CLI Reference Guide

CLI Reference Guide

This page was not added to the PDF due to the following tag(s): article:topic-guide

About This CLI Reference Guide

Overview

This About page contains general information about this guide, including the audience, typographic conventions, and how to search the content.

Audience

This guide is intended for experienced network administrators and network architects who understand your organization's existing TCP/IP network and who need to configure load balancing (reverse proxy) or a forward proxy using LineRate.

Conventions

This guide uses the following symbols and typographic conventions.

Convention	Definition
`Monospaced bold`	Text in a monospaced bold font represents commands or other text that you type exactly as you see it.
`<angle brackets>`	Text in a monospaced bold font inside angle brackets represents a placeholder that describes what you must type.
`[square brackets]`	Text in a monospaced bold font inside square brackets represents an optional command or option.
Monospaced	Text in a monospaced font represents output or results the system displays.
Bold	Text in bold shows keys to press and items to select or click, such as menu items or buttons.
	Shows the beginning of a procedure.
 **Caution**	Cautions contain critical information about configuring your system or data.
 **Note**	Notes contain important information that may affect how you install or configure your system.
 **Tip**	Tips contain best practices or useful information to help you when configuring your system.
	Shows that the content is for advanced users.

Example IP Addresses

Throughout this guide, we use example IP addresses for both internal (private) and external (public) uses.

For private addresses, we use the IP addresses designated in RFC 1918:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

For public addresses, we use the IP addresses designated for documentation in RFC 5737:

- 192.0.2.0/24 (TEST-NET-1)
- 198.51.100.0/24 (TEST-NET-2)
- 203.0.113.0/24 (TEST-NET-3)

Searching the Guide

The search box at the top-right of each page lets you enter a term or phrase to search for. By default, the system searches all pages in the LineRate content. Searches are not case sensitive. By default, searches find plurals and other matches from word stems, such as tests, testing, tested, and tester if you search for test.

You can search for a single term such as:

```
interface
```

Or

```
certificate
```

You can also search for an exact phrase surrounded by double quotes such as:

```
"real server"
```

Or

```
"IP address"
```

Relevance Level

By default, the system sorts the search results by relevance. The relevance is determined by a weighting algorithm that takes into consideration the page title, content, tags, and attachments. The relevance is also affected by the page rating (thumb up or down) and by how often other users select a page to view from similar searches.

Searches can return a large number of results. You can narrow your searches a number of ways by:

- Limiting your search to a specific tree
- Using term modifiers
- Using Boolean operators

Limiting a Search to Specific Tree

If you only want to search one area or tree of a guide, you can limit your search to that tree. For example, if you only want to search the Configure Command tree of the 2.2 Release of the CLI Reference Guide for the term "interface," you can enter your search like this:

```
+(path:300Release_2.2/200CLI_Reference_Guide/Configure_Commands/*) AND interface
```

You can further narrow the search using the term modifiers and Boolean operators (described below):

```
+(path:300Release_2.2/200CLI_Reference_Guide/Configure_Commands/*) AND interface
    AND CARP

+(path:300Release_2.2/*) AND load AND balancer
```

For a tree-specific search, words in quotes are not treated as a specific phrase. The search does an OR search for any words in quotes, so you may not want to use quotes and use AND instead, as shown in the example above.

A few steps to help with this type of search:

1. Navigate to the tree you want to search.
2. In your browser's address bar, copy the address of the page.
   - You only need the part after the "https://docs.lineratesystems.com/".
3. Using the syntax example above, type in your search and paste in the path of the page you want to search.

Term Modifiers

The search supports modifying query terms to provide a wide range of searching options.

Wildcard Searches

The guides support single- and multiple-character wildcard searches with single terms (not within phrase queries).

To perform a single-character wildcard search, use the ? symbol.

To perform a multiple-character wildcard search, use the * symbol.

The single-character wildcard search looks for terms that match that with the single character replaced. For example, to search for "text" or "test" you can use the search:

```
te?t
```

The multiple-character wildcard search looks for 0 or more characters. For example, to search for test, tests or tester, you can use the search:

```
test*
```

You can also use the wildcard searches in the middle of a term.

```
te*t
```

You cannot use a * or ? symbol as the first character of a search.

Fuzzy Searches

The guide supports fuzzy searches based on the Levenshtein Distance or Edit Distance algorithm. To do a fuzzy, search use the tilde ~ symbol at the end of a single word. Fuzzy searches work for multiple characters. For example, to search for a term similar in spelling to "roam" use the fuzzy search:

```
roam~
```

This search will find terms like foam and roams.

You can add an optional parameter to specify the required similarity. The value is between 0 and 1. With a value closer to 1, only terms with a higher similarity will be matched. For example:

```
roam~0.6
```

The default is 0.5.

Proximity Searches

The guide supports finding words that are within a specific distance from each other. To do a proximity search, use the tilde ~ symbol at the end of a phrase. For example, to search for a "feature" and "standard" within 10 words of each other in a document use the search:

```
"feature standard"~10
```

Boosting a Term

The guide provides the relevance level of matching documents based on the terms found. To boost a term, use the caret ^ symbol with a boost factor (a number) at the end of the term you are searching. The higher the boost factor, the more relevant the term will be.

Boosting allows you to control the relevance of a document by boosting its term. For example, if you are searching for:

```
mindtouch search
```

and you want the term "mindtouch" to be more relevant boost it using the ^ symbol along with the boost factor next to the term. You would type:

```
mindtouch^4 search
```

This will make documents with the term mindtouch appear more relevant. You can also boost phrases as in the example:

```
"mindtouch search"^4 "Apache"
```

By default, the boost factor is 1. Although the boost factor must be positive, it can be less than 1 (e.g. 0.2)

Boolean Operators

Boolean operators allow terms to be combined through logic operators. MindTouch supports AND, +, OR, NOT, and - as Boolean operators.

 Boolean operators must be ALL CAPS.

OR

The OR operator is the default conjunction operator. This means that if there is no Boolean operator between two terms, the OR operator is used. The OR operator links two terms and finds a matching document if either of the terms exist in a document. This is equivalent to a union using sets. The symbol || can be used in place of the word OR.

To search for documents that contain either "mindtouch search" or just "mindtouch" use the query:

```
"mindtouch search" mindtouch
```

or

```
"mindtouch search" OR mindtouch
```

AND

The AND operator matches documents where both terms exist anywhere in the text of a single document. This is equivalent to an intersection using sets. You can use the symbol && in place of the word AND.

To search for documents that contain "mindtouch search" and "Advanced" use the query:

```
"mindtouch search" AND "Advanced"
```

+

The + (required operator) requires that the term after the + symbol exist somewhere in a document.

To search for documents that must contain "search" and may contain "advanced," use the query:

```
+search advanced
```

NOT

The NOT operator excludes documents that contain the term after NOT. This is equivalent to a difference using sets. You can use the symbol ! in place of the word NOT.

To search for documents that contain "mindtouch search" but not "Advanced" use the query:

```
"mindtouch search" NOT "Advanced"
```

 The NOT operator cannot be used with just one term. For example, the following search will return no results:

NOT "mindtouch search"

Grouping

The guide supports using parentheses to group clauses to form sub queries. This can be very useful if you want to control the Boolean logic for a query.

To search for either "mindtouch" or "search" and "advanced" use the query:

```
(mindtouch OR search) AND advanced
```

This eliminates any confusion and makes sure you that website must exist and either term mindtouch or search may exist.

Escaping Special Characters

The Guide supports escaping special characters that are part of the query syntax. The current list of special characters is:

+ - && || ! ( ) { } [ ] ^ " ~ * ? : \

To escape these character use the \ before the character. For example, to search for (1+1):2 use the query:

```
\(1\+1\)\:2
```

Legal Notices

Copyright

Trademarks

Deprecated CLI Commands

Overview

This section describes the commands deprecated in release 2.2 and previous releases.

Deprecated CLI Commands

No CLI commands are deprecated in LineRate release 2.2.

The following CLI commands are deprecated in LineRate release 2.0:

Deprecated command	Replaced by	Notes
`load-balancer`	proxy	If you load a configuration from release 1.6 or earlier that uses the load-balancer command on a 2.x system and save the configuration, the system automatically migrates the command to the replacement command.
`show load-balancer`	show proxy	

CLI General Reference

Overview

This section describes information that applies throughout the rest of the *CLI Reference Guide*. It contains information about the CLI parameter types and other information that applies to using many CLI commands in the following sections:

- Technical Support
- Parameter Types
- IP Address Notation

Also see the following:

- Accessing the LineRate CLI
- Using the Command Line Interface

Technical Support

Support tools are available to help you answer your questions whenever and wherever you need help. From the documentation to the global technical community you can collaborate with on DevCentral, LineRate self-service tools help you solve issues quickly and proactively.

The LineRate Support page can help you find the resources you need.

Parameter Types

1. Parameter Type Definitions

Parameter Type Definitions

The table below defines the parameter types used throughout the *CLI Reference Guide*.

Parameter type	Definition
Integer	An optional + or - sign followed by one or more digits (0-9).
Real	Everything an Integer can be and can also include a decimal point and more digits after the decimal point.
Word	A string that can be made up only of letters (upper or lower), digits, dashes, and periods. Quotes around a Word are not accepted.
String	Can be a Word, in which case no quotation marks are needed. Or, can be a string of any characters, but must be surrounded by double quotation marks.
EscString	Can be a string and can contain simple escaped characters like \", \r, \n, \t.
IPv4Addr	IP address in IPv4 dotted decimal notation (aaa.bbb.ccc.ddd). Depending on the context, this could be a single host address, an IP subnet, or an IP subnet mask.
IPv6Addr	IP address in IPv6 notation per http://tools.ietf.org/html/rfc2373 section 2.2. For IPv6, this always refers to a single host address.
IPAddr	Either an IPv4Addr or IPv6Addr.
IPv4AddrMask	IPv4 host address and mask or IP subnet (that is, range of hosts) in CIDR notation (aaa.bbb.ccc.ddd/mm). See http://en.wikipedia.org/wiki/CIDR.
IPv6AddrMask	IPv6 host address and mask or IP subnet (that is, range of hosts) in CIDR notation, see http://tools.ietf.org/html/rfc2373 section 2.3.
IPAddrMask	Either an IPv4AddrMask or IPv6AddrMask.

IP Address Notation

LineRate supports both IPv4 and IPv6. You can specify the IP address and subnet mask in any of the following formats:

- **`192.0.2.1/24`**— example of an IPv4 address with a 24-bit subnet mask using CIDR notation.
- **`192.0.2.1 255.255.255.0`**—equivalent to above using net mask notation.
- **`2001:DB8::/64`**—example of an IPv6 address with a 64 bit subnet mask using CIDR notation.

If you need more information about IP addresses and subnet masks, see these sites for more information:

- http://en.wikipedia.org/wiki/IP_address
- http://en.wikipedia.org/wiki/CIDR_notation

Configure Commands

This page was not added to the PDF due to the following tag(s): article:topic-guide

Boot Mode Commands

boot

Configure version to use to reload the system.

Use

Whenever you upgrade LineRate, the system retains the previous version, including all configuration settings at the time of the upgrade. If needed, you can reload any previous version by setting the version you want to reload using the boot command.

To see the previous versions available, use the following commands:

```
bash "ls /base/persist"
```

 Be sure to use **`write`** to save your change after using the boot command, then use **`reload`** to actually reload to the specified version.

Default Setting

Current system software version

Command Mode

configure

Syntax

```
boot system <version>
```

Configure version of LineRate to reload

Parameter	Type	Description
version	String	Version to reload

Related Commands

[Reload Mode Commands]

REST API Reference - [boot]

Certificate Mode Commands

Use the following commands to configure certificates.

certificate

Create or modify a certificate or certificate bundle.

Use

Use to configure certificates for use with SSL. Example uses of certificates are for performing SSL offload by attaching to a virtual-ip or for management access to the system by attaching to the REST server.

To set up certificates, you must have access to your certificate files. The LineRate software supports PEM format certificates. You must configure the primary certificate that identifies the virtual IP on LineRate and the chain of certificates that lead to the root certificate that a client will trust.

We recommend giving each certificate a meaningful name that helps identify the certificate. For example, you might use the domain name or security settings in the name.

Default Setting

By default, the system creates a key, certificate, and SSL profile, each called self-signed, that you can use to test the SSL function.

Command Mode

configure

Syntax

```
[no] certificate <cert_name>
```

Create or modify a single primary or chain certificate

```
[no] certificate bundle <bundle_name>
```

Create or modify a certificate bundle of chain certificates

Parameter	Type	Description
bundle_name	Word	Name of the certificate bundle
cert_name	Word	Name of the certificate

Example

See Setting Up Certificates for SSL Termination.

Related Commands

Key Mode Commands

SSL Mode Commands

REST API Reference - certificateBundles

REST API Reference - certificates

pem-format

Specify the certificate content in a PEM-encoded (base64) string.

Use

Use to paste certificate text into either a single certificate or a certificate bundle.

After entering the command, press **Enter** to paste certificate text. If needed, you can do the paste several times to add multiple chain certificates to a bundle. Type **`quit`** to finish.

Default Setting

None

Command Mode

config-certificate

Syntax

`pem-format`

Specify the certificate content in a PEM-encoded (base64) encoded string.

Example

See Setting Up Certificates for SSL Termination.

Related Commands

Key Mode Commands

SSL Mode Commands

Exit Command (config mode)

Use

Use to exit the current config mode. When in config mode, this takes you back one mode level. The config mode prompt includes the config mode you are in. When you first go into config mode, the prompt looks like this:

```
LROS(config)#
```

For information about modes, see Command Line Modes.

Default Setting

None

Command Mode

config

Syntax

`exit`

Examples

If you are in configure real server HTTP service mode for a real server called rs1:

```
LROS(config-rserver-http:rs1)#
```

then enter exit, you will go back one mode to configure real server mode:

```
LROS(config-rserver-http:rs1)# exit

LROS(config-rserver:rs1)#
```

Failover Mode Commands

failover

Use

Use to create a failover group that you attach to CARP interfaces. Failover groups complement CARP groups. Failover groups provide failover protection for a complete path from clients that use a specific virtual IP to the web servers accessed by that virtual IP. Failover groups provide failover protection for the path on a single LineRate system, while CARP groups provide failover protection for one segment of the path across multiple LineRate systems.

For a failover group example, see Failover Group Example.

Default Setting

None

Command Mode

config

Syntax

```
[no] failover group <group_name>
```

Create or modify a failover group

Parameter	Type	Description
group_name	Word	Name of the failover group

Related Commands

carp

REST API Reference - failover

description

Use

Use to create a description of the failover group.

Default Setting

None

Command Mode

config-failover-group

Syntax

`[no] description <desc>`

Description for this failover group

`no description`

Removes the description for this failover group

Parameter	Type	Description
desc	String	Description string

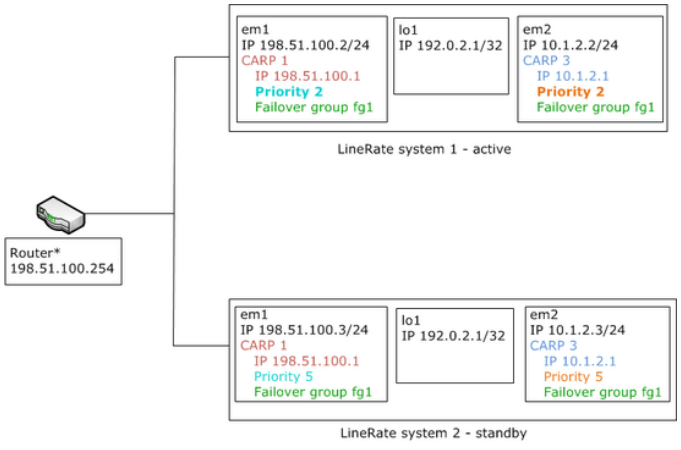
Failover Group Example

Failover Group Overview

You attach failover groups to CARP interfaces. Failover groups complement CARP groups. Failover groups provide failover protection for a complete path from clients that use a specific virtual IP to the web servers accessed by that virtual IP.

Configuring failover groups on CARP groups permits failover for a complete data path between two or more LineRate systems. This example shows an active/standby configuration where CARP is configured on the both the outside interfaces (client access) and the inside interfaces (to your servers). In addition, the same failover group (fg1) is configured on each CARP group on both systems.

In the active/standby configuration (shown below), the CARP groups on one LineRate system are configured as the active groups, based on their lower priority. The other LineRate system is configured identically, except the CARP groups have a higher priority, making that system the standby.



Example active/standby configuration using CARP and a failover group

What You Need

For this configuration, as shown above, you need the following:

- One router
- Two LineRate systems with the following:
  - One loopback interface on each
  - One outside interface on each
  - One inside interface on each
- IP addresses for the following:
  - One for the router
  - One for the virtual IP (will also be the IP address for the loopback interface)
  - One for the outside and inside interfaces (one unique IP for each interface)
  - One for each CARP group

How the Configuration Works

The code example in the next section sets up the following configuration (including some items not shown in the diagram above):

- Each LineRate system is configured as follows:
  - Each inside interface (on em2) has a unique IP address (10.1.2.2/24 and 10.1.2.3/24).
  - Each inside interface is part of CARP group 3 with the shared CARP IP set to 10.1.2.1.
    - The standby interface in each CARP group monitors the status of the active interface, making sure the active interface is working.
    - The priority on the active system is set to 2.
    - The priority of the standby system is set to 5.
  - Each outside interface (on em1) has a unique IP address (198.51.100.2/24 and 198.51.100.3/24).
  - Each outside interface is part of CARP group 1 with the shared CARP IP set to 198.51.100.1.
    - The standby interface in each CARP group monitors the status of the active interface, making sure the active interface is working.
    - The shared CARP IP is used as a next hop gateway address for the router to reach the virtual IP address.
    - The priority on the active system is set to 2.
    - The priority of the standby system is set to 5.
  - The virtual IP address is configured as 192.0.2.1 80.
  - A loopback interface is configured with the virtual IP as its address (192.0.2.1/32).
  - The virtual IP address is attached to a virtual server.
  - The real servers are attached to the virtual server.
  - The default IP route on each LineRate system is set to 0.0.0.0/0 198.51.100.254.
- The router is configured via a static IP route to send requests for destination IP 192.0.2.1 (the VIP) to 198.51.100.1 (the CARP IP).

Requests to the virtual IP 192.0.2.1 are sent to through the router, which sends them to the active system based on its routing and the CARP priority. The active system then sends the requests to the virtual server, to the real server, to the outside interface, then on to the back-end servers. If anything in the path from the active system's em1 interface to the em2 interface fails, the failover group sets both CARP group priorities to 24 (lowest priority and one more than the user-configurable priority range). The standby system will become master for both CARP groups and will accept traffic for 192.0.2.1 (the VIP) using 198.51.100.1 (the CARP IP).

Without the failover group, a failure in the data path (for example em2 fails) would only cause CARP 3 to change from the active system to the standby system. However, the rest of the data path from the router through em1 and lo1 would remain on the originally active system. The traffic would stop at the failed em2 interface. The failover group tells the system to move everything in the data path to the standby system, making it the active system.

The configuration on all LineRate systems meant for CARP failover must have the identical configuration, *except* for the following:

- Host name
- Unique management IP
- Unique IP on each CARP interface
- CARP priority

Annotated Configuration

Below is the configuration that is key to configuring active/standby failover group. We have excluded parts of the configuration to focus only on what is required for this configuration to work. In the section below, we have included the complete show run output, which has the complete configuration for a simple load balancer using an active/standby falilover group for a data path for both systems.

Configuration for LineRate system 1 (active):

Commands	**Description**
`failover group fg1`	Creates the failover group called fg1.

`interface em1`
`  ip address`
`  198.51.100.2/24`
`  carp 1`
`  priority 2`
`  ip 198.51.100.1`
	Sets the unique IP address for the interface em1.
	
	Adds CARP group 1 with a priority of 2 on the em1 interface, making this system the master, and sets the shared CARP IP address.
	
	Attaches the failover group called fg1 to CARP group 1.

attach failover group fg1	
interface em2 **ip address 10.1.2.2/24** **carp 3** **priority 2** **ip 10.1.2.1** **attach failover group fg1**	Sets the unique IP address for the interface em2. Adds CARP group 3 with a priority of 2 on the em2 interface, making this system the master, and sets the shared CARP IP address. Attaches the failover group called fg1 to CARP group 3.
interface lo1 **ip address 192.0.2.1/32**	Sets the IP address for the loopback interface lo1, which is the same as the virtual IP created below.
ip route 0.0.0.0/0 198.51.100.254	Sets the system's default IP route for all traffic to 198.51.100.254, the router's IP address.
real-server base **webserver_base** **real-server rs1** **ip address 10.1.2.101 80** **base webserver_base** **admin-status online**	Creates a real server base (base settings omitted for this example) webserver_base. Creates a real server called rs1 based on webserver_base, assigns 10.1.2.101 as the real server's IP address, and brings the real server online.
virtual-ip base **web_base** **admin-status online**	Creates a virtual IP base (base settings omitted for this example) web_base. Creates a virtual IP called vip1 based on web_base, assigns 192.0.2.1 as the virtual IP address, and brings the virtual IP base online.

`virtual-ip vip1 ip 192.0.2.1 80 base web_base`	

`virtual-server vs1`
`lb-algorithm round-robin`
`service http`
`attach virtual-ip vip1 default`
`attach real-server rs1`
	Creates the virtual server vs1 and attaches the virtual IP vip1 and the real server rs1 to it.
	
	Also sets the load balancing algorithm and service to HTTP

Configuration for LineRate system 2 (standby; configuration differences from system 1 shown in blue):

Commands	**Description**
`failover group fg1`	Creates the failover group called fg1.

`interface em1`
`ip address 198.51.100.3/24`
`carp 1`
`priority 5`
`ip 198.51.100.1`
 	
`attach failover group fg1`
	Sets the unique IP address for the interface em1.
	
	Adds CARP group 1 with a priority of 5 on the em1 interface, making this system the standby, and sets the shared CARP IP address.
	
	Attaches the failover group called fg1 to CARP group 1.

`interface em2` `ip address 10.1.2.3/24`	Sets the unique IP address for the interface em2.

`carp 3`
`priority 5`
`ip 10.1.2.1`
 	
`attach failover group fg1`
	Adds CARP group 3 with a priority of 5 on the em2 interface, making this system the standby, and sets the shared CARP IP address.
	
	Attaches the failover group called fg1 to CARP group 3.

`interface lo1` `ip address 192.0.2.1/32`	Sets the IP address for the loopback interface lo1, which is the same as the virtual IP created below.
`ip route 0.0.0.0/0 198.51.100.254`	Sets the system's default IP route for all traffic to 198.51.100.254, the router's IP address.

`real-server base webserver_base`
 	
`real-server rs1`
`ip address 10.1.2.101 80`
`base webserver_base`
`admin-status online`
	Creates a real server base (base settings omitted for this example) webserver_base.
	
	Creates a real server called rs1 based on webserver_base, assigns 10.1.2.101 as the real server's IP address, and brings the real server online.

`virtual-ip base web_base`
`admin-status online`
 	
`virtual-ip vip1 ip 192.0.2.1 80 base web_base`
	Creates a virtual IP base (base settings omitted for this example) web_base.
	
	Creates a virtual IP called vip1 based on web_base, assigns 192.0.2.1 as the virtual IP address, and brings the virtual IP base online.

`virtual-server vs1` `lb-algorithm round-robin`	Creates the virtual server vs1 and attaches the virtual IP vip1 and the real server rs1 to it.

service http **attach virtual-ip vip1 default** **attach real-server rs1**	Also sets the load balancing algorithm and service to HTTP

Example Show Run Output

The show run output below includes only the sections related to an active/standby CARP configuration.

Configuration for LineRate system 1 (active):

```
Building configuration...
!
hostname example-host-a
!
username admin secret encrypted "$2a$04$7TYufYOKVQ8i8bblVtZ1ierxZXzcH5mR/
    QeaZH8WnWRzVEkPt0MgS" uid 2000
!
failover group fg1
!
interface em1
 ip address 198.51.100.2 255.255.255.0
 carp 1
  priority 2
  ip 198.51.100.1
  attach failover group fg1
!
interface em2
 ip address 10.1.2.2 255.255.255.0
 carp 3
  priority 2
  ip 10.1.2.1
  attach failover group fg1
!
interface lo1
 ip address 192.0.2.1 255.255.255.255
!
ip route 0.0.0.0/0 198.51.100.254
!
real-server base webserver_web
!
real-server rs1
 ip address 10.1.2.101 80
 base webserver_base
 admin-status online
```

```
!
virtual-ip base web_base
 admin-status online
!
virtual-ip vip1 ip 192.0.2.1 80 base web_base
!
virtual-server vs1
 lb-algorithm round-robin
 service http
 attach virtual-ip vip1 default
 attach real-server rs1
```

Configuration for LineRate system 2 (standby):

```
Building configuration...
!
hostname example-host-b
!
username admin secret encrypted "$2a$04$7TYufYOKVQ8i8bblVtZ1ierxZXzcH5mR/
   QeaZH8WnWRzVEkPt0MgS" uid 2000
!
failover group fg1
!
interface em1
 ip address 198.51.100.3 255.255.255.0
 carp 1
  priority 5
  ip 198.51.100.1
  attach failover group 1
!
interface em2
 ip address 10.1.2.2 255.255.255.0
 carp 3
  priority 5
  ip 10.1.2.1
  attach failover group fg1
!
interface lo1
 ip address 192.0.2.1 255.255.255.255
!
ip route 0.0.0.0/0 198.51.100.254
!
real-server base webserver_web
!
real-server rs1
 ip address 10.1.2.101 80
 base webserver_base
 admin-status online
!
virtual-ip base web_base
 admin-status online
!
virtual-ip vip1 ip 192.0.2.1 80 base web_base
!
```

```
virtual-server vs1
 lb-algorithm round-robin
 service http
 attach virtual-ip vip1 default
 attach real-server rs1
```

Forward Proxy Mode Commands

forward-proxy

Create or modify a forward proxy.

Use

The LineRate forward proxy capability provides a proxy function from one network to another. A common use case for a forward proxy is for connections from your private network to the Internet.

A forward proxy lets you insert custom logic created with scripts. Scripts can perform a variety of functions, including gathering usage statistics, redirecting requests to your own cache, blocking of access to specific sites, managing cookies, and much more.

To use a forward proxy effectively, be sure to create and attach a virtual IP that includes the range of Internet IP addresses you want to go through the forward proxy.

When attaching a virtual IP to a forward proxy, the virtual IP must not include any of the system's own IP addresses. For a virtual IP with a single IP address, do not set the virtual IP's IP address to one of the system's own IP addresses. For a virtual IP with a range of addresses, you must ensure that the IP address range does not contain any of the system's own IP addresses. This may mean you need to break the virtual IP into multiple virtual IPs. See Configuring a range for a virtual IP with forward proxy for more detail and an example.

Default Setting

None

Command Mode

config

Syntax

`[no] forward-proxy <forward_proxy_name>`

Create or modify a forward proxy

Parameter	Type	Description
forward_proxy_name	Word	Name of the forward proxy

Related

CLI Reference - Show Forward Proxy Commands

REST API Reference - forwardProxy

admin-status

Bring an object, such as a health monitor, real server, or virtual IP, online or offline. After you create an object, you must bring it online.

Use

You typically set the offline status only when you want to disable the object or block connections to the web server during maintenance or system reconfiguration.

Default Setting

offline

Command Mode

config-fproxy

Syntax

[no] admin-status offline

Take the forward proxy out of service

[no] admin status online

Put the forward proxy in service

no admin status

Remove the configured admin status

attach

Use

Use to attach an object, such as a virtual IP, to the current forward proxy.

 When attaching a virtual IP to a forward proxy, the virtual IP must not include any of the system's own IP addresses. For a virtual IP with a single IP address, do not set the virtual IP's IP address to one of the system's own IP addresses. For a virtual IP with a range of addresses, you must ensure that the IP address range does not contain any of the system's own IP addresses. This may mean you need to break the virtual IP into multiple virtual IPs. See Configuring a range for a virtual IP with forward proxy for more detail and an example.

Default Setting

None

Command Mode

config-fproxy

Syntax

[no] attach virtual-ip <vip>

Configure one or more virtual IPs that this forward proxy will listen on

Parameter	Type	Description
vip	Word	Name of a virtual IP to attach

description

Create a description for the forward proxy.

Use

Use to create a description of the forward proxy use and any other information.

Default Setting

None

Command Mode

config-fproxy

Syntax

`[no] description <desc>`

Description for the forward proxy

`no description`

Remove the description for the forward proxy

Parameter	Type	Description
desc	String	Description string. Enter a maximum of 255 characters.

ip

Configure DSCP settings.

Use

Use to configure how LineRate populates the Differentiated Services Code Point (DSCP) bits as packets pass through to provide quality of service for IP networks. Use the options to configure where the bits are going to or the specific value to use. For the valid values and more information about DSCP, see http://en.wikipedia.org/wiki/Differentiated_Services_Code_Point.

When you enable the copy option, any value that was previously set is changed to 0. Conversely, when you set a value, the copy option is changed to 0, disabling the copy.

Default Setting

None (system does not do anything with the DSCP bits)

Command Mode

None

Syntax

`[no] ip dscp client <value>`

Configure IP packet DSCP settings for flows to the client

`[no] ip dscp client copy`

Copy IP packet DSCP bits from the server to the client

`[no] ip dscp server <value>`

Configure IP packet DSCP settings for flows to the server

`[no] ip dscp server copy`

Copy IP packet DSCP bits from the client to the server

Parameter	Type	Description
value	Integer	Set IP packet DSCP value for flows to the client or server.

mode

Specify the forward proxy mode

Use

Use to configure the forward proxy mode. In release 2.x, the mode is transparent by default and cannot be changed.

Default Setting

transparent

Command Mode

config-fproxy

Syntax

[no] mode transparent

Proxy using the incoming IP address as the destination (default)

service http

Configure HTTP as the service type for the forward proxy.

Use

Sets the forward proxy to be an HTTP proxy, operating at layer 7. You must use this service type to enable scripts to interact with HTTP requests and responses. Use for HTTPS as well, if you want the LineRate to terminate the SSL connection and be able to understand the encrypted HTTP messages.

Default Setting

Service default is http.

Command Mode

config-fproxy

Syntax

[no] service http

Enable HTTP service

keepalive-timeout

Close the forward proxy TCP connection if there is no HTTP activity in this amount of time.

Use

If there are no active HTTP transactions (that is, no active requests or responses) to a server for the specified time (in seconds), the system closes the TCP connection to the server, reclaiming resources. This can help avoid problems that some HTTP servers have when connections are kept open indefinitely.

Setting this to a value of 0 disables the timeout, which directs the system not to close server TCP connections due to this timeout.

Default Setting

0 (disabled)

Command Mode

config-fproxy-http

Syntax

`[no] keepalive-timeout <timeout>`

Closes the TCP connection if there is no activity in this amount of time

`no keepalive-timeout`

Removes the TCP connection timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional)

request-idle-timeout

Configure the number of seconds to wait to receive any part of a request before closing the connection.

Use

The system closes the connection if it takes longer than the specified time (in seconds) to receive any part of the request from the client.

Consider the size of a typical client request as well as the user environment to set this value. For example, an application where users upload photos or HD videos using mobile devices would need a longer timeout than simple web pages due to mobile bandwidth and device processing limitations.

Default Setting

0 (disabled)

Command Mode

config-fproxy-http

Syntax

`[no] request-idle-timeout <timeout>`

Number of seconds to wait for new request data before closing the connection

no request-idle-timeout

Remove the request idle timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional)

response-idle-timeout

Configure the number of seconds to wait for new response data before closing the connection.

Use

The system closes the connection if, after transmitting the request to the server, it takes longer than the specified time (in seconds) either to receive any part of the response from the HTTP server or to transmit any part of the response to the client.

Consider the size of a typical response for your application as well as the user environment to set this value. For example, an application where users download HD videos using mobile devices would need a longer timeout than simple web pages due to mobile bandwidth and device processing limitations.

Default Setting

0 (disabled)

Command Mode

config-fproxy-http

Syntax

[no] response-idle-timeout <timeout>

Number of seconds to wait for new response data before closing the connection

no response-idle-timeout

Removes the response idle timeout

Parameter	Type	Description

timeout	Real	Number of seconds (can be fractional)

response-timeout

Configure the number of seconds to wait for the server to respond to a request.

Use

The system sends an HTTP 504 error response to the client and closes the connection if the HTTP server takes longer than the specified time (in seconds) to respond to a request.

Consider the amount of time the web server takes to respond to any request. The response-timeout must always be configured to be higher than the amount of time it takes for any of the web servers to respond to a request.

Default Setting

0 (disabled)

Command Mode

config-fproxy-http

Syntax

`[no] response-timeout <timeout>`

Number of seconds to wait for the server to respond to a request

`no response-timeout`

Removes the response timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional)

service tcp

Configure TCP as the service type for the forward proxy. The system does load balancing of TCP connections without processing the TCP payload.

Use

Sets the forward proxy to be a TCP proxy, operating at layer 4. Use this service type for non-HTTP protocols, such as SMTP email, or if you do not want the LineRate to interact with HTTP requests and responses. You can also use this service type for HTTPS if you do not want the LineRate to terminate the SSL connection.

Default Setting

Service default is http.

Command Mode

config-fproxy

Syntax

```
[no] service tcp
```

Sets the service type to TCP

data-idle-timeout

Use

The system closes the TCP connection if there is no activity in this amount of time.

Default Setting

0 (disabled)

Command Mode

config-fproxy-tcp

Syntax

```
[no] data-idle-timeout <timeout>
```

Closes the TCP connection if there is no activity in this amount of time

```
no data-idle-timeout
```

Removes the data idle timeout

Parameter	Type	Description

timeout	Real	Number of seconds (can be fractional)

Health Monitor Mode Commands

Use the following commands to configure health monitors.

health-monitor

Configures a health monitor

Use

Use to create a health monitor to regularly check that servers are up and able to accept connections or respond to requests. You can attach a health monitor to a real server and to a real server base.

For HTTP health monitors, you can configure the type of requests and responses.

Default Setting

None

Command Mode

configure

Syntax

`[no] health-monitor <name>`

Configures a health monitor

Parameter	Type	Description
name	Word	Name of the health monitor

Related Commands

CLI Reference - Real Server Mode Commands

REST API Reference - monitor

admin-status

Bring an object, such as a health monitor, real server, or virtual IP, online or offline. After you create an object, you must bring it online.

Use

You typically set the offline status only when you want to disable the object or block connections to the web server during maintenance or system reconfiguration.

Default Setting

offline

Command Mode

config

Syntax

admin-status offline

Bring the object offline

admin-status online

Bring the object online

no admin-status

Remove the directly configured admin status

attach

Attach an object, such as an SSL profile, to this health monitor

Use

Use to attach an SSL profile to the health monitor to override the real server's settings.

When you attach the health monitor to a real server, the health monitor uses the IP address and port from the real server to access the web server. If the real server uses SSL, the health monitor automatically uses the same SSL profile.

If you want the health monitor to use a different SSL configuration to connect to the web server, you can create and attach a different SSL profile to the health monitor. This SSL profile overrides the SSL profile attached to the real server and tells the health monitor to use this SSL profile.

If the real server uses SSL, but you want to health monitor to connect without using SSL, use the none form of the command. The none form of the command overrides the real server's SSL profile and tells the health monitor to connect using HTTP, not HTTPS.

Default Setting

None

Command Mode

config-hmonitor

Syntax

[no] attach ssl profile <name>

Attach or replace the SSL profile for this health monitor. If no profile is set on the health monitor, the SSL profile is obtained from each real server being monitored.

[no] attach ssl profile none

Set this health monitor to do non-SSL HTTP. Overrides the SSL profile set on each real server being monitored.

Parameter	Type	Description
name	Word	Name of the SSL profile to attach.

Related Commands

Real Server Mode Commands

SSL Mode Commands

description

Description for the health monitor.

Use

Use to create a description of the health monitor use and any other information.

Default Setting

None

Command Mode

config-hmonitor

Syntax

`[no] description <desc>`

Description for health monitor

`no description`

Removes the description

Parameter	Type	Description

desc	String	Description string. Enter a maximum of 255 characters.

interval

Time to wait between health monitor probes.

Use

Use to set how often the health monitor probes the server. In setting the interval, consider how the interval affects both the server and LineRate. You do not want the health monitor probes to overwhelm either system by being too frequent or to not detect a problem with a server by being too infrequent.

Default Setting

10

Command Mode

config-hmonitor

Syntax

```
[no] interval <intvl>
```

Time to wait between monitor probes

```
no interval
```

Removes the configured time to wait between monitor probes

Parameter	Type	Description
intvl	Real	Number of seconds (can be fractional).

invert

Reverse the success criteria, that is, mark the real server as down if the configured response criteria are met.

Use

Use to reverse the success criteria for the health monitor. When using invert, the health monitor marks the real server as down if the configured response is received. If a response other than the configured response is received, then the real server is marked as up.

One use case for invert is to support steering of traffic to a service that is in an active-standby configuration. You configure the health monitors such that when the active real server is functioning, it is marked a up, and the backup real server is marked as down. If the active real server is not functioning and the backup is functioning, the backup is marked as up. If both are not functioning, then both of them are marked as down.

Below is an example configuration:

- Real server rs1 is the primary, and real server rs2 is the secondary.
- Health monitor hm1 is attached to rs1 and monitors rs1's IP address.
- Health monitor hm2 is attached to rs2 and monitors rs2's IP address.
- Health monitor hm3 is attached to rs2 and monitors rs1's IP address. hm3 has the same settings as hm1 (same values for interval, server-up, etc.), but it also has the invert option set.
- If rs1 goes down, the system reports that rs1 is down and rs2 is up, making rs2 the active, and rs1 the standby.

Default Setting

None

Command Mode

config-hmonitor

Syntax

```
[no] invert
```

Marks the real server as down when the expected response is received

logging

Configure logging level.

Use

Use for debugging to log information about health monitor probes. Debugging on multiple health monitors will affect system performance. Normally, you do not need to use logging.

Default Setting

Disabled

Command Mode

config-hmonitor

Syntax

`[no] logging detailed`

Enable detailed logging (sets to most verbose) for debugging when instructed to set by technical support personnel

`[no] logging detailed <level>`

Enable detailed logging (level indicates verbosity)

Parameter	Type	Description
level	Integer	Level for detailed logging.

override-ip

Configure an IP address to use for monitoring.

Use

Use to configure an IP address to use for monitoring instead of using the IP address configured on the real server.

When you attach the health monitor to a real server, the health monitor uses the IP address and port from the real server to access the web server. If you want the health monitor to use a different IP address, you can override the IP address.

Default Setting

None

Command Mode

config-hmonitor

Syntax

`[no] override-ip <IP_address>`

Configure an IP address to use for monitoring instead of using the IP address configured on the real server

`no override-ip`

Remove the configured IP address to use for monitoring instead of using the IP address configured on the real server

Parameter	Type	Description
IP_address	IPAddr	IP address to use instead of the IP address on the real server.

override-port

Configure a port to use for monitoring.

Use

Use to configure a port to use for monitoring instead of using the port configured on the real server.

When you attach the health monitor to a real server, the health monitor uses the IP address and port from the real server to access the web server. If you want the health monitor to use a different port, you can override the port.

Default Setting

None

Command Mode

config-hmonitor

Syntax

`[no] override-port <port>`

Configure a port to use for monitoring instead of using the port configured on the real server

`no override-port`

Remove the port to use for monitoring instead of using the port configured on the real server

Parameter	Type	Description
port	Integer	Port to use instead of the port on real server.

server-down

Criteria to move a server from up to down.

Use

Use to determine when a server that is up should be marked as down, because the specified number of health probes has failed. The ratio you enter should take into account network and other transient conditions that may cause health probes to fail, but which you want to disregard.

For example, if you enter 8/10, this sets the threshold for marking the server as down to the failure of 8 out the last 10 health probes. If the health probe fails 8 out of the last 10 times, the system takes the server offline.

When the system takes a server offline, the system no longer sends traffic to that server.

Default Setting

1/10 (any single probe failure will mark the server as down)

Command Mode

config-hmonitor

Syntax

`[no] server-down <frac>`

Criteria to move server from up to down

no server-down

Remove the configured criteria to move server from up to down

Parameter	Type	Description
frac	String	Enter the ratio of failed to total probes (m/n where m of the last n health probes must fail).

Related Commands

server-up

server-up

Criteria to move a server from down to up.

Use

Use to determine when a server that is down should be marked as up because the specified number of health probes has succeeded. The ratio you enter should take into account network conditions and other transient conditions that may cause health probes to fail, but which you wish to disregard.

For example, if you enter 9/10, this sets the threshold for marking the server up to the success of 9 out the last 10 health probes. If the health probe succeeds 9 out of the last 10 times, the system puts the server back online.

When the system takes a server offline, the system no longer sends traffic that server. When the server comes back online, the system starts to send traffic to the server.

Default Setting

2/2

Command Mode

config-hmonitor

Syntax

[no] server-up <frac>

Criteria to move server from down to up

`no server-up`

Remove the configured criteria to move server from down to up

Parameter	Type	Description
frac	String	Enter the ratio of successful to total probes (m/n where m of the last n health probes must succeed).

Related Commands

server-down

timeout

Time (in seconds) to wait before considering a probe as failed.

Use

Use to set how long the system waits to determine if a specific health probe has failed. The health monitor will determine an individual health probe to fail if it does not respond within this time.

One use of this setting is to test the server's response time.

Default Setting

3

Command Mode

config-hmonitor

Syntax

`[no] timeout <intvl>`

Time to wait before considering a probe as failed

`no timeout`

Remove the configured time to wait before considering a probe as failed

Parameter	Type	Description
intvl	Real	Number of seconds (can be fractional)

type tcp

Set type of health monitor to TCP.

Use

Use this type for servers that are not using HTTP, for example, an email server. The health monitor establishes a TCP connection. If the connection is successful, the health probe is successful.

Default Setting

None

Command Mode

config-hmonitor

Syntax

`[no] type tcp`

Set as TCP health monitor

`no type`

Remove the health monitor type

type http

Set type of health monitor to HTTP.

Use

Use this for web servers that are using HTTP. This type must match the service setting on any real sever to which the health monitor is attached. Use for HTTPS as well.

The health monitor sends HTTP requests. You can configure the request body, header, method, version, and status. You can also specify the response header, body, status, and version.

Default Setting

Head request

Command Mode

config-hmonitor

Syntax

`[no] type http`

Set as HTTP health monitor and enter HTTP options mode

`no type`

Remove the health monitor type

request-body

Configure data to be included as body of request.

Use

Use to specify the data for the request body.

Default Setting

None

Command Mode

http

Syntax

`[no] request-body <body>`

Data to be included as body of request

`no request-body`

Removes the data to be included as body of request

Parameter	Type	Description

body	EscString	Body content as a string that can contain simple escaped characters like \", \r, \n, \t.

request-header

Configure additional HTTP header to include in the HTTP request.

Use

Use to specify one or more headers to include in the HTTP request. Specify in the format: header type: header_name=value

You might use this to tell the web server that the request is coming from a health monitor or to ensure the server will respond to the request.

Default Setting

None

Command Mode

http

Syntax

```
[no] request-header <header_value>
```

Additional HTTP header to include in the HTTP request

Parameter	Type	Description
header_value	String	Header and value as a string (for example, "Cookie: foo=bar; id=me")

request-method

Configure HTTP request method or verb indicating action to perform.

Use

Use the set the HTTP request method you want to use. The available methods are: GET, HEAD, POST, TRACE, OPTIONS, DELETE, and CONNECT.

Default Setting

HEAD

Command Mode

http

Syntax

`[no] request-method <method>`

HTTP request method or verb indicating action to perform

`no request-method`

Remove the configured HTTP request method or verb indicating action to perform

Parameter	Type	Description
method	Word	Request method (for example, GET, POST, HEAD).

request-target

Configure URL relative to the root of real server, (for example "/health.html").

Use

Use to specify an absolute or relative URL that points to where the health monitor should look for a file.

Default Setting

/

Command Mode

http

Syntax

`[no] request-target <target>`

URL relative to the root of real-server, (for example, "/health.html")

`no request-target`

Remove the configured URL relative to the root of real-server

Parameter	Type	Description
target	String	Request target string.

request-version

Configure HTTP protocol version included in HTTP header.

Use

Use the specify the HTTP protocol version that the health monitor uses for requests. For example, "HTTP/1.1"

Default Setting

None

Command Mode

http

Syntax

`[no] request-version <version>`

HTTP protocol version included in HTTP header

`no request-version`

Remove the configured HTTP protocol version included in HTTP header

Parameter	Type	Description
version	String	Version as string (for example, "HTTP/1.1")

response-body

Configure the expected response body (exact match).

Use

Use if you want to specify the exact content of the entire response body. For the response to be successful, the body must exactly match what you specify here.

Default Setting

None

Command Mode

http

Syntax

`[no] response-body <body>`

Expected response body (exact match)

`no response-body`

Removes the configured response body

Parameter	Type	Description
body	EscString	Expected response body as a string.

response-header

Configure a specific header/value pair for the response.

Use

Use if you want to specify the exact content of a response header and value pair. For the response to be successful, the header must exactly match what you specify here.

Default Setting

None

Command Mode

http

Syntax

`[no] response-header`

Require response to have a particular header/value pair

Parameter	Type	Description

header	String	Header as a string (for example, "Content-type: text/html")

response-no-header

Configure the response to not have a specific header.

Use

Use if you want to specify content that the response header must not contain. For the response to be successful, the header must not contain what you specify here.

Default Setting

None

Command Mode

http

Syntax

```
[no] response-no-header
```

Require response to not have a particular header

Parameter	Type	Description
header	String	Header name as a string (for example, "Content-length").

response-status

Expected HTTP response status code indicating the outcome of the request.

Use

Use if you want to specify the exact content of a HTTP response status code. For the response to be successful, the status code must exactly match what you specify here.

Default Setting

200 OK

Command Mode

http

Syntax

`[no] response-status <status>`

Expected HTTP response status code indicating the outcome of the request

`no response-status`

Removes the configured expected HTTP response status code

Parameter	Type	Description
status	Integer	HTTP status code (for example., "200" for "200 OK")

response-version

Configure the expected HTTP protocol response version included in HTTP header.

Use

Use if you want to specify the exact HTTP response version. For the response to be successful, the version must exactly match what you specify here. For example, "HTTP/1.1"

Default Setting

None

Command Mode

http

Syntax

`[no] response-version <version>`

Expected HTTP protocol response version included in the HTTP header

`no response-version`

Removes the configured expected HTTP protocol response version included in the HTTP header

Parameter	Type	Description
version	String	HTTP Version as string (for example, "HTTP/1.0").

Host Name Mode Commands

hostname

Use

Use to set the host name for the system. The host name changes the system prompt and the system logs.

The naming rules for a host name are:

- Must not be longer than 63 characters
- Must start with letter
- Must end with letter or digit
- Can use only letters, digits, hyphens, and periods

Default Setting

LROS

Command Mode

configure

Syntax

`hostname <name>`

Set the host name for the device

Parameter	Type	Description
name	String	Name of the system

Related

REST API Reference - hostname

Interface Mode Commands

Use the following commands to configure interfaces.

interface

Configure settings for a network interface for management or data use.

Use

Typically, you want to configure one interface for management use and one or more other interfaces for data use.

To see the names of your interfaces, use the following:

- CLI command: **`show interfaces`**
    - The following information displays:

        - A list of all interfaces on the system displays. It is possible for the system to have an interface that LineRate cannot detect.
        - The first line is the interface name and its status. The interface name is based on the driver for the interface type. The remaining lines list the information available about the interface (typically, manufacturer, model, MAC address, speed in kilobits, and more).
        - Below are the names used for some common interfaces:
            - em—Intel 1Gb interface
            - igb—Intel 1Gb interface
            - bce—Broadcom 1Gb interface
            - ix—Intel 10 Gb interface
            - oce—Emulex 10 Gb interface
            - lo—Loopback interface (internal interface)
            - po—Port channel interface
- REST API node: /status/system/interface?op=list
    - The names and current settings for every interface display. You can find all of the information that is included in the **`show interfaces`** command in the hierarchy below /status/system/interface.

You can create up 4094 subinterfaces on an interface. Subinterfaces are disabled by default when you create them. Use the encapsulation command to set up trunked ports for VLANs.

You can create port channel groups to aggregate two to eight interfaces for increased throughput. An interface can be a member of only one port channel.

Port channels generally act like any other interface, and you can configure most settings on them. For example, you configure a VLAN on the port channel, not on the member interfaces. The table below describes the effect of settings on the physical interface before adding it to a port channel, and the settings you can configure on a port channel.

Interface setting	Effect when configured on physical interface before adding to port channel	Effect when configured on port channel interface	Notes
carp	Not valid. You must remove CARP before adding the interface to a port channel.	Valid.	
channel-group	Valid.	Not valid. You cannot nest channel groups (port channels).	
description	Valid.	Valid.	
encapsulation	Not valid. You must remove encapsulation before adding the interface to a port channel.	Valid.	

Interface setting	Effect when configured on physical interface before adding to port channel	Effect when configured on port channel interface	Notes
flowcontrol	Valid. Remains in effect after adding to a port channel.	Not valid.	
ip	Ignored. IP address remains on interface, but has no effect.	Valid.	When you remove the interface from the port channel, the IP address will be active.
ipv6	Ignored. IP address remains on interface, but has no effect.	Valid.	When you remove the interface from the port channel, the IP address will be active.
mtu	Ignored. The MTU remains on interface, but has no effect.	Valid.	When you remove the interface from the port channel, the MTU will be active.

Default Settings

interface—enabled

port channel—enabled

subinterface—disabled

Command Mode

config

Syntax

`[no] interface <interface_name>`

Configure an interface

`[no] interface <interface_name>.<n>`

Create or configure a subinterface. Use a subinterface to set up trunked ports for VLANs.

`[no] interface port-channel <chan_num>`

Configure a port channel

Parameter	Type	Description

chan_num	Integer	Port channel number, which you can specify as just the number or as po<n>. You must use po<n> when you configure the port channel.
interface_name	String	Name of an interface. Interface name may be one of the following: \n physical interface (for example, ix0), a port channel (for example, po0), or subinterface (for example ix0.1, po0.1).

Related Commands

carp

encapsulation

REST API Reference - interface

carp

Set up CARP redundancy protocol.

Use

Use CARP to provide high availability and permit the system to fail over from one host to another. In LineRate, you configure CARP on each interface that you want in a group using the same Virtual Host ID (VHID). The Ethernet segment must support Ethernet and IP multicast. You can configure multiple CARP groups on an interface.

CARP Group Setup

Each CARP group must have a unique VHID on a particular VLAN and must not be the same as a VRRP VHID on the same VLAN. The system uses 241 as the IP protocol number.

Each CARP group can be master on only one system at a time. If you only want one system to be master and accept traffic at a time (active/passive deployment), you only need one CARP group, which is a single VHID. However, if you want to create an active/active deployment, you need the same number of CARP groups (VHIDs) as the number of systems that you would like to be active. For example, if you want to create a cluster of 5 instances where 3 systems are active and 2 are passive, you need 3 CARP groups.

The interface with the lowest priority value in the group becomes the master. Every interface in a group should have a unique priority within the group. The master does not advertise until the IP addresses are configured for the group.

Best Practice: We recommend leaving gaps in the carp priority numbering, so you can insert new systems into the priority ordering. In particular, do not configure any system to have

priority 1 so that at a later time, any system's priority can be set to 1 to manually have that system take over the master role.

The master replies to ARP requests for the group's IP addresses using the group's unique MAC address. The group MAC address uses the same formula as the VRRP MAC address (00-00-5E-00-01-x) with the last octet set to the group's VHID.

If using a password, all members of the group must have the same password, and only interfaces with the same password and group ID can be part of the group.

CARP Failover

The system supports subsecond failover, because advertisements are subsecond. The backup interface does not fail over until it misses seeing three (default) advertisements from the master. This takes into account dropped advertisement packets on congested networks and avoids unnecessary failovers.

When a backup believes that the master has failed, the backup sends out a gratuitous ARP reply to identify that the backup is now master for the IP address and group MAC address. The backup then responds to ARP requests for the IP address.

Use shutdown to test failover by shutting down the master. You can also use shutdown to keep the CARP configuration, when you want to shut down CARP on the interface to force a failover while reconfiguring CARP.

Direct-attach vs. Gateway Mode

If you're using virtual IPs in direct-attach mode (configuring the virtual IP in the CARP group), put the virtual IP in the CARP group. You do not configure it on the interface, but only in the CARP group. From a network topology point of view, any IP configured in CARP is like it is configured on the interface, meaning that the system will ARP for that IP address on that interface.

If you are using gateway mode (configuring the gateway address in the CARP group), put the gateway IP in the CARP group, not the virtual IP.

Port Channels

For a port channel:

- Before adding an interface to a port channel, you must remove it from the CARP group.
- You can configure CARP on a port channel.

For CARP examples, see CARP Example - Active/Standby and CARP Example - Active/Active.

Note: If you installed using VMWare, you must configure your vSwitches and hosts to permit CARP. See Configuring vSwitches and Hosts to Permit CARP and Failover.

Default Setting

None

Command Mode

config

Syntax

`carp <vhid>`

Set the CARP group ID on this interface

`no carp <vhid>`

Remove all configuration for CARP group ID on this interface

Parameter	Type	Description
vhid	Integer	Virtual Host ID (VHID) used to identify the CARP group. The value must be unique on the Ethernet segment and must not be the same as a VRRP VHID on the same Ethernet segment. Valid values are 1 - 254.

advert-base

Use

Use to set the base time interval for CARP advertisements in milliseconds. CARP advertisements are sent based on this interval plus an offset determined by the priority.

Advertising interval in milliseconds = advert-base + (priority * 4).

The default advert-base is 100 ms. We do not recommend changing the base time interval.

Default Setting

100 ms

Command Mode

config

Syntax

`[no] advert-base <base_time>`

Set the base time interval for CARP advertisements.

`no advert-base`

Remove the base time interval for CARP advertisements.

Parameter	Type	Description
base_time	Integer	Base time interval for CARP advertisements in milliseconds

attach

Use

Use to attach a failover group to the CARP interface. The failover group works with the CARP settings for high availability of the interfaces. For more information, see one of the following:

- CLI Reference - Failover Mode Commands.
- REST API Reference - failover

Default Setting

None

Command Mode

config

Syntax

`[no] attach <failover_grp>`

Attach the specified failover group to this interface.

`no attach`

Remove the attached failover group from this interface.

Parameter	Type	Description
group_name	Word	Name of the failover group

authentication

Use

Use to set the string to generate the authentication key. All systems participating in the CARP group must have the same password.

Default Setting

None

Command Mode

config

Syntax

`[no] authentication <pass>`

Set authentication password for the CARP group.

`no authentication`

Remove the authentication password for the CARP group

Parameter	Type	Description
pass	String	CARP group password.

description

Create a description for the interface.

Use

Use to create a description of the interface type, use, and any other information to identify the interface. Enter a maximum of 255 characters.

Default Setting

None

Command Mode

config

Syntax

`[no] description <desc>`

Description for this CARP group

`no description`

Remove the description for this CARP group

Parameter	Type	Description
desc	String	Description of this CARP group.

down-threshold

Use

Use to set the threshold for detecting that the CARP master is down. Number of expected CARP advertisement packets that must be lost before the master is declared down, and the next lowest priority interface becomes master. Default is 3.

Default Setting

3

Command Mode

config

Syntax

`[no] down-threshhold <thresh>`

Set threshold for detecting that the CARP master is down.

`no down-threshhold`

Remove the threshold for detecting that the CARP master is down.

Parameter	Type	Description
thresh	Integer	Threshold value.

ip

Use

Use to configure the IPv4 address for the CARP group. This is the address for which this VHID will manage failover.

Default Setting

None

Command Mode

config

Syntax

`[no] ip <addr>`

IPv4 address for which this VHID will manage failover

Parameter	Type	Description
addr	IPAddr	IPv4 address for the CARP group. See IP Address Notation.

ipv6

Use

Use to configure the IPv6 address for the CARP group. This is the address for which this VHID will manage failover.

For IPv6, the physical interface must have an IPv6 address and the CARP IP address must be within the subnet of the physical interface's IPv6 address.

Default Setting

None

Command Mode

config

Syntax

```
[no] ipv6 <addr>
```

IPv4 address for which this VHID will manage failover

Parameter	Type	Description
addr	IPAddr	IPv6 address for the CARP group. See IP Address Notation.

priority

Use

Use to set the CARP priority, also known as advskew, for this interface. The lowest priority interface in a group is the master. Valid user-configurable values are 1 - 23.

Default Setting

1

Command Mode

config

Syntax

`[no] priority <priority>`

Set the CARP priority, also known as advskew.

`no priority`

Remove the CARP priority, also known as advskew.

Parameter	Type	Description
priority	Integer	CARP priority for this interface.

shutdown

Use

Stops this system from participating in the CARP group, but does not remove the CARP group configuration.

Default Setting

None

Command Mode

config

Syntax

`[no] shutdown`

Stops this system from participating in the CARP group.

channel-group

Use

Use to add this interface to a port channel group.

You can create port channel groups to aggregate two to eight interfaces for increased throughput. An interface can be a member of only one port channel.

Port channels generally act like any other interface, and you can configure most settings on them. For example, you configure a VLAN on the port channel, not on the member interfaces. The table below describes the effect of settings on the physical interface before adding it to a port channel, and the settings you can configure on a port channel.

Interface setting	Effect when configured on physical interface before adding to port channel	Effect when configured on port channel interface	Notes
carp	Not valid. You must remove CARP before adding the interface to a port channel.	Valid.	
channel-group	Valid.	Not valid. You cannot nest channel groups (port channels).	
description	Valid.	Valid.	
encapsulation	Not valid. You must remove encapsulation before adding the interface to a port channel.	Valid.	
flowcontrol	Valid. Remains in effect after adding to a port channel.	Not valid.	
ip	Ignored. IP address remains on interface, but has no effect.	Valid.	When you remove the interface from the port channel, the IP address will be active.
ipv6	Ignored. IP address remains on interface, but has no effect.	Valid.	When you remove the interface from the port channel, the IP address will be active.
mtu	Ignored. The MTU remains on interface, but has no effect.	Valid.	When you remove the interface from the port channel, the MTU will be active.

The mode setting on each port channel member interface must be set the same.

Port channels in LineRate have two possible modes:

- Active—Sends out LACP probing packets to the switch. The switch can be set to active or passive LACP. If one interface fails, the switch will use the remaining interfaces configured for

the port channel. Whenever the port channel configuration changes, the port channel and the switch negotiate the settings. The port channel is not active during the negotiation period.

- On (static)—Switch must be configured the same way. If one interface in the port channel fails, the whole port channel fails.

Default Setting

None

Command Mode

config-int

Syntax

`[no] channel-group <chan_num> mode active`

Select LACP in active mode

`[no] channel-group <chan_num> mode on`

Select static link aggregation

`no channel-group`

Remove the configured interface from the channel group

Parameter	Type	Description
chan_num	Integer	Port channel number, which you can specify as just the number (<chan_num>) or as po<chan_num>. You must use po<chan_num> when you configure the port channel.

description

Create a description for the interface.

Use

Use to create a description of the interface type, use, and any other information to identify the interface. Enter a maximum of 255 characters.

Default Setting

No description

Command Mode

config

Syntax

`description <desc>`

Description for the interface

`no description`

Remove the description

Parameter	Type	Description
desc	String	Description string. Enter a maximum of 255 characters.

encapsulation

Set the encapsulation type for a subinterface.

Use

You can only set the encapsulation type for subinterfaces. Use subinterfaces and encapsulation for VLANs to create a trunked port (also called a tagged port). The system encapsulates the packets using the specified protocol and VLAN tag.

To set the encapsulation, which uses the IEEE 802.1Q standard, you must know the switch tag for the VLAN you are configuring.

For a port channel:

- Before adding an interface to a port channel, you must remove encapsulation.
- You can configure encapsulation on a port channel.

Default Setting

No encapsulation

Command Mode

config

Syntax

`[no] encapsulation dot1q <tag>`

IEEE 802.1Q virtual LAN encapsulation

`no encapsulation`

Remove the configured encapsulation

Parameter	Type	Description
tag	Integer	VLAN tag used on the switch. Valid range: 1-4094.

Related Commands

interface

flowcontrol receive

Ethernet flow control receive operation.

Use

Generally, we recommend leaving flow control off for both receive and send, unless you have a specific reason to turn it on.

Turning on flowcontrol receive tells the port to honor the pause frame from a sending interface. Leaving flow control off tells the interface to ignore the pause frame from the sending interface.

 Note: The system supports flow control only on ix interfaces. Flow control is off by default on ix interface types. For other interfaces, the default varies based on the interface type.

 Caution: Turning on flowcontrol receive could stop all traffic on the interface.

For a port channel:

- Any flow control setting configured on an interface remains in effect when you add the interface to a port channel.
- You cannot configure flow control on a port channel.

Default Setting

Depends on interface type. See Note above.

Command Mode

config

Syntax

`flowcontrol receive off`

Always ignore pause frames received

`flowcontrol receive on`

Always respond to pause frames received

`no flowcontrol receive`

Remove the configured flowcontrol receive

flowcontrol send

Use

Generally, we recommend leaving flow control off for both receive and send, unless you have a specific reason to turn it on.

Turning flow control on tells the interface to send the pause frame when the interface is overloaded. Leaving flow control off tells the interface to not to send the pause frame when the interface is overloaded.

Note: The system supports flow control only on ix interfaces. Flow control is off by default on ix interface types. For other interfaces, the default varies based on the interface type.

For a port channel:

- Any flow control setting configured on an interface remains in effect when you add the interface to a port channel.
- You cannot configure flow control on a port channel.

Default Setting

Depends on interface type. See Note above.

Command Mode

config

Syntax

`flowcontrol send off`

Never send pause frames for flow control

`flowcontrol send on`

Enable sending of pause frames for flow control

`no flowcontrol send`

Remove the configured flowcontrol send

ip address

Set IPv4 address for the interface.

Use

You can configure more than one IP address on an interface. Each virtual IP must also be assigned to an interface or to a CARP group.

When you set the IP address, the system automatically also adds a local route based on the IP address.

An interface can have more than one IP address. Unlike other systems you may work with, LineRate adds each specified IP address to the interface. Use the no form of the command to remove an IP address.

For a port channel:

- Any IP address configured on an interface can remain before adding the interface to a port channel, but the IP address is ignored. When you remove the interface from the port channel, the IP address will be active.
- You can configure IP addresses on a port channel.

Default Setting

None

Command Mode

config

Syntax

`[no] ip address <addr>`

Set the IPv4 address

`[no] ip address <addr> <mask>`

Set the IPv4 address and netmask

Parameter	Type	Description
addr	IPv4Addr	IPv4 address assigned to the interface. See IP Address Notation.
mask	IPv4Addr	IPv4 address and netmask assigned to the interface. See IP Address Notation.

ipv6 address

Set IPv6 address for the interface.

Use

You can configure more than one IP address on an interface. Each virtual IP must also be assigned to an interface or to a CARP group.

When you set the IP address, the system automatically also adds a local route based on the IP address.

An interface can have more than one IP address. Unlike other systems you may work with, LineRate adds each specified IP address to the interface. Use the no form of the command to remove an IP address.

For a port channel:

- Any IP address configured on an interface can remain before adding the interface to a port channel, but the IP address is ignored. When you remove the interface from the port channel, the IP address will be active.
- You can configure IP addresses on a port channel.

Default Setting

None

Command Mode

config

Syntax

`[no] ipv6 address <addrMask>`

Specifies an IPv6 address.

`[no] ipv6 address <addrMask> link-local`

Specifies that the IPv6 address is link-local. Address must begin with fe80.

`[no] ipv6 enable`

Enables IPv6 processing on the interface with an automatically generated link-local address, even if no other IPv6 addresses are configured.

Parameter	Type	Description
addrMask	IPv6AddrMask	IPv6 address and prefix bits. See IP Address Notation

mtu

Interface maximum transmission unit (MTU).

Use

Sets the MTU for the Ethernet interface. When using VLANs, how the system takes into account the VLAN header varies based on the Ethernet hardware and driver.

Different Ethernet hardware and associated drivers implement different types of support for VLAN. Some provide the ability to create slightly longer packets (such as1508 bytes). If the hardware and driver support the ability to add the VLAN header overhead in addition to the MTU, the system does that and generates packets up to 1508. If the hardware and driver do not support adding the VLAN header, the system lowers the MTU then adds the VLAN header, resulting in packets that are the MTU size.

LineRate software supports jumbo frames up to 9000 bytes. However, some Ethernet devices have lower MTU limits.

For assistance in determining what your hardware/driver supports, contact F5 technical support.

For a port channel:

- Any MTU configured on an interface can remain before adding the interface to a port channel, but the MTU is ignored. When you remove the interface from the port channel, the MTU will be active.
- You can configure MTU on a port channel.

Default Setting

1500

Command Mode

config

Syntax

`mtu <mtu>`

Interface maximum transmission unit (MTU)

`no mtu`

Remove the configured MTU

Parameter	Type	Description
mtu	Integer	MTU size (bytes)

shutdown

Disable the interface.

Use

You may want to shut down interfaces that you are not using. Shutting down an interface prevents traffic from entering or leaving that interface and also shuts down all CARP groups attached to that interface.

Default Setting

None

Command Mode

config

Syntax

`shutdown`

Disable the interface

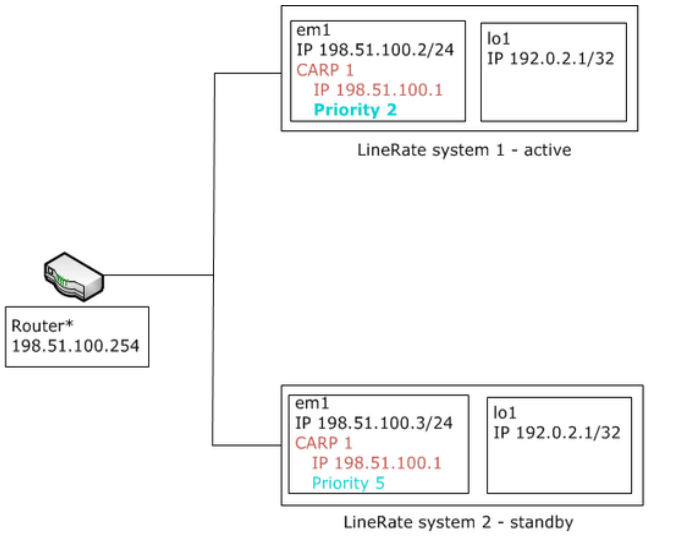
CARP Example - Active/Standby

CARP Overview - Active/Standby Configuration

Configuring CARP on interfaces permits failover for one network segment between two or more LineRate systems. This example shows an active/standby configuration where CARP is configured on the outside interfaces (client access).

In the active/standby configuration (shown below), the CARP group on one LineRate system is configured as the active group, based on its lower priority. The other LineRate system is configured identically, except the CARP group has a higher priority, making it the standby.

To focus on how CARP works, this example shows only the outside segment (network traffic coming into the LineRate system).



Example active/standby configuration using CARP

What You Need

For this configuration, as shown above, you need the following:

- One router
- Two LineRate systems with the following:
  - One loopback interface on each
  - One outside interface on each
- IP addresses for the following:
  - One for the router
  - One for the virtual IP (will also be the IP address for the loopback interface)
  - One for the outside interfaces (one unique IP each)
  - One for the CARP group

How the Configuration Works

The code example in the next section sets up the following configuration (including some items not shown in the diagram above):

- Each LineRate system is configured as follows:
  - Each outside interface (on em1) has a unique IP address (198.51.100.2/24 and 198.51.100.3/24).
  - Each outside interface is part of CARP group 1 with the shared CARP IP set to 198.51.100.1. The shared CARP IP is used as a next hop gateway address for the router to reach the virtual IP address.

    - The priority on the active system is set to 2.
    - The priority of the standby system is set to 5.
  - The virtual IP address is configured as 192.0.2.1 80.
  - A loopback interface is configured with the virtual IP as its address (192.0.2.1/32).
  - The virtual IP address is attached to a virtual server.
  - The real servers are atttached to the virtual server.
  - The default IP route on each LineRate system is set to 0.0.0.0/0 198.51.100.254.
- The router is configured via a static IP route to send requests for destination IP 192.0.2.1 (the VIP) to 198.51.100.1 (the CARP IP).

Requests to the virtual IP 192.0.2.1 are sent to through the router, which sends them to the active system based on its routing and the CARP priority. The active system then sends the requests to the virtual server and to the real server. If anything in the path to the active system's em1 interface fails, the standby system will become master for the CARP group and will accept traffic for 192.0.2.1 (the VIP) using 198.51.100.1 (the CARP IP).

The configuration on all LineRate systems meant for CARP failover must have the identical configuration, *except* for the following:

- Host name
- Unique management IP
- Unique IP on each CARP interface
- CARP priority

Annotated Configuration

Below is the configuration that is key to configuring active/passive CARP. We have excluded parts of the configuration to focus only on what is required for this CARP configuration to work. In the section below, we have included the complete show run output, which has the complete configuration for a simple load balancer using active/passive CARP for the outside interface for both systems.

Configuration for LineRate system 1 (active):

Commands	Description

`interface em1`
`  ip address 198.51.100.2/24`
`  carp 1`
`  priority 2`
`  ip 198.51.100.1`
	Sets the unique IP address for the interface em1.
	
	Adds CARP group 1 with a priority of 2 on the em1 interface, making this system the master, and sets the shared CARP IP address.

Commands	Description
`interface lo1` `  ip address 192.0.2.1/32`	Sets the IP address for the loopback interface lo1, which is the same as the virtual IP created below.
`ip route 0.0.0.0/0 198.51.100.254`	Sets the system's default IP route for all traffic to 198.51.100.254, the router's IP address.

`real-server base webserver_base`
`  admin-status online`
`real-server rs1`
`  ip address 10.1.2.101 80`
`  base webserver_base`
	Creates a real server base webserver_base with base setting "admin-status online".
	Creates a real server called rs1 based on webserver_base, assigns 10.1.2.101 as the real server's IP address, and brings the real server online through the inherited base setting.

`virtual-ip base web_base`
`  admin-status online`
`virtual-ip vip1 ip 192.0.2.1 80`
`  base web_base`
	Creates a virtual IP base web_base with base setting "admin-status online".
	Creates a virtual IP called vip1 based on web_base, assigns 192.0.2.1 as the virtual IP address, and brings the virtual IP base online through the inherited base setting.

`virtual-server vs1`
`  lb-algorithm round-robin`
`  service http`
`  attach virtual-ip vip1 default`
`  attach real-server rs1`
	Creates the virtual server vs1 and attaches the virtual IP vip1 and the real server rs1 to it.
	Also sets the load balancing algorithm and service to HTTP

Configuration for LineRate system 2 (standby; configuration differences from system 1 shown in blue):

Commands	Description

`interface em1`
`  ip address 198.51.100.3/24`
`  carp 1`
`  priority 5`
`  ip 198.51.100.1`
	Sets the unique IP address for the interface em1.
	
	Adds CARP group 1 with a priority of 5 on the em1 interface, making this system the standby, and sets the shared CARP IP address.

`interface lo1` `  ip address 192.0.2.1/32`	Sets the IP address for the loopback interface lo1, which is the same as the virtual IP created below.
`ip route 0.0.0.0/0 198.51.100.254`	Sets the system's default IP route for all traffic to 198.51.100.254, the router's IP address.

`real-server base webserver_base`
 	
`  admin-status online`
 	
`real-server rs1`
`  ip address 10.1.2.101 80`
`  base webserver_base`
	Creates a real server base webserver_base with base setting "admin-status online".
	
	Creates a real server called rs1 based on webserver_base, assigns 10.1.2.101 as the real server's IP address, and brings the real server online through the inherited base setting.

`virtual-ip base web_base`
`  admin-status online`
 	
`virtual-ip vip1 ip 192.0.2.1 80`
`  base web_base`
	Creates a virtual IP base web_base with base setting "admin-status online".
	
	Creates a virtual IP called vip1 based on web_base, assigns 192.0.2.1 as the virtual IP address, and brings the virtual IP base online through the inherited base setting.

`virtual-server vs1`
`  lb-algorithm round-robin`
`  service http`
`  attach virtual-ip vip1 default`
`  attach real-server rs1`
	Creates the virtual server vs1 and attaches the virtual IP vip1 and the real server rs1 to it.
	
	Also sets the load balancing algorithm and service to HTTP

Commands	Description

Example Show Run Output

The show run output below includes only the sections related to an active/standby CARP configuration.

Configuration for LineRate system 1 (active):

```
Building configuration...
!
hostname example-host-a
!
username admin secret encrypted "$2a$04$7TYufYOKVQ8i8bblVtZ1ierxZXzcH5mR/
    QeaZH8WnWRzVEkPt0MgS" uid 2000
!
interface em1
 ip address 198.51.100.2 255.255.255.0
 carp 1
  priority 2
  ip 198.51.100.1
!
interface lo1
 ip address 192.0.2.1 255.255.255.255
!
ip route 0.0.0.0/0 198.51.100.254
!
real-server base webserver_base
 admin-status online
!
real-server rs1
 ip address 10.1.2.101 80
 base webserver_base
!
virtual-ip base web_base
 admin-status online
!
virtual-ip vip1 ip 192.0.2.1 80 base web_base
!
virtual-server vs1
 lb-algorithm round-robin
 service http
 attach virtual-ip vip1 default
 attach real-server rs1
```

Configuration for LineRate system 2 (standby):

```
Building configuration...
```

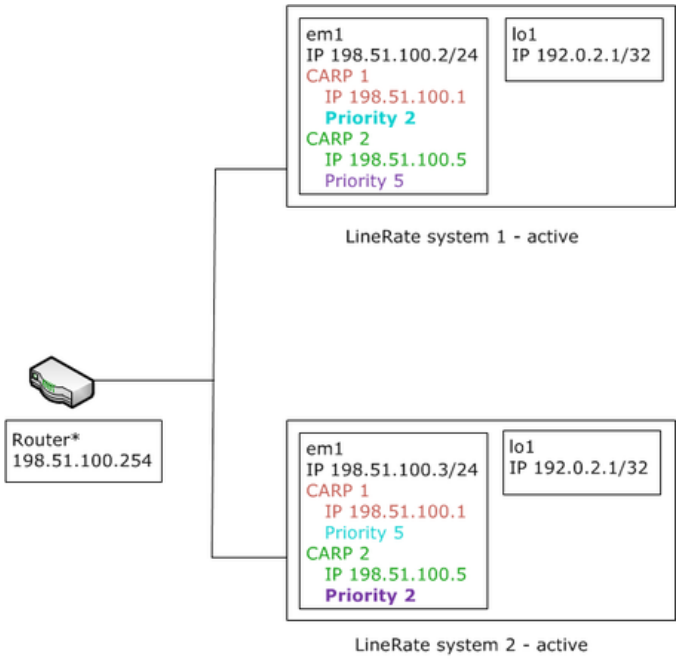
```
!
hostname example-host-b
!
username admin secret encrypted "$2a$04$7TYufYOKVQ8i8bblVtZ1ierxZXzcH5mR/QeaZH8WnWRzVEkPt0MgS" uid
    2000
!
interface em1
 ip address 198.51.100.3 255.255.255.0
 carp 1
  priority 5
  ip 198.51.100.1
!
interface lo1
 ip address 192.0.2.1 255.255.255.255
!
ip route 0.0.0.0/0 198.51.100.254
!
real-server base webserver_base
 admin-status online
!
real-server rs1
 ip address 10.1.2.101 80
 base webserver_base
!
virtual-ip base web_base
 admin-status online
!
virtual-ip vip1 ip 192.0.2.1 80 base web_base
!
virtual-server vs1
 lb-algorithm round-robin
 service http
 attach virtual-ip vip1 default
 attach real-server rs1
```

CARP Example - Active/Active

CARP Overview - Active/Active Configuration

Configuring CARP on interfaces permits failover for one network segment between two or more LineRate systems. This example shows an active/active configuration where CARP is configured on the outside interfaces (client access).

In the active/active configuration (shown below), you use two CARP groups and configure both on each system. The difference is that CARP 1 is active on system 1 and standby on system 2, based on its lower priority. Conversely, CARP 2 is standby on system 1 and active on system 2, based on its higher priority.



Example active/active configuration using CARP

To focus on how CARP works, this example shows only the outside segment (network traffic coming into the LineRate system. CARP works between interfaces in the group for one network segment (for example, inbound or outbound, but not both).

What You Need

For this configuration, as shown above, you need the following:

- One router that supports Equal Cost Multi-Path (ECMP) routing, using flow affinity or hashing.
- Two LineRate systems with the following:
  - One loopback interface on each
  - One outside interface on each
- IP addresses for the following:
  - One for the router
  - One for the virtual IP (will also be the IP address for the loopback interface)
  - One for the real server
  - Two for the outside interfaces (unique IP address assigned to each)
  - Two for the CARP groups (one IP address for each group)

If your router does not support ECMP or cannot use hashing or flow affinity to distribute all packets for the same TCP flow to the same next hop gateway, you can still use active/active CARP, but you must use more than one virtual IP. In that scenario, no two systems will be able to be active for a single virtual IP at once, because the router cannot distribute traffic for a single IP to multiple active systems.

How the Configuration Works

The code example in the next section sets up the following configuration (including some items not shown in the diagram above):

- Each LineRate system is configured as follows:
  - Each outside interface (on em1) has a unique IP address (198.51.100.2/24 and 198.51.100.3/24).
  - Each outside interface is part of CARP group 1 with the shared CARP IP set to 198.51.100.1. The shared CARP IP is used as a next hop gateway address for the router to reach the virtual IP address.
    - The priority on system 1 (the active system) is set to 2.
    - The priority on system 2 (the standby system) is set to 5.
  - Each outside interface is also part of CARP group 2 with the shared CARP IP set to 198.51.100.5. The shared CARP IP is used as a next hop gateway address for the router to reach the virtual IP address.
    - The priority on system 2 (the active system) is set to 2.
    - The priority on system 1 (the standby system) is set to 5.
  - The virtual IP address is configured as 192.0.2.1 80.
  - A loopback interface is configured with the virtual IP as its address (192.0.2.1/32).
  - The virtual IP address is attached to the virtual server.
  - The real servers are attached to the virtual server.
  - The default IP route on each LineRate system is set to 0.0.0.0/0 198.51.100.254.
- The router is configured with two static routes, both to the same virtual IP address as the destination, but with the two shared CARP IP addresses as the next hop addresses:
  - 192.0.2.1 (VIP) via gateway IP 198.51.100.1 (the CARP 1 IP)
  - 192.0.2.1 (VIP) via gateway IP 198.51.100.5 (the CARP 2 IP)

Requests to the virtual IP 192.0.2.1 are sent to the router, which uses Equal Cost Multi-Path (ECMP) hashing to send some TCP flows to CARP group 1 and the rest of the flows to CARP group 2. If both LineRate systems are operational and their interfaces are up, that means LineRate system 1 will receive all traffic for CARP group 1 and LineRate system 2 will receive all traffic for CARP group 2. If one of the systems fails or its interface goes down, the remaining system will receive all traffic for both CARP groups.

Although this example shows only two systems in the LineRate cluster for simplicity, this can be extended to a larger cluster with an arbitrary number of active systems and an additional number of

standby systems. You need one CARP group per **active** system in this configuration. All systems should have all CARP groups configured, but each with different priorities for that group.

The configuration on all LineRate systems meant for CARP failover must have the identical configuration, *except* for the following:

- Host name
- Unique management IP
- Unique IP on each CARP interface
- CARP priority

The real servers in this active/active configuration cannot use the client-source-address NAT profile (**`attach nat profile builtin client-source-address`**). Using this NAT profile would not permit the real server to know which LineRate system to send traffic back to, because the real server would only know the client's IP address not the IP address of the LineRate system.

Annotated Configuration

Below is the configuration that is key to configuring active/active CARP. We have excluded parts of the configuration to focus only on what is required for this CARP configuration to work. In the section below, we have included the complete show run output, which has the complete configuration for a simple load balancer using active/active CARP for the outside interface for both systems.

Configuration for LineRate system 1 (active):

Commands	Description

`interface em1`
`  ip address 198.51.100.2/24`
`  carp 1`
`  priority 2`
`  ip 198.51.100.1`
`  carp 2`
`  priority 5`
`  ip 198.51.100.5`
	Sets the unique IP address for the interface em1.
	
	Adds CARP group 1 with a priority of 2 on the em1 interface, making this system the master, and sets the shared CARP IP address.
	
	Adds CARP group 2 with a priority of 5 on the em1 interface, making this system the standby, and sets the shared CARP IP address.

Commands	Description
`interface lo1` `  ip address 192.0.2.1/32`	Sets the unique IP address for the loopback interface lo1, which is the same as the virtual IP created below.
`ip route 0.0.0.0/0 198.51.100.254`	Sets the system's default IP route for all traffic to 198.51.100.254, the router's IP address.

`real-server base webserver_base`
 	
`real-server rs1`
`  ip address 10.1.2.101 80`
`  base webserver_base`
`  admin-status online`
	Creates a real server base (base setings omitted for this example) webserver_base.
	
	Creates a real server called rs1 based on webserver_base, assigns 10.1.2.101 as the real server's IP address, and brings the real server online.

`virtual-ip base web_base`
`  admin-status online`
 	
`virtual-ip vip1 ip 192.0.2.1 80`
`  base web_base`
	Creates a virtual IP base (base setings omitted for this example) web_base.
	
	Creates a virtual IP called vip1 based on web_base, assigns 192.0.2.1 as the virtual IP address, and brings the virtual IP base online.

`virtual-server vs1`
`  lb-algorithm round-robin`
`  service http`
`  attach virtual-ip vip1 default`
`  attach real-server rs1`
	Creates the virtual server vs1 and attaches the virtual IP vip1 and the real server rs1 to it.
	
	Also sets the load balancing algorithm and service to HTTP

Configuration for LineRate system 2 (active; configuration differences from system 1 shown in blue):

Commands	Description

`interface em1`
`  ip address 198.51.100.3/24`
`  carp 1`
`  priority 5`
`  ip 198.51.100.1`
`  carp 2`
`  priority 2`
`  ip 198.51.100.5`
	Sets the unique IP address for the interface em1.
	
	Adds CARP group 1 with a priority of 5 on the em1 interface, making this system the standby, and sets the shared CARP IP address.
	Adds CARP group 2 with a priority of 2 on the em1 interface, making this system the master, and sets the shared CARP IP address.

`interface lo1` `  ip address 192.0.2.1/32`	Sets the unique IP address for the loopback interface lo1, which is the same as the virtual IP created below.
`ip route 0.0.0.0/0 198.51.100.254`	Sets the system's default IP route for all traffic to 198.51.100.254, the router's IP address.

`real-server base webserver_base`
 	
`real-server rs1`
`  ip address 10.1.2.101 80`
`  base webserver_base`
`  admin-status online`
	Creates a real server base (base setings omitted for this example) webserver_base.
	
	Creates a real server called rs1 based on webserver_base, assigns 10.1.2.101 as the real server's IP address, and brings the real server online.

`virtual-ip base web_base`
`  admin-status online`
 	
`virtual-ip vip1 ip 192.0.2.1 80`
`  base web_base`
	Creates a virtual IP base (base setings omitted for this example) web_base.
	
	Creates a virtual IP called vip1 based on web_base, assigns 192.0.2.1 as the virtual IP address, and brings the virtual IP base online.

`virtual-server vs1`
`  lb-algorithm round-robin`
`  service http`
`  attach virtual-ip vip1 default`
`  attach real-server rs1`
	Creates the virtual server vs1 and attaches the virtual IP vip1 and the real server rs1 to it.
	
	Also sets the load balancing algorithm and service to HTTP

Commands	Description

Example Show Run Output

The show run output below includes only the sections related to an active/active CARP configuration.

Configuration for LineRate system 1 (active):

```
Building configuration...
!
hostname example-host-a
!
username admin secret encrypted "$2a$04$7TYufYOKVQ8i8bblVtZ1ierxZXzcH5mR/
    QeaZH8WnWRzVEkPt0MgS" uid 2000
!
interface em1
 ip address 198.51.100.2 255.255.255.0
 carp 1
  priority 2
  ip 198.51.100.1
 carp 2
  priority 5
  ip 198.51.100.5
!
interface lo1
 ip address 192.0.2.1 255.255.255.255
!
ip route 0.0.0.0/0 198.51.100.254
!
real-server base webserver_web
!
real-server rs1
 ip address 10.1.2.101 80
 base webserver_base
 admin-status online
!
virtual-ip base web_base
 admin-status online
!
virtual-ip vip1 ip 192.0.2.1 80 base web_base
!
virtual-server vs1
 lb-algorithm round-robin
 service http
 attach virtual-ip vip1 default
 attach real-server rs1
```

Configuration for LineRate system 2 (active):

```
Building configuration...
!
hostname example-host-b
!
username admin secret encrypted "$2a$04$7TYufYOKVQ8i8bblVtZ1ierxZXzcH5mR/QeaZH8WnWRzVEkPt0MgS" uid
    2000
!
interface em1
 ip address 198.51.100.3 255.255.255.0
 carp 1
  priority 5
  ip 198.51.100.1
 carp 2
  priority 2
  ip 198.51.100.5
!
interface lo1
 ip address 192.0.2.1 255.255.255.255
!
ip route 0.0.0.0/0 198.51.100.254
!
real-server base webserver_web
!
real-server rs1
 ip address 10.1.2.101 80
 base webserver_base
 admin-status online
!
virtual-ip base web_base
 admin-status online
!
virtual-ip vip1 ip 192.0.2.1 80 base web_base
!
virtual-server vs1
 lb-algorithm round-robin
 service http
 attach virtual-ip vip1 default
 attach real-server rs1
```

IP Mode Commands

Use the following commands to configure global IP settings.

ip dns

Configure parameters for domain name resolution.

Use

Use to configure system-wide settings for Domain Name System (DNS), which allows the name-to-IP resolution of domain names.

 By default, the system comes with a ready-to-use DNS configuration. You can use the default servers or configure the system to use your own servers. For more information, see name-server below.

Default Setting

None

Command Mode

configure

Syntax

```
[no] ip dns
```

Configure parameters for domain name resolution.

Related Commands

REST API Reference - dns

admin-status

Bring an object, such as a health monitor, real server, or virtual IP, online or offline. After you create an object, you must bring it online.

Use

You typically set the offline status only when you want to disable the object or block connections to the web server during maintenance or system reconfiguration.

Default Setting

offline

Command Mode

config

Syntax

`admin-status offline`

Bring the object offline

`admin-status online`

Bring the object online

`no admin-status`

Remove the directly configured admin status

domain-list

Configure a list of domains to search to complete unqualified host names.

Use

Use to confgure up to six domains to search to complete unqualified host names. Enter the domains, in priority order, separated by spaces and the whole string surrounded by double quotes. Use up to a total of 256 characters. The system searches the list in the order entered.

Default Setting

None

Command Mode

config-ip-dns

Syntax

`domain-list <domain_list>`

Configure a list of domains to search to complete unqualified host names.

Parameter	Type	Description
domain_list	String	Space-separated list of domains to search, each to be tried in turn. The search list is limited to 6 domains with a total of 256 characters.

name-server

Configure a space-separated prioritized list of domain name servers.

Use

Use to configure up to three domain name servers. Enter the IP addresses, in priority order, separated by spaces.

By default, the system is configured with two domain name servers (8.8.8.8 and 8.8.4.4). These are the Google Public DNS servers. You can continue to use these servers or configure the system to use your own servers.

Default Setting

None

Command Mode

config-ip-dns

Syntax

`name-server <addr1> [<addr2>] [<addr3>]`

Parameter	Type	Description
addr1	IPAddr	IPv4 or IPv6 address of first name server.
addr2	IPAddr	IPv4 or IPv6 address of second, optional, name server.
addr3	IPAddr	IPv4 or IPv6 address of third, optional, name server.

retries

Configure number of times to retry a name server request.

Use

Use to configure the number of times the system tries to contact each domain name server, after the first contact. The system tries each name server in the list, then starts the retries of each server.

The default is 1 retries. The maximum permitted is 4 retries. To configure no retries, set to 0.

Default Setting

1

Command Mode

config-ip-dns

Syntax

`[no] retries <dns_retries>`

Configure number of times to retry a request

Parameter	Type	Description
dns_retries	Integer	Number of times a request must be retried on each name server before giving up. Default is 1, allowed maximum is 4.

timeout

Configure the initial amount of time to wait for a response from a name server before retrying the query.

Use

Use to configure how long the system should wait for a reply from a domain name server, before trying the next server.

The default is 3 seconds. The maximum permitted is 30 seconds.

Default Setting

3

Command Mode

config-ip-dns

Syntax

```
[no] timeout <dns_timeout>
```

Configure the initial amount of time to wait for a response from a name server before retrying the query.

Parameter	Type	Description
dns_timeout	Integer	Initial timeout value in seconds. Default is 3 seconds, allowed maximum is 30 seconds. The resolver uses the initial timeout value for the first retry, and applies an exponential back-off algorithm for subsequent retries.

ip route

Configure global IPv4 settings.

Use

Use to configure the IPv4 routes for the system. You can set routes to go through a specific system interface or through another system, likely a router, on your network. Be sure to configure routes to include every subnet the system needs.

When configuring routes, you can set the MTU to use for that route. You can also set an MTU for each system interface, but the interface MTU will be overriden by the MTU setting here. Determine your MTU based on your network infrastructure.

Default Setting

None

Command Mode

configure

Syntax

`[no] ip route <subnet_cidr> <intf_name> [mtu <mtu>]`

Set the IPv4 route through a system interface; optionally specify the MTU for this route

`[no] ip route <v4_subnet> <v4_mask> <intf_name> [mtu <mtu>]`

Set the IPv4 route through a system interface; optionally specify the MTU for this route

`[no] ip route <subnet_cidr> <next_hop> [mtu <mtu>]`

Set the IPv4 route through a router; optionally specify the MTU for this route

`[no] ip route <v4_subnet> <v4_mask> <v4_next_hop> [mtu <mtu>]`

Set the IPv4 route through a router; optionally specify the MTU for this route

Parameter	Type	Description
intf_name	String	Name of an interface where all hosts on the destination network are directly connected.
mtu	Integer	MTU for this route
v4_subnet	IPv4Addr	Destination subnet in dotted decimal notation.
v4_mask	IPv4Addr	Destination subnet mask in dotted decimal notation.
v4_next_hop	IPv4Addr	Forwarding router's address.

subnet_cidr	IPAddrMask	Destination subnet and prefix length using CIDR notation.
next_hop	IPAddr	Forwarding router's address.

Related Commands

Interface Mode Commands
REST API Reference - ip

ipv6 route

Configure global IPv6 settings.

Use

Use to configure the IPv6 routes for the system. You can set routes to go through a specific system interface or through another system, likely a router, on your network. Be sure to configure routes to include every subnet the system needs.

Default Setting

None

Command Mode

configure

Syntax

`[no] ipv6 route <v6_addr> <intf_name>`

Set the IPv6 route through a system interface.

`[no] ipv6 route <v6_addr> <intf_name> <v6_next_hop>`

Set the IPv6 route through a router.

`[no] ipv6 route <v6_addr> <v6_next_hop>`

Set the IPv6 route through a router.

Parameter	Type	Description

intf_name	String	Name of an interface where all hosts on the destination network are directly connected or name of the interface for a link-local address.
v6_addr	IPv6AddrMask	Destination network and prefix length
v6_next_hop	IPv6Addr	Forwarding router's address

Related Commands

Interface Mode Commands

ip routing

Enable or disable permitting packets to flow through the system.

Use

Use to enable or disable IP gateway function to allow packets to flow through the system.

Enable when LineRate is the only gateway to another network, such as the Internet.

Default Setting

Disabled

Command Mode

configure

Syntax

`[no] ip routing`

Enables forwarding of IP traffic through the system

ip filter-list

Configure rules to allow or deny traffic.

Use

An IP filter is an ordered list of rules that you can apply to IP traffic to permit some traffic and deny other traffic. IP filters are highly scalable and can support over 100,000 rules in a single list. You can attach one IP filter list to a virtual IP to permit or deny traffic from specific client source IP addresses or subnets. To see the order in which the rules are processed, use:

- CLI command—**`show run`**

- REST API—/status/config/running

You can create multiple IP filter lists to use with different virtual IPs.

Default Setting

None

Command Mode

config

Syntax

```
[no] ip filter-list <filter_name>
```

Create a list of IP addresses and networks used for filtering

Parameter	Type	Description
filter_name	Word	Name of the IP filter list

Related Commands

virtual-ip attach ip-filter

show ip filter-list

priority

Configure the priority of IP filter rules.

Use

Configure the priority of a rule as a real number. Rules are ordered from low to high, with lower numbers representing higher priority. Multiple rules can share the same priority. You can insert a new rule at a new priority between two existing rules by choosing a priority number between those two rules, including a fractional priority number.

The system processes the rules with the highest priority first, then works through to the lower priority rules.

Default Setting

None

Command Mode

config

Syntax

`[no] <priority> deny <addr>`

Automatically close connections from IP addresses that match this rule without processing any requests.

`[no] <priority> deny <addrMask>`

Automatically close connections from IP addresses that match this rule without processing any requests.

`[no] <priority> permit <addr>`

Process requests from IP addresses that match this rule

`[no] <priority> permit <addrMask>`

Process requests from IP addresses that match this rule

Parameter	Type	Description
addr	IPAddr	IPv4 or IPv6 host address to match. See IP Address Notation.
addrMask	IPAddrMask	IPv4 or IPv6 subnet and mask to match. See IP Address Notation.
priority	Real	Priority of this rule as a real number. Rules are ordered from low to high, with lower numbers representing higher priority.

Key Mode Commands

Use the following commands to configure cryptographic keys for use with SSL.

key

Create or modify a cryptographic key.

Use

You set up a private key object to correspond to each primary certificate you need. The system supports using one private key to generate more than one primary certificate and the use of separate private keys for individual primary certificates.

You need access to your private key file. The LineRate software supports keys in PEM format.

We recommend giving each key a meaningful name that helps identify the key. For example, you might use the domain name or security settings in the name.

Default Setting

By default, the system creates a key, certificate, and SSL profile, each called self-signed, that you can use to test the SSL function.

Command Mode

configure

Syntax

```
[no] key <key_name>
```

Create or modify a cryptographic key

Parameter	Type	Description
key_name	Word	Name of the key object

Example

See Setting Up Certificates for SSL Termination

Related Commands

Certificate Mode Commands

SSL Mode Commands

REST API Reference - keys

pem-format

Specify the key content in a PEM-encoded (base64) string.

Use

Use to paste key content into a key.

- CLI command—After entering the command, press **Enter** to paste key text. Type **`quit`** to finish.
- REST API—Paste the key text into the data key.

Default Setting

None

Command Mode

config

Syntax

`pem-format`

Specify the key content in a PEM-encoded (base64) string.

Example

See Setting Up Certificates for SSL Termination

Related Commands

Certificate Mode Commands

SSL Mode Commands

Licensing Mode Commands

licensing

Configure licensing.

Use

To purchase a license, send an email to linerate-sales@f5.com or call 1.855.LINERATE (1.855.546.3728).

If you purchased a license for LineRate, you must provide information to identify the hardware, then you will receive download link for the license file in an email. You must install the license file.

The rate limits for HTTP requests, TCP connections, and Mb per second are based on the license you purchased.

Default Setting

None

Command Mode

config

Syntax

```
[no] licensing
```

Configure licensing.

Related

REST API Reference - licensing

feature

Install the license for a feature.

Use

Use to install the license you receive by email. Currently, the only feature you can license is called base. The license defines the expiration date of the license, the number of HTTP connections per second, and the number of TCP connections per second.

For the CLI, copy and paste the license from the email when prompted, then type **`quit`** on a line by itself.

Starting two days before the expiration date, a banner displays when you log in that feature license is expiring.

Default Setting

None

Command Mode

config-licensing

Syntax

`[no] feature base`

Install the license for base system

revocation-list

For future use.

signing-certificate

Configure X.509 licensing certificates.

Use

Use only when instructed to by LineRate personnel. Use to install the X.509 signing certificate for certain licenses.

For the CLI, copy and paste the certificate from the email when prompted, then type **`quit`** on a line by itself.

Default Setting

None

Command Mode

config-licensing

Syntax

`[no] signing-certificate <cert_name>`

Configure X.509 licensing certificates

Parameter	Type	Description
cert_name	Word	Name of certificate.

Load Balancer Mode Commands (Deprecated)

 In LineRate version 2.0, the load-balancer command is deprecated and is replaced by the proxy command. The corresponding REST API nodes are not affected.

Use the following commands to configure the load balancer. These are advanced commands.

load-balancer max-client-conns

Use

Every connection that the LineRate accepts consumes some system memory while it is being processed. The amount of memory is variable and affected by factors such as script processing, SSL offload and key size, and network conditions. If all memory on the system is exhausted, the system may terminate a proxy process to free up memory (an "out-of-memory kill"), which disrupts all traffic through that process.

We recommend correcting this situation by adding more memory to the system, dividing the load among more systems, or improving the memory efficiency of scripts. However, you can adjust this setting as a temporary solution, after consultation with technical support.

If your system experiences out-of-memory kills, you can configure this setting to allow fewer total client connections. This lessens the memory that is consumed, which prevents out-of-memory kills, but it also lessens the total number of clients that can be served at one time. When clients in excess of this limit attempt to connect, they are discarded with a TCP Reset. Many clients report this as "Connection Reset by Peer."

Default Setting

0 (disabled; no limit)

Command Mode

configure

Syntax

`[no] load-balancer max-client-conns <conns>`

Maximum number of simultaneous client connections

Parameter	Type	Description
conns	Integer	Maximum number of simultaneous client connections

Related

REST API Reference - maxClientConns

load-balancer processes

Use

Use to reduce the total number of processes dedicated to proxying traffic.

The default "auto" setting allocates the optimal number of processes. The system ignores commands to increase the number of proxy processes beyond the auto setting. Normally, you want to leave the default setting (auto), unless directed to change it by your sales engineer or technical support. Reducing the number of proxy processes may aid in script debugging or leave available CPU resources for management tasks.

The proxy processes are listed as "lb_http" in the output of the **`bash top -HS`** or **`bash ps aux`** command. There is always one extra lb_http process running for management.

Default Setting

auto

Command Mode

configure

Syntax

`[no] load-balancer processes <procs>`

Setting the number of processes used for the load balancer

`[no] load-balancer processes auto`

Allow the system to choose how many HTTP load balancer processes to run

Parameter	Type	Description
procs	Integer	Number of load balancer processes to run per system

Related

REST API Reference - processes

Logging Mode Commands

logging file

Use

Configure logging file name and options for a local log file. The file name can be an absolute path or a path relative to the /var/log directory.

For logging to start, you must set up one or more filters, attach them, and set the admin status to online.

Default Setting

None

Command Mode

config

Syntax

```
[no] logging file <file_name>
```

Parameter	Type	Description

file_name	String	Log file name. Can be an absolute path or a path relative to the /var/log directory.

Related

REST API Reference - syslog

admin-status

Use

Enable or disable logging to this file.

Default Setting

Offline

Command Mode

loggingfile

Syntax

`[no] admin-status offline`

Disable logging to this file

`[no] admin-status online`

Enable logging to this file

`no admin-status`

Remove the configured admin status

attach

Use

Attach an object, such as a filter, to this log file. If you attach multiple filters to the same file, the system logs all levels included in the filters.

Default Setting

None

Command Mode

loggingfile

Syntax

`[no] attach filter <filter_name>`

Attach the filter to filter messages to this log file

Parameter	Type	Description
filter_name	Word	Name of the configured filter.

logging filter

Use
Use to create filters that set the priority/severity of messages.

Default Setting

None

Command Mode

config

Syntax

`[no] logging filter <filter_name>`

Configure the types of the messages to log.

Parameter	Type	Description
filter_name	Word	Name of the filter.

Related

REST API Reference - syslog

level

Use

Use to configure the priority/severity of the messages to filter. Specify a single value or a range per command using either a number, symbolic name, numeric range, or a symbolic range. Ranges are indicated with a hyphen (-), but without any spaces.

Symbolic names (with the numeric value after) in high to low priority order are: emerg (0), alert (1), crit (2), err (3), warning (4), notice (5), info (6), and debug (7).

Default Setting

None

Command Mode

loggingfilter

Syntax

```
[no] level <level_str>
```

The priority/severity of the messages to filter

Parameter	Type	Description
level_str	Word	Specify the level as either a number, symbolic name, numeric range, or a symbolic range. Ranges are indicated with a hyphen (-), but without any spaces. Symbolic names (with the numeric value after) in high to low priority order are: emerg (0), alert (1), crit (2), err (3), warning (4), notice (5), info (6), and debug (7).

logging server

Use

Configure a name for a remote logging server to send logging data to. For logging to start, you must set up one or more filters, attach them, and set the admin status to online.

Default Setting

None

Command Mode

config

Syntax

`[no] logging server <server_name>`

Configure logging to a remote logging server

Parameter	Type	Description
server_name	Word	Configure a name for the server.

Related

REST API Reference - syslog

admin-status

Use

Enable or disable logging to this remote server.

Default Setting

Offline

Command Mode

loggingserver

Syntax

`[no] admin-status offline`

Disable logging to this remote server

`[no] admin-status online`

Enable logging to this remote server

no admin-status

Remove the configured admin status

attach

Use

Attach a filter to this remote logging server. If you attach multiple filters to the same server, the system logs all levels included in the filters.

Default Setting

None

Command Mode

loggingserver

Syntax

[no] attach filter <filter_name>

Configure the filter to filter messages to log

Parameter	Type	Description
filter_name	Word	Name of the configured filter

ip

Use

Configure the IP address of this remote logging server.

Default Setting

None

Command Mode

loggingserver

Syntax

[no] ip <addr>

Configure the IP address of this server

[no] ip

Remove the configured the IP address of this server

Parameter	Type	Description
addr	IPAddr	IP address for the remote server.

port

Use

Configure a different port on which this remote logging server receives messages.

Default Setting

514

Command Mode

loggingserver

Syntax

[no] port <port_num>

Configure the port on which this server receives messages

no port

Remove the configure the port on this server

Parameter	Type	Description
port_num	Integer	Port number to connect to this server.

protocol

Use

Configure the protocol to talk to this remote logging server. You can use either UDP or TCP.

Default Setting

UDP

Command Mode

loggingserver

Syntax

`[no] protocol tcp`

Use TCP to connect to this server

`[no] protocol udp`

Use UDP to connect to this server

`no protocol`

Remove the configured protocol for this server

No Command (Config Mode)

Negates or removes the specified command or item.

Objects That Can Have a Base

For objects that can have a base, use no to remove a direct setting from the object. The no form of the command tells the object to inherit that setting from its base. If the object does not have a base or that setting is not configured in its base, the object returns to its default.

For example, if you use **`no attach health-monitor <hm_name>`** for a real server, the specified health monitor will no longer be attached to the real server. However, the real server may still have a different health monitor attached to it directly, and it may inherit one or more health monitors from its base.

Another common example is the real server's admin status. You may want to create a real server base and set the admin status in the base to online. Then configure the real server to inherit its settings from the base. When you want to take just one real server called rs1 offline, use these commands:

```
real-server rs1

admin-status offline
```

This takes the real server offline, overriding the setting from the base. When you want the real server to inherit its admin status from its base again, use these commands:

```
real-server rs1

no admin-status offline
```

This ensures that the admin status is not set locally on the real server, but that it inherits its setting from the base.

Command Mode

config

Syntax

```
no <command> <value_to_remove>
```

Examples

```
no attach health-monitor <hm_name>
```

Objects That Cannot Have a Base, Commands That Can Have a Single Value

For objects that cannot have a base and commands that can have a single value, the no form of the command removes that value and returns the object back to its default.

For example:

- **`no description <desc>`**—Any object that can have a description, such as an interface, can have only one description, therefore, this command removes the description specified and returns the object to its default.
- **`no attach ssl profile <profile_name>`**—A REST server can have only one SSL profile attached, therefore, this command removes the SSL profile specified and returns the SSL profile setting to its default.

Command Mode

config

Syntax

```
no <command> <value_to_remove>
```

Objects That Cannot Have a Base, Commands That Can Have Multiple Values

For objects that cannot have a base and commands that can have multiple values, the no form of the command removes the specified value.

For example:

- **`no real-server <name>`**—Any object that can have a name, such as a real server or virtual server, can have more than one name, therefore, this command removes the real server specified, but does not affect other real servers you have created.
- **`no ip address <addr>`**—Real servers, interfaces, and other objects can have more than one IP address assigned to them, therefore, this command removes the specified IP address, but does not affect any other IP addresses assigned to the object.

- **`no hostname <name>`**—Virtual servers can have more than one hostname assigned to them, therefore, this command removes the specified hostname, but does not affect any other hostnames assigned to it.

Command Mode

config

Syntax

```
no <command> <value_to_remove>
```

npm Mode Commands

npm

Configure npm registry options. For information about installing Node Package Modules, see Scripting Mode Commands.

Use

Use to configure options for the LineRate Scripting npm, which is a package manager similar to the one provided for the Node.js platform.

The registry option lets you create one or more custom registries from which you can install Node Packaged Modules. If the registry requires secure access, you can attach a certificate to the registry.

Default Setting

None

Command Mode

config

Syntax

```
[no] npm registry <reg_name>
```

Create or manage npm registries.

Parameter	Type	Description
reg_name	Word	Name of the npm registry to create or manage.

Related

REST API Reference - npm

attach

Attach objects, such as certificates, to the npm registry.

Use

Use to attach a certificate to the secure location of a custom npm registry. You must first configure certificates to attach.

Default Setting

None

Command Mode

config-npm-registry

Syntax

`[no] attach certificate <cert_name>`

Attach or replace certificate to the npm registry.

Parameter	Type	Description
cert_name	Word	Name of the certificate to attach.

Related

Certificate Mode Commands

url

Specify the URL of a custom npm registry.

Use

Use to specify the URL of a custom npm registry to use for installing Node Packaged Modules.

Default Setting

None

Command Mode

config-npm-registry

Syntax

`[no] url <url_str>`

Specify the URL of the npm registry

Parameter	Type	Description
url_str	String	URL string. Protect URLs with quotes (for example, "http://registry.npmjs.org").

NTP Mode Commands (config)

ntp server

Use

Use to set an IP adress of a network time protocol server (NTP) to use to control the system time. You can set up more than one NTP server.

Default Setting

None

Command Mode

config

Syntax

`ntp server <addr>`

Sets the IP address of the NTP server to use

Parameter	Type	Description
addr	IPAddrI	IP address of NTP server.

Related

REST API Reference - ntp

LineRate

Phone Home Mode Commands

phone-home

Settings for configuring free tier licensing and sending phone home data.

Use

For the free tier license, you need to configure phone home with your DevCentral username and password, which you created when you downloaded the installation file. Your system will automatically contact the phone home server and configure your two-week, free tier license.

For phone home to work, you also need to configure an ip route. For the configuration needed to enable a free tier license and a purchased license, see Configuring Licensing.

Starting two days before expiration, the system attempts to phone home every minute to get a new license. A banner about the expiring license also displays when you log in. As long as the DevCentral username and password are configured, phone home will renew your license for another two weeks.

In addition to licensing, phone home sends the following data to F5:

Data Sent	Frequency	Purpose
Core files	As they occur	To proactively diagnose errors.
Output of **`show tech-support detailed`**	Hourly	To provide information about configuration and usage.
System logs (all files in /var/log/)	Hourly	To proactively diagnose errors.
Script events (create, remove, online, offline, run-time errors, inline script code)	As they occur, written to disk hourly (or when the amount collected exceeds a threshold	To proactively diagnose errors.
Per-script statistics	Every 30 minutes	To see how much scripting is being used.

Data Sent	Frequency	Purpose
On-disk scripts and dependencies (contents of /home/linerate/data/scripting/)	Every 4 hours	To proactively diagnose errors.

All of your passwords are protected in the phone home data.

Default Setting

None

Command Mode

config

Syntax

`[no] phone-home`

Settings for phone home

Related Commands

REST API Reference - phoneHome

userid

Configure DevCentral username and password.

Use

Use to configure the DevCentral username and password, which you created when you downloaded the installation file, for the free tier license and to send phone home data.

Normally, you should use the clear form of the password, and the system encrypts the password. You can see the encrypted password the following ways:

- CLI command—**`show run`**
- REST API—/config/phonehome/ihealth/password

Use the encrypted form of the password only when you need to use an existing password on another system. You can copy the encrypted password from the **`show run`** output or from the /status/config/running node, then paste it into the encrypted form of the command on another system.

Default Setting

None

Command Mode

config-phone-home

Syntax

`[no] userid <username> secret clear <password>`

Specifies an unencrypted password will follow

`[no] userid <username> secret encrypted <password>`

Specifies an encrypted password will follow

Parameter	Type	Description
password	String	The unencrypted (cleartext) DevCentral account password.
username	String	DevCentral username.

Proxy Mode Commands

proxy max-client-conns

Use

Every connection that the LineRate accepts consumes some system memory while it is being processed. The amount of memory is variable and affected by factors such as script processing, SSL offload and key size, and network conditions. If all memory on the system is exhausted, the system may terminate a proxy process to free up memory (an "out-of-memory kill"), which disrupts all traffic through that process.

We recommend correcting this situation by adding more memory to the system, dividing the load among more systems, or improving the memory efficiency of scripts. However, you can adjust this setting as a temporary solution, after consultation with technical support.

If your system experiences out-of-memory kills, you can configure this setting to allow fewer total client connections. This lessens the memory that is consumed, which prevents out-of-memory kills, but it also lessens the total number of clients that can be served at one time. When clients in excess of this limit attempt to connect, they are discarded with a TCP Reset. Many clients report this as "Connection Reset by Peer."

Default Setting

0 (disabled; no limit)

Command Mode

configure

Syntax

```
[no] proxy max-client-conns <conns>
```

Set the maximum number of simultaneous client connections

`no proxy max-client-conns`

Remove the maximum number of simultaneous client connections

Parameter	Type	Description
conns	Integer	Maximum number of simultaneous client connections

Related

REST API Reference - maxClientConns

proxy processes

Use

Use to reduce the total number of processes dedicated to proxying traffic.

The default "auto" setting allocates the optimal number of processes. The system ignores commands to increase the number of proxy processes beyond the auto setting. Normally, you want to leave the default setting (auto), unless directed to change it by your sales engineer or technical support. Reducing the number of proxy processes may aid in script debugging or leave available CPU resources for management tasks.

The proxy processes are listed as "lb_http" in the output of the **`bash top -HS`** or **`bash ps aux`** command. There is always one extra lb_http process running for management.

Default Setting

auto

Command Mode

configure

Syntax

`[no] proxy processes <procs>`

Setting the number of processes used for the proxy

`[no] proxy processes auto`

Allow the system to choose how many proxy processes to run

`no proxy processes`

Remove the manually configured number of processes used for the proxy

Parameter	Type	Description
procs	Integer	Number of proxy processes to run per system

Related

REST API Reference - processes

Real Server Mode Commands

Use the following commands to configure real servers. For an example real server configuration, see Configuring Real Servers.

real-server

Create or modify a real server for load balancing.

Use

Real servers represent a service, for example a web server, that the load balancer (reverse proxy) is distributing the client requests to. Each load balancer requires at least one real server. The IP address and port for the real server must match the IP address and port of the service on the server or proxy server the real server talks to.

You can create multiple real servers, for example, for different application types, hardware capabilities (such as CPU or memory), or security settings (such as SSL).

We recommend giving each real server a meaningful name, based on its use. When naming real servers, also consider how you want to group them and use names that facilitate grouping using simple regular expressions. For example, naming real servers as rs-ssl1, rs-ssl2, and so on, permits the use of a simple regular expression (rs-ssl.*) to add the real servers to a group.

For more information, see:

CLI Reference - group and members

REST API Reference - realServerGroup and memberRegex

We also recommend creating one or more real server bases to make configuring real servers more consistent. See base.

Default Setting

By default, no real servers exist.

When you create a real server, the default settings are:

- admin-status—offline
- attach—nothing attached
- base—none
- description—none
- ip address—none
- service—service is set to http

Command Mode

config

Syntax

In one command, you can set the real server name, IP address, port, base, and group, as shown in the syntax examples.

`[no] real-server <real_server_name>`

Create or modify a real server.

`[no] real-server <real_server_name> ip <addr> <port>`

Create or modify a real server and assign it an IP address and port.

`[no] real-server <real_server_name> ip <addr> <port> base <base_name_inherit>`

Create or modify a real server, assign it an IP address and port, and set the base of the real server.

`[no] real-server base <base_name>`

Create or modify a real server base.

`no base`

Remove all bases from the real server.

Parameter	Type	Description
addr	IPAddr	IPv4 or IPv6 address for the real server. See IP Address Notation.
base_name	Word	Name of the base to create or modify. See base.
base_name_inherit	Word	Base from which this real server should inherit. See base.
port	Integer	Port number to connect to the real server
real_server_name	Word	Name of the real server

Related

REST API Reference - realServer

admin-status

Bring an object, such as a health monitor, real server, or virtual IP, online or offline. After you create an object, you must bring it online.

Use

You typically set the offline status only when you want to disable the object or block connections to the web server during maintenance or system reconfiguration.

Default Setting

offline

Command Mode

config

Syntax

admin-status offline

Bring the object offline

admin-status online

Bring the object online

no admin-status

Remove the directly configured admin status

attach

Attach an object to the real server.

Use

Use to attach an object, such as a health monitor, NAT profile, SSL profile, or TCP options group, to the current real server.

Default Setting

None

Command Mode

config

Syntax

[no] attach health-monitor <hm_name>

Attach a health monitor to the current real server. You can attach multiple health monitors identified by unique names.

[no] attach nat profile builtin client-source-address

Attach and enable a built-in NAT profile that causes the proxy to use the client's address to establish a connection to the current real server. The client-source-address is the only available NAT profile.


For an active/active configuration (as described in this example), you cannot use the client-source-address NAT profile. Using this NAT profile would not permit the real server to know which LineRate system to send traffic back to, because the real server would only know the client's IP address not the IP address of the LineRate system.

`[no] attach ssl profile <ssl_profile_name>`

Attach or replace the SSL profile for the current real server. Only one ssl profile can be attached to a real server.

`[no] attach tcp-options-group <tcp_options>`

Attach or replace the TCP options group for connections to the current real server.

Parameter	Type	Description
hm_name	Word	Name of the health monitor to attach. See Health Monitor Mode Commands and Configuring a Health Monitor.
ssl_profile_name	Word	Name of the SSL profile to attach. See Configuring SSL.
tcp_options	Word	TCP options group name

description

Create a description for the real server.

Use

Use to create a description of the real server use and any other information.

Default Setting

None

Command Mode

config

Syntax

`description <desc>`

Description for the real server

`no description`

Remove the description

Parameter	Type	Description
desc	String	Description string. Enter a maximum of 255 characters.

ip address

Set the IP address for the real server.

Use

Use to set the IP address and port for the real server. The IP address must match the IP address and port of the server or proxy server the real server talks to.

Default Setting

None

Command Mode

config

Syntax

`[no] ip address <addr> <port>`

IPv4 or IPv6 address for the real server

`no ip`

Remove the configured IP address

Parameter	Type	Description
addr	IPAddr	IPv4 or IPv6 address for the real server
port	Integer	Port number to connect to the real server

max-connections

Define the maximum number of simultaneous connections to the current real server.

Use

Set this to the maximum number of connections the service running on a server can handle.

Default Setting

0 (no limit)

Command Mode

config

Syntax

`[no] max-connections <max>`

Maximum number of connections to the current real server

`no max-connections`

Remove the directly configured maximum number of connections to the current real server

Parameter	Type	Description
max	Integer	Number of connections

service http

Configure HTTP as the service type for the real server.

Use

Sets the current real server to be compatible with layer 7 load balancing, for web use. Use this for web servers that are using HTTP. The service setting on a real server must match the service setting on any virtual server to which the real server is attached. Use for HTTPS as well.

Command Mode

config

Default Setting

Service default is http.

Syntax

```
service http
```

is-proxy

Indicate if the real server is a proxy server.

Use

Use this only when the real server is talking to a proxy server rather than an actual web server.

Command Mode

config

Default Setting

Disabled

Syntax

```
[no] is-proxy
```

Indicate if the real server is an HTTP proxy server

Related Commands

tunnel timeout

keepalive-timeout

Close the real server TCP connection if there is no HTTP activity in this amount of time.

Use

If there are no active HTTP transactions (that is, no active requests or responses) to a server for the specified time (in seconds), the system closes the TCP connection to the server, reclaiming resources. This can help avoid problems that some HTTP servers have when connections are kept open indefinitely.

Setting this to a value of 0 disables the timeout, which directs the system not to close server TCP connections due to this timeout.

Command Mode

config

Default Setting

0 (disabled)

Syntax

`[no] keepalive-timeout <timeout>`

Close TCP connection if there is no HTTP activity in this amount of time

`no keepalive-timeout`

Remove the configured keepalive timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional)

max-in-flight

Maximum number of requests that can be pending on one HTTP connection at a time.

Use

The default of 1 lets the system send only one request at a time on a connection. The system waits for the web server to send the response before sending the next request. The default works in many configurations, especially when the web server has low latency.

If you set this to more than 1, the system can send the specified number of requests to the web server on a single connection, without waiting for the response to each request. Use a higher number when the web server has high latency. This should not exceed the pipeline depth setting of the web server.

Command Mode

config

Default Setting

1

Syntax

`[no] max-in-flight <requests>`

Maximum number of requests that can be pending on one HTTP connection at a time

`no max-in-flight`

Remove the directly configured max in flight

Parameter	Type	Description
requests	Integer	Number of requests

request-idle-timeout

Configure the number of seconds to wait to receive any part of a request before closing the connection.

Use

The system closes the connection if it takes longer than the specified time (in seconds) to receive any part of the request from the client.

Consider the size of a typical client request as well as the user environment to set this value. For example, an application where users upload photos or HD videos using mobile devices would need a longer timeout than simple web pages due to mobile bandwidth and device processing limitations.

Command Mode

config

Default Setting

0 (disabled)

Syntax

`[no] request-idle-timeout <timeout>`

Number of seconds to wait for new request data before closing the connection

`no request-idle-timeout`

Remove the directly configured request idle timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional)

request-rate-limit

Set the real server rate limit.

Use

Use to set the number of HTTP requests per second to send to the real server. Consider your traffic patterns and the capabilities of all of the servers this real server handles.

Use the burst size to set the maximum number of HTTP requests the servers can handle in a burst above the rate limit. This value may be no less than 2% of the rate limit. By default, the burst size is the same as the request rate limit you configure. For most situations, this works well. Change this setting only if your requests tend to come in bursts.

Default Setting

0 (no limit)

Command Mode

config-rserver-http

Syntax

`[no] request-rate-limit <limit>`

Set the real server rate limit

`no request-rate-limit`

Remove the directly configured rate limit

`[no] request-rate-limit <limit> burst-size <burst>`

Set the maximum burst size (default is the rate limit)

Parameter	Type	Description
burst	Integer	Maximum number of requests handled in a burst above the rate limit. This value may be no less than 2% of the rate limit.
limit	Integer	Limit in requests per second

response-idle-timeout

Configure the number of seconds to wait for new response data before closing the connection.

Use

The system closes the connection if, after transmitting the request to the server, it takes longer than the specified time (in seconds) either to receive any part of the response from the HTTP server or to transmit any part of the response to the client.

Consider the size of a typical response for your application as well as the user environment to set this value. For example, an application where users download HD videos using mobile devices would need a longer timeout than simple web pages due to mobile bandwidth and device processing limitations.

Command Mode

config

Default Setting

0 (disabled)

Syntax

`[no] response-idle-timeout <timeout>`

Number of seconds to wait for new response data before closing the connection

`no response-idle-timeout`

Remove the directly configured response idle timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional)

response-timeout

Configure the number of seconds to wait for the server to respond to a request.

Use

The system sends an HTTP 504 error response to the client and closes the connection if the HTTP server takes longer than the specified time (in seconds) to respond to a request.

Consider the amount of time the web server takes to respond to any request. The response-timeout must always be configured to be higher than the amount of time it takes for any of the web servers to respond to a request.

Command Mode

config

Default Setting

0 (disabled)

Syntax

`[no] response-timeout <timeout>`

Number of seconds to wait for the server to respond to a request

`no response-timeout`

Remove the directly configured response timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional)

tunnel idle-timeout

Configure HTTP tunnel options.

Use

The system closes the tunnel connection if it takes longer than the specified time (in seconds) to receive a request from the client.

Use this only when the real server is talking to an HTTP proxy server rather than an actual web server.

Configuration Mode

config

Default Setting

0 (disabled)

Syntax

`[no] tunnel idle-timeout <timeout>`

Closes the HTTP tunnel connection if there is no activity in this amount of time

`no tunnel idle-timeout`

Remove the directly configured tunnel idle timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional).

Related Commands

isproxy

service tcp

Configure TCP as the service type for the real server. The system does load balancing of TCP connections without processing the TCP payload.

Use

Use this service type for servers that are not using HTTP, for example, an email server.

Configuration Mode

config

Default Setting

Service default is http.

Command Mode

config

Syntax

```
[no] service tcp
```

Sets the service type to TCP

data-idle-timeout

Use

The system closes the TCP connection if there is no activity in this amount of time.

Configuration Mode

config

Default Setting

0 (disabled)

Syntax

```
[no] data-idle-timeout <timeout>
```

Closes the TCP connection if there is no activity in this amount of time

```
no data-idle-timeout
```

Remove the directly configured data idle timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional)

base

A "base" in LineRate is a type of template that allows you to reuse common portions of configuration across multiple objects. Each base can inherit from another base, overriding properties from that base. This lets you create basic configurations that you can reuse and build upon.

Use the no command to remove the specified base from the real server. The real server will then use the parameters configured specifically for the real server or the default settings for any parameter that is not configured.

See the Related Commands for the commands you can use when configuring a real server base.

Default Setting

None

Command Mode

config

Syntax

```
[no] base <base_name>
```

Create or modify a real server base.

`no base`

Remove all bases from the real server.

Parameter	Type	Description
base_name	Word	Name of real server base to inherit from

Example

See Creating a Real Server Base.

Related Commands

You can use the following real server commands when configuring a real server base:

admin-status

attach

max-connections

service - including all settings for HTTP and TCP service

request-rate-limit

REST API Reference - realServerBase

group

Create a real server group.

Use

Use to create real server groups to make configuration more efficient. You can attach a real server group to a virtual server. Group real servers based on those served by the same virtual IP and virtual server.

Configuration Mode

config

Default Setting

None

Syntax

```
[no] real-server group <rs_group_name>
```

Create a real server group.

Parameter	Type	Description
rs_group_name	Word	Name of the real server group to create.

Related

REST API Reference - realServerGroup

member

Add the named real server to the group.

Use

Use to add individual real servers to the group.

Configuration Mode

config

Default Setting

None

Syntax

```
[no] member <real_server_name>
```

Add named real server to the group

Parameter	Type	Description

real_server_name	Word	Real server name

members

Specify group membership for multiple real servers.

Use

Use a regular expression to dynamically add multiple real servers to the group. The system automatically adds real servers to the group, as they are created, if their names match the regular expression.

Configuration Mode

config

Default Setting

None

Syntax

`[no] members by regex <regex>`

Real servers that match regex will be members of the group

Parameter	Type	Description
regex	String	Regular expression to use to match real server names (Perl syntax)

Examples

`members by regex "^rs.*"`

Adds any real server whose name starts with rs to the group

REST Server Mode Commands

Use the following commands to configure the REST server.

rest-server

Configure access to the REST server.

Use

Use to configure the HTTP server on the LineRate system for Representational State Transfer (REST) access. Log in to the REST server using the same login and password that you use for the LineRate system. By default, the REST server is activated on SSL port 8443.

The REST server uses the following HTTP verbs: GET, PUT, POST, and DELETE and lets you do the following:

- Configure the system (add, change, or delete configuration)
- Retrieve system configuration
- Retrieve various statistics and counters used to monitor the system

By default, the system configuration permits access to the REST server on any local interface (on port 8443) from any remote host. To connect to the REST server, point your client (custom REST client application or browser) to the REST server's IP address and port to establish a secure HTTP connection.

Before interacting with the REST API, you must do the following:

1. Send a POST request to path /login with the message body containing the username and password credentials. For example, using curl you could authenticate with the following request:

   ```
   curl -k -d "username=admin&password=changeme"
       "https://<server_ip>:<server_port>/login"
   ```
2. Successful authentication redirects you to the top-level path ("/") and returns a cookie with a unique session ID. Authentication failure redirects you back to the login path (/login).
3. Send the session ID returned as part of the authentication response cookie in all future requests to continue using the session. For example, using curl you could continue using the session as follows:

```
curl -k -i "https://<server_ip>:<server_port>/lrs/api/?op=list" -H "Cookie:
   connect.sid:...."
```

To log out, send a GET request to path /logout. For example, using curl you could log out using the following request:

```
curl -k -i "https://<server_ip>:<server_port>/logout" -H "Cookie: connect.sid:...."
```

Default Settings

allow from any

allow to any 8443

Command Mode

config

Related

REST API Reference Guide

REST API Reference - restServer

allow from

Specify remote address information.

Use

Permit connections from the specified IP addresses to the REST server.

Default Setting

allow from any

Command Mode

config

Syntax

```
[no] allow from <v4Addr>
```

Remote IPv4 address for a single remote host

```
[no] allow from <v4Subnet> <v4Mask>
```

Remote IPv4 subnet and mask using subnet and net mask notation

[no] allow from <v4AddrMask>

Remote IPv4 subnet and mask using CIDR notation

[no] allow from <v6AddrMask>

Remote IPv6 subnet and mask using CIDR notation

[no] allow from any

Allow access from any remote network

Parameter	Type	Description
v4Addr	IPv4Addr	Remote IPv4 address for a single host
v4AddrMask	IPv4AddrMask	Remote IPv4 subnet and mask using CIDR notation
v4Mask	IPv4Mask	Remote IPv4 network mask using net mask notation
v4Subnet	IPv4Subnet	Remote IPv4 subnet using net mask notation
v6AddrMask	IPv6AddrMask	Remote IPv6 subnet and mask using CIDR notation

allow to

Specify local address information.

Use

Permit connections to the specified IP addresses to access the REST server. The system uses port 8443 for access to the REST server. The IP addresses you specify must be configured on a system interface.

Default Setting

allow to any 8443

Command Mode

config

Syntax

`[no] allow to <addr>`

Local address to access

`[no] allow to <addr> <port>`

Local address and port to access

`[no] allow to any`

Allow connections to all local IP addresses

`[no] allow to any <port>`

Allow connections to all local IP addresses using the specified port

Parameter	Type	Description
addr	IPAddr	IPv4 or IPv6 address on a local interface
port	Integer	TCP port number on which to allow access

attach

Use

Use to attach an SSL profile to the REST server.


The REST server is not accessible until an SSL profile is attached to it.

Default Setting

By default, the system creates a key, certificate, and SSL profile, each called self-signed, that you can use to test the SSL function.
The default SSL profile is attached to the REST server.

Command Mode

config

Syntax

[no] attach ssl profile <name>

Attach or replace the SSL profile for the REST server.

Parameter	Type	Description
name	Word	Name of the SSL profile to attach.

Related Commands

SSL

logging

Enable verbose logging that tracks all REST server commands.

Use

Without turning on logging, the system only logs logins and logouts. Normally, you do not want to turn on logging. Use logging when debugging REST server issues.

To access the log file, use SSH to access the system, go into bash, and look at the file /var/log/controller.messages.

Default Setting

Disabled

Command Mode

config

Syntax

[no] logging

session-idle-timeout

Close login session if there is no activity in the specified amount of time.

Use

The system uses a persistent cookie, so you stay logged in until the specified time elapses or until you log out of the REST server.

Default Setting

3600 seconds

Command Mode

config

Syntax

`[no] session-idle-timeout <timeout>`

Closes login session if there is no activity in this amount of time

`no session-idle-timeout`

Removes the configured session idle timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional).

Scheduler Mode Commands

scheduler

Use

Configure operating system process scheduling. Normally, you do not need to change this configuration.

Setting this to **`manual`** may affect system performance, as well as the response to management input. If you set **`scheduler cpu`** to **`manual`**, we recommend that you also configure scheduling for each class of processes using **`process-class`**, with the assistance of technical support personnel.

Default Setting

None

Command Mode

config

Syntax

`scheduler cpu auto`

Allow the system to automatically assign CPU resources to processes

`scheduler cpu manual`

Manually assign CPU resources to processes.

Related

REST API Reference - scheduler

process-class

Use

Configure scheduling for a class of processes. If you set **scheduler cpu** to **manual**, we recommend that you also configure scheduling for each class of processes using **process-class**, with the assistance of technical support personnel.

Default Setting

cpu auto

Command Mode

schedulercpu

Syntax

[no] process-class clock <cpu_list>

Process that handles operating system periodic tasks

no process-class clock

Remove the configured CPU list from the process that handles operating system periodic tasks

[no] process-class default <cpu_list>

Includes all processes that are not already manually configured

no process-class default

Remove the configured CPU list from default, which includes all processes that are not already manually configured

[no] process-class health <cpu_list>

Processes for the health monitor

no process-class health

Remove the configured CPU list from the processes for the health monitor

[no] process-class lb-http <cpu_list>

Processes for the proxy (both HTTP and TCP)

no process-class lb-http

Remove the configured CPU list from the processes for the proxy (both HTTP and TCP)

[no] process-class net-rx <cpu_list>

Processes that handle receiving network traffic

no process-class net-rx

Remove the configured CPU list from the processes that handle receiving network traffic

[no] process-class redis-server <cpu_list>

Process that provides key/value cache for the LineRate Scripting (see http://redis.io/)

no process-class redis-server

Remove the configured CPU list from the process that provides key/value cache for the LineRate Scripting

[no] process-class regex <regex> <cpu_list>

A regular expression that will be matched against process names to select processes included in this class

Parameter	Type	Description
cpu_list	String	Comma-separated list of CPU numbers that this process class is allowed to run on. A range can be specified with a dash. Must be enclosed in double quotes. Example: "0,4-7"
regex	String	A regular expression that will be matched against process names to select processes included in this class.

Script Mode Commands

script

Configure or modify a script.

Use

Use to create or change scripts and to configure script settings. For more information about scripts, see the Scripting Developer's Guide and Scripting API Reference Guide.

Default Setting

None

Command Mode

config

Syntax

```
[no] script <script_name>
```

Configure or modify a script

Parameter	Type	Description
script_name	Word	Name of the script.

Related

REST API Reference - script

admin-status

Bring an object, such as a health monitor, real server, or virtual IP, online or offline. After you create an object, you must bring it online.

Use

You typically set the offline status only when you want to disable the object or block connections to the web server during maintenance or system reconfiguration.

Default Setting

offline

Command Mode

config

Syntax

`admin-status offline`

Bring the object offline

`admin-status online`

Bring the object online

`no admin-status`

Remove the directly configured admin status

edit

Edit an existing script or create a new one using a text editor.

Use

Use to edit an existing script or create a new script using an editor. You can edit a script directly in LineRate and make changes to simple scripts or small changes to running scripts. For larger scripts or in a production environment, we recommend using a version control system to track changes to scripts.

You can edit scripts if they are inline (defined using the **`source inline`** command) or if they are in a file (**`source file`**). If a script has not been previously created, using the editor makes it **`source inline`**.

After you save and exit the editor, the changes are applied: each proxy process stops running the old script and starts running the new script, as if you had configured it using **`source inline`** or **`source file`**. If you save a script with a syntax error, the old script is halted, and the new script will fail to run.

The default and only supported editor is pico. In pico, save by pressing `Ctrl-O` and exit by pressing `Ctrl-X`. If you are prompted to save to a particular file name, accept the file name that the editor suggests (in the case of **`source inline`** this will be a temporary file; **`for source file`** it will be the existing file name).

The emacs and vim editors are experimental in this release.

The documentation for editors is available from the following:

- **pico (default)** (pico-alpine-2.00_1): http://www.washington.edu/alpine/mans.html#pico or type `Ctrl-G` in pico
- **emacs** (emacs21): http://www.delorie.com/gnu/docs/emacs/emacs_toc.html or type `Ctrl-h Ctrl-h` in emacs
- **vim** (vim-lite-7.3.669): http://vimdoc.sourceforge.net/htmldoc/usr_toc.html or type `:help <cr>` in vim

Default Setting

pico

Command Mode

config-script

Syntax

`[no] edit [emacs | vim]`

Edit the current script using the default editor (pico) or the specified editor.

restart-mode

Set script restart mode.

Use

Use to set the script restart mode. This setting defines the restart behavior when a script encounters unrecoverable errors:

- auto (default)—Script automatically restarts on unrecoverable errors.
- manual—Script does not restart automatically on unrecoverable errors. To restart in manual mode, set the admin-state to offline and then online.

Default Setting

auto

Command Mode

config-script

Syntax

`[no] restart-mode auto`

Set script to automatically restart on unrecoverable errors.

`[no] restart-mode manual`

Set script to not restart automatically on unrecoverable errors. To restart in manual mode, set the admin-state to off and then on.

source

Specify JavaScript source for the script.

Use

Use to specify JavaScript source for the script:

- inline—Type or copy and paste a script directly into the command line.
- file—Call a file that contains the script. Recommended for longer scripts of about 20 or more lines.

Default Setting

None

Command Mode

config-script

Syntax

`source file <file_str>`

Specify path name to file containing source.

`source inline <end_word>`

Specify source inline.

Parameter	Type	Description
end_word	String	Specify end-word string to signal the end of the source. Note: Enter this string on a line by itself at the end of the source block to indicate the end of the JavaScript source code.
file_str	String	Path to the script file, which can be absolute (for example, "/home/linerate/data/scripting/proxy/myscript.js") or relative to the scripting home directory /home/linerate/data/scripting (for example, "proxy/myscript.js"). Protect path name with quotes.

SNMP Mode Commands

snmp-server

Use

Use to configure the SNMP server (agent) to permit access to SNMP data.

The MIBs that the system uses are available from the following locations:

- F5-specific MIBs—/usr/linerate/mibs on any LineRate system.
- Standard RFC MIBs—/usr/local/share/snmp//mibs on any LineRate system.

To view the snmp-server settings, use **`show run`**.

Default Setting

None

Command Mode

configure

Syntax

`[no] snmp-server`

Configure snmp agent options

Related

REST API Reference - snmp

community

Use

Use to configure the string to permit read-only access to SNMP.

Default Setting

None

Command Mode

snmpserver

Syntax

`[no] community <comm_str>`

Configure the community access string

Parameter	Type	Description
comm_str	Word	Specify the community string token

contact

Use

Use to configure system contact information for SNMP.

Default Setting

None

Command Mode

snmpserver

Syntax

`[no] contact <contact_info>`

Configure the system contact information

`no contact`

Remove the configured contact information

Parameter	Type	Description
contact_info	String	String that describes the system contact information

enable

Use

Use to enable one or more IP addresses on the system for SNMP access. Typically, you use the management IP address.

Use the "all" setting to permit SNMP access on each IP address configured on the system.

Default Setting

Default port—161

Command Mode

snmpserver

Syntax

`[no] enable udpv4 <addr>`

Configure the IPv4/UDP parameters for the SNMP agent

`[no] enable udpv4 <addr> <port>`

Configure the SNMP agent to accept UDP connections only on this IPv4 address

`[no] enable udpv4 all`

Configure the SNMP agent to accept UDP connections on all IPv4 addresses

`[no] enable udpv4 all <port>`

Configure the SNMP agent to accept UDP connections on all IPv4 addresses

`[no] enable udpv6 <addr>`

Configure the IPv6/UDP parameters for the SNMP agent

`[no] enable udpv6 <addr> <port>`

Configure the SNMP agent to accept UDP connections only on this IPv6 address

`[no] enable udpv6 all`

Configure the SNMP agent to accept UDP connections on all IPv6 addresses

`[no] enable udpv6 all <port>`

Configure the SNMP agent to accept UDP connections on all IPv6 addresses

Parameter	Type	Description
addr	IPAddr	Configure the SNMP agent to accept UDP connections only on this IP address.
port	Integer	Configure the SNMP agent to accept UDP connections only on this port.

location

Use

Use to configure information to identify the system's location.

Default Setting

None

Command Mode

snmpserver

Syntax

`[no] location <location_info>`

Configure the system location information

`no location`

Remove the configured location information

Parameter	Type	Description
location_info	String	String that describes the system location information

SSH Mode Commands

Use the following commands to configure SSH access to the LineRate system.

ssh

Configure options for Secure Shell (SSH) access to the LineRate system.

Use

SSH provides secure, remote access to the system.

Default Settings

allow from any

allow to any

Command Mode

config

Related

REST API Reference - ssh

allow from

Use

Permit SSH access from the specified IP address or network.

Default Setting

```
allow from any
```

Command Mode

Config

Syntax

`[no] allow from <v4Addr>`

Remote IPv4 address for a single remote host

`[no] allow from <v4Subnet> <v4Mask>`

Remote IPv4 subnet and mask using subnet and net mask notation

`[no] allow from <v4AddrMask>`

Remote IPv4 subnet and mask using CIDR notation

`[no] allow from <v6AddrMask>`

Remote IPv6 subnet and mask using CIDR notation

`[no] allow from any`

Allow access from any remote network

Parameter	Type	Description
v4Addr	IPv4Addr	Remote IPv4 address for a single host
v4AddrMask	IPv4AddrMask	Remote IPv4 subnet and mask using CIDR notation
v4Mask	IPv4Mask	Remote IPv4 network mask using net mask notation
v4Subnet	IPv4Subnet	Remote IPv4 subnet using net mask notation
v6AddrMask	IPv6AddrMask	Remote IPv6 subnet and mask using CIDR notation

allow to

Use

Permit SSH access to the specified IP address on the LineRate system.


The SSH server listens on the default port, 22, if a port number is not specified in the configuration.

Default Setting

```
allow to any
```

Command Mode

Config

Syntax

```
[no] allow to <addr>
```

Local address to access

```
[no] allow to <addr> <port>
```

Local address and port to access

```
[no] allow to any
```

Allow connections to all local IP addresses

```
[no] allow to any <port>
```

Allow connections to all local IP addresses using the specified port

Parameter	Type	Description
addr	IPAddr	IPv4 or IPv6 address on a local interface
port	Integer	TCP port number on which to allow access

SSL Mode Commands

SSL Overview

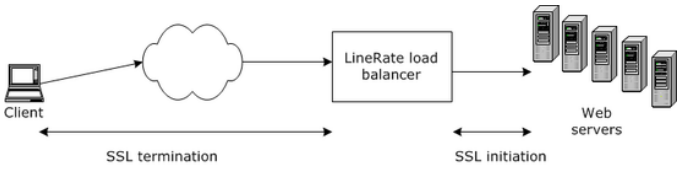
Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are closely related technologies that provide communication security over an insecure network, such as the Internet. TLS is a standardized protocol, defined by IETF RFCs, and is the successor to the non-standardized SSL protocol. The LineRate software supports both TLS and SSL, but the system and documentation refers to both protocols collectively as "SSL," following the most common industry terminology.

Use

The LineRate software supports two types of SSL connections:

- SSL termination—SSL connection from the client to the LineRate load balancer.
- SSL initiation—SSL connection from the LineRate load balancer to the web server.

The diagram below shows the two types of SSL.



SSL configuration in LineRate

By using the SSL termination feature in LineRate, you can move the computationally intensive SSL processing off your web servers and onto the LineRate, allowing your web servers to concentrate on performing application tasks. Or, if your application requires greater security on your internal network, you can use SSL initiation together with SSL termination to provide end-to-end SSL security, while still allowing the LineRate to do full layer 7 load balancing.

For an example SSL configuration, see Configuring SSL.

ssl profile

Create or configure an SSL profile.

Use

Use to define the security settings you want to use for SSL access. You can use an SSL profile for either termination SSL or initiation SSL or both. You need a separate SSL profile for each unique primary certificate that you want to use with a virtual IP. You can also have separate SSL profiles for the same primary certificate, but use different settings in each profile.

We recommend giving each SSL profile a meaningful name that helps identify it. For example, you might use the domain name or security settings in the name.

Attaching an SSL profile to a virtual IP configures that virtual IP to always use SSL. The virtual IP will no longer accept connections from clients unless they perform SSL negotiation.

We also recommend creating one or more SSL profile bases to make configuring SSL profiles more consistent.
See base.

Default Setting

By default, the system creates a key, certificate, and SSL profile, each called self-signed, that you can use to test the SSL function.

Command Mode

Config

Syntax

`[no] ssl profile <profile_name>`
Creates or manages the specified SSL profile

`[no] ssl profile base <base_name>`
Creates or manages the specified SSL profile base

`no base`
Removes all configured bases from the SSL profile

Parameter	Type	Description

base_name	Word	Name of the SSL profile base to create or manage
profile_name	Word	Name of the SSL profile to create or manage

Related Commands
virtual-ip

REST API Reference - ssl

attach

Attaches objects to the SSL profile.

Use

Use to attach an object, such as certificates and keys, to the SSL profile.

The system lets you attach one or more individual chain certificates as well as certificate bundles to an SSL profile. Each chain certificate identifies an intermediate Certificate Authority (CA) that can authenticate the primary certificate for the profile.

A certificate bundle is a single file that contains multiple chain certificates concatenated together. The bundle can include related and unrelated chain certificates. The system automatically looks for the chain certificates that correspond to the primary certificate.

Default Setting

None

Command Mode

Config

Syntax

`attach chain-certificate <cert_name>`

Applies only to termination SSL. Add certificates to the certificate chain for this profile. You can associate multiple chain certificates and/or chain certificate bundles with a profile.

`attach chain-certificate bundle <bundle_name>`

Applies only to termination SSL. Add a bundle of certificates to the certificate chain for this profile.

`attach primary-certificate <cert_name>`

Applies only to termination SSL. A certificate to be presented as the identity of this system during SSL negotiations. A profile can have only one primary certificate.

attach private-key <key_name>

Applies only to termination SSL. The private key that was used to create the primary certificate associated with this profile.

Parameter	Type	Description
bundle_name	Word	Name of a certificate bundle to add to the certificate chain
cert_name	Word	Name of a certificate to be used for identification of this system
key_name	Word	Name of a private key

cipher-list

Controls which ciphers this profile will allow.

Use

Use to set the encryption algorithm, authentication method, and the protocol version the SSL profile uses for SSL connections. The system uses the OpenSSL library. The cipher list applies to both termination and initiation SSL. For termination SSL, the system uses the order set here to control the cipher priority.

To see how the cipher string resolves based on what you enter, use of of the following to look at the ordered cipher list:

- CLI command— **show ssl profile <ssl_profile_name>**
- REST node— /status/ssl/profile/<name>/cipherNameList

The Westmere-class processor (required, see System Requirements) accelerates AES processing, so we recommend placing AES higher in the list to improve performance.

For authentication, the system supports RSA and DSS certificates. For the protocol version, the system supports SSLv2, SSLv3, and TLSv1.

Cipher Strings

For the encryption algorithm, the system supports the following cypher strings:

- ALL—All supported cipher suites except the eNULL ciphers which must be explicitly enabled. The cipher suites are reasonably ordered by default.

- COMPLEMENTOFALL—The cipher suites not enabled by ALL, currently being eNULL.
- HIGH—High encryption cipher suites. This currently means those with key lengths larger than 128 bits, and some cipher suites with 128-bit keys.
- MEDIUM—Medium encryption cipher suites, currently some of those using 128-bit encryption.
- LOW—Low encryption cipher suites, currently those using 64- or 56-bit encryption algorithms but excluding export cipher suites.
- EXP, EXPORT, EXPORT40—Export encryption algorithms.
- eNULL, NULL—The NULL ciphers, that is, those offering no encryption. Because these offer no encryption at all and are a security risk they are disabled unless explicitly included.
- aNULL—The cipher suites offering no authentication. This is currently the anonymous DH algorithms. These cipher suites are vulnerable to a "man in the middle" attack, so their use is normally discouraged.
- kRSA, RSA—Cipher suites using RSA key exchange.
- kEDH—Cipher suites using ephemeral DH key agreement.
- aRSA—Cipher suites using RSA authentication, that is, the certificates carry RSA keys.
- aDSS, DSS—Cipher suites using DSS authentication, that is, the certificates carry DSS keys.
- TLSv1, SSLv3, SSLv2—TLSv1, SSLv3, or SSLv2 cipher suites respectively.
- DH—Cipher suites using DH, including anonymous DH.
- ADH—anonymous DH cipher suites.
- AES—Cipher suites using AES.
- 3DES—Cipher suites using triple DES.
- DES—Cipher suites using DES (not triple DES).
- RC4—Cipher suites using RC4.
- RC2—Cipher suites using RC2.
- IDEA—Cipher suites using IDEA.
- SEED—Cipher suites using SEED.
- MD5—Cipher suites using MD5.

- SHA1, SHA—Cipher suites using SHA1

Cipher List Format

The cipher list consists of one or more cipher strings separated by colons. Commas or spaces are also acceptable separators but colons are normally used. The actual cipher string can take several different forms:

- It can consist of a single cipher suite such as RC4-SHA.
- It can represent a list of cipher suites containing a certain algorithm or cipher suites of a certain type. For example, SHA1 represents all ciphers suites using the digest algorithm SHA1, and SSLv3 represents all SSL v3 algorithms.
- Lists of cipher suites can be combined in a single cipher string using the + character. This is used as a logical and operation. For example, SHA1+DES represents all cipher suites containing the SHA1 and the DES algorithms.
- Each cipher string can be optionally preceded by the characters !, -, or +.
    - Using !, the ciphers are permanently deleted from the list. The ciphers deleted can never reappear in the list even if they are explicitly stated.
    - Using -, the ciphers are deleted from the list, but some or all of the ciphers can be added again by later options.
    - Using +, the ciphers are moved to the end of the list. This option doesn't add any new ciphers it just moves matching existing ones.
    - If none of these characters is present, the string is just interpreted as a list of ciphers to be appended to the current preference list. If the list includes any ciphers already present they will be ignored, that is, they will not moved to the end of the list.
    - Additionally, the cipher string @STRENGTH can be used at any point to sort the current cipher list in order of encryption algorithm key length.

Cipher Suite Names

The following list is the supported SSL cipher suites names using their OpenSSL format. Some cipher suite names do not include the authentication. In these cases, RSA authentication is used.

The list is in alphabetical order and does reflect any priority ordering.

AECDH-RC4-SHA
AES128-GCM-SHA256
AES128-SHA
AES128-SHA256

AES256-GCM-SHA384
AES256-SHA
AES256-SHA256
DES-CBC3-SHA
EXP-DES-CBC-SHA
EXP-RC4-MD5
EXP-RC4-MD5
NULL-MD5
NULL-SHA
NULL-SHA256
RC4-MD5
RC4-MD5
RC4-SHA

Default Setting

HIGH:!ADH:!SSLv2:!PSK:!ECDH:!kEDH:+AES:+3DES

This default means the system uses the cipher suites that are considered to be high security, but not those that authenticate with Diffie–Hellman, not SSLv2 protocol, not preshared keys, not elliptical curve Diffie–Hellman, not ephemeral Diffie–Hellman key exchange. And of what remains, move those that have AES in them to the end of the list, then move 3DES ciphers to the end. Therefore, the system uses AES ciphers, if possible, then 3DES.

Command Mode

Config

Syntax

`cipher-list openssl <list>`
Specifies the cipher list in openssl string format

`no cipher-list`
Removes the configured cipher list

Parameter	Type	Description
list	String	Colon-separated list of cipher strings or cipher suite names

protocol-disable-list

Use

Use to set which SSL protocols are disabled.

To provide maximum security against known security vulnerabilities, SSLv2 and SSLv3 are disabled by default.

LineRate supports the following protocols: SSLv2, SSLv3, and TLSv1.

Default Setting

SSLv2:SSLv3

Command Mode

Config

Syntax

```
protocol-disable-list <protocols>
```

Set the list of disabled SSL protocols

```
protocol-disable-list lros-default
```

Set the list of disabled protocols to the default list

Parameter	Type	Description
protocols	String	Colon-separated list of SSL protocols to disable. For example: SSLv2:SSLv3:TLSv1.

session cache

Set session cache memory size.

Use

Use to improve performance for SSL connections by storing session keys. The system saves and can reuse an SSL session key for client connections. The cached session key lets the system establish the SSL connection to the same client, bypassing the normal SSL handshake process.

This applies to termination SSL only. You set the amount of system RAM you want to use to store session keys. To determine the space needed to use session keys effectively, consider the following:

- The more active clients you have, the more space you need for the keys, which are generally about 2 KiB.
- To be beneficial, the cache must be large enough to store session keys between connections from the same client.

- Consider how many connections per second you expect and how much time you expect between the same client connecting multiple times.

In a failover situation, where the primary system fails over to a secondary system, the session cache is not copied to the secondary system.

You can use both session cache and session tickets at the same time. If the client supports session tickets, the system uses the session ticket. If the client does not support session tickets, the system uses session cache.

Default Setting

auto, which is 10 MiB

Command Mode

config

Syntax

`[no] session cache <size>`
Configure session cache in bytes

`[no] session cache <size> kibibytes`
Specified cache size is in kibibytes (system multiples the <size> x 1024)

`[no] session cache <size> mibibytes`
Specified cache size is in mibibytes (system multiples the <size> x 1024 x 1024)

`session cache auto`
Enable session caching with automatic cache size

`session cache disable`
Disable session caching

`no session cache`
Removes the configured session cache

Parameter	Type	Description
size	Integer	Set session caching to the specified size in bytes. Minimum cache size is 20480 bytes.

session tickets

Enable or disable SSL session tickets.

Use

Use to improve performance for SSL connections by creating session tickets that the client stores. If the client's browser and operating system support session tickets, the system uses a valid session ticket to establish the SSL connection to the same client, bypassing the normal SSL handshake process.

This applies to termination SSL only.

You can use both session cache and session tickets at the same time. If the client supports session tickets, the system uses the session ticket. If the client does not support session tickets, the system uses session cache.

In a failover situation, where the primary system fails over to a secondary system, the session tickets still work, because the client stores them.

Default Setting

Disabled

Command Mode

Config

Syntax

`session tickets disable`
Disable session tickets

`session tickets enable`
Enable session tickets

`no session tickets`
Removes the configured session tickets

base

Inherit parameters from an SSL profile base.

Use

A "base" in LineRate is a type of template that allows you to reuse common portions of configuration across multiple objects. Each base can inherit from another base, overriding properties from that base. This lets you create basic configurations that you can reuse and build upon.

See Working with Bases.

Use the no command to remove the specified base from the SSL profile. The SSL profile will then use the parameters configured configured for any other base it inherits from, specifically for the SSL profile, or the default settings for any parameter that is not configured.

See the Related Commands for the commands you can use when configuring an SSL profile base.

Default Setting

None

Command Mode

Config

Syntax

`base <name>`
Inherit parameters from an SSL profile base

Parameter	Type	Description
name	Word	Name of an SSL profile base to inherit from

Related Commands

ssl_profile

attach

cipher-list

session cache

session tickets

System Mode Commands

system kernel-exception

Use

Use to configure what the system does with error information if the system kernel has an unrecoverable error. The default is text only.

Configure for full output only when directed by technical support personnel. The full output creates a file the same size as the system memory and significantly slows the system restart.

You can enable both full and text.

Default Setting

None

Command Mode

config

Syntax

`system kernel-exception dump full`

Dump entire contents of memory, which will take a long time. Configure for full output only when directed by technical support personnel.

`system kernel-exception dump text`

Dump brief textual information representing system state.

Related

REST API Reference - exception

TCP Mode Commands

tcp-options-group

Use

Use to create or modify a TCP options group. Create a TCP options group only when you want to enable the Nagle algorithm. The Nagle algorigthm should be left disabled for most network applications.

Default Setting

None

Command Mode

config

Syntax

```
[no] tcp-options-group <name>
```

Create or modify a TCP options group

Parameter	Type	Description
name	Word	Name of the TCP options group.

Related

REST API Reference - tcpOptions

nagle-delay

Use

Use to enable the Nagle algorithm for TCP sessions. Collects multiple small data writes together to reduce the total number of TCP packets sent on the network. This option is disabled by default and should be left disabled for most network applications.

Default Setting

Disabled

Command Mode

tcpOptions

Syntax

`[no] nagle-delay`

Enables the Nagle algorithm for TCP sessions. Collects multiple small data writes together to reduce the total number of TCP packets sent on the network. This option should be left disabled for most network applications.

User Name Mode Commands

username

Configure local user authentication

Use

Use to configure users who need access to the LineRate system. All users have access to all functions. The system tracks user logins and logouts and a few key actions, such as reload.

If you include characters other than letters (upper or lower), digits, dashes, and periods in a password, you must put double quotes around it.

If you do not specify a user ID, the system automatically assigns a unique user ID, starting with 2000.

Normally, you should use the clear form of the password, and the system encrypts the password. You can see the encrypted password the following ways:

- CLI command—**`show run`**
- REST API—/config/users/<user_name>/password/hash5

Use the encrypted form of the password only when you need to use an existing password on another system. You can copy the encrypted password from the **`show run`** output or from the /status/config/ running node, then paste it into the encrypted form of the command on another system.

Default Setting

admin/changme

Command Mode

configure

Syntax

`[no] username <username> secret clear <password> [uid <userid>]`

Specifies the user name and an unencrypted secret; optionally specifies the user ID

`[no] username <username> secret encrypted <password> [uid <userid>]`

Specifies the user name and an encrypted secret; optionally specifies the user ID

Parameter	Type	Description
password	String	The unencrypted (cleartext) user account secret
userid	Integer	Numeric user ID
username	Word	User name. Maximum of 16 characters.

Related

REST API Reference - users

Virtual IP Mode Commands

Use the following commands to configure virtual IPs.

virtual-ip

Create or modify a virtual IP for reverse proxy (load balancing) or forward proxy.

Use

For either a load balancing or forward proxy use case, the system requires at least one virtual IP. The virtual IP is a configuration object that represents the interface that clients connect to. You can create as many virtual IPs as you need. For an overview of how virtual IPs are used in a load balancing use case, see LineRate Overview.

We recommend giving each virtual IP a meaningful name that helps identify the virtual IP. For example, you might use the application or service type (such as serving similar web content) or security settings (such as SSL) in the name.

Use to set the IP address or IP address range and port for the virtual IP. This designates the IP addresses that the system will accept traffic for.

Note: The virtual IP alone will not enable ARP on the interface. For the interface to respond to ARP on an IP address, that IP address must be configured on the system interface. Most reverse proxy

configurations should therefore configure the same IP address both on the virtual IP and the system interface.

You can set either a specific IP address and port or a range of IP addresses for a specific port. The range includes both addresses you specify as the range start and end. A range cannot overlap any other range on the system for the same port.

If a virtual IP has a specific IP assigned to it that falls within the range of another virtual IP, the system sends all traffic to the virtual IP with the specific IP address.

 When attaching a virtual IP to a forward proxy, the virtual IP must not include any of the system's own IP addresses. For a virtual IP with a single IP address, do not set the virtual IP's IP address to one of the system's own IP addresses. For a virtual IP with a range of addresses, you must ensure that the IP address range does not contain any of the system's own IP addresses. This may mean you need to break the virtual IP into multiple virtual IPs. See Configuring a range for a virtual IP with forward proxy for more detail and an example.

The system handles routed virtual IPs. Even if you set a large range of IP addresses for a virtual IP, the system only sends an ARP reply if an IP address in the range is configured on an interface. However, the system will accept traffic for any IP address in the range.

Command Mode

config

Syntax

`virtual-ip <name>`

Create or modify a virtual IP for load balancing

`virtual-ip <name> ip <addr> <port>`

IPv4 or IPv6 address of interface for client access

`virtual-ip <name> ip <addr> <port> base <baseName>`

Set the base that the virtual IP will inherit from

`[no] virtual-ip <name> ip range <startaddr> <endaddr> <port>`

Set a range of IPv4 or IPv6 addresses for client access

`[no] virtual-ip <name> ip range <startaddr> <endaddr> <port> base <base_name>`

Set a range of IPv4 or IPv6 addresses for client access and set the base that the virtual IP will inherit from

`virtual-ip base <name>`

Create or modify a virtual IP base for virtual IPs to inherit

`no base`

Remove the configured base

Parameter	Type	Description
addr	IPAddr	IPv4 or IPv6 address for interface configured for client access
baseName	Word	Name of base that the virtual IP will inherit from
endaddr	IPv4Addr	Ending IPv4 or IPv6 address for interface configured for client access
name	Word	Name of the virtual IP
port	Integer	Port number to connect to on the real server
startaddr	IPv4Addr	Starting IPv4 or IPv6 address for interface configured for client access

Related

REST API Reference - virtualIP

admin-status

Bring an object, such as a health monitor, real server, or virtual IP, online or offline. After you create an object, you must bring it online.

Use

You typically set the offline status only when you want to disable the object or block connections to the web server during maintenance or system reconfiguration.

Default Setting

offline

Command Mode

config

Syntax

admin-status offline

Bring the object offline

admin-status online

Bring the object online

no admin-status

Remove the directly configured admin status

attach

Attach an object.

Use

Use to attach an object, such as an IP filter, SSL profile, or TCP options group , to the current virtual IP.

Default Setting

None

Command Mode

config

Syntax

[no] attach ip-filter <name>

Attach or replace the filter used to decide if hosts may connect to this virtual IP.

[no] attach ssl profile <name>

Attach or replace the SSL profile for the current virtual IP.

[no] attach tcp-options-group <name>

Attach or replace the TCP options group for connections to the current virtual IP.

Parameter	Type	Description
name	Word	Name of the IP filter, SSL profile, or TCP options group to attach. See Configuring SSL.

incoming-queue-length

Length of the queue for established incoming connections.

Use

How many simultaneous established connections that the system can process internally at any given time. A shorter queue length helps prevent too many connections from starting and exhausting resources. Determine based on the speed of your processor and the number of connections you expect per second.

Default Setting

0, which is infinite

Command Mode

config

Syntax

`incoming-queue-length <max>`

Length of the queue for established incoming connections.

`no incoming-queue-length`

Remove the directly configured incoming queue length.

Parameter	Type	Description
max	Integer	Number of items allowed in the queue

ip address

Use to set the IP address or IP address range and port for the virtual IP. This designates the IP addresses that the system will accept traffic for.

Note: The virtual IP alone will not enable ARP on the interface. For the interface to respond to ARP on an IP address, that IP address must be configured on the system interface. Most reverse proxy configurations should therefore configure the same IP address both on the virtual IP and the system interface.

You can set either a specific IP address and port or a range of IP addresses for a specific port. The range includes both addresses you specify as the range start and end. A range cannot overlap any other range on the system for the same port.

If a virtual IP has a specific IP assigned to it that falls within the range of another virtual IP, the system sends all traffic to the virtual IP with the specific IP address.

When attaching a virtual IP to a forward proxy, the virtual IP must not include any of the system's own IP addresses. For a virtual IP with a single IP address, do not set the virtual IP's IP address to one of the system's own IP addresses. For a virtual IP with a range of addresses, you must ensure that the IP address range does not contain any of the system's own IP addresses. This may mean you need to break the virtual IP into multiple virtual IPs. See Configuring a range for a virtual IP with forward proxy for more detail and an example.

The system handles routed virtual IPs. Even if you set a large range of IP addresses for a virtual IP, the system only sends an ARP reply if an IP address in the range is configured on an interface. However, the system will accept traffic for any IP address in the range.

LineRate supports both IPv4 and IPv6. You can specify the IP address and subnet mask in any of the following formats:

- **`192.0.2.1/24`**— example of an IPv4 address with a 24-bit subnet mask using CIDR notation.
- **`192.0.2.1 255.255.255.0`**—equivalent to above using net mask notation.
- **`2001:DB8::/64`**—example of an IPv6 address with a 64 bit subnet mask using CIDR notation.

If you need more information about IP addresses and subnet masks, see these sites for more information:

- http://en.wikipedia.org/wiki/IP_address
- http://en.wikipedia.org/wiki/CIDR_notation

Default Setting

None

Command Mode

config

Syntax

`[no] ip address <addr> <port>`

IPv4 or IPv6 address for the virtual IP

`[no] ip range <start_addr_v4> <end_addr_v4> <port>`

IPv4 address ending the range (inclusive)

`[no] ip range <start_addr_v6> <end_addr_v6> <port>`

IPv6 address ending the range (inclusive)

`no ip`

Remove the configured IPv4 or IPv6 settings

Parameter	Type	Description
addr	IPv4Addr	Virtual IPv4 address
end_addr_v4	IPv4Addr	IPv4 address ending the range (inclusive)
end_addr_v6	IPv6Addr	IPv6 address ending the range (inclusive)
port	Integer	TCP port number on this virtual IP address
start_addr_v4	IPv4Addr	IPv4 address starting the range (inclusive)
start_addr_v6	IPv6Addr	IPv6 address starting the range (inclusive)

max-embryonic-conns

Maximum number of embryonic TCP connections to allow at one time.

Use

Maximum number of partially established (half-open) connections to allow at one time on the virtual IP. Fewer embryonic connections helps prevent too many connections from starting and exhausting resources. Determine based on the speed of your processor and the number of connections you expect per second.

Default Setting

0 (disabled)

Command Mode

config

Syntax

`max-embryonic-conns <max>`

Maximum number of embryonic connections to allow at one time on the virtual IP

`no max-embryonic-conns`

Remove the directly configured max embryonic conns

Parameter	Type	Description
max	Integer	Number of embryonic connections allowed

service http

The nodes below this one configure settings for the HTTP service type.

Set the service type for the virtual IP in the serviceType node.

Command Mode

config

Default Setting

Service default is http.

Syntax

keepalive-timeout

Close the client TCP connection if there is no active HTTP request or response in this amount of time.

Use

HTTP allows for a client to open a TCP connection to a server and send more than one HTTP request through the connection. Many HTTP clients will allow an idle TCP connection to remain open for long periods of time so that it may be used for a future HTTP request. Configuring keepalive-timeout directs the system to close TCP connections after they have been idle (i.e. no active HTTP request or response) for the configured amount of time, thus reclaiming system resources.

 Correctly configuring this setting is very important to prevent the system from running out of connection resources.

The value for this setting should be set relative to the expected maximum number of connections per second that this virtual IP will handle and the maximum number of simultaneous open connections you would like to maintain on this virtual IP. A good estimate for this value can be obtained with the following formula:

keepalive timeout = max simultaneous conns / (max expected conns per second)

Default Setting

10 seconds

Command Mode

config

Syntax

keepalive-timeout <timeout>

Closes the TCP connection if there is no HTTP activity in this amount of time

no keepalive-timeout

Remove the directly configured keepalive timeout

Parameter	Type	Description
timeout	Real	Number of seconds (can be fractional)

max-in-flight

Maximum number of requests that can be pending on one HTTP connection at a time.

Use

The default is 1, which lets the virtual IP send only one request at a time from a specific client. The virtual IP waits for the real server to send the response before sending the next request from the client. The default works in many configurations, especially when the web server has low latency.

If you set this to more than 1, the virtual IP can send the specified number of requests to the real server from a specific client, without waiting for the response to each request. Use a higher number when the web server has high latency. This should not exceed the pipeline depth setting of the web server.

Default Setting

1

Command Mode

config

Syntax

`max-in-flight <max>`

Number of outstanding requests that can be in process at one time

`no max-in-flight`

Remove the directly configured max in flight

Parameter	Type	Description
max	Integer	Number of requests

max-request-header-size

Maximum number of request header bytes to process.

Use

If the header size exceeds this value, the virtual IP closes the connection. This secures the system's memory.

If your application or environment supports a large number of cookies, you may need to increase this value. A proxy server also can add cookies to the header.

Default Setting

32 KB

Command Mode

config

Syntax

`max-request-header-size <max>`

Maximum number of request header bytes to process

`max-request-header-size unlimited`

Indicates unlimited number of bytes

`no max-request-header-size`

Remove the directly configured max request header size

Parameter	Type	Description
max	Integer	Number of bytes

service tcp

This node is for future use.

Set the service type for the virtual IP in the serviceType node.

Default Setting

None

Command Mode

config

Syntax

```
service tcp
```

base

Inherit parameters from the specified base.

Use

A "base" in LineRate is a type of template that allows you to reuse common portions of configuration across multiple objects. Each base can inherit from another base, overriding properties from that base. This lets you create basic configurations that you can reuse and build upon.

See Working with Bases.

Use the no command to remove the specified base from the virtual IP. The virtual IP will then use the parameters configured for any other base it inherits from, specifically for the virtual IP, or the default settings for any parameter that is not configured.

See the Related Commands for the commands you can use when configuring a virtual IP base.

Default Setting

None

Command Mode

config

Syntax

```
[no] base <name>
```

Parameter	Type	Description
name	Word	Name of virtual IP base to inherit from

Related Commands

admin-status

attach

incoming-queue-length

max-embryonic-conns

service http

service tcp

REST API Reference - virtualIPBase

Virtual Server Mode Commands

Use the following commands to configure virtual servers. For an example virtual server configuration, see Configuring a Virtual Server.

virtual-server

Configure a virtual server.

Use

Each load balancing (reverse proxy) configuration requires at least one virtual server. The virtual server is a configuration object that acts as a reverse proxy and ties together one or more virtual IPs and real servers. You also set the load balancing algorigthm on the virtual server.

We recommend giving each virtual server a meaningful name that helps identify the server use. For example, you might name a virtual server based on the application and the resources that the virtual server is load balancing traffic to (real servers).

Default Setting

None

Command Mode

config

Syntax

`virtual-server <name>`
Create or modify a virtual server for load balancing

Parameter	Type	Description
name	Word	Name of the virtual server

Related

REST API Reference - virtualServer

attach

Attach an object to the virtual-server.

Use

Use to attach a virtual IP or real server to this virtual server. You normally attach one virtual IP to a virtual server and attach many real servers to a virtual server.

Default Setting

None

Command Mode

config

Syntax

`attach real-server <name>`

Name of a real server to attach to this virtual server

`attach real-server <name> weight <wt>`

Specify a weight for this real server. The weight affects how the system load balances when using the load balancing algorithm weighted round robin. The weight assigned to a specific real server overrides the weight assigned to a group the real server is a member of.

`attach real-server group <group_name>`

Name of a real server group whose members will be attached to this virtual server

`attach real-server group <group_name> weight <wt>`

Specify a weight for all real servers in this group. The weight affects how the system load balances real servers in the group when using the load balancing algorithm weighted round robin.

`attach virtual-ip <vip> default`

For service HTTP. Sets the virtual IP as the default for the virtual server and directs the incoming HTTP requests with a hostname that does not match the configured list to this virtual server. Use when you have the same virtual IP attached to more than one virtual server, and you use hostnames with the virtual servers.

Parameter	Type	Description
name	Word	Name of the real server to attach
vip	Word	Name of a virtual IP to attach
wt	Real	Weight for this real server. For example, if you assign a weight of 1 to one real server and 3 to another real server, the load balancer sends three times as many connections to the real server with the weight of 3.

description

Create a description for the virtual server.

Use

Use to create a description of the virtual server use and any other information.

Default Setting

None

Command Mode

config

Syntax

`description <desc>`

Description for the virtual server

`no description`

Remove the description

Parameter	Type	Description
desc	String	Description string. Enter a maximum of 255 characters.

ip

Configure DSCP settings.

Use

Use to configure how LineRate populates the Differentiated Services Code Point (DSCP) bits as packets pass through to provide quality of service for IP networks. Use the options to configure where the bits are going to or the specific value to use. For the valid values and more information about DSCP, see http://en.wikipedia.org/wiki/Differentiated_Services_Code_Point.

When you enable the copy option, any value that was previously set is changed to 0. Conversely, when you set a value, the copy option is changed to 0, disabling the copy.

Default Setting

None

Command Mode

config-vserver

Syntax

`[no] ip dscp client <value>`

Configure IP packet DSCP settings for flows to the client

`[no] ip dscp client copy`

Copy IP packet DSCP bits from the server to the client

`[no] ip dscp server <value>`

Configure IP packet DSCP settings for flows to the server

`[no] ip dscp server copy`

Copy IP packet DSCP bits from the client to the server

Parameter	Type	Description
value	Integer	Set IP packet DSCP value for flows to the client or server.

lb-algorithm

Assign the load balancing algorithm.

Use

Use to set the algorithm the load balancer uses to balance requests through the virtual server. Available algorithms are:

- Round robin—Strict round robin load balancing algorithm, regardless of the connections in queue for each server.
- URL hash—Sends the same URL to the same web server each time, based on the system-generated hash. An example use case for this load balancing algorithm is to permit the web server to use cached data effectively. For most situations, the URL hash process works well. If you find that a specific URL is overloading the real servers because of a very high number of requests, use one of the following to set the rehash to a higher value:
  - CLI—Use the rehash option shown in the examples below.
  - REST API—Use the urlHashRehash node.
- Round robin free—Round robin algorithm that only sends requests to those servers that are not already at maximum connections (ones with free connections).
- Weighted round robin—Weighted round robin algorithm that only sends requests to those servers that are not already at maximum connections (ones with free connections). The weighting creates a preference, in the form of a ratio, for the web servers with higher weights. Use the weights to rank web servers based on load capacity or speed. For example, if you assign a weight of 1 to one real server and 3 to another real server, the load balancer sends three times as many connections to the real server with the weight of 3.
- Weighted least connections—Sends requests to the server that has the fewest number of connections, which helps to even out the connections across servers. The weighting creates a preference, in the form of a ratio, for the web servers with higher weights. Use the weights to rank web servers based on load capacity or speed. For example, if you assign a weight of 1 to one real server and 3 to another real server, the load balancer sends three times as many connections to the real server with the weight of 3. The default weight is 1.

Default Setting

round-robin-free

Command Mode

config

Syntax

[no] lb-algorithm round-robin

Set to round robin algorithm.

[no] lb-algorithm url-hash

Set to URL hash algorithm.

[no] lb-algorithm round-robin-free

Set to round robin free algorithm.

[no] lb-algorithm weighted-round-robin-free

Set to weighted round robin algorithm.

[no] lb-algorithm weighted-least-conns

Set to weighted least connections algorithm that sends requests to those servers with the fewest open connections.

[no] lb-algorithm url-hash rehash <rehash>

Set to URL hash rehash algorithm.

no lb-algorithm

Remove the configured algorithm.

Parameter	Type	Description
rehash	Real	The default for the URL hash algorithm is to automatically tune the rehash parameter and to attempt to use more than one real server if the primary real server becomes overloaded. A user-configured value of 0 turns rehashing off so that the system will never direct the same URL to more

		than one real server. When using URL hash, increase only if a specific web server is overloaded because of a very high number of requests.

service http

Enable HTTP service.

Use

Sets the current virtual server to be compatible with layer 7 load balancing, for web use. Use this for web servers that are using HTTP. The service setting on a virtual server must match the service setting on any real server attached to the virtual server. Use for HTTPS as well.

Default Setting

Service default is http.

Command Mode

config

Syntax

```
service http
```

Enables HTTP service

client-ip-header

Header name for all HTTP requests.

Use

Use to label the name of the header where the system places the client's IP address.

Default Setting

None

Command Mode

config

Syntax

client-ip-header <header_name>

Text for header that the proxy will add to all HTTP requests

no client-ip-header

Remove the configured client IP header

Parameter	Type	Description
header_name	String	String to name the request header that contains the client IP address.

forward-connect-requests

Indicate if this virtual server should forward HTTP connect requests.

Use

If you expect to see connect requests to your web servers, enable this to permit forwarding of connect requests. If you do not expect to see connect requests to your web servers, disable this to block the connect requests at the load balancer.

Default Setting

Disabled

Command Mode

config

Syntax

forward-connect-requests

Permit this virtual server to forward HTTP connect requests

hostname

Assigns a host name that this virtual server will respond to.

Use

Use when you have the same virtual IP attached to more than one virtual server, and you use host names to direct requests to a specific virtual server. Assign the host name of the web server that you want the virtual server to respond to.

Default Setting

None

Command Mode

config

Syntax

`hostname <name>`

A hostname that this virtual server will respond to

Parameter	Type	Description
name	Word	Host name string (example: www.example.com)

persist

Use

Use to enable session cookie persistence, which lets the system send subsequent requests from a client to the same server to improve performance when the servers cache content or for applications which require requests from a client to be consistently sent to the same server.

Give the cookie a name that will be unique, to differentiate it from any cookies the servers may use.

Default Setting

0 (Disabled)

Command Mode

config-vserver-http

Syntax

`[no] persist cookie <name>`

Enable session persistence by setting HTTP cookies

`[no] persist cookie <name> expires <lifetime>`

Set how long the cookie persists, in seconds, from the time of last client request. Use 0 to set no expiration, but the cookie still expires when the session ends.

`no persist cookie`

Remove the configured persist cookie

Parameter	Type	Description
lifetime	Integer	Set how long the cookie persists, in seconds, from the time of last client request. Use 0 to set no expiration, but the cookie still expires when the session ends.
name	Word	Name of cookie to use for load balancing (for example, route).

request-rate-limit

Use

Use to set the number of HTTP requests per second to send to the virtual server. Consider your traffic patterns and the capabilities of all of the servers this virtual server handles.

Use the burst size to set the maximum number of HTTP requests the servers can handle in a burst above the rate limit. This value may be no less than 2% of the rate limit. By default, the burst size is the same as the request rate limit you configure. For most situations, this works well. Change this setting only if your requests tend to come in bursts.

Default Setting

0 (no limit)

Command Mode

config-vserver-http

Syntax

`[no] request-rate-limit <limit>`

Set the virtual server request rate limit

`[no] request-rate-limit <limit> burst-size <burst>`

Set the maximum burst size (default is the rate limit)

Parameter	Type	Description
burst	Integer	Maximum number of requests handled in a burst above the rate limit. This value may be no less than 2% of the rate limit.
limit	Integer	Maximum number of requests per second.

strip-request-header

Delete all HTTP request headers with the specified name.

Use

Use to remove specific request headers, perhaps when headers, like cookies, are very large. You can also use this to strip the byte-range request header, which forces the web server to respond with the entire content rather than only those bytes requested by the byte-range header. This is especially useful if the web servers are proxies that perform layer 7 inspection.

Default Setting

None

Command Mode

config

Syntax

`strip-request-header <header_name>`

Delete all HTTP request headers with the specified name

Parameter	Type	Description
header_name	String	Name of header to remove.

tcp-multiplex

Enables TCP multiplexing for this virtual server.

Use

Enables multiplexing of HTTP requests processed by this virtual server to its associated web servers on one TCP connection. When enabled, the system permits multiple requests on a single TCP connection, resulting in better performance. When disabled, the system sends only one request, then closes the connection, and opens a new connection for the next request, resulting in more overhead.

For most situations, enabling TCP multiplexing improves performance. Make sure your web servers support handling multiple requests on a single TCP connection.

Default Setting

Disabled

Command Mode

config

Syntax

`tcp-multiplex`

Enables TCP multiplexing for this virtual server

service tcp

Enable TCP service.

Use

Sets the current virtual IP to handle TCP connections. Use this service type for servers that are not using HTTP, for example, an email server.

Default Setting

Service default is http.

Command Mode

config

Syntax

service tcp

Enables TCP service

Exec Commands

This page was not added to the PDF due to the following tag(s): article:topic-guide

Backup Mode Commands

backup

Back up the current configuration or system version.

Use

Use to create one of the following backup types:

- Quick—Backs up your current configuration (all files in /home/linerate, except the backups directory).
- Full—Backs up all files for the current version of LineRate, including all configuration and files. When you create a full backup, you can use it to do either a quick or full restore.

 Backups are specific to a version. You cannot use a version 1.5 backup to do a restore on a version 1.4 system.

You can save backups locally or to another server on your network.

We recommend naming backups with the system version, date, time, and backup type (quick or full). For local backups, the system automatically adds .tar.bz2 as the file extension. For backups to another server, you should add the file extension of either .tbz or .tar.bz2 to your file name.

To see existing local backups, use either of the following:

- CLI command—**`show backup list`**
- REST node—/status/system/util/backup/list or /status/system/util/backup/list-detail

Default Setting

None

Command Mode

exec

Syntax

`backup <uri>`

Does a quick backup of the current configuration

backup <uri> delete

Deletes the specified backup

backup <uri> full

Does a full backup of the current version of the system

Parameter	Type	Description
uri	String	URI to put the backup. If it does not end with '.tbz' or .tar.bz2', the file extension is appended (for local files only). The system supports file:// and scp://. If a protocol prefix is not given, the system saves the file locally in /home/linerate/backups. Use quotes around URIs (for example, "file:///home/linerate/backups/orig-config.tar.bz2" or just "orig-config"). We recommend naming backups with the LineRate version, date, and time.

Examples

backup "ver1.5may11-2012-13:00"

Backs up all files in /home/linerate/ to the /home/linerate/backups directory and adds the .tar.bz2 file extension to the file.

backup "file:///home/linerate/backups/ver1.5...-13:00.tar.bz2"

This does exactly the same thing as the command above.

backup "scp://<server>/<directory>/ver1.5may11-2012-13:00.tar.bz2"

Uses secure copy to back up your configuration to the specified server and directory.

backup "scp://<server>/<directory>/ver1.5may11-2012-13:00.tar.bz2" full

Uses secure copy to back up all of version 1.5 to the specified server and directory.

Related Commands

Restore Mode Commands

REST API Reference - backup

Bash Mode Commands

bash

Open a bash shell in interactive mode

Use

Use to access the bash shell to perform functions directly.

 **Caution:** Take great care when using bash. Do not use bash to configure interfaces, network settings, /etc files, or any other configuration, because LineRate will not be aware of the configuration changes. **Do not make changes to the system configuration using bash unless technical support personnel instruct you to.**

Following are some common and safe uses of bash:

- Copy files from or to the system using scp.
- Use the **`lros_shell --config <path_to_config_file_name>`** command to reuse a configuration file from another system to configure this system. You must copy the configuration file to the system before using the command.
- Find the names of files on the system.
- Using the **`tail`** command with log files.

Default Setting

None

Command Mode

exec

Syntax

`bash`

Open a bash shell in interactive mode

bash "<args>"

Execute a single bash command and return to the LROS shell prompt.

Parameter	Type	Description
args	String	Bash shell command to execute

Interactive Mode Example

```
example_host# bash
  *******WARNING*******
The bash prompt allows unrestricted access to the system.
It is possible to configure the system in ways that cannot
be shown in the UI, that may lead to incorrect operation
of the system, and that may or may not be persistent after a
reload of the system.  F5 recommends that no
configuration be made from bash unless directed by support.
  *********************

[admin@example_host ~]$ lros_shell --config <config_file_name>
```

Single Command Mode Example

```
example_host# bash "sudo tail -f /var/log/controller.messages"
```

Using SCP

When you upgrade your system, you must download the upgrade file, then install it. One way to do the installation is copy the upgrade file to the LineRate system using scp before performing the upgrade.

 **Caution:** Make sure you do not overwrite existing system files using scp.

Syntax

scp <local_image.upg.gz> admin@<LineRate Proxy_ip>:/home/linerate/.

Use this command from the system on your network where you downloaded the upgrade file to copy the upgrade file to the LineRate system.

```
LROS# bash

scp <remote_username>@<remote_system_ip>:<path_to_image> .
```

exit

Use these commands from the LineRate system to copy the upgrade file from the system on your network where you downloaded it to the LineRate system. This copies the file to the /home/linerate directory.

Clear Mode Commands

clear

Use

Use to remove routes from the routing table and reinstall the static routes you have configured in LineRate. The clear process removes any routes configured using bash or another process outside of the LineRate console.

Default Setting

None

Command Mode

exec

Syntax

`clear ip route *`

Clears the IPv4 routing table and reinstalls configured static routes

`clear ipv6 route *`

Clears the IPv6 routing table and reinstalls configured static routes

Related Commands

ip route

ipv6 route

REST API Reference - clear

Configure Command

Use

Use to go into configuration mode to change settings for LineRate software. Configure has many submodes that you may enter when using a command in config mode. The prompt changes to reflect the config submode.

For example, when you type:

configure

the prompt changes to:

```
LROS(config)#
```

When you type:

real-server rs1

the prompt changes to:

```
LROS(config-rserver:rs1)#
```

Default Setting

None

Command Mode

exec

Syntax

`configure`

Enter configuration mode.

`configure terminal`

This form of the command does the same thing and is available for compatibility if you are accustomed to other network operating systems.

Related Commands

Using Command Line Modes

Copy Mode Commands

copy

Copy a file.

Use

Use to copy a file specifying the source and destination file names. You may want to use this to copy the current running configuration or startup configuration file. See Running Config and Startup Config.

 Take care when copying to the startup-config file, because the source file overwrites the contents of the start-up config file. The system uses the start-up config file when it loads, and improper configuration could cause instability or cause the system not to load.

You can also use this command to copy a snippet of configuraion that you have in a file to the running config. The copy command applies the commands, as if you typed them in, and adds them to the running config. Copying to the running config does not overwrite the contents of the running config.

Default Setting

None

Command Mode

exec

Syntax

`copy <src> <dst>`

Copy the source to the destination file

`copy <src> running-config`

Add the source file configuration to the current operating (unsaved) configuration; this does not ovewrite the running config file

`copy <src> startup-config`

 Overwrite the startup config file with the contents of the source file.

`copy running-config <dst>`

Copy the current (unsaved) running config to the destination file

`copy running-config startup-config`

Copy the current (unsaved) configuration to the startup-config file (same as using **`write`**)

`copy startup-config <dst>`

Copy the current (saved) startup config to the destination file

`copy startup-config running-config`

Add the current (saved) startup config to the running config file; this does not overwrite the running config file. This lets you return to the saved configuration.

Parameter	Type	Description
dst	String	Destination file name
src	String	Source file name

Related Commands

Write Command

REST API Reference - copy

Debug Mode Commands

debug

Enables debugging output

Use

Use only when directed by technical support personnel.

 Enabling this function may have a significant impact on system performance. Use only with extreme caution on a production system.

Default Setting

None

Command Mode

exec

Syntax

`[no] debug controller cpu-profile`

Enables CPU profiling for the system management controller

`[no] debug proxy cpu-profile`

Enables CPU profiling for the proxy

`[no] debug proxy heap-profile`

Enables heap profiling for the proxy

`[no] debug proxy trace`

Enables detailed session tracing for the proxy

`[no] debug uiprotocol messages`

Enables debugging output for UI Protocol messages

Related Commands

REST API Reference - debug

js-heap-profile

Use
Scripts may have bugs that cause objects to be retained longer than needed. These retained objects use memory on the system, which can affect performance. You can inspect the heap objects that are retained by a script to determine if all of the retained objects should be retained, or if a bug has caused some objects to be retained too long.
One tool to inspect the Javascript heap is the Google Chrome heap profiler, which is available in any Google Chrome browser.

For more information about the Chrome Heap Profiler, refer to:

https://developers.google.com/chrome-developer-tools/docs/heap-profiling

A system with Google Chrome Developer Tools is required, this is the profiler system. The profiler system can run any operating system for which Chrome is available.

Default Setting

None

Command Mode

exec

Syntax

`[no] debug proxy js-heap-profile`

Enables JavaScript heap profiling for the proxy

js-run-gc

Use
Use to run garbage collection on the scripting engine. The scripting engine runs Javascript, which is a garbage-collected language. Scripting has a high-performance generational garbage collector, so running this command is unnecessary. You should not run this command while processing traffic, because it will halt traffic processing while it runs. It can take longer than one second to run.

Default Setting

None

Command Mode

exec

Syntax

`[no] debug proxy js-run-gc`

Run garbage collection on the scripting engine. Note: running the garbage collector will impact performance.

Exit Command (exec mode)

Use

Use to exit the LROS shell. When in exec mode (LROS# prompt) directly on the LineRate system, this returns you to the login prompt for the system. When in exec mode from SSH, this returns you to the SSH prompt.

Default Setting

None

Command Mode

exec

Syntax

```
exit
```

Halt Mode Commands

halt

Shutdown and power off the system

Use

Use to shut down the system, including the hardware.

If a backup, restore, or upgrade is in progress, we recommend waiting until these processes finish. They may take a few minutes. Using the force option risks disk corruption.

Default Setting

None

Command Mode

exec

Syntax

`halt`

Shut down and power off the system, with a confirmation prompt

`halt force`

Force a shutdown even if backup, restore, or upgrade is in progress, with a confirmation prompt. Risk of disk corruption.

`halt no-prompt`

Shut down and power off the system, but no confirmation prompt

`halt force no-prompt`

Force a shutdown even if backup, restore, or upgrade is in progress, but no confirmation prompt. Risk of disk corruption. Also valid: **`halt no-prompt force`**.

Related Commands

REST API Reference - halt

License Mode Commands

license

Install the license file.

Use

Use to install the license file on a LineRate system. To purchase a license, send an email to linerate-sales@f5.com or call 1.855.LINERATE (1.855.546.3728).

If you purchased a license for LineRate, you must provide information to identify the hardware, then you will receive an email with a download link for the license file.

You must install the license file. Currently, the only feature you can license is called base.

Licenses define the expiration date of the license and the rate limits for:

- HTTP requests per second—When limit is reached, clients receive an HTTP 503 error.
- TCP connections per second—When limit is reached, clients receive a connection reset.
- Mb per second (bandwidth)—When limit is reached, the system slows traffic. See below for information about how the system counts bandwidth.
- Burst duration—Acts as a multiplier for each licensed rate limit. The <burst_duration> x each rate limit sets the maximum number of HTTP requests, TCP connections, and Mb per second the system can handle in a burst above the rate limit. For example, if the rate limit is 100 HTTP requests per second and the burst duration is 5, the system can handle bursts up to 500 HTTP requests per second (5x the limit).

For licenses with a larger burst size, as in the free tier license, the **`show licensing`** commands report the rate limits in minutes.

Starting two days before the expiration date, a banner displays when you log in that feature license is expiring.

Understanding How the System Counts Bandwidth

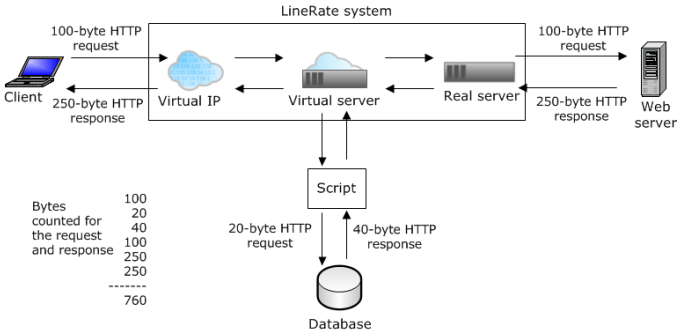
The system uses rate shaping (also called traffic shaping) to limit the bandwidth. On HTTP services, all layer 7 payload traffic is counted, but nothing at other layers. This will cause the throughput to appear to be more at the interface level than the licensed rate. For TCP services, the traffic counted is the layer 4 payload.

The bandwidth is limited at all points where data can enter or exit the data path. The points controlled are:

- HTTP virtual IP (request/response)
- HTTP real server (request/response)
- HTTP forward proxy (request/response)
- TCP virtual IP (read/write)
- TCP real server (read/write)
- TCP forward proxy (read/write)
- LineRate Scripting client.http.request
- LineRate Scripting script client.http.response
- LineRate Scripting script TCP socket (read/write)

In the example below, a 100-byte HTTP request generates a 250-byte response from the web server and consumes 760 bytes of bandwidth:

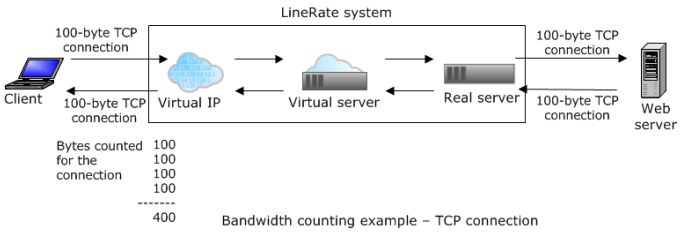
- 100 bytes from the client to the virtual server
- 20 bytes from the virtual server to the external database via a script
- 40 bytes from the external database to the virtual server via a script
- 100 bytes from the real server to the web server
- 250 bytes from web server to real server
- 250 bytes from the virtual IP to the client



Bandwidth counting example - HTTP request

In the example below, a 100-byte TCP connection generates a 100-byte connection to the web server and consumes 400 bytes of bandwidth:

- 100 bytes from the client to the virtual server
- 100 bytes from the real server to the web server
- 100 bytes from web server to real server
- 100 bytes from the virtual IP to the client



Default Setting

None

Command Mode

exec

Syntax

`license install base <uri>`

Configure licensing.

`license install force base <uri>`

Force a license install even if an active license for this feature currently exists.

Parameter	Type	Description
uri	String	URI of the JSON-formatted license file to install. System supports file:// protocol. If the protocol prefix is not given, start the URI with a / for an absolute path and start without a / for a relative path from /home/linerate. A maximum file size of 100 kB is enforced. Protect URIs with quotes (for example, "file:///home/linerate/license", or just "license")

Related

REST API Reference - license

No Command (exec mode)

Negates or removes the specified command or item.

Use

Use to turn off debug commands.

Default Setting

None

Command Mode

exec

Syntax

```
no debug <option>
```

Related Command

No Command (config mode)

NTP Mode Commands (exec)

1. ntp

ntp

Execute NTP time synchroniziation.

Use

Use to synchronize the system's clock with an NTP server immediately. The NTP server is specified by an IP address. Unlike configuring NTP using the **`config ntp server`** command, this immediate synchronization jumps the system's clock forward or backward to be synchronized, and will not correct for future drift.

Default Setting

None

Command Mode

exec

Syntax

`ntp sync <server>`

Synchronize with a specified NTP server

Parameter	Type	Description
server	IPAddr	IP address of the NTP server

Reload Mode Commands

Reload

Shut down and restart the system.

Use

Use to reload the current version or a different version of the system. If you have used the **boot** command or /config/system/boot/version node to set up a different version to reload, use the **reload** command or /exec/system/reload node when you are ready to actually reload that version.

 The **reload** command or /exec/system/reload node ends all connections and shuts down the system, then restarts the system.

If an upgrade, restore, or backup is in process, the system lets you know and does not continue with the reload.

 Use the force option only in extreme situations. Forcing a reload during an upgrade, restore, or backup may corrupt the system, and the system may not function at all.

Default Setting

None

Command Mode

exec

Syntax

reload

Shut down system and restart

reload force

Force a reload even if backup, restore, or upgrade is in progress. Risk of disk corruption.

Related Commands

Boot Mode Commands

REST API Reference - reload

Restore Mode Commands

restore

Restore configuration or full system version from backup.

Use

After using the backup command to create a backup, you can restore from the backup in one of the following ways:

- Quick—Restores your current configuration (all files in /home/linerate, except the backups directory). You can use either a quick or full backup to do this type of restore.
- Full—Restores all files for the current version of LineRate. You can only use a full backup for this type of restore. Use this type of restore if you find some system files are missing or have become corrupted.

 Backups are specific to a version. You cannot use a version 1.5 backup to do a restore on a version 1.4 system.

To see existing local backups, use either of the following:

- CLI command—**`show backup list`**
- REST node—/status/system/util/backup/list or /status/system/util/backup/list-detail

 After you do a restore, the system immediately does a reload. The reload shuts down the system, then restarts it.

Default Setting

None

Command Mode

exec

Syntax

restore <uri/>

Does a quick restore from the specificed backup

restore <uri> full</uri>

Does a full restore from the specified backup

Parameter	Type	Description
uri	String	URI of the backup to restore from. The system supports file:// and scp://. If a protocol prefix is not given, the system looks for a local file in /home/linerate/backups. Use quotes around URIs (for example, "file:///home/linerate/backups/orig-config.tar.bz2" or just "orig-config").

Examples

restore "ver1.5may11-2012-13:00"

Restores the configuration from a file in /home/linerate/backups.

restore "file:///home/linerate/backups/ver1.5may11-2012-13:00.tar.bz2"

This does exactly the same thing as the command above.

restore "scp://<server>/<directory>/ver1.5may11-2012-13:00.tar.bz2"</directory></server>

Uses secure copy to restore your configuration from the specified server, directory and file.

```
restore "scp://<server>/<directory>/ver1.5may11-2012-13:00.tar.bz2"
   full</directory></server>
```

Uses secure copy to restore all of version 1.5 to the specified server, directory, and file.

Related Commands

Backup Mode Commands

Reload Mode Commands

REST API Reference - restore

Scripting Mode Commands

scripting

Execute commands related to the scripting system.

Use

Use to install and uninstall Node Packaged Modules.

Modules installed using **npm** are available globally (to all scripts) using **require('<module-name>')**, and are stored on the system in directory: /home/linerate/data/scripting/lib/node_modules.

Install

Use to install a Node Packaged Module. The **npm** command is a package manager similar to the one provided for the Node.js platform.
The **scripting install npm** command installs a Node Packaged Module from the same default registry as the npm for Node platform, if no registry is specified.

The install command downloads and installs the specified module. The installation may take from a few seconds to a few minutes. Nothing displays during the installation. After a successful installation, the name and version of the module, and where it was installed displays, along with any warnings and any dependencies. All dependencies are also installed.

The name of the Node Packaged Module (<module_name> in the CLI and "name" key in REST) is a string and can be one of the following:

- <tarball-file>—Install from a tarball file already available on your system.

  scripting npm install /tmp/sampleModule.tgz
- <tarball-url>—Install from a remote tarball.

  scripting npm install https://devcentral.f5.com/linerate/modules/tarball/sampleModule.tar.gz
- <name>—Install the latest version of package published on npm.

  scripting npm install async
- <name@version>—Install the specified version of a package published on npm.

```
scripting npm install async@0.2.9
```

If you see errors after you attempt to install, you can find more information about the errors in the following default system logs:

- /var/logs/controller.messages—This file will have some additional information.
- /home/linerate/data/scripting/npm-debug.log—This file will have even more information and is overwritten with each npm error.

Registry

Use the registry option to specify the registry that you want to install from. If not specified, the system uses the official public npm registry: http://registry.npmjs.org. Before specifiying another registry in the install command, you must configure the registry. See npm Mode Commands.

Uninstall

Use to uninstall a Node Packaged Module that you no longer need.

Uninstalling a module leaves behind files in cache. You can use **`cache clean`** to remove those file.

Cache Clean

Uninstalling a Node Packaged Module leaves behind files in cache. To be sure that reinstalling the module uses the latest version, use **`cache clean`** to remove the cache files.

Default Setting

None

Command Mode

exec

Syntax

`scripting npm cache clean`

Clear npm cache, after uninstalling a module.

`scripting npm install <module_name>`

Install specified module.

`scripting npm install <module_name> registry <reg_name>`

Specify an npm registry to use for the installation.

`scripting npm install <module_name> registry <reg_name> timeout <sec>`

Specify an npm registry and timeout period to use for the installation.

`scripting npm uninstall <module_name>`

Uninstall the specified module.

Parameter	Type	Description
module_name	String	Name or URL of the node packaged module. Protect name or URLs with quotes ("underscore@1.4.4" or "https://github.com/documentcloud/underscore/archive/1.4.4.tar.gz").
reg_name	Word	Name of the npm registry.
sec	Integer	Number of seconds the system should wait for a response from the registry. The default is 90 seconds.

Related Commands

REST API Reference - scripting

CLI Reference - npm Mode Commands

Show Commands

This page was not added to the PDF due to the following tag(s): article:topic-guide

Show Backup Commands

show backup

Use

Use to list the the available backup files.

Default Setting

None

Command Mode

exec

Syntax

`show backup list`

Shows a list of available backups

`show backup list detailed`

Shows detail about available backups, including the backup type. This command may take a few minutes.

Example

`show backup list`

```
Available backups:
   Backup Name
   --------------------------------
   mybackup1.tar.bz2
   ver1.5may11-2012-13:00.tar.bz2
   ver1.5.0-5-22-12.tar.bz2
   ver1.5.0-5-22-12--1.tar.bz2
   ver1.5.6-8-27-12.tar.bz2
```

`show backup list detailed`

```
Available backups:
    Backup Name                       Type     Version
    ------------------------------------------------
    mybackup1.tar.bz2                 partial  N/A
    ver1.5may11-2012-13:00.tar.bz2    partial  N/A
    ver1.5.0-5-22-12.tar.bz2          partial  N/A
    ver1.5.0-5-22-12--1.tar.bz2       partial  N/A
    ver1.5.6-8-27-12.tar.bz2          full     1.5.6
```

Related

Backup Mode Commands

Restore Mode Commands

REST API Reference - backup

Show Boot Commands

1. show boot

show boot

Use

Use to view the LineRate version that will load when you use the **`reload`** command. You can change the version that will load using one of the following:

- CLI command—**`boot system <version>`**
- REST node—/config/system/boot/version

Default Setting

None

Command Mode

exec

Syntax

`show boot system`

Shows the version that will load with the next reload.

Related

Boot Mode Commands

Reload Mode Commands

REST API Reference - boot

Show CARP Commands

show carp

Use

Use to show information about your CARP settings.

Default Setting

None

Command Mode

exec

Syntax

`show carp`

Show CARP settings for all CARP groups and interfaces on the system

`show carp <vhid>`

Show CARP settings only for the specified CARP group

`show carp interface <intf>`

Show CARP settings only for the specified CARP interface

Parameter	Type	Description
intf	String	Name of an interface.
vhid	Integer	Virtual Host ID (VHID) used to identify the CARP group.

Examples

`show carp`

```
bce0 - Group 43
  State is Backup since Tue Jul 10 09:42:16 2012
  Administrative state is up
  Virtual IP address is 202.0.115.2/32
  Virtual MAC is 0000.5e00.012b
  Priority 24
  Master is 202.0.119.1 (remote), priority is 20

em1 - Group 42
  State is Init since Tue Jul 10 09:42:16 2012
  Administrative state is down
  Virtual IP address is 201.0.115.2/32
  Virtual MAC is 0000.5e00.012a
  Priority 24
  Master is Unknown, priority is Unknown

em1.2001 - Group 77
  State is Init since Tue Jul 10 09:42:16 2012
  Administrative state is down
  Virtual IP address is 101.0.115.2/32
  Virtual MAC is 0000.5e00.014d
  Priority 24
  Master is Unknown, priority is Unknown

bce0.2002 - Group 78
  State is Backup since Tue Jul 10 09:42:16 2012
  Administrative state is up
  Virtual IP address is 102.0.115.2/32
  Virtual MAC is 0000.5e00.014e
  Priority 24
  Master is 102.0.119.1 (remote), priority is 20
```

show carp 77

```
em1.2001 - Group 77
  State is Init since Tue Jul 10 09:42:16 2012
  Administrative state is down
  Virtual IP address is 101.0.115.2/32
  Virtual MAC is 0000.5e00.014d
  Priority 24
  Master is Unknown, priority is Unknown
```

show carp interface em1.2001

```
em1.2001 - Group 77
  State is Init since Tue Jul 10 09:42:16 2012
  Administrative state is down
  Virtual IP address is 101.0.115.2/32
  Virtual MAC is 0000.5e00.014d
```

```
  Priority 24
  Master is Unknown, priority is Unknown
```

Related

interface CARP

REST API Reference - carp

Show Certificate Commands

Use the following commands to display information about certificates.

show certificate

Use

Use to display information about existing certifcates and certificate bundles.

The detailed form of the commands for a specific certifcate or bundle are useful to see the valid dates, the common name (CN), and the key size.

Default Setting

None

Command Mode

exec

Syntax

`show certificate brief`

Display summary information for all installed certificates and certificate bundles

`show certificate bundle brief`

Display summary information for all installed certificate bundles

`show certificate <cert_name>`

Display the detailed information about the specified single certificate

`show certificate bundle <bundle_name>`

Display the detailed information about the specified certificate bundle

Parameter	Type	Description

bundle_name	Word	Name of the certificate bundle
cert_name	Word	Name of the certificate

Examples

```
show cert brief
Certificate                      Subject Common Name (CN)
---------------------------------------------------------------
cert_chain_secure.example.com1  Example Corp Intermediate CA
cert_prim_secure.example.com    secure.example.com
self-signed                      lros-default-host

Certificate Bundle  Subject Common Name (CN)
-------------------------------------------
cert_bndlsecure.example.com  VeriSign Class 3 Public Primary Certification Authority -
   G5
                             VeriSign Class 3 International Server CA - G3

                             VeriSign Class 3 Secure Server CA - G3

show certificate bundle brief
Certificate Bundle           Subject Common Name (CN)

-------------------------------------------------------------------------------------
   ---
cert_bndlsecure.example.com  VeriSign Class 3 Public Primary Certification Authority -
   G5
                             VeriSign Class 3 International Server CA - G3

                             VeriSign Class 3 Secure Server CA - G3

show certificate cert_chain_secure.example.com1
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 1 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, ST=Colorado, L=Louisville, O=Example Corp., CN=Example Corp. CA
        Validity
            Not Before: Feb 11 02:12:30 2012 GMT
            Not After : Feb  8 02:12:30 2022 GMT
        Subject: C=US, ST=Colorado, L=Louisville, O=Example Corp., CN=Example Corp
   Intermediate CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d9:70:81:18:70:b5:29:8e:ef:01:87:3f:d9:e3:
                    68:fe:8e:6d:39:09:fc:0f:64:99:28:59:74:3c:56:
                    bf:ad:24:f2:8b:74:0a:99:fb:3c:8f:aa:bf:95:37:
                    a0:94:b8:46:fb:6f:eb:77:aa:67:ba:4d:ce:17:b6:
                    4a:ee:e6:42:19:db:ed:32:c8:75:cf:15:0b:2d:9b:
```

```
                a5:41:0e:aa:0d:bd:c0:aa:0a:d5:b9:07:27:a9:31:
                88:c4:45:92:f7:c5:21:f3:5b:87:59:04:51:7b:1c:
                0e:8a:35:b6:c9:3b:6e:01:40:f5:08:82:ad:6a:ce:
                81:57:0a:d3:96:2f:72:25:67:29:67:ef:dc:8b:84:
                4d:9b:b2:ad:5b:3c:c5:eb:66:3c:7c:5e:42:65:e2:
                0d:21:ed:f2:2e:af:b4:a6:2f:8e:c1:f3:83:d3:2d:
                92:76:3c:d0:aa:92:fa:bb:d8:9b:87:85:ef:d8:09:
                85:2a:ad:82:d2:27:47:04:59:65:3c:7a:53:92:8f:
                89:c4:7c:4c:f3:02:0c:85:e6:5c:78:15:9b:33:f5:
                b9:f8:20:e2:25:ac:bb:ac:ca:bc:b9:14:85:62:e1:
                d6:07:8b:b4:95:bf:77:f6:b4:36:6a:52:46:4e:0a:
                31:90:63:c5:26:b4:43:32:7d:49:5f:3a:1c:58:9b:
                eb:79
            Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
        a3:b8:7c:7d:e3:87:03:29:35:10:36:38:86:a6:f3:53:a7:58:
        b8:bf:07:fb:33:fb:35:bf:eb:88:d7:78:0c:b1:02:df:71:71:
        ac:14:43:10:2d:54:08:a3:24:56:db:79:09:47:a9:2c:2d:ee:
        de:c8:50:21:e5:48:d0:e0:6e:7e:f8:e1:80:79:97:59:f8:06:
        6b:fd:98:ee:c0:ef:57:12:2f:b8:2d:52:31:97:b3:ec:e9:c2:
        6e:8a:fe:fa:9e:d9:98:a2:a3:a9:05:a8:25:52:ac:6f:bb:c1:
        04:80:1e:3a:a8:7d:3c:1e:e2:2c:c0:f0:95:4f:6e:2a:0b:e7:
        fb:e2:a7:e9:f5:a2:18:e3:f7:43:f6:1f:7c:63:2b:64:51:bd:
        81:dd:a1:b5:11:f3:16:05:da:ab:5a:67:bf:25:07:e1:f6:c7:
        08:3d:42:69:4b:b6:8c:99:60:c7:eb:ce:46:63:0d:49:4c:9c:
        25:42:01:6c:c4:c0:d3:18:ad:11:dc:70:81:48:17:07:1f:02:
        a9:6c:27:80:44:9d:61:94:66:d0:55:3e:d5:a4:3d:3d:c5:b7:
        25:de:c2:00:ec:b4:25:89:31:c2:f6:8f:ae:91:ba:76:ce:90:
        41:bc:34:e1:7e:92:26:33:7d:50:5a:2c:09:1a:3b:b6:4a:c8:
        27:ec:26:73
```

`show certificate bundle <bundle_name>`

Displays each certificate in the bundle the same way the **`show certificiate <cert_name>`** command (above) does.

Related Commands

Certificate Mode Commands

REST API Reference - certificateBundles

REST API Reference - certificates

Show Debug Commands

debug

Use

Use only when directed by technical support personnel to enable debugging features.

 Enabling this function may have a significant impact on system performance. Use only with extreme caution on a production system.

Default Setting

None

Command Mode

exec

Syntax

`show debug load-balancer object-count`

Shows counts of objects present in the system

`show debug sysdb internals show-internal`

Displays internal nodes

`show debug sysdb path-only show-internal`

Displays internal nodes

`show debug sysdb value <name>`

Shows the sysdb value

`show debug sysdb value <name> detailed`

Shows more detail about the sysdb value

Parameter	Type	Description
name	Word	Sysdb value to show.

Related

REST API Reference - debug

Show Failover Commands

show failover

Use

Use to show failover group information.

Default Setting

None

Command Mode

exec

Syntax

`show failover group <group_name>`

Show information about failover group

`show failover group brief`

Show summary of failover group status

Parameter	Type	Description
group_name	Word	Name of failover group.

Example

`show failover group <group_name>`

```
Group Status:         Ready
    Last Change:          Tue Oct 16 10:43:19 2012 UTC
    Associated Elements:
    Type                  Identifier                    Status
```

```
   ----                 ----------                   ------
   interface            em0 carp 1                   Active
   interface            po1 carp 1                   Standby
```

show failover group brief

```
Group  Status  Last Change
   --------------------------------------------
   fg1    Ready   Tue Oct 16 10:43:19 2012 UTC
```

Related

Failover Mode Commands

REST API Reference - failover

Show Forward Proxy Commands

show forward proxy

Use

Use to display information and statistics about a forward proxy.

Default Setting

None

Command Mode

exec

Syntax

`show forward-proxy <fp_name>`

Shows information about forward-proxy objects

`show forward-proxy <fp_name> statistics`

Shows statistics related to a forward-proxy

`show forward-proxy <fp_name> statistics detailed`

Shows all detailed statistics related to a forward-proxy

`show forward-proxy <fp_name> statistics http-requests`

Shows forward-proxy http request statistics

`show forward-proxy <fp_name> statistics http-response-codes`

Shows forward-proxy http response-code statistics

`show forward-proxy <fp_name> statistics http-responses`

Shows forward-proxy http response statistics

`show forward-proxy brief`

Show summary of forward-proxy status

Parameter	Type	Description
fp_name	Word	Show information about a specific forward-proxy

Example

Related

Forward Proxy Mode Commands

REST API Reference - forwardProxy

Show Interfaces Commands

show interfaces

Use

Use to show information about your interface settings.

Default Setting

None

Command Mode

exec

Syntax

`show interfaces`

Show configuration and statistics for all interfaces on the system

`show interfaces <intf>`

Show configuration and statistics for the specified interface

Parameter	Type	Description
intf	String	Name of an interface.

Example

```
show interfaces
em0 is up, line protocol is up
  Hardware is Intel82540EM, address is 0800.2765.c103
  Internet address is 10.200.0.1/24, broadcast is 10.200.0.255
  MTU 1500 bytes, BW 1000000 Kbit
  Full-duplex, 1 Gb/s, auto-negotiation: on
    3506 packets input, 288045 bytes
    Received 0 multicast
```

```
    0 input errors
    0 packets dropped
    2132 packets output, 272058 bytes
    Sent 0 multicast/broadcast
    0 output errors
em1 is up, line protocol is up
  Hardware is Intel82540EM, address is 0800.2767.241e
  Internet address is 192.0.2.1/24, broadcast is 192.0.2.255
  MTU 1500 bytes, BW 1000000 Kbit
  Full-duplex, 1 Gb/s, auto-negotiation: on
    0 packets input, 0 bytes
    Received 0 multicast
    0 input errors
    0 packets dropped
    1 packets output, 42 bytes
    Sent 0 multicast/broadcast
    0 output errors
em2 is up, line protocol is up
  Hardware is Intel82540EM, address is 0800.27b6.162a
  Internet address is 10.1.2.1/24, broadcast is 10.1.2.255
  MTU 1500 bytes, BW 1000000 Kbit
  Full-duplex, 1 Gb/s, auto-negotiation: on
    0 packets input, 0 bytes
    Received 0 multicast
    0 input errors
    0 packets dropped
    161330 packets output, 6775860 bytes
    Sent 0 multicast/broadcast
    0 output errors
lo0 is up, line protocol is up
  Hardware is Loopback, address is 6c.6f.30.0
  Internet address is 127.0.0.1/8
  Internet address is ::1/128
  MTU 16384 bytes, BW 0 Kbit
    20308 packets input, 1373855 bytes
    Received 0 multicast
    0 input errors
    0 packets dropped
    20338 packets output, 1375794 bytes
    Sent 0 multicast/broadcast
    0 output errors
```

show interface em0

```
em0 is up, line protocol is up
  Hardware is Intel82540EM, address is 0800.2765.c103
  Internet address is 10.200.0.1/24, broadcast is 10.200.0.255
  MTU 1500 bytes, BW 1000000 Kbit
  Full-duplex, 1 Gb/s, auto-negotiation: on
    3551 packets input, 291627 bytes
    Received 0 multicast
    0 input errors
    0 packets dropped
    2156 packets output, 276266 bytes
    Sent 0 multicast/broadcast
```

```
    0 output errors
```

Related

Interface Mode Commands

Show LACP Commands

Show Port Channel Mode Commands

REST API Reference - interface

Show IP Commands

show ip

Display IPv4-related settings.

Use

This command has the following uses for IPv4:

- To see the status and configuration of interfaces.
- To see IP routes.
  - A connected route (C) is one the system automatically adds based on the interface IP address.
  - A static route (S) is one you created using the ip route command.
  - The detail form of IP routes includes additional internal details about the routing table that you can use for troubleshooting. For example, the detail includes ARP entries.
- To see IP traffic counters. These counters are helpful for network troubleshooting. In general, if a counter increases at an unusual rate, you should determine why. Some increases may reflect hacking attempts. Below are a few key counters and their descriptions:
  - IP statistics
    - Rcvd
      - First line (totals and local destinations)—Normal to have very high numbers.
      - All other counters—If a counter increases at an unusual rate, you should determine why. Some increases may reflect hacking attempts.
    - Frags
      - reassembled, fragmented, fragments—Normal to see these increase at a high rate.
      - timeouts—Normal to increase, but not as rapidly as the counters mentioned above.
    - Sent
      - First line (generated and forwarded)—Normal to have very high numbers.
      - no route—Indicates a configuration problem with the system or with a gateway. The system does not have a route for a destination. Check the system default route.
      - insufficient buffers—If this increases at an unusual rate, the system is overloaded.
  - ICMP statistics
    - Rcvd

        - unreachable—Normal to have a high number. The system receives unreachable errors when the system tries to send a large packet to a client or real server and a gateway cannot accept a packet that large or when the real server does not have the correct port set to accept connections.
        - echo and echor reply—Reflect ping commands. If this number increases at an unusual rate, it could be mean someone is trying to hack the system.
    - Sent
        - echo and echo reply—Reflect ping commands.

Default Setting

None

Command Mode

exec

Syntax

`show ip interface brief`

Show a status summary of IP interfaces

`show ip route [detailed]`

Show IPv4 routing information

`show ip traffic`

IPv4 protocol statistics

Related Commands

Interface Mode Commands

IP Route Example

REST API Reference - ip

show ipv6

Use

This command has the following uses:

- To see IPv6 routes.
    - A connected route (C) is one the system automatically adds based on the interface IP address.
    - A static route (S) is one you created using the ip route command.

    - The detail of IP routes includes additional internal details about the routing table that you can use for troubleshooting. For example, the detail includes ARP entries.
- To see IPv6 traffic counters. These counters are helpful for network troubleshooting. In general, if a counter increases at an unusual rate, you should determine why. Some increases may reflect hacking attempts. Below are a few key counters and their descriptions:
    - IP statistics
        - Rcvd
            - First line (totals and local destinations)—Normal to have very high numbers.
            - All other counters—If a counter increases at an unusual rate, you should determine why. Some increases may reflect hacking attempts.
        - Frags
            - reassembled, fragmented, fragments—Normal to see these increase at a high rate.
            - timeouts—Normal to increase, but not as rapidly as the counters mentioned above.
        - Sent
            - First line (generated and forwarded)—Normal to have very high numbers.
            - no route—Indicates a configuration problem with the system or with a gateway. The system does not have a route for a destination. Check the system default route.
            - insufficient buffers—If this increases at an unusual rate, the system is overloaded.
    - ICMP statistics
        - Rcvd
            - unreachable—Normal to have a high number. The system receives unreachable errors when the system tries to send a large packet to a client or real server and a gateway cannot accept a packet that large or when the real server does not have the correct port set to accept connections.
            - echo and echo reply—Reflect ping commands. If this number increases at an unusual rate, it could be mean someone is trying to hack the system.
        - Sent
            - echo and echo reply—Reflect ping commands.

Default Setting

None

Command Mode

exec

Syntax

`show ipv6 route [detailed]`

Show IPv6 routing information

`show ipv6 traffic`

IPv6 protocol statistics

Related Commands

Interface Mode Commands

REST API Reference - ip

show ip dns

Use

Use to display the configured DNS settings.

Default Setting

None

Command Mode

exec

Syntax

show ip dns

Show dns information

Example

```
show ip dns
   Configuration
     Name Servers:      198.51.100.25
     Search Domains:    int.f5.com
     Timeout (seconds): <default> (3)
     Retries:           <default> (1)
     Admin Status:      online
```

Related Commands
ip dns

REST API Reference - ip

show ip filter-list

Use

Use to display information about the filter list.

To see the actual rules in the filter list, use **`show run`**.

Default Setting

None

Command Mode

exec

Syntax

`show ip filter-list [filter_name]`

Display information about the specified ip filter list

Parameter	Type	Description
filter_name	Word	Name of the IP filter list

Related Commands

ip filter-list

virtual-ip attach ip-filter

REST API Reference - connFilter

Show LACP Commands

show lacp

Use

Use to show LACP settings on port channels configured to use LACP.

Default Setting

None

Command Mode

exec

Syntax

`show lacp`

Show the status of LACP port channels

`show lacp <chan_num>`

Show the status of LACP port channels

`show lacp <chan_num> internal`

Show information about the local LACP state

`show lacp <chan_num> neighbor`

Show information about the neighbor LACP state

`show lacp internal`

Show information about the local LACP state

`show lacp neighbor`

Show information about the neighbor LACP state

Parameter	Type	Description
chan_num	Integer	Port channel number. Use only the number. For example, **show lacp 1**.

Example

show lacp

```
State codes:
     A = Active          P = Passive
     S = Short Timeout   L = Long Timeout
     G = Aggregatable    I = Individual
     + = In Sync         - = Out of Sync

  C = Collecting
     D = Distributing
     ! = Using default for peer
     X = Timeout Expired
   Port Channel 1:
   Intf  Peer      |  Port  SysId                    Key     Priority  State
   -------------------------------------------------------------------------------
   em1   Internal  |  2     8000,08-00-27-D8-0E-D4   0x00F0  32768     ALG+CD (!)
   em1   Neighbor  |  0     FFFF,00-00-00-00-00-00   0x0000  65535     PLG+CD

Port Channel 2:
   Intf  Peer  |  Port  SysId  Key  Priority  State
   -----------------------------------------------
```

[This port channel is configured to use "on" (static) mode.]

show lacp 1

```
State codes:
     A = Active          P = Passive
     S = Short Timeout   L = Long Timeout
     G = Aggregatable    I = Individual
     + = In Sync         - = Out of Sync

  C = Collecting
     D = Distributing
     ! = Using default for peer
     X = Timeout Expired
   Port Channel 1:
   Intf  Peer      |  Port  SysId                    Key     Priority  State
   -------------------------------------------------------------------------------
```

```
   em1   Internal  |  2     8000,08-00-27-D8-0E-D4  0x00F0  32768     ALG+CD (!)
   em1   Neighbor  |  0     FFFF,00-00-00-00-00-00  0x0000  65535     PLG+CD
```

show lacp 1 internal

```
State codes:
     A = Active          P = Passive
     S = Short Timeout   L = Long Timeout
     G = Aggregatable    I = Individual
     + = In Sync         - = Out of Sync

  C = Collecting
     D = Distributing
     ! = Using default for peer
     X = Timeout Expired
   Port Channel 1:
   Intf  |  Port  SysId                   Key     Priority  State
   ------------------------------------------------------------------------
   em1   |  2     8000,08-00-27-D8-0E-D4  0x00F0  32768     ALG+CD (!)
```

show lacp 1 neighbor

```
Port Channel 1:
   Intf  |  Port  SysId                   Key     Priority  State
   ---------------------------------------------------------------
   em1   |  0     FFFF,00-00-00-00-00-00  0x0000  65535     PLG+CD
```

Related

Show Interfaces Commands

Show Port Channel Mode Commands

REST API Reference - channelPorts

Show Licensing Commands

show licensing

Use

Use to display licensing information.

Default Setting

None

Command Mode

exec

Syntax

`show licensing brief`

Show summary of all licenses

`show licensing detailed`

Show details of all licenses

`show licensing feature base`

Show details of base license

`show licensing feature base detailed`

Show details of the base license

`show licensing host-id`

Display host ID information

`show licensing host-id detailed`

Not used

`show licensing host-id xml`

Not used

show licensing statistics

Show statistics for licensed rate limits

Examples

show licensing brief

```
Feature Active Expiration HTTP Req/s Limit TCP Conn/s Limit Process Limit Phone Home

-------------------------------------------------------------------------------------
   ---

base true 08/30/2013 15:28:10 100 100 1 on
```

show licensing detailed

```
base:
   Active: true
   Expiration: 08/30/2013 15:28:10
   Username: testlab
   Version: TESTLAB
   UUID: 33373436-3933-5355-4531-34354B39584C
   HTTP Requests Per Sec: 100
   TCP Connections Per Sec: 100
   Processes: 1
   Phone Home: on
```

show licensing feature base

```
Active: true
   Expiration: 08/30/2013 15:28:10
   Username: testlab
   Version: TESTLAB
   UUID: 33373436-3933-5355-4531-34354B39584C
   HTTP Requests Per Sec: 100
   TCP Connections Per Sec: 100
   Processes: 1
   Phone Home: on
```

show licensing feature base detailed

```
Active: true
   Expiration: 08/30/2013 15:28:10
   Username: testlab
```

```
    Version: TESTLAB
    UUID: 33373436-3933-5355-4531-34354B39584C
    HTTP Requests Per Sec: 999999
    TCP Connections Per Sec: 999999
    Processes: 0
    Phone Home: off

base license contents:
    MIME-Version: 1.0
    Content-Type: multipart/signed; protocol="application/x-pkcs7-signature";
    micalg="sha1"; boundary="----5543FBAFD7D995A02B48C4D917217AC2"

This is an S/MIME signed message

------5543FBAFD7D995A02B48C4D917217AC2
    {
    "Username": "testlab",
    "UUID": "33373436-3933-5355-4531-34354B39584C",
    "Rate Limit": {
    "HTTP Requests Per Sec": "100",
    "Processes": "1",
    "TCP Connections Per Sec": "100"
    },
    "Feature": "base",
    "Version": "TESTLAB",
    "Expiration": "1377898090",
    "Phone Home": "on"
    }
    ------5543FBAFD7D995A02B48C4D917217AC2
    Content-Type: application/x-pkcs7-signature; name="smime.p7s"
    Content-Transfer-Encoding: base64
    Content-Disposition: attachment; filename="smime.p7s"

MIIJZgYJKoZIhvcNAQcCoIIJVzCCCVMCAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3
    DQEHAaCCBZEwggWNMIIEdaADAgECAgEEMA0GCSqGSIb3DQEBBQUAMIGUMSYwJAYD
    VQQDEx1GNSBOZXR3b3JrcywgSW5jLiBMaW5lUmF0ZSBDQTELMAkGA1UEBhMCVVMx
    EzARBgNVBAgTCldhc2hpbmd0b24xEDAOBgNVBAcTB1NlYXR0bGUxGjAYBgNVBAoT
    EUY1IE5ldHdvcmtzLCBJbmMuMRowGAYJKoZIhvcNAQkBFgtpbmZvQGY1LmNvbTAe
    Fw0xMzA2MTgyMNOTALICENSETYyMDMxNTlaMIGCMaaaaaQswCQYDVQQGEwJVUzET
    MBEGA1UECBMKV2FzaGluZ3RvbjEaMBgGA1UEChMRRjUgTmV0d29ya3MsIEluYy4x
    ETAPBgNVBAsTCExpbmVSYXRlMRMwEQYDVQQDEwp3d3cuZjUuY29tMRowGAYJKoZI
    hvcNAQkBFgtpbmZvQGY1LmNvbTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoC
    ggIBAN14te2AJzlrpD2FE65uQNpKZdB5CM5BeuG1x9UMPewIjPukt0HNb8LMQFX8
    WEW2/2B9BXRExY9LlNOdr+lJ6EmPy6lCYY3eqetqZIC//9NVBoi+ryg6YG8g901O
```

tAM5LecdOCCD1ZlYiuLVw4JemnegQ2/HZea7NLt4OR31deiL3hhXTR899eUWpMy5
44WhZW0+u06evD0+uySEqjYJyWHqQ84n4jmOOIQ5lTLpNMLG23XGLvYbNjdbAJFo
Qn+hrABL7y2mrKe1urqt+2nAKIu10nwII1d1PbQLOfy7BsyNugJDkgBFxy7YKEcw
OH4gW4d15mPKW7nQr7EzvT3KlrHqdvkXg8MF/n6hcKvnUMig7WeaVt1KbCHryqnc
js1nssJ94yqAwl6YHwuxMQj+o+gJV3mjH2SE8+yxmSLnANsblhAKvaqyJsexhAlb
EwpqnXoecsqUO+YsbqZPQxMVZQHlx8lPWmOr5m+j3my45Fylc87uYMXUETeGMgJs
fqOJQEScgXRJIJHaZnD+C7f4Dy48ondWkp8u5JY3CdiipZz+e7ywCcSlclKyWT9/
0uKqtNLZkGw+bbHjqAh0q8rFUBqD5Erdw8YyHJneEw0SCV5onzBSn6Qql4FGsFqZ
iHIaCcWSJVw2Ngq5K/f2k344oYdREcn0NOTALICENSEmyDAgMBAAGjgfkwgfYw
CQYDVR0TBAIwADAdBgNVHQ4EFgQUyOnnkqF2G04H9F+Nnwf4RvJAp14wgckGA1Ud
IwSBwTCBvoAUtSmPki//bC2cNQhY8yYwMYuGNJahgZqkgZcwgZQxJjAkBgNVBAMT
HUY1IE5ldHdvcmtzLCBJbmMuIExpbmVSYXRlIENBMQswCQYDVQQGEwJVUzETMBEG
A1UECBMKV2FzaGluZ3RvbjEQMA4GA1UEBxMHU2VhdHRsZTEaMBgGA1UEChMRRjUg
TmV0d29ya3MsIEluYy4xGjAYBgkqhkiG9w0BCQEWC2luZm9AZjUuY29tggkA/ako
JHLoI1wwDQYJKoZIhvcNAQEFBQADggEBAAh4+qcYWJtoF9z+jwvFLe5D+ljYEa82
2Su+jPrmc//B7GP4FLoF7Ber76yvAwLy1jXFhQ8De30R9cJL24P+KxlI+RlumabM
FNcMCyKtx8HVXL6NdXJirle+MgIKaaV/kTsJ4ZivSDJWKfkbTZRsTwqsz4nkyMko
vB78xEjLHzshhBMACOTALICENSEbbbbbAoytGKncDRjeml9a7+BV0KsjOpbzOsWY
6YgIXIojcRw8JN3UB1uns39CFDLPsLQcLM6eTHydLO5kjEs669/+1K9hzM1zLcF6
F2JRrTVKDK+DAk62QmSBsSaQV+wIGaCZAhh70EXC2if5M/Xvrqs91Y8xggOdMIID
mQIBATCBmjCBlDEmMCQGA1UEAxMdRjUgTmV0d29ya3MsIEluYy4gTGluZVJhdGUg
Q0ExCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpXYXNoaW5ndG9uMRAwDgYDVQQHEwdT
ZWF0dGxlMRowGAYDVQQKExFGNSBOZXR3b3JrcywgSW5jLjEaMBgGCSqGSIb3DQEJ
ARYLaW5mb0BmNS5jb20CAQQwCQYFKw4DAhoFAKCB2DAYBgkqhkiG9w0BCQMxCwYJ
KoZIhvcTHISWILLNOTWORKb3DQEJBTEPFw0xMzA4MTYyMTI4MTBaMCMGCSqGSIb3
DQEJBDEWBBTscxbPTF6XFUwYBJGpQofUETaN0DB5BgkqhkiG9w0BCQ8xbDBqMAsG
CWCGSAFlAwQBKjALBglghkgBZQMEARYwCwYJYIZIAWUDBAECMAoGCCqGSIb3DQMH
MA4GCCqGSIb3DQMCAgIAgDANBggqhkiG9w0DAgIBQDAHBgUrDgMCBzANBggqhkiG
9w0DAgIBKDANBgkqhkiG9w0BAQEFAASCAgCg0B9TrmrJoB3ym4+v7Y1z7DvO9MnE
5EF/jIcCR4xTktzxmAn77law/H/JqA+U3HHLIUKb3J06lIbZsOVi7cNR2wvThtc/
YyQ9h3ouYOOSXbGKVFNRLzCwnf38Zato9ozbyfc0kn6H+8dX00h3Uq9x9zUZcb/l
k2rfk2XMjQgjB8eUxVWExXWuj1bg/gW9ldvz0tEZM1mndiqOn9+5/MqUGmqv+Nze
djGxA9rDx5qSzbxx0J6jezHrMdSY3XkT/MM1BaIKOCeozuqAv3ZKgzQT3LCLsdky
7vYz4VIR7aL/YT6sw1I8CIJo0I44bOGVoVTpVNyMslc17q4WEBKoZfWXhhfEgYQX
wjHZDbNeXY+AYpeNIumjv8KeG+3mHjFxIewPX58c3pOIKdUeuyWKUw41KfnQ4KaG
mXPOu+LuN2hWijIoePBHk7IHOcUnweQykF4yv/IUegwXhI4l+Qn8+e5p27QT26V+
ibeQ68nK1baSXcUBevrDhMQrfkhZWzjThKqrb/cDW5JZOH+AwYvwdFjI1Z6FZhr+
UtTF33JsUBo9P7nP4TlwK7x4q3pOTCaboHfWaoMUL7UngENGwsYIWLHcrd4ek9eY
MR8oDqP6R816ih4grxcKjM0FoYqN8OEyLeQCFkvc/oBaqPsdRhhjpUTNDmhJrDeX
plDjb5nZjGBI/w==

------5543FBAFD7D995A02B48C4D917217AC2--

show licensing host-id

```
2AEF8MWA-45BC-4B4E-953C-7A65B1F433333368
```

Related

CLI Reference - Licensing Mode Commands

REST API Reference - licensing

Show Load Balancer Commands (Deprecated)


In LineRate version 2.0, the **`show`** load-balancer command is deprecated and is replaced by the show proxy command. The corresponding REST API nodes are not affected.

Use the following commands to display statistics for the load balancer.

show load-balancer statistics

Use

Use to display statistics related to the load balancer, which includes combined, global statistics for the whole system.

The real server limits vary based on the amount of system RAM:

- Less than 48 GB memory—8,000 real servers.
- 48 GB memory or more—24,000 real servers.

The virtual IP limits vary based on the amount of system RAM:

- Less than 48 GB memory—2,000 VIPs.
- 48 GB memory or more—6,000 VIPs.

The virtual server limits vary based on the amount of system RAM:

- Less than 48 GB memory— 2,000 virtual servers.
- 48 GB memory or more—6,000 virtual servers.

The proxy limits vary based on the amount of system RAM:

- Less than 48 GB memory— 2,000 proxies.
- 48 GB memory or more—6,000 proxies.

Default Setting

None

Command Mode

exec

Syntax

show load-balancer statistics

Display load balancer statistics

show load-balancer statistics detailed

Display detailed load balancer statistics

show load-balancer limits

Display load balancer limits; limits vary based on the amount of system RAM

Examples

show load-balancer statistics

```
Load Balancer Clients
Open Connections: 0
Total Input Traffic (1 min avg): 0.000 bits/sec
Total Output Traffic (1 min avg): 0.000 bits/sec
SSL Input Traffic (1 min avg): 0.000 bits/sec
SSL Output Traffic (1 min avg): 0.000 bits/sec
New Connection Rate: 0 connections/sec
HTTP Request Rate: 0 requests/sec
Load Balancer Servers
Open Connections: 0
Input Traffic (1 min avg): 0.000 bits/sec
Output Traffic (1 min avg): 0.000 bits/sec
```

show load-balancer statistics detailed

```
eventLoopTime: 0.000000
eventLoopTime 1 Min Avg: 0.000000
eventLoopTime 5 Min Avg: 0.000000
keventTime: 0.000000
keventTime 1 Min Avg: 0.624966
keventTime 5 Min Avg: 0.624993
connClientOpen: 0
connClientReadSize: 0.000000
connClientReadSize 1 Min Avg: 0.000000
connClientReadSize 5 Min Avg: 0.000000
connClientWriteSize: 0.000000
connClientWriteSize 1 Min Avg: 0.000000
connClientWriteSize 5 Min Avg: 0.000000
connAcceptMax: 0.000000
connAcceptMax 1 Min Avg: 0.000000
connAcceptMax 5 Min Avg: 0.000000
connAccepted: 0.000000
connAccepted 1 Min Avg: 0.000000
connAccepted 5 Min Avg: 0.000000
connAcceptReportedAvail: 0.000000
connAcceptReportedAvail 1 Min Avg: 0.000000
connAcceptReportedAvail 5 Min Avg: 0.000000
httpClientRequestLatency: 0.000000
httpClientRequestLatency 1 Min Avg: 0.000000
```

```
httpClientRequestLatency 5 Min Avg: 0.000000
httpClientRespLatency: 0.000000
httpClientRespLatency 1 Min Avg: 0.000000
httpClientRespLatency 5 Min Avg: 0.000000
httpClientRespInitLatency: 0.000000
httpClientRespInitLatency 1 Min Avg: 0.000000
httpClientRespInitLatency 5 Min Avg: 0.000000
httpClientXactionLatency: 0.000000
httpClientXactionLatency 1 Min Avg: 0.000000
httpClientXactionLatency 5 Min Avg: 0.000000
connClientBytesRx: 0
connClientBytesRx/sec: 0
connClientBytesRx/sec 1 Min Avg: 0
connClientBytesRx/sec 5 Min Avg: 0

[Output continues with many additional lines of detail.]
```

show load-balancer limits

```
Maximum Virtual IPs: 2000
Maximum Virtual Servers: 2000
Maximum Real Servers: 8000
```

Related

CLI Reference - Load Balancer Mode Commands

REST API Reference - stats

REST API Reference - limits

Show NTP Commands

show ntp

Use

Use the view the NTP servers configured in the system and information about each one.

Default Setting

None

Command Mode

exec

Syntax

`show ntp`

Shows the IP address of the configured NTP servers

Related

CLI Reference - NTP Mode Commands

REST API Reference - ntp

Show Phone Home Commands

show phone-home

Use

Use to display phone home configuration information. Use the redacted form to remove password details.

Default Setting

None

Command Mode

exec

Syntax

`show phone-home`

Shows phone home configuration with encrypted password

`show phone-home redacted`

Shows phone home configuration with XXXXX as the password

Example

`show phone-home`

```
Phone Home Status
     Status:                OK
     Error Timestamp:       n/a
     Next Upload Time:      11/25/2013 15:33:00
   Phone Home Configuration
     F5 Login URL:          https://login.f5.com/resource/loginAction.jsp
     F5 Upload URL:         https://asb.f5.com/appsupport-analyzer/api/appsupport-
   bundles/
     Last Upload Time:      11/18/2013 15:33:03
```

```
      F5 User Name:           user@f5.com
      F5 Password:            abc2//zTOPw=
```

show phone-home redacted

```
Phone Home Status
      Status:                 OK
      Error Timestamp:        n/a
      Next Upload Time:       11/25/2013 15:33:00
   Phone Home Configuration
      F5 Login URL:           https://login.f5.com/resource/loginAction.jsp
      F5 Upload URL:          https://asb.f5.com/appsupport-analyzer/api/appsupport-
   bundles/
      Last Upload Time:       11/18/2013 15:33:03
      F5 User Name:           user@f5.com
      F5 Password:            XXXXXXXX
```

Related

Phone Home Mode Commands

Show Port Channel Mode Commands

show port-channel

Use

Use to view the configuration of port channels.

The Status column shows the following:

- Inactive—This interface is not being used by the port channel. For example, it may be administratively down, the cable may be unplugged, or LACP may have negotiated that it not be used. See the Reason column in the CLI, or the reason REST node for this interface.
- Operational—This interface is being used by the port channel.

The Reason column shows the following:

Unset	The port channel's mode has not been set or does not match the rest of the interfaces in the port channel.
Static	The port channel's mode is set to "on."
LACP	The port channel's mode is set to "active," which means it is using LACP.
Admin Status Up or Admin Status Down	The port channel's admin status is set to online or offline.
Link Protocol Up or Link Protocol Down	The connection to the port channel is not working for some hardware reason (cable disconnected, failed interface, etc.).
Unknown	The system cannot determine the status of the port channel.

Default Setting

None

Command Mode

exec

Syntax

show port-channel

Show the configured port channels

show port-channel <chan_num>

Show the configuration of the specified port channel

Parameter	Type	Description
chan_num	Integer	Port channel number. Use only the number. For example, **show port-channel 1**.

Example

show port channel

```
Port Channel 1:
   Port Protocol Mode Status Reason
   ---------------------------------------
   em1 LACP Active Operational LACP

Port Channel 2:
   Port Protocol Mode Status Reason
   -----------------------------------
   em2 Static On Operational Static
```

show port channel 1

```
Port Channel 1:
   Port Protocol Mode Status Reason
   ---------------------------------------
   em1 LACP Active Operational LACP
```

Related

Show Interfaces Commands

Show LACP Commands

REST API Reference - channelPorts

Show Proxy Commands

Use the following commands to display statistics for the proxy.

show proxy statistics

Use

Use to display statistics related to the proxy, which includes combined, global statistics for the whole system.

The real server limits vary based on the amount of system RAM:

- Less than 48 GB memory—8,000 real servers.
- 48 GB memory or more—24,000 real servers.

The virtual IP limits vary based on the amount of system RAM:

- Less than 48 GB memory—2,000 VIPs.
- 48 GB memory or more—6,000 VIPs.

The virtual server limits vary based on the amount of system RAM:

- Less than 48 GB memory— 2,000 virtual servers.
- 48 GB memory or more—6,000 virtual servers.

The proxy limits vary based on the amount of system RAM:

- Less than 48 GB memory— 2,000 proxies.
- 48 GB memory or more—6,000 proxies.

Default Setting

None

Command Mode

exec

Syntax

```
show proxy statistics
```

Display load balancer statistics

show proxy statistics detailed

Display detailed load balancer statistics

show proxy limits

Display load balancer limits; limits vary based on the amount of system RAM

Examples

show proxy statistics

```
Proxy Clients
Open Connections: 0
Total Input Traffic (1 min avg): 0.000 bits/sec
Total Output Traffic (1 min avg): 0.000 bits/sec
SSL Input Traffic (1 min avg): 0.000 bits/sec
SSL Output Traffic (1 min avg): 0.000 bits/sec
New Connection Rate: 0 connections/sec
HTTP Request Rate: 0 requests/sec
Proxy Servers
Open Connections: 0
Input Traffic (1 min avg): 0.000 bits/sec
Output Traffic (1 min avg): 0.000 bits/sec
```

show proxy statistics detailed

```
Proxy Global Statistics
eventLoopTime: 0.000000
eventLoopTime 1 Min Avg: 0.000000
eventLoopTime 5 Min Avg: 0.000000
keventTime: 0.000000
keventTime 1 Min Avg: 0.624966
keventTime 5 Min Avg: 0.624993
connClientOpen: 0
connClientReadSize: 0.000000
connClientReadSize 1 Min Avg: 0.000000
connClientReadSize 5 Min Avg: 0.000000
connClientWriteSize: 0.000000
connClientWriteSize 1 Min Avg: 0.000000
connClientWriteSize 5 Min Avg: 0.000000
connAcceptMax: 0.000000
connAcceptMax 1 Min Avg: 0.000000
connAcceptMax 5 Min Avg: 0.000000
connAccepted: 0.000000
connAccepted 1 Min Avg: 0.000000
connAccepted 5 Min Avg: 0.000000
connAcceptReportedAvail: 0.000000
connAcceptReportedAvail 1 Min Avg: 0.000000
connAcceptReportedAvail 5 Min Avg: 0.000000
httpClientRequestLatency: 0.000000
httpClientRequestLatency 1 Min Avg: 0.000000
httpClientRequestLatency 5 Min Avg: 0.000000
```

```
httpClientRespLatency: 0.000000
httpClientRespLatency 1 Min Avg: 0.000000
httpClientRespLatency 5 Min Avg: 0.000000
httpClientRespInitLatency: 0.000000
httpClientRespInitLatency 1 Min Avg: 0.000000
httpClientRespInitLatency 5 Min Avg: 0.000000
httpClientXactionLatency: 0.000000
httpClientXactionLatency 1 Min Avg: 0.000000
httpClientXactionLatency 5 Min Avg: 0.000000
connClientBytesRx: 0
connClientBytesRx/sec: 0
connClientBytesRx/sec 1 Min Avg: 0
connClientBytesRx/sec 5 Min Avg: 0

[Output continues with many additional lines of detail.]
```

show proxy limits

```
Maximum Forward Proxies: 0
Maximum Virtual IPs: 2000
Maximum Virtual Servers: 2000
Maximum Real Servers: 8000
```

Related

Proxy Mode Commands

REST API Reference - stats

REST API Reference - limits

Show Real Server Commands

Use the following commands to display information about real servers.

show real-server

Use

Use to display general information about existing real servers.

For the **`show real-server <rs_name>`** command, the third column shows that the configuration comes from of the following:

- set locally—The setting is configured directly for the real server.
- inherited from <base_name>—The setting is inherited from the listed real server base.
- default—The setting is not configured directly for the real server or in the real server base.

Default Setting

None

Command Mode

exec

Syntax

`show real-server <rs_name>`

Show the configuration and status of the specified real server

`show real-server brief`

Show summary of real server configuration, status, and current statistics (snapshot at the time you ran the command)

Parameter	Type	Description
rs_name	Word	Name of the real server.

Examples

```
show real-server rsweb1
Configuration
  Address:                10.1.2.101:8080 set locally
  Admin Status:           online          inherited from rsbase_web
  Max. Connections:       1000            inherited from rsbase_web
  TCP Options:            <none>          default
  SSL Profile:            ssl_prof_init1  inherited from rsbase_web
  Service Type:           http            inherited from rsbase_web
  Health Monitors:
    Name                  Svc             Admin                     SSL Profile
    hm_web1               http            online                    <UNSET>
  Max. Request In Flight: 1               default
  Keepalive Timeout:      10 s            inherited from rsbase_web
  Response Timeout:       60 s            inherited from rsbase_web
  Response Idle Timeout:  60 s            inherited from rsbase_web
  Request Idle Timeout:   0 s             default
  Tunnel Idle Timeout:    0 s             default

  Is Proxy:               0               default

  Request Rate Limit:     0 reqs/sec

  Request Rate Limit Burst Size: <auto>
Current Status
  Health Status: down
```

```
   show real-server brief
Name    Address      Port  Svc   Admin  Health Conns Rx Mbps Tx Mbps
-------------------------------------------------------------------
rsweb1  10.1.2.101   8080  http  online  down   0     0.0     0.0
rsweb2  10.1.2.102   8080  http  online  down   0     0.0     0.0
```

Related Commands

Real Server Mode Commands

REST API Reference - realServer

REST API Reference - realServerGroup

show real-server statistics

Use

Use to display a snapshot of various statistics related to real server traffic and connections. The statistics only show the initiation side of connections.

Default Setting

None

Command Mode

exec

Syntax

`show real-server <rs_name> statistics`

Shows statistics related to a real server

`show real-server <rs_name> statistics connection-errors`

Shows real server connection error statistics. If you see a lot of errors, you should investigate the cause.

`show real-server <rs_name> statistics connections`

Shows real server connection statistics

`show real-server <rs_name> statistics detailed`

Shows all statistics related to a real server

`show real-server <rs_name> statistics http-requests`

Shows real server HTTP request statistics

`show real-server <rs_name> statistics http-response-codes`

Shows real server HTTP response code statistics

`show real-server <rs_name> statistics http-responses`

Shows real server HTTP response statistics

`show real-server <rs_name> statistics server-latency`

Shows real server latency statistics

`show real-server <rs_name> statistics traffic`

Shows real server traffic statistics

Parameter	Type	Description
rs_name	Word	Name of the real server.

Examples

```
show real-server rsweb1 statistics
Connections
  Open:           0
  Opened:         0 connections/sec
Connection Errors
  Timed out:      0
  Lost:           0
  Reset:          0
  Refused:        0
  Addr In Use:    0
  Idle timeout:   0
Traffic
  L7 Receive:     0 bits/sec
  L7 Send:        0 bits/sec
HTTP Requests
  Request Rate:   0 requests/sec
  Total:          0
  Queue Size:     0
HTTP Responses
  Response Rate:  0 responses/sec
  Errored:        0
Server Latency
  Transaction:    0 s
Connect Tunnels
  Open:           0
  Opened Rate:    0 tunnels/sec
```

```
show real-server rsweb1 statistics connection-errors
Connection Errors
  Timed out:           0
  Lost:                0
  Reset:               0
  Refused:             0
  Addr In Use:         0
  Idle timeout:        0
  Server Closed Early: 0
  Client Closed Early: 0
```

```
show real-server rsweb1 statistics connections
Connections
  Open:            0
  Opened:          0 connections/sec
```

```
    1 Min Average: 0 connections/sec
    5 Min Average: 0 connections/sec
  Total Opened:    0
  Closed:          0
  Need More Conns: 0
  Initiated:       0
```

show real-server rs1 statistics http-requests

```
HTTP Requests
  Request Rate:    0 requests/sec
    1 Min Average: 0 requests/sec
    5 Min Average: 0 requests/sec
  Total:           0
  Abandoned:       0
  Queue Size:      0
    1 Min Average: 0
    5 Min Average: 0
  Timed Out:       0
```

show real-server rs1 statistics http-response-codes

```
HTTP Response Codes  Count  % of total count
  100:               0      0
  101:               0      0
  102-199:           0      0
  200:               0      0
  201:               0      0
  202:               0      0
  203:               0      0
  204:               0      0
  205:               0      0
  206:               0      0
  207-299:           0      0
  300:               0      0
  301:               0      0
  302:               0      0
  303:               0      0
  304:               0      0
  305:               0      0
  306:               0      0
  307:               0      0
  308-399:           0      0
[Output continues]
```

show real-server rs1 statistics http-responses

```
HTTP Responses
  Response Rate:        0 responses/sec
    1 Min Average:      0 responses/sec
    5 Min Average:      0 responses/sec
  Total:                0
  Good:                 0
  Errored:              0
```

```
  Extraneous:           0
  Abandoned:            0
  Timed Out:            0
  Idle Timed Out:       0
  Error receiving body: 0
```

show real-server rs1 statistics server-latency

```
Server Latency
  Transaction:     0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Request:         0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Initial:         0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Response:        0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
```

show real-server rs1 statistics traffic

```
Traffic
  L7 Receive:        0 bits/sec
    1 Min Average:   0 bits/sec
    5 Min Average:   0 bits/sec
  L7 Send:           0 bits/sec
    1 Min Average:   0 bits/sec
    5 Min Average:   0 bits/sec
  L7 Total Received: 0 Bytes
  L7 Total Sent:     0 Bytes
  Avg Read Size:     0 Bytes
    1 Min Average:   0 Bytes
    5 Min Average:   0 Bytes
  Avg Write Size:    0 Bytes
    1 Min Average:   0 Bytes
    5 Min Average:   0 Bytes
```

Related Commands

Real Server Mode Commands

show real-server group

Use

Use to display information about real server groups.

Default Setting

None

Command Mode

exec

Syntax

`show real-server group <gr_name>`

Show real server configuration, status, and statistics for real servers in the specified group

`show real-server group <gr_name> regex`

Show the regular expression used to add group members

Parameter	Type	Description
gr_name	Word	Name of the real server group.

Examples

`show real-server group rsgroup_websilo1`

```
rsgroup_websilo1 Group Members
Name    Address     Port  Svc   Admin   Health  Conns  Rx Mbps  Tx Mbps
-------------------------------------------------------------------------
rsweb1  10.1.2.101  8080  http  online  down    0      0.0      0.0
rsweb2  10.1.2.102  8080  http  online  down    0      0.0      0.0
```

`show real-server group rsgroup_websilo1 regex`

rsgroup_websilo1 Regular Expression Matches Matches for "rsweb.*"

Related Commands

Real Server Mode Commands

show real-server route

Use

Use to display information about real server routes.

Default Setting

None

Command Mode

exec

Syntax

show real-server route

Show summary of routes to all real servers, with the IP address of each real server and the gateway to each real server

Example

```
show real-server route
   Name   Address    Gateway Intf MTU
   -----------------------------------
   rsweb1 10.1.2.101 -       em2  1500

   rsweb2 10.1.2.102 -       em2  1500
```

Related Commands

Real Server Mode Commands

Show REST Server Commands

show rest-server

Use

Use to display configuration and status information about the REST server.

Default Setting

None

Command Mode

exec

Syntax

`show rest-server`

Show information about the HTTP server for REST API access

Example

```
show rest-server
Configuration
  Listening Interfaces:
    0.0.0.0:8443
    [::]:8443
  Remote Addresses:
    any
  SSL Profile:          self-signed
  Logging:              0
  Session Idle Timeout: 3.6 ks
Current Status
  UP
```

Related

REST Server Mode Commands

REST API Reference - restServer

Show Running Config Commands

show running-config

Use

Use to view the configuration you are currently running, including changes you have made since you last saved the configuration. The system has a default running config that it runs if it cannot find a startup config.

Default Setting

Default settings in the default running config are described in each command section.

Command Mode

exec

Syntax

`show running-config`

Shows the whole running configuration.

`show running-config brief`

Show the running configuration without IP filters and certificate and key data.

`show running-config diffs`

Show the differences from startup-config.

`show running-config redacted`

Show the configuration with sensitive values (passwords, keys) redacted.

Example

`show running-config brief`

```
Building configuration...
    !
```

```
hostname example-host
!
username admin secret encrypted "$2a$04$7TYufYOKVQ8i8bblVtZ1ierxZXzcH5mR/
QeaZH8WnWRzVEkPt0MgS" uid 2000
!
interface em0
ip address 10.0.2.15 255.255.255.0
ip address 10.200.0.1 255.255.255.0
!
interface em1
ip address 192.0.2.1 255.255.255.0
!
interface em2
ip address 10.1.2.1 255.255.255.0
!
interface em3
ip address 0.0.0.0 255.0.0.0
ip address 10.126.1.242 255.255.0.0
!
ip route 0.0.0.0/0 192.0.2.2
!
health-monitor hm_web1
interval 5
timeout 1
server-down "8/10"
server-up "9/10"
type http
request-method GET
request-target "/health.html"
admin-status online
!
ssl profile self-signed
attach primary-certificate self-signed
attach private-key self-signed
!
ssl profile ssl_prof_init1
!
ssl profile ssl_prof_secure.example.com
attach primary-certificate cert_prim_secure.example.com
attach private-key key_secure.example.com
attach chain-certificate cert_chain_secure.example.com1
!
real-server base rsbase_web
```

```
admin-status online
max-connections 1000
service http
response-timeout 60
response-idle-timeout 60
keepalive-timeout 10
attach ssl profile ssl_prof_init1
attach health-monitor hm_web1
!
real-server rsweb1 ip 10.1.2.101 8080 base rsbase_web
real-server rsweb2 ip 10.1.2.102 8080 base rsbase_web
!
virtual-ip base vipbase_web1
admin-status online
service http
keepalive-timeout 5
!
virtual-ip vipweb1
ip address 192.0.2.1 80
base vipbase_web1
attach ssl profile ssl_prof_secure.example.com
!
real-server group rsgroup_websilo1
members by regex "rsweb.*"
!
virtual-server websilo1
lb-algorithm round-robin-free
service http
attach virtual-ip vipweb1 default
attach real-server group rsgroup_websilo1
!
ssh
allow from 10.200.0.0/24
allow from 10.0.2.0/24
allow to 10.200.0.1 22
allow to 10.0.2.15 22
!
rest-server
allow from any
allow to any 8443
attach ssl profile self-signed
!
license-manager
```

```
    ip 1 10.126.64.38
    !
    certificate cert_chain_secure.example.com1
    ! Cert data not shown in brief output
    !
    certificate cert_prim_secure.example.com
    ! Cert data not shown in brief output
    !
    certificate self-signed
    ! Cert data not shown in brief output
    !
    certificate bundle cert_bndlsecure.example.com
    ! Cert data not shown in brief output
    !
    key key_secure.example.com
    ! Key data not shown in brief output
    !
    key self-signed
    ! Key data not shown in brief output
```

show running-config diffs

```
Building configuration...

--- startup-config
    +++ running-config
    @@ -14,10 +14,15 @@
    ip address 10.1.2.1 255.255.255.0
    !
    interface em3
    + ip address 0.0.0.0 255.0.0.0
    ip address 10.126.1.242 255.255.0.0
    !
    ip route 0.0.0.0/0 192.0.2.2
    !
    +health-monitor hm2
    + override-port 20
    + attach ssl profile ssl_prof_init1
    +!
    health-monitor hm_web1
    interval 5
    timeout 1
```

Related

Getting Started Guide - Complete Example Show Run Output

Getting Started Guide - Running Config and Startup Config

Getting Started Guide - Show Startup Config Commands

REST API Reference - running

Show Scripting Commands

show scripting

Use

Use to display information about the scripting system, including the installed Node Packaged Modules.

Default Setting

None

Command Mode

exec

Syntax

`show scripting packages bundled`

Show packages bundled with the system

`show scripting packages bundled <module_name>`

Show packages bundled with the system

`show scripting packages user`

Show information about packages installed by users

`show scripting packages user <module_name>`

Show information about packages installed by users

`show scripting version`

Show versions of scripting engines

Parameter	Type	Description

rmodule_name	Word	Name of bundled module.

Examples

show scripting packages bundled

```
Package Version Description

-------------------------------------

redis   0.8.2   Redis client library
```

show scripting packages bundled <module_name>

```
Version: 0.8.2
   Description:
     Redis client library
   Dependencies:
     redis@0.8.2/usr/linerate/node/scripting/node_modules/redis
     +-- (empty)
```

show scripting packages user

```
   Package    Version      Description
   ------------------------------------------------------------------------
   mysql      2.0.0-alpha8 A node.js driver for mysql. It is written in Ja...
   underscore 1.5.1        JavaScript's functional programming helper libr...
```

show scripting packages user <module_name>

```
Version: 2.0.0-alpha8
   Description:
     A node.js driver for mysql. It is written in JavaScript, does not require
   compiling, and is 100% MIT licensed.
   Dependencies:
     mysql@2.0.0-alpha8/home/linerate/data/scripting/lib/node_modules/mysql
     |-- bignumber.js@1.0.1
     +-- require-all@0.0.3
```

show scripting version

```
   Node API compatibility version: 0.8.3
   LineRate Scripting engine version: 0.1.0
```

Related

CLI Reference - Scripting Mode Commands

REST API Reference - scripting

Show Script Commands

Use the following commands to display information about scripts.

show script

Use

Use to display information about existing scripts.

Default Setting

None

Command Mode

exec

Syntax

`show script <script_name>`

Show information about the specified script.

`show script <script_name> statistics`

Shows statistics related to the specified script.

`show script <script_name> statistics detailed`

Shows all detailed statistics related to the specified script.

`show script brief`

Show summary information for all configured scripts.

Parameter	Type	Description
script_name	Word	Name of the script.

Examples

show script add-proxyhost

```
   Admin Status:             online
   Restart Mode:             manual
   Created At:               Tue Oct 22 17:55:05 2013 UTC
   Attached Entities:
     Proxy Request Listeners
       Name                  Count
       vs1                   2 / 5

   Per-process Properties:
     Busy Timeouts
       Timeout      Count      Status
       1.063s       1 / 5
       1.604s       1 / 5
       1.605s       1 / 5
       1.606s       1 / 5
       1.607s       1 / 5

   Status:                   DOWN: syntax error
   Last Error
     Timestamp:              Tue Oct 22 17:56:31 2013 UTC
     Stage:                  compile
     Message:
       encountered the following errors when compiling script
       add-proxyhost:1: SyntaxError: Unexpected token ;
       a = ;
           ^
```

show script add-proxyhost statistics

```
Proxy
     Requests Redirected:      0
     Request Events:           0
   TCP Connections
     Listening:                0
     Open Server:              0
     Open Client:              0
   Timers
     Pending:                  0
   Miscellaneous
     Unrecoverable exceptions: 0
     Auto-restarts:            0
```

show script add-proxyhost statistics detailed

```
Proxy
     Requests Redirected:      0
     Request Events:           0
   TCP Connections
     Listening:                0
     Open Server:              0
     Open Client:              0
   Timers
     Pending:                  0
   Miscellaneous
     Unrecoverable exceptions: 0
     Auto-restarts:            0
   LROS(config)# show script add-proxyhost statistics detailed
   Proxy
     Requests Redirected:      0
     Request Events:           0
   TCP Connections
     Listening:                0
     Open Server:              0
     Open Client:              0
   Timers
     Pending:                  0
   Miscellaneous
     Unrecoverable exceptions: 0
     Auto-restarts:            0
```

show script brief

```
   Name           Admin  Status         Rstrt-mode  Excpt  Rstrt  Req-Listeners
   ------------------------------------------------------------------------------
   add-proxyhost  off    DOWN:compiled  auto        0      0      <none>
   test_script    off    DOWN:admin     auto        0      0      <none>
```

Related

Script Mode Commands

Show SSL Commands

1. show ssl profile <profile_name>
2. show ssl statistics

show ssl profile <profile_name>

Use

Use to display information about the specified SSL profile.

Default Setting

None

Command Mode

exec

Syntax

show ssl profile <profile_name>

Parameter	Type	Description
profile_name	Word	Name of the SSL profile.

Example

```
show ssl profile ssl_prof_secure.example.com
Configuration
  Primary Cert Name:        cert_prim_secure.example.com                set locally
  Private Key Name:         key_secure.example.com                      set locally
  Disabled Protocols List: lros-default
  Cipher List:              HIGH:!ADH:!SSLv2:!PSK:!ECDH:!kEDH:+AES:+3DES default
  SSL Version List:         <none>                                      default
  SSL Session Cache Mode:   auto size                                   default
  SSL Session Cache Size:   10 Mi                                       default
  SSL Session Tickets Mode: enabled                                     default
Active Protocols: TLSv1
Ordered cipher list
  AES256-SHA
  AES128-SHA
```

```
  DES-CBC3-SHA
```

Related

SSL Mode Commands

REST API Reference - ssl

show ssl statistics

Use

Use to display SSL statistics, aggregated across all profiles.

Default Setting

None

Command Mode

exec

Syntax

show ssl statistics

Example

```
LROS# show ssl statistics
Statistics
  Session Ticket Successes (Termination):          0
  Session Ticket Failures (Termination):           0
  Session Cache Hits (Termination):                0
  Session Cache Misses (Termination):              0
  Cached Session Expired (Termination):            0
  Sessions Reused (Termination):                   0
  Sessions negotiated without reuse (Termination): 0
  Sessions Reused (Initiation):                    0
  Sessions negotiated without reuse (Initiation):  0
```

Related

SSL Mode Commands

Show Startup Config Commands

show startup-config

Use

Use to view the saved configuration that the system will use when you next restart the LineRate software.

Default Setting

None

Command Mode

exec

Syntax

`show startup-config`

Shows the saved configuration.

Example

show startup-config

```
example-host# show start
    !
    hostname example-host
    !
    username admin secret encrypted "$2a$04$7TYufYOKVQ8i8bblVtZ1ierxZXzcH5mR/
    QeaZH8WnWRzVEkPt0MgS" uid 2000
    !
    interface em0
    ip address 10.0.2.15 255.255.255.0
    ip address 10.200.0.1 255.255.255.0
    !
    interface em1
```

```
ip address 192.0.2.1 255.255.255.0
!
interface em2
ip address 10.1.2.1 255.255.255.0
!
interface em3
ip address 10.126.1.242 255.255.0.0
!
ip route 0.0.0.0/0 192.0.2.2
!
health-monitor hm_web1
interval 5
timeout 1
server-down "8/10"
server-up "9/10"
type http
request-method GET
request-target "/health.html"
admin-status online
!
ssl profile self-signed
attach primary-certificate self-signed
attach private-key self-signed
!
ssl profile ssl_prof_init1
!
ssl profile ssl_prof_secure.example.com
attach primary-certificate cert_prim_secure.example.com
attach private-key key_secure.example.com
attach chain-certificate cert_chain_secure.example.com1
!
real-server base rsbase_web
admin-status online
max-connections 1000
service http
response-timeout 60
response-idle-timeout 60
keepalive-timeout 10
attach ssl profile ssl_prof_init1
attach health-monitor hm_web1
!
real-server rsweb1 ip 10.1.2.101 8080 base rsbase_web
real-server rsweb2 ip 10.1.2.102 8080 base rsbase_web
```

```
!
virtual-ip base vipbase_web1
admin-status online
service http
keepalive-timeout 5
!
virtual-ip vipweb1
ip address 192.0.2.1 80
base vipbase_web1
attach ssl profile ssl_prof_secure.example.com
!
real-server group rsgroup_websilo1
members by regex "rsweb.*"
!
virtual-server websilo1
lb-algorithm round-robin-free
service http
attach virtual-ip vipweb1 default
attach real-server group rsgroup_websilo1
!
ssh
allow from 10.200.0.0/24
allow from 10.0.2.0/24
allow to 10.200.0.1 22
allow to 10.0.2.15 22
!
rest-server
allow from any
allow to any 8443
attach ssl profile self-signed
!
license-manager
ip 1 10.126.64.38
!
certificate cert_chain_secure.example.com1
pem-format
-----BEGIN CERTIFICATE-----
MIIDUDCCAjgCAQEwDQYJKoZIhvcNAQEFBQAwaDELMAkGA1UEBhMCVVMxETAPBgNV
! ...
-----END CERTIFICATE-----
quit
!
certificate cert_prim_secure.example.com
```

```
pem-format
-----BEGIN CERTIFICATE-----
MIIDUTCCAjkCAQEwDQYJKoZIhvcNAQEFBQAwdDELMAkGA1UEBhMCVVMxETAPBgNV
! ...
-----END CERTIFICATE-----
quit
!
certificate self-signed
pem-format
-----BEGIN CERTIFICATE-----
MIIDOjCCAiKgAwIBAgIJAPm1YLOdNan3MA0GCSqGSIb3DQEBBQUAMBwxGjAYBgNV
! ...
-----END CERTIFICATE-----
quit
!
certificate bundle cert_bndlsecure.example.com
pem-format
-----BEGIN CERTIFICATE-----
MIIEkDCCA/mgAwIBAgIQGwk7eGCW2je7pFGURsiWeDANBgkqhkiG9w0BAQUFADBf
! ...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIGKTCCBRGgAwIBAgIQZBvoIM4CCBPzLU0tldZ+ZzANBgkqhkiG9w0BAQUFADCB
! ...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIF7DCCBNSgAwIBAgIQbsx6pacDIAm4zrz06VLUkTANBgkqhkiG9w0BAQUFADCB
! ...
-----END CERTIFICATE-----
quit
!
key key_secure.example.com
pem-format
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA7beACBTJLlEEE3qaf+qBg7P5Igz2zzdY22pbZ0wO2vGaZK2R
! ...
-----END RSA PRIVATE KEY-----
quit
!
key self-signed
pem-format
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAvhALFtrOHMGc9bgYcFB5spZFVrH2SF8+lan5hsnO1ejXB5Nx
```

```
! ...
-----END RSA PRIVATE KEY-----
quit
```

Related

Running Config and Startup Config

Show Running Config Commands

REST API Reference - startup

Show TCP Commands

show tcp statistics

Use

Use to view the TCP protocol statistics.

Default Setting

None

Command Mode

exec

Syntax

`show tcp statistics`

Shows TCP protocol statistics

Example

`show tcp statistics`

```
Rcvd: 142024 Total
    0 checksum error, 0 bad offset, 0 too short, 0 invalid
    72072 packet (1158327 bytes) in sequence
    57 dup packets (0 bytes)
    0 partially dup packets (0 bytes)
    0 out-of-order packets (0 bytes)
    0 packets (0 bytes) with data after window
    0 packets after close
    0 window probe packets, 6797 window update packets
    6911 dup ack packets, 0 ack packets with unsent data
    50463 ack packets (1251571 bytes)
    Sent: 173290 Total, 0 urgent packets
    51321 control packets
```

```
    58560 data packets (1237978 bytes)
    0 data packets (0 bytes) retransmitted
    63409 ack only packets (21 delayed)
    0 window probe packets, 0 window update packets
    37735 Connections initiated, 6800 connections accepted, 13599 connections established
    44527 Connections closed (including 0 dropped, 967 embryonic dropped)
    14449 Total rxmt timeout, 0 connections dropped in rxmt timeout
    1021 Keepalive timeout, 65 keepalive probe, 956 Connections dropped in keepalive

27 Total open connections
    12 listen, 13 established, 2 syn sent, 0 syn received
    0 last ack, 0 fin wait 1, 0 fin wait 2
    0 close wait, 0 closing, 0 time wait, 0 closed
```

Related

TCP Mode Commands

REST API Reference - tcp

Show Tech Support Commands

show tech-support

Use

Use to view information about the system configuration, statistics, and much more for use by technical support personnel.

Default Setting

None

Command Mode

exec

Syntax

`show tech-support`

Show system information useful for LineRate tech support.

`show tech-support <fileuri>`

Show system information useful for LineRate tech support.

`show tech-support detailed`

Show detailed system information useful for LineRate tech support.

`show tech-support detailed <fileuri>`

Show detailed system information useful for LineRate tech support.

Parameter	Type	Description

fileuri	String	URI of a target file for tech-support dump. Protect URIs with quotes (e.g. "file:///home/linerate/tech-support" or "scp://example.com/tech-support").

Related

REST API Reference - tech-support

Show Version Commands

show version

Use

Use to show the system hardware and software information, including the LineRate version.

Default Setting

None

Command Mode

exec

Syntax

`show version`

System hardware and software information

Example

`show version`

```
F5 Networks LROS Version 2.2.0-RC4
Copyright (c) 2009-2013 F5 Networks, Inc.

RELEASE IMAGE (1dd189b/3e1441a/b2e5c0d)
Built Tue Nov 26 20:12:24 MST 2013
  by builder@cayman:/build/cayman/jenkins/workspace/2.2/lrs_release
Node API compatibility version: 0.8.3
LineRate Scripting engine version: 0.1.0

example-host uptime is 17 hours, 19 minutes
```

Related

REST API Reference - version

Show Virtual IP Commands

show virtual-ip

Use

Use to display information about virtual IPs.

Default Setting

None

Command Mode

exec

Syntax

`show virtual-ip <vip_name>`

Show information about the specified virtual IP

`show virtual-ip brief`

Show summary of virtual IP status

Parameter	Type	Description
vip_name	Word	Name of the virtual IP.

Examples

```
show virtual-ip vipweb1
Configuration
  Address:                  192.0.2.1:80                  set locally
  Admin Status:             online                        inherited from vipbase_web1
  SSL Profile:              ssl_prof_secure.example.com   set locally
  Max Conn. Queue Size:     <system maximum>              default
```

```
  TCP Options:              <none>                      default
  IP Filter:                <none>                      default
  Service Type:             http                        inherited from vipbase_web1
  Max Request In Flight:    1                           default
  Keepalive Timeout:        5 s                         inherited from vipbase_web1
  Max Request Header Size: 32768                        default
Current Status
  UP
```

```
show virtual-ip brief
Name     Address     Port  Svc   Admin   Conns  Rx Mbps  Tx Mbps
-------------------------------------------------------------------
vipweb1  192.0.2.1  80    http  online  0      0.0      0.0
```

Related

Virtual IP Mode Commands

REST API Reference - virtualIP

show virtual-ip statistics

Use

Use to display snapshot of statistics related to the virtual IP. The statistics only show the termination side of connections (from clients).

Default Setting

None

Command Mode

exec

Syntax

show virtual-ip <vip_name> statistics

Shows statistics related to a virtual IP

show virtual-ip <vip_name> statistics client-latency

Shows client latency statistics related to a virtual IP

show virtual-ip <vip_name> statistics connection-errors

Shows connection-error statistics related to a virtual IP

show virtual-ip <vip_name> statistics connections

Shows connection statistics related to a virtual IP

show virtual-ip <vip_name> statistics detailed

Shows all statistics related to a virtual IP

show virtual-ip <vip_name> statistics http-requests

Shows http request statistics related to a virtual IP

show virtual-ip <vip_name> statistics http-response-codes

Shows http response-code statistics related to a virtual IP

show virtual-ip <vip_name> statistics http-responses

Shows http response statistics related to a virtual IP

show virtual-ip <vip_name> statistics traffic

Shows traffic related to a virtual IP

Parameter	Type	Description
vip_name	Word	Name of the virtual IP.

Examples

```
show virtual-ip vipweb1 statistics
Connections
  Open:                   0
  Opened:                 0 connections/sec
Connection Errors
  Max Client Conns:       0
  Lost - process died:    0
  Timed Out:              0
  Reset:                  0
  Refused:                0
  Addr In Use:            0
  Idle Timeout:           0
  Filtered:               0
  No Virtual Server:      0
  SSL Profile Error:      0
  SSL Negotiation Failed: 0
Traffic
```

```
  L7 Received:            0 bits/sec
  L7 Sent:                0 bits/sec
  L7 SSL Received:        0 bits/sec
  L7 SSL Sent:            0 bits/sec
HTTP Requests
  Total:                  0
  Request Rate:           0 requests/sec
  Total errors:           0
HTTP Responses
  Started:                0
  Completed:              0
  Timed Out:              0
  Idle Timed Out:         0
  LB Forbidden:           0
  LB 5xx Error:           0
Client Latency
  Transaction:            0 s
```

show virtual-ip vipweb1 statistics client-latency

```
Client Latency
  Transaction:     0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Request:         0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Initial:         0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Response:        0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
```

show virtual-ip vipweb1 statistics connection-errors

```
Connection Errors
  Max Client Conns:       0
  Max Client Conns:       0 errors/sec
    1 Min Average:        0 errors/sec
    5 Min Average:        0 errors/sec
  Lost - process died:    0
  Timed Out:              0
  Reset:                  0
  Refused:                0
  Addr In Use:            0
  Idle Timeout:           0
  Filtered:               0
  No Virtual Server:      0
  SSL Profile Error:      0
  SSL Negotiation Failed: 0
```

show virtual-ip vipweb1 statistics connections

```
Connections
  Open:                     0
  Opened:                   0 connections/sec
    1 Min Average:          0 connections/sec
    5 Min Average:          0 connections/sec
  Total Opened:             0
  Closed:                   0
  SSL Negotiation attempts: 0
  SSL Connections Opened:   0
  Accept Events:            0 events/sec
    1 Min Average:          0 events/sec
    5 Min Average:          0 events/sec
  Total Accept Events:      0
  Accepted per loop:        0
    1 Min Average:          0
    5 Min Average:          0
  Accept Max:               0
    1 Min Average:          0
    5 Min Average:          0
  Available to Accept:      0
    1 Min Average:          0
    5 Min Average:          0
```

show virtual-ip vipweb1 statistics http-requests

```
HTTP Requests
  Total:            0
  Request Rate:     0 requests/sec
    1 Min Average:  0 requests/sec
    5 Min Average:  0 requests/sec
  Good:             0
  Total errors:     0
  Connect Requests: 0
  Timed Out:        0
```

show virtual-ip vipweb1 statistics http-response-codes

```
HTTP Response Codes  Count  % of total count
  100:               0      0
  101:               0      0
  102-199:           0      0
  200:               0      0
  201:               0      0
  202:               0      0
  203:               0      0
  204:               0      0
  205:               0      0
  206:               0      0
  207-299:           0      0
  300:               0      0
  301:               0      0
  302:               0      0
  303:               0      0
  304:               0      0
```

```
  305:               0      0
  306:               0      0
  307:               0      0
  308-399:           0      0
[Output continues]
```

show virtual-ip vipweb1 statistics http-responses

```
HTTP Responses
  Started:               0
  Completed:             0
  Timed Out:             0
  Idle Timed Out:        0
  LB Forbidden:          0
  LB 5xx Error:          0
  LB 5xx Error Rate:     0 errors/sec
    1 Min Average:       0 errors/sec
    5 Min Average:       0 errors/sec
  LB Connects Forbidden: 0
```

show virtual-ip vipweb1 statistics traffic

```
Traffic
  L7 Received:           0 bits/sec
    1 Min Average:       0 bits/sec
    5 Min Average:       0 bits/sec
  L7 Sent:               0 bits/sec
    1 Min Average:       0 bits/sec
    5 Min Average:       0 bits/sec
  L7 SSL Received:       0 bits/sec
    1 Min Average:       0 bits/sec
    5 Min Average:       0 bits/sec
  L7 SSL Sent:           0 bits/sec
    1 Min Average:       0 bits/sec
    5 Min Average:       0 bits/sec
  L7 Total Received:     0 Bytes
  L7 Total Sent:         0 Bytes
  L7 Total SSL Received: 0 Bytes
  L7 Total SSL Sent:     0 Bytes
  Avg Read Size:         0 Bytes
    1 Min Average:       0 Bytes
    5 Min Average:       0 Bytes
  Avg Write Size:        0 Bytes
    1 Min Average:       0 Bytes
    5 Min Average:       0 Bytes
```

Related

Virtual IP Mode Commands

Show Virtual Server Commands

Use the following commands display information about your virtual servers.

show virtual-server

Use

Use to view information about existing virtual servers.

For the **`show virtual-server brief`** command, if you see the Q-size column increasing over time, it may mean that you need additional web servers to handle the client connections.

Default Setting

None

Command Mode

exec

Syntax

`show virtual-server <vs_name>`

Show information about a virtual server configuration and status, as well of its virtual IPs and real servers

`show virtual-server brief`

Show summary of virtual server status

Parameter	Type	Description
vs_name	Word	Name of the virtual server.

Example

```
show virtual-server websilo1
Configuration
  LB Algorithm:                   round-robin-free
      Service Type:                  http
      Real Server Groups:
        Name                         Weight
        rsgroup_websilo1             1 (default)
      Real Servers:
        <none>
      Virtual IPs:
        Name                         Address            Port Svc  Admin
        vipweb1                      192.0.2.1          80   http online
    HTTP Configuration
      TCP Multiplex:                 0
      Client IP Header:              <none>
      Forward Connect Req:           0
      Session Persistence:           Cookie (lrscookie)  Expires: 1800s
      Strip Headers:
        <none>
      Hostnames:
        <none>
      Request Rate Limit:            <UNSET>reqs/sec
      Request Rate Limit Burst Size: <auto>

show virtual-server brief
Name       Svc   Req/sec  Q-size
--------------------------------
websilo1  http  0        0
```

Related

Virtual Server Mode Commands

show virtual-server statistics

Use

Use to display a snapshot of various statistics about virtual server traffic. The statistics show both the initiation and termination sides of the connections.

Default Setting

None

Command Mode

exec

Syntax

show virtual-server <vs_name> statistics

Shows statistics related to a virtual server

show virtual-server <vs_name> statistics detailed

Shows all detailed statistics related to a virtual-server

show virtual-server <vs_name> statistics http-requests

Shows virtual-server http request statistics

show virtual-server <vs_name> statistics http-response-codes

Shows virtual-server http response-code statistics

show virtual-server <vs_name> statistics http-responses

Shows virtual-server http response statistics

show virtual-server <vs_name> statistics latency

Shows virtual-server latency statistics

Examples

```
how virtual-server websilo1 statistics
HTTP Requests (Termination Side)
  Total:                          0
  Request Rate:                   0 requests/sec
  Total errors:                   0
HTTP Responses (Termination Side)
  Total:                          0
  Timed Out:                      0
  Idle Timed Out:                 0
  LB Forbidden:                   0
  LB 5xx Error:                   0
Client Latency
  Transaction:                    0 s
HTTP Requests (Initiation Side)
  Request Rate:                   0 requests/sec
  Total:                          0
  Total Queue Size:               0
HTTP Responses (Initiation Side)
  Total:                          0
  Errored:                        0
Server Latency
  Transaction:                    0 s
Connect Tunnels
  Open:                           0
  Opened Rate:                    0 tunnels/sec
```

show virtual-server websilo1 statistics http-requests

```
HTTP Requests (Termination Side)
  Total:                          0
  Request Rate:                   0 requests/sec
    1 Min Average:                0 requests/sec
    5 Min Average:                0 requests/sec
  Good:                           0
  Timed Out:                      0
  Total errors:                   0
HTTP Requests (Initiation Side)
  Request Rate:                 0 requests/sec
    1 Min Average:              0 requests/sec
    5 Min Average:              0 requests/sec
  Total:                        0
  Abandoned:                    0
  Total Queue Size:             0
    1 Min Average:              0
    5 Min Average:              0
  Request Mgr Q Size:           0
    1 Min Average:              0
    5 Min Average:              0
  Server Q Size:                0
    1 Min Average:              0
    5 Min Average:              0
  Timed Out:                    0
```

show virtual-server websilo1 statistics http-response-codes

```
HTTP Response Codes (Initiation Side)  Count  % of total count
  100:                                 0      0
  101:                                 0      0
  102-199:                             0      0
  200:                                 0      0
  201:                                 0      0
  202:                                 0      0
  203:                                 0      0
  204:                                 0      0
  205:                                 0      0
  206:                                 0      0
  207-299:                             0      0
  300:                                 0      0
  301:                                 0      0
  302:                                 0      0
  303:                                 0      0
  304:                                 0      0
  305:                                 0      0
  306:                                 0      0
  307:                                 0      0
  308-399:                             0      0
```

show virtual-server websilo1 statistics http-responses

```
HTTP Responses (Termination Side)
  Total:                          0
  Timed Out:                      0
```

```
  Idle Timed Out:                 0
  LB Forbidden:                   0
  LB 5xx Error:                   0
  LB 5xx Error Rate:              0 errors/sec
    1 Min Average:                0 errors/sec
    5 Min Average:                0 errors/sec
HTTP Responses (Initiation Side)
  Response Rate:                  0 responses/sec
    1 Min Average:                0 responses/sec
    5 Min Average:                0 responses/sec
  Total:                          0
  Good:                           0
  Errored:                        0
  Extraneous:                     0
  Abandoned:                      0
  Timed Out:                      0
  Idle Timed Out:                 0
  Error receiving body:           0
```

show virtual-server websilo1 statistics latency

```
Client Latency
  Transaction:     0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Request:         0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Initial:         0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Response:        0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
Server Latency
  Transaction:     0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Request:         0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Initial:         0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
  Response:        0 s
    1 Min Average: 0 s
    5 Min Average: 0 s
```

Related

Virtual Server Mode Commands

Terminal Mode Commands

terminal

Set terminal parameters.

Use

Use set the number of lines per screen in your terminal window for the current CLI session.

Auto setting attempts to change the number of lines when you change the terminal window size. If this does not work the way you want, you can set it to a specific number of lines.

Use **`terminal length 0`** when sending screen output to a file; the output has no limit.

Default Setting

auto

Command Mode

exec

Syntax

`terminal length <termlen>`

Set the number of lines on a screen

`terminal length auto`

Try to automatically determine the number of lines per screen for the current terminal

Parameter	Type	Description
termlen	Integer	Number of lines on screen (0 for no pausing).

Upgrade Command

Upgrade the system software.

Use

Use to upgrade LROS software to a new version. The upgrade will retain all of your configuration and lets you roll back to a previously installed software version using either of the following:

- CLI command: **`boot system`**
- REST node: /config/system/boot/version

To subscribe to software release notifications, contact your sales account representative.

Download the upgrade file and save the upgrade image file, which has the extension `.upg.gz`, on the LineRate system or a web server on your intranet. For where to download the upgrade file, see Downloads.

 We recommend upgrading during a maintenance window. The upgrade process causes a system reload. During the reload, you will lose all connections to the LineRate system for at least a few minutes.

The upgrade command may also be used to install an earlier version of software (to "downgrade") in limited circumstances. If the earlier version of software is already installed, you should use the **`boot system`** command to switch to that version of software. If the earlier version of software is not installed, the LineRate only supports installing the maintenance release immediately prior to the currently running software version via the upgrade process. To find the prior version of software that can be used with the upgrade process, see the release notes for your currently running software version.

Default Setting

None

Command Mode

exec

Syntax

`upgrade`

Parameter	Type	Description
img	String	URI of new LROS image file. Enclose URIs in quotes (for example, "file:///home/linerate/LROS-version.upg.gz" or "http://10.1.1.1/software/LROS-version.upg.gz").

Examples

`upgrade "LROS-1.5.3-R-x64.upg.gz"`

This command performs an upgrade using the image LROS-1.5.3-R-x64.upg.gz, which is in the default directory, /home/linerate on the LineRate system. That upgrade file must already be copied onto the LineRate system.

`upgrade "file:///home/linerate/tmp/LROS-1.5.3-R-x64.upg.gz"`

This command performs an upgrade using the image LROS-1.5.3-R-x64.upg.gz, which is in the directory /home/linerate/tmp on the LineRate system. That upgrade file must already be copied onto the LineRate system.

`upgrade "http://10.1.1.1/lros_images/LROS-1.5.3-R-x64.upg.gz"`

This command performs an upgrade using the image file LROS-1.5.3-R-x64.upg.gz, which is stored on the intranet web server at 10.1.1.1 in the /lros_images directory.

Procedure

If you are upgrading from version 2.1.0-BETA2 to version 2.2.0, see the note and the steps below the note.

To upgrade LineRate:

1. Back up your configuration using the backup command.
   - See Backup Mode Commands.
2. Download the upgrade file from https://linerate.f5.com/downloads to home/linerate on your LineRate system.
   - You may see several files available. Be sure to download the version from the email you received. The upgrade file ends in `.upg.gz`.
   - To copy the image from a system on your local network to the LineRate system, use a command similar to the following:
     `scp <local_image.upg.gz> admin@<LineRate_ip>:/home/linerate/.</local_image.upg.gz>`
3. From the LineRate system, type:
   `upgrade`

- Don't forget to enclose the image name in double quotes.
- The system tells you that the upgrade requires a reload when complete and asks if you want to continue

4. Type:
   yes
   - If you have not saved your running config, the system asks if you want to save it.
5. To save your configuration, type:
   yes
   - The upgrade installs. This may take a few minutes. If you are downloading the file and installing the upgrade, it may take longer. The system reloads after the upgrade. When the login prompt displays, the upgrade is complete.
   - After the system comes back up and you log in, you can use the following command to verify the version:
     show version

Related

REST API Reference - upgrade

Upgrading from version 2.1.0-BETA2 to version 2.2.0

Note: Because of the bug LRS-18273, when upgrading from version 2.1.0-BETA2 to version 2.2.0, be sure to follow the steps below.

To upgrade from version 2.1.0-BETA2 to version 2.2.0:

1. Save the config by typing:
   write
2. Go to bash mode by typing:
   bash
3. Edit /home/linerate/data/startup-config and update the "feature base" and "signing-certificate" subcommands of "licensing" such that "quit" is in a new line by itself, as shown below.

   Original config section:

   licensing
   signing-certificate phsigner
   ----BEGIN CERTIFICATE----
   …
   …
   ----END CERTIFICATE----quit
   feature base
   MIME-Version: 1.0
   …
   …
   ------23E33EA9EF57D387D8C212A25E875753—quit
   !

Edited config section:

licensing
 signing-certificate phsigner
----BEGIN CERTIFICATE----
…
…
----END CERTIFICATE----
quit
 feature base
MIME-Version: 1.0
…
…
------23E33EA9EF57D387D8C212A25E875753—
quit
!

4. Save the file and exit back to lros_shell by typing:
   `exit`
5. Upgrade the system as indicated in the Procedure above, but **do not** save the running configuration when prompted.

Write Command

Use

Use to save the running config to the startup config. See Running Config and Startup Config.

Default Setting

None

Command Mode

exec

Syntax

`write`

Saves the running config to the startup config.

`write memory`

This form of the command does the same thing and is available for compatibility if you are accustomed to other network operating systems.