
TOMORROW
starts here.
CISCO
Cisco live!

OpenStack Deployment in the Enterprise

BRKDCT-2367

Shannon McFarland – CCIE #5245

Principal Engineer

@eyepv6

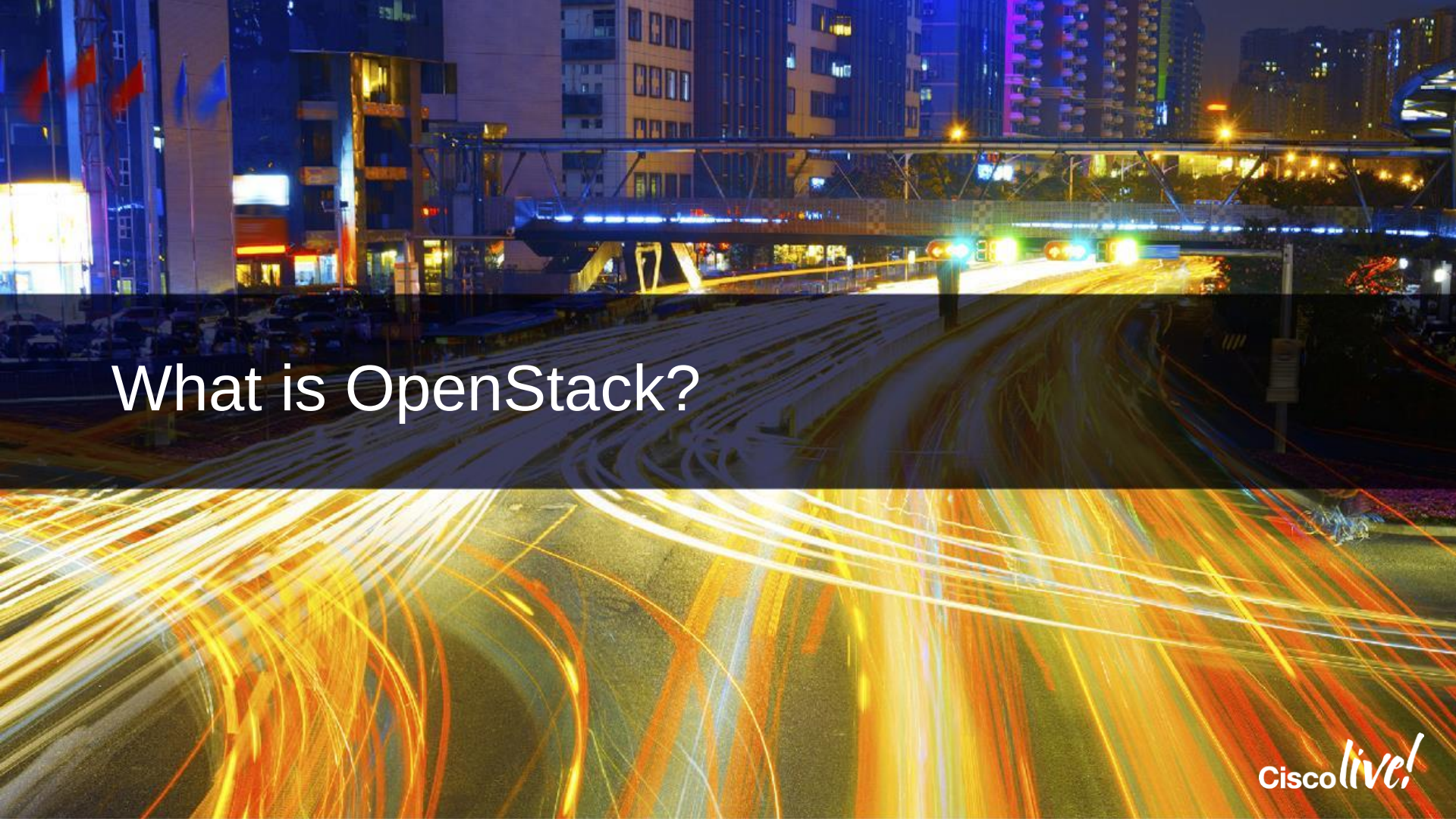
Cisco Live DISTINGUISHED Speaker

Cisco live!

Agenda

- What is OpenStack?
- OpenStack Participation
- OpenStack Deployment in the Enterprise
- Deployment Walk-thru
- Cisco Product Integration
- Demo
- Conclusion

© 2014 Cisco and/or its affiliates. All rights reserved.



What is OpenStack?

Cisco live!

OpenStack

- “OpenStack is a collection of open source technologies delivering a massively scalable cloud operating system” - openstack.org
- Basically, it is a full open source cloud stack that can be used as a starting point for a private or public cloud
- Releases are on a 6-month interval: Havana (November 2013 is the latest release and Icehouse is next
- Unreal community growth since its inception
- Timeline:
  - NASA Launches Nebula - One of the first cloud computing platforms built for Federal Government Private Cloud
  - March 2010: Rackspace Open Sources Cloud Files software, aka Swift
  - May 2010: NASA open sources compute software, aka “Nova”
  - June 2010: OpenStack is formed
  - July 2010: The inaugural Design Summit
  - April 2012: OpenStack foundation formed
  - November 2013: Havana released
  - April 2014: Icehouse Released
- OpenStack is not a 1:1 replacement for your fill-in-the-blank DC Server Virtualization Platform

© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

OpenStack is “Project” Based

Compute

“Nova”

- Houses VMs
- API driven
- Support for multi-hypervisors

Storage

Image, Object, Block

“Glance, Swift, Cinder”

- Instance/VM image storage
- Cloud object storage
- Persistent block level storage

Dashboard

“Horizon”

- Web app for controlling OpenStack resources
- Self-service portal

Identity

“Keystone”

- Centralized policies
- Tenant mgmt.
- RBAC
- Ext. integration (LDAP)

Networking

“Neutron”

- Networking as a service
- Multiple models
- IP address mgmt.
- Plugins to external HW

Metering

“Telemetry”

- Central collection point
- Metering and monitoring

Orchestration

“Heat”

- Template-based orchestration engine
- More rapid deployment of applications

Database

“Trove”

-DBaaS

-Single-tenant DB within instance

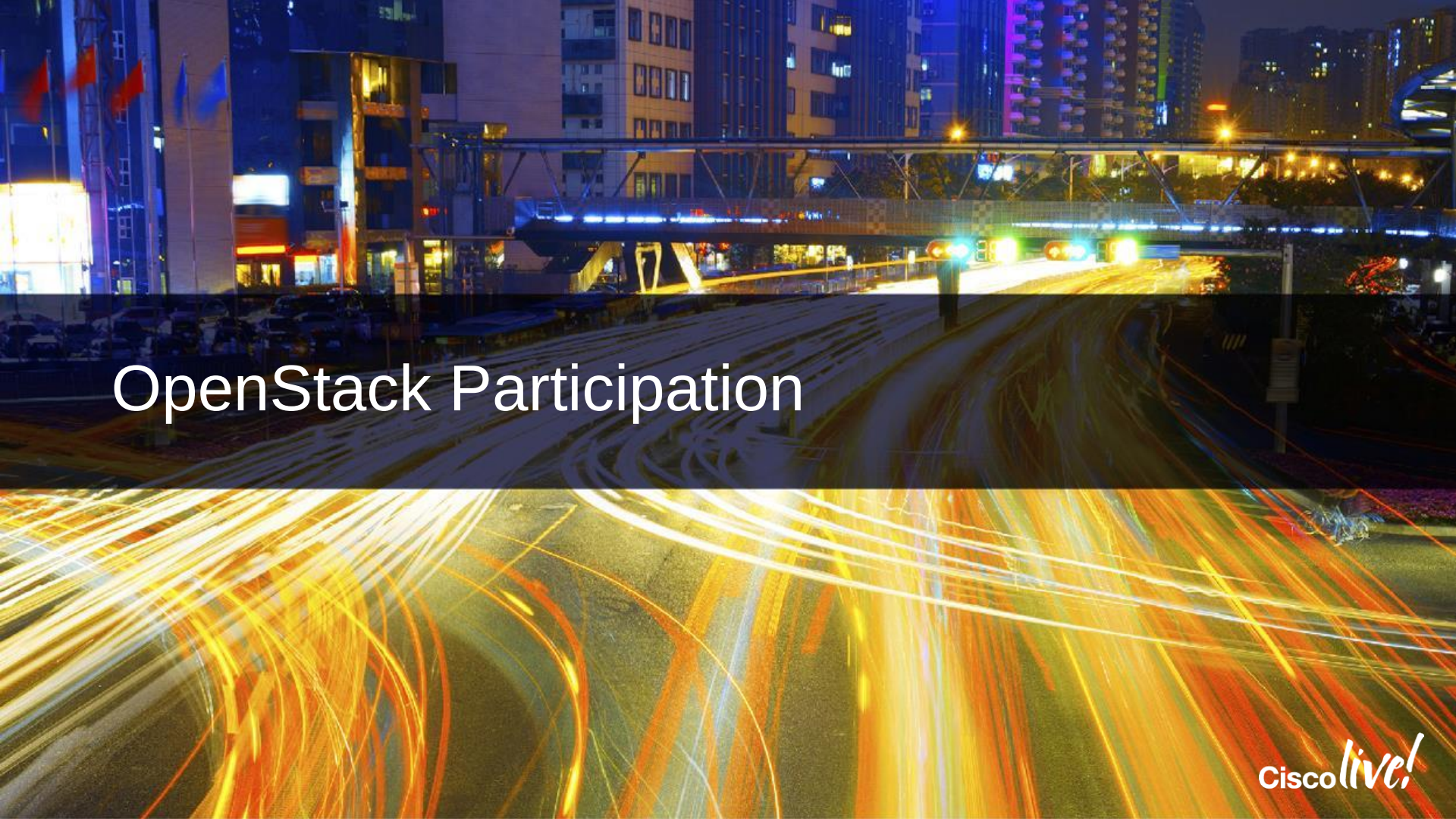
New!

© 2014 Cisco and/or its affiliates. All rights reserved.

Getting Started

- Try/Dev/Demo:
  - http://devstack.org/
  - http://www.stackops.com/
  - http://trystack.org/
- Cisco OpenStack Installer – Havana Release:
  - http://docwiki.cisco.com/wiki/OpenStack:Havana:All-in-One
  - http://docwiki.cisco.com/wiki/OpenStack:Havana:LBaaS
  - http://docwiki.cisco.com/wiki/OpenStack:Havana:VPNaaS
  - http://docwiki.cisco.com/wiki/OpenStack:_Havana:_2-Role_Nexus
  - http://docwiki.cisco.com/wiki/Openstack:Havana-Openstack-Installer
  - http://docwiki.cisco.com/wiki/OpenStack_Havana_Release:_High-Availability_Manual_Deployment_Guide

© 2014 Cisco and/or its affiliates. All rights reserved.



OpenStack Participation

Cisco live!

Who is Involved in OpenStack?

- You name it – Compute, Storage, Networking vendors, Universities, Gov't, massive pile of OpenStack-specific startups
- Traditional HW vendors – Cisco, HP, Dell, Arista, etc…
- Providers – Rackspace, AT&T, Comcast, etc…
- Startups – PistonCloud, SwiftStack and many, many more…
- Distributions & Support – Red Hat, Canonical, SUSE
- Some are focused on only small parts of OpenStack such as driving object storage features (SwiftStack), or automated deployment and support (PistonCloud) or networking and compute pull-thru as well as project leadership (Cisco – Nexus, UCS, services, Neutron)

© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco + OpenStack

- Cisco is deeply involved on many fronts and we will get even more involved over time
- Multiple groups within Cisco working on OpenStack: Design, Deployment, Product integration (UCS, Nexus, CSR, CIAC), AS, WebEx, etc…
- External portals are being developed and matured:
  - External Cisco.com: www.cisco.com/go/openstack
  - External Docwiki: http://docwiki.cisco.com/wiki/OpenStack
  - GitHub Cisco Repository: https://github.com/CiscoSystems/puppet_openstack_builder
- Multiple simultaneous efforts underway

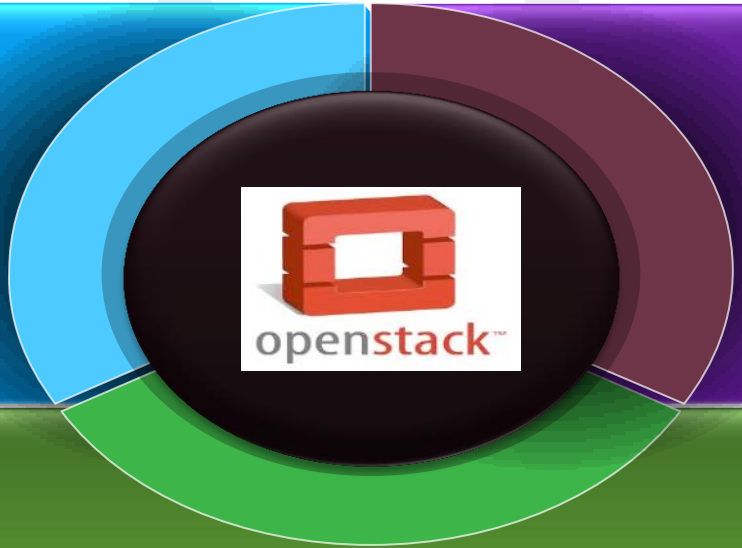
© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

Cisco's Focus on OpenStack - Today

Community

- Neutron – Network Service
- Horizon – Dashboard
- Keystone – Identity
- Swift – Object Storage
- Ceph/Cinder – Block Storage
- Automation – PuppetLabs
- HA Design



Engineering

- Cisco Product Integration
- Nexus Plugins – Neutron
- UCS
- CIAC
- Co-developed solutions (Red Hat, Canonical, SUSE)

Customers

- Cisco Designs on specific releases in 'beachhead' accounts
- Start simple, build from there – Focus on automation and HA
- Evangelization of what Cisco is doing - Thought Leadership – Help customers know What, When, Where & How

© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

Cisco + Other Distributions/Vendors

- Cisco.com OpenStack: http://www.cisco.com/web/solutions/openstack/index.html
- Red Hat: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/OpenStack/RHEL-UCS/Red-Hat-Openstack-Platform-UCS.pdf - BRAND NEW CVDhttp://www.cisco.com/c/dam/en/us/td/docs/unified_computing/ucs/UCS_CVDs/ucs_rhos.pdf http://www.cisco.com/c/dam/en/us/products/collateral/switches/nexus-7000-series-switches/wp_openstack.pdf http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/unified-fabric/solution-brief-c22-729865.pdf
- Ubuntu: http://www.cisco.com/c/dam/en/us/td/docs/unified_computing/ucs/UCS_CVDs/ucs_ubuntu.pdf
- FlexPod: http://nt-ap.com/IfgPlx
- Solution Accelerator Paks: http://www.cisco.com/web/solutions/openstack/le_sb_open.pdf

© 2014 Cisco and/or its affiliates. All rights reserved.

Reference

Distro/Vendor Supported Installers

- Red Hat OpenStack (RHOS/RDO) – PackStack and Foreman:
  http://www.redhat.com/openstack/
  http://openstack.redhat.com/Main_Page
- Canonical/Ubuntu – MAAS and JuJu: http://www.ubuntu.com/cloud
- SUSE: https://www.suse.com/products/suse-cloud/features/
- Mirantis Fuel: http://software.mirantis.com/main/
- Piston Cloud: http://www.pistoncloud.com/
- Many others …

© 2014 Cisco and/or its affiliates. All rights reserved.

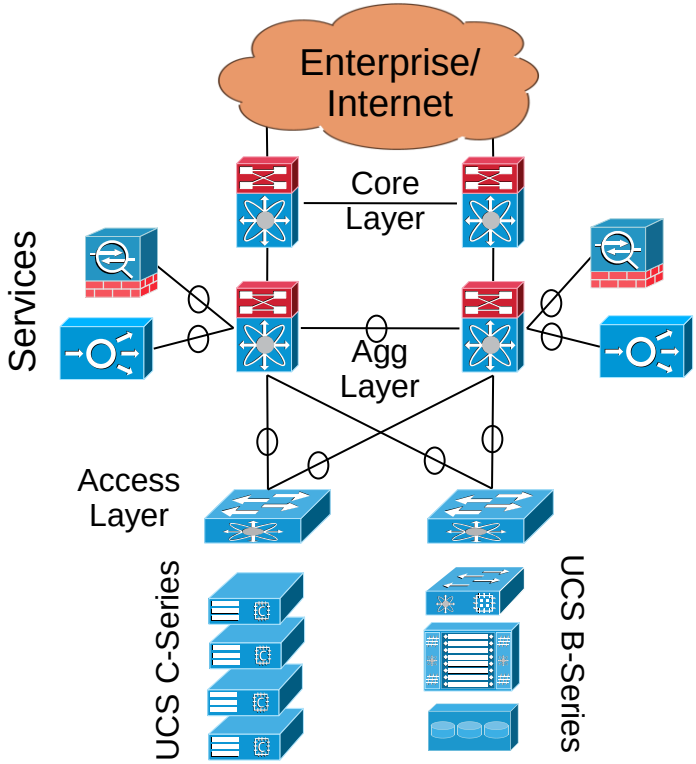
Cisco live!

OpenStack Deployment in the Enterprise

Cisco live!

What Really Changes in my Data Center/Internet Edge?

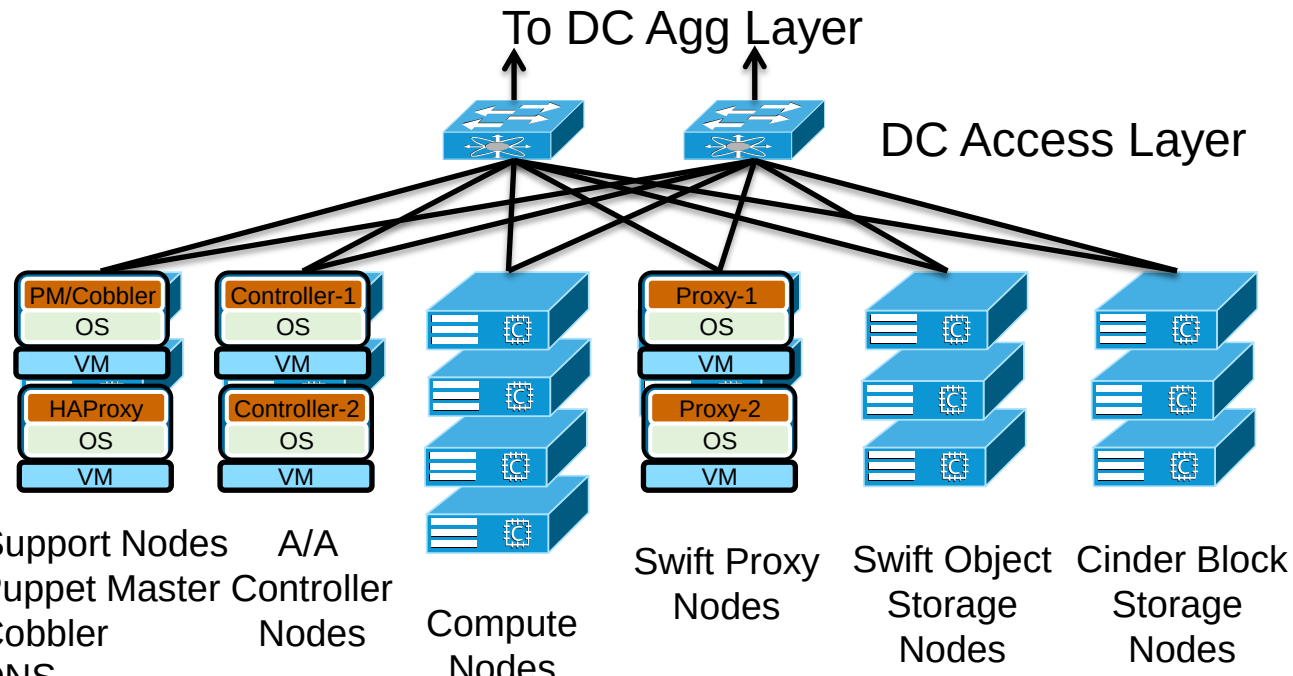
- OpenStack components live South of the Top-of-Rack switch
- Your existing DC, Internet Edge and BN architecture stays the same
- It's about the compute, storage and orchestration/management tiers
- Even your apps go largely unchanged



© 2014 Cisco and/or its affiliates. All rights reserved.

OpenStack Nodes/Roles

- Example on UCS C-series
- Active/Active controllers
- HAProxy/Keepalived or HW SLB for Swift Proxy Nodes
- Object and block storage
  - Images, app data
  - Persistent storage
- Support nodes (Ctrl/Proxy also) often run as VMs or can be baremetal



*Can run as VMs

© 2014 Cisco and/or its affiliates. All rights reserved.

To Automate or Not and How Much to Automate

- Manually deploy it all? Automate only the OpenStack setup? Automate OpenStack + Apps?
- Single Shot – Manually setup everything (the best way to learn OpenStack):
  - http://docwiki.cisco.com/wiki/OpenStack_Havana_Release:_High-Availability_Manual_Deployment_Guide
- Semi-Automatic – Use automation for ‘some’ of the setup and maintain/modify manually:
  - http://docwiki.cisco.com/wiki/Openstack:Havana-Openstack-Installer
  - http://docwiki.cisco.com/wiki/OpenStack:Havana:All-in-One
  - http://puppetlabs.com/
  - http://www.opscode.com/chef/
  - https://juju.ubuntu.com/
- Automatic – Automate everything with Puppet, Chef, JuJu or turnkey automation stuff

© 2014 Cisco and/or its affiliates. All rights reserved.

High-Level Planning Summary

- Deploy OpenStack in existing ‘pod’ or a new one?
- Hardware inventory – All rack servers, all blade servers, HW + VMs
- What app(s) do you plan to run in the new deployment?
- To multi-tenant or not? This is a functional and business topic as much as a technical one – Always deploy with multi-tenancy in mind
- IP address planning – NAT inside OpenStack? No NAT? Overlapping IPs?
- Automation choices
- Use a ‘pure’ OpenStack (only OpenStack projects) deployment or a hybrid deployment where you use some of what OpenStack offers and leverage 3rd party applications/management/monitoring services
- Knowing the limitations of current high-availability/disaster-recovery (HA/DR) models with OpenStack
- Other stuff we will talk about along the way ….

© 2014 Cisco and/or its affiliates. All rights reserved.

Network Decisions

- OpenStack Networking
  - http://docs.openstack.org/admin-guide-cloud/content/section_networking-scenarios.html
  - Many vendor plugins (OVS, Ryu, etc..)
  - Flat, Routers with NAT, VLAN Trunking, GRE, VXLAN
- Scale
  - VLAN number limitations for large tenant + networking environments
  - GRE-based
  - VPNaaS – Manual configuration in large full-mesh setup
- Network Tuning – Linux kernel, networking and vSwitch-specific (OVS) tuning is critical:
  - **libvirt_type**: **kvm** or **qemu**
  - vhost-net (‘**modprobe vhost-net**’): http://www.linux-kvm.com/content/how-maximize-virtio-net-performance-vhost-net https://ask.openstack.org/en/question/6140/quantum-neutron-gre-slow-performance/
  - Test Offload settings: ‘**ethtool -K eth1 gro off**’ - http://www.linuxcommand.org/man_pages/ethtool8.html

© 2014 Cisco and/or its affiliates. All rights reserved.

Which Networking Model?

- A few choices: Private Networks with Per-Tenant Routers, Provider Routers, Provider Network Extensions with VLANs (No NAT)
- Most enterprises use the VLAN model as they have no need for NAT within the OpenStack system – Most of their NAT stuff is on the edge (i.e. edge FW, SLB, Proxy, Routers)
- Very large enterprise deployments will run into VLAN numbering limitations when the system is deployed in a brownfield design (sharing VLANs with other PODs)
- Also, know that, today, a Neutron router-free deployment limits capabilities such as VPNaaS and/or LBaaS which depend on the L3-agent (Neutron router)

© 2014 Cisco and/or its affiliates. All rights reserved.

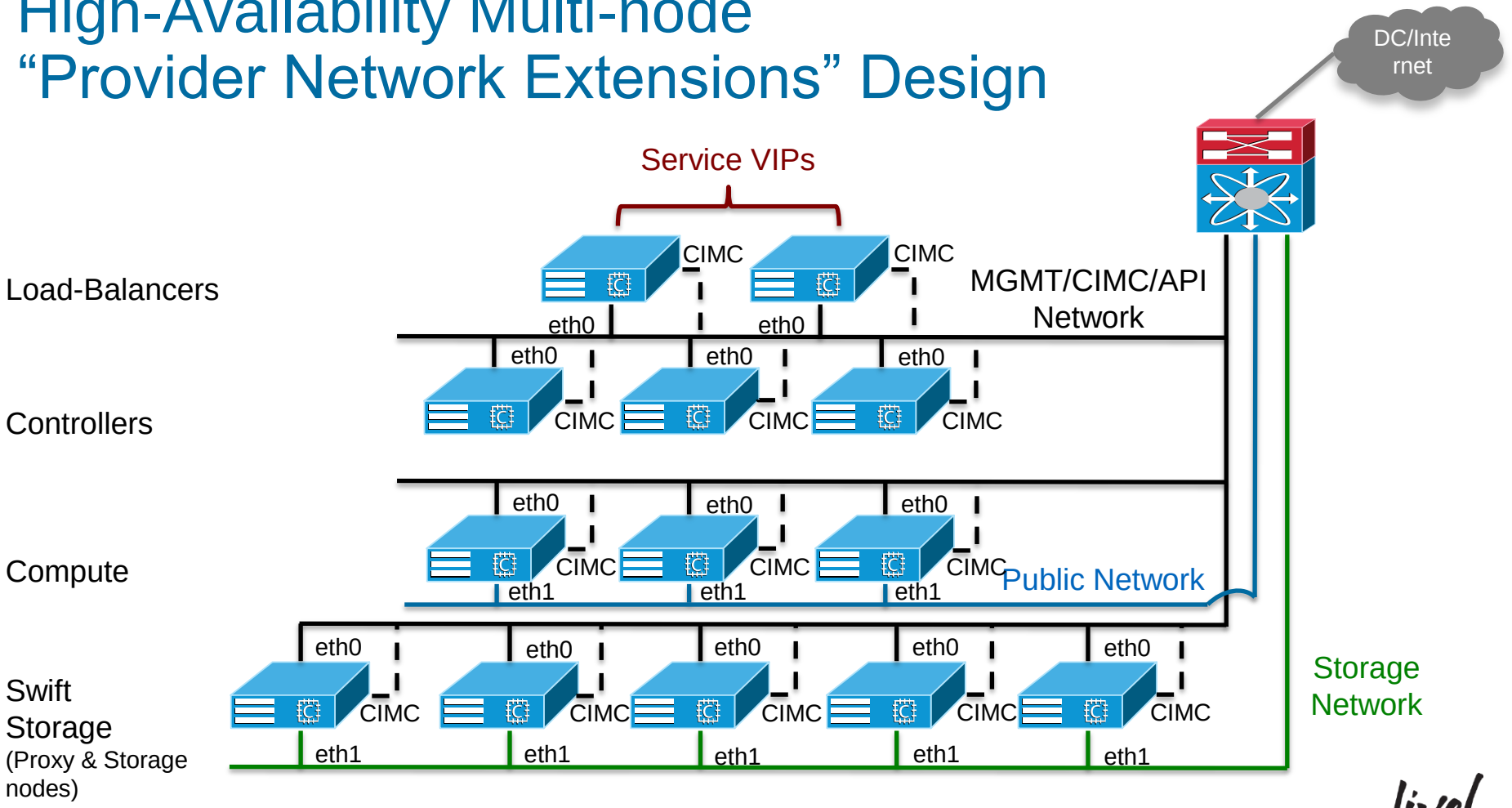
Cisco live!

High Availability Decisions

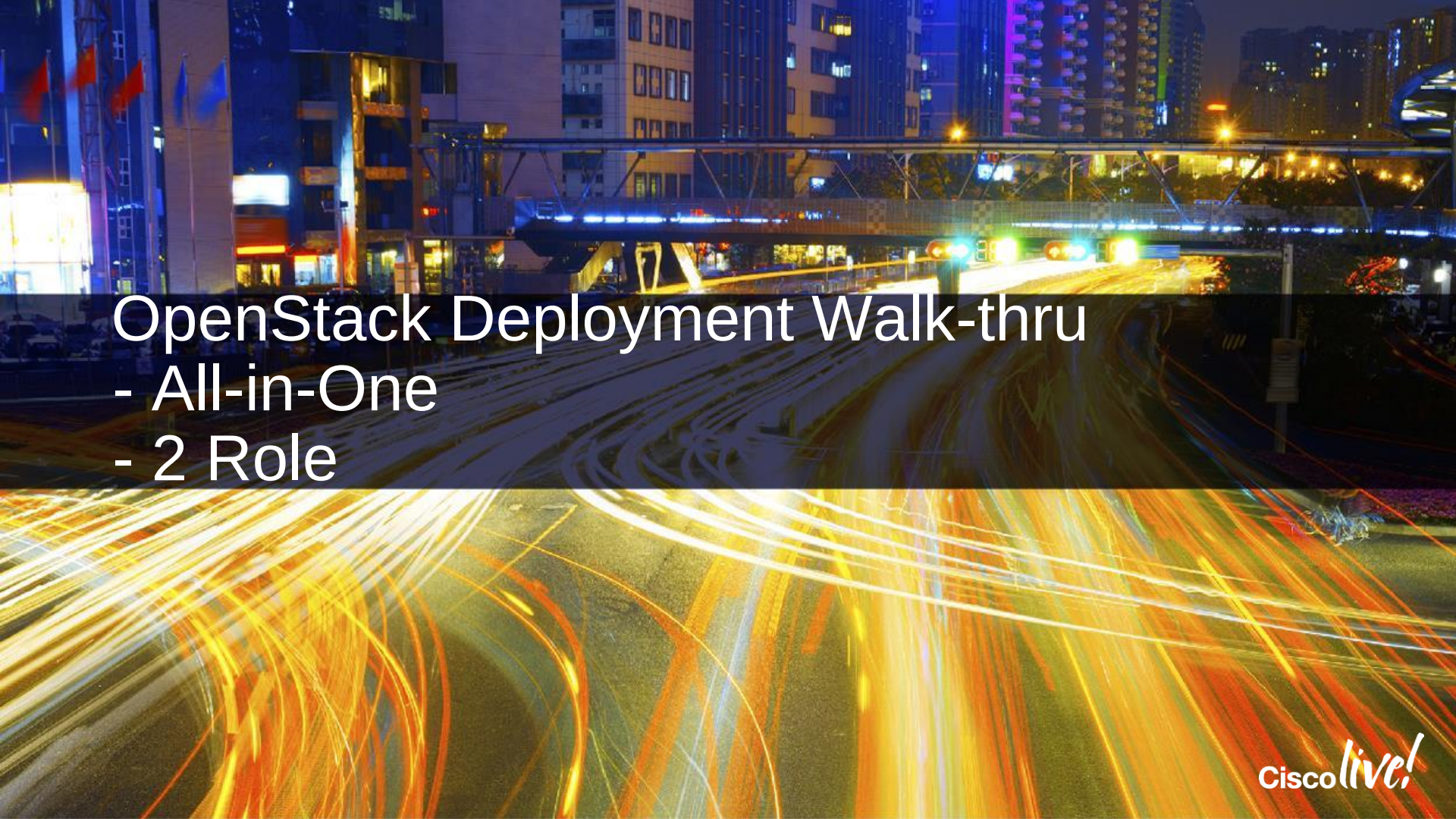
- Know what you don't know
- Pick your release – HA matures on every release: Folsom (sucked for HA) -> Grizzly (getting better) -> Havana (MUCH better)– You may have to use other open source tools to get a complete system highly available
- Cisco HA design – http://docwiki.cisco.com/wiki/OpenStack_Havana_Release:_High-Availability_Manual_Deployment_Guide
- Automated using Compressed HA (3 nodes) or Full HA (redundant control nodes, swift proxies, swift storage nodes) - http://docwiki.cisco.com/wiki/Openstack:Havana-Openstack-Installer
- Many components are:
  - Databases: Options include MySQL-WSREP and Galera
  - Message Queue: RabbitMQ Clustering and RabbitMQ Mirrored Queues
  - API/Web services: HAProxy, Keepalived, traditional SLB
  - Swift proxy nodes: HAProxy, Keepalived, traditional SLB
  - Swift nodes: Architecturally designed to be available (i.e. multiple copies of objects)
  - Compute node: Nothing directly HA, but can use Migration for planned maintenance windows
- Puppet HA: Search "puppet master redundancy" or "masterless puppet" – you will land plenty of reading choices ;-)

© 2014 Cisco and/or its affiliates. All rights reserved.

High-Availability Multi-node "Provider Network Extensions" Design



© 2014 Cisco and/or its affiliates. All rights reserved.


OpenStack Deployment Walk-thru
- All-in-One
- 2 Role
Cisco live!

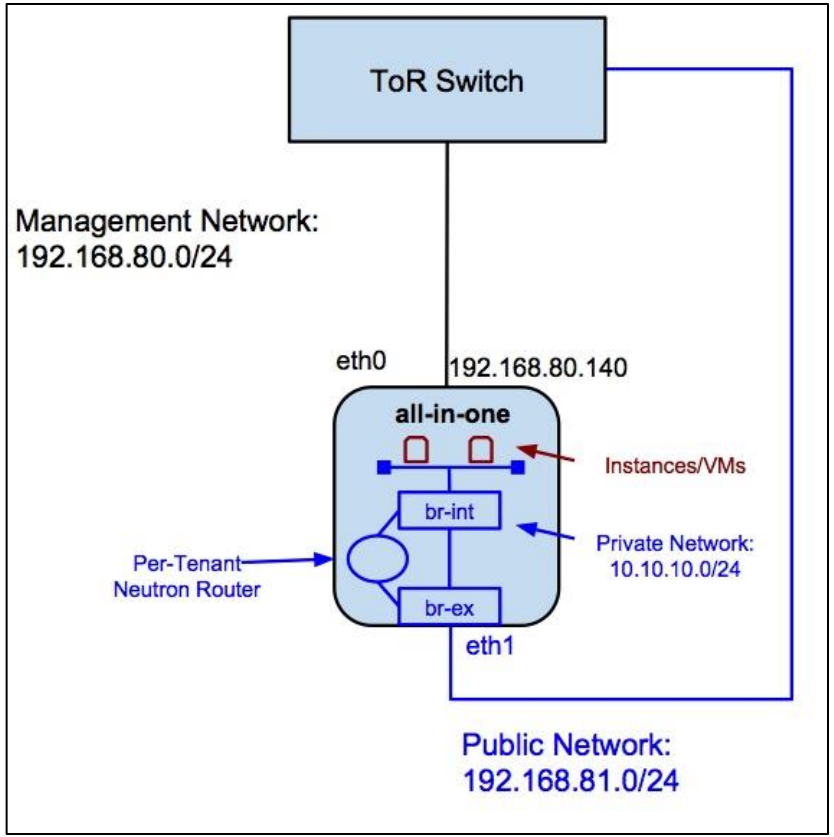
Getting Started – All-in-One (AIO) Deployment

- A much simpler process for Havana release – Using the Puppet Hiera Data Model: http://docwiki.cisco.com/wiki/Openstack:Havana-Openstack-Installer
- The All-in-One Scenario is a fast and simple way to get started with OpenStack and provides a baseline setup to add compute nodes, various networking and storage models
- Should be used for testing, proof-of-concepts (PoC) and, potentially, for small production deployments (personal cloud, branch deployment):http://docwiki.cisco.com/wiki/OpenStack:Havana:All-in-One
- Cisco AIO includes:
  - Single node for Control, Compute, Cinder, LBaaS, FWaaS, VPNaaS and non-redundant Swift testing
  - Uses Open vSwitch (OVS) with Private Networks and Per-Tenant Routers as default

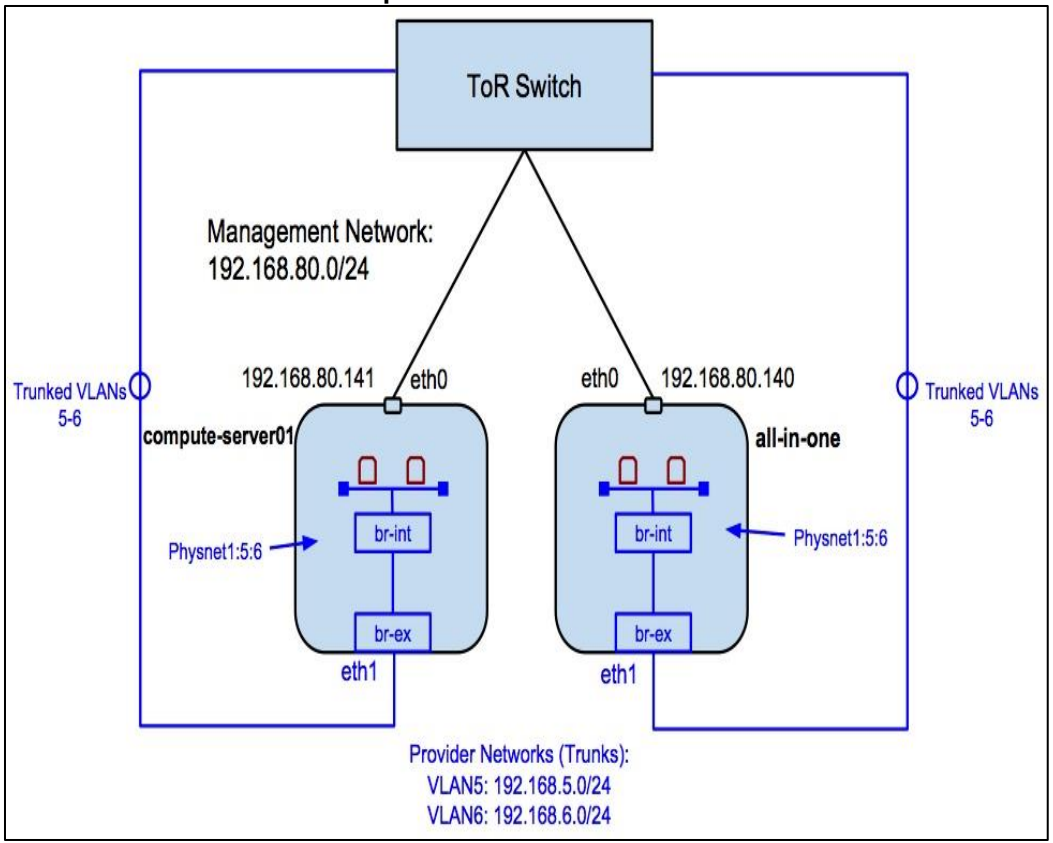
© 2014 Cisco and/or its affiliates. All rights reserved.

Multiple AIO Networking Scenarios

AIO – Private Network



AIO + Compute – VLAN Network



© 2014 Cisco and/or its affiliates. All rights reserved.
Cisco Public

AIO node

- Manually install Ubuntu 12.04 LTS (Havana) or 14.04 (Icehouse) or use automated method (PXE)
- Easiest to set the hostname to ‘all-in-one’ but you can use another hostname just so you update the /root/puppet_openstack_builder/data/role_mappings.yaml file
- Setup a second interface (next slide) to be used for bridging of VM/Instance traffic to external destinations
- Run as root or equivalent
- Minimum HW I run AIO on is (can be less for your testing):
  - 20 GB HD
  - 4 GB Memory (depending on how many instances your run)
  - Can run on real HW or VMs (VMware Fusion/Workstation, VirtualBox, etc..) – Need to enable nested VM support

© 2014 Cisco and/or its affiliates. All rights reserved.

AIO - Interfaces

```
# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.80.140
        netmask 255.255.255.0
        network 192.168.80.0
        broadcast 192.168.80.255
        gateway 192.168.80.1
        # dns-* options are implemented by the resolvconf package, if installed
        dns-nameservers 8.8.8.8 8.8.4.4
        dns-search example.com

auto eth1
iface eth1 inet manual
        up ifconfig $IFACE 0.0.0.0 up
        up ip link set $IFACE promisc on
        down ifconfig $IFACE 0.0.0.0 down
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Install GIT, COI, Exports

- Install git:

```
apt-get install -y
git
```

- Download the Cisco OpenStack Installer:

```
cd /root && git clone -b havana https://github.com/CiscoSystems/puppet_openstack_builder
&& cd puppet_openstack_builder && git checkout h.2
```

- Perform the export of vendor and scenario:

```
export vendor=cisco
export scenario=all_in_one
```

- Run the install script:

```
cd ~/puppet_openstack_builder/install-scripts
./install.sh 2>&1 | tee install.log
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Non-Standard Ethernet Interfaces or Hostname

- The install script assumes you are using eth0 and eth1, if not then set the default_interface (AKA: ‘management’ interface):

```
export default_interface=eth1  # This is the interface you logged into via ssh
```

- Export the interface being used for external VM/instance access:

```
export external_interface=eth2
```

- If you have a custom hostname other than ‘all-in-one’, edit the /root/puppet_openstack_builder/data/role_mappings.yaml file – Example here is using ‘all-in-one-test1’:

```
all-in-one-test1: all_in_one
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Verification

- After the installation is complete, verify that the services are running:

```
root@all-in-one:~# nova-manage service list
Binary           Host                 Zone             Status     State Updated_At
nova-consoleauth all-in-one           internal         enabled    :-)   2014-03-11 17:34:17
nova-scheduler   all-in-one           internal         enabled    :-)   2014-03-11 17:34:16
nova-conductor   all-in-one           internal         enabled    :-)   2014-03-11 17:34:13
nova-compute     all-in-one           nova             enabled    :-)   2014-03-11 17:34:13
nova-cert        all-in-one           internal         enabled    :-)   2014-03-11 17:34:17
```

- Login into the OpenStack Dashboard (Default username/PW = admin/Cisco123):

```
http://ip-of-your-aio
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

Networking

Reference

```
# Source the "openrc" file to export authentication/tenant info
root@all-in-one:~# source openrc
# Create a neutron public network
root@all-in-one:~# neutron net-create Public_Net --router:external=True
# Create a neutron subnet for the public network. Set starting address to be higher than
upstream DC agg-layer HSRP addresses (.1, .2, .3)
root@all-in-one:~# neutron subnet-create --name Public_Subnet --allocation-pool
start=192.168.81.10,end=192.168.81.254 Public_Net 192.168.81.0/24
# Create internal (data) network used by "openstack" tenant created by puppet process
root@all-in-one:~# neutron net-create Net10
# Create a subnet for the private network. Alter DNS servers if needed.
root@all-in-one:~# neutron subnet-create --name Net10_Subnet Net10 10.10.10.0/24 --
dns_nameservers list=true 8.8.8.8 8.8.4.4
# Create a neutron router
root@all-in-one:~# neutron router-create os-router-1
# Add neutron router interface to previously create private subnet
root@all-in-one:~# neutron router-interface-add os-router-1 Net10_Subnet
# Set the neutron router's gateway to the public network (Like a default gw)
root@all-in-one:~# neutron router-gateway-set os-router-1 Public-Net
```

Modify Security Group Rules – Download Images

- Example permits ICMP and SSH:

```
neutron security-group-rule-create --protocol icmp --direction ingress default

neutron security-group-rule-create --protocol tcp --port-range-min 22 --port-range-max 22
--direction ingress default
```

- Download Images:

```
wget http://download.cirros-cloud.net/0.3.1/cirros-0.3.1-x86_64-disk.img

glance image-create --name cirros-x86_64 --is-public True --disk-format qcow2 -
-container-format ovf --file cirros-0.3.1-x86_64-disk.img --progress
```

```
wget http://cloud-images.ubuntu.com/precise/current/precise-server-cloudimg-amd64-
disk1.img

glance image-create --name precise-x86_64 --is-public True --disk-format qcow2 --
container-format bare --file precise-server-cloudimg-amd64-disk1.img --progress
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Upload SSH Keys

- Generate SSH Keys or use existing keys and upload to Nova:

```
# Generate a new SSH key pair
root@all-in-one:~# ssh-keygen
# Add a new keypair into Nova
root@all-in-one:~/.ssh# nova keypair-add --pub_key ~/.ssh/id_rsa.pub aio-key
```


© 2014 Cisco and/or its affiliates. All rights reserved.

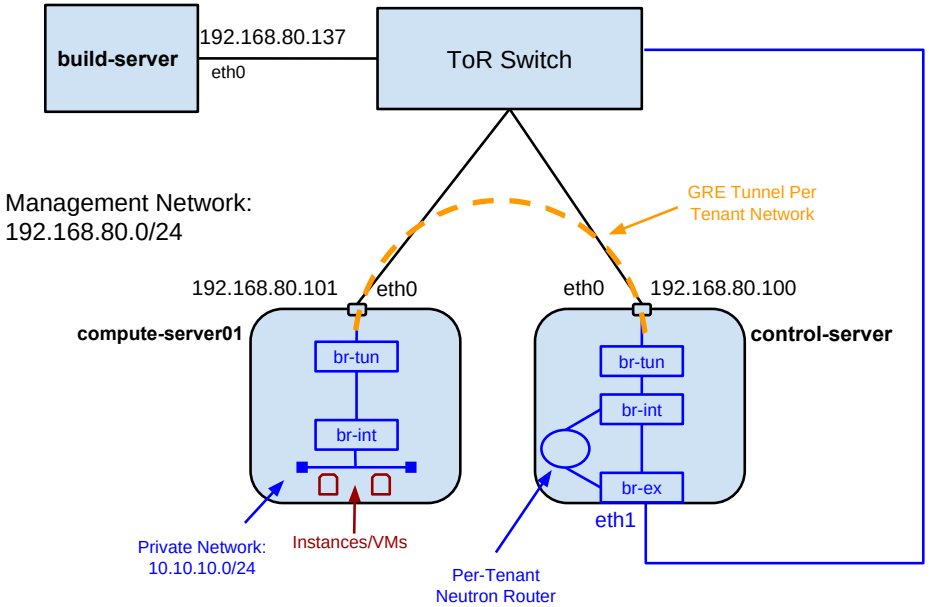
Cisco live!

2 Role Scenario

Cisco live!

Cisco Design on Havana – Automated Deployment

- 2_role scenario from: http://docwiki.cisco.com/wiki/Openstack:Havana-Openstack-Installer
- Example will include:
  - Build Server running: Puppet master, Cobbler, etc…
  - Controller node
  - Compute node
  - Ubuntu 12.04 LTS



© 2014 Cisco and/or its affiliates. All rights reserved.

Reference

Automated Deployment Steps

- **Document your addressing (IP/MAC)/node roles**
- **Cable your servers and configure any networking gear needed by OpenStack**
- **Deploy the build server**
- **Customize the build server (Modify YAML file(s))**
- **Kick-off control server and compute server(s) builds**
- Neutron setup
- Download images and upload into Glance (if not using test script in step 6)
- Boot instance, test connectivity
- Modify setup to meet your needs
- Have a nice day ☺

© 2014 Cisco and/or its affiliates. All rights reserved.

Install Ubuntu, GIT, COI, Exports

- Minimum requirements for build server: 2 GB RAM, 20 GB storage, Internet connectivity and on the same network as management interfaces of OpenStack nodes
- Install git:

```
apt-get install -y
git
```

- Download the Cisco OpenStack Installer:

```
cd /root && git clone -b havana https://github.com/CiscoSystems/puppet_openstack_builder
&& cd puppet_openstack_builder && git checkout h.2
```

- Perform the export of vendor and scenario:

```
export vendor=cisco
export scenario=2_role
```

- Run the install script”

```
cd ~/puppet_openstack_builder/install-scripts
./install.sh 2>&1 | tee install.log
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Customize YAML Files for 2 Role Scenario

- Modify the /etc/puppet/data/hiera_data/user.common.yaml file for your setup
- NOTE: The following slides show only a portion of the total options in the file:

```
# Set a proxy if you use one
#proxy: 'http://proxy-server.domain.com:8080'
# Set the domain name
domain_name: example.com
# Set the controller IP address(s)
controller_public_address: 192.168.80.100
controller_internal_address: 192.168.80.100
controller_admin_address: 192.168.80.100
# Set the interface values
internal_ip: "%{ipaddress_eth0}"
external_interface: eth1
public_interface: eth0
private_interface: eth0
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

user.common.yaml - continued

```
# Set the IP address of the build-server
cobbler_node_ip: 192.168.80.137

# Set the subnet, mask and gateway for the nodes
node_subnet: '192.168.80.0'
node_netmask: '255.255.255.0'
node_gateway: '192.168.80.1'

# The Ubuntu user and password used during PXE install
admin_user: localadmin
password_crypted:
$6$UfgWxrIv$k4KfzAEMqMg.fppmSOTd0usI4j6gfjs0962.JXsoJRWa5wMz8yQk4SfInn4.WZ3L/MCt5u.62tHD
GB36EhiKF1

# Linux install drive
install_drive: /dev/sda

# Interface the VNC proxyclient will listen on (usually same as 'public interface')
nova::compute::vncserver_proxyclient_address: "%{ipaddress_eth0}"
```

user.common.yaml - continued

```
cinder_db_password: cinder_pass
glance_db_password: glance_pass
keystone_db_password: key_pass
nova_db_password: nova_pass
network_db_password:   quantum_pass
database_root_password: mysql_pass
cinder_service_password: cinder_pass
glance_service_password: glance_pass
nova_service_password: nova_pass
ceilometer_service_password: ceilometer_pass
admin_password: Cisco123
admin_token: keystone_admin_token
network_service_password: quantum_pass
rpc_password: openstack_rabbit_password
metadata_shared_secret: metadata_shared_secret
horizon_secret_key: horizon_secret_key
ceilometer_metering_secret: ceilometer_metering_secret
ceilometer_db_password: ceilometer
heat_db_password: heat
heat_service_password: heat_pass
```

Don’t use these in production ;-)

Hostnames

- If you have custom hostnames other than what is listed in /etc/puppet/data/role_mappings.yaml file then you need to edit that file and make the changes:

```
control-server: controller
compute-server01: compute
build-server: build
```

- Also, you need to ensure that those hostnames are modified in the /etc/puppet/data/hiera_data/user.common.yaml file

© 2014 Cisco and/or its affiliates. All rights reserved.

Two Methods for Building Control/Compute Nodes

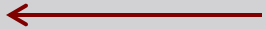
1. You can manually build each node by hand – Ubuntu 12.04 LTS install, interface configuration, patches, etc…
2. Use the build-server with Cobbler and PXE booting

- If you use option 2 then you will need to gather the IP address and MAC info for each node and edit the file /etc/puppet/data/cobbler/cobbler.yaml

© 2014 Cisco and/or its affiliates. All rights reserved.

cobbler.yaml

```
kopts: "netcfg/get_nameservers=8.8.8.8 \
netcfg/no_default_route=false \
netcfg/get_gateway=192.168.80.1 \
control-server:
  hostname: "control-server.example.com"
  power_address: "192.168.80.4"
  interfaces:
    eth0:
      mac-address: "00:50:56:3A:83:A3"
      dns-name: "control-server.example.com"
      ip-address: "192.168.80.100”
compute-server01:
  hostname: "compute-server01.example.com"
  power_address: "192.168.80.5"
  interfaces:
    eth0:
      mac-address: "00:50:56:37:EA:CA"
      dns-name: "compute-server01.example.com"
      ip-address: "192.168.80.101”
```



Super important:

Default is to not set a ‘gateway’ on nodes.

Change to “false” to enable default gw

Build Server – Apply the Manifest

- Run “puppet apply” against the site.pp file we just built -

```
root@build-server:~# puppet apply -v /etc/puppet/manifests/site.pp
```

- Puppet apply will install the following on the build server as well as prepare for deployment of the OpenStack nodes we defined earlier:
  - ntpd – Time synchronization
  - tftpd-hpa – TFTP server for PXE boot of OpenStack nodes
  - dnsmasq – DNS and DHCP server
  - cobbler – Installation and boot management
  - apt-cacher-ng – Caching proxy for package installation
  - nagios – Infrastructure monitoring application
  - collectd – Statistics collection
  - graphite/carbon – Real-time graphing system
  - apache – Web server for hosting graphite, nagios and puppet services

© 2014 Cisco and/or its affiliates. All rights reserved.

Begin Node Deployment

- Deploy each node:

```
cd /root/puppet_openstack_builder/scripts
bash clean_node.sh control-server
```

```
cd /root/puppet_openstack_builder/scripts
bash clean_node.sh compute-server01
```

- Each node will be rebooted/powered-on (If using CIMC)
- Ubuntu will be installed
- Puppet agent will start the install process for OpenStack

© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

Cisco Product Integration - Nexus

Cisco live!

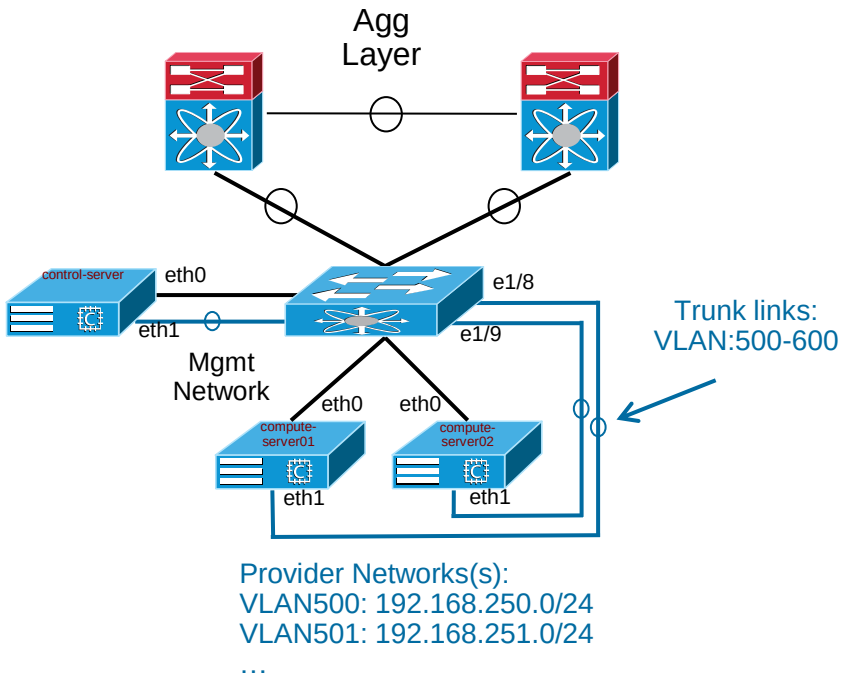
Nexus – Support for OpenStack

- Nexus 1000
  - Red Hat and Ubuntu - KVM
  - 512 servers per VSM and scaling to future with federations
  - VLAN - 4096, VXLAN – 16000 segments, 32000 ports, 300+ veths/vem
  - Enhanced VXLAN – No multicast requirement in a VSM and in future across VSMs
  - VSM on any hypervisor or Nexus1010
  - NAT is supported/overlapping IP support

  http://www.cisco.com/c/en/us/support/switches/nexus-1000v-kvm/tsd-products-support-series-home.html
- Nexus 3000 and Higher
  - http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps11541/data_sheet_c78-727737.html
- Cisco OpenStack Installer with Nexus Plugin:
  http://docwiki.cisco.com/wiki/OpenStack:Grizzly-Nexus-Plugin
  http://docwiki.cisco.com/wiki/OpenStack:_Havana:_2-Role_Nexus

© 2014 Cisco and/or its affiliates. All rights reserved.

Nexus Plugin Example Topology



- Stuff we care about in the user.common.yaml file that are relevant to the diagram:
  - Switch ports that connect to the eth1 on each compute node
  - That the appropriate interface on the controller is configured to trunk all of the same VLANs that will be used by instances (attached to eth1 on compute nodes)
  - That the uplinks from ToR to Agg layer switches has all of the trunks/VLANs configured ahead of time
- Multiple ToR switches and host FEX setups are supported

© 2014 Cisco and/or its affiliates. All rights reserved.
Cisco Public

Steps to Enable Nexus Integration

- In the /etc/puppet/data/global_hiera_params/common.yaml file change the network_type to ‘vlan’:

```
tenant_network_type: vlan
```

- In the /etc/puppet/data/hiera_data/tenant_network_type/vlan.yaml file set the VLAN and OVS mappings:

```
neutron::agents::ovs::bridge_mappings:
  - "physnet1:br-ex"
neutron::plugins::ovs::network_vlan_ranges: physnet1:500:600
neutron::plugins::ovs::tenant_network_type: vlan
neutron::agents::ovs::enable_tunneling: false
```

- In the /etc/puppet/data/class_groups/network_controller.yaml file uncomment:

```
- "coe::nexus"
- "neutron::plugins::cisco"
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Steps to Enable Nexus Integration - Continued

- In the /etc/puppet/data/data_mappings/common.yaml file uncomment:

```
# Cisco plugin support
nexus_plugin:
  - neutron::plugins::cisco::nexus_plugin
vswitch_plugin:
  - neutron::plugins::cisco::vswitch_plugin
nexus_credentials:
  - coe::nexus::nexus_credentials
nexus_config:
  - coe::nexus::nexus_config
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Steps to Enable Nexus Integration - Continued

- In the /etc/puppet/data/hiera_data/user.common.yaml file update the Nexus info:

```
neutron::core_plugin: 'neutron.plugins.cisco.network_plugin.PluginV2'
nexus_plugin: 'neutron.plugins.cisco.nexus.cisco_nexus_plugin_v2.NexusPlugin'
vswitch_plugin: 'neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2'

nexus_credentials:
  - '10.121.10.39/username/password'

nexus_config:
  'bldr-n3k-1':
    'ip_address': '10.121.10.39'
    'username': 'username'
    'password': 'password'
    'ssh_port': 22
    'servers':
      'compute-server01': '1/8'
      'compute-server02': '1/9'
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

Example Nexus Config – Pre First Instance Boot

```
interface Ethernet1/1
  description to N7k-agg-1
  switchport mode trunk
  switchport trunk allowed vlan 13,500-600

interface Ethernet1/2
  description to N7k-agg-2
  switchport mode trunk
  switchport trunk allowed vlan 13,500-600

interface Ethernet1/5
  description to control-server Management
  switchport mode trunk
  switchport trunk allowed vlan 13
  speed 1000

interface Ethernet1/6
  description to control-server Data
  switchport mode trunk
  switchport trunk allowed vlan 13,500-600
  speed 1000
```

```
interface Ethernet1/7
  description to compute-server01 Management
  switchport mode trunk
  switchport trunk allowed vlan 13
  speed 1000

interface Ethernet1/8
  description to compute-server01 Data
  switchport mode trunk
  speed 1000

interface mgmt0
  ip address 10.121.10.39/24
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Neutron Setup

Reference

```
# Source the "openrc" file to export authentication/tenant info

root@control-server:~# source openrc

# Create a Neutron provider network

root@control-server:~# neutron net-create vlan500 --provider:network_type vlan --
provider:physical_network physnet1 --provider:segmentation_id 500 --shared --
router:external=True

# Create a Neutron subnet for the provider network. Set starting address to be higher than
upstream DC agg-layer HSRP addresses (.1, .2, .3)

root@control-server:~# neutron subnet-create --name subnet500 --allocation-pool
start=192.168.250.10,end=192.168.250.250 vlan500 192.168.250.0/24 --dns_nameservers list=true
10.121.12.10
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

Before/After of Nexus Configuration

Before First Instance Launch:

```
interface Ethernet1/7
  description to compute-server01 Management
  switchport mode trunk
  switchport trunk allowed vlan 13
  speed 1000

interface Ethernet1/8
  description to compute-server01 Data
  switchport mode trunk
  speed 1000

interface mgmt0
  ip address 10.121.10.39/24
```

After First Instance Launch:

```
vlan 500
  name q-500

interface Ethernet1/7
  description to compute-server01 Management
  switchport mode trunk
  switchport trunk allowed vlan 13
  speed 1000

interface Ethernet1/8
  description to compute-server01 Data
  switchport mode trunk
  switchport trunk allowed vlan 500
  speed 1000
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

LBaaS

- A service to provide basic load-balancing of VMs/Instances within the OpenStack cluster
- Can leverage plugins for LBaaS to control external virtual or physical load-balancers (i.e. F5, A10, Citrix, Cisco ACE)
- The default LB provider is HAProxy
- Used with Neutron router
- My guide on deploying and using LBaaS with a COI AIO node: http://docwiki.cisco.com/wiki/OpenStack:Havana:LBaaS
- We will walk-thru this in the demo!

© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

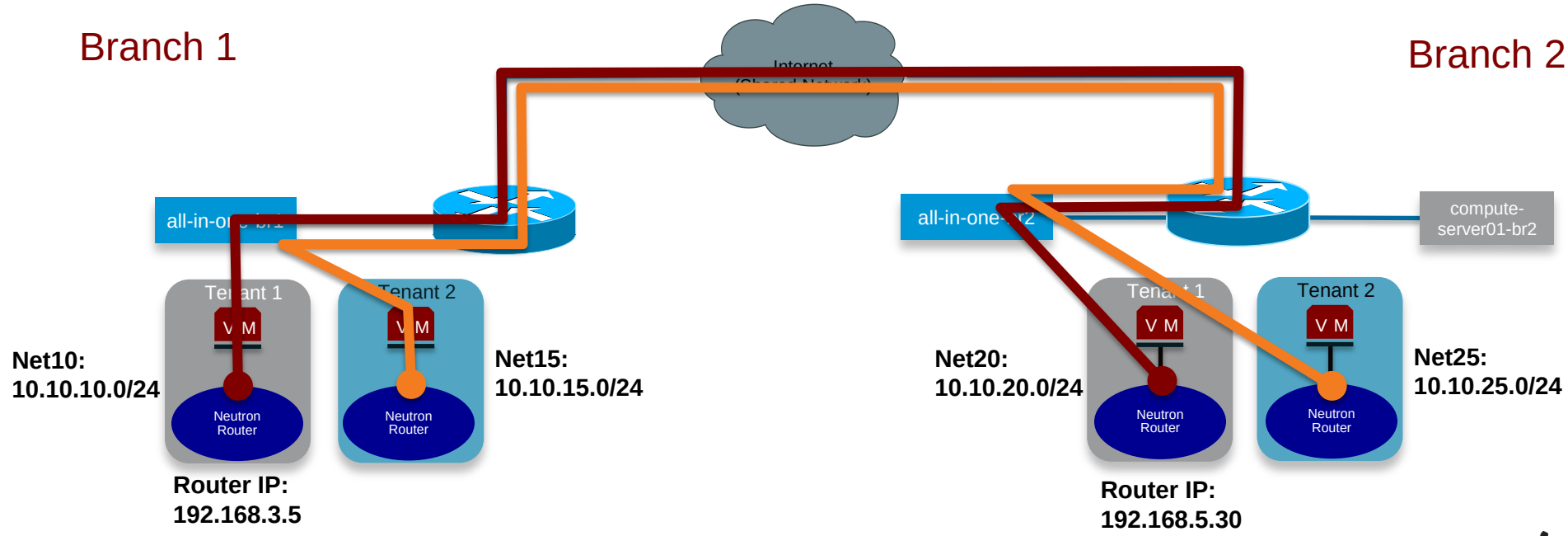
VPNaaS

- A service to provide IPsec VPN connectivity on a per-router/per-tenant basis
- Manual configuration via CLI or OpenStack Dashboard
- As with any IPsec site-to-site VPN, large deployments with lots of sites/tenants will require a lot of configuration due to mesh-type connectivity
- Used with Neutron router
- Cisco provides CSR as a means of deploying VPN as well
  - CSR Example with Devstack: http://docwiki.cisco.com/wiki/Install_and_Setup_of_Cisco_Cloud_Services_Router_(CSR)_for_OpenStack_VPN
- My guide on deploying VPNaaS using two AIO nodes: http://docwiki.cisco.com/wiki/OpenStack:Havana:VPNaaS

© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

VPN Topology

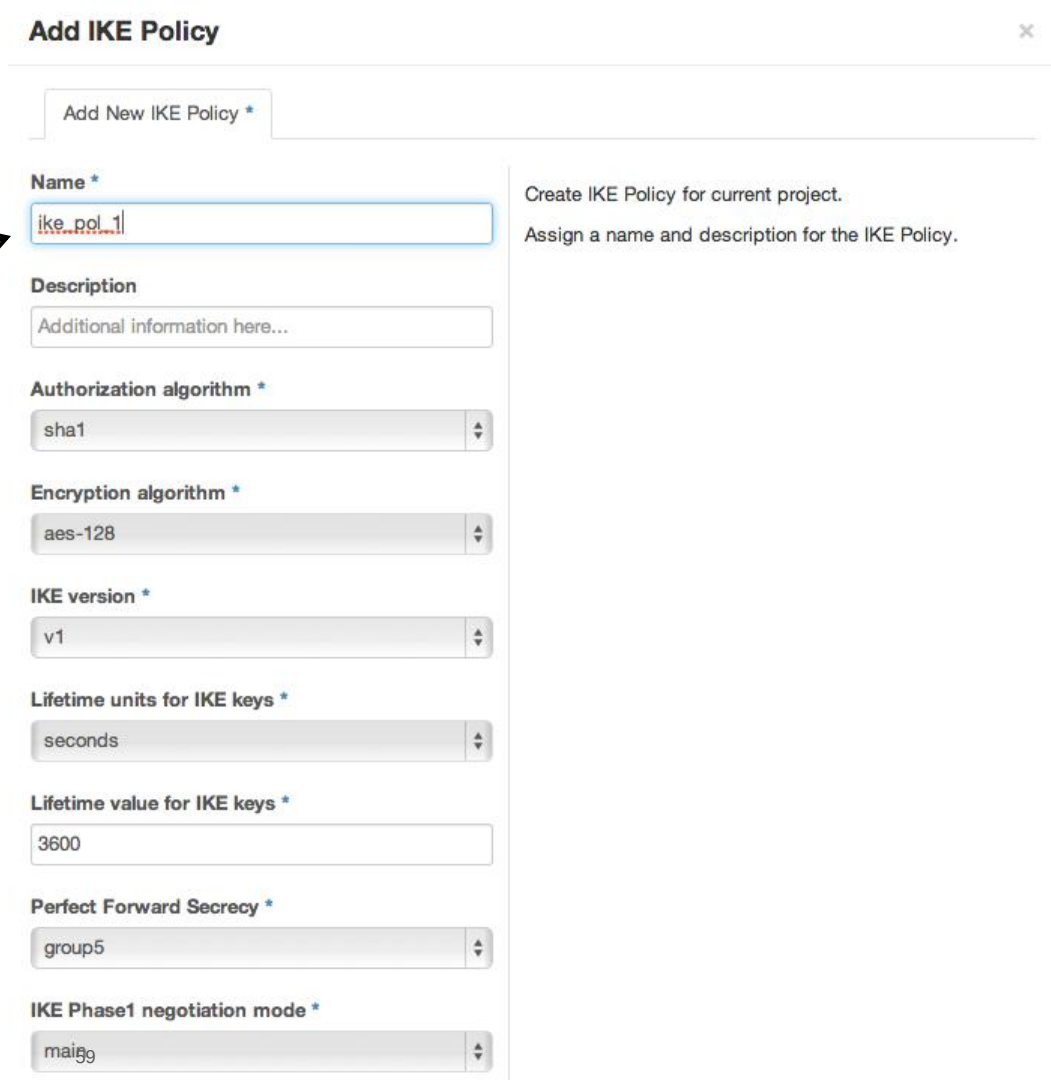


© 2014 Cisco and/or its affiliates. All rights reserved. Cisco Public

Setup IKE Policies – Dashboard Example

Add name

These are all defaults

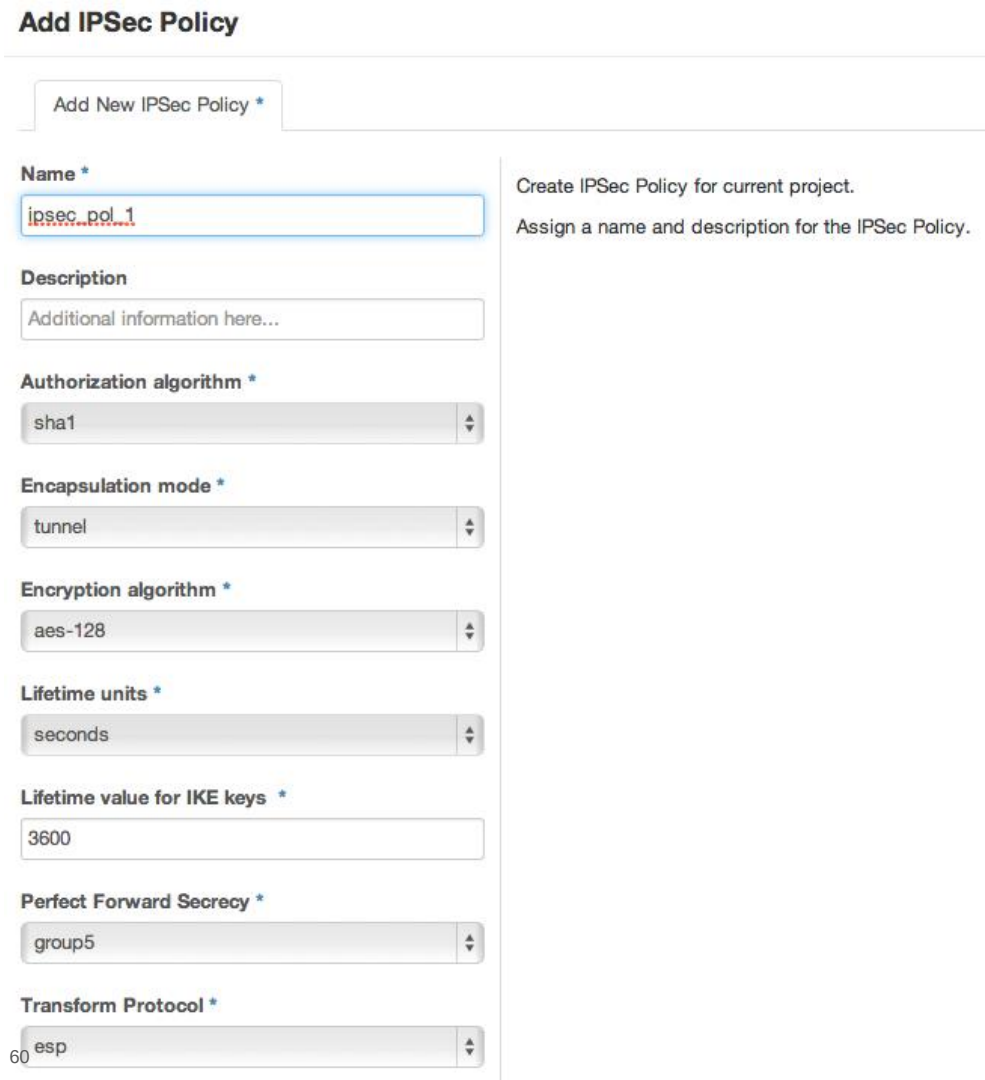


© 2014 Cisco and/or its affiliates. All rights reserved.
Cisco Public

Setup IPsec Policies – Dashboard Example

Add name

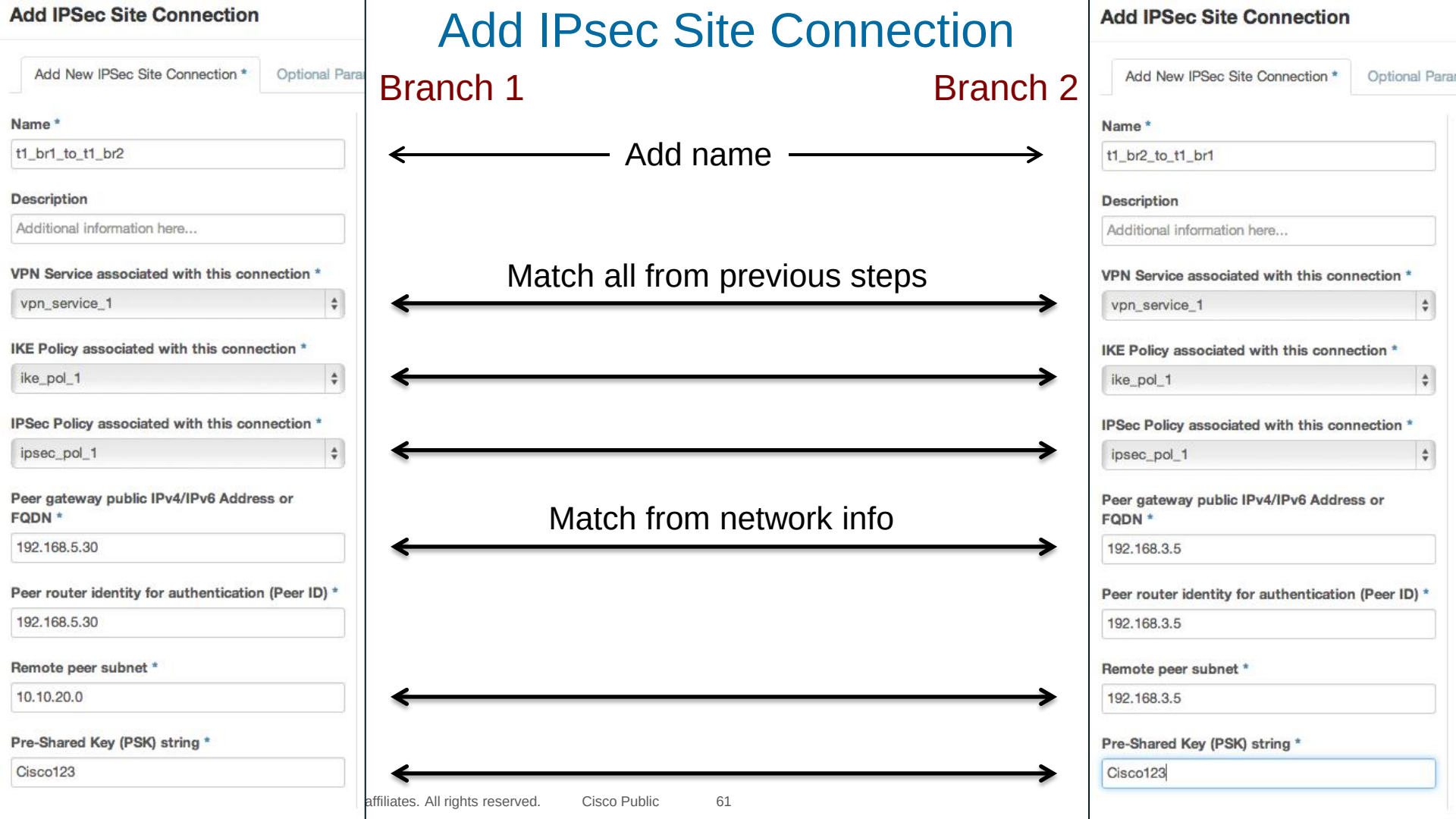
These are all defaults



© 2014 Cisco and/or its affiliates. All rights reserved.

Add IPsec Site Connection

Branch 1 Branch 2



affiliates. All rights reserved.

Launch Instances

- Note: Depending on your firewall/security group setup, you may need to open up UDP/500 for ISAKMP and IP Protocol 50 for ESP
- Boot an instance at each site within the tenant you created the VPNaaS policies:

```
# Boot an instance using ID for the "net10" network
root@all-in-one-br1:~# nova boot --image cirros-x86_64--flavor m1.tiny --key_name ctrl-key --
nic net-id=c57e37d2-4a49-4056-b119-1351906ebaf4 --security-groups Branch1 br1-instance-1


# Boot an instance using ID for the "net20" network
root@all-in-one-br2:~# nova boot --image cirros-x86_64--flavor m1.tiny --key_name ctrl-key --
nic net-id=719e1969-ee29-4a11-af3c-60fa018d9af0 --security-groups Branch2 br2-instance-1
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Initiate Connections Between Sites

- Connect to an instance and ping instance at the other site:

```
root@all-in-one:~# ip netns exec qrouter-e81de88d-d339-47c2-ae0b-37f2948d1147 ssh cirros@10.10.20.3
The authenticity of host '10.10.20.3 (10.10.20.3)' can't be established.
RSA key fingerprint is 5e:1f:9f:11:da:d4:5c:24:ed:d7:2c:37:5d:5c:40:be.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.20.3' (RSA) to the list of known hosts.
$ ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2): 56 data bytes
64 bytes from 10.10.10.2: seq=0 ttl=62 time=4.112 ms
64 bytes from 10.10.10.2: seq=1 ttl=62 time=1.689 ms
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Is it really a VPN?

- Setup:

```
09:55:52.617054 IP 192.168.5.30.isakmp > 192.168.3.5.isakmp: isakmp: phase 1 ? ident
09:55:52.623361 IP 192.168.5.30.isakmp > 192.168.3.5.isakmp: isakmp: phase 1 ? ident
09:55:52.626899 IP 192.168.5.30.isakmp > 192.168.3.5.isakmp: isakmp: phase 1 ? ident[E]
09:55:52.635606 IP 192.168.5.30.isakmp > 192.168.3.5.isakmp: isakmp: phase 2/others ? oakley-
quick[E]
09:56:18.255468 IP 192.168.5.30.isakmp > 192.168.3.5.isakmp: isakmp: phase 2/others ? inf[E]
09:56:18.255557 IP 192.168.5.30.isakmp > 192.168.3.5.isakmp: isakmp: phase 2/others ? inf[E]
```

- Connection traffic between instances:

```
09:59:05.002332 IP 192.168.5.30 > 192.168.3.5: ESP(spi=0xea65be5d,seq=0x7), length 132
09:59:06.002231 IP 192.168.5.30 > 192.168.3.5: ESP(spi=0xea65be5d,seq=0x8), length 132
09:59:07.002531 IP 192.168.5.30 > 192.168.3.5: ESP(spi=0xea65be5d,seq=0x9), length 132
```


© 2014 Cisco and/or its affiliates. All rights reserved.

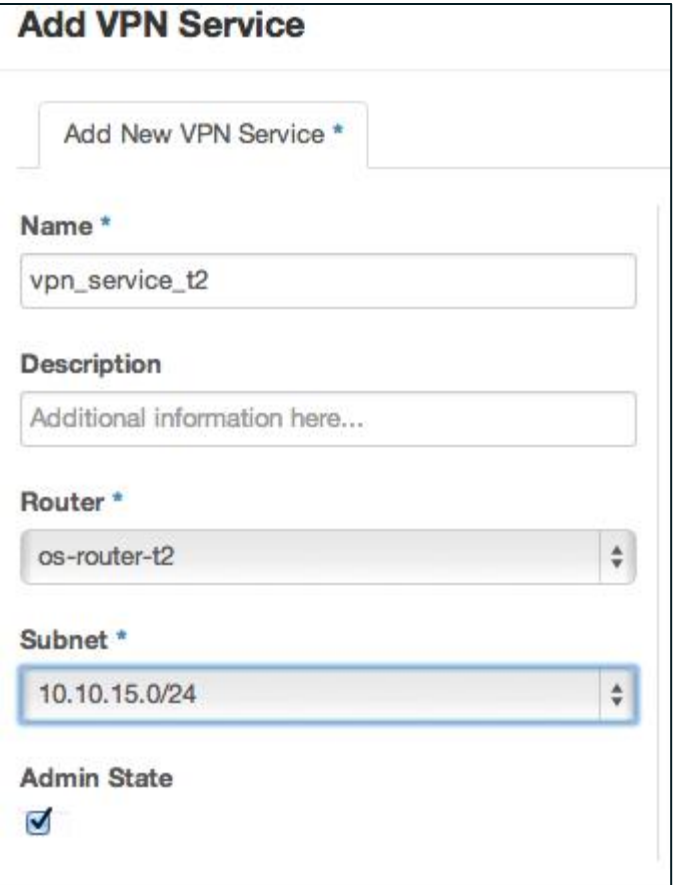
Cisco live!

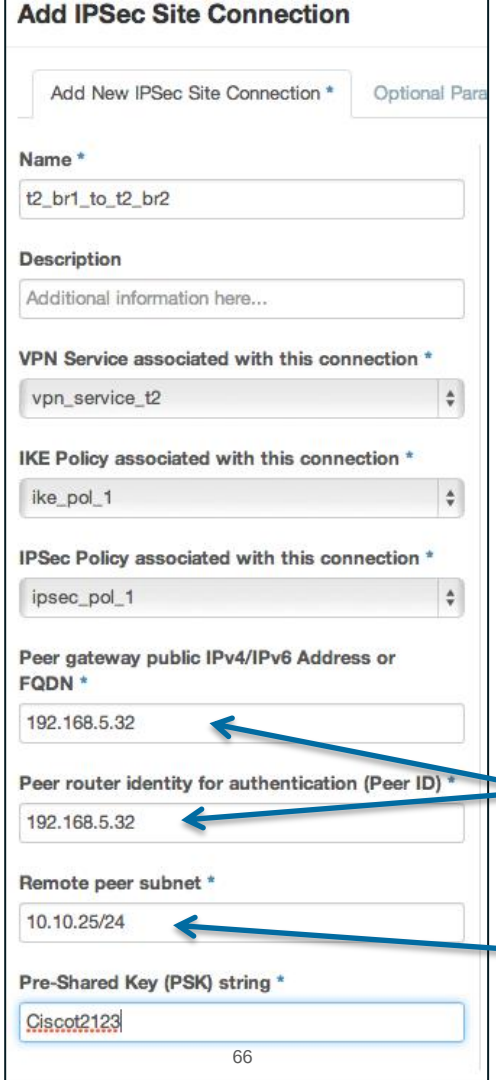
Setup Tenant 2

- Repeat previous steps:
  - Setup new Project for Tenant 2
  - Create/associate relevant Project users/groups
  - Create new private network/subnet
  - Create new Neutron Router (set gateway/add interface to private network)
  - Add VPN service using new private network and router for Tenant 2
  - Add IPsec site connection using new peer router info

© 2014 Cisco and/or its affiliates. All rights reserved.

Branch 1 – Tenant 2 Example





Reference

Branch 2 – Tenant 2 Neutron Router IP

Branch 2 – Tenant 2 Private Network

Testing Instances Between Sites – Tenant 2

Reference

- Check to see if you can connect between sites on for Tenant 2 instances BUT not Tenant 2-to-Tenant 1

```
root@all-in-one-br1:~# neutron router-list
+--------------------------------------+--------------+-----------------------------------------------------------------------+
| id                                   | name         | external_gateway_info                                                 |
|
+--------------------------------------+--------------+-----------------------------------------------------------------------+
| 689df3c4-e5ae-42e5-99f4-70e9dcb8993f | os-router-1  | {"network_id": "3964c2bd-9562-408c-9489-f1a643c128f8",
"enable_snat": true} |
| aa3e1b12-524a-460e-901f-3f9b7c8fb0e0 | os-router-t2 | {"network_id": "3964c2bd-9562-408c-9489-f1a643c128f8",
"enable_snat": true} |
+--------------------------------------+--------------+-----------------------------------------------------------------------+
root@all-in-one-br1:~# ip netns exec qrouter-aa3e1b12-524a-460e-901f-3f9b7c8fb0e0 ssh
cirros@10.10.15.2
$ ping 10.10.25.2
PING 10.10.25.2 (10.10.25.2): 56 data bytes
64 bytes from 10.10.25.2: seq=0 ttl=62 time=3.452 ms
64 bytes from 10.10.25.2: seq=1 ttl=62 time=1.766 ms
$ ping 10.10.20.2
PING 10.10.20.2 (10.10.20.2): 56 data bytes
--- 10.10.20.2 ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
```

Ping Instance on Tenant 2 – Branch 2

Ping Instance on Tenant 1 – Branch 2

© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

References for Storage Info

- OpenStack Storage: https://www.openstack.org/software/openstack-storage/
- Block Storage: http://docs.openstack.org/havana/config-reference/content/ch_configuring-openstack-block-storage.html
- Object Storage:http://docs.openstack.org/havana/config-reference/content/ch_configuring-object-storage.html
- Cinder How-to (We will Demo):http://docwiki.cisco.com/wiki/OpenStack:Havana:Cinder-Volume-Test
- Cinder Deep Dive (Grizzly): https://wiki.openstack.org/wiki/File:Cinder-grizzly-deep-dive-pub.pdf
- CEPH Storage: http://ceph.com/docs/master/rados/

© 2014 Cisco and/or its affiliates. All rights reserved.


DEMO

Conclusion

- Next time: Scale, HA, apps, network design impact and some new breakouts (OpenStack storage session)
- OpenStack is for real and maturing at a rapid pace
- Many different players involved and it is evolving rapidly
- Align yourself with market leaders who have strong partnerships
- There is still a lot of focus on getting OpenStack Deployed, but we are progressing rapidly towards true operational issues:
  - Scale
  - Application deployment
  - Upgrades
- Start now!
- Get involved in the community – open source enjoys the major advantage of feature velocity

© 2014 Cisco and/or its affiliates. All rights reserved.

Participate in the “My Favorite Speaker” Contest

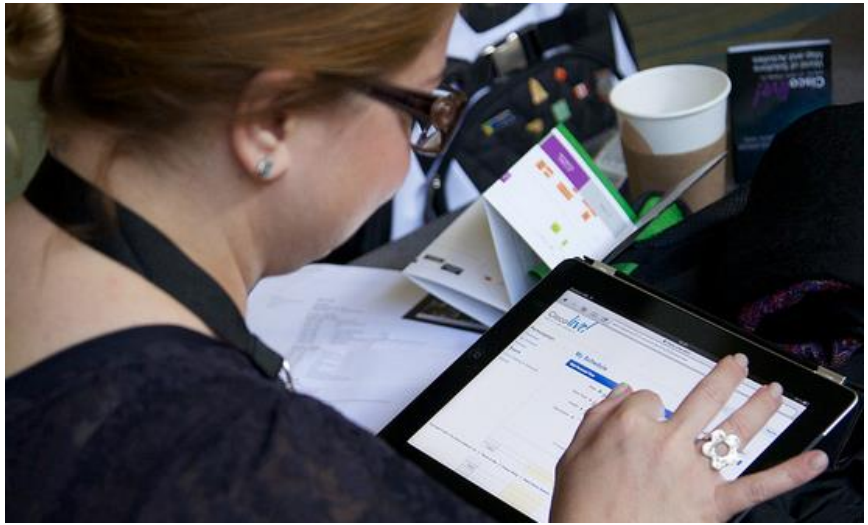
Promote Your Favorite Speaker and You Could be a Winner

- Promote your favorite speaker through Twitter and you could win $200 of Cisco Press products (@CiscoPress)
- Send a tweet and include
  - Your favorite speaker’s Twitter handle @eyepv6
  - Two hashtags: #CLUS #MyFavoriteSpeaker
- You can submit an entry for more than one of your “favorite” speakers
- Don’t forget to follow @CiscoLive and @CiscoPress
- View the official rules at http://bit.ly/CLUSwin

© 2014 Cisco and/or its affiliates. All rights reserved.

Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Complete your session evaluation through the Cisco Live mobile app or visit one of the interactive kiosks located throughout the convention center.



Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online

© 2014 Cisco and/or its affiliates. All rights reserved.

Continue Your Education

- Demos in the Cisco Campus
- Walk-in Self-Paced Labs
- Table Topics
- Meet the Engineer 1:1 meetings

© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!


Thank you.

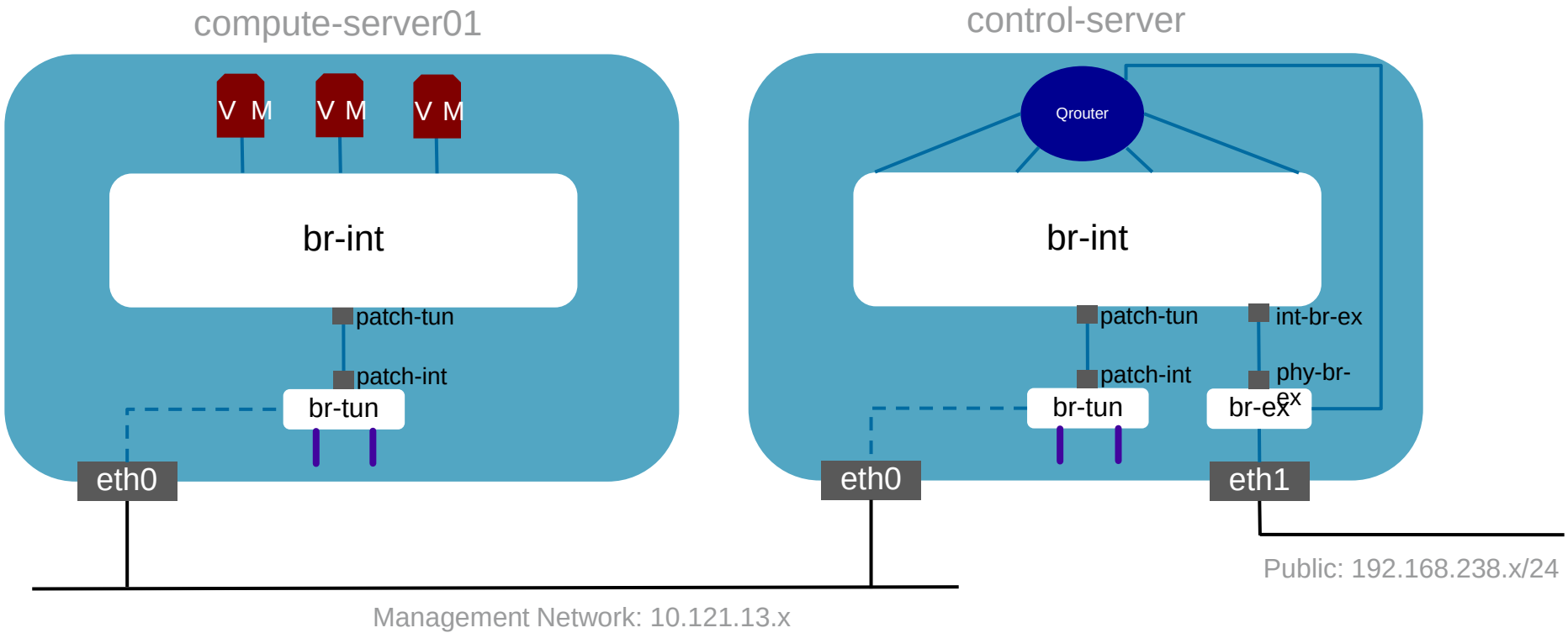

CISCO


Reference Slides


Understanding Neutron + OVS for example.com
Cisco live!

Example – Network Layout

Host View



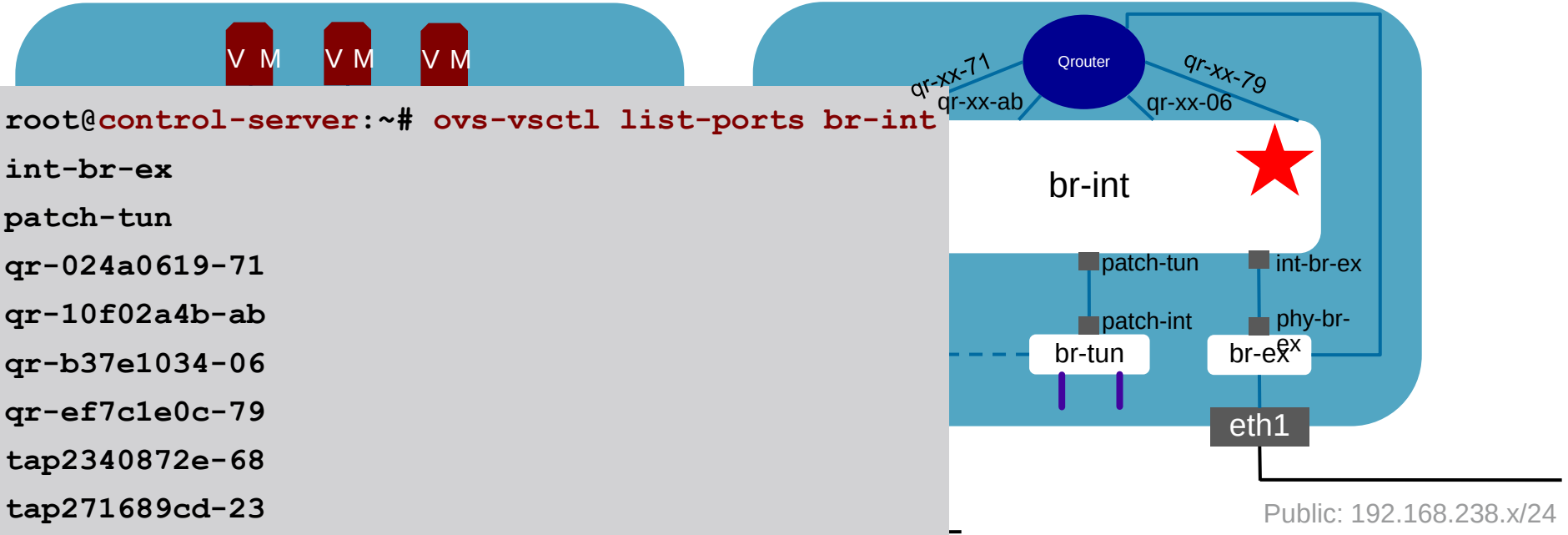
© 2014 Cisco and/or its affiliates. All rights reserved.

Example – OVS Bridge/Neutron Router

"br-int" view

compute-server01

control-server

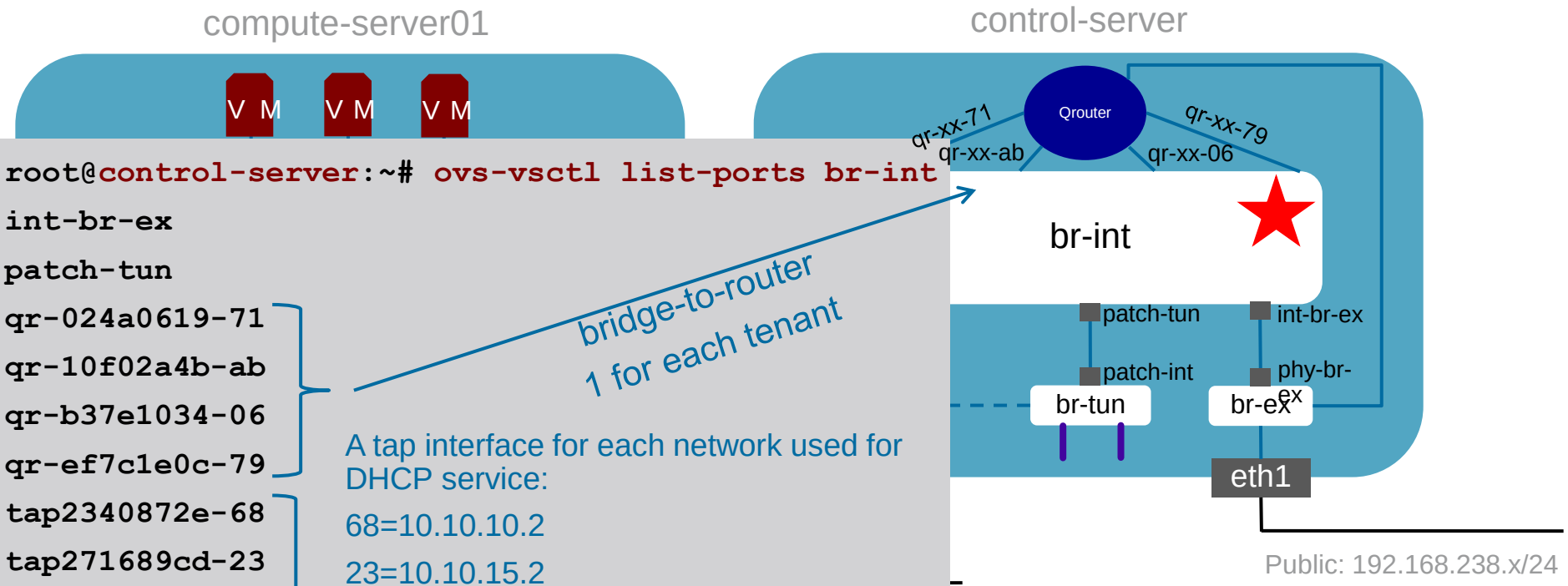



```
root@control-server:~# ovs-vsctl list-ports br-int
int-br-ex
patch-tun
qr-024a0619-71
qr-10f02a4b-ab
qr-b37e1034-06
qr-ef7c1e0c-79
tap2340872e-68
tap271689cd-23
tap3fe91abf-c8
tap60a25081-14
tap6d3911a5-44
```

Cisco live!

Example – OVS Bridge/Neutron Router

"br-int" view qr-xx & tapxx




```
root@control-server:~# ovs-vsctl list-ports br-int
int-br-ex
patch-tun
qr-024a0619-71
qr-10f02a4b-ab
qr-b37e1034-06
qr-ef7c1e0c-79
tap2340872e-68
tap271689cd-23
tap3fe91abf-c8
tap60a25081-14
tap6d3911a5-44
```

bridge-to-router
1 for each tenant

A tap interface for each network used for DHCP service:

68=10.10.10.2

23=10.10.15.2

c8=192.168.238.5

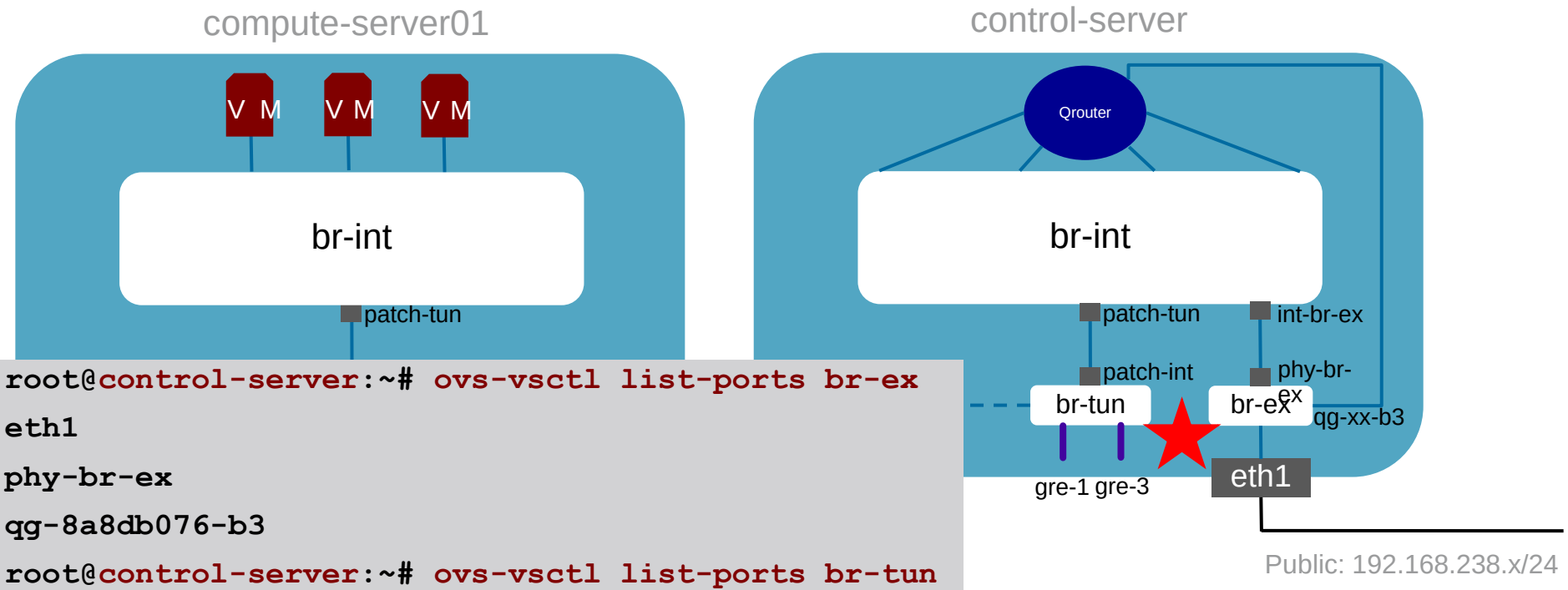
14=10.10.20.2

44=10.10.25.2

Cisco live!

Example – OVS Bridge/Neutron Router

"br-ex" & br-tun view

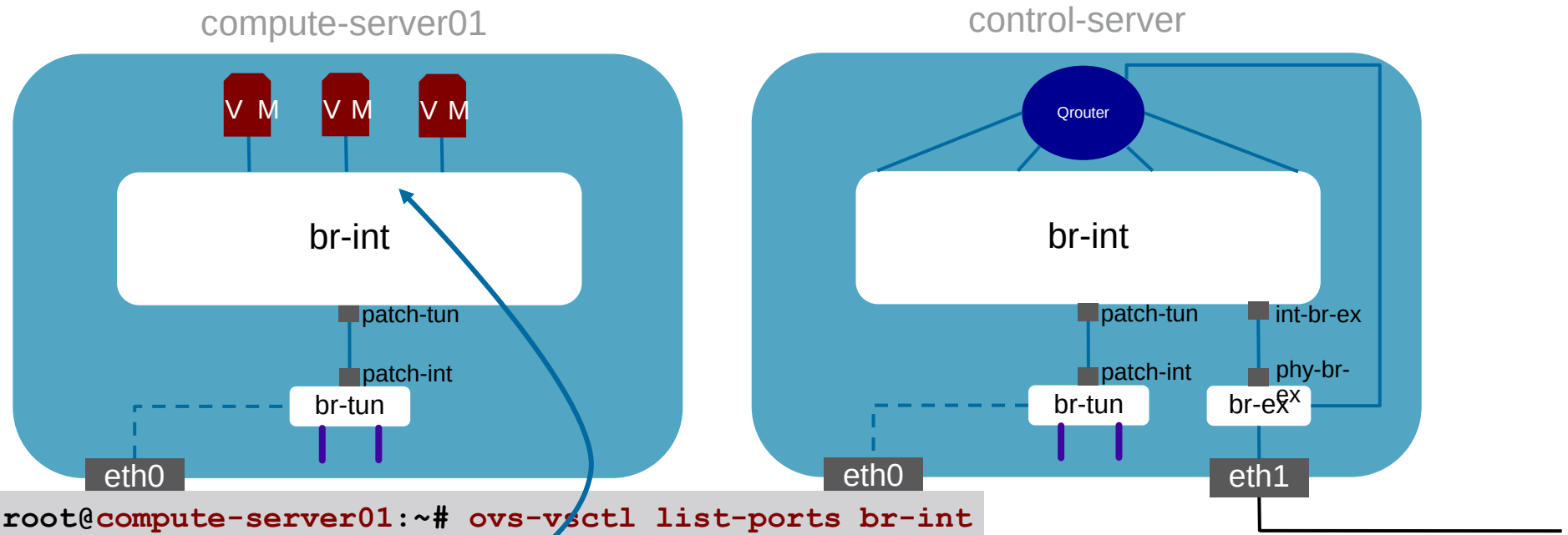



```
root@control-server:~# ovs-vsctl list-ports br-ex
eth1
phy-br-ex
qg-8a8db076-b3
root@control-server:~# ovs-vsctl list-ports br-tun
gre-1
gre-3
patch-int
```

Cisco live!

Example – OVS Bridge/Neutron Router

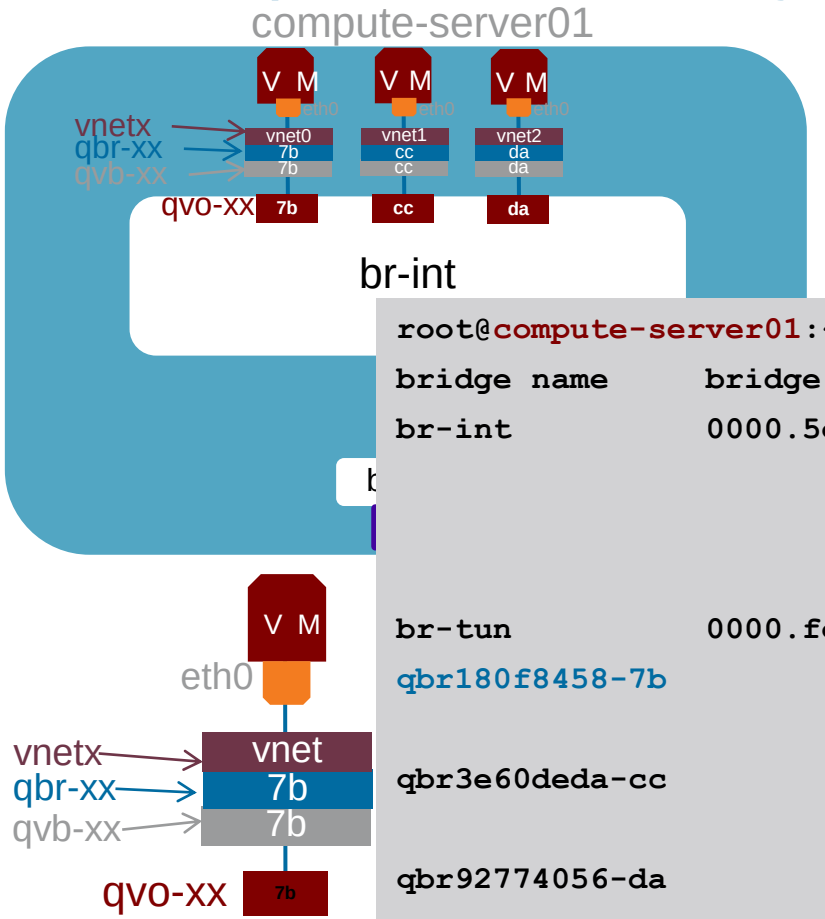
compute-server01 “br-int” view




```
root@compute-server01:~# ovs-vsctl list-ports br-int
patch-tun
qvo180f8458-7b
qvo3e60deda-cc
qvo92774056-da
```

Cisco live!

Example – OVS Bridge/Neutron Router

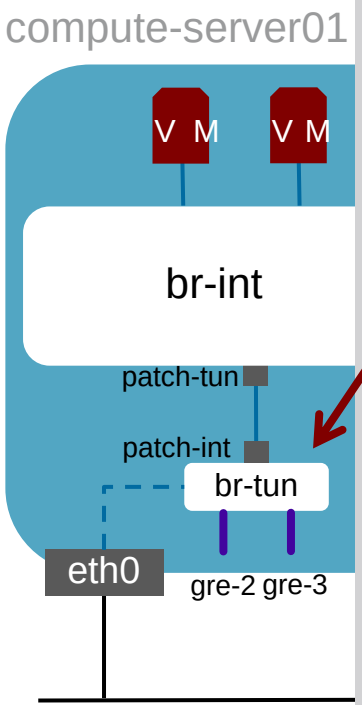


*Thanks to Etsuji Nakai for the original detailed overview of OVS/Neutron ports : http://www.slideshare.net/enakai/how-quantum-configures-virtual-networks-under-the-hood

```
root@compute-server01:~# brctl show
bridge name     bridge id               STP enabled     interfaces
br-int          0000.5e15d719a548       no              int-br-ex
                                                        qvo180f8458-7b
                                                        qvo3e60deda-cc
                                                        qvo92774056-da
br-tun          0000.febc48d02540       no
qbr180f8458-7b          8000.1a425eeda354       no              qvb180f8458-7b
                                                                vnet0
qbr3e60deda-cc          8000.8a70b498c8ce       no              qvb3e60deda-cc
                                                                vnet2
qbr92774056-da          8000.3e21bdf7dd5b       no              qvb92774056-da
                                                                vnet1
```


© 2014 Cisc

Example – OVS Bridge/Neutron Router




```
root@compute-server01:~# ovs-vsctl show
ac44a899-5f10-4ff9-8dad-902fa7c10e5e
...
Bridge br-tun
        Port "gre-2"
            Interface "gre-2"
                type: gre
                options: {in_key=flow, out_key=flow, remote_ip="10.121.13.50"}
        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="10.121.13.52"}
        Port br-tun
            Interface br-tun
                type: internal
```

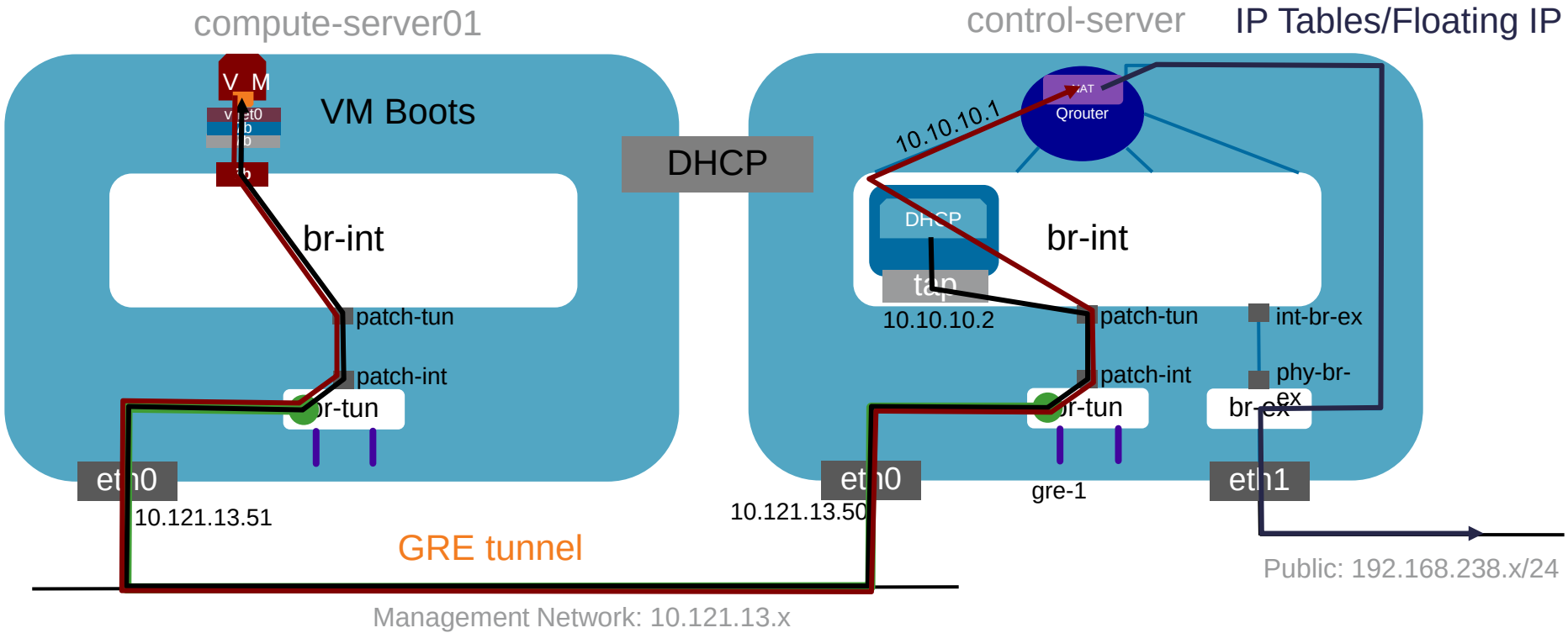
control-server

compute-server02

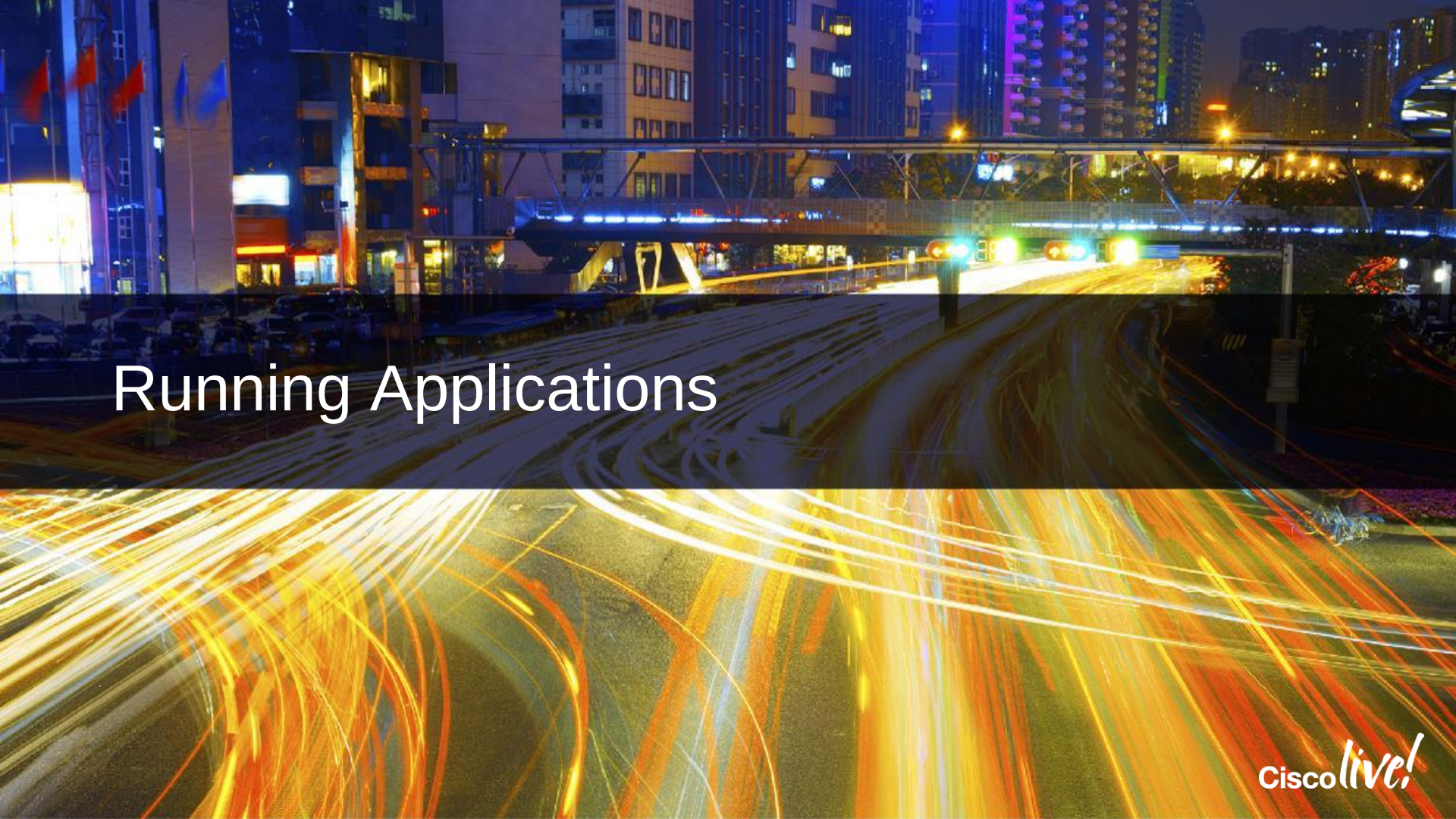
© 2014 Cisc

Example – Basic VM Traffic Flow

High-Level Walk-Thru



© 2014 Cisco and/or its affiliates. All rights reserved.



Running Applications

Cisco live!

Multiple Paths to Managing Images/Apps

- Docker:
  - http://www.docker.io/
  - https://wiki.openstack.org/wiki/Docker
- VMBuilder:
  - http://docwiki.cisco.com/wiki/OpenStack:VM_Build
  - https://launchpad.net/vmbuilder
  - https://help.ubuntu.com/12.04/serverguide/jeos-and-vmbuilder.html
- Disk Image Builder:
  - https://github.com/stackforge/diskimage-builder
- Heat – Template based orchestration engine :
  - https://wiki.openstack.org/wiki/Heat
  - https://github.com/openstack/heat
- Salt Cloud
  - https://github.com/saltstack/salt-cloud
- Baseline images + automated application deployment (scripts, Puppet, Chef)
- Template images – Prebuilt with apps installed and deployed from Glance

© 2014 Cisco and/or its affiliates. All rights reserved.

VMBuilder

Reference

- https://help.ubuntu.com/12.04/serverguide/jeos-and-vmbuilder.html
- Build images from KVM installed machine
- Create a configuration file and run vmbuilder:

```
# vmbuilder kvm ubuntu --hostname=base2 \
> --destdir=/var/lib/libvirt/images/base2
```

- Or via full CLI with options:

```
vmbuilder kvm ubuntu --suite precise --flavour virtual \
--arch amd64 -o --libvirt qemu:///system --ip 10.121.13.77 \
--hostname base2 --part vmbuilder.partition \
--user localadmin --name localadmin --pass ubuntu \
-m 2048 --cpus 1 --addpkg unattended-upgrades \
--addpkg openssh-server --addpkg puppet --addpkg git
```

```
root@builder:~# more /etc/vmbuilder.cfg
[DEFAULT]
tmpfs = suid,dev,size=2G
arch   = amd64
domain = example.com
ip     = 10.121.13.77
mask   = 255.255.255.0
net    = 10.121.13.0
bcast  = 10.121.13.255
gw     = 10.121.13.1
dns    = 10.121.12.10
user   = localadmin
name   = localadmin
pass   = ubuntu
firstboot = /etc/vmbuilder/firstscripts/firstboot.sh
[kvm]
libvirt = qemu:///system
bridge = virbr0
virtio_net = true
mem     = 2048
cpus    = 2
[ubuntu]
proxy = http://10.129.16.14:8080
suite = precise
flavour = virtual
#install-mirror = http://10.121.13.17:3142/
components = main,universe
addpkg = openssh-server, unattended-upgrades, git, vim, puppet
```

Puppet on Baseline Instances

- Puppet is installed via baseline image or manually installed
- Puppet master or local puppet (masterless) is built and manifests defined
  - Use same PM as the OpenStack build used or your production PM(s) for apps
- Puppet agent runs (or local puppet apply) and apps for that instance are installed and configured
  - Alternatively, install via puppet modules: http://forge.puppetlabs.com/
- Test apps

© 2014 Cisco and/or its affiliates. All rights reserved.

Reference

Puppet Agent Run – Example w/LAMP

- On Puppet Master - /etc/puppet/manifests/site.pp

```
# Nodes for web server instances
node 'sales-web-01' {
    include lamp
}
```

- LAMP layout on PM:

```
root@build-server:~# tree /etc/puppet/modules/lamp/
/etc/puppet/modules/lamp/
├── files
│   ├── apache2.conf
│   ├── index.php
│   └── php5.conf
└── manifests
    └── init.pp
```

- Puppet Agent run on instance:

```
Info: Applying configuration version '1363712915'
Debug: /Stage[main]/Lamp/Exec[mysqlpasswd]/require: requires Package[mysql-server]
Debug: /Stage[main]/Lamp/Exec[mysqlpasswd]/require: requires Package[apache2]
Debug: /Stage[main]/Lamp/Exec[mysqlpasswd]/notify: subscribes to Service[mysql]
Debug: /Stage[main]/Lamp/Exec[mysqlpasswd]/notify: subscribes to Service[apache2]
Debug: /Stage[main]/Lamp/Service[apache2]/require: requires Package[apache2]
Debug: /Stage[main]/Lamp/Exec[userdir]/require: requires Package[apache2]
```


© 2014 Cisco and/or its affiliates. All rights reserved.

Cisco live!

Monitoring

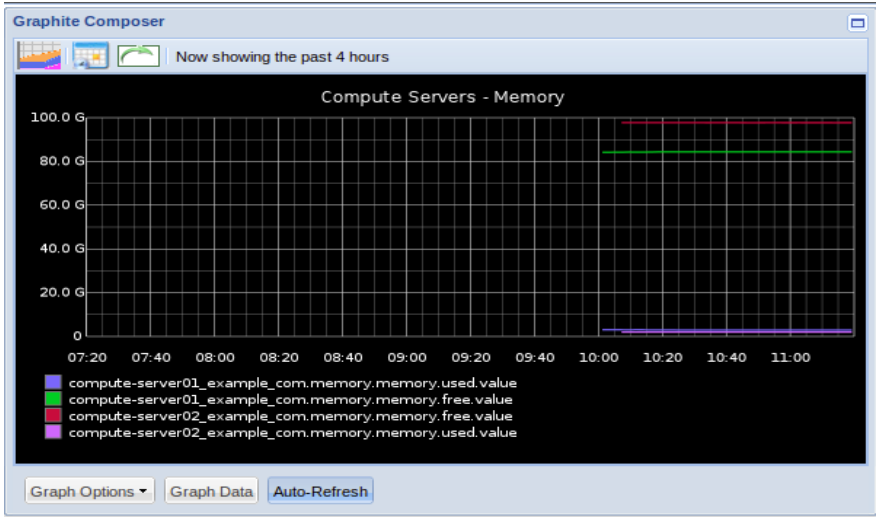
Cisco live!

Basic Monitoring is Available – Nagios/Graphite/Collectd

- http://<build-server>/nagios3 - Health monitoring of OpenStack nodes

Host	Status	Last Check	Duration	Status Information
compute-server01	UP	2013-03-19 12:03:47	4d 1h 29m 7s	PING OK - Packet loss = 0%, RTA = 0.27 ms
compute-server02	UP	2013-03-19 12:04:57	4d 1h 28m 57s	PING OK - Packet loss = 0%, RTA = 0.31 ms
control-server	UP	2013-03-19 12:05:47	0d 1h 9m 8s	PING OK - Packet loss = 0%, RTA = 0.30 ms

- http://<build-server>:8190 – Main Graphite performance console
- http://<build-server>:8190/dashboard/ - User/Self-service performance console
- http://www.nagios.org/
- http://graphite.wikidot.com/
- http://collectd.org/



© 2014 Cisco and/or its affiliates. All rights reserved.