BIG-IP® e-Commerce Controller

version 3.3

Service and Support Information

Product Version

This manual applies to version 3.3 of the BIG-IP® e-Commerce Controller.

Obtaining Technical Support

Web	tech.f5.com
Phone	(206) 272-6888
Fax	(206) 272-6802
Email (support issues)	support@f5.com
Email (suggestions)	feedback@f5.com

Contacting F5 Networks

Web	www.f5.com
Toll-free phone	(888) 961-7242
Corporate phone	(206) 272-5555
Fax	(206) 272-5556
Email	sales@f5.com
Mailing Address	501 Elliott Avenue West Seattle, Washington 98119

Legal Notices

Copyright

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described herein. F5 reserves the right to change specifications at any time without notice.

Trademarks

F5, BIG-IP, and 3-DNS are registered trademarks of F5 Networks, Inc. SEE-IT, GLOBAL-SITE, EDGE-FX, and FireGuard are trademarks of F5 Networks, Inc. Other product and company names are registered trademarks or trademarks of their respective holders.

Export Regulation Notice

The BIG-IP® e-Commerce Controller may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this BIG-IP® e-Commerce Controller from the United States.

Export Warning

This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

FCC Compliance

This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference.

Operation of this equipment in a residential area may cause interference, in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This class A digital apparatus complies with Canadian I CES-003.

Standards Compliance

The product conforms to ANSI/UL Std 1950 and Certified to CAN/CSA Std. C22.2 No. 950.

Acknowledgments

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 the NetBSD Foundation, Inc. and its contributors.

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 Charles Hannum.

This product includes software developed by Charles Hannum, by the University of Vermont and State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.

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).

In the following statement, "This software" refers to the parallel port driver: This software is a component of "386BSD" developed by William F. Jolitz, TeleMuse.

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 developed by Darren Reed. (© 1993-1998 by Darren Reed).

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.

F5 Networks Limited Warranty

This warranty will apply to any sale of goods or services or license of software (collectively, "Products") from F5 Networks, Inc. ("F5"). Any additional or different terms including terms in any purchase order or order confirmation will have no effect unless expressly agreed to in writing by F5. Any software provided to a Customer is subject to the terms of the End User License Agreement delivered with the Product.

Limited Warranty

Software. F5 warrants that for a period of 90 days from the date of shipment: (a) the media on which the software is furnished will be free of defects in materials and workmanship under normal use; and (b) the software substantially conforms to its published specifications. Except for the foregoing, the software is provided AS IS.

In no event does F5 warrant that the Software is error free, that the Product will operate with any software or hardware other than that provided by F5 or specified in the documentation, or that the Product will satisfy Purchaser's own specific requirements.

Hardware. F5 warrants that the hardware component of any Product will, for a period of one year from the date of shipment from F5, be free from defects in material and workmanship under normal use.

Remedy. Purchaser's exclusive remedy and the entire liability of F5 under this limited warranty and any other guarantee made by F5 is, at F5's option, to repair or replace any Product or component that fails during the warranty period at no cost to Purchaser. Products returned to F5 must be pre-authorized by F5 with a Return Material Authorization (RMA) number marked on the outside of the package, and sent prepaid, insured, and packaged appropriately for safe shipment. The repaired or replaced item will be shipped to Purchaser, at F5's expense, no later than 7 days after receipt by F5. Title to any returned Products or components will transfer to F5 upon receipt. F5 will replace defective media or documentation or, at its option, undertake reasonable efforts to modify the software to correct any substantial non-conformance with the specifications.

Restrictions. The foregoing limited warranties extend only to the original Purchaser, and do not apply if a Product (a) has been altered, except by F5, (b) has not been installed, operated, repaired, or maintained in accordance with F5's instructions, (c) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident, or (d) has been operated outside of the environmental specifications for the Product. F5's limited software warranty does not apply to software corrections or upgrades.

Support, Upgrades. F5 provides software telephone support services at no charge for 90 days following the installation of any Product: Monday through Friday, from 6 a.m. to 6 p.m. Pacific time, excluding F5's holidays. Such support will consist of responding to trouble calls as reasonably required to make the Product perform as described in the Specifications. For advisory help requests, which are calls of a more consultative nature than a standard trouble call, F5 will provide up to two hours of telephone service at no charge. Additional service for advisory help requests may be purchased at F5 Networks' then-current standard service fee. During this initial 90

day period, Customer is entitled, at no charge, to updated versions of covered software such as bug fixes, and incremental enhancements as designated by minor revision increases. In addition, Customer will receive special pricing on upgraded versions of covered Products such as new clients, new modules, and major enhancements designated by major revision increases. Customer may purchase a Maintenance Agreement for enhanced maintenance and support services.

DISCLAIMER; LIMITATION OF REMEDY: EXCEPT FOR THE WARRANTIES SPECIFICALLY DESCRIBED HEREIN, F5 DOES NOT MAKE ANY GUARANTEE OR WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO PRODUCTS, SPECIFICATIONS, SUPPORT, SERVICE, OR ANYTHING ELSE. F5 HAS NOT AUTHORIZED ANYONE TO MAKE ANY REPRESENTATION OR WARRANTY OTHER THAN AS PROVIDED ABOVE. F5 DISCLAIMS ANY AND ALL WARRANTIES AND GUARANTEES, EXPRESS, IMPLIED, OR OTHERWISE, ARISING WITH RESPECT TO THE PRODUCTS OR SERVICES DELIVERED HEREUNDER, INCLUDING BUT NOT LIMITED TO THE WARRANTY OF MERCHANTABILITY, THE WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTY OF NON-INFRINGEMENT OF THE INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY. F5 WILL HAVE NO OBLIGATION OR LIABILITY, WHETHER ARISING IN CONTRACT (INCLUDING WARRANTY), TORT (INCLUDING ACTIVE, PASSIVE, OR IMPUTED NEGLIGENCE, STRICT LIABILITY, OR PRODUCT LIABILITY), OR OTHERWISE FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES INCLUDING BUT NOT LIMITED TO LOSS OF USE, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF REVENUE, LOSS OF BUSINESS, OR OTHER FINANCIAL LOSS ARISING OUT OF OR IN CONNECTION WITH ANY OF THE PRODUCTS OR OTHER GOODS OR SERVICES FURNISHED TO CUSTOMER BY F5, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

End-user Software License

IMPORTANT - READ BEFORE INSTALLING OR OPERATING THIS PRODUCT

CAREFULLY READ THE TERMS AND CONDITIONS OF THIS LICENSE BEFORE INSTALLING OR OPERATING THIS PRODUCT - BY INSTALLING, OPERATING OR KEEPING THIS PRODUCT FOR MORE THAN THIRTY DAYS AFTER DELIVERY YOU INDICATE YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS OF THIS AGREEMENT, PROMPTLY CONTACT F5 NETWORKS, INC. ("F5") TO ARRANGE FOR RETURN OF THE PRODUCT FOR A REFUND.

1. Scope. This License applies to the software component ("Software") of the F5 product identified above ("Product") and any corrections, updates, new releases and new versions of such software. This License is a legal agreement between F5 and the single entity ("Licensee") that has acquired Software from F5 under applicable terms and conditions.

2. License Grant. Subject to the terms of this License, F5 grants to Licensee a non-exclusive, non-transferable license to use the Software in object code form with an unlimited number of servers. Other than as specifically described herein, no right or license is granted to Licensee to any of F5's trademarks, copyrights, or other intellectual property rights. The Software incorporates certain third party software, which is used subject to licenses from the respective owners. The protections given to F5 under this License also apply to the suppliers of this third party software, who are intended third party beneficiaries of this License.

3. Restrictions. The Software, documentation, and the associated copyrights and other intellectual

property rights are owned by F5 or its licensors, and are protected by law and international treaties. Licensee may not copy or reproduce the Software, and may not copy or translate the written materials without F5's prior, written consent. Licensee may not copy, modify, reverse compile, or reverse engineer the Software, or sell, sub-license, rent, or transfer the Software or any associated documentation to any third party.

4. Export Control. F5's standard Software incorporates cryptographic software. Licensee agrees to comply with the Export Administration Act, the Export Control Act, all regulations promulgated under such Acts, and all other US government regulations relating to the export of technical data and equipment and products produced therefrom, which are applicable to Licensee. In countries other than the US, Licensee agrees to comply with the local regulations regarding importing, exporting, or using cryptographic software.

5. Limited Warranty. F5 warrants that for a period of 90 days from the date of shipment: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software substantially conforms to its published specifications. Except for the foregoing, the Software is provided AS IS. In no event does F5 warrant that the Software is error free, that it will operate with any software or hardware other than that provided by F5 or specified in the documentation, or that the Software will satisfy Licensee's own specific requirements.

   a. Remedy. Licensee's exclusive remedy and the entire liability of F5 under this limited warranty and any other guarantee made by F5 is, at F5's option, to repair or replace any Software that fails during the warranty period at no cost to Licensee. Any Product returned to F5 must be pre-authorized by F5 with a Return Material Authorization (RMA) number marked on the outside of the package, and sent prepaid, insured and packaged appropriately for safe shipment. The repaired or replaced item will be shipped to Licensee, at F5's expense, no later than 7 days after receipt by F5. Title to any returned Products or components will transfer to F5 upon receipt. F5 will replace defective media or documentation or, at its option, undertake reasonable efforts to modify the Software to correct any substantial non-conformance with the specifications.

   b. Restrictions. The foregoing limited warranties extend only to the original Licensee, and do not apply if the Software or the Product (a) has been altered, except by F5, (b) has not been installed, operated, repaired, or maintained in accordance with F5's instructions, (c) has been subjected to abnormal physical or electrical stress, misuse, negligence or accident or (d) has been operated outside of the environmental specifications for the Product. F5's limited software warranty does not apply to software corrections or upgrades.

6. Infringement Indemnity. F5 will, at its expense, defend any suit brought against Licensee based upon a claim that the Software as delivered by F5 directly infringes a valid patent or copyright. F5 will pay costs and damages finally awarded against Licensee directly attributable to any such claim, but only on condition that (a) F5 is notified in writing of such claim within ten days following receipt by Licensee; (b) F5 has sole control of the defense and settlement negotiations, (c) Licensee provides F5 all information and communications received by Licensee concerning such claim, and (d) Licensee provides reasonable assistance to F5 when requested. F5 will have the right, at its option and expense, (i) to obtain for Licensee rights to use the Software, (ii) to replace or modify the Software so it becomes non-infringing , or (iii) to accept return of the Software in exchange or for a credit not to exceed the purchase price paid by Licensee for such Software. The foregoing, subject to the following restrictions, states the exclusive liability of F5 to Licensee concerning infringement.

   a. Restrictions. F5 will have no liability for any claim of infringement based on: (i) use of a superseded or altered release of the Software, (ii) use of the Software in combination with equipment or software

not supplied or specified by F5 in the Software documentation where the Software would not itself be infringing, (iii) use of the Software in an application or environment not described in the Software Documentation or (iv) Software that has been altered or modified in any way by anyone other than F5 or according to F5's instructions.

7. U.S. Government Restricted Rights. The Software was developed at private expense and is provided with "RESTRICTED RIGHTS." Use, duplication or disclosure by the government is subject to restrictions as set forth in FAR 52.227-14 and DFARS 252.227-7013 et. seq. or its successor. The use of this Software by the government constitutes acknowledgment of F5's and its licensors' rights in the Software.

8. DISCLAIMER; LIMITATION OF REMEDY. EXCEPT FOR THE WARRANTIES SPECIFICALLY DESCRIBED HEREIN, F5 AND ITS THIRD PARTY LICENSORS DO NOT MAKE ANY GUARANTEE OR WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO THE SOFTWARE, SPECIFICATIONS, SUPPORT, SERVICE OR ANYTHING ELSE. NEITHER F5 NOR ITS THIRD PARTY LICENSORS HAVE AUTHORIZED ANYONE TO MAKE ANY REPRESENTATIONS OR WARRANTIES OTHER THAN AS PROVIDED ABOVE. F5 AND ITS THIRD PARTY LICENSORS DISCLAIM ANY AND ALL WARRANTIES AND GUARANTEES, EXPRESS, IMPLIED OR OTHERWISE, ARISING, WITH RESPECT TO THE SOFTWARE OR SERVICES DELIVERED HEREUNDER, INCLUDING BUT NOT LIMITED TO THE WARRANTY OF MERCHANTABILITY, THE WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTY OF NON-INFRINGEMENT OF THE INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY. THE COLLECTIVE LIABILITY OF F5 AND ITS THIRD PARTY LICENSORS UNDER THIS LICENSE WILL BE LIMITED TO THE AMOUNT PAID FOR THE PRODUCT. F5 AND ITS THIRD PARTY LICENSORS WILL HAVE NO OBLIGATION OR LIABILITY, WHETHER ARISING IN CONTRACT (INCLUDING WARRANTY), TORT (INCLUDING ACTIVE, PASSIVE OR IMPUTED NEGLIGENCE, STRICT LIABILITY OR PRODUCT LIABILITY) OR OTHERWISE FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES INCLUDING BUT NOT LIMITED TO LOSS OF USE, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF REVENUE, LOSS OF BUSINESS OR OTHER FINANCIAL LOSS ARISING OUT OF OR IN CONNECTION WITH THE SOFTWARE OR OTHER GOODS OR SERVICES FURNISHED TO LICENSEE BY F5, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

9. Termination. The license granted in Section 2 is effective until terminated, and will automatically terminate if Licensee fails to comply with any of its provisions. Upon termination, Licensee will destroy the Software and documentation and all copies or portions thereof.

10. Miscellaneous. This Agreement will be governed by the laws of the State of Washington, USA without regard to its choice of law rules. The provisions of the U.N. Convention for the International Sale of Goods will not apply. Any provisions found to be unenforceable will not affect the enforceability of the other provisions contained herein, but will instead be replaced with a provision as similar in meaning to the original as possible. This Agreement constitutes the entire agreement between the parties with regard to its subject matter. No modification will be binding unless in writing and signed by the parties.

Table of Contents

Introduction

Chapter 1

Configuring an SSL Accelerator

Chapter 2

Using an SSL Accelerator Cell Configuration

Chapter 3
Using an SSL Accelerator Half Sandwich

Chapter 4
Essential Configuration Tasks

Introduction

- Getting started
- Using the Administrator Kit
- What's new in version 3.3
- Learning more about the BIG-IP Controller product family

Getting started

Before you start installing the controller, we recommend that you browse the Administrator Guide and find the load balancing solution that most closely addresses your needs. Briefly review the basic configuration tasks and the few pieces of information you should gather in preparation for completing the tasks, such as IP addresses and host names.

Once you find your solution and gather the necessary network information, turn to this Installation Guide for hardware installation instructions, and then return to the Administrator Guide to follow the steps for setting up your chosen solution.

Choosing a configuration tool

The BIG-IP platform offers both web-based and command line configuration tools, so that users can work in the environment that they are most comfortable with.

The First-Time Boot utility

All users will use the First-Time Boot utility, a wizard that walks you through the initial system set up. The First-Time Boot utility automatically starts the first time you turn the controller on, and it prompts you to enter basic system information including a root password and the IP addresses that will be assigned to the network interfaces. The Installation Guide provides detailed information about the specific pieces of information that the First-Time Boot utility prompts you to enter.

The Configuration utility

The Configuration utility is a web-based administrative application that you use to configure and monitor the load balancing setup on the BIG-IP Controller. Once you complete the installation instructions described in this guide, you can use the Configuration utility to the configuration steps outlined in the Administrator Guide. In the Configuration utility, you can also monitor current system performance, and download administrative tools such as the

SNMP MIB or the SSH client. The Configuration utility requires Netscape Navigator version 4.7 or later, or Microsoft Internet Explorer version 4.1 or later.

The bigpipe and bigtop command line utilities

The bigpipe™ utility is the command line counter-part to the Configuration utility. Using **bigpipe** commands, you can configure virtual servers, open ports to network traffic, and configure a wide variety of features. To monitor the BIG-IP Controller, you can use certain **bigpipe** commands, or you can use the bigtop™ utility, which provides real-time system monitoring. You can use the command line utilities directly on the BIG-IP Controller, or you can execute commands via a remote shell, such as the SSH client (included with the global release only), or a Telnet client (for countries restricted by cryptography export laws). The BIG-IP Controller Reference Guide provides detailed information about command line syntax.

Using the Administrator Kit

The *BIG-IP® Controller Administrator Kit* provides simple steps for quick, basic configuration, and also provides detailed information about more advanced features and tools, such as the **bigpipe** command line utility. The information is organized into the guides described below.

- ***Installation Guide***
  The Installation Guide walks you through the basic steps needed to get the hardware plugged in and the system connected to the network. Most users turn to this guide only the first time that they set up a BIG-IP Controller. The Installation Guide also covers general network administration issues, such as setting up common network administration tools including Sendmail.
- ***Administrator Guide***
  The Administrator Guide provides examples of common load balancing solutions supported by the particular type of BIG-IP Controller you purchased. For example, in the BIG-IP HA

Controller Administrator Guide, you can find everything from a basic web server load balancing solution to a firewall load balancing solution.

- ***Reference Guide***
  The Reference Guide provides basic descriptions of individual BIG-IP objects, such as pools, nodes, and virtual servers. It also provides syntax information for **bigpipe** commands, configuration utilities, configuration files, and system utilities.
- ***F-Secure SSH User Guide***
  This guide is distributed only with BIG-IP Controllers that support the F-Secure SSH client (a tool used for remote command line access). It provides information about setting up and using the SSH client.

Stylistic conventions

To help you easily identify and understand certain types of information, all F5 Networks administrative documentation uses the stylistic conventions described below.

WARNING

All examples in F5 Networks documentation use only non-routable IP addresses. When you set up the solutions we describe, you must use IP addresses suitable to your own network in place of our sample addresses.

Identifying new terms

When we first define a new term, the term is shown in bold italic text. For example, a ***virtual server*** is a the combination of an IP address and port that maps to a set of back-end servers.

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, the **bigpipe vip** command requires that you include at least one <**node**> variable.

Identifying references to other documents

We use italic text to denote a reference to another document. In references where we provide the name of a book as well as a specific chapter or section in the book, we show the book name in bold, italic text, and the chapter/section name in italic text to help quickly differentiate the two. For example, you can find information about bigpipe commands in the *bigpipe Command Referenc*e section of the ***BIG-IP Controller Reference Guide***.

Identifying command syntax

We show actual, 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 sets the BIG-IP Controller load balancing mode to Round Robin:

```
bigpipe lb rr
```

Table 1 explains additional special conventions used in command line syntax.

Item in text	Description
****	Continue to the next line without typing a line break.
< >	You enter text for the enclosed item. For example, if the command has **<your name>**, type in your name.
|	Separates parts of a command.
[ ]	Syntax inside the square brackets is optional.
...	Indicates that you can type a series of items.

Table 1 *Command line syntax conventions*

Finding additional help and technical support resources

In addition to this administrator guide, you can find technical documentation about the BIG-IP Controller in the following locations:

- **Release notes**
  The release note for the current version of the BIG-IP Controller is available from the web server on the BIG-IP Controller. The release note contains the latest information for the current version, including a list of new features and enhancements, a list of fixes, and, in some cases, a list of known issues.
- **Online help for BIG-IP Controller features**
  You can find help online in three different locations:
  - The web server on the BIG-IP Controller has PDF versions of the guides included in the Administrator Kit. BIG-IP Controller upgrades replace these guides with updated versions as appropriate.
  - The web-based Configuration utility has online help for each screen. Simply click the **Help** button in the toolbar.
  - Individual **bigpipe** commands have online help, including command syntax and examples, in standard UNIX man page format. Simply type the command followed by the question mark option (**-?**), and the BIG-IP Controller displays the syntax and usage associated with the command.
- **Third-party documentation for software add-ons**
  The web server on the BIG-IP Controller contains online documentation for all third-party software included with the BIG-IP Controller, such as GateD.
- **Technical support via the World Wide Web**
  The F5 Networks Technical Support web site, **http://tech.F5.com**, provides the latest technical notes, answers to frequently asked questions, updates for administrator guides (in PDF format), and the Ask F5 natural language question and answer engine. To access this site, you need to obtain a customer ID and a password from the F5 Help Desk.

What's new in version 3.3

The BIG-IP Controller offers the following major new features in version 3.3, in addition to many smaller enhancements.

BIG-IP e-Commerce Controller

The BIG-IP e-Commerce Controller is a new member of the BIG-IP product family. You can use the BIG-IP e-Commerce Controller to process SSL connections to your network. This controller contains a specific set of software and hardware features that accelerate SSL connections.

BIG-IP Cache Controller

This version of the BIG-IP Controller is available as the BIG-IP Cache Controller. The BIG-IP Cache Controller version contains a specific set of features from the BIG-IP Controller that maximizes the efficiency of caches in your network. In addition to the load balancing features available with this controller, this version of the controller has new rule syntax that provides the ability to redirect HTTP requests to caches in your network. These features include:

- **Cacheable content determination**
  This feature enables you to determine the type of content you cache on the basis of any combination of elements in the header of an HTTP request.
- **Content affinity**
  This feature assures that the same cache serves the same content subset even when caches become temporarily unavailable or when caches are added to or deleted from the cache pool.
- **Hot content load balancing**
  When configured, this feature identifies highly requested content and redirects these requests to a hot pool for load balancing.
- **Intelligent cache population**
  When configured, this feature allows caches to retrieve content from other caches in addition to the origin web server.

Performance enhancements

This version of the BIG-IP Controller includes internal performance enhancements. These enhancements improve the overall performance of the BIG-IP Controller.

Learning more about the BIG-IP Controller product family

The BIG-IP Controller platform offers many different software systems. These systems can be stand-alone, or can run in redundant pairs, with the exception of the BIG-IP e-Commerce Controller, which is only available as a stand-alone system. You can easily upgrade from any special-purpose BIG-IP Controller to the BIG-IP HA Controller, which supports all BIG-IP Controller features.

- **The BIG-IP LB Controller**
  The BIG-IP LB Controller provides basic load balancing features.
- **The BIG-IP FireGuard Controller**
  The BIG-IP FireGuard Controller provides load balancing features that maximize the efficiency and performance of a group of firewalls.
- **The BIG-IP Cache Controller**
  The BIG-IP Cache Controller uses content-aware traffic direction to maximize the efficiency and performance of an group of cache servers.
- **The BIG-IP e-Commerce Controller**
  The BIG-IP e-Commerce Controller uses SSL acceleration technology to increase the speed and reliability of the secure connections that drive e-commerce sites.

❖ **The BIG-IP HA Controller**
The BIG-IP HA Controller provides all features from the basic BIG-IP LB Controller to the advanced BIG-IP FireGuard, BIG-IP Cache Controller, and BIG-IP e-Commerce Controller products.

◆ Note

BIG-IP Controllers distributed outside of the United States to a select few countries, regardless of system type, do not support encrypted communications. They do not include the F-Secure SSH client, nor do they support SSL connections to the BIG-IP web server. Instead, you can use the standard Telnet, FTP, and HTTP protocols to connect to the unit and perform administrative functions.

1

Configuring an SSL Accelerator

- Introducing the SSL Accelerator
- Configuring the SSL Accelerator

Introducing the SSL Accelerator

The BIG-IP e-Commerce Controller accepts HTTPS connections (HTTP over SSL), connects to a web server, retrieves the page, and then sends the page to the client.

A key component of the SSL Accelerator feature is that the controller can retrieve the web page using an unencrypted HTTP request to the content server. With this feature, you can configure an SSL gateway on the BIG-IP e-Commerce Controller that decrypts HTTP requests that are encrypted with SSL. Decrypting the request offloads SSL processing from the servers to the BIG-IP Controller. This also allows the BIG-IP e-Commerce Controller to use the header of the HTTP request to intelligently control how the request is handled.

When the SSL gateway on the BIG-IP e-Commerce Controller connects to the content server, it uses the original client's IP address and port as its source address and port, so that it appears to be the client (for logging purposes).

Configuring the SSL Accelerator

There are several steps required to set up the SSL Accelerator on the BIG-IP Controller. These steps include:

- Generating a key and obtaining a certificate
- Installing certificates from the certification authority (CA)
- Creating an SSL gateway
- Enabling, disabling, or deleting a proxy
- Displaying configuration information for an SSL gateway from the command line

Generating a key and obtaining a certificate

In order to use the SSL Accelerator feature you must obtain a valid x509 certificate from an authorized certification authority (CA). The following list contains some companies that are certification authorities:

- Verisign (http://www.verisign.com)
- Digital Signature Trust Company (http://secure.digsigtrust.com)
- GlobalSign (http://www.globalsign.com)
- GTE Cybertrust (http://www.cybertrust.gte.com)
- Entrust (http://www.entrust.net)

You can generate a key, a temporary certificate, and a certificate request form with the Configuration utility or from the command line.

We recommend using the Configuration utility for this process. The certification process is generally handled through a web page. Parts of the process require you to cut and paste information from a browser window in the Configuration utility to another browser window on the web site of the certification authority (CA).

Additional information about keys and certificates

You must have a separate certificate for each domain name on each redundant pair of BIG-IP Controllers, regardless of how many non-SSL web servers are load balanced by the BIG-IP Controller.

If you are already running an SSL server you can use your existing keys to generate temporary certificates and request files. However, you must obtain new certificates if the ones you have are not for the following web server types:

- Apache
- OpenSSL

- Stronghold

WARNING

The BIG-IP Controller does not support Microsoft Internet Information Server (IIS) certificates. You must generate new certificates for your servers if they currently use IIS certificates.

Generating a key and obtaining a certificate using the Configuration utility

To obtain a valid certificate, you must have a private key. If you do not have a key, you can use the Configuration utility on the BIG-IP Controller to generate a key and a temporary certificate. You can also use the Configuration utility to create a request file you can submit to a certification authority (CA). You must complete three tasks in the Configuration utility to create a key and generate a certificate request:

- Generate a certificate request
- Submit the certificate request to a CA and generate a temporary certificate
- Install the SSL certificate from the CA

Each of these tasks is described in detail in the following section.

To create a new certificate request using the Configuration utility

1. In the navigation pane, click **Proxies**.
   The Proxies screen opens.
2. On the toolbar, click **Create SSL Certificate Request**.
   The New SSL Certificate Request screen opens.
3. In the Key Information section, select a key length and key file name.

   a) **Key Length**
   Select the key length you want to use for the key. You can choose either **512** or **1024** bytes.

   b) **Keyfile Name**
   Type the name of the key file. This should be the fully qualified domain name of the server for which you want to request a certificate. You must add the **.key** file extension to the name.
4. In the Certificate Information section, type the information specific to your company. This information includes:

   - **Country**
     Type the two letter ISO code for your country, or select it from the list. For example, the two-letter code for the United States is US.
   - **State or Province**
     Type the full name of your state or province, or select it from the list. You must enter a state or province.
   - **Locality**
     Type the city or town name.
   - **Organization**
     Type the name of your organization.
   - **Organizational Unit**
     Type the division name or organizational unit.
   - **Domain Name**
     Type the name of the domain upon which the server is installed.

- **Email Address**
  Type the email address of a person who can be contacted about this certificate.
- **Challenge Password**
  Type the password you want to use as the challenge password for this certificate. The CA uses the challenge password to verify any changes you make to the certificate at a later date.
- **Retype Password**
  Retype the password to verify the password you entered for the challenge password.

5. Click the **Generate Certificate Request** button.
   After a short pause, the SSL Certificate Request screen opens.
6. In the SSL Certificate Request screen, you can start the process of obtaining a certificate from a certification authority and you can generate and install a temporary certificate:

   - **Begin the process for obtaining a certificate from CA**
     Click the URL of a certification authority (CA) to begin the process of obtaining a certificate for the server. After you select a CA, follow the directions on their web site to submit the certificate request. After your certificate request is approved, and you receive a certificate back from the CA, see *Installing certificates from the CA using the Configuration utility*, on page 1-9, for information about installing it on the BIG-IP Controller.
   - **Generate and install a temporary certificate**
     Click the **Generate Self-Signed Certificate** button to create a self-signed certificate for the server. We recommend that you use the temporary certificate for testing only. You should only take your site live after you receive a properly-signed certificate from a certification authority. When you click this button, a temporary certificate is created and installed on the BIG-IP Controller. This certificate is valid for 30 days.

This temporary certificate allows you to set up an SSL gateway for the SSL Accelerator while you wait for a CA to return a permanent certificate.

Generating a key and obtaining a certificate from the command line

To obtain a valid certificate, you must have a private key. If you do not have a key, you can use the **genconf** and **genkey** utilities on the BIG-IP Controller to generate a key and a temporary certificate. The **genkey** and **gencert** utilities automatically generate a request file you can submit to a certification authority (CA). If you have a key, you can use the **gencert** utility to generate a temporary certificate and request file. These utilities are described in the following list:

- **genconf**
  This utility creates a key configuration file that contains specific information about your organization. The **genkey** utility uses this information to generate a certificate.
- **genkey**
  After you run the **genconf** utility, run this utility to generate a temporary 30-day certificate for testing the SSL Accelerator on the BIG-IP Controller. This utility also creates a request file that you can submit to a certification authority (CA) to obtain a certificate.
- **gencert**
  If you already have a key, run this utility to generate a temporary certificate, and to create a request file for the SSL Accelerator.

To generate a key configuration file using the genconf utility

If you do not have a key, you can generate a key and certificate with the **genconf** and **genkey** utilities. First, run the **genconf** utility from the root (/) with the following commands:

```
cd /
/var/asr/gateway/bin/genconf
```

The utility prompts you for information about the organization for which you are requesting certification. This information includes:

- The fully qualified domain name (FQDN) of the server
- The two letter ISO code for your country
- The full name of your state or province
- The city or town name
- The name of your organization
- The division name or organizational unit

For example, Figure 1.1 contains entries for the server **my.server.net**:

```
Common Name (full qualified domain name): my.server.net
Country Name (ISO 2 letter code): US
State or Province Name (full name): WASHINGTON
Locality Name (city, town, etc.): SEATTLE
Organization Name (company): MY COMPANY
Organizational Unit Name (division): WEB UNIT
```

Figure 1.1 *Example entries for the* ***genconf*** *utility*

After you run the **genconf** utility, you can run the **genkey** utility to create a temporary certificate and a request file.

To generate a key using the genkey utility

After you run the **genconf** utility, you can generate a key with the **genkey** utility. Type the following command from the root (/) to run the **genkey** utility:

```
cd /
/var/asr/gateway/bin/genkey <server_name>
```

For the <**server_name**>, type the fully qualified domain name (FQDN) of the server to which the certificate applies. After the utility starts, it prompts you to verify the information created by the **genconf** utility. After you run this utility, a certification request form is created in the following directory:

```
/var/asr/gateway/requests/<fqdn>.req
```

The **<fqdn>** is the fully qualified domain name of the server. Please contact your certification authority (CA) and follow their instructions for submitting this request form.

In addition to creating a request form you can submit to a certification authority, this utility also generates a temporary certificate. The temporary certificate is located in:

```
/var/asr/gateway/certs/<fqdn>.cert
```

The **<fqdn>** is the fully qualified domain name of the server.

Note that you must copy the key and certificate to the other controller in a redundant system.

This temporary certificate is good for thirty days, after which time you should have a valid certificate from your CA. If you do not have a certificate within 30 days, you can re-run this program.

WARNING

Be sure to keep your previous key if you are still undergoing certification. The certificate you receive is valid only with the key that originally generated the request.

To generate a certificate with an existing key using the gencert utility

To generate a temporary certificate and request file to submit to the certification authority using the **gencert** utility, you must first copy an existing key for a server into the following directory on the BIG-IP Controller:

```
/var/asr/gateway/private/
```

After you copy the key into this directory, type the following command at the command line:

```
cd /
/var/asr/gateway/bin/gencert <server_name>
```

For the **<server_name>**, type the fully qualified domain name (FQDN) of the server to which the certificate applies. After the utility starts, it prompts you for various information. After you run this utility, a certification request form is created in the following directory:

/var/asr/gateway/requests/<fqdn>.req

The **<fqdn>** is the fully qualified domain name of the server. Please contact your certification authority (CA) and follow its instructions for submitting this request form.

Installing certificates from the certification authority (CA)

After you obtain a valid x509 certificate from a certification authority (CA) for the SSL Accelerator, you must copy it onto each BIG-IP Controller in the redundant configuration. You can configure the accelerator with certificates from the Configuration utility or from the command line.

Installing certificates from the CA using the Configuration utility

1. In the navigation pane, click **Proxies**.
   The Proxies screen opens.
2. On the toolbar, click **Install SSL Certificate**.
   The Install SSL Certificate screen opens.
3. In the **Certfile Name** box, type the fully qualified domain name of the server with the file extension **.cert**. Note that if you generated a temporary certificate when you submitted a request to the CA, you need to select the name of the certificate from the drop down list. This allows you to overwrite the temporary certificate with the certificate from the CA.
4. Paste the text of the certificate into the Install SSL Certificate window. Make sure you include the **Begin Certificate** line and the **End Certificate** line. For an example of a certificate, see Figure 1.2.
5. Click the **Write Certificate File** button.

```
-----BEGIN CERTIFICATE-----
MIIB1DCCAX4CAQAwDQYJKoZIhvcNAQEEBQAwdTELMAkGA1UEBhMCVVMxCzAJBgNV
BAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMRQwEgYDVQQKEwtGNSBOZXR3b3JrczEc
MBoGA1UECxMTUHJvZHVjdCBEZXZlbG9wbWVudDETMBEGA1UEAxMKc2VydmVyLm5l
dDAeFw0wMDA0MTkxNjMxNTlaFw0wMDA1MTkxNjMxNTlaMHUxCzAJBgNVBAYTAlVT
MQswCQYDVQQIEwJXQTEQMA4GA1UEBxMHU2VhdHRsZTEUMBIGA1UEChMLRjUgTmV0
d29ya3MxHDAaBgNVBAsTE1Byb2R1Y3QgRGV2ZWxvcG1lbnQxEzARBgNVBAMTCnNl
cnZlci5uZXQwXDANBgkqhkiG9w0BAQEFAANLADBIAkEAsfCFXq3Jt+FevxUqBZ9T
Z7nHx9uaF5x9V5xMZYgekjc+LrF/yazhmq4PCxrws3gvJmgpTsh50YJrhJgfs2bE
gwIDAQABMA0GCSqGSIb3DQEBBAUAA0EAd1q6+u/aMaM2qdo7EjWx14TYQQGomYoq
eydlzb/3FOiJAynDXnGnSt+CVvyRXtvmG7V8xJamzkyEpZd4iLacLQ==
-----END CERTIFICATE-----
```

Figure 1.2 *An example of a certificate*

After the certificate is installed, you can continue with the next step to creating an SSL gateway for the server.

Installing certificates from the CA from the command line

Copy the certificate into the following directory on each BIG-IP Controller in a redundant system:

`/var/asr/gateway/certs/`

◆ Note

The certificate you receive from the certification authority (CA) should overwrite the temporary certificate generated by ***genkey*** *or* ***gencert****.*

If you used the **genkey** or **gencert** utilities to generate the request file, a copy of the corresponding key should already be in the following directory on the BIG-IP Controller:

```
/var/asr/gateway/private/
```

WARNING

The keys and certificates must be in place on both controllers in a redundant system before you configure the SSL Accelerator. You must do this manually; the configuration synchronization utilities do not perform this function.

Create an SSL gateway

After you create the HTTP virtual server for which the SSL Accelerator handles connections, the next step is to create an SSL gateway.

To creating an SSL gateway using the Configuration utility

1. In the navigation pane, click **Proxies**.
   The Proxies screen opens.
2. On the toolbar, click **Add Proxy**.
   The Add Proxy screen opens.
3. In the **Proxy Address** box, type the IP address for the SSL gateway.
4. In the **Proxy Netmask** box, type the netmask you want to use for the SSL gateway. If you leave this setting blank, the BIG-IP Controller creates a default based on the network class of the IP address on the external (destination processing) interface. Type a user-defined netmask only if necessary.
5. In the **Proxy Broadcast** box, type the broadcast address you want to use for this SSL gateway. The BIG-IP Controller automatically generates a broadcast address if you do not type one. Type a user-defined broadcast address only if necessary.
6. In the **Proxy Port** box, type the port number that the proxy server uses, or select a service from the list box. Note that if you select a service, the Configuration utility uses the default port number associated with that service.

7. For **Interface**, select the destination processing interface on which you want to create the SSL gateway. Select **default** to allow the Configuration utility to select the interface based on the network address of the SSL gateway. If you choose **None**, the BIG-IP Controller does not create an alias and generates no ARPs for the virtual IP address.
8. In the **Destination Address** box, type the IP address or host name of the node to which the SSL gateway maps.
9. In the **Destination Port** box, type a port name or number, such as port **80** or **http**, or select the service name from the drop-down list.
10. In the **SSL Certificate** box, type the name of the SSL certificate you installed on the BIG-IP Controller. You can select the certificate you want to use from the drop down list.
11. In the **SSL Key** box, type the name of the SSL key for the certificate you installed on the BIG-IP Controller. You can select the key from the drop down list. It is important that you select the key used to generate the certificate you selected in the **SSL Certificate** box.
12. Click **Apply**.

Creating an SSL gateway from the command line

Use the following command syntax to create an SSL gateway. Use this syntax if you want to configure a gateway by specifying a bitmask instead of a netmask and broadcast address:

```
bigpipe proxy <ip>:<port> [/bitmask] [<ifname>] target server
   <ip>:<port> ssl enable key <key> cert <cert>
```

Use this syntax if you want to configure a gateway by specifying a netmask and broadcast address instead of a bitmask:

```
bigpipe proxy <ip>:<port> [<ifname>] netmask <ip> [broadcast <ip>]
   target server <ip>:<port> ssl enable key <key> cert <cert>
```

As an example, you can create an SSL gateway, from the command line, that looks like this:

```
bigpipe proxy 10.1.1.1:443 exp0 { netmask 255.255.255.0
   broadcast 10.1.1.255 target server 20.1.1.1:80 ssl enable key
   my.server.net.key cert my.server.net.cert }
```

Note that when the configuration is written out in the **bigip.conf** file, the line **ssl enable** is automatically added. When the SSL gateway is written in the **/etc/bigip.conf** file, it looks like this:

```
proxy 10.1.1.1:443 exp0 {
    netmask 255.255.255.0
    broadcast 10.1.1.255
    target server 20.1.1.1:80
    ssl enable
    key my.server.net.key
    cert my.server.net.cert
}
```

Figure 1.3 *An example SSL gateway configuration*

Enabling, disabling, or deleting an SSL gateway

After you have created an SSL gateway, you can enable, disable it, or delete it using the Configuration utility or from the command line.

Enabling or disabling an SSL gateway using the Configuration utility

1. In the navigation pane, click **Proxies**.
   The Proxies screen opens.
2. In the Proxies list, select the SSL gateway you want to enable or disable.
   The Proxy Properties screen opens.
3. In the Proxy Properties screen, clear the **Enable** check box to disable the Proxy, or check the **Enable** box to enable the SSL gateway.
4. Click **Apply**.

Deleting an SSL gateway using the Configuration utility

1. In the navigation pane, click **Proxies**.
   The Proxies screen opens.
2. In the Proxies list, select the SSL gateway you want to delete.
   The Proxy Properties screen opens.
3. On the toolbar, click **Delete**.

Enabling, disabling, or deleting an SSL gateway from the command line

You can enable, disable, or delete an SSL gateway with the following syntax:

```
bigpipe proxy <ip>:<port> enable
bigpipe proxy <ip>:<port> disable
bigpipe proxy <ip>:<port> delete
```

For example, if you want to enable the SSL gateway 209.100.19.22:443,you could type the following command:

```
bigpipe proxy 209.100.19.22:443 enable
```

If you want to disable the SSL gateway 209.100.19.22:443, type the following command:

```
bigpipe proxy 209.100.19.22:443 disable
```

For example, if you want to delete the SSL gateway 209.100.19.22:443, type the following command:

```
bigpipe proxy 209.100.19.22:443 delete
```

Displaying the configuration for an SSL gateway from the command line

You can view the configuration information for an SSL gateway from the command line with the **show** keyword.

Displaying configuration information for an SSL accelerator gateway from the command line

Use the following syntax to view the configuration for the specified SSL gateway:

```
bigpipe proxy <ip>:<port> show
```

For example, if you want to view configuration information for the SSL gateway 209.100.19.22:80, type the following command:

```
bigpipe proxy 209.100.19.22:80 show
```

```
SSL PROXY +---> 11.12.1.200:443 -- Originating Address -- Enabled   Unit 1
          |     Key File Name balvenie.scotch.net.key
          |     Cert File Name balvenie.scotch.net.cert
          +===> 11.12.1.100:80 -- Destination Address -- Server


SSL PROXY +---> 11.12.1.120:443 -- Originating Address -- Enabled   Unit 1
          |     Key File Name balvenie.scotch.net.key
          |     Cert File Name balvenie.scotch.net.cert
          +===> 11.12.1.111:80 -- Destination Address -- Server
```

Figure 1.4 *Output from the* ***bigpipe proxy show*** *command*

2

Using an SSL Accelerator Cell Configuration

- Introducing the SSL accelerator cell configuration
- Configuring the BIG-IP Controller which load balances the SSL accelerator cells
- Setting the default route on each node in the cell

Introducing the SSL accelerator cell configuration

This chapter explains how to set up a scalable SSL accelerator configuration. This configuration is useful for any enterprise that handles a large amount of encrypted traffic.

With this configuration, you can increase the scale of the network by adding a new ***cell***. A cell consists of an SSL accelerator and one or more nodes for which it proxies SSL connections.

Figure 2.1 shows a configuration of an SSL acclerator cell. The SSL accelerator cell described in this chapter includes BIG-IP Controllers 1a and 1b, the SSL accelerator **accelerator1**, and **Node1** and **Node2**.

The following sections refer to Figure 2.1 as an example of how you can set up such a configuration.

◆ Note

The IP addresses shown in the example configuration are fictitious. When implementing your configuration, choose IP addresses that are consistent with your network or networks.

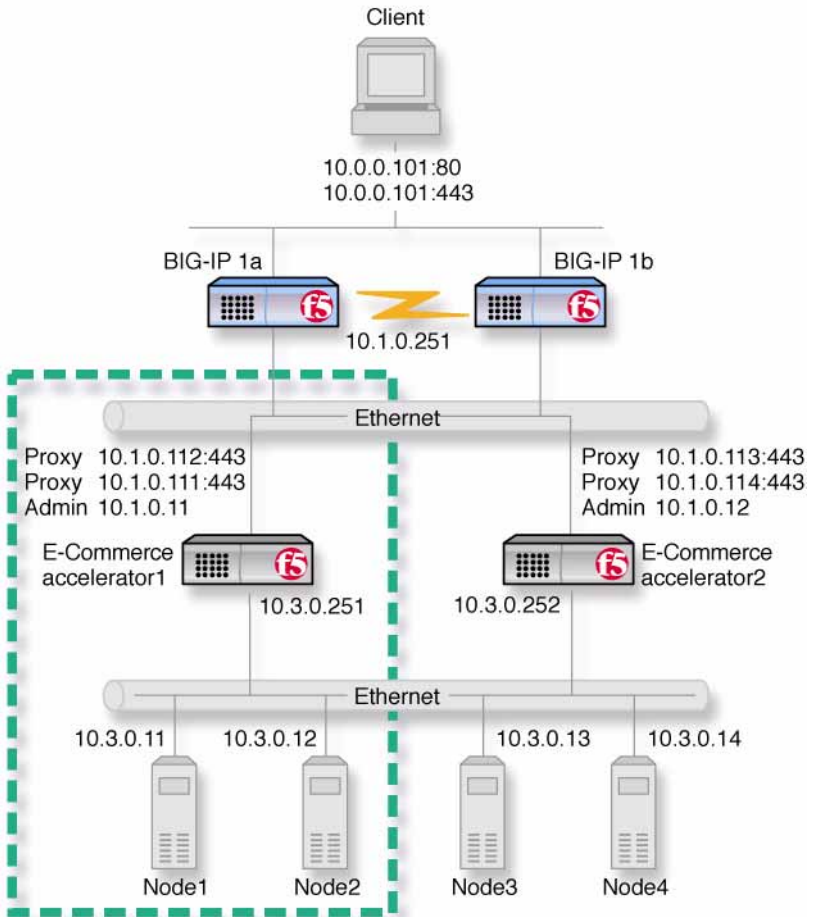

Figure 2.1 *An SSL accelerator cell configuration. The cell is outlined by the dashed line.*

Configuration tasks

To configure an SSL accelerator cell, you must configure the BIG-IP Controller redundant pair that load balances the SSL accelerators, each SSL accelerator, and each node that handles connections from the SSL accelerator.

First, complete the following tasks on the BIG-IP Controller that you want to use to load balance connections to the SSL accelerators:

- Configure interfaces on the BIG-IP Controller redundant system.
- Modify the **/etc/netstart** file on the BIG-IP Controller that you want to use to load balance the SSL accelerators.
- Create two load balancing pools. One pool load balances HTTP connections using the IP addresses of the web servers, the other pool load balances SSL connections to the SSL accelerators.
- Create virtual servers that reference the load balancing pools. Create one virtual server for the pool load balancing the SSL connections, and another virtual server that references the pool that load balances the HTTP connections to the SSL accelerators.
- Enable port 80 and port 443 on the controller.

Next, complete the following tasks for the SSL accelerator in the cell:

- Set up an SSL gateway for each node for which the SSL accelerator handles connections.
- Enable port 443.
- Set the idle connection timer for port 443.
- Turn on IP forwarding.

Finally, complete the following task on each node in the cell:

- Set the default route on each node in the cell to point to the internal interface (source processing) of the SSL accelerator serving that cell.

Configuring the BIG-IP Controller which load balances the SSL accelerator cells

To configure the BIG-IP Controller which load balances the SSL accelerator cells, complete the following tasks on the BIG-IP Controller. This section describes how to complete each task.

- Configure interfaces on the BIG-IP Controller.
- Modify the **/etc/netstart** file on the BIG-IP Controller that you want to use to load balance the SSL accelerators.
- Create two load balancing pools One pool load balances HTTP connections using the IP addresses of the web servers, the other pool load balances SSL connections from the SSL accelerators.
- Create virtual servers that reference the load balancing pools.
- Enable port 80 and port 443 on the controller.

Configuring interfaces on the BIG-IP Controller

You must configure the interfaces on the redundant BIG-IP Controller system (1a and 1b, in Figure 2.1) to process source and destination addresses. Note that in a basic controller configuration, one interface is configured as an internal interface (source processing), and the other interface is configured as an external interface (destination processing).

In order for the SSL accelerator cell load balancing to work, you must turn destination processing **on** for the internal interface, and source processing **on** for the external interface.

To configure source and destination processing using the Configuration utility

1. In the navigation pane, click **NICs**.
   The Network Interface Cards screen opens. You can view the current settings for each interface in the Network Interface Card table.

2. In the Network Interface Card table, click the name of the interface you want to configure.
   The Network Interface Card Properties screen opens.
   - To enable source processing for this interface, click the **Enable Source Processing** check box.
   - To enable destination processing for this interface, click the **Enable Destination Processing** check box.
3. Click the **Apply** button.

To configure source and destination processing from the command line

Use the following syntax to configure source and destination processing on the specified interface:

```
bigpipe interface <interface> dest [ enable | disable ]
bigpipe interface <interface> source [ enable | disable ]
```

The following example command enables destination processing on the interface **exp0**:

```
bigpipe interface exp0 dest enable
```

The following example command enables source processing on the interface **exp1**:

```
bigpipe interface exp1 source enable
```

Add routes for nodes to /etc/netstart

In order for traffic to pass through this configuration correctly, you must configure routes for the nodes in the SSL accelerator cell configuration on the BIG-IP Controller. Add the routes for the nodes to the end of **/etc/netstart**. In the example shown in Figure 2.1, you must add routes for **Node1**, **Node2**, **Node3**, and **Node4**. The entries look like this in the **/etc/netstart** file:

```
route add -host 10.3.0.11 -gateway 10.1.0.11
route add -host 10.3.0.12 -gateway 10.1.0.11
route add -host 10.3.0.13 -gateway 10.1.0.12
route add -host 10.3.0.14 -gateway 10.1.0.12
```

Create load balancing pools

This section describes how to create the load balancing pools required for the SSL accelerator configuration described in Figure 2.1. The two pools you need to create are:

- A load balancing pool for connections using the IP addresses of the web server. For this example, the HTTP pool is named **http_virtual**. This pool contains the following members:
  Node1 (10.3.0.11)
  Node2 (10.3.0.12)
  Node3 (10.3.0.13)
  Node4 (10.3.0.14)
- A load balancing pool for SSL connections from the SSL accelerators. For this example, the SSL accelerator is named **ssl_gateways**. This pool contains the following member:
  accelerator1 (10.1.0.111)
  accelerator2 (10.1.0.112)
  accelerator3 (10.1.0.113)
  accelerator4 (10.1.0.114)

◆ Note

Note that the SSL accelerator pool should contain the SSL accelerator for each SSL accelerator cell.

To create a pool using the Configuration utility

1. In the navigation pane, click **Pools**.
   The Pools screen opens.
2. In the toolbar, click the **Add Pool** button.
   The Add Pool screen opens.
3. In the Add Pool screen, configure the load balancing method, persistence attributes, and members for the pool.

   Configuration note

   - For this example, you could create an HTTP pool named **http_virtual**. This pool contains the following members:
     Node1 (10.3.0.11)

Node2 (10.3.0.12)
Node3 (10.3.0.13)
Node4 (10.3.0.14)

- For this example, you could create an SSL accelerator pool named **ssl_gateways**. This pool contains the following member:
accelerator1 (10.1.0.111)
accelerator2 (10.1.0.112)
accelerator3 (10.1.0.113)
accelerator4 (10.1.0.114)
- For additional information about configuring a pool, click the **Help** button.

To define a pool from the command line

To define a pool from the command line, use the following syntax:

```
bigpipe pool <pool_name> {lb_mode <lb_mode> member
   <member_definition> ... member <member_definition>}
```

For example, if you want to create the pool **http_virtual** and the pool **ssl_gateways**, you would type the following commands:

```
bigpipe pool http_virtual { lb_mode rr member 10.3.0.11:80 member
   10.3.0.12:80 member 10.3.0.13:80 member 10.3.0.14:80 }
bigpipe pool ssl_gateways { lb_mode rr member 10.1.0.111:80 member
   10.1.0.112:80 member 10.1.0.113:80 member 10.1.0.114:80 }
```

Create the virtual servers

Create a virtual server that references the pool load balancing the SSL connections, and another virtual server that references the pool that load balances the HTTP connections through the SSL accelerators.

To define a standard virtual server that references a pool using the Configuration utility

1. In the navigation pane, click **Virtual Servers**.

2. On the toolbar, click **Add Virtual Server**.
   The Add Virtual Server screen opens.

3. Fill in the attributes for the virtual server.

   Configuration notes

   - To create the configuration described in Figure 2.1, create a virtual server **10.0.0.101** on port **443** that references the pool of SSL accelerators.
   - To create the configuration described in Figure 2.1, create a virtual server **10.0.0.101** on port **80** that references the pool of content servers.
   - For additional information about this screen, click the **Help** button on the tool bar.

To define a standard virtual server mapping from the command line

Type the **bigpipe vip** command as shown below. Also, note that you can use host names in place of IP addresses, and that you can use standard service names in place of port numbers.

```
bigpipe vip <virt IP>:<port> use pool <pool_name>
```

To create the virtual servers for the configuration in Figure 2.1, you could type the following commands, where the pool of SSL accelerators is named **ssl_gateways** and the pool for HTTP requests is named **http_virtual**:

```
bigpipe vip 10.0.0.101:443 use pool ssl_gateways
bigpipe vip 10.0.0.101:80 use pool http_virtual
```

Enable ports 80 and 443 on the BIG-IP Controller

For security reasons, the BIG-IP Controller ports do not accept traffic until you enable them. In this configuration, the BIG-IP Controller accepts traffic on port 443 for SSL, and port 80 for HTTP. For this configuration to work, you must enable port 80 and port 443.

Use the following command to enable these ports:

```
bigpipe port 80 443 enable
```

Configuring an SSL accelerator for use in a cell

The next part of the process in configuring an SSL accelerator cell is to configure the SSL accelerator. Complete the following tasks on each SSL accelerator in the cell:

- Set up an SSL gateway for each node for which the SSL accelerator handles connections.
- Enable port 443.
- Set the idle connection timer for port 443.
- Turn on IP forwarding.

Set up an SSL gateway for each node in the SSL accelerator cell

The first task you must complete on the SSL accelerator it to set up an SSL gateway for each node for which the SSL accelerator handles connections. Using the example for creating an SSL Accelerator cell in Figure 2.1, you create two SSL gateways on **accelerator1**:

- An SSL gateway (10.1.0.111) with **Node1** (10.3.0.11) as a target
- An SSL gateway (10.1.0.112) with **Node2** (10.3.0.12) as a target

The following section includes procedures for adding an SSL gateway to the SSL Accelerator configuration.

Creating an SSL gateway using the Configuration utility

1. In the navigation pane, click **Proxies**.
   The Proxies screen opens.
2. On the toolbar, click **Add Proxy**.
   The Add Proxy screen opens.

3. In the **Proxy Address** box, type the IP address for the SSL gateway. For **accelerator1** SSL accelerator cell, the IP address for the gateway is **10.1.0.111**. For **accelerator2** SSL accelerator cell, the IP address for the gateway is **10.1.0.112**.
4. In the **Proxy Netmask** box, type the netmask you want to use for the SSL gateway. If you leave this setting blank, the BIG-IP Controller creates a default based on the network class of the IP address on the external (destination processing) interface. Type a user-defined netmask only if necessary.
5. In the **Proxy Broadcast** box, type the broadcast address you want to use for this SSL gateway. The BIG-IP Controller automatically generates a broadcast address if you do not type one. Type a user-defined broadcast address only if necessary.
6. In the **Proxy Port** box, type the port number that the proxy server uses, or select a service from the list box. Note that if you select a service, the Configuration utility uses the default port number associated with that service.
7. For **Interface**, select the destination processing interface on which you want to create the SSL gateway. Select **default** to allow the Configuration utility to select the interface based on the network address of the SSL gateway.
8. In the **Destination Address** box, type the IP address or host name of the node to which the SSL gateway maps.
9. In the **Destination Port** box, type a port name or number, such as port **80** or **http**, or select the service name from the drop-down list.
10. In the **SSL Certificate** box, type the name of the SSL certificate you installed on the BIG-IP Controller. You can select the certificate you want to use from the drop down list.

11. In the **SSL Key** box, type the name of the SSL key for the certificate you installed on the BIG-IP Controller. You can select the key from the drop down list. It is important that you select the key used to generate the certificate you selected in the **SSL Certificate** box.
12. Click **Apply**.

Creating an SSL gateway from the command line

Use the following command syntax to create an SSL gateway. Use this syntax if you want to configure a gateway

```
bigpipe proxy <ip>:<port> [<ifname>] netmask <ip> [broadcast <ip>]
   target server <ip>:<port> ssl enable key <key> cert <cert>
```

For example, to create the SSL gateways for the **accelerator1** SSL accelerator cell, you would use the following commands:

```
bigpipe proxy 10.1.0.111:443 exp0 { netmask 255.255.255.0
   broadcast 10.1.0.255 target server 10.3.0.11:80 ssl enable key
   my.server.net.key cert my.server.net.cert }
bigpipe proxy 10.1.0.111:443 exp0 { netmask 255.255.255.0
   broadcast 10.1.0.255 target server 10.3.0.12:80 ssl enable key
   my.server.net.key cert my.server.net.cert }
```

Enable port 443

For security reasons, the ports on the SSL accelerator do not accept traffic until you enable them. In this configuration, the SSL accelerator accepts traffic on port 443 for SSL. For this configuration to work, you must enable port 443. Use the following command to enable this port:

```
bigpipe port 443 enable
```

Set the idle connection timer for port 443

In the SSL accelerator cell configuration, you should set the idle connection timer to clean up closed connections on port 443. You need to set an appropriate idle connection time-out value so that valid connections are not disconnected, and closed connections are cleaned up in a reasonable time.

To set the idle connection time-out using the Configuration utility

1. In the navigation pane, click **Virtual Servers**.
2. In the Virtual Servers list, click the virtual server you configured for SSL connections.
   The Virtual Server Properties screen opens.
3. In the **Port** box, click the port. For the example in this section, choose **443**.
   The Global Virtual Port Properties screen opens.
4. In the **Idle connection timeout TCP (seconds)** box, type a time-out value for TCP connections. The recommended time-out setting is 10 seconds.
5. In the **Idle connection timeout UDP (seconds)** box, type a time-out value for TCP connections. The recommended time-out setting is 10 seconds.
6. Click **Apply**.

To set the idle connection time-out in the /etc/bigip.conf file

To set the idle connection time-out in the **/etc/bigip.conf** file, edit the following lines:

```
treaper <port> <seconds>
udp <port> <seconds>
```

For the example in Figure 2.1, the entries look like this:

```
treaper 443 10
udp 443 10
```

The **<seconds>** value is the number of seconds a connection is allowed to remain idle before it is terminated. The **<port>** value is the port on the wildcard virtual server for which you are configuring out of path routing. The recommended value for the TCP and UDP connection timeouts is 10 seconds.

Turn on IP forwarding

In order for traffic from the nodes to be routed back to the client correctly, you must turn on IP forwarding for the SSL accelerator in the cell.

IP forwarding is a property of the BIG-IP Controller system, and it is controlled by the system control variable **net.inet.ip.forwarding**.

To set the IP forwarding system control variable using the Configuration utility

1. In the navigation pane, click the BIG-IP Controller icon.
   The BIG-IP System Properties screen opens.
2. On the toolbar, click **Advanced Properties**.
   The BIG-IP System Control Variables screen opens.
3. Check the **Allow IP Forwarding** box.
4. Click the **Apply** button.

To set the IP forwarding system control variable from the command line

Use the standard **sysctl** command to set the variable. The default setting for the variable is **0**, which is **off**. You want to change the setting to **1**, which is **on**:

```
sysctl -w net.inet.ip.forwarding=1
```

To permanently set this value, you can use a text editor, such as **vi** or **pico**, to manually edit the **/etc/rc.sysctl** file. For additional information about editing this file, see ***BIG-IP Controller Reference Guide***, *System Control Variables*.

Setting the default route on each node in a cell

The final task you must complete for this configuration is to set the default route on each node in the cell to point to the internal interface (source processing) of the SSL accelerator serving that cell.

In the configuration described in Figure 2.1, the default routes for the content servers should be set like this:

- You should set the default route on **Server1** and **Server2** to the internal address of **accelerator1**, which is **10.3.0.251**.
- You should set the default route on **Server3** and **Server4** to the internal address of **accelerator2**, which **10.3.0.252**.

Note

For information about how to set the default route on the content servers in your network, refer to the documentation provided with the content server.

3

Using an SSL Accelerator Half Sandwich

- Introducing the SSL accelerator half sandwich configuration
- Configuring the BIG-IP Controller handling inbound traffic
- Configuring each SSL accelerator
- Configuring the BIG-IP Controller that load balances the content servers
- Configuring the content servers

Introducing the SSL accelerator half sandwich configuration

This chapter explains how to set up a scalable SSL accelerator configuration. This configuration is useful for any enterprise that handles a large amount of encrypted traffic.

With this configuration, you can increase the scale of the network by adding new SSL accelerators to the configuration. You can use this configuration to load balance encrypted traffic to the SSL accelerators while standard HTTP traffic is sent directly to a second BIG-IP Controller which load balances the connections to the content servers.

Figure 3.1 shows a configuration of an SSL accelerator half-sandwich. The following sections refer to Figure 3.1 as an example of how you can set up such a configuration.

◆ Note

The IP addresses shown in the example configuration are fictitious. When implementing your configuration, choose IP addresses that are consistent with your network or networks.

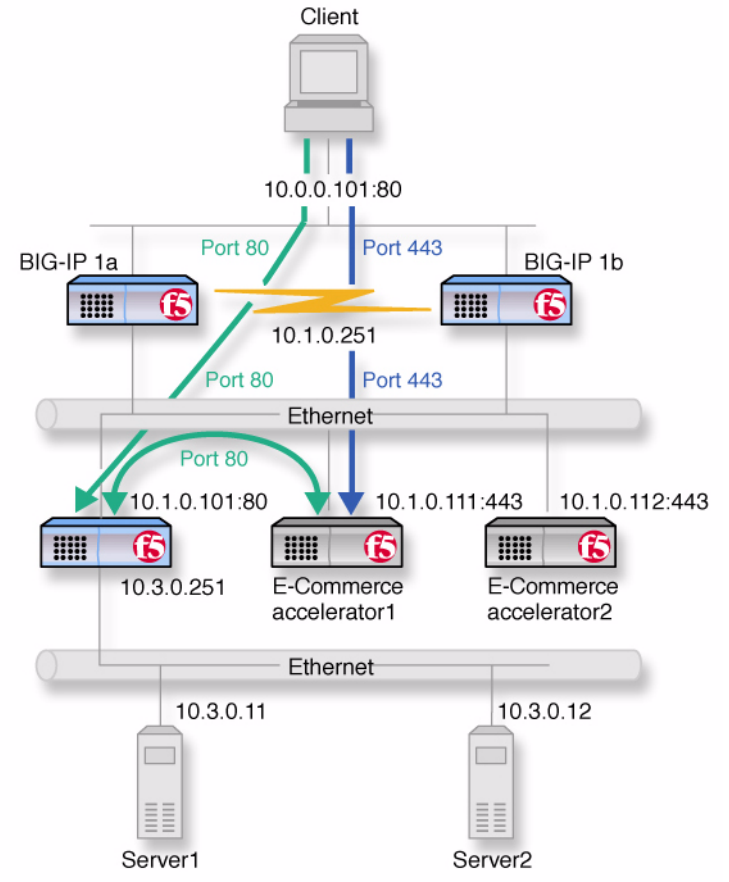

Figure 3.1 *An SSL accelerator half-sandwich.*

Configuration tasks

First, complete the following tasks on the BIG-IP Controllers **1a** and **1b** that you want to use to load balance traffic coming into your network:

- Create two load balancing pools. One pool load balances HTTP connections using the IP address of a virtual server on the second BIG-IP Controller (10.3.0.251), and another pool that load balances SSL connections to the SSL accelerators.
- Create two virtual servers. One virtual server references the pool that contains the IP address of the virtual server on the other controller. The second virtual server references the pool for load balancing the SSL accelerators.
- Enable port 80 and port 443 on the controller.

Next, complete the following tasks on each SSL accelerator in the half-sandwich:

- Configure interfaces on each SSL accelerator.
- Set up an SSL gateway that points to the virtual server that handles HTTP requests on the BIG-IP Controller (10.3.0.251).

Next, complete the following tasks on the BIG-IP Controller (10.3.0.251) that load balances HTTP requests from the SSL accelerators and HTTP requests from the BIG-IP Controllers **1a** and **1b**:

- Configure interfaces on the second BIG-IP Controller.
- Create a pool of web servers that handle HTTP connections.
- Create a pool of devices from which the controller receives HTTP connections.
- Create one virtual server that handles connections for the content servers.
- Creating a last hop pool of devices from which the controller receives requests
- Adding the last hop pool from which this controller recieves HTTP connections to the virtual server
- Enable port 80.

Next, complete the following tasks on each content server:

- Set the default route on each node in the cell to point to the internal IP address of the second BIG-IP Controller.

Configuring the BIG-IP Controllers handling inbound traffic

First, complete the following tasks on the BIG-IP Controllers **1a** and **1b** that you want to use to load balance traffic coming into your network:

- Create two load balancing pools. One pool load balances HTTP connections using the IP address of a virtual server on the second BIG-IP Controller (10.3.0.251), and another pool load balances SSL connections to the SSL accelerators.
- Create two virtual servers. One virtual server references the pool that contains the IP address of the virtual server on the other controller. The second virtual server references the pool for load balancing the SSL accelerators.
- Enable port 80 and port 443 on the controller.

Create load balancing pools for HTTP and SSL requests

Create two load balancing pools. One pool load balances HTTP connections using the IP address of a virtual server on the second BIG-IP Controller (10.3.0.251), and another pool load balances SSL connections to the SSL accelerators.

This section describes how to create the load balancing pools required for the SSL accelerator configuration described in Figure 3.1. The two pools you need to create are:

- A load balancing pool that load balances HTTP connections using the IP address of a virtual server on the second BIG-IP Controller (10.3.0.251). For this example, the HTTP pool is named **http_virtual**. This pool contains the member **10.1.0.101:80**.
- A load balancing pool for SSL connections to the SSL accelerators. For this example, the SSL accelerator is named **ssl_gateways**. This pool contains the following members:
  accelerator1 (10.1.0.111:443)
  accelerator2 (10.1.0.112:443)

To create a pool using the Configuration utility

1. In the navigation pane, click **Pools**.
   The Pools screen opens.
2. In the toolbar, click the **Add Pool** button.
   The Add Pool screen opens.
3. In the Add Pool screen, configure the load balancing method, persistence attributes, and members for the pool.

   Configuration notes

   - For this example, you could create an HTTP pool named **http_virtual**. This pool contains the following member: **10.1.0.101:80**
   - For this example, you could create an SSL accelerator pool named **ssl_gateways**. This pool contains the following members:
     accelerator1 (10.1.0.111:443)
     accelerator2 (10.1.0.112:443)
   - For additional information about configuring a pool, click the **Help** button.

To define a pool from the command line

To define a pool from the command line, use the following syntax:

```
bigpipe pool <pool_name> {lb_method <lb_method> member
   <member_definition> ... member <member_definition>}
```

For example, if you want to create the pool **http_virtual** and the pool **ssl_gateways**, you would type the following command:

```
bigpipe pool http_virtual { lb_mode rr member 11.1.0.101:80 }
bigpipe pool ssl_gateways { lb_mode rr member 10.1.0.111:443 member
   10.1.0.112:443 }
```

Creating the virtual servers that reference the HTTP and SSL pools

Create a virtual server that references the pool load balancing the SSL connections, and another virtual server that references the pool that load balances the HTTP connections to the SSL accelerators.

To define a standard virtual server that references a pool using the Configuration utility

1. In the navigation pane, click **Virtual Servers**.
2. On the toolbar, click **Add Virtual Server**.
   The Add Virtual Server screen opens.
3. Fill in the attributes for the virtual server.

   Configuration notes

   - To create the configuration described in Figure 3.1, create a virtual server **10.0.0.101** on port **443** that references the pool of SSL accelerators.
   - To create the configuration described in Figure 3.1, create a virtual server **10.0.0.101** on port **80** that references the pool of content servers.
   - For additional information about this screen, click the **Help** button on the tool bar.

To define a standard virtual server mapping from the command line

Type the **bigpipe vip** command as shown below. Also, you can use host names in place of IP addresses, and you can use standard service names in place of port numbers.

```
bigpipe vip <virt IP>:<port> use pool <pool_name>
```

To create the virtual servers for the configuration in Figure 3.1, you could type the following commands, where the pool of SSL accelerators is named **ssl_gateways** and the pool for HTTP requests is named **http_virtual**:

```
bigpipe vip 10.0.0.101:443 use pool ssl_gateways
```

```
bigpipe vip 10.0.0.101:80 use pool http_virtual
```

Enable ports 80 and 443

For security reasons, the BIG-IP Controller ports do not accept traffic until you enable them. In this configuration, the BIG-IP Controller accepts traffic on port 443 for SSL, and port 80 for HTTP. For this configuration to work, you must enable port 80 and port 443. Use the following command to enable these ports:

```
bigpipe port 80 443 enable
```

Configuring each SSL accelerator

Next, complete the following tasks on each SSL accelerator in the half-sandwich:

- Configure interfaces on each SSL accelerator.
- Set up an SSL gateway that points to the virtual server that handles HTTP requests on the BIG-IP Controller (10.3.0.251).
- Set the idle connection timer for port 443.

Configuring interfaces on each SSL accelerator

You must configure the interfaces on the each SSL accelerator to process source and destination addresses. In a basic controller configuration, one interface is configured as an internal interface (source processing), and the other interface is configured as an external interface (destination processing).

In order for the SSL accelerator half sandwich to work, you must turn destination processing **on** for the internal interface, and source processing **on** for the external interface.

To configure source and destination processing using the Configuration utility

1. In the navigation pane, click **NICs**.
   The Network Interface Cards screen opens. You can view the current settings for each interface in the Network Interface Card table.
2. In the Network Interface Card table, click the name of the interface you want to configure.
   The Network Interface Card Properties screen opens.
   - To enable source processing for this interface, click the **Enable Source Processing** check box.
   - To enable destination processing for this interface, click the **Enable Destination Processing** check box.
3. Click the **Apply** button.

To configure source and destination processing from the command line

Use the following syntax to configure source and destination processing on the specified interface:

```
bigpipe interface <interface> dest [ enable | disable ]
bigpipe interface <interface> source [ enable | disable ]
```

The following example command enables destination processing on the interface **exp0**:

```
bigpipe interface exp0 dest enable
```

The following example command enables source processing on the interface **exp1**:

```
bigpipe interface exp1 source enable
```

Setting up an SSL gateway that points to the HTTP virtual server on the second BIG-IP Controller

The next step is to set up an SSL gateway that points to the virtual server that handles HTTP requests on the BIG-IP Controller (10.3.0.251). The SSL gateway passes the HTTP request to the BIG-IP Controller which then load balances them to the content servers.

The first task you must complete on the SSL accelerator it to set up an SSL gateway for each node for which the SSL accelerator handles connections. Using the example for creating an SSL Accelerator cell in Figure 3.1, you create an SSL gateway on **accelerator1** and an SSL gateway **on accelerator2**:

- An SSL gateway on **accelerator1** that has the virtual server **10.1.0.101:80** as a target
- An SSL gateway on **accelerator2** that has the virtual server **10.1.0.101:80** as a target

The following section includes procedures for adding an SSL gateway to the SSL Accelerator configuration.

Creating an SSL gateway using the Configuration utility

1. In the navigation pane, click **Proxies**.
   The Proxies screen opens.
2. On the toolbar, click **Add Proxy**.
   The Add Proxy screen opens.
3. In the **Proxy Address** box, type the IP address for the SSL gateway. For **accelerator1** SSL accelerator, the IP address for the gateway is **10.1.0.111:443**. When you create the second SSL gateway for **accelerator2**, the IP address for the gateway is **10.1.0.112:443**.
4. In the **Proxy Port** box, type the port number that the proxy server uses, or select a service from the list box. Note that if you select a service, the Configuration utility uses the default port number associated with that service.

5. For **Interface**, select the destination processing interface on which you want to create the SSL gateway. Select **default** to allow the Configuration utility to select the interface based on the network address of the SSL gateway.
6. In the **Destination Address** box, type the IP address or host name of the node to which the SSL gateway maps. In this example, the destination should be the virtual server **10.1.0.101** on the second BIG-IP Controller.
7. In the **Destination Port** box, type a port name or number, such as port **80** or **http**, or select the service name from the drop-down list.
8. In the **SSL Certificate** box, type the name of the SSL certificate you installed on the BIG-IP Controller. You can select the certificate you want to use from the drop down list.
9. In the **SSL Key** box, type the name of the SSL key for the certificate you installed on the BIG-IP Controller. You can select the key from the drop down list. It is important that you select the key used to generate the certificate you selected in the **SSL Certificate** box.
10. Click **Apply**.

Creating an SSL gateway from the command line

Use the following command syntax to create an SSL gateway. Use this syntax if you want to configure a gateway:

```
bigpipe proxy <ip>:<port> [<ifname>] netmask <ip> [broadcast <ip>]
   target server <ip>:<port> ssl enable key <key> cert <cert>
```

For example, to create the SSL gateways for the **accelerator1** SSL accelerator cell, you would use the following commands:

```
bigpipe proxy 10.1.0.111:443 exp0 { netmask 255.255.255.0
   broadcast 10.1.0.255 target server 10.1.0.101:80 ssl enable key
   my.server.net.key cert my.server.net.cert }
bigpipe proxy 10.1.0.112:443 exp0 { netmask 255.255.255.0
   broadcast 10.1.0.255 target server 10.1.0.101:80 ssl enable key
   my.server.net.key cert my.server.net.cert }
```

Configuring the BIG-IP Controller that load balances the content servers

Next, complete the following tasks on the BIG-IP Controller (10.3.0.251) that load balances HTTP requests from the SSL accelerators and HTTP requests from the BIG-IP Controllers **1a** and **1b**:

- Configure interfaces on the second BIG-IP Controller.
- Create a pool of web servers that handle HTTP connections.
- Create a virtual server that handles connections for the content servers.
- Creating a last hop pool of devices from which the controller receives requests
- Adding the last hop pool from which this controller recieves HTTP connections to the virtual server
- Enable port 80.
- Set the default route on the controller to the internal IP alias of the BIG-IP Controllers **1a** and **1b**.

Configure interfaces for the BIG-IP Controller

You must configure the interfaces on the redundant BIG-IP Controller system (**1a** and **1b**, in Figure 3.1) to process source and destination addresses. Note that in a basic controller configuration, one interface is configured as an internal interface (source processing), and the other interface is configured as an external interface (destination processing).

In order for the SSL accelerator cell load balancing to work, you must turn destination processing **on** for the internal interface, and source processing **on** for the external interface.

To configure source and destination processing in the Configuration utility

1. In the navigation pane, click **NICs**.
   The Network Interface Cards screen opens. You can view the current settings for each interface in the Network Interface Card table.
2. In the Network Interface Card table, click the name of the interface you want to configure.
   The Network Interface Card Properties screen opens.
   - To enable source processing for this interface, click the **Enable Source Processing** check box.
   - To enable destination processing for this interface, click the **Enable Destination Processing** check box.
3. Click the **Apply** button.

To configure source and destination processing from the command line

Use the following syntax to configure source and destination processing on the specified interface:

```
bigpipe interface <interface> dest [ enable | disable ]
bigpipe interface <interface> source [ enable | disable ]
```

The following example command enables destination processing on the interface **exp0**:

```
bigpipe interface exp0 dest enable
```

The following example command enables source processing on the interface **exp1**:

```
bigpipe interface exp1 source enable
```

Creating a pool for the content servers

This section describes how to create the load balancing pools required for the SSL accelerator configuration described in Figure 3.1.

The pool you need to create is a load balancing pool for connections using the IP addresses of the web server. For this example, the HTTP pool is named **http_virtual**. This pool contains the following members:
Server1 (10.3.0.11)
Server2 (10.3.0.12)

To create a pool using the Configuration utility

1. In the navigation pane, click **Pools**.
   The Pools screen opens.
2. In the toolbar, click the **Add Pool** button.
   The Add Pool screen opens.
3. In the Add Pool screen, configure the load balancing method, persistence attributes, and members for the pool.

   Configuration notes

   - For this example, you could create an HTTP pool named **http_virtual**. This pool contains the following members:
     server1 (10.3.0.11)
     server2 (10.3.0.12)
   - For additional information about configuring a pool, click the **Help** button.

To define a pool from the command line

To define a pool from the command line, use the following syntax:

```
bigpipe pool <pool_name> {lb_mode <lb_mode> member
   <member_definition> ... member <member_definition>}
```

For example, if you want to create the pool **http_virtual**, you would type the following command:

```
bigpipe pool http_virtual { lb_mode rr member 10.3.0.11:80 member
   10.3.0.12:80 }
```

Creating a virtual server that references the HTTP pool

Next, create a virtual server that references the pool load balancing HTTP connections.

To define a standard virtual server that references a pool using the Configuration utility

1. In the navigation pane, click **Virtual Servers**.
2. On the toolbar, click **Add Virtual Server**.
   The Add Virtual Server screen opens.
3. Fill in the attributes for the virtual server.

   Configuration notes

   - To create the configuration described in Figure 3.1, create a virtual server **10.0.0.101** on port **80** that references the pool of content servers.
   - For additional information about this screen, click the **Help** button on the tool bar.

To define a standard virtual server mapping from the command line

Type the **bigpipe vip** command as shown below. Also, remember that you can use host names in place of IP addresses, and that you can use standard service names in place of port numbers.

```
bigpipe vip <virt IP>:<port> use pool <pool_name>
```

To create the virtual server for the configuration in Figure 3.1, you could type the following command, where the pool for HTTP requests is named **http_virtual**:

```
bigpipe vip 10.0.0.101:80 use pool http_virtual
```

Creating a last hop pool of devices from which the controller receives requests

This section describes how to create the load balancing pools required for the SSL accelerator configuration described in Figure 3.1.

The pool you need to create is a load balancing pool you can use for a last hop pool for connections received from other devices by the controller. For this example, the HTTP pool is named **http_sources**. This pool contains the following members:
BIG-IP 1a and 1binternal alias (10.1.0.251)
accelerator1 (10.1.0.111)
accelerator (10.1.0.112)

To create a pool using the Configuration utility

1. In the navigation pane, click **Pools**.
   The Pools screen opens.
2. In the toolbar, click the **Add Pool** button.
   The Add Pool screen opens.
3. In the Add Pool screen, configure the load balancing method, persistence attributes, and members for the pool.

   Configuration notes

   - For this example, you could create an HTTP pool named **http_sources**. This pool contains the following members:
     BIG-IP 1a and 1binternal alias (10.1.0.251:any)
     accelerator1 (10.1.0.111:any)
     accelerator (10.1.0.112:any)
   - Specify **any** for the port for each member.
   - For additional information about configuring a pool, click the **Help** button.

To define a pool from the command line

To define a pool from the command line, use the following syntax:

```
bigpipe pool <pool_name> {lb_mode <lb_mode> member
   <member_definition> ... member <member_definition>}
```

For example, if you want to create the pool **http_sources**, you would type the following command:

```
bigpipe pool http_sources { lb_mode rr member 10.1.0.251:any member
   10.1.0.112:any member 10.1.0.112:any }
```

Adding the last hop pool from which this controller recieves HTTP connections to the virtual server

The next step is to add the last hop pool of all the devices (**http_sources**) from which the controller recieves HTTP connections. This pool includes each SSL accelerator that passes on HTTP connections to the second BIG-IP Controller.

To configure a last hop pool using the Configuration utility

1. In the navigation pane, click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the Virtual Servers Current List, select the virtual server configured for HTTP connections. In this example the virtual server is **10.1.0.101:80**.
   The Virtual Server Properties screen opens.
3. In the Last Hop Pool section, select the pool for load balancing HTTP connections from all devices. In this example, the pool is **http_sources**.
4. Click the **Apply** button.

To configure a last hop pool from the command line

To configure a last hop pool, you must first create a pool that contains the routers for the BIG-IP Controller. Use the following command to configure a last hop pool for the virtual server **10.1.0.101:80** that uses the pool **http_sources**.

```
bigpipe vip 10.1.0.101:80 lasthop pool http_sources
```

Enable port 80

For security reasons, the BIG-IP Controller ports do not accept traffic until you enable them. In this configuration, the BIG-IP Controller accepts traffic on port 80 for HTTP. For this configuration to work, you must enable port 80. Use the following command to enable this port:

```
bigpipe port 80 enable
```

Configuring the content servers

The final task you must complete for this configuration is to set the default route on each server to point to the internal interface (source processing) of the second BIG-IP Controller (10.3.0.251).

In the configuration described in Figure 3.1, the default routes for the content servers should be set like this:

You should set the default route on **Server1** and **Server2** to the internal address of the second BIG-IP Controller, which is **10.3.0.251**.

4

Essential Configuration Tasks

- Determining which configuration tasks to do
- Allowing access to ports and services
- Configuring timer settings
- Configuring NATs and IP forwarding for nodes

Determing which configuration tasks to do

Before you follow the instructions in this chapter, you need to browse through the prior chapters to find the specific load balancing solution you want to set up. Each load balancing chapter describes the configuration tasks you need to complete to set up the solution, but it points you to this chapter for the actual configuration steps.

This chapter covers the essential configuration tasks that all users must complete, regardless of the chosen load balancing solution. The chapter also includes optional configuration tasks that most users find they want to do. In the individual load balancing solution chapters, you can find information about which optional configuration tasks and advanaced features may be right for you.

Basic configuration tasks

- **Allow access to ports and services**
  The services and ports on a BIG-IP Controller are locked down and cannot accept connections until you specifically open them to network access. For each service that one or more of your virtual servers supports, you need to open the corresponding port number for network access. However, ports are automatically enabled when you use them in virtual server definition in the Configuration utility.
- **Configure the timer settings**
  The BIG-IP Controller supports several timer settings, but for a simple configuration, there are only two that you need to set. First, you need to set the amount of time that idle connections are allowed to remain open. Second, you need to set the frequency at which the BIG-IP Controller checks nodes to make sure they are up and available to accept connections passed on by a virtual server.

Optional configuration tasks

This chapter also covers additional configuration options that users typically add to a simple configuration, including:

- **Configure NATs or IP forwarding**
  You can set up network address translation (NAT) or IP forwarding to allow direct connections to and from nodes.

WARNING

When you set configuration options in the Configuration utility, they are immediately saved to the appropriate configuration file. However, when you set configuration options using the ***bigpipe*** *command line utility, they are temporarily stored in system memory, and are not saved to a configuration file unless you execute the* ***bigpipe -s*** *command. For more information about this command, see the* ***BIG-IP Controller Reference Guide****, bigpipe Command Reference.*

Table 4.1 describes the different types of connection configurations available on the BIG-IP Controller.

	NAT	SNAT	IP Forwarding	Virtual server	Forwarding virtual server
Security	Medium	High	Low (see following note)	High	High
Routable addresses required on the internal network	No	No	Yes	No	Yes
Protocols	TCP and UDP	TCP and UDP	Any IP protocol	TCP and UDP	TCP and UDP
NT Domain support	No	No	Yes	No	Yes
Active FTP support	No	Yes	Yes	Yes	Yes

Figure 4.1 *Connection configuration options for the BIG-IP Controller*

	NAT	SNAT	IP Forwarding	Virtual server	Forwarding virtual server
Connection origination	Any direction	One direction	Any direction	One direction	One direction
Ports	Does not matter	Does not matter	Does not matter	Uses specific ports or wildcard	Uses specific ports or wildcard
Setup for specific nodes or hosts	Yes	Yes, but can use wildcards	No	Yes, but can use wildcard	Yes
Load balancing	No	No	No	Yes	No

Figure 4.1 *Connection configuration options for the BIG-IP Controller*

◆ Note

Although IP forwarding does not require setup for specific hosts, the BIG-IP Controller supports IP filters that you can configure to restrict traffic.

Allowing access to ports and services

One of the security features of the BIG-IP Controller is that all ports on the controller are locked down and unavailable for service unless you specifically open them to network access. Before clients can use the virtual servers you have defined, you must allow access to each port that the virtual servers use.

This is the third task of the four essential tasks you must complete for a basic configuration. You must perform this task after you create a pool and a virtual server that references the pool, and before you configure the timer settings.

Tip

Virtual servers using the same service actually share a port on the BIG-IP Controller. This command is global, you only need to open access to a port once; you do not need to open access to a port for each instance of a virtual server that uses it.

To allow access to services using the Configuration utility

Any time you create a virtual server and define a port or service with the Configuration utility, the port or service is automatically enabled.

To allow access to services from the command line

Using the **bigpipe port** command, you can allow access to one or more ports at a time.

```
bigpipe port <port>... <port> enable
```

For example, in order to enable HTTP (port 80) and Telnet (port 23) services, you can enter the following **bigpipe port** command:

```
bigpipe port 80 23 443 enable
```

WARNING

*In order for FTP to function properly, you must allow both ports 20 and 21 (or **ftp-data** and **ftp**).*

Configuring the timer settings

Configuring timer settings is the fourth task of the four essential tasks you must complete for a basic configuration. You must perform this task after you configure virtual servers and after you allow access to services and ports.

There are two essential timer settings that you need to configure:

- The node ping timer defines how often the BIG-IP Controller will ping node addresses to verify whether a node is **up** or **down**. It also defines how long the BIG-IP Controller waits for a response from a node before determining that the node is unresponsive and marking the node **down**.
- The idle connection timer defines how long an inactive connection is allowed to remain open before the BIG-IP Controller deletes the record of the connection, closing it and disconnecting the client.

The service check timer is optional, and you need to set it only if you want the BIG-IP Controller to check to see if a service, or even specific content, is available on a particular node.

◆ Note

If you plan to use simple service checks, or ECV or EAV service checks, you need to set the service check timer.

Setting the node ping timer

The node ping timer is an essential setting on the BIG-IP Controller that determines how often the BIG-IP Controller checks node addresses to see whether they are **up** and available or **down** and unavailable. The node ping timer setting applies to all nodes configured for use by the BIG-IP Controller, and it is part of the BIG-IP Controller system properties.

◆ Note

The ping interval should be set to occur about three times during every timeout period. For example, if you set the ping value to 5 seconds, we recommend that you set the timeout to 16 seconds.

To set the node ping timer using the Configuration utility

1. In the navigation pane, click the BIG-IP Controller icon. The BIG-IP System Properties screen opens.

2. In the Node Ping section of the table, in the **Ping** box, type the frequency (in seconds) at which you want the BIG-IP Controller to ping each node address it manages. A setting of 5 seconds is adequate for most configurations.
3. In the Node Ping section of the table, in the **Timeout** box, type the number of seconds you want the BIG-IP Controller to wait to receive a response to the ping.

Configuration notes

- If the BIG-IP Controller does not receive a response to the ping before the node ping timeout expires, the BIG-IP Controller marks the node **down** and does not use it for load balancing. A setting of 16 seconds is adequate for most configurations
- For additional information about the options on this screen, click the **Help** button.

To set the node ping timer from the command line

To define node ping settings, you use two commands. First, you set the node ping frequency using the **bigpipe tping_node** command, and then you set the node ping timer using the **bigpipe timeout_node** command.

```
bigpipe tping_node <seconds>
bigpipe timeout_node <seconds>
```

For example, the following commands sets the ping frequency at 5 seconds, and the timer to 16 seconds, which should be adequate for most configurations.

```
bigpipe tping_node 5
bigpipe timeout_node 16
```

Setting the timer for reaping idle connections

The BIG-IP Controller supports two timers for reaping idle connections, one for TCP traffic and one for UDP traffic. These timers are essential, and if they are set too high, or not at all, the BIG-IP Controller may run out of memory. Each individual port on the BIG-IP Controller has its own idle connection timer settings.

WARNING

The BIG-IP Controller accepts UDP connections only if you set the UDP idle connection timer.

To set the inactive connection timer using the Configuration utility

1. In the navigation pane, click the expand button (+) next to **Virtual Servers**.
   The Virtual Server tree opens and displays the Ports option.
2. Click **Ports**.
   The Global Virtual Ports screen opens.
3. In the Global Virtual Ports screen, click the port number or service name for which you want to configure the idle connection timeout.

   Configuration notes

   - For the HTTP connections, we recommend setting the **Idle Connection Timeout TCP** to 60 seconds. For other services such as Telnet, higher settings may be necessary.
   - In the **Idle Connection Timeout UDP** box, type the number of seconds you want to elapse before the BIG-IP Controller drops UDP connections.
   - For additional information about the options on this screen, click the **Help** button.

To set TCP idle connection timers from the command line

Use the **bigpipe treaper** to define a TCP idle connection timeout for one or more ports at a time. For HTTP connections we recommend only 60 seconds, but for other services such as Telnet we recommend higher settings. The default setting for this timer is 16 minutes (1005 seconds). Use the following syntax for this command:

```
bigpipe treaper <port>... <port> <seconds>
```

For example, the following command sets a 120 second time limit for idle connections on port 443:

```
bigpipe treaper 443 120
```

To set UDP idle connection timers from the command line

You can define a UDP idle connection timeout for one or more ports at a time using the **bigpipe udp** command.

```
bigpipe udp <port>... <port> <seconds>
```

For example, the following command sets a 120-second time limit for idle connections on port 53:

```
bigpipe udp 53 120
```

Setting the service check timer

The service check feature is similar to node ping, but instead of testing the availability of a server, it tests the availability of a particular service running on a server. The service check timer affects the three different types of service checks: simple service check, ECV service check, and EAV service check. To set up simple service check, you need only set the service check timer as described below.

Note that each individual service managed by the BIG-IP Controller has its own service check timer settings.

To set the service check timer using the Configuration utility

1. In the navigation pane, click the expand button (+) next to **Nodes**.
   The Nodes tree opens and displays the Ports option.
2. Click **Ports**.
   The Global Node Ports screen opens.
3. In the Global Node Port Properties screen, click the port for which you want to configure the service check timer

   Configuration notes

   - For the **Frequency** setting, we recommend 5 seconds for most configurations.
   - For the **Timeout** setting, we recommend 16 seconds for most configurations.
   - For additional information about the options on this screen, click the **Help** button.

To set the service check timer on the command line

To define service check settings, you actually use two commands. First, you set the service check frequency using the **bigpipe tping_svc** command, and then set the service check timer using the **bigpipe timeout_svc** command.

```
bigpipe tping_svc <port> <seconds>
bigpipe timeout_svc <port> <seconds>
```

For example, the following sequence of commands sets the service check frequency at 5 seconds, and the timer to 16 seconds, which is adequate for most configurations.

```
bigpipe tping_svc 80 5
bigpipe timeout_svc 80 16
```

Configuring NATs and IP forwarding for nodes

Configuring NATs and IP forwarding are optional tasks you can configure after you have completed the three main tasks of a basic configuration. This means you already have:

- Configured virtual servers
- Configured access to ports and services
- Configured the timer settings

After you complete the basic tasks, you can configure network address translation and IP forwarding on the BIG-IP Controller.

The IP addresses that identify nodes on the BIG-IP Controller's internal network need not be routable on the external network. This protects nodes from illegal connection attempts, but it also prevents nodes (and other hosts on the internal network) from receiving direct administrative connections, or from initiating connections to clients, such as mail servers or databases, on the BIG-IP Controller's external interface (destination processing).

Using network address translation resolves this problem. Network address translations (NATs) assign to a particular node a routable IP address that the node can use as its source IP address when connecting to servers on the BIG-IP Controller's external interface. You can use the NAT IP address to connect directly to the node through the BIG-IP Controller, rather than having the BIG-IP Controller send you to a random node according to the load balancing mode. IP forwarding provides functionality similar to a NAT. If your network does not support NATs, you may want to consider using IP forwarding.

◆ Note

In addition to these options, you can set up forwarding virtual servers which allow you to selectively forward traffic to specific addresses. The BIG-IP Controller maintains statistics for forwarding virtual servers. For more information about forwarding virtual servers, see the ***BIG-IP Controller Reference Guide****.*

There are three configuration options on the BIG-IP Controller that you can use to control network access, and you need to identify which method is suitable for your needs:

- **Network Address Translation (NAT)**
  A network translation address provides a routable alias IP address that a node can use as its source IP address when making or receiving connections to clients on the external network. You can configure a unique NAT for each node address included in a virtual server mapping.

  NATs do not support port translation, and are not appropriate for FTP. You cannot define a NAT if you configure a default SNAT.

- **Secure Network Address Translation (SNAT)**
  A secure network address translation provides functionality similar to that of firewalls. A SNAT defines a routable alias IP address that one or more nodes can use as a source IP address only when making connections to hosts on the external network. SNAT addresses support port translation, and they also prevent hosts on the external network from connecting directly to the node.

  SNAT only supports TCP and UDP. SNAT also features support for both passive and active FTP. You cannot define a NAT if you define a default SNAT.

- **IP forwarding**
  IP forwarding does not translate node addresses. Instead, it simply exposes the node's IP address to the BIG-IP Controller's external network so that clients can use it as a standard routable address. When you turn IP forwarding on, the BIG-IP Controller acts as a router when it receives connection requests for node addresses. IP forwarding itself does not provide security features, but you can use the IP filter feature to implement a layer of security (see ***Setting up IP forwarding*** on page 4-15) that can help protect your nodes.

WARNING

NATs and SNATs do not support the NT Domain or CORBA IIOP. Instead of using NATs or SNATs, you need to configure IP forwarding (see ***Setting up IP forwarding*** on page 4-15*).*

Defining a standard network address translation (NAT)

When you define standard network address translations (NATs), you need to create a separate NAT for each node that requires a NAT. You also need to use unique IP addresses for NAT addresses; a NAT IP address cannot match an IP address used by any virtual or physical servers in your network. You can configure a NAT with the Configuration utility or from the command line.

To configure a NAT using the Configuration utility

1. In the navigation pane, click **NATs**.
   The Network Address Translations screen opens.
2. On the toolbar, click **Add NAT**.
   The Add Nat screen opens.
3. Use the fields provided on the Add Nat screen to configure a NAT.

   Configuration note

   - For additional information about the options on this screen, click the **Help** button.

To configure a NAT from the command line

The **bigpipe nat** command defines one NAT for one node address.

```
bigppipe nat <node addr> to <NAT addr>
```

Defining a secure network address translation (SNAT)

When you define secure network address translations (SNATs), you can assign a single SNAT address to multiple nodes. Note that a SNAT address does not necessarily have to be unique; for example, it can match the IP address of a virtual server.

SNAT addresses have global properties that apply to all SNATs that you define in the BIG-IP Controller configuration as well as to the SNAT mappings you define. You can configure SNATs in the Configuration utility or from the command line.

Setting SNAT global properties

The SNAT feature supports three global properties that apply to all SNAT addresses:

- **Connection limits**
  The connection limit applies to each node that uses a SNAT, and each individual SNAT can have a maximum of 50,000 simultaneous connections.
- **TCP idle connection timeout**
  This timer defines the number of seconds that TCP connections initiated using a SNAT address are allowed to remain idle before being automatically disconnected.
- **UDP idle connection timeout**
  This timer defines the number of seconds that UDP connections initiated using a SNAT address are allowed to remain idle before being automatically disconnected. This value should not be set to **0**.

To configure SNAT global properties from the Configuration utility

1. In the navigation pane, click **Secure NATs**.
   The Secure Network Address Translations screen opens.
2. In the Secure Network Address Translation screen, configure a SNAT.

   Configuration notes

   - To turn connection limits off, type **0** in the **Connection Limit** box to turn connection limits off. If you turn connection limits on, keep in mind that each SNAT can support only 50,000 simultaneous connections.
   - The **UDP Idle Connections** value should not be set to **0**.
   - For additional information about the options on this screen, click the **Help** button.

To configure SNAT global properties from the command line

Configuring global properties for a SNAT requires that you enter three **bigpipe** commands. The following command sets the maximum number of connections you want to allow for each node using a SNAT.

```
bigpipe snat limit <value>
```

The following commands set the TCP and UDP idle connection timeouts:

```
bigpipe snat timeout tcp <seconds>
bigpipe snat timeout udp <seconds>
```

Configuring SNAT address mappings

Once you have configured the SNAT global properties, you can configure SNAT address mappings. The SNAT address mappings define each SNAT address, and also define the node or group of nodes that uses the SNAT address. Note that a SNAT address does not necessarily have to be unique; for example, it can match the IP address of a virtual server. A SNAT address cannot match an address already in use by a NAT or another SNAT address.

To configure a SNAT mapping using the Configuration utility

1. In the navigation pane, click **Secure NATs**.
   The Secure Network Address Translations screen opens.
2. On the toolbar, click **Add SNAT**.
   The Add SNAT screen opens.
3. To Configure the SNAT, fill in the fields on the screen.

   Configuration note

   - For additional information about the options on this screen, click the **Help** button.

To configure a SNAT mapping from the command line

The **bigpipe snat** command defines one SNAT for one or more node addresses.

```
bigpipe snat map <node addr>... <node addr> to <SNAT addr>
```

For example, the command below defines a secure network address translation for two nodes:

```
bigpipe snat map 192.168.75.50 192.168.75.51 to 192.168.100.10
```

Setting up IP forwarding

If you do not want to translate addresses with a NAT or SNAT, you can use the IP forwarding configuration option. IP forwarding is an alternate way of allowing nodes to initiate or receive direct connections from the BIG-IP Controller's external network. IP forwarding exposes all of the node IP addresses to the external network, making them routable on that network. If your network uses the NT Domain or CORBA IIOP, IP forwarding is an option for direct access to nodes.

To set up IP forwarding, you need to complete two tasks:

- **Turn IP forwarding on**
  The BIG-IP Controller uses a system control variable to control IP forwarding, and its default setting is **off**.
- **Verify the routing configuration**
  You probably have to change the routing table for the router on the BIG-IP Controller's external network. The router needs to direct packets for nodes to the BIG-IP Controller, which in turn directs the packets to the nodes themselves.

Turning on IP forwarding

IP forwarding is a property of the BIG-IP Controller system, and it is controlled by the system control variable **net.inet.ip.forwarding**.

To set the IP forwarding system control variable using the Configuration utility

1. In the navigation pane, click the BIG-IP Controller icon.
   The BIG-IP System Properties screen opens.
2. On the toolbar, click **Advanced Properties**.
   The BIG-IP System Control Variables screen opens.
3. Check the **Allow IP Forwarding** box.

Configuration note

- For additional information about the options on this screen, click the **Help** button.

To set the IP forwarding system control variable from the command line

Use the standard **sysctl** command to set the variable. The default setting for the variable is **0**, which is **off**. You want to change the setting to **1**, which is **on**:

```
sysctl -w net.inet.ip.forwarding=1
```

To permanently set this value, you can use a text editor, such as **vi** or **pico**, to manually edit the **/etc/rc.sysctl** file. For additional information about editing this file, see the ***BIG-IP Controller Reference Guide***, *BIG-IP Controller System Control Variables.*

Addressing routing issues for IP forwarding

Once you turn on IP forwarding, you probably need to change the routing table on the default router. Packets for the node addresses need to be routed through the BIG-IP Controller. For details about changing the routing table, refer to your router's documentation.

Glossary

attributes

An attribute is a variable that the cache statement uses to direct requests. Attributes can be either required or optional.

BIG-IP active unit

In a redundant system, the controller which currently load balances connections. If the active unit in the redundant system fails, the standby unit assumes control and begins to load balance connections.

BIG-IP web server

The web server that runs on a BIG-IP Controller and hosts the Configuration utility.

bigpipe

A utility that provides command line access to the BIG-IP Controller.

BIG/stat

A statistical monitoring utility that ships on the BIG-IP Controller. This utility provides a snap-shot of statistical information.

BIG/top

A statistical monitoring utility that ships on the BIG-IP Controller. This utility provides real-time information.

big3d

A monitoring utility that collects metrics information about paths between a BIG-IP Controller and a specific local DNS server. The **big3d** utility runs on BIG-IP Controllers and it forwards metrics information to a 3-DNS Controller.

BIND (Berkeley Internet Name Domain)

The most common implementation of DNS, which provides a system for matching domain names to IP addresses.

cacheable content determination

Determines the type of content you cache on the basis of any combination of elements in the HTTP header.

cacheable content expression

An expression that determines, based on evaluating variables in the HTTP header of the request, whether or not a BIG-IP Cache Controller directs a given request to a cache server or to an origin server. Any content that does not meet the criteria in the cacheable content expression is deemed non-cacheable.

cache_pool

Specifies a pool of cache servers to which requests are directed in a manner that optimizes cache performance. The BIG-IP Cache Controller directs all requests bound for your origin server to this pool, unless you have configured the hot content load balancing feature and the request is for **hot** (frequently requested) content. See also *hot* and *origin server.*

chain

A series of filtering criteria used to restrict access to an IP address. The order of the criteria in the chain determines how the filter is applied, from the general criteria first, to the more detailed criteria at the end of the chain.

content affinity

Ensures that a given subset of content remains associated with a given cache server to the maximum extent possible, even when cache servers become unavailable, or are added or removed. This feature also maximizes efficient use of cache memory.

content demand status

A measure of the frequency with which content in a given **hot** content subset is requested over a given **hit_period**. Content demand status is either **hot**, in which case the number of requests for content in the hot

content subset during the most recent **hit_period** has exceeded the **hot_threshold**, or **cool**, in which case the number of requests during the most recent hit period is less than the **cool_threshold**. See also *cool, cool_threshold, hit_period, hot, hot content subset,* and *hot_threshold.*

content_hash_size

Specifies the number of units, or hot content subsets, into which the content is divided when determining whether content is **hot** or **cool**. The requests for all content in a given subset are summed, and a state (hot or cool) is assigned to each subset. The **content_hash_size** should be within the same order of magnitude as the actual number of requests possible. For example, if the entire site is composed of 500,000 pieces of content, a **content_hash_size** of 100,000 is typical.

If you specify a value for **hot_pool**, but do not specify a value for this variable, the cache statement uses a default hash size of 10 subsets. See also *cool, hot,* and *hot content subset.*

cookie persistence

Cookie persistence is a mode of persistence you can configure on the BIG-IP Controller where the controller stores persistent connection information in a cookie.

cool

A term used to describe content demand status when using hot content load balancing. See also *content demand status, hot,* and *hot content load balancing.*

cool_threshold

Specifies the maximum number of requests for given content that will cause that content to change from hot to cool at the end of the hit period.

If you specify a variable for hot_pool, but do not specify a value for this variable, the cache statement uses a default cool threshold of 10 requests. See also *cool, hit_period,* and *hot.*

default wildcard virtual server

A virtual server that has an IP address and port number of **0.0.0.0:0**. This virtual server accepts all traffic which does not match any other virtual server defined in the configuration.

destination processing

Destination processing means that the interface rewrites the destination address of an incoming packet.

destination translation

Included in destination processing, destination translation means that the interface rewrites the destination address of an incoming packet. See also *destination processing*.

dynamic load balancing

Dynamic load balancing modes use current performance information from each node to determine which node should receive each new connection. The different dynamic load balancing modes incorporate different performance factors.

dynamic load balancing modes

Dynamic load balancing modes base connection distribution on live data, such as current server performance and current connection load.

dynamic site content

A type of site content that is automatically generated each time a user accesses the site. Examples are current stock quotes or weather satellite images.

EAV service check

A service check feature that uses an external program to determine if a node is **up** or **down** based on whether the node returns specific content.

EAV service check is only one of the three types of service checks available on a BIG-IP Controller. See also *service check*, and *external service checker program*.

ECV service check

A service check feature that allows you to determine if a node is up or down based on whether the node returns specific content. ECV service check is only one of the three types of service checks available on a BIG-IP Controller. See also *service check*.

Extended Application Verification (EAV)

A service check feature that uses an external program to determine if a node is **up** or **down** based on whether the node returns specific content.

Extended Content Verification (ECV)

A service check feature that allows you to determine if a node is **up** or **down** based on whether the node returns specific content.

external interface

A network interface on the BIG-IP Controller configured to process destination requests. In a basic configuration, this interface has the administration ports locked down. In a normal configuration, this is typically a network interface on which external clients request connections to internal servers.

external service checker program

A custom program that performs a service check on behalf of the BIG-IP Controller. See also, *EAV service check*.

F-Secure SSH

An encryption utility that allows secure shell connections to a remote system.

fail-over

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-over cable

The cable that directly connects the two controller units together in a redundant system.

Fastest mode

A dynamic load balancing mode that bases connection distribution on which server currently exhibits the fastest response time to node pings.

FDDI (Fiber Distributed Data Interface)

A multi-mode protocol for transmitting data on optical-fiber cables up to 100 Mbps.

First-Time Boot utility

A utility that walks you through the initial system configuration process. The First-Time Boot utility runs automatically when you turn on a controller for the first time.

forward proxy caching

A configuration, in which a BIG-IP Cache Controller redundant system uses content-aware traffic direction to enhance the efficiency of an array of cache servers storing Internet content for internal users.

hit_period

In products that support BIG-IP Cache, the **hit_period** specifies the period in seconds over which to count requests for particular content before determining whether to change the state (hot or cool) of the content.

If you specify a value for **hot_pool**, but do not specify a value for this variable, the cache statement uses a default hit period of 10 seconds. See also *cool*, *hot*, and *hot_pool*.

host

A network server which manages one or more virtual servers that the 3-DNS Controller uses for load balancing.

hot

In products that support BIG-IP Cache, **hot** is a term used to define frequently requested content based on the number of requests in a given time period for a given hot content subset. See also *hot content subset*.

hot_pool

A designated group of cache servers to which requests are load balanced when the requested content is hot. If a request is for hot content, the BIG-IP Cache Controller redundant system directs the request to this pool.

hot content load balancing

Identifies **hot**, or frequently requested, content on the basis of number of requests in a given time period for a given hot content subset. A hot content subset is different from, and typically smaller than, the content subsets used for content striping. Requests for hot content are redirected to a cache server in the hot pool, a designated group of cache servers. This feature maximizes the use of cache server processing power without significantly affecting the memory efficiency gained by cacheable content determination. See also *hot*, *hot content subset*, and *hot_pool*.

hot content subset

A hot content subset is different from, and typically smaller than, the content subsets used for cacheable content determination. This is created once content has been determined to be **hot** and is taken or created from the content subset. See also *cacheable content determination.*

hot_threshold

Specifies the minimum number of requests for content in a given hot content subset that will cause that content to change from **cool** to **hot** at the end of the period.

If you specify a value for **hot_pool**, but do not specify a value for this variable, the cache statement uses a default hot threshold of 100 requests. See also *cool*, *hot*, *hot content subset*, and *hot_pool*.

ICMP (Internet Control Message Protocol)

An Internet communications protocol used to determine information about routes to destination addresses, such as virtual servers managed by BIG-IP Controllers and 3-DNS Controllers.

intelligent cache population

Allows caches to retrieve content from other caches in addition to the origin web server. This feature is useful only when working with non-transparent cache servers, which can receive requests that are destined for the cache servers themselves, as opposed to transparent cache servers, which can intercept requests destined for a web server but cannot themselves receive requests. Intelligent cache population minimizes the load on the origin web server and speeds cache population. See also *non-transparent cache server* and *transparent cache server.*

internal interface

A network interface on the BIG-IP Controller configured to process source requests. In a basic configuration, this interface has the administration ports open. In a normal configuration, this is typically a network interface which handles connections from internal servers.

iQuery

A UDP based protocol used to exchange information between BIG-IP Controllers and 3-DNS Controllers. The iQuery protocol is officially registered for port 4353.

last hop

A last hop is the previous hop a connection took to get to the BIG-IP Controller. You can configure the BIG-IP Controller to send packets back to the device from which they originated when that device is part of a last hop pool.

Least Connections mode

A dynamic load balancing mode that bases connection distribution on which server currently manages the fewest open connections.

load balancing mode

A particular method of determining how to distribute connections across an array.

loopback adapter

A loopback adapter is a software interface that is not associated with an actual network card. The nPath routing configuration requires you to configure loopback adapters on servers.

MAC (Media Access Control)

A protocol that defines the way workstations gain access to transmission media, most widely used in reference to LANs. For IEEE LANs, the MAC layer is the lower sublayer of the data link layer protocol.

MAC Address

An address used to represent hardware devices on an Ethernet network.

member

A reference to a node when it is included in a particular virtual server mapping. Virtual server mappings typically include multiple member nodes.

mirroring

A feature on the BIG-IP Controller that preserves connection and persistence information in a BIG-IP Controller redundant system.

miss request

A miss request results from a request for content a cache does not have.

named

The name server daemon, which manages domain name server software.

NAT (Network Address Translation)

An alias IP address that identifies a specific node managed by the BIG-IP Controller to the external network.

node

A specific combination of an IP address and port number associated with a server in the array managed by the BIG-IP Controller.

node address

The IP address associated with one or more nodes. This IP address can be the real IP address of a network server, or it can be an alias IP address on a network server.

node alias

A node address that the BIG-IP Controller uses to verify the status of multiple nodes. When the BIG-IP Controller uses a node alias to check node status, it pings the node alias. If the BIG-IP Controller receives a response to the ping, it marks all nodes associated with the node alias as **up**, and if it does not receive a response to the ping, the BIG-IP Controller marks all nodes associated with the node alias as **down**.

node ping

A feature that the BIG-IP Controller uses to determine whether nodes are **up** or **down**. Node ping sends standard echo pings to servers and transparent devices. If the server or device responds to the ping, it marks the related nodes **up**. If the server or device does not respond to the ping, it marks the related nodes **down**.

node port

The port number or service name hosted by a specific node.

node status

Node status indicates whether a node is **up** and available to receive connections, or **down** and unavailable. The BIG-IP Controller uses the node ping and service check features to determine node status.

non-cacheable content

Content that is not identified in the cacheable content condition part of a cache rule statement. See also *cacheable content condition*.

non-transparent cache server

Cache servers that can receive requests that are destined for the cache servers themselves.

origin server

The web server on which all original copies of your content reside.

origin_pool

Specifies a pool of servers that contain original copies of all content. Requests are load balanced to this pool when any of the following are true: the requested content is not cacheable, no cache server is available, or the BIG-IP Cache Controller redundant system is redirecting a request from a cache server that did not have the requested content.

Observed mode

A dynamic load balancing mode that bases connection distribution on a combination of two factors: the server that currently hosts the fewest connections, and also has the fastest response time.

persistence

A series of related connections received from the same client, having the same session ID. When persistence is turned on, a controller sends all connections having the same session ID to the same node instead of load balancing the connections.

pool

A pool is a group of devices that you want the Big_IP Cache Controller redundant system to load balance.

port

A number that is associated with a specific service supported by a host. Refer to the Services and Port Index for a list of port numbers and corresponding services.

port-specific wildcard virtual server

A wildcard virtual server address that uses a port number other than **0**.

Predictive mode

A dynamic load balancing mode that bases connection distribution on a combination of two factors: the server that currently hosts the fewest

connections, but also has the fastest response time. Predictive mode also ranks server performance over time, and passes connections to servers which exhibit an improvement in performance rather than a decline.

Priority mode

A static load balancing mode that bases connection distribution on server priority levels. The BIG-IP Controller distributes connections in a round robin fashion to all nodes in the highest priority group. If all the nodes in the highest priority group become unavailable, the BIG-IP Controller begins to pass connections to nodes in the next lower priority group.

rate class

A rate class determines the volume of traffic allowed through a rate filter.

ratio

A parameter that assigns a weight to a virtual server for load balancing purposes.

Ratio mode

The Ratio load balancing mode distributes connections across an array of virtual servers in proportion to the ratio weights assigned to each individual virtual server.

receive expression

A receive expression is the text string that the BIG-IP Controller looks for in the web page returned by a web server during an extended content verification (ECV) service check.

redundant system

A pair of controllers that are configured for fail-over. In a redundant system, there are two controller units, one running as the active unit and one running as the standby unit. If the active unit fails, the standby unit takes over and manages connection requests.

remote administrative IP address

An IP address from which a controller allows shell connections, such as Telnet or SSH.

remote server acceleration

A configuration in which a BIG-IP Cache Controller redundant system uses content-aware traffic direction to enhance the efficiency of an array of cache servers that cache content for a remote web server.

Round Robin mode

A static load balancing mode that bases connection distribution on a set server order. Round Robin mode sends a connection request to the next available server in the order.

send string

A send string is the request that the BIG-IP Controller sends to the web server during an extended content verification (ECV) service check.

service check

A BIG-IP Controller feature that determines whether a node is up or down. When a BIG-IP Controller issues a service check, it attempts to connect to the service hosted by the node. If the connection is successful, the node is **up**. If the connection fails, the node is **down**. See also *ECV service check*, and *EAV service check*.

SNAT (Secure Network Address Translation)

A SNAT is a feature you can configure on the BIG-IP Controller. A SNAT defines a routable alias IP address that one or more nodes can use as a source IP address when making connections to hosts on the external network.

SNMP (Simple Network Management Protocol)

The Internet standard protocol, defined in STD 15, RFC 1157, developed to manage nodes on an IP network.

sod (switch over daemon)

A daemon that controls the fail-over process in a redundant system.

source processing

Source processing means that the interface rewrites the source of an incoming packet.

standby unit

A controller in a redundant system that is always prepared to become the active unit if the active unit fails.

stateful site content

Content that maintains dynamic information for clients on an individual basis and is commonly found on e-commerce sites. For example, a site that allows a user to fill a shopping cart, leave the site, and then return and purchase the items in the shopping cart at a later time has stateful site content which retains the information for that client's particular shopping cart.

static load balancing modes

Static load balancing modes base connection distribution on a pre-defined list of criteria; it does not take current server performance or current connection load into account.

static site content

A type of site content that is stored in HTML pages, and changes only when an administrator edits the HTML document itself.

sticky mask

A sticky mask is a special IP mask that you can configure on the BIG-IP Controller. This mask optimizes sticky persistence entries by grouping more of them together.

stripes

In products that support caching, **stripes** are cacheable content subsets distributed among your cache servers.

transparent cache server

A cache server that can intercept requests destined for a web server, but are incapable of receiving requests.

transparent node

A node that appears as a router to other network devices, including the BIG-IP Controller.

virtual address

An IP address associated with one or more virtual servers managed by the BIG-IP Controller.

virtual port

The port number or service name associated with one or more virtual servers managed by the BIG-IP Controller. A virtual port number should be the same TCP or UDP port number to which client programs expect to connect.

virtual server

A specific combination of a virtual address and virtual port, associated with a content site that is managed by a BIG-IP Controller or other type of host server.

virtual server mapping

The group of nodes across which a virtual server load balances connections for a given site.

watchdog timer card

A hardware device that monitors the BIG-IP Controller for hardware failure.

wildcard virtual server

A virtual server that uses an IP address of **0.0.0.0**. A wildcard virtual server accepts connection requests for destinations outside of the local network. Wildcard virtual servers are included only in Transparent Node Mode configurations.

F5 Networks

BIG·IP®

Local Traffic Controller

Index

B

C

D

E

F

H

I

L

M

N

P

R

S

T

U

V

W