

Arista - Physical Network Infrastructure for OpenStack and Container

patarakorn@arista.com

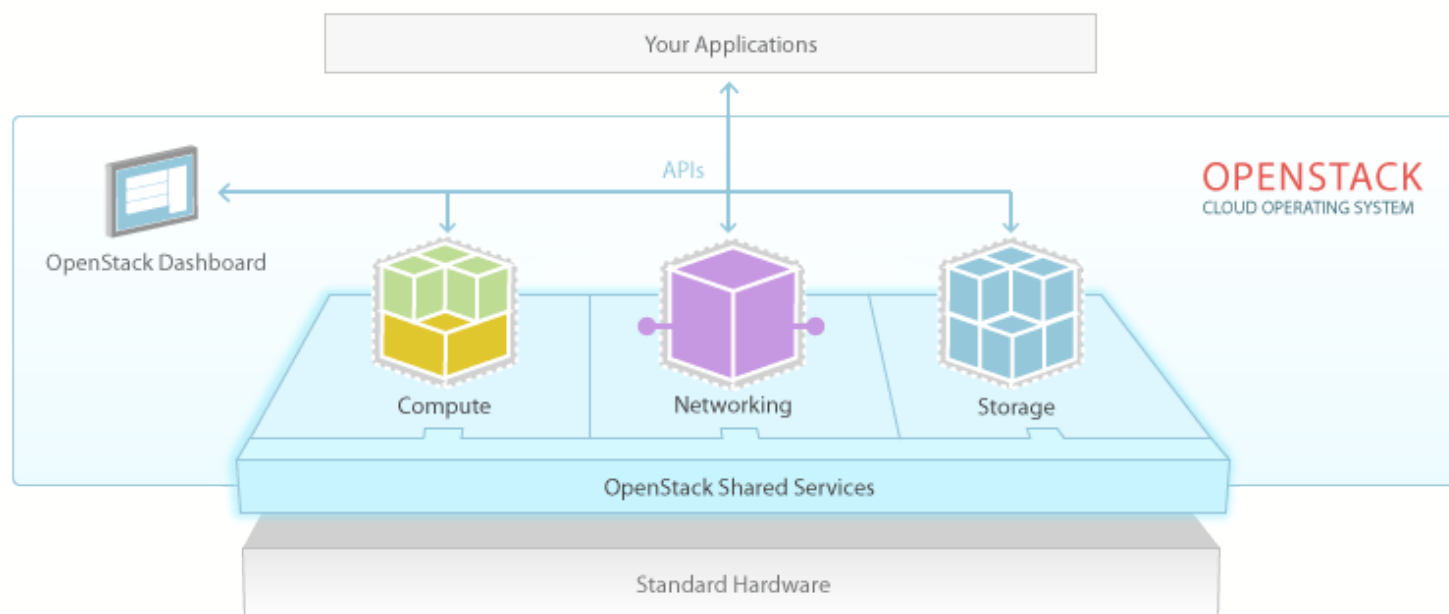
Patarakorn Vaeteewootacharn

System Engineer, Thailand, **Arista** Networks



OpenStack Overview

OpenStack Overview

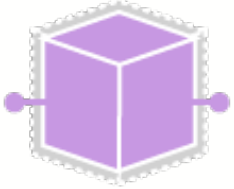


OpenStack is a **cloud operating system** that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.

OpenStack Core Services



- Compute ("Nova") provides virtual servers upon demand.



- Network ("Neutron") is a pluggable, scalable and API-driven system for managing networks and IP addresses.

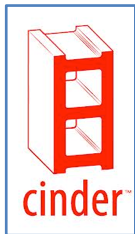


- Block Storage ("Cinder") provides persistent block storage to guest VMs

OpenStack Core Services



- Dashboard ("Horizon") provides a modular web-based user interface for all the OpenStack services



- Object Store ("Swift") provides object storage. It allows you to store or retrieve files
- Image ("Glance") provides a catalog and repository for virtual disk images



- Identity ("Keystone") provides authentication and authorization for all the OpenStack services

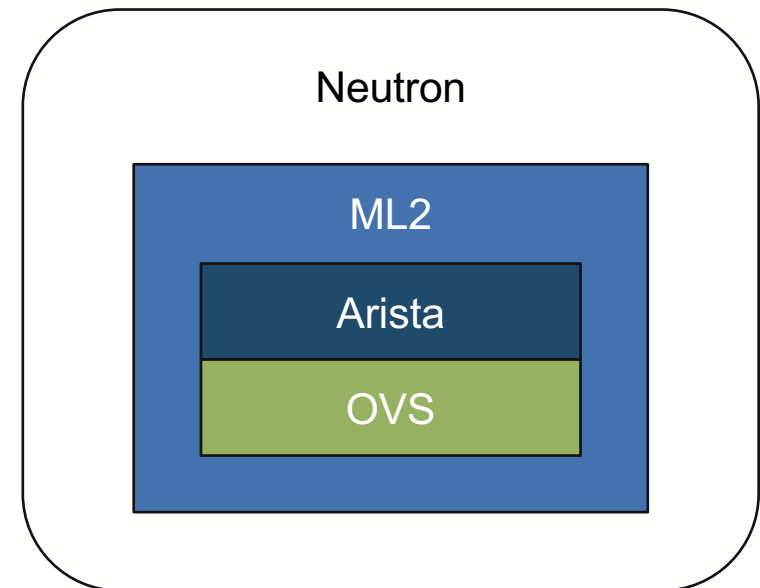
Neutron Overview

- OpenStack Networking Service
 - Introduced in the Folsom release
- Manages
 - Tenant Networks
 - Logical Routers
 - L4-7 Services – NAT, LBAAS, FWAAS, ...



Neutron - Modular Layer 2

- Plugin/API architecture for networking
- ML2 Plugin
 - Allows multiple layer 2 implementations
 - Type Driver (vlan, vxlan, etc..)
 - Mechanism Driver (ovs, arista, etc..)
- Receives calls from other services such as Horizon and Nova



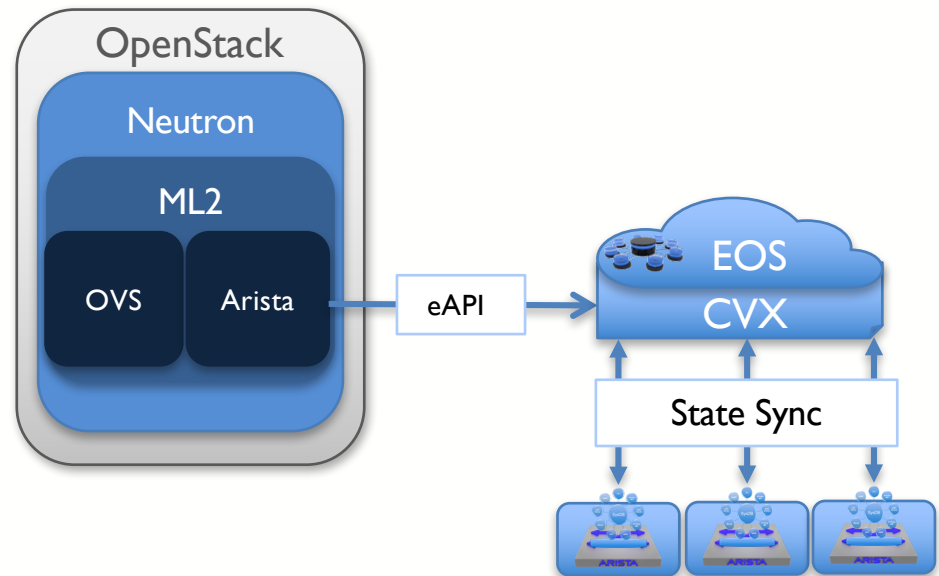
Arista CVX Integration with OpenStack

Arista OpenStack Integration

- Arista Focus - Provide end-to-end tenant network visibility and provisioning across virtual and physical infrastructure for the highest performance VLAN and VXLAN fabrics
- Key Solution Features
 - Provide full visibility to the network team of the OpenStack state
 - Automatically provision the physical network in response to tenant configuration within OpenStack

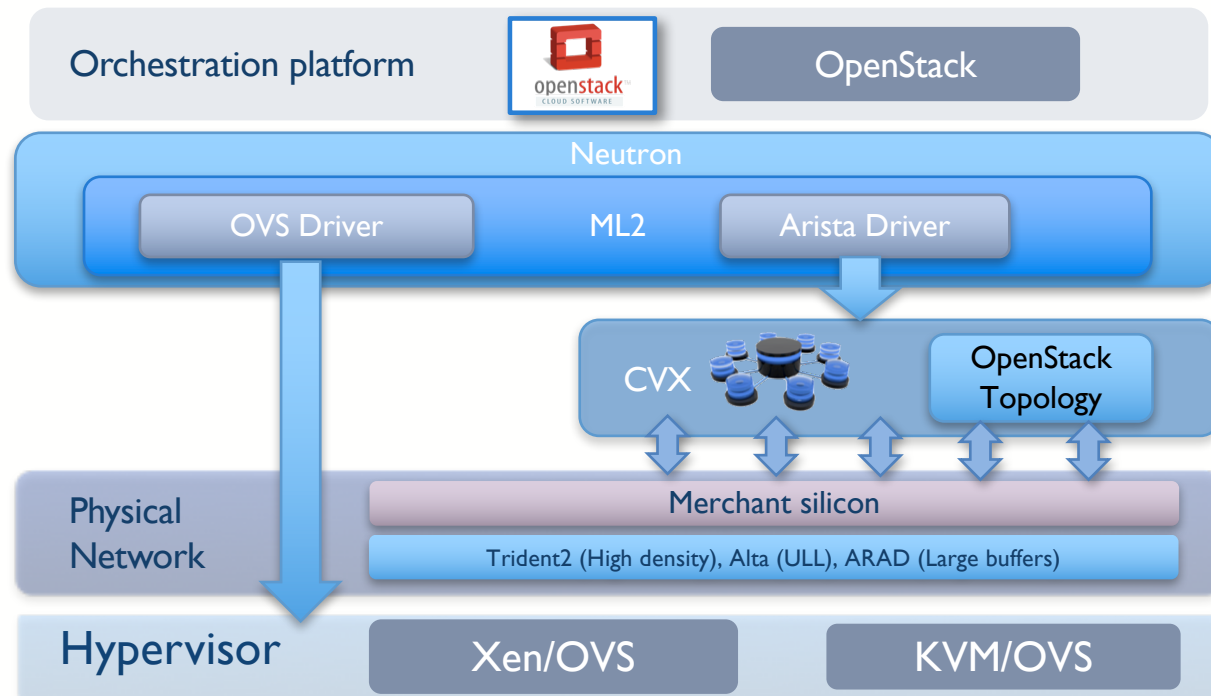
Arista OpenStack Integration

- Arista CVX
 - Collection of global network topology from switches
 - Provisioning of the switches via eAPI
- Arista ML2 mechanism driver
 - Notify created OpenStack state to the CVX
 - Communication via the Arista eAPI



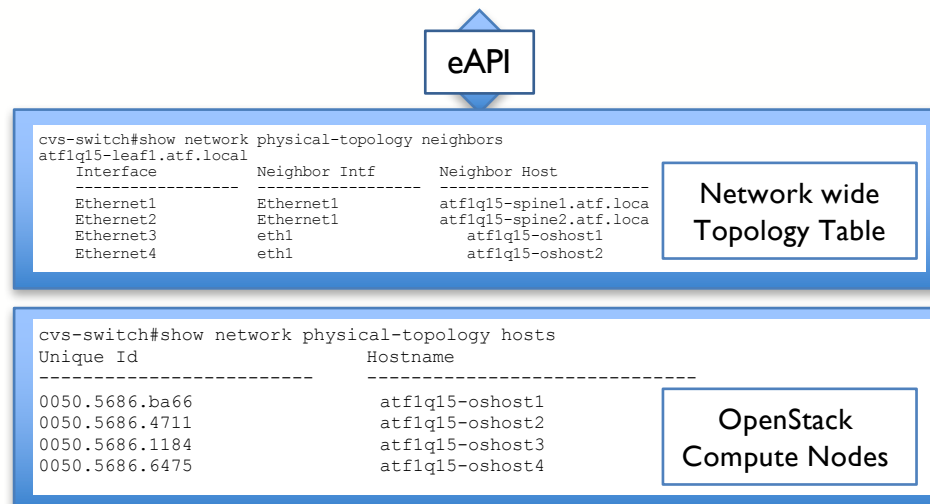
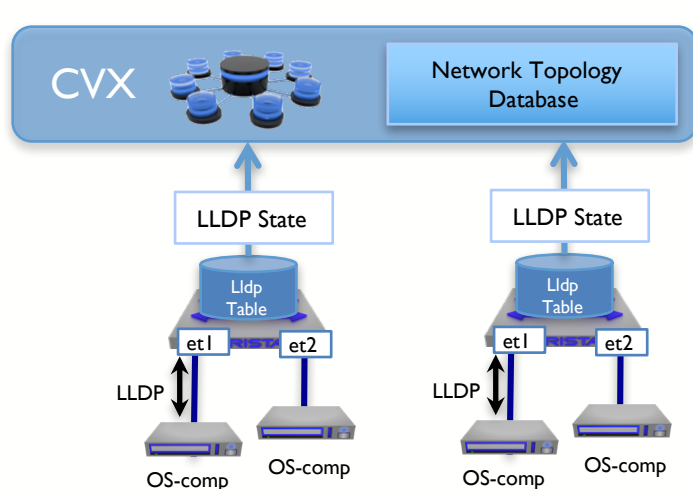
Arista OpenStack Integration

- Network orchestration with OpenStack Neutron ML2 Plugin



Arista OpenStack Integration

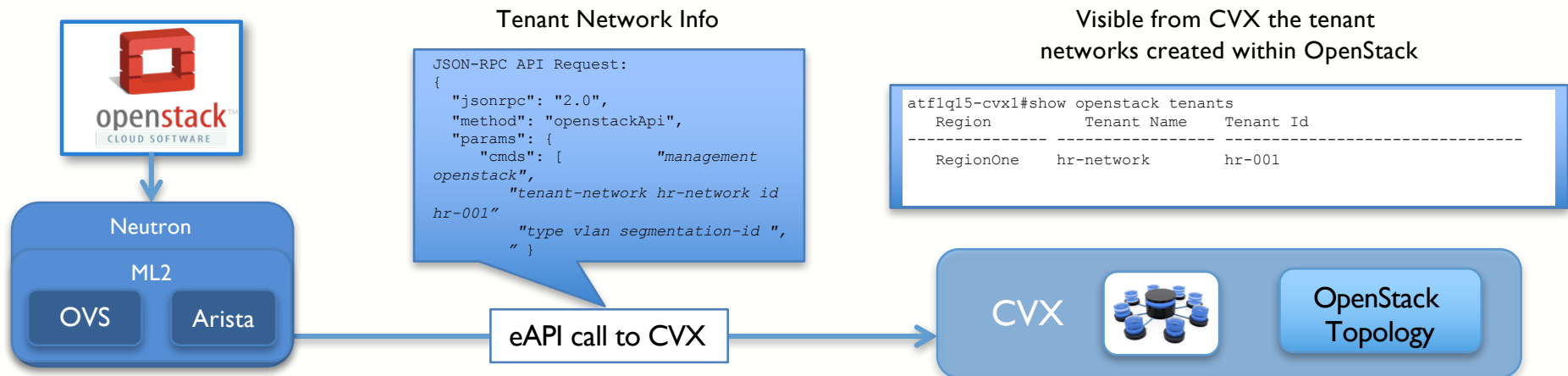
- CVX builds a full topology of the physical infrastructure
 - Leaf switch builds their local topology table using standard LLDP
 - Contains directly attached OpenStack compute nodes, which will host the virtual machines
 - CVX mounts the local LLDP tables, providing a network wide view
 - CVX knows the physical location (switch and interface) each OS compute node is attached



Network wide topology visible from CVX eAPI to consume the info northbound

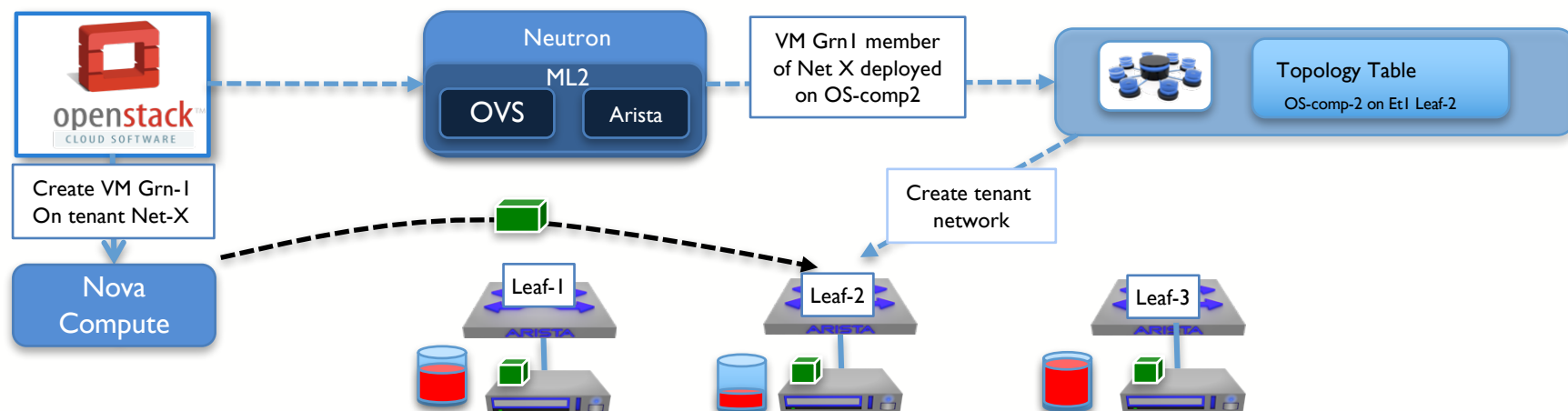
Arista OpenStack Integration

- Step1: OpenStack, Horizon portal create a tenant network
 - ML2 plugin with the Arista driver installed
 - Plugin converts the network create call to an eAPI to CVX
 - CVX is now aware of all networks created with OpenStack
 - Network only deployed on the physical infrastructure if a VM is attached to the network



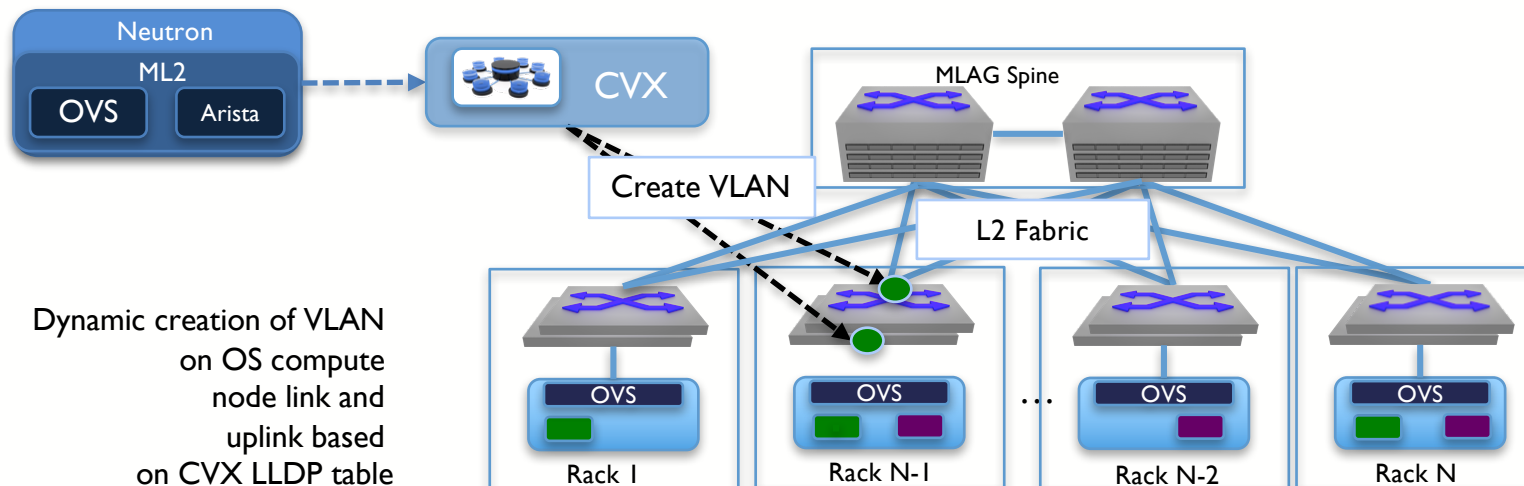
Arista OpenStack Integration

- Step 2: OpenStack, Horizon portal create VM and apply to network
 - Virtual Machine created within Horizon and attached to the tenant's network
 - Nova-scheduler deploys the VM on the compute node based on available resources
 - CVX has visibility of the compute node the VM is deployed on via the plug
 - From the global LLDP table maps the Compute node to a switch port, and dynamically deploys the tenant network on the switch



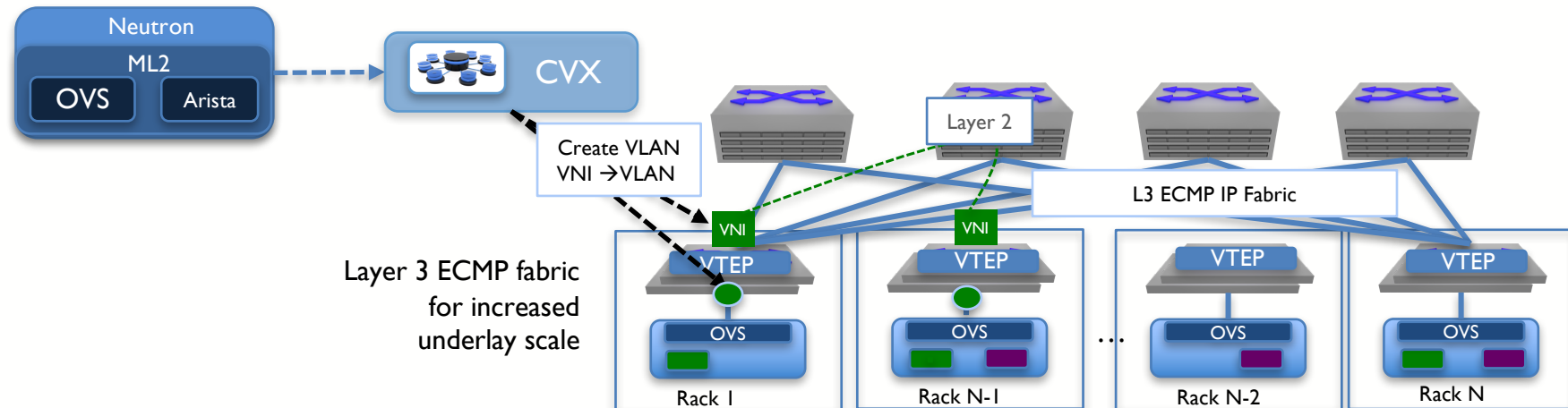
Arista OpenStack Integration – VLAN Based

- Network Type created by the CVX can be a simple Layer 2 VLAN
 - Tenants VMs deployed within dedicated tenant VLANs
 - CVX dynamically deploys the tenant VLAN on the appropriate Leaf edge port and Uplink
 - Deployed on a standard MLAG (Layer 2) topology



Arista OpenStack Integration – VXLAN Based

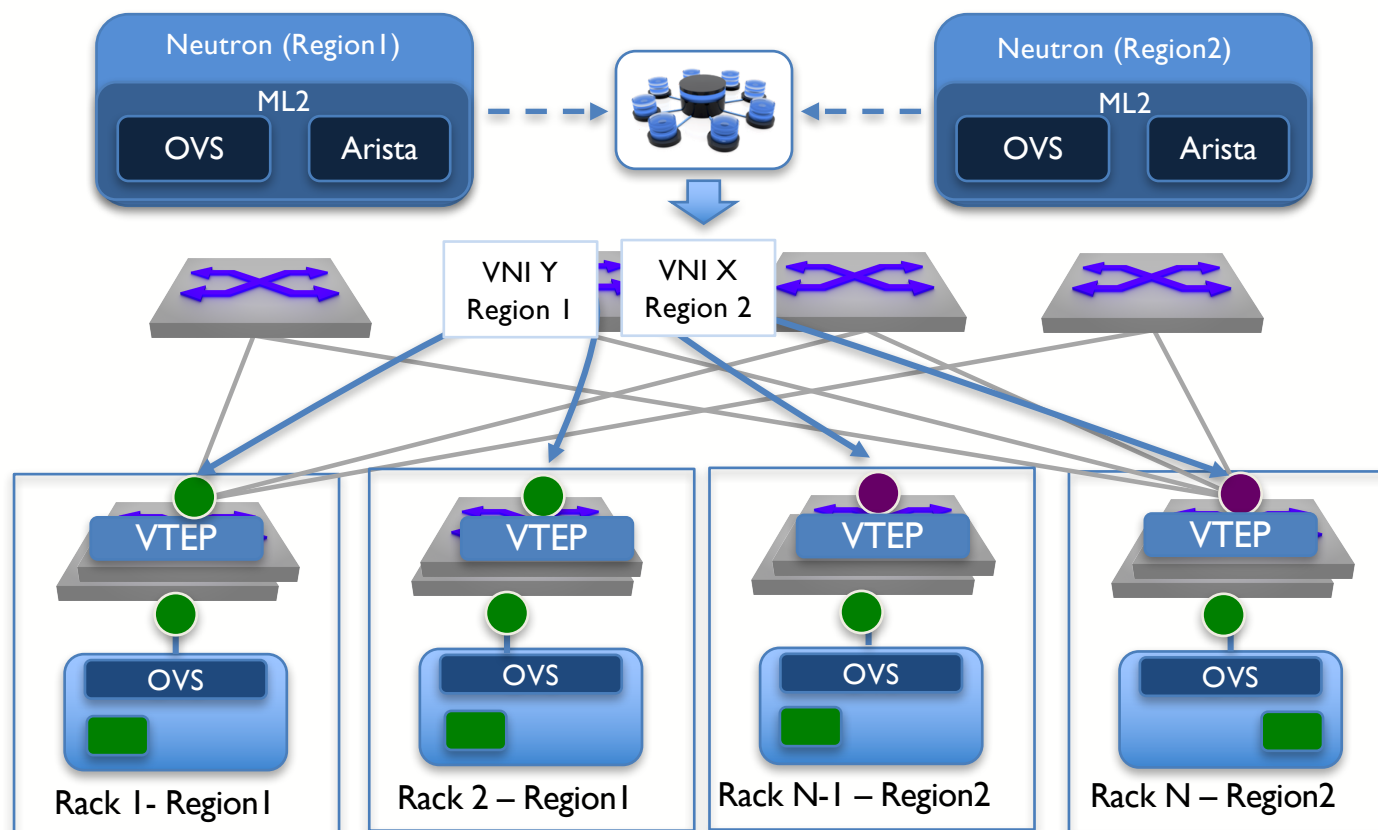
- OpenStack integration with VXLAN
 - Scalable IP fabric with a Layer 3 ECMP design
 - Hardware VXLAN VTEP configured on every leaf switch
 - Layer 2 connectivity between rack via VXLAN across the L3 fabric



Scaling OpenStack

- Multiple OpenStack clusters supported per CVX instance
- Can be combined with other network virtualization
 - NSX
 - Etc
- VXLAN breaks out of the 4K VLAN limit
 - 16M VNIs mapped to locally significant VLANs

Multi-Tenant OpenStack Deployment

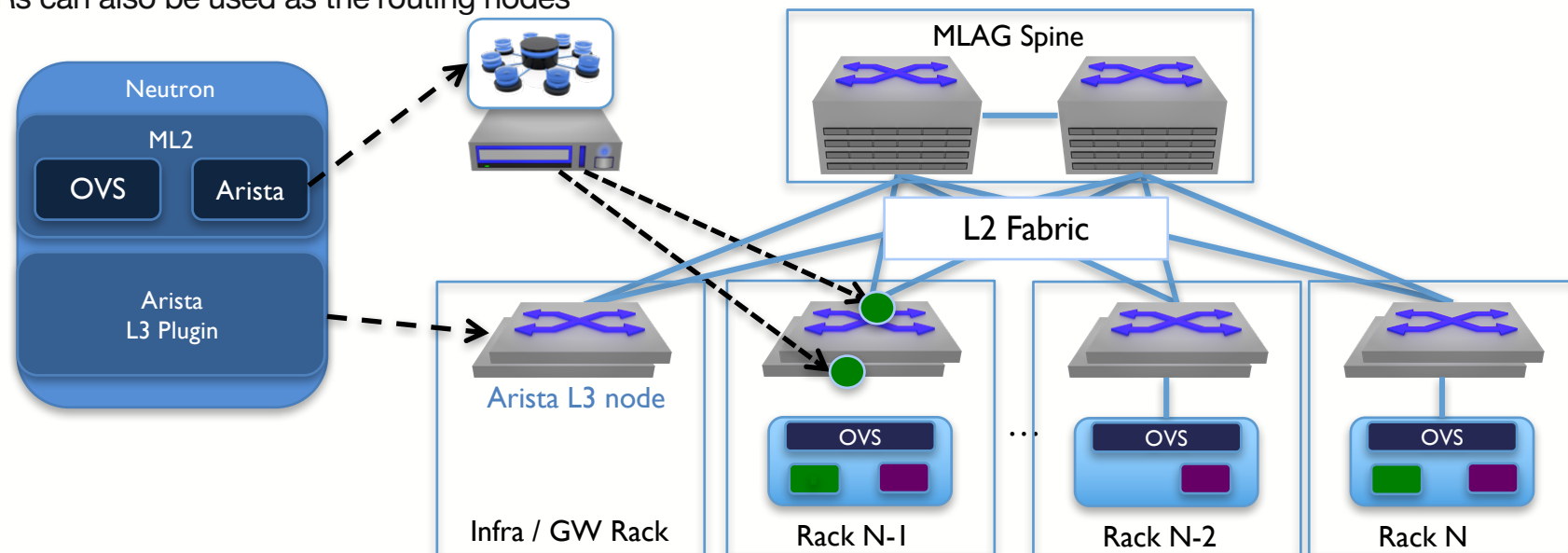


Routing with OpenStack

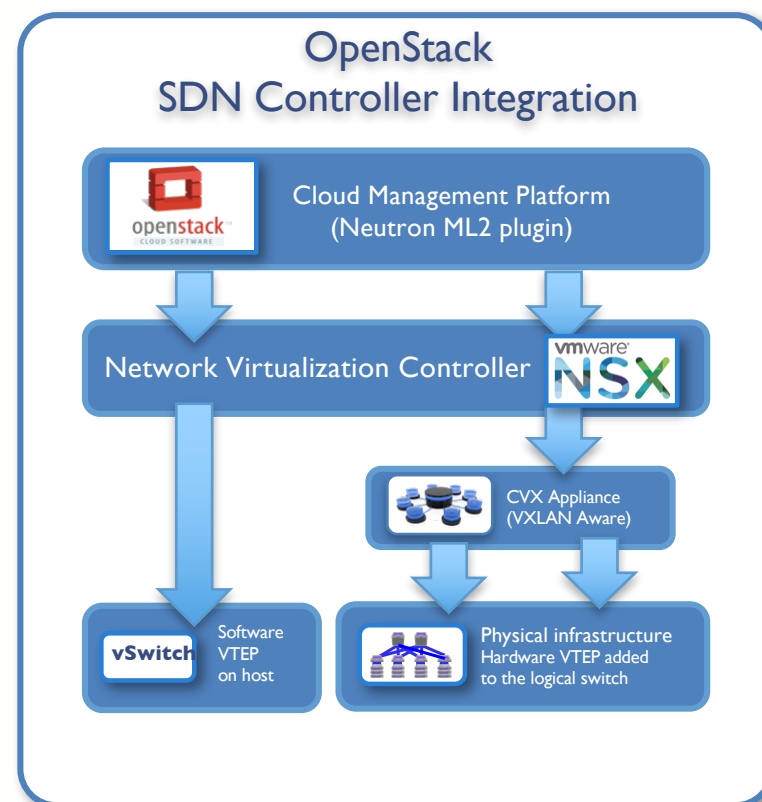
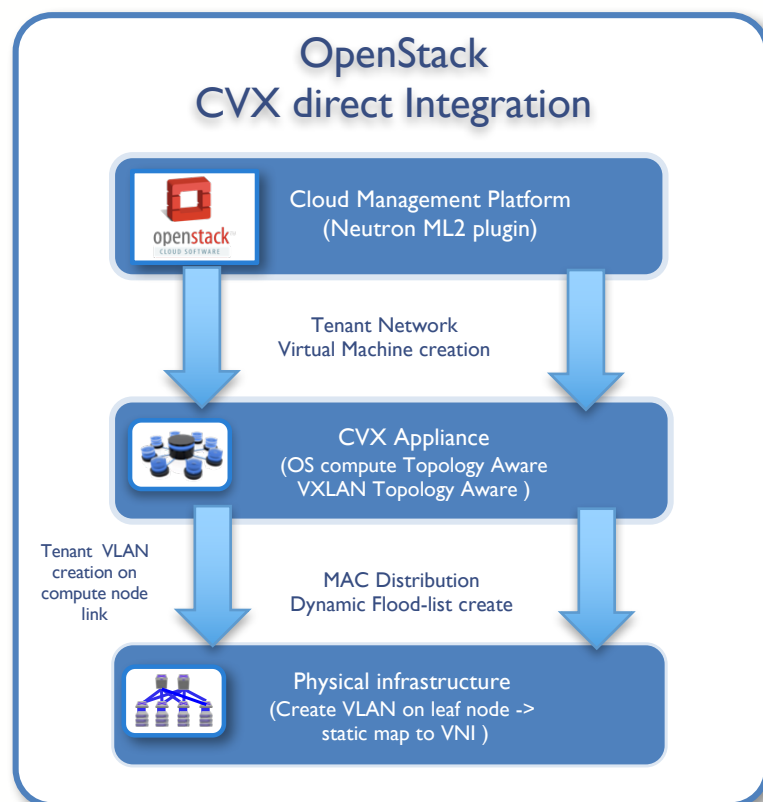
- L2 up until now, how do you route?
- Can be performed by a Network Node
- Allows connectivity between tenants and external networks
 - NAT Support
 - VRF Support
- Limited by software
- Alternative is perform this at the switch...with limitations!

OpenStack Integration – L3 Plugin

- Arista L3 plugin provisions SVIs over eAPI in response to tenant's creating logical routers
- Routing happens at dedicated network nodes
 - Pair of MLAGed physical devices
 - Active-Active HA via MLAG
 - Performs routing for the OpenStack cluster
 - » Can be scaled out horizontally by tenant as needed
- TORs can also be used as the routing nodes

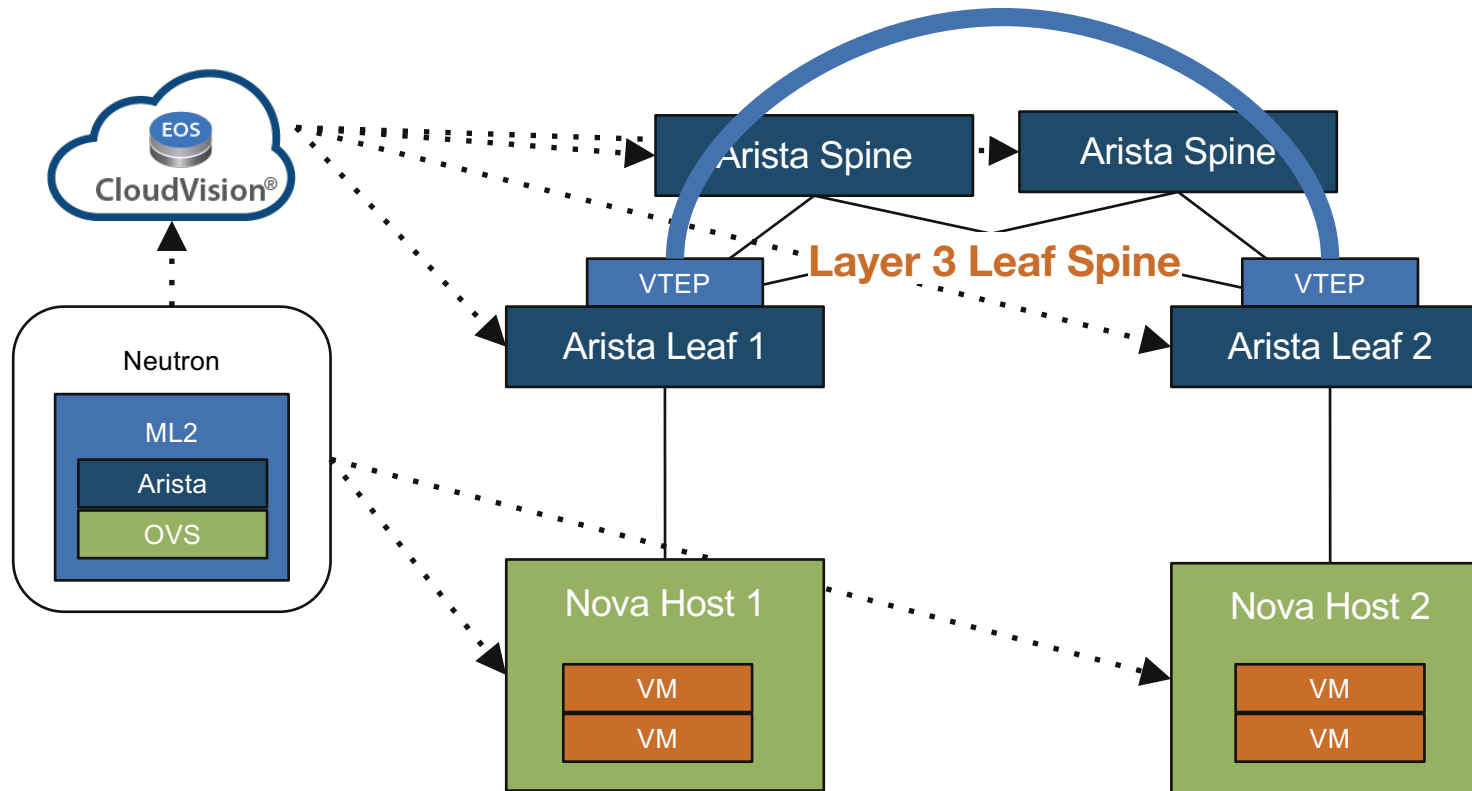


OpenStack Orchestration models



OpenStack Integration - Demo

Demo Topology



Arista - Docker

patarakorn@arista.com

Patarakorn Vaeteewootacharn
System Engineer, Thailand, **Arista** Networks

Introducing Containers

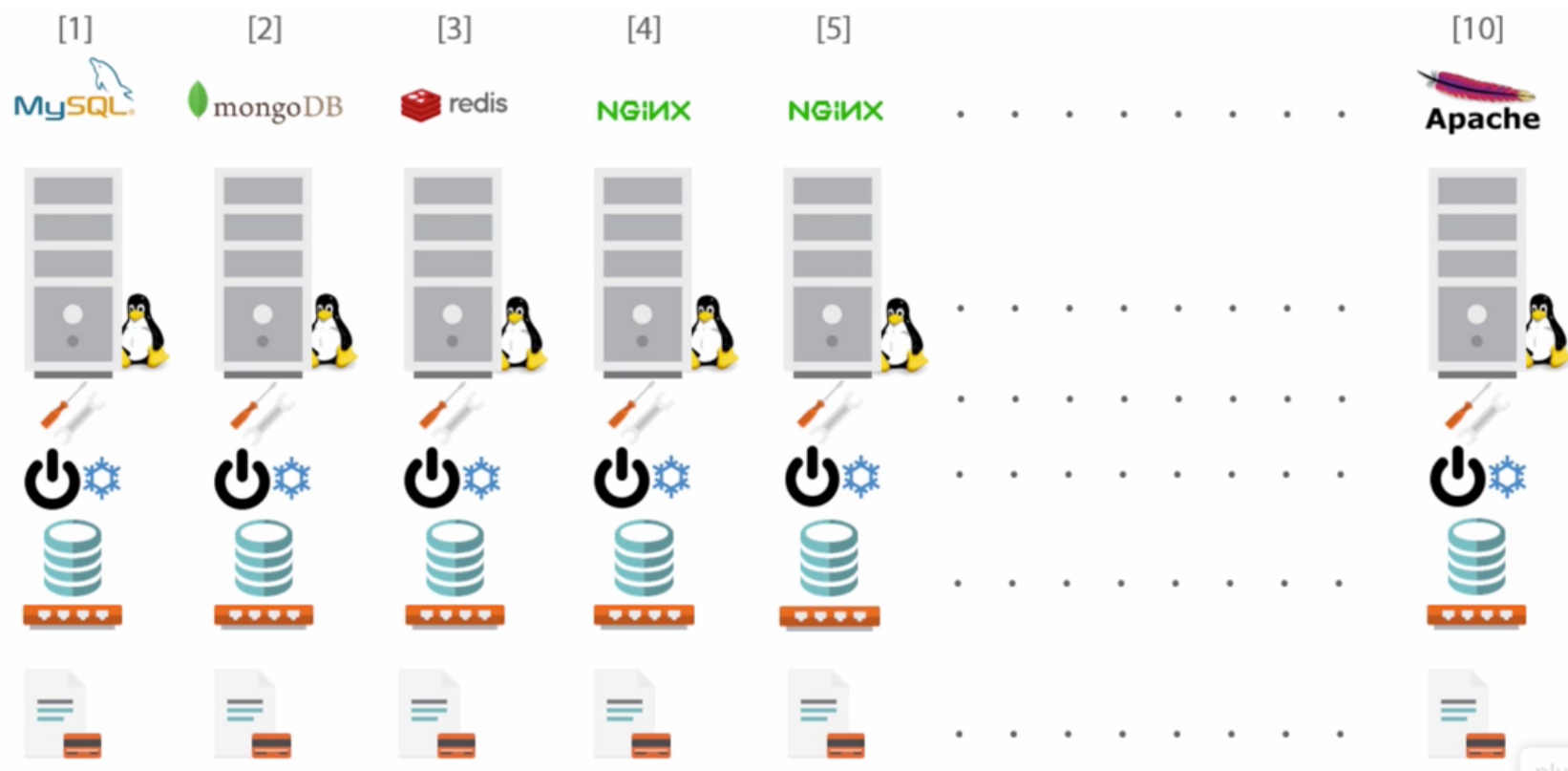
Introducing Containers

Server : Application
1 : 1

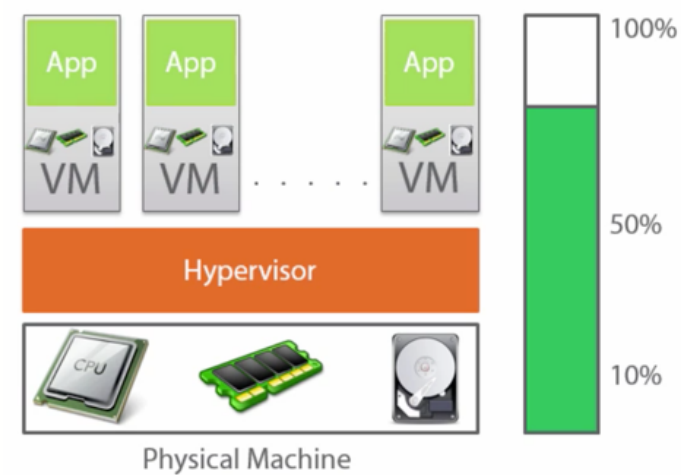
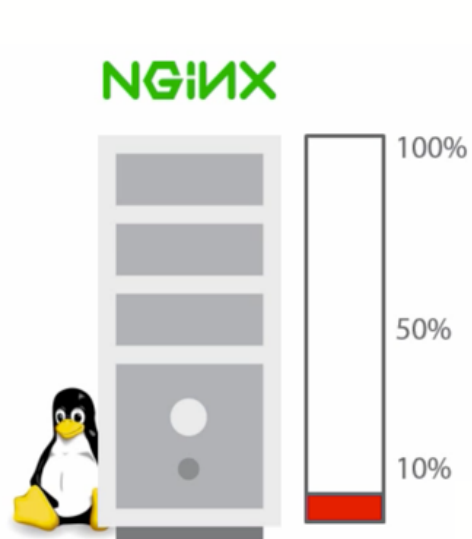
Introducing Containers

Server : Application

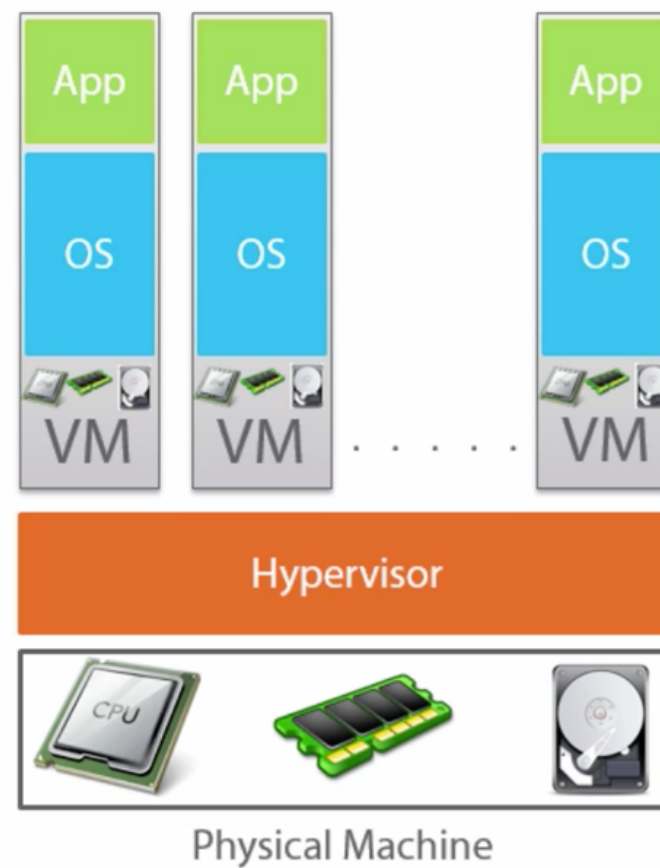
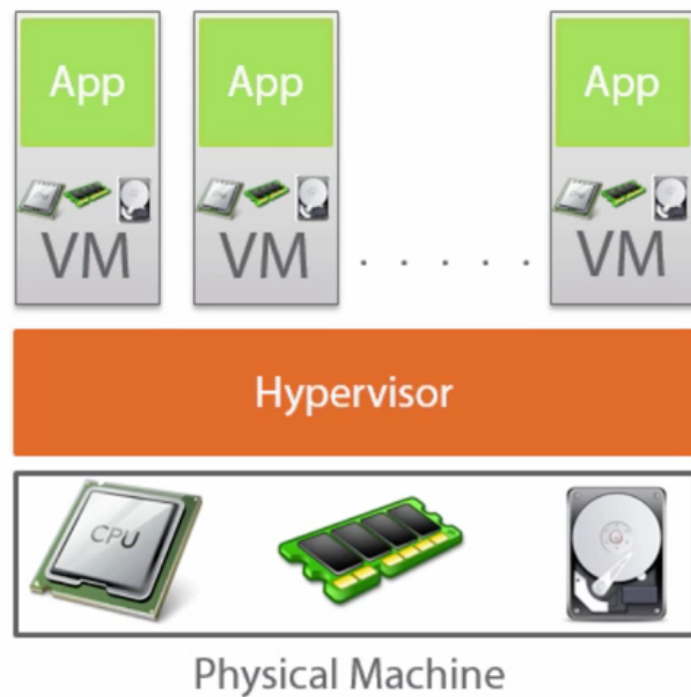
1 : 1



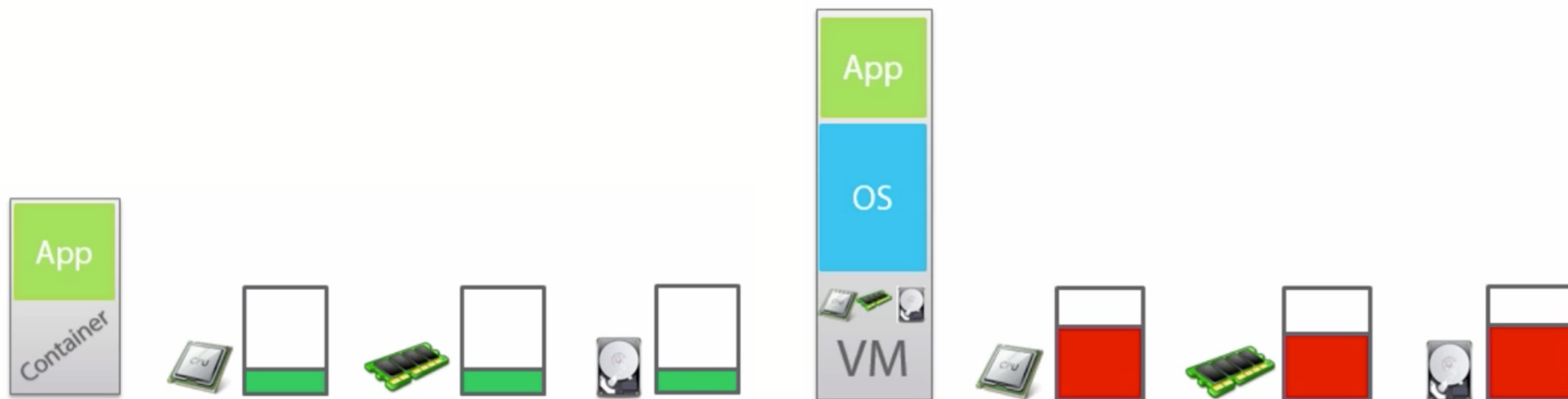
Introducing Containers



Introducing Containers

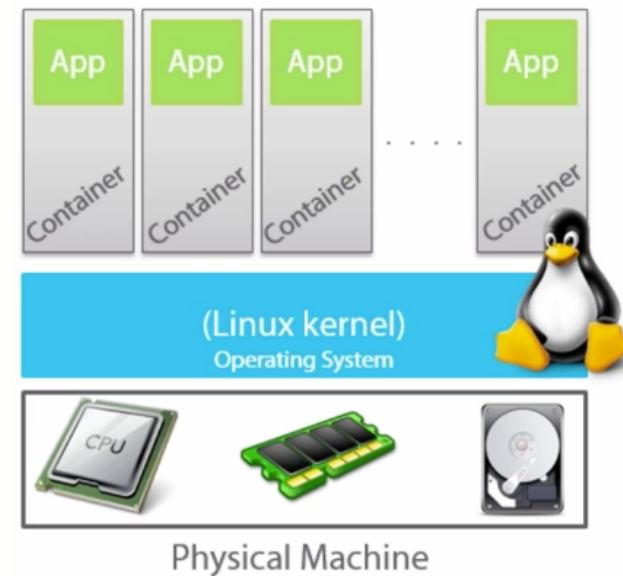
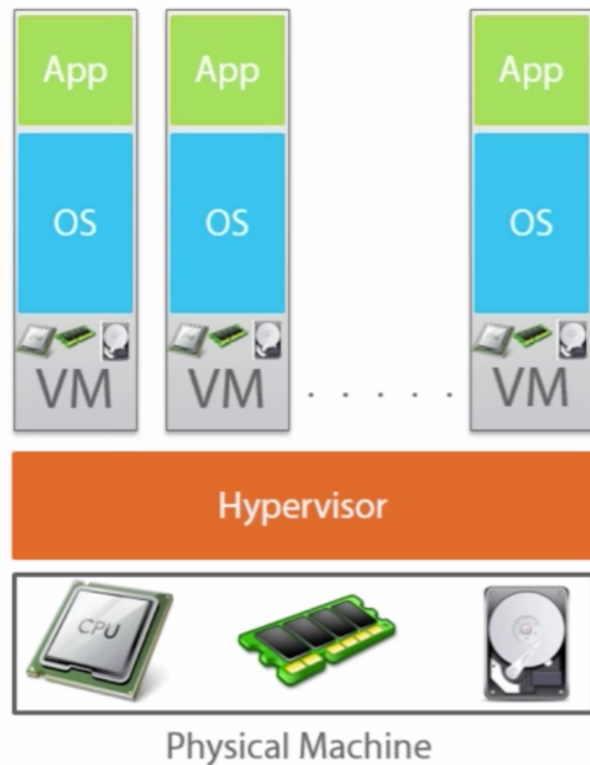


Introducing Containers



- Containers are more lightweight than Virtual Machines

Introducing Containers



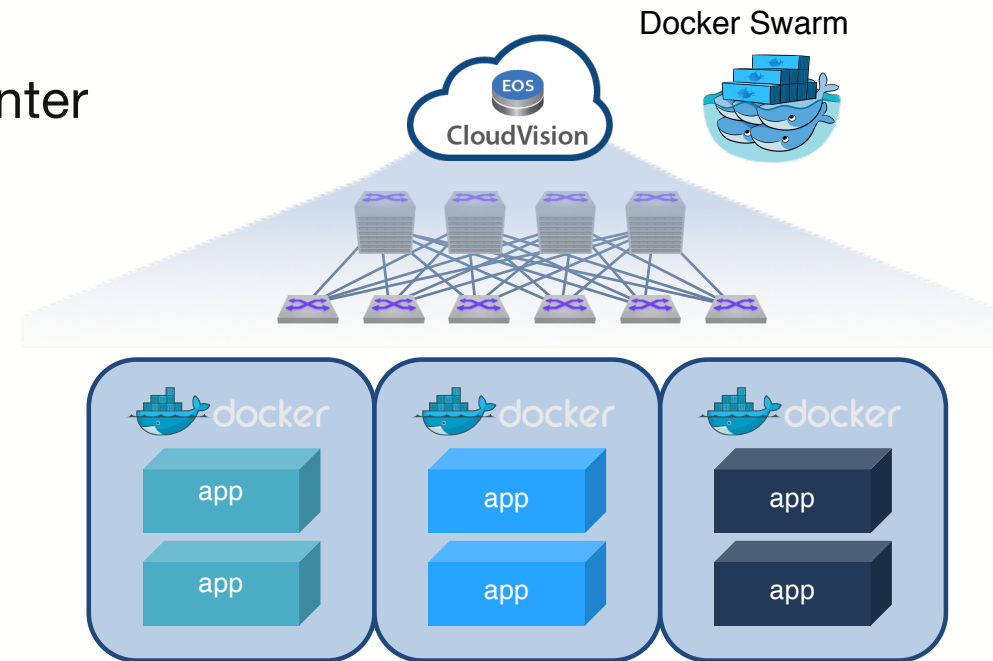
- **Containers consume less CPU, RAM and disk resource than Virtual Machines**

Run Docker on EOS



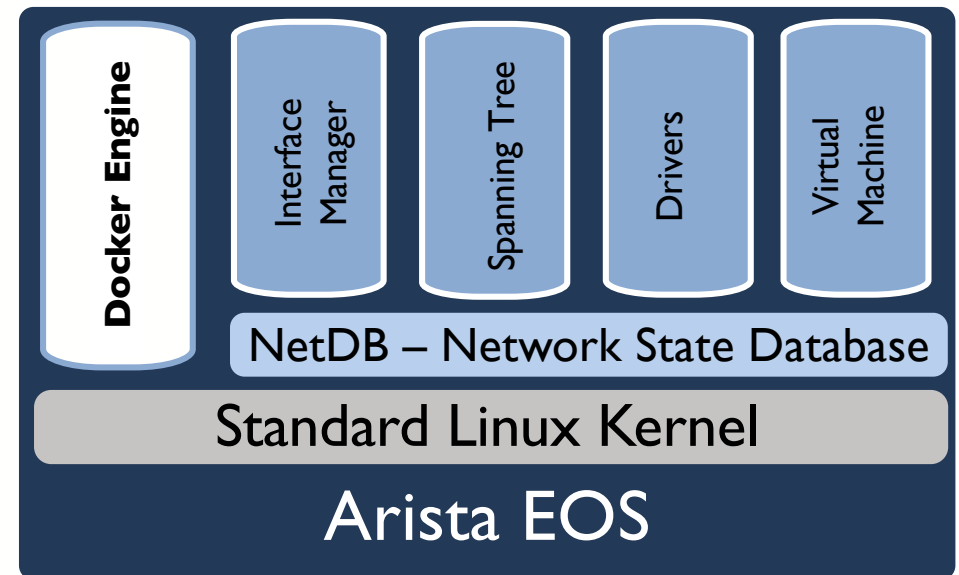
Arista and Docker Solutions

- 1) Deploy Containers on EOS with Docker
- 2) Monitor Containers inside the Data Center



Docker Containers on EOS

- Simplify development and deployment of applications on switches
- Deploy any of the thousands apps from Docker Hub
- Deploy enterprise apps with Docker Trusted Registry
- Powerful abstraction for DevOps tools
- Made possible by running a true, unmodified Linux kernel



Natural extension of Arista's EOS+ Platform

What would you run on a switch?

- sflowtool
- DHCP Server
- iperf
- fping
- etc..

Example: sflowtool

<http://www.inmon.com/technology/sflowTools.php>

“The sFlow toolkit provides command line utilities and scripts for analyzing sFlow data”

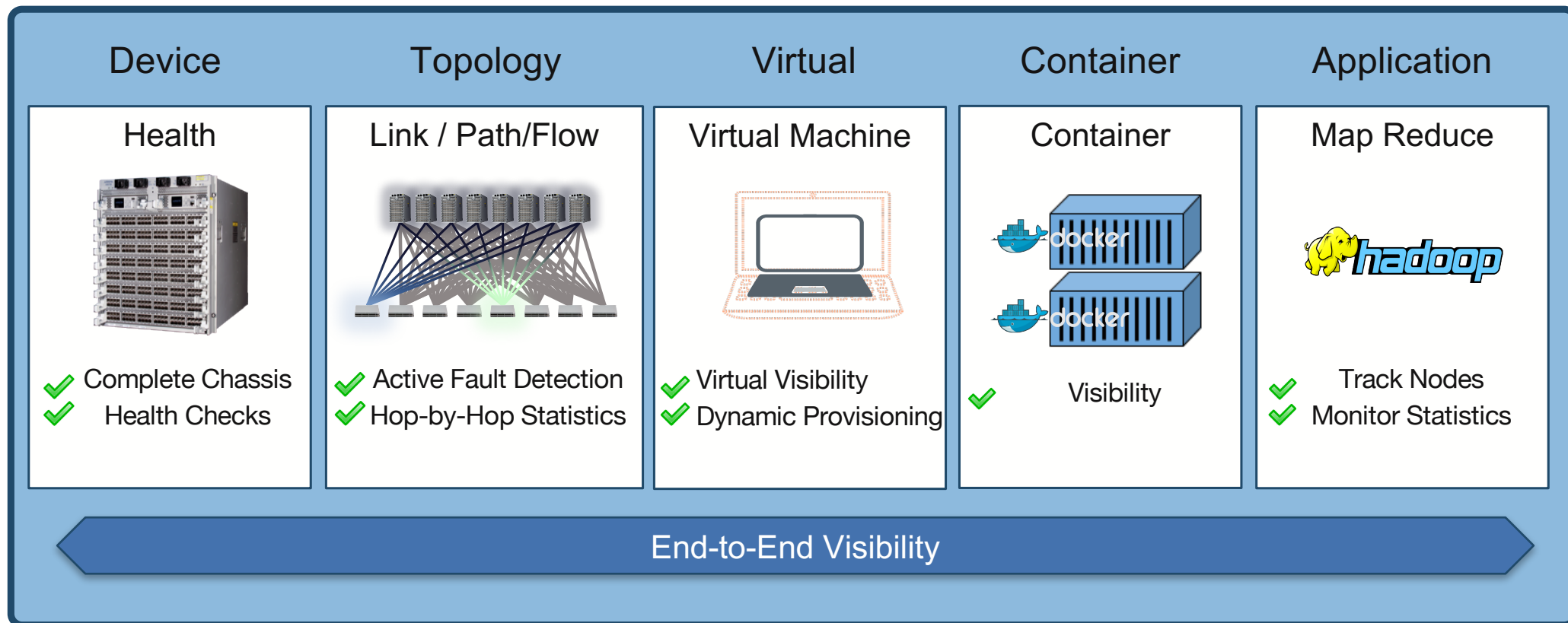
“sflowtool is available as [source code](#). A pre-compiled version of sflowtool is available for [Windows](#).”

This means you would need to compile it yourself.

Alternative: grab a container from Docker Hub

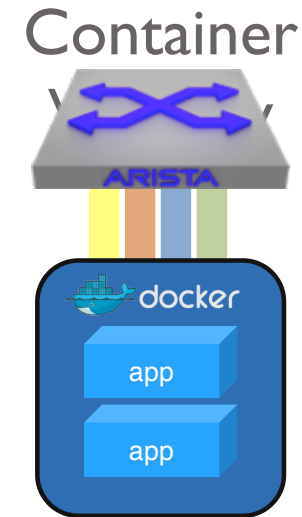
Container Tracer

Visibility via Network Telemetry



Container Tracer

- Displays container workloads
- Provides current Docker node and exposed port information
- Reduces MTTR and improves visibility



```
AristaToR#show containertracer swarm
```

NODE	CONTAINER NAME	PORTS	BRIDGE	PORTS
lnx150	/sad_euclid	80/tcp 443/tcp	docker0	Ethernet13
lnx150	/serene_wright	172.28.168.150:5201->5201/tcp	docker0	Ethernet13
lnx150	/aristademo1	443/tcp 80/tcp	docker0	Ethernet13

```
AristaToR#
```

Visibility for Containers, as VMTracer is for VMs

Container Tracer

```
bleaf5#show dockertracer containers
```

NODE	CONTAINER NAME	PORTS	BRIDGE	PORTS
lnx150	/some-nginx	443/tcp 80/tcp	docker0	Ethernet13
lnx150	/serene_mestorf	172.28.168.150:8082->80/tcp	docker0	Ethernet13
lnx150	/romantic_curie		docker0	Ethernet13
lnx150	/elated_lalande		docker0	Ethernet13
lnx150	/evil_torvalds		docker0	Ethernet13
lnx151	/serene_blackwell	172.28.168.151:8081->80/tcp	docker0	Ethernet15
lnx151	/naughty_shockley		docker0	Ethernet15
lnx151	/prickly_bartik		docker0	Ethernet15
lnx151	/jovial_poincare		docker0	Ethernet15
lnx151	/stupefied_poitras		docker0	Ethernet15
lnx151	/furious_poitras		docker0	Ethernet15
lnx151	/desperate_newton		docker0	Ethernet15

```
bleaf5#
```

Visibility for Containers, as VMTracer is for VMs

Docker - Demo



Thank You

www.arista.com