# Chapter __ :

## Neutron Networking

1. Openstack networking basics
2. Networking services
3. How VLAN Tagging works
4. Services and Component

# __: Neutron Networking

In this chapter we will cover:

1. Openstack networking basics
2. Networking services
3. How VLAN Tagging works
4. Services and Component

## 1. **Openstack Networking Basics**:

Neutron is the *networking component* of OpenStack, and is a standalone service alongside other services such as Nova (Compute), Glance (Image), Keystone (authentication), and Horizon (Dashboard). Like those services, the deployment of Neutron involves deploying several processes on each host.

- Neutron relies on Keystone for authentication and authorization of all API requests.
- Nova interacts with Neutron through API calls. As part of creating an instance, nova-compute communicates with the Neutron API to plug each virtual NIC on the instance into a particular Neutron network through the use of Open vSwitch.
- Horizon has a basic integration with the Neutron API, and allows tenants to create networks and subnets. Users are able to provision NICs to instances that connect to tenant networks and/or provider networks to provide connectivity to and from instances.

Thanks to its pluggable infrastructure, third-party and community developers can create plugins to extend the use and capabilities of Neutron within a cloud. There are plugins for LBaaS (load-balancing as a service), VPNaaS (VPN-as-a-service), Layer 2, Layer 3 and more.

OPEN VSWITCH / HOW DOES IT FIT?

Open vSwitch is an open source, software-based virtual switch that is utilized by Neutron. It can operate both as a soft switch running within the hypervisor, and as the control stack for physical switching devices. The Neutron openvswitch plugin consists of two components:

- A *plugin* loaded at runtime by the Neutron service. The plugin processes API calls and stores the resulting logical network data and mappings in a database backend.

- An *agent* that runs on each compute node. This agent gathers the configuration and mappings from the central database and communicates with the local Open vSwitch instance to configure flows and implement the network based on the logical data model.

For OpenStack, Open vSwitch is installed as a kernel module or userspace-only process. Much like a physical switch, Open vSwitch is responsible for the proper tagging and forwarding of traffic based on OVS port configuration. Aside from building the initial bridge(s), Neutron handles most all other interaction with OVS via the openvswitch plugin. It is possible to manipulate OVS outside of Neutron for further networking requirements, but these scenarios are outside the scope of this article.

## BASIC CONNECTIVITY / PROVIDER AND TENANT NETWORKS

One of the core requirements of a networking service for OpenStack is to provide connectivity to and from instances.

There are two categories of networks that can be created within Neutron:

- Provider Networks
- Tenant Networks

Both network types can be used to provide connectivity to and from instances. However, you must configure at least one provider network in your environment. A provider network can be used directly by instances themselves, or as the front-end (WAN) network for a Neutron router.

Provider networks are networks created by the OpenStack administrator that map directly to an existing physical network in the data center. An example of this would be networks behind a set of firewalls or load balancers that are routable within your data center. Useful network types in this category are *flat*(untagged) and *vlan* (802.1q tagged). It is possible to allow provider networks to be shared among tenants as part of the network creation process.

*Tenant networks* are networks created by users within tenants, or groups of users. By default, networks created with tenants are not shared among other tenants. Useful network types in this category are *vlan*(802.1q tagged) and *gre* (unique id). With the use of the L3 agent and Neutron routers, it is possible to route between GRE-based tenant networks. Without a Neutron router, these networks are effectively isolated from each other (and everything else, for that matter).

## 2. Networking Services:

OpenStack Networking adds a layer of virtualized network services which gives tenants the capability to architect their own virtual networks. OpenStack Networking allows tenants to create advanced virtual network topologies including services such as firewalls, load balancers, and virtual private networks (VPNs).

**VLANs**

VLANs are realized as packets on a specific physical network containing IEEE 802.1Q headers with a specific VLAN ID (VID) field value. VLAN networks sharing the same physical network are isolated from each other at L2, and can even have overlapping IP address spaces. Each distinct physical network supporting VLAN networks is treated as a separate VLAN trunk, with a distinct space of VID values. Valid VID values are 1 through 4094.

**L2 tunneling**

Network tunneling encapsulates each tenant/network combination with a unique "tunnel-id" that is used to identify the network traffic belonging to that combination. The tenant's L2 network connectivity is independent of physical locality or underlying network design. By encapsulating traffic inside IP packets, that traffic can cross Layer-3 boundaries, removing the need for preconfigured VLANs and VLAN trunking. Tunneling adds a layer of obfuscation to network data traffic, reducing the visibility of individual tenant traffic from a monitoring point of view.

OpenStack Networking currently supports both GRE and VXLAN encapsulation.The choice of technology to provide L2 isolation is dependent upon the scope and size of tenant networks that will be created in your deployment.

**Firewalls**

FW-as-a-Service (FWaaS) is considered an experimental feature for the Kilo release of OpenStack Networking. FWaaS addresses the need to manage and leverage the rich set of security features provided by typical firewall products which are typically far more comprehensive than what is currently provided by security groups. Both Freescale and Intel developed third-party plug-ins as extensions in OpenStack Networking to support this component in the Kilo release.

There are four main services that interact with OpenStack Networking. In a typical OpenStack deployment these services map to the following security domains:

- OpenStack dashboard: Public and management
- OpenStack Identity: Management
- OpenStack compute node: Management and guest

- OpenStack network node: Management, guest, and possibly public depending upon neutron-plugin in use.
- SDN services node: Management, guest and possibly public depending upon product used.

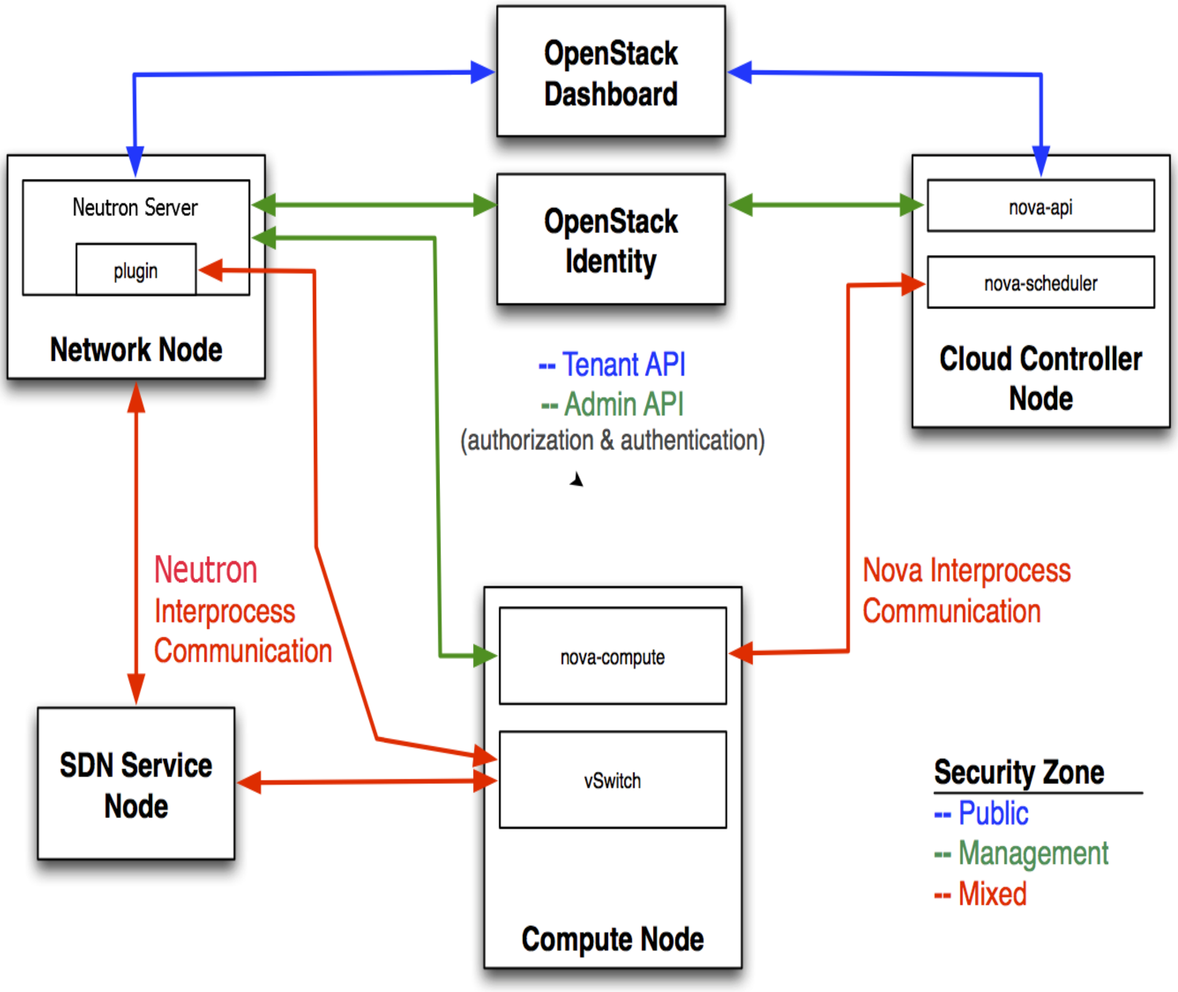


Figure: Openstack Networking Sevices

## 3. How VLAN Tagging works:

In order to understand VLAN tagging, let's understand some basic concepts.

A **local network** is a network that can only be realized on a single host. This is only used in proof-of-concept or development environments, because just about any other OpenStack environment will have multiple compute hosts and/or a separate network host.

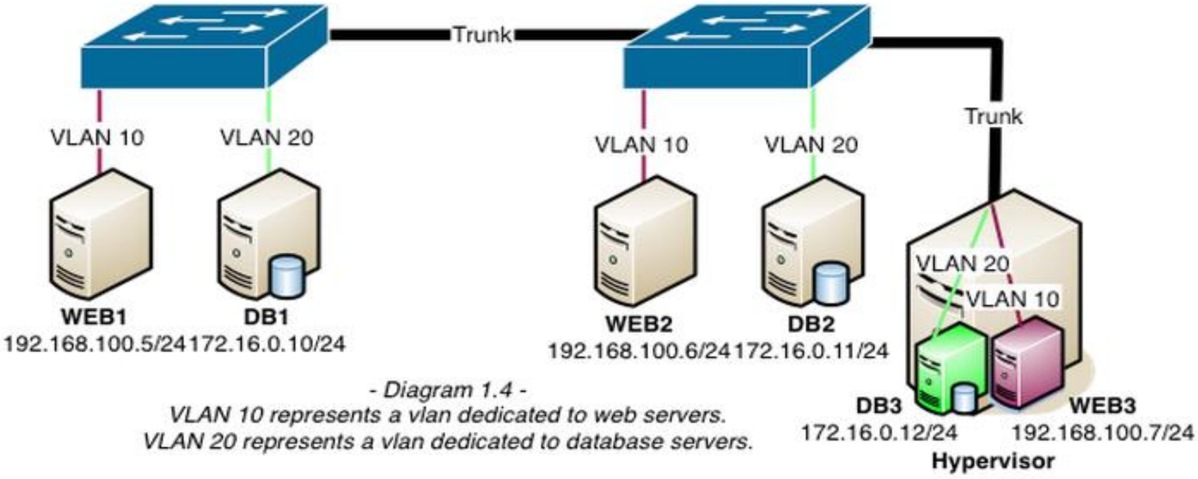
A **flat network** is a network that does not provide any segmentation options. A traditional L2 ethernet network is a "flat" network. Any servers attached to this network are able to see the same broadcast traffic and can contact each other without requiring a router. In a flat network, everyone shares the same network segment. For example, say 2 tenants are sharing the cluster, and this segment is 10.4.128.0/20 - VM1 from tenant 1 might get assigned 10.4.128.3, VM1 from tenant 2 might get 10.4.128.4, and so on. This means that tenant 1 can see the traffic from tenant 2. Not a good thing in most cases.

A **vlan network** is one that uses VLANs for segmentation. When you create a new network in Neutron, it will be assigned a VLAN ID from the range you have configured in your Neutron configuration. In a VLAN network, tenants are separated because each is assigned to a VLAN. In OpenVSwitch plugin (or ML2 with OVS driver), OVS allocate an internal VLAN for each tenant. These VLANs provide separation amongst the tenants (as VLANs are designed to do). It also means that tenants can specify the same subnet and overlap in that subnet range - VM1 from tenant 1 can get assigned IP 10.4.128.3 and VM1 from tenant 2 can also get 10.4.128.3, without conflict. This makes life easier for administrators because they don't have to worry about tenants that want the same subnet and address allocations, because the VLANs keep them separate.

## VLAN Tagging

When VLANs span multiple switches, VLAN Tagging is required. A VLAN is a method of creating independent logical networks within a physical network. VLAN Tagging is the practice of inserting a VLAN ID into a packet header in order to identify which VLAN (Virtual Local Area Network) the packet belongs to. More specifically, switches use the VLAN ID to determine which port, or interface, to send a broadcast packet to.

At a basic level on a Cisco switch there are two types of switchports: **access ports** and **trunk ports**. Switchports configured as access ports are placed into a single vlan and can communicate with other switchports in the same vlan. Switchports configured as trunks allow traffic from multiple vlans to traverse a single interface. The switch adds a tag to the Ethernet frame that contains the corresponding vlan ID as the frame enters the trunk. As the frame exits the trunk on the other side, the vlan tag is stripped and the traffic forwarded to its destination. Common uses of trunk ports include uplinks to other switches and more importantly, hypervisors serving virtual machines from various networks.



*- Diagram 1.4 -*
*VLAN 10 represents a vlan dedicated to web servers.*
*VLAN 20 represents a vlan dedicated to database servers.*

*Traffic between WEB1 and WEB2/WEB3 gets tagged on the trunk as VLAN 10.*
*Traffic between DB1 and DB2/DB3 gets tagged on the trunk as VLAN 20.*

*Traffic entering or leaving the hypervisor gets tagged appropriately.*

*Neither group of servers, DB or WEB, can communicate between each other in this scenario.*

### How does this apply to neutron?

When using VLAN tagging as an isolation mechanism a VLAN tag is allocated by Neutron from a pre-defined VLAN tags pool and assigned to the newly created network. By provisioning VLAN tags to the networks Neutron allows creation of multiple isolated networks on the same physical link.  The big difference between this and other platforms is that the user does not have to deal with allocating and managing VLANs to networks. The VLAN allocation and provisioning is handled by Neutron which keeps track of the VLAN tags, and responsible for allocating and reclaiming VLAN tags.

Neutron allows users to create multiple provider or tenant networks using vlan IDs that correspond to real vlans in the data center. A single OVS bridge can be utilized by multiple provider and tenant networks using different vlan IDs, allowing instances to communicate with other instances across the environment, and also with dedicated servers, firewalls, load balancers and other networking gear on the same Layer 2 vlan.

## 4. Services and Components

### Network Components

### Switches

A switch is a device that is used to connect devices on a network. Switches forward packets on to other devices, using packet switching to pass data along only to devices that need to receive it. Switches operate at layer 2 of the OSI model.

### Routers

A router is a networking device that connects multiple networks together. Routers are connected to two or more networks. When they receive data packets, they use a routing table to determine which networks to pass the information to.

**Firewalls**

A firewall is a network device that controls the incoming and outgoing network traffic based on an applied rule set.

**Load balancers**

A load balancer is a network device that distributes network or application traffic across a number of servers.

OpenStack Networking allows us to create and manage network objects, such as networks, subnets, and ports, which other OpenStack services can use. Plug-ins can be implemented to accommodate different networking equipment and software, providing flexibility to OpenStack architecture and deployment.

The Networking service, code-named neutron, provides an API that lets you define network connectivity and addressing in the cloud. The Networking service enables operators to leverage different networking technologies to power their cloud networking. The Networking service also provides an API to configure and manage a variety of network services ranging from L3 forwarding and NAT to load balancing, perimeter firewalls, and virtual private networks.

It includes the following components:

**API server**

The OpenStack Networking API includes support for Layer 2 networking and IP address management (IPAM), as well as an extension for a Layer 3 router construct that enables routing between Layer 2 networks and gateways to external networks. OpenStack Networking includes a growing list of plug-ins that enable interoperability with various commercial and open source network technologies, including routers, switches, virtual switches and software-defined networking (SDN) controllers.

### OpenStack Networking plug-in and agents

Plugs and unplugs ports, creates networks or subnets, and provides IP addressing. The chosen plug-in and agents differ depending on the vendor and technologies used in the particular cloud. It is important to mention that only one plug-in can be used at a time.

### Messaging queue

Accepts and routes RPC requests between agents to complete API operations. Message queue is used in the ML2 plug-in for RPC between the neutron server and neutron agents that run on each hypervisor, in the ML2 mechanism drivers for *Open vSwitch* and *Linux bridge*.

## References:

- http://docs.openstack.org/security-guide/networking/securing-services.html
- http://docs.openstack.org/admin-guide-cloud/networking_introduction.html
- http://docs.openstack.org/icehouse/install-guide/install/apt/content/neutron-concepts.html
- https://developer.rackspace.com/blog/neutron-networking-the-building-blocks-of-an-openstack-cloud/
- https://ask.openstack.org/en/question/51388/whats-the-difference-between-flat-gre-and-vlan-neutron-network-types/
- http://www.firewall.cx/networking-topics/vlan-networks/219-vlan-tagging.html
- http://www.dummies.com/how-to/content/how-virtual-local-area-networks-vlans-work.html
- https://blogs.oracle.com/ronen/entry/diving_into_openstack_network_architecture
- http://docs.openstack.org/networking-guide/
- http://docs.openstack.org/networking-guide/intro_networking_components.html
- http://docs.openstack.org/networking-guide/intro_os_networking_overview.html
- https://developer.rackspace.com/blog/neutron-networking-the-building-blocks-of-an-openstack-cloud/