Overview

This chapter contains the following sections:

Cisco Nexus 1000V for KVM and OpenStack

The Cisco Nexus 1000V for KVM consists of two main components:

- Virtual Ethernet Module (VEM)—A software component that is deployed on each KVM host. Each VM on the host is connected to the VEM through virtual Ethernet (vEth) ports.
- Virtual Supervisor Module (VSM)—The Management component that controls multiple VEMs and helps in the definition of VM-focused network policies. It is deployed either as a virtual appliance on any KVM host or on the Cisco Cloud Services Platform appliance.

Each of these components is tightly integrated with the OpenStack environment:

- The VEM is a hypervisor-resident component and is tightly integrated with the KVM architecture.
- The VSM is integrated with OpenStack using the OpenStack Neutron Plug-in.

- The OpenStack Neutron API has been extended to include two additional user-defined resources:
  - Network profiles as logical groupings of network segments.

    Note In Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2), network profiles are created automatically for each network type. Network profile creation by administrators is not supported.

  - Policy profiles group port policy information, including security.

Using OpenStack, you create VMs, networks, and subnets on the Cisco Nexus 1000V for KVM, by defining components such as the following:

- Tenants
- Network segments, such as VLANs, VLAN trunks, and VXLANs
- IP address pools (subnets)

Using the Cisco Nexus 1000V for KVM VSM, you create port profiles (called policy profiles in OpenStack), which define the port policy information, including security settings.

When a VM is deployed, a port profile is dynamically created on the Cisco Nexus 1000V for KVM for each unique combination of policy port profile and network segment. All other VMs deployed with the same policy to this network reuse this dynamic port profile.

Note You must consistently use OpenStack for all VM network, subnet, and port configurations. If you create VM networks, subnets, and ports directly on the VSM, the configuration is lost when the OpenStack synchronization occurs.

Tenants

OpenStack has a concept of identity user management called a tenant (also called a project). A tenant is a container used to group resources and/or identity objects. Depending on the how OpenStack is being deployed, a tenant might correspond to a customer, account, organization, or project.

Network Segments

A network segment is an isolated Layer 2 network with a unique broadcast domain (similar to a VLAN). A network segment also facilitates the availability of the network resources to a virtual machine. In OpenStack, a network segment is a VLAN or VXLAN type of network, which provides isolation on virtual networks.

You create a virtual network on the OpenStack Controller using the OpenStack dashboard or the OpenStack CLI commands. When you create a virtual network of type VLAN or VXLAN on the OpenStack controller, OpenStack triggers the auto-creation of a network segment with VLANs or VXLANs on the VSM.

For information about how to create a virtual network, see one the following chapters:

- Creating a Virtual Network Using the OpenStack Dashboard

• Creating a Virtual Network Using the OpenStack CLI

Policy and Network Separation

In the Cisco Nexus 1000V for OpenStack environment, features and network segments are independently associated with the interfaces. The independent association allows you to assign the same set of features on the interfaces that are spread across multiple dynamically-allocated network segments. With this capability, a network administrator can define the policy profiles and export policy profiles to the OpenStack environment. The OpenStack cloud administrator can allocate the network segments from the network pools dynamically, and associate the virtual machine (VM) interfaces to the policy profile and the allocated network segment. This decoupling provides the flexibility to allocate network segments dynamically while grouping the network features to be applied on the interfaces.

IP Pool Templates

An IP pool template represents a block of IP addresses and other network configuration (for example, default gateways or DNS servers) that can be assigned to VMs on a given network. The IP pool templates are the address templates that are applied to the network segments.

The server administrator manages the IP addresses for the virtual environment and assigns a range of IP addresses to the hosts and to the virtual machines that are running inside the OpenStack-managed environment. When creating a subnet for a VM network, the network administrator assigns a range of IP addresses that can be used by the VMs in the network.

The IP pool templates can be reused in the environments with the same IP Address spacing, for example, the duplicate IP addresses are used on the different network segments.

Port Profiles

A port profile is a collection of the interface-level configuration attributes. The network administrator creates a consistent network policy across the similar VM interfaces by defining the Virtual Ethernet port profiles. The network administrator can also create a port profile for the VM hosts adapters. The profile defines the policy to be applied on the physical Ethernet adapters on the servers.

Dynamic Port Profiles

When a VM is deployed, a port profile is dynamically created on the Cisco Nexus 1000V for each unique combination of the Port Classification, the VM Network, and the VM subnet. All other VMs deployed with the same policy to this network reuse this dynamic port profile. This dynamic port profile is a combination of network isolation and network policy.

Note The auto-generated profile should not be modified, inherited in any other port profiles, or referenced in any other configuration. Any changes or references should be in the port-profile inherited by the dynamic port-profile.

When a port-attach notification is received, the port profile UUID and the network segment UUID are generated. A UUID is a globally unique identifier that is used to provide a unique reference for the port profile and the network segment. When a UUID is generated, a new dynamic port profile is created on the VSM that combines the policy profile and the network segment (VLAN/VXLAN). This automatically created port profile is inherited on the interface. If more than one port uses the same combination of the port profile and the network segment, the port profile is shared. The port profiles are dynamically created during the interface attach process.

Bridge Domain

A bridge domain is a Layer 2 flood domain, used for Layer 2 isolation of ports. A bridge domain is distinguished by an identifier, such as a VXLAN segment ID.

Types of OpenStack Networks

Before creating a network using OpenStack, it is important to understand how OpenStack defines these types of networks:

- Virtual network—An OpenStack networking Layer 2 network (identified by a universally unique identifier [UUID] and optional name) whose ports can be attached as vNICs to OpenStack compute instances and to various OpenStack networking agents.
- Physical network—A network connecting virtualization hosts (i.e. OpenStack compute nodes) with each other and with other network resources. Each physical network may support multiple virtual networks. The provider extension and the plugin configurations identify physical networks using simple string names.
- Tenant network—A typical virtual network created by or for a tenant. The tenant is not aware of how that network is physically realized.
- Provider network—A virtual network administratively created to map to a specific network in the data center, typically to enable direct access to non-OpenStack resources on that network. Tenants can be given access to provider networks.
- VLAN Trunk network—A virtual network realized as packets on a specific physical network containing IEEE 802.1Q headers with a specific VLAN ID (VID) field value. VLAN networks that share the same physical network are isolated from each other at Layer 2, and can even have overlapping IP address spaces. Each distinct physical network that supports VLAN networks is treated as a separate VLAN trunk, with a distinct space of VID values. Valid VID values are 1 through 4094.
- Flat network—A virtual network realized as packets on a specific physical network containing no IEEE 802.1Q header. Each physical network can realize at most one flat network.
- Local network—A virtual network that allows communication within each host, but not across a network. Local networks are intended mainly for single-node test scenarios, but may have other uses.

Comparison of Network Terminology

Cisco Nexus 1000V for KVM and OpenStack use many of the same components and concepts. However, they have given these components and concepts different terminology. The following table defines these components and concept and maps the ones that are different.

Cisco Nexus 1000V for KVM	OpenStack	Description
—	Linux KVM	Linux Kernel-based virtual machine that functions as a hypervisor.
—	OpenStack Controller	Point of management.
—	Neutron	Point of network management.
Logical network	Container object	Server nodes (virtual machines), network nodes, and network services that logically isolate network traffic and partition needed resources.
Network segment pool	Cisco network profile	A container that allows you to associate IP address blocks and other network configuration settings with a neutron network. OpenStack supports VLAN, overlay (VXLAN), and trunk types.
Network segment	Network	Represents an isolated virtual Layer 2 network domain (similar to a VLAN); Can also be regarded as a virtual or logical switch.
IP pool template	Subnet	Represents a block of IP addresses and other network configuration (for example, default gateways or DNS servers) that are assigned to VMs on a given network.
Network vEthernet	—	The combination of a network segment and a port profile policy.
Network vEthernet port	port	Ports that represent virtual (or logical) switch ports on a given network.
Dynamic port profile	—	An automatically generated combination of a policy port profile and network segment. Dynamic port profiles have **vmn** as a prefix.

Cisco Nexus 1000V for KVM	OpenStack	Description
Bridge domain	—	A bridge domain object is created only in the Virtual Supervisor Module (VSM) and not in OpenStack. When a VXLAN network is created, Openstack requests the creation of a bridge domain in VSM. The newly created bridge-domain is used to configure the VXLAN network segment.

Neutron-to-VSM Configuration Synchronization

In order to keep the Neutron service and the VSM configurations in synchronization, the Cisco Nexus 1000V Neutron Plug-in has the capability to restore the configuration in the VSM under certain situations.

Rollback for Neutron Resources

The Neutron service rolls back to the previous configuration if it fails to create or update a resource on the VSM. If the Neutron service sends a resource request to the VSM and the VSM responds with an HTTP error, the Neutron service deletes the resource and all of its associated bindings and logs an exception in the Neutron server logs.

State Synchronization

Note Starting with Release 5.2(1)SK3(2.2), you no longer have to restart the Neutron service to trigger a full synchronization. However, a bridge domain synchronization between the Neutron service and the VSM only happens when the Neutron service restarts.

An automatic state mismatch check on the VSM is performed every five minutes unless you change the default duration. To change the default duration, edit the **sync_interval** parameter located in the /etc/neutron/plugin.ini file. If there is a state mismatch, a create or delete operation on the VSM is performed to get it in sync with the Neutron. The resources that are synchronized include network profiles, networks, subnets, ports, and bridge domains. If there are only certain resources out of sync on the VSM, synchronization will occur only for those resources.

Policy profiles that are missing from the VSM or in use in Neutron are not restored automatically as part of a full synchronization. You must manually create them on the VSM using the same UUID values before you restart the Neutron server to trigger a full synchronization.

Synchronizing a Fresh VSM

Use this procedure to perform a full synchronization of the VSM after a reload or as a part of the VSM recovery (fresh VSM bring-up).

Before You Begin

Before beginning this procedure, you must know or do the following:

- You are logged into the CLI in EXEC mode.

Procedure

Step 1 From the old VSM, copy the running configuration to an external location.
copy running-config tftp://*external-location*

Step 2 Generate the **selective-config-file**.
```
show running config static
```

Step 3 Enable the management communication using either Telnet or Secure Shell (SSH).

a) On the Nova-Cloud-Controller (Neutron Server), locate the `cisco_plugins.ini` configuration file at the following path: `/etc/neutron/plugins/cisco`.

b) In the `cisco_n1K` section, add enable_sync_on_start=True in the `cisco_plugins.ini` file.

Step 4 Restart the Neutron server to start the Neutron-VSM synchronization process.
```
root@ncc:~# service neutron-server restart
neutron-server stop/waiting
neutron-server start/running, process 24157
```

Step 5 Verify if all the configurations and vEthernet interface have come up on the VSM.
```
show running-config
```