# Rackspace Cloud Load Balancers API 2.0 (Early Access)

*Last updated:* February 22, 2016

Use the following links to go directly to user and reference information for using the Rackspace Cloud Load Balancers service REST API.

**Deleted:** jump

- Getting Started Guide
- Developer Guide
- API Reference
- Release Notes

**Deleted:** r

## About the API

The Rackspace Cloud Load Balancers service enables customers to quickly load-balance multiple cloud servers or external servers for optimal resource utilization by using a simple Representational State Transfer (REST) web service interface.

**Deleted:** Cloud

**Deleted:** Servers

**Deleted:** through

**Deleted:** The Cloud Load Balancers FAQ in the Knowledge Center provides more information about the service.¶
¶

## Additional resources

We welcome feedback, comments, and bug reports. Visit the Rackspace customer portal at https://feedback.rackspace.com/.

Use the following links to learn more about the Rackspace Cloud Load Balancers service and API.

**Deleted:** below

- For general information about Cloud Load Balancers, see the Cloud Load Balancers FAQ in the Rackspace How-To site.

**Comment [KH1]:** Update to new link to FAQ: https://support.rackspace.com/how-to/cloud-load-balancers-faq/

**Deleted:** Knowledge Center

- For information about starting to use Cloud Load Balancers and Cloud Servers, see the Getting Started Guide.
- To learn about using Cloud Load Balancers from the Rackspace Cloud Control panel, see Configure a load balancer in the Rackspace How-To site.
- To learn about using Rackspace Cloud SDKs, see Software Development Kits & Tools.
- To get information about other Rackspace Cloud services APIs, see the API Developer documentation.

**Deleted:** getting started using

**Deleted:** refer

**Deleted:** to

**Deleted:** service

**Comment [KH2]:** Update to the new link: https://support.rackspace.com/how-to/configure-a-load-balancer/

**Deleted:** the Knowledge Center article

**Deleted:** ing

**Deleted:** Load

**Deleted:** B

**Deleted:** Docs

# Developer Guide

This guide is intended to assist software developers who want to develop applications by using the REST application programming interface (API) for the Rackspace Cloud Load Balancers service.

To use the information provided here, you should have a general understanding of the Load Balancers service and have access to an installation of the Cloud Load Balancers service. You should also be familiar with the following technologies:

- RESTful web services
- HTTP/1.1
- JSON serialization format

**Comment [KH3]:** Looks italic online, but shouldn't be.

**Comment [KH4]:** Looks italic online, but shouldn't be.

# Concepts

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

**Deleted:** :

## Load balancer

A *load balancer* is a logical device that 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.

**Formatted:** Font: Italic

**Deleted:** which

## Listener

A *listener* is an object that contains data that pertains to the “listening” port and the protocol that the load balancer accepts incoming traffic on, otherwise known as the *front end* of the configuration. The front-end configuration also determines and contains the back-end data, such as pools and their members to which incoming traffic is directed.

**Formatted:** Font: Italic

**Deleted:** containing

**Deleted:** ing

**Deleted:** ,

**Deleted:** “*frontend*

**Formatted:** Font: Italic

**Deleted:** ”

**Deleted:** its

# Virtual IP

A *virtual IP* (VIP) 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.

**Formatted:** Font: Italic

# Member

A *member* is a back-end device, such as a server, that provides a service on a specified IP address and port.

**Formatted:** Font: Italic

**Deleted:** providing

# Pool

A *pool* is a logical set of devices, such as web servers, that you group together to receive and process traffic. Instead of sending client traffic to the destination IP address specified in the client request, the system sends the request to any of the servers that are members of that pool.

**Formatted:** Font: Italic

# Health monitor

A *health monitor* is a configurable feature of each load balancer. It is used to determine whether a back-end member is usable for processing a request. The load balancing service currently supports *active health monitoring*.

**Formatted:** Font: Italic

**Deleted:** or not

**Formatted:** Font: Italic

Active health monitoring is a technique that uses synthetic transactions executed at periodic intervals to determine the condition of a member. 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 a member is suitable for handling traffic. Active health monitoring is not applied by default and must be enabled per load balancer.

**Deleted:** Active health monitor¶

**Deleted:** or not

The active health monitor can use one of the following types of probes:

**Deleted:** four

- PING
- HTTP

- HTTPS
- TCP

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

**Deleted:** servicing

## 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.

**Formatted:** Font: Italic

# General API information

The information in this section is relevant to all operations of the API. For details about specific operations, see the API Reference.

**Deleted:** Ideas explained here

**Deleted:** are

**Comment [KH5]:** Make this a link to the "API Reference" section.

**Deleted:** Operations chapter

**Deleted:** for details about specific operations

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

**Deleted:** ReSTful

**Deleted:** products

**Deleted:** in the Rackspace Cloud suite

**Deleted:** load

**Deleted:** balancing

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

## Authentication

Every REST request against the Load Balancers service requires the inclusion of a specific authorization token, supplied in the `X-Auth-Token` HTTP header of each API request. You get a token by submitting an authentication request with valid account credentials to the following Rackspace Cloud Identity API service endpoint:

**Deleted:** ReST

**Deleted:** Service

```
https://identity.api.rackspacecloud.com/v2.0
```

For details, see the following information:

- Authenticate to the Rackspace Cloud
- Rackspace Cloud Identity API developer documentation

# Service access endpoints

**Deleted:** and

The Load Balancers service is a regionalized service. It allows the user to select a region into which a load balancer is to be provisioned.

**Deleted:** load balancing

**Deleted:** caller

To determine which region to operate against, select an endpoint from the following table.

**Deleted:** below

**Tip:** To help you decide which regionalized endpoint to use, read about special considerations for choosing a region.

**Comment [KH6]:** Update link to https://support.rackspace.com/how-to/about-regions/.

## Regionalized service endpoints

| Region | Endpoint |
|---|---|
| Northern Virginia (IAD) | https://iad.networks.api.rackspacecloud.com/v2.0/lbaas/ |

**Note:** The service catalog returned in the authentication response specifies the correct service access endpoint for your account to use for accessing Cloud Load Balancers. Use the service `type (rax:load-balancer-v2)` to locate the correct endpoint in the service catalog.

When making a Cloud Load Balancers API call, place the endpoint at the beginning of the request URL. For example, the URL to use to create a load balancer is `https://iad.networks.api.rackspacecloud.com/v2.0/lbaas/loadbalancers`. Note that the `v2.0` component in the URL indicates that you are using version 2.0 of the Cloud Load Balancers API.

**Deleted:** C

**Deleted:** :

# Request and response types

The Rackspace Cloud Load Balancers API supports the JSON data serialization format. The request format is specified by using the `Content-Type` header and is required for operations that have a request body. The response format can be specified in requests either by using the `Accept` header or by adding a `.json` extension to the request URI. If no response format is specified, JSON is the default.

| Format | Accept header | Query extension | Default |
|---|---|---|---|
| JSON | application/json | .json | Yes |

# 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. Rate limits are processed via the Repose service.

**Note:** You can submit a request to Rackspace Support for an increase in load balancer limits. Each request must be approved before limits can be modified. Limits can be increased only up to the maximum limit (such as 50 nodes per load balancer).

## Rate limits

Rate limits are specified in 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 `^/v2.0/lbaas` matches the `/v2.0/lbaas` portion  of the following URI: `https://iad.networks.api.rackspacecloud.com/v2.0/lbaas/`.

The following table shows the default rate limits for each method.

**Deleted:** either

**Deleted: Header**

**Deleted: Extension**

**Formatted:** Font: Italic

**Formatted:** Font: Italic

**Comment [KH7]:** The section that follows "Rate limits" and deals with absolute limits (I think) is titled "Quotas." If quotas are absolute limits, then I would suggest revising this sentence to say "Absolute limits, also called *quotas*, are fixed." That way the connection between the terms is introduced.

If they are not the same, then revise this intro to talk about quotas instead of absolute limits.

**Comment [KH8]:** I would move this to somewhere in the "Rate limits" section.

**Deleted:** 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.

**Deleted:** We specify r

**Deleted:** terms of

**Deleted:** ‘

**Deleted:** ’

**Deleted:** would

**Deleted:** portion

### Default rate limits

| Method | URI | Regex | Default limit |
|---|---|---|---|
| GET | `/v2.0/*` | `^/2.0/.*` | 5 per second |
| GET | `/v2.0/*` | `^/2.0/.*` | 100 per minute |
| POST | `/v2.0/*` | `^/2.0/.*` | 2 per second |
| POST | `/v2.0/*` | `^/2.0/.*` | 25 per minute |
| PUT | `/v2.0/*` | `^/2.0/.*` | 5 per second |
| PUT | `/v2.0/*` | `^/2.0/.*` | 50 per minute |
| DELETE | `/v2.0/*` | `^/2.0/.*` | 2 per second |
| DELETE | `/v2.0/*` | `^/2.0/.*` | 50 per minute |

Rate limits are applied in order relative to the method, going from least to most specific. For example, although the threshold for submitting a POST request to `/v2.0/*` is 25 per minute, you cannot submit a POST request to `/v2.0/*` more than 2 times per second because the rate limit for any POST request is 2 per second. If you exceed the thresholds established for your account, a 413 (Rate Control) HTTP response is returned with a `Retry-After` header to notify the client when it can attempt to try again.

To find your account's settings for these rate limits, see Determine limits programmatically.

**Deleted:** Verb

**Deleted:** RegEx

**Formatted:** Code

**Deleted:** /

**Formatted:** Code

**Deleted:** /

**Formatted:** Code

**Deleted:** /

**Formatted:** Code

**Deleted:** /

**Formatted:** Code

**Deleted:** /

**Formatted:** Code

**Deleted:** /

**Formatted:** Code

**Deleted:** /

**Formatted:** Code

**Deleted:** /

**Deleted:** verb

**Deleted:**

**Formatted:** Code Char, Font: (Default) Times New Roman, Font color: Auto

**Deleted:** one

**Formatted:** Code Char, Font: (Default) Times New Roman, Font color: Auto

**Deleted:** In the event

**Deleted:** they

**Deleted:** 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).¶

**Deleted:** S

**Deleted:** to find your current account settings for these limits

## Quotas

Quotas 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, because it is applied per batch delete API request.

**Comment [KH9]:** Are quotas the same as absolute limits? If so, I would add here: "Quotas, also called absolute limits, specify…"

**Deleted:** since

The system applies default values for each quota, as shown in the following table.

**Deleted:** 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 |
|---|---|---|
| loadbalancer | Total number of load balancers that can be added to a Cloud account | 10 |
| listener | Total number of listeners that can be added to a load balancer | 20 |
| pool | Total number of pools that can be added to a listener | 20 |
| member | Total number of members that can be added to a load balancer | 75 |
| healthmonitor | Total number of health monitors that can be added to a pool | 20 |

**Deleted:** .

**Deleted:** .

**Deleted:** .

**Deleted:** .

**Deleted:** .

To find your account's settings for these quotas, see Determine limits programmatically.

**Deleted:** S

**Deleted:** to find your current account settings for these quotas

## Determine limits programmatically

Applications can programmatically determine current limits for an account by using the following URI:

**Deleted:** rate

**Deleted:** s

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

Error response codes: loadbalancerFault (400, 500), serviceUnavailable (503), unauthorized (401), badRequest (400), overLimit (413)

**Deleted:** Response

**Deleted:** Code

**Deleted:** (s)

This operation does not require a request body. Following is an example response.

**Example: List rate limits, JSON response**

```
{
  "limits" : {
    "rate" : {
      "values": [
        {
          "uri" : "/v2.0/*",
          "regex" : "^/2.0/.*",
          "limit" : [
            {
              "verb" : "GET",
              "value" : 600000,
              "remaining" : 426852,
              "unit" : "HOUR",
              "next-available" : "2011-02-22T19:32:43.835Z"
            }
          ]
        }
      ]
    }
  }
}
```

# Date and time format

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

```
YYYY-MM-DD'T'hh:mm:ssZ
```

For example,

May 19, 2016 at 8:07:08 AM, GMT-5 would have the following format:

```
2016-05-19T08:07:08-05:00
```

**Deleted:** :

**Deleted:** /

**Deleted:** ¶
*Load Balancer service date and time format*

**Deleted:** yyyy

**Deleted:** dd'T'HH

**Deleted:** ¶

**Moved down [1]:** See the table below for a description of the date/time format codes.¶

**Deleted:** th

**Deleted:** 2011

**Deleted:** 2011

The following table describes the date and time format codes.

**Moved (insertion) [1]**

**Deleted:** See t

**Deleted:** below for a description of

**Deleted:** /

## Date and time format codes

| | |
|---|---|
| YYYY | Four-digit year |
| MM | Two-digit month |
| DD | Two-digit day |
| T | Separator for date and time |
| hh | Two-digit hour (00-23) |
| mm | Two-digit minute |
| ss | Two-digit second |
| Z | RFC 822 time zone (offset from GMT). If Z is not replaced with the offset from GMT, it indicates a 00:00 offset. |

**Deleted:** yyyy

**Deleted:** /

**Deleted:** HH

**Deleted:** 8601

**Deleted:** timezone

# API behavior and statuses

The Load Balancer API is considered to be asynchronous because mutable operations (that is, POST, PUT, and DELETE) are often queued and then handled accordingly. All successful asynchronous API operations have a normal response code of 202.

The load balancer `status` attribute is closely linked to mutable operations because it is updated depending on the operation or the state of the load balancer. The following table lists the possible load balancer statuses.

**Deleted:** , errors, and suggestions

**Deleted:** This section describes the behavior of the API, errors, and provides suggestions for working with the API.¶
**API behavior¶**

**Deleted:** LBaaS

**Deleted:** /

**Deleted:** /

**Deleted:** up

**Deleted:** as

**Deleted:** and/

**Deleted:** Please see the

**Deleted:** below for a list of

**Deleted:** Please note that any load balancer can have at most one operation requested at a time. Thus, issuing parallel mutable requests per load balancer is not allowed and only one of the requests will be accepted should concurrent mutable requests be issued. Issuing concurrent non-mutable requests (that is, GET) is allowed.¶

### LOAD BALANCER STATUSES

| Name | Description |
|---|---|
| ACTIVE | The load balancer is active. |
| ERROR | The load balancer is in an error state. |
| PENDING_CREATE | The load balancer has a create action pending. |
| PENDING_DELETE | The load balancer has a deletion pending. |
| PENDING_UPDATE | The load balancer has an update action pending. |

**Note:** There is not currently a DELETED status, which means that if you use the API to request details about a DELETED object, you will receive a 404 Not Found response.

Any load balancer can have only one mutable operation requested at a time. If concurrent mutable requests are issued for a load balancer, only one of the requests is accepted. Issuing concurrent non-mutable requests (that is, GET) is allowed.

To determine when a mutable operation is complete, poll the load balancer details once every 5-10 seconds to determine if the load balancer has changed back to an `ACTIVE` status. After the load balancer is back in the `ACTIVE` status, another mutable operation can be accepted.

**Comment [KH10]:** Most table and example titles are not formatted as headings, but just as bold regular text. Apparently tagging a heading as h5 makes it all caps. I would reformat this to just use bold regular text.

**Deleted:** on

**Deleted:** ¶
**API errors**¶
Error responses contain a body with the validation error, code, and specific message related to the error. Use this information to diagnose what went wrong during the API operation.

**Deleted: Suggestions**¶
The most common issue an API user will come across is determining

**Deleted:** action

**Deleted:** . It is suggested that the API user

**Deleted:** (

**Deleted:** per

**Deleted:** is recommended) in order

**Deleted:** Once

# Listener protocols

The following table describes the supported listener protocols.

### Listener protocols

**Deleted:** Table:

| Name | Description |
|---|---|
| TCP | Basic TCP. This protocol can be used as the protocol for any TCP-based protocol. |
| HTTP | This protocol load balances HTTP traffic and passes it to the members. |
| HTTPS | HTTPS passthrough. This protocol does not inspect the packet payloads, but just passes them to the pool of members. |
| TERMINATED_HTTPS | This protocol terminates HTTPS at the load balancer. It decrypts and passes unencrypted data to the pool of members. |

**Deleted:** will

**Deleted:** do any inspection of

The following table describes the supported protocols for pools.

### Supported protocols for pools

**Deleted:** Table:

| Name | Description |
|---|---|
| TCP | Basic L4 protocol |
| HTTP | Higher layer protocol based on TCP |
| HTTPS | Encrypted protocol based on TCP |

**Deleted:** .

**Deleted:** .

**Deleted:** .

# Role-based access control (RBAC)

Role-based access control (RBAC) restricts access to the capabilities of Rackspace Cloud services, including the Cloud Load Balancers API, to authorized users only. RBAC enables Rackspace Cloud customers to specify which account users of their Cloud account have access to which Cloud Load Balancers API service capabilities, based on roles defined by Rackspace (see Roles available for Cloud Load Balancers). The permissions to perform certain operations in Cloud Load Balancers API – create, read, update, delete – are assigned to specific roles. The account owner user assigns these roles, either global (multiproduct) or product-specific (for example Cloud Load Balancers), to account users.

**Deleted:** product roles and capabilities

## Assign roles to account users

The account owner (identity:user-admin) can create account users on the account and then assign roles to those users. The roles grant the account users specific permissions for accessing the capabilities of the Cloud Load Balancers service. Each account has only one account owner, and that role is assigned by default to any Rackspace Cloud account when the account is created.

See the *Cloud Identity Client Developer Guide* for information about how to perform the following tasks:

**Deleted:** on

- Add account users
- Add role to user
- Delete global role from user

The account owner (identity:user-admin) role cannot hold any additional roles because it already has full access to all capabilities.

## Roles available for Cloud Load Balancers

Three roles (observer, creator, and admin) can be used to access the Cloud Load Balancers API specifically. The following table describes these roles and their permissions.

**Cloud Load Balancers product roles and capabilities**

| Role name | Role permissions |
|---|---|
| lbaas:admin | This role provides create, read, update, and delete permissions in Cloud Load Balancers, where access is granted. |
| lbaas:creator | This role provides create, read and update permissions in Cloud Load Balancers, where access is granted. |
| lbaas:observer | This role provides read permission in Cloud Load Balancers, where access is granted. |

**Deleted:** Create

**Deleted:** Read

**Deleted:** Update

**Deleted:** Delete

**Deleted:** Create

**Deleted:** Read

**Deleted:** Update

**Deleted:** Read

Additionally, two multiproduct roles apply to all products. Users with multiproduct roles inherit access to future products when those products become RBAC-enabled. The following table describes these roles and their permissions.

**Deleted:** section

##### MULTIPRODUCT GLOBAL ROLES AND PERMISSIONS

**Comment [KH11]:** Most table and example titles are not formatted as headings, but just as bold regular text (see the title of the preceding table). Apparently tagging a heading as h5 makes it all caps. I would reformat this to just use bold regular text.

| Role name | Role permissions |
|---|---|
| admin | This role provides create, read, update, and delete permissions in all products, where access is granted. |
| observer | This role provides read permission in all products, where access is granted. |

**Deleted: Name**

**Deleted: Permissions**

**Deleted:** Create

**Deleted:** Read

**Deleted:** Update

**Deleted:** Delete

**Deleted:** Read

## Resolve conflicts between RBAC multiproduct versus custom (product-specific) roles

The account owner can set roles for both multiproduct and Cloud Load Balancers scope, and it is important to understand how any potential conflicts among these roles are resolved. When two roles appear to conflict, the role that provides the more extensive permissions takes precedence. Therefore, admin roles take precedence over observer and creator roles, because admin roles provide more permissions.

The following table shows two examples of how potential conflicts between user roles in the Control Panel are resolved.

**Deleted:** :

[Add a table title]

| Permission configuration | View of permission in the Control Panel | Can the user perform product admin functions in the Control Panel? |
|---|---|---|
| User is assigned the following roles: multiproduct observer and Cloud Load Balancers admin | Appears that the user has only the multiproduct observer role | Yes, for Cloud Load Balancers only. The user has the observer role for the rest of the products. |
| User is assigned the following roles: multiproduct admin and Cloud Load Balancers observer | Appears that the user has only the multiproduct admin role | Yes, for all of the products. The Cloud Load Balancers observer role is ignored. |

## RBAC permissions cross-reference to Cloud Load Balancers API operations

API operations for Cloud Load Balancers may or may not be available to all roles. To see which operations are permitted to invoke which calls, see the How-To article Permissions Matrix for Role-Based Access Control (RBAC).

**Deleted:** please review

**Deleted:** Knowledge Center

**Comment [KH12]:** Update to new URL: https://support.rackspace.com/how-to/permissions-matrix-for-role-based-access-control-rbac/

# API Reference

Learn about the available Cloud Load Balancers API resources and methods and see request and response examples.

You can use the Cloud Load Balancers API operations to interact directly with the Load Balancer service. You can also perform Load Balancer operations by using the Rackspace command-line interface (CLI), the SDK, or the Cloud Control Panel.

# Load balancers

You can configure all documented features of the load balancer when you create it by providing the additional elements or attributes in the request.

The `vip_subnet_id` specified for the load balancer determines the type of IP address and what network it is allocated on.

All load balancers also have a `status` attribute that shows the current configuration status of the device. This status is immutable by the user and is updated automatically based on state changes within the service. The following table describes the possible statuses.

### Load balancer statuses

| Name | Description |
|---|---|
| `PENDING_CREATE` | The load balancer is being provisioned for the first time and configuration is being applied to bring the service online. The service cannot yet serve incoming requests. |
| `ACTIVE` | The load balancer is configured properly and ready to serve traffic to incoming requests via the configured virtual IP addresses. |
| `PENDING_UPDATE` | The load balancer is online but configuration changes are being applied to update the service based on a previous request. |
| `PENDING_DELETE` | The load balancer is online but configuration changes are being applied to begin deletion of the service based on a previous request. |

**Comment [KH13]:** I'm assuming this is a reference to the Rack CLI, and if I am wrong, ignore the edits and just ensure that whatever is named here is the correct name. If I am right, well, it doesn't look to me like Rack CLI supports Load Balancers yet.

**Deleted:** Command

**Deleted:** L

**Deleted:** Client

**Deleted:** from

**Deleted:** panel

**Deleted:** Users

**Deleted:** at creation time

**Deleted:** simply

**Deleted:** This document provides an overview of all the features the load balancing service supports.

**Formatted:** Code Char, Font: (Default) Times New Roman, Font color: Auto

**Deleted:** caller

**Deleted:** <#>When a load balancer is first created, it is placed into a `PENDING_CREATE` status while the configuration is being generated and applied based on the request. Once the configuration is applied and finalized, it is in an `ACTIVE` status. In the event of a configuration change or update, the status of the load balancer changes to `PENDING_UPDATE` to signify configuration changes are in progress but are not yet been finalized. Load balancers in a `SUSPENDED` status are configured to reject traffic and does not forward requests to back-end members.¶

**Deleted:** Table 4.1.

**Formatted:** Font: Bold

**Deleted:** `ACTIVE` ...

**Deleted:** Load

**Deleted:** Load

**Deleted:** Load

| Name | Description |
|---|---|
| SUSPENDED | The load balancer was 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. |

# List load balancers

```
GET /v2.0/lbaas/loadbalancers
```

This operation lists all load balancers that are associated with your tenant account.

This operation returns a list, which might be empty. Each element in the list is a load balancer that can contain the following attributes:

- id
- tenant_id
- name
- description
- vip_subnet_id
- vip_address
- admin_state_up
- listeners
- provisioning_status
- operating_status

**Deleted:** Load

**Deleted:** has been

**Formatted:** Code Char, Font: +Body (Calibri), 11 pt

**Deleted:** The update operation allows the caller to change one or more of the following attributes:¶
name¶
description¶
admin_state_up¶

**Deleted:** Lists

**Comment [KH14]:** I'm confused about this list. If these are the possible attributes that could be returned, shouldn't they just all be listed in the table of body parameter for the response? Some of these are listed there, and some aren't. I don't think that you need this list, if you update that table to include all the possible response parameters.

The following table shows the possible response codes for this operation.

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

## Request

This operation does not accept a request body.

## Response

The following table shows the body parameters for the response.

| Parameter | Style | Type | Description |
|---|---|---|---|
| loadbalancers | plain | xsd:string | A load balancers object. |
| id | plain | csapi:uuid | The UUID for the load balancer. |
| name | plain | xsd:string | The load balancer name. |
| description | plain | xsd:string | The load balancer description. |
| vip_address | plain | xsd:ip | The virtual IP (VIP) address |
| status | plain | xsd:string | The status of the load balancer, which indicates whether the load balancer |

**Comment [KH15]:** For all of the method sections, consider moving the response codes table to the "Response" section. It seems like it would be better placed there.

**Deleted:** This

**Deleted:** :

**Deleted: Code**

**Deleted:** Request

**Deleted:** has

**Deleted:** This

**Deleted:** :

**Deleted:** loadbalancers

**Deleted:** of the VIP

**Deleted:** .

**Deleted:** I

| Parameter | Style | Type | Description |
|---|---|---|---|
| | | | is operational. |
| admin_state_up | plain | xsd:boolean | The administrative state of the load balancer, which is up (true) or down (false). |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the VIP address. Only administrative users can specify a tenant UUID other than their own. |

Example: List load balancers, JSON response

```
{
  "loadbalancers": [
    {
      "id": "3b98602c-3cfe-4f91-bfa4-c3a11c9e7fe0",
      "name": "Example LB",
      "description": "A very simple example load balancer.",
      "tenant_id": "783b31af-6635-48b2-a807-091d9973e3a9",
      "admin_state_up": true,
      "status": "ACTIVE"
    },
    {
      "id": "c617c538-daa5-4ead-be88-59521d8745a7",
      "name": "Example LB",
      "description": "A very simple example load balancer.",
      "tenant_id": "783b31af-6635-48b2-a807-091d9973e3a9",
      "admin_state_up": true,
      "status": "ACTIVE"
    }
  ]
}
```

# Create a load balancer

```
POST /v2.0/lbaas/loadbalancers
```

This operation provisions a new load balancer based on the configuration defined in the request.

**Comment [KH16]:** In the online doc, the link to this section is actually going to the “Create a load balancer” section in the Getting Started Guide part of the doc. I guess we have to be careful to create distinct permalinks for headings that are very similar.

**Deleted:** object

After the request is validated and progress has started on the provisioning process, a response is returned. The response contains a unique identifier for the load balancer and the status of provisioning the load balancer.

The `provisioning_status` of the load balancer in the response can have one of the following values: `ACTIVE`,`PENDING_CREATE`, or `ERROR`.

If the status is `PENDING_CREATE`, you can view the progress of the provisioning operation by performing a GET operation on `/lbaas/loadbalancers/<loadbalancer_id>`. When the status of the load balancer changes to `ACTIVE`, the load balancer was successfully provisioned and is ready to handle traffic.

If the request cannot be fulfilled because of insufficient or invalid data, the service returns the HTTP `Bad Request (400)` response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

Users with an administrative role can create load balancers on behalf of other tenants by specifying a `tenant_id` attribute different than their own.

## Example: Create a load balancer

- `tenant_id`. only required if the caller has an administrative role and wants to create a load balancer for another tenant.
- `vip_subnet_id`. The network on which to allocate the VIP address for the load balancer. A tenant can only create load balancer VIPs on networks that are authorized by the policy, such as her own networks or shared or provider networks.

Some attributes receive default values if you omit them from the request:

- `admin_state_up`. Default is `true`.
- `name`. Default is an empty string.
- `description`. Default is an empty string.

A user can supply a `vip_address` field if she owns the subnet on which the load balancer's VIP will be created. If a `vip_address` is omitted from the payload, the LBaaS service allocates a VIP address from the subnet of the load balancer VIP.

**Deleted:** object

**Deleted:** object

**Deleted:** the caller

**Deleted:** operational

**Deleted:** for

**Deleted:** handling

**Deleted:** due to

**Deleted:** You can configure all documented features of the load balancer at creation time by specifying the additional elements or attributes in the request.¶

**Deleted:** If the request cannot be fulfilled due to insufficient data or data that is not valid, the service returns the HTTP `Bad Request(400)` response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.¶
You can configure all documented features of the load balancer at creation time by specifying the additional elements or attributes in the request.¶
Users with an administrative role can create load balancers on behalf of other tenants by specifying a `tenant_id` attribute that is different than their own.¶
¶

**Comment [KH17]:** I would simply ensure that all of the relevant information is in the request parameters table, and delete all this. The duplication is not necessary.

The following table shows the possible response codes for this operation.

Deleted: This

Deleted: :

| Response code | Name | Description |
|---|---|---|
| 201 | Created | The request was fulfilled and the new resource was created. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 404 | Not Found | The requested item was not found. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Comment [KH18]:** The intro mentions the possibility of a `Bad Request (400)` response code, but I don't see that listed in this table. Should it be?

**Deleted: Code**

**Deleted:** has been

**Deleted:** resulted in a

**Deleted:** being

**Deleted:** due to

**Deleted:** above

**Deleted:** has

## Request

**Deleted:** Example. Create load balancer: JSON request¶

The following table shows the body parameters for the request.

**Deleted:** This list

**Deleted:** :

**Comment [KH19]:** Add a row to the table for `admin_state_up`.

| Parameter | Style | Type | Description |
|---|---|---|---|
| name (optional) | plain | xsd:string | The load balancer name. The name does not have to be unique. If you omit the name, the default value is an empty string. |
| description (optional) | plain | xsd:string | The load balancer description. If you omit the description, the default value is an empty string. |
| vip_subnet_id | plain | csapi:uuid | The UUID of the subnet on which to allocate the virtual IP (VIP) address. Tenants can create load balancer VIP addresses only on networks that are |

**Deleted:** Does

| Parameter | Style | Type | Description |
|---|---|---|---|
| | | | authorized by the policy, such as their own networks or shared or provider networks. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the VIP address. Only administrative users can specify a tenant UUID other than their own. |
| vip_address (optional) | plain | xsd:ip | The VIP address. If this parameter is omitted from the request, the service allocates a VIP address from the subnet of the load balancer VIP. |
| provider (optional) | plain | xsd:string | The name of the provider. |

**Comment [KH19]:** Add a row to the table for `admin_state_up`.

**Comment [KH20]:** The intro mentioned that this parameter is required only if the user has an administrative role and wants to create a load balancer for another tenant. If that is true, wouldn't that make this an optional parameter?

**Deleted:** of the VIP.

**Comment [KH21]:** What is a provider, in this context?

## Example: Create a load balancer, JSON request

```
{
  "loadbalancer": {
    "name": "loadbalancer1",
    "description": "simple lb",
    "tenant_id": "b7c1a69e88bf4b21a8148f787aef2081",
    "vip_subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
    "vip_address": "10.0.0.4",
    "admin_state_up": true
  }
}
```

# Response

**Moved down [2]:** Example. Create load balancer: JSON response¶

The following table shows the body parameters for the response.

**Deleted:** This

**Deleted:** list

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| loadbalancer | plain | xsd:string | A load balancers object. |

**Comment [KH22]:** Table doesn't list `listeners` or `vip_subnet_status`, both of which are shown in the example. Add them?

**Deleted:** loadbalancers

| Parameter | Style | Type | Description |
|---|---|---|---|
| id | plain | csapi:uuid | The UUID for the load balancer. |
| name | plain | xsd:string | The load balancer name. |
| description | plain | xsd:string | The load balancer description. |
| vip_address | plain | xsd:ip | The virtual IP (VIP) address. |
| status | plain | xsd:string | The status of the load balancer, which indicates whether the load balancer is operational. |
| admin_state_up | plain | xsd:boolean | The administrative state of the load balancer, which is up (true) or down (false). |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the VIP. Only administrative users can specify a tenant UUID other than their own. |

**Comment [KH22]:** Table doesn't list listeners or vip_subnet_status, both of which are shown in the example. Add them?

**Deleted:** of the VIP

**Deleted:** . I

**Comment [KH23]:** The example shows operating_status. If that is the same as this, update this parameter name.

## Example: Create a load balancer, JSON response

**Moved (insertion) [2]**

**Deleted:** .

**Deleted:** :

```
{
  "loadbalancer": {
    "admin_state_up": true,
    "description": "simple lb",
    "id": "a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "listeners": [],
    "name": "loadbalancer1",
    "operating_status": "ONLINE",
    "provisioning_status": "ACTIVE",
    "tenant_id": "b7c1a69e88bf4b21a8148f787aef2081",
    "vip_address": "10.0.0.4",
    "vip_subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2"
  }
}
```

# Show load balancer details

**Comment [KH24]:** In the online doc, the link to this section is actually going to the “Show load balancer details” section in the Getting Started Guide part of the doc.

```
GET /v2.0/lbaas/loadbalancers/{loadbalancer_id}
```

This operation returns the load balancer object identified by `loadbalancer_id`. If the user is not an administrative user and the load balancer object does not belong to the user’s tenant account, the service returns the HTTP `Forbidden (403)` response code.

**Deleted:** a

**Deleted:** her

If this operation succeeds, it returns a load balancer element that can contain the following attributes:

- `id`
- `tenant_id`
- `name`
- `description`
- `vip_subnet_id`
- `vip_address`
- `admin_state_up`
- `provisioning_status`
- `operating_status`

**Comment [KH25]:** Again, I would delete this list and just ensure that all if the attributes/parameters are listed in the response body parameters table.

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 403 | Forbidden | The server understood the request, but won’t fulfill it. |
| 404 | Not Found | The requested item was not found. |

**Deleted: Code**

**Deleted:** Request

**Deleted:** is refusing to

| Response code | Name | Description |
|---|---|---|
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** Code

**Deleted:** due to

**Deleted:** above

**Deleted:** has

## Request

This operation does not accept a request body.

## Response

**Moved down [3]:** Example. Show load balancer details: JSON response¶

The following table shows the body parameters for the response.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| loadbalancer | plain | xsd:string | A load balancers object. |
| id | plain | csapi:uuid | The UUID for the load balancer. |
| name | plain | xsd:string | The load balancer name. |
| description | plain | xsd:string | The load balancer description. |
| vip_address | plain | xsd:ip | The virtual IP (VIP) address. |
| vip_subnet_id | plain | csapi:uuid | The UUID of the subnet on which to allocate the VIP address. |

**Deleted:** loadbalancers

**Deleted:** of the VIP

**Deleted:** virtual IP

**Deleted:** (VIP)

| Parameter | Style | Type | Description |
|---|---|---|---|
| status | plain | xsd:string | The status of the load balancer, which indicates whether the load balancer is operational. |
| admin_state_up | plain | xsd:boolean | The administrative state of the load balancer, which is up (true) or down (false). |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the VIP address. Only administrative users can specify a tenant UUID other than their own. |

**Deleted:** . I

### Example: Show load balancer details, JSON response

**Moved (insertion) [3]**

**Deleted:** .

**Deleted:** :

```
{
 "loadbalancer":{
  "id":"8992a43f-83af-4b49-9afd-c2bfbd82d7d7",
  "name":"Example LB",
  "description":"A very simple example load balancer.",
  "vip_address":"1.2.3.4",
  "vip_subnet_id":"SUBNET_ID",
  "tenant_id":"7725fe12-1c14-4f45-ba8e-44bf01763578",
  "admin_state_up":true,
  "status":"ACTIVE"
 }
}
```

## Update a load balancer

```
PUT /v2.0/lbaas/loadbalancers/{loadbalancer_id}
```

This operation enables you to change one or more of the following load balancer attributes:

**Moved (insertion) [4]**

**Deleted:** The update

- name
- description

- admin_state_up

If the request is validated, the service returns the Accepted (202) response code. Check that the load balancer provisioning_status has changed to ACTIVE to confirm that the update has taken effect. If the load balancer provisioning_status is PENDING_UPDATE, you can poll the load balancer object by using a GET operation to wait for the changes to be applied.

The following table shows the possible response codes for this operation.

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 400 | Bad Request | The request is missing one or more elements, or the values of some elements are invalid. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

## Request

The following table shows the body parameters for the request.

| Parameter | Style | Type | Description |
|---|---|---|---|

**Deleted:** Upon

**Deleted:** successful validation of

**Deleted:** A caller should c

**Deleted:** the caller

**Deleted:** ¶

**Moved up [4]:** The update operation enables you to change one or more of the following load balancer attributes:¶
<#>name¶
<#>description¶
<#>admin_state_up¶

**Deleted:** ¶
<#>name¶
<#>description¶
<#>admin_state_up¶

**Deleted:** This operation returns the updated load balancer object. The provisioning_status value can be ACTIVE,PENDING_UPDATE, or ERROR.¶
This

**Deleted:** :

**Comment [KH26]:** The preceding text mentions the 202 code. Should that be listed in the table?

**Deleted:** Code

**Deleted:** Request

**Deleted:** above

**Deleted:** has

**Moved down [5]:** Example. Update load balancer: JSON request¶

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| loadbalancer | plain | xsd:string | A load balancers object. |
| name (optional) | plain | xsd:string | The load balancer name. The name does not have to be unique. |
| description (optional) | plain | xsd:string | The load balancer description. |
| admin_state_up | plain | xsd:boolean | The administrative state of the load balancer, which is up (true) or down (false). |

### Example: Update a load balancer, JSON request

```
{
  "loadbalancer": {
    "admin_state_up": false,
    "description": "simple lb2",
    "name": "loadbalancer2"
  }
}
```

## Response

The following table shows the body parameters for the response.

| Parameter | Style | Type | Description |
|---|---|---|---|
| loadbalancer | plain | xsd:string | A load balancers object. |
| id | plain | csapi:uuid | The UUID for the load balancer. |
| name | plain | xsd:string | The load balancer name. |

**Deleted:** loadbalancers

**Deleted:** Does

**Moved (insertion) [5]**

**Deleted:** .

**Deleted:** :

**Moved down [6]:** Example. Update load balancer: JSON response¶

**Deleted:** This list

**Deleted:** :

**Comment [KH27]:** The table does not include all of the possible parameters, shown in the response example. I would image that this table should be the same as the one for Create load balancer, so perhaps you could just copy that here.

**Deleted:** loadbalancers

| Parameter | Style | Type | Description |
|---|---|---|---|
| description | plain | xsd:string | The load balancer description. |
| vip_address | plain | xsd:ip | The virtual IP (VIP) address. |
| status | plain | xsd:string | The status of the load balancer, which indicates whether the load balancer is operational. |
| admin_state_up | plain | xsd:boolean | The administrative state of the load balancer, which is up (true) or down (false). |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the VIP. Only administrative users can specify a tenant UUID other than their own. |

**Comment [KH27]:** The table does not include all of the possible parameters, shown in the response example. I would image that this table should be the same as the one for Create load balancer, so perhaps you could just copy that here.

**Deleted:** of the VIP

**Deleted:** . I

Example: Update a load balancer, JSON response

**Moved (insertion) [6]**

**Deleted:** .

**Deleted:** :

```
{
  "loadbalancer": {
    "admin_state_up": false,
    "description": "simple lb2",
    "id": "a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "listeners": [],
    "name": "loadbalancer2",
    "operating_status": "ONLINE",
    "provisioning_status": "PENDING_UPDATE",
    "tenant_id": "b7c1a69e88bf4b21a8148f787aef2081",
    "vip_address": "10.0.0.4",
    "vip_subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2"
  }
}
```

# Remove a load balancer

```
DELETE /v2.0/lbaas/loadbalancers/{loadbalancer_id}
```

This operation removes a load balancer and its associated configuration from the tenant account. All configuration data is immediately purged and cannot be recovered.

**Deleted:** Removes

**Deleted:** ¶
Any and all

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 204 | No Content | The server has fulfilled the request but does not need to return an entity body. |
| 400 | Bad Request | The request is missing one or more elements, or the values of some elements are invalid. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted: Code**

**Comment [KH28]:** "a response body"?

**Deleted:** -

**Deleted:** above

**Deleted:** has

## Request

This operation does not accept a request body.

## Response

This operation does not return a response body.

# Listeners

A *listener* is an object that contains data that pertains to the “listening” port. This object defines the front end of the configuration and contains the back-end data such as pools and its members.

**Deleted:** containing

**Formatted:** Font: Italic

**Deleted:** pertaining

**Deleted:** “frontend”

## List listeners

```
GET /v2.0/lbaas/loadbalancers
```

**Comment [KH29]:** This is the same URI as the list load balancers operation. Shouldn’t this be specific to listeners?

This operation lists all load balancers that are associated with your tenant account.

**Comment [KH30]:** Should list the listeners, right? Revise this intro as needed to be correct.

**Deleted:** Lists

This operation returns a list, which might be empty. Each list element is a listener that can contain the following attributes:

- id
- tenant_id
- name
- description
- protocol
- protocol_port
- connection_limit
- default_pool_id
- admin_state_up
- loadbalancers
- default_tls_container_ref
- sni_container_refs

**Comment [KH31]:** Delete this list, and just ensure that all of these items are covered in the response parameters table. Hm. You need to add a response parameters table!

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |

**Deleted:** Code

**Deleted:** Request

| Response code | Name | Description |
|---|---|---|
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** Code

**Deleted:** has

## Request

This operation does not accept a request body.

## Response

[Insert a table that lists all of the response body parameters.]

**Deleted:** This operation does not accept a request body.

Example: List listeners, JSON response

**Deleted:** .

**Deleted:** :

```
{
  "listeners": [
    {
      "admin_state_up": true,
      "connection_limit": 100,
      "default_pool_id": null,
      "description": "",
      "id": "35cb8516-1173-4035-8dae-0dae3453f37f",
      "loadbalancers": [
        {
          "id": "a9729389-6147-41a3-ab22-a24aed8692b2"
        }
      ],
      "name": "",
      "protocol": "HTTP",
      "protocol_port": 80,
      "tenant_id": "3e4d8bec50a845fcb09e03a4375c691d",
      "default_tls_container_ref":
"https://barbican.endpoint/containers/a36c20d0-18e9-42ce-88fd-82a35977ee8c",
      "sni_container_refs": [
      "https://barbican.endpoint/containers/b36c20d0-18e9-42ce-88fd-82a35977ee8d",
      "https://barbican.endpoint/containers/c36c20d0-18e9-42ce-88fd-82a35977ee8e"
     ]
   }
  ]
}
```

# Create a listener

```
POST /v2.0/lbaas/listeners
```

This operation provisions a new listener based on the configuration defined in the request. After the request is validated and the provisioning process begins, a response is returned. The response contains a unique identifier for the listener.

At a minimum, you must specify the following listener attributes:

- tenant_id
- loadbalancer_id
- description
- protocol

Some attributes receive default values if you omit them from the request. See the body parameters table for details.

If the request cannot be fulfilled due to insufficient or invalid data, the service returns the HTTP Bad Request (400) response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

Users with an administrative role can create listeners on behalf of other tenants by specifying a tenant_id attribute different than their own.

**Comment [KH32]:** In the online doc, the link to this section is actually going to the "Create a listener" section in the Getting Started Guide part of the doc. I guess we have to be careful to create distinct permalinks for headings that are very similar.

**Deleted:** object

**Deleted:** object

**Deleted:** object

**Deleted:** these

**Deleted:** . Required only if the caller has an administrative role and wants to create a listener for another tenant.

**Deleted:** . The load balancer on which this listener is provisioned. A tenant can only create listeners on load balancers authorized by policy. For example, her own load balancers.

**Deleted:** . The load balancer description.

**Comment [KH33]:** Don't provide descriptions here. Put all of the relevant information in the parameter table.

**Deleted:** The protocol for which the front end listens. Must be HTTP, HTTPS, TCP, or TERMINATED_HTTPS.

**Deleted:** :

**Deleted:** <#>protocol_port. The port on which the front end listens. Must be an integer from 1 to 65535.¶
<#>default_tls_container_ref. The reference to a container that holds TLS secrets. If you also specifysni_container_refs, this container is the default. This parameter is required for the TERMINATED_HTTPS protocol.¶
<#>sni_container_refs. A list of references to containers that hold TLS secrets that are used for Server Name Indication (SNI). This parameter is required for the TERMINATED_HTTPS protocol.¶
<#>admin_state_up. Default is true.¶
<#>name. Default is an empty string.¶
<#>description. Default is an empty string.¶
<#>connection_limit. Default is -1, which indicates an infinite limit.¶

**Deleted:** You can configure all documented features of the listener at creation time by specifying the additional elements or attributes in the request.¶

A listener cannot be created if the load balancer that it is attempting to attach to does not have a provisioning_status of ACTIVE.

**Deleted:** updated

**Deleted:** be

**Deleted:** ed

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 201 | Created | The request was fulfilled and a new resource was created. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 404 | Not Found | The requested item was not found. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Comment [KH34]:** The preceding text mentions the 400 response code. Should that be included in the table?

**Deleted:** Code

**Deleted:** has been

**Deleted:** resulted in

**Deleted:** being

**Deleted:** due to

**Deleted:** above

**Deleted:** has

## Request

**Moved down [7]:** Example. Create listener: JSON request¶

The following table shows the body parameters for the request.

**Deleted:** This list

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| listener | plain | xsd:string | A listener object. |

| Parameter | Style | Type | Description |
|---|---|---|---|
| default_pool_id (optional) | plain | csapi:uuid | The UUID of the default pool. It must have compatible protocol with the listener. |
| name | plain | xsd:string | The listener name. If you don’t specify a value, the default is an empty string. |
| description (optional) | plain | xsd:string | The listener detailed description. If you don’t specify a value, the default is an empty string. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the virtual IP (VIP) address. Only administrative users can specify a tenant UUID other than their own. |
| connection_limit (optional) | plain | xsd:int | The maximum number of connections permitted for this load balancer. The default is -1, which indicates an infinite limit. |
| protocol | plain | xsd:string | The protocol for which the front end listens. Valid values are `HTTP`, `HTTPS`, `TCP`, and `TERMINATED_HTTPS`. |
| protocol_port | plain | xsd:int | The TCP or UDP port on which the front edn listens. The value must be an integer from 1 to 65535. |
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the load balancer, which is up (true) or down (false). Set this attribute to `false` to create the listener in an administratively down state. The default is `true`. |
| loadbalancer_id | plain | csapi:uuid | The UUID for the load balancer on which the listener is provisioned. Tenants can create listeners only on load balancers authorized by policy, for example, their own load balancers. |

**Deleted:** VIP

**Comment [KH35]:** Should this be optional?

**Deleted:** is infinite

**Deleted:** to load balance

**Deleted:** A v

**Deleted:** is

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** or

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** to listen

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** .

| Parameter | Style | Type | Description |
|---|---|---|---|
| default_tls_container_ref (optional) | plain | xsd:string | A reference to a container of Transport Layer Security (TLS) secrets. If you also specify `sni_container_refs`, this container is the default. This parameter is required for the `TERMINATED_HTTPS` protocol. |
| sni_container_refs<br>(optional) | plain | xsd:list | A list of references to containers that hold TLS secrets that are used for Server Name Indication (SNI). This parameter is required for the `TERMINATED_HTTPS` protocol. |

Example: Create a listener, JSON request

```
{
  "listener": {
    "admin_state_up": true,
    "connection_limit": 100,
    "description": "listener one",
    "loadbalancer_id": "a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "name": "listener1",
    "protocol": "HTTP",
    "protocol_port": "80",
    "default_tls_container_ref":
"https://barbican.endpoint/containers/a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "sni_container_refs": [
      "https://barbican.endpoint/containers/b36c20d0-18e9-42ce-88fd-
82a35977ee8d",
      "https://barbican.endpoint/containers/c36c20d0-18e9-42ce-88fd-
82a35977ee8e"
    ]
  }
}
```

## Response

[Insert a response parameter table.]

Example: Create a listener, JSON response

**Comment [KH36]:** This portion appears to be a link.

**Comment [KH37]:** Strangely formatted in the file.

**Deleted:** TLS secrets

**Moved (insertion) [7]**

**Deleted:** .

**Deleted:** :

**Deleted:** .

**Deleted:** :

```
{
  "listener": {
    "admin_state_up": true,
    "connection_limit": 100,
    "default_pool_id": null,
    "description": "listener one",
    "id": "39de4d56-d663-46e5-85a1-5b9d5fa17829",
    "loadbalancers": [
      {
        "id": "a36c20d0-18e9-42ce-88fd-82a35977ee8c"
      }
    ],
    "name": "listener1",
    "protocol": "HTTP",
    "protocol_port": 80,
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
    "default_tls_container_ref":
"https://barbican.endpoint/containers/a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "sni_container_refs": [
      "https://barbican.endpoint/containers/b36c20d0-18e9-42ce-88fd-
82a35977ee8d",
      "https://barbican.endpoint/containers/c36c20d0-18e9-42ce-88fd-
82a35977ee8e"
    ]
  }
}
```

## Show listener details

```
GET /v2.0/lbaas/listeners/{listener_id}
```

This operation returns the listener object identified by listener_id. If the user is not an administrative user and the listener object does not belong to the user’s tenant account, the service returns the HTTP Forbidden (403) response code.

**Deleted:** a

**Deleted:** her

**Deleted:** call

If this operation succeeds, it returns a listener element that can contain the following attributes:

- id
- tenant_id
- name

- description
- protocol
- protocol_port
- connection_limit
- default_pool_id
- admin_state_up
- loadbalancers
- default_tls_container_ref
- sni_container_refs

**Comment [KH38]:** Delete this list and ensure that all the parameters are listed in the response body parameter table. Well, you will need to *create* a response body parameter table!

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 403 | Forbidden | The server understood the request, but won’t fulfill it. |
| 404 | Not Found | The requested item was not found. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** Code

**Deleted:** Request

**Deleted:** is refusing to

**Deleted:** due to

**Deleted:** above

**Deleted:** has

## Request

This operation does not accept a request body.

## Response

[Insert a response body parameter table.]

### Example: Show listener details, JSON response

**Deleted:** .

**Deleted:** :

```
{
  "listener": {
    "admin_state_up": true,
    "connection_limit": 100,
    "default_pool_id": null,
    "description": "",
    "id": "35cb8516-1173-4035-8dae-0dae3453f37f",
    "loadbalancers": [
      {
        "id": "a9729389-6147-41a3-ab22-a24aed8692b2"
      }
    ],
    "name": "",
    "protocol": "HTTP",
    "protocol_port": 80,
    "tenant_id": "3e4d8bec50a845fcb09e03a4375c691d",
    "default_tls_container_ref":
"https://barbican.endpoint/containers/a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "sni_container_refs": [
      "https://barbican.endpoint/containers/b36c20d0-18e9-42ce-88fd-
82a35977ee8d",
      "https://barbican.endpoint/containers/c36c20d0-18e9-42ce-88fd-
82a35977ee8e"
    ]
  }
}
```

# Update a listener

PUT /v2.0/lbaas/listeners/{listener_id}

The update operation enables you to change one or more of the following listener attributes:

**Deleted:** This operation updates the attributes of a listener. Upon successful validation of the request, the service returns the HTTPAccepted (202) response code.¶

**Deleted:** the caller

- name
- description
- admin_state_up
- connection_limit
- default_tls_container_ref
- sni_container_refs

**Notes:**

- You cannot update the listener id, tenant_id, loadbalancer_id, loadbalancers, default_pool_id, protocol, and protocol_port listener attributes. Attempting to update an immutable attribute results in the HTTP Immutable (422) response code. If the request is validated, the service returns the HTTP Accepted (202) response code.

- You can update a listener only if the load balancer to which the listener is attached has a provisioning_status of ACTIVE.

**Deleted:** cannot

**Deleted:** does not have

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 400 | Bad Request | The request is missing one or more elements, or the values of some elements are invalid. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service | The service is not available. |

**Comment [KH39]:** What about the 202 and 422 codes mentioned in the preceding text. Should they be listed in the table?

**Deleted:** Code

**Deleted:** Request

**Deleted:** above

**Deleted:** has

| Response code | Name | Description |
|---|---|---|
| | Unavailable | |

**Comment [KH39]:** What about the 202 and 422 codes mentioned in the preceding text. Should they be listed in the table?

**Deleted:** Code

## Request

The following table shows the body parameters for the request.

**Deleted:** Example. Update listener: JSON request

**Comment [KH40]:** Apply the same edits from the “Create a listener” table, as applicable.

| Parameter | Style | Type | Description |
|---|---|---|---|
| listener | plain | xsd:string | A listener object. |
| default_pool_id (optional) | plain | csapi:uuid | The UUID of the default pool. It must have compatible protocol with the listener. |
| name | plain | xsd:string | The listener name. |
| description (optional) | plain | xsd:string | The listener detailed description. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the VIP. Only administrative users can specify a tenant UUID other than their own. |
| connection_limit (optional) | plain | xsd:int | The maximum number of connections permitted for this load balancer. The default is infinite . |
| protocol | plain | xsd:string | The protocol to load balance. A valid value is HTTP, HTTPS, TCP, or TERMINATED_HTTPS. |
| protocol_port | plain | xsd:int | The TCP or UDP port on which to listen. |
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the load balancer, which is up (true) or down (false). Set this attribute to false to create the listener in an administratively down state. |

| Parameter | Style | Type | Description |
|---|---|---|---|
| loadbalancer_id | plain | csapi:uuid | The UUID for the load balancer. |
| default_tls_container_ref (optional) | plain | xsd:string | A reference to a container of TLS secrets. |
| sni_container_refs<br>(optional) | plain | xsd:list | A list of references to TLS secrets. |

**Comment [KH40]:** Apply the same edits from the “Create a listener” table, as applicable.

Example: Update a listener, JSON request

```
{
  "listener": {
    "admin_state_up": false,
    "connection_limit": 200,
    "description": "listener two",
    "name": "listener2",
    "default_tls_container_ref":
"https://barbican.endpoint/containers/a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "sni_container_refs": [
      "https://barbican.endpoint/containers/b36c20d0-18e9-42ce-88fd-
82a35977ee8d",
      "https://barbican.endpoint/containers/c36c20d0-18e9-42ce-88fd-
82a35977ee8e"
    ]
  }
}
```

## Response

[Insert a response body parameter table.]

Example: Update a listener, JSON response

**Deleted:** .

**Deleted:** :

```
{
  "listener": {
    "admin_state_up": false,
    "connection_limit": 200,
    "default_pool_id": null,
    "description": "listener two",
    "id": "39de4d56-d663-46e5-85a1-5b9d5fa17829",
    "loadbalancers": [
      {
        "id": "a36c20d0-18e9-42ce-88fd-82a35977ee8c"
      }
    ],
    "name": "listener2",
    "protocol": "HTTP",
    "protocol_port": 80,
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
    "default_tls_container_ref":
"https://barbican.endpoint/containers/a36c20d0-18e9-42ce-88fd-82a35977ee8c",
    "sni_container_refs": [
      "https://barbican.endpoint/containers/b36c20d0-18e9-42ce-88fd-
82a35977ee8d",
      "https://barbican.endpoint/containers/c36c20d0-18e9-42ce-88fd-
82a35977ee8e"
    ]
  }
}
```

# Remove a listener

```
DELETE /v2.0/lbaas/listeners/{listener_id}
```

This operation removes a listener and its associated configuration from the tenant account. All configuration data is immediately purged and cannot be recovered.

**Deleted:** ny and a

You can delete a listener only if the load balancer to which the listener is attached has a provisioning_status of ACTIVE.

**Deleted:** cannot

**Deleted:** it

**Deleted:** does not have

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response Code | Name | Description |
|---|---|---|

| Response Code | Name | Description |
|---|---|---|
| 204 | No Content | The server has fulfilled the request but does not need to return an entity body. |
| 400 | Bad Request | The request is missing one or more elements, or the values of some elements are invalid. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** -

**Deleted:** due to

**Deleted:** above

**Deleted:** has

## Request

This operation does not accept a request body.

## Response

This operation does not return a response body.

# Pools

A *pool* is a logical set of devices, such as web servers, that you group together to receive and process traffic. Instead of sending client traffic to the destination IP address specified in the client request, the system sends the request to any of the servers that are members of that pool.

**Formatted:** Font: Italic

## List pools

```
GET /v2.0/lbaas/pools
```

This operation returns a response body that contains a list of pools associated with the tenant account. Each pool element in the list can contain the following attributes:

**Deleted:** , which might be empty

- id
- tenant_id
- name
- description
- protocol
- lb_algorithm
- session_persistence
- admin_state_up
- listeners
- members
- healthmonitor_id

**Comment [KH41]:** Delete this list an ensure that all of the parameters are listed in the parameter table.

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |

**Deleted:** Code

**Deleted:** Request

| Response code | Name | Description |
|---|---|---|
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

## Request

This operation does not accept a request body.

## Response

The following table shows the body parameters for the response.

| Parameter | Style | Type | Description |
|---|---|---|---|
| pools | plain | xsd:list | A list of pool objects. |
| admin_state_up | plain | xsd:boolean | The administrative state of the load balancer, which is up (true) or down (false). |
| description | plain | xsd:string | The pool detailed description. |
| healthmonitor_id | plain | csapi:uuid | The UUID of the associated health monitor. |
| id | plain | csapi:uuid | The listener ID. |
| lb_algorithm | plain | xsd:string | The load-balancer algorithm– such as round robin (`ROUND_ROBIN`), least connections (`LEAST_CONNECTIONS`), and source IP (`SOURCE_IP`)–that is used to distribute traffic to the pool members. This value, which must be supported, depends on the load-balancer provider. The round robin |

**Deleted:** Code

**Deleted:** has

**Deleted:** Example. List pools: JSON response¶

**Deleted:** This list

**Deleted:** :

**Deleted:** for

**Comment [KH42]:** pool ?

**Deleted:** name

**Deleted:** , which is

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** -

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** , and so on,

**Comment [KH43]:** ? Isn't Rackspace the load balancer provider?

**Deleted:** is

**Deleted:** ent

**Deleted:** -

| Parameter | Style | Type | Description |
|---|---|---|---|
| | | | algorithm must be supported. |
| listeners | plain | xsd:list | A list of the listeners that belong to the pool. |
| members | plain | xsd:list | A list of the members that belong to the pool. |
| name (optional) | plain | xsd:string | The pool name. The name does not have to be unique. |
| protocol | plain | xsd:string | The protocol of the pool, which is `TCP`, `HTTP`, or `HTTPS`. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the virtual IP (VIP) address. Only administrative users can specify a tenant UUID other than their own. |

**Deleted:** The

**Deleted:** The

**Deleted:** It

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

### Example: List pools, JSON response

```
{
  "pools": [
    {
      "admin_state_up": true,
      "description": "simple pool",
      "healthmonitor_id": null,
      "id": "4c0a0a5f-cf8f-44b7-b912-957daa8ce5e5",
      "lb_algorithm": "ROUND_ROBIN",
      "listeners": [
        {
          "id": "35cb8516-1173-4035-8dae-0dae3453f37f"
        }
      ],
      "members": [],
      "name": "pool1",
      "protocol": "HTTP",
      "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c"
    }
  ]
}
```

# Create a pool

POST /v2.0/lbaas/pools

This operation provisions a new pool based on the configuration defined in the request. After the request is validated and progress has started on the provisioning process, a response is returned. The response contains a unique identifier for the pool.

The request must specify the following pool attributes:

- tenant_id
- protocol
- lb_algorithm
- protocol_port
- listener_id

Some attributes receive default values if you omit them from the request. For details, see the request parameters table.

If the request cannot be fulfilled because of insufficient or invalid data, the service returns the HTTP Bad Request (400) response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

Users with an administrative role can create pools on behalf of other tenants by specifying a tenant_id attribute that is different than their own.

You cannot update a pool if the load balancer to which it is attempting to attach does not have a provisioning_status of ACTIVE.

The following table shows the possible response codes for this operation.

| Response code | Name | Description |
|---|---|---|
| 201 | Created | The request was fulfilled and a new resource was created. |

**Comment [KH44]:** In the online doc, the link to this section is actually going to the "Create a pool" section in the Getting Started Guide part of the doc.

**Deleted:** object

**Deleted:** object

**Deleted:** object

**Deleted:** The caller of this operation

**Deleted:** these

**Deleted:** , at a minimum

**Deleted:** . Required only if the caller has an administrative role and wants to create a pool for another tenant.

**Deleted:** . The protocol for which this pool and its members listen. A valid value is TCP, HTTP, or HTTPS.

**Deleted:** . The load-balancer algorithm, such as ROUND_ROBIN, LEAST_CONNECTIONS, and SOURCE_IP, that is used to distribute traffic to the pool members. This value, which must be supported, is dependent on the load-balancer provider.

**Deleted:** . The port on which the front end listens. Must be an integer from 1 to 65535.

**Deleted:** . The ID of the listener in which this pool becomes the default pool. Each listener can have only one default pool.

**Deleted:** :

**Deleted:** ¶
<#>admin_state_up. Default is true.¶
<#>name. Default is an empty string.¶
<#>description. Default is an empty string.¶
<#>session_persistence. Default is an empty dictionary.¶

**Deleted:** due to

**Deleted:** Users can configure all documented features at creation time by providing the additional elements or attributes in the request.¶

**Deleted:** be

**Deleted:** ed

**Deleted:** This

**Deleted:** :

**Comment [KH45]:** What about 400, which is mentioned in the preceding text. Should that be listed here?

**Deleted:** Code

**Deleted:** has been

**Deleted:** resulted in

**Deleted:** being

| Response code | Name | Description |
|---|---|---|
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 404 | Not Found | The requested item was not found. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Comment [KH45]:** What about 400, which is mentioned in the preceding text. Should that be listed here?

**Deleted:** Code

**Deleted:** due to

**Deleted:** above

**Deleted:** has

## Request

**Moved down [8]:** Example. Create pool: JSON request¶

The following table shows the body parameters for the request.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| tenant_id (optional) | plain | csapi:uuid | The UUID of the tenant who owns the pool. Only administrative users can specify a tenant UUID other than their own. |
| name (optional) | plain | xsd:string | The pool name. The name does not have to be unique. If you do not specify a value, the default is an empty string. |
| description | plain | xsd:string | The human-readable description for the pool. If you do not |

**Comment [KH46]:** The intro says that `protocol_port` must be specified, but I don't see that in the table or example. Should the intro be changed list `protocol` instead?

Also, the table does not list `session_persistence`.

**Deleted:** Does

| Parameter | Style | Type | Description |
|---|---|---|---|
| (optional) | | | specify a value, the default is an empty string. |
| protocol | plain | xsd:string | The protocol of the pool, which is TCP, HTTP, or HTTPS. |
| subnet_id | plain | csapi:uuid | The UUID of the subnet on which to allocate the virtual IP (VIP) address. |
| lb_algorithm | plain | xsd:string | The load-balancer algorithm~~-~~such as round robin (ROUND_ROBIN), least connections (LEAST_CONNECTIONS), and source IP (SOURCE_IP)~~-~~that is used to distribute traffic to the pool members. This value, which must be supported, depends on the load-balancer provider. The round robin algorithm must be supported. |
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the load balancer, which is up (true) or down (false). The default is true. |
| listener_id (optional) | plain | csapi:uuid | The UUID of the listener in which this pool becomes the default pool. Each listener can have only one default pool.. |

**Comment [KH46]:** The intro says that protocol_port must be specified, but I don't see that in the table or example. Should the intro be changed list protocol instead?

Also, the table does not list session_persistence.

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** , which is

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** -

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** , and so on,

**Comment [KH47]:** What does "provider" mean here?

**Deleted:** is dependent

**Deleted:** -

## Example: Create a pool, JSON request

**Moved (insertion) [8]**

**Deleted:** .

**Deleted:** :

```
{
  "pool": {
    "admin_state_up": true,
    "description": "simple pool",
    "lb_algorithm": "ROUND_ROBIN",
    "listener_id": "39de4d56-d663-46e5-85a1-5b9d5fa17829",
    "name": "pool1",
    "protocol": "HTTP",
    "session_persistence": {
      "cookie_name": "my_cookie",
      "type": "APP_COOKIE"
    }
  }
}
```

## Response

The following table shows the body parameters for the response.

| Parameter | Style | Type | Description |
|---|---|---|---|
| pool | plain | xsd:dict | A pool object. |
| status | plain | xsd:string | A pool object. |
| protocol | plain | xsd:string | The protocol of the pool, which is TCP, HTTP, or HTTPS. |
| description (optional) | plain | xsd:string | The description of the pool. |
| tenant_id (optional) | plain | csapi:uuid | The UUID of the tenant who owns the pool. Only administrative users can specify a tenant UUID other than their own. |
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the pool, which is up (true) or down (false). |
| name | plain | xsd:string | The pool name. The name does not have to be unique. |
| members | plain | xsd:list | The list of members that belong to the pool. |
| lb_algorithm | plain | xsd:string | The load-balancer algorithm such as round robin (ROUND_ROBIN), least connections (LEAST_CONNECTIONS), and source IP (SOURCE_IP) that is used to distribute traffic to the pool members. This value, which must be supported, depends on the load-balancer provider. The round robin algorithm must be supported. |
| healthmonitor_id | plain | xsd:string | The UUID of the health monitor. |

**Deleted:** Example. Create pool: JSON response¶

**Deleted:** This

**Deleted:** :

**Comment [KH48]:** I don't see status listed in the example. If it is indeed a possible parameter, then revise the definition here to be accurate.

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** for

**Deleted:** Does

**Deleted:** , which is

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** -

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** , and so on,

**Comment [KH49]:** See my question in previous sections.

**Deleted:** is dependent

**Deleted:** -

| Parameter | Style | Type | Description |
|---|---|---|---|
| (optional) | | | |
| session_persistence (optional) | plain | xsd:string | The session persistence algorithm. This algorithm is a dictionary with `type` and `cookie_name` keys. |
| id | plain | csapi:uuid | The UUID of the pool. |
| subnet_id | plain | csapi:uuid | The UUID of the subnet. |
| vip_id | plain | csapi:uuid | The UUID of the virtual IP (VIP) address. |
| healthmonitors_status | plain | xsd:string | The statuses of the health monitors that are associated with the pool. |

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** for

### Example: Create a pool, JSON response

```
{
  "pool": {
    "admin_state_up": true,
    "description": "simple pool",
    "healthmonitor_id": null,
    "id": "12ff63af-4127-4074-a251-bcb2ecc53ebe",
    "lb_algorithm": "ROUND_ROBIN",
    "listeners": [
      {
        "id": "39de4d56-d663-46e5-85a1-5b9d5fa17829"
      }
    ],
    "members": [],
    "name": "pool1",
    "protocol": "HTTP",
    "session_persistence": {
      "cookie_name": "my_cookie",
      "type": "APP_COOKIE"
    },
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c"
  }
}
```

# Show pool details

```
GET /v2.0/lbaas/pools/{pool_id}
```

This operation returns the pool object identified by pool_id. If the user is not an administrative user and the pool object does not belong to the user's tenant account, the service returns the HTTP Forbidden (403) response code.

**Deleted:** a

**Deleted:** her

**Deleted:** call

If this operation succeeds, it returns a pool element that can contain the following attributes:

- id
- tenant_id
- name
- description
- protocol
- lb_algorithm
- session_persistence
- admin_state_up
- listeners
- members
- healthmonitor_id

**Comment [KH50]:** Delete this list and ensure that all the parameters are defined in the parameter table.

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication |

**Deleted: Code**

**Deleted:** Request

| Response code | Name | Description |
|---|---|---|
|  |  | token. |
| 403 | Forbidden | The server understood the request, but won’t fulfill it. |
| 404 | Not Found | The requested item was not found. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted: Code**

**Deleted:** is refusing to

**Deleted:** due to

**Deleted:** above

**Deleted:** has

## Request

This operation does not accept a request body.

## Response

**Moved down [9]:** *Example. Show pool details: JSON response*

The following table shows the body parameters for the response.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| pools | plain | xsd:list | A list of pool objects. |
| admin_state_up | plain | xsd:boolean | The administrative state of the pool, which is up (true) or down (false). |

| Parameter | Style | Type | Description |
|---|---|---|---|
| description | plain | xsd:string | The description of the pool. |
| healthmonitor_id | plain | xsd:string | The UUID of the health monitor. |
| id | plain | csapi:uuid | The UUID of the pool. |
| lb_algorithm | plain | xsd:string | The load-balancer algorithm such as round robin (ROUND_ROBIN), least connections (LEAST_CONNECTIONS), and source IP (SOURCE_IP) that is used to distribute traffic to the pool members. This value, which must be supported, depends on the load-balancer provider. The round-robin algorithm must be supported. |
| listeners | plain | xsd:list | A list of the listeners that belong to the pool. |
| members | plain | xsd:list | A list of the members that belong to the pool. |
| name (optional) | plain | xsd:string | The pool name. The name does not have to be unique. |
| protocol | plain | xsd:string | The protocol of the pool, which is TCP, HTTP, or HTTPS. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the pool. Only administrative users can specify a tenant UUID other than their own. |

**Deleted:** for

**Deleted:** for

**Deleted:** , which is

**Formatted:** Font: (Default) Courier New

**Deleted:** -

**Deleted:** , and so on,

**Formatted:** Font: (Default) Courier New

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** is dependent

**Comment [KH51]:** ?

**Deleted:** The

**Deleted:** The

**Deleted:** Does

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

## Example: Show pool details, JSON response

**Moved (insertion) [9]**

**Deleted:** .

**Deleted:** :

```
{
  "pool": {
    "admin_state_up": true,
    "description": "simple pool",
    "healthmonitor_id": null,
    "id": "4c0a0a5f-cf8f-44b7-b912-957daa8ce5e5",
    "lb_algorithm": "ROUND_ROBIN",
    "listeners": [
      {
        "id": "35cb8516-1173-4035-8dae-0dae3453f37f"
      }
    ],
    "members": [],
    "name": "pool1",
    "protocol": "HTTP",
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c"
  }
}
```

# Update a pool

```
PUT /v2.0/lbaas/pools/{pool_id}
```

The update operation enables you to change the following pool attributes:

**Deleted:** This operation updates the attributes of a pool. Upon successful validation of the request, the service returns the HTTP Accepted (202) response code.
¶

**Deleted:** the caller

**Deleted:** one or more of

- name
- description
- admin_state_up
- lb_algorithm
- session_persistence

**Notes:**

- You cannot update the pool id, tenant_id, listener_id, listeners, healthmonitor_id, protocol, and members immutable attributes. If you try to update any of these attributes, the service returns the HTTP Immutable (422) response code. If the request is validated, the service returns the HTTP Accepted (202) response code.
- You can update a pool only if the load balancer to which the pool is attached has a provisioning_status of ACTIVE.

**Deleted:** cannot

**Deleted:** it

**Deleted:** does not have

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 400 | Bad Request | The request is missing one or more elements, or the values of some elements are invalid. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Comment [KH52]:** Should the table list 202 and 422?

**Deleted: Code**

**Deleted:** Request

**Deleted:** above

**Deleted:** has

## Request

**Moved down [10]:** ***Example. Update pool: JSON request***

This table shows the body parameters for the request:

| Parameter | Style | Type | Description |
|---|---|---|---|
| pool | plain | xsd:dict | A pool object. |
| name (optional) | plain | xsd:string | A human-readable name for the pool. The name does not have to be unique. |
| description | plain | xsd:string | A human-readable description of the pool. |
| lb_method | plain | xsd:string | The load-balancer algorithm such as round robin |

**Deleted:** Does

**Deleted:** for

**Deleted:** , which is

| Parameter | Style | Type | Description |
|---|---|---|---|
| (optional) | | | (ROUND_ROBIN), least connections (LEAST_CONNECTIONS), source IP (SOURCE_IP) that is used to distribute traffic to the pool members. This value, which must be supported, depends on the load-balancer provider. The round-robin algorithm must be supported. |
| admin_state_up | plain | xsd:boolean | The administrative state of the pool, which is up (true) or down (false). |

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** -

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** , and so on,

**Deleted:** is

**Comment [KH53]:** ?

**Deleted:** dependent

### Example: Update a pool, JSON request

**Moved (insertion) [10]**

**Deleted:** .

**Deleted:** :

```
{
  "pool": {
    "admin_state_up": false,
    "description": "pool two",
    "lb_algorithm": "LEAST_CONNECTIONS",
    "name": "pool2",
    "session_persistence": {
      "type": "HTTP_COOKIE"
    }
  }
}
```

## Response

**Deleted:** *Example. Update pool: JSON response*

The following table shows the body parameters for the response.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| pools | plain | xsd:list | A list of pool objects. |
| admin_state_up | plain | xsd:boolean | The administrative state of the pool, which is up (true) or down (false). |

| Parameter | Style | Type | Description |
|---|---|---|---|
| description | plain | xsd:string | The description of the pool. |
| healthmonitor_id | plain | xsd:string | The UUID of the health monitor. |
| id | plain | csapi:uuid | The UUID of the pool. |
| lb_algorithm | plain | xsd:string | The load-balancer algorithm such as round robin (ROUND_ROBIN), least connections (LEAST_CONNECTIONS), and source IP (SOURCE_IP) that is used to distribute traffic to the pool members. This value, which must be supported, depends on the load-balancer provider. The round-robin algorithm must be supported. |
| listeners | plain | xsd:list | A list of the listeners that belong to the pool. |
| members | plain | xsd:list | A list of the members that belong to the pool. |
| name (optional) | plain | xsd:string | The pool name. The name does not have to be unique. |
| protocol | plain | xsd:string | The protocol of the pool, which is TCP, HTTP, or HTTPS. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the pool. Only administrative users can specify a tenant UUID other than their own. |

**Deleted:** for

**Deleted:** for

**Deleted:** , which is

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** -

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** , and so on,

**Formatted:** Code Char, Font: +Body (Calibri)

**Comment [KH54]:** ?

**Deleted:** is dependent

**Deleted:** The

**Deleted:** The

**Deleted:** Does

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

### Example: Update a pool, JSON response

```
{
  "pool": {
    "admin_state_up": false,
    "description": "pool two",
    "healthmonitor_id": null,
    "id": "12ff63af-4127-4074-a251-bcb2ecc53ebe",
    "lb_algorithm": "LEAST_CONNECTIONS",
    "listeners": [
      {
        "id": "39de4d56-d663-46e5-85a1-5b9d5fa17829"
      }
    ],
    "members": [],
    "name": "pool2",
    "protocol": "HTTP",
    "session_persistence": {
      "cookie_name": null,
      "type": "HTTP_COOKIE"
    },
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c"
  }
}
```

## Remove a pool

```
DELETE /v2.0/lbaas/pools/{pool_id}
```

This operation removes a pool and its associated configuration from the tenant account. All configuration data is immediately purged and cannot be recovered.

**Deleted:** Any and a

You can delete a pool only if the load balancer to which the pool is attached has a provisioning_status of ACTIVE.

**Deleted:** cannot

**Deleted:** it

**Deleted:** does not have

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 204 | No Content | The server has fulfilled the request but does not need to return an entity body. |
| 400 | Bad Request | The request is missing one or more elements, or the values of some elements are invalid. |

**Deleted:** Code

**Deleted:** -

| Response code | Name | Description |
|---|---|---|
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** Code

**Deleted:** due to

**Deleted:** above

**Deleted:** has

## Request

This operation does not accept a request body.

## Response

This operation does not return a response body.

# Members

The *members* defined by the load balancer are responsible for servicing the requests received through the load balancer's virtual IP (VIP) address. By default, the load balancer employs a basic health check that ensures the member is listening on its defined port. The member is checked when it is added and at regular intervals as defined by the load balancer health check configuration. If a member 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 does not forward connections to it and its status is listed as OFFLINE. Only members that are in an ONLINE status can receive and service traffic from the load balancer.

All members have an associated *condition* that indicates whether the member is ENABLED, DISABLED, or DRAINING.

- Members that are in the ENABLED condition can receive new connections and service traffic from the load balancer.
- Members that are in the DISABLED condition cannot accept any new connections regardless of session-persistence configuration. Existing connections are forcibly terminated.
- When a member is in the DRAINING condition, the traffic manager does not send any additional *new* connections to the member, but honors established sessions. If the traffic manager receives a request and session persistence requires that the member is used, the traffic manager uses it.

**Note:** The condition of a member is not the same as its status. The condition attribute is mutable and gives the user control over how to manage requests to the member. The status attribute is immutable and is updated by the load balancing service based on whether the member *can* service requests.

If the WEIGHTED_ROUND_ROBIN load balancer algorithm mode is selected, then the user should assign the relevant weight to the member as part of the `weight` attribute of the member element. When the algorithm of the load balancer is changed to WEIGHTED_ROUND_ROBIN and the members do not already have an assigned weight, the service automatically sets the weight to 1 for all members.

[Everything I suggested deleting here was redundant.]

**Deleted:** at the time of addition

**Deleted:** back-end

**Deleted:** can

**Formatted:** Font: Italic

**Formatted:** Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

**Deleted:** an

**Deleted:** can

**Deleted:** The

**Deleted:** represents a member that

**Deleted:** accept or service traffic.

**Deleted:** A

**Deleted:** represents a member that stops

**Deleted:** from sending

**Deleted:** As mentioned earlier, the status is determined by the passive or active health monitors.

**Deleted:** Do not confuse the

**Deleted:** with

**Deleted:** on

**Deleted:** or not

**Deleted:** caller

**Deleted:** s

**Formatted:** Code Char, Font color: Auto

**Deleted:** Do not confuse the condition of a member with its status. The condition attribute is mutable and gives the user control on how to manage requests to the member. The status attribute is immutable and is updated by the load balancing service based on whether or not the member *can* service requests.¶
Every member in the load balancer has an associated condition which determines its role within the load balancer.¶
The following table lists the required and optional attributes:¶
Table. Required and optional attributes
¶
Name

...

# List pool members

GET /v2.0/lbaas/pools/{pool_id}/members

This operations lists all of the members that are associated with the specified pool.

**Deleted:** Lists

**Deleted:** a

**Deleted:** that is associated with your tenant account. The list of members includes only members that belong to the pool object identified by pool_id.

This operation returns a list, which might be empty. Each element in the list is a member that can contain the following attributes:

- id
- tenant_id
- address
- protocol_port
- weight
- subnet_id
- admin_state_up

**Comment [KH55]:** I suggest deleting this list and ensuring that the parameter table contains all the relevant parameters/attributes.

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** Code

**Deleted:** Request

**Deleted:** has

## Request

This operation does not accept a request body.

## Response

**Moved down [11]:** Example. List pool members: JSON response¶

The following table shows the body parameters for the response.

**Deleted:** This list

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| members | plain | xsd:list | A list of member objects. |
| id | plain | csapi:uuid | The UUID of the member. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the member. Only administrative users can specify a tenant UUID other than their own. |
| pool_id | plain | csapi:uuid | The UUID of the pool to which the member belongs. |
| address | plain | xsd:ip | The IP address of the member. |
| protocol_port | plain | xsd:int | The port where the application is hosted. |
| weight | plain | xsd:int | The portion of requests or connections that the member services compared to the other members of the pool. A value of 0 means that the member does not participate in load balancing but still accepts persistent connections. Valid values are from 0 to 256. |
| admin_state_up | plain | xsd:boolean | The administrative state of the member, which is up (true) or down (false). |
| status | plain | xsd:string | The status of the member, which indicates whether the member is operational. |

**Deleted:** for

**Deleted:** The UUID for the pool.

**Deleted:** The weight of a member determines the

**Deleted:** it

**Deleted:** -

**Comment [KH56]:** The text that I suggested deleting in the intro said that the value for `weight` must be an integer from 1 to 100. That seems at odds with what is said here, and I am wondering which is correct.

**Deleted:** A v

**Deleted:** is

**Deleted:** . I

Example: List pool members, JSON response

Moved (insertion) [11]

Deleted: .

Deleted: :

```
{
  "members": [
    {
      "address": "10.0.0.8",
      "admin_state_up": true,
      "id": "9a7aff27-fd41-4ec1-ba4c-3eb92c629313",
      "protocol_port": 80,
      "subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
      "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
      "weight": 1
    }
  ]
}
```

# Add a member to a pool

```
POST /v2.0/lbaas/pools/{pool_id}/members
```

This operation provisions a new member and adds it to a pool based on the configuration defined in the request. After the request is validated and progress has started on the provisioning process, a response is returned. The response contains a unique identifier for the member.

Deleted: object

Deleted: object

Deleted: object

At a minimum, you must specify the following pool attributes:

- tenant_id
- address
- protocol_port

Deleted: . Only required if the caller has an administrative role and wants to create a pool for another tenant.

Deleted: . The IP address of the member to receive traffic from the load balancer.

Deleted: The port on which the member is listening to receive traffic.

Some attributes receive default values if you omit them from the request. See the body parameters table for details.

Deleted: :

Deleted: ¶
<#>admin_state_up. Default is true.¶
<#>weight. Default is 1.¶

If you omit the subnet_id parameter, LBaaS uses the vip_subnet_id parameter value for the subnet ID.

If the request fails because of incorrect data, the service returns the HTTP Bad Request (400) response code with information about the failure in the response body. Validation errors require that you correct the error and submit the request again.

**Deleted:** due to

Users with an administrative role can create members on behalf of other tenants by specifying a tenant_id attribute that is different than their own.

**Deleted:** To configure all documented member features at creation time, specify additional elements or attributes in the request.¶

To add a member, the load balancer must have a provisioning_status of ACTIVE.

**Deleted:** update

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

**Comment [KH57]:** Should the table list 400?

**Deleted:** Code

| Response code | Name | Description |
|---|---|---|
| 201 | Created | The request has been fulfilled and a new resource was created. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 404 | Not Found | The requested item was not found. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** resulted in

**Deleted:** being

**Deleted:** due to

**Deleted:** above

**Deleted:** has

## Request

**Moved down [12]:** Example. Add member to a pool: JSON request¶

The following table shows the body parameters for the request.

**Deleted:** This list

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| member | plain | xsd:dict | A member object. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the member. Only administrative users can specify a tenant UUID other than their own. |
| address | plain | xsd:ip | The IP address of the member. |
| protocol_port | plain | xsd:int | The port where the application is hosted. |
| subnet_id (optional) | plain | xsd:int | The UUID of the subnet on which the member resides. If you omit this parameter, the service uses the vip_subnet_id parameter value for the subnet UUID. |

**Deleted:** The UUID for the pool.

**Deleted:** LBaaS

Example. Add a member to a pool: JSON request

**Moved (insertion) [12]**

```
{
  "member": {
    "address": "10.0.0.8",
    "admin_state_up": true,
    "protocol_port": "80",
    "subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
    "weight": "1"
  }
}
```

## Response

**Moved down [13]:** Example. Add member to pool: JSON response¶

The following table shows the body parameters for the response.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| member | plain | xsd:dict | A member object. |
| id | plain | csapi:uuid | The UUID of the member. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the member. Only administrative users can specify a tenant UUID other than their own. |
| subnet_id (optional) | plain | xsd:int | The UUID of the subnet on which the member resides. If you omit this parameter, the service uses the vip_subnet_id parameter value for the subnet UUID. |
| address | plain | xsd:ip | The IP address of the member. |
| protocol_port | plain | xsd:int | The port where the application is hosted. |
| weight (optional) | plain | xsd:int | The portion of requests or connections that the member services compared to the other members of the pool. A value of 0 means that the member does not participate in load balancing but still accepts persistent connections. Valid values are from 0 to 256. |
| admin_state_up | plain | xsd:boolean | The administrative state of the member, which is up (true) or down (false). |
| status | plain | xsd:string | The status of the member, which indicates whether the member is operational. |

### Example: Add a member to a pool, JSON response

```
{
  "member": {
    "address": "10.0.0.8",
    "admin_state_up": true,
    "id": "9a7aff27-fd41-4ec1-ba4c-3eb92c629313",
```

**Deleted:** for

**Deleted:** The UUID for the pool.

**Deleted:** LBaaS

**Deleted:** weight of a member determines the

**Deleted:** it

**Deleted:** -

**Comment [KH58]:** See my comment about this sentence in the preceding section.

**Deleted:** A v

**Deleted:** is

**Deleted:** . I

**Moved (insertion) [13]**

**Deleted:** .

**Deleted:** :

```
    "protocol_port": 80,
    "subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
    "weight": 1
  }
}
```

# Show pool member details

```
GET /v2.0/lbaas/pools/{pool_id}/members/{member_id}
```

This operation returns the specified member that belongs to the specified pool. If the user is not an administrative user and the pool or member object does not belong to the user's tenant account, the service returns the HTTP Forbidden (403) response code.

The following table shows the possible response codes for this operation.

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 403 | Forbidden | The server understood the request, but won't fulfill it. |
| 404 | Not Found | The requested item was not found. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |

**Deleted:** show

**Deleted:** a

**Deleted:** object identified by member_id

**Deleted:** a

**Deleted:** object identified by pool_id

**Deleted:** her

**Deleted:** If this operation succeeds, it returns a pool element that can contain the following attributes:¶
<#>id¶
<#>tenant_id¶
<#>address¶
<#>protocol_port¶
<#>weight¶
<#>subnet_id¶
<#>admin_state_up¶
This

**Deleted:** :

**Deleted:** Code

**Deleted:** Request

**Deleted:** is refusing to

**Deleted:** due to

**Deleted:** above

| Response code | Name | Description |
|---|---|---|
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** Code

**Deleted:** has

## Request

This operation does not accept a request body.

## Response

**Moved down [14]:** Example. Show pool member details: JSON response¶

The following table shows the body parameters for the response.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| member | plain | xsd:dict | A member object. |
| id | plain | csapi:uuid | The UUID of the member. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the member. Only administrative users can specify a tenant UUID other than their own. |
| pool_id | plain | csapi:uuid | The UUID of the pool to which the member belongs. |
| address | plain | xsd:ip | The IP address of the member. |
| protocol_port | plain | xsd:int | The port where the application is hosted. |

**Deleted:** for

**Deleted:** The UUID for the pool.

| Parameter | Style | Type | Description |
|---|---|---|---|
| weight (optional) | plain | xsd:int | The portion of requests or connections that the member services compared to the other members of the pool. A value of 0 means that the member does not participate in load balancing but still accepts persistent connections. A valid value is from 0 to 256. |
| admin_state_up | plain | xsd:boolean | The administrative state of the member, which is up (true) or down (false). |
| status | plain | xsd:string | The status of the member, which indicates whether the member is operational. |

**Deleted:** weight of a member determines the

**Deleted:** it

**Deleted:** -

**Comment [KH59]:** See previous comment.

**Deleted:** . I

Example: Show pool member details, JSON response

**Moved (insertion) [14]**

**Deleted:** .

**Deleted:** :

```
{
  "member":{
    "address":"10.0.0.8",
    "admin_state_up":true,
    "id":"9a7aff27-fd41-4ec1-ba4c-3eb92c629313",
    "protocol_port":80,
    "pool_id":"a5a8839d-1ac3-41f9-9aae-f375fa4da50a",
    "tenant_id":"1a3e005cf9ce40308c900bcb08e5320c",
    "weight":1
  }
}
```

# Update a pool member

```
POST /v2.0/lbaas/pools/{pool_id}/members/{member_id}
```

This operation enables you to change one or more of the following member attributes:

- weight
- admin_state_up

**Deleted:** Upon successful validation of the request, the service returns the HTTP OK (200) response code.¶
The update

**Deleted:** these

**Deleted:** pool

**Notes:**

- You cannot update the member id, tenant_id, address, protocol_port, and subnet_id attributes. If you attempt to update any of these attributes, the service returns the HTTP Immutable (422) response code.

**Deleted:** ID

- You can update a member only if the attached load balancer has a provisioning_status of ACTIVE.

**Deleted:** not

**Deleted:** does not have

The following table shows the possible response codes for this operation.

**Deleted:** This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 400 | Bad Request | The request is missing one or more elements, or the values of some elements are invalid. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** Code

**Deleted:** Request

**Deleted:** above

**Deleted:** has

## Request

**Moved down [15]:** Example. Update pool member: JSON request¶

The following table shows the body parameters for the request.

**Deleted:** This list

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| member | plain | xsd:dict | A member object. |
| pool_id (optional) | plain | csapi:uuid | The UUID of the pool to which the member belongs. |
| weight | plain | xsd:int | The portion of requests or connections that the member services compared to the other members of the pool. A value of 0 means that the member does not participate in load balancing but still accepts persistent connections. A valid value is from 0 to 256. |
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the member, which is up (true) or down (false). |

**Deleted:** weight of a member determines the

**Deleted:** it

**Deleted:** -

**Comment [KH60]:** See previous comment.

### Example: Update a pool member, JSON request

**Moved (insertion) [15]**

**Deleted:** :

```
{
  "member": {
    "admin_state_up": false,
    "weight": 5
  }
}
```

## Response

**Deleted:** Example. Update pool member: JSON response¶

The following table shows the body parameters for the response.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| member | plain | xsd:dict | A member object. |
| id | plain | csapi:uuid | The UUID of the member. |

**Deleted:** for

| Parameter | Style | Type | Description |
|---|---|---|---|
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the member. Only administrative users can specify a tenant UUID other than their own. |
| pool_id | plain | csapi:uuid | The UUID of the pool to which the member belongs. |
| address | plain | xsd:ip | The IP address of the member. |
| protocol_port | plain | xsd:int | The port where the application is hosted. |
| weight | plain | xsd:int | The portion of requests or connections that the member services compared to the other members of the pool. A value of 0 means that the member does not participate in load balancing but still accepts persistent connections. A valid value is from 0 to 256. |
| admin_state_up | plain | xsd:boolean | The administrative state of the member, which is up (true) or down (false). |
| status | plain | xsd:string | The status of the member, which indicates whether the member is operational. |

**Deleted:** The UUID for the pool.

**Deleted:** weight of a member determines the

**Deleted:** it

**Deleted:** -

**Comment [KH61]:** See previous comment

**Deleted:** . I

```
{
  "member": {
    "address": "10.0.0.8",
    "admin_state_up": false,
    "id": "9a7aff27-fd41-4ec1-ba4c-3eb92c629313",
    "protocol_port": 80,
    "subnet_id": "013d3059-87a4-45a5-91e9-d721068ae0b2",
    "tenant_id": "1a3e005cf9ce40308c900bcb08e5320c",
    "weight": 5
  }
}
```

## Remove a member from a pool

```
DELETE /v2.0/lbaas/pools/{pool_id}/members/{member_id}
```

This operation removes the specified member from the specified pool and removes its associated configuration from the tenant account.

All configuration data is immediately purged and cannot be recovered.

A member can be deleted only if the attached load balancer has a provisioning_status of ACTIVE.

The following table shows the possible response codes for this operation.

| Response Code | Name | Description |
|---|---|---|
| 204 | No Content | The server has fulfilled the request but does not need to return an entity body. |
| 400 | Bad Request | The request is missing one or more elements, or the values of some elements are invalid. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

## Request

This operation does not accept a request body.

**Deleted:** a

**Deleted:** a

**Deleted:** Any and all

**Deleted:** not

**Deleted:** does not have

**Deleted:** This

**Deleted:** :

**Deleted:** -

**Deleted:** due to

**Deleted:** above

**Deleted:** has

### Response

This operation does not return a response body.

# Health monitors

The Load Balancers service includes a health monitoring operation that periodically checks your back-end members to ensure that they are responding correctly. If a member does not respond, it is removed from rotation until the health monitor determines that the member is functional.

**Deleted:** Monitors

**Deleted:** load

**Deleted:** balancing

The health check also is performed against every member that is added to ensure that the member is operating correctly before it services traffic. Only one health monitor can be enabled on a load balancer at a time.

**Deleted:** In addition to being performed periodically, t

**Deleted:** properly

**Deleted:** allowing it to service

**Deleted:** is allowed to

## List health monitors

```
GET /v2.0/lbaas/healthmonitors
```

This operation lists all health monitors associated with your tenant account.

The following table shows the possible response codes for this operation.

| Response code | Name | Description |
|---|---|---|
| 200 | Success | The request succeeded. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** This operation returns a list, which might be empty, each element in the list is a health monitor that can contain the following attributes:¶
<#>id¶
<#>tenant_id¶
<#>type¶
<#>delay¶
<#>timeout¶
<#>max_retries¶
<#>http_method¶
<#>url_path¶
<#>expected_codes¶
<#>admin_state_up¶
<#>pool_id¶
<#>pools¶

Example: List health monitors¶
This

**Deleted:** :

**Deleted:** Code

**Deleted:** Request

**Deleted:** has

## Request

This operation does not accept a request body.

## Response

The following table shows the body parameters for the response.

| Parameter | Style | Type | Description |
|---|---|---|---|
| healthmonitor | plain | xsd:dict | A health monitor object. |
| id | plain | csapi:uuid | The UUID of the health monitor. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the health monitor. Only administrative users can specify a tenant UUID other than their own. |
| type | plain | xsd:string | The type of probe sent by the load balancer to verify the member state. Valid values are PING, TCP, HTTP, and HTTPS. |
| delay | plain | xsd:int | The time, in seconds, between sending probes to members. |
| timeout | plain | xsd:int | The maximum number of seconds for a monitor to wait for a connection to be established before it times out. This value must be less than the delay value. |
| max_retries | plain | xsd:int | The number of connection failures that are allowed before the status of the member is changed to INACTIVE. Valid values are from 1 to 10. |
| http_method | plain | xsd:string | The HTTP method that the monitor uses for requests. |

**Moved down [16]:** Example. List health monitors: JSON response¶

**Deleted:** This

**Deleted:** :

**Comment [KH62]:** Add rows for pool_id and pools.

**Deleted:** healthmonitor

**Deleted:** for

**Deleted:** healthmonitor

**Deleted:** A v

**Deleted:** is

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** or

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** allowed

**Deleted:** changing

**Deleted:** A v

**Deleted:** is

| Parameter | Style | Type | Description |
|---|---|---|---|
| (optional) | | | |
| url_path (optional) | plain | xsd:string | The HTTP path of the request sent by the monitor to test the health of a member. A valid value is a string that begins with a forward slash (/). |
| expected_codes (optional) | plain | xsd:string | The list of HTTP status codes expected in response from the member to declare it healthy. Specify one of the following values:<br>• A single value, such as 200.<br>• A list, such as 200, 202.<br>• A range, such as 200-204. |
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the health monitor, which is up (true) or down (false). Set this attribute to false to create the monitor in an administratively down state. |
| status | plain | xsd:string | The status of the health monitor, which indicates whether the health monitor is operational. |

**Comment [KH62]:** Add rows for pool_id and pools.

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** listener

Example: List health monitors, JSON response

**Moved (insertion) [16]**

**Deleted:** .

**Deleted:** :

```
{
  "healthmonitors": [
    {
      "admin_state_up": true,
      "delay": 1,
      "expected_codes": "200,201,202",
      "http_method": "GET",
      "id": "0a9ac99d-0a09-4b18-8499-a0796850279a",
      "max_retries": 5,
      "pools": [
        {
          "id": "74aa2010-a59f-4d35-a436-60a6da882819"
        }
      ],
      "tenant_id": "6f3584d5754048a18e30685362b88411",
      "timeout": 1,
      "type": "HTTP",
      "url_path": "/index.html"
    }
  ]
}
```

# Create a health monitor

**Comment [KH63]:** URL for this section goes to the “Create a health monitor” section in the getting started guide.

```
POST /v2.0/lbaas/healthmonitors
```

This operation provisions a new health monitor based on the configuration defined in the request. After the request is validated and progress has started on the provisioning process, a response is returned. The response contains a unique identifier for the health monitor.

**Deleted:** object

**Deleted:** object

**Deleted:** object

The request must specify the following health monitor attributes, at a minimum:

**Deleted:** The caller of this operation

**Deleted:** these

- tenant_id
- type
- delay
- timeout
- max_retries
- pool_id

**Deleted:** . Only required if the caller has an administrative role and wants to create a health monitor for another tenant.

**Deleted:** . The type of health monitor. Must be one of TCP, HTTP, HTTPS

**Deleted:** . The interval in seconds between health checks.

**Deleted:** . The time in seconds that a health check times out.

**Deleted:** . Number of failed health checks before marked as OFFLINE.

**Deleted:** . The pool that this health monitor will monitor.

Some attributes receive default values if you omit them from the request, and are useful only when you specify a health monitor type of HTTP or HTTPS:

**Deleted:** only

**Deleted:** (S)

- http_method
- url_path
- expected_codes
- admin_state_up

**Deleted:** . Default is GET.

**Deleted:** . Default is /.

**Deleted:** . The expected http status codes to get from a successful health check. Default is 200.

**Deleted:** . Default is true.

If the request cannot be fulfilled because of insufficient data or data that is not valid, an HTTP `400 (Bad Request)` error response is returned with information regarding the nature of the

**Deleted:** due to

**Formatted:** Code Char, Font: (Default) Times New Roman, Font color: Auto failure in the response body. Failures in the validation process are non-recoverable and require you to correct the cause of the failure and resend the request.

**Deleted:** the caller

**Deleted:** POST

**Deleted:** again

Users with an administrative role can create health monitors on behalf of other tenants by specifying a tenant_id attribute different than their own.

**Deleted:** You can configure all documented features of the health monitor at creation time by specifying the additional elements or attributes in the request.¶

To create a health monitor, the load balancer to which it is attached must have an `ACTIVE` provisioning status.

**Deleted:** update

**Deleted:** being

**Formatted:** Code Char, Font: (Default) Times New Roman, Font color: Auto

The following table shows the possible response codes for this operation.

**Deleted:** Example: Create a health monitor¶ This

**Deleted:** :

| Response code | Name | Description |
|---|---|---|
| 201 | Created | The request was fulfilled, and a new resource was created. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 404 | Not Found | The requested item was not found. |
| 409 | Conflict | The request could not be completed because of a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is greater than the allowed limit. |
| 500 | Load Balancer Fault | The load balancer experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Deleted:** Code

**Deleted:** has been

**Deleted:** resulted in

**Deleted:** being

**Deleted:** due to

**Deleted:** above

**Deleted:** has

## Request

**Deleted:** Example. Create health monitor: JSON request¶

The following table shows the body parameters for the request.

| Parameter | Style | Type | Description |
|---|---|---|---|
| healthmonitor | plain | xsd:dict | A health monitor object. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the health monitor. Only administrative users can specify a tenant UUID other than their own. |
| type | plain | xsd:string | The type of probe sent by the load balancer to verify the member state. Valid values are `PING`, `TCP`, `HTTP`, and `HTTPS`. |
| delay | plain | xsd:int | The time, in seconds, between sending probes to members. |
| timeout | plain | xsd:int | The maximum number of seconds for a monitor to wait for a connection to be established before it times out. This value must be less than the `delay` value. |
| max_retries | plain | xsd:int | The number of connection failures that are allowed before the status of the member is changed to INACTIVE. Valid values are from 1 to 10. |
| http_method (optional) | plain | xsd:string | The HTTP method that the monitor uses for requests. |
| url_path (optional) | plain | xsd:string | The HTTP path of the request sent by the monitor to test the health of a member. A valid value is a string that begins with a forward slash (/). |
| expected_codes (optional) | plain | xsd:string | The list of HTTP status codes expected in response from the member to declare it healthy. Specify one of the following values:<br>• A single value, such as `200`.<br>• A list, such as `200, 202`. |

**Deleted:** This

**Deleted:** :

**Comment [KH64]:** Add a row for `pool_id`

**Deleted:** healthmonitor

**Deleted:** healthmonitor

**Deleted:** A valid value is

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** or

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** allowed

**Deleted:** changing

**Deleted:** A v

**Deleted:** is

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

| Parameter | Style | Type | Description |
|---|---|---|---|
| | | | - A range, such as 200-204. |
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the load balancer, which is up (true) or down (false). Set this attribute to false to create the listener in an administratively down state. |

**Comment [KH64]:** Add a row for pool_id

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

### Example: Create a health monitor, JSON request

```
{
  "healthmonitor": {
    "admin_state_up": true,
    "delay": "1",
    "expected_codes": "200,201,202",
    "http_method": "GET",
    "max_retries": 5,
    "pool_id": "74aa2010-a59f-4d35-a436-60a6da882819",
    "timeout": 1,
    "type": "HTTP",
    "url_path": "/index.html"
  }
}
```

## Response

**Deleted:** Example. Create health monitor: JSON response¶

The following table shows the body parameters for the response.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| healthmonitor | plain | xsd:dict | A health monitor object. |
| id | plain | csapi:uuid | The UUID of the health monitor. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the health monitor. Only administrative users can specify a tenant UUID other than |

**Comment [KH65]:** Add rows for pool_id and pools.

**Deleted:** healthmonitor

**Deleted:** for

**Deleted:** healthmonitor

| Parameter | Style | Type | Description |
|---|---|---|---|
| | | | their own. |
| type | plain | xsd:string | The type of probe sent by the load balancer to verify the member state. Valid values are `PING`, `TCP`, `HTTP`, and `HTTPS`. |
| delay | plain | xsd:int | The time, in seconds, between sending probes to members. |
| timeout | plain | xsd:int | The maximum number of seconds for a monitor to wait for a connection to be established before it times out. This value must be less than the `delay` value. |
| max_retries | plain | xsd:int | The number of connection failures that are allowed before the status of the member changes to INACTIVE. Valid values are from 1 to 10. |
| http_method (optional) | plain | xsd:string | The HTTP method that the monitor uses for requests. |
| url_path (optional) | plain | xsd:string | The HTTP path of the request sent by the monitor to test the health of a member. A valid value is a string that begins with a forward slash (/). |
| expected_codes (optional) | plain | xsd:string | The list of HTTP status codes expected in response from the member to declare it healthy. Specify one of the following values:<br>• A single value, such as `200`.<br>• A list, such as `200, 202`.<br>• A range, such as `200-204`. |
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the health monitor, which is up (true) or down (false). Set this attribute to `false` to create the listener in an administratively down state. |

**Comment [KH65]:** Add rows for `pool_id` and `pools`.

**Deleted:** A valid value is

**Deleted:** or

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Deleted:** allowed

**Deleted:** changing

**Deleted:** A v

**Deleted:** is

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

**Formatted:** Code Char, Font: +Body (Calibri)

| Parameter | Style | Type | Description |
|---|---|---|---|
| status | plain | xsd:string | The status of the health monitor, which indicates whether the health monitor is operational. |

**Comment [KH65]:** Add rows for pool_id and pools.

```
{
  "healthmonitor": {
    "admin_state_up": true,
    "delay": 1,
    "expected_codes": "200,201,202",
    "http_method": "GET",
    "id": "0a9ac99d-0a09-4b18-8499-a0796850279a",
    "max_retries": 5,
    "pools": [
      {
        "id": "74aa2010-a59f-4d35-a436-60a6da882819"
      }
    ],
    "tenant_id": "6f3584d5754048a18e30685362b88411",
    "timeout": 1,
    "type": "HTTP",
    "url_path": "/index.html"
  }
}
```

# Show health monitor details

```
GET /v2.0/lbaas/healthmonitors/{healthmonitor_id}
```

This operation returns details about the specified health monitor. If the user is not an administrative user and the health monitor object does not belong to the user's tenant account, the service returns the HTTP Forbidden (403) response code.

The following table shows the possible response codes for this operation.

| Response code | Name | Description |
|---|---|---|

**Deleted:** a

**Deleted:** object identified by healthmonitor_id

**Deleted:** her

**Deleted:** If this operation succeeds, it returns a health monitor element that can contain the following attributes:¶
<#>id¶
<#>tenant_id¶
<#>type¶
<#>delay¶
<#>timeout¶
<#>max_retries¶
<#>http_method¶
<#>url_path¶
<#>expected_codes¶
<#>admin_state_up¶
<#>pool_id¶
<#>pools¶

Example: Show health monitor details¶
This

**Deleted:** :

**Comment [KH66]:** Make the same edits as indicated for other response code tables.

**Deleted:** Code

| Response code | Name | Description |
|---|---|---|
| 200 | Success | Request succeeded. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 403 | Forbidden | The server understood the request, but is refusing to fulfill it. |
| 404 | Not Found | The requested item was not found. |
| 409 | Conflict | The request could not be completed due to a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is above the allowed limit. |
| 500 | Load Balancer Fault | The load balancer has experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Comment [KH66]:** Make the same edits as indicated for other response code tables.

**Deleted:** Code

## Request

This operation does not accept a request body.

## Response

**Deleted:** Example. Show health monitor details: JSON response¶

The following table shows the body parameters for the response.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|

**Comment [KH67]:** Make the same edits as indicated in previous tables, including adding rows for `pools` and `pool_id`.

| Parameter | Style | Type | Description |
|---|---|---|---|
| healthmonitor | plain | xsd:dict | A healthmonitor object. |
| id | plain | csapi:uuid | The UUID for the health monitor. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the healthmonitor. Only administrative users can specify a tenant UUID other than their own. |
| type | plain | xsd:string | The type of probe sent by the load balancer to verify the member state. A valid value is PING, TCP, HTTP, or HTTPS. |
| delay | plain | xsd:int | The time, in seconds, between sending probes to members. |
| timeout | plain | xsd:int | The maximum number of seconds for a monitor to wait for a connection to be established before it times out. This value must be less than the delay value. |
| max_retries | plain | xsd:int | The number of allowed connection failures before changing the status of the member to INACTIVE. A valid value is from 1 to 10. |
| http_method (optional) | plain | xsd:string | The HTTP method that the monitor uses for requests. |
| url_path (optional) | plain | xsd:string | The HTTP path of the request sent by the monitor to test the health of a member. A valid value is a string that begins with a forward slash (/). |
| expected_codes (optional) | plain | xsd:string | The list of HTTP status codes expected in response from the member to declare it healthy. Specify one of the following values:<br>A single value, such as 200.<br>A list, such as 200, 202. |

**Comment [KH67]:** Make the same edits as indicated in previous tables, including adding rows for `pools` and `pool_id`.

| Parameter | Style | Type | Description |
|---|---|---|---|
| | | | A range, such as 200-204. |
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the health monitor, which is up (true) or down (false). Set this attribute to false to create the listener in an administratively down state. |
| status | plain | xsd:string | The status of the health monitor, which indicates whether the health monitor is operational. |

**Comment [KH67]:** Make the same edits as indicated in previous tables, including adding rows for `pools` and `pool_id`.

Example: Show health monitor details, JSON response

```
{
  "healthmonitor": {
    "admin_state_up": true,
    "delay": 1,
    "expected_codes": "200,201,202",
    "http_method": "GET",
    "id": "0a9ac99d-0a09-4b18-8499-a0796850279a",
    "max_retries": 5,
    "pools": [
      {
        "id": "74aa2010-a59f-4d35-a436-60a6da882819"
      }
    ],
    "tenant_id": "6f3584d5754048a18e30685362b88411",
    "timeout": 1,
    "type": "HTTP",
    "url_path": "/index.html"
  }
}
```

## Update a health monitor

```
PUT /v2.0/lbaas/healthmonitors/{healthmonitor_id}
```

This operation enables you to change one or more of the following health monitor attributes:

- delay
- timeout
- max_retries
- http_method
- url_path
- expected_codes
- admin_state_up

The health monitor ID, tenant_id, pool_id, and type are immutable attributes and cannot be updated. If you specify an unsupported attribute, the service returns the HTTP Immutable (422) response code. If the request is successful, the service returns the HTTP `Accepted (202)` response code.

You can update a health monitor only if the attached load balancer has a provisioning_status value of ACTIVE.

The following table shows the possible response codes for this operation.

| Response Code | Name | Description |
|---|---|---|
| 200 | Success | Request succeeded. |
| 400 | Bad Request | The request is missing one or more elements, or the values of some elements are invalid. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 413 | Over Limit | The number of items returned is above the allowed limit. |
| 500 | Load Balancer Fault | The load balancer has experienced a fault. |

**Formatted:** Space After: 0 pt

**Deleted:** Upon successful validation of the request, the service returns the HTTP Accepted (202) response code.
¶
The update

**Formatted:** Code Char, Font: (Default) Times New Roman, Font color: Auto

**Deleted:** This

**Deleted:** :

**Comment [KH68]:** Make the same edits as indicated in previous response codes tables.

| Response Code | Name | Description |
|---|---|---|
| 503 | Service Unavailable | The service is not available. |

**Comment [KH68]:** Make the same edits as indicated in previous response codes tables.

## Request

**Moved down [17]:** Example. Update health monitor: JSON request¶

The following table shows the body parameters for the request.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| healthmonitor | plain | xsd:dict | A healthmonitor object. |
| delay (optional) | plain | xsd:int | The time, in seconds, between sending probes to members. |
| timeout (optional) | plain | xsd:int | The maximum number of seconds for a monitor to wait for a connection to be established before it times out. This value must be less than the delay value. |
| max_retries | plain | xsd:int | The number of allowed connection failures before changing the status of the member to INACTIVE. A valid value is from 1 to 10. |
| http_method (optional) | plain | xsd:string | The HTTP method that the monitor uses for requests. |
| url_path | plain | xsd:string | The HTTP path of the request sent by the monitor to test the health of a member. A valid value is a string that begins with a forward slash (/). |
| expected_codes (optional) | plain | xsd:string | The list of HTTP status codes expected in response from the member to declare it healthy. Specify one of |

| Parameter | Style | Type | Description |
|---|---|---|---|
| | | | the following values:<br>A single value, such as 200.<br>A list, such as 200, 202.<br>A range, such as 200-204. |
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the health monitor, which is up (true) or down (false). Set this attribute to false to create the listener in an administratively down state. |

Example: Update a health monitor, JSON request

**Moved (insertion) [17]**

**Deleted:** .

**Deleted:** :

```
{
  "healthmonitor": {
    "admin_state_up": false,
    "delay": "2",
    "expected_codes": "200",
    "http_method": "POST",
    "max_retries": 2,
    "timeout": 2,
    "url_path": "/page.html"
  }
}
```

## Response

**Moved down [18]:** Example. Update health monitor: JSON response¶

The following table shows the body parameters for the response.

**Deleted:** This

**Deleted:** :

| Parameter | Style | Type | Description |
|---|---|---|---|
| healthmonitor | plain | xsd:dict | A healthmonitor object. |

**Comment [KH69]:** Make the same edits as indicated in previous tables, including adding rows for pools and pool_id.

| Parameter | Style | Type | Description |
|---|---|---|---|
| id | plain | csapi:uuid | The UUID for the health monitor. |
| tenant_id | plain | csapi:uuid | The UUID of the tenant who owns the healthmonitor. Only administrative users can specify a tenant UUID other than their own. |
| type | plain | xsd:string | The type of probe sent by the load balancer to verify the member state. A valid value is PING, TCP, HTTP, or HTTPS. |
| delay | plain | xsd:int | The time, in seconds, between sending probes to members. |
| timeout | plain | xsd:int | The maximum number of seconds for a monitor to wait for a connection to be established before it times out. This value must be less than the delay value. |
| max_retries | plain | xsd:int | The number of allowed connection failures before changing the status of the member to INACTIVE. A valid value is from 1 to 10. |
| http_method (optional) | plain | xsd:string | The HTTP method that the monitor uses for requests. |
| url_path (optional) | plain | xsd:string | The HTTP path of the request sent by the monitor to test the health of a member. A valid value is a string that begins with a forward slash (/). |
| expected_codes (optional) | plain | xsd:string | The list of HTTP status codes expected in response from the member to declare it healthy. Specify one of the following values:<br>• A single value, such as 200.<br>• A list, such as 200, 202.<br>• A range, such as 200-204. |

**Comment [KH69]:** Make the same edits as indicated in previous tables, including adding rows for `pools` and `pool_id`.

**Formatted:** Indent: Left: 0.02"

| Parameter | Style | Type | Description |
|---|---|---|---|
| admin_state_up (optional) | plain | xsd:boolean | The administrative state of the health monitor, which is up (true) or down (false). Set this attribute to false to create the listener in an administratively down state. |
| status | plain | xsd:string | The status of the health monitor, which indicates whether the health monitor is operational. |

**Comment [KH69]:** Make the same edits as indicated in previous tables, including adding rows for `pools` and `pool_id`.

Example: Update a health monitor, JSON response

**Moved (insertion) [18]**

**Deleted:** .

**Deleted:** :

```
{
  "healthmonitor": {
    "admin_state_up": false,
    "delay": 2,
    "expected_codes": "200",
    "http_method": "POST",
    "id": "0a9ac99d-0a09-4b18-8499-a0796850279a",
    "max_retries": 2,
    "pools": [
      {
        "id": "74aa2010-a59f-4d35-a436-60a6da882819"
      }
    ],
    "tenant_id": "6f3584d5754048a18e30685362b88411",
    "timeout": 2,
    "type": "HTTP",
    "url_path": "/page.html"
  }
}
```

# Remove a health monitor

```
DELETE /v2.0/lbaas/healthmonitors/{healthmonitor_id}
```

This operation removes the specified health monitor and its associated configuration from the tenant account.

**Deleted:** a

All configuration data is immediately purged and cannot be recovered.

**Deleted:** Any and all

You can delete a health monitor only if the attached load balancer has a provisioning_status value of ACTIVE.

**Deleted:** not

**Deleted:** does not have

The following table shows the possible response codes for this operation.

**Deleted:** Example: Delete a health monitor¶
This

**Deleted:** :

| Response Code | Name | Description |
|---|---|---|
| 204 | No Content | The server has fulfilled the request but does not need to return an entity-body. |
| 400 | Bad Request | The request is missing one or more elements, or the values of some elements are invalid. |
| 401 | Unauthorized | You are not authorized to complete this operation. This error can occur if the request is submitted with an invalid authentication token. |
| 409 | Conflict | The request could not be completed due to a conflict with the current state of the resource. |
| 413 | Over Limit | The number of items returned is above the allowed limit. |
| 500 | Load Balancer Fault | The load balancer has experienced a fault. |
| 503 | Service Unavailable | The service is not available. |

**Comment [KH70]:** Use the same edits as indicted in previous response code tables.

## Request

This operation does not accept a request body.

## Response

This operation does not return a response body.

# Release Notes

Learn about new features, enhancements, known issues, resolved issues, and other important details about Rackspace Cloud Load Balancers API 2.0 service updates.

For information about using the API, see the documentation overview.

## v2.xx.xx, xx x, 2016

**Comment [KH71]:** See the guidelines at https://github.com/rackerlabs/docs-rackspace/blob/master/style-guide/release-notes-guidelines.md#structure to determine what this heading should look like.

### What’s new

This is the initial Early Access (EA) release of the Rackspace Cloud Load Balancers API, v2.

### Resolved issues

None for this release.

### Known issues

None for this release.