New and Updated Commands

BIG-IP systems, version 9.0

MAN-0133-00

Product Version

This manual applies to version 9.0 of BIG-IP® Local Traffic Manager™, BIG-IP® Load Balancer Limited™, and BIG-IP® SSL Accelerator™.

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.

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

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

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

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

This product includes software developed by Jared Minch.

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

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

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

This product includes RRDtool software developed by Tobi Oetiker (http://www.rrdtool.com/index.html) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation <http://www.apache.org/>.

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (http://www.nominum.com).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

Table of Contents

1

Introducing BIG-IP System Commands

- Purpose of this guide
- Introducing the BIG-IP system command line interface

Purpose of this guide

The release of the BIG-IP version 9 system includes several changes and enhancements to the BIG-IP command line interface. The command line interface is one way that customers can configure and manage the BIG-IP system.

If you are an existing BIG-IP system user who has used the command line interface in previous versions of the product, and you want to continue using it, this guide can provide an overview of the changes. This guide is therefore targeted for existing BIG-IP system users familiar with the BIG-IP command line interface, and provides the following information:

- Summaries of all network management, system management, and local traffic management utilities and commands available from the command line
- Summaries of all network management, system management, and local traffic management tasks that you can perform to manage the BIG-IP system
- Procedures that have changed due to the release of new or modified commands, or due to commands becoming obsolete. Note that if a procedure requires the same commands as in previous versions of the BIG-IP system but the detailed syntax of the commands has changed, then those procedures are not included in this guide. Instead, this guide refers the reader to the online man pages for correct syntax information.

If the specific commands or combinations of commands you once used to perform a command line task in version 4.5+ are the same in BIG-IP version 9, the procedures for those tasks are not included here. Instead, you can refer to the ***BIG-IP Reference Guide*** (version 4.5) for that information. However, if commands or combinations of commands are different in version 9 from those in version 4.5+, then those tasks are included in this guide.

For example, this guide includes the procedure for configuring SSL certificate-based authorization using a remote LDAP server, because the commands have changed. Instead of using the 4.5+ commands **bigpipe authz** and **bigpipe proxy**, you now use the commands **bigpipe auth** and **bigpipe profile**.

Also included in this guide are command line procedures for managing any new features in version 9. Thus, for example, the guide includes the command line procedures for implementing HTTP compression and sever-side connection pooling.

Introducing the BIG-IP system command line interface

The BIG-IP traffic management system is a powerful combination of hardware and software elements, designed to meet your traffic-management needs in the most efficient, scalable, reliable, and secure ways possible. Although the primary tool for managing the BIG-IP system is the browser-based Configuration utility, there are other tools available that are command-line-based. That is, there are commands and utilities that you can either type at a command-line prompt (such as the BIG-IP system prompt or the LINUX operating system prompt), or use within scripts such as iRules™.

While some of these utilities and commands are provided as part of the BIG-IP system, others are industry-standard tools that you can use to further enhance the power of the BIG-IP system.

Understanding command-line utilities and tools

There are several command-line utilities and tools that you can use to manage the BIG-IP system:

- **The config utility**
  You use the **config** utility to define the IP address, network mask, and gateway for the management (**MGMT**) port, when you initially set up your BIG-IP system.
- **The bigpipe utility**
  The **bigpipe** utility is a set of commands that you can use to configure elements of the BIG-IP system such as load balancing pools and virtual servers. Using **bigpipe** commands, you can manage the BIG-IP base network and system, and you can control local application traffic to suit your exact needs.
- **The bigtop utility**
  The **bigtop** utility is a command that provides statistical monitoring, and displays connections and throughput. You can set a refresh interval and specify a sort order for this statistical information.
- **The bigstart command**
  With the **bigstart** command, you can start, stop, restart, and check the status of various daemons, such as **snmpd**.

The industry-standard tools that you can also use to manage the BIG-IP system are:

- **The Syslog daemon**
  The **Syslog** daemon is a LINUX operating system daemon that tracks system events and stores them in log files. This daemon can track not only LINUX system events, but BIG-IP system events, too. The system stores the **Syslog** configuration file in the **/etc/syslog.conf** directory and stores the log output in the files in the **/var/log** directory.

- **The Tools Command Language (Tcl) programming language**
  The Tools Command Language (Tcl) programming language is an industry-standard programming language that you can use to create BIG-IP iRules. ***iRules*** are scripts you can write to direct and manipulate the way that the BIG-IP system manages application traffic.
- **The openssl utility**
  A component of the industry-standard OpenSSL toolkit, the **openssl** utility is a set of commands that perform various cryptographic functions, such as generating SSL certificates and keys.

Important

*This document does not provide the complete syntax for **bigpipe** commands. For complete syntax information, see the online man page for each **bigpipe** command.*

Using the bigpipe utility

You can display information on each **bigpipe** command, using a set of online man pages included in the BIG-IP system. These man pages provide all of the detailed syntax you need to use the **bigpipe** commands.

Displaying man pages

The BIG-IP product distribution includes a complete set of online man pages for the **bigpipe** commands. To display a man page for a **bigpipe** command, type the **man** command, followed by the command name, at the command line prompt. For example, to display the syntax for the **bigpipe vlan** command, type:

```
man vlan
```

An alternative way to display a man page is to type the keyword **help** after a **bigpipe** command. For example, you can type:

```
bigpipe vlan help
```

In some cases, a **bigpipe** command might not have an individual man page. In this case, you can display a master help page by simply typing:

```
bigpipe help
```

Understanding syntax conventions

To help you use the **bigpipe** commands, Table 1.1 explains the conventions that appear in the syntax section of the **bigpipe** man pages.

Item in text	Description
\	Indicates that syntax continues to the next line without typing a line break.
< >	Indicates that you should enter text for the enclosed item. For example, if the command syntax shows **<your name>**, type your name.
|	Separates alternate options for a command.
[]	Indicates that syntax inside the brackets is optional.
...	Indicates that you can type a series of items.

Table 1.1 *Command line conventions*

Tip

*When using the **bigpipe** command, you can either type the entire command at the command-line prompt, or you can simply type the letter **b**. For example, to configure a load balancing pool, you can type this command:* ***b pool <pool_name> <arguments>***

For more information

In addition to this guide, you can find information about the command line interface in the following locations:

- **Online man pages**
  The BIG-IP product includes a complete set of online man pages for the commands that make up the **bigpipe** utility. For more information see *Using the bigpipe utility*, on page 1-3.
- **HTML command line interface guide on CD**
  This is a printable collection of the **bigpipe** man pages included with the version 9 product.
- **The LINUX Syslog daemon man page**
  This man page is included with the standard set of LINUX operating system man pages.
- **User-supplied third-party Tcl reference books**
  Various third-party reference books on the Tcl programming language are available that you can use to write iRules for managing local application traffic.

2

Managing the Base Network

- Customizing the BIG-IP base network
- Summarizing the bigpipe commands
- Performing network management tasks

Customizing the BIG-IP base network

Before you configure a BIG-IP system to manage local application traffic, you must use the Setup utility to configure what is referred to as a base network configuration. The base network components are:

- Interfaces
- Routes
- Self IP addresses
- Packet Filters
- Trunks (802.3ad Link Aggregation)
- Spanning Tree Protocol (STP)
- VLANs and VLAN groups
- ARP

Once you have configured base network components using the Setup utility, you are free to customize the configuration of those components. To customize the configuration of your base network components, you can use the browser-based Configuration utility, or you can use the **bigpipe** utility.

Summarizing the bigpipe commands

The **bigpipe** utility that is provided with the BIG-IP system includes a number of commands designed to help you customize the BIG-IP base network configuration.

Table 2.1 provides a concise listing of the individual **bigpipe** commands that you can use to manage the BIG-IP network. For more details on these commands, see the online man pages.

bigpipe Command	Description
help	Displays online help for an individual **bigpipe** command. For example, you can display information on the **bigpipe vlan** command by typing **b vlan help**. For more information, see Chapter 1, *Introducing BIG-IP System Commands*.
arp	Creates static ARP addresses, and lists static and dynamic ARP addresses.
failover	Sets the BIG-IP system as active or standby.
global	Sets global variable definitions.
interface	Sets options on individual interfaces.
load	Loads the BIG-IP system configuration and resets it.
self	Assigns a self IP address for a VLAN or interface.
stp	Implements spanning tree protocol (STP).
trunk	Aggregates links to form a trunk.
vlan	Defines VLANs, VLAN mappings, and VLAN properties.
vlangroup	Defines VLAN groups.

Table 2.1 *bigpipe commands for BIG-IP network management*

Performing network management tasks

Network management tasks are tasks that you do to customize the base network, using the **bigpipe** utility that the BIG-IP system provides.

Table 2.2 lists the tasks you can perform to further customize the base network that you configured using the Setup utility. For each task, the table shows the commands or utilities you use to perform that task.

Important

The command syntax shown in Table 2.2 is not exhaustive. For each command, see the corresponding man page for the correct syntax.

Tasks to customize your base network configuration	Command or utility to use
Display status and settings for interfaces.	bigpipe interface show
Set the media type on an interface.	bigpipe interface <interface_name> media <media_type>
Set the duplex mode on an interface.	bigpipe interface <interface_name> duplex<duplex>
Create a VLAN.	bigpipe vlan <vlan_name> interfaces add <interface_list>
Rename a VLAN.	bigpipe vlan <old_vlan_name> rename <new_vlan_name>
Delete a VLAN.	bigpipe vlan <vlan_name> delete
Configure interfaces to accept traffic from multiple VLANS.	bigpipe vlan <vlan_name> interfaces add tagged <interface_list>
View entries in the L2 forwarding table.	bigpipe vlan fdb show
Add entries to the L2 forwarding table.	bigpipe vlan fdb <mac> interface <interface> add
Delete entries from the L2 forwarding table.	bigpipe vlan fdb <mac> delete
Create a VLAN group.	bigpipe vlangroup <vlangroup_name> vlans add <vlan_name>
Verify that L2 forwarding is enabled.	bigpipe vlangroup show
Prevent the forwarding of proxy ARP requests.	bigpipe vlangroup <name> proxy exclude <IP_list>
Assign a self IP address to a VLAN group.	bigpipe self <ip address> vlan <vlangroup_name>
Specify a value in seconds for downed links.	bigpipe db Failover.Standby.LinkDownTime <value>
Allow or disallow services, such as SSH, for each self IP.	bigpipe self <self IP_list> allow <service_list | all | none>

Table 2.2 *Tasks for customizing a base network*

Tasks to customize your base network configuration	Command or utility to use
Set the failover timeout, in seconds, for a VLAN.	bigpipe vlan <vlan_name> timeout <value>
Enable the failsafe.	bigpipe vlan <vlan_name> failsafe enable
View the interfaces mapped to all VLAN.	bigpipe vlan show
View the interfaces mapped to a VLAN.	bigpipe vlan <name> show
View the MAC addresses for the BIG-IP interfaces.	bigpipe interface show verbose
Set the MAC address to be shared by redundant units.	bigpipe vlan <vlan_name> mac masq <MAC_addr>
Add a self IP address to a VLAN.	bigpipe self <IP_address> vlan <name>
Add a route.	bigpipe route <ip_addr> vlan <name>/
Create a trunk.	bigpipe trunk <name> interface <interface_list> add
Manage Spanning Tree Protocol (STP).	See the man pages for the commands stp_instance and man stp.
View interfaces.	bigpipe interface <interface_name> show

Table 2.2 *Tasks for customizing a base network (Continued)*

The following two sections describe command line tasks that are different in version 9 systems.

Implementing packet filtering

Packet filters provide a level of access control by filtering packets from a client based on criteria that you specify. You can specify these criteria by configuring a packet filter's general properties, and by creating a packet filter rule.

To implement packet filtering

Enable packet filtering using the **bigpipe packet filter** command.

When using this command, you can specify a packet filter rule to provide access control, rate shaping, or logging.

Configuring routing

You can add or remove routes for the switch interfaces, as well as the management interface.

To add and configure routes

Use either the **bigpipe route** or the **bigpipe route mgmt** command, specifying a list of route keys and a resource (gateway IP address, pool name, VLAN name, or reject). For more information, see the **route** man page.

3

Managing the BIG-IP System

- Introducing BIG-IP system management
- Understanding BIG-IP system management tools
- Performing BIG-IP system management tasks

Introducing BIG-IP system management

The BIG-IP system includes several command-line tools that you can use to perform routine system management tasks such as creating and managing administrative user accounts, displaying traffic statistics, and managing BIG-IP units in a redundant system configuration.

With these tools, you can manage many parts of the system:

- The management port
- BIG-IP host name and IP address
- Global system properties
- High Availability
- User configuration archives
- System services (for example, SSH and HTTP)
- SNMP
- Logging
- qkview and tcpdump (diagnostic tools)
- Serial console
- Real-time statistics

For information on configuring the BIG-IP system to control local application traffic, see Chapter 4, *Managing Local Application Traffic*.

Understanding BIG-IP system management tools

The tools that you can use to manage the BIG-IP system are:

- **config** command
- **bigpipe** utility
- **bigtop** utility
- **bigstart** command
- **halt** command
- **reboot** command
- **hostname** command
- **printdb** command
- **ssh** and **scp** commands

Table 3.1 provides a concise listing of the commands that you use to manage the BIG-IP system. For more details on these commands, see the online man pages.

Command	Description
config Command	
config	Configures the IP address, network mask, and gateway on the management (**MGMT**) port. Use this command prior to licensing the BIG-IP system and do not confuse it with the **bigpipe config** command or the BIG-IP Configuration utility.
bigpipe Commands	
config	Synchronizes the **/config/bigip.conf** between the two BIG-IP units in a redundant system.
conn	Prints a list of current connections.
daemon	Manages the failover settings of various BIG-IP system daemons.
db	Loads configuration information into the bigdb database and displays bigdb information.
global	Sets global variable definitions and resets global statistics.
-h and help	Displays online help for **bigpipe** command syntax.
ha	Displays the HA (high availability) table.
http	Displays statistics related to HTTP traffic.
icmp	Displays statistics related to ICMP traffic.
ip	Displays statistics related to IP traffic.
load	Loads the BIG-IP system configuration, and resets it.

Table 3.1 *Commands for BIG-IP system management*

Command	Description
memory	Displays memory statistics.
merge	Loads a saved BIG-IP system configuration without resetting the current configuration.
mgmt	Modifies the settings of the management (**MGMT**) port.
nat	Resets statistics for network address translations (NATs).
node	Displays nodes, resets statistics for nodes, and removes nodes from service.
reset	Clears the BIG-IP system configuration and counter values.
save	Writes the current configuration to a file.
snat	Manages secure network address translations (SNATs).
unit	Displays the unit number assigned to a particular BIG-IP system.
verify	Parses the command line and checks syntax without executing the specified command.
version	Displays the **bigpipe** utility version number.
virtual	Displays status and statistical information for virtual servers, resets virtual server statistics, and removes virtual servers from service.
bigtop Commands	
bigtop	Displays real-time statistics.
Other Commands	
halt	Shuts down the BIG-IP software application.
bigstart	Restarts the SNMP agent **bigsnmpd**.
printdb	Prints the values of one or more entries in the bigdb database.
ssh and scp	Access command line interfaces on other SSH-enabled devices, and copy files to or from a BIG-IP system.

Table 3.1 *Commands for BIG-IP system management (Continued)*

Performing BIG-IP system management tasks

Table 3.2 lists the tasks that you can perform to maintain the BIG-IP system. For each task, the table shows the commands or utilities you use to perform that task.

Important

The command syntax shown in Table 3.2 is not exhaustive. For each command, see the corresponding man page for the correct syntax.

.

Tasks to manage your BIG-IP system	Command or utility to use
Modify the IP address, network mask, and management route of the designated management port for the BIG-IP system.	config or bigpipe mgmt
Modify the host name of the BIG-IP system.	Setup utility. Use the browser-based Configuration utility.
Modify the IP address of the BIG-IP system.	Setup utility. Use the browser-based Configuration utility.
Specify whether the BIG-IP system is a single device or part of a redundant pair.	Setup utility. Use the browser-based Configuration utility.
Modify the time zone of the BIG-IP system time.	Setup utility. Use the browser-based Configuration utility.
Change the password of the **root** account.	Setup utility. Use the browser-based Configuration utility.
Change the password of the **admin** account.	Use the browser-based Configuration utility.
Enable or disable the **support** account.	Use the browser-based Configuration utility.
Allow console access for a user.	Use the browser-based Configuration utility.
Specify the IP addresses from which an SSH user can access the BIG-IP system.	Use the browser-based Configuration utility.
Manage the web certificate-key pair for a server.	openssl utility. For more information, see Chapter 4, *Managing Local Application Traffic*.
Enable or disable the hardware monitor.	Use the browser-based Configuration utility.
Designate action in the event of an SSL Accelerator failure.	Use the browser-based Configuration utility.
Designate action in the event of a switch board failure.	Use the browser-based Configuration utility.
Designate action in the event of a BIG-IP daemon failure.	bigpipe daemon help
Designate action in the event of a VLAN failure.	bigpipe vlan help

Table 3.2 *System management tasks*

Tasks to manage your BIG-IP system	Command or utility to use
Manage configuration archives.	bigpipe config
Start or stop various BIG-IP system services.	bigstart help
View status/history of BIG-IP system services.	bigstart status
Start or stop the SNMP agent.	bigstart shutdown snmpd
Display performance statistics for the BIG-IP system, such as uptime and total number of connections.	bigpipe global
Reset statistics for one or all virtual servers.	bigpipe virtual <name> stats reset
Reset statistics for one or all nodes.	bigpipe node [<ip_address><service>] stats reset
Reset statistics one or all virtual ports.	bigpipe service [<service>] stats reset
Reset statistics for one or all SNATs.	bigpipe snat [<original_address>] stats reset
Reset statistics for one or all NATs.	bigpipe nat [<original_address>] stats reset
Reset reset statistics globally.	bigpipe global stats reset
Display connection information.	bigpipe conn
Display statistical information for a service	Use the browser-based Configuration utility.
Enable or disable a service.	Use the browser-based Configuration utility.
Display real-time statistics.	bigtop <options>
Power down the BIG-IP software application.	halt
Reboot the BIG-IP software application.	reboot
Reload the BIG-IP configuration.	bigpipe load
Display system properties such as host name, version, and CPU count.	hostname, bigpipe version, bigpipe global
Boot the BIG-IP system from the network on next boot.	bigpipe db Boot.NetReboot enable
Enable or disable the LCD System menu.	*Use the browser-based Configuration utility.*
Specify the time servers that the system uses to update the system time.	*Use the browser-based Configuration utility.*
Specify the name servers that the system uses to validate DNS lookups and resolve host names.	*Use the browser-based Configuration utility.*
Enable or disable VLAN-keyed connections.	bigpipe db Connection.VlanKeyed enable | disable

Table 3.2 *System management tasks (Continued)*

Tasks to manage your BIG-IP system	Command or utility to use
Specify that the system finds the maximum transmission unit (MTU) that it can send over a path without fragmenting TCP packets.	bigpipe mgmt route, bigpipe route, bigpipe vlan
Specify the percentage of memory usage at which the system stops allowing new connections.	bigpipe db Connection.AdaptiveReaper.Hiwater <num>
Specify the percentage of memory usage at which the system begins silently purging stale connections without sending reset (RST) packets to clients.	bigpipe db Connection.AdaptiveReaper.Lowwater <num>
Specify the number of new or untrusted TCP connections that can be established before the system activates the SYN Cookies authentication method for subsequent TCP connections.	bigpipe db Connection.SynCookies.Threshold <num>
Specify that all VLANs should share a single MAC.	bigpipe global vlans unique_mac enable | disable
Enable or disable SNAT packet forwarding.	bigpipe global snats any_ip enable | disable
Manage destination address entries.	Use the browser-based Configuration utility.
View information on service-related events.	less /var/log/ltm
View messages logged by the Syslog utility.	less /var/log/messages
View information on packet filters.	keep, and less /var/log/pktfilter
Set logging options for local-traffic events.	Use the browser-based Configuration utility.
Set logging options for auditing events.	Use the browser-based Configuration utility.
View list of existing user accounts.	No command line interface. Use the browser-based Configuration utility.
Create a user account.	No command line interface. Use the browser-based Configuration utility.
Modify a user account.	No command line interface. Use the browser-based Configuration utility.
Specify the location of user authentication data.	No command line interface. Use the browser-based Configuration utility.
Assign user role to a remote user account.	No command line interface. Use the browser-based Configuration utility.
Run a QKView report.	qkview
Run a TCP dump report.	See solution report SOL2246 on **http://tech.f5.com**.

Table 3.2 *System management tasks (Continued)*

In addition to the tasks that you can perform with BIG-IP utilities and commands, there are tasks that you perform by directly editing certain files with your favorite text editor, such as the LINUX **vi** editor. Table 3.3 lists these tasks and the system configuration files you edit to perform them. For more information on system configuration files, see *Viewing and modifying system configuration files*, on page 3-24.

Other BIG-IP system maintenance tasks	File Name
Create or modify an SNMP trap record.	/etc/alertd/alert.conf
Deny UDP connections to the SNMP agent.	/etc/hosts.deny
Define hosts that are allowed to access the SNMP agent.	/etc/hosts.allow
Configure the SNMP agent.	/config/snmp/snmpd.conf
Specify whether to send an SNMP trap based on a regular expression.	/etc/alertd/alert.conf
Configure the Syslog utility to pipe specified message types through **checktrap.pl**.	/etc/syslog.conf

Table 3.3 *Other BIG-IP maintenance tasks*

The following sections describe some of the system management tasks that you can perform on the BIG-IP system.

Configuring the MGMT port

Before you license the BIG-IP system, you must configure the management port (**MGMT**). You do this by running the **config** command at the command line prompt.

Configuring the management port for the first time

1. Run the **config** command.
2. Specify the IP address of the management (**MGMT**) port.
3. Specify a network mask for the IP address.
4. Specify an IP address for the management route.

Modifying management port settings

If you have licensed the BIG-IP product and want to go back and modify the settings that you configured with the **config** command, you can use the **bigpipe mgmt** command.

Setting failover for BIG-IP system services

You can use the **bigpipe daemon** command to define the action that you want the BIG-IP system to take when certain system services fail. Table 3.4 lists these services.

Service	Definition
mcpd	Messaging and configuration
tmm	Traffic Management
bigd	Health Monitors
sod	Failover
bcm56xxd	Switch hardware driver

Table 3.4 *BIG-IP system services with failover settings*

Displaying protocol statistics

You can use the **bigpipe** utility to display statistics for various types of network traffic. You can use these commands to display protocol-related statistics:

- **bigpipe http**
- **bigpipe icmp**
- **bigpipe ip**
- **bigpipe tcp**
- **bigpipe udp**

You can also display global statistics using the **bigpipe global** command.

Working with the bigtop utility

The **bigtop**™ utility is a real-time statistics display utility. The display shows the date and time of the latest reboot, and lists activity in bits, bytes, or packets. The **bigtop** utility accepts options you use to customize the display of information. For example, you can set the interval at which the data is refreshed, and you can specify a sort order. The **bigtop** utility displays the statistics as shown in Figure 3.1.

```
                   |  bits  since       |  bits in prior     |   current
                   |  Nov 28 18:47:50   |  3 seconds         |   time
BIG-IP      ACTIVE |---In----Out---Conn-|---In----Out---Conn-|   00:31:59
227.19.162.82          1.1G  29.6G 145     1.6K     0      0

virtual ip:port        |---In----Out---Conn-|---In----Out---Conn-|-Nodes Up--
217.87.185.5:80       1.0G  27.4G 139.6K   1.6K     0      0       2
217.87.185.5:20      47.5M  2.1G    3.1K     0      0      0       2
217.87.185.5:20      10.2M 11.5M    2.6K     0      0      0       2

NODE  ip:port      |---In----Out---Conn-|---In----Out---Conn-|--State----
129.186.40.17:80    960.6M  27.4G  69.8K   672      0      0      UP
129.186.40.17:20     47.4M   2.1G   3.1K     0      0      0      UP
129.186.40.18:80    105.3M 189.0K  69.8K   1.0K     0      0      UP
129.186.40.17.21      9.4M  11.1M   1.3K     0      0      0      UP
129.186.40.18:21    700.8K 414.7K   1.3K     0      0      0      UP
129.186.40.18:20      352    320      1      0      0      0      UP
```

Figure 3.1 *The bigtop screen display*

Using bigtop command options

The syntax for the **bigtop** command is as follows:

```
bigtop  [options...]
```

Table 3.5 lists and describes the options you can use with the **bigtop** command.

Option	Description
-bytes	Displays counts in bytes (the default is bits).
-conn	Sorts by connection count (the default is to sort by byte count).
-delay <value>	Sets the interval at which data is refreshed (the default is four seconds).
-delta	Sorts by count since last sample (the default is to sort by total count).
-help	Displays bigtop help.
-nodes <value>	Sets the number of nodes to print (the default is to print all nodes).
-nosort	Disables sorting.

Table 3.5 *bigtop command options*

Option	Description
-once	Prints the information once and exits.
-pkts	Displays the counts in packets (the default is bits).
-scroll	Disables full-screen mode.
-virtuals <value>	Sets the number of virtual servers to print (the default is to print all virtual servers).

Table 3.5 *bigtop command options*

Using runtime commands in bigtop

Unless you specified the **-once** option, the **bigtop** utility continually updates the display at the rate indicated by the **-delay** option. You can also use the following runtime options at any time:

- The **u** option cycles through the display modes: bits, bytes, and packets.
- The **q** option quits the **bigtop** utility.

Exiting the bigtop utility

To exit the **bigtop** utility, simply type **q** at the command line prompt.

Configuring SNMP on a BIG-IP system

SNMP configuration files

The SNMP options that you specify in the Configuration utility are written to one or more of the following configuration files. If you prefer, you can configure SNMP by directly editing the appropriate files with a text editor rather than using the Configuration utility.

- **hosts.deny**
  This file denies all UDP connections to the SNMP agent.
- **hosts.allow**
  This file specifies which hosts are allowed to access the SNMP agent.
- **snmpd.conf**
  This file configures the SNMP agent.
- **snmptrap.conf**
  For the BIG-IP system, the configuration in **/etc/snmptrap.conf** determines which messages generate traps, and what those traps are. Edit this file only if you want to add traps.
- **syslog.conf**
  Configure **/etc/syslog.conf** to pipe specified message types through **checktrap.pl**.

/etc/hosts.deny

This file must be present to deny by default all UDP connections to the SNMP agent. The contents of this file are as follows:

```
ALL : ALL
```

/etc/hosts.allow

The **/etc/hosts.allow** file is used to specify which hosts are allowed to access the SNMP agent. There are two ways to configure access to the SNMP agent with the **/etc/host.allow** file. You can type in an IP address, or list of IP addresses, that are allowed to access the SNMP agent, or you can type in a network address and mask to allow a range of addresses in a subnetwork to access the SNMP agent.

For a specific list of addresses, type in the list of addresses you want to allow to access the SNMP agent. Addresses in the list must be separated by blank space or by commas. The basic syntax is as follows:

```
daemon: <IP address> <IP address> <IP address>
```

For example, you can type the following line which sets the SNMP agent to accept connections from the IP addresses specified:

```
bigsnmpd: 128.95.46.5 128.95.46.6 128.95.46.7
```

For a range of addresses, the basic syntax is as follows, where **daemon** is the name of the daemon, and **IP/MASK** specifies the network that is allowed access. The **IP** must be a network address:

```
daemon: IP/MASK
```

For example, you might use the following line which sets the **bigsnmpd** daemon to allow connections from the **128.95.46.0/255.255.255.0** address:

```
bigsnmpd: 128.95.46.0/255.255.255.0
```

The preceding example allows the 254 possible hosts from the network address **128.95.46.0** to access the SNMP daemon. Additionally, you may use the keyword **ALL** to allow access for all hosts or all daemons.

◆ Note

192.168.1/24 CIDR syntax is not allowed.

The /etc/snmpd.conf file

The **/etc/snmpd.conf** file controls most of the SNMP agent. This file is used to set up and configure certain traps, passwords, and general SNMP variable names. A few of the necessary variables are listed below:

- **System Contact Name**
  The System Contact is a MIB-II simple string variable defined by almost all SNMP boxes. It usually contains a user name, as well as an email address. This is set by the **syscontact** key.
- **Machine Location (string)**
  The Machine Location is a MIB-II variable that almost all boxes support. It is a simple string that defines the location of the box. This is set by the **syslocation** key.
- **Community String**
  The community string clear text password is used for basic SNMP security. This also maps to VACM groups, but for initial read/only access it is limited to only one group.
- **Trap Configuration**
  Trap configuration is controlled by these entries in the **/etc/snmpd.conf** file:
  - **trapsink <host>**
    This sets the host to receive trap information. The **<host>** is an IP address.
  - **trapport <port>**
    This sets the port on which traps are sent. There must be one **trapport** line for each **trapsink** host.
  - **trapcommunity <community string>**
    This sets the community string (password) to use for sending traps. If set, it also sends a trap upon startup: **coldStart(0)**.

- **authtrapenable <integer>**
  Setting this variable to **1** enables traps to be sent for authentication warnings. Setting it to **2** disables it.
- **data_cache_duration <seconds>**
  This is the time in seconds during which data is cached. The default value for this setting is one second.

◆ Note

*A **trapport** line controls all **trapsink** lines that follow it until another **trapport** line appears. Therefore, to change the trap port for a trap sink, the new **trapport** line must be inserted before the trap sink's **trapsink** line, with no other **trapport** lines in between. The same logic follows for **trapcommunity** lines.*

/etc/snmptrap.conf

This configuration file includes OID, trap, and regular expression mappings. The configuration file specifies whether to send a specific trap based on a regular expression. An excerpt of the configuration file is shown in Figure 3.2.

```
# Default traps.
.1.3.6.1.4.1.3375.1.1.110.2.6 (ROOT LOGIN) ROOT LOGIN
.1.3.6.1.4.1.3375.1.1.110.2.5 (denial) REQUEST DENIAL
.1.3.6.1.4.1.3375.1.1.110.2.4 (BIG-IP Loading) SYSTEM RESET
.1.3.6.1.4.1.3375.1.1.110.2.3 (Service detected UP) SERVICE UP
.1.3.6.1.4.1.3375.1.1.110.2.2 (Service detected DOWN) SERVICE DOWN
#.1.3.6.1.4.1.3375.1.1.110.2.1 (error) Unknown Error
#.1.3.6.1.4.1.3375.1.1.110.2.1 (failure) Unknown Failure
```

Figure 3.2 *Excerpt from the **/etc/snmptrap.conf** file*

Some of the OIDs have been permanently mapped to BIG-IP system specific events. The OIDs that are permanently mapped for the BIG-IP system include:

- Root login
- Request denial
- System reset
- Service up
- Service down

You may, however, insert your own regular expressions and map them to the **110.1 OID**. The **/etc/snmptrap.conf** file contains two examples for mapping your own OIDs:

- Unknown error
- Unknown failure

By default, the lines for these files are commented out. Use these OIDs for miscellaneous events. When lines match your expression, they are sent to your management software with the 110.2.1 OID.

If you change this file, restart the SNMP agent **bigsnmpd** as follows:

```
bigstart restart bigsnmpd
```

For the 3-DNS Controller, the configuration in **/etc/3dns_snmptrap.conf** determines which messages generate traps and what those traps are. Edit this file only if you want to add traps.

Configuring snmpd to send responses out of different ports or addresses

You can configure the **snmpd** to respond on different ports or bind the daemon to a specific interface. Use the following syntax to configure **snmpd**:

```
snmpd -p [(udp|tcp):]port[@address][,...]
```

Use this command to make the agent list on the specified list of sockets instead of the default port, which is port 161. Separate multiple ports by commas. You can specify transports by prepending the port number with the transport name (**udp** or **tcp**) followed by a colon.

To bind to a particular interface, you can specify the address you want it to bind with. For example, you can specify the following command to make the agent listen on UDP port 161 for any address, TCP port 161 for any address, and UDP port 9161 on only the interface associated with the localhost address.

```
snmpd -p 161,tcp:161,9161@localhost
```

◆ Note

*The **-T** flag changes the default transport mapping to use (in the previous example, the default transport mapping is UDP).*

Working with the bigdb database

The bigdb™ database holds certain configuration information for the BIG-IP system. Most BIG-IP system utilities use the configuration stored in the bigdb database. You can load configuration information into this bigdb database.

Setting values for bigdb variables

Using the **bigpipe db** command, you can view a bigdb variable, set a new value for a variable, or reset a variable to the default value. If you want to modify the values of variable attributes, such as the variable's data type, you must modify the bigdb database directly. For more information, see *Setting values for bigdb attributes*.

To view the value of a bigdb variable

To view the value of a bigdb variable, type the **bigpipe db** command along with the key name. If you do not specify a key name, the system displays variable values.

```
bigpipe db [<key>] [show]
```

To set the value of a bigdb variable

To set a variable to a specific value, type the **bigpipe db** command along with the key name and a value:

```
b db <key> <value>
```

To set a variable to the default value, type the **bigpipe db** command with the key name and the **reset** keyword:

```
b db <key> reset
```

Setting values for bigdb attributes

You can modify the values of the attributes that are associated with a bigdb variable in the bigdb database. To do this, you must directly edit the file **/config/bigDB.dat**, using your favorite text editor. For a printout of bigdb database entries, see Figure 3.4, on page 3-19.

The attributes associated with a bigdb variable are:

- **Variable name (key)**
  The name for the variable (key). An example is **Bigip.Failover.ActiveMode**.
- **Value**
  The value associated with variable. The system stores this value as a string.
- **Default value**
  The value that the system uses when the variable is otherwise undefined.

- **Type**
  The data type that the system uses to constrain and validate the value. Types are not case-sensitive and can be any of the following: **string**, **integer** (for signed integer), **unsigned_integer**, **ipaddress**, or **enum**.
- **Realm**
  An attribute indicating where a value is relevant (not case-sensitive). Allowed values are: **Local** or **Common**. The system persists both **Local** and **Common** variables, and transfers **Common** variables to a peer during **config sync** operations.
- **Minimum value**
  The minimum value for variables of type **integer** and **unsigned_integer**. This is the shortest length for strings.
- **Maximum value**
  The maximum value for variables of type **integer** and **unsigned_integer**. This is the maximum length for strings.
- **Enumerated value**
  A list of values allowed. The first character is a delimiter for items.

Figure 3.3 shows an example of the format of variable entries in the **/config/bigDB.dat** file.

```
[Bigstart.ChildWaitSec]
value=15
default=10
type=unsigned_integer
min=0
max=32767
realm=common
#
# Open a debug output file for each of the respective monitor
# when set to "true" or "yes"
#
[Bigip.HttpAgents.WMI.LogEnabled]
default=true
realm=local
type=enum
enum=|true|false|yes|no|
```

***Figure 3.3** The format of the **/config/bigDB.dat** file*

To modify bigdb variable attributes

1. Use the **bigstart** command to shut down the bigdb service:

   ```
   bigstart shutdown bigdbd
   ```

2. Using a text editor, edit one or more attribute values in the **/config/bigDB.dat** file.
3. Use the **bigstart** command to restart the bigdb service:

   ```
   bigstart startup bigdbd
   ```

Printing bigdb variables

You can print the values of any bigdb variable and its attributes, using the **printdb** command. You can tailor your printout to print by realm, variable name, or variable name range.

Figure 3.4 shows an example of the output from the **printdb** command.

```
*****************************************************

Name: Bigdb.LogLevel
          Realm:          common
          Type:           unsigned_integer
          Default:        6
          Min:           0
           Max:           7

*****************************************************

Name: Bigdb.UpdatePause
          Realm:          common
          Type:           unsigned_integer
          Default:        30
          Min:            0
           Max:            30
```

Figure 3.4 *Sample printout of bigdb entries*

Working with the Syslog utility

The BIG-IP system supports logging using the **Syslog** utility. The system generates logs automatically, and saves them in user-specified files. These logs contain all changes made to the BIG-IP system configuration, such as those made with the **bigpipe virtual** command, or other **bigpipe** commands, as well as all critical events that occur in the system.

◆ Note

You can configure the Syslog utility to send mail or activate pager notification based on the priority of the logged event.

The Syslog log files track system events based on information defined in the **/etc/syslog.conf** file. You can view the log files in a standard text editor, or with the **less** file page utility.

Table 3.6 shows sample Syslog messages for events that are specific to the BIG-IP system.

.

Sample message	Description
bigd: allowing connections on port **20**	A user specifically allowed connections on virtual port **20**.
bigd: node **192.168.1.1** detected up	The **192.168.1.1** node address was successfully pinged by the BIG-IP system.
bigd: added service port **20** to node **192.168.1.1**	A user defined a new node, **192.168.1.1:20**.
kernel: security: port denial **207.17.112.254:4379** -> **192.168.1.1:23**	A client was denied access to a specific port. The client is identified as coming from **207.17.112.254:4379**, and the destination node is **192.168.1.1:23**.

Table 3.6 *Sample Syslog messages*

Removing and returning items to service

Once you have completed the initial configuration on the BIG-IP system, you may want to temporarily remove specific items from service for maintenance purposes. For example, if a specific network server needs to be upgraded, you may want to disable the nodes associated with that server, and then enable them once you finish installing the new hardware and bring the server back online.

If you specifically disable the nodes associated with the server, the BIG-IP system allows the node to go down only after all the current connections are complete. During this time, the BIG-IP system does not attempt to send new connections to the node. Although the BIG-IP system monitoring features would eventually determine that the nodes associated with the server are down, specifically removing the nodes from service can prevent interruptions on long duration client connections.

You can remove the entire BIG-IP system from service, or you can remove the following individual items from service:

- Virtual servers
- Virtual addresses
- Virtual ports
- Nodes

Removing the BIG-IP system from service

The BIG-IP system offers a Maintenance mode, which allows you to remove the BIG-IP system from network service. This is useful if you want to perform hardware maintenance, or make extensive configuration changes. When you activate Maintenance mode, the BIG-IP system no longer accepts connections to the virtual servers it manages. However, it allows the existing connections to finish processing so that current clients are not interrupted.

The **bigpipe maint** command toggles the BIG-IP system into or out of Maintenance mode. Use the following command to put the BIG-IP system into maintenance mode:

```
bigpipe maint
```

If the BIG-IP system runs in Maintenance mode for less than 20 minutes and you return the machine to the normal service, the BIG-IP system quickly begins accepting connections. However, if the BIG-IP system runs in Maintenance mode for more than 20 minutes, returning the unit to service involves updating all network ARP caches. This process can take a few seconds, but you can speed the process up by reloading the **/config/bigip.conf** file using the following command:

```
bigpipe -f /config/bigip.conf
```

Removing individual virtual servers and virtual addresses from service

The BIG-IP system also supports taking only selected virtual servers, and virtual addresses out of service, rather than removing the BIG-IP system itself from service. Each **bigpipe** command that defines virtual servers and their components supports **enable** and **disable** keywords, which allow you to remove or return the elements from service.

When you remove a virtual address from service, it affects all virtual servers associated with the virtual address.

Enabling and disabling virtual servers and virtual addresses

The **bigpipe virtual** command allows you to enable or disable individual virtual servers, as well as virtual addresses.

To enable or disable a virtual server

To enable or disable a virtual server, use the appropriate command syntax:

```
bigpipe virtual <virtual addr>:<virtual port> enable | disable
```

To enable or disable a virtual address, use the appropriate command syntax:

```
bigpipe virtual <virtual addr> enable | disable
```

Removing individual nodes from service

You can enable or disable individual and nodes from the command line.

To enable and disable nodes

The **bigpipe node** command allows you to enable or disable individual nodes.

To enable or disable a node, use the appropriate command syntax:

```
b node <node addr>:<node port> enable
b node <node addr>:<node port> disable
```

Viewing the currently defined virtual servers and nodes

When used with the **show** parameter, **bigpipe** commands typically display currently configured elements. For example, the **bigpipe virtual show** command displays all currently defined virtual servers, and the **bigpipe node** command displays all nodes currently included in virtual server mappings.

Viewing and modifying system configuration files

The BIG-IP system contains several configuration files that store essential information. You can use your favorite text editor to view or modify these files. Modifying a configuration file is sometimes necessary when there is no browser-based or command line interface to configure a feature. Table 3.7 lists the configuration files on the BIG-IP system.

File	Description
alert.conf	Stores definitions of SNMP traps (system default alerts).
user_alert.conf	Stores definitions of SNMP traps (user-defined alerts).
/config/bigip.conf	Stores all configuration objects for managing local application traffic, such as virtual servers, load balancing pools, profiles, and SNATs.
/config/bigip_base.conf	Stores BIG-IP self IP addresses and VLAN and interface configurations.
/config/bigip.license	Stores authorization information for the BIG-IP system.
/etc/bigconf.conf	Stores the user preferences for the Configuration utility.
/config/bigconfig/openssl.conf	Holds the configuration information for how the SSL library interacts with browsers, and how key information is generated.
/config/user.db	Holds various configuration information. This is known as the bigdb database.
/config/bigconfig/httpd.conf	Holds configuration information for the web server.
/config/bigconfig/users	The web server password file. Contains the user names and passwords of the people permitted to access whatever is provided by the webserver.
/etc/hosts	Stores the hosts table for the BIG-IP system.
/etc/hosts.allow	Stores the IP addresses of workstations that are allowed to make administrative shell connections to the BIG-IP system.
/etc/hosts.deny	Stores the IP addresses of workstations that are not allowed to make administrative shell connections to the BIG-IP system.
/etc/rateclass.conf	Stores rate class definitions.
/etc/ipfwrate.conf	Stores IP filter settings for filters that also use rate classes.
/etc/snmpd.conf	Stores SNMP configuration settings.
/etc/snmptrap.conf	Stores SNMP trap configuration settings.
/config/ssh	Contains the SSH configuration and key files.

Table 3.7 *BIG-IP system configuration files*

File	Description
/etc/sshd_config	This is the configuration file for the secure shell server (SSH). It contains all the access information for people trying to get into the system by using SSH.
/config/routes	Contains static route information.

Table 3.7 *BIG-IP system configuration files*

4

Managing Local Application Traffic

- Introducing local application traffic configuration
- Local traffic management tools
- Performing local traffic management tasks

Introducing local application traffic configuration

There are many tasks that you can perform to customize the way that the BIG-IP system manages local network traffic. The primary command-line tool that you use to perform these tasks is the **bigpipe** utility. When managing SSL traffic, however, there are other tools you can use in addition to the **bigpipe** utility.

Local traffic management tools

The command-line tools that you can use to manage local traffic passing through the BIG-IP system are:

- The **bigpipe** utility
- The OpenSSL toolkit

The **bigpipe** utility is the primary command-line tool that you can use to manage local traffic. Table 4.1 lists and briefly describes the **bigpipe** commands related to local traffic management For more details on these commands, see the online man pages..

Command	Description
help	Displays online help for an individual **bigpipe** command.
auth	Creates the specified type of authentication configuration object. This command is new in version 9 systems and removes the need for the former **bigpipe authz** command.
class	Creates a class and displays all classes included with BIG-IP system.
conn	Shows information about current connections such as the source IP address, virtual server and port, and node.
db	Allows you to configure certain settings globally.
monitor	Defines a health check monitor.
nat	Defines external network address translations for nodes.
node	Defines node property settings.
ocsp responder	Creates or modifies an OCSP responder object, required for SSL OCSP remote authentication. This command is new in version 9 systems.
pool	Defines load balancing pools.
profile	Creates or modifies any type of profile that you specify. This command is new in version 9 systems and removes the need for the former **bigpipe proxy** command.
radius server	Creates or modifies a RADIUS server object, required for RADIUS remote authentication. This command is new in version 9 systems.
rule	Defines load balancing rules.
service	Defines properties for services.
snat	Defines and sets options for SNAT (Secure NAT).
snatpool	Defines and sets options for SNAT pools. This command is new in version 9 systems.
virtual	Defines virtual servers, virtual server mappings, and virtual server properties.

Table 4.1 *bigpipe commands for managing local network traffic*

Performing local traffic management tasks

Using the tools listed in the previous section, you can perform a number of local traffic management tasks. Table 4.2 lists those tasks that you can perform using the **bigpipe** utility.

For many of these tasks, you use multiple **bigpipe** commands in combination. In cases where the commands you use to perform a task differ from those that you used in pre-9.0 versions of the BIG-IP system, this section contains revised procedures, following table 4.2.

Important

The command syntax shown in Table 4.2 is not exhaustive. For each command, see the corresponding man page for the correct syntax.

.

Tasks to configure local traffic management	Command or utility to use
Create and configure a virtual server.	bigpipe virtual
Create and configure a node.	bigpipe node
Create and configure a load balancing pool.	bigpipe pool, bigpipe virtual pool
Monitor the health of a pool member.	bigpipe monitor, bigpipe pool
Monitor the performance of a pool member using the dynamic ratio load balancing method.	bigpipe monitor, bigpipe pool, third-party plug-ins
Manage HTTP traffic.	bigpipe profile http, bigpipe virtual profile
Manage Fast HTTP traffic.	bigpipe profile fasthttp, bigpipe virtual profile
Manage FTP traffic.	bigpipe profile ftp, bigpipe virtual profile
Manage layer 4 traffic.	bigpipe profile layer4, bigpipe virtual profile
Manage TCP traffic.	bigpipe profile tcp, bigpipe virtual profile
Manage UDP traffic.	bigpipe profile udp, bigpipe virtual profile
Configure connection pooling.	bigpipe profile oneconnect, bigpipe virtual profile
Manage Real-time Streaming Protocol (RTSP) traffic.	bigpipe profile stream, bigpipe virtual profile
Implement session persistence (excluding terminated SSL sessions).	bigpipe profile persist, bigpipe virtual persist
Implement persistence for terminated SSL sessions.	bigpipe rule
Enable Keep-Alive support for HTTP/1.0 requests.	bigpipe profile

Table 4.2 *Local traffic management tasks*

Tasks to configure local traffic management	Command or utility to use
Configure compression for HTTP server responses.	bigpipe profile
Configure authentication using a remote LDAP server.	bigpipe profile, bigpipe auth
Configure authentication using a remote RADIUS server.	bigpipe profile, bigpipe auth, bigpipe radius server
Configure authentication using a remote TACACS+ server.	bigpipe profile, bigpipe auth
Configure certificate-based authorization using a remote LDAP server.	bigpipe profile, bigpipe auth
Configure authentication using a remote SSL OCSP responder.	bigpipe profile, bigpipe auth, bigpipe ocsp responder
Implement secure network address translations (SNATs).	bigpipe snat, bigpipe snat translation, bigpipe snatpool, bigpipe rule (optional)
Implement rate shaping to customize throughput.	bigpipe rate class, bigpipe packet filter rule, bigpipe virtual, bigpipe rule (optional)
Create a class for use within an iRule.	bigpipe class
Customize the management of individual connections.	bigpipe rule
Display statistical information for a virtual server or virtual address.	bigpipe virtual <ip_address>:[<service>] show
Display statistical information for a service.	bigpipe <service_name>
Display statistical information for a node.	bigpipe node <ip_address> show
Display statistical information for a SNAT.	bigpipe snat <snat_address> show
Enable or disable a virtual server.	bigpipe virtual <name> enable | disable
Enable or disable a virtual address.	bigpipe virtual <name> enable | disable
Enable or disable a node.	bigpipe node <ip_address>:<sercice> enable | disable

Table 4.2 *Local traffic management tasks (Continued)*

The following sections describe some of the local traffic management tasks that you can perform on the BIG-IP system.

Setting up a basic load balancing configuration

Once you have configured your base network, you can easily set up a basic, local traffic management system by implementing a profile, a load balancing pool, and a virtual server.

To set up a basic load balancing configuration

1. Decide what types of traffic you want the BIG-IP system to manage, as well as whether you want to implement session persistence, connection persistence, and remote authentication.
2. For each decision in step 1, decide whether you want to use the corresponding default profile that the BIG-IP system provides, or whether you want to create a custom profile.
3. If you want to create custom profiles, use the **bigpipe profile** command, specifying the appropriate type of profile as an argument to the **bigpipe profile** command.
   If you do not want to create custom profiles, skip this step.
4. Create one or more load balancing pools, using the **bigpipe pool** command.
5. Create a virtual server, using the **bigpipe virtual** command, and assign to it any profiles and pools that you created. If you are using default profiles, some of those profiles might already be assigned to the virtual server by default.

Managing traffic types

To manage a particular type of network traffic, such as HTTP traffic, you can either create a custom profile of that type (recommended) or modify the default, system-supplied profile of that type (not recommended). After creating or modifying the profile, you then assign the profile to a virtual server. You can manage these types of traffic:

- HTTP
- FTP
- Layer 4
- TCP
- UDP
- Client SSL
- Server SSL

You can also enable session persistence and connection persistence, as well as authenticate network traffic using various types of remote authentication servers. For more information, see the following sections:

- *Implementing session persistence*, on page 4-15
- *Implementing connection persistence*, on page 4-15

For more information on profiles, see the **profile** man page, as well as the man page for each profile type.

To manage a specific type of network traffic

1. Create a profile for a specific type of traffic, such as SSL, using the **bigpipe profile** command. For example, you can manage client-side SSL traffic by using the command **bigpipe profile clientssl** and specifying its arguments.
2. Assign the profile to a virtual server, using the **bigpipe virtual** command.

Optionally, you can write an iRule that includes various commands, which dynamically modify profile settings. For more information, see the ***Configuration Guide for Local Traffic Management***.

Setting Link QoS and IP ToS levels on packets

You can use the **bigpipe** utility to set Quality of Service (QoS) and Type of Service (ToS) levels on packets. You can do this not only for all traffic targeted to a load balancing pool, but also for specific types of traffic, such as layer 4, TCP, and UDP traffic.

To set QoS and ToS levels

1. Decide whether you want to set QoS and ToS levels for traffic targeted for an entire pool or for specific types of traffic, or both.
   a) If you want to set the QoS and ToS levels for an entire pool, use the **bigpipe pool** command with one or more of the following arguments: **link qos to client**, **link qos to server**, **ip tos to client**, and **ip tos to server**.
   b) If you want to set the QoS and ToS levels for certain types of traffic, use the **bigpipe profile** command to create or modify a Fast L4, TCP, or UDP profile.
2. Verify that the pool or the profile that you created or modified is assigned to a virtual server. To do this, use the following syntax:

```
bigpipe virtual <name> list
```

Setting idle timeout values

You can use the **bigpipe** utility to set timeout values for layer 4, HTTP, TCP, or UDP connections that remain idle. You do this by creating or modifying a Fast L4, Fast HTTP, TCP, or UDP profile.

To set idle timeout values

1. Create or modify a Fast L4, Fast HTTP, TCP, or UDP profile, using the **bigpipe profile** command.
2. Specify the **idle timeout** argument to set a timeout value.
3. Verify that the profile you created or modified is assigned to a virtual server.

Generating SSL certificates

When you want the BIG-IP system to manage SSL traffic (that is, authenticate, decrypt, and encrypt SSL traffic), you must generate SSL certificates that the BIG-IP system can use as part of the authentication process.

To generate SSL certificates from the command line, you can use the industry-standard OpenSSL toolkit. You can generate Certificate Authority (CA) certificates, client certificates, certificates for web sites, and CRLs. You can also perform a number of other certificate-related tasks.

Generating CA certificates

You can use the OpenSSL toolkit to generate CA certificates that are trusted for client authentication.

To generate a CA certificate

1. Create a key for the CA, using the **openssl** command.
   For example:

   ```
   openssl genrsa -rand .rand -out bigmirror-ca.key 1024
   ```

2. Create a request for a certificate, using the key that you created in step 1.
   For example:

   ```
   openssl req -x509 -new -key bigmirror-ca.key -out
   bigmirror-ca.crt
   ```

3. Create a configuration file that causes a CRL distribution point to use LDAP.
   For example, this command sequence creates a configuration file named **bigmirror-ca.ext**:

   ```
   echo -e '[ v3_ca
   ]\nsubjectKeyIdentifier=hash\nauthorityKeyIdentifier=keyi
   d:always,issuer:always\nbasicConstraints = CA:true\n[
   crl_ext ]\ncrlDistributionPoints=@crldp\n[ crldp
   ```

```
]\nURI=ldap://192.168.33.100:389/dc=bigmirror,dc=com?cert
ificateRevocationList;binary?sub?cn=DistPoint1' >
bigmirror-ca.ext
```

4. Generate a CA certificate that is trusted for client authentication, using the previously generated key and certificate.

   a) If you want to generate the CA certificate with the LDAP CRL distribution point, use the **openssl x509** command, as in the following example:

   ```
   openssl x509 -in bigmirror-ca.crt -out
   bigmirror-ca.trusted.crt -signkey bigmirror-ca.key
   -days 300 -addtrust clientAuth -addtrust serverAuth
   -setalias "Bigmirror CA" -extensions v3_ca -extensions
   crl_ext -extfile bigmirror-ca.ext
   ```

   b) If you want to generate the CA certificate without the LDAP CRL distribution point, use the **openssl x509** command, as in the following example:

   ```
   openssl x509 -in bigmirror-ca.crt -out
   bigmirror-ca.trusted.crt -signkey bigmirror-ca.key
   -days 300 -addtrust clientAuth -addtrust serverAuth
   -setalias "Bigmirror CA" -extensions v3_ca
   ```

5. Generate a non-trusted (default) CA certificate.
   For example:

   ```
   openssl x509 -in bigmirror-ca.trusted.crt -clrtrust -out
   bigmirror-ca.crt
   ```

6. Convert the certificate to DER format for browsers (import this into browsers).
   For example:

   ```
   openssl x509 -inform pem -outform der -in
   bigmirror-ca.crt -out bigmirror-ca.der
   ```

Creating client certificates

For client-side authentication between a client and a BIG-IP system, you can create a certificate for that client.

To create a client certificate

1. Generate a client key.
   For example:

   ```
   openssl genrsa -rand .rand -out auser1.key 1024
   ```

2. Generate a client certificate request, using the previously-generated key.
   For example:

   ```
   openssl req -new -out auser1.req -key auser1.key
   ```

3. Generate a client certificate.
   In the following example, the certificate is named **auser1.crt**.

   a) If you want to generate the client certificate with the LDAP CRL distribution point, use the **openssl x509** command, as in the following example:

   ```
   openssl x509 -req -in auser1.req -out auser1.crt
   -CAkey bigmirror-ca.key -CA bigmirror-ca.crt -days 300
   -CAcreateserial -CAserial serial -extensions crl_ext
   -extfile bigmirror-ca.ext
   ```

   a) If you want to generate the client certificate without the LDAP CRL distribution point, use the **openssl x509** command, as in the following example:

   ```
   openssl x509 -req -in auser1.req -out auser1.crt
   -CAkey bigmirror-ca.key -CA bigmirror-ca.crt -days 300
   -CAcreateserial -CAserial serial
   ```

4. Create a PKCS12 file using the above key and certificate pairs.
   For example:

   ```
   openssl pkcs12 -export -in auser1.crt -inkey auser1.key
   -out auser1.p12 -name "auser1 pkcs12"
   ```

Creating a certificate for a web site

For server-side authentication between a web site and a BIG-IP system, you can create a certificate for that web site.

To create a certificate for a web site

1. Create a key. For example:

   ```
   openssl genrsa -rand .rand -out www.test.net.key 1024
   ```

2. Generate a certificate request using the key that you generated in step 1. For example:

   ```
   openssl req -new -key www.test.net.key -out
   www.test.net.req
   ```

3. Using the request that you generated in step 2, generate a certificate named for the web site.

   a) If you want to generate the certificate with the LDAP CRL distribution point, use the **openssl x509** command, as in the following example:

   ```
   openssl x509 -req -in www.test.net.req -out
   www.test.net.crt -CAkey bigmirror-ca.key -CA
   bigmirror-ca.crt -days 300 -CAcreateserial -CAserial
   serial -extensions crl_ext -extfile bigmirror-ca.ext
   ```

   a) If you want to generate the certificate without the LDAP CRL distribution point, use the **openssl x509** command, as in the following example:

```
openssl x509 -req -in www.test.net.req -out
www.test.net.crt -CAkey bigmirror-ca.key -CA
bigmirror-ca.crt -days 300 -CAcreateserial -CAserial
serial
```

Working with certificate revocation

You can use the OpenSSL toolkit to create a certificate revocation list (CRL). The BIG-IP system checks a CRL to see if a client or server certificate being presented for authentication has been revoked.

You can also use the toolkit to revoke a certificate.

To create a certificate revocation list

1. Create a configuration file for the serial or index option. For example:

   ```
   echo -e
   'default_ca=ca\n[ca]\ndatabase=index.txt\nserial=serial'
   > bigmirror-ca.config
   ```

2. Generate a CRL that expires in thirty days. For example:

   ```
   openssl ca -config bigmirror-ca.config -gencrl -crldays
   30 -keyfile bigmirror-ca.key -cert bigmirror-ca.crt -out
   bigmirror-ca.crl
   ```

To revoke a certificate

Revoke a client certificate, using the **openssl** command. For example, to revoke the client certificate **auser1.crt**:

```
ropenssl ca -config bigmirror-ca.config -keyfile
bigmirror-ca.key -cert bigmirror-ca.crt -revoke auser1.crt
```

Associating keys and certificates with SSL profiles

You can associate a key and a certificate with an SSL profile by using the **bigpipe profile** command and specifying the key and certificate file names as arguments. For more information, see the man page for the **profile** command.

Performing other certificate-related tasks

There are a number of other SSL-certificate-related tasks that you can perform, using the **openssl** utility.

To verify a certificate

Use this command to verify a certificate:

```
openssl verify -CAfile bigmirror-ca.crt www.test.net.crt
```

To view a CRL

Use this command to view a CRL:

```
openssl crl -in bigmirror-ca.crl -text -noout
```

To view certificate information

Use this command to view certificate information:

```
openssl x509 -in www.test.net.crt -text -noout
```

To convert a certificate to PEM format

Use this command to convert a certificate from PKCS12 (.P12 or.PFX) format to PEM format:

```
openssl pkcs12 -in auser1.p12 -out auser1.pem
```

To add a password to an RSA key

Use this command to add a password to an RSA key:

```
openssl rsa -in auser1.key -out auser1-enc.key -des3 -passout
pass:secret
```

To strip a password from an RSA key

Use this command to strip a password from an RSA key:

```
openssl rsa -in auser1-enc.key -out auser1.key -passin
pass:secret
```

Configuring remote server authentication

When you want to configure the BIG-IP system to use a remote server for authenticating application traffic, you use the **bigpipe auth**, **bigpipe profile**, and **bigpipe virtual** commands. The types of authentication servers that you can use to authenticate network traffic are:

- LDAP servers
- RADIUS servers
- TACACS+ servers
- SSL Client Certificate LDAP servers
- SSL OCSP responders

If the remote authentication server is an SSL OCSP responder or a RADIUS server, you also use the **bigpipe ocsp responder** or **bigpipe radius server** command.

To configure the BIG-IP system for remote authentication

1. Create an authentication configuration object of the appropriate type, using the **bigpipe auth** command.
2. Create an authentication profile of the same type as the configuration object, using the **bigpipe profile** command and specifying the configuration object name as one of the profile settings.
3. If the remote authentication server is an SSL OCSP responder or a RADIUS server, create the appropriate object.
   a) For an SSL OCSP responder, create an SSL OCSP responder object, using the **bigpipe ocsp responder** command.
   b) For a RADIUS server, create a RADIUS server object, using the **bigpipe radius server** command.
4. Associate the authentication profile with a virtual server, using the **bigpipe virtual** command.

Associating health monitors with pools and nodes

To associate a health monitor with a pool or a node, you must create a monitor, create a pool or node, and then associate the monitor with the pool.

To associate a health monitor with a load balancing pool

1. Create a monitor, using the **bigpipe monitor** command, for monitoring the health of the servers that make up your load balancing pool.
2. Configure a load balancing pool with the **bigpipe pool monitor** or **bigpipe pool monitor all** command, specifying the name of the health monitor that you want to use to monitor the pool members. Using these commands, you can assign the same monitor to all pool members, or you can assign different health monitors to individual pool members.
3. Assign the pool to a virtual server, using the **bigpipe virtual pool** command.

To associate a health monitor with a node

1. Create a monitor, using the **bigpipe monitor** command, for monitoring the health of a node.
2. Configure a node with the **bigpipe node monitor** command, specifying the name of the monitor that you want to use to monitor the node.

Configuring HTTP compression

To configure the BIG-IP system to compress HTTP server responses, you use the **bigpipe profile** and **bigpipe virtual** commands.

To configure HTTP compression

1. Configure the compression-related settings of an HTTP profile, using the **bigpipe profile http** command.
2. Assign the HTTP profile to a virtual server, using the **bigpipe virtual** command.

Redirecting HTTP requests

You can redirect HTTP requests by configuring an HTTP profile and specifying a fallback host within the profile.

To redirect HTTP requests

1. Using the **bigpipe profile http** command, create or modify an HTTP profile, specifying a value for the **fallback** argument. You can specify either a URI or the default fallback host, or you can specify that you want no HTTP redirection.
2. Verify that the HTTP profile you created or modified is assigned to a virtual server.

Rewriting HTTP redirections

You can rewrite HTTP redirections by configuring an HTTP profile and specifying that you want the BIG-IP system to rewrite certain HTTP redirections.

To rewrite HTTP redirections

1. Using the **bigpipe profile http** command, create or modify an HTTP profile, specifying a value for the **redirect rewrite** argument.
2. Verify that the HTTP profile you created or modified is assigned to a virtual server.

Inserting and erasing HTTP headers

You can insert headers into HTTP requests or remove headers from HTTP requests by configuring an HTTP or Fast HTTP profile.

To insert or erase HTTP headers

1. Using the **bigpipe profile http** command, create or modify an HTTP profile, specifying a value for either the **header insert**, **header erase**, or **insert xforwarded for** attributes.
2. Verify that the HTTP or Fast HTTP profile you created or modified is assigned to a virtual server.

Tip

*You can also manipulate HTTP headers by configuring a Fast HTTP profile, using the **bigpipe profile fasthttp** command.*

Configuring clone pools

Clone pools are designed for intrusion detection. You can implement clone pools by configuring a virtual server. A clone pool receives all of the same traffic as the normal pool. You therefore use clone pools to *copy* traffic to intrusion detection systems.

To configure a clone pool

Using the **bigpipe virtual** command, create or modify a virtual server, specifying a value for the **clone pool** argument.

Implementing session persistence

To implement session persistence for connections passing through a virtual server, you use the **bigpipe profile** and **bigpipe virtual** commands. You can implement these types of session persistence:

- Cookie
- Destination Address Affinity
- Microsoft Remote Desktop Protocol (MSRDP)
- Session Initiation Protocol (SIP)
- Source Address Affinity
- SSL
- Universal

To configure session persistence

1. Create a persistence profile, using the **bigpipe profile** command, that corresponds to the type of persistence you want to implement.
2. Assign the persistence profile to a virtual server, using the **bigpipe virtual persist** and **bigpipe virtual fallback persist** commands.

Implementing connection persistence

To implement connection persistence, you can add **Keep-Alive** headers into HTTP /1.0 headers where none exist. (By default, HTTP/1.1 connections include **Keep-Alive** support.) You can also enable a feature known as connection pooling, which keeps server-side connections open for re-use by other client requests. You enable Keep-Alive support and connection pooling by creating or modifying an HTTP or Fast HTTP profile, as well as a OneConnect profile.

To add Keep-Alive headers into HTTP requests

1. To ensure that HTTP connections stay open, use the **bigpipe profile http** command and specify the **oneconnect transformations** argument. This ensures that the BIG-IP system inserts a **Connection:Keep-Alive** header into any HTTP /1.0 request that does not already contain one.
2. Make sure that you have assigned the HTTP or Fast HTTP profile to a virtual server, using the **bigpipe virtual** command.

To enable connection pooling

1. Using the **bigpipe profile oneconnect** command, configure a profile for connection pooling.
2. Assign the profile to a virtual server, using the **bigpipe virtual profile** command.

◆ Tip

*You can also configure connection persistence settings by configuring a Fast HTTP profile, using the **bigpipe profile fasthttp** command.*

Unchunking and rechunking HTTP response data

If you want to unchunk a chunked HTTP response for the purpose of inspecting the content, you can enable unchunking by configuring an HTTP profile.

To configure HTTP response chunking

1. Using the **bigpipe profile http** command, create or modify an HTTP profile and specify the **response** argument.
2. Make sure that you have assigned the HTTP profile to a virtual server, using the **bigpipe virtual** command.

Implementing SNATs

There are two basic ways to create a SNAT. You can either directly assign a translation address to one or more original IP addresses, or you can create a SNAT pool and then assign the SNAT pool to the original IP addresses. In the latter case, the BIG-IP system automatically selects a translation address from the assigned SNAT pool.

Note that you can assign these types of mappings from within an iRule.

Mapping a single translation address to an original address

1. Designate an IP address as a translation address, using the **bigpipe snat translation** command.
2. Map the translation address to one or more original IP addresses, using the **bigpipe snat** command or the **bigpipe rule** command.

Mapping a SNAT pool to an original address

1. Create a pool of translation addresses (that is, SNAT pool), using the bigpipe snatpool command.
2. Map the SNAT pool to one or more original IP addresses, using either the **bigpipe snat** command or the **bigpipe rule** command.

Configuring a last hop pool

By default, the auto last hop feature is enabled on the BIG-IP system. If you want to disable that feature and instead explicitly define a last hop router, you can create a last hop pool and assign it to a virtual server.

To configure a last hop pool

1. Using the **bigpipe pool** command, create a last hop pool that contains the router inside addresses.
2. Assign the last hop pool to a virtual server, using the **bigpipe virtual lasthop pool** command.
3. If you have not assigned an SSL profile to the virtual server, assign the profile to the virtual server, using the **bigpipe virtual profile** command.

Implementing rate shaping

To implement rate shaping, you must create a rate class, and then assign the rate class to a virtual server or a packet filter rule.

To implement rate shaping

1. Create one or more rate classes, using the **bigpipe rate class** command.
2. Assign the rate classes to a virtual server or a packet filter rule, using either the **bigpipe virtual** command or the **bigpipe packet filter** command.

Implementing iRules

To implement an iRule from the command line, you use the following procedure.

To implement an iRule

1. Write a script using the industry-standard Tools Command Language (Tcl) and the commands that the BIG-IP system provides as Tcl extensions. Do not attempt to use any Tcl commands that the

BIG-IP system has disabled. BIG-IP system extensions to Tcl, as well as disabled Tcl commands, are listed in the ***Configuration Guide for Local Traffic Management***.

2. Create an iRule by using the **bigpipe rule** command and giving the name of the Tcl script as an argument.
3. Assign the iRule to a virtual server, using the **bigpipe virtual rule** command.

Index

I

K

L

M

N

O

P

Q

R

S

T

U

V