



CHAPTER 1

Configuring Traffic Policies for Server Load Balancing

This chapter describes how to configure the Cisco Application Control Engine (ACE) module to use classification (class) maps and policy maps to:

- Filter and match interesting network traffic based on various criteria
- Load-balance that traffic to real servers in server farms using one of the ACE load-balancing predictor methods

It contains the following major sections:

- [Overview of SLB Traffic Policies](#)
- [Layer 7 SLB Traffic Policy Configuration Quick Start](#)
- [Layer 3 and Layer 4 SLB Traffic Policy Configuration Quick Start](#)
- [Configuring Layer 7 Class Maps for SLB](#)
- [Configuring a Layer 7 Policy Map for SLB](#)
- [Configuring an HTTP Parameter Map](#)
- [Configuring a Layer 3 and Layer 4 Class Map for SLB](#)
- [Configuring a Layer 3 and Layer 4 Policy Map for SLB](#)
- [Applying a Layer 3 and Layer 4 Policy to an Interface](#)
- [Displaying Load-Balancing Configurational Information and Statistics](#)



Note

For details about configuring network address translation (NAT), see the *Cisco Application Control Engine Module Security Configuration Guide*.

Overview of SLB Traffic Policies

You classify inbound network traffic destined to, or passing through, the ACE based on a series of flow match criteria specified by a class map. Each class map defines a traffic classification; network traffic that is of interest to you. A policy map defines a series of actions (functions) that you want applied to a set of classified inbound traffic.

ACE traffic policies support the following server load-balancing (SLB) traffic attributes:

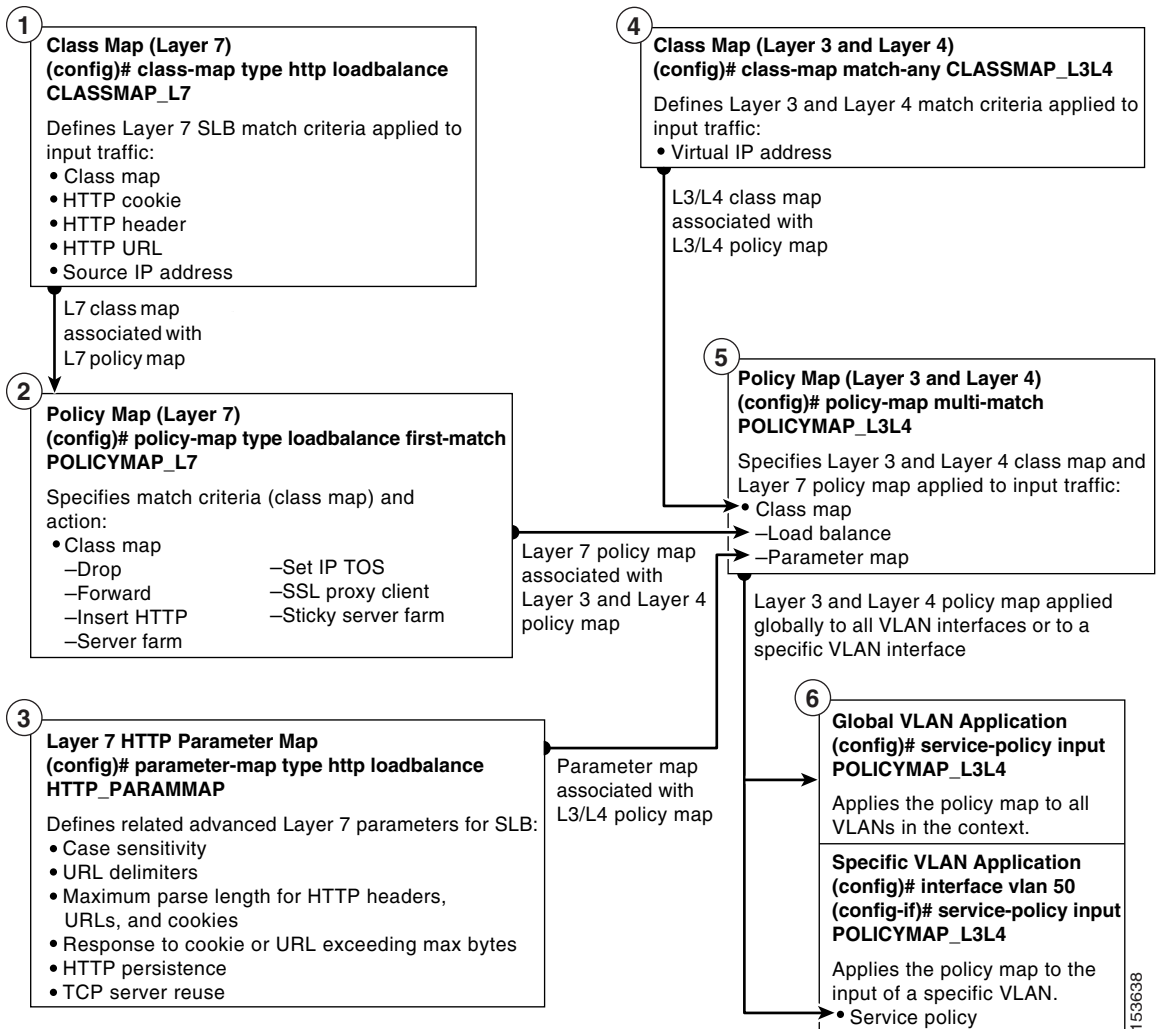
- **Layer 3 and Layer 4 connection information**—Source or destination IP address, source or destination port, virtual IP address, and IP protocol
- **Layer 7 protocol information**—HTTP cookie, HTTP URL, HTTP header, and SSL

The three steps in the traffic classification process consist of:

1. Creating a class map using the **class-map** command and the associated **match** commands, which comprise a set of match criteria related to Layer 3 and Layer 4 traffic classifications or Layer 7 protocol classifications.
2. Creating a policy map using the **policy-map** command, which refers to the class maps and identifies a series of actions to perform based on the traffic match criteria.
3. Activating the policy map by associating it with a specific VLAN interface or globally with all VLAN interfaces using the **service-policy** command as the means to filter traffic received by the ACE.

The flow chart shown in [Figure 1-1](#) provides a basic overview of the process required to build and apply the Layer 3, Layer 4, and Layer 7 policies that the ACE uses for SLB. The flow chart also illustrates how you associate the various components of the SLB policy configuration with each other.

Figure 1-1 Server Load-Balancing Configuration Flow Diagram



153638

Layer 7 SLB Traffic Policy Configuration Quick Start

[Table 1-1](#) provides a quick overview of the steps required to configure a Layer 7 class map and a Layer 7 policy map. Each step includes the CLI command and a reference to the procedure required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the sections following [Table 1-1](#).

Table 1-1 Layer 7 SLB Policy Configuration Quick Start

Task and Command Example

1. If you are operating in multiple contexts, observe the CLI prompt to verify that you are operating in the desired context. If necessary, change to, or directly log in to, the correct context.

```
host1/Admin# changeto c1
host1/C1#
```

The rest of the examples in this table use the Admin context for illustration purposes, unless otherwise specified. For details on creating contexts, refer to the *Cisco Application Control Engine Module Administration Guide*.

2. Enter configuration mode.

```
host1/Admin# config
Enter configuration commands, one per line. End with CNTL/Z
host1/Admin(config)#
```

3. Create a Layer 7 class map for SLB. See the “[Configuring Layer 7 Class Maps for SLB](#)” section.

```
host1/Admin(config)# class-map type http loadbalance match-all
L7SLBCLASS
host1/Admin(config-cmap-http-lb)#
```

Table 1-1 Layer 7 SLB Policy Configuration Quick Start (continued)

Task and Command Example

4. Configure one or more of the following match criteria for the Layer 7 SLB class map:

- Define a cookie for HTTP load balancing. See the “[Defining a Cookie for HTTP Load Balancing](#)” section.

```
host1/Admin(config-cmap-http-lb)# match http cookie  
TESTCOOKIE1 cookie-value 123456
```

- Define an HTTP header for load balancing. See the “[Defining an HTTP Header for Load Balancing](#)” section.

```
host1/Admin(config-cmap-http-lb)# match http header Host  
header-value .*cisco.com
```

- Define a URL for HTTP load balancing. See the “[Defining a URL for HTTP Load Balancing](#)” section.

```
host1/Admin(config-cmap-http-lb)# match http url  
/WHATSNEW/LATEST.*
```

- Define a source IP match statement. See the “[Defining Source IP Address Match Criteria](#)” section.

```
host1/Admin(config-cmap-http-lb)# match source-address  
192.168.11.2 255.255.255.0
```

-
5. Use the **exit** command to reenter configuration mode.

```
host1/Admin(config-cmap-http-lb)# exit  
host1/Admin(config)#
```

-
6. Create a Layer 7 policy map for SLB. See the “[Configuring a Layer 7 Policy Map for SLB](#)” section.

```
host1/Admin(config)# policy-map type loadbalance first-match  
L7SLBPOLICY  
host1/Admin(config-pmap-lb)#
```

-
7. Associate the Layer 7 class map that you created in step 2 with the Layer 7 policy map that you created in step 5. See the “[Associating a Layer 7 Class Map with a Layer 7 Policy Map](#)” section.

```
host1/Admin(config-pmap-lb)# class L7SLBCLASS  
host1/Admin(config-pmap-lb-c)#
```

Table 1-1 Layer 7 SLB Policy Configuration Quick Start (continued)**Task and Command Example**

8. Specify one or more of the following policy-map actions that you want the ACE to take when network traffic matches a class map:

- Instruct the ACE to discard packets that match a policy map. See the [“Discarding Requests”](#) section.

```
host1/Admin(config-pmap-lb-c) # drop
```

- Instruct the ACE to forward packets that match a policy map without load balancing them. [“Forwarding Requests Without Load Balancing”](#)

```
host1/Admin(config-pmap-lb-c) # forward
```

- Enable HTTP header insertion. See the [“Configuring HTTP Header Insertion”](#) section.

```
host1/Admin(config-pmap-lb-c) # insert-http Host header-value
www.cisco.com
```

- Enable load balancing to a server farm. See the [“Enabling Load Balancing to a Server Farm \(Configuring a Backup Server\)”](#) section.

```
host1/Admin(config-pmap-lb-c) # serverfarm FARM2 backup FARM3
sticky
```

- Specify the IP differentiated services code point (DSCP) of packets within the traffic class. See the [“Configuring a Sticky Server Farm”](#) section.

```
host1/Admin(config-pmap-lb-c) # set ip tos 8
```

- If you are using SSL Initiation (ACE acting as an SSL client), specify an SSL proxy service. See the [“Specifying an SSL Proxy Service”](#) section. For more information about SSL, refer to the *Cisco Application Control Engine Module SSL Configuration Guide*.

```
host1/Admin(config-pmap-lb-c) # ssl-proxy client
PROXY_SERVICE1
```

- If you want to use stickiness (connection persistence), specify a sticky server farm for load balancing. See the [“Configuring a Sticky Server Farm”](#) section.

```
host1/Admin(config-pmap-lb-c) # sticky-serverfarm
STICKY_GROUP1
```

Table 1-1 Layer 7 SLB Policy Configuration Quick Start (continued)

Task and Command Example

9. Before you can use a Layer 7 policy map for load balancing, you must associate it with a Layer 3 and Layer 4 SLB policy map. Create the Layer 3 and Layer 4 class map and policy map, then associate the Layer 7 policy map with the Layer 3 and Layer 4 policy map. Finally, associate the Layer 3 and Layer 4 policy map with an interface. See the following sections:

- [Configuring a Layer 3 and Layer 4 Class Map for SLB](#)
- [Configuring a Layer 3 and Layer 4 Policy Map for SLB](#)
- [Applying a Layer 3 and Layer 4 Policy to an Interface](#)

-
10. (Recommended) Use the following commands to display your class-map and policy-map configurations and statistics (see the “[Displaying Load-Balancing Configurational Information and Statistics](#)” section):

```
host1/Admin# show running-config class-map
host1/Admin# show running-config policy-map
```

-
11. (Optional) If necessary, save your configuration changes to Flash memory.

```
host1/Admin# copy running-config startup-config
```

Layer 3 and Layer 4 SLB Traffic Policy Configuration Quick Start

[Table 1-2](#) provides a quick overview of the steps required to configure a Layer 3 and Layer 4 class map and a Layer 3/4 policy map. Each step includes the CLI command and a reference to the procedure required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the sections following [Table 1-2](#).

Table 1-2 Layer 3 and Layer 4 SLB Policy Configuration Quick Start

Task and Command Example

1. If you are operating in multiple contexts, observe the CLI prompt to verify you are operating in the desired context. Change to, or directly log in to, the correct context if necessary.

```
host1/Admin# changeto C1
host1/C1#
```

For details on creating contexts, refer to the *Cisco Application Control Engine Module Virtualization Configuration Guide*.

2. Enter configuration mode.

```
host1/Admin# config
Enter configuration commands, one per line. End with CNTL/Z
host1/Admin(config)#
```

3. Create a Layer 3 and Layer 4 SLB class map. See the “[Configuring a Layer 3 and Layer 4 Class Map for SLB](#)” section.

```
host1/Admin(config)# class-map L4VIPCLASS
host1/Admin(config-cmap)#
```

4. Define a virtual IP (VIP) address match statement. See the “[Defining VIP Address Match Criteria](#)” section.

```
host1/Admin(config-cmap)# match virtual-address 192.168.1.10 tcp
port eq 80
```

5. Use the **exit** command to reenter configuration mode.

```
host1/Admin(config-cmap)# exit
host1/Admin(config)#
```

Table 1-2 Layer 3 and Layer 4 SLB Policy Configuration Quick Start (continued)**Task and Command Example**

6. Create a Layer 3 and Layer 4 policy map. See the “[Configuring a Layer 3 and Layer 4 Policy Map for SLB](#)” section.

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY
host1/Admin(config-pmap)#
```

7. Associate the Layer 3 and Layer 4 class map that you created in step 2 with the policy map you created in step 4. See the “[Associating a Layer 3 and Layer 4 Class Map with a Policy Map](#)” section.

```
host1/Admin(config-pmap)# class L4VIPCLASS
host1/Admin(config-pmap-c)#
```

8. Specify one or more of the following policy-map actions that you want the ACE to take when network traffic matches a class map:

- Enable the ACE to advertise the IP address of a virtual server as the host route. See the “[Enabling the Advertising of a Virtual Server IP Address](#)” section.

```
host1/Admin(config-pmap-c)# loadbalance vip advertise active frequency 30
```

- Enable a VIP to reply to ICMP ECHO requests. For example, if a user sends an ICMP ECHO request to a VIP, this command instructs the VIP to send an ICMP ECHO-REPLY. See the “[Enabling a VIP to Reply to ICMP Requests](#)” section.

```
host1/Admin(config-pmap-c)# loadbalance vip icmp-reply
```

- Associate a Layer 7 SLB policy map with the Layer 3 and Layer 4 policy map to provide an entry point for Layer 7 classifications. See the “[Associating a Layer 7 SLB Policy Map with a Layer 3 and Layer 4 SLB Policy Map](#)” section.

```
host1/Admin(config-pmap-c)# loadbalance policy L7SLBPOLICY
```

- Enable a VIP for SLB operations.

```
host1/Admin(config-pmap-c)# loadbalance vip inservice
```

Table 1-2 Layer 3 and Layer 4 SLB Policy Configuration Quick Start (continued)

Task and Command Example

9. Activate a policy map and attach it to an interface. See the [“Applying a Layer 3 and Layer 4 Policy to an Interface”](#) section.

```
host1/Admin(config)# interface VLAN50
host1/Admin(config-if)# service-policy input L4SLBPOLICY
host1/Admin(config-if)# Ctrl-z
```

10. (Recommended) Use the following commands to display your class-map and policy-map configurations and statistics (see the [“Displaying Load-Balancing Configurational Information and Statistics”](#) section):

```
host1/Admin# show running-config class-map
host1/Admin# show running-config policy-map
host1/Admin# show service-policy name [detail]
```

11. (Optional) If necessary, save your configuration changes to Flash memory.

```
host1/Admin# copy running-config startup-config
```

Configuring Layer 7 Class Maps for SLB

A Layer 7 SLB class map contains match criteria that classify specific Layer 7 network traffic. This section describes how to create a class map for Layer 7 SLB based on HTTP cookies, HTTP headers, HTTP URLs, source IP addresses, or SSL ciphers.

To create a Layer 7 class map for SLB and enter the class-map configuration mode, use the **class-map type http loadbalance** command in configuration mode. The syntax of this command is:

```
class-map type http loadbalance [match-all | match-any] map_name
```

You can configure multiple **match** commands in a single class map to specify the matching criteria. For example, you can configure a Layer 7 load-balancing class map to define multiple URLs, cookies, and HTTP headers in a group that you then associate with a traffic policy. The **match-all** and **match-any** keywords determine how the ACE evaluates multiple match statement operations when multiple match criteria exist in a class map.

The keywords, arguments, and options are:

- **http loadbalance**—Specifies a Layer 7 load-balancing class map.
- **match-all** | **match-any**—(Optional) Determines how the ACE evaluates Layer 7 HTTP SLB operations when multiple match criteria exist in a class map. The class map is considered a match if the match commands meet one of the following conditions:
 - **match-all** —(Default) Network traffic needs to satisfy all of the match criteria (implicit AND) to match the Layer 7 load-balancing class map. The **match-all** keyword is applicable only for match statements of different Layer 7 load-balancing types. For example, specifying a **match-all** condition for URL, HTTP header, and URL cookie statements in the same class map is valid. However, specifying a **match-all** condition for multiple HTTP headers or multiple cookies with the same names or multiple URLs in the same class map is invalid.

- **match-any**—Network traffic needs to satisfy only one of the match criteria (implicit OR) to match the HTTP load-balancing class map. The **match-any** keyword is applicable only for match statements of the same Layer 7 load-balancing type. For example, the ACE does not allow you to specify a **match-any** condition for URL, HTTP header, and URL cookie statements in the same class map but does allow you to specify a **match-any** condition for multiple URLs, or multiple HTTP headers or multiple cookies with different names in the same class map.
- *map_name*—Unique identifier assigned to the class map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. The class-map name is used for both the class map and to associate the class map with a policy map.

For example, to create a Layer 7 load-balancing class map named L7SLBCLASS, enter:

```
host1/Admin(config)# class-map type http loadbalance match-any  
L7SLBCLASS  
host1/Admin(config-cmap-http-lb)#
```

To remove a Layer 7 load-balancing class map from the configuration, enter:

```
host1/Admin(config)# no class-map type http loadbalance match-any  
L7SLBCLASS
```

Continue with the following sections to specify match criteria for the Layer 7 class map:

- [Configurational Considerations](#)
- [Defining a Cookie for HTTP Load Balancing](#)
- [Defining an HTTP Header for Load Balancing](#)
- [Defining a URL for HTTP Load Balancing](#)
- [Defining Source IP Address Match Criteria](#)
- [Nesting Layer 7 HTTP SLB Class Maps](#)

Configurational Considerations

When you are creating a class map for SLB, note the following configurational considerations:

- There can be a maximum of 10 total cookie names and header names for each Layer 3 and Layer 4 class. You can allocate the number of cookie names and header names in any combination as long as you do not exceed the maximum of 10.
- You can associate a maximum of 1024 instances of the same type of regex with each Layer 3 and Layer 4 policy map. This limit applies to all Layer 7 policy-map types, including generic, HTTP, RADIUS, RDP, RTSP, and SIP. You configure regexes in:
 - Match statements in Layer 7 class maps
 - Inline match statements in Layer 7 policy maps
 - Layer 7 hash predictors for server farms
 - Layer 7 sticky expressions in sticky groups
 - Header insertion and rewrite (including SSL URL rewrite) expressions in Layer 7 action lists
- The ACE restricts the nesting of class maps to two levels to prevent you from including one nested class map in a different class map.
- The maximum number of class maps for each ACE is 8192.

Defining a Cookie for HTTP Load Balancing

The ACE performs regular expression matching against the received packet data from a particular connection based on the cookie expression. You can configure a maximum of 10 total cookie names and header names per class in any combination. To configure the class map to make Layer 7 SLB decisions based on the name and string of a cookie, use the **match http cookie** command in class-map configuration mode. The syntax of this command is:

```
[line_number] match http cookie {name | secondary name} cookie-value  
expression
```

The keywords, arguments, and options are:

- *line_number*—(Optional) Use line numbers only for editing or deleting the individual **match** commands. For example, you can enter **no line_number** to delete long **match** commands instead of entering the entire line. The sequence numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.
- *name*—Specifies a unique cookie name. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.



Note If certain characters are used in the cookie name, such as an underscore (`_`), hyphen (`-`), period (`.`), or semicolon (`:`), replace those characters with the equivalent percent encoding (`% HEX HEX`) characters. For example, to configure the cookie name `Regex_MatchCookie`, replace the underscore (`_`) character with the equivalent `%5F` percent encoding character and enter the cookie name as **Regex%5FMatchCookie** in the CLI.

- **secondary name**—Specifies a cookie in a URL string. You can specify the delimiters for cookies in a URL string using a command in an HTTP parameter map. For more information, see the “[Defining URL Delimiters](#)” section.
- **cookie-value expression**—Specifies a unique cookie value expression. Enter an unquoted text string with no spaces and a maximum of 255 alphanumeric characters. Alternatively, you can enter a text string with spaces provided that you enclose the entire string in quotation marks (“”). The ACE supports the use of regular expressions for matching string expressions. [Table 1-3](#) lists the supported characters that you can use for matching string expressions.

Table 1-3 Special Characters for Matching String Expressions

| Convention | Description |
|------------|--|
| . | One of any character. |
| .* | Zero or more of any character. |
| \. | Period (escaped). |
| [charset] | Match any single character from the range. |

Table 1-3 Special Characters for Matching String Expressions (continued)

| Convention | Description |
|-----------------|--|
| [^charset] | Do not match any character in the range. All other characters represent themselves. |
| () | Expression grouping. |
| (expr1 expr2) | OR of expressions. |
| (expr)* | 0 or more of expression. |
| (expr)+ | 1 or more of expression. |
| expr{m,n} | Repeat the expression between <i>m</i> and <i>n</i> times, where <i>m</i> and <i>n</i> have a range of 1 to 255. |
| expr{m} | Match the expression exactly <i>m</i> times. The range for <i>m</i> is from 1 to 255. |
| expr{m,} | Match the expression <i>m</i> or more times. The range for <i>m</i> is from 1 to 255. |
| \a | Alert (ASCII 7). |
| \b | Backspace (ASCII 8). |
| \f | Form-feed (ASCII 12). |
| \n | New line (ascii 10). |
| \r | Carriage return (ASCII 13). |
| \t | Tab (ASCII 9). |
| \v | Vertical tab (ASCII 11). |
| \0 | Null (ASCII 0). |
| \\ | Backslash. |
| \x## | Any ASCII character as specified in two-digit hexadecimal notation. |

The following example specifies that the Layer 7 class map load balance on a cookie with the name of testcookie1.

```
host1/Admin(config)# class-map type http loadbalance match-any
L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 100 match http cookie testcookie1
cookie-value 123456
```

To remove an HTTP cookie match statement from the class map, enter:

```
host1/Admin(config-cmap-http-lb)# no 100
```

Defining an HTTP Header for Load Balancing

The ACE performs regular expression matching against the received packet data from a particular connection based on the HTTP header expression. You can configure a maximum of 10 total HTTP header names and cookie names per class. To configure a class map to make Layer 7 SLB decisions based on the name and value of an HTTP header, use the **match http header** command in class-map configuration mode.

The syntax of this command is:

```
[line_number] match http header name header-value expression
```

The keywords, arguments, and options are:

- *line_number*—(Optional) Use line numbers only for editing or deleting the individual **match** commands. For example, you can enter **no** *line_number* to delete long **match** commands instead of entering the entire line. The sequence numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.
- *name*—Specifies the name of the generic field in the HTTP header. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. Alternatively, you can enter a text string with spaces provided that you enclose the entire string in quotation marks (“”). For a list of standard header field names, see [Table 1-4](#).
- **header-value** *expression*—Specifies the header value expression string to compare against the value in the specified field in the HTTP header. Enter a text string with a maximum of 255 alphanumeric characters. The ACE supports the use of regular expressions for header matching. Expressions are stored in a header map in the form *header-name: expression*. Header expressions allow spaces, provided that the entire string that contains spaces is quoted. If you use a **match-all** class map, all headers in the header map must be matched. See [Table 1-3](#) for a list of the supported characters that you can use in regular expressions.

Table 1-4 Standard HTTP Header Fields

| Field Name | Description |
|------------------------|---|
| Accept | A semicolon-separated list of representation schemes (content type metainformation values) that will be accepted in the response to the request. |
| Accept-Charset | The character sets are acceptable for the response. This field allows clients capable of understanding more comprehensive or special-purpose character sets to signal that capability to a server that can represent documents in those character sets. |
| Accept-Encoding | Restricts the content encoding that a user will accept from the server. |
| Accept-Language | The ISO code for the language in which the document is written. The language code is an ISO 3316 language code with an optional ISO639 country code to specify a national variant. |
| Authorization | Specifies that the user agent wants to authenticate itself with a server, usually after receiving a 401 response. |
| Cache-Control | Directives that must be obeyed by all caching mechanisms along the request/response chain. The directives specify behavior intended to prevent caches from adversely interfering with the request or response. |
| Connection | Allows the sender to specify connection options. |
| Content-MD5 | An MD5 digest of the entity-body that provides an end-to-end integrity check. Only a client or an origin server can generate this header field. |
| Expect | Used by a client to inform the server about what behaviors the client requires. |
| From | Contains the e-mail address of the person that controls the requesting user agent. |

Table 1-4 Standard HTTP Header Fields (continued)

| Field Name | Description |
|--------------------------|---|
| Host | The Internet host and port number of the resource being requested, as obtained from the original URI given by the user or referring resource. The Host field value MUST represent the naming authority of the origin server or gateway given by the original URL. |
| If-Match | Used with a method to make it conditional. A client that has one or more entities previously obtained from the resource can verify that one of those entities is current by including a list of their associated entity tags in the If-Match header field. The purpose of this feature is to allow efficient updates of cached information with a minimum amount of transaction overhead. It is also used, on updating requests, to prevent inadvertent modification of the wrong version of a resource. As a special case, the value “*” matches any current entity of the resource. |
| Pragma | Pragma directives understood by servers to whom the directives are relevant. The syntax is the same as for other multiple-value fields in HTTP, for example, the accept field, a comma-separated list of entries, for which the optional parameters are separated by semicolons. |
| Referer | The address (URI) of the resource from which the URI in the request was obtained. |
| Transfer-Encoding | Indicates what (if any) type of transformation has been applied to the message body in order to safely transfer it between the sender and the recipient. |

Table 1-4 Standard HTTP Header Fields (continued)

| Field Name | Description |
|------------|--|
| User-Agent | Information about the user agent, for example a software program originating the request. This information is for statistical purposes, the tracing of protocol violations, and automated recognition of user agents for the sake of tailoring responses to avoid particular user agent limitations. |
| Via | Used by gateways and proxies to indicate the intermediate protocols and recipients between the user agent and the server on requests, and between the origin server and the client on responses. |

The following example specifies that the Layer 7 class map load balance on an HTTP header named Host:

```
host1/Admin(config)# class-map type http loadbalance match-any
L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 100 match http header Host
header-value .*cisco.com
```

The following example uses regular expressions in a class map to emulate a wildcard search to match the header value expression string:

```
host1/Admin(config)# class-map type http loadbalance match-any
L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 10 match http header Host
header-value .*cisco.com
host1/Admin(config-cmap-http-lb)# 20 match http header Host
header-value .*yahoo.com
```

The following example specifies that the Layer 7 class map load balance on an HTTP header named Via:

```
host1/Admin(config)# class-map type http loadbalance match-any
L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 30 match http header Via
header-value 192.*
```

To remove all HTTP header match criteria from the L7SLBCLASS class map, enter:

```
host1/Admin(config-cmap-http-lb)# no 10
host1/Admin(config-cmap-http-lb)# no 20
host1/Admin(config-cmap-http-lb)# no 30
```

Defining a URL for HTTP Load Balancing

The ACE performs regular expression matching against the received packet data from a particular connection based on the HTTP URL string. To configure a class map to make Layer 7 SLB decisions based on URL name and, optionally, HTTP method, use the **match http url** command in class-map configuration mode. The syntax of this command is:

```
[line_number] match http url expression [method name]
```

The keywords, arguments, and options are:

- *line_number*—(Optional) Use line numbers only for editing or deleting the individual **match** commands. For example, you can enter **no line_number** to delete long **match** commands instead of entering the entire line. The *line_number* line numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.
- *expression*—Specifies the URL, or portion of a URL, to match. Enter a URL string from 1 to 255 alphanumeric characters. The ACE performs matching on whatever URL string appears after the HTTP method, regardless of whether the URL includes the host name. The ACE supports the use of regular expressions for matching URL strings. See [Table 1-3](#) for a list of the supported characters that you can use in regular expressions.



Note

When matching URLs, keep in mind that period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use the “[]” character classes to match these symbols (for example, enter “www[.]xyz[.]com” instead of “www.xyz.com”). You can also use a backslash (\) to escape a dot (.) or a question mark (?).

- **method *name***—(Optional) Specifies the HTTP method to match. Enter a method name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. The method can either be one of the standard HTTP 1.1 method names (OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, or CONNECT) or a text string that must be matched exactly (for example, CORVETTE).

The following example specifies that the Layer 7 class map load balance on a specific URL.

```
host1/Admin(config)# class-map type http loadbalance L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 10 match http url /whatsnew/latest.*
```

The following example uses regular expressions to emulate a wildcard search to match on any .gif or .html file.

```
host1/Admin(config)# class-map type http loadbalance match-any
L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 100 match http url *.*gif
host1/Admin(config-cmap-http-lb)# 200 match http url *.*html
```

To remove a URL match statement from the L7SLBCLASS class map, enter **no** and the line number. For example, to remove line 100, enter:

```
host1/Admin(config-cmap-http-lb)# no 100
```

**Note**

If you did not use line numbers to enter the original URL match statement, you can obtain the line number from your running configuration. To display the running configuration, enter **show running-config**.

Defining Source IP Address Match Criteria

To configure the class map to make Layer 7 SLB decisions based on a client source IP address, use the **match source-address** command in class-map configuration mode. If this command is the only match criteria in the class map, the ACE considers it to be a Layer 3 and Layer 4 class map.

The syntax of this command is:

```
[line_number] match source-address ip_address [netmask]
```

The arguments and options are:

- *line_number*—(Optional) Use line numbers only for editing or deleting the individual **match** commands. For example, you can enter **no line_number** to delete long **match** commands instead of entering the entire line. The line numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.
- *ip_address*—Source IP address of the client. Enter the IP address in dotted-decimal notation (for example, 192.168.11.2).
- *netmask*—(Optional) Subnet mask of the IP address. Enter the netmask in dotted-decimal notation (for example, 255.255.255.0). The default is 255.255.255.255.

The following example specifies that the class map match on source IP address 192.168.11.2 255.255.255.0.

```
host1/Admin(config)# class-map http type loadbalance match-any
L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 50 match source-address 192.168.11.2
255.255.255.0
```

To remove the source IP address match statement from the class map, enter:

```
host1/Admin(config-cmap-http-lb)# no 50
```

Nesting Layer 7 HTTP SLB Class Maps

The nesting of class maps allows you to achieve complex logical expressions for Layer 7 HTTP-based SLB. To identify one Layer 7 HTTP SLB class map that is to be used as a matching criterion for another Layer 7 class map, use the **match class-map** command in class-map configuration mode.



Note

The ACE restricts the nesting of class maps to two levels to prevent you from including a nested class map under another class map.

The syntax of this command is:

```
[line_number] match class-map map_name
```

The keywords, arguments, and options are:

- *line_number*—(Optional) Inclusion of a line number can assist you in editing or deleting the individual match commands. For example, you can enter **no line_number** to delete long match commands instead of entering the entire line.
- *map_name*—Specifies the name of an existing Layer 7 load-balancing class map.

The **match class-map** command allows you to combine the use of the **match-any** and **match-all** keywords in the same class map. To combine **match-all** and **match-any** characteristics in a class map, create a class map that uses one **match** command (either **match any** or **match all**) and then use this class map as a match statement in a second class map that uses a different match type.

For example, assume that commands A, B, C, and D represent separate match criteria, and you want Layer 7 HTTP traffic that matches A, B, or C and D (A or B or [C and D]) to satisfy the class map. Without the use of nested class maps, traffic would either have to match all four match criteria (A and B and C and D) or match any of the match criteria (A or B or C or D) to satisfy the class map. By creating a single class map that uses the **match-all** keyword for match criteria C and D (criteria E), you can then create a new **match-any** class map that uses match criteria A, B, and E. The new traffic class contains your desired classification sequence: A or B or E, which is equivalent to A or B or [C and D].

The following example shows how to combine the characteristics of two class maps, one with **match-any** and one with **match-all** characteristics, into a single class map by using the **match class-map** command.

```
host1/Admin(config)# class-map type http loadbalance match-any CLASS3
host1/Admin(config-cmap-http-lb)# 100 match http url .*gif
host1/Admin(config-cmap-http-lb)# 200 match http header Host
header-value XYZ
host1/Admin(config-cmap-http-lb)# exit

host1/Admin(config)# class-map type http loadbalance match-all CLASS4
host1/Admin(config-cmap-http-lb)# 10 match class-map CLASS3
host1/Admin(config-cmap-http-lb)# 20 match source-address 192.168.11.2
host1/Admin(config-cmap-http-lb)# exit
```

To remove the nested class map from the Layer 7 class map, enter:

```
host1/Admin(config-cmap-http-lb)# no 10
```

Configuring a Layer 7 Policy Map for SLB

To use a Layer 7 SLB policy map, first create the policy map and define **match** statements and policy actions. Because Layer 7 policy maps are child policies, you must then associate a Layer 7 policy map with the appropriate Layer 3 and Layer 4 policy map to provide an entry point for Layer 7 SLB traffic classification. You cannot directly apply a Layer 7 policy map on an interface; you can activate only a Layer 3 and Layer 4 policy map on an interface or globally on all interfaces in a context.

For background information on the role of policy maps in the ACE, refer to the *Cisco Application Control Engine Module Administration Guide*.

The ACE treats as a Layer 3 and Layer 4 policy any policy map that has only source IP configured as the match criteria in the class map or inline match, or that has the default class configured as the class map, and there are no Layer 7 policy actions configured.

To create a Layer 7 SLB policy map and enter policy-map configuration mode, use the **policy-map type** command in configuration mode. The syntax of this command is:

```
policy-map type loadbalance first-match map_name
```

The keywords and arguments are:

- **loadbalance first-match**—Specifies a policy map that defines Layer 7 HTTP SLB decisions. The **first-match** keyword defines the execution for the Layer 7 load-balancing policy-map. The ACE executes only the action specified against the first-matching classification.
- *map_name*—Specifies the identifier assigned to the policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

The following example illustrates how to create a Layer 7 SLB policy map:

```
host1/Admin(config)# policy-map type loadbalance first-match  
L7SLBPOLICY  
host1/Admin(config-pmap-lb)#
```

To remove a policy map from the ACE, enter:

```
host1/Admin(config)# no policy-map type loadbalance first-match  
L7SLBPOLICY
```

Continue with the following sections to enable the use of sequence numbers, define inline match statements, and define policy-map actions:

- [Defining Match Statements Inline in a Layer 7 Policy Map](#)
- [Associating a Layer 7 Class Map with a Layer 7 Policy Map](#)
- [Specifying Layer 7 SLB Policy Actions](#)
- [Associating a Layer 7 Policy Map with a Layer 3 and Layer 4 Policy Map](#)

Adding a Layer 7 Policy Map Description

Use the **description** command to provide a brief summary about the Layer 7 policy map.

Access the policy map configuration mode to specify the **description** command.

The syntax of this command is:

```
description text
```

Use the *text* argument to enter an unquoted text string with a maximum of 240 alphanumeric characters.

For example, to add a description that the policy map is to insert HTTP headers, enter:

```
host1/Admin(config-pmap-lb)# description insert HTTP headers
```

To remove the description from the policy map, enter:

```
host1/Admin(config-pmap-lb)# no description
```

Defining Match Statements Inline in a Layer 7 Policy Map

Layer 7 SLB policy maps allow you to enter a single inline SLB match criteria in the policy map without specifying a traffic class. The inline Layer 7 SLB policy map match commands function the same as with the Layer 7 SLB class map match commands. However, when you use an inline **match** command, you can specify an action for only a single match statement in the Layer 7 policy.



Note

To specify actions for multiple match statements, use a class map as described in the [“Associating a Layer 7 Class Map with a Layer 7 Policy Map”](#) section.

The syntax of an inline match command is:

```
match name1 match_statement [insert-before name2]
```

**Note**

You can associate a maximum of 1024 instances of the same type of regex with each Layer 3 and Layer 4 policy map. This limit applies to all Layer 7 policy-map types, including generic, HTTP, RADIUS, RDP, RTSP, and SIP. You configure regexes in:

- Match statements in Layer 7 class maps
 - Inline match statements in Layer 7 policy maps
 - Layer 7 hash predictors for server farms
 - Layer 7 sticky expressions in sticky groups
 - Header insertion and rewrite (including SSL URL rewrite) expressions in Layer 7 action lists
-

The arguments and options are:

- *name1*—Specifies the name assigned to the inline match command. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
- *match_statement*—Specifies the individual Layer 7 SLB match criteria.
- **insert-before** *name2*—(Optional) Places the current match statement ahead of an existing class map or other match statement specified by the *name2* argument in the policy-map configuration. The ACE does not save the sequence reordering as part of the configuration.

For information about configuring inline match statements in a Layer 7 SLB policy map, see the following sections in the “[Configuring Layer 7 Class Maps for SLB](#)” section:

- [Defining a Cookie for HTTP Load Balancing](#)
- [Defining an HTTP Header for Load Balancing](#)
- [Defining a URL for HTTP Load Balancing](#)
- [Defining Source IP Address Match Criteria](#)

**Note**

The *line_number* argument described in the above-referenced sections is only for use with **match** statements in class maps. Otherwise, the descriptions of **match** statements in Layer 7 class maps and inline **match** statements in Layer 7 policy maps are the same.

Associating a Layer 7 Class Map with a Layer 7 Policy Map

To associate an existing Layer 7 class map with a Layer 7 policy map, use the **class** command. The syntax of this command is:

```
class {name1 [insert-before name2] | class-default}
```

The keywords, arguments, and options are:

- **name1**—Specifies the name of a previously defined traffic class, configured with the **class-map** command, to associate traffic to the traffic policy. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
- **insert-before name2**—(Optional) Places the current class map ahead of an existing class map or match statement specified by the *name2* argument in the policy-map configuration. The ACE does not save the sequence reordering as part of the configuration.
- **class-default**—Reserved, well-known class map created by the ACE. You cannot delete or modify this class. All traffic that fails to meet the other matching criteria in the named class map belongs to the default traffic class. If none of the specified classifications match the traffic, then the ACE performs the action specified under the **class class-default** command. The **class-default** class map has an implicit **match any** statement in it enabling it to match all traffic.

The following example illustrates the use of the **insert-before** option to define the position of a class map in the policy map:

```
host1/Admin(config-pmap-lb)# class L7SLBCLASS insert-before http_class  
host1/Admin(config-pmap-lb-c)#
```

To remove a class map from a Layer 7 policy map, enter:

```
host1/Admin(config-pmap-lb)# no class L7SLBCLASS
```

The following example illustrates the use of the **class class-default** command.

```
host1/Admin(config-pmap-lb)# class L7SLBCLASS insert-before http_class
host1/Admin(config-pmap-lb-c)# exit
host1/Admin(config-pmap-lb)# class class-default
host1/Admin(config-pmap-lb-c)#
```

Specifying Layer 7 SLB Policy Actions

After you associate a Layer 7 SLB class map with a Layer 7 SLB policy map or specify inline **match** commands, you need to specify the actions that the ACE should take when network traffic matches a class map or inline **match** command. To specify the Layer 7 SLB policy actions, use the commands described in the following subsections:

- [Discarding Requests](#)
- [Forwarding Requests Without Load Balancing](#)
- [Configuring HTTP Header Insertion](#)
- [Enabling Load Balancing to a Server Farm \(Configuring a Backup Server\)](#)
- [Configuring a Sorry Server Farm](#)
- [Configuring a Sticky Server Farm](#)
- [Specifying the IP Differentiated Services Code Point of Packets](#)
- [Specifying an SSL Proxy Service](#)

Discarding Requests

To instruct the ACE to discard packets that match a particular policy map, use the drop command in load-balancing policy-map class configuration mode. The syntax of this command is:

```
drop
```

For example, enter:

```
host1/Admin(config-pmap-lb-c)# drop
```

To reset the behavior of the ACE to the default of accepting packets that match a policy map, enter:

```
host1/Admin(config-pmap-lb-c)# no drop
```

Forwarding Requests Without Load Balancing

To instruct the ACE to forward requests that match a particular policy map without performing loadbalancing on the request, use the **forward** command in load-balancing policy-map class configuration mode. The syntax of this command is:

```
forward
```

For example, enter:

```
host1/Admin(config-pmap-lb-c) # forward
```

To reset the behavior of the ACE to the default of loadbalancing packets that match a policy map, enter:

```
host1/Admin(config-pmap-lb-c) # no forward
```

Configuring HTTP Header Insertion

When the ACE uses network address translation (NAT) to translate the source IP address of a client to a VIP, servers need a way to identify that client for the TCP and IP return traffic. To identify a client whose source IP address has been NATed, you can instruct the ACE to insert a generic header and string value of your choice in the client HTTP request. For details about configuring NAT, see the *Cisco Application Control Engine Module Security Configuration Guide*.



Note

With either TCP server reuse or persistence rebalance enabled, the ACE inserts a header in every client request. For information about TCP server reuse, see the [“Configuring TCP Server Reuse”](#) section. For information about persistence rebalance, see [“Enabling HTTP Persistence Rebalance”](#) section.

To insert a generic header and value in an HTTP request, use the **insert-http** command in policy map load balance class configuration mode. You can enter multiple **insert-http** commands for each class. The syntax of this command is:

```
insert-http name header-value expression
```

**Note**

You can associate a maximum of 1024 instances of the same type of regex with each Layer 3 and Layer 4 policy map. This limit applies to all Layer 7 policy-map types, including generic, HTTP, RADIUS, RDP, RTSP, and SIP. You configure regexes in:

- Match statements in Layer 7 class maps
- Inline match statements in Layer 7 policy maps
- Layer 7 hash predictors for server farms
- Layer 7 sticky expressions in sticky groups
- Header insertion and rewrite (including SSL URL rewrite) expressions in Layer 7 action lists

The keywords and arguments are:

- *name*—Specifies the name of the HTTP header to insert in the client HTTP request. Enter an unquoted text string with no spaces and a maximum of 255 alphanumeric characters. You can specify any custom header name that you want, subject to the maximum character length. You can also enter any of the predefined header names in [Table 1-4](#), regardless of whether that header name already exists in the client request header. The ACE does not overwrite any existing header information in the client request.
- **header-value** *expression*—Specifies the header-value expression string to insert in the HTTP header. Enter a text string with a maximum of 255 alphanumeric characters.

You can also specify the following special **header-value** expressions using the following special parameter values:

- *%is*—Inserts the source IP address in the HTTP header
- *%id*—Inserts the destination IP address in the HTTP header
- *%ps*—Inserts the source port in the HTTP header
- *%pd*—Inserts the destination port in the HTTP header

**Note**

For Microsoft Outlook Web Access (OWA), specify the field name as HTTP_FRONT_END_HTTPS with a value of ON.

For example, in an SSL configuration, you could insert a generic field called ClientCert and the header value could be the client certificate or a portion thereof.

For example, to insert the header name Host with a header value of www.cisco.com in an HTTP client request header, enter:

```
host1/Admin(config)# policy-map type loadbalance first-match
L7SLBPOLICY
host1/Admin(config-pmap-lb)# class L7SLBCLASS
host1/Admin(config-pmap-lb-c)# insert-http Host header-value
www.cisco.com
```

The header name and value will appear in the HTTP header as:

```
Host: www.cisco.com
```

To remove the HTTP header name and value from the policy map, enter:

```
host1/Admin(config-pmap-c)# no insert-http Host header-value
www.cisco.com
```

Enabling Load Balancing to a Server Farm (Configuring a Backup Server)

To load-balance client requests for content to a server farm, use the **serverfarm** command in policy-map class configuration mode. Server farms are groups of networked real servers that contain the same content and that typically reside in the same physical location. The syntax of this command is:

```
serverfarm name1 [backup name2 [sticky] [aggregate-state]]
```

The keywords, arguments, and options are:

- *name1*—A unique identifier of the primary server farm. Enter an unquoted text string with no spaces and a maximum of 64 characters.
- **backup** *name2*—(Optional) Designates an existing server farm as a backup server farm in case all the servers in the original server farm become unavailable. When at least one server in the primary server farm becomes available again, the ACE sends all connections to the primary server farm. Enter the name of an existing server farm that you want to specify as a backup server farm as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.



Note If all servers in the server farm fail and you did not configure a backup server farm, the ACE sends a reset (RST) to a client in response to a content request.

- **sticky**—(Optional) Applies stickiness to the backup server farm. If all real servers in the primary server farm fail, the ACE sends client requests to the backup server farm. If you configure this **sticky** option, the backup server farm becomes sticky using source IP address stickiness by default. When at least one real server in the primary server farm becomes available again, the sticky backup server farm continues to service existing connections and new requests over existing sticky connections. The ACE sends all new connection requests to the primary server farm. If you want all connections to return to the primary server farm after it comes back up, do not configure this option. See the “[Configuring a Sorry Server Farm](#)” section.
- **aggregate-state**—(Optional, but recommended) Specifies that the state of the specified server farm is tied to the state of all the real servers in that server farm and in the backup server farm if configured. The ACE declares the primary server farm down if all real servers in the primary server farm and all real servers in the backup server farm are down.

The following example specifies the **serverfarm** command as an action in the load-balancing policy map.

```
host1/Admin(config)# policy-map type loadbalance first-match
L7SLBPOLICY
host1/Admin(config-pmap-lb)# class L7SLBCLASS
host1/Admin(config-pmap-lb-c)# serverfarm FARM2 backup FARM3 sticky
```

To remove the server-farm action from the Layer 7 load-balancing policy map, enter:

```
host1/Admin(config-pmap-lb-c)# no serverfarm FARM2
```

Configuring a Sorry Server Farm

When the primary server farm and any configured redirect servers are unavailable, you can instruct the ACE to send client requests to a sorry server farm. A sorry server is a server in a backup server farm with content stating that the Web page, resource, or service that a client requested is temporarily unavailable. When at least one server in the primary server farm returns to service, the ACE directs clients back to the primary server farm.

To configure a sorry server, use the **serverfarm** command in policy-map class configuration mode as described in the “[Enabling Load Balancing to a Server Farm \(Configuring a Backup Server\)](#)” section. In this case, configure a backup server farm, but do *not* use the **sticky** option of the **backup** keyword and do *not* associate the serverfarm with a sticky group. For more information about configuring stickiness, see [Chapter 5, Configuring Stickiness](#). Also, configure the **predictor hash source-ip** command as the load-balancing method for the server farms. This predictor method guarantees that client connections will return to the primary server farm when it comes back up. For information about configuring the server farm predictor, see [Chapter 1, Configuring Real Servers and Server Farms](#).

For example, to configure a sorry server farm, enter:

```
host1/Admin(config)# rserver SERVER1
host1/Admin(config-rserver-host)# ip address 192.168.12.4
host1/Admin(config-rserver-host)# inservice
host1/Admin(config)# rserver SERVER2
host1/Admin(config-rserver-host)# ip address 192.168.12.5
host1/Admin(config-rserver-host)# inservice
host1/Admin(config)# rserver SERVER3
host1/Admin(config-rserver-host)# ip address 192.168.12.6
host1/Admin(config-rserver-host)# inservice
host1/Admin(config)# rserver SERVER4
host1/Admin(config-rserver-host)# ip address 192.168.12.7
host1/Admin(config-rserver-host)# inservice

host1/Admin(config)# serverfarm SFARM1
host1/Admin(config-sfarm-host)# predictor hash source-ip
host1/Admin(config-sfarm-host)# rserver SERVER1
host1/Admin(config-sfarm-host-rs)# inservice
host1/Admin(config-sfarm-host)# rserver SERVER2
host1/Admin(config-sfarm-host-rs)# inservice

host1/Admin(config)# serverfarm SFARM2
host1/Admin(config-sfarm-host)# predictor hash source-ip
host1/Admin(config-sfarm-host)# rserver SERVER3
host1/Admin(config-sfarm-host-rs)# inservice
host1/Admin(config-sfarm-host)# rserver SERVER4
host1/Admin(config-sfarm-host-rs)# inservice

host1/Admin(config)# class-map type http loadbalance match-any
L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 100 match http header Host
header-value .*cisco.com
```

```

host1/Admin(config)# policy-map type loadbalance first-match
L7SLBPOLICY
host1/Admin(config-pmap-lb)# class L7SLBCLASS
host1/Admin(config-pmap-lb-c)# serverfarm SFARM1 backup SFARM2
aggregate-state

```

Configuring a Sticky Server Farm

To specify that requests matching a Layer 7 policy map be load balanced to a sticky server farm, use the **sticky-serverfarm** command in load-balancing policy-map class configuration mode. The syntax of this command is:

sticky-serverfarm *name*

The *name* argument specifies the identifier of an existing sticky group. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. For information about sticky groups, see [Chapter 5, Configuring Stickiness](#).

For example, enter:

```

host1/Admin(config)# policy-map type loadbalance first-match
L7SLBPOLICY
host1/Admin(config-pmap-lb)# class L7SLBCLASS
host1/Admin(config-pmap-lb-c)# sticky-serverfarm STICKY_GROUP1

```

To remove the sticky server farm from the policy map, enter:

```

host1/Admin(config-pmap-lb-c)# no sticky-serverfarm STICKY_GROUP1

```

Specifying the IP Differentiated Services Code Point of Packets

To specify the IP differentiated services code point (DSCP) of packets in a policy map, use the **set ip tos** command in policy-map class configuration mode. This command marks a packet by setting the IP DSCP bit in the Type of Service (TOS) byte. Once the IP DSCP bit is set, other Quality of Service (QoS) services can operate on the bit settings.

The syntax of this command is:

set ip tos *value*

The *value* argument specifies the IP DSCP value. Enter an integer from 0 to 255. The default is to not modify the ToS field.

The following example specifies the **set ip tos** command as a QoS action in the Layer 7 load-balancing policy map. All packets that satisfy the match criteria of L7SLBCLASS are marked with the IP DSCP value of 8. How packets marked with the IP DSCP value of 8 are treated is determined by the network configuration.

```
host1/Admin(config)# policy-map type loadbalance first-match  
L7SLBPOLICY
```

```
host1/Admin(config-pmap)# class L7SLBCLASS
```

```
host1/Admin(config-pmap-lb-c)# set ip tos 8
```

To reset the ACE behavior to the default of not modifying the TOS byte value, enter:

```
host1/Admin(config-pmap-lb-c)# no set ip tos 8
```

Specifying an SSL Proxy Service

The ACE uses an SSL proxy service in a Layer 7 policy map to loadbalance outbound SSL initiation requests to SSL servers. In this case, the ACE acts as an SSL client sending an encrypted request to an SSL server. For more information about SSL initiation, refer to the *Cisco Application Control Engine Module SSL Configuration Guide*.

To specify an SSL proxy service in a policy map, use the **ssl-proxy** command in load-balancing policy-map class configuration mode. The syntax of this command is:

```
ssl-proxy client name
```

The *name* argument specifies the identifier of an existing SSL proxy service. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, enter:

```
host1/Admin(config-pmap-lb-c)# ssl-proxy client PROXY_SERVICE1
```

To remove the SSL proxy service from the policy map, enter:

```
host1/Admin(config-pmap-lb-c)# no ssl-proxy client PROXY_SERVICE1
```

Associating a Layer 7 Policy Map with a Layer 3 and Layer 4 Policy Map

To associate a Layer 7 SLB policy with a Layer 3 and Layer 4 SLB policy, use the **service-policy type loadbalance** command in policy-map class configuration mode. For details, see the [“Associating a Layer 7 SLB Policy Map with a Layer 3 and Layer 4 SLB Policy Map”](#) section.

Configuring an HTTP Parameter Map

A parameter map is a means to combine related HTTP actions for use in a Layer 3 and Layer 4 policy map. You reference this parameter map in the policy map using the **appl-parameter http advanced-options** command. See the [“Associating an HTTP Parameter Map with a Layer 3 and Layer 4 Policy Map”](#) section.

To configure advanced HTTP behavior for SLB connections, use the **parameter-map type http** command in configuration mode. The syntax of this command is:

```
parameter-map type http name
```

The *name* argument specifies the identifier assigned to the parameter map. Enter an unquoted text string with no spaces and a maximum of 32 alphanumeric characters.

For example, enter:

```
host1/Admin(config)# parameter-map type http HTTP_PARAMETER_MAP
```

To remove an HTTP parameter map from the configuration, enter:

```
host1/Admin(config)# no parameter-map type http HTTP_PARAMETER_MAP
```

Use the commands in the following sections to define the advanced HTTP parameter map:

- [Disabling Case Sensitivity Matching](#)
- [Defining URL Delimiters](#)
- [Setting the Maximum Number of Bytes to Parse for Cookies, HTTP Headers, and URLs](#)

- [Enabling HTTP Persistence Rebalance](#)
- [Configuring TCP Server Reuse](#)

Disabling Case Sensitivity Matching

By default, the ACE CLI is case sensitive. To disable case-sensitivity matching for HTTP only, use the **case-insensitive** command in HTTP parameter-map configuration mode. With case-insensitive matching enabled, upper- and lower-case letters are considered the same. When case sensitivity is disabled, it applies to:

- HTTP header names and values
- HTTP cookie names and values
- URL strings
- HTTP deep inspection (for details, refer to the *Cisco Application Control Engine Module Security Configuration Guide*)

The syntax of this command is:

```
case-insensitive
```

For example, to disable case sensitivity, enter:

```
host1/Admin(config-parammap-http) # case-insensitive
```

To reenable case-sensitive matching after it has been disabled, enter:

```
host1/Admin(config-parammap-http) # no case-insensitive
```

Defining URL Delimiters

To define a list of ASCII-character delimiters that you can use to separate the cookies in a URL string, use the **set secondary-cookie-delimiters** command in HTTP parameter-map configuration mode. The syntax of this command is:

```
set secondary-cookie-delimiters text
```

The *text* argument identifies the list of delimiters. Enter an unquoted text string with no spaces and a maximum of 4 characters. The order of the delimiters in the list does not matter. The default list of delimiters is: /&#+.

Cookies and their delimiters appear in GET request lines. In the following example of a GET request line, the ampersand (&) that appears between name-value pairs is the secondary cookie delimiter. The question mark (?) begins the URL query and is not configurable.

```
GET /default.cgi?user=me&hello=world&id=2 HTTP/1.1
```

For example, to specify the secondary cookie delimiters as !@#\$, enter:

```
host1/Admin(config-parammap-http)# set secondary-cookie-delimiters
!@#$
```

To reset the delimiter list to the default of /&#+, enter:

```
host1/Admin(config-parammap-http)# no set secondary-cookie-delimiters
!@#$
```

Setting the Maximum Number of Bytes to Parse for Cookies, HTTP Headers, and URLs

By default, the maximum number of bytes that the ACE parses to check for a cookie, HTTP header, or URL is 2048. If a cookie, HTTP header, or URL exceeds the default value, the ACE drops the packet and sends a RST (reset) to the client browser. You can increase the number of bytes that the ACE parses using the **set header-maxparse-length** command in HTTP parameter-map configuration mode. The syntax of this command is:

```
set header-maxparse-length bytes
```

The *bytes* argument specifies the maximum number of bytes to parse for the total length of all cookies, HTTP headers, and URLs. Enter an integer from 1 to 65535. The default is 2048 bytes.

For example, to set the HTTP header maximum parse length to 8192, enter:

```
host1/Admin(config-parammap-http)# set header-maxparse-length 8192
```

To reset the HTTP header maximum parse length to the default of 2048 bytes, enter:

```
host1/Admin(config-parammap-http)# no set header-maxparse-length
```

Configuring the ACE Behavior when a URL or Cookie Exceeds the Maximum Parse Length

To configure how the ACE handles cookies, HTTP headers, and URLs that exceed the maximum parse length, use the **length-exceed** command in HTTP parameter-map configuration mode. The syntax of this command is:

```
length-exceed { continue | drop }
```

The keywords are:

- **continue**—Continue load balancing. When you specify this keyword, the **persistence-rebalance** command is disabled if the total length of all cookies, HTTP headers, and URLs exceeds the maximum parse length value. For details on setting the maximum parse length, see the “[Setting the Maximum Number of Bytes to Parse for Cookies, HTTP Headers, and URLs](#)” section.
- **drop**—(Default) Stop load balancing and discard the packet.

For example, enter:

```
host1/Admin(config-parammap-http)# length-exceed continue
```

To reset the ACE behavior to the default of stopping load balancing and discarding a packet when its URL or cookie exceeds the maximum parse length, enter:

```
host1/Admin(config-parammap-http)# no length-exceed continue
```

Enabling HTTP Persistence Rebalance

Persistence is sometimes referred to as *connection keepalive*. With persistence rebalance enabled, each subsequent HTTP request on the same TCP connection is load balanced independently. Persistence rebalance allows the ACE to load-balance each HTTP request to a potentially different Layer 7 class and/or real server.

Another effect of persistence rebalance is that header insertion and cookie insertion, if enabled, occur for every request instead of only the first request. For information about header insertion, see the “[Configuring HTTP Header Insertion](#)” section in this chapter. For information about cookie insertion, see the “[Enabling Cookie Insertion](#)” section in [Chapter 5, Configuring Stickiness](#).

By default, persistence rebalance is disabled. To enable the persistence rebalance feature, use the **persistence-rebalance** command in HTTP parameter-map configuration mode. The syntax of this command is:

persistence-rebalance



Note

If a real server is enabled with the NTLM Microsoft authentication protocol, we recommend that you leave persistence rebalance disabled. NTLM is a security measure that is used to perform authentication with Microsoft remote access protocols. When a real server is enabled with NTLM, every connection to the real server must be authenticated; typically, each client user will see a pop-up window prompting for a username and password. Once the connection is authenticated, all subsequent requests on the same connection will not be challenged. However, when the server load balancing function is enabled and configured with persistence rebalance, a subsequent request may point to a different real server causing a new authentication handshake.

The following example specifies the **parameter-map type http** command to configure URL cookie delimiter strings, set the maximum number of bytes to parse for URLs and cookies, and enable HTTP persistence:

```
host1/Admin(config)# parameter-map type http http_parameter_map
host1/Admin(config-parammap-http)# secondary-cookie-delimiters !@#
host1/Admin(config-parammap-http)# header-maxparse-length 4096
length-exceed continue
host1/Admin(config-parammap-http)# persistence-rebalance
```

To reset persistence to the default setting of disabled, enter:

```
host1/Admin(config-parammap-http)# no persistence-rebalance
```

Configuring TCP Server Reuse

TCP server reuse allows the ACE to reduce the number of open connections on a server by allowing connections to persist and be reused by multiple client connections. The ACE maintains a pool of TCP connections based on TCP options. New client connections can reuse those connections in the pool provided that the new client connections and prior server connections share the same TCP options. For information about configuring how the ACE handles TCP options, refer to the *Cisco Application Control Engine Module Security Configuration Guide*.

To ensure proper operation of this feature, observe the following TCP server reuse configurational recommendations and restrictions:

- Ensure that the ACE MSS is the same as the server MSS.
- Configure port address translation (PAT) on the interface that is connected to the real server. PAT prevents collisions when a client stops using a server connection and then that connection is reused by another client. Without PAT, if the original client tries to reuse the original server connection, it is no longer available. For details about configuring PAT, refer to the *Cisco Application Control Engine Module Security Configuration Guide*.
- Configure on the ACE the same TCP options that exist on the TCP server.
- Ensure that each server farm is homogeneous (all real servers within a server farm have identical configurations).

Another effect of TCP server reuse is that header insertion and cookie insertion, if enabled, occur for every request instead of only the first request. For information about header insertion, see the [“Configuring HTTP Header Insertion”](#) section in this chapter. For information about cookie insertion, see the [“Enabling Cookie Insertion”](#) section in [Chapter 5, Configuring Stickiness](#).

To configure TCP server reuse use the **server-conn reuse** command in HTTP parameter-map configuration mode. The syntax of this command is:

server-conn reuse

For example, to enable TCP server reuse, enter:

```
host1/Admin(config-parammap-http)# server-conn reuse
```

To disable TCP server reuse, enter:

```
host1/Admin(config-parammap-http)# no server-conn reuse
```

To display TCP server reuse information, use the **show conn detail** command in Exec mode. For details, see .

Configuring a Layer 3 and Layer 4 Class Map for SLB

A Layer 3 and Layer 4 class map contains match criteria to classify network traffic that can pass through the ACE. The ACE uses these Layer 3 and Layer 4 traffic classes to perform server load balancing (SLB). For a Layer 3 and Layer 4 traffic classification, the match criteria in a class map include the VIP address, protocol, and port of the ACE. You can configure multiple commands in a single class map to specify the match criteria in a group that you then associate with a traffic policy.

To create a Layer 3 and Layer 4 class map to classify network traffic passing through the ACE and enter class-map configuration mode, use the **class-map** command in configuration mode. The syntax of this command is:

```
class-map [match-all | match-any] map_name
```

The arguments and options are:

- **match-all** | **match-any**—(Optional) Determines how the ACE evaluates Layer 3 and Layer 4 network traffic when multiple match criteria exist in a class map. The class map is considered a match if the **match** commands meet one of the following conditions.
 - **match-all**—(Default) Network traffic needs to satisfy all of the match criteria (implicit AND) to match the class map.
 - **match-any**—Network traffic needs to satisfy only one of the match criteria (implicit OR) to match the load-balancing class map.
- *map_name*—Specifies the name assigned to the class map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. The class name is used for both the class map and to configure policy for the class in the policy map.

For example, to create a class map named L4VIPCLASS that specifies the network traffic must satisfy all the match criteria (**match-all** is the default), enter:

```
host1/Admin(config)# class-map L4VIPCLASS
```

To remove the class map from the configuration, enter:

```
host1/Admin(config)# no class-map L4VIPCLASS
```

After you have created a Layer 3 and Layer 4 class map for SLB, use the commands in the following sections to configure a description and the VIP match criteria for the class map:

- [Defining a Class Map Description](#)
- [Defining VIP Address Match Criteria](#)

Defining a Class Map Description

To provide a brief summary about the Layer 3 and Layer 4 class map, use the **description** command in class-map configuration mode. The syntax of this command is:

```
description text
```

For the *text* argument, enter an unquoted text string with no spaces and a maximum of 240 alphanumeric characters.

The following example specifies a description that the class map is to filter network traffic to the server.

```
host1/Admin(config)# class-map match-any L4VIPCLASS  
host1/Admin(config-cmap)# description filter server traffic
```

To remove the description from the class map, enter:

```
host1/Admin(config-cmap)# no description
```

Defining VIP Address Match Criteria

To define a 3-tuple flow of VIP address, protocol, and port as matching criteria for SLB, use the **match virtual-address** command in class-map configuration mode. You can configure multiple match criteria statements to define the VIPs for SLB. The syntax of this command is:

```
[line_number] match virtual-address vip_address {[mask] | any | {tcp | udp  
| any | eq port_number | range port1 port2}} | protocol_number}
```

The keywords, arguments, and options are:

- *line_number*—(Optional) Assists you in editing or deleting individual match commands. For example, you can enter **no line_number** to delete long match commands instead of entering the entire line.
- *vip_address*—Virtual IP (VIP) address of the virtual server in the ACE, specified in dotted-decimal notation (192.168.1.2). VIPs are public addresses owned by the customer.
- *mask*—(Optional) Mask for the VIP address of the ACE to allow connections to an entire network, specified in dotted decimal format (255.255.255.0).
- **any**—Wildcard value for the IP protocol value.
- **tcp | udp**—Specifies the protocol, TCP or UDP.
 - **any**—Wildcard value for the TCP or UDP port number. With **any** used in place of either the **eq** or **range** values, packets from any incoming port match.
 - **eq port_number**—Specifies that the TCP or UDP port number must match the specified value. Enter an inter from 0 to 65535. A value of 0 instructs the ACE to include all ports. Alternatively, you can enter the keyword name of a well-known TCP port from [Table 1-5](#) or a well-known UDP port from [Table 1-6](#).

Table 1-5 Well-Known TCP Port Numbers and Key Words

| Key Word | Port Number | Description |
|-----------------|-------------|--|
| domain | 53 | Domain Name System (DNS) |
| ftp | 21 | File Transfer Protocol (FTP) |
| ftp-data | 20 | FTP data connections |
| http | 80 | Hyper Text Transfer Protocol (HTTP) |
| https | 443 | HTTP over TLS or SSL (HTTPS) |
| irc | 194 | Internet Relay Chat (IRC) |
| matip-a | 350 | Mapping of Airline Traffic over Internet Protocol (MATIP) Type A |
| nntp | 119 | Network News Transport Protocol (NNTP) |
| pop2 | 109 | Post Office Protocol (POP) v2 |

Table 1-5 Well-Known TCP Port Numbers and Key Words (continued)

| Key Word | Port Number | Description |
|---------------|-------------|--|
| pop3 | 110 | Post Office Protocol (POP) v3 |
| rtsp | 554 | Real Time Stream control Protocol (RTSP) |
| smtp | 25 | Simple Mail Transfer Protocol (SMTP) |
| telnet | 23 | Telnet |
| www | 80 | World Wide Web (WWW) |

Table 1-6 Well-Known UDP Port Numbers and Key Words

| Key Word | Port Number | Description |
|---------------------|-------------|--|
| domain | 53 | Domain Name System |
| wsp | 9200 | Connectionless Wireless Session Protocol (WSP) |
| wsp-wtls | 9202 | Secure Connectionless WSP |
| wsp-wtp | 9201 | Connection-based WSP |
| wsp-wtp-wtls | 9203 | Secure Connection-based WSP |

- **range** *port1 port2*—Specifies a port range to use for the TCP or UDP port. Valid port ranges are 0 to 65535. A value of 0 instructs the ACE to match all ports.
- *protocol_number*—Specifies the number of an IP protocol. Enter an integer from 1 to 254 that represents an IP protocol number.

**Note**

The ACE always attempts to match incoming traffic to the configured classes in a Layer 4 multi-match policy on a first-match basis. When you configure two or more class maps with the same VIP address match criteria and you configure the protocol as **any** in the first class map, the ACE will not match incoming traffic to a more specific class map (one with a specified protocol and port) that follows the non-specific class map in a policy map. For example, if you configure **match virtual-address 192.168.12.15 any** in the first class map and **match virtual-address 192.168.12.15 tcp eq https** in the second class map and associate the classes in that order in a policy map, any HTTPS traffic received by the ACE will never match the intended class. Therefore, the ACE will loadbalance the traffic to the wrong server and you will receive The Page Cannot Be Displayed error in your browser. Always configure the more specific class first, or else you may experience unexpected results. If you configure the two classes in separate Layer 4 policy maps, be sure to apply the policies in the correct order on the interface using a service policy.

The following example specifies that the class map L4VIPCLASS matches traffic destined to VIP address 192.168.1.10 and TCP port number 80.

```
host1/Admin(config)# class-map L4VIPCLASS
host1/Admin(config-cmap)# match virtual-address 192.168.1.10 tcp port
eq 80
```

To remove the VIP match statement from the class map, enter:

```
host1/Admin(config-cmap)# no match virtual-address 192.168.1.10 tcp
port eq 80
```

Configuring a Layer 3 and Layer 4 Policy Map for SLB

For a Layer 3 and Layer 4 SLB traffic classification, you create an SLB policy map that contains actions that are related to a VIP. A policy map associates a predefined traffic class (class map) with a series of actions to be performed on the traffic that matches the classifications defined in the traffic class. At the Layer 3 and Layer 4 network traffic level, there is a single policy map for each network traffic feature. The Layer 3 and Layer 4 policy maps are typed accordingly and, through the **service-policy** command, applied to a single interface or globally to all interfaces in a context.

The sequence in which the ACE applies actions for a specific policy are independent of the actions configured for a class inside a policy. The ACE follows an implicit feature lookup order that is dictated by the policy map actions and features, which can mean that the user-configured order of class maps does not necessarily have an effect on the lookup order. For example, if you configure one or more security ACLs in a policy map, an ACL may not allow a certain flow through the ACE even if you want to perform operations such as SLB on that flow. For details on the lookup order that the ACE uses, refer to the *Cisco Application Control Engine Module Administration Guide*.

To create an SLB policy map, use the **policy-map** command in configuration mode. The syntax of this command is:

```
policy-map multi-match map_name
```

The keywords, arguments, and options are:

- **multi-match**—Allows the inclusion of multiple Layer 3 and Layer 4 network traffic-related actions in the same policy map, enabling the ACE to execute all possible actions applicable for a specific classification, for example, SLB, NAT (see the *Cisco Application Control Engine Module Security Configuration Guide*), and AAA.
- *map_name*—Specifies the unique identifier of the policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, to create a Layer 3 and Layer 4 network traffic policy map, enter:

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY  
host1/Admin(config-pmap)#
```

To remove a policy map from the configuration, enter:

```
host1/Admin(config)# no policy-map multi-match L4SLBPOLICY
```

If there are multiple instances of actions of the same type (feature) configured in a policy map, the ACE performs the first action encountered of that type.

Defining a Layer 3 and Layer 4 Policy Map Description

Use the **description** command to provide a brief summary about the Layer 3 and Layer 4 policy map.

Access the policy map configuration mode to specify the **description** command.

The syntax of this command is:

```
description text
```

Use the *text* argument to enter an unquoted text string with a maximum of 240 alphanumeric characters.

For example, to specify a description that the policy map is to filter network traffic to a VIP, enter:

```
host1/Admin(config-pmap) # description filter traffic matching a VIP
```

To remove the description from the class map, enter:

```
host1/Admin(config-pmap) # no description
```

Associating a Layer 3 and Layer 4 Class Map with a Policy Map

To associate a Layer 3 and Layer 4 SLB class map with a Layer 3 and Layer 4 SLB policy map, use the **class** command in policy-map configuration mode. The syntax of this command is:

```
class {name1 [insert-before name2] | class-default}
```

The keywords, arguments, and options are:

- *name1*—Specifies the name of a previously defined traffic class configured with the **class-map** command. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
- **class-default**—Reserved, well-known class map created by the ACE. You cannot delete or modify this class. All traffic that fails to meet the other matching criteria in the named class map belongs to the default traffic class. If none of the specified classifications match the traffic, then the ACE performs the action specified under the **class class-default** command. The **class-default** class map has an implicit **match any** statement in it enabling it to match all traffic.

- **insert-before** *name2*—(Optional) Places the current class map ahead of an existing class map specified by the *name2* argument in the policy-map configuration. The ACE does not preserve the command in the running configuration, but does retain the configured order of class maps in the policy map.

The following example illustrates the use of the **insert-before** command to define the sequential order of two class maps in the policy map:

```
host1/Admin(config-pmap)# class L4VIPCLASS insert-before FILTERHTML
```

To remove a class map from a Layer 3 and Layer 4 policy map, enter:

```
host1/Admin(config-pmap)# no class L4VIPCLASS
```

Specifying Layer 3 and Layer 4 SLB Policy Actions

After you associate a Layer 3 and Layer 4 class map with an SLB policy map, you need to specify the actions that the ACE should take when network traffic matches one or more match statements in a class map. To specify the Layer 3 and Layer 4 SLB policy actions, use the commands described in the following subsections:

- [Associating a Layer 7 SLB Policy Map with a Layer 3 and Layer 4 SLB Policy Map](#)
- [Associating an HTTP Parameter Map with a Layer 3 and Layer 4 Policy Map](#)
- [Associating a Connection Parameter Map with a Layer 3 and Layer 4 Policy Map](#)
- [Enabling the Advertising of a Virtual Server IP Address](#)
- [Enabling a VIP to Reply to ICMP Requests](#)
- [Enabling a VIP](#)

Associating a Layer 7 SLB Policy Map with a Layer 3 and Layer 4 SLB Policy Map

The ACE treats all Layer 7 policy maps as child policies, so you must always associate a Layer 7 SLB policy map with a Layer 3 and Layer 4 SLB policy map. You can apply on an interface or globally on all interfaces in a context only a Layer 3 and Layer 4 policy map. For details on creating a Layer 7 load-balancing policy map, see the “[Configuring a Layer 7 Policy Map for SLB](#)” section.

To associate a Layer 7 SLB policy map with a Layer 3 and Layer 4 SLB policy map, use the **loadbalance** command in policy-map class configuration mode.

The syntax of this command is:

loadbalance *policy name*

The **policy name** keyword and argument specifies the identifier of an existing Layer 7 SLB policy map. Enter the name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

The following example references the Layer 7 L7SLBPOLICY policy map within the Layer 3 and Layer 4 L4SLBPOLICY policy map.

```
host1/Admin(config)# policy-map type loadbalance first-match  
L7SLBPOLICY  
host1/Admin(config-pmap-lb)# class L7SLBCLASS  
host1/Admin(config-pmap-lb-c)# serverfarm FARM2
```

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY  
host1/Admin(config-pmap)# class L4SLBCLASS  
host1/Admin(config-pmap-c)# loadbalance policy L7SLBPOLICY
```

To dissociate the Layer 7 SLB policy from the Layer 3 and Layer 4 SLB policy, enter:

```
host1/Admin(config-pmap-c)# no loadbalance policy L7SLBPOLICY
```

Associating an HTTP Parameter Map with a Layer 3 and Layer 4 Policy Map

To configure advanced HTTP parameters, you can create a parameter map to define related actions. See the “[Configuring an HTTP Parameter Map](#)” section.

To associate an HTTP parameter map with a Layer 3 and Layer 4 policy map, use the **appl-parameter http advanced-options** command in policy-map class configuration mode.

The syntax of this command is:

```
appl-parameter http advanced-options name
```

The *name* argument identifies the name of an existing HTTP parameter map. Parameter maps aggregate HTTP traffic-related actions together. For details, see the “[Configuring an HTTP Parameter Map](#)” section.

The following example specifies the **appl-parameter http advanced-options** command as an action for the SLB policy map.

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY  
host1/Admin(config-pmap)# class FILTERHTTP  
host1/Admin(config-pmap-c)# appl-parameter http advanced-options  
HTTP_PARAM_MAP1
```

To dissociate the HTTP parameter map as an action from the SLB policy map, enter:

```
host1/Admin(config-pmap-c)# no appl-parameter http advanced-options  
HTTP_PARAM_MAP1
```

Associating a Connection Parameter Map with a Layer 3 and Layer 4 Policy Map

To configure TCP/IP normalization and connection parameters, you can create a connection parameter map to define related actions. Refer to the *Cisco Application Control Engine Module Security Configuration Guide*.

To associate a connection parameter map with a Layer 3 and Layer 4 policy map, use the **connection advanced-options** command in policy-map class configuration mode. The syntax of this command is:

```
connection advanced-options name
```

For the *name* argument enter the name of an existing parameter map as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, enter:

```
host1/Admin(config-pmap-c)# connection advanced-options TCP_PARAM_MAP
```

To dissociate the TCP parameter map from a policy map, enter:

```
host1/Admin(config-pmap-c)# no connection advanced-options
TCP_PARAM_MAP
```

Enabling the Advertising of a Virtual Server IP Address

To allow the ACE to advertise the IP address of the virtual server as the host route, use the **loadbalance vip advertise** command in policy-map class configuration mode. This function is used with route health injection (RHI) to allow the ACE to advertise the availability of a VIP address throughout the network.

The syntax of this command is:

```
loadbalance vip advertise [active] | [metric number]
```

The keywords, arguments, and options are:

- **active**—(Optional) Allows the ACE to advertise the IP address of the virtual server (VIP) as the host route only if there is at least one active real server in the server farm. Without the **active** option, the ACE always advertises the VIP whether or not there is any active real server associated with this VIP.
- **metric number**—(Optional) Specifies the distance metric for the route. Enter the metric value that needs to be entered in the routing table. Valid values are 1 through 254.

The following example specifies the **advertise** command as an action for the SLB policy map.

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY
host1/Admin(config-pmap)# class FILTERHTTP
host1/Admin(config-pmap-c)# loadbalance vip advertise active
```

To stop advertising the host route as an action from the SLB policy map, enter:

```
host1/Admin(config-pmap-c)# no loadbalance vip advertise active
```

Enabling a VIP to Reply to ICMP Requests

To enable a VIP to reply to ICMP ECHO requests, use the **loadbalance vip icmp-reply** command in policy-map class configuration mode. For example, if a user sends an ICMP ECHO request to a VIP, this command instructs the VIP to send an ICMP ECHO-REPLY. The syntax of this command is:

```
loadbalance vip icmp-reply [active]
```

The **active** option instructs the ACE to reply to an ICMP request only if the configured VIP is active. If the VIP is not active and the **active** option is specified, the ACE discards the ICMP request and the request times out.

**Note**

The **loadbalance vip icmp-reply** command alone controls a ping to a VIP on the ACE. This command implicitly downloads an ICMP access control list entry for the VIP. When you configure this command on the ACE, any configured ACLs that deny ICMP traffic have no effect on a client's ability to ping the VIP.

To complete the configuration when you configure the **active** option of this command, be sure to configure a Telnet probe and associate it with the server farm. The probe monitors the health of all the real servers in the server farm and ensures that the VIP responds with an ICMP ECHO REPLY only if the server port is active. If the server port is down or unreachable, the probe fails and the VIP stops responding to the ECHO request. For details about configuring probes, see [Chapter 4, Configuring Health Monitoring](#).

**Note**

For security reasons, the ACE does not allow pings from an interface on a VLAN on one side of the ACE through the module to an interface on a different VLAN on the other side of the module. For example, you cannot ping a VIP from a server if the VIP is on a VLAN that is different from the server VLAN.

For example, enter:

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY  
host1/Admin(config-pmap)# class FILTERHTTP  
host1/Admin(config-pmap-c)# loadbalance vip icmp-reply active
```

Enabling a VIP

To enable a VIP for SLB operations, use the **loadbalance vip inservice** command in policy-map class configuration mode. The syntax of this command is:

loadbalance vip inservice

The following example specifies the **loadbalance vip inservice** command as an action for the SLB policy map.

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY
host1/Admin(config-pmap)# class FILTERHTTP
host1/Admin(config-pmap-c)# loadbalance vip icmp-reply active
host1/Admin(config-pmap-c)# loadbalance vip inservice
```

To disable a VIP, enter:

```
host1/Admin(config-pmap-c)# no loadbalance vip inservice
```

Applying a Layer 3 and Layer 4 Policy to an Interface

Use the **service-policy** command to:

- Apply a previously created policy map.
- Attach the traffic policy to a specific VLAN interface or globally to all VLAN interfaces in the same context.

The **service-policy** command is available at both the interface configuration mode and at the configuration mode. Specifying a policy map in the configuration mode applies the policy to all of the VLAN interfaces associated with a context.

The syntax of this command is:

service-policy input *policy_name*

The keywords, arguments, and options are:

- **input**—Specifies that the traffic policy is to be attached to the input direction of an interface. The traffic policy evaluates all traffic received by that interface.
- *policy_name*—Specifies the name of a previously defined policy map, configured with a previously created **policy-map** command. The name can be a maximum of 64 alphanumeric characters.

Note the following when applying a service policy:

- Policy maps, applied globally in a context, are internally applied on all interfaces associated with the context.
- You can apply the policy in an input direction only.
- A policy activated on an interface overwrites any specified global policies for overlapping classification and actions.
- The ACE allows only one policy of a specific feature type to be activated on a given interface.
- You can configure a maximum of 4000 policy maps per ACE.
- You can apply a maximum of 128 service policies on each interface.

When you configure multiple VIPs on an interface, the match criteria for incoming traffic follow the order in which you configure the service-policy statements on that interface. Each service policy that you configure on an interface applies a Layer 3 and Layer 4 multi-match policy map to the interface. Each multi-match policy map may contain one or more features as defined in the class maps associated with the policy map.

Because service policies do not have line numbers, the order in which you configure them on an interface is extremely important. The reason is that the ACE has an implicit feature lookup order as follows:

1. Access control (permit or deny a packet)
2. Management traffic
3. TCP normalization and connection parameters
4. Server load balancing
5. Application protocol inspection
6. Source NAT
7. Destination NAT

When you apply multiple service policies to an interface, the ACE appends the last service policy at the end of the list. If you need to change the order of the service policies on an interface, you must first remove the service policies and then add them back in the appropriate order. This process is disruptive to the network.

As an alternative to reordering the service policies, you can configure multiple class maps in the same multi-match policy map, where you can define the order of the class maps. This process is not disruptive to the network. When you add new class maps to an existing policy, use the **insert-before** command to place the new class map in the desired order.

The following example shows how to configure two class maps in a policy map, where the VIP-ACCESS-MANAGER-80 is the more specific class map. To ensure that the ACE matches traffic to the more specific classification, configure VIP-ACCESS-MANAGER-80 class map first under the LB-TRAFFIC policy map. Note that this example and the one that follows include only the Layer 3 and Layer 4 traffic classification portions of the configuration.

```
class-map match-all VIP-ACCESS-MANAGER-ANY
  2 match virtual-address 10.238.45.200 tcp eq any
class-map match-all VIP-ACCESS-MANAGER-80
  2 match virtual-address 10.238.45.200 tcp eq www
policy-map multi-match LB-TRAFFIC
  class VIP-ACCESS-MANAGER-80
    loadbalance vip inservice
    loadbalance policy POLICY-ACCESS-MANAGER-80
    loadbalance vip icmp-reply active
  class VIP-ACCESS-MANAGER-ANY
    loadbalance vip inservice
    loadbalance policy POLICY-ACCESS-MANAGER-ANY
    loadbalance vip icmp-reply active

interface vlan 758
  description CLIENT-SIDE-VLAN
  bridge-group 100
  access-group input ALL
  service-policy input LB-TRAFFIC
  no shutdown
```

The following example shows how to use two policy maps, one for each class map, and achieve the same results as in the previous example. In this example, you configure the more specific policy map first under the interface using a service policy.

```
policy-map multi-match LB-TRAFFIC-80
  class VIP-ACCESS-MANAGER-80
    loadbalance vip inservice
    loadbalance policy POLICY-ACCESS-MANAGER-80
    loadbalance vip icmp-reply active
```

```
policy-map multi-match LB-TRAFFIC-ANY
  class VIP-ACCESS-MANAGER-ANY
    loadbalance vip inservice
    loadbalance policy POLICY-ACCESS-MANAGER-ANY
    loadbalance vip icmp-reply active

interface vlan 758
  description CLIENT-SIDE-VLAN
  bridge-group 100
  access-group input ALL
  service-policy input LB-TRAFFIC-80
  service-policy input LB-TRAFFIC-ANY
  no shutdown
```

The following example specifies an interface VLAN and applies the Layer 3 and Layer 4 SLB policy map to the VLAN:

```
host1/Admin(config)# interface vlan50
host1/Admin(config-if)# mtu 1500
host1/Admin(config-if)# ip address 172.20.1.100 255.255.0.0
host1/Admin(config-if)# service-policy input L4SLBPOLICY
```

To apply the Layer 3 and Layer 4 SLB policy map to all interfaces in the context:

```
host1/Admin(config)# service-policy input L4SLBPOLICY
```

To detach a traffic policy from an interface, enter:

```
host1/Admin(config-if)# no service-policy input L4SLBPOLICY
```

To globally detach a traffic policy from a context, enter:

```
host1/Admin(config)# no service-policy input L4SLBPOLICY
```

When you detach a traffic policy either:

- Individually from the last VLAN interface on which you applied the service policy or
- Globally from all VLAN interfaces in the same context

The ACE automatically resets the associated service-policy statistics. The ACE performs this action to provide a new starting point for the service-policy statistics the next time you attach a traffic policy to a specific VLAN interface or globally to all VLAN interfaces in the same context.

Displaying Load-Balancing Configurational Information and Statistics

Use the commands in the following sections to display configurational information and statistics for Layer 3 and Layer 4, and Layer 7 class maps and policy maps. This section contains the following subsections:

- [Displaying Class-Map Configurational Information](#)
- [Displaying Policy-Map Configurational Information](#)
- [Displaying Service-Policy Configurational Information](#)
- [Displaying Parameter Map Configurational Information](#)
- [Displaying Load-Balancing Statistics](#)
- [Displaying HTTP Parameter Map Statistics](#)
- [Displaying Service-Policy Statistics](#)

Displaying Class-Map Configurational Information

To display class-map configurational information, use the **show running-config class-map** command in Exec mode. This command displays the names of all the class maps configured in the contexts to which you have access and the **match** statements configured in each class map. The syntax of this command is:

```
show running-config class-map
```

Displaying Policy-Map Configurational Information

To display policy-map configurational information, use the **show running-config policy-map** command in Exec mode. This command displays the names of all the policy maps configured in the contexts to which you have access and the class maps and actions configured in each policy map.

The syntax of this command is:

```
show running-config policy-map
```

Displaying Service-Policy Configurational Information

To display service-policy configurational information, use the **show running-config service-policy** command in Exec mode. This command displays the names of all the service policies configured in the contexts to which you have access. The syntax of this command is:

```
show running-config service-policy
```

Displaying Parameter Map Configurational Information

To display a list of parameter maps and their configurations, use the **show running-config parameter-map** command in Exec mode. The syntax of this command is:

```
show running-config parameter-map
```

Displaying Load-Balancing Statistics

To display load-balancing statistics, use the **show stats loadbalance** command in Exec mode. The syntax of this command is:

```
show stats loadbalance
```

For example, enter:

```
host1/Admin# show stats loadbalance
```

Table 1-8 describes the fields in the **show stats loadbalance** command output for an HTTP parameter map.

Table 1-7 Field Descriptions for the show stats loadbalance Command Output

| Field | Description |
|-------------------------------------|---|
| Total Version Mismatch | VIP configuration changed during the load-balancing decision and the ACE rejected the connection. |
| Total Layer4 Decisions | Total number of times the ACE made a load-balancing decision at Layer 4. |
| Total Layer4 Rejections | Total number of Layer 4 connections that the ACE rejected. |
| Total Layer7 Decisions | Total number of times the ACE made a load-balancing decision at Layer 7. |
| Total Layer7 Rejections | Total number of Layer 7 connections that the ACE rejected. |
| Total Layer4 LB Policy Misses | Total number of connection requests that did not match a configured Layer 4 load-balancing policy. |
| Total Layer7 LB Policy Misses | Total number of connection requests that did not match a configured Layer 7 load-balancing policy. |
| Total Times Rserver Was Unavailable | Total number of times that the ACE attempted to loadbalance a request to a real server in a server farm, but the real server was NOTINSERVICE or was otherwise unavailable. |
| Total ACL denied | Total number of connection requests that were denied by an ACL. |

Displaying HTTP Parameter Map Statistics

To display statistics for an HTTP parameter map, use the **show parameter-map** command in Exec mode. The syntax of this command is:

```
show parameter-map name
```

The name argument specifies the identifier of an HTTP parameter map. Enter an unquoted text string with a maximum of 64 alphanumeric characters.

For example, to display statistics for the HTTP parameter map called HTTP_PARAMMAP, enter:

```
host1/Admin# show parameter-map HTTP_PARAMMAP
```

Table 1-8 describes the fields in the **show parameter-map** command output for an HTTP parameter map.

Table 1-8 Field Descriptions for the show parameter-map Command Output

| Field | Description |
|------------------------------|---|
| Parameter-map | Unique identifier of the HTTP parameter map |
| Type | HTTP |
| Server-side connection reuse | Status of TCP server reuse feature: enabled or disabled |
| Case-insensitive parsing | Status of the case-insensitive command: enabled or disabled |
| Persistence-rebalance | Status of the persistence-rebalance command: enabled or disabled |
| Header-maxparse-length | Configured value or the default value of the header-maxparse-length command |
| Content-maxparse-length | Configured value or the default value of the content-maxparse-length command |
| Parse length-exceed action | Configured action for the length-exceed command: continue or drop |
| Urlcookie-delimiters | Configured URL cookie delimiters |

Displaying Service-Policy Statistics

To display statistics for service policies enabled globally within a context or on a specific interface, use the **show service-policy** command. The syntax of this command is:

```
show service-policy policy_name [detail]
```

The keywords, arguments, and options are:

- *policy_name*—Displays statistics for the specified existing policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
- **detail**—(Optional) Displays detailed statistics for the specified policy map applied to an interface.



Note

The ACE updates the counters that the **show service-policy** command displays after the applicable connections are closed.

[Table 1-9](#) describes the fields in the **show service-policy detail** command output.

Table 1-9 *Field Descriptions for the show service-policy detail Command Output*

| Field | Description |
|----------------|---|
| Status | Current operational state of the service policy. Possible states are: ACTIVE or INACTIVE. |
| Description | User-entered description of the policy map |
| Interface | The VLAN ID of the interface to which the policy map has been applied |
| Service Policy | Unique identifier of the policy map |
| Class | Name of the class map associated with the service policy |
| Loadbalance | |
| L7 Policy | Name of the Layer 7 policy map associated with the service policy |

Table 1-9 *Field Descriptions for the show service-policy detail Command Output (continued)*

| Field | Description |
|---------------------|--|
| VIP Route Metric | Specifies the distance metric for the route as specified with the loadbalance vip advertise command. The ACE writes the value you specify in its routing table. Possible values are integers from 1 to 254. |
| VIP Route Advertise | Operational state of the loadbalance vip advertise command: ENABLED or DISABLED. This command is used with route health injection (RHI) to allow the ACE to advertise the availability of a VIP address throughout the network. |
| VIP State | Operational state of the virtual server: INSERVICE or OUTOFSERVICE |
| Curr Conns | Number of active connections to the VIP |
| Dropped Conns | Number of connections that the ACE discarded |
| Client Pkt Count | Number of packets received from the client |
| Server Pkt Count | Number of packets received from the server |
| Hit Count | Number of times a connection was established with this VIP |
| Client Byte Count | Number of bytes received from the client |
| Server Byte Count | Number of bytes received from the server |
| L4 Policy Stats | |
| Total Req/Resp | Total number of requests and responses for the policy map |
| Total Allowed | Total number of packets received and allowed |
| Total Dropped | Total number of packets received and discarded |
| Total Logged | Total number of errors logged |

Table 1-9 *Field Descriptions for the show service-policy detail Command Output (continued)*

| Field | Description |
|-----------------------|---|
| L7 loadbalance policy | Identifier of the Layer 7 policy map |
| Class-map | Identifier of the class map associated |
| LB action | Actions specified with the Layer 7 policy map |
| Hit count | Number of times a connection was established with this policy |
| Dropped conns | Number of connections associated with this policy that were dropped |

Displaying HTTP Statistics

To display HTTP statistics, including header insertion and server reuse statistics, use the **show stats http** command in Exec mode. The syntax of this command is:

```
show stats http
```

For example, enter:

```
host1/Admin# show stats http
```

```
+-----+
+----- HTTP statistics -----+
+-----+
LB parse result msgs sent : 0          , TCP data msgs sent      : 0
Inspect parse result msgs : 0          , SSL data msgs sent     : 0
      sent
TCP fin/rst msgs sent      : 0          , Bounced fin/rst msgs sent: 0
SSL fin/rst msgs sent      : 0          , Unproxy msgs sent      : 0
Drain msgs sent            : 0          , Particles read         : 0
Reuse msgs sent            : 0          , HTTP requests          : 0
Reproxied requests        : 0          , Headers removed        : 0
Headers inserted          : 0          , HTTP redirects         : 0
HTTP unproxy conns        : 0          , Pipeline flushes       : 0
Whitespace appends        : 0          , Second pass parsing     : 0
Response entries recycled : 0          , Analysis errors        : 0
Header insert errors      : 0          , Max parselen errors    : 0
```

```
Static parse errors      : 0           , Resource errors      : 0
Invalid path errors     : 0           , Bad HTTP version errors : 0
```

Clearing SLB Statistics

This section describes the commands you can use to clear load-balancing statistics. It includes the following subsections:

- [Clearing Load-Balancing Statistics](#)
- [Clearing Service-Policy Statistics](#)
- [Clearing HTTP Statistics](#)

Clearing Load-Balancing Statistics

To clear all load-balancing statistics in the current context, use the **clear stats loadbalance** command in Exec mode. The syntax of this command is:

```
clear stats loadbalance
```

For example, enter:

```
host1/Admin# clear stats loadbalance
```



Note

If you have redundancy configured, then you need to explicitly clear load-balancing statistics on both the active and the standby ACEs. Clearing statistics on the active module alone will leave the standby module's statistics at the old values.

Clearing Service-Policy Statistics

To clear service policy statistics, use the **clear service-policy** command. The syntax of this command is:

```
clear service-policy policy_name
```

For the *policy_name* argument, enter the identifier of an existing policy map that is currently in service (applied to an interface).

For example, to clear the statistics for the policy map L4SLBPOLICY that is currently in service, enter:

```
host1/Admin# clear service-policy L4SLBPOLICY
```

**Note**

If you have redundancy configured, then you need to explicitly clear service-policy statistics on both the active and the standby ACEs. Clearing statistics on the active module alone will leave the standby module's statistics at the old values.

Clearing HTTP Statistics

To clear all HTTP statistics in the current context, use the **clear stats http** command in Exec mode. The syntax of this command is:

```
clear stats http
```

For example, enter:

```
host1/Admin# clear stats http
```

**Note**

If you have redundancy configured, then you need to explicitly clear HTTP statistics on both the active and the standby ACEs. Clearing statistics on the active module alone will leave the standby module's statistics at the old values.

Where to Go Next

To configure health probes for your real servers, proceed to [Chapter 4, Configuring Health Monitoring](#). To configure stickiness (connection persistence), see [Chapter 5, Configuring Stickiness](#). To configure firewall load balancing (FWLB), see [Chapter 6, Configuring Firewall Load Balancing](#).