhanced collision handling.

Important: *The FIX LL firmware does not support hardware SYN cookies. Be sure to enable software SYN cookie protection.*

Task summary

There are several tasks you can perform to implement hardware-optimized FIX low-latency electronic trading.

Task list

Licensing 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 specific license. The license must enable both of the following features:

- Advanced LTM® Protocols
- FIX Low Latency

Please contact your F5® Networks support representative to acquire the necessary license.

Selecting a firmware to use for hardware-optimized FIX LL

You can use the TMOS Shell (`tmsh`) to choose whether to use the FIX low latency (FIX LL) firmware.

1. Connect to the system using the serial console.
2. Log in to the command-line interface of the system using the root account.
3. Open the TMOS Shell (`tmsh`).
   ```
   tmsh
   ```
4. Enable the FIX LL firmware.
   ```
   modify sys db pva.fix.lowlatency value enable
   ```
5. Reboot the system to update the FPGA.
   reboot

   Important: *If you are switching to the FIX LL firmware for the first time, the system performs an HSB update to load the new firmware. Do not interrupt the progress of the firmware update. When the firmware update completes, the system reboots, and then you will be able to use the FIX LL firmware.*
6. (Optional) To switch back to the default firmware:
   a) Disable the FIX LL firmware.
      ```
      modify sys db pva.fix.lowlatency value disable
      ```
   b) Reboot the system to update the FPGA.
      ```
      reboot
      ```

Creating a custom Fast L4 profile for hardware-optimized FIX LL

You can create a custom Fast L4 profile to manage Layer 4 traffic for FIX low latency (FIX LL).

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. From the **PVA Acceleration** list, select **Dedicated**.
6. Select the **PVA Flow Aging** and **PVA Flow Evict** check boxes.
7. For the **PVA Offload Dynamic** setting, retain the default value (**Enabled**).
8. For the **PVA Dynamic Client Packets** setting, retain the default value (**1**).
9. For the **PVA Dynamic Server Packets** setting, retain the default value (**0**).
10. Select the **Loose Close** check box only for a one-arm virtual server configuration.
11. Set the **TCP Close Timeout** setting, according to the type of traffic that the virtual server will process.
12. Select the **Software SYN Cookie Protection** check box.
13. 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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) Click **Add**.
5. Click **Finished**.
6. Repeat these steps for each pool you want to create.

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. From the **Type** list, select **Performance (Layer 4)**.
5. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *The IP address you type must be available and not in the loopback network.*
6. From the **Configuration** list, select **Advanced**.
7. From the **Protocol** list, select **TCP**.
8. From the **Protocol Profile (Client)** list, select the custom Fast L4 profile you defined for low-latency FIX trading.
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 the relevant pool name.
12. Click **Finished**.

The virtual server is configured to use the specified Fast L4 profile and pool. If a client initiates a FIX connection with this virtual server, the connection uses the Fast L4 (ePVA) hardware.

Implementation result

This implementation configures a BIG-IP® system to manage low-latency electronic trading functionality, optimizing the system for predictable latency and jitter. Clients who send FIX packets to the virtual server's Destination address all receive this low-latency service.

Implementing APM System Authentication

Overview: Configuring authentication for a remote system based on APM

As an administrator in a large computing environment, you might prefer to store user accounts remotely, on a dedicated authentication server. When you want to use a remote server to authenticate traffic that manages a BIG-IP® system, you can store BIG-IP system administrative accounts on an AAA server. BIG-IP APM® supports AAA servers such as HTTP, LDAP, RADIUS, Active Directory, and TACACS+. To complete the authentication process, you must add the newly configured AAA action to an access policy. You can find more information about AAA authentication and access policies in *BIG-IP Access Policy Manager: Authentication and Single Sign-On* and *BIG-IP Access Policy Manager: Visual Policy Editor*.

Important: *System authentication using APM methods will not work if the user name and password contains Unicode characters (for example, Chinese characters) or the symbols ampersand (&), colon (:), less than (<), and apostrophe (').*

Creating a user authentication based on APM

Before you begin:

- Verify that the BIG-IP® system user accounts have been created on the remote authentication server.
- Verify that the appropriate user groups, if any, are defined on the remote authentication server.

You can configure the BIG-IP® system to use an APM® server for authenticating BIG-IP® system user accounts, that is, traffic that passes through the management interface (MGMT).

1. On the Main tab, click **System** > **Users**.
2. On the menu bar, click **Authentication**.
3. Click **Change**.
4. From the **User Directory** list, select **Remote - APM Based**.
5. For the **Access Profile** setting, click the + button.
   The screen refreshes to show general properties.
6. In the **Name** field, type a name for the access profile.

   Note: *An access profile name must be unique among all access profile and any per-request policy names.*

   ---

7. From the **Default Language** list, select a language.

   The default is **English (en)**.
8. From the **Authentication Type** list, select the type of authentication for the APM based remote user authentication.
   The screen refreshes to show areas and settings specific to the authentication type.
9. Fill in the fields.
10. Click **Finished**.

You can now authenticate administrative traffic for user accounts that are stored on a remote APM server. If you have no need to configure group-based user authorization, your configuration tasks are complete.

Example access policy using APM LDAP authentication

This is an example of an access policy with all the associated elements that are needed to authenticate and authorize your users with LDAP authentication.

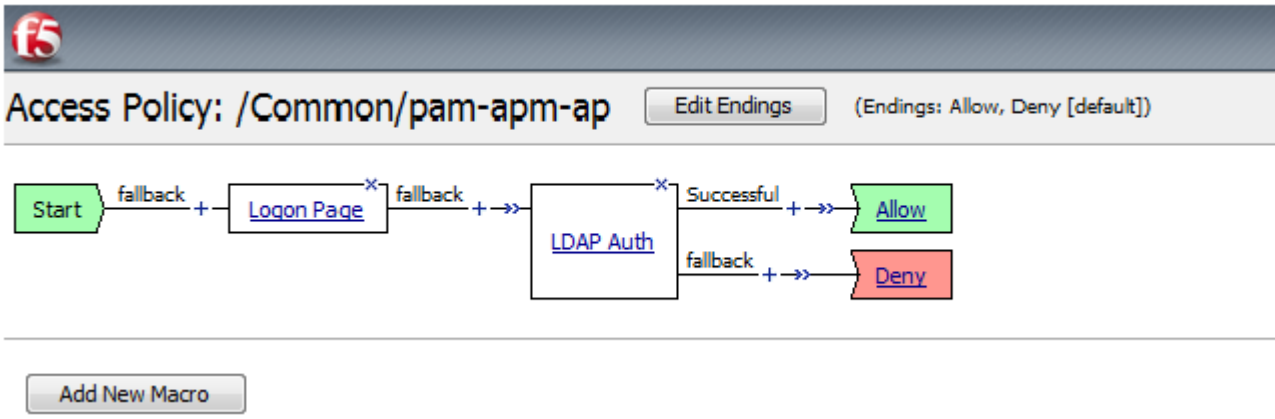


Figure 17: Example of an access policy for LDAP Auth

Configuring an SSL Intercept Explicit Proxy Mode

About SSL intercept explicit proxy mode

A typical SSL intercept explicity proxy mode configuration includes two BIG-IP devices, one configured to manage half-proxy client traffic and one configured to manage half-proxy server traffic. When the ingress BIG-IP system receives a client request, SSL decrypts the request. The ingress BIG-IP system then sends metadata to the egress BIG-IP system by means of the out-of-band TCP connection and sends the request data to the inspection device. When the egress BIG-IP system receives the metadata through the out-of-band connection and the request from the inspection device, it uses the information in the metadata, re-encrypts the request, and forwards it to the destination server.

The following illustration depicts an example configuration.

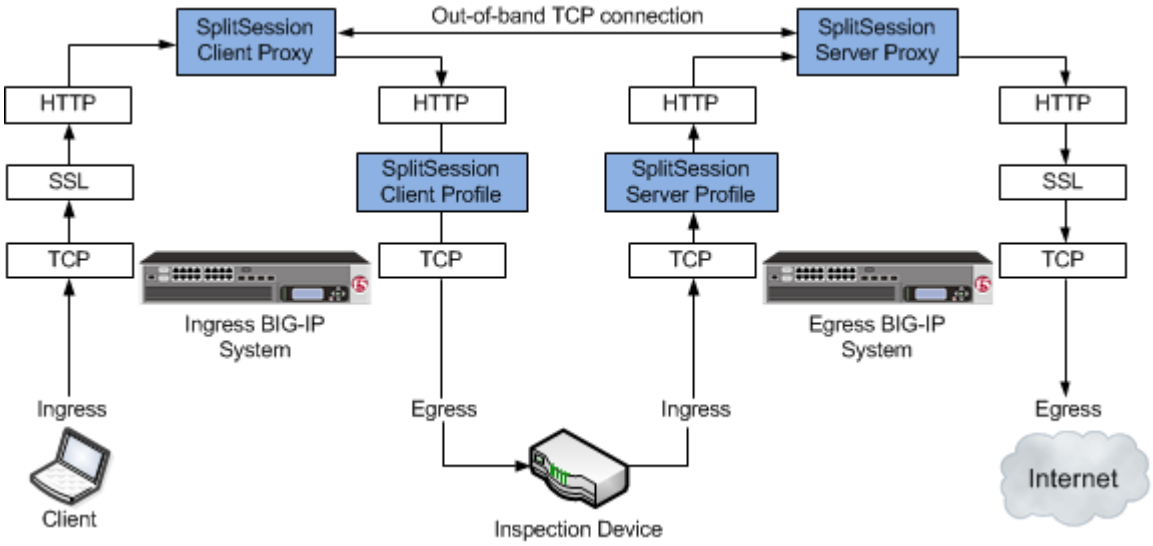


Figure 18: An example SSL intercept explicity proxy mode configuration

The SplitSession Client profile type

The SplitSession Client profile defines the client parameters in an SSL intercept explicit proxy mode configuration. This profile enables you to configure a Peer Port, which specifies the port for the SplitSession peer that is connected to the out-of-band connection, and the Peer IP address, which specifies the IP address for the SplitSession peer that is connected to the out-of-band connection.

The SplitSession Server profile type

The SplitSession Server profile defines the server parameters in an SSL intercept explicit proxy mode configuration. This profile enables you to configure a Listen Port, which specifies the port that the SplitSession server listens on for the out-of-band connection, and the Listen IP address, which specifies the IP address that the SplitSession server listens on for the out-of-band connection.

Task Summary

Complete these tasks to configure an SSL intercept explicit proxy configuration.

Creating a SplitSession Client profile

Creating a SplitSession Client profile

You can create a SplitSession Client profile to define the client parameters in an SSL intercept explicit proxy mode configuration.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Other** > **SplitSession Client**.
   The SplitSession Client profile list screen opens.
2. Click **Create**.
   The New SplitSession Client Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, retain the default value or select another existing profile of the same type.
5. In the **Peer Port** field, type a value for the the port of the SplitSession peer assigned to the out-of-band connection.
6. In the **Peer IP** field, type the IP address of the SplitSession peer assigned to the out-of-band connection.
7. Click **Finished**.

A SplitSession Client profile to define the client parameters in an SSL intercept explicit proxy mode configuration is available to assign to a virtual server.

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 SSL 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 other settings, as required.
8. Click **Finished**.

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

Creating a pool to process HTTP traffic for an inspection device

You can create a pool that includes an inspection device 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. 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**.
5. Click **Finished**.

The new pool appears in the Pools list.

Creating an ingress explicit proxy virtual server

Before you configure an ingress explicit proxy virtual server, you need to configure a SplitSession Client profile and pool to assign to the virtual server.

You can configure an ingress explicit proxy virtual server to manage the client split-session half-proxy traffic from a client to the inspection device.

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.
5. In the **Source Address** field, type `0.0.0.0/0` for the source address and prefix length.
6. In the **Destination Address** field, type an IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, to select all IP addresses, an IPv4 address/prefix is `0.0.0.0/0`, and an IPv6 address/prefix is `::/0`. To specify a network, an IPv4 address/prefix is `10.07.0.0` or `10.07.0.0/24`, and an IPv6 address/prefix is `ffe1::/64` or `2001:ed8:77b5::/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *For best results, F5® recommends that you enter the subnet that matches your destination server network.*

   ---
7. In the **Service Port** field, type `443` or select **HTTPS** from the list.
8. From the **HTTP Profile** list, select **http**.
9. For the **SSL Profile (Client)** setting, select a client SSL profile.
10. From the **Protocol** list, select **TCP**.
11. From the SplitSession Client Profile list, select **splitsessionclient** or a custom SplitSession Client profile.
12. From the **Default Pool** list, select the name of the HTTP server pool that you previously created.
13. Click **Finished**.

An ingress explicit proxy virtual server is configured to manage the client split-session half-proxy traffic from a client to the inspection device.

Creating a SplitSession Server profile

You can create a SplitSession Server profile to define the server parameters in an SSL intercept explicit proxy mode configuration.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Other** > **SplitSession Server**.
   The SplitSession Server profile list screen opens.
2. Click **Create**.
   The New SplitSession Server Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, retain the default value or select another existing profile of the same type.
5. In the **Listen Port** field, type a value for the the port of the SplitSession server listens on for the out-of-band connection.
6. In the **Listen IP** field, type the IP address of the SplitSession server listens on for the out-of-band connection.
7. Click **Finished**.

A SplitSession Server profile to define the server parameters in an SSL intercept explicit proxy mode configuration is available to assign to a virtual server.

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 Server SSL 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 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 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. For the **Health Monitors** setting, assign **https** or **https_443** by moving it from the **Available** list 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. Use the **New Members** setting to add each resource that you want to include in the pool:
   a) In the **Address** field, type an IP address.
   b) In the **Service Port** field type `443` , 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 appears in the Pool List screen.

Creating an egress explicit proxy virtual server

Before you configure an egress explicit proxy virtual server, you need to configure a SplitSession Server profile and pool to assign to the virtual server.

You can configure an egress explicit proxy virtual server to manage the server split-session half-proxy traffic from an inspection device to a 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 **Description** field, type a description of the virtual server.
5. In the **Source Address** field, type `0.0.0.0/0` for the source address and prefix length.
6. In the **Destination Address** field, type an IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, to select all IP addresses, an IPv4 address/prefix is `0.0.0.0/0`, and an IPv6 address/prefix is `::/0`. To specify a network, an IPv4 address/prefix is `10.07.0.0` or `10.07.0.0/24`, and an IPv6 address/prefix is `ffe1::/64` or `2001:ed8:77b5::/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

Note: *For best results, F5® recommends that you enter the subnet that matches your destination server network.*

7. In the **Service Port** field, type `443` or select **HTTPS** from the list.
8. For the **SSL Profile (Server)** setting, select a server SSL profile.
9. From the **Protocol** list, select **TCP**.
10. From the SplitSession Server Profile list, select **splitsessionserver** or a custom SplitSession Server profile.
11. From the **Default Pool** list, select the name of the HTTP server pool that you previously created.
12. Click **Finished**.

An egress explicit proxy virtual server is configured to manage the server split-session half-proxy traffic from an inspection device to a server.

Configuring an Explicit HTTP Proxy Chain

Overview: Configuring an explicit HTTP proxy chain

An explicit HTTP proxy chain configuration enables you to load balance traffic from a BIG-IP® device through a pool of proxy devices. When establishing an explicit HTTP proxy chain, the BIG-IP explicit proxy device sends an HTTP request to a remote proxy device, which connects to the requested host and port. Once the connection succeeds between the BIG-IP explicit proxy device and the remote proxy device, a tunnel is opened between the BIG-IP explicit proxy device and the remote proxy device, which allows other protocols to pass unimpeded through the tunnel.

The following illustration depicts a typical explicit HTTP proxy chain configuration.

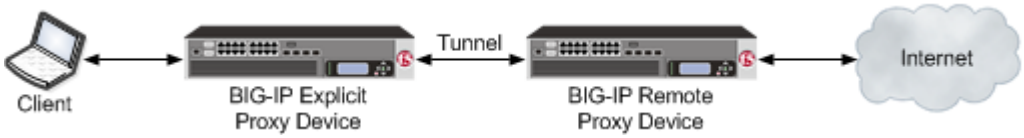


Figure 19: A typical explicit HTTP proxy chain configuration

About HTTP Proxy Connect profiles

The HTTP Proxy Connect profile enables a BIG-IP® device to connect to a remote, down-stream proxy device. A client connects to the BIG-IP device, which selects a remote proxy device from a pool of proxy devices. An HTTP CONNECT handshake tells the selected remote proxy device where to connect. When the connection is established, it becomes an opaque tunnel. Any protocol can use the tunnel between the BIG-IP device and the remote proxy.

When an HTTP profile is assigned to the virtual server, the HTTP CONNECT handshake is automatically configured. If an HTTP profile not assigned to the virtual server, for example, when you have opaque SSL traffic, you can use `HTTP::proxy chain` iRule commands to configure the destination to which the remote proxy device routes traffic.

Task Summary

Complete these tasks to configure an explicit HTTP proxy chain configuration.

Creating a custom HTTP Proxy Connect profile
Creating a load balancing pool
Creating a virtual server for explicit HTTP proxy connection

Creating a custom HTTP Proxy Connect profile

You can create a custom HTTP Proxy Connect profile and assign it to a virtual server to load balance HTTP traffic through a pool of proxy devices.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Other** > **HTTP Proxy Connect**.
   The **HTTP Proxy Connect** profile list screen opens.
2. Click **Create**.
   The New HTTP Proxy Connect Profile screen opens.
3. In the **Name** field, type a unique name for the profile.

4. From the **Parent Profile** list, retain the default value or select another existing profile of the same type.
5. Select the **Custom** check box.
6. Select the **Default State** check box.
7. Click **Finished**.

The custom HTTP Proxy Connect profile is available to assign to a virtual server.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of proxy devices that you group together to receive and process traffic) to efficiently distribute the load on your 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) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for explicit HTTP proxy connection

You can create a virtual server to load balance HTTP traffic through a pool of remote proxy devices.

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 Address** field, type the IP address in CIDR format.

The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

Note: *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. From the **HTTP Proxy Connect Profile** list, select a profile.
7. In the Resources area of the screen, from the **Default Pool** list, select the relevant pool name.

A virtual server is available to load balance HTTP traffic through a pool of remote proxy devices

Manipulating HTTPS Traffic by Using a Third-Party Device

Overview: Manipulating HTTPS traffic by using a third-party device

You can configure a BIG-IP® device to manage HTTPS traffic by using a third-party device that can intercept and modify the traffic, as necessary. This configuration provides SSL decryption, manipulation, and re-encryption while appearing relatively transparent at layer 2.

When you configure a virtual server to use the Transparent Nexthop control, traffic matching the virtual server is sent to the specified interface and the layer 2 addressing on the ingress packet is preserved. Configuring the Transparent Nexthop to specify the VLAN that is configured with the inspection device eliminates the need to configure a pool, NAT, SNAT, or other load balancing functionality to the inspection device.

Important: *Transparent Nexthop functionality requires a license that supports that functionality. If the Transparent Nexthop control does not appear on the New Virtual Server screen, contact your F5® Networks support representative to acquire the necessary license.*

The basic process used in this configuration is as follows:

1. A client sends an HTTPS request to a server by means of the BIG-IP device.
2. The BIG-IP device intercepts the request, decrypts it, and forwards the request as cleartext to the inspection device.
3. The inspection device receives and, as necessary, modifies the cleartext request.
4. The inspection device forwards the cleartext request to the server by means of the BIG-IP device.
5. The BIG-IP device re-encrypts the cleartext request and sends the ciphertext request to the server.
6. The server sends a response to the client by means of the BIG-IP device.
7. The BIG-IP device receives the response, decrypts it, and forwards the response as cleartext to the inspection device.
8. The inspection device receives and, as necessary, modifies the cleartext response.
9. The inspection device forwards the cleartext response to the client by means of the BIG-IP device.
10. The BIG-IP device re-encrypts the cleartext response and sends the ciphertext response to the client.

The following illustration shows an example of a BIG-IP device that manages HTTPS traffic modified by a third-party device.

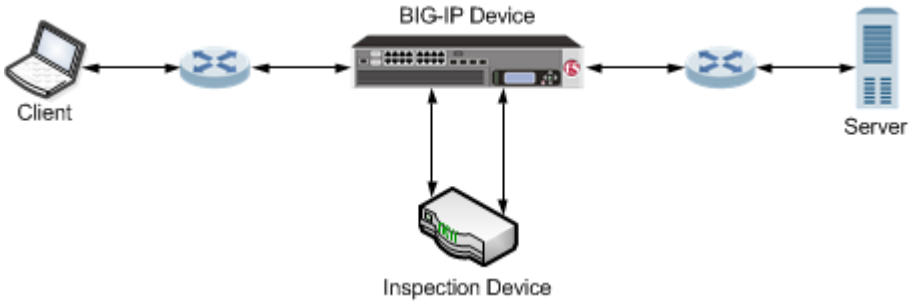


Figure 20: An example configuration of a BIG-IP device managing HTTPS traffic modified by a third-party device.

Task Summary

Complete these tasks to configure a BIG-IP® device to manage HTTPS traffic by using a third-party device that can intercept and modify the traffic, as necessary.

Creating a VLAN

When you create a VLAN, each of the specified interfaces can process traffic destined for that VLAN. You can create a VLAN for use with an inspection device, as necessary.

1. On the Main tab, click **Network** > **VLANs**.
   The VLAN List screen opens.
2. Click **Create**.
   The New VLAN screen opens.
3. In the **Name** field, type a unique name for the VLAN.
4. In the **Tag** field, type a numeric tag, between 1-4094, for the VLAN, or leave the field blank if you want the BIG-IP system to automatically assign a VLAN tag.

   The VLAN tag identifies the traffic from hosts in the associated VLAN.
5. For the **Interfaces** setting:
   a) From the **Interface** list, select an interface number.
   b) From the **Tagging** list, select **Tagged**.
   c) Click **Add**.
6. If you want the system to verify that the return route to an initial packet is the same VLAN from which the packet originated, select the **Source Check** check box.
7. In the **MTU** field, retain the default number of bytes (**1500**).
8. From the **Configuration** list, select **Advanced**.
9. For the **Hardware SYN Cookie** setting, select or clear the check box.

   When you enable this setting, the BIG-IP system triggers hardware SYN cookie protection for this VLAN.

   Enabling this setting causes additional settings to appear. These settings appear on specific BIG-IP platforms only.
10. For the **Syncache Threshold** setting, retain the default value or change it to suit your needs.

    The **Syncache Threshold** value represents the number of outstanding SYN flood packets on the VLAN that will trigger the hardware SYN cookie protection feature.

    When the **Hardware SYN Cookie** setting is enabled, the BIG-IP system triggers SYN cookie protection in either of these cases, whichever occurs first:

    - The number of TCP half-open connections defined in the LTM® setting **Global SYN Check Threshold** is reached.
    - The number of SYN flood packets defined in this **Syncache Threshold** setting is reached.
11. For the **SYN Flood Rate Limit** setting, retain the default value or change it to suit your needs.

    The **SYN Flood Rate Limit** value represents the maximum number of SYN flood packets per second received on this VLAN before the BIG-IP system triggers hardware SYN cookie protection for the VLAN.
12. Click **Finished**.
    The screen refreshes, and displays the new VLAN in the list.

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 SSL 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 other settings, as required.
8. Click **Finished**.

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

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 Server SSL 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 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 VLAN group

Create a VLAN group that includes the internal and external VLANs using transparent mode. Packets received by a VLAN in the VLAN group are copied onto the other VLAN. This allows traffic to pass through the BIG-IP® system on the same IP network.

1. On the Main tab, click **Network** > **VLANs** > **VLAN Groups**.
   The VLAN Groups list screen opens.
2. Click **Create**.
   The New VLAN Group screen opens.
3. In the **Name** field, type the name `myvlangroup`.
4. For the **VLANs** setting, from the **Available** list, select **internal** and **external**, and then move them to the **Members** list.
5. From the **Transparency Mode** list, select **Transparent**.
6. Click **Finished**.

Creating a virtual server to manage client-side HTTPS traffic

You can specify a virtual server that manages client-side HTTPS traffic sent to a third-party device to manipulate traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
2. Click **Create**.
3. In the **Name** field, type a name for the virtual server.
4. From the **Type** list, select **Standard**.
5. In the **Destination Address/Mask** field, type a destination IP address in CIDR format.
6. For the **Service Port** setting, type `443` in the field, or select **HTTPS** from the list.
7. From the **Protocol Profile (Client)** list, select **splitsession-default-tcp**.
8. From the **Configuration** list, select **Advanced**.
9. From the **HTTP Profile** list, select **http**.
10. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you created previously, and using the Move button, move the name to the **Selected** list.
11. From the **VLAN and Tunnel Traffic** list, select **Enablerd on**.
12. For the **VLANs and Tunnels** setting, move the clientside VLAN to the **Selected** list.
13. From the **Transparent Nexthop** list, select the VLAN that you created for the inspection device.
14. Click **Finished**.

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

Creating a virtual server to manage server-side traffic

You can specify a virtual server that manages server-side traffic sent from a third-party device to manipulate traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
2. Click **Create**.
3. In the **Name** field, type a name for the virtual server.
4. From the **Type** list, select **Standard**.
5. In the **Destination Address/Mask** field, type a destination IP address in CIDR format.
6. For the **Service Port** setting, type `80` in the field, or select **HTTP** from the list.

7. From the **Configuration** list, select **Advanced**.
8. From the **Protocol Profile (Server)** list, select **splitsession-default-tcp**.
9. From the **HTTP Profile** list, select **http**.
10. For the **SSL Profile (Server)** setting, from the **Available** list, the name of the Server SSL profile you created previously, and using the Move button, move the name to the **Selected** list.
11. From the **VLAN and Tunnel Traffic** list, select **Enabled on**.
12. For the **VLANs and Tunnels** setting, move the VLAN that you created for the inspection device to the **Selected** list.
13. From the **Transparent Nexthop** list, select the serverside VLAN.
14. Click **Finished**.

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

Legal Notices

Legal notices

Publication Date

This document was published on August 13, 2018.

Publication Number

MAN-0293-15

Copyright

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*.

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.

Index

F

H

I

L

M

N

O

P

R

S

T

U

V

W

X