White paper

OpenStack networking: The challenges and a new hope

nuagenetworks
From Nokia

Abstract

Despite the promise of highly automated and on-demand networking, the reality is that networking in OpenStack can be complex and presents a new set of challenges. Therefore, it is important for anyone considering a move to OpenStack to choose the right networking solution for their OpenStack deployment: one that fulfills the specific networking needs and application requirements.

In this white paper we examine the current networking model in OpenStack and highlight the limitations of standard networking provided by OpenStack Neutron (ML2 OVS). Then, we explore two networking solutions designed to address these limitations: Open Virtual Networking (OVN) and the Nuage Networks™ Virtualized Services Platform (VSP). Our goal is to share our findings from working with hundreds of enterprises and communications service providers as they embarked on their IT transformation journey and deployed OpenStack in production.

Contents

2 Abstract

4 Introduction

4 OpenStack networking: Neutron (OVS)

4 Implementation

6 Advantages

6 Limitations

6 OpenStack networking: Neutron (OVN)

7 Implementation

7 Advantages

7 Limitations

8 OpenStack networking: Neutron (Nuage Networks VSP Plugin)

8 Nuage Networks VSP components

9 Nuage Networks VSP as a Neutron backend

12 Comparison of Neutron, OVN and Nuage Networks VSP

13 Conclusion

14 Abbreviations

Introduction

OpenStack is a scalable, open cloud-computing platform for private, public and hybrid clouds. Started as an open-source project in 2010, it has grown exponentially over the past few years. OpenStack is now a mature, de facto standard foundation for building any type of cloud and is suitable for the vast majority of enterprise and Telco cloud use cases.

OpenStack is largely used by various organizations that also actively contribute to its development. It consists of multiple community-based projects that are used to implement distributed compute, storage and networking. Neutron, one of these projects, is meant to provide network connectivity as a service inside the OpenStack cloud.

This white paper examines how networking in OpenStack is implemented today and explores the various networking alternatives available. We start by looking at Neutron's technical implementation details and identifying the limitations of this implementation. Next, we describe the common limitations of using Neutron with the standard Modular Layer 2 ML2 Open vSwitch (ML2 OVS) backend.

We then examine the Neutron backend solution proposed by the Open Virtual Networking (OVN) project, which has been created to address some of the ML2 OVS limitations.

Finally, we explore the Nuage Networks Virtualized Service Platform (VSP) and how it can be used as a better and more comprehensive alternative to OVS and OVN.

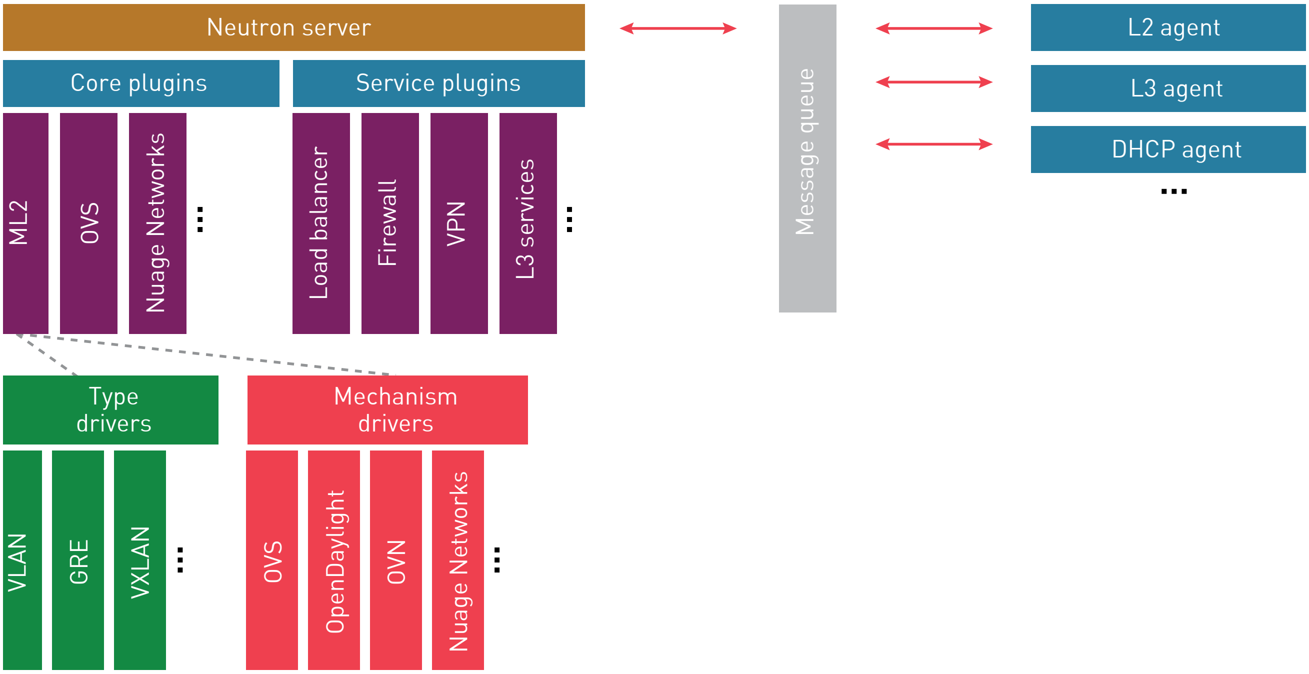
OpenStack networking: Neutron (OVS)

Neutron is the project code name for the OpenStack Networking service. It handles the creation and management of the networking infrastructure for cloud workloads. Neutron provides network connectivity from the moment of workload creation until its destruction.

Implementation

Figure 1 shows the Neutron ML2 architecture.

FIGURE 1. Neutron ML2 architecture



The ML2 plugin is a framework meant to replace the existing Neutron monolithic architecture and greatly simplify the effort for developing new plugins. It provides two kinds of drivers:

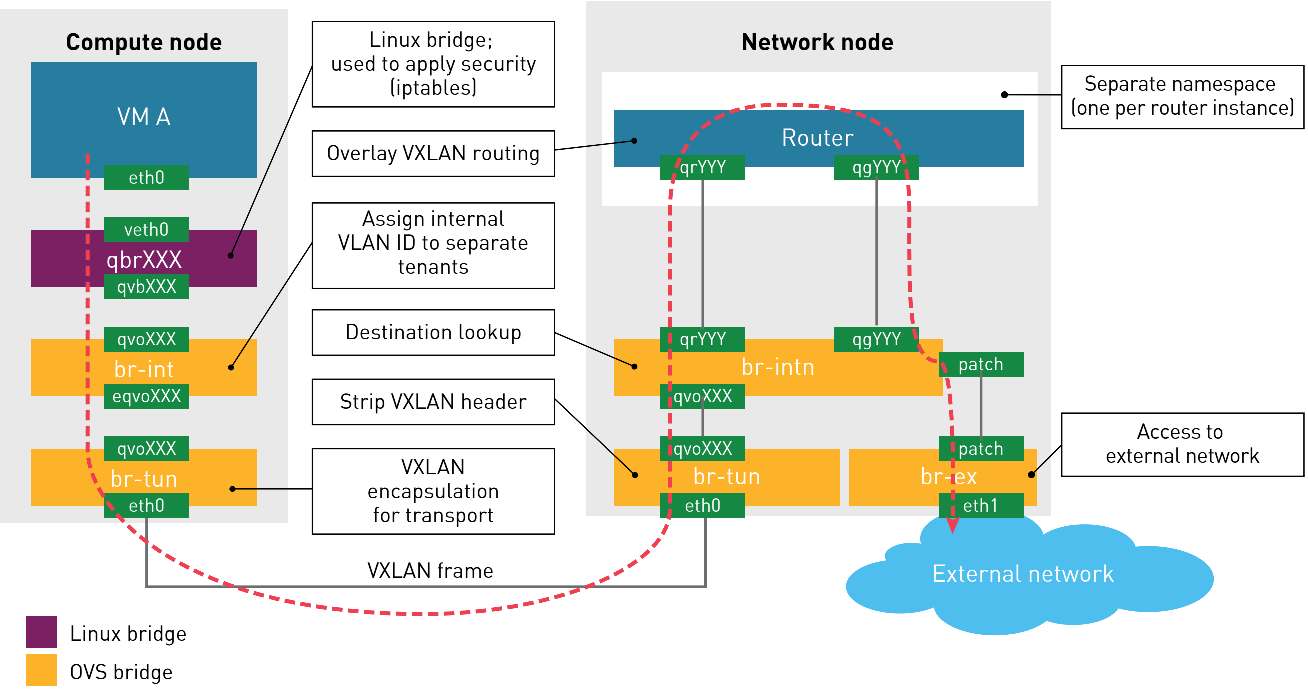
- **Type drivers:** Maintain any needed type-specific network state, and perform provider network validation and tenant network allocation. The ML2 plugin includes drivers for the local, flat, VLAN, GRE and VXLAN network types.
- **Mechanism drivers:** Define how to access a network of a particular type.

The most common implementation of OpenStack networking today uses the VXLAN type driver and OVS mechanism driver.

The Neutron server communicates with a series of agents (L2, L3, DHCP, etc.) using a remote procedure call (RPC) over the RabbitMQ bus. The roles of these agents are: programming the OVS (L2), enforcing the security group rules (L2), and managing router and external connectivity (L3).

Figure 2 takes a close look at how Neutron ML2 OVS is implemented and shows the different bridges inside a compute node and a network node.

FIGURE 2. Data path for external communication



In a communication session between Virtual Machine A (VM A) and an external network, the data packet traverses a multitude of interfaces, bridges and namespaces until it finally reaches the external network. Each of these elements has its own role.

- The qbrXXX Linux bridge is there only to provide a way to apply iptables rules on the VM interface.
- The integration OVS bridge (br-int) is used for VM traffic isolation between tenants. It implements the isolation by adding locally significant VLAN headers.
- The tunneling OVS bridge (br-tun) deals with transport encapsulation. It adds a VXLAN header so the packet can be sent to the right supervisor.
- Each router instance is put in a separate Linux namespace. This is done so it can deal with multi-tenancy and overlapping IP spaces.
- The external bridge (br-ex) is used to send the traffic to the external network.

Advantages

This design has its advantages.

- It uses Linux native mechanisms to apply security and implement multi-tenancy.
- In case of a network node outage, the "control plane" continues to be operational because the "data-path learning" method is used.

Limitations

However, the Neutron architecture also has some drawbacks.

- The multitude and heterogeneity of network bridges, interfaces and namespaces
  - Leads to degraded performance because of the increased amount of inter-process packet handling
- The absence of a proper, separate control plane
  - All control plane information uses RPC over the RabbitMQ bus. This is a big scalability issue because with the increasing number of events on the platform, the bus gets congested.
- The centralization of the network node
  - All routed traffic and traffic that has gone through Network Address Translation (NAT) needs to go through the network node. As a result, the network node becomes a bottleneck and single point of failure.
- The current OpenvSwitch implementation does not support VXLAN multicast flooding and uses unicast source replication instead.
  - This leads to unicast flooding of Broadcast and Unknown Multicast (BUM) traffic to all hypervisors.
  - Address Resolution Protocol (ARP) requests are processed by tunnel and integration bridges of all hosts before the requests are discarded.

Some of these limitations have been addressed with the addition of several improvements. The absence of distributed routing can be fixed by activating a distributed virtual router (DVR)[1]. The unicast source-replication problem can be fixed by installing an ARP responder[2] and L2 population drivers[3]. However, in large deployments this can dramatically increase the size of the flow tables.

Overall, the limitations can lead to major scalability issues. As a result, what often starts as a small OpenStack platform becomes slow and unresponsive as it gets larger and widely used.

OpenStack networking: Neutron (OVN)

As discussed in the previous section, the standard OpenStack Neutron ML2 OVS implementation has some serious scalability and performance issues. The open-source community tried to address the issues in projects, including L2Population, ARP responder and DVR, but this requires additional installation and configuration. Also, these community projects are only a collection of separate independent packages that have their own set of dependencies and repositories. As a result, these projects need to be maintained independently and their attempt to solve some challenges has resulted in the introduction of new ones.

Open Virtual Networking (OVN) is a project meant to address Neutron scalability limitations by replacing the entire Neutron backend with an OVS-only software-defined networking (SDN) controller. Unlike the classic Neutron OVS implementation, OVN supports distributed virtual routing, DHCP, NAT and security groups on the same OVS bridge without employing additional namespaces or bridges. This greatly reduces the delays caused by inter-process packet handling, and more important, enables flows to benefit from kernel fast-path switching.

[1] Distributed Virtual Router – a Neutron project (https://wiki.openstack.org/wiki/Neutron/DVR)

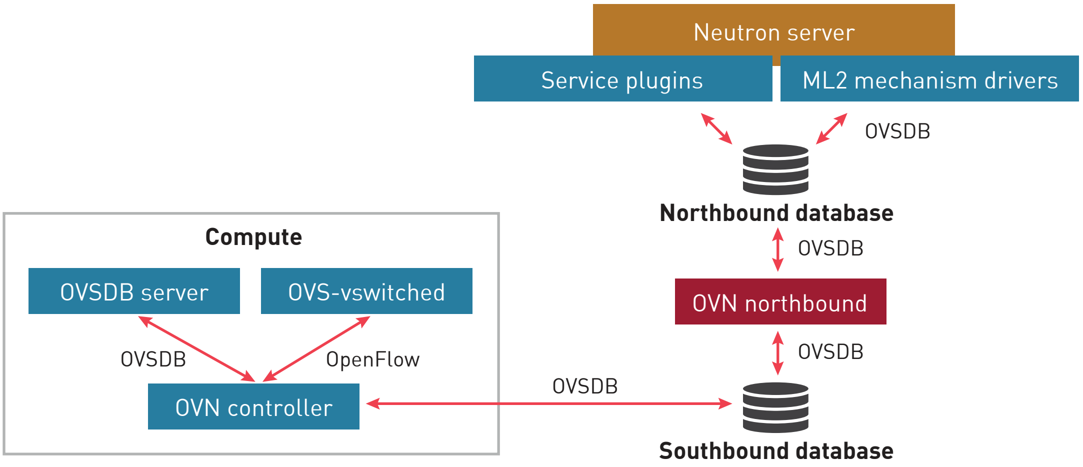
[2] https://assafmuller.com/2014/05/21/ovs-arp-responder-theory-and-practice/

[3] https://wiki.openstack.org/wiki/L2population

Implementation

Figure 3 shows the OVN architecture: the interaction with the OpenStack Neutron server as well as the internal communication among the various components.

FIGURE 3. OVN architecture



The northbound database is populated by Neutron's OVN ML2 plugin with logical network constructs (distributed routers, logical switches, ports, etc.). This data is then imported into the southbound database by the OVN northd daemon in the form of logical flows. These logical flows are fed to the OVN controllers on each compute node so they can be transformed into OpenFlow forwarding instructions for the OVS forwarding tables.

Unlike classic Neutron OVS, OVN has a "proper" control plane. It does not use RPC over the RabbitMQ bus, but instead uses OVSDB (Open vSwitch Database Management Protocol) as a configuration and provisioning protocol. The security is applied directly at the OVS port, using the kernel conntrack module: there is no longer any need for separate Linux bridges.

Advantages

Overall, OVN has solved most of Neutron's limitations. The performance at scale has increased greatly[4]. The data path is much simpler, with much less tromboning between the different bridges and namespaces.

Limitations

Despite OVN's advantages over Neutron, several challenges still exist.

Limited VXLAN support

The OVN standard uses Generic Network Virtualization Encapsulation (GENEVE) as a transport protocol between hypervisors and supports VXLAN only for hypervisor-to-gateway communications. GENEVE is a requirement because it provides the ability to include additional metadata, which OVN uses to pass additional IDs (source/destination port IDs) in the packet. However, GENEVE is less popular than VXLAN. As a result, there might be some limitations to this implementation. For example, with GENEVE we cannot benefit from VXLAN offloading capabilities present on some network interface cards (NICs).

Centralized gateway capability

The gateway function is the ability to transfer traffic between overlay and underlay networks.

Although OVN provides distributed virtual routing capabilities, the gateway function is still centralized. This means that, like the network node in the Neutron implementation, a centralized gateway can be a bottleneck.

[4] https://blog.russellbryant.net/2016/12/19/comparing-openstack-neutron-ml2ovs-and-ovn-control-plane/

A separate control plane for every platform

Although having a proper control plane is a big improvement over the RabbitMQ bus in the Neutron implementation, the current OVN implementation supports only one separate control plane per platform/ Cloud Management System (CMS). This may be a serious limitation for heterogeneous or multi-cloud platforms; for example, an OpenStack platform and a Kubernetes cluster cannot be connected by the same OVN system.

Lack of control plane and data plane integration with existing gateways

One important part of designing an SDN solution for a cloud management system is its integration with the existing infrastructure. This can be useful in the following cases:

- Migration from legacy infrastructure to OpenStack cloud or vice versa
- Use of hybrid clouds

OVN lacks the control plane and data plane integration with existing gateways that is required for these use cases. One way to achieve this integration is by using the standardized L3 VPN protocols for seamless extension of L3 VPN from these third-party gateways into OpenStack routers or networks.

OpenStack networking: Neutron (Nuage Networks VSP Plugin)

The Nuage Networks Virtualized Services Platform (VSP) is a comprehensive SDN platform that makes the network as readily consumable as compute resources across the datacenter, enterprise WAN and public cloud providers. It does this by ensuring rapid and efficient delivery of highly customizable application services in and across multi-tenanted datacenters.

As an SDN platform for private cloud network automation in the enterprise datacenter, the Nuage Networks VSP enables the deployment of massively scalable cloud-based services with the agility and performance demanded by dynamic application environments. The Nuage Networks VSP creates a datacenter network that automatically establishes connectivity between compute resources upon creation. Leveraging programmable business logic and a powerful policy engine, the Nuage Networks VSP provides an open and highly responsive solution that scales to meet the stringent needs of massive multi-tenant datacenters. This software-based solution can be deployed over existing datacenter network fabrics and to public cloud providers.

Nuage Networks VSP components

Virtualized Services Directory

The Virtualized Services Directory (VSD) is a programmable policy and analytics engine. It provides a flexible and hierarchical network policy framework that enables IT administrators to define and enforce resource policies in a user-friendly manner.

The VSD contains a multi-tenanted service directory that supports role-based administration of users, compute and network resources. It also manages network resource assignments such as IP and MAC addresses.

For service assurance, the VSD allows the definition of sophisticated statistics rules such as collection frequencies, rolling averages and samples, as well as Threshold Crossing Alerts (TCAs). When a TCA occurs, it triggers an event that can be exported to external systems through a generic messaging bus. Statistics are aggregated over hours, days and months, and are stored to facilitate data mining and performance reporting.

The VSD can be deployed as a stand-alone or clustered solution depending on scaling needs.

Virtualized Services Controller

The Virtualized Services Controller (VSC) is the industry’s most powerful and scalable SDN controller. It functions as the robust network control plane for datacenters, maintaining a full view of per-tenant network and service topologies.

Through the VSC, virtual routing and switching constructs are established to program the network-forwarding plane using the OpenFlow™ protocol. Multiple VSC instances can be federated within and across datacenters by leveraging Multiprotocol — Border Gateway Protocol (MP-BGP) — a proven and highly scalable network technology.

Virtual Routing and Switching

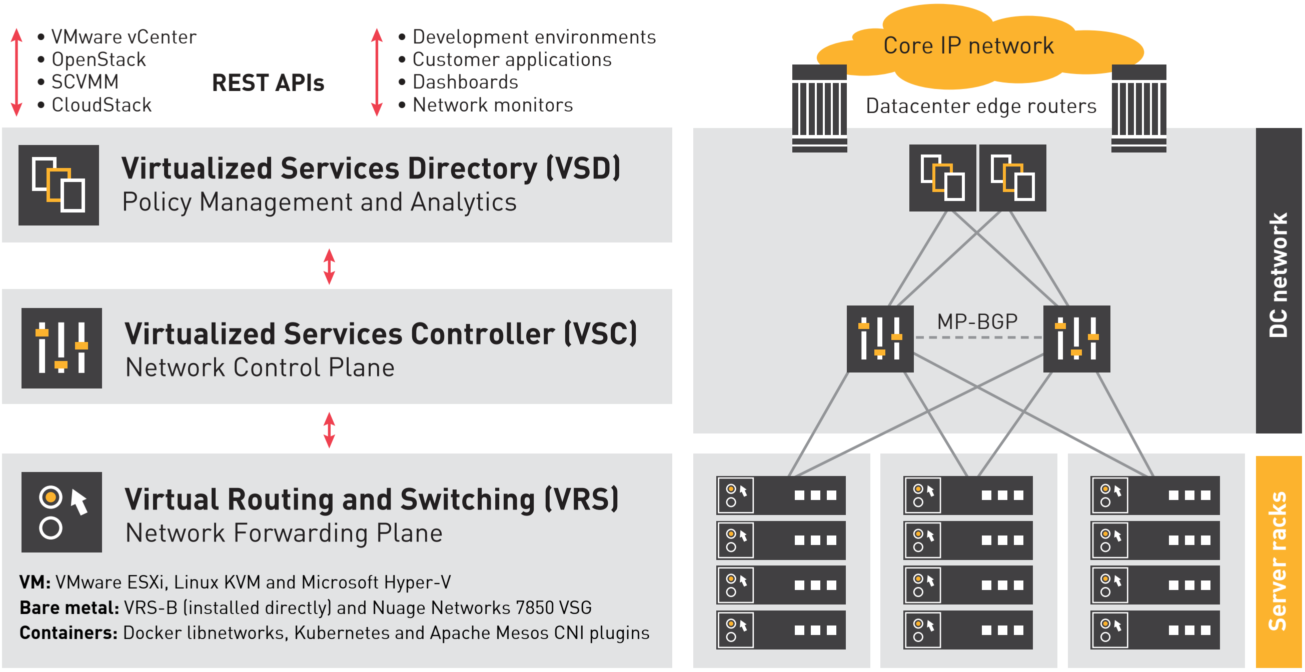
The Virtual Routing and Switching (VRS) component is an enhanced OVS implementation that constitutes the network forwarding plane. It encapsulates and de-encapsulates user traffic, enforcing L2-L4 traffic policies as defined by the VSD.

VRS tracks VM creation, migration and deletion events to dynamically adjust network connectivity. VRS supports multiple hypervisors and container-ready platforms in virtualized server environments. It also operates as a gateway for bare-metal servers or service appliances.

The Nuage Networks VSP also includes a physical network appliance, the Nuage Networks 7850 Virtualized Services Gateway (VSG). The 7850 VSGs serve as overlay network tunnel endpoints where needed, such as for integration with physical servers, as well as working with leading networking vendors' top-of-rack switches for VXLAN termination. To support bare-metal applications, a software VRS-B (bare metal) may also be deployed directly on the physical server, avoiding the need for a VXLAN-compliant top-of-rack switch.

Figure 4 shows the VSP architecture.

FIGURE 4. Nuage Networks VSP architecture


• VMware vCenter
• OpenStack
• SCVMM
• CloudStack
REST APIs
• Development environments
• Customer applications
• Dashboards
• Network monitors
Virtualized Services Directory (VSD)
Policy Management and Analytics
Virtualized Services Controller (VSC)
Network Control Plane
Virtual Routing and Switching (VRS)
Network Forwarding Plane
VM: VMware ESXi, Linux KVM and Microsoft Hyper-V
Bare metal: VRS-B (installed directly) and Nuage Networks 7850 VSG
Containers: Docker libnetworks, Kubernetes and Apache Mesos CNI plugins
Core IP network
Datacenter edge routers
MP-BGP
DC network
Server racks

Nuage Networks VSP as a Neutron backend

An alternative to the previously discussed solutions (Neutron ML2 OVS and Neutron OVN) is to use the Nuage Networks VSP as a Neutron backend.

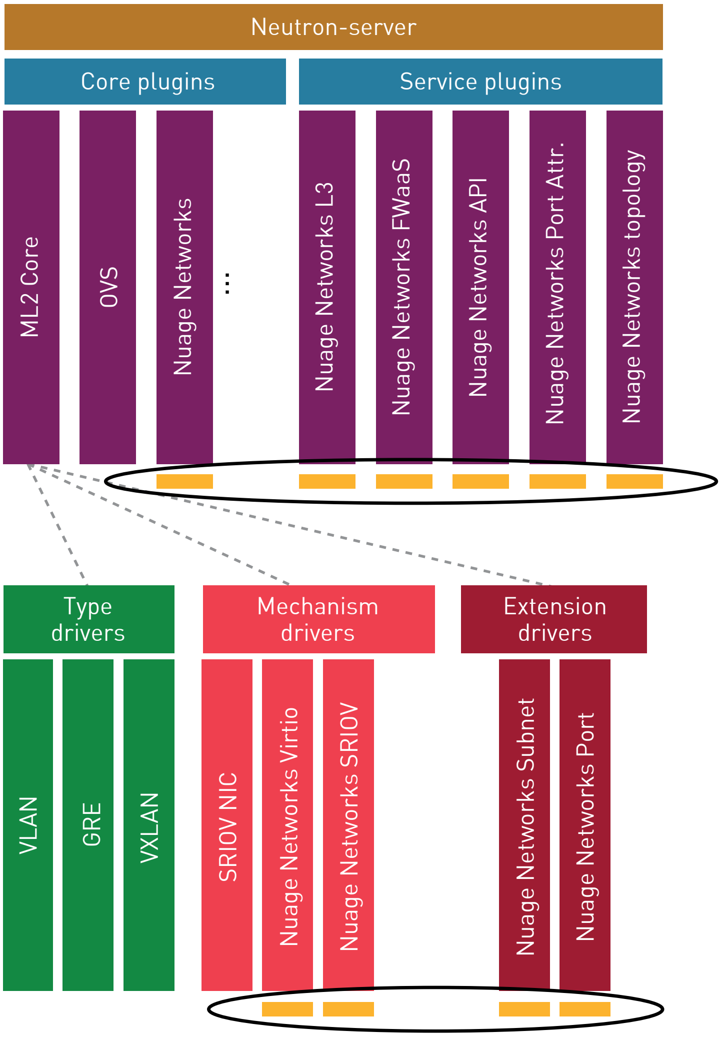
Implementation

As in the previous solutions, in this implementation the Nuage Networks VRS is based on OVS for data forwarding. However, the Nuage Networks VRS version of OVS has a slightly modified user-space agent, which allows it to add some improvements and does not require any OVS kernel modifications.

The Nuage Networks 5.3.3 release supports OpenStack releases Pike and Queens. Nuage Networks has developed an ML2 plugin, which allows it to connect to the Neutron server and start operating as a networking backend.

As shown in the plugin architecture on Figure 5, the Nuage Networks VSP replaces the "message bus and agents" elements of the Neutron ML2 architecture from Figure 1 with its own service plugins as well as extension drivers. In Figure 5, the Nuage Networks components are shown in the ovals.

FIGURE 5. Neutron ML2 Nuage Networks OpenStack architecture



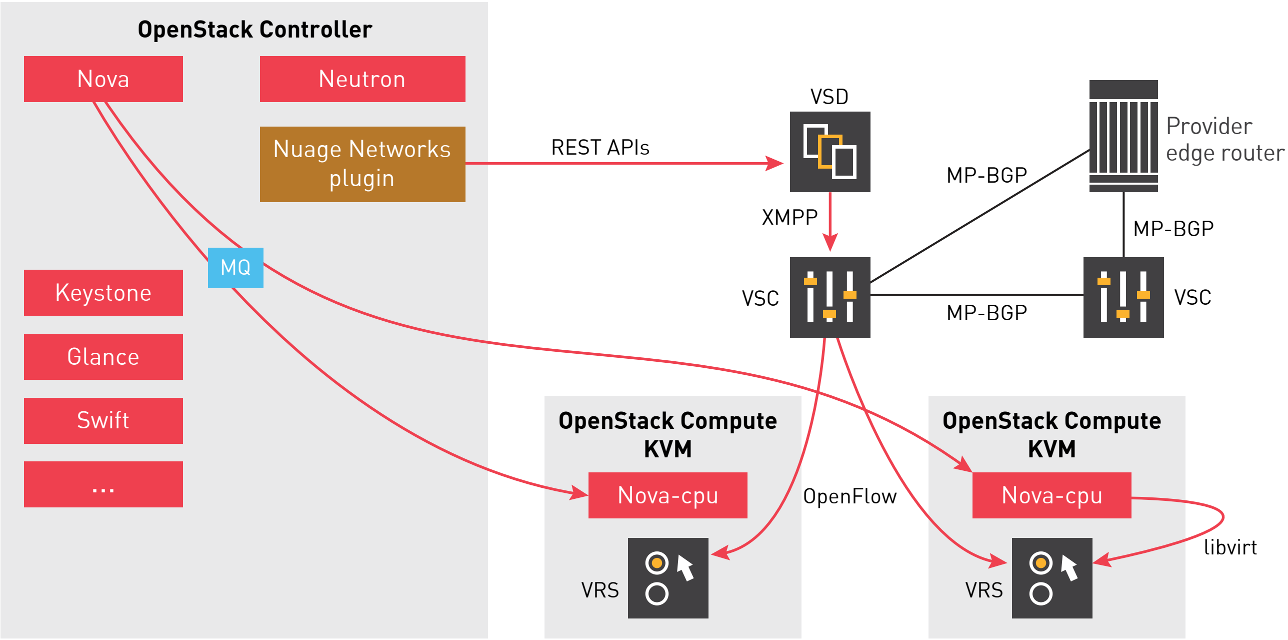
Overall, the Nuage Networks OpenStack integration consists of the following components:

- Nuage Neutron plugin and Neutron Client extension
- Heat resource plugin
- Horizon extension
- The metadata agent

Figure 6 shows the interaction between the various OpenStack and Nuage components. There are a few things to point out here.

- The OpenStack plugin resides on the OpenStack controller and is the component that transforms networking configuration instructions from Neutron into REST API calls to the VSD, to provision network elements.
- As previously stated, the whole Advanced Message Queuing Protocol (AMQP) bus has been replaced by Nuage Networks VSP components.
- The control plane is managed by the Nuage Networks VSCs, which program the VRS modules using the OpenFlow protocol. The VSCs translate network constructs from the Nuage Networks VSD into OVS flows in each Nuage Networks VRS.
- The VSP uses a single OVS bridge, where it implements secure network connectivity.
- The control plane scales using MP-BGP. In this example we have two VSCs–but that is not a requirement.
- The Nuage Networks VSP is agent-less: it has no L2, L3 or DHCP agents.

FIGURE 6. OpenStack Nuage Networks architecture datapath



Improvements over Neutron ML2 OVS

Compared to the issues with Neutron ML2 OVS, the VSP implementation of OpenStack networking has the following improvements:

- A proper control plane that scales horizontally using BGP
- The capability to interface with all provider edge/customer edge routers that are BGP capable, to provide seamless connectivity with legacy datacenter networks
- Use of a single namespace and bridge throughout the entire data plane:
  - Flows can benefit from kernel fast-path switching
  - Absence of inter-process packet tromboning
- A VXLAN-based datapath enables leveraging VXLAN offload on NICs

Last but not least, after replacing the Neutron backend with the Nuage Networks VSP, we have noticed significant performance improvements at scale. An extensive presentation from OpenStack Summit can be seen at: https://www.openstack.org/videos/paris-2014/nuage-networks-pets-cattle-and-herding-dogs

Improvements over OVN

In addition to all the issues solved by adding the Nuage Networks VSP to OpenStack, this implementation includes the following improvements over OVN.

- A single platform for everything

The Nuage Networks VSP supports multiple workloads, hypervisors and cloud management systems:

- VMware vSphere, OpenStack, Apache CloudStack, Microsoft Hyper-V CMS, Microsoft System Center Virtual Machine Manager (SCVMM)
- VMware ESXi, Linux KVM, Microsoft Hyper-V hypervisors
- VMs, containers and bare-metal workloads

All of this can have a single control plane and be managed from a single interface.

- Unlike OVN, the Nuage Networks VSP does not have a single gateway node. Local breakout to underlay/the internet is possible, and distributed NAT is supported. Each node's VRS instance can act as a gateway between overlay and underlay. As a result, there is no longer a bottleneck issue.
- The Nuage Networks VSP SDN controller (Virtualized Services Controller [VSC]) is based on the Nokia Service Router Operating System (SR OS). As a result, the VSC supports all existing protocols to allow interconnection with legacy datacenter equipment, which ensures a proper integration with the existing environment.

Comparison of Neutron, OVN and Nuage Networks VSP

The following table lists the main issues identified in Neutron ML2 OVS and their status in the OVN and Nuage Networks VSP implementations.

Issue	Neutron ML2/OVS	OVN	Nuage Networks VSP
Complex data path (multiple bridges/ interfaces) in OVS	No resolution	Solved; only one OVS bridge; default namespace	Solved; only one OVS Bridge, default namespace
No dedicated control plane	No resolution	Solved through addition of a control plane	Solved through addition of a control plane
Centralized gateway node/NAT	No resolution	No resolution	Solved through local breakout capability with NAT support
BUM flooding	Unicast source replication/ data-path learning	Unicast source replication/ control-plane learning	Unicast source replication/ control-plane learning
VXLAN support for transport	Supported	Only supported for third-party VTEPs	Supported
Support for additional CMS under the same control plane	Not supported	Not supported	Supported (VMware, containers, Hyper-V, etc.)

Conclusion

OpenStack has grown to be a very complex ecosystem. Keeping the network operational in large OpenStack deployments can be a challenge. Therefore, it is vital to correctly identify your requirements and priorities in a network platform.

It is true that the tendency to automate everything keeps imposing itself, but automation can fail. And when this happens, it is important not to forget that humans are the ones to take corrective action.

Today, a vast choice of networking solutions exist. Enterprises can choose between open-source or commercial networking software. When choosing a network solution for an OpenStack platform, there are several questions to ask yourself.

- How big is your platform today?
- How fast will this platform grow in the future?
- What changes will the network need to support? For example, connection with other clouds/datacenters or bare-metal workloads?
- Will your cloud become heterogeneous in the future (with multiple cloud platforms, VMware, Microsoft Hyper-V, Kubernetes clusters, etc.)?
- Is hybrid cloud an option (burst into the public cloud, Amazon Web Services (AWS), Microsoft Azure, etc.)?
- Will this platform be created from scratch or as a result of migration from a legacy platform?
- Last, but not least: Who will manage this networking solution?

While commercial solutions are not free, it is important to remember that "open-source" also does not mean "free". Without the proper expertise, poorly maintained open-source solutions will eventually be very expensive.

Nuage Networks from Nokia is a comprehensive multi-tenant solution, offering networking connection for hybrid environments. The solution can be deployed on any type of hypervisor (VMware ESXi, KVM, Microsoft Hyper-V), supports the majority of cloud management platforms (VMware vSphere, OpenStack, Apache CloudStack, Microsoft Hyper-V CMS, Microsoft SCVMM) while interconnecting any type of workload (VMs, containers or bare-metal servers). In addition, Nuage Networks' customers can benefit from the 24x7 support of Nokia support teams around the world.

For more information about the Nuage Networks VSP, visit http://www.nuagenetworks.net/

Abbreviations

API	application programming interface
ARP	Address Resolution Protocol
BUM	broadcast and unknown multicast
CMS	Cloud Management System
DHCP	Dynamic Host Configuration Protocol
DVR	distributed virtual router
GENEVE	Generic Network Virtualization Encapsulation
GRE	Generic Route Encapsulation
KVM	Kernel-based Virtual Machine
LAN	local area network
MAC	media access control
ML2	Modular Layer 2
MP-BGP	Multiprotocol Border Gateway Protocol
NAT	network address translation
NIC	network interface card
OVN	Open Virtual Networking
OVS	Open vSwitch
OVSDB	Open vSwitch Database Management Protocol
RPC	remote procedure call
SCVMM	System Center Virtual Machine Manager
SDN	software-defined network/networking
SR-IOV	Single Root I/O Virtualization
TCA	Threshold Crossing Alerts
VM	virtual machine
VLAN	virtual LAN
VNS	Virtualized Network Services
VPN	virtual private network
VSC	Virtualized Services Controller
VSD	Virtualized Services Directory
VSG	Virtualized Services Gateway
VSP	Virtualized Services Platform
VTEP	VXLAN tunnel endpoint
VXLAN	Virtualized eXtensible LAN
WAN	wide area network

nuagenetworks
From Nokia

Document code: SR1810030025EN (November) CID205803