

Integration with Non-HP Hardware

Contents

- Integration with Non-HP Hardware.....3**
 - A10 Networks Integration Overview..... 3
 - A10 Pre-Installation Checklist.....3
 - Install the LBaaS v2 Driver..... 7
 - Test A10 Integration..... 7
 - F5 BIG-IP Integration Overview.....7
 - F5 Pre-Installation Checklist..... 8
 - Install the F5 Driver..... 11
 - Verify Load Balancing..... 15

HPE Helion OpenStack Integration with Non-HPE Hardware

With HPE Helion OpenStack you have the freedom to choose the hardware you want to run cloud compute, storage, and networking. To help you deploy a multi-vendor datacenter, HPE verifies and certifies hardware that is both HPE and non-HPE hardware.

At HPE, the Helion Ready Program for Hardware Vendors (IHV) team has been working to ensure that the following third party hardware integrates with HPE Helion OpenStack:

A10 Networks

provide application delivery networking, Distributed Denial of Service (DDoS) protection, DDoS attack mitigation, load balancing, and next generation firewall solutions.

F5 Networks

provide load balancing and a range of integrated products to improve the speed and security of the delivery of web-based applications.

SolidFire clusters

provide an all-flash storage system that handles data placement, data protection, and performance.

TinTri Storage Systems

provide flash-based storage products specifically designed for virtualized and cloud environments.

The following topics will help you prepare, install, configure, and verify these hardware products in your HPE Helion OpenStack 2.0 cloud:

- [Integrating A10 Networks LBaaS v2 with Helion Openstack](#)
- [Integrating F5 BIG-IP with Helion Openstack](#)

A10 Networks Integration Overview

A10 Networks' OpenStack integration enables a dynamic provisioning of application networking (L4-L7) services on the OpenStack platform, providing the automation and agility expected in a cloud infrastructure.

Adding the A10 LBaaS v2 driver to your Helion OpenStack deployment provides the LBaaS (load-balancer-as-a-service) extension to Neutron and adds functionality for interfacing with other appliances. The A10 Networks' LBaaS plug-in module integrates with the LBaaS extension to provide the functionality for interfacing with the A10 Thunder appliances, including physical, virtual and hybrid appliances. RESTful APIs are used to manage LBaaS in a static configuration. Integrating A10 Thunder 930 into your deployment can increase your efficiency standard to deliver 200,000 Layer 4 connections per second (CPS) and 5 Giga bytes per second (Gbps) of application throughput in a 1U appliance.

To introduce load-balancing features into the Neutron core you must perform the following tasks:

1. [Complete the A10 Networks Pre-Installation Checklist](#)
2. [Install and Configure the LBaaS v2 Driver](#)
3. [Test A10 Networks Integration](#)

A10 Pre-Installation Checklist

Before installing the A10 Networks LBaaS v2 driver in your Helion OpenStack deployment, make sure you fulfill all of the following requirements:

A10 Pre-Installation Checklist

<input type="checkbox"/>	Item
	Use Supported Hardware Configuration
	Configure Your Environment Proxy Settings
	Verify You Have a Working Helion OpenStack Cluster
	Configure the Helion OpenStack Cluster with Glance Image
	Configure the Helion OpenStack Cluster with an External Network
	Configure the Helion OpenStack Cluster with a Tenant Network
	Verify Floating IP is Working
	Manually Install Python Packages Into Each Controller Node

Use a Supported Hardware Configuration

For testing purposes, the following hardware configuration was used to test the A10 Networks LBaaS driver integration with Helion OpenStack 2.0:

Appliance	Requirement
Make	A10 Networks
Model	Thunder ADC 930
Operating System	64 bit Advanced Core OS Version 2.7.1-P5 (build 68)

For more information on the A10 Networks Thunder ADC hardware specifications, see the [Thunder ADC Datasheet](#).

Configure Your Environment Proxy Settings

To install the required Python packages, you must know your proxy settings for the HTTP protocol and add them to your configuration file.

To find your current proxy settings:

1. At the command prompt, run:

```
ENV|grep -i proxy
```

To set your environment proxy settings:

1. To open the configuration file in the vi text editor, at the command prompt, run:

```
#vi etc/sampledirectory/Virtualenv.config
```

2. To modify data in the configuration file and go into insert mode, press **i**.
3. To set the HTTP proxy, enter your settings in the following format:

```
export http_proxy=http://<proxy-server-ip>:<port>
```

For example:

```
export http_proxy=http://10.1.24.18:8080
```

Verify You Have a Working Helion OpenStack Cluster

Perform the following action to see a list of active clusters and their status.

1. To view a list of clusters for the given vCenter, run:

```
#eon cluster-list --vcenter-id
```

Sample Output

```
# eon cluster-list --vcenter-id BC9DED4E-1639-481D-B190-2B54A2BF5674
```

MOID	Name	Datacenter	Import Status
domain-22	cluster2	DC1	imported


Configure the Helion OpenStack Cluster with Glance Image

The Glance project provides the following features:

- image registration
- discovery service
- image delivery service

These services are used by Nova to deliver images from object stores, such as OpenStack's Swift service to Nova's compute nodes. Glance may be deployed in a number of ways.

For instructions on how to deploy Glance, refer to the Glance product documentation: [Glance](#)

 **Warning:** All deployments require the existence of the other core OpenStack services deployed through Juju charms, specifically: mysql, keystone and nova-cloud-controller. The referenced documentation assumes these services have already been deployed.

Configure the Helion OpenStack Cluster with an External Network


You must have an external network set up to allow your Compute instances to reach the internet. There are multiple methods you can use to create this external network. The HPE Helion OpenStack Documentation provides information on two ways you can create an external network. Click on the following links for more information.

To configure an external network, choose one of the following options:

- [Using the Ansible Playbook](#) The HPE Helion OpenStack installer provides an Ansible playbook that will create this network for use across your projects. This playbook will query the Networking service for an existing external network, and then create a new one if you do not already have one. The resulting external network will have the name ext-net with a subnet matching the CIDR you specify in the command.
- [Using the NeutronClient CLI](#) Use the command line tool from your lifecycle manager to create an external network.

Configure the Helion OpenStack Cluster with a Tenant Network

In OpenStack user interfaces and documentation, a group of users is referred to as a project or tenant. These terms are interchangeable. In an OpenStack environment, you can create tenant networks using the Networking service (code named Neutron). Tenant networks are used to isolate access to Compute resources.

 **Note:** Refer to the Helion OpenStack 2.0 Documentation for general information on [Creating a Private Network](#).

1. To create a tenant network, run:

```
# neutron net-create tenant-network
```

```
# neutron subnet-create -name tenant-subnet -gateway <ip-
address> -- allocation pool start <ip-address>,end=<ip-address> tenant-
network <ip-address/prefix>
```

2. To provision the baremetal node, run:

```
# nova boot --nic net-id=<tenant network id> --image <glance image id> --
flavor baremetal --key-name <keypair-name>
```

Using the following values:

tenant network id

is the UUID of the tenant network

image

is the name of ID of image (see **nova image-list**

flavor

is the name or ID of flavor

keypair-name

is the name of the keypair to use

```
#nova image-list
```

ID	Name	Status	Server
ca797ded-5fa0-4ef2-8635	hoscg-2.0	Active	
2394d02e-2f58-4b85-b9e4	hoscg-2.0	Active	

3. To view details about the network, run:

```
#neutron net-show tenant-network
```

Sample Output

Field	Value
admin_state_up	True
ID	2c4f260c-507a-47a7-a0f4-7faebfe312b0
mtu	0
name	tenant-network
provider:network_type	vlan
provider:physical_network	physnet1
provider:segmentation_id	102
router:external	False
shared	True
subnets	6a952e6b-903f-41db-b31d-62dc52c65dd0
tenant_id	c288a2816237465c8ea6efc16dc05b88

Verify Floating IP is Working

When a single OpenStack cluster boots up, fixed IPs are allocated dynamically by the nova-network component. A fixed IP works well when you need connectivity only between instances inside your cloud deployment.

To provide IP addresses that are publicly routable and to allow users to explicitly allocate an IP address, the cloud administrator must configure a pool of floating IPs. When an instance starts up, this pool allows users to allocate floating IP addresses to their instances. The result is that the floating IP address is now visible to different ISPs or external networks outside of your cloud deployment. If an instance dies, the user can reuse the floating IP by attaching it to another instance. Only one floating IP address can be allocated to an instance at any given time.

Perform the following actions to verify that floating IP Pools are configured:

1. To list all pools that provide floating IP addresses, run:

```
# nova floating-ip-pool-list
```

Sample Output

Name	Public Test
 Attention: If this list is empty, the cloud administrator must configure a pool of floating IP addresses. OpenStack provides documentation on how to Configure A Pool of Floating IPs .	

2. To list all floating IP addresses that are allocated to the current project, run:

```
# nova floating-ip-list
```

Sample Output

IP	Instance ID	Fixed IP	Pool
172.24.4.255	4a60ff6a-7a3c-49d7-9515-86ae501044c6	10.0.0.2	public
172.24.4.226	None	None	Public

For each floating IP address that is allocated to the current project, the command outputs the floating IP address, the ID for the instance to which the floating IP address is assigned, the associated fixed IP address, and the pool from which the floating IP address was allocated.

Manually Install Python Packages Into Each Controller Node

The term “package” in this context is being used as a synonym for a distribution, or a bundle of software to be installed. Do not confuse it with the kind of package that you import in your Python source code (i.e. a container of modules).

Refer to the Python Website to read more about [Installing Packages](#).

Install the LBaaS v2 Driver

This content is under construction.

Test A10 Integration

This content is under construction.

F5 BIG-IP Integration Overview

Adding F5 Network products to your deployment can provide critical application delivery services consistently across the data center, private cloud, and public cloud. This allows hybrid users to experience the availability and performance they expect.

F5 Networks BIG-IP product family includes purpose-built hardware, software, and virtualized solutions. Depending on the appliance you use, one or more BIG-IP modules can be added to get the functionality your deployment requires.

Creating an application centered strategy using F5 BIG-IP is supported in all of the following scenarios:

- architecting a private cloud
- migrating and re-architecting existing applications to a public cloud IaaS (load balancing as a service) provider
- moving to Software as a Service (SaaS)
- deploying applications in a hybrid environment
- ensuring business continuity



Note: Keep in mind that flexible licensing options are available, based on where you want to deploy.

To integrate F5 BIG-IP features into your OpenStack deployment:

1. [Complete the F5 Preinstallation Checklist](#)
2. [Install the F5 Plug-In and Drivers](#)
3. [Verify Load Balancing](#)

F5 Pre-Installation Checklist

Before installing the F5 BIG-IP driver in your Helion OpenStack deployment, make sure you fulfill all of the following requirements:

<input type="checkbox"/>	Item
<input type="checkbox"/>	Use the Supported Hardware Configuration
<input type="checkbox"/>	Verify the F5 Device
<input type="checkbox"/>	Create the supported network configuration



Attention: F5 BIG-IP currently supports only LBaaS Version 1. This is the version that was used to validate F5 BIG-IP integration with Helion OpenStack 2.0. The F5 LBaaS driver that was tested with Helion OpenStack 2.0 is planned to be officially supported in the next release of OpenStack.

Use the Supported Hardware Configuration

HPE Helion OpenStack supports the F5 hardware configurations listed in the [HPE Helion Ready Solution Catalog](#).

For testing purposes, the following hardware configuration was used to test F5 BIG-IP deployment with Helion OpenStack 2.0:

Appliance	Requirement
Make	F5 BIG-IP Load Balancer Device
Model	F5 BIG-IP 3600
Operating System	12.0.0 (Build 0.0.606) OpenStack Kilo - F5 BIG-IP SM
	Attention: F5 BIG-IP is officially not supported in OpenStack Kilo.

Verify the F5 Device

Integrating F5 BIG-IP with Helion OpenStack 2.0 requires that your F5 device is configured and available.

For more information on the initial configuration of your F5 device, refer to the documentation you received from F5 Networks when the device was purchased.

The following links provide more information from the F5 Support Web site:

- [BIG-IP initial configuration](#)
- [Configure High Availability](#)

Create Supported F5 Network Configuration

To integrate F5 Networks BIG-IP into your HPE Helion OpenStack deployment, you must create the following networks and verify they are working before you can install F5 drivers for BIG IP.

To configure your network to prepare for BIG-IP:

1. [Create a management network](#)
2. [Create a high availability network](#)
3. [Create a Virtual Tunnel Endpoint \(VTEP\) network](#)
4. [Verify your network connectivity](#)

Create a Management Network

Integrating F5 BIG-IP with Helion OpenStack 2.0 requires that you create a Helion management network. This network is a VLAN that is not part of Helion cluster. The same VLAN that you create in these steps must be used to configure the F5 device for management network. The network you create in the following steps must be the same network VLAN that the F5 device management network is on.

To create a management network:

1. To create the network, replace the sample settings with the settings from your deployment in the following command and run:

```
#neutron net-create f5-mgmt --provider:physical_network physnet1 --
provider:network_type vlan --provider:segmentation_id 102
```

2. To create a subnet, replace the sample settings with the settings from your deployment in the following command and run:

```
#neutron subnet-create f5-mgmt 10.1.36.0/24 --name f5-mgmt-subnet --
gateway 10.1.36.1 --allocation-pool start=10.1.36.15,end=10.1.36.200
```

3. To create a router for the network, run:

```
#neutron router-create f5-mgmt-router
```

4. To create an interface between the router and the subnet, run:

```
#neutron router-interface-add f5-mgmt-router f5-mgmt-subnet
```

5. To set the router gateway to the external network, run:

```
#neutron router-gateway-set f5-mgmt-router ext-net
```

Create a High Availability Network

Integrating F5 BIG-IP with Helion OpenStack requires that you create a high availability (HA) network. You must create this network using a VLAN that is not in use anywhere else.

To create a high availability network:

1. To create the network, use the settings for your deployment in the following command and run:

```
#neutron net-create F5-HA --provider:physical_network physnet1 --
provider:network_type vlan --provider:segmentation_id 103
```

2. To create a subnet, use the settings for your deployment in the following command and run:

```
#neutron subnet-create F5-HA 10.1.37.0/24 --name F5-HA-subnet --gateway 10.1.37.1
```

3. To create a router for the network, run:

```
#neutron router-create F5-HA-router
```

4. To create an interface between the router and the subnet, run:

```
#neutron router-interface-add F5-HA-router F5-HA-subnet
```

5. To set the router gateway to the external network, run:

```
#neutron router-gateway-set F5-HA-router ext-net
```

Create a Virtual Tunnel Endpoint (VTEP) Network

Integrating F5 BIG-IP with Helion OpenStack 2.0 requires that you create a virtual tunnel endpoint (VTEP) network. You must create this network using a free IP range from your GUEST_VXLAN VLAN.

To create a VTEP network:

1. To create the network, use the settings for your deployment in the following command and run:

```
#neutron net-create f5-vtep --provider:physical_network physnet1 --provider:network_type vlan --provider:segmentation_id 106
```

2. To create a subnet, use the settings for your deployment in the following command and run:

```
#neutron subnet-create f5-vtep 10.1.40.0/24 --name f5-vtep-subnet --gateway 10.1.40.1 --allocation-pool start=10.1.40.200,end=10.1.40.25
```

3. To create a router for the network, run:

```
#neutron router-create F5-vtep-router
```

4. To create an interface between the router and the subnet, run:

```
#neutron router-interface-add F5-vtep-router F5-vtep-subnet
```

5. To set the router gateway to the external network, run:

```
#neutron router-gateway-set F5-vtep-router ext-net
```

Verify Your Network Connectivity

After creating the management, high availability (HA), and virtual tunnel endpoint (VTEP) networks, you should verify connectivity before installing the F5 BIG-IP driver/plugin. When the PING test passes on both Controller and Compute nodes, you can install the F5 LBaaS Driver/Plug-in.

To verify connectivity:

1. Launch a virtual machine (VM) on your mgmt (management) network.
2. To ping an IP address, from the Controller node, run:

```
#ping <ipaddress_of_Compute_node>
```

3. Repeat these steps for Compute node.



Attention: Repeat these steps for the HA (high availability) and VTEP (virtual tunnel endpoint) networks before installing the LBaaS (load balancing as a service) driver.

Install the F5 Driver

At the time of the Helion OpenStack 2.0 release, the F5 neutron-LBaaS driver is only available in the following file formats:

- .deb
- .rpm

These formats cannot be directly installed by Helion OpenStack. Therefore it is necessary to manually install the driver using the following steps:

1. *Download the archives from the F5 Networks site*
2. *Extract the files relevant to BIG IP*
3. *Copy the files to the neutron virtual environment*
4. *Install the F5 agent*
5. *Run the F5 agent*



Attention: Steps 1-3 should be repeated on all neutron control plane nodes.

Download the archives from the F5 Networks site

Before downloading files you must install the screen utility on the lifecycle manager (deployer node). Then make sure you have stopped all existing neutron LBaaS agents on the computer node Bbefore you can place the downloaded files into the neutron environment.

To download and extract the F5 files:

1. Log into the lifecycle manager.
2. To install the screen utility, run:

```
#sudo apt-get install screen
```

3. To create an ansible playbook to stop all LBaaS agents, run:

```
#cd ~scratch/ansible/next/hos/ansible
```

4. Create a file with the following name:

```
neutron-lbaas-stop.yml
```

5. Add the following lines to the neutron-lbaas-stop.yml file, replacing names with those used in your environment:

```
- hosts: NEU-LBAV2
  sudo: yes
  roles:
    - neutron-common
    - neutron-lbaasv2-agent

  tasks:
    - include: roles/neutron-lbaasv2-agent/tasks/stop.yml

-hosts: NEU-LBAV1
sudo: yes
roles:
  - neutron-common
  - neutron-lbaasv2-agent

tasks:
```


```
- include: roles/neutron-lbaasv2-agent/tasks/stop.yml
```

6. To stop existing LBaaS agents using the ansible playbook, run:

```
#ansible-playbook -i hosts/verb_hosts neutron-lbaas-stop.yml
```

7. To verify that lbaas agents have stopped, run:

```
#neutron agent-list
```

 **Attention:** Keep in mind that it takes several minutes for an updated list to be shown by this command.

8. To download the F5 driver Debian files (Version 1.0.10), run


```
#cd ~
#wget --no-check-certificate -O F5.tgz https://
devcentral.f5.com/d/openstack-neutron-lbaas-driver-and-agent?download=true
```

9. To verify the contents of the downloaded archive, run:

```
# md5sum F5.tgz
```

Sample output

```
e0a441b07874e7958de0bc5d733b9cca F5.tgz
```

 **Attention:** You should repeat these steps for the high availability and virtual tunnel endpoint (VTEP) networks before installing the LBaaS driver.

Extract the files relevant to BIG IP


The extraction procedure will extract several files, however you are only interested in the files with the ".deb" extension. Then you must test running the F5 agent on multiple nodes.

To extract the files from the F5 .deb archives and test the agent:

1. In the same directory where you downloaded the archive, run:

```
#tar xvfz F5.tgz
```

2. Copy the .deb archives onto appropriate nodes.

 **Attention:** Use the following guidelines:


- The API driver goes onto all neutron API control plane nodes
- The agent and common packages go on one control plane node

3. To test running the agent on multiple nodes, run the following commands:

```
#cd f5
#scp f5-oslbaasv1-driver_1.0.10-1_all.deb stack@padawan-ccp-c1-m1-
mgmt:
#scp f5-oslbaasv1-driver_1.0.10-1_all.deb stack@padawan-ccp-c1-m2-
mgmt:
#scp f5-oslbaasv1-driver_1.0.10-1_all.deb stack@padawan-ccp-c1-m3-
mgmt:
#scp f5-bigip-common_1.0.10-1_all.deb stack@padawan-ccp-c1-m1-mgmt:
#scp f5-oslbaasv1-agent_1.0.10-1_all.deb stack@padawan-ccp-c1-m1-
mgmt:
```

Copy the files to the neutron virtual environment

Since the F5 .deb files are delivered via distribution packages, and Helion OpenStack expects the files to be under site packages, you must move the files to the correct location. Then you can edit the settings to identify the default load balancing service provider. Make sure you restart the neutron server when you are finished.

 **Attention:** Repeat all of these steps on all neutron controller plane nodes.

To copy the files to the neutron virtual environment:

1. To install the F5 API driver .deb files on all controller nodes, run the following commands:

```
#ssh stack@padawan-ccp-cl-m1-mgmt
#dpkg -x f5-oslbaasv1-driver_1.0.10-1_all.deb f5-driver
#cd f5-driver/usr/lib/python2.7/dist-packages
#sudo cp -rP * /opt/stack/service/neutron/venv/lib/python2.7/
site-packages/
#sudo touch /opt/stack/service/neutron/venv/lib/python2.7/
site-packages/f5/oslbaasv1driver/__init__.py
```

You can choose to either set F5 as the default load balancing service provider, or you set another provider as the default. First determine which LOADBALANCER service provider you want to set as the default. If you have multiple service providers, only one of them may have the ":default" tag.

To set the default load balancing service provider, choose one of the following options:

- Set F5 as the default
- Set another provider as the default

To set F5 as the default load balancing service provider:

1. In a text editor, open the following file:


```
/opt/stack/service/neutron/etc/neutron.conf
```

2. Find the following service provider entry:

```
LOADBALANCER:Haproxy
```

3. If you do not want to provide access to the HAProxy driver, delete the line containing the LOADBALANCER:Haproxy entry.
4. If you do want to provide access to the HAProxy driver, delete the following tag from the end of the line :default.
5. In the neutron.conf file you have open, add the following file:

```
service_provider=LOADBALANCER:F5:f5.oslbaasv1driver.drivers.plugin_driver.F5PluginDri
```

 **Caution:** Only remove the ":default" tag for the HAPROXY line. The VPN line should have a ":default" tag at the end. If this is not configured correctly you will have issues in getting the neutron-server service up and running.

Sample of a working neutron.conf file:

```
[service_providers]
#service_provider =
LOADBALANCERV2:Haproxy:neutron_lbaas.drivers.haproxy.plugin_driver.HaproxyOnHostPlugin
# To enable LBaaSv1, comment the above service_provider line and
uncomment the below
service_provider =
LOADBALANCER:Haproxy:neutron.services.loadbalancer.drivers.haproxy.plugin_driver.Haproxy
```

```

service_provider =
VPN:openswan:neutron_vpnaas.services.vpn.service_drivers.ipsec.IPsecVPNDriver:default
service_provider =
LOADBALANCER:F5:f5.oslbaasv1driver.drivers.plugin_driver.F5PluginDriver:default

```

To set F5 as the default load balancing service provider:

1. In a text editor, open the following file:

```
/opt/stack/service/neutron/etc/neutron.conf
```

2. Add the following file:

```
service_provider=LOADBALANCER:F5:f5.oslbaasv1driver.drivers.plugin_driver.F5PluginDriver:default
```

Before installing the F5 agent on the controller node, you must restart the server and check the neutron server log.

1. To restart the neutron server, run:

```
#sudo service neutron-server restart
```



Attention: Keep in mind that it takes several minutes for this process to completed.

2. Return to controller node one and open the following file: `neutron-server.log`.
3. Verify the following conditions:

- there are no stack traces
- the driver has loaded



Tip: If the driver does not load, verify the existence of the following file in all the f5 subdirectories in the site packages:

```
__init__.py
```

4. To install the F5 agent code on one controller node, run the following commands:

```

#ssh stack@helion-ccp-cl-m1-mgmt
#sudo apt-get install python-suds # Needed by F5, not in
hLinux.
#dpkg -x f5-bigip-common_1.0.10-1_all.deb f5-common
#dpkg -x f5-oslbaasv1-agent_1.0.10-1_all.deb f5-agent
#sudo cp -rP f5-common/usr/lib/python2.7/dist-packages/f5/* /
opt/stack/service/neutron/venv/lib/python2.7/site-packages/f5/
#sudo cp -rP f5-agent/usr/lib/python2.7/dist-packages/f5/* /
opt/stack/service/neutron/venv/lib/python2.7/site-packages/f5/
#sudo cp -rP f5-agent/etc/* /opt/stack/service/neutron/etc
#sudo cp -rP f5-agent/usr/bin/* /opt/stack/service/neutron/
venv/bin

```

5. In a text editor, open the following file:

```
/opt/stack/service/neutron/etc/neutron/f5-oslbaasv1-agent.ini
```

6. Add the following lines to this file:

Code	Description
<code>f5_ha_type = standalone</code>	add this line if you are only using one F5 server
<code>f5_vtep_selfip_name = 'VTEP'</code>	'VTEP' is the default. Make sure this matches the selfIP name on F5 server

Code	Description
icontrol_hostname = 10.1.36.5	this is the management IP address of the F5 server

To run the F5 agent:

1. Log in to the lifecycle manager (deployer node).
2. Run the agent under screen so that it will continue after disconnection.
3. run the following commands:

```
#ssh stack@helion-cp1-cl-m1-mgmt
#cd /opt/stack/service/neutron/venv/bin
#sudo -u neutron bash
#./python2.7 f5-oslbaasv1-agent --config-file /opt/stack/
service/neutron/etc/neutron.conf --config-file /opt/stack/service/neutron/
etc/neutron/f5-oslbaasv1-agent.ini --log-file /var/log/neutron/f5.log
```

To confirm the F5 agent status:

1. From deployer node or any controller node, run:

```
neutron agent-list
```

2. Verify the following settings:

Item	Status
f5-oslbaasv1-agent(Loadbalance agent binary)	: -) (alive status)
admin_state_up	True

3. Login to the Horizon Dashboard with administrator credentials.
4. Click on **Admin**, then, **System**, then **System Information**, then **Network Agents**.
5. Verify the following settings:

Item	Status
f5-oslbaasv1-agent(Loadbalance agent name)	Enabled
State	Up

Verify Load Balancing

To verify that load balancing with the F5 driver is working involves the following steps:

1. Create two tenant subnets, one externally-accessible public network, and one internal private network.
2. Connect the networks with a neutron router.
3. Create the test instances and loadbalancer VIP on the private subnet.
4. Configure the VIP to be accessed from the public network using a floating IP.

To complete these steps, use the following checklist:

<input type="checkbox"/>	Task	Additional Details
<input type="checkbox"/>	Create appropriate private and public networks and subnets	<ol style="list-style-type: none"> 1. Create the networks 2. Create a router connecting these networks
<input type="checkbox"/>	Boot at least two instances on a private subnet running a simple web server	<ol style="list-style-type: none"> 1. Define appropriate security groups allowing ICMP, SSH, and HTTP traffic

□	Task	Additional Details
		<ol style="list-style-type: none"> 2. Configure each web server so that its output easily identifies which instance it is running on 3. Use the 'curl' command to verify that the web server
	Create a Pool	
	Create a VIP on the private subnet	
	Create Members for each web server instance	
	Create a floating IP on the public network and associate it with the VIP	
	Log into the F5 device and verify that the appropriate objects have been created on the device	
	Boot an instance on the public network	<ol style="list-style-type: none"> 1. Log into this instance 2. Use 'curl' to query the floating IP associated with the VIP 3. Repeated queries of the VIP FIP should show load balancing between the members