Load Balancing Service Manage

(Nov. 11, 2010)

rackspace ®

HOSTING

docs.rackspacecloud.com/api

Load Balancing Service Management API Specification

(11/11/10)

Table of Contents

List of Tables

1. Management API Operations

1.1. Listing, Creating, Updating, and Deleting Load Balancing Clusters

Table 1.1.

Verb	URI	Description	Access Level
GET	/clusters	Retrieve a list of all clusters	Support, Service Admin
POST	/clusters	Create a new cluster	Service Admin
PUT	/clusters/clusterId	Upate the name, description, username, or password of a cluster	Service Admin
DELETE	/clusters/clusterId	Remove a cluster	Service Admin

Normal Response Code(s): 200

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

1.1.1. Description

The cluster operations allow for listing and manipulating clusters. In order to remove a cluster, it may not have any host machines or virtual IP's associated with it. To create a new cluster, the caller must supply their username, password, cluster name, cluster description and datacenter attributes within the request.

The GET response contains numerous attributes (numberOfLoadBalancingConfigurations, numberOfUniqueCustomers, numberOfHostMachines, and utilization) that are generated and immutable. These attributes are calculated based on configurations associated with this cluster.

1.1.2. Sample XML Response (/clusters)

```
<clusters xmlns="http://docs.openstack.org/loadbalancers/api/mgmt/
v1.0">
    <cluster
        id="1"
        name="Cluster Alpha"
        description="The best cluster, ever."
        datacenter="DFW"
        numberOfLoadBalancingConfigurations="410"
        numberOfUniqueCustomers="348"
        numberOfHostMachines="8"
        utilization="59%" />
```

```xml
    <cluster
        id="2"
        name="Cluster Beta"
        description="The second best cluster, ever."
        datacenter="ORD"
        numberOfLoadBalancingConfigurations="580"
        numberOfUniqueCustomers="490"
        numberOfHostMachines="8"
        utilization="76%" />
</clusters>
```

1.1.3. Sample JSON Response (/clusters)

```json
{"clusters": {
        "cluster": [
            {
                "id": "1",
                "name": "Cluster Alpha",
                "description": "The best cluster, ever.",
                "dataCenter": "DFW",
                "numberOfLoadBalancingConfigurations": "410",
                "numberOfUniqueCustomers": "348",
                "numberOfHostMachines": "8",
                "utilization": "59%"
            },
            {
                "id": "1",
                "name": "Cluster Beta",
                "description": "The second best cluster, ever.",
                "dataCenter": "DFW",
                "numberOfLoadBalancingConfigurations": "580",
                "numberOfUniqueCustomers": "490",
                "numberOfHostMachines": "8",
                "utilization": "76%"
            }
        ]
    }
}
```

1.1.4. Sample XML Response (/clusters/clusterId)

```xml
<cluster
        id="1"
        name="Cluster Alpha"
        description="The best cluster, ever."
        datacenter="DFW"
        numberOfLoadBalancingConfigurations="410"
```

```
                numberOfUniqueCustomers="348"
                numberOfHostMachines="8"
                utilization="59%" />
```

1.1.5. Sample JSON Response (/clusters/clusterId)

```
{"cluster": {
        "id": "1",
        "name": "Cluster Alpha",
        "description": "The best cluster, ever.",
        "dataCenter": "DFW",
        "numberOfLoadBalancingConfigurations": "410",
        "numberOfUniqueCustomers": "348",
        "numberOfHostMachines": "8",
        "utilization": "59%"
    }
}
```

1.1.6. Sample XML Request (POST) (/clusters)

```
<cluster
        xmlns="http://docs.openstack.org/loadbalancers/api/
management/v1.0"
        name="Cluster Gamma"
        description="The third best cluster, ever."
        dataCenter="DFW"
        username="my username"
        password="my password"/>
```

1.1.7. Sample JSON POST Request (/clusters)

```
{"cluster": {
        "name": "Cluster Gamma",
        "description": "The third best cluster, ever.",
        "datacenter": "DFW",
        "username" : "my username",
        "password" : "my password"
    }
}
```

1.1.8. Sample XML PUT Request (/clusters/clusterid)

```
<cluster
```

```
        xmlns="http://docs.openstack.org/loadbalancers/api/
management/v1.0"
        name="Cluster Delta"
        description="The fourth best cluster, ever."
        username="aaa"/>
```

1.1.9. Sample JSON PUT Request (/clusters/clusterid)

```
{"cluster": {
        "name": "Cluster Delta",
        "description": "The fourth best cluster, ever.",
        "username": "aaa"
    }
}
```

1.2. Listing, Creating, Updating, and Deleting Load Balancing Host Machines

Table 1.2.

Verb	URI	Description	Access Level
GET	/hosts	Retrieve a list of all hosts within the region	Support, Service Admin
GET	/clusters/clusterId/ hosts	Retrieve a list of all hosts belonging to a cluster	Support, Service Admin
POST	/hosts	Create a new load balancing host	Service Admin
PUT	/hosts/hostId	Update the mutable attributes of a load balancing host	Service Admin
DELETE	/hosts/hostId	Remove a load balancing host	Service Admin
GET	/clusters/clusterId/ endpoint	Retrieves the host that is currently the active SOAP endpoint	Support, Service Admin
PUT	/hosts/hostId/ endpoint/enable	Enable this host's SOAP endpoint (Warning: can be overwritten by pollendpoints call)	Service Admin
PUT	/hosts/hostId/ endpoint/disable	Disable this host's SOAP endpoint	Service Admin

Verb	URI	Description	Access Level
		(Warning: can be overwritten by pollendpoints call)	
PUT	/clusters/ pollendpoints	tests all endpoints and sets soap_endpoint_active if endpoint works	Service Admin

1.2.1. Description

The host operations allow for retrieval of host configuration data and statistics as well as the ability to add, manipulate or delete load balancing hosts.

1.2.1.1. Creating a New Host

When creating a new host, the caller must supply the following attributes:

- name

- clusterid

- coreDeviceId

- zone

- maxConcurrentConnections

- managementIpAddress

- managementSoapInterface

Additionally, the system will attempt to validate that it can properly access the host based on the managementSoapInterface, managementSoapUsername, and managementSoapPassword. The system should automatically apply a BURN_IN status to all newly added host machines.
<node> For an example of how a new host is created, see the "Sample XML Post Request" below. </node>

1.2.1.2. Mutable Attributes

The following list of attributes are mutable via the PUT HTTP operation:

- name

- coreDeviceId

- status (Active, Active Target, Maintenance, Failover

- maxConcurrentConnections

- managementIpAddress

- managementSoapInterface

The GET response contains numerous attributes (utilization, numberOfLoadBalancingConfigurations, and numberOfUniqueCustomers) that are generated based on state data and are immutable. These attributes are calculated based on configurations associated with this host.

1.2.2. Sample XML Response (/hosts) (/clusters/clusterId/hosts)

```
<hosts xmlns="http://docs.openstack.org/loadbalancers/api/mgmt/
v1.0">
   <host
      id="1"
      name="host1"
      clusterId="1"
      coreDeviceId="14410"
      zone="A"
      status="ACTIVE_TARGET"
      maxConcurrentConnections="150000"
      managementIpAddress="10.1.1.1"
      managementSoapInterface="http://10.1.1.1:9090/soap"
      utilization="60%"
      numberOfLoadBalancingConfigurations="414"
      numberOfUniqueCustomers="141"
      soapEndpointActive="true"/>
   <host
      id="2"
      name="host2"
      clusterId="1"
      coreDeviceId="15510"
      zone="B"
      status="ACTIVE_TARGET"
      maxConcurrentConnections="150000"
      managementIpAddress="10.1.1.2"
      managementSoapInterface="http://10.1.1.2:9090/soap"
      utilization="59%"
      numberOfLoadBalancingConfigurations="520"
      numberOfUniqueCustomers="515"
      soapEndpointActive="true/>
</hosts>
```

1.2.3. Sample JSON Response (/hosts) (/clusters/clusterId/hosts)

```
{"hosts": {
      "host": [
         {
            "id": "1",
```

```
                "name": "host1",
                "clusterId": "1",
                "coreDeviceId": "14410",
                "zone": "A",
                "status": "ACTIVE_TARGET",
                "maxConcurrentConnections": "150000",
                "managementIpAddress": "10.1.1.1",
                "managementSoapInterface": "http://10.1.1.1:9090/
soap",
                "utilization": "60%",
                "numberOfLoadBalancingConfiguraions": "414",
                "numberOfUniqueCustomers": "141",
                "soapEndpointActive": "true"
            },
            {
                "id": "2",
                "name": "1",
                "clusterId": "1",
                "coreDeviceId": "15510-44140",
                "zone": "B",
                "status": "ACTIVE_TARGET",
                "maxConcurrentConnections": "150000",
                "managementIpAddress": "10.1.1.2",
                "managementSoapInterface": "httpd://10.1.1.2:9090/
soap",
                "utilization": "59%",
                "numberOfLoadBalancingConfigurations": "520",
                "numberOfUniqueCustomers": "515",
                "soapEndpointActive": "true"
            }
        ]
    }
}
```

1.2.4. Sample XML Request (POST) (/hosts)

```
<host xmlns="http://docs.openstack.org/loadbalancers/api/
management/v1.0"
        clusterId="1"
        coreDeviceId="SomeCoreDevice"
        managementIp="12.34.56.78"
        managementSoapInterface="https://SomeSoapNode.com:9090"
        maxConcurrentConnections="5"
        name="someName"
        status="BURN_IN"
        trafficManagerName="zues01.blah.blah"
        type="FAILOVER"
        zone="B"
        soapEndpointActive="true" />
```

1.2.5. Sample JSON Request (POST) (/hosts)

```json
{"host": {
        "status": "BURN_IN",
        "name": "someName",
        "zone": "B",
        "type": "FAILOVER",
        "managementIp": "12.34.56.78",
        "trafficManagerName": "zues01.blah.blah",
        "clusterId": 1,
        "maxConcurrentConnections": 5,
        "coreDeviceId": "SomeCoreDevice",
        "managementSoapInterface": "https://
SomeSoapNode.com:9090",
        "soapEndpointActive": "true"
    }
}
```

1.2.6. Sample XML PUT Request (/hosts/hostid)

```xml
<host xmlns="http://docs.openstack.org/loadbalancers/api/
management/v1.0"
        name="my-next-host"
        coreDeviceId="144410-44001"
        status="BURN_IN"
        maxConcurrentConnections="150000"
        managementIpAddress="10.1.1.2"
        managementSoapInterfac="http://10.1.1.2:9090/soap"
        soapEndpointActive="true" />
```

1.2.7. Sample JSON PUT Request (/hosts/hostid)

```json
{"host": {
        "name": "my-next-host",
        "coreDeviceId": "144410-44001",
        "status": "BURN_IN",
        "maxConcurrentConnections": "150000",
        "managementIpAddress": "10.1.1.2",
        "managementSoapInterface": "http://10.1.1.2:9090/soap",
        "soapEndpointActive": "true"
    }
}
```

1.3. Virtual IP Management

Table 1.3.

Verb	URI	Description	Access Level
GET	/virtualips	Retrieve a list of all virtual IPs associated to the region	Service Admin
GET	/clusters/clusterId/ virtualips	Retrieve a list of all virutal IPs associate to a cluster	Service Admin
GET	/virtualips/ virtualIpId/ loadbalancers	Retrieve a list of load balancers associated with this virtual IP	Service Admin
GET	/loadbalancers/ loadBalancerId/ virtualips	Retrieve a list of all virtual IPs associated with a load balancer	Service Admin
POST	/clusters/clusterId/ virtualips	Assign one-or-more additional virtual IPs to a cluster	Service Admin
POST	/clusters/clusterId/ virtualipblocks	Assign a set of IP blocks to the cluster	Service Admin
DELETE	/clusters/clusterId/ virtualips/virtualIpId	Remove the specified virtual IP from the cluster	Service Admin

1.3.1. Description

The virtual IP operations allow the caller to view, create, and remove virtual IPs from an environment. Virtual IPs are automatically assigned to every newly created load balancer and can be added on-demand by a support or service administrator with proper justification. Management of the service requires blocks of IP addresses to be allocated from time-to-time to ensure availability for customers.

In order to assign a virtual IP to the environment via the POST operation, the caller must supply the address, and type attributes as part of the virtualip element. A sample POST request can be found below.

Note
In the event a virtual IP must be removed from the cluster, the DELETE operation can be used; however, to delete a virtual IP it must not have a load balancer associated to it.

Note
To add a single or multiple specific virtual ip's simply specify the first and lastIp fields identically(not a range).

1.3.2. Sample XML Response (/virtualips) (/clusters/clusterId/virtualips)

```
<virtualips>
    <virtualip id="411"
        loadBalancerId="1"
        clusterId="1"
        address="98.128.33.1"
        type="PUBLIC" />
    <virtualip
        id="501"
        clusterId="1"
        address="10.41.133.4"
        type="SERVICENET" />
</virtualips>
```

1.3.3. Sample JSON Response (/virtualips) (/clusters/clusterId/virtualips)

```
{"virtualips": {
        "virtualip": [
            {
                "id": "411",
                "loadBalancerId": "1",
                "clusterId": "1",
                "address": "98.128.33.1",
                "type": "PUBLIC"
            },
            {
                "id": "501",
                "loadBalancerId": "1",
                "clusterId": "1",
                "address": "10.41.133.4",
                "type": "SERVICENET"
            }
        ]
    }
}
```

1.3.4. Sample XML POST Request (/clusters/clusterId/virtualips)

```
<virtualips>
```

```
          <virtualip
             address="98.128.33.4"
             type="PUBLIC" />
          <virtualip
             address="98.128.33.5"
             type="PUBLIC" />
</virtualips>
```

1.3.5. Sample XML Request (POST) (/clusters/clusterId/virtualipsblocks)

```
<virtualIpBlocks type="PUBLIC"
       xmlns="http://docs.openstack.org/loadbalancers/api/
management/v1.0"
       xmlns:ns2="http://docs.openstack.org/loadbalancers/api/
v1.0">
 <ns2:virtualIpBlock firstIp="10.0.0.5" lastIp="10.0.0.5"/>
 <ns2:virtualIpBlock firstIp="172.16.0.64" lastIp="172.16.0.64"/>
</virtualIpBlocks>
```

1.3.6. Sample JSON Request (POST) (/clusters/clusterId/virtualips)

```
{"virtualIpBlocks": [
        {
            "firstIp": "10.0.0.5",
            "lastIp": "10.0.0.5"
        },
        {
            "firstIp": "172.16.0.64",
            "lastIp": "172.16.0.64"
        },
        {
            "firstIp": "192.168.0.3",
            "lastIp": "192.168.0.3"
        }
    ],
    "type": "SERVICENET"
}
```

1.4. Rate Limiting a Load Balancer

Table 1.4.

Verb	URI	Description	Access Level
GET	/loadbalancers/ loadBalancerId/ ratelimit	Retrieve the rate limit information for a specified load balancer	Service Admin
POST	/loadbalancers/ loadBalancerId/ ratelimit	Impose a new rate limit on the specified load balancer	Service Admin
PUT	/loadbalancers/ loadBalancerId/ ratelimit	Modify the expiration time and/or ticket associated with the rate limit	Service Admin
DELETE	/loadbalancers/ loadBalancerId/ ratelimit	Immediately remove the defined rate limit	Service Admin
DELETE	/loadbalancers/ loadBalancerId/ ratelimit/ removeoldlimits	Immediately remove expired limits	Service Admin

1.4.1. Description

Rate limiting allows a service administrator to artificially limit the number of requests that are permitted to transit a given load balancer. This action can be taken when a particular load balancer is servicing questionable traffic, the victim of a DDoS, etc. To define a rate limit, the caller must supply a ticket id, maximum number of connections per second, and an expiration time.

Additionally, rate limit information is returned as part of the "extended details of a load balancer", which is documented below.

1.4.2. Sample XML Response (/loadbalancers/ loadBalancerId/ratelimit)

```
<rateLimit
    ticketId="44410"
    expirationTime="2010-10-17T00:00:00-05:00"
    maxRequestsPerSecond="150" />
```

1.4.3. Sample JSON Response (/loadbalancers/ loadBalancerId/ratelimit)

```
{"rateLimit": {
        "ticketId": "44410",
        "expirationTime": "2010-10-17 00:00:00",
        "maxRequestPerSecond": "150"
    }
}
```

1.4.4. Sample XML Request (POST) (/loadbalancers/ loadBalancerId/ratelimit)

```
<rateLimit xmlns="http://docs.openstack.org/loadbalancers/api/
management/v1.0"
    ticketId="44410"
    expirationTime="2010-10-17T00:00:00"
    maxRequestsPerSecond="150" />
```

1.4.5. Sample JSON Request (POST) (/loadbalancers/ loadBalancerId/ratelimit)

```
{"rateLimit": {
        "ticketId": "44411",
        "expirationTime": "2010-10-17 00:00:00",
        "maxRequestsPerSecond": "150"
    }
}
```

1.4.6. Sample XML Request (PUT) (/loadbalancers/ loadBalancerId/ratelimit)

```
<rateLimit xmlns="http://docs.openstack.org/loadbalancers/api/
management/v1.0"
    ticketId="44411"
    expirationTime="2010-10-18 00:00:00"
    maxRequestsPerSecond="150" />
```

1.4.7. Sample JSON Request (PUT) (/loadbalancers/ loadBalancerId/ratelimit)

```
{"rateLimit": {
        "ticketId": "44411",
        "expirationTime": "2010-10-17 00:00:00",
        "maxRequestsPerSecond": "150"
    }
}
```

1.5. Moving Load Balancers Between Host Machines / Adjusting Stickiness of Load Balancer Hosts

Table 1.5.

Verb	URI	Description	Access Level
PUT	/loadbalancers/ loadBalancerId/sticky	Make the specified load balancer sticky	Support, Service Admin
DELETE	/loadbalancers/ loadBalancerid/sticky	Turn off the "sticky" flag on the specified load balancer	Support, Service Admin
GET	loadbalancers/ loadBalancerId/hosts	Get the host assignment(s) for a specified load balancer	Service Admin
PUT	/loadbalancers/ loadBalancerId/hosts	Update the host assignment(s) for a specified load balancer	Service Admin
PUT	/loadbalancers/ reassignhosts	update the host assignment(s) for a list of load balancers	Service Admin

1.5.1. Description

Service administrators may re-assign a load balancer to a different host machine within the same cluster. This action can be taken if capacity warrants it or if a particular configuration needs to be isolated from others within the environment. Both active and failover hosts may be changed.

Additionally, this operation allows for a service administrator to define a load balancer's host configurations as being "sticky", which will prohibit the system from automatically moving this configuration between hosts to balance the host performance.

A load balancer that is defined as being ACTIVE on multiple hosts will allow the load balancer host machines to service traffic for a single VIP across multiple systems. This is an advanced feature that should be used cautiously as it can potentially amplify DDoS and other types of malicious traffic.

1.5.2. Sample XML Request (GET) (/loadbalancers/ loadBalancerId/host)

```
<host xmlns="http://docs.rackspacecloud.com/loadbalancers/api/
management/v1.0"
            id="1"
            name="host1"
            clusterId="1"
            coreDeviceId="10"
            maxConcurrentConnections="1500"
            managementIp="10.2.2.3"
            trafficManagerName="ztm-n01.dev.lbaas.rackspace.com"
            managementSoapInterface="https://173.203.239.70:9090/
soap" />
```

1.5.3. Sample XML Request (PUT) (/loadbalancers/ reassignhosts)

```
<loadBalancers xmlns="http://docs.rackspacecloud.com/
loadbalancers/api/management/v1.0">
   <loadBalancer id="49" >
     <host id="2"/>
   </loadBalancer>
</loadBalancers>
```

1.5.4. Sample JSON Request (PUT) (/loadbalancers/ loadBalancerId/hosts)

```
{"loadbalancers": {
        "loadbalancer": [
           {
               "id": "1"
           }
        ],
         "host": [
         {
           "id": "2"
         }
        ]
```

```
        }
}
```

1.5.5. Sample XML Response (/loadbalancers/reassignhosts)

```
<loadBalancers xmlns="http://docs.rackspacecloud.com/
loadbalancers/api/management/v1.0">
    <loadBalancer id="1" />
</loadBalancers>
```

1.5.6. Sample JSON Response (/loadbalancers/reassignhosts)

```
{"loadbalancers": {
        "loadbalancer": [
            {
                "id": "1"
            },
            {
                "id": "2"
            }
        ]
    }
}
```

1.6. Host machine Configuration Backup & Restoration

Table 1.6.

Verb	URI	Description	Access Levels
GET	/hosts/backups	View a list of available backups for all host machines	Service Admin
GET	/hosts/hostId/ backups	View a list of available backups for a specified host machine	Service Admin
GET	/hosts/hostId/ subnetmappings	View the list of subnet mappings for this host	Service Admin
PUT	/hosts/hostId/ subnetmappings	Add subnet mappings on this host	Service Admin

Verb	URI	Description	Access Levels
POST	/host/hostId/ subnetmappings/ delete	Delete the specified subnet mappings on this host	Service Admin
PUT	/hosts/hostId/ activate	Activate Host Machine	Service Admin
PUT	/hosts/hostId/ inactive	Deactivate Host Machine	Service Admin
PUT	/hosts/hostId/ backups	Create a new backup of the host configuration	Service Admin
DELETE	/hosts/hostId/ backups/backupId	Purge a backup with the specified backup ID	Service Admin
PUT	/hosts/hostId/ backups/backupId/ restore	Restore the specified backup to the host machine	Service Admin

1.6.1. Description

The host backup and restoration tools allow for service administrators to take periodic backups of the current state of the configuration on any given host machine. These functions allow the caller to view, create new backups, purge backups, and restore any available configuration.

1.6.2. Non mutable Parameters:

• backupTime

• hostId

• id

Callers are not required to supply request bodies for the PUT (Restore Backup) operations.

1.6.3. Creating a New Backup

When creating a new backup, the caller must supply the following attributes:

• name

1.6.4. Sample XML Response (/hosts/backups)

```
<backups>
    <backup
        id="1"
        name="NightlyBackUp"
        backupTime="2010-10-17 00:00:00"
```

```
          hostId="1234" />
</backups>
```

1.6.5. Sample JSON Response (/hosts/backups)

```
{"backups": {
        "backup": {
            "id": "1",
            "name": "NightlyBackUp",
            "backupTime": "2010-10-17 00:00:00"
            "hostId": "1234"
        }
    }
}
```

1.6.6. Sample XML Response (hosts/hostId/backups)

```
<backups>
    <backup
        id="1"
        name="great backup"
        backupTime="2010-10-17 00:00:00" />
</backups>
```

1.6.7. Sample JSON Response (hosts/hostId/backups)

```
{"backups": {
        "backup": {
            "id": "1",
            "name": "NightlyBackUp",
            "backupTime": "2010-10-17 00:00:00"
        }
    }
}
```

1.6.8. Sample XML Request (POST) (/hosts/hostId/backups)

```
<backup
    xmlns="http://docs.openstack.org/loadbalancers/api/management/
v1.0"
    name="NightlyBackUp" />
```

1.6.9. Sample JSON Request (POST) (/hosts/hostId/backups)

```
{"backup": {
        "name": "NightlyBackup"
    }
}
```

1.6.10. Sample XML Response (GET) (/hosts/hostId/subnetmappings)

```
<?xml version="1.0" ?>
<ns2:hostssubnet xmlns="http://docs.rackspacecloud.com/
loadbalancers/api/v1.0" xmlns:ns2="http://docs.rackspacecloud.com/
loadbalancers/api/management/v1.0">
    <ns2:hostsubnet name="api-zxtm-
n01.stage.dfw1.stabletransit.com">
        <ns2:netInterface name="eth0">
            <ns2:cidr block="10.69.1.0/24"/>
            <ns2:cidr block="10.69.0.0/24"/>
        </ns2:netInterface>
    </ns2:hostsubnet>
</ns2:hostssubnet>
```

1.6.11. Sample XML Request (PUT) (/hosts/hostId/subnetmappings)

```
<?xml version="1.0" ?>
<ns2:hostssubnet xmlns="http://docs.rackspacecloud.com/
loadbalancers/api/v1.0" xmlns:ns2="http://docs.rackspacecloud.com/
loadbalancers/api/management/v1.0">
        <ns2:hostsubnet name="n01.zeus.something">
            <ns2:netInterface name="eth1">
              <ns2:cidr block="192.168.0.0/16"/>
            </ns2:netInterface>
        </ns2:hostsubnet>
</ns2:hostssubnet>
```

1.6.12. Sample JSON Request (PUT) (/hosts/hostId/subnetmappings)

```
{
    "hostsubnets": [
```

```
        {
            "netInterfaces": [
                {
                    "cidrs": [
                        {
                            "block": "192.168.0.0/16"
                        }
                    ],
                    "name": "eth1"
                }
            ],
            "name": "n01.zeus.something"
        }
    ]
}
```

1.7. Customer List by Host/Cluster

Table 1.7.

Verb	URI	Description	Access Level
POST	/hosts/customers	Generate a customer list by host	Support, Service Admin
POST	/clusters/customers	Generate a customer list by cluster	Support, Service Admin

1.7.1. Description

The generated customer list allows external services, which contain customer contact information, to query the load balancing service to determine the breakdown of customers at either the cluster or host level.

1.7.2. Sample XML Request ByName (/hosts/customers) (/clusters/customers)

```
<byIdOrName
        xmlns="http://docs.openstack.org/loadbalancers/api/
management/v1.0"
        xmlns:ns2="http://docs.openstack.org/loadbalancers/
api/v1.0"
        name="newhost" />
```

1.7.3. Sample XML Request ById (/hosts/customers) (/clusters/customers)

```
<byIdOrName
           xmlns="http://docs.openstack.org/loadbalancers/api/
management/v1.0"
           xmlns:ns2="http://docs.openstack.org/loadbalancers/
api/v1.0"
           id="2" />
```

1.7.4. Sample XML Response (/hosts/customers) (/clusters/customers)

```
<?xml version="1.0" ?>
<customerList xmlns="http://docs.openstack.org/loadbalancers/
api/management/v1.0" xmlns:ns2="http://docs.openstack.org/
loadbalancers/api/v1.0">
 <customer accountId="666">
  <loadBalancer id="52" name="LB.4707"/>
  <loadBalancer id="53" name="LB.3133"/>
  <loadBalancer id="54" name="LB.6882"/>
  <loadBalancer id="55" name="LB.5095"/>
  <loadBalancer id="56" name="LB.3135"/>
  <loadBalancer id="57" name="LB.3399"/>
  <loadBalancer id="58" name="LB.1341"/>
  <loadBalancer id="59" name="LB.2750"/>
  <loadBalancer id="60" name="LB.1980"/>
  <loadBalancer id="61" name="LB.1237"/>
 </customer>
 <customer accountId="999">
  <loadBalancer id="47" name="LB.7090"/>
  <loadBalancer id="48" name="LB.3226"/>
  <loadBalancer id="49" name="LB.912"/>
  <loadBalancer id="50" name="LB.6697"/>
  <loadBalancer id="51" name="LB.6897"/>
 </customer>
 <customer accountId="354934">
  <loadBalancer id="45" name="LB.5001">
   <ns2:nodes>
    <ns2:node id="202" ip="21.248.193.163"/>
    <ns2:node id="204" ip="142.120.171.164"/>
    <ns2:node id="203" ip="231.162.141.253"/>
    <ns2:node id="205" ip="15.162.23.37"/>
    <ns2:node id="201" ip="182.254.203.171"/>
   </ns2:nodes>
  </loadBalancer>
  <loadBalancer id="46" name="LB.9540">
   <ns2:nodes>
    <ns2:node id="209" ip="200.47.43.240"/>
    <ns2:node id="207" ip="168.80.57.76"/>
    <ns2:node id="206" ip="229.220.95.81"/>
```

```
        <ns2:node id="208" ip="114.48.233.151"/>
        <ns2:node id="210" ip="228.221.170.235"/>
      </ns2:nodes>
    </loadBalancer>
  </customer>
</customerList>
```

1.7.5. Sample JSON Response (/hosts/hostId/customers) (/clusters/clusterId/customers)

```
{"customerlist": {
        "customer": {
            "id": "327957",
            "loadbalancer": [
                {
                    "id": "1",
                    "name": "lb-1"
                },
                {
                    "id": "2",
                    "name": "lb-2"
                }
            ]
        }
    }
}
```

2. Capacity and Availability Reports

2.1. Load Balancing Host Capacity Planning Board

Table 2.1.

Verb	URI	Description	Access Level
GET	/hosts/capacityreport	Generate a capacity planning report for all load balancing hosts	Service Admin
GET	/hosts/hostId/ capacityreport	Generate a capacity planning report for a specified load balancing host	Service Admin

2.1.1. Description

The load balancing host capacity reports provide insight into the available capacity of a given host machine. It also supplies an estimated runway of available capacity based on historical provisioning activity (activity taking place over a period of one week).

2.1.2. Sample XML Response (/hosts/capacityreport) (/ hosts/hostId/capacityreport)

```
<hostcapacityreports>
    <hostcapacityreport
            hostId="1"
            hostName="The Best Host Ever"
            totalConcurrentConnectionCapacity="150000"
            availableConcurrentConnections="100000"
            allocatedConcurrentConnections="50000"
            allocatedConcurrentConnectionsToday="14000"

 allocatedConcurrentConnectionsInPastSevenDays="25000"
            remainingDaysOfCapacity="9.55" />
</hostcapacityreports>
```

2.1.3. Sample JSON Response (/hosts/capacityreport) (/ hosts/hostId/capacityreport)

```
{"hostcapacityreports": {
        "hostcapacityreport": {
            "hostId": "1",
```

```
            "hostName": "The Best Host Ever",
            "totalConcurrentConnectionCapacity": "150000",
            "availableConcurrentConnections": "100000",
            "allocatedConcurrentConnections": "50000",
            "allocatedConcurrentConnectionsToday": "14000",
            "remainingDaysOfCapacity": "9.55"
        }
    }
}
```

2.2. Load Balancing Cluster Virtual IP Availability Report

Table 2.2.

Verb	URI	Description	Access Level
GET	/virtualips/ availabilityreport	Generate an availability report for all virtual IPs associated to any cluster.	Service Admin
GET	/clusters/clusterId/ virtualips/ availabilityreport	Generate an availability report for all virtual IPs associated to the specific cluster	Service Admin

2.2.1. Description

The virtual IP availability report provides insight into the availability of both public and ServiceNet IP addresses that are provisioned to a cluster. It also supplies an estimated runway of available capacity based on historical provisioning activity (activity taking place over a period of one week).

2.2.2. Sample XML Response (/virtualips/availabilityreport) (/cluster/clusterId/availabilityreport)

```
<virtualipavailabilityreports>
  <virtualipavailabilityreport
        clusterId="1"
        clusterName="My Little Cluster"
        totalPublicIpAddresses="254"
        totalServiceNetAddresses="254"
        freeAndClearPublicIpAddresses="128"
        freeAndClearServiceNetIpAddresses="128"
        publicIpAddressesInHolding="14"
```

```
        serviceNetIpAddressesInHolding="21"
        publicIpAddressesAllocatedToday="15"
        serviceNetIpAddressesAllocatedToday="4"
        allocatedPublicIpAddressesInLastSevenDays="45"
        allocatedServiceNetIpAddressesInLastSevenDays="15"
        remainingDaysOfPublicIpAddresses="50.11"
        remainingDaysOfServiceNetIpAddresses="14.41" />
</virtualipavailabilityreports>
```

2.2.3. Sample JSON Response (/virtualips/availabilityreport) (/cluster/clusterId/availabilityreport)

```
{"virtualipavailabilityreport": {
        "clusterId": "1",
        "clusterName": "My Little Cluster",
        "totalPublicIpAddresses": "254",
        "totalServiceNetAddresses": "254",
        "freeAndClearPublicIpAddresses": "128",
        "freeAndClearServiceNetIpAddresses": "128",
        "publicIpAddressesInHolding": "14",
        "serviceNetIpAddressesInHolding": "21",
        "publicIpAddressesAllocatedToday": "15",
        "serviceNetIpAddressesAllocatedToday": "4",
        "allocatedPublicIpAddressesInLastSevenDays": "45",
        "allocatedServiceNetIpAddressesInLastSevenDays": "15",
        "remainingDaysOfPublicIpAddresses": "50.11",
        "remainingDaysOfServiceNetIpAddresses": "14.41"
    }
}
```

3. Extended Customer API Operations

3.1. Assigning New Virtual IPs to a Load Balancer

Table 3.1.

Verb	URI	Description	Access Level
POST	/loadbalancers/ loadBalancerId/ virtualips	Provision an additional virtual IP to a load balancer	Support, Service Admin

3.1.1. Description

This feature allows a user to provision a new PUBLIC or SERVICENET address to the specified load balancer. This feature is restricted because a user must justify the need for additional IP addresses due to IANA requirements. An example of a reasonable justification would be the need to have a dedicated IP address for SSL termination.

To retrieve the assigned virtual IP, the caller must perform a subsequent GET on the / loadbalancers/loadBalancerId/virtualips URI.

3.1.2. Sample XML POST Request (/loadbalancers/ loadBalancerId/virtualips)

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

3.1.3. Sample JSON POST Request (/loadbalancers/ loadBalancerId/virtualips)

```
]{"virtualip": {
        "type": "PUBLIC"
    }
}
```

3.2. Synchronizing a Load Balancer

Table 3.2.

Verb	URI	Description	Access Level
POST	/loadbalancers/ loadBalancerId/sync	Synchronizes the load balancer with the authoritative source	Support, Service Admin

3.2.1. Description

This feature allows for a caller to synchronize the specified load balancer with a particular authoritative resource. At the time of this writing the authoritative source is the database in which all load balancer configurations are stored. We may plan on adding Zeus as an authoritative source later. If a caller chooses to synchronize from the database then Zeus will be updated to match the load balancer's configuration in the database. The caller only needs to send a POST request without a body to the specified URI. This feature should only be used in the event that Zeus contains mismatching configuration information (which should hopefully never even happen).

3.3. Suspending a Load Balancer

Table 3.3.

Verb	URI	Description	Access Level
GET	/loadbalancers/ loadBalancerId/ suspension	Retrieve the current suspension information for the load balancer, if any	Support, Service Admin
POST	/loadbalancers/ loadBalancerId/ suspension	Issues an immediate service suspension for a load balancer	Support, Service Admin
DELETE	/loadbalancers/ loadBalancerId/ suspension	Removes the suspension for a load balancer	Support, Service Admin

3.3.1. Description

This feature allows for a caller to suspend and or unsuspend the specified load balancer. In order to suspend a load balancer, the caller must supply a reason, the ticket issuer(user) and a ticket identifier, which can be viewed by other users with support and service admin access levels.

If a caller requests suspension details for an unsuspended load balancer, an empty suspension element (<suspension />) will be returned.

Note
While customers are not permitted to delete suspended load balancers, a user with elevated permissions may do so by issuing a DELETE request against the / loadbalancers/loadBalancerId URI.

3.3.2. Sample XML Response (/loadbalancers/ loadBalancerId/suspension)

```
<suspension
```

```
   xmlns="http://docs.openstack.org/loadbalancers/api/management/
v1.00"
   reason="User suspected of fraud"
   ticketId="1054"
   user="jdoe" />
```

3.3.3. Sample JSON Response (/loadbalancers/ loadBalancerId/suspension)

```
{"suspension": {
        "reason": "User suspected of fraud",
        "ticketId": "1054",
        "user": "jdoe"
    }
}
```

3.3.4. Sample XML POST Request (/loadbalancers/ loadBalancerId/suspension)

```
<suspension
   xmlns="http://docs.openstack.org/loadbalancers/api/management/
v1.0"
   reason="Non-payment"
   ticketId="4141"
   user="bob"/>
```

3.3.5. Sample JSON Response (/loadbalancers/ loadBalancerId/suspension)

```
{"suspension": {
        "reason": "Non-payment",
        "ticketId": "4141",
        "user": "bob"
    }
}
```

3.4. Viewing Extended Details of a Load Balancer

Table 3.4.

Verb	URI	Description	Access Level
GET	/loadbalancers/ loadBalancerId/ extendedview	View extended details of a load balancer	Support, Service Admin

3.4.1. Description

This operation provides the detailed output for a specific load balancer configured and associated with the designated account. It differs from the customer view because it provides the total active connections, host machine information, and a rate limit (if applicable) as part of the loadbalancer element.

Note

An extended detail view is not available for a list of load balancers.

3.4.2. Sample XML Response (/loadbalancers/ loadBalancerId/extendedview)

```
<loadBalancer xmlns="http://docs.rackspacecloud.com/loadbalancers/
api/v1.0"
   id="2000"
   name="sample-loadbalancer"
   protocol="HTTP"
   port="80"
   algorithm="RANDOM"
   status="ACTIVE"
   totalActiveConnections="340">
   <host id="1" type="ACTIVE" />
   <currentUsage
      incomingTransfer="1.40"
      outgoingTransfer="20.14" />
   <virtualIps>
      <virtualIp
         id="1000"
         address="206.10.10.210"
         type="PUBLIC" />
   </virtualIps>
   <nodes>
      <node
         nodeId="1041"
         ip="10.1.1.1"
         port="80"
         condition="ENABLED"
```

```
                    status="ONLINE" />
          <node
             nodeId="1411"
             ip="10.1.1.2"
             port="80"
             condition="ENABLED"
             status="ONLINE" />
      </nodes>
      <sessionPersistence persistenceType="HTTP_COOKIE"/>
      <connectionLimits
         minConnections="10"
         maxConnectionsFromIp="100"
         maxConnectionRateFromIp="50"
         maxConnectionRateTimer="60" />
      <connectionLogging enabled="false" />
      <cluster name="c1.dfw1" />
         <ratelimit
             ticketId="4100"
             expirationTime="2010-10-17 00:00:00"
             maxRequestsPerSecond="100" />
      <created time="2010-06-01T12:00:00Z" />
      <updated time="2010-06-01T12:00:00Z" />
</loadBalancer>
```

3.4.3. Sample JSON Response (/loadbalancers/ loadBalancerId/extendedview)

```
{"loadbalancer": {
       "id": "2000",
       "name": "sample-loadbalancer",
       "protocol": "HTTP",
       "port": "80",
       "algorithm": "RANDOM",
       "status": "ACTIVE",
       "totalActiveConnections": "340",
       "currentUsage": {
           "incomingTransfer": "1.40",
           "outgoingTransfer": "20.14"
       },
       "host": {
               "id": "1",
               "type": "ACTIVE"
        },
       "virtualIps": {
           "virtualIp": {
               "id": "1000",
               "address": "206.10.10.2010",
```

```json
                "type": "PUBLIC"
            }
        },
        "nodes": {
            "node": [
                {
                    "nodeId": "1041",
                    "ip": "10.1.1.1",
                    "port": "9090",
                    "condition": "ENABLED",
                    "status": "ONLINE"
                },
                {
                    "nodeId": "1042",
                    "ip": "10.1.1.2",
                    "port": "80",
                    "condition": "ENABLED",
                    "status": "ONLINE"
                }
            ]
        },
        "connectionLogging": {
            "enabled": "true"
        },
        "rateLimit": {
            "ticketId": "1123",
            "expirationTime": "1283277190574",
            "maxRequestsPerSecond": "37"
        },
        "sessionPersistence": {
            "persistenceType": "HTTP_COOKIE"
        },
        "connectionLimits": {
            "minConnections": "10",
            "maxConnectionsFromIp": "100",
            "maxConnectionsRateFromIp": "50",
            "maxConnectionsRateTimer": "60"
        },
        "cluster": {
            "name": "c1.dfw1",
            "description": "Cluster Description"
        },
        "created": {
            "time": "2010-06-01 00: 00: 00"
        },
        "updated": {
            "time": "2010-06-01 00: 00: 00"
        }
    }
}
```

3.5. Account Loadbalancers and Usage (Billing)

Table 3.5.

Verb	URI	Description	Access Level
GET	/accounts/accountId/ loadbalancers	Retrieves a list of load balancers on this account	Support, Service Admin
GET	/accounts/billing/ date	Retrieve usage for all accounts for a single day	Support, Service Admin, Billing

Normal Response Code(s): 200

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

3.5.1. Description

A user can list all loadbalancers and their usage for the given account using these methods. The date parameters follow the yyyy-MM-dd date format. Please note that usage is retained for a maximum of 90 days.

3.5.2. Sample XML Response (/accounts/accountId/ loadbalancers)

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:accountLoadBalancers xmlns="http://docs.openstack.org/
loadbalancers/api/v1.0" xmlns:ns2="http://docs.openstack.org/
loadbalancers/api/management/v1.0" accountId="1106">
    <ns2:accountLoadBalancer loadBalancerId="61"
 loadBalancerName="test1" clusterId="1" clusterName="Cluster
 Delta" protocol="HTTP" status="DELETED"/>
    <ns2:accountLoadBalancer loadBalancerId="62"
 loadBalancerName="test2" clusterId="1" clusterName="Cluster
 Delta" protocol="HTTPS" status="DELETED"/>
    <ns2:accountLoadBalancer loadBalancerId="66"
 loadBalancerName="My first loadbalancer" clusterId="1"
 clusterName="Cluster Delta" protocol="HTTP" status="ACTIVE"/>
    <ns2:accountLoadBalancer loadBalancerId="75"
 loadBalancerName="test3" clusterId="1" clusterName="Cluster
 Delta" protocol="HTTPS" status="DELETED"/>
    <ns2:accountLoadBalancer loadBalancerId="76"
 loadBalancerName="test4" clusterId="1" clusterName="Cluster
 Delta" protocol="HTTPS" status="PENDING_UPDATE"/>
    <ns2:accountLoadBalancer loadBalancerId="77"
 loadBalancerName="My second loadbalancer" clusterId="1"
 clusterName="Cluster Delta" protocol="HTTPS" status="ACTIVE"/>
```

```
</ns2:accountLoadBalancers>
```

3.6. Load Balancer Status URL Management

More often then not a LoadBalancer gets stuck in pending update due to a broken transaction that should have rolled back in the database but didn't.

In this case an ops personal can use the following URL's to correct the issues.

3.6.1. Examples

```
loadbalancers/68/setstatus/ACTIVE
loadbalancers/68/setstatus/BUILD
loadbalancers/68/setstatus/DELETED
loadbalancers/68/setstatus/ERROR
loadbalancers/68/setstatus/PENDING_DELETE
loadbalancers/68/setstatus/PENDING_UPDATE
```

3.7. Listing and Acknowledging Alerts

This operations allows user to list all the alerts in the system. Additional information about the alert can be retrieved by alert id. The status of the alert can be changed to ACKNOWLEDGED using the PUT method

Table 3.6.

Verb	URI	Description	Access Level
GET	/alert	Retrieve all alerts, if any	Support, Service Admin
GET	/alerts/ unacknowledged	Retrieve unacknowledged alerts, if any	Support, Service Admin
GET	/alers/alertId	Retrieve details for the alert	Support, Service Admin
GET	/loadbalancers/ loadBalancerId/alerts	Retrieve all alerts for the specified load balancer, if any	Support, Service Admin
PUT	/alerts/alertId/ acknowledged	Acknowledge alert. Changes the status of alert to ACKNOWLEDGED	Support, Service Admin

3.7.1. Sample XML Response (/alerts)

```
<alerts>
    <alert id="1" accountId="38" loadbalancerid="1000"
 messageName="Error updating node" status=UNACKNOWLEDGED"
 created="2010-11-23"/>
</alerts>
```

3.7.2. Adding group rate limiting

This operation allows user to add a group that will be used for API rate limiting. The definitions of the group are maintained in the rate limiting configuration file. The name of the group added has to match the name of the group in the configuration file

Table 3.7.

Verb	URI	Description	Access Level
POST	/groups? name=xxx&default=Y	Add group name and default	Support, Service Admin

3.8. Viewing Events

These operations allow support to view events.

Table 3.8.

Verb	URI	Description	Access Level
GET	/event/account/ accountId/ loadbalancer? startDate=2010-1-1&endDate=2010-1-1	List recent load balancer service events on this account	Support, Service Admin

3.8.1. GET /management/event/account/accountId/ loadbalancer?startDate=2010-1-1&endDate=2010-1-1

The date parameters are optional. The results will list will be all load balancer service events that occurred between the start and end dates. by default the start date will be 60 days from the time of the request and the end date will be the date of the request. The result list will be inclusive for both dates.

3.8.2. Sample XML Response (/alerts)

```
<accountLoadBalancerServiceEvents xmlns="http://
docs.openstack.org/loadbalancers/api/v1.0" accountId="546428">
        <loadBalancerServiceEvents loadbalancerId="8">
            <loadBalancerServiceEvent author="abc"
 description="LoadBalancer successfully created" category="CREATE"
 severity="INFO" created="01/06/2011 10:37"/>
```

```
            </loadBalancerServiceEvents>
</accountLoadBalancerServiceEvents>
```