ARISTA TECHNICAL BULLETIN

# OPENSTACK NEUTRON INTEGRATION

## INSIDE



**Arista EOS® has extensive integration with the OpenStack Neutron project, giving customers a powerful network platform on which to run OpenStack deployments. By leveraging the Arista ML2 driver and Layer 3 service plugin, operators can automatically provision tenant networks across the physical infrastructure. This combination provides a high performance OpenStack networking environment over VLAN and VXLAN based fabrics, and enhanced visibility into how the virtual tenant networks map onto the physical infrastructure.**

## OVERVIEW

OpenStack is a leading open source solution for both public and private cloud deployments. The OpenStack solution is composed of a number of individual projects to address various components of a cloud. This bulletin focuses on the Neutron project and the points of integration between Neutron and Arista EOS. By leveraging the programmability of EOS and the open APIs of Neutron, customers are able to deploy a network infrastructure that is automated and orchestrated through OpenStack APIs or dashboards.

ARISTA

## ARISTA MODULAR LAYER 2 (ML2) DRIVER FOR VLAN

End-to-end automated provisioning of tenant networks requires configuration of devices from multiple vendors, such as configuring both Open vSwitch (OVS) on a hypervisor and an Arista top of rack (ToR) switch. This has led to the development of the Neutron Modular Layer 2 (ML2) plugin, allowing multiple Layer 2 drivers to be used in a single plugin.

The Neutron ML2 Plugin separates the decision of what resources to allocate to a given tenant network (done by a "type driver") from how that information is then provisioned across the virtual and physical infrastructure (done by "mechanism drivers"). Multiple mechanism drivers can be registered in parallel for different parts of your infrastructure. For example, one mechanism driver can be used for a virtual switch like OVS, while another can manage your Arista physical network infrastructure. The ML2 plugin architecture allows mechanism drivers to be changed independently, or for new mechanism drivers to be added in the future, as the needs of the network change. Arista contributed to the design and implementation of the mechanism driver infrastructure within the ML2 plugin, as well as wrote an Arista mechanism driver for automating the provisioning of an Arista physical network.

The Arista ML2 mechanism driver enables Neutron to automate VLAN provisioning on Arista switches. Through LLDP, the switches are aware of what compute nodes are connected. As VM instances are created on compute nodes, the Ethernet trunk port between the ToR and compute node is automatically configured to allow the required VLANs. The Arista ML2 mechanism driver provisions VLANs in parallel with the virtual switch driver, such as OVS, that configures the VLANs on the virtual switch on the hypervisor host, and provides tight integration between network and compute provisioning.

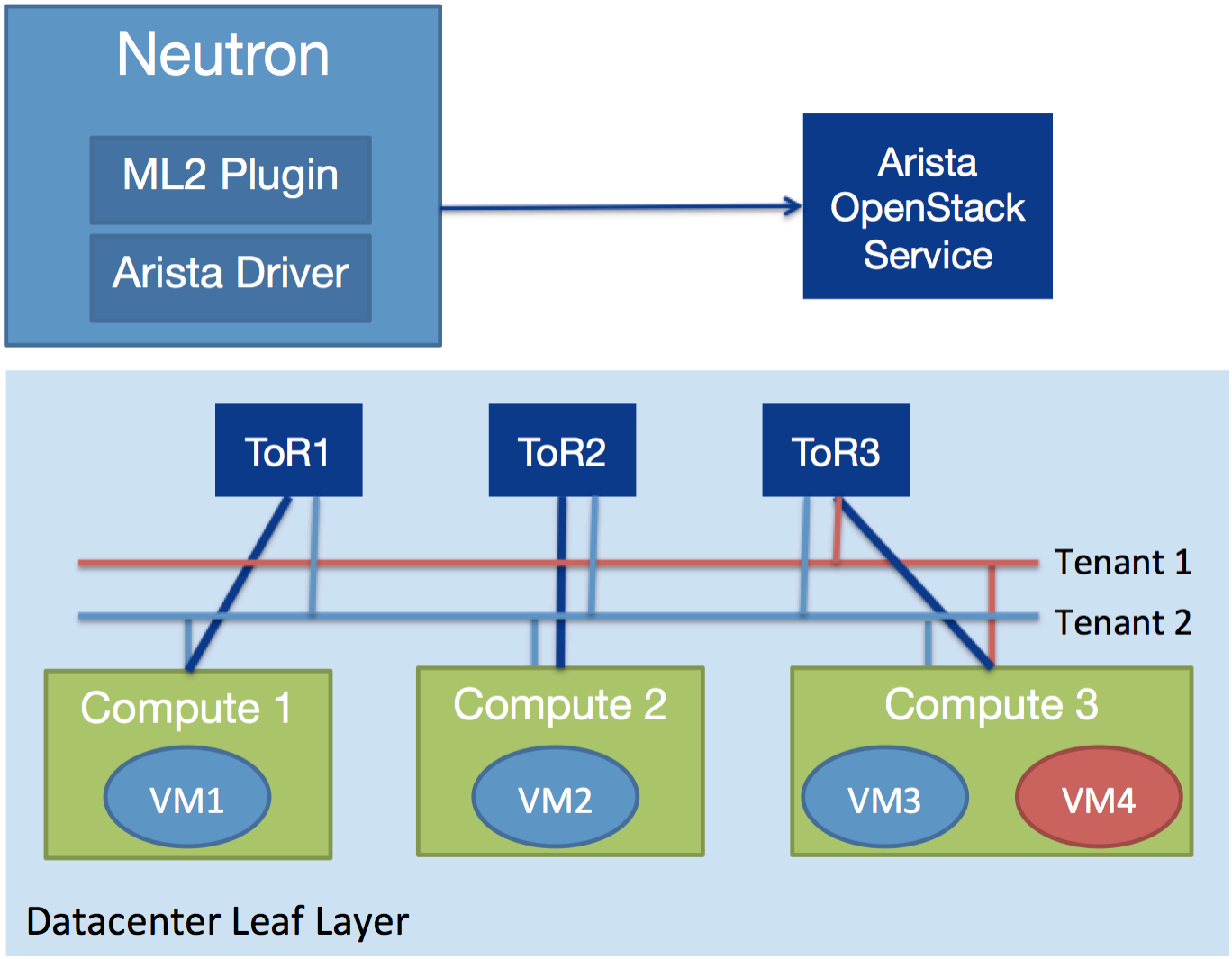


**Figure 1:** Arista ML2 Mechanism Driver

In addition to provisioning the Neutron networks on switches, Arista EOS has commands that can be run on the switches themselves to help with troubleshooting and monitoring Neutron network configurations. For instance, a network administrator can determine what VLAN ID corresponds to a Neutron network quickly using the switch command line interface (CLI).

## EXAMPLES

### show openstack networks

This command shows the networks configured in Neutron, VLAN IDs associated with them, and a VNI mapping if it exists (described in detail later in this document).

```
leaf1.16:47:03#show openstack networks

Region: RegionOne


  Tenant Name: admin
```

```
Tenant Id: 733e7ea13321445ba9d234249798460d

 Network Name Network Id                            Seg Type Seg Id  Maps to VNI
 ------------ ------------------------------------ -------- ------- -----------
 public       2db1eeb6-18f7-4747-bee0-bd2cdd56d8e0 vlan     1101    11101



Tenant Name: demo
Tenant Id: 911682ad3f364c2791aaad31d3c920b7

 Network Name  Network Id                             Seg Type Seg Id Maps to VNI
 ------------- ------------------------------------ -------- ------ -----------
 test network  1c74cb1f-ee00-46d2-9550-9ca0bf04326f vlan     1102   11102
 private       29d42c5d-93f8-4f76-b57c-474afba9f5c9 vlan     1100   11100
```

**show openstack vms**

This command shows the virtual machines currently active in Nova, the host, and the Neutron network that is assigned to the VM.

```
leaf1.16:47:07#show openstack vms
Region: RegionOne
  Tenant Name: demo
  Tenant Id: 911682ad3f364c2791aaad31d3c920b7
    VM Name      VM Id                                  Host      Network Name
    ----------- ------------------------------------ ----------- -------------
     Test  VM    936a8f48-b817-4c56-83bf-40fe7ab26bc6     stack1    test  network
```

## ARISTA ML2 DRIVER FOR HARDWARE BASED VXLAN

The Arista ML2 Driver also supports provisioning for VXLAN-based architectures. A VXLAN-base network delivers layer 2 connectivity for VMs with layer 3 scalability at the leaf and spine.. With hardware VTEP support on many Arista switches, a VXLAN network can be predefined between the leaf switches. This allows the standard ML2 VLAN type driver to configure VLANs on the trunks between the switches and compute nodes (as discussed in the ML2 section), but use VXLAN tunnels between racks.

Using this model, VXLAN encapsulation and decapsulation is done at wire speed in hardware on the switch, removing the performance penalty from using a software-based VTEP. The Arista EOS VXLAN implementation supports VXLAN VNI to VLAN mapping to be done centrally rather than on every switch. The central VXLAN VNI mapping also manages the VNIs present on each switch, only adding them as needed to support the VLANs in use on the switch..

Figure 2 shows a VXLAN deployment scenario across a layer 3 leaf-spine network. In this scenario a central EOS instance assigns and maps VXLAN VNIs to the VLANs configured on the ports connected to the host. This creates a virtual layer 2 network overlay across a routed IP infrastructure. When traffic leaves a VM, it is tagged with a particular VLAN. Once the traffic arrives at the switch, the VLAN tag is removed and it is encapsulated on the VXLAN overlay network to be carried across the layer 3 network to the switch connected to the destination VM. When the encapsulated traffic reaches the destination VTEP, the VXLAN header is removed and then sent on the appropriate Ethernet port, tagged with the VLAN. The Arista ML2 driver behaves in the same manner as before. No additional configuration is needed on the OpenStack nodes since the communication between the compute nodes and switch still uses VLAN tagging and EOS handles the VXLAN mapping.

Arista's VXLAN implementation allows for unicast flooding of Broadcast, Unknown, and Multicast (BUM) traffic through head-end replication (HER), and automatic distribution of VTEP flood lists. This removes the need for multicast or static VTEP configuration on the switches.

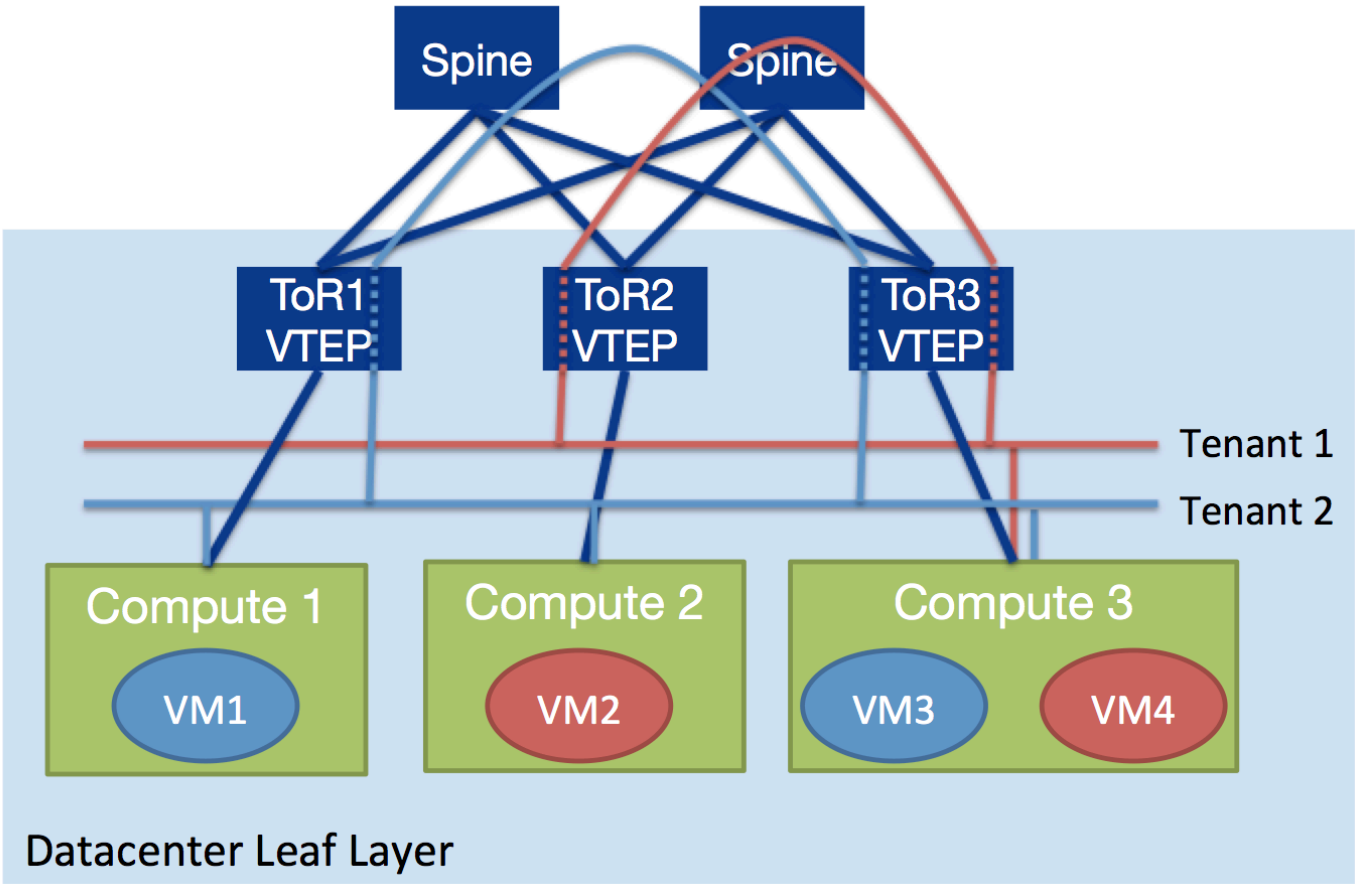


**Figure 2:** Arista Hardware VXLAN provisioning

**EXAMPLES**

### show openstack networks

As described above, this command not only shows the Neutron networks, but the VLAN to VNI for each network.

```
leaf1.16:47:03#show openstack networks

Region: RegionOne


  Tenant Name: admin

  Tenant Id: 733e7ea13321445ba9d234249798460d


   Network Name Network Id                            Seg Type Seg Id  Maps to VNI

   ------------ ------------------------------------ -------- ------- -----------

   public       2db1eeb6-18f7-4747-bee0-bd2cdd56d8e0 vlan     1101    11101




  Tenant Name: demo

  Tenant Id: 911682ad3f364c2791aaad31d3c920b7


   Network Name  Network Id                            Seg Type Seg Id Maps to VNI

   ------------- ------------------------------------ -------- ------ -----------

   test network  1c74cb1f-ee00-46d2-9550-9ca0bf04326f vlan     1102   11102

   private       29d42c5d-93f8-4f76-b57c-474afba9f5c9 vlan     1100   11100
```

### show openstack config networks vni mapping

This command shows the VLAN to VNI mappings that are provisioned across the network. They are only configured in one location and propagated to the VTEPs.

```
leaf1.16:49:10#show openstack config networks vni mapping

Region: RegionOne


   VLAN range    VNI range

---------------- -----------

   1100-1199     11100-11199
```

**show interfaces vxlan 1**

In this command you may notice that the mappings are learned dynamically, as opposed to static configuration on the VTEP. We are using head-end replication on this VTEP to forward BUM traffic to the other VTEPs that are members of a VNI.

```
leaf1.16:37:13#show interfaces vxlan 1

Vxlan1 is up, line protocol is up (connected)

  Hardware is Vxlan

  Source interface is Loopback0 and is active with 3.3.3.3

  Replication/Flood Mode is headend with Flood List Source: VCS

  Remote MAC learning via VCS

  Static vlan to vni mapping is

  Dynamic vlan to vni mapping for 'vcs' is

    [1100, 11100]     [1102, 11102]

  Headend replication flood vtep list is:

   100 3.3.3.3         4.4.4.4

  1100 3.3.3.3

  1102 3.3.3.3
```

## ARISTA LAYER 3 SERVICE PLUGIN

In the OpenStack Juno release, Arista has added an Arista layer 3 service plugin. This plugin replaces the existing Neutron layer 3 service plugin. It creates switched virtual interfaces (SVIs) on TOR switches when a virtual router is created in Neutron. Once configured, the hardware switch becomes the default gateway for the VMs, and all routing can be done in hardware on the switch, instead of in software at the Neutron network node. In a multi-link aggregation (MLAG) environment the switches can be configured to use virtual IP addresses to leverage Virtual ARP (VARP) for first hop redundancy. Pushing the routing functionality into a hardware switch can provide increased performance and scalability versus the default software based router.

In Figure 3, when a virtual router is created in Neutron, the Arista layer 3 service plugin configures both ToR switches with IP addresses in the subnet assigned to the Neutron network. For example, if the subnet is 1.1.1.0/24 for VLAN 100, then ToR1 would have a VLAN 100 SVI created with IP address of 1.1.1.253 and a virtual IP of 1.1.1.1. ToR2 would also have an SVI created, but with IP addres of 1.1.1.254 and virtual IP of 1.1.1.1. When the VMs are instantiated, they would use 1.1.1.1 as their default gateway, and use the hardware ToR switch for routing.

When a VXLAN overlay is used between the ToR switches, the VXLAN routing functionality of Arista EOS allows the layer 3 functionality to work across VXLAN VNIs as well.

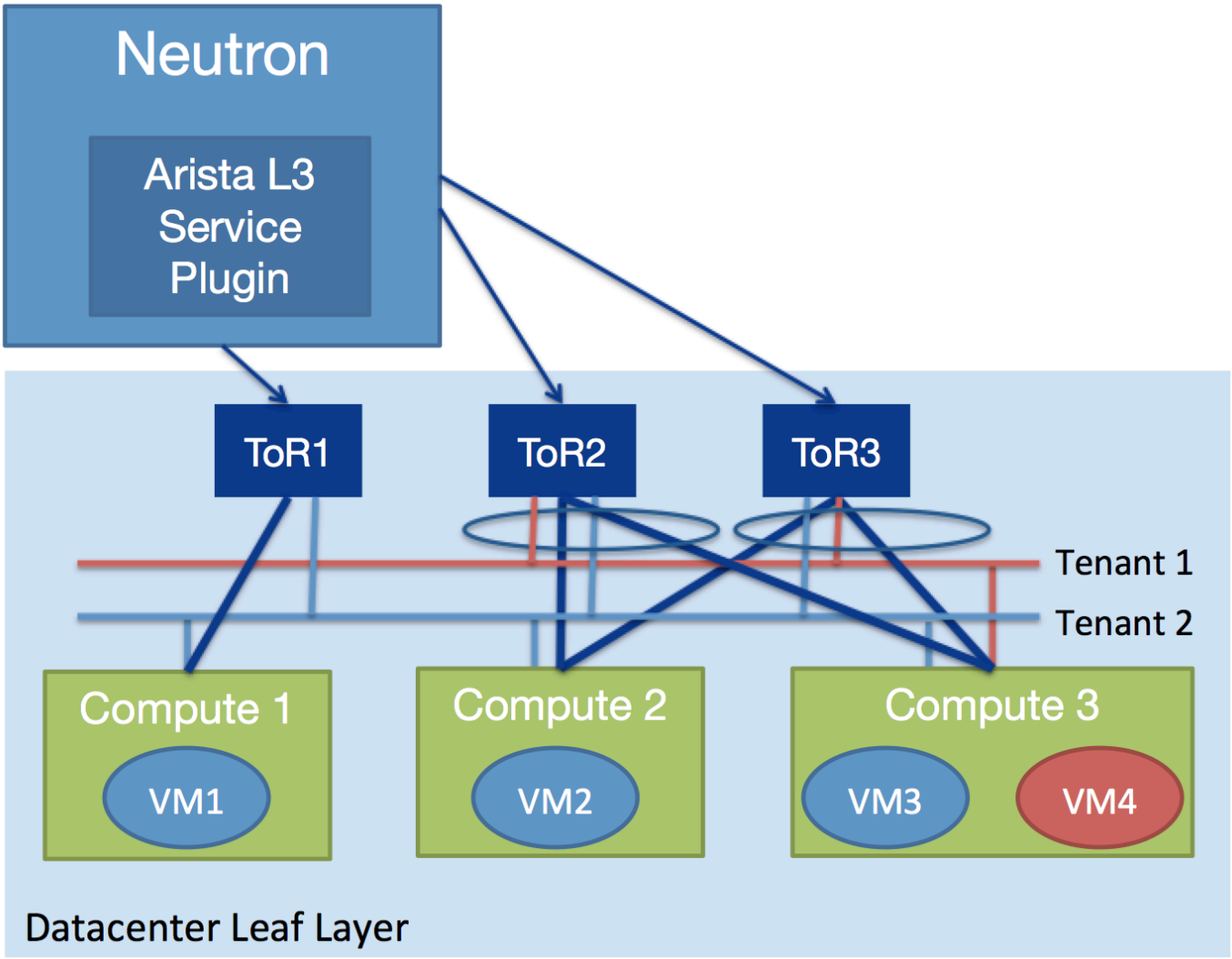


**Figure 3:** Arista Layer 3 Service Plugin

## CONCLUSION

The Arista OpenStack solution provides a number of ways for administrators to orchestrate their Arista switches. The ML2 plugin automates the provisioning of VLANs on Arista switches and can optionally be combined with an Arista VXLAN overlay to provide the functionality across a Layer 3 core. With the Arista layer 3 service plugin, a hardware switch can serve as the routing gateway, even in a VXLAN environment where VXLAN routing is required.

The ability to orchestrate the physical network devices provisioning within the OpenStack solution is a significant achievement for the market. Arista has consistently led the way in providing new and open functionality to the OpenStack community and has an extensive set of features designed to address the needs of scaling an OpenStack cloud.

**Table 1:** Minimum supported software versions

| OpenStack Integration Feature | Minimum Supported OpenStack Version | Minimum Supported EOS Version |
|---|---|---|
| Arista ML2 Mechanism Driver for VLAN | Havana | 4.12.1 |
| Arista Layer 3 Service Plugin | Juno | 4.14.5F |
| Arista ML2 Driver for Hardware based VXLAN | Juno | 4.14.5F |


**Santa Clara—Corporate Headquarters**
5453 Great America Parkway
Santa Clara, CA 95054
Tel: 408-547-5500
www.arista.com

**Ireland**—International Headquarters
4130 Atlantic Avenue
Westpark Business Campus
Shannon
Co. Clare, Ireland

**Singapore**—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989