



# Cisco Nexus Fabric Enabler Overview

---

- [Assumptions, page 1](#)
- [Architecture, page 2](#)
- [Cisco Nexus Fabric Enabler System Flow, page 4](#)

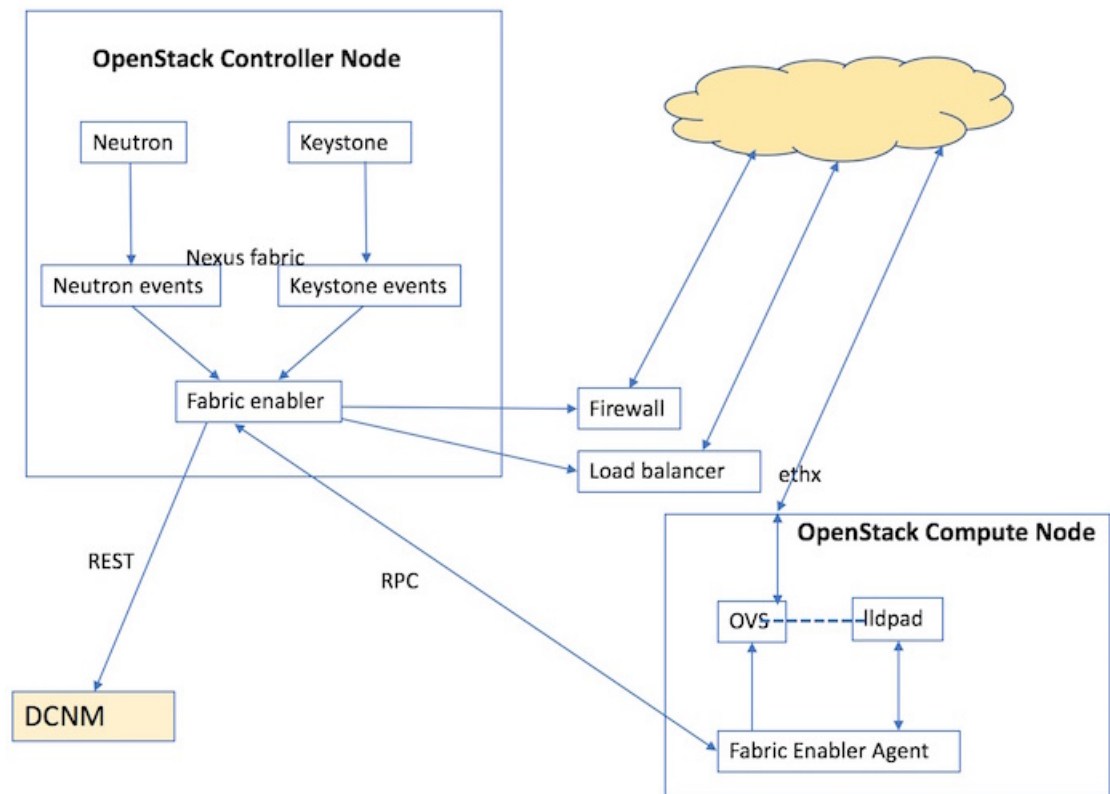
## Assumptions

- 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:

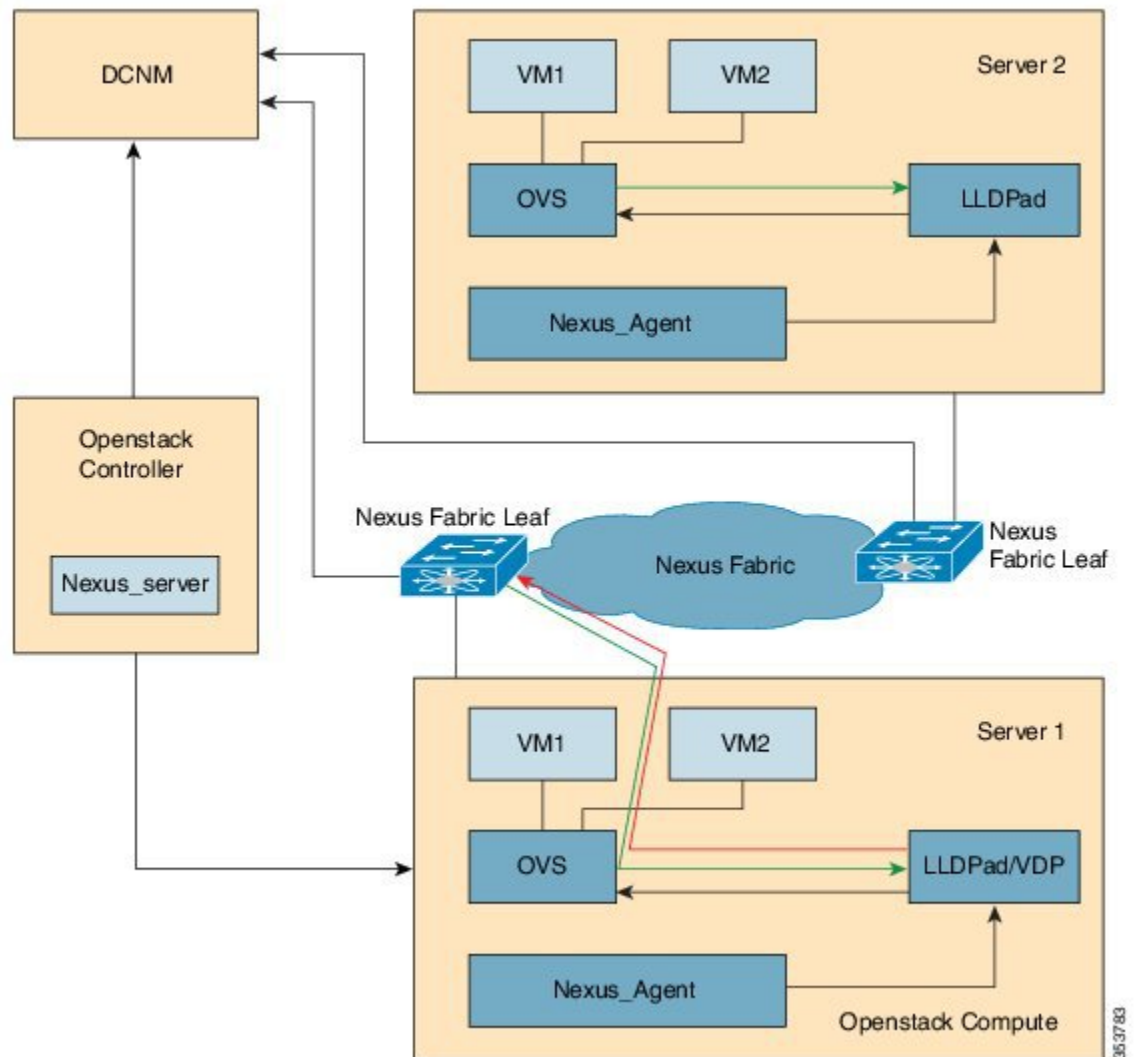
- It is an open source software supporting VDP. Refer to the IEEE 802.1QBG document for more details on VDP functionality.

## **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.

---