NetVisor OS Openstack ML2 Plugin Deployment Guide

Arista Networks

www.arista.com

Arista NetVisor Version 2022.7.0.2
DOC-05950-01

Headquarters	Support	Sales
5453 Great America Parkway Santa Clara, CA 95054, USA +1-408-547-5500	+1-408 547-5502 +1-866 476-0000	+1-408 547-5501 +1-866 497-0000
www.arista.com	support@arista.com	sales@arista.com

Table of Contents

Preface 4
Audience 5
Conventions 6
Document Feedback 7
Obtaining Documentation and Submitting a Service Request 9
Glossary 10
Understanding the ML2 Plugin 11
System Requirements 15
Downloading and Installing the Pluribus ML2 Plugin 16
Configuring a Network using Pluribus Neutron ML2 Plugin 20
Using the Horizon Dashboard to Configure a Network 30
Guidelines, Errors, and Exceptions 33
Related NetVisor OS CLI commands 35
Appendix A - Generic Installation 38

Preface

- Audience
- Conventions
- Documentation Feedback
- Obtaining Documentation and Submitting a Service Request

Audience

This publication is for experienced network administrators responsible for configuring and maintaining network switches with some expertise in the following areas:

- Network administration
- OpenStack network administration
- Storage administration
- Server administration
- Application delivery administration
- Network security administration

Conventions

This document uses the following conventions:

Convention	Indication
Bold font	Keywords, user interface elements, and user-entered text appear in **bold** font.
Italic font	Document titles, new or emphasized terms, and variables that you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x|y|z}	Required elements are grouped in curly braces and are separated by vertical bars.
[x|y|z]	Optional parameters are grouped in brackets and separated by vertical bars.
String	A non-quoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.
courier font	Command Line Interface (CLI) commands and samples appear in courier font.
< >	Non-printing characters such as passwords are indicated by angle brackets.
[ ]	Default responses to system prompts are in square brackets.
:	Indicates that you enter the following text at the command prompt.

Note:	Indicates information of special interest.
Caution!	Indicates a situation that could cause equipment failure or loss of data.
TIP!	Indicates information that can help you solve a problem.
Warning:	Indicates information that could impact you or your network.
Time Saver:	Indicates information that can help you save time.

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to: pln-doc-feedback@arista.com

We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, please visit www.pluribusnetworks.com.

Glossary

Glossary of NetVisor UNUM and NetVisor OS Terms

To review the Glossary, refer to the online document.

Understanding the NetVisor OpenStack Neutron ML2 Plugin

OpenStack is a cloud operating system that can be deployed in both public and private clouds and it controls large pools of compute, storage, and networking resources throughout a datacenter. These resources are managed and provisioned through a dashboard, CLI, or through APIs with common authentication mechanisms.

NetVisor uses the OpenStack plugin mechanism to create a Layer 2 network (VLAN), the NetVisor Modular Layer 2 (ML2) plugin, which provides powerful and simplified overlay automation services, allows customers to create a network by configuring the VLANs and switch ports on NetVisor Unified Cloud Fabric over VXLAN tunnel or overlays. For details on Unified Cloud Fabric, see the *Configuring and Administering the Unified Cloud Fabric* chapter in the *Configuration Guide for NetVisor OS*.

The integration of NetVisor ML2 Driver Plugin with the OpenStack Architecture is explained below with the help of the topology diagrams. The networking (Neutron) component of OpenStack architecture is integrated through redundant programmable REST APIs to the NetVisor Unified Cloud Fabric (refer **Figure 1**).

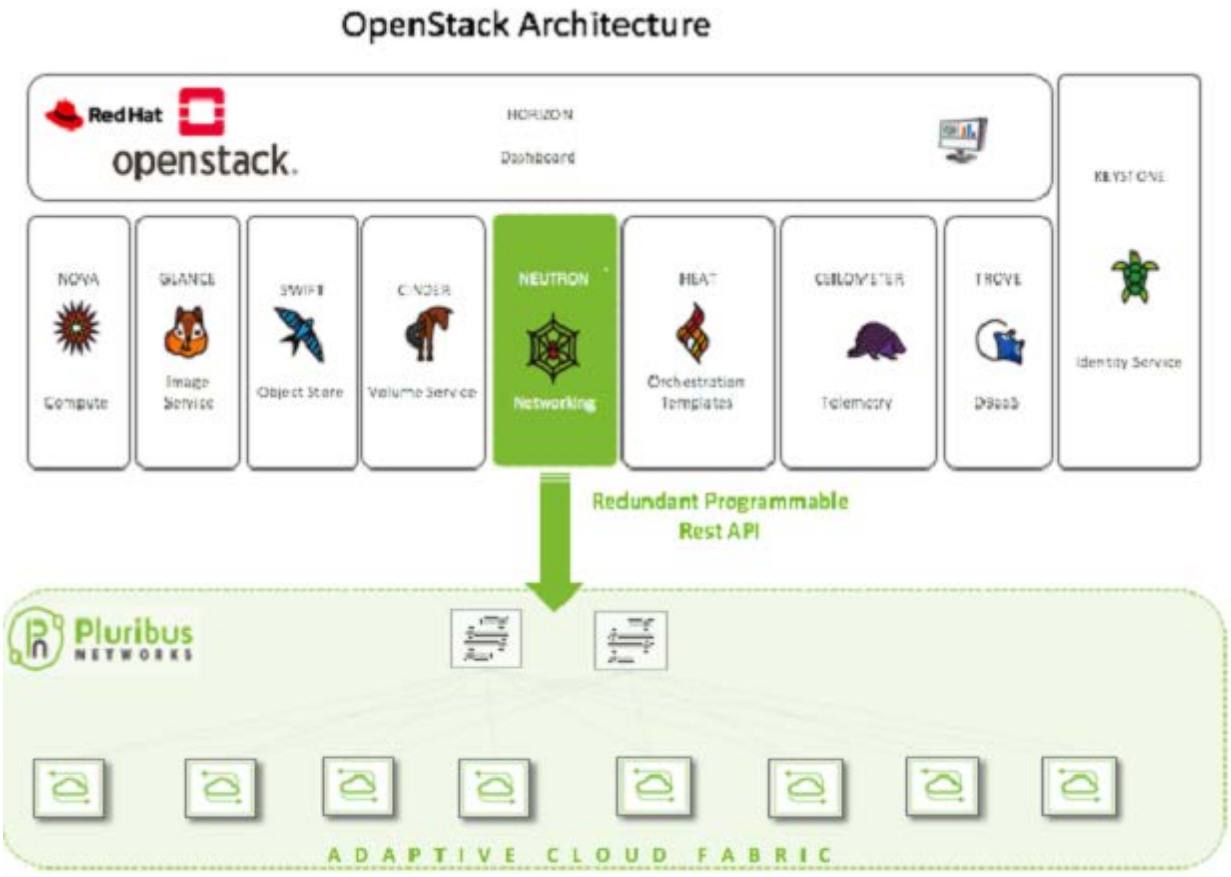


Figure 1: Unified Cloud Fabric - OpenStack Integration

Further, the NetVisor ML2 driver plugin integrates with the Mechanism drivers running on the compute or network nodes of the OpenStack architecture and establishes communication between the Unified Cloud Fabric and the OpenStack Architecture (refer **Figure 2**).

The NetVisor OpenStack Neutron ML2 Plugin (cont'd)

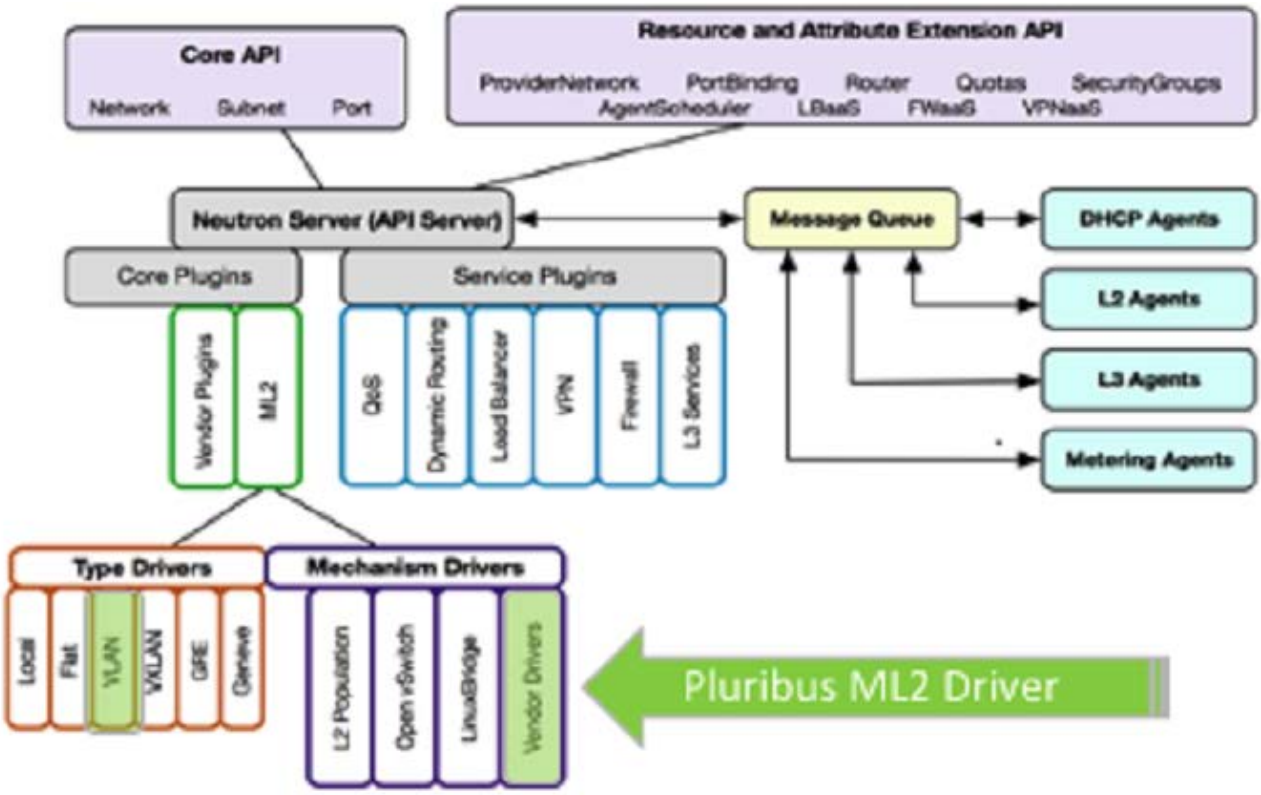


Figure 2: NetVisor ML2 Driver Plugin Integration with Neutron (Networking) Server

As shown in ***Figures 3,*** the ML2 plugin uses the Pluribus REST APIs to configure the NetVisor switches. That is, the ML2 plugin running on the controller node interfaces with the REST APIs to configure the NetVisor switches in the fabric. For this deployment to work, you must configure a username/password in the ML2 config file on the OpenStack controller (triple 0, aka 0-0-0, which is OpenStack-on-OpenStack). These authentication credentials apply to all the switches on the fabric (that is, this is not on per tenant basis) and the credentials (password) is stored in an encrypted base64 format on the ML2 plugin.

The NetVisor Networks OpenStack Neutron ML2 Plugin (cont'd)

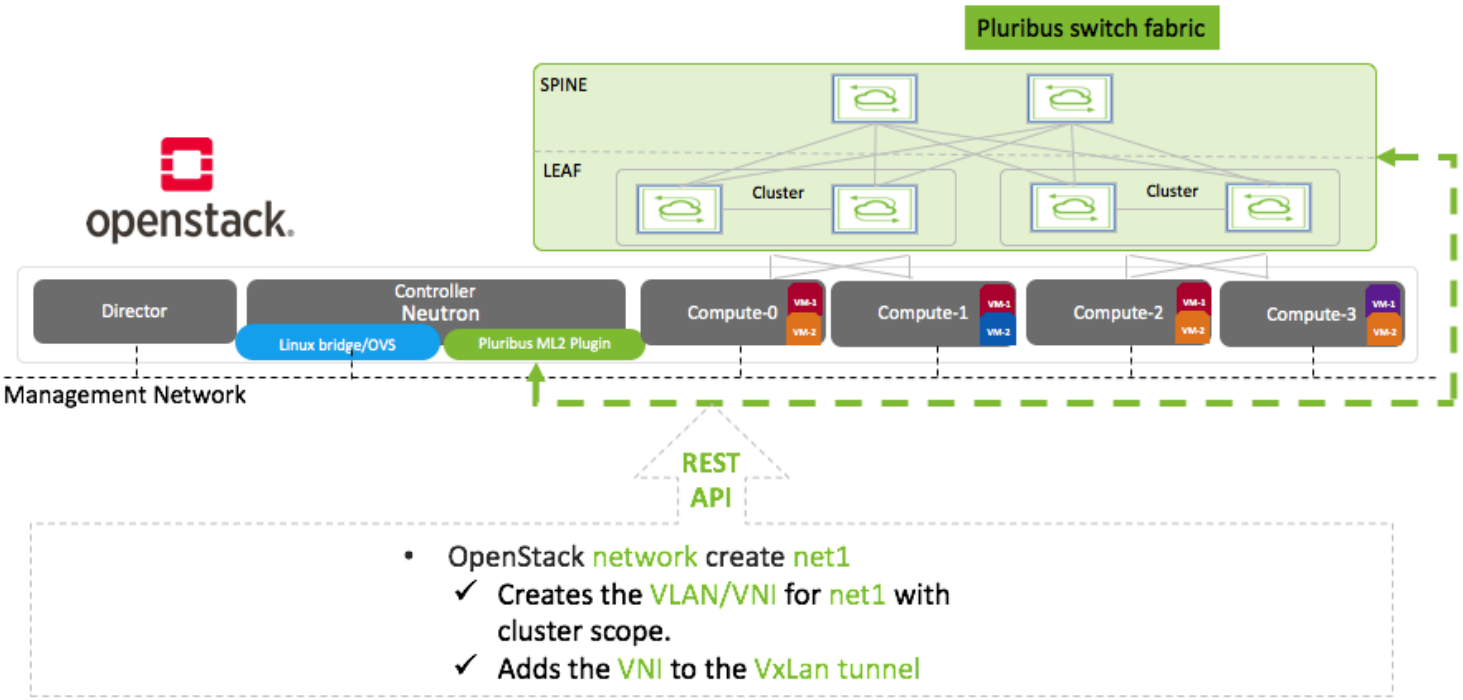


Figure 3: Integration of Unified Cloud Fabric with OpenStack Components using NetVisor ML2 Plugin

The NetVisor OpenStack Neutron ML2 Plugin (cont'd)

The NetVisor ML2 plugin, which provides powerful and simplified overlay automation services, enables seamless connectivity for east-west traffic between virtual machines (VMs) running on compute nodes that are connected to the Unified Cloud Fabric. The VLANs used by the VMs are configured across each cluster and the VLANs are stitched across the fabric using VXLAN tunnels (VXLAN overlays) by the ML2 plugin (refer **Figure 4**).

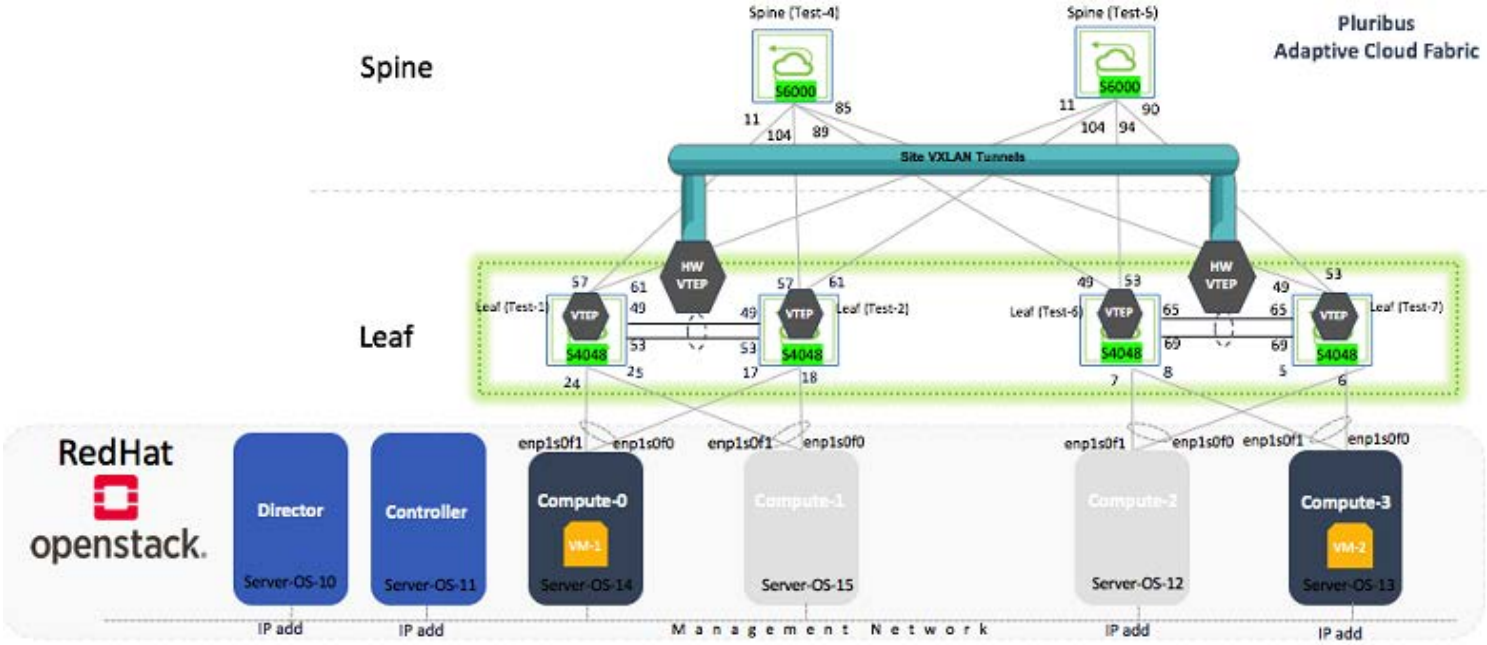


Figure 4: VXLAN Tunnel in Unified Cloud Fabric with OpeComponents

System Requirements and Guidelines

The following system requirements are needed to install and use the NetVisor ML2 plugin:

- Red Hat OpenStack Controller (TripleO deployment, a.k.a, OpenStack-On-OpenStack deployment) : Use any OpenStack controller running Queens release.

> **Note**: The NetVisor features are validated on Red Hat Controller (Queens Release).

- Compute nodes
- Director (TripleO deployment)

Pre-Requisites:

- Enable Web server on the NetVisor switch, see details in the subsequent sections. You can verify if the web server is enabled or not by using the `admin-service-show` command on the NetVisor switch.
- Install and enable Link Layer Discovery Protocol daemon (LLDPd) on the compute and controller nodes. Use the lldp-show command on the NetVisor switch to verify if all the host names are displayed.
- Configure a username/password in the ML2 config file on the OpenStack controller (triple 0). These authentication credentials apply to all the switches on the fabric (that is, this is not on per tenant basis) and the credentials (password) is stored in an encrypted base64 format on the ML2 plugin.

For more details, see the Configuring a Network using NetVisor ML2 Plugin section later in this document.

Usage Guidelines:

- While creating a VLAN network, the VXLAN network identifier (VNI) gets automatically chosen and populated by the ML2 plugin, which is in the format: *11million+ VLAN_ID*.

> **Note**: The VNI range, <11million+4K> is reserved for OpenStack networks and hence we recommend not to use the VLAN ID range of 4K as the VNI calculation is: *11million+ VLAN_ID.*

- In NetVisor OS version 5.2, the NetVisor ML2 plugin supports cluster-based topologies with VXLAN tunnels established. Therefore, all Leaf switches must be part of a cluster in the fabric. That is, standalone switches are not supported by the ML2 plugin.

Downloading and Installing the NetVisor ML2 Plugin

Note: For details on installing the Red Hat controllers and platforms, refer to the Red Hat documentation:

The NetVisor ML2 plugin package is available as a Docker image, which you can download from GitHub. You must download the plugin package and install on the Red Hat OpenStack controller.

Installing the ML2 Plugin on a Red Hat Controller

To install the NetVisor ML2 plugin on a Red Hat Controller:

Based on the RedHat OpenStack Platform and NetVisor OS version, follow *step 1.1* if the plugin package image is available and you want to just pull the docker image from Docker Hub. Follow *step 1.2* if you want to build the docker image and then install the image on the controller.

1) To install the ML2 plugin :

1.1. Download and install image from https://hub.docker.com/ for RedHat OpenStack Platform (TripleO).

Each repository's tag has the RedHat OpenStack version and the NetVisor OS (nvOS) version. For example, if the repository is:

```
pluribusnetworks/openstack-neutron-server-pn:13.0-105.1580118143-5.2.0,
   RedHat OpenStack version = 13.0-105.1580118143
   and
   nvOS version = 5.2.0
```

then, download the docker image - `docker pull pluribusnetworks/openstack-neutron-server-pn:13.0-105.1580118143-5.2.0`

OR

1.2. Build the docker image from source:

- Obtain NetVisor ML2 plugin from Arista TAC
- Extract the tarbal of NetVisor ML2 plugin and cd to the directory that is extracted out of the tarbal and then do: `cd networking-pluribus`
- Build the RPM package: `sudo python setup.py bdist_rpm #`
- Change the directory to find `networking-pluribus-<version>.noarch.rpm`
- Change the directory back to `PNopenstack/networking-pluribus`
- Modify the Dockerfile to:

```
 $ cat Dockerfile
FROM <registry>/rhosp13/openstack-neutron-server<image>:<tag> =====> modify to the
```

```
current neutron server image
MAINTAINER Pluribus Networks
USER root
RUN rpm -e networking-pluribus | echo "1"
COPY dist/networking-pluribus-<version>.noarch.rpm /tmp =====> modify version to the one
generated in dist/
RUN rpm -ivvh /tmp/networking-pluribus-<version>.noarch.rpm =====> modify version to the
one generated in dist/
USER neutron
```

- Build a docker image: `sudo docker build `pwd``
- Tag the image - `docker tag <IMAGE ID> pluribusnetworks/openstack-neutron-server-pn:13.0-105.1580118143-5.2.0`

2) Tag and push 'pluribusnetworks/openstack-neutron-server-pn' to Local Registry

```
sudo docker tag <IMAGE ID> <Local-registry-IP: port>/rhosp13/openstack-neutron-server-pn:13.0-
105.1580118143-5.2.0
sudo docker push <Local-registry-IP: port>rhosp13/openstack-neutron-server-pn:13.0-105.1580118143-
5.2.0
```

3) Update DockerNeutronApiImage and DockerNeutronConfigImage in `overcloud_images.yaml` to the latest version of NetVisor OpenStack Plugin as suggested for that NetVisor OS version.

Edit `overcloud_images.yaml` to:

```
DockerNeutronApiImage: <Local-registry-IP: port>rhosp13/openstack- neutron-server-pn:13.0-
105.1580118143-5.2.0
   DockerNeutronConfigImage: <Local-registry-IP: port>rhosp13/openstack- neutron-server-pn:13.0-
105.1580118143-5.2.0
```

4) Deploy the overcloud from the director node.

5) After the stack changes the status to *UPDATE_COMPLETE*, login to the overcloud and make changes suggested below:

   1) Edit `entry_points.txt` file from the location: `/var/lib/config-data/puppet-generated/neutron/etc/neutron/plugins/ml2/`

```
[neutron.ml2.mechanism_drivers]
pluribus = networking_pluribus.mech_driver.driver:PluribusMechanismDriver
```

   For example: `cd /usr/lib/python2.7/site-packages/neutron-12.0.5-py2.7.egg-info` and Edit `entry_points.txt`.

   2) Edit the `/var/lib/config-data/puppet-generated/neutron/etc/neutron/plugins/ml2/ml2_conf.ini`, add "`pluribus`" to `mechanism_drivers`, and add the `PLURIBUS_PLUGINS` section:

```
mechanism_drivers=openvswitch,baremetal,pluribus
[PLURIBUS_PLUGINS]
# Pluribus switch IP address
#
# pn_switch = 10.20.20.20:80,10.20.20.21:80
```

```
#
# Pluribus switch username and password for REST api access
#
# username = network-admin
#
# password = <base 64 encoded password>
```

The Format of specifying the Pluribus switch is: `pn_switch, <primary_ip:web_port>,<secondary_ip:web_port>`

Note: Because the NetVisor ML2 plugin uses RESTful APIs to configure the Unified Cloud Fabric, you must select two seed nodes in the fabric that can act as the primary node and the secondary node for the REST API to parse. Enter the management IP addresses of the selected seed nodes (primary and secondary) as the `primary_ip` and the `secondary_ip` respectively in the above configuration. Ensure to enter the IP addresses without giving space between primary IP address and secondary IP address as shown in the example above.

Note: You must enable web services on these selected (primary and secondary) nodes.

The web port for each switch can be obtained from the show output of the NetVisor OS CLI command, `admin-service-show` on the NetVisor switch. The web port can be any configurable port on the switch.

Note: The `ml2_conf.ini` and `entry_points.txt` are available in the path: `/var/lib/config-data/puppet-generated/neutron/etc/neutron/plugins/ml2/` and are used by the neutron-server installed on the `neutron_api docker` container.

1. Restart *neutron-server* by restarting the docker container *neutron_api* using the command:

   ```
   sudo docker restart neutron_api
   ```

2. Check the status of the docker image using the command:

   ```
   sudo docker ps -a | grep neutron_api
   ```

For example,

```
[heat-admin@cloud-controller-0 neutron]$ sudo docker ps | grep neutron_api
7cda7ac43cee 192.168.24.1:8787/rhosp13/openstack-neutron-server-pn-ml2:rev10 "dumb-init --singl..."
About an hour ago Up About an hour (healthy) neutron_api
[heat-admin@cloud-controller-0 neutron]$ sudo docker stop neutron_api
neutron_api
[heat-admin@cloud-controller-0 neutron]$ sudo docker start neutron_api
neutron_api
```

For installation details on PackStack deployment, see the Appendix A- ML2 Plugin Installation: PackStack

Deployment section.

Configuring a Network using NetVisor ML2 Plugin

Note: As a network administrator, you must first provision and configure the NetVisor switch fabric. Use OpenStack to configure the NetVisor switch ports so that the VMs in the compute node sends east-west traffic through the NetVisor switch fabric.

Note: For details on the OpenStack configuration commands, see the OpenStack documentation from the Red Hat website or other vendor specific websites.

You can access the ML2 plugin and configure a VLAN network by using two methods: (1) using the OpenStack CLI commands and/or (2) using the Horizon Dashboard.

To configure a network:

Note: Enable web server on the NetVisor switches (the two seed nodes: primary and secondary) specified in the `ml2_conf.ini` file. Use the `admin-service-show` command on the NetVisor switch to verify if the web server is enabled or not.

1) Create the OpenStack network by using the OpenStack CLI command. For example, use the openstack network create <name> --provider-network-type vlan <segment id> command.

   This command creates a VLAN network on the NetVisor switch and associates a VXLAN ID to the VLAN network. The VLAN ID is fetched from the OpenStack Neutron server and the corresponding VXLAN ID will be *<11million + VLAN ID>*. Then, the VXLAN network identifier (VNI) gets added to all the VXLAN Tunnel Endpoint (VTEPs), if configured on the switch fabric, or is added to all static tunnels. All the cluster ports also get added to the VLAN network.

2) Launch the VM instances by using the OpenStack CLI command, `openstack server create`. For example, `openstack server create --image cirros --flavor m1.tiny --network test-network MyInstance.`

Configuring a Network using NetVisor ML2 Plugin (cont'd)

Table - 1 Mapping actions of NetVisor ML2 plugin OpenStack commands on the NetVisor switch fabric:

OpenStack Commands/Actions	Mapping action on NetVisor switches
`openstack network create`	Creates a VLAN network with scope `cluster` on the NetVisor switch fabric and associates the VXLAN network ID (VNI) with the VLAN network, if configured.
`openstack network delete`	Deletes the previously configured VLAN network from the switch fabric.
`openstack subnet create`	The ML2 plugin receives a port create/update request along with the hostname of the controller. The plugin fetches all the controller facing ports on the fabric and adds them to VLAN network. For example, to create a subnet 24, use the command: [root@server-os-4 ~(keystone_admin)]# openstack subnet create --subnet-range *192.10.10.0/24* --network *scorpius* subnet-1. Use the NetVisor OS command, `lldp-show` on the switch to verify the created/updated port details that got added to the VLAN network. See the *Related NetVisor OS CLI commands* section to view show output details.

Table - 1 Mapping actions of NetVisor ML2 plugin OpenStack commands (cont'd)

OpenStack Commands/Actions	Mapping action on NetVisor switches
	Note: Ensure to create the subnets before you execute the `server create` command.
`openstack server create`	When a VM instance is launched, the NetVisor ML2 plugin receives port update request along with the binding host or the compute node details on which the VM is being spun. The ML2 plugin automatically detects all the host facing ports by using the LLDP information on the switch fabric and adds them to the VLAN network.
`openstack server delete`	When a VM instance is deleted, the ML2 plugin receives the port delete request and decides whether to remove switch ports from VLAN or not because multiple VM instances co-exist on the same compute node and in the same network. The host facing ports are removed from the VLAN only if it is the last instance that is being deleted.

Table - 1 Mapping actions of NetVisor ML2 plugin OpenStack commands (cont'd)

OpenStack Commands/Actions	Mapping action on NetVisor switches
Alternatively, use the following commands to manually add or delete ports later.	
`openstack port create`	**Note**: To add a host facing port to the VLAN network, use either of the two commands: `openstack server create` or `openstack port create`. When a port create/update request is received along with the binding host (compute node) information, the NetVisor ML2 plugin automatically detects all the host facing ports on the fabric and adds them to the VLAN received. This is an optional command provided, if you want to configure the VLAN network for a particular host. For example, `openstack port create test-port --host linux-host1 --network test-network`. This command adds the correct switch ports to previously created VLAN for the compute node `linux-host1`. The ML2 plugin queries all the ports connected to the host `linux-host1` by using the LLDP details and adds them to the VLAN network.
`openstack port delete`	Removes all the host facing ports from the VLAN if that is the last port being deleted as explained in `openstack server delete` command description. For example, the `openstack port delete test-port` command removes the port from the VLAN with which it was associated earlier.

Configuring a Network using NetVisor ML2 Plugin (cont'd)

Further to the OpenStack commands described in the table; below is an example of a sample configuration for creating a VLAN network on the OpenStack controller by using the ML2 plugin:

- Create an OpenStack network, `scorpius` with type, `vlan` and ID, 259 on the controller by using the OpenStack commands. The plugin programs VLAN 259 (example here) on all the cluster pair switches in the fabric.

```
[root@server-os-4 ~(keystone_admin)]# openstack network create scorpius --provider-network-type vlan --
provider-physical-network physnet1 --provider-segment 259
```

```
+---------------------------+-----------------------------------+
| Field                     | Value                             |
+---------------------------+-----------------------------------+
| admin_state_up            | UP                                |
| availability_zone_hints   |                                   |
| availability_zones        |                                   |
| created_at                | 2020-03-02T06:12:34Z              |
| description               |                                   |
| dns_domain                | None                              |
| id                        | cd2ee636-50ce-4800-8e46-8f1d9625b54c |
| ipv4_address_scope        | None                              |
| ipv6_address_scope        | None                              |
| is_default                | False                             |
| is_vlan_transparent       | None                              |
| mtu                       | 1500                              |
| name                      | scorpius                          |
| port_security_enabled     | True                              |
| project_id                | df11960a3ec14733849e4b8dd83cf484  |
| provider:network_type     | vlan                              |
| provider:physical_network | physnet1                          |
| provider:segmentation_id  | 259                               |
| qos_policy_id             | None                              |
| revision_number           | 2                                 |
| router:external           | Internal                          |
| segments                  | None                              |
| shared                    | False                             |
| status                    | ACTIVE                            |
| subnets                   |                                   |
| tags                      |                                   |
| updated_at                | 2020-03-02T06:12:34Z              |
+---------------------------+-----------------------------------+
[root@server-os-4 ~(keystone_admin)]#
```

Configuring a Network using NetVisor ML2 Plugin (cont'd)

- On the NetVisor switch, verify if a `cluster` scoped VLAN network is created and if the network has all the cluster scope ports within the VLAN by using the NetVisor OS CLI command:

```
CLI (network-admin@sw-1) > vlan-show id 259

switch id  type    vxlan  auto-vxlan  scope   description    active  ports   untagged-ports active-edge-ports
------ --- ------ ------- ---------- ------- -------------- ------ ---------- ------------- --------------
sw-1   259 public 11000259   no       cluster  ostack-vlan-259  yes   0,27-30,272  none           none
sw1-1 259 public 11000259   no       cluster  ostack-vlan-259  yes   23-26,272    none           none
sw1-2 259 public 11000259   no       cluster  ostack-vlan-259  yes   23-26,272    none           none
sw-2   259 public 11000259   no       cluster  ostack-vlan-259  yes   0,27-30,272  none           none
```

> **Note**: The VXLAN in the show output is of the range <11 million+VLAN ID>.

Configuring a Network using NetVisor ML2 Plugin (cont'd)

- Assign a Subnet, `192.10.10.0/24` to the previously created network, `Scorpius` on the controller:

```
[root@server-os-4 ~(keystone_admin)]# openstack subnet create --subnet-range 192.10.10.0/24 --network
scorpius subnet-1
+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| allocation_pools  | 192.168.76.2-192.168.76.254          |
| cidr              | 192.168.76.0/24                      |
| created_at        | 2020-03-02T06:16:04Z                 |
| description       |                                      |
| dns_nameservers   |                                      |
| enable_dhcp       | True                                 |
| gateway_ip        | 192.168.76.1                         |
| host_routes       |                                      |
| id                | eb932f42-4184-4411-99f5-16c3f33cfbf0 |
| ip_version        | 4                                    |
| ipv6_address_mode | None                                 |
| ipv6_ra_mode      | None                                 |
| name              | subnet-1                             |
| network_id        | cd2ee636-50ce-4800-8e46-8f1d9625b54c |
| prefix_length     | None                                 |
| project_id        | df11960a3ec14733849e4b8dd83cf484     |
| revision_number   | 0                                    |
| segment_id        | None                                 |
| service_types     |                                      |
| subnetpool_id     | None                                 |
| tags              |                                      |
| updated_at        | 2020-03-02T06:16:04Z                 |
+-------------------+--------------------------------------+
[root@server-os-4 ~(keystone_admin)]
```

- Verify the subnet list by using the command:

```
[root@server-os-4 ~(keystone_admin)]# openstack subnet list
+--------------------------------------+----------+--------------------------------------+----------------+
| ID                                   | Name     | Network                              | Subnet         |
+--------------------------------------+----------+--------------------------------------+----------------+
| eb932f42-4184-4411-99f5-16c3f33cfbf0 | subnet-1 | cd2ee636-50ce-4800-8e46-8f1d9625b54c | 192.168.76.0/24 |
+--------------------------------------+----------+--------------------------------------+----------------+
[root@server-os-4 ~(keystone_admin)]#
```

Configuring a Network using NetVisor ML2 Plugin (cont'd)

- Create a VM instance on server-os-9 compute node:

```
[root@server-os-4 ~(keystone_admin)]# openstack server create --image cirros --flavor m1.tiny --availability-
zone nova:server-os-9.pluribusnetworks.com   --nic net-id=cd2ee636-50ce-4800-8e46-8f1d9625b54c vm-1
+-------------------------------------+----------------------------------------------+
| Field                               | Value                                        |
+-------------------------------------+----------------------------------------------+
| OS-DCF:diskConfig                   | MANUAL                                       |
| OS-EXT-AZ:availability_zone         | nova                                         |
| OS-EXT-SRV-ATTR:host                | None                                         |
| OS-EXT-SRV-ATTR:hypervisor_hostname | None                                         |
| OS-EXT-SRV-ATTR:instance_name       |                                              |
| OS-EXT-STS:power_state              | NOSTATE                                      |
| OS-EXT-STS:task_state               | scheduling                                   |
| OS-EXT-STS:vm_state                 | building                                     |
| OS-SRV-USG:launched_at              | None                                         |
| OS-SRV-USG:terminated_at            | None                                         |
| accessIPv4                          |                                              |
| accessIPv6                          |                                              |
| addresses                           |                                              |
| adminPass                           | yR7hki8eH9Di                                 |
| config_drive                        |                                              |
| created                             | 2020-03-10T22:44:11Z                         |
| flavor                              | m1.tiny (1)                                  |
| hostId                              |                                              |
| id                                  | 32b54fc5-b6bb-4778-8e19-48ed28e1e4a6         |
| image                               | cirros (ddac232d-8cc2-45f2-91b8-2628f5c17697) |
| key_name                            | None                                         |
| name                                | vm-1                                         |
| progress                            | 0                                            |
| project_id                          | df11960a3ec14733849e4b8dd83cf484             |
| properties                          |                                              |
| security_groups                     | name='default'                               |
| status                              | BUILD                                        |
| updated                             | 2020-03-10T22:44:11Z                         |
| user_id                             | 93b2ce819c0347a9aa6ecf7ab5c4ba28             |
| volumes_attached                    |                                              |
+-------------------------------------+----------------------------------------------+
[root@server-os-4 ~(keystone_admin)]#
```

- You can use the `openstack network delete` command to delete the previously created network or `openstack port delete` command to delete the ports.

Configuring a Network using NetVisor ML2 Plugin (cont'd)

After the above configurations are done, verify the details by using the NetVisor OS command, `vlan-show id` *259*:

```
CLI (network-admin@aq-os-2) > vlan-show id 259

switch   id  type   vxlan    replicators scope   description    active stats ports              untagged-ports active-edge-ports
-------- --- ------ -------- ---------- ------  -------       ------- ---- -----------      -------------- -------------
aq-os-2  259 public 11000259  none      cluster ostack-vlan-259 yes    no    0-2,27-30,272-273    none           273
leo-os-1 259 public 11000259  none      cluster ostack-vlan-259 yes    no    23-26,272            none           none
leo-os-2 259 public 11000259  none      cluster ostack-vlan-259 yes    no    23-26,272            none           none
aq-os-1  259 public 11000259  none      cluster ostack-vlan-259 yes    no    0-2,11,27-30,272-273 none           273
```

Note that in the above output, the VXLAN IDs corresponds to <11million+VLAN ID> and also the ports are added to the VLAN 259.

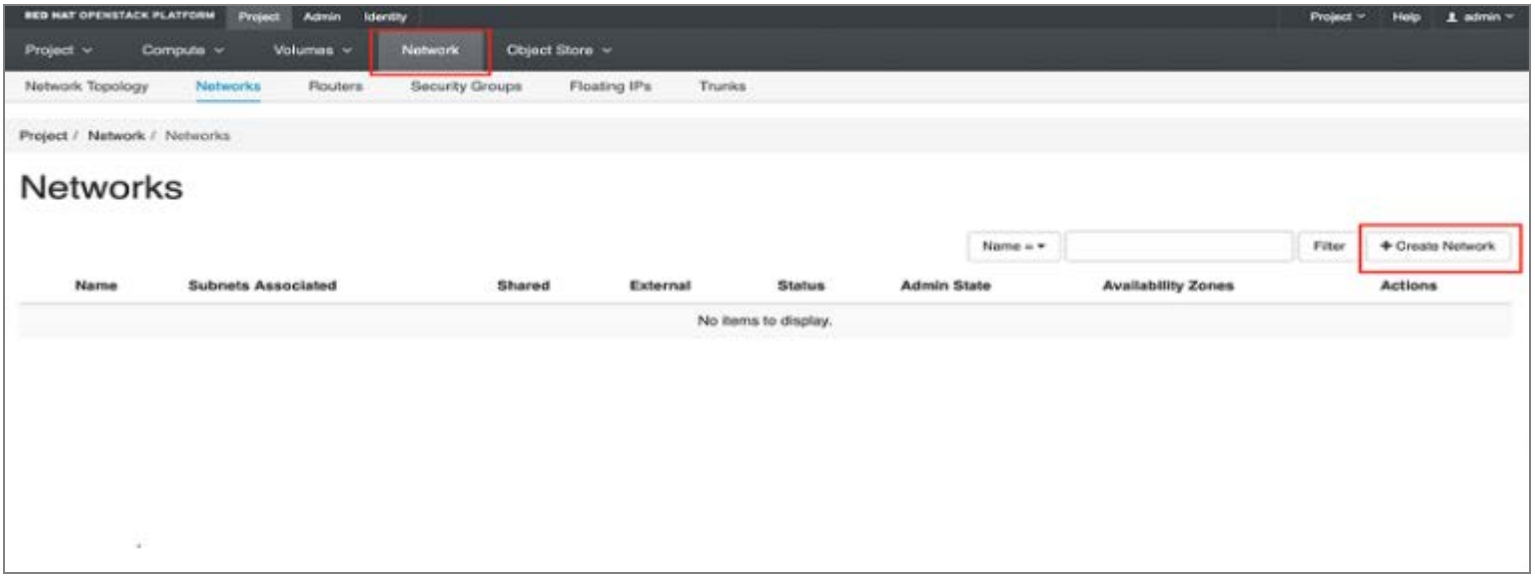
For more details on verifying the configurations using the corresponding NetVisor OS CLI commands, see the Related NetVisor OS CLI commands section.

Using the Horizon Dashboard to Configure a Network

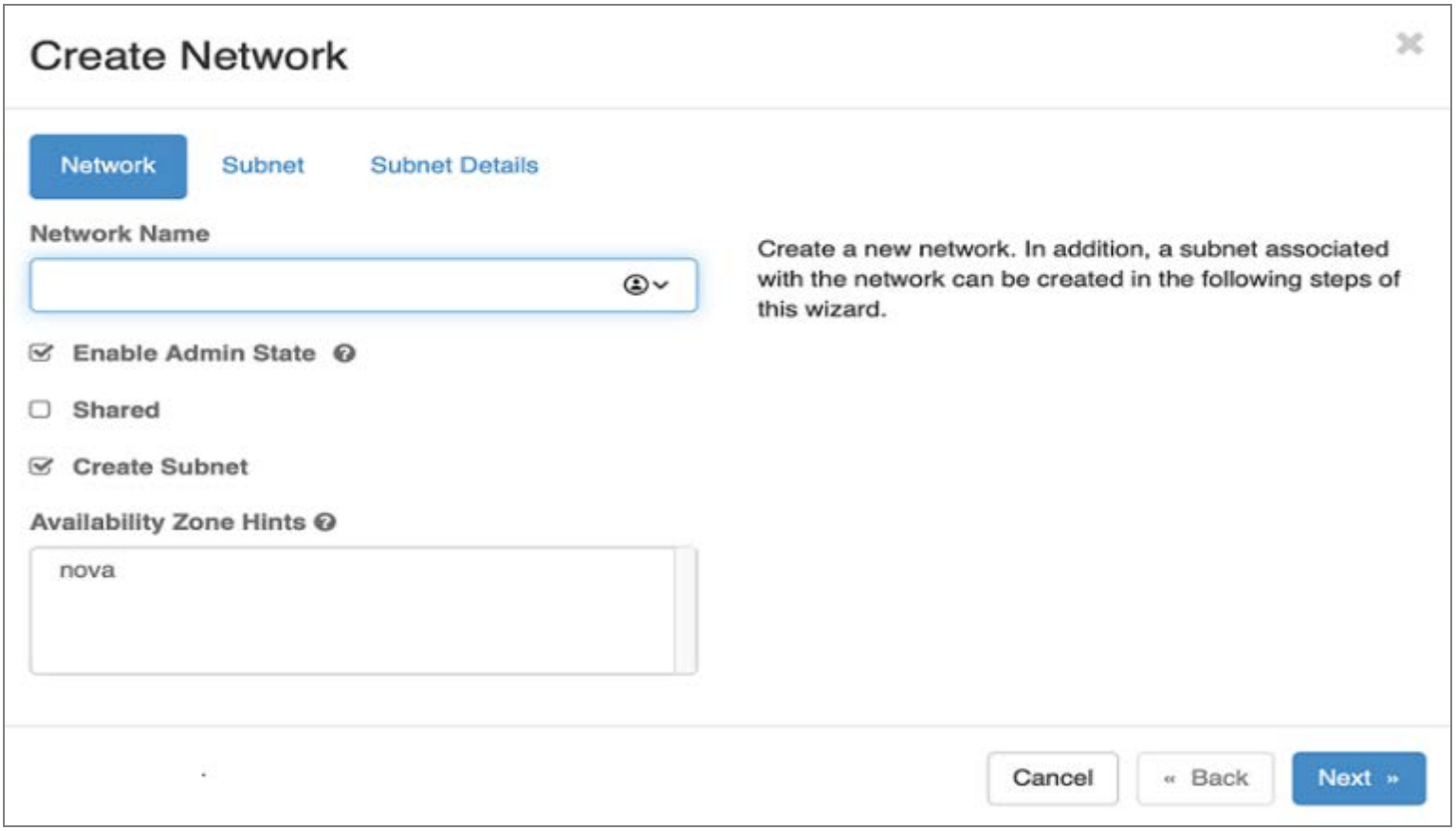
In addition to using the OpenStack CLI commands, you can also use the Horizon dashboard to configure a VLAN network. Follow the steps below to create a network:

Refer to Red Hat OpenStack Horizon Dashboard for the latest and most updated user interface. The below screenshots are only for representative purposes.

1. On the Horizon dashboard, see below screen-shot to access the user interface:

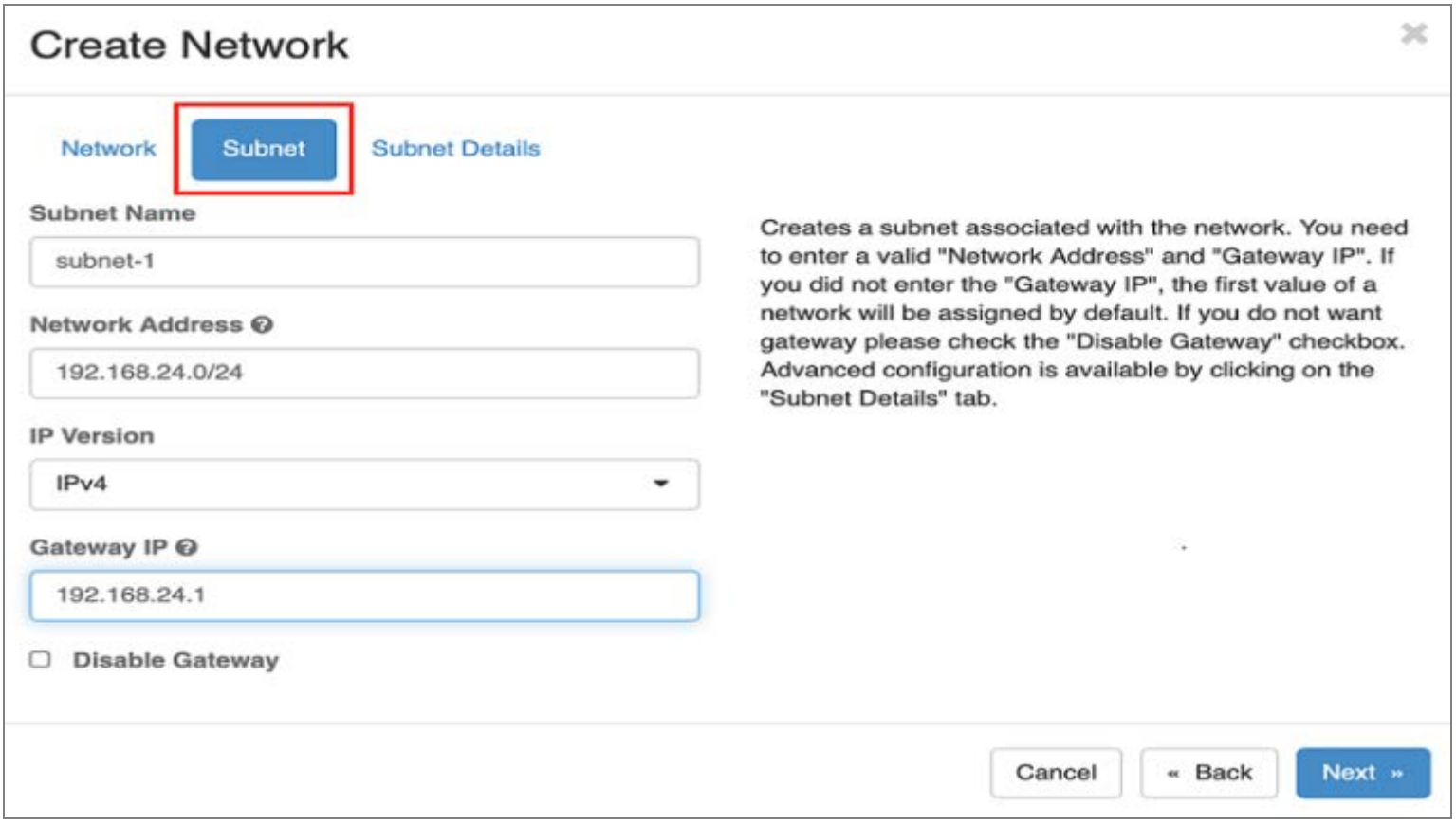


2. To create a network using the dashboard:

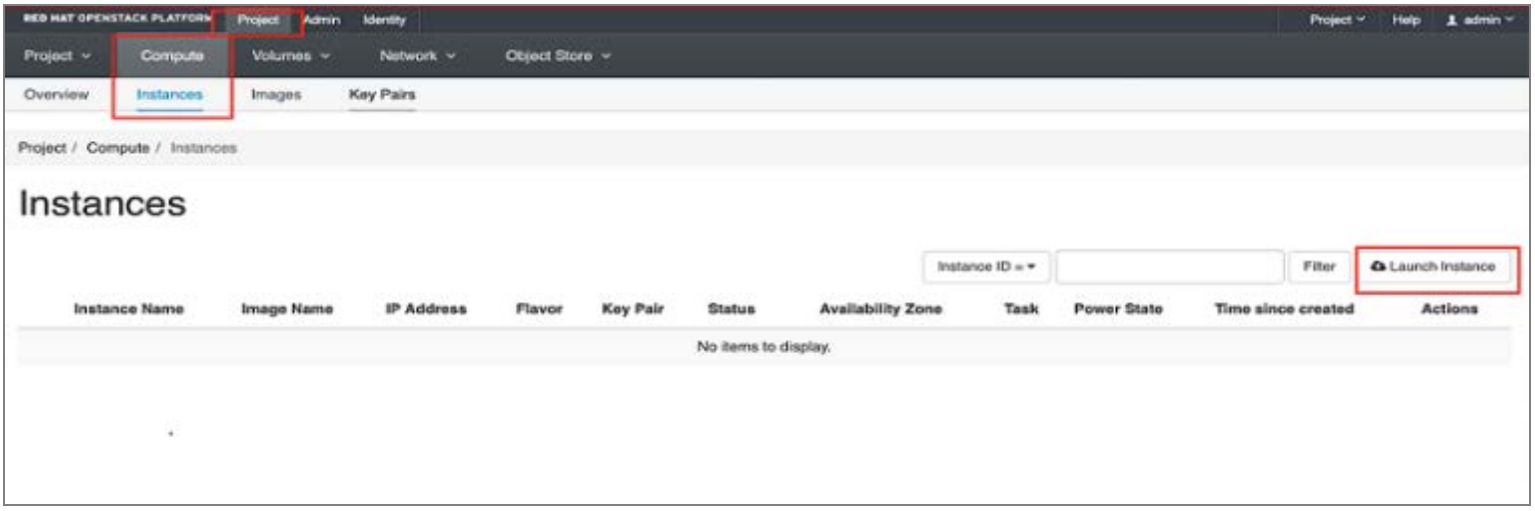

Create Network
Network
Subnet
Subnet Details
Network Name
Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.
Enable Admin State
Shared
Create Subnet
Availability Zone Hints
nova
Cancel
« Back
Next »

Using the Horizon Dashboard to Configure a Network (cont'd)

3. To create a subnet using the dashboard:



4. To create a VM instance using the dashboard:



Guidelines, Errors, and Exceptions

The following is a list of guidelines, limitations and exceptions to keep in mind while using the NetVisor ML2 plugin with the OpenStack deployment:

- **Vendor/versions of supported OpenStack**: The NetVisor ML2 plugin is supported on OpenStack Queens release and with RedHat OpenStack 13.0 distro. However, the ML2 plugin is generic and may be compatible with later releases also.

- **Supported deployment models of OpenStack** : The ML2 plugin package is available from GitHub based on the OpenStack controller implementation. For Red Hat OpenStack controller, the ML2 plugin is packaged as a *Docker* image. For other controllers, the plugin can be downloaded from Git Hub.

- **Pre-requisites** on NetVisor switches before creating on OpenStack node:
    - Configure all the host (compute/controller) facing ports as edge ports
    - Configure leaf clusters
    - Create VTEP and ensure auto-tunnels are created or create static tunnels.

- **Configurations required on NetVisor switch before using OpenStack** : You must first create the VTEPs (which in turn creates the auto-tunnels) or static tunnels on the NetVisor switch fabric.

- **VLAN vs VXLAN provisioning**: When you configure a VLAN network using the `openstack network create` command, the ML2 plugin creates the VLAN and assigns a VNI from a predefined offset (11million). The VNI is added to all the VTEPs or static tunnels. The VLANs used by the VMs running on the compute nodes are configured on the NetVisor fabric by the ML2 plugin. The ML2 plugin adds or removes the VNI to the tunnel.

> **Note**: NetVisor ML2 plugin supports only VLAN-type network. Requests for other network types such VXLAN, Flat, and GRE will fail to create.

- **Supported OpenStack controllers**: The ML2 plugin for Netvisor version 5.2.0 supports Red Hat OpenStack controller. However, the ML2 plugin is generic and should work with Ubuntu OpenStack controllers also.

Guidelines, Errors, and Exceptions (cont'd)

Below is a list of error messages and exceptions to check for while configuring a VLAN network:

- All the errors are logged in `/var/log/container/neutron/server.log` for Red Hat OpenStack controller (overcloud) and in `/var/log/neutron/server.log` for other OpenStack deployments.

- Errors that you may encounter while creating an OpenStack network by using the `openstack network create` command:
  - If a VLAN already exists on the fabric, the ML2 plugin displays an error message: *Vlan <vlan-id> already exists on fabric*.
  - If a VXLAN is already in use, that is, if the VNI selected for a VLAN is already used by another VLAN in the fabric, an error message: *vxlan <vni> already in use* is displayed.
  - If there are no VTEPs or static tunnels configured on the switch fabric, an error message: *Vlan create failed: vtep and tunnels not present on the switch fabric* is displayed.

- Error messages you may encounter while deleting an OpenStack network by using the `openstack network delete` command.
  - If the VLAN does not exist on the cluster switch, an error message: *VLAN <vlan-id> does not exist in switch <switch name>* is logged in *server.log* file. However, the ML2 plugin deletes the VLAN on other clusters and does not display any failure messages.

- Errors displayed while creating an openstack port create and launching an instance:
  - *No switch port found for host <hostname>*: When a port create/update request is received by the plugin along with the binding host details, the plugin finds the host facing ports by using the `lldp-show` command on NetVisor switch. If no host facing ports are found, an error message: *No switch port found for host <hostname>* is displayed. The `openstack port create` command is executed only if the ML2 plugin can find at least one switch port for the host.
  - When some host facing ports are down, those ports are not displayed in the `lldp-show` command output and are not added to the VLAN. The ML2 plugin runs a daemon thread to periodically monitor ports table and checks for missing ports that were not added to the VLAN by comparing the `lldp-show` command output. When the port is back up and appears in the `lldp-show` command output, then the daemon thread adds the ports to the VLAN.

- Other errors:
  - If the switch fails to configure the fabric for any reason and returns an error, then the plugin raises an exception with a failure message in the REST API response.
  - If both the primary and secondary IP addresses specified in *ml2_conf.ini* file are unreachable during the configuration, the plugin displays an error message: *The switches <switch name> are offline*.

Related NetVisor OS CLI commands

You can verify the mappings of OpenStack configurations on the NetVisor switch fabric by using the following NetVisor OS CLI commands:

- `admin-service-show`
- `cluster-show`
- `lldp-show`
- `vlan-show`

For example, below is a sample output for the `admin-service-show` command to check the web ports:

```
CLI (network-admin@switch-2) > admin-service-show

switch     if  ssh nfs web web-ssl web-ssl-port web-port snmp net-api icmp
--------  ---- --- --- --- ------- ------------ -------- ---- ------- ----
switch-2  mgmt  on  off on  off     443          80      off  off     on
switch-2  data  on  off on  off     443          80      off  off     on
switch-1  mgmt  on  off on  off     443          80      off  off     on
switch-1  data  on  off on  off     443          80      off  off     on
```

To change the web port details, use the `admin-service-modify if mgmt web` command:

```
CLI (network-admin@switch-2) > admin-service-modify if mgmt web
```

After you run the `openstack network create` command, verify the configuration on the NetVisor switch by using the `cluster-show` command. The cluster ports are added to the VLAN network. :

```
CLI (network-admin@aq-os-2) > cluster-show

switch   name           state  cluster-node-1 cluster-node-2 tid  mode   ports     remote-ports cluster-sync-    cluster-sync-
                                                                                                                  timeout(ms)
         offline-count
-------- -------------- ------ -------------- -------------- ---- ------ --------- ------------ -------------- -------------
leo-os-1 Cluster-leo-os online leo-os-1       leo-os-2       8075 slave  23-26,272 23-26,272    2000                    3
leo-os-2 Cluster-leo-os online leo-os-1       leo-os-2       8075 master 23-26,272 23-26,272    2000                    3
aq-os-2  Cluster-aq-os  online aq-os-2        aq-os-1        7968 slave  27-30,272 27-30,272    2000                    3
aq-os-1  Cluster-aq-os  online aq-os-2        aq-os-1        7968 master 27-30,272 27-30,272    2000                    3
```

Related NetVisor OS CLI commands (cont'd)

After you run the openstack subnet create command, verify using the `lldp-show` command along with the controller node details on the switch:

```
CLI (network-admin@aq-os-1) > lldp-show sys-name server-os-4.pluribusnetworks.com

switch  local-port chassis-id        port-id           port-desc sys-name
------- ---------- ----------------- ----------------- --------- ------------------------------
aq-os-1 1          0c:c4:7a:6c:5c:52 00:12:c0:88:0c:36 enp4s0f0  server-os-4.pluribusnetworks.com
aq-os-1 2          0c:c4:7a:6c:5c:52 00:12:c0:88:0c:37 enp4s0f1  server-os-4.pluribusnetworks.com
aq-os-1 11         0c:c4:7a:6c:5c:52 0c:c4:7a:6c:5c:53 eno2      server-os-4.pluribusnetworks.com
aq-os-2 1          0c:c4:7a:6c:5c:52 00:12:c0:88:0c:38 ens73f0   server-os-4.pluribusnetworks.com
aq-os-2 2          0c:c4:7a:6c:5c:52 00:12:c0:88:0c:39 ens73f1   server-os-4.pluribusnetworks.com
```

In the above output, *server-os-4.pluribusnetworks.com* is the controller name and all the above ports are added to the VLAN network (VLAN 259 in this example). Verify using the `vlan-show id` *259* command:

```
CLI (network-admin@aq-os-2) > vlan-show id 259

switch   id  type   vxlan   replicators scope   description    active stats ports              untagged-ports active-edge-ports
-------- --- ------ -------- ---------- ------  -------       ------- ---- -----------      -------------- -------------
aq-os-2  259 public 11000259  none      cluster ostack-vlan-259 yes    no    0-2,27-30,272-273    none           273
leo-os-1 259 public 11000259  none      cluster ostack-vlan-259 yes    no    23-26,272            none           none
leo-os-2 259 public 11000259  none      cluster ostack-vlan-259 yes    no    23-26,272            none           none
aq-os-1  259 public 11000259  none      cluster ostack-vlan-259 yes    no    0-2,11,27-30,272-273 none           273
```

Related NetVisor OS CLI commands (cont'd)

After you run the OpenStack server/port create, verify using the `lldp-show` command along with the compute node details on the switch:

```
CLI (network-admin@aq-os-1) > lldp-show sys-name server-os-9.pluribusnetworks.com

switch   local-port chassis-id       port-id          port-desc sys-name
-------- ---------- ---------------- ---------------- --------- ------------------------------
leo-os-1 18         40:8d:5c:d6:20:e3 00:12:c0:80:30:23 enp1s0f1  server-os-9.pluribusnetworks.com
leo-os-2 18         40:8d:5c:d6:20:e3 00:12:c0:80:30:22 enp1s0f0  server-os-9.pluribusnetworks.com
```

In the above output, server-os-9.pluribusnetworks.com is the compute node name and the above ports are added to the VLAN network (VLAN 259 in this example). Verify using the `vlan-show id` *259* command:

```
CLI (network-admin@aq-os-2) > vlan-show id 259

switch   id  type   vxlan    replicators scope   description     active stats ports               untagged-ports active-edge-ports
-------- --- ------ -------- ----------- ------- --------------- ------ ----- ------------------- -------------- ----------------
aq-os-2  259 public 11000259 none        cluster ostack-vlan-259 yes    no    0-2,27-30,272-273    none           273
leo-os-1 259 public 11000259 none        cluster ostack-vlan-259 yes    no    18,23-26,272         none           18
leo-os-2 259 public 11000259 none        cluster ostack-vlan-259 yes    no    18,23-26,272         none           18
aq-os-1  259 public 11000259 none        cluster ostack-vlan-259 yes    no    0-2,11,27-30,272-273 none           273
```

For details on the above NetVisor OS commands, see the *Configuration Guide for NetVisor OS* and *Command Reference Guides for NetVisor OS* from the Pluribus Networks website.

Appendix A- ML2 Plugin Installation: PackStack Deployment

> For details on installing PackStack deployment, refer to PackStack page:
> https://www.rdoproject.org/install/packstack/

To download, install, and configure the NetVisor OpenStack ML2 plugin:

1. Install the NetVisor ML2 Plugin package on to a RedHat Openstack Controller:

    a) Obtain NetVisor ML2 plug in from Arista TAC
    b) Extract the tarbal of NetVisor ML2 plugin and `cd` to the directory that is extracted out of the tarbal and then do: `cd networking-pluribus`
    c) `sudo python setup.py install`

2. Edit `entry_points.txt file in neutron.egg-info`

```
[neutron.ml2.mechanism_drivers]
pluribus = networking_pluribus.mech_driver.driver:PluribusMechanismDriver
```

    For example:

```
pip show neutron
cd /usr/lib/python2.7/site-packages/neutron-12.0.5-py2.7.egg-info
Edit entry_points.txt
```

3. Edit the `/etc/neutron/plugins/ml2/ml2_conf.ini`, add "`pluribus`" to mechanism drivers, and add the `PLURIBUS_PLUGINS` section: `mechanism_drivers=openvswitch,baremetal,`**`pluribus`**

```
[PLURIBUS_PLUGINS]
# Pluribus switch IP address
#
# pn_switch = 10.20.20.20:80,10.20.20.21:80
#
# Pluribus switch username and password for REST api access
#
# username = network-admin
#
# password = <base 64 encoded password>
```

Appendix A- ML2 Plugin Installation: PackStack Deployment (cont'd)

The Format of specifying the NetVisor switch is:
`pn_switch,<primary_ip:web_port>,<secondary_ip:web_port>`

Note: Because the NetVisor ML2 plugin uses RESTful APIs to configure the Unified Cloud Fabric, you must select two seed nodes in the fabric that can act as the primary node and the secondary node for the REST API to parse. Enter the management IP addresses of the selected seed nodes (primary and secondary) as the `primary_ip` and the `secondary_ip` respectively in the above configuration. Ensure to enter the IP addresses without giving space between primary IP address and secondary IP address as shown in the example above.

Note: You must enable web services on these selected (primary and secondary) nodes.

The web port for each switch can be obtained from the show output of the NetVisor OS CLI command, `admin-service-show` on the NetVisor switch. The web port can be any configurable port on the switch.

Note: The `ml2_conf.ini and entry_points.txt` are available in the path: `/var/lib/config-data/puppet-generated/neutron/etc/neutron/plugins/ml2/` and are used by the neutron-server installed on the `neutron_api docker` container.

1. Restart neutron-server service:

   ```
   sudo systemctl restart neutron-server
   ```

2. Check the Neutron logs and verify the status of neutron-server service from: `/var/log/neutron/server.log`

   ```
   sudo systemctl status neutron-server
   ```

Note: Adding the pluribus keyword to the `ml2_conf.ini` file ensures the Neutron server to fetch NetVisor ML2 driver while configuring the network.