Cisco OpenStack Integration for Nexus Dashboard 4.1 – Installation Guide

Contents

Introduction

OpenStack provides a flexible software architecture designed to build cloud-computing environments. Its reference software implementation supports multiple Layer 2 transport options such as VLAN, GRE, and VXLAN. Additionally, the Neutron project within OpenStack offers software-based Layer 3 forwarding capabilities.

Purpose of this Guide

This guide provides step-by-step instructions for integrating OpenStack with the Nexus Dashboard Fabric Controller (NDFC), a component of Nexus Dashboard (ND). It covers the overall architecture, installation procedures, and configuration steps to ensure seamless integration with OpenStack's Neutron microservice.

Target Audience

This guide is intended for network engineers, cloud administrators, and IT professionals familiar with ND, NDFC, and OpenStack.

Overview

The integration enables OpenStack workflows to configure fabric networking through Nexus Dashboard. It uses the NDFC Modular Layer 2 (ML2) Mechanism Driver and Topology Agents to provide seamless connectivity between virtualized compute instances and network switches.

Figure 1 is a system diagram of integration. Cloud users interact with OpenStack, managing resources such as networks, subnets, routers, and VMs, which results in REST requests to Nexus Dashboard and consequently programs the fabric switches accordingly.

This solution requires deploying Cisco topology agents in each OpenStack compute node. The topology agents use LLDP discovery to create host-to-fabric mappings and transmit this information to OpenStack over the message bus using remote procedure calls (RPCs). OpenStack then uses these mappings to determine which switches and switch ports to configure when cloud users create virtual networks.

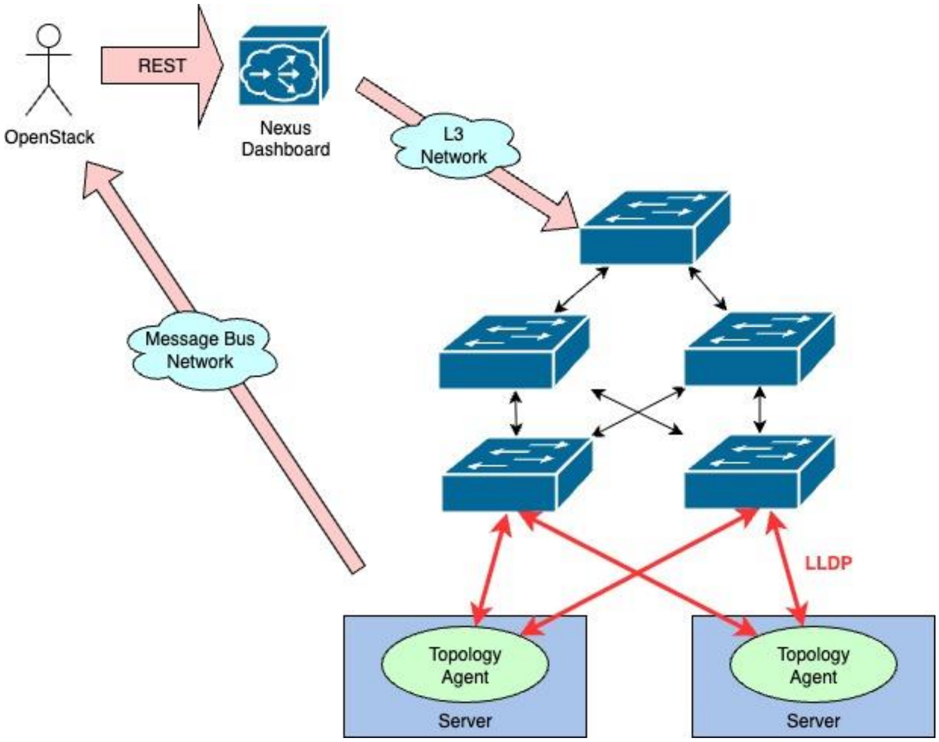

Figure 1.
OpenStack with NDFC System Diagram.

A key component of the integration involves mapping OpenStack resources to Nexus Dashboard resources. When cloud users or administrators create, update, or delete specific Neutron resources, the corresponding resources are automatically created, updated, or deleted in Nexus Dashboard and, as a result, in the fabric switches. Table 1 lists the mappings used by this integration.

Table 1. OpenStack to Nexus Dashboard Resource Mappings.

OpenStack Resource	Nexus Dashboard Resource
Project	VRF
Network	Network
Subnet	Subnet gateway of the network
Port	Network

Benefits of Using Nexus Dashboard with OpenStack

- The Layer 3 gateway for Neutron resources is configured at the switch level, enabling hardware-accelerated Layer 2 and Layer 3 functions. This setup enhances scalability and performance, especially in large-scale deployments.
- Integration of BGP-EVPN fabric with VXLAN.
- Automated network topology discovery with LLDP.

- Simplified and automated fabric network management tailored for OpenStack deployments.
- Easy management of Layer 2 and Layer 3 networks.
- Support for legacy Open vSwitch (OVS) as well as Open Virtual Networking (OVN) SDN frameworks.

Limitations of Using Nexus Dashboard with OpenStack

The following are limitations of this integration:

- Only VLAN type networks are supported.
- Must use “provider networks” workflow. The provider networks workflow requires cloud administrators to pre-provision networks in OpenStack and make them available to cloud users. They are referred to as provider networks because the cloud admin uses the provider networks extensions in OpenStack to fully specify the parameters of the network: encapsulation type (VLAN), segmentation ID (VLAN ID), and physical network. These extensions are only available to administrative users. Non-administrative users can then attach VMs to these networks. This workflow usually implies that Layer 3 forwarding is performed outside of OpenStack orchestration.
- OpenStack routers must not be attached to networks. Since routers represent Layer 3 forwarding in OpenStack, attaching them would require OpenStack to orchestrate forwarding, which conflicts with the hardware-accelerated Layer 3 forwarding in the fabric. Overlay networking solutions like Open Virtual Networking (OVN) perform software-based distributed Layer 3 forwarding, which bypasses the fabric’s hardware acceleration. Using the provider networks workflow delegates Layer 3 forwarding outside OpenStack, enabling cloud users to benefit from hardware-accelerated Layer 3 data planes.
- Only IPv4 subnets are supported.

Installation

This section explains how to install the Cisco OpenStack Integration for Nexus Dashboard. Installation is performed using a pip-based method, as an installer is not provided. Depending on your deployment’s installer, the installation steps may vary. Note that this integration does not support migrating existing OpenStack clouds using other network virtualization technologies. Deployment must be either a greenfield installation or an upgrade of an existing environment that already uses this integration.

System Requirements

- Nexus Dashboard version 4.1 or later.
- Supported Cisco Nexus N9K switches.

Nexus Dashboard Cluster Requirements

- Data Center VXLAN EVPN fabric.
- Ingress replication enabled.
- Fabric pre-provisioned with appropriate VRF and VLAN configurations.

OpenStack Node Requirements

- Compute and Controller nodes with networking-cisco package installed.
- Compute nodes must have network interface that supports LLDP.

- Compute nodes should be directly connected to NX-OS leaf switches or ToR switches.
- OpenStack Caracal (2024.1) release installed on all OpenStack nodes.

Network Requirements

- IPv4 addressing (initial integration does not support IPv6).
- VXLAN overlay with VLANs between hypervisors and switches.

  OpenStack external networking configured manually for tenant connectivity.

Software Requirements

- Nexus Dashboard version 4.1.
- Supported OpenStack version (For more information, see Cisco OpenStack Integration for Nexus Dashboard 4.1 - Release Notes to verify compatibility).

  Neutron ML2 Plugin Mechanism Driver for NDFC. (For more information, see Cisco OpenStack Integration for Nexus Dashboard 4.1 - Release Notes for version number).

Creating and Installing OpenStack Packages for Nexus Dashboard

The `networking-cisco` repository must be installed on all compute and controller hosts. For pip-based installations, you must create a source distribution. The steps below outline how to create this source distribution. In the following examples, the version tag 8.1.0 is used as a reference. For more information, see Cisco OpenStack Integration for Nexus Dashboard 4.1 - Release Notes to get the correct tag.

1. Clone the `networking-cisco` repository and change directory to networking-cisco:

```
$ git clone https://github.com/noironetworks/networking-cisco
$ cd networking-cisco
```

2. Find the `networking-cisco` tag that matches your OpenStack release:

```
$ git tag -n
$ git checkout 8.1.0
```

3. Create a tarball of the release:

```
$ sudo python3 setup.py sdist
```

4. Transfer the tarball to the controller and compute hosts and install it:

```
$ tar -xzf networking-cisco-8.1.0.tar.gz
$ sudo pip3 install networking-cisco-8.1.0
```

Configuring Nexus Dashboard for OpenStack Integration

After installing the `networking-cisco` source package, each host must be configured for operation. Controller hosts should be set up to use the NDFC ML2 mechanism driver, while compute hosts need to be configured to run the Topology Agent.

Step 1: Configure Neutron on the Controller Node

1. Update the file: /etc/neutron/plugins/ml2/ml2_conf.ini:

```
mechanism_drivers = ndfc,ovn,logger
 [ndfc]
enable_keystone_notification_purge = True
keystone_notification_exchange = keystone
```

```
keystone_notification_topic = notifications
ndfc_ip = <Nexus Dashboard REST/UIIP>
user = <user>
pwd = <password>
fabric_name = <fabric_name>
Switch_sync_interval = 1800
```

2. Verify OpenStack VLAN allocations in the same file:

   Neutron enables you to define VLAN allocation ranges for each physical network (physnet). Cloud administrators should verify that these ranges do not overlap with any VLAN ranges specified in templates used during fabric provisioning in Nexus Dashboard, such as VRF templates. Below is an example of this configuration in OpenStack, under the “ml2_type_vlan” section:

```
[ml2_type_vlan]
network_vlan_ranges=physnet1:1701:1750
```

3. Run the `networking-cisco` database migrations:

```
$ neutron-db-manage --subproject networking-cisco upgrade head
```

4. Restart the neutron-server service:

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

Step 2: Configure Keystone

When you create, update, or delete projects in OpenStack, you must notify Neutron so it can perform the corresponding VRF operations in Nexus Dashboard.

Follow these steps to ensure that Neutron receives updates from the Keystone microservice.

1. Update the file: /etc/keystone/keystone.conf file**:**

```
[oslo_messaging_notifications]
driver = messagingv2
transport_url = <transport_url>
For example:
transport_url=rabbit://guest:DK5xlGtNbT511xNQvaJQSioFp@overcloud-controller-
0.internalapi.localdomain:5672/?ssl=0
```

2. Restart the Keystone service:

```
$ sudo systemctl restart apache2
```

Step 3: Configure and Start Topology Agent on Compute Nodes

1. Install the `networking-cisco` package on all compute nodes.

2. Create the file: /etc/neutron/plugins/ml2/lldp.ini

```
[lldp_topology_agent]
```

   topology_handlers = networking_cisco.agent.nxos_topology.NxosTopologyHandler

3. Create the file with content: /etc/neutron/rootwrap.d/cisco-apic.filters

```
# neutron-rootwrap command filters for nodes on which
# neutron is expected to control network
#
```

```
# This file should be owned by (and only-writeable
# by) the root user format seems to be
      # cmd-name: filter-name, raw-command, user, args
[Filters]
# cisco-apic filters
lldpctl: CommandFilter, lldpctl, root
```

4. Edit the file: /etc/neutron/neutron.conf file:
    - Add the hostname under the [DEFAULT] section:

```
host = <compute_node_hostname>
```

- Add the transport_url.

```
transport_url = <transport_url>
```

5. Install and start the LLDP agent:

```
$ sudo /usr/local/bin/neutron-cisco-topology-agent \
        --config-file /etc/neutron/neutron.conf \
        --config-file /etc/neutron/plugins/ml2/lldp.ini \
  --config-file /etc/neutron/plugins/ml2/ml2_conf.ini
```

Verification and Testing

1. Verify the installation of the networking-cisco package:

```
$ pip show networking-cisco
```

2. Confirm the correct configuration of Neutron ML2 Mechanism Driver by creating and attaching networks in OpenStack.

3. Verify the host name of the compute by running below command on the compute node:

```
$ hostname
```

4. Verify that the `transport_url` in the `keystone.conf` file in keystone and the `neutron.conf` files in the compute nodes is same as the `transport_url` in the `neutron.conf` file in `neutron_ap.`

5. **Create OpenStack Resources:** To thoroughly verify the integration, create the following resources in OpenStack:

    Test VRF creation:

    - Create a new project in OpenStack:

```
openstack project create <project-name>
```

- Verify that the corresponding VRF is created in the Nexus Dashboard.

Test network creation:

- Create a new network in OpenStack:

```
openstack network create <network-name>  --project <project-name> --provider-network-type
vlan
```

- Verify the corresponding network is created in the Nexus Dashboard.
- Create subnet in OpenStack:

```
openstack subnet create <subnet-name> --project <project-name> --network <network-name>--
subnet-range 10.10.10.0/24
```

- Verify the subnet gateway IP is added to the corresponding network in Nexus Dashboard.

Test network attachment:

- Launch OpenStack instances on the network.
- Verify that that the network is attached to corresponding Leafs/Tors in Nexus Dashboard.

6. Test connectivity between OpenStack instances and verify configurations in Nexus Dashboard.
7. **DB tool for Listing Records**: These commands allow you to verify what host links and ToRs are currently registered in the database. This is useful for confirming that the topology agents are correctly discovering and reporting information, and for general monitoring.

List NXOS Host Links:

```
$ neutron-cisco-db-tool \
--config-file /etc/neutron/neutron.conf \
list-nxos-links
```

List NXOS ToRs:

```
$ neutron-cisco-db-tool \
--config-file /etc/neutron/neutron.conf \
list-nxos-tors
```

8. **DB tool for Deleting Records**: The database deletion tool commands are essential for maintenance and troubleshooting. For example, when a compute node or a ToR switch is decommissioned or replaced, its related database entries can become outdated or " stale." Executing these delete commands removes such obsolete records, keeping the database up-to-date with the current network topology. This cleanup helps avoid problems caused by the system trying to communicate with components that no longer exist.

Delete NXOS Host Links:

```
$ neutron-cisco-db-tool \
--config-file /etc/neutron/neutron.conf \
delete-nxos-links --condition "host_name='host1'”
```

Delete NXOS ToRs:

```
$ neutron-cisco-db-tool \
--config-file /etc/neutron/neutron.conf \
delete-nxos-tors \
--condition "tor_serial_number='1234'"
```

9. After running the database tool commands, verify that the expected records are listed or deleted.

Troubleshooting

- **Issue: Database migration failure**
  - Check the neutron-db-manage logs for errors.
  - Verify connectivity between the OpenStack controller and the database.

- **Issue: Keystone notification errors**
  - Ensure the transport_url matches the value in neutron.conf.
- **Issue: LLDP agent not starting**
  - Verify the lldp.ini configuration file for errors.
  - Check that the lldpd package is installed.
- **Issue: Stale entries in topology databases**
  - List database entries using the database tool list commands.
  - Delete the stale entries using the database tool delete commands.

Upgrading

1. Backup current OpenStack neutron configuration files and database.
2. Follow the same steps as installation using the latest version of the networking-cisco package.
3. Test the upgraded setup for connectivity and functionality.

Glossary

ND	Nexus Dashboard
ML2	Modular Layer 2
OVN	Open Virtual Networking
VXLAN	Virtual Extensible LAN

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **https://www.cisco.com/go/offices**.

Printed in USA Cxx-xxxxxx-xx 01/22