

# Rackspace Cloud Load Balancers

## Developer Guide

API v1.0 (Mar 9, 2012)



[docs.rackspace.com/api](https://docs.rackspace.com/api)

# Rackspace Cloud Load Balancers Developer Guide

API v1.0 (2012-03-09)

Copyright © 2010-2012 Rackspace US, Inc. All rights reserved.

This document is intended for software developers interested in developing applications using the Rackspace Cloud Load Balancers Application Program Interface (API). The document is for informational purposes only and is provided "AS IS."

RACKSPACE MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, AS TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS DOCUMENT AND RESERVES THE RIGHT TO MAKE CHANGES TO SPECIFICATIONS AND PRODUCT/SERVICES DESCRIPTION AT ANY TIME WITHOUT NOTICE. RACKSPACE SERVICES OFFERINGS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS MUST TAKE FULL RESPONSIBILITY FOR APPLICATION OF ANY SERVICES MENTIONED HEREIN. EXCEPT AS SET FORTH IN RACKSPACE GENERAL TERMS AND CONDITIONS AND/OR CLOUD TERMS OF SERVICE, RACKSPACE ASSUMES NO LIABILITY WHATSOEVER, AND DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO ITS SERVICES INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT.

Except as expressly provided in any written license agreement from Rackspace, the furnishing of this document does not give you any license to patents, trademarks, copyrights, or other intellectual property.

Rackspace®, Rackspace logo and Fanatical Support® are registered service marks of Rackspace US, Inc. All other product names and trademarks used in this document are for identification purposes only and are property of their respective owners.

# Table of Contents

|  |    |
|--|----|
| 1. Overview .....                                | 1  |
| 1.1. Intended Audience .....                     | 1  |
| 1.2. Document Change History .....               | 1  |
| 1.3. Additional Resources .....                  | 2  |
| 2. Concepts .....                                | 3  |
| 2.1. Load Balancer .....                         | 3  |
| 2.2. Virtual IP .....                            | 3  |
| 2.3. Error Page .....                            | 3  |
| 2.4. Node .....                                  | 3  |
| 2.5. Health Monitor .....                        | 3  |
| 2.5.1. Active Health Monitor .....               | 3  |
| 2.6. Session Persistence .....                   | 4  |
| 2.7. Connection Logging .....                    | 4  |
| 3. General API Information .....                 | 5  |
| 3.1. Authentication .....                        | 5  |
| 3.1.1. Geographic Endpoints .....                | 5  |
| 3.1.2. Retrieving the Authentication Token ..... | 5  |
| 3.2. Service Access/Endpoints .....              | 9  |
| 3.3. Request/Response Types .....                | 10 |
| 3.4. Content Compression .....                   | 10 |
| 3.5. Persistent Connections .....                | 11 |
| 3.6. Paginated Collections .....                 | 11 |
| 3.7. Limits .....                                | 11 |
| 3.7.1. Rate Limits .....                         | 11 |
| 3.7.2. Absolute Limits .....                     | 12 |
| 3.7.3. Determining Limits Programmatically ..... | 13 |
| 3.8. Faults .....                                | 14 |
| 3.8.1. serviceFault .....                        | 14 |
| 3.8.2. badRequest .....                          | 14 |
| 3.8.3. immutableEntity .....                     | 15 |
| 3.8.4. itemNotFound .....                        | 15 |
| 3.8.5. loadBalancerFault .....                   | 15 |
| 3.8.6. outOfVirtualIps .....                     | 16 |
| 3.8.7. overLimit .....                           | 16 |
| 3.8.8. serviceUnavailable .....                  | 16 |
| 3.8.9. unauthorized .....                        | 16 |
| 3.8.10. unprocessableEntity .....                | 17 |
| 3.9. Date/Time Format .....                      | 17 |
| 4. API Operations .....                          | 18 |
| 4.1. Load Balancers .....                        | 18 |
| 4.1.1. List Load Balancers .....                 | 18 |
| 4.1.2. List Load Balancer Details .....          | 21 |
| 4.1.3. Create Load Balancer .....                | 24 |
| 4.1.4. Update Load Balancer Attributes .....     | 29 |
| 4.1.5. Remove Load Balancer .....                | 31 |
| 4.2. Error Page .....                            | 31 |
| 4.2.1. Default Error Page .....                  | 31 |
| 4.2.2. Custom Error Page .....                   | 31 |

|  |    |
|--|----|
| 4.2.3. Error Page Operations .....           | 32 |
| 4.3. List Load Balancer Stats .....          | 33 |
| 4.4. Nodes .....                             | 34 |
| 4.4.1. List Nodes .....                      | 34 |
| 4.4.2. Add Nodes .....                       | 37 |
| 4.4.3. Modify Nodes .....                    | 39 |
| 4.4.4. Remove Nodes .....                    | 40 |
| 4.5. Virtual IPs .....                       | 40 |
| 4.5.1. List Virtual IPs .....                | 40 |
| 4.5.2. Add Virtual IP Version 6 .....        | 41 |
| 4.5.3. Remove Virtual IP .....               | 42 |
| 4.6. Usage Reports .....                     | 43 |
| 4.6.1. List Usage .....                      | 43 |
| 4.7. Access Lists .....                      | 44 |
| 4.7.1. Manage Access Lists .....             | 45 |
| 4.8. Monitors .....                          | 47 |
| 4.8.1. Monitor Health .....                  | 48 |
| 4.9. Sessions .....                          | 51 |
| 4.9.1. Manage Session Persistence .....      | 51 |
| 4.10. Connections .....                      | 53 |
| 4.10.1. Log Connections .....                | 53 |
| 4.10.2. Throttle Connections .....           | 54 |
| 4.11. Protocols .....                        | 56 |
| 4.11.1. List Load Balancing Protocols .....  | 56 |
| 4.12. Algorithms .....                       | 57 |
| 4.12.1. List Load Balancing Algorithms ..... | 58 |
| 4.13. SSL Termination .....                  | 58 |

## List of Tables

|  |    |
|--|----|
| 3.1. Regionalized Service Endpoints .....        | 9  |
| 3.2. JSON and XML Response Formats .....         | 10 |
| 3.3. Encoding Headers .....                      | 10 |
| 3.4. Default Rate Limits .....                   | 12 |
| 3.5. Explanation of Date/Time Format Codes ..... | 17 |
| 4.1. Virtual IP Types .....                      | 25 |
| 4.2. Load Balancer Statuses .....                | 30 |
| 4.3. Load Balancer Node Conditions .....         | 39 |
| 4.4. Virtual IP Types .....                      | 40 |
| 4.5. Health Monitor Types .....                  | 48 |
| 4.6. Session Persistence Modes .....             | 51 |
| 4.7. Load Balancing Algorithms .....             | 57 |
| 4.8. Attributes for SSL Termination .....        | 59 |
| 4.9. Optional SSL Attributes .....               | 61 |

## List of Examples

|  |    |
|--|----|
| 3.1. Auth Request: XML .....   | 6  |
| 3.2. Auth Request: JSON .....  | 6  |
| 3.3. Auth Response: XML .....  | 6  |
| 3.4. Auth Response: JSON .....   | 7  |
| 3.5. List Rate Limits Response: XML .....  | 13 |
| 3.6. List Rate Limits Response: JSON .....   | 13 |
| 3.7. List Absolute Limits Response: XML .....  | 14 |
| 3.8. List Absolute Limits Response: JSON .....                                       | 14 |
| 3.9. Fault Response, badRequest .....  | 15 |
| 3.10. Fault Response, immutableEntity .....  | 15 |
| 3.11. Fault Response, itemNotFound .....   | 15 |
| 3.12. Fault Response, loadBalancerFault .....  | 15 |
| 3.13. Fault Response, outOfVirtualIps .....  | 16 |
| 3.14. Fault Response, overLimit .....  | 16 |
| 3.15. Fault Response, serviceUnavailable .....                                       | 16 |
| 3.16. Fault Response, unauthorized .....   | 17 |
| 3.17. Fault Response, unprocessableEntity .....                                      | 17 |
| 3.18. Load Balancer Service Date/Time Format .....                                   | 17 |
| 4.1. List Load Balancers Response: XML .....   | 18 |
| 4.2. List Load Balancers Response: JSON .....  | 19 |
| 4.3. List Atom Load Balancers Response: XML .....                                    | 20 |
| 4.4. List Load Balancers With Node Specified: XML .....                              | 20 |
| 4.5. List Load Balancers With Node Specified: JSON .....                             | 21 |
| 4.6. List Load Balancer Details Response: XML .....                                  | 21 |
| 4.7. List Load Balancers Details Response: JSON .....                                | 22 |
| 4.8. List Atom Load Balancer Response: XML .....                                     | 23 |
| 4.9. Create Load Balancer (Required Attributes) Request: XML .....                   | 25 |
| 4.10. Create Load Balancer (Required Attributes) Request: JSON .....                 | 25 |
| 4.11. Create Load Balancer (Required Attributes) Response: XML .....                 | 26 |
| 4.12. Create Load Balancer (Required Attributes) Response: JSON .....                | 26 |
| 4.13. Create Load Balancer (Required Attributes with Shared IP) Request: XML .....   | 27 |
| 4.14. Create Load Balancer (Required Attributes with Shared IP) Request: JSON .....  | 28 |
| 4.15. Create Load Balancer (Required Attributes with Shared IP) Response: XML .....  | 28 |
| 4.16. Create Load Balancer (Required Attributes with Shared IP) Response: JSON ..... | 29 |
| 4.17. Update Load Balancer Attributes Request: XML .....                             | 30 |
| 4.18. Update Load Balancer Attributes Request: JSON .....                            | 30 |
| 4.19. List Error Page Response: XML .....  | 32 |
| 4.20. List Error Page Response: JSON .....   | 32 |
| 4.21. Add Error Page Request: XML .....  | 32 |
| 4.22. Add Error Page Request: JSON .....   | 32 |
| 4.23. Add Error Page Response: XML .....   | 33 |
| 4.24. Add Error Page Response: JSON .....  | 33 |
| 4.25. List Load Balancer stats Response: XML .....                                   | 34 |
| 4.26. List Load Balancers Stats Response: JSON .....                                 | 34 |
| 4.27. List Node Response: XML .....  | 35 |
| 4.28. List Node Response: JSON .....   | 35 |
| 4.29. List Atom Node Response: XML .....   | 35 |
| 4.30. List Nodes Response: XML .....   | 36 |

|  |    |
|--|----|
| 4.31. List Nodes Response: JSON .....  | 36 |
| 4.32. List Atom Nodes Response: XML .....                                    | 37 |
| 4.33. Add Nodes Request: XML .....   | 38 |
| 4.34. Add Nodes Request: JSON .....  | 38 |
| 4.35. Add Nodes Response: XML .....  | 38 |
| 4.36. Add Nodes Response: JSON .....   | 39 |
| 4.37. Update Node Condition Request: XML .....                               | 40 |
| 4.38. Update Node Condition Request: JSON .....                              | 40 |
| 4.39. List Virtual IPs Response: XML .....                                   | 41 |
| 4.40. List Virtual IPs Response: JSON .....                                  | 41 |
| 4.41. List Atom Virtual IPs Response: XML .....                              | 41 |
| 4.42. Add Virtual IP Version 6 Request: XML .....                            | 42 |
| 4.43. Add Virtual IP Version 6 Request: JSON .....                           | 42 |
| 4.44. Add Virtual IP Version 6 Response: XML .....                           | 42 |
| 4.45. Add Virtual IP Version 6 Response: JSON .....                          | 42 |
| 4.46. Report Load Balancer Usage Response: XML .....                         | 43 |
| 4.47. Report Load Balancer Usage Response: JSON .....                        | 44 |
| 4.48. List Access List Response: XML .....                                   | 45 |
| 4.49. List Access List Response: JSON .....                                  | 45 |
| 4.50. List Atom Access List Response: XML .....                              | 46 |
| 4.51. Update Access List Attributes Request: XML .....                       | 47 |
| 4.52. Update Access List Attributes Request: JSON .....                      | 47 |
| 4.53. Monitor Connections Request: XML .....                                 | 48 |
| 4.54. Monitor Connections Request: JSON .....                                | 48 |
| 4.55. Monitor Connections Response: XML .....                                | 49 |
| 4.56. Monitor Connections Response: JSON .....                               | 49 |
| 4.57. List Atom Health Monitor Response: XML .....                           | 49 |
| 4.58. Monitor HTTP Response: XML .....                                       | 50 |
| 4.59. Monitor HTTP Response: JSON .....                                      | 50 |
| 4.60. Monitor HTTPS Response: XML .....                                      | 51 |
| 4.61. Monitor HTTPS Response: JSON .....                                     | 51 |
| 4.62. List Session Persistence Configuration Response: XML .....             | 52 |
| 4.63. List Session Persistence Configuration Response: JSON .....            | 52 |
| 4.64. List Atom Session Persistence Response: XML .....                      | 52 |
| 4.65. Set Session Persistence Type Request: XML .....                        | 52 |
| 4.66. Set Session Persistence Type Request: JSON .....                       | 53 |
| 4.67. List Connection Logging Configuration Response: XML .....              | 53 |
| 4.68. List Connection Logging Configuration Response: JSON .....             | 53 |
| 4.69. Enable Connection Logging Request: XML .....                           | 53 |
| 4.70. Enable Connection Logging Request: JSON .....                          | 54 |
| 4.71. List Connection Throttling Configuration Response: XML .....           | 54 |
| 4.72. List Connection Throttling Configuration Response: JSON .....          | 55 |
| 4.73. Update Connection Throttling Configuration Request: XML .....          | 55 |
| 4.74. Update Connection Throttling Configuration Request: JSON .....         | 55 |
| 4.75. List Atom Connection Throttling Response: XML .....                    | 55 |
| 4.76. List Load Balancing Protocols Response: XML .....                      | 56 |
| 4.77. List Load Balancing Protocols Response: JSON .....                     | 56 |
| 4.78. List Load Balancing Algorithms Response: XML .....                     | 58 |
| 4.79. List Load Balancing Algorithms Response: JSON .....                    | 58 |
| 4.80. View Load Balancing SSL Termination Configuration Response: XML .....  | 61 |
| 4.81. View Load Balancing SSL Termination Configuration Response: JSON ..... | 64 |

|  |    |
|--|----|
| 4.82. Update Load Balancing SSL Termination Full Certification Request: XML .....  | 66 |
| 4.83. Update Load Balancing SSL Termination Full Certification Request: JSON ..... | 69 |
| 4.84. Update Load Balancing SSL Termination Attribute Request: XML .....           | 70 |
| 4.85. Update Load Balancing SSL Termination Attribute Request: JSON .....          | 70 |



# 1. Overview

## 1.1. Intended Audience

This guide is intended for software developers who want to create applications using the Rackspace Cloud Load Balancers API. It assumes the reader has a general understanding of load balancing concepts and is familiar with:

- ReSTful web services
- HTTP/1.1 conventions
- JSON and/or XML serialization formats
- Atom syndication format

## 1.2. Document Change History

This version of the Developer Guide replaces and obsoletes all previous versions. The most recent changes are described in the table below:

| Revision Date | Summary of Changes   |
|---------------|--|
| Mar 9, 2012   | <ul style="list-style-type: none"><li>• Added note about ServiceNet in Section 3.2, "Service Access/Endpoints".</li><li>• Added note to customers describing who should not use SSL Termination in Section 4.13, "SSL Termination".</li><li>• Corrected note below Example 4.81 "View Load Balancing SSL Termination Configuration Response: JSON" about wrapping the key/certificates in a newline character in Java in Section 4.13, "SSL Termination".</li></ul>  |
| Mar 7, 2012   | <ul style="list-style-type: none"><li>• Updated endpoints for authentication for Section 3.1.1, "Geographic Endpoints".</li><li>• Updated Auth Response examples for Section 3.1.2, "Retrieving the Authentication Token" to show the latest service catalog.</li><li>• Updated Section 3.7.3, "Determining Limits Programmatically" to fix limits and absolutelimits API calls.</li><li>• Fixed Create Load Balancer (Required Attributes) Request: XML example to be consistent with the JSON example in Section 4.1.3, "Create Load Balancer".</li><li>• Added Section 4.13, "SSL Termination" for new SSL termination feature.</li></ul> |
| Jan 27, 2012  | <ul style="list-style-type: none"><li>• Deleted section 2.5.1 "Passive Health Monitor (No Longer Available)" since the passive health monitor has been disabled.</li></ul>   |
| Jan 16, 2012  | <ul style="list-style-type: none"><li>• Updated information in Table 4.1, "Virtual IP Types" to reflect that a default IPV6 is being provisioned when creating a load balancer.</li></ul>  |
| Jan 4, 2012   | <ul style="list-style-type: none"><li>• Added support for Auth 1.1 in Section 3.1, "Authentication".</li><li>• Updated information for Section 3.2, "Service Access/Endpoints".</li><li>• A default IPV6 is being provisioned when creating a load balancer. Refer to Table 4.1, "Virtual IP Types". (This was turned off in Release 1.6.13 as a temporary mitigation strategy.)</li><li>• Added Section 4.3, "List Load Balancer Stats".</li></ul>  |
| Nov 22, 2011  | <ul style="list-style-type: none"><li>• Updated information in Table 4.1, "Virtual IP Types".</li></ul>  |
| Nov 18, 2011  | <ul style="list-style-type: none"><li>• Updated Section 4.4.1, "List Nodes" with note about <code>weight</code> attributes.</li></ul>  |
| Nov 17, 2011  | <ul style="list-style-type: none"><li>• Updated section 2.5.1 "Passive Health Monitor (No Longer Available)".</li><li>• Added Section 3.9, "Date/Time Format".</li><li>• Updated Section 4.1.1, "List Load Balancers" with information for query parameter <code>nodeaddress</code>.</li><li>• Added Section 4.2, "Error Page".</li></ul>  |
| Oct 13, 2011  | <ul style="list-style-type: none"><li>• Updated examples as needed to match the latest API.</li></ul>  |
| Oct 11, 2011  | <ul style="list-style-type: none"><li>• Updated Atom examples.</li></ul>   |
| Sep 26, 2011  | <ul style="list-style-type: none"><li>• Updated Section 3.7.2, "Absolute Limits".</li></ul>  |

| Revision Date | Summary of Changes   |
|---------------|--|
| Sep 7, 2011   | <ul style="list-style-type: none"><li>• Added basic TCP load balancing to allow customers to load balance traffic for any TCP-based protocol. SeeSection 4.11, "Protocols".</li><li>• An X-Forwarded-For (XFF) HTTP header is now automatically inserted for all inbound HTTP requests. SeeSection 4.1.3, "Create Load Balancer".</li><li>• Added programmatic support for determining Cloud Load Balancer source addresses. See Section 4.1.2, "List Load Balancer Details" andSection 4.1.3, "Create Load Balancer".</li><li>• Added support for batch delete of Load Balancers, Nodes, Virtual IPs, and Access List Entries. SeeSection 4.1.5, "Remove Load Balancer",Section 4.4.4, "Remove Nodes", Section 4.5.3, "Remove Virtual IP", andSection 4.7.1, "Manage Access Lists", respectively.</li><li>• UpdatedSection 3.7.2, "Absolute Limits"and added <code>absolutelimits</code> API call to Section 3.7.3, "Determining Limits Programmatically" to return the current absolute limits for the account.</li><li>• Updated examples to match the current API.</li></ul> |
| Jul 8, 2011   | <ul style="list-style-type: none"><li>• Added service endpoint for London (LON) region.</li></ul>  |
| Jul 7, 2011   | <ul style="list-style-type: none"><li>• Added support for accessing the Authentication Service, based on whether your account is US-based or UK-based.</li></ul>   |
| Jun 29, 2011  | <ul style="list-style-type: none"><li>• Added support for IPv6.</li></ul>  |
| Apr 19, 2011  | <ul style="list-style-type: none"><li>• Added details to support initial GA release.</li></ul>   |
| Mar 2, 2011   | <ul style="list-style-type: none"><li>• Revised code samples and formatting to address initial beta feedback.</li></ul>  |
| Feb 23, 2011  | <ul style="list-style-type: none"><li>• Initial release for public beta.</li></ul>   |

## 1.3. Additional Resources

You can download the most current version of this document from the Rackspace Cloud website at <http://docs.rackspacecloud.com/loadbalancers/api/clb-devguide-latest.pdf>.

For information about getting started using Cloud Load Balancers and Cloud Servers, refer to *Getting Started with Rackspace Cloud Load Balancers and Servers* at <http://docs.rackspacecloud.com/loadbalancers/api/clb-getting-started-latest.pdf>.

For more details about the Rackspace Cloud Load Balancer service, refer to [http://www.rackspace.com/cloud/cloud\\_hosting\\_products/loadbalancers/](http://www.rackspace.com/cloud/cloud_hosting_products/loadbalancers/). This site also offers links to Rackspace's official support channels, including knowledge base articles, forums, phone, chat, and email.

You can also follow updates and announcements via twitter at <http://www.twitter.com/Rackspace>.

## 2. Concepts

To use the Rackspace Cloud Load Balancers API effectively, you should understand several key concepts:

### 2.1. Load Balancer

A load balancer is a logical device which belongs to a cloud account. It is used to distribute workloads between multiple back-end systems or services, based on the criteria defined as part of its configuration.

### 2.2. Virtual IP

A virtual IP is an Internet Protocol (IP) address configured on the load balancer for use by clients connecting to a service that is load balanced. Incoming connections are distributed to back-end nodes based on the configuration of the load balancer.

### 2.3. Error Page

An error page is the html file that is shown to the end user when an error in the service has been thrown. By default every virtual server is provided with the default error file. It is also possible to submit a custom error page via the Load Balancers API. Refer to Section 4.2.3, "Error Page Operations" for details.

### 2.4. Node

A node is a back-end device providing a service on a specified IP and port.

### 2.5. Health Monitor

A health monitor is a configurable feature of each load balancer. It is used to determine whether or not a back-end node is usable for processing a request. The load balancing service currently supports active health monitoring.

#### 2.5.1. Active Health Monitor

Active health monitoring is a technique that uses synthetic transactions executed at periodic intervals to determine the condition of a node. One of the advantages of active health monitoring is that it does not require active transactions to be processed by the load balancer to determine whether or not a node is suitable for handling traffic. Active health monitoring is not applied by default and must be enabled per load balancer.

The active health monitor can use one of three types of probes:

- connect
- HTTP
- HTTPS

These probes are executed at configured intervals; in the event of a failure, the node status changes to `OFFLINE` and the node will not receive traffic. If, after running a subsequent test, the probe detects that the node has recovered, then the node's status is changed to `ONLINE` and it is capable of servicing requests.

## 2.6. Session Persistence

Session persistence is a feature of the load balancing service. It attempts to force subsequent connections to a service to be redirected to the same node as long as it is online.

## 2.7. Connection Logging

The connection logging feature allows logs to be delivered to a Cloud Files account every hour. For HTTP-based protocol traffic, these are Apache-style access logs. For all other traffic, this is connection and transfer logging.

## 3. General API Information

Ideas explained here are relevant to all operations of the API. For details of specific API operations, see the "API Operations" chapter.

The Rackspace Cloud Load Balancers API is implemented using a ReSTful web service interface. Like other products in the Rackspace Cloud suite, the load balancing service shares a common token-based authentication system that allows seamless access between products and services.



### Note

All requests to authenticate and operate the service are performed using HTTPS on TCP port 443.

## 3.1. Authentication

Every ReST request against the Load Balancers Service requires the inclusion of a specific authorization token, supplied by the `X-Auth-Token` HTTP header. Customers obtain this token by first using the Rackspace Cloud Authentication Service and supplying a valid username and API access key.

### 3.1.1. Geographic Endpoints

The Rackspace Cloud Authentication Service serves as the entry point to all Rackspace Cloud APIs and is itself a ReSTful web service.

To access the Authentication Service, you must know whether your account is US-based or UK-based:

- US-based accounts authenticate through <https://identity.api.rackspacecloud.com/v1.1/>.
- UK-based accounts authenticate through <https://lon.identity.api.rackspacecloud.com/v1.1/>.

Your account may be based in either the US or the UK; this is not determined by your physical location but by the location of the Rackspace retail site which was used to create your account:

- If your account was created via <http://www.rackspacecloud.com>, it is a US-based account.
- If your account was created via <http://www.rackspace.co.uk>, it is a UK-based account.

If you are unsure how your account was created, use the Rackspace contact information at either site to ask for help.

### 3.1.2. Retrieving the Authentication Token

| Verb | URI   | Description  |
|------|-------|--|
| POST | /auth | Authenticate to receive a token and a service catalog. |

Normal Response Code(s): 200, 203

Error Response Code(s): unauthorized (401), userDisabled (403), badRequest (400), authFault (500), serviceUnavailable (503)

The authenticate operation provides clients with an authentication token and a list of regional cloud endpoints.

### Example 3.1. Auth Request: XML

```
<?xml version="1.0" encoding="UTF-8"?>

<credentials xmlns="http://docs.rackspacecloud.com/auth/api/v1.1"
  username=❶"hub_cap"
  key=❷"a86850deb2742ec3cb41518e26aa2d89" />
```

### Example 3.2. Auth Request: JSON

```
{
  "credentials": {
    "username": ❶"hub_cap",
    "key": ❷"a86850deb2742ec3cb41518e26aa2d89"
  }
}
```

- ❶ The username supplied here is your common Rackspace Cloud username.
- ❷ The key is your API access key. The key can be obtained from the Rackspace Cloud Control Panel in the **Your Account / API Access** section.

### Example 3.3. Auth Response: XML

```
<?xml version="1.0" encoding="UTF-8"?>

<auth xmlns="http://docs.rackspacecloud.com/auth/api/v1.1">
  <token id="asdasdasd-adsasdasd-asdasdasd-adsadsasd"
    expires=❶"2010-11-01T03:32:15-05:00" />
  <serviceCatalog❷>
    <service name=❸"cloudFiles">
      <endpoint
        region=❹"DFW"
        v1Default=❺"true"
        publicURL=❻"https://storage.clouddrive.com/v1/RackCloudFS_demo"
        internalURL=❼"https://storage-snet.clouddrive.com/v1/RackCloudFS_demo" />
      <endpoint
        region="ORD"
        publicURL="https://otherstorage.clouddrive.com/v1/RackCloudFS_demo"
        internalURL="https://otherstorage-snet.clouddrive.com/v1/RackCloudFS_demo" />
    </service>
    <service name="cloudFilesCDN">
      <endpoint
        region="DFW"
        v1Default="true"
```

```

        publicURL="https://cdn.clouddrive.com/v1/RackCloudFS_demo" />
    </endpoint>
    region="ORD"
    publicURL="https://othercdn.clouddrive.com/v1/RackCloudFS_demo" />
</service>
<service name="cloudServers">
    <endpoint>
        vlDefault="true"
        publicURL="https://servers.api.rackspacecloud.com/v1.0/322781" />
    </service>
    <service name="cloudServersOpenStack">
        <endpoint>
            region="DFW"
            vlDefault="false"
            publicURL="https://dfw.servers.api.rackspacecloud.com/v2/
322781" />
        </service>
    </serviceCatalog>
</auth>

```

### Example 3.4. Auth Response: JSON

```

{
  "auth":{
    "token":{
      "id":"asdasdasd-adsasdads-asdasdasd-adsadsasd",
      "expires":❶"2010-11-01T03:32:15-05:00"
    },
    "serviceCatalog"❷:{
      "cloudFiles"❸:[
        {
          "region":❹"DFW",
          "vlDefault":❺true,
          "publicURL":❻"https://storage.clouddrive.com/v1/
RackCloudFS_demo",
          "internalURL":❼"https://storage-snet.clouddrive.com/v1/
RackCloudFS_demo"
        },
        {
          "region":"ORD",
          "publicURL":"https://otherstorage.clouddrive.com/v1/
RackCloudFS_demo",
          "internalURL":"https://otherstorage-snet.clouddrive.com/
v1/RackCloudFS_demo"
        }
      ],
      "cloudFilesCDN":[
        {
          "region":"DFW",
          "vlDefault":true,
          "publicURL":"https://cdn.clouddrive.com/v1/
RackCloudFS_demo"
        },
        {
          "region":"ORD",
          "publicURL":"https://othercdn.clouddrive.com/v1/
RackCloudFS_demo"
        }
      ]
    }
  }
}

```

```

    }
  ],
  "cloudServers": [
    {
      "v1Default": true,
      "publicURL": "https://servers.api.rackspacecloud.com/v1.0/
322781"
    }
  ],
  "cloudServersOpenStack" : [
    {
      "v1Default" : false,
      "publicURL" : "https://dfw.servers.api.rackspacecloud.com/
v2/322781",
      "region" : "DFW"
    }
  ]
}
}
}

```

- ❶ This token can be presented to a service as evidence of authentication. Tokens are valid for a finite duration.

The `expires` attribute denotes the time after which the token will automatically become invalid. A token may be manually revoked before the time identified by the `expires` attribute; `expires` predicts a token's maximum possible lifespan but does not guarantee that it will reach that lifespan. Clients are encouraged to cache a token until it expires.

- ❷ The service catalog lists the services this user can access. In this example, the user can access two compute services (Cloud Servers OpenStack and Cloud Servers) and two object storage services (Cloud Files Content Distribution Network (CDN), and Cloud Files). The catalog listing for each service provides at least one endpoint URL for that service. Other information, such as regions and versions and tenants, is provided if it's relevant to this user's access to this service.
- ❸ The service name attribute identifies each unique service in the catalog. Once a service is created, its name does not change.
- ❹ A service may expose endpoints in different regions. Regional endpoints allow clients to provision resources in a manner that provides high availability.

Some services are not region-specific. These services supply a single non-regional endpoint and do not provide access to internal URLs.

- ❺ The `v1Default` attribute denotes that an endpoint is being returned in version 1.0 of the Cloud Authentication Service. The default value of `v1Default` is `false`; clients should assume the value is `false` when the attribute is missing. Auth 1.0 does not offer support for regional endpoints and therefore only returns one endpoint per service. Resources stored in endpoints where `v1Default` is `false` will not be seen by Auth 1.0 clients.
- ❻ An endpoint can be assigned public and internal URLs. A public URL is accessible from anywhere. Access to a public URL usually incurs traffic charges. Internal URLs are only accessible to services within the same region. Access to an internal URL is free of charge.



Authentication tokens are typically valid for 24 hours. Applications should be designed to re-authenticate after receiving a 401 (Unauthorized) response from a service endpoint.



### Important

If you are programmatically parsing an authentication response, please be aware that service names are stable for the life of the particular service and can be used as keys. You should also be aware that a user's service catalog can include multiple uniquely-named services which perform similar functions. For example, `cloudServersOpenStack` is the OpenStack version of compute whereas `cloudServers` is the legacy version of compute; the same user can have access to both services. In Auth 2.0, the service type attribute can be used as a key by which to recognize similar services; see the tip below.



### Tip

Beginning with Auth 2.0, the service catalog includes a service type attribute to identify services that perform similar functions but have different names; for example, `type= "compute"` identifies compute services such as `cloudServers` and `cloudServersOpenStack`. Some developers have found the service type attribute to be useful in parsing the service catalog. For Auth 2.0 (also known as the Cloud Identity Service), you can see the service type attribute in the "Authentication Response with Headers" sample in the Cloud Identity Client Developer Guide at [http://docs.rackspace.com/auth/api/v2.0/auth-client-devguide/content/Request\\_Response\\_Types-d1e149.html](http://docs.rackspace.com/auth/api/v2.0/auth-client-devguide/content/Request_Response_Types-d1e149.html).



### Note

Currently, load balancing service endpoints are not published in the Auth response with the account number, which is a required element of the service endpoints. Refer to the next section for details about service endpoints.

## 3.2. Service Access/Endpoints

The load balancing service is a regionalized service. It allows the caller to select a region into which a load balancer is to be provisioned.

To determine which region to operate against, select an endpoint from the table below. Replace the sample account ID number, `1234`, with your actual account number returned as part of the authentication service response. You will find the actual account number after the final `/` in the `publicURL` field returned by the authentication response. In Example 3.4, "Auth Response: JSON", the account id is `322781`, as you can see from the `publicURL` field for `cloudServers`: `"https://servers.api.rackspacecloud.com/v1.0/322781"`.

**Table 3.1. Regionalized Service Endpoints**

| Region                 | Endpoint  |
|------------------------|---|
| Chicago (ORD)          | <code>https://ord.loadbalancers.api.rackspacecloud.com/v1.0/ 1234/</code> |
| Dallas/Ft. Worth (DFW) | <code>https://dfw.loadbalancers.api.rackspacecloud.com/v1.0/ 1234/</code> |
| London (LON)           | <code>https://lon.loadbalancers.api.rackspacecloud.com/v1.0/ 1234/</code> |

If load balancing Cloud Servers, you can determine the appropriate region to select by viewing your Cloud Servers list and creating a load balancer within the same region as the datacenter in which your Cloud Server resides. When your resources reside in the same region as your load balancer, devices are in close proximity to each other and can take advantage of ServiceNet connectivity for free data transfer between services.



### Note

ServiceNet is an internal Rackspace-only, multi-tenant network connection within each Rackspace datacenter. ServiceNet IPs are not accessible via the public internet. Rackspace customers may configure resources to utilize an internal IP address so that traffic over the ServiceNet network is not billed.

If load balancing external servers, you can determine the appropriate region to select by choosing the region that is geographically as close to your external servers as possible.

## 3.3. Request/Response Types

The Rackspace Cloud Load Balancers API supports both the JSON and XML data serialization formats. The request format is specified using the `Content-Type` header and is required for operations that have a request body. The response format can be specified in requests using either the `Accept` header or adding an `.xml` or `.json` extension to the request URI. Note that it is possible for a response to be serialized using a format different from the request. If no response format is specified, JSON is the default. If conflicting formats are specified using both an `Accept` header and a query extension, the query extension takes precedence.

Some operations support an Atom representation that can be used to efficiently determine when the state of services has changed.

**Table 3.2. JSON and XML Response Formats**

| Format | Accept Header        | Query Extension | Default |
|--------|----------------------|-----------------|---------|
| JSON   | application/json     | .json           | Yes     |
| XML    | application/xml      | .xml            | No      |
| Atom   | application/atom+xml | .atom           | No      |

## 3.4. Content Compression

Request and response body data may be encoded with gzip compression to accelerate interactive performance of API calls and responses. This is controlled using the `Accept-Encoding` header in the request from the client and indicated by the `Content-Encoding` header in the server response. Unless the header is explicitly set, encoding defaults to disabled.

**Table 3.3. Encoding Headers**

| Header Type       | Name             | Value |
|-------------------|------------------|-------|
| HTTP/1.1 Request  | Accept-Encoding  | gzip  |
| HTTP/1.1 Response | Content-Encoding | gzip  |

## 3.5. Persistent Connections

By default, the API supports persistent connections via HTTP/1.1 keepalives. All connections will be kept alive unless the connection header is set to close.

To prevent abuse, HTTP sessions have a timeout of 20 seconds before being closed.



### Note

The server may close the connection at any time and clients should not rely on this behavior.

## 3.6. Paginated Collections

To reduce load on the service, list operations will return a maximum of 100 items at a time. To navigate the collection, the `limit` and `marker` parameters (for example, `?limit=50&marker=1`) can be set in the URI. If a marker beyond the end of a list is given, an empty list is returned. Note that list operations never return 404 (itemNotFound) faults.

## 3.7. Limits

All accounts, by default, have a preconfigured set of thresholds (or limits) to manage capacity and prevent abuse of the system. The system recognizes two kinds of limits: rate limits and absolute limits. Rate limits are thresholds that are reset after a certain amount of time passes. Absolute limits are fixed.



### Note

If the default limits are too low for your particular application, please contact Rackspace Cloud support to request an increase. All requests require reasonable justification.

### 3.7.1. Rate Limits

We specify rate limits in terms of both a human-readable wild-card URI and a machine-processable regular expression. The regular expression boundary matcher '^' takes effect after the root URI path. For example, the regular expression `^/v1.0/1234/loadbalancers` would match the bolded portion of the following URI: `https://ord.loadbalancers.api.rackspacecloud.com /v1.0/1234/loadbalancers`.

**Table 3.4. Default Rate Limits**

| Verb   | URI     | RegEx    | Default Limit |
|--------|---------|----------|---------------|
| GET    | /v1.0/* | ^/1.0/.* | 5/second      |
| GET    | /v1.0/* | ^/1.0/.* | 100/minute    |
| POST   | /v1.0/* | ^/1.0/.* | 2/second      |
| POST   | /v1.0/* | ^/1.0/.* | 25/minute     |
| PUT    | /v1.0/* | ^/1.0/.* | 5/second      |
| PUT    | /v1.0/* | ^/1.0/.* | 50/minute     |
| DELETE | /v1.0/* | ^/1.0/.* | 2/second      |
| DELETE | /v1.0/* | ^/1.0/.* | 50/minute     |

Rate limits are applied in order relative to the verb, going from least to most specific. For example, although the threshold for **POST** to `/v1.0/*` is 25 per minute, one cannot **POST** to `/v1.0/*` more than 2 times per second because the rate limit for any **POST** is 2 per second. In the event you exceed the thresholds established for your account, a 413 (Rate Control) HTTP response will be returned with a `Retry-After` header to notify the client when they can attempt to try again.

A request may be submitted to Cloud Support for an increase in load balancer limits. Each request must be approved before limits can be modified. Limits may only be increased up to the maximum limit (such as 50 nodes per load balancer).

Refer to Section 3.7.3, “Determining Limits Programmatically” to find your current account settings for these limits.

## 3.7.2. Absolute Limits

Absolute limits specify the maximum number of load balancers that can exist per Cloud account and the maximum number of resources that can exist per load balancer. The batch delete limit is the exception, since it is applied per batch delete API request.

The system applies default values for load balancer account and/or resource limits that can be increased by submitting a request to Cloud Support. Each request must be approved internally before limits can be modified.

| Name               | Description  | Default Limit |
|--------------------|--|---------------|
| LOADBALANCER_LIMIT | Total number of load balancers that can be added to a Cloud account.   | 25            |
| NODE_LIMIT         | Total number of nodes that can be added to a load balancer.            | 25            |
| IPV6_LIMIT         | Total number of IPV6 addresses that can be added to a load balancer.   | 25            |
| BATCH_DELETE_LIMIT | Total number of resources that can be listed per batch delete request. | 10            |
| ACCESS_LIST_LIMIT  | Total number of network items that can be added to a load balancer.    | 100           |

Refer to Section 3.7.3, “Determining Limits Programmatically” to find your current account settings for these limits.

### 3.7.3. Determining Limits Programmatically

Applications can programmatically determine current rate limits and absolute limits for an account using the following URIs:

| Verb | URI                           | Description   |
|------|-------------------------------|---|
| GET  | /loadbalancers/limits         | Return the current rate limits for the account.     |
| GET  | /loadbalancers/absolutelimits | Return the current absolute limits for the account. |

Normal Response Code(s): 200

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

These operations do not require a request body.

#### Example 3.5. List Rate Limits Response: XML

```
<limits xmlns="http://docs.openstack.org/common/api/v1.0">
  <rates>
    <rate uri="/v1.0/*" regex="^/1.0/.*">
      <limit
        verb="GET"
        value="600000"
        remaining="426852"
        unit="HOUR"
        next-available="2011-02-22T19:32:43.835Z"/>
    </rate>
  </rates>
</limits>
```

#### Example 3.6. List Rate Limits Response: JSON

```
{
  "limits" : {
    "rate" : {
      "values": [
        {
          "uri" : "/v1.0/*",
          "regex" : "^/1.0/.*",
          "limit" : [
            {
              "verb" : "GET",
              "value" : 600000,
              "remaining" : 426852,
              "unit" : "HOUR",
              "next-available" : "2011-02-22T19:32:43.835Z"
            }
          ]
        }
      ]
    }
  }
}
```

```
}  
}
```

### Example 3.7. List Absolute Limits Response: XML

```
<limits xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">  
  <absolute>  
    <limit name="IPV6_LIMIT" value="25"/>  
    <limit name="LOADBALANCER_LIMIT" value="25"/>  
    <limit name="BATCH_DELETE_LIMIT" value="10"/>  
    <limit name="ACCESS_LIST_LIMIT" value="100"/>  
    <limit name="NODE_LIMIT" value="25"/>  
  </absolute>  
</limits>
```

### Example 3.8. List Absolute Limits Response: JSON

```
{  
  "absolute":  
    [  
      {"name": "IPV6_LIMIT", "value": 25},  
      {"name": "LOADBALANCER_LIMIT", "value": 25},  
      {"name": "BATCH_DELETE_LIMIT", "value": 10},  
      {"name": "ACCESS_LIST_LIMIT", "value": 100},  
      {"name": "NODE_LIMIT", "value": 25}  
    ]  
}
```

## 3.8. Faults

API calls that return an error return one of the following fault objects. All fault objects extend from the base `fault`, `serviceFault`, for easier exception handling for languages that support it.

### 3.8.1. `serviceFault`

The `serviceFault` and by extension all other faults include `message` and `detail` elements which contain strings describing the nature of the fault as well as a `code` attribute representing the HTTP response code for convenience. The `code` attribute of the fault is for the convenience of the caller so that they may retrieve the response code from the HTTP response headers or directly from the fault object if they choose. The caller should not expect the `serviceFault` to be returned directly but should instead expect only one of the child faults to be returned.

### 3.8.2. `badRequest`

This fault indicates that the data in the request object is invalid; for example, a string was used in a parameter that was expecting an integer. The fault will wrap validation errors.

### Example 3.9. Fault Response, `badRequest`

```
<badRequest xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  code="400">
  <message>Validation fault</message>
  <details>The object is not valid</details>
  <validationErrors>
    <message>Node ip is invalid. Please specify a valid ip.</message>
  </validationErrors>
</badRequest>
```

## 3.8.3. `immutableEntity`

This fault is returned when a user attempts to modify an item that is not currently in a state that allows modification. For example, load balancers in a status of `PENDING_UPDATE`, `BUILD`, or `DELETED` may not be modified.

### Example 3.10. Fault Response, `immutableEntity`

```
<immutableEntity code="422" xmlns="http://docs.openstack.org/loadbalancers/
api/v1.0">
  <message>The object at the specified URI is immutable and can not be
  overwritten.</message>
</immutableEntity>
```

## 3.8.4. `itemNotFound`

### Example 3.11. Fault Response, `itemNotFound`

```
<itemNotFound code="404" xmlns="http://docs.openstack.org/loadbalancers/api/
v1.0">
  <message>Object not Found</message>
</itemNotFound>
```

## 3.8.5. `loadBalancerFault`

The `loadBalancerFault` fault shall be returned in the event that an error occurred during a load balancer operation.

### Example 3.12. Fault Response, `loadBalancerFault`

```
<loadBalancerFault code="401" xmlns="http://docs.openstack.org/loadbalancers/
api/v1.0">
  <message>Invalid authentication token. Please renew</message>
</loadBalancerFault>
```

### 3.8.6. outOfVirtualIps

This fault indicates that there are no virtual IPs left to assign to a new load balancer. In practice, this fault should not occur, as virtual IPs will be ordered as capacity is required. If you do experience this fault, contact support so that we may make more IPs available.

#### Example 3.13. Fault Response, outOfVirtualIps

```
<outOfVirtualIps code="500" xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <message>
    Out of virtual IPs. Please contact support so they can allocate more
    virtual IPs.
  </message>
</outOfVirtualIps>
```

### 3.8.7. overLimit

This fault is returned when the user has exceeded a currently allocated limit.

#### Example 3.14. Fault Response, overLimit

```
<overLimit code="413" xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <message>Your account is currently over the limit so your request could
  not be processed.</message>
</overLimit>
```

### 3.8.8. serviceUnavailable

This fault is returned when the service is unavailable, such as when the service is undergoing maintenance. Note that this does not necessarily mean that the currently configured loadbalancers are unable to process traffic; it simply means that the API is currently unable to service requests.

#### Example 3.15. Fault Response, serviceUnavailable

```
<serviceUnavailable code="500" xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <message>The Load balancing service is currently not available</message>
</serviceUnavailable>
```

### 3.8.9. unauthorized

This fault is returned when the user is not authorized to perform an attempted operation.



### Example 3.16. Fault Response, unauthorized

```
<unauthorized code="404" xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <message>You are not authorized to execute this operation.</message>
</unauthorized>
```

## 3.8.10. unprocessableEntity

This fault is returned when an operation is requested on an item that does not support the operation, but the request is properly formed.

### Example 3.17. Fault Response, unprocessableEntity

```
<unprocessableEntity code="422" xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <message>The Object at the specified URI is unprocessable.</message>
</unprocessableEntity>
```

## 3.9. Date/Time Format

The Load Balancer service uses an ISO-8601 compliant date format for the display and consumption of date/time values.

### Example 3.18. Load Balancer Service Date/Time Format

```
yyyy-MM-dd'T'HH:mm:ssZ
```

See the table below for a description of the date/time format codes.

May 19th, 2011 at 8:07:08 AM, GMT-5 would have the following format:

```
2011-05-19T08:07:08-0500
```

**Table 3.5. Explanation of Date/Time Format Codes**

| Code | Description                        |
|------|------------------------------------|
| yyyy | Four digit year                    |
| MM   | Two digit month                    |
| dd   | Two digit day of month             |
| T    | Separator for date/time            |
| HH   | Two digit hour of day (00-23)      |
| mm   | Two digit minutes of hour          |
| ss   | Two digit seconds of the minute    |
| Z    | RFC-822 timezone (offset from GMT) |

## 4. API Operations

This chapter explains specific API operations. For ideas relevant to all API operations, see the "General API Information" chapter.

### 4.1. Load Balancers

#### 4.1.1. List Load Balancers

| Verb | URI            | Description  | Representation  |
|------|----------------|--|-----------------|
| GET  | /loadbalancers | List all load balancers configured for the account (IDs, names and status only). | XML, JSON, Atom |

Normal Response Code(s): 200

Error Response Code(s): loadBalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation provides a list of all load balancers configured and associated with your account.

A limited set of load balancer attributes including the following are returned in the response object:

- name
- id
- status
- created
- updated

This operation does not require a request body.

List load balancer example responses are shown below:

#### Example 4.1. List Load Balancers Response: XML

```
<?xml version="1.0" ?>
<loadBalancers xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <loadBalancer id="71" name="lb-site1" status="ACTIVE"
    protocol="HTTP" port="80" algorithm="RANDOM">
    <virtualIps>
      <virtualIp id="403" address="206.55.130.1" ipVersion="IPV4"
        type="PUBLIC" />
    </virtualIps>
    <created time="2010-12-13T15:38:27-06:00" />
    <updated time="2010-12-13T15:38:38-06:00" />
  </loadBalancer>
  <loadBalancer id="166" name="lb-site2" status="ACTIVE"
    protocol="HTTP" port="80" algorithm="RANDOM">
    <virtualIps>
```

```
<virtualIp id="401" address="206.55.130.2" ipVersion="IPV4"
  type="PUBLIC" />
</virtualIps>
<created time="2010-12-13T15:38:27-06:00" />
<updated time="2010-12-13T15:38:38-06:00" />
</loadBalancer>
</loadBalancers>
```

### Example 4.2. List Load Balancers Response: JSON

```
{
  "loadBalancers": [
    {
      "name": "lb-site1",
      "id": 71,
      "protocol": "HTTP",
      "port": 80,
      "algorithm": "RANDOM",
      "status": "ACTIVE",
      "virtualIps": [
        {
          "id": 403,
          "address": "206.55.130.1",
          "type": "PUBLIC",
          "ipVersion": "IPV4"
        }
      ],
      "created": {
        "time": "2010-11-30T03:23:42Z"
      },
      "updated": {
        "time": "2010-11-30T03:23:44Z"
      }
    },
    {
      "name": "lb-site2",
      "id": 166,
      "protocol": "HTTP",
      "port": 80,
      "algorithm": "RANDOM",
      "status": "ACTIVE",
      "virtualIps": [
        {
          "id": 401,
          "address": "206.55.130.2",
          "type": "PUBLIC",
          "ipVersion": "IPV4"
        }
      ],
      "created": {
        "time": "2010-11-30T03:23:42Z"
      },
      "updated": {
        "time": "2010-11-30T03:23:44Z"
      }
    }
  ]
}
```

### Example 4.3. List Atom Load Balancers Response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <link rel="next"
        href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/1234/
loadbalancers.atom?page=2"/>
  <title type="text">Parent Feed</title>
  <id>1234-loadbalancers</id>
  <author>
    <name>Rackspace Cloud</name>
  </author>
  <entry>
    <title type="text">Access List Updated</title>
    <summary
      type="text">Access list successfully updated with the
following network item: id: '2155', address: '206.160.163.210', type: 'DENY'
    </summary>
    <author>
      <name>tvardema</name>
    </author>
    <link href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/
1234/loadbalancers/141/accesslist/" />
    <id>1234-loadbalancers-141-accesslist-2011971658310</id>
    <category term="UPDATE" />
    <updated>2011-04-07T16:58:31.000Z</updated>
  </entry>
</feed>
```

A *changes-since* query parameter is supported that lists all load balancers that have changed since the specified date/time. For example, using the URI `/loadbalancers?changes-since=2011-05-19T08:07:08-0500` lists all load balancers that have changed since May 19th, 2011 at 8:07:08 AM, GMT-5. Refer to Section 3.9, “Date/Time Format” for information about specifying the date/time.

To view deleted load balancers, add the query parameter `?status=DELETED` to the end of the URI. A deleted load balancer is immutable and irrecoverable.

To find out if any load balancer for an account has a specific node attached to it, use the query parameter *nodeaddress* to specify the IP address for the desired node. For example, using the URI `/loadbalancers?nodeaddress=10.1.1.1` will search all load balancers for the account for a node with IP address 10.1.1.1, and return a list including the id, name, and status of all load balancers attached:

### Example 4.4. List Load Balancers With Node Specified: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<loadBalancers xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <loadBalancer id="3" name="First Loadbalancer" status="ACTIVE"/>
  <loadBalancer id="2" name="Second Loadbalancer" status="PENDING_UPDATE"/>
  <loadBalancer id="8" name="Third Loadbalancer" status="ERROR"/>
</loadBalancers>
```

### Example 4.5. List Load Balancers With Node Specified: JSON

```
{ "loadBalancers": [
  {
    "name": "First Loadbalancer",
    "id": 1,
    "status": "ACTIVE"
  },
  {
    "name": "Second Loadbalancer",
    "id": 2,
    "status": "PENDING_UPDATE"
  },
  {
    "name": "Third Loadbalancer",
    "id": 8,
    "status": "ERROR"
  }
]
```

## 4.1.2. List Load Balancer Details

| Verb | URI                                   | Description                                  | Representations |
|------|---------------------------------------|--|-----------------|
| GET  | /loadbalancers/ <i>loadBalancerId</i> | List details of the specified load balancer. | JSON, XML, Atom |

Normal Response Code(s): 200

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation provides detailed output for a specific load balancer configured and associated with your account. This operation is not capable of returning details for a load balancer which has been deleted.

This operation does not require a request body.

Notice in the following examples that API users are now able to programmatically derive the source IP addresses of our load balancers using the `sourceAddresses` label included at the bottom of the list load balancer details response. This feature is useful for customers who are automating the deployment of infrastructure and need to determine the IP addresses of requests coming from our load balancers for the purpose of creating more robust firewall rules.

### Example 4.6. List Load Balancer Details Response: XML

```
<loadBalancer xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  id="2000"
  name="sample-loadbalancer"
  protocol="HTTP"
```

```
port="80"
algorithm="RANDOM"
status="ACTIVE">
<connectionLogging enabled="false" />
<virtualIps>
  <virtualIp
    id="1000"
    address="206.10.10.210"
    type="PUBLIC"
    ipVersion="IPV4" />
</virtualIps>
<nodes>
  <node
    id="1041"
    address="10.1.1.1"
    port="80"
    condition="ENABLED"
    status="ONLINE" />
  <node
    id="1411"
    address="10.1.1.2"
    port="80"
    condition="ENABLED"
    status="ONLINE" />
</nodes>
<sessionPersistence persistenceType="HTTP_COOKIE" />
<connectionThrottle
  minConnections="10"
  maxConnections="100"
  maxConnectionRate="50"
  rateInterval="60" />
<cluster name="cl.dfw1" />
<created time="2010-11-30T03:23:42Z" />
<updated time="2010-11-30T03:23:44Z" />
<sourceAddresses ipv4Servicenet="10.0.0.0" ipv4Public="10.12.99.28"
  ipv6Public="2001:4801:79f1:1::1/64" />
</loadBalancer>
```

### Example 4.7. List Load Balancers Details Response: JSON

```
{
  "loadBalancer": {
    "id": 2000,
    "name": "sample-loadbalancer",
    "protocol": "HTTP",
    "port": 80,
    "algorithm": "RANDOM",
    "status": "ACTIVE",
    "connectionLogging": {
      "enabled": true
    },
    "virtualIps": [
      {
        "id": 1000,
        "address": "206.10.10.210",
        "type": "PUBLIC",
        "ipVersion": "IPV4"
      }
    ]
  }
}
```

```
    ],
    "nodes": [
      {
        "id": 1041,
        "address": "10.1.1.1",
        "port": 80,
        "condition": "ENABLED",
        "status": "ONLINE"
      },
      {
        "id": 1411,
        "address": "10.1.1.2",
        "port": 80,
        "condition": "ENABLED",
        "status": "ONLINE"
      }
    ],
    "sessionPersistence": {
      "persistenceType": "HTTP_COOKIE"
    },
    "connectionThrottle": {
      "minConnections": 10,
      "maxConnections": 100,
      "maxConnectionRate": 50,
      "rateInterval": 60
    },
    "cluster": {
      "name": "cl.dfw1"
    },
    "created": {
      "time": "2010-11-30T03:23:42Z"
    },
    "updated": {
      "time": "2010-11-30T03:23:44Z"
    },
    "accountLoadBalancerServiceEvents": {
      "accountId": 406271
    },
    "sourceAddresses": {
      "ipv6Public": "2001:4801:79f1:1::1/64",
      "ipv4ServiceNet": "10.0.0.0",
      "ipv4Public": "10.12.99.28"
    }
  }
}
```

#### Example 4.8. List Atom Load Balancer Response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <link rel="next"
        href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/1234/
loadbalancers/141.atom?page=2"/>
  <title type="text">Load Balancer Feed</title>
  <id>1234-loadbalancers-141</id>
  <author>
    <name>Rackspace Cloud</name>
  </author>
  <entry>
    <title type="text">Load Balancer Successfully Updated</title>
    <summary>
```

```
type="text">Load balancer successfully updated with algorithm:
'RANDOM', protocol: 'HTTP', port: '80'
</summary>
<author>
  <name>tvardema</name>
</author>
<link href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/
1234/loadbalancers/141"/>
<id>1234-loadbalancers-141-2011961339450</id>
<category term="UPDATE"/>
<updated>2011-04-06T13:39:45.000Z</updated>
</entry>
</feed>
```

### 4.1.3. Create Load Balancer

| Verb | URI            | Description   |
|------|----------------|---|
| POST | /loadbalancers | Create a new load balancer with the configuration defined by the request. |

Normal Response Code(s): 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation asynchronously provisions a new load balancer based on the configuration defined in the request object. Once the request is validated and progress has started on the provisioning process, a response object will be returned. The object will contain a unique identifier and status of the request. Using the identifier, the caller can check on the progress of the operation by performing a **GET** on `loadbalancers/id`. If the corresponding request cannot be fulfilled due to insufficient or invalid data, an HTTP 400 (Bad Request) error response will be returned with information regarding the nature of the failure in the body of the response. Failures in the validation process are non-recoverable and require the caller to correct the cause of the failure and **POST** the request again.

An HTTP load balancer will have the X-Forwarded-For (XFF) HTTP header set by default. This header will contain the actual source address of the client, which many web applications are already designed to use when determining the source address for a request.



#### Note

A load balancer's name must be less than or equal to 128 characters.



#### Note

Users may configure all documented features of the load balancer at creation time by simply providing the additional elements or attributes in the request. This document provides an overview of all the features the load balancing service supports.

You may specify the type of virtualip to add along with the creation of a loadbalancer. The following table describes the options.



**Table 4.1. Virtual IP Types**

| Version       | Type       | Outcome         |
|---------------|------------|-----------------|
| Not Specified | PUBLIC     | IPV4 & IPV6     |
| IPV4          | PUBLIC     | IPV4            |
| IPV6          | PUBLIC     | IPV6            |
| Not Specified | SERVICENET | IPV4            |
| IPV4          | SERVICENET | IPV4            |
| IPV6          | SERVICENET | Failure Message |

**Example 4.9. Create Load Balancer (Required Attributes) Request: XML**

```
<loadBalancer xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  name="a-new-loadbalancer"
  port="80"
  protocol="HTTP">
  <virtualIps>
    <virtualIp type="PUBLIC"/>
  </virtualIps>
  <nodes>
    <node address="10.1.1.1" port="80" condition="ENABLED"/>
  </nodes>
</loadBalancer>
```

**Example 4.10. Create Load Balancer (Required Attributes) Request: JSON**

```
{
  "loadBalancer": {
    "name": "a-new-loadbalancer",
    "port": 80,
    "protocol": "HTTP",
    "virtualIps": [
      {
        "type": "PUBLIC"
      }
    ],
    "nodes": [
      {
        "address": "10.1.1.1",
        "port": 80,
        "condition": "ENABLED"
      }
    ]
  }
}
```

Notice in the following examples that API users are now able to programmatically derive the source IP addresses of our load balancers using the `sourceAddresses` label included at the bottom of the create load balancer (required attributes) response. This feature is useful for customers who are automating the deployment of infrastructure and need to determine the IP addresses of requests coming from our load balancers for the purpose of creating more robust firewall rules.

**Example 4.11. Create Load Balancer (Required Attributes) Response: XML**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<loadBalancer xmlns="http://docs.openstack.org/loadbalancers/api/v1.0" id=
"2198" name="a-new-loadbalancer"
      algorithm="RANDOM" protocol="HTTP" port="80" status="BUILD">
  <virtualIps>
    <virtualIp id="13" address="10.0.0.16" ipVersion="IPV4" type="PUBLIC"/>
  >
    <virtualIp id="9000135" address=
"fd24:f480:ce44:91bc:1af2:15ff:0000:0003" ipVersion="IPV6" type="PUBLIC"/>
  </virtualIps>
  <nodes>
    <node id="2206" address="10.1.1.1" port="80" condition="ENABLED"
status="ONLINE" weight="1"/>
  </nodes>
  <cluster name="host2_cluster1"/>
  <created time="2011-06-01T08:08:41-05:00"/>
  <updated time="2011-06-01T08:08:41-05:00"/>
  <connectionLogging enabled="false"/>
  <sourceAddresses ipv4ServiceNet="10.0.0.0" ipv4Public="10.12.99.29"
ipv6Public="2001:4801:79f1:1::3/64"/>
</loadBalancer>
```

**Example 4.12. Create Load Balancer (Required Attributes) Response: JSON**

```
{
  "loadBalancer": {
    "name": "a-new-loadbalancer",
    "id": 2200,
    "port": 80,
    "protocol": "HTTP",
    "algorithm": "RANDOM",
    "status": "BUILD",
    "cluster": {
      "name": "host2_cluster1"
    },
    "nodes": [ {
      "address": "10.1.1.1",
      "id": 2208,
      "port": 80,
      "status": "ONLINE",
      "condition": "ENABLED",
      "weight": 1
    } ],
    "virtualIps": [ {
      "address": "10.0.0.18",
      "id": 15,
      "type": "PUBLIC",
      "ipVersion": "IPV4"
    },
    {
      "address": "fd24:f480:ce44:91bc:1af2:15ff:0000:0005",
      "id": 9000137,
      "type": "PUBLIC",
      "ipVersion": "IPV6"
    } ]
  }
}
```

```

    },
    "created": {
      "time": "2011-06-01T08:20:09-05:00"
    },
    "updated": {
      "time": "2011-06-01T08:20:09-05:00"
    },
    "connectionLogging": {
      "enabled": false
    },
    "sourceAddresses": {
      "ipv6Public": "2001:4801:79f1:1::1/64",
      "ipv4Servicenet": "10.0.0.0",
      "ipv4Public": "10.12.99.28"
    }
  }
}

```

To conserve IPv4 address space, Rackspace highly recommends sharing Virtual IPs between your load balancers. If you have at least one load balancer, you may create subsequent load balancers that share a single virtual IP by issuing a **POST** and supplying a virtual IP ID instead of a type. Additionally, this feature is highly desirable if you wish to load balance both an unsecured and secure protocol using one IP or DNS name. For example, this method makes it possible to use the same load balancing configuration to support HTTP and HTTPS.



### Note

Load balancers sharing a virtual IP *must* utilize a unique port.

### Example 4.13. Create Load Balancer (Required Attributes with Shared IP) Request: XML

```

<loadBalancer xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  name="a-new-loadbalancer"
  port="80"
  protocol="HTTP">
  <virtualIps>
    <virtualIp id="2341"/>
  </virtualIps>
  <nodes>
    <node address="10.1.1.1" port="80" condition="ENABLED" />
  </nodes>
</loadBalancer>

```

### Example 4.14. Create Load Balancer (Required Attributes with Shared IP) Request: JSON

```
{
  "loadBalancer": {
    "name": "a-new-loadbalancer",
    "port": 80,
    "protocol": "HTTP",
    "virtualIps": [
      {
        "id": 2341
      }
    ],
    "nodes": [
      {
        "address": "10.1.1.1",
        "port": 80,
        "condition": "ENABLED"
      }
    ]
  }
}
```

### Example 4.15. Create Load Balancer (Required Attributes with Shared IP) Response: XML

```
<loadBalancer xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  id="144"
  name="a-new-loadbalancer"
  algorithm="RANDOM"
  protocol="HTTP"
  port="83"
  status="BUILD">
  <virtualIps>
    <virtualIp
      id="39"
      address="206.10.10.210"
      ipVersion="IPV4"
      type="PUBLIC" />
  </virtualIps>
  <nodes>
    <node
      id="653"
      address="10.1.1.1"
      port="80"
      condition="ENABLED"
      status="ONLINE"
      weight="1" />
  </nodes>
  <cluster name="ztm-n03.staging1.lbaas.rackspace.net" />
  <created time="2011-02-08T21:19:55Z" />
  <updated time="2011-02-08T21:19:55Z" />
  <connectionLogging enabled="false" />
</loadBalancer>
```

### Example 4.16. Create Load Balancer (Required Attributes with Shared IP) Response: JSON

```
{
  "loadBalancer": {
    "name": "a-new-loadbalancer",
    "id": 144,
    "protocol": "HTTP",
    "port": 83,
    "algorithm": "RANDOM",
    "status": "BUILD",
    "cluster": {
      "name": "ztm-n01.staging1.lbaas.rackspace.net"
    },
    "nodes": [
      {
        "address": "10.1.1.1",
        "id": 653,
        "port": 80,
        "status": "ONLINE",
        "condition": "ENABLED",
        "weight": 1
      }
    ],
    "virtualIps": [
      {
        "address": "206.10.10.210",
        "id": 39,
        "type": "PUBLIC",
        "ipVersion": "IPV4"
      }
    ],
    "created": {
      "time": "2011-04-13T14:18:07Z"
    },
    "updated": {
      "time": "2011-04-13T14:18:07Z"
    },
    "connectionLogging": {
      "enabled": false
    }
  }
}
```

## 4.1.4. Update Load Balancer Attributes

| Verb | URI                                   | Description                               |
|------|---------------------------------------|---|
| PUT  | /loadbalancers/ <i>loadBalancerId</i> | Update the properties of a load balancer. |

Normal Response Code(s): 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation asynchronously updates the attributes of the specified load balancer. Upon successful validation of the request, the service will return a 202 (Accepted) response code.

A caller can poll the load balancer with its ID to wait for the changes to be applied and the load balancer to return to an `ACTIVE` status.

This operation allows the caller to change one or more of the following attributes:

- name
- algorithm
- protocol
- port

This operation does not return a response body.



### Note

The load balancer's ID and status are immutable attributes and cannot be modified by the caller. Supplying an unsupported attribute will result in a 400 (badRequest) fault.

### Example 4.17. Update Load Balancer Attributes Request: XML

```
<loadBalancer xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  name="sample-loadbalancer"
  algorithm="RANDOM"
  protocol="HTTP"
  port="80" />
```

### Example 4.18. Update Load Balancer Attributes Request: JSON

```
{ "loadBalancer": {
  "name": "sample-loadbalancer",
  "algorithm": "RANDOM",
  "protocol": "HTTP",
  "port": 80,
  "connectionLogging": true
}
```

All load balancers also have a status attribute to signify the current configuration status of the device. This status is immutable by the caller and is updated automatically based on state changes within the service. When a load balancer is first created, it will be placed into a `BUILD` status while the configuration is being generated and applied based on the request. Once the configuration is applied and finalized, it will be in an `ACTIVE` status. In the event of a configuration change or update, the status of the load balancer will change to `PENDING_UPDATE` to signify configuration changes are in progress but have not yet been finalized. Load balancers in a `SUSPENDED` status are configured to reject traffic and will not forward requests to back-end nodes.

**Table 4.2. Load Balancer Statuses**

| Name   | Description  |
|--------|--|
| ACTIVE | Load balancer is configured properly and ready to serve traffic to incoming requests via the configured virtual IPs. |

| Name           | Description   |
|----------------|---|
| BUILD          | Load balancer is being provisioned for the first time and configuration is being applied to bring the service online. The service will not yet be ready to serve incoming requests. |
| PENDING_UPDATE | Load balancer is online but configuration changes are being applied to update the service based on a previous request.  |
| PENDING_DELETE | Load balancer is online but configuration changes are being applied to begin deletion of the service based on a previous request.   |
| SUSPENDED      | Load balancer has been taken offline and disabled; contact Support.   |
| ERROR          | The system encountered an error when attempting to configure the load balancer; contact Support.  |
| DELETED        | Load balancers in DELETED status can be displayed for at least 90 days after deletion.  |

## 4.1.5. Remove Load Balancer

| Verb   | URI                                      | Description  |
|--------|--|--|
| DELETE | /loadbalancers/ <i>loadBalancerId</i>    | Remove a load balancer from the account.             |
| DELETE | /loadbalancers?id= <i>loadBalancerId</i> | Batch delete load balancers given specified id list. |

Normal Response Code(s): 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

The remove load balancer function removes the specified load balancer and its associated configuration from the account. Any and all configuration data is immediately purged and is not recoverable.

To batch delete load balancers, provide a query parameter list of load balancer ids. For example: /loadbalancers?id=*loadBalancerId*&id= *loadBalancerId* The current default limit is ten ids per request. Any and all configuration data is immediately purged and is not recoverable. By chance one of the items in the list cannot be removed due to its current status a 400:BadRequest is returned along with the ids of the ones the system identified as potential failures for this request.

This operation does not require a request body.

This operation does not return a response body.

## 4.2. Error Page

An error page is the html file that is shown to an end user who is attempting to access a load balancer node that is offline/unavailable.

### 4.2.1. Default Error Page

During provisioning, every load balancer is configured with a default error page that gets displayed when traffic is requested for an offline node.

### 4.2.2. Custom Error Page

A single custom error page may be added per account load balancer with an HTTP protocol. Page updates will override existing content. If a custom error page is deleted,

or the load balancer is changed to a non-HTTP protocol, the default error page will be restored.

### 4.2.3. Error Page Operations

| Verb   | URI  | Description   |
|--------|--|---|
| GET    | /loadbalancers/ <i>loadBalancerId</i> /errorpage | List error page configured for the specified load balancer. |
| PUT    | /loadbalancers/ <i>loadBalancerId</i> /errorpage | Set custom error page for the specified load balancer.      |
| DELETE | /loadbalancers/ <i>loadBalancerId</i> /errorpage | Delete custom error page for the specified load balancer.   |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

#### Example 4.19. List Error Page Response: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<errorpage xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <content>
    <html> DEFAULT ERROR PAGE
  </html>
</content>
</errorpage>
```

#### Example 4.20. List Error Page Response: JSON

```
{ "errorpage":
  { "content": "<html> DEFAULT ERROR PAGE</html>" }
}
```

#### Example 4.21. Add Error Page Request: XML

```
<errorpage xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <content>
    <![CDATA[
      <html> DEFAULT ERROR PAGE
    </html>
    ]]>
  </content>
</errorpage>
```

#### Example 4.22. Add Error Page Request: JSON

```
{ "errorpage":
  { "content": "\n<html>\n    DEFAULT ERROR PAGE\n</html>\n" }
}
```



```
}
```

### Example 4.23. Add Error Page Response: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<errorpage xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <content>
    <html> DEFAULT ERROR PAGE
  </html>
</content>
</errorpage>
```

### Example 4.24. Add Error Page Response: JSON

```
{ "errorpage":
  { "content": "<html> DEFAULT ERROR PAGE</html>" }
}
```

## 4.3. List Load Balancer Stats

| Verb | URI                                 | Description   | Representations |
|------|-------------------------------------|---|-----------------|
| GET  | /loadbalancers/loadBalancerId/stats | provides detailed stats output for a specific load balancer | JSON, XML       |

Normal Response Code(s): 200

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation provides detailed stats output, including the following information, for a specific load balancer configured and associated with the user's account:

- **connectTimeOut** – Connections closed by this load balancer because the 'connect\_timeout' interval was exceeded.
- **connectError** – Number of transaction or protocol errors in this load balancer.
- **connectFailure** – Number of connection failures in this load balancer.
- **dataTimedOut** – Connections closed by this load balancer because the 'timeout' interval was exceeded.
- **keepAliveTimedOut** – Connections closed by this load balancer because the 'keepalive\_timeout' interval was exceeded.
- **maxConn** – Maximum number of simultaneous TCP connections this load balancer has processed at any one time.

This operation does not require a request body.

### Example 4.25. List Load Balancer stats Response: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<stats xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
      connectTimeout="10" connectError="20" connectFailure="1" dataTimedOut=
"30" keepAliveTimedOut="40" maxConn="50"/>
```

### Example 4.26. List Load Balancers Stats Response: JSON

```
{
  "connectTimeout":10,
  "connectError":20,
  "connectFailure":30,
  "dataTimedOut":40,
  "keepAliveTimedOut":50,
  "maxConn":60
}
```

## 4.4. Nodes

The nodes defined by the load balancer are responsible for servicing the requests received through the load balancer's virtual IP. By default, the load balancer employs a basic health check that ensures the node is listening on its defined port. The node is checked at the time of addition and at regular intervals as defined by the load balancer health check configuration. If a back-end node is not listening on its port or does not meet the conditions of the defined active health check for the load balancer, then the load balancer will not forward connections and its status will be listed as `OFFLINE`. Only nodes that are in an `ONLINE` status will receive and be able to service traffic from the load balancer.

All nodes have an associated status that indicates whether the node is `ONLINE`, `OFFLINE`, or `DRAINING`. Only nodes that are in `ONLINE` status will receive and be able to service traffic from the load balancer. The `OFFLINE` status represents a node that cannot accept or service traffic. A node in `DRAINING` status represents a node that stops the traffic manager from sending any additional new connections to the node, but honors established sessions. If the traffic manager receives a request and session persistence requires that the node is used, the traffic manager will use it. The status is determined by the passive or active health monitors.

If the `WEIGHTED_ROUND_ROBIN` load balancer algorithm mode is selected, then the caller should assign the relevant weights to the node as part of the `weight` attribute of the node element. When the algorithm of the load balancer is changed to `WEIGHTED_ROUND_ROBIN` and the nodes do not already have an assigned weight, the service will automatically set the weight to "1" for all nodes.

### 4.4.1. List Nodes

| Verb | URI  | Description                                    |
|------|--|--|
| GET  | /loadbalancers/ <i>loadBalancerId</i> /nodes | List node(s) configured for the load balancer. |

| Verb | URI   | Description                      |
|------|---|----------------------------------|
| GET  | /loadbalancers/ <i>loadBalancerId</i> /nodes/ <i>nodeId</i> | List details of a specific node. |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation does not require a request body.



### Note

The weight attributes are only displayed for load balancers that use the WEIGHTED\_LEAST\_CONNECTIONS or WEIGHTED\_ROUND\_ROBIN algorithms.

### Example 4.27. List Node Response: XML

```
<node
  id="410"
  address="10.1.1.1"
  port="80"
  condition="ENABLED"
  status="ONLINE"
  weight="12"/>
```

### Example 4.28. List Node Response: JSON

```
{ "node": {
  "id": 410,
  "address": "10.1.1.1",
  "port": 80,
  "condition": "ENABLED",
  "status": "ONLINE",
  "weight": 12
}
```

### Example 4.29. List Atom Node Response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <link rel="next"
    href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/1234/
loadbalancers/141/nodes/18536.atom?page=2"/>
  <title type="text">Node Feed</title>
  <id>1234-loadbalancers-141-nodes-18536</id>
  <author>
    <name>Rackspace Cloud</name>
  </author>
  <entry>
    <title type="text">Node Status Updated</title>
```

```
<summary type="text">Node '18536' status changed to 'OFFLINE' for load
balancer '141'</summary>
<author>
  <name>Rackspace Cloud</name>
</author>
<link href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/
1234/loadbalancers/141/nodes/18536"/>
<id>1234-loadbalancers-141-nodes-18536-201195217490</id>
<category term="UPDATE"/>
<updated>2011-04-05T21:07:49.000Z</updated>
</entry>
</feed>
```

### Example 4.30. List Nodes Response: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<nodes xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <node
    id="410"
    address="10.1.1.1"
    port="80"
    condition="ENABLED"
    status="ONLINE"
    weight="3"/>
  <node
    id="411"
    address="10.1.1.2"
    port="80"
    condition="ENABLED"
    status="ONLINE"
    weight="8"/>
  <node
    id="412"
    address="10.1.1.3"
    port="80"
    condition="DISABLED"
    status="ONLINE"
    weight="12"/>
</nodes>
```

### Example 4.31. List Nodes Response: JSON

```
{
  "nodes": [
    {
      "id": "410",
      "address": "10.1.1.1",
      "port": 80,
      "condition": "ENABLED",
      "status": "ONLINE",
      "weight": 3
    },
    {
      "id": "411",
      "address": "10.1.1.2",
      "port": 80,
```

```
        "condition": "ENABLED",
        "status": "ONLINE",
        "weight": 8
      },
      {
        "id": "412",
        "address": "10.1.1.3",
        "port": 80,
        "condition": "DISABLED",
        "status": "ONLINE",
        "weight": 12
      }
    ]
  }
```

### Example 4.32. List Atom Nodes Response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <link rel="next"
        href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/1234/
loadbalancers/141/nodes.atom?page=2"/>
  <title type="text">Nodes Feed</title>
  <id>1234-loadbalancers-141-nodes</id>
  <author>
    <name>Rackspace Cloud</name>
  </author>
  <entry>
    <title type="text">Node Status Updated</title>
    <summary type="text">Node '650' status changed to 'OFFLINE' for load
balancer '141'</summary>
    <author>
      <name>Rackspace Cloud</name>
    </author>
    <link href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/
1234/loadbalancers/141/nodes/650"/>
    <id>1234-loadbalancers-141-nodes-650-2011952125350</id>
    <category term="UPDATE"/>
    <updated>2011-04-05T21:25:35.000Z</updated>
  </entry>
</feed>
```

## 4.4.2. Add Nodes

| Verb | URI  | Description                          |
|------|--|--------------------------------------|
| POST | /loadbalancers/ <i>loadBalancerId</i> /nodes | Add a new node to the load balancer. |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

When a node is added, it is assigned a unique identifier that can be used for management operations such as changing the condition or removing it. Every load balancer is dual-

homed on both the public Internet and ServiceNet; as a result, nodes can either be internal ServiceNet addresses or addresses on the public Internet.

### Example 4.33. Add Nodes Request: XML

```
<nodes xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <node address="10.1.1.1" port="80" condition="ENABLED" />
  <node address="10.2.2.1" port="80" condition="ENABLED" />
  <node address="10.2.2.2" port="88" condition="ENABLED" weight="10" />
</nodes>
```

### Example 4.34. Add Nodes Request: JSON

```
{ "nodes": [
  {
    "address": "10.1.1.1",
    "port": 80,
    "condition": "ENABLED"
  },
  {
    "address": "10.2.2.1",
    "port": 80,
    "condition": "ENABLED"
  },
  {
    "address": "10.2.2.2",
    "port": 88,
    "condition": "ENABLED",
    "weight": 10
  }
]
}
```

### Example 4.35. Add Nodes Response: XML

```
<nodes xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <node
    address="10.2.2.3"
    id="185"
    port="80"
    condition="ENABLED"
    status="ONLINE"
    weight="1" />
  <node
    address="10.2.2.4"
    id="186"
    port="80"
    condition="ENABLED"
    status="ONLINE"
    weight="1" />
</nodes>
```

### Example 4.36. Add Nodes Response: JSON

```
{ "nodes": [
  {
    "address": "10.2.2.3",
    "id": 185,
    "port": 80,
    "status": "ONLINE",
    "condition": "ENABLED",
    "weight": 1
  },
  {
    "address": "10.2.2.4",
    "id": 186,
    "port": 80,
    "status": "ONLINE",
    "condition": "ENABLED",
    "weight": 1
  }
]
```

## 4.4.3. Modify Nodes

| Verb       | URI   | Description  |
|------------|---|--|
| <b>PUT</b> | /loadbalancers/ <i>loadBalancerId</i> /nodes/ <i>nodeId</i> | Modify the configuration of a node on the load balancer. |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation does not require a request body.



### Note

The node's IP and port are immutable attributes and cannot be modified with a **PUT** request. Supplying an unsupported attribute will result in a 400 (badRequest) fault. A load balancer supports a maximum of 25 nodes; the maximum weight of a node is 100.

Every node in the load balancer has an associated condition which determines its role within the load balancer.

**Table 4.3. Load Balancer Node Conditions**

| Name     | Description   |
|----------|---|
| ENABLED  | Node is permitted to accept new connections.  |
| DISABLED | Node is not permitted to accept any new connections regardless of session persistence configuration. Existing connections are forcibly terminated.              |
| DRAINING | Node is allowed to service existing established connections and connections that are being directed to it as a result of the session persistence configuration. |

### Example 4.37. Update Node Condition Request: XML

```
<node condition="ENABLED" weight="12"/>
```

### Example 4.38. Update Node Condition Request: JSON

```
{ "node": {  
    "condition": "ENABLED",  
    "weight": 59  
  }  
}
```

## 4.4.4. Remove Nodes

| Verb   | URI   | Description                                     |
|--------|---|---|
| DELETE | /loadbalancers/ <i>loadBalancerId</i> /nodes/ <i>nodeId</i>                       | Remove a node from the load balancer.           |
| DELETE | /loadbalancers/ <i>loadBalancerId</i> /nodes?id= <i>nodeId</i> &id= <i>nodeId</i> | Batch delete the nodes given specified id list. |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation does not require a request body. To batch delete nodes, provide a query parameter list of node ids. For example: /nodes?id=*nodeId*&id=*nodeId* The current default limit is ten ids per request. Any and all configuration data is immediately purged and is not recoverable. By chance one of the items in the list cannot be removed due to its current status a 400:BadRequest is returned along with the ids of the ones the system identified as potential failures for this request

## 4.5. Virtual IPs

A virtual IP (VIP) makes a load balancer accessible by clients. The load balancing service supports either a public VIP, routable on the public Internet, or a ServiceNet address, routable only within the region in which the load balancer resides.

**Table 4.4. Virtual IP Types**

| Name       | Description   |
|------------|---|
| PUBLIC     | An address that is routable on the public Internet. |
| SERVICENET | An address that is routable only on ServiceNet.     |

### 4.5.1. List Virtual IPs

| Verb | URI   | Description   |
|------|---|---|
| GET  | /loadbalancers/ <i>loadBalancerId</i> /virtualips | List all virtual IPs associated with a load balancer. |

Normal Response Code(s): 200, 202



Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This request does not require a request body.

### Example 4.39. List Virtual IPs Response: XML

```
<virtualIps xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <virtualIp
    id="1000"
    address="206.10.10.210"
    type="PUBLIC"/>
</virtualIps>
```

### Example 4.40. List Virtual IPs Response: JSON

```
{ "virtualIps": [
  {
    "id": 1000,
    "address": "206.10.10.210",
    "type": "PUBLIC"
  }
]
```

### Example 4.41. List Atom Virtual IPs Response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <link rel="next"
    href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/1234/
loadbalancers/141/virtualips.atom?page=2"/>
  <title type="text">Virtual Ips Feed</title>
  <id>1234-loadbalancers-141-virtualips</id>
  <author>
    <name>Rackspace Cloud</name>
  </author>
  <entry>
    <title type="text">Virtual Ip Successfully Added</title>
    <summary type="text">Virtual ip successfully added</summary>
    <link href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/
1234/loadbalancers/141"/>
    <id>1234-loadbalancers-141-20111011631330</id>
    <category term="CREATE"/>
    <updated>2011-04-11T16:31:33.000Z</updated>
  </entry>
</feed>
```

## 4.5.2. Add Virtual IP Version 6

| Verb | URI                                      | Description              |
|------|--|--------------------------|
| POST | /loadbalancers/loadBalancerId/virtualips | Add virtual IP version 6 |

Normal Response Code(s): 200, 202

Error Response code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

#### Example 4.42. Add Virtual IP Version 6 Request: XML

```
<virtualIp xmlns="http://docs.openstack.org/loadbalancers/api/v1.0" type="PUBLIC" ipVersion="IPV6" />
```

#### Example 4.43. Add Virtual IP Version 6 Request: JSON

```
{
  "type": "PUBLIC",
  "ipVersion": "IPV6"
}
```

#### Example 4.44. Add Virtual IP Version 6 Response: XML

```
<virtualIp xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  id="9000133"
  address="fd24:f480:ce44:91bc:1af2:15ff:0000:0001"
  ipVersion="IPV6"
  type="PUBLIC" />
```

#### Example 4.45. Add Virtual IP Version 6 Response: JSON

```
{
  "address": "fd24:f480:ce44:91bc:1af2:15ff:0000:0002",
  "id": 9000134,
  "type": "PUBLIC",
  "ipVersion": "IPV6"
}
```

### 4.5.3. Remove Virtual IP

| Verb   | URI   | Description  |
|--------|---|--|
| DELETE | /loadbalancers/loadBalancerId/virtualips/ id                            | Remove a virtual IP.                                   |
| DELETE | /loadbalancers/loadBalancerId/virtualips?id=virtualipId &id=virtualipId | Batch delete the virtual IPs specified in the id list. |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This request does not require a request body.



## Note

All load balancers must have at least one virtual IP associated with them at all times. Attempting to delete the last virtual IP will result in a 400 (badRequest) fault. To batch delete virtual ips, provide a query parameter list of virtual ip ids. For example: `/virtualips?id=virtualIpId&id=virtualIpId`. The current default limit is ten ids per request. Any and all configuration data is immediately purged and is not recoverable. If one or more of the items in the list cannot be removed due to its current status, a 400 (badRequest) is returned along with the ids of the ones the system identified as potential failures for this request.

## 4.6. Usage Reports

### 4.6.1. List Usage

| Name | URI   | Description               |
|------|---|---------------------------|
| GET  | <code>/loadbalancers/usage?<br/>startTime=2010-12-15&amp;endTime=2011-5-1</code>                | List account level usage. |
| GET  | <code>/loadbalancers/loadBalancerId/usage?<br/>startTime=2010-12-15&amp;endTime=2011-5-1</code> | List historical usage.    |
| GET  | <code>/loadbalancers/loadBalancerId/usage/current</code>  | List current usage.       |

Normal Response Code(s): 200

Error Response Code(s): loadBalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation does not require a request body.

The load balancer usage reports provide a view of all transfer activity, average number of connections, and number of virtual IPs associated with the load balancing service. Current usage represents all usage recorded within the preceding 24 hours. Values for both `incomingTransfer` and `outgoingTransfer` are expressed in bytes transferred. The optional `startTime` and `endTime` parameters can be used to filter all usage. If the `startTime` parameter is supplied but the `endTime` parameter is not, then all usage beginning with the `startTime` will be provided. Likewise, if the `endTime` parameter is supplied but the `startTime` parameter is not, then all usage will be returned up to the `endTime` specified.



## Note

Historical usage data is available for up to 90 days of service activity.

### Example 4.46. Report Load Balancer Usage Response: XML

```
<loadBalancerUsage xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <loadBalancerUsageRecord
    id="394"
    averageNumConnections="0.0"
    incomingTransfer="0"
```

```
        outgoingTransfer="0"  
        numVips="1"  
        numPolls="32"  
        startTime="2010-12-21T12:32:07-06:00"  
        endTime="2010-12-21T16:23:54-06:00"  
        vipType="PUBLIC"/>  
<loadBalancerUsageRecord  
    id="473"  
    averageNumConnections="0.0"  
    incomingTransfer="0"  
    outgoingTransfer="0"  
    numVips="2"  
    numPolls="5"  
    startTime="2010-12-21T12:32:07-06:00"  
    endTime="2010-12-21T12:36:30-06:00"  
    vipType="PUBLIC"/>  
</loadBalancerUsage>
```

### Example 4.47. Report Load Balancer Usage Response: JSON

```
{  
  "loadBalancerUsageRecords": [  
    {  
      "id": 394,  
      "averageNumConnections": 0.0,  
      "incomingTransfer": 0,  
      "outgoingTransfer": 0,  
      "numVips": 1,  
      "numPolls": 32,  
      "startTime": "2010-12-21T12:32:07-06:00",  
      "endTime": "2010-12-21T16:23:54-06:00" ,  
      "vipType": "PUBLIC",  
    },  
    {  
      "id": 473,  
      "averageNumConnections": 0.0,  
      "incomingTransfer": 0,  
      "outgoingTransfer": 0,  
      "numVips": 2,  
      "numPolls": 5,  
      "startTime": "2010-12-21T12:32:07-06:00",  
      "endTime": "2010-12-21T12:36:30-06:00" ,  
      "vipType": "PUBLIC"  
    }  
  ]  
}
```

## 4.7. Access Lists

The access list management feature allows fine-grained network access controls to be applied to the load balancer's virtual IP address. A single IP address, multiple IP addresses, or entire network subnets can be added as a `networkItem`. Items that are configured with the `ALLOW` type will always take precedence over items with the `DENY` type. To reject traffic from all items except for those with the `ALLOW` type, add a `networkItem` with an address of `"0.0.0.0/0"` and a `DENY` type.

## 4.7.1. Manage Access Lists

| Name          | URI  | Description   |
|---------------|--|---|
| <b>GET</b>    | /loadbalancers/ <i>loadBalancerId</i> /accesslist  | List the access list.   |
| <b>POST</b>   | /loadbalancers/ <i>loadBalancerId</i> /accesslist  | Create a new access list or append to an existing access list.      |
| <b>DELETE</b> | /loadbalancers/ <i>loadBalancerId</i> /accesslist/ <i>networkItemId</i>                              | Remove a network item from the access list.                         |
| <b>DELETE</b> | /loadbalancers/ <i>loadBalancerId</i> /accesslist  | Remove the entire access list.                                      |
| <b>DELETE</b> | /loadbalancers/ <i>loadBalancerId</i> /accesslist?id= <i>networkItemId</i> &id= <i>networkItemId</i> | Batch delete the access list network items given specified id list. |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation does not require a request body.

When issuing a **POST** to add to an access list, one or more network items are required. If a populated access list already exists for the load balancer, it will be appended to with subsequent **POST** requests. One access list may include up to 100 network items. A single address or subnet definition is considered unique and cannot be duplicated between items in an access list. There are two **DELETE** operations for the access list. One allows for deletion of multiple network items in an access list, or the entire accesslist, and the other for deletion of a specific network item in the access list.

### Example 4.48. List Access List Response: XML

```
<accessList xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <networkItem
    id="1000"
    address="206.160.165.40"
    type="ALLOW" />
  <networkItem
    id="1001"
    address="206.160.165.0/24"
    type="DENY" />
</accessList>
```

### Example 4.49. List Access List Response: JSON

```
{
  "accessList": [
    {
      "address": "206.160.163.21",
      "id": 23,
      "type": "DENY"
    },
    {
      "address": "206.160.165.11",
      "id": 24,
      "type": "DENY"
    }
  ]
}
```

```

    },
    {
      "address": "206.160.163.21",
      "id": 25,
      "type": "DENY"
    },
    {
      "address": "206.160.165.11",
      "id": 26,
      "type": "DENY"
    },
    {
      "address": "206.160.123.11",
      "id": 27,
      "type": "DENY"
    },
    {
      "address": "206.160.122.21",
      "id": 28,
      "type": "DENY"
    },
    {
      "address": "206.140.123.11",
      "id": 29,
      "type": "DENY"
    },
    {
      "address": "206.140.122.21",
      "id": 30,
      "type": "DENY"
    }
  ]
}

```

### Example 4.50. List Atom Access List Response: XML

```

<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <link rel="next"
    href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/1234/
loadbalancers/141/accesslist.atom?page=2"/>
  <title type="text">Access List Feed</title>
  <id>1234-loadbalancers-141-accesslist</id>
  <author>
    <name>Rackspace Cloud</name>
  </author>
  <entry>
    <title type="text">Access List Updated</title>
    <summary
      type="text">Access list successfully updated with the
following network item: id: '2155', address: '206.160.163.210', type: 'DENY'
    </summary>
    <author>
      <name>tvardema</name>
    </author>
    <link href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/
1234/loadbalancers/141/accesslist/" />
    <id>1234-loadbalancers-141-accesslist-2011971658310</id>
  </entry>
</feed>

```

```
<category term="UPDATE"/>
<updated>2011-04-07T16:58:31.000Z</updated>
</entry>
<entry>
  <title type="text">Access List Updated</title>
  <summary
    type="text">Access list successfully updated with the
following network item: id: '2156', address: '206.160.165.110', type: 'DENY'
  </summary>
  <author>
    <name>tvardema</name>
  </author>
  <link href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/
1234/loadbalancers/141/accesslist/" />
  <id>1234-loadbalancers-141-accesslist-2011971658310</id>
  <category term="UPDATE"/>
  <updated>2011-04-07T16:58:31.000Z</updated>
</entry>
</feed>
```

#### Example 4.51. Update Access List Attributes Request: XML

```
<accessList xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <networkItem
    address="206.160.165.1"
    type="ALLOW" />
  <networkItem
    address="206.160.165.2"
    type="DENY" />
</accessList>
```

#### Example 4.52. Update Access List Attributes Request: JSON

```
{
  "accessList": [
    {
      "address": "206.160.163.21",
      "type": "DENY"
    },
    {
      "address": "206.160.165.11",
      "type": "DENY"
    }
  ]
}
```

## 4.8. Monitors

The load balancing service includes a health monitoring operation which periodically checks your back-end nodes to ensure they are responding correctly. If a node is not responding, it is removed from rotation until the health monitor determines that the node is functional. In addition to being performed periodically, the health check also is performed against every node that is added to ensure that the node is operating properly before allowing it

to service traffic. Only one health monitor is allowed to be enabled on a load balancer at a time.

Every health monitor has a `type` attribute to signify what kind of monitor it is.

**Table 4.5. Health Monitor Types**

| Name    | Description                          |
|---------|--------------------------------------|
| CONNECT | Health monitor is a connect monitor. |
| HTTP    | Health monitor is an HTTP monitor.   |
| HTTPS   | Health monitor is an HTTPS monitor.  |

## 4.8.1. Monitor Health

| Verb   | URI  | Description   |
|--------|--|---|
| GET    | /loadbalancers/ <i>loadBalancerId</i> /healthmonitor | Retrieve the health monitor configuration, if one exists. |
| PUT    | /loadbalancers/ <i>loadBalancerId</i> /healthmonitor | Update the settings for a health monitor.                 |
| DELETE | /loadbalancers/ <i>loadBalancerId</i> /healthmonitor | Remove the health monitor.                                |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

### 4.8.1.1. Monitor Connections

The monitor connects to each node on its defined port to ensure that the service is listening properly. The connect monitor is the most basic type of health check and does no post-processing or protocol specific health checks. It includes several configurable properties:

- `delay`: The minimum number of seconds to wait before executing the health monitor. (Must be a number between 1 and 3600)
- `timeout`: Maximum number of seconds to wait for a connection to be established before timing out. (Must be a number between 1 and 300)
- `attemptsBeforeDeactivation`: Number of permissible monitor failures before removing a node from rotation. (Must be a number between 1 and 10)

#### Example 4.53. Monitor Connections Request: XML

```
<healthMonitor xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  type="CONNECT"
  delay="10"
  timeout="10"
  attemptsBeforeDeactivation="3" />
```

#### Example 4.54. Monitor Connections Request: JSON

```
{
  "type": "CONNECT",
  "delay": 10,
```



```
"timeout": 10,  
  "attemptsBeforeDeactivation": 3  
}
```

#### Example 4.55. Monitor Connections Response: XML

```
<healthMonitor xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"  
  type="CONNECT"  
  delay="10"  
  timeout="10"  
  attemptsBeforeDeactivation="3" />
```

#### Example 4.56. Monitor Connections Response: JSON

```
{ "healthMonitor": {  
    "type": "CONNECT",  
    "delay": 10,  
    "timeout": 10,  
    "attemptsBeforeDeactivation": 3  
  }  
}
```

#### Example 4.57. List Atom Health Monitor Response: XML

```
<?xml version='1.0' encoding='UTF-8'?>  
<feed xmlns="http://www.w3.org/2005/Atom">  
  <link rel="next"  
    href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/1234/  
loadbalancers/141/healthmonitor.atom?page=2"/>  
  <title type="text">Health Monitor Feed</title>  
  <id>1234-loadbalancers-141-healthmonitor</id>  
  <author>  
    <name>Rackspace Cloud</name>  
  </author>  
  <entry>  
    <title type="text">Health Monitor Successfully Updated</title>  
    <summary  
      type="text">Health monitor successfully updated with type:  
'HTTP', delay: '10', timeout: '10', attemptsBeforeDeactivation: '3', path: '/  
, statusRegex: '^[234][0-9][0-9]$', bodyRegex: '^[234][0-9][0-9]$\'  
    </summary>  
    <author>  
      <name>tvardema</name>  
    </author>  
    <link href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/  
1234/loadbalancers/141/healthmonitor/" />  
    <id>1234-loadbalancers-141-healthmonitor-201142022120</id>  
    <category term="UPDATE" />  
    <updated>2011-02-11T00:22:12.000Z</updated>  
  </entry>  
</feed>
```

### 4.8.1.2. Monitor HTTP and HTTPS

The HTTP and HTTPS monitor is more intelligent than the connect monitor. It is capable of processing an HTTP or HTTPS response to determine the condition of a node. It supports the same basic properties as the connect monitor and includes additional attributes that are used to evaluate the HTTP response. Available attributes are:

- **delay**: The minimum number of seconds to wait before executing the health monitor. (Must be a number between 1 and 3600)
- **timeout**: The maximum number of seconds to wait for a connection to be established before timing out. (Must be a number between 1 and 300)
- **attemptsBeforeDeactivation**: The number of permissible monitor failures before removing a node from rotation. (Must be a number between 1 and 10)
- **path**: The HTTP path that will be used in the sample request
- **statusRegex**: A regular expression that will be used to evaluate the HTTP status code returned in the response.
- **bodyRegex**: A regular expression that will be used to evaluate the contents of the body of the response.

#### Example 4.58. Monitor HTTP Response: XML

```
<healthMonitor xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  type="HTTP"
  delay="10"
  timeout="10"
  attemptsBeforeDeactivation="3"
  path="/"
  statusRegex="^[234][0-9][0-9]$"
  bodyRegex="^[234][0-9][0-9]$" />
```

#### Example 4.59. Monitor HTTP Response: JSON

```
{ "healthMonitor": {
  "type": "HTTP",
  "delay": 10,
  "timeout": 10,
  "attemptsBeforeDeactivation": 3,
  "path": "/",
  "statusRegex": "^[234][0-9][0-9]$",
  "bodyRegex": "^[234][0-9][0-9]$"
}
```

### Example 4.60. Monitor HTTPS Response: XML

```
<healthMonitor xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  type="HTTPS"
  delay="10"
  timeout="10"
  attemptsBeforeDeactivation="3"
  path="/"
  statusRegex="^[234][0-9][0-9]$"
  bodyRegex="" />
```

### Example 4.61. Monitor HTTPS Response: JSON

```
{ "healthMonitor": {
  "type": "HTTPS",
  "delay": 10,
  "timeout": 10,
  "attemptsBeforeDeactivation": 3,
  "path": "/",
  "statusRegex": "^[234][0-9][0-9]$",
  "bodyRegex": "^[234][0-9][0-9]$"
}
```

## 4.9. Sessions

### 4.9.1. Manage Session Persistence

Session persistence is a feature of the load balancing service which forces multiple requests from clients to be directed to the same node. This is common with many web applications that do not inherently share application state between back-end servers.

**Table 4.6. Session Persistence Modes**

| Name        | Description   |
|-------------|---|
| HTTP_COOKIE | A session persistence mechanism that inserts an HTTP cookie and is used to determine the destination back-end node. This is supported for HTTP load balancing only. |

| Verb   | URI  | Description                             |
|--------|--|---|
| GET    | /loadbalancers/loadBalancerId/sessionpersistence | List session persistence configuration. |
| PUT    | /loadbalancers/loadBalancerId/sessionpersistence | Enable session persistence.             |
| DELETE | /loadbalancers/loadBalancerId/sessionpersistence | Disable session persistence.            |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

### Example 4.62. List Session Persistence Configuration Response: XML

```
<sessionPersistence xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  persistenceType="HTTP_COOKIE"/>
```

### Example 4.63. List Session Persistence Configuration Response: JSON

```
{
  "sessionPersistence": {
    "persistenceType": "HTTP_COOKIE"
  }
}
```

### Example 4.64. List Atom Session Persistence Response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <link rel="next"
    href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/1234/
loadbalancers/141/sessionpersistence.atom?page=2"/>
  <title type="text">Session Persistence Feed</title>
  <id>1234-loadbalancers-141-sessionpersistence</id>
  <author>
    <name>Rackspace Cloud</name>
  </author>
  <entry>
    <title type="text">Session Persistence Successfully Updated</title>
    <summary type="text">Session persistence successfully updated to
'HTTP_COOKIE'</summary>
    <author>
      <name>tvardema</name>
    </author>
    <link href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/
1234/loadbalancers/141/sessionpersistence"/>
    <id>1234-loadbalancers-141-sessionpersistence-201142028460</id>
    <category term="UPDATE"/>
    <updated>2011-02-11T00:28:46.000Z</updated>
  </entry>
</feed>
```

### Example 4.65. Set Session Persistence Type Request: XML

```
<sessionPersistence xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  persistenceType="HTTP_COOKIE"/>
```

**Example 4.66. Set Session Persistence Type Request: JSON**

```
{
  "sessionPersistence": {
    "persistenceType": "HTTP_COOKIE"
  }
}
```

## 4.10. Connections

### 4.10.1. Log Connections

| Verb | URI   | Description                                       |
|------|---|---|
| GET  | /loadbalancers/loadBalancerId/connectionlogging | View current configuration of connection logging. |
| PUT  | /loadbalancers/loadBalancerId/connectionlogging | Enable or disable connection logging.             |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation allows the user to view the current connection logging configuration, enable connection logging, or disable connection logging.

This operation does not require a request body.

**Example 4.67. List Connection Logging Configuration Response: XML**

```
<connectionLogging xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  enabled="true" />
```

**Example 4.68. List Connection Logging Configuration Response: JSON**

```
{
  "connectionLogging": {
    "enabled": true
  }
}
```

**Example 4.69. Enable Connection Logging Request: XML**

```
<connectionLogging xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  enabled="true" />
```

### Example 4.70. Enable Connection Logging Request: JSON

```
{
  "connectionLogging": {
    "enabled": true
  }
}
```

## 4.10.2. Throttle Connections

| Verb   | URI  | Description                                  |
|--------|--|--|
| GET    | /loadbalancers/loadBalancerId/connectionthrottle | List connection throttling configuration.    |
| PUT    | /loadbalancers/loadBalancerId/connectionthrottle | Update throttling configuration.             |
| DELETE | /loadbalancers/loadBalancerId/connectionthrottle | Remove connection throttling configurations. |

Normal Response Code(s): 200, 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation does not require a request body.

The connection throttling feature imposes limits on the number of connections per IP address to help mitigate malicious or abusive traffic to your applications. The following properties can be configured based on the traffic patterns for your sites.

- **minConnections**: Allow at least this number of connections per IP address before applying throttling restrictions. Setting a value of 0 allows unlimited simultaneous connections; otherwise, set a value between 1 and 1000.
- **maxConnections**: Maximum number of connection to allow for a single IP address. Setting a value of 0 will allow unlimited simultaneous connections, otherwise set a value between 1 and 100000.
- **maxConnectionRate**: Maximum number of connections allowed from a single IP address in the **definedrateInterval**. Setting a value of 0 allows an unlimited connection rate; otherwise, set a value between 1 and 100000.
- **rateInterval**: Frequency (in seconds) at which the **maxConnectionRate** is assessed. For example, a **maxConnectionRate** of 30 with a **rateInterval** of 60 would allow a maximum of 30 connections per minute for a single IP address. This value must be specified between 1 and 3600.

### Example 4.71. List Connection Throttling Configuration Response: XML

```
<connectionThrottle xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  minConnections="10"
  maxConnections="100"
  maxConnectionRate="50"
  rateInterval="60" />
```

**Example 4.72. List Connection Throttling Configuration Response: JSON**

```
{ "connectionThrottle": {  
    "maxConnections": 100,  
    "minConnections": 10,  
    "maxConnectionRate": 50,  
    "rateInterval": 60  
  }  
}
```

**Example 4.73. Update Connection Throttling Configuration Request: XML**

```
<connectionThrottle xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"  
  minConnections="10"  
  maxConnections="100"  
  maxConnectionRate="50"  
  rateInterval="60" />
```

**Example 4.74. Update Connection Throttling Configuration Request: JSON**

```
{  
  "maxConnections": 10,  
  "minConnections": 100,  
  "maxConnectionRate": 50,  
  "rateInterval": 60  
}
```

**Example 4.75. List Atom Connection Throttling Response: XML**

```
<?xml version='1.0' encoding='UTF-8'?>  
<feed xmlns="http://www.w3.org/2005/Atom">  
  <link rel="next"  
    href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/1234/  
loadbalancers/141/connectionthrottle.atom?page=2"/>  
  <title type="text">Connection Throttle Feed</title>  
  <id>1234-loadbalancers-141-connectionthrottle</id>  
  <author>  
    <name>Rackspace Cloud</name>  
  </author>  
  <entry>  
    <title type="text">Error Updating Connection Throttle</title>  
    <summary type="text">Could not update the connection throttle at this  
time</summary>  
    <link href="https://ord.loadbalancers.api.rackspacecloud.com/v1.0/  
1234/loadbalancers/141/connectionthrottle/" />  
    <id>1234-loadbalancers-141-connectionthrottle-2011881846570</id>  
    <category term="UPDATE" />  
    <updated>2011-03-29T18:46:57.000Z</updated>  
  </entry>  
</feed>
```

## 4.11. Protocols

### 4.11.1. List Load Balancing Protocols

| Verb | URI                      | Description                                  |
|------|--------------------------|--|
| GET  | /loadbalancers/protocols | List all supported load balancing protocols. |

Normal Response Code(s): 200

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation does not require a request body.

All load balancers must define the protocol of the service which is being load balanced. The protocol selection should be based on the protocol of the back-end nodes. When configuring a load balancer, the default port for the given protocol will be selected unless otherwise specified.

#### Example 4.76. List Load Balancing Protocols Response: XML

```
<protocols xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <protocol name="FTP" port="21" />
  <protocol name="HTTP" port="80" />
  <protocol name="IMAPv4" port="143" />
  <protocol name="POP3" port="110" />
  <protocol name="LDAP" port="389" />
  <protocol name="LDAPS" port="636" />
  <protocol name="HTTPS" port="443" />
  <protocol name="IMAPS" port="993" />
  <protocol name="POP3S" port="995" />
  <protocol name="SMTP" port="25" />
  <protocol name="TCP" port="0" />
</protocols>
```

#### Example 4.77. List Load Balancing Protocols Response: JSON

```
{ "protocols": [
  {
    "name": "HTTP",
    "port": 80
  },
  {
    "name": "FTP",
    "port": 21
  },
  {
    "name": "IMAPv4",
    "port": 143
  },
  ...
]
```



```

    {
      "name": "POP3",
      "port": 110
    },
    {
      "name": "SMTP",
      "port": 25
    },
    {
      "name": "LDAP",
      "port": 389
    },
    {
      "name": "HTTPS",
      "port": 443
    },
    {
      "name": "IMAPS",
      "port": 993
    },
    {
      "name": "POP3S",
      "port": 995
    },
    {
      "name": "LDAPS",
      "port": 636
    },
    {
      "name": "TCP",
      "port": 0
    }
  ]
}

```

## 4.12. Algorithms

All load balancers utilize an algorithm that defines how traffic should be directed between back-end nodes. The default algorithm for newly created load balancers is `RANDOM`, which can be overridden at creation time or changed after the load balancer has been initially provisioned. The algorithm name is to be constant within a major revision of the load balancing API, though new algorithms may be created with a unique algorithm name within a given major revision of the service API.

**Table 4.7. Load Balancing Algorithms**

| Name                       | Description   |
|----------------------------|---|
| LEAST_CONNECTIONS          | The node with the lowest number of connections will receive requests.   |
| RANDOM                     | Back-end servers are selected at random.  |
| ROUND_ROBIN                | Connections are routed to each of the back-end servers in turn.   |
| WEIGHTED_LEAST_CONNECTIONS | Each request will be assigned to a node based on the number of concurrent connections to the node and its weight.   |
| WEIGHTED_ROUND_ROBIN       | A round robin algorithm, but with different proportions of traffic being directed to the back-end nodes. Weights must be defined as part of the load balancer's node configuration. |

## 4.12.1. List Load Balancing Algorithms

| Verb | URI                       | Description                                   |
|------|---------------------------|---|
| GET  | /loadbalancers/algorithms | List all supported load balancing algorithms. |

Normal Response Code(s): 200

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

This operation does not require a request body.

### Example 4.78. List Load Balancing Algorithms Response: XML

```
<algorithms xmlns="http://docs.openstack.org/loadbalancers/api/v1.0">
  <algorithm name="LEAST_CONNECTIONS" />
  <algorithm name="RANDOM" />
  <algorithm name="ROUND_ROBIN" />
  <algorithm name="WEIGHTED_LEAST_CONNECTIONS" />
  <algorithm name="WEIGHTED_ROUND_ROBIN" />
</algorithms>
```

### Example 4.79. List Load Balancing Algorithms Response: JSON

```
{ "algorithms": [
  {
    "name": "LEAST_CONNECTIONS"
  },
  {
    "name": "RANDOM"
  },
  {
    "name": "ROUND_ROBIN"
  },
  {
    "name": "WEIGHTED_LEAST_CONNECTIONS"
  },
  {
    "name": "WEIGHTED_ROUND_ROBIN"
  }
]
```

## 4.13. SSL Termination



### Note

Note to customers who have Personally Identifiable Information (PII) in their data. You should *not* be using the SSL Termination feature. Examples of PII are

HIPPA protected health information, Gramm-Leach-Bliley financial information, credit card information, and any personal data that when disclosed could result in the theft of the person's identity.

| Verb   | URI   | Description  |
|--------|---|--|
| GET    | /loadbalancers/ <i>loadBalancerId</i> /ssltermination | View the load balancers SSL termination configuration. |
| PUT    | /loadbalancers/ <i>loadBalancerId</i> /ssltermination | Update/configure SSL termination.                      |
| DELETE | /loadbalancers/ <i>loadBalancerId</i> /ssltermination | Remove SSL termination.                                |

Normal Response Code(s): 202

Error Response Code(s): loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

The SSL Termination feature allows a load balancer user to terminate SSL traffic at the load balancer layer versus at the web server layer. A user may choose to configure SSL Termination using a key and an SSL certificate or an (Intermediate) SSL certificate.

When SSL Termination is configured on a load balancer, a secure shadow server is created that listens only for secure traffic on a user-specified port. This shadow server is only visible to and manageable by the system. Existing or updated attributes on a load balancer with SSL Termination will also apply to its shadow server. For example, if Connection Logging is enabled on an SSL load balancer, it will also be enabled on the shadow server and Cloud Files logs will contain log files for both.



### Note

SSL Termination may only be configured on load balancers with non-secure protocols. For example, SSL Termination can be applied to an HTTP load balancer, but not to an HTTPS load balancer.



### Warning

If SSL is enabled on a load balancer that is configured with nodes that are NOT in the same datacenter, then decrypted traffic will be sent in clear text over the public internet to the external node(s) and will no longer be secure.

A combination of the following required and optional attributes can be used to create the desired SSL configuration per account load balancer:

**Table 4.8. Attributes for SSL Termination**

| Name       | Default | Description   | Required |
|------------|---------|---|----------|
| securePort | N/A     | The port on which the SSL termination load balancer will listen for secure traffic.<br><br>The securePort must be unique to the existing LB protocol/port combination. For example, port 443. | Yes      |
| privatekey | N/A     | The private key for the SSL certificate.  | Yes      |

| Name   | Default | Description  | Required  |
|--|---------|--|---|
|  |         | The private key is validated and verified against the provided certificate(s).   |   |
| certificate  | N/A     | The certificate used for SSL termination.<br><br>The certificate is validated and verified against the key and intermediate certificate if provided.   | Yes   |
| intermediatecertificate  | N/A     | The user's intermediate certificate used for SSL termination.<br><br>The intermediate certificate is validated and verified against the key and certificate credentials provided.                                    | Yes, only when configuring Intermediate SSL Termination [a] |
| enabled  | true    | Determines if the load balancer is enabled to terminate SSL traffic.<br><br>If <code>enabled = false</code> , the load balancer will retain its specified SSL attributes, but will <i>not</i> terminate SSL traffic. | No  |
| secureTrafficOnly  | false   | Determines if the load balancer may accept <i>only</i> secure traffic.<br><br>If <code>secureTrafficOnly = true</code> , the load balancer will <i>not</i> accept non-secure traffic.                                | No  |
| [a] A user may only provide an intermediateCertificate when accompanied by a certificate, private key, and securePort. It may not be added to an existing SSL configuration as a single attribute in a future request. |         |  |   |

## Update SSL Termination

The following attributes may be updated without overwriting a load balancer's existing SSL certificate and key specifications:

- `enabled`
- `secureTrafficOnly`
- `securePort`

These configurable attributes must be provided (individually or grouped) in a request *without* specifying any certificate/key combination if a user does *not* want the system to overwrite the existing SSL certificate/key configuration.

If a user wants to replace the existing SSL configuration, a new certificate, key, and securePort combination must be provided instead of, or in addition to, the optional/editable attributes.

**Table 4.9. Optional SSL Attributes**

| Optional SSL Attributes                      | Non-SSL Traffic | SSL Traffic |
|--|-----------------|-------------|
| enabled = true (default)                     | Yes             | Yes         |
| enabled = false                              | Yes             | No          |
| secureTrafficOnly = true                     | No              | Yes         |
| secureTrafficOnly = false (default)          | Yes             | Yes         |
| enabled = true<br>secureTrafficOnly = true   | No              | Yes         |
| enabled = true<br>secureTrafficOnly = false  | Yes             | Yes         |
| enabled = false<br>secureTrafficOnly = false | Yes             | No          |
| enabled = false<br>secureTrafficOnly = true  | Yes             | No          |

The examples to view the load balancers SSL termination configuration are shown below:

#### Example 4.80. View Load Balancing SSL Termination Configuration Response: XML

```
<?xml version="1.0" encoding="UTF-8"?>

<sslTermination xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  enabled="true" securePort="500" secureTrafficOnly="false">
  <privatekey>-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAqSXePu8qLmniU7jNxoWq3SLkR8txMsl1gFYftpq7NIFaGfzV
f4ZswYdEYDVWWRepQjS0TvsB0d5+useUy/pcdZAlQLnn+540iLkvxKPVMzojUbG6
yOAmjC/xAZuExJHtfCrRHUQ4WQCwqyqANfP81ylinAb0zJGbtWUreV+nv8Ue77qX
77fOuqI6zOHinGZU7125XGLcVUpght8UtHZBzz2ahoftZ97DhUyQiSJCqCaHXJd3Q
eIHAq9qc7hu+usiYZWz34A0lw/gAl+RYcdvVc8kIwWxpiSieqqBPOwNzN5B0+9uu
5sDzMGMFnnSWcNKIPumX0rke3xFUL3UD6GJwvWIDAQABAoIBABQ7alT+yH3avm6j
OUHYtTJUPrf1VqnrFpMh061E3sWN/1gCbQse6hlP77bOSnDHqsA3i6Wy0mnAiOW
esVXQf3x6vLOCdiH+OKtu+/6ZMMG3jikWKI0ZYf5KAu4LW5RwiVK/c5RXagPtBIV
OFa7w299h0EAeAGMHSLaYhPXhDokyJa6yDkAQL3n+9L3V8kNWeCELfrqXnXF4X0K
CJp622tS/fW6kzppJyLJ4GPK9HNMpu02/n2Z7swWypfF+7set+9/aNTooDYWzCu
dbnRgqEIG1IP8+t6HG6x9VuJjVJLIW/WLITnQ/WTRXOQHGBhazgmwe1GPDxsQgXu
/wIcsIkCgYEASi0q+QhmJyoAm8vTHjo6+DD06YYTvSODLJOpoQr1ncGGDJ/evBw
x+9QsK3veXMBAK5G7Xss32IuXbBfjqQ89+/q/YT4BnS3T0OQa2WlR8tURNphCDr5
B3yD212kJTTeHC+p7BI9zhnWXD9kImh4vm4XcOsc9iqOSCZkGfvrPRsCgYEAS46t
Y85v2Pk235r1BPbgKwqYR+jElH4VWKu+EguUeQ4BLS47KktlLhvHtwrTv/UZ+1Px
8gSJTgyy7iEmzcGwPf1/MI5xg+DPgGhbr2G8EvrThmdHy+rPF2YSp1iBmJ4xq/1r
6XYKvf6ST3iujxTPU5xPEDUSLSh2ejJD/ddqSS0CgYEAkIdxyDa//8ObWWIjObSY
+4zIMBcyKFeernNKeMH/3FeW+neBOT/Sh7CgblK/28y1WUIZVghlOzePTC0BB+7c
b0eFUQ0YzF204rc+XW8coCt2xJEQaCtXxinUqGq1jmrIFNyv/MBt9BA+DSkcrRZp
js9SEyV1r+yPOyRvB7eIjhMCgYEAkd5yG+fkUlc6bfNb4/mPaUgFKD4AHUZEnzF+
ivhfW0y4+nGBXT285/VnjNs9508Aek3jmyJ2TTLhlbSW6obUX7flsR03Q1TLHd0p
xtPWT3D3kHotDwslzDN/KfYr6klxvvB0z0e3OFxsjiVTYiecuqb8UAVdTSED1Ier
Vre+v80CgYB86OqcAlR3diNaIwHgwK5kP2NAH1DaSwZXoobYpdkjsUQfJN5jwJbD
4/6HY0doc5xe0z8B+01VUZC+QA0gdXgHbmLZBIUeQU8sE4hGELoe/eWULXGwI91M
FyEWg03jZj8FkFh2954zwU6BOcbeL+9GrTdTPulvuHoTitmNEye4iw==
-----END RSA PRIVATE KEY-----</privatekey>
<certificate>-----BEGIN CERTIFICATE-----
```

```
MIIEWjCCA0KgAwIBAgIGATTGu/tMA0GCSqGSIb3DQEBBQUAMHkxCzAJBgNVBAYT
AlVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVGV4YXMxGjAYBgNVBAoTEVJh
Y2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLEwtSYWNRXhwiENBNTYMBYGA1UEAxMP
Y2E1LnJhY2tleHAub3JnMB4XDTEyMDE0MDgwNVoXDTM5MDUzMDUzMDUzMDUzMDUz
gZcxZCZAJBgNVBAYTAlVTMQ4wDAYDVQQIEwVUZXhhczEUMBIGA1UEBxMLU2FuIEFu
dG9uaW8xEDA0BgNVBAoTB1JhY2tleHAxEDA0BgNVBAStB1JhY2tleZXYxPjA8BgNV
BAMNW15c2l0ZS5jb20vZW1haWxBZGRyZXNzPXB0aWxsaxAudG9vaGlseEByYWNr
c3BhY2UyY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEaQsXePu8q
LmniuU7jNxoWq3SLkR8txMslgFYftpq7NIFaGfzVf4ZswYdEYDVWWRepQjS0TvsB
0d5+usEUy/pcdZAlQLnn+540iLkvxKPVmzojUbG6yOAmjC/xAZuExJHtfCrRHUQ4
WQCWyyqANfP8lylinAb0zJGbtWUreV+nv8Ue77qX77fOuqI6zOHinGZU7l25XGLc
VUpght8UthZBzz2ahoftZ97DhUyQiSJQCaHXJd3QeIHAq9qc7hu+usiYZWz34A0l
w/gAl+RYcdvVc8kIwXpiSieeqBPowNzN5B0+9uu5sDzMGMFnnSWcNKIPumX0rke
3xFUL3UD6GJvwIDAQAB04HIMIHFMIgJBgNVHSMegZswgZiAFIkXQizRaftxVdAL
P/Fb/F2ht017oX2kezB5MQswCQYDVQQGEwJVUzEOMAwGA1UECBMFVGV4YXMxGjAM
BgNVBACtBVRleGFzMR0wGAYDVQQKEwFSYWNrU3BhY2UgSG9zdGluZzEUMBIGA1UE
CxMLUmfja0V4cCBDQTQxGDAWBgNVBAMTD2NhNC5yYWNrZXhwLm9yZ4IBAjaDBgNV
HQ4EFgQUQUXhjcelJhjJDA4nhYcbeebMrIGYwDQYJKoZIhvcNAQEFBQADggEBACLe
vxcDSx9luQocluancb+vfkaNpvfAxOkUtrdRSHGXxvUkf/EJpIyG/M0jt5CLmEpE
UedeCfLRN+Qnsqt589ZemWWJwth/Jbu0wQodfSolcP0J2GFZDyTd5cWgm0Ix8A/
ZRGzNnTx3xskv6/1oh7so9ULppEbOsZTNqQ4ahbxbiaR2iDTQGF3XKSHha8093RB
YlnFahKZ2j0CpYvg0ljfN0Lv7j7Sm6GBA74n2OrGuB14H27wklD+PtIEFniyxKbq
5TD00l4yDgkR7PscmkZqK22GP9c3fQkmXodtpVlwrjcSaxxVWYm+S24XvMFERs3j
yXef+VJ0H+voAvxgbAk=
-----END CERTIFICATE-----</certificate>
<intermediateCertificate>-----BEGIN CERTIFICATE-----
MIIErzCCAY+gAwIBAgIBAjaNBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzEO
MAwGA1UECBMFVGV4YXMxGjAMBgNVBACtBVRleGFzMR0wGAYDVQQKEwFSYWNrU3Bh
Y2UgSG9zdGluZzEUMBIGA1UECxMLUmfja0V4cCBDQTQxGDAWBgNVBAMTD2NhNC5y
YWNrZXhwLm9yZzAeFw0xMjAxMTIxNzU3MDZaFw0xNDAxMTAxNzU3MDZaMHkxCzAJ
BgNVBAYTAlVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVGV4YXMxGjAYBgNV
BAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLEwtSYWNRXhwiENBNTYMBYGA
1UEAxMPY2E1LnJhY2tleHAub3JnMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAsVK6npit7Q3NLlVjkpiDj+QuIoYrhHTL5KKzj6CrtQsFYukELlYEKNlM
/dv8id/PkmdQ0wCnSk8d69CZKgO4hpN60/b2aU1/vQcrW5lv3fI8x4wLu2Ri92vJ
f04RiZ3Jyc0rgrfGyLyNjcnMIMjnfV7mQyy+7cMGKCDgaLzUGNyR5E/Mi4cERana
xyplnZi3DjAl1Kwums9cx5VzS0PolRyBsu7Xnpv3Fp2QqCBgdX8uaR5RuSak40/5
Jv2ORv28mi9AFu2AIRj6lrDdaLQGAXnbDk8b0ImEvVOe/QASsgTSmz0tn3q9Yejl
peQ9PFImVr2TymTF6UarGRHCWIdldQIDAQAB04HZMIHwMA8GA1UdEwEB/wQFMAMB
Af8wgaMGA1UdIwSBmzCBMIAUoeopOMWIEeYgtsI+T+ZjXWkc4ahfaR7MHkxCzAJ
BgNVBAYTAlVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVGV4YXMxGjAYBgNV
BAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLEwtSYWNRXhwiENBMzEYMBYG
A1UEAxMPY2E0LnJhY2tleHAub3JnggECMB0GA1UdDgQWBBSJF0Is0Wn7cVQ2iz/x
W/xdobdNeZANBgkqhkiG9w0BAQUFAAOCAQEAHUIe5D3+/j4yca1bxXg0egL0d6ed
Cam/1+E/SHxFJmlL0fkMndQqy/P3lPBNrHPdNw3CwK5hqFGL8oWGLifRmMvLWhBo
wDlwmzm++FQeEthhl7gBkgECxZ+U4+WRiqo9ZiHWDf49nr8gUONF/qnHHkXTOZKo
vB34N2y+nONDvyzky2wzbvU46dW7Wc6Lp2nLTt4amC66V973V31Vlpbzg3C0K7sc
PA2GGTsiW6Nf1mLd4fECgXslaQggoAKax7QY2yKrXLN5tmrHHTHV3fIvLbSNFJbl
dZsGmy48UFF4pBHdhne8bCat8KgK3BJbOXqNrUxxI6Jc/Hcl9AfppFIEGw==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIErzCCAY+gAwIBAgIBAjaNBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzEO
MAwGA1UECBMFVGV4YXMxGjAMBgNVBACtBVRleGFzMR0wGAYDVQQKEwFSYWNrU3Bh
Y2UgSG9zdGluZzEUMBIGA1UECxMLUmfja0V4cCBDQTQxGDAWBgNVBAMTD2NhMy5y
YWNrZXhwLm9yZzAeFw0xMjAxMTIxNzU3MDZaFw0xNDAxMTAxNzU3MDZaMHkxCzAJ
BgNVBAYTAlVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVGV4YXMxGjAYBgNV
BAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLEwtSYWNRXhwiENBNDYMBYGA
1UEAxMPY2E0LnJhY2tleHAub3JnMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApOqRiZRrgNSHs9VW3sfowlfQzepczUK1X+4SxpxIjHFN8QS+zQeYocHP
zdpHGQCLG35pWtY0iKmjMcA6AZZ8KHE0tCmGmOjEB2gjlAwOa0eHb2NHN44duu/n
```

```
ESEn2Njr05r2/q9bihjy7qQlVCrcRcXAQpj2F7t875Rq90a0d+AlHfGtN8su/S6y
G/fbUjP4fvIAzDjuhPoDlCGlZlJqo7EAy1kaqwh4jzvUt1WYcreRXNe6FJ4EMTyY
oeC/mbA9m/ZszlFE7WR2auY2yC2Q3gHBzTmJtvuxNTCn96n0EFpzzXBz0W7l9gu
jd+ikFjzt3Y5KhQMNMlXEMP80tvdPQIDAQABo4HZMIHWMA8GA1UdEwEB/wQFMAMB
Af8wgaMGA1UdIwSBmzCBmIAUQS5J4Ijc/J47kM0yVk5k1DH1Oo6hfaR7MHkxCzAJ
BgNVBAYTAlVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVGV4YXNMGjAYBgNV
BAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLewtSYWNRXhwIENBMjEYMBYG
AlUEAxMPY2EyLnJhY2tleHAub3JnggECMB0GA1UdDgQWBBSH6ik4xYgR5ga2Swj5
P5mNdYpzhjANBgkqhkiG9w0BAQUFAAOCAQEALmWrm7OXBru1H/1IqxNL+/Uky6BB
01Acwi7ESNDnsKd/m2G+SudlXy3v+fI6Im1qWBM8XthDHaYBQmjFTr+qOkbhQhOR
Z+T5s+zPF0yYo5hYU3xtotU84SusRFMZyW0KzIwgRvRsMexZmenCTNH0OW7J2/C
hLJ5rBZ9oX2X7arB65JdTu/EI/Zt32I83Xh/+GtK8mZegP12GOyDSnxuWyZi7noK
2lzoWKcxFo+qMwORgJ3ZO7BqANMUYQUoytK9nxJZUHBSpuQ08Kq9LTuIpdtyoJD
fGgT3quNreSCMmaTqxCGaTSOk1BuQDEbsVX+gYvULGfePNIUHYyFKdTA0w==
```

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

```
MIIErzCCAY+gAwIBAgIBA jANBgkqhkiG9w0BAQUFADB5MQswCQYDVQGEwJVUzEO
MAwGA1UECBMFVGV4YXNMGjAYBgNVBACtBVRleGFzMR0wGAYDVQQKEwFSYWNrU3Bh
Y2UgSG9zdGluZzEUMBIGA1UECXMUMFja0V4cCBDQTlxGDAWBgNVBAMTD2NhMi5y
YWNrZXhwLm9yZzAeFw0xMjAxMTIxNzU3MDRaFw0xNDAMTAxNzU3MDRaMHkxCzAJ
BgNVBAYTAlVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVGV4YXNMGjAYBgNV
BAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLewtSYWNRXhwIENBMjEYMBYG
AlUEAxMPY2EzLnJhY2tleHAub3JnMIIIBI jANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAMtodLv2WXOJgtUtcDJR6GYztsHsUoZQ+ jJg2N0bC0UmZbjbtKx+w+N1m
FBI6G5pMYCBzi3d0VGicGD3ZSIKEqoSnf3PHW5wJEJQjFqNcI0wcxJGrPACp3Th5
4bmLwUnxQt9OK+icmRMwvgtxPf6zk14JUC830oQ8WNyOXlT4qxJqSwDK51sViTYO
P912oyKmDqguKgs1xgWQz78ABWbRgu2Yg9+R9GybvUcyiSolqox+F1XVOoA8tF1E
1U8h3b1XCW80rzdHICvSulMnVGHa2gWyWpznQjinzuilQJZbtdDLEcFZJef1Tnl
/7Fh5Xo6n5KH4RclpheKaMkMoU2PBQIDAQABo4HZMIHWMA8GA1UdEwEB/wQFMAMB
Af8wgaMGA1UdIwSBmzCBmIAUfVXL/xzk1fBzmAKxZtd5YYcp3NmhfAR7MHkxGDAW
BgNVBAMTD2NhMS5yYWNrZXhwLm9yZzEUMBIGA1UECXMUMFja0V4cCBDQTEGjAY
BgNVBAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLewtSYWNRXhwIENBMjEY
CBMFVGV4YXNMGjAYBgNVBAYTAlVTggECMB0GA1UdDgQWBRRBLkngiNz8njuQzTJW
TmTUMfU6 jJANBgkqhkiG9w0BAQUFAAOCAQEAH9qo0y5EZSUpX2barHEKujeuLQnK
4cIyAoGBzyBTm9vev0ezLMXwXp/3J9KTSizLfrZRPmW2rFhy738nf6rI8aCCi+KE
afyI1EJTRZmgxDbANwVcK+k85yuWf4P27+4WL82E7c26wghldh52YLiz+GnfQMib
vTuSPbUubcg67CfEL7c4tgqhMzmcpKZwKbgzla0JkYfeLq8boclFYN+Rka9lo7OG
tyLdgpJ+aLwxQzgvAlqMLUilma026i8cN7kw56uNalVwSft6s39JVdlRYhrwoAAy
9T/mt/ioL4NW2rbC3XJVKSD+tRyfeb+5YjmGkPJKof19Ys5+Vro7N0n08g==
```

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

```
MIIErzCCAY+gAwIBAgIBA jANBgkqhkiG9w0BAQUFADB5MRgwFgYDVQQDEw9jYTEu
cmFja2V4cC5vcmcxFDASBgNVBAStClJhY2tFeHAgQ0ExMR0wGAYDVQQKEwFSYWNr
U3BhY2UgSG9zdGluZzEOMAwGA1UEBxMFVGV4YXNMGjAYBgNVBAGtBVRleGFzMR0w
CQYDVQGEwJVUzAeFw0xMjAxMTIxNzU3MDRaFw0xNDAMTAxNzU3MDRaMHkxCzAJ
BgNVBAYTAlVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVGV4YXNMGjAYBgNV
BAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLewtSYWNRXhwIENBMjEYMBYG
AlUEAxMPY2EyLnJhY2tleHAub3JnMIIIBI jANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAuEvwdPdXfltl7FbLUOSDPEMBRKcZwnNpfqNK2b7X5ADYFFvaLMHW6PGr
SHDRBpqpwmvJ28xgKZ+CoxHJhdHAWmTvk6h9ku08o8oyIBpD6YDNe95ApSvUCs
DTS3DW8GpNeHCKBPkUci4EazSeGkuKEPg+xWZoLm0USiTAbnbuskG/5ASw+KQNKU
DcBHKBylYm6KSlxkz+XOJO5hrMqGbe0bhRClqqQIh5WDMdriA5aLm07lFqmnwXz
koYsTmCwbbMMyl1FzDSA59k1BB+IA3UvD9LFbmH0GVWkueo5fOaqTcNkdSF34pG
GbnZYA4rGrgVBwxbjCzRmB2fCgtjEwIDAQABo4HZMIHWMA8GA1UdEwEB/wQFMAMB
Af8wgaMGA1UdIwSBmzCBmIAUOMPfFuJzzCcpUTLox0wDdc5iIt6hfaR7MHkxGDAW
BgNVBAMTD2NhMS5yYWNrZXhwLm9yZzEUMBIGA1UECXMUMFja0V4cCBDQTEGjAY
BgNVBAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLewtSYWNRXhwIENBMjEY
CBMFVGV4YXNMGjAYBgNVBAYTAlVTggEBMB0GA1UdDgQWBRR9Vcv/HOTV8HOYArFm
13lhhync2TANBgkqhkiG9w0BAQUFAAOCAQEAZlYt/0Calmm7fPN0kzixof50xej
GJ4LjELTaawVLEfl3dcmoAbqcGlaygAGxTVoSw47j3k0OyABUBSfGoWUkav2lkQg
```

```

rXUEnx8ToplVAvn/qZHTrrzJCLBk/K/BzBhBnVf3ma5GkJOkcwQd3Cn7FjKz19Be
oisPp9fQ5WBeRO5QizJDjgJ8LS63ST0lni7/U2EhBIdfoBM5vMnGhc5Ns6mamPjJ
jH3zzLdtGaN6UzjUUUVTAoah0qHsL4K7haFA0uiJldiCt8mZfN7F6nzb23GvUAdK
ZLtkSGD042R/ppnfdZ5NautNxA9tNVH0pkjXkba/qzGz935brilSvxIzzg==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDnzCCAogAwIBAgIBATANBgkqhkiG9w0BAQUFADB5MRgwFgYDVQQDEw9jYTEu
cmFja2V4cC5vcmcxZDASBgNVBAsTC1JhY2tFeHAgQ0ExMR0wGAYDVQQKExFSYWNr
U3BhY2UgSG9zdGluZzEOMAwGA1UEBxMFVGV4YXNjZjAMBgNVBAGTBVRleGFzMQsw
CQYDVQQGEwJVUzAeFw0xMjAxMTIxNzU3MDRaFw0xNDAxMTEzNzU3MDRaMHkxGDAW
BgNVBAMTD2NhMS5yYWNrZXhwLm9yZzEUMBIGA1UECzMUMFja0V4cCBDAQTEGjAY
BgNVBAoTEVJhY2tFeGFjZSBib3N0aW5nMQ4wDAYDVQQHEWVUZXhhczEOMAwGA1UE
CBMFVGV4YXNjZjAMBgNVBAYTA1VTMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAn+my3GNUG8jOEnWMREdDzjLskljm3mPtPUVJCyf6pQmXbpAsCp8mpQH
L7AS2BVHImpq7762Q29u46j+W+6wmdn3rZaZsQ6HZrkvlzTxip6oJtMszobkrdsB
ZFTH2kvNWpktgAauxc9Dr6oinBYGr62vFz+LI93CPloI7gv7N8YABkdWnNuqrYdtA
wE4OMdXylkWWi7jENZdRmb8A6qmQj1NZmv5Jgwggxy40fH4m88GK098Pr16oerlX
als7HdWcPk3iglOhxN0+sg88mufWNr71YsQ5bloVhtv/5qzsq/DdPrOpffhJYRPs
A+YgavRfrKSWz4fuZOBqaXGnNdf+NQIDAQABoziwMDAPBgNVHRMBAf8EBTADAQH/
MB0GA1UdDgQWBQBQ4w98W4nPMJylRMuJHTAN1zmIi3jANBgkqhkiG9w0BAQUFAAOCA
AQEAMjB0DHQn5C2WpWXZEEEAQvGmzC/NvoJ9K7Kkizpd9I8GOz5/cpLtEXSqd1q7
2aOrLb9b5jtuWiu9rpkxo/vX5jMCPHW/jr+51v2InSfe8SJSgcciGFdFBz++rve
DhMvprCgbwWnyqHd+2B8KoLt9k/x5MUWPTRMtlonOve7+wgidwgyQLeZuQp0jg8
/dGFHwFi/6Ns2Cd5UKT8sbt221N0uatddQ9bwJ0dFg0tvh6aVNRa121mYtmtSsU9
BF9RsonnOUCtYQRR+ovVvAyT0XKbfixtwnpdW26vd5BKJQ1X5i3WlrrssQwzPYBIW
LE3/pvvh3Ar83QycrLE/w1/KA==
-----END CERTIFICATE-----</intermediateCertificate>
</sslTermination>

```

### Example 4.81. View Load Balancing SSL Termination Configuration Response: JSON

```

{
  "certificate": "-----BEGIN CERTIFICATE-----\n
nMIIEXTCCA0WgAwIBAgIGATTEAJK3MA0GCSqGSIb3DQEBAQUAMIGDMRkwFwYDVQQD\
nExBUZXXN0IENBIFNudWIgS2V5MRcwFQYDVQQLEw5QbGF0Zm9ybmSBMYmFhcEaMBGg\
nA1UEChMRUmfja3NwYWNlIEhvc3RpbmcxZDASBgNVBACTC1NhbiBBbnRvbmlvMQ4w\
nDAYDVQQIEwVUZXhhczELMAkGA1UEBhMCVVMwHhcNMTIwMTA5MTk0NjQ1WhcNMTQw\
nMTA4MTk0NjQ1WjCBGjELMAkGA1UEBhMCVVMxZjAMBgNVBAGTBVRleGFzMRQwEgYD\
nVQQHEwTYW4gQW50b25pbzEaMBGGA1UEChMRUmfja3NwYWNlIEhvc3RpbmcxZzAV\
nBgNVBAsTDlBsYXRmb3JtIExiYWZzMRgwFgYDVQQDEw9UZXN0IENSaWVudCBLZXkw\
nggEiMA0GCSqGSIb3DQEBQUAA4IBDwAwggEKAoIBAQAi51Iy1FhHtNLT8CONVfc\
nOBfAsP2D5es1qhrOWHCGLgAuDMksBsCc7FPo5PSBOMQ+6z8HtCFbrLoC5/Zx0F5b\
nfVegjA+xKjI2HGASsYHHM0BFEH2UjUcJrWiMWtxQuW6Phbqulo7JwjmygMEMIkeK\
nf+FtkE9mrq+E8K40/thrjxl3/ZcJD1+3dcp+ZuzVJ2t1E4iGKCx79IZFsYsKiuf+\
n+E0i6iGvvI6UcbcZxVxQj2TPlJkFuoX5KdGClIX9Dr9y6aJ4SCh+GRhvH1+DTaz0\
nnCvghachHZtIeztRDqxWApj00zs93dSelrviMXDr8fqyEAGg7YIhgui0azBsWCen\
nAgMBAAGjgdUwgdIwgbAGAlUdIwSBQDCBpYAUNpxlPc6cGA7KqEwHmHBTZMA7lSh\
ngYmkgyYwgmYGTAXBgNVBAMTEFRlc3QgQ0EgU1RlYiBLZXkxZzAVBgNVBAsTDlBs\
nYXRmb3JtIExiYWZzMR0wGAYDVQQKExFSYWNrc3BhY2UgSG9zdGluZzEUMBIGA1UE\
nBxMLU2FuIEFudG9uaW8xDjAMBgNVBAGTBVRleGFzMQswCQYDVQQGEwJVUzE0BATA\
nBgNVHQ4EFgQUueOfsJZZOHwJHZwBy6u0swnpccwDQYJKoZIhvcNAQEFBQADggEB\
nAFNuqSVUaotUJOWDv4z7Kbi6JFPtjDht5ORw4BdVYLRD4h9DACAFzPrPV2ym/Osp\
nhNMdZq6msZku7MdOSQVhdeGWRsNk3M8O9Hg7cVzPNXOF3iNoo3irQ5tURut44xs4\
nWw5YwQqS9WYUY5snD8tm7YlRQTPfhg+678xIq/zWCv/u+fSnfVvlnlhLVQkBeG/Y\
nghluMaTIpUKTGEjIAGtGP7wwiXpTR/HyfhztUSTaWc1Ef7zoKT9LL5z3IVlhC2\njVWz
+RwYs98LjMuksJFoHqRfWyYhCIym0jb6GTwaEmpxajc+d7OLNQdnoEGoUYGP\nYjtfkRYg265ESMA
+Kww4Xy8=\n-----END CERTIFICATE-----\n",

```



```

    "enabled":true,
    "secureTrafficOnly":false,
    "privatekey":"-----BEGIN RSA PRIVATE KEY-----\n
nMIEEpAIBAAKCAQEAwIudSMpRZx7TS0/AtDVX3DgXwLD9g+XrNaoazlhwhpYALgzJ\
nLAbAnOxt6T0T0gTpkPus/B7QhW6y6Auf2cdBeW31XoIwPsSoyNhxgErGBxzNARRB9\
nliIHCalojFrcULluj4W6rpaOycI5soDBJiJHin/hbZBPZq6vhPCuNP7Ya48Zd/2X\
nCQ9ft3XKfmbS1SdrdROIhigse/SGRbMrCorn/vhNIuohr7y0LHG3GcVcUI9k6ZSZ\
nBbqF6+ZA4ApSF/Q6/cumieEgofhkYbx5fg02s9Jwr4IWnIR2bSHs7UQ6sVgKYzjs7\
nPd3Unpa74jFw6/H6shABoO2CIYLotGmQbFgnpwIDAQABAoIBAQCBCQ+PCiC1JHNv\
ntUzfeCA5ZR4F9JbxHdRTUnxEbOB8UWotckQfTScoAvj4yvDQ42DrCZxj/UOdvFOs\
nPufZvlp91bIz1alugWjE+p8n5+2hIaegoTyHoWZKBfxak0myj5KYfHZvKlbmv1ML\
nXV4TwEVRfAIG+v87QTY/UUxuF5vR+BpKlbgUJLfPUFFvJUd184qsJ44pToxaYUd/\
nh5YAGC00U4ay1KVSAUnTkPNZ01PG/rWU6w6WcTvNRLMd8DzFLTKLogQfHhbExAF\
n+sXPWjWSzbBRP107fHqQ96QQh4VFiy/7w9W+sDKQyV6U117OSXs6aZ4f+lq4rJTI\
n1FG96YiBAoGBA01tiH0hloWDBYfJB3KJj6CQQsDGwtHo/DEgznFVP4XwEVbZ98Ha\
nBfBCn3sAybbaiakyCV1Hwj7kfHmZPDHbrCUSFX7quu/2zPK+w03LZXSYu4YsguSa\
nRedInN33PpdlPhLyQdWSu5SVHJDF6xn22vlyxeILH3ooLg2WOFMPmVAoGBAM+b\
nUG/a7iyfpaQKYyufASxz6SeFaDY+ZYEX45L112H8Pu+Ie/qzon+bzLB9FIH8GP6+\
nQpQgmm/p37U2gD1zChUv7iW6OfQBKk9rWvMpfRF6d7YHquElejhizfTZ+ntBV/VY\
ndOYEczxhRdW7keLpatYaaWUy/VboRZmlz/9JGqVLAoGAHfQNmFc0cgk4IowEj7a3\
ntTnh61tub/i+FynwRykfazcDyXaeLPDtfQe8gVh5H8h6W+y9P9BjJVnDVVrX1RAn\
nbiJ1EupLPF5sVDapW8ohTOXgfbGTGXBNuUW+4Nv+IDno+mz/RhjkPYHpnM0I7c/5\
ntGzOZsC/2hjNgT8I0+MWav0CgYEAuULdJeQVlKalI6HtW2GnlURRVJ49H+LQkY6e\
nW3+cw2jo9LI0CMWSphNvNrN3wIMp/vHj0fHCP0pSapDvIWbuQXfzKaGko7UCf7rK\
nf6GvZrChkV4IREBAb97j8bMvThxClMNqFfU0rFZyXP+0MOyhFQyertswrgQ6T+Fi\
n2mnvKD8CgYAmJHP3NTDRMoMRyAzonJ6nEaGUBAgNmivTaUWMe0+leCvAdwB89gzC\
nTKbm3eDUg/6Va3X6ANh3wsfIOe4RXXxcbcFDk9R4z02M5gfLSjYm5Q87EBZ2hrdj\
nM2gLI7dt6thx0J8lR8xRHBEMrVBdgp0glgQzo5dAV88/kpkZVps8Q==\n-----END RSA
PRIVATE KEY-----\n",
    "intermediateCertificate":"-----BEGIN CERTIFICATE-----\n
nMIIDtTCCAp2gAwIBAgIBATANBgkqhkiG9w0BAQUFADCBGzEZMBcGA1UEAxMQVGZv\
ndCBDQSBTVHViIEtleTEXMBUGA1UECXMouGxhdGZvcml0gTGJhYXNjZjYAYBgNVBAoT\
nEVJhY2tzcGFzZSBib3N0aW5nMRQwEgYDVQOHEwtYVW4gQW50b25pbzEOMAwGA1UE\
nCBMFVGV4YXNjZSBib3N0aW5nMRQwEgYDVQOHEwtYVW4gQW50b25pbzEOMAwGA1UE\
nNDU0OwogYXNjZSBib3N0aW5nMRQwEgYDVQOHEwtYVW4gQW50b25pbzEOMAwGA1UE\
nYXRmb3JtIEExYWZzMRowGAYDVQQKEzFSYWNrc3BhY2UgSG9zdGluZzEUMBIGA1UE\
nBxMLU2FuIEFudG9uaW8xDjAMBgNVBAgTBVRleGFzMQswCQYDVQQGEwJVUzCCASIw\
ndQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANh551wTVwQvNoEZjq1zGdYz9ja\
nXXdjizn8AJhjHLOAallfPtvCfTEgKanhdoyz5FnhQE8HbDAop/KNS11N2UMvd15f\
nZNLTSjJrNtedqxQwXN/i3bpyBxNvejUH2NjVlmyj+5CJYwCzWalvI/gLPq/A3as\
no2EQqt3f3U8unRgn0zXLRdYxV9MrUzNAmdipPNvNrsVdrCgA42rgF/8KsyRVQfJCX\
nfN7PGCfrsC3YaUvhyraWxNnXiZMYTNa9wEeBZLUw8SlEtpalZsvui+TPXu3USNZ\
nVnWH8Lb6ENlnoX0VBwo62fjOG3JzhNkoJawi3brQyDdINovaf7iPrs/T8CAwEA\
nAaMyMDAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUUNpx1Pc6cGA7KqEwHmHB\
nTZMA7lQwDQYJKoZIhvcNAQEFBQADggEBAMoRgH3iTG3t317viLkoY+lNMHUGHuR7\
nb3mn9MidJKyYVewe6hCDIN6WY4fUoJmMW9wFJWJIo0hRMNHL3n3tq8HP2j20Mxy8\
nacPdfGZJa+jjBw72CrIGdobKaFduI1IEDBA1pNdZIJ+EulrtqrMesnIt92WaypIS\
n8JycbIgDMCIyC0ENHEk8UW1C6429c7OZAsplMTbHME/1R4btXjkdfRyZJjdJ2yL2\
n8cjZDUDMCPTdW/ycP07Gkq30RB5tACB5aZdaCn2YaKC8FsEdhff4X7xEOfOEHWeg\
nSRxADDj8Lx1MT6QpR07hCiDyHfTctbqzI0igjX630h7xXSA0f+JVta8=\n-----END
CERTIFICATE-----\n",
    "securePort":443}

```



## Note

All requests to SSL termination in the JSON format require the key/certificates to be in "proper" JSON format, meaning that all raw line feed characters should be wrapped in a newline character. So if the user pastes in the key

from a `mykey.key` file, JSON will not properly handle the field. The key/certificates can be wrapped in a newline character in Java, for example, using `string.replaceAll("\n", "\\n")`. Please refer to the examples above for working, "pre-formatted" examples.

The examples to update the load balancers SSL termination full certification request are shown below:

### Example 4.82. Update Load Balancing SSL Termination Full Certification Request: XML

[illegible]

```
HQ4EFgQUQUXHjcelJhjJDA4nhYcbeMrIGYwDQYJKoZIhvcNAQEFBQADggEBACLe
vxCD5x9luQocluancb+vfkAnpvfAxOkUtrdRSHGXxvUkf/EJpIyG/M0jt5CLmEpE
UedeCF1RN+Qnsqt589ZemWWJwth/Jbu0wQodfSolcP0J2GFZDyTd5cWgm0Ix8A/
ZRGzNnTx3xskv6/1Oh7so9ULppEbOsZTNqQ4ahbxbiaR2iDTQGF3XKSHha8093RB
YlnFahKZ2j0CpYvg0ljJfN0Lv7jSm6GBA74n2OrGuB14H27wklD+PtIEFniyxKbq
5TD0014yDgkR7PscmkZqK22GP9c3fQkmXodtpVlwRjcSAXxVWYm+S24XvMFERs3j
yXef+VJ0H+voAvxgbAk=
-----END CERTIFICATE-----</certificate>
<intermediateCertificate>-----BEGIN CERTIFICATE-----
MIIErzCCAy+gAwIBAgIBAJANBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzEO
MAwGA1UECBMFVG4YXMXDjAMBgNVBAcTBVRleGFzMR0wGAYDVQQKEwF5SYWNRU3Bh
Y2UgSG9zdGluZzEUMBIGA1UECXMUMFja0V4cCBBDQTMxGDAWBgNVBAMTD2NhNC5y
YWNrZXhwLm9yZzAeFw0xMjAxMTIxNzU3MDZaFw0xNDAMTAxNzU3MDZaMHkxXzAJ
BgNVBAYTALVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVG4YXMXGjAYBgNV
BAoTEVJhY2tTcGFjZSBIb3N0aW5nMRQwEgYDVQQLewtSYWNRXhwIENBNTEYMBYG
A1UEAxMPY2E1LnJhY2tleHAub3JnMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAsVK6npit7Q3NLLVjkpiDj+QuIoYrhHTL5KKzj6CrtQsFYukEL1YEKNLM
/dv8id/PkmdQ0wCNsk8d69CZKgo4hpN6O/b2aU1/vQcrW5lv3fI8x4wLu2Ri92vJ
f04RiZ3Jyc0rgrfGyLyNjcnMIMjnFV7mQyy+7cMGKCDgaLzUGNyR5E/Mi4cERana
xyp1nZi3DjA1lKwums9cx5VzS0PolRyBsu7Xnpv3Fp2QqCBgdX8uaR5RuSak40/5
Jv2ORv28mi9AFu2AIRj6lrbDaLQGAXnbDk8b0ImEvVOe/QASsgTsmzOtn3q9YeJl
peQ9PFImVr2TymTF6UarGRHCWidldQIDAQABo4HZMIHWMA8GA1UdEwEB/wQFMAMB
Af8wgaMGA1UdIwSBmzCBMIAUoeopOMWIEeYgtsI+T+zjXWKc4ahfar7MHkxCzAJ
BgNVBAYTALVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVG4YXMXGjAYBgNV
BAoTEVJhY2tTcGFjZSBIb3N0aW5nMRQwEgYDVQQLewtSYWNRXhwIENBNTEYMBYG
A1UEAxMPY2E1LnJhY2tleHAub3JnMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApoQriZRRgNSHs9VW3sfowlfQzpeczUK1X+4SxpxIjHFN8QS+zQeYochP
zdpHGcQLG35pwtY0iKmjMcA6AzZ8KHE0tCmGm0jEB2gjlAwOa0eHb2NHN44duu/n
ESEN2NJR05r2/q9bihjy7qQ1VCrcRcXAQpj2F7t875Rq90a0d+AlHfGtN8su/S6y
G/fbuJp4fvIAzDjuhPoD1CG1zIJqo7EAy1kaqwh4jzvUt1WYcreRXNe6FJ4EMtyY
oeC/mbA9m/Zsz1FE7WR2auY2yC2Q3gHBzTmJtvuxNTCn96n0EFpzzXBz0W7w19gu
jd+ikFjzt3Y5KhQMNM LXEMP80tvdPQIDAQABo4HZMIHWMA8GA1UdEwEB/wQFMAMB
Af8wgaMGA1UdIwSBmzCBMIAUQSS5J4Ijc/J47kM0yV5k1DH1Oo6hfar7MHkxCzAJ
BgNVBAYTALVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVG4YXMXGjAYBgNV
BAoTEVJhY2tTcGFjZSBIb3N0aW5nMRQwEgYDVQQLewtSYWNRXhwIENBNTEYMBYG
A1UEAxMPY2E1LnJhY2tleHAub3JnMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA1ACwi7ESNDnsKd/m2G+SUD1Xy3v+fI6Im1qWBM8XthdHAYBQmjFTr+qokbhQhOR
Z+T5s+zPF0yYo5hYU3xtotUL84SusRFMZyW0KzIwgrvRsMexZmenCTNH0OW7J2/C
hLJ5rBZ9oX2X7arB65JdTu/EI/Zt32I83Xh/+GtK8mZegP12GOyDSnxuWyZi7noK
21zowKcxFo+qMwOrgJ3ZO7BqANMUYQHUoytK9nxJZUHBSpuQ08Kq9LTuIpdtyoJD
fGgT3quNreSCMmaTqxCGaTSOk1BuQDEbsVX+gYvULGfepNIUHYyFKdTA0w==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIErzCCAy+gAwIBAgIBAJANBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzEO
MAwGA1UECBMFVG4YXMXDjAMBgNVBAcTBVRleGFzMR0wGAYDVQQKEwF5SYWNRU3Bh
Y2UgSG9zdGluZzEUMBIGA1UECXMUMFja0V4cCBBDQTMxGDAWBgNVBAMTD2NhMy5y
YWNrZXhwLm9yZzAeFw0xMjAxMTIxNzU3MDZaFw0xNDAMTAxNzU3MDZaMHkxXzAJ
BgNVBAYTALVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVG4YXMXGjAYBgNV
BAoTEVJhY2tTcGFjZSBIb3N0aW5nMRQwEgYDVQQLewtSYWNRXhwIENBNTEYMBYG
A1UEAxMPY2E1LnJhY2tleHAub3JnMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEApoQriZRRgNSHs9VW3sfowlfQzpeczUK1X+4SxpxIjHFN8QS+zQeYochP
zdpHGcQLG35pwtY0iKmjMcA6AzZ8KHE0tCmGm0jEB2gjlAwOa0eHb2NHN44duu/n
ESEN2NJR05r2/q9bihjy7qQ1VCrcRcXAQpj2F7t875Rq90a0d+AlHfGtN8su/S6y
G/fbuJp4fvIAzDjuhPoD1CG1zIJqo7EAy1kaqwh4jzvUt1WYcreRXNe6FJ4EMtyY
oeC/mbA9m/Zsz1FE7WR2auY2yC2Q3gHBzTmJtvuxNTCn96n0EFpzzXBz0W7w19gu
jd+ikFjzt3Y5KhQMNM LXEMP80tvdPQIDAQABo4HZMIHWMA8GA1UdEwEB/wQFMAMB
Af8wgaMGA1UdIwSBmzCBMIAUQSS5J4Ijc/J47kM0yV5k1DH1Oo6hfar7MHkxCzAJ
BgNVBAYTALVTMQ4wDAYDVQQIEwVUZXhhczEOMAwGA1UEBxMFVG4YXMXGjAYBgNV
BAoTEVJhY2tTcGFjZSBIb3N0aW5nMRQwEgYDVQQLewtSYWNRXhwIENBNTEYMBYG
A1UEAxMPY2E1LnJhY2tleHAub3JnMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA1ACwi7ESNDnsKd/m2G+SUD1Xy3v+fI6Im1qWBM8XthdHAYBQmjFTr+qokbhQhOR
Z+T5s+zPF0yYo5hYU3xtotUL84SusRFMZyW0KzIwgrvRsMexZmenCTNH0OW7J2/C
hLJ5rBZ9oX2X7arB65JdTu/EI/Zt32I83Xh/+GtK8mZegP12GOyDSnxuWyZi7noK
21zowKcxFo+qMwOrgJ3ZO7BqANMUYQHUoytK9nxJZUHBSpuQ08Kq9LTuIpdtyoJD
fGgT3quNreSCMmaTqxCGaTSOk1BuQDEbsVX+gYvULGfepNIUHYyFKdTA0w==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIErzCCAy+gAwIBAgIBAJANBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzEO
```

```
MAwGA1UECBMFVGv4YXNjZjAMBGNVBAcTBVRleGFzMRowGAYDVQQKEExFSYWNrU3Bh
Y2UgSG9zdGluZzEUMBIGA1UECXMUMFja0V4cCBDAQIwGDAWBgNVBAMTD2NhMi5y
YWNrZXhwLm9yZzAeFw0xMjAxMTIwMjUzMDRaFw0xNDAMTAxNzUzMDRaMHkxZCZAJ
BgNVBAYTAlVTMQ4wDAYDVQQIEWVUZShhczEOMAwGA1UEBxMFVGv4YXNjZjAeFw0x
BAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLewtSYWNrRXhwIENBMzEYMBYG
A1UEAxMPY2EzLnJhY2tleHAub3JnMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAmtoDLv2WxOJgtUtdDJR6GYztsHsUoZQ+ jJg2N0bC0UmZbjbtKx+w+Nlm
FbiBG5pMYCBzi3d0VGicGD3ZSIKEqoSnf3PHW5wJEJQjFqNcI0wcxJGrPACp3Th5
4bmLwUnxQt9OK+icmRMwvqtxPf6zk14JUC830oQ8WNYOXlT4qxJqSwDK51sViTYO
P912oyKmDgguKgs1xgWQz78ABWbRgu2Yg9+R9GybvUcyiSolqox+F1XVOoA8tF1E
lU8h3b1XCW80rzrdHICvSulMnVGhA2gWyWpznQjinzui1QJZbtDLEcfZJef1Tnl
/7Fh5Xo6n5KH4Rc1pheKaMkMoU2PBQIDAQABo4HZMIHWMA8GA1UdEwEB/wQFMAMB
Af8wgaMGA1UdIwSBmzCBMIAUfVXL/xzk1fBzmAKxZtd5YYcp3NmhfAR7MHkxGDAW
BgNVBAMTD2NhMS5yYWNrZXhwLm9yZzEUMBIGA1UECXMUMUMFja0V4cCBDAQTEGjAY
BgNVBAoTEVJhY2tTcGFjZSBib3N0aW5nMQ4wDAYDVQQHEWVUZShhczEOMAwGA1UE
CBMFVGv4YXNjZCZAJBgNVBAYTAlVTggECMB0GA1UdDgQWBRRBLkngiNz8njuQzTJW
TmTUMfU6 jJANBgkqhkiG9w0BAQUFAAOCAQEAH9qo0y5EZSUPX2baRHEkUjeuLQnK
4cIyAoGBzyBTm9vev0ezLMwXp/3J9KTSizLfrZZPMw2rFhy738nf6rI8aCCi+KE
afyI1EJTRZmgxDbANwVcK+k85yuWf4P27+4WL82E7c26wghldh52YLIz+GnfQMib
vTuSPbUubcg67CfEL7c4tgqhMzmcpKZwKbgzla0JkYfeLq8boc1FYn+RKA9lo7OG
tyLdgpJ+aLwxQzgvAlqMLUilma026i8cN7kw56uNa1VwSft6s39JVdlRYhrwAAy
9T/mt/ioL4NW2rbC3XJVKSD+tRyfeb+5YjmGkPJKof19Ys5+Vro7N0n08g==
```

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

```
MIIEFzCCAy+gAwIBAgIBAgjANBgkqhkiG9w0BAQUFADB5MRgwFgYDVQQDEw9jYTEu
cmFja2V4cC5vcmcxFDASBgNVBAsTC1JhY2tFeHAQOExMRRowGAYDVQQKEExFSYWNr
U3BhY2UgSG9zdGluZzEOMAwGA1UEBxMFVGv4YXNjZjAeFw0xMjAxMTIwMjUzMDRa
Fw0xNDAMTAxNzUzMDRaMHkxZCZAJBgNVBAYTAlVTMQ4wDAYDVQQIEWVUZShhczEOMAw
GA1UEBxMFVGv4YXNjZjAeFw0xBAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLewtSYWNrRXhwIENBMzEYMBYG
A1UEAxMPY2EzLnJhY2tleHAub3JnMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAuEvwdPdXfltl7FbLUOSDPEMBRkcZwnNpfqNk2b7X5ADYFFvaLMHW6PGr
SHDRBpqpwmvj28xgKZ+CoxHJhdHAWmTvk6h9ku08o8oyIBpD6YDNe95ApSvUCs
DTS3DW8GpNeHCKBPkUci4EazSeGkuKEpG+xWZoLm0USiTAbnbuskG/5ASw+KQNKU
DcBHKBYlym6KS1xkz+XOJO5hrMqGbe0bhhRCLqqQIh5WDMDrIA5aLm071FqmnwXz
koVsTmCwbbMMyl1fZDSA59klBB+IA3UvD9LFbmH0GVWkueo5f0AqTcNkdSFC34pG
GbnZYA4rGrgVBwxbjCzRmB2fCgTjEwIDAQABo4HZMIHWMA8GA1UdEwEB/wQFMAMB
Af8wgaMGA1UdIwSBmzCBMIAUOMPfFuJzzCcpUTLox0wDdc5iIt6hfAR7MHkxGDAW
BgNVBAMTD2NhMS5yYWNrZXhwLm9yZzEUMBIGA1UECXMUMUMFja0V4cCBDAQTEGjAY
BgNVBAoTEVJhY2tTcGFjZSBib3N0aW5nMQ4wDAYDVQQHEWVUZShhczEOMAwGA1UE
CBMFVGv4YXNjZCZAJBgNVBAYTAlVTggECMB0GA1UdDgQWBRRB9Vcv/HOTV8HOYArFm
13lhync2TANBgkqhkiG9w0BAQUFAAOCAQEAZ1Yt/0Calmm7fPN0kzixof50xej
GJ4LjELTaawVLEfl3dcmoAbqcGlaygAGxTVoSwa7j3k00yABUBSfGoWUkav2lkQg
rXUEnx8ToplVAvn/qZHTrrzJCLBk/K/BzBhBnVf3ma5GkJ0kcwQd3Cn7FjKz19Be
oisPp9fQ5WBeRO5QizJDjg78LS63ST0lni7/U2EhBIdfoBM5vMnGhc5Ns6mamPjJ
jh3zzLdtGan6UzjUUUVTAoah0qHsL4K7haFA0uiJldiCt8mZfn7F6nzb23GVuAdK
ZLtkSGD042R/ppnfdZ5NautNx9tNVH0pkjXkba/qzGz935brilSvxIzzg==
```

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

```
MIIDnzCCAogAwIBAgIBATANBgkqhkiG9w0BAQUFADB5MRgwFgYDVQQDEw9jYTEu
cmFja2V4cC5vcmcxFDASBgNVBAsTC1JhY2tFeHAQOExMRRowGAYDVQQKEExFSYWNr
U3BhY2UgSG9zdGluZzEOMAwGA1UEBxMFVGv4YXNjZjAeFw0xMjAxMTIwMjUzMDRa
Fw0xNDAMTAxNzUzMDRaMHkxZCZAJBgNVBAYTAlVTMQ4wDAYDVQQIEWVUZShhczEOMAw
GA1UEBxMFVGv4YXNjZjAeFw0xBAoTEVJhY2tTcGFjZSBib3N0aW5nMRQwEgYDVQQLewtSYWNrRXhwIENBMzEYMBYG
A1UEAxMPY2EzLnJhY2tleHAub3JnMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAAn+myn3GNUG8jOEnwMREdDzjLsk1jm3mPtPUVJCyfpQmXbpAsCp8mpQH
L7AS2BVHImpq7762Q29u46j+W+6wmdn3rZaZsQ6HZrkvlzTxip6oJtMs zobkrdsB
ZFTH2kvNWpktgAuxc9Dr6oinBYGr62vFz+LI93CPlOI7gv7N8YABkdWnNuqrYdtA
wE4OMdXy1kWWi7jENZdRmb8A6qmQj1NZmv5Jgwggy40fH4m88GK098Pr16oer1X
```

```
als7HdWCpk3iglOhxN0+sg88mufWNR7lYsQ5bloVhtv/5qzsq/DdPrOpffhJYRPs
A+YgavRfrKSWz4fuZOBqaXGnNdf+NQIDAQABozIwMDAPBgNVHRMBAf8EBTADAQH/
MB0GA1UdDgQWBQBQ4w98W4nPMJylRMuJHTAN1zmIi3jANBgkqhkiG9w0BAQUFAAOc
AQEAMjB0DHQn5C2WpWXZEEEAQvGmzC/NvoJ9K7Kkizpd9I8GOz5/cpLtEXSqdIq7
2aOrLb9b5jtuWiu9rpko/vX5jMCPHW/jr+5lV2InSfe8SJSgcciGFdFBz++rve
DhMvprCgbWwnyqHd+2B8KoLt9k/x5MUWPTRMtLonOve7+wgIwdgyQLeZuQp0jg8
/dGFHwFi/6Ns2Cd5UKT8sbt22lN0uatddQ9bwJ0dFg0tvh6aVNRa12lmYtmtSsU9
BF9RsonnOUtCYQRR+ovVvAyT0XKBfixtwnpW26vd5BKJQ1X5i3WlrsQwzPYBIW
LE3/pvvbh3Ar83QycrLE/wl/KA==
-----END CERTIFICATE-----</intermediateCertificate>
</sslTermination>
```

### Example 4.83. Update Load Balancing SSL Termination Full Certification Request: JSON

```
{
  "certificate": "-----BEGIN CERTIFICATE-----\n
nMIIEHTCCA0WgAwIBAgIGATTEAjk3MA0GCSqGSIb3DQEBBQUAMIGDMRkwFwYDVQQDD\
nExBUZUXN0IENBIFNudWIgS2V5MRcwFQYDVQQLLEw5QbGF0Zm9ybSBMYmFhc2EaMBG\
nAlUEChMRUmFja3NwYWNlIEhvc3RpbmcxPDASBgNVBACTC1NhbiBBbnRvbmlvMQ4w\
nDAYDVQQIEwVUZXBhc2ELMAKGA1UEBhMCVVMwHhcNMTIwMTA5MTk0NjQ1WhcNMTQw\
nMTA4MTk0NjQ1WjCBgJELMAKGA1UEBhMCVVMwDjAMBgNVBAGTBVRleGFzMRQwEgYD\
nVQQHEwTYW4gQW50b25pbzEaMBGGA1UEChMRUmFja3NwYWNlIEhvc3RpbmcxPDAS\
nBgNVBAsTDlBsYXRmb3JtIExiYWZzMRgwFgYDVQQDEw9UZUXN0IENSaWVudCBLZXkw\
nggEiMA0GCSqGSIb3DQEBBQUAA4IBDwAwgEKAoIBAQDAi5lIylFnHtNLT8CONVfc\
nOBfAsP2D5eslqhrOWHCGlgAuDMksBsCc7FPo5PSBOMQ+6z8HtCFbrLoC5/Zx0F5b\
nfVegjA+xKjI2HGASsYHHM0BFEH2UjUcJrWiMWtxQuW6Phbqulo7JwjmymEMeIkeK\
nf+FtkE9mrq+E8K40/thrjx13/ZcJD1+3dcp+ZuzVJ2t1E4iGKCx79IZFsysKiuf+\
n+E0i6iGvvI6UcbCZxVxQj2TPlJkFuoX5kDgClIX9Dr9y6aJ4SCh+GRhvHl+DTaz0\
nnCvghachHztIeztRDqxWApj0Ozs93dSelrviMXDr8fgyEAGg7YIhgui0aZBsWCen\
nAgMBAAGjgdUwgdIwgbAGAlUdIwSBqDCBpYAUNpxlPc6cGA7KqEwHMMHBTZMA7lSh\
ngYmkgYYwgYMxGTAXBgNVBAMTEFRlc3QgQ0EgUlRlYiBLZXkxZzAVBgNVBAsTDlBs\
nYXRmb3JtIExiYWZzMRQwEgYDVQKEExFSYWNrc3BhY2UgSG9zdGluZzEUMBIGAlUE\
nBxMLU2FuIEFudG9uaW8xZDjAMBgNVBAGTBVRleGFzMRQwEgYDVQQGEwJVU4IBATAd\
nBgNVHQ4EFgQUlueOfsJZZOHwJHZwBy6u0swnpccwDQYJKoZIhvcNAQEFBQADggEB\
nAFNuqSVUaotUJoWdv4z7Kbi6JFpTjDht5ORw4BdVYlRD4h9DACAfzPrPV2ym/Osp\
nhNMdZq6msZku7MdOSQVhdeGWrSNk3M8O9Hg7cVzPNXOF3iNoo3irQ5tURut44xs4\
nWw5YWQqS9WYUY5snD8tm7YlRQTPfhg+678xIq/zWCv/u+FSnfVvlnlhLVQkEeG/Y\
nghluMaTIPUKTGEjIAGtpGP7wwIcXptR/HyfzhTUSTaWc1Ef7zoKT9LL5z3IVlhC2\njVWz
+RwYs98LjMuksJFoHqRfWyYhCIym0jb6GTwaEmpXajc+d7OLNQdnoEGoUYGP\nYjtfkRYG265ESMA
+Kww4Xy8=\n-----END CERTIFICATE-----\n",
  "enabled": true,
  "secureTrafficOnly": false,
  "privatekey": "-----BEGIN RSA PRIVATE KEY-----\n
nMIIEpAIBAAKCAQEAWIudSMpRZx7TS0/AtDVX3DgXwLD9g+XrNaoazlhwhpYALgzJ\
nLAbAnOxT6OT0gTpkPus/B7QhW6y6Auf2cdBeW3lXoIwPsSoyNhxgErGBxzNARRB9\
nlIiHCAlojFrcULluj4W6rpa0ycI5soDBJiJHin/hbZBPZq6vhPCuNP7Ya48Zd/2X\
nCQ9ft3XKfmbSlSdrdROIhigse/SGRbMrCorn/vhNIuohr7yOlHG3GcVcUI9k6ZSZ\
nBbqF+ZA4ApSF/Q6/cumieEgofhkYbx5fg02s9Jwr4IwnIR2bSHs7UQ6sVgKYzjs7\
nPd3Unpa74jFw6/H6shABO2CIYLotGmQbFgnpWIDAQABAoIBAQCBQC+PCiClJHNv\
ntUzfeCA5ZR4F9JbxHdRTUnxEbOB8UWotckQfTScoAvj4yvdQ42DrCZxj/UodvFOs\
nPufZvlp9lbiZlalugWjE+p8n5+2hIaegoTyHoWZKBfxak0myj5KYfHZvKlbmv1ML\
nXV4TwEVRfAIG+v87QTY/UUxuF5vR+BpKIbgUJLfPUFFvJUdl84qsJ44pToxaYUd/\
nh5YAGC00U4ay1KVSAUnTkPNZ0lPG/rWU6w6WcTvNRLMd8DzFLTKLogQfHhBExAF\
n+sXPwjWSzbBRP107fHqq96QqH4VfiY/7w9W+sDKQyV6U1l7OSXs6aZ4f+lq4rJTI\
nlFG96YiBAoGBA01tiH0h1oWDBYfJB3KJf6CQQsDGwtHo/DEgznFVP4XwEVbZ98Ha\
nBfBCn3sAybbaikyCV1Hwj7kfHmZPDHbrCUSFX7quu/2zPK+w03lZKXSyu4YsguSa\
nRedInN33PpdnlPhLyQdWSu5sVHJDF6xn22vlyxeILH3ooLg2WOFMPmVAoGBAM+b\
```



```
nUG/a7iyfpaQKYyFAsXz6SeFaDY+ZYeX45L112H8Pu+Ie/qzon+bzLB9FIH8GP6+\
nQpQgmm/p37U2gD1zChUv7iW6OfQBKk9rWvMpfRF6d7YHquElejhizfTZ+ntBV/VY\
ndOYEczxhrdW7keLpatYaaWUy/VboRZmlz/9JGqVLAoGAHfqNmFc0cgk4IowEj7a3\
ntTNh6ltub/i+FynwRykfazcdYXaeLPDtfQe8gVh5H8h6W+y9P9BjJVnDVVrXlRAn\
nbiJlEupLPF5sVdapW8ohTOXgfbGTGXBNuUW+4Nv+IDno+mz/RhjkPYHpnM0I7c/5\
ntGzOZsC/2hjNgT8I0+MWav0CgYEAuULdJeQVlKalI6HtW2GnlurRVJ49H+LQkY6e\
nW3+cw2jo9LI0CMWSphNvNrN3wIMp/vHj0fHCP0pSAPDvIWbuQXfzKaGko7UCf7rK\
nf6GvZRCHkV4IREBA97j8bMvThxClMNqFfU0rFZyXP+0MOyhFYertswrgQ6T+Fi\
n2mnvKD8CgYAmJHP3NTDRMoMRYAzonJ6nEaGUbAgNmivTaUWMe0+leCvAdwD89gzC\
nTKbm3eDUg/6Va3X6ANh3wsfIOe4RXXxcbcFDk9R4zO2M5gfLSjYm5Q87EBZ2hrdj\
nM2gLI7dt6thx0J8lR8xRHBEMrVBdgp0g1gQzo5dAV88/kpkZVps8Q==\n-----END RSA
PRIVATE KEY-----\n",
    "intermediateCertificate": "-----BEGIN CERTIFICATE-----\
nMIIDtTCCAp2gAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgZEMBcGA1UEAxMQVGvZ\
ndCBDQSBTVHVhIEtleTEXMBUGA1UECXMouUGxhdGZvcu0gTGJhYXNjYAYBNVBAOT\
nEVJhY2tzcGFjZSBIb3N0aW5nMRQwEgYDVQHEwtTYW4gQW50b25pbzEOMAwGA1UE\
nCBMFVGV4YXNjYAYBNVBAOTAlVTMB4XDTEyMDEwOTE5NDU0OVoXDTE0MDEwODE5\
nNDU0OVoYGMxGTAXBgNVBAMTEFRlc3QgQ0EgU1RlYiBLZXkxZmZAVBgNVBASTDlBs\
nYXRmb3JtIEExYWZzMRowGAYDVQQKEzFSYWNrc3BhY2UgSG9zdGluZzEUMBIGA1UE\
nBxMLU2FuIEFudG9uaW8xDjAMBgNVBAgTBVRleGFzMzQwCQYDVQGEwJVUzCCASIw\
nDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANh55lwTVwQvNoEZjq1zGdYz9jA\
nXXdjizn8AJhJHLOAallfPtvCfTEgKanhdoz5FnhQE8HbDAop/KNS1lN2UMvdl5f\
nZNLTSjJrNtedqxQwxN/i3bpyBxNvejUH2NjVlmmj+5CJYwCzWalvI/gLPq/A3as\
nO2EQqtF3U8unRgn0zXLRdYxV9MrUzNAMdipPNvNrsVdrCgA42rgF/8KsyRvQfJCX\
nfN7PGCfrsC3YaUvhyraWxNnXIzMYTNa9wEeBZLUw8SlEtpalZsvui+TPXu3USNZ\
nVnWH8Lb6ENlnoX0VBwo62fjOG3JzhNkoJawi3bRqyDdINovafr7iPrs/T8CAwEA\
nAaMyMDAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUNpx1Pc6cGA7KqEwHMHb\
nTZMA7lQwDQYJKoZIhvcNAQEFBQADggEBAMoRgh3iTG3t317viLkoY+lnMHUgHuR7\
nb3mn9MidJKyYVewe6hCDIN6WY4fUoJmMW9wFJWJIo0hRMNHL3n3tq8HP2j20Mxy8\
nacPdFGZJa+jBw72CrIGdobKaFduIlIEDBAlpNdZIJ+EulrtqrMesnIt92WaypIS\
n8JycbIgDMciyC0ENHEk8UWlC6429c7OZAsplMTbHME/1R4btXjkdfrYZJjdJ2yL2\
n8cjZDUDMCPTdW/ycP07Gkq30RB5tACB5aZdaCn2YaKc8FsEdhff4X7xEOfOEHWEq\
nSRxADDj8Lx1MT6QpR07hCiDyHfTctbqzI0igjX63Oh7xXSa0f+JVta8=\n-----END
CERTIFICATE-----\n",
    "securePort": 443}
```

The examples to update the load balancers SSL termination attribute request are shown below:


#### Example 4.84. Update Load Balancing SSL Termination Attribute Request: XML

```
<?xml version="1.0" encoding="UTF-8"?>

<sslTermination xmlns="http://docs.openstack.org/loadbalancers/api/v1.0"
  enabled="true" securePort="443" secureTrafficOnly="true"/>
```

#### Example 4.85. Update Load Balancing SSL Termination Attribute Request: JSON

```
{
  "enabled": "true",
  "securePort": 443,
  "secureTrafficOnly": "true"
}
```



This operation does not require a request body.

There is no example for the Delete Load Balancing SSL Termination Request.

This operation does not require a request body.

This operation does not return a response body.