Cisco Nexus Fabric OpenStack Enabler Installation for the VXLAN BGP EVPN fabric

Hardware and Software Requirements

Cisco Nexus Fabric Enabler is a set of software applications that interacts with OpenStack through its open APIs to allow users to connect Cisco Nexus 5600, 6000, 7000 and 9000 Series platform switches as the network to the OpenStack compute nodes to form a cloud.

The uplink port of the server is directly connected to the fabric leaf switch and it requires that there is no middle device that has a bridging functionality, as LLDP needs to run between the server port and leaf switch

port to signal the VM presence to the fabric for reachability. However, FEX devices can be connected between the server and the leaf switch as NX-OS 7.3 and above supports LLDP proxy running in the FEX.

OpenStack Juno release October 2014 or later, needs to be installed on any server using an OpenStack installer of your choice. This guide does not cover OpenStack installation.

- The qualification for this release is based on an OpenStack installation using DevStack.
- The qualification information of using other major third party OpenStack installers will be made available in the release notes document.

There are three ways to install OpenStack. They are given below:

- Use DevStack for installation. This is primarily for development and testing purposes and not for production level installation.
- Use a third-party supported OpenStack installation package. Nexus Fabric Enabler is officially qualified with Red Hat Enterprise Linux (RHEL) OSP 7.
- Use the OpenStack general installation guidelines.

In addition to OpenStack installation, the other required installation is LLDP Agent Daemon (lldpad), an open source free software, similar to OpenStack. The lldpad software is an agent daemon that supports VDP. For Linux distributions that packages lldpad using the source code after May 2015, follow the official installation procedure of the distribution to install lldpad. For example, in RHEL 7.2, install lldpad using **yum install lldpad**. Otherwise, follow the respective README notes to build and install the latest versions of lldpad on all servers used as compute nodes.

Note that, since OpenStack and its associated pieces of software for this purpose is open source software, it generally requires you to install the various software applications on your target servers as the applications are from different sources.

Cisco Nexus Fabric Enabler can be downloaded from the website https://github.com/CiscoSystems/fabric_enabler.

Supported Cisco Nexus Hardware and Software versions

Enabler Version	DCNM Version	Supported Platform	Cisco Nexus Switch Version	Fabric Type	OpenStack Release and Distribution
2.0	Cisco DCNM 10.1(1)	Nexus 9300	Cisco NX-OS 7.0(3)I5(2)	Cisco Programmable Fabric (PF) with VXLAN BGP EVPN	• Verified with RHEL OSP 8 (Liberty based). • Verified with Mitaka Release (devstack, Centos7.x, Ubuntu 14.04).

Enabler Version	DCNM Version	Supported Platform	Cisco Nexus Switch Version	Fabric Type	OpenStack Release and Distribution
2.0	Cisco DCNM 10.1(1)	Nexus 5600	Cisco NX-OS 7.3(0)N1(1)	Cisco Programmable Fabric (PF) with VXLAN BGP EVPN Dynamic Fabric Automation (DFA)	• Verified with RHEL OSP 8 (Liberty based). • Verified with RHEL OSP 7 (Kilo based). • Verified with Mitaka Release (devstack, Centos7.x, Ubuntu 14.04).
1.1	Cisco DCNM 7.2(3)	Nexus 5600 Nexus 6000	Cisco NX-OS 7.1(0)N1(1)	DFA	Verified with RHEL OSP 6 (Juno based).

Note Cisco Nexus Fabric Enabler only supports the native DHCP server of OpenStack. For supporting any other external DHCP server, the DHCP server should have an IP Address Management (IPAM) plugin integrated with OpenStack. This is needed because, the IP address allocation module in OpenStack is a separate entity from the DHCP server and so it should be synchronized with the DHCP server. Usually, the IPAM plugins ensure that the DHCP server is synchronized with the IP address allocation logic in OpenStack.

Before you start

Following are the prerequisites for installing Nexus fabric OpenStack Enabler:

- Ensure that the fabric is up and running with the Power On Auto Provisioning (POAP) procedure being completed. This does not have any dependency with enabler installation or working, but this is a needed step for the overall functionality. POAP can be done even after the enabler installation is complete.
- Ensure that the interface on the switch connected to the server (uplink) has the following configuration:
  - **switchport mode trunk**
  - **encapsulation dynamic vdp**. This command is needed in switch versions NX-OS 7.3 and above.
- Ensure you know enough about OpenStack in general and have it installed and running in your setup (without the L3 service, or the Neutron router configured). OpenStack should be installed on the controllers and compute nodes.
- Ensure that the keystone and neutron services are running on the same server (the controller).
- Ensure each OpenStack compute node is connected to the leaf switch and the OpenStack control node is connected to Cisco DCNM through an IP network. Both the compute node and control node should be connected to the switches.

- Ensure that DCNM is installed and reachable from the controller node (OpenStack Controller) and from the Nexus Fabric cluster. Without this step, the enabler server process will gracefully exit.
- Ensure that all nodes have the same user account with the same credentials, and also that password-less sudo is enabled on all nodes.

  It can be checked in /etc/sudoers

  ```
  %sudo ALL=(ALL:ALL) NOPASSWD: ALL
  ```
- The RabbitMQ server must run on all the nodes. RabbitMQ is a general messaging scheme used by OpenStack for passing information.
- Ensure that lldpad is installed on all the controller and compute nodes. Refer the *Install lldpad* section for more details.

Install lldpad

On RHEL 7, lldpad can be installed by using the command **sudo yum install lldpad**.

On the other Linux distributions, ensure that their package manager is picking up lldpad after May, 2015. If not, the following needs to be done:

- Clone the repository used for installation—Open-LLDP.org.
- Follow the instructions in the README file to build and install lldpad. You do not need the Linux kernel installation, but only the application.

Use the lldpad open source community mailing list http://open-lldp.org/ for any general queries.

Note All servers that will act as controller and compute nodes will need lldpad installation. The lldpad installation is also needed on the controllers because the solution uses the native DHCP server provided by OpenStack, and the DHCP virtual interface also needs to be auto-configured in the attached leaf switch.

Note The OpenStack DHCP service is supported while the Cisco DCNM DHCP service is not supported.

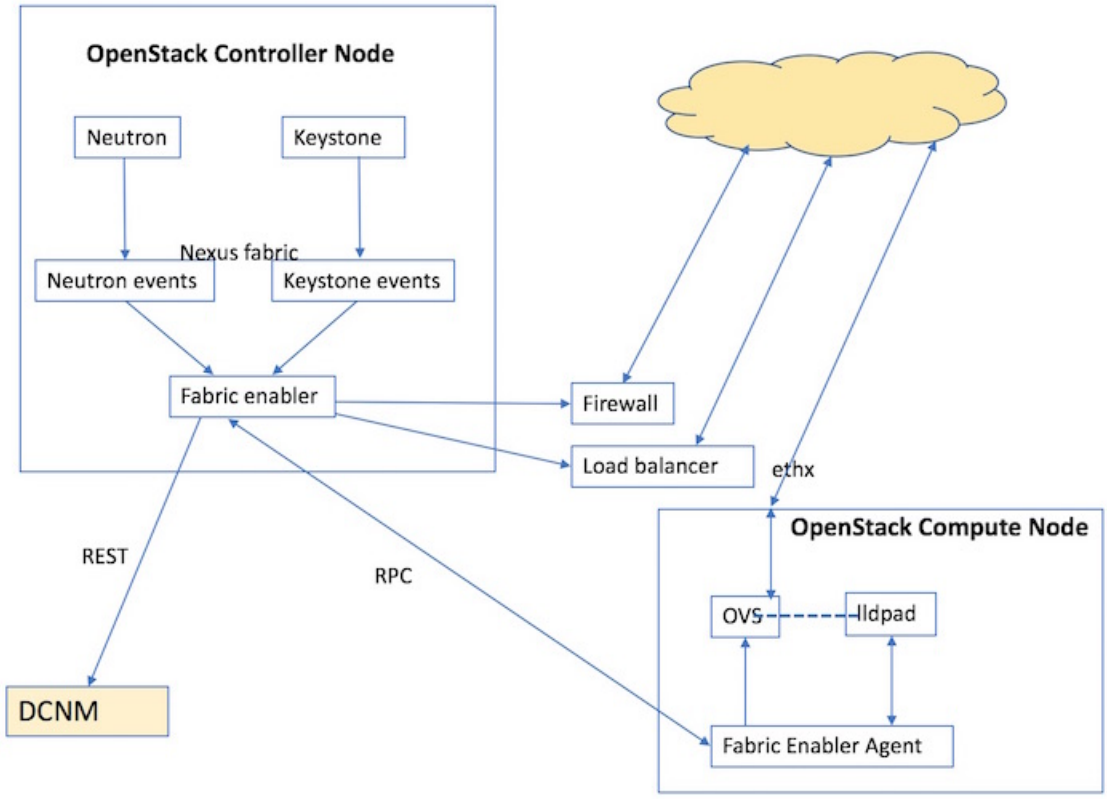
Assumptions

- You have an understanding of Cisco Nexus Fabric system architecture. For more details, see the Deploy a VXLAN Network with an MP-BGP EVPN Control Plane white paper.
- You have an understanding of OpenStack. This document does not explain the functionality of Openstack and its components.

Architecture

The architecture and the different components of the Nexus Fabric Enabler solution is shown below:

Figure 1: Nexus Fabric Enabler architecture



Nexus Fabric Enabler is a solution to integrate OpenStack with the Cisco Nexus VXLAN BGP EVPN fabric using DCNM as the controller. In other words, it offers the Programmable (VXLAN BGP EVPN) Fabric or the Dynamic Fabric Automation (DFA) solution with OpenStack as the orchestrator. The solution has three key components:

1. Nexus Fabric Enabler server.
2. Nexus Fabric Enabler agent.
3. LLDP Agent Daemon (lldpad).

The components are described below. This solution is not a plugin that runs in the context of OpenStack. The Nexus Fabric Enabler solution listens to OpenStack notifications and acts on it to achieve the functionality. In other words, Nexus Fabric Enabler is loosely coupled with OpenStack.

Nexus Fabric Enabler Server

Nexus Fabric Enabler server functions are given below:

- The server runs as a service in the OpenStack Controller node. Both Systemd and Upstart scripts are provided to manage the service.
- In case of HA setups, the server runs on only one controller. In case of RHEL OSP 7 or 8, the enabler server can be managed by PaceMaker (using **pcs** CLIs), which can decide on the controller to launch the Fabric Enabler server.
- The server listens to event notifications from OpenStack's Keystone and Neutron services through RabbitMQ.
- The server communicates project and network events to DCNM. A project create event in OpenStack translates to an organization create in DCNM along with a partition create. Every project in OpenStack maps to an organization/partition in DCNM. Similarly, a network create in OpenStack translates to a network create in DCNM under the partition, which was created during project create.
- The server manages segmentation for tenant networks. Every network created in OpenStack is assigned a unique segment ID.
- The server communicates VM events to the appropriate Nexus Enabler agent. When a VM is launched through OpenStack, the Nexus Enabler server also gets the compute node that is going to host the VM. The VM information, along with the relevant project and network details, are communicated to the Nexus Enabler agent running in the compute node.
- Database management—The MySQL database is used for persistency so that the state is not lost between process restarts.
- Error handling—The Nexus enabler server has to ensure that DCNM and OpenStack remain in sync. If a project or network creation/deletion fails in DCNM temporarily, the enabler server has to retry the operation. Similarly, auto-configuration for a VM can also fail in the switch temporarily, in which case the enabler server needs to ensure that the enabler agent retries the operation.
- The server is completely restartable.

Nexus Fabric Enabler Agent

Fabric Enabler agent functions are given below:

- The Fabric Enabler agent runs as a service in all the OpenStack (controller and compute) nodes. Both Systemd and Upstart scripts are provided to manage the service.
- The agent performs uplink detection and programs OVS flows to pass through LLDP/VDP packets, through the OVS bridge. This is a feature by which the Fabric Enabler agent dynamically determines which interface of the compute node is connected to the ToR switch. This interface is referred as the *uplink* interface.
    - Unlike many deployment scenarios in OpenStack, it is not necessary to provide the uplink interface of the server during OpenStack installation. The interface information need not be consistent in all the servers. One compute server can have 'eth2' as the uplink interface and another server can have 'eth5' as the uplink interface.
    - This function works for port-channel (bond) uplink interfaces.
    - This works only when the uplink interface is not configured with an IP address.
    - When more than one interface is connected to the ToR switch, and if the interfaces are not a part of a port-channel (bond), then the uplink detection will pick the first available interface. For special

scenarios, such as this one, it is recommended to turn off uplink detection and provide the uplink interface for this compute node in the configuration file as explained in the next chapter.

- The agent communicates vNIC information of a VM along with Segment ID to VDP (running in the context of lldpad) after being notified by the Fabric Enabler server. This communication is done periodically. It is needed for failure cases because lldpad is stateless for vNICs for which auto-configuration has failed. So, the vNIC information needs to be periodically refreshed to lldpad. This is also needed during lldpad restarts, because lldpad does not store the vNIC information in a persistent memory.
- The agent retrieves all the VMs running on this compute node from the Fabric Enabler server at the beginning. The Fabric Enabler agents do not have a persistent DB (like MySQL). It relies on the server to maintain the information. This is similar to OpenStack's Neutron server and agent component.
- The agent programs the OVS flows based on the VLAN received from the switch through VDP.
- The agent is completely restartable.

LLDP Agent Daemon (lldpad)

Pointers about lldpad are given below:

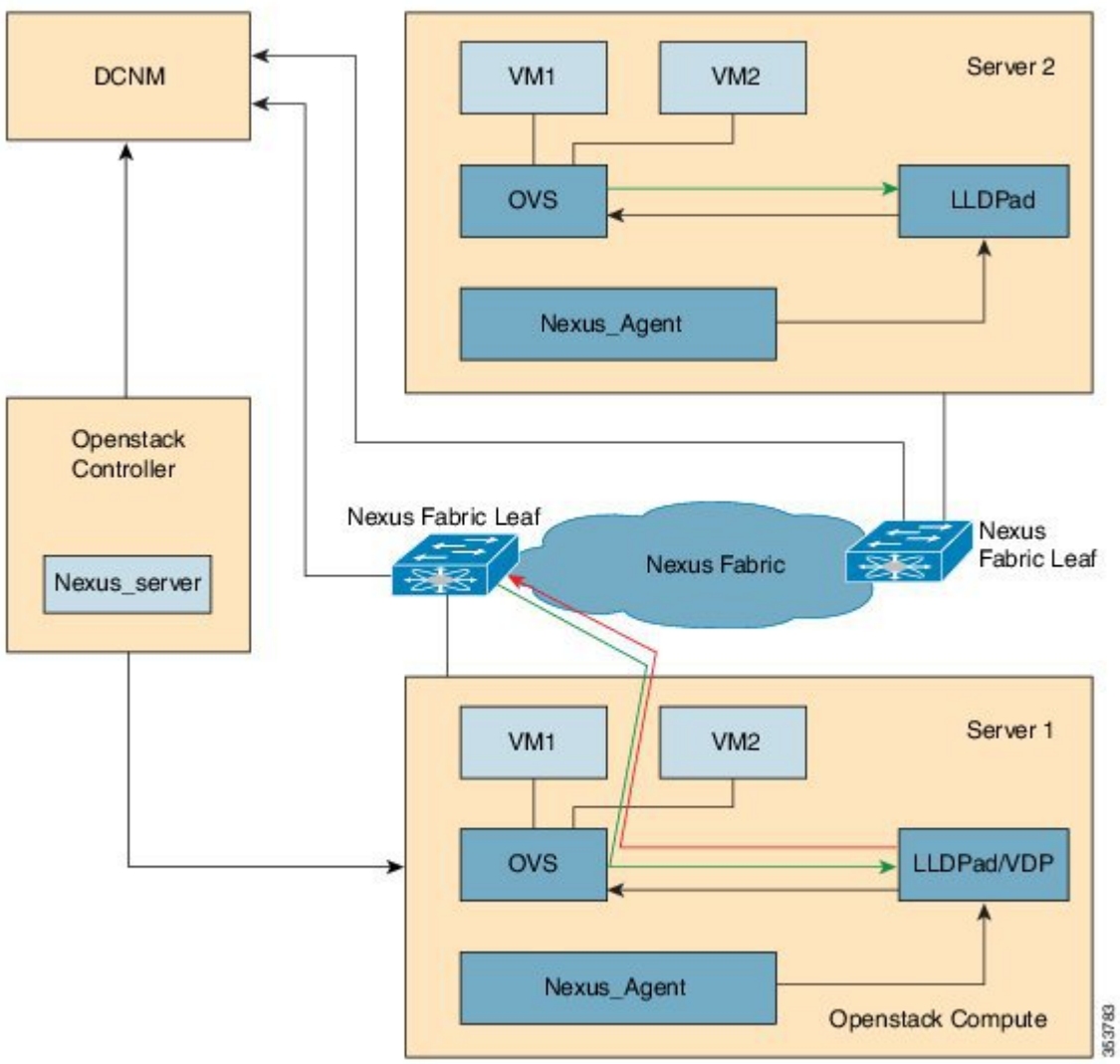
- It is an open source software supporting VDP. Refer to the IEEE 802.1QBG document for more details on VDP functionality.

Cisco Nexus Fabric Enabler System Flow

OpenStack serves as one of orchestrators of the cloud enabled through the Cisco Nexus fabric. For this release, all the orchestration is done using OpenStack's *Horizon* dashboard graphical user interface, CLIs, or OpenStack generic open APIs.

The following diagram provides a high level overview of the OpenStack orchestrator with the Cisco Nexus fabric as its network connecting all the compute nodes:

Figure 2: System Flow



The control flow can be summarized as follows:

1. OpenStack cloud administrator creates a tenant or project. OpenStack Keystone component notifies the Cisco Nexus Fabric Enabler server running in the control node about the creation of the tenant. The server sends the tenant information to Cisco Prime DCNM through the Cisco Prime DCNM REST API. The tenant's user name and password are created by the administrator too.
2. The tenant logs into OpenStack and creates the network. The OpenStack Neutron components notifies the Cisco Nexus Fabric Enabler server of the network Information (subnet/mask, tenant name), and the server automatically sends the network creation request to Cisco Prime DCNM through the Cisco Prime DCNM REST APIs.

3 A VM instance is launched, specifying the network that the instance will be part of.

4 The server sends the network information (subnet/mask, tenant name) and VM information to the Cisco Nexus Fabric Enabler agent running in the corresponding compute node.

5 VSI Discovery and configuration protocol (VDP), running in the context of lldpad, gets notified by the agent about the VM and the segment ID associated with the VM.

6 VDP, in turn, communicates with the leaf switch and sends the VM's information with the segment ID.

7 The leaf switch contacts Cisco Prime DCNM with the segment ID for retrieving the network attributes and auto-configure the compute node facing the interface for this tenant VRF.

8 The leaf switch responds with the VLAN information that is to be used for tagging the VM's traffic. The VLAN to be used will be the first available VLAN from the predefined VLAN pool (created using the **system fabric dynamic-vlans** command) on each individual leaf switch. The selected one is the leaf switch significance resource.

9 lldpad module notifies the Cisco Nexus Fabric Enabler agent with the VLAN information provided by the leaf switch.

10 The agent configures OVS such that the VM traffic to the network contains the VLAN information/tag provided by the leaf switch. The VM's vNIC is operational only at this point.

Note **Attention—**Refer the prerequisites for installing Nexus fabric OpenStack Enabler. It is noted in the *Before You Start* section of the *Hardware and Software Requirements* chapter.

Installation Overview

Apart from OpenStack installation, the Nexus Fabric Enabler also needs to be installed. As described previously, OpenStack can be installed in multiple ways. Apart from installation, the OpenStack configuration file needs to have certain settings for the solution to work. The changes in the configuration file are needed for the following:

- Configuring OpenStack networking to work in conjunction with the VXLAN BGP EVPN farbic (Programmable Fabric) or DFA. This is done when OpenStack is installed for the first time. This step is dependent on the distribution. For example, RHEL OSP 7 may have a different command to perform this operation than Mirantis.
- Enabling notification functionality in OpenStack, which can be received by the Nexus Fabric Enabler. This step is done by the installer provided by the Nexus Fabric Enabler. This step is mostly common across distributions.

The Nexus Fabric Enabler solution is officially qualified with RHEL OSP 7. The solution is also tested with DevStack, which is not intended for production. This chapter starts with a section on Red Hat OSP installation that provides pointers on *configuring OpenStack networking to work in conjunction with Programmable Fabric or DFA*. Then, the installation mechanism of Nexus Fabric Enabler is explained in detail.

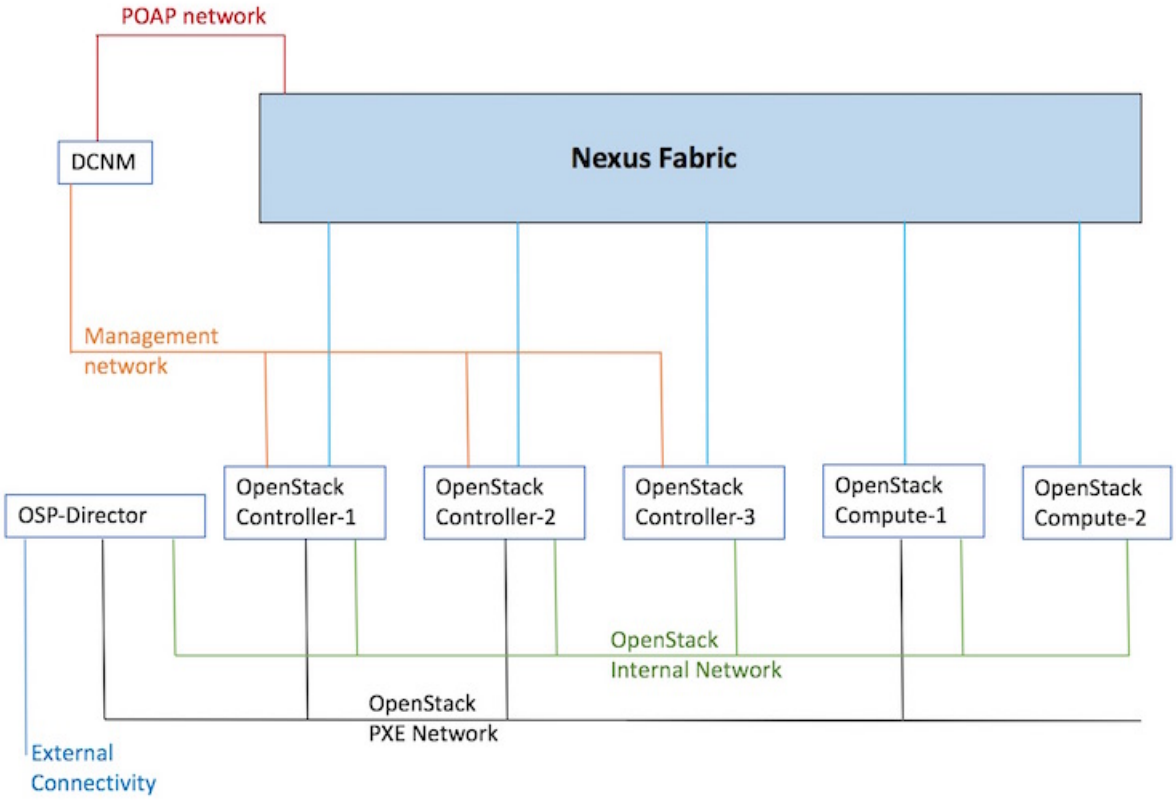
OpenStack Installation

Topology

Note **Important—**A sample topology is shown below. This is a critical step as it lays down the foundation for installation. It is highly recommended that you follow the same wiring scheme. This topology is based on RHEL OSP 7/8 with the Nexus fabric, Fabric Enabler, and DCNM. In the topology, the interface that is connected to the cluster should be operationally up (the corresponding command is **sudo ifconfig eth0 up**).

DCNM is also connected to the OpenStack control node through an IP network. Refer the OSP installation guide for more information on the OpenStack internal and PXE network.

Figure 3: Sample topology



RHEL OSP 7 Installation

Note Refer the official Red Hat documentation on how to install the Overcloud and the Undercloud.

Before deploying the Overcloud, implement the following steps to ensure compatibility with Nexus Fabric Enabler.

1. The Neutron type drivers must include *local*. Add these Neutron Type Drivers in *network-environment.yaml*, the network environment file—*local*, *flat*, *vlan*, *gre*, and *vxlan*.
2. Disable Neutron tunneling with the -- *neutron-disable-tunneling* option.
3. Set the Neutron Network Type to *local* with the -- *neutron-disable-tunneling* option.
4. Set the Neutron bridge mapping with the -- *neutron-bridge-mappings ethd:br-ethd* option.

An example for an Overcloud deployment command that is compatible with Nexus Fabric Enabler is given below:

```
openstack overcloud deploy --templates \

-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /home/stack/templates/network-environment.yaml \
-e /home/stack/templates/storage-environment.yaml \
--control-flavor control \
--compute-flavor compute \
--ntp-server clock.cisco.com \
--neutron-network-type local \
--neutron-disable-tunneling \
--neutron-bridge-mappings ethd:br-ethd \
--compute-scale 2 \
--verbose
```

HA for RHEL OSP 7

High availability (HA) provides continuous operation. The RHEL OSP7 director provides high availability to an OpenStack Platform environment through the controller node cluster. The director installs a set of the same components on each controller node and manages them as one whole service. Having a cluster provides a fallback in case of operational failures on a single controller node.

The following example shows how to install the Overcloud with redundant controllers:

```
openstack overcloud deploy --templates \

-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /home/stack/templates/network-environment.yaml \
-e /home/stack/templates/storage-environment.yaml \
--control-flavor control \
--compute-flavor compute \
--ntp-server clock.cisco.com \
--neutron-network-type local \
--neutron-disable-tunneling \
--neutron-bridge-mappings ethd:br-ethd \
--control-scale 3 \
--compute-scale 2 \
--verbose
```

Note For HA to work, you need a minimum of three controllers.

Note If compute nodes are connected to fabric leaf nodes through a port-channel/bond, make sure the Linux bond interfaces are used, since OVS bond interfaces will not work with Fabric Enabler. Create bond interfaces before installing and running the Openstack Fabric Enabler.

Verification of Configuration Files

This section is generally independent of the OpenStack installation method. But, check with the distribution as to where the configuration files are placed, and if the name of the configuration file has changed. As described earlier, the type_drivers should be set to local. *openvswitch* is used as the vswitch for the VXLAN BGP EVPN Programmable Fabric or DFA solution. Verify the following:

1. Ensure that *./etc/neutron/plugins/ml2/ml2_conf.ini* is configured as follows:

```
[ml2]
type_drivers = local mechanism_drivers = openvswitch

[ovs]
bridge_mappings = ethd:br-ethd

([ml2_type_flat], [ml2_type_vlan], [ml2_type_gre] and [ml2_type_vxlan] sections should
not be specified)
```

2. Ensure that *./etc/neutron/neutron.conf* is configured as follows:

```
core_plugin = neutron.plugins.ml2.plugin.Ml2Plugin
```

   Ensure that the tunnel type is not set. The *keystone_authtoken* should be similar to the following setting:

```
[keystone_authtoken]
signing_dir = /var/cache/neutron
auth_uri = http://<ip address of controller>:5000/v2.0
cafile = /opt/stack/data/ca-bundle.pem
identity_uri = http://<ip address of controller>:35357
auth_host = <ip address of controller>
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = neutron
admin_password = password
```

3. Ensure that *./etc/nova/ nova.conf* has the *keystone_authtoken* section(s) set similar to the following

```
[keystone_authtoken]
signing_dir = /var/cache/nova
admin_password = password
admin_user = nova
admin_tenant_name = service
auth_uri = http://<ip address of controller>:5000/v2.0
cafile = /opt/stack/data/ca-bundle.pem
identity_uri = http://<ip address of controller>:35357
auth_protocol = http
auth_port = 35357
auth_host = <ip address of controller>
```

Note Different OpenStack distributions may have the configuration file placed in a different directory. Or, the name of the configuration file can be different. Ensure that the appropriate configuration file has the above contents set correctly.

Cisco Nexus Fabric Enabler Installation

The sections below provides Cisco Nexus Fabric Enabler installation information.

Prerequisites

The following pre-requisite is applicable for RHEL OSP 7/8 based setups.

Note All the changes to Neutron and Nova files as well as extra rpm installation, Neutron DB patching for the firewall, disabling selinux, etc, that are described for a non-HA setup are still applicable.

- Cisco Nexus Fabric Enabler requires RabbitMQ to listen to requests made to the VIP address. You should configure this manually as other OpenStack components do not need them.

Installation in all the nodes (Fresh install)

1. Login to the OSP 7 Director Node for an RHEL OSP 7 or OSP 8 setup. In case of a non-production setup like DevStack, login to the controller node.
2. git clone -b rel_2_0_0 https://github.com/CiscoSystems/fabric_enabler ofe
3. cd ofe
4. Edit enabler_conf.ini as given in the next section.
5. The following command will install the Fabric Enabler on all the controller and compute nodes. This command will also install the Fabric Enabler in case of a HA setup. The two scenarios involving proxy requirement are given below:

   - If a proxy for reaching the Internet is not needed, use these commands:
     ```
     python setup_enabler.py  --vendor-os-release=rhel-osp7 --mysql-user=root
     --mysql-host=localhost
     ```
   - If a proxy for reaching the Internet is needed, use these commands:
     ```
     python setup_enabler.py  --vendor-os-release=rhel-osp7 --mysql-user=root
     --mysql-host=localhost --https-proxy=<proxy>
     ```
     <proxy> is the https proxy, such as *https://proxy.esl.cisco.com:80* from Cisco Labs.

   In case of HA setups for RHEL OSP 7 or OSP 8, the Enabler server will be started by Pacemaker. The Enabler server will be running in one of the servers. In case of a crash or the controller node going down, Pacemaker will ensure to restart the Enabler server in one of the available controllers. The Enabler agent and lldpad will be started in all the HA controller nodes and compute nodes.

Installation on a single node

Typically, this is needed when a new compute node is added after installing the Fabric Enabler on all the nodes:

1. Login to the OSP7 Director Node in case of RHEL OSP 7 or OSP 8 setups. In case of a non-production setup like DevStack, login to the controller node.
2. git clone -b rel_2_0_0 https://github.com/CiscoSystems/fabric_enabler ofe
3. cd ofe
4. Edit enabler_conf.ini as given in the next section.
5. Use the Nova list to get a list of controllers and compute node IP addresses.
6. For each compute node that is newly added, install the Fabric Enabler using these commands:
```
python setup_enabler.py --compute-name=<compute ip address> --remote-user=heat-admin
--https-proxy=<proxy>
```

Upgrading Fabric Enabler to the new version

The following are needed when you want to install a new version of the Fabric Enabler:

1. Ensure proxy variables are set properly.
2. Update the Fabric Enabler git repository using the following command:

```
git pull
```

Note If the git repository is not available, a new one needs to be cloned.

3. Ensure *enabler_conf.ini* is up to date. If needed, copy it from a controller.
4. If a proxy for reaching the internet is not needed, use these commands:

```
python setup_enabler.py  --vendor-os-release=rhel-osp7 --upgrade=True
```

5. If a proxy for reaching the internet is needed, use these commands::

```
python setup_enabler.py  --vendor-os-release=rhel-osp7 --https-proxy=<proxy> --upgrade=True
```

   a. More upgrade options/pointers are given below:

      - The upgrade process will upgrade and restart the Cisco Nexus Fabric Enabler server and agent.
      - At times, lldpad also needs to be restarted. If so, add *--restart-lldpad=True* to the above commands.
      - It is possible to update a single controller or compute note by using the *--control-name* and *--compute-name* options, respectively.

enabler_conf.ini

The following table describes the sections and fields in the *enabler_conf.in* file.

Note Unless specified as *(Optional)* in the Field Name column, the field is mandatory.

Table 1: Section—[General]

Field Name	Description	Default Value	Fabric Type
compute_user (Optional)	Compute node user name that can be used with the **ssh** command for remote logins. Also, the user must be a sudoer assuming all compute nodes have the same	Not Applicable	VXLAN BGP EVPN (Programmable Fabric) DFA
compute_passwd (Optional)	Compute node password that can be used with the **ssh** command for remote logins.	Not Applicable	VXLAN BGP EVPN DFA
node (Optional)	A *comma* separated list of hosts for which a static uplink is configured. The node name should be a fully qualified domain name (such as **host1.example.com**)	Not Applicable	VXLAN BGP EVPN DFA
node_uplink (Optional)	A comma separated list of uplink ports on the server connected to the leaf switch. This parameter and the *node* parameter are mandatory if a static uplink is desired.	Not Applicable	VXLAN BGP EVPN DFA

ucs_fi_evb_dmac (Optional)	If OpenStack is running in any UCS FI blade server, enter the EVB DMAC address that is configured in the fabric. The Fabric Enabler software running in the node will detect if it is a UCS FI blade server, but the interface connected to the switch has be included in the 'node_uplink' configuration, along with the node.	01:80:c2:12:34:56	VXLAN BGP EVPN DFA

This is the section about Cisco DCNM, which you have installed separately and set the right access credentials to be used by the OpenStack Cisco Nexus Fabric Enabler. Ensure that the *gateway_mac* value matches your POAP template setting in Cisco DCNM for your leaf switch, and you use the right range of segment IDs administrated by your Fabric Manager.

Table 2: Section—[dcnm]

Field Name	Description	Default Value	Fabric Type
dcnm_ip	IP address of the DCNM. It should be reachable from the OpenStack controller node.	Not Applicable	VXLAN BGP EVPN DFA
dcnm_user	DCNM server login credentials.	Not Applicable	VXLAN BGP EVPN DFA
dcnm_amqp_user	DCNM server RabbitMQ messaging credentials.	Not Applicable	VXLAN BGP EVPN DFA
dcnm_password	DCNM server password.	Not Applicable	VXLAN BGP EVPN DFA
gateway_mac (Optional)	Gateway MAC address. This should be the same as the MAC address configured on the leaf switch nodes.	20:20:00:00:00:AA	VXLAN BGP EVPN DFA

orchestrator_id (Optional)	Orchestartor ID used for registering the segment ID range on DCNM. If there are multiple setups using the same DCNM, ensure different orchestrator IDs are used.	Openstack Controller	VXLAN BGP EVPN DFA
segmentation_id_min	The minimum Segment ID value. It is a 24 bit integer value.	4097	VXLAN BGP EVPN DFA
segmentation_id_max	The maximum Segment ID value. It is a 24 bit integer value.	16777216	VXLAN BGP EVPN DFA
segmentation_reuse_timeout (Optional)	Duration after which an *available* segment ID can be reused. Once a segment ID is released, it will only be reused after 1 hour. If this functionality is not needed, enter a value of 0. Alternatively, to change the default value of 1 hour, uncomment the below and enter a different number (as an integer value).	1 hour	VXLAN BGP EVPN DFA
dcnm_net_ext (Optional)	The suffix of a network name when it is created by DCNM. This is usable for a scenario when network creation is done in DCNM and the Fabric Enabler populates this in OpenStack.	(DCNM)	VXLAN BGP EVPN DFA
dcnm_dhcp_leases (Optional)	The lease file name of the DHCP server on the DCNM.	/var/lib/dhcpd/dhcpd.leases	VXLAN BGP EVPN DFA
default_cfg_profile (Optional)	Default configuration profile when creating a network in DCNM.	defaultNetworkEvpnProfile defaultNetworkUniversalTfProfile	VXLAN BGP EVPN DFA
default_vrf_profile (Optional)	Default VRF profile name for a partition in DCNM.	vrf-common-evpn vrf-common-default	VXLAN BGP EVPN DFA

Table 3: Section—[dfa_rpc]

Field Name	Description	Default Value	Fabric Type
transport_url	Transport URL parameter for RPC.	Not Applicable An example is given below: `transport_url='rabbit://username:password@%(ip)s:5672//'` The *ip* address is of the controller when there is only one controller. In a HA environment with multiple controllers in a cluster, the IP address is the virtual IP address (VIP) of the controller cluster. You should replace the *username* and *password* based on your setting. These credentials should be the same as the one you used to configure RabbitMQ, by default available in the location */etc/rabbitmq/rabbitmq.config*.	VXLAN BGP EVPN DFA

Table 4: Section—[dfa_mysql]

Field Name	Description	Default Value	Fabric Type
connection	MYSQL DB connection option	Not Applicable An example is given below: `connection=mysql://username:password@localhost/cisco_dfa?charset=utf8` The *localhost* is applicable if there is only one controller. In a HA environment with multiple controllers in a cluster, a localhost will be replaced with the *VIP* of the controller cluster. You should replace the *username* and *password* based on your setting.	VXLAN BGP EVPN DFA

Table 5: Section—[dfa_notify]

Field Name	Description	Default Value	Fabric Type

cisco_dfa_notify_queue (Optional)	Notification queue name for DFA enabler. service_name: keystone and neutron.	cisco_dfa_%(service_name)s_notify	VXLAN BGP EVPN DFA

Table 6: Section—[dfa_log]

Field Name	Description	Default Value	Fabric Type
log_file (Optional)	Log file name. DEPRECATED (use Log file prefix instead). If log file name and directory is not specified, the default is the standard output.	fabric_enabler.log	VXLAN BGP EVPN DFA
log_file_prefix (Optional)	The prefix will be used by Fabric Enabler processes to create log files. If the default prefix of fabric_enabler is used, the Fabric Enabler server's log files will be fabric_enabler_server.log and the Fabric Enabler agent's log file will be fabric_enabler_agent.log	fabric_enabler	VXLAN BGP EVPN DFA
log_dir (Optional)	The directory name for the log file.	Current directory	VXLAN BGP EVPN DFA
log_level (Optional)	Enabler debugging output level. Set to DEBUG to see the debbugging output	WARNING	VXLAN BGP EVPN DFA

Table 7: Section—[dfa_agent]

Field Name	Description	Default Value	Fabric Type
integration_bridge (Optional)	OVS Neutron Agent related configuration. Ensure that this is the same as what is configured for the OVS Neutron Agent.	br-int	VXLAN BGP EVPN DFA

external_dfa_bridge (Optional)	OVS Neutron Agent related configuration. Ensure that this is the same as what is configured for the OVS Neutron Agent.	br-ethd	VXLAN BGP EVPN DFA

Table 8: Section—[vdp]

Field Name	Description	Default Value	Fabric Type
mgrid2 (Optional)	Refer to IEEE 801.1QBG standard documentation.	0	VXLAN BGP EVPN DFA
typeid (Optional)	Refer to IEEE 801.1QBG standard documentation.	0	VXLAN BGP EVPN DFA
typeidver (Optional)	Refer to IEEE 801.1QBG standard documentation.	0	VXLAN BGP EVPN DFA
vsiidfrmt (Optional)	Refer to IEEE 801.1QBG standard documentation.	5	VXLAN BGP EVPN DFA
hints (Optional)	Refer to IEEE 801.1QBG standard documentation.	Not Applicable	VXLAN BGP EVPN DFA
filter (Optional)	Refer to IEEE 801.1QBG standard documentation.	4	VXLAN BGP EVPN DFA
vdp_sync_timeout (Optional)	Query to lldpad for every VSI. If a VSI is not present in lldpad, an associate request is sent to lldpad.	15	VXLAN BGP EVPN DFA

Note

- Ensure that the segment ID does not overlap with other segment ID ranges (for example, the Cisco DCNM segment ID uses the default 30,000 to 49,999).
- It requires all control and compute nodes that have the same username and password, and this is your Linux account on the servers, to run as control/compute nodes.

Post Installation

Verify the Cisco Nexus Fabric Enabler server, Cisco Nexus Fabric Enabler agent, lldpad, and the existence of notification queues, as shown below:

Cisco Nexus Fabric Enabler server verification

On an RHEL OSP HA setup, a sample output is shown below on a controller:

```
[heat-admin@overcloud-controller-0 ~]$ sudo pcs resource | grep fabric-enabler-server
 fabric-enabler-server  (systemd:fabric-enabler-server):       Started overcloud-controller-1
```

The command displays the controller node where the Fabric Enabler server is running. Alternatively, the following commands can be used on the controller where the Fabric Enabler server is running:

- sudo systemctl status fabric-enabler-server [For a Redhat/CentOs based controller]
- sudo status fabric-enabler-server [For a Ubuntu based one]
- ps –ef | grep fabric-enabler-server [Any setup]

Cisco Nexus Fabric Enabler agent verification

The Enabler agent runs on all controller and compute nodes. Run the below commands on a node where verification is needed. This sample is specific to Red Hat based setups that have system based startup scripts.

```
[heat-admin@overcloud-controller-0 ~]$ sudo systemctl status fabric-enabler-agent
â fabric-enabler-agent.service - Cluster Controlled fabric-enabler-agent
   Loaded: loaded (/usr/lib/systemd/system/fabric-enabler-agent.service; enabled; vendor
preset: disabled)
  Drop-In: /run/systemd/system/fabric-enabler-agent.service.d
           ââ50-pacemaker.conf
   Active: active (running) since Fri 2016-09-23 01:31:01 EDT; 1 weeks 0 days ago
```

Alternatively, the following commands can be used:

- sudo status fabric-enabler-agent [On Ubuntu based servers]
- ps –ef | grep fabric-enabler-agent [Any setup]

lldpad verification

lldpad runs on all controller and compute nodes. Run the below commands on the compute node where verification is needed. The sample is specific to Red Hat based setups that have system based startup scripts.

```
[heat-admin@overcloud-controller-0 ~]$ sudo systemctl status lldpad
â lldpad.service - Cluster Controlled lldpad
   Loaded: loaded (/usr/lib/systemd/system/lldpad.service; enabled; vendor preset: disabled)

  Drop-In: /run/systemd/system/lldpad.service.d
           ââ50-pacemaker.conf
   Active: active (running) since Sun 2016-08-07 22:05:27 EDT; 1 months 22 days ago
 Main PID: 6041 (lldpad)
   CGroup: /system.slice/lldpad.service
           ââ6041 /usr/sbin/lldpad –t
```

Alternatively, the following commands can be used:

- sudo status lldpad [On Ubuntu based servers]

- ps –ef | grep lldpad [Any setups]

Verify existence of notification queues

```
sudo rabbitmqctl list_queues | grep cisco cisco_dfa_keystone_notify.info 0
cisco_dfa_neutron_notify.info 0
```

Create Project and Launch VM

The information provided in this section is generic to OpenStack and it is provided here for your convenience with the exception of ConfigProfile, which is Cisco Nexus fabric specific.

Steps to Create a Project

Follow these steps to create a project:

1. Login to the Horizon dashboard as an administrator. Use the password that you used in the OpenStack configuration file.
2. Click **Projects** and then **Create Project**.
3. Enter relevant project information and click **Create Project** to create the project.

Note The project name is used as vrfName in the fabric (vrfName = “project_name:CTX”) for fabric auto-configuration. The fabric limits the size of the vrfName string to 32 characters. Ensure that the project name length is less than 29 characters when creating the project. Do not use hyphens in the project name.

DCI Support

You can use OpenStack to configure the DC Inter-connect function. Support is only provided for Layer-3 DCI with the Cisco Prime DCNM 7.1(1) release, and Cisco NX-OS 7.1(0)N1(1) release or later.

As part of the project name string, type *xyx:dci_id:129* to enable DCI support ('129' is used here as an example). Type *xyz* or *xyz:dci_id:0* to remove DCI support for this project.

The integer 129 is the DCI ID. Cisco Prime DCNM uses it as an indication that the user desires to auto-configure the border leaf switches with this VRF, and extend to the DCI edge devices(s). If the value is *0*, Cisco Prime DCNM removes VRF configurations from the border leaf switches and the configurations that extend the VRF from the border leaf switch to the DC edge device(s).

Steps to Create a User for the Project

Follow these steps to create a user for the project:

1. Click **Users**, and then **Create User**.

2 Fill in all the fields, select the project you just created and select the role as *admin*. The network information will not be populated correctly to DCNM if you fail to specify the role as *admin*.

Steps to Create the Network

Follow these steps to create the network:

1 Login as a user using login credentials created by the administrator.

2 Click the **Project** tab.

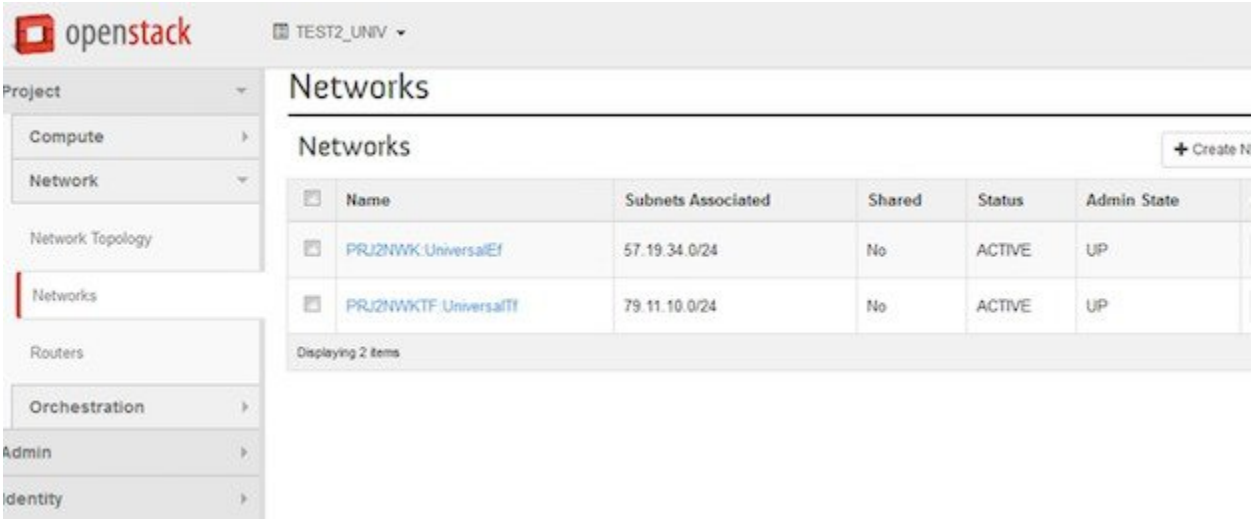
3 Click **Networks** and then click **Create Network**. Specify a Name for the network and go to the **subnet** tab. This is mandatory.

4 Specify a Network Address for the subnet.

Use non-default network profiles

By default, for Cisco Prime DCNM with version 7.1, defaultNetworkUniversalEfProfile is the network profile used automatically by the system. Additionally, defaultNetworkUniversalTfProfile can also be specified when creating a network in OpenStack. A sample screen shot is given below:

Figure 4: Default Network



Following are the supported network profiles with Cisco Prime DCNM version 7.1(1):

- defaultNetworkUniversalEfProfile
- defaultNetworkUniversalTfProfile
- defaultNetworkL2Profile

If it is an upgrade from version 7.0(1) or 7.0(2) to 7.1(1), the default profile will be defaultNetworkIpv4EfProfile, and the supported profiles will be the sum of the profiles for versions 7.0(1), 7.0(2) and 7.1(1) or later, as shown below:

- defaultNetworkIpv4EfProfile
- defaultNetworkIpv4TfProfile
- defaultNetworkL2Profile
- defaultNetworkUniversalEfProfile
- defaultNetworkUniversalTfProfile
- defaultNetworkL2Profile

The syntax to use non-default profiles when creating a network is given below. In the examples, network_name signifies the name of the network followed by a sub-string of the profile name:

- network_name:L2
- network_name: Ipv4Ef
- network_name: Ipv4Tf
- network_name: UniversalTf
- network_name: UniversalEf

Use defaultNetworkL2Profile

If this profile is chosen when a network is created in OpenStack, DCNM DHCP server will not assign an IP address for the VM associated with the network. Users are required to configure a static IP address for the VM. Additionally, the following command needs to be run on the OpenStack control node:

```
$fabric_enabler_cli
Cisco Nexus Fabric Command Line Interface
(Nexus-Fabric) set_static_ip --mac fa:16:3e:72:ab:dc --ip 136.10.0.16
```

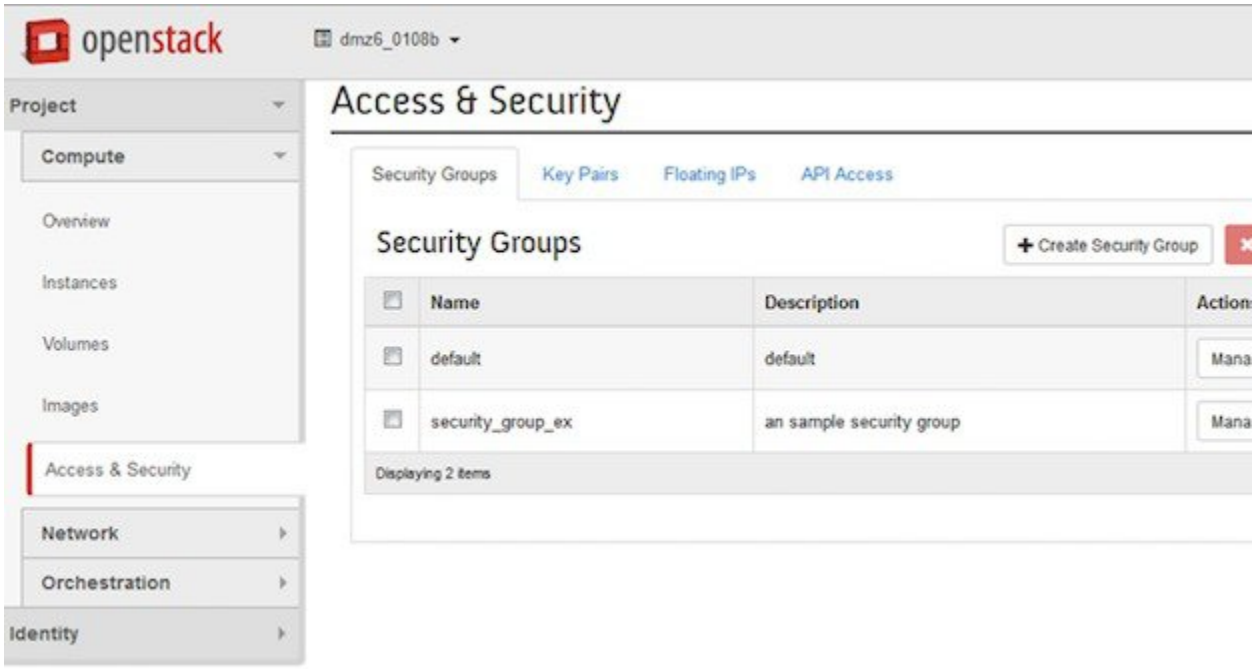
The MAC address is the VM's vNIC and the IP address is the statically configured VM IP address. When a VM is removed from OpenStack, the above entry is automatically removed by the system.

Steps to Create a Security Group

You need to create and add a security group with appropriate rules before launching the VM. Create a security group and security rules that allow DHCP (from DCNM) and your data traffic to go through. After logging into Horizon as a user, click **Project** > **Compute** > **Access Security**. Use the **Create Security Group** tab to

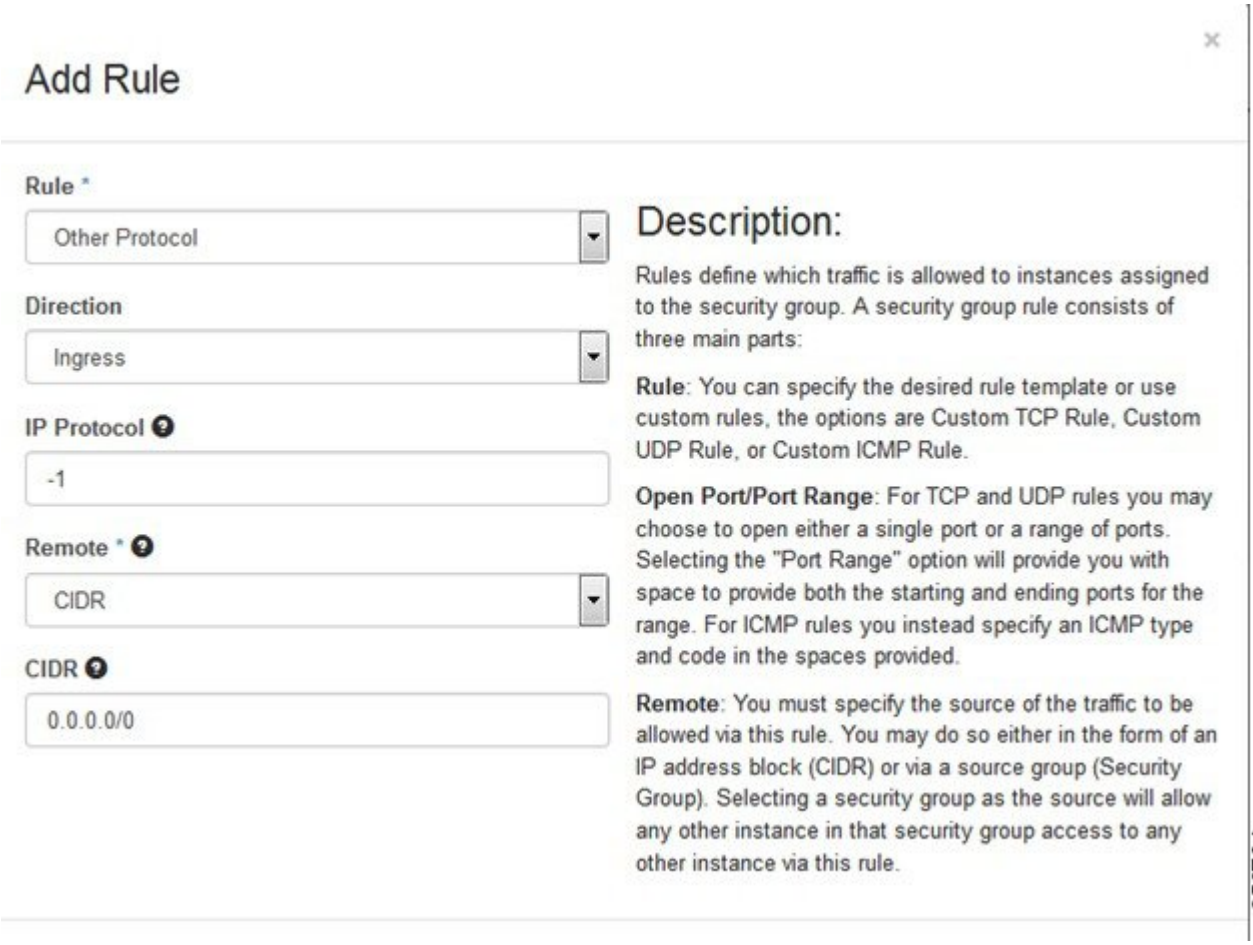
create a security group. After a security group is created, it appears in the **Security Groups** tab. A newly added group *security_group_ex* is displayed in the following sample screen shot.

Figure 5: Access and Security



Click **Manage Rules** for the security group you just created and add new rules. For example, if the following rule displayed in the *Add Rule* screen shot is added for the security group, it will allow all traffic.

Figure 6: Add Rule



Steps to Launch the VM

Follow these steps to launch the VM:

1. Click **Instances** and then click **Launch Instance**.
2. Click the **Image** drop-down menu and select the image.

   By default, the *CirrOS* image is selected.
3. Specify a name for the Instance.
4. Select the **Security** tab and choose the security group created (it is recommended to uncheck the default rule and select the one you specified).
5. Click the **Networking** tab and select the network from the **Available network** list.

6 Click **Launch**.

Limitations and Caveats

Known limitations and caveats are given below:

- Null Field Exception does not support IPv6.
- The VMs' internal vNIC interface state (*up* or *down*) is not detected by VDP. (Use VM creation or deletion as a potential workaround.)
- Servers with Converged Network Adapters (CNA) are not supported with OpenStack. The user needs to use a NIC.
- OpenStack DHCP is supported, while Cisco DCNM DHCP is not supported.

Technical Support

Following is the technical support model for using OpenStack:

- OpenStack is an on open source software platform, and is generally supported through its community using a best effort approach. If you use a third-party OpenStack installer and their support model, all generic OpenStack related support should come from the third-party support license.
- Support for the OpenStack Cisco Nexus Fabric Enabler part is provided from the Cisco Nexus Solution Team.