

NOMS 2014 Tutorial Sessions

TUTORIAL T1

Monday, May 5, 2014

Practical Cloud Deployment and Management with Openstack and Programmable Networking

Masum Z. Hasan, Cisco Systems, U.S.A.
Horst Dümcke, Marlysys SAS, France

NOMS 2014 Tutorial Co-chairs:
Mehmet Ulema, Manhattan College, U.S.A.
Burkhard Stiller, University of Zürich, Switzerland

Practical Cloud Deployment and Management with Openstack and Programmable Networking

Masum Z. Hasan, Horst Dümcke

IEEE/IFIP Network Operations and Management Symposium, NOMS 2014,

5-9 May 2014

Krakow, Poland

About the Speakers

Masum Z. Hasan, PhD
Technology Director at the Cloud CTO office
masum@cisco.com

Horst Dümcke
Founder and President, www.marlysys.com
hdumcke@marlysys.com

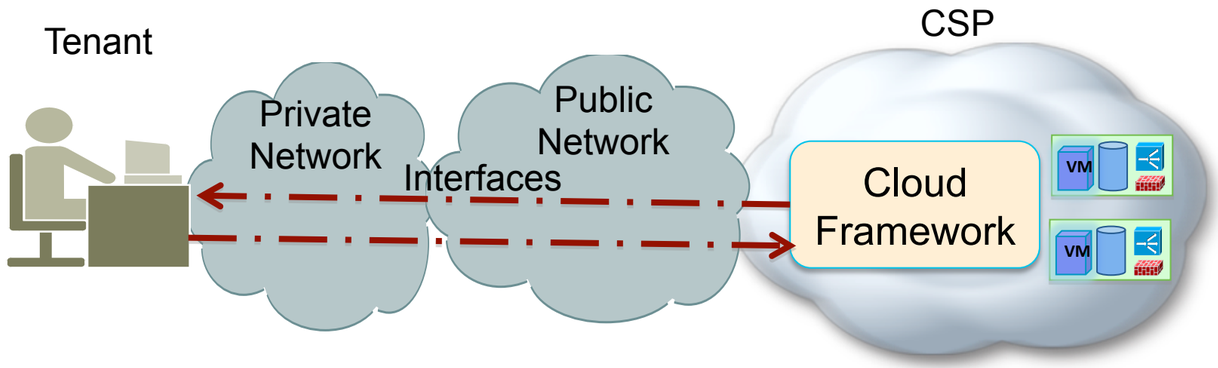
Agenda

- Introduction to Core Cloud Concepts
- Cloud Networking
  - Network Virtualization
  - End-to-end Multitenant Isolation
  - SDN / Programmable Networking
    - OpenFlow
    - Cisco OnePK
    - Open Daylight
- Cloud Framework Openstack
  - Framework introduction
  - Use cases and labs/demo
- Virtual Clouds over Multiple Clouds: Seamless Cloud with Openstack and Programmable Networking

Introduction to Core Cloud Concepts

Definition – Basic Terms

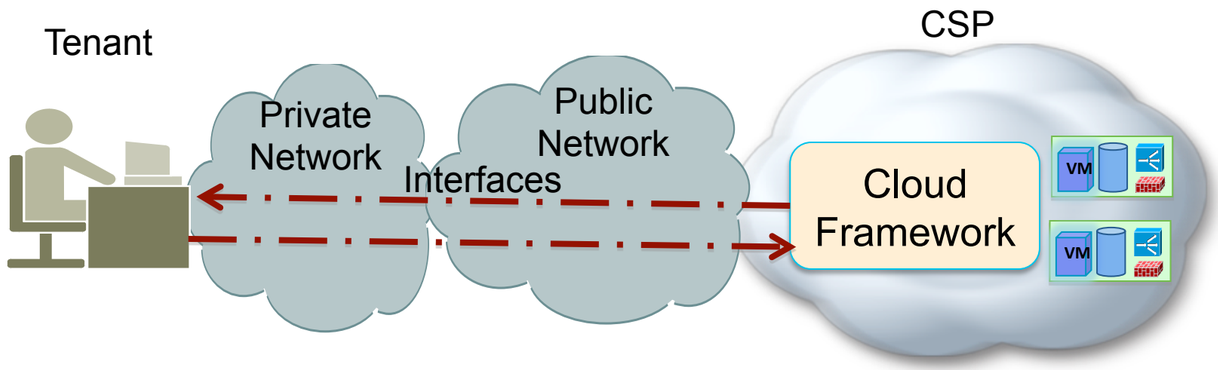
- Cloud Users or Tenants or Cloud Service Consumers: consumes resources from Cloud
  - Enterprises
  - End Consumers
  - Tenant Admins
  - Tenant end-users
- CSP: Cloud Service Provider: Offers Cloud services



- Cloud Resources: Compute, Storage, Network
- CRUD (Create / Read / Update / Delete) of resources
- API (Application Program Interfaces)
  - Includes Tenant facing Service Interfaces
- Cloud Framework: A (software) system, can be Distributed, that controls all aspects of a Cloud deployment
  - Managed by CSP Admins

Definition – Basic Terms

- Network: Without network there is no Cloud
  - From Tenant perspective Cloud is somewhere there over the network
  - Multiple segments of networks, but at a high level Cloud accessed over
    - Private Network: End Consumer home network or Enterprise intranet
    - Public Network: Network SP Private MAN/WAN and/or Internet
- Elements of network can themselves be Cloud resources



Definition – Basic Terms - Distinguishing Features of Cloud: OREMAP

- On-demand CRUD of resources
  - Anytime tenants want resources, it should get it
- Rapid CRUD of resources
  - Requested resources should be CRUD by CSP in seconds to minutes
- Elastic at any scale
  - Tenants should be able to Grow / shrink resource count / usage at any scale
- Multi-tenancy
  - CSP should support sharing of physical resources by totally separate organizations / entities
  - Tenants should be isolated from each other
- Abstractions
  - Need to support multiple and complex technologies for same resource, but users should be shielded from the diversity and complexity
  - Cloud facilitates outsourcing of IT functionalities → IT admins not in full control of resources / infrastructure anymore (specially in case of public / hybrid/ community / seamless Cloud, rather the CSP → provide abstracted view of resources / infrastructure
  - Multiple users, multiple systems → different levels of visibility and access
- Pay-per-use
  - Tenants are billed and pay per resource usage

Core Cloud Service Types

- Cloud resources are accessed via services
- Depending on resource types there can various types of services
- Core service types
  - IaaS: Infrastructure as a Service
    - Main focus of this tutorial
  - PaaS: Platform as a Service
  - SaaS: Software as a Service

IaaS

- Infrastructure elements, logical and physical, are offered as on-demand resources
- Typical Infrastructure resources
  - Virtual Computer or VM
  - Virtual storage or storage objects
  - Virtual Network elements or components or functions

- IaaS CSP, example:
  - Amazon EC2, S3
  - Verizon Terremark
  - RackSpace
  - Microsoft Azure
- Infrastructure vendors (such as Cisco) provides infrastructure, frameworks and systems used by CSPs to deploy IaaS
- Open source Cloud Framework OpenStack with its various controllers facilitates deployment of IaaS

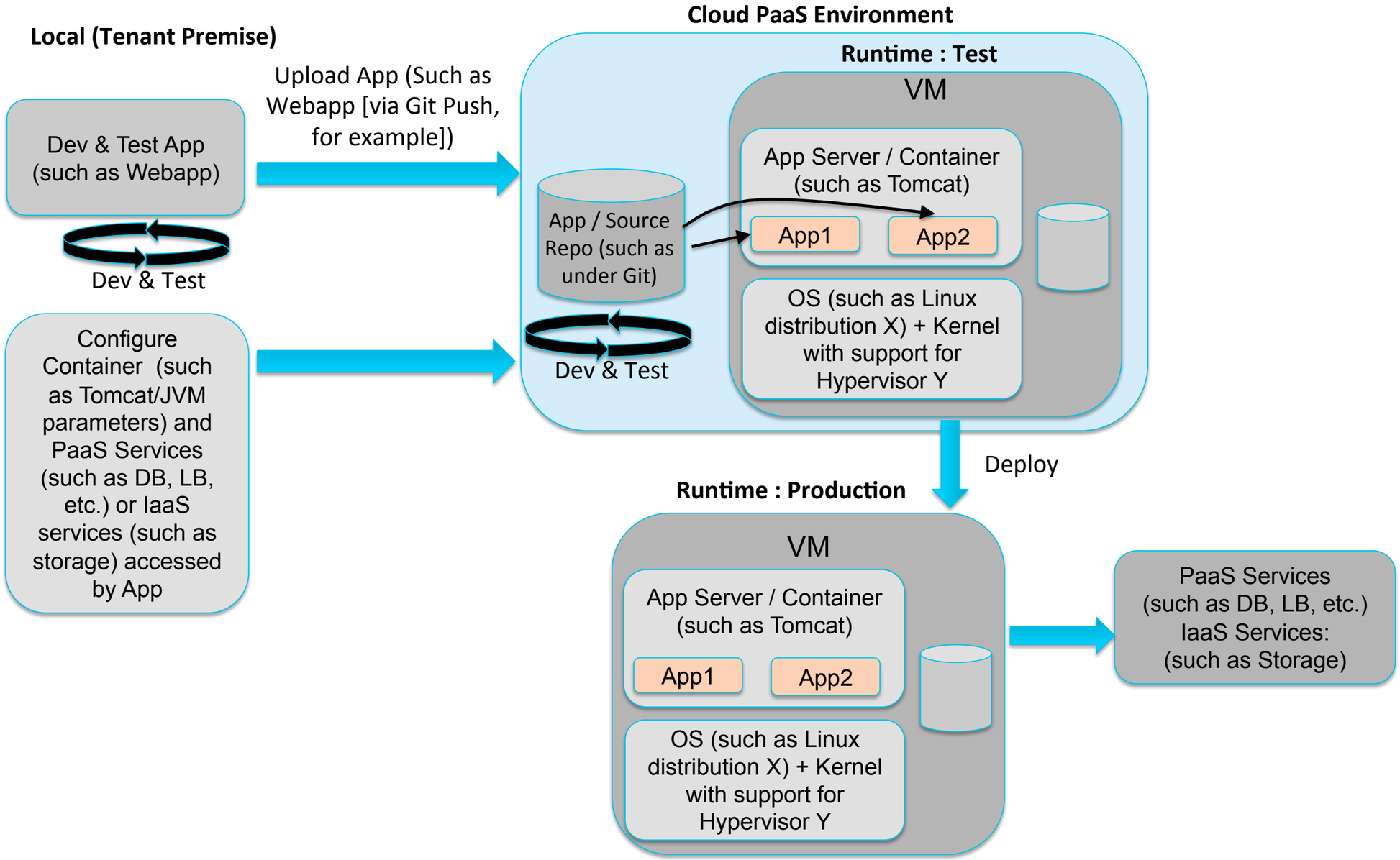
PaaS

- Software platforms are provided as services
  - Application servers
    - Example: Apache Tomcat, Red Hat Jboss, IBM WebSphere
  - Middleware platforms or services
    - Messaging or Queue service, DB service (SQL or so-called no-SQL or Big-data)
- Users upload code, which is executed on application servers and may interface with middleware services

- Full software / application development life cycle may be supported in a PaaS Cloud
  - Source control, build, test, etc.
- Underlying infrastructure resources (VM, load-balancer, storage, etc.) are acquired by PaaS system as needed
- Users do not have to acquire, buy and manage app server, middleware and dev & test platforms

- PaaS CSPs, Example:
  - Microsoft Azure PaaS
  - Google App Engine
  - VMWare Cloud Foundry
  - Amazon Elastic BeanStalk
- Application developed for one CSP PaaS will typically not be portable as is to another CSP PaaS

PaaS


Local (Tenant Premise)
Dev & Test App (such as Webapp)
Dev & Test
Configure Container (such as Tomcat/JVM parameters) and PaaS Services (such as DB, LB, etc.) or IaaS services (such as storage) accessed by App
Upload App (Such as Webapp [via Git Push, for example])
Cloud PaaS Environment
Runtime : Test
VM
App Server / Container (such as Tomcat)
App1
App2
App / Source Repo (such as under Git)
OS (such as Linux distribution X) + Kernel with support for Hypervisor Y
Dev & Test
Deploy
Runtime : Production
VM
App Server / Container (such as Tomcat)
App1
App2
OS (such as Linux distribution X) + Kernel with support for Hypervisor Y
PaaS Services (such as DB, LB, etc.)
IaaS Services: (such as Storage)

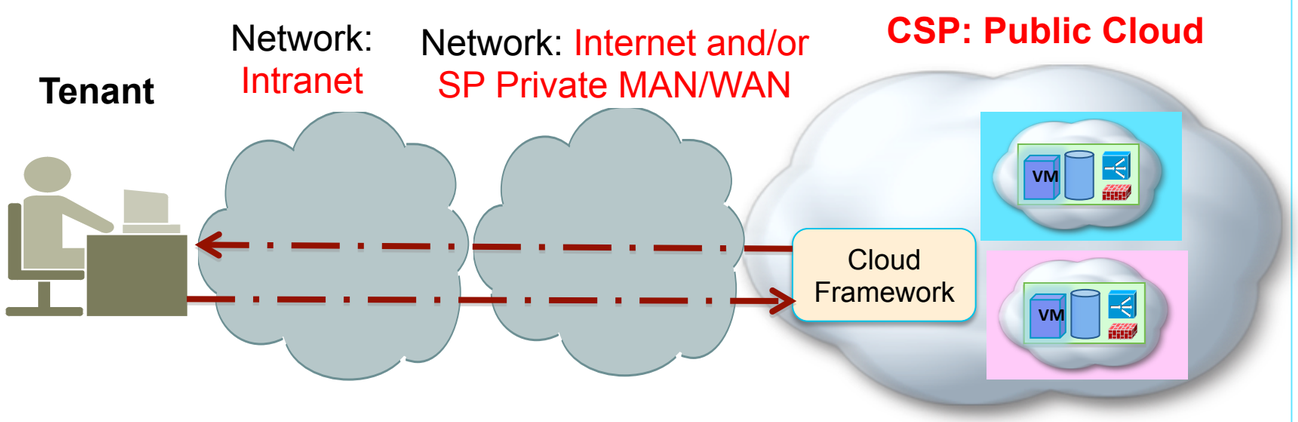
SaaS

- SaaS: Software as a Service
- Full-fledged finished application products typically used by end-users or developers are offered as services
  - Email, Calendaring, CRM, Document service

- Users (developers) can customize application *instance* using SDK and APIs provided by a SaaS provider
- SaaS provided by one provider can be interfaced with another
  - SalesForce and Google interface (CRM+Gmail)
  - VMForce: VMware + SalesForce in SalesForce Cloud

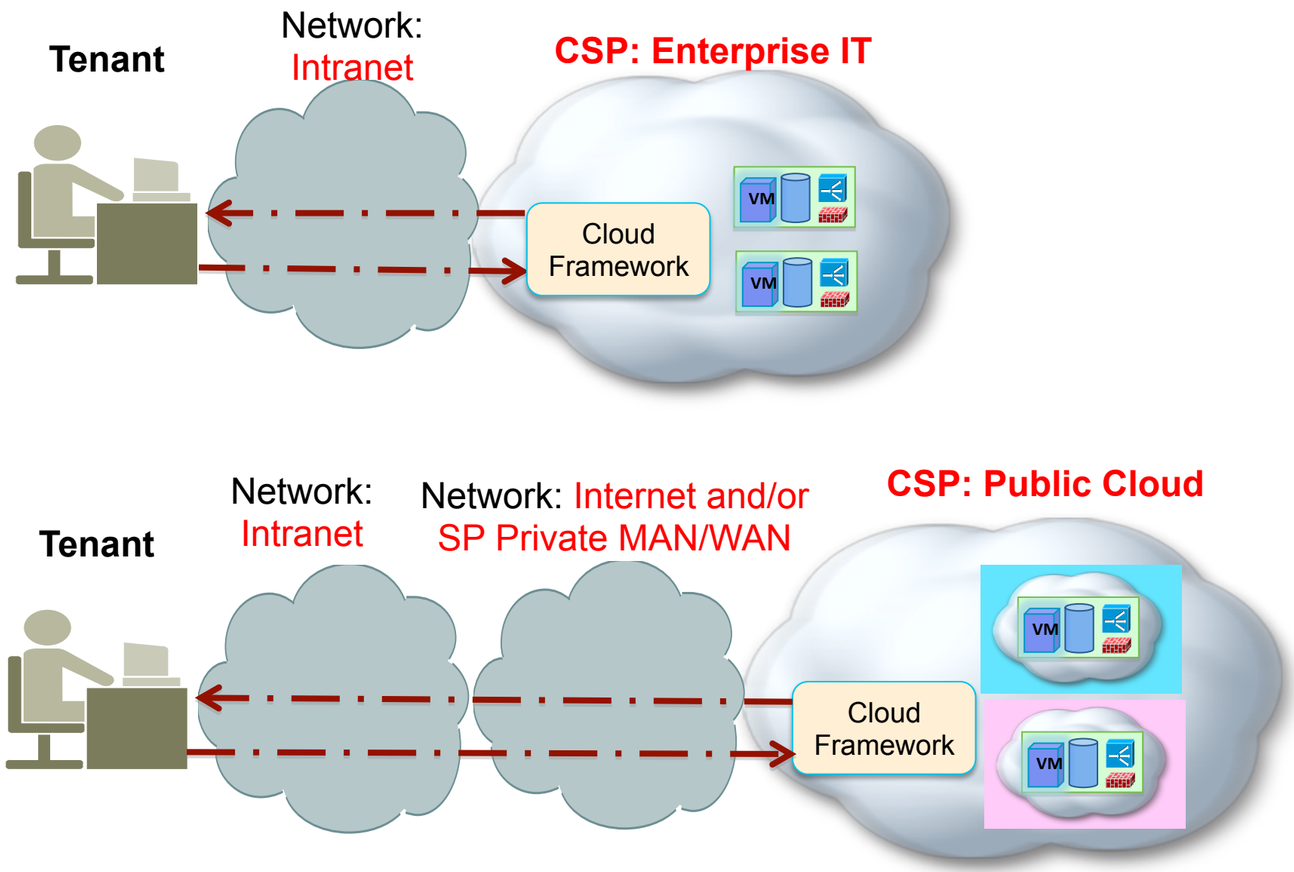
- Cisco Webex: Collaboration SaaS
- Cisco Hosted Unified Communication

Cloud Deployment Models – Public Cloud



- NIST definition: *The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.*
- A multi-tenant Cloud for use by any tenant

Cloud Deployment Models – Private Cloud



- NIST definition: *The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises*
- Enterprise IT (DCs) becomes CSP
- Can be a private Cloud that is virtual hosted in a Public Cloud managed by enterprise IT or CSP

Cloud Deployment Models – Community Cloud

- NIST definition: *The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.*

- More like *extranet* VPN for shared use

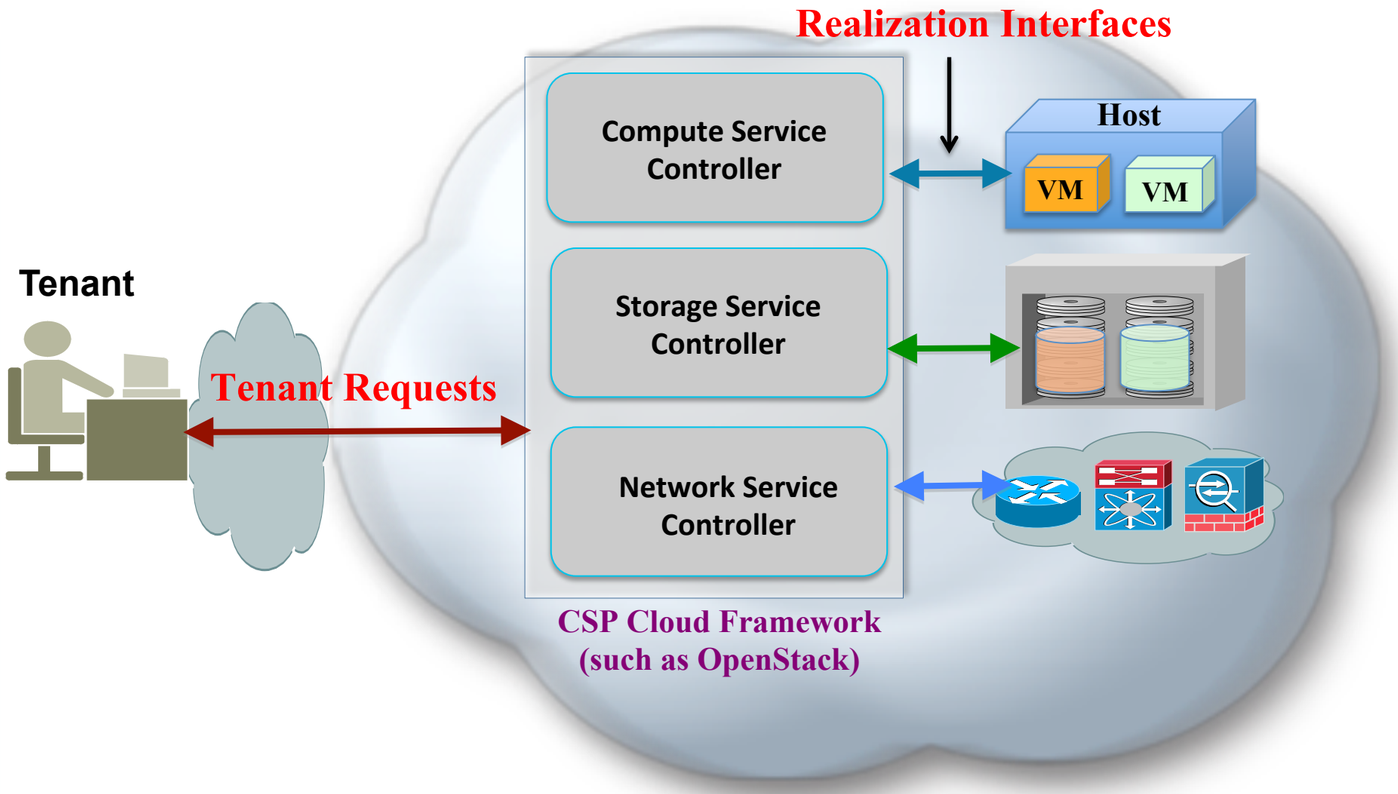
Cloud Deployment Models – Hybrid Cloud

- NIST Definition*: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are **bound together** by standardized or proprietary technology that enables **data and application portability** (e.g., cloud bursting for load balancing between clouds)*

- Note, multiple Clouds can be *bound together* at network level without the need for *data and app portability*

Cloud Controller / Frameworks / Systems

- Cloud Frameworks or Controllers performs all *Cloud Functions*
  - Fulfill tenants' Cloud resource CRUD requests in the Cloud infrastructure
  - Monitoring, accounting of resource usage and service assurance
- Controllers can be distributed and multiple Distributed systems may be interfaced with each other
- Multiple interfaces at multiple levels may be invoked on-demand to perform the Cloud Functions

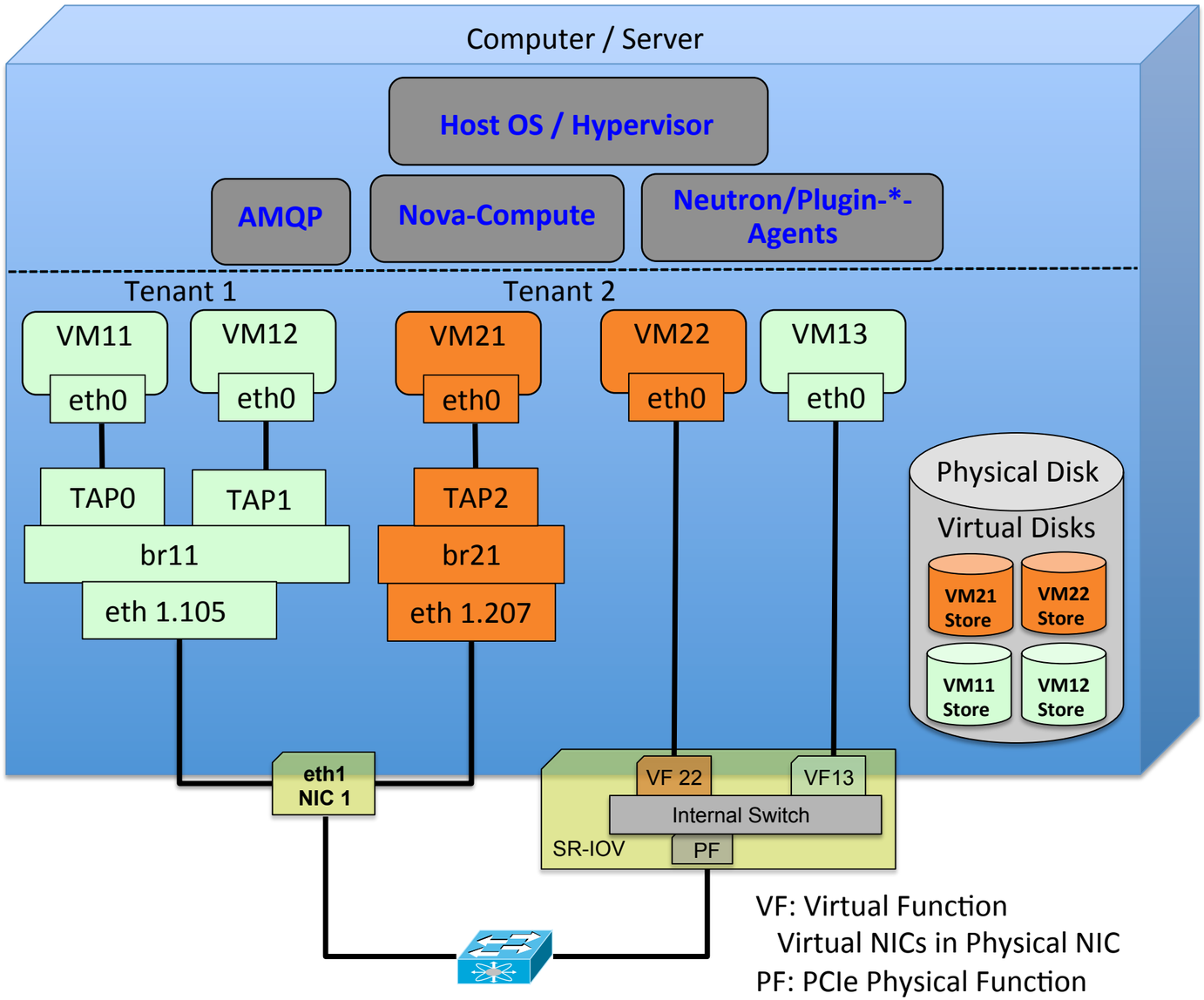


High-level simplified architecture

What Facilitates OREMAP: Virtualization is Key

- In a typical environment a physical or logical device or element is configured or provisioned for use (such as a server or network element)
- In a Cloud, a whole resource should be available on-demand, rapidly and elastically CRUD with multiple tenants sharing a physical resource
  - A whole computer, storage, network element
  - Not just plain configuration of a physical or logical element
- The best possible way to achieve this is via **virtualization** of resources
  - A virtual computer or virtual machine (VM) with everything in it that a physical computer may have
  - Similarly, storage and network

Element Virtualization

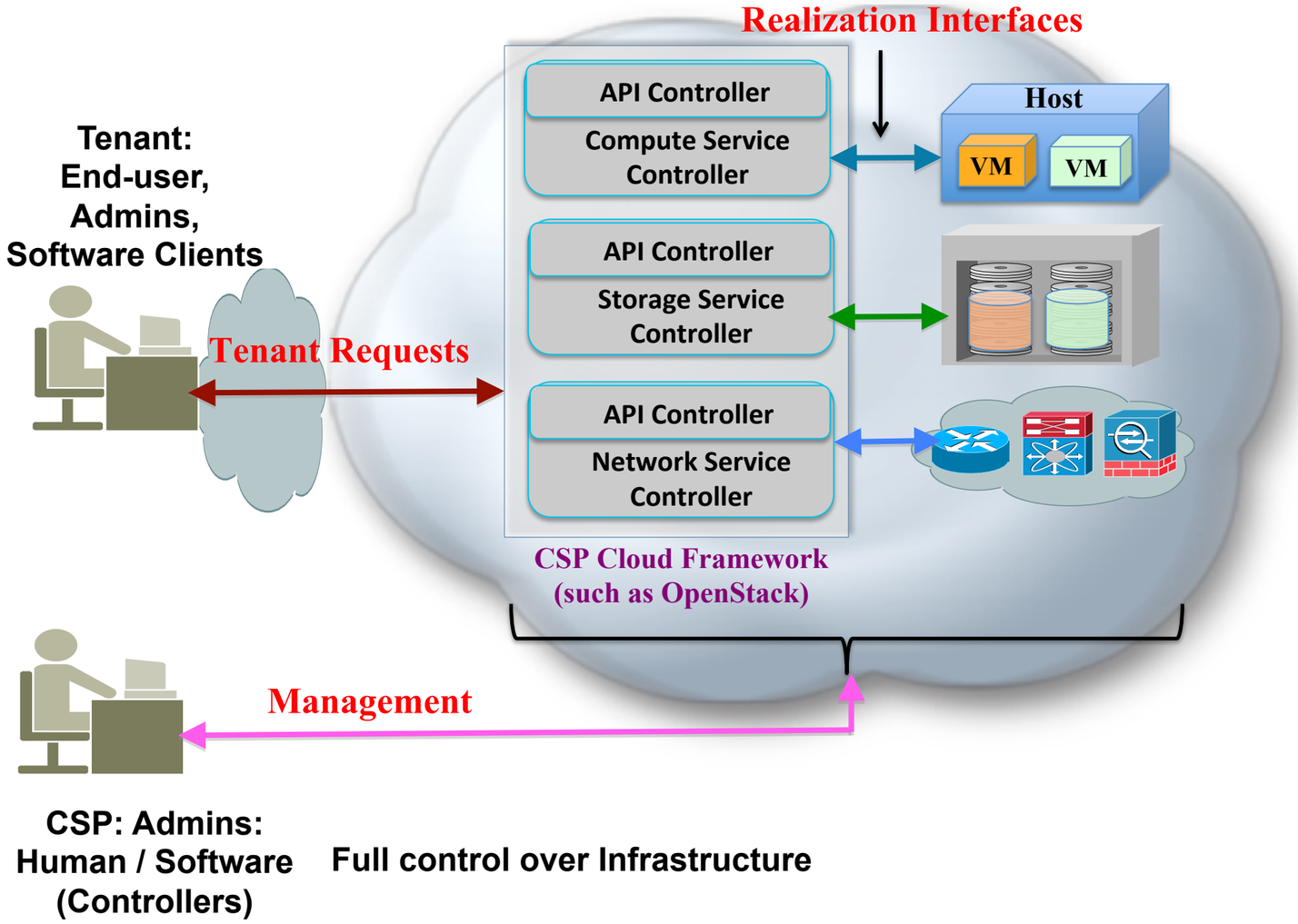


- Computer: VM
- Storage: Virtual Disk, Storage Objects
- Network: Software based Virtual Bridge / Switch / NIC
  - More later

Cloud Abstraction: Users Perspective: Tenant and CSP Users

"All problems in computer science can be solved by another level of indirection."
- Prof David Wheeler, U Cambridge

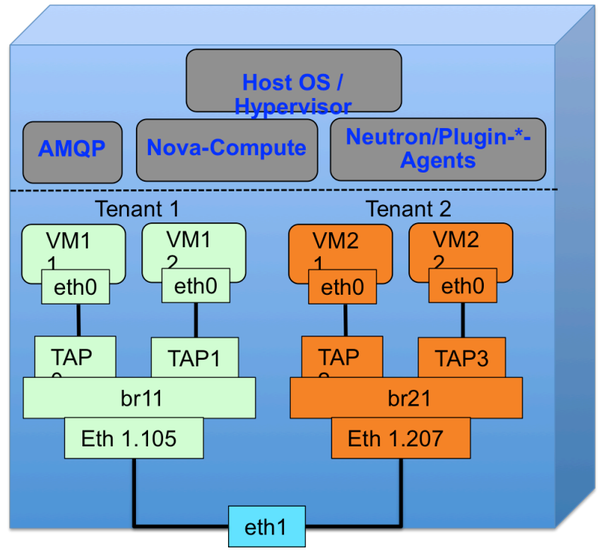
- Multiple levels of users, both human and software
- Tenant level
- CSP level
- There can be multiple admins: compute, storage, network, security, DC, WAN, etc.
- These users have layered access (at different abstraction level)



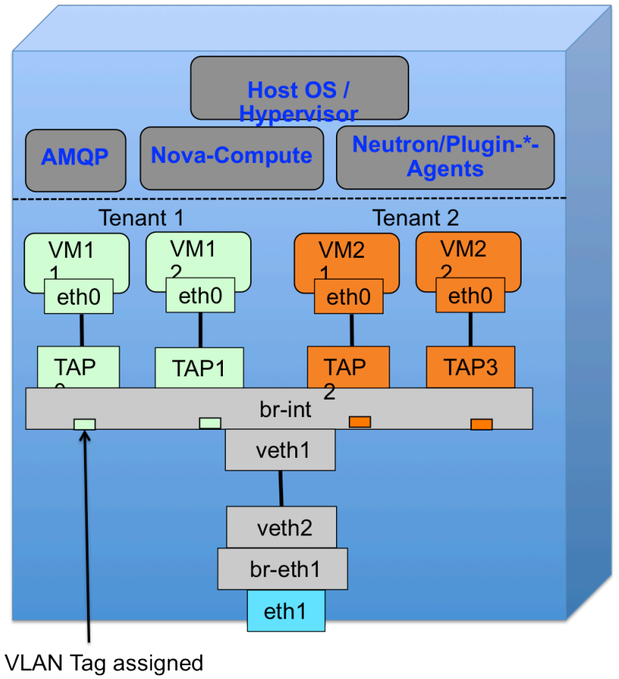
Cloud Resource Abstraction: Tenant facing

- CSP controlled Resources / infrastructure are abstracted and exposed to tenants
- Abstractions are defined by CSP / Controllers (actually Cloud Controller developers / vendors)
- KVM, Xen, VMware, Hyper-V, etc. VM

  → One abstraction
- Linux bridge, OVS, Nexus 1K, VLAN, VxLAN, VRF → One or two abstractions



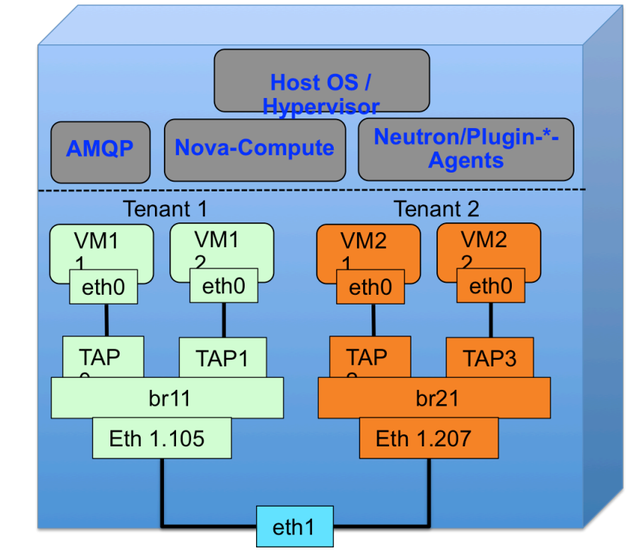
Linux Bridge



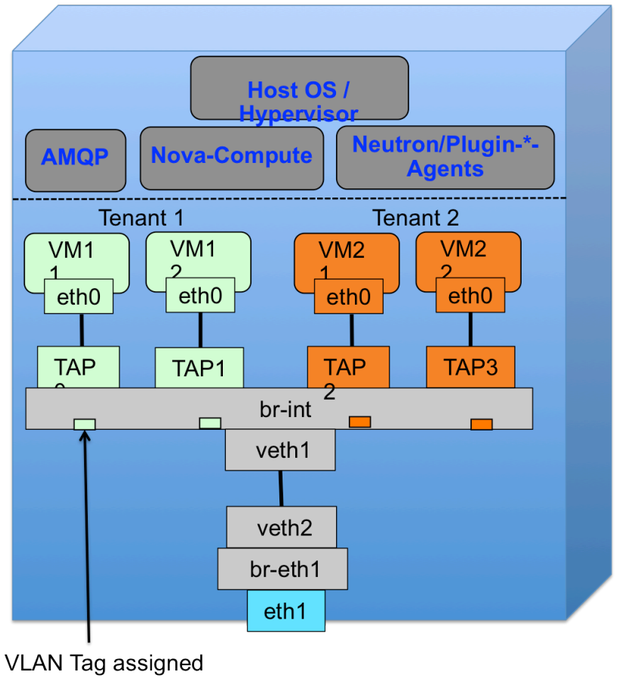
OVS

Cloud Resource Abstraction: Tenant facing

- Tenant Admin (TA) can define abstractions
  - For example, TA can extend or create new VM flavors (m1.small, m1.large etc.) if CSP allows or supports
- TA can CRUD certain abstract resources, Tenant end-users cannot
  - For example, CRUD Virtual Networks



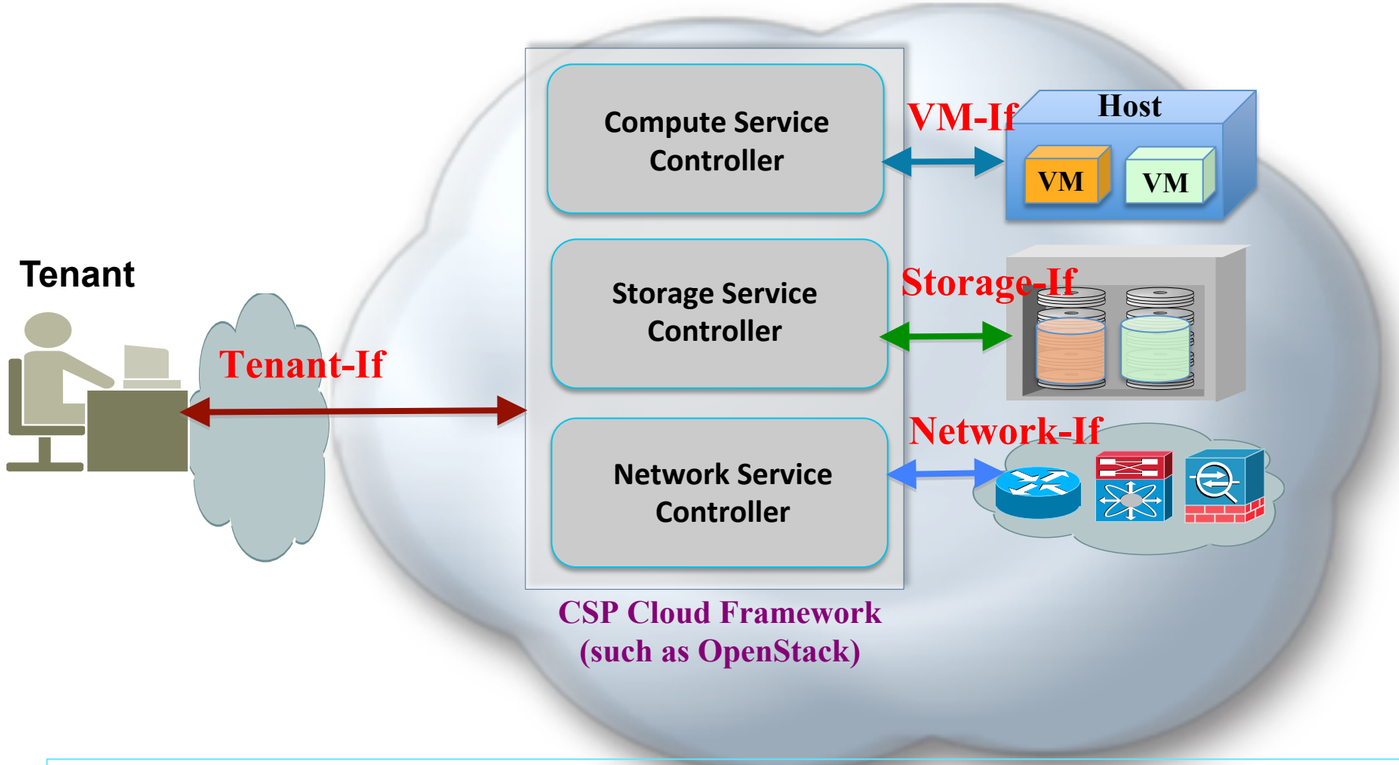
Linux Bridge



OVS

On-demand: On-demand Capable Interfaces

- To support on-demand characteristics of Cloud, *On-demand capable* interfaces are needed at all levels of Cloud Frameworks and infrastructure
- *Programmable* interfaces are on-demand capable



Tenant-If: Tenant facing [RESTful] IaaS Compute, Storage, Network, or PaaS / SaaS Interfaces or API

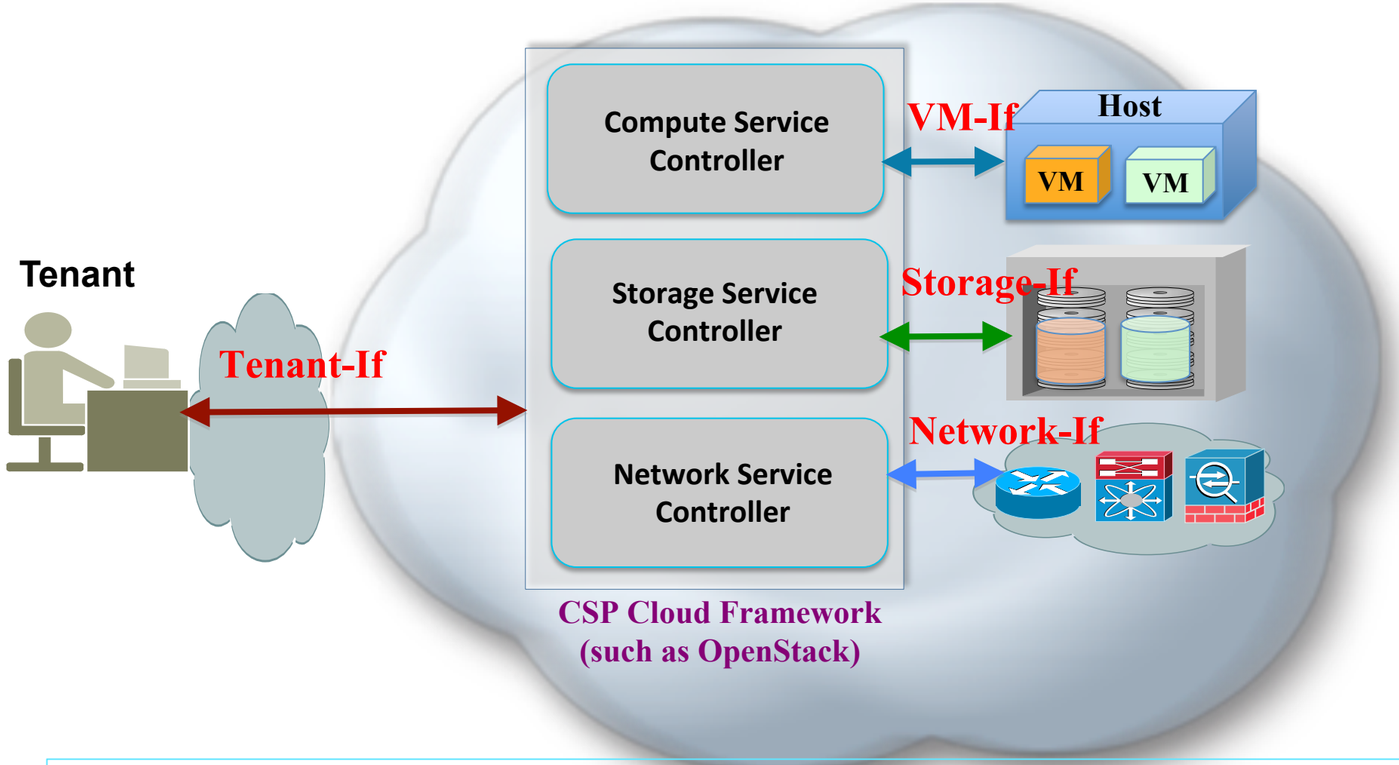
VM-If: Libvirt (KVM/QEMU) or Xen or VMWare or Hyper-V API for VM Life-cycle Management interfaces

Storage-If: Low level storage management and control interfaces

Network-If: OS (such as Linux) Networking or NM or Signaling or OpenFlow or Cisco OnePK Interfaces

On-demand: On-demand Capable Interfaces

- Tenants interacts with the Cloud using the Tenant facing Interfaces (Tenant-If)
  - To request CRUD of resources
  - Other Cloud functions
  - These interfaces incorporate tenant facing abstractions
- Infrastructure Interfaces
  - CSP Cloud Framework realizes tenant requests (via Tenant-If) in the infrastructure
  - Cloud framework invokes on-demand infrastructure level interfaces (VM-if, Storage-if, Network-If)
  - These interfaces may support relevant abstractions



Tenant-If: Tenant facing [RESTful] IaaS Compute, Storage, Network, or PaaS / SaaS Interfaces or API

VM-If: Libvirt (KVM/QEMU) / Xen / VMWare / Hyper-V API for VM Life-cycle Management interfaces

Storage-If: Low level storage management and control interfaces

Network-If: OS (such as Linux) Networking / NM / Signaling / OpenFlow / Cisco OnePK / Openstack Neutron / Open Daylight Interfaces

On-demand Interfaces: RESTful API

- REST (Representational State Transfer) based API widely used specially for tenant facing interfaces
- OpenStack Tenant facing API Example:
  - CRUD: **POST/PUT/GET/HEAD** http://<Cloud Service End Point (Cloud Service IP address & Port)>/<API Version>/<tenant ID>/**<Resource>/<resource Instance ID> {<body>}**
  - Action: **POST/PUT/GET/HEAD** http://<Service IP address & Port>/<API Version>/<tenant ID>/ **<Resource>/<resource Instance ID>/action {<body>}**
- In the request body and response XML and/or JSON format supported

- Example: Virtual Server Creation (via Openstack API)
  - POST <Compute Controller URL1> /v2/<Tenant1 UUID1>/**servers**

```
{   "server" :  {
        "name" : "VM11",
        "imageRef" : <URL2>/images/<Image UUID1>,
       ”flavorRef" : <URL1>/flavors/1,
       "networks":
       [ {
            "uuid": NET1-UUID1,
             "fixed_ip": "10.0.0.10”  } ]   }
}
```

- Example: Virtual Server Boot Action (via Openstack API)
  - POST …/**servers/UUID1/action**

```
{ "reboot" : { "type" : "HARD" } }
<!– HARD: Power cycle;
     SOFT: signal to OS to restart -->
```

Cloud Abstraction, Tenant-If and Infrastructure Mapping: Linux Bridge

POST <URL1> v2/<tanant UUID1>/**servers**

{ "server" : { "name" : ”VM11",

"imageRef" : ”<URL2>/images/<Image-UUID1>,

”flavorRef" : <URL1>/flavors/1,

"networks": [{ "uuid": <NET1-UUID1>,
"fixed_ip": "10.0.0.10” }] } }

Nova

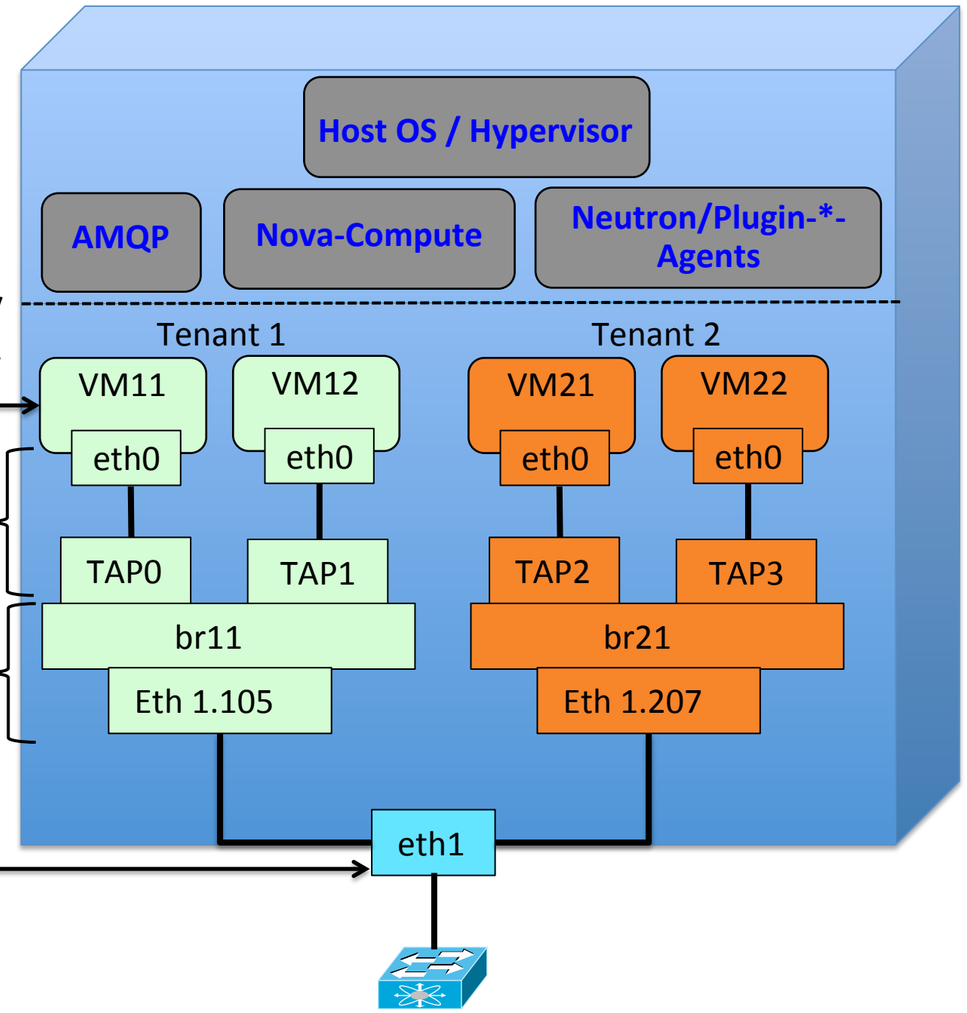
POST v2.0/**Ports**
{"port": {"name": "port1“, "admin_state_up": true,
"device_id": <VM-UUID1>, "network_id": <NET1-UUID1>} }

Neutron

POST v2.0/**Networks**
{"network”: { "name": ”NET1”, "admin_state_up": true,
"provider:network_type": "vlan",
"provider:physical_network": ”VNET1",
"provider:segmentation_id": 105 } }

Neutron

In linuxbridge_conf.ini:
[VLANS]
tenant_network_type = vlan
network_vlan_ranges = VNET1:100:110
[LINUX_BRIDGE]
physical_interface_mappings: VNET1:eth1



Cloud Abstraction, Tenant-If and Infrastructure Mapping: OVS

POST <URL1> v2/<tanant UUID1>/**servers**

{ "server" : { "name" : ”VM11",

"imageRef" : ”<URL2>/images/<Image-UUID1>,

”flavorRef" : <URL1>/flavors/1,

"networks": [{ "uuid": <NET1-UUID1>,
"fixed_ip": "10.0.0.10” }] } }

Nova

POST v2.0/**Ports**
{"port": {"name": "port1“, "admin_state_up": true,
"device_id": <VM-UUID1>, "network_id": <NET1-UUID1>} }

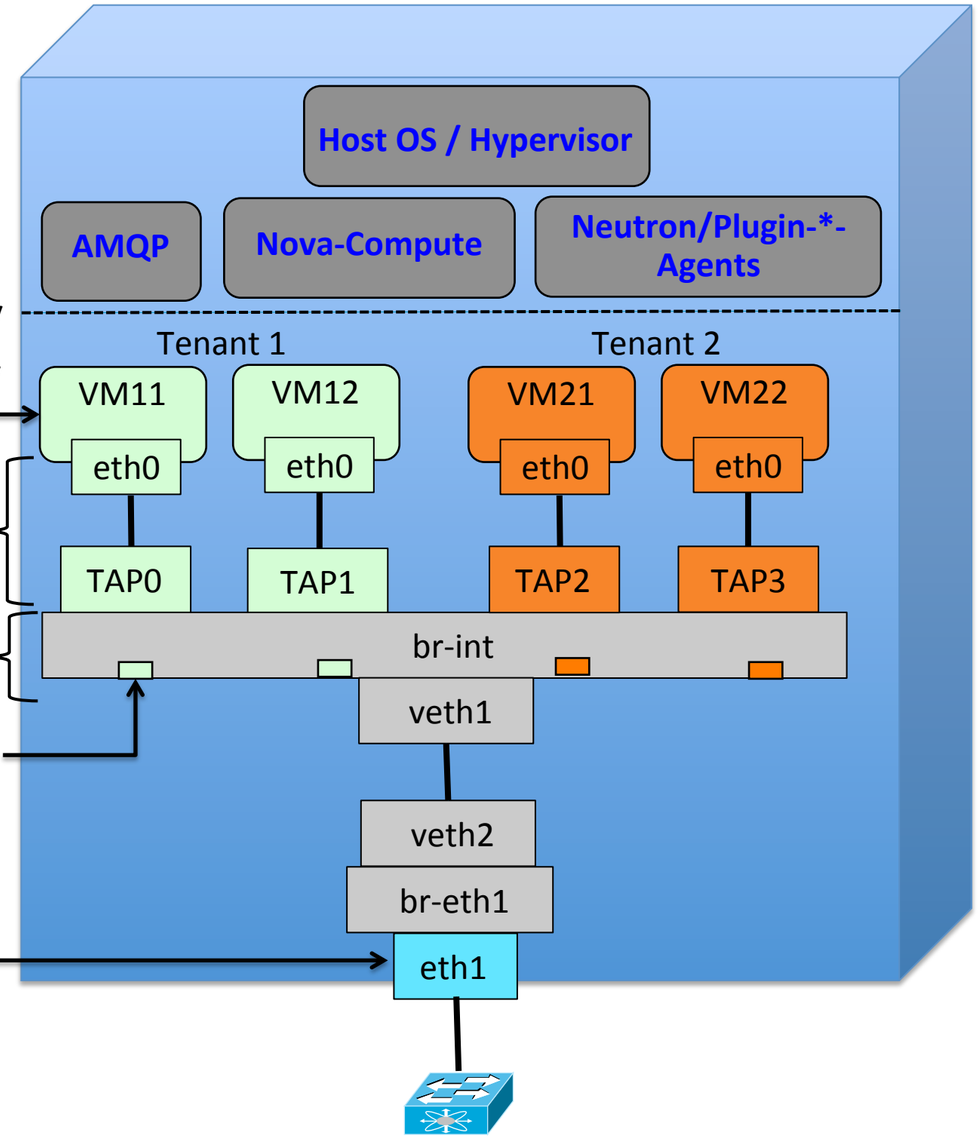
Neutron

POST v2.0/**Networks**
{"network”: { "name": ”NET1”, "admin_state_up": true,
"provider:network_type": "vlan",
"provider:physical_network": ”VNET1",
"provider:segmentation_id": 105 } }

Neutron

VLAN Tag assigned

In ovs_neutron_plugin.ini:
tenant_network_type = vlan
network_vlan_ranges =VNET1:100:110
integration_bridge = br-int
bridge_mappings = VNET1:br-eth1

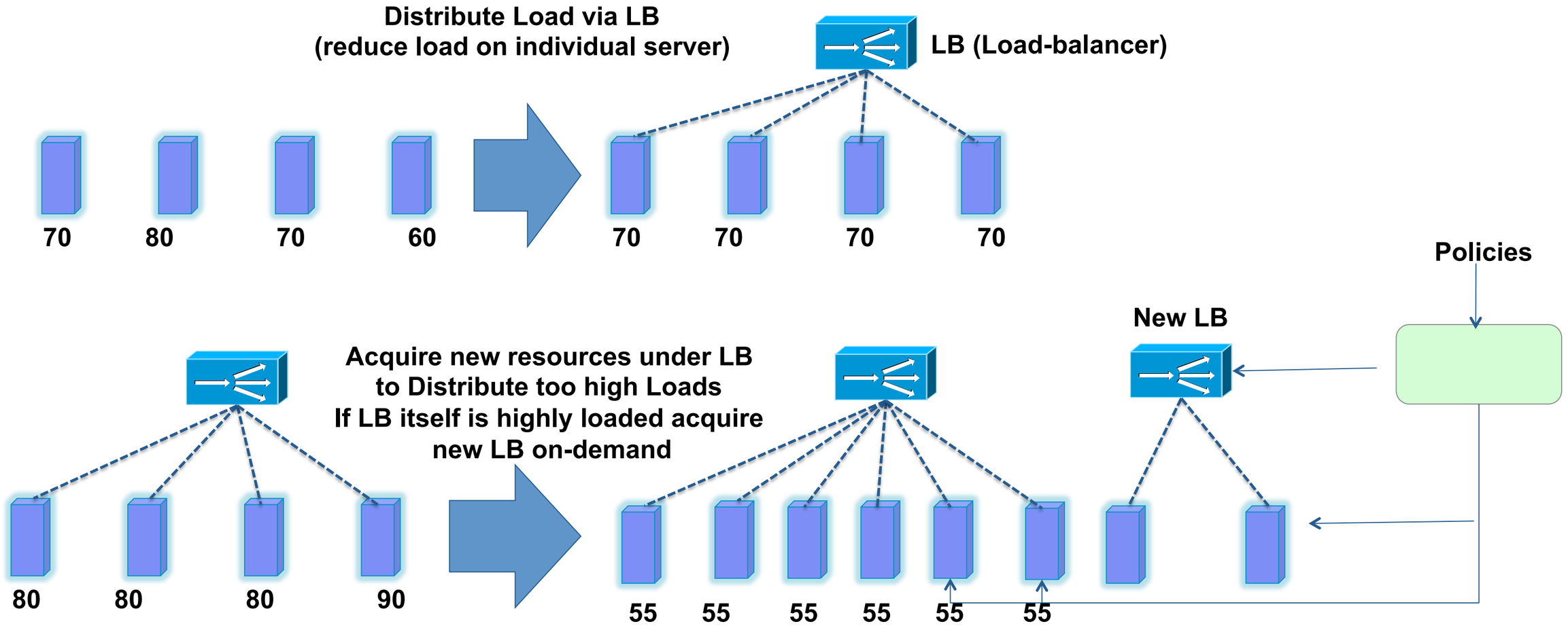


Elastic Scaling

- Cloud computing allows scaling (create / delete) of resources elastically
- When more resources are needed it is possible to ***scale up*** by acquiring more powerful compute, storage, network elements (hardware)
- But Cloud allows ***scaling out*** / **back** by incremental addition / deletion of virtual elements
  - Obviously a CSP still needs to acquire powerful hardware, but a tenant does not have to

Scaling Out Network Resources

- Not only compute and storage resources can be scaled out (typical focus) in a Cloud
- But also network resources



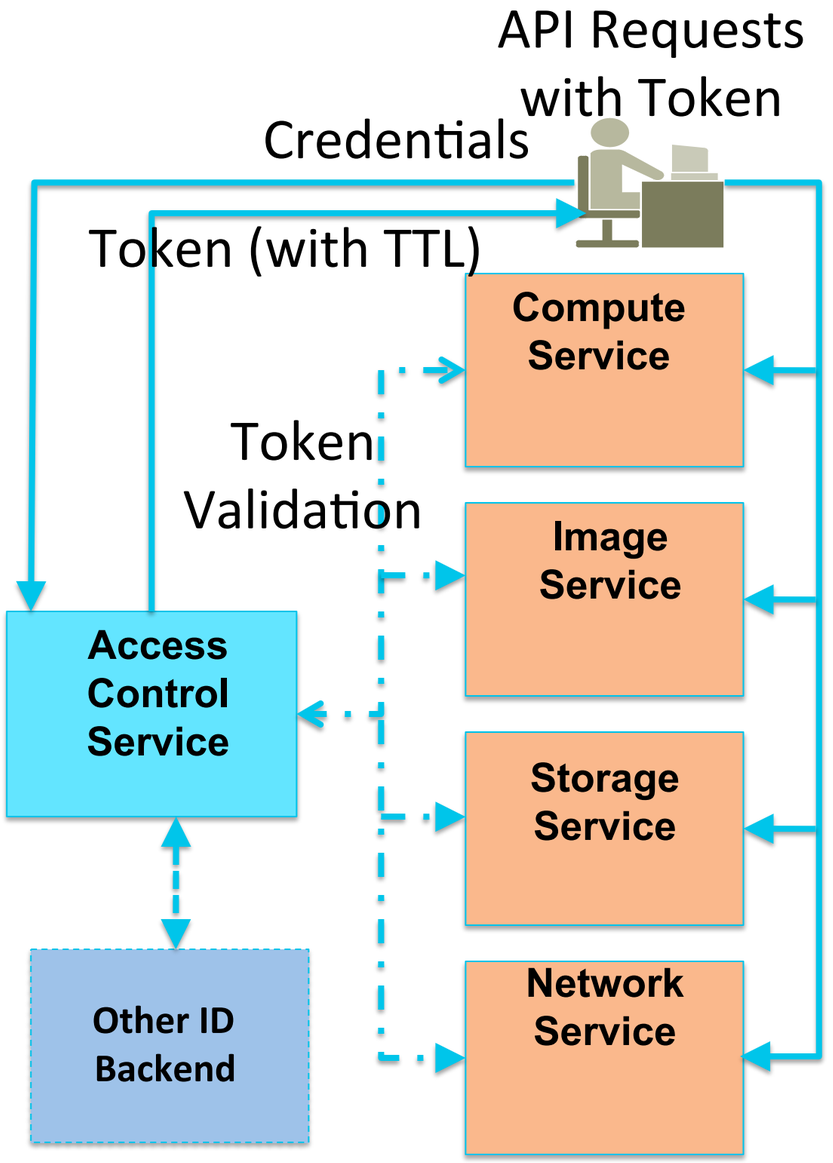
Access Control

- Access control in a Cloud has to be considered from multiple perspectives beyond user ID management and AA (authentication and authorization)

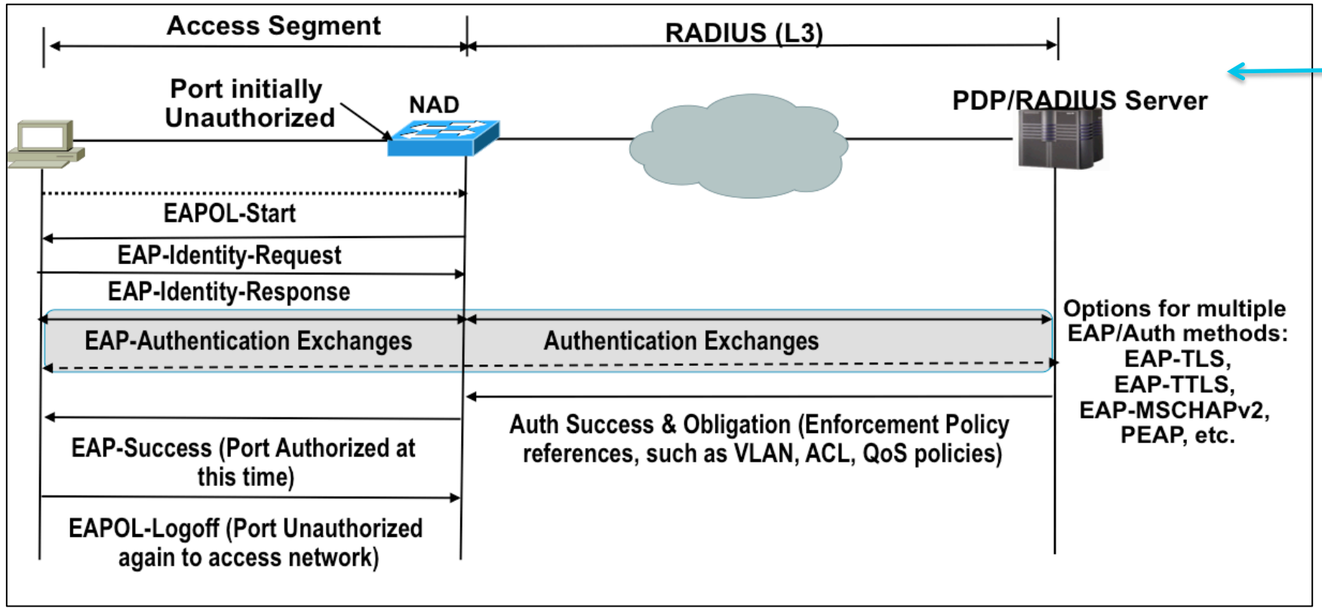
- Single sign-on is imperative in the presence of a multiple services that may be “touched” for a single CRUD action
  - For example, Looking at this example of virtual server creation, it is obvious multiple services are touched

```
POST <URL1> v2/<tanant UUID1>/servers
  { "server" : { "name" : ”VM11",
    "imageRef" : ”<URL2>/images/<Image-UUID1>,
    ”flavorRef" : <URL1>/flavors/1,
    "networks": [ {  "uuid": <NET1-UUID1>, "fixed_ip": "10.0.0.10” } ] } }
```

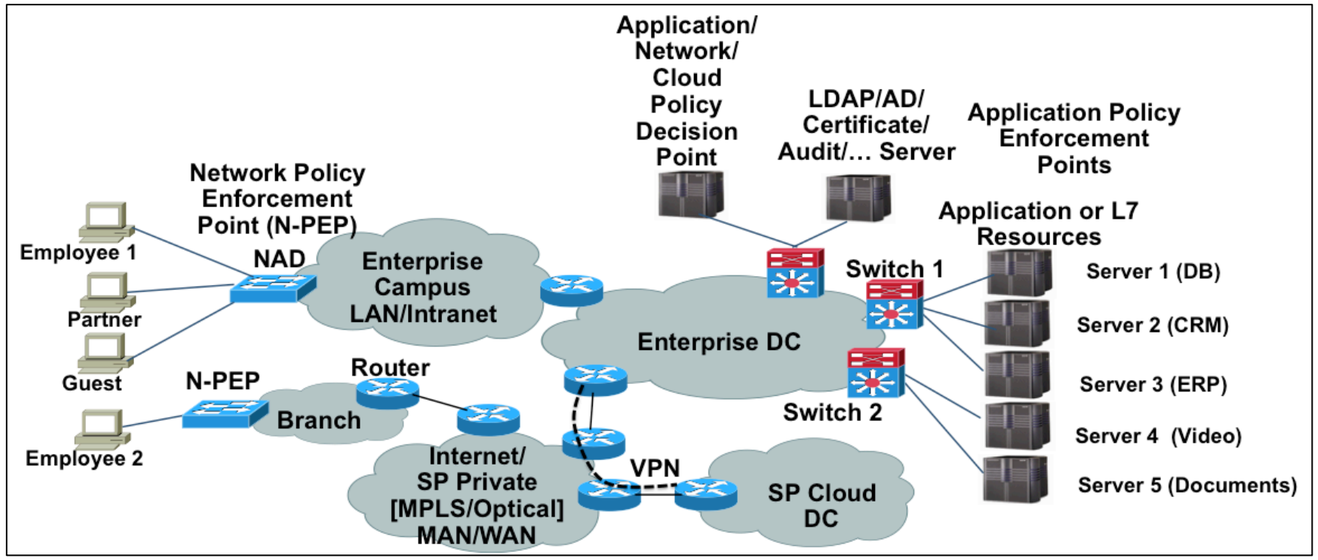
  - Originating service (compute in this case) should be able to authorize the user automatically across services
  - A Token based approach may be useful
- Role based: Example: An IaaS admin can create a virtual network, but a regular user cannot



Access Control: Public / Hybrid Cloud



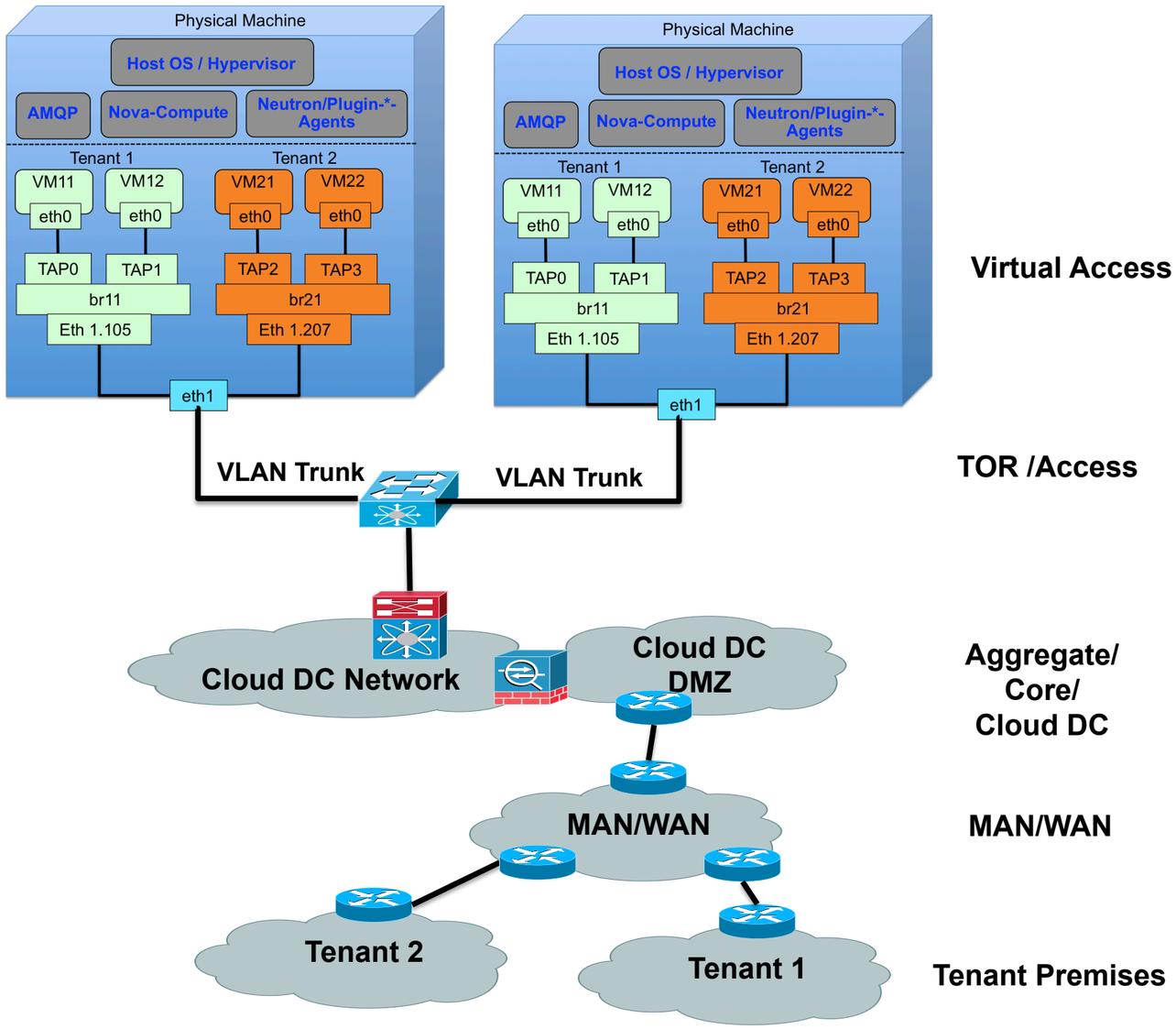
Typical network access control flow in an enterprise



- Network access control in enterprise should be integrated / interfaced with Cloud and Cloud resource access control

Cloud Networking
Network Virtualization

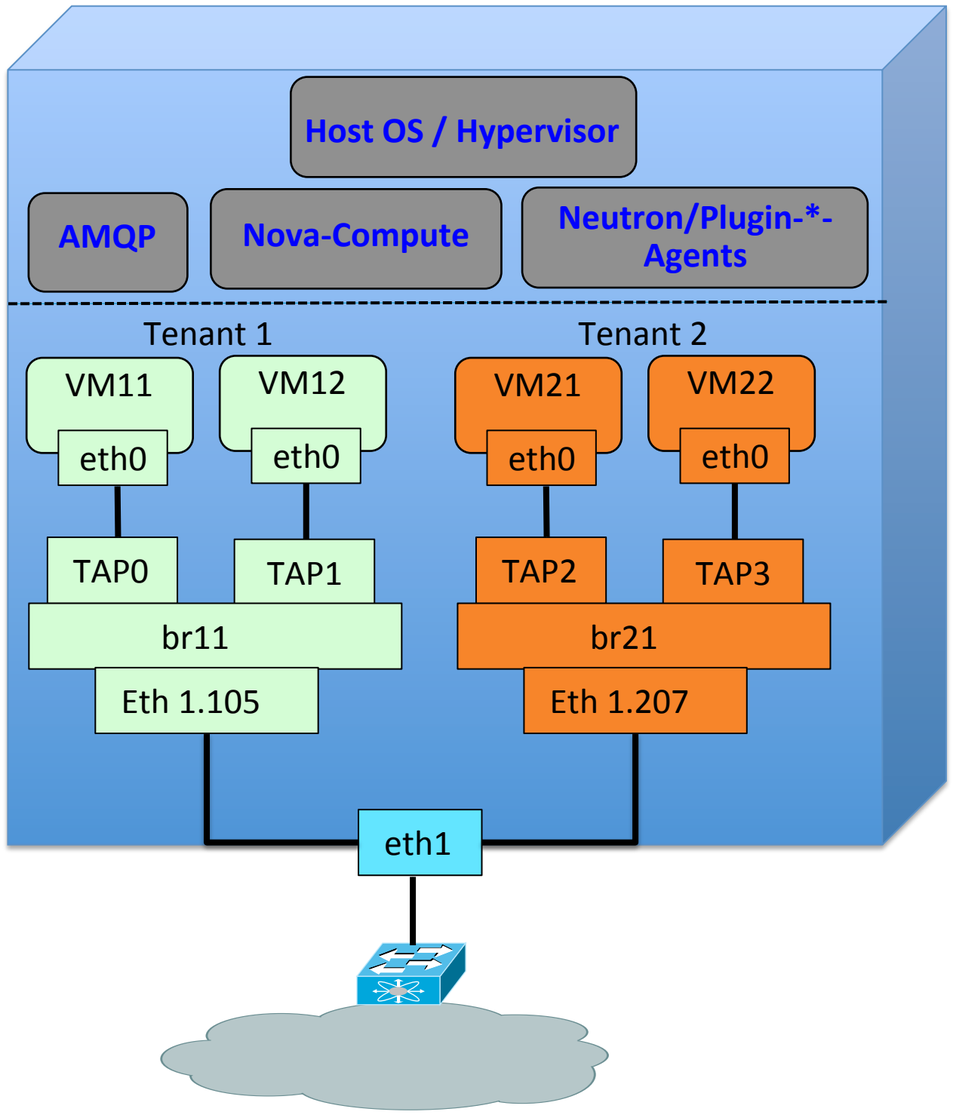
Introduction



- We will consider end-to-end (E2E) network, not just virtual network within computer

- Virtual Access Network
- Hardware Access Network or TOR (top of the rack)
- Cloud DC Network
- Cloud DC MAN/WAN Access
- MAN/WAN
- Tenant premises: Campus, DC

Virtual Access Network – Virtual Embedded Switch / Bridge



- With computer virtualization, *virtualized access switch* (software switch) moves within the physical computer
- Access ports/VLANs within the server
- Also known as *Virtual Embedded Bridge* (VEB)
  - Software based IEEE 802.1Q and relevant switching / bridging capabilities
  - Linux Bridge, OVS, Cisco Nexus 1K

Virtual Access Network – VEB to Hardware

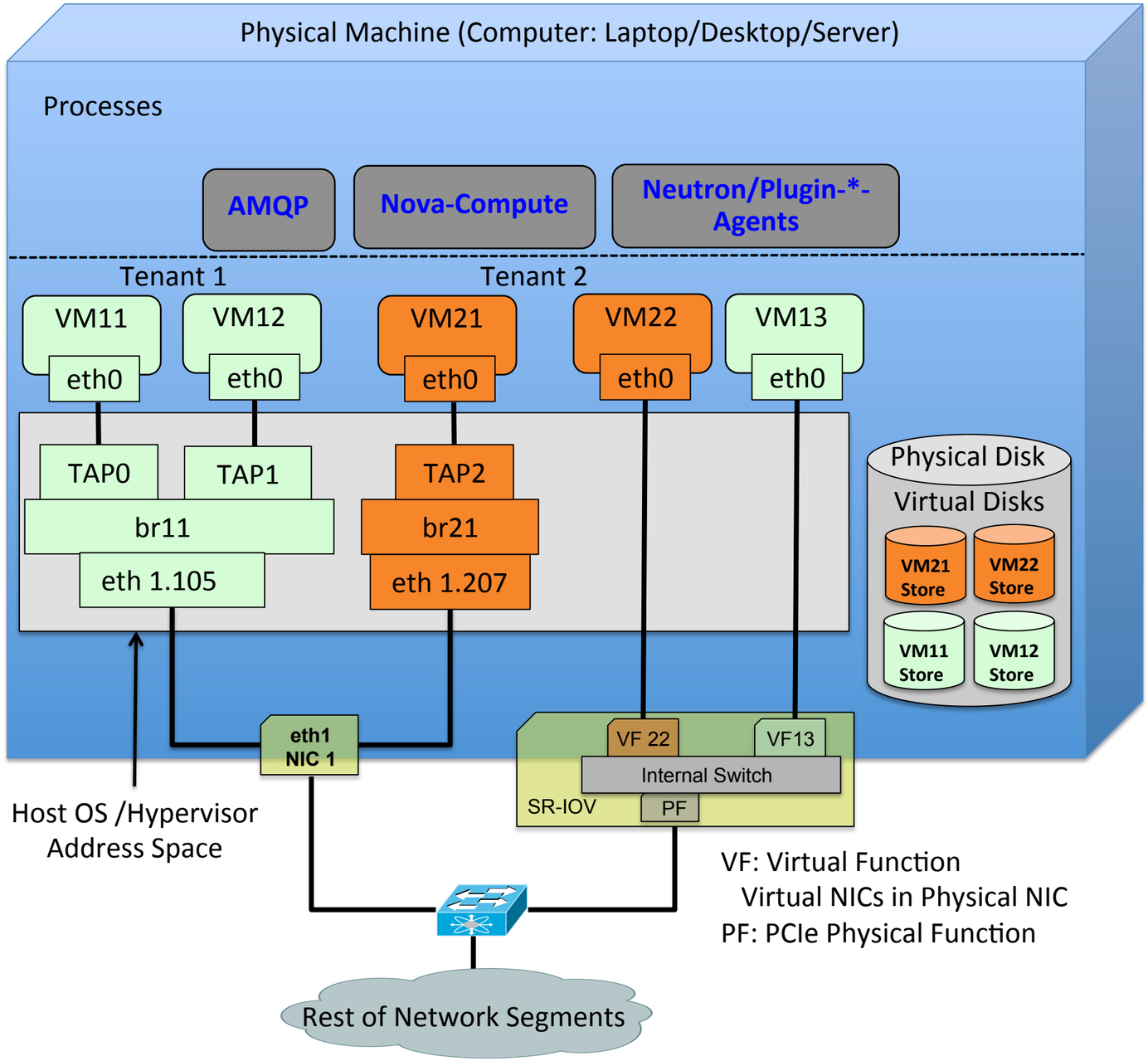
- Software based Virtual Switch has limited capabilities
- Software based switch adds extra load to host server
- I/O performance suffers

- Move VEB / Virtual Switch to Hardware

- In Hardware NIC on Computer with multiple Virtual NICs
  - SR-IOV (Single Root – IO Virtualization)
  - Cisco UCS M81KR VIC (Virtual Interface Card)

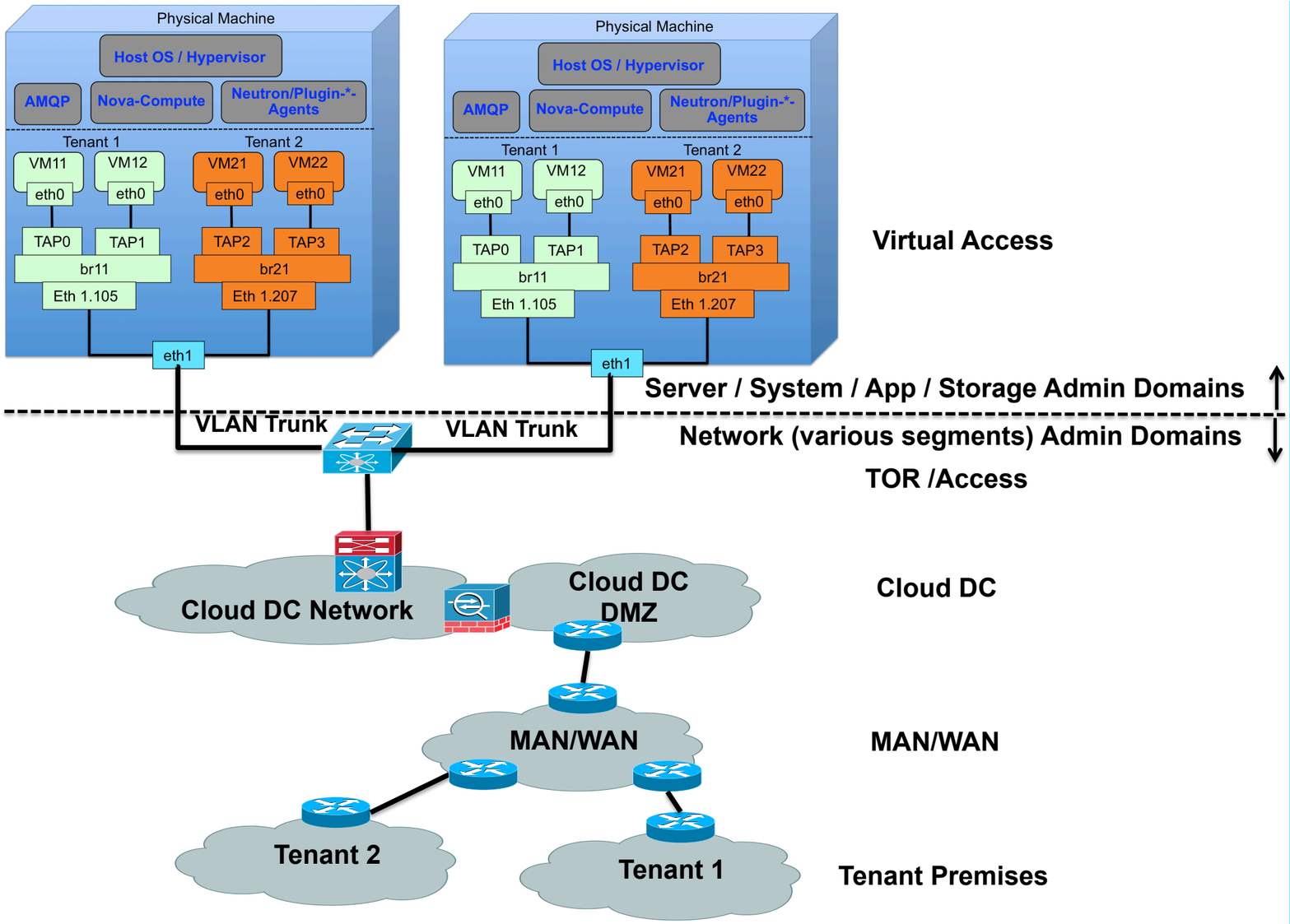
- To upstream Hardware Access or Aggregate Switch
  - IEEE 802.1BR (VBE: Virtual Bridge Port Extension; Cisco term: FEX / VM-FEX / VN-Tag)
  - IEEE 802.1qbg (Edge Virtual bridging)

Virtual Access Network – Direct IO Device Assignment



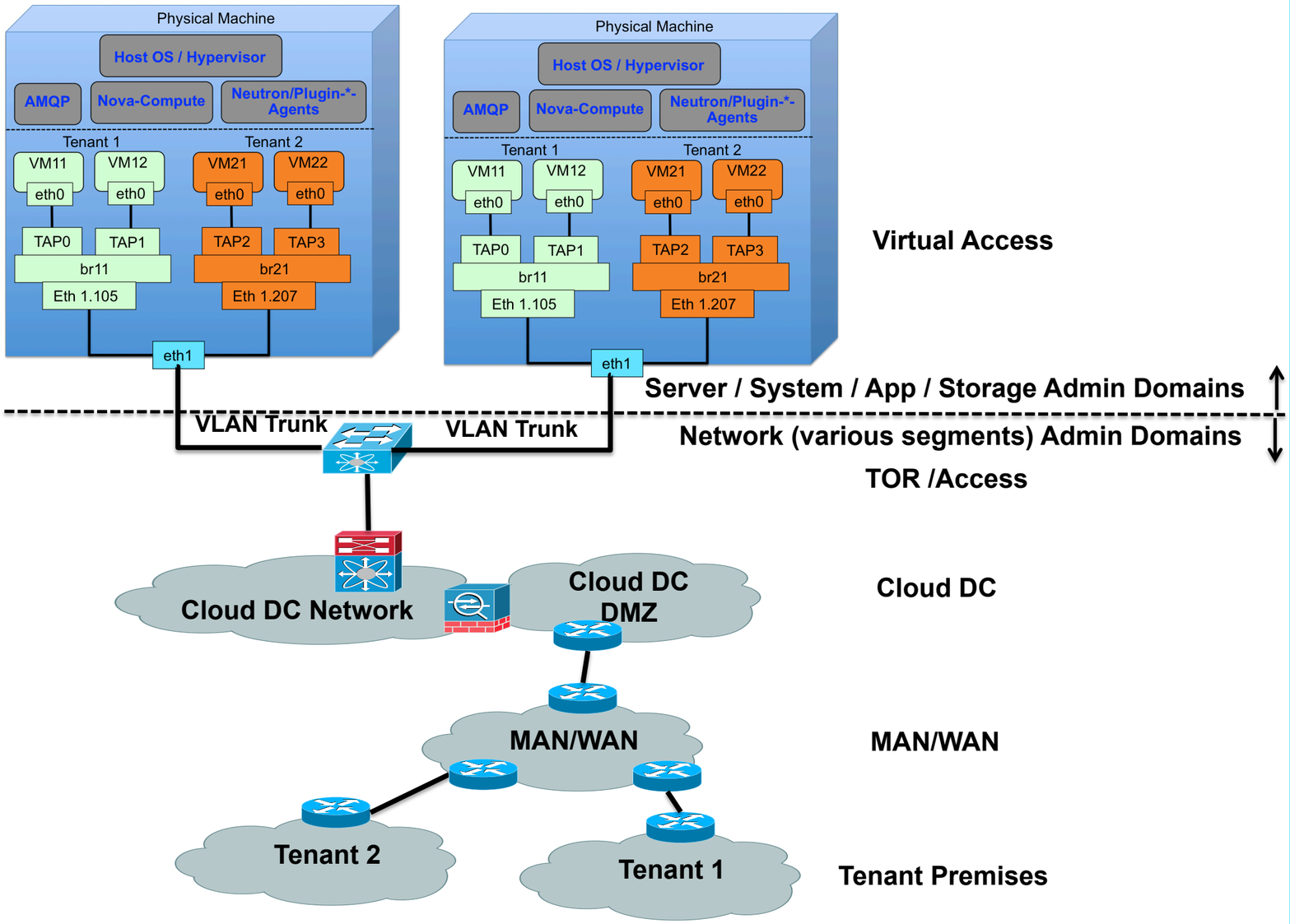
- Also known as *PCI-pass-through* or *Hypervisor-bypass*
- SR-IOV
  - Multiple Physical Functions (PF: up to 16 currently) with full PCIe capabilities
  - Multiple Virtual NIC / adapters or Virtual Functions (VF) with light-weight PCIe capabilities
  - Guest OS sees the associated VF as an attached PCIe device
- Need special-purpose hardware/chip, such as IOMMU (Intel VT-d, for example)
- Full hypervisor-bypass may not be possible, specially for interrupts → special solution needed

Issues with VEB



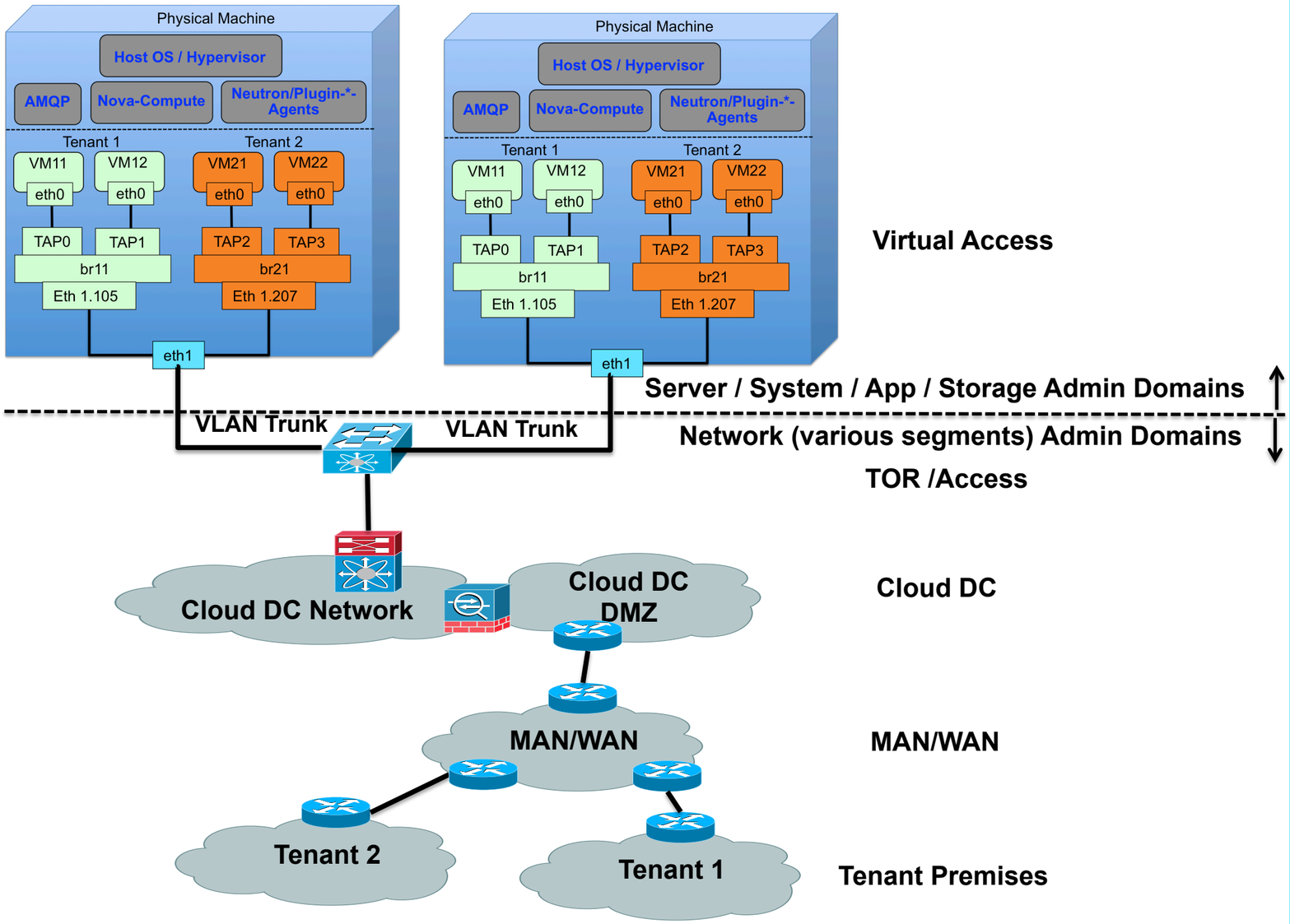
- An infrastructure typically has multiple *Admin Domains*
  - D1: Server / System / App / Storage Domains
  - D2: Network domains with its own subdomains

Issues with VEB



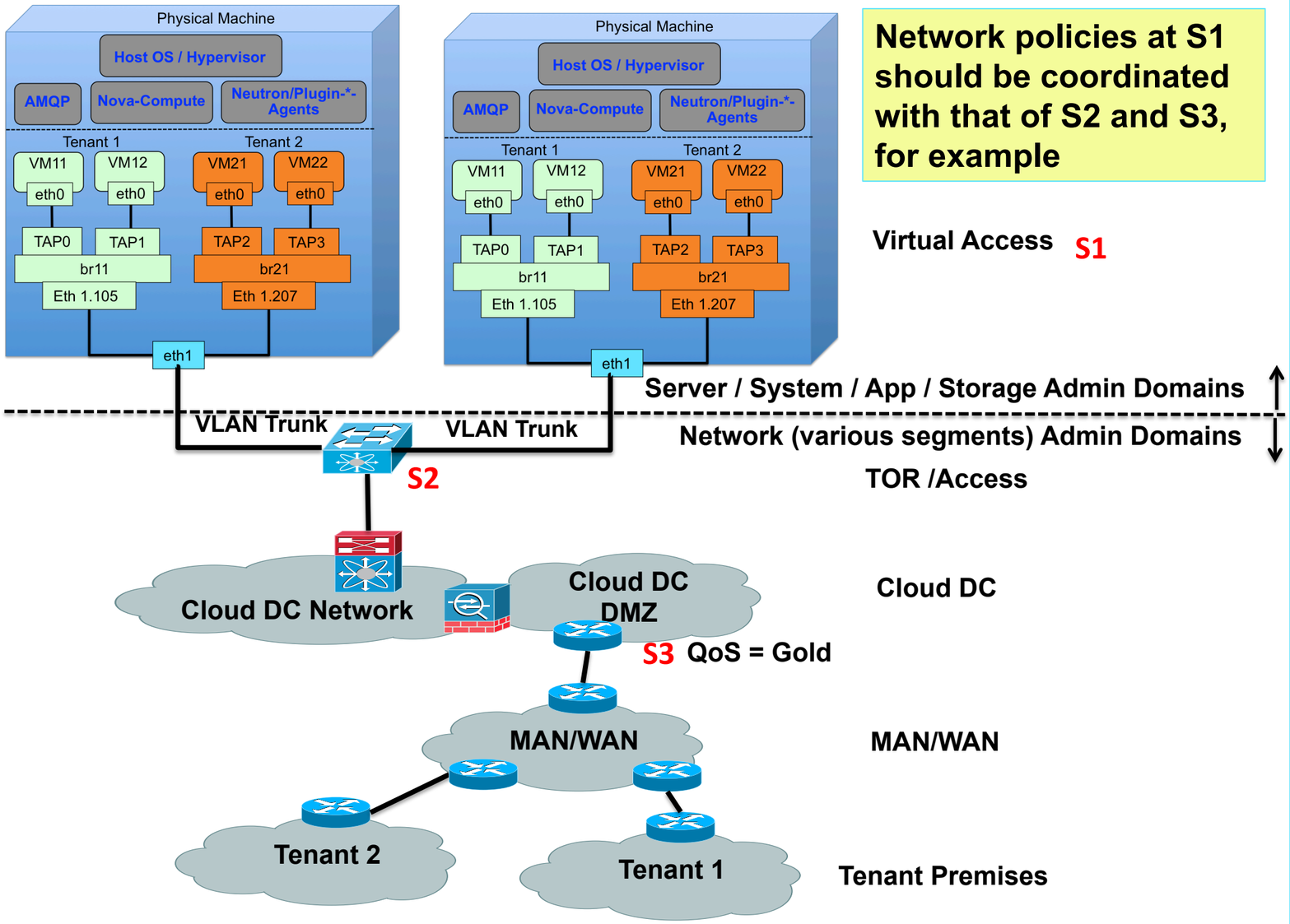
- With VEB, Access Switches moves from D2 to D1 with implications or challenges
  - Consistent Network Policy programming and application
  - Comprehensive and centralized visibility of network traffic
  - Virtual and physical network resource (VEB and ports) co-management
  - Huge number of virtual ports may have to be dealt with
  - Too many network management control points

Issues with VEB: Admin Domains



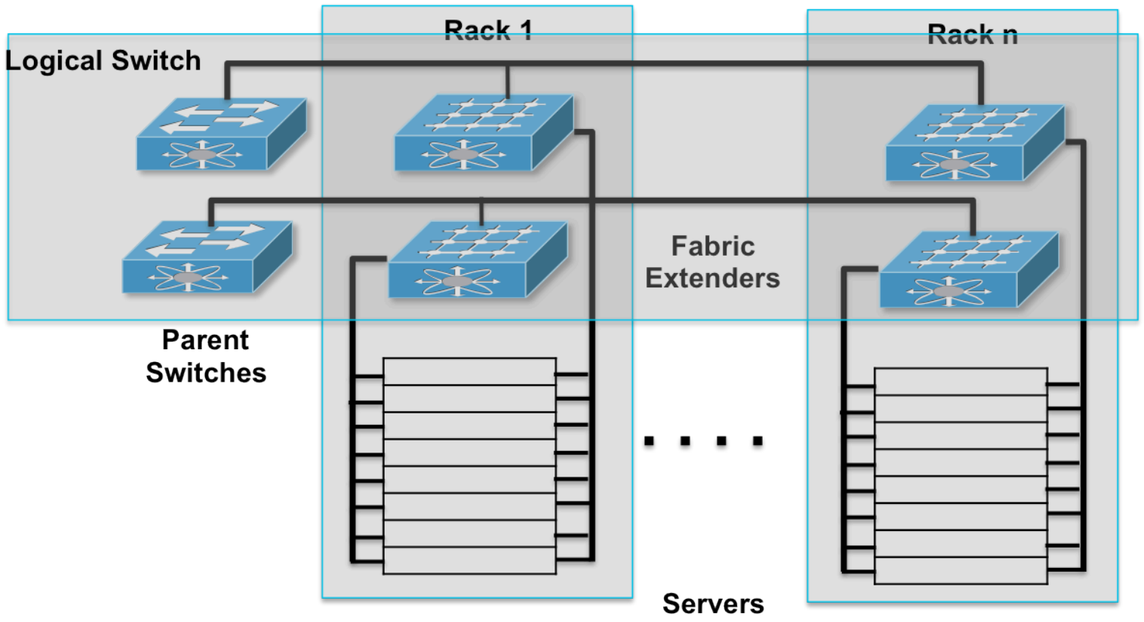
- Idealistic: Turn domains into one Single Domain!!
  - Division of expertise / labor → Admin domains
  - Infrastructure will not magically transform into one admin domain overnight or ever → Division of labor / expertise
  - Even in *fully automated (!!)* environment domain division useful

Admin Domains – Coordination



- Typically Net Admin (via NM tools) applies complex network configurations on behalf of server/system/app domains
- Now Server admin controls Network Policies at S1
- For consistency network policy programming / management should be coordinated

Fabric Extender (FEX) in Access Network

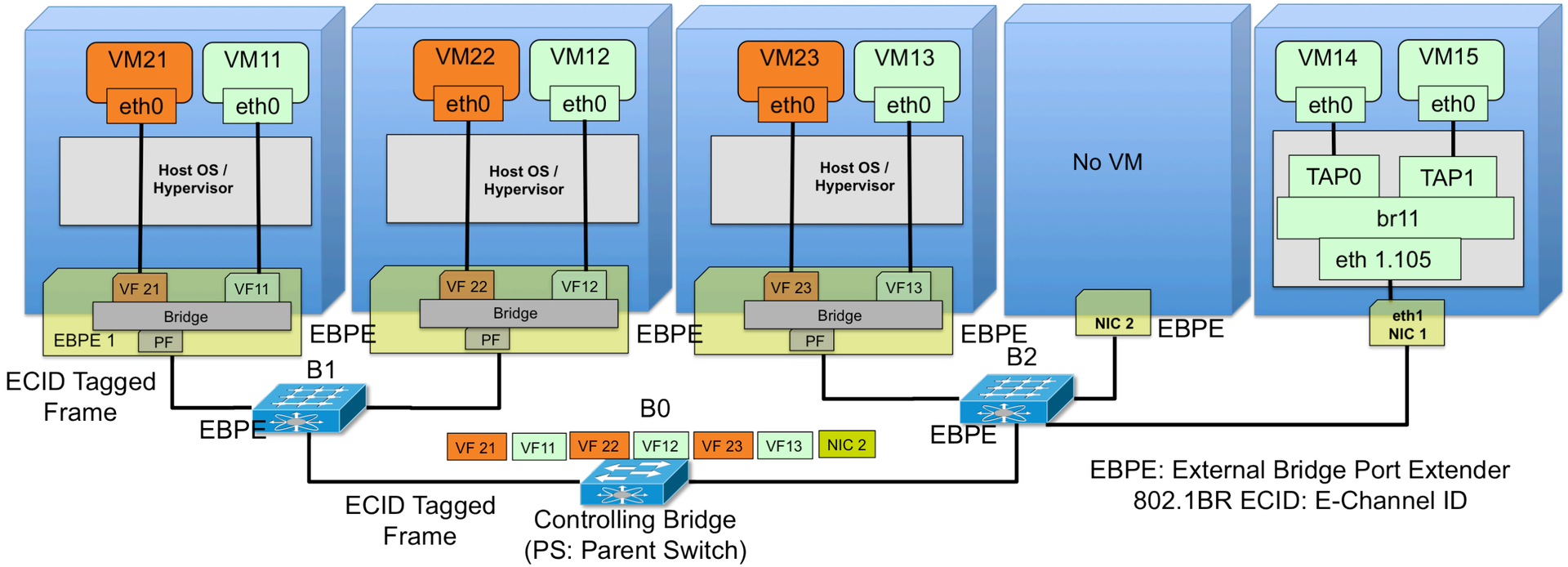


Parent Switches: such as Cisco Nexus 5K
Fabric Extenders: Such as Cisco Nexus 2K

- Fabric Extender (Chassis Virtualization) enables admin domain coordination
- A set of bridges / switches and ports as one big Logical bridge / switch
- Consider 8 FEX with 48 host / server ports → 384 ports
- 384 ports visible, managed, controlled from Parent switches (PS)
- Forwarding / Control plane on PS

- Network policies (templates) can be configured and applied on 384 ports centrally
- FEX ←→ PS links: 4-16 10Gbps, for example
- **IEEE 8021.BR** standardizes FEX, including extension to incorporate VM environment

IEEE 802.1BR

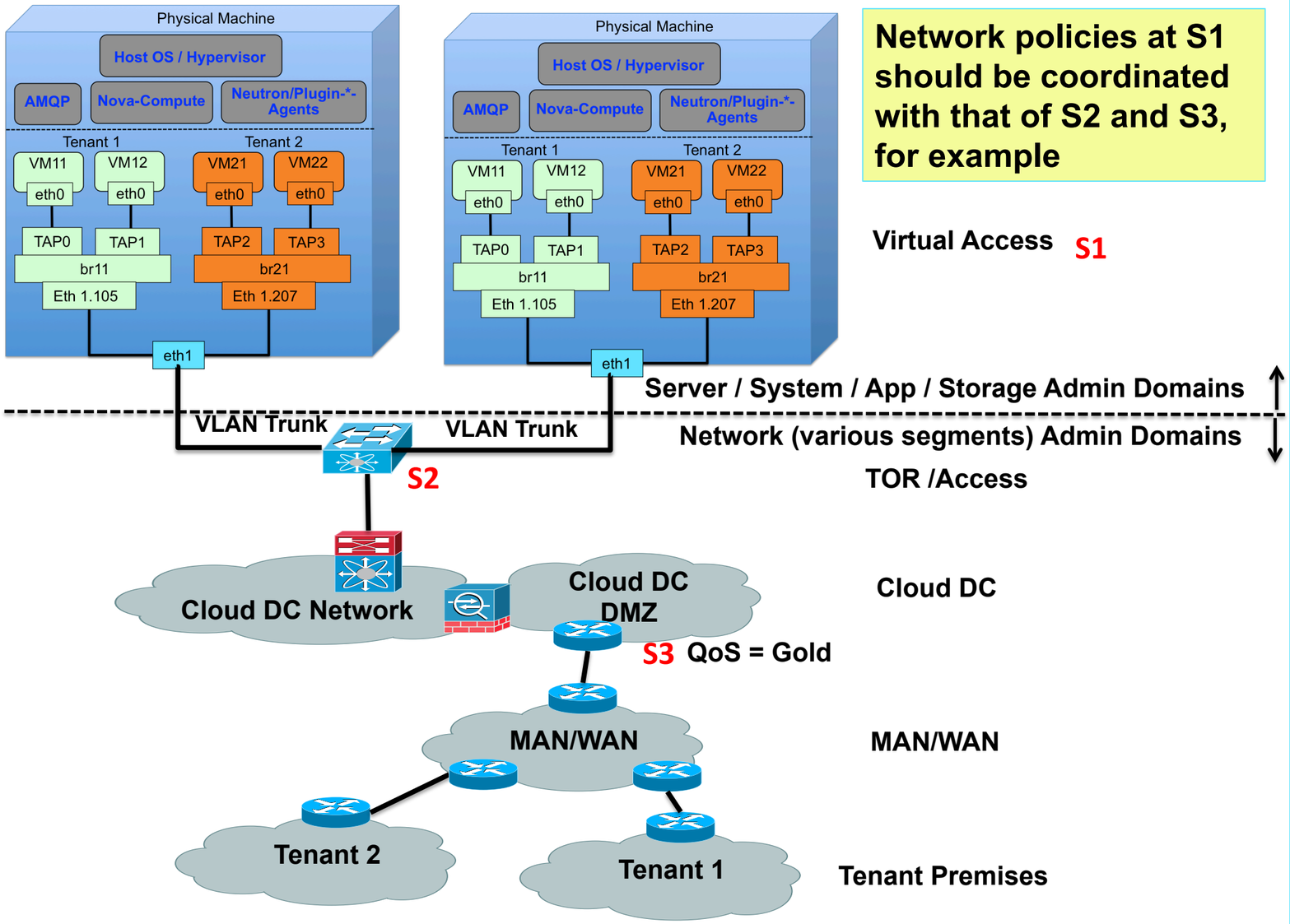


App Payload
TCP/UDP
IP
802.1BR E-Tag
Ethernet (802.1Q)

- [Note, various, sometimes confusing terms used, such as vNIC, VIF, Veth, Vethernet, etc.; we'll use our own]
- Virtual NICs (VNIC) appears as a virtual port (VPRT) on the Controlling Bridge
  - VNIC is called [VM attached] Extended Port in 802.1BR
    - VFs in SRIOV or VIF in Cisco VIC
  - VPRT is called C-VLAN (C: Customer) Component Port in 802.1BR

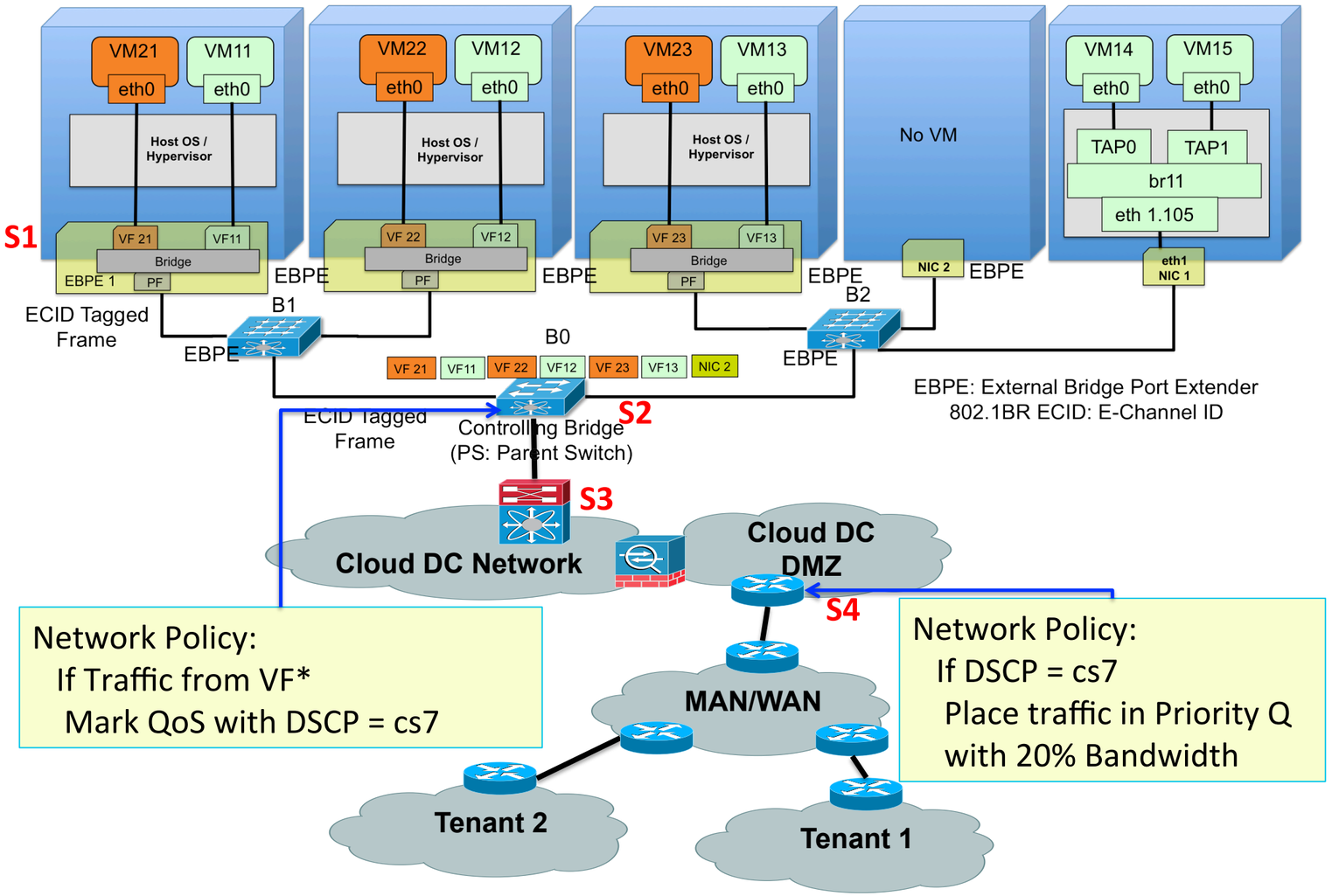
- When VF21 VNIC created and attached to VM by a Server Admin
- EBPE1 will send *extended port create message* to B0
- B0 creates VPRT and assigns ECID
- Any management and control action can be performed on VPRT on B0

Management / Control (M / C) Without 802.1BR



- S1 programmed / configured / provisioned for network policies by Server / VM Management System
- S2, S3 and other points by Network Management Systems
- With 802.1BR support, M / C points are reduced, specially at the DC access and aggregate with huge number of physical and virtual ports

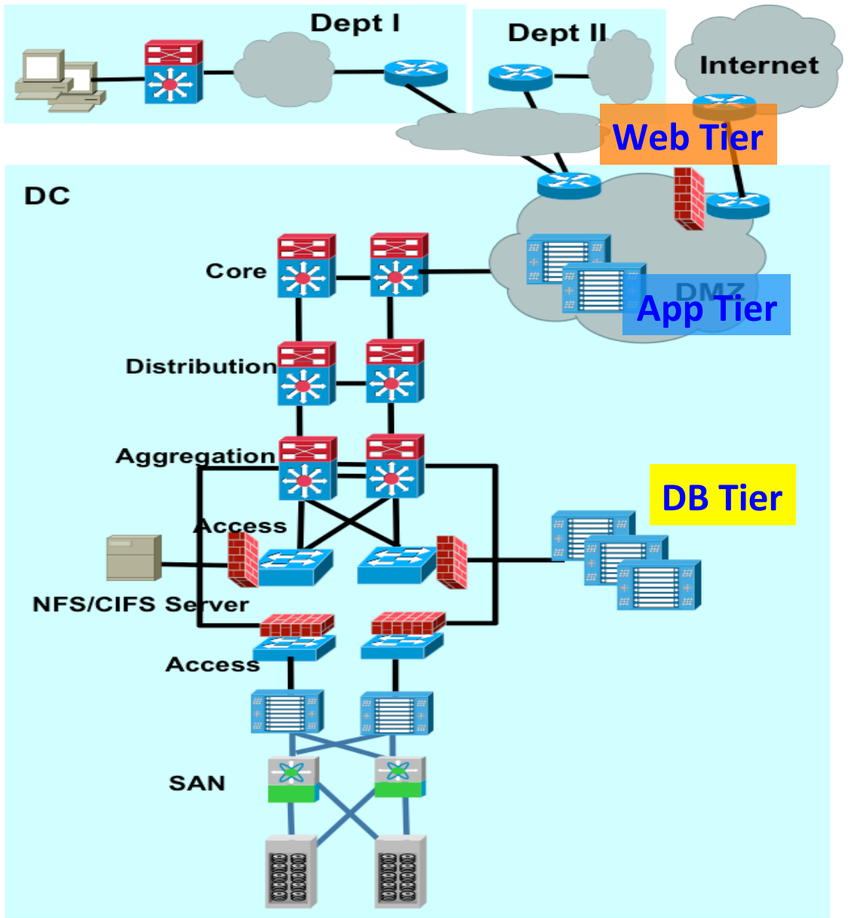
Admin Domains – Coordination – With IEEE 802.1BR



- A Network Admin defines policies (also known as *port-profiles*) on B0
- Port-profile IDs (PPID) sent to Server admin systems (through means outside the scope of 802.1BR)
  - For example, Cisco Nexus 5K supports this with VMware vCenter
  - With programmable interfaces, communication between admin domains becomes easier
- Server admin creates VNIC, associates a PPID and attaches VNIC to VM
- 802.1BR will then create VPORT in B0 with PPID automatically attached to VPORT

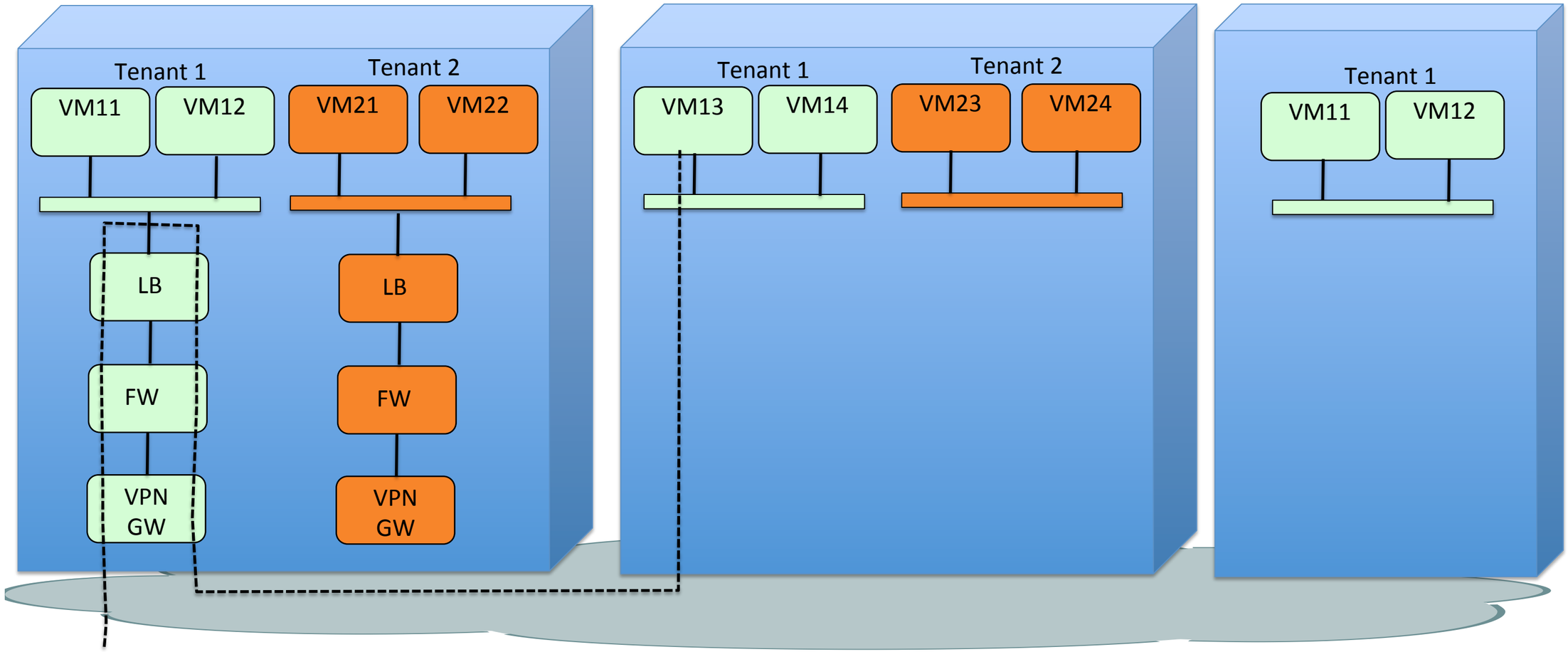
Virtualized DC Tiers

- Classical DC architecture



- Virtualize Classical DC architecture and move to Cloud
- Web, App, DB Tiers virtualized
- Network Function Virtualization
  - Firewalls, App/Web Load-balancers, VPN Gateways, L3 Functionalities

Network Function Virtualization (NFV)

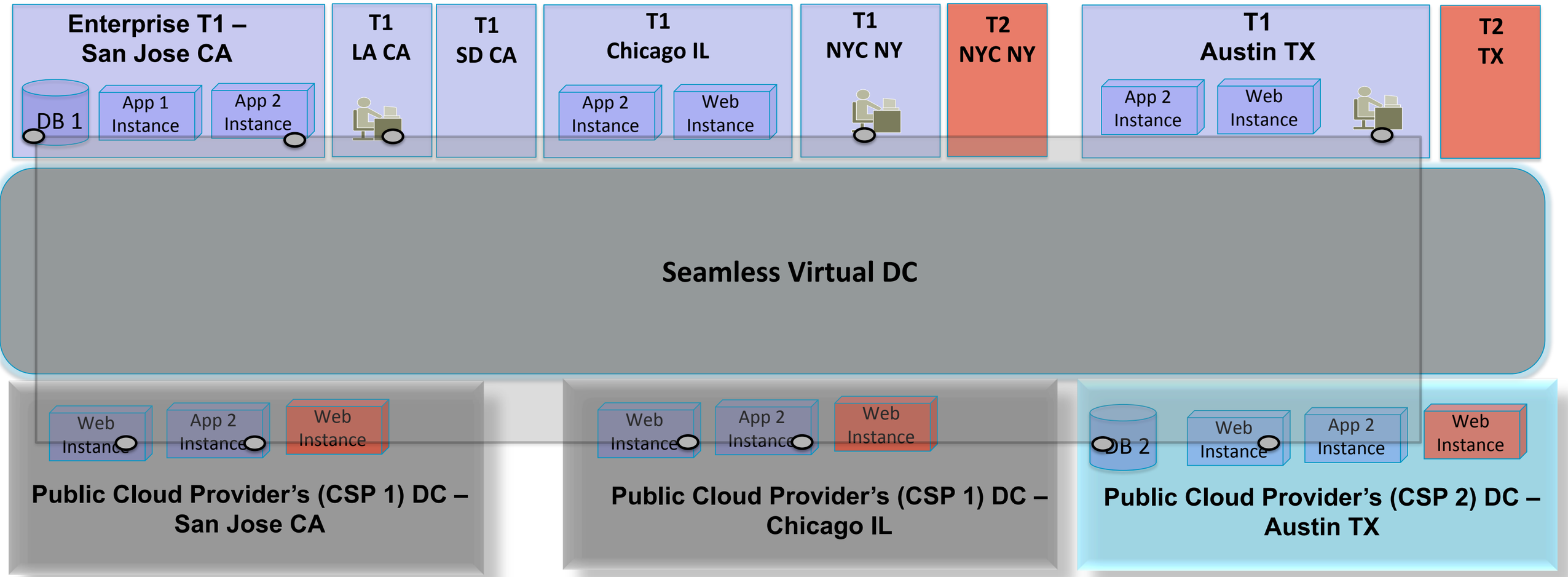


- NFV elements can be in VM (appliance) form factor and / or integrated embedded in Hypervisor and can be distributed

- Commercial NFV examples (Cisco):
  - CSR (Cloud Services Router) 1Kv – VPN GR, L3 Functionalities
  - ASA 1Kv – FW
- Openstack NFV capabilities

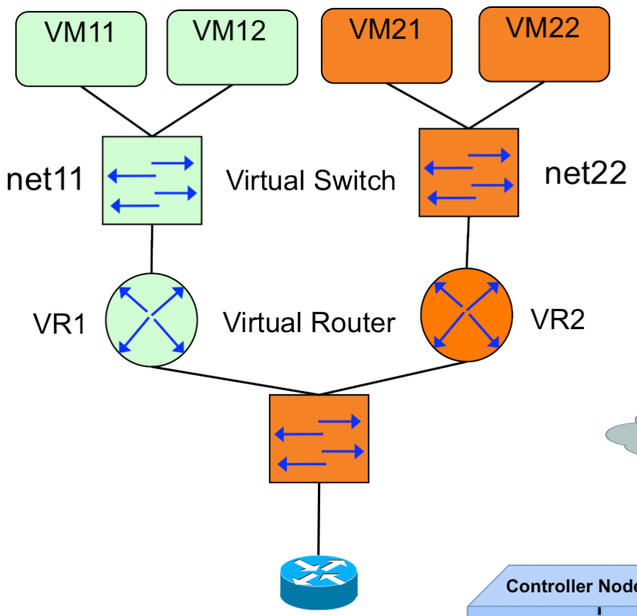
Take it to next level: Highly Distributed Cloud-based Virtual DC

- Virtualized Web / App/ DB Tiers and NFV distributed over multiple Cloud DCs / Sites and Multiple Clouds
- Since they are distributed Seamlessness should be provided → Seamless Virtual Cloud
  - Can have multiple instances of virtual DC / Seamless Cloud

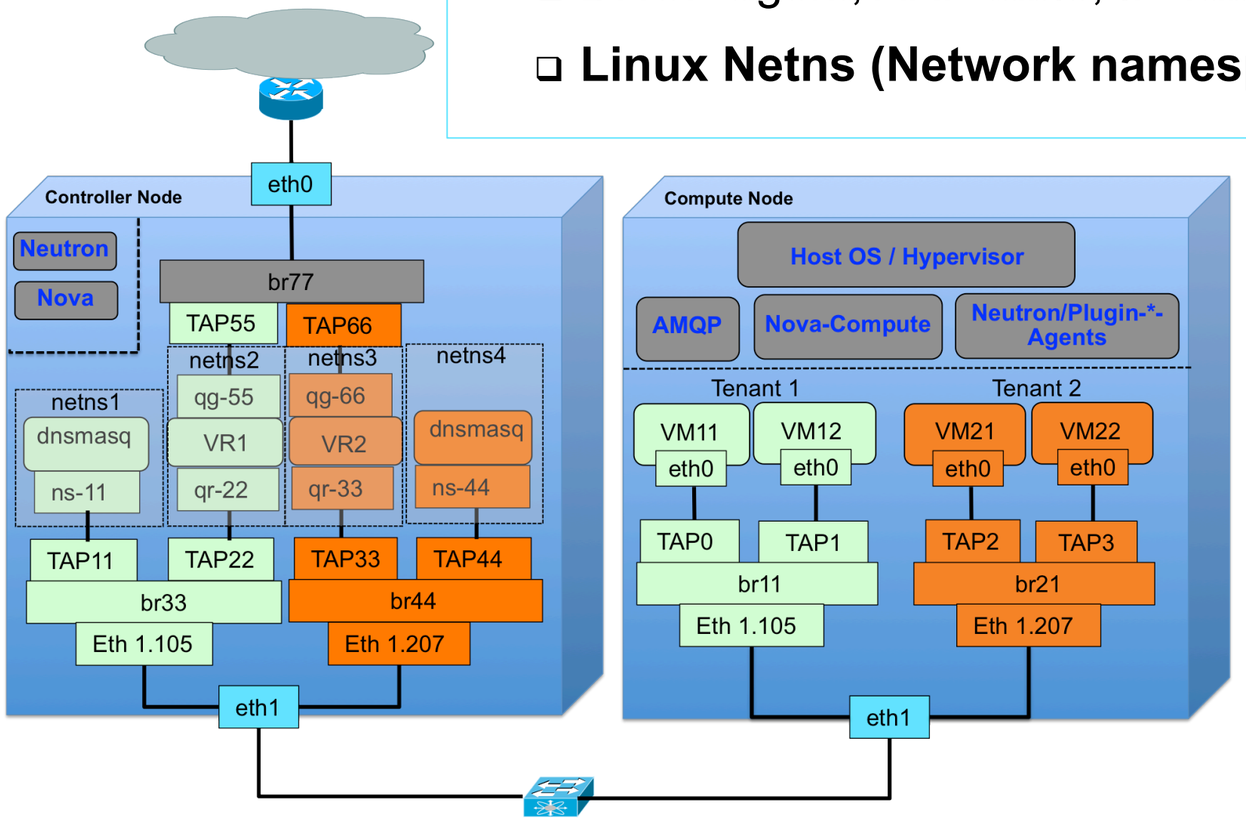


Cloud Networking:
End-to-end Multi-tenant Isolation

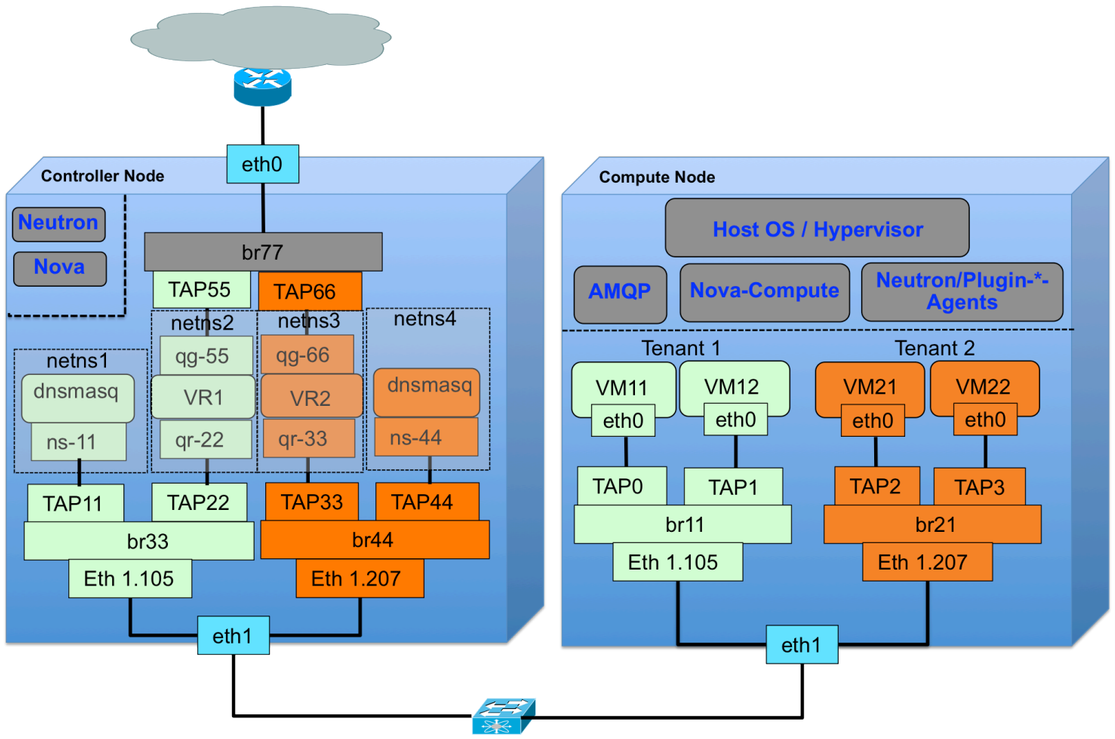
Isolation in Hypervisor



- VM provides Isolation
- Isolation should be E2E
- Various resources that used should be isolated within the Host / Hypervisor
  - DHCP agent, ARP table, IP Tables (for VR)
  - **Linux Netns (Network namespace)** can be used



Isolation in Hypervisor



- Physical interfaces and other resources reside in Default or Global NS, such as TAP11 and br33
- ip netns add netns1
- ip link add ns11 type veth peer name TAP11
- ip link set ns11 netns netns1
- ip netns exec netns1 ip link set dev ns11 up
- ip netns exec netns1 ifconfig ns11 10.10.11.1/24 up

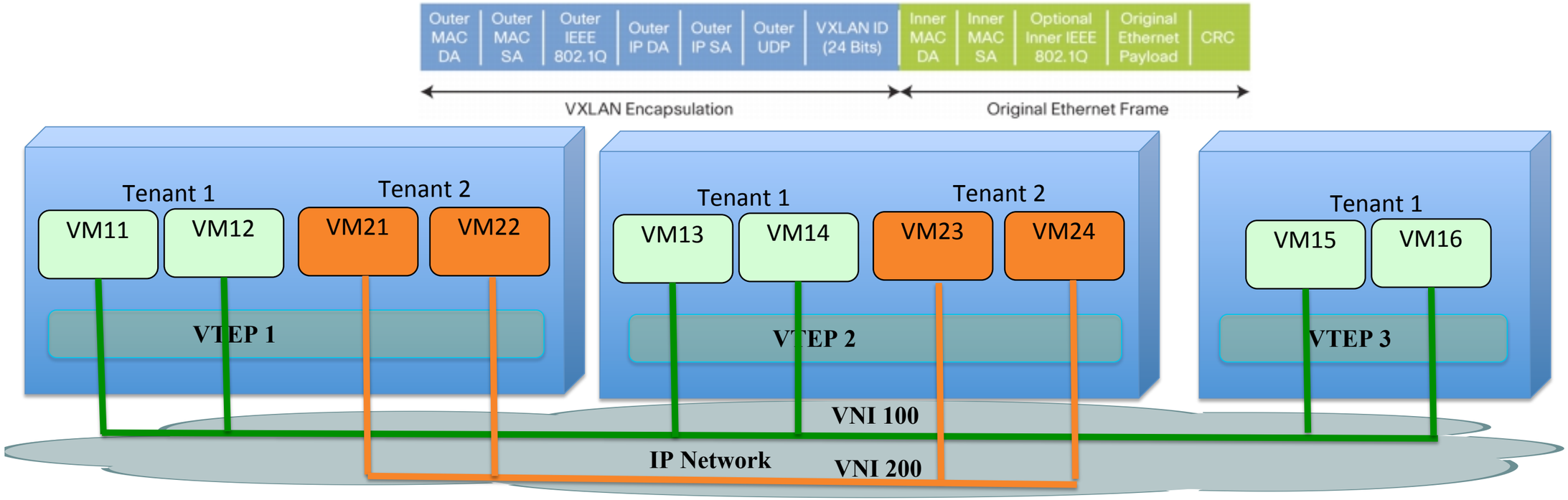
Isolation in DC Access Network

- 802.1Q VLAN → 12 bit ID → max 4K VLAN → does not scale in multitenant Public Cloud
- VXLAN (Virtual Extensible LAN) is one solution to address scalability for multitenant isolation in DC access network

VXLAN

- VXLAN: 24 bit ID → 16 million VXLAN segments
- MAC in UDP over IP multicast network – L2 Overlay network over L3 IP Network
  - Thus VXLAN segments can span over L3 network

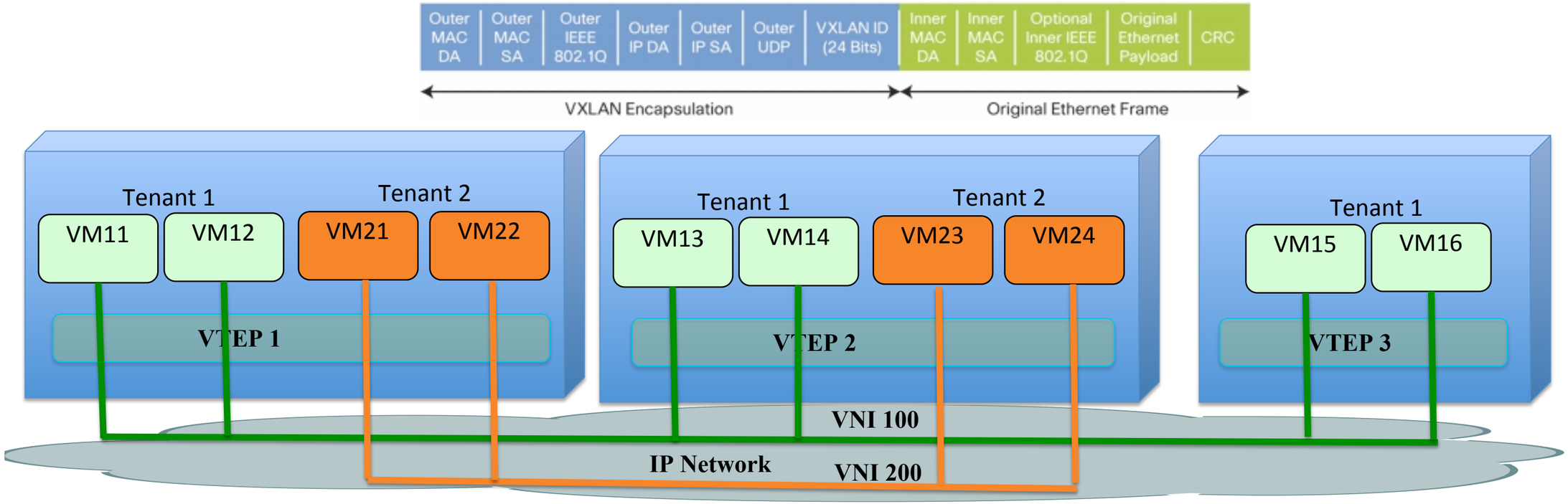
- Each Server joins a IP multicast group (IPMC) for BUM (broadcast, unknown, multicast)
- Because of UDP based can support ECMP (equal cost multiple paths) for bandwidth load balancing
  - UDP pseudorandom source port based hashing (by switches supporting 5-tuple based hashing)



VXLAN VTEP

- A VXLAN Tunnel End Point (VTEP) in hypervisor / virtual access switch
  - VTEP is assigned its own IP address → Outer MAC / IP identifies VTEP
- VTEP uses IP multicast to populate Forwarding Table

- When VMs communicate, VTEP performs following
  - VXLAN UDP encapsulation for VM-VM direct communication
  - IP multicast packet for ARP requests
  - Populates Forwarding Table in VTEP as needed
    - Source VM MAC, source VTEP IP/MAC, VNI association



VXLAN VTEP – Multicast Group

- VTEP uses IP multicast to populate Forwarding Table
- Multicast protocols should be supported in switches / routers: IGMP, PIM
- Each node belonging to a specific VXLAN segment (VS) joins a multicast group (MG)
- 1:1 correspondence to a VS to MG is not possible
  - Multicast address range 224.0.0.0 to 239.255.255.255
- One or more Transport VLAN used to carry VXLAN traffic!
  - If one VLAN, PIM not needed
- IGMP Snooping on L2 switches and Querier on L3 routers should be enabled
  - L2 switches snoops multicast traffic to find out which port subscribed to which MG

VXLAN VTEP- Forwarding Table Learning

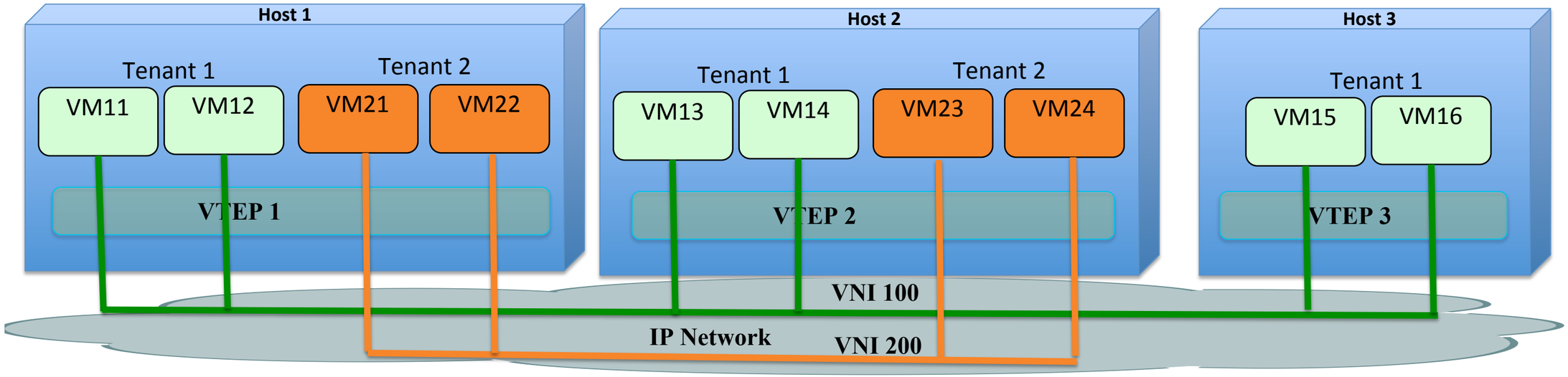
Dest: Multicast MAC Address	VTEP1 MAC	…	MG Address (like 239.1.1.10)	VETP1 IP Address	Including, UDP destination port 4789 …	VNI # …	FFF…FF (ARP broadcast)	VM11 MAC	….

Outer Encapsulation provided by VTEP | **Original Arp request sent by Host 1**

Host 2 and 3 learns

VM11 MAC	VTEP1 IP	VNI #	…
….			

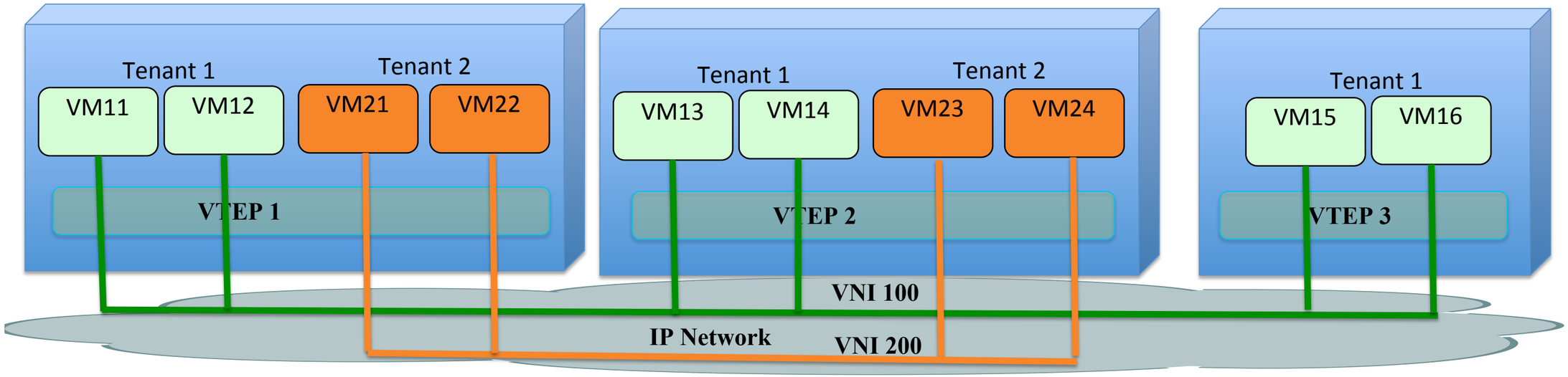




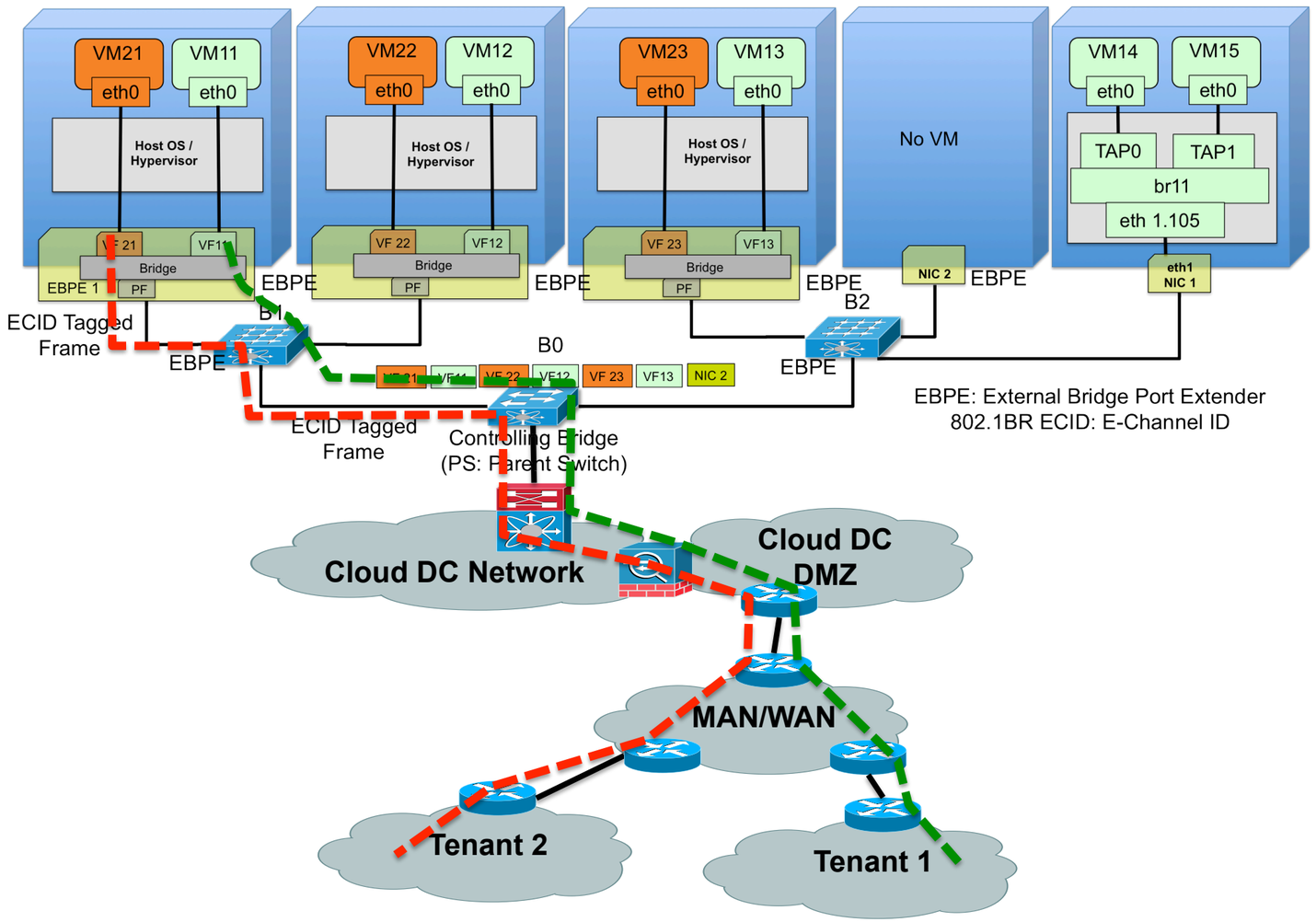
VXLAN Multitenant Isolation

- Each VNI or VXLAN segment provides multitenant isolation
  - From Host / Hypervisor / virtual access switch (VTEP) to host / hypervisor / virtual access switch (VTEP)
  - VTEP can be supported on any switch / router, not just hypervisor / virtual access switch





E2E Isolation in Network Segments



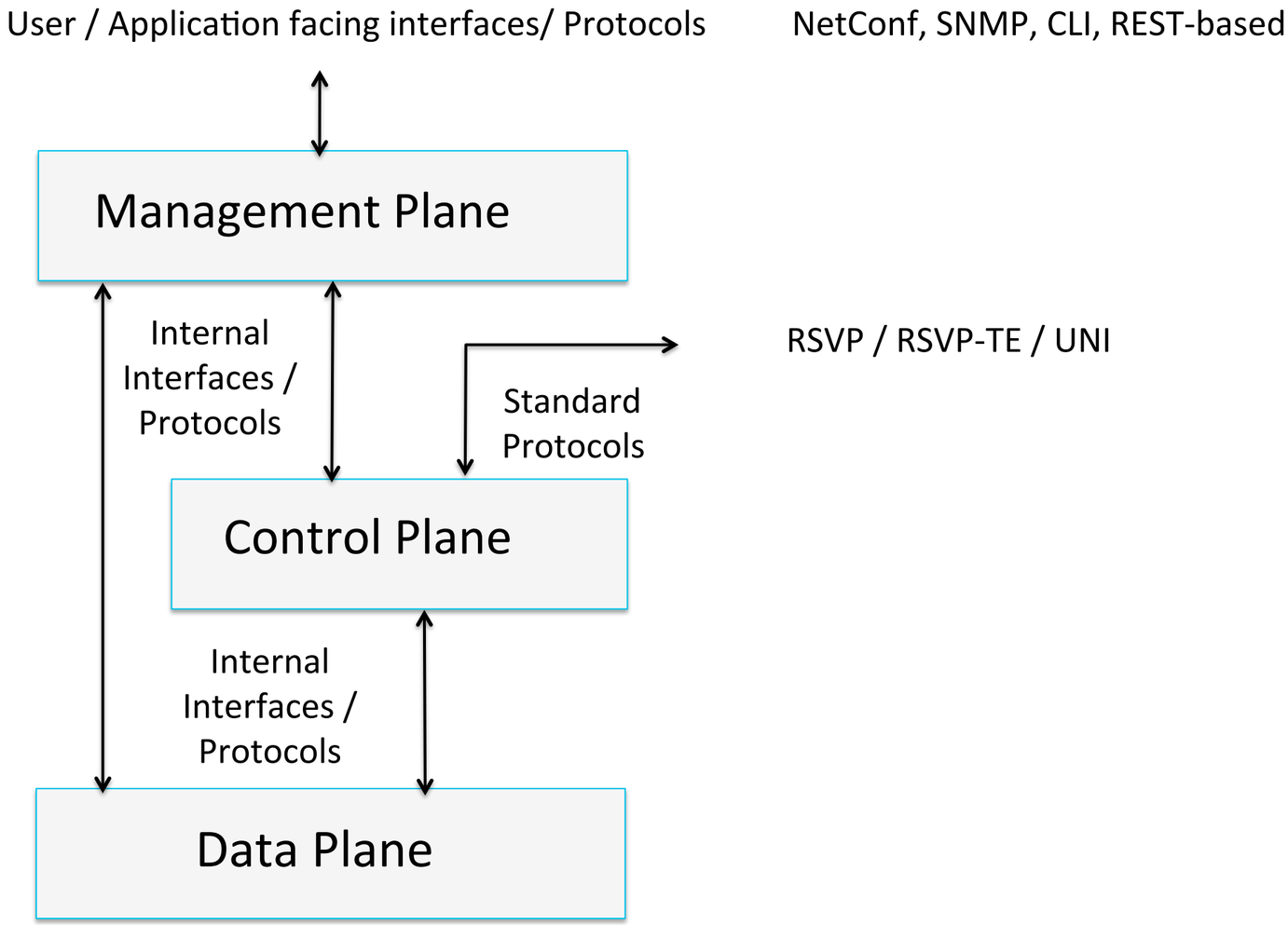
Isolation in Access → **VM / 802.1BR / VLAN/ VXLAN / GRE**

Isolation in Cloud DC up to DC Edge → **VLAN VRF-Lite Potentially VXLAN**

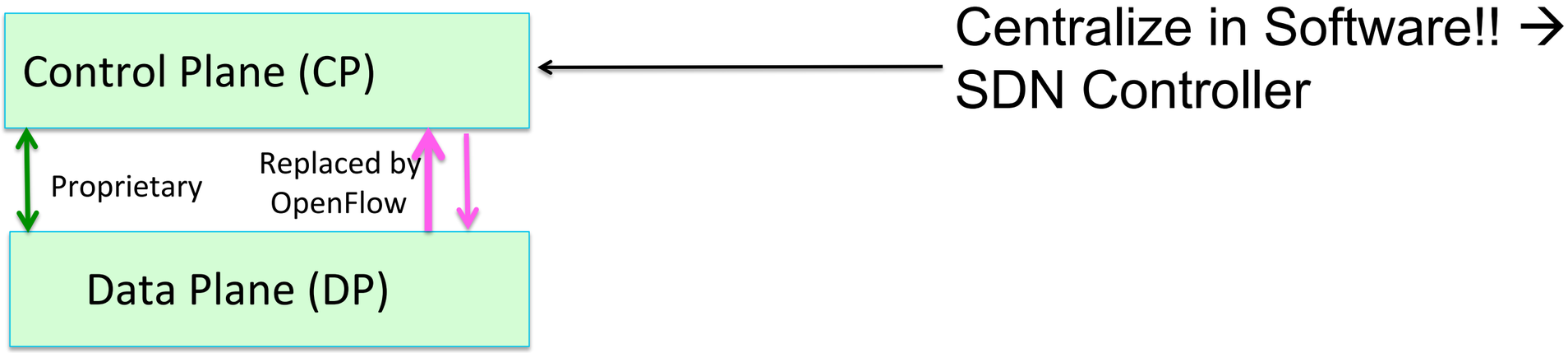
Isolation in MAN/ WAN up to Tenant Edge → **IPSEC Site-to-site MPLS VPN Open VPN**

Cloud Networking:
SDN / Programmable Networking

Existing Frameworks



SDN and OpenFlow – ONF Definition



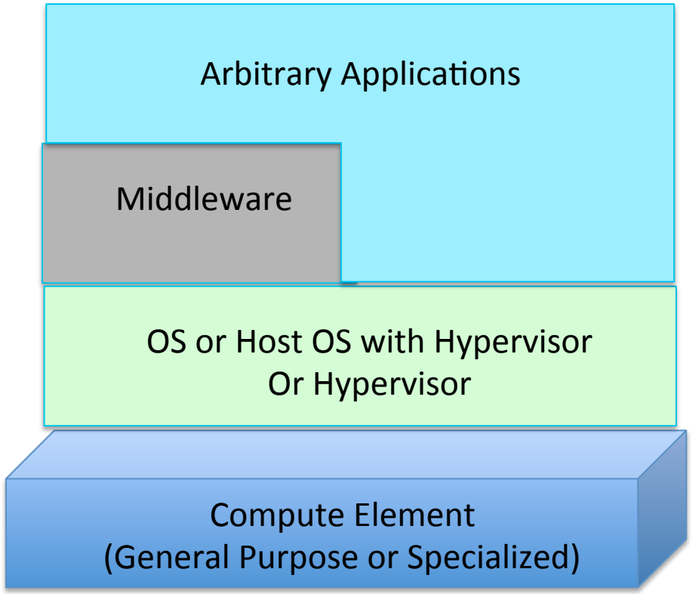
OpenFlow: "…open standard that enables researchers to run experimental protocols in campus networks. Provides standard hook for researchers to run experiments, without exposing internal working of vendor devices……"

http://www.openflow.org/wp/learnmore/

- Pretty much centered on dealing with Forwarding ("programming" forwarding path)
- There is more to networking than just forwarding

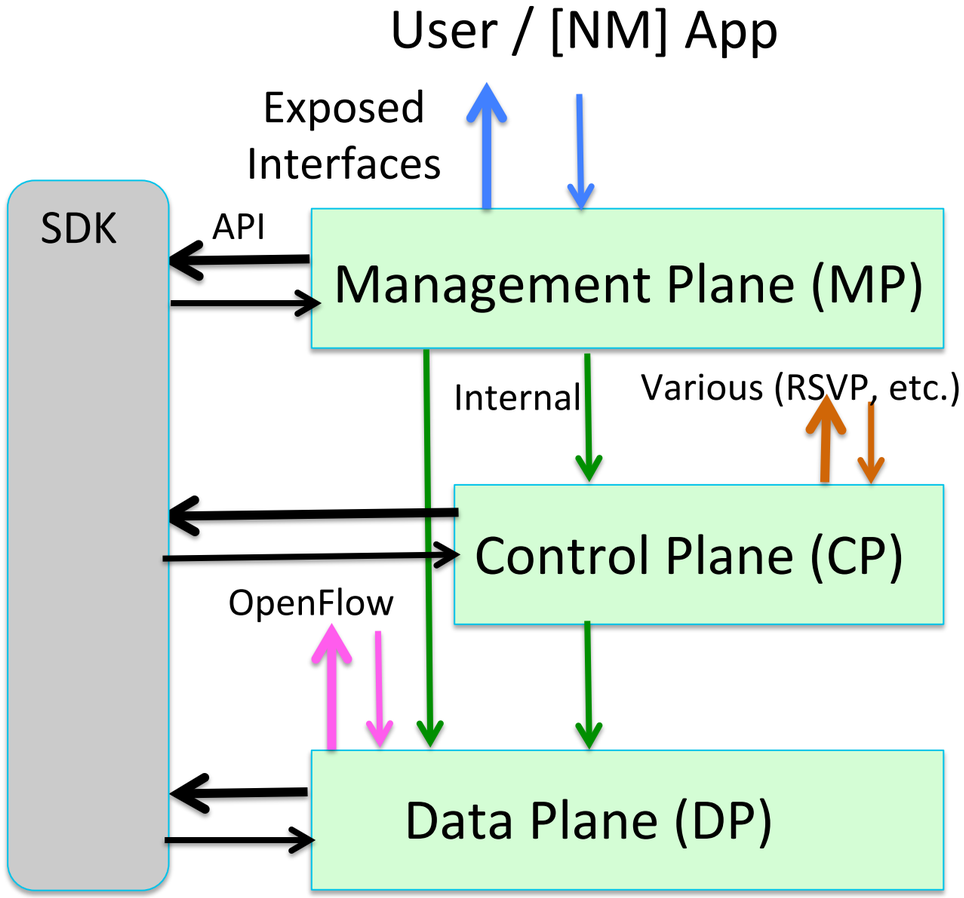
Programmable Networking

- SDN/OpenFlow: Major Focus: Flow-directed Forwarding path control
- ***Programmable Networking*** is the way to go
- Make existing capabilities programmable with new innovations
- Leverage best of the both worlds



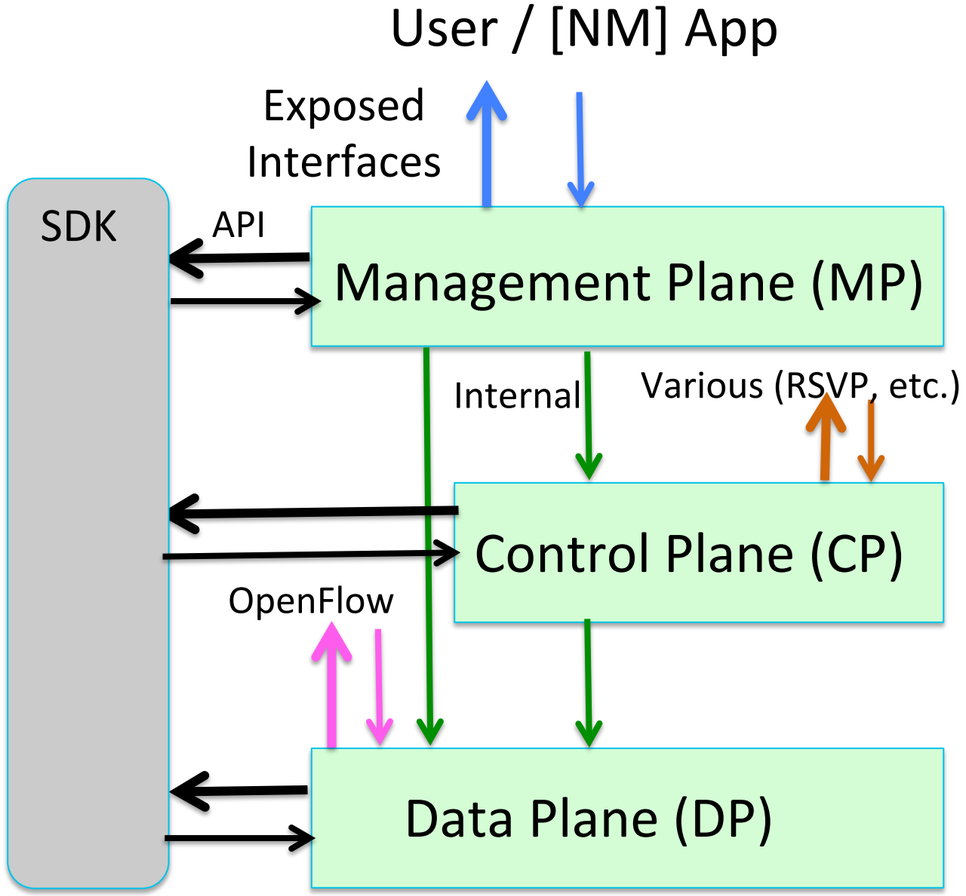
Programmable Networking with Programming Framework

- Consider Network as a computer or cluster and distributed DB
  - Program it
- Load and execute arbitrary programs on network: Examples:
  - Algorithms for new Longest Prefix Match
  - Heuristic algorithms for constraint based routing
  - Firewall ALG (Application Level GW) for a new application / proprietary protocol
  - Logical (overlay) topology for a tenant's Seamless Virtual Cloud



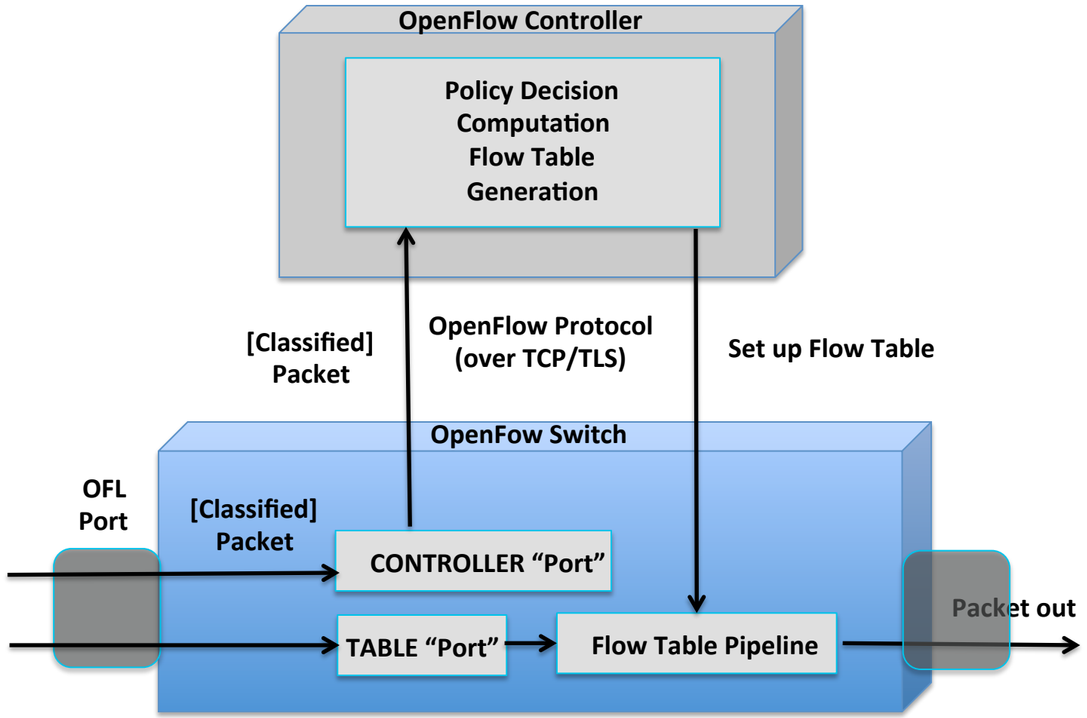
Programmable Networking with Programming Framework

- A full programming framework with published API and SDK support
- Developed programs can be run
  - Within (a user space) of these planes or network element (NE)
  - Outside of planes or NE in / as a Controller or Applications



SDN / Programmable Networking: OpenFlow

OpenFlow Switching



OpenFlow Switch Flow Switching Model (v1.1.0)

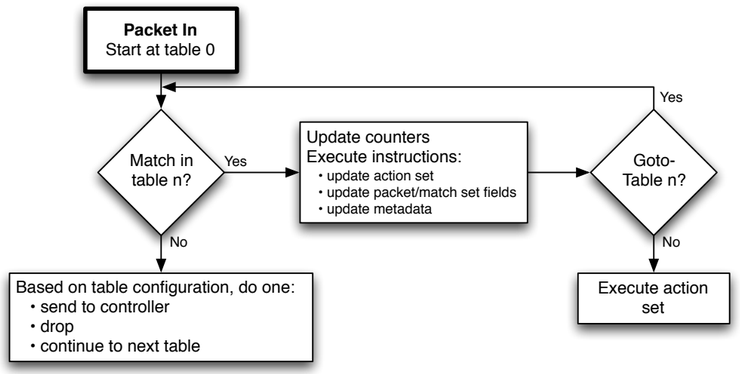
- Flow table entries (CRUD by Controllers)
- Multiple tables can be chained together for pipeline processing

Match Fields	Counters	Instructions

On

Count of Packets, Bytes, drop, error, etc.

Ingress Port	Metadata	Ether src	Ether dst	Ether type	VLAN id	VLAN priority	MPLS label	MPLS traffic class	IPv4 src	IPv4 dst	IPv4 proto / ARP opcode	IPv4 ToS bits	TCP/ UDP / SCTP src port / ICMP Type	TCP/ UDP / SCTP dst port / ICMP Code



From OpenFlow Sec 1.1.0 Implemented

Apply-Actions <action(s)>: Immediate actions
Ex. Output to port, Push VLAN, Set QoS, etc.

Clear-Actions

Write-Actions <action(s)>: specified actions are merged with current action set

Write-metadata <metadata/mask>

Goto-Table <next-table-id>

OpenFlow Hybrid Switch

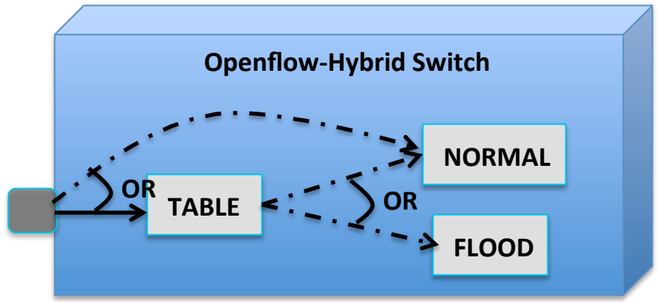
- OpenFlow-only switches support only OpenFlow operation
  - All packets are processed by the OpenFlow pipeline, and can not be processed otherwise

- OpenFlow-hybrid switches support both OpenFlow operations and typical (Ethernet) switching functions
  - Packet classification is non-OpenFlow based
  - Once classified, following options for processing

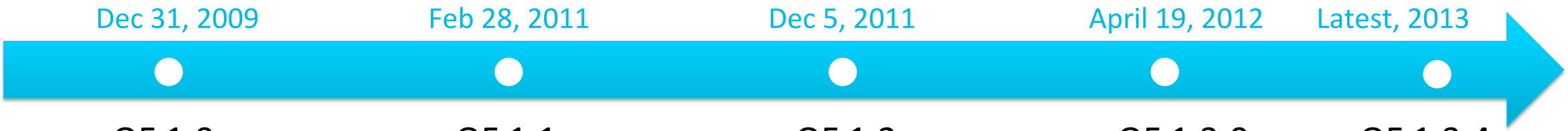
    OpenFlow Pipeline (sent to TABLE conceptual port)

    Non-OpenFlow pipeline (NORMAL)

    TABLE to NORMAL or FLOOD (for flooding)



OpenFlow Versions



OF 1.0

- Single Table
- L2, IPv4 focused matching

OF 1.1

- Multiple Tables
- MPLS, VLAN matching
- Groups: {Any-,Multi-}cast
- ECMP

OF 1.2

- IPv6
- Flexible-length matching

OF 1.3.0

- 802.1ah PBB (MAC-in-MAC)
- Multiple parallel channels between Switch and Controller

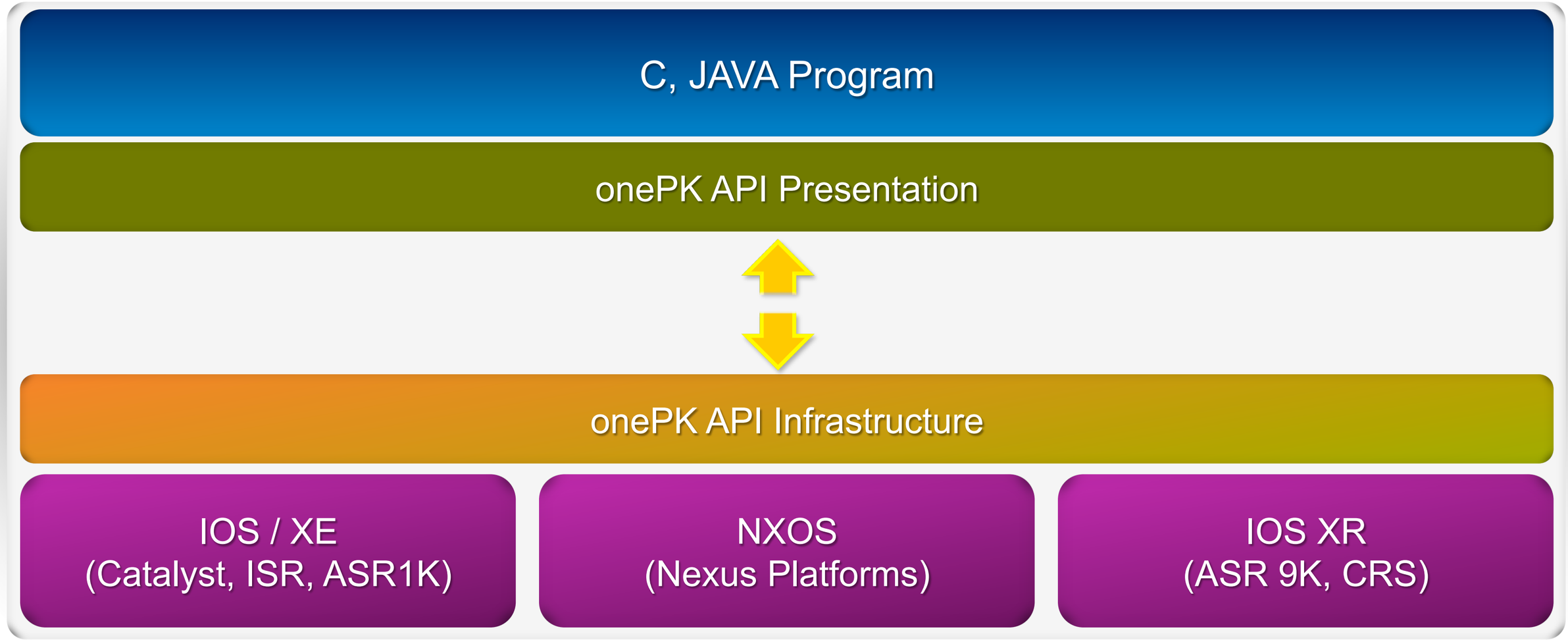
OF 1.3.4

SDN / Programmable Networking: Cisco OnePK

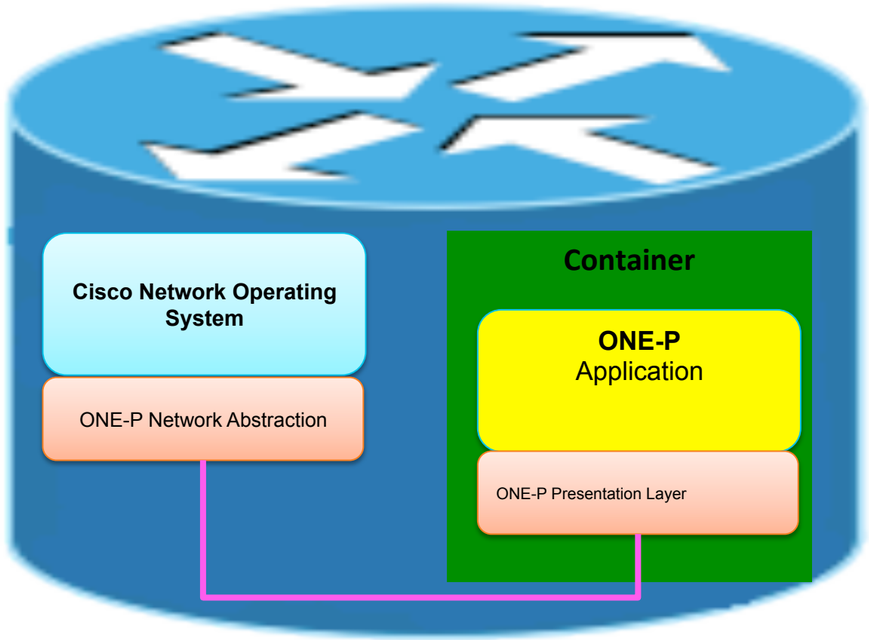
Introduction

- Cisco OnePK is a framework for programming the network
  - A network that is widely deployed with rich set of advanced features
  - It provides an abstraction and API to program against those rich features
  - There are multiple deployment models for OnePK programs

onePK Architecture

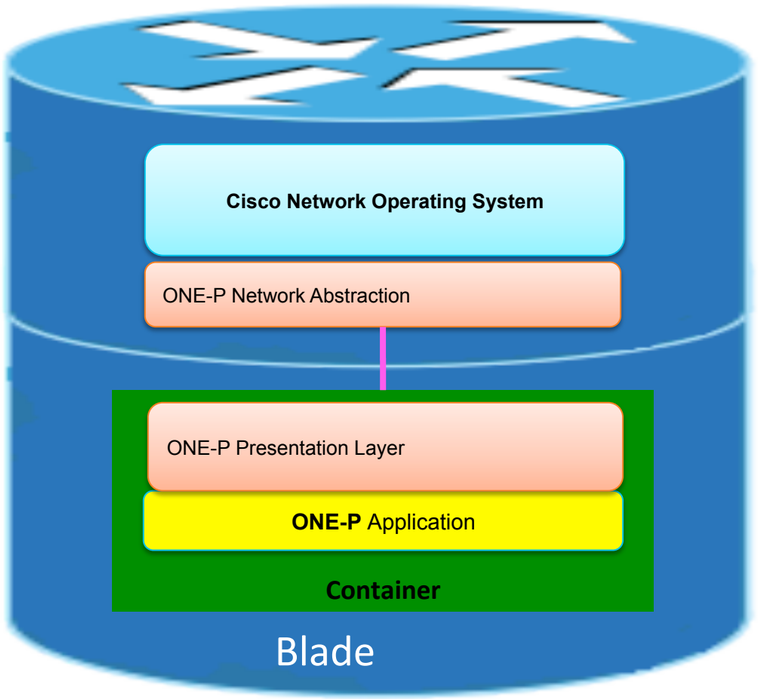


OnePK Application Deployment Model – Process Hosting Mode



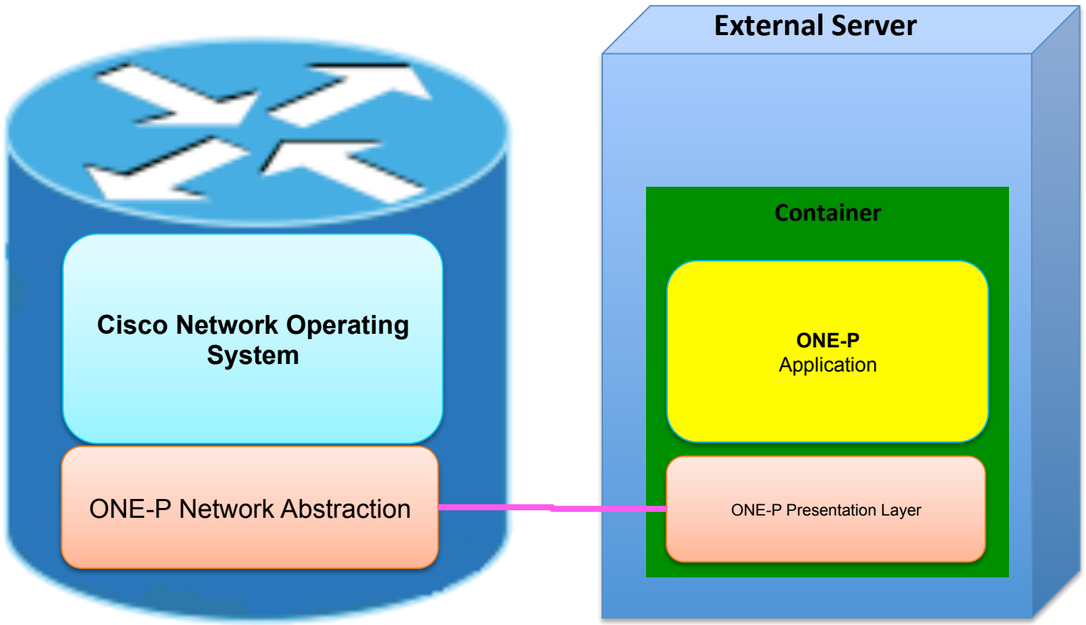
- Applications run within network element
- Applications are hosted within a container
- Low latency access to forwarding path
- Facilitate advanced application specific local processing / decision making without traversing network

OnePK Application Deployment Model – Blade Hosting Mode



- Applications run within a blade (such as a Linux blade) on network element
- Applications are hosted within a container
- Medium latency access to forwarding path
- Better isolation and more compute resources

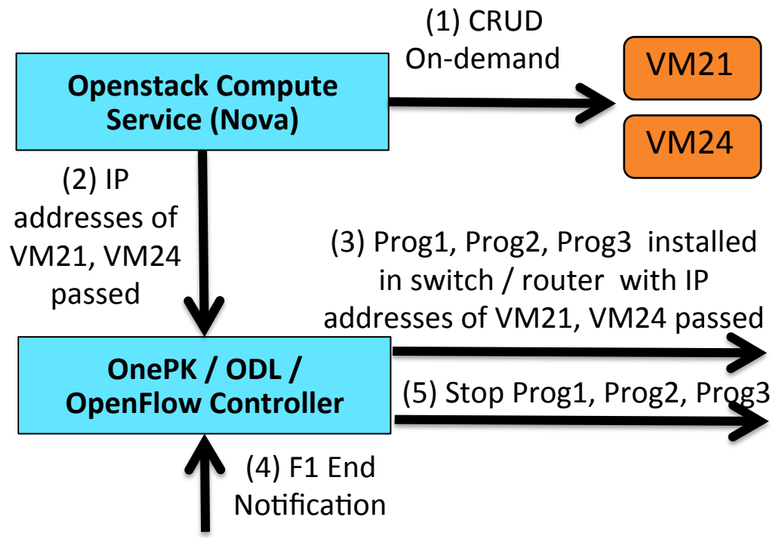
OnePK Application Deployment Model – End Node Hosting Mode

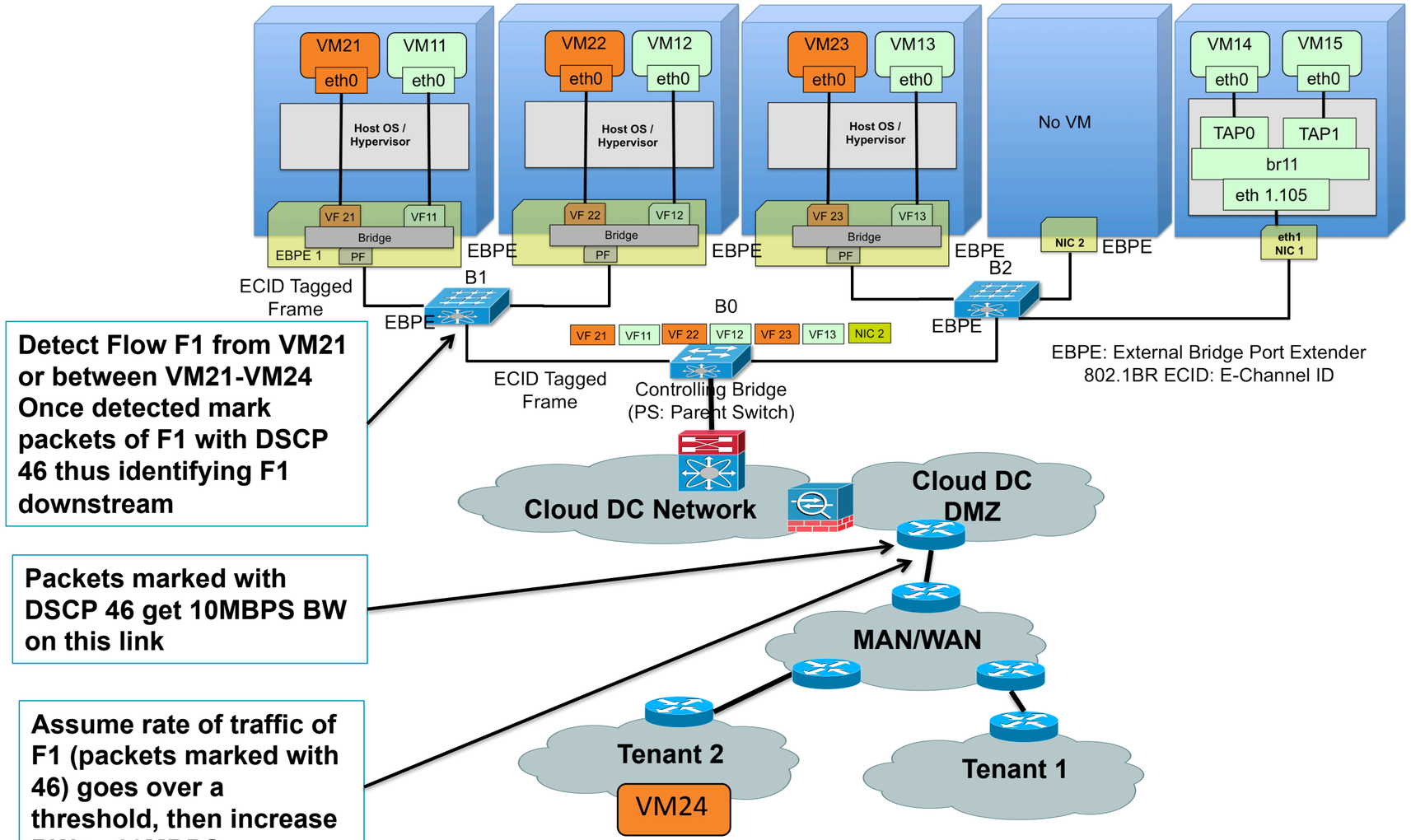


- Applications run on commodity devices (servers, laptops, mobile devices)
- Higher latency access to forwarding path
- High isolation and more compute resources

Cloud + SDN Use Case: Flow Directed Dynamic QoS

- Static configuration is not that useful in a Cloud environment
- Resources are CRUD on-demand
- Flow detection and QoS marking should be programmable → Prog1
- Similarly BW increase / decrease → Prog2
- VM21-VM24 info (such as IP address) will be passed to Prog1 dynamically by a Cloud Controller
- Detect when F1 ends → Prog3
- When the flow F1 ends, Prog1, Prog2, Prog3 can be terminated
- Prog2 can be get input from a complex program (traffic engg program)





Code Sample: App to NE session

- Create an App instance and connect to an NE to obtain session handle (for future communication)

```
/* Create App instance */
NetworkApplication networkApplication = NetworkApplication.getInstance();
networkApplication.setName(applicationName);
```

```
/* Get an NE instance by IP address of the NE */
InetAddress ipAddress = InetAddress.getByName(elementAddr);
networkElement = networkApplication.getNetworkElement(ipAddress);
```

```
/* Connect to the NE to establish session */
SessionConfig sessionConf = new SessionConfig(SessionTransportMode.SOCKET);
sessionHandle = networkElement.connect(username, password, sessionConf);
```

- Draft sample code – not 100% accurate

QoS Policy Programming Model

- **Filter**: Packet content is inspected based on Filter specification → also known as **Classification**

- **Action**: That are performed on packets or Queues (such as DSCP marking or bandwidth shaping) when Filter condition(s) satisfies

- **Target**: Policies are applied on a target, such as an NE interface

Code Sample: QoS Policy

```
import com.cisco.onep.policy.*;

NQP = new QosPolicy(elemA); private static QosPolicy NQP = null; private static NetworkElement NEI = null;
```

```
/* Create Filter Container – API term ClassMap; multiple ClassMap can be added to a QoS Policy */

class1 = new ClassMap (NEI, ClassMap.ClassOper.CLASS_OPER_OR);

NQP.addClass (class1, 10);
```

```
/* Create Filter and add it to a ClassMap: Filter by VM_IP passed by Compute Controller*/

FL1= new FilterProtocol (VM_IP);

class1.addFilter (FL1);
```

```
/* Mark packets with DSCP 46 for priority queuing in upstream interfaces and 10 MBPS BW guarantee on Target interface*/

A1 = new ActionMark (ActionMark.MarkType.ONEP_MARK_IP_DSCP, 46);

NQP.addAction (class1, A1);

A2 = new ActionClassQueue (10, OnepBandwidthUnits.ONEP_BW_UNITS_MBPS);

NQP.addAction (class1, A2);
```

```
/* Policy NQP is applied on output of interface e0/1 of NE */

Target target = new Target (NEI.getInterfaceByName("e0/1"),
Target.Location.ONEP_TARGET_LOCATION_HARDWARE_DEFINED_OUTPUT);

NQP.applyToTarget (target);
```

Draft sample code – not 100% accurate

SDN / programmable networking: Open Daylight

What is Project OpenDaylight?

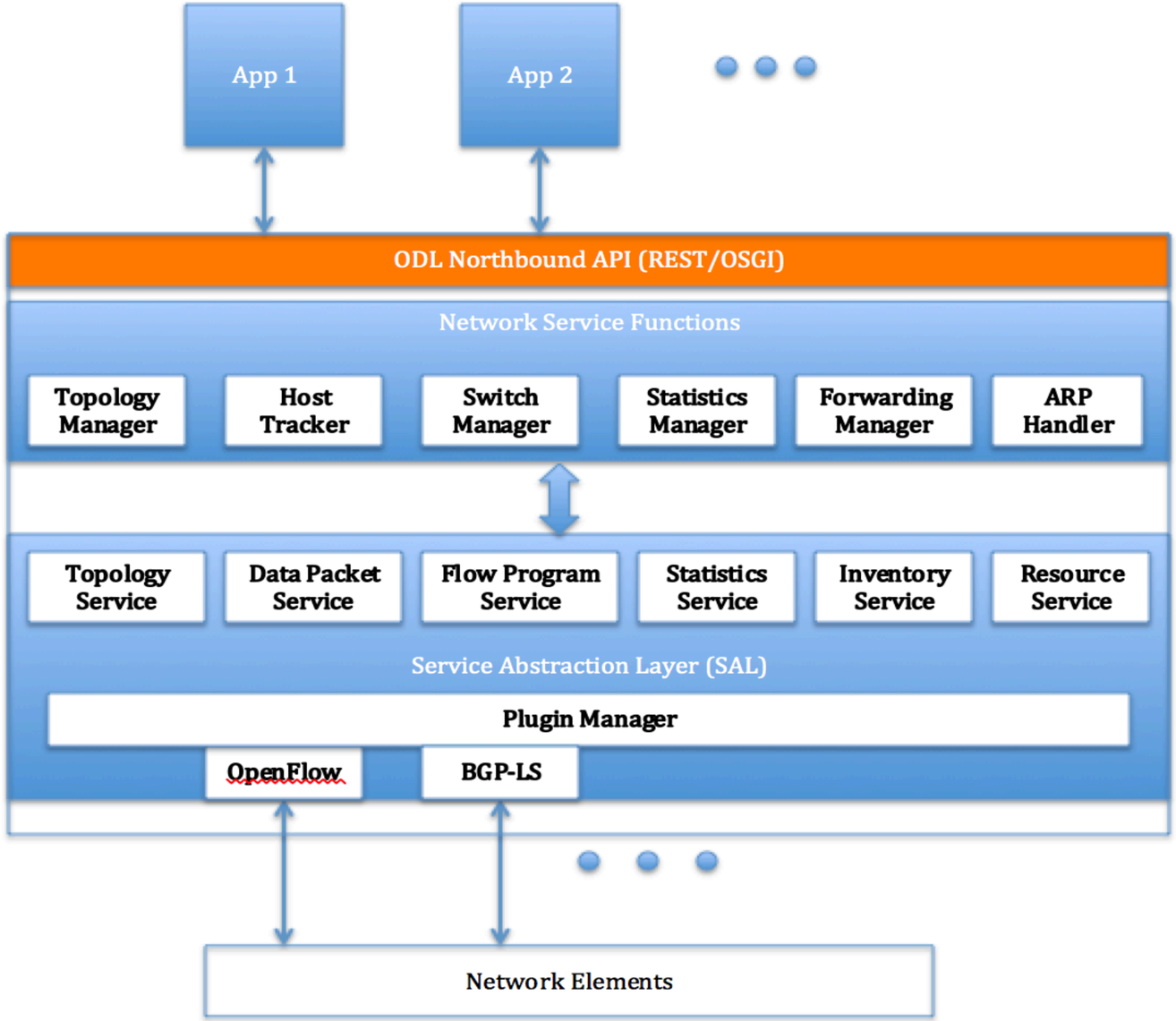
Open source project under the Linux Foundation

Furthering the adoption and innovation of Software Defined Networking (SDN) through the creation of a common industry supported framework.



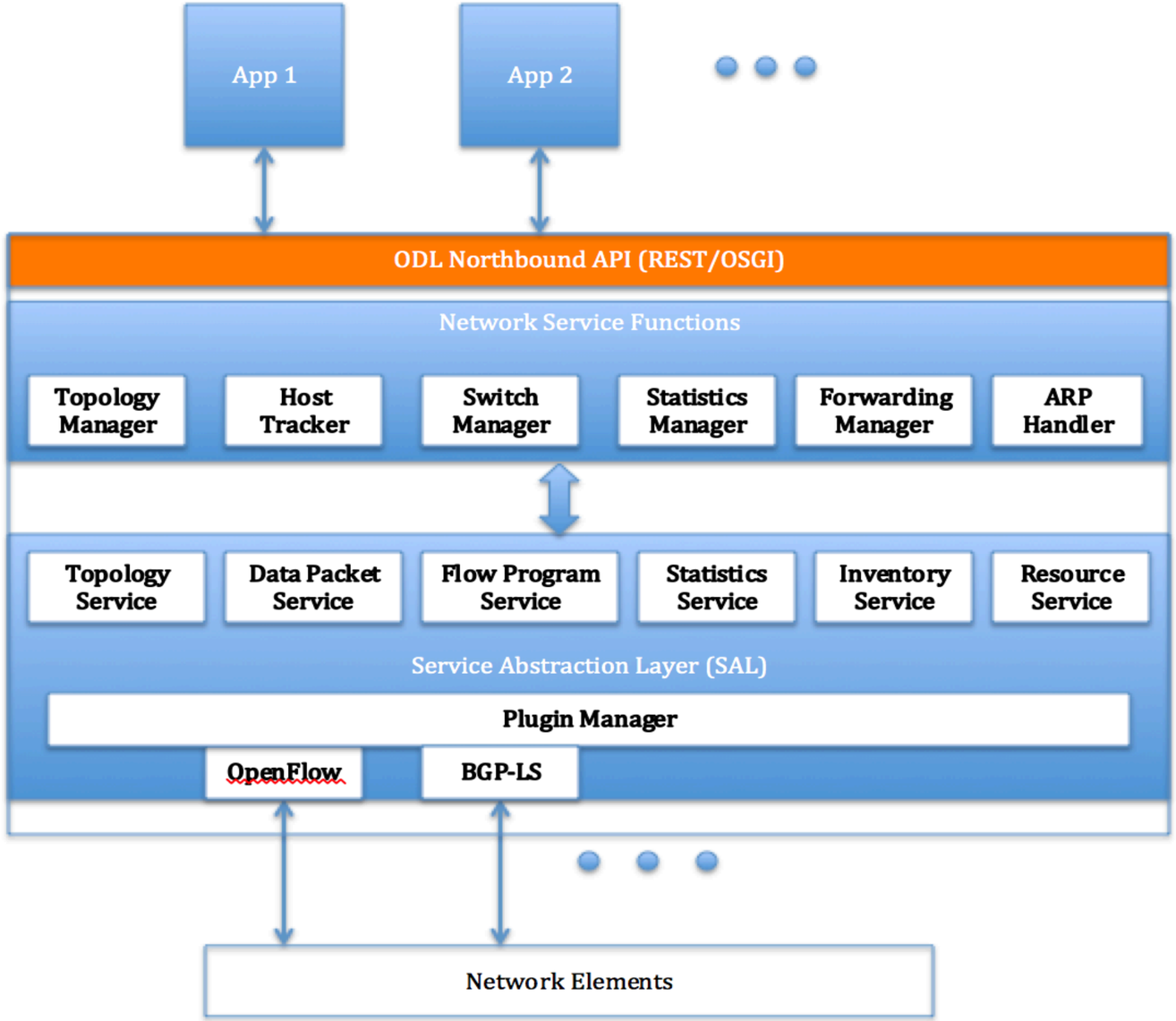
Members as of May 16, 2013 and growing

Open Daylight Controller Architecture – Service Abstraction Layer



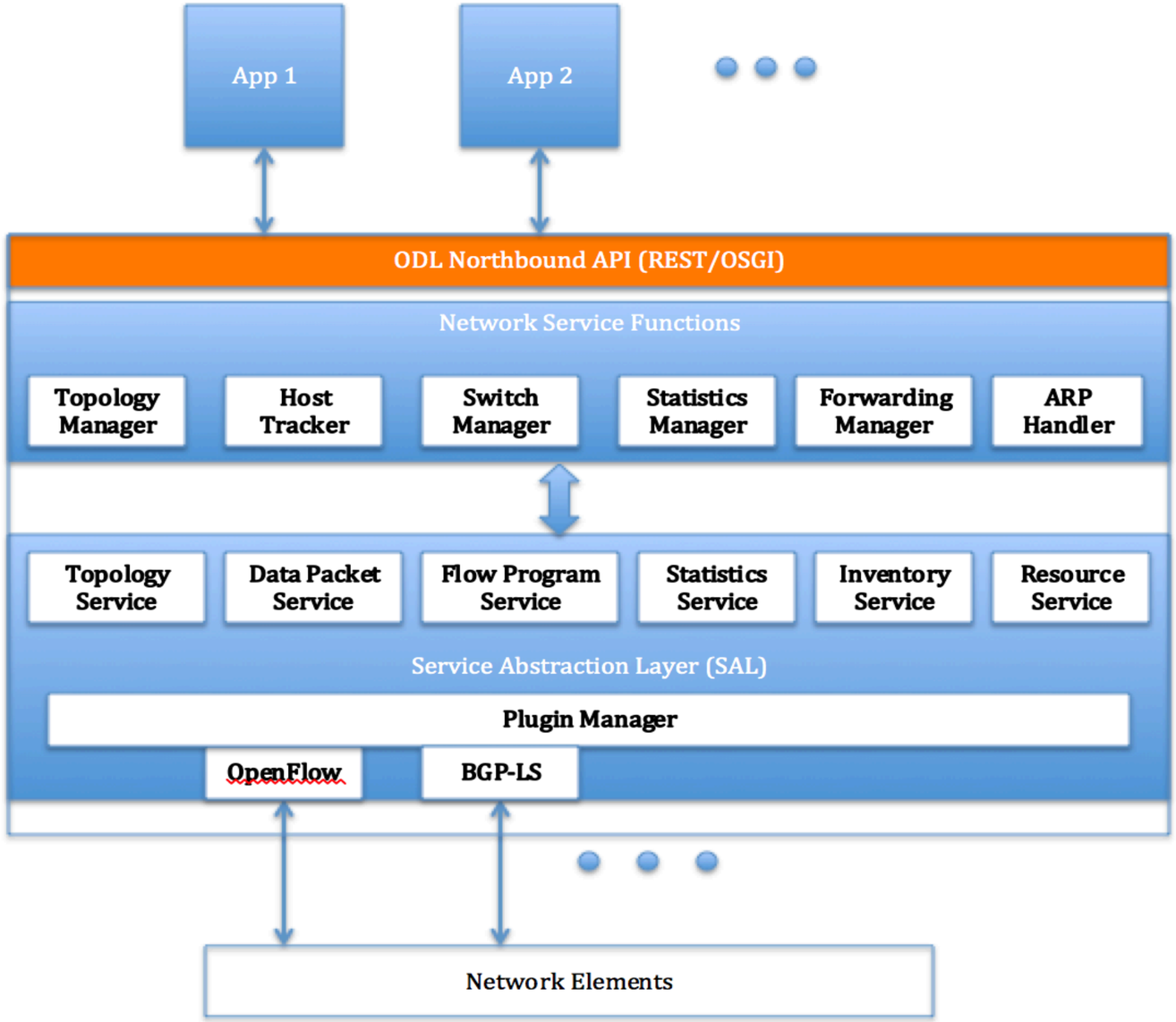
- A layer providing south-bound protocol and interface (SBPI) agnostic services to upper layers
- Each SBPI is incorporated in this layer as plugin
- Individual capabilities of each SBPI is abstracted

Open Daylight Controller Architecture – Service Abstraction Layer



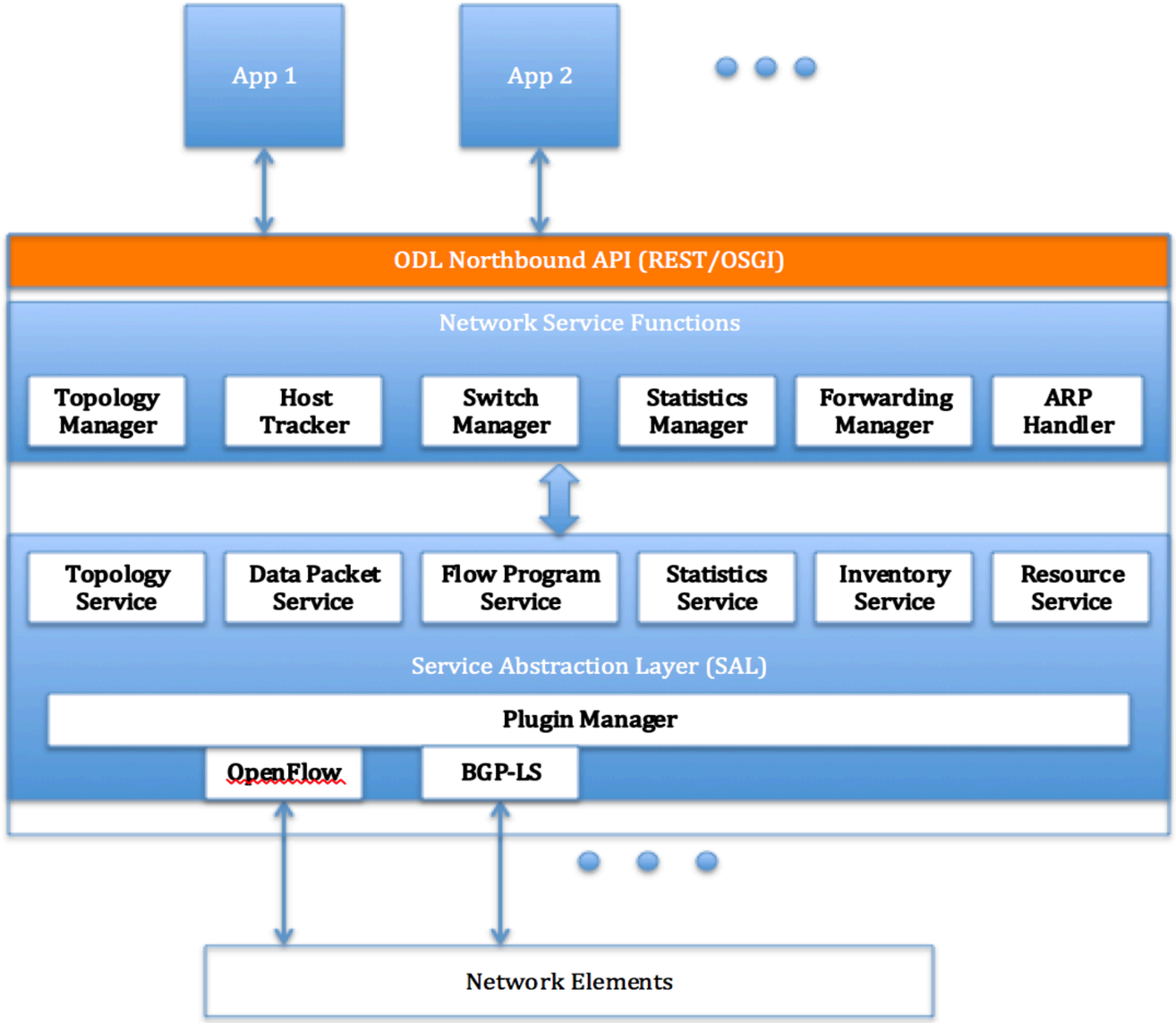
- Topology Service: To Discover nodes and links to be used to build topology
- Data Packet Service: To deliver packets from network devices to upper layer module (e.g. ARP Handler) or user applications
- Flow Programming Service – To program network elements with flow match/actions rules (such as based on OpenFlow)
- Statistics Service – To collect network statistics from network elements
- Inventory Service – To provide inventory information for network nodes and node connectors (ports)

Open Daylight Controller Architecture – Services/Business/ Orchestration Layer – Base Services



- Topology Manager: keeps track of discovered nodes and links
- Host Tracker: Tracks information on connected hosts
- Switch Manager: Information about network elements
- Statistics Manager: Statistics of the infrastructure
- Forwarding Manager: Deals with routes / forwarding tables
- ARP Handler: Deals with ARP
- Topology Manager makes use of information from other managers

Open Daylight Controller Architecture – NB API



- Applications outside of Controller can use REST based API to interact with Controller service
- Applications can also reside in the controller address space via the OSGi (Open Service Gateway initiative) framework
  - OSGi: Also known as dynamic module system for Java defines an architecture for modular application development

Open Daylight Flow Programming REST API Example

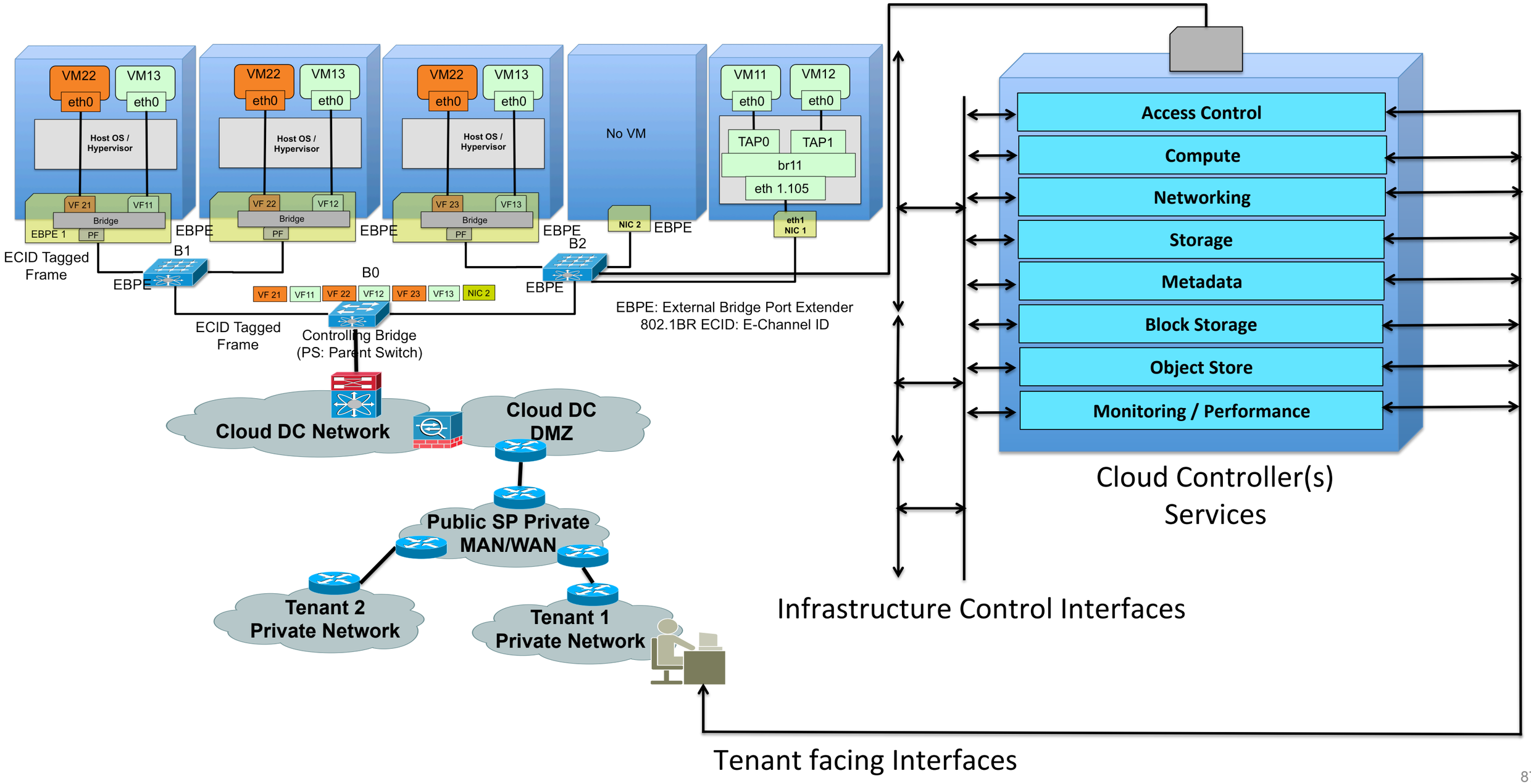
PUT http://localhost:8080/controller/nb/v2/**flowprogrammer**/default/node/OF/00:00:00:00:00:00:00:01/**staticFlow**/flow1

Request body:

```
{
   "installInHw":"true",
   "name":"flow1",
   "node":{ "id":"00:00:00:00:00:00:00:01", "type":"OF" },
   "ingressPort":"1",
   "priority":"500",
   "etherType":"0x800",
   "nwSrc":"9.9.1.1",
   "actions":[
      "OUTPUT=2”,
      “tosBits=46”
   ]
}
```

Cloud Framework Openstack

Cloud Controllers



Agenda

- General Introduction
- Openstack Services Introduction and Architecture
- Openstack Compute Service (Nova) Usage
- Openstack Networking (Neutron) Architecture and Usage

OpenStack: The Open Source Cloud Operating System

- A Cloud Platform
  - A collection of interrelated software components delivering capabilities to build and manage cloud infrastructure.
- A global community of developers devoted to innovation and openness
- Flexibility in deployment and features
- Standards for broad deployment
- No fear of vendor "lock-in"

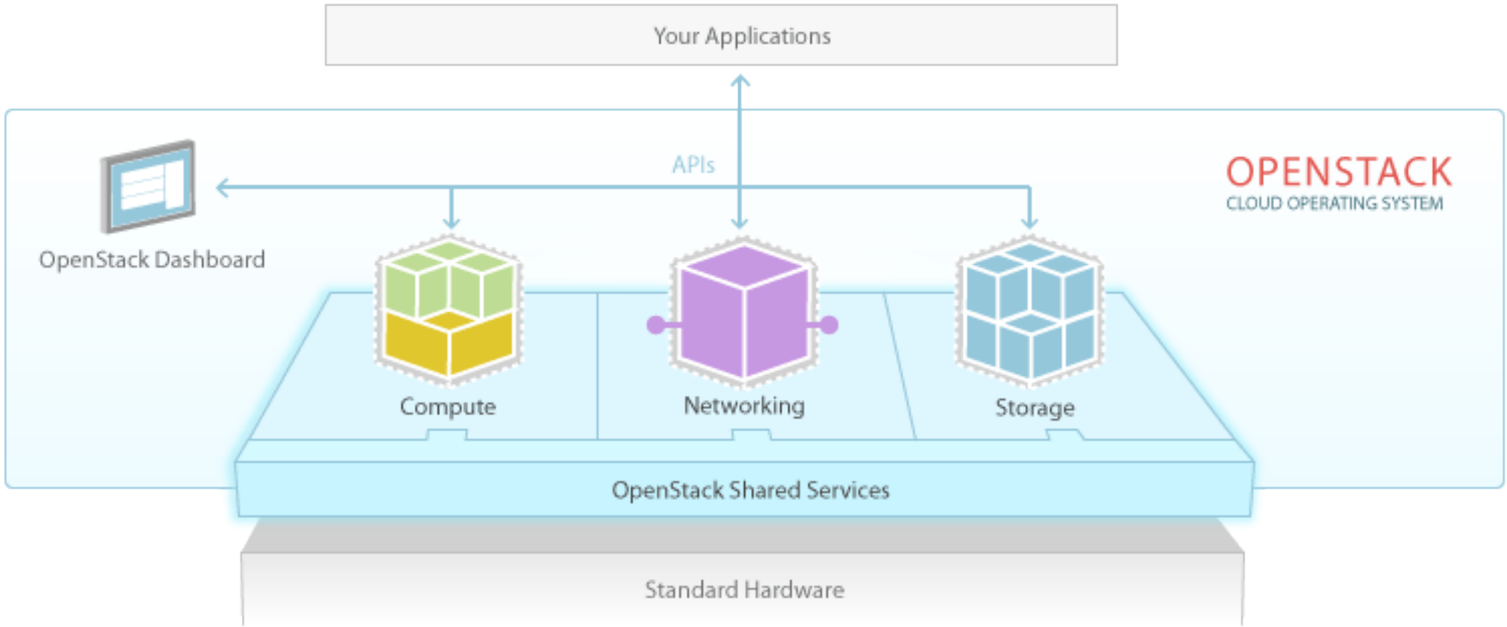


Release Cycle and Naming

- Austin: The first design summit took place in Austin, TX
- Bexar: San Antonio, TX (Bexar county).
- Cactus: Cactus is a city in Texas
- Diablo: Diablo is a city in the bay area near Santa Clara, CA
- Essex: Essex is a city near Boston, MA
- Folsom: Folsom is a city near San Francisco, CA
- Grizzly: San Diego, CA: Grizzly is an element of the state flag of California
- Havana: Havana is an unincorporated community in Oregon
- Icehouse: Ice House is a street in Hong Kong
- Juno: Juno is a locality in Georgia

https://wiki.openstack.org/wiki/Release_Naming

Openstack High Level View



Core Projects



Computing

Nova: Software to provision virtual machines on standard server hardware at massive scale

Glance: Image services for discovering, registering, and retrieving virtual machine images



Networking

Neutron: Software to provide "networking as a service" between interface devices managed by other Openstack services



Storing

Swift: Object storage to reliably store billions of objects distributed across standard server hardware

Cinder: Software to provide persistent block storage to Openstack services

Common Projects

- OpenStack Identity (Keystone)
- OpenStack Dashboard (Horizon)
- Telemetry (Ceilometer)
- Orchestration (Heat)
- OpenStack Common Library (Oslo)

Identity Service (Keystone) Introduction

- An authentication and authorization (AA) system
- Organized into a group of internal services
- Provides a HTTP front-end to clients (components, users, etc.)
- Support for multiple back-ends
  - Allows Keystone to adapt to a wide-range of environments

Keystone Architecture

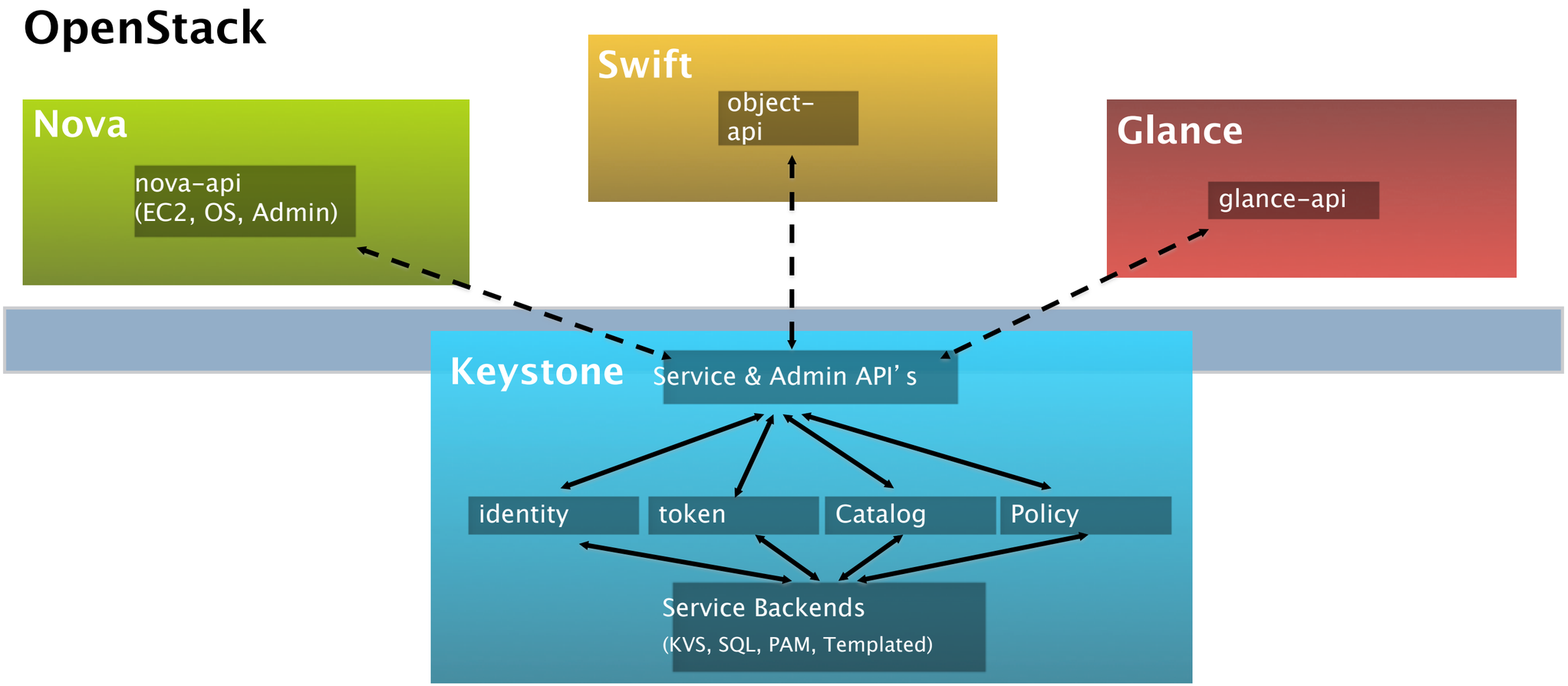
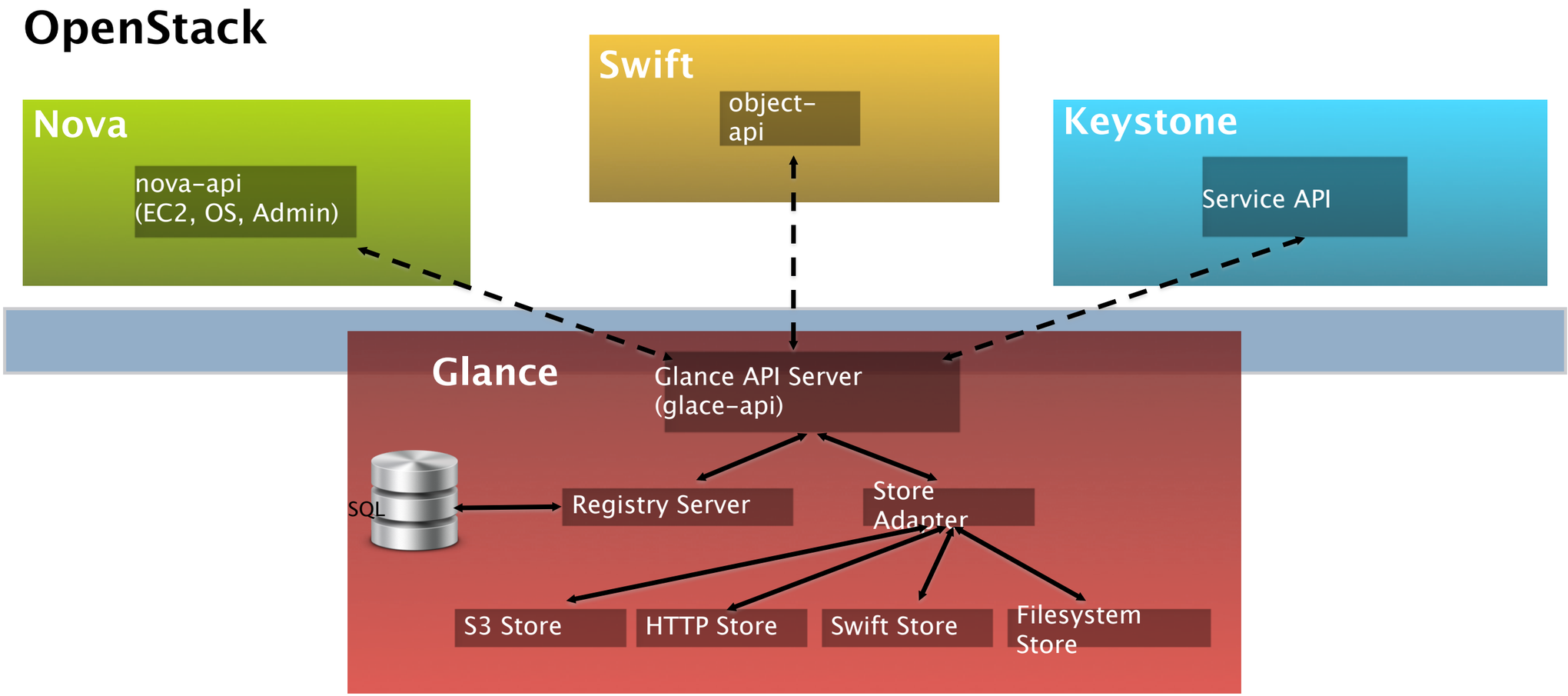


Image Service (Glance) Introduction

- Designed to be adaptable
- 3 primary services- API, Registry, and Store Adapter
- Provides services for discovering, registering, and retrieving virtual machine images
- Images can be stored in a variety of locations

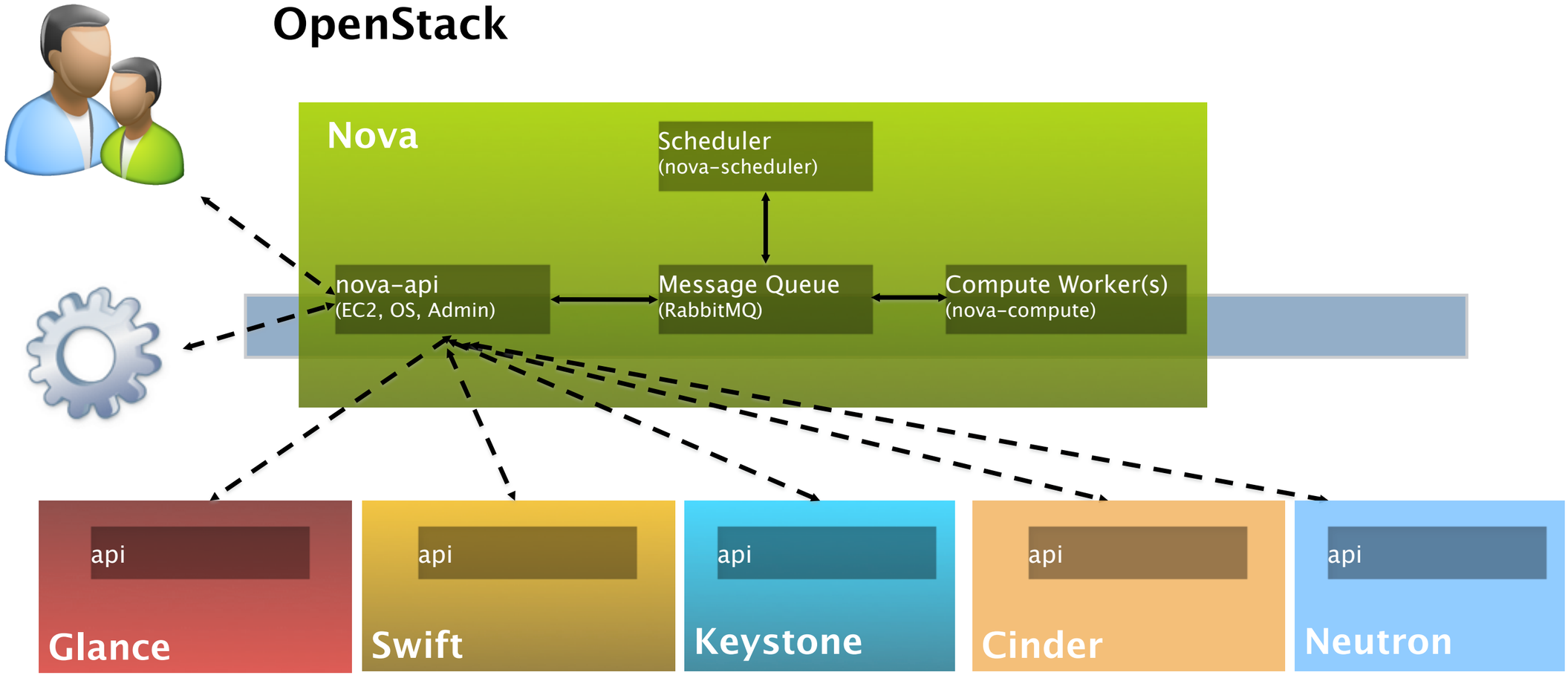
Glance Architecture



Compute Service (Nova) Introduction

- Primary component of the OpenStack IaaS platform
- Shared-nothing and modular software architecture
  - Services communicate using a message queue (AMQP)
- Uses a SQL-based database for storing information
  - Shared by all system components

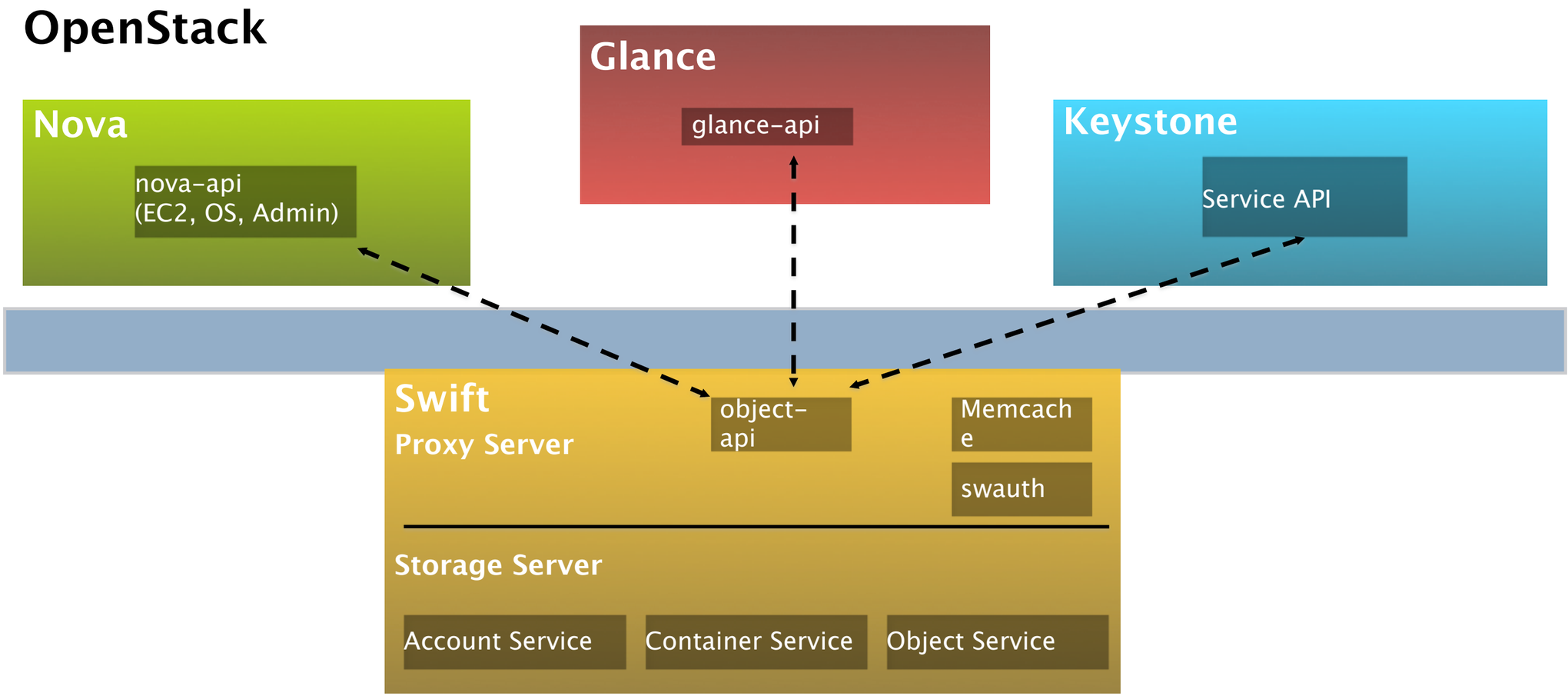
Nova Architecture



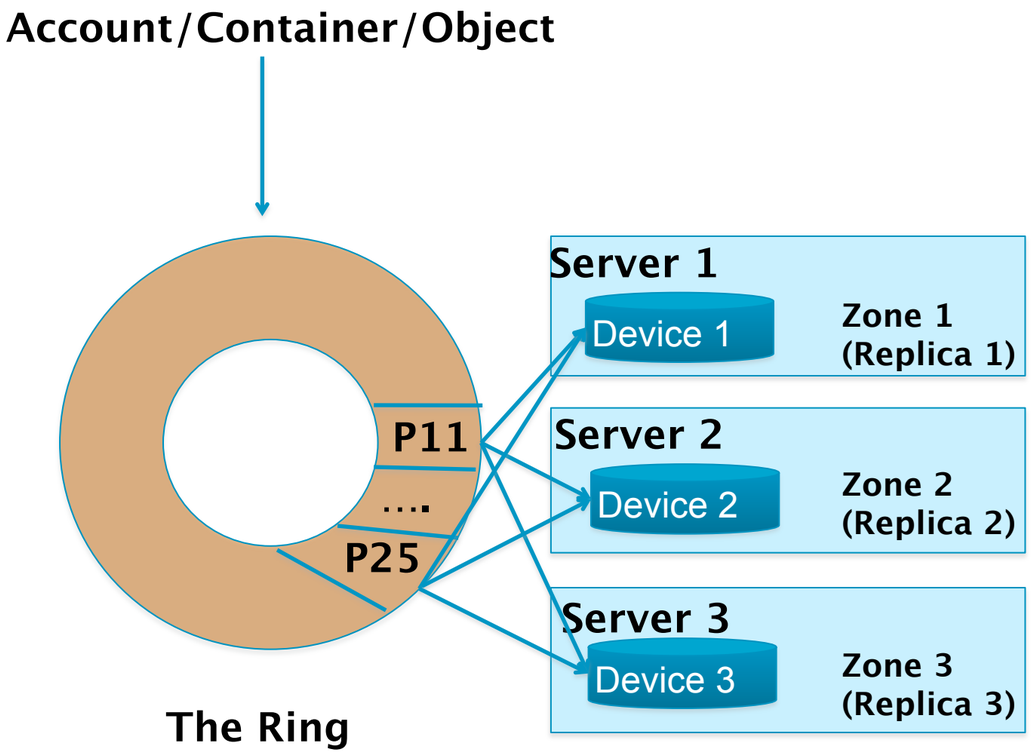
Object Store Service (Swift) Introduction

- Object-based storage system
  - NOT a block-based system like iSCSI
  - Consists of flexibly-sized data containers (objects)
- Distributed architecture
  - Avoids a single point of failure
  - Massively scalable (billions of objects and PB's of data)
- Server hard drives turn into a pool of storage

Swift Architecture



Swift Functionality



- The Ring
  - Mapping between entities and physical location
- Proxy Server
  - Ties together end users and the Swift back-end services, provides public API endpoint.
- Object Server
  - Simple blob storage server to store, retrieve, and delete objects on local disk
- Container Server
  - Lists objects
- Account Server
  - Similar to container, but lists containers

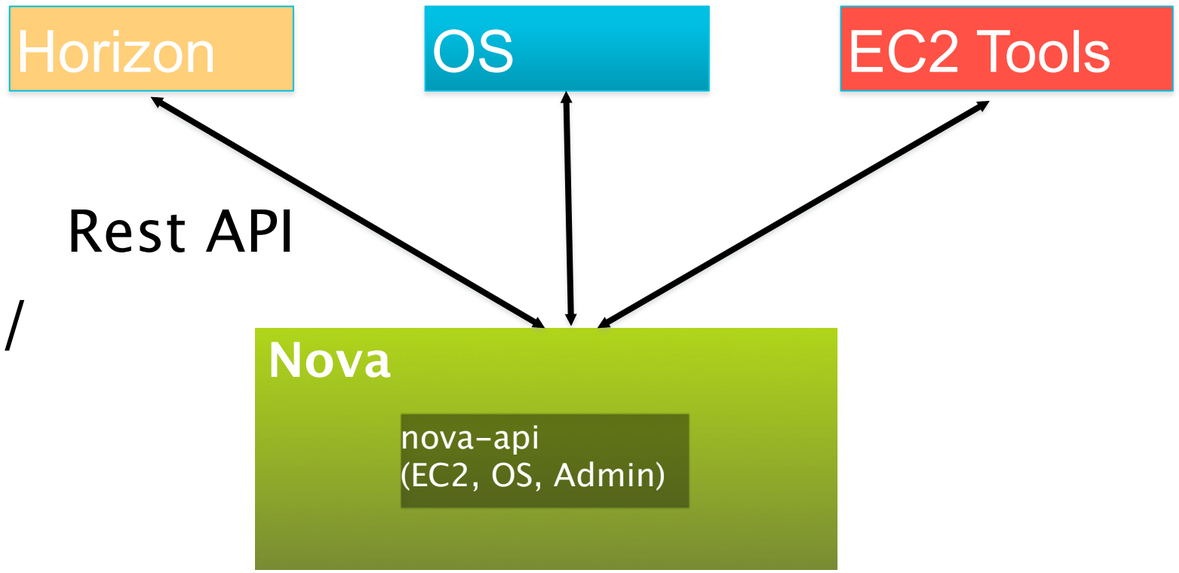
How to Deploy Openstack

- Devstack: http://devstack.org
- Automated:
  - Puppet
  - Chef
  - RDO, Juju and Charms

How to Use Openstack

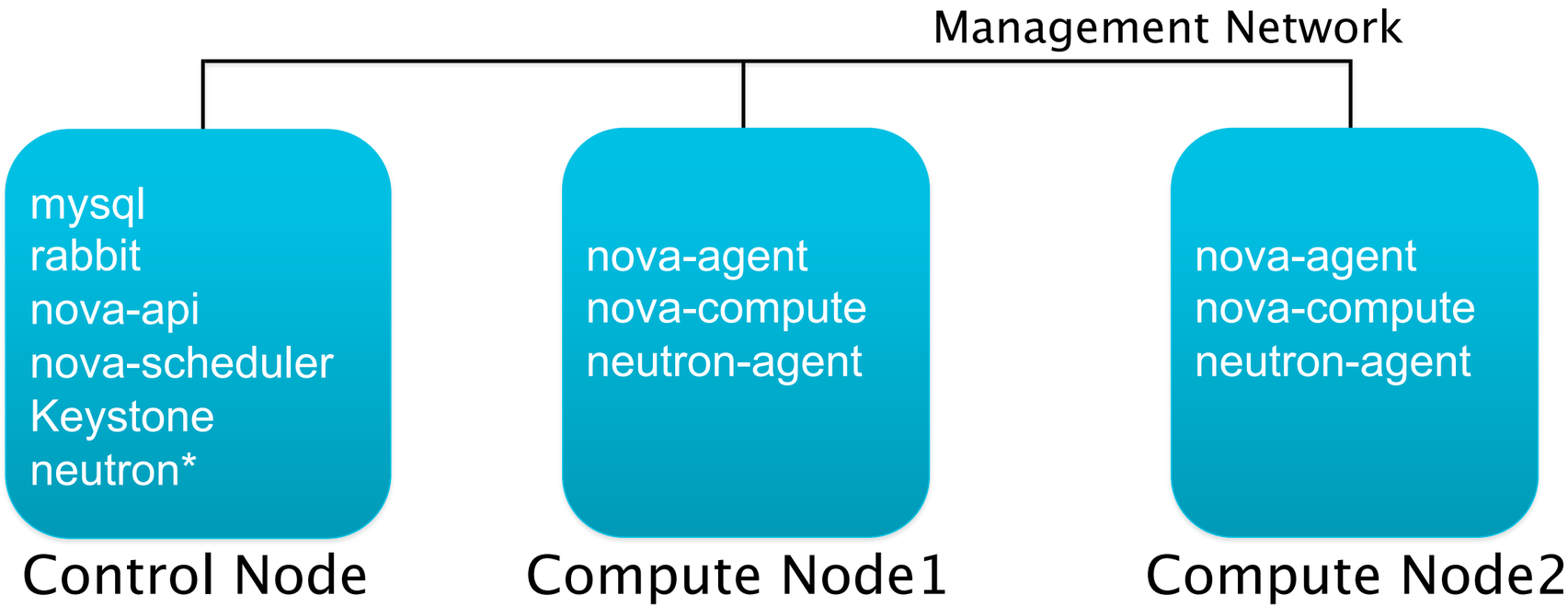
- GUI - Horizon
- CLI
- EC2 tools
- REST

http://docs.openstack.org/api/

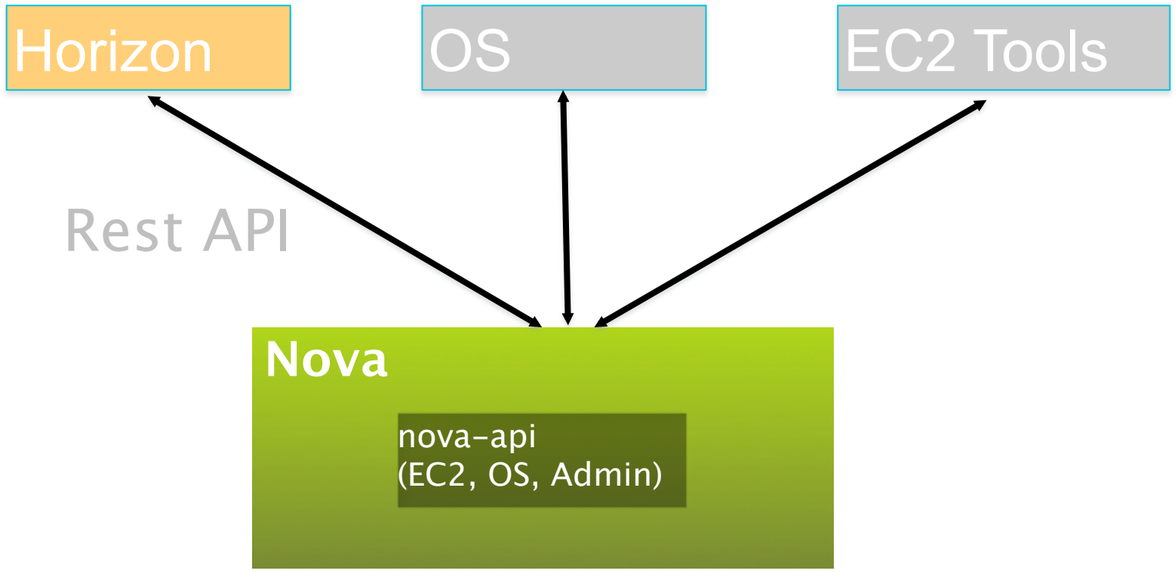


Demo Setup

- 3 nodes
  - Control node
  - 2 compute nodes



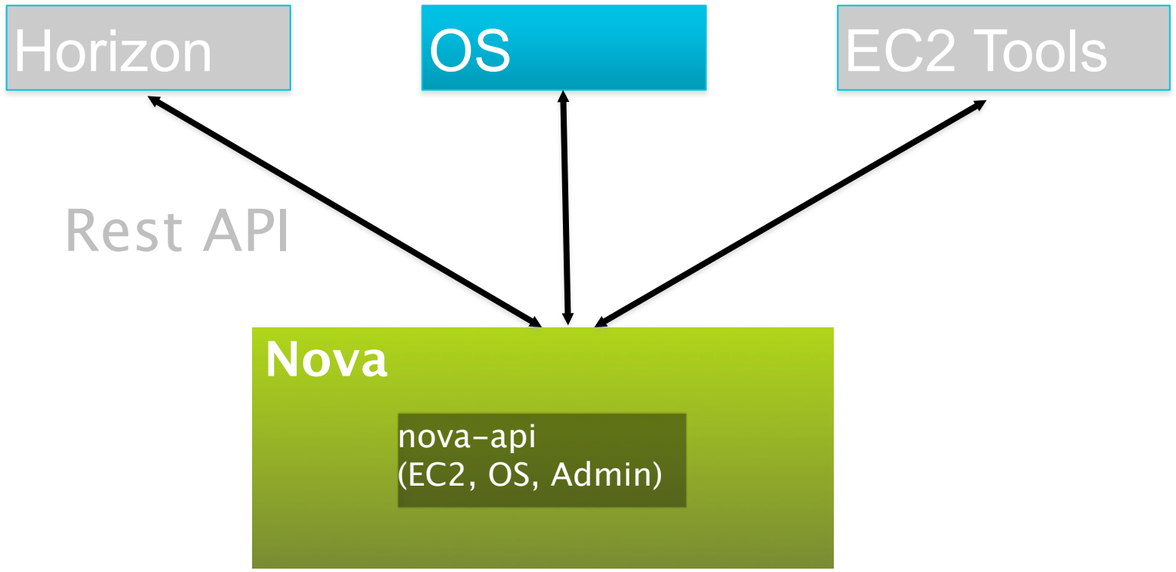
How to Use Openstack



Demo Video

[Nova Part 1]

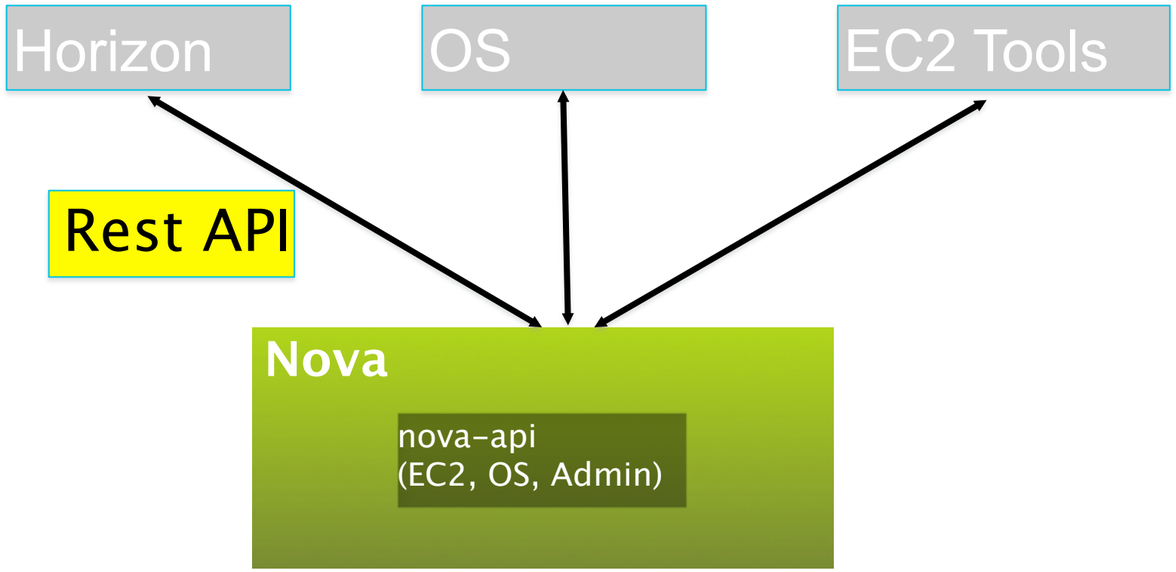
How to Use Openstack



Demo Video

[Nova Part 2]

How to Use Openstack

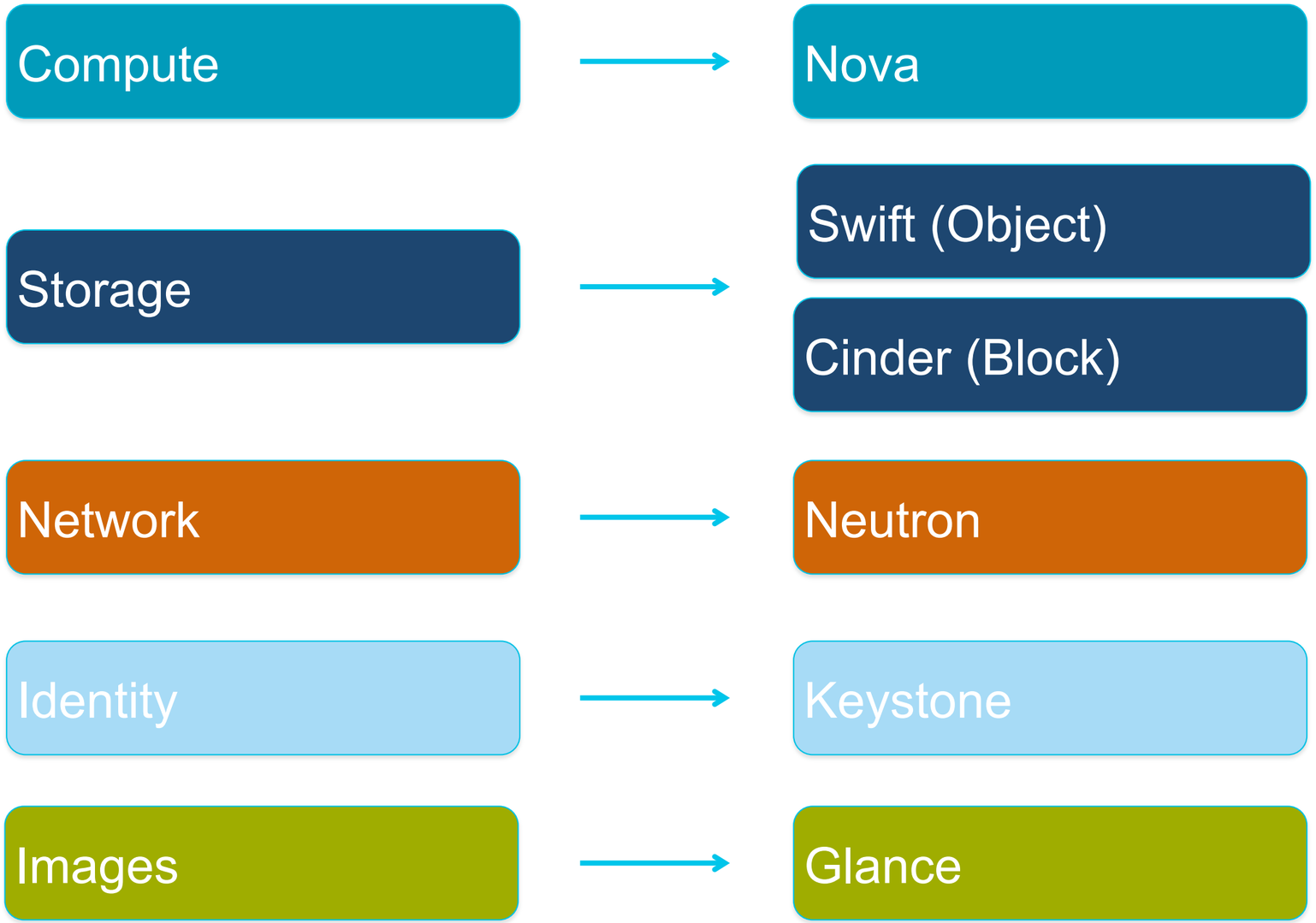


Demo Video

[Nova Part 3]

Neutron

DC Architecture mapped to OpenStack services



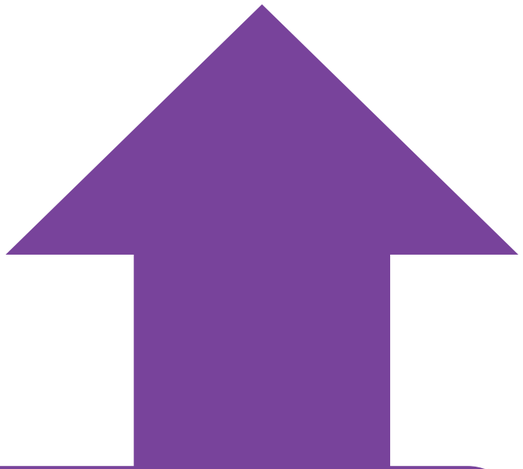
Neutron Architecture

Neutron API

API Extensions

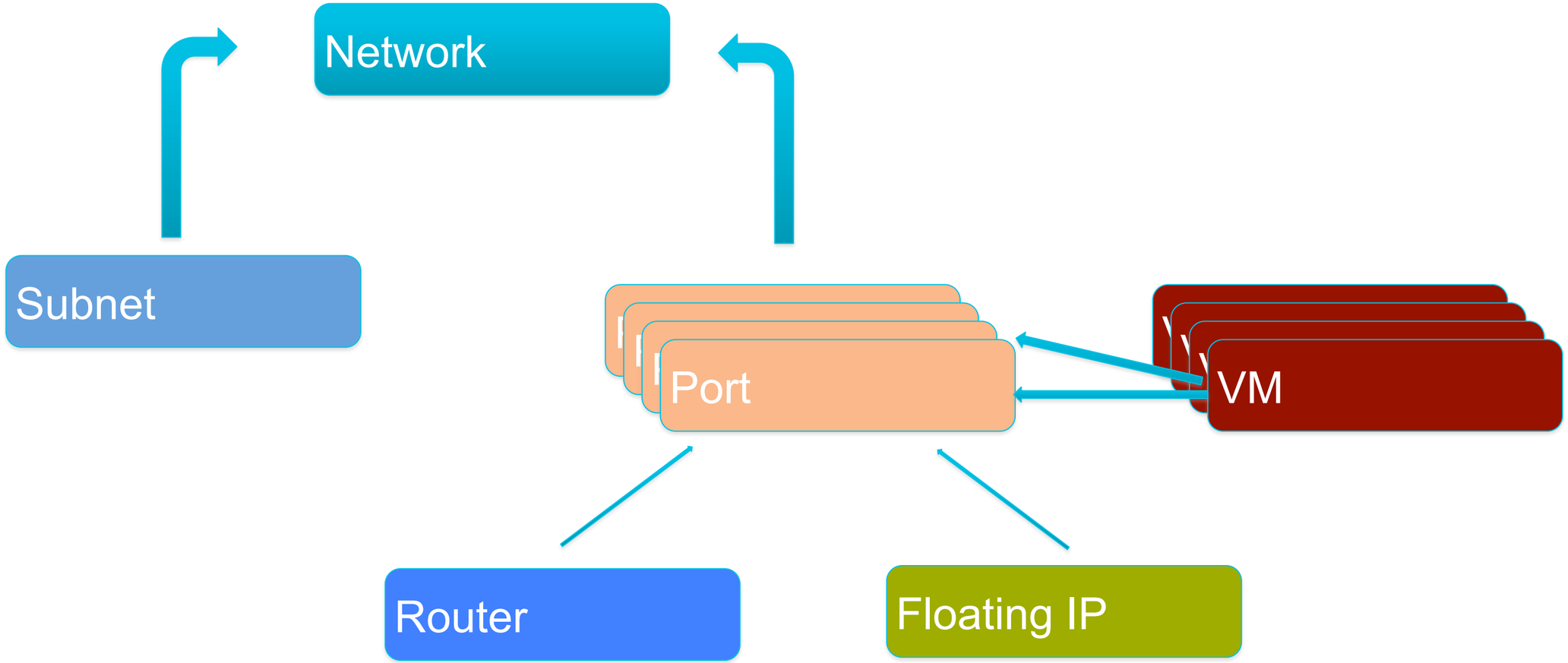
Neutron Service
- L2/L3 network abstraction and management
- Device and service attachment framework
- Does NOT implement any abstractions

Neutron Plug-in API



Vendor/User Plug-In
- Maps abstraction to implementation on physical network
- Makes all decisions about *how* a network is implemented
- Can provide additional features through API extensions

Neutron Network Abstraction

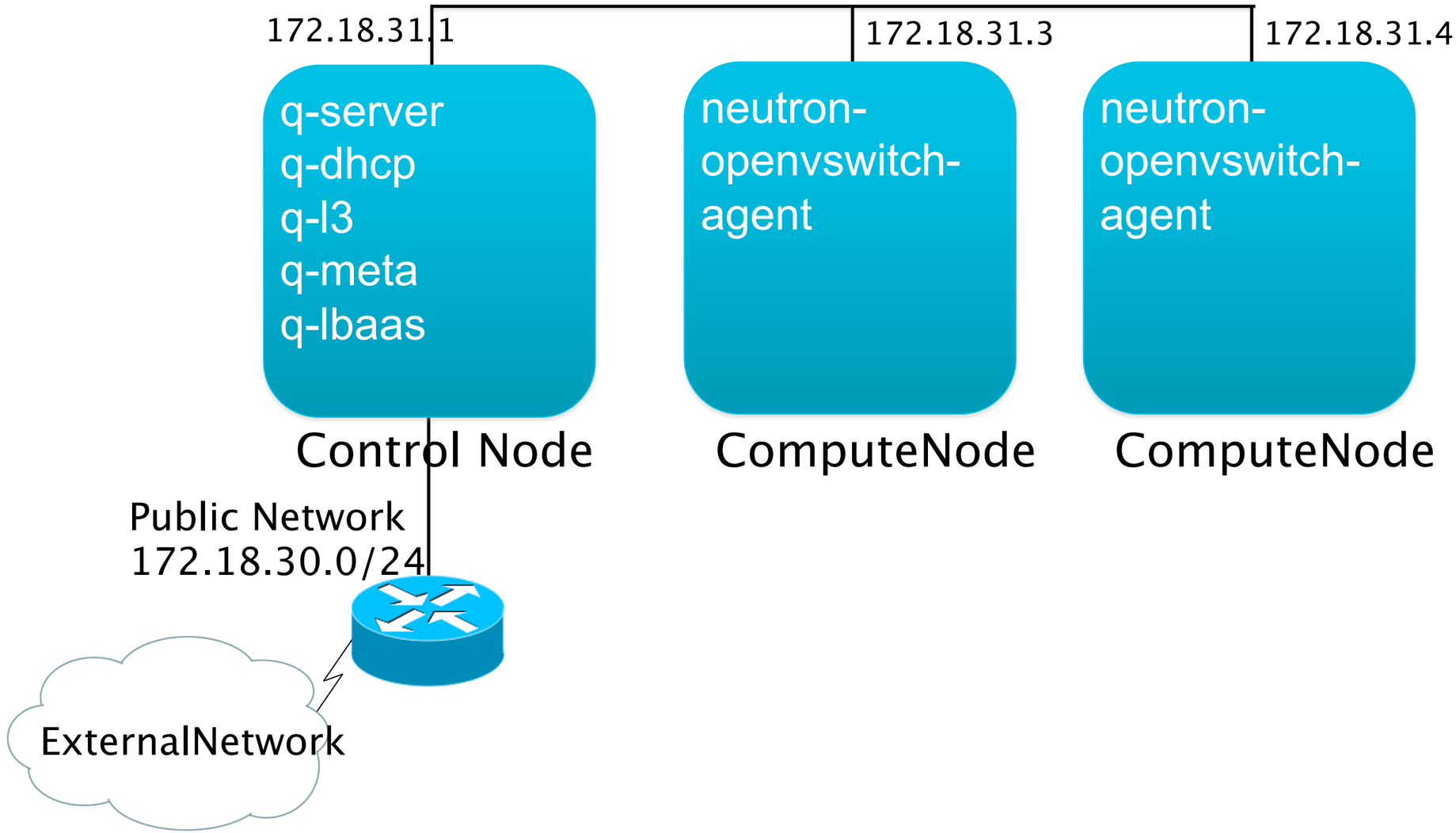


Neutron Realization and Plugins

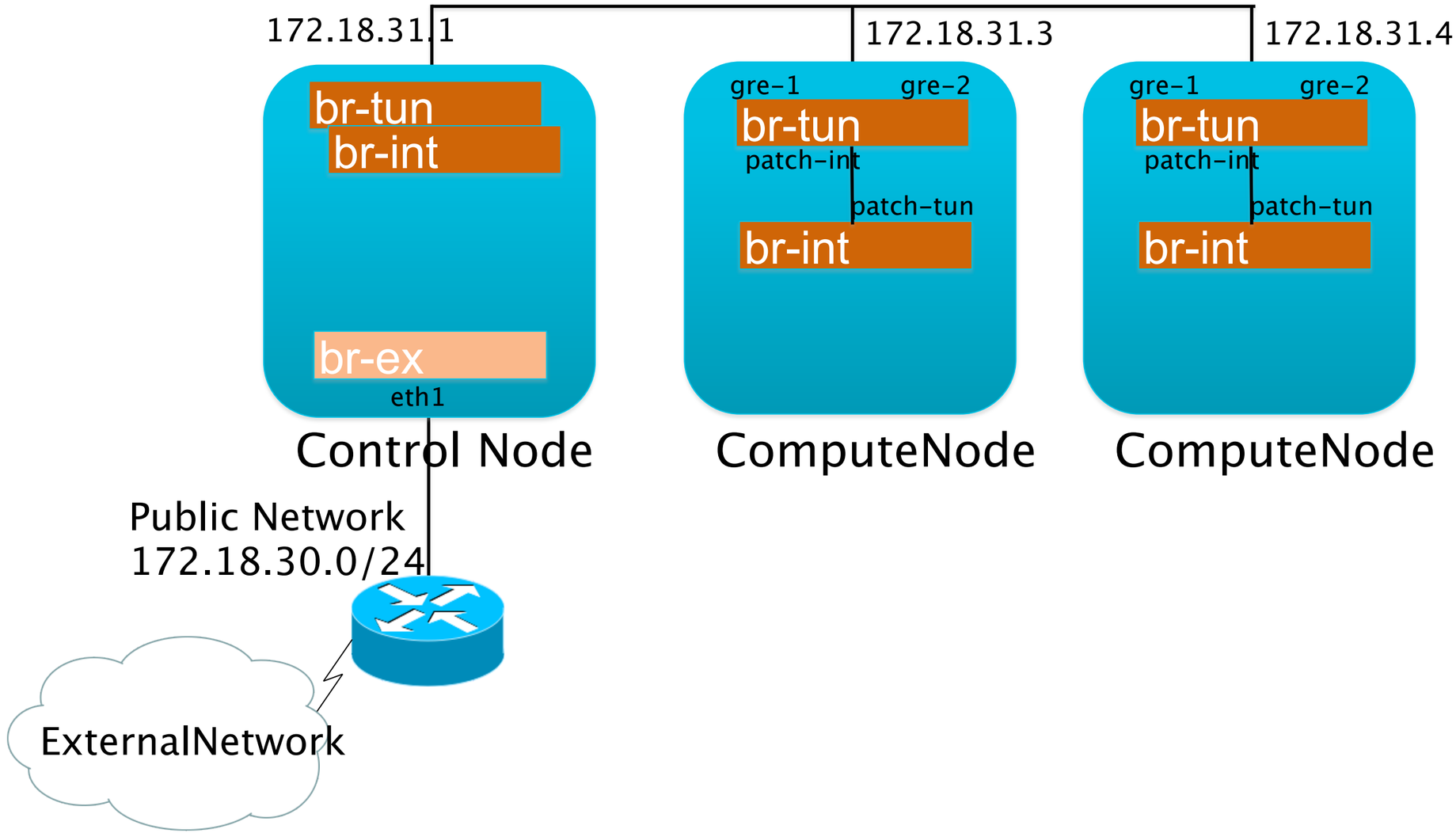
- Network Technologies
  - VLANs
  - Tunnels (GRE, VXLAN, …)
- Plugins
  - Open vSwitch Plugin
  - Cisco UCS/Nexus Plugin
  - Linux Bridge Plugin
  - OpenFlow Controller Plugin (different vendors)

Neutron Lab

Demo Setup


172.18.31.1
172.18.31.3
172.18.31.4
q-server
q-dhcp
q-l3
q-meta
q-lbaas
neutron-openvswitch-agent
neutron-openvswitch-agent
Control Node
ComputeNode
ComputeNode
Public Network
172.18.30.0/24
ExternalNetwork

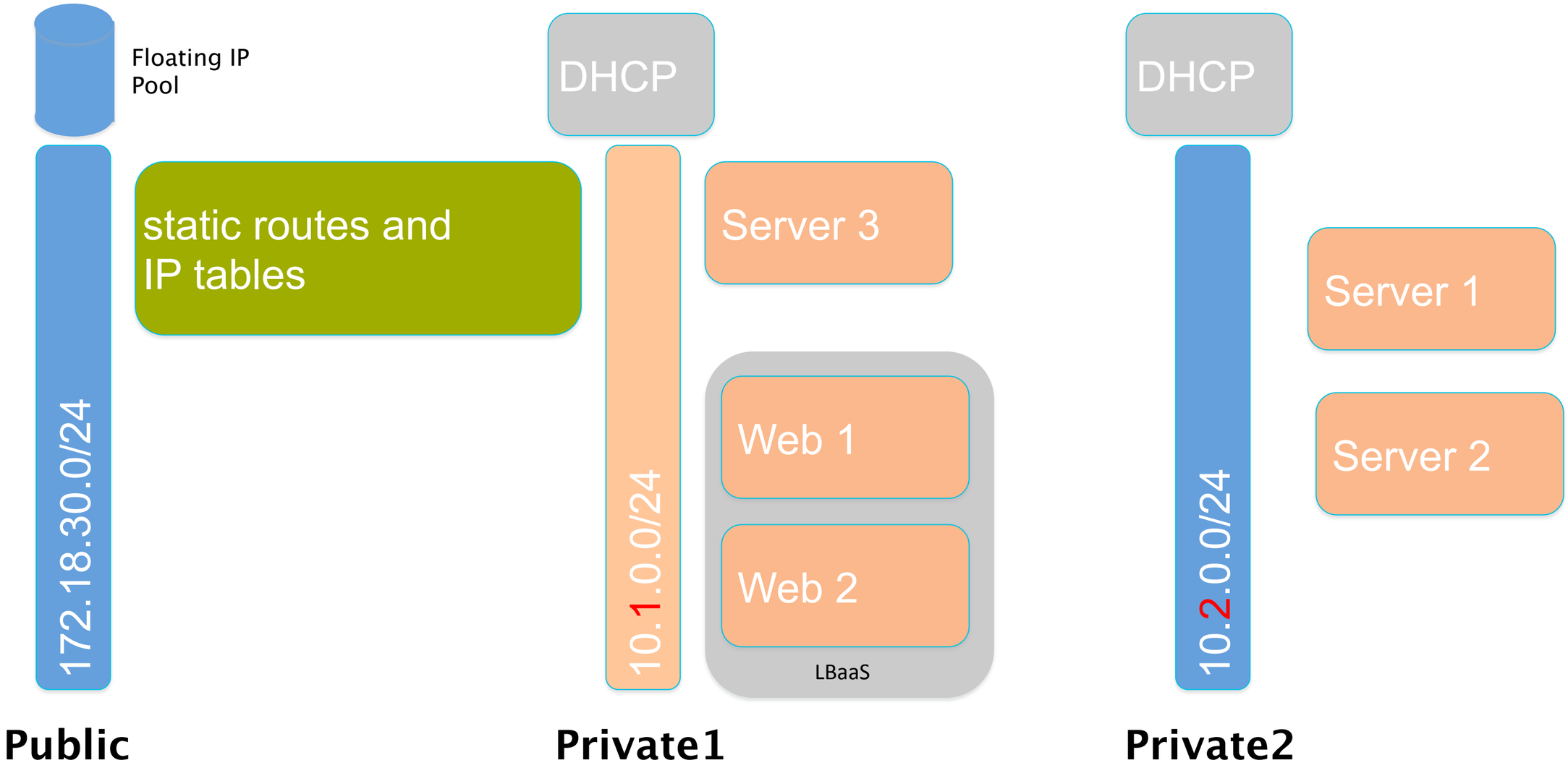
Demo Setup (OVS)



Neutron Lab

- Create 2 networks
- Create VMs on each network
- Discuss L2 connectivity, DHCP, show ping
- Create Router between the 2 networks, show ping
- Add public network to router, discuss NAT, ping external server
- Floating IPs
- Neutron Security Groups
- Neutron LBaas
- Meta Data Server

Demo Scenario



Network Namespaces

```
openstack@control:~/devstack$ source openrc admin admin
openstack@control:~/devstack$ neutron net-list
+--------------------------------------+----------+--------------------------------------------------+
| id                                   | name     | subnets                                          |
+--------------------------------------+----------+--------------------------------------------------+
| 036460df-90d3-45b2-b063-1ece8d0bd061 | public   | daaed30b-f2fc-4c28-a1c9-52ab008cf474 172.18.30.0/24 |
| 5fdb3fc3-2caf-4c06-87e3-0c7b1f4ce6a3 | private1 | 9c3e6b43-da73-4841-972a-18b57186ab81 10.1.0.0/24    |
| d74c68be-a785-4a62-9e4f-81e66693fc69 | private2 | 57c9434d-94ff-4e24-ac35-eb4305b74448 10.2.0.0/24    |
+--------------------------------------+----------+--------------------------------------------------+
openstack@control:~/devstack$ ip netns list
qdhcp-036460df-90d3-45b2-b063-1ece8d0bd061
qdhcp-5fdb3fc3-2caf-4c06-87e3-0c7b1f4ce6a3
qdhcp-d74c68be-a785-4a62-9e4f-81e66693fc69
```

Network Namespaces

```
openstack@control:~/devstack$ NET_ID=`neutron net-list | grep private2 | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack$ sudo ip netns exec qdhcp-$NET_ID ip addr list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
13: tap7d16f078-f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:1d:b8:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.2.0.2/24 brd 10.2.0.255 scope global tap7d16f078-f6
    inet6 fe80::f816:3eff:fe1d:b8d3/64 scope link
       valid_lft forever preferred_lft forever
```

DHCP Server

```
openstack@control:~/devstack$ ps aux | grep $NET_ID

nobody   26628  0.0  0.0  28900   876 ?        S    15:34   0:00 dnsmasq --no-hosts --no-resolv --strict-order --bind-
interfaces --interface=tap7d16f078-f6 --except-interface=lo --pid-file=/opt/stack/data/neutron/dhcp/d74c68be-
a785-4a62-9e4f-81e66693fc69[m[K/pid --dhcp-hostsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/host --
dhcp-optsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/opts --dhcp-script=/opt/stack/neutron/bin/
neutron-dhcp-agent-dnsmasq-lease-update --leasefile-ro --dhcp-range=set:tag0,10.2.0.0,static,120s --conf-file= --
domain=openstacklocal

root     26629  0.0  0.0  28900   244 ?        S    15:34   0:00 dnsmasq --no-hosts --no-resolv --strict-order --bind-
interfaces --interface=tap7d16f078-f6 --except-interface=lo --pid-file=/opt/stack/data/neutron/dhcp/d74c68be-
a785-4a62-9e4f-81e66693fc69/pid --dhcp-hostsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/host --dhcp-
optsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/opts --dhcp-script=/opt/stack/neutron/bin/neutron-
dhcp-agent-dnsmasq-lease-update --leasefile-ro --dhcp-range=set:tag0,10.2.0.0,static,120s --conf-file= --domain=openstacklocal

root     26653  0.0  1.1  93376 22636 ?        S    15:34   0:00 /usr/bin/python /usr/local/bin/neutron-ns-metadata-proxy --
pid_file=/opt/stack/data/neutron/external/pids/[01;31m[Kd74c68be-a785-4a62-9e4f-81e66693fc69[m[K.pid --
network_id=[01;31m[Kd74c68be-a785-4a62-9e4f-81e66693fc69[m[K --state_path=/opt/stack/data/neutron --metadata_port=80 --debug --
verbose
```

OVS

```
openstack@control:~/devstack$ sudo ovs-vsctl show | more
9cd9bbc0-c130-437c-bee7-0aaa1dfa98c3
    Bridge br-ex
        Port "eth1"
            Interface "eth1"
        Port br-ex
            Interface br-ex
                type: internal
   more…
```

OVS

```
Bridge br-int
        Port br-int
            Interface br-int
                type: internal
        Port "tapf4824d73-da"
            tag: 2
            Interface "tapf4824d73-da"
                type: internal
        Port patch-tun
            Interface patch-tun
                type: patch
                options: {peer=patch-int}
  more…
```

OVS

```
        Port "tap7d16f078-f6"
            tag: 1
            Interface "tap7d16f078-f6"
                type: internal
        Port "tap0150932a-03"
            tag: 3
            Interface "tap0150932a-03"
                type: internal
more…
```

OVS

```
Bridge br-tun
    Port "gre-2"
        Interface "gre-2"
            type: gre
            options: {in_key=flow, out_key=flow, remote_ip="172.18.31.3"}
    Port patch-int
        Interface patch-int
            type: patch
            options: {peer=patch-tun}
    Port "gre-3"
        Interface "gre-3"
            type: gre
            options: {in_key=flow, out_key=flow, remote_ip="172.18.31.4"}
    Port br-tun
        Interface br-tun
            type: internal
ovs_version: "1.9.0"
```

Switch Context to Tenant User

```
openstack@control:~/devstack$ cd context_im_user

openstack@control:~/devstack/context_im_user$ source imrc

openstack@control:~/devstack/context_im_user$ env | grep OS_

OS_PASSWORD=Gh3nt

OS_AUTH_STRATEGY=keystone

OS_AUTH_URL=http://172.18.31.1:5000/v2.0/

OS_USERNAME=im_user

OS_TENANT_NAME=iee_im

OS_CACERT=/opt/stack/data/CA/int-ca/ca-chain.pem

OS_NO_CACHE=1
```

Upload Images to Glance

```
openstack@control:~/devstack/context_im_user$ glance index
ID                                   Name                           Disk Format          Container Format     Size
------------------------------------ ------------------------------ -------------------- ------------------ --------------
b3ff5600-c04a-41fc-83af-3c0ec82540d3 cirros-0.3.1-x86_64-uec-ramdis ari                  ari                        3714968
5bbb1bf3-e080-4e9e-bf16-436cf4cddc6f cirros-0.3.1-x86_64-uec        ami                  ami                       25165824
fda13171-d675-41a8-9f49-913b4583f4ec cirros-0.3.1-x86_64-uec-kernel aki                  aki                        4955792

openstack@control:~/devstack/context_im_user$ glance add name=f17-x86_64-openstack-sda is_public=true disk_format=qcow2
container_format=bare < ../images/f17-x86_64-openstack-sda.qcow2

Added new image with ID: 65c4804c-fe8b-42f9-b127-835924b9117d

openstack@control:~/devstack/context_im_user$ glance add name=precise-amd64 is_public=true disk_format=qcow2
container_format=ovf < ../images/precise-server-cloudimg-amd64-disk1.img

Added new image with ID: e239bfe9-5101-4a8b-a4cc-ed486e1d6398

openstack@control:~/devstack/context_im_user$ IMAGE_ID=`glance image-list | grep f17-x86_64-openstack-sda | awk -F' ' '{ print
$2 }'`
```

Neutron Security Groups

```
openstack@control:~/devstack/context_im_user$ neutron security-group-list

+--------------------------------------+---------+-------------+
| id                                   | name    | description |
+--------------------------------------+---------+-------------+
| f7d59e63-bafa-4a3d-a3b8-c0870d293b63 | default | default     |
+--------------------------------------+---------+-------------+
```

Neutron Security Groups

```
openstack@control:~/devstack/context_im_user$ neutron security-group-create web_server --description 'IM demo security group'

Created a new security_group:

+-----------------------------------------------------------------------------------------+
| Field                | Value
|
+-----------------------------------------------------------------------------------------+
| description          | IM demo security group
|
| id                   | 2bd1c2ed-8de2-47fc-947b-56220556777f
|
| name                 | web_server
|
| security_group_rules | {"remote_group_id": null, "direction": "egress", "remote_ip_prefix": null, "protocol": null,
"tenant_id": "e815e6d44c6747308d30adabff1dac99", "port_range_max": null, "security_group_id":
"2bd1c2ed-8de2-47fc-947b-56220556777f", "port_range_min": null, "ethertype": "IPv4", "id":
"bc392565-2f9d-4fb8-80a5-98895260f415"} |
|                      | {"remote_group_id": null, "direction": "egress", "remote_ip_prefix": null, "protocol": null,
"tenant_id": "e815e6d44c6747308d30adabff1dac99", "port_range_max": null, "security_group_id":
"2bd1c2ed-8de2-47fc-947b-56220556777f", "port_range_min": null, "ethertype": "IPv6", "id": "a9779f42-1225-45e1-
a9a8-27ecd94e334e"} |
| tenant_id            | e815e6d44c6747308d30adabff1dac99
|
+-----------------------------------------------------------------------------------------+
```

Neutron Security Groups

```
openstack@control:~/devstack/context_im_user$ neutron security-group-rule-create --direction ingress --protocol tcp  --
port_range_min 22 --port_range_max 22 web_server

Created a new security_group_rule:

+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| direction         | ingress                              |
| ethertype         | IPv4                                 |
| id                | 44123185-099f-457e-a9e0-1fb37c104889 |
| port_range_max    | 22                                   |
| port_range_min    | 22                                   |
| protocol          | tcp                                  |
| remote_group_id   |                                      |
| remote_ip_prefix  |                                      |
| security_group_id | 2bd1c2ed-8de2-47fc-947b-56220556777f |
| tenant_id         | e815e6d44c6747308d30adabff1dac99     |
+-------------------+--------------------------------------+
```

Neutron Security Groups

```
openstack@control:~/devstack/context_im_user$ neutron security-group-rule-create --direction ingress --protocol tcp  --
port_range_min 80 --port_range_max 80 web_server

Created a new security_group_rule:

+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| direction         | ingress                              |
| ethertype         | IPv4                                 |
| id                | 4eaa761b-4c51-4e71-82fb-da13969d24e7 |
| port_range_max    | 80                                   |
| port_range_min    | 80                                   |
| protocol          | tcp                                  |
| remote_group_id   |                                      |
| remote_ip_prefix  |                                      |
| security_group_id | 2bd1c2ed-8de2-47fc-947b-56220556777f |
| tenant_id         | e815e6d44c6747308d30adabff1dac99     |
+-------------------+--------------------------------------+
```

Neutron Security Groups

```
openstack@control:~/devstack/context_im_user$ neutron security-group-rule-create --direction ingress --protocol icmp web_server

Created a new security_group_rule:

+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| direction         | ingress                              |
| ethertype         | IPv4                                 |
| id                | 8790f25d-84d7-4275-8435-f9ebf271cce2 |
| port_range_max    |                                      |
| port_range_min    |                                      |
| protocol          | icmp                                 |
| remote_group_id   |                                      |
| remote_ip_prefix  |                                      |
| security_group_id | 2bd1c2ed-8de2-47fc-947b-56220556777f |
| tenant_id         | e815e6d44c6747308d30adabff1dac99     |
+-------------------+--------------------------------------+

openstack@control:~/devstack/context_im_user$ SECGROUP_ID=`neutron security-group-list | grep web_server | awk -F' ' '{ print
$2 }'`
```

Neutron Port

```
openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep private2 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name server1_port $NET_ID

Created a new port:

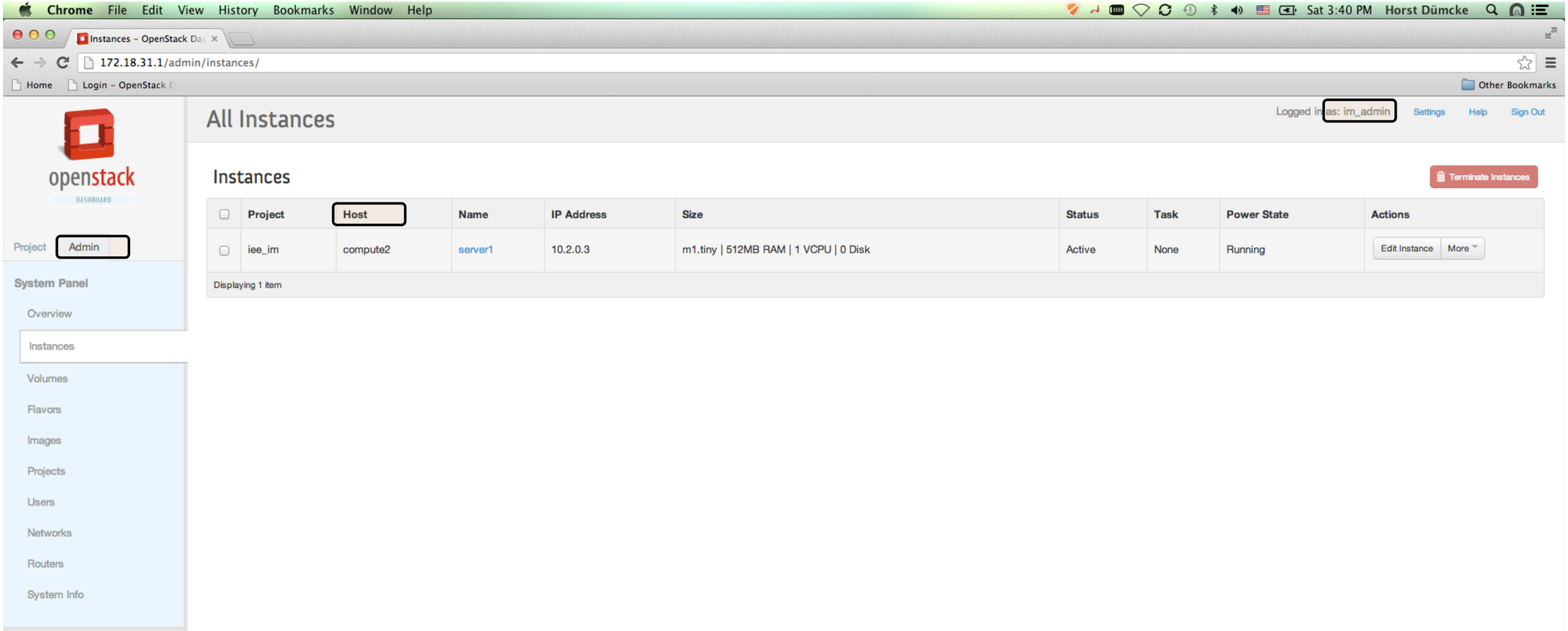
+-----------------+-------------------------------------------------------------------------------+
| Field           | Value                                                                         |
+-----------------+-------------------------------------------------------------------------------+
| admin_state_up  | True                                                                          |
| device_id       |                                                                               |
| device_owner    |                                                                               |
| fixed_ips       | {"subnet_id": "57c9434d-94ff-4e24-ac35-eb4305b74448", "ip_address": "10.2.0.3"} |
| id              | 40070dfd-5900-431d-bb85-581a9ab60223                                          |
| mac_address     | fa:16:3e:a2:b4:2e                                                             |
| name            | server1_port                                                                  |
| network_id      | d74c68be-a785-4a62-9e4f-81e66693fc69                                          |
| security_groups | 2bd1c2ed-8de2-47fc-947b-56220556777f                                          |
| status          | DOWN                                                                          |
| tenant_id       | e815e6d44c6747308d30adabff1dac99                                              |
+-----------------+-------------------------------------------------------------------------------+
```

Nova Boot Server

```
openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID server1

+-----------------------------+--------------------------------------+
| Property                    | Value                                |
+-----------------------------+--------------------------------------+
| status                      | BUILD                                |
| updated                     | 2013-05-25T13:38:29Z                 |
| OS-EXT-STS:task_state       | scheduling                           |
| key_name                    | None                                 |
| image                       | f17-x86_64-openstack-sda             |
| hostId                      |                                      |
| OS-EXT-STS:vm_state         | building                             |
| flavor                      | m1.tiny                              |
| id                          | dbbeec72-92e2-4285-960e-32d01bdc0b50 |
| security_groups             | [{u'name': u'default'}]              |
| user_id                     | 8da345659e694f7f8d8138af8fbd9c42     |
| name                        | server1                              |
| adminPass                   | CUXEiPk4LyhY                         |
| tenant_id                   | e815e6d44c6747308d30adabff1dac99     |
| created                     | 2013-05-25T13:38:29Z                 |
| OS-DCF:diskConfig           | MANUAL                               |
| metadata                    | {}                                   |
| accessIPv4                  |                                      |
| accessIPv6                  |                                      |
| progress                    | 0                                    |
| OS-EXT-STS:power_state      | 0                                    |
| OS-EXT-AZ:availability_zone | nova                                 |
| config_drive                |                                      |
+-----------------------------+--------------------------------------+
```

Nova Boot Server



Virsh on Compute Node

```
openstack@compute2:~/devstack$ virsh list
 Id    Name                           State
----------------------------------------------------
 1     instance-00000001              running
```

Virsh on Compute Node

```
openstack@compute2:~/devstack$ virsh dumpxml instance-00000001 | more
<domain type='kvm' id='1'>
  <name>instance-00000001</name>
  <uuid>dbbeec72-92e2-4285-960e-32d01bdc0b50</uuid>
  <memory unit='KiB'>524288</memory>
  <currentMemory unit='KiB'>524288</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <sysinfo type='smbios'>
    <system>
      <entry name='manufacturer'>OpenStack Foundation</entry>
      <entry name='product'>OpenStack Nova</entry>
      <entry name='version'>2013.1.1</entry>
      <entry name='serial'>de5fd950-eb69-11e1-1000-000000000009</entry>
      <entry name='uuid'>dbbeec72-92e2-4285-960e-32d01bdc0b50</entry>
    </system>
```

Virsh on Compute Node (Network Interface)

```
  <devices>
...skipping lines
    </interface>
      <mac address='fa:16:3e:a2:b4:2e'/>
      <source bridge='qbr40070dfd-59'/>
      <target dev='tap40070dfd-59'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
...skipping lines

</domain>
```

OVS on Compute Node

```
openstack@compute2:~/devstack$ sudo ovs-vsctl show
887df6af-dccf-4b15-a7db-48f15865dcf5
    Bridge br-int
        Port "qvo40070dfd-59"
            tag: 1
            Interface "qvo40070dfd-59”
        Port br-int
            Interface br-int
                type: internal
        Port patch-tun
            Interface patch-tun
                type: patch
                options: {peer=patch-int}
    Bridge br-tun
        Port "gre-2"
            Interface "gre-2"
                type: gre
                options: {in_key=flow, out_key=flow, remote_ip="172.18.31.3"}
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
        Port "gre-1"
            Interface "gre-1"
                type: gre
                options: {in_key=flow, out_key=flow, remote_ip="172.18.31.1"}
        Port br-tun
            Interface br-tun
                type: internal
    ovs_version: "1.9.0"
```

KVM on Compute Node

```
openstack@compute2:~/devstack$ virsh dumpxml instance-00000001 | grep tap

      <target dev='tap40070dfd-59'/>

openstack@compute2:~/devstack$ ps aux | grep tap

1066994 18.6  0.0 3868964 287304 ?      Sl   15:38   0:15 qemu-system-x86_64 -machine accel=kvm:tcg -name instance-00000001 -S
   -M pc-i440fx-1.4 -m 512 -smp 1,sockets=1,cores=1,threads=1 -uuid dbbeec72-92e2-4285-960e-32d01bdc0b50 -smbios
   type=1,manufacturer=OpenStack Foundation,product=OpenStack Nova,version=2013.1.1,serial=de5fd950-
   eb69-11e1-1000-000000000009,uuid=dbbeec72-92e2-4285-960e-32d01bdc0b50 -no-user-config -nodefaults -chardev
   socket,id=charmonitor,path=/var/lib/libvirt/qemu/instance-00000001.monitor,server,nowait -mon
   chardev=charmonitor,id=monitor,mode=control -rtc base=utc,driftfix=slew -no-shutdown -device piix3-usb-uhci,id=usb,bus=pci.
   0,addr=0x1.0x2 -drive file=/opt/stack/data/nova/instances/dbbeec72-92e2-4285-960e-32d01bdc0b50/disk,if=none,id=drive-virtio-
   disk0,format=qcow2,cache=none -device virtio-blk-pci,scsi=off,bus=pci.0,addr=0x4,drive=drive-virtio-disk0,id=virtio-
   disk0,bootindex=1 –netdev tap,fd=24,id=hostnet0,vhost=on,vhostfd=25 -device virtio-net-pci,netdev=hostnet0,id=net0,mac=fa:
   16:3e:a2:b4:2e,bus=pci.0,addr=0x3 -chardev file,id=charserial0,path=/opt/stack/data/nova/instances/
   dbbeec72-92e2-4285-960e-32d01bdc0b50/console.log -device isa-serial,chardev=charserial0,id=serial0 -chardev
   pty,id=charserial1 -device isa-serial,chardev=charserial1,id=serial1 -device usb-tablet,id=input0 -vnc 0.0.0.0:0 -k en-us -
   vga cirrus -device virtio-balloon-pci,id=balloon0,bus=pci.0,addr=0x5
```

IPtables on Compute Node

```
openstack@compute2:~/devstack$ sudo iptables -L

...skipping lines

Chain neutron-openvswi-i40070dfd-5 (1 references)

target     prot opt source               destination

DROP       all  --  anywhere             anywhere             state INVALID

RETURN     all  --  anywhere             anywhere             state RELATED,ESTABLISHED

RETURN     tcp  --  anywhere             anywhere             tcp dpt:ssh

RETURN     tcp  --  anywhere             anywhere             tcp dpt:http

RETURN     icmp --  anywhere             anywhere

RETURN     udp  --  10.2.0.2             anywhere             udp spt:bootps dpt:bootpc

...skipping lines
```

DHCP Server on Control Node

```
openstack@control:~/devstack/context_im_user$ ps aux | grep $NET_ID

nobody   26628  0.0  0.0  28900   964 ?        S    15:34   0:00 dnsmasq --no-hosts --no-resolv --strict-order --bind-
interfaces --interface=tap7d16f078-f6 --except-interface=lo --pid-file=/opt/stack/data/neutron/dhcp/d74c68be-
a785-4a62-9e4f-81e66693fc69/pid --dhcp-hostsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/host --dhcp-
optsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/opts --dhcp-script=/opt/stack/neutron/bin/neutron-
dhcp-agent-dnsmasq-lease-update --leasefile-ro --dhcp-range=set:tag0,10.2.0.0,static,120s --conf-file= --domain=openstacklocal

root     26629  0.0  0.0  28900   452 ?        S    15:34   0:00 dnsmasq --no-hosts --no-resolv --strict-order --bind-
interfaces --interface=tap7d16f078-f6 --except-interface=lo --pid-file=/opt/stack/data/neutron/dhcp/d74c68be-
a785-4a62-9e4f-81e66693fc69/pid --dhcp-hostsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/host --dhcp-
optsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/opts --dhcp-script=/opt/stack/neutron/bin/neutron-
dhcp-agent-dnsmasq-lease-update --leasefile-ro --dhcp-range=set:tag0,10.2.0.0,static,120s --conf-file= --domain=openstacklocal

root     26653  0.0  1.1  93376 22636 ?        S    15:34   0:00 /usr/bin/python /usr/local/bin/neutron-ns-metadata-proxy --
pid_file=/opt/stack/data/neutron/external/pids/d74c68be-a785-4a62-9e4f-81e66693fc69.pid --network_id=d74c68be-
a785-4a62-9e4f-81e66693fc69 --state_path=/opt/stack/data/neutron --metadata_port=80 --debug --verbose

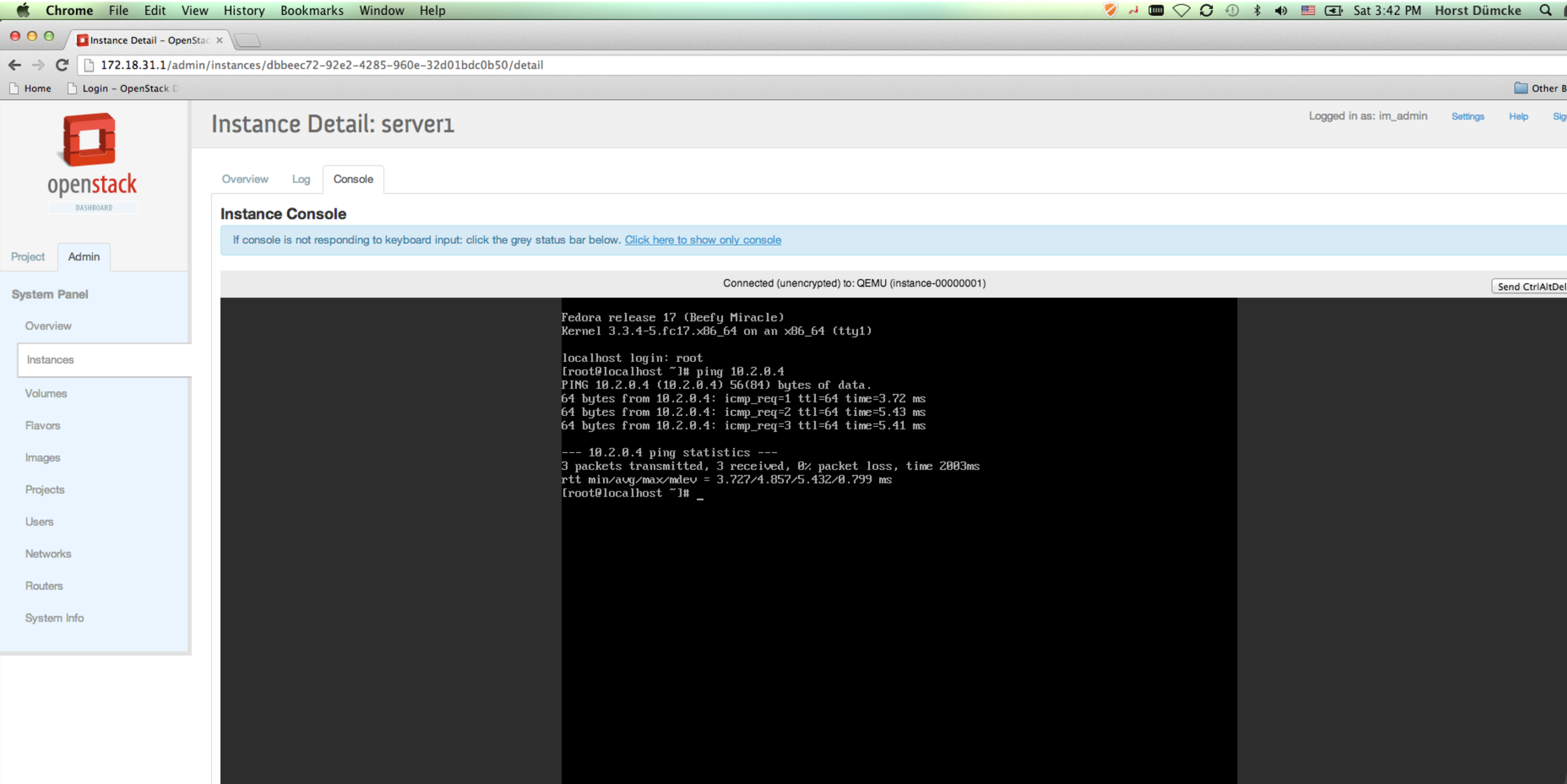
openstack@control:~/devstack/context_im_user$ cat /opt/stack/data/neutron/dhcp/$NET_ID/host

fa:16:3e:a2:b4:2e,10-2-0-3.openstacklocal,10.2.0.3
```

Boot Server 2

```
openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name server2_port $NET_ID
...skipping output
openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep server2_port | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID server2
...skipping output
```

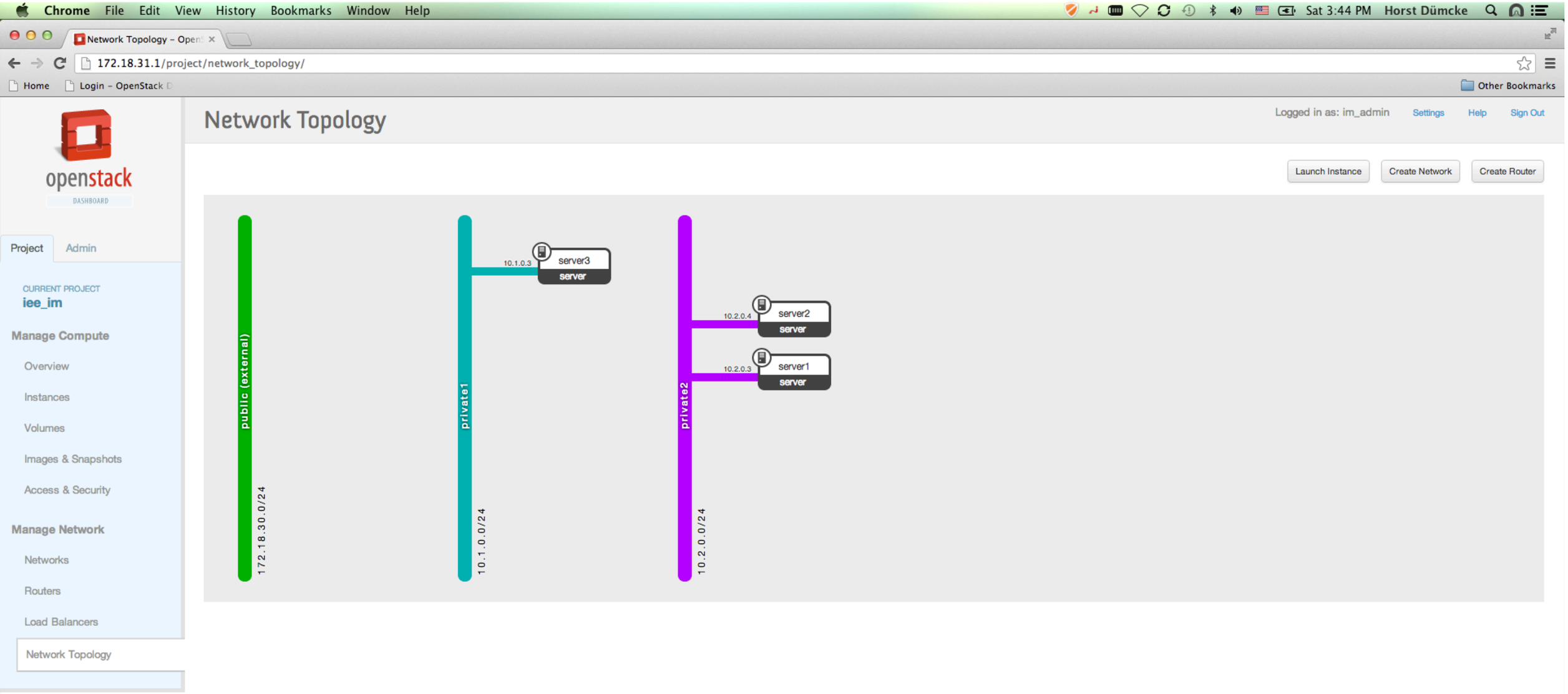
Ping Server 2



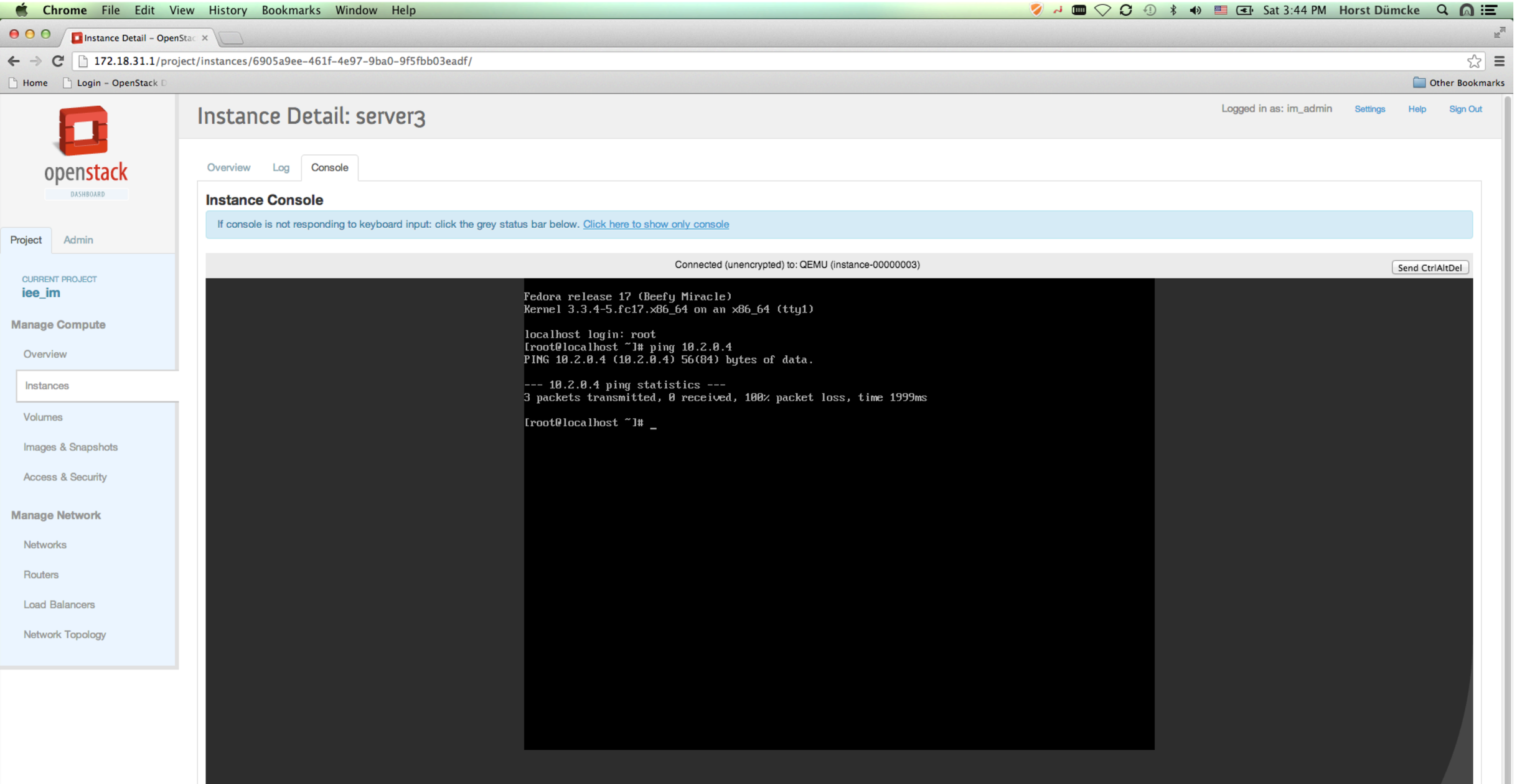
Boot Server 3

```
openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep private1 | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name server3_port $NET_ID
...skipping output
openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep server3_port | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID server3
...skipping output
```

Network Topology



Ping Server 2



Create Router

```
openstack@control:~/devstack/context_im_user$ neutron router-create router1


Created a new router:

+-----------------------+--------------------------------------+
| Field                 | Value                                |
+-----------------------+--------------------------------------+
| admin_state_up        | True                                 |
| external_gateway_info |                                      |
| id                    | 4ae28bf7-2f80-4f3b-be83-c420e0eb19e9 |
| name                  | router1                              |
| status                | ACTIVE                               |
| tenant_id             | e815e6d44c6747308d30adabff1dac99     |
+-----------------------+--------------------------------------+

openstack@control:~/devstack/context_im_user$ SUBNET_ID=`neutron subnet-list | grep 10.2.0.0 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron router-interface-add router1 ${SUBNET_ID}

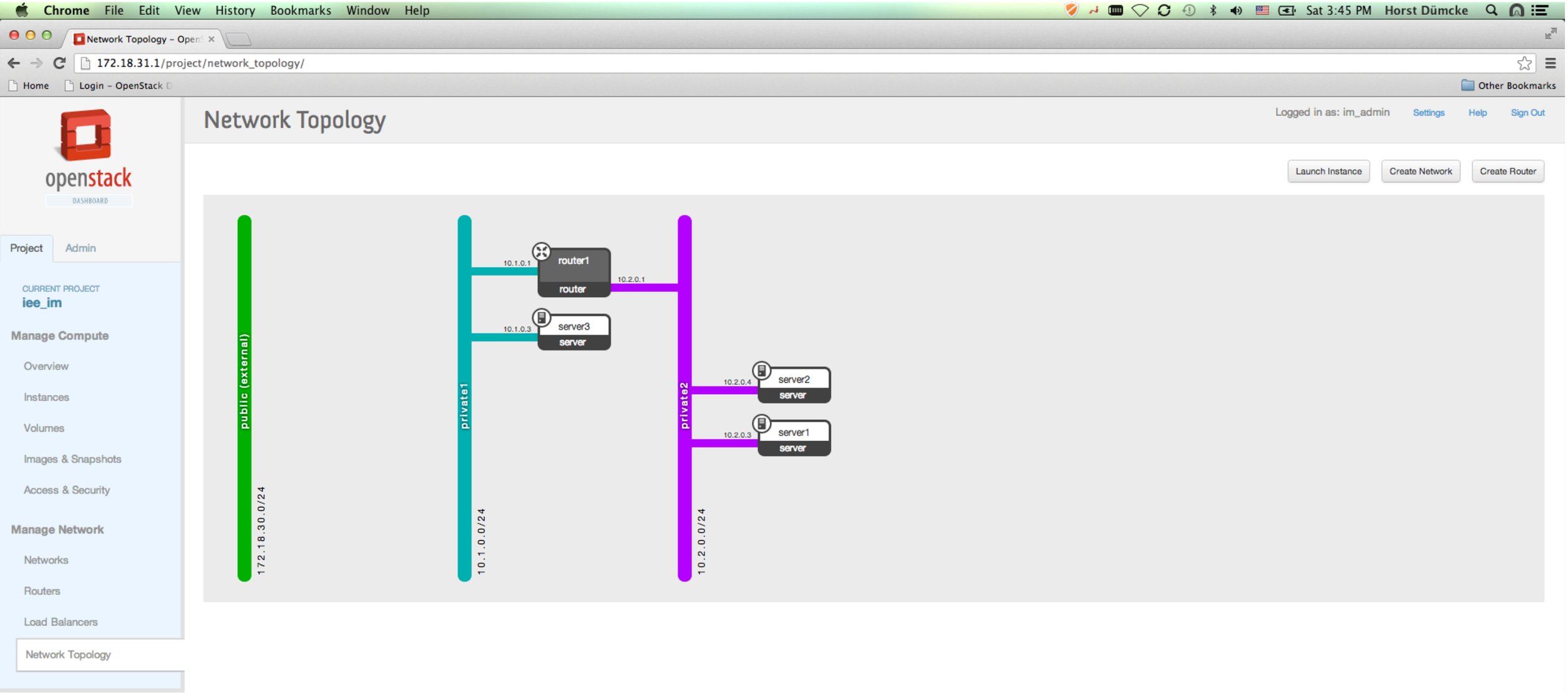
Added interface to router router1

openstack@control:~/devstack/context_im_user$ SUBNET_ID=`neutron subnet-list | grep 10.1.0.0 | awk -F' ' '{ print $2 }'`

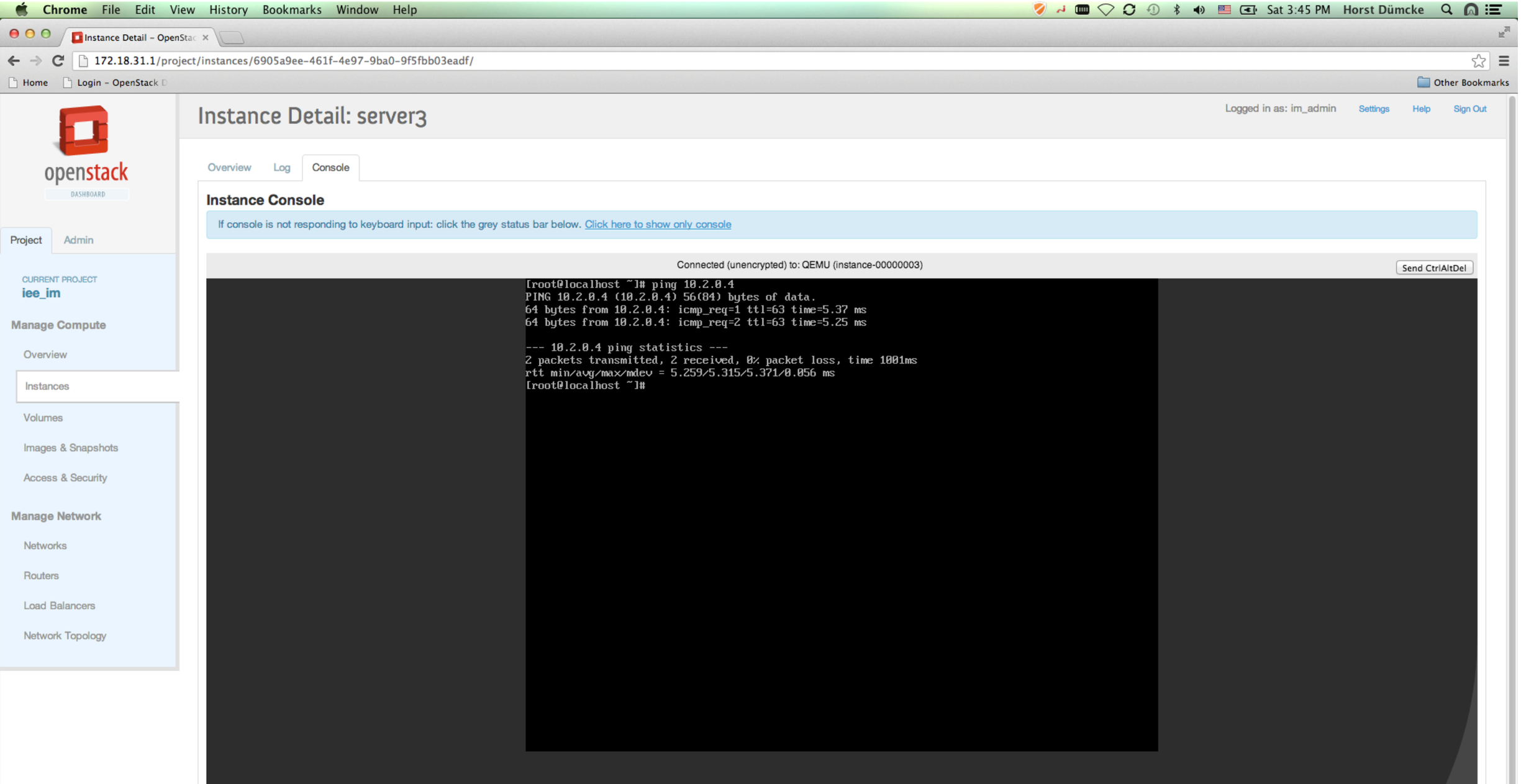
openstack@control:~/devstack/context_im_user$ neutron router-interface-add router1 ${SUBNET_ID}

Added interface to router router1
```

Network Topology



Ping Server 2



Add Gateway

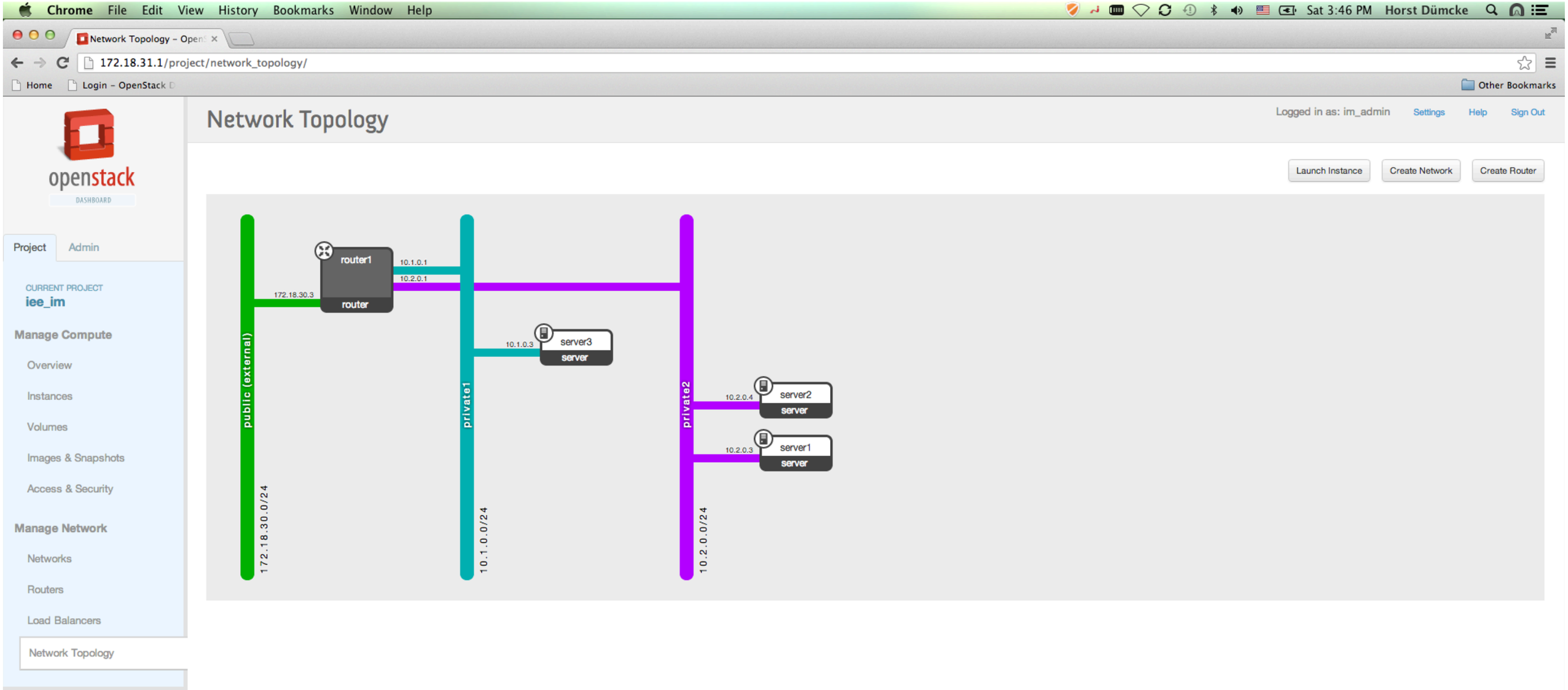
```
openstack@control:~/devstack/context_im_user$ neutron router-gateway-set router1 public

Set gateway for router router1

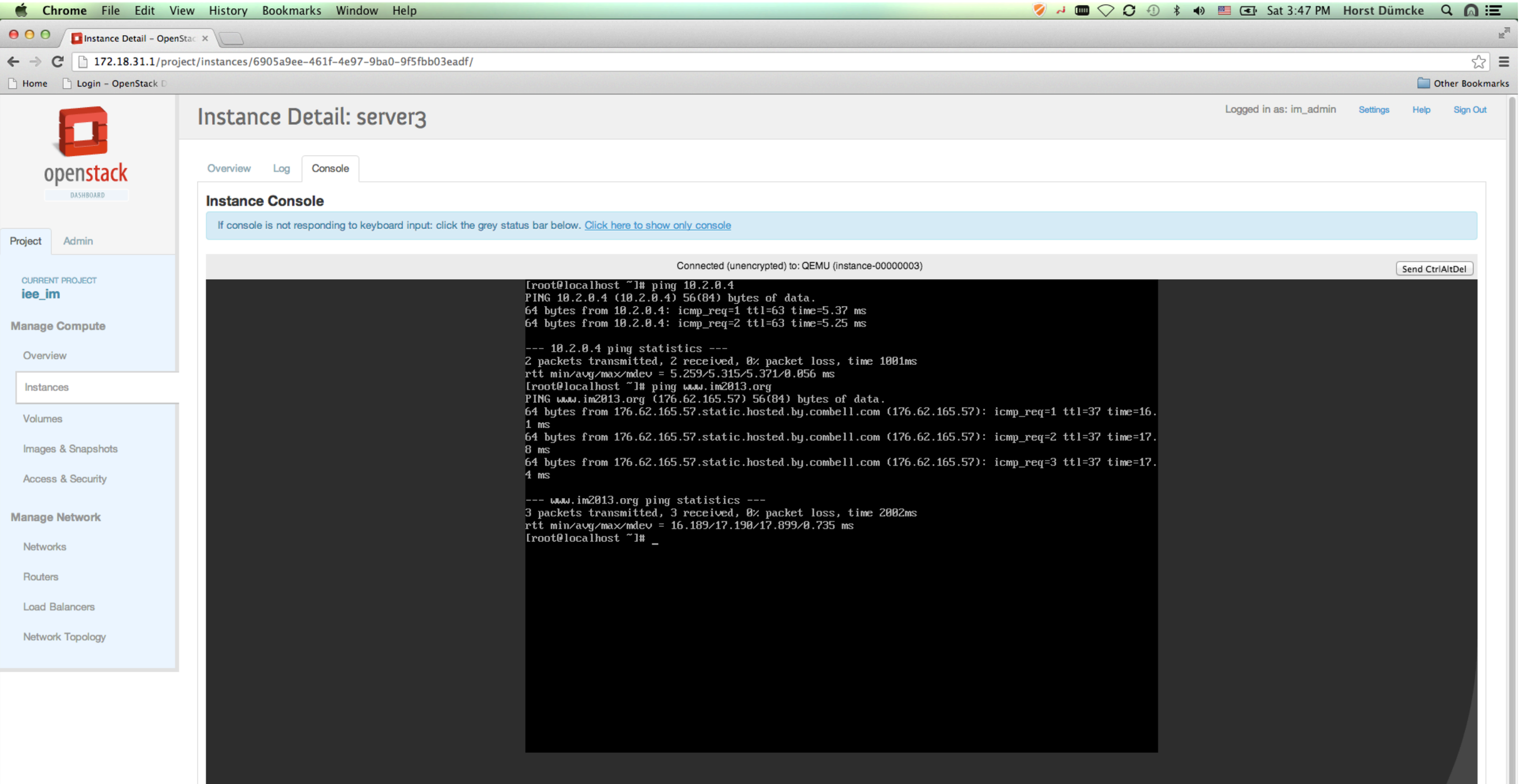
openstack@control:~/devstack/context_im_user$ neutron router-list

+--------------------------------------+---------+-----------------------------------------------------------+
| id                                   | name    | external_gateway_info                                     |
+--------------------------------------+---------+-----------------------------------------------------------+
| 4ae28bf7-2f80-4f3b-be83-c420e0eb19e9 | router1 | {"network_id": "036460df-90d3-45b2-b063-1ece8d0bd061"} |
+--------------------------------------+---------+-----------------------------------------------------------+
```

Network Topology



Ping External Network



Router Network Namespace

```
openstack@control:~/devstack/context_im_user$ ROUTER_ID=`neutron router-list | grep router1 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ sudo ip netns exec qrouter-$ROUTER_ID ip addr list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
16: qr-1b2d50ea-93: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:76:06:93 brd ff:ff:ff:ff:ff:ff
    inet 10.2.0.1/24 brd 10.2.0.255 scope global qr-1b2d50ea-93
    inet6 fe80::f816:3eff:fe76:693/64 scope link
       valid_lft forever preferred_lft forever
17: qr-026e163f-f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:7b:6a:64 brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.1/24 brd 10.1.0.255 scope global qr-026e163f-f6
    inet6 fe80::f816:3eff:fe7b:6a64/64 scope link
       valid_lft forever preferred_lft forever
18: qg-e4106235-97: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:9a:28:58 brd ff:ff:ff:ff:ff:ff
    inet 172.18.30.3/24 brd 172.18.30.255 scope global qg-e4106235-97
    inet 172.18.30.4/32 brd 172.18.30.4 scope global qg-e4106235-97
    inet 172.18.30.5/32 brd 172.18.30.5 scope global qg-e4106235-97
    inet 172.18.30.6/32 brd 172.18.30.6 scope global qg-e4106235-97
    inet6 fe80::f816:3eff:fe9a:2858/64 scope link
       valid_lft forever preferred_lft forever

openstack@control:~/devstack/context_im_user$ sudo ip netns exec qrouter-$ROUTER_ID ip route list
default via 172.18.30.1 dev qg-e4106235-97
10.1.0.0/24 dev qr-026e163f-f6  proto kernel  scope link  src 10.1.0.1
10.2.0.0/24 dev qr-1b2d50ea-93  proto kernel  scope link  src 10.2.0.1
172.18.30.0/24 dev qg-e4106235-97  proto kernel  scope link  src 172.18.30.3
```

OVS on Control Node

```
openstack@control:~/devstack/context_im_user$ sudo ovs-vsctl show | more
9cd9bbc0-c130-437c-bee7-0aaa1dfa98c3
    Bridge br-ex
        Port "eth1"
            Interface "eth1"
        Port br-ex
            Interface br-ex
                type: internal
        Port "qg-e4106235-97"
            Interface "qg-e4106235-97"
                type: internal
More…
```

OVS on Control Node

```
Bridge br-int
...skipping lines
        Port "qr-026e163f-f6"
            tag: 2
            Interface "qr-026e163f-f6"
                type: internal
...skipping lines
        Port "qr-1b2d50ea-93"
            tag: 1
            Interface "qr-1b2d50ea-93"
                type: internal
```

Add Floating IP Address

```
openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep public | awk -F' ' '{ print \$2 }'`

openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep 10.1.0.3 | awk -F' ' '{ print \$2 }'`

openstack@control:~/devstack/context_im_user$ neutron floatingip-create --port_id=$PORT_ID $NET_ID

Created a new floatingip:

+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    | 10.1.0.3                             |
| floating_ip_address | 172.18.30.4                          |
| floating_network_id | 036460df-90d3-45b2-b063-1ece8d0bd061 |
| id                  | fa56af80-2718-40ef-9d8d-b465f5676a8a |
| port_id             | 5f79b923-d5f0-4f2b-8af9-f2538fb5d480 |
| router_id           | 4ae28bf7-2f80-4f3b-be83-c420e0eb19e9 |
| tenant_id           | e815e6d44c6747308d30adabff1dac99     |
+---------------------+--------------------------------------+
```

Ping Server from External Network

```
[root@Cobbler ~]# ping 172.18.30.4
PING 172.18.30.4 (172.18.30.4) 56(84) bytes of data.
64 bytes from 172.18.30.4: icmp_req=2 ttl=61 time=2.11 ms
64 bytes from 172.18.30.4: icmp_req=3 ttl=61 time=4.36 ms
64 bytes from 172.18.30.4: icmp_req=4 ttl=61 time=5.50 ms
```

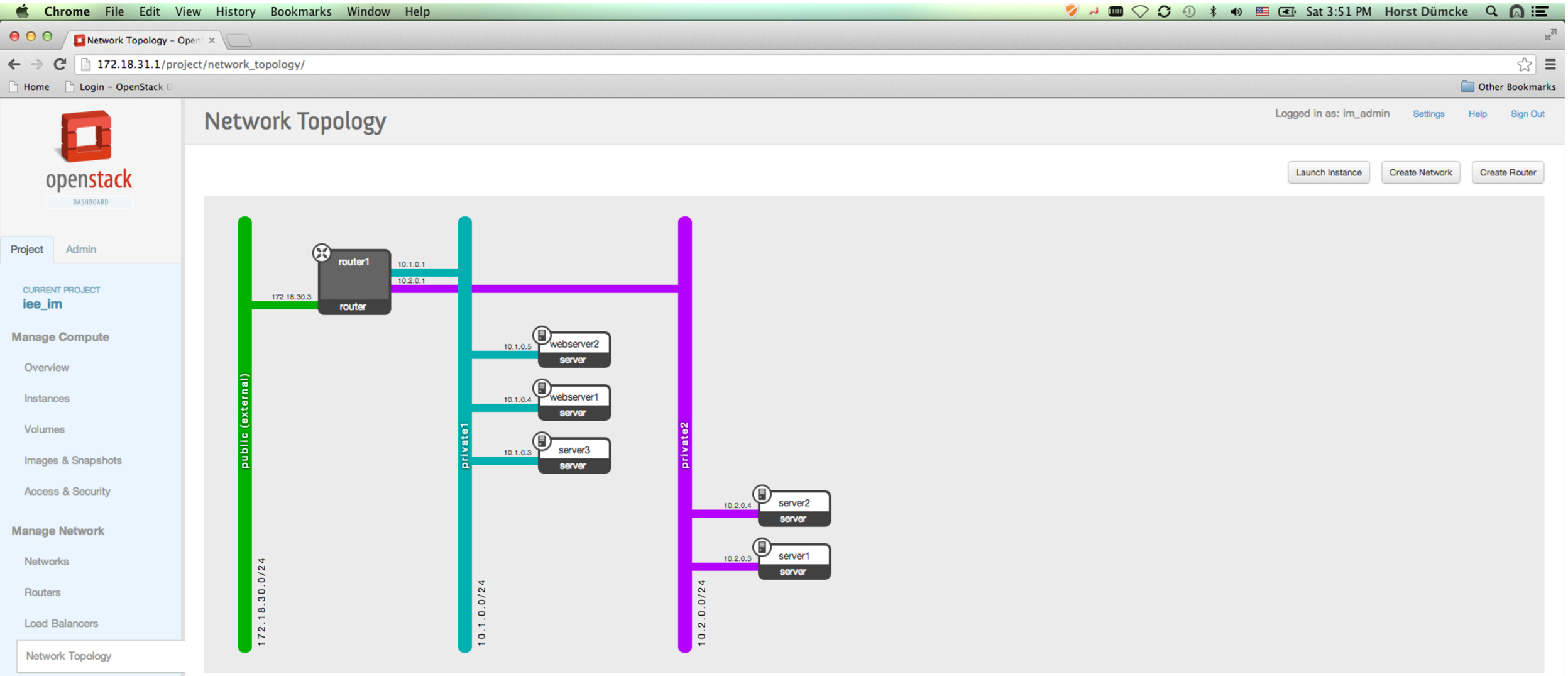
Iptables on Control Node

```
openstack@control:~/devstack/context_im_user$ NETNS=`ip netns list | grep qrouter`
openstack@control:~/devstack/context_im_user$ sudo ip netns exec $NETNS iptables -L -t nat  | grep SNAT
SNAT       all  --  10.1.0.3             anywhere             to:172.18.30.4
SNAT       all  --  10.2.0.0/24          anywhere             to:172.18.30.3
SNAT       all  --  10.1.0.0/24          anywhere             to:172.18.30.3
openstack@control:~/devstack/context_im_user$ sudo ip netns exec $NETNS iptables -L -t nat  | grep DNAT
DNAT       all  --  anywhere             172.18.30.4          to:10.1.0.3
ACCEPT     all  --  anywhere             anywhere             ! ctstate DNAT
DNAT       all  --  anywhere             172.18.30.4          to:10.1.0.3
```

Boot Webservers

```
openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep private1 | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name webserver1_port $NET_ID
...skipping output
openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep webserver1_port | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID webserver1
...skipping output
openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name webserver2_port $NET_ID
...skipping output
openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep webserver1_port | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID webserver2
...skipping output
```

Network Topology



Create LB Pool

```
openstack@control:~/devstack/context_im_user$ SUBNET_ID=`neutron subnet-list | grep 10.1.0.0 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron lb-pool-create --lb-method ROUND_ROBIN --name im_lbpool --protocol HTTP
--subnet-id $SUBNET_ID

Created a new pool:

+------------------+--------------------------------------+
| Field            | Value                                |
+------------------+--------------------------------------+
| admin_state_up   | True                                 |
| description      |                                      |
| health_monitors  |                                      |
| id               | 64ceba3f-326c-4fa8-8d3b-55439bde4864 |
| lb_method        | ROUND_ROBIN                          |
| members          |                                      |
| name             | im_lbpool                            |
| protocol         | HTTP                                 |
| status           | PENDING_CREATE                       |
| subnet_id        | 9c3e6b43-da73-4841-972a-18b57186ab81 |
| tenant_id        | e815e6d44c6747308d30adabff1dac99     |
| vip_id           |                                      |
+------------------+--------------------------------------+
```

Add Server to LB Pool

```
openstack@control:~/devstack/context_im_user$ SERVER_ID=`nova list | grep webserver1 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ NW_ADDR=`nova show $SERVER_ID | grep network | awk -F' ' '{ print $5 }'`

openstack@control:~/devstack/context_im_user$ neutron lb-member-create --address $NW_ADDR --protocol-port 80 im_lbpool

Created a new member:

+----------------+--------------------------------------+
| Field          | Value                                |
+----------------+--------------------------------------+
| address        | 10.1.0.4                             |
| admin_state_up | True                                 |
| id             | d61f2fa9-203e-472d-bc8f-f009d085fe98 |
| pool_id        | 64ceba3f-326c-4fa8-8d3b-55439bde4864 |
| protocol_port  | 80                                   |
| status         | PENDING_CREATE                       |
| tenant_id      | e815e6d44c6747308d30adabff1dac99     |
| weight         | 1                                    |
+----------------+--------------------------------------+

openstack@control:~/devstack/context_im_user$ SERVER_ID=`nova list | grep webserver2 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ NW_ADDR=`nova show $SERVER_ID | grep network | awk -F' ' '{ print $5 }'`

openstack@control:~/devstack/context_im_user$ neutron lb-member-create --address $NW_ADDR --protocol-port 80 im_lbpool

...skipping output
```

Create Health Monitor

```
openstack@control:~/devstack/context_im_user$ neutron lb-healthmonitor-create --delay 3 --type HTTP --max-retries 3 --timeout 3

Created a new health_monitor:

+----------------+--------------------------------------+
| Field          | Value                                |
+----------------+--------------------------------------+
| admin_state_up | True                                 |
| delay          | 3                                    |
| expected_codes | 200                                  |
| http_method    | GET                                  |
| id             | 0a2fd417-63fd-4d2f-bb84-c2c32bca0fde |
| max_retries    | 3                                    |
| status         | PENDING_CREATE                       |
| tenant_id      | e815e6d44c6747308d30adabff1dac99     |
| timeout        | 3                                    |
| type           | HTTP                                 |
| url_path       | /                                    |
+----------------+--------------------------------------+

openstack@control:~/devstack/context_im_user$ HM_ID=`neutron lb-healthmonitor-list | grep HTTP | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron lb-healthmonitor-associate $HM_ID im_lbpool

Associated health monitor 0a2fd417-63fd-4d2f-bb84-c2c32bca0fde
```

Create VIP

```
openstack@control:~/devstack/context_im_user$ SUBNET_ID=`neutron subnet-list | grep 10.1.0.0 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ POOL_ID=`neutron lb-pool-list | grep im_lbpool | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron lb-vip-create --name im_vip --protocol-port 80 --protocol HTTP --subnet-
id $SUBNET_ID im_lbpool

Created a new vip:

+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| address           | 10.1.0.6                             |
| admin_state_up    | True                                 |
| connection_limit  | -1                                   |
| description       |                                      |
| id                | 53594387-ec6a-4246-8b4d-f03c7a7bc5a3 |
| name              | im_vip                               |
| pool_id           | 64ceba3f-326c-4fa8-8d3b-55439bde4864 |
| port_id           | d5166ca5-eca7-4686-91ba-8cd45a01ad36 |
| protocol          | HTTP                                 |
| protocol_port     | 80                                   |
| status            | PENDING_CREATE                       |
| subnet_id         | 9c3e6b43-da73-4841-972a-18b57186ab81 |
| tenant_id         | e815e6d44c6747308d30adabff1dac99     |
+-------------------+--------------------------------------+
```

Show HAproxy

```
openstack@control:~/devstack/context_im_user$ ps aux | grep haproxy

nobody   16237  0.0  0.0  20412   988 ?        Ss   15:52   0:00 haproxy -f /opt/stack/data/neutron/lbaas/
64ceba3f-326c-4fa8-8d3b-55439bde4864/conf -p /opt/stack/data/neutron/lbaas/64ceba3f-326c-4fa8-8d3b-55439bde4864/pid
```

Show Haproxy Config File

```
openstack@control:~/devstack/context_im_user$ POOL_ID=`neutron lb-pool-list | grep im_lbpool | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ cat /opt/stack/data/neutron/lbaas/$POOL_ID/conf

global
        daemon
        user nobody
        group nogroup
        log /dev/log local0
        log /dev/log local1 notice
        stats socket /opt/stack/data/neutron/lbaas/64ceba3f-326c-4fa8-8d3b-55439bde4864/sock mode 0666 level user
defaults
        log global
        retries 3
        option redispatch
        timeout connect 5000
        timeout client 50000
        timeout server 50000
frontend 53594387-ec6a-4246-8b4d-f03c7a7bc5a3
        option tcplog
        bind 10.1.0.6:80
        mode http
        default_backend 64ceba3f-326c-4fa8-8d3b-55439bde4864
        option forwardfor
backend 64ceba3f-326c-4fa8-8d3b-55439bde4864
        mode http
        balance roundrobin
        option forwardfor
        timeout check 3s
        option httpchk GET /
        http-check expect rstatus 200
        server cc63a8dc-42f7-4451-a6fd-4d1bea48ba07 10.1.0.5:80 weight 1 check inter 3s fall 3
        server d61f2fa9-203e-472d-bc8f-f009d085fe98 10.1.0.4:80 weight 1 check inter 3s fall 3
```

Show Haproxy Network Namespace

```
openstack@control:~/devstack/context_im_user$ ip netns list

qlbaas-64ceba3f-326c-4fa8-8d3b-55439bde4864
qrouter-4ae28bf7-2f80-4f3b-be83-c420e0eb19e9
qdhcp-036460df-90d3-45b2-b063-1ece8d0bd061
qdhcp-5fdb3fc3-2caf-4c06-87e3-0c7b1f4ce6a3
qdhcp-d74c68be-a785-4a62-9e4f-81e66693fc69

openstack@control:~/devstack/context_im_user$ sudo ip netns exec qlbaas-$POOL_ID ip addr list

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
19: tapd5166ca5-ec: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:bb:6b:0a brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.6/24 brd 10.1.0.255 scope global tapd5166ca5-ec
    inet6 fe80::f816:3eff:febb:6b0a/64 scope link
       valid_lft forever preferred_lft forever
```

Map VIP to Floating IP Address

```
openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep public | awk -F' ' '{ print \$2 }'`

openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep 10.1.0.6 | awk -F' ' '{ print \$2 }'`

openstack@control:~/devstack/context_im_user$ neutron floatingip-create --port_id=$PORT_ID $NET_ID

Created a new floatingip:

+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    | 10.1.0.6                             |
| floating_ip_address | 172.18.30.5                          |
| floating_network_id | 036460df-90d3-45b2-b063-1ece8d0bd061 |
| id                  | f7f4298e-ff47-4134-af26-ce6aae2d8a16 |
| port_id             | d5166ca5-eca7-4686-91ba-8cd45a01ad36 |
| router_id           | 4ae28bf7-2f80-4f3b-be83-c420e0eb19e9 |
| tenant_id           | e815e6d44c6747308d30adabff1dac99     |
+---------------------+--------------------------------------+
```

Prepare Webservers

```
yum –y install httpd

iptables –F

echo ‘this is server [1,2]’ > /var/www/html/index.html

service http start
```

Test VIP from External Network

```
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 1
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 2
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 1
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 2
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 1
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 2
```

Meta Data Insertion (Generate Key Pair)

```
openstack@control:~/devstack/context_im_user$ ssh-keygen -f /tmp/id_rsa -t rsa -N ''
Generating public/private rsa key pair.
Your identification has been saved in /tmp/id_rsa.
Your public key has been saved in /tmp/id_rsa.pub.
The key fingerprint is:
47:62:e9:40:1d:f5:77:d2:bb:84:ee:fe:52:0e:83:a0 openstack@control
The key's randomart image is:
+--[ RSA 2048]----+
|       ...o.     |
|      .  .. .   . |
|       . + . . o o|
|        + +   ..o.|
|         S o .. o |
|        E . ..o...|
|             .=. |
|            .. . |
|            .oo. |
+-----------------+
openstack@control:~/devstack/context_im_user$ nova keypair-add --pub_key /tmp/id_rsa.pub noms_key
```

Meta Data Insertion (Boot Server)

```
openstack@control:~/devstack/context_im_user$ IMAGE_ID=`glance image-list | grep precise-amd64 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep private1 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name server4_port $NET_ID

...skipping output

openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep server4_port | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID  --key_name
noms_key server4  --meta greeting='Hi from Krakow'

...skipping output

openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep public | awk -F' ' '{ print \$2 }'`

openstack@control:~/devstack/context_im_user$ neutron floatingip-create --port_id=$PORT_ID $NET_ID

Created a new floatingip:

+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    | 10.1.0.7                             |
| floating_ip_address | 172.18.30.6                          |

...skipping output
```

SSH to Server from Outside Network

```
[root@Cobbler ~]# scp 172.18.31.1:/tmp/id_rsa .

The authenticity of host '172.18.31.1 (172.18.31.1)' can't be established.

RSA key fingerprint is f8:49:5d:4d:3b:04:96:70:1e:e0:e6:5b:d8:17:f2:37.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '172.18.31.1' (RSA) to the list of known hosts.


root@172.18.31.1's password:

id_rsa
100% 1679     1.6KB/s   00:00


[root@Cobbler ~]# ssh -i id_rsa -l ubuntu 172.18.30.6



...skipping output

Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-34-virtual x86_64)

...skipping output


ubuntu@server4:~$
```

Browse Meta Data from VM

```
ubuntu@server4:~$ curl http://169.254.169.254/

1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
latest

ubuntu@server4:~$ curl http://169.254.169.254/latest/meta-data/public-keys/

0=noms_key
```

Browse Meta Data from VM

```
ubuntu@server4:~$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDXRfXP3i2MoCyLrTKNmbObAGmSIurBHvJKzKLl5rju9hld8OOgrVHJexvatGlzhRFs30InY9xlw/
xcqeN2bDkr6w5Dh06Z41wA8rcQeqKDmIRskrABcalFtQ03vV3p+PgDTZEOmAFW4xRsqDrsqyrzb+IItnHZV/rH0Es3daal
+DIuaOQOTya9wNXVwJJdypKaGeSRbqKnsv8KgJA7riWZuLNYxqEgaAt4FugmsmvLoRX95d9bjAw/
JnEQdHe0lCKKfbIwjDYBAx6ctSjuPcGXR1TNwI5hPGnV06r3sU8adGIlRD4Ow64sbty35V0OtZXezZcwW1XwOP4v4WnV0noJ openstack@control

ubuntu@server4:~$ cat .ssh/authorized_keys

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDXRfXP3i2MoCyLrTKNmbObAGmSIurBHvJKzKLl5rju9hld8OOgrVHJexvatGlzhRFs30InY9xlw/
xcqeN2bDkr6w5Dh06Z41wA8rcQeqKDmIRskrABcalFtQ03vV3p+PgDTZEOmAFW4xRsqDrsqyrzb+IItnHZV/rH0Es3daal
+DIuaOQOTya9wNXVwJJdypKaGeSRbqKnsv8KgJA7riWZuLNYxqEgaAt4FugmsmvLoRX95d9bjAw/
JnEQdHe0lCKKfbIwjDYBAx6ctSjuPcGXR1TNwI5hPGnV06r3sU8adGIlRD4Ow64sbty35V0OtZXezZcwW1XwOP4v4WnV0noJ openstack@control

ubuntu@server4:~$ dpkg --get-selections | grep -i cloud

cloud-init                                      install
cloud-initramfs-growroot                        install
cloud-initramfs-rescuevol                       install
cloud-utils                                     install
```

Browse Meta Data from VM

```
ubuntu@server4:~$ curl http://169.254.169.254/openstack/latest/meta_data.json

{"random_seed": "AmZkWggOH5eKAPfZUbiZPCx/Z02jYXNSY1e0bbmv4rmFTfeBXK3zPec/FehD5T5q8rM/
hPPChV5rGTI00rMpLH5q1hHyrrlqOactWiNBz7283ZSN5WCNHr6W01mfBsK8m4UrKTTZLw3jmmK+bPmdlRrwBkJukJ9yDSohz8nw8XbU+ER
+dwDLTrxAg729tOxmPiWS59f+xruqrdPvjHfJjlKXktQaT34pDLPMp0FKTzxOaXTm3kfBmRORZ2pBYldDxPzBoF6lQDKRrTuBn/
3fR5gREWkTADAceu82k8tPRT7gNDz3uc1S/QmU9DNht40WgQInV3A9pj6EnO/0lsjGggOIzT3PnGCGZRt+qrASctd9bD6vbiJgS4mwta3NLxciylnP2CEn4xrG/LS/
1FND1rXRJizV49GleoljaMxdWPcOgU51WcmKiFUvOvEoaHuNnc8qadFSEc53rAnINXAhWMMh
+n3S8pxBEMr8k99WNKkDdNlbefubHFMdkKT01rQ7VjwNTIpmAikvgQHG3IgJVn7CkhtVbD40E8TpTLUEdYhd55G4ca1pTJAuRdr6+dOQfN1QSv2Y+lzO6cvao
+YjU9Qi9hGTqMriZukw9j1/9cjbDOUfrHbQKjIm6H4Uql8HpKw8lfsM4fhYzIMMVDaSeiR2AwOZT2EII8y07U2K4Bk=", "uuid": "2eeb7abd-bcb0-48ce-9cdb-
c2f295b3d9bb", "availability_zone": "nova", "hostname": "server4.novalocal", "launch_index": 0, "meta": {"greeting": "Hi from
Krakow"}, "public_keys": {"noms_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDXRfXP3i2MoCyLrTKNmbObAGmSIurBHvJKzKLl5rju9hld8OOgrVHJexvatGlzhRFs30InY9xlw/
xcqeN2bDkr6w5Dh06Z41wA8rcQeqKDmIRskrABcalFtQ03vV3p+PgDTZEOmAFW4xRsqDrsqyrzb+IItnHZV/rH0Es3daal
+DIuaOQOTya9wNXVwJJdypKaGeSRbqKnsv8KgJA7riWZuLNYxqEgaAt4FugmsmvLoRX95d9bjAw/
JnEQdHe0lCKKfbIwjDYBAx6ctSjuPcGXR1TNwI5hPGnV06r3sU8adGIlRD4Ow64sbty35V0OtZXezZcwW1XwOP4v4WnV0noJ openstack@control\n"}, "name":
"server4"}
```

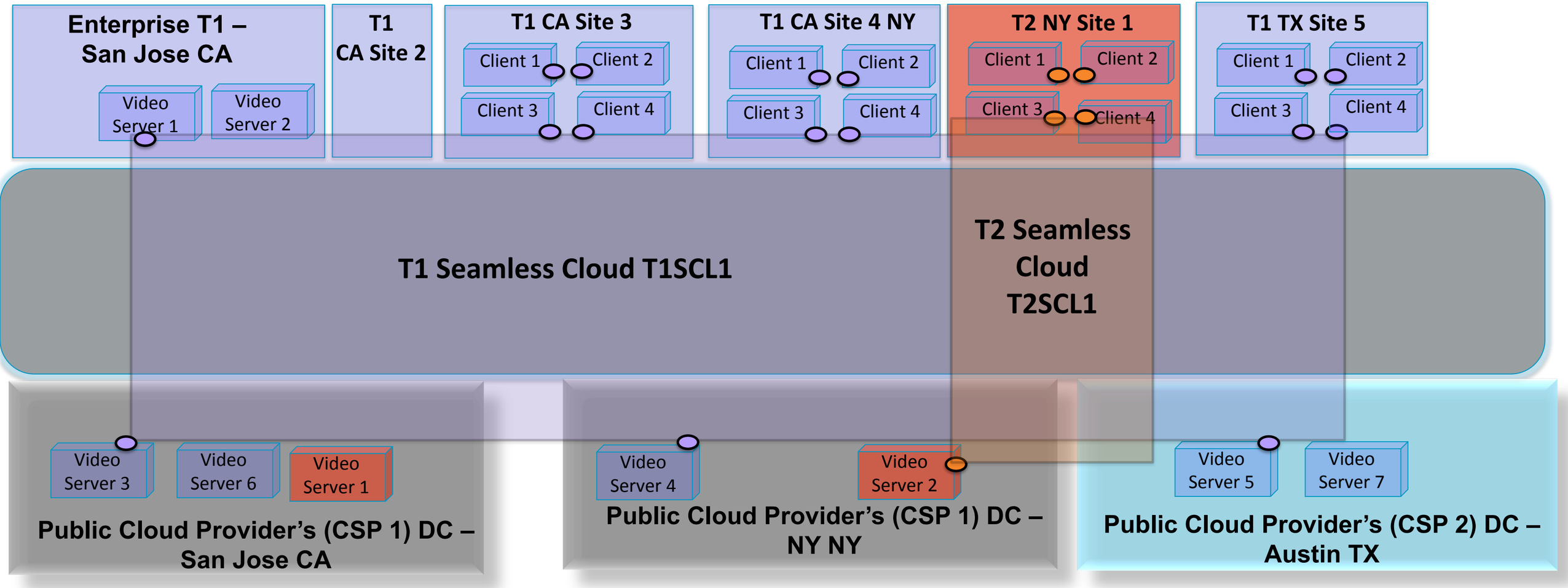
Cloud-init

- Software package that handles early initialization of a cloud instance.
- Behavior can be configured via user-data
  - Nova boot –user-data
- Some of the things it configures are:
  - setting a default locale
  - setting hostname
  - generate ssh private keys
  - adding ssh keys to user's .ssh/authorized_keys so they can log in
  - setting up ephemeral mount points

Virtual Clouds over Multiple Clouds: Seamless Virtual Clouds

Distributed Media Application Seamless Cloud

- A Cloud with resources spanning multiple sites and multiple Clouds
- A Tenant sees it as one Seamless Cloud → As one Subnet even though widely distributed over multiple network segments and Clouds
- Seamless is provided by employing and programming various features in network



Distributed Media Application Seamless Cloud

Demo Video

Q&A

- Q&A

Practical Cloud Deployment and Management with Openstack and Programmable Networking

Masum Z. Hasan, Horst Dümcke

IEEE/IFIP Network Operations and Management Symposium, NOMS 2014,

5-9 May 2014

Krakow, Poland

About the Speakers

Masum Z. Hasan, PhD
Technology Director at the Cloud CTO office
masum@cisco.com

Horst Dümcke
Founder and President, www.marlysys.com
hdumcke@marlysys.com

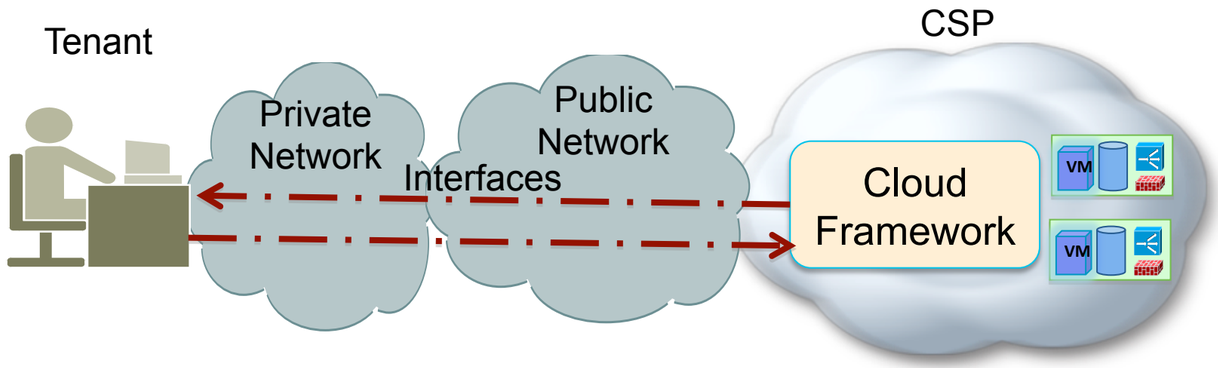
Agenda

- Introduction to Core Cloud Concepts
- Cloud Networking
  - Network Virtualization
  - End-to-end Multitenant Isolation
  - SDN / Programmable Networking
    - OpenFlow
    - Cisco OnePK
    - Open Daylight
- Cloud Framework Openstack
  - Framework introduction
  - Use cases and labs/demo
- Virtual Clouds over Multiple Clouds: Seamless Cloud with Openstack and Programmable Networking

Introduction to Core Cloud Concepts

Definition – Basic Terms

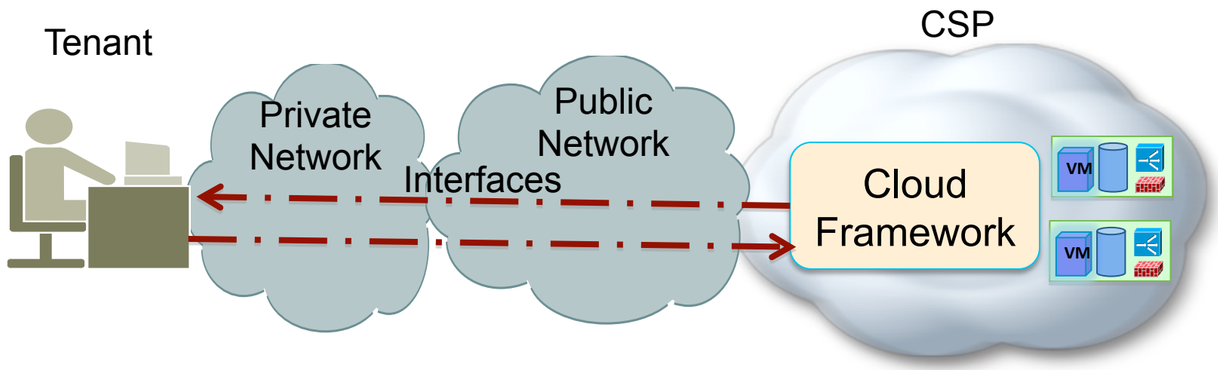
- Cloud Users or Tenants or Cloud Service Consumers: consumes resources from Cloud
  - Enterprises
  - End Consumers
  - Tenant Admins
  - Tenant end-users
- CSP: Cloud Service Provider: Offers Cloud services



- Cloud Resources: Compute, Storage, Network
- CRUD (Create / Read / Update / Delete) of resources
- API (Application Program Interfaces)
  - Includes Tenant facing Service Interfaces
- Cloud Framework: A (software) system, can be Distributed, that controls all aspects of a Cloud deployment
  - Managed by CSP Admins

Definition – Basic Terms

- Network: Without network there is no Cloud
  - From Tenant perspective Cloud is somewhere there over the network
  - Multiple segments of networks, but at a high level Cloud accessed over
    - Private Network: End Consumer home network or Enterprise intranet
    - Public Network: Network SP Private MAN/WAN and/or Internet
- Elements of network can themselves be Cloud resources



Definition – Basic Terms - Distinguishing Features of Cloud: OREMAP

- On-demand CRUD of resources
  - Anytime tenants want resources, it should get it
- Rapid CRUD of resources
  - Requested resources should be CRUD by CSP in seconds to minutes
- Elastic at any scale
  - Tenants should be able to Grow / shrink resource count / usage at any scale
- Multi-tenancy
  - CSP should support sharing of physical resources by totally separate organizations / entities
  - Tenants should be isolated from each other
- Abstractions
  - Need to support multiple and complex technologies for same resource, but users should be shielded from the diversity and complexity
  - Cloud facilitates outsourcing of IT functionalities → IT admins not in full control of resources / infrastructure anymore (specially in case of public / hybrid/ community / seamless Cloud, rather the CSP → provide abstracted view of resources / infrastructure
  - Multiple users, multiple systems → different levels of visibility and access
- Pay-per-use
  - Tenants are billed and pay per resource usage

Core Cloud Service Types

- Cloud resources are accessed via services
- Depending on resource types there can various types of services
- Core service types
  - IaaS: Infrastructure as a Service
    - Main focus of this tutorial
  - PaaS: Platform as a Service
  - SaaS: Software as a Service

IaaS

- Infrastructure elements, logical and physical, are offered as on-demand resources
- Typical Infrastructure resources
  - Virtual Computer or VM
  - Virtual storage or storage objects
  - Virtual Network elements or components or functions

- IaaS CSP, example:
  - Amazon EC2, S3
  - Verizon Terremark
  - RackSpace
  - Microsoft Azure
- Infrastructure vendors (such as Cisco) provides infrastructure, frameworks and systems used by CSPs to deploy IaaS
- Open source Cloud Framework OpenStack with its various controllers facilitates deployment of IaaS

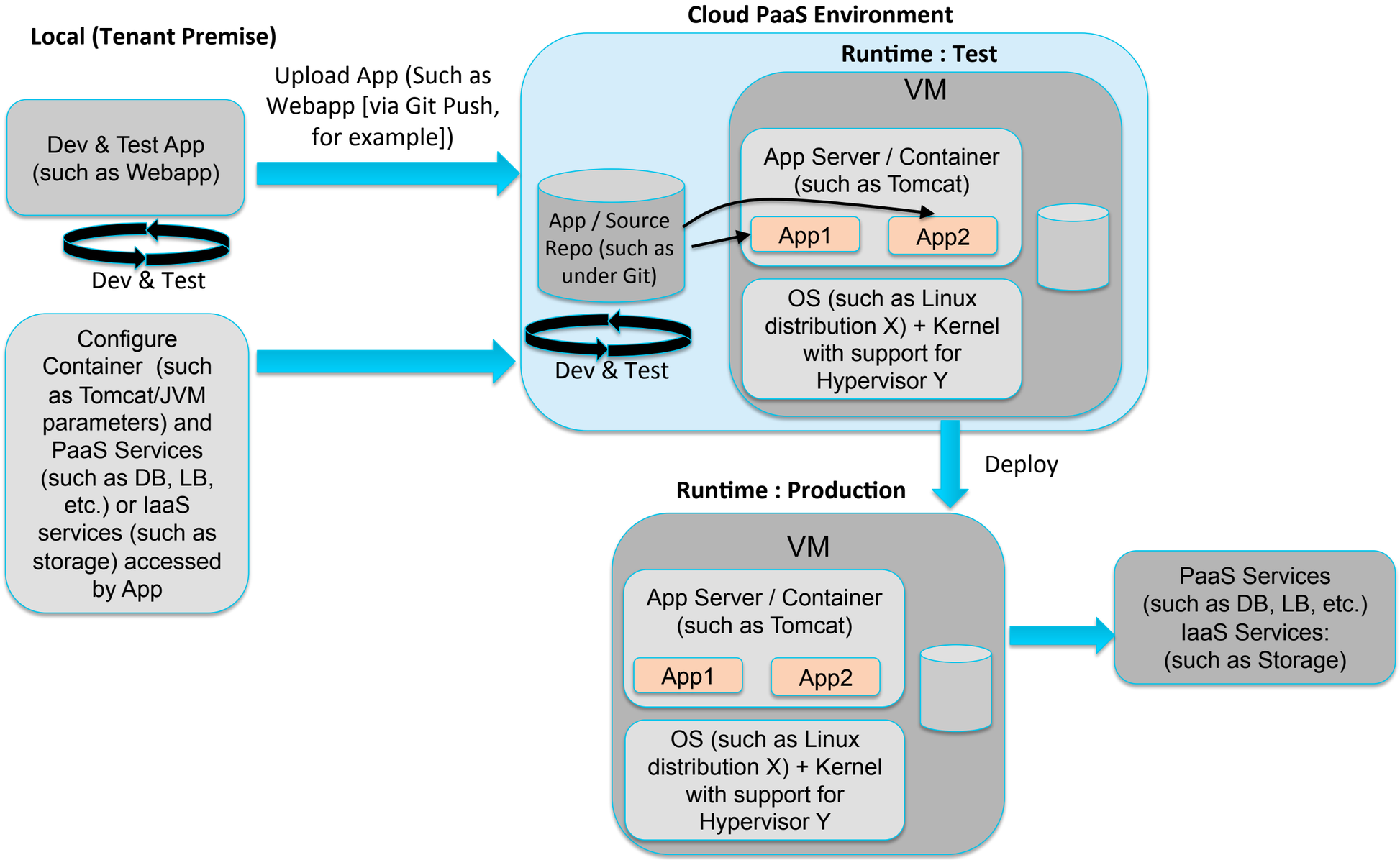
PaaS

- Software platforms are provided as services
  - Application servers
    - Example: Apache Tomcat, Red Hat Jboss, IBM WebSphere
  - Middleware platforms or services
    - Messaging or Queue service, DB service (SQL or so-called no-SQL or Big-data)
- Users upload code, which is executed on application servers and may interface with middleware services

- Full software / application development life cycle may be supported in a PaaS Cloud
  - Source control, build, test, etc.
- Underlying infrastructure resources (VM, load-balancer, storage, etc.) are acquired by PaaS system as needed
- Users do not have to acquire, buy and manage app server, middleware and dev & test platforms

- PaaS CSPs, Example:
  - Microsoft Azure PaaS
  - Google App Engine
  - VMWare Cloud Foundry
  - Amazon Elastic BeanStalk
- Application developed for one CSP PaaS will typically not be portable as is to another CSP PaaS

PaaS


Local (Tenant Premise)
Dev & Test App (such as Webapp)
Dev & Test
Configure Container (such as Tomcat/JVM parameters) and PaaS Services (such as DB, LB, etc.) or IaaS services (such as storage) accessed by App
Upload App (Such as Webapp [via Git Push, for example])
Cloud PaaS Environment
Runtime : Test
VM
App Server / Container (such as Tomcat)
App1
App2
App / Source Repo (such as under Git)
OS (such as Linux distribution X) + Kernel with support for Hypervisor Y
Dev & Test
Deploy
Runtime : Production
VM
App Server / Container (such as Tomcat)
App1
App2
OS (such as Linux distribution X) + Kernel with support for Hypervisor Y
PaaS Services (such as DB, LB, etc.)
IaaS Services: (such as Storage)

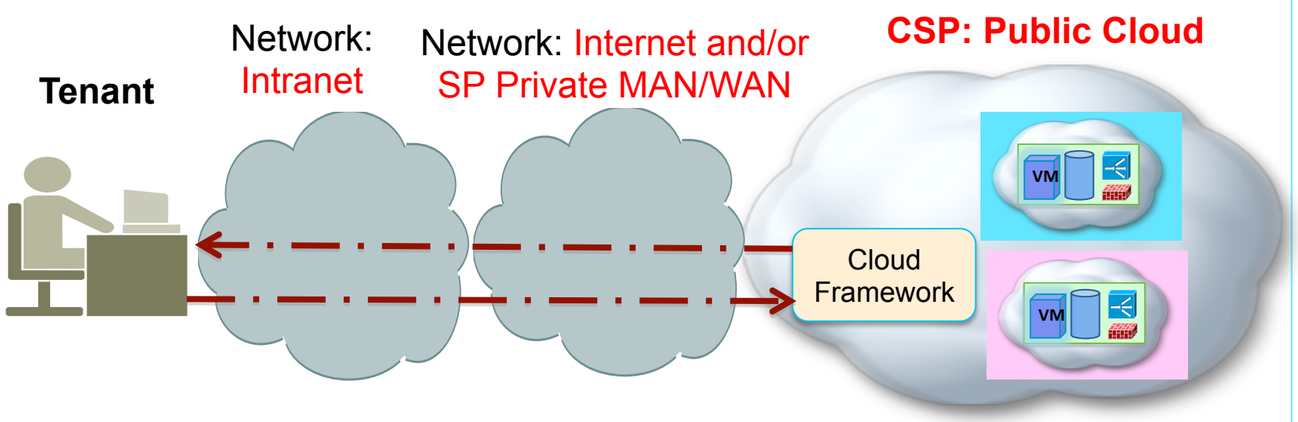
SaaS

- SaaS: Software as a Service
- Full-fledged finished application products typically used by end-users or developers are offered as services
  - Email, Calendaring, CRM, Document service

- Users (developers) can customize application *instance* using SDK and APIs provided by a SaaS provider
- SaaS provided by one provider can be interfaced with another
  - SalesForce and Google interface (CRM+Gmail)
  - VMForce: VMware + SalesForce in SalesForce Cloud

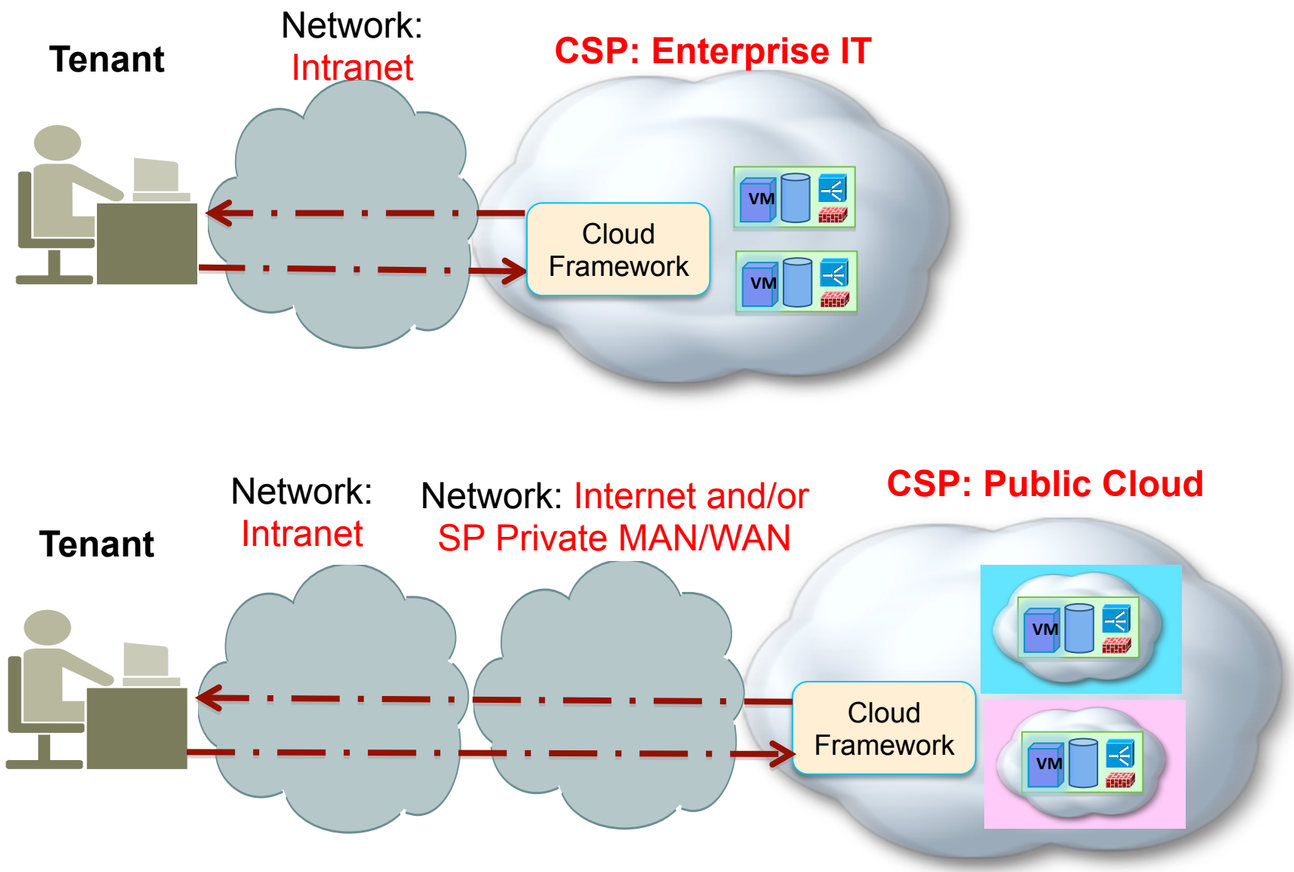
- Cisco Webex: Collaboration SaaS
- Cisco Hosted Unified Communication

Cloud Deployment Models – Public Cloud



- NIST definition: *The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.*
- A multi-tenant Cloud for use by any tenant

Cloud Deployment Models – Private Cloud



- NIST definition: *The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises*
- Enterprise IT (DCs) becomes CSP
- Can be a private Cloud that is virtual hosted in a Public Cloud managed by enterprise IT or CSP

Cloud Deployment Models – Community Cloud

- NIST definition: *The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.*

- More like *extranet* VPN for shared use

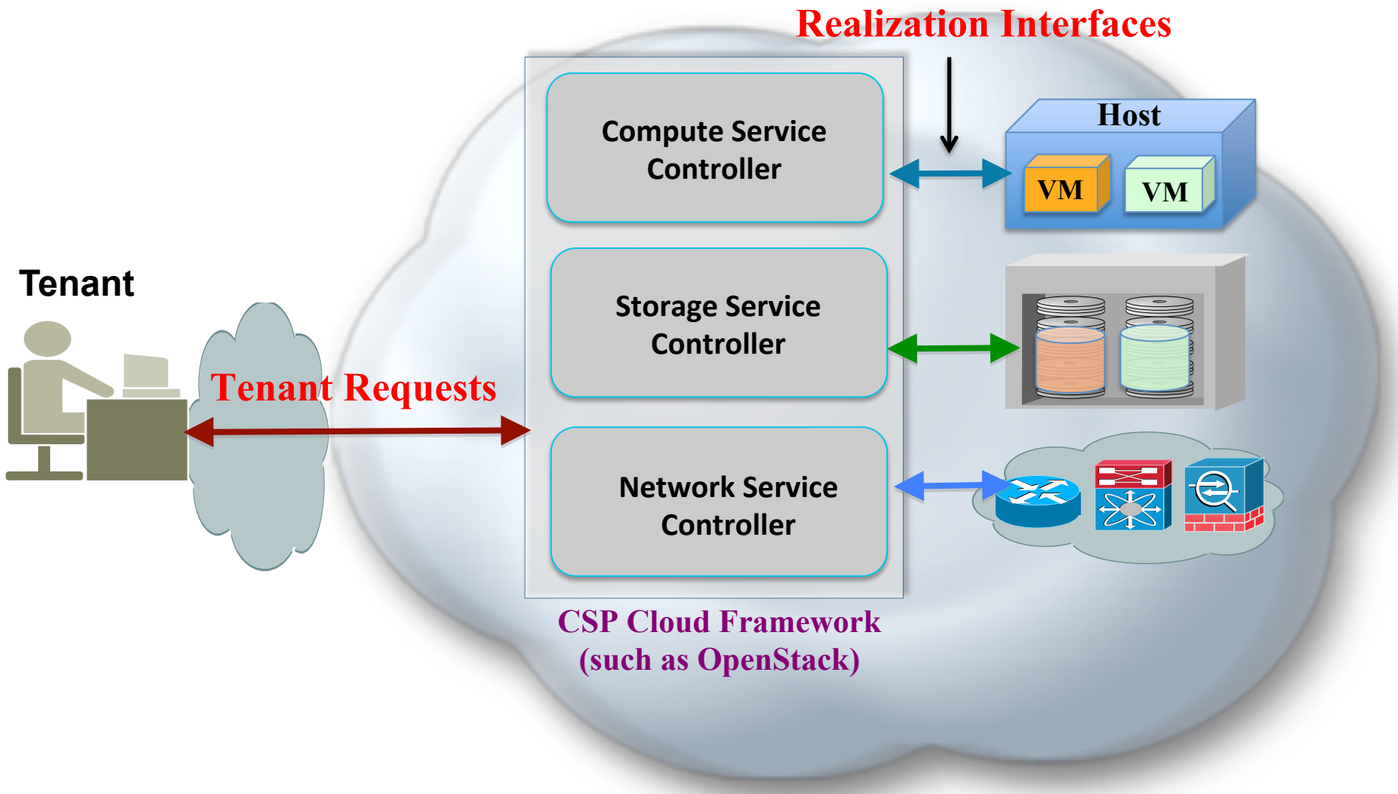
Cloud Deployment Models – Hybrid Cloud

- NIST Definition*: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are **bound together** by standardized or proprietary technology that enables **data and application portability** (e.g., cloud bursting for load balancing between clouds)*

- Note, multiple Clouds can be *bound together* at network level without the need for *data and app portability*

Cloud Controller / Frameworks / Systems

- Cloud Frameworks or Controllers performs all *Cloud Functions*
  - Fulfill tenants' Cloud resource CRUD requests in the Cloud infrastructure
  - Monitoring, accounting of resource usage and service assurance
- Controllers can be distributed and multiple Distributed systems may be interfaced with each other
- Multiple interfaces at multiple levels may be invoked on-demand to perform the Cloud Functions

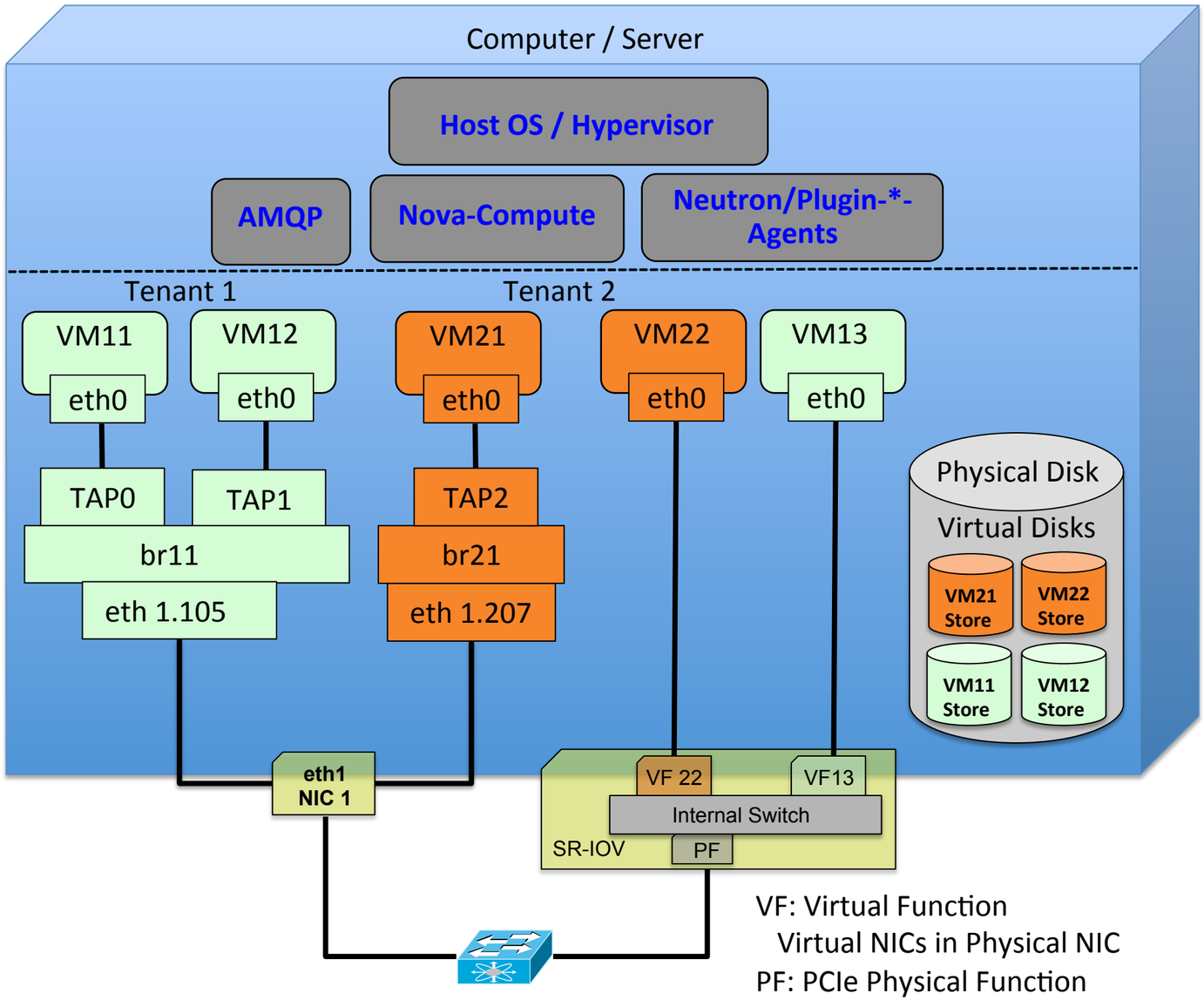


High-level simplified architecture

What Facilitates OREMAP: Virtualization is Key

- In a typical environment a physical or logical device or element is configured or provisioned for use (such as a server or network element)
- In a Cloud, a whole resource should be available on-demand, rapidly and elastically CRUD with multiple tenants sharing a physical resource
  - A whole computer, storage, network element
  - Not just plain configuration of a physical or logical element
- The best possible way to achieve this is via **virtualization** of resources
  - A virtual computer or virtual machine (VM) with everything in it that a physical computer may have
  - Similarly, storage and network

Element Virtualization

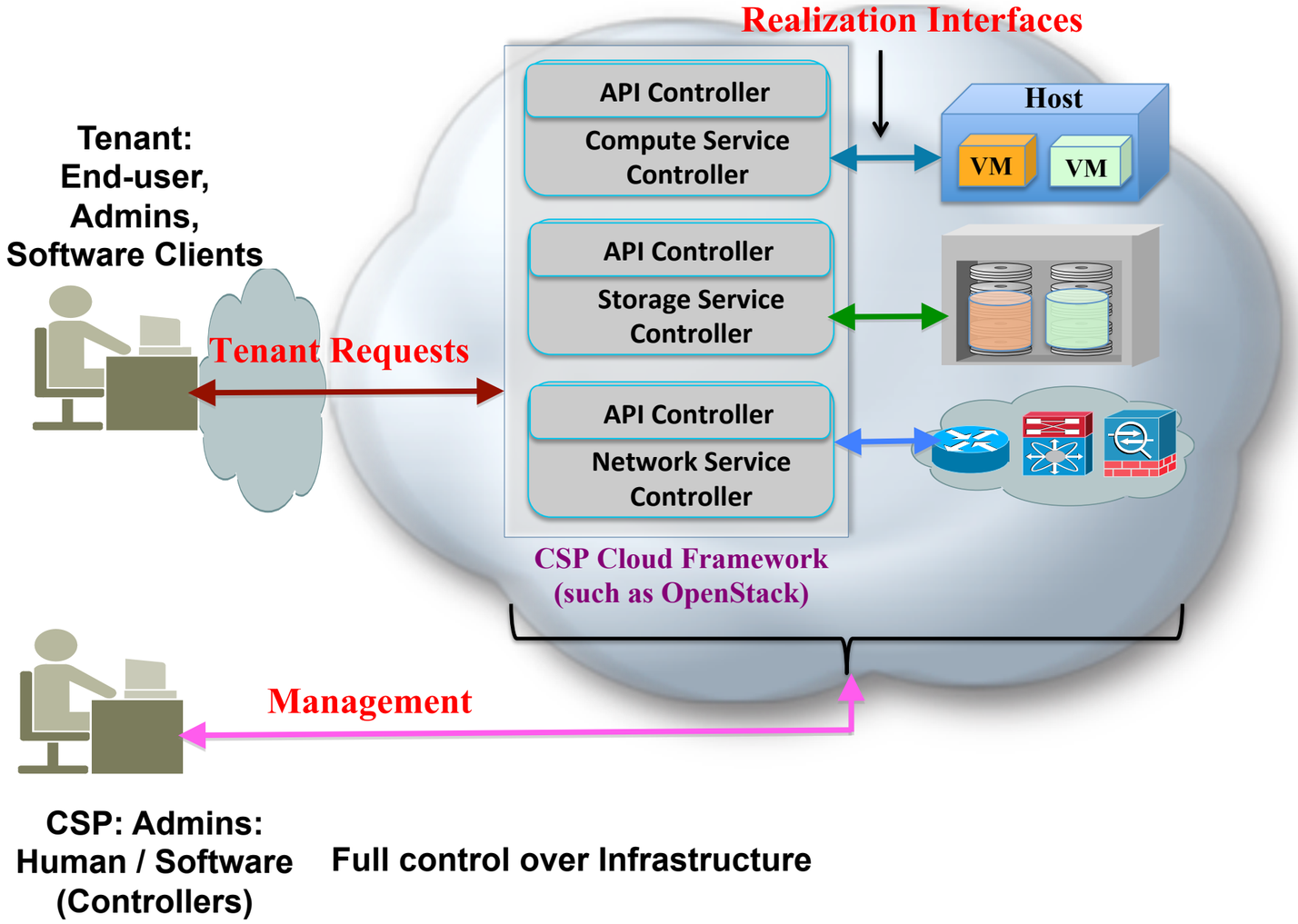


- Computer: VM
- Storage: Virtual Disk, Storage Objects
- Network: Software based Virtual Bridge / Switch / NIC
  - More later

Cloud Abstraction: Users Perspective: Tenant and CSP Users

"All problems in computer science can be solved by another level of indirection."
- Prof David Wheeler, U Cambridge

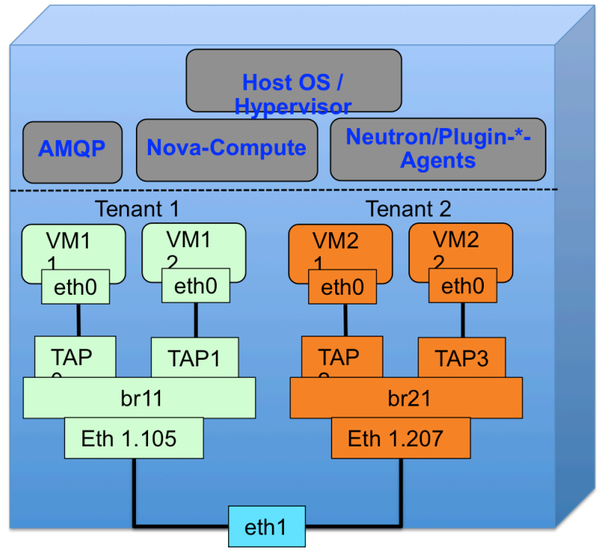
- Multiple levels of users, both human and software
- Tenant level
- CSP level
- There can be multiple admins: compute, storage, network, security, DC, WAN, etc.
- These users have layered access (at different abstraction level)



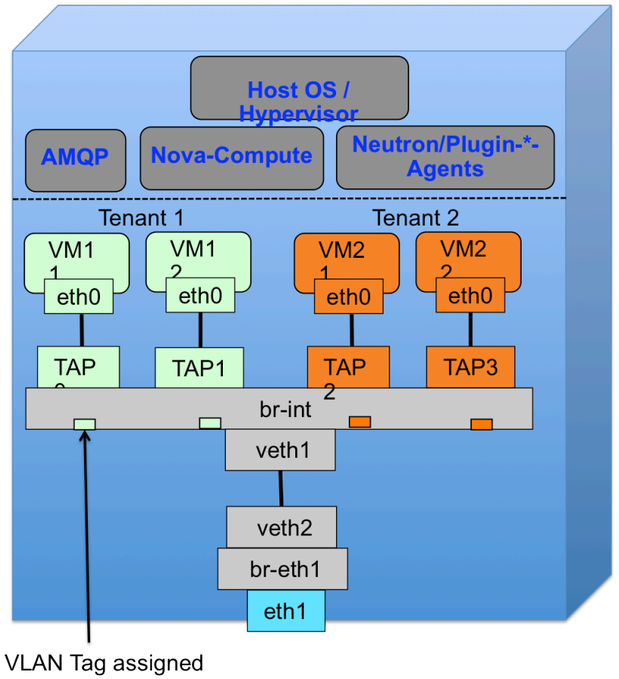
Cloud Resource Abstraction: Tenant facing

- CSP controlled Resources / infrastructure are abstracted and exposed to tenants
- Abstractions are defined by CSP / Controllers (actually Cloud Controller developers / vendors)
- KVM, Xen, VMware, Hyper-V, etc. VM

  → One abstraction
- Linux bridge, OVS, Nexus 1K, VLAN, VxLAN, VRF → One or two abstractions



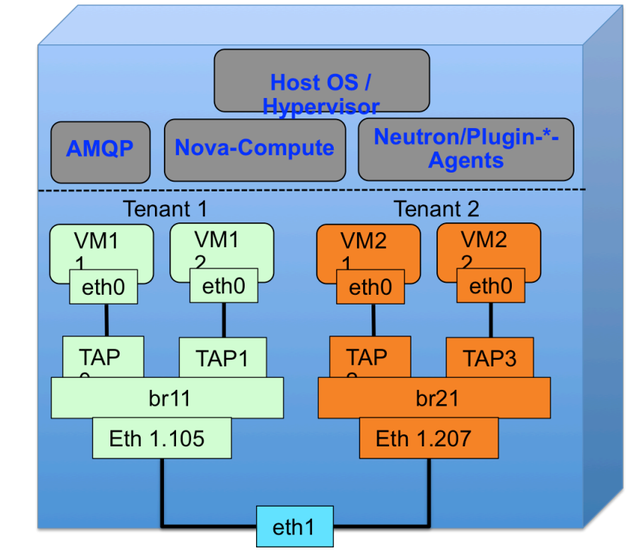
Linux Bridge



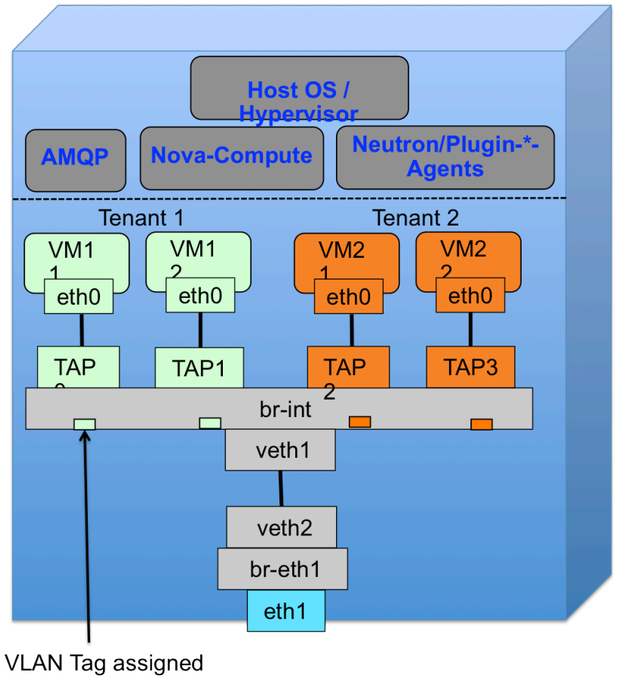
OVS

Cloud Resource Abstraction: Tenant facing

- Tenant Admin (TA) can define abstractions
  - For example, TA can extend or create new VM flavors (m1.small, m1.large etc.) if CSP allows or supports
- TA can CRUD certain abstract resources, Tenant end-users cannot
  - For example, CRUD Virtual Networks



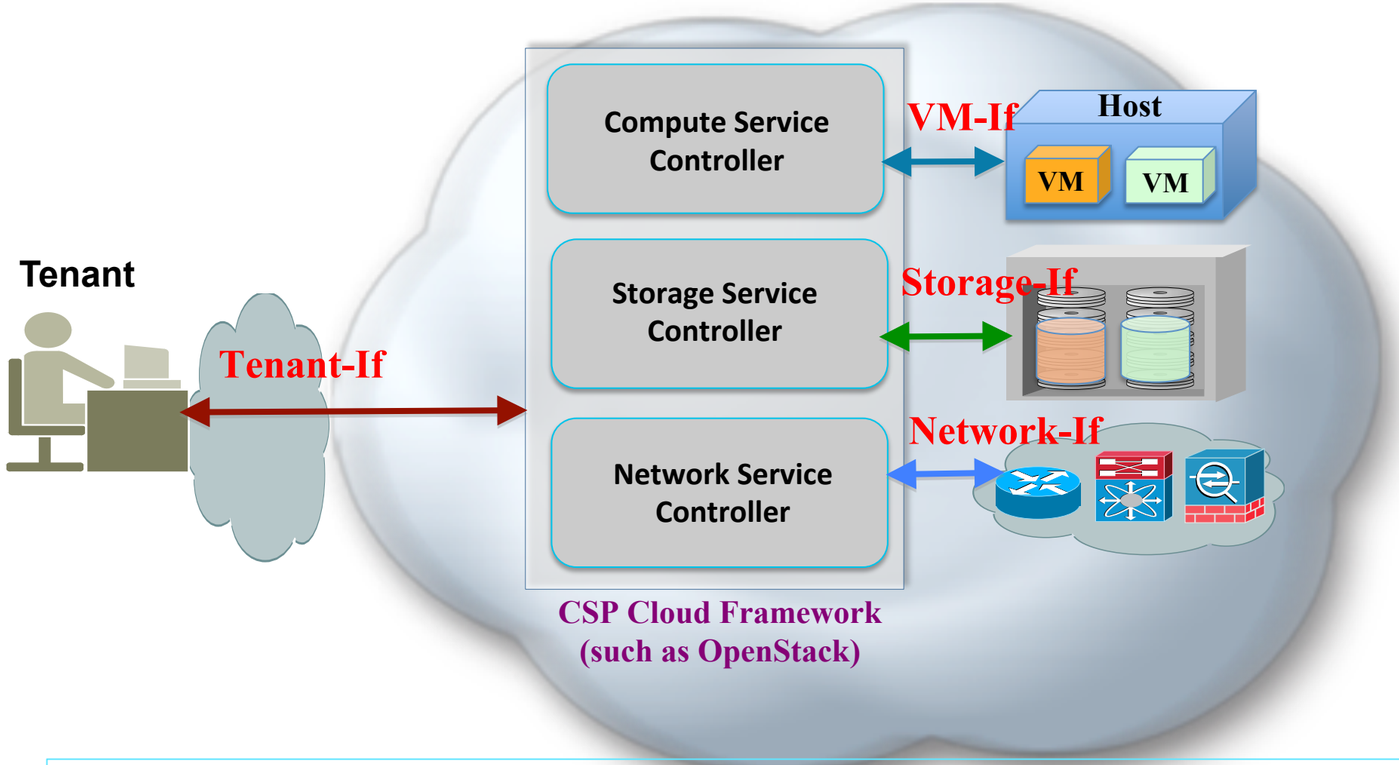
Linux Bridge



OVS

On-demand: On-demand Capable Interfaces

- To support on-demand characteristics of Cloud, *On-demand capable* interfaces are needed at all levels of Cloud Frameworks and infrastructure
- *Programmable* interfaces are on-demand capable



Tenant-If: Tenant facing [RESTful] IaaS Compute, Storage, Network, or PaaS / SaaS Interfaces or API

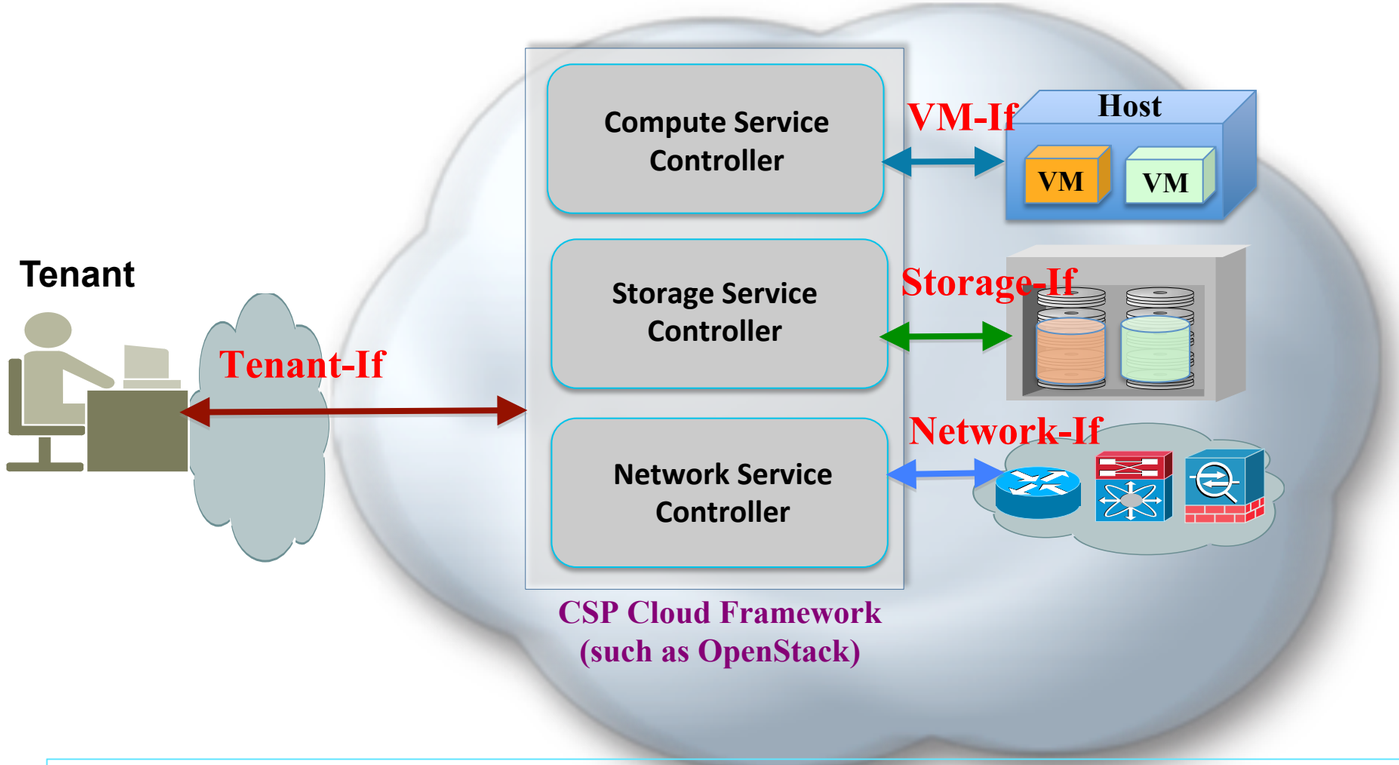
VM-If: Libvirt (KVM/QEMU) or Xen or VMWare or Hyper-V API for VM Life-cycle Management interfaces

Storage-If: Low level storage management and control interfaces

Network-If: OS (such as Linux) Networking or NM or Signaling or OpenFlow or Cisco OnePK Interfaces

On-demand: On-demand Capable Interfaces

- Tenants interacts with the Cloud using the Tenant facing Interfaces (Tenant-If)
  - To request CRUD of resources
  - Other Cloud functions
  - These interfaces incorporate tenant facing abstractions
- Infrastructure Interfaces
  - CSP Cloud Framework realizes tenant requests (via Tenant-If) in the infrastructure
  - Cloud framework invokes on-demand infrastructure level interfaces (VM-if, Storage-if, Network-If)
  - These interfaces may support relevant abstractions



Tenant-If: Tenant facing [RESTful] IaaS Compute, Storage, Network, or PaaS / SaaS Interfaces or API

VM-If: Libvirt (KVM/QEMU) / Xen / VMWare / Hyper-V API for VM Life-cycle Management interfaces

Storage-If: Low level storage management and control interfaces

Network-If: OS (such as Linux) Networking / NM / Signaling / OpenFlow / Cisco OnePK / Openstack Neutron / Open Daylight Interfaces

On-demand Interfaces: RESTful API

- REST (Representational State Transfer) based API widely used specially for tenant facing interfaces
- OpenStack Tenant facing API Example:
  - CRUD: **POST/PUT/GET/HEAD** http://<Cloud Service End Point (Cloud Service IP address & Port)>/<API Version>/<tenant ID>/**<Resource>/<resource Instance ID> {<body>}**
  - Action: **POST/PUT/GET/HEAD** http://<Service IP address & Port>/<API Version>/<tenant ID>/ **<Resource>/<resource Instance ID>/action {<body>}**
- In the request body and response XML and/or JSON format supported

- Example: Virtual Server Creation (via Openstack API)
  - POST <Compute Controller URL1> /v2/<Tenant1 UUID1>/**servers**

```
{   "server" :  {
        "name" : "VM11",
        "imageRef" : <URL2>/images/<Image UUID1>,
       ”flavorRef" : <URL1>/flavors/1,
       "networks":
       [ {
            "uuid": NET1-UUID1,
             "fixed_ip": "10.0.0.10”  } ]   }
}
```

- Example: Virtual Server Boot Action (via Openstack API)
  - POST …/**servers/UUID1/action**

```
{ "reboot" : { "type" : "HARD" } }
<!– HARD: Power cycle;
     SOFT: signal to OS to restart -->
```

Cloud Abstraction, Tenant-If and Infrastructure Mapping: Linux Bridge

POST <URL1> v2/<tanant UUID1>/**servers**

{ "server" : { "name" : ”VM11",

"imageRef" : ”<URL2>/images/<Image-UUID1>,

”flavorRef" : <URL1>/flavors/1,

"networks": [{ "uuid": <NET1-UUID1>,
"fixed_ip": "10.0.0.10” }] } }

Nova

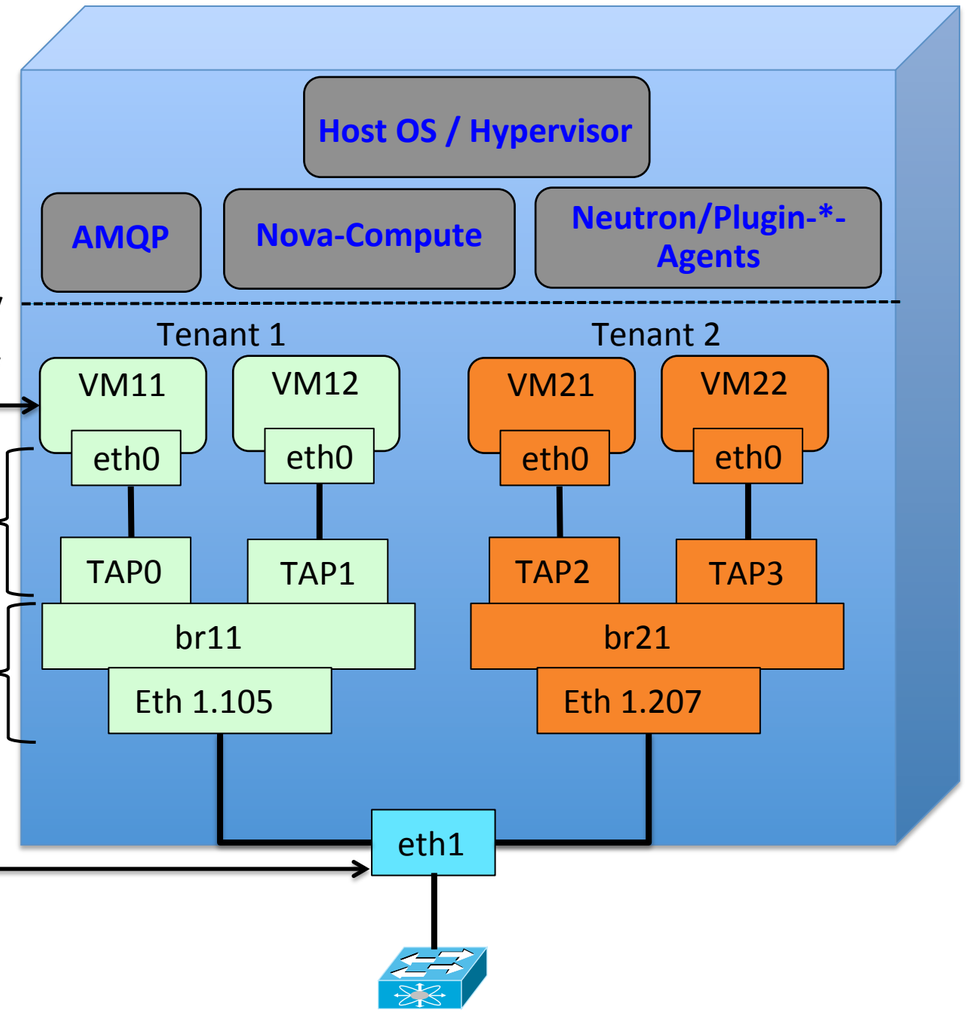
POST v2.0/**Ports**
{"port": {"name": "port1“, "admin_state_up": true,
"device_id": <VM-UUID1>, "network_id": <NET1-UUID1>} }

Neutron

POST v2.0/**Networks**
{"network”: { "name": ”NET1”, "admin_state_up": true,
"provider:network_type": "vlan",
"provider:physical_network": ”VNET1",
"provider:segmentation_id": 105 } }

Neutron

In linuxbridge_conf.ini:
[VLANS]
tenant_network_type = vlan
network_vlan_ranges = VNET1:100:110
[LINUX_BRIDGE]
physical_interface_mappings: VNET1:eth1



Cloud Abstraction, Tenant-If and Infrastructure Mapping: OVS

POST <URL1> v2/<tanant UUID1>/**servers**

{ "server" : { "name" : ”VM11",

"imageRef" : ”<URL2>/images/<Image-UUID1>,

”flavorRef" : <URL1>/flavors/1,

"networks": [{ "uuid": <NET1-UUID1>,
"fixed_ip": "10.0.0.10” }] } }

Nova

POST v2.0/**Ports**
{"port": {"name": "port1“, "admin_state_up": true,
"device_id": <VM-UUID1>, "network_id": <NET1-UUID1>} }

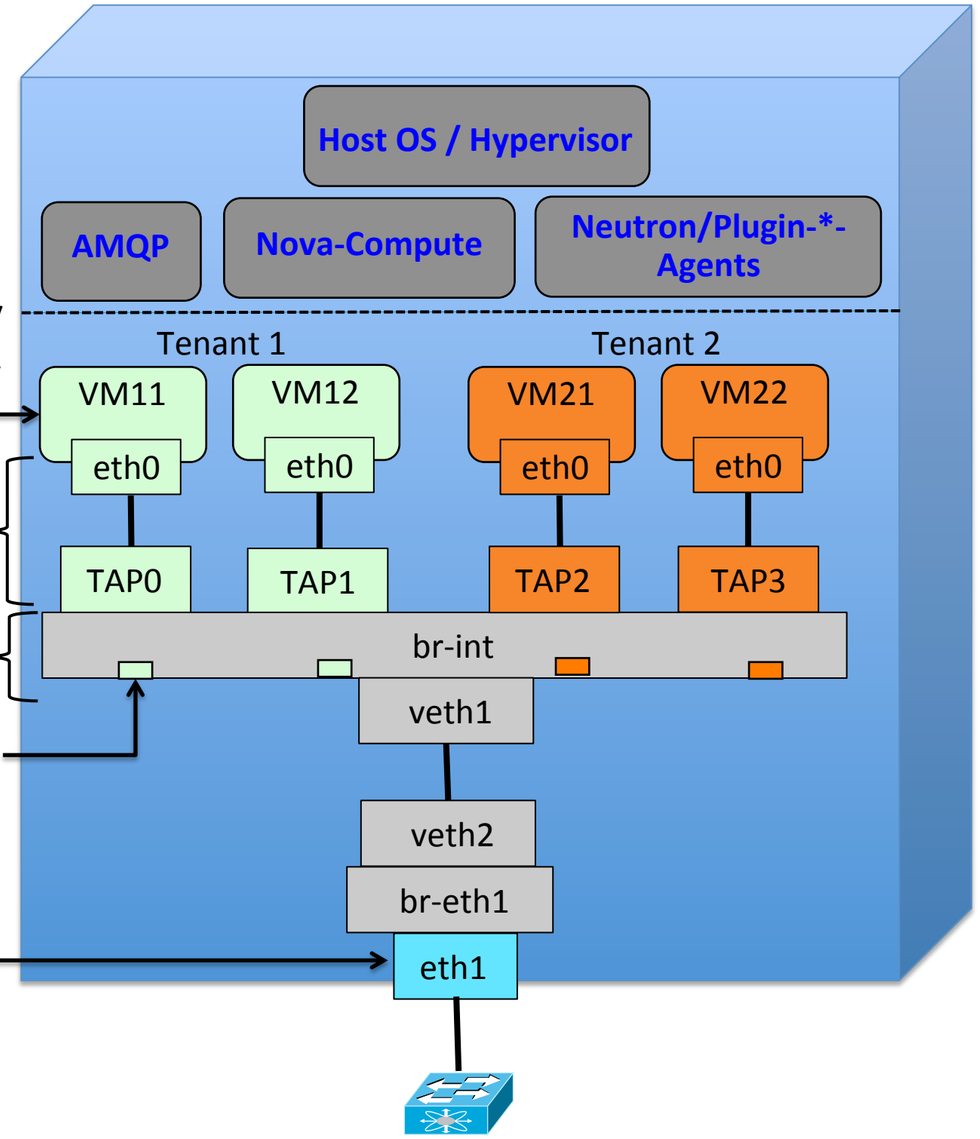
Neutron

POST v2.0/**Networks**
{"network”: { "name": ”NET1”, "admin_state_up": true,
"provider:network_type": "vlan",
"provider:physical_network": ”VNET1",
"provider:segmentation_id": 105 } }

Neutron

VLAN Tag assigned

In ovs_neutron_plugin.ini:
tenant_network_type = vlan
network_vlan_ranges =VNET1:100:110
integration_bridge = br-int
bridge_mappings = VNET1:br-eth1

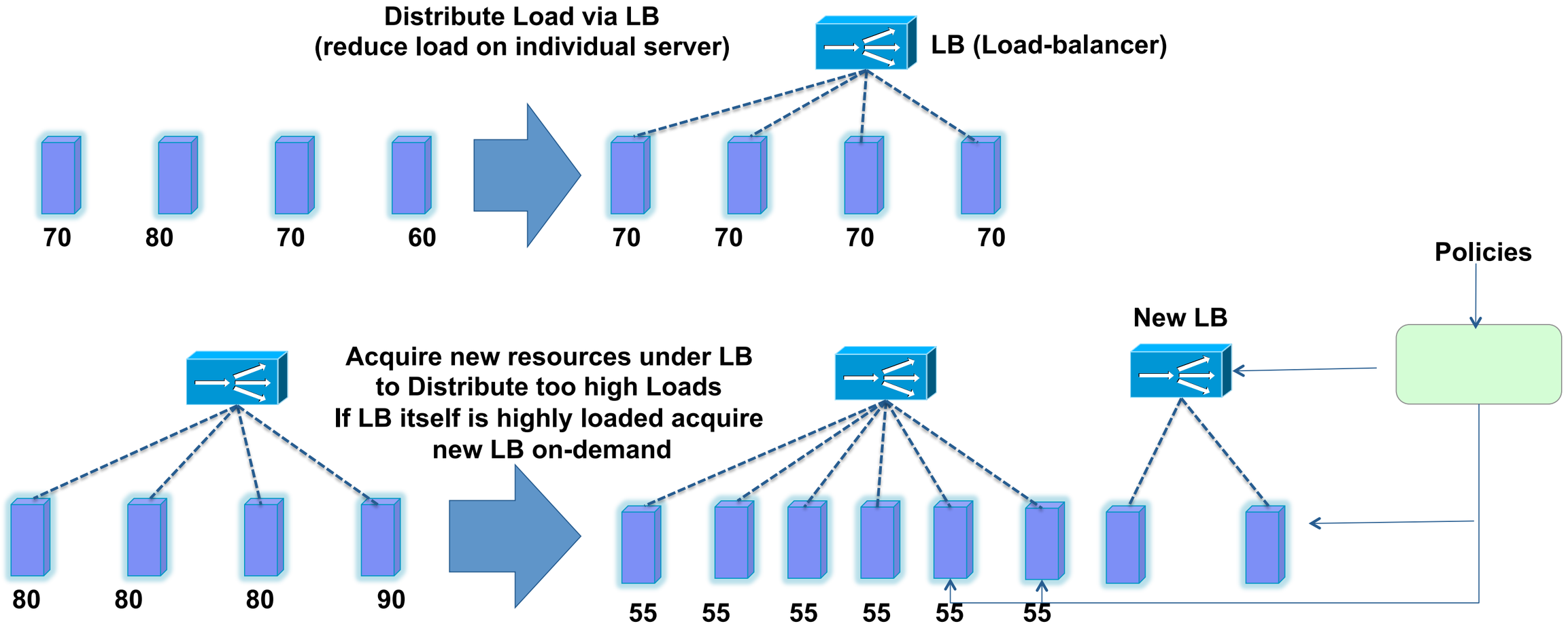


Elastic Scaling

- Cloud computing allows scaling (create / delete) of resources elastically
- When more resources are needed it is possible to ***scale up*** by acquiring more powerful compute, storage, network elements (hardware)
- But Cloud allows ***scaling out*** / **back** by incremental addition / deletion of virtual elements
  - Obviously a CSP still needs to acquire powerful hardware, but a tenant does not have to

Scaling Out Network Resources

- Not only compute and storage resources can be scaled out (typical focus) in a Cloud
- But also network resources



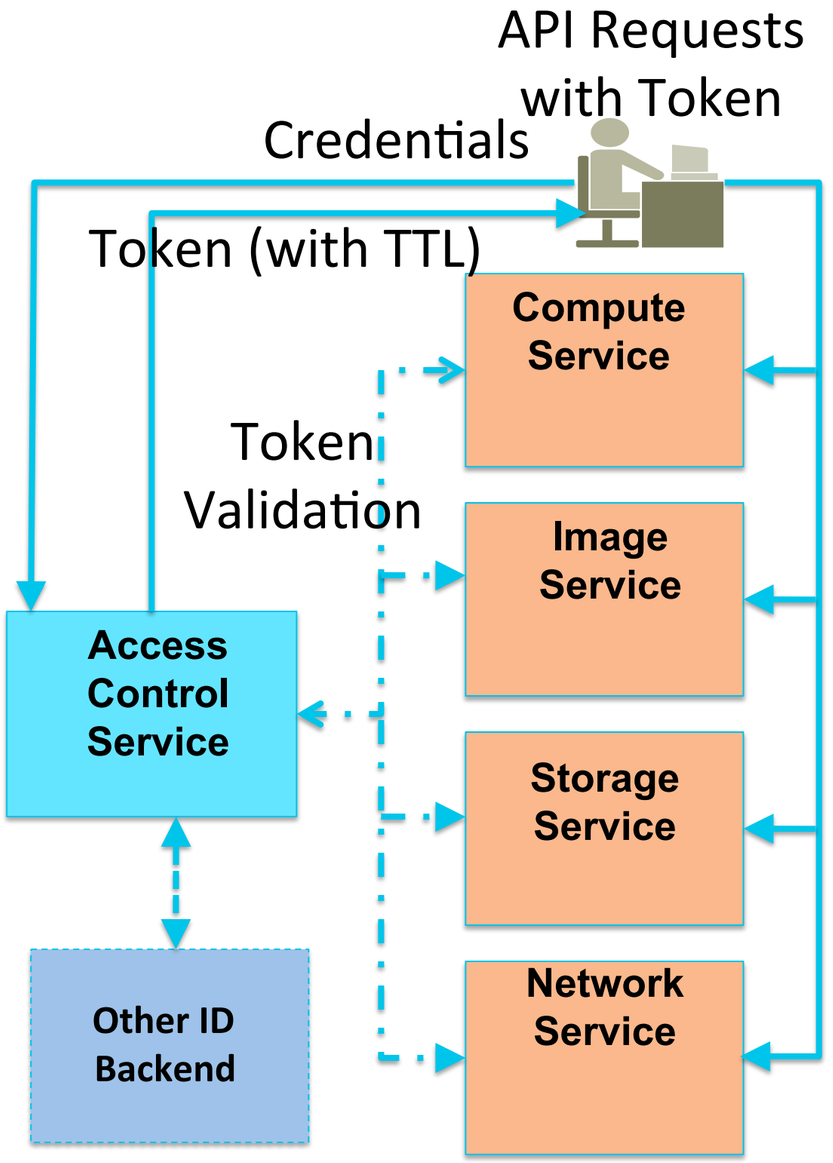
Access Control

- Access control in a Cloud has to be considered from multiple perspectives beyond user ID management and AA (authentication and authorization)

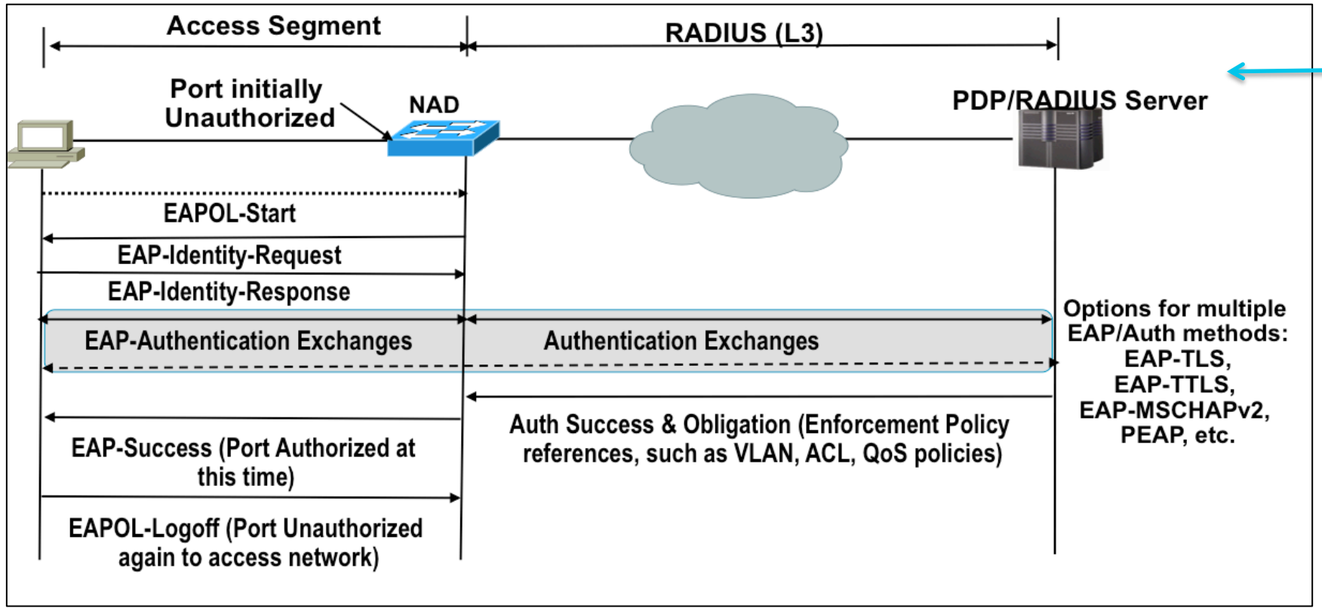
- Single sign-on is imperative in the presence of a multiple services that may be “touched” for a single CRUD action
  - For example, Looking at this example of virtual server creation, it is obvious multiple services are touched

```
POST <URL1> v2/<tanant UUID1>/servers
  { "server" : { "name" : ”VM11",
    "imageRef" : ”<URL2>/images/<Image-UUID1>,
    ”flavorRef" : <URL1>/flavors/1,
    "networks": [ {  "uuid": <NET1-UUID1>, "fixed_ip": "10.0.0.10” } ] } }
```

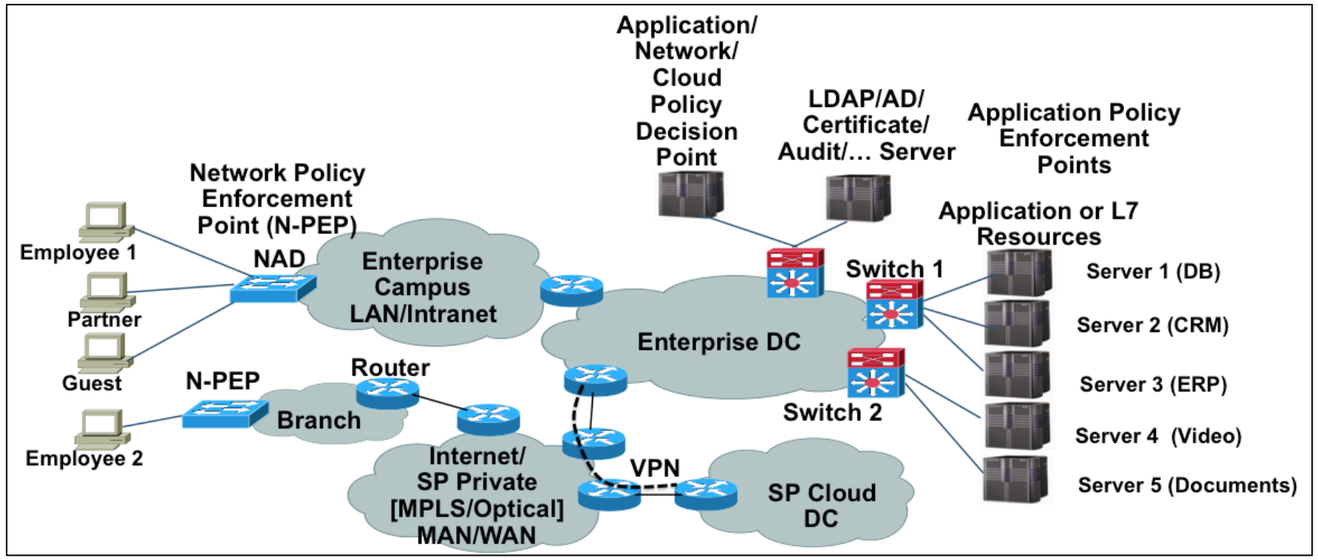
  - Originating service (compute in this case) should be able to authorize the user automatically across services
  - A Token based approach may be useful
- Role based: Example: An IaaS admin can create a virtual network, but a regular user cannot



Access Control: Public / Hybrid Cloud



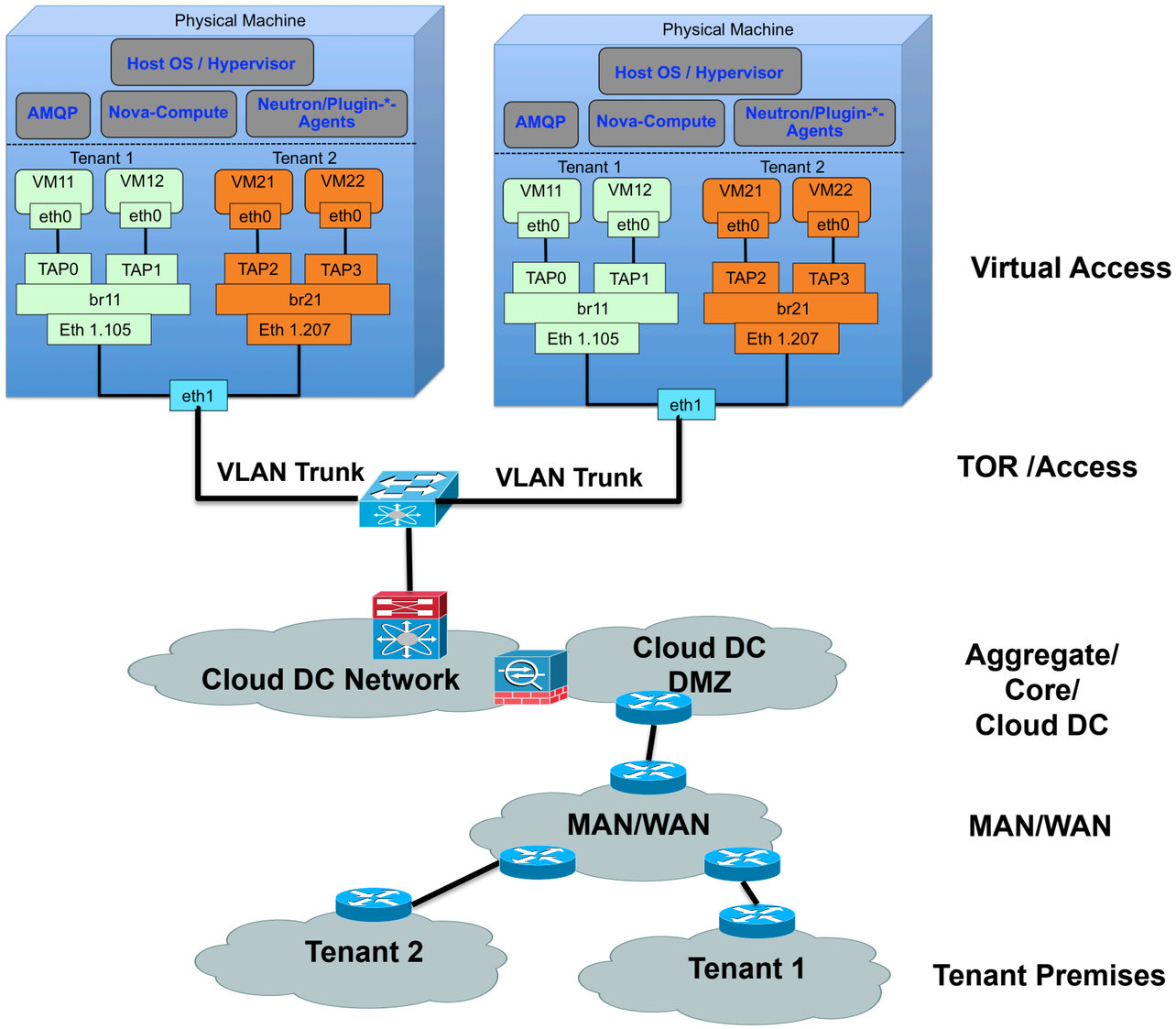
Typical network access control flow in an enterprise



- Network access control in enterprise should be integrated / interfaced with Cloud and Cloud resource access control

Cloud Networking
Network Virtualization

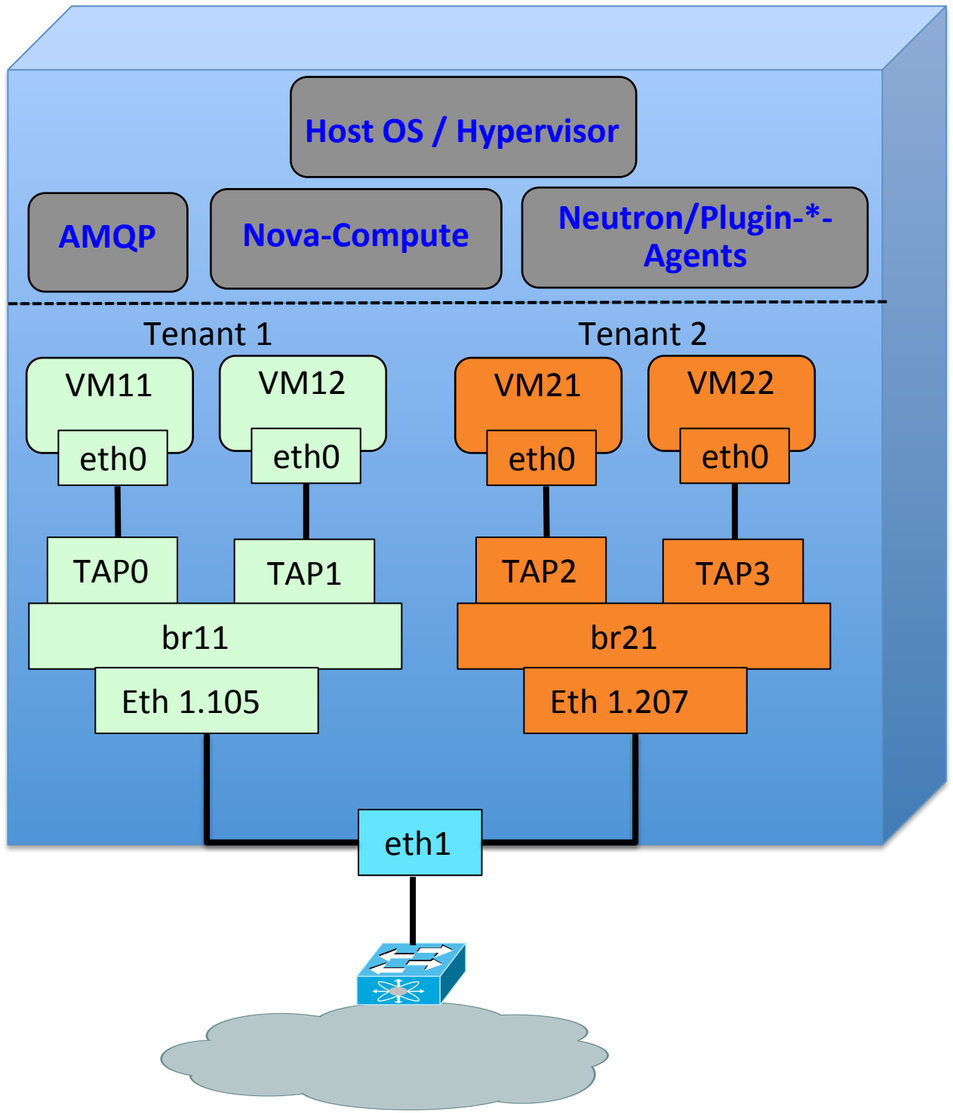
Introduction



- We will consider end-to-end (E2E) network, not just virtual network within computer

- Virtual Access Network
- Hardware Access Network or TOR (top of the rack)
- Cloud DC Network
- Cloud DC MAN/WAN Access
- MAN/WAN
- Tenant premises: Campus, DC

Virtual Access Network – Virtual Embedded Switch / Bridge



- With computer virtualization, *virtualized access switch* (software switch) moves within the physical computer
- Access ports/VLANs within the server
- Also known as *Virtual Embedded Bridge* (VEB)
  - Software based IEEE 802.1Q and relevant switching / bridging capabilities
  - Linux Bridge, OVS, Cisco Nexus 1K

Virtual Access Network – VEB to Hardware

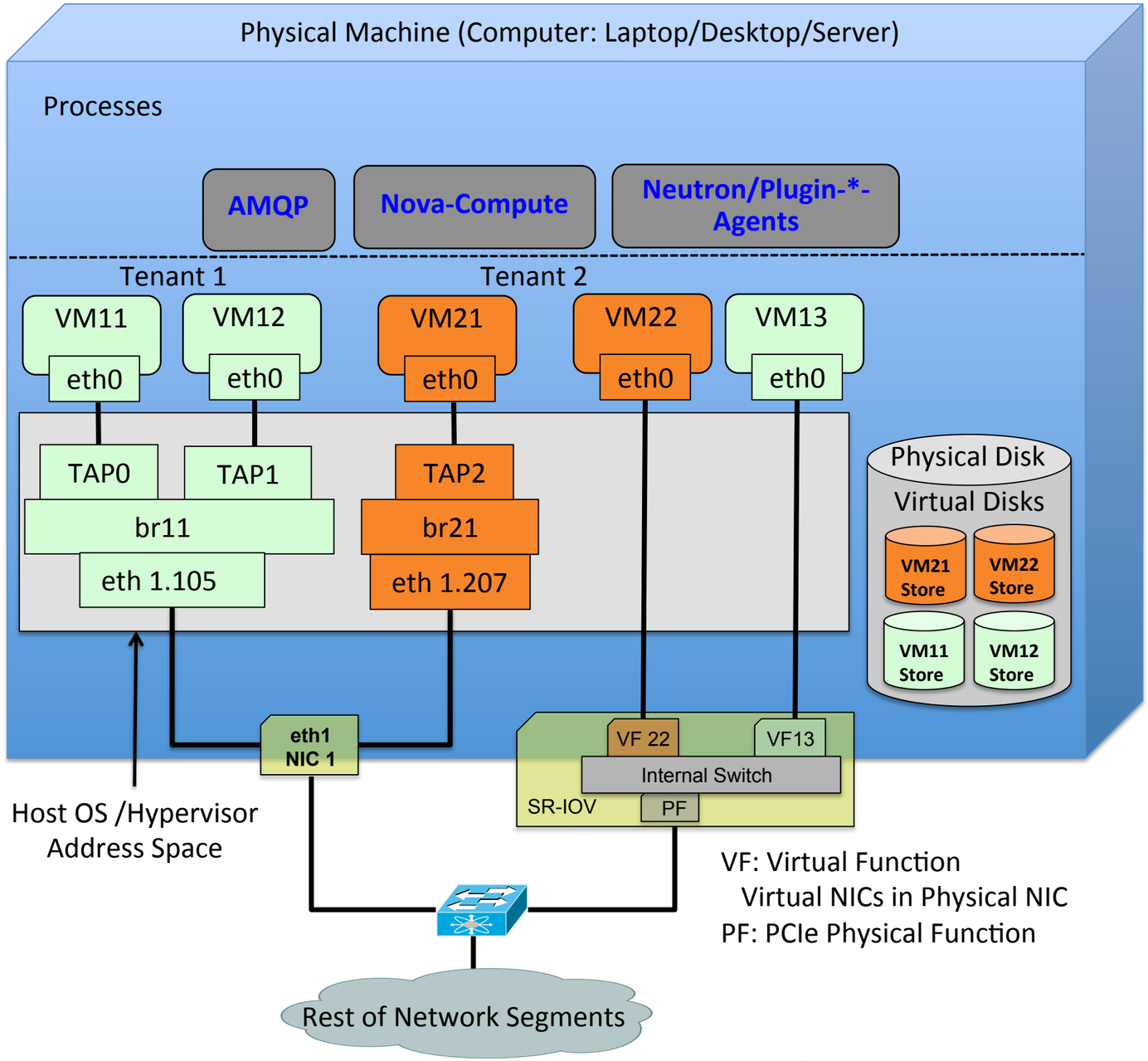
- Software based Virtual Switch has limited capabilities
- Software based switch adds extra load to host server
- I/O performance suffers

- Move VEB / Virtual Switch to Hardware

- In Hardware NIC on Computer with multiple Virtual NICs
  - SR-IOV (Single Root – IO Virtualization)
  - Cisco UCS M81KR VIC (Virtual Interface Card)

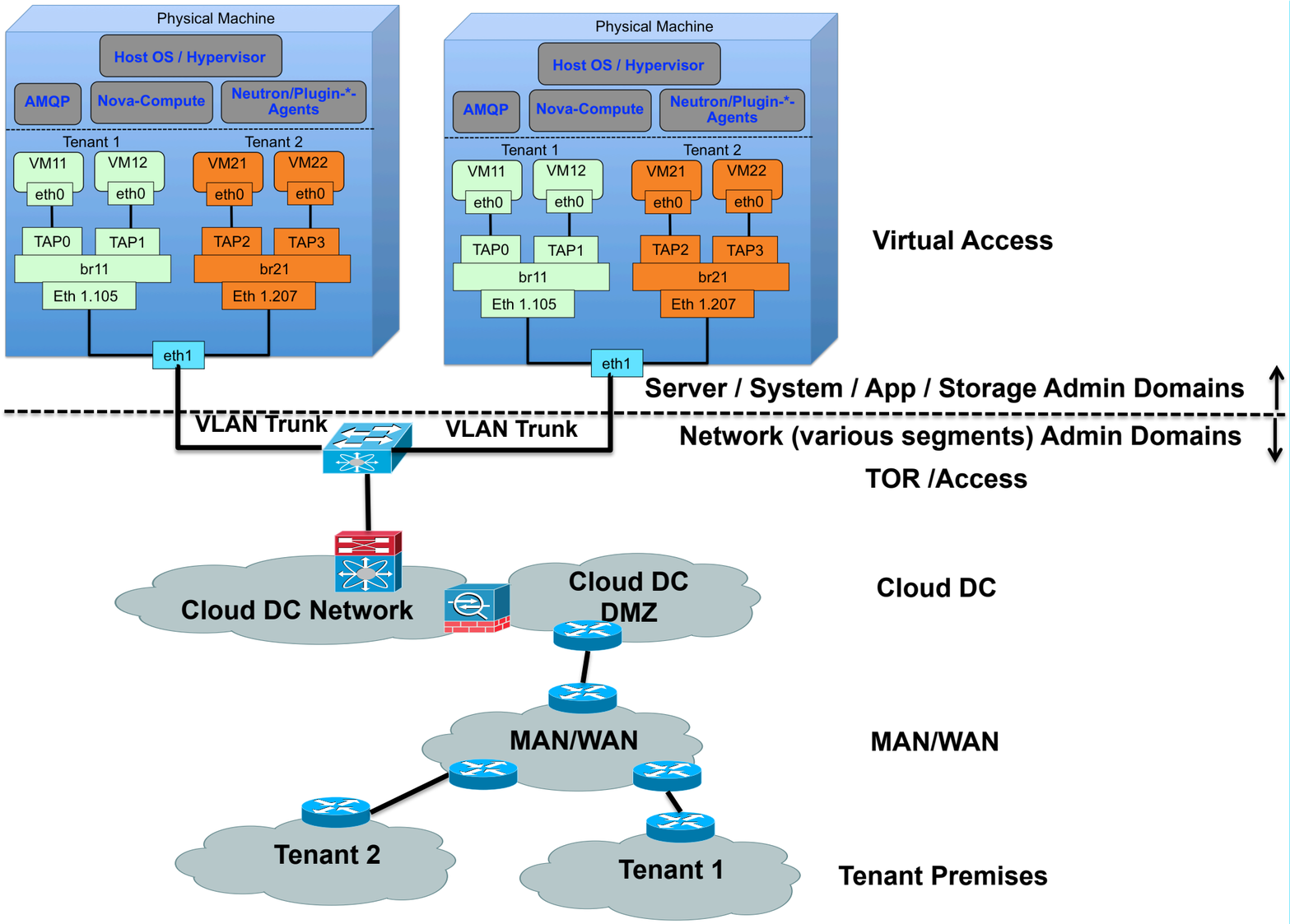
- To upstream Hardware Access or Aggregate Switch
  - IEEE 802.1BR (VBE: Virtual Bridge Port Extension; Cisco term: FEX / VM-FEX / VN-Tag)
  - IEEE 802.1qbg (Edge Virtual bridging)

Virtual Access Network – Direct IO Device Assignment



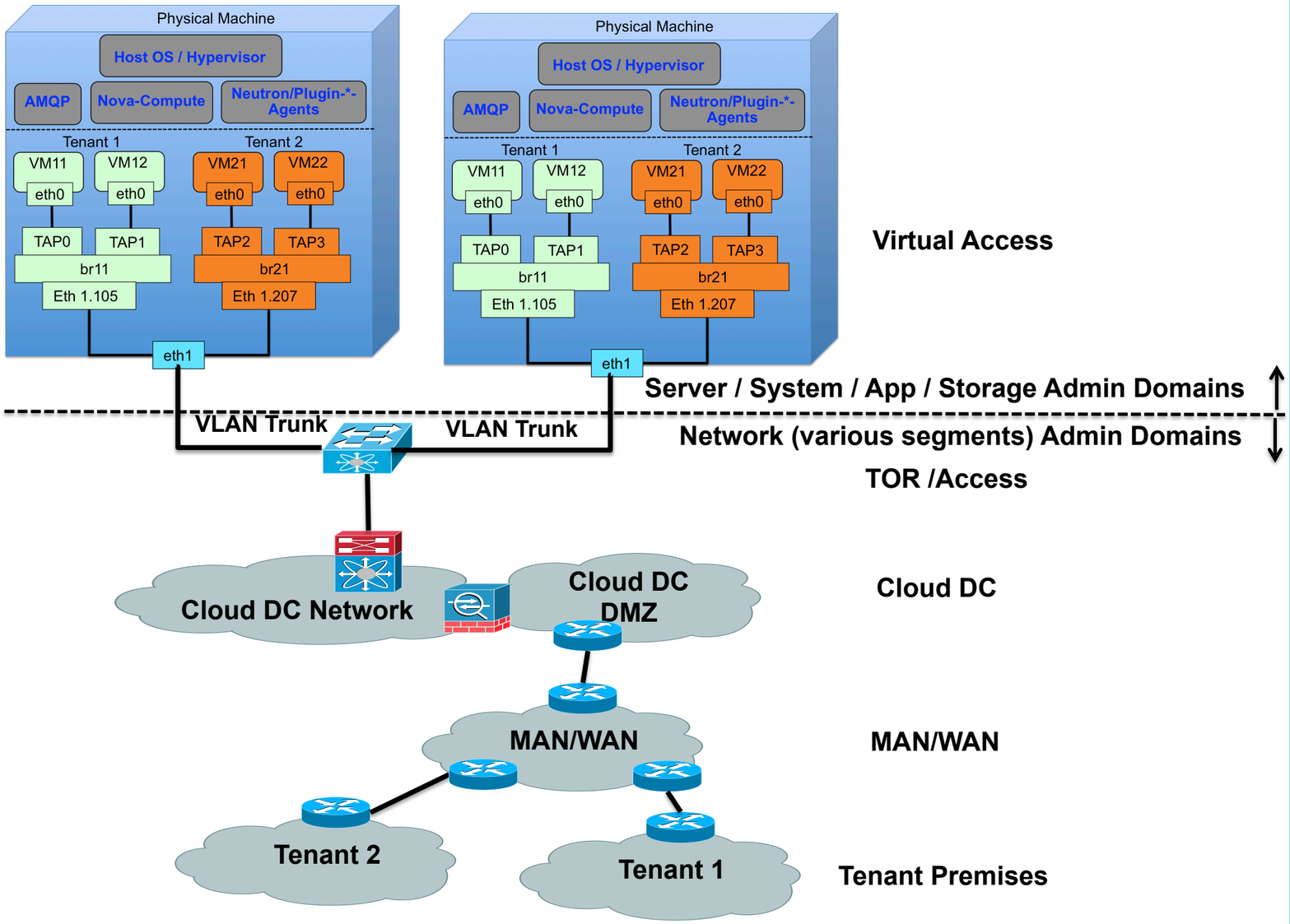
- Also known as *PCI-pass-through* or *Hypervisor-bypass*
- SR-IOV
  - Multiple Physical Functions (PF: up to 16 currently) with full PCIe capabilities
  - Multiple Virtual NIC / adapters or Virtual Functions (VF) with light-weight PCIe capabilities
  - Guest OS sees the associated VF as an attached PCIe device
- Need special-purpose hardware/chip, such as IOMMU (Intel VT-d, for example)
- Full hypervisor-bypass may not be possible, specially for interrupts → special solution needed

Issues with VEB



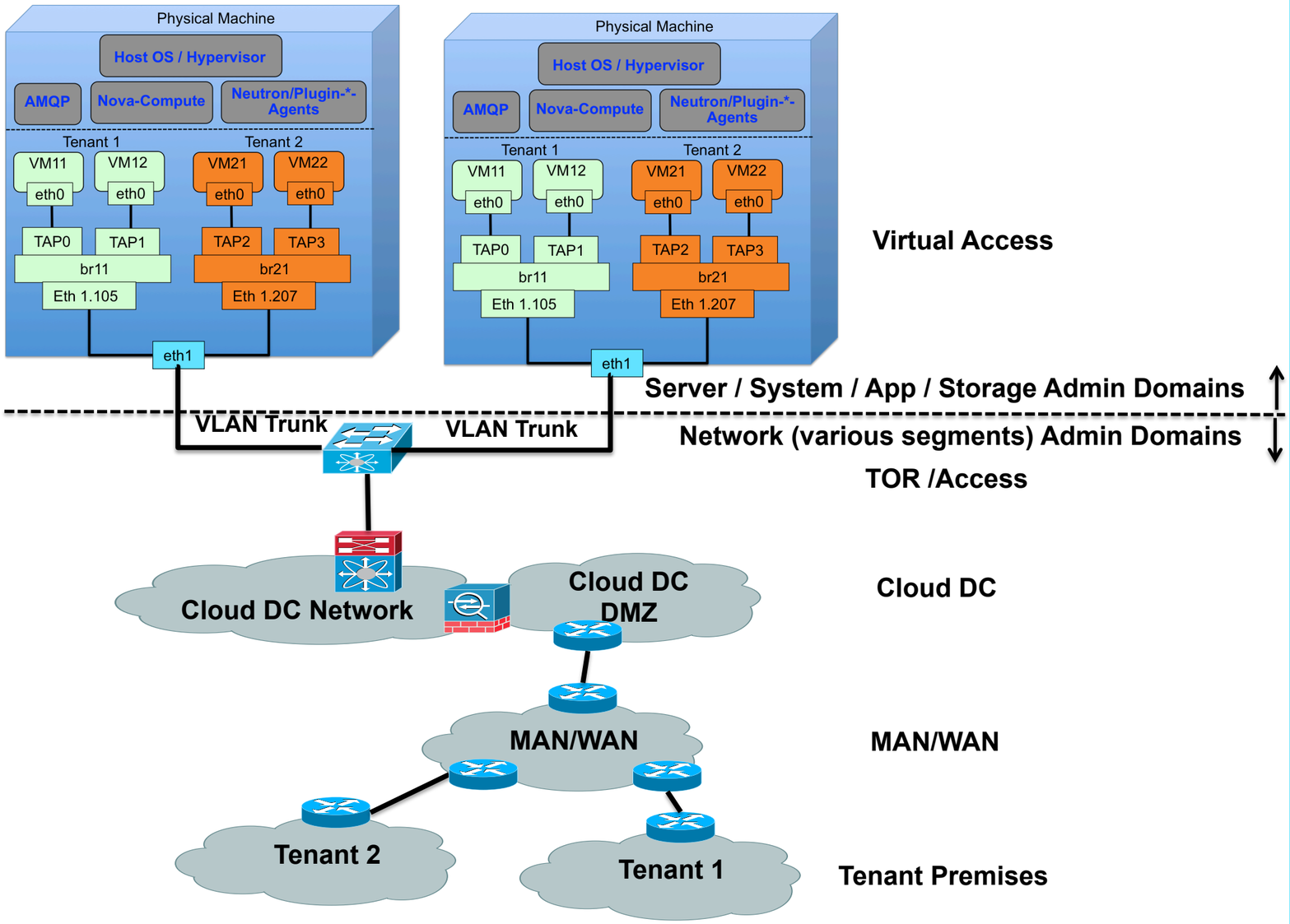
- An infrastructure typically has multiple *Admin Domains*
  - D1: Server / System / App / Storage Domains
  - D2: Network domains with its own subdomains

Issues with VEB



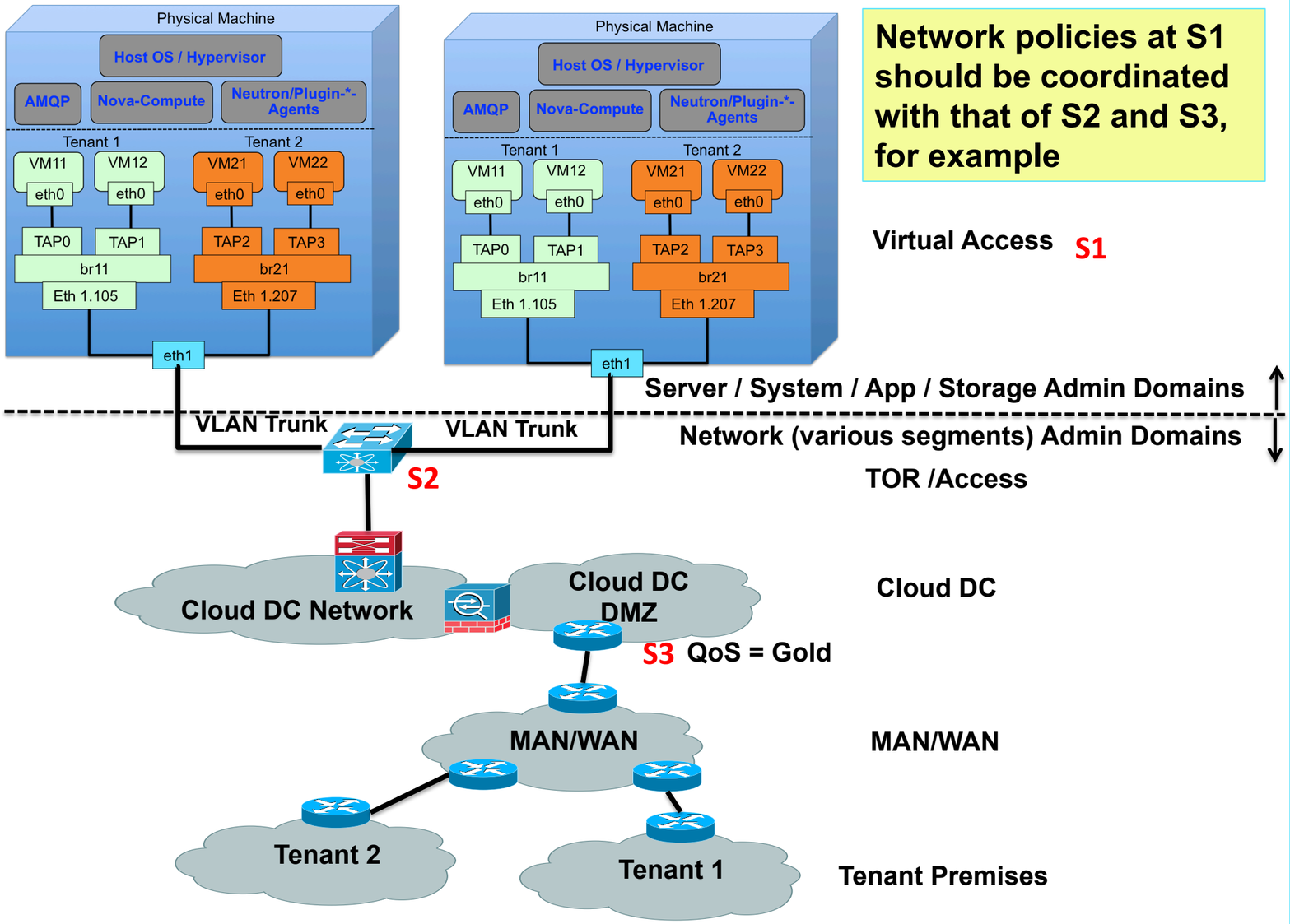
- With VEB, Access Switches moves from D2 to D1 with implications or challenges
  - Consistent Network Policy programming and application
  - Comprehensive and centralized visibility of network traffic
  - Virtual and physical network resource (VEB and ports) co-management
  - Huge number of virtual ports may have to be dealt with
  - Too many network management control points

Issues with VEB: Admin Domains



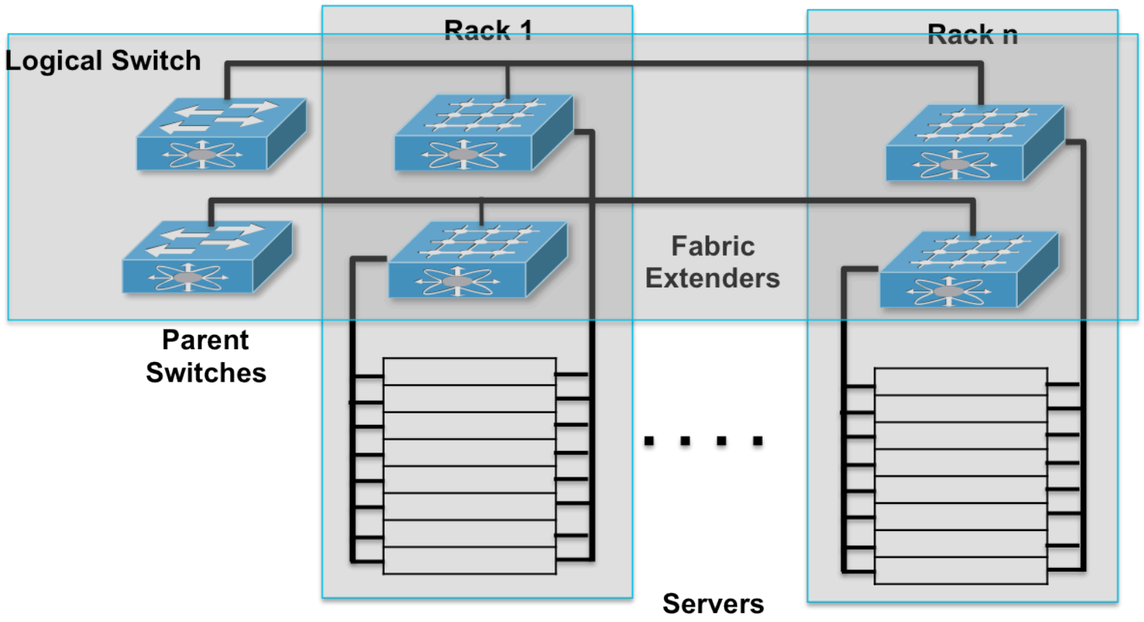
- Idealistic: Turn domains into one Single Domain!!
  - Division of expertise / labor → Admin domains
  - Infrastructure will not magically transform into one admin domain overnight or ever → Division of labor / expertise
  - Even in *fully automated (!!)* environment domain division useful

Admin Domains – Coordination



- Typically Net Admin (via NM tools) applies complex network configurations on behalf of server/system/app domains
- Now Server admin controls Network Policies at S1
- For consistency network policy programming / management should be coordinated

Fabric Extender (FEX) in Access Network

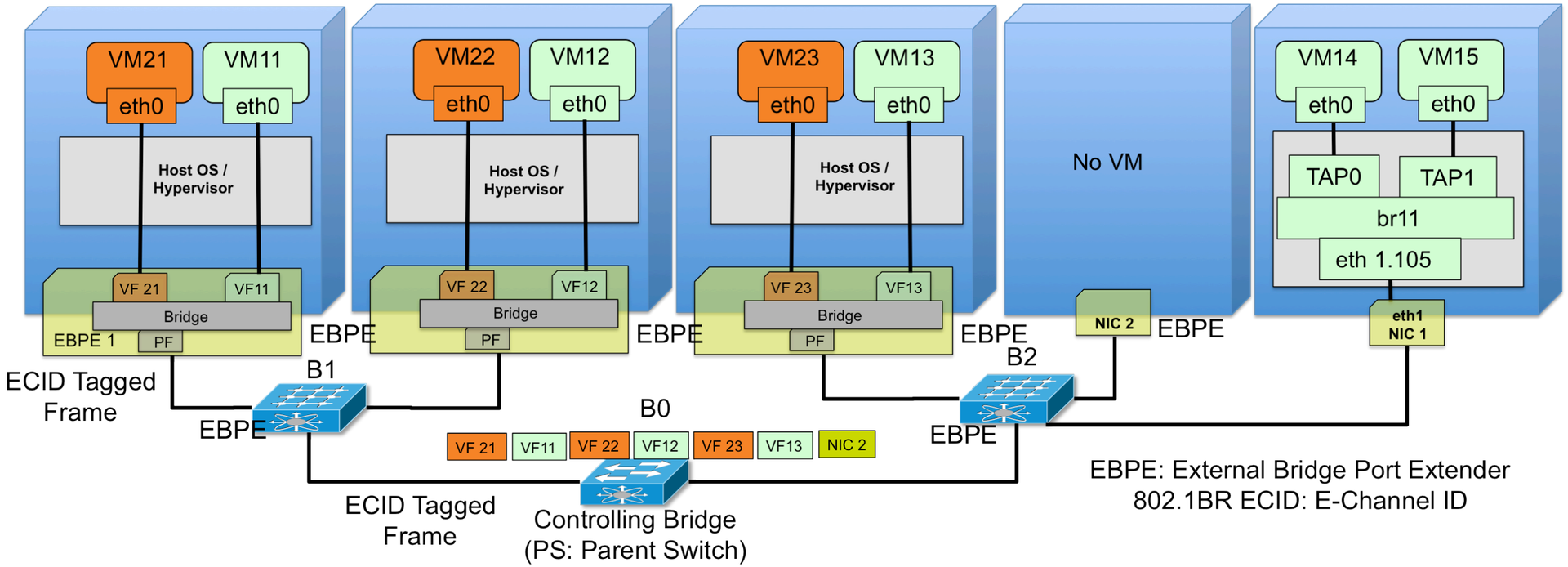


Parent Switches: such as Cisco Nexus 5K
Fabric Extenders: Such as Cisco Nexus 2K

- Fabric Extender (Chassis Virtualization) enables admin domain coordination
- A set of bridges / switches and ports as one big Logical bridge / switch
- Consider 8 FEX with 48 host / server ports → 384 ports
- 384 ports visible, managed, controlled from Parent switches (PS)
- Forwarding / Control plane on PS

- Network policies (templates) can be configured and applied on 384 ports centrally
- FEX ←→ PS links: 4-16 10Gbps, for example
- **IEEE 8021.BR** standardizes FEX, including extension to incorporate VM environment

IEEE 802.1BR

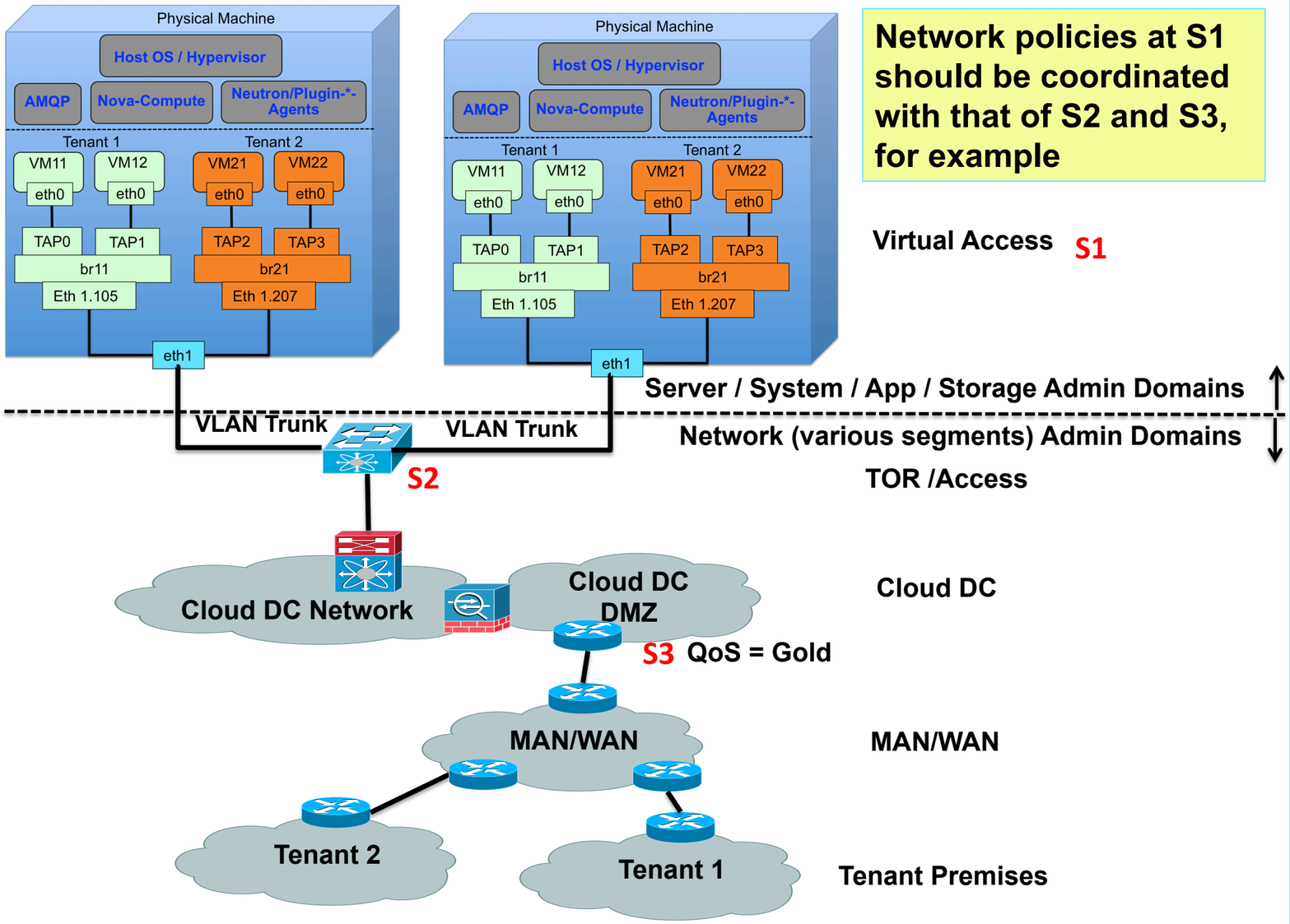


App Payload
TCP/UDP
IP
802.1BR E-Tag
Ethernet (802.1Q)

- [Note, various, sometimes confusing terms used, such as vNIC, VIF, Veth, Vethernet, etc.; we'll use our own]
- Virtual NICs (VNIC) appears as a virtual port (VPRT) on the Controlling Bridge
  - VNIC is called [VM attached] Extended Port in 802.1BR
    - VFs in SRIOV or VIF in Cisco VIC
  - VPRT is called C-VLAN (C: Customer) Component Port in 802.1BR

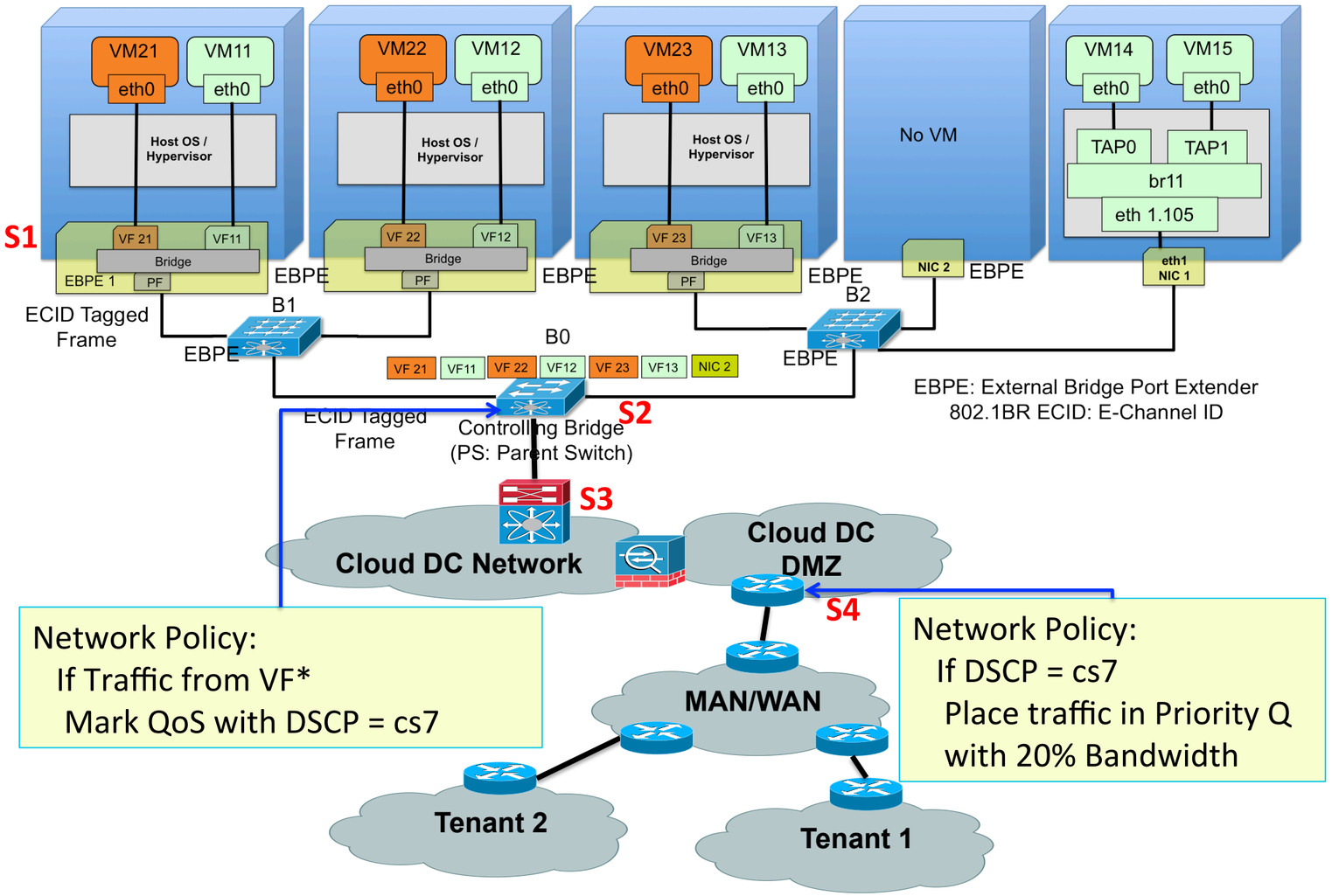
- When VF21 VNIC created and attached to VM by a Server Admin
- EBPE1 will send *extended port create message* to B0
- B0 creates VPRT and assigns ECID
- Any management and control action can be performed on VPRT on B0

Management / Control (M / C) Without 802.1BR



- S1 programmed / configured / provisioned for network policies by Server / VM Management System
- S2, S3 and other points by Network Management Systems
- With 802.1BR support, M / C points are reduced, specially at the DC access and aggregate with huge number of physical and virtual ports

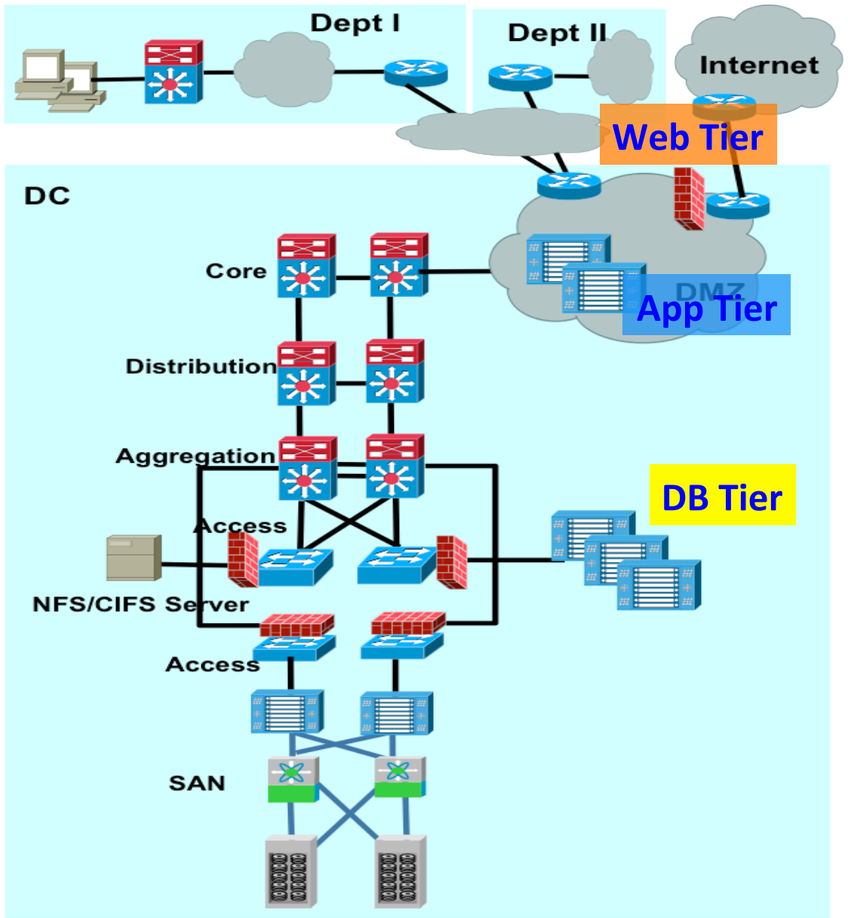
Admin Domains – Coordination – With IEEE 802.1BR



- A Network Admin defines policies (also known as *port-profiles*) on B0
- Port-profile IDs (PPID) sent to Server admin systems (through means outside the scope of 802.1BR)
  - For example, Cisco Nexus 5K supports this with VMware vCenter
  - With programmable interfaces, communication between admin domains becomes easier
- Server admin creates VNIC, associates a PPID and attaches VNIC to VM
- 802.1BR will then create VPORT in B0 with PPID automatically attached to VPORT

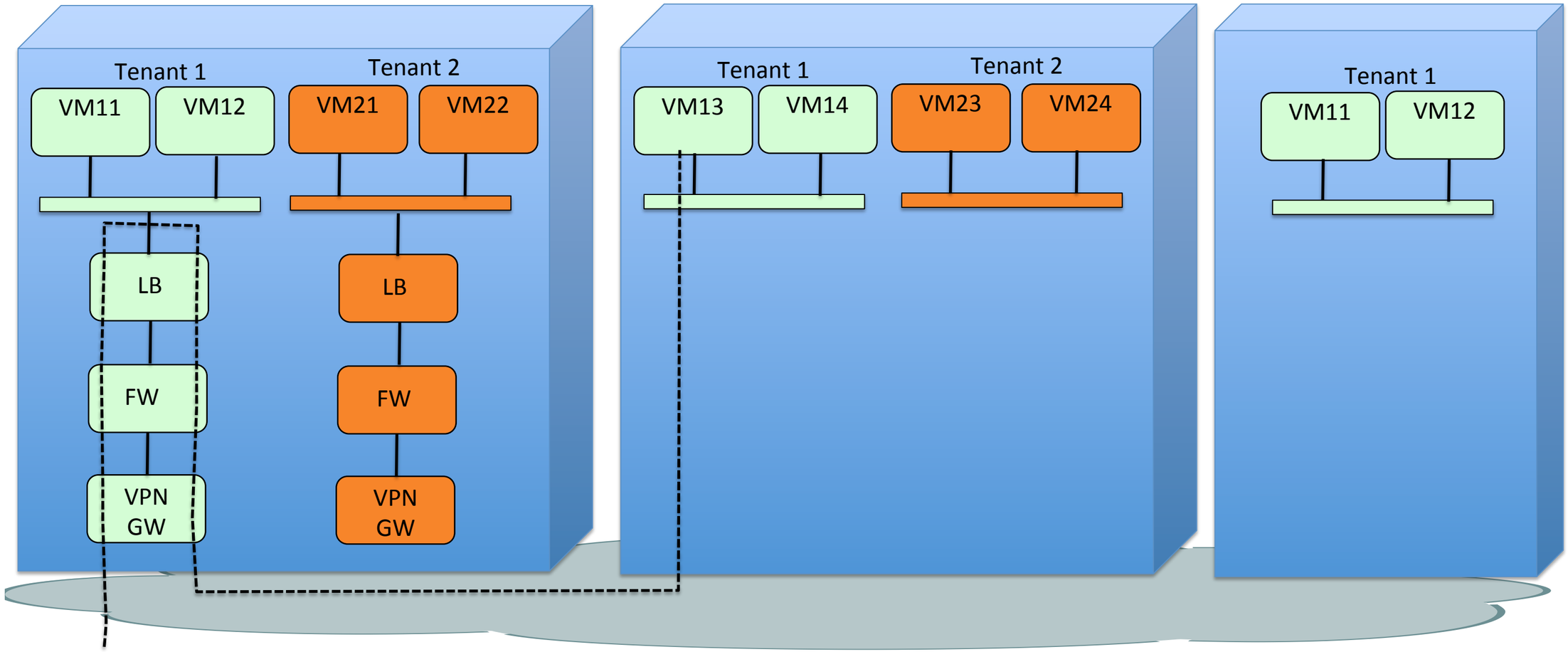
Virtualized DC Tiers

- Classical DC architecture



- Virtualize Classical DC architecture and move to Cloud
- Web, App, DB Tiers virtualized
- Network Function Virtualization
  - Firewalls, App/Web Load-balancers, VPN Gateways, L3 Functionalities

Network Function Virtualization (NFV)

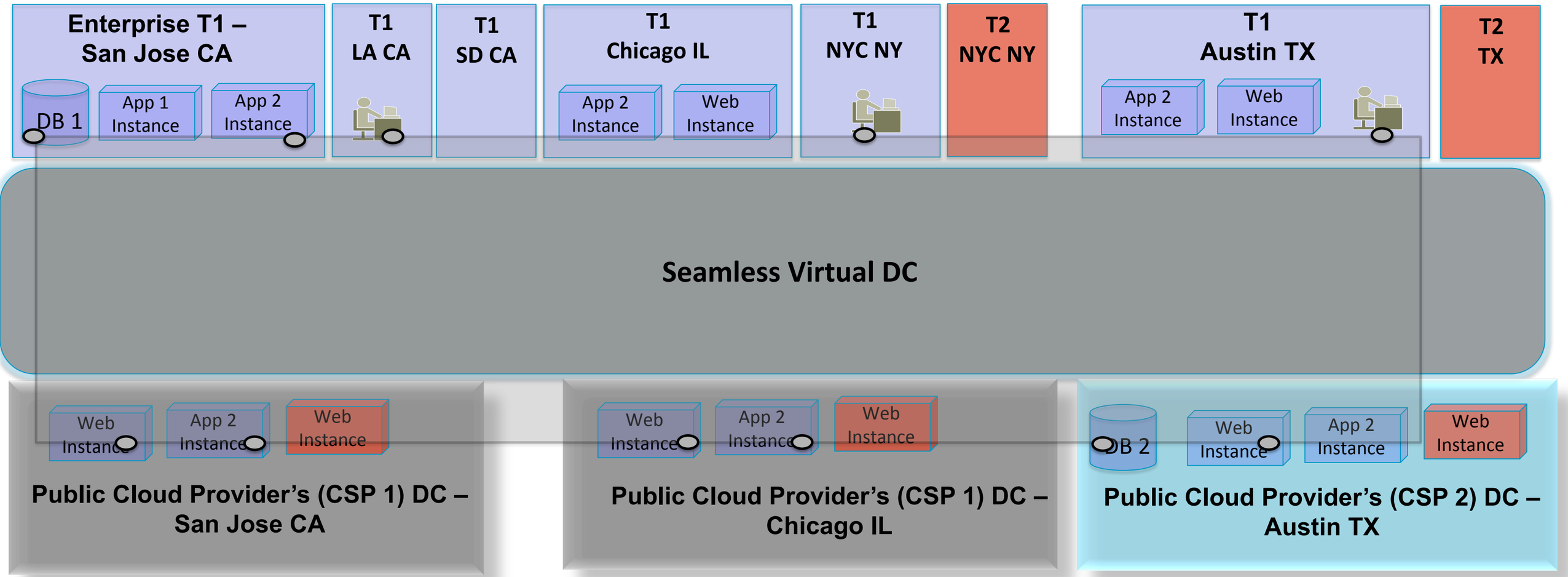


- NFV elements can be in VM (appliance) form factor and / or integrated embedded in Hypervisor and can be distributed

- Commercial NFV examples (Cisco):
  - CSR (Cloud Services Router) 1Kv – VPN GR, L3 Functionalities
  - ASA 1Kv – FW
- Openstack NFV capabilities

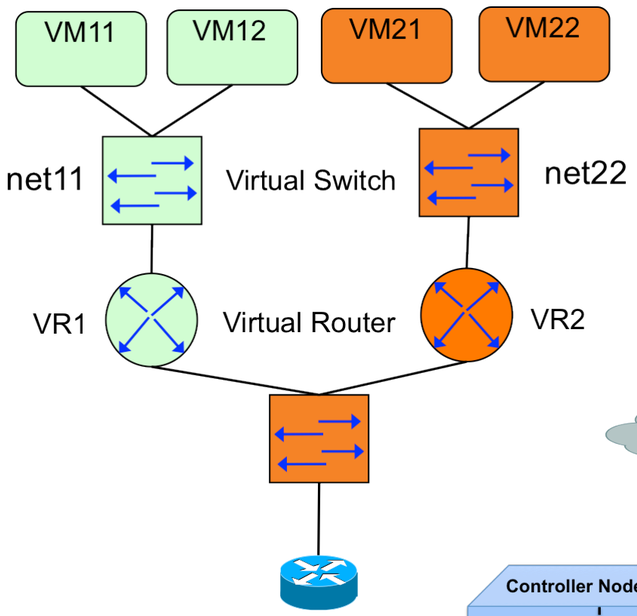
Take it to next level: Highly Distributed Cloud-based Virtual DC

- Virtualized Web / App/ DB Tiers and NFV distributed over multiple Cloud DCs / Sites and Multiple Clouds
- Since they are distributed Seamlessness should be provided → Seamless Virtual Cloud
  - Can have multiple instances of virtual DC / Seamless Cloud

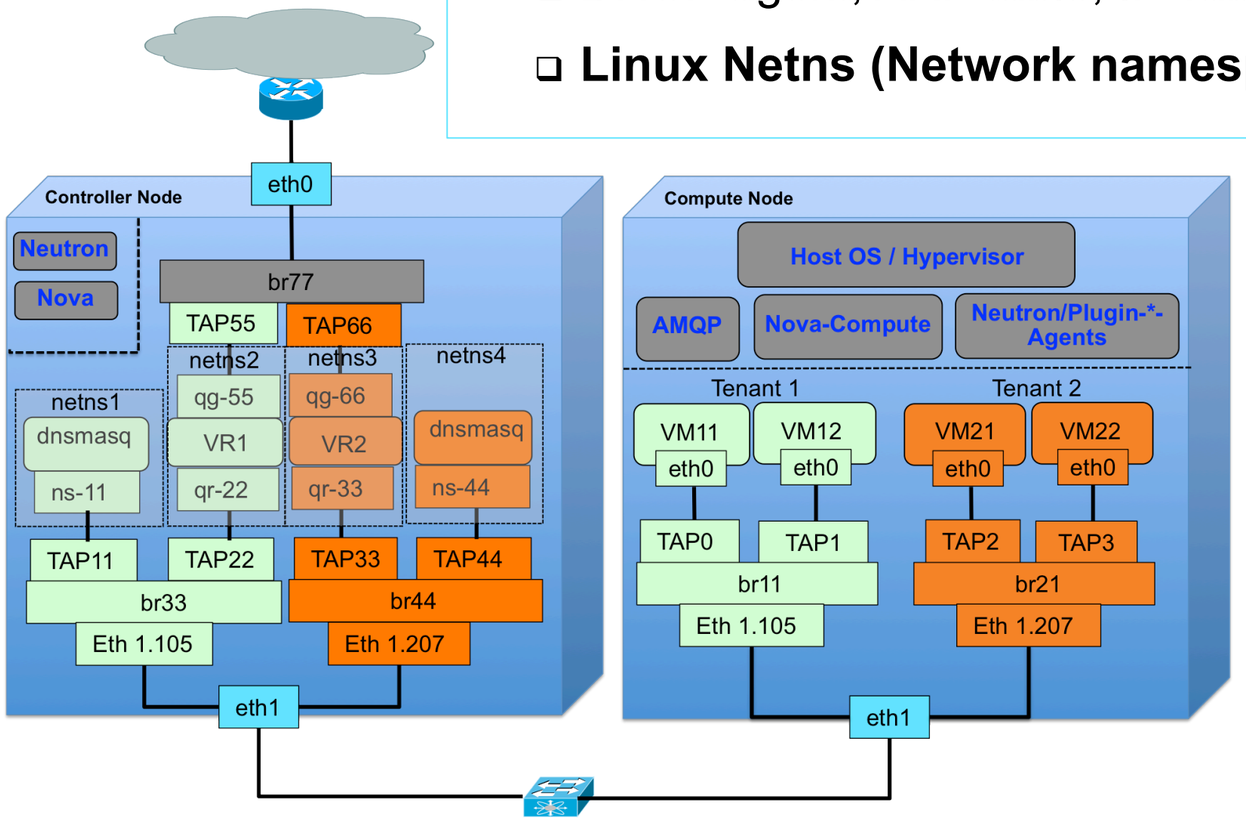


Cloud Networking:
End-to-end Multi-tenant Isolation

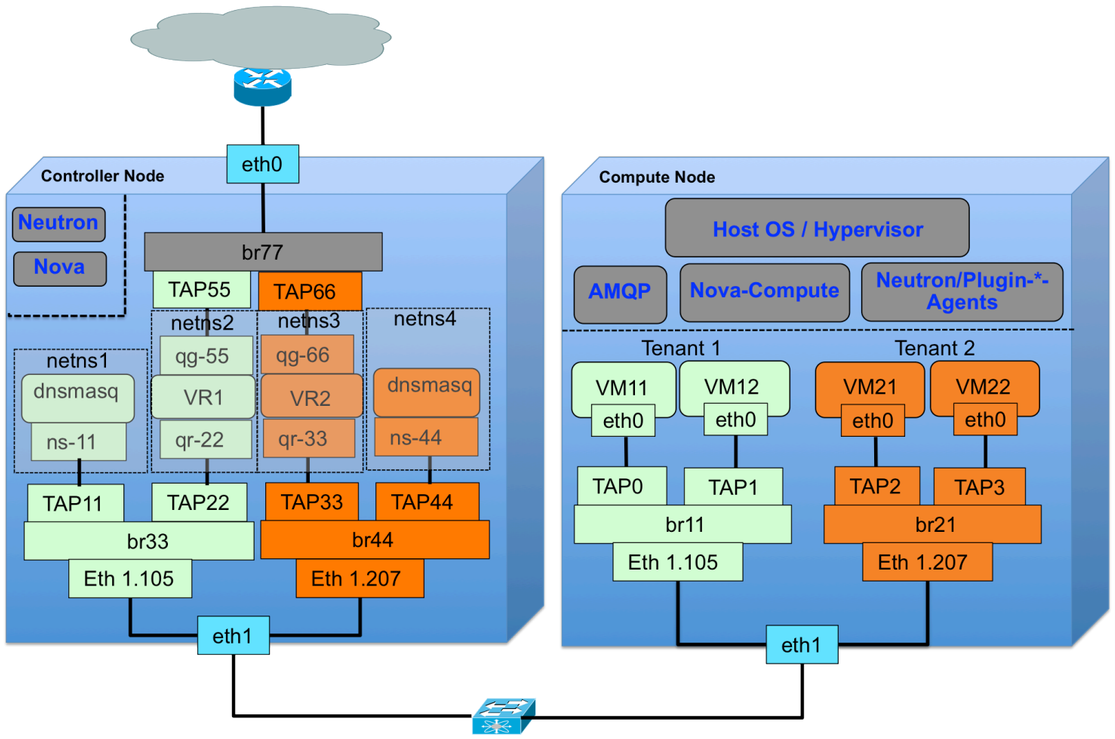
Isolation in Hypervisor



- VM provides Isolation
- Isolation should be E2E
- Various resources that used should be isolated within the Host / Hypervisor
  - DHCP agent, ARP table, IP Tables (for VR)
  - **Linux Netns (Network namespace)** can be used



Isolation in Hypervisor



- Physical interfaces and other resources reside in Default or Global NS, such as TAP11 and br33
- ip netns add netns1
- ip link add ns11 type veth peer name TAP11
- ip link set ns11 netns netns1
- ip netns exec netns1 ip link set dev ns11 up
- ip netns exec netns1 ifconfig ns11 10.10.11.1/24 up

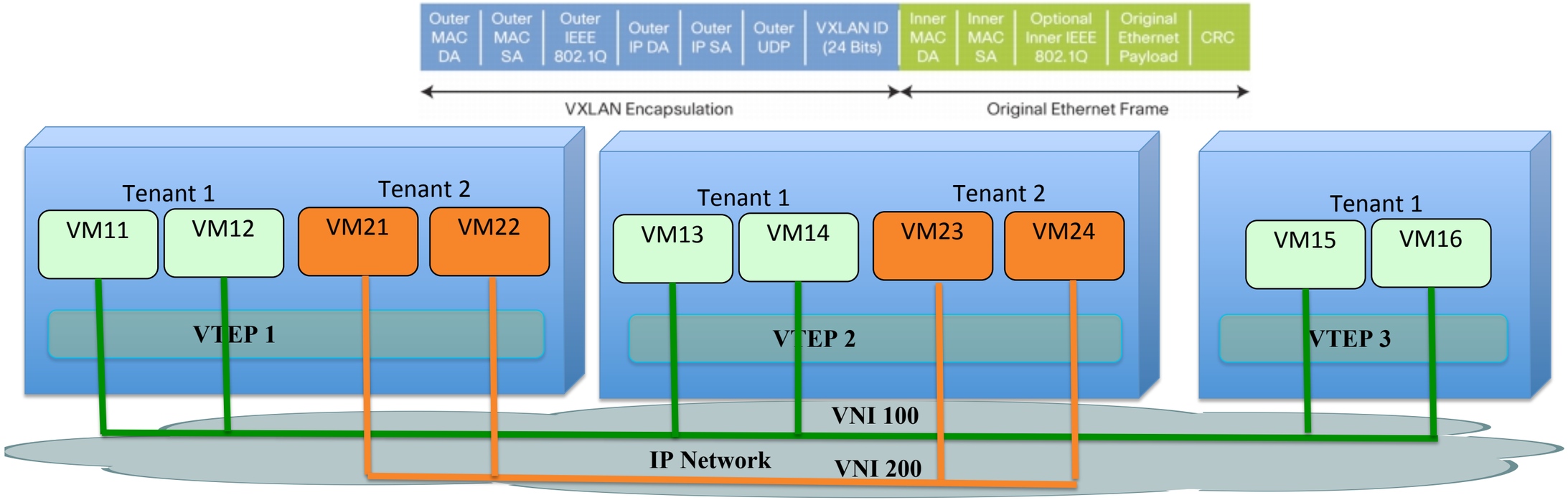
Isolation in DC Access Network

- 802.1Q VLAN → 12 bit ID → max 4K VLAN → does not scale in multitenant Public Cloud
- VXLAN (Virtual Extensible LAN) is one solution to address scalability for multitenant isolation in DC access network

VXLAN

- VXLAN: 24 bit ID → 16 million VXLAN segments
- MAC in UDP over IP multicast network – L2 Overlay network over L3 IP Network
  - Thus VXLAN segments can span over L3 network

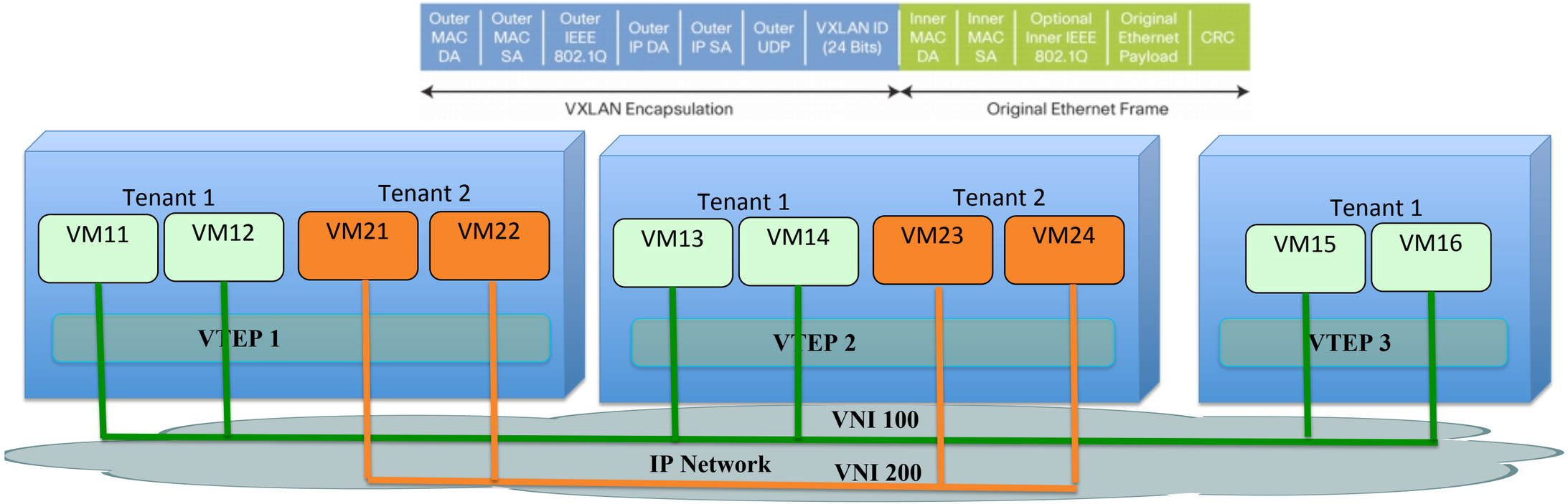
- Each Server joins a IP multicast group (IPMC) for BUM (broadcast, unknown, multicast)
- Because of UDP based can support ECMP (equal cost multiple paths) for bandwidth load balancing
  - UDP pseudorandom source port based hashing (by switches supporting 5-tuple based hashing)



VXLAN VTEP

- A VXLAN Tunnel End Point (VTEP) in hypervisor / virtual access switch
  - VTEP is assigned its own IP address → Outer MAC / IP identifies VTEP
- VTEP uses IP multicast to populate Forwarding Table

- When VMs communicate, VTEP performs following
  - VXLAN UDP encapsulation for VM-VM direct communication
  - IP multicast packet for ARP requests
  - Populates Forwarding Table in VTEP as needed
    - Source VM MAC, source VTEP IP/MAC, VNI association



VXLAN VTEP – Multicast Group

- VTEP uses IP multicast to populate Forwarding Table
- Multicast protocols should be supported in switches / routers: IGMP, PIM
- Each node belonging to a specific VXLAN segment (VS) joins a multicast group (MG)
- 1:1 correspondence to a VS to MG is not possible
  - Multicast address range 224.0.0.0 to 239.255.255.255
- One or more Transport VLAN used to carry VXLAN traffic!
  - If one VLAN, PIM not needed
- IGMP Snooping on L2 switches and Querier on L3 routers should be enabled
  - L2 switches snoops multicast traffic to find out which port subscribed to which MG

VXLAN VTEP- Forwarding Table Learning

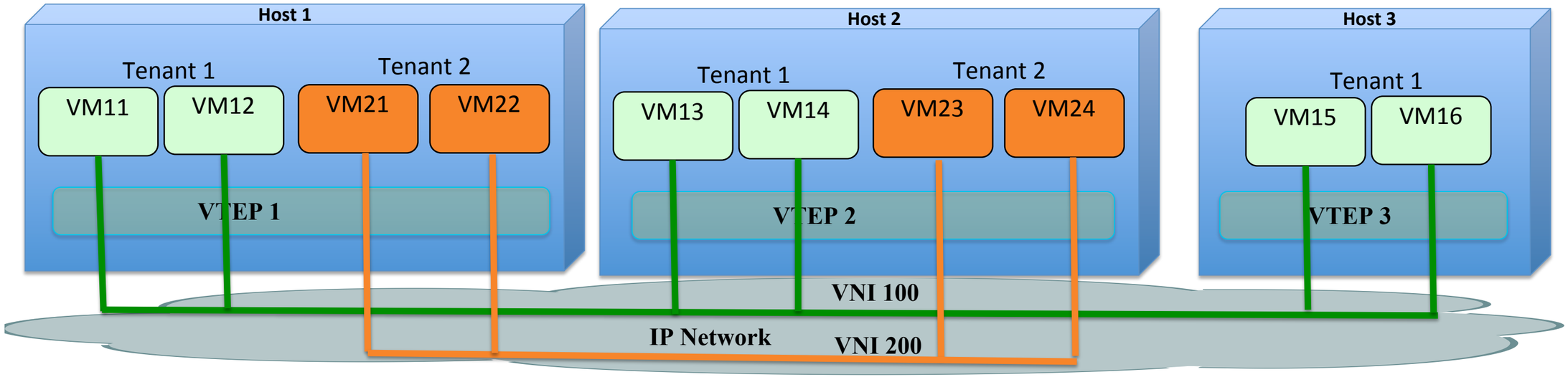
Dest: Multicast MAC Address	VTEP1 MAC	…	MG Address (like 239.1.1.10)	VETP1 IP Address	Including, UDP destination port 4789 …	VNI # …	FFF…FF (ARP broadcast)	VM11 MAC	….

Outer Encapsulation provided by VTEP | **Original Arp request sent by Host 1**

Host 2 and 3 learns

VM11 MAC	VTEP1 IP	VNI #	…
….			

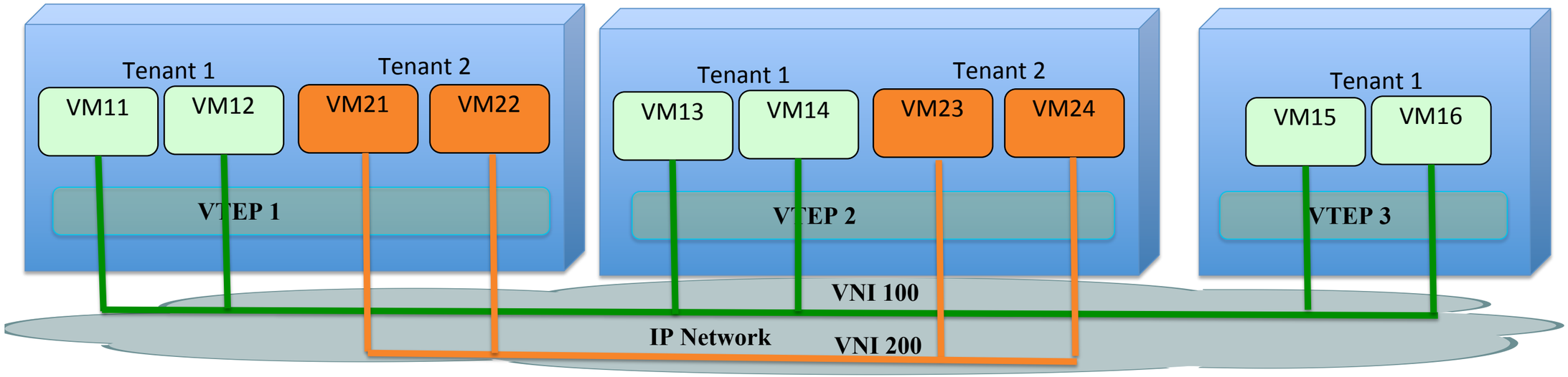




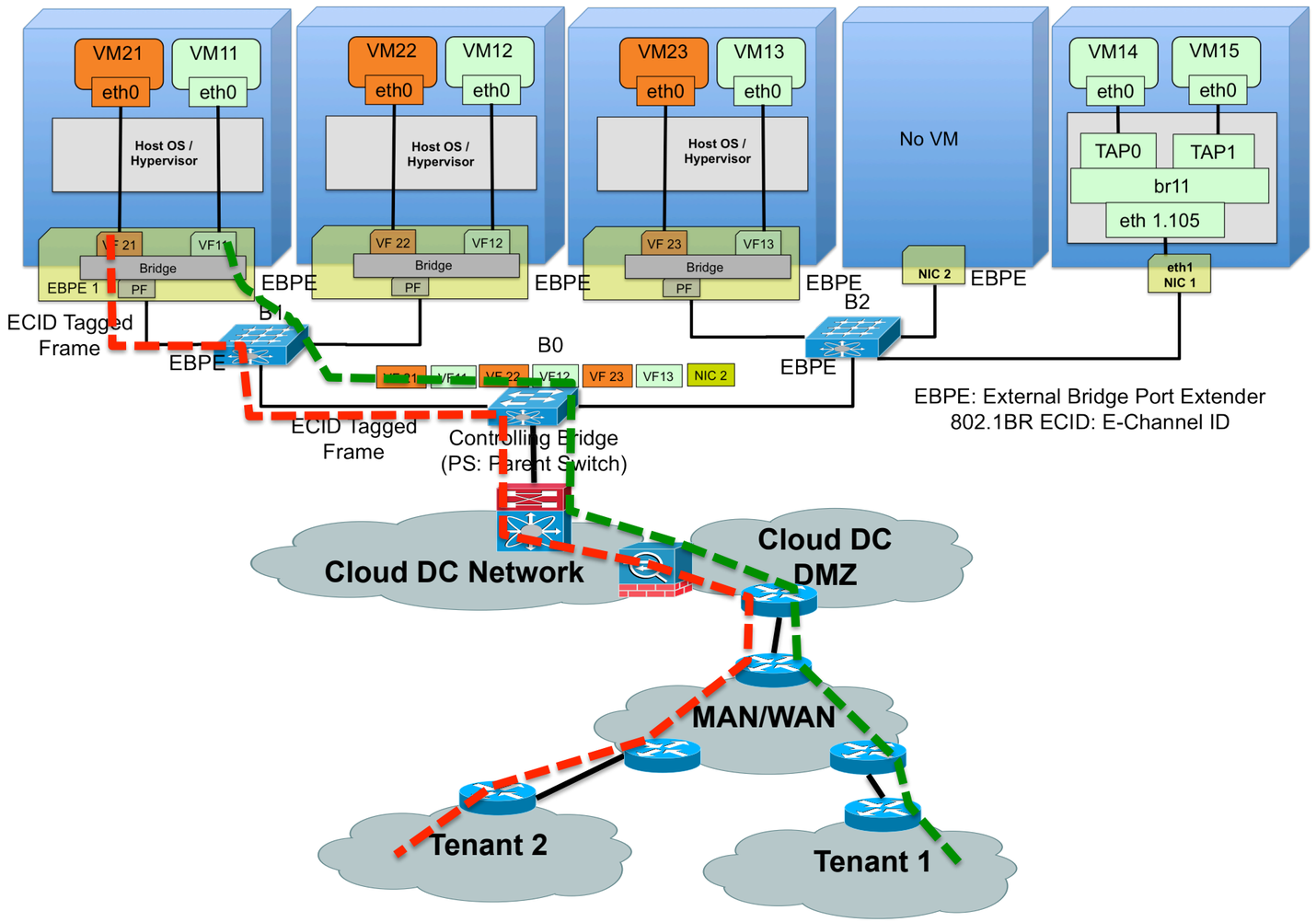
VXLAN Multitenant Isolation

- Each VNI or VXLAN segment provides multitenant isolation
  - From Host / Hypervisor / virtual access switch (VTEP) to host / hypervisor / virtual access switch (VTEP)
  - VTEP can be supported on any switch / router, not just hypervisor / virtual access switch





E2E Isolation in Network Segments



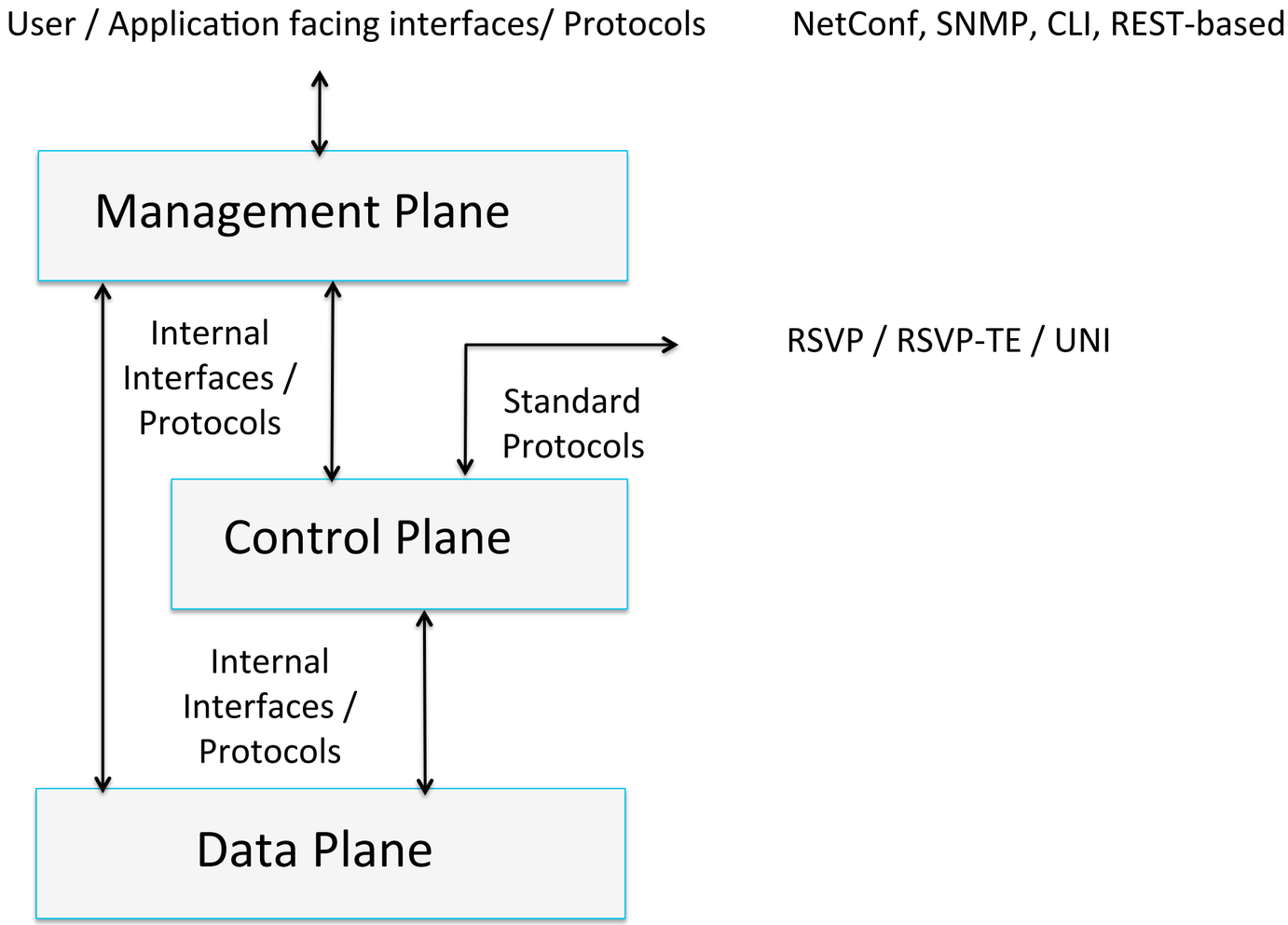
Isolation in Access → **VM / 802.1BR / VLAN/ VXLAN / GRE**

Isolation in Cloud DC up to DC Edge → **VLAN VRF-Lite Potentially VXLAN**

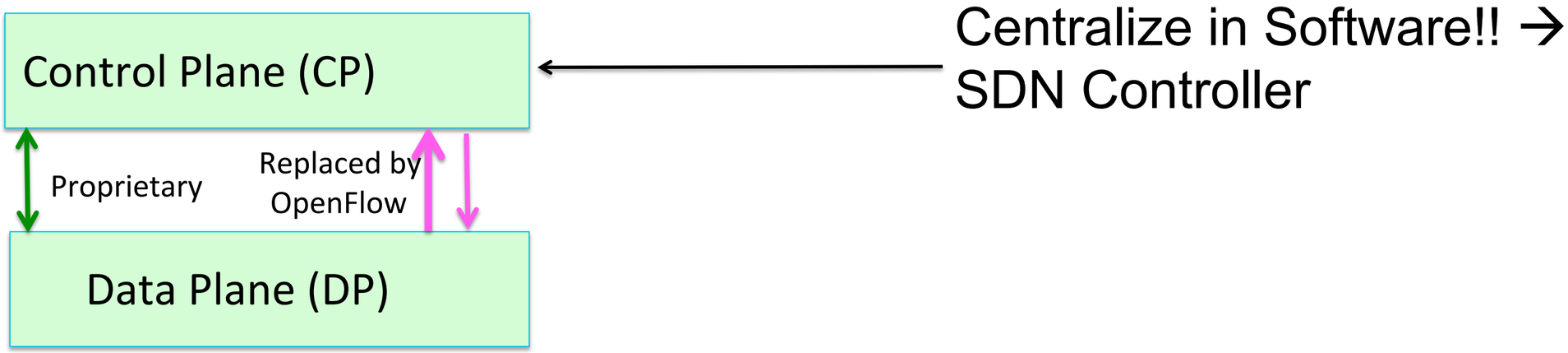
Isolation in MAN/ WAN up to Tenant Edge → **IPSEC Site-to-site MPLS VPN Open VPN**

Cloud Networking:
SDN / Programmable Networking

Existing Frameworks



SDN and OpenFlow – ONF Definition



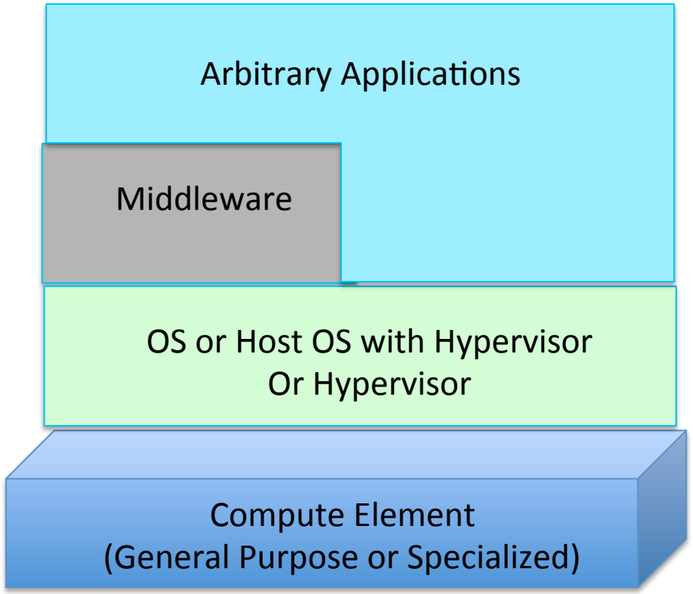
OpenFlow: "…open standard that enables researchers to run experimental protocols in campus networks. Provides standard hook for researchers to run experiments, without exposing internal working of vendor devices……"

http://www.openflow.org/wp/learnmore/

- Pretty much centered on dealing with Forwarding ("programming" forwarding path)
- There is more to networking than just forwarding

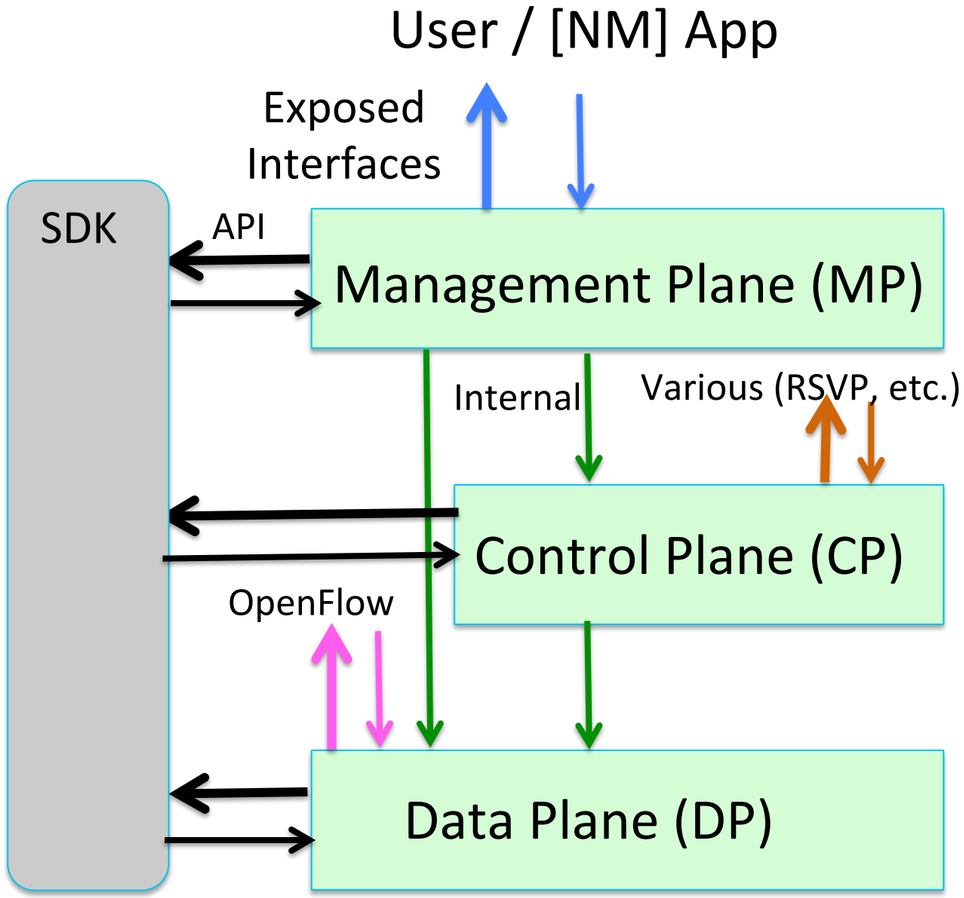
Programmable Networking

- SDN/OpenFlow: Major Focus: Flow-directed Forwarding path control
- ***Programmable Networking*** is the way to go
- Make existing capabilities programmable with new innovations
- Leverage best of the both worlds



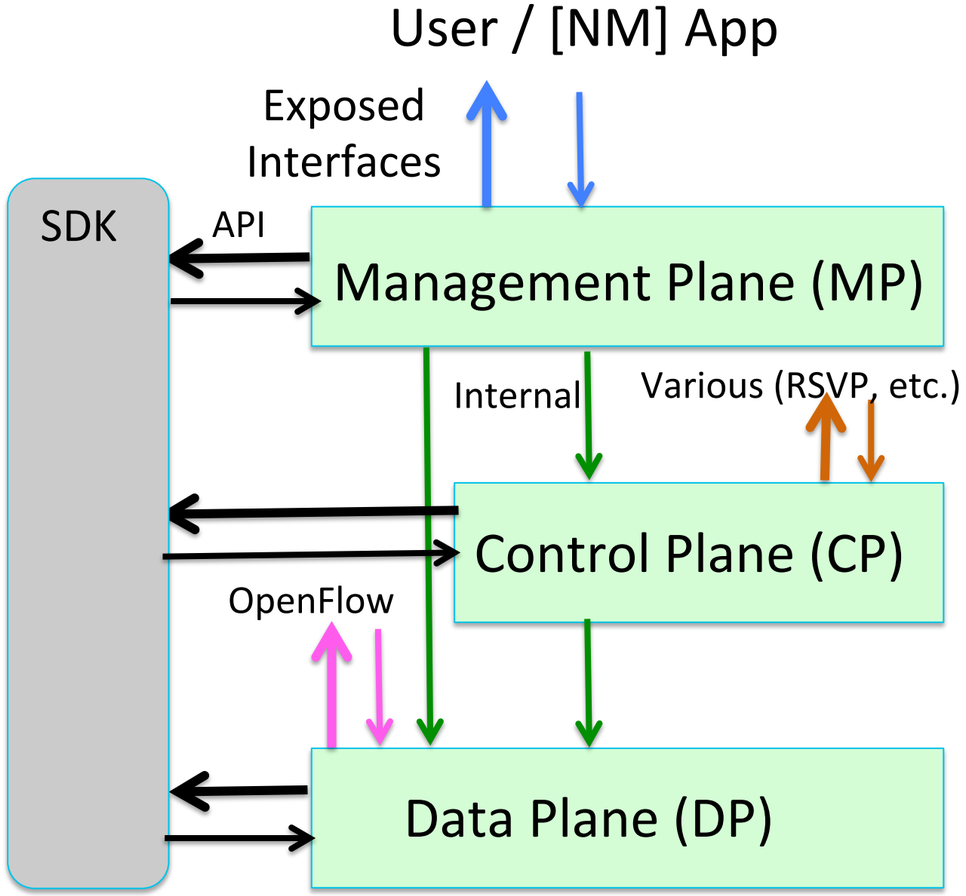
Programmable Networking with Programming Framework

- Consider Network as a computer or cluster and distributed DB
  - Program it
- Load and execute arbitrary programs on network: Examples:
  - Algorithms for new Longest Prefix Match
  - Heuristic algorithms for constraint based routing
  - Firewall ALG (Application Level GW) for a new application / proprietary protocol
  - Logical (overlay) topology for a tenant's Seamless Virtual Cloud



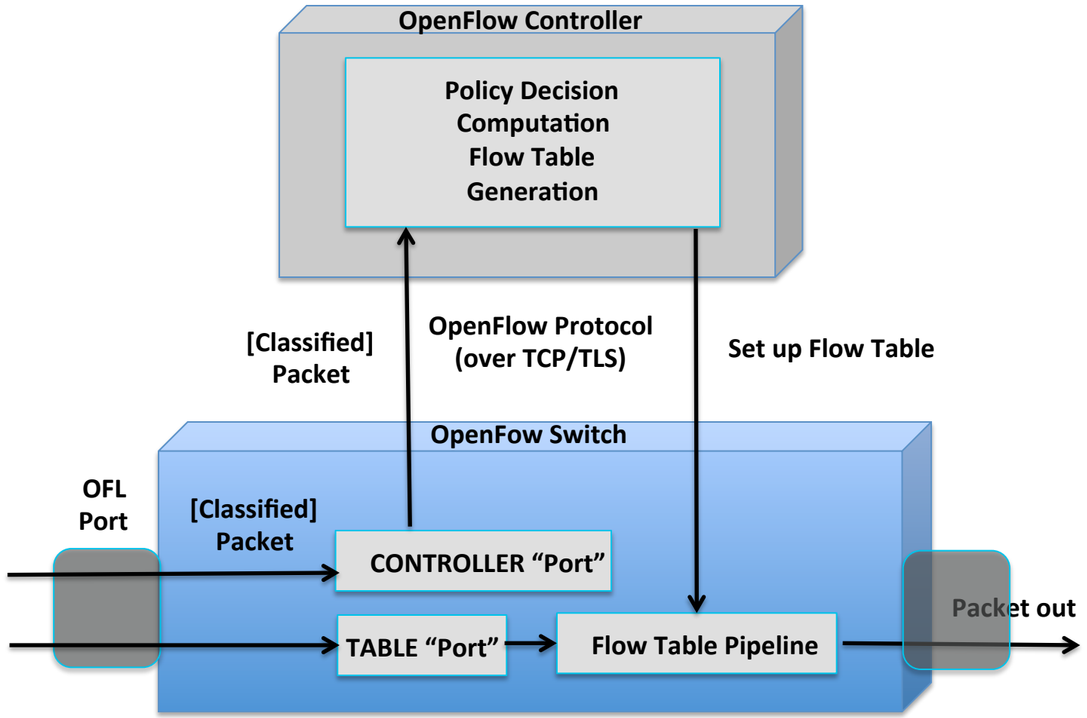
Programmable Networking with Programming Framework

- A full programming framework with published API and SDK support
- Developed programs can be run
  - Within (a user space) of these planes or network element (NE)
  - Outside of planes or NE in / as a Controller or Applications



SDN / Programmable Networking: OpenFlow

OpenFlow Switching



OpenFlow Switch Flow Switching Model (v1.1.0)

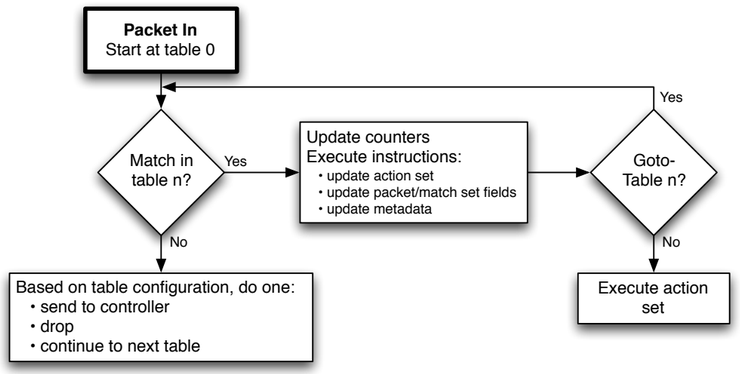
- Flow table entries (CRUD by Controllers)
- Multiple tables can be chained together for pipeline processing

Match Fields	Counters	Instructions

On

Count of Packets, Bytes, drop, error, etc.

Ingress Port	Metadata	Ether src	Ether dst	Ether type	VLAN id	VLAN priority	MPLS label	MPLS traffic class	IPv4 src	IPv4 dst	IPv4 proto / ARP opcode	IPv4 ToS bits	TCP/ UDP / SCTP src port / ICMP Type	TCP/ UDP / SCTP dst port / ICMP Code



From OpenFlow Sec 1.1.0 Implemented

Apply-Actions <action(s)>: Immediate actions
Ex. Output to port, Push VLAN, Set QoS, etc.

Clear-Actions

Write-Actions <action(s)>: specified actions are merged with current action set

Write-metadata <metadata/mask>

Goto-Table <next-table-id>

OpenFlow Hybrid Switch

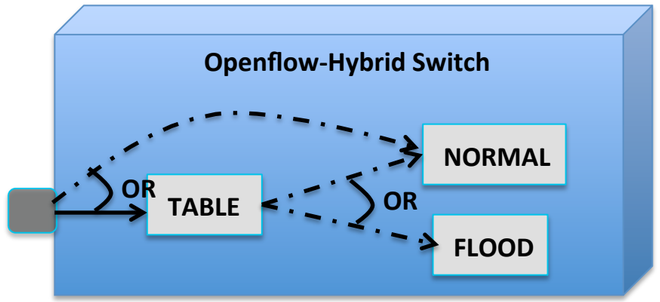
- OpenFlow-only switches support only OpenFlow operation
  - All packets are processed by the OpenFlow pipeline, and can not be processed otherwise

- OpenFlow-hybrid switches support both OpenFlow operations and typical (Ethernet) switching functions
  - Packet classification is non-OpenFlow based
  - Once classified, following options for processing

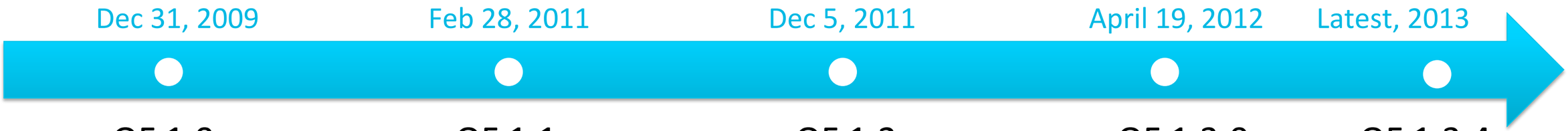
    OpenFlow Pipeline (sent to TABLE conceptual port)

    Non-OpenFlow pipeline (NORMAL)

    TABLE to NORMAL or FLOOD (for flooding)



OpenFlow Versions



OF 1.0

- Single Table
- L2, IPv4 focused matching

OF 1.1

- Multiple Tables
- MPLS, VLAN matching
- Groups: {Any-,Multi-}cast
- ECMP

OF 1.2

- IPv6
- Flexible-length matching

OF 1.3.0

- 802.1ah PBB (MAC-in-MAC)
- Multiple parallel channels between Switch and Controller

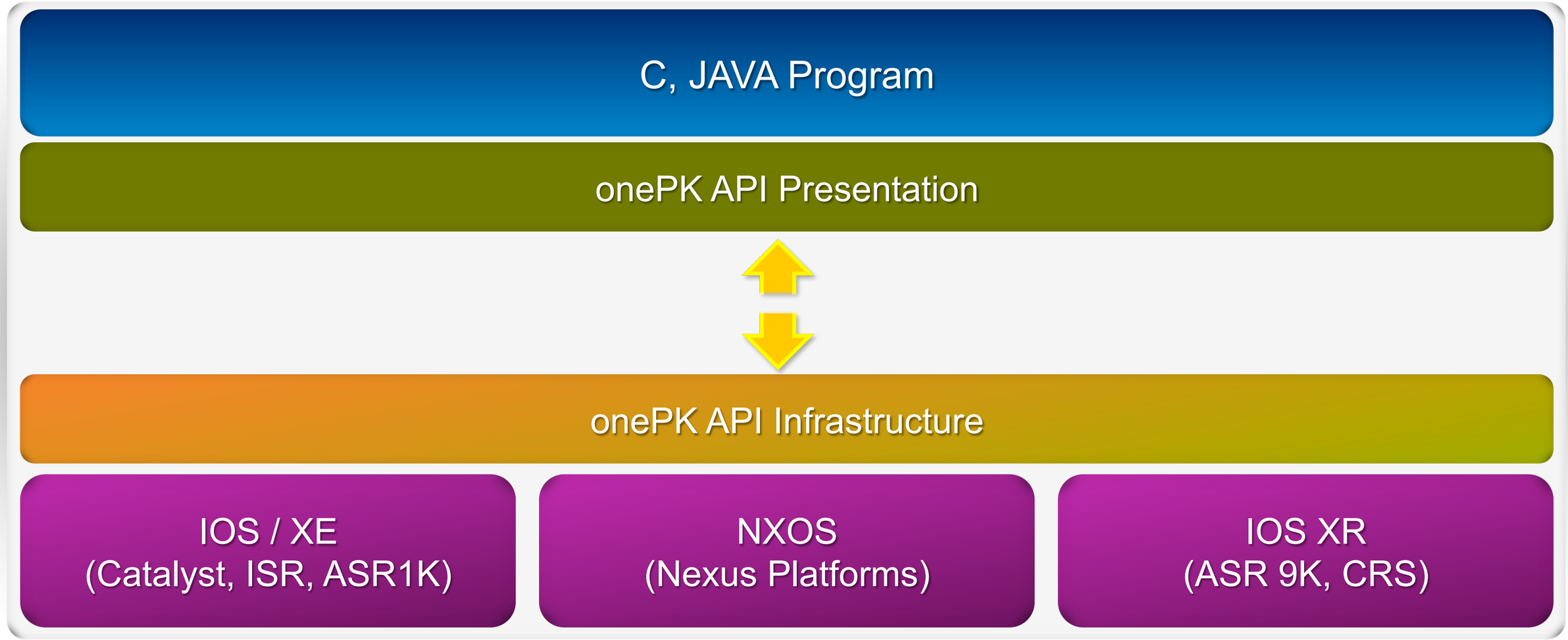
OF 1.3.4

SDN / Programmable Networking: Cisco OnePK

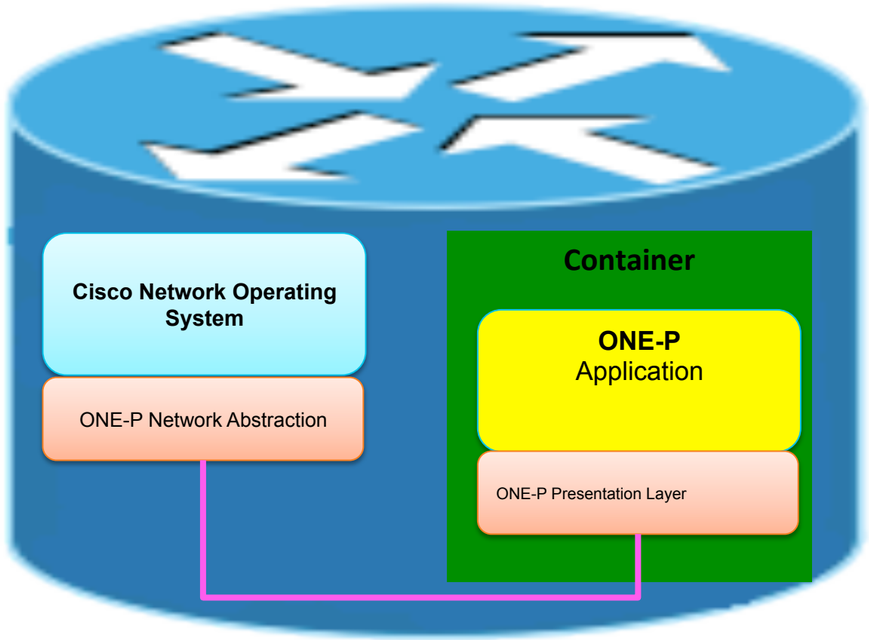
Introduction

- Cisco OnePK is a framework for programming the network
  - A network that is widely deployed with rich set of advanced features
  - It provides an abstraction and API to program against those rich features
  - There are multiple deployment models for OnePK programs

onePK Architecture

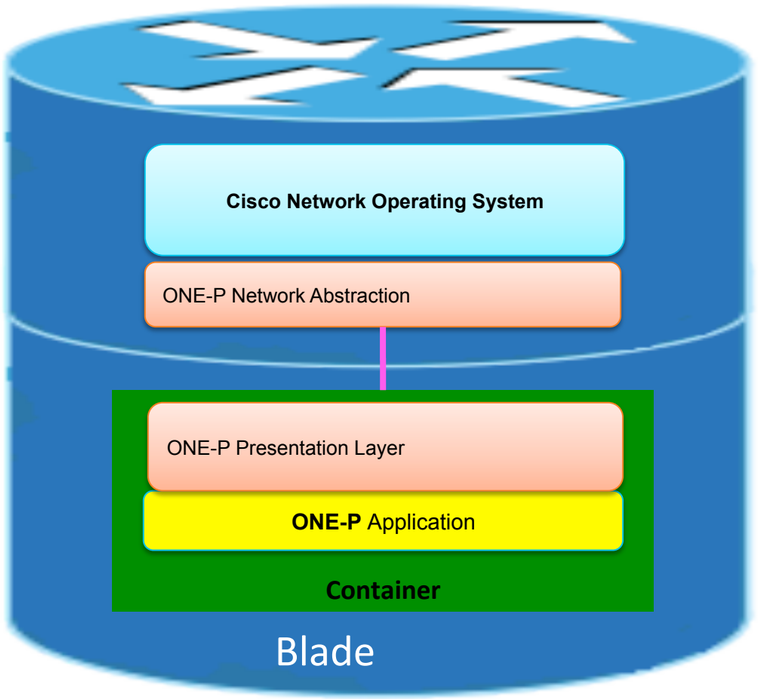


OnePK Application Deployment Model – Process Hosting Mode



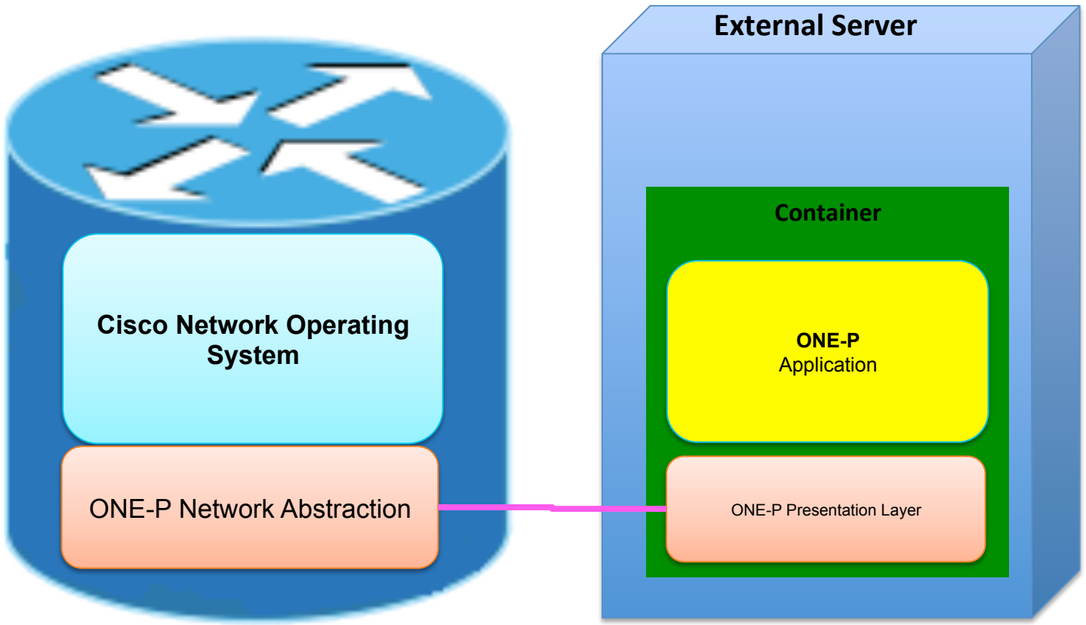
- Applications run within network element
- Applications are hosted within a container
- Low latency access to forwarding path
- Facilitate advanced application specific local processing / decision making without traversing network

OnePK Application Deployment Model – Blade Hosting Mode



- Applications run within a blade (such as a Linux blade) on network element
- Applications are hosted within a container
- Medium latency access to forwarding path
- Better isolation and more compute resources

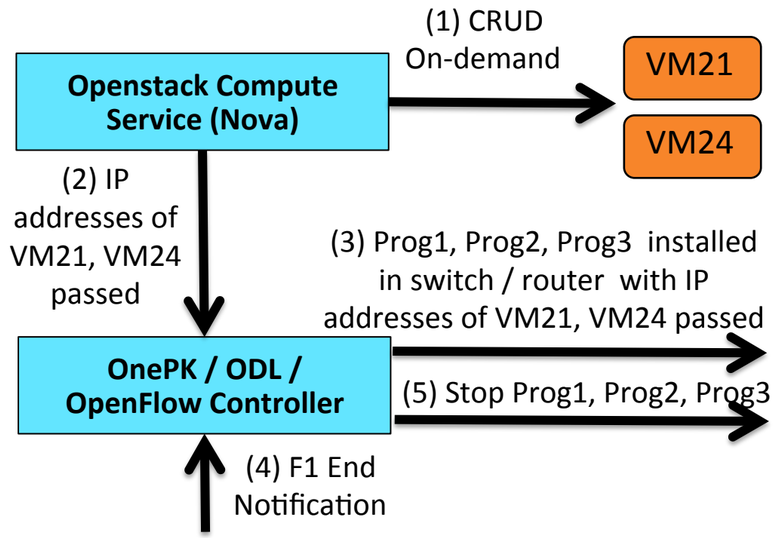
OnePK Application Deployment Model – End Node Hosting Mode

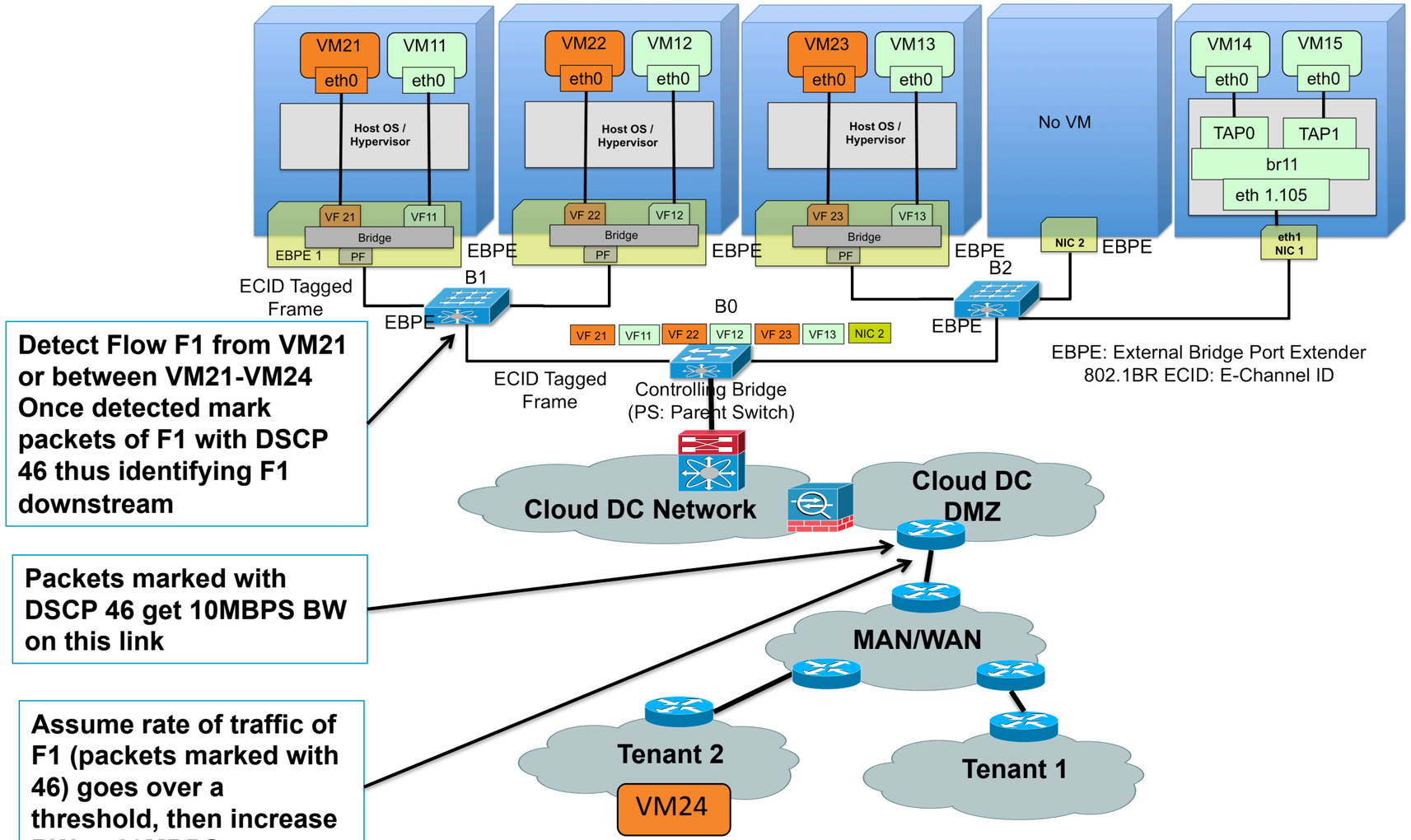


- Applications run on commodity devices (servers, laptops, mobile devices)
- Higher latency access to forwarding path
- High isolation and more compute resources

Cloud + SDN Use Case: Flow Directed Dynamic QoS

- Static configuration is not that useful in a Cloud environment
- Resources are CRUD on-demand
- Flow detection and QoS marking should be programmable → Prog1
- Similarly BW increase / decrease → Prog2
- VM21-VM24 info (such as IP address) will be passed to Prog1 dynamically by a Cloud Controller
- Detect when F1 ends → Prog3
- When the flow F1 ends, Prog1, Prog2, Prog3 can be terminated
- Prog2 can be get input from a complex program (traffic engg program)





Code Sample: App to NE session

- Create an App instance and connect to an NE to obtain session handle (for future communication)

```
/* Create App instance */
NetworkApplication networkApplication = NetworkApplication.getInstance();
networkApplication.setName(applicationName);
```

```
/* Get an NE instance by IP address of the NE */
InetAddress ipAddress = InetAddress.getByName(elementAddr);
networkElement = networkApplication.getNetworkElement(ipAddress);
```

```
/* Connect to the NE to establish session */
SessionConfig sessionConf = new SessionConfig(SessionTransportMode.SOCKET);
sessionHandle = networkElement.connect(username, password, sessionConf);
```

- Draft sample code – not 100% accurate

QoS Policy Programming Model

- **Filter**: Packet content is inspected based on Filter specification → also known as **Classification**

- **Action**: That are performed on packets or Queues (such as DSCP marking or bandwidth shaping) when Filter condition(s) satisfies

- **Target**: Policies are applied on a target, such as an NE interface

Code Sample: QoS Policy

```
import com.cisco.onep.policy.*;

NQP = new QosPolicy(elemA); private static QosPolicy NQP = null; private static NetworkElement NEI = null;
```

```
/* Create Filter Container – API term ClassMap; multiple ClassMap can be added to a QoS Policy */

class1 = new ClassMap (NEI, ClassMap.ClassOper.CLASS_OPER_OR);

NQP.addClass (class1, 10);
```

```
/* Create Filter and add it to a ClassMap: Filter by VM_IP passed by Compute Controller*/

FL1= new FilterProtocol (VM_IP);

class1.addFilter (FL1);
```

```
/* Mark packets with DSCP 46 for priority queuing in upstream interfaces and 10 MBPS BW guarantee on Target interface*/

A1 = new ActionMark (ActionMark.MarkType.ONEP_MARK_IP_DSCP, 46);

NQP.addAction (class1, A1);

A2 = new ActionClassQueue (10, OnepBandwidthUnits.ONEP_BW_UNITS_MBPS);

NQP.addAction (class1, A2);
```

```
/* Policy NQP is applied on output of interface e0/1 of NE */

Target target = new Target (NEI.getInterfaceByName("e0/1"),
Target.Location.ONEP_TARGET_LOCATION_HARDWARE_DEFINED_OUTPUT);

NQP.applyToTarget (target);
```

Draft sample code – not 100% accurate

SDN / programmable networking: Open Daylight

What is Project OpenDaylight?

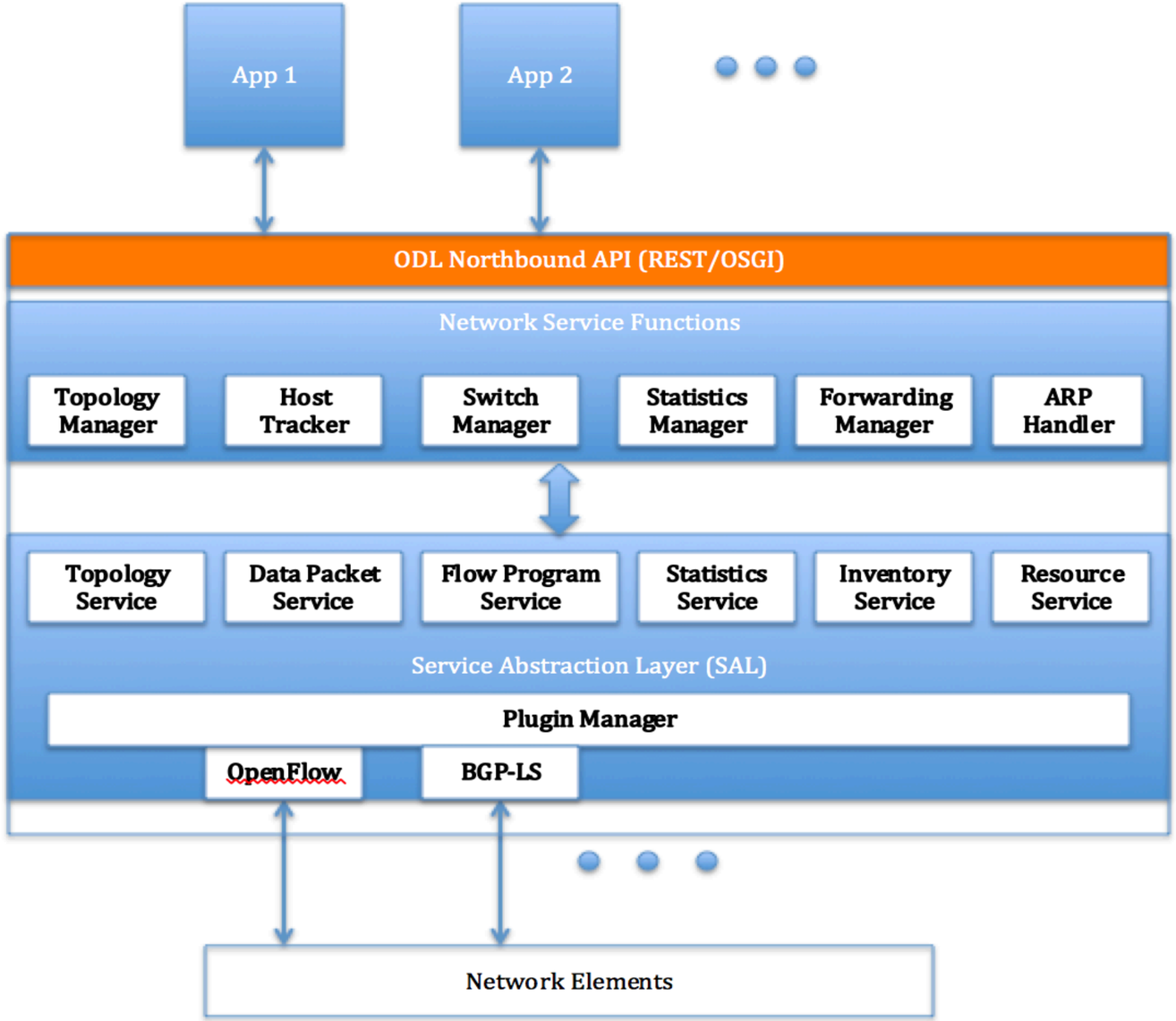
Open source project under the Linux Foundation

Furthering the adoption and innovation of Software Defined Networking (SDN) through the creation of a common industry supported framework.



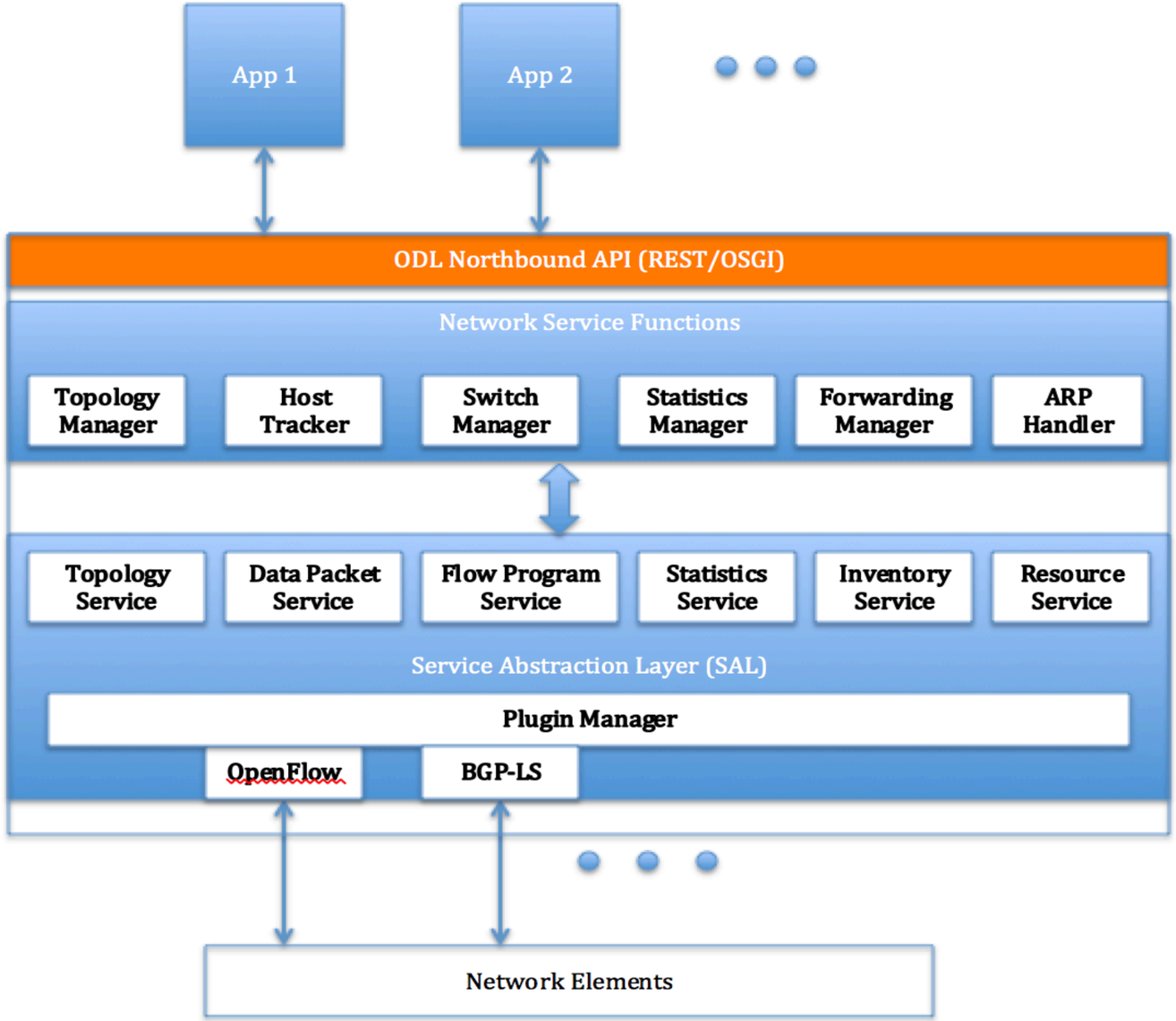
Members as of May 16, 2013 and growing

Open Daylight Controller Architecture – Service Abstraction Layer



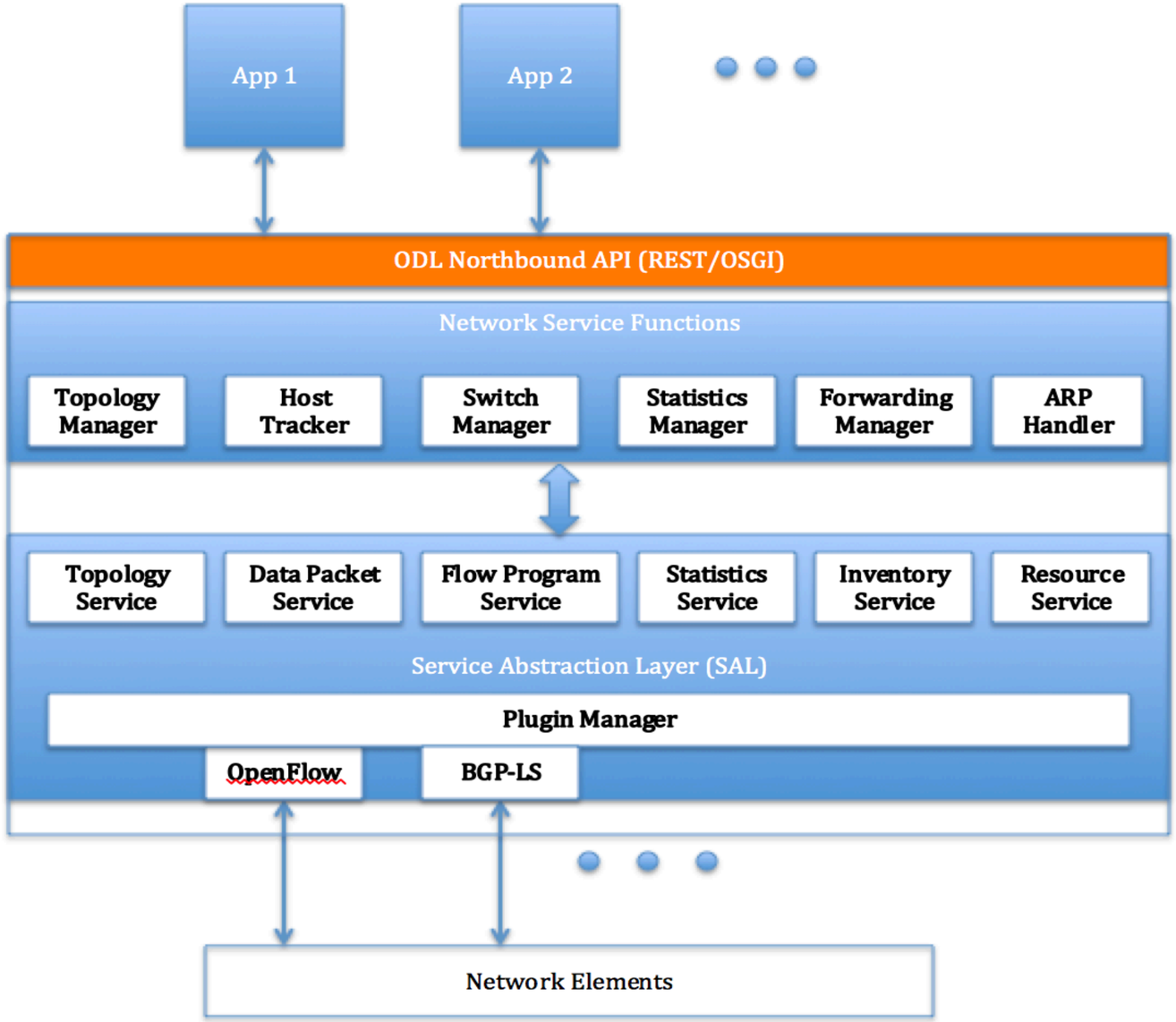
- A layer providing south-bound protocol and interface (SBPI) agnostic services to upper layers
- Each SBPI is incorporated in this layer as plugin
- Individual capabilities of each SBPI is abstracted

Open Daylight Controller Architecture – Service Abstraction Layer



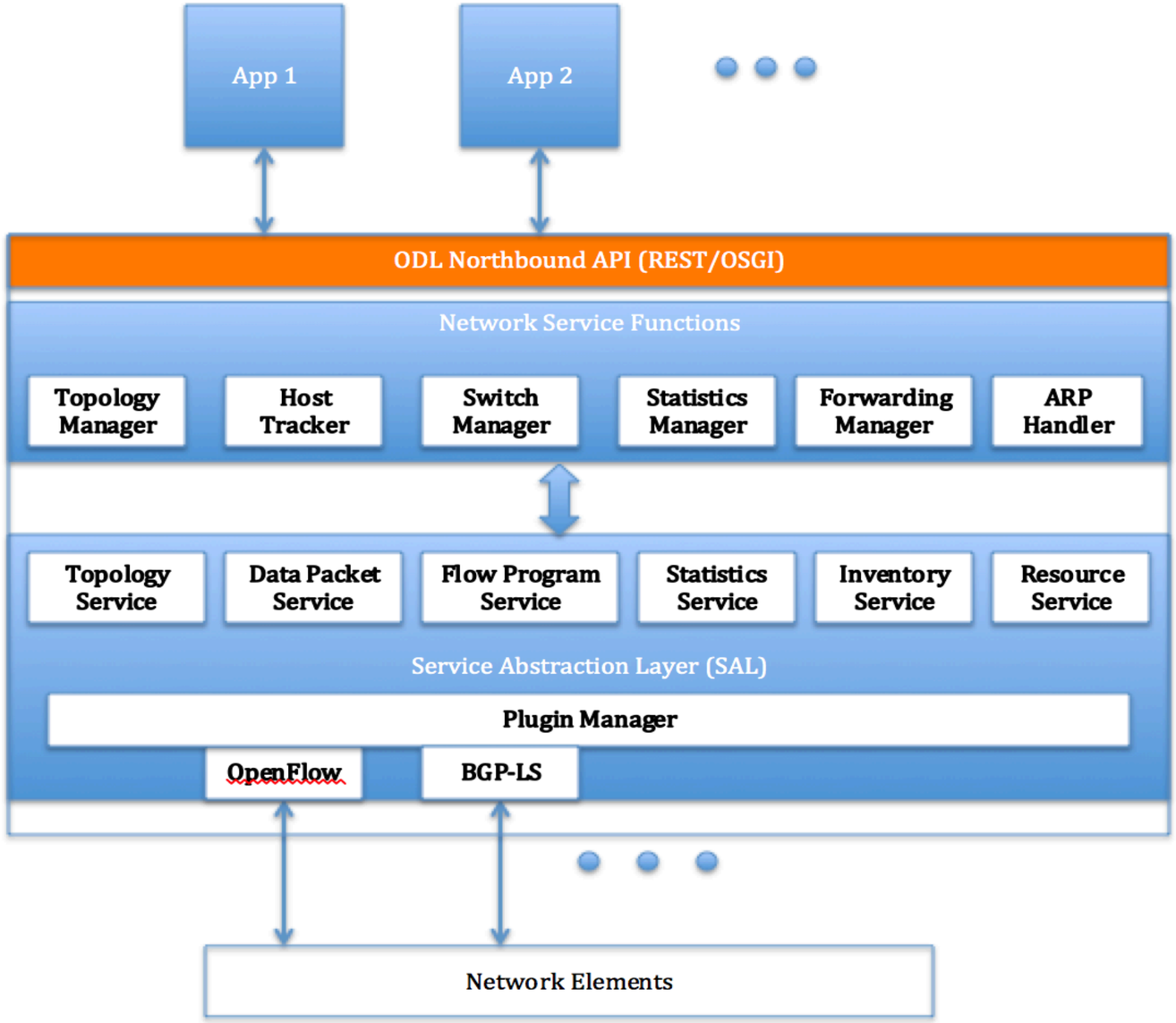
- Topology Service: To Discover nodes and links to be used to build topology
- Data Packet Service: To deliver packets from network devices to upper layer module (e.g. ARP Handler) or user applications
- Flow Programming Service – To program network elements with flow match/actions rules (such as based on OpenFlow)
- Statistics Service – To collect network statistics from network elements
- Inventory Service – To provide inventory information for network nodes and node connectors (ports)

Open Daylight Controller Architecture – Services/Business/ Orchestration Layer – Base Services



- Topology Manager: keeps track of discovered nodes and links
- Host Tracker: Tracks information on connected hosts
- Switch Manager: Information about network elements
- Statistics Manager: Statistics of the infrastructure
- Forwarding Manager: Deals with routes / forwarding tables
- ARP Handler: Deals with ARP
- Topology Manager makes use of information from other managers

Open Daylight Controller Architecture – NB API



- Applications outside of Controller can use REST based API to interact with Controller service
- Applications can also reside in the controller address space via the OSGi (Open Service Gateway initiative) framework
  - OSGi: Also known as dynamic module system for Java defines an architecture for modular application development

Open Daylight Flow Programming REST API Example

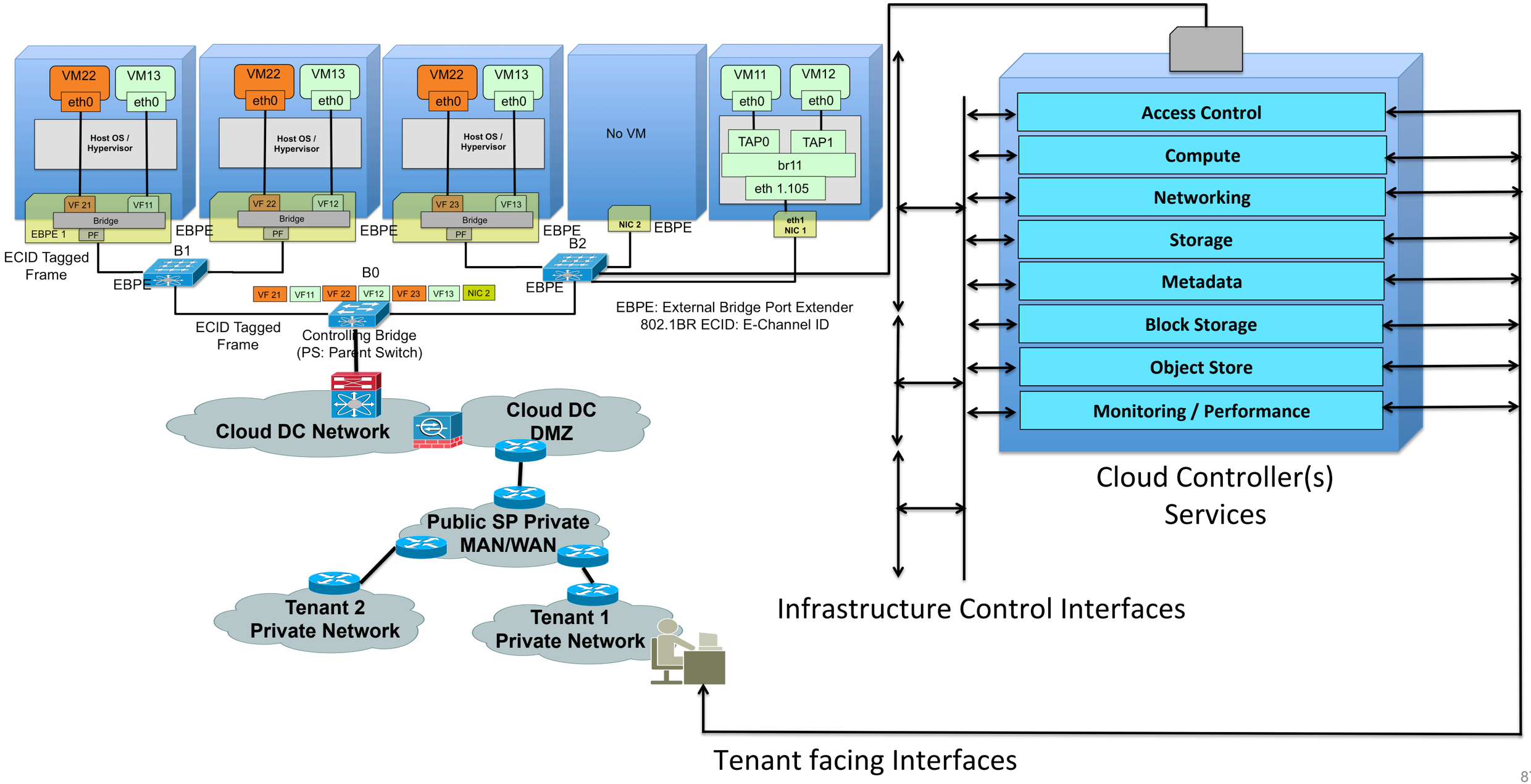
PUT http://localhost:8080/controller/nb/v2/**flowprogrammer**/default/node/OF/00:00:00:00:00:00:00:01/**staticFlow**/flow1

Request body:

```
{
   "installInHw":"true",
   "name":"flow1",
   "node":{ "id":"00:00:00:00:00:00:00:01", "type":"OF" },
   "ingressPort":"1",
   "priority":"500",
   "etherType":"0x800",
   "nwSrc":"9.9.1.1",
   "actions":[
      "OUTPUT=2”,
      “tosBits=46”
   ]
}
```

Cloud Framework Openstack

Cloud Controllers



Agenda

- General Introduction
- Openstack Services Introduction and Architecture
- Openstack Compute Service (Nova) Usage
- Openstack Networking (Neutron) Architecture and Usage

OpenStack: The Open Source Cloud Operating System

- A Cloud Platform
  - A collection of interrelated software components delivering capabilities to build and manage cloud infrastructure.
- A global community of developers devoted to innovation and openness
- Flexibility in deployment and features
- Standards for broad deployment
- No fear of vendor "lock-in"

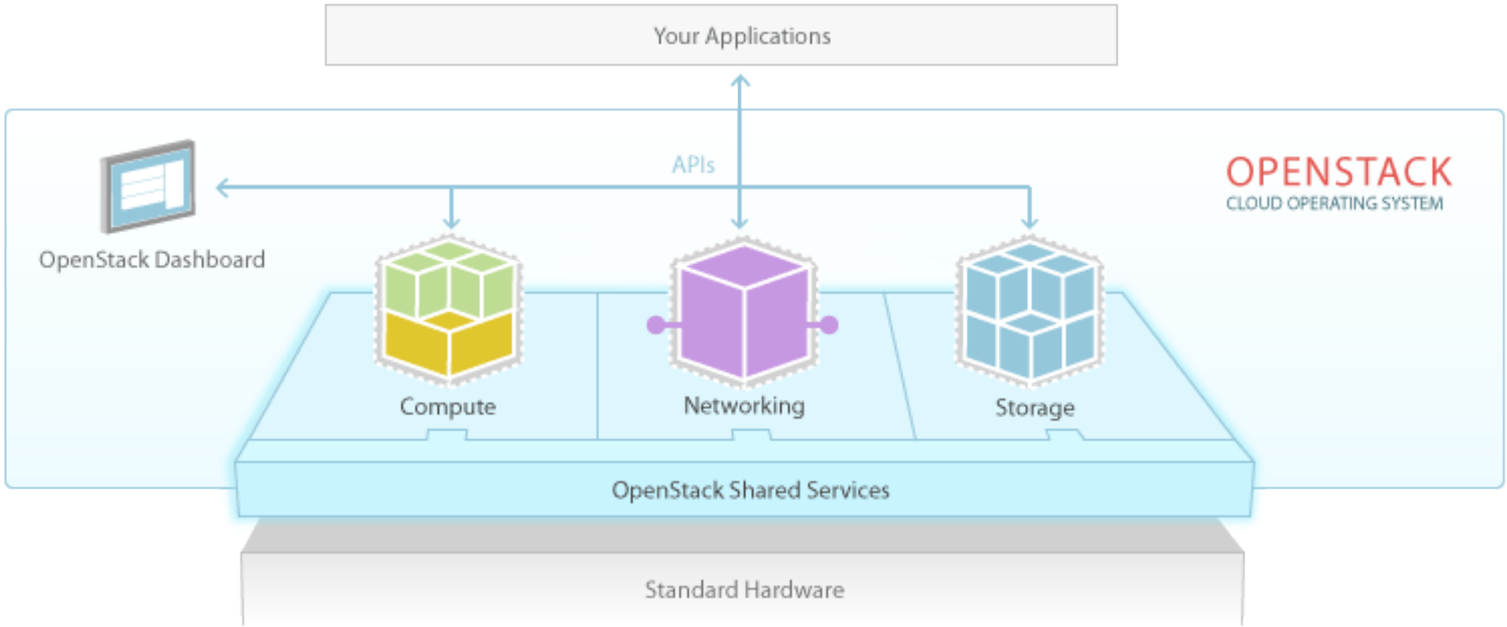


Release Cycle and Naming

- Austin: The first design summit took place in Austin, TX
- Bexar: San Antonio, TX (Bexar county).
- Cactus: Cactus is a city in Texas
- Diablo: Diablo is a city in the bay area near Santa Clara, CA
- Essex: Essex is a city near Boston, MA
- Folsom: Folsom is a city near San Francisco, CA
- Grizzly: San Diego, CA: Grizzly is an element of the state flag of California
- Havana: Havana is an unincorporated community in Oregon
- Icehouse: Ice House is a street in Hong Kong
- Juno: Juno is a locality in Georgia

https://wiki.openstack.org/wiki/Release_Naming

Openstack High Level View



Core Projects



Computing

Nova: Software to provision virtual machines on standard server hardware at massive scale

Glance: Image services for discovering, registering, and retrieving virtual machine images



Networking

Neutron: Software to provide "networking as a service" between interface devices managed by other Openstack services



Storing

Swift: Object storage to reliably store billions of objects distributed across standard server hardware

Cinder: Software to provide persistent block storage to Openstack services

Common Projects

- OpenStack Identity (Keystone)
- OpenStack Dashboard (Horizon)
- Telemetry (Ceilometer)
- Orchestration (Heat)
- OpenStack Common Library (Oslo)

Identity Service (Keystone) Introduction

- An authentication and authorization (AA) system
- Organized into a group of internal services
- Provides a HTTP front-end to clients (components, users, etc.)
- Support for multiple back-ends
  - Allows Keystone to adapt to a wide-range of environments

Keystone Architecture

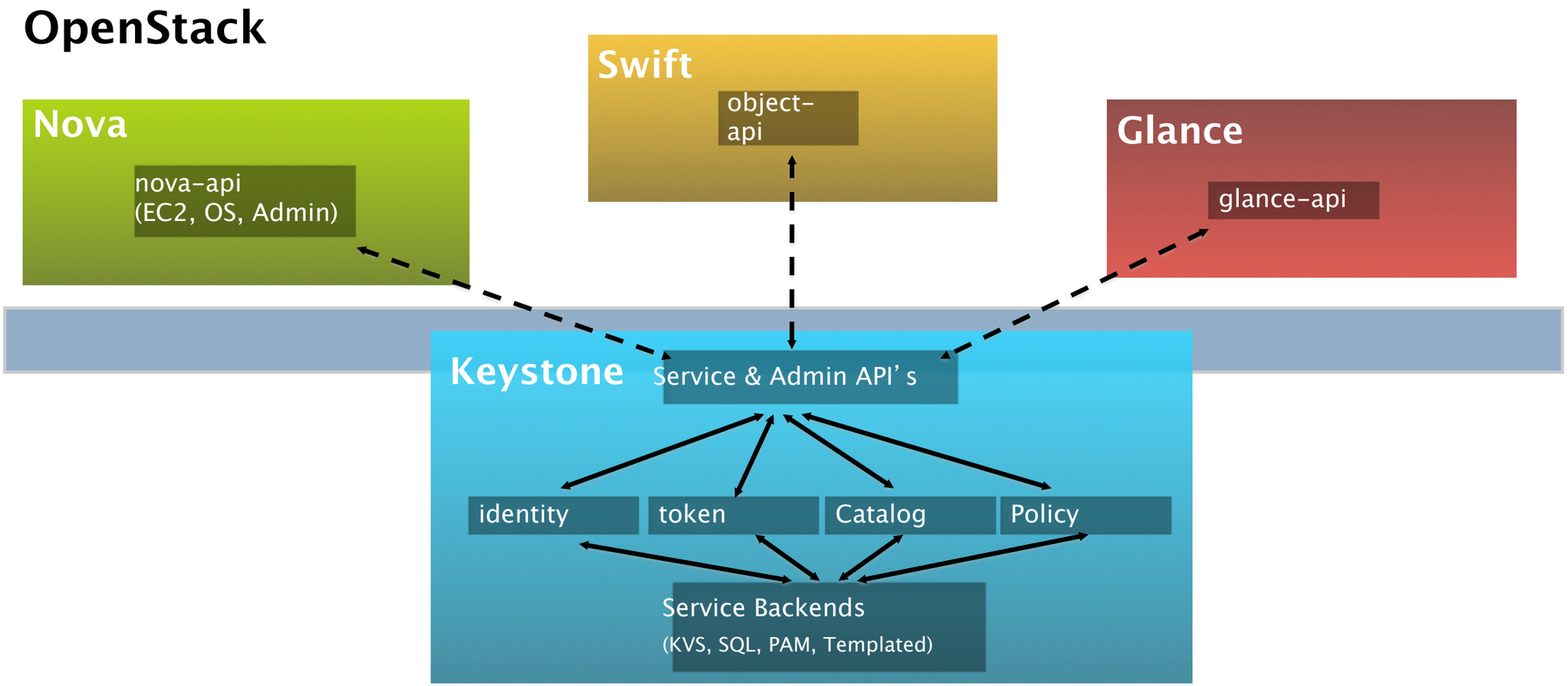
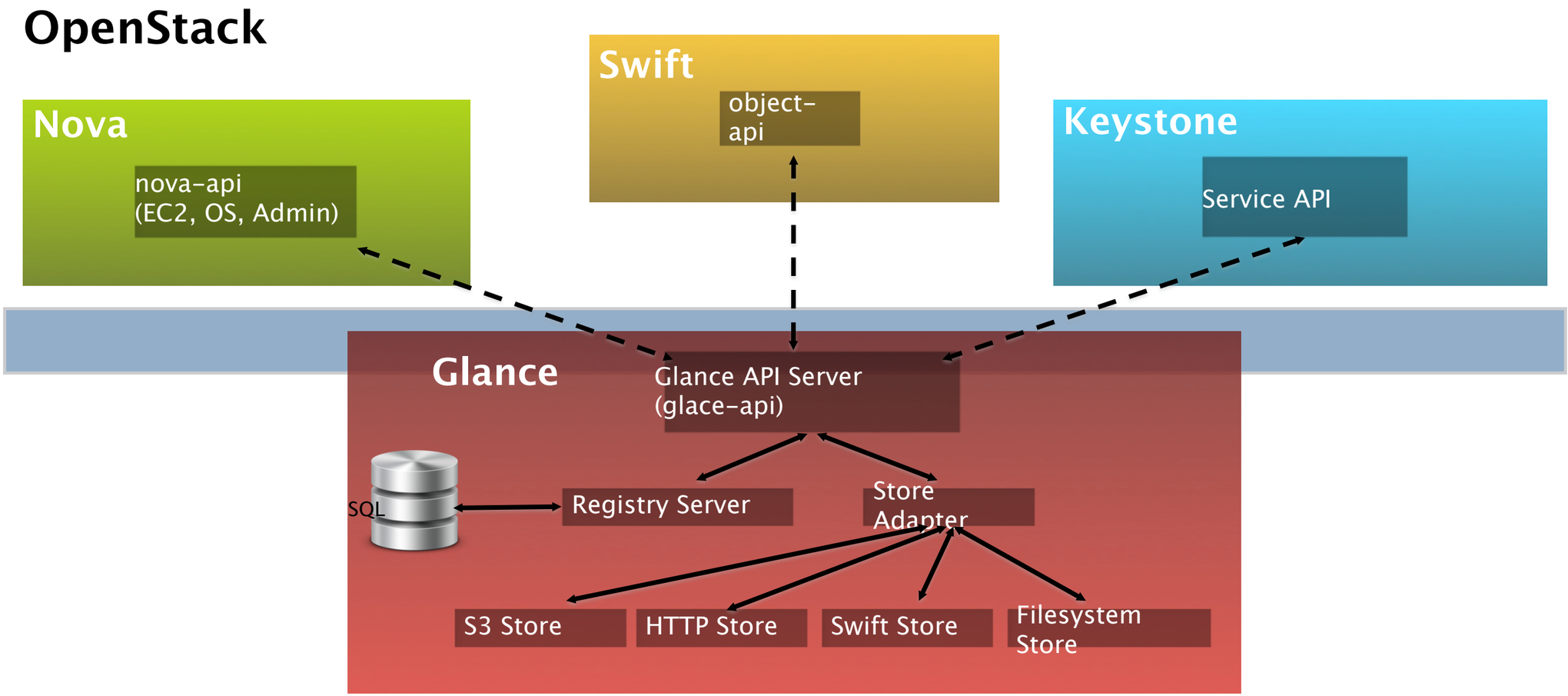


Image Service (Glance) Introduction

- Designed to be adaptable
- 3 primary services- API, Registry, and Store Adapter
- Provides services for discovering, registering, and retrieving virtual machine images
- Images can be stored in a variety of locations

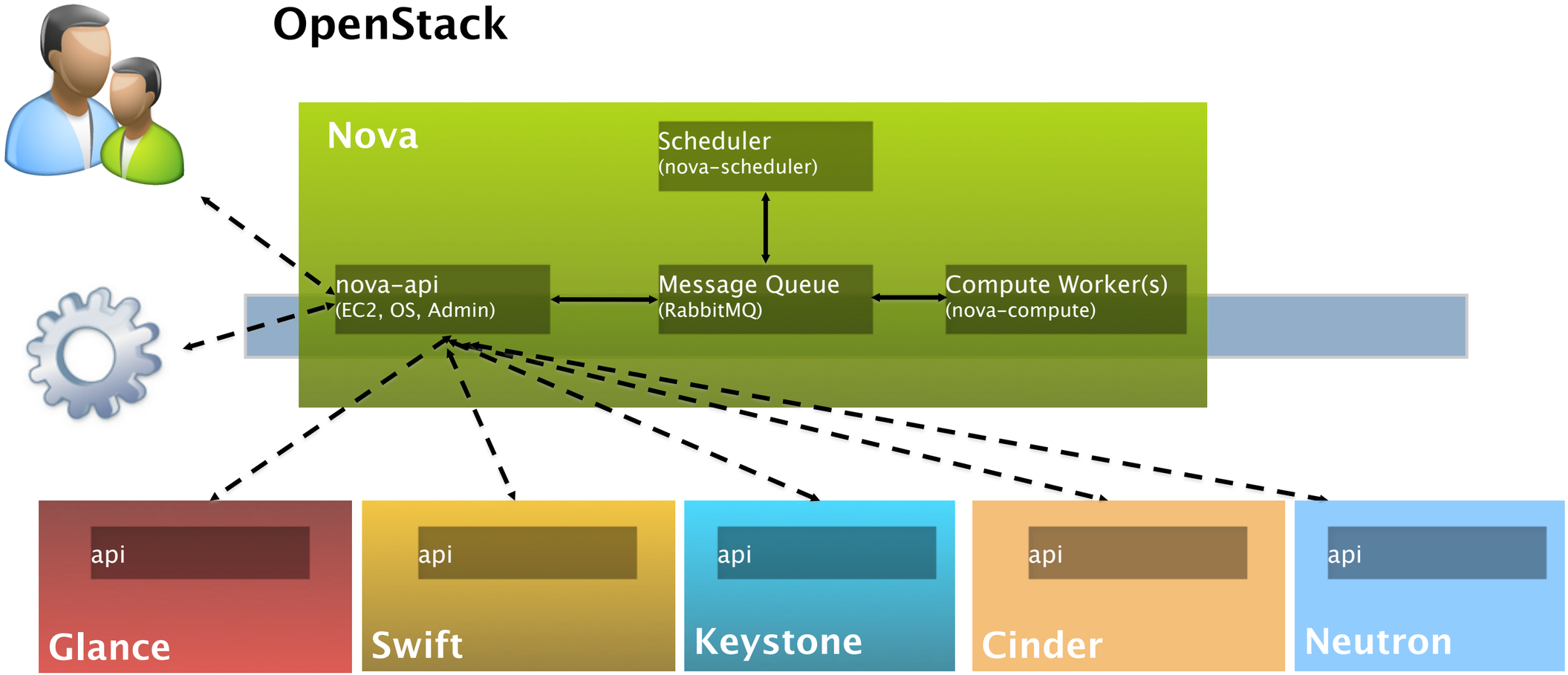
Glance Architecture



Compute Service (Nova) Introduction

- Primary component of the OpenStack IaaS platform
- Shared-nothing and modular software architecture
  - Services communicate using a message queue (AMQP)
- Uses a SQL-based database for storing information
  - Shared by all system components

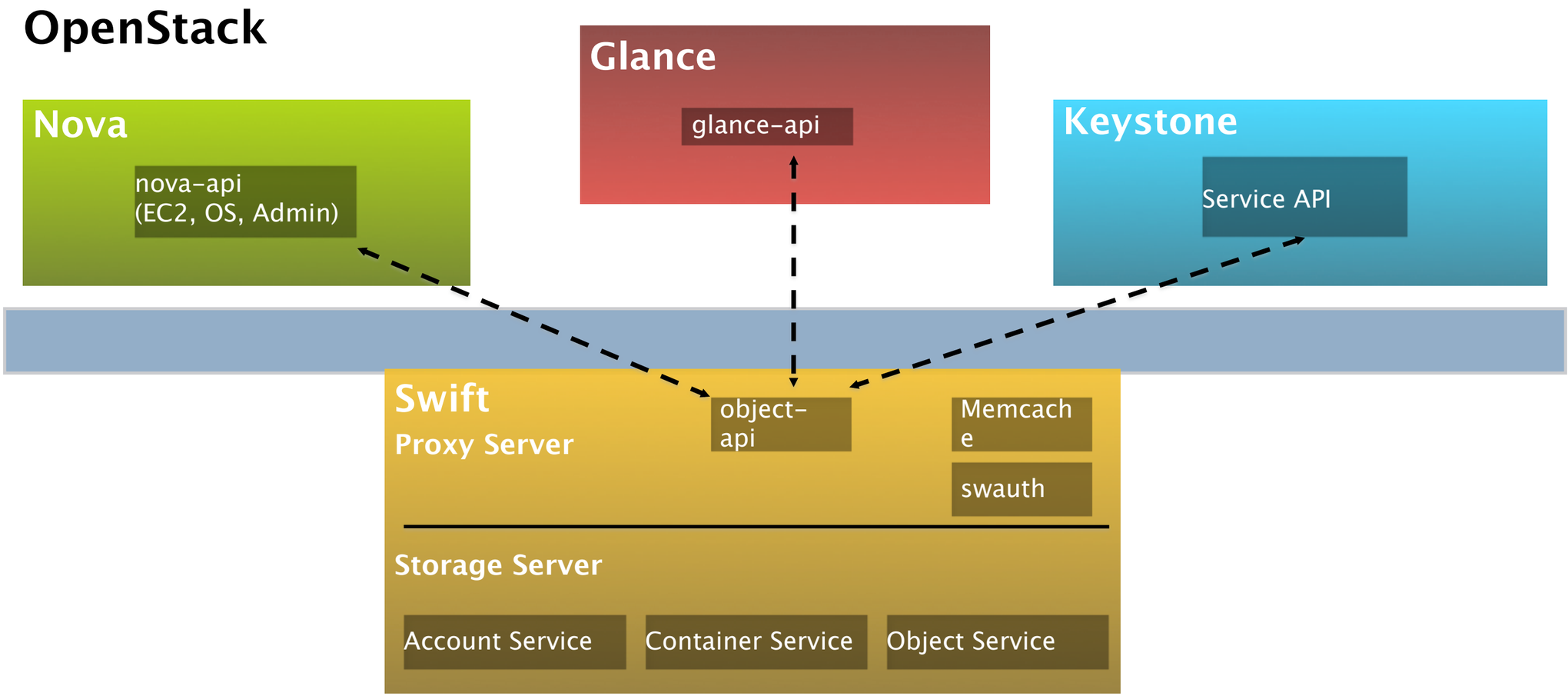
Nova Architecture



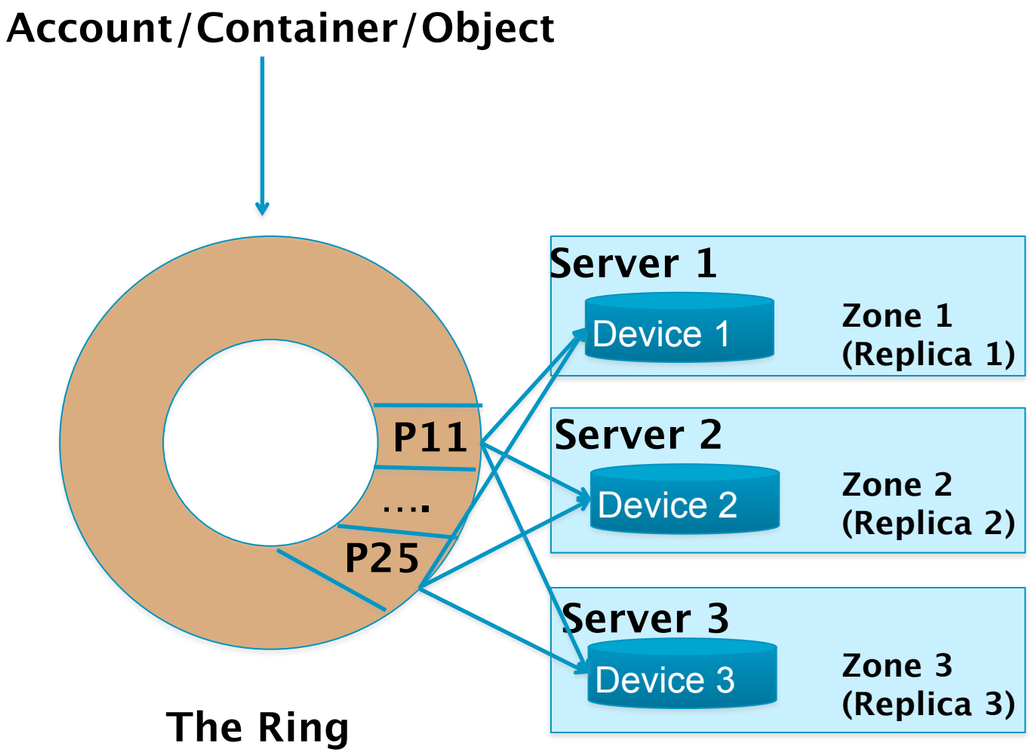
Object Store Service (Swift) Introduction

- Object-based storage system
  - NOT a block-based system like iSCSI
  - Consists of flexibly-sized data containers (objects)
- Distributed architecture
  - Avoids a single point of failure
  - Massively scalable (billions of objects and PB's of data)
- Server hard drives turn into a pool of storage

Swift Architecture



Swift Functionality



- The Ring
  - Mapping between entities and physical location
- Proxy Server
  - Ties together end users and the Swift back-end services, provides public API endpoint.
- Object Server
  - Simple blob storage server to store, retrieve, and delete objects on local disk
- Container Server
  - Lists objects
- Account Server
  - Similar to container, but lists containers

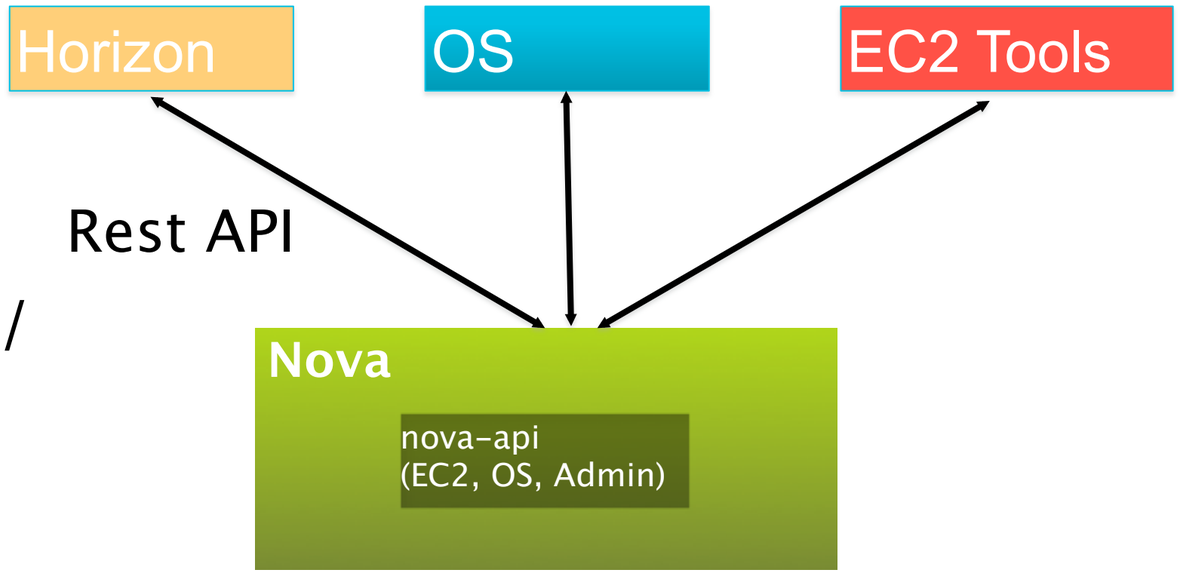
How to Deploy Openstack

- Devstack: http://devstack.org
- Automated:
  - Puppet
  - Chef
  - RDO, Juju and Charms

How to Use Openstack

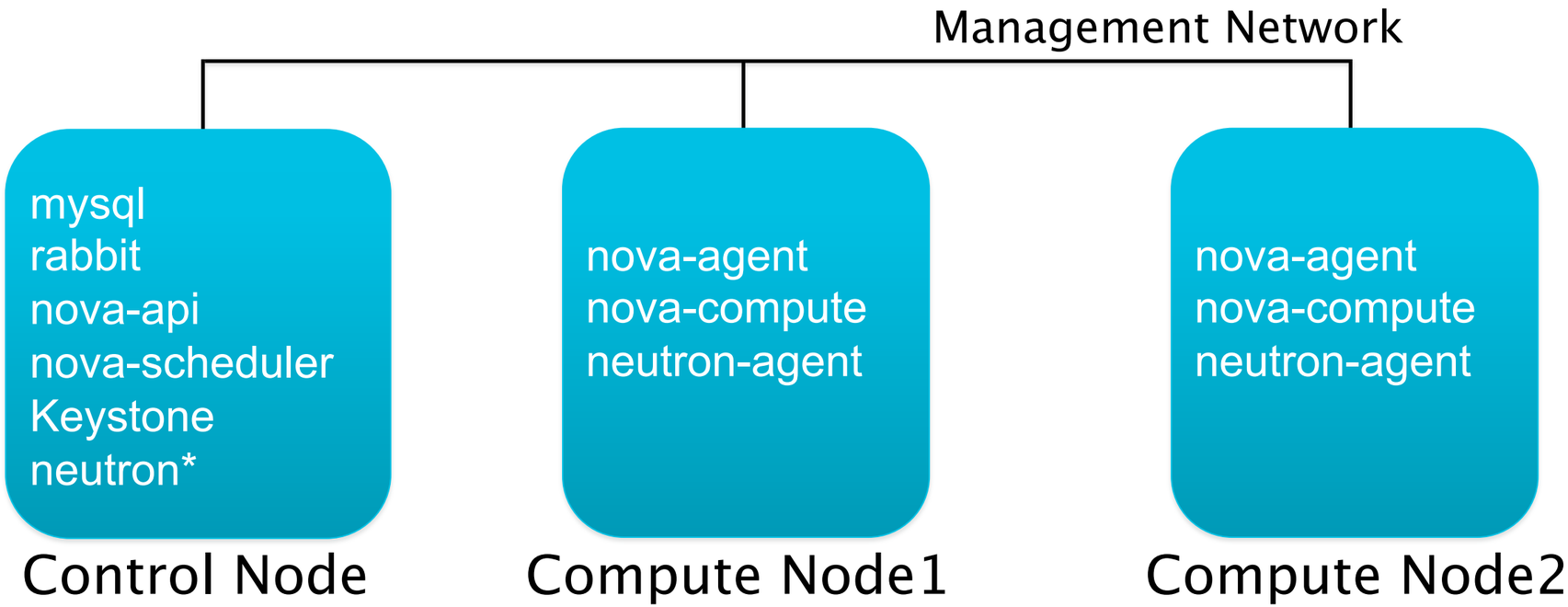
- GUI - Horizon
- CLI
- EC2 tools
- REST

http://docs.openstack.org/api/

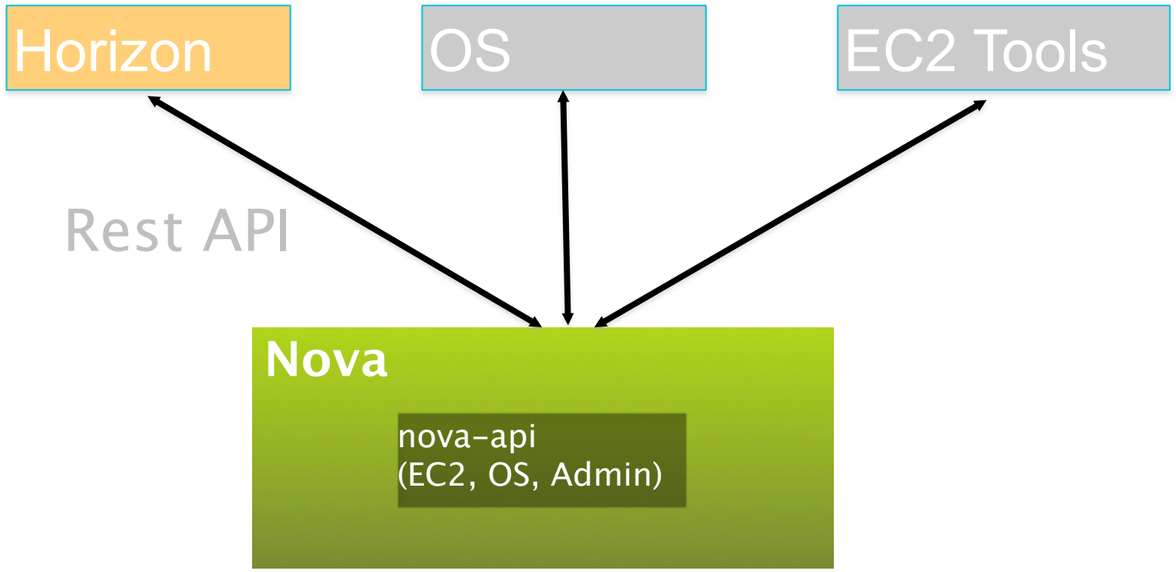


Demo Setup

- 3 nodes
  - Control node
  - 2 compute nodes



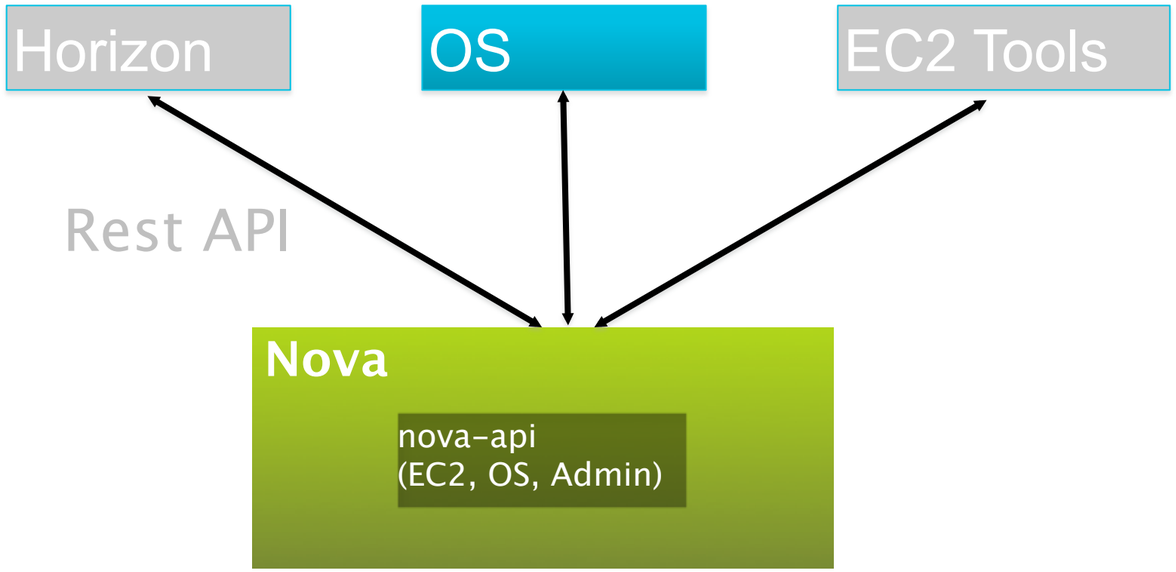
How to Use Openstack



Demo Video

[Nova Part 1]

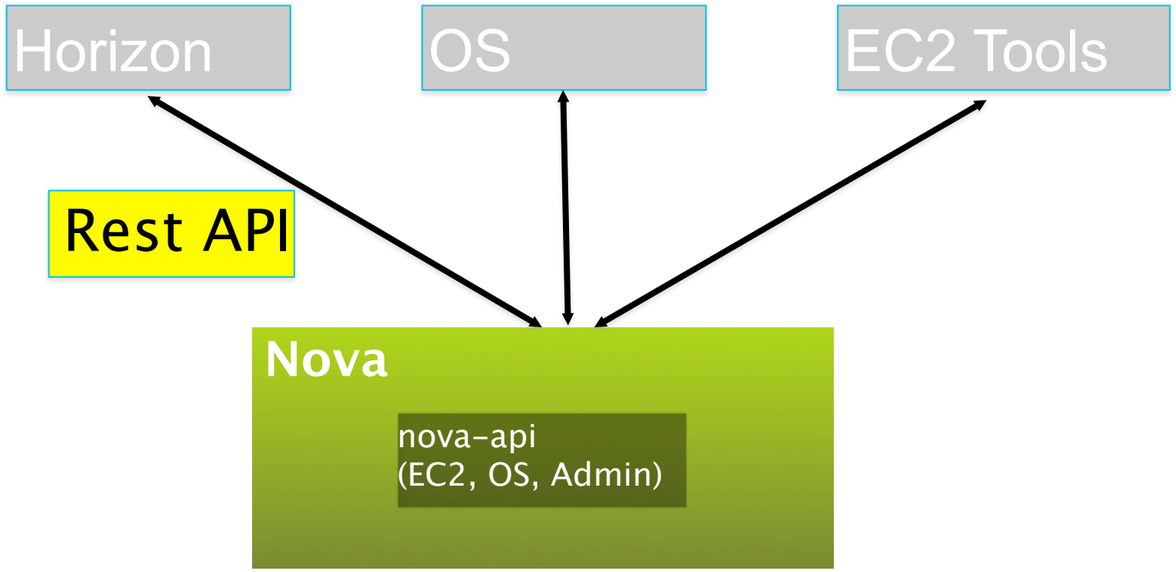
How to Use Openstack



Demo Video

[Nova Part 2]

How to Use Openstack

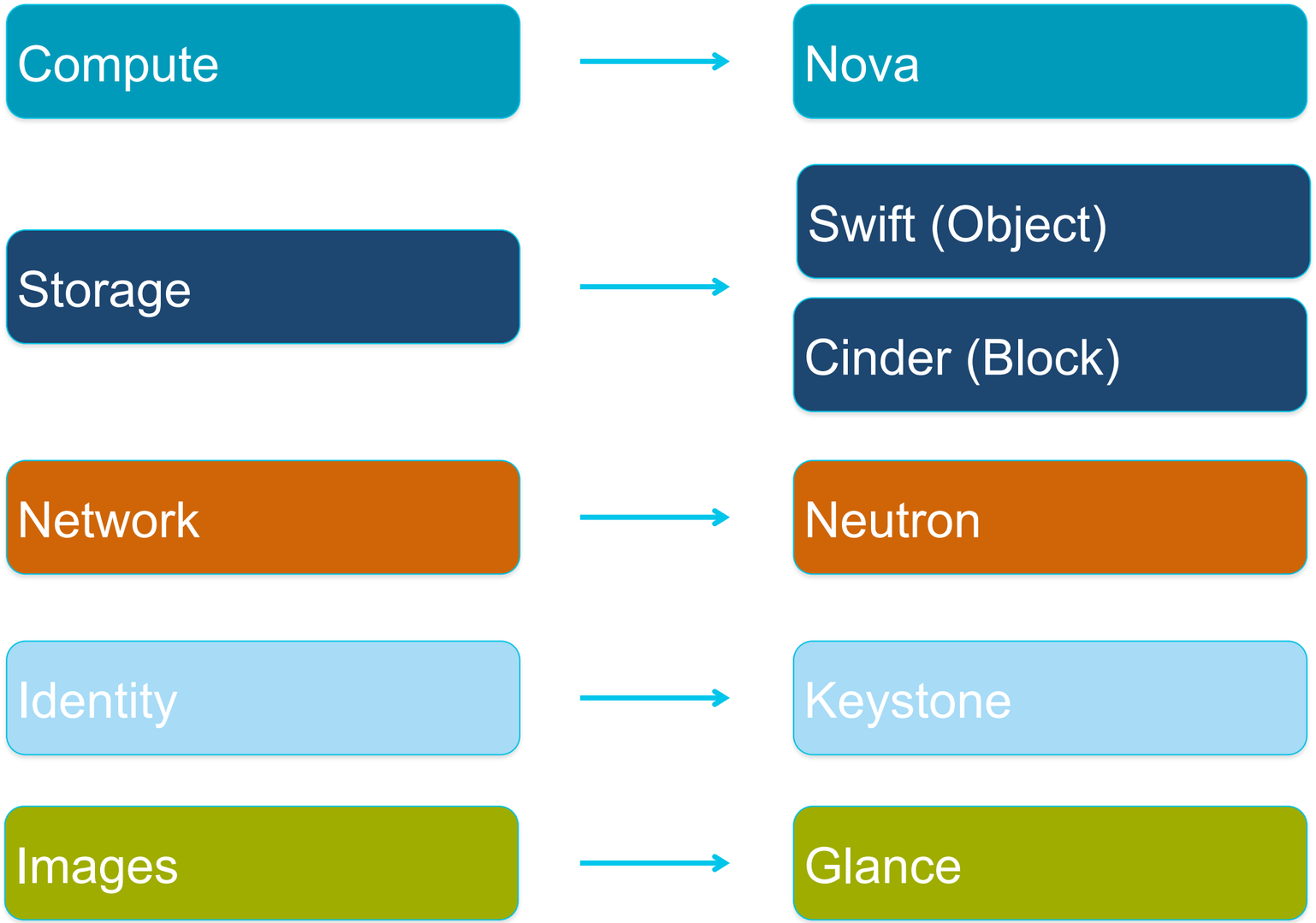


Demo Video

[Nova Part 3]

Neutron

DC Architecture mapped to OpenStack services



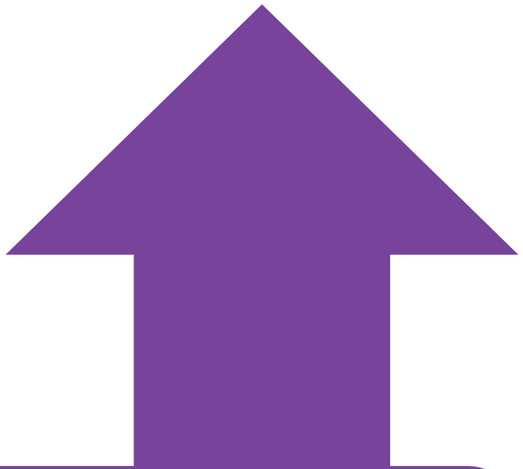
Neutron Architecture

Neutron API

API Extensions

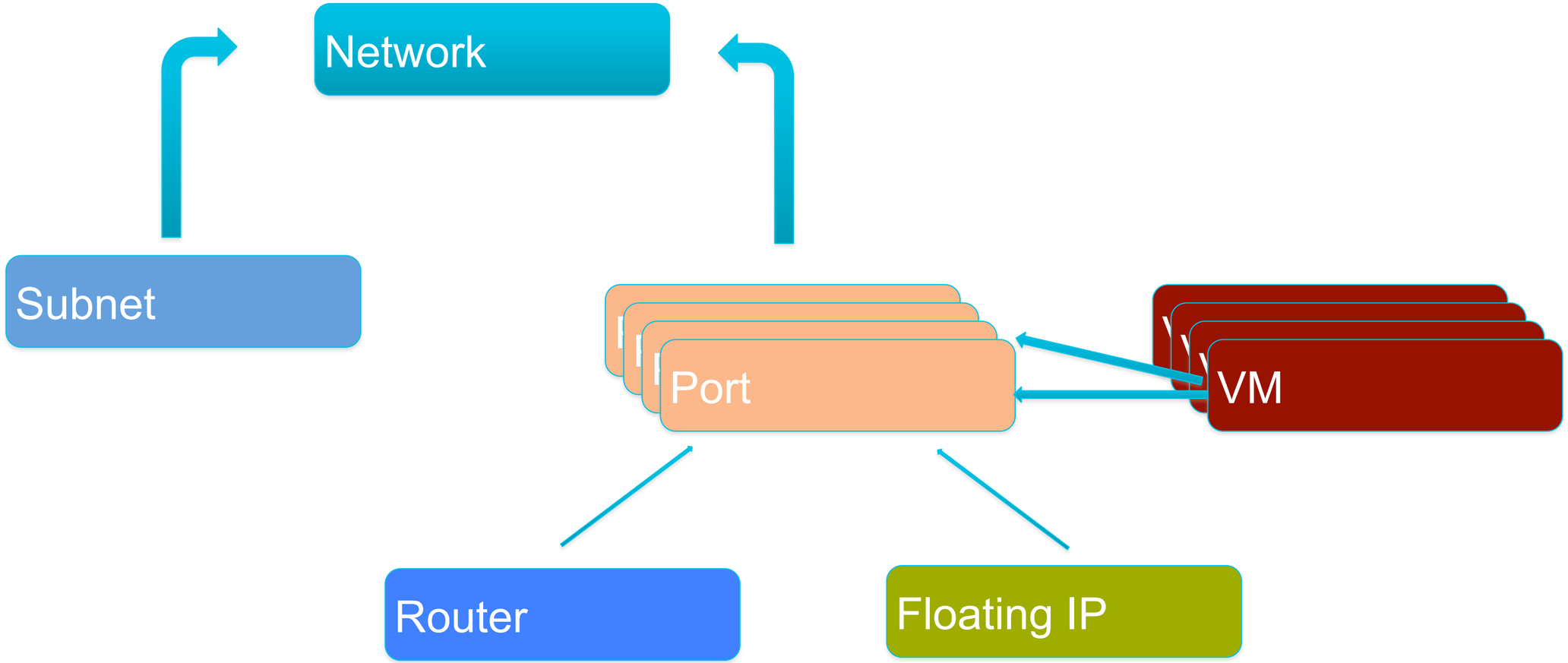
Neutron Service
- L2/L3 network abstraction and management
- Device and service attachment framework
- Does NOT implement any abstractions

Neutron Plug-in API



Vendor/User Plug-In
- Maps abstraction to implementation on physical network
- Makes all decisions about *how* a network is implemented
- Can provide additional features through API extensions

Neutron Network Abstraction

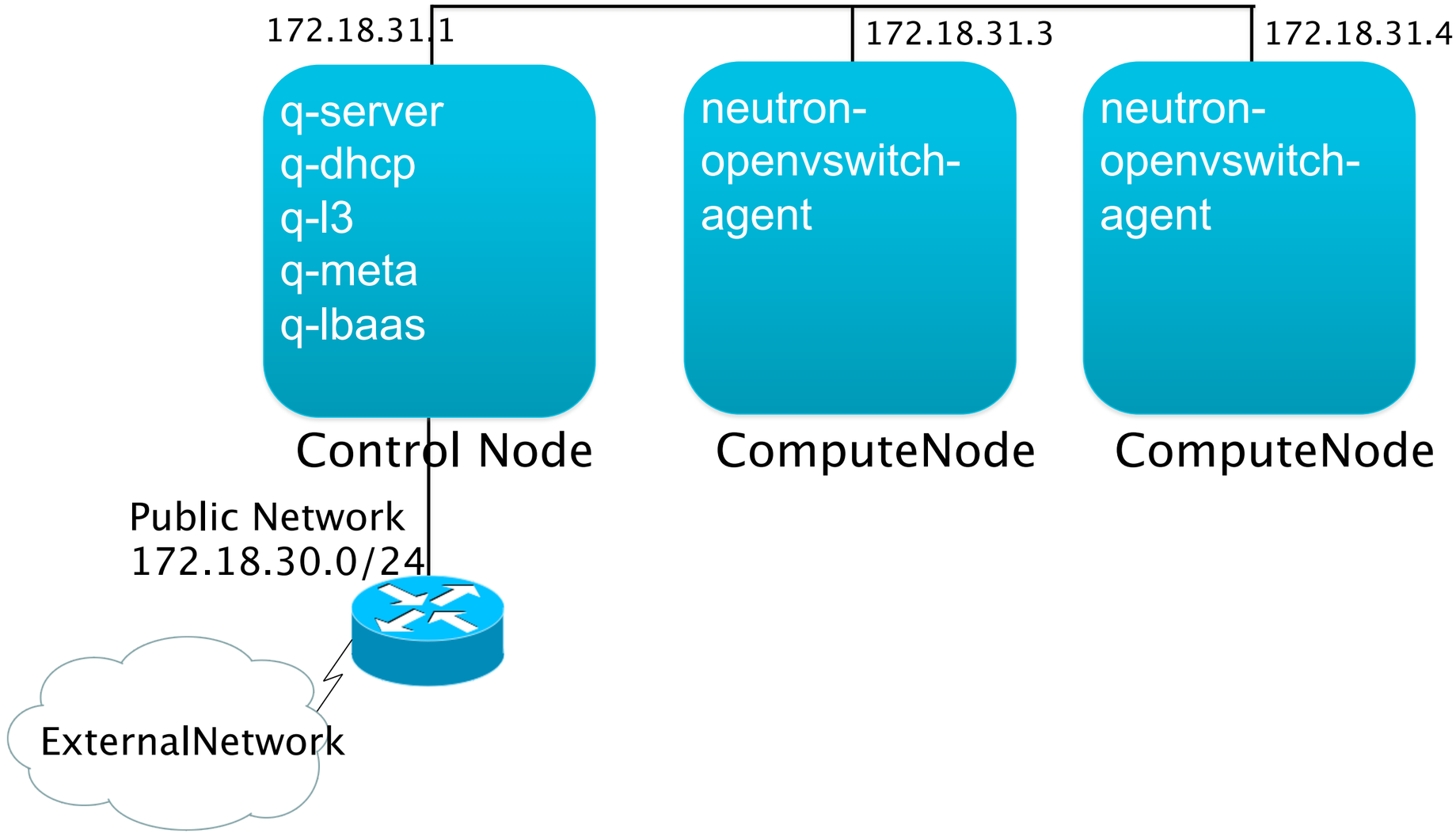


Neutron Realization and Plugins

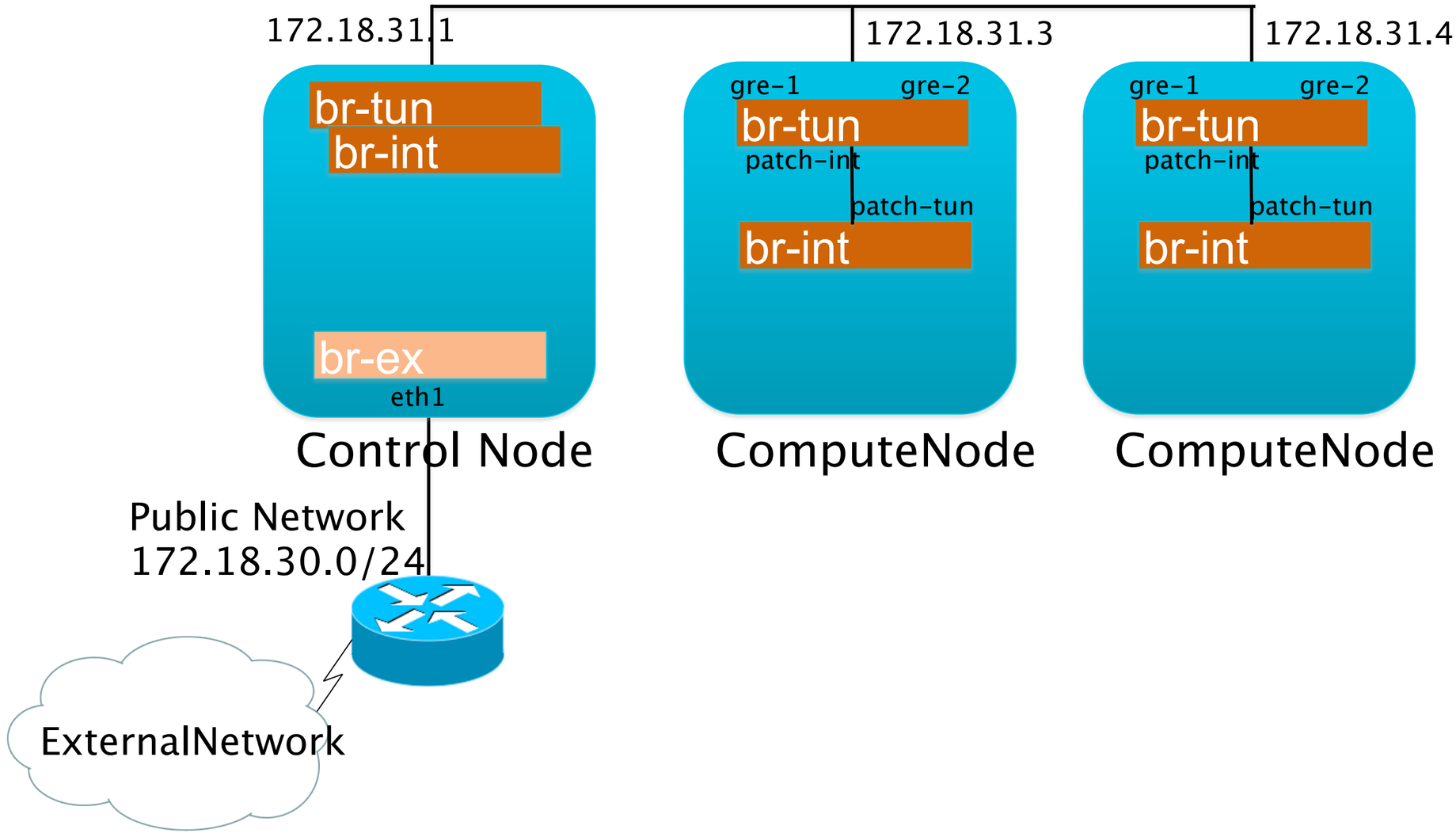
- Network Technologies
  - VLANs
  - Tunnels (GRE, VXLAN, …)
- Plugins
  - Open vSwitch Plugin
  - Cisco UCS/Nexus Plugin
  - Linux Bridge Plugin
  - OpenFlow Controller Plugin (different vendors)

Neutron Lab

Demo Setup


172.18.31.1
172.18.31.3
172.18.31.4
q-server
q-dhcp
q-l3
q-meta
q-lbaas
neutron-openvswitch-agent
neutron-openvswitch-agent
Control Node
ComputeNode
ComputeNode
Public Network
172.18.30.0/24
ExternalNetwork

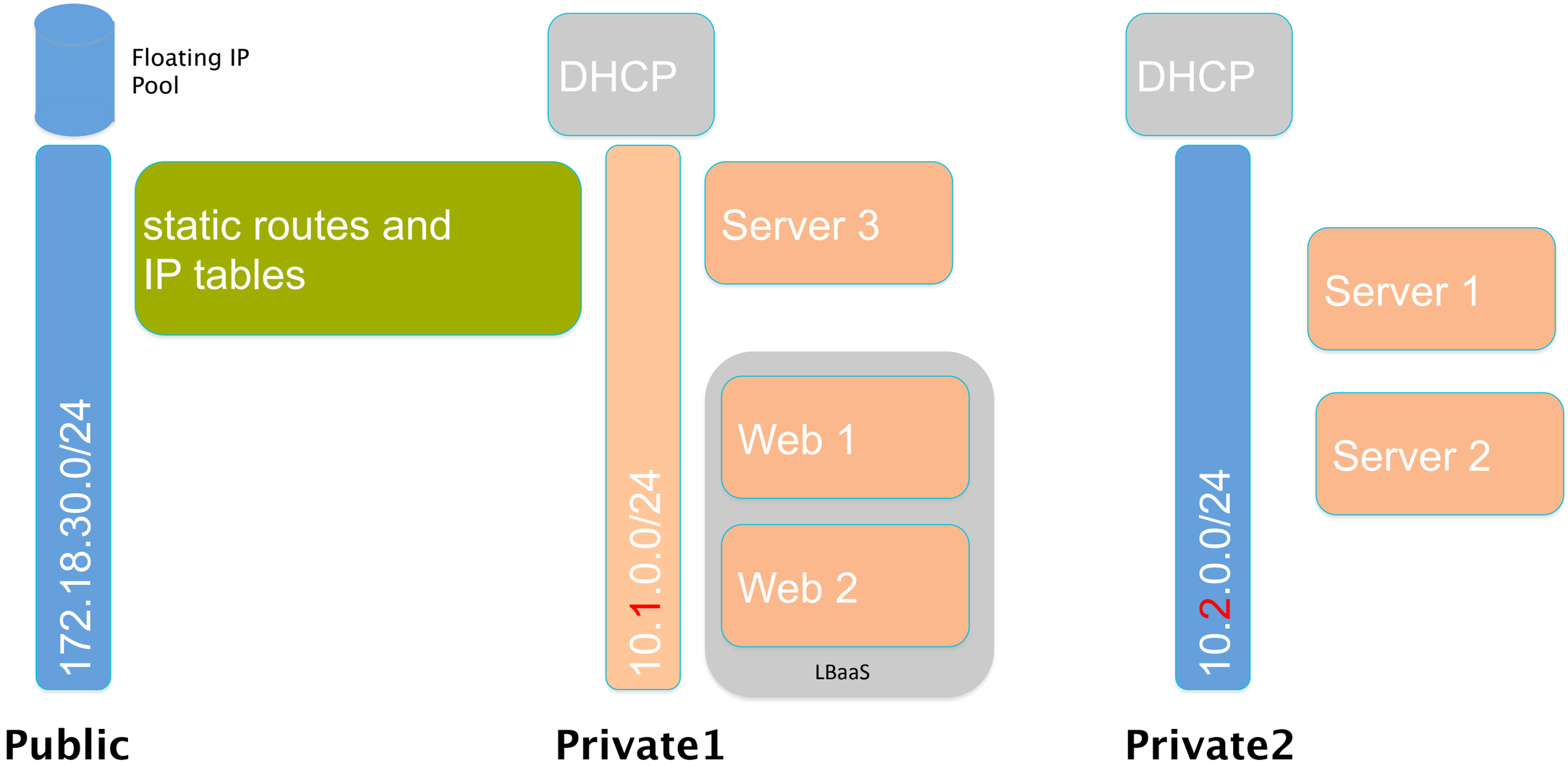
Demo Setup (OVS)



Neutron Lab

- Create 2 networks
- Create VMs on each network
- Discuss L2 connectivity, DHCP, show ping
- Create Router between the 2 networks, show ping
- Add public network to router, discuss NAT, ping external server
- Floating IPs
- Neutron Security Groups
- Neutron LBaas
- Meta Data Server

Demo Scenario



Network Namespaces

```
openstack@control:~/devstack$ source openrc admin admin
openstack@control:~/devstack$ neutron net-list
+--------------------------------------+----------+--------------------------------------------------+
| id                                   | name     | subnets                                          |
+--------------------------------------+----------+--------------------------------------------------+
| 036460df-90d3-45b2-b063-1ece8d0bd061 | public   | daaed30b-f2fc-4c28-a1c9-52ab008cf474 172.18.30.0/24 |
| 5fdb3fc3-2caf-4c06-87e3-0c7b1f4ce6a3 | private1 | 9c3e6b43-da73-4841-972a-18b57186ab81 10.1.0.0/24    |
| d74c68be-a785-4a62-9e4f-81e66693fc69 | private2 | 57c9434d-94ff-4e24-ac35-eb4305b74448 10.2.0.0/24    |
+--------------------------------------+----------+--------------------------------------------------+
openstack@control:~/devstack$ ip netns list
qdhcp-036460df-90d3-45b2-b063-1ece8d0bd061
qdhcp-5fdb3fc3-2caf-4c06-87e3-0c7b1f4ce6a3
qdhcp-d74c68be-a785-4a62-9e4f-81e66693fc69
```

Network Namespaces

```
openstack@control:~/devstack$ NET_ID=`neutron net-list | grep private2 | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack$ sudo ip netns exec qdhcp-$NET_ID ip addr list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
13: tap7d16f078-f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:1d:b8:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.2.0.2/24 brd 10.2.0.255 scope global tap7d16f078-f6
    inet6 fe80::f816:3eff:fe1d:b8d3/64 scope link
       valid_lft forever preferred_lft forever
```

DHCP Server

```
openstack@control:~/devstack$ ps aux | grep $NET_ID

nobody   26628  0.0  0.0  28900   876 ?        S    15:34   0:00 dnsmasq --no-hosts --no-resolv --strict-order --bind-
interfaces --interface=tap7d16f078-f6 --except-interface=lo --pid-file=/opt/stack/data/neutron/dhcp/d74c68be-
a785-4a62-9e4f-81e66693fc69[m[K/pid --dhcp-hostsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/host --
dhcp-optsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/opts --dhcp-script=/opt/stack/neutron/bin/
neutron-dhcp-agent-dnsmasq-lease-update --leasefile-ro --dhcp-range=set:tag0,10.2.0.0,static,120s --conf-file= --
domain=openstacklocal

root     26629  0.0  0.0  28900   244 ?        S    15:34   0:00 dnsmasq --no-hosts --no-resolv --strict-order --bind-
interfaces --interface=tap7d16f078-f6 --except-interface=lo --pid-file=/opt/stack/data/neutron/dhcp/d74c68be-
a785-4a62-9e4f-81e66693fc69/pid --dhcp-hostsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/host --dhcp-
optsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/opts --dhcp-script=/opt/stack/neutron/bin/neutron-
dhcp-agent-dnsmasq-lease-update --leasefile-ro --dhcp-range=set:tag0,10.2.0.0,static,120s --conf-file= --domain=openstacklocal

root     26653  0.0  1.1  93376 22636 ?        S    15:34   0:00 /usr/bin/python /usr/local/bin/neutron-ns-metadata-proxy --
pid_file=/opt/stack/data/neutron/external/pids/[01;31m[Kd74c68be-a785-4a62-9e4f-81e66693fc69[m[K.pid --
network_id=[01;31m[Kd74c68be-a785-4a62-9e4f-81e66693fc69[m[K --state_path=/opt/stack/data/neutron --metadata_port=80 --debug --
verbose
```

OVS

```
openstack@control:~/devstack$ sudo ovs-vsctl show | more
9cd9bbc0-c130-437c-bee7-0aaa1dfa98c3
    Bridge br-ex
        Port "eth1"
            Interface "eth1"
        Port br-ex
            Interface br-ex
                type: internal
   more…
```

OVS

```
Bridge br-int
        Port br-int
            Interface br-int
                type: internal
        Port "tapf4824d73-da"
            tag: 2
            Interface "tapf4824d73-da"
                type: internal
        Port patch-tun
            Interface patch-tun
                type: patch
                options: {peer=patch-int}
  more…
```

OVS

```
        Port "tap7d16f078-f6"
            tag: 1
            Interface "tap7d16f078-f6"
                type: internal
        Port "tap0150932a-03"
            tag: 3
            Interface "tap0150932a-03"
                type: internal
more…
```

OVS

```
Bridge br-tun
    Port "gre-2"
        Interface "gre-2"
            type: gre
            options: {in_key=flow, out_key=flow, remote_ip="172.18.31.3"}
    Port patch-int
        Interface patch-int
            type: patch
            options: {peer=patch-tun}
    Port "gre-3"
        Interface "gre-3"
            type: gre
            options: {in_key=flow, out_key=flow, remote_ip="172.18.31.4"}
    Port br-tun
        Interface br-tun
            type: internal
ovs_version: "1.9.0"
```

Switch Context to Tenant User

```
openstack@control:~/devstack$ cd context_im_user

openstack@control:~/devstack/context_im_user$ source imrc

openstack@control:~/devstack/context_im_user$ env | grep OS_

OS_PASSWORD=Gh3nt

OS_AUTH_STRATEGY=keystone

OS_AUTH_URL=http://172.18.31.1:5000/v2.0/

OS_USERNAME=im_user

OS_TENANT_NAME=iee_im

OS_CACERT=/opt/stack/data/CA/int-ca/ca-chain.pem

OS_NO_CACHE=1
```

Upload Images to Glance

```
openstack@control:~/devstack/context_im_user$ glance index
ID                                   Name                           Disk Format          Container Format     Size
------------------------------------ ------------------------------ -------------------- ------------------ --------------
b3ff5600-c04a-41fc-83af-3c0ec82540d3 cirros-0.3.1-x86_64-uec-ramdis ari                  ari                        3714968
5bbb1bf3-e080-4e9e-bf16-436cf4cddc6f cirros-0.3.1-x86_64-uec        ami                  ami                       25165824
fda13171-d675-41a8-9f49-913b4583f4ec cirros-0.3.1-x86_64-uec-kernel aki                  aki                        4955792

openstack@control:~/devstack/context_im_user$ glance add name=f17-x86_64-openstack-sda is_public=true disk_format=qcow2
container_format=bare < ../images/f17-x86_64-openstack-sda.qcow2

Added new image with ID: 65c4804c-fe8b-42f9-b127-835924b9117d

openstack@control:~/devstack/context_im_user$ glance add name=precise-amd64 is_public=true disk_format=qcow2
container_format=ovf < ../images/precise-server-cloudimg-amd64-disk1.img

Added new image with ID: e239bfe9-5101-4a8b-a4cc-ed486e1d6398

openstack@control:~/devstack/context_im_user$ IMAGE_ID=`glance image-list | grep f17-x86_64-openstack-sda | awk -F' ' '{ print
$2 }'`
```

Neutron Security Groups

```
openstack@control:~/devstack/context_im_user$ neutron security-group-list

+--------------------------------------+---------+-------------+
| id                                   | name    | description |
+--------------------------------------+---------+-------------+
| f7d59e63-bafa-4a3d-a3b8-c0870d293b63 | default | default     |
+--------------------------------------+---------+-------------+
```

Neutron Security Groups

```
openstack@control:~/devstack/context_im_user$ neutron security-group-create web_server --description 'IM demo security group'

Created a new security_group:

+-----------------------------------------------------------------------------------------+
| Field                | Value
|
+-----------------------------------------------------------------------------------------+
| description          | IM demo security group
|
| id                   | 2bd1c2ed-8de2-47fc-947b-56220556777f
|
| name                 | web_server
|
| security_group_rules | {"remote_group_id": null, "direction": "egress", "remote_ip_prefix": null, "protocol": null,
"tenant_id": "e815e6d44c6747308d30adabff1dac99", "port_range_max": null, "security_group_id":
"2bd1c2ed-8de2-47fc-947b-56220556777f", "port_range_min": null, "ethertype": "IPv4", "id":
"bc392565-2f9d-4fb8-80a5-98895260f415"} |
|                      | {"remote_group_id": null, "direction": "egress", "remote_ip_prefix": null, "protocol": null,
"tenant_id": "e815e6d44c6747308d30adabff1dac99", "port_range_max": null, "security_group_id":
"2bd1c2ed-8de2-47fc-947b-56220556777f", "port_range_min": null, "ethertype": "IPv6", "id": "a9779f42-1225-45e1-
a9a8-27ecd94e334e"} |
| tenant_id            | e815e6d44c6747308d30adabff1dac99
|
+-----------------------------------------------------------------------------------------+
```

Neutron Security Groups

```
openstack@control:~/devstack/context_im_user$ neutron security-group-rule-create --direction ingress --protocol tcp  --
port_range_min 22 --port_range_max 22 web_server

Created a new security_group_rule:

+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| direction         | ingress                              |
| ethertype         | IPv4                                 |
| id                | 44123185-099f-457e-a9e0-1fb37c104889 |
| port_range_max    | 22                                   |
| port_range_min    | 22                                   |
| protocol          | tcp                                  |
| remote_group_id   |                                      |
| remote_ip_prefix  |                                      |
| security_group_id | 2bd1c2ed-8de2-47fc-947b-56220556777f |
| tenant_id         | e815e6d44c6747308d30adabff1dac99     |
+-------------------+--------------------------------------+
```

Neutron Security Groups

```
openstack@control:~/devstack/context_im_user$ neutron security-group-rule-create --direction ingress --protocol tcp  --
port_range_min 80 --port_range_max 80 web_server

Created a new security_group_rule:

+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| direction         | ingress                              |
| ethertype         | IPv4                                 |
| id                | 4eaa761b-4c51-4e71-82fb-da13969d24e7 |
| port_range_max    | 80                                   |
| port_range_min    | 80                                   |
| protocol          | tcp                                  |
| remote_group_id   |                                      |
| remote_ip_prefix  |                                      |
| security_group_id | 2bd1c2ed-8de2-47fc-947b-56220556777f |
| tenant_id         | e815e6d44c6747308d30adabff1dac99     |
+-------------------+--------------------------------------+
```

Neutron Security Groups

```
openstack@control:~/devstack/context_im_user$ neutron security-group-rule-create --direction ingress --protocol icmp web_server

Created a new security_group_rule:

+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| direction         | ingress                              |
| ethertype         | IPv4                                 |
| id                | 8790f25d-84d7-4275-8435-f9ebf271cce2 |
| port_range_max    |                                      |
| port_range_min    |                                      |
| protocol          | icmp                                 |
| remote_group_id   |                                      |
| remote_ip_prefix  |                                      |
| security_group_id | 2bd1c2ed-8de2-47fc-947b-56220556777f |
| tenant_id         | e815e6d44c6747308d30adabff1dac99     |
+-------------------+--------------------------------------+

openstack@control:~/devstack/context_im_user$ SECGROUP_ID=`neutron security-group-list | grep web_server | awk -F' ' '{ print
$2 }'`
```

Neutron Port

```
openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep private2 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name server1_port $NET_ID

Created a new port:

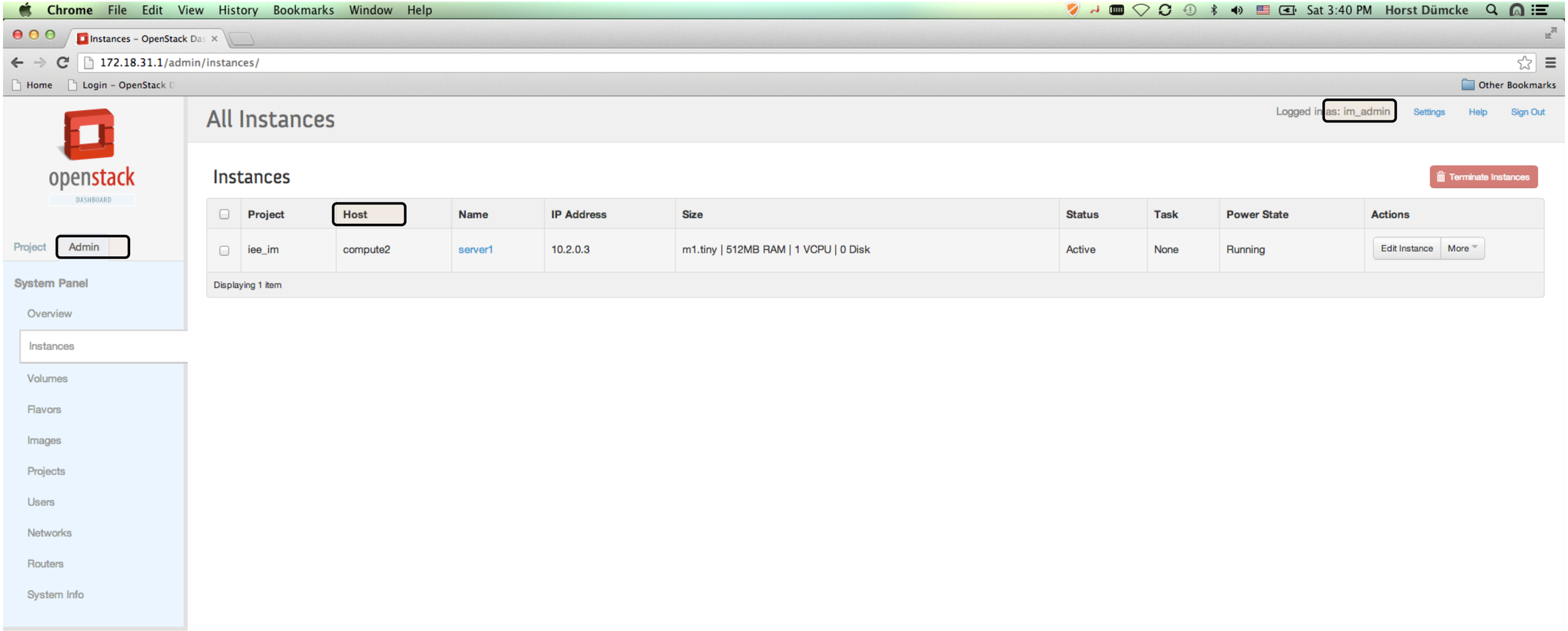
+-----------------+-------------------------------------------------------------------------------+
| Field           | Value                                                                         |
+-----------------+-------------------------------------------------------------------------------+
| admin_state_up  | True                                                                          |
| device_id       |                                                                               |
| device_owner    |                                                                               |
| fixed_ips       | {"subnet_id": "57c9434d-94ff-4e24-ac35-eb4305b74448", "ip_address": "10.2.0.3"} |
| id              | 40070dfd-5900-431d-bb85-581a9ab60223                                          |
| mac_address     | fa:16:3e:a2:b4:2e                                                             |
| name            | server1_port                                                                  |
| network_id      | d74c68be-a785-4a62-9e4f-81e66693fc69                                          |
| security_groups | 2bd1c2ed-8de2-47fc-947b-56220556777f                                          |
| status          | DOWN                                                                          |
| tenant_id       | e815e6d44c6747308d30adabff1dac99                                              |
+-----------------+-------------------------------------------------------------------------------+
```

Nova Boot Server

```
openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID server1

+-----------------------------+--------------------------------------+
| Property                    | Value                                |
+-----------------------------+--------------------------------------+
| status                      | BUILD                                |
| updated                     | 2013-05-25T13:38:29Z                 |
| OS-EXT-STS:task_state       | scheduling                           |
| key_name                    | None                                 |
| image                       | f17-x86_64-openstack-sda             |
| hostId                      |                                      |
| OS-EXT-STS:vm_state         | building                             |
| flavor                      | m1.tiny                              |
| id                          | dbbeec72-92e2-4285-960e-32d01bdc0b50 |
| security_groups             | [{u'name': u'default'}]              |
| user_id                     | 8da345659e694f7f8d8138af8fbd9c42     |
| name                        | server1                              |
| adminPass                   | CUXEiPk4LyhY                         |
| tenant_id                   | e815e6d44c6747308d30adabff1dac99     |
| created                     | 2013-05-25T13:38:29Z                 |
| OS-DCF:diskConfig           | MANUAL                               |
| metadata                    | {}                                   |
| accessIPv4                  |                                      |
| accessIPv6                  |                                      |
| progress                    | 0                                    |
| OS-EXT-STS:power_state      | 0                                    |
| OS-EXT-AZ:availability_zone | nova                                 |
| config_drive                |                                      |
+-----------------------------+--------------------------------------+
```

Nova Boot Server



Virsh on Compute Node

```
openstack@compute2:~/devstack$ virsh list
 Id    Name                           State
----------------------------------------------------
 1     instance-00000001              running
```

Virsh on Compute Node

```
openstack@compute2:~/devstack$ virsh dumpxml instance-00000001 | more
<domain type='kvm' id='1'>
  <name>instance-00000001</name>
  <uuid>dbbeec72-92e2-4285-960e-32d01bdc0b50</uuid>
  <memory unit='KiB'>524288</memory>
  <currentMemory unit='KiB'>524288</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <sysinfo type='smbios'>
    <system>
      <entry name='manufacturer'>OpenStack Foundation</entry>
      <entry name='product'>OpenStack Nova</entry>
      <entry name='version'>2013.1.1</entry>
      <entry name='serial'>de5fd950-eb69-11e1-1000-000000000009</entry>
      <entry name='uuid'>dbbeec72-92e2-4285-960e-32d01bdc0b50</entry>
    </system>
```

Virsh on Compute Node (Network Interface)

```
  <devices>
...skipping lines
    </interface>
      <mac address='fa:16:3e:a2:b4:2e'/>
      <source bridge='qbr40070dfd-59'/>
      <target dev='tap40070dfd-59'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
...skipping lines

</domain>
```

OVS on Compute Node

```
openstack@compute2:~/devstack$ sudo ovs-vsctl show
887df6af-dccf-4b15-a7db-48f15865dcf5
    Bridge br-int
        Port "qvo40070dfd-59"
            tag: 1
            Interface "qvo40070dfd-59”
        Port br-int
            Interface br-int
                type: internal
        Port patch-tun
            Interface patch-tun
                type: patch
                options: {peer=patch-int}
    Bridge br-tun
        Port "gre-2"
            Interface "gre-2"
                type: gre
                options: {in_key=flow, out_key=flow, remote_ip="172.18.31.3"}
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
        Port "gre-1"
            Interface "gre-1"
                type: gre
                options: {in_key=flow, out_key=flow, remote_ip="172.18.31.1"}
        Port br-tun
            Interface br-tun
                type: internal
    ovs_version: "1.9.0"
```

KVM on Compute Node

```
openstack@compute2:~/devstack$ virsh dumpxml instance-00000001 | grep tap

      <target dev='tap40070dfd-59'/>

openstack@compute2:~/devstack$ ps aux | grep tap

1066994 18.6  0.0 3868964 287304 ?      Sl   15:38   0:15 qemu-system-x86_64 -machine accel=kvm:tcg -name instance-00000001 -S
   -M pc-i440fx-1.4 -m 512 -smp 1,sockets=1,cores=1,threads=1 -uuid dbbeec72-92e2-4285-960e-32d01bdc0b50 -smbios
   type=1,manufacturer=OpenStack Foundation,product=OpenStack Nova,version=2013.1.1,serial=de5fd950-
   eb69-11e1-1000-000000000009,uuid=dbbeec72-92e2-4285-960e-32d01bdc0b50 -no-user-config -nodefaults -chardev
   socket,id=charmonitor,path=/var/lib/libvirt/qemu/instance-00000001.monitor,server,nowait -mon
   chardev=charmonitor,id=monitor,mode=control -rtc base=utc,driftfix=slew -no-shutdown -device piix3-usb-uhci,id=usb,bus=pci.
   0,addr=0x1.0x2 -drive file=/opt/stack/data/nova/instances/dbbeec72-92e2-4285-960e-32d01bdc0b50/disk,if=none,id=drive-virtio-
   disk0,format=qcow2,cache=none -device virtio-blk-pci,scsi=off,bus=pci.0,addr=0x4,drive=drive-virtio-disk0,id=virtio-
   disk0,bootindex=1 –netdev tap,fd=24,id=hostnet0,vhost=on,vhostfd=25 -device virtio-net-pci,netdev=hostnet0,id=net0,mac=fa:
   16:3e:a2:b4:2e,bus=pci.0,addr=0x3 -chardev file,id=charserial0,path=/opt/stack/data/nova/instances/
   dbbeec72-92e2-4285-960e-32d01bdc0b50/console.log -device isa-serial,chardev=charserial0,id=serial0 -chardev
   pty,id=charserial1 -device isa-serial,chardev=charserial1,id=serial1 -device usb-tablet,id=input0 -vnc 0.0.0.0:0 -k en-us -
   vga cirrus -device virtio-balloon-pci,id=balloon0,bus=pci.0,addr=0x5
```

IPtables on Compute Node

```
openstack@compute2:~/devstack$ sudo iptables -L

...skipping lines

Chain neutron-openvswi-i40070dfd-5 (1 references)

target     prot opt source               destination

DROP       all  --  anywhere             anywhere             state INVALID

RETURN     all  --  anywhere             anywhere             state RELATED,ESTABLISHED

RETURN     tcp  --  anywhere             anywhere             tcp dpt:ssh

RETURN     tcp  --  anywhere             anywhere             tcp dpt:http

RETURN     icmp --  anywhere             anywhere

RETURN     udp  --  10.2.0.2             anywhere             udp spt:bootps dpt:bootpc

...skipping lines
```

DHCP Server on Control Node

```
openstack@control:~/devstack/context_im_user$ ps aux | grep $NET_ID

nobody   26628  0.0  0.0  28900   964 ?        S    15:34   0:00 dnsmasq --no-hosts --no-resolv --strict-order --bind-
interfaces --interface=tap7d16f078-f6 --except-interface=lo --pid-file=/opt/stack/data/neutron/dhcp/d74c68be-
a785-4a62-9e4f-81e66693fc69/pid --dhcp-hostsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/host --dhcp-
optsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/opts --dhcp-script=/opt/stack/neutron/bin/neutron-
dhcp-agent-dnsmasq-lease-update --leasefile-ro --dhcp-range=set:tag0,10.2.0.0,static,120s --conf-file= --domain=openstacklocal

root     26629  0.0  0.0  28900   452 ?        S    15:34   0:00 dnsmasq --no-hosts --no-resolv --strict-order --bind-
interfaces --interface=tap7d16f078-f6 --except-interface=lo --pid-file=/opt/stack/data/neutron/dhcp/d74c68be-
a785-4a62-9e4f-81e66693fc69/pid --dhcp-hostsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/host --dhcp-
optsfile=/opt/stack/data/neutron/dhcp/d74c68be-a785-4a62-9e4f-81e66693fc69/opts --dhcp-script=/opt/stack/neutron/bin/neutron-
dhcp-agent-dnsmasq-lease-update --leasefile-ro --dhcp-range=set:tag0,10.2.0.0,static,120s --conf-file= --domain=openstacklocal

root     26653  0.0  1.1  93376 22636 ?        S    15:34   0:00 /usr/bin/python /usr/local/bin/neutron-ns-metadata-proxy --
pid_file=/opt/stack/data/neutron/external/pids/d74c68be-a785-4a62-9e4f-81e66693fc69.pid --network_id=d74c68be-
a785-4a62-9e4f-81e66693fc69 --state_path=/opt/stack/data/neutron --metadata_port=80 --debug --verbose

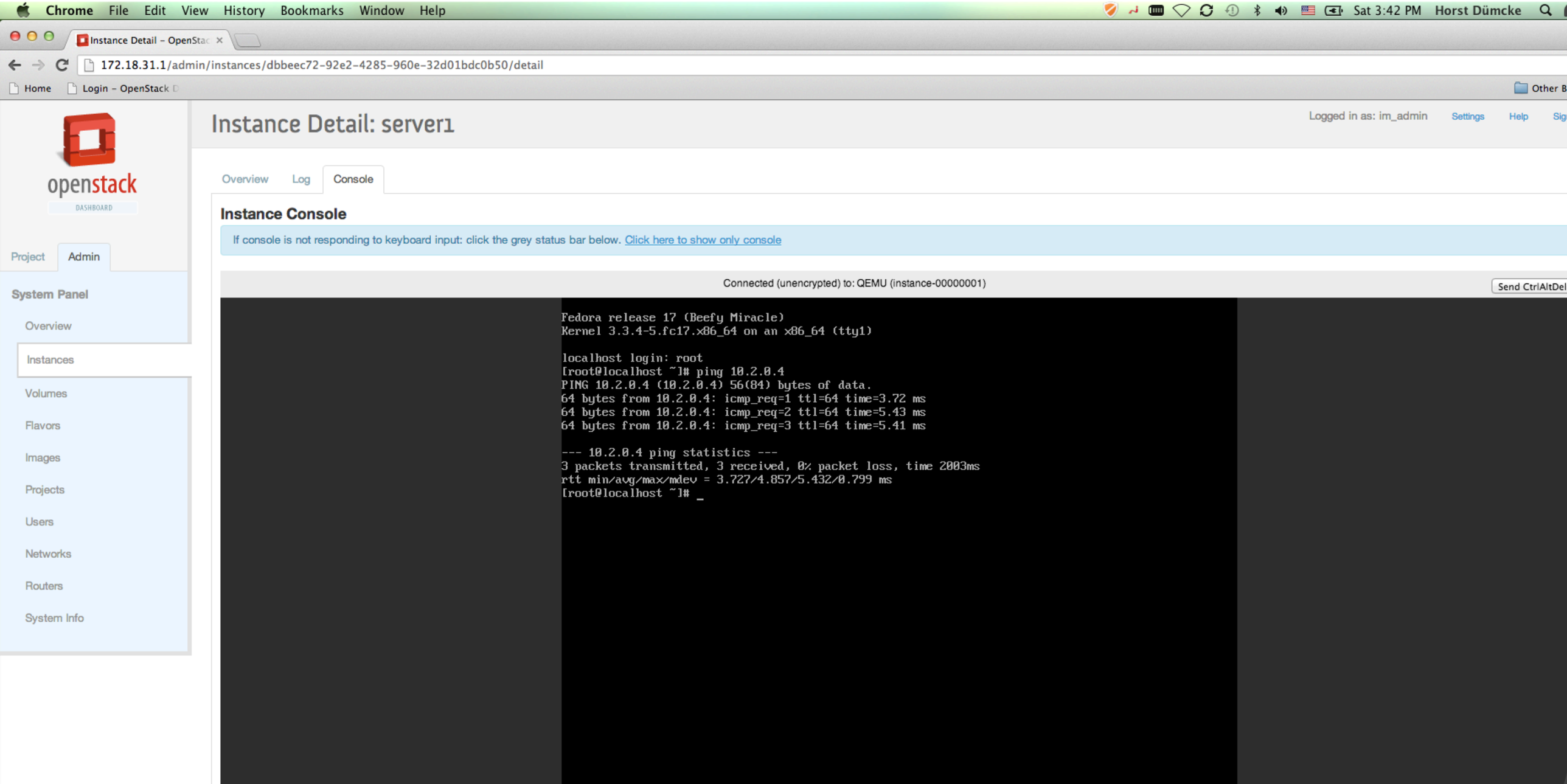
openstack@control:~/devstack/context_im_user$ cat /opt/stack/data/neutron/dhcp/$NET_ID/host

fa:16:3e:a2:b4:2e,10-2-0-3.openstacklocal,10.2.0.3
```

Boot Server 2

```
openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name server2_port $NET_ID
...skipping output
openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep server2_port | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID server2
...skipping output
```

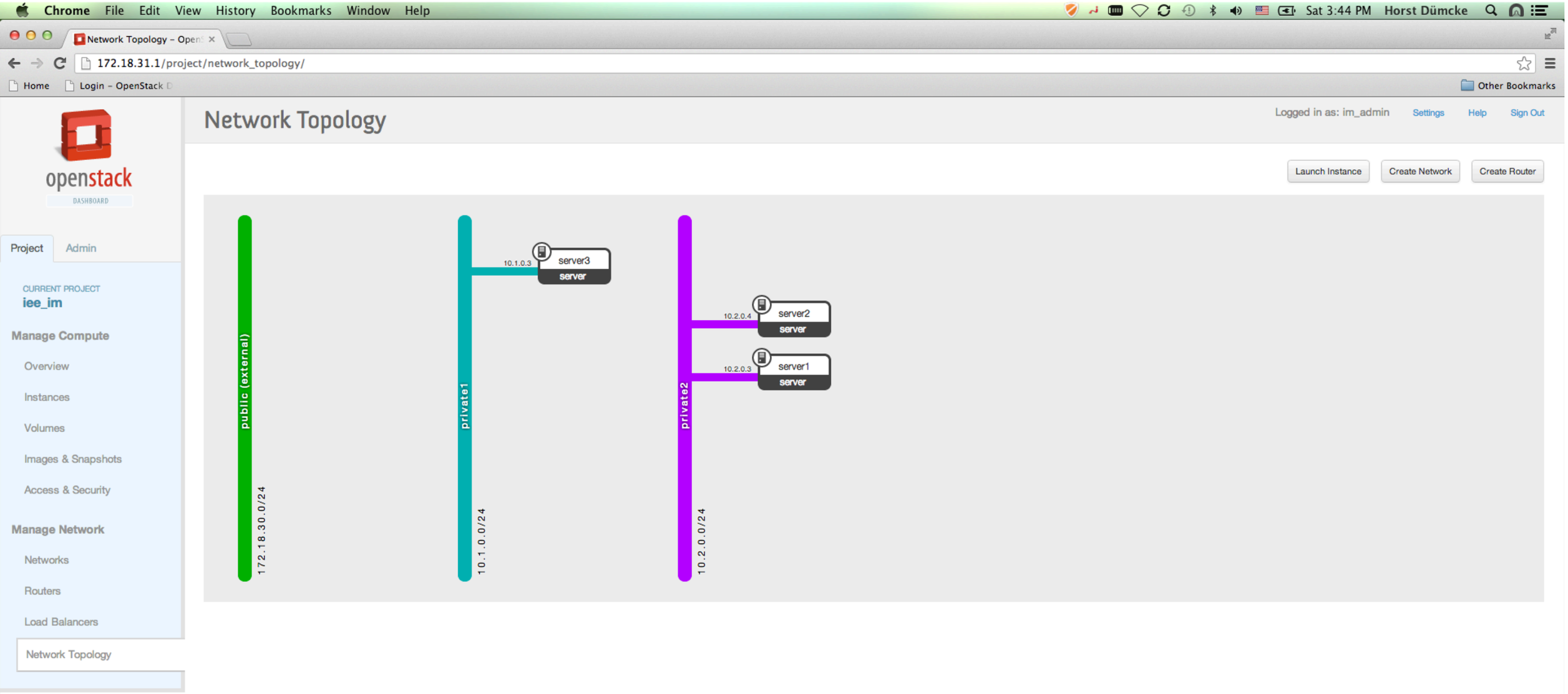
Ping Server 2



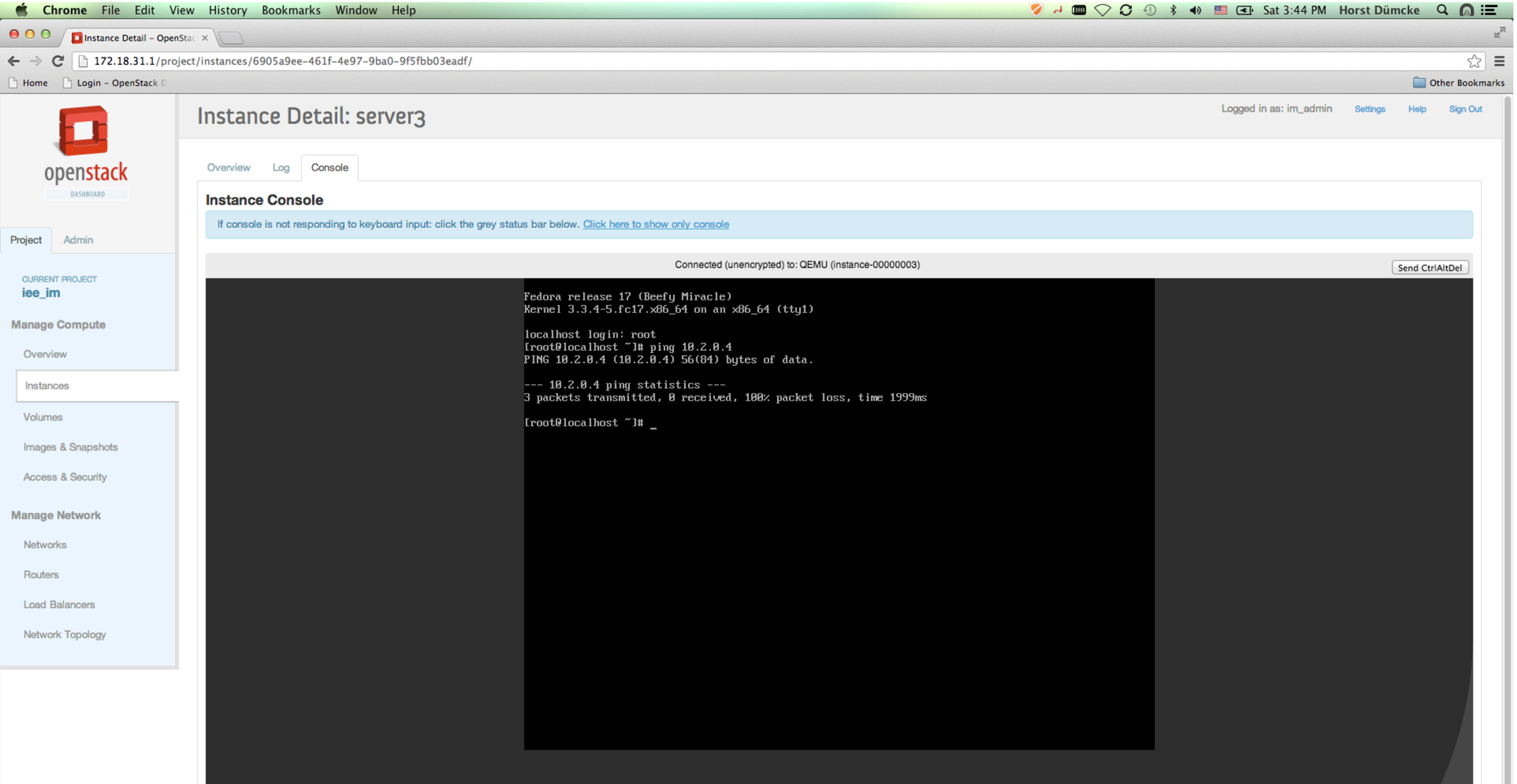
Boot Server 3

```
openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep private1 | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name server3_port $NET_ID
...skipping output
openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep server3_port | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID server3
...skipping output
```

Network Topology



Ping Server 2



Create Router

```
openstack@control:~/devstack/context_im_user$ neutron router-create router1


Created a new router:

+-----------------------+--------------------------------------+
| Field                 | Value                                |
+-----------------------+--------------------------------------+
| admin_state_up        | True                                 |
| external_gateway_info |                                      |
| id                    | 4ae28bf7-2f80-4f3b-be83-c420e0eb19e9 |
| name                  | router1                              |
| status                | ACTIVE                               |
| tenant_id             | e815e6d44c6747308d30adabff1dac99     |
+-----------------------+--------------------------------------+

openstack@control:~/devstack/context_im_user$ SUBNET_ID=`neutron subnet-list | grep 10.2.0.0 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron router-interface-add router1 ${SUBNET_ID}

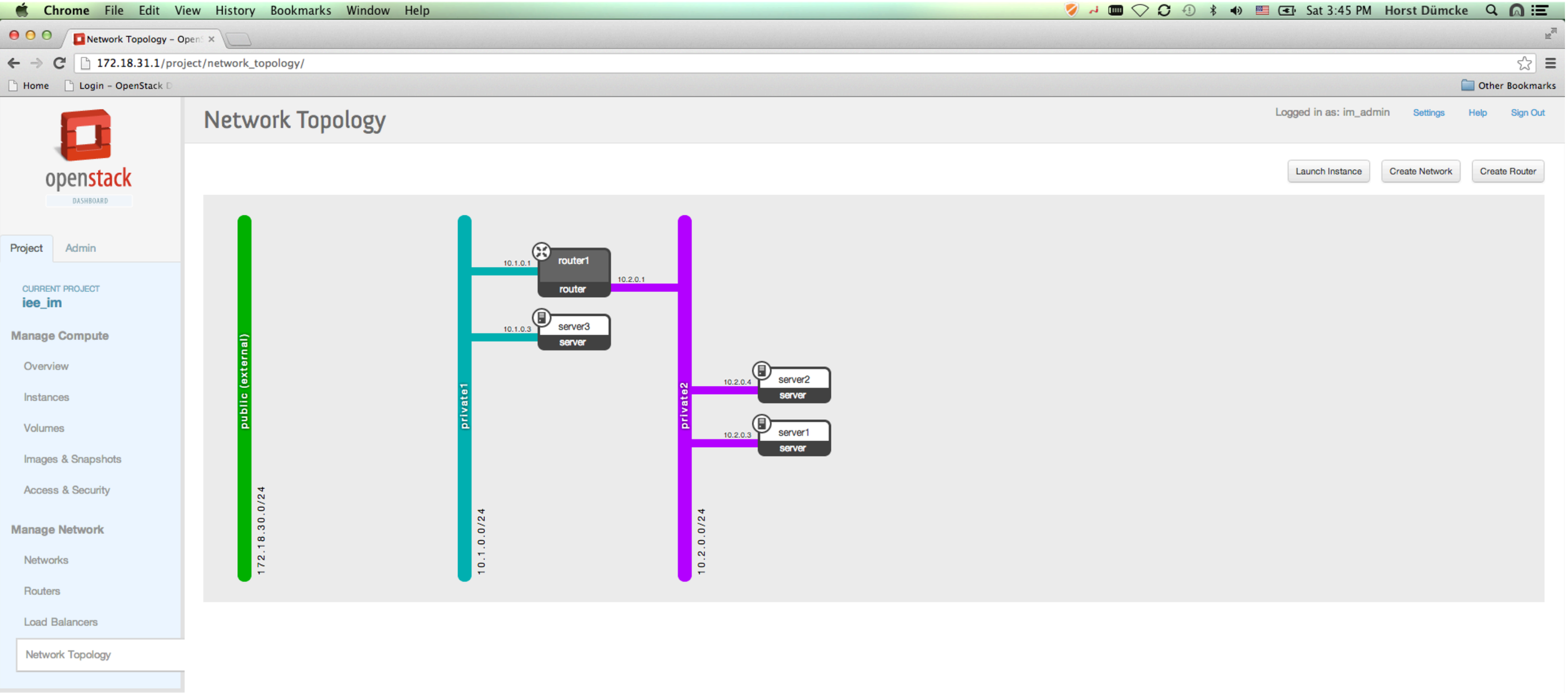
Added interface to router router1

openstack@control:~/devstack/context_im_user$ SUBNET_ID=`neutron subnet-list | grep 10.1.0.0 | awk -F' ' '{ print $2 }'`

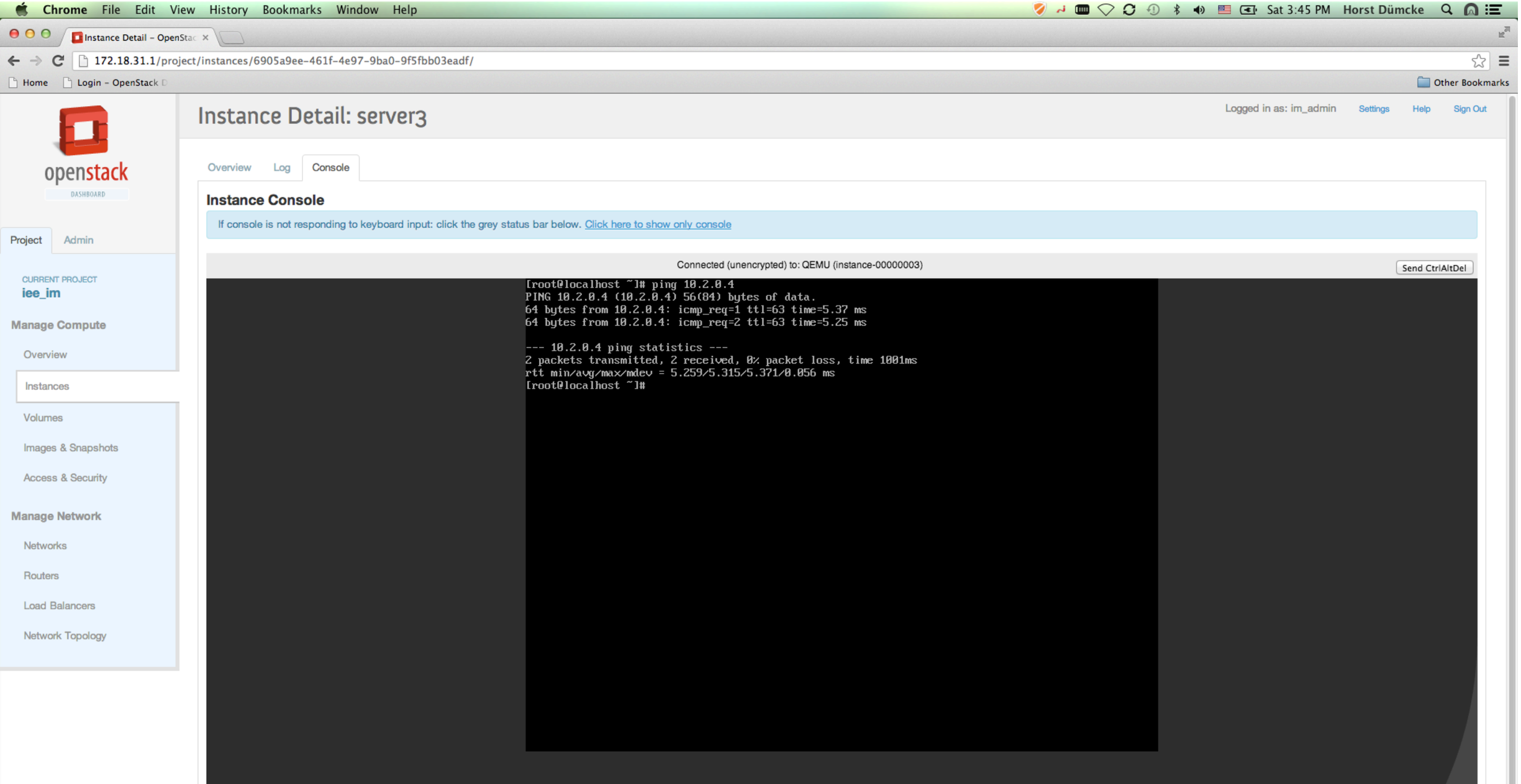
openstack@control:~/devstack/context_im_user$ neutron router-interface-add router1 ${SUBNET_ID}

Added interface to router router1
```

Network Topology



Ping Server 2



Add Gateway

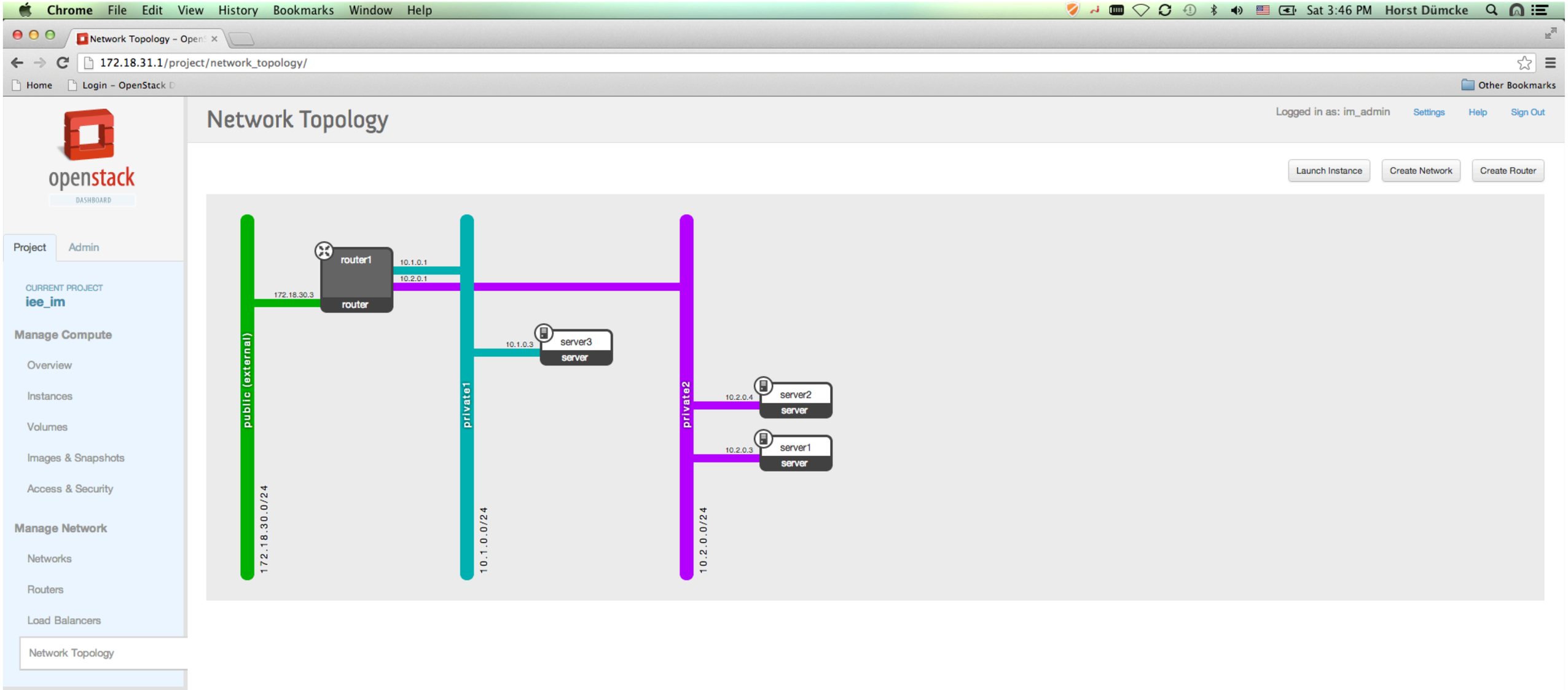
```
openstack@control:~/devstack/context_im_user$ neutron router-gateway-set router1 public

Set gateway for router router1

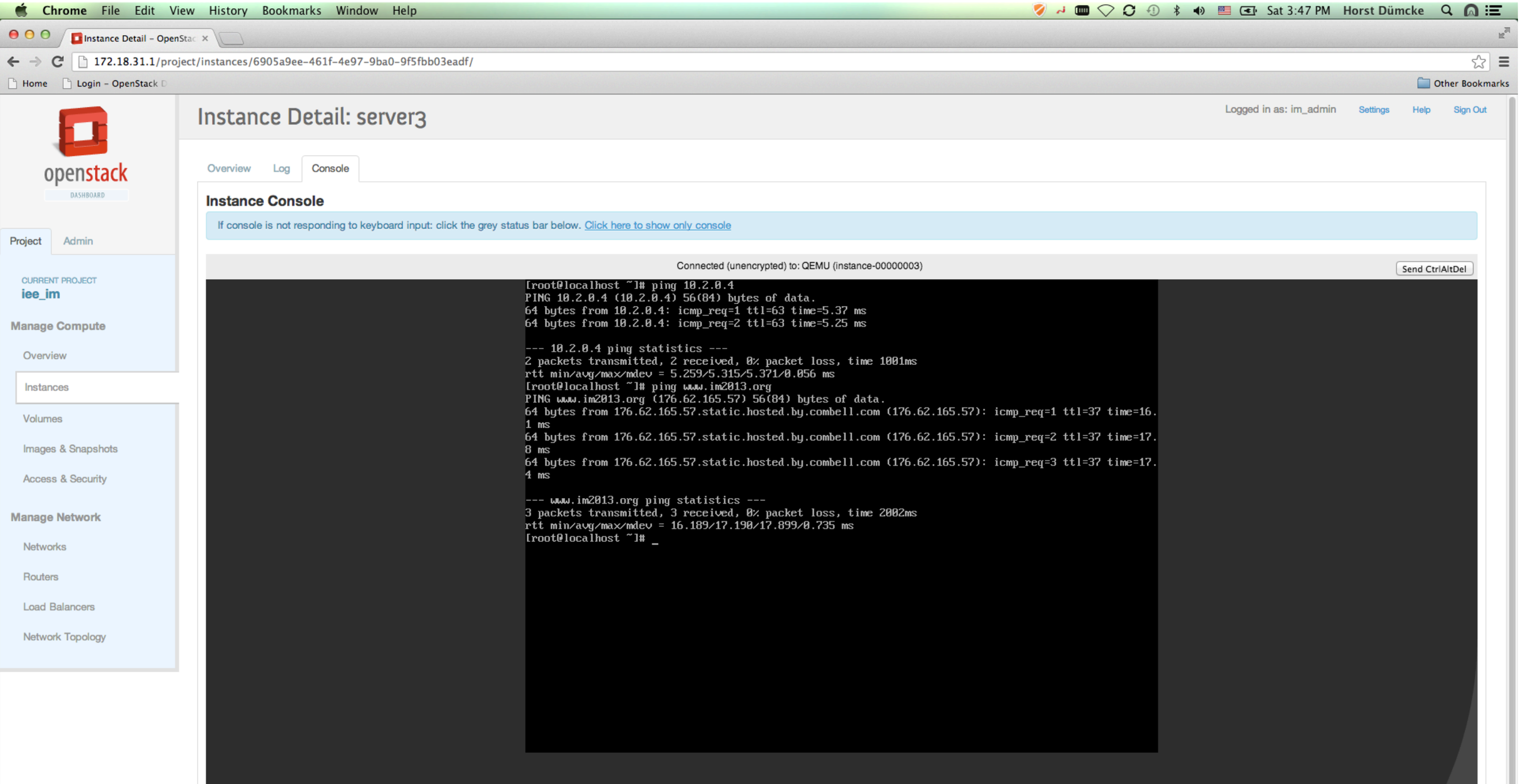
openstack@control:~/devstack/context_im_user$ neutron router-list

+--------------------------------------+---------+-----------------------------------------------------------+
| id                                   | name    | external_gateway_info                                     |
+--------------------------------------+---------+-----------------------------------------------------------+
| 4ae28bf7-2f80-4f3b-be83-c420e0eb19e9 | router1 | {"network_id": "036460df-90d3-45b2-b063-1ece8d0bd061"} |
+--------------------------------------+---------+-----------------------------------------------------------+
```

Network Topology



Ping External Network



Router Network Namespace

```
openstack@control:~/devstack/context_im_user$ ROUTER_ID=`neutron router-list | grep router1 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ sudo ip netns exec qrouter-$ROUTER_ID ip addr list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
16: qr-1b2d50ea-93: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:76:06:93 brd ff:ff:ff:ff:ff:ff
    inet 10.2.0.1/24 brd 10.2.0.255 scope global qr-1b2d50ea-93
    inet6 fe80::f816:3eff:fe76:693/64 scope link
       valid_lft forever preferred_lft forever
17: qr-026e163f-f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:7b:6a:64 brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.1/24 brd 10.1.0.255 scope global qr-026e163f-f6
    inet6 fe80::f816:3eff:fe7b:6a64/64 scope link
       valid_lft forever preferred_lft forever
18: qg-e4106235-97: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:9a:28:58 brd ff:ff:ff:ff:ff:ff
    inet 172.18.30.3/24 brd 172.18.30.255 scope global qg-e4106235-97
    inet 172.18.30.4/32 brd 172.18.30.4 scope global qg-e4106235-97
    inet 172.18.30.5/32 brd 172.18.30.5 scope global qg-e4106235-97
    inet 172.18.30.6/32 brd 172.18.30.6 scope global qg-e4106235-97
    inet6 fe80::f816:3eff:fe9a:2858/64 scope link
       valid_lft forever preferred_lft forever

openstack@control:~/devstack/context_im_user$ sudo ip netns exec qrouter-$ROUTER_ID ip route list
default via 172.18.30.1 dev qg-e4106235-97
10.1.0.0/24 dev qr-026e163f-f6  proto kernel  scope link  src 10.1.0.1
10.2.0.0/24 dev qr-1b2d50ea-93  proto kernel  scope link  src 10.2.0.1
172.18.30.0/24 dev qg-e4106235-97  proto kernel  scope link  src 172.18.30.3
```

OVS on Control Node

```
openstack@control:~/devstack/context_im_user$ sudo ovs-vsctl show | more
9cd9bbc0-c130-437c-bee7-0aaa1dfa98c3
    Bridge br-ex
        Port "eth1"
            Interface "eth1"
        Port br-ex
            Interface br-ex
                type: internal
        Port "qg-e4106235-97"
            Interface "qg-e4106235-97"
                type: internal
More…
```

OVS on Control Node

```
Bridge br-int
...skipping lines
        Port "qr-026e163f-f6"
            tag: 2
            Interface "qr-026e163f-f6"
                type: internal
...skipping lines
        Port "qr-1b2d50ea-93"
            tag: 1
            Interface "qr-1b2d50ea-93"
                type: internal
```

Add Floating IP Address

```
openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep public | awk -F' ' '{ print \$2 }'`

openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep 10.1.0.3 | awk -F' ' '{ print \$2 }'`

openstack@control:~/devstack/context_im_user$ neutron floatingip-create --port_id=$PORT_ID $NET_ID

Created a new floatingip:

+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    | 10.1.0.3                             |
| floating_ip_address | 172.18.30.4                          |
| floating_network_id | 036460df-90d3-45b2-b063-1ece8d0bd061 |
| id                  | fa56af80-2718-40ef-9d8d-b465f5676a8a |
| port_id             | 5f79b923-d5f0-4f2b-8af9-f2538fb5d480 |
| router_id           | 4ae28bf7-2f80-4f3b-be83-c420e0eb19e9 |
| tenant_id           | e815e6d44c6747308d30adabff1dac99     |
+---------------------+--------------------------------------+
```

Ping Server from External Network

```
[root@Cobbler ~]# ping 172.18.30.4
PING 172.18.30.4 (172.18.30.4) 56(84) bytes of data.
64 bytes from 172.18.30.4: icmp_req=2 ttl=61 time=2.11 ms
64 bytes from 172.18.30.4: icmp_req=3 ttl=61 time=4.36 ms
64 bytes from 172.18.30.4: icmp_req=4 ttl=61 time=5.50 ms
```

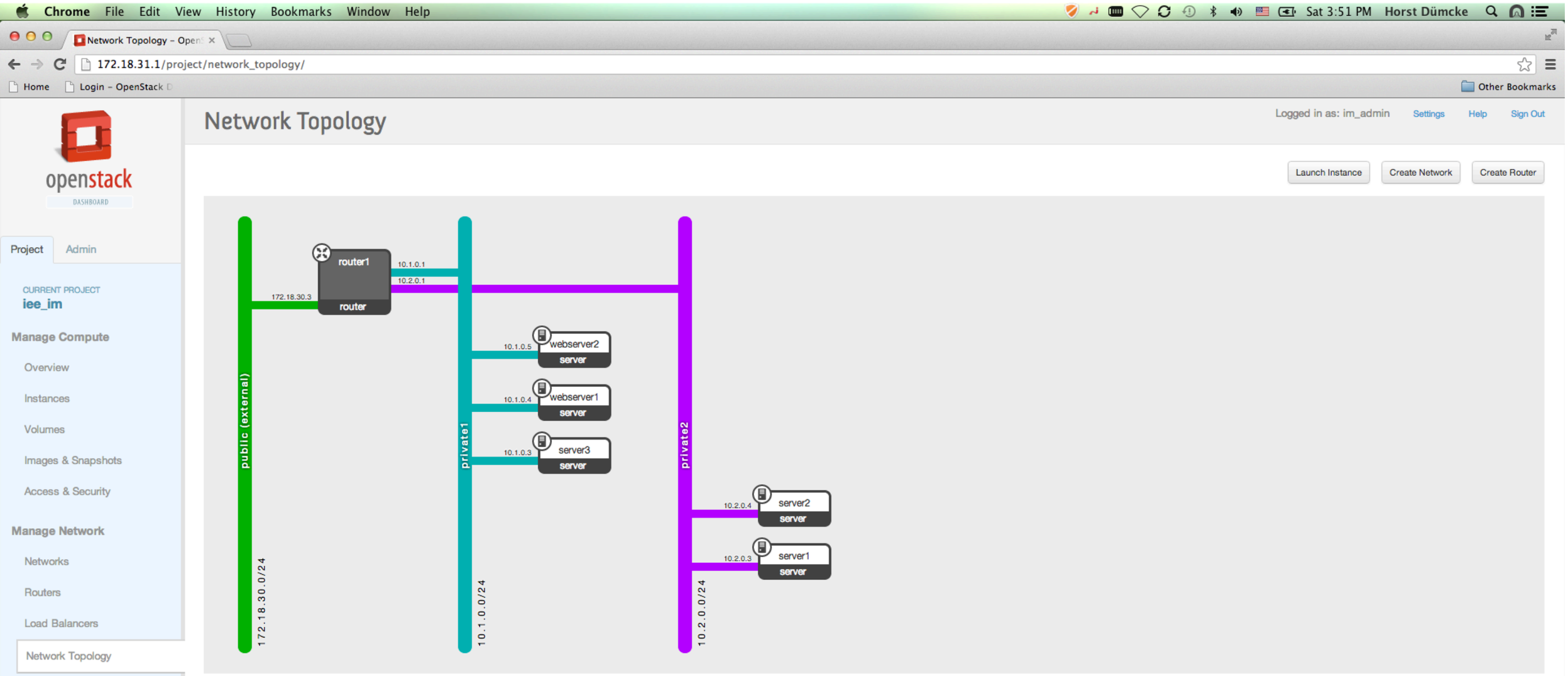
Iptables on Control Node

```
openstack@control:~/devstack/context_im_user$ NETNS=`ip netns list | grep qrouter`
openstack@control:~/devstack/context_im_user$ sudo ip netns exec $NETNS iptables -L -t nat  | grep SNAT
SNAT       all  --  10.1.0.3             anywhere             to:172.18.30.4
SNAT       all  --  10.2.0.0/24          anywhere             to:172.18.30.3
SNAT       all  --  10.1.0.0/24          anywhere             to:172.18.30.3
openstack@control:~/devstack/context_im_user$ sudo ip netns exec $NETNS iptables -L -t nat  | grep DNAT
DNAT       all  --  anywhere             172.18.30.4          to:10.1.0.3
ACCEPT     all  --  anywhere             anywhere             ! ctstate DNAT
DNAT       all  --  anywhere             172.18.30.4          to:10.1.0.3
```

Boot Webservers

```
openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep private1 | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name webserver1_port $NET_ID
...skipping output
openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep webserver1_port | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID webserver1
...skipping output
openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name webserver2_port $NET_ID
...skipping output
openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep webserver1_port | awk -F' ' '{ print $2 }'`
openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID webserver2
...skipping output
```

Network Topology



Create LB Pool

```
openstack@control:~/devstack/context_im_user$ SUBNET_ID=`neutron subnet-list | grep 10.1.0.0 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron lb-pool-create --lb-method ROUND_ROBIN --name im_lbpool --protocol HTTP
--subnet-id $SUBNET_ID

Created a new pool:

+------------------+--------------------------------------+
| Field            | Value                                |
+------------------+--------------------------------------+
| admin_state_up   | True                                 |
| description      |                                      |
| health_monitors  |                                      |
| id               | 64ceba3f-326c-4fa8-8d3b-55439bde4864 |
| lb_method        | ROUND_ROBIN                          |
| members          |                                      |
| name             | im_lbpool                            |
| protocol         | HTTP                                 |
| status           | PENDING_CREATE                       |
| subnet_id        | 9c3e6b43-da73-4841-972a-18b57186ab81 |
| tenant_id        | e815e6d44c6747308d30adabff1dac99     |
| vip_id           |                                      |
+------------------+--------------------------------------+
```

Add Server to LB Pool

```
openstack@control:~/devstack/context_im_user$ SERVER_ID=`nova list | grep webserver1 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ NW_ADDR=`nova show $SERVER_ID | grep network | awk -F' ' '{ print $5 }'`

openstack@control:~/devstack/context_im_user$ neutron lb-member-create --address $NW_ADDR --protocol-port 80 im_lbpool

Created a new member:

+----------------+--------------------------------------+
| Field          | Value                                |
+----------------+--------------------------------------+
| address        | 10.1.0.4                             |
| admin_state_up | True                                 |
| id             | d61f2fa9-203e-472d-bc8f-f009d085fe98 |
| pool_id        | 64ceba3f-326c-4fa8-8d3b-55439bde4864 |
| protocol_port  | 80                                   |
| status         | PENDING_CREATE                       |
| tenant_id      | e815e6d44c6747308d30adabff1dac99     |
| weight         | 1                                    |
+----------------+--------------------------------------+

openstack@control:~/devstack/context_im_user$ SERVER_ID=`nova list | grep webserver2 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ NW_ADDR=`nova show $SERVER_ID | grep network | awk -F' ' '{ print $5 }'`

openstack@control:~/devstack/context_im_user$ neutron lb-member-create --address $NW_ADDR --protocol-port 80 im_lbpool

...skipping output
```

Create Health Monitor

```
openstack@control:~/devstack/context_im_user$ neutron lb-healthmonitor-create --delay 3 --type HTTP --max-retries 3 --timeout 3

Created a new health_monitor:

+----------------+--------------------------------------+
| Field          | Value                                |
+----------------+--------------------------------------+
| admin_state_up | True                                 |
| delay          | 3                                    |
| expected_codes | 200                                  |
| http_method    | GET                                  |
| id             | 0a2fd417-63fd-4d2f-bb84-c2c32bca0fde |
| max_retries    | 3                                    |
| status         | PENDING_CREATE                       |
| tenant_id      | e815e6d44c6747308d30adabff1dac99     |
| timeout        | 3                                    |
| type           | HTTP                                 |
| url_path       | /                                    |
+----------------+--------------------------------------+

openstack@control:~/devstack/context_im_user$ HM_ID=`neutron lb-healthmonitor-list | grep HTTP | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron lb-healthmonitor-associate $HM_ID im_lbpool

Associated health monitor 0a2fd417-63fd-4d2f-bb84-c2c32bca0fde
```

Create VIP

```
openstack@control:~/devstack/context_im_user$ SUBNET_ID=`neutron subnet-list | grep 10.1.0.0 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ POOL_ID=`neutron lb-pool-list | grep im_lbpool | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron lb-vip-create --name im_vip --protocol-port 80 --protocol HTTP --subnet-
id $SUBNET_ID im_lbpool

Created a new vip:

+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| address           | 10.1.0.6                             |
| admin_state_up    | True                                 |
| connection_limit  | -1                                   |
| description       |                                      |
| id                | 53594387-ec6a-4246-8b4d-f03c7a7bc5a3 |
| name              | im_vip                               |
| pool_id           | 64ceba3f-326c-4fa8-8d3b-55439bde4864 |
| port_id           | d5166ca5-eca7-4686-91ba-8cd45a01ad36 |
| protocol          | HTTP                                 |
| protocol_port     | 80                                   |
| status            | PENDING_CREATE                       |
| subnet_id         | 9c3e6b43-da73-4841-972a-18b57186ab81 |
| tenant_id         | e815e6d44c6747308d30adabff1dac99     |
+-------------------+--------------------------------------+
```

Show HAproxy

```
openstack@control:~/devstack/context_im_user$ ps aux | grep haproxy

nobody   16237  0.0  0.0  20412   988 ?        Ss   15:52   0:00 haproxy -f /opt/stack/data/neutron/lbaas/
64ceba3f-326c-4fa8-8d3b-55439bde4864/conf -p /opt/stack/data/neutron/lbaas/64ceba3f-326c-4fa8-8d3b-55439bde4864/pid
```

Show Haproxy Config File

```
openstack@control:~/devstack/context_im_user$ POOL_ID=`neutron lb-pool-list | grep im_lbpool | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ cat /opt/stack/data/neutron/lbaas/$POOL_ID/conf

global
        daemon
        user nobody
        group nogroup
        log /dev/log local0
        log /dev/log local1 notice
        stats socket /opt/stack/data/neutron/lbaas/64ceba3f-326c-4fa8-8d3b-55439bde4864/sock mode 0666 level user
defaults
        log global
        retries 3
        option redispatch
        timeout connect 5000
        timeout client 50000
        timeout server 50000
frontend 53594387-ec6a-4246-8b4d-f03c7a7bc5a3
        option tcplog
        bind 10.1.0.6:80
        mode http
        default_backend 64ceba3f-326c-4fa8-8d3b-55439bde4864
        option forwardfor
backend 64ceba3f-326c-4fa8-8d3b-55439bde4864
        mode http
        balance roundrobin
        option forwardfor
        timeout check 3s
        option httpchk GET /
        http-check expect rstatus 200
        server cc63a8dc-42f7-4451-a6fd-4d1bea48ba07 10.1.0.5:80 weight 1 check inter 3s fall 3
        server d61f2fa9-203e-472d-bc8f-f009d085fe98 10.1.0.4:80 weight 1 check inter 3s fall 3
```

Show Haproxy Network Namespace

```
openstack@control:~/devstack/context_im_user$ ip netns list

qlbaas-64ceba3f-326c-4fa8-8d3b-55439bde4864
qrouter-4ae28bf7-2f80-4f3b-be83-c420e0eb19e9
qdhcp-036460df-90d3-45b2-b063-1ece8d0bd061
qdhcp-5fdb3fc3-2caf-4c06-87e3-0c7b1f4ce6a3
qdhcp-d74c68be-a785-4a62-9e4f-81e66693fc69

openstack@control:~/devstack/context_im_user$ sudo ip netns exec qlbaas-$POOL_ID ip addr list

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
19: tapd5166ca5-ec: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:bb:6b:0a brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.6/24 brd 10.1.0.255 scope global tapd5166ca5-ec
    inet6 fe80::f816:3eff:febb:6b0a/64 scope link
       valid_lft forever preferred_lft forever
```

Map VIP to Floating IP Address

```
openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep public | awk -F' ' '{ print \$2 }'`

openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep 10.1.0.6 | awk -F' ' '{ print \$2 }'`

openstack@control:~/devstack/context_im_user$ neutron floatingip-create --port_id=$PORT_ID $NET_ID

Created a new floatingip:

+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    | 10.1.0.6                             |
| floating_ip_address | 172.18.30.5                          |
| floating_network_id | 036460df-90d3-45b2-b063-1ece8d0bd061 |
| id                  | f7f4298e-ff47-4134-af26-ce6aae2d8a16 |
| port_id             | d5166ca5-eca7-4686-91ba-8cd45a01ad36 |
| router_id           | 4ae28bf7-2f80-4f3b-be83-c420e0eb19e9 |
| tenant_id           | e815e6d44c6747308d30adabff1dac99     |
+---------------------+--------------------------------------+
```

Prepare Webservers

```
yum –y install httpd

iptables –F

echo ‘this is server [1,2]’ > /var/www/html/index.html

service http start
```

Test VIP from External Network

```
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 1
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 2
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 1
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 2
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 1
[root@Cobbler ~]# wget -O - http://172.18.30.5 2 2> /dev/null
this is server 2
```

Meta Data Insertion (Generate Key Pair)

```
openstack@control:~/devstack/context_im_user$ ssh-keygen -f /tmp/id_rsa -t rsa -N ''
Generating public/private rsa key pair.
Your identification has been saved in /tmp/id_rsa.
Your public key has been saved in /tmp/id_rsa.pub.
The key fingerprint is:
47:62:e9:40:1d:f5:77:d2:bb:84:ee:fe:52:0e:83:a0 openstack@control
The key's randomart image is:
+--[ RSA 2048]----+
|       ...o.     |
|      .  .. .   . |
|       . + . . o o|
|        + +   ..o.|
|         S o .. o |
|        E . ..o...|
|             .=. |
|            .. . |
|            .oo. |
+-----------------+
openstack@control:~/devstack/context_im_user$ nova keypair-add --pub_key /tmp/id_rsa.pub noms_key
```

Meta Data Insertion (Boot Server)

```
openstack@control:~/devstack/context_im_user$ IMAGE_ID=`glance image-list | grep precise-amd64 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep private1 | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ neutron port-create --security-group web_server --name server4_port $NET_ID

...skipping output

openstack@control:~/devstack/context_im_user$ PORT_ID=`neutron port-list | grep server4_port | awk -F' ' '{ print $2 }'`

openstack@control:~/devstack/context_im_user$ nova boot --flavor 1 --nic port-id=$PORT_ID --image $IMAGE_ID  --key_name
noms_key server4  --meta greeting='Hi from Krakow'

...skipping output

openstack@control:~/devstack/context_im_user$ NET_ID=`neutron net-list | grep public | awk -F' ' '{ print \$2 }'`

openstack@control:~/devstack/context_im_user$ neutron floatingip-create --port_id=$PORT_ID $NET_ID

Created a new floatingip:

+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    | 10.1.0.7                             |
| floating_ip_address | 172.18.30.6                          |

...skipping output
```

SSH to Server from Outside Network

```
[root@Cobbler ~]# scp 172.18.31.1:/tmp/id_rsa .

The authenticity of host '172.18.31.1 (172.18.31.1)' can't be established.

RSA key fingerprint is f8:49:5d:4d:3b:04:96:70:1e:e0:e6:5b:d8:17:f2:37.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '172.18.31.1' (RSA) to the list of known hosts.


root@172.18.31.1's password:

id_rsa
100% 1679     1.6KB/s   00:00


[root@Cobbler ~]# ssh -i id_rsa -l ubuntu 172.18.30.6



...skipping output

Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-34-virtual x86_64)

...skipping output


ubuntu@server4:~$
```

Browse Meta Data from VM

```
ubuntu@server4:~$ curl http://169.254.169.254/

1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
latest

ubuntu@server4:~$ curl http://169.254.169.254/latest/meta-data/public-keys/

0=noms_key
```

Browse Meta Data from VM

```
ubuntu@server4:~$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDXRfXP3i2MoCyLrTKNmbObAGmSIurBHvJKzKLl5rju9hld8OOgrVHJexvatGlzhRFs30InY9xlw/
xcqeN2bDkr6w5Dh06Z41wA8rcQeqKDmIRskrABcalFtQ03vV3p+PgDTZEOmAFW4xRsqDrsqyrzb+IItnHZV/rH0Es3daal
+DIuaOQOTya9wNXVwJJdypKaGeSRbqKnsv8KgJA7riWZuLNYxqEgaAt4FugmsmvLoRX95d9bjAw/
JnEQdHe0lCKKfbIwjDYBAx6ctSjuPcGXR1TNwI5hPGnV06r3sU8adGIlRD4Ow64sbty35V0OtZXezZcwW1XwOP4v4WnV0noJ openstack@control

ubuntu@server4:~$ cat .ssh/authorized_keys

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDXRfXP3i2MoCyLrTKNmbObAGmSIurBHvJKzKLl5rju9hld8OOgrVHJexvatGlzhRFs30InY9xlw/
xcqeN2bDkr6w5Dh06Z41wA8rcQeqKDmIRskrABcalFtQ03vV3p+PgDTZEOmAFW4xRsqDrsqyrzb+IItnHZV/rH0Es3daal
+DIuaOQOTya9wNXVwJJdypKaGeSRbqKnsv8KgJA7riWZuLNYxqEgaAt4FugmsmvLoRX95d9bjAw/
JnEQdHe0lCKKfbIwjDYBAx6ctSjuPcGXR1TNwI5hPGnV06r3sU8adGIlRD4Ow64sbty35V0OtZXezZcwW1XwOP4v4WnV0noJ openstack@control

ubuntu@server4:~$ dpkg --get-selections | grep -i cloud

cloud-init                                      install
cloud-initramfs-growroot                        install
cloud-initramfs-rescuevol                       install
cloud-utils                                     install
```

Browse Meta Data from VM

```
ubuntu@server4:~$ curl http://169.254.169.254/openstack/latest/meta_data.json

{"random_seed": "AmZkWggOH5eKAPfZUbiZPCx/Z02jYXNSY1e0bbmv4rmFTfeBXK3zPec/FehD5T5q8rM/
hPPChV5rGTI00rMpLH5q1hHyrrlqOactWiNBz7283ZSN5WCNHr6W01mfBsK8m4UrKTTZLw3jmmK+bPmdlRrwBkJukJ9yDSohz8nw8XbU+ER
+dwDLTrxAg729tOxmPiWS59f+xruqrdPvjHfJjlKXktQaT34pDLPMp0FKTzxOaXTm3kfBmRORZ2pBYldDxPzBoF6lQDKRrTuBn/
3fR5gREWkTADAceu82k8tPRT7gNDz3uc1S/QmU9DNht40WgQInV3A9pj6EnO/0lsjGggOIzT3PnGCGZRt+qrASctd9bD6vbiJgS4mwta3NLxciylnP2CEn4xrG/LS/
1FND1rXRJizV49GleoljaMxdWPcOgU51WcmKiFUvOvEoaHuNnc8qadFSEc53rAnINXAhWMMh
+n3S8pxBEMr8k99WNKkDdNlbefubHFMdkKT01rQ7VjwNTIpmAikvgQHG3IgJVn7CkhtVbD40E8TpTLUEdYhd55G4ca1pTJAuRdr6+dOQfN1QSv2Y+lzO6cvao
+YjU9Qi9hGTqMriZukw9j1/9cjbDOUfrHbQKjIm6H4Uql8HpKw8lfsM4fhYzIMMVDaSeiR2AwOZT2EII8y07U2K4Bk=", "uuid": "2eeb7abd-bcb0-48ce-9cdb-
c2f295b3d9bb", "availability_zone": "nova", "hostname": "server4.novalocal", "launch_index": 0, "meta": {"greeting": "Hi from
Krakow"}, "public_keys": {"noms_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDXRfXP3i2MoCyLrTKNmbObAGmSIurBHvJKzKLl5rju9hld8OOgrVHJexvatGlzhRFs30InY9xlw/
xcqeN2bDkr6w5Dh06Z41wA8rcQeqKDmIRskrABcalFtQ03vV3p+PgDTZEOmAFW4xRsqDrsqyrzb+IItnHZV/rH0Es3daal
+DIuaOQOTya9wNXVwJJdypKaGeSRbqKnsv8KgJA7riWZuLNYxqEgaAt4FugmsmvLoRX95d9bjAw/
JnEQdHe0lCKKfbIwjDYBAx6ctSjuPcGXR1TNwI5hPGnV06r3sU8adGIlRD4Ow64sbty35V0OtZXezZcwW1XwOP4v4WnV0noJ openstack@control\n"}, "name":
"server4"}
```

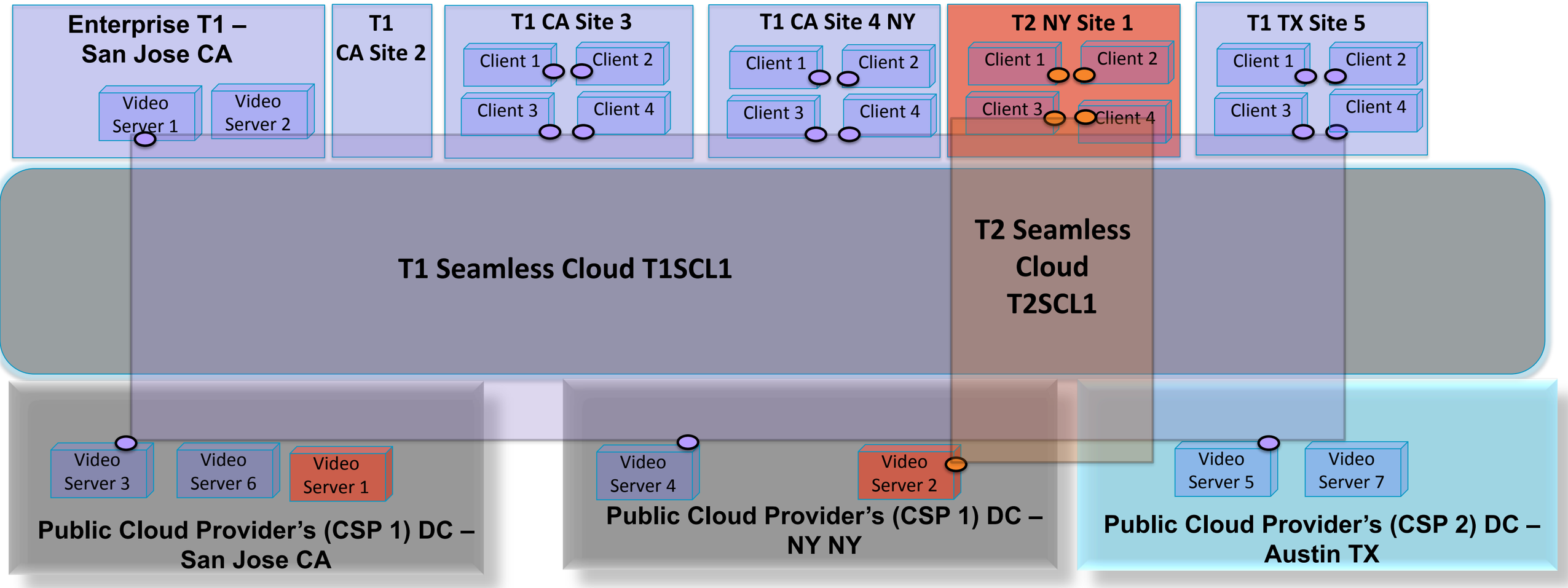
Cloud-init

- Software package that handles early initialization of a cloud instance.
- Behavior can be configured via user-data
  - Nova boot –user-data
- Some of the things it configures are:
  - setting a default locale
  - setting hostname
  - generate ssh private keys
  - adding ssh keys to user's .ssh/authorized_keys so they can log in
  - setting up ephemeral mount points

Virtual Clouds over Multiple Clouds: Seamless Virtual Clouds

Distributed Media Application Seamless Cloud

- A Cloud with resources spanning multiple sites and multiple Clouds
- A Tenant sees it as one Seamless Cloud → As one Subnet even though widely distributed over multiple network segments and Clouds
- Seamless is provided by employing and programming various features in network



Distributed Media Application Seamless Cloud

Demo Video

Q&A

- Q&A