Configuration Guide for the VIPRION® System

version 10.1

MAN-0312-00

Product Version

This manual applies to the VIPRION® hardware platform and BIG-IP® software created by F5 Networks, Inc.

Publication Date

This manual was published on August 20, 2010.

Legal Notices

Acknowledgments

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, "This software" refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to "NetBSD," "FreeBSD," "Mach" (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at http://www.perl.com.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

Table of Contents

1

Introducing the VIPRION System

- Overview of the VIPRION system
- Configuration summary
- About this guide
- Finding help and technical support resources

Overview of the VIPRION system

The VIPRION® system is a complete traffic management solution that offers high performance, reliability, scalability, and ease of management. Based on chassis and blade technology, this system is designed to meet the needs of large, enterprise networking environments that normally require multiple BIG-IP systems to process large volumes of application traffic.

The VIPRION system consists of a chassis with a four-blade capacity. The four blades work together as one powerful system to process application traffic. For traffic coming into a single virtual server, the system distributes that traffic over multiple blades, utilizing the full multi-processing capacity of each blade. Moreover, if a blade unexpectedly becomes unavailable, another blade can complete the processing of the request.

Figure 1.1 shows the VIPRION system with a four-slot chassis.

Figure 1.1 *A four-slot chassis with four blades installed*

Specifically, the VIPRION system includes the following features:

- **A chassis with blades**
  The multi-slot chassis significantly reduces the amount of rack space required for the BIG-IP systems by housing blades instead of traditional switch systems. Hardware resources such as cooling and power systems, normally required for individual BIG-IP systems, are now part of the chassis instead.
- **Cluster technology**
  The VIPRION system's cluster technology means that all blades in the cluster function as one high-performance VIPRION system. A ***cluster*** is a group of slots in the VIPRION system chassis. Each slot in the cluster represents a ***cluster member***, and any blades that you insert into the slots of a cluster work together as a single VIPRION system to process

application traffic. With cluster technology, you utilize the power of multiple blades, but manage the entire cluster as if it were a single system.

- **Live installation**
  When you upgrade the BIG-IP software on a running system, the system automatically upgrades the BIG-IP software on all blades in the cluster.
- **Configuration synchronization**
  The primary blade automatically propagates the system configuration to all secondary blades, even when a new blade is introduced into the cluster.
- **Connection mirroring**
  When you create a virtual server for the cluster, you can enable connection mirroring. ***Connection mirroring*** ensures that if a blade, or a cluster within a redundant system configuration, becomes unavailable, the system can still process any existing connections.

Configuration summary

Before configuring a VIPRION system, it is helpful to have a brief overview of cluster configuration and the available configuration tools.

Overview of cluster configuration

The VIPRION system contains a four-slot chassis. Each system is configured with a default cluster named **default**. This default cluster includes the four slots as its cluster members.

One of the first configuration tasks that you perform is to insert the blades and then assign a unique cluster IP address to the default cluster. When you subsequently use this IP address to log on to the system, you can access the Configuration utility to continue configuring the system. For example, as with a non-clustered system, you can configure features such as trunks, VLANs, administrative partitions, and virtual servers. If you have a redundant system configuration, you can configure failover IP addresses, as well as connection mirroring between clusters.

Once you have completed the configuration, all blades in the cluster function as a single, powerful VIPRION system.

Choosing a configuration tool

To configure and maintain the VIPRION system, you can use either the browser-based Configuration utility or one of the command line interfaces (**tmsh** or **bigpipe**).

Using the Configuration utility, you can also monitor current system performance and view a network map that shows the virtual servers that you have created, along with the pools (and pool members) that the virtual servers reference.

For information on setting user preferences for the Configuration utility, see the ***TMOS® Management Guide for BIG-IP Systems***. For information on supported browsers, see the applicable release notes on the Ask F5SM Knowledge Base web site, **https://support.F5.com**.

About this guide

Before you use this guide, F5 Networks recommends that you read the guide ***Setting Up the VIPRION® Platform***.

Before you continue with adjusting or customizing the VIPRION system configuration, complete these tasks:

- Choose a configuration tool. For more information, see *Choosing a configuration tool* on this page.
- Familiarize yourself with additional resources such as product guides and online help. For more information, see *Additional information*, following.
- Review the stylistic conventions that appear in this chapter. For more information, see *Stylistic conventions*, on page 1-5.

Additional information

In addition to this guide, there are other sources of the documentation you can use in order to work with the VIPRION system and platform hardware. The following guides are available in PDF format from the Ask F5^{SM} Knowledge Base web site, **http://support.f5.com**. These guides are also available from the first web page you see when you log on to the administrative web server on the VIPRION system. The guides and their descriptions are:

- ***Platform Guide: VIPRION®***
  This guide includes comprehensive hardware information about the VIPRION platform. The guide also contains important environmental warnings.
- ***Setting Up the VIPRION$^{q(}$ Platform***
  This guide contains all information required for initially installing and configuring the platform hardware. Topics include removing the system from its packaging, installing the rackmount kit, inserting blades, and licensing the VIPRION system software.
- ***BIG-IP® Systems: Getting Started Guide***
  This guide contains detailed information about installing upgrades to the BIG-IP system. It also contains information about licensing and provisioning the BIG-IP system software, and connecting the system to a management workstation or network.
- ***TMOS® Management Guide for BIG-IP® Systems***
  This guide contains any information you need to configure and maintain the network and system-related components of the BIG-IP system. With this guide, you can perform tasks such as configuring VLANs, assigning self IP addresses, creating administrative user accounts, and managing a redundant system.
- ***Configuration Guide for BIG-IP® Local Traffic Manager™***
  This guide contains any information you need for configuring the BIG-IP system to manage local network traffic. With this guide, you can perform tasks such as creating virtual servers and load balancing pools, configuring application and persistence profiles, implementing health monitors, and setting up remote authentication.
- ***BIG-IP® Local Traffic Manager™: Implementations***
  This guide contains complete procedures for implementing specific goals, such as processing SSL traffic with data compresson, or assigning privileges to remotely-authenticated user accounts. This guide ties together the detailed information contained in the ***Configuration Guide for BIG-IP® Local Traffic Manager™*** and the ***TMOS® Management Guide for BIG-IP® Systems*** to help you implement specific traffic-management configurations.

- ***Bigpipe Utility Reference Guide***
  This guide contains syntax information for using the **bigpipe** utility command line interface.
- ***Traffic Management Shell (tmsh) Reference Guide***
  This guide contains syntax information for using for the **tmsh** command line interface.

Stylistic conventions

To help you easily identify and understand important information, all of our documentation uses the stylistic conventions described here.

Using the examples

All examples in this document use only private class IP addresses. When you set up the configurations we describe, you must use valid IP addresses suitable to your own network in place of our sample addresses.

Identifying new terms

To help you identify sections where a term is defined, the term itself is shown in bold italic text. For example, a ***floating IP address*** is an IP address assigned to a VLAN and shared between two computer systems.

Identifying references to objects, names, and commands

We apply bold text to a variety of items to help you easily pick them out of a block of text. These items include web addresses, IP addresses, utility names, and portions of commands, such as variables and keywords.

For example, with the **bigpipe self <ip_address> show** command, you can specify a specific self IP address to show by specifying an IP address for the **<ip_address>** variable.

Identifying references to other documents

We use italic text to denote a reference to another document or section of a document. We use bold, italic text to denote a reference to a book title. For example, for installation instructions, see the guide titled ***Configuration Guide for BIG-IP® Local Traffic Manager™***.

Identifying command syntax

We show complete commands in bold Courier text. Note that we do not include the corresponding screen prompt, unless the command is shown in a figure that depicts an entire command line screen. For example, the following command shows the configuration of the specified pool name:

```
bigpipe self <ip_address> show
```

or

```
b self <ip_Address> show
```

Table 1.1 explains additional special conventions used in command line syntax.

Item in text	Description
\	Indicates that the command continues on the following line, and that users should type the entire command without typing a line break.
< >	Identifies a user-defined parameter. For example, if the command has **<your name>**, type in your name, but do not include the brackets.
|	Separates parts of a command.
[]	Indicates that syntax inside the brackets is optional.
...	Indicates that you can type a series of items.

Table 1.1 *Command line syntax conventions*

Finding help and technical support resources

You can find additional technical documentation and product information in the following locations:

- **Online help for local traffic management**
  The Configuration utility has online help for each screen. The online help contains descriptions of each control and setting on the screen. Click the Help tab in the left navigation pane to view the online help for a screen.
- **Welcome screen in the Configuration utility**
  The Welcome screen in the Configuration utility contains links to many useful web sites and resources, including:
  - The F5 Networks Technical Support web site
  - The F5 Solution Center
  - The F5 DevCentral web site
  - Plug-ins, SNMP MIBs, and SSH clients
- **F5 Networks Technical Support web site**
  The F5 Networks Technical Support web site, **https://support.f5.com**, provides the latest documentation for the product, including:
  - Release notes for the VIPRION system, current and past
  - Updates for guides (in PDF form)
  - Technical notes
  - Answers to frequently asked questions
  - The Ask F5^{SM} Knowledge Base

To access this site, you need to register at **https://support.f5.com**.

2

Performing Initial System Configuration

- Introducing initial system configuration
- Setting up the system to process application traffic
- Setting up VLANs for redundancy

Introducing initial system configuration

If you have followed the instructions in ***Setting Up the VIPRION® Platform***, you should now have the following items:

- A chassis with all blades installed
- A license for the BIG-IP® system software
- A cluster named **default** that contains four cluster members (that is, slots)
- A cluster IP address assigned to the cluster

With this setup, you are now ready to perform additional configuration tasks so that the VIPRION® system can process application traffic, and if you have a redundant system configuration, remain available in the event of failover. These configuration tasks consist of creating various BIG-IP system objects on the system. Specifically, the objects you create are:

- Trunks, VLANs, and self IP addresses for application traffic
- A load balancing pool with pool members
- One or more health monitors
- A virtual server
- Trunks, VLANs, and self IP addresses for failover

The VIPRION system creates these configuration objects on the primary blade, and then the cluster synchronization process automatically propagates the configuration objects to those other blades.

If you have implemented a redundant system configuration, the standby cluster is capable of processing traffic in the event that the active cluster becomes unavailable.

The VIPRION system provides the flexibility you need to deploy the system in almost any physical or logical networking environment. For simplicity, however, the remainder of this chapter contains the procedures for implementing the most common configuration:

- The first procedure involves configuring BIG-IP system objects on the cluster, for processing application traffic. If you have a redundant system configuration, you must perform these configuration tasks on both units of the configuration. For more information, see *Setting up the system to process application traffic*, on page 2-2.
- The second procedure pertains to redundant system configurations only, specifically with respect to configuring an additional VLAN to use for failover between clusters. Again, if you have a redundant system configuration, you must perform this configuration task on both units of the redundant system configuration. For more information, see *Setting up VLANs for redundancy*, on page 2-5.

Setting up the system to process application traffic

To process application traffic, you must first configure some BIG-IP system network objects, such as VLANs, self IP addresses, and trunks, and then some local traffic objects such as a virtual server and a load balancing pool.

Important

F5 Networks recommends that you assign a management IP address to each cluster member (slot), and that you cable the management interface on each blade to the management network. This ensures that whenever the blade with the primary designation changes, you can still access the cluster using the cluster IP address. For information on assigning management IP addresses to cluster members, see Chapter 3, ***Managing Clusters****.*

Configuring trunks, VLANs, and self IP addresses

The first set of tasks consists of configuring some BIG-IP system objects, namely an external and an internal trunk, an external and an internal VLAN, and a self IP address for each VLAN.

Figure 2.1 shows the resulting configuration.

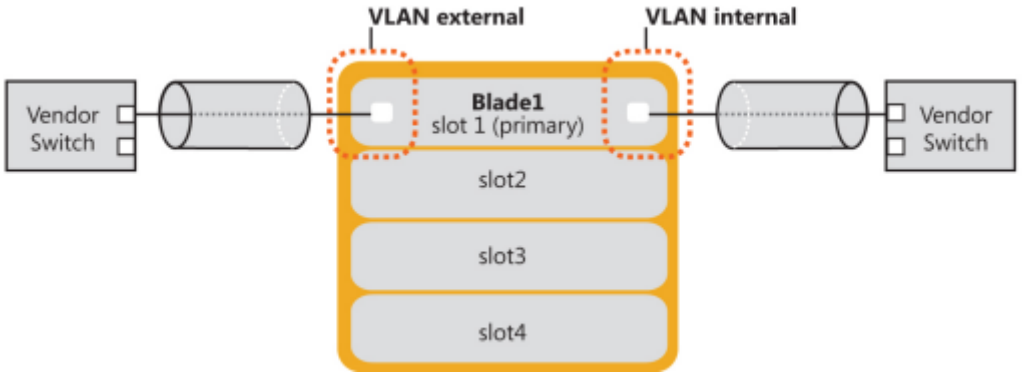

Figure 2.1 *Blade 1 with Layer 2 configuration data*

Important

Before you perform the following procedure, F5 Networks recommends that you read some background material on BIG-IP system VLANs, trunks, and self IP addresses. You can find this information in the most recent version of the ***TMOS® Management Guide for BIG-IP® Systems****.*

To configure trunks, VLANs, and self IP addresses

1. On the peer (vendor) switch on the external network, create a trunk that includes the four external interfaces to which you have physically connected the external interfaces of the four blades.

   Note: *If the peer switch is configured to use Link Aggregation Control Protocol (LACP), you must enable LACP.*

2. On the VIPRION system, using the cluster IP address, access the Configuration utility.
3. Create a trunk, and if the peer switch is configured to use LACP, enable LACP on the new trunk:
   a) On the Main tab of the navigation pane, expand **Network**, and click **Trunks**.
   The Trunks screen opens.
   b) At the upper right corner of the screen, click **Create**.
   The New Trunk screen opens.
   c) Assign the name **trunk_ext**, and assign an external interface of blade 1 to the trunk.
   d) Enable LACP mode, if required.
   e) Click **Finished**.
4. Create a VLAN for **trunk_ext**:
   a) On the Main tab of the navigation pane, expand **Network**, and click **VLANs**.
   The VLANs screen opens.
   b) On the upper right corner of the screen, click **Create**.
   The New VLAN screen opens.
   c) Configure a VLAN named **external**, assigning **trunk_ext** as an untagged interface.
   d) Click **Finished**.
5. Create a self IP address for VLAN **external**:
   a) On the Main tab of the navigation pane, expand **Network**, and click **Self IPs**.
   The Self IPs screen opens.
   b) At the upper right corner of the screen, click **Create**.
   The New Self IP screen opens.
   c) In the **IP Address** box, type an IP address.
   d) In the **Netmask** box, type a netmask for the IP address.
   e) From the **VLAN** list, select **external**.
   f) Click **Finished**.
6. On the Main tab of the navigation pane, expand **Network**, and click **Interfaces**.
   The Interfaces screen opens.

7. Verify the status of the blade 1 external interface.
   The interface (link) assigned to **trunk_ext** should appear as **UP**.
8. Return to step 1 and repeat the procedure for the internal network.
   To summarize:
   a) On the internal peer switch, create a trunk that includes an internal interface of blade 1.
   b) On the VIPRION system, create a trunk named **trunk_int** that includes one of blade 1's internal blade interfaces.
   c) Create an internal VLAN named **internal**, assigning **trunk_int** as an untagged interface.
   d) Create a self IP address for VLAN **internal**.
9. If you have a redundant system configuration, repeat this procedure on the peer system.

Configuring virtual servers and load balancing pools

After configuring the VLANs, self IP addresses, and trunks, you can configure one or more virtual servers and their associated load balancing pools. For information on configuring these objects, see the ***Configuration Guide for BIG-IP® Local Traffic Manager™***.

Understanding cluster synchronization

By now, you have configured the BIG-IP system objects required for processing application traffic. If you have a redundant system configuration, you have configured an additional VLAN, with a self IP address, that the two units can use for failover communication.

The system ensures that the configuration data is synchronized to all blades in the cluster. This synchronization process occurs immediately after you perform each configuration transaction on the primary blade.

Figure 2.2, on page 2-5 shows the effect of cluster synchronization on a second blade within the cluster. The figure shows that the system has propagated the configuration data of the primary blade to the secondary blade.

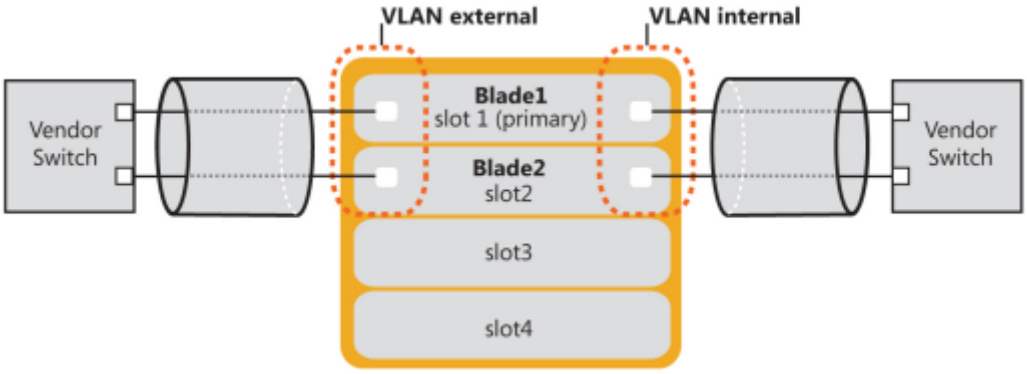

Figure 2.2 *Effect of cluster synchronization on blade 2*

The system is now ready to process application traffic. Whenever you make configuration changes to the cluster in the future, the system automatically ensures that those changes are synchronized across all blades in the cluster.

Setting up VLANs for redundancy

If you are planning to set up a redundant system configuration (that is, cluster-to-cluster failover capability), F5 recommends that you create an additional VLAN on each system of the redundant system configuration. This additional VLAN is used for the two clusters to communicate with each other before, during, and after failover. Note that you can create a trunk to assign to this VLAN, rather than assigning individual interfaces.

You can then create a self IP address for the new VLAN. Use the following procedure to create this VLAN and its self IP address. After you have followed this procedure, and cluster synchronization has occurred, the system is ready to fail over if the active cluster becomes unavailable in the future.

To configure additional VLANs and self IP addresses for a redundant system

1. On the Main tab of the navigation pane, expand **Networks,** and click **VLANs**.
   The VLANs screen opens.
2. At the upper right corner of the screen, click **Create**.
3. Configure a VLAN named **HA**, assigning the interface or trunk that the VLAN will use to communicate with its peer before, during, and after failover.
   You assign this interface (or trunk) as an untagged interface.

4. Click **Finished**.
5. On the Main tab of the navigation pane, click **Self IPs**.
   The Self IPs screen opens.
   a) At the upper right corner of the screen, click **Create**.
      The New Self IP screen opens.
   b) In the **IP Address** box, type a unique IP address.

      ***Note**: This self IP address does not need to be a floating IP address.*
   c) In the **Netmask** box, type a netmask for the IP address.
   d) From the **VLAN** list, select **HA**.
   e) Click **Finished**.
6. If you have a redundant system configuration, repeat this procedure on the peer system.

3

Managing Clusters

- Introducing cluster configuration
- Displaying cluster and cluster member properties
- Enabling and disabling cluster members
- Configuring cluster-related IP addresses

Introducing cluster configuration

The VIPRION® system is based on the concept of clusters that run the BIG-IP® system within a VIPRION system chassis. A ***cluster*** is a set of slots with blades in them. The blades in a cluster work together as one powerful VIPRION system to process network traffic. Blades in a cluster not only share the overall workload, but can also mirror each others' connections so that if a blade is taken out of service or becomes unavailable for some reason, any in-process connections remain intact.

Before a VIPRION system can process network traffic, you need to configure the cluster. To ensure that this task is simple and straightforward, the VIPRION system includes both a set of cluster management screens within the Configuration utility, and a set of cluster-relevant commands within the **bigpipe** utility. With these screens and commands, you can:

- Display cluster and cluster member properties, including status
- Enable or disable a cluster member
- Change the management IP addresses of a cluster and the cluster members (slots)

Displaying cluster and cluster member properties

As previously mentioned, you can access the Configuration utility on the VIPRION system using a special cluster IP address. After accessing the Configuration utility, you can navigate to the Cluster screen and see the properties of the default cluster. You can also see the properties of the slots and blades that represent the cluster members.

Displaying the properties of a cluster

Using the Configuration utility, you can view the properties of a cluster, and list the cluster members and their attributes. Table 3.1 describes the specific cluster properties that you can view.

Property	Description
Name	Displays the name of the cluster.
Cluster IP Address	Displays the IP address assigned to the cluster. You can click this IP address to change it.
Network Mask	Displays the network mask for the cluster IP address.
Primary Member	Displays the number of the slot that holds the primary blade in the cluster.
Software Version	Displays the version number of the BIG-IP software that is running on the cluster.

Table 3.1 *Properties of a cluster*

Property	Description
Software Build	Displays the build number of the BIG-IP software that is running on the cluster.
Hotfix Build	Displays the build number of any BIG-IP software hotfix that is running on the cluster.
Chassis 400-level BOM	Displays the bill-of-materials (BOM) number for the chassis.
Status	Displays an icon and descriptive text that indicates whether there are sufficient available members of the cluster.

Table 3.1 *Properties of a cluster (Continued)*

The Cluster screen also lists the members of the cluster and information about those members. The information associated with each cluster member includes:

- **Status**
  The Status column indicates whether the cluster member is available or unavailable.
- **Slot**
  The Slot column indicates the number of the slot. You can click this number to display the properties of that cluster member.
- **Blade serial number**
  The Blade Serial Number column displays the serial number for the blade currently in that slot.
- **Enabled**
  The Enabled column indicates whether that cluster member is currently enabled.
- **Primary**
  The Primary column indicates whether that cluster member is currently the primary slot.
- **HA State**
  The HA State column indicates whether the cluster member is used in a redundant system for high availability.

To view the cluster properties and list the cluster members

1. Using the cluster IP address, access the Configuration utility.
2. On the Main tab of the navigation pane, expand **System**, and click **Clusters**.
   The Cluster screen opens, showing the properties of the cluster, as well as listing the cluster members.

Displaying the properties of cluster members

When you view the Cluster screen to display the list of cluster members, you can click a slot number and display the properties of that slot. Table 3.2 describes the properties of cluster member that you can view.

Property	Description
Slot ID	Displays the slot number of the cluster member.
Cluster Member IP Address	Displays the management IP address of the slot.
Network Mask	Displays the network mask for the management IP address.
Blade Serial Number	Displays the serial number of the blade in the slot.
Primary Member	Indicates whether the cluster member is the primary member.
Enabled	Indicates whether the blade is enabled.

Table 3.2 *Properties of a cluster member*

Enabling and disabling cluster members

On the Cluster screen, you can not only display cluster and cluster member properties, but you can also enable or disable a cluster member. You do this using the **Enable** or **Disable/Yield** button.

To enable or disable a cluster member

1. On the Main tab of the navigation pane, expand **System**, and click **Clusters**.
   The Cluster screen opens, showing the properties of the cluster, as well as listing the cluster members.
2. Locate the cluster member you want to enable or disable, and to the left of the Status icon, check the Select box.
3. Click **Enable** or **Disable/Yield**.

Configuring cluster-related IP addresses

In addition to displaying cluster and cluster member properties, and enabling and disabling cluster members, you can change the cluster IP address of the cluster, and define or change management IP addresses for the cluster members.

F5 Networks strongly recommends that you define a management IP address for each slot. By doing so, you ensure that you can access any blade in the cluster if access through the cluster IP address is unsuccessful.

Important

All management IP addresses for the individual cluster members (slots) must be on the same network as the cluster IP address. Also, the gateway and netmask for the cluster IP address become the default values for the gateway and netmask for each individual management IP address.

Table 3.3 describes the settings you can configure.

Setting Type	Setting	Description
Management IP address	IP Address	Specifies the management IP address that you want to assign to the cluster. This IP address is used to access the Configuration utility, as well as to function as a cluster identifier for the peer cluster in a redundant system configuration.
	Network Mask	Specifies the network mask for the cluster IP address.
	Management Route	Specifies the gateway for the cluster IP address. Typically, this is the default route.
Cluster Member IP Address	Slot 1 IP Address	Specifies the management IP address associated with slot 1 of the cluster. You can also set this value to **None**.
	Slot 2 IP Address	Specifies the management IP address associated with slot 2 of the cluster. You can also set this value to **None**.
	Slot 3 IP Address	Specifies the management IP address associated with slot 3 of the cluster. You can also set this value to **None**.
	Slot 4 IP Address	Specifies the management IP address associated with slot 4 of the cluster. You can also set this value to **None**.

Table 3.3 *Configuring cluster-related management IP addresses*

To change a cluster-related management IP address

1. On the Main tab of the navigation pane, expand **System**, and click **Clusters**.
   The Cluster screen opens, showing the properties of the cluster, as well as listing the cluster members.
2. On the menu bar, click **Management IP Address**.
   The Management IP Address screen opens.
3. Locate the specific management IP address or cluster member IP address that you would like to change, and type the new IP address.
4. Click **Update**.

4

Configuring Clusters for Redundancy

- Introducing redundancy for VIPRION systems
- Designating a redundant pair
- Configuring redundancy properties
- Configuring network failover
- Configuring HA groups
- Configuring the default route on each target server
- Synchronizing configuration data
- Configuring fail-safe settings
- Manually changing the failover status of a cluster

Introducing redundancy for VIPRION systems

In Chapter 1, *Introducing the VIPRION System*, we described the VIPRION® system as a complete traffic management solution. That is, the VIPRION system meets the needs of large, enterprise networking environments that normally require multiple BIG-IP systems to process large volumes of application traffic. In this chapter, we introduce redundancy for VIPRION systems, which provides another level of reliability for a network environment.

Important

For VIPRION systems, use this chapter instead of the high availability chapter in the ***TMOS® Management Guide for BIG-IP® Systems****.*

When a VIPRION system consists of one cluster, you configure it as a single device. When a VIPRION system consists of two clusters (one cluster per chassis), you can configure the system for redundancy.

When you create a redundant system configuration, F5 Networks recommends that you configure the pair of clusters in active/standby mode. This means that one cluster is active and processing network traffic, while the other cluster is up and available to process traffic, but is in a standby state. If the active cluster becomes unavailable, the standby cluster automatically becomes active, and begins processing network traffic.

For example, you can configure one cluster to process traffic for virtual servers **A** and **B**. The standby cluster monitors the active cluster using either the cluster IP address, or the self IP address of VLAN **HA**. (The procedure for creating VLAN **HA** is described in Chapter 2, *Performing Initial System Configuration*.) This is part of the network failover process. If communications fail, the standby cluster initiates a failover and becomes the active cluster. The newly-active cluster then processes all new connections.

A standby cluster that becomes active normally remains active until an event occurs that requires the other cluster to become active again. When the failed cluster becomes available again, you can force the currently active cluster to change its state from active to standby, thereby initiating failback. Or, if you have set a preference for the cluster that failed to be the active cluster, the system automatically fails back, and that cluster begins processing traffic again.

WARNING

When you configure the system for redundancy, there is an option to configure the pair in active-active mode. However, F5 does not recommend active-active mode for a redundant system configuration.

Understanding redundancy for VIPRION systems

Failover, failback, network mirroring, configuration synchronization, and fail-safe are processes that ensure that the units of a redundant system configuration continue to process network traffic even in the event of a system failure. Table 4.1 contains a description of each of these processes.

Redundancy Process	Description
Failover	A process that occurs when one cluster of a redundant system configuration becomes unavailable, thereby requiring the peer cluster to assume the processing of traffic originally targeted for the unavailable cluster. To facilitate coordination of the failover process, you configure each cluster with a unit ID (1 or 2). For information about configuring network-based failover, see *Configuring network failover*, on page 4-8.
HA scoring	A process in which the system calculates an overall health score for a unit in a redundant system configuration, based on a weight that you assign to each trunk, pool, and cluster in an HA group. The unit that has the best overall score at any given time becomes or remains the active unit. For information about HA scoring, see *Configuring HA groups*, on page 4-11.
Failback	A process where, following failover, the previously unavailable cluster, which is configured with a preferred redundancy state of active, becomes available once again, and automatically begins processing traffic. For information about configuring failback, see *Designating a redundant pair*, on page 4-4.
Network Mirroring	A process whereby in-process connections on the active unit are automatically mirrored to the standby unit. This prevents any disruption in service if failover occurs.
Configuration Synchronization (ConfigSync)	A process where you replicate the configuration of one cluster on the peer cluster. Because data is shared in this way, each cluster can process the other cluster's traffic when failover occurs. For more information, see *What is configuration synchronization?*, on page 4-20.
Fail-safe	A process in which each cluster of a redundant system configuration monitors certain aspects of the system or network, detects interruptions, and consequently takes some action, such as initiating failover to the peer cluster. For information about configuring fail-safe, see *Configuring fail-safe settings*, on page 4-23.

Table 4.1 *Processes that ensure reliability for a redundant system configuration*

Summarizing redundant system configuration tasks

You can use the Configuration utility to separately configure each cluster in a redundant system configuration. After you configure the clusters, you also need to configure the default route on each back-end server to which the system sends network traffic.

The remainder of this chapter describes in detail all of the tasks that you must perform to set up a redundant system configuration. In summary, you must:

- Configure settings on each cluster, such as redundancy mode, failover IP addresses, and an HA group.
- Configure the default route on each back-end server.
- Synchronize the configuration data from cluster 1 to cluster 2.
- Configure fail-safe settings for system hardware and services, gateway router traffic, and VLAN traffic.

Once a redundant system configuration is operational, the BIG-IP system displays an indicator in the upper-left corner of all Configuration utility screens, as shown in Figure 4.1, to report the following information:

- The cluster you are currently managing (Unit 1 or Unit 2)
- The current state of the cluster (active or standby)
- Configuration synchronization status (this display is optional)

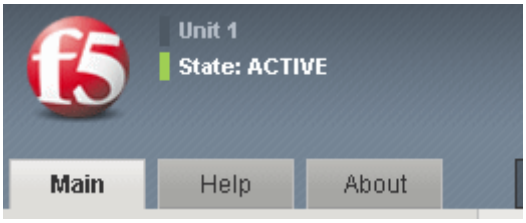

Figure 4.1 *Cluster status indicator in the Configuration utility*

Note

Only users with either the ***Administrator*** *or* ***Resource Administrator*** *user role can create a redundant system configuration.*

Designating a redundant pair

When you configure redundancy for a cluster, you must first designate the cluster as being a unit of a redundant system configuration. To do this, you use the Platform screen of the Configuration utility.

To designate a redundant pair

1. Access the Configuration utility, using the cluster IP address of one of the clusters in a redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **Platform**.
   The General screen opens.
3. From the **High Availability** list, select **Redundant Pair**.
4. From the **Unit ID** list, select **1**.

   Note: *When you configure the other cluster in the redundant system configuration, you must specify a* ***Unit ID*** *of* ***2****.*
5. Click **Update**.

Configuring redundancy properties

When you configure redundancy properties for a VIPRION system, you can use the default values on the Redundancy Properties screen. For a description of each setting on the Redundancy screen, as well as the default value for each setting, see Table 4.2, following.

Setting	Description	Default Value
Redundancy Mode	Specifies whether the redundant system configuration runs in **Active/Standby** or **Active/Active** mode. ***Important:*** *F5 recommends that you always use the* ***Active/Standby*** *mode for a redundant system configuration.*	**Active/Standby**
Redundancy State Preference	Specifies the preferred redundancy state for this cluster. F5 recommends that you use the default value, **None**. **None** specifies that this cluster does not have a preferred redundancy state. In this case, failback does not automatically occur, because a standby cluster that becomes active due to failover remains active until failover occurs again. However, you can initiate failback when the unavailable cluster becomes available again, by manually forcing the currently active cluster to revert to a standby state. **Active** specifies that this cluster is the preferred active cluster. If this cluster becomes unavailable, and failover occurs, when this cluster becomes available again, failback occurs automatically, making this cluster, once again, the active cluster. You should specify **Active** on only one cluster in a redundant system configuration. **Standby** specifies that this cluster is the preferred standby cluster. If the active cluster becomes unavailable, and failover occurs, this cluster becomes active; however, when the other cluster becomes available again, failback occurs automatically, making this cluster, once again, the standby cluster. This setting should only be set to **Active** on one cluster in a redundant system configuration. You should specify **Standby** on only one cluster in a redundant system configuration.	**None** (recommended)
Unit ID	Specifies the unit ID of a cluster in a redundant system configuration.	Defaults from the value in the **Unit ID** list on the General screen
Link Down Time on Failover	Specifies the amount of time, in seconds, that the interfaces for any external VLANs are down when the cluster fails over and goes into a standby state. Specifying a value other than **0** for this setting causes other vendor switches to use the specified time to learn the MAC address of the newly-active unit. You also use this setting to prompt peer switches to reset and relearn their ARP tables after a failover. The allowed value for this setting is **0** to **10**, inclusive. Setting the value to **0** disables this setting.	**0.0** seconds

Table 4.2 *Redundancy Properties screen settings*

To configure redundancy properties

1. Access the Configuration utility, using the cluster IP address of one of the clusters in the redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The Redundancy Properties screen opens.
3. From the **Redundancy Mode** list, either retain the default value of **Active/Standby**, or select **Active/Active**. For more information, see *Configuring the redundancy mode*, on page 4-6.
4. From the **Redundancy State Preference** list, select a preferred state for the unit you are configuring. For more information, see *Specifying a redundancy state preference*, on page 4-6.
5. If you want an interface down time other than **0.0**, type a value in the **Link Down Time on Failover** box. For more information, see *Specifying the link down-time on failover*, on page 4-7.

Configuring the redundancy mode

Configuring the redundancy mode means specifying whether your redundant system runs in active/standby mode or active-active mode. If you choose active/standby mode, you can specify whether you prefer that unit to be an active or a standby unit when both units are available for processing traffic. For information on setting a preference for a unit's redundancy state, see *Specifying a redundancy state preference*, on page 4-6.

Specifying a redundancy state preference

When you configure a system to be part of an active/standby redundant system, you can specify whether you want that system to function primarily as the active system or the standby system in the event that both units can be active at the same time. You can also can specify that you have no preference.

The preferences you can set are:

- **None**
  Specifies that this unit does not have a preferred redundancy state. In this case, failback does not normally occur, because a standby unit that becomes active due to failover remains active until failover occurs again. However, you can actively initiate failback when the unavailable unit becomes available again, by forcing the currently active unit to revert to a standby state. For more information, see *To force an active cluster into a standby state*, on page 4-28.
- **Active**
  Specifies that this unit is the preferred active unit. If you choose this option, the unit can be in a standby state due to a failover event. However, failback to this unit is automatic when this unit becomes available.

- **Standby**
  Specifies that this unit is the preferred standby unit. If you choose this option, then the unit can be in an active state due to a failover condition. However, failback to the peer unit is automatic when the peer unit becomes available.

Specifying the link down-time on failover

When configuring a unit of a redundant system configuration, you can use the **Link Down Time on Failover** setting to specify the amount of time, in seconds, that interfaces for any VLANs on external devices are down when the unit fails over and goes to a standby state. Specifying a value other than **0** for this setting causes other vendor switches to use the specified time to learn the MAC address of the newly-active unit.

The allowed value for this setting is **0** to **10**. Setting the value to **0** disables this setting.

Configuring network failover

When you configure ***network failover*** for a redundant system configuration, you configure each cluster to use the network to determine the status of the active cluster.

To do this, you specify, on each unit (chassis):

- The cluster floating management IP address of the peer cluster
- Unicast and multicast failover settings

Best practice

When you configure network failover addresses for a VIPRION system, you should configure both the **Unicast** and the **Multicast** settings. The best way to configure these addresses is as follows:

- The **Unicast** entry should contain the self IP addresses of the two **HA** VLANs that you configured earlier (on the local and remote units). For example, if the self IP addresses for the two **HA** VLANs are **10.10.10.2** and **10.10.10.3**, you add the unicast entry as in this example, where **bigip_unit2** is a configuration identifier that you specify for the entry:

  bigip_unit2|10.10.10.2|10.10.10.3|1026

- The **Multicast** entry should contain the local interface **MGMT** and a multicast address, as well as the applicable service port. For example:

  bigip_unit2|MGMT|224.0.0.245|62960

If you cannot configure a multicast address on the management port, the **Unicast** setting should contain the individual management IP addresses of each possible slot pair (resulting in four entries for two 2-slot clusters).

In the following example, each entry begins with a configuration identifier indicating the local slot and remote slot to which the entry applies, along with the applicable local and remote management IP addresses and the service port:

slot1_slot1|10.62.40.201|10.62.40.205|1026

slot1_slot2|10.62.40.201|10.62.40.206|1026

slot2_slot1|10.62.40.202|10.62.40.205|1026

slot2_slot2|10.62.40.202|10.62.40.206|1026

For added redundancy, you can add another unicast entry that specifies the self IP addresses corresponding to the two **HA** VLANs that you created earlier. For example, if the self IP addresses for the two **HA** VLANs are **10.10.10.2** and **10.10.10.3**, you can add this unicast entry:

bigip_B|10.10.10.2|10.10.10.3|1026

Understanding individual Network Failover settings

You use the Network Failover screen to configure the failover entries. Table 4.3 shows the network failover settings that you can configure for a cluster, followed by a description of each setting, and the default value of the setting.

Setting	Description	Default Value
Failover Status	Specifies the status of the cluster. When you create a redundant system configuration, you set the status of one cluster to **active** and the status of the other cluster to **standby**. To force a cluster to be unavailable, you select **forced offline**.	**Active**
Minimum Blades Up Enabled	Specifies, when checked, that the cluster fails over to the other cluster when the number of enabled slots in the active cluster falls below the number specified in the **Minimum Blades Up** setting.	Checked
Minimum Blades Up	Specifies the minimum number of blades in the cluster that must be enabled for the cluster to remain active. When the number of enabled blades falls below this number, the cluster fails over to the peer cluster.	**0**
Peer Management Address	Specifies the cluster floating management IP address of the peer cluster. This IP address is the address by which the cluster identifies its peer cluster.	**::**
Unicast settings		
Configuration Identifier	Specifies a unique identifier, required for each unicast entry.	No default value
Local Address	Specifies the self IP address associated with a VLAN on the cluster you are configuring. Failover messages from the cluster to its peer originate from this address.	No default value
Remote Address	Specifies the self IP address associated with a VLAN on the remote cluster. This is the IP address on the peer that receives a failover message from the cluster you are configuring.	No default value
Port	Identifies the service that you want to process the unicast failover communication traffic between the clusters.	**1026**
Multicast ***Important:*** *When configuring multicast settings, F5 recommends that you connect the management port on each blade to your management network.*		
Configuration Identifier	Specifies a unique identifier, required for each multicast entry.	No default value
Interface	Specifies the interface of the unit you are configuring.	**eth0**
Remote Address	Specifies a multicast address.	**224.0.0.245**
Port	Identifies the service that you want to process the multicast failover communication traffic between the clusters.	**62960**

Table 4.3 *Settings on the Network Failover screen*

To configure network failover for a cluster

1. Access the Configuration utility, using the cluster IP address of one of the clusters in the redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The Redundancy Properties screen opens.
3. On the menu bar, click **Network Failover**.
   The Network Failover screen opens.
4. From the **Failover Status** list, select **active**.

   Note: *When you configure the other cluster in the redundant system configuration, select* ***standby*** *from the* ***Failover Status*** *list.*
5. To enable the **Minimum Blades Up** setting, check the **Minimum Blades Up Enabled** check box.
6. In the **Minimum Blades Up** box, type the number of blades in this cluster that must be available for the cluster to remain active.
7. In the **Peer Management Address** box, type the cluster floating management IP address of the peer cluster.
8. In the Unicast area, configure these settings:
   a) In the **Configuration Identifier** box, type a unique identifier, such as the name of the peer cluster.
   b) In the **Local Address** box, type the self IP address of a local VLAN (such as VLAN **HA**).
   c) In the **Remote Address** box, type the self IP address associated with a VLAN on the peer cluster (such as VLAN **HA**).
   d) In the **Port** box, retain the default value.
9. In the Multicast area, configure these settings:
   a) In the **Configuration Identifier** box, type a unique identifier, such as the name of the peer cluster.
   b) In the **Interface** box, specify an interface, such as **MGMT**.
   c) In the **Remote Address** box, type a multicast address or retain the default value.
   d) In the **Port** box, retain the default value.
10. Click **Update**.

Configuring HA groups

The BIG-IP system includes a feature known as an HA group. An ***HA group*** is a set of trunks, pools, or clusters (or any combination of these) that you want the BIG-IP system to use to calculate an overall health score for a unit in a redundant system configuration. A health score is based on the number of members that are currently available for any trunks, pools, and clusters in the HA group, combined with a weight that you assign to each trunk, pool, and cluster. The unit that has the best overall score at any given time becomes or remains the active unit.

An HA group is typically configured to fail over based on trunk health in particular. Trunk configurations are not synchronized between units, which means that the number of trunk members on the two units often differs whenever a trunk loses or gains members. The HA group feature allows failover to occur based on changes to trunk health instead of on system or VLAN failure.

Only one HA group can exist on the BIG-IP system. By default, the HA group feature is disabled.

One of the benefits of configuring an HA group is a feature known as ***fast failover***. When you configure the HA group, the process of one BIG-IP unit failing over to the other based on HA scores is noticeably faster than if failover occurs due to a hardware or daemon failure.

Understanding HA score calculation

The BIG-IP system calculates an HA score based on these criteria:

- The number of available members for each object (such as a trunk)
- The weight that you assign to each object in the HA group
- The threshold you specify for each object (optional)
- The active bonus value that you specify for the HA group

Specifying a weight

A ***weight*** is a health value that you assign to each object in the HA group (that is, pool, trunk, and cluster). The weight that you assign to each object must be in the range of **10** through **100**. The maximum overall score that the BIG-IP system can potentially calculate for a unit is the sum of the individual weights for the HA group objects, plus the active bonus value. (For information on the **Active Bonus** setting, see *Specifying an active bonus*, on page 4-13.)

Table 4.4, on page 4-12 shows an example of how the system calculates a score for the unit, based solely on the weight of objects in the HA group. In this example, the HA group contains two pools (**my_http_pool** and **my_ftp_pool**) and one trunk (**my_trunk1**). A user has assigned a weight to each object.

Object Name	Members	Available Members	User-assigned Weight	HA Score
my_http_pool	8	5 (62.5%)	**50**	31 (60% x 50)
my_ftp_pool	6	6 (100%)	**20**	20 (100% x 20)
my_trunk1	4	3 (75%)	**30**	23 (75% x 30)
				Total unit score = 74

Table 4.4 *Example of an HA score calculation for a unit*

On each unit, the system uses each weight, along with a percentage that the system derives for each object (the percentage of the object's members that are available), to calculate a score for each object.

The system then adds the scores to determine a total score for the unit. The unit with the highest score becomes or remains the active unit in the redundant system configuration.

Note that if you have configured VLAN failsafe, and the VLAN fails on an active unit, the unit goes offline regardless of its score, and its peer becomes active.

Specifying a threshold

For each object in an HA group, you can specify an optional setting known as a threshold. A ***threshold*** is a value that specifies the number of object members that must be available to prevent failover. If the number of available members dips below the threshold, the BIG-IP system assigns a score of **0** to the object, so that the score of that object no longer contributes to the overall score of the unit.

For example, if a trunk in the HA group has four members and you specify a threshold value of **3**, and the number of available trunk members falls to 2, then the trunk contributes a score of **0** to the total unit score.

If the number of available object members equals or exceeds the threshold value, or you do not specify a threshold, the BIG-IP system calculates the score as described previously, by multiplying the percentage of available object members by the weight for each object and then adding the scores to determine the overall unit score.

For the procedure on specifying a threshold, see *Configuring an HA group*, on page 4-16.

◆ Tip

*Do not configure the **tmsh** attribute **min-up-members** on any pool that you intend to include in the HA group.*

Specifying an active bonus

An ***active bonus*** is an amount that the BIG-IP system automatically adds to the overall score of the active unit. An active bonus ensures that the active unit remains active when the unit's score would otherwise temporarily fall below the score of the standby unit.

A common reason to specify an active bonus is to prevent failover due to ***flapping***, the condition where failover occurs frequently as a trunk member toggles between availability and unavailability. In this case, you might want to prevent the HA scoring feature from triggering failover each time a trunk member is lost. You might also want to prevent the HA scoring feature from triggering failover when you make minor changes to the BIG-IP system configuration, such as adding or removing a trunk member.

Suppose that the HA group *on each un*it contains a trunk with four members, and you assign a weight of **30** to each trunk. Without an active bonus defined, if the trunk on unit 1 loses some number of members, failover occurs because the overall calculated score for unit 1 becomes lower than that of unit 2. Table 4.5 shows the scores that could result if the trunk on unit 1 loses one trunk member and no active bonus is specified.

Event	Members Available		Trunk score		Unit Score		Failover?
	Unit 1	Unit 2	Unit 1	Unit 2	Unit 1	Unit 2	
Unit 1 is active (initial state)	4 (100%)	4 (100%)	30	30	**30**	30	No
Unit 1 loses a trunk member	3 (75%)	4 (100%)	23 (75% of 30)	30 (100% of 30)	**23**	30	**Yes**

***Table 4.5** Failover when no active bonus is specified*

You can prevent this failover from occurring by specifying an active bonus value. In our example, if we specify an active bonus of **10** (the default value), the score of the active unit changes from **23** to **33**, thereby ensuring that the score of the active unit remains equal to or higher than that of the standby unit (**30**).

Although you specify an active bonus value on each unit, the BIG-IP system uses the active bonus specified on the active unit only, to contribute to the score of the active unit. The BIG-IP system never uses the active bonus on the standby unit to contribute to the score of the standby unit.

Important

*An exception to this behavior is when the active unit score is **0**. If the score of the active unit is **0**, the system does not add the active bonus to the active unit score.*

To decide on an active bonus value, calculate the trunk score for some number of failed members (such as one of four members), and then specify an active bonus that results in a trunk score that is greater than or equal to the weight that you assigned to the trunk.

For example, if you assigned a weight of **30** to the trunk, and one of the four trunk members fails, the trunk score becomes **23** (75% of **30**), putting the unit at risk for failover. However, if you specified an active bonus of **7** or higher, failover would not actually occur, because a score of **7** or higher, when added to the score of **23**, is greater than or equal to **30**.

For the procedure on specifying an active bonus, see *Configuring an HA group*, on page 4-16.

Example

You can prevent failover from occur i ng by specifying an active bonus value. Table 4.6, on page 4-15 and the list that follows shows how configuring an active bonus for the active unit can affect failover.

	Event	Trunk Members Available		Trunk Score		Active Bonus		Unit Score (Trunk Score + Active Bonus)		Failover?
		Unit 1	Unit 2	Unit 1	Unit 2	Unit 1	Unit 2	Unit 1	Unit 2	
1	Unit 1 is active (initial state)	4 (100%)	4 (100%)	30 (100% of 30)	30 (100% of 30)	**10**	0	**40**	30	No
2	Unit 1 loses a trunk member	3 (75%)	4 (100%)	23 (75% of 30)	30 (100% of 30)	**10**	0	**33**	30	No
3	Unit 1 loses another trunk member	2 (50%)	4 (100%)	15 (50% of 30)	30 (100% of 30)	**10**	0	25	**30**	Yes
4	Unit 1 goes to standby. Unit 2 goes active.	2 (50%)	4 (100%)	15 (50% of 30)	30 (100% of 30)	0	**10**	15	**40**	N/A
5	Unit 2 loses a trunk member	2 (50%)	3 (75%)	15 (50% of 30)	23 (75% of 30)	0	**10**	15	**33**	No
6	Unit 1 regains two trunk members	4 (100%)	3 (75%)	30 (100% of 30)	23 (75% of 30)	0	**10**	30	**33**	No

Table 4.6 *Example of configuring an active bonus to prevent failover*

To help understand Table 4.6, the row numbers in the left column of the table correspond to the explanations below:

1 Unit 1 is active (initial state)
With all trunk members available on both units, and the active bonus configured, the active unit (unit 1) retains the higher unit score and therefore remains active.

2 Unit 1 loses a trunk member
The unit score for unit 1 is still higher than the score for unit 2, due to an active bonus value of **10**.

3 **Unit 1 loses another trunk member**
With an active bonus of **10**, failover occurs when 50% of the members are lost.

4 **Unit 1 switches to standby mode and unit 2 becomes active**
Once the active unit (unit 1) has failed over to unit 2, the active bonus on unit 1 no longer applies, thus reducing its score from **25** to **15**. The active bonus on unit 2 is then applied, increasing unit 2's score from **30** to **40**.

5 **Unit 2 loses a trunk member**
If the active unit (unit 2) loses a trunk member, the score on unit 2 is still higher than unit 1 (with two unavailable members), due to the active bonus.

6 **Unit 1 regains two trunk member**
Unit 2 remains the active unit even when one trunk member is unavailable, due to the active bonus.

Configuring an HA group

To configure the system so that failover can occur based on an HA score, you must specify values for the properties of an HA group. The BIG-IP system allows you to configure one HA group only; you cannot create additional HA groups.

Once you have configured HA group properties, the BIG-IP system uses that configuration to calculate an overall HA score for each unit of the redundant system configuration.

To configure an HA group

1. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
2. On the menu bar, click **HA Group**.
3. In the HA Group Properties area of the screen, configure these settings:

Setting	Required Value
HA Group Name	Assign a name to the HA group.
Enable	Verify that the Enable box is checked.
Active Bonus	Specify an integer the represents the amount by which you want the system to increase the overall score of the active unit. The purpose of the active bonus is to prevent failover when minor or frequent changes occur to the configuration of a pool, trunk, or cluster.

Setting	Required Value
Pools	In the Available box, click a pool name and use the Move button to move the pool name to the Selected box. This populates the table that appears along the bottom of the screen with information about the pool.
Trunks	In the Available box, click a trunk name and use the Move button to move the trunk name to the Selected box. This populates the table that appears along the bottom of the screen with information about the trunk.
Clusters	In the Available box, click a cluster name and use the Move button to move the cluster name to the Selected box. This populates the table that appears along the bottom of the screen with information about the cluster. ***Note**: This setting appears on VIPRION systems only.*

4. In the table displayed along the bottom of the screen, configure these settings:

Setting	Required Value
Threshold	For each pool or trunk in the HA group, specify an integer for a threshold value. This value is optional.
Weight	For each pool or trunk in the HA group, specify an integer for the weight. The allowed weight for an HA group object ranges from **10** through **100.** This value is required.

5. Click **Create**.

Viewing HA scores and other details

You can view the HA score that the BIG-IP system has calculated for each unit at any given time. To view the score, as well as other details about the HA group, use the **tmsh** command line interface.

◆ Note

*You must have permission to access the advanced shell **tmsh**.*

To view the HA score and other details

At the system prompt on unit 1, type:

```
tmsh
/sys
show ha-group <HA_group_name> details
```

Repeat the commands on unit 2.

To compare the HA scores of both units

You can compare the score of the HA score on the current unit with the HA score of the peer unit. At the system prompt on either unit, type:

```
tmsh
/sys
show ha-status all-properties
```

Configuring the default route on each target server

Once you have configured each cluster in a redundant system configuration, you can configure the default route on each back-end server.

◆ Tip

For any back-end server that receives requests from a cluster in a redundant system configuration, you do not need to perform this step if the relevant virtual server is associated with a SNAT.

When failover occurs in a redundant system configuration, the surviving cluster begins handling the inbound virtual server connections (those targeted to back-end servers) that are normally processed by the failed cluster. The surviving cluster also begins handling the outbound connections (those originating from back-end servers) that are normally destined for the failed cluster.

For a redundant system configuration to do this properly, you need to set the default route for each back-end server to the shared, floating IP address assigned to the cluster that normally processes the server's responses. This ensures that the back-end server can successfully send a response to the surviving cluster.

For example, for an active/standby configuration, if the server **http_server** normally receives connections from cluster 1 of the system, and the floating IP address shared by the two clusters is **11.12.11.3**, you must set the default route for server **http_server** to **11.12.11.3.** Then, if cluster 1 goes out of service, the surviving cluster (cluster 2) can receive the server's response because the IP address **11.12.11.3** is a shared IP address.

◆ Note

Follow your server vendor's instructions for the procedure on setting a default route.

Synchronizing configuration data

Once you have completed the initial configuration of the clusters of a redundant system configuration, you must synchronize the configurations of the two clusters.

◆ Note

*Only users with either the **Administrator** or **Resource Administrator** user role can synchronize configuration data.*

What is configuration synchronization?

When you synchronize data from one cluster to another, you are giving the target cluster the data that it needs to assume traffic processing for its peer when failover occurs. Examples of configuration data that a target cluster receives during configuration synchronization are virtual servers, SNATs, and floating IP addresses.

When you have a redundant system configuration, it is essential that each cluster shares, or synchronizes, its current configuration data with its peer cluster. If a cluster does not share its configuration data with its peer, the surviving cluster cannot process traffic for that peer cluster. For this reason, you must synchronize configuration data when you initially create the redundant system configuration, and then repeatedly, on an ongoing basis. You need to repeatedly synchronize data because a cluster's configuration data typically changes over time during normal system maintenance, such as when you add a virtual server or create a new profile, and the cluster must share those changes with its peer.

◆ Tip

*Only users with either the **Administrator** or **Resource Administrator** role assigned to their user accounts can perform configuration synchronization. Consequently, only these BIG-IP system user accounts appear on the ConfigSync screen, in the **ConfigSync User** setting.*

With respect to configuration synchronization, you can use the Configuration utility to:

- View or specify the peer IP address to use for synchronization
- Enable or disable encryption of configuration data prior to synchronization
- Enable or disable the global display of synchronization status

Configuring ConfigSync settings

You configure the configuration synchronization settings on the ConfigSync screen. F5 recommends that you set up configuration synchronization to take place over VLAN **HA**. This isolates the traffic between the clusters in a redundant system configuration from the production traffic.

Table 4.7 describes the configuration synchronization settings that you can configure for a cluster, as well as the default value for each setting.

Setting	Description	Default Value
ConfigSync Peer	Specifies the type of IP address that you want the system to use when synchronizing the configuration data. The two possible values are: **Use Primary Connection Mirror Address** This value causes the system to use the peer cluster's primary connection mirroring address. F5 recommends that this address be the self IP address associated with VLAN **HA**. For more information, see *Configuring network mirroring settings*, on page 5-2. **Specify IP Address** This value allows you to type an IP address of your choice for the **ConfigSync Peer Address** setting (described below). Select this value when you do not want the system to use the peer cluster's primary failover address to synchronize the configuration.	**Use Primary Connection Mirror Address**
ConfigSync Peer Address	Specifies the IP address of the peer cluster that you want the system to use for configuration synchronization. When the **ConfigSync Peer** setting is set to **Use Primary Connection Mirror Address** (the default setting), you cannot configure the **ConfigSync Peer Address** setting. However, if you set the **ConfigSync Peer** setting to **Specify IP Address**, you can type an IP address for the peer cluster.	**::**
ConfigSync User	Specifies the name of the user whose credentials are used to authenticate one cluster to the other cluster. In order for the ConfigSync process to function correctly, the name and password for this user must be the same on both clusters. When you change the **ConfigSync User** password using the Configuration utility, the system prompts you to update the password on the peer cluster. ***Note:*** *When you use the* ***bigpipe*** *utility to change this name and password, the system does not display a message prompting you to change the user name and password on the peer cluster. You must remember to make the same changes on the peer cluster.*	**admin** ***Note:*** *The system does not require a password for the* ***admin*** *account.*
Encryption	Enables the system to encrypt the configuration data immediately prior to synchronization. The two possible values are **on** and **off**. When you enable encryption, you must configure the **Passphrase** and **Verify Passphrase** settings	**off**
Passphrase	Specifies a password for encryption.	No default
Verify Passphrase	Confirms the encryption password when you type it again.	No default

Table 4.7 *ConfigSync screen settings*

Setting	Description	Default Value
Detect ConfigSync Status	Specifies that the system displays synchronization status on each screen of the Configuration utility.	not checked
Synchronize	Performs a configuration synchronization between two clusters, when you click either of the following buttons: **Synchronize TO Peer** Writes the configuration data of the cluster you are currently configuring to the peer cluster. **Synchronize FROM Peer** Writes the configuration data of the peer cluster to the cluster you are currently configuring. ***Note:*** *The self IP address of the peer cluster appears in the* ***ConfigSync Peer Address*** *field.*	No default

Table 4.7 *ConfigSync screen settings (Continued)*

Performing configuration synchronization

Once you have configured the clusters in a redundant system configuration, you must synchronize the configurations.

To perform configuration synchronization

1. Access the Configuration utility, using the cluster IP address of one of the clusters in the redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The Redundancy Properties screen opens.
3. On the menu bar, click **ConfigSync**.
4. To perform a configuration synchronization, click one of the following buttons:
   - **Synchronize TO Peer**
     Use this button when the cluster you are currently configuring contains updated configuration data that you want to share with the peer cluster.
   - **Synchronize FROM Peer**
     Use this button when the peer cluster contains updated configuration data that you want to share with the cluster you are currently configuring.

   Note: *The self IP address of the peer cluster appears in the* ***ConfigSync Peer Address*** *box.*

Configuring fail-safe settings

Fail-safe is the ability of a VIPRION system to monitor certain aspects of the system or network, detect interruptions, and consequently take some action. In the case of a redundant system configuration, a cluster can detect a problem and initiate failover to the peer cluster. When you configure the fail-safe feature on a VIPRION system, you are specifying the particular events that cause failover to occur in a redundant system configuration. The fail-safe feature applies to:

- System services
- Traffic between the VIPRION system and a gateway router
- Traffic on a VLAN

◆ Note

*Only users with either the **Administrator** or **Resource Administrator** user role can configure fail-safe.*

Configuring system fail-safe

You can configure the BIG-IP system to monitor various system services and then take some action if the system detects a heartbeat failure. Table 4.8 lists these services and the possible actions available to the system when a heartbeat failure occurs.

Service	Actions available upon heartbeat failure
BIGD (health monitors)	**Restart**, **Restart All**, **Reboot**, **Go Offline**, **Go Offline Abort TM**, and **No Action**. The default is **Restart**.
MCPD (messaging and configuration)	**Restart**, **Restart All**, **Reboot**, **Go Offline**, **Go Offline Abort TM**, and **No Action**. The default is **Restart All**.
SOD (high availability management)	**Reboot**, **Restart All**, and **No Action**. The default is **Restart All**.
TMM (traffic management)	**Restart, Restart All, Reboot, Go Offline, Go Offline and Restart, Go Offline and Down Links and Restart.** The default is **Go Offline and Down Links and Restart**.
BCM56XXD (switch hardware driver)	**Restart** and **No Action**. The default is **Restart**.
CLUSTERD (cluster-wide status management)	**Go Offline and Down Links and Restart.**
TMROUTED (routing table management)	**Restart**, **Restart All**, **Reboot**, **Go Offline**, **Go Offline Abort TM, No Action**. The default is **Restart**.

***Table 4.8** System services and actions available upon heartbeat failure*

Table 4.9 describes each heartbeat failure action.

Action	Description
Restart	Selected service restarts.
Restart All	All system services restart.
Reboot	System reboots.
Go Offline	System becomes unavailable.
Go Offline and Restart	System becomes unavailable, and selected service restarts.
Go Offline Abort TM	System becomes unavailable, and the traffic management service stops.
Go Offline and Down Links	System and all links become unavailable.
Go Offline and Down Links and Restart	System and all links become unavailable, and the selected service restarts.
No Action	System takes no action.

Table 4.9 *Descriptions of system service heartbeat failure actions*

To configure system-service monitoring

1. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The Redundancy Properties screen opens.
2. From the Fail-safe menu, choose System.
   The System Fail-safe screen opens.
3. In the System Services area, in the Name column, click the name of the service you want to monitor.
   The screen for that service opens.
4. From the **Heartbeat Failure** list, select an action, or retain the default setting.
5. Click **Finished**.
   The System Fail-safe screen displays.

Configuring gateway fail-safe

Fail-safe features on the VIPRION system provide network failure detection based on network traffic. One type of network failure detection is known as gateway fail-safe. ***Gateway fail-safe*** monitors traffic between the active cluster and a pool containing a gateway router, thereby protecting the cluster from a loss of an internet connection by triggering a failover when a gateway router is unreachable for a specified duration. If you want failover to occur when a gateway router is unreachable, you can configure the gateway fail-safe feature. It is important to note that you use this feature when the peer cluster uses a different route to the Internet.

You can configure gateway fail-safe using the Configuration utility. Configuring gateway fail-safe means designating a load balancing pool as a gateway fail-safe pool.

Configuring gateway pool settings

Table 4.10 describes the settings that you can configure when you designate a an existing load balancing pool as a gateway fail-safe pool, as well as the default value for each setting.

Setting	Description	Default Value
Gateway Pool	Specifies the name of the pool.	No default
Unit ID	Specifies the Unit ID number of the peer cluster on which the gateway pool is configured.	No default
Threshold	Specifies the minimum number of gateway pool members that must be available to avoid the designated action	**0**
Action	Specifies the action that the system takes when the number of available gateway pool members drops below the designated threshold.	**Failover**

Table 4.10 *Add Gateway Pool screen settings*

To configure gateway fail-safe

1. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The Redundancy Properties screen opens.
2. From the Fail-safe menu, choose Gateway.
   The Gateway Fail-safe screen opens.
3. In the upper-right corner of the screen, click **Add**.
   The Add Gateway Pool screen opens.
4. From the **Gateway Pool** list, select the name of a load balancing pool.
5. From the **Unit ID** list, select a unit ID for the pool (**1** or **2**).

6. In the **Threshold** box, type the number of pool members that must be available to avoid the action designated in the **Action** box.
7. Retain the default value in the **Action** box, **Failover**.
8. Click **Finished**.

Configuring VLAN fail-safe

For maximum reliability, the BIG-IP system supports failure detection on all VLANs. When you configure the fail-safe option on a VLAN, the system monitors network traffic going through a VLAN. If the system detects a loss of traffic on a VLAN and the fail-safe timeout period has elapsed, the system attempts to generate traffic by issuing ARP requests to nodes accessible through the VLAN. The BIG-IP system also generates an ARP request for the default route, if the default router is accessible from the VLAN. Failover is averted if the system is able to send and receive any traffic on the VLAN, including a response to its ARP request.

If the BIG-IP system does not receive traffic on the VLAN before the timeout period expires, the system either restarts all system services, reboots, or fails over to the peer unit, depending on how you configure the VLAN fail-safe feature. The default action is to reboot the system.

WARNING

You should configure the fail-safe option on a VLAN only after the VIPRION system is in a stable production environment. Otherwise, routine network changes might cause failover unnecessarily.

There are two ways to configure VLAN fail-safe: from the Redundancy Properties screen, or from the VLANs screen.

To configure VLAN fail-safe using the Redundancy Properties screen

1. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The Redundancy Properties screen opens.
2. From the Fail-safe menu, choose VLANs.
   The VLAN Fail-safe screen opens.
3. In the upper-right corner of the screen, click **Add**.
   The Add VLAN screen opens.
4. From the **VLAN** list, select a VLAN name.
5. In the **Timeout** box, specify the period of time during which traffic should be detected on the VLAN, after which the designated action will occur. The default value, in seconds, is **90**.

6. From the **Action** list, select the action that the system should take when the timeout period expires.
7. Click **Finished**.

To configure VLAN fail-safe using the VLANs screen

1. On the Main tab of the navigation screen, expand **Network**, and click **VLANs**.
   The VLAN list screen opens.
2. Click the name of the VLAN for which you want to configure fail-safe.
3. From the **Configuration** list, select **Advanced**.
4. In the **Fail-safe** setting, check the box.
   This shows additional settings.
5. In the **Fail-safe Timeout** box, specify the period of time during which traffic should be detected on the VLAN, after which the designated action occurs. The default value, in seconds, is **90**.
6. From the **Action** list, select the action that the system should take when the timeout period expires.
7. Click **Update**.

Manually changing the failover status of a cluster

After you initially configure the clusters in a redundant system configuration, the system determines the failover status of each cluster based on how you configured the system, and on the current state of the system. However, you can also manually change the failover status of a cluster. For example, when you want to perform maintenance on the active cluster in a redundant system configuration, you can force that cluster into a standby state. If necessary, you can also force a cluster to be unavailable by changing the failover status to **forced_offline**.

Forcing an active cluster into a standby state

Normally, the system determines the failover status of a cluster. However, you can force the active cluster into a standby state. When you do this, the standby cluster automatically becomes active, and processes the application traffic on behalf of the two clusters.

To force an active cluster into a standby state

1. Access the Configuration utility, using the cluster IP address of the cluster that you want to force into a standby state.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The Redundancy Properties screen opens.
3. Click **Force to Standby**.

Changing the failover state of a cluster to FORCED_OFFLINE

Normally, the system determines the failover status of a cluster. If necessary, you can force a cluster to be unavailable by changing the failover state to **FORCED_OFFLINE**.

When you force the active cluster offline, the standby cluster automatically becomes active and processes the application traffic on behalf of the two clusters. However, if failover occurs, the cluster you force offline cannot become active until you release the cluster from its **FORCED OFFLINE** state.

When you force the standby cluster offline, the active cluster simply continues to process the application traffic.

Important

*Before you force a cluster offline, if connection mirroring is enabled, you must change the value of the connection mirroring settings for each cluster in the redundant system configuration from **Within cluster** to **Between clusters**. This enables the system to continue processing current connections*

without interruption. For instructions on how to modify the connection mirroring settings, see ***Configuring mirroring on a VIPRION system****, on page 5-2.*

To force an active cluster offline

1. Access the Configuration utility, using the cluster IP address of the active cluster in a redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The Redundancy Properties screen opens.
3. Click **Force Offline**.

To release an active cluster from an OFFLINE state

1. Access the Configuration utility, using the cluster IP address of the active cluster in a redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The Redundancy Properties screen opens.
3. Click **Release Offline**.

5

Configuring Mirroring for VIPRION Systems

- Introducing mirroring
- Understanding mirroring for VIPRION systems
- Configuring mirroring on a VIPRION system
- Configuring mirroring when performing system maintenance

Introducing mirroring

When you configure mirroring for a VIPRION® system, you enhance the reliability of the system. You do this by configuring the system to process connections without interruption, even in the case of a system failure. This chapter defines mirroring, and describes how to configure mirroring for VIPRION systems.

Understanding mirroring for VIPRION systems

There are two types of mirroring that you can configure for a VIPRION system:

- **Connection mirroring**
  The VIPRION system mirrors network connections.
- **Session persistence mirroring**
  The VIPRION system mirrors session persistence records.

When you configure mirroring, you can configure the VIPRION system for:

- **Intra-cluster mirroring**
  The VIPRION system mirrors the connections and session persistence records within the cluster, that is, between the blades in the cluster. You can configure intra-cluster mirroring on both single devices and redundant system configurations. It is important to note that F5 does not support intra-cluster mirroring for Layer 7 (non-FastL4) virtual servers.
- **Inter-cluster mirroring**
  The VIPRION system mirrors the connections and session persistence records that the blades in a cluster are processing to the other cluster in a redundant system configuration. You can configure inter-cluster mirroring on a redundant system configuration only, and only when the identical number of blades are present in the identical number of slots in each of the two clusters of the redundant system configuration.

Intra-cluster mirroring and inter-cluster mirroring are mutually exclusive. F5 recommends that you configure intra-cluster mirroring, except when you want to perform maintenance on the active cluster of a redundant system configuration. For more information, see *Configuring mirroring when performing system maintenance*, on page 5-7.

◆ Note

Connection mirroring enhances the reliability of your system; however, it might degrade system performance.

The remainder of this chapter describes how to configure mirroring for VIPRION systems.

Configuring mirroring on a VIPRION system

When you want to add an extra level of reliability to a VIPRION system, you can configure one or more virtual servers to mirror connections. You can also configure the virtual servers to mirror session persistence records.

The first step to configuring mirroring is to access a cluster using the cluster's floating IP address. Then you must perform two tasks to configure mirroring on the system:

- **Configure network mirroring**
  Specify the way that you want the system to mirror connections.
- **Configure the virtual servers**
  Enable connection mirroring on each applicable virtual server. Optionally, you can configure the virtual server to mirror session persistence records.

Configuring network mirroring settings

You can configure network mirroring settings for a cluster. Table 5.1 shows the network mirroring settings with a description, and the default value of each setting.

Setting	Description	Default Value
Network Mirroring	Specifies whether you want to mirror connections and session persistence records between blades within the cluster (intra-cluster mirroring) or between the clusters in a redundant system configuration (inter-cluster mirroring).	**Within cluster**
Mirroring Address - Self	Specifies the static self IP address of VLAN **HA** of the cluster you are configuring.	**::**
Mirroring Address - Peer	Specifies the static self IP address of VLAN **HA** on the peer cluster.	**::**
Alternate Mirroring Address - Self	Specifies an alternate address to use for mirroring on the cluster you are configuring.	**::**
Alternate Mirroring Address - Peer	Specifies an alternate address to use for mirroring on the peer cluster.	**::**

Table 5.1 *Network mirroring screen settings*

Configuring intra-cluster mirroring on a VIPRION system

When you configure intra-cluster mirroring, you configure a cluster to mirror the connections it is processing between the blades (within the cluster). You do this on the Redundancy Properties screen.

To configure intra-cluster mirroring

1. Access the Configuration utility, using the cluster IP address of one of the clusters in the redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The System Fail-safe screen opens.
3. On the menu bar, click **Network Mirroring**.
   The Redundancy Properties screen opens.
4. From the **Network Mirroring** list, select **Within cluster**.
5. Click **Update**.
6. If you are creating a redundant system configuration, on the menu bar, click **ConfigSync**.
   The ConfigSync screen opens.
7. Click the **Synchronize TO Peer** button.
   The system applies the configuration change to the peer cluster in the redundant system configuration.

In order to complete the intra-cluster mirroring configuration, you must perform the following procedure, *Configuring virtual servers to mirror connections*.

Configuring inter-cluster mirroring on a VIPRION system

When you configure inter-cluster mirroring, you configure a cluster to mirror its connections to another cluster. You do this on the Redundancy Properties screen.

To configure inter-cluster mirroring

1. Access the Configuration utility, using the cluster IP address of one of the clusters in the redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The System Fail-safe screen opens.
3. On the menu bar, click **Network Mirroring**.
   The Redundancy Properties screen opens.
4. From the **Network Mirroring** list, select **Between clusters**.
5. Configure mirroring addresses as described in *To configure the connection mirroring addresses for a cluster*, on page 5-8.
6. Click **Update**.
7. On the menu bar, click **Network Failover**.
   The Network Failover screen opens.
8. Click on the **Minimum blades up** check-box.
9. Enter the number of blades installed on each VIPRION system. This ensures that failover occurs when any blade becomes unavailable and that mirroring behaves as expected.
10. If you are creating a redundant system configuration, on the menu bar, click **ConfigSync**.
    The ConfigSync screen opens.
11. Click the **Synchronize TO Peer** button.
    The system applies the configuration change to the peer cluster in the redundant system configuration.

In order to complete the inter-cluster mirroring configuration, you must perform the following procedure, *Configuring virtual servers to mirror connections*.

Configuring virtual servers to mirror connections

After you configure a VIPRION system to mirror connections within the cluster, you must enable connection mirroring on any applicable virtual servers. These virtual servers must be assigned a Fast L4 profile. You do this on the Virtual Servers screen.

To configure virtual servers to mirror connections

1. Access the Configuration utility, using the cluster IP address of one of the clusters in the redundant system configuration.
2. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
3. Click the name of the virtual server that you want to mirror the connections that it processes.
   The configuration screen for that virtual server opens.

   Tip: *BIG-IP systems support intra-cluster mirroring only for virtual servers that reference a Fast L4 profile.*
4. From the **Configuration** list, select **Advanced**.
   The Configuration area expands to include additional settings.
5. To enable connection mirroring, click the **Connection Mirroring** check box.
6. Click **Update**.
7. If you are creating a redundant system configuration, on the menu bar, click **ConfigSync**.
   The ConfigSync screen opens.
8. Click the **Synchronize TO Peer** button.
   The system applies the configuration change that you made in this procedure to the peer cluster in the redundant system configuration.

Configuring virtual servers to mirror session persistence records

You can also configure a virtual server to mirror the session persistence records of the traffic the virtual server processes. To do this, you configure session persistence mirroring on a profile, and then associate the profile with the virtual server that you want to mirror the session persistence records of the traffic that it processes.

To configure virtual servers to mirror session persistence records

1. Access the Configuration utility, using the cluster IP address of one of the clusters in the redundant system configuration.
2. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The HTTP Profiles screen opens.
3. Click the name of the Persistence profile on which you want to enable persistence record mirroring.
   The configuration screen for that profile opens.

Note: *The system cannot mirror stateless persistence methods, such as Cookie Insert mode, because the persistence information is not maintained in a table on the VIPRION system; however, the persistence information is maintained after a failover because the cookie submitted by the client contains that information.*

4. Check the **Mirror Persistence** box.
5. Click **Update**.
6. Repeat steps 2 through 5 for each profile on which you want to enable persistence record mirroring.
7. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
8. Click the name of the virtual server that you want to configure to mirror session persistence records for the traffic that it processes.
   The configuration screen for that virtual server opens.
9. On the menu bar, click **Resources**.
10. From the **Default Persistence Profile** list, select the profile that you want to associate with the virtual server.

    Tip: *With the exception of Cookie Hash mode, cookie persistence does not require mirroring, because the system stores persistence information in a cookie on the client.*
11. From the **Fallback Persistence Profile** list, select the profile that you want the system to use if it cannot use the specified default persistence profile.
12. Click **Update**.
13. Repeat steps 7 through 12 for each virtual server that you want to configure to mirror session persistence records.
14. If you are creating a redundant system configuration, on the menu bar, click **ConfigSync**.
    The ConfigSync screen opens.
15. Click the **Synchronize TO Peer** button.
    The system applies the configuration change to the peer cluster in the redundant system configuration.

Configuring mirroring when performing system maintenance

Sometimes, you might need to perform maintenance on the active unit of a redundant system configuration while still maintaining in-process Fast L4 connections and persistence records. In this case, you must change the mirroring configuration before you perform the system maintenance.

As mentioned previously, there are two types of connection mirroring that you can configure on a redundant system configuration: intra-cluster and inter-cluster. During normal operation of an active/standby configuration, F5 recommends intra-cluster mirroring as the best practice for the processing of application traffic. Intra-cluster mirroring ensures that if a blade with active connections suddenly becomes unavailable, another blade in the same cluster can process those connections.

However, if you want to take the active unit of a redundant system configuration offline to perform system maintenance, you must first configure the system for inter-cluster mirroring. Inter-cluster mirroring ensures that if a cluster with active connections becomes unavailable, the peer cluster can process those connections.

There are a number of steps you must perform before, during, and after the maintenance window to ensure that the system processes traffic without interruption:

- **Configure mirroring IP addresses**
  First, you must configure each cluster with the IP addresses that you want the cluster to use for mirroring. For step-by-step instructions, see *Configuring connection mirroring addresses for a cluster*, on page 5-8.
- **Configure inter-cluster mirroring**
  Then you must configure inter-cluster mirroring on both clusters, beginning with the standby cluster. For step-by-step instructions, see *Configuring inter-cluster mirroring during a maintenance window*, on page 5-9.
- **Force the active cluster offline**
  When you are ready to perform maintenance on the active cluster, you then force the active cluster offline. For step-by-step instructions, see *Forcing an active cluster offline*, on page 5-10.
- **Configure mirroring on the newly-active cluster**
  Now that the newly-active cluster is processing application traffic, and the other cluster is offline, you must configure the active cluster to mirror connections to other blades in the same cluster. For step-by-step instructions, see *Configuring a cluster for intra-cluster mirroring*, on page 5-10.

- **Change cluster status**
  When your maintenance is complete, you have two options:
  - You can change the status of the cluster that you have been performing maintenance on to **standby**, and configure the cluster for intra-cluster mirroring. For step-by-step instructions, see *Manually changing the status of a cluster*, on page 5-11, and *Configuring a cluster for intra-cluster mirroring*, on page 5-10.
  - You can change the status of the cluster that you have been performing maintenance on to **active**. However, in order for the system to process connections without interruption, before you do this, you must configure the clusters for inter-cluster mirroring. Then, after the currently active cluster becomes the standby cluster, you must again configure the clusters for intra-cluster mirroring. For step-by-step instructions, see *Manually changing the status of a cluster*, on page 5-11, *Configuring inter-cluster mirroring during a maintenance window*, on page 5-9, and *Configuring a cluster for intra-cluster mirroring*, on page 5-10.

Configuring connection mirroring addresses for a cluster

When you configure inter-cluster mirroring for a redundant system configuration, you configure the IP addresses that you want each cluster to use for mirroring. On each cluster, you must configure the primary connection mirroring address for both the cluster and the peer cluster.

You must access each cluster separately to configure the mirroring addresses.

To configure the connection mirroring addresses for a cluster

1. Access the Configuration utility, using the cluster IP address of one of the clusters in the redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The System Fail-safe screen opens.
3. On the menu bar, click **Network Mirroring**.
   The Redundancy Properties screen opens.
4. In the Mirroring Address **Self** box, remove the notation **::**, and then type the static self IP address of VLAN **HA** on the cluster you are configuring.
5. In the Mirroring Address **Peer** box, remove the notation **::**, and then type the static self IP address of VLAN **HA** on the peer cluster.
6. Click the **Update** button.
7. Repeat steps 1 through 6 for the other cluster in the redundant system configuration.

Configuring inter-cluster mirroring during a maintenance window

When you want to perform maintenance on a redundant system configuration, you must configure the system for inter-cluster mirroring. This enables the system to maintain the long-lived connections and persistence records, which the virtual servers associated with Fast L4 profiles are processing, during the maintenance window.

It is very important to configure inter-cluster mirroring on the standby cluster, before you configure inter-cluster mirroring on the active cluster. That way, when you configure inter-cluster mirroring on the active cluster, the standby cluster is ready to process the mirrored connections.

After you configure inter-cluster mirroring on the active cluster, you can verify that the system is mirroring the connections and persistence records to the standby cluster. However, you must give the system time to incorporate the configuration change. If you are mirroring connections, wait 10 seconds before you check the system. If you are mirroring persistence records, wait 35 seconds before you check the system.

To configure inter-cluster mirroring for the clusters in a redundant system configuration

1. Access the Configuration utility, using the cluster IP address of the standby cluster in the redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The System Fail-safe screen opens.
3. On the menu bar, click **Network Mirroring**.
   The Redundancy Properties screen opens.
4. From the **Network Mirroring** list, select **Between clusters**.
5. Click **Update**.
6. After you configure the standby cluster for inter-cluster mirroring, only then do you configure the active cluster for inter-cluster mirroring. To do this, repeat steps 1 through 5 for the active cluster.
7. Check that the system is mirroring connections and persistent records between the clusters.

Forcing an active cluster offline

When you are ready to perform maintenance on the active cluster, you force the active cluster offline. When you force the active cluster offline, the standby cluster becomes active, and begins processing the connections that the previously active cluster was processing.

To force the active cluster offline

1. Access the Configuration utility, using the cluster IP address of the active cluster in the redundant system configuration.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The System Fail-safe screen opens.
3. On the menu bar, click **Failover**.
   The Failover screen opens.
4. From the **Failover Status** list, select **forced_offline**.
5. Click the **Update** button.
   The active cluster goes offline, and the standby cluster becomes active and begins to process application traffic.

Configuring a cluster for intra-cluster mirroring

Now that the newly active cluster is processing application traffic, and the other cluster in the redundant system configuration is offline, the system cannot mirror connections between the clusters. Therefore, you must modify the mirroring configuration on the newly active cluster so that it mirrors connections to the other blades in the cluster.

To configure intra-cluster mirroring for a cluster

1. Access the Configuration utility, using the cluster IP address of the cluster that you want to configure for intra-cluster mirroring.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The System Fail-safe screen opens.
3. On the menu bar, click **Network Mirroring**.
   The Redundancy Properties screen opens.
4. From the **Network Mirroring** list, select **Within cluster**.
5. Click **Update**.
   The system mirrors the connections that the cluster is processing to between the blades in the cluster.

Manually changing the status of a cluster

When you complete maintenance on a cluster that you have forced offline, you can change the status of the cluster to either **active** or **standby** mode, depending upon your needs.

When a redundant system configuration is configured in an active/standby configuration, and you change the status of a cluster to **active**, the peer cluster becomes the standby cluster. Likewise, when you change the status of a cluster to **standby**, the peer cluster becomes the active cluster.

To manually change the status of a cluster

1. Access the Configuration utility, using the cluster IP address of the cluster for which you want to change the status.
2. On the Main tab of the navigation pane, expand **System**, and click **High Availability**.
   The System Fail-safe screen opens.
3. On the menu bar, click **Failover**.
   The Failover screen opens.
4. From the **Failover Status** list, select either **active** or **standby**.
5. Click the **Update** button.
   The cluster status changes to the status you selected in step 4.

6

Configuring Advanced Routing Modules

- Introducing the advanced routing modules
- Platform and deployment considerations
- Configuring the advanced routing modules
- Configuring Route Health Injection

Introducing the advanced routing modules

As described in the ***TMOS® Management Guide for BIG-IP® Systems***, the BIG-IP® system has two route tables: a Traffic Management Microkernel (TMM) table for routing application traffic, and a host table for routing BIG-IP system management traffic. At a minimum, the TMM route table contains static routes that you specify, using the Routes screen of the Configuration utility.

In addition to adding static entries to the TMM route table, however, you can configure the BIG-IP system to add entries into the TMM and host route tables dynamically, that is, without user intervention. You can do this by licensing the optional set of advanced routing modules and then configuring them on the BIG-IP system. The ***advanced routing modules*** consist of industry-standard dynamic routing protocols that enable the BIG-IP system to establish relationships with other routers on a network for the purpose of sharing route information on a regular basis.

Supported protocols

An optional feature of the BIG-IP system, the advanced routing modules support several protocols. Table 6.1 lists the protocols included in the advanced routing modules, as well as related information.

Protocol name and version	Description	Daemon	IP version supported
BGP-4	Border Gateway Protocol (BGP) 4 with multi-protocol extension is a dynamic routing protocol for external networks that supports the IPv4 and IPv6 addressing formats.	bgpd	IPv4 and IPv6
IS-IS	Intermediate System-to-Intermediate System (IS-IS) is a dynamic routing protocol for internal networks, based on a link-state algorithm.	isisd	IPv4 and IPv6
OSPFv2	The Open Shortest Path First (OSPF) protocol is a dynamic routing protocol for internal networks, based on a link-state algorithm. OSPF is an alternative to the RIP protocol. OSPFv2 supports the IPv4 addressing format.	ospfd	IPv4
OSPFv3	This protocol is an enhanced version of OSPFv2 supporting the IPv6 addressing format.	ospf6d	IPv6
RIPv1/RIPv2	Routing Information Protocol (RIP) is a dynamic routing protocol for internal networks, based on a distance-vector algorithm (number of hops). RIPv1 and RIPv2 support the IPv4 addressing format.	ripd	IPv4
RIPng	This protocol is an enhanced version of RIPv2 supporting the IPv6 addressing format.	ripngd	IPv6

Table 6.1 *Supported protocols in the advanced routing modules*

Using any of the advanced routing modules, the BIG-IP system can:

- Dynamically add routes to the TMM and host route tables
- Advertise and redistribute virtual address routes to other routers

Note

To enable the Advanced Routing Modules feature, you might need to purchase an additional license key. For more information, contact F5 Networks.

For background information on configuring the advanced routing modules, see the Ask F5SM Knowledge Base (**http://support.f5.com**).

For more information

Throughout this guide, the text contains references to various ZebOS® Intelligent Network Software reference or configuration guides. These guides are available online, through the Ask F5SM web site, **http://support.f5.com**.

To locate a ZebOS guide

1. Log on to the Ask F5SM web site, **http://support.f5.com**.
2. In the **Search by Keywords** box, type a ZebOS guide title.
3. Press **Enter**.

The ZebOS guides that are available online are:

- ***NSM Command Reference***
- ***BGP Command Reference***
- ***IS-IS Command Reference***
- ***OSPF Command Reference***
- ***RIP Command Reference***
- ***Core Configuration Guide***

Platform and deployment considerations

Before configuring the advanced routing modules on the BIG-IP system, it is helpful to understand how the modules behave with respect to few key areas. They are:

- The VIPRION® platform
- Active/standby configurations
- BGP IPv6 next-hop address selection
- Route domains and route propagation

The remainder of this section describes these behaviors.

Understanding dynamic routing on a VIPRION system

If you have a VIPRION system, it is helpful to understand how the cluster environment affects the dynamic routing functionality.

Appearance as a single router

On a VIPRION system, configuration and operation of the dynamic routing subsystem behaves as if the cluster was a single router. This means that a cluster always appears as a single router to any peer routers regardless of the dynamic routing protocol being used.

From a management perspective, the VIPRION system is designed to appear as if you are configuring and managing the routing configuration on a single appliance. When you use the cluster IP address to configure the advanced routing modules, you transparently configure the primary blade in the cluster. The cluster synchronization process ensures that those configuration changes are automatically propagated to the other blades in the cluster.

Dynamic routing control plane redundancy

The dynamic routing subsystem takes advantage of the redundancy provided by the cluster environment of VIPRION chassis, for the purpose of providing redundancy for the dynamic routing control plane.Two key aspects of dynamic routing control plane redundancy are the operational modes of the enabled advanced routing modules, and the cluster's appearance to the routing modules as a single router.

Understanding operational modes

Enabled advanced routing modules run on every blade in a cluster in one of following operational modes: MASTER, STANDBY, or SLAVE. Table 6.2, on page 6-4 shows the operational modes for primary and secondary blades, on both the active cluster and the standby cluster.

Blade type	Active cluster	Standby cluster	Note
Primary	MASTER mode	STANDBY mode	The advanced routing modules: • Actively participate in dynamic routing protocol communication with peer routers • Maintain TMM and host route tables on all blades in the cluster
Secondary	SLAVE mode	SLAVE mode	The advanced routing modules: • Do not transmit any dynamic routing protocol traffic • Track communication between a module and the peer routers, or wait for transition to MASTER or STANDBY mode

Table 6.2 *Operational modes for advanced routing modules per blade type*

You can display the current operational mode from within the IMI shell (**imish**) using the command **show state**.

As described in Table 6.2, the advanced routing modules operate in the MASTER mode on the primary blade of the active cluster, in STANDBY mode on the primary blade of the standby cluster, and in SLAVE mode on all secondary blades.

In MASTER and STANDBY modes, the advanced routing modules actively participate in dynamic routing protocol communication with peer routers and maintain TMM and host route tables on all blades in the cluster. All routes learned by way of dynamic routing protocols on the primary blade are (in real-time) propagated to all secondary blades. The difference between MASTER and STANDBY mode is in the parameters of advertised routes, with the goal to always make the active unit the preferred next hop for all advertised routes.

In SLAVE mode, advanced routing modules do not transmit any dynamic routing protocol traffic. Depending on dynamic routing protocol, modules either track dynamic routing protocol communication between the advanced routing module operating in MASTER mode and peer routers, or simply wait for transition to MASTER or STANDBY mode.

Transition from SLAVE to MASTER or STANDBY mode takes advantage of standard dynamic routing protocol graceful restart functionality. For more information, see *Graceful restart considerations*, on page 6-4.

Graceful restart considerations

The ***graceful restart*** function allows the dynamic routing protocol control plane to move from one blade to another without disruption to traffic. Graceful restart is enabled for most supported protocols and address families by default. (See the following note.)

◆ Note

Graceful restart in BGP is not supported for IPv6 peers.

To operate successfully, the graceful restart function must be supported and enabled on all peer routers with which the VIPRION system exchanges routing information. If one or more peer routers does not support graceful restart for one or more enabled dynamic routing protocols, a change in the primary blade causes full dynamic routing reconvergence and probably traffic disruption. The traffic disruption is caused primarily by peer routers discarding routes advertised by the VIPRION system.

Complete forwarding information (TMM and host route tables) is always preserved on VIPRION systems during primary blade changes, regardless of support for graceful restart on peer routers.

Runtime monitoring of dynamic routing in individual blades

Startup configuration (in the **/config/ZebOS.conf** file) is automatically copied to all secondary blades, and the new configuration is loaded when the running configuration is saved on the primary blade. You can display the running configuration on both primary and secondary blades.

The runtime state (displayed using the IMI shell **show** commands) can be used on both the primary and secondary blades; however, some information displayed on secondary blades might differ from the information on the primary blade. Information displayed on the primary blade should be used for troubleshooting purposes, because only the primary blade both actively participates in dynamic routing communication and controls route tables on all blades.

In particular, the following information is not present or not correct on secondary blades:

- Configuration and status of BGP, OSPF version 3, and RIP (all versions)
- Status of OSPF version 2
- ZebOS forwarding information base (FIB)

Understanding dynamic routing in active/standby configurations

When deploying the advanced routing modules in a redundant system configuration, there are some general topics to consider, as well as considerations for systems using OSPF.

Design principles

You should consider the following facts when implementing the advanced routing modules on an active/standby system:

- Each unit of a redundant system configuration must be configured as a separate router. This means that:
  - The advanced routing modules must be configured separately on each unit of the redundant system.
  - For protocols that include the router ID attribute, each unit of the redundant system configuration must have a unique router ID.

- When using BGP, RIP, or IS-IS, both units of the redundant system automatically advertise their shared, floating self IP address as the next hop for all advertised routes. This ensures that peer routers use the shared self IP address as the next hop for *all* routes advertised by the BIG-IP system.
- When you have configured Route Health Injection (RHI), only the active unit advertises routes to virtual addresses. For more information on configuring RHI, see *Configuring Route Health Injection*, on page 6-14.

OSPF behavior in active/standby configurations

To guarantee correct operation in redundant active/standby configurations, the BIG-IP system automatically changes the runtime state of the OSPF subsystem. The purpose of a runtime state change is to modify advertised link state information in way that makes the standby unit the least preferred next-hop router. The next hop IP address in this case is the self IP address of the advertising router.

Figure 6.3 describes the changes that the BIG-IP system makes to the runtime state.

Module	Description of runtime change
OSPFv2	The OSPF interface cost is increased on all interfaces to the maximum value when the unit is in standby mode.
	All external type 2 Link State Advertisements (LSAs) are aged out.
OSPFv3	The OSPF interface cost is increased on all interfaces to the maximum value.

Table 6.3 *Description of OSPF runtime state changes*

The BIG-IP system reverses the runtime state changes after the unit transitions to active mode. You can see the effect of these changes by displaying OSPF interface status and listing the OSPF link state database.

To display OSPF interface status

Type the following command in the IMI shell:

```
show ip ospf interface
```

To list the OSPF link state database

Type the following command in the IMI shell:

```
show IP ospf database external self-originate
```

For information on how to use the IMI shell to configure an OSPF routing module, see the ZebOS ***OSPF Command Reference*** guide on **http://support.f5.com**.

BGP IPv6 next-hop address selection

For IPv6 addressing, the BGP module allows you to advertise one or two next-hop addresses for each route. Selection of addresses that are advertised to a peer depends on the following:

- The addresses configured on the interface over which a connection to the peer is established
- The local address used for the connection to the peer
- A multi-hop configuration (that is, whether a peer is connected directly or indirectly)

Table 6.4 shows several practical combinations of configuration parameters for selecting next-hop addresses, and shows the resulting set of advertised addresses.

Link-local autoconf. (LL-A)	Link-local (LL)	Link-local floating (LL-F)	Global (G)	Global floating (G-F)	EBGP multihop	Advertised nexthop addresses
P						LL-A
X	P					LL
X	P	X				LL-F
X	P		X			G,LL
X			P			G, LL-A
X	X	X	P		X	G
X	X	X	P			LL-F
X	P	X	X			LL-F
X			P	X		G-F
X	P		X	X		GF, LL
X	X		P	X		GF
X	P	X	X	X		LL-F
X	X	X	P	X		GF-F

P = peering address **X** = configured address

Table 6.4 *BGP configuration parameters for selecting next-hop addresses (IPv6 only)*

Considerations for route domains and route propagation

If you are using the route domains feature, you can implement dynamic routing, but only for route domain **0**. Route domain **0** is the default route domain on the BIG-IP system and the only route domain of which the advanced routing modules have knowledge. For this reason, you should consider the following facts:

- All dynamic routing peers must be connected to networks in the default route domain (route domain **0**).
- You can implement non-default route domains and simultaneously use dynamic routing in the default route domain. However, the advanced routing modules can use and advertise only those objects (routes, self IP addresses, and virtual addresses) that pertain to the default route domain.

Propagating dynamic routes

The advanced routing modules acquire knowledge of routes from other routers and present these routes to the **nsm** daemon. The **nsm** daemon then determines whether or not to communicate these routes to other BIG-IP system processes, based on criteria such as the information stored in a module's configuration data.

On the BIG-IP system, dynamically-learned routes must be propagated to the separate TMM route table in order for the system to use them for directing application traffic. To accomplish this, the BIG-IP system includes a daemon named **tmrouted**. The **nsm** daemon communicates with the **tmrouted** daemon about any new, modified, or deleted dynamic routes. When learning of such changes, the **tmrouted** daemon communicates with the **mcpd** daemon, and the **mcpd** daemon updates the TMM and host route tables.

Configuring the advanced routing modules

Regardless of the specific advanced routing module you want to configure, the procedure for configuring an advanced routing module on a BIG-IP system is similar. In general, you must complete a few basic tasks.

Table 6.5 shows the tasks to perform, along with the relevant configuration tool. Also included are references to additional information.

Task	Configuration tool	For more information
Enable and disable the modules	**zebos** command	• *Enabling an advanced routing module*, on page 6-10 • *Disabling an advanced routing module*, on page 6-10
Create a configuration for the enabled module	IMI shell	*Using the IMI shell*, on page 6-11
Start, stop, and restart the modules, Display status of the modules	**bigstart** and **zebos** commands	• *Starting the advanced routing modules*, on page 6-11 • *Stopping the advanced routing modules*, on page 6-12 • *Restarting advanced routing modules*, on page 6-12 • *Displaying the status of advanced routing modules*, on page 6-12

Table 6.5 *Overview of tasks for configuring the advanced routing modules*

Note

Before configuring an advanced routing module, see the section ***Understanding dynamic routing on a VIPRION system****, on page 6-3.*

Note that typing the **zebos** command with no arguments produces the message in Figure 6.1 about syntax usage.

```
usage: zebos (enable|disable) bgp|isis|ospf|ospf6|rip|ripng

  zebos check
  zebos (cmdd command1)(,command2)
```

Figure 6.1 *Syntax for the* ***zebos*** *command*

Enabling an advanced routing module

If you have recently licensed the routing modules but have not started them yet, the first step is to enable the relevant module. When you enable a module, the system starts the corresponding protocol daemon. If this is the first enabled module, it also starts the required common daemons (**nsm** and **imi**). Daemons supporting enabled modules are automatically started when the system starts up.

Note

The system does not propagate enabled modules at runtime to the peer unit in a redundant configuration. For the OSPFv2 and OSPFv3 modules, this could have adverse effects. You can prevent these effects by always enabling a module on the active unit, and then synchronizing the configuration to the standby unit.

To enable an advanced routing module

At the BIG-IP system prompt, type the **zebos** command and specify a routing module, using this syntax:

```
zebos enable <protocol>
```

where **<protocol>** represents **bgp**, **isis**, **ospf**, **ospf6**, **rip**, or **ripng**.

Disabling an advanced routing module

When you disable a routing module, note the following facts:

- The relevant daemon is stopped immediately (and will not start with the next reboot). If this module is the last enabled module, the common daemons (**nsm** and **imi**) are also stopped.
- After a module is disabled, the relevant configuration is removed from the runtime configuration (as displayed by the IMI shell command **show running**), but it remains in the configuration file (**/config/ZebOS.conf**) until you save the running configuration (using the command **write file**).

Note

The system does not propagate disabled modules at runtime to the peer unit in a redundant configuration. For the OSPFv2 and OSPFv3 modules, this could have adverse effects. You can prevent these effects by always disabling a module on the standby unit, and then synchronizing the configuration to the active unit.

To disable an advanced routing module

To permanently disable a routing protocol, type the following at the BIG-IP system prompt:

```
zebos disable <protocol>
```

where **<protocol>** represents **bgp**, **isis**, **ospf**, **ospf6**, **rip**, or **ripng**.

To disable the entire dynamic routing subsystem, type the following at the BIG-IP system prompt:

```
bigstart disable tmrouted
```

Using the IMI shell

You configure the advanced routing modules using the IMI shell, which is a command-line shell. Using the ***IMI shell***, you can configure every aspect of dynamic routing. The IMI shell provides command completion and in-line help for configuration and management commands.

Note

*If you are accustomed to using the **vtysh** command to configure the advanced routing modules, you can continue to do so. The IMI shell is backward-compatible with the VTY Shell.*

The IMI shell also allows you to monitor operation of the modules at runtime. You can list the current state of the route table, display the internal state of individual routing modules, and so on.

Important

*Always use the IMI shell to configure an advanced routing module, rather than directly updating the **ZebOS.conf** file with a text editor. The IMI shell provides error-checking and command completion, two features that help to ensure correct configuration of the module. Note that a runtime configuration created with the IMI shell is lost when routing modules restart, unless it is saved to the configuration file using the IMI shell **write file** command. Note, too, that F5 Networks provides no technical support for configurations created by modifying the **ZebOS.conf** file directly.*

Starting the advanced routing modules

Once you have enabled one or more modules, the BIG-IP system automatically starts the enabled module whenever you perform a reboot.

If you need to manually start the daemons, you can type the following command at the BIG-IP system command-line prompt:

```
bigstart start tmrouted
```

This command starts all daemons necessary to run enabled routing modules.

Stopping the advanced routing modules

To temporarily stop all dynamic routing protocols and remove all dynamic routes from the system, you can use this command:

```
bigstart stop tmrouted
```

To stop all dynamic routing protocols permanently, see *To disable an advanced routing module*, on page 6-11.

Restarting advanced routing modules

To restart all advanced routing modules, type the following command at the BIG-IP system command-line prompt:

```
bigstart restart tmrouted
```

This command is useful for reloading a configuration from the **ZebOS.conf** file. The command basically performs the **bigstart stop** and **bigstart start** actions in sequence. As a result, the command causes all dynamic routes to temporarily disappear from the system.

Displaying the status of advanced routing modules

To display the current status of the advanced routing modules, type the following command at the BIG-IP system command-line prompt:

```
bigstart status tmrouted
```

If the advanced routing modules have been started, the command produces output similar to the following:

```
tmrouted run (pid <number>) <time> minutes <n> starts
```

The **<number>** variable is the process identification number (PID) of the supervisory daemon **tmrouted**. The **<time>** variable is the time since the modules have been started. The **<n>** variable can be used if **tmrouted** was restarted since the last reboot.

If the modules have not been started (or have been stopped), the command displays the following:

```
tmrouted down <time> seconds, normally up
```

In this case, the **<time>** variable represents the time since the modules were stopped, or since the system booted.

To query the status of individual daemons, type following command at the BIG-IP system command-line prompt:

```
zebos check
```

If the modules have been enabled and started, this command lists every currently-running routing daemon, in the format shown in Figure 6.2.

```
nsm      is running [25939]
imi      is running [25938]
bgpd     is running [25940]
ospfd    is running [25941]
ospf6d   is running [25942]
```

Figure 6.2 *Sample output showing running daemons with process IDs.*

If the advanced routing modules have not been started, the **zebos check** command produces no output.

Configuring Route Health Injection

In addition to dynamically adding learned routes to the TMM route table, the advanced routing modules can redistribute routes to any virtual addresses that you have defined on the BIG-IP system. Advertising routes to virtual addresses based on the status of attached virtual servers is known as ***Route Health Injection (RHI)***.

◆ Note

If you are using the route domains feature, there are additional considerations for route advertisement. For more information, see ***Considerations for route domains and route propagation****, on page 6-8.*

Advertising routes to virtual addresses

The system creates one advertised route for every virtual address that:

- Has route advertisement enabled.
- Is considered to be up based on the status of virtual servers using the address.

An advertised destination depends on the virtual address and its netmask.

Figure 6.3 shows a sample entry in the **bigip.conf** file. This entry advertises a route to address **10.1.1.1/32**.

```
virtual address 10.1.1.1 {
   mask 255.255.255.255
   route advertisement enable
}
```

Figure 6.3 *Sample configuration for route advertisement*

◆ Tip

You can view the above configurations using the ***bigpipe virtual address list*** *command.*

Note that virtual address objects are created automatically, based on configured virtual server objects. For example:

- For a virtual server with a destination address of **10.1.1.1:80**, the BIG-IP system automatically creates the virtual address **10.1.1.1**, with a netmask of **255.255.255.255**.
- For a virtual server with a destination address of **10.2.0.0:any** and a netmask of **255.255.0.0**, the BIG-IP system automatically creates the virtual address **10.2.0.0**, with a netmask of **255.255.0.0**.

Specifying conditions for advertising routes to virtual addresses

Advertisement of virtual addresses can depend on the status of virtual servers using the address as their destination (that is, virtual servers that are attached to that address). A virtual address can be advertised as follows:

- **When any virtual server is available**
  When set to this value, the BIG-IP system advertises the virtual address when any one of the virtual servers associated with the virtual address is available. This is the default value.
- **When all virtual servers are available**
  When set to this value, the BIG-IP system advertises the virtual address only when all of the virtual servers associated with the virtual address are available.
- **Always**
  When set to this value, the BIG-IP system always advertises the virtual address, regardless of the availability of virtual servers associated with the virtual address.

Note

*A virtual server is considered to be available when it is in the **UP** (green) or **UNCHECKED** (blue) state.*

To advertise a virtual address

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. On the menu bar, click **Virtual Address List**.
   This displays the list of virtual addresses.
3. In the Address column, click the IP address that you want to advertise.
   This opens the Virtual Addresses screen.
4. From the **Advertise Routes** list, select a value or retain the default value.
   For more information on this setting, see *Specifying conditions for advertising routes to virtual addresses*, on page 6-15.
5. For the **Route Advertisement** setting, check the box.
   This enables route advertisement for this virtual address.
6. Click **Update**.

You can also advertise a virtual address using the **bigpipe virtual address** command in the **bigpipe** utility. For more information, see the ***Bigpipe Utility Reference Guide***.

Once you have advertised a virtual address, you can then configure the advanced routing modules to redistribute that address to other routers.

Redistributing routes to virtual addresses

If you have enabled the Route Health Injection (RHI) feature, you can redistribute routes for virtual addresses to other devices on the network, using any supported advanced routing modules.

You must explicitly configure each advanced routing module to redistribute routes for advertised virtual addresses, to ensure that other routers on the network learn these routes. Also, for purposes of redistribution, the advanced routing modules consider any route generated through RHI to be a host route.

You can use route maps to filter or modify an advertised route. For more information, see *Fine-tuning route redistribution using route maps*, following.

◆ Note

You must configure route redistribution for IPv4 addresses separately from that of IPv6 addresses. This is true for all advanced routing modules, including those that pertain to both IPv4 and IPv6 formats such as BGP.

Figure 6.4 shows a sample entry in the OSPF configuration. When you add this statement to the OSPF configuration, using the IMI shell, the BIG-IP system redistributes the route to the virtual address that was previously advertised in Figure 6.3.

```
router ospf
   redistribute kernel
```

Figure 6.4 *Sample configuration entry for redistributing advertised routes*

When adding this entry into the configuration data, you can optionally specify a **route-map** reference that specifies the route map to use for filtering routes prior to redistribution. Figure 6.5 shows a sample entry for route filtering in the advanced routing module configuration.

```
redistribute kernel route-map external-out
```

Figure 6.5 *Sample configuration entry for filtering advertised routes*

Fine-tuning route redistribution using route maps

Route maps provide an extremely flexible mechanism for fine-tuning redistribution of routes using the advanced routing modules. For a full description of all available route map commands, see the ZebOS ***NSM Command Reference*** guide. The ZebOS ***Core Configuration Guide*** provides examples of using route maps with all supported dynamic routing protocols. Both of these guides are available on the Ask F5 Knowledge Base web site **http://support.f5.com**.

On the BIG-IP system, you typically use route maps to filter redistributed host routes. Figure 6.6 shows an OSPF configuration that uses a route map to limit the redistribution of host routes to only those routes with a destination in the **10.10.10.0/24** subnet. (Note that the line numbers shown in the example are not part of the actual configuration, but are included in the figure to identify each line for the explanation that follows.)

```
1 router ospf
2   redistribute kernel route-map virtual-addresses-out
3 ...
4 access-list virtual-addresses permit 10.10.10.0/24
5
6 route-map virtual-addresses-out permit 10
7   match ip address virtual-addresses
```

Figure 6.6 *Using a route map to filter routes in an OSPF configuration*

The explanation of each pertinent line in the figure is as follows:

- Line 1 and 2: Configures redistribution of host routes using the OSPF module with the route map **virtual-addresses-out** applied.
- Line 4: Defines the access list **virtual-addresses**, which permits all addresses in the **10.10.10.0/24** subnet.
- Line 6: Defines an entry with the serial number **10** of the route map **virtual-addresses-out**. The entry causes the module to accept all route-matching-specified criteria (due to the **permit** directive in the definition.
- Line 7: Configures matching criteria for the route map entry. The entry matches all routes whose destination prefix is accepted by the access list **virtual-addresses**. An access list accepts a prefix if the prefix is equal to, or fully contained in, the prefix specified in the access list. For example, a route with the destination **10.10.10.1/32** matches the access list, and the route with the destination **10.10.0.0/16** does not match the access list.

Glossary

advanced routing modules

The advanced routing modules constitute an optional feature that provides support for industry-standard dynamic routing protocols. The protocols that the advanced routing modules support are: BGP, IS-IS, OSPFv2, OSPFv3, RIPv1, RIPv2, and RIPng.

blade

A blade fits into a slot of a VIPRION® system chassis. Multiple blades within a chassis can work together to process traffic from a single virtual server.

chassis

The chassis for the VIPRION system is a multi-slot frame that significantly reduces the amount of rack space required for a system.

cluster

A cluster is a group of slots on the VIPRION system chassis. When you insert blades into the slots of a cluster, the blades function as a single system to process application traffic.

clusterd

The **clusterd** daemon initializes the slots in a chassis to form a cluster.

cluster IP address

A cluster IP address is a floating management IP address associated with the primary slot in a cluster. When the blade in the primary slot is removed or becomes unavailable, another slot in the cluster automatically becomes the primary slot, and the cluster IP address becomes associated with the new primary slot. You use this IP address to manage the cluster. See also *cluster* and *Configuration utility*.

cluster member

A cluster member is a slot on the VIPRION system chassis that you specify as being part of a cluster when you create the cluster.

cluster member IP address

A cluster member IP address is a management IP address associated with a slot on the VIPRION system chassis. When you replace the blade in a slot, the system automatically associates the new blade with the slot-specific IP address. See also *slot*.

cluster member status

A cluster member can have one of the following statuses: **Available**, **Unavailable**, or **Removed**.

cluster status

A cluster can have one of the following statuses: **Active**, **Standby**, **Offline** (a condition in which a cluster cannot become active), **Forced Offline** (a condition in which an Administrator initiates the Offline status of a cluster). See also, *cluster*.

cluster synchronization

Cluster synchronization propagates Layer 2 and local traffic configuration data from the primary blade to the secondary blades in a cluster.

Configuration utility

The Configuration utility is the browser-based application that you use to configure the BIG-IP system.

connection mirroring

When you enable mirroring on a virtual server, the VIPRION system mirrors connections either to every blade in the cluster, or between clusters. When you enable persistence for the virtual server, the system also mirrors the connections' persistence records to every blade in the cluster, or between clusters. Connection mirroring prevents any interruption in service when either a blade on a running system becomes unavailable, or a cluster in a redundant system configuration becomes unavailable. See also *persistence mirroring*.

failover

Failover is the process whereby a standby unit in a redundant system takes over when a software failure or a hardware failure is detected on the active unit.

fail-safe

Fail-safe is the ability of a VIPRION system to monitor certain aspects of the system or network, detect interruptions, and consequently take some action.

gateway fail-safe

Gateway fail-safe monitors traffic between the active VIPRION system and a pool containing a gateway router, thereby protecting the system from a loss of an internet connection by triggering a failover when a gateway router is unreachable for a specified duration.

inter-cluster mirroring

Inter-cluster mirroring is a mode of operation where the VIPRION system mirrors the connections being processed by a cluster to the other cluster in the redundant system configuration. Inter-cluster mirroring is only possible when both clusters have the identical number of blades. Inter-cluster mirroring is not to be confused with *intra-cluster mirroring*. These two types of mirroring are mutually exclusive.

interface

Interfaces are the physical ports on the slots of a VIPRION system. The status of an interface is one of the following: **Down**, **Up**, or **Unpopulated**.

intra-cluster mirroring

Intra-cluster mirroring is a mode of operation where the VIPRION system mirrors the connections being processed by a blade to other blades in the same cluster. Intra-cluster mirroring is not to be confused with *inter-cluster mirroring*. These two types of mirroring are mutually exclusive. Intra-cluster mirroring is the best practice.

management interface

The management interface is a special port on each slot of a VIPRION system chassis. The system uses this port to manage administrative traffic. The management interface does not forward user application traffic, such as traffic slated for load balancing.

missing blade

A missing blade is a blade that is either offline or that has been removed from a slot that is configured as a cluster member. See also *blade*.

network failover

Network failover is the process by which an active cluster uses its peer's failover IP address when failing over to that peer.

network mirroring

Network mirroring is also known as connection mirroring. See also *connection mirroring*.

pending interface

A pending interface is a slot that has been configured as a cluster member, but in which either no blade is plugged, or the blade is offline. See also *interface*.

persistence mirroring

Persistence mirroring is a mode of operation where the VIPRION system mirrors persistence records to other blades in the same cluster, or to the other cluster in a redundant system configuration. See also *inter-cluster mirroring* and *intra-cluster mirroring*.

primary blade

A primary blade is the blade in a cluster that is currently accepting management and application traffic requests, and dispersing the workload to secondary blades in the cluster. When the primary blade is removed or becomes unavailable, another blade in the cluster automatically becomes the primary blade. See also *secondary blade*.

primary slot

A primary slot is any slot in a VIPRION system chassis that contains a primary blade.

secondary blade

A secondary blade is any blade in a cluster to which the primary blade in the cluster disperses workload.

slot

A slot is a part of the chassis that holds a blade. A slot can be a member of a cluster if configured as such.

state mirroring

State mirroring is a mode of operation where any state that is not associated with a specific connection is mirrored to other blades in the same cluster. Currently, this only includes syncookie secrets.

Index

H

I

L

M

N

O

P

R

S

T

U

V

W