API

Load Balancers

Load Balancers. From the perspective of the user, this is the root of the object tree.

A Load Balancer can have the following statuses:

`Status`	**Description**
`DEFERRED`	An entity has been created but is not yet linked to a load balancer. This is not a functioning state.
`PENDING_CREATE`	An entity is being created, but it is not yet functioning.
`PENDING_UPDATE`	An entity has been updated. It does remain in a functioning state.
`PENDING_DELETE`	An entity is in the process of being deleted.
`ACTIVE`	An entity is in a normal functioning state.
`INACTIVE`	Applies to members that fail health checks.
`ERROR`	Something has gone wrong. This may be either a functioning or non-functioning state.

Create

```
POST /loadbalancers
```

Description: Create a Load Balancer. TBD: Status of load balancer that is just created but has no listeners, since listeners cannot be assigned during this creation.

Parameters:

- **name:** Name of the load balancer.
- **description:** (optional) Detailed description of the load balancer.
- **vip_subnet_id*:*** subnet from which to allocate a virtual IP address.
- **vip_address:** (optional) ip address to assign to VIP.
- **tenant_id:** tenant will own this load balancer.
- **admin_state_up:** boolean that defines whether an active load balancer is functioning or not

Example request body:

```
{"loadbalancer" :
  {
    "name": "Example LB",
    "description": "A very simple example load balancer.",
    "vip_subnet_id": "SUBNET_ID",
    "tenant_id": "7725fe12-1c14-4f45-ba8e-44bf01763578"
  }
}
```

Example response body:

```
{"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": "PENDING_CREATE"
  }
}
```

Retrieve

`GET /loadbalancers`

Description: Lists all Load Balancers.

Example Request body:

None

Example response body:

```
{
  "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"
    },
  ]
}
```

`GET /loadbalancers/{load_balancer_id}`
Description: Shows the details of the configuration of a specific Load Balancer
Example Request Body:
None

Example Response Body:

```
{"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
`PUT /loadbalancers/{load_balancer_id}`
Description: Updates the alterable attributes of an existing Load Balancer.

Parameters:

- **name:** (optional) Name
- **description:** (optional) Detailed description of the load balancer.
- **admin_state_up:** (optional)

Request Body:

```
{"loadbalancer" :
  {
    "name": "A New Example LB",
    "description": "A new very simple example load balancer."
  }
}
```

Response Body:

```
{"loadbalancer":
  {
    "id": "8992a43f-83af-4b49-9afd-c2bfbd82d7d7",
    "name": "A New Example LB",
    "description": "A new 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"
  }
}
```

Delete

`DELETE /loadbalancers`

Disallowed

`DELETE /loadbalancers/{load_balancer_id}`

Description: Deletes a Load Balancer. Any Listeners that this Load Balancer is using will be detached but not deleted.

Example Request Body:

None

Example Response Body:

None

Stats

`GET /loabalancer/{load_balancer_id}/stats`

Description: provides metrics on a loadbalancer

Example Request Body:
None

Example Response Body:

```
{
    "active_connections": 0,
    "total_connections": 0,
    “bytes_in: 0,
    “bytes_out: 0
}
```

Listeners

Listeners represent a single listening port and can optionally provide TLS termination.

Create

`POST /listeners`

Description: Creates a listener.

Parameters:

- **name:** Name
- **description:** (optional) Detailed description of the listener.
- **loadbalancer_id:** (optional) ID of load balancer.
- **default_pool_id:** (optional) ID of default pool. Must have compatible protocol with listener.
- **connection_limit:** (optional) Maximum connections this load balancer can have. Default is infinite.
- **protocol:** Protocol to load balance: HTTP, HTTPS, TCP, UDP
- **protocol_port:** TCP (or UDP) port to listen on
- **tenant_id:** Tenant that owns this listener.
- **admin_state_up**: (optional) If set to false, listener will be created in an administratively down state

Example Request Body:

```
{
  "listener": {
    "default_pool_id": "8311446e-8a13-4c00-95b3-03a92f9759c7",
    "name": "Example HTTPS Listener",
    "description": "A very simple example of an HTTPS listener."
    "tenant_id": "352686b7-c4b2-44ec-a458-84239713f685",
    "connection_limit": 200,
    "protocol": "https",
    "protocol_port": "443",
    "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66",
  }
}
```

Example Response Body:

```
{
  "listener": {
    "id": "9bb09b2b-a0e3-4fa7-8020-76efd4b30d30",
    "default_pool_id": "8311446e-8a13-4c00-95b3-03a92f9759c7",
```

```
    "tenant_id": "352686b7-c4b2-44ec-a458-84239713f685",
    "name": "Example HTTPS Listener",
    "description": "A very simple example of an HTTPS listener."
    "protocol": "https",
    "protocol_port": "443",
    "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66",
    "admin_state_up": true,
    "status": "ACTIVE"
  }
}
```

Retrieve

`GET /listeners`

Description: Lists all listeners

Example Request body:

None

Example response body:

```
{
  "listeners": [
    {
      "id": "fddd4b15-faca-4a60-af4c-43317b9ad363",
      "tenant_id": "783b31af-6635-48b2-a807-091d9973e3a9",
      "name": "Example HTTPS Listener",
      "description": "A very simple example of an HTTPS listener.",,
      "protocol":    "https",
      "protocol_port": "443",
      "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66",
      "admin_state_up": true,
      "status": "ACTIVE"
    },
    {
      "id": "9341aafc-7e7d-4807-8199-9e81dadd0577",
      "tenant_id": "783b31af-6635-48b2-a807-091d9973e3a9",
      "name": "Example HTTPS Listener",
      "description": "A very simple example of an HTTPS listener.",,
      "protocol":    "https",
      "protocol_port": "443",
      "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66",
       "admin_state_up": true,
      "status": "ACTIVE"
```

```
    }
  ]
}
```

`GET /listeners/{listener_id}`
Description: Get configuration information for a specific listener

Example Request body:
None

Example response body:
```
{
  "listener": {
    "id": "9bb09b2b-a0e3-4fa7-8020-76efd4b30d30",
    "default_pool_id": "8311446e-8a13-4c00-95b3-03a92f9759c7",
    "tenant_id": "352686b7-c4b2-44ec-a458-84239713f685",
    "name": "Example HTTPS Listener",
    "description": "A very simple example of an HTTPS listener.",
    "protocol": "https",
    "protocol_port": "443",
    "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66",
    "admin_state_up": true,
    "status": "ACTIVE"
  }
}
```

Update
`PUT /listeners/{listener_id}`

Description: Updates alterable attributes of an existing listener.

Parameters:
- **name:** (optional) Name
- **description:** (optional) Detailed description of the listener.
- **default_pool_id:** (optional if not already assigned) ID of default pool. Must have compatible protocol with listener.
- **loadbalancer_id:** (optional if not already assigned) ID of existing load balancer.
- **admin_state_up:** (optional) enabled or disabled

Example Request body:
```
{"listener":
```

```
 {
   "name": ""A New Example HTTPS Listener""
 }
}
```

Example response body:

```
{
  "listener": {
    "id": "9bb09b2b-a0e3-4fa7-8020-76efd4b30d30",
    "default_pool_id": "8311446e-8a13-4c00-95b3-03a92f9759c7",
    "tenant_id": "352686b7-c4b2-44ec-a458-84239713f685",
    "name": "A New Example HTTPS Listener",
    "description": "A very simple example of an HTTPS listener.",
    "protocol": "https",
    "protocol_port": "443",
    "load_balancer_id": "b8a35470-f65d-11e3-a3ac-0800200c9a66",
    "admin_state_up": true,
    "status": "ACTIVE"
  }
}
```

Delete

`DELETE /listeners`

Disallowed

`DELETE /listeners/{listener_id}`

Description: Deletes an orphaned listener. Will return an error if listener is still in use by any load balancers. Any pools that this listener is using will be detached but not deleted.

Example Request Body:

None

Example Response Body:

None

Pools

Pools are groupings of backend member servers to which client requests are forwarded.

Create

`POST /pools`

Description: Creates a pool.

Parameters:

- **name:** Name
- **description:** (optional) description of a pool
  - Default: ""
- **tenant_id:** tenant that owns this
- **protocol:** Protocol use to connect to members: HTTP, HTTPS, TCP
- **session_persistence**: (optional) Session persistence algorithm that should be used (if any). This is a dictionary that has keys of "type" and "cookie_name".
  - Default: {}
- **lb_algorithm:** round-robin, least-connections, etc. (load balancing provider dependent, but round-robin must be supported).
- **healthmonitor_id:** (optional) ID of existing health monitor.
  - Default: null
- **admin_state_up:** (optional) enabled or disabled
  - Default: true

Example Request Body:

```
{
  "pool": {
    "name": "Example pool",
    "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",
    "protocol": "HTTP",
    "lb_algorithm": "ROUND_ROBIN"
  }
}
```

Example Response Body:

```
{
  pool: {
    "id": "332abe93-f488-41ba-870b-2ac66be7f853",
    "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",
    "name": "Example pool",
    "description": "",
```

```
    "protocol": "tcp",
    "lb_algorithm": "ROUND_ROBIN",
    "session_persistence": {},
    "healthmonitor_id": null,
    "members": [],
    "admin_state_up": true,
    "status": "ACTIVE"
  }
}
```

Retrieve

`GET /pools`

Description: Lists all pools

Example Request body:

None

Example response body:

```
{
  "pools": [
    {
         "id": "332abe93-f488-41ba-870b-2ac66be7f853",
         "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",
         "name": "Example pool",
         "description": "",
         "protocol": "tcp",
         "lb_algorithm": "ROUND_ROBIN",
         "session_persistence": {},
         "healthmonitor_id": null,
         "members": [],
         "admin_state_up": true,
         "status": "ACTIVE"
    },
    {
         "id": "332abe93-f488-41ba-870b-2ac66be7f854",
         "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",
         "name": "Example pool",
         "description": "",
         "protocol": "tcp",
         "lb_algorithm": "ROUND_ROBIN",
         "session_persistence": {},
         "healthmonitor_id": null,
```

```
        "members": [],
        "admin_state_up": true,
        "status": "ACTIVE"
    }
  ]
}
```

`GET /pools/{pool_id}`
Description: Get configuration information for a specific pool

Example Request body:
None

Example response body:
```
{
  pool: {
    "id": "332abe93-f488-41ba-870b-2ac66be7f853",
    "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",
    "name": "Example pool",
    "description": "",
    "protocol": "tcp",
    "lb_algorithm": "ROUND_ROBIN",
    "session_persistence": {},
    "healthmonitor_id": null,
    "members": [],
    "admin_state_up": true,
    "status": "ACTIVE"
  }
}
```

Update
`PUT /pools/{pool_id}`

Description: Updates alterable attributes of an existing pool.

Parameters
- **name:** Name
- **description:** description of the pool
- **session_persistence**: Session persistence algorithm that should be used (if any). Includes “SOURCE_IP”, “COOKIE”, etc.

- **lb_algorithm:** round-robin, least-connections, etc. (load balancing provider dependent, but round-robin must be supported).
- **healthmonitor_id:** (only if not already assigned) id of healthmonitor id
- **admin_state_up:** enabled or disabled

Example Request body:

```
{
  "pool": {
    "lb_algorithm": "LEAST_CONNECTIONS",
    "name": "New pool name",
  }
}
```

Example response body:

```
{
  "pool": {
    "id": "332abe93-f488-41ba-870b-2ac66be7f853",
    "tenant_id": "19eaa775-cf5d-49bc-902e-2f85f668d995",
    "name": "New pool name",
    "description": "",
    "protocol": "tcp",
    "lb_algorithm": "LEAST_CONNECTIONS",
    "session_persistence": {},
    "healthmonitor_id": null,
    "members": [],
    "admin_state_up": true,
    "status": "ACTIVE"
  }
}
```

Delete

`DELETE /pools`

Disallowed

`DELETE /pools/{pool_id}`

Description: Deletes an orphaned pool. Will return an error if pool is still in use by any listeners. Upon successful deletion, any child primitives that this pool is using will be detached but not deleted.

Example Request Body:

None

Example Response Body:
None

Members

Members are back-end server nodes that process client requests.

Create

`POST /pools/{pool_id}/members`

Description:
Members are individual backend services which are being load balanced. Usually these would be web application servers. They are represented as a pool, IP address, Layer 4 port tuple.

Parameters:

- **address:** IP address of pool member
- **protocol_port:** TCP (or UDP) port
- **weight:** (optional) positive integer indicating relative portion of traffic from pool this member should receive (e.g. a member with a weight of 10 will receive five times as much traffic as a member with weight 2)
    - Default: 1
- **admin_state_up**: (optional) If set to false, member will be created in an administratively down state
    - Default: true
- **tenant_id:** tenant to which this member is owned
- **subnet_id:** subnet in which to access this member

Example Request Body:

```
{
  member: {
    "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
    "address": "192.0.2.14",
    "protocol_port": 8080,
    "subnet_id": "SUBNET_ID"
  }
}
```

Example Response Body:

```
{
  member: {
    "id": "975592ca-e308-48ad-8298-731935ee9f45",
    "address": "192.0.2.14",
    "protocol_port": 8080,
    "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
    "admin_state_up": true,
    "weight": 1,
```

```
    "subnet_id": "SUBNET_ID",
    "status": "DOWN"
  }
}
```

Retrieve

`GET /pools/{pool_id}/members`

Description: Lists all members on a pool

Example Request body:

None

Example Response body:

```
{
  "members": [
    {
       "id": "975592ca-e308-48ad-8298-731935ee9f45",
       "address": "192.0.2.14",
       "protocol_port": 8080,
       "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
       "admin_state_up": true,
       "weight": 1,
       "subnet_id": "SUBNET_ID",
       "status": "DOWN"
    },
    {
       "id": "975592ca-e308-48ad-8298-731935ee9f46",
       "address": "192.0.2.14",
       "protocol_port": 8080,
       "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
       "admin_state_up": true,
       "weight": 1,
       "subnet_id": "SUBNET_ID",
       "status": "DOWN"
    },
  ]
}
```

`GET /pools/{pool_id}/members/{member_id}`

Description: Get configuration information for a specific member

Example Request body:

None

Example response body:

```
{
  member: {
     "id": "975592ca-e308-48ad-8298-731935ee9f45",
     "address": "192.0.2.14",
     "protocol_port": 8080,
     "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
     "admin_state_up": true,
     "weight": 1,
     "subnet_id": "SUBNET_ID",
     "status": "DOWN"
  }
}
```

Update

```
PUT /pools/{poolid}/members/{member_id}
```

Description: Updates alterable attributes of an existing member.

Parameters:
weight: Weight of member
admin_state_up: enabled or disabled

Example Request body:

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

Example Response body:

```
{
  member: {
     "id": "975592ca-e308-48ad-8298-731935ee9f45",
     "address": "192.0.2.14",
     "protocol_port": 8080,
     "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
     "admin_state_up": false,
     "weight": 1,
     "subnet_id": "SUBNET_ID",
```

```
      "status": "DOWN"
   }
}
```

Delete

```
DELETE /members
```

Disallowed

```
DELETE /pools/{pool_id}/members/{member_id}
```

Description: Deletes a member from a pool.

Example Request Body:
None
Example Response Body:
None

Health Monitors

Health monitors describe the probes that are used to determine the health of pool members

Create

```
POST /healthmonitors
```

Description: Creates a health monitor.

Parameters:

- **type:** Protocol used for health monitor: HTTP, HTTPS, TCP, PING
- **tenant_id:** tenant that owns this healthmonitor
- **delay**: Time in seconds between probes
- **timeout:** Time in seconds to timeout each probe
- **max_retries**: Maximum consecutive health probe tries
- **http_method:** (optional) http method monitor uses to make request
    - Default: "GET"
- **url_path**: (optional) Path portion of URI that will be probed if type is HTTP(S)
    - Default: “/”
- **expected_error_codes:** (optional) Expected HTTP codes for a passing HTTP(S) monitor
    - Default: "200"
- **admin_state_up:** (optional) enabled or disabled

- "Default": true

Example Request Body:

```
{
  "healthmonitor": {
    "type" : "HTTP",
    "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
    "delay" : 20,
    "timeout": 10,
    "max_retries: 5,
    "url_path" : "/check",
    "expected_codes" : "200-299"
  }
}
```

Example Response Body:

```
{
  "healthmonitor": {
     "id" : "f3eeab00-8367-4524-b662-55e64d4cacb5",
     "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
     "type" : "HTTP",
     "delay" : 20,
     "timeout": 10,
     "max_retries": 5,
     "http_method": "GET",
     "url_path" : "/check",
     "expected_codes" : "200-299",
     "admin_state_up": true,
     "status": "ACTIVE"
   }
}
```

Retrieve

`GET /healthmonitors`

Description: Lists all health monitors

Example Request body:
None

Example response body:

```
{
```

```
  "healthmonitors": [
    {
     "id" : "f3eeab00-8367-4524-b662-55e64d4cacb5",
     "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
     "type" : "HTTP",
     "delay" : 20,
     "timeout": 10,
     "max_retries": 5,
     "http_method": "GET",
     "url_path" : "/check",
     "expected_codes" : "200-299",
     "admin_state_up": true,
     "status": "ACTIVE"
    },
  ]
}
```

```
GET /healthmonitors/{healthmonitor_id}
```

Description: Get configuration information for a specific health monitor

Example Request body:

None

Example response body:

```
{
  "healthmonitor": {
     "id" : "f3eeab00-8367-4524-b662-55e64d4cacb5",
     "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
     "type" : "HTTP",
     "delay" : 20,
     "timeout": 10,
     "max_retries": 5,
     "http_method": "GET",
     "url_path" : "/check",
     "expected_codes" : "200-299",
     "admin_state_up": true,
     "status": "ACTIVE"
   }
}
```

Update

```
PUT /healthmonitors/{healthmonitor_id}
```

Description: Updates alterable attributes of an existing health monitor.

Parameters:

- **delay**: Time in seconds between probes
- **timeout:** Time in seconds to timeout each probe
- **max_retries**: Maximum consecutive health probe tries
- **http_method:** http method monitor uses to make request
- **url_path**: Path portion of URI that will be probed if type is HTTP(S)
- **expected_codes:** Expected HTTP codes for a passing HTTP(S) monitor
- **admin_state_up:** enabled or disabled

Example Request body:

```
{
  "healthmonitor": {
        "url_path": "/healthcheck",
        "delay": 3,
  }
}
```

Example response body:

```
{
  "healthmonitor": {
     "id" : "f3eeab00-8367-4524-b662-55e64d4cacb5",
     "tenant_id": "453105b9-1754-413f-aab1-55f1af620750",
     "type" : "HTTP",
     "delay" : 3,
     "timeout": 10,
     "max_retries": 5,
     "http_method": "GET",
     "url_path" : "/healthcheck",
     "expected_codes" : "200-299",
     "admin_state_up": true,
     "status": "ACTIVE"
   }
}
```

Delete

```
DELETE /healthmonitors
```

Disallowed

```
DELETE /healthmonitors/{healthmonitor_id}
```

Description: Deletes an orphaned health monitor. Will return an error if it's is still in use by any pools.

Example Request Body:
None
Example Respone Body:
None