

**OPEN STACK - NEUTRON**

(MSEE FALL 2016)

Submitted by:

Arpit Singh (010810514)

**Introduction**

Cloud computing is computing using the internet. Previously, one used to run applications from software downloaded on a computer hardware directly or server in a data center. Cloud computing is a computing model that allows the user to use computing power, storage devices and the network as a service instead of investing in these systems. Cloud facilitates the user to access these devices remotely. This whole infrastructure setup described above is known as Cloud. The cloud technology allows to access the same kind of application on the internet, the application is deployed on a server and accessed remotely. The cloud computing increases the efficiency of this whole process. Some important advantages of using cloud are as follows,

Flexibility is a cloud-based service that facilitates the growing business or the business that has fluctuating bandwidth demands. This property can be used in both ways either for scaling up the storage capacity or for shredding out the capacity.

Every business need robust disaster discovery but the issue is with the smaller businesses, Small business firms cannot afford the required cash and expertise required for a disaster discovery. Cloud computing services provide this disaster recovery as a compliment to other services.

Automatic Software Updates This service helps the system administrators a lot, this allows the regular software updates and security updates, so no need to waste time maintaining the system.

Capital Expenditure Free This service helps cut the cost of this whole system, a user can subscribe to model according to the need. This service adds to the ease of setup and management of their infrastructure.

Work from anywhere This is the most important service provided by these cloud service providers, one can work from anywhere if the internet connection is available, and in addition to this the cloud services offer mobile apps that even removes the device restriction for the user.

Security is the most important issue, lost laptops are the biggest problems, loss of a laptop is nothing compared to the sensitive data that it stores, the cloud data is free from all these risks.

For managing a Cloud computing network, a set of rules or protocols are required. These rules are described in an Open Stack protocol suite. For managing such a large network, a separate unit Neutron is deployed. Neutron is responsible for managing and setup of a network architecture. Neutron architecture and working is clearly described below.

**Introduction to Open stack – Neutron**

The open stack is a platform designed by Rack space and NASA in 2010. This project was made to help the organizations in running the cloud computing services in their traditional hardware. Open stack is open source that means the source code of this whole architecture is made available to public with a license that provides the right to study and edit the software according to the user demand. This stack consists of its main computing unit (Nova), the networking unit (Neutron), Object storage unit (Swift), Block storage unit (Cinder), Identity management unit (Keystone), and for searching (Search-light). All these software platforms are used to manage various multivendor hardware and Software components throughout a data center.

The computing unit Nova is the main controlling unit of this open stack Infrastructure. Nova is basically used for managing the computer resources, using virtualization techniques. The open stack is written in python programming language and uses many libraries such as Eventlet, this library allows simultaneous programming.

Kombu, this library is used for massaging and Quality of service related protocols. These are mainly the communication network related protocols for queueing and advanced messaging.

SQLAlchemy, this is a python SQL toolkit. This is an Object relation mapper that provides the programmer a flexibility to utilize all the services of SQL. This library includes the database management, database editing, and queries for database access.

Complete open stack architecture is designed in such a way that provides the users the ability to use the traditional devices, third-party devices, and other legacy systems altogether.

Cinder, the Block storage unit provides the data storing devices for the Virtual Instances of computing devices in an open stack. This block storage service is responsible for managing the creation, attaching and detaching the memory blocks to servers.

Swift is the unit responsible for storing the data in form of objects. The swift unit is a scalable system for storage, Unlike the Cinder block storage, in swift, the data is stored in disk drives. These disk drives are spread throughout in a data center. The open stack architecture is responsible ensures that the integrity of its data is maintained and data is not replicated. Replication of data leads to the memory wastage leads to inefficiency. The swift module is basically responsible for saving unstructured data. Unstructured data is the data that grows without any constraint.

Keystone, the identity management unit is responsible for maintaining a user database, this service maintains a directory of all the users and the services provided to those users. Keystone is the only authentication system across the cloud architecture for managing the username and passwords.

Searchlight, the search tool inside the open stack provides search capabilities that runs across the cloud services. Searchlight is integrated into the horizon module inside the open stack as described below.

Horizon, also known as the dashboard for the open stack. Horizon provides the administrators and users a graphical user interface to access the cloud services. The dashboard design consists of third party products and services that include the monitoring system and management tools.

The Open stack provides networking service known as Neutron. The Neutron is an Application programming Interface that defines the connectivity and addresses inside the cloud. An application programming interface also known as API is a tool that enables the developers to create software applications. The Neutron is a controller that can be referred to as a Networking as a service (NaaS) project within the Open stack. Networking as a service means to provide a virtual network as a service from the owners of a Network Infrastructure. The Neutron provides the users multiple networking technologies inside the cloud Network.

When open stack was in its early days, the Neutron was actually shared with the Nova module and hence was called the Nova-Networking. But when the demand for it increased and Nova networking proved to be less flexible, then it was separated and developed as an independent identity called Neutron. This new Open stack entity is responsible for various network entities such as ports, interfaces, switches, routers, floating IPs, and Security groups. All these above-mentioned APIs are similar to the Software defined networking’s North Bound API. Northbound API of a Software Defined Network is the interface between the applications and the SDN controllers.

**Building Blocks of Neutron**

As the network size increased it became hard to manage such a large network, so the computation and storage are virtualized, and SDN is used for its maintenance. Networks are hard to evolve, due to ongoing innovations in system software and various operating system. The networks are still stuck in past because the routing algorithms change slowly, network management protocols are still very old.

Software-defined Networking is the architecture that separates the network control plane from the forwarding plane. The control plane is the controlling unit for the network peripherals like Routers, the control plane is responsible for defining the paths for the incoming packets and designing the route tables.

Control plane calculates the router’s state. It also determines how the packets are forwarded and where the packets are forwarded. The control plane is implemented with distributed protocols, manual configuration, and scripting or centralized computation.

The forwarding plane also known as data plane is responsible for deciding what to do with the packets arriving on an inbound interface. This includes the processing and delivery of packets with local forwarding state. The forwarding decision is made according to the forwarding state and the packet header.

The basic functionality of SDN is to separate these two entities for the better and efficient control over the network because abstracting control from forwarding plane allows dynamically controlling the networking devices.

SDN leverages the advantages of Network function virtualization (NFV). Network function virtualization offers a way to design the networking services, deploy them and also manage the networking services.

Network function virtualization separates the network functions such as Network Address Translation, firewalls, Intrusion detection systems and Domain name services. Network Address Translation or NAT is a mapping of private IP address to Public IP addresses. Domain Name Services or Domain name service is to map the hostnames to their respective IP addresses and return back to the user.

Intrusion detection system is a software application that monitors network or a system for any malicious activity or attacks. NFV is the concept of using software running on commercial off the shelf servers instead of dedicated hardware network appliances such as routers, switches, and firewalls.

The separation of network control system and the forwarding system in SDN provides a centralized view. The centralized view in a distributed network provides more efficient services.

NFV uses software instead of using traditional hardware devices and hence separates the various services like caching and DNS.

**Open Flow Neutron Working and Application**

Open stack architecture is divided into three levels, the lower level is standard hardware devices, the computing architecture, memory, Networking devices. The middle layer consists of Storage, Networking, and Compute. All these three units are united in an Open Stack Dashboard. Above all, this is the Application software, the governing body of Open Stack.

If a user logs in, then the user is presented with the main Dashboard. The dashboard has computed API, Network API, Storage API in it. The user just needs to take care of these three basic units, rest is handled through the Open Stack APIs.

Neutron provides networking as a service, Neutron provides the rich topologies. A rich topology is in which the network has many paths as a backup. If a link breaks or some another breakdown occurs, then the rich network can settle the network back again in no time. Neutron is technology agnostic this means that the neutron can be used in an unbiased way towards the use of technology to solve multiple problems. Neutron provides following advanced services support.

LBaaS or Load balancer as a service, this service facilitates the incoming requests evenly to all the servers. This leads to the efficient utilization of all the resources. This scheme allows the equal distribution of resource, avoiding overburden on a particular server.

For managing the incoming requests, the Load Balancer service uses Round Robin scheme, in this scheme, the turns are allotted to the user circularly, one by one the turns are provided.

Second method is source IP, under this scheme the requests from a particular IP address is consistently directed to an instance, this method is useful in cases when the heavy loads occur inside the network, if a server is handling low load then the traffic can be directed to that server instead of dividing among all the servers.

Third scheme for the request handling is “Least connections”, under this scheme the requests are allocated according to the least number of connections, suppose there are multiple servers and a new request arrives then instead of checking the resource availability, the server that has the least active connections is selected, although that server may already have maximum number of connections, but because of their inactive connections the server is selected.

FWaas or Firewall as a service, provides the firewalls to open stack objects such as projects, routers, and ports. The main concept with open stack firewalls is to apply the firewall policies and firewall rules to the open stack. A firewall policy is a set of rules, these rules are made of various attributes such as port ranges, protocol, and Ip addresses. These attributes are predefined according to the demands and then they are matched with the action to take such as allow or deny a traffic at ingress or the egress point. There are two versions of FWaaS,

FWaaS v1 provides the protection to routers. A firewall protects all the internal ports of a router. In this case, FWaas v2 service provided is more dedicated towards the usage, in routers policies are designed for the ingress and egress ports.

**Virtual private network as a service**

Also known as VPNaaS is the service provided for sharing the Network over the internet. VPN service provides multiple tunneling and security features, these features include static as well as dynamic routing.

Tunneling means to create a secure link between two hosts that provides encryption to the data passed through it.

IPsec is one other important feature of VPNaaS, IPsec stands for internet protocol security, VPN provides this as a service. IPsec is responsible for providing end to end authentication and encryption to the data between hosts. When a packet is sent from an end to another through an IPsec VPN tunnel, the packet is encapsulated by an IPsec frame and an IP frame at the outer end, which is responsible for the data transmission.

IPsec introduces two types of security services,

Authentication header, this service is responsible for the data authentication and for checking the data integrity during transmission.

Encapsulating Security Payload also known as ESP is responsible for authenticating the sender as well as provides the data encapsulation.

Design Goal of Open Stack Neutron

Open stack is designed keeping in mind the following goals,

Unified API, Previously, the APIs were only built application related and according to the hardware demands, but now a single API can run on various machines. A single code can be shared between Mac and iOS, the same code is compatible with wide range of machines, it can support 32 and 64-bit applications using the same binaries. This code sharing is made possible with the use of Unified APIs.

Small Core, in a network the central unit is responsible for providing services to various other networks or customers connected to it, this central entity is known as the Network core. The network core is responsible mainly for communication in a PSTN network for routing phone calls across the network. Typically, the network cores refer to high capacity facilities that connect various subnets together. Neutron provides a small network core that is easy to maintain. Neutron provides the core networks with a mesh topology that provides a connection between subnets. Technologies used n the network core are of data link layer and Network layer.

Pluggable open architecture is a software architecture that provides the flexibility to use this software without any dependencies, this type of software applications contains channels of information and it is very rare that the plugins communicate with each other. In this type of software, the central functions are of main importance.

Extensible, this property of Neutron allows the software architecture to add additional functionalities. The additional plugins have no effect on the system’s internal structure or the data flow inside the architecture. Extensibility of a software architecture is an implementation on the basis of its architecture. The main aim or central focus is to provide the change to the typical functionalities without impacting existing system functionalities.

Growing ecosystem, this design of open stack is with keeping into consideration the stealthily growing network size and user demands. Open networking came out to be a boon for the cloud industry and seems to be the most important next generation architecture.

**Basic deployment of Open Stack Neutron**

The main entity in this architecture is the Neutron server, which is connected to a Database. A neutron-server is a web server that provides the Neutron API. Neutron server is also responsible for passing all the web service calls to the Neutron plugins for further processing.

After this architecture, there is a unit called message queue that connects the neutron-server to a Stack consisting of four units of L2 Agent, L3 agent, DHCP agent, Advanced Services.

A message queue, this unit is responsible for the messaging services for its internal components. AMQP or Advanced Message Queue Protocol is the messaging technology used in Message queues. The internal units of open stack Nova, Cinder, Quantum, all these units communicate internally using AMQP and through each other using Rest call. Rest also called Representational state transfer is a client-server communications protocol. HTTP protocol is used in almost all the Rest cases. Rest is the architecture that is used for designing the network applications.

The AMQP broker or the client is used between two internal components of an open stack and allows the components to communicate in a loosely fashioned way. Loosely fashioned communication means the components need not know about the other components or do not make use of them.

L2 Agent, this agent is responsible for creating layer 2 connectivity between the instances. The L2 agent is also responsible for creating bridges with open stack nova for filtering the malicious connections. Open virtual switch neutron agent can also be configured in such a way that networking technologies can be used for project isolation.

L3 Agent, Neutron’s Layer 3 agents are responsible for creating links between the Layer 3 and layer 2 protocols. The L3 agent provides API that is responsible for creating extensions to allow admins and users to create routers. These routers also known as neutron i3 agent are used to create a connection to the layer 2. The L3 agent uses Linux IP stack. IP stack of Linux comprises of IP tables, for L3 forwarding and NAT.

Neutron agents use Linux namespaces. Linux namespaces are used to provide isolated forwarding. Similar to the DHCP namespaces that are responsible for every network defined in neutron, each router will have its own namespace with a name based on its UUID. UUID also known as the (universally unique identifier) is a standard used for software construction. UUID is a 128-bit key value. The main use of UUID is to distribute a system without the need of any central coordination.

DHCP Agent is a networking service that is responsible for scheduling tasks. The scheduler allows the usage of multiple agents across nodes. High availability can be achieved in a network by assigning more than one DHCP agent per network.

Adv service, Neutron provides various advanced services, one of them Is Akanda. Akanda is a multi-process service that provides multi-threaded orchestration service. This service currently supports routers, load balancers, Virtual private networks, and firewalls. Akanda is a simple architecture compatible with various third party devices. This service runs on the top of Neutron server and connects message queue and service instances together.

**Neutron Network Types**

There are four types of networks in a neutron architecture. Local network, flat network, VLAN network, and gre or vxlan network.

Local Network is a network that can only be deployed on a single host is known as the local network. These types of networks are used in development environments or proof of concepts. Local networks are created such that the traffic is only restricted to the node it is created on. Local networks do not support DHCP and metadata. Local networks are useful for environments where the complex technology is used, and the tenant wants to keep some traffic constrained to a small number of virtual machines on a single host.

Flat Network packets are transmitted irrespective of their sizes, the Flat network does not provide any segmentation options. An example of a flat network is an Ethernet network. In an Ethernet network, the attached servers are able to see the broadcast traffic without the use of a router, and every server can contact each other. Flat networks are used to attach the Nova servers to the provider networks. Existing networks are known as the provider networks.

A VLAN network is a group of various VLANs. In a neutron VLAN network whenever a new network is created it is assigned a VLAN ID. This ID is from the range that is defined while configuring the neutron configurations. VLAN networks require that all the switches are configured, such that they trunk the corresponding VLANs.

gre and vxlan networks, both these networks are very similar to each other, they can overlay the networks that use encapsulating network traffic. GRE or VXLAN receives a tunnel id similar to a VLAN tunnel id, but an overlay network of gre does not require any synchronization with the open stack configuration.

Tenant overlay networks are useful for installations when there are a larger number of tenants available. This network allows tenants to create very descriptive network layouts. Tenant overlay allows the tenants to overlap the networks and share IP address spaces. For accessing a network remotely tenant overlay network allows floating IPS. These networks can also be connected to VLAN networks using L2 gateways.

**Neutron Network Components**

Similar to a traditional network, neutron network consists of various networking devices like subnets, ports, routers, network address translation and various other services.

Neutron subnets are used to divide a network into smaller units logically is known as sub netting. In a traditional network IP addresses are divided into network and host part, with the use of subnets in a network various sub hosts can be divided. Similarly, sub netting provides a feature to improve the workflow. Neutron pools facilitate a service of taking over a pool of address and then managing it, whenever one needs to create a subnet, a new address can be fetched from the pool. Subnet pools also provide a facility to manage address across various projects. This also prevents the addresses from overlapping. If suppose an address is from an externally routable pool, then also all the projects have the routable and unique address.

Neutron subnet pool service is useful in two scenarios, IPv6 networks, because open stack does not support floating IPv6 addresses and when one needs to create a route directly from an external network to a project network.

Neutron port is an address that is used for logically separating the processes. In a cloud network, the main ingress and egress points are the ports for any type of data traffic although the major entities across the network seem to be routers, firewalls, and other visible hardware devices. The open stack provides rich support for the virtual networking devices. Ports in an open stack network are usually created automatically, although the user can manually create the ports using command line interface.

Open stack networking ports are mainly build using virtual interfaces on the hypervisor on which the machine is running. Virtual machine’s IP addresses are stored with the port addresses attached to it. There are four types of ports in a neutron network,

The First is “compute: nova”, this pair indicates that the post is attached to a virtual machine. These types of ports are created automatically when the machine instances are created. Compute in "compute: nova" represents that the port is linked to the nodes.

“Network: DHCP”, this pair represents that the port is associated with the DHCP server. The network part in this pair implies that this port is created for the network node. When the first VM instance is created a DHCP port in started in its corresponding network.

“Network: router\_interface”, this pair represents the IP interface of a gateway router of a tenant network and its corresponding virtual machine. This port is created when a user adds a new interface on a router. This address is associated with the namespace of an open stack router.

“Network: router\_gateway”, when a user runs a set gateway command a new port is created, this port pair for the outer world represents a gateway. So for this, the special name is given as network router gateway.

Neutron namespaces and IP tables

Namespaces in a neutron network are used to allow multiple instances of a routing table, this property is useful in cases when multiple machines are installed within a single Linux box. Similarly, namespaces are used in a network to separate network domains, interfaces, and IP tables into independent entities.

Namespaces have a wide range of advantages, first is to allow overlapping IPs. This allows the tenants to can created multiple IPs, namespaces maintain the IP overlap without any exceptions. This is an important attribute for the cloud users because they can leverage this property and can create any subnet of their choice without any conflict with that of another tenant. Linux network namespace is required mainly on the network I3 agent and neutron DHCP agent IP if overlapping IP is in use. Hence the hosts running in these two domains must be assigned to network namespaces.

If the machine does not support namespaces then the following limitations should be noted,

The neutron I3 agent is limited to provide a single virtual route per node. But If the namespaces are supported a single neutron agent can be used to host several virtual routers in a single network.

Each neutron agent is configured with a Universally Unique ID, that identifies the router that the agent hosts. But this process complicates the deployment, this makes it impossible for tenants to self-service the routers. With the help of namespace, the router configuration with the UUID’s is possible.

If the namespace is not supported, the neutron I3 agent and the neutron DHCP agent have to be installed on separate hosts. This separation is because the L3 agent and DHCP agent creates no isolation. This leads to a vulnerability, because if a user manipulated a routing table then he can gain access to another user.

Neutron Routers are similar to the traditional routers, neutron routers provide the routing instructions in a cloud network. Neutron routers are divided according to the number of users in contrast to the traditional network in which the routers are according to the network. Admin creates Neutron routers. These routers support both IP4 and IPv6. Neutron routers act as a gateway router it is capable of routing information from internal networks to outer networks and vice-versa. In addition to this, the neutron routers are also capable of routing traffic between internal networks.

Neutron Network address translation or Neutron NAT is a concept of separating private address from the public address. NAT allows the modification of source or destination addresses of an IP packet while the packet is in transit. Generally, the sender and receiver are unaware of this manipulation. In open stack, the Linux servers are responsible for the NAT implementation, not the hardware routers. These servers use the IP tables to implement NAT.

SNAT or Source network address translation router modifies the sender’s IP address. This service is mainly responsible for enabling the host with private addresses to connect to the public internet.

DNAT or Destination Network Address Translation, this NAT router modifies the address in IP packet headers. Open stack uses destination network address translation to route the packets to the open stack service from various instances. Open stack changes the packet’s destination address so they reach the network interface.

ONAT or one to one NAT is the service the NAT router maps a private IP addresses and public IP addresses. Open stack uses this mapping to implement floating IP addresses.

All the above-described technologies are used under the Open stack architecture. The core functions of Neutron include the functionality to provide connectivity to all the network instances from inside and outside of its network. The neutron allows programmatically configuring the network devices remotely and connectivity with the Virtual LANs.

VLAN DESCRIPTION AND SETUP IN NEUTRON

Virtual Network Interfaces, open stack uses the KVM and QEMU drivers for the virtualization platform in its NOVA configuration. When an instance is started for the first time, it is allotted a virtual port to each of its network interfaces.

KVM or kernel-based virtual machine is a virtualization platform used in Linux kernel. This platform acts as a hypervisor that holds and manages multiple virtual machine instances.

The kernel-based virtual machine creates a virtual interface known as tap interface. This interface exists on the computer node that is hosting that instance. Tap interface is associated with the guest instance at the network interface. In open stack the name of a tap interface corresponds to the Neutron port UUID, it is a unique identifier for the instance in which it is plugged into.

Virtual Network Switches

Neutron supports various types of physical and virtual switches. These switches include support for all type of Linux bridges and Open v Switch virtual switches. Switches and bridges are used interchangeably in Neutron networks.

A Linux bridge represents a virtual switch running on a host that connect multiple interfaces of that network. A bridge connects an interface of a physical device to one or more tap or virtual interfaces. These physical interfaces Ethernet interfaces or bonded interfaces. A virtual interface here includes both the Ethernet interfaces and tap. A Linux bridge supports multiple virtual or physical network interfaces.

Linux bridge works in 6 stages, the first stage consists of various Virtual machine instances that are connected to a Kernel virtual machine. This machine is connected to multiple tap interfaces that are connected to a provider bridge. This provider bridge is connected to the external Ethernet ports that are further connected to the Physical network switch.

Open v Switch or open virtual switch operated as a software-based switch. Open v switch uses virtual network bridges to forward packets. These switches are also responsible for implementing the flow rules on the forwarding packets between hosts. Neutron setups that use open v switch utilize at least three virtual switches or bridges. Neutron setups also include various other networking devices like provider integration and tunnel bridge. Virtual switches can be connected to the cross networks, similar to physical switch connections using cross-connect cables.

The biggest problem in manually configuring the switches is when the wrong combination is set up on a bridge, this leads to the bridging loops and caused issues with the network. That is why it is suggested in an open stack networks to prevent manually configuring the network to avoid bridging loops.

Neutron supports overlay networks. This is the technology that allows scaling across the cloud. This scaling does not modify the underlying physical infrastructure. This is accomplished using L2 in L3 overlay networking technology. There are two types of overlay networking technology GRE and VXLAN. These two technologies are discussed previously in this paper.

Neutron builds the point to point tunnels between all the network and computing nodes across the cloud network. Neutron uses the management interfaces or other dedicated interfaces for accordingly configuring this network.

Point to point tunnels described above is also known as the mesh networks. In a mesh network, every another host is connected to every host in its network. A cloud architecture consists of a controller that runs network services and three nodes for computing purposes.

GRE and Vxlan are created for users leveraging the ML2 plugin. In a gre or vxlan network, Unique ID is created that provides the data encapsulation, which ensures the data integrity. Every packet between the instances is encapsulated on one host and forwarded to the other through a point to point GRE or VXLAN tunnel. When the packet reaches the destination its header related to the tunnel are extracted and stripped to see the forwarding information from the connected bridge to the instance.

A raw header before the tunnel encapsulation consists of just two fields IP header and IP payload. During encapsulation, the packet is appended with the extra information of source and destination addresses of compute nodes. VXLAN and GRE data including a Virtual network interface or ID. Source and Destination address on the VM instances. Layer 4, contains TCP and UDP data.

Finally, the packet after tunnel encapsulation consists of four layers. The first layer consists of IP payload, the second layer is Inner IP header, the inner IP header identifies the original sender and the recipient. The third layer is Tunnel IP header and the last layer is outer IP header. The data inside the last outer IP header contains information about the tunnel endpoints. This data guides the neutron about the compute nodes any other nodes that are running the L3 and network node services such as DHCP.

As described above the GRE and VXLAN network traffic is encapsulated. So many physical network devices cannot be connected to this network. Neutron router solves this connectivity problem and supports connectivity between both the networks.

CONNECTIVITY ISSUES WHEN USING OVERLAY NETWORKS

When the above-described packets are added to the packet, sometimes these packets exceed the maximum transmission unit allowed. MTU is the maximum packet or frame size allowed, that can be sent over the network. This proves to be the important concern when the overlay networking technologies are used. Default maximum transmission unit size is 1500 bytes that exceed when VXLAN headers are used. Network communications experience strange issues on exceeding these limits. This leads to partial data failures over an SSH connection or failure in transmitting large payloads between instances.

This problem can be prevented by using lower MTUs of interfaces in a virtual machine instance, 1450 bytes in place of 1500 bytes. This will allow the VXLAN encapsulation overhead and will avoid the connectivity issues. The DHCP can be configured to allocate a lower MTU to instances during the DHCP lease offer.

**VLAN NETWORK TYPES SUPPORTED IN NEUTRON**

With the use of Modular layer 2 or ML2 plugin, neutron allows five types of VLAN networks. An ML2 plugin framework allows the open stack network to utilize the various layer 2 networking technologies simultaneously.

The first is a Local Network, this network is called so because this network is isolated from other nodes and networks. The instances connected to this network can communicate only with the other hosts that are connected to the same network. Those hosts may be inside the same network from where the request is initiated or may reside in any other node.

The second is a Flat network, in this type of network VLAN Tagging does not takes place. In some types of flat network configuration, the instances can be on the same network as the host machine. No connection segregation takes place in this network.

The third type of network is VLAN network, these network leverages 802.1q tagging to separate the network traffic. 802.1 q is the networking standard that allows VLANs on an Ethernet Network, this also allows VLAN tagging for the Ethernet frames.

The fourth Type is VXLAN network, these networks are used to encapsulate packets over a traditional network to solve the switching infrastructure limitations. VXLAN uses a USID also known as Unique segmentation ID to segregate traffic from other VXLAN networks. In vxlan network, the traffic is encapsulated when sent from one instance to another with the help of USID and sent over the layer 3 network. VXLAN uses UDP for the layer 3 transmissions, at layer 3 the packet is stripped and forwarded to the instance.

Fifth is the GRE Network, a gre network is similar to a vxlan network. Similar to a vxlan network gre sends a traffic from one instance to another after encapsulating the packet over the layer 3 network. In gre also a unique segmentation Id is used to differentiate traffic from other GRE networks. Just the difference is gre used IP protocol instead of a UDP protocol for traffic routing.

**PLUGIN AND DRIVER IN NEUTRON NETWORKING**

For maintaining connectivity, the neutron network uses plugins, drivers, and agents. These devices maintain the connectivity to and from the instances. ML2 plugin as described above can leverage multiple layer two technology simultaneously using the mechanism drivers. The two drivers used by neutron are Linux Bridge and Open v Switch.

Linux Bridge is known for its ease of trouble shooting and dependability, but Linux bridge lacks the support for some neutron features such as Distributed virtual routers. Linux bridge supports five types of Interfaces Tap interface, physical interface, VLAN interface, VXLAN interface, Linux Bridges.

Hypervisor creates and uses a tap interface, such as KVM or QEMU. Tap interfaces are used to connect the host to a guest operating system in a virtual machine instance. Guest operating system receives the Ethernet frames sent to the tap devices on the host. Guest operating system after receiving these packets injects them into the host network stack.

A physical interface is a representation of an interface that id pluggable into the physical network hardware. The physical interface is generally labeled as etho0 eth1 and so on depending on the host operating system.

A VLAN interface facilitates VLAN tagging for the Ethernet frames in Linux. A VLAN interface ca be created using iproute2 command. A VLAN interface is generally labeled as ethx <vlan>, this interface is associated with its respective interface. A VXLAN interface is responsible for encapsulating and forwarding the traffic based on the parameters such as vxlan network identifier also known as VNI and vxlan tunnel end point.

OPEN vSWITCH (OVS) DRIVER is a software based switch that allows virtual network bridges for forwarding packets between hosts. Neutron leverages only a subset of Open v switch features. The three main components are described below,

The kernel module, this module of open v switch is similar to ASIC on a hardware switch. This module is the data plane of open v switch that is responsible for all the packet processing tasks.

The vSwitch daemon, this is a Linux daemon process that runs in user space on every physical host. This process manages how the kernel module described above will be programmed.

The Database server, open v switch uses a local database on every host. This is also called open v switch database server (OVSDB). This module maintains the configuration of virtual switches.

Both OVS and Linux bridge have their own disadvantages and advantages. OVS is compatible with modern switches in chipsets, While Linux bridge has more throughput as compared to the OVS. OVS provides encapsulation and compression of packets that gives more kernel instability as compared to Linux bridge.

In practical usages when one is looking for the reliability and stability, easier to trouble-shoot, maximum throughput Linux Bridge is the best choice while If one wants an admin friendly driver than OVS VLAN proves to be the best choice.

**CONCLUSION**

This paper gives a description about how Cloud computing is used to manage devices and data remotely over the internet. This paper describes how a large network consisting of millions of networking devices is managed. The first part of this paper narrates the need of cloud computing and therefore the need of open stack and Neutron in it.

In Cloud computing, the data is stored on the internet instead of Computer’s hard drive. Cloud provides these services in various forms, for example, Software as a service or SaaS, this service allows the user to access the applications over the internet. Infrastructure as a service or IaaS provides the developers with the hardware requirements remotely. Platform as a service or PaaS facilitates the user with a remote machine, allowing the user to leverage its processor, storage devices, and other software development requirements.

OpenStack Neutron is a software defined networking project that is responsible for delivering networking as a service (NaaS). Neutron has replaced the traditional hardware appliances and has virtualized everything. Neutron is an independent service that is often deployed to run several services across a number of nodes. The main components of Neutron Are Neutron server and plugin agent. Neutron provides the cloud users application programming interfaces to build networking topologies. It is also responsible for configuring the policies. Open stack uses virtual machines, and various machines can be connected together, for this, the open stack uses virtual LAN technology.

This paper also gives a description about how VLAN trunk is used with the open stack neutron and how the demand of growing network is fully filled with the use of SDN controllers.

In a Neutron network, many plugins do not allow VLAN tagged traffic to pass through the network. VLANs are required for many networks virtualization functions. This paper briefs the process about how the VLAN tagged traffic can be delivered to other hosts. The most common requirement of Virtual machines used over NFV is that they need to communicate over L2 channels. VLAN tags are used for segregating these channels. In a neutron network, many networks are created using plugins, there is no direct meaning of 'network'. The solutions to this problem include decomposing trunks into smaller networks from its individual Virtual LANs. During the network creation, the user can request for a transparent VLAN trunk. VLAN tagging also serves as encapsulation, and hence not require to be firewalled. Other than this a complementary port based VLAN is also available, this method permits a set of networks to a pre-defined port as a VLAN trunk. A SDN controller is a software application that is used to manage the network logically. SDN is based on open flow protocol. In this, the traditional hardware devices are replaced with virtual devices, also called open v switch or software switches.

In a traditional network with hardware routers and switches also known as the flat network were used. The negative aspects of this flat network include the single broadcast domain. To solve this problem Virtual Local area network or VLANs are used. VLAN helps in creating smaller collision domains that are easily manageable. In flat networks, they were more vulnerable to the security attacks because the malicious user could see the whole network. But this segregation due to VLAN solved this security threat problem.

VLAN segregation is used in a web hosting network where one VLAN is dedicated to a web server inside the Demilitarized zone and another for the database. A routing device is required for these devices to communicate.

VLAN TAGGING, in a switch there are two ports, trunk ports, and the access ports. Switch ports are configured such that a single VLAN can communicate with other ports. While, the trunk ports are responsible for connecting multiple VLANs. The switch adds an Ethernet tag to the frame consisting VLAN ID.

All hosts in a network if connected to an access port inside the same VLAN, all the hosts communicate with each other on the same network. VLANS helps in this scenario, the virtual network allows to separate hosts and instance traffic, it allows to create multiple networks similar to inside and outside network segregation. Neutron allows multiple users or multiple tenant networks. These multiple user networks are build using VLAN IDs that are mapped to the real Virtual local area networks. Neutron also provides the facility to use multiple VLAN IDs over a single open virtual switch, this single switch can be shared by multiple providers and users. This shared virtual network also allows sharing dedicated servers, load balancers, firewalls, and other networking peripherals.

The open stack is the future of cloud computing, it will advance and will provide the facility to integrate containers. Open stack is being adopted by all the tech giants as their cloud toolkit, open stack due to its wide architecture and module containing Nova (the computing unit), Swift (Storage system for objects and files), Cinder (Block storage component), Neutron (Networking component), Horizon (Dashboard of open stack), Keystone (Identity manager), Glance (Virtual machine image services), Ceilometer (Telemetry services), Heat (Orchestration of various parts of open stack). All these units together work best for any type of Industry as its cloud computing toolkit.

**Reference:**

1. http://docs.openstack.org/

2. https://wiki.openstack.org/wiki/Neutron

3. http://docs.openstack.org/developer/neutron/

4. http://www.sparkmycloud.com/blog/openstack-vlan-networking-overview/

5. “Learning OpenStack Networking” by James Denton

6. http://www.s3it.uzh.ch/blog/openstack-neutron-vlan/

7. https://developer.rackspace.com/blog/neutron-networking-vlan-provider-networks/

8. https://specs.openstack.org/openstack/neutron-specs/specs/kilo/nfv-vlan-trunks.html

9. https://www.openstack.org/assets/presentation-media/OpenStackOverVxlan.pdf