Heat

1 F5 OpenStack Integrations Documentation 3
2 Enhancements and issues 55
3 Acknowledgements 57
The F5 Integration for OpenStack provides a means for OpenStack users to administer and orchestrate L4-L7 services on BIG-IP device(s) using the OpenStack CLI and OpenStack API.
1.1 F5 OpenStack Heat Integration

F5’s OpenStack Heat Integration consists of the F5 OpenStack Heat templates and F5 OpenStack Heat plugins.

1.1.1 General Prerequisites

The F5 OpenStack Heat Integration’s documentation set assumes that you:

- already have an operational with OpenStack Neutron and OpenStack Heat installed;¹
- are familiar with OpenStack Horizon and the OpenStack CLI ; and
- are familiar with F5 BIG-IP Local Traffic Manager (LTM) concepts and the BIG-IP configuration utility (aka, the BIG-IP admin web interface).
- have a BIG-IP Virtual Edition (VE) qcow.zip image file (any size) licensed appropriately for your needs.²

Caveats

- You must host your BIG-IP VE image in a location accessible via ‘http’. The F5 Heat templates can not retrieve files via ‘https’. File uploads via OpenStack Horizon are not supported.
- BIG-IP VE images come in different sizes. Choose the correct F5 flavor for your image.

1.1.2 Heat Plugins

The F5 OpenStack Heat plugins enable BIG-IP objects for use in OpenStack. The Heat plugins use the F5 Python SDK to communicate with BIG-IP via the iControl REST API.

Install the F5 Heat Plugins

¹ Unsure how to get started with OpenStack? Consult one of F5’s OpenStack Platform Partners.
² How to Buy
1.1.3 Heat Templates

You can use the F5-Supported Heat Templates to provision resources and BIG-IP services in an OpenStack cloud. F5’s Heat templates follow the OpenStack Heat Orchestration Template (HOT) specification. You can use the templates in conjunction with F5 iApps to deploy BIG-IP VE instances and Local Traffic Manager services.

**Warning:** The F5 Heat template library contains supported and unsupported templates. The unsupported templates provided in the f5-openstack-heat GitHub repo are ‘use-at-your-own-risk’.

Templates in the unsupported directory are not supported by F5, regardless of your account’s support agreement status.

Browse the F5-Supported Heat Templates

See also:
- f5-openstack-heat
- f5-openstack-heat-plugins
- F5 OpenStack Heat templates
- F5 OpenStack Heat plugins

1.2 How to Deploy OpenStack Heat Templates

Use the instructions provided here to deploy any of the templates in F5’s heat orchestration template library.

1.2.1 OpenStack CLI

**Platform Requirements**

- python-openstackclient configured with the python-heatclient plugin
- F5 OpenStack Heat plugins

The OpenStack Heat documentation extensively covers creation and management of Heat stacks. This document describes F5’s OpenStack development team’s preferred means of stack creation:

- define a Heat template’s required parameters in a YAML file, and
- deploy the template via the command line.

**Note:** This example uses the F5-supported Deploy basic load balancer template.
1. Download the template you want to deploy from the F5-Supported Heat Templates library.

2. Save the definition for each of the template’s required configuration parameters in an environment file.

```
parameters:
  client_server_image: <image-name>
  client_server_flavor: <nova-flavor>
  key_name: <ssh-key-name>
  client_server_sec_group: open
  client_network: <client_vlan>
  server_network: <server_vlan>
  bigip_un: $BIGIP_USER
  bigip_pw: $BIGIP_PASS
  vs_name: virtual_server1
  pool_name: pool1
  bigip_fip:
  vs_vip:
  vs_port: 443
  pool_member_port: 8080
```

Replace the following parameters with the appropriate values for your environment.

- image
- flavor
- ssh-key
- security group
- network

**Tip:** Protect your BIG-IP login information.

Store your username and password as environment variables and reference them in your environment file.

3. Create the Heat stack.

```
openstack stack create -f deploy-lb.yaml -e deploy-lb.params.yaml
```

### 1.2.2 OpenStack Horizon

Follow the directions below to deploy a Heat stack using the OpenStack Horizon dashboard.

1. Go to *Orchestration* → *Stacks*.
2. Click *Launch Stack*.
3. Choose the *Template File* from its location on your machine, then click *Next*.
4. Provide the information required for the Heat engine to build your stack.
5. Click *Launch*.

**Hint:** In the *Stacks* table, the status changes to *Create complete* when the deployment finishes.
1.3 How to Deploy a BIG-IP VE instance and set up a basic load balancer

This guide demonstrates how to use the OpenStack Heat orchestration service to onboard, deploy, and provision a F5 BIG-IP Virtual Edition (VE) instance. We provide instructions for deploying the Heat templates via the OpenStack Horizon dashboard.

This guide uses the following F5-Supported Heat templates:

- BIG-IP VE image patch and upload: Add an OpenStack-ready BIG-IP VE image to Glance.
- BIG-IP VE Standalone, 3-nic: Deploy a standalone, 3-nic BIG-IP VE instance.
- Deploy basic load balancer: Deploy a simple load balancer on the BIG-IP VE instance.

1.3.1 Before You Begin

In addition to the basic F5 Heat prerequisites, you will need the following to follow this guide:

- An external network set up in Neutron that can access the internet.
- F5 OpenStack Heat plugins installed on the Neutron controller.
- SSH key(s) set up in OpenStack.
- (Optional) Add the F5 flavors to OpenStack.

Caveats

- The Heat engine needs to access the BIG-IP VE image file via “http”. The F5 Heat templates cannot retrieve files via “https” and do not support file uploads.

1.3.2 Add the BIG-IP VE image to Glance

Add a Ubuntu image

The BIG-IP image patch and upload template uses a Ubuntu server to run a patch script on the BIG-IP VE qcow image. If you don’t already have a Ubuntu image in Glance, you’ll need to add one.

1. Copy the download URL for a Ubuntu 14.04 server image.\(^1\)
2. Add the image to Glance:
   - In Horizon, go to Compute \(\rightarrow\) Images.

\(^1\) Ubuntu 14.04 is the only version verified by the OpenStack development team. Other versions may work, but deployments using unverified versions are not supported by F5.
• Click Create Image and paste the image URL in the Image Location field.
• Enter the requested information, including the minimum requirements for your image (we used 7GB disk and 520MB RAM).
• Click Create to add the image to Glance.

**Patch and upload the BIG-IP VE image**

1. Download the BIG-IP VE image patch and upload heat template.
2. Define the stack parameters in a YAML file.

**Tip:** The sample environment file below shows the values you’ll need to provide when deploying the BIG-IP VE image patch and upload heat stack.

Before you deploy your stack, replace the example values with the correct information for your environment.

```yaml
parameters:
onboard_image: ubuntu-14.04.5-server-amd64.iso
# provide the name of a private network in your OpenStack cloud
private_network: mgmt_net
# provide your Keystone authentication URL
f5_image_import_auth_url: http://1.2.3.4:5000/v2.0
# provide the name of the OpenStack tenant (project) you want to import the # image to
f5_image_import_tenant: admin
# provide your OpenStack user name
f5_image_import_user: admin
# provide your OpenStack password
f5_image_import_password: $MY_PASSWORD
# DO NOT CHANGE THIS ENTRY
# This F5 Networks GitHub repo contains the image patch script
image_prep_url: https://github.com/F5Networks/f5-openstack-image-prep.git
# Provide the URL for your BIG-IP qcow.zip image
# Must be accessible via http
f5_ve_image_url: http://path/to/my/big-ip-ve-image
f5_ve_image_name: BIGIP-12.1.2.0.0.249.LTM.qcow2.zip
# Provide the name of an SSH key configured in OpenStack
# This key will be used to authenticate to the Ubuntu server
image_prep_key: my-openstack-ssh-key
```

Sample image patch and upload env file

3. Deploy the Heat stack using the OpenStack CLI.

```bash
openstack stack create -f patch_upload_ve_image.yaml -e patch-upload-ve-image.params.yaml
```

**1.3.3 Deploy a BIG-IP VE instance**

1. Download the BIG-IP VE standalone, 3-nic heat template.
2. Define the stack parameters in a YAML file.

**Tip:** The sample environment file below shows the values you’ll need to provide when deploying the BIG-IP VE standalone, 3-nic heat stack.

Before you deploy your stack, replace the example values with the correct information for your environment.

```yaml
parameters:
  ve_image: BIGIP-12.1.2.0.0.249.LTM.qcow2
  # Default is m1.medium
  ve_flavor: m1.medium
  # Provide the name of an SSH key configured in OpenStack
  f5_ve_os_ssh_key: my-openstack-ssh-key
  # Create a password for the BIG-IP admin user account
  admin_password: $ADMIN_PASSWORD
  # This value is set to be hidden
  root_password: $ROOT_PASSWORD
  # This value is set to be hidden
  license: <license_string>
  # Provide the name of the Neutron network that has access to the internet
  external_network: neutron.ext.network
  # Provide the name of a Neutron network to assign to the BIG-IP mgmt interface
  mgmt_network: test
  # Provide the name of an existing Neutron network to connect to the BIG-IP 1.1
  network_1: tmm_network
  # Assign a name to the 1.1 network on the BIG-IP instance
  network_1_name: network-1.1
  # Provide the name of an existing Neutron network to connect to the BIG-IP 1.2
  network_2: tmm_network
  # Assign a name to the 1.2 network on the BIG-IP instance
  network_2_name: network-1.2
  # Provide the default gateway IP or hostname; defaults to 'none'
  default_gateway: none
```

Sample BIG-IP VE standalone 3-nic env file

3. Deploy the Heat stack using the OpenStack CLI.

```bash
openstack stack create -f f5_ve_standalone_3_nic.yaml -e f5_ve_standalone_3_nic.params.yaml
```

**Assign a Floating IP Address to the BIG-IP instance**

Use the OpenStack Horizon dashboard to assign a floating IP address to the BIG-IP VE instance.

1. Go to **Project → Compute → Instances**, then choose **Associate Floating IP** from the drop-down menu in the **Actions** column.
2. Select a **Floating IP** from the **IP Address** drop-down menu.
3. In the port drop-down, select a port for your BIG-IP instance that corresponds to the external VLAN.
4. Click Associate.

**Tip:** If there aren’t any floating IP addresses available in the drop-down menu:
- Click + to generate a floating IP address.
- Click Allocate.

The availability of these actions may depend on your OpenStack user privileges.

### 1.3.4 Create a basic load balancer on the BIG-IP VE instance

1. Download the Deploy basic load balancer heat template.
2. Define the stack parameters in a YAML file.

**Tip:** The sample environment file below shows the values you’ll need to provide when deploying the Basic Load Balancer heat stack. Before you deploy your stack, replace the example values with the correct information for your environment.

```yaml
parameters:
  client_server_image: <image-name>
  client_server_flavor: <nova-flavor>
  key_name: <ssh-key-name>
  client_server_sec_group: open
  client_network: <client_vlan>
  server_network: <server_vlan>
  bigip_un: $BIGIP_USER
  bigip_pw: $BIGIP_PASS
  vs_name: virtual_server1
  pool_name: pool1
  bigip_fip:
  vs_vip:
  vs_port: 443
  pool_member_port: 8080
```

Sample Deploy Basic Load Balancer env file

3. Deploy the Heat stack using the OpenStack CLI.

```
openstack stack create -f deploy_lb.yaml -e deploy_lb.params.yaml
```

### 1.3.5 Next Steps

1. Configure your BIG-IP device.

**Tip:**
- You can access the BIG-IP from the OpenStack dashboard via System → Instances → Console.
• To log in to the BIG-IP configuration utility, copy its floating IP address from the Instance screen in the dashboard, then paste it into your browser’s address bar. You must use https to connect.
• You can connect to the BIG-IP instance via ssh using the floating IP address and the ssh key you provided in the environment file.

2. Set up the F5 Neutron LBaaS integration to communicate with your new BIG-IP instance.

1.4 F5 Integration for OpenStack Neutron LBaaS

The F5 Integration for OpenStack Neutron LBaaS orchestrates BIG-IP Application Delivery Controllers (ADCs) with OpenStack Networking (Neutron) services. The Integration consists of the F5 Agent for OpenStack Neutron and F5 Driver for OpenStack LBaaSv2, which work together to configure F5 BIG-IP Local Traffic Manager (LTM) objects via the OpenStack Networking API.

1.4.1 General Prerequisites

This documentation set assumes that you:

• already have an operational with OpenStack Neutron and OpenStack Heat installed;¹
• are familiar with OpenStack Horizon and the OpenStack CLI; and
• are familiar with BIG-IP LTM concepts, the BIG-IP configuration utility, and tmsh commands.

See also:
See the F5 OpenStack Solution Test Plan for information about minimum supported deployments.

1.4.2 F5 Driver for OpenStack LBaaSv2

The F5 Driver for OpenStack LBaaSv2, or F5 driver, is F5’s OpenStack Neutron LBaaSv2 service provider driver. It picks up Neutron LBaaS calls from the RPC messaging queue and assigns them to the F5 Agent for OpenStack Neutron.

1.4.3 F5 Agent for OpenStack Neutron

The F5 Agent for OpenStack Neutron, or F5 agent, translates from “OpenStack” to “F5”. It receives tasks from the Neutron RPC messaging queue, converts them to iControl REST API calls (using the F5 Python SDK), and sends the calls to the BIG-IP device(s).

1.4.4 Key OpenStack Concepts

Agent-Tenant Affinity

When the Neutron LBaaS plugin loads the F5 driver, it creates a global messaging queue. The F5 Agent for OpenStack Neutron sends all callbacks and status updates to this global queue. The F5 driver picks up LBaaS requests from the global messaging queue in a round-robin fashion, then assigns the tasks to an available F5 agent instance based on “agent-tenant affinity”.

¹ Unsure how to get started with OpenStack? Consult one of F5’s OpenStack Platform Partners.
1. User issues `neutron lbaas` command.
2. F5 LBaaSv2 service provider driver picks up the LBaaS task from the Neutron RPC messaging queue.
3. F5 LBaaSv2 service provider driver assigns the LBaaS task to the F5 OpenStack BIG-IP Controller.
4. The F5 OpenStack BIG-IP Controller periodically sends status updates back to the Neutron database.

Fig. 1.1: F5 Integration for OpenStack Neutron LBaaS Architecture

1. User issues `neutron lbaas` command.
2. The F5 LBaaSv2 service provider driver assigns the task to the F5 OpenStack BIG-IP Controller.
3. The F5 OpenStack BIG-IP Controller issues an iControl REST API call to the BIG-IP device to add/edit the requested object.

Fig. 1.2: F5 Agent for OpenStack Neutron traffic flow
Agent-tenant affinity is a relationship between an F5 agent instance and an OpenStack “tenant”, or project. In brief, once an F5 agent handles an LBaaS request for a particular OpenStack tenant, the F5 agent has “agent-tenant affinity” with that tenant. That instance will handle all future LBaaS requests for that tenant (with a few caveats, noted below).

How “agent-tenant affinity” applies in LBaaS task assignment:

**Table 1.1: Agent-tenant affinity for a new load balancer**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>You request a new load balancer (neutron lbaas-loadbalancer-create).</td>
</tr>
<tr>
<td>2.</td>
<td>The F5 LBaaSv2 driver checks the Neutron database to find out if an F5 agent instance already has affinity with the tenant the load balancer request is for.</td>
</tr>
<tr>
<td>3.</td>
<td>If the F5 LBaaSv2 driver finds an F5 agent instance that has affinity with the load balancer’s tenant, it assigns the request to that instance.</td>
</tr>
<tr>
<td>4.</td>
<td>If the F5 LBaaSv2 driver doesn’t find an F5 agent instance that has affinity with the load balancer’s tenant_id, it selects an active F5 agent instance at random. The selected instance binds to the requested load balancer. It will handle all future LBaaS requests for that load balancer.</td>
</tr>
</tbody>
</table>

**Table 1.2: Agent-tenant affinity for an existing load balancer**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>You update an existing load balancer (neutron lbaas-loadbalancer-update).</td>
</tr>
<tr>
<td>2.</td>
<td>The F5 LBaaSv2 driver checks the Neutron database to find out if an F5 agent instance is already bound to the load balancer.</td>
</tr>
<tr>
<td>3.</td>
<td>If the F5 LBaaSv2 driver doesn’t find a bound F5 agent instance for the load balancer, it looks for an instance that has affinity with the load balancer’s tenant, then assigns the request to that instance.</td>
</tr>
<tr>
<td>4.</td>
<td>If the F5 LBaaSv2 driver doesn’t find an F5 agent instance that has affinity with the load balancer’s tenant_id, it selects an active F5 agent instance at random. The selected instance binds to the requested load balancer. It will handle all future LBaaS requests for that load balancer.</td>
</tr>
</tbody>
</table>

**Important:** If the F5 agent bound to a load balancer is inactive, the F5 LBaaSv2 driver looks for other active agents with the same Set up F5 agent to use the new environment. The F5 LBaaSv2 driver assigns the task to the first available agent it finds. The inactive F5 agent remains bound to the load balancer, with the expectation that it will eventually come back online and be able to handle future requests.

**Important:** If you delete an F5 agent, you should also delete all of its bound load balancers.

To find all load balancers associated with a specific F5 agent:
1.4.5 Partnerships and certifications

The F5 Integration for OpenStack Neutron LBaaS provides under-the-cloud multi-tenant infrastructure L4-L7 services for Neutron tenants. In addition to community OpenStack participation, F5 maintains partnerships with several OpenStack platform vendors. Each partner has a defined certification process that includes requirements for testing the F5 Integration for OpenStack Neutron LBaaS for vendor and community OpenStack compatibility. See the Solution Test Plan for more information.

See also:
- f5-openstack-lbaasv2-driver
- f5-openstack-agent
- F5 Agent for OpenStack Neutron
- F5 Driver for OpenStack LBaaSv2

1.5 F5 LBaaSv2 Quick Reference

<table>
<thead>
<tr>
<th>Applies to:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F5 LBaaS version(s)</td>
<td>OpenStack version(s)</td>
</tr>
<tr>
<td>v10.0+</td>
<td>Newton</td>
</tr>
</tbody>
</table>

To install previous versions, see the F5 Agent for OpenStack Neutron documentation.

This reference sheet provides the basic information you need to get the F5 Integration for OpenStack Neutron LBaaS up and running using the latest release: v 10.0.0.

1.5.1 Tasks

1. Download and install the F5 service provider package; the F5 agent; and the F5 Driver for OpenStack LBaaSv2.
   - Debian
     ```bash
curl -O -L https://github.com/F5Networks/neutron-lbaas/releases/download/v10.0.0/f5.tgz
tar xvf f5.tgz -C /usr/lib/python2.7/dist-packages/neutron_lbaas/
curl -L -O https://github.com/F5Networks/f5-openstack-agent/releases/download/v10.0.0/python-f5-openstack-agent_10.0.0-1_1404_all.deb
curl -L -O https://github.com/F5Networks/f5-common-python/releases/download/v2.3.3/python-f5-sdk_2.3.3-1_1404_all.deb
curl -L -O https://github.com/F5Networks/f5-icontrol-rest-python/releases/download/v1.3.0/python-f5-icontrol-rest_1.3.0-1_1404_all.deb
```
curl -L -O https://github.com/F5Networks/f5-openstack-lbaasv2-driver/releases/download/v10.0.0/python-f5-openstack-lbaasv2-driver_10.0.0-1_1404_all.deb
dpkg -i python-f5-icontrol-rest_1.3.0-1_1404_all.deb
dpkg -i python-f5-sdk_2.3.3-1_1404_all.deb
dpkg -i python-f5-openstack-agent_10.0.0-1_1404_all.deb
dpkg -i python-f5-openstack-lbaasv2-driver_10.0.0-1_1404_all.deb

• RPM

curl -O -L https://github.com/F5Networks/neutron-lbaas/releases/download/v10.0.0/f5.tgz
tar xvf f5.tgz -C /usr/lib/python2.7/site-packages/neutron_lbaas/drivers/
curl -L -O https://github.com/F5Networks/f5-common-python/releases/download/v2.3.3/f5-sdk-2.3.3-1.el7.noarch.rpm
curl -L -O https://github.com/F5Networks/f5-icontrol-rest-python/releases/download/v1.3.0/f5-icontrol-rest-1.3.0-1.el7.noarch.rpm
curl -L -O https://github.com/F5Networks/f5-openstack-agent/releases/download/v10.0.0/f5-openstack-agent-10.0.0-1.el7.noarch.rpm
curl -L -O https://github.com/F5Networks/f5-openstack-lbaasv2-driver/releases/download/v10.0.0/f5-openstack-lbaasv2-driver-10.0.0-1.el7.noarch.rpm
sudo rpm -ivh f5-icontrol-rest-1.3.0-1.el7.noarch.rpm f5-sdk-2.3.3-1.el7.noarch.rpm f5-openstack-agent-10.0.0-1.el7.noarch.rpm f5-openstack-lbaasv2-driver-10.0.0-1.el7.noarch.rpm

• Pip

Download the F5 driver package (f5.tgz) and install it in the correct path for your OS (see above examples). Then, run the commands shown below to install the F5 agent and F5 driver packages.

pip install git+https://github.com/F5Networks/f5-openstack-agent@v10.0.0
pip install git+https://github.com/F5Networks/f5-openstack-lbaasv2-driver@v10.0.0

2. Set up Neutron to use the F5 service provider driver.

**Important:** The Neutron configuration steps may differ from the instructions provided, depending on your OpenStack platform. Please see our partners’ documentation for more information.

• HPE Helion OpenStack
• Mirantis OpenStack
• RedHat OpenStack Platform

3. Set up the F5 agent.

Sample configuration files:

• Global Routed Mode example
• GRE example¹
• VLAN example

¹ You need a Better or Best license if you plan to use GRE or VxLAN tunnels in an L2-adjacent under-the-cloud deployment.
• VXLAN example

4. Start the F5 agent.

**CentOS**

```bash
systemctl enable f5-openstack-agent
systemctl start f5-openstack-agent
sudo systemctl stop f5-openstack-agent.service
```

**Ubuntu**

```bash
service f5-oslbaasv2-agent start
service f5-oslbaasv2-agent stop
```

### 1.5.2 What’s Next

- *Set up a basic load balancer using the Neutron CLI.*
- Discover how the F5 agent *maps Neutron commands to BIG-IP objects.*

**Applies to:**

<table>
<thead>
<tr>
<th>F5 LBaaS version(s)</th>
<th>OpenStack version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v8.x.x</td>
<td>Liberty</td>
</tr>
<tr>
<td>v9.x.x</td>
<td>Mitaka</td>
</tr>
<tr>
<td>v10.x.x</td>
<td>Newton</td>
</tr>
</tbody>
</table>

### 1.6 Basic Environment Requirements

This document provides the minimum basic requirements for using the F5 Integration for OpenStack Neutron LBaaS in OpenStack.

#### 1.6.1 OpenStack Requirements

The *OpenStack installation guides* cover the requirements for specific environments.

We recommend that you install and configure the following OpenStack services. Each of these is necessary for one or more F5 OpenStack integrations.
### 1.6.2 BIG-IP Device Requirements

**Important:**

- You must have the appropriate license for the BIG-IP features you wish to use.
- All numbers shown in the table below are per BIG-IP device.

<table>
<thead>
<tr>
<th>Deployment</th>
<th>NICs</th>
<th>VLANs¹</th>
<th>Tunnels²</th>
<th>VTEPs³</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone overcloud</td>
<td>2</td>
<td>2</td>
<td>n/a</td>
<td>n/a</td>
<td>any</td>
</tr>
<tr>
<td>Standalone undercloud</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>better or best</td>
</tr>
<tr>
<td>Pair overcloud</td>
<td>3</td>
<td>3</td>
<td>n/a</td>
<td>n/a</td>
<td>any</td>
</tr>
<tr>
<td>Pair undercloud</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>better or best</td>
</tr>
<tr>
<td>Scalen cluster overcloud</td>
<td>3</td>
<td>3</td>
<td>n/a</td>
<td>n/a</td>
<td>any</td>
</tr>
<tr>
<td>Scalen cluster undercloud</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>better or best</td>
</tr>
</tbody>
</table>

See also:

- [F5 OpenStack Releases and Support Matrix](#)
- [BIG-IP LTM Release Notes](#)

#### Applies to:

<table>
<thead>
<tr>
<th>F5 LBaaS version(s)</th>
<th>OpenStack version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v8.x.x</td>
<td>Liberty</td>
</tr>
<tr>
<td>v9.x.x</td>
<td>Mitaka</td>
</tr>
<tr>
<td>v10.x.x</td>
<td>Newton</td>
</tr>
</tbody>
</table>

---

¹ The LBaaS2 dashboard panels are available in OpenStack Mitaka and later releases.

² Two VLANs = data & management. Three VLANs = data, management, and HA. See Configuring the basic BIG-IP network for more information.

³ Tunnels can be either VxLAN or GRE.

⁴ If you’re using a tunnel to reach an undercloud BIG-IP, you must configure the BIG-IP virtual tunnel endpoint (VTEP) **before** launching the F5 agent.
1.7 Neutron to BIG-IP Command Mapping

When you issue `neutron lbaas` commands on your OpenStack Neutron controller, the F5 Agent for OpenStack Neutron configures objects on your BIG-IP device(s). This document describes how OpenStack Neutron LBaaS objects correspond to BIG-IP objects and what actions the F5 agent takes for each `neutron lbaas-loadbalancer` CLI command.

F5 LBaaSv2 uses the `f5-sdk` to communicate with BIG-IP via the F5 iControl REST API. The table below shows the corresponding iControl endpoint and BIG-IP object for each `neutron lbaas-` ‘create’ command.

<table>
<thead>
<tr>
<th>Neutron command</th>
<th>iControl REST API endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>neutron lbaas-loadbalancer-create</code></td>
<td><code>https://&lt;icontrol_endpoint&gt;:443/mgmt/tm/sys/folder/</code></td>
</tr>
<tr>
<td><code>neutron lbaas-listener-create</code></td>
<td><code>https://&lt;icontrol_endpoint&gt;:443/mgmt/tm/ltm/virtual/</code></td>
</tr>
<tr>
<td><code>neutron lbaas-pool-create</code></td>
<td><code>https://&lt;icontrol_endpoint&gt;:443/mgmt/tm/ltm/pool/</code></td>
</tr>
<tr>
<td><code>neutron lbaas-member-create</code></td>
<td><code>https://&lt;icontrol_endpoint&gt;:443/mgmt/tm/ltm/pool/</code> <del>Project_os_tenant_id</del>pool1/members/</td>
</tr>
<tr>
<td><code>neutron lbaas-healthmonitor-create</code></td>
<td><code>https://&lt;icontrol_endpoint&gt;:443/mgmt/tm/ltm/monitor/</code></td>
</tr>
</tbody>
</table>

The sections below cover the settings F5 agent applies to a standalone, overcloud BIG-IP device. The actual settings applied for a given command can vary depending on your existing BIG-IP device configurations and network architecture.

**Tip:** To view the actual API calls the F5 agent sends to the BIG-IP device(s), *set the F5 agent's DEBUG level* to ‘True’ and view the logs (`/var/log/neutron/f5-openstack-agent.log`).

1.7.1 Start the F5 Agent for OpenStack Neutron

`systemctl start f5-openstack agent` When you first start the F5 Agent for OpenStack Neutron:

- it reads the `vtep` self IP defined in the F5 agent config file;
- the F5 agent discovers the BIG-IP `vtep` IP address and advertises it to Neutron as its `tunneling_ip`;
- the F5 Driver for OpenStack LBaaSv2 adds a new port for the `vtep` to the OVS switch;
- the F5 agent adds profiles for all tunnel types to the BIG-IP device(s).

1.7.2 Create a Neutron LBaaS Load Balancer

`neutron lbaas-loadbalancer-create` The F5 agent creates the following:

- new BIG-IP partition
- BIG-IP forwarding database (FDB) records for all peers in the network
- new BIG-IP route domain
- new BIG-IP self IP on the specified subnet (this is the IP address at which the BIG-IP device can receive traffic for this load balancer)
• new tunnel (uses the vtep as the local address and the BIG-IP vxlan profile created when the F5 agent started)\(^1\)
• new SNAT pool list/SNAT translation list\(^2\)

In addition, the F5 driver adds a Neutron port for each SNAT address.

• If BIG-IP SNAT mode is off and you have \texttt{f5\_snat\_addresses\_per\_subnet} set to 0, the BIG-IP acts as a gateway and handles all return traffic from members.

• If BIG-IP SNAT mode is on and you have \texttt{f5\_snat\_addresses\_per\_subnet} set to 0, the BIG-IP device uses SNAT automap.

### 1.7.3 Create a Neutron LBaaS Listener

\texttt{neutron lbaas-listener-create} The F5 agent creates a new BIG-IP virtual server in the specified partition.

• uses the \texttt{Fast L4} protocol
• uses the IP address Neutron assigned to the load balancer
• uses the route domain created for the load balancer
• if you’re using tunnels, traffic is only handled in the tunnel assigned to the load balancer
• for secure listeners using the \texttt{TERMINATED\_HTTPS} protocol:\(^3\)
  – fetches the certificate/key container from Barbican.
  – adds the key and certificate to the BIG-IP device(s).
  – creates a custom SSL profile using \texttt{clientssl} as the parent profile.
  – adds the new SSL profile to the virtual server.

### 1.7.4 Create a Neutron LBaaS Pool

\texttt{neutron lbaas-pool-create} The F5 agent adds a new pool to the specified virtual server.

### 1.7.5 Create a Neutron LBaaS Member

\texttt{neutron lbaas-member-create} The F5 agent adds a new member to the requested pool using the specified IP address and port.

• If there is a Neutron port associated with the specified IP address and subnet, the F5 agent creates a forwarding database (FDB) entry for the member on the BIG-IP device(s).\(^4\)

• When you add a member to a pool for the first time, the BIG-IP pool status changes.

• When you create a member with a specific IP address for the first time, the F5 agent also creates a new BIG-IP node for that IP address.

\(^1\) If using \texttt{global routed mode}, F5 agent doesn’t create a tunnel. Instead, all traffic goes to the load balancer’s self IP address.

\(^2\) You can set the number of SNAT addresses to create via the \texttt{f5\_snat\_addresses\_per\_subnet} setting in the \texttt{L3 Segmentation Mode Settings} section of the F5 agent configuration file.

\(^3\) See \texttt{Set up the F5 Agent for OpenStack Barbican}.

\(^4\) The F5 agent will not create a FDB entry if the pool member IP address and subnet don’t have a corresponding Neutron port. In such cases, warning messages print to the \texttt{f5-openstack-agent} and \texttt{neutron-server} logs.
1.7.6 Create a Neutron LBaaS Health Monitor

**neutron lbaas-healthmonitor-create**  The F5 agent creates a new BIG-IP health monitor for the specified pool.

- Creating a health monitor for a pool for the first time makes the BIG-IP pool status change.
- Health monitors directly affect the status and availability of BIG-IP pools and pool members. Any additions or changes may change the status of the specified pool.

### Applies to:

<table>
<thead>
<tr>
<th>F5 LBaaS version(s)</th>
<th>OpenStack version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v8.x.x</td>
<td>Liberty</td>
</tr>
<tr>
<td>v9.x.x</td>
<td>Mitaka</td>
</tr>
<tr>
<td>v10.x.x</td>
<td>Newton</td>
</tr>
</tbody>
</table>

1.8 Deploy a basic load balancer

The series of code samples provided here demonstrate how to configure a basic loadbalancer via the OpenStack Neutron CLI with the F5 Integration for OpenStack Neutron LBaaS. The OpenStack CLI documentation has a full list of all `neutron lbaas` commands.

**Important:** The LBaaSv2 CLI commands begin with `lbaas-`. Commands beginning with `lb-` are part of the deprecated OpenStack LBaaS v1 project.

**Example:** `Project_9572afc14db14c8a806d8c8219446e7b`

1.8.1 Create a Neutron LBaaS load balancer

**Tip:** Neutron LBaaS load balancer == BIG-IP partition

The name assigned to the new BIG-IP partition follows the naming convention `<environment-prefix>_<openstack-tenant-ID>`. The default environment prefix is `Project_`. You can change the environment prefix in the F5 agent configuration file.

Specify the name you want to assign to the load balancer and the existing OpenStack subnet you want to assign to it.

```
$ neutron lbaas-loadbalancer-create --name lb1 private-subnet
```

1.8.2 Add a BIG-IP virtual server

**Tip:** Neutron LBaaS listener == BIG-IP virtual server
Specify the name you want to assign to the virtual server; the name of the load balancer (BIG-IP partition) you want to create the virtual server for; and the protocol type and port you’d like to use.

```bash
$ neutron lbaas-listener-create --name vs1 --loadbalancer lb1 --protocol HTTP --protocol-port 8080
```

### 1.8.3 Add a secure BIG-IP virtual server

**Important:** OpenStack uses the Transport Layer Security (TLS) protocol to secure network traffic. You must configure Barbican and Keystone before you can create a secure BIG-IP virtual server.

1. Set up Keystone and Barbican, if you haven’t already.²
2. Complete the Certificate Manager settings section of the F5 agent configuration file.
3. Create a listener using the TERMINATED_HTTPS protocol; specify the location of the Barbican container where the certificate you want to use for authentication lives.

   The F5 agent will add this certificate to the BIG-IP device(s) and use it to create a new BIG-IP SSL profile.

```bash
$ neutron lbaas-listener-create --name vs2 --protocol TERMINATED_HTTPS --protocol-port 8443 --loadbalancer lb1 --default-tls-container-ref http://localhost:9311/v1/containers/db50dbb3-70c2-44ea-844c-202e06203488
```

### 1.8.4 Create a pool

When you create a pool, specify the name you want to assign to the pool; the load balancing method you want to use; the name of the virtual server (listener) you want to attach the pool to; and the protocol type the pool should use.

```bash
$ neutron lbaas-pool-create --name pool1 --lb-algorithm ROUND_ROBIN --listener vs1 --protocol HTTP
$ neutron lbaas-pool-create --name pool2 --lb-algorithm ROUND_ROBIN --listener vs2 --protocol HTTPS
```

### 1.8.5 Create a pool member

When creating a pool member, specify the existing OpenStack subnet you want to assign to it; the IP address the member should process traffic on; the protocol port; and the name or UUID of the pool you want to attach the member to.

```bash
$ neutron lbaas-member-create --subnet private-subnet --address 172.16.101.89 --protocol-port 80 pool1
```

### 1.8.6 Create a health monitor

When creating a health monitor, specify the delay; monitor type; number of retries; timeout period; and the name of the pool you want to monitor.

² See OpenStack’s [How to create a TLS load balancer](link) for more information and configuration instructions.
1.8.7 What’s Next

Use the BIG-IP configuration utility to verify that all of your Neutron LBaaS objects appear on the BIG-IP device(s).

1. Log in to the BIG-IP configuration utility at the management IP address (e.g., https://1.2.3.4/tmui/login.jsp).
2. Use the Partition drop-down menu to select the correct partition for your load balancer.
3. Go to Local traffic → Virtual Servers to view your new virtual server.
4. Click on the virtual server name to view the pool, pool member, and health monitor.

1.9 Set up SSL offloading with OpenStack Barbican

<table>
<thead>
<tr>
<th>F5 LBaaS version(s)</th>
<th>OpenStack version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v8.x.x</td>
<td>Liberty</td>
</tr>
<tr>
<td>v9.x.x</td>
<td>Mitaka</td>
</tr>
<tr>
<td>v10.x.x</td>
<td>Newton</td>
</tr>
</tbody>
</table>

OpenStack Barbican is OpenStack’s certificate management service. It provides a secure location where users can store secrets such as SSH keys, private keys, certificates, and user passwords. The F5 Agent for OpenStack Neutron can use Barbican certificates to perform BIG-IP SSL offloading.

1.9.1 Prerequisites

- Administrator access to both BIG-IP device(s) and OpenStack cloud.
- OpenStack Barbican certificate manager configured and operational.
- Existing BIG-IP SSL profile (optional).

1.9.2 Configure SSL offloading using OpenStack Barbican Secrets

Edit the Certificate Manager Settings section of the F5 agent configuration file.

1. Enable the F5 Barbican certificate manager.

```bash
cert_manager = f5_openstack_agent.lbaasv2.drivers.bigip.barbican_cert. BarbicanCertManager
```

2. Provide the Keystone authentication data for your environment.
### Table 1.5: Keystone authentication data

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth_version</td>
<td>Keystone version (v2 or v3)</td>
</tr>
<tr>
<td>os_auth_url</td>
<td>Keystone authentication URL</td>
</tr>
<tr>
<td>os_username</td>
<td>OpenStack username</td>
</tr>
<tr>
<td>os_password</td>
<td>OpenStack password</td>
</tr>
<tr>
<td>os_tenant_name</td>
<td>OpenStack tenant name (v2 only)</td>
</tr>
<tr>
<td>os_user_domain_name</td>
<td>OpenStack domain in which the user account resides (v3 only)</td>
</tr>
<tr>
<td>os_project_name</td>
<td>OpenStack project name (v3 only; this is the same data as os_tenant_name in v2)</td>
</tr>
<tr>
<td>os_project_domain_name</td>
<td>OpenStack domain in which the project resides</td>
</tr>
</tbody>
</table>

```python
#
cert_manager = f5_openstack_agent.lbaasv2.drivers.bigip.barbican_cert.
   BarbicanCertManager
#
...
# Keystone v2 authentication:
#
# auth_version = v2
# os_auth_url = http://localhost:5000/v2.0
# os_username = admin
# os_password = changeme
# os_tenant_name = admin
#
# Keystone v3 authentication:
#
auth_version = v3
os_auth_url = http://localhost:5000/v3
os_username = myusername
os_password = mypassword
os_user_domain_name = default
os_project_name = myproject
os_project_domain_name = default
#
```

3. Set the parent BIG-IP SSL profile.

```python
#
f5_parent_ssl_profile = clientssl
#
```

**Tip:** The profile F5 agent creates on the BIG-IP device inherit settings from the parent you define. The profile must already existing on the BIG-IP device; if it does not, F5 agent uses `clientssl` as the default parent profile.

### 1.9.3 Learn more

Once you’ve added secrets to a Barbican container, you can reference the container’s URI in `neutron lbaas` commands.
1.9.4 Use Case

When you configure Client SSL or Server SSL profiles and assign them to a virtual server, the BIG-IP device handles the SSL processing. This conserves resources on the destination servers and lets you enforce custom BIG-IP SSL processing rules. When the F5 agent adds a Barbican certificate to a BIG-IP device, it can either inherit settings from an existing BIG-IP SSL profile or create a new SSL profile on the device.

You can use Client SSL (the most common use case) to decrypt client requests before sending them on to the destination server and encrypt server responses before sending them back to the client.

See also:
- Create a secure BIG-IP virtual server

1.10 L7 Routing

L7 routing takes its name from layer 7 of the OSI Model, also called the application layer. The F5 Agent for OpenStack Neutron passes L7 content switching policies and rules from OpenStack Neutron to a BIG-IP device. The

BIG-IP device then processes the application data of request traffic as it passes through a virtual server and applies corresponding Local Traffic Manager (LTM) policies and rules to make routing decisions. You can use the OpenStack API to define the content conditions and the actions that they should trigger.

**Important:** See the OpenStack Neutron LBaaS Layer 7 rules documentation for a full explanation of the L7 rules and L7 policies.

It is essential to understand these concepts before proceeding with this document.

### 1.10.1 Neutron LBaaSv2 API L7 Policies and Rules

In Neutron, an L7 Policy is a collection of L7 rules associated with a Listener; it may also have an association to a back-end pool. Policies describe actions that the load balancing software should take if all of the rules in the policy return “true” (or, in other words, they match).

### 1.10.2 OpenStack Policy/Rules versus BIG-IP Local Traffic Manager Policy/Rules

The Neutron L7 terminology does not directly align with the common vocabulary of BIG-IP Local Traffic Manager (LTM). Keep the following key differences in mind:

- BIG-IP LTM policies also have a set of rules; in LTM, it is the rules, not the policies, that specify what action to take.
- BIG-IP devices evaluate policies attached to a virtual server regardless of whether the associated rules are true.
- BIG-IP LTM rules, not policies, have an ordinal.

The table below shows how the terms from each software domain correspond.

<table>
<thead>
<tr>
<th>Neutron LBaaS L7 term</th>
<th>BIG-IP LTM equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy</td>
<td>Policy Rules (wrapper_policy)</td>
</tr>
<tr>
<td>Policy Action</td>
<td>Rule Action</td>
</tr>
<tr>
<td>Policy Position</td>
<td>Rule Ordinal</td>
</tr>
<tr>
<td>Rule</td>
<td>Rule Conditions</td>
</tr>
</tbody>
</table>

The BIG-IP LTM policy has a name, description, set of rules, and a strategy defining how to evaluate the rules. OpenStack L7 policies in are similar to a collection BIG-IP LTM policy rules evaluated using the ‘First match’ strategy.

BIG-IP LTM rules have conditions, actions, and an ordinal. The F5 agent creates the LTM rules based on the OpenStack L7 policy and rule attributes.

**Neutron LBaaSv2 API L7 rules to BIG-IP LTM policy mapping**

The F5 agent maps a combination of L7Policy and L7Rule elements to TMOS traffic policies (and, in the case of specific L7Rule compare_types, iRules).

All L7 Rule types map directly to TMOS traffic policy match conditions:
<table>
<thead>
<tr>
<th>L7 Rule Type</th>
<th>TMOS Traffic Policy Match Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
<td>HTTP Host</td>
</tr>
<tr>
<td>Path</td>
<td>HTTP URI + path</td>
</tr>
<tr>
<td>File_Type</td>
<td>HTTP URI + extension</td>
</tr>
<tr>
<td>Header</td>
<td>HTTP Header</td>
</tr>
<tr>
<td>Cookie</td>
<td>HTTP Cookie</td>
</tr>
</tbody>
</table>

The LBaaS L7 Rules requirement to “execute the first L7Policy that returns a match” directly maps to the TMOS “first-match” traffic policy execution strategy.

Four of the five L7Rule compare_type values map directly to TMOS traffic policy rule conditions:

<table>
<thead>
<tr>
<th>L7 Rule Compare Type</th>
<th>L7 ‘–invert’ Specified</th>
<th>TMOS Traffic Policy Rule Match Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTS_WITH</td>
<td>No</td>
<td>Begins with</td>
</tr>
<tr>
<td>STARTS_WITH</td>
<td>Yes</td>
<td>Does not begin with</td>
</tr>
<tr>
<td>ENDS_WITH</td>
<td>No</td>
<td>Ends with</td>
</tr>
<tr>
<td>ENDS_WITH</td>
<td>Yes</td>
<td>Does not end with</td>
</tr>
<tr>
<td>EQUAL_TO</td>
<td>No</td>
<td>Is</td>
</tr>
<tr>
<td>EQUAL_TO</td>
<td>Yes</td>
<td>Is not</td>
</tr>
<tr>
<td>CONTAINS</td>
<td>No</td>
<td>Contains</td>
</tr>
<tr>
<td>CONTAINS</td>
<td>Yes</td>
<td>Does not contain</td>
</tr>
<tr>
<td>REGEX</td>
<td>X</td>
<td>No direct mapping</td>
</tr>
</tbody>
</table>

All L7Policy actions map directly to TMOS traffic policy rule actions:

<table>
<thead>
<tr>
<th>L7 Policy Action</th>
<th>TMOS Traffic Policy Rule Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reject</td>
<td>Reset traffic</td>
</tr>
<tr>
<td>RedirectToUrl</td>
<td>Redirect</td>
</tr>
<tr>
<td>RedirectToPool</td>
<td>Forward traffic to pool</td>
</tr>
</tbody>
</table>

**Caveats**

- The REGEX comparison type is not supported in this release.

**Usage**

L7 Routing doesn’t require any F5 agent configuration changes. Rather, you define L7 switching policies and rules when creating or updating a Neutron LBaaS listener.

The **CLI example** below from the OpenStack Neutron Wiki demonstrates how to define rules and policies using the OpenStack CLI.

```bash
# Create a listener
neutron lbaas-create-listener listener1
# Create a pool
neutron lbaas-create-pool pool1
# Create a policy
neutron --policy policy1 lbaas-create-l7policy --name "policy1" --listener "listener1" -- --action redirect_to_pool --pool "pool1" --position 1
# Create a rule for this policy
```

---

2 OpenStack Foundation
Once the below operation has completed, a new policy will exist on the device called 'wrapper_policy'.

It will have a single rule called redirect_to_pool_1.

The F5 agent implements the above Neutron LBaaS policies and rules on the BIG-IP device as follows.

```bash
ltm policy wrapper_policy {
    controls { forwarding }
    last-modified 2016-12-05:09:19:05
    partition Project_9065d69e806a4b4894a47fed7484a006
    requires { http }
    rules {
        reject_1 {
            actions {
                0 {
                    forward
                    reset
                }
            }
            conditions {
                0 {
                    http-uri
                    path
                    contains
                    values { i_t }
                }
                1 {
                    http-cookie
                    name cky
                    ends-with
                    values { i }
                }
            }
            ordinal 1
        }
    }
    status legacy
    strategy /Common/first-match
}
```

Learn more

- OpenStack Neutron LBaaS Layer 7 rules documentation
- OpenStack Neutron LBaaSv2 l7 Wiki
- BIG-IP Local Traffic Management – Getting Started with Policies
1.11 Redundancy and Scale Out

Running the F5 agent on different hosts helps ensure that the F5 agent and F5 driver processes remain alive and functional if a host goes down. Multiple F5 agent instances can manage a single BIG-IP device or cluster if each runs on a separate host.

1.11.1 Prerequisites

- Administrator access to both the BIG-IP device(s) and the OpenStack cloud.
- All hosts running the F5 Integration for OpenStack must have the Neutron and Neutron LBaaS packages installed.
- All hosts running the F5 Integration for OpenStack must use the same Neutron database.

1.11.2 Caveats

- You can not run multiple F5 agent instances on the same host to manage the same BIG-IP device or cluster.
- In the standard multi-agent deployment, you can’t specify which F5 agent instance you want to use to create a new load balancer (meaning you can’t choose which BIG-IP device/cluster to create the new partition on).
- Use differentiated service environments if you need a greater degree of control over which F5 agent instance(s) handle specific LBaaS requests.

1.11.3 Configuration

1. Copy the Neutron and Neutron LBaaS configuration files from the Neutron controller to each host on which you want to run an F5 agent instance.

   ```bash
cp /etc/neutron/neutron.conf <hostname>:/etc/neutron/neutron.conf
cp /etc/neutron/neutron_lbaas.conf <hostname>:/etc/neutron/neutron_lbaas.conf
```

2. Install the F5 Agent for OpenStack Neutron on each host.

3. Copy your F5 agent configuration file from the Neutron controller to each host.

   ```bash
cp /etc/neutron/services/f5/f5-openstack-agent.ini <hostname>:/etc/neutron/services/f5/f5-openstack-agent.ini
```

Tip:
• If you are managing an active-standby pair or cluster with config sync turned on:\footnote{Using configuration synchronization in clusters managed by the F5 Integration for OpenStack Neutron LBaaS is not recommended. See \textit{Manage BIG-IP clusters} for more information.}
  – Set the \texttt{ha\_type} to \texttt{standalone}.
  – Provide the iControl endpoint for one (1) of the BIG-IP devices in the cluster.

• If you are managing a cluster that has config sync turned on for a device group within the cluster:
  – Set the \texttt{ha\_type} to \texttt{pair} or \texttt{scalen}.
  – Provide the iControl endpoint for one (1) of the BIG-IP devices in the device group and the endpoint for a device outside the group (\texttt{pair}).
  – \texttt{OR}–
  – Provide the iControl endpoint for one (1) of the BIG-IP devices in the device group and the endpoint for each device in the cluster that is not automatically syncing its configurations with the group. (\texttt{scalen})

4. Start the F5 agent on each host.

\textbf{CENTOS}

\begin{verbatim}
systemctl enable f5-openstack-agent
systemctl start f5-openstack-agent
\end{verbatim}

\textbf{UBUNTU}

\begin{verbatim}
service f5-oslbaasv2-agent start
\end{verbatim}

\subsection{1.11.4 Learn more}

Spreading the request load for an environment across multiple F5 agent instances helps to avoid F5 agent overload and loss of functionality.

If you are well versed in containerized environments, you can run each F5 agent instance in a separate container on your Neutron controller. If the service provider driver is in the container’s build context, you don’t need to install it in each container.

- The \texttt{neutron.conf} and \texttt{neutron-lbaas.conf} files must be present in each container.
- The service provider driver does not need to run in the container if you’re building from the Neutron controller.

\begin{shaded}
\textbf{Warning:} F5 Networks does not support container service deployments in OpenStack.
\end{shaded}

See also:

- Configure the F5 Agent for OpenStack Neutron
- Manage BIG-IP Clusters with F5 LBaaSv2
- Differentiated Service Environments
1.12 Manage BIG-IP Clusters

You can use the F5 Integration for OpenStack Neutron LBaaS to manage BIG-IP device service clusters with high availability, mirroring, and failover services in your OpenStack cloud.

Clustering provides a greater degree of redundancy than a standalone device offers. It helps to avoid service interruptions that could otherwise occur if a device should go down.

Learn more

1.12.1 Prerequisites

- Administrator access to both BIG-IP devices and OpenStack cloud.
- Licensed, operational BIG-IP device service cluster.

Tip: If you do not already have a BIG-IP cluster deployed in your network, you can use the F5 BIG-IP: Active-Standby Pair Heat template to create an overcloud sync-failover device group.

1.12.2 Caveats

- The F5 Agent for OpenStack Neutron can manage clusters of two (2) to four (4) BIG-IP devices. Active-standby, or “pair”, mode applies to two-device clusters; scalen applies to clusters of more than two (2) devices.
- The administrator login must be the same on all BIG-IP devices in the cluster.
- F5 strongly advises against using configuration synchronization in clusters managed by the F5 agent.

1.12.3 Configuration

Edit the device settings and Device Driver/iControl driver settings sections of the F5 agent configuration file.

1. Set the HA mode to pair or scalen.

```
vi /etc/neutron/services/f5/f5-openstack-agent.ini
...
# HA mode
#
 f5_ha_type = pair   \ 2-device cluster
 f5_ha_type = scalen \ 2-4 device cluster
#
#
```
2. Add the iControl endpoint (IP address) for each BIG-IP device in the cluster and the admin login credentials. Values must be comma-separated.

```
# icontrol_hostname = 1.2.3.4,5.6.7.8
# icontrol_username = myusername
# icontrol_password = mypassword
#
```

### 1.12.4 Learn more

The F5 Integration for OpenStack Neutron LBaaS can manage BIG-IP Sync-Failover device groups when you set High Availability mode to pair or scalen.

![Diagram of BIG-IP scalen cluster](image)

Fig. 1.4: BIG-IP scalen cluster

The F5 agent expects to find a specific number of iControl endpoints (the `icontrol_hostname` configuration parameter) based on the `f5_ha_type`, as noted below.
Table 1.7: F5 Integration for OpenStack Neutron LBaaS high availability (HA) options

<table>
<thead>
<tr>
<th>HA type</th>
<th>Number of iControl endpoints expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>standalone</td>
<td>1</td>
</tr>
<tr>
<td>pair</td>
<td>2</td>
</tr>
<tr>
<td>scalen</td>
<td>&gt; 2</td>
</tr>
</tbody>
</table>

**F5 LBaaSv2 and BIG-IP Auto-sync**

**Important:** The F5 Agent for OpenStack Neutron applies LBaaS configuration changes to each BIG-IP device in a cluster at the same time, in real time. For this reason, **do not** use configuration synchronization (config sync) in clusters managed by the F5 Integration for OpenStack Neutron LBaaS.

For example, if you create a load balancer for a device group using config sync, the create command will succeed on the first device in the group and fail on the others. The failure occurs because config sync has already created the requested partition on each device in the cluster.

If you need to sync a BIG-IP device group, do so manually after making changes to Neutron LBaaS objects.

**Danger:** If you must use config sync mode, set the `f5_ha_type` to `standalone` and enter the iControl endpoint for one (1) of the BIG-IP devices in the group.

If you choose to do so, **you must manually replace the iControl endpoint** in the F5 agent configuration file with the iControl endpoint of another device in the group if the configured device should fail.

While it is possible to use config sync for a device group after creating a new load balancer, it is not recommended. **F5 has not tested or verified this functionality.**

See also:
- *Manage BIG-IP vCMP clusters*

**Applies to:**

<table>
<thead>
<tr>
<th>F5 LBaaS version(s)</th>
<th>OpenStack version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v8.1+</td>
<td>Liberty</td>
</tr>
<tr>
<td>v9.1+</td>
<td>Mitaka</td>
</tr>
<tr>
<td>v10.0+</td>
<td>Newton</td>
</tr>
</tbody>
</table>

### 1.13 Manage BIG-IP vCMP Systems

You can manage a BIG-IP using Virtual Clustered Multiprocessing (vCMP) with the F5 Integration for OpenStack Neutron LBaaS.

*Learn more*
1.13.1 Prerequisites

- Licensed, operational BIG-IP chassis with support for vCMP.
- Licensed, operational BIG-IP vCMP guest running on a vCMP host.
- Administrative access to the vCMP host(s) and guest(s) you will manage with F5 LBaaSv2.
- F5 Agent for OpenStack Neutron and F5 Driver for OpenStack LBaaSv2 installed on the Neutron controller.
- Knowledge of BIG-IP vCMP configuration and administration.

1.13.2 Caveats

- VLAN and FLAT are the only network types supported for use with vCMP.

1.13.3 Configuration

Edit the Device driver/iControl driver settings and L2 Segmentation Mode settings sections of the F5 agent configuration file.

1. Add the `icontrol_vcmp_hostname`. Multiple values can be comma-separated.

```ini
# icontrol_vcmp_hostname = 1.2.3.4
#
```

2. Configure the `icontrol_hostname` parameter with the IP address(es) of the vCMP guest(s):

```ini
# icontrol_hostname = 10.11.12.13, 14.15.16.17
#
```

3. Set `advertised_tunnel_types` to `vlan` or `flat`, as appropriate for your environment.

   **Tip:** Leave the `advertised_tunnel_types` setting empty (as in the example below) if the ML2 plugin `provider:network_type` is FLAT or VLAN.

```ini
# advertised_tunnel_types =
#
```

1.13.4 Learn more

Virtual Clustered Multiprocessing lets you run multiple instances of BIG-IP software on a single hardware platform. The vCMP host allocates a share of the hardware resources to each vCMP guest. Each guest has its own management IP address, self IP addresses, virtual servers, and so on. Each guest can receive and process application traffic with no knowledge of other guests on the system. vCMP allows you to delegate management of BIG-IP instances, meaning users who need to manage LBaaS objects don’t need to have full administrative access to the BIG-IP device.
1.13.5 Use Case

The F5 agent can manage one (1) or more vCMP hosts with one (1) or more guests per host. You can configure vCMP hosts as a device service cluster. 1 If a vCMP host fails (taking its guests with it), another vCMP host in the cluster can take over managing its traffic.

1.14 Manage a single BIG-IP device with multiple Controller instances

<table>
<thead>
<tr>
<th>Applies to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>F5 LBaaS version(s)</td>
</tr>
<tr>
<td>v8.1+</td>
</tr>
<tr>
<td>v9.1+</td>
</tr>
<tr>
<td>v10.0+</td>
</tr>
</tbody>
</table>

You can manage the same BIG-IP device or cluster with multiple instances of the F5 Agent for OpenStack Neutron running in differentiated service environments. A differentiated service environment is a uniquely-named environment that has:

- a dedicated F5 driver,
- a dedicated messaging queue, and
- a dedicated F5 agent.

In a multiple-agent setup, each F5 agent manages a distinct environment that corresponds to a specific BIG-IP partition.

**Important:**

- The F5 environment generator, a tool built in to the F5 Driver for OpenStack LBaaSv2, creates new service environments for you and configures Neutron to use the new service provider drivers.
- Differentiated service environments are not compatible with Virtual Clustered Multiprocessing (vCMP) systems. BIG-IP devices cannot share data or resources across differentiated service environments; this precludes the use of vCMP because vCMP guests share global VLAN IDs.

1.14.1 Set up a new service environment

1. Generate a new custom environment on the Neutron controller.

Listing 1.2: Example: Add a custom environment called “dev1”

```
add_f5agent_environment dev1
```

**Tip:** The environment name field must be eight (8) characters or less.

---

1 See Device Service Clustering for vCMP Systems
2. Check the Neutron LBaaS configuration file to verify that the new service provider driver is active.

```
less /etc/neutron/neutron_lbaas.conf
...
[service_providers]
service_provider = LOADBALANCERV2:F5Networks:neutron_lbaas.drivers.f5.driver_v2.
→F5LBaaSV2Driver:default
service_provider = LOADBALANCERV2:dev1:neutron_lbaas.drivers.f5.driver_v2.
→F5LBaaSV2Driver
...
```

1.14.2 Set up F5 agent to use the new environment

In the F5 agent configuration file:

1. Replace the default `environment_prefix` with the name of the new service environment.

```
vi /etc/neutron/services/f5/f5-openstack-agent.ini
#
# environment_prefix = 'dev1'
#
```

2. Add/update the iControl endpoints and login credentials for the BIG-IP devices you want to include in the service group.

```
#
icontrol_hostname = 1.2.3.4, 5.6.7.8
#
...
#
icontrol_username = myusername
...
#
icontrol_password = mypassword
#
```

3. Save the file with a new name.

```
:w f5-openstack-agent_dev1.ini
```

Set up the new environment on additional hosts

[OPTIONAL]

Take the step below if you want to run the F5 agent in differentiated service environments on multiple hosts.\(^1\)

1. Copy the F5 agent, Neutron, and Neutron LBaaS configuration files from the Neutron controller to each additional host.

\(^1\) Running F5 agent instances on one (1) or more additional hosts provides redundancy and a degree of protection against individual host failure. See *F5 Agent for OpenStack Neutron Redundancy and Scale-out* for more information.
1.14.3 Restart the services

1. Restart Neutron.

```
systemctl restart neutron-server     \ CENTOS
service neutron-server restart      \ UBUNTU
```

2. Restart the F5 agent.

**CENTOS**

```
systemctl restart f5-openstack-agent
```

**UBUNTU**

```
service f5-oslbaasv2-agent restart
```

**Important:** Restart the F5 agent on each host to which you copied the updated configuration file.

1.14.4 Create a load balancer in the new service environment

1. When you create a new load balancer, pass in the name of the new service environment using the `--provider` flag.

```
(neutron) lbaas-loadbalancer-create --name lb_dev1 --provider dev1 b3fa44a0-3187-4a49-853a-24819bc24d3e
```

Created a new loadbalancer:

```
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| admin_state_up      | True                                 |
| description         |                                      |
| id                  | fcd874ce-6dad-4aef-9e69-98d1590738cd |
| listeners           |                                      |
| name                | lb_dev1                              |
| operating_status    | OFFLINE                              |
| provider            | dev1                                 |
| provisioning_status | PENDING_CREATE                       |
| tenant_id           | 1b2b505dafbc487fb805c6c9de9459a7      |
| vip_address         | 10.1.2.7                             |
| vip_port_id         | 079eb9e5-dc63-4dbf-bc15-f38f5fdeee92  |
| vip_subnet_id       | b3fa44a0-3187-4a49-853a-24819bc24d3e  |
+---------------------+--------------------------------------+
```

1.14. Manage a single BIG-IP device with multiple Controller instances 35
1.14.5 Learn more

When the F5 Agent for OpenStack Neutron uses the default service environment prefix – Project – the F5 Driver for OpenStack LBaaSv2 assigns LBaaS tasks to each F5 agent instance from the global messaging queue.

When you create a new service environment (for example, “dev”, “prod”, “test”, etc.), you’re really creating a new LBaaS service provider driver and uniquely-named messaging queue. The F5 environment generator creates the driver and adds it to the service providers list in the Neutron LBaaS config file. When you issue a `neutron lbaas-loadbalancer-create` command with the `--provider` flag, that F5 driver instance receives the task in its dedicated messaging queue; it then assigns the task to an F5 agent instance in its environment group. By default, F5 agent instances in an environment group receive tasks in a round-robin fashion; you can set up capacity-based scale out for a greater degree of control over how the F5 Driver for OpenStack LBaaSv2 chooses which F5 agent instances receive tasks.

**Use Case**

Use differentiated service environments if:

1. You want to run multiple F5 agent instances **on the same host** to manage the same BIG-IP device/cluster. Each unique service environment corresponds to a distinct BIG-IP partition, so the F5 agent processes don’t overlap and cause errors.

2. You want a finer degree of control over which BIG-IP device the F5 agent creates LBaaS objects on. In the default set-up, the F5 Driver for OpenStack LBaaSv2 assigns tasks from the global messaging queue to the first available F5 agent instance it finds. This means that, when using the default environment, you can’t control which BIG-IP device gets any given object. Custom service environments allow you to specify which F5 agent instance/group – and, therefore, which BIG-IP device/cluster – should handle a given LBaaS task.

### Applies to:

<table>
<thead>
<tr>
<th>F5 LBaaS version(s)</th>
<th>OpenStack version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v8.1+</td>
<td>Liberty</td>
</tr>
<tr>
<td>v9.1+</td>
<td>Mitaka</td>
</tr>
<tr>
<td>v10.0+</td>
<td>Newton</td>
</tr>
</tbody>
</table>

1.15 Capacity-Based Scale Out

When using **differentiated service environments**, you can configure capacity metrics for the F5 Agent for OpenStack Neutron to provide scale out across multiple BIG-IP device groups.

**Learn more**

### 1.15.1 Prerequisites

- Administrator access to both the BIG-IP devices and the OpenStack cloud.
• F5 Agent for OpenStack Neutron installed on all hosts.
• F5 Driver for OpenStack LBaaSv2 installed on the Neutron controller.
• One F5 OpenStack service provider driver instance installed on the Neutron controller for each custom environment you want to use.

1.15.2 Caveats

• All hosts running the F5 Integration for OpenStack Neutron LBaaS must use the same Neutron database.
• You can not manage a single BIG-IP device or cluster with multiple F5 agent instances running in the same service environment on the same host.¹

1.15.3 Configuration

Edit the following items in the F5 agent configuration file.

1. Set the desired `environment_group_number`.

```
# Environment Settings
environment_group_number = 1
```

2. Provide the iControl endpoint and login credentials for one (1) of the BIG-IP devices in the device group.

```
icontrol_hostname = 1.2.3.4
icontrol_username = myusername
icontrol_password = mypassword
```

3. Define the capacity score metrics.

¹ See differentiated service environments for information about running multiple F5 agent instances on the same host.
Table 1.8: Capacity score settings

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>throughput</td>
<td>total throughput in bps of the TMOS devices</td>
</tr>
<tr>
<td>inbound_throughput</td>
<td>throughput in bps inbound to TMOS devices</td>
</tr>
<tr>
<td>outbound_throughput</td>
<td>throughput in bps outbound from TMOS devices</td>
</tr>
<tr>
<td>active_connections</td>
<td>number of concurrent active actions on a TMOS device</td>
</tr>
<tr>
<td>tenant_count</td>
<td>number of tenants associated with a TMOS device</td>
</tr>
<tr>
<td>node_count</td>
<td>number of nodes provisioned on a TMOS device</td>
</tr>
<tr>
<td>route_domain_count</td>
<td>number of route domains on a TMOS device</td>
</tr>
<tr>
<td>vlan_count</td>
<td>number of VLANS on a TMOS device</td>
</tr>
<tr>
<td>tunnel_count</td>
<td>number of GRE and VxLAN overlay tunnels on a TMOS device</td>
</tr>
<tr>
<td>ssltps</td>
<td>the current measured SSL TPS count on a TMOS device</td>
</tr>
<tr>
<td>clientssl_profile_count</td>
<td>the number of clientside SSL profiles defined</td>
</tr>
</tbody>
</table>

```python
# Environment Settings

# capacity_policy = throughput:1000000000, active_connections: 250000, route_domain_count: 512, tunnel_count: 2048
```

### 1.15.4 Learn more

The F5 agent `environment_group_number` and `environment_capacity_score` configuration parameters allow the F5 Driver for OpenStack LBaaSV2 to assign requests to the group that has the lowest capacity score. The `environment_group_number` provides a convenient way for the F5 driver to identify F5 agent instances that are available to handle requests for any of the devices in a given group.

You can configure a variety of capacity metrics via the `capacity_policy` configuration parameter. These metrics contribute to the overall `environment_capacity_score` for the environment group. Each F5 agent instance calculates the capacity score for its group and reports the score back to the Neutron database.

To find the capacity score, the F5 agent divides the collected metric by the max specified for that metric in the `capacity_policy` configuration parameter. An acceptable reported `environment_capacity_score` is between zero (0) and one (1). If an agent instance in the group reports an `environment_capacity_score` of one (1) or greater, the device is at capacity.

As demonstrated in the figure, when the F5 driver receives a new LBaaS request, it consults the Neutron database. It uses the `environment_group_number` and the group’s last reported `environment_capacity_score` to assign the task to the group with the lowest utilization. The F5 driver then selects an F5 agent instance from the group (at random) to handle the request.

If any F5 agent instance has previously handled requests for the specified tenant, that F5 agent instance receives the task. If that F5 agent instance is a member of a group for which the last reported `environment_capacity_score` is above capacity, the F5 driver assigns the request to an F5 agent instance in a different group where capacity is under the limit.

**Danger:** If all F5 agent instances in all environment groups are at capacity, **LBaaS service requests will fail.** LBaaS objects created in an environment that has no capacity left will show an error status.
1. F5 LBaaS version(s) | OpenStack version(s)
---|---
v8.1+ | Liberty
v9.1+ | Mitaka
v10.0+ | Newton

Neutron **hierarchical port binding** allows you to use the F5 Integration for OpenStack Neutron LBaaS with software-defined networking (SDN). Once you tell the F5 Agent for OpenStack Neutron what top of rack (ToR) L3 switch and port (in other words, which network segment) the BIG-IP devices connect to, it can connect LBaaS services to the BIG-IP device(s) for dynamically-created VLANs in that segment.
Hierarchical port binding allows for the creation of disconnected services. The F5 agent polls the Neutron database looking for the VLANs requested for the disconnected services. When it discovers the VLANs, the F5 agent creates the requested objects on the BIG-IP device(s).

### 1.16.1 Prerequisites

- Your BIG-IP license must include SDN services (Better or Best).

### 1.16.2 Caveats

- VLAN is the only ML2 network type supported for use with hierarchical port binding.
- Each F5 agent instance managing a BIG-IP device service cluster must use the same f5_network_segment_physical_network.¹
- All F5 agent instances in a service environment group must use the same binding settings.

**Set up the F5 Agent for OpenStack Neutron to use hierarchical port binding**

1. Edit the F5 agent configuration file:

   ```
   vi /etc/neutron/services/f5/f5-openstack-agent.ini
   ```

2. Set the hierarchical port binding settings in the L2 Segmentation Mode Settings section as appropriate for your environment.

   **Listing 1.4: Hierarchical Port Binding Example**

   ```
   #
   f5_network_segment_physical_network = edgeswitch002ports0305
   #
   f5_network_segment_polling_interval = 10
   #
   f5_pending_services_timeout = 60
   ```

**Learn more**

### 1.16.3 Disconnected Services

Disconnected services are LBaaS objects for which the designated Neutron network isn’t bound to physical network segment yet. When the F5 agent discovers the designated Neutron network, the “disconnected services” connect to it automatically. The F5 agent polling frequency and “pending services timeout” allow for a degree of variation in the timing of the VLAN deployment and the request to create the LBaaS objects for it.

### 1.16.4 Use Case

Use hierarchical port binding if you want your undercloud physical BIG-IP device or cluster to control traffic for networks dynamically created via SDN. As noted in the OpenStack documentation, this can be useful if you need your Neutron deployment to scale beyond the 4K-VLANs-per-physical network limit.²

---

¹ See Agent Redundancy and Scale Out
² OpenStack ML2 Hierarchical Port Binding specs.
1. BIG-IP cluster connects to a top of rack switch in the data center.
2. SDN dynamically creates VLANS in the same network segment as BIG-IP cluster.
3. The new VLANS are added to OpenStack Neutron.
4. F5 LBaaSv2 Driver discovers the VLAN IDs in the Neutron ML2 database.
5. F5 LBaaSv2 Driver notifies the F5 OpenStack BIG-IP Controller of the new VLANS.
6. F5 OpenStack BIG-IP Controller creates VLAN tags and configures the BIG-IP devices.

Fig. 1.6: F5 LBaaSv2 Hierarchical Port Binding
1.17 F5 Integration for OpenStack Neutron LBaaS - Troubleshooting

**Important:** If you can’t find objects you created using the F5 Integration for OpenStack Neutron LBaaS on your BIG-IP device, check your partition. By default, the F5 Agent for OpenStack Neutron creates all objects in a partition named with the environment prefix (default is `Project_`) and the OpenStack tenant ID.

Example: `Project_9572afc14db14c8a806d8c8219446e7b`

1.17.1 Set the Logging Level to DEBUG

To troubleshoot general problems, set the Neutron and F5 agent `debug` setting to `True`. This creates extensive logs in `/var/log/neutron/neutron-server.log` and `/var/log/neutron/f5-openstack-agent.log`, respectively.

1. Set the DEBUG log level output for Neutron:

```bash
sudo vi /etc/neutron/neutron.conf
[DEFAULT]
...
# Print debugging output (set logging level to DEBUG instead of default WARNING).
 debug = True
```

2. Set the DEBUG log level output for F5 agent.

```bash
sudo vi /etc/neutron/services/f5/f5-openstack-agent.ini
#
[DEFAULT]
# Show debugging output in log (sets DEBUG log level output).
 debug = True
...
```

1.17.2 What to do if the F5 Agent for OpenStack Neutron isn’t running

Check the agent list in the OpenStack Horizon dashboard, or run `neutron agent-list` on the Neutron controller (or any other host where you installed F5 agent).

If `f5-openstack-agent`, or `f5-oslbaasv2-agent`, doesn’t appear in the agent list the F5 agent isn’t running.

**Here are a few things you can try:**

- Check the logs for error messages.

  ```bash
grep "ERROR" /var/log/neutron/f5-openstack-agent.log
  ```

- Check the status of the `f5-openstack-agent` service.

  ```bash
  sudo systemctl status f5-openstack-agent.service \ CENTOS
  sudo service f5-oslbaasv2-agent status \ UBUNTU
  ```

- Make sure you can connect to the BIG-IP
and verify that the following entries in the F5 agent configuration file are correct:

- iControl hostname (can be a DNS-recognized hostname or the BIG-IP device’s management IP address)
- username (account must have permission to manage LTM objects in the OpenStack tenant’s partition; see
  )
- password

• When using **Global Routed Mode**:

Comment out (#) the `vtep` lines (shown below) in the F5 agent configuration file.

```bash
#f5_vtep_folder = 'Common'
#f5_vtep_selfip_name = 'vtep'
```

• When using **L2/L3 segmentation mode**:

Verify that the `advertised_tunnel_types` setting in the F5 agent configuration file matches the Neutron network’s `provider:network_type`. If the settings don’t match, check your network configurations and make corrections as needed.

```bash
neutron net-show <network_name>
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin_state_up</td>
<td>True</td>
</tr>
<tr>
<td>id</td>
<td>05f61e74-37e0-4c30-a664-762dfef1a221</td>
</tr>
<tr>
<td>mtu</td>
<td>0</td>
</tr>
<tr>
<td>name</td>
<td>bigip_external</td>
</tr>
<tr>
<td>provider:network_type</td>
<td>vxlan</td>
</tr>
<tr>
<td>provider:physical_network</td>
<td></td>
</tr>
<tr>
<td>provider:segmentation_id</td>
<td>84</td>
</tr>
<tr>
<td>router:external</td>
<td>False</td>
</tr>
<tr>
<td>shared</td>
<td>False</td>
</tr>
<tr>
<td>status</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>subnets</td>
<td></td>
</tr>
<tr>
<td>tenant_id</td>
<td>1a35d6558b59423e83f4500f1ebc1c6c</td>
</tr>
</tbody>
</table>

### 1.17.3 F5 Agent for OpenStack Neutron does not handle LBaaS requests correctly

**Verify that you only have one agent running per environment, per host**

If you see more than one entry for `f5-openstack-agent` or `f5-oslbaasv2-agent`, and you haven’t configured your host to use **multiple agents**, deactivate one of them.

The commands below may help you to identify which agent to deactivate.

```bash
neutron agent-list \ list all running agents
neutron agent-show <agent_id> \ show the details for a specific agent
```
neutron lbaas-loadbalancer-list-on-agent <agent_id> \ list the loadbalancers on an agent.
neutron lbaas-loadbalancer-show <loadbalancer_id> \ show the details for a specific load balancer

Make sure you’re not running LBaaSv1 and LBaaSv2 at the same time

1. Remove the entry for the lbaasv1 plugin from the Neutron configuration file (/etc/neutron/neutron.conf), if it exists.

```
[service_plugins]
service_plugin = router,neutron_lbaas.services.loadbalancer.plugin.
    \ LoadBalancerPluginv2 CORRECT

service_plugins = router,lbaas,lbaasv2 INCORRECT
```

2. Remove or comment out (#) the entry for the F5 LBaaSv1 service provider driver in the Neutron LBaaS configuration file (/etc/neutron/neutron_lbaas.conf).

```
[service_providers]
service_provider = LOADBALANCERV2:F5Networks:neutron_lbaas.drivers.f5.driver_v2.
    \ F5LBaaSv2Driver:default
# Must be in form:
# service_provider = <service_type>:<name>:<driver>[:default]
# List of allowed service types includes LOADBALANCER
# Combination of <service type> and <name> must be unique; <driver> must also be unique
# This is multiline option
# service_provider = LOADBALANCER:name:lbas_plugin_driver_path:default
# service_provider = LOADBALANCER:F5:f5.oslbaasv1driver.drivers.plugin_driver.
    \ F5PluginDriver:default
# service_provider = LOADBALANCER:OVSfwaas:neutron_lbaas.services.loadbalancer.
    \ OVSfwaasDriver:default
# service_provider = LOADBALANCER:Haproxy:neutron_lbaas.services.loadbalancer.
    \ HaproxyOnHostPluginDriver:default
# service_provider = LOADBALANCER:radware:neutron_lbaas.services.loadbalancer.
    \ radware.plugin_driver.RadwareLoadBalancerDriver:default
# service_provider = LOADBALANCER:NetScaler:neutron_lbaas.services.loadbalancer.
    \ NetScalerPluginDriver:default
# service_provider = LOADBALANCER:Embrane:neutron_lbaas.services.loadbalancer.
    \ EmbraneLbaas:default
# service_provider = LOADBALANCER:A10Networks:neutron_lbaas.services.loadbalancer.
    \ A10NetworksThunderDriver:default
# service_provider = LOADBALANCER:VMWareEdge:neutron_lbaas.services.loadbalancer.
    \ VMwareLoadBalancerDriver:default
```

1.17.4 VxLAN traffic doesn’t reach BIG-IP device

1. Check the BIG-IP vtep port lockdown settings. The default setting for port lockdown behavior does not include VxLAN traffic. Set the vtep to ‘Allow All’ to ensure that the BIG-IP device allows VxLAN traffic from the OpenStack cloud.

2. Check the VxLAN port binding.

If you’re using the default Open vSwitch (ovs) core plugin, run the command `ovs-vsctl show` to view a list of configured bridges and associated ports. As shown in the example below, there should be a remote_ip address for a VxLAN tunnel that corresponds to the self IP identified in the F5 agent configuration file.
Listing 1.5: The ovs bridge has a remote_ip address that corresponds to the BIG-IP vtep self IP address.

```
# ON NEUTRON CONTROLLER
[user@host-19 ~(keystone_user)]$ sudo ovs-vsctl show
f08cd9da-cf33-4bc6-bdd2-960caed1136c
Bridge br-ex
  ... 
  Bridge br-tun
    fail_mode: secure
    Port "vxlan-c9001901"
      Interface "vxlan-c9001901"
        type: vxlan
          options: {df_default="true", in_key=flow, local_ip="201.0.20.1", out_key=flow, remote_ip="201.0.25.1"}
    Port br-tun
      Interface br-tun
        type: internal
    Port "vxlan-0a020264"
      Interface "vxlan-0a020264"
        type: vxlan
          options: {df_default="true", in_key=flow, local_ip="201.0.20.1", out_key=flow, remote_ip="10.2.2.100"}
  ... 
```

TMSH

```
admin@{localhost}(cfg-sync Standalone)(Active)(/Common)(tmos.net)# list self vtep
  net self vtep {
    address 10.2.2.100/16
    allow-service all
    traffic-group traffic-group-local-only
    vlan external
  }
```

1.18 How to add the F5 flavors to OpenStack Nova

**Important:** By default, only admin users can create Nova flavors. See OpenStack Horizon - Manage flavors for more information.

You can add custom F5 flavors to OpenStack Nova using the OpenStack CLI. Commands follow the format

```
$ openstack flavor create FLAVOR_NAME --id FLAVOR_ID --ram RAM_IN_MB --disk ROOT_DISK_IN_GB --vcpus NUMBER_OF_VCPUS
```

The values shown here reflect the requirements for BIG-IP Virtual Edition images v12.0 and later.

1.18.1 f5-1slot

```
$ openstack flavor create f5-1slot --id auto --ram 4096 --disk 20 --vcpus 2
```

1.18. How to add the F5 flavors to OpenStack Nova
1.18.2 f5-ltm

```bash
$ openstack flavor create f5-ltm --id auto --ram 4096 --disk 50 --vcpus 2
```

1.18.3 f5-all

```bash
$ openstack flavor create f5-all --id auto --ram 4096 --disk 140 --vcpus 4
```

Tip: If you’re using the BIG-IP Application Acceleration Manager (AAM) module, increase the disk size to 160.

1.19 Releases and Versioning

See also:

Please see the Partners page for information about our OpenStack distribution platform partnerships and certifications.

1.19.1 Documentation

This documentation release, version 1.0, corresponds to the following version(s) of each F5 OpenStack Integration component.

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>f5-openstack-agent</td>
<td>8.x, 9.x, 10.x</td>
</tr>
<tr>
<td>f5-openstack-heat</td>
<td>7.x, 8.x, 9.x, 10.x</td>
</tr>
<tr>
<td>f5-openstack-heat-plugins</td>
<td>7.x, 8.x, 9.x, 10.x</td>
</tr>
<tr>
<td>f5-openstack-lbaasv2-driver</td>
<td>8.x, 9.x, 10.x</td>
</tr>
<tr>
<td>f5-openstack-lbaasv1¹</td>
<td>7.x, 8.x, 9.x</td>
</tr>
</tbody>
</table>

¹ F5 ceased to repair defects and perform maintenance on the F5 OpenStack LBaaS version 1 integration as of the Openstack Ocata release (April 2017). See the Neutron LBaaSV1 section below for more information.
1.19.2 F5 OpenStack Heat Integration

Table 1.10: OpenStack & F5 BIG-IP compatibility

<table>
<thead>
<tr>
<th>F5 Heat templates</th>
<th>F5 Heat plugins</th>
<th>OpenStack version</th>
<th>BIG-IP version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.x</td>
<td>7.x</td>
<td>Kilo</td>
<td>11.5.2+ 11.6.x</td>
</tr>
<tr>
<td>8.x</td>
<td>8.x</td>
<td>Liberty</td>
<td>11.5.2+ 11.6.x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12.1.x 13.0.x</td>
</tr>
<tr>
<td>9.x</td>
<td>9.x</td>
<td>Mitaka</td>
<td>11.5.2+ 11.6.x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12.1.x 13.0.x</td>
</tr>
<tr>
<td>10.x</td>
<td>10.x</td>
<td>Newton</td>
<td>11.5.2+ 11.6.x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12.1.x 13.0.x</td>
</tr>
</tbody>
</table>

1.19.3 F5 OpenStack LBaaS Connector

Neutron LBaaSv2

Table 1.11: OpenStack LBaaSv2 & F5 BIG-IP compatibility

<table>
<thead>
<tr>
<th>F5 LBaaS Connector version(s)</th>
<th>OpenStack version</th>
<th>BIG-IP version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.x</td>
<td>Liberty</td>
<td>11.5.2+ 11.6.x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12.1.x 13.0.x</td>
</tr>
<tr>
<td>9.x</td>
<td>Mitaka</td>
<td>11.5.2+ 11.6.x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12.1.x 13.0.x</td>
</tr>
<tr>
<td>10.x</td>
<td>Newton</td>
<td>11.5.2+ 11.6.x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12.1.x 13.0.x</td>
</tr>
</tbody>
</table>

Table 1.12: Linux OS Compatibility

<table>
<thead>
<tr>
<th>F5 LBaaS Connector version(s)</th>
<th>RHEL version(s)</th>
<th>Ubuntu version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.x, 9.x, 10.x</td>
<td>6, 7</td>
<td>12, 14</td>
</tr>
</tbody>
</table>

Neutron LBaaSv1

Important: End of Technical Support for F5 OpenStack LBaaS version 1

The F5 OpenStack LBaaS version 1 integration reached End of Technical Support (EOTS) effective with the Openstack Ocata release in April 2017.
This announcement is in compliance with the OpenStack community deprecation of the OpenStack Neutron LBaaS version 1 plugin. F5 encourages customers to move to OpenStack LBaaS version 2.

For additional information, please refer to the F5 End of Life policy.

The table below is for informational purposes only.

<table>
<thead>
<tr>
<th>F5 LBaaSv1 Connector version(s)</th>
<th>OpenStack version</th>
<th>BIG-IP version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.x</td>
<td>Kilo</td>
<td>11.5.2+, 11.6.x, 12.0.x</td>
</tr>
<tr>
<td>8.x</td>
<td>Liberty</td>
<td>11.5.2+, 11.6.x, 12.0.x</td>
</tr>
<tr>
<td>9.x</td>
<td>Mitaka</td>
<td>11.5.2+, 11.6.x, 12.0.x</td>
</tr>
</tbody>
</table>

### 1.20 Solution Test Plan

#### 1.20.1 Introduction

This document outlines the process for validating an Openstack deployment utilizing F5 products to provide Neutron LBaaSv2 (load balancing) services. It includes use cases that encompass the set of standard F5 OpenStack solution deployments, for testing purposes.

### F5 OpenStack Integrated Solutions

F5 produces integration solutions that orchestrate BIG-IP Application Delivery Controllers (ADC) with OpenStack Networking (Neutron) services. F5 OpenStack LBaaSv2 integration provides under-the-cloud multi-tenant infrastructure L4-L7 services for Neutron tenants.

In addition to community OpenStack participation, F5 maintains partnerships with several OpenStack platform vendors. Each vendor defines a certification process, including test requirements, that expand on or focus tests available in community OpenStack. This document presumes use of a certified deployment to the extent vendor tests have been, or will be, run to prove the validity of the deployment.

Community OpenStack and platform vendor tests exercise the generic LBaaSv2 integration. F5 OpenStack tests exercise F5-specific capabilities across multiple network topologies. They are complementary to community and platform vendor tests.

All F5 OpenStack tests are available in the same open source repository as the product code. You may execute these tests via tempest and tox – consistent with the OpenStack community – to self-validate a deployment.

Use cases derive from real-world scenarios representing repeatable deployments of the most commonly-used F5 OpenStack features. Use case tests validate the combination of OpenStack, F5 BIG-IP ADC, and F5 OpenStack products.

#### 1.20.2 Prerequisites

**OpenStack**

- Operational OpenStack Newton cloud deployed in accordance with minimal documented requirements:
  - Deployment configuration will vary to match test architectures described within each use case;
  - 1 host machine for a Controller node;
– 1 host machine for a Compute node.

• BIG-IP Nova flavor.

**TMOS**

• Supported TMOS version.

• If using Virtual Edition:

  – LTM_1SLOT KVM qcow2 image built using the supported BIG-IP onboarding Heat template;

  – Instance deployed using the supported BIG-IP VE Standalone, 3-NIC Heat template.

• Operational BIG-IP device or device service cluster with Better or Best license (including LTM and SDN Services).

• Initial configuration orchestrated to match the deployment architecture, per the F5 LBaaSv2 documentation.

**F5 OpenStack LBaaSv2**

• F5 Agent for OpenStack Neutron, F5 driver package, and F5 Driver for OpenStack LBaaSv2 installed on all hosts you want to use to provision BIG-IP services.¹

• Agent configured according to the needs of the use case test architecture.

### 1.20.3 Test Plan

Community OpenStack tests (not required, but recommended) are available to exercise the following key components:

• OpenStack Neutron for network topology deployment;

• OpenStack Nova for test web application deployment;

• OpenStack Neutron for LBaaSv2 service deployment:
  
  – Instructions for executing Tempest tests;

  – Tests compatible with F5 OpenStack LBaaSv2 are in the community repository.

F5 OpenStack tests exercise the following key components:

• F5 OpenStack LBaaSV2 plugin driver ();

• F5 OpenStack Agent ().

Each use case requires execution of tests over one or more standard network deployments, detailed below.

**Network Architectures**

**NA1: Global Routed Mode**

Edge deployment architecture using only OpenStack networking provider networks, with F5 OpenStack agents deployed in Global Routed Mode.

¹ The F5 driver package is available for download at https://github.com/F5Networks/neutron-lbaas/releases/download/v10.0.0/f5.tgz.
NA2: L2 Adjacent Mode

Micro-segmentation architecture using tenant networks, with F5 agents deployed in L2 Adjacent Mode. Execute tests for VLAN and then VxLAN network types.

F5 OpenStack tests supplement the community tests and exercise F5 LBaaS-specific features.

Use Cases

UC1: Community LBaaSV2

This use case focuses on basic integration of BIG-IP LTM to provide services through the OpenStack LBaaSV2 API. LBaaSV2 features tested include load balancers, listeners, pools, members, and monitors. LTM features tested include virtual servers, client TLS decryption, http profiles, multiple pools, cookie persistence, and monitored pool members. OpenStack networking APIs provide pool member state and virtual server statistics.

<table>
<thead>
<tr>
<th>Category</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectures</td>
<td>1, 2</td>
</tr>
<tr>
<td>Tests</td>
<td>neutron-lbaas</td>
</tr>
<tr>
<td></td>
<td>f5-openstack-lbaasv2-driver</td>
</tr>
</tbody>
</table>

1.21 Nova Flavors for BIG-IP Virtual Edition images

BIG-IP Virtual Edition (VE) images come in different sizes. Use the table below to determine the correct “f5 flavor” for your image.
Important: OpenStack Newton does not have any default Nova flavors. The Nova flavors shown in the table apply to OpenStack Mitaka and earlier.

Table 1.15: F5 Nova flavors

<table>
<thead>
<tr>
<th>BIG-IP version</th>
<th>BIG-IP VE Image</th>
<th>F5 flavor</th>
<th>Nova Flavor</th>
<th>Flavor Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.5, 11.6</td>
<td>LTM_1SLOT</td>
<td>f5-1slot</td>
<td>m1.medium</td>
<td>4096M RAM, 20GB disk, 2vCPUs</td>
</tr>
<tr>
<td></td>
<td>LTM</td>
<td>f5-ltm</td>
<td>m1.medium</td>
<td>4096M RAM, 40GB disk, 2vCPUs</td>
</tr>
<tr>
<td></td>
<td>ALL</td>
<td>f5-all</td>
<td>m1.large</td>
<td>8192M RAM, 120GB disk, 4vCPUs</td>
</tr>
<tr>
<td>12.0, 12.1, 13.0</td>
<td>LTM_1SLOT</td>
<td>f5-1slot</td>
<td>m1.medium</td>
<td>4096M RAM, 20GB disk, 2vCPUs</td>
</tr>
<tr>
<td></td>
<td>LTM</td>
<td>f5-ltm</td>
<td>m1.large</td>
<td>4096M RAM, 50GB disk, 2vCPUs</td>
</tr>
</tbody>
</table>
|                | ALL             | f5-all    | m1.xlarge   | 8192M RAM, 140-160GB disk, 4vCPUs

See also:
- How to add the F5 Nova flavors to OpenStack
- OpenStack Admin Guide: Manage flavors

1.22 OpenStack Platform Partnerships

F5 has tested and certified its integration with OpenStack distribution platform vendors such as Red Hat, Mirantis, and HPE, as well as SDN partners Cisco, PLUMgrid, and Nuage. That means you can count on fully supported, validated OpenStack implementations.

1 160GB is the largest possible disk size for a BIG-IP VE image as of v13.0.0. See the BIG-IP VE Linux KVM Setup Guide for more information.
Each of F5’s OpenStack distribution partners provides comprehensive documentation for installing and configuring OpenStack.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Platform version(s)</th>
<th>F5 Connector version(s)</th>
<th>OpenStack version</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPE Helion OpenStack</td>
<td>v4.x</td>
<td>v9.x</td>
<td>Mitaka</td>
</tr>
<tr>
<td></td>
<td>v3.x</td>
<td>v8.x</td>
<td>Liberty</td>
</tr>
<tr>
<td></td>
<td>v2.x</td>
<td>v7.x</td>
<td>Kilo</td>
</tr>
<tr>
<td>Mirantis OpenStack</td>
<td>v9.0</td>
<td>v9.x</td>
<td>Mitaka</td>
</tr>
<tr>
<td></td>
<td>v7.0</td>
<td>v7.x</td>
<td>Kilo</td>
</tr>
<tr>
<td>RedHat OpenStack Platform</td>
<td>v9</td>
<td>v9.x</td>
<td>Mitaka</td>
</tr>
<tr>
<td></td>
<td>v8</td>
<td>v8.x</td>
<td>Liberty</td>
</tr>
<tr>
<td></td>
<td>v7</td>
<td>v7.x</td>
<td>Kilo</td>
</tr>
</tbody>
</table>

1.23 Acknowledgements

1.24 F5 Networks Contributor License Agreement

Before you start contributing to any project sponsored by F5 Networks, Inc. (F5) on GitHub, you will need to sign a Contributor License Agreement (CLA). If you are signing as an individual, we recommend that you talk to your employer (if applicable) before signing the CLA since some employment agreements may have restrictions on your contributions to other projects. Otherwise by submitting a CLA you represent that you are legally entitled to grant the licenses recited therein. If your employer has rights to intellectual property that you create, such as your contributions, you represent that you have received permission to make contributions on behalf of that employer, that your employer has waived such rights for your contributions, or that your employer has executed a separate CLA with F5.

If you are signing on behalf of a company, you represent that you are legally entitled to grant the license recited therein. You represent further that each employee of the entity that submits contributions has authorization to submit such contributions on behalf of the entity pursuant to the CLA.

Click the link below to download the PDF:

F5 Contributor License Agreement (CLA)

1.25 F5 Development Branching Strategy

Important: Development for all F5® OpenStack projects in GitHub follows this branching strategy.

1.25.1 Key Points

- Development takes place on feature branches, which tend to follow the naming convention feature.<featurename>. These branches are either created from master or from the branch that corresponds to the earliest OpenStack version to which the bugfix/change applies (e.g., liberty).
- Feature branches are periodically merged into their ‘parent’ branches, most often in preparation for a release.
- Branches which correspond to specific OpenStack versions (e.g., kilo, liberty, mitaka) are periodically merged up the chain to ensure commits which apply to multiple versions make it into each version’s branch.
• **master** is the branch on which all development for new OpenStack versions occurs.

**Development Example**

1. Create `feature.newton` from `master`.
2. All development to support the Newton release takes place on the `feature.newton` branch.
3. Merge the `feature.newton` branch into `master` when development for the release is complete.
4. Create the `newton` branch from `master`.
   • All Newton version releases (e.g., 10.0.1, 10.0.2, etc.) come from the `newton` branch.
   • All bugfixes for Newton and later versions happen on feature branches created from, and then merged back into, the `newton` branch.
5. `master` then moves ‘ahead’ of `newton` in the branching structure; it is now the basis for all development for the Ocata release.

**Bugfix Example**

1. Create a bugfix feature branch from `liberty`—for example, `bugfix#.liberty`—because that’s the earliest OpenStack version in which the bug occurred.
2. Merge the `bugfix#.liberty` branch into `liberty` when development is complete.
3. Merge `liberty` up to `mitaka`.
4. Merge `mitaka` up to `newton`.
5. Merge `newton` up to `master`.
6. The bugfix is then automatically included in `ocata`, which the development team will create from `master`.
Fig. 1.7: Branching Strategy
CHAPTER 2

Enhancements and issues

If you would like to request new documentation or notify us of a docs bug, please file an issue in GitHub.
Acknowledgements

Please see the Acknowledgements page for a comprehensive list of the projects used to build and publish this documentation.