All BIG-IP solutions include either BIG-IP Local Traffic Manager (LTM) or BIG-IP DNS (formerly BIG-IP Global Traffic Manager). Each provide powerful core functionality to your application services.

More Power, On Demand
# Contents

## About This Guide
- Before using this guide
- Limits of this guide
- Glossary
- Customization
- Issue escalation
- Feedback and notifications
- Configuration utility
- Command-line syntax
- Finding other documents

## Introduction
- Core concepts
- BIG-IP LTM nomenclature
- Topologies
- CMP/TMM basics

## BIG-IP LTM Load Balancing
- Load balancing types
- Monitors
- Troubleshoot load balancing problems

## BIG-IP LTM network address objects
- Virtual address
- Address translation
- Self IP address
- IPv4/IPv6 gateway behavior
- Auto Last Hop
- Service Provider protocols

## BIG-IP LTM Virtual Servers
- Virtual servers as listeners
- Address matching precedence
Translation options 43
Traffic group interaction 43
Clone pools 44
Profile use by virtual servers 44
Common Virtual server types 44
Other virtual server types 45
Virtual server considerations 46
Virtual server troubleshooting 48

**BIG-IP LTM Profiles** 52
Protocol profiles 52
TCP 53
OneConnect 54
HTTP profiles 56
HTTP Compression profile 57
Web Acceleration profile 58
SSL Profiles 58
BIG-IP LTM policies 61
Other protocols and profiles 65
Troubleshooting 66

**BIG-IP DNS/DNS Services** 67
DNS Services features 67
Upgrading to BIG-IP DNS 12.0 and later 67
BIG-IP DNS/DNS Services Basics 68
BIG-IP DNS/DNS Services Core Concepts 69
BIG-IP DNS load balancing 80
BIG-IP DNS Minimal Response Setting 85
BIG-IP DNS Architectures 89
BIG-IP DNS iQuery 90
BIG-IP DNS Device Service Clustering 94
BIG-IP DNS query logging 94
BIG-IP DNS Statistics 95

**iRules** 97
iRules anatomy 97
### CONTENTS

- iRules considerations ........................................... 102
- iRules troubleshooting ........................................ 103

### Logging ......................................................... 106
- Logging levels .................................................. 106
- Local logging .................................................. 106
- Remote Logging ............................................... 107
- Logging considerations ...................................... 109

### Optimizing the Support Experience .......................... 110
- F5 technical support commitment .......................... 110
- F5 certification ............................................... 111
- Self-help ....................................................... 112
- F5 training programs and education ...................... 115
- Engage F5 Support ........................................... 115

### Legal Notices .................................................. 126
- Trademarks .................................................... 126
- Patents .......................................................... 126
- Notice ......................................................... 126
- Publication Date .............................................. 127
- Copyright ...................................................... 127

### Change List ..................................................... 128
## Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>F5 documentation coverage</td>
<td>2</td>
</tr>
<tr>
<td>1.1</td>
<td>BIG-IP system full proxy architecture</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>One-armed topology</td>
<td>7</td>
</tr>
<tr>
<td>1.3</td>
<td>Two-armed topology</td>
<td>8</td>
</tr>
<tr>
<td>1.4</td>
<td>Two-armed topology without source address</td>
<td>9</td>
</tr>
<tr>
<td>1.5</td>
<td>Two-armed topology with source address translation</td>
<td>10</td>
</tr>
<tr>
<td>1.6</td>
<td>nPath topology</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>Ratio load balancing</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>Least Connections load balancing</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>Load balancing using Priority Group Activation</td>
<td>16</td>
</tr>
<tr>
<td>2.4</td>
<td>Load balancing using Priority Group Activation, with nodes down</td>
<td>17</td>
</tr>
<tr>
<td>3.1</td>
<td>Connections flow using NAT</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Connections flow using SNAT</td>
<td>33</td>
</tr>
<tr>
<td>5.1</td>
<td>OneConnect source mask connection reuse</td>
<td>55</td>
</tr>
<tr>
<td>5.2</td>
<td>SSL offload</td>
<td>59</td>
</tr>
<tr>
<td>5.3</td>
<td>SSL re-encryption</td>
<td>59</td>
</tr>
<tr>
<td>6.1</td>
<td>Cache miss</td>
<td>72</td>
</tr>
<tr>
<td>6.2</td>
<td>Cache hit</td>
<td>72</td>
</tr>
<tr>
<td>6.3</td>
<td>Resolver DNS cache</td>
<td>73</td>
</tr>
<tr>
<td>6.4</td>
<td>DNSSEC chain of trust</td>
<td>77</td>
</tr>
</tbody>
</table>
FIGURES—

Figure 6.5: Wide IP with A and AAAA virtual servers in a single pool 82
Figure 6.6: Wide IP with CNAME pool configuration 83
Figure 6.7: “A” Resource Record configuration 83
Figure 6.8: “AAAA” Resource Record configuration 84
Figure 6.9: CNAME For Type A Resource Record 84
Figure 6.10: CNAME For Type A Resource Record with Static Target 84
Figure 6.11: Type MX Wide IP 85
Figure 6.12: SRV Wide IPs configuration 85
Figure 6.13: NAPTR type Wide IP configuration 85
Figure 6.14: BIG-IP LTM/BIG-IP DNS configuration 91
Figure 8.1: Remote logging using high-speed logging 108
Tables

Table 0.1 Command-line syntax  3
Table 4.1 Virtual servers  42
Table 4.2 Connection table for virtual servers  43
Table 4.3 Standard and Performance (Layer 4) virtual servers use case criteria  47
Table 4.4 tcpdump commands  49
Table 5.1 Common BIG-IP LTM profiles  52
Table 5.2 BIG-IP 13.0 and later base profiles  54
Table 5.2 HTTP proxy modes  56
Table 5.3 Common HTTP Compression profile options  58
Table 5.4 Common SSL profile options  60
Table 5.5 Profile types  65
About This Guide

The goal of this guide is to help F5® customers keep their BIG-IP® system healthy, optimized, and performing as designed. It was written by F5 engineers who assist customers with solving complex problems every day. Some of these engineers were customers before joining F5, and their unique perspective and hands-on experience serves the guides F5 customers have requested.

This guide describes common information technology procedures, as well as those which are exclusive to BIG-IP systems. There may be procedures particular to your industry or business that are not identified. While F5 recommends the procedures outlined in this guide, they are intended to supplement your existing operations requirements and industry standards. F5 suggests that you read and consider the information provided to find the procedures to suit your implementation, change-management process, and business-operations requirements. Doing so can result in higher productivity and fewer unscheduled interruptions.

Refer to “Feedback and notifications” for information on how to help improve future versions of the guide.

Before using this guide

To get the most out of this guide, first complete the following steps, as appropriate to your implementation:

- Install your F5 platform according to its requirements and recommendations. Search the AskF5™ (support.f5.com) for “platform guide” to find the appropriate guide.
- Follow the general environmental guidelines in the hardware platform guide to make sure of proper placement, airflow, and cooling.
- Set recommended operating thresholds for your industry, accounting for predictable changes in load. For assistance contact F5 Professional Services (f5.com/support/professional-services).
- Familiarize yourself with F5 technology concepts and reviewed and applied appropriate recommendations from F5 BIG-IP TMOS: Operations Guide.

Note For information about how to locate F5 product manuals, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

Limits of this guide

This guide does not focus on installation, setup, or configuration of your BIG-IP system or modules. There is a wealth of documentation covering these areas in AskF5 (support.f5.com) The F5 self-help community, DevCentral™ (devcentral.f5.com), is also a good place to find answers about initial deployment and configuration.

The following figure shows where the F5 operations guides can best be applied in the product life cycle.
Glossary

A glossary is not included in this guide. Instead, the Glossary and Terms page (f5.com/glossary) offers an up-to-date and complete listing and explanation of common industry and F5-specific terms.

Customization

Customization may benefit your implementation. You can get help with customization from a subject matter expert, such as a professional services consultant, from F5 Professional Services (f5.com/support/professional-services).

Issue escalation

Refer to Optimizing the Support Experience for issue escalation information.

If you have an F5 websupport contract, you can open a support case by clicking Contact support on AskF5 (support.f5.com)
Feedback and notifications

F5 frequently updates the operations guides and new guides may be released as needed. If you would like to be notified when new or updated content is available, or if you have feedback, corrections, or suggestions to improve this guide, email opsguide@f5.com. F5 internal users can file a request using Service-Now.

Configuration utility

The BIG-IP Configuration utility is the name of the graphic user interface (GUI) of the BIG-IP system and its modules. It is a browser-based application you can use to install, configure, and monitor your BIG-IP system.

For more information about the Configuration utility, refer to Introducing BIG-IP Systems in BIG-IP Systems: Getting Started Guide.

Note For information about how to locate F5 product manuals, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

Command-line syntax

We show command line input and output in courier font. The corresponding prompt is not included. For example, the following command shows the configuration of the specified pool name:

```
tmsh show /ltm pool my_pool
```

The following table explains additional special conventions used in command-line syntax:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&gt;</td>
<td>Identifies a user-defined variable parameter. For example, if the command has <code>&lt;your name&gt;</code>, type in your name but do not include the brackets.</td>
</tr>
<tr>
<td>[]</td>
<td>Indicates that syntax inside the brackets is optional.</td>
</tr>
<tr>
<td>...</td>
<td>Indicates that you can type a series of items.</td>
</tr>
</tbody>
</table>

TMOS Shell syntax

The BIG-IP system includes a utility known as the TMOS® Shell (tmsh) that you can use to configure and manage the system at the command line. Using tmsh, you can configure system features and set up network elements. You can also configure the BIG-IP system to manage local and global traffic passing through the system and view statistics and system performance data.

You can run tmsh and issue commands in the following ways:

- You can issue a single tmsh command at the BIG-IP system command line using the following syntax:

  ```
tmsh [command] [module ... module] [component] [options]
  ```
You can open `tmsh` by typing `tmsh` at the BIG-IP system command line:

```
(tmsh)#
```

At the `tmsh` prompt, you can issue the same command syntax, leaving off `tmsh` at the beginning.

**Note** You can use the command line utilities directly on the BIG-IP system console, or you can run commands using a remote shell, such as the SSH client or a Telnet client. For more information about command line utilities, refer to the *Traffic Management Shell (tmsh) Reference Guide*.

Finding other documents

For information about how to locate F5 product manuals, refer to AskF5 article: [K12453464: Finding product documentation on AskF5](https://askf5.com/kb/K12453464).
Introduction

The BIG-IP® Local Traffic Manager™ (LTM®) module manages and optimizes traffic for network applications and clients.

Some typical tasks that BIG-IP LTM can perform include:

- Automatically balancing application traffic amongst multiple servers
- Caching and compressing HTTP content
- Inspecting and transforming application content
- Offloading SSL decryption to BIG-IP LTM
- Isolating applications from harmful network traffic

BIG-IP LTM runs on TMOS®, which is the base platform software on all F5® hardware platforms and BIG-IP Virtual Edition (VE).

F5 hardware platforms include acceleration hardware, which may include SSL acceleration, compression acceleration, and an embedded Packet Velocity Accelerator chip (ePVA), which can significantly increase the available throughput of BIG-IP LTM.

BIG-IP VE is available for several popular hypervisor products, which allows BIG-IP LTM to be integrated into a virtual data center infrastructure. Additionally, it is available on a number of leading cloud providers.

Core concepts

BIG-IP LTM treats all network traffic as stateful connection flows. Even connectionless protocols such as User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP) are tracked as flows. BIG-IP LTM is a default-deny device: unless traffic matches a configured policy, it is rejected. The BIG-IP system acts as a full proxy. Therefore, connections through BIG-IP LTM are managed as two distinct connection flows: a client-side flow and a server-side flow. The following figure models the BIG-IP system full proxy architecture.

![Figure 1.1: BIG-IP system full proxy architecture](image-url)
BIG-IP LTM nomenclature

- **Node**: A configuration object represented by the IP address of a device on the network.

- **Pool member**: A node and service port to which BIG-IP LTM can load balance traffic. Nodes can be members of multiple pools.

- **Pool**: A configuration object that groups pool members together to receive and process network traffic in a fashion determined by a specified load balancing algorithm.

- **Monitor**: A configuration object that checks the availability or performance of network resources such as pool members and nodes.

- **Virtual server**: A virtual server allows BIG-IP systems to send, receive, process, and relay network traffic.

- **Profile**: Configuration objects that can be applied to virtual servers to affect the behavior of network traffic.

Topologies

BIG-IP systems can be deployed to a network with little to no modification to existing architecture. However, to optimize the performance of your network and applications, some environment changes may be required to take full advantage of the multipurpose functionality of BIG-IP systems.

There are three basic topologies for load-balanced applications with BIG-IP LTM: one-armed, two-armed, and nPath. nPath is also known as direct server return (DSR).

**One-armed**

In a one-armed topology, the virtual server is on the same subnet and VLAN as the pool members.

Source address translation must be used in this configuration to ensure that server response traffic returns to the client by way of BIG-IP LTM.
One-armed topology details

- Allows for rapid deployment.
- Requires minimal change to network architecture to implement.
- Allows for full use of BIG-IP LTM feature set.
- Does not allow asymmetrical routing of server traffic. Because the system translates the source address, the client IP address is not visible to pool members. (You can change this for HTTP traffic by using the X-Forwarded-For header).

Two-armed

In the two-armed topology, the virtual server is on a different VLAN from the pool members. The BIG-IP system routes traffic between them.
Source address translation may or may not be required, depending on overall network architecture. If the network is designed so that pool member traffic is routed back to BIG-IP LTM, it is not necessary to use source address translation.

**Figure 1.3: Two-armed topology**

*Source address translation in a two-armed topology*

If pool member traffic is not routed back to BIG-IP LTM, it is necessary to use source address translation to ensure it is translated back to the virtual server IP.

The following figure shows a two-armed deployment without source address translation:
Figure 1.4: Two-armed topology without source address

The following figure shows the same deployment *with* source address translation:
Two-armed topology details

- Allows preservation of client source IP.
- Allows for full use of BIG-IP LTM feature set.
- Allows BIG-IP LTM to protect pool members from external exploitation.
- May require network topology changes to ensure return traffic traverses BIG-IP LTM.

nPath

In the nPath or direct server return (DSR) topology, return traffic from pool members is sent directly to clients without first traversing the BIG-IP LTM. This allows for higher theoretical throughput because BIG-IP LTM only manages the incoming traffic and does not process the outgoing traffic.

This deployment significantly reduces the available BIG-IP LTM features. For more information, refer to Configuring nPath Routing in BIG-IP Local Traffic Manager: Implementations.
**nPath topology details**

- Allows maximum theoretical throughput.
- Preserves client IP addresses to pool members.
- Limits availability of usable features of BIG-IP LTM and other modules.
- Requires modification of pool members and network.
- Requires more complex troubleshooting.

**CMP/TMM basics**

Data plane traffic on BIG-IP LTM is handled by the Traffic Management Microkernel (TMM). TMM operates on the
concept of Clustered Multiprocessing™ (CMP®), which creates multiple TMM process. To achieve high performance network traffic processing, a proprietary F5 layer-4 algorithm distributes traffic to the various TMMs.

For more information about CMP, refer to the following AskF5™ articles:

- **K14358: Overview of Clustered Multiprocessing - (11.3.0 and later)**
- **K14248: Overview of Clustered Multiprocessing - (11.0.0 - 11.2x)**
BIG-IP LTM Load Balancing

BIG-IP® systems are designed to distribute client connections to load balancing pools, which are typically made up of multiple pool members. The selected load balancing method (or algorithm) determines how connections are distributed across pool members.

Your choice of load balancing method may depend on factors such as pool member availability or session affinity:

- Some load balancing methods distribute connections evenly across pool members, regardless of the current workload of pool members or nodes. Such methods work best with homogeneous server pools and workloads.
- Some load balancing methods favor higher performing servers. These methods may result in deliberately uneven traffic distribution across pool members.
- Some load balancing methods consider factors which change at run time, such as current connection count or capacity.

Load balancing types

Load balancing methods fall into one of two distinct categories: static or dynamic.

**Static load balancing** methods distribute incoming connections in a uniform and predictable manner regardless of load factor or current conditions.

**Dynamic load balancing** methods distribute connections by factoring in the current conditions when making a load balancing decision.

**Static load balancing methods**

**Round Robin** load balancing is the default method on the BIG-IP system. It is static because, except for pool member status, the system does not use dynamic run-time information to select the next pool member. Round Robin works best when the pool members are roughly equal in processing and memory capacity and application requests use server resources uniformly.

**Ratio** load balancing distributes new connections across available members in proportion to a user-defined ratio using a weighed Round Robin pattern. The ratio can be set for each member or each node.

Ratio load balancing is useful when pool members have varying available capacity. For example, if a pool contains one fast server and three slower servers, the ratio can be set so the fast server receives more connections.

The following figure shows a load balancing example with connections distributed in a 3:2:1:1 ratio.
Dynamic load balancing methods

**Least Connections** load balancing distributes new connections across available members, based on the current connection count between BIG-IP system and the server. It does not account for connections the servers may have with other systems.

This mode is a popular general-purpose distribution method. It can be especially useful when supporting long-lived connections like FTP and TELNET. Over time, connections are distributed relatively evenly across the pool members.

If multiple pool members currently have a similar number of low connections, traffic is distributed in a Round-Robin pattern.

The following figure shows how six connections would be distributed, with the connection counts shown below each pool member and assuming that no connections terminate.
**Fastest** load balancing distributes new connections to the member or node that currently has the fewest outstanding layer 7 (L7) requests. If the virtual server lacks either a TCP or L7 profile, BIG-IP LTM® cannot track requests and responses. In this case, load balancing falls back to Least Connections load balancing distribution.

Fastest load balancing is useful for distributing traffic to pool members that may have varying response times due to load that previous requests have placed on the system. Over time, connections are distributed relatively evenly if all servers have similar capabilities.

There are other static and dynamic load balancing methods. For more information about additional methods, search AskF5™ (support.f5.com).

**Priority Group Activation**

Priority Group Activation load balancing allows pool members to be used only if preferred pool members are
unavailable. The BIG-IP system assigns each pool member a priority and sends connections to the highest priority pool members first. The system assigns a minimum number of available members, and if fewer pool members become available, the system activates the next highest priority pool members.

For example, in the following figure, the system assigns three physical hardware pool members as **Priority 10**. Three additional virtual machines are deployed as backup and assigned as **Priority 5**.

If the Priority Group Activation setting is 2, when all of the pool members are available, the system uses only the physical nodes in the **Priority 10** group.

Figure 2.3: Load balancing using Priority Group Activation

However, if fewer than 2 **Priority 10** pool members become available, the system automatically uses virtual machines in **Priority 5** group.
Priority Group Activation does not modify persistence behavior. Any new connections sent to lower-priority members continue to persist even when higher priority members become available again.

**FallBack host (HTTP only)**

If all members of a pool fail, virtual servers configured with a FallBack host in an attached HTTP profile can send an HTTP redirect message to the client. This allows you to send connections to an alternate site or to an apology server. You configure this through the HTTP profile.
Monitors

The BIG-IP system uses monitors to check whether or not pool members are eligible to service application traffic. Monitors periodically send specific requests to pool members and evaluate their health based on the members' response or lack thereof. Monitors can make explicit requests to an application, causing the application to perform an action which, in turn, tests vital server resources of that application, such as a SQL database.

For more information about monitors, refer to BIG-IP Local Traffic Manager: Monitors Reference or DevCentral™.

Monitor benefits

When implemented properly, a monitor can alert you to stability and availability issues that may exist with an application or arise as the result of a change to application infrastructure.

You can also use monitors to remove pool members from load balancing during scheduled maintenance windows. This is preferable to manually disabling the pool members because it does not require application owners to have operator-level access to the BIG-IP system and it can be done without requiring syncing your configuration when you remove and return the pool member to service.

Monitor components

BIG-IP systems include native support for a wide number of protocols and proprietary applications and services, including TCP, UDP, HTTP, HTTPS, SIP, LDAP, SOAP, MSSQL, MySQL, and others.

Generally, monitors are each configured with a send string and an expected receive string. You also have the option to configure monitors to send to and receive from alias hosts and ports, if necessary. Each monitor type contains further configurable profile information, which may be simple or complex, depending on the type.

If the available options for a particular monitor don’t suit your needs, custom monitors can be created and run using external scripts or by constructing custom strings which are sent to the application over TCP or UDP.

Monitor implementation

In typical usage, a monitor sends its send string to a pool member using the same protocol as normal traffic.

For example, in the case of an HTTP-based application, the monitor makes an HTTP request to a web page on the pool member. If the pool member sends a timely response, the response is compared with the receive string configured in the monitor profile.

Tip F5 recommends using a maintenance monitor with an application health monitor.

Choose an effective send string

F5 recommends that you configure send strings and receive strings as explicitly as possible.

HTTP receive strings are matched against both HTTP content (for example, HTML, JSON, or plain text) and the
HTTP headers. For more information, refer to AskF5 article: K3618: BIG-IP HTTP health monitors match receive string against payload and HTTP headers (including cookies).

F5 recommends against matching receive strings against the HTTP response code (such as 200 for a good response) because some web applications generate 404 or page not found errors using the 200 response code. Additionally, other headers may include 200, such as Content-Length: 120055 or a Set-Cookie: header containing those digits.

To avoid unexpected matches, F5 recommends explicitly setting the receive string to HTTP/1.1 200 OK (using HTTP 1.1).

Some applications have built-in uniform response identifiers (URIs) that can check application health. You may need to contact the application vendor to find out whether or not URIs are implemented.

For custom applications, you can build a page that runs a test suite and responds to BIG-IP system health checks. For example:

```plaintext
ltm monitor http finance.example.com _ health _ http _ monitor {
    adaptive disabled
    defaults-from http
    destination *:*
    interval 5
    ip-dscp 0
    recv "HTTP/1.1 200 OK"
    send "GET /app-health-check/status HTTP/1.1\r\nHost: finance.example.com\r\nConnection: close\r\n\r"
    time-until-up 0
    timeout 16
}
```

Enabling or disabling pool members using monitors

You can enable or disable a pool member using a monitor instead of logging in to the BIG-IP system and doing it manually. Because operator-level access is not required to enable or disable a pool member using a monitor, you can delegate this task to application-owner teams while maintaining your access security.

For example, you can create an additional pool-level monitor to check for a given URI on the server’s file system using HTTP. Checking for this file in a known good location tells the BIG-IP system whether or not the application owners want the pool member to serve traffic. It provides an easy way to remove the pool member from load balancing by either renaming it or moving it to an alternate location.

For example:

```plaintext
ltm monitor http finance.example.com _ enable _ http _ monitor {
```
adaptive disabled
defaults-from http
destination *:*interval 5
ip-dscp 0
recv “HTTP/1.1 200 OK”
send “GET /app-health-check/enable HTTP/1.1
Host: finance.example.com
Connection: close”
time-until-up 0
timeout 16
}
ltm monitor http finance.example.com _ health _ http _ monitor {
adaptive disabled
defaults-from http
destination *:*interval 5
ip-dscp 0
recv “HTTP/1.1 200 OK”
send “GET /app-health-check/status HTTP/1.1
Host: finance.example.com
Connection: close”
time-until-up 0
timeout 16
}
ltm pool finance.example.com _ pool {
members {
10.1.1.10:http {
address 10.1.1.10
session monitor-enabled
state up
}
10.1.1.11:http {
address 10.1.1.11
session monitor-enabled
state down

10.1.1.12:http {
  address 10.1.1.12
  session monitor-enabled
  state down
}

monitor finance.example.com _ enable _ http _ monitor and finance.example.com _ health _ http _ monitor

You can pull pool members from load balancing if the site is determined to be unhealthy or if an application owner renames the /app-health-check/enable file. You can also use a disable string to disable the pool member. You can use this method if the health check page is dynamic and the application returns different content when unavailable or in maintenance mode.

Monitoring multiple applications or websites

If your servers host multiple applications or websites, F5 recommends monitoring each virtual host application separately so that each can be marked up or down independently of the others. This requires either an individual virtual server for each virtual host or an iRules® iRule to match virtual hosts to pools as appropriate.

For example, a set of web servers host finance.example.com, hr.example.com, and intranet.example.com as virtual hosts. The following is sample output from a configuration in which all three sites are configured together as a single pool, with multiple monitors:

```
ltm pool example.com _ pool {
  members {
    10.1.1.10:http {
      address 10.1.1.10
      session monitor-enabled
      state checking
    }
    10.1.1.11:http {
      address 10.1.1.11
      session monitor-enabled
```
state checking
}

10.1.1.12:http {
   address 10.1.1.12
   session monitor-enabled
   state checking
}

}
mmonitor finance.example.com_http_monitor and hr.example.com_http_monitor and intranet.example.com_http_monitor

}

Compare the previous example with the following sample output, in which each site is configured with a separate pool, each with its own set of monitors:

ltm pool finance.example.com_pool {
   members {
      10.1.1.10:http {
         address 10.1.1.10
         session monitor-enabled
         state down
      }
      10.1.1.11:http {
         address 10.1.1.11
         session monitor-enabled
         state down
      }
      10.1.1.12:http {
         address 10.1.1.12
         session monitor-enabled
         state down
      }
   }
}
monitor finance.example.com _ http _ monitor

ltm pool hr.example.com _ pool {
members {
10.1.1.10:http {
address 10.1.1.10
session monitor-enabled
state down
}
10.1.1.11:http {
address 10.1.1.11
session monitor-enabled
state down
}
10.1.1.12:http {
address 10.1.1.12
session monitor-enabled
state down
}
}

monitor hr.example.com _ http _ monitor

ltm pool intranet.example.com _ pool {
members {
10.1.1.10:http {
address 10.1.1.10
session monitor-enabled
state down
}
10.1.1.11:http {
address 10.1.1.11
}}
The latter configuration allows for more refined monitoring and better control.

**Monitoring performance**

Monitors can check system performance by measuring pool member loads and response times. You can use Simple Network Management Protocol (SNMP) and Windows Management Instrumentation (WMI) monitors to evaluate server load. Additionally, the BIG-IP system includes an adaptive monitoring setting to evaluate server response time.

Adaptive monitoring allows you to require a pool member to be removed from load balancing eligibility if it does not pass a configured monitor requirement. For example:

```
ltm monitor http finance.example.com _ http _ monitor {
    adaptive enabled
    adaptive-limit 150
    adaptive-sampling-timespan 180
    defaults-from http
    destination *:*
    interval 5
    ip-dscp 0
    recv "HTTP/1.1 200 OK"
    send "GET /finance/app-health-check/\r\n\r\nHost: finance.example.com\r\nConnection: close\r\n"
    time-until-up 0
    timeout 16
}
```
Building send and receive strings

You can use tools such as curl to automatically build and send HTTP headers and to make that information visible in a useful fashion. For example:

```bash
$ curl -v -H "Host: finance.example.com" http://10.1.1.10/app-health-check/status
* STATE: INIT => CONNECT handle 0x8001f418; line 1028 (connection #0)
* Hostname was NOT found in DNS cache
* Trying 10.1.1.10...
* STATE: CONNECT => WAITCONNECT handle 0x8001f418; line 1076 (connection #0)
* Connected to 10.1.1.10 (10.1.1.10) port 80 (#0)
* STATE: WAITCONNECT => DO handle 0x8001f418; line 1195 (connection #0)
> GET /app-health-check/status HTTP/1.1
> User-Agent: curl/7.37.0
> Accept: */*
> Host: finance.example.com
>
* STATE: DO => DO DONE handle 0x8001f418; line 1281 (connection #0)
* STATE: DO DONE => WAITPERFORM handle 0x8001f418; line 1407 (connection #0)
* STATE: WAITPERFORM => PERFORM handle 0x8001f418; line 1420 (connection #0)
* HTTP 1.1 or later with persistent connection, pipelining supported
< HTTP/1.1 200 OK
* Server nginx/1.0.15 is not blacklisted
< Server: nginx/1.0.15
< Date: Wed, 10 Dec 2014 04:05:40 GMT
< Content-Type: application/octet-stream
< Content-Length: 90
< Last-Modified: Wed, 10 Dec 2014 03:59:01 GMT
< Connection: keep-alive
< Accept-Ranges: bytes
<
<html>
```
Everything is super groovy at finance.example.com baby!

The client headers are reconstituted into the format that the BIG-IP system expects. In this example:

```
GET /app-health-check/status HTTP/1.1
Host: finance.example.com
Connection: close

```

Testing receive strings (BIG-IP 11.6 - 13.x)

After a receive string is built, one way to validate its format is to send the string directly over TCP to the pool member using `printf` and `netcat (nc)`. For example:

```
$ printf "GET /app-health-check/status HTTP/1.1\r\nHost: finance.example.com\r\nConnection: close\r\n\n" | nc 10.1.1.10 80

HTTP/1.1 200 OK
Server: nginx/1.0.15
Date: Wed, 10 Dec 2014 04:32:53 GMT
Content-Type: application/octet-stream
Content-Length: 90
Last-Modified: Wed, 10 Dec 2014 03:59:01 GMT
Connection: close
Accept-Ranges: bytes

Everything is super groovy at finance.example.com baby!

If you are working with an SSL-enabled website, you can apply the same method with OpenSSL to send a BIG-IP system-formatted monitor string:
$ printf "GET /app-health-check/status HTTP/1.1\r\nHost: finance.example.com\r\nConnection: close\r\n" | openssl s_client -connect finance.example.com:443 -quiet

depth=0 C = US, ST = WA, L = Seattle, O = MyCompany, OU = IT, CN = localhost.localdomain, emailAddress = root@localhost.localdomain

verify error:num=18:self signed certificate
verify return:1

depth=0 C = US, ST = WA, L = Seattle, O = MyCompany, OU = IT, CN = localhost.localdomain, emailAddress = root@localhost.localdomain

verify return:1

HTTP/1.1 200 OK
Server: nginx/1.0.15
Date: Wed, 10 Dec 2014 17:53:23 GMT
Content-Type: application/octet-stream
Content-Length: 90
Last-Modified: Wed, 10 Dec 2014 03:59:01 GMT
Connection: close
Accept-Ranges: bytes

<html>
<body>
Everything is super groovy at finance.example.com baby!
</body>
</html>

read:errno=0

You can use a packet capture tool such as Wireshark to capture working requests and to view headers and responses.

Testing supported monitors (BIG-IP 13.1.0 and later)

Beginning in BIG-IP 13.1.0 you can test supported monitors to ensure that they are working properly before you assign them to network resources. It is not possible to test a monitor after you assign it to a resource.

All but the following monitors are supported:
• inband
• module-score
• radius-accounting
• real-server
• sasp
• snmp-dca-base
• snmp-dca
• virtual-location
• wmi

Note: If you attempt to run a monitor test on an assigned monitor, the system displays a message similar to the following example:

(tmos)# run ltm monitor http test_http destination 10.1.20.11:80
01020037:3: The requested monitor instance (/Common/test_http 10.1.20.11 80 ltm-pool-member) already exists.
(tmos)#

Additionally, the audit log shows a message similar to the following example:

pid=30432 user=root folder=/Common module=(tmos)# status=[Command OK] cmd_data=run ltm monitor icmp icmp destination 10.1.20.11:

Results of monitor tests are visible in the Configuration utility by navigating to Local Traffic > Monitors and clicking the Test tab.

You can type the following tmsh commands to run and stop monitor tests and show the results:

run ltm monitor <monitor type> <monitor name> destination <addr>[<port>]
stop ltm monitor <monitor type> <monitor name>
show ltm monitor <monitor type> <monitor name> test-result

Note: The monitor fires at a configured interval, and the show ltm monitor command shows the current results of the monitor as it occurs in the master control program daemon (mcpd).

You can type the following iControl REST commands to run a monitor test and view the results of a monitor test:

‘{"command":"run","utilCmdArgs":"<monitor type> /Common/<monitor name> destination <addr>[<port>]"}’

GET /mgmt/tm/ltm/monitor/<monitor type>/<monitor name>/stats?options=test-result
The system logs monitor results to the /var/log/monitors/<monitor name> file, which contains extended information about what the monitor is doing.

**Note:** Test results do not trigger any system action. For instance, if a test result shows a resource as **UP** or **DOWN**, the system does not mark the intended resource accordingly.

**Upstream monitoring and alerting**

If you have configured your BIG-IP system to send SNMP traps to an upstream monitoring system, you should be able to configure the monitoring system to send email alerts to application owners when pool member state changes occur. Doing so allows BIG-IP you to delegate monitoring functions to application owners.

**Troubleshoot load balancing problems**

Load balancing issues typically fall into one of the following categories:

- **Imbalanced load balancing**, in which connections are not distributed as expected across all pool members.
- **No load balancing**, in which all connections go to one pool member.
- **Traffic failure**, in which no connections go to any pool member.

**Imbalanced load balancing**

Imbalanced load balancing occurs when a larger-than-expected proportion of connections go to a subset of pool members. Depending on the load balancing method chosen, some imbalance can be expected, as with the Ratio load balancing method.

When diagnosing imbalanced load balancing, consider the following conditions:

- Persistence.
- Server conditions when using dynamic load balancing methods.
- Server conditions when using node-based rather than member-based load balancing methods.
- iRules affecting/overriding load balancing decisions.
- Monitor flapping (pool members marked down and then up repeatedly within a short period of time).

**No load balancing**

No load balancing occurs when all connections are directed to one pool member.

When diagnosing imbalanced load balancing, consider the following conditions:

- Monitors marking pool members down and staying down.
- Fallback host defined in HTTP profile.
Persistence.

Server conditions when using dynamic load balancing methods.

Server conditions when using node-based rather than member-based load balancing methods.

iRules affecting/overriding load balancing decisions.

Traffic failure

Traffic failure occurs where the BIG-IP system receives initial traffic from the client but cannot direct the traffic to a pool member, or when connections to a pool member fail.

When diagnosing traffic failure, consider the following conditions:

- Improper status (for example, monitor marking down when it should be up or vice versa).
- Inappropriate use of profiles (for example, not using a server-ssl profile with an SSL server).
- Server conditions.
- iRules affecting/overriding load balancing decisions.
- iRules errors.
- Insufficient source-address-translation addresses available.

References

For more information on troubleshooting load balancing issues, refer to the following documents:

- K12531: Troubleshooting health monitors
- K3224: HTTP health checks may fail even though the node is responding correctly
- K13898: Determining which monitor triggered a change in the availability of a node or pool member (11.x)
- K10430: Causes of uneven traffic distribution across BIG-IP pool members
- K7820: Overview of SNAT features Contains information about insufficient source-address-translation issues in the SNAT port exhaustion section.
- The iRules chapter of this guide and DevCentral have information about troubleshooting iRules. K7820: Overview of SNAT features contains information about insufficient source-address-translation issues in the SNAT port exhaustion section.
BIG-IP LTM network address objects

This chapter covers the various network-address object types and how they are handled by BIG-IP® LTM®.

When an environment uses both IPv4 and IPv6 addressing, the BIG-IP system can also act as a protocol gateway, allowing clients that have only one IP stack to communicate with hosts that use the alternate addressing scheme.

Additional features that help support a disparate IPv4/IPv6 environment not covered by this guide include NAT IPv6 to IPv4 (NAT6to4), and DNS6to4.

Virtual address

A virtual address is a BIG-IP LTM object limited in scope to an IP address or IP netblock level. It is a mechanism by which BIG-IP LTM virtual server objects are assigned to traffic groups.

You can use a virtual address to control how the BIG-IP system responds to IP-level traffic destined for a TMM-managed IP address that is not otherwise matched by a specific virtual server object.

For example, you can configure the BIG-IP system so it does not send Internet control message protocol (ICMP) echo-replies when all of the virtual servers associated with a virtual address are unavailable.

The BIG-IP system creates virtual addresses automatically when it creates a virtual server. Virtual addresses can also be created automatically using the TMOS® Shell (tmsh), but they are generally only useful in conjunction with an associated virtual server. For more information, refer to “Traffic group interaction”.

Arp, ICMP echo, and route advertisement are behaviors are controlled in the virtual address configuration.

Arp

The ARP property of the virtual address controls whether or not the BIG-IP system responds to arp- and IPv6-neighbor discovery requests for the virtual address. Arp is disabled by default for network virtual addresses.

**Note** Disabling a virtual server doesn’t cause the BIG-IP system to stop responding to arp requests for the virtual address.

ICMP echo

In BIG-IP 11.3 and later, the ICMP echo property of the virtual address controls how the BIG-IP system responds to ICMP echo (ping) requests sent to the virtual address. You can enable or disable ICMP echo responses. In BIG-IP 11.5.1 and later, you can selectively enable ICMP echo responses based on the state of the virtual servers associated with the virtual address.

For more information refer to AskF5™ article: K16885: Controlling the BIG-IP ARP and ICMP echo response behavior for a virtual address.

Route advertisement

When you use Route Health Injection (RHI) with dynamic routing protocols, the virtual address object controls the
addition of routing entries to the routing information base.

For more information on route health injection, refer to Configuring route advertisement on virtual addresses in BIG-IP TMOS: IP Routing Administration.

For more information about virtual addresses, refer to About Virtual Addresses in BIG-IP Local Traffic Management: Basics.

Note For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

Address translation

Network address translation

The BIG-IP system uses network address translation (NAT) to map one IP address to another, typically between public and private IP addresses. NAT allows bidirectional traffic through the BIG-IP system.

NAT contains an origin address and a translation address. Connections initiate from the origin address and pass through the BIG-IP system, which uses the translation address as the source on the server side. Connections initiated to the translation address pass through the BIG-IP system to the origin address.

In BIG-IP 11.3 and later, the source and destination address configuration in virtual server configurations can add flexibility.

Secure network address translation

The BIG-IP system uses secure network address translation (SNAT) to map a set of source addresses on the client side of the system to an alternate set of source addresses on the server side. You can’t initiate a connection through SNAT on the BIG-IP system.
In BIG-IP 11.2.x and earlier, the system uses SNAT objects to provide unidirectional access through the BIG-IP system. In BIG-IP 11.3 and later, you can configure the source and destination address of a virtual server configured to achieve the same access with more flexibility.

For more information on SNATs, refer to NATS and SNATs in BIG-IP TMOS: Routing Administration.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

**SNAT pool**

A SNAT pool is a group of IP addresses from which the BIG-IP system chooses a translation address. Each address in a SNAT pool also automatically creates a SNAT translation if one does not already exist.

**SNAT translation**

A SNAT translation sets the properties of an address the system uses as a translation or in a SNAT pool.

You may need to change some properties to use SNAT translation. These include setting the traffic-group designation and setting whether or not the BIG-IP system responds to address resolution protocol (arp) requests for the translation address.

**Ephemeral port exhaustion**

For protocols such as UDP and TCP, network stacks keep track of connections by the source and destination address, protocol, and source and destination protocol ports. When using a SNAT, source address becomes limited to the addresses used for the SNAT. This is okay when actual traffic uses a variety of ports and destination addresses, but if the number of destination addresses is small, the BIG-IP system may not be able to establish a
connection due to ephemeral port exhaustion. Typically this problem can be alleviated by using a SNAT pool or adding more addresses to a SNAT pool. For more information, refer to AskF5 article: K8246: How BIG-IP system handles SNAT port exhaustion.

For more information on NATs and SNATs, refer to NATs and SNATs in BIG-IP TMOS: Routing Administration.

Note For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

Self IP address

Self IP addresses are configured on the BIG-IP system and associated with a VLAN. These addresses define which networks are locally attached to BIG-IP.

Depending on the port lockdown settings for the self IP, these addresses can be used to communicate with the BIG-IP system. However, F5 recommends that you perform management of BIG-IP system through the management interface on a protected management network using SSH and HTTPS, not through the self IPs.

Self IP addresses are also used by the BIG-IP system when it needs to initiate communications with other hosts using routes associated with the self IP networks. For example, monitor probes are initiated from a non-floating self IP address.

In high-availability environments, configuring floating self IP addresses for each traffic group on each VLAN eases configuration going forward.

- Any virtual servers using an auto-map source address translation setting works properly in a fail-over event if mirroring is configured.
- If routes need to be configured on adjacent hosts, they can be configured to utilize the floating address to ensure that the routed traffic passes through the active BIG-IP system.

IPv4/IPv6 gateway behavior

Since the BIG-IP system is a full proxy and can communicate with both IPv4 and IPv6, the BIG-IP system can be used as a gateway between the two protocols. For example, it is possible to create an IPv6 virtual server to service IPv6 clients that uses an IPv4 pool.

When the BIG-IP system converts from one address type to another, it needs to map the client address to the same type as the pool members. In order to do this, the BIG-IP system automatically uses an automap source address translation for the connection. As a result, it chooses a self IP address of the same type as the pool member. If these connections are mirrored, ensure that there is a floating self IP address of the appropriate type configured. It is possible to override this automatic behavior by manually specifying a source address translation method on the virtual server.

Auto Last Hop

When the Auto Last Hop feature is enabled (default), the BIG-IP system sends return traffic for the client-side connection back to the source MAC address associated with the ingress traffic using the router it transited on ingress. Auto Last Hop behaves this way even if the routing table points to a different gateway IP address or
interface. When Auto Last Hop is disabled, response traffic can traverse a different return path when load balancing transparent devices. This may result in asymmetric routing.

Auto Last Hop may be disabled globally or at the virtual server, SNAT, NAT, and VLAN object levels. However, most BIG-IP LTM implementations function best with Auto Last Hop enabled. For more information about Auto Last Hop, refer to AskF5 article: K13876: Overview of the Auto Last Hop setting.

Take care when using Auto Last Hop in conjunction with neighboring routers that implement hop redundancy protocols (FHRP) such as HSRP or VRRP, as failover issues may occur. For more information, refer to AskF5 article: K9487: BIG-IP support for neighboring VRRP/HSRP routers.

Networks using Cisco Nexus switches introduce additional challenges; however, corrective adjustments can be made on the Nexus switches to alleviate these issues. See AskF5 article: K12440: Neighboring devices using both Virtual PortChannels and HSRP may discard return frames from the BIG-IP system and the relevant Cisco documentation.

Service Provider protocols

FTPS support for passthrough mode

An extension to the file transfer protocol (FTP), FTPS supports TLS and SSL cryptographic protocols.

Typically, personal devices (such as smart phones) use FTPS and operating systems which support FTP but lack an SSH/SFTP client. No Carrier Grade NAT (CGNAT) license is required for FTPS, since the feature is equally applicable to CGNAT, SNAT, and BIG-IP AFM.

In BIG-IP 12.0 and later, the FTP Application-Level Gateway (ALG) Profile handles secure FTPS traffic and allows passthrough.

**Note** FTPS negates the functionality of a Protocol Security Module (PSM) FTP Profile.

For information on using FTP-ALG profiles, refer to Using ALG Profiles in BIG-IP CGNAT: Implementations.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

SCTP support for multi-streaming and multi-homing

Stream Control Transmission Protocol (SCTP) carries call control signals using IP networks. It improves on TCP and UDP by integrating components of each, and SCTP provides benefits to connection-oriented communications.

In BIG-IP 12.0 and later, SCTP supports multi-streaming and multi-homing. Multi-streaming enables passing of several independent streams of data chunks in parallel. Multi-homing provides redundant paths to increase reliability.

Client-side implications

Because of the VIP listener design, the BIG-IP system cannot accept initial SCTP association setup requests (INIT
chunks) sent by the client to an alternative address. A conventional VIP has no such listener to accept them. When
the path the BIG-IP primary address becomes unavailable, a client can’t setup an association to the BIG-IP
system using one of its secondary SCTP addresses. According to the SCTP socket API defined in RFC 6458,
SCTP can listen and connect to multiple addresses, but RFC 4960 says this is unnecessary, so the client-side
impact should be minimal.

Server-side implications

A node object represents a single address, not multiple addresses, and an SCTP association can only be initiated
from that single address. Because of this, the BIG-IP system’s ability to handle server failure modes is impaired.
Any other IP addresses owned by the same server must be discovered once a load balancing choice has been
made and a connection established.

If the primary address of the server is unreachable, the initial SCTP association setup irretrievably fails. There is
no second chance to connect using another IP addresses specified in the node object.

Migration from MBLB to MRF

In BIG-IP 11.5 and later, Service Provider protocol support began the process of migration from Message-Based
Load Balancing (MBLB) to Message Routing Framework (MRF). MRF functionality for SIP load balancing is
supported in BIG-IP 11.6 and later, and for Diameter load balancing in BIG-IP 12.0 and later.

BIG-IP supports client-server protocols and is optimized to manage large numbers of connections, where each
connection provides a communication between one client and one server. In such an environment, the client
always initiates requests and the server always provides responses. A single load balance sequence selects the
server from the virtual server’s pool and messages flow between client and server for the life of the connection.
When the transaction completes, the connection closes and a new connection can be opened for the next
transaction.

Many of the control plane protocols for Service Provider networks differ from this design. For example, Diameter
and SIP can have devices that switch between the client and server roles and create long-lived connections used
as L7 tunnels. A single connection can pass messages intended for different devices on behalf of different users.

MBLB functionality

MBLB enables a BIG-IP system to load balance traffic for message-based protocols. An MBLB proxy supports
message-based protocols by making a load-balancing decision for each message. Once the system forwards the
request to the selected server, the server-side connection is removed from the client-side connection to allow the
next request to be directed to other servers, if required. MBLB proxy maintains an association table, which stores
the relationship between the client-side connection and all of the server-side connections. When the system
receives a server response, the system uses the association table to locate the originator of the request and
forward the response to that client.

MRF functionality

MRF provides a protocol-independent L7 messaging routing framework which operates independently of the
underlying connection-oriented full proxy. Each protocol implementation of MRF provides a protocol-specific
route table, identifying the best route for a message, determined by the message attributes. Associations between client-side and server-side connections are no longer required.

Any connection can originate a request or a response, and any connection can provide the destination for a request or a response. MRF maintains a table of all connections to each peer. Based on a configuration, a new connection can be created or a previous connection reused when forwarding a message to a peer (even when the connection exists on a different TMM).

MRF functionality provides significant benefits compared to MBLB functionality.

Standards support

MRF provides full support of message-based protocols in the Information Management Services (IMS) infrastructure. For example, the BIG-IP system supports the following protocols.

- **Diameter.** MRF Diameter routes all messages through a single connection between the BIG-IP system and its peer. It responds to a capabilities exchange as a Diameter node with its own configurable identity and supported applications.

- **SIP.** MRF SIP identifies and fails many of the illegal messages defined in the SIP torture test that is described in RFC4475. It implements loop and maximum forwarding detection functionality.

  **Tip** MRF performs operations for many use cases without requiring iRules.

Reuse of existing connections

- MRF functionality maintains a table of all open connections to peers. If a message is routed to a peer, it can send the message to that peer through an existing connection.

- If a connection mode is per-peer, then all TMMs use the same connection. This provides support for protocols like Diameter that specify that a single connection must exist between two peers.

The router profile that is attached to the virtual server specifies the router instance to which it belongs. The router instance owns the table of open connections. All virtual servers that share the same routing instance can share the same connections. The connection table holds all open connections, including client-side (peer-initiated) and server-side (system-initiated) connections.

Routes

All protocols which implement MRF allow you to add protocol specific static routes to the router instance to control how messages are forwarded. The protocol implementation specifies the attributes to match against the message, determining the best route for forwarding the message.

A route can contain a list of peers. A peer contains both a pool and a transport configuration, which allows a route to contain a list of servers that can require different connection parameters. A peer-selection mode specifies the mode for selecting peers from the peer list: sequential or ratio.

- **Sequential peer-selection mode.** The first peer is selected, unless all of its pool members are marked as down.
• **Ratio peer-selection mode.** The ratio field for each peer is used to calculate the relative probability of that peer’s selection.

Each peer can contain a pool and a transport configuration, which allows each peer in a peer list to use different settings to create an outgoing connection. The peer also specifies the connection mode and number of connections, which determine how connections to the peer are reused.

If no pool is specified, the local IP address and port of the originating connection for the message being routed are used as the remote IP and port of the outgoing connection.

If no transport configuration is specified, the parameters of the incoming connection are used to create the outgoing connection. A transport configuration specifies the parameters for an outgoing connection, such as the profiles used, source-address translation, and iRules scripts.

**iRules variables**

Because the incoming connection and outgoing connection of a message are never associated, and can exist on different TMMs, MRF provides a new method for sharing iRules variables. MRF provides new iRules commands to deliver iRules variables with the message to the outgoing connection.

**Automatic response routing**

MRF protocol implementations can route response messages back to the originating connection of the request. Compare per blade with per peer, per tmm and per-client routing. **Per-client** is a special case mode in which the system creates a different egress connection to each peer from each client trying to connect. Peering the ingress and egress connection in this manner ensures that the response is always returned to the connection which originally sent the message.

**Rule routing**

MRF has iRules commands to route a message to a specific connection or peer.

**Persistence**

MRF records both endpoints of a session, which permits persistence for subsequent messages in the session, routing them through existing connections or new connections, as required.

The configuration of the persistence record’s key is protocol specific, and the setting is not a function of the protocol profile. Each virtual server contains a session profile. Because multiple virtual servers can share the same router instance, and each virtual server can be configured with a different protocol profile, the actual keys used for each session can be different, depending on the origination of the message.

**MRF SIP load balancing**

The SIP operation mode, specifically load balancing, comprises the MRF SIP use case. You can configure an operation mode in the SIP Router profile. All virtual servers that share the same router profile instance use the same operation mode.
When configured for load balancing, the BIG-IP system processes and delivers SIP control messages to a SIP endpoint. Note that the BIG-IP system does not manage SIP media flows in a load balancing configuration. A configurable persistence table stores routing and load balancing decisions. No additional state is maintained. SIP media flows travel by means of a different path, or are managed by components other than the SIP control messages.

In the default configuration, the BIG-IP system inserts a **Via** header into the message used by the recipient, for routing the response message. This **Via** header includes an encrypted branch parameter, containing the details of the originating connection, and the persistence key. When the BIG-IP system receives a response message, it removes the topmost **Via** header, decrypts its branch attribute, and uses that attribute to set the next hop of the message to the originating flow. This process allows the response message to avoid routing and instead get forwarded directly to the originator of the request. If the originating connection has closed, SIP functionality uses the next topmost through the attribute from the message, and routes the message to the provided IP address and port. No other attributes of the SIP header and Session Description Protocol (SDP) payload are modified in a SIP load balancing operation mode.

You can use a SIP load balancing operation mode for multiple use cases. A BIG-IP system can combine multiple use cases, as necessary.

**MRF SIP load balancing configuration**

SIP load balancing allows a group (or groups) of SIP servers to act as a single SIP device. The BIG-IP system configures the SIP endpoint with the address of the virtual server representing the group of servers. Because SIP endpoints route media between them, the SIP servers do not use media flows. The BIG-IP system can also configure persistence to deliver subsequent messages to the same proxy server.

**MRF SIP forwarding configuration**

A SIP forwarding configuration receives a message on one VLAN and forwards it to another VLAN. This configuration is often used with SIP load balancing to allow the proxy server to invite a SIP endpoint to a call that exists on an internal network. In the forwarding use case, the originating device knows the address of the destination device and the BIG-IP system’s only role is to forward the message to the destination.

The BIG-IP system does not manage media in a SIP Load Balancing operation mode. If media is required to pass through the BIG-IP system, a forwarding generic UDP forwarding virtual is required.

**MRF SIP routing configuration**

SIP routing allows the BIG-IP system to deliver messages to different endpoints, based on attributes of the message. The MRF SIP implementation can route a message based on the message’s **request-uri**, **from-url**, **to-uri**, and the originating virtual server. You can use an iRules script to route messages based on other attributes of the saved state.

The BIG-IP system does not manage media in a SIP load balancing operation mode. If media is required to pass through the BIG-IP system, a forwarding generic UDP forwarding virtual is required.
MRF Diameter load balancing configuration

Diameter load balancing provides capacity scaling and high availability for Diameter signaling servers. Load balancing functionality steers Diameter signaling traffic to a pool of servers, based on static Diameter routes, distributing the load across pool members. Diameter Attribute Value Pair (AVP) parameters (SESSION-ID by default) determine load balancing persistence. Last hop information, which the BIG-IP system saves on the egress TMM, defines the response routing, which skips the route lookup for response messages.

MRF Diameter load balancing with persistence

You can use the Diameter Session profile to configure persistence. An Attribute Value Pair (AVP) value, extracted from a message that initiates a new session, keys the persistence records. You can use the Persist AVP setting in the Session Profile or iRules. To disable persistence, in the Persist Type list, select None.

Session profile persistence default configuration

The first SESSION-ID parameter in a Diameter message determines the default persistence. A Diameter message can specify any valid AVP value, but only messages that contain the specified AVP value are routed, as determined by the persistence record.

Nested AVP values can also be specified, using the following format:

```
Outer-Attribute[index]:Nested-Attribute[index].
```

For example, to configure persistence that is based on the first Subscription-Id-Data value that is nested within the second Subscription-Id AVP, type the following:

```
Subscription-Id[1]:Subscription-Id-Data[0] in the Persist AVP field.
```

MRF Diameter routing configuration

A Diameter routing configuration uses static routes to manage traffic among realms. In a MRF static route, you must specify a peer and a destination-realm, that is, the realm of the destination address. In this example, two peers use the same Diameter Session profile and transport configuration. The BIG-IP system routes messages based on the static route's Application-Id and Destination-Realm values. Only messages with attribute values that match the values in the Diameter Route configuration use that static route.

Diameter Configuration Wizard

For those new to Diameter message routing, in BIG-IP 13.0 and later, an iApp LX wizard is available. Distinct from Traffix® Signaling Delivery Controller (SDC), the wizard allows you to implement complex Diameter use cases by automatically provisioning the BIG-IP LTM objects and iRules® required to dynamically route Diameter messages based on a combination of Diameter attribute value pairs (AVPs). Features include the following:

- Client-side and server-side Stream Control Transfer Protocol (SCTP)
- AVP transformation
- Rule-based session binding
• Rule-based routing
• Pre-built 3rd Generation Partnership Project (3GPP) dictionaries

The wizard harnesses the power of iRules and the REST framework for management and orchestration (MANO), supports SCTP, Load balancing, AVP transformation, and session management.

Important F5 strongly recommends using Google Chrome or Mozilla Firefox to configure the Diameter Configuration Wizard. Internet Explorer is not recommended.

For more information, see Using the Diameter Configuration Wizard in BIG-IP Service Provider: Message Routing Administration.

SIP Wizard

For those new to SIP, beginning in BIG-IP 13.1.0, you can use an iAppLX plug-in SIP wizard to configure common SIP load balancing scenarios, to manipulate SIP Headers, and to customize log output, without needing to be familiar with the MRF architecture.

Like the Diameter Wizard, the SIP Wizard relies on iControl REST to provision existing iRules and BIG-IP LTM Message Routing objects.

The wizard does the following:

• Implements basic SIP load balancing use cases.
• Uses DNS to look up the IP address of the SIP Proxy from the Request-URI.
• Supports either UDP or TCP transport protocol, with the same protocol at either end. (Switching between protocols requires UDP retransmission support.

Note: SIP cannot be used with SCTP at this time.

• Manipulates SIP headers (via, Record-Route, Route, and so on). You can use it to manage or extend the list of SIP headers available for transformation.
• Customizes SIP message logging.

Note: SIP MR ALG mode and Transport Layer Security are not supported.

Logging

Detailed logging options can help with troubleshooting and debugging. Additionally, you can use the tool for customer call logging since it supports external call logging.

Restarting Diameter and SIP wizards

If you encounter issues with the Diameter or SIP wizards, you can restart the restnodced process it by typing the bigstart restart restnodced command. After restarting the restnodced process, check the process log at /var/log/restnodced/restnodced.log for messages indicating that the wizard has successfully restarted.
BIG-IP LTM Virtual Servers

Virtual servers as listeners

A listener is a network address space object that matches network traffic for management by Traffic Management Microkernel (TMM). Listeners are most frequently virtual servers and can listen for traffic as broad as a Classless InterDomain Routing (CIDR) netblock, such as 10.1.0.0/16, down to a specific IP address, protocol, and port combination, such as 10.1.1.20:80 (TCP).

There are a number of virtual server types, each with different capabilities, depending on their type and scope. For more information, see BIG-IP Local Traffic Manager: Concepts.

Note: For information about how to locate F5® product guides, refer to AskF5™ article: K12453464: Finding product documentation on AskF5.

Address matching precedence

BIG-IP® LTM® may receive traffic flows that match the configuration of multiple virtual servers. To avoid conflict between these configurations, BIG-IP LTM has an order of precedence for selecting which virtual server the traffic matches. While the matching order can be complex, depending on the installed software version and applied configuration, the overall process matches traffic to virtual servers based on the most specific to least specific criteria.

The following tables show the order of precedence for matching five sample virtual servers with the corresponding connection table in BIG-IP 11.3 or later:

Table 4.1 Virtual servers

<table>
<thead>
<tr>
<th>Virtual server name</th>
<th>Destination</th>
<th>Service port</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>MyVS1</td>
<td>192.168.1.101/32</td>
<td>80</td>
<td>0.0.0.0/0</td>
</tr>
<tr>
<td>MyVS2</td>
<td>192.168.1.101/32</td>
<td>*</td>
<td>192.168.2.0/24</td>
</tr>
<tr>
<td>MyVS3</td>
<td>192.168.1.101/32</td>
<td>*</td>
<td>192.168.2.0/25</td>
</tr>
<tr>
<td>MyVS4</td>
<td>0.0.0.0/0</td>
<td>80</td>
<td>192.168.2.0/24</td>
</tr>
<tr>
<td>MyVS5</td>
<td>192.168.1.0/24</td>
<td>*</td>
<td>0.0.0.0/0</td>
</tr>
</tbody>
</table>
Table 4.2 Connection table for virtual servers

<table>
<thead>
<tr>
<th>Inbound source address</th>
<th>Inbound destination address</th>
<th>Virtual server selected by the BIG-IP system</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.2.1</td>
<td>192.168.1.101:80</td>
<td>MyVS3 is selected because the destination address matches MyVS1, MyVS2, and MyVS3. The source address matches both MyVS2 and MyVS3, but MyVS3 has a subnet mask narrower than MyVS2.</td>
</tr>
<tr>
<td>192.168.2.151</td>
<td>192.168.1.101:80</td>
<td>MyVS2 is selected because the destination address matches MyVS1, MyVS2, and MyVS3. The source address matches only MyVS2.</td>
</tr>
<tr>
<td>192.168.10.1</td>
<td>192.168.1.101:80</td>
<td>MyVS1 is selected because the destination address matches MyVS1, MyVS2, and MyVS3. The source address matches only MyVS1.</td>
</tr>
<tr>
<td>192.168.2.1</td>
<td>192.168.1.200:80</td>
<td>MyVS5 is selected because the destination address matches MyVS5.</td>
</tr>
</tbody>
</table>

For more information on virtual server precedence, refer to the following AskF5 articles:

- K6459: Order of precedence for virtual server matching (9.x - 11.2.1)
- K14800: Order of precedence for virtual server matching (11.3.0 and later)

Translation options

On L4 virtual servers with a default BIG-IP LTM configuration, most network address translation happens automatically. However, if the BIG-IP system is not configured as the default gateway for pool members, or if you cannot otherwise guarantee that return traffic traverses the BIG-IP system, it may be necessary to assign a SNAT automap or a SNAT pool. Doing so ensures that traffic transits the BIG-IP system on egress and that responses are properly returned to the client.

Traffic group interaction

Traffic groups allow active/active distribution of virtual servers and their associated objects across BIG-IP system devices. To assign a virtual server to a particular traffic group, set the associated virtual address object to the appropriate traffic group.

When creating traffic groups, note the following:

- When using automap source IP address translation, assign the appropriate floating self IPs to each traffic group.
- When using a SNAT pool source address translation, the SNAT pool must contain SNAT translations assigned to the same traffic group as the virtual server’s virtual address. This configuration fosters seamless failover since the addresses used for NAT traffic can follow the traffic group to the next available BIG-IP system device.
Clone pools

Clone pools allow the BIG-IP system to remotely clone TMM-managed traffic to third-party intrusion detection systems (IDS) or intrusion prevention systems (IPS) for further examination. Cloning traffic in this way may offer a more flexible and efficient solution than using span ports or network traps.

You can use clone pools to select either client-side or server-side traffic for remote cloning. In the case of SSL accelerated traffic, cloning the server-side traffic allows it to be seen unencrypted by the IDS or IPS, without requiring export of SSL key pairs for external decryption.

Clone pools also allow for load balancing of mirrored traffic to more than one IDS or IPS system.

For more information about clone pools, refer to BIG-IP Local Traffic Manager: Concepts.

Note For information about how to locate F5® product guides, refer to AskF5™ article: K12453464: Finding product documentation on AskF5.

Profile use by virtual servers

Virtual servers use profiles to enhance traffic management by the BIG-IP system, allowing for protocol-level control. For example, a virtual server with an HTTP profile can also use cookie-based persistence profiles and iRules that perform HTTP inspection and modification.

For more information on profiles, refer to “BIG-IP LTM Profiles”.

Common Virtual server types

Standard

A Standard virtual server directs client traffic to a load balancing pool and is the most basic type of virtual server. It is a general purpose virtual server that does everything not expressly provided by the other type of virtual servers.

A Standard virtual server type allows for a full-proxy service—that is, a client-side connection and a server-side connection with data passing between them. F5 recommends Standard virtual servers whenever L7 intelligence is or may required.

F5 recommends using a Standard virtual server type under the following conditions:

• You need to ensure that requests from the same client are routed to the same server using information in the protocol stream.
• You want to inspect, modify, or log data in the protocol stream.
• You want BIG-IP LTM to perform protocol enforcement.
• You want supplemental DDoS protection.

For more information, refer to AskF5 article: K14163: Overview of BIG-IP virtual server types (11.x).
Performance (Layer 4)

A Performance (Layer 4) virtual server type has a FastL4 profile associated with it and uses Packet Velocity ASIC (PVA) to increase packet processing speed.

F5 recommends using a Performance (Layer 4) virtual server type when little or no L4 or L7 processing is required.

While the BIG-IP system can perform source and destination IP address translation as well as port translation, load balancing decisions are limited in scope since minimal L7 information is available.

On F5 hardware platforms that support it, a Performance (Layer 4) virtual server can offload connection flows to the ePVA, which can result in higher throughput and lower latency.

Forwarding (IP)

A Forwarding (IP) virtual server forwards packets directly to the destination IP address specified in the client request. It uses the routing table to make forwarding decisions based on the destination address for the server-side connection flow. Also, it has no pool members to load balance.

You can use a Forwarding (IP) virtual server to forward IP traffic in the same way as any other router. To enable stateless forwarding, you have to set FastL4 profile options on the virtual server.

You can also define specific network destinations and source masks for virtual servers and/or enable them only on certain VLANs. This allows precise control of how network traffic is handled when forwarded.

For example, you can use a wildcard virtual server with Source Address Translation, enabled for outbound traffic, and then add an additional network virtual server with Source Address Translation, disabled for traffic destined for other internal networks. Or you can use a Performance (Layer 4) virtual server to select certain traffic for inspection by a firewall or IDS.

However, if you use a Performance (Layer 4) virtual server type, ensure that Translate address and Translate port options are disabled. These options are automatically disabled when a virtual server is configured with a network destination address.

For more information about forwarding IP virtual servers, refer to AskF5 article: K7595: Overview of IP forwarding virtual servers.

Other virtual server types

Forwarding (Layer 2)

A Forwarding (Layer 2) virtual server typically shares the same IP address as a node in an associated VLAN. It is typically used in conjunction with a VLAN group. Otherwise, it is similar to the Forwarding (IP) virtual server type.

Stateless

A Stateless virtual server type performs minimal packet processing and does not create connection flows. It supports only UDP and is only recommended in specific, limited situations.
Reject

A Reject virtual server type rejects any packets which would create a new connection flow. Use cases for this type of virtual server include blocking out a port or IP address within a range covered by a forwarding or other wildcard virtual server. For more information, refer to AskF5 article K14163: Overview of BIG-IP virtual server types (11.x).

Performance (HTTP)

A Performance (HTTP) virtual server has a FastHTTP profile associated with it. The Performance (HTTP) virtual server and related profile increase the speed at which the virtual server processes HTTP requests.

While it may be the fastest way to pass HTTP traffic under certain circumstances, Performance (HTTP) virtual servers have specific requirements and limitations. Refer to AskF5 article K8024: Overview of the FastHTTP profile before deploying a Performance (HTTP) virtual server type.

Miscellaneous

BIG-IP LTM supports other virtual server types, but these are less common and have specific use cases. As of BIG-IP 12.0, these use server types include DHCP, Internal, and Message Routing. For more information on these virtual server types, refer to AskF5 article K14163: Overview of BIG-IP virtual server types (11.x).

Virtual server considerations

Automatically created listeners

The BIG-IP system automatically creates and manages some listeners for certain types of traffic when the listeners run in conjunction with relevant service profiles.

For example, when you use active FTP on a virtual server with an FTP profile assigned, the BIG-IP system dynamically allocates FTP data port listeners to map outbound connections from pool members to the inbound virtual server listener address. This allows the client to see the connection originating from the IP address it expects.

Listeners can also be created manually with the listen iRules® command.

SNAT and NAT objects also automatically create listeners to allow the BIG-IP system to pass traffic involving one or more hosts that require address translation.

Virtual server connection behavior for TCP

Each type of virtual server type has a unique way of handling TCP connections. For more information about TCP connection behavior for virtual server types, refer to AskF5 article K8082: Overview of TCP connection setup for BIG-IP LTM virtual server types.

Standard vs Performance (Layer 4)

While Standard and Performance (Layer 4) virtual servers appear to perform similar functions, there are
differences in capabilities between the types. The following table shows use case criteria for each.

<table>
<thead>
<tr>
<th>Use Standard type when</th>
<th>Use Performance (Layer 4) type when</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Using other BIG-IP modules (BIG-IP ASM®, BIG-IP APM®, BIG-IP AAM®, etc.)</td>
<td>• No L7 interaction is needed</td>
</tr>
<tr>
<td>• Using SSL profiles</td>
<td>• Not using or passing SSL through BIG-IP LTM</td>
</tr>
<tr>
<td>• Using any L7 profile</td>
<td>• Making a policy-based forwarding decision with L4 information only</td>
</tr>
<tr>
<td>• Using layer L7 iRule events</td>
<td>• Using only L1-4 iRule events</td>
</tr>
<tr>
<td>• Using layer L7 persistence methods</td>
<td></td>
</tr>
</tbody>
</table>

For more information on selecting the appropriate virtual server for HTTP traffic, refer to AskF5 article: K4707: Choosing appropriate profiles for HTTP traffic.

**Forwarding (IP) vs Performance (Layer 4)**

Both Forwarding (IP) and Performance (Layer 4) virtual servers use FastL4 functionality, and both can use a custom FastL4 profile. However, Performance (Layer 4) virtual servers allow for a pool to be configured and to act as a gateway pool. This is commonly used to load balance firewalls. Forwarding (IP) virtual servers do not have pools and use the routing table to direct server-side connections.

**Source port preservation**

When mapping a connection to a pool member, the BIG-IP system typically doesn't need to translate the ephemeral port used by the client, but exceptions exist.

The default setting for Source Port option is Preserve. This setting causes the BIG-IP system to retain the source port of the client whenever possible. However, on configurations where virtual server port and pool member port are different, or when the source port is already in use, the BIG-IP system may modify the client source port to ensure return traffic is sent to the correct TMM CMP instance. The behavior is common in instances with SSL offload because the virtual server has a service port of 443 but the pool member typically has a service port of 80. This behavior does not affect application operation, but it may be confusing when troubleshooting.

For more information on source port preservation, refer to the following AskF5 article:

- K11004: Port Exhaustion on CMP systems.
Virtual server troubleshooting

Capture network traffic with tcpdump

For information about capturing network traffic with tcpdump, refer to the following AskF5 articles:

- K411: Overview of packet tracing with the tcpdump utility
- K13637: Capturing internal TMM information with tcpdump

OSI L1-3

Standard virtual server types operate at L4 on the OSI model. When troubleshooting virtual servers or other listeners, consider L1–3 (physical, data-link, and network) and how they contribute to a healthy connection.

Physical layer

The physical layer is the physical connection between BIG-IP LTM and other devices, primarily Ethernet cables and fiber optic cables. It includes low-level link negotiation and can be verified by observing interface state.

Data-link layer

The data-link layer is primarily Ethernet and Link Aggregation Control Protocol (LACP). You can validate that the data-link layer is functioning by observing LACP status of trunks or by capturing network traffic to ensure Ethernet frames are arriving at BIG-IP LTM.

Network layer

The network layer is IP, arp, and ICMP. Arp (or IPv6 neighbor discovery) is a prerequisite for IP to function.

L3 troubleshooting

Your troubleshooting process should include layer 3 investigation.

Confirm arp resolution by checking the arp table on the BIG-IP system and other IP-aware devices on the local network. If arp resolution fails, a lower-layer interruption or misconfiguration may exist. If arp resolution succeeds, confirm that packets from the client are reaching BIG-IP LTM. If they don’t, a firewall may be blocking packets or routing may not be configured well enough.

Use ping, traceroute, or similar tools to verify IP connectivity between the BIG-IP system and the client and dependent resources, such as pool members.

Use tcpdump to troubleshoot

You can use the tcpdump utility to capture network traffic for troubleshooting purposes. tcpdump allows you to view live packet data at the command line or to write captured traffic to standard packet capture (PCAP) files which can be examined with Wireshark or other tools at a later time.
Commonly used options for tcpdump:

- `i interface`

This option indicates from which interface to capture data. It is required. The specified interface can be one of the following options:

- A specific VLAN
- `0.0` (all VLANs)
- `eth0` (mgmt interface traffic)
- A specific front-panel interface (1.1, 1.2, and so on)

For VLANs and `0.0`, you can specify the `:p` flag on the interface to capture traffic for the peer flow to any that match the filter as well.

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-s0</code></td>
<td>Configures <strong>tcpdump</strong> to capture the complete packet. (Recommended when troubleshooting.)</td>
</tr>
<tr>
<td><code>-w /var/tmp/filename.pcap</code></td>
<td>Configures <strong>tcpdump</strong> to write captured packet data to a standard PCAP file which you can inspect using Wireshark or <strong>tcpdump</strong>. When you specify the <code>-w</code> flag, <strong>tcpdump</strong> does not print packet information to the terminal.</td>
</tr>
<tr>
<td><code>-nn</code></td>
<td>Disables DNS address lookup and shows ports numerically instead of using a name from <code>/etc/services</code>. Only useful when packets are not being written to a file. Recommended.</td>
</tr>
</tbody>
</table>

**Note** F5 recommends writing packet captures to `/var/tmp` to avoid service interruptions when working with large captures.

tcpdump packet filters

When troubleshooting, isolate the desired traffic by specifying packet filters whenever possible, particularly on systems processing large volumes of traffic.

The command syntax in the following examples can be used to capture traffic with tcpdump.

Sample data used in examples:

- Client IP: 192.168.1.1
- Virtual server: 192.168.2.1:80
- Client-side vlan: VLAN1
Server-side VLAN (two-armed topology): **VLAN2**

Pool member: 192.168.2.2:80 for one-armed examples.

Add 192.168.3.2:80 for two-armed examples.

Non-floating self IP addresses: 192.168.2.3 and 192.168.3.3

Floating self IP addresses: 192.168.2.4 and 192.168.3.4

To capture arp traffic only on the client-side VLAN and log to the terminal

- Type the following command syntax:
  
tcpdump -i VLAN1 -nn -s0 arp

To capture all traffic from a particular client to /var/tmp/example.pcap

- Type the following command syntax:
  
tcpdump -i 0.0 -s0 -w /var/tmp/example.pcap host 192.168.1.1

  If Source Address Translation is enabled, the above command does not capture server-side traffic. To do so, use the :p flag (a client-side flow’s peer is its respective server-side flow):
  
tcpdump -i 0.0:p -s0 -w /var/tmp/example.pcap host 192.168.1.1

To capture client-side traffic only from a sample client to the desired virtual server

- Type the following command syntax:
  
tcpdump -i VLAN1 -s0 -w /var/tmp/example.pcap host 192.168.1.1 and host 192.168.2.1 and tcp port 80

To capture traffic only to a pool member in a one-armed configuration (excluding non-floating self IP to exclude monitor traffic)

- Type the following command syntax:
  
tcpdump -i VLAN1 -s0 -w /var/tmp/example.pcap host 192.168.2.2 and tcp port 80 and not host 192.168.2.3

To capture client-side traffic as well, in a two-armed configuration, use this modified version of the last command

- Type the following command syntax:
  
tcpdump -i 0.0:p -s0 -w /var/tmp/example.pcap host 192.168.3.2 and tcp port 80 and not host 192.168.3.3

You can find more information for using **tcpdump** filters by searching [AskF5](support.f5.com).
Using tcpdump remotely

Beginning in BIG-IP 13.1.0, you can run `tcpdump` to collect information on another system, which may be valuable for on-site troubleshooting. BIG-IP 13.1.0 and later supports piping packets over a Generic Routing Encapsulation (GRE) tunnel to perform remote system analysis (typically a workstation running Wireshark to capture and analyze packets). This adds the flexibility to not require the host system to manage resource capturing packets.

When you run `tcpdump` to collect information on another system, the utility coordinates with `mcpd` and `tmm` processes to open a GRE tunnel towards a specified address and then forwards captured packets over this connection in either raw or in a specified encapsulation type. After `tcpdump` stops, tunnel flow is torn down and the resources are released.

**Note:** The remote system on which you collect information must be on the local network. Remote support does not include different geographical locations or across the Internet. If you use `tcpdump`, it must be supported by your environment and your environment must support GRE tunneling protocol.

**Important:** By using `tcpdump` to collect information on a remote system, you can easily expose unencrypted data over insecure links.
BIG-IP LTM Profiles

Profiles allow the BIG-IP® LTM® system to understand or interpret supported network traffic types and affect the behavior of managed traffic flowing through it.

BIG-IP systems include a set of standard (default) profiles for a wide array of applications and use cases, such as performing protocol enforcement of traffic, enabling connection and session persistence, and implementing client authentication.

The following table lists common default BIG-IP LTM profiles:

<table>
<thead>
<tr>
<th>Profile type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>L7 protocols, such as HTTP, FTP, DNS, and SMTP.</td>
</tr>
<tr>
<td>Persistence</td>
<td>Settings govern routing of existing or related connections or requests.</td>
</tr>
<tr>
<td>Protocol</td>
<td>L4 protocols, such as TCP, UDP, and Fast L4.</td>
</tr>
<tr>
<td>SSL</td>
<td>Enable the interception, offloading, and authentication of transport layer encryption.</td>
</tr>
<tr>
<td>Other</td>
<td>Provide various types of advanced functionality.</td>
</tr>
</tbody>
</table>

For detailed information about BIG-IP protocol profiles, see Protocol Profiles in BIG-IP Local Traffic Management: Profiles Reference.

Note For information about how to locate F5® product guides, refer to AskF5™ article: K12453464: Finding product documentation on AskF5.

Profile management

F5 recommends the following guidelines for working with default profiles:

- Changes to the default profile settings are lost when the BIG-IP system software is upgraded. Make sure that data is backed up before upgrading.

- Create a child profile of the default profile and make custom updates to the child rather than altering the default profile itself. Changes to default profiles may have unexpected consequences for custom profiles that inherit default settings.

Tip A custom profile can inherit settings from another custom profile. This allows for efficient management of profiles for related applications requiring similar-but-unique configuration options.

Protocol profiles

There are a number of profiles available that are jointly described as protocol profiles. For more information on the following profiles, refer to Protocol Profiles in BIG-IP Local Traffic Manager: Profiles Reference, along with the referenced solutions.
Fast L4

FastL4 profiles are used for Performance (Layer 4), Forwarding (Layer 2), and Forwarding (IP) virtual servers. The FastL4 profile can also be used to enable stateless IP level forwarding of network traffic in the same fashion as an IP router. For more information, refer to AskF5™ article: K7595: Overview of IP forwarding virtual servers.

Fast HTTP

The FastHTTP profile is a scaled-down version of the HTTP profile optimized for speed under controlled traffic conditions. It can only be used with the Performance HTTP virtual server and is designed to speed up certain types of HTTP connections and reduce the number of connections to servers.

Because the FastHTTP profile is optimized for performance under ideal traffic conditions, the HTTP profile is recommended when load balancing most general-purpose web applications.

Refer to AskF5 article: K8024: Overview of the FastHTTP profile before deploying performance (HTTP) virtual servers.

TCP

TCP profiles are configuration tools that help you to manage TCP network traffic. Many of the configuration settings of TCP profiles are standard SYSCTCL types of settings, while others are unique to the BIG-IP® system.

TCP profiles are important because they are required for implementing certain types of other profiles. For example, by implementing TCP, HTTP, Rewrite, HTML, and OneConnect™ profiles, along with a persistence profile, you can take advantage of various traffic management features, such as:

- Content spooling, to reduce server load
- OneConnect™, to pool idle server-side connections
- Layer 7 session persistence, such as hash or cookie persistence
- iRules® for managing HTTP traffic
- HTTP data compression
- HTTP pipelining
- URI translation
- HTML content modification
- Rewriting of HTTP redirections

The BIG-IP system includes several pre-configured TCP profiles that you can use as is. In addition to the default tcp profile, the system includes TCP profiles that are pre-configured to optimize LAN and WAN traffic, as well as traffic for mobile users. You can use the pre-configured profiles as is, or you can create a custom profile based on
a pre-configured profile and then adjust the values of the settings in the profiles to best suit your particular network environment.

TCP base profiles

In BIG-IP 13.0 and later, several new TCP base profiles are available in addition to the existing profiles, which have been updated and are now read-only.

<table>
<thead>
<tr>
<th>TCP profile</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp</td>
<td>Existing. Provides “safe” new features. Serves as parent to all new F5 base profiles.</td>
</tr>
<tr>
<td>tcp-legacy</td>
<td>Existing. Derived from tcp profile. Serves as parent to all previously existing base profiles.</td>
</tr>
<tr>
<td>f5-tcp-progressive</td>
<td>New. Most performance features enabled.</td>
</tr>
<tr>
<td>f5-tcp-lan</td>
<td>New. Loosely based on tcp-lan-optimized, which is deprecated in BIG-IP 13.0 and later.</td>
</tr>
<tr>
<td>f5-tcp-wan</td>
<td>New. Loosely based on tcp-wan-optimized, which is deprecated in BIG-IP 13.0 and later.</td>
</tr>
<tr>
<td>f5-tcp-mobile</td>
<td>New. Loosely based on tcp-mobile-optimized, which is deprecated in BIG-IP 13.0 and later.</td>
</tr>
</tbody>
</table>

TCP autotune

In BIG-IP 13.0 and later, the TCP profile includes an autotune feature which allows you to autotune TCP send and receive buffers as well as the proxy buffer used to interface with the rest of the connection flow.

For general information on the TCP profile, refer to AskF5 article: [K7759: Overview of the TCP profile](#).

UDP

Use of a user datagram protocol (UDP) profile is mandatory on standard UDP virtual servers.

For more detail on the UDP profile refer to AskF5 article: [K7535: Overview of the UDP profile](#).

SCTP

Stream control transmission protocol (SCTP) is a L4 transport protocol, designed for message-oriented applications that transport signaling data, such as Diameter.

OneConnect

The OneConnect profile may improve HTTP performance by reducing connection setup latency between the BIG-IP system and pool members, as well as minimizing the number of open connections to them.

The OneConnect profile maintains a pool of connections to the configured pool members. When there are idle
connections available in the connection pool, new client connections uses these existing pool member connections if the configuration permits. When used in conjunction with an HTTP profile, each client request from an HTTP connection is load-balanced independently.

For information about settings within the OneConnect profile, refer to About OneConnect Profiles in **BIG-IP Local Traffic Management: Profiles Reference**.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

OneConnect use cases

HTTP requests can be load-balanced to different pool members on a request-by-request basis by enabling OneConnect on a virtual server. When using an HTTP profile without a OneConnect profile, the BIG-IP system makes a load balancing decision based on the first request received on the connection, and all subsequent requests on that connection continues to go to the same pool member.

iRules can be used to select pools and pool members based on information contained in each request, such as the URI path or user-agent string.

The OneConnect source mask setting manages connection re-use, and is applied to the client address on the server-side of a connection to determine its eligibility for connection re-use. In this way, it is possible to limit which clients can share server-side connections with each other.

![Figure 5.1: OneConnect source mask connection reuse](image)

Refer to AskF5 article: K5911: Managing connection reuse using OneConnect source mask.
OneConnect limitations

F5® recommends careful consideration of the following limitations and implications before using a OneConnect profile:

- OneConnect is intended to work with traffic that can be inspected by the BIG-IP system at L7.
- OneConnect requires the use of an HTTP profile to process HTTP traffic.
- When using the default OneConnect profile, the pool member cannot rely on the client IP address information in the network headers to accurately represent the source of the request. Consider using the X-Forwarded-For option in an HTTP profile to pass the client IP address to the pool members using an HTTP header.
- Applications that utilize connection-based authentication such as NT Lan Manager (NTLM), may need additional profiles or may not be compatible with OneConnect. For more information, refer to AskF5 article: K10477: Optimize NTLM traffic in BIG-IP 10.x or later or Other Profiles in BIG-IP Local Traffic Management: Profiles Reference.

**Note** For information about how to locate F5® product guides, refer to AskF5™ article: K12453464: Finding product documentation on AskF5.

- Before using OneConnect with non-HTTP applications, refer to AskF5 article: K7208: Overview of the OneConnect profile.

HTTP profiles

BIG-IP LTM includes several profile types used for optimizing or augmenting HTTP traffic, including the following:

- HTTP
- HTTP compression
- Web acceleration

HTTP

The HTTP profile enables the use of HTTP features in BIG-IP LTM policies and iRules. It is required for using other HTTP profile types, such as HTTP compression and web acceleration.

In BIG-IP 11.5 and later, HTTP profiles can operate in one of the following proxy modes:

**Table 5.2 HTTP proxy modes**

<table>
<thead>
<tr>
<th>Proxy</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse</td>
<td>You can specify the Reverse Proxy Mode to enable the BIG-IP system to manage responses from multiple servers.</td>
</tr>
</tbody>
</table>
The Explicit Proxy Mode enables the BIG-IP system to handle HTTP proxy requests and function as a gateway. By configuring browser traffic to use the proxy, you can control whether to allow or deny a requested connection, based on configured policies. The Explicit Proxy Mode requires a DNS resolver, specified in the Explicit Proxy area of the screen.

The Transparent Proxy Mode enables the BIG-IP system to forward invalid HTTP traffic to a specified server, instead of dropping the connection. By configuring an HTTP profile to forward invalid HTTP traffic, you can manage various atypical service provider scenarios, such as HTTP traffic from non-browser clients that function as web browsers.

Here are some of the most common options:

- **Response chunking** changes the behavior of BIG-IP LTM handling of a chunked response from the server. By default it is set to **Selective**, which reassembles and then re-chunk chunked responses while preserving unchunked responses. This is the recommended value. However, setting this to **Unchunk** may be required if clients are unable to process chunked responses.

- **Connect Transformations** controls whether BIG-IP LTM modifies Connection: Close headers sent in response to HTTP/1.0 requests to **Keep-Alive**. This is enabled by default. It allows HTTP/1.0 clients to take advantage of OneConnect.

- **Redirect Rewrite** allows BIG-IP LTM to modify or strip redirects sent by pool members in order to prevent clients from being redirected to invalid or unmanaged URLs. This defaults to **None**, which disables the feature.

- **Insert X-Forwarded-For** causes BIG-IP LTM to insert an X-Forwarded-For HTTP header in requests. This reveals the client IP address to servers even when source address translation is used. By default, this is **Disabled**.

- **HTTP Strict Transport Security (HSTS)** causes BIG-IP LTM 12.0 and later to insert the RFC6797 STS header into the server response back to the client. For more information about HSTS, refer to the OWASP website (https://www.owasp.org). **Disabled** by default.

For more information, refer to **HTTP Profiles** in **BIG LTM Concepts**.

HTTP Compression profile

An HTTP Compression profile allows BIG-IP LTM to compress HTTP response content, which can substantially reduce network traffic. Although data compression can also be performed by most web servers, using BIG-IP LTM to compress traffic for load balanced applications can help minimize server load and increase total capacity.

If the web server has already compressed the data, the BIG-IP system cannot perform traffic inspection.
**Note** F5 recommends reviewing AskF5 and/or manuals regarding the options available for HTTP compression profiles before implementing this feature. However, the default profile compresses most text including HTML, JavaScript, and XML data. Most image data cannot be further compressed.

The following table lists some common HTTP Compression profile options and their details:

<table>
<thead>
<tr>
<th>Option</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep Accept Encoding</td>
<td>Disabled by default. Enable only if you want server to compress data instead of having BIG-IP LTM do it. Default behavior is to strip the Accept-Encoding header from requests, which prevents HTTP servers from compressing responses.</td>
</tr>
<tr>
<td>HTTP/1.0 Requests</td>
<td>Disabled by default. Allows BIG-IP LTM to compress responses to HTTP/1.0 requests.</td>
</tr>
<tr>
<td>CPU Saver</td>
<td>Enabled by default. (Recommended.) Allows BIG-IP LTM to reduce compression levels and even disable compression when CPU use reaches a predefined threshold.</td>
</tr>
<tr>
<td>gzip</td>
<td>Allows you to adjust the compression level. Higher values increase compression efficiency in exchange for a higher latency and higher resource cost in both CPU and memory. Default settings balance resource use, allow minimal latency, and promote effective compression.</td>
</tr>
</tbody>
</table>

For more information about HTTP compression profile options, refer to AskF5 article: [K15434: Overview of the HTTP Compression profile](https://www.askf5.com/knowledgebase/K15434).

### Web Acceleration profile

A Web Acceleration profile enables caching behavior in BIG-IP LTM, according to guidelines set in RFC 2616. Web acceleration is highly recommended if you are using HTTP compression and have a large amount of static content, as BIG-IP LTM can cache the compressed response, resulting in reduced BIG-IP LTM and server load.

F5 recommends customizing cache settings to maximize retention of static data. For more information, refer to AskF5 article: [K14903: Overview of the Web Acceleration profile](https://www.askf5.com/knowledgebase/K14903).

### SSL Profiles

BIG-IP LTM supports encrypting both the client-side and server-side flows on full proxy virtual servers. Client SSL profiles encrypt and decrypt the client-side flow (BIG-IP LTM acts as the SSL server), and server SSL profiles encrypt and decrypt the server-side flow (BIG-IP LTM acts as the SSL client).

When no SSL profile is enabled on a virtual server but the pool members expect SSL, it is called SSL Passthrough and no L7 profiles may be used. A FastL4 virtual server may be preferred for maximum performance in this instance.
SSL Offload

In SSL offload mode, only a client SSL profile is used. While the connection between the client and the BIG-IP system is encrypted, the connection from BIG-IP LTM to the pool members is unencrypted. This completely removes the requirement for the pool members to perform SSL encryption and decryption, which can reduce resource usage on pool members and improve overall performance.

SSL offload should only be used on trusted, controlled networks.

**Note** The BIG-IP Remote Crypto Offload feature may be limited in performance due to multiple factors, which can include licensing and remote host hardware capabilities.

SSL re-encryption

In SSL re-encryption mode, both client SSL and server SSL profiles are configured. The connection to pool members is also encrypted. This requires that the pool members perform SSL encryption and decryption as well, but offers security on the server-side flow.

Server SSL profiles can also be configured with a client certificate to authenticate using **SSL client certificate auth** to the pool member. This method can be used to ensure that the service can only be accessed using BIG-IP LTM virtual server, even if a client can initiate connections directly to the pool members.

You can use an server SSL profile without a client SSL profile, but the client-side connection is not encrypted.

Other SSL features

- **Proxy SSL** allows clients to do client certificate authentication directly to the pool member while still allowing BIG-IP LTM to inspect the traffic.
SSL Forward Proxy (BIG-IP 11.5.0 and later) allows BIG-IP LTM to intercept SSL traffic destined for external third-party servers. For more information refer to the relevant sections of *BIG-IP Local Traffic Manager: Concepts* and *BIG-IP Local Traffic Manager: Monitors Reference* for your version.

**Note** For information about how to locate F5® product guides, refer to AskF5™ article: K12453464: Finding product documentation on AskF5.

### SSL profile options

The following table lists some common SSL profile options and their details:

#### Table 5.4  Common SSL profile options

<table>
<thead>
<tr>
<th>Option</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate</td>
<td>Specifies the certificate BIG-IP LTM offers. On client SSL profiles, this must be a server certificate; on server SSL, it must be a client certificate.</td>
</tr>
<tr>
<td>Key</td>
<td>Specifies the key which is used for authentication purposes. It must match the configured certificate. Instructions to verify this are available on AskF5. When attempting to configure a certificate and key, BIG-IP LTM generates an error if they do not match.</td>
</tr>
<tr>
<td>Chain (ClientSSL only)</td>
<td>Allows you to configure an intermediate or chain certificate that BIG-IP LTM presents in addition to its own certificate. Contact your certificate authority to determine whether you need this.</td>
</tr>
<tr>
<td>Passphrase</td>
<td>Allows you to specify the passphrase if you are using an encrypted key. This increases security because if the key is lost it is still not compromised. The passphrase is also encrypted in the configuration using the SecureVault TMOS® feature.</td>
</tr>
<tr>
<td>Ciphers</td>
<td>Specifies which ciphers are enabled for use by this SSL profile in an OpenSSL-format cipher list. Defaults vary between client and server and between versions.</td>
</tr>
<tr>
<td>Options</td>
<td>May be used to selectively disable different SSL and TLS versions, if required by your security policy.</td>
</tr>
<tr>
<td><strong>Client Authentication</strong> (client SSL only)</td>
<td>Contains parameters which allow BIG-IP LTM to verify certificates presented by an SSL client.</td>
</tr>
<tr>
<td><strong>Server Authentication</strong> (server SSL only)</td>
<td>Contains parameters which allow BIG-IP LTM to validate the server certificate; normally, it is ignored.</td>
</tr>
</tbody>
</table>

**Note** On client SSL profiles in BIG-IP 11.6 and later, the **Certificate Key Chain** section must be saved using the **Add** button. You can now configure multiple key and certificate pairs (including a chain certificate, if required) per SSL profile: one for RSA and one for DSA.

For a complete list of SSL profiles and their options, refer to AskF5 articles: K14783: Overview of the Client SSL profile (11.x - 13.x) and K14806: Overview of the Server SSL profile (11.x - 13.x).

### Maximum Aggregate Renegotiation option

In BIG-IP 12.0 and later, you can use a **Maximum Aggregate** setting for renegotiation to mitigate DDoS attacks.
When a client connects to an SSL server, a DDoS attack can occur by the client sending requests for renegotiation of parameters. Because the server has to work harder than the client, it is possible for a malicious attack to over-burden the server to the point that its resources can no longer accept any more traffic, good or bad.

In earlier versions F5 provides the ability to protect against such attacks by limiting the number of renegotiations per connection; however, an attacker could use multiple connections to generate the DDoS attack, staying under the max for the setting in each of those connections.

Recommendation is to estimate a conservative baseline for maximum number of what the box can tolerate under its workload and then set the aggregate number to leave some ceiling available. This feature drops all SSL connections that are renegotiating above that limit, good or bad. A legitimate client issuing a single renegotiation request are dropped if malicious clients have already pushed the renegotiation count above the limit. This feature is a last resort to preserve the box rather than punishing the attackers.

In BIG-IP 12.0 and later, with the **Maximum Aggregate Renegotiation** option, you can set a cumulative number of parameter renegotiations possible from all profiles. The option is disabled by default. F5 recommends talking with an F5 Consultant Service before enabling.

**Cipher strings and SSL/TLS version**

For information on specifying SSL/TLS cipher strings for use with SSL profiles, refer to following AskF5 articles:

- [K13163: SSL ciphers supported on BIG-IP platforms (11.x - 13.x)](https://support.f5.com/kb/article/K13163)
- [K15194: Overview of BIG-IP SSL/TLS cipher suite](https://support.f5.com/kb/article/K15194)
- [K8802: Using SSL ciphers with BIG-IP Client SSL and Server SSL profiles](https://support.f5.com/kb/article/K8802)

**BIG-IP LTM policies**

In BIG-IP 11.4.0 and later, BIG-IP LTM policies supersede HTTP class profiles. BIG-IP LTM policies allow functionality previously only available through iRules.

BIG-IP LTM policies are assigned to virtual servers on the **Resources** page under **General Properties** and **Rules**.

**General properties**

The most important settings in **General Properties** are **Requires** and **Controls**.

- **Requires** allows you to specify what information the policy is able to use to make decisions.
- **Controls** allows you to specify the types of actions it is able to take.

Some selections from **Requires** and **Controls** need particular profiles to be assigned to virtual servers which use this policy. For example, a **Requires** setting of **http** depends on an HTTP profile, and a **Controls** setting of **server-ssl** depends on a server SSL profile.

**Note** In BIG-IP 12.0.1 and later, **Requires** and **Controls** are automatically configured.
Rules

Each Rule is composed of **Conditions** and **Actions**.

- **Conditions** define the requirements for the rule to run.
- **Actions** define the action taken if the **Conditions** match.

By default, the first Rule with matching **Conditions** is applied. This behavior can be changed using the **Strategy** setting under **General Properties**. For in-depth information about this behavior, refer to AskF5 article: [K15085: Overview of the Local Traffic Policies feature (11.4.0 - 12.0.0)](https://www.askf5.com/K15085).

A common use case for BIG-IP LTM policies is to select a pool based on HTTP URI. For this you need a BIG-IP LTM policy that **Requires** http and **Controls** forwarding. You can then configure Rules with an operand of http-uri and an **Action** with a target of forward to a pool.

BIG-IP LTM policies offer room for expansion: Rules can have multiple operands (all must match for the rule to match) or actions. Using the previous example, if some pools for a virtual server require ServerSSL while others do not, you could add **Controls** and **Action** of server-ssl to **disable** ServerSSL when it is not needed. You can also introduce advanced HTTP caching or compression behavior using BIG-IP LTM policies. For example, if some pools are just for static content, you may want to enable caching only for those pools.

BIG-IP LTM policies offer a broad range of capabilities, and new versions frequently introduce additional capabilities. For more information, refer to **BIG-IP LTM Concepts**.

**Note** For information about how to locate F5® product guides, refer to AskF5™ article: [K12453464: Finding product documentation on AskF5](https://www.askf5.com/K12453464).

Persistence profiles

Many applications served by BIG-IP LTM are session-based and require the client to be load balanced to the same pool member for the duration of that session. BIG-IP LTM can accomplish this requirement through persistence.

When a client connects to a virtual server for the first time, a load balancing decision is made, then the configured persistence record is created for that client. All subsequent connections that the client makes to that virtual server are sent to the same pool member for the life of that persistence record.

**Common persistence types**

Cookie persistence uses an HTTP cookie stored on a client’s computer to allow the client to reconnect to the same server previously visited at a web site. Because a cookie is an object of the HTTP protocol, use of a cookie persistence profile requires that the virtual server also have an HTTP profile assigned. The cookie persistence profile works in the following methods:

**Cookie insert**

BIG-IP LTM inserts an additional cookie into the response of a pool member. By default, the cookie has no explicit expiration time, making it valid for the life of the browser session. This is the most common and seamless method of cookie persistence to deploy. For more information on the cookie insert method refer to AskF5 article: [K6917: Overview of BIG-IP persistence cookie encoding](https://www.askf5.com/K6917).
Note Unencrypted cookies may be subject to unintended information disclosure (this is the default behavior). If the security policies of your organization require this cookie value to be encrypted, refer to AskF5 article: K14784: Configuring BIG-IP cookie encryption (10.x - 12.x).

Cookie passive and cookie rewrite

With these methods, the pool member must provide all or part of the cookie contents. Properly implementing the cookie passive and cookie rewrite method requires close collaboration between the BIG-IP LTM administrator and the application owner.

Source address affinity

Directs session requests to the same pool member based on the source IP address of a packet. This profile can also be configured with a network mask so that multiple clients can be grouped into a single persistence method. The network mask can also be used to reduce the amount memory used for storing persistence records. The defaults for this method are a timeout of 180 seconds and a mask of 255.255.255.255 (one persistence record per source IP). For more information, refer to AskF5 article: K5779: BIG-IP LTM memory allocation for Source Address Affinity persistence records.

Universal persistence directs session requests to the same pool member based on customizable logic written in an iRules. This persistence method is commonly used in many BIG-IP iApps.

Other persistence types

The following persistence methods are also available.

SSL persistence

SSL persistence tracks non-terminated SSL sessions using the SSL session ID. Even when the client's IP address changes, BIG-IP LTM still recognizes the connection as being persistent based on the session ID. This method can be problematic when working with clients who frequently renegotiate the SSL session.

For more information, refer to AskF5 article: K3062: Using SSL session ID persistence.

Destination address affinity

Destination address affinity directs session requests to the same pool member based on the destination IP address of a packet. This method is commonly used for load balancing outbound traffic over dual WAN circuits or to multiple network devices. This method should not be used when load balancing application traffic to load balance pools.

Hash persistence

Hash persistence is similar to universal persistence, except that the persistence key is a hash of the data, rather than the data itself. Hash persistence may be useful when it is necessary to obfuscate the data that is being persisted upon.
Microsoft Remote Desktop Protocol persistence

Microsoft Remote Desktop Protocol (MSRDP) persistence tracks sessions between clients and pool members based on either a token provided by a Microsoft Terminal Services Session Directory/TS Session Broker server or the username from the client.

**Note** Since the routing tokens may end up identical for some clients, the BIG-IP system may persist the RDP sessions to the same RDP servers.

You can use Session Initiation Protocol (SIP) persistence for servers receiving SIP messages sent through UDP, SCTP, or TCP.

**Persistence performance considerations**

Each persistence method has a different performance implication. Some methods require the BIG-IP system to maintain an internal state table, which consumes memory and CPU resources. Other persistence methods may not be granular enough to ensure uniform distribution amongst pool members. For example, the source address affinity persistence method may make it difficult to uniquely identify clients on corporate networks whose source addresses are masked by firewalls and proxy servers.

**CARP hash persistence**

The source address, destination address, and hash persistence profiles support a Cache Array Routing Protocol (CARP) hash algorithm. When using CARP hash, the BIG-IP system performs a computation with the persistence key and each pool member to deterministically choose a pool member. Given a set of available pool members, a client connection is always directed to the same pool member. If a pool member becomes unavailable, the sessions to that pool member is distributed among the remaining available pool members, but sessions to the available pool members remains unaffected.

**CARP advantages**

- The BIG-IP system maintains no state about the persistence entries, so CARP does not increase memory utilization.
- There is no need to mirror persistence. Given a persistence key and the same set of available pool members, two or more BIG-IP systems reaches the same conclusion.
- Persistence does not expire: Given the same set of available pool members, a client is always directed to the same pool member.

**CARP disadvantages**

- When a pool member becomes available (either due to addition of a new pool member or change in monitor state), new connections from some clients are directed to the newly available pool member at a disproportionate rate.

For more information about CARP hash, refer to AskF5 article: [K11362: Overview of the CARP hash algorithm](#).
For more information on the configuration for persistence profiles, refer to *BIG-IP Local Traffic Manager: Concepts* for your BIG-IP version.

**Note** For information about how to locate F5® product guides, refer to AskF5™ article: K12453464: Finding product documentation on AskF5.

Other protocols and profiles

To manage application layer traffic, you can use any of the following profile types. For more information refer to *BIG-IP Local Traffic Manager: Profiles Reference*.

**Table 5.5 Profile types**

<table>
<thead>
<tr>
<th>Profile type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Transfer Protocol (FTP) profile</td>
<td>Allows modifying a few FTP properties and settings to your specific needs.</td>
</tr>
<tr>
<td>Domain Name System (DNS) profile</td>
<td>Allows users to configure various DNS attributes and allow for many of the BIG-IP system DNS features such as DNS caching, DNS IPv6 to IPv4 translation, DNSSEC, etc.</td>
</tr>
<tr>
<td>Real Time Streaming Protocol (RTSP)</td>
<td>A network control protocol designed for use in entertainment and communications systems to control streaming media servers.</td>
</tr>
<tr>
<td>Internet Content Adaptation Protocol (ICAP)</td>
<td>Used to extend transparent proxy servers, thereby freeing up resources and standardizing the way in which new features are implemented.</td>
</tr>
<tr>
<td>Request Adapt or Response Adapt profile</td>
<td>Instructs an HTTP virtual server to send a request or response to a named virtual server of type Internal for possible modification by an Internet Content Adaptation Protocol (ICAP) server.</td>
</tr>
<tr>
<td>Diameter</td>
<td>An enhanced version of the Remote Authentication Dial-In User Service (RADIUS) protocol. When you configure a Diameter profile, the BIG-IP system can send client-initiated Diameter messages to load balancing servers.</td>
</tr>
<tr>
<td>Remote Authentication Dial-In User Service (RADIUS)</td>
<td>Used to load balance RADIUS traffic.</td>
</tr>
<tr>
<td>Session Initiation Protocol (SIP)</td>
<td>A signaling communications protocol, widely used for controlling multimedia communication sessions, such as voice and video calls over Internet Protocol (IP) networks.</td>
</tr>
<tr>
<td>Simple Mail Transfer Protocol (SMTP)</td>
<td>Secures SMTP traffic coming into the BIG-IP system. When you create an SMTP profile, BIG-IP Protocol Security Manager provides several security checks for requests sent to a protected SMTP server.</td>
</tr>
<tr>
<td>SMTPS profile</td>
<td>Provides a way to add SSL encryption to SMTP traffic quickly and easily.</td>
</tr>
<tr>
<td>iSession® profile</td>
<td>Tells the system how to optimize traffic. Symmetric optimization requires an iSession profile at both ends of the iSession connection.</td>
</tr>
<tr>
<td>Profile type</td>
<td>Details</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Rewrite profile</td>
<td>Instructs BIG-IP LTM to display websites differently on the external network than on an internal network and can also be used to instruct the BIG-IP system to act as a reverse proxy server.</td>
</tr>
<tr>
<td>Extensible Markup Language (XML)</td>
<td>Causes the BIG-IP system to perform XML content-based routing requests to an appropriate pool, pool member, or virtual server based on specific content in an XML document.</td>
</tr>
<tr>
<td>Speedy (SPDY)</td>
<td>An open-source web application layer protocol developed by Google in 2009 and is primarily geared toward reducing Web page latency. By using the SPDY profile, the BIG-IP system can enhance functionality for SPDY requests.</td>
</tr>
<tr>
<td>Financial Information Exchange (FIX)</td>
<td>An electronic communications protocol for international real-time exchange of information related to the securities transactions and markets.</td>
</tr>
<tr>
<td>Video Quality of Experience (QOE)</td>
<td>Allows assessment of an audience’s video session or overall video experience, providing an indication of application performance.</td>
</tr>
</tbody>
</table>

**Troubleshooting**

To resolve most traffic-related issues, review the virtual server and networking-level configurations. These may include pools and address translation settings. You may also need to review the following:

- Architectural design for the virtual server in question to verify the correct profile is used.
- Custom profiles and their associated options to ensure they modify traffic behavior as anticipated.
- Next, revert to an unmodified default profile and observe changes in behavior.
This chapter reviews BIG-IP® DNS™ offerings available from F5®.

DNS Services features

**BIG-IP DNS** (formerly BIG-IP GTM™) is a DNS-based module which monitor the availability and performance of global resources, such as distributed applications, in order to control network traffic patterns.

**DNS caching** is a DNS services feature that can provide responses for frequently requested DNS records from a cache maintained in memory on BIG-IP systems. This feature can be used to replace or reduce load on other DNS servers.

**DNS Express®** is a DNS Services feature that allows the BIG-IP system to act as an authoritative slave server. DNS Express relies on NOTIFY, AXFR, and IXFR to transfer zone data from a master authoritative server and store it in memory for high performance. DNS Express does not offer DNS record management capability.

**Response Policy Zones (RPZ)** allow the BIG-IP system, when configured with a DNS cache, to filter DNS requests based on the resource name being queried. This can be used to prevent clients from accessing known-malicious sites.

Upgrading to BIG-IP DNS 12.0 and later

**Prerequisites**

F5 requires BIG-IP 11.x to be installed before upgrading. Upgrades to BIG-IP DNS 12.0 and later from versions prior to 11.x are not supported.

TMOS® Shell (tmsh) scripts on BIG-IP 11.x and earlier use DNS pool and Wide IP commands do not work correctly. They must be converted to use the new pool and Wide IP Type parameters.

If any DNS objects are currently active in the configuration DNS, configuration files created on BIG-IP devices prior to BIG-IP 12.0 (for example bigip_DNS.conf, UCS, SCF) cannot be loaded on a BIG-IP device running BIG-IP 12.0.

To load the DNS configuration, the DNS configuration must be blank. Load an empty bigip_DNS.conf and save the configuration. Once the DNS configuration are blanked, a bigip_DNS.conf, UCS, or SCF created in a prior version can be loaded.

It is possible to use the deprecated DNS iControl API interfaces Pool, Wide IP, Application, and Pool Member commands against a BIG-IP devise running BIG-IP 12.0.

If the configuration has not been modified to use any of the newly supported types, F5 recommends that you transition to using the new DNS iControl API interfaces Poolv2, Wide IPv2.
BIG-IP DNS/DNS Services Basics

BIG-IP DNS is the module built to monitor the availability and performance of global resources and use that information to manage network traffic patterns.

With BIG-IP DNS module you can:

- Direct clients to local servers for globally-distributed sites using a GeoIP database.
- Change the load balancing configuration according to current traffic patterns or time of day.
- Set up global load balancing among disparate BIG-IP LTM® systems and other hosts.
- Monitor real-time network conditions.
- Integrate a content delivery network from a CDN provider.

To implement BIG-IP DNS you need to understand the following terminology and basic functionality:

- **Configuration synchronization** ensures the rapid distribution of BIG-IP DNS settings among BIG-IP DNS systems in a synchronization group.

- **Load balancing** divides work among resources so that more work gets done in the same amount of time and, in general, all users get served faster. BIG-IP DNS selects the best available resource using either a static or a dynamic load balancing method. When using a static load balancing method, BIG-IP DNS selects a resource based on a pre-defined pattern. When using a dynamic load balancing method, BIG-IP DNS selects a resource based on current performance metrics.

- **Prober pool** is an ordered collection of one or more BIG-IP systems that can be used to monitor specific resources.

- **Wide IP** is a mapping of a fully-qualified domain name (FQDN) to a set of virtual servers that host the domains content, such as a web site, an e-commerce site, or a content delivery network (CDN). BIG-IP DNS intercepts requests for domain names that are Wide IPs and answers them based on the Wide IP configuration.

- **iQuery** is an XML protocol used by BIG-IP DNS to communicate with other BIG-IP systems.

- **BIG-IP DNS listener** is a specialized virtual server that provides DNS services.

- **Probe** is an action the BIG-IP system takes to acquire data from other network resources. BIG-IP DNS uses probes to track the health and availability of network resources.

- **Data center** is where BIG-IP DNS consolidates all the paths and metrics data collected from the servers, virtual servers, and links.

- **Virtual server** is a combination of IP address and port number that points to a resource that provides access to an application or data source on the network.

- **Link** is a logical representation of a physical device (router) that connects the network to the Internet.

- **Domain Name System Security Extensions (DNSSEC)** is an industry-standard protocol that functions to
provide integrity for DNS data.

For more information, refer to "BIG-IP DNS load balancing".

BIG-IP DNS/DNS Services Core Concepts

This section covers DNS Express, DNS cache, Auto-Discovery, Address translation, and ZoneRunner are covered in this chapter.

Configuration synchronization

Configuration synchronization ensures the rapid distribution of BIG-IP DNS settings to other BIG-IP DNS systems that belong to the same synchronization group. A BIG-IP DNS synchronization group might contain both BIG-IP DNS and BIG-IP Link Controller systems.

Configuration synchronization occurs in the following manner:

• When a change is made to a BIG-IP DNS configuration, the system broadcasts the change to the other systems in BIG-IP DNS synchronization group.

• When a configuration synchronization is in progress, the process must either complete or time out before another configuration synchronization can occur.

It is important to have a working Network Time Protocol (NTP) configuration because BIG-IP DNS relies on timestamps for proper synchronization.

BIG-IP DNS listeners

A listener is a specialized virtual server that provides DNS services on port 53 and at the IP address assigned to the listener. When a DNS query is sent to the listener, BIG-IP DNS either handles the request locally or forwards the request to the appropriate resource.

BIG-IP DNS responds to DNS queries on a per-listener basis. The number of listeners created depends on the network configuration and the destinations to which specific queries are to be sent. For example, a single BIG-IP DNS can be the primary authoritative server for one domain, while forwarding other DNS queries to a different DNS server. BIG-IP DNS always manages and responds to DNS queries for the Wide IPs that are configured on the system.

Data centers and virtual servers

All of the resources on a network are associated with a data center. BIG-IP DNS consolidates the paths and metrics data collected from the servers, virtual servers, and links in the data center. BIG-IP DNS uses that data to conduct load balancing and route client requests to the best-performing resource, based on a variety of factors.

BIG-IP DNS may send all requests to one data center when another data center is down. This may work well for disaster recovery sites.

Alternatively, BIG-IP DNS might send a request to the data center that has the fastest response time.
Or, BIG-IP DNS may send a request to the data center that is located closest to the client’s source address. For example, the system may send a client located in France to a host also located France rather than the United States, greatly reducing traffic round-trip times.

**Note** The resources associated with a data center are available only when the data center is also available.

### Virtual servers

A virtual server is a specific IP address and port number that points to a resource on the network. In the case of host servers, this IP address and port number likely point to the resource itself. With load balancing systems, virtual servers are often proxies that allow the load balancing server to manage a resource request across a large number of resources.

**Tip** You can configure virtual server status to be dependent only on the timeout value of the monitor associated it. In a multi-bladed environment, this configuration ensures that when the primary blade in a cluster becomes unavailable, the gtmd agent on the new primary blade has time to establish new iQuery connections with and receive updated status from other BIG-IP systems.

**Note** The big3d agent on the new primary blade has 90 seconds to run (the timeout value of the BIG-IP monitor) before it times out.

### Links

A link is an optional BIG-IP DNS or BIG-IP Link Controller configuration object which represents a physical device that connects a network to the Internet. BIG-IP DNS tracks the performance of links. Performance results influence the availability of pools, data centers, Wide IPs, and distributed applications.

When you create one or more links, the BIG-IP system uses the following logic to automatically associate virtual servers with the link objects:

- BIG-IP DNS and BIG-IP Link Controller™ associate the virtual server with the link by matching the subnet addresses of the virtual server, link, and self IP address. Most of the time, the virtual server is associated with the link that is on the same subnet as the self IP address.

- In some cases, BIG-IP DNS and BIG-IP Link Controller cannot associate the virtual server and link because the subnet addresses do not match. When this occurs, the system associates the virtual server with the default link which is assigned to the data center. This association may cause issues if the link that is associated with the virtual server does not provide network connectivity to the virtual server.

- If the virtual server is associated with a link that does not provide network connectivity to that virtual server, BIG-IP DNS and BIG-IP Link Controller may incorrectly return the virtual server IP address in the DNS response to a Wide IP query even if the link is disabled or marked as down.

### DNS Express

**DNS Express** enables the BIG-IP system to function as a replica authoritative nameserver and answer DNS queries at high speeds. However, since DNS Express doesn’t use BIND DNS software, it doesn’t have the same security vulnerabilities as a typical BIND implementation.
DNS Express supports the standard DNS NOTIFY protocol from primary authoritative nameservers and uses the AXFR/IXFR mechanism to transfer zone data. The primary authoritative nameserver is not listed in the start of authority (SOA) of the zone data, and is therefore protected, or hidden.

Optionally, transaction signature (TSIG) may be used to secure the zone data transferred from the primary nameserver.

DNS Express doesn’t support modifying records. Instead, records are modified on the primary nameserver and DNS Express is notified of the changes. However, the BIG-IP system may be configured as the primary authoritative nameserver using Zonerunner.

**DNS Anycast**

You can configure IP Anycast for DNS services on BIG-IP systems that have the advanced routing module license. Anycast describes a one-to-nearest communication between a client and the nearest recipient within a group. The routing protocol directs client queries to a recipient in the target group based on the routing algorithm for the specified protocol. This capability improves reliability and performance, while distributing load across multiple BIG-IP systems.

In order to enable IP Anycast for DNS services, ZebOS dynamic routing needs to be enabled and configured with the appropriate protocol for your deployment. Then, listeners must be configured on each of the BIG-IP systems with the shared IP Anycast address and Route Advertisement enabled under the advanced settings of the listener.

**DNS Anycast benefits**

Reliability is improved because DNS queries are sent to the Anycast IP address that is defined on multiple BIG-IP systems. If a system becomes unavailable, the route to that system is removed dynamically.

Performance is improved by routing queries to the nearest BIG-IP system.

Distributing the load across multiple, geographically distributed BIG-IP systems helps mitigate distributed denial-of-service attacks (DDoS).

**DNS cache**

The DNS cache feature is available as a DNS add-on module for BIG-IP LTM. DNS Cache has three different configurable forms of DNS cache: **Transparent**, **Resolver**, and **Validating Resolver**.

**Transparent DNS cache**

The transparent cache object is configurable on the BIG-IP system to use external DNS resolvers to resolve queries, and then cache the responses from the multiple external resolvers. When a consolidated cache is in front of external resolvers (each with their own cache), it can produce a much higher cache hit percentage.
F5 recommends that you configure the BIG-IP system to forward queries which cannot be answered from the cache to a pool of local DNS servers, rather than to the local BIND instance because BIND performance is slower than using multiple external resolvers.

For systems using the DNS Express feature, if DNS DNS Express is available to answer, no further processing is needed.

Tip It is possible to configure the local BIND instance on the BIG-IP system to act as an external DNS resolver. However, the performance of BIND is slower than using a resolver cache.
Validating Resolver DNS cache

The Validating Resolver DNS cache may be configured to recursively query public DNS servers, validate the identity of the DNS server sending the responses, and then cache the responses. The next time the system receives a query for a response that exists in the cache, the system returns the DNSSEC-compliant response from the cache.

The validating resolver cache contains messages, resource records, the nameservers the system queries to resolve DNS queries, and DNSSEC keys.

For more information about setting up each of the DNS express caching methodologies, refer to DNS Cache: Implementations.

Note For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

DNS Cache Optimization

DNS Cache optimization can be complex. You want to maximize the number of DNS cache hits while conserving allocated memory.

There are diminishing returns to cache too much. An average cache has a hit rate of 80–85 percent, and a majority of those records typically require less than 1 gigabyte of space. If your cache is larger than 1 gigabyte, you may be taking up memory space by caching objects that are not requested often.

To optimize your cache, set it for a few hundred MB, observe the cache, and then make adjustments as necessary.
To view and adjust cache size using the Configuration utility

1. Navigate to: Local Traffic >> DNS Caches : DNS Cache List.
2. Click the name of the cache.
3. Note the Message Cache size.
   This setting is in bytes and is a per Traffic Management Microkernel (TMM) setting.
4. Multiply the number of CPU cores by this value to get the total memory allocation space.
5. Note the Resource Record Cache size.
   This setting is in bytes and is a per TMM setting.
6. Multiply the number of CPU cores by this value to get the total memory allocation space.
7. Set the Nameserver Cache Count field as desired.
8. In the Unsolicited Reply Threshold field, change the default value if you are using the BIG-IP system to monitor for unsolicited replies using SNMP.

Note The BIG-IP system always rejects unsolicited replies.

The default value of 0 (off) indicates the system does not generate SNMP traps or log messages when rejecting unsolicited replies.

Changing the default value alerts you to a potential security attack, such as cache poisoning or DDoS. For example, if you specify 1,000,000 unsolicited replies, each time the system receives 1,000,000 unsolicited replies, it generates an SNMP trap and log message.

Check the number of cache hits that you receive as seen in the cache statistics. Clear the counters and check again during peak usage hours.

Observe and record the percentage of cache hits in relation to the total cache size.

9. Decrease the cache settings and check the cache statistics again. Note whether or not the number of cache hits remain the same.
10. Continue to adjust the cache settings until you reach a desired balance between the numbers of cache hits versus cache sizing.

iRules® may also be used to manipulate a DNS response given from cached resource records. For example:

Assume that we wish to decrease the number of repeat DNS cache hits if the number of available pool members for a Transparent Cache drops beneath a given number. This goal can be achieved if we double the TTL given to a client if the number of pool members drops beneath x.
Hardware acceleration

For BIG-IP LTM 12.0 running the appropriate hardware, DNS response cache and protocol validation can be accelerated in hardware to lighten load, help mitigate DDoS attacks, and improve response time for DNS queries. Disabled by default, these options can be enabled on the DNS Profile of the UDP listener with the following prerequisites:

- BIG-IP LTM 12.0 or later installed on the appropriate hardware.
- Intelligent L7 Bitstream is enabled on the BIG-IP system.

**Note** Hardware acceleration features only work on BIG-IP LTM 12.x used with B2250 blades in the 2x00 series VIPRION chassis with L7 Intelligent Bitstream enabled. The BIG-IP system does not prevent you from enabling the features, even if your system does not meet these criteria. Enabling them without the appropriate criteria has no effect on your BIG-IP system.

DNS response cache in hardware

The DNS Response cache is designed to service millions of queries per second. It caches responses from any of the software caches listed in the previous section as well as DNS Express. Given their dynamic nature, GTM responses are never cached. The Hardware cache is flushed every second, requiring another DNS query.

Protocol validation in hardware

By moving some protocol validation into hardware, system performance is enhanced by quickly discarding malformed DNS queries. This relieves the burden on the CPU and provides increased DDoS protection.

**Troubleshooting**

If these options do not function, make sure of the following:

- You have BIG-IP LTM 12.0 or later installed on the appropriate hardware.
- You have enabled L7 Intelligent Bitstream.
- You have enabled the options in the DNS Profile.

**Tip** To enable these options on your DNS profile, F5 recommends making a copy of the default profile and updating the copy, leaving the default unchanged.

With the Hardware Acceleration features enabled, most queries are answered from cache.

**To view statistics in the Configuration utility**

- Navigate to **Statistics >> Module Statistics: DNS: Delivery**
DNSSEC

DNSSEC is an extension to the Domain Name Service (DNS) that ensures the integrity of data returned by domain name lookups by incorporating a chain of trust in the DNS hierarchy. DNSSEC provides origin authenticity, data integrity and secure denial of existence.

Specifically, origin authenticity ensures that resolvers can verify that data has originated from the correct authoritative source. Data Integrity verifies that responses are not modified in-flight, and Secure Denial of Existence ensures that when there is no data for a query, that the authoritative server can provide a response that proves no data exists.

The basis of DNSSEC is public key cryptography (PKI). A chain of trust is built with public-private keys at each layer of the DNS architecture.

DNSSEC key types

DNSSEC uses two kinds of keys: key-signing keys and zone signing keys.

- Key signing key is used to sign other keys in order to build the chain of trust. This key is sometimes cryptographically stronger and has a longer lifespan than a Zone signing key.
- Zone signing key is used to sign the data that is published in a zone. DNSSEC uses the key signing keys and zone signing keys to sign and verify records within DNS.

DNSSEC chain of trust

When a user requests a site, DNS translates the domain name into an IP address through a series of recursive lookups that form a “chain” of requests. Each stop in this chain inherently trusts the other parts of the chain, and this trust may be exploited by an attack.

For example, if an attack manipulates a servers or some traffic along the chain, it can redirect the client to a website where malware is waiting.

DNSSEC mitigates this problem by validating the response of each part of the chain with digital signatures. These signatures help build a “chain of trust” that DNS can rely on when answering requests. To form the chain of trust, DNSSEC starts with a “trust anchor” and everything below that trust anchor is trusted. Ideally, the trust anchor is the root zone.

ICANN published the root zone trust anchor, and root operators began serving the signed root zone in July, 2010. With the root zone signed, all other zones below it can also be signed, thus forming a solid and complete chain of trust. Additionally, ICANN also lists the Top Level Domains that are currently signed and have trust anchors published as DS records in the root zone.
The following figure shows the building blocks for the chain of trust from the root zone:

![Figure 6.4: DNSSEC chain of trust](image)

For more information, refer to **Configuring DNSSEC** in *BIG-IP DNS Services*.

**Auto-discovery**

Auto-discovery is a process through which BIG-IP DNS automatically identifies resources that it manages. BIG-IP DNS can discover two types of resources: virtual servers and links.

Each resource is discovered on a per-server basis, so you can employ auto-discovery only on the servers you specify.

The auto-discovery feature of BIG-IP DNS has three modes that control how the system identifies resources:

- **Disabled**: BIG-IP DNS does not attempt to discover any resources. Auto-discovery is disabled on BIG-IP DNS by default.
• Enabled: BIG-IP DNS regularly checks the server to discover any new resources. If a previously discovered resource cannot be found, BIG-IP DNS deletes it from the system.

• Enabled (No Delete): BIG-IP DNS regularly checks the server to discover any new resources. Unlike the Enabled mode, the Enabled (No Delete) mode does not delete resources, even if the system cannot currently verify their presence.

**Note** Enabled and Enabled (No Delete) modes query the servers for new resources every 30 seconds by default.

**Important** Auto-discovery must be globally enabled at the server and link levels, and the frequency at which the system queries for new resources must be configured.

For information about enabling auto-discovery on virtual servers and links, refer to Discovering resources automatically in the *Configuration Guide for BIG-IP Global Traffic Manager*.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: [K12453464: Finding product documentation on AskF5](https://www.askf5.com/knowledgebase/articles/K12453464-Finding-product-documentation-on-AskF5).

### Address translation

Several objects in BIG-IP DNS allow the specification of address translation. Address translation is used in cases where the object is behind a Network Address Translation (NAT). For example, a virtual server may be known by one address on the Internet but another address behind the firewall. When configuring these objects, the address is the external address and is returned in any DNS responses generated by BIG-IP DNS.

When probing, the BIG-IP system may use either the address or translation, depending on the situation. As a general rule, if both the BIG-IP system performing the probe and the target of the probe are in the same data center and both have a translation, the probe uses the translations. Otherwise, the probe uses the address.

Specifying a translation on a BIG-IP server causes virtual server auto-discovery to silently stop working. This is because BIG-IP DNS has no way of knowing what the externally visible address should be for the discovered virtual server address, which is a translation. For more information, refer to AskF5 article: [K9138: BIG-IP GTM system disables virtual server auto-discovery for BIG-IP systems that use translated virtual server addresses](https://www.askf5.com/knowledgebase/articles/K9138-BIG-IP-GTM-system-disables-virtual-server-auto-discovery-for-BIG-IP-systems-that-use-translated-virtual-server-addresses).

### ZoneRunner

ZoneRunner™ is an F5 product used for zone file management on BIG-IP DNS. You may use the ZoneRunner utility to create and manage DNS zone files and configure the BIND instance on BIG-IP DNS. With the ZoneRunner utility, you can:

• Import and transfer DNS zone files.

• Manage zone resource records.

• Manage views.

• Manage a local nameserver and the associated configuration file, `named.conf`.

• Transfer zone files to a nameserver.
**BIG-IP DNS/DNS SERVICES—BIG-IP DNS/DNS SERVICES CORE CONCEPTS**

- Import only primary zone files from a nameserver.

BIG-IP DNS ZoneRunner utility uses dynamic update to make zone changes. All changes made to a zone using dynamic update are written to the zone’s journal file.

**Important** F5 recommends that you let the ZoneRunner utility manage the DNS/BIND file rather than manually editing the file. If manual editing is required, the zone files must be frozen to avoid issues with name resolution and dynamic updates.

To prevent the journal files from being synchronized if BIG-IP DNS is configured to synchronize DNS zone files, the zone must be frozen on all BIG-IP DNS systems.

For more information refer to **ZoneRunner** in **BIG-IP GTM: Concepts**.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: [K12453464: Finding product documentation on AskF5](#).

**iRules**

iRules can be attached to or associated with a Wide IP, and in BIG-IP 11.5.0 and later it can be attached to or associated with the DNS listener.

When using iRules with BIG-IP DNS, there are two possible places to attach iRules: either to the Wide IP or to the DNS listener. The iRules commands and events available depend on where the iRules are attached in the configuration. Some iRules functions require BIG-IP LTM to be provisioned alongside BIG-IP DNS.

BIG-IP DNS 12.0 and later includes changes to DNS iRules. Because a name is no longer sufficient to identify a Wide IP, the iRules command syntax was updated to allow explicit declaration of the Wide IP name and Type. However, a Wide IP type is not always required for iRules dealing with DNS. If a Wide IP type is not specified, the Type is automatically set during run-time to the Type of the Wide IP utilizing iRules. Therefore, it is possible to have the same iRule attached to multiple Wide IPs of different Types and allow TMM to set the Type as the iRule is executed.

**tmsh**

In BIG-IP 12.0 and later, tmsh commands support a new **Type** attribute. The following shows the updates to command syntax, which includes the Type parameters.

- `tmsh show DNS Wide IP <A|AAAA|CNAME|MX|SRV|NAPTR> [Wide IP Name]`
- `tmsh show DNS pool <A|AAAA|CNAME|MX|SRV|NAPTR> [Pool Name]`

**iControl and iControl REST**

The iControl® API allows administrators to interact with their BIG-IP devices programmatically to update configurations, view statistics, etc. Two versions of the iControl API exist: iControl and iControl REST.
iControl changes in BIG-IP DNS 12.0 and later

BIG-IP 12.0 includes changes to both iControl and iControl REST. Both versions of iControl can interface with the DNS module on BIG-IP. The addition of the Type field when defining Wide IPs necessitate a change to both iControl APIs. The support for existing customer iControl applications is limited and is explained in greater detail in the next two sections.

iControl

BIG-IP 12.0 and later supports iControl for DNS without any changes to existing applications. However, iControl only works with DNS configurations that do not use any of the new resource record types introduced in BIG-IP 12.0.

To learn more about iControl, visit F5 DevCentral iControl Wiki Home.

iControl REST

To learn more about iControl REST, visit the F5 DevCentral iControl REST Home.

BIG-IP 12.0 and later does not support iControl REST for DNS without changing the existing application. The iControl REST API functions are based on the underlying tmsh structure on the BIG-IP device. The tmsh commands for both Wide IPs and pools change in this release; therefore, the iControl REST API changes.

BIG-IP DNS load balancing

Monitors

BIG-IP DNS uses health monitors to determine the availability of the virtual servers used in its responses to DNS requests. Detailed information about monitors can be found in BIG-IP Global Traffic Manager: Monitors Reference.

Note For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

Probers

When running a monitor, BIG-IP DNS may request that another BIG-IP device probe the target of the monitor. BIG-IP DNS may choose a BIG-IP system in the same data center with the target of the monitor to actually send the probe and report back the status. This can minimize the amount of traffic that traverses the WAN.

The external and scripted monitors listed in the documentation use an external file to check the health of a remote system. BIG-IP DNS may request that another BIG-IP system in the configuration run the monitor. In order for the monitor to succeed, the remote BIG-IP system must have a copy of the external file.

Prober enhancements

BIG-IP DNS uses probers to decide if pool members are available based on algorithm criteria and then mark those
resources up or down based on the results. In BIG-IP 13.0 and later, you can configure prober criteria to better control your resources. While in previous versions prober selection is performed automatically, in BIG-IP 13.0 and later, you can control the prober selection using **Prober Preference** and **Prober Fallback** options, in addition to the existing **Prober Pool** option.

BIG-IP 13.0 also enhances the **Availability Requirements** setting by adding a **Require option** which allows you to configure the number of probers using the same monitor at the same time. In earlier versions, if any prober fails then the resource is marked down but now you can configure the monitor probers so that the resource is marked down only if the configured number of probers are unsuccessful. These new monitor rules are available for BIG-IP DNS servers, virtual servers, pools, and pool members.

**bigip monitor**

The bigip monitor can be used to monitor BIG-IP systems. It uses the status of virtual servers determined by the remote BIG-IP system rather than sending individual monitor probes for each virtual server. It is recommended that BIG-IP LTM be configured to monitor its configuration elements so that it can determine the status of its virtual servers. This virtual server status is reported through the bigip monitor back to BIG-IP DNS. This is an efficient and effective way to monitor resources on other BIG-IP systems.

**Application of monitors**

In BIG-IP DNS configuration, monitors can be applied to the server, virtual server, pool and pool member objects. The monitor defined for the server is used to monitor all of its virtual servers unless the virtual server overrides the monitor selection. Likewise, the monitor defined for the pool is used to monitor all of its pool members, unless the pool member overrides the monitor selection.

It is important not to over-monitor a pool member. If a monitor is assigned to the server and/or virtual server and also to the pool and/or pool member, then both of the monitors fire, effectively monitoring the virtual server twice. In most cases, monitors should be configured at the server/virtual server or at the pool/pool member, but not both.

**Prober pools**

Prober pools allow the specification of particular set of BIG-IP devices that BIG-IP DNS may use to monitor a resource. This might be necessary in situations where a firewall is between certain BIG-IP systems and monitored resource, but not between other BIG-IP systems and those same resources. In this case, a prober pool can be configured and assigned to the server to limit probe requests to those BIG-IP systems that can reach the monitored resource. For more information about prober pools, refer to **About Prober pools** in *BIG-IP Global Traffic Manager: Concepts*.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: [K12453464: Finding product documentation on AskF5](#).

**Wide IPs**

A Wide IP maps a fully qualified domain name (FQDN) to one or more pools. The pools contain virtual servers. When a Local Domain Name Server (LDNS) makes a request for a domain that matches a Wide IP, the configuration of the Wide IP determines which virtual server address should be returned.
Wide IP names can contain the wildcard characters * (to match one or more characters) and ? (to match one character).

For more information about Wide IPs, refer to Wide IPs in BIG-IP Global Traffic Manager: Concepts.

Note For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

Record types

BIG-IP DNS 12.0 and later supports the following Wide IP resource record types:

- MX – Mail Exchanger
- SRV – Available Services
- NAPTR – Naming Authority Pointer Record

Wide IP definitions consist of a Fully Qualified Domain Name (FQDN) and a Resource Record type. Each Wide IP must have a RR type associated with it, and any pools attached to the Wide IP must contain members of the same RR type.

Note In BIG-IP 12.0 and later, pools can no longer be a mix of RR types. In versions 11.x - 11.6, pools may contain both A and AAAA records.

The notable exception to this are CNAME pools. A CNAME pool can be attached to any Wide IP type, but it must either contain a static CNAME definition (FQDN) or a Wide IP with the RR type that is the same as Wide IP that has the CNAME pool attached.

Prior versions of BIG-IP DNS support three DNS resource record (RR) types for Wide IPs (WIP) and three Wide IPs resource record types: A records, AAAA records, and CNAME records. Pool members for a Wide IP can be a mixture of both A and AAAA virtual servers. CNAME records are configured as an attribute of a pool an attached to a Wide IP.

The following figure shows a Wide IP with a Pool and both A and AAAA Virtual Servers.

![Wide IP with A and AAAA virtual servers in a single pool](image)

The following figure shows a Wide IP with a CNAME configured on the pool
Additional RR types are supported in versions earlier than BIG-IP 12.0, but they are configured in Zonerunner and are not Wide IPs.

For more information on Zonerunner™ RR types, refer to Using ZoneRunner to Configure DNS Zones in BIG-IP DNS Services: Implementations.

**Note** DNSSEC utilizes additional resource record types. However, these record types do not correspond to a specific Wide IP configuration on the BIG-IP device.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

**Wide IPs and pools**

This section covers how to configure Wide IPs and pools on BIG-IP devices in BIG-IP DNS 12.0 and later.

Pool definitions have changed significantly for the new resource record types. Pools attached to Wide IPs can now contain other Wide IP names as members. This change is necessary because resource records of types MX, SRV, NAPTR and CNAME return FQDNs rather than IP addresses.

Wide IPs must be configured for each RR type it needs to answer. This means multiple Wide IPs exist with the same hostname but different resource record types.

**A and AAAA records**

The diagrams below show how to configure resource records of types A and AAAA. In BIG-IP 12.0 and later, each RR type has its own Wide IP to handle queries for that RR type.
CNAME records

The following figures show how CNAMEs are configured for a type A resource record and for a static target.

The following figure shows a Wide IP of type A configured. The Wide IP has a pool of type CNAME associated with it. The CNAME pool has a single pool member that is not an IP address. Instead, the pool contains the name of another Wide IP of type A. The CNAME pool must contain either another CNAME or a Wide IP of the same type as the original Wide IP requested.

MX records

The following figure shows how Wide IPs of type MX are configured.
In the first two boxes, the Wide IP type is MX and the pool type is MX. The MX pool contains hostnames for other Wide IPs configured on the BIG-IP.

Since type A and type AAAA records are valid MX hostnames, a single hostname in the MX pool points to any A or AAAA Wide IPs configured with that hostname.

**SRV records**

The following figure shows how Wide IPs of type SRV are configured. The left-most Wide IP and pool attached to it both have type SRV. The SRV pool points to a Wide IP also configured on the BIG-IP system.

**NAPTR records**

The figure below shows how Wide IPs of type NAPTR are configured. The left-most Wide IP and pool attached to it both have type NAPTR. A NAPTR pool can point to Wide IPs on the same BIG-IP with types SRV, A or AAAA. In this example, the NAPTR pool contains both type SRV and type A Wide IPs.

**BIG-IP DNS Minimal Response Setting**

In BIG-IP 12.0 and later, a configuration setting at the Wide IP level called **Minimal Response**. This value
determines whether DNS attempts resolve and return the IP addresses associated with hostnames contained in Wide IP pools. The default value for **Minimal Response** is **Enabled**. F5 chose this default because it preserves the default behavior from BIG-IP versions prior to BIG-IP 12.0.

**Examples**

Look at the examples below that show the results of a `dig` command for the hostname `_sip._udp.example.com`. When **Minimal Response** is enabled, only the **Answer Section** is returned and contains the Wide IP names contained in the SRV pool.

The following example shows Wide IP with **Minimal Response** enabled (Default):

<table>
<thead>
<tr>
<th>Question Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>_sip._udp.example.com</code> IN SRV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Answer Section:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>_sip._udp.example.com.3222</code> IN SRV 20 10 6050 sip01.example.com</td>
</tr>
<tr>
<td><code>_sip._udp.example.com.3222</code> IN SRV 10 10 6050 sip02.example.com</td>
</tr>
</tbody>
</table>

When **Minimal Response** is **Disabled**, the DNS resolves the hostnames in the **Answer** section and provides the information in the **Additional** section of the query response. The following example shows Wide IP with Minimal Response disabled.

<table>
<thead>
<tr>
<th>Question Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>_sip._udp.example.com</code> IN SRV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Answer Section:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>_sip._udp.example.com.3222</code> IN SRV 20 10 6050 sip01.example.com</td>
</tr>
<tr>
<td><code>_sip._udp.example.com.3222</code> IN SRV 10 10 6050 sip02.example.com</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Additional Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>_sip._udp.example.com.3222</code> IN A 10.10.10.50</td>
</tr>
<tr>
<td><code>_sip._udp.example.com.3222</code> IN A 10.10.10.51</td>
</tr>
</tbody>
</table>
Effects on DNS Performance

A single query to a Wide IP may include multiple additional name resolutions to Wide IP pool members. Refer to “NAPTR records” for an example of a name resolution that requires multiple additional name resolutions.

Configuring the DNS with Minimal Response disabled can have performance impacts due to the additional steps the DNS must perform for each query to a Wide IP.

Unless a specific need requires it, F5 recommends Minimal Response be set to the default of Enabled.

Load balancing logic

BIG-IP DNS provides a tiered load balancing mechanism for Wide IP resolution. At the first tier, BIG-IP DNS chooses an appropriate pool of servers, and then, at the second tier, it chooses an appropriate virtual server.

For complete information about BIG-IP DNS load balancing, refer to About load balancing and Global Traffic Manager in BIG-IP Global Traffic Manager: Concepts.

Note For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

The Wide IP has four static load balancing methods available for choice of an available pool.

The Wide IP pool has several dynamic and static load balancing methods available to choose an available pool member.

The dynamic load balancing methods rely on metrics gathered to make a load balancing decision.

The static load balancing methods make a load balancing decision based on a set pattern. The pool allows the specification of preferred, alternate and fallback load balancing options.

Not every load balancing method is available for each of the options.

When choosing an available pool member, the preferred method is tried first. When using a dynamic load balancing method as the preferred load balancing method, it is possible for load balancing to fail. For example, if the round-trip time method is chosen, when the first request arrives from an LDNS, there are no metrics available for it. Since there are no metrics available, the preferred method fails and BIG-IP DNS schedules the gathering of round trip time metrics for that LDNS.

When the preferred method fails, the system falls back to the alternate method; therefore, when using a dynamic preferred method, it is important to specify an alternate method.

The fallback method is used to ensure that the a resource is returned from the pool. The fallback method ignores the availability status of the resource being returned.

There are load balancing methods available that do not actually load balance. Methods such as none, drop packet, return to DNS, and fallback IP control the behavior of load balancing, but do not actually use the configured pool members.
Maximum number of available pool members

The BIG-IP DNS system can select multiple pool members in making load balancing decisions, and can use multiple pool members in a DNS response. In versions earlier than BIG-IP DNS 13.0.0, the maximum number of available pool members that the system can return in a response is limited to 16.

Beginning in BIG-IP DNS 13.0.0, the maximum number of available pool members that the system can return is 500.

The following GSLB pool types are affected by this change:

- A
- AAAA
- MX
- SRV
- NAPTR

You set the value on a pool-by-pool basis in the Maximum Answers Returned attribute. Large DNS responses will necessitate the switch to TCP, and could affect performance.

**Note** The number of pool members returned in a DNS response is the lower number between the maximum number chosen for the pool, and the number of available pool members.

Topology

BIG-IP DNS can make load balancing decisions based upon the geographical location of the LDNS making the DNS request. The location of the LDNS is determined from a GeoIP database. In order to use topology, the administrator must configure topology records describing how BIG-IP DNS should make its load balancing decisions. For more information on topology load balancing, refer to Using Topology Load Balancing to Distribute DNS Requests to Specific Resources in *BIG-IP Global Traffic Manager: Load Balancing* or AskF5 article K13412: Overview of BIG-IP DNS Topology records (11.x - 12.x).

Topology load balancing can be used to direct users to the servers that are geographically close, or perhaps to direct users to servers that have localized content.

BIG-IP system software provides a pre-populated database that provides a mapping of IP addresses to geographic locations. The administrator can also create custom group call regions. For example, it is possible to create a custom region that groups certain IP addresses together or that groups certain countries together.

Updates for the GeoIP database are provided on a regular basis. AskF5 article K11176: Downloading and installing updates to the IP geolocation database contains information about updating the database.

Topology records are used to map an LDNS address to a resource. The topology record contains three elements:

- A request source statement that specifies the origin LDNS of a DNS request.
- A destination statement that specifies the pool or pool member to which the weight of the topology record is assigned.
A weight that the BIG-IP system assigns to a pool or a pool member during the load balancing process.

When determining how to load balance a request, BIG-IP DNS uses the object that has the highest weight according the matching topology records.

**Important** When configuring topology load balancing at the Wide IP level, topology records with a pool destination statement must exist. Other destination statement types (such as data center or country) may be used when using topology as a pool level load balancing method.

### BIG-IP DNS Architectures

This section describes three common deployment methodologies for using a BIG-IP system in a DNS environment. For more information, refer to *BIG-IP Global Traffic Manager: Implementations*.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: [K12453464: Finding product documentation on AskF5](#).

#### Delegated mode

When operating in delegated mode, requests for Wide IP resource records are redirected or delegated to BIG-IP DNS. The BIG-IP system does not see all DNS requests, and operates on requests for records that are sent to it.

For more information, refer to *BIG-IP Global Traffic Manager: Implementations*.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: [K12453464: Finding product documentation on AskF5](#).

#### Screening mode

When operating in screening mode, BIG-IP DNS sits in front of one or more DNS servers. This configuration allows for easy implementation of additional BIG-IP system features for DNS traffic because DNS requests for records other than Wide IPs pass through BIG-IP DNS. If the request matches a Wide IP, BIG-IP DNS responds to the request. Otherwise, the request is forwarded to the DNS servers. This configuration can provide the following benefits:

- **DNS query validation:** When a request arrives at BIG-IP DNS, BIG-IP DNS validates that the query is well formed. BIG-IP DNS can drop malformed queries, protecting the back end DNS servers from seeing the malformed queries.

- **DNSSEC dynamic signing:** When the responses from the DNS server pass back through BIG-IP DNS, it is possible for BIG-IP DNS to sign the response. This allows the use of DNSSEC with an existing zone and DNS servers, but takes advantage of any cryptographic accelerators in a BIG-IP device.

- **Transparent Caching:** When the responses from the DNS server pass back through the BIG-IP system, it can cache the response. Future requests for the same records can be served directly from the BIG-IP system reducing the load on the back-end DNS servers.

For more information, refer to *BIG-IP Global Traffic Manager: Implementations*.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: [K12453464: Finding](#).
Replacing a DNS server

It is also possible for the BIG-IP system to operate as a stand-alone, authoritative DNS server for one or more zones. In this configuration, all DNS requests for a zone are sent to BIG-IP DNS. Any requests for a Wide IP are handled by BIG-IP DNS and other requests are sent to the local bind instance on the BIG-IP system. ZoneRunner is used to manage the records in the local-bind instance. For more information, refer to Replacing a DNS Server with BIG-IP DNS in **BIG-IP Global Traffic Manager: Implementations**.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: [K12453464: Finding product documentation on AskF5](#).

BIG-IP DNS iQuery

- iQuery® is an XML protocol that BIG-IP systems use to communicate with each other. BIG-IP DNS uses iQuery for various tasks:
  - Determining the health of objects in BIG-IP DNS configuration.
  - Exchanging information about BIG-IP DNS synchronization group state.
  - Providing a transport for synchronizing BIG-IP DNS configuration throughout the synchronization group.
  - Communicating LDNS path probing metrics.
  - Exchanging Wide IP persistence information.
  - Gathering BIG-IP system configuration when using auto-discovery.

All of these tasks combined provide a BIG-IP DNS synchronization group with a unified view of BIG-IP DNS configuration and state.

**iQuery Agents**

All BIG-IP DNS devices are iQuery clients. The gtmd process on each BIG-IP DNS device connects to the big3d process on every BIG-IP server defined in BIG-IP DNS configuration, which includes both BIG-IP DNS and BIG-IP LTM.

These are long-lived connections made using TCP port 4353. This set of connections among BIG-IP DNS devices and between BIG-IP DNS and BIG-IP LTM devices is called an iQuery mesh.

iQuery communication is encrypted using SSL. The devices involved in the communication authenticate each other using SSL certificate-based authentication. For information, refer to Communications Between BIG-IP DNS and Other Systems in **BIG-IP Global Traffic Manager: Concepts**.

**Note** For information about how to locate F5 product guides, refer to AskF5 article: [K12453464: Finding product documentation on AskF5](#).
sends monitor requests using iQuery to another iQuery server that is closer to the target of the monitor. All BIG-IP DNS devices in the synchronization group agrees on which BIG-IP DNS is responsible for initiating the monitoring request. The result of the monitoring request is sent by the iQuery server to all BIG-IP DNS devices connected to it that are participating in the synchronization group.

**Note** A lack of a unified view of the iQuery mesh causes unpredictable behavior. For example, if each BIG-IP DNS device is not connected to the same set of other BIG-IP DNS devices, there can be disagreement of monitor responsibility resulting in object availability flapping ("flapping" is when a device is marked down and up repeatedly).

---

**big3d software versioning**

The version of the big3d software installed on each device must be the same or later than the version of software used on BIG-IP DNS devices. For information on updating the big3d software, refer to **Running the big3d_install**.
Determining the health of the iQuery mesh

Reviewing log files or SNMP traps

The /var/log/gtm log file contains information about connection status changes to big3d agents. When a new connection is established to a big3d agent or when a connection is lost, a log message is generated.

Example connection lost messages:

```
alert gtmd[8663]: 011a500c:1: SNMP _ TRAP: Box 10.14.20.209 state change green --> red
             (Box 10.14.20.209 on Unavailable)
```

```
```

Example connection established messages:

```
alert gtmd[8663]: 011a500b:1: SNMP _ TRAP: Box 10.14.20.209 state change red --> green
```

```
```

If a connection to a configured BIG-IP server is down, repeated Connection in progress to messages is generated:

```
notice gtmd[8663]: 011ae020:5: Connection in progress to 10.14.20.209
```

```
notice gtmd[8663]: 011ae020:5: Connection in progress to 10.14.20.209
```

```
notice gtmd[8663]: 011ae020:5: Connection in progress to 10.14.20.209
```

tmsh

You can use tmsh show gtm iquery command to display that status of all of the iQuery connections on a BIG-IP DNS device. The command displays each IP address:

```
# tmsh show gtm iquery

---------------------------------------
Gtm::IQuery: 10.12.20.207
---------------------------------------
```

```
Server gtm-1
Data Center DC1
iQuery State connected
```
Query Reconnects                      1
Bits In                            8.2M
Bits Out                          47.7K
Backlogs                              0
Bytes Dropped                        96
Cert Expiration Date  10/29/24 04:38:53
Configuration Time    12/08/14 16:37:49
---------------------------------------
Gtm::IQuery: 10.14.20.209
---------------------------------------
Server                            gtm-3
Data Center                         DC2
iQuery State                  connected
Query Reconnects                      0
Bits In                            8.2M
Bits Out                          45.7K
Backlogs                              0
Bytes Dropped                        0
Cert Expiration Date  10/29/24 04:38:53
Configuration Time    12/08/14 16:37:49

For more information, refer to AskF5 article K13690: Troubleshooting BIG-IP DNS synchronization and iQuery connections (11.x - 13.x).

iqdump

You can use the iqdump command to check the communication path and SSL certificate-based authentication from a BIG-IP DNS to another device in the iquery mesh.

The syntax of the iqdump command is iqdump <ip address> <synchronization group name>. When using the iqdump command, BIG-IP DNS synchronization group name is optional.

For example:

# iqdump 10.14.20.209

<!-- Local hostname: gtm1.example.com -->

<!-- Connected to big3d at: ::ffff:10.14.20.209:4353 -->
Note The `iqdump` command continues to run until it is interrupted by pressing `Ctrl-C`.

If there is a problem with the communication path or the SSL authentication, the `iqdump` command fails and reports an error.

The version of BIG-IP software being run on the remote system is reported in the version XML stanza. The version of the big3d software running on the remote system is reported in the `<big3d>` XML stanza.


### BIG-IP DNS Device Service Clustering

In BIG-IP 13.0 and later, BIG-IP DNS includes full support for Device Service Clustering (DSC). In previous versions, devices configured in a DSC failover device group are not fully supported when more than two BIG-IP systems are included in the cluster.

For example, if a server object is configured as BIG-IP Redundant, virtual server auto-discovery does not function properly but instead added then deleted non-floating virtual services in your device group.

In BIG-IP 13.0 and later, BIG-IP DNS creates a server object representing any number of BIG-IP systems in the cluster. Virtual servers are tracked both by the server and the device; therefore a virtual server is removed from your device group only if it is first removed from all devices which interacted with it.

### BIG-IP DNS query logging

BIG-IP DNS can log debugging information about the decision making process when resolving a Wide IP. These logs can report which pool and virtual server were chosen for a Wide IP resolution or why BIG-IP DNS was unable to respond.

AskF5 article: [K14615: Configuring the BIG-IP DNS system to log Wide IP request information](https://community.f5.com/t5/AskF5-BIG-IP-System-Management/Configuring-the-BIG-IP-DNS-system-to-log-Wide-IP-request-information/ta-p/2051308) contains information about enabling query logging. This logging should only be enabled only for troubleshooting and not for long periods of time.

For DNS configurations that provide services beyond Wide IP resolution, for example DNS recursive resolvers or DNS Express, it is possible to enable DNS query and response logging. For more information, refer to [External Monitoring of BIG-IP Systems: Implementations](https://community.f5.com/t5/AskF5-Other-BIG-IP-System-Management/External-Monitoring-of-BIG-IP-Systems-Implementations/ta-p/2051308).
BIG-IP DNS Statistics

The BIG-IP system maintains statistics for objects throughout the system. Due to the variety of statistics gathered and the breadth of configuration elements covered, it is impractical to cover them within this guide. Statistics are documented throughout the user manuals where they are featured.

You can view statistics in the Configuration utility by going to Statistics > Module Statistics. In tmsh, statistics are visible using the show command with particular objects. Typically, statistics are either gauge or counters. A gauge keeps track of a current state, for example current connections. A counter keeps an incrementing count of how many times a particular action is taken, for example total requests.

Reviewing statistics that use counters may provide insight into the proportion of traffic which is valid, or perhaps may indicate that there is a configuration error.

For example, comparing the Dropped Queries counter to Total Queries counter shows that there are some drops, but this may not be of concern because the drops are fairly low as a percentage of the total.

```
# show gtm Wide IP www.example.com

Gtm::Wide IP: www.example.com

Status
Availability: available
State: enabled
Reason: Available

Requests
Total: 1765
A: 1552
AAAA: 213
Persisted: 0
Resolved: 1762
Dropped: 3
Load Balancing
Preferred: 1760
```
Statistics are also available for polling using SNMP and can be polled, cataloged over time, and graphed by a Network Management Station (NMS).
iRules

iRules® is a BIG-IP® feature which plays a critical role in advancing the flexibility of the BIG-IP system. iRules can be written to make load balancing decisions, persisting, redirecting, rewriting, discarding, and logging client sessions.

iRules can be used to augment or override default BIG-IP LTM® behavior, enhance security, optimize sites for better performance, provide robust capabilities necessary to guarantee application availability, and ensure a successful user experience on your sites.

iRules technology is implemented using Tool Command Language (Tcl). Tcl is known for speed, embeddability, and usability. iRules may be composed using most native Tcl commands, as well as a robust set of BIG-IP LTM/ BIG-IP DNS extensions provided by F5®.

Documentation that covers the root Tcl language can be found at the Tcl Developer Exchange (tcl.tk) and the Tcl Reference Manual (http://tmml.sourceforge.net).

Note These links take you to resources outside of AskF5™, and it is possible that the documents may be removed without our knowledge.

iRules anatomy

An individual iRule consists of one or more event declarations, each containing Tcl code that is executed when that event occurs.

Events

Events are an F5 extension to Tcl. They provide an event-driven framework for the execution of iRules code in the context of a connection flow. Events are triggered by internal traffic processing state changes. Some events are triggered for all connection flows, whilst others are profile-specific, meaning the event can only be triggered if an associated profile has been applied to the virtual server processing the flow in question.

For example, the CLIENT_ACCEPTED event is triggered for all flows when the flow is established. In the case of TCP connections, it is triggered when the three-way handshake is completed and the connection enters the ESTABLISHED state. In contrast, the HTTP_REQUEST event are only triggered on a given connection flow if an HTTP profile is applied to the virtual server. For example, the following iRules evaluates every HTTP request from a client against a list of known web robot user agents:

```
when HTTP_REQUEST {
switch -glob [string tolower [HTTP::header User-Agent]] {
    "*scooter*" -
    "*slurp*" -
    "*msnbot*" -
}
```
 Commands and functions

Commands and functions are responsible for the majority of the work within iRules. They allow you to do things like get the URI of an HTTP request (HTTP::uri), or encrypt data with an Advanced Encryption Standard (AES) key (AES::encrypt). In addition to the standard Tcl command set, F5 adds additional commands that are either global in scope (TCP::client_port, IP::addr, ...) or specific to a certain profile. For example, switch, HTTP::Header, and pool:

```
when HTTP_REQUEST {
    switch -glob [string tolower [HTTP::header User-Agent]] {
        "*scooter*" -
        "*slurp*" -
        "*msnbot*" -
        "*fast-*" -
        "*teoma*" -
        "*googlebot*" {
            # Send bots to the bot pool
            pool slow _ webbot _ pool

            }
    }
}
```

A full list of commands and functions can be found in the Command section of the iRules Wiki on DevCentral.

Note: Several native commands built into the Tcl language have been disabled in the iRules implementation due to relevance, and other complications that can cause an unexpected halt in the traffic flow (file IO, Socket calls, and others.). The list of disabled commands can be found in the iRulesWiki.
Operators

An operator is a token that forces evaluation and comparison of two conditions. In addition to the built in Tcl operators (==, <=, >=, ...), operators such as starts_with, contains, and ends_with have been added to act as helpers for common comparisons. For example, in the iRules below, the < operator compares the number of available pool members against 1, while the starts_with evaluates the requested URI against a known value:

```
when HTTP _ REQUEST {
  if { [active _ members [LB::server pool]] < 1 } {
    HTTP::respond 503 content {<html><body>Site is temporarily unavailable. Sorry!</body><html>}
    return
  }
  if { [HTTP::uri] starts _ with "/nothing-to-see-here/" } {
    HTTP::respond 403 content {<html><body>Too bad, so sad!</body><html>}
    return
  }
}
```

A list of the F5 specific operators added to Tcl for iRules are available on the iRules Wiki on DevCentral.

Variables

iRules uses two variable scopes: static and local. In addition, procedures may accept arguments which populate temporary variables scoped to the procedure.

Static variables are similar to Tcl's global variables and are available to all iRules on all flows. A static variable’s value is inherited by all flows using that iRules, and they are typically set only in RULE_INIT and read in other events. They are commonly used to toggle debugging or perform minor configuration such as the names of datagroups that are used for more complete configuration. Static variables are denoted by a static:: prefix. For example, a static variable may be named static::debug.

Local, or flow, variables are local to connection flows. Once set they are available to all iRules and iRules events on that flow. These are most of what iRules use, and can contain any data. Their value is discarded when the connection is closed. They are frequently initialized in the CLIENT_CONNECTED event.

It is very important to understand the variable scope in order to preclude unexpected conditions as some variables may be shared across pools and events and in some cases, globally to all connections. For more information, refer to iRues 101 - #03 - Variables on DevCentral.

iRules context

For every event that you specify within an iRules, you can also specify a context, denoted by the keywords client-side or server-side. Because each event has a default context associated with it, you need only declare a context if
you want to change the context from the default.

The following example includes the event declaration `CLIENT_ACCEPTED`, as well as the iRules command `IP::remote_addr`. In this case, the IP address that the iRules command returns is that of the client, because the default context of the event declaration `CLIENT_ACCEPTED` is client-side.

```plaintext
when CLIENT_ACCEPTED {
    if { [IP::addr [IP::remote_addr] equals 10.1.1.80] } {
        pool my_pool
    }
}
```

Similarly, if you include the event declaration `SERVER_CONNECTED` as well as the command `IP::remote_addr`, the IP address that the iRules command returns is that of the server, because the default context of the event declaration `SERVER_CONNECTED` is server-side.

```plaintext
when SERVER_CONNECTED {
    if { [IP::addr [IP::addr [clientside [IP::remote_addr]] equals 10.1.1.80] } {
        discard
    }
}
```

### iRules example

The following HTML Comment Scrubber example iRules is composed of several events that depend upon each other, as well as numerous operators and commands used to evaluate client supplied requests as well as the data contained within the server response.

```plaintext
when RULE_INIT {
    set static::debug 0
}
when HTTP_REQUEST {
```
# Don’t allow data to be chunked

if { [HTTP::version] eq "1.1" } {
    if { [HTTP::header is _ keepalive] } {
        HTTP::header replace "Connection" "Keep-Alive"
    }
    HTTP::version "1.0"
}

when HTTP _ RESPONSE {
    if { [HTTP::header exists "Content-Length"] && [HTTP::header "Content-Length"] < 1000000 } {
        set content _ length [HTTP::header "Content-Length"]
    } else {
        set content _ length 1000000
    }
    if { $content _ length > 0 } {
        HTTP::collect $content _ length
    }
}

when HTTP _ RESPONSE _ DATA {
    # Find the HTML comments
    set indices [regexp -all -inline -indices {<!--(?:[^-]|--|--[^-|-]|[^-]-[^-])*--\s*>} [HTTP::payload]]
    # Replace the comments with spaces in the response
    if { $static::debug } {
        log local0. "Indices: $indices"
    }
    foreach idx $indices {
        set start [lindex $idx 0]
        set len [expr {[lindex $idx 1] - $start + 1}]
        if { $static::debug } {
            log local0. "Start: $start, Len: $len"
        }
    }
}
iRules considerations

Performance implications

While iRules provide tremendous flexibility and capability, deploying them adds troubleshooting complexity and some amount of processing overhead on the BIG-IP system itself. As with any advanced functionality, it makes sense to weigh all the pros and cons.

Before using iRules, determine whether the same functionality can be performed by an existing BIG-IP feature. Many of the more popular iRules functionalities have been incorporated into the native BIG-IP feature set over time. Examples include cookie encryption and header insertion using the HTTP profile, URI load balancing using BIG-IP LTM policies and, in BIG-IP versions 11.3.x and lower, the HTTP class profile.

Even if requirements cannot be met by the existing capabilities of the BIG-IP system, it is still worth considering whether or not the goal can be accomplished through other means, such as updating the application itself. However, iRules are often the most time-efficient and best solution in situations when it may take months to get an update to an application. The negative impacts of using iRules, such as increased CPU load, can be measured and managed through using iRules tools and by writing iRules so they behave as efficiently as possible.

Programming considerations

The following practices may be useful when it comes to developing your iRules.

- Write readable code: While Tcl supports multiple commands in a single line, writing iRules in this way can make them difficult to read.
- Choose meaningful variable names.
- Write reusable code. If you perform certain operations frequently, consider creating a single iRules that is associated with multiple virtual servers. In BIG-IP 11.4 and later, you can also define and call procedures in iRules.
- Write efficient code. For example, avoid data type conversions unless necessary (from strings to numbers).
  - Use the return command to abort further execution of a rule if the conditions it was intended to evaluate have been matched.
  - Use timing commands to measure iRules before you put them into production.
- Use data classes (data groups) to store variable and configuration information outside of the iRules code itself. This allows you to make updates to data that needs to change frequently without the risk of introducing coding errors. It also helps make the iRules portable as the same code can be used for QA and
production while the data classes themselves can be specific to their respective environments.

- Use a stats profile instead of writing to a log to determine how many times an iRules has fired or a particular action has been taken.

iRules assignment to a virtual server

When you assign multiple iRules as resources for a virtual server, it is important to consider the order in which you list them on the virtual server. This is because BIG-IP LTM processes duplicate iRules events in the order in which the applicable iRules are listed. An iRules event can therefore terminate the triggering of events, thus preventing BIG-IP LTM from triggering subsequent events.

For more information, refer to AskF5 article K13033: Constructing CMP-compatible iRules.

iRules troubleshooting

This section covers a brief guide for iRules in relation to BIG-IP LTM and BIG-IP DNS™.

**Note** This guide assumes that the initial configuration of your BIG-IP system is complete and you are using iRules to implement additional functionality.

The tools identified within this chapter are not an exhaustive list of all possibilities, and while there may be other tools, these are intended to cover the basics.

Syntax

Tools that you may use to write and check that the syntax of your iRules is correct include:

- **iRules Editor**, integrated code editor for iRules, built by F5 to develop iRules with full syntax highlighting, colorization, textual auto-complete, integrated help, etc.

- **Notepad++**, a text and source code editor for use with Microsoft Windows.

Client/server debug tools

After the creation of iRules, the intended functionality needs to be verified. Depending on the expected functionality, you may need to check client, the BIG-IP system, and server response to the traffic and the iRules. Many browsers include extensions that assist with web debugging, such as FireBug for Mozilla Firefox and Google Chrome developer tools.

Additionally, some third-party tools that may facilitate debugging include:

- **HttpWatch**, an HTTP sniffer for Internet Explorer®, Firefox®, iPhone®, and iPad® that provides insights into the performance of your iRules.

- **Fiddler**, an HTTP debugging proxy server application.

- **Wireshark**, a free and open source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education.
DevCentral

DevCentral is a community-oriented resource to bring engineers together to learn, solve problems, and figure things out. iRules-specific resources on DevCentral™ include:

- iRules Reference Wiki
- Official documentation
- Tutorials
- Articles
- Q&A forum, examples (CodeShare)
- Podcasts (videos)

Considerations when writing iRules

When writing iRules, keep in mind the following considerations:

- iRules development should take place on a non-production system.
- Verify your iRules often while building it.
- While error messages may be cryptic, review them carefully as they often indicate where a problem exists.
- Log within the iRules liberally, both to watch variables and to determine iRules progression. Consider using or evaluating a variable to turn off unnecessary debug logging in production.
- Many iRules commands have software version requirements. Check DevCentral to verify the version requirements of the intended commands.
- Use the `catch` command to encapsulate commands which are likely to throw errors based on input and other factors. Instead of having an iRules fail and the connection subsequently dropped, use the `catch` command to perform an alternate action.
- Thoroughly test your iRules to ensure intended functionality. Even once an iRules compiles, it may have logical errors. BIG-IP LTM cannot detect logical errors, so while an iRules may compile and execute, it may behave in such a way that precludes the BIG-IP system from returning a valid response to the client.

Review your syntax

The F5 iRules editor helps you identify common issues, but not all mistakes are readily detected. A common cause of errors is improper use of braces.

**Tip** If you use opening brace, bracket, or parenthesis in an iRule, you must also use the corresponding closing character. The Configuration utility may return intimidating or cryptic-seeming error messages when attempting to run an iRule which contains incorrect usage of brackets, braces, or parentheses.

It is important to leave sufficient whitespace to make the code readable and manageable. iRules generates errors due to the extra data between the closing braces of the `if` and `elseif` and before the `elseif` and `else`. Syntax

104
validation can fail due to poorly placed comments.

Test your iRules

When debugging iRules it is important to capture the connection data from end to end.

Initialize the tools before you make your initial connection to the BIG-IP system, and do not stop gathering data until the connection has fully completed successfully, or failed.

Deploy your iRules

Consider versioning your iRules for simplicity of deployment and maintainability. It is easy to introduce new errors or bugs, and having the old iRules to fall back on greatly decreases your time to resolution if something goes wrong. Promoting an iRules from a test environment to a production environment goes a lot more smoothly if you do not need to make any code updates to the iRules itself when saving it.
Logging

Your BIG-IP® LTM® and BIG-IP DNS™ logs can be helpful in maintaining the stability and health of your systems. Events can be logged either locally or remotely depending on your configuration. Logging is covered extensively in the BIG-IP TMOS: Operations Guide. This chapter covers some important concepts as well as topics related to BIG-IP LTM and BIG-IP DNS.

Logging levels

For each type of system event you can set the minimum log level. When you specify a minimum log level, all alerts at the minimum level and higher are logged. The logging levels follow the typical Linux-standard levels.

Lowering the minimum log level increases the volume of logs that are produced. Use caution when setting the minimum log level to the lowest value (Debug) as this may result in large volumes of log messages which may can have a negative impact on the performance of the BIG-IP system.

Default local log levels are set on the BIG-IP system in order to convey important information. F5® recommends changing default log levels only when needed to assist with troubleshooting. When troubleshooting is complete, log levels should be reset to their default value.

For more detailed information logging, refer to the following resources:

- AskF5™ article: K13455: Overview of BIG-IP logging BigDB database keys (11.x - 12.x).
- AskF5 article: K5532: Configuring the level of information logged for TMM-specific events.

Note: For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.

Local logging

By default, the BIG-IP system logs using syslog to the local file system. Most local log files can be managed and viewed using the BIG-IP Configuration Utility.

Important log files for BIG-IP LTM include:

- /var/log/ltm
- /var/log/tmm
- /var/log/pktfilter

Important log files for BIG-IP DNS and DNS include:

- /var/log/dns
Remote Logging

Although you can configure the BIG-IP system to log locally, F5 recommends logging to a pool of high-speed remote logging servers. You can set up remote logging use the legacy Linux syslog-ng functionality or use the high-speed logging functions managed by TMOS.

Remote Logging using syslog-ng

Remote syslog servers can be added through BIG-IP configuration utility or through the TMOS Shell (tmsh). For more information, refer to AskF5 article: K13080: Configuring the BIG-IP system to log to a remote syslog server (10.x - 13.x).

The default log levels only apply to local logging, not remote logging. This means that when you define a remote syslog server, the BIG-IP system sends all syslog messages to the remote server. Remote logging filters can be customized through the Configuration utility and through tmsh.

Customizing remote logging using syslog-ng requires the use of tmsh. Customizing allows you to do the following:

- Log to a remote server using TCP
- Set logging levels
- Direct specific messages to specific destinations
- Define multiple remote logging servers

For more information, refer to AskF5 article: K13333: Filtering log messages sent to remote syslog servers (11.x - 13.x).

Remote logging using high-speed logging

In BIG-IP 11.3 and later, high-speed logging allows users to more easily filter and direct log messages produced by system processes.

The configuration process requires the definition of a pool of remote logging servers and an unformatted remote high-speed log destination that references the pool.

If ArcSight, Splunk, or Remote Syslog servers are used, formatted log destinations need to be defined. The log filter allows you to define your filtering criteria, such as severity, source, or system process that created the log message, and a message ID for limiting the logging to one specific instance of a specific message type.

Refer to External Monitoring of BIG-IP Systems: Implementations for information about configuring remote high-speed logging for several scenarios, including DNS Logging, Protocol Security Events, Network Firewall Events, DoS/DDoS Protection Events, and others.
Remote monitoring

You can use SNMP to manage and monitor BIG-IP LTM and BIG-IP DNS. SNMP is an Internet Protocol used for managing nodes on an IP network. Setting up SNMP traps on the BIG-IP system involves defining the trap destinations and the events that result when a trap is sent.

You can download Specific Management Information Bases (MIB) for BIG-IP LTM and BIG-IP DNS from the welcome page of the BIG-IP Configuration utility, or from the /usr/share/snmp/mibs directory on the BIG-IP file system.

The MIB for BIG-IP LTM object is in the F5-BIGIP-LOCAL-MIB.txt file.

The MIB for BIG-IP DNS objects is in the F5-BIGIP-GLOBAL-MIB.txt file.

Other F5 MIBs may be necessary. For more information, refer to AskF5 article: K13322: Overview of BIG-IP MIB files (10.x - 12x) and SNMP Trap Configuration in BIG-IP TMOS: Implementations.

Note In BIG-IP LTM 12.1 and later, CSR Support Standard MIBs catalog is expanded to support IETF standard MIBs.

Note For information about how to locate F5 product guides, refer to AskF5 article: K12453464: Finding product documentation on AskF5.
Logging considerations

When changing the minimum logging level, do so with great care to avoid generating large log files and impacting the system's performance.

Minimize the amount of time that the log level is set to **Debug**. When debug logging is no longer required, be sure to set log levels back to their default values.

The following AskF5 resources list several log events and their default values:

- **K5532**: Configuring the level of information logged for TMM-specific events.
- **K73208321**: Overview of BIG-IP logging BigDB database keys (13.x)

If more than one monitor is assigned to a pool member, you may need to determine which monitor triggered an event. For more information, refer to “Monitors”.

When creating, modifying and testing iRules®, use debug logging only in a testing environment.

- Before an iRules is promoted to production, remove statements typically used during the development and testing phases that write information to the logs for sanity checking.
- To determine how many times an iRules has fired or a particular action has been taken, use a stats profile instead of writing to a log.

Monitor BIG-IP LTM and BIG-IP DNS modules for the following log events:

- LTM/DNS: Pool members/nodes down or flapping
- LTM/DNS: Pool has no members
- LTM: Port exhaustion
- DNS: “Connection in progress” messages
Optimizing the Support Experience

F5 technical support commitment

F5®, striving to continuously improve its support service and create closer customer relationships, designed to provide assistance with specific break-fix issues and ongoing maintenance of F5 products, F5 professional support services are consistently high-quality.

This means:

- F5 network support engineers conduct themselves professionally at all times.
- F5 is committed to providing the best customer experience possible.
- F5 treats customers are with respect and give them every consideration possible.
- F5 aims to provide resolutions the first time, every time.
- You can ask for manager escalation for unresolved or “site down” issues.

Some technical support issues arise from configuration errors, either within the BIG-IP® system or with other devices in the network. In other cases, a misunderstanding of BIG-IP capabilities can lead to support questions and issues. Although F5 does everything possible to prevent defects in BIG-IP hardware and software, these issues may still arise periodically. Regardless of the root cause of a problem, the goal is to resolve any issues quickly.

F5 technical support offerings

A variety of technical support offerings are available to provide the right level of support for any organization.

F5 Standard and Premium Support include remote assistance from F5 network support engineers, both online and over the phone.

Premium Plus customers receive priority status at F5, with fast, easy access to a dedicated team of senior-level, F5-certified network support engineers and a Technical Account Manager.

To learn more, refer to F5 Technical Support Offerings or send email to services@f5.com.

Professional services

Take advantage of the full range of F5 Professional Services to help you design, customize, and implement a solution that is right for your IT infrastructure and which supports your business goals.

Professional Services (f5.com/support/professional-services) provides information on a wide range of F5 Professional Services offerings and Professional Services Partners. You can use our online forms to request Consulting Services OnDemand for custom, shorter scope consulting engagements, or iRules® OnDemand to get fast access to iRules scripts tailored to your specific needs.

You can make an online request for specific support services by filling out a request form:
OPTIMIZING THE SUPPORT EXPERIENCE—F5 CERTIFICATION

- [Consulting request form](https://www.f5.com/services/professional-services/request-f5-professional-services).

GUARDIAN Professional Services Partners

F5 GUARDIAN® Professional Services Partners are authorized as installation providers and are also available to assist you. F5 GUARDIANS are selected because they have the skills and experience required to ensure successful implementations of F5 BIG-IP installations.

Refer to [F5 GUARDIAN Professional Service Partners](f5.com/support/professional-services#guardian) for a complete list of partners.

F5 certification

F5 Certified® exams test the skills and knowledge necessary to be successful when working with today's application delivery challenges. Our technically relevant and appropriate exams deliver consistently reproducible results that guarantee excellence in those that achieve certification.

Certification levels

F5 Certified! is the F5 certification program, with a progressive program of four levels (Administrator, Specialist, Expert, and Professional), each of which build on the skills and knowledge demonstrated on previous exams.

C1 – F5 Certified BIG-IP Administrator (F5-CA)

The starting point for all certifications: a certified BIG-IP Administrator has basic network and application knowledge to be successful in application delivery.

C2 – F5 Certified Technology Specialists (F5-CTS)

The Technology Specialist certification assures employers that the candidate is fully qualified to design, implement, and maintain that specific product and its advanced features.

C3 – F5 Certified Solution Expert (F5-CSE)

The Solution Expert focuses on how F5 technologies combine with industry technology to create real-world business solutions.

C4 – F5 Certified Application Delivery Engineer (F5-CADE)

The Application Delivery Engineer certification exam and requirements are still under development.

C5 – F5 Certified Application Delivery Architect (F5-CADA)

The Application Delivery Architect certification exam and requirements are still under development.

Certificate expiration

F5 certifications are valid for two (2) years. Three months before the expiration date, the holder becomes recertification-eligible and can register for the exam necessary to re-certify. Only the last exam in the highest level
Certification achieved needs to be retaken.

**Certification beta program**

F5 uses beta exams in the creation of all our exams and to maintain their relevancy and accuracy after production. Beta exams are open to all and give candidates an opportunity to have an impact on the F5 Certified program. While beta exams are twice as long, they cost less than regular exams and give candidates the chance to leave feedback on the exam. Beta exams are critical to our exam development process and a great way to change the F5 Certified program for the better.

**Get involved**

There are a several ways to get involved with the F5 certification beta program:

- Beta participation. Interested in taking our beta exams? Contact us at F5Certification@f5.com to learn more.
- Exam development. Contact us at F5Certification@f5.com if you’re interested in helping us create our Certified exams.
- LinkedIn community. Join us on LinkedIn (https://www.linkedin.com/company/f5-networks) for answers to frequently asked questions, community developed resources, and more.

**Note:** This link takes you to a resource outside of F5, and it is possible that the document may be removed without our knowledge.

Visit F5 Credential Manager System (certification.f5.com) for information or follow the steps to get registered.

**Self-help**

F5 offers a number of resources to assist in managing and supporting your F5 systems:

- AskF5™ (support.f5.com)
- Downloads (downloads.f5.com) User name and password required.
- Security Updates (interact.f5.com/AskF5-SubscriptionCenter.html)
- BIG-IP iHealth® (f5.com/support/tools/ihealth)
- TechNews (interact.f5.com/AskF5-SubscriptionCenter.html)
- RSS feeds (https://support.f5.com/csp/article/K9957)
- DevCentral (devcentral.f5.com/) User name and password required.
- F5 Training Programs and Education (f5.com/education/training)
AskF5

AskF5 (support.f5.com) is a great resource for thousands of articles and other documents to help you manage your F5 products more effectively. Step-by-step instructions, downloads, and links to additional resources give you the means to solve known issues quickly and without delay, and to address potential issues before they become reality.

Whether you want to search the knowledge base to research an issue, or you need the most recent news on your F5 products, AskF5 is your source for product manuals, operations guides, and release notes, including the following:

- F5 announcements
- Known issues
- Security advisories
- Recommended practices
- Troubleshooting tips
- How-to documents
- Changes in behavior
- Diagnostic and firmware upgrades
- Hotfix information
- Product life cycle information

Downloads

Downloads are available from the F5 website. F5 strongly recommends that you keep your F5 software up-to-date, including hotfixes, security updates, OPSWAT updates, BIG-IP Application Security Manager™ (ASM®) signature files, and geolocation database updates. All software downloads are available from F5 Downloads (https://downloads.f5.com).

Security updates

You can receive timely security updates and BIG-IP ASM attack signature updates from F5. When remote vulnerabilities are discovered, F5 implements, tests, and releases security hotfixes for any vulnerable supported version, and sends an email alert to the F5 Security mailing list. F5 encourages customers with an active support account to subscribe to this list. For more information, refer to AskF5 article: K41942608: Overview of AskF5 security advisory articles.
BIG-IP iHealth

The BIG-IP iHealth® (iHealth.f5.com) diagnostic viewer is among the most important preventative tools to verify the proper operation of your BIG-IP system. It ensures hardware and software are functioning at peak efficiency and helps detect and address issues that may potentially affect F5 systems. BIG-IP iHealth is not integrated within the BIG-IP system. It is hosted by F5 and can be accessed with any web browser.

F5 recommends you generate a BIG-IP iHealth QKView file on the BIG-IP system and upload it to iHealth on a weekly basis in order to benefit from the many regularly occurring diagnostic updates. Uploading QKView files to iHealth also provides F5 technical support with access to your QKView files if you open a support case.

By reviewing the iHealth output, many of the issues commonly experienced by customers can be resolved without the need for opening a support case with F5.

TechNews

Communications Preference Center provides two email publications to help keep administrators up-to-date on various F5 updates and other offerings:

- **TechNews Weekly eNewsletter** Up-to-date information about product and hotfix releases, new and updated articles, and new feature notices.

- **TechNews Notifications** Do you want to get release information, but not a weekly eNewsletter? Sign up to get an HTML notification email any time F5 releases a product or hotfix.

- **Security Alerts** Receive timely security updates and ASM attack signature updates from F5.

AskF5 recent additions and updates

You can subscribe to F5 RSS feeds to stay informed about new documents pertaining to your installed products or products of interest. The Recent additions and updates page on AskF5 provides an overview of all the documents recently added to AskF5.

New and updated articles are published over RSS. You can configure feeds that pertain to specific products, product versions, and/or document sets. You can also aggregate multiple feeds into your RSS reader to display one unified list of all selected documents.

DevCentral

DevCentral™ (devcentral.f5.com) is an online forum of F5 employees and customers that provides technical documentation, discussion forums, blogs, media and more, related to application delivery networking. DevCentral is a resource for education and advice on F5 technologies and is especially helpful for iRules and iApps® developers. Access to DevCentral is free, but registration is required.
As a DevCentral member, you can do the following:

- Ask forum questions.
- Rate and comment on content.
- Contribute to “wikis.”
- Download lab projects.
- Join community interest groups.
- Solve problems and search for information.
- Attend online community events.
- View educational videos.

F5 training programs and education

F5 provides training programs and education, including traditional classroom learning opportunities, live online training, and free, self-paced online courses to help you get the most out of your investment. F5 Education (f5.com/education/training) provides links to course schedules, pricing, and registration details. It also has information about alternative training solutions such as virtual and web-based training for those who cannot attend training in person.

- **In-person courses**: F5 courses are available in multiple training facilities across five continents. Each one combines instructor presentations, classroom discussions, and interactive labs. The hands-on learning environment helps provide a fast track to accomplishing your goals.

- **Virtual instructor-led training**: Remote on-line courses mirror classroom training. Participants watch the remote instructors’ live lecture online, participate in discussions, and perform lab exercises using remote desktop control.

- **Free online training**: You can use the self-paced Getting Started series of free, web-based courses to learn how to deploy F5 solutions to address your most common application delivery problems.

Engage F5 Support

F5 Support is designed to provide support for specific break-fix issues for customers with active support contracts. For more information about F5 scope of support, refer to Support Policies.
F5 Support resources

F5 Support resources are available 24 hours a day, seven days a week, and are distributed around the world in multiple support centers. Live support is provided by our professional network support engineers. Hours of availability may vary depending on the service contract with F5.

Contact numbers

Standard, Premium, and Premium Plus Support customers can open and manage cases by calling one of the contact numbers listed below.

North America

North America: 1-888-882-7535 or (206) 272-6500
Traffix® Support Only: 1-855-849-5673 or (206) 272-5774

Outside North America

Outside North America, Universal Toll-Free: +800 11 ASK 4 F5 or (800 11275 435)

Additional contact numbers by country

Australia: 1800 784 977
China: 010 5923 4123
Egypt: 0800-000-0537
Greece: 00-800-11275435
Hong Kong: 001-800-11275435
India: 000-800-650-1448; 000-800-650-0356 (Bharti Air users)
Indonesia: 001-803-657-904
Israel: 972-37630516
Japan: 81-3-5114-3260 or 0066-33-812670
Malaysia: 1-800-814994
New Zealand: 0800-44-9151
Philippines: 1-800-1-114-2564
Saudi Arabia: 800-844-7835
Singapore: 6411-1800
South Africa: 080-09-88889
OPTIMIZING THE SUPPORT EXPERIENCE—ENGAGE F5 SUPPORT

South Korea: 002-800-11275435
Taiwan: 00-800-11275435
Thailand: 001-800-12-0666763
United Arab Emirates: 8000-3570-2437
United Kingdom: 44-(0)8707-744-655
Vietnam: 120-11585

Open a support case

F5 provides several resources to help find solutions to problems. Before opening a support case with F5 technical support, check to see if the issue you are encountering is already documented.

The following is a list of resources to consult before opening a support case with F5:

- Deployment guides and white papers provide information about specific deployment configurations.
- AskF5 provides many articles including known issues, how-to guides, security issues, release notes, and general information about products. Many of the issues customers encounter are already documented on this site.
- BIG-IP iHealth enables customers to upload QKView files in order to verify operation of any BIG-IP system.

Gather information to open a support case

If your issue cannot be solved using the resources listed, and you need to open a support case, you must first gather several pieces of important information about your issue. Providing full and accurate information helps speed the path to resolution. The required information for the majority of situations is summarized below:

- The serial number or base registration key of the specific BIG-IP system requiring support. For more information, refer to AskF5 article: K917: Finding the serial number or registration key of your F5 device.
- A full description of the issue. A clear problem statement is the best tool in helping to troubleshoot issues. Your description should include as much of the following information as you can provide.
- Occurrences and changes: The date and times of initial and subsequent recurrences. Did this issue arise at implementation or later? Were there any changes or updates made to the BIG-IP system prior to the issue arising? If so, what were they?
- Symptoms: Ensuring your list of symptoms is as detailed as possible gives more information for support personnel to correlate with.
- Scope of the problem: Note whether the problem is system-wide or limited to a particular configuration feature, service, or element (such as VLAN, interface, application service, virtual server, pool, and so on).
- BIG-IP component: The feature, configuration element, or service being used when the problem occurred (for example: portal access, network access, authentication services, VDI, Exchange).
• Steps to reproduce: The steps to reproduce the problem as accurately and in as much detail as possible. Include expected behavior (what should happen) as well as actual behavior (what does happen).

• Errors: Complete text of any error messages produced.

• Environment: Current usage of the system. (Is this unit in production? If so, is there currently a workaround in place?)

• Browsers: Types and versions, if applicable.

• Changes: System changes made immediately prior to the problem’s first occurrence. This may include upgrades, hardware changes, network maintenance, and so on. Have any changes been made to resolve the problem? If so, what were they?

• Issue Severity: A description of the impact the issue is having on your site or case severity
  
  • Severity 1: Software or hardware conditions on your F5 device are preventing the execution of critical business activities. The device does not power up or is not passing traffic.
  
  • Severity 2: Software or hardware conditions on your F5 device are preventing or significantly impairing high-level commerce or business activities.
  
  • Severity 3: Software or hardware conditions on your F5 device are creating degradation of service or functionality in normal business or commerce activities.
  
  • Severity 4: Questions regarding configurations (“how to”), troubleshooting non-critical issues, or requests for product functionality that are not part of the current product feature set.

• Contact and availability information including alternate contacts authorized to work on the problem with F5 Support. When there are more personnel available to work with F5 Support, the resolution of your issue may be expedited.

• Remote access information, if possible.

• A QKView file obtained while problem symptoms are manifesting. A QKView of the system before the occurrence is also useful. F5 recommends archiving QKView files regularly. For more information, refer to BIG-IP iHealth in the **TMOS Operations Guide**.

**Note** For information about how to locate F5 product manuals, refer to AskF5 article: [K12453464: Finding product documentation on AskF5](#).

• Product-specific information: Software versions and types of equipment in use.

• Platform and system. Version and provisioned software modules of the affected system.

To locate platform and system information using **tmsh** at the command line

• Type the following command:

  ```
  tmsh show /sys hardware
  ```

  Output appears similar to the following example:
<SNIP some of the output>

**Platform**

Name: BIG-IP 3900

**BIOS Revision**

F5 Platform: C106 OBJ-0314-03 BIOS (build: 010) Date: 02/15/12

**Base MAC**

00:01:d7:be:bf:80

**System Information**

Type: C106

Chassis Serial: f5-jspv-lzxw

Level 200/400 Part: 200-0322-02 REV C

To copy software version and build number information at the command line

1. Type the following command:

   cat /VERSION

   Output appears similar to the following example:

   **Product:** BIG-IP
   **Version:** 11.6.0
   **Build:** 0.0.401
   **Sequence:** 11.6.0.0.0.401.0
   **BaseBuild:** 0.0.401
   **Edition:** Final
   **Date:** Mon Aug 11 21:08:03 PDT 2014
   **Built:** 140811210803
   **Changelist:** 1255500
   **JobID:** 386543

2. Highlight and copy the output information and include it with your support case.
To copy provisioned module information at the command line

1. Type the following command:

```
tmsh list /sys provision
```

Output appears similar to the following example:

```
sys provision afm { }
sys provision am { }
sys provision apm {
  level nominal
}
sys provision asm { }
sys provision avr { }
sys provision fps { }
sys provision gtm { }
sys provision lc { }
sys provision ltm {
  level minimum
}
sys provision pem { }
sys provision swg { }
```

2. Highlight and copy the output information and include it with your support case.

Open a support case

If you cannot find the answer to your problem using the resources listed above, you can open a support case online, using F5 Support (f5.com/support).

Before you open a support case, you need to log in to F5. If you do not have an F5 login, you’ll need to register for one.

To register for support access


2. Click Register for an F5 Support Account.

3. Enter your email address.
4. Enter your contact information. If you have a support contract, click **I have a support contract and need access to MySupport**.

5. Enter your serial number or registration key in the **Serial Number or Registration Key (optional)** field.

After you’ve submitted your information, your service contract is reviewed. If your information is accurate you receive an email from MySupport, and you can use this to open your case.

### Send information to Support

After you have the information listed in “Gather information to open a support case”, transfer it to F5 technical support following the steps in “Share diagnostic files with F5 technical support”. For more information, refer to AskF5 article: [K2486: Providing files to F5 Technical Support](#).

### Share diagnostic files with F5 technical support

You can provide files to F5 Support using BIG-IP iHealth or Support Files, the F5 file transfer tool. Support Files complies with global data protection standards to safeguard the data you send.

### Prerequisites

Two categories of customers provide files to F5 Technical Support:

- Permanent account holders—Customers who have an F5 support account, including a user email and password for the AskF5 website
- Temporary users—Associates of permanent account holders who assist them with uploading or downloading files for an F5 service request

### Obtaining a Support Files tool user name and password for a temporary user

If you are a permanent account holder who wants a temporary user to upload or download files for one of your service requests, you must provide them with a user name and password for logging in to the Support Files website.

The user name is the case ID from your service request, and the password is in the activity notes of your service request. You can also request the password directly from F5 Technical Support via email. Temporary passwords cannot be provided over the phone.

**Note** Temporary user credentials are only active for a specific service request.

To locate a password for a temporary user in the activity notes of your service request

1. Open the **Service Request Details** view.
2. In the **Activities** section, locate the temporary password which should appear in the following format:
Uploading QKView files to BIG-IP iHealth

BIG-IP iHealth allows you to quickly diagnose the health and proper operation of your BIG-IP system, and provides a convenient location for you to send diagnostic data for case resolution with F5 Technical Support.

If you are running BIG-IP 10.x or later and need to provide a QKView file to F5, the preferred way to do so is to upload the file to the BIG-IP iHealth website. For more information, refer to K12878: Generating diagnostic data using the qkview utility.

Uploading and downloading files using support files

Accessing support files

To access Support Files as a permanent account holder

1. In the upper-right corner of the AskF5 home page, click My Support.
   If you have not already logged in, you are prompted to do so.

2. Under Service Requests, click the service request for which you want to upload or download files.

3. On the right side of the Service Request Details section, click Manage Attachments.
   The F5 Support Files site opens in a new window.

To access Support Files as a temporary user

- Navigate to the Support Files website and log in using the user name and password provided by a permanent account holder.

Uploading files using a web browser

1. On the Home folder page, click the folder for the service request you want.

2. Click Incoming (upload to F5).

3. In the upper-right corner of the Upload to F5 page, click the Upload files icon.

4. In the Upload files dialog box, click Browse.

5. In the Open dialog box, navigate to each file you want to upload, point at it with your cursor, select the check box that displays, and when you are finished selecting files, click Open.

   Either a success or failure notification displays. When an upload fails, close the notification and try
Again.

6. In the Upload Files dialog box, click Done.

**Note** You cannot see the files in the folder after uploading them.

### Downloading files using a web browser

1. On the Home folder page, click the folder for the service request you want.
2. Click Outgoing (download from F5).
3. Select the check box next to the files you want to download.
4. In the upper-right corner of the Download from F5 page, click the Download selected items icon.
5. Depending on which browser you have, you may be prompted to save your files to the location of your choice, or they may simply download to your Downloads folder.

### Uploading files using SFTP

For permanent account holders, your user name and password are the same as your AskF5 credentials. For temporary users, your user name is the service request number, and your password must be the correct passphrase for that service request.

**Important** Support Files does not support the Secure Copy (SCP) protocol.

**Note** Support Files supports the SFTP protocol, but only a subset of features provided by many SFTP clients. The SFTP server does not support or allow setting file ownership or permissions, updating timestamps, or creating symlinks.

**Note** The supportfiles.f5.com RSA server key MD5 fingerprint is MD5: 04:a6:4b:9b:d4:eb:48:97:15:e6:7f:90:64:bf:35:96 and the SHA256 fingerprint is SHA256:stQCq50hEwDPfRMeRf/Ya9dXcm1KCdx5i5iOwODgNU.

1. From the F5 device, SFTP to the supportfiles.f5.com site using the following syntax:

   ```sftp user@host```

   For example:

   ```sftp cl23456@supportfiles.f5.com```

   or

   ```sftp 1-12345678@supportfiles.f5.com```
OPTIMIZING THE SUPPORT EXPERIENCE—ENGAGE F5 SUPPORT

**Note** On the first attempt to connect, you must accept the host key. You should compare that output with the fingerprints listed in this article.

2. When prompted for the password, enter the email address of the user associated with the case.

3. To upload the requested files, use the following command syntax for each file:

```plaintext
put <name_of_file> <SR_number>/INCOMING/
```

For example:

```plaintext
put mybigip.conf C123456/INCOMING/
```

4. To exit the SFTP utility when all files have been uploaded, type the following command:

```plaintext
quit
```

For more information, refer to the manual or man pages for the SFTP utility by typing `man sftp` at the command line.

You may also use external SFTP applications to upload files if they are on a workstation or other system.

---

**Downloading files using SFTP**

To download a file from Support Files, you must use the exact file name and location (path) provided by your F5 Technical Support representative.

**Important** Support Files does not support the SCP protocol.

**Note** Support Files supports the SFTP protocol, but only a subset of features provided by many SFTP clients. The SFTP server does not support or allow setting file ownership or permissions, updating timestamps, or creating symlinks.

1. From the F5 device, SFTP to the [supportfiles.f5.com](http://supportfiles.f5.com) site using the following syntax:

```plaintext
sftp user@host
```

For example:

```plaintext
sftp C123456@supportfiles.f5.com
```

or

```plaintext
sftp 1-12345678@supportfiles.f5.com
```

2. When prompted for the password, enter the password.

3. To list the files available for download, type the following command:

```plaintext
ls C123456/OUTGOING
```
4. To download the requested files, use the following command syntax for each file:

**Note** The `<full_path_to_file>` section indicates the full path to the file.

```
get <full_path_to_file>
```

**For example:**
```
get C123456/OUTGOING/ Hotfix-BIGIP-12.1.0.0.40.1434-ENG.iso
```

5. To exit the SFTP utility when all files have been downloaded, type the following command:

```
quit
```

You may also use external SFTP applications to download files if the files are on a workstation or on another system.
Legal Notices

Trademarks

AAM, Access Policy Manager, Advanced Client Authentication, Advanced Firewall Manager, Advanced Routing, AFM, APM, Application Acceleration Manager, Application Security Manager, Applications without Constraints, ARX, AskF5, ASM, BIG-IP, BIG-IP EDGE GATEWAY, BIG-IQ, BIG-IP iControl, Cloud Extender, CloudFucious, Cloud Manager, Clustered Multiprocessing, CMP, COHESION, Data Manager, DDoS Frontline, DDoS Hybrid Defender, DDoS SWAT, Defense.net, DevCentral, DevCentral [DESIGN], DNS Express, DSC, DSI, Edge Client, Edge Gateway, EDGE MOBILE, EDGE MOBILITY, EdgePortal, ELEVATE, EM, Enterprise Manager, ENGAGE, F5, F5 Agility, F5 iApps, F5[DESIGN], F5 Certified [DESIGN], F5 iControl, F5 LINK CONTROLLER, F5 Networks, F5SalesXchange [DESIGN], F5Synthesis, f5Synthesis, F5Synthesis[DESIGN], F5 TechXchange [DESIGN], F5 TMOS, Fast Application Proxy, Fast Cache, FCINCO, Global Traffic Manager, GTM, GUARDIAN, Herculon, iApps, IBR, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iCall, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSeries, iSession, L7 RateShaping, LC, Link Controller, Local Traffic Manager, LTM, LineRate Operating System, LineRate Point, LineRate Precision, LineRate Systems [DESIGN], LROS, LTM, Message Security Manager, MSM, OneConnect, Packet Velocity, PEM, Policy Enforcement Manager, Protocol Security Manager, PSM, Real Traffic Policy Builder, Ready Defense, SalesXchange, ScaleN, Signalling Delivery Controller, Silverline, Silverline Threat Intelligence, SDC, SSL Acceleration, SSL Everywhere, SSL Orchestration, SDAC (except in Japan), StrongBox, SuperVIP, SYN Check, SYNTHESIS, TCP Express, TDR, TechXchange, TMOS, TotALL, Traffic Management Operating System, TrafficX Systems, Traffic Systems [DESIGN], Transparent Data Reduction, UNITY, VAULT, vCMP, VE F5 [DESIGN], Versafe, Versafe [DESIGN], VIPRION, Virtual Clustered Multiprocessing, WAF Express, WebSafe, We Make Apps Go [DESIGN], We Make Apps GO, and ZoneRunner, are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries, and may not be used without express written consent.

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. See the F5 Patents page (https://www.f5.com/about/guidelines-policies/patents).

Notice

THE SOFTWARE, SCRIPTING, AND COMMAND EXAMPLES ARE PROVIDED “AS IS,” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE, SCRIPTING AND COMMAND EXAMPLES, OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE, SCRIPTING, AND COMMAND EXAMPLES.
Publication Date

This document was published in December 2018.

Copyright

Copyright © 2013-2018, F5 Networks®, Inc. All rights reserved.

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.
## Change List

<table>
<thead>
<tr>
<th>Date</th>
<th>Chapter/Section</th>
<th>Change</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2015</td>
<td>BIG-IP LTM Load Balancing/</td>
<td>Correction to description of Figure 2.3: Load balancing.</td>
<td>Error</td>
</tr>
<tr>
<td></td>
<td>BIG-IP LTM Load Balancing Methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>August 2015</td>
<td>All</td>
<td>Updates to formatting</td>
<td>New Operations Guide style.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Addition of surveys</td>
<td></td>
</tr>
<tr>
<td>November 2015</td>
<td>All</td>
<td>Revision for style</td>
<td>BIG-IP 12.0 release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updates for BIG-IP 12.0</td>
<td></td>
</tr>
<tr>
<td>June 2017</td>
<td>All</td>
<td>Updates for BIG-IP 13.0</td>
<td>BIG-IP 13.0 release</td>
</tr>
<tr>
<td>August 2017</td>
<td>Optimizing the Support Experience</td>
<td>Removed reference to F5 Dropbox</td>
<td>Dropbox use discontinued</td>
</tr>
<tr>
<td>September 2017</td>
<td>Optimize the Support Experience</td>
<td>Added section on providing files to F5 Technical Support using F5 Support Files</td>
<td>New</td>
</tr>
<tr>
<td></td>
<td>BIG-IP DNS/ DNS Services/DNS cache</td>
<td>Updated text and Note re DNS cache and DNS Express</td>
<td>Error</td>
</tr>
<tr>
<td>April 2018</td>
<td>BIG-IP LTM Load Balancing/Monitors</td>
<td>Add new section on testing monitors before assigning them to resource.</td>
<td>BIG-IP 13.1.x release</td>
</tr>
<tr>
<td></td>
<td>BIG-IP LTM Network Address Objects/Service Provider Protocols</td>
<td>Added section on SIP Wizard.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BIG-IP LTM Virtual Servers/Virtual Server Troubleshooting</td>
<td>Added section on using <code>tcpdump</code> remotely.</td>
<td></td>
</tr>
<tr>
<td>December 2018</td>
<td>All</td>
<td>Updates to formatting and links</td>
<td>BIG-IP 14.x release</td>
</tr>
</tbody>
</table>