



# Cisco UCS Integrated Infrastructure with Red Hat Enterprise Linux OpenStack Platform and Red Hat Ceph Storage

## Deployment Guide

Last Updated: May 4 2016



## About Cisco Validated Designs

---

The CVD program consists of systems and solutions designed, tested, and documented to facilitate faster, more reliable, and more predictable customer deployments. For more information visit

<http://www.cisco.com/go/designzone>.

ALL DESIGNS, SPECIFICATIONS, STATEMENTS, INFORMATION, AND RECOMMENDATIONS (COLLECTIVELY, "DESIGNS") IN THIS MANUAL ARE PRESENTED "AS IS," WITH ALL FAULTS. CISCO AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THE DESIGNS, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE DESIGNS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS ARE SOLELY RESPONSIBLE FOR THEIR APPLICATION OF THE DESIGNS. THE DESIGNS DO NOT CONSTITUTE THE TECHNICAL OR OTHER PROFESSIONAL ADVICE OF CISCO, ITS SUPPLIERS OR PARTNERS. USERS SHOULD CONSULT THEIR OWN TECHNICAL ADVISORS BEFORE IMPLEMENTING THE DESIGNS. RESULTS MAY VARY DEPENDING ON FACTORS NOT TESTED BY CISCO.

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0809R)

© 2016 Cisco Systems, Inc. All rights reserved.

# Table of Contents

About Cisco Validated Designs .....	2
Executive Summary .....	10
Solution Overview.....	11
Introduction .....	11
Audience .....	11
Purpose of the Document .....	11
Solution Summary.....	12
Deployment Architecture .....	13
Solution Design.....	14
Physical Topology.....	14
Solution Overview.....	16
System Hardware and Software Specifications.....	18
Bill of Materials .....	19
Solution Components .....	21
Cisco Unified Computing System.....	21
Server Pools .....	21
Cisco UCS Blades Distribution in Chassis .....	21
Service Profiles.....	22
Cisco UCS vNIC Configuration.....	23
Red Hat Linux OpenStack Platform 7 Director .....	25
Reference Architecture Workflow .....	26
Control .....	27
Compute .....	27
Ceph-Storage .....	27
Network Isolation.....	27
Provisioning.....	27
External .....	28
Internal API.....	28
Tenant .....	28
Storage .....	28
Storage Management .....	28
Network Types by Server Role.....	28
Tenant Network Types.....	29

VLAN segmentation mode .....	29
VXLAN segmentation mode .....	29
GRE segmentation mode .....	29
Cluster Manager and Proxy Server .....	29
High Availability .....	30
Cluster Management and Proxy Server .....	30
Cisco ML2 Plugins .....	31
Instance creation work flow .....	32
Deployment Hardware .....	35
Cabling Details.....	35
Physical Cabling .....	37
Cabling Logic .....	39
Cisco UCS Configuration .....	40
Configure Cisco UCS Fabric Interconnects .....	40
Configure the Cisco UCS Global Policies .....	42
Configure Server Ports for Blade Discovery and Rack Discovery.....	43
Configure Network Uplinks .....	46
Create KVM IP Pools.....	46
Create MAC Pools .....	47
Create UUID Pools.....	50
Create Server Pools for Controller, Compute and Ceph Storage Nodes.....	52
Create VLANs.....	54
Create a Network Control Policy.....	59
Create vNIC Templates.....	60
Create Boot Policy.....	67
Create a Maintenance Policy .....	70
Create an IPMI Access Policy .....	71
Create a Power Policy .....	73
Create a QOS system class .....	75
Create Storage Profiles for the Controller and Compute Blades.....	75
Create Storage Profiles for Cisco UCS C240 M4 Server Blades.....	82
Create Service Profile Templates for Controller Nodes .....	85
Create Service Profile Templates for Compute Nodes .....	101
Create Service Profile Templates for Ceph Storage Nodes.....	114
Create Service Profile for Undercloud ( OSP7 Director ) Node.....	120

Create Service Profiles for Controller Nodes.....	134
Create Service Profiles for Compute Nodes.....	135
Create Service Profiles for Ceph Storage Nodes .....	137
Create LUNs for the Ceph OSD and Journal Disks .....	138
Create Port Channels for Cisco UCS Fabrics .....	145
Cisco Nexus Configuration.....	148
Configure the Cisco Nexus 9372 PX Switch A.....	148
Configure the Cisco Nexus 9372 PX Switch B .....	149
Enable Features on the Switch.....	149
Enable Jumbo MTU .....	150
Create VLANs.....	150
Configure the Interface VLAN (SVI) on the Cisco Nexus 9K Switch A.....	151
Configure the Interface VLAN (SVI) on the Cisco Nexus 9K Switch B.....	152
Configure the VPC and Port Channels on Switch A.....	154
Configure the VPC and Port Channels on the Cisco Nexus 9K Switch B.....	155
Verify the Port Channel Status on the Cisco Nexus Switches.....	156
Cisco UCS Validation Checks .....	159
Install the Operating System on the Undercloud Node .....	161
Undercloud Setup .....	172
Undercloud Installation .....	172
Post Undercloud Installation Checks .....	175
Introspection.....	177
Pre-Installation Checks for Introspection .....	177
Run Introspection.....	187
Create Flavors .....	189
Set Flavors.....	189
Overcloud Setup.....	192
Customize Heat Templates .....	192
Single NIC VLAN Templates .....	193
Bond with VLAN Templates .....	194
Cisco UCS Configuration .....	194
Yaml Configuration Files Overview.....	196
Pre-Installation Checks Prior to Deploying Overcloud .....	201
Deploying Overcloud .....	203
Debugging Overcloud Failures .....	204

Overcloud Post Deployment Process .....	204
Overcloud Post-Deployment Configuration.....	209
Health Checks .....	213
Functional Validation.....	216
Performance and Scale Testing .....	217
Rally Testing and Measuring Compute Scaling.....	217
Prerequisites and Install.....	217
Test Methodology.....	217
Instance Sizing .....	219
Rally Scenario Task Configuration.....	219
Rally Tests with 1000 Virtual Machines .....	220
Rally Tests with 2000 Virtual Machines .....	224
Cisco Plugins.....	225
Conclusion from Rally Tests.....	226
Ceph Benchmark Tool for Ceph Scalability .....	226
Ceph Configuration.....	227
Create Virtual Machines for Ceph Testing .....	228
Cisco UCS C240 M4 LFF Results.....	228
Cisco UCS C240 M4 SFF Results.....	230
Analysis.....	231
Live Migration .....	232
Live Migration Introduction and Scope.....	232
Configuring Possibilities.....	232
Tunneling Memory Transfer .....	232
Auto Convergence.....	232
Test Methodology.....	233
Results .....	235
Recommendations .....	235
Upscaling the POD.....	236
Scale Up Storage Nodes.....	236
Provision the New Server in Cisco UCS .....	236
Run Introspection.....	240
Run Overcloud Deployment .....	242
Post Deployment Health Checks.....	243
Scale Up Compute Nodes.....	245

Provision the New Blade in UCS .....	245
Run Introspection.....	246
Run Overcloud Deploy .....	247
Post Deployment and Health Checks .....	248
High Availability .....	249
High Availability of Software Stack.....	250
OpenStack Services .....	250
High Availability of Hardware Stack .....	253
HA of Fabric Interconnects .....	253
Hardware Failures of IO Modules .....	255
HA on Nexus Switches .....	258
Creating Virtual Machines .....	261
HA on Controller Blades .....	266
HA on Compute Blades .....	271
HA on Storage Nodes .....	276
HA on Undercloud Node .....	286
Hardware Failures of Blades .....	287
Types of Failures .....	287
OpenStack Dependency on Hardware .....	287
IPMI Address .....	287
<b>NIC's and MAC addresses</b> .....	288
Local Disk .....	288
Cisco UCS Failure Scenarios .....	288
Hard Disk Failure .....	288
Blade Replacement.....	288
Case Study .....	289
Insert the New Blade into the Chassis.....	292
Fault Injection .....	292
Health Checks .....	294
Remove Failed Blade from Inventory.....	296
Change IPMI Address .....	296
Insert Old Disks .....	298
Associate Service Profile .....	298
Reboot the Server.....	300
Post Replacement Steps.....	300

Health Checks Post Replacement .....	301
Frequently Asked Questions .....	303
Cisco Unified Computing System.....	303
OpenStack.....	303
Troubleshooting.....	305
Cisco Unified Computing System.....	305
Undercloud Install.....	305
Introspection.....	306
Cleaning Up Failed Introspection .....	307
Updating Incorrect MAC or IPMI Addresses.....	308
Running Introspection on Failed Nodes.....	309
Overcloud Install.....	309
Debug Network Issues.....	310
Debug Ceph Storage Issues .....	310
Debug Heat Stack Issues.....	310
Overcloud Post-Deployment Issues.....	312
N1KV Plugin Checks .....	313
Nexus Plugin Checks .....	314
Cisco UCS Manager Plugin Checks .....	315
Run Time Issues .....	316
Best Practices.....	318
List of Bugs.....	319
Reference Documents .....	321
Conclusion.....	322
Appendix A.....	323
Undercloud instackenv.json .....	323
Overcloud Templates.....	326
network-environment.yaml.....	326
storage-environment.yaml.....	327
controller.yaml.....	327
compute.yaml.....	330
ceph-storage.yaml .....	332
ceph.yaml (C240M4L) .....	334
ceph.yaml (C240M4S).....	335
cisco-plugins.yaml .....	336

wipe_disk.yaml (C240M4L) .....	338
wipe_disk.yaml (C240M4S) .....	339
post_config.yaml .....	340
nameserver_ntp.yaml .....	340
run.sh .....	341
create_network_router.sh .....	341
create_vm.sh .....	341
boot-from-volume.json .....	343
Appendix B .....	345
network-environment.yaml .....	345
controller.yaml .....	345
run.sh .....	348
About the Authors .....	350
Acknowledgements .....	351



## Executive Summary

---

Cisco Validated Design program consist of systems and solutions that are designed, tested, and documented to facilitate and improve customer deployments. These designs incorporate a wide range of technologies and products into a portfolio of solutions that have been developed to address the business needs of our customers.

The reference architecture described in this document is a realistic use case for deploying Red Hat Enterprise Linux OpenStack Platform 7 on Cisco UCS blade and rack servers. The document covers step by step instructions for setting UCS hardware, installing Red Hat Linux OpenStack Director, issues and workarounds evolved during installation, integration of Cisco Plugins with OpenStack, what needs to be done to leverage High Availability from both hardware and software, use case of Live Migration, performance and scalability tests done on the configuration, lessons learnt, best practices evolved while validating the solution and a few troubleshooting steps, etc.

Cisco UCS Integrated Infrastructure for Red Hat Enterprise Linux OpenStack Platform is all in one solution for deploying OpenStack based private Cloud using Cisco Infrastructure and Red Hat Enterprise Linux OpenStack platform. The solution is validated and supported by Cisco and Red Hat, to increase the speed of infrastructure deployment and reduce the risk of scaling from proof-of-concept to full enterprise production.

# Solution Overview

---

## Introduction

Automation, virtualization, cost, and ease of deployment are the key criteria to meet the growing IT challenges. Virtualization is a key and critical strategic deployment model for reducing the Total Cost of Ownership (TCO) and achieving better utilization of the platform components like hardware, software, network and storage. The platform should be flexible, reliable and cost effective for enterprise applications.

Cisco UCS solution implementing Red Hat Enterprise Linux OpenStack Platform provides a very simplistic yet fully integrated and validated infrastructure to deploy VMs in various sizes to suit your application needs. Cisco Unified Computing System (UCS) is a next-generation data center platform that unifies computing, network, storage access, and virtualization into a single interconnected system, which makes Cisco UCS an ideal platform for OpenStack architecture. The combined architecture of Cisco UCS platform, Red Hat Enterprise Linux OpenStack Platform and Red Hat Ceph Storage can accelerate your IT transformation by enabling faster deployments, greater flexibility of choice, efficiency, and lower risk. Furthermore, Cisco Nexus series of switches provide the network foundation for the next-generation data center.

This deployment guide provides the audience a step by step instruction of Installing Red Hat Linux OpenStack Director and Red Hat Ceph Storage on Cisco UCS blades and rack servers. The traditional complexities of installing OpenStack are simplified by Red Hat Linux OpenStack Director while Cisco UCS Manager Capabilities bring an integrated, scalable, multi-chassis platform in which all resources participate in a unified management domain. The solution included in this deployment guide is a partnership from Cisco Systems, Inc., Red Hat, Inc., and Intel Corporation.

## Audience

The audience for this document includes, but is not limited to, sales engineers, field consultants, professional services, IT managers, partner engineers, IT architects, and customers who want to take advantage of an infrastructure that is built to deliver IT efficiency and enable IT innovation. The reader of this document is expected to have the necessary training and background to install and configure Red Hat Enterprise Linux, Cisco Unified Computing System (UCS) and Cisco Nexus Switches as well as a high level understanding of OpenStack components. External references are provided where applicable and it is recommended that the reader be familiar with these documents.

Readers are also expected to be familiar with the infrastructure, network and security policies of the customer installation.

## Purpose of the Document

This document describes the step by step installation of Red Hat Enterprise Linux OpenStack Platform 7 and Red Hat Ceph Storage 1.3 architecture on Cisco UCS platform. It also discusses about the day to day operational challenges of running OpenStack and steps to mitigate them, High Availability use cases, Live Migration, common troubleshooting aspects of OpenStack along with Operational best practices.

## Solution Summary

This solution is focused on Red Hat Enterprise Linux OpenStack Platform 7 (based on the upstream OpenStack Kilo release) and Red Hat Ceph Storage 1.3 on Cisco Unified Computing System. The advantages of Cisco UCS and Red Hat Enterprise Linux OpenStack Platform combine to deliver an OpenStack Infrastructure as a Service (IaaS) deployment that is quick and easy to setup. The solution can scale up for greater performance and capacity or scale out for environments that require consistent, multiple deployments. It provides:

Converged infrastructure of Compute, Networking, and Storage components from Cisco UCS is a validated enterprise-class IT platform, rapid deployment for business critical applications, reduces costs, minimizes risks, and increase flexibility and business agility Scales up for future growth.

Red Hat Enterprise Linux OpenStack Platform 7 on Cisco UCS helps IT organizations accelerate cloud deployments while retaining control and choice over their environments with open and inter-operable cloud solutions. It also offers redundant architecture on compute, network, and storage perspective. The solution comprises of the following key components:

- Cisco Unified Computing System (UCS)
  - Cisco UCS 6200 Series Fabric Interconnects
  - Cisco VIC 1340
  - Cisco VIC 1227
  - Cisco 2204XP IO Module or Cisco UCS Fabric Extenders
  - Cisco B200 M4 Servers
  - Cisco C240 M4 Servers
- Cisco Nexus 9300 Series Switches
- Cisco Nexus 1000v for KVM
- Cisco Nexus Plugin for Nexus Switches
- Cisco UCS Manager Plugin for Cisco UCS
- Red Hat Enterprise Linux 7.x
- Red Hat Enterprise Linux OpenStack Platform Director
- Red Hat Enterprise Linux OpenStack Platform 7
- Red Hat Ceph Storage 1.3

The scope is limited to the infrastructure pieces of the solution. It does not address the vast area of the OpenStack components and multiple configuration choices available in OpenStack.

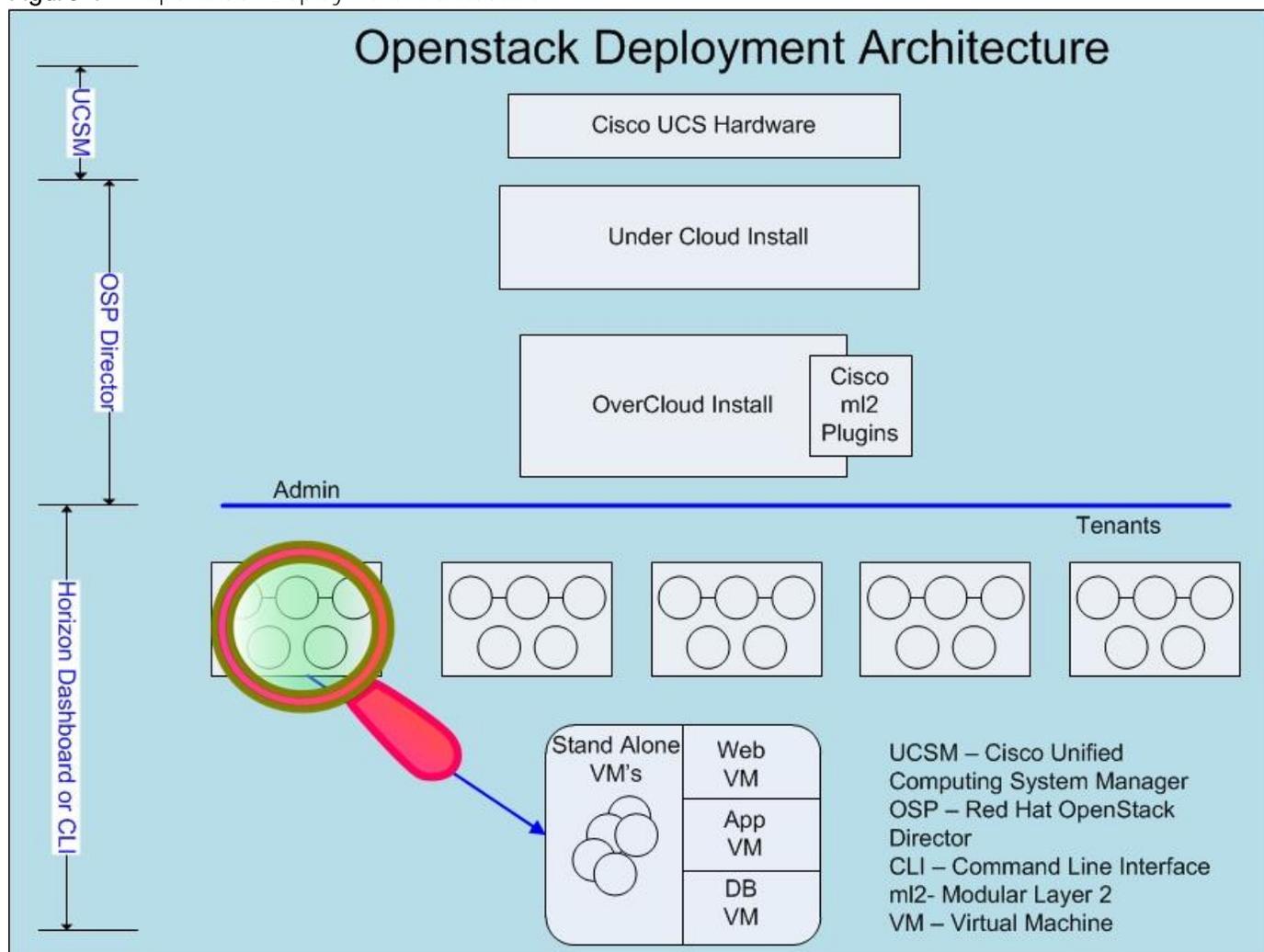
# Deployment Architecture

This architecture is based on Red Hat Enterprise Linux OpenStack platform build on Cisco UCS hardware is an integrated foundation to create, deploy, and scale OpenStack cloud based on Kilo OpenStack community release. Kilo version introduces Red Hat Linux OpenStack Director (RHEL-OSP), a new deployment tool chain that combines the functionality from the upstream TripleO and Ironic projects with components from previous installers.

The reference architecture use case provides a comprehensive, end-to-end example of deploying RHEL-OSP7 cloud on bare metal using OpenStack Director and services through heat templates.

The first section in this Cisco Validated Design covers setting up of Cisco hardware the blade and rack servers, chassis and Fabric Interconnects and the peripherals like Nexus 9000 switches. The second section explains the step by step install instructions for installing cloud through RHEL OSP Director. The final section includes the functional and High Availability tests on the configuration, Performance, Live migration tests, and the best practices evolved while validating the solution.

Figure 1 OpenStack Deployment Architecture



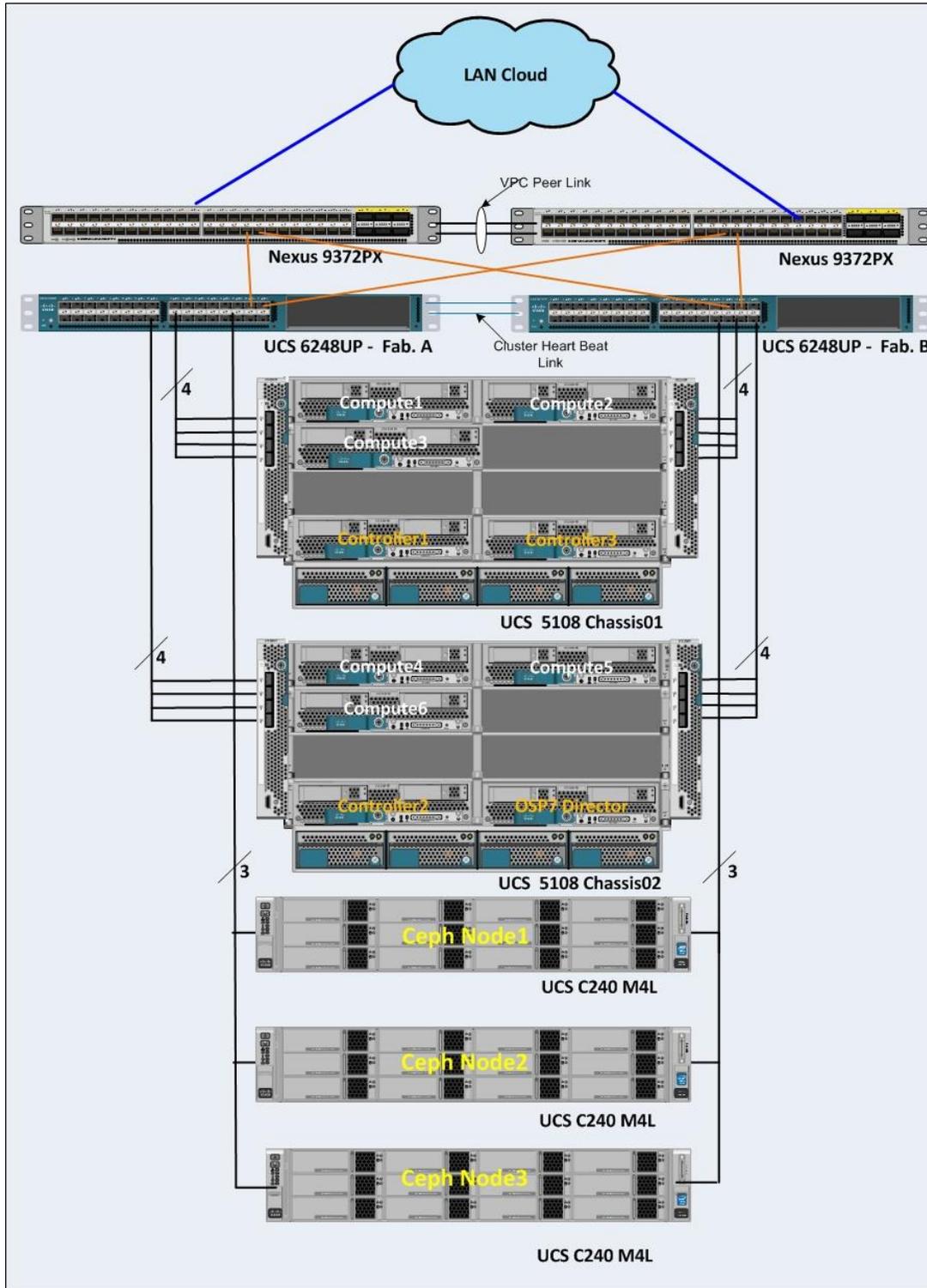
## Solution Design

---

### Physical Topology

Figure 2 illustrates the physical topology of this solution.

**Figure 2** Physical Topology



The configuration comprised of 3 controller nodes, 6 compute nodes, 3 storage nodes, a pair of UCS Fabrics and Nexus switches, where most of the tests were conducted. In another configuration the system had 20 Compute nodes, 12 Ceph nodes and 3 controllers distributed across 3 UCS chassis where few install and scalability tests were performed. Needless to say that architecture is scalable horizontally and vertically within the chassis.

- More Compute Nodes and Chassis can be added as desired.
- More Ceph Nodes for storage can be added. The Ceph nodes can be UCS C240M4L or C240M4S.
- If more bandwidth is needed, Cisco IO Modules can be 2208XP as opposed to 2204XP used in the configuration.
- Both Cisco Fabric Interconnects and Cisco Nexus Switches can be on 96 port switches instead of 48 ports as shown above.

## Solution Overview

This solution components and diagrams are implemented per the [Design Guide](#) and basic overview is provided below.

The Cisco Unified Computing System is an integrated, scalable, multi-chassis platform in which all resources participate in a unified management domain. The Cisco Unified Computing System accelerates the delivery of new services simply, reliably, and securely through end-to-end provisioning and migration support for both virtualized and non-virtualized systems. Cisco UCS manager using single connect technology manages servers and chassis and performs auto-discovery to detect inventory, manage, and provision system components that are added or changed.

The Red Hat Enterprise Linux OpenStack Platform IaaS cloud on Cisco UCS servers is implemented as a collection of interacting services that control compute, storage, and networking resources.

OpenStack Networking handles creation and management of a virtual networking infrastructure in the OpenStack cloud. Infrastructure elements include networks, subnets, and routers. Because OpenStack Networking is software-defined, it can react in real-time to changing network needs, such as creation and assignment of new IP addresses.

Compute serves as the core of the OpenStack cloud by providing virtual machines on demand. Compute supports the libvirt driver that uses KVM as the hypervisor. The hypervisor creates virtual machines and enables live migration from node to node.

OpenStack also provides storage services to meet the storage requirements for the above mentioned virtual machines.

The Keystone provides user authentication to all OpenStack systems.

The solution also includes OpenStack Networking ML2 Core components.

Cisco Nexus 1000V OpenStack solution is an enterprise-grade virtual networking solution, which brings Security, Policy control, and Visibility together with Layer2/Layer 3 switching at the hypervisor layer. When it comes to application visibility, Cisco Nexus 1000V provides insight into live and historical VM migrations and advanced automated troubleshooting capabilities to identify problems in seconds.

The Cisco Nexus driver for OpenStack Neutron allows customers to easily build their infrastructure-as-a-service (IaaS) networks using the industry's leading networking platform, delivering performance, scalability, and stability with the familiar manageability and control you expect from Cisco® technology.

**Cisco UCS Manager Plugin configures compute blades with necessary VLAN's.** The Cisco UCS Manager Plugin talks to the Cisco UCS Manager application running on Fabric Interconnect.

## System Hardware and Software Specifications

Table 1 lists the Hardware and Software releases used for solution verification.

**Table 1 Required Hardware Components**

	Hardware	Quantity	Firmware Details
Director	Cisco UCS B200M4 blade	1	2.2(5)
Controller	Cisco UCS B200M4 blade	3	2.2(5)
Compute	Cisco UCS B200M4 blade	6	2.2(5)
Storage	Cisco UCS C240M4L rack server	3	2.2(5)
Fabrics Interconnects	Cisco UCS 6248UP FIs	2	2.2(5)
Nexus Switches	Cisco Nexus 9372 NX-OS	2	7.0(3)I1(3)

**Table 2 Software Specifications**

	Software	Version
Operating System	Red Hat Enterprise Linux	7.2
OpenStack Platform	Red Hat Enterprise Linux OpenStack Platform	RHEL-OSP 7.2
	Red Hat Enterprise Linux OpenStack Director	RHEL-OSP 7.2
	Red Hat Ceph Storage	1.3
Cisco N1000V	VSM and VEM modules	5.2(1)SK3(2.2x)
Plugins	Cisco Nexus Plugin	RHEL-OSP 7.2
	Cisco UCSM Plugin	RHEL-OSP 7.2
	Cisco N1KV Plugin	RHEL-OSP 7.2

## Bill of Materials

This section contains the Bill of Materials used in the configuration.

Component	Model	Quantity	Comments
OpenStack Platform Director Node	Cisco UCS B200M4 blade	1	CPU - 2 x E5-2630 V3 Memory - 8 x 16GB 2133 MHz DIMM - total of 128G Local Disks - 2 x 300 GB SAS disks for Boot Network Card - 1x1340 VIC Raid Controller - Cisco MRAID 12 G SAS Controller
Controller Nodes	Cisco UCS B200M4 blades	3	CPU - 2 x E5-2630 V3 Memory - 8 x 16GB 2133 MHz DIMM - total of 128G Local Disks - 2 x 300 GB SAS disks for Boot Network Card - 1x1340 VIC Raid Controller - Cisco MRAID 12 G SAS Controller
Compute Nodes	Cisco UCS B200M4 blades	6	CPU - 2 x E5-2660 - V3 Memory - 16 x 16GB 2133 MHz DIMM - total of 256G Local Disks - 2 x 300 GB SAS disks for Boot Network Card - 1x1340 VIC Raid Controller - Cisco MRAID 12 G SAS Controller
Storage Nodes	Cisco UCS C240M4L rack servers	3	CPU - 2 x E5-2630 - V3 Memory - 8 x 16GB 2133 MHz DIMM - total of 128G Internal HDD - None <b>Ceph OSD's - 8 x 6TB SAS Disks</b>

Component	Model	Quantity	Comments
			Ceph Journals - <b>2 x 400GB SSD's</b> OS Boot - 2 x 1TB SAS Disks Network Cards - 1 x VIC 1227 Raid Controller - Cisco MRAID 12 G SAS Controller
Chassis	Cisco UCS 5108 Chassis	2	
IO Modules	IOM 2204 XP	4	
Fabric Interconnects	Cisco UCS 6248UP Fabric Interconnects	2	
Switches	Cisco Nexus 9372PX Switches	2	



Deployment and a few performance tests have been evaluated on another configuration with similar hardware and software specifications as listed above but with 20 Compute nodes and 12 Ceph storage nodes.

# Solution Components

## Cisco Unified Computing System

### Server Pools

Server pools will be utilized to divide the OpenStack server roles for ease of deployment and scalability. These pools will also decide the placement of server roles within the infrastructure. The following pools were created.

- OpenStack Controller Server pool
- OpenStack Compute Server pool
- OpenStack Ceph Server pool



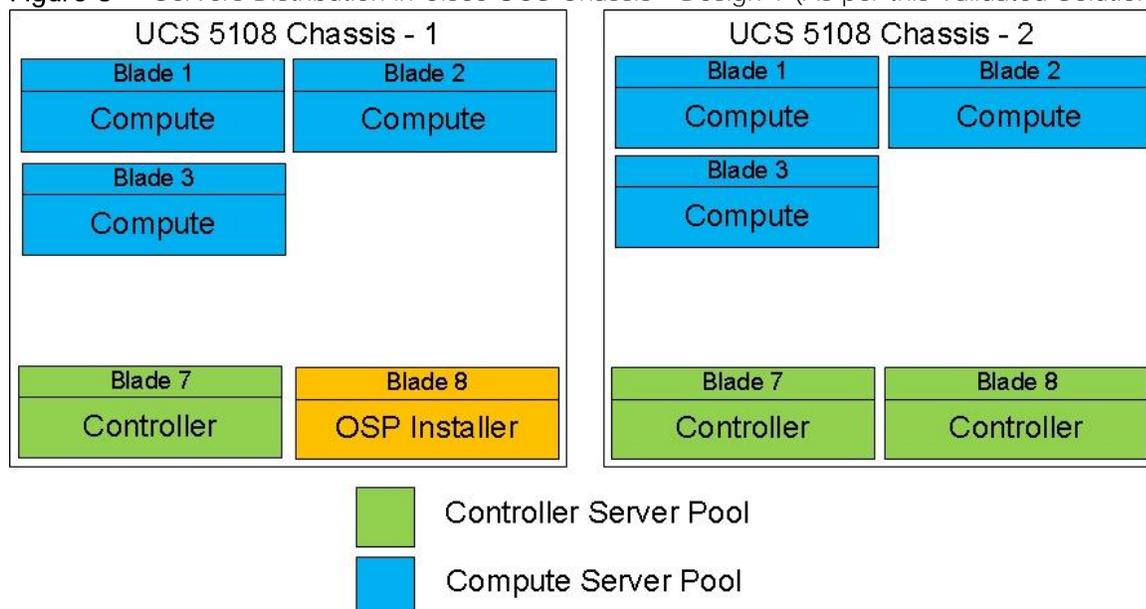
The Undercloud node will be a single server and is not associated with any pool. It is a standalone template and is used to create a service profile clone.

The compute server pool allows quick provisioning of additional hosts by adding the new servers to the compute server pool. The newly provisioned compute hosts can be added into an existing OpenStack environment through introspection and Overcloud deploy, covered later in this document.

### Cisco UCS Blades Distribution in Chassis

Figure 3 lists the server distribution in the Cisco UCS Chassis.

**Figure 3** Servers Distribution in Cisco UCS Chassis - Design 1 (As per this Validated Solution)



The controllers and computes are distributed across the chassis. This gives High Availability to the stack though a failure of Chassis per se does not happen. There is only one Installer node in the system and can

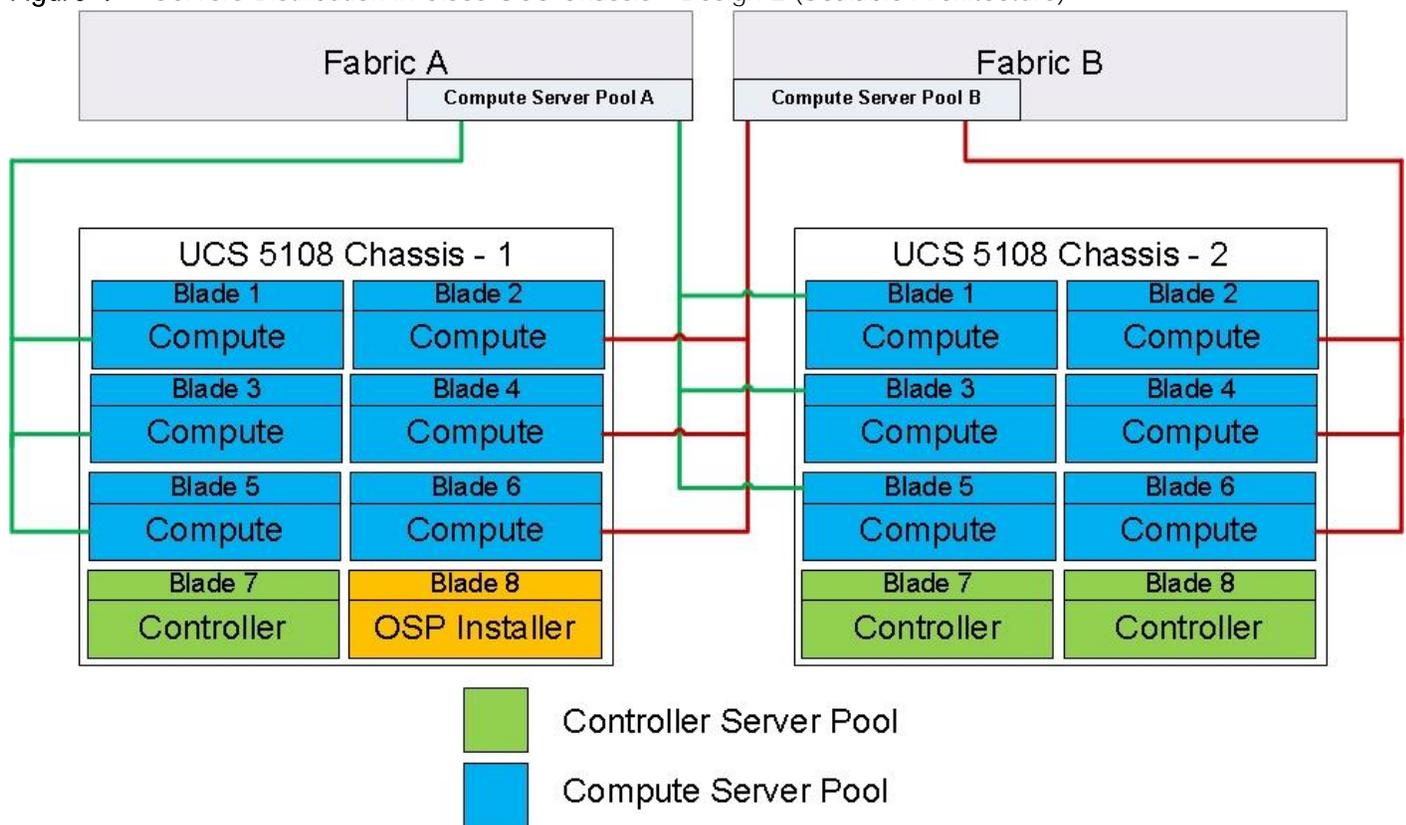
be added in any one of the Chassis as above. In case of larger deployments having 3 or more chassis, it is recommended to distribute one controller in each chassis.

In larger deployments where the chassis are fully loaded with blades a better approach while creating server pools could be distribute manually the tenant and storage traffic across the Fabrics.

Compute pools are created as listed below:

- OpenStack Compute Server pool A
- OpenStack Compute Server pool B
- The Compute Server pool A can be used for the blades on the left side of the chassis pinned to Fabric A, while the Compute Server pool B can be used for the blades on the right side of the chassis. This is achieved with pool A using vNICs pinned to Fabric A while pool B tenant vNICs pinned to Fabric B.

**Figure 4** Servers Distribution in Cisco UCS Chassis - Design 2 (Scalable Architecture)



The above method ensures that the tenant traffic is distributed evenly across both the fabrics.

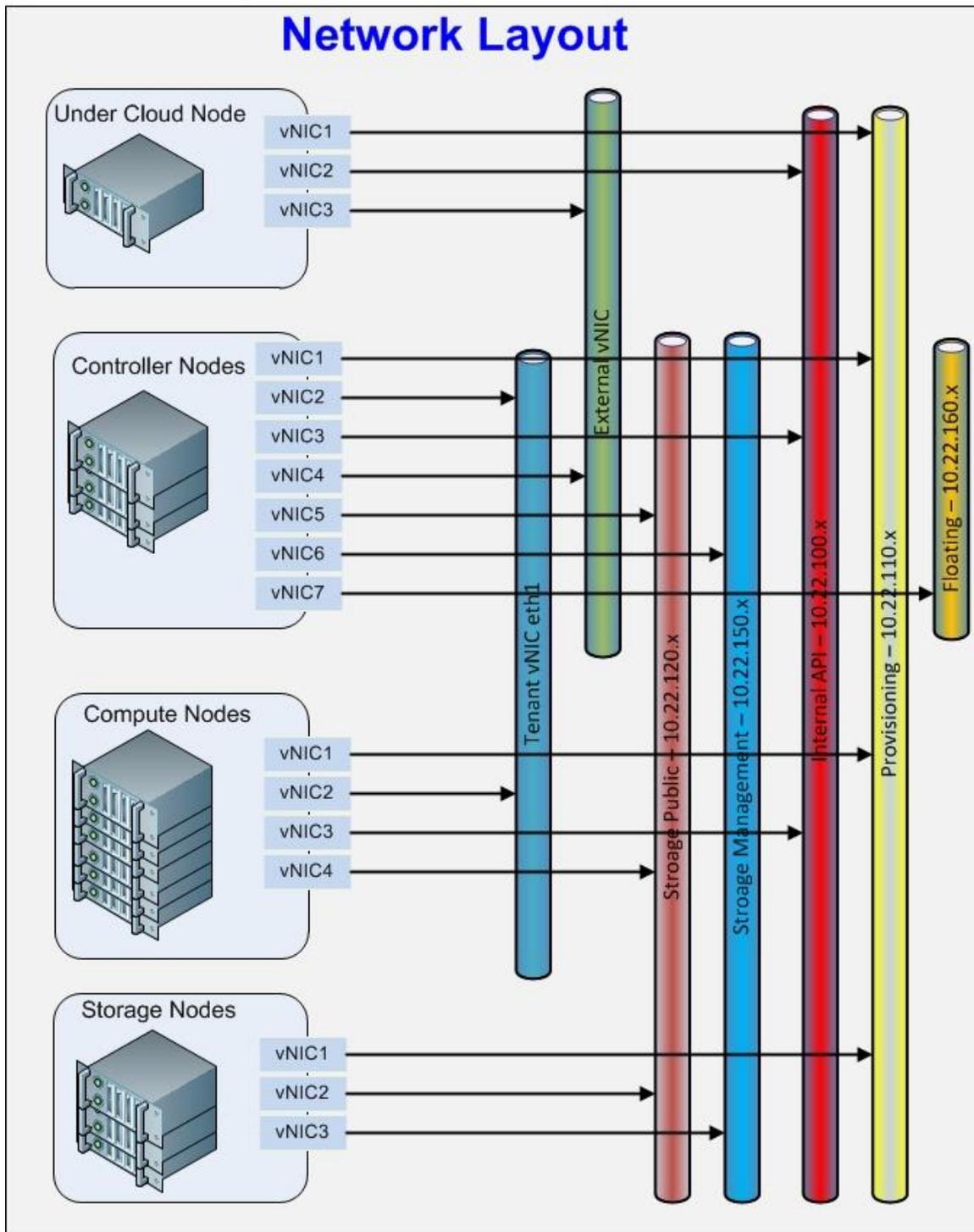
## Service Profiles

Service profiles will be created from the service templates. However once successfully created, they will be unbound from the templates. The vNIC to be used for tenant traffic needs to be identified as eth1. This is to take care of the current limitation in [Cisco UCSM kilo plugin](#) for OpenStack. This is being addressed while this document is being written and will be taken care in the future releases.

## Cisco UCS vNIC Configuration

Figure 5 illustrates the network layout.

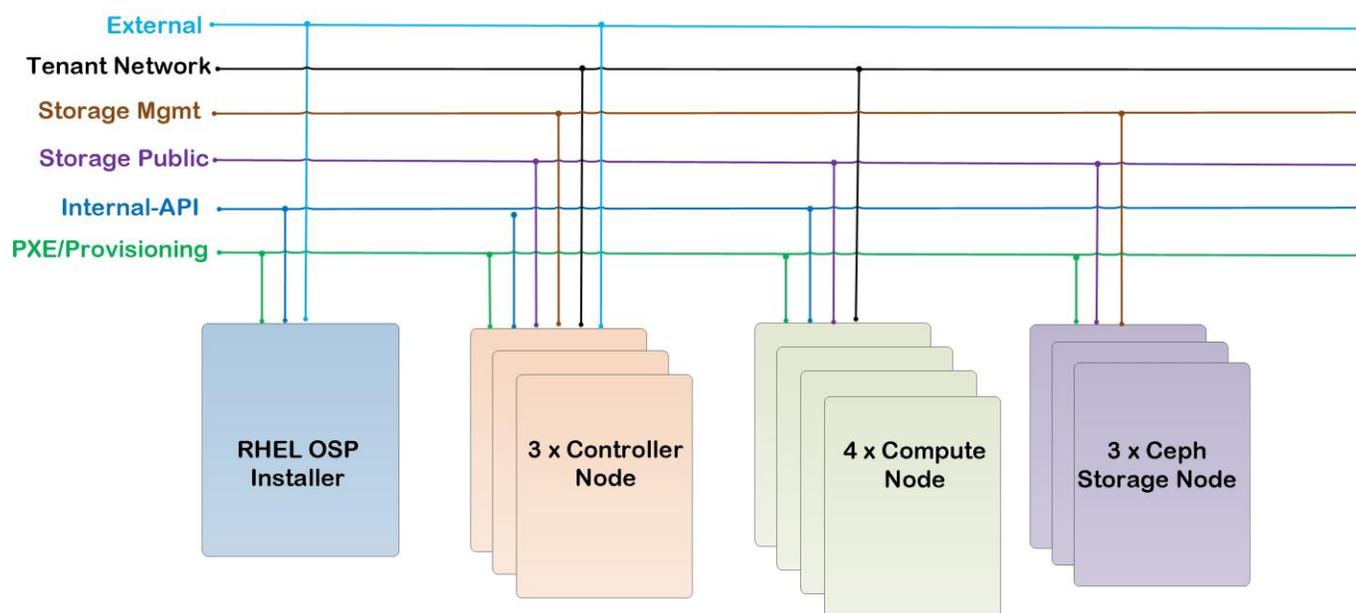
Figure 5 Network Layout





A Floating or Provider network is not necessary. It has been included in the configuration because of the limitation in the IP's. Virtual machines can be configured to have direct access through the external network too.

A separate network layout was also verified in another POD without any floating IP's. This is for customers who do not have the limitations of external IP's as encountered in the configuration. However most of the tests were performed with floating IP's only. The Network Topology in this design is almost similar to what shown above. The virtual machines can be accessed directly from the external work. The below diagram depicts how the network was configured in this POD without floating network. With this you need not have floating vNIC interface for Controller Service profile, nor you will need a section of block in controller.yaml for floating ip and passing the floating parameter in your overcloud deploy command. Refer [Appendix B](#) for details.



The family of vNICs are placed in the same Fabric Interconnect to avoid an extra hop to the upstream Nexus switches.

The following categories of vNICs are used in the setup:

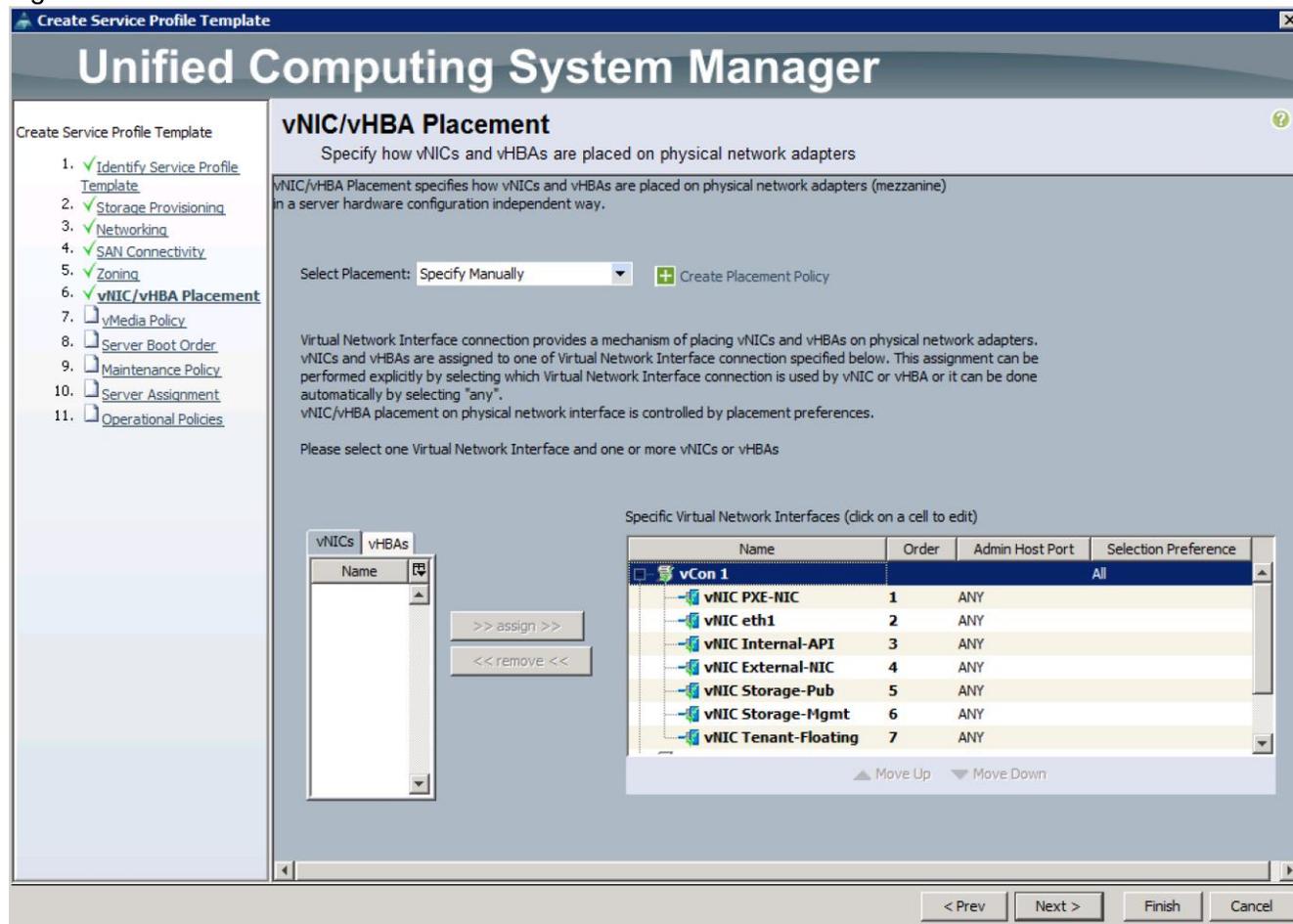
- Provisioning Interfaces pxe vNICs are pinned to Fabric A
- Tenant vNICs are pinned to Fabric A
- Internal API vNICs are pinned to Fabric B
- External Interfaces vNICs are pinned to Fabric A
- Storage Public Interfaces are pinned to Fabric A
- Storage Management Interfaces are pinned to Fabric B

Only one Compute server pool is created in the setup. However, we may create multiple pools if desired as mentioned above.



While configuring vNICs in templates and with failover option enabled in Fabrics, the vNICs order has to be specified manually as shown below.

Figure 6 vNIC Placement



The order of vNIC's has to be pinned as above for consistent PCI device naming options. The above is an example of controller blade. The same has to be done for all the other servers, the Compute and Storage nodes. This order should match the Overcloud heat templates NIC1, NIC2, NIC3, and NIC4.

## Red Hat Linux OpenStack Platform 7 Director

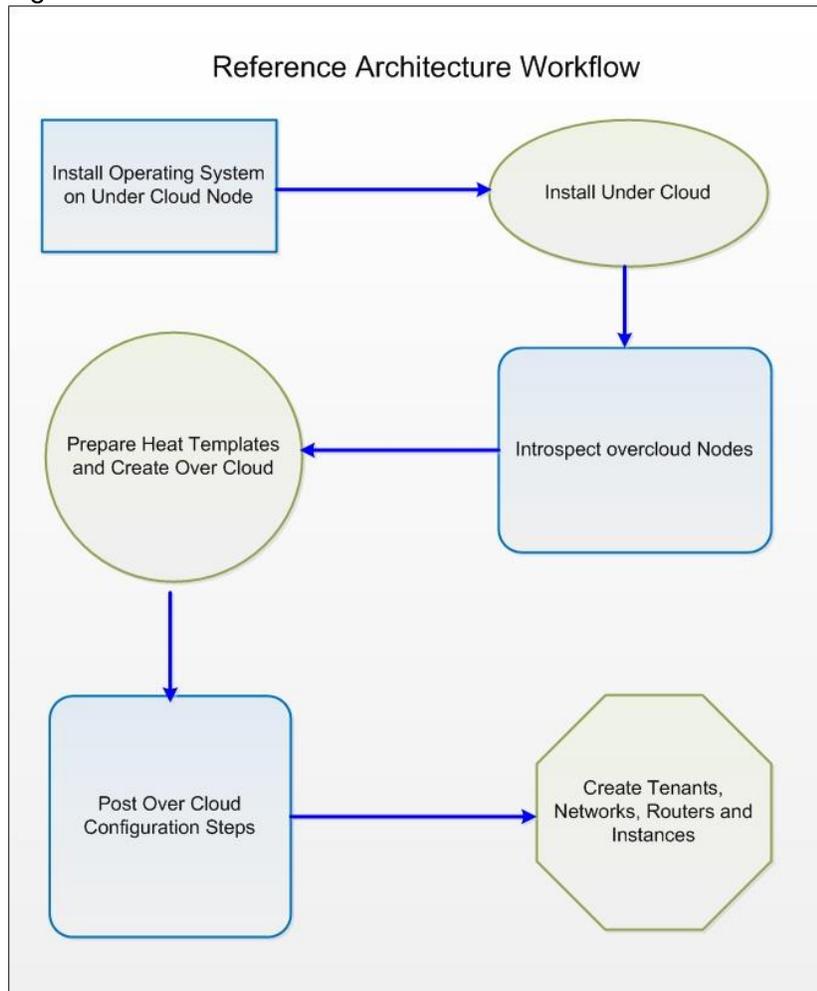
Red Hat Linux OpenStack Platform (RHEL-OSP7) delivers an integrated foundation to create, deploy, and scale a more secure and reliable public or private OpenStack cloud. RHEL-OSP7 starts with the proven foundation of Red Hat Enterprise Linux and integrates Red Hat's OpenStack Platform technology to provide production ready cloud platform. RHEL-OSP7 Director is based on community based Kilo OpenStack release. Red Hat RHEL-OSP7 introduces a cloud installation and lifecycle management tool chain. It provides

- Simplified deployment through ready-state provisioning of bare metal resources
- Flexible network definitions
- High Availability with Red Hat Enterprise Linux Server High Availability
- Integrated setup and Installation of Red Hat Ceph Storage 1.3

## Reference Architecture Workflow

Figure 7 illustrates the reference architecture workflow.

**Figure 7** Reference Architecture Workflow



Red Hat Linux OpenStack Platform Director is a new set of tool chain introduced with Kilo that automates the creation of Undercloud and Overcloud nodes as above. It performs the following:

- Install Operating System on Undercloud Node
- Install Undercloud Node
- Perform Hardware Introspection
- Prepare Heat templates and Install Overcloud
- Implement post Overcloud configuration steps
- Create Tenants, Networks and Instances for Cloud

Undercloud Node is the deployment environment while Overcloud nodes are referred to nodes actually rendering the cloud services to the tenants.

The Undercloud is the TripleO (OOO – OpenStack over OpenStack) control plane. It uses native OpenStack APIs and services to deploy, configure, and manage the production OpenStack deployment. The Undercloud defines the Overcloud with Heat templates and then deploys it through the Ironic bare metal provisioning service. OpenStack Director includes predefined Heat templates for the basic server roles that comprise the Overcloud. Customizable templates allow Director to deploy, redeploy, and scale complex Overclouds in a repeatable fashion.

Ironic gathers information about bare metal servers through a discovery mechanism known as introspection. Ironic pairs servers with bootable images and installs them through PXE and remote power management.

Red Hat Linux OpenStack Director deploys all servers with the same generic image by injecting Puppet modules into the image to tailor it for specific server roles. It then applies host-specific customizations through Puppet including network and storage configurations. While the Undercloud is primarily used to deploy OpenStack, the Overcloud is a functional cloud available to run virtual machines and workloads.

The following subsections detail the roles that comprise the Overcloud.

## Control

This role provides endpoints for REST-based API queries to the majority of the OpenStack services. These include Compute, Image, Identity, Block, Network, and Data processing. The controller nodes also provide **the supporting facilities for the API's, database, load balancing, messaging, and distributed memory** objects. They also provide external access to virtual machines. The controller can run as a standalone server or as a High Availability (HA) cluster. The current configuration was configured with HA.

## Compute

This role provides the processing, memory, storage, and networking resources to run virtual machine instances. It runs the KVM hypervisor by default. New instances are spawned across compute nodes in a round-robin fashion based on resource availability.

## Ceph-Storage

Ceph is a distributed block, object store and file system. This role deploys Object Storage Daemon (OSD) nodes for Ceph clusters. It also installs the Ceph Monitor service on the controller. The instance distribution is influenced by the currently set filters. The default filters can be altered if needed; for more information, please refer to the [OpenStack documentation](#).

## Network Isolation

OpenStack requires multiple network functions. While it is possible to collapse all network functions onto a single network interface, isolating communication streams in their own physical or virtual networks provides better performance and scalability. Each OpenStack service is bound to an IP on a particular network. In a cluster a service virtual IP is shared among all of the HA controllers.

## Provisioning

The Control plane installs Overcloud through this network. All nodes must have a physical interface attached to the provisioning network. This network carries DHCP/PXE and TFTP traffic. It must be provided on a dedicated interface or native VLAN to the boot interface. The provisioning interface can also act as a default

gateway for to Overcloud; the compute and storage nodes use this provisioning gateway interface on the Undercloud node.

## External

The External network is used for hosting the Horizon dashboard and the Public APIs, as well as hosting the floating IPs that are assigned to VMs. The Neutron L3 routers which perform NAT are attached to this interface. The range of IPs that are assigned to floating IPs should not include the IPs used for hosts and VIPs on this network.

## Internal API

This network is used for connections to the API servers, as well as RPC messages using RabbitMQ and connections to the database. The Glance Registry API uses this network, as does the Cinder API. This network is typically only reachable from inside the OpenStack Overcloud environment, so API calls from outside the cloud will use the Public APIs.

## Tenant

Virtual machines communicate over the tenant network. It supports three modes of operation: VXLAN, GRE, and VLAN. VXLAN and GRE tenant traffic is delivered through software tunnels on a single VLAN. Individual VLANs correspond to tenant networks in the case where VLAN tenant networks are used.

## Storage

This network carries storage communication including Ceph, Cinder, and Swift traffic. The virtual machine instances communicate with the storage servers through this network. Data-intensive OpenStack deployments should isolate storage traffic on a dedicated high bandwidth interface, i.e. 10 GB interface. The Glance API, Swift proxy, and Ceph Public interface services are all delivered through this network.

## Storage Management

Storage management communication can generate large amounts of network traffic. This network is shared between the front and back end storage nodes. Storage controllers use this network to access data storage nodes. This network is also used for storage clustering and replication traffic.

Network traffic types are assigned to network interfaces through Heat template customizations prior to deploying the Overcloud. Red Hat Enterprise Linux OpenStack Platform Director supports several network interface types including physical interfaces, bonded interfaces (not with Cisco UCS Fabric Interconnects), and either tagged or native 802.1Q VLANs.

## Network Types by Server Role

The previous section discussed server roles. Each server role requires access to specific types of network traffic. The network isolation feature allows Red Hat Enterprise Linux OpenStack Platform Director to segment network traffic by particular network types. When using network isolation, each server role must have access to its required network traffic types.

By default, Red Hat Enterprise Linux OpenStack Platform Director collapses all network traffic to the provisioning interface. This configuration is suitable for evaluation, proof of concept, and development

environments. It is not recommended for production environments where scaling and performance are primary concerns.

## Tenant Network Types

Red Hat Enterprise Linux OpenStack Platform 7 supports tenant network communication through the OpenStack Networking (Neutron) service. OpenStack Networking supports overlapping IP address ranges across tenants through the Linux **kernel's network namespace capability**. It also supports three default networking types:

### VLAN segmentation mode

Each tenant is assigned a network subnet mapped to an 802.1q VLAN on the physical network. This tenant networking type requires VLAN-assignment to the appropriate switch ports on the physical network.

### VXLAN segmentation mode

The VXLAN mechanism driver encapsulates each layer 2 Ethernet frame sent by the VMs in a layer 3 UDP packet. The UDP packet includes an 8-byte field, within which a 24-bit value is used for the VXLAN Segment ID. The VXLAN Segment ID is used to designate the individual VXLAN over network on which the communicating VMs are situated. This provides segmentation for each Tenant network

### GRE segmentation mode

The GRE mechanism driver encapsulates each layer 2 Ethernet frame sent by the VMs in a special IP packet using the GRE protocol (IP type 47). The GRE header contains a 32-bit key which is used to identify a flow or virtual network in a tunnel. This provides segmentation for each Tenant network.



Cisco Nexus Plugin is bundled in OpenStack Platform 7 kilo release. While it can support both VLAN and VXLAN configurations, only VLAN mode is validated as part of this design. VXLAN will be considered in future releases when the current VIC 1340 Cisco interface card will be certified on VXLAN and Red Hat operating system.

---

## Cluster Manager and Proxy Server

Two components drive HA for all core and non-core OpenStack services: the cluster manager and the proxy server.

The cluster manager is responsible for the startup and recovery of an inter-related services across a set of **physical machines**. It **tracks the cluster's internal state across multiple machines**. State changes trigger appropriate responses from the cluster manager to ensure service availability and data integrity.

This section describes the steps to configure networking for Overcloud. The network setup used in the configuration as shown in [Figure 5](#) earlier.

The configuration is done using Heat Templates on the Undercloud prior to deploying the Overcloud. These steps need to be followed after the Undercloud install. In order to use network isolation, we have to define the Overcloud networks. Each will have an IP subnet, a range of IP addresses to use on the subnet and a VLAN ID. These parameters will be defined in the network environment file. In addition to the global settings there is a template for each of the nodes like controller, compute and Ceph that determines the NIC configuration for each role. These have to be customized to match the actual hardware configuration.

Heat communicates with Neutron API running on the Undercloud node to create isolated networks and to assign neutron ports on these networks. Neutron will assign a static port to each port and Heat will use these **static IP's to configure networking on the** Overcloud nodes. A utility called os-net-config runs on each node at provisioning time to configure host level networking.

Table 3 lists the VLANs that are created on the configuration.

**Table 3 VLANs**

VLAN Name	VLAN Purpose	VLAN ID or VLAN Range Used in This Design for Reference
PXE	Provisioning Network VLAN	110
Internal-API	Internal API Network	100
External	External Network	215
Storage Public	Storage Public Network	120
Storage Management	Storage Cluster or Management Network	150
Floating	Floating Network	160

## High Availability

Red Hat Linux OpenStack Director's approach is to leverage Red Hat's distributed cluster system.

### Cluster Management and Proxy Server

The cluster manager is responsible for the startup and recovery of an inter-related services across a set of **physical machines. It tracks the cluster's internal state across multiple machines. State changes trigger** appropriate responses from the cluster manager to ensure service availability and data integrity.

In the HA model Clients do not directly connect to service endpoints. Connection requests are routed to service endpoints by a proxy server.

Cluster manager provides state awareness of other machines to coordinate service startup and recovery, shared quorum to determine majority set of surviving cluster nodes after failure, data integrity through fencing and automated recovery of failed instances.

Proxy servers help in load balancing connections across service end points. The nodes can be added or removed without interrupting service.

Red Hat Linux OpenStack Director uses HAproxy and Pacemaker to manage HA services and load balance connection requests. With the exception of RabbitMQ and Galera, HAproxy distributes connection requests to active nodes in a round-robin fashion. Galera and RabbitMQ use persistent options to ensure requests go only to active and/or synchronized nodes. Pacemaker checks service health at one second intervals. Timeout settings vary by service.

The combination of Pacemaker and HAProxy:

- Detects and recovers machine and application failures

- Starts and stops OpenStack services in the correct order
- Responds to cluster failures with appropriate actions including resource failover and machine restart and fencing

RabbitMQ, memcached, and mongodb do not use HAProxy server. These services have their own failover and HA mechanisms.

## Cisco ML2 Plugins

OpenStack Modular Layer 2 (ML2) allows separation of network segment types and the device specific implementation of segment types. ML2 **architecture consists of multiple ‘type drivers’ and ‘mechanism drivers’**. Type drivers manage the common aspects of a specific type of network while the mechanism driver manages specific device to implement network types.

Type drivers

- VLAN
- GRE
- VXLAN

Mechanism drivers

- Cisco UCSM
- Cisco Nexus
- Cisco Nexus 1000v
- Openvswitch, Linuxbridge

The Cisco Nexus driver for OpenStack Neutron allows customers to easily build their Infrastructure-as-a-Service (**iaaS**) **networks using the industry’s leading networking** platform, delivering performance, scalability, and stability with the familiar manageability and control you expect from Cisco® technology. ML2 Nexus drivers dynamically provision OpenStack **managed VLAN’s on Nexus switches. They configure the trunk ports with the dynamically created VLAN’s solving the logical port count issue on Nexus switches. They** provide better manageability of the network infrastructure.

ML2 UCSM drivers dynamically provision OpenStack **managed VLAN’s on Fabric Interconnects. They configure VLAN’s on Controller and Compute node VNIC’s.** The Cisco UCS Manager Plugin talks to the Cisco UCS Manager application running on Fabric Interconnect and is part of an ecosystem for Cisco UCS Servers that consists of Fabric Interconnects and IO modules. The ML2 Cisco UCS Manager driver does not support configuration of Cisco UCS Servers, whose service profiles are attached to Service Templates. This is to prevent that same VLAN configuration to be pushed to all the service profiles based on that template. The plugin can be used after the Service Profile has been unbound from the template.

Cisco Nexus 1000V OpenStack offers rich features, which are not limited to the following:

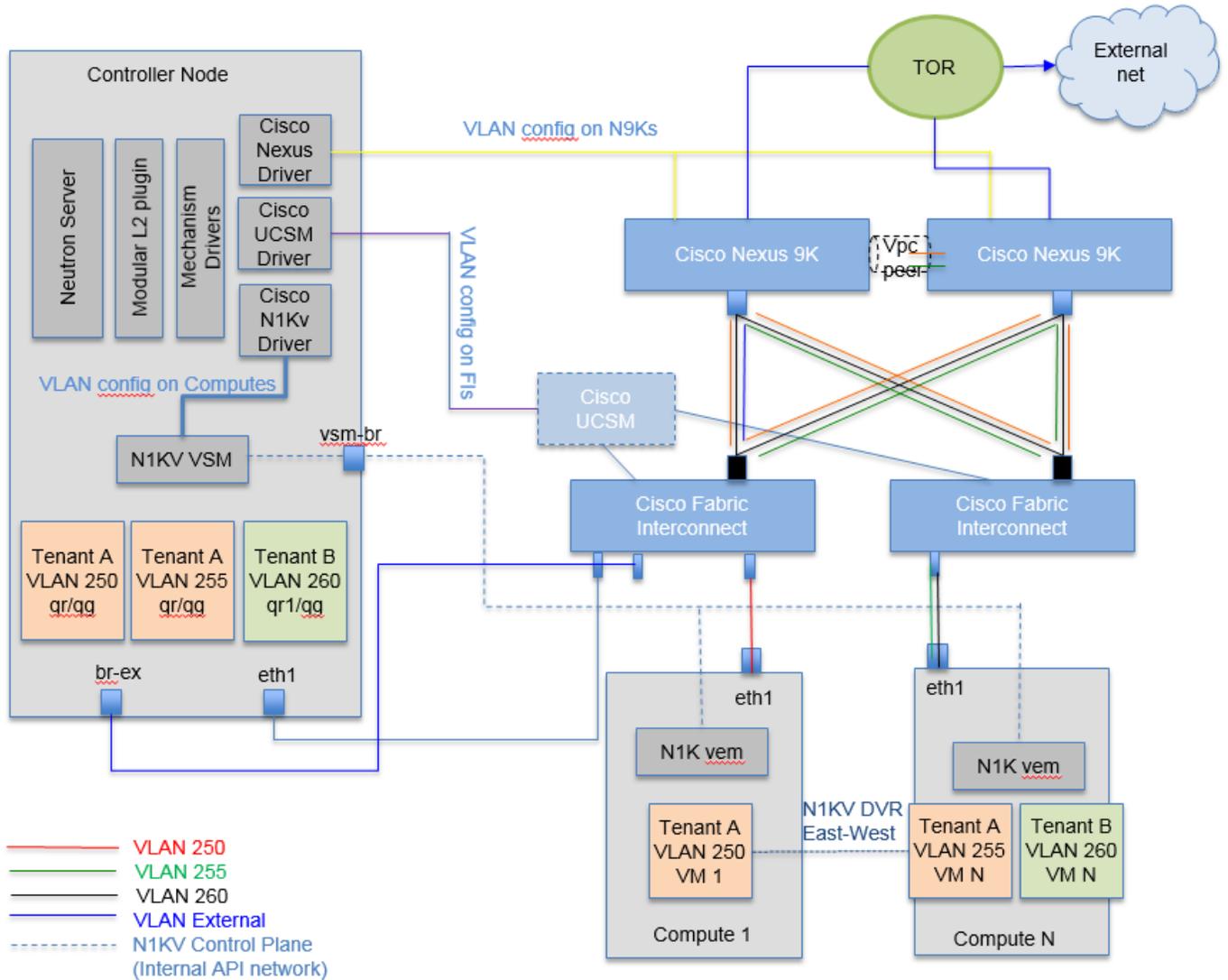
- Layer2/Layer3 Switching
- East-West Security
- Policy Framework

- Application Visibility

All the monitoring, management and functionality features offered on the Nexus 1000V are consentient with the physical Nexus infrastructure. This enables customer to reuse the existing tool chains to manage the new virtual networking infrastructure as well. Along with this, customer can also have the peace of mind that the feature functionality they enjoyed in the physical network will now be the same in the virtual network.

Figure 8 Cisco Plugin Integration with OpenStack

### Cisco Plugins Integration with OpenStack



### Instance creation work flow

To create a virtual machine, complete the following steps:

1. Dashboard/CLI authenticates with Keystone.
2. Dashboard/CLI sends nova-boot to nova-api.
3. nova-api validates the token with keystone.

4. nova-api checks for conflicts, if not creates a new entry in database.
5. nova-api sends rpc.call to nova-scheduler and gets updated host-entry with host-id.
6. nova-scheduler picks up the request from the queue.
7. nova-scheduler sends the rpc.cast request to nova-compute for launching an instance on the appropriate host after applying filters.
8. nova-compute picks up the request from the queue.
9. nova-compute sends the rpc.call request to nova-conductor to fetch the instance information such as host ID and flavor (RAM, CPU, and Disk).
10. nova-conductor picks up the request from the queue.
11. nova-conductor interacts with nova-database and picks up instance information from queue.
12. nova-compute performs the REST with auth-token to glance-api. Then, nova-compute retrieves the Image URI from the Image Service, and loads the image from the image storage.
13. glance-api validates the auth-token with keystone and nova-compute gets the image data.
14. nova-compute performs the REST call to network API to allocated and configure the network
15. neutron server validates the token and creates network info.
16. Nova-compute performs REST to volume API to attach volume to the instance.
17. Cinder-api validates the token and provides block storage info to nova-compute.
18. Nova compute generates data for the hypervisor driver.
19. DHCP and/or Router port bindings by neutron on controller nodes triggers Cisco ML2 plugins:
  - UCSM driver creates VLAN and trunks the eth1 vNICs **for the controller node's service**-profile
  - Nexus driver creates VLAN and trunks the switches port/s mapped to the controller node
  - N1KV VSM receives the logical port information from Neutron. DHCP or Router agents create the port on the N1KV VEM bridge
20. **Virtual Machine's Instance's Port bindings to a Compute Node** triggers again ML2:
  - UCSM driver creates VLAN and trunks the eth1 vNICs for the compute node's service-profile
  - Nexus driver creates VLAN and trunks the switches' **port(s)** mapped to the compute node
  - VSM receives logical port information from Neutron. Nova agent creates the port on the N1KV VEM bridge on the compute node



## Deployment Hardware

---

This section details the deployment hardware used in this solution.

### Cabling Details

Table 4 lists the cabling information.

Table 4 Cabling Details

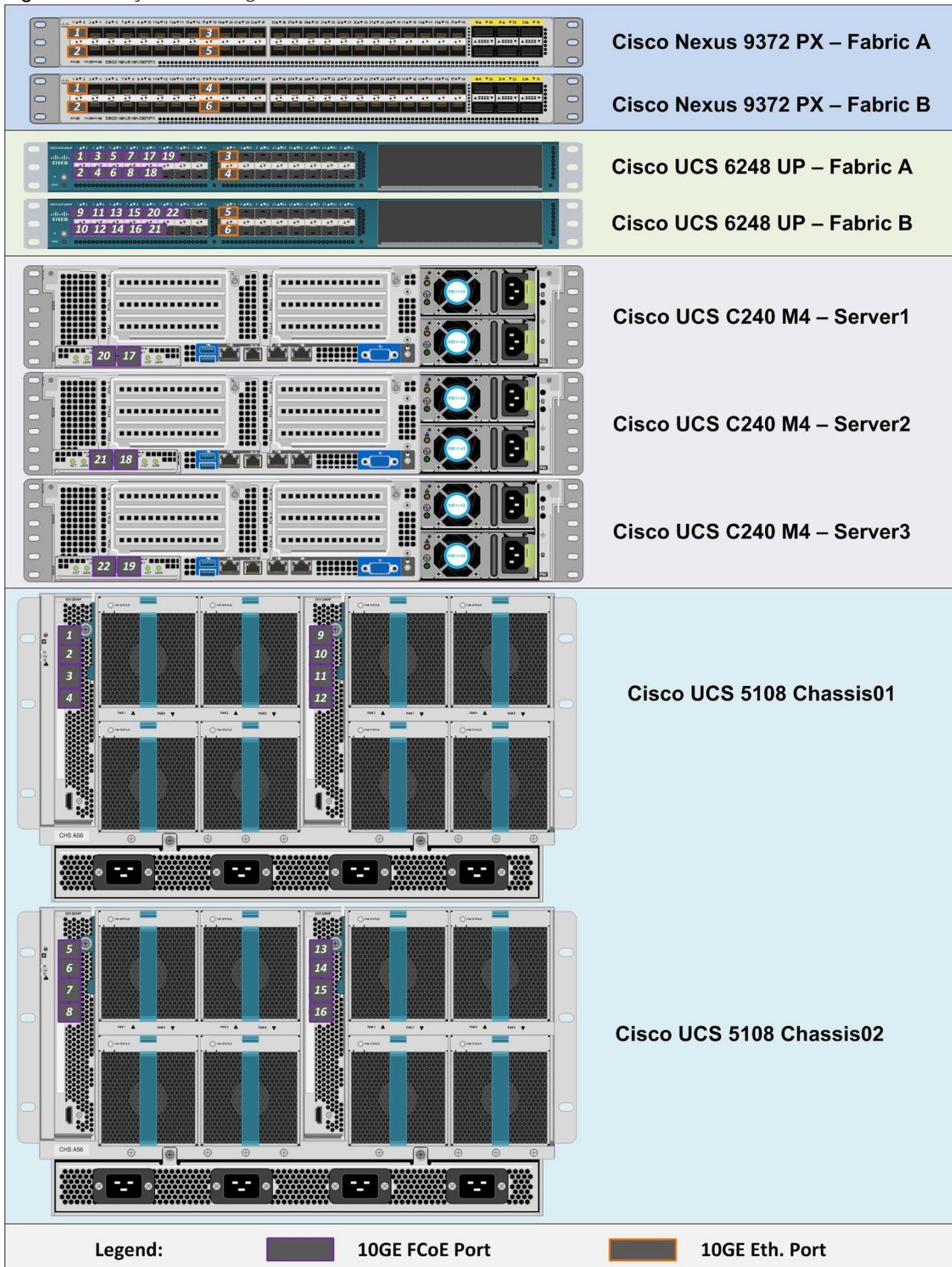
Local Device	Cable Order	Cable Type	Local Port	Connection	Remote Device	Remote Port	Purpose	
Cisco UCS Fabric Interconnect A	1	10G Twin-Ax	Eth1/1	10GbE/FCoE	Chassis 1 FEX A (left)	port 1	To connect UCS chassis1 to UCS Fabric InterconnectA	
	2	10G Twin-Ax	Eth1/2	10GbE/FCoE	Chassis 1 FEX A (left)	port 2	To connect UCS chassis1 to UCS Fabric InterconnectA	
	3	10G Twin-Ax	Eth1/3	10GbE/FCoE	Chassis 1 FEX A (left)	port 3	To connect UCS chassis1 to UCS Fabric InterconnectA	
	4	10G Twin-Ax	Eth1/4	10GbE/FCoE	Chassis 1 FEX A (left)	port 4	To connect UCS chassis1 to UCS Fabric InterconnectA	
	5	10G Twin-Ax	Eth1/5	10GbE/FCoE	Chassis 2 FEX A (left)	port 1	To connect UCS chassis2 to UCS Fabric InterconnectA	
	6	10G Twin-Ax	Eth1/6	10GbE/FCoE	Chassis 2 FEX A (left)	port 2	To connect UCS chassis2 to UCS Fabric InterconnectA	
	7	10G Twin-Ax	Eth1/7	10GbE/FCoE	Chassis 2 FEX A (left)	port 3	To connect UCS chassis2 to UCS Fabric InterconnectA	
	8	10G Twin-Ax	Eth1/8	10GbE/FCoE	Chassis 2 FEX A (left)	port 4	To connect UCS chassis2 to UCS Fabric InterconnectA	
	9	10G Twin-Ax	Eth1/9	10GbE/FCoE	C240 M4 - Server1 - VIC1227	Port 1	To connect UCS C240 Sr1 to UCS Fabric InterconnectA	
	10	10G Twin-Ax	Eth1/10	10GbE/FCoE	C240 M4 - Server2 - VIC1227	Port 1	To connect UCS C240 Sr2 to UCS Fabric InterconnectA	
	11	10G Twin-Ax	Eth1/11	10GbE/FCoE	C240 M4 - Server3 - VIC1227	Port 1	To connect UCS C240 Sr3 to UCS Fabric InterconnectA	
	3	10G Twin-Ax	Eth1/17	10GbE/FCoE	Nexus 9372 Switch A	Eth 1/17	To connect UCS FI-A Networks to Nexus 9k switch A	
	4	10G Twin-Ax	Eth1/18	10GbE/FCoE	Nexus 9372 Switch B	Eth 1/17	To connect UCS FI-B Networks to Nexus 9k switch B	
	1	1G RJ 45	MGMT0	1GbE	Any Management Switch (TOR)	Any	To Connect Management of UCS Fabric Interconnect	
	2	1G RJ 45	L1	1GbE	UCS Fabric Interconnect B	L1	Cluster connection between UCS Fls.	
	3	1G RJ 45	L2	1GbE	UCS Fabric Interconnect B	L2	Cluster connection between UCS Fls.	
Cisco UCS Fabric Interconnect B	12	10G Twin-Ax	Eth1/1	10GbE/FCoE	Chassis 1 FEX B (Right)	port 1	To connect UCS chassis1 to UCS Fabric InterconnectB	
	13	10G Twin-Ax	Eth1/2	10GbE/FCoE	Chassis 1 FEX B (Right)	port 2	To connect UCS chassis1 to UCS Fabric InterconnectB	
	14	10G Twin-Ax	Eth1/3	10GbE/FCoE	Chassis 1 FEX B (Right)	port 3	To connect UCS chassis1 to UCS Fabric InterconnectB	
	15	10G Twin-Ax	Eth1/4	10GbE/FCoE	Chassis 1 FEX B (Right)	port 4	To connect UCS chassis1 to UCS Fabric InterconnectB	
	16	10G Twin-Ax	Eth1/5	10GbE/FCoE	Chassis 2 FEX B (Right)	port 1	To connect UCS chassis2 to UCS Fabric InterconnectB	
	17	10G Twin-Ax	Eth1/6	10GbE/FCoE	Chassis 2 FEX B (Right)	port 2	To connect UCS chassis2 to UCS Fabric InterconnectB	
	18	10G Twin-Ax	Eth1/7	10GbE/FCoE	Chassis 2 FEX B (Right)	port 3	To connect UCS chassis2 to UCS Fabric InterconnectB	
	19	10G Twin-Ax	Eth1/8	10GbE/FCoE	Chassis 2 FEX B (Right)	port 4	To connect UCS chassis2 to UCS Fabric InterconnectB	
	20	10G Twin-Ax	Eth1/9	10GbE/FCoE	C240 M4 - Server1 - VIC1227	Port 2	To connect UCS C240 Sr1 to UCS Fabric InterconnectB	
	21	10G Twin-Ax	Eth1/10	10GbE/FCoE	C240 M4 - Server2 - VIC1227	Port 2	To connect UCS C240 Sr2 to UCS Fabric InterconnectB	
	22	10G Twin-Ax	Eth1/11	10GbE/FCoE	C240 M4 - Server3 - VIC1227	Port 2	To connect UCS C240 Sr3 to UCS Fabric InterconnectB	
		5	10G Twin-Ax	Eth1/17	10GbE/FCoE	Nexus 9372 Switch A	Eth 1/18	To connect UCS FI-A Networks to Nexus 9k switch A
		6	10G Twin-Ax	Eth1/18	10GbE/FCoE	Nexus 9372 Switch B	Eth 1/18	To connect UCS FI-B Networks to Nexus 9k switch B
		4	1G RJ 45	MGMT0	1GbE	Any Management Switch (TOR)	Any	To Connect Management of UCS Fabric Interconnect
	NA	1G RJ 45	L1	1GbE	UCS Fabric Interconnect A	L1	Cluster connection between UCS Fls.	
	NA	1G RJ 45	L2	1GbE	UCS Fabric Interconnect A	L2	Cluster connection between UCS Fls.	

Local Device	Cable Order	Cable Type	Local Port	Connection	Remote Device	Remote Port	Purpose
Cisco Nexus 9372 - Switch A	1	10G Twin-Ax	Eth1/1	10GbE/FCoE	Cisco Nexus 9372 Swith B	Eth1/1	For VPC peerlink
	2	10G Twin-Ax	Eth1/2	10GbE/FCoE	Cisco Nexus 9372 Swith B	Eth1/2	For VPC peerlink
	NA	10G Twin-Ax	Eth1/17	10GbE	Cisco UCS Fabric Interconnect A	Eth1/17	To connect UCS FI-A Networks to Nexus 9k switch A
	NA	10G Twin-Ax	Eth1/18	10GbE	Cisco UCS Fabric Interconnect B	Eth1/17	To connect UCS FI-B Networks to Nexus 9k switch B
	7	1G RJ 45	Eth1/23	1GbE	Upstream Switch	Any	To connect Nexus SwithA Data Network to Upstream switch
	5	1G RJ 45	MGMT0	1GbE	Any Management Switch (TOR)	Any	To connect Management of Nexus switch A
Cisco Nexus 9372 - Switch B	NA	10G Twin-Ax	Eth1/1	10GbE/FCoE	Cisco Nexus 9372 Swith A	Eth1/1	For VPC peerlink
	NA	10G Twin-Ax	Eth1/2	10GbE/FCoE	Cisco Nexus 9372 Swith A	Eth1/2	For VPC peerlink
	NA	10G Twin-Ax	Eth1/17	10GbE	Cisco UCS Fabric Interconnect A	Eth1/18	To connect UCS FI-A Networks to Nexus 9k switch A
	NA	10G Twin-Ax	Eth1/18	10GbE	Cisco UCS Fabric Interconnect B	Eth1/18	To connect UCS FI-B Networks to Nexus 9k switch B
	8	1G RJ 45	Eth1/23	1GbE	Upstream Switch	Any	To connect Nexus SwitchB Data Network to Upstream switch
	6	1G RJ 45	MGMT0	100MbE	Any Management Switch (TOR)	Any	To connect Management of Nexus switch B

## Physical Cabling

Figure 10 illustrates the physical cabling used in this solution.

Figure 10 Physical Cabling



Please note the port numbers on VIC1227 card. Port 1 is on the right and Port 2 is on the left.

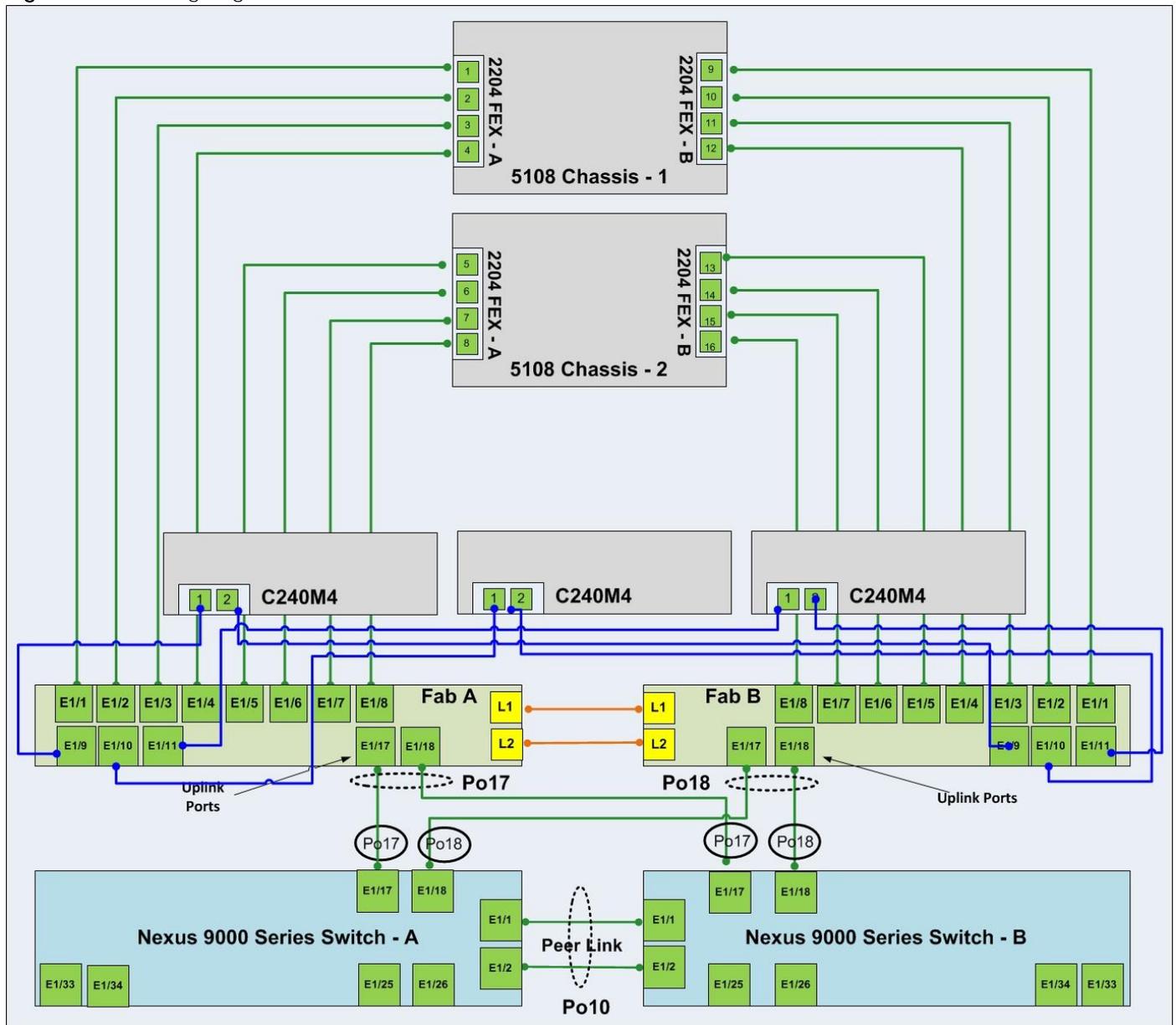
<http://www.cisco.com/c/dam/en/us/products/interfaces-modules/ucs-virtual-interface-card-1227/kO71144-large.jpg>



## Cabling Logic

0 illustrates the cabling logic used in this solution.

Figure 11 Cabling Logic



## Cisco UCS Configuration

### Configure Cisco UCS Fabric Interconnects

Configure the Fabric Interconnects after the cabling is complete. To hook up the console port on the Fabrics, complete the following steps:



Please replace the appropriate addresses for your setup.

Cisco UCS 6248UP Switch A

Connect the console port to the UCS 6248 Fabric Interconnect switch designated for Fabric A:

```

Enter the configuration method: console
Enter the setup mode; setup newly or restore from backup.(setup/restore)? setup
You have chosen to setup a new fabric interconnect? Continue? (y/n): y
Enforce strong passwords? (y/n) [y]: y
Enter the password for "admin": <password>
Enter the same password for "admin": <password>
Is this fabric interconnect part of a cluster (select 'no' for standalone)?
(yes/no) [n]:y
Which switch fabric (A|B): A
Enter the system name: UCS-6248-FAB
Physical switch Mgmt0 IPv4 address: 10.22.100.6
Physical switch Mgmt0 IPv4 netmask: 255.255.255.0
IPv4 address of the default gateway: 10.22.100.1
Cluster IPv4 address: 10.22.100.5
Configure DNS Server IPv4 address? (yes/no) [no]: y
DNS IPv4 address: <<var_nameserver_ip>>
Configure the default domain name? y
Default domain name: <<var_dns_domain_name>>
Join centralized management environment (UCS Central)? (yes/no) [n]: Press Enter
You will be prompted to review the settings.
If they are correct, answer yes to apply and save the configuration. Wait for the
login prompt to make sure that the configuration has been saved.

```

### Cisco UCS 6248UP Switch B

Connect the console port to Peer UCS 6248 Fabric Interconnect switch designated for Fabric B:

```

Enter the configuration method: console
Installer has detected the presence of a peer Fabric interconnect. This Fabric
interconnect will be added to the cluster. Do you want to continue {y|n}? y
Enter the admin password for the peer fabric interconnect: <password>
Physical switch Mgmt0 IPv4 address: 10.22.100.7
Apply and save the configuration (select "no" if you want to re-enter)? (yes/no):
yes

```

Verify the connectivity:

After completing the FI configuration, verify the connectivity as below by logging to one of the Fabrics or the VIP address and checking the cluster state or extended state as shown below:

```
UCSO-6248-FAB-B# show cluster extended-state
Cluster Id: 0x1992ea1a116111e5-0x8ace002a6a3bbba1

Start time: Sun Dec 27 17:05:59 2015
Last election time: Tue Jan 12 22:00:18 2016

B: UP, PRIMARY
A: UP, SUBORDINATE

B: memb state UP, lead state PRIMARY, mgmt services state: UP
A: memb state UP, lead state SUBORDINATE, mgmt services state: UP
  heartbeat state PRIMARY_OK

INTERNAL NETWORK INTERFACES:
eth1, UP
eth2, UP

HA READY
Detailed state of the device selected for HA storage:
Chassis 1, serial: FOX1830H2WT, state: active
Chassis 2, serial: FOX1830GSDP, state: active
Server 3, serial: FCH1912V08S, state: active
UCSO-6248-FAB-B# show cluster state
Cluster Id: 0x1992ea1a116111e5-0x8ace002a6a3bbba1

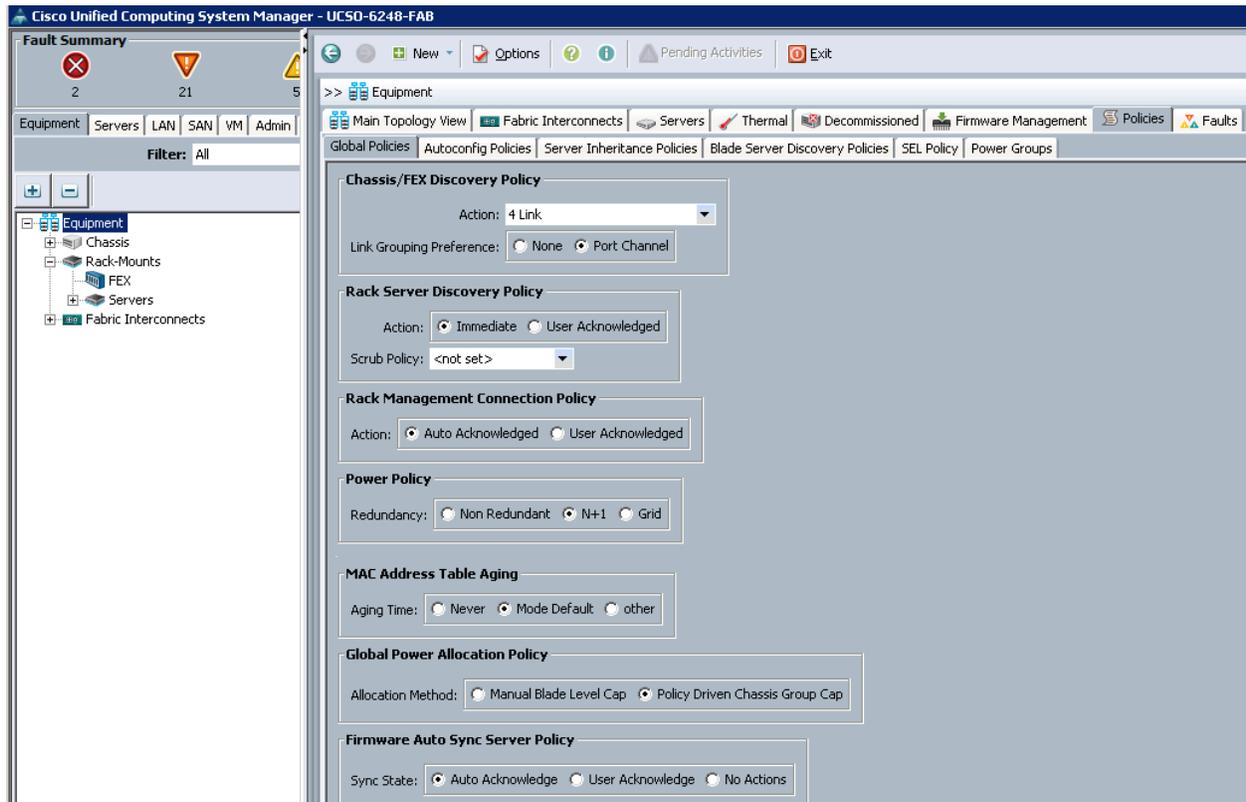
B: UP, PRIMARY
A: UP, SUBORDINATE

HA READY
UCSO-6248-FAB-B# █
```

## Configure the Cisco UCS Global Policies

To configure the Global policies, log into UCS Manager GUI, and complete the following steps:

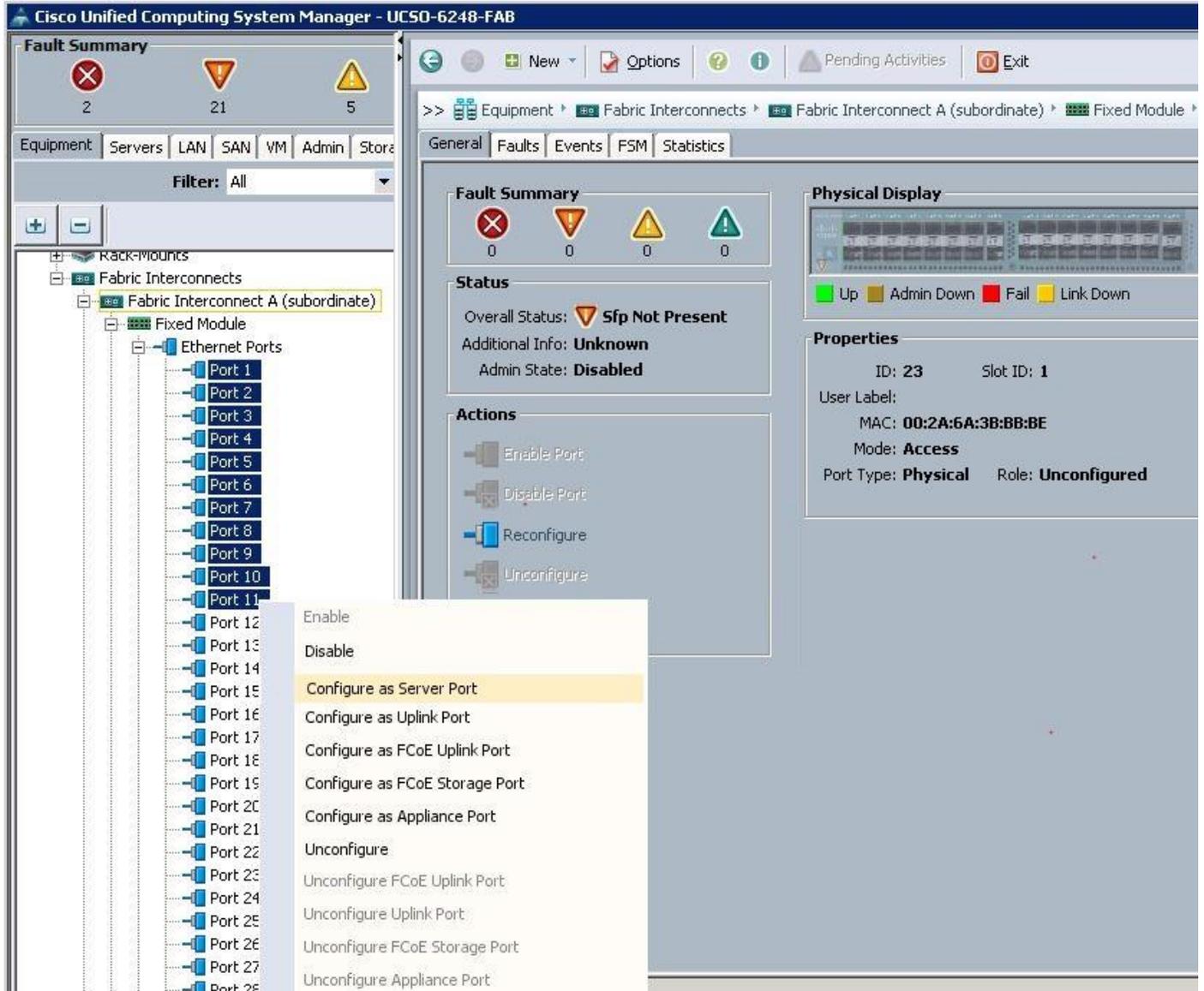
1. Under Equipment → Global Policies;
  - a. Set the Chassis/FEX Discovery Policy to match the number of uplink ports that are cabled between the chassis or fabric extenders and to the fabric interconnects.
  - b. Set the Power policy based on the input power supply to the UCS chassis. In general, UCS chassis with 5 or more blades recommends minimum of 3 power supplies with N+1 configuration. With 4 power supplies, 2 on each PDUs the recommended power policy is Grid.
  - c. Set the Global Power allocation Policy as Policy driven Chassis Group cap.
  - d. Click Save changes to save the configuration.



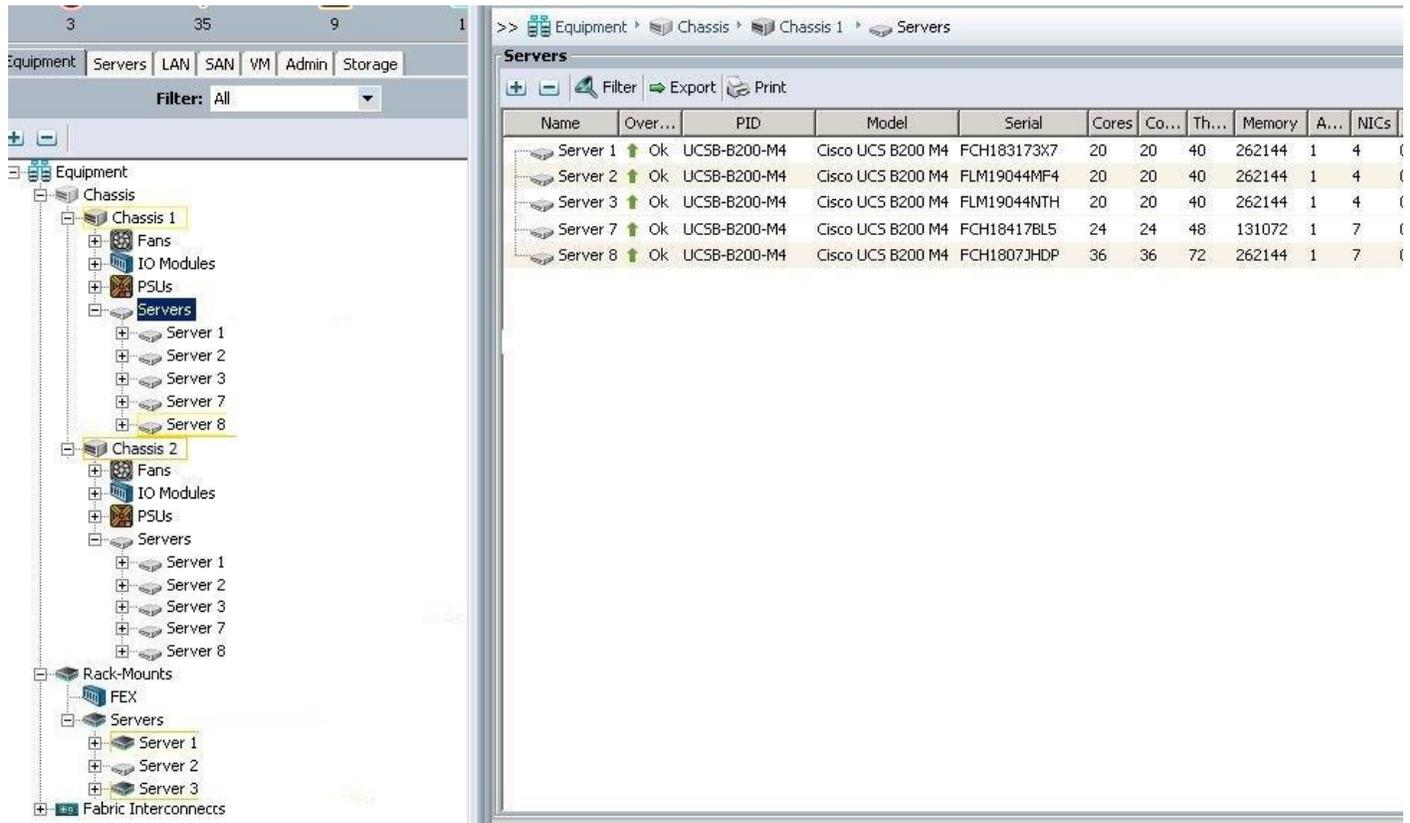
## Configure Server Ports for Blade Discovery and Rack Discovery

Navigate to each Fabric Interconnect and configure the server ports on Fabric Interconnects. Complete the following steps:

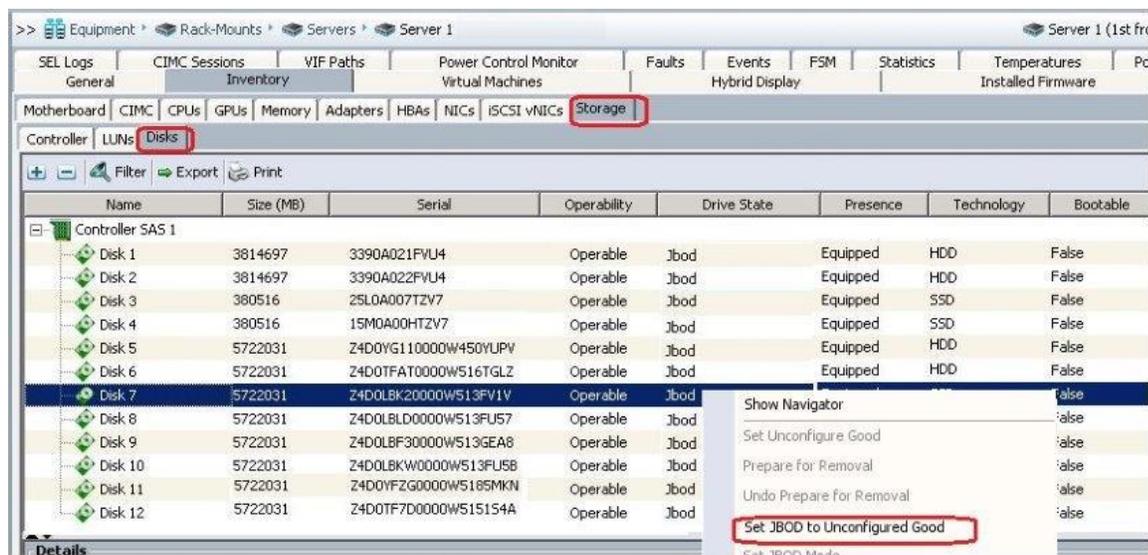
1. Under Equipment → Fabric Interconnects → Fabric Interconnect A → Fixed Module → Ethernet Ports;
  - a. Select the ports (Port 1 to 8) that are connected to the left side of each UCS chassis FEX 2204, right-click them and select Configure as Server Port.
  - b. Select the ports (Port 9 to 11) that are connected to the 10G MLOM (VIC1227) port1 of each UCS C240 M4, right-click them, and select Configure as Server Port.
  - c. Click Save Changes to save the configuration.
  - d. Repeat steps 1 and 2 on Fabric Interconnect B and save the configuration.



After this the blades and rack servers will be discovered as shown below:



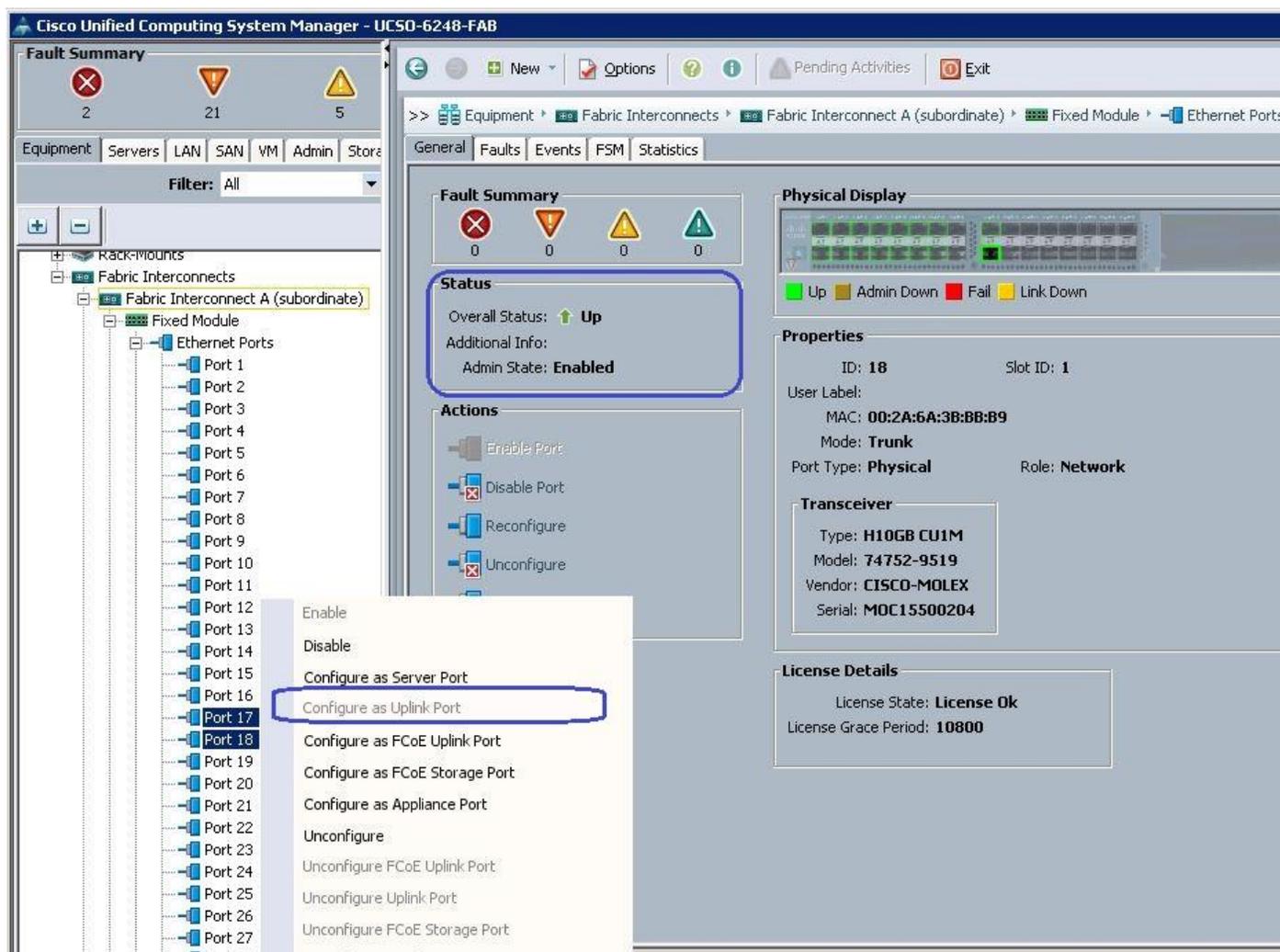
Navigate to each blade and rack servers to make sure that the disks are in Unconfigured Good state, else convert jbod to Unconfigured as below. The below diagram show how to convert a disk to Unconfigured Good state.



## Configure Network Uplinks

Navigate to each Fabric Interconnects and configure the Network Uplink ports on Fabric Interconnects. Complete the following steps:

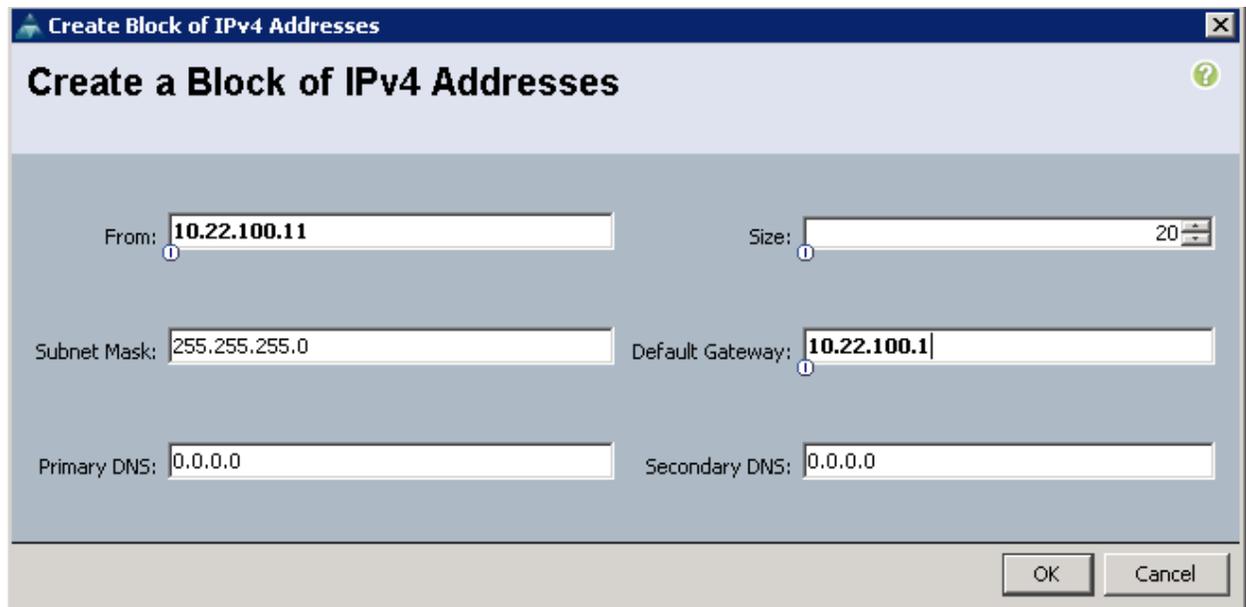
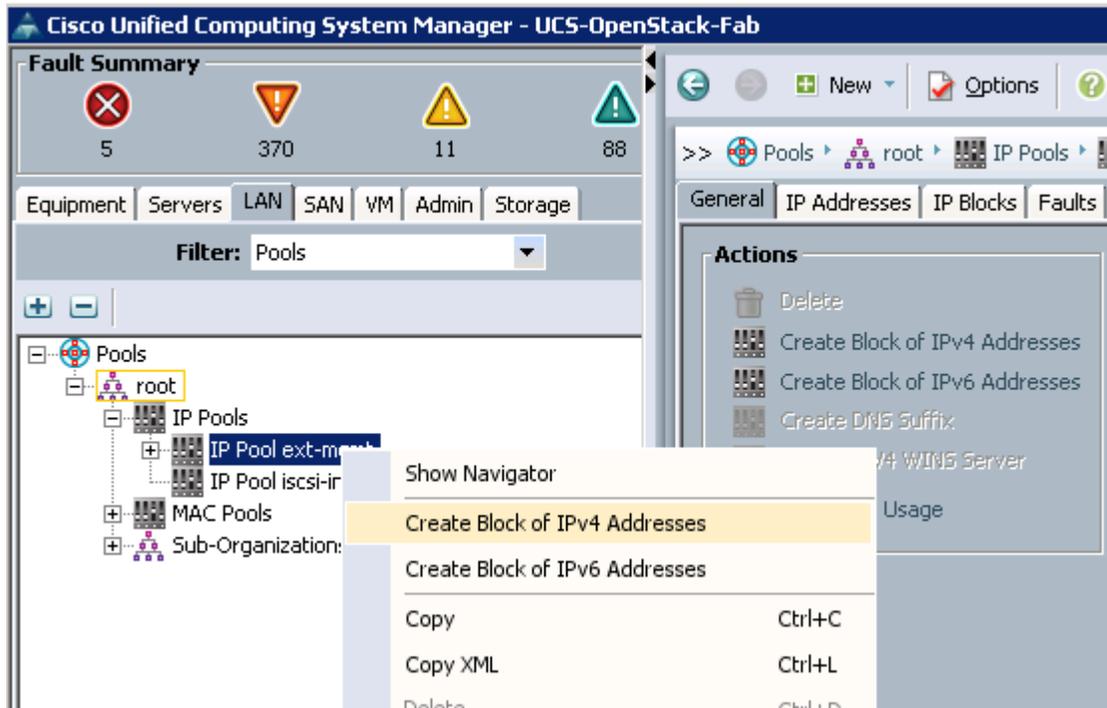
1. Under Equipment → Fabric Interconnects → Fabric Interconnect A → Fixed Module → Ethernet Ports
  - a. Select the port 17 and Port18 that are connected to Nexus 9k switches, right-click them and select Configure as Uplink Port.
  - b. Click Save Changes to save the configuration.
  - c. Repeat the steps 1 and 2 on Fabric Interconnect B.



## Create KVM IP Pools

To access the KVM console of each UCS Server, create the KVM IP pools from the UCS Manager GUI, and complete the following steps:

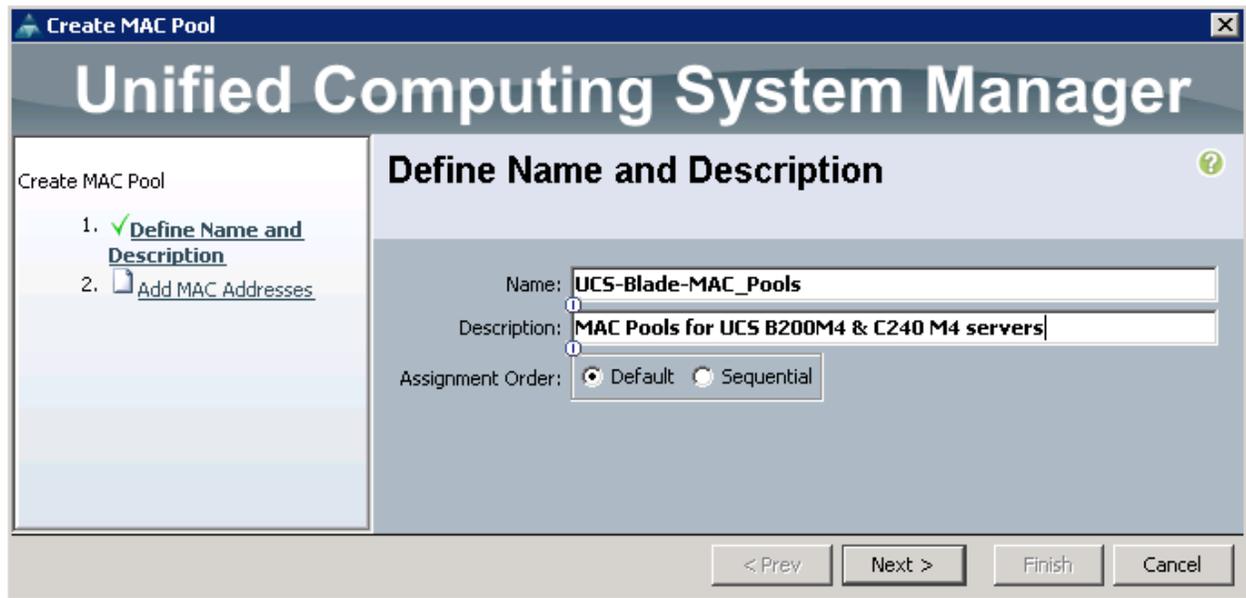
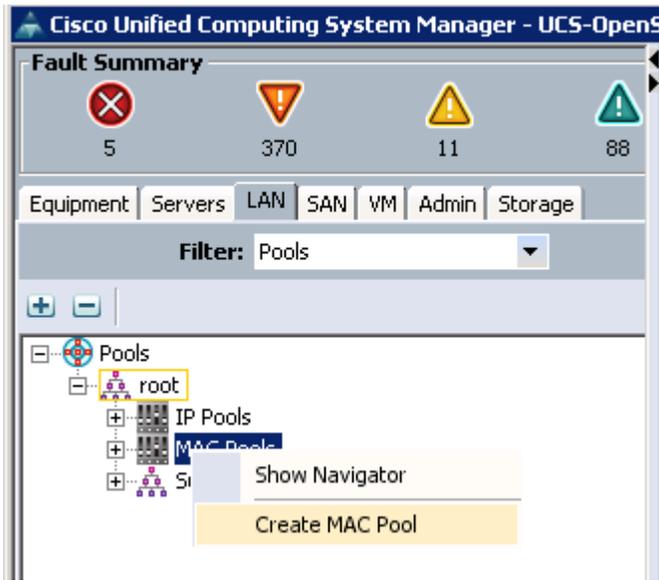
1. Under LAN → Pools → root → IP Pools → IP Pool ext-mgmt → right-click and select Create Block of IPv4 addresses.
2. Specify the Starting IP address, subnet mask and gateway and size.



### Create MAC Pools

To configure a MAC address for each Cisco UCS Server VNIC interface, create the MAC pools from the Cisco UCS Manager GUI, and complete the following steps:

1. Under LAN → Pools → root → MAC Pools → right-click and select Create MAC Pool.
2. Specify the name and description for the MAC pool.

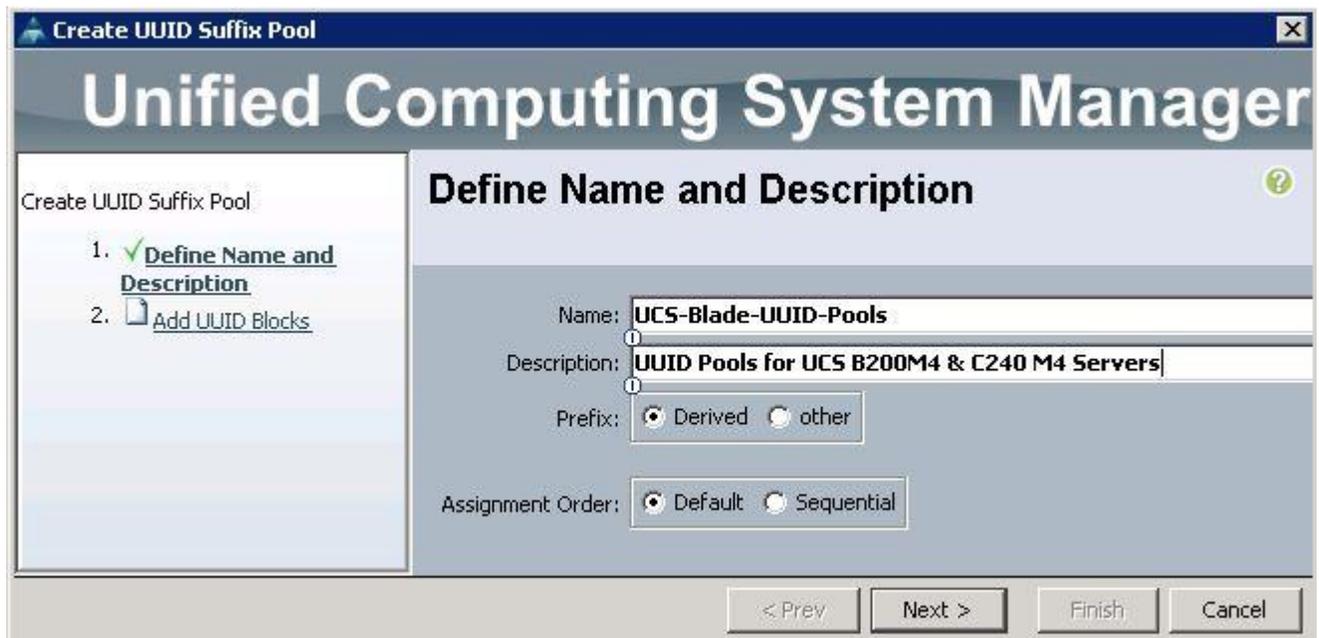
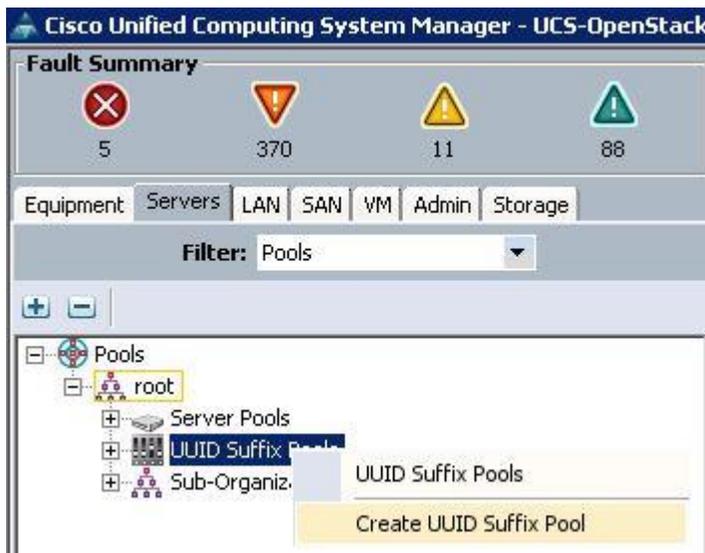


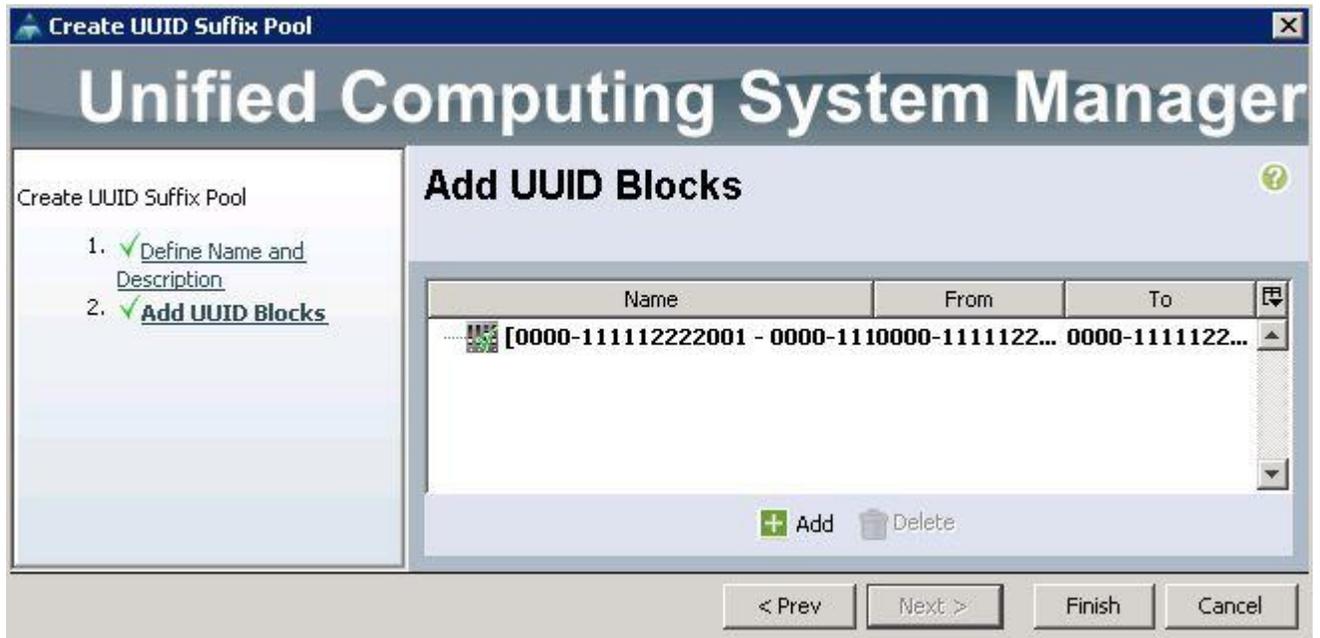
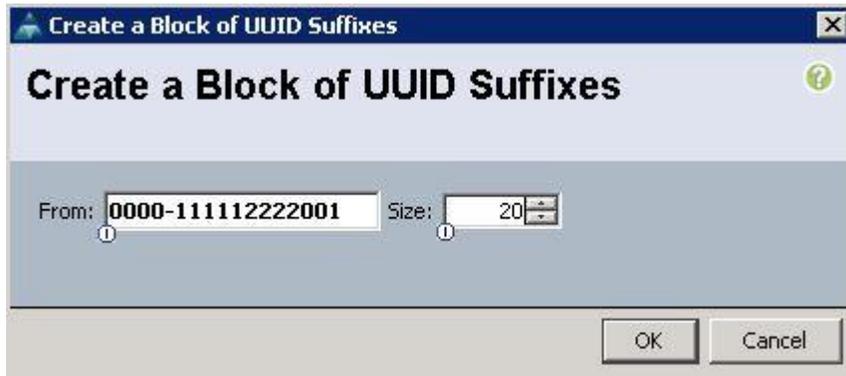
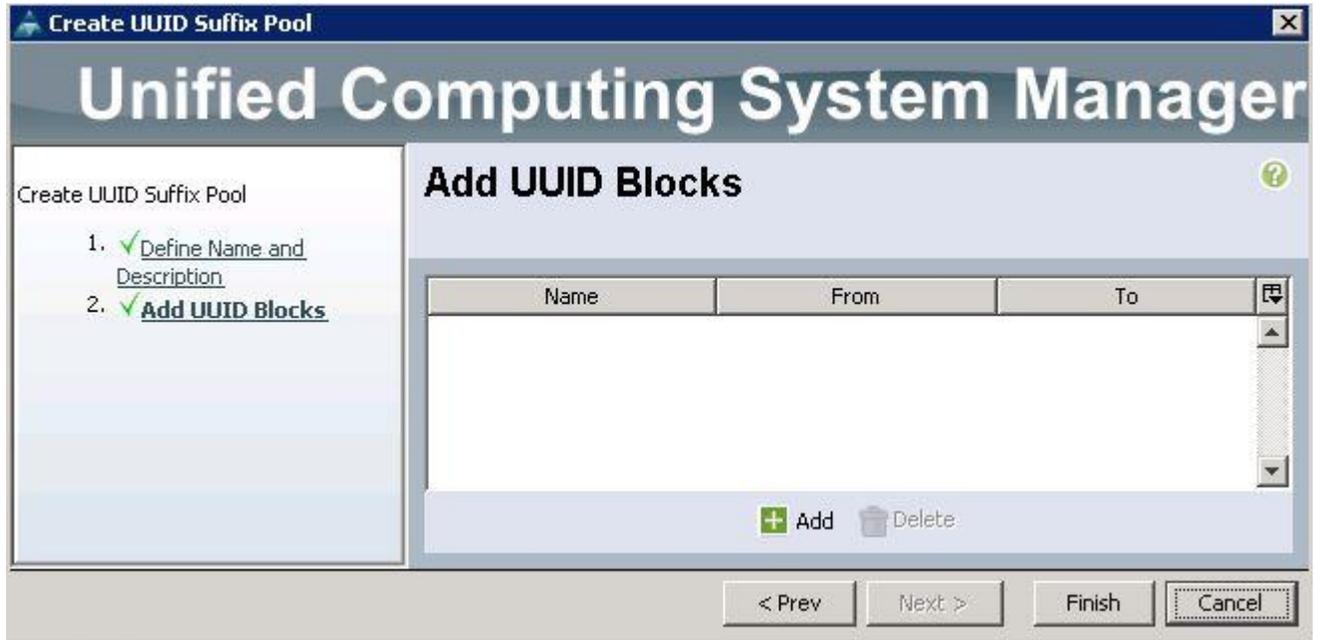


## Create UUID Pools

To configure the UUID pools for each UCS Server, create the UUID pools from the Cisco UCS Manager GUI, complete the following steps:

1. Under Servers → Pools → root → UUID Suffix Pools → right-click and select Create UUID Suffix Pool.
2. Specify the name and description for the UUID pool.
3. Click Add.
4. Specify the UUID Suffixes and size for the UUID pool.
5. Click Finish to complete the UUID pool creation.

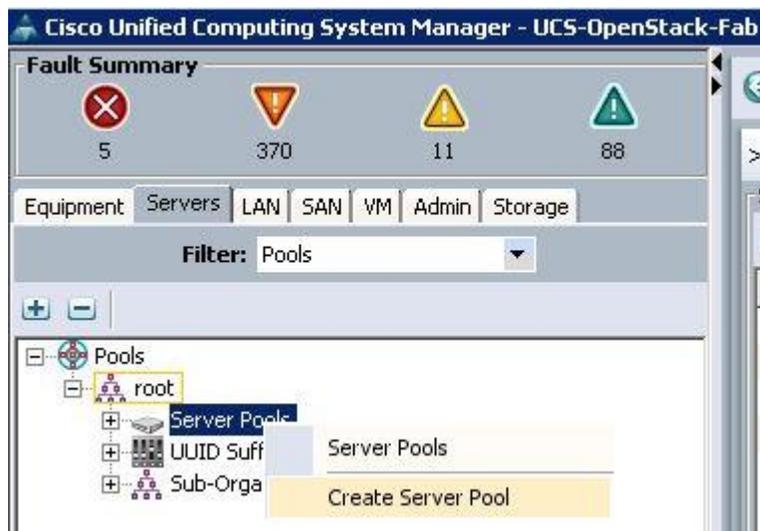


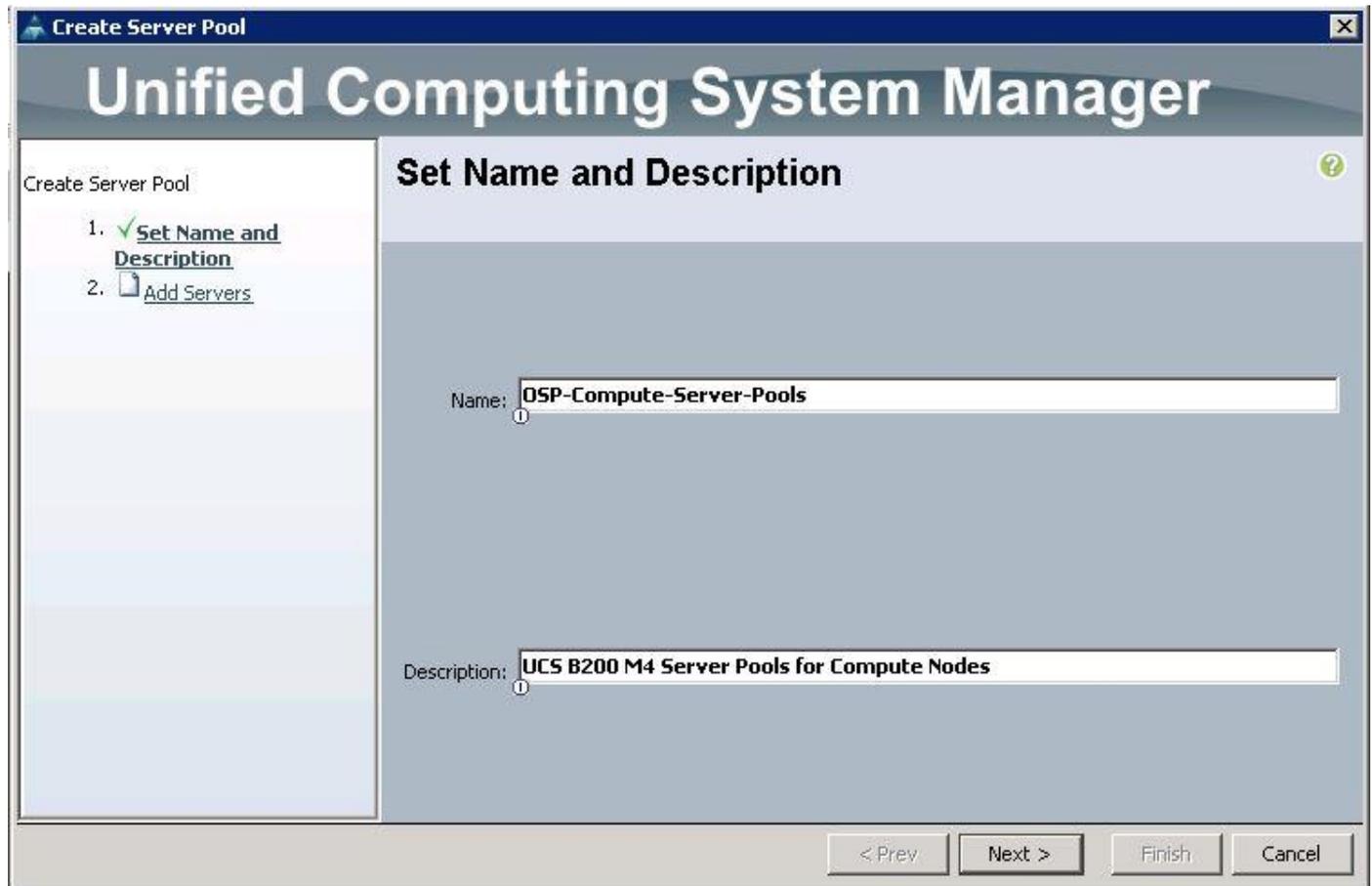


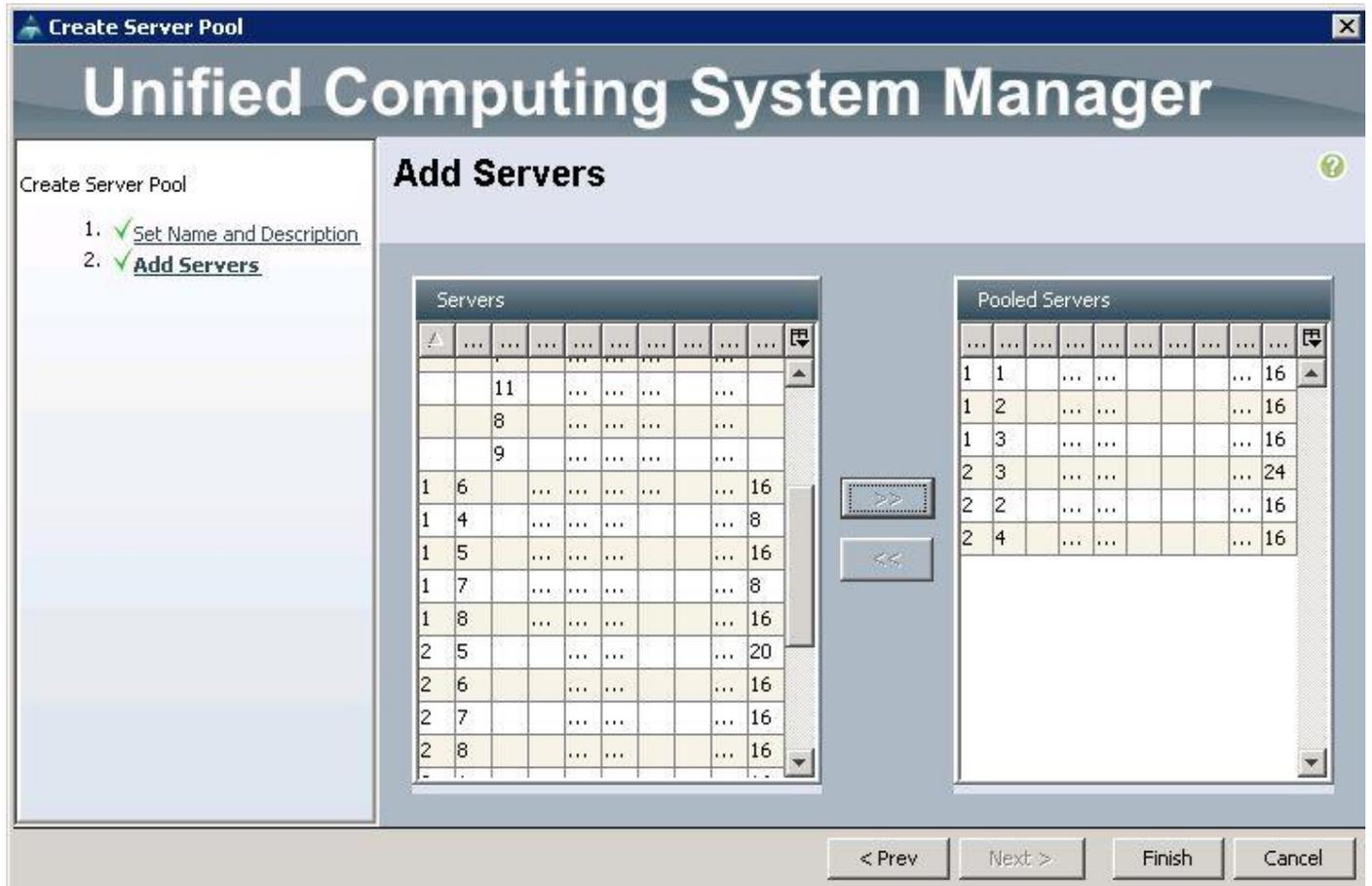
## Create Server Pools for Controller, Compute and Ceph Storage Nodes

To configure Server pools for Controller, Compute and Ceph Storage Servers, create Server pools from the UCS Manager GUI, and complete the following steps:

1. Under Servers → Pools → root → Server Pools → right-click and select Create Server Pool.
2. Specify the name and description for the Server pool for Compute Nodes.
3. Similarly, create Server pools for Controller and Ceph Storage Nodes.



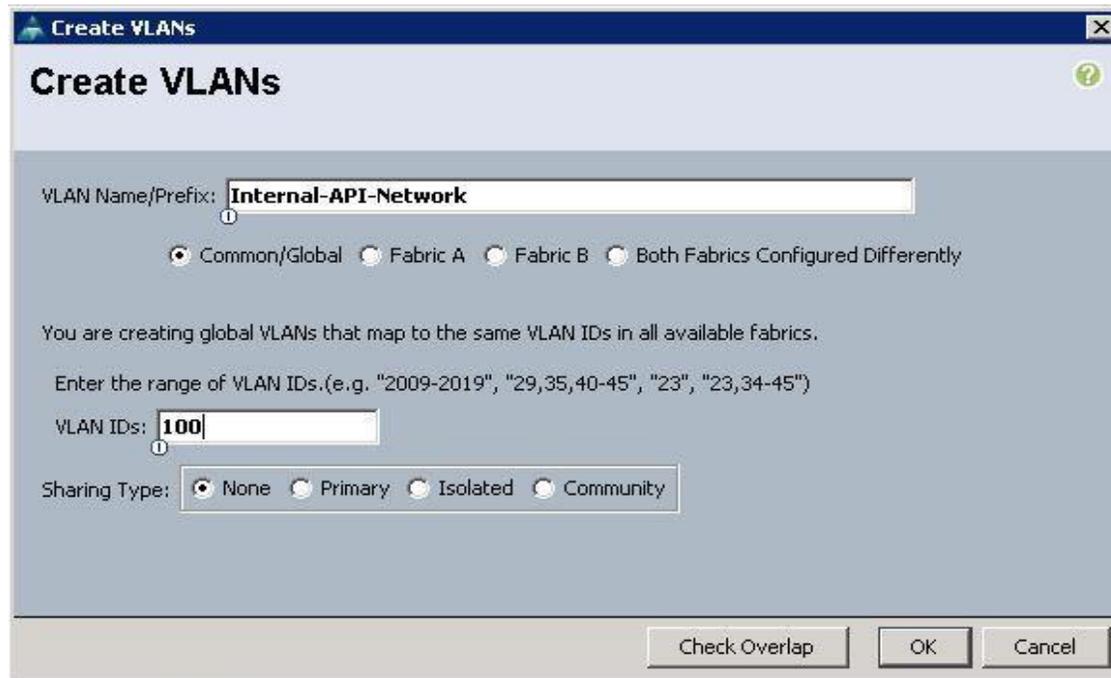
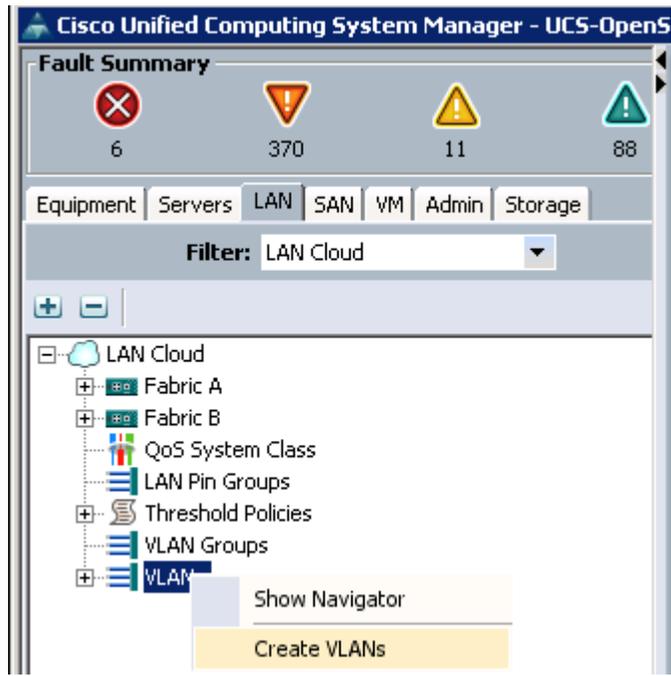




## Create VLANs

To create VLANs for all OpenStack networks for Controller, Compute and Ceph Storage Servers, from the UCS manager GUI, complete the following steps:

1. Under LAN → LAN Cloud → VLANs → right-click and select Create VLANs.



2. Specify the VLAN name as PXE-Network for Provisioning and specify the VLAN ID as 110 and click OK.

**Create VLANs**

VLAN Name/Prefix:

Common/Global  Fabric A  Fabric B  Both Fabrics Configured Differently

You are creating global VLANs that map to the same VLAN IDs in all available fabrics.

Enter the range of VLAN IDs.(e.g. "2009-2019", "29,35,40-45", "23", "23,34-45")

VLAN IDs:

Sharing Type:  None  Primary  Isolated  Community

3. Specify the VLAN name as Storage-Public for accessing Ceph Storage Public Network and specify the VLAN ID as 120 and click OK.

**Create VLANs**

VLAN Name/Prefix:

Common/Global  Fabric A  Fabric B  Both Fabrics Configured Differently

You are creating global VLANs that map to the same VLAN IDs in all available fabrics.

Enter the range of VLAN IDs.(e.g. "2009-2019", "29,35,40-45", "23", "23,34-45")

VLAN IDs:

Sharing Type:  None  Primary  Isolated  Community

4. Specify the VLAN name as Tenant-Internal-Network and specify the VLAN ID as 130 and click OK.

**Create VLANs**

VLAN Name/Prefix:

Common/Global  Fabric A  Fabric B  Both Fabrics Configured Differently

You are creating global VLANs that map to the same VLAN IDs in all available fabrics.

Enter the range of VLAN IDs.(e.g. "2009-2019", "29,35,40-45", "23", "23,34-45")

VLAN IDs:

Sharing Type:  None  Primary  Isolated  Community

5. Specify the VLAN name as Storage-Mgmt-Network for Managing Ceph Storage Cluster and specify the VLAN ID as 130 and click OK.

**Create VLANs**

VLAN Name/Prefix:

Common/Global  Fabric A  Fabric B  Both Fabrics Configured Differently

You are creating global VLANs that map to the same VLAN IDs in all available fabrics.

Enter the range of VLAN IDs.(e.g. "2009-2019", "29,35,40-45", "23", "23,34-45")

VLAN IDs:

Sharing Type:  None  Primary  Isolated  Community

6. Specify the VLAN name as External-Network and specify the VLAN ID as 215 and click OK.

**Create VLANs**

VLAN Name/Prefix:

Common/Global
  Fabric A
  Fabric B
  Both Fabrics Configured Differently

You are creating global VLANs that map to the same VLAN IDs in all available fabrics.

Enter the range of VLAN IDs.(e.g. "2009-2019", "29,35,40-45", "23", "23,34-45")

VLAN IDs:

Sharing Type:
  None
  Primary
  Isolated
  Community

- Specify the VLAN name as Tenant-Floating-Network for accessing Tenant instances externally and specify the VLAN ID as 160 and click OK.



This network is Optional. In this solution, we only used a 24 bit netmask for the External network that had a limitation of 250 IPs for tenant VMs. Due to this limitation, we used a 20 bit netmask for the Tenant Floating Network.

**Create VLANs**

VLAN Name/Prefix:

Common/Global
  Fabric A
  Fabric B
  Both Fabrics Configured Differently

You are creating global VLANs that map to the same VLAN IDs in all available fabrics.

Enter the range of VLAN IDs.(e.g. "2009-2019", "29,35,40-45", "23", "23,34-45")

VLAN IDs:

Sharing Type:
  None
  Primary
  Isolated
  Community

The screenshot below shows the output of VLANs for all the OpenStack Networks created above.

Name	ID	Type	Transport	Native	VLAN Sharing
VLAN External-Network (215)	215	Lan	Ether	No	None
VLAN Internal-API-Network (100)	100	Lan	Ether	No	None
VLAN PXE-Network (110)	110	Lan	Ether	No	None
VLAN Storage-Mgmt-Network (150)	150	Lan	Ether	No	None
VLAN Storage-Public-Network (120)	120	Lan	Ether	No	None
VLAN Tenant-Floating-IP-Network (...)	160	Lan	Ether	No	None
VLAN Tenant-Internal-Network (130)	130	Lan	Ether	No	None
VLAN default (1)	1	Lan	Ether	Yes	None

## Create a Network Control Policy

To configure the Network Control policy from the UCS Manager, complete the following steps:

1. Under LAN → Policies → root → Network Control Policies → right-click and select Create Network Control Policy.
  - a. Specify the name and choose CDP as Enabled. Select the MAC register mode as "All hosts VLANs" and Action on Uplink fail as "Link Down" and click OK.

**Create Network Control Policy**

Name:

Description:

CDP:  Disabled  Enabled

MAC Register Mode:  Only Native Wlan  All Host Vlan

Action on Uplink Fail:  Link Down  Warning

**MAC Security**

Forge:  Allow  Deny

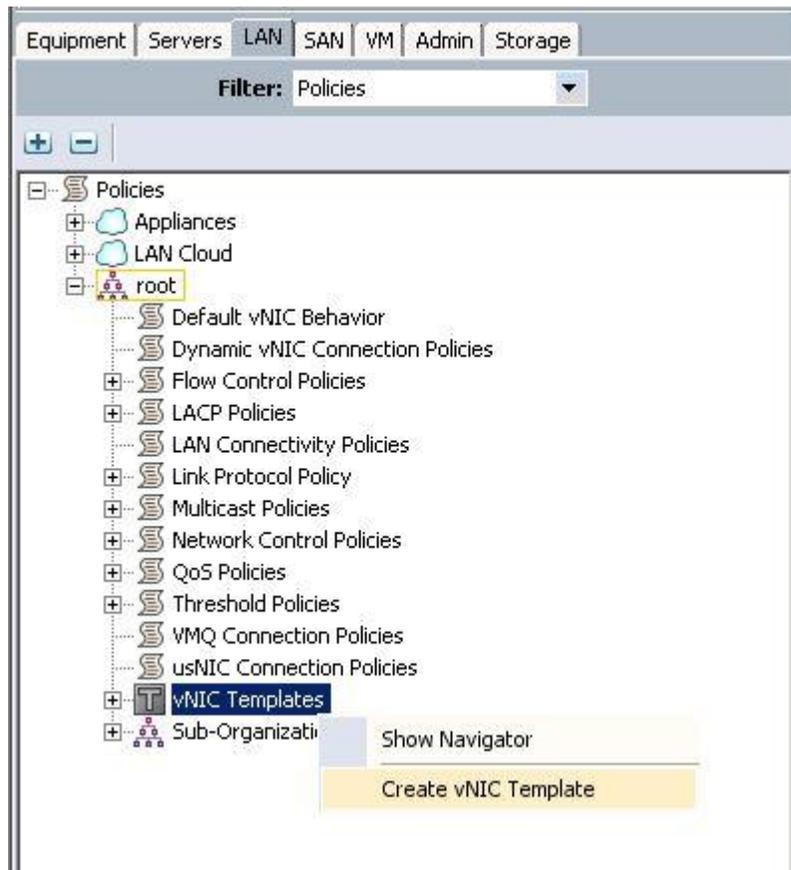
## Create vNIC Templates

To Configure vNIC templates for each UCS Server vNIC interfaces, create vNIC templates from the UCS Manager GUI, complete the following steps:



The storage management network is configured with 9000 MTU.

1. Under LAN → Policies → root → vNIC Templates → right-click and select Create vNIC Template.



- a. Create vNIC template for Internal-API network. Specify the name, description, Fabric ID, VLAN ID and choose MAC pools from the drop-down list.

**Create vNIC Template**

Name:

Description:

Fabric ID:  Fabric A  Fabric B  Enable Failover

**Target**

Adapter  
 VM

**Warning**  
If **VM** is selected, a port profile by the same name will be created.  
If a port profile of the same name exists, and updating template is selected, it will be overwritten

Template Type:  Initial Template  Updating Template

**VLANs**

Filter Export Print

Select	Name	Native VLAN
<input type="checkbox"/>	default	<input type="radio"/>
<input type="checkbox"/>	External	<input type="radio"/>
<input checked="" type="checkbox"/>	Internal-API	<input type="radio"/>
<input type="checkbox"/>	OSP-PXE	<input type="radio"/>
<input type="checkbox"/>	Storage-Mgmt	<input type="radio"/>
<input type="checkbox"/>	Storage-Pub	<input type="radio"/>

**Create VLAN**

MTU:

MAC Pool:

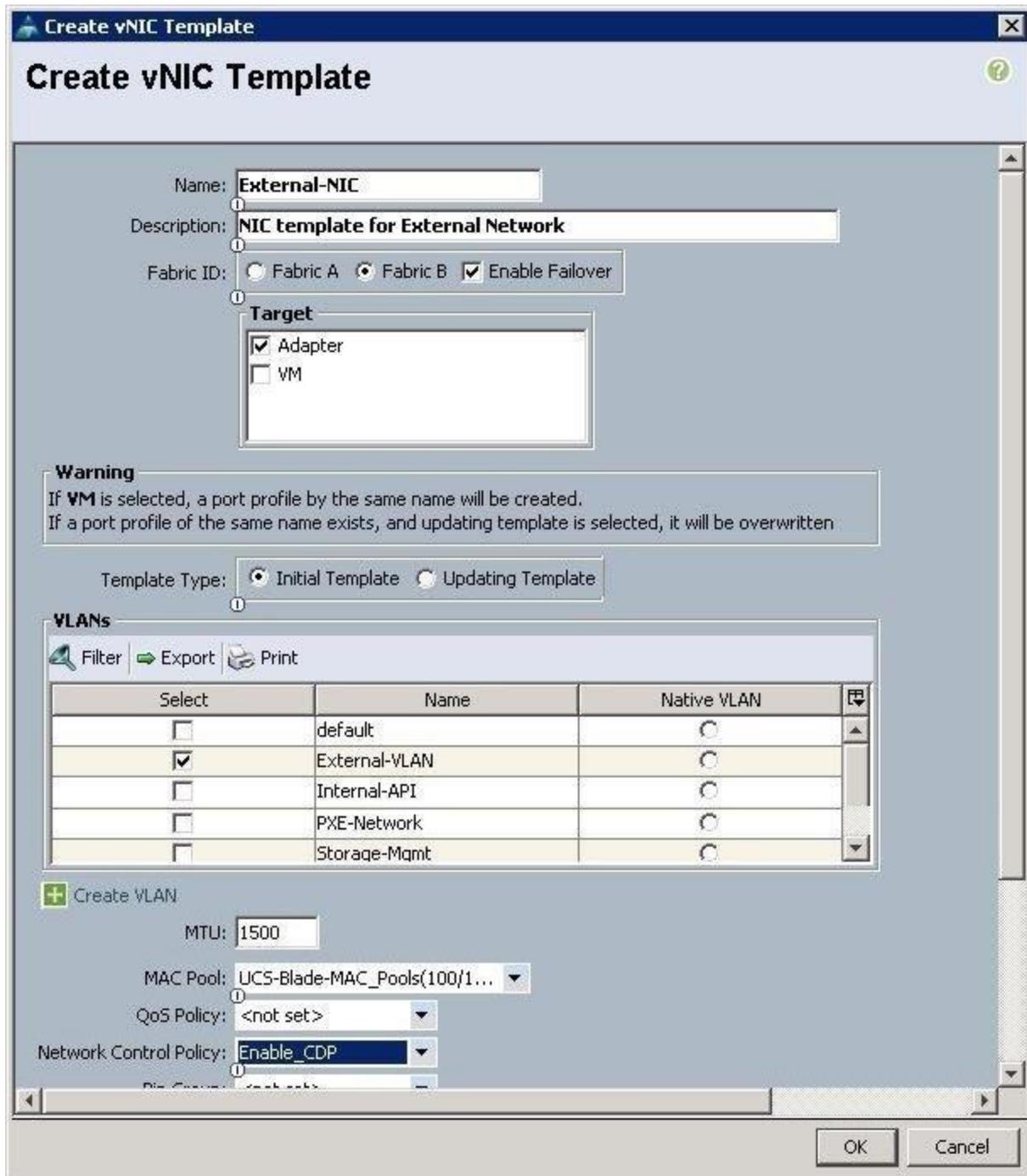
QoS Policy:

Network Control Policy:

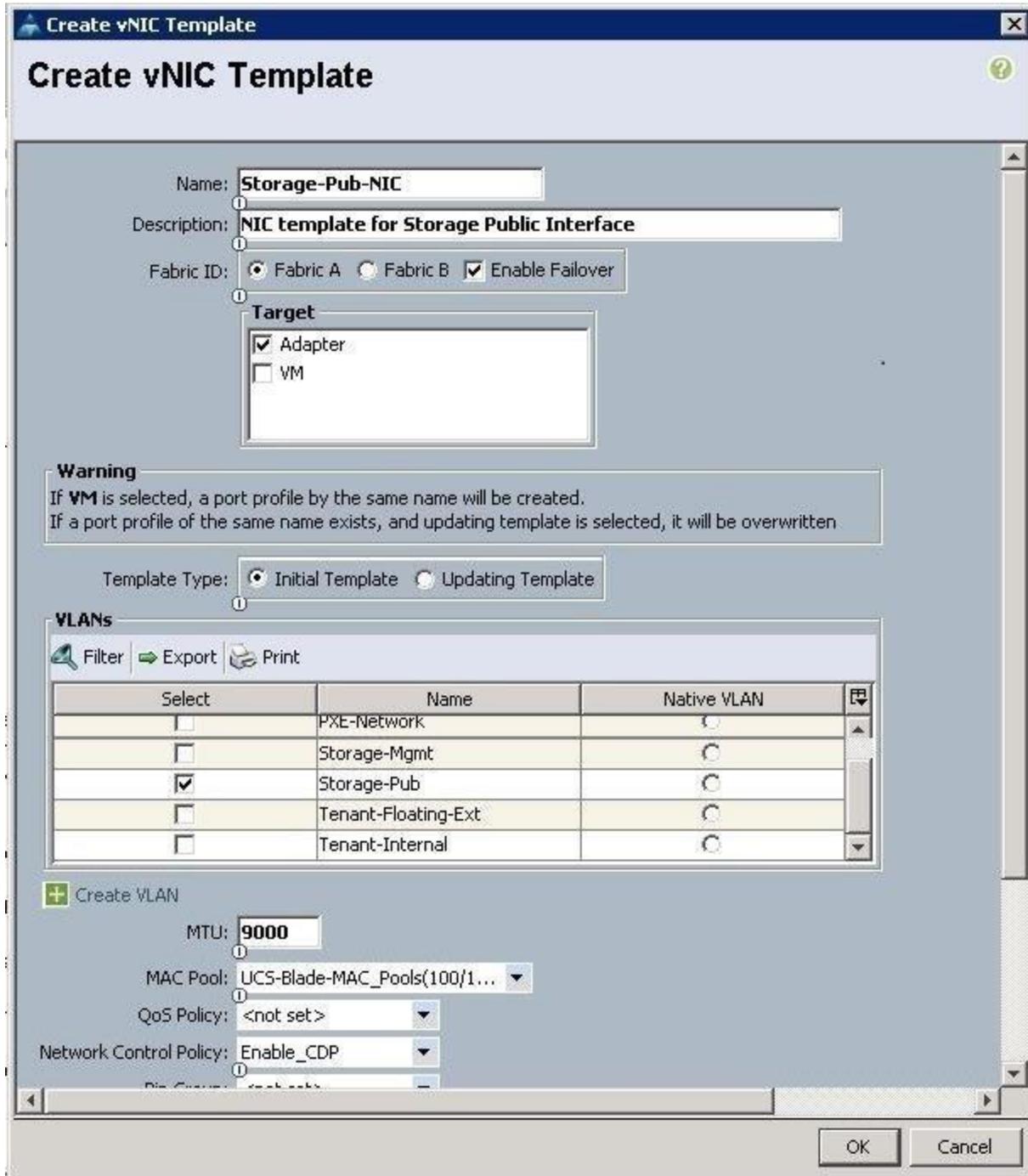
Pin Group:

OK Cancel

b. Create vNIC template for External Network.



c. Create the vNIC template for Storage Public Network.



d. Create vNIC template for Storage Mgmt Cluster network.

**Create vNIC Template**

Name:

Description:

Fabric ID:  Fabric A  Fabric B  Enable Failover

**Target**

Adapter  
 VM

**Warning**  
If **VM** is selected, a port profile by the same name will be created.  
If a port profile of the same name exists, and updating template is selected, it will be overwritten

Template Type:  Initial Template  Updating Template

**VLANs**

Filter Export Print

Select	Name	Native VLAN
<input type="checkbox"/>	PXE-Network	<input type="radio"/>
<input checked="" type="checkbox"/>	Storage-Mgmt	<input type="radio"/>
<input type="checkbox"/>	Storage-Pub	<input type="radio"/>
<input type="checkbox"/>	Tenant-Floating-Ext	<input type="radio"/>
<input type="checkbox"/>	Tenant-Internal	<input type="radio"/>

**Create VLAN**

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Pin Group:

OK Cancel

e. Create vNIC template for Tenant Internal Network.

**Create vNIC Template**

Name:

Description:

Fabric ID:  Fabric A  Fabric B  Enable Failover

**Target**

Adapter  
 VM

**Warning**  
If **VM** is selected, a port profile by the same name will be created.  
If a port profile of the same name exists, and updating template is selected, it will be overwritten

Template Type:  Initial Template  Updating Template

**VLANs**

Select	Name	Native VLAN
<input type="checkbox"/>	Internal-API	<input type="radio"/>
<input type="checkbox"/>	OSP-PXE	<input type="radio"/>
<input type="checkbox"/>	Storage-Mgmt	<input type="radio"/>
<input type="checkbox"/>	Storage-Pub	<input type="radio"/>
<input type="checkbox"/>	Tenant-Floating-Ext	<input type="radio"/>
<input checked="" type="checkbox"/>	Tenant-Internal	<input type="radio"/>

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Pin Group:

Stats Threshold Policy:

f. Create vNIC template for Tenant Floating Network.

**Create vNIC Template**

Name:

Description:

Fabric ID:  Fabric A  Fabric B  Enable Failover

**Target**

Adapter  
 VM

**Warning**  
If **VM** is selected, a port profile by the same name will be created.  
If a port profile of the same name exists, and updating template is selected, it will be overwritten

Template Type:  Initial Template  Updating Template

**VLANs**

Filter Export Print

Select	Name	Native VLAN
<input type="checkbox"/>	PXE-Network	<input type="radio"/>
<input type="checkbox"/>	Storage-Mgmt	<input type="radio"/>
<input type="checkbox"/>	Storage-Pub	<input type="radio"/>
<input checked="" type="checkbox"/>	Tenant-Floating-Ext	<input type="radio"/>
<input type="checkbox"/>	Tenant-Internal	<input type="radio"/>

**+ Create VLAN**

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Dis. Group:

OK Cancel

After completion, you can see the vNIC templates for each traffic.

Name	VLAN	Native VLAN
vNIC Template External-NIC	External-VLAN	<input type="radio"/>
vNIC Template Internal-API-NIC	Internal-API	<input type="radio"/>
vNIC Template PXE-NIC	PXE-Network	<input checked="" type="radio"/>
vNIC Template Storage-Mgmt-NIC	Storage-Mgmt	<input type="radio"/>
vNIC Template Storage-Pub-NIC	Storage-Pub	<input type="radio"/>
vNIC Template Tenant-Internal	Tenant-Internal	<input type="radio"/>
vNIC Template Tenant_Floating	Tenant-Floating-Ext	<input type="radio"/>

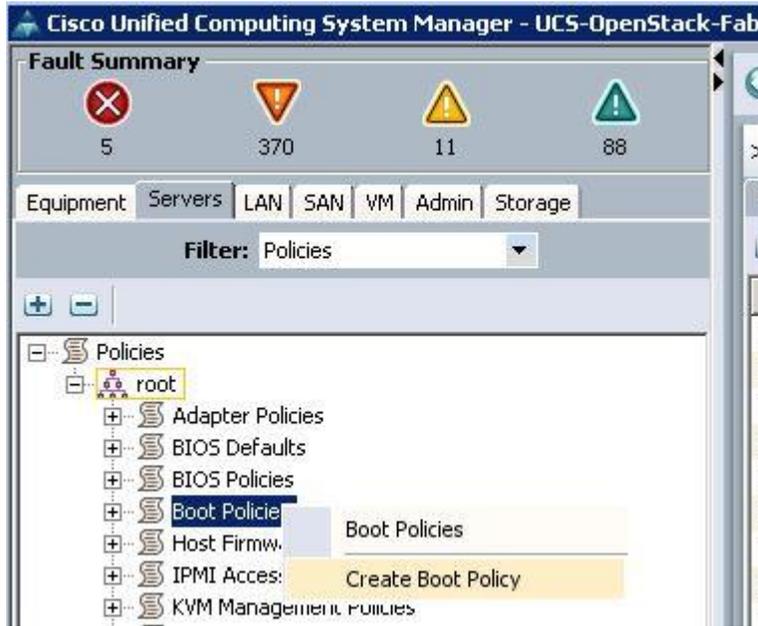


For storage interfaces, a MTU value of 9000 has been added.

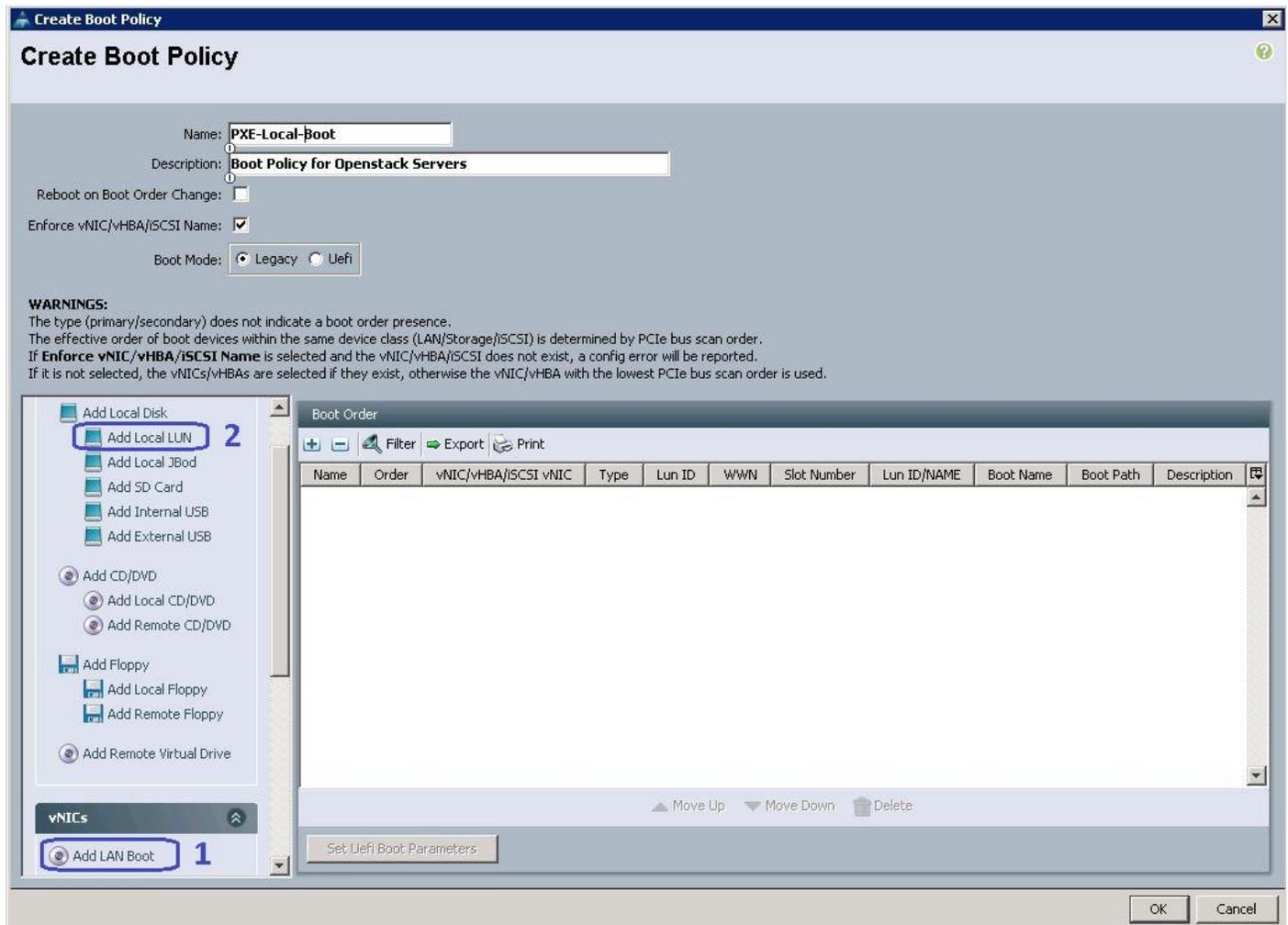
## Create Boot Policy

To configure the Boot policy for the Cisco UCS Servers, create a Boot Policy from the Cisco UCS Manager GUI, and complete the following steps:

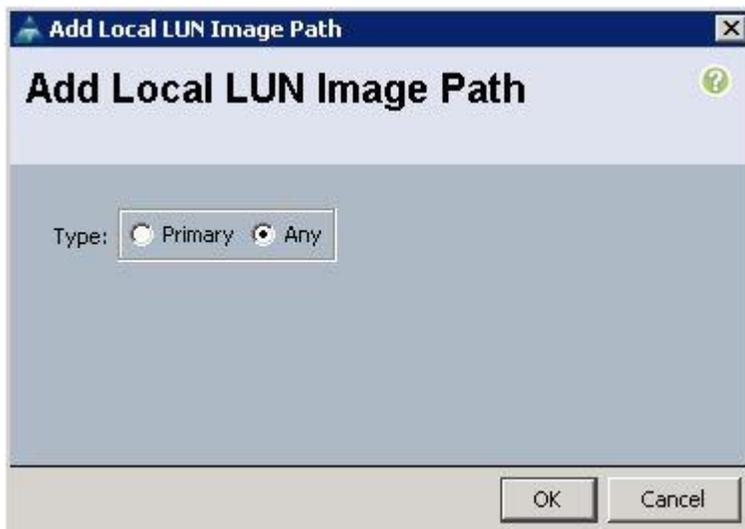
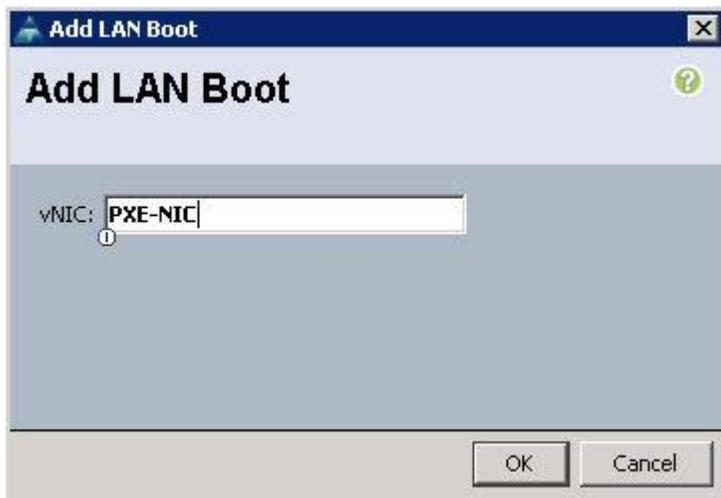
1. Under Server → Policies → root → Boot Policies → right-click and select Create Boot Policy.



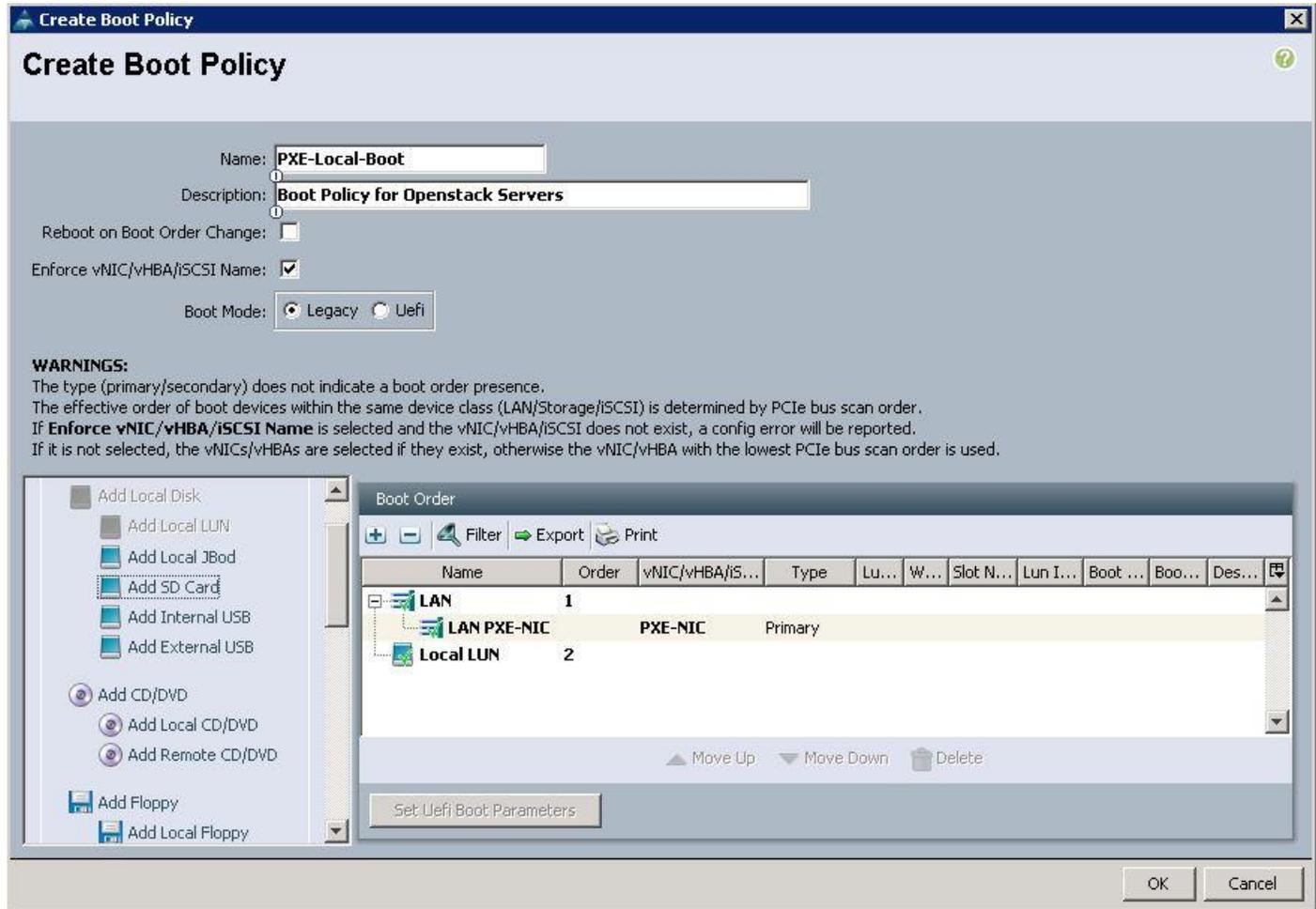
- a. Specify the name and description. Select the First boot order as LAN boot and specify the actual vNIC name of the PXE network (PXE-NIC). Then select the second boot order and click Add Local LUN.



b. Specify the vNIC Name as PXE-NIC.



c. Make sure the First boot order is PXE NIC and second boot order is Local LUN and click OK.

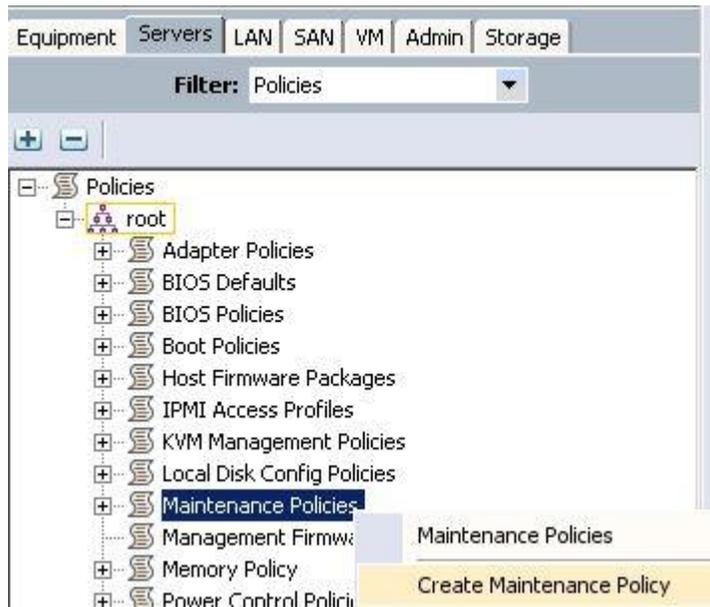


## Create a Maintenance Policy

A maintenance policy determines a pre-defined action to take when there is a disruptive change made to the service profile associated with a server. When creating a maintenance policy you have to select a reboot policy which defines when the server can reboot once the changes are applied.

To configure the Maintenance policy from the Cisco UCS Manager, complete the following steps:

1. Under Server → Policies → root → Maintenance Policies → right-click and select Create Maintenance Policy.



## Create an IPMI Access Policy

This policy allows you to determine whether IPMI commands can be sent directly to the server, using the IP address (KVM IP address).

To configure the IPMI Access profiles from the Cisco UCS Manager, complete the following steps:

1. Under Server → Policies → root → IPMI Access profiles → right-click and select Create IPMI Access Profile.



a. Specify the name and click IPMI over LAN as Enabled and click “+”.



b. Specify the username and password. Choose Admin for the Role and click OK.

**Create IPMI User**

Name:

Password:

Confirm Password:

Role:  Read Only  Admin

OK Cancel

c. Click OK to create the IPMI access profile.

**Create IPMI Access Profile**

Name:

Description:

IPMI Over LAN:  Disable  Enable

**IPMI Users**

Filter Export Print

Name	Role
admin	Admin

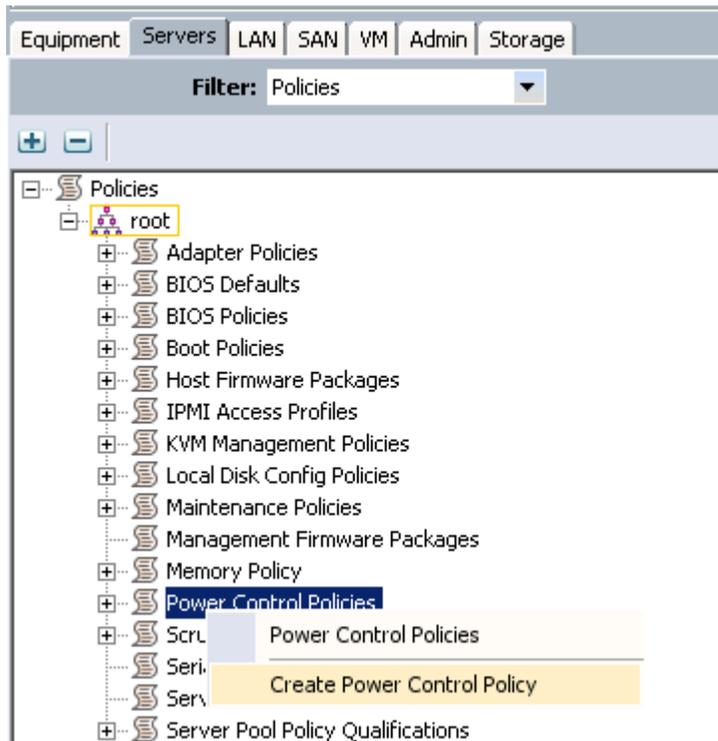
OK Cancel

## Create a Power Policy

Cisco UCS uses the priority set in the power control policy, along with the blade type and configuration, to calculate the initial power allocation for each blade within a chassis. During normal operation, the active blades within a chassis can borrow power from idle blades within the same chassis. If all blades are active and reach the power cap, service profiles with higher priority power control policies take precedence over service profiles with lower priority power control policies.

To configure the Power Control policy from the UCS Manager, complete the following steps:

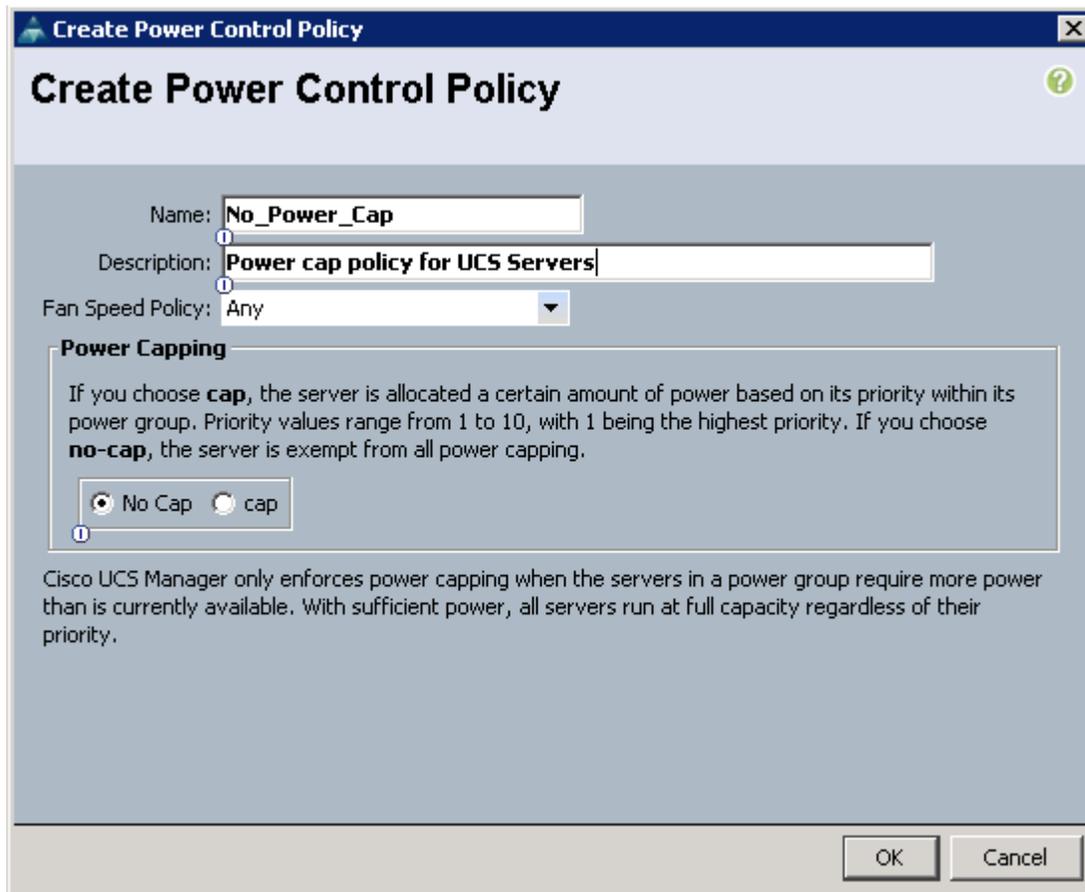
1. Under Server → Policies → root → Power Control Policies → right-click and select Create Power Control Policy.



- a. Specify the name and description. Choose Power Capping as No Cap.

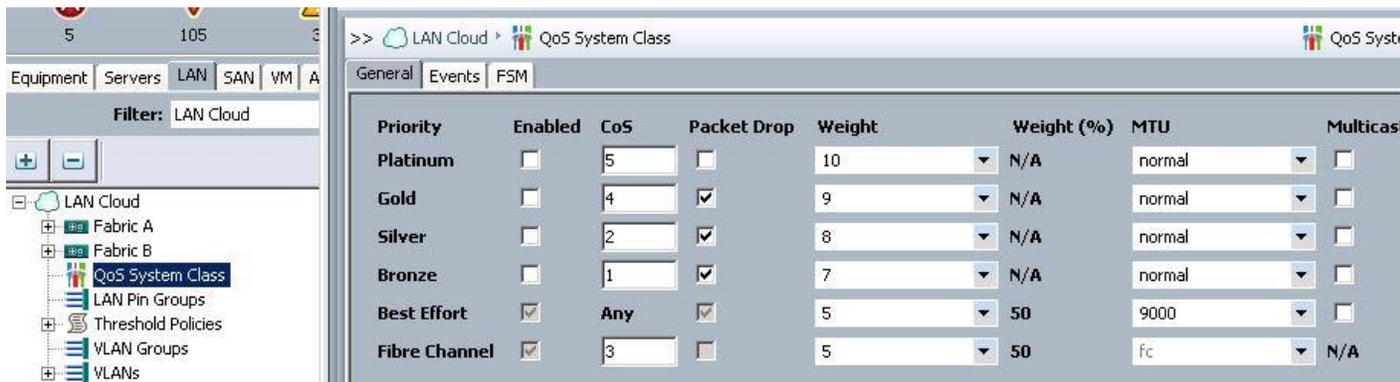


No Cap keeps the server runs at full capacity regardless of the power requirements of the other servers in its power group. Setting the priority to no-cap prevents Cisco UCS from leveraging unused power from that particular blade server. The server is allocated the maximum amount of power that that blade can reach.



### Create a QOS system class

Create a QOS system class as shown below:



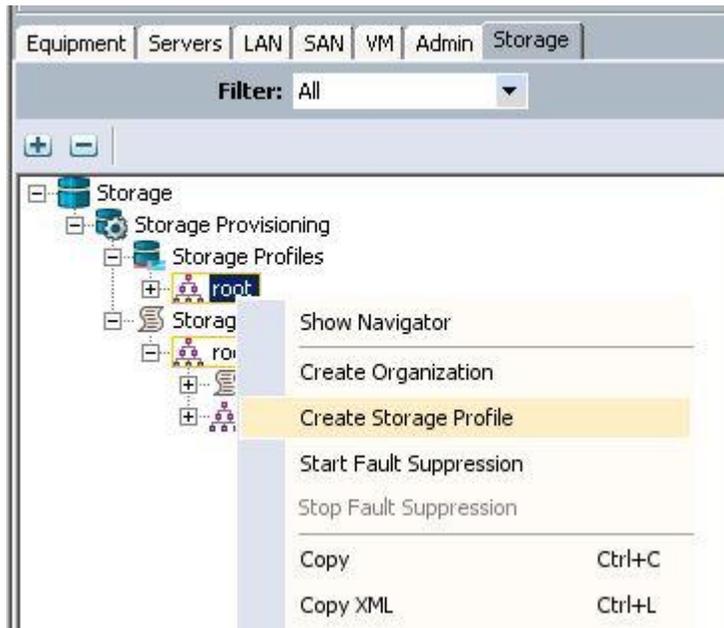
Select the Best Effort class as MTU 9000, which will be leveraged in vNIC templates for storage public and storage management vNIC's.

### Create Storage Profiles for the Controller and Compute Blades

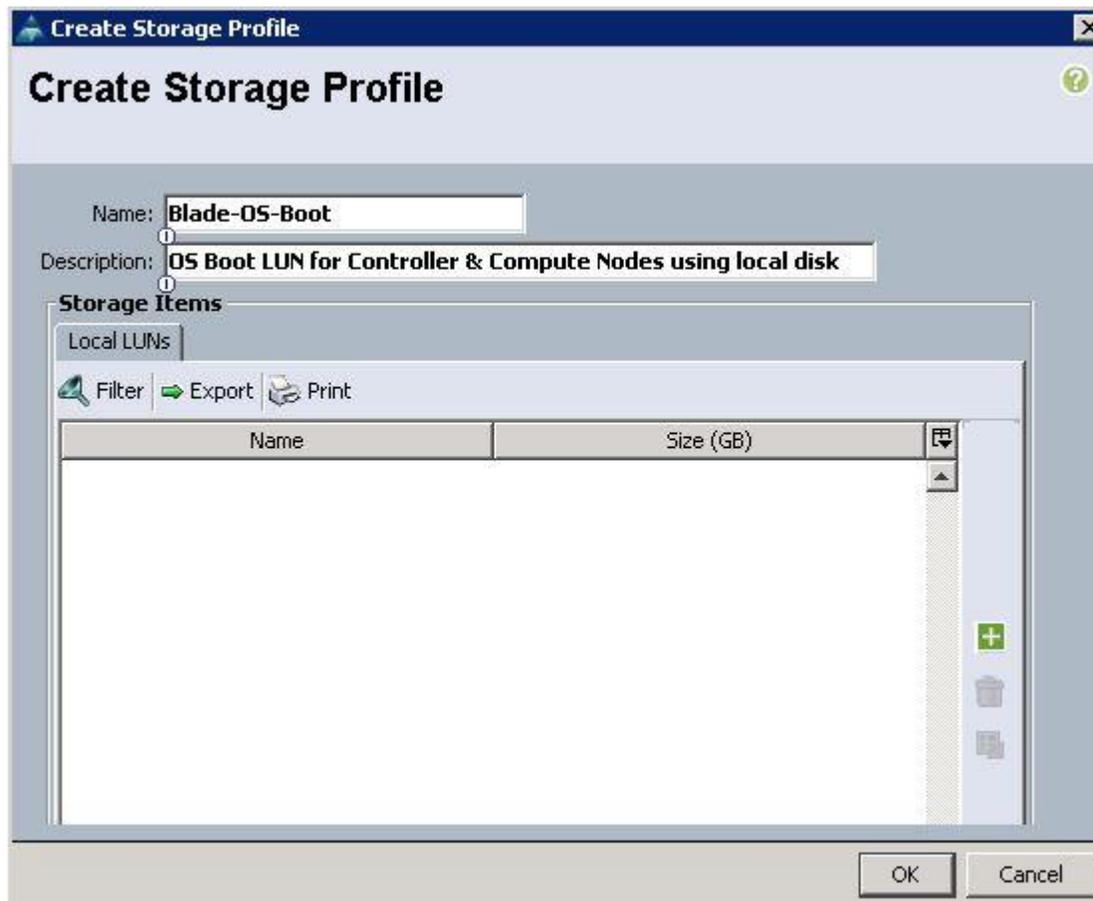
To allow flexibility in defining the number of storage disks, roles and usage of these disks, and other storage parameters, you can create and use storage profiles. LUNs configured in a storage profile can be used as boot LUNs or data LUNs, and can be dedicated to a specific server. You can also specify a local LUN as a boot device. However, LUN resizing is not supported.

To configure Storage profiles from the Cisco UCS Manager, complete the following steps:

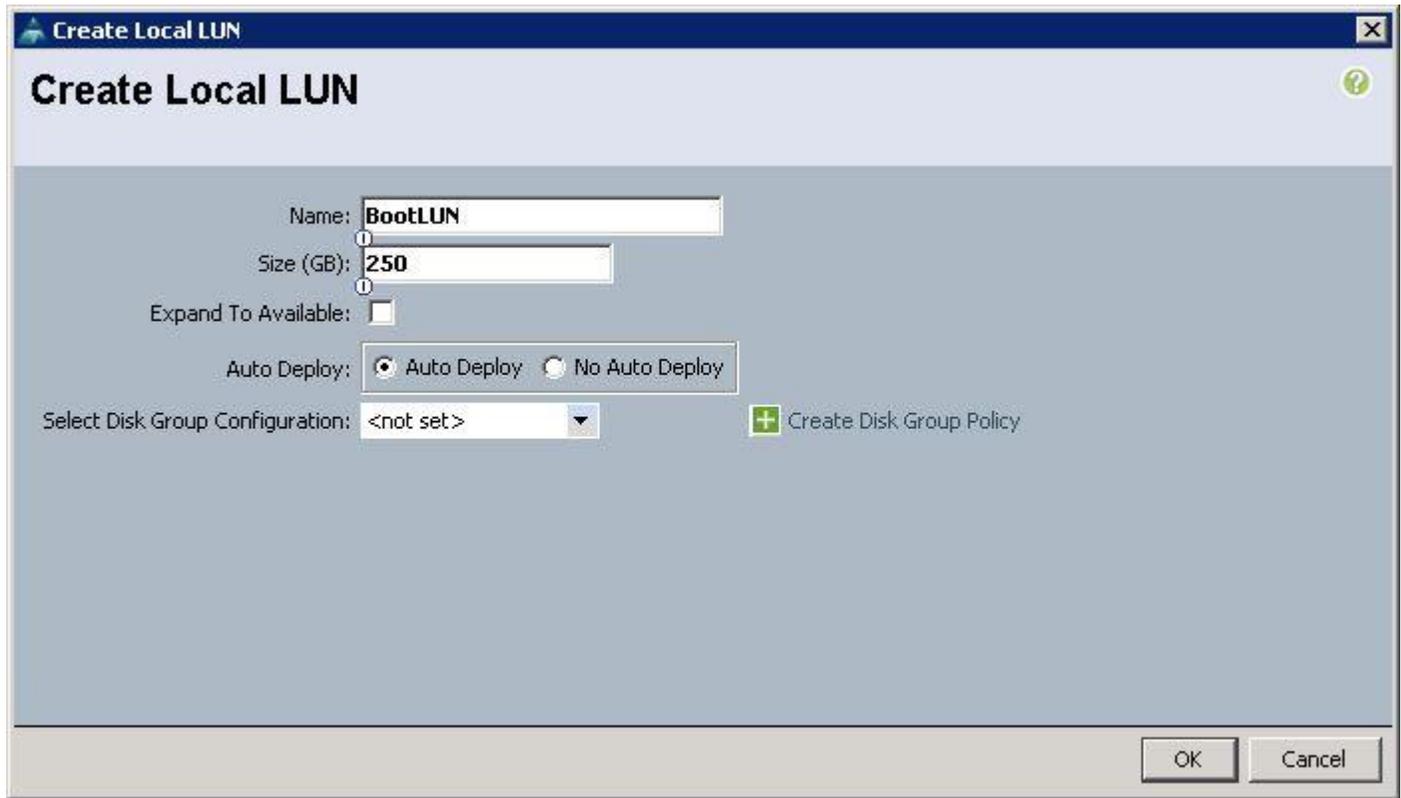
1. Under Storage → Storage Provisioning → Storage Profiles → root → right-click and select Create Storage Profile.



- a. Specify the name and click “+”.



- b. Specify the Local LUN name and size as 250 in GB and click Auto Deploy.
- c. To configure RAID levels and configure the number of disks for the disk group, select Create Disk Group Policy.



- d. Specify the name and choose RAID level as RAID 1 Mirrored. RAID1 is recommended for the Local boot LUNs.
- e. Select Disk group Configuration (Manual) and click "+". **Keep** the Virtual Drive configuration with the default values.

**Create Disk Group Policy**

Name:

Description:

RAID Level:

Disk Group Configuration (Automatic)  Disk Group Configuration (Manual)

**Disk Group Configuration (Manual)**

Filter Export Print

Slot Number	Role	Span ID
-------------	------	---------

**Virtual Drive Configuration**

Strip Size (KB):

Access Policy:  Platform Default  Read Write  Read Only  Blocked

Read Policy:  Platform Default  Read Ahead  Normal

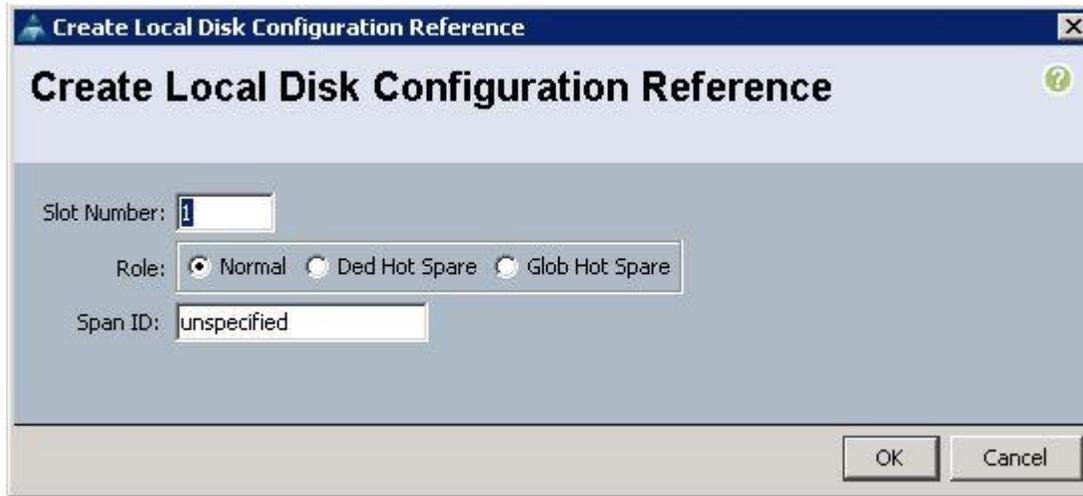
Write Cache Policy:  Platform Default  Write Through  Write Back Good Bbu  Always Write Back

IO Policy:  Platform Default  Direct  Cached

Drive Cache:  Platform Default  No Change  Enable  Disable

OK Cancel

- f. Specify Disk Slot Number as 1 and Role as Normal.



g. Create another Local Disk configuration with the Slot number as 2 and click OK.



In this solution, we used Local Disk 1 and Disk 2 as the boot LUNs with RAID 1 mirror configuration.

---

**Create Disk Group Policy**

Name:

Description:

RAID Level:

Disk Group Configuration (Automatic)
  Disk Group Configuration (Manual)

**Disk Group Configuration (Manual)**

Filter Export Print

Slot Number	Role	Span ID
1	Normal	Unspecified
2	Normal	Unspecified

**Virtual Drive Configuration**

Strip Size (KB):

Access Policy:  Platform Default  Read Write  Read Only  Blocked

Read Policy:  Platform Default  Read Ahead  Normal

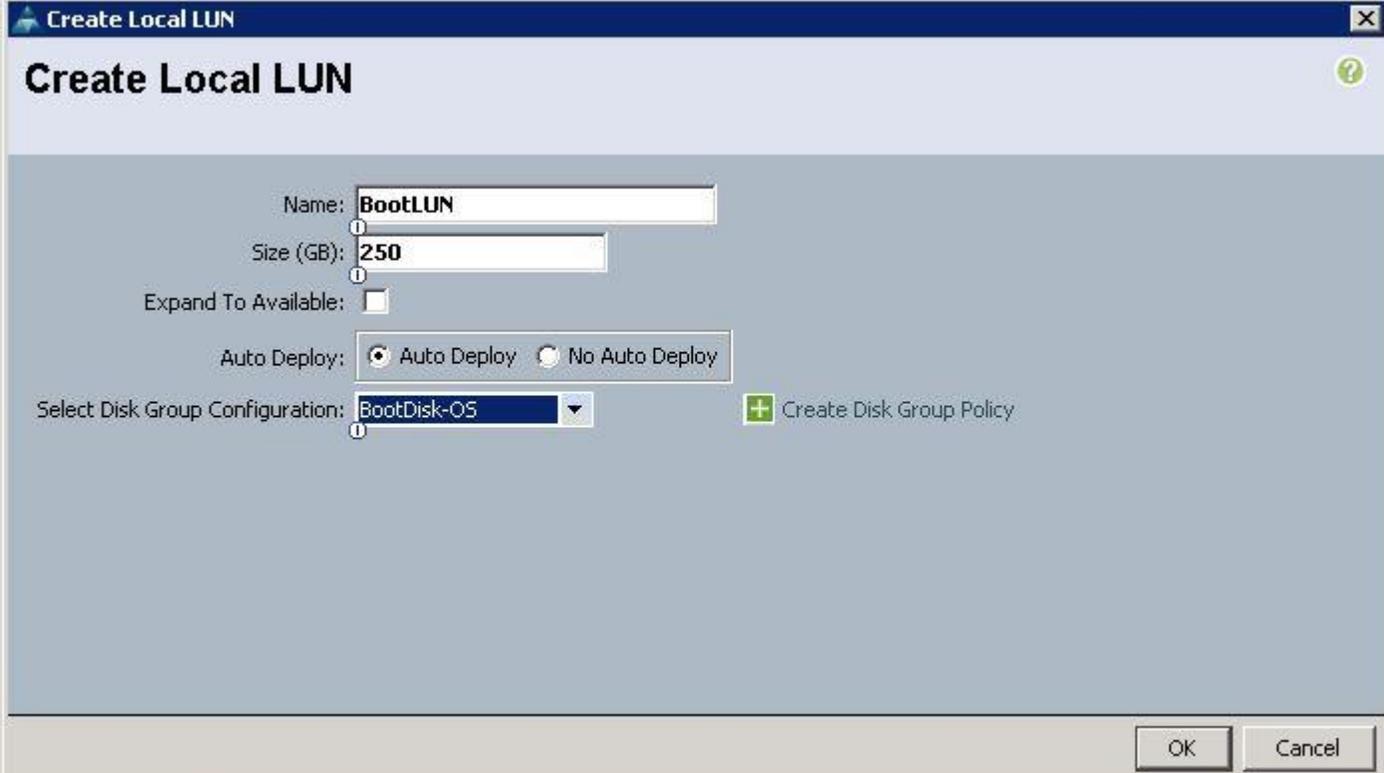
Write Cache Policy:  Platform Default  Write Through  Write Back Good Bbu  Always Write Back

IO Policy:  Platform Default  Direct  Cached

Drive Cache:  Platform Default  No Change  Enable  Disable

OK Cancel

h. Choose the Disk group policy Boot Disk-OS for the Local Boot LUN.



**Create Local LUN**

Name:

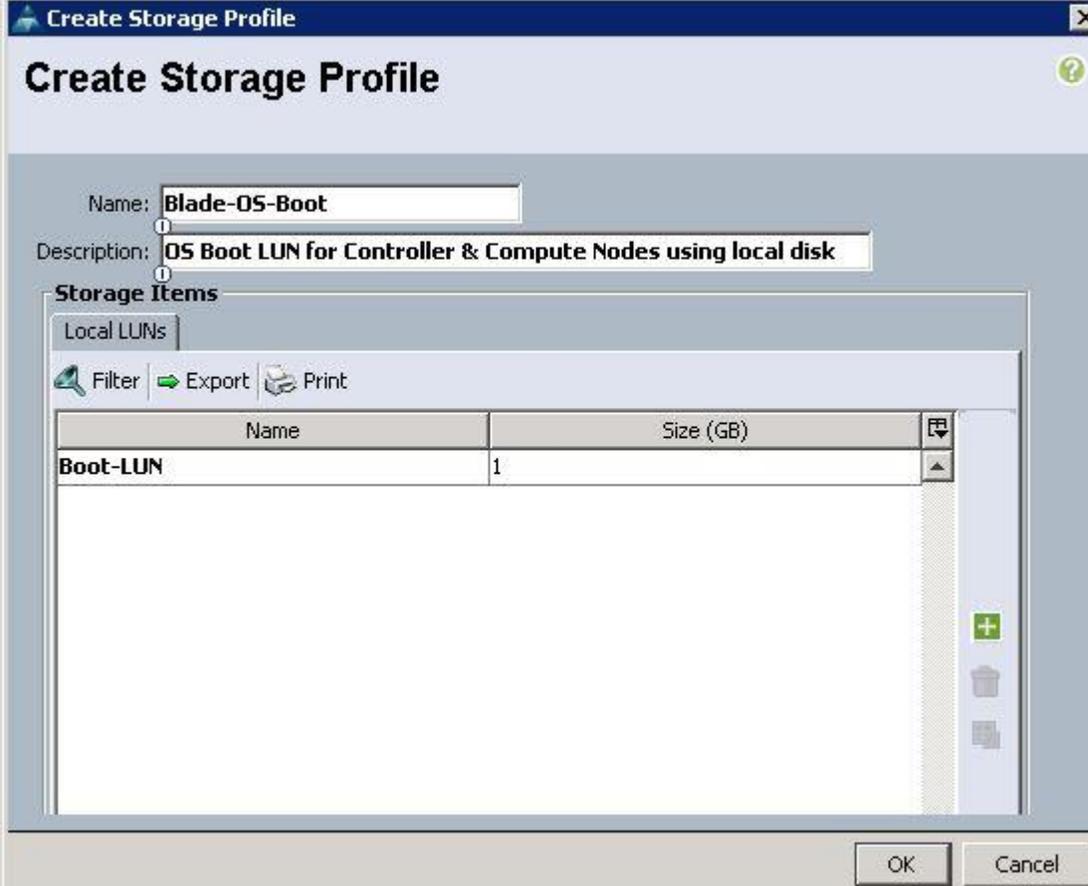
Size (GB):

Expand To Available:

Auto Deploy:  Auto Deploy  No Auto Deploy

Select Disk Group Configuration:  [+ Create Disk Group Policy](#)

- i. Click OK to confirm the Storage profile creation.



**Create Storage Profile**

Name:

Description:

**Storage Items**

Local LUNs

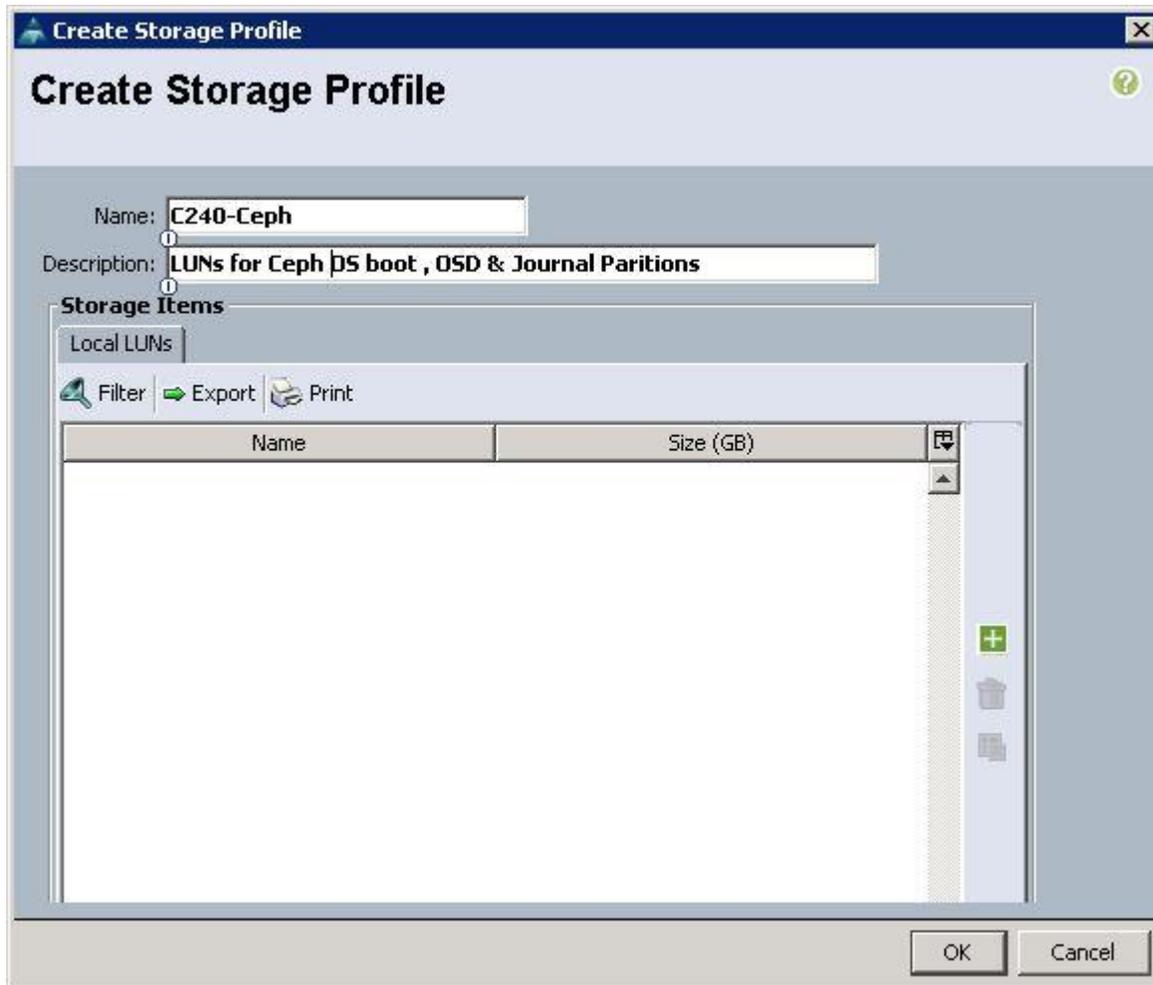
[Filter](#) [Export](#) [Print](#)

Name	Size (GB)
Boot-LUN	1

## Create Storage Profiles for Cisco UCS C240 M4 Server Blades

To configure the Storage profiles from the UCS Manager, complete the following steps:

1. Under Storage → Storage Provisioning → Storage Profiles → root → right-click and select Create Storage Profile.
  - a. Specify the Storage profile name as C240-Ceph for the Ceph Storage Servers. Click “+”.



- b. Specify the LUN name and size in GB. For the Disk group policy creation, select Disk Group Configuration for Ceph nodes as Ceph-OS-Boot similar to “BootDisk-OS” disk group policy as above.

**Create Disk Group Policy**

Name:

Description:

RAID Level:

Disk Group Configuration (Automatic)  Disk Group Configuration (Manual)

**Disk Group Configuration (Manual)**

Filter Export Print

Slot Number	Role	Span ID
1	Normal	Unspecified
2	Normal	Unspecified

**Virtual Drive Configuration**

Strip Size (KB):

Access Policy:  Platform Default  Read Write  Read Only  Blocked

Read Policy:  Platform Default  Read Ahead  Normal

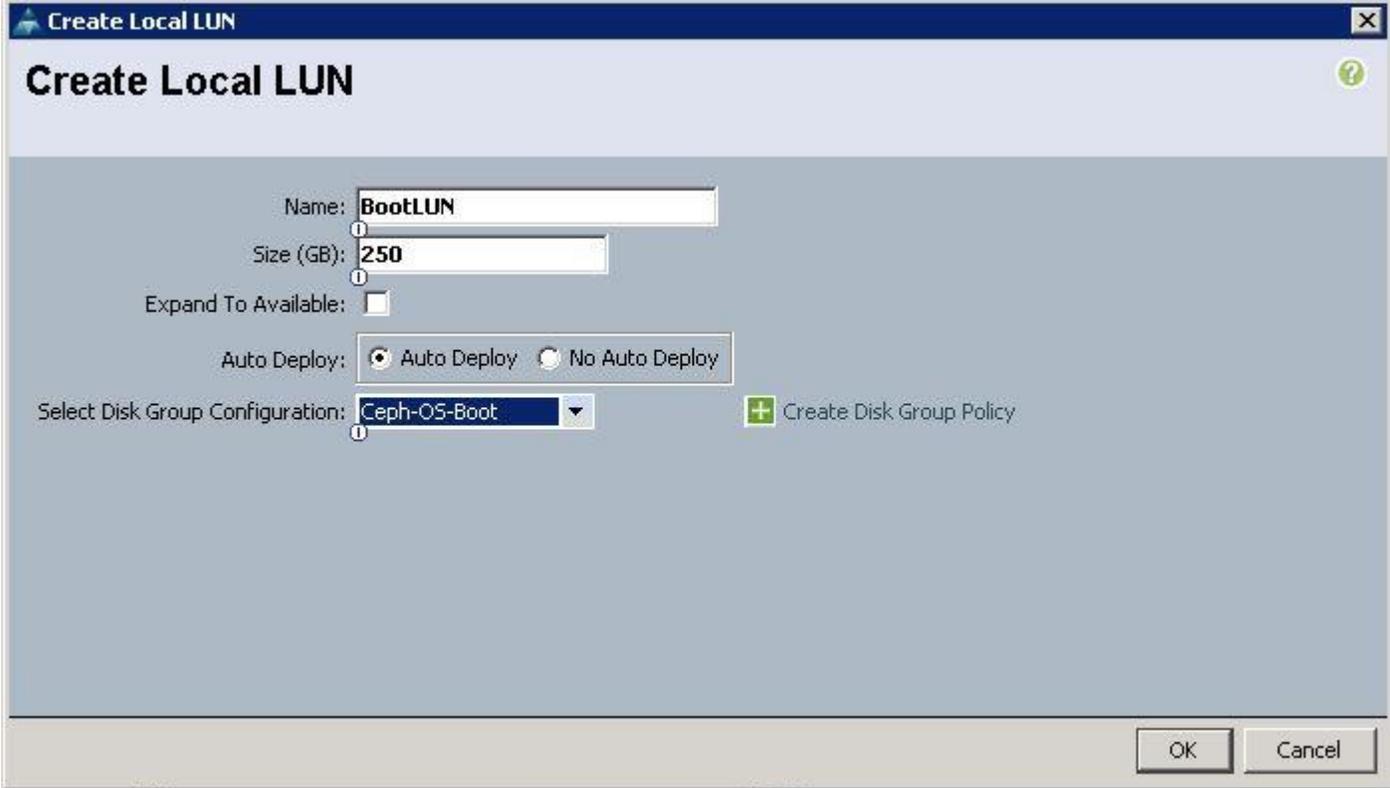
Write Cache Policy:  Platform Default  Write Through  Write Back Good Bbu  Always Write Back

IO Policy:  Platform Default  Direct  Cached

Drive Cache:  Platform Default  No Change  Enable  Disable

OK Cancel

- c. After successful creation of Disk Group Policy, choose Disk Group Configuration as Ceph-OS-Boot and click OK.

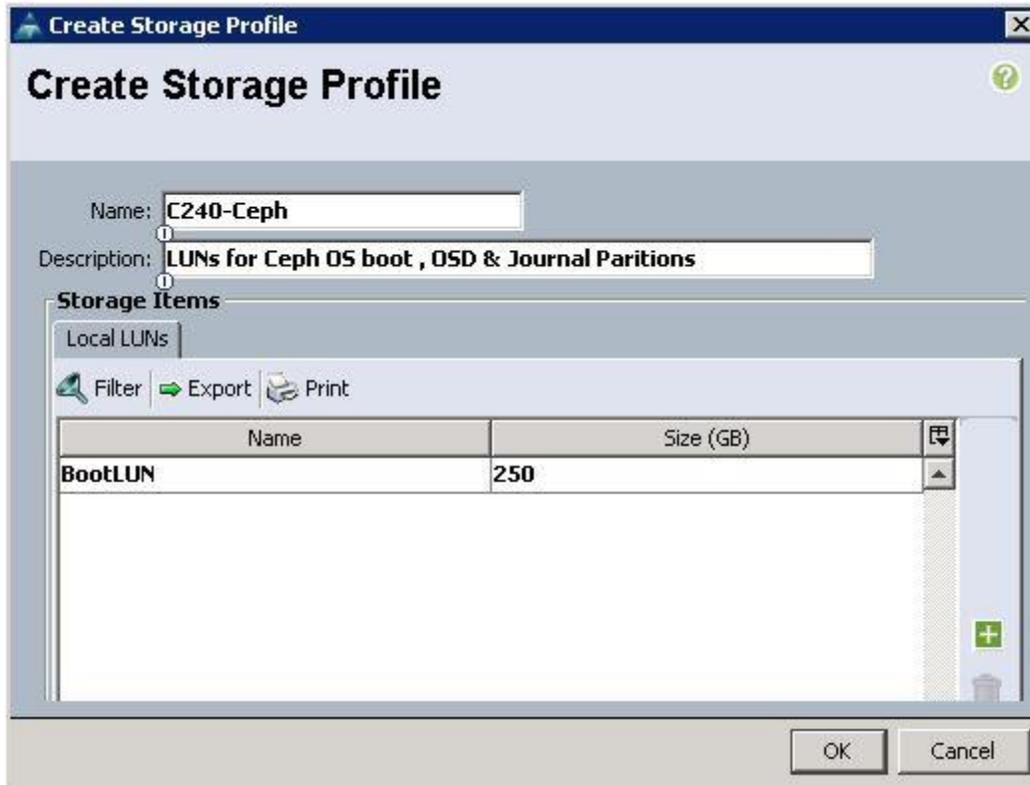


The screenshot shows a dialog box titled "Create Local LUN" with a blue header bar. The main area is light blue and contains the following fields and controls:

- Name:** A text input field containing "BootLUN".
- Size (GB):** A text input field containing "250".
- Expand To Available:** An unchecked checkbox.
- Auto Deploy:** Two radio buttons, with "Auto Deploy" selected.
- Select Disk Group Configuration:** A dropdown menu showing "Ceph-OS-Boot".
- Create Disk Group Policy:** A green plus icon followed by the text "Create Disk Group Policy".

At the bottom right, there are two buttons: "OK" and "Cancel".

- d. Click OK to complete the Storage Profile creation for the Ceph Nodes.

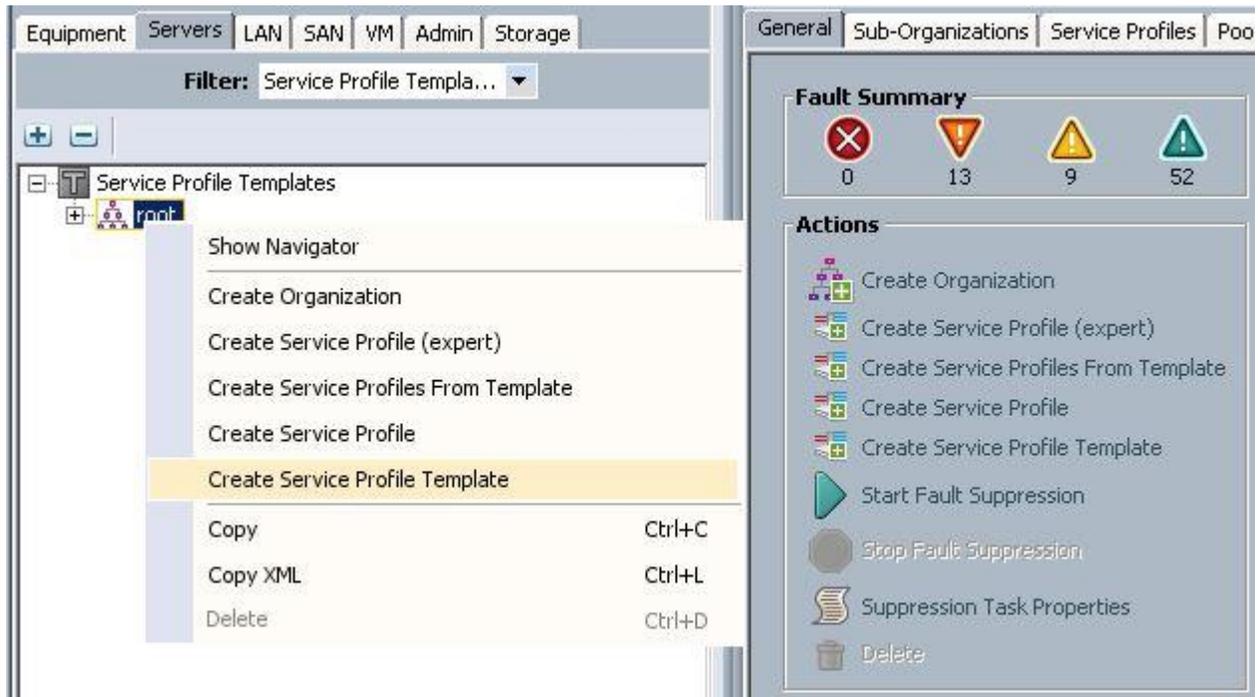


For the Cisco UCS C240 M4 servers, the LUN creation for Ceph OSD disks (6TB SAS) and Ceph Journal disks (400GB SSDs) still remains on the Ceph Storage profile. Due to the Cisco UCS Manager limitations, we have to create OSD LUNs and Journal LUNs after the Cisco UCS C240 M4 server has been successfully associated with the Ceph Storage Service profiles.

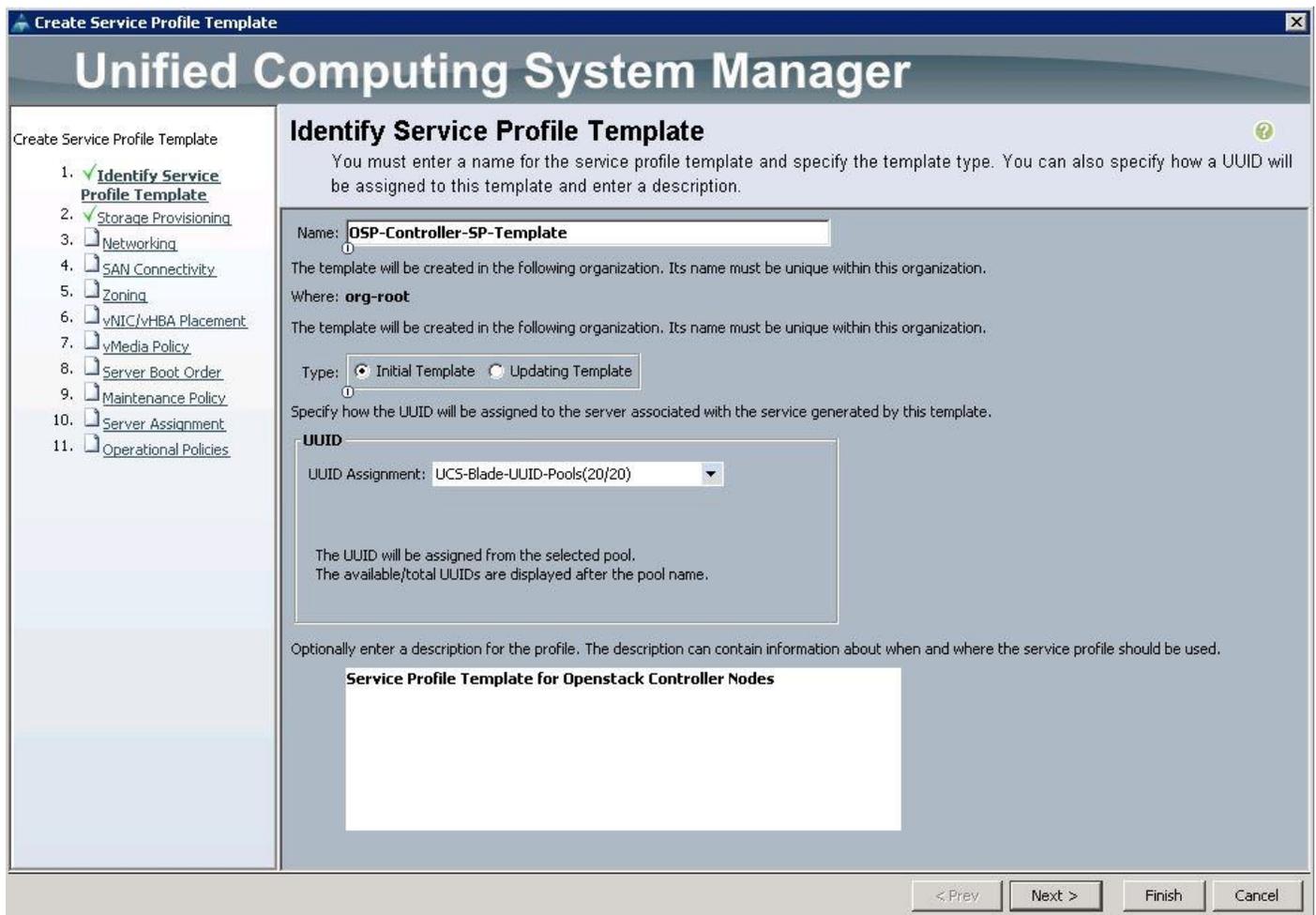
## Create Service Profile Templates for Controller Nodes

To configure the Service Profile Templates for the Controller Nodes, complete the following steps:

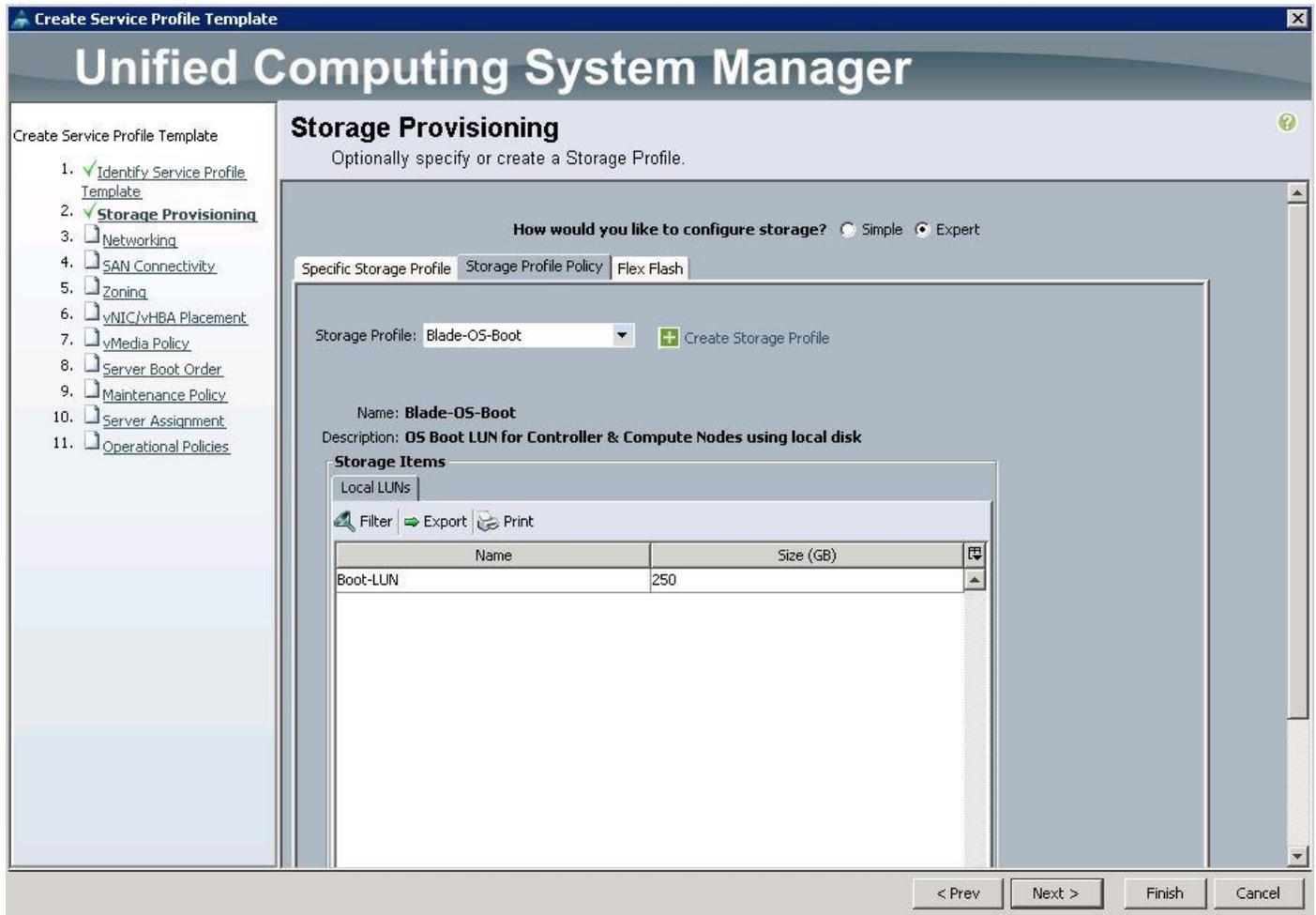
1. Under Servers → Service Profile Templates → root → right-click and select Create Service Profile Template.



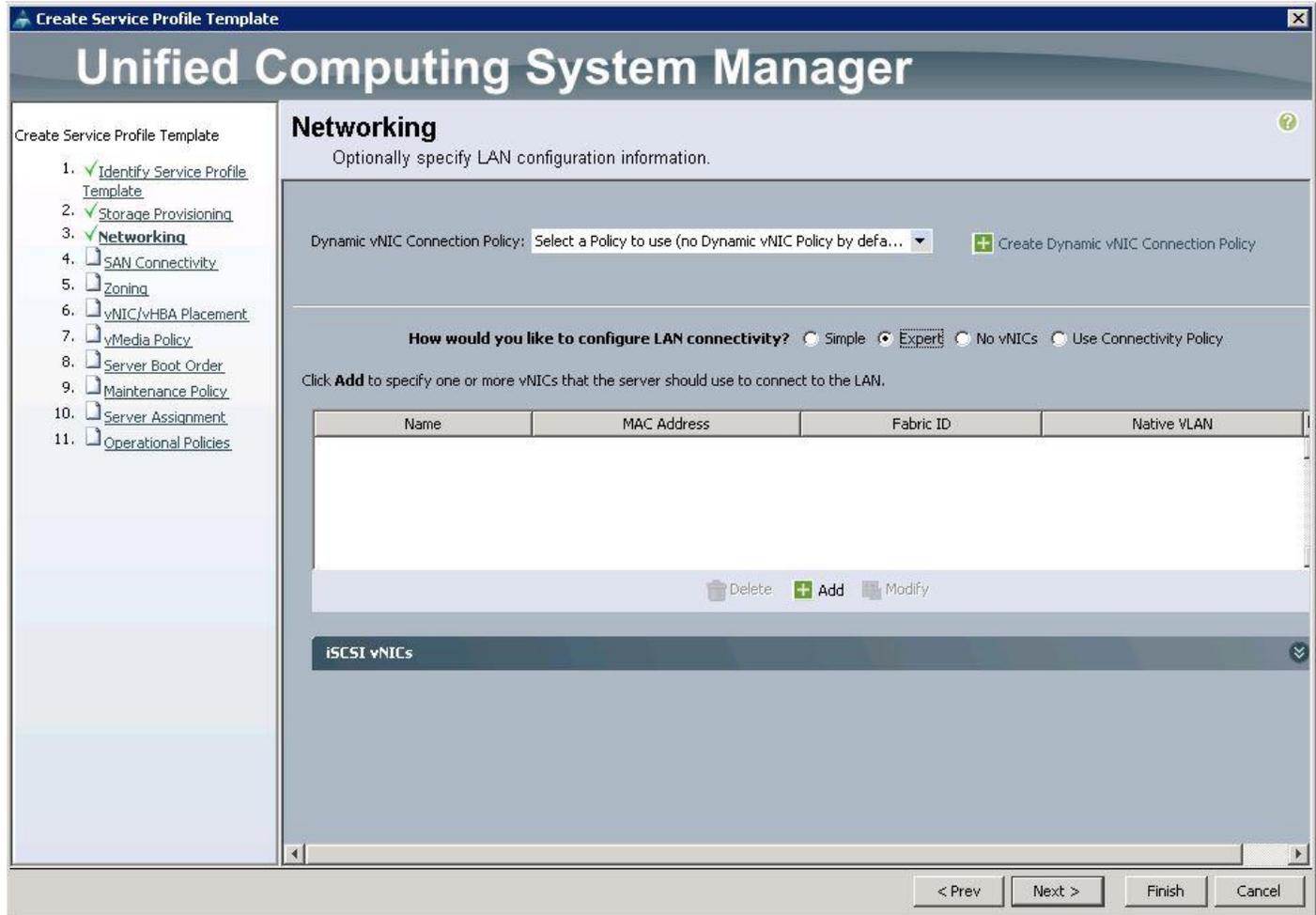
- a. Specify the Service profile template name for the Controller node as OSP-Controller-SP-Template. Choose the UUID pools previously created from the drop-down list and click Next.



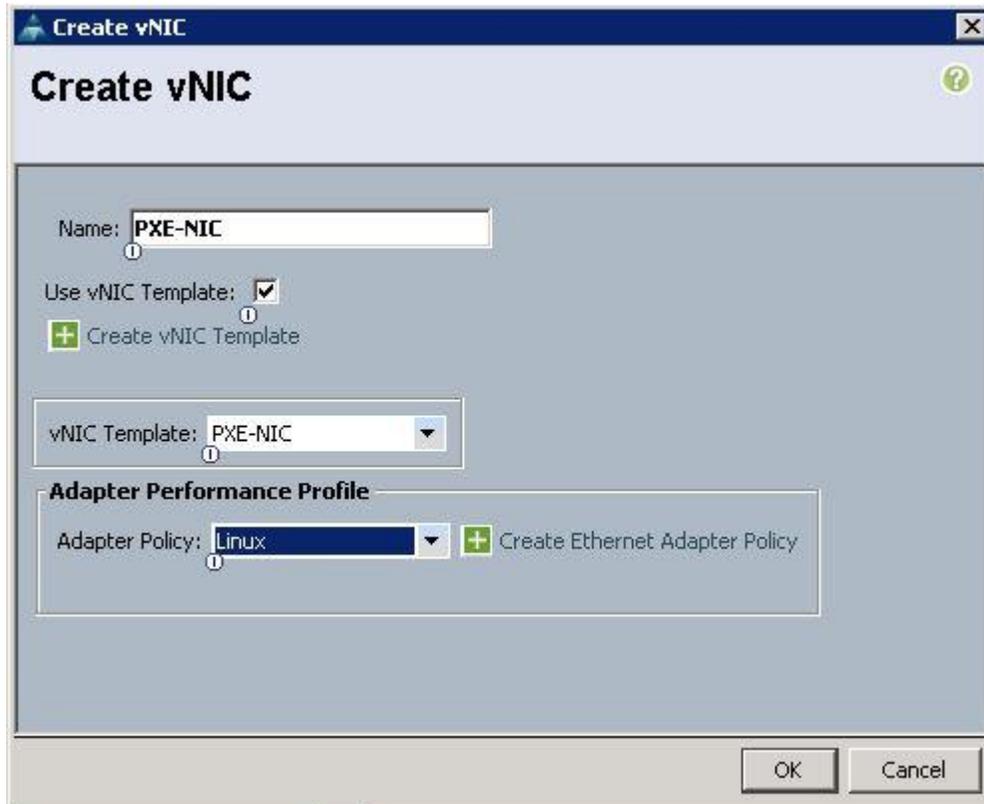
- b. For Storage Provisioning, choose Expert and click Storage profile Policy and choose the Storage profile Blade-OS-boot previously created from the drop-down list and click Next.



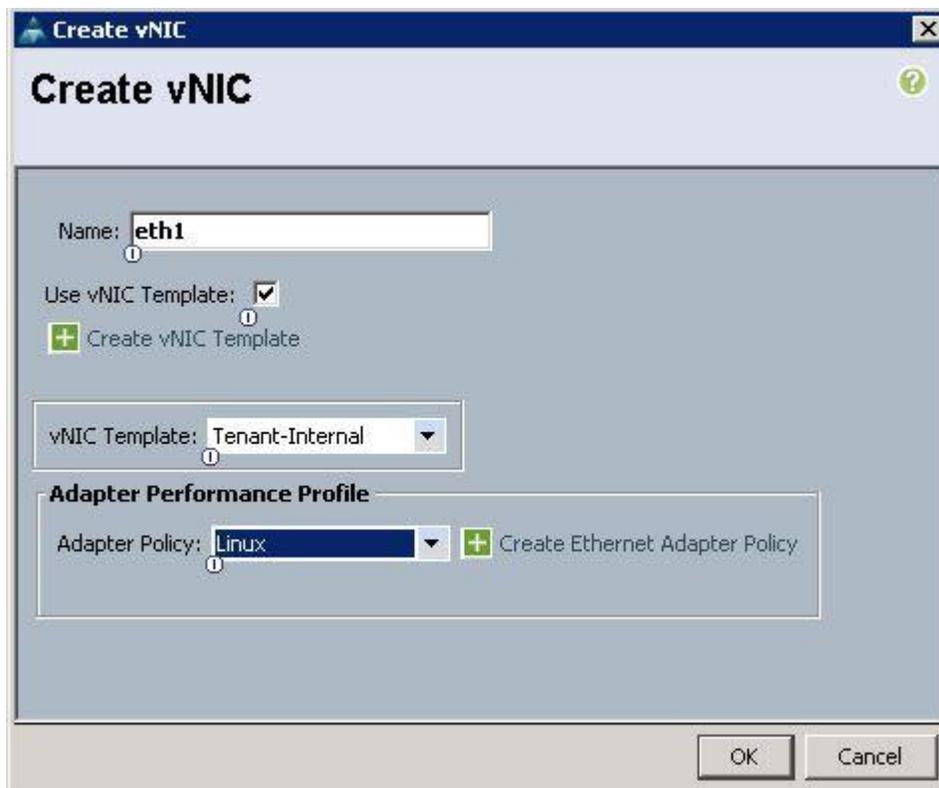
- c. For Networking, choose Expert and click “+”.



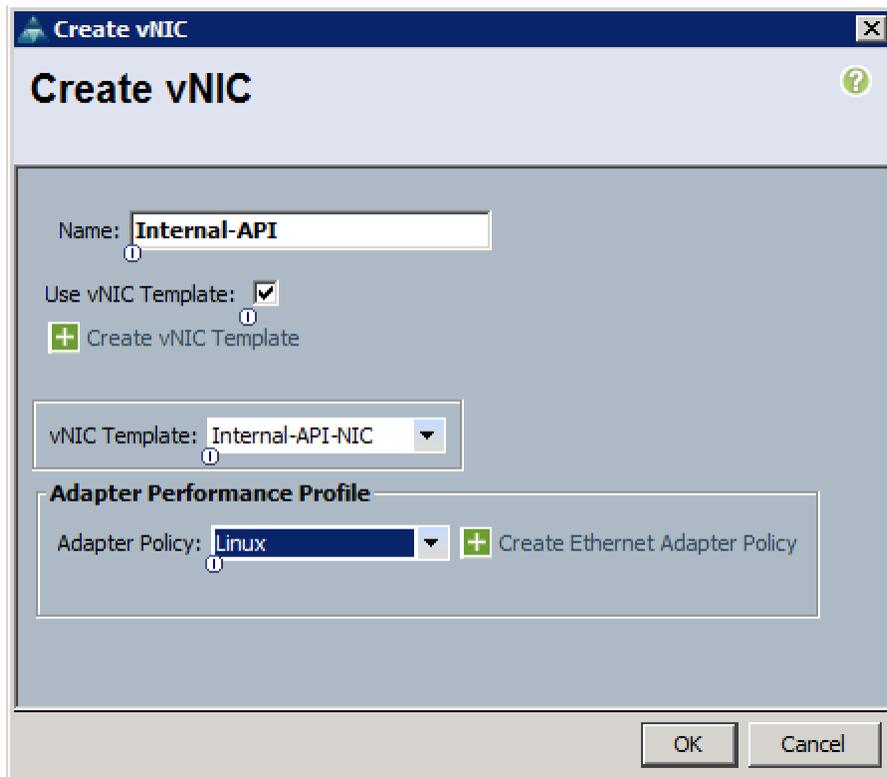
- d. Create the vNIC interface for PXE or Provisioning network as PXE-NIC and click the check box Use vNIC template.
- e. Under vNIC template, choose the PXE-NIC template previously created from the drop-down list and choose Linux for the Adapter Policy.



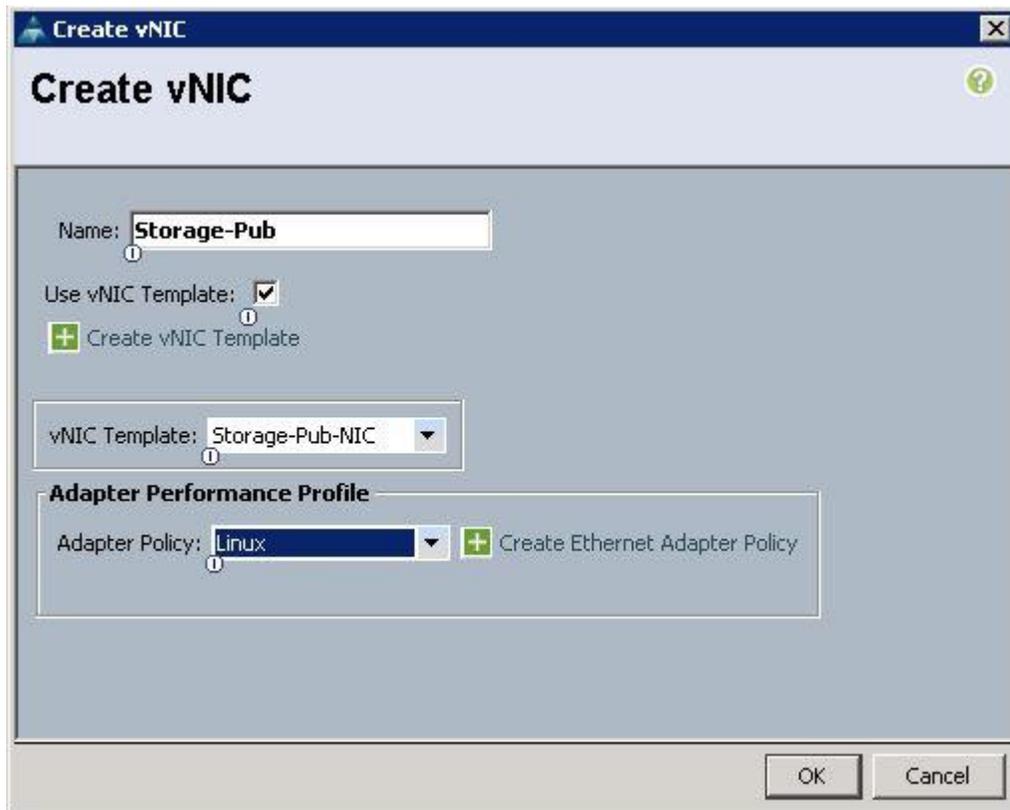
- f. Create the vNIC interface for Tenant Internal Network as eth1 and then under vNIC template, choose the “Tenant-Internal” template we created before from the drop-down list and choose Adapter Policy as “Linux”.



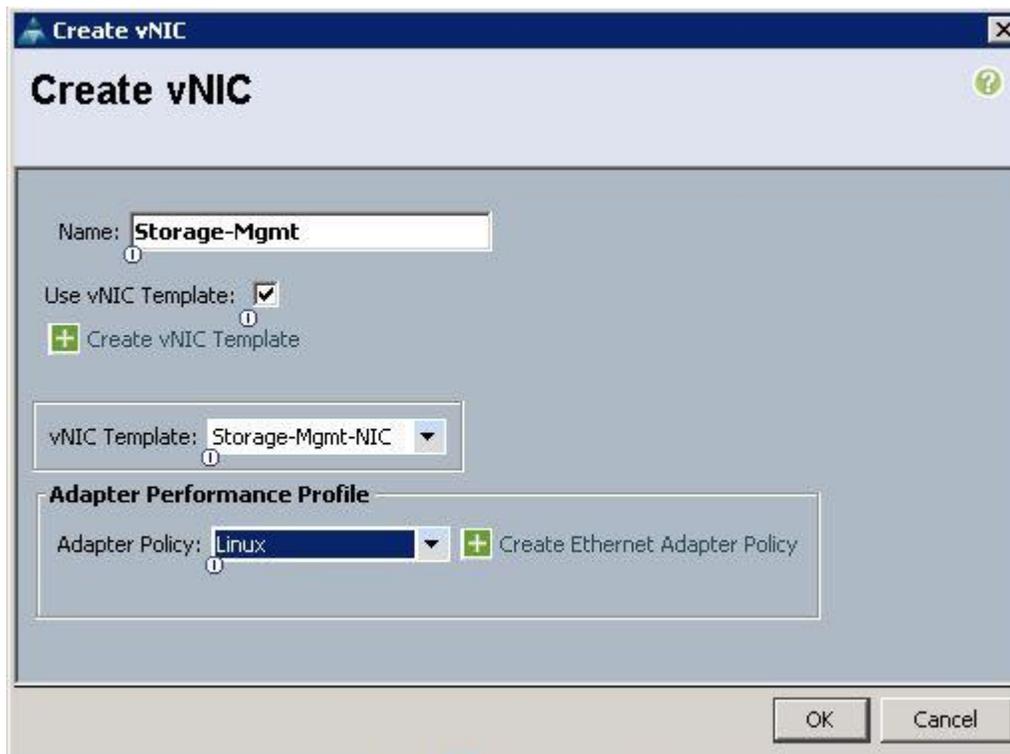
- g. Create the vNIC interface for Internal API network as Internal-API and click the check box for Use vNIC template.
- h. Under vNIC template, choose the Internal-API-NIC template previously created from the drop-down list and choose Linux for the Adapter Policy.



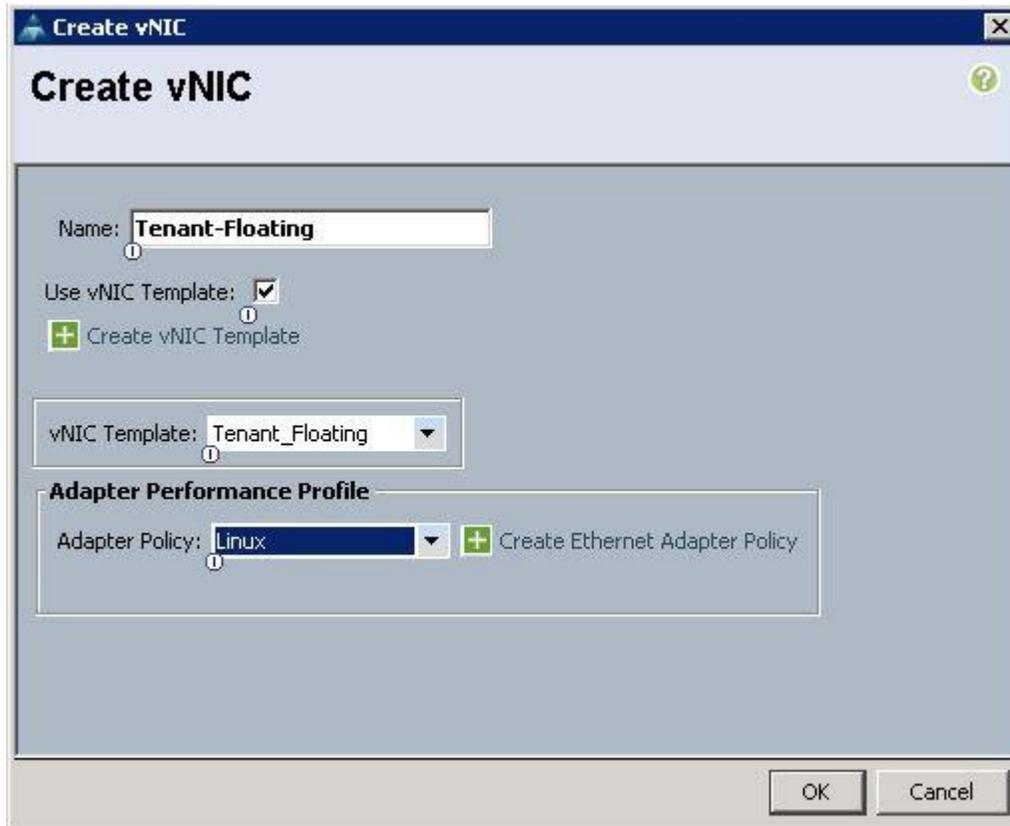
- i. Create the vNIC interface for Storage Public Network as Storage-Pub and click the check box for Use vNIC template.
- j. Under vNIC template, choose the Storage-Pub-NIC template previously created from the drop-down list and choose Linux for the Adapter Policy.



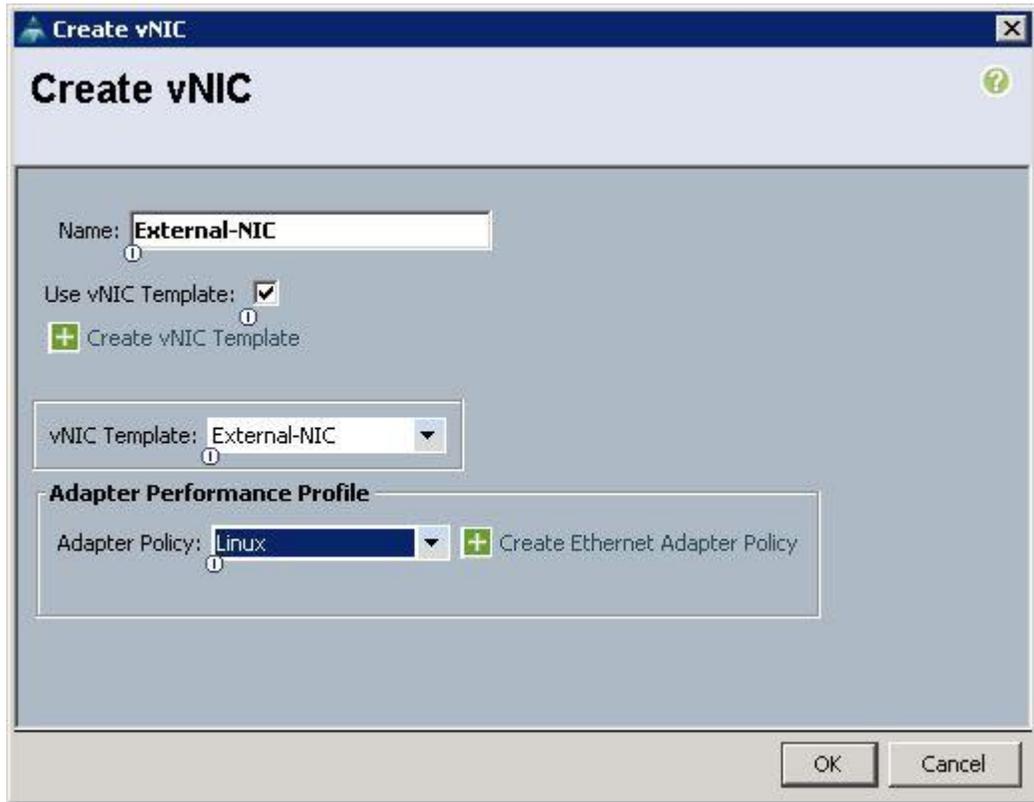
- k. Create the vNIC interface for Storage Mgmt Cluster Network as Storage-Mgmt and click the check box for Use vNIC template.
- l. Under vNIC template, choose the Storage-Mgmt-NIC template previously created from the drop-down list and choose Linux for the Adapter Policy.



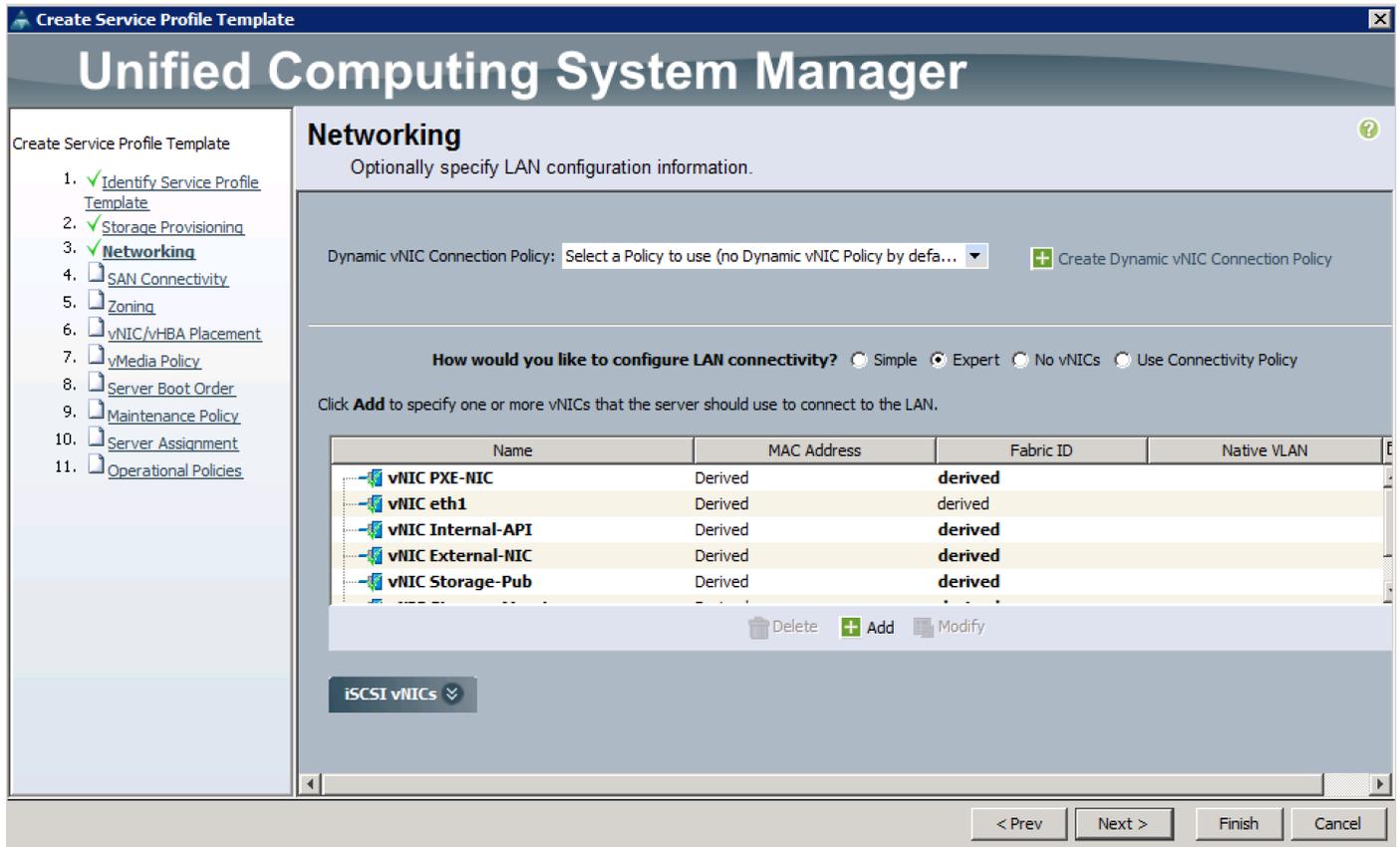
- m. Create the vNIC interface for Floating Network as Tenant-Floating and click the check box the Use vNIC template.
- n. Under the vNIC template, choose the Tenant-Floating template previously created from the drop-down list and choose Linux for the Adapter Policy.



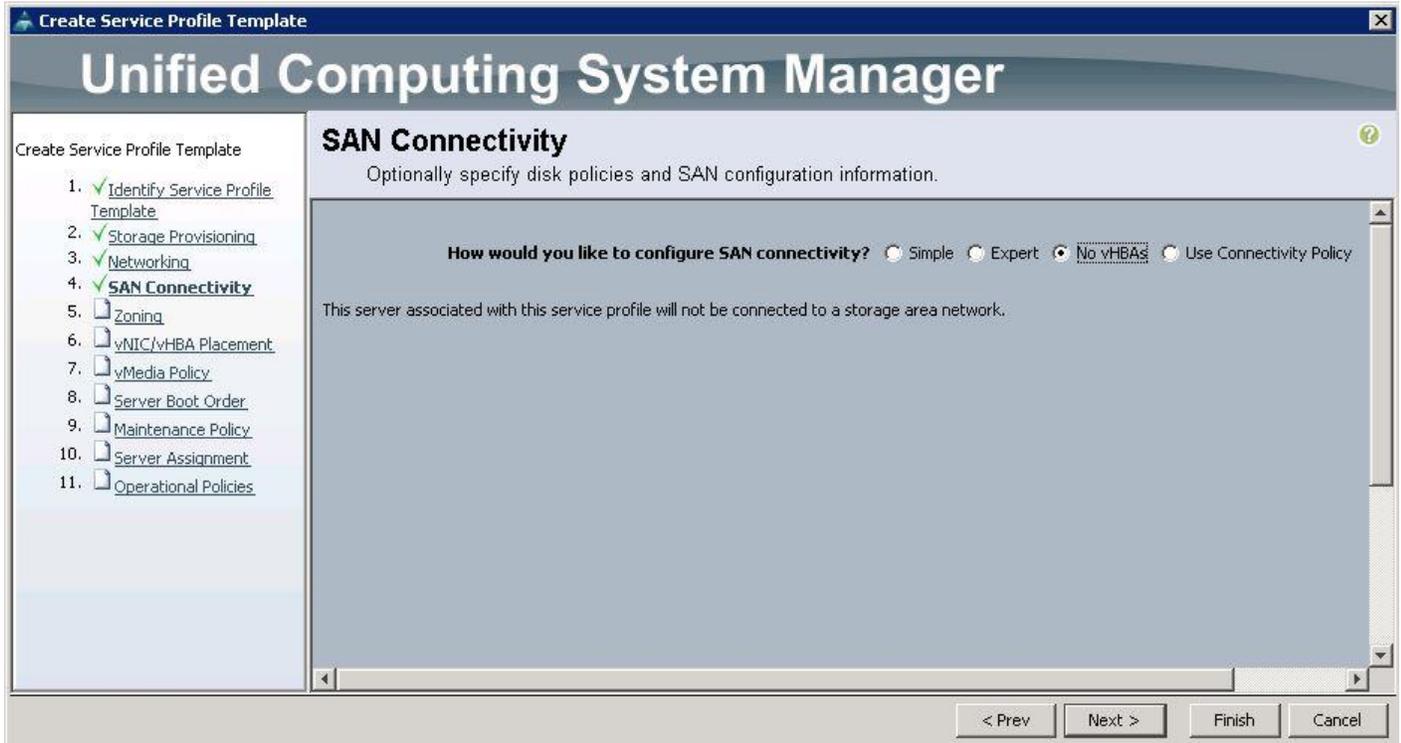
- o. Create the vNIC interface for External Network as External-NIC and click the check box the Use vNIC template.
- p. Under the vNIC template, choose the External-NIC template previously created from the drop-down list and choose Linux for the Adapter Policy.



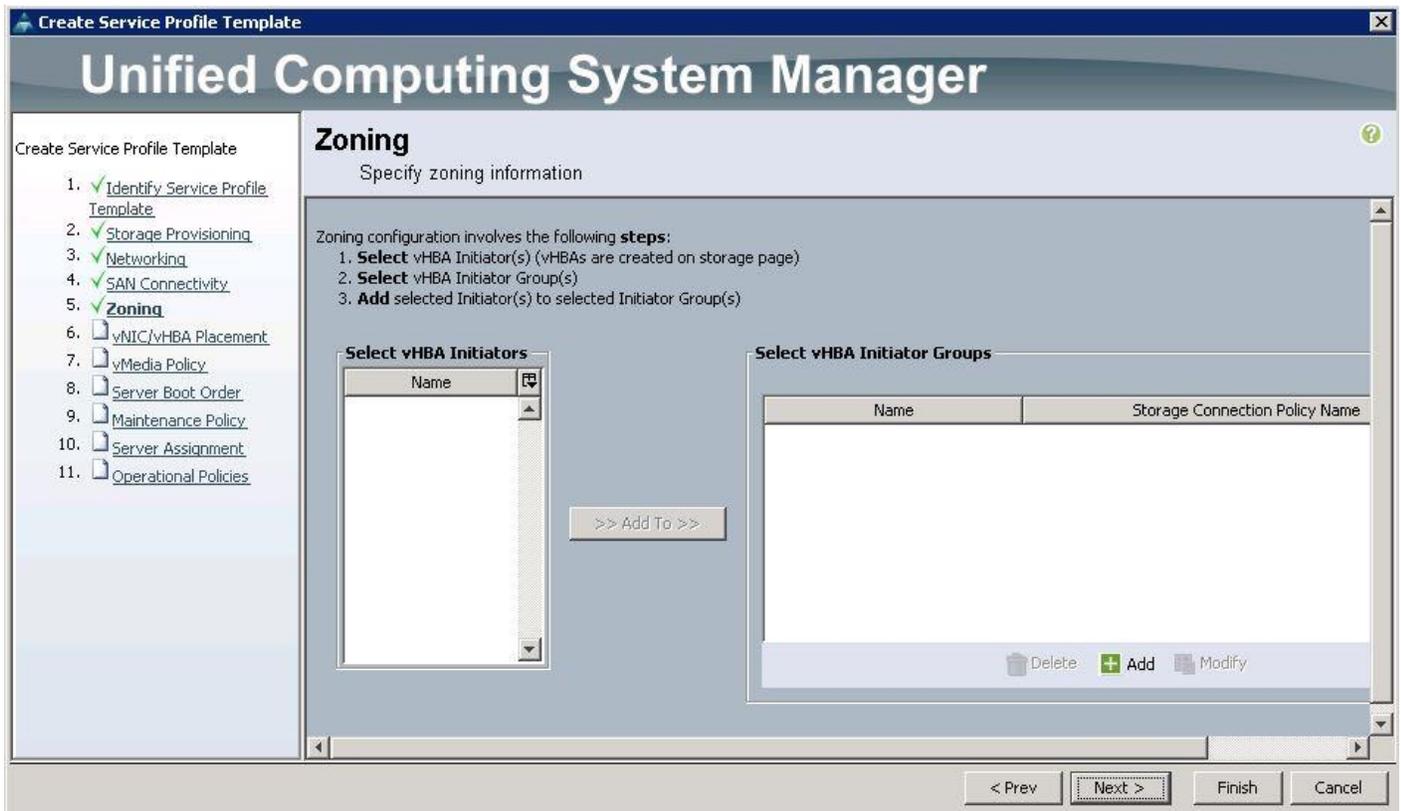
q. After a successful vNIC creation, click Next.



r. Under the SAN connectivity, choose No VHBAs and click Next.



s. Under Zoning, click Next.



t. Under vNIC/vHBA Placement, choose the vNICs PCI order as shown below and click Next.

Create Service Profile Template
✕

Unified Computing System Manager

Create Service Profile Template

1.  Identify Service Profile Template
2.  Storage Provisioning
3.  Networking
4.  SAN Connectivity
5.  Zoning
6.  **vNIC/vHBA Placement**
7.  vMedia Policy
8.  Server Boot Order
9.  Maintenance Policy
10.  Server Assignment
11.  Operational Policies

### vNIC/vHBA Placement ?

Specify how vNICs and vHBAs are placed on physical network adapters

vNIC/vHBA Placement specifies how vNICs and vHBAs are placed on physical network adapters (mezzanine) in a server hardware configuration independent way.

Select Placement: Specify Manually + Create Placement Policy

Virtual Network Interface connection provides a mechanism of placing vNICs and vHBAs on physical network adapters. vNICs and vHBAs are assigned to one of Virtual Network Interface connection specified below. This assignment can be performed explicitly by selecting which Virtual Network Interface connection is used by vNIC or vHBA or it can be done automatically by selecting "any".  
vNIC/vHBA placement on physical network interface is controlled by placement preferences.

Please select one Virtual Network Interface and one or more vNICs or vHBAs

vNICs
vHBAs

Name	
External-NIC	▲
Storage-M...	
Storage-Pub	
Tenant-Flo...	▼

>> assign >>  
<< remove <<

Specific Virtual Network Interfaces (click on a cell to edit)

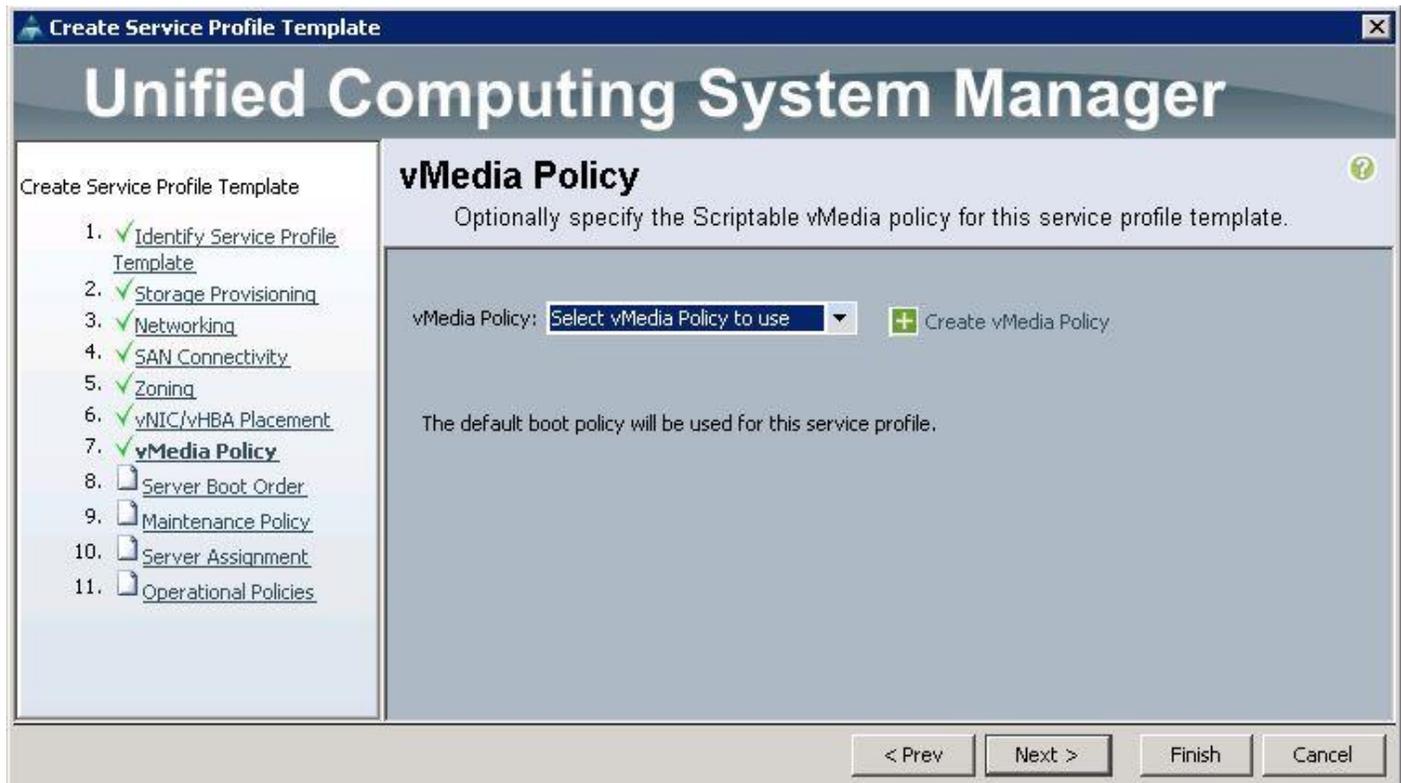
Name	Order	Admin Host Port	Selection Preference
vCon 1			All
vNIC PXE-NIC	1	ANY	
vNIC eth1	2	ANY	
vNIC Internal-API	3	ANY	
vCon 2			All
vCon 3			All
vCon 4			All

▲ Move Up
▼ Move Down

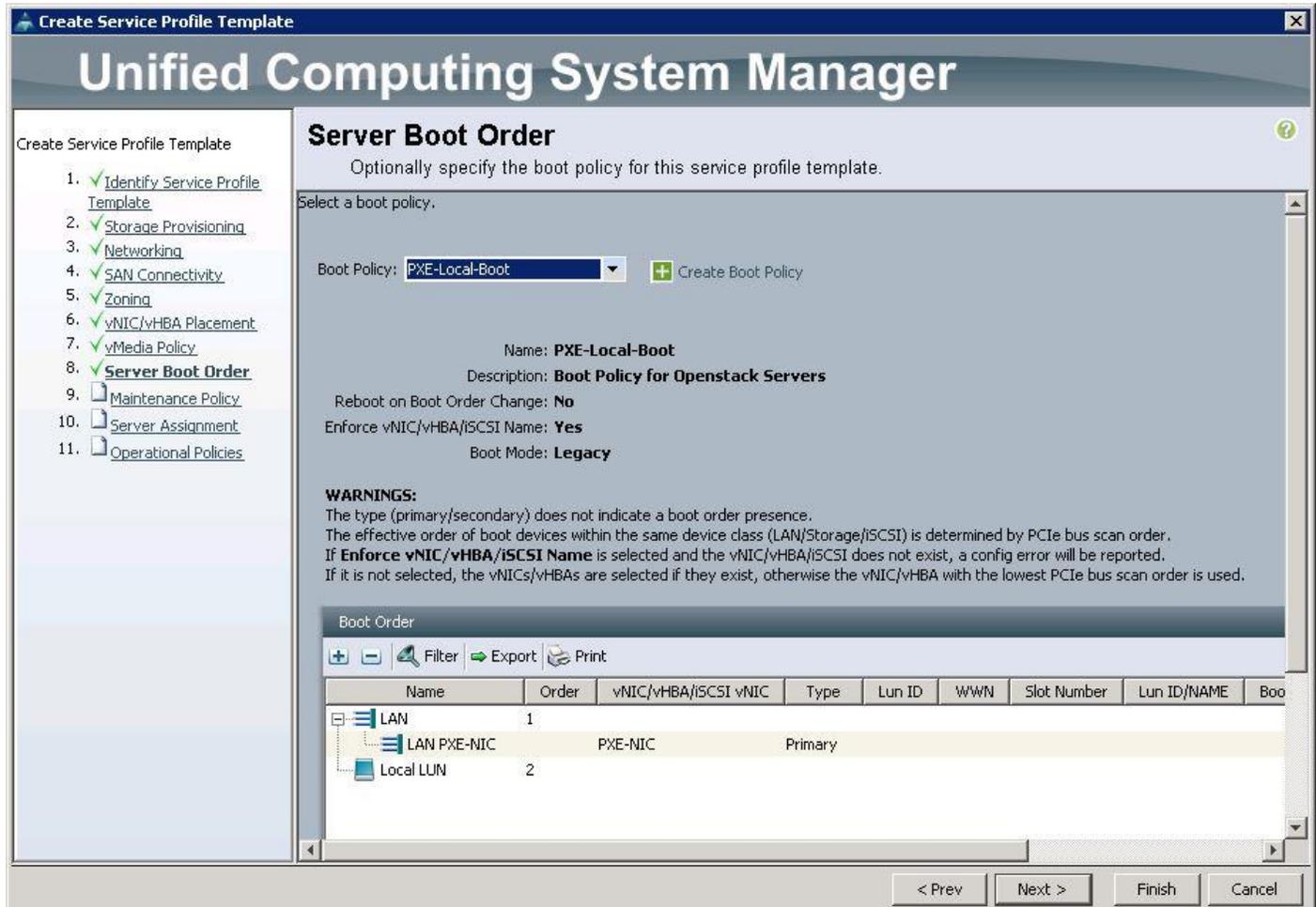
< Prev
Next >
Finish
Cancel



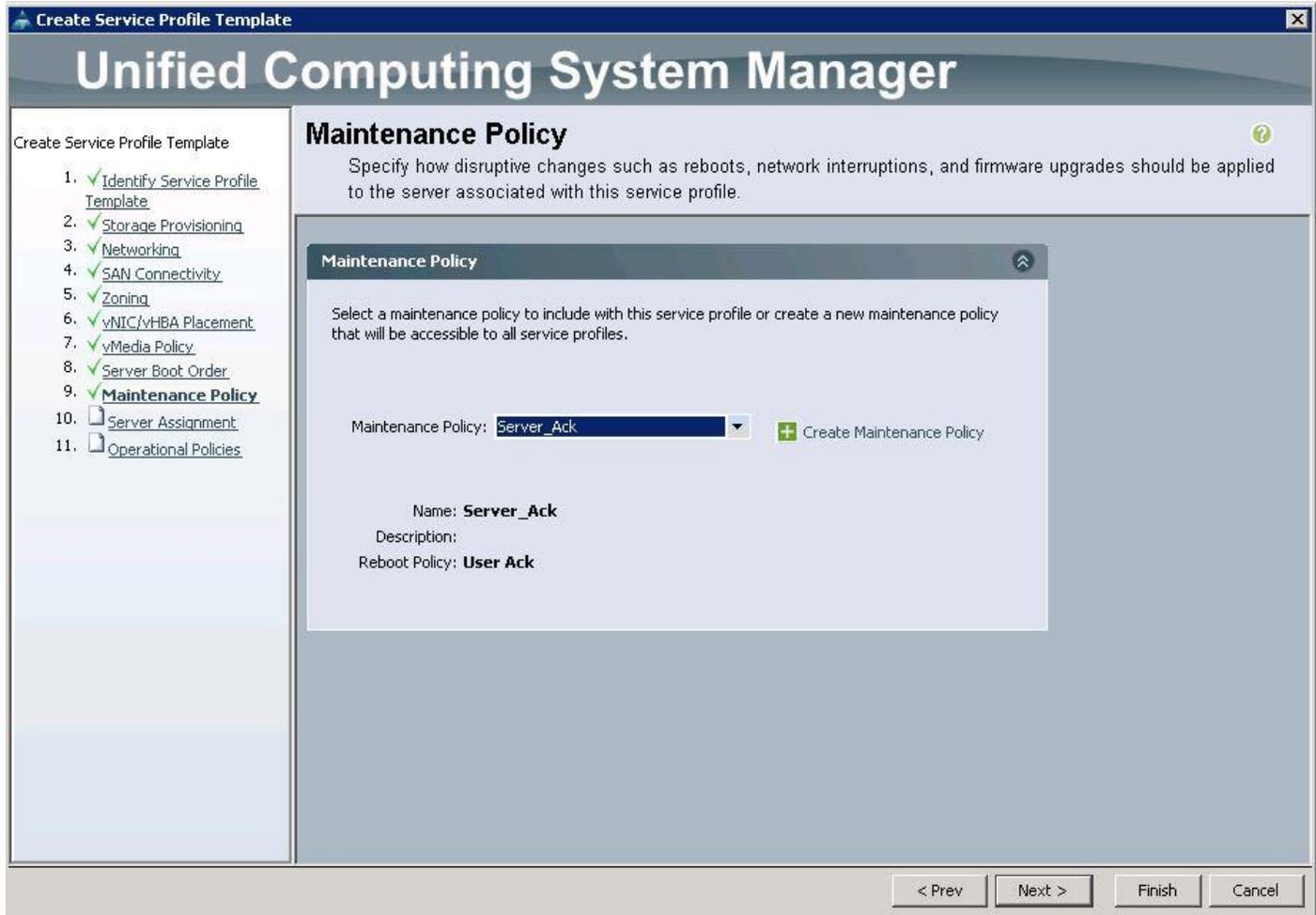
- u. Under vMedia Policy, click Next.



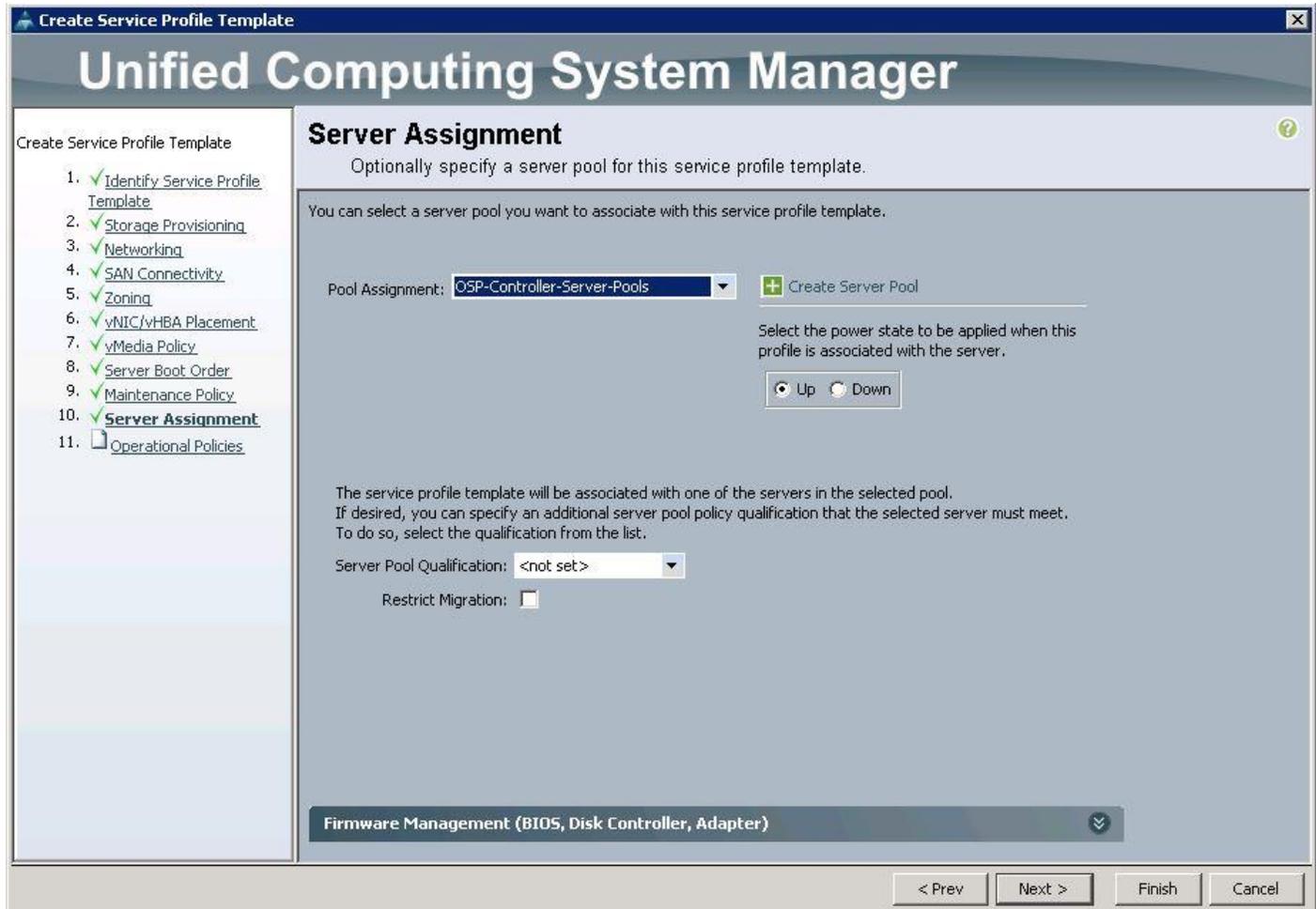
- v. Under Server Boot Order, choose the boot policy as PXE-LocalBoot previously created, from the drop-down list and click Next.



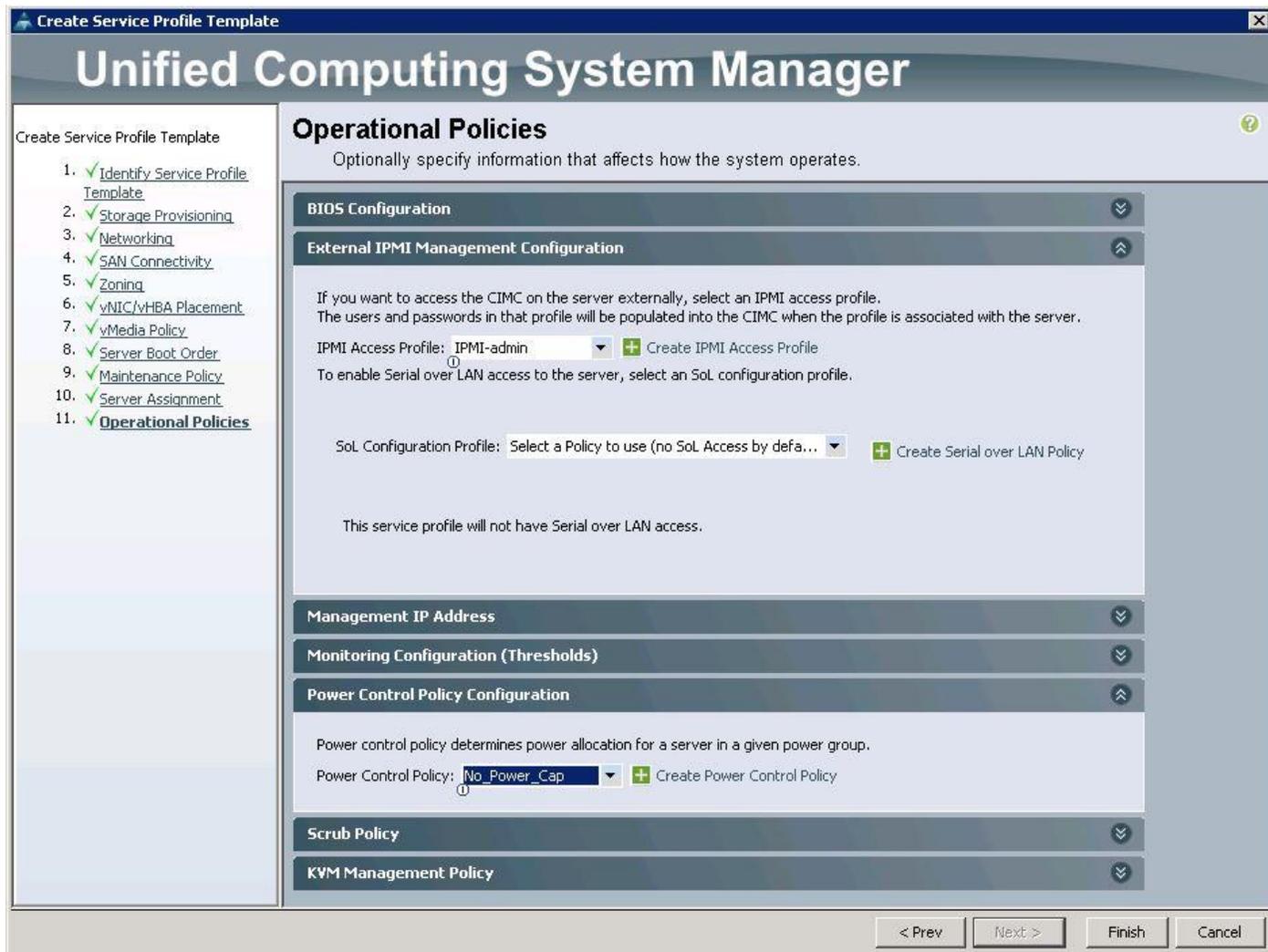
- w. Under Maintenance Policy, choose Server\_Ack previously created, from the drop-down list and click Next.



- x. Under Server Assignment, choose the Pool Assignment as OSP-Controller-Server-Pools previously created, from the drop-down list and click Next.



- y. Under Operational Policies, choose the IPMI Access Profile as IPMI\_admin previously created, from the drop-down list and choose the Power Control Policy as No\_Power\_Cap and click Finish.



## Create Service Profile Templates for Compute Nodes

To create the Service Profile templates for the Compute nodes, complete the following steps:

1. Specify the Service profile template name for the Controller node as OSP-Compute-SP-Template.
2. Choose the UUID pools previously created from the drop-down list and click Next.

**Create Service Profile Template**

# Unified Computing System Manager

Create Service Profile Template

1. **Identify Service Profile Template**
2. Storage Provisioning
3. Networking
4. SAN Connectivity
5. Zoning
6. vNIC/vHBA Placement
7. vMedia Policy
8. Server Boot Order
9. Maintenance Policy
10. Server Assignment
11. Operational Policies

## Identify Service Profile Template

You must enter a name for the service profile template and specify the template type. You can also specify how a UUID will be assigned to this template and enter a description.

Name:

The template will be created in the following organization. Its name must be unique within this organization.

Where: **org-root**

The template will be created in the following organization. Its name must be unique within this organization.

Type:  Initial Template  Updating Template

Specify how the UUID will be assigned to the server associated with the service generated by this template.

**UUID**

UUID Assignment:

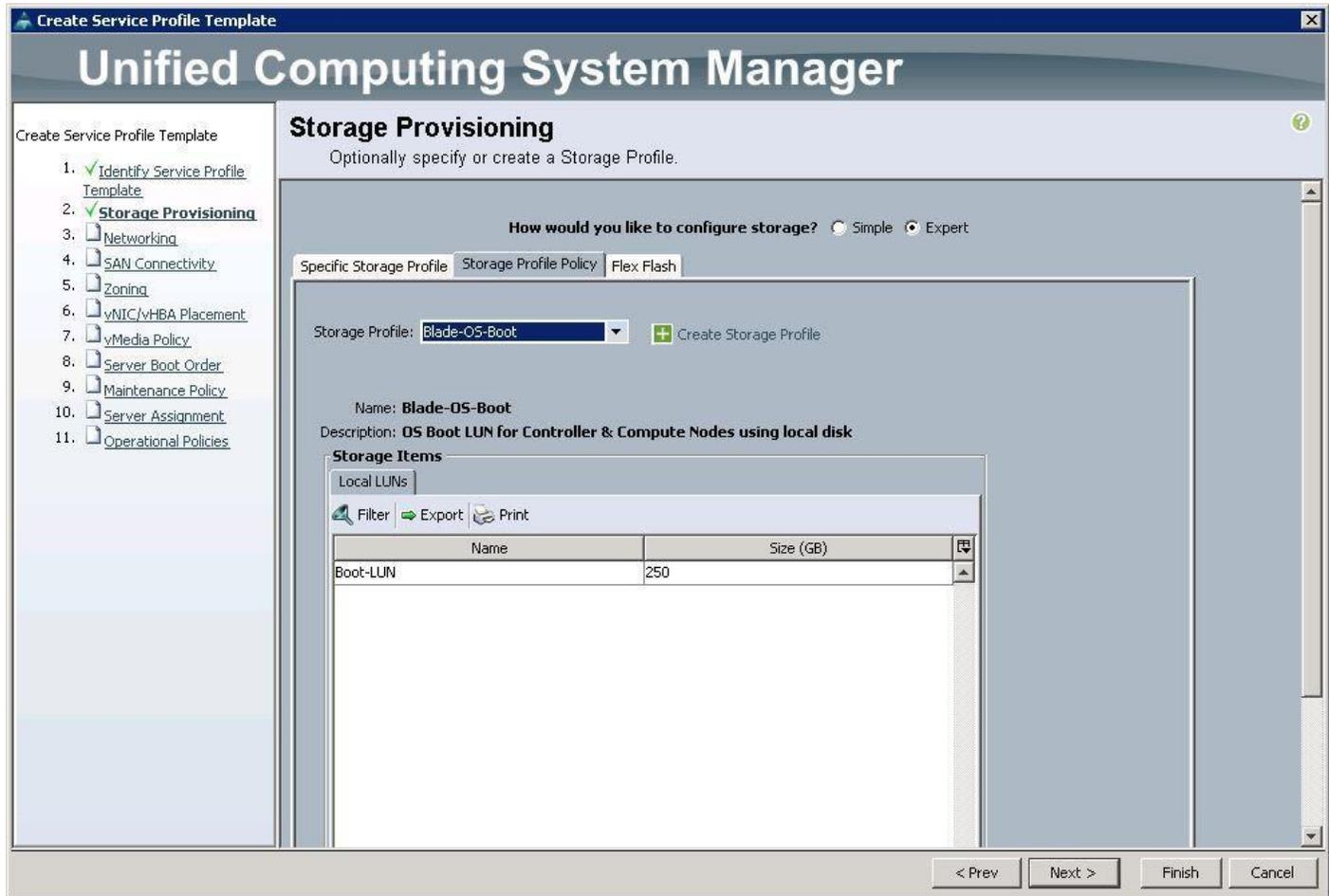
The UUID will be assigned from the selected pool.  
The available/total UUIDs are displayed after the pool name.

Optionally enter a description for the profile. The description can contain information about when and where the service profile should be used.

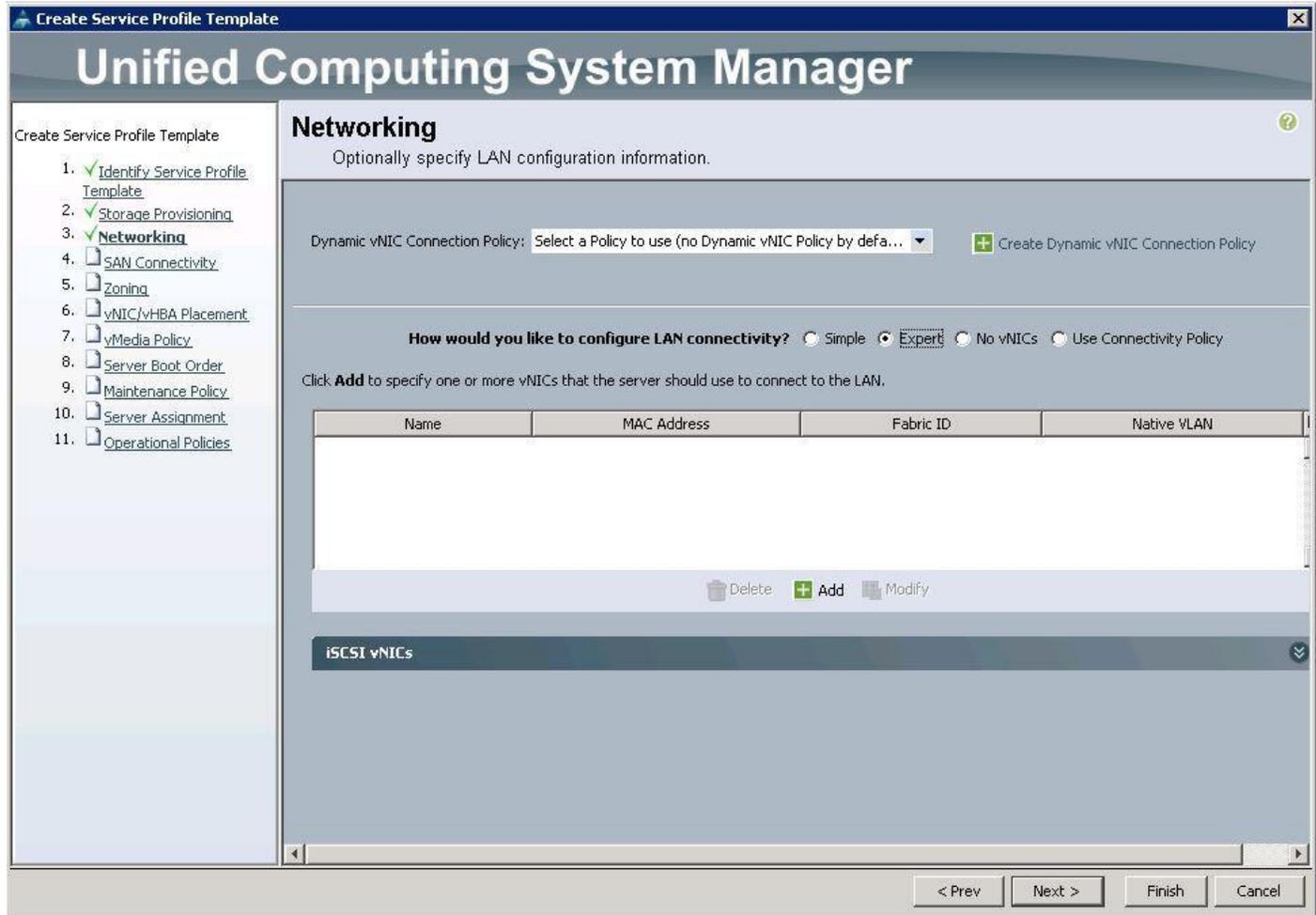
**Service Profile Template for Openstack Compute Nodes**

< Prev Next > Finish Cancel

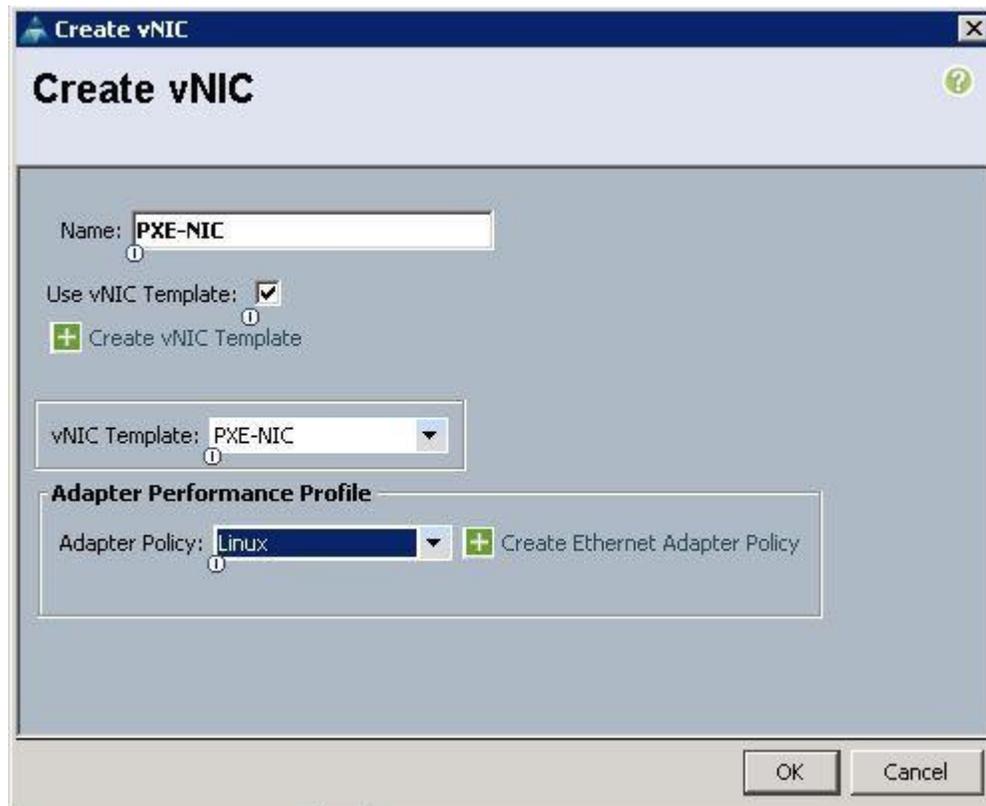
3. For Storage Provisioning, choose Expert and click Storage Profile Policy and choose the Storage profile Blade-OS-boot previously created, from the drop-down list and click Next.



4. For Networking, choose Expert and click "+".



5. Create the vNIC interface for PXE or Provisioning network as PXE-NIC and click the check box for Use vNIC template.
6. Under the vNIC template, choose the PXE-NIC template previously created, from the drop-down list and choose Linux for the Adapter Policy.

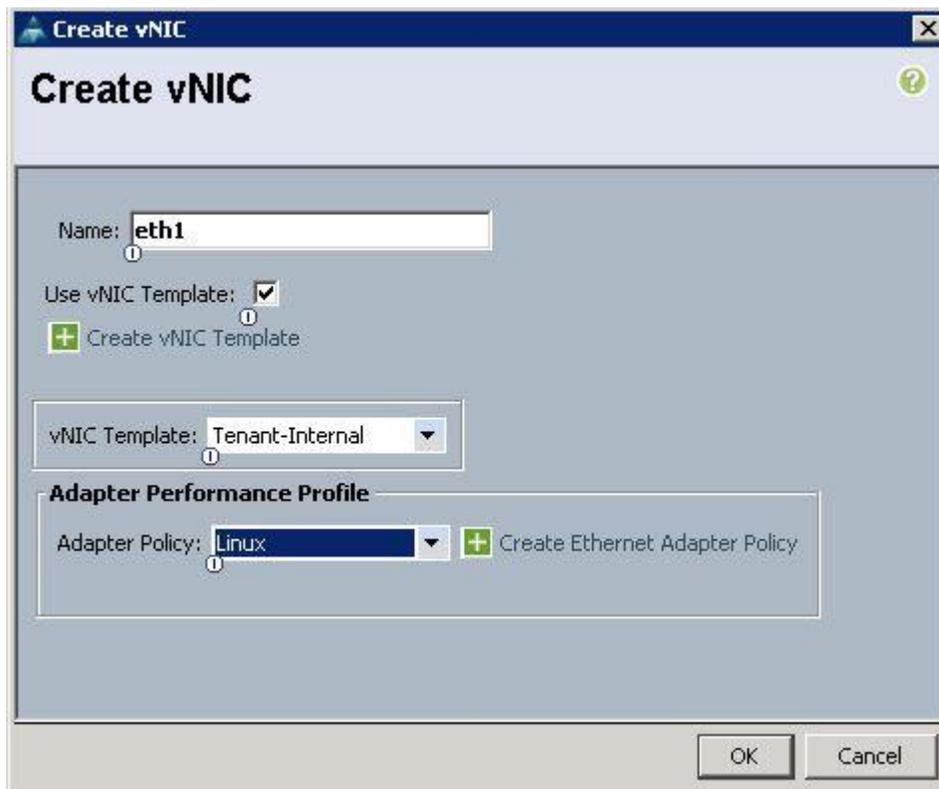


7. Create the vNIC interface for Tenant Internal Network as eth1 and then under vNIC template, choose the **“Tenant-Internal”** template we created before from the drop-down list and choose Adapter Policy as **“Linux”**.

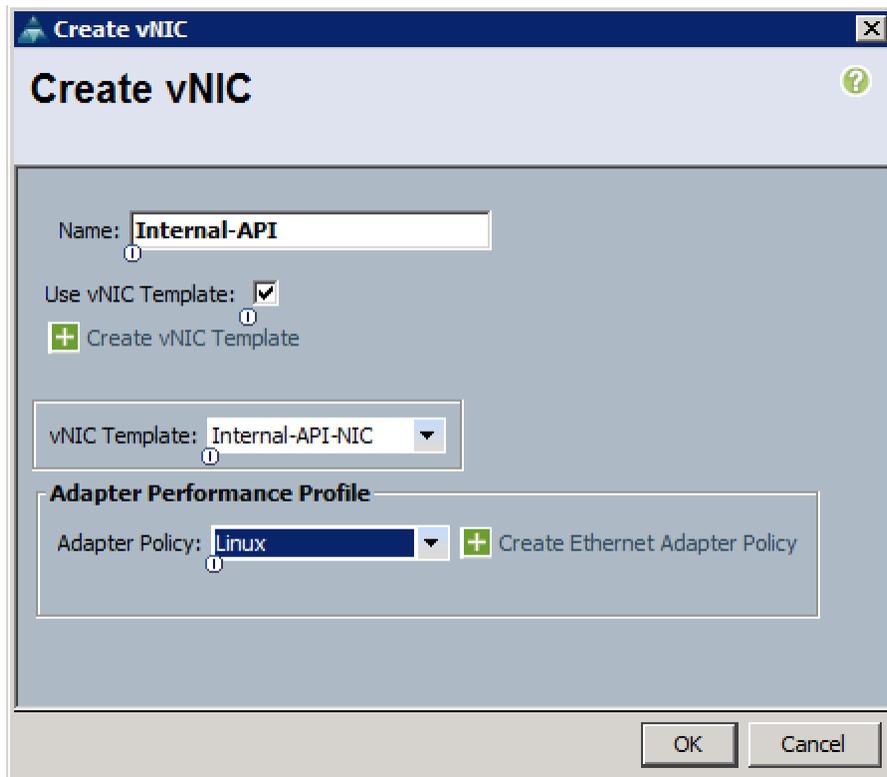


Due to the Cisco UCS Manager Plugin limitations, we have created eth1 as vNIC for Tenant Internal Network..

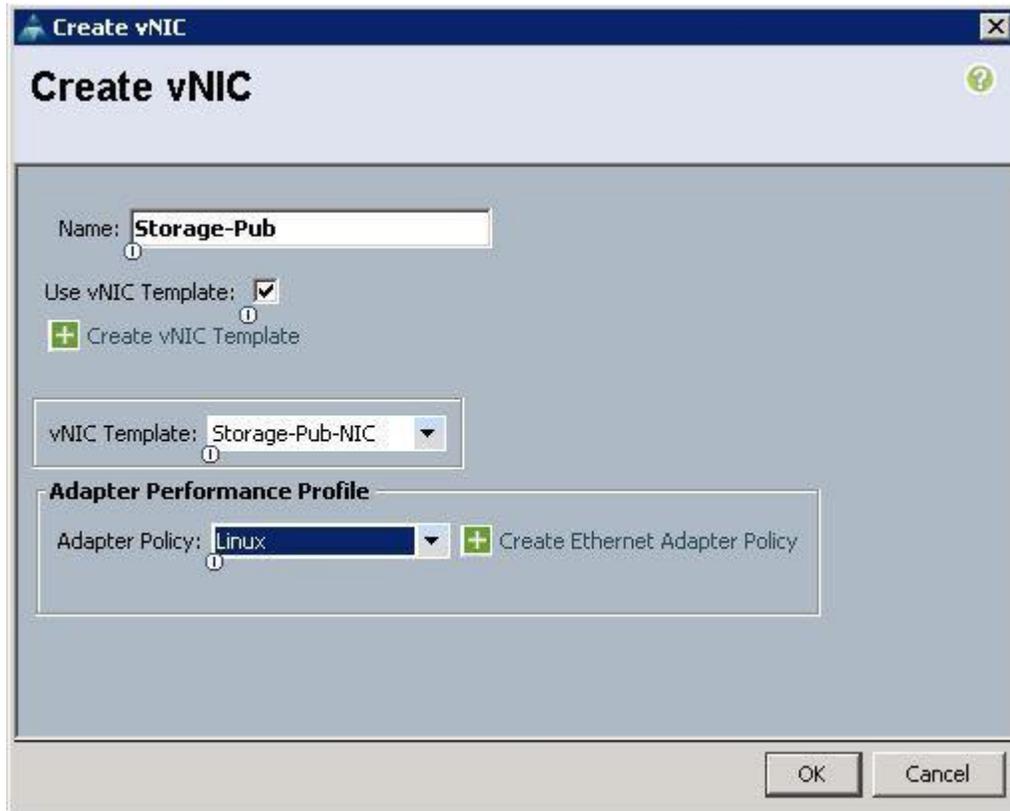
---



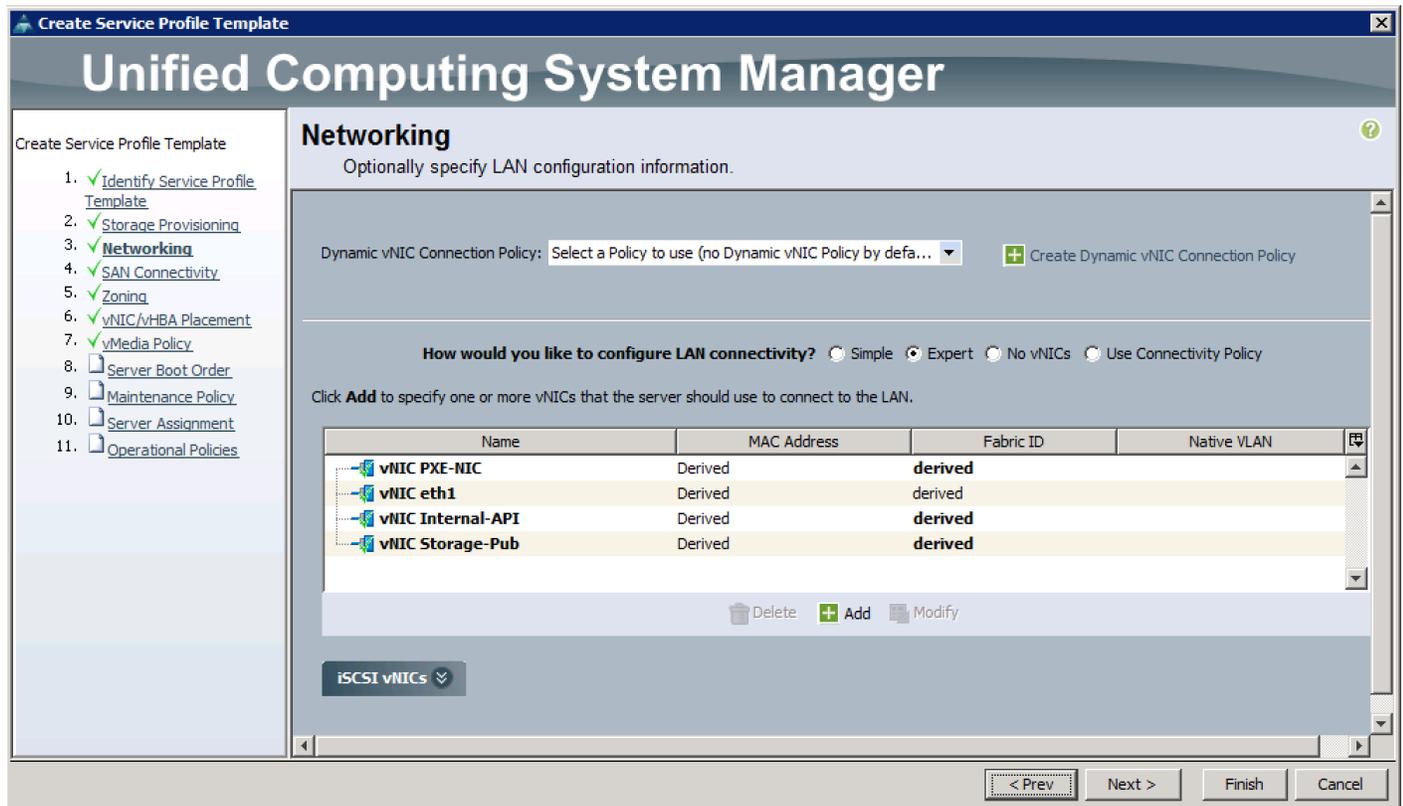
8. Create the vNIC interface for Internal API network as Internal-API and click the check box for vNIC template.
9. Under the vNIC template, choose the Internal-API template previously created, from the drop-down list and choose Linux for the Adapter Policy.



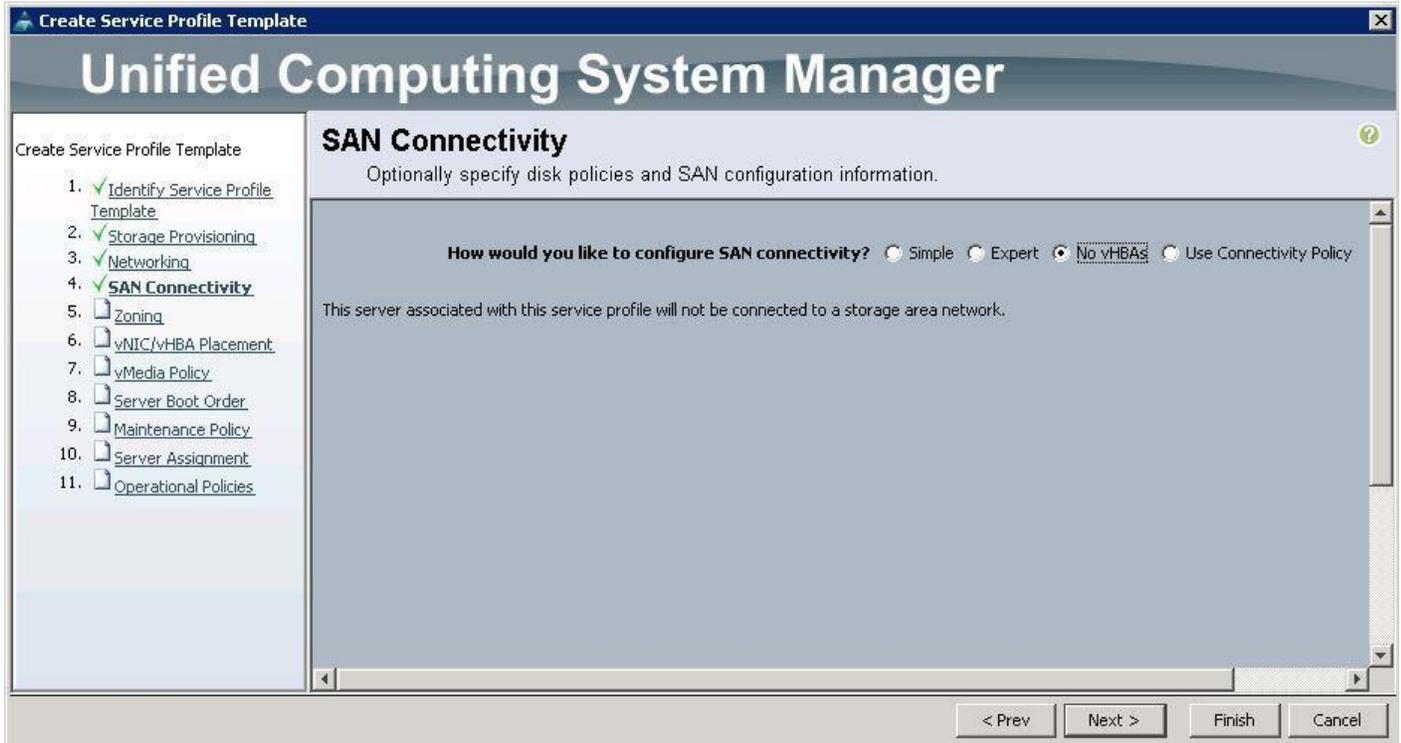
10. Create the vNIC interface for Storage Public Network as Storage-Pub and click the check box for Use vNIC template.
11. Under the vNIC template, choose the Storage-Pub-NIC template previously created, from the drop-down list and choose Linux for the Adapter Policy.



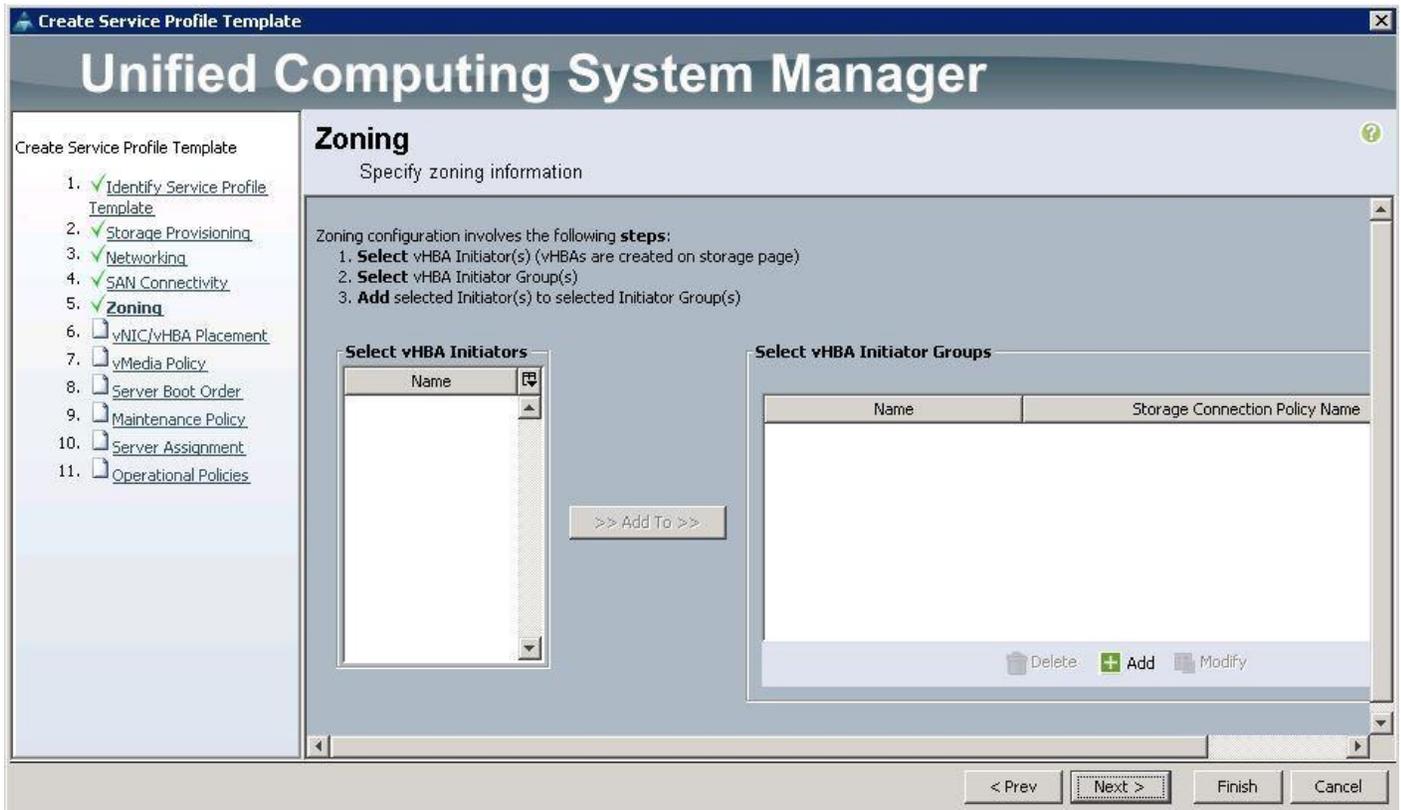
12. After a successful vNIC creation, click Next.



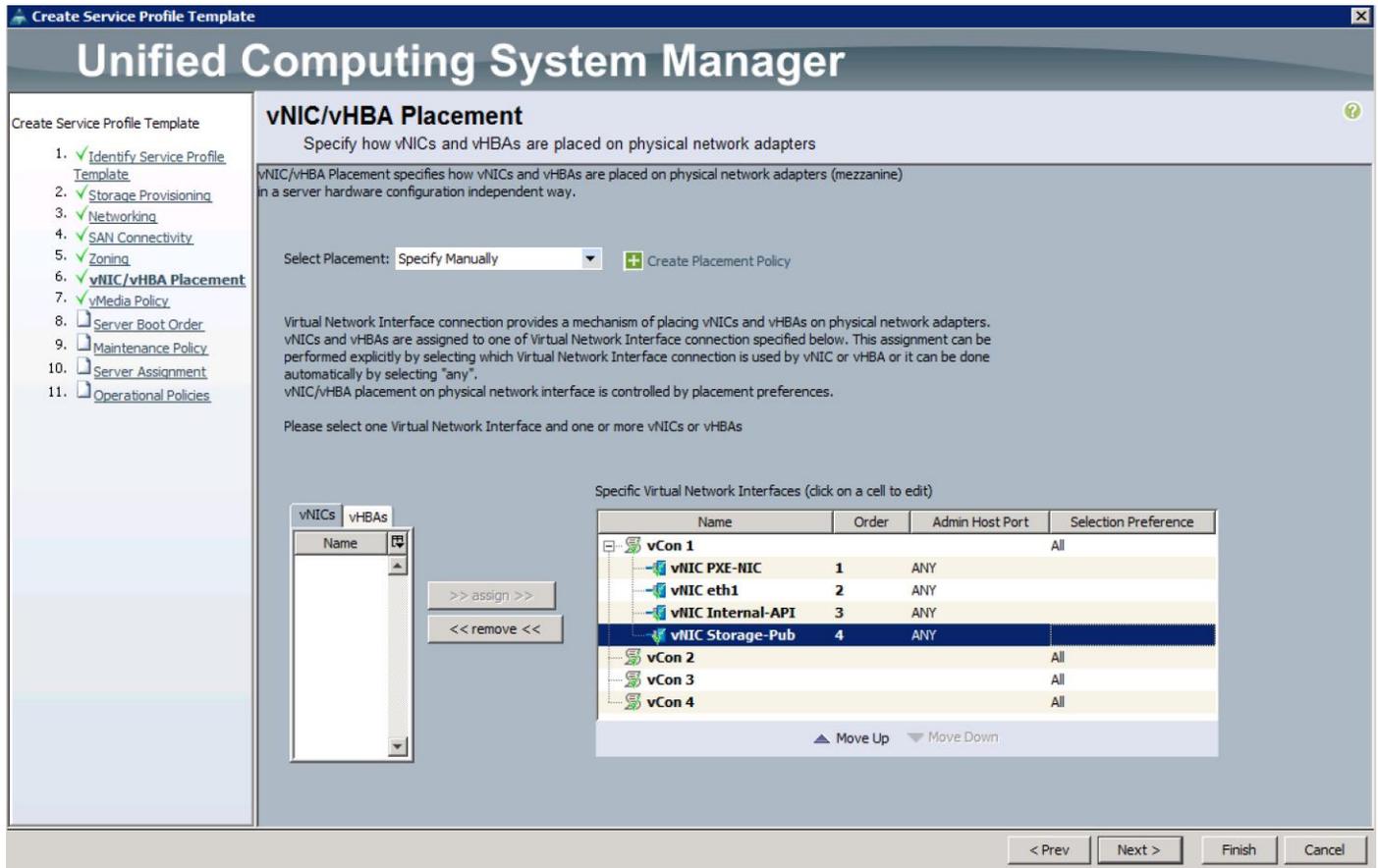
13. Under SAN connectivity, choose No VHBAs and click Next.



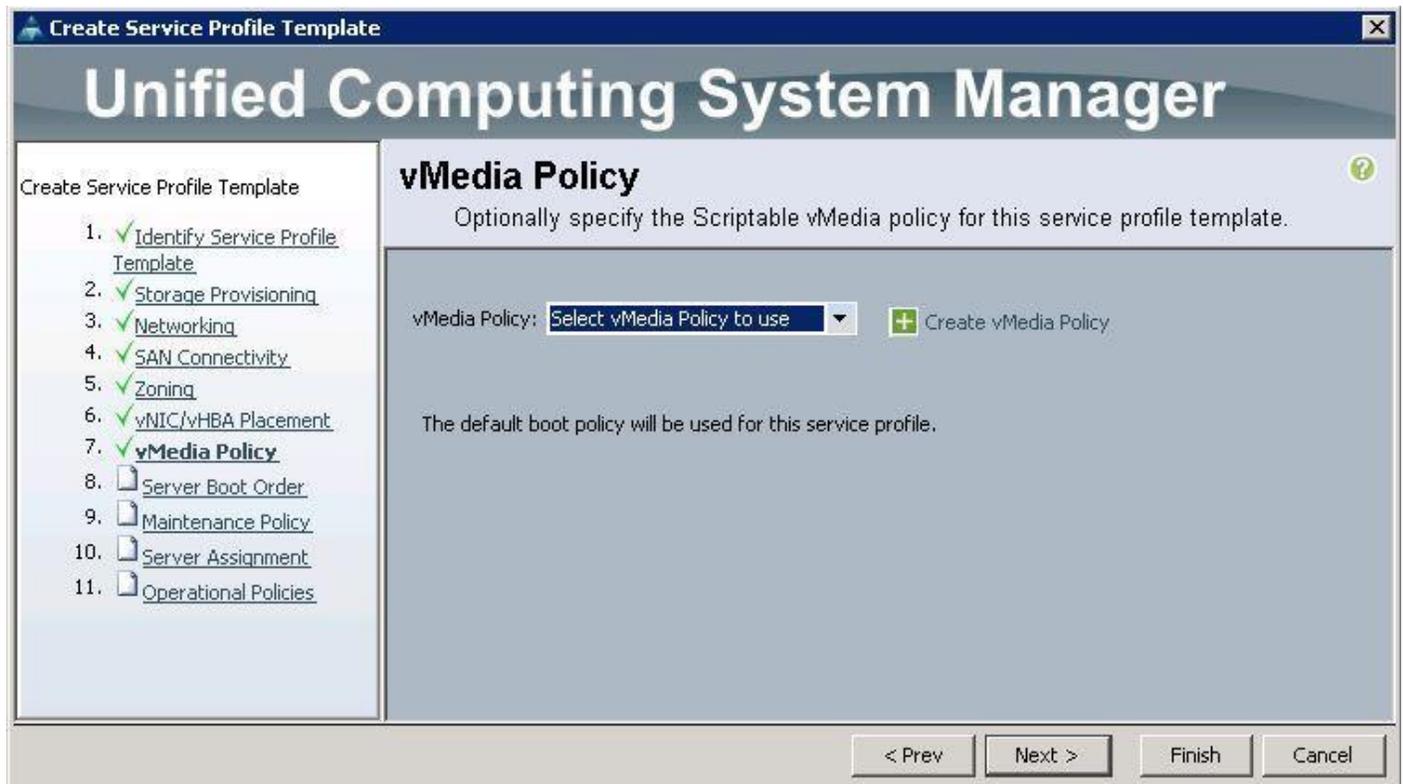
14. Under Zoning, click Next.



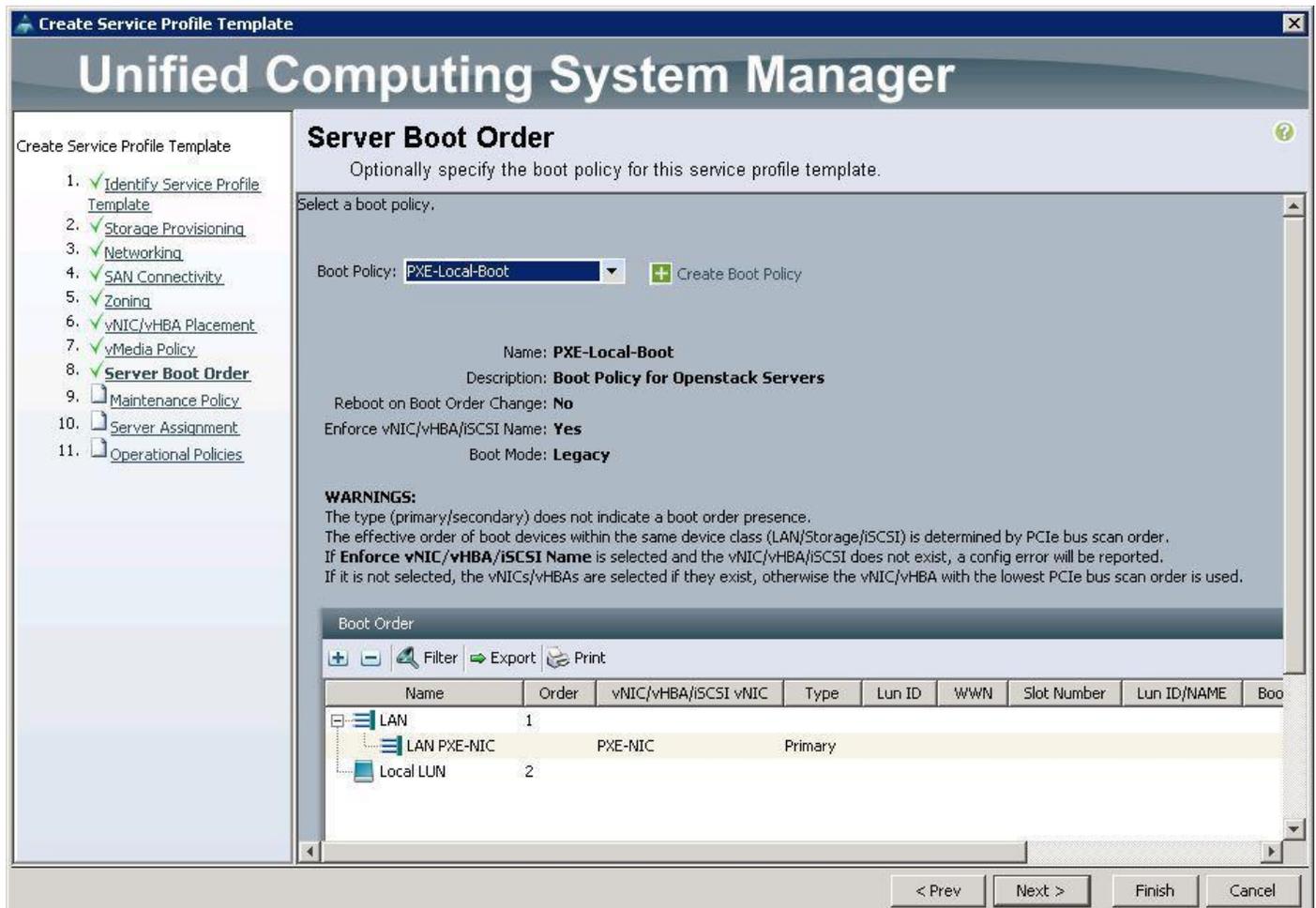
15. Under vNIC/vHBA Placement, choose the vNICs PCI order as shown below and click Next.



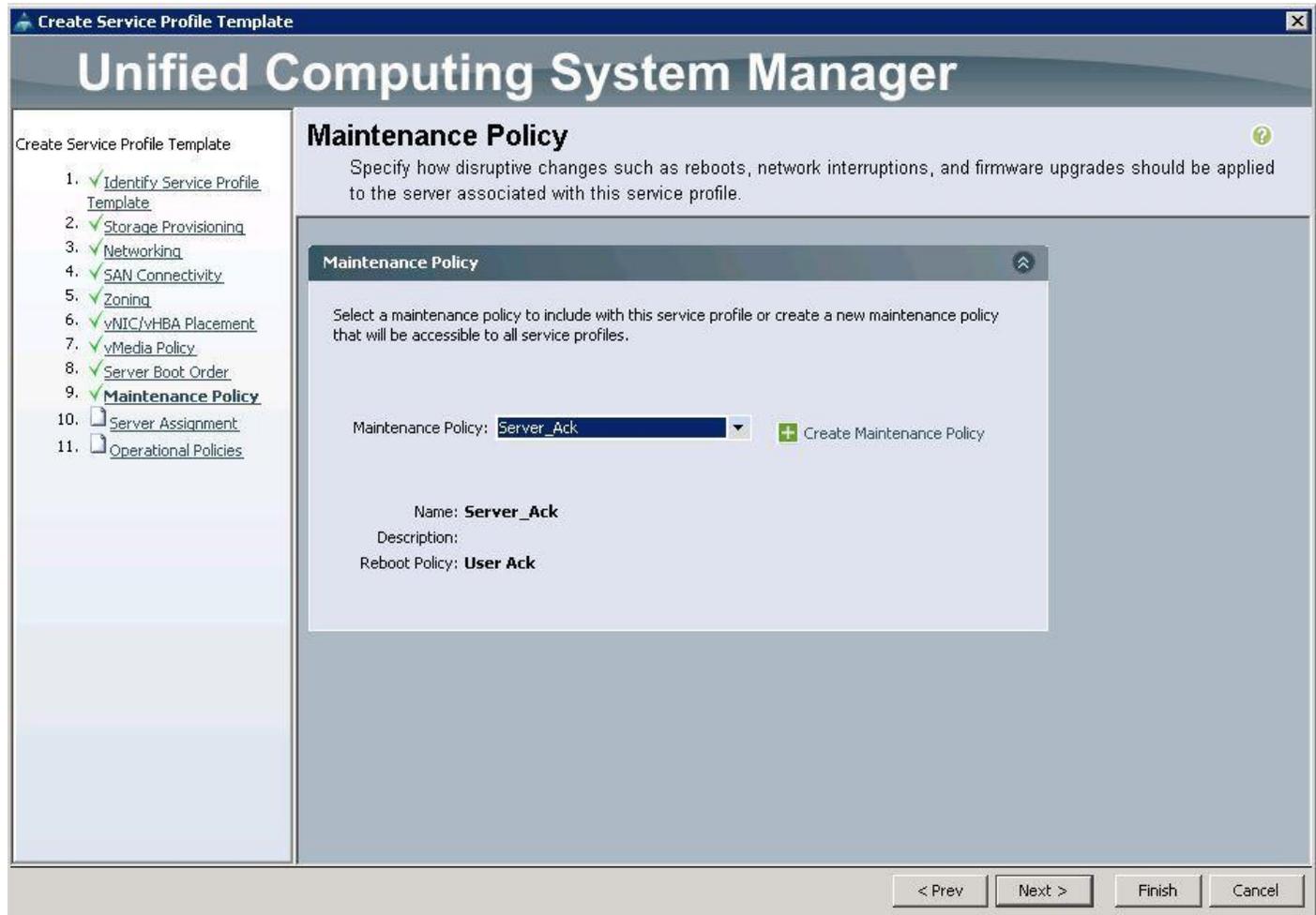
16. Under vMedia Policy, click Next.



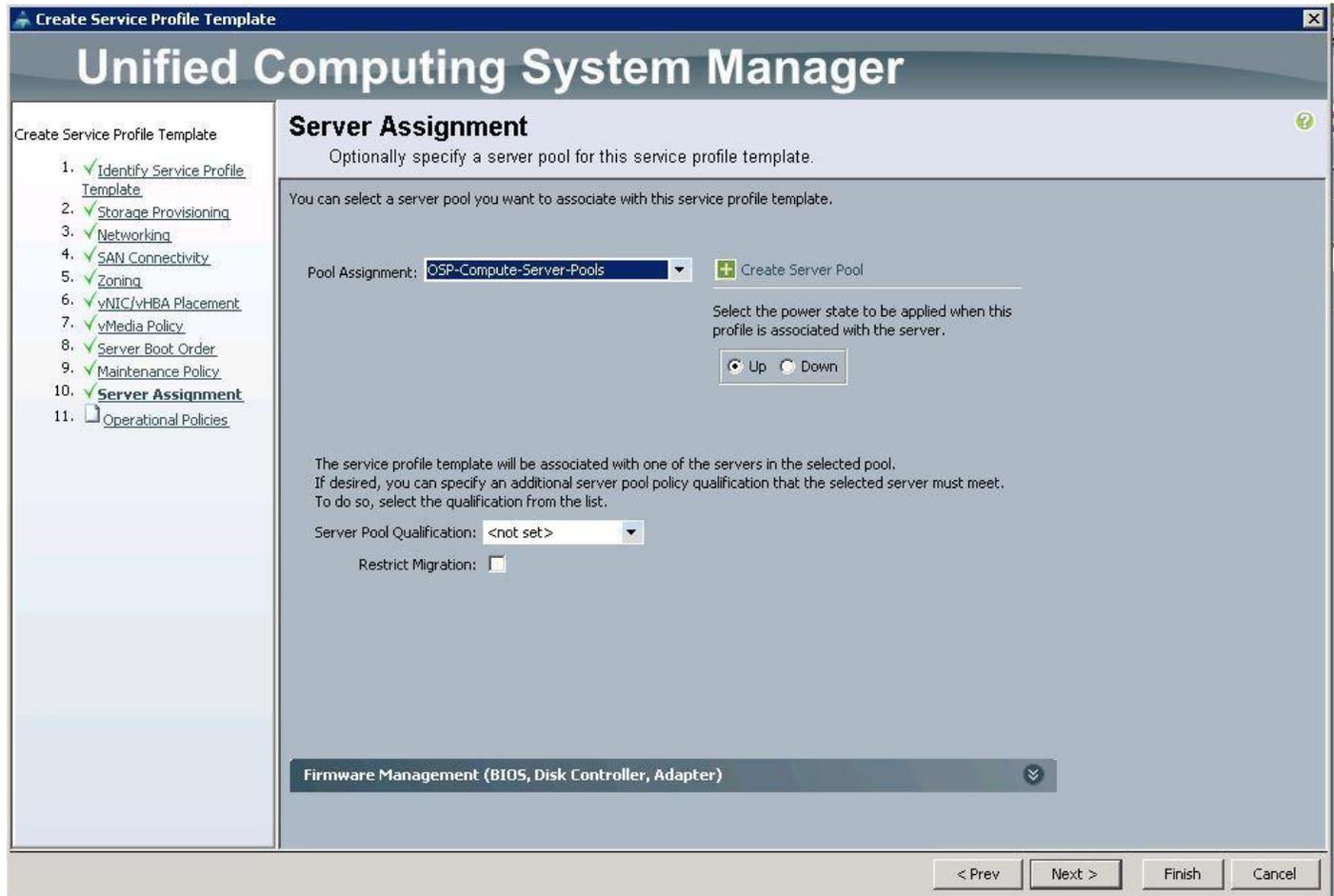
- Under Server Boot Order, choose boot policy as PXE-LocalBoot we created from the drop-down list and click Next.



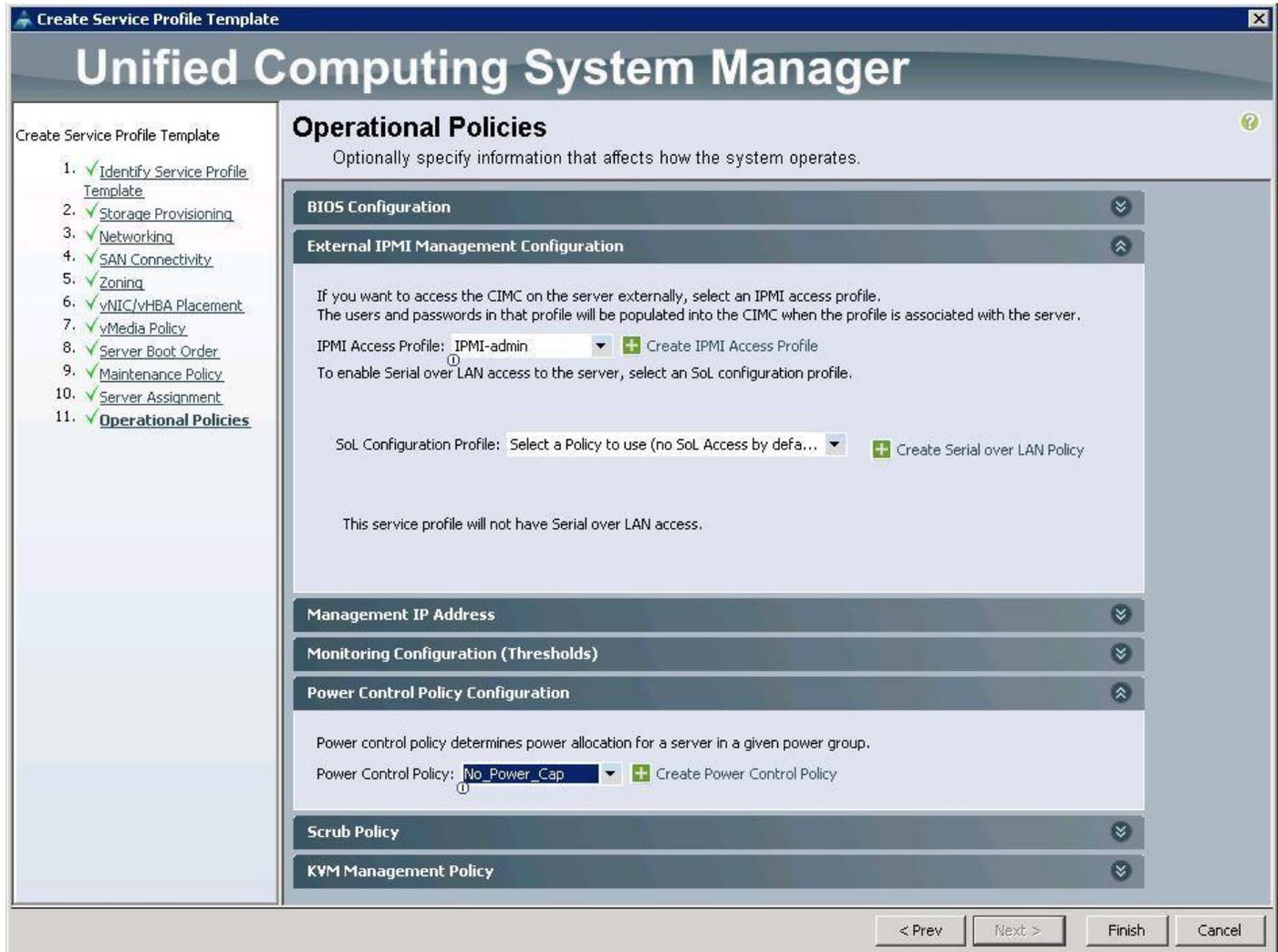
- Under Maintenance Policy, choose Server\_Ack previously created, from the drop-down list and click Next.



19. Under Server assignment, choose Pool Assignment as OSP-Compute-Server-Pools previously, created from the drop-down list and click Next.



20. Under Operational Policies, choose the IPMI Access Profile as IPMI\_admin previously created, from the drop-down list and choose the Power Control Policy as No\_Power\_Cap and click Finish.



## Create Service Profile Templates for Ceph Storage Nodes

To create the Service Profile templates for the Ceph Storage nodes, complete the following steps:

1. Specify the Service profile template name for the Ceph storage node as OSP-Ceph-Storage-SP-Template. Choose the UUID pools previously created, from the drop-down list and click Next.

Create Service Profile Template

# Unified Computing System Manager

Create Service Profile Template

1.  **Identify Service Profile Template**
2.  Storage Provisioning
3.  Networking
4.  SAN Connectivity
5.  Zoning
6.  vNIC/vHBA Placement
7.  vMedia Policy
8.  Server Boot Order
9.  Maintenance Policy
10.  Server Assignment
11.  Operational Policies

## Identify Service Profile Template

You must enter a name for the service profile template and specify the template type. You can also specify how a UUID will be assigned to this template and enter a description.

Name:

The template will be created in the following organization. Its name must be unique within this organization.  
Where: **org-root**

The template will be created in the following organization. Its name must be unique within this organization.

Type:  Initial Template  Updating Template

Specify how the UUID will be assigned to the server associated with the service generated by this template.

**UUID**

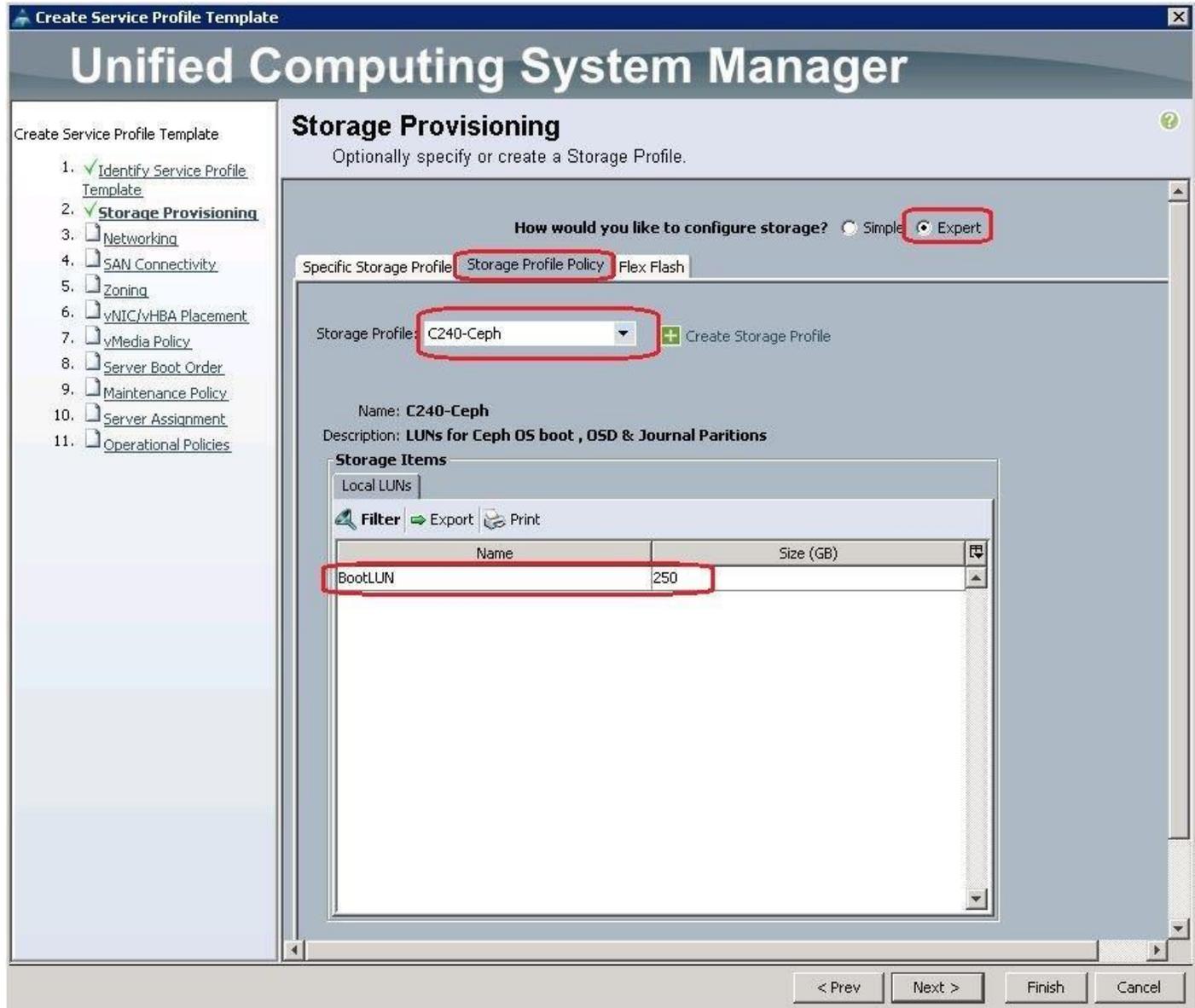
UUID Assignment:

The UUID will be assigned from the selected pool.  
The available/total UUIDs are displayed after the pool name.

Optionally enter a description for the profile. The description can contain information about when and where the service profile should be used.

**Service Profile Template for Openstack Ceph Storage Nodes**

< Prev    Next >    Finish    Cancel



2. Create vNIC's for PXE, Storage-Pub and Storage-Mgmt following steps similar to controller as [mentioned here](#).

**Create Service Profile Template**

# Unified Computing System Manager

**Networking**

Optionally specify LAN configuration information.

Dynamic vNIC Connection Policy:  + Create Dynamic vNIC Connection Policy

How would you like to configure LAN connectivity?  Simple  Expert  No vNICs  Use Connectivity Policy

Click **Add** to specify one or more vNICs that the server should use to connect to the LAN.

Name	MAC Address	Fabric ID	Native VLAN
vNIC PXE-NIC	Derived	<b>derived</b>	
vNIC Storage-Pub	Derived	<b>derived</b>	
vNIC Storage-Mgmt	Derived	<b>derived</b>	

🗑️ Delete + ➕ Add 📄 Modify

**iSCSI vNICs**

Create Service Profile Template
✕

# Unified Computing System Manager

Create Service Profile Template

1.  Identify Service Profile Template
2.  Storage Provisioning
3.  Networking
4.  SAN Connectivity
5.  Zoning
6.  **vNIC/vHBA Placement**
7.  vMedia Policy
8.  Server Boot Order
9.  Maintenance Policy
10.  Server Assignment
11.  Operational Policies

## vNIC/vHBA Placement

Specify how vNICs and vHBAs are placed on physical network adapters

vNIC/vHBA Placement specifies how vNICs and vHBAs are placed on physical network adapters (mezzanine) in a server hardware configuration independent way.

Select Placement: Specify Manually + Create Placement Policy

Virtual Network Interface connection provides a mechanism of placing vNICs and vHBAs on physical network adapters. vNICs and vHBAs are assigned to one of Virtual Network Interface connection specified below. This assignment can be performed explicitly by selecting which Virtual Network Interface connection is used by vNIC or vHBA or it can be done automatically by selecting "any".  
vNIC/vHBA placement on physical network interface is controlled by placement preferences.

Please select one Virtual Network Interface and one or more vNICs or vHBAs

vNICs	vHBAs
Name	Name

>> assign >>  
<< remove <<

Specific Virtual Network Interfaces (click on a cell to edit)

Name	Order	Admin Host Port	Selection Preference
vCon 1			All
- vNIC PXE-NIC	1	ANY	
- vNIC Storage-Pub	2	ANY	
- vNIC Storage-Mgmt	3	ANY	
vCon 2			All
vCon 3			All
vCon 4			All

▲ Move Up   ▼ Move Down

< Prev
Next >
Finish
Cancel

# Unified Computing System Manager

Create Service Profile Template

1.  Identify Service Profile Template
2.  Storage Provisioning
3.  Networking
4.  SAN Connectivity
5.  Zoning
6.  vNIC/vHBA Placement
7.  vMedia Policy
8.  **Server Boot Order**
9.  Maintenance Policy
10.  Server Assignment
11.  Operational Policies

## Server Boot Order

Optionally specify the boot policy for this service profile template.

Select a boot policy:

Boot Policy: PXE-Local-Boot + Create Boot Policy

Name: **PXE-Local-Boot**  
 Description: **Boot Policy for Openstack Servers**

Reboot on Boot Order Change: **No**  
 Enforce vNIC/vHBA/iSCSI Name: **Yes**  
 Boot Mode: **Legacy**

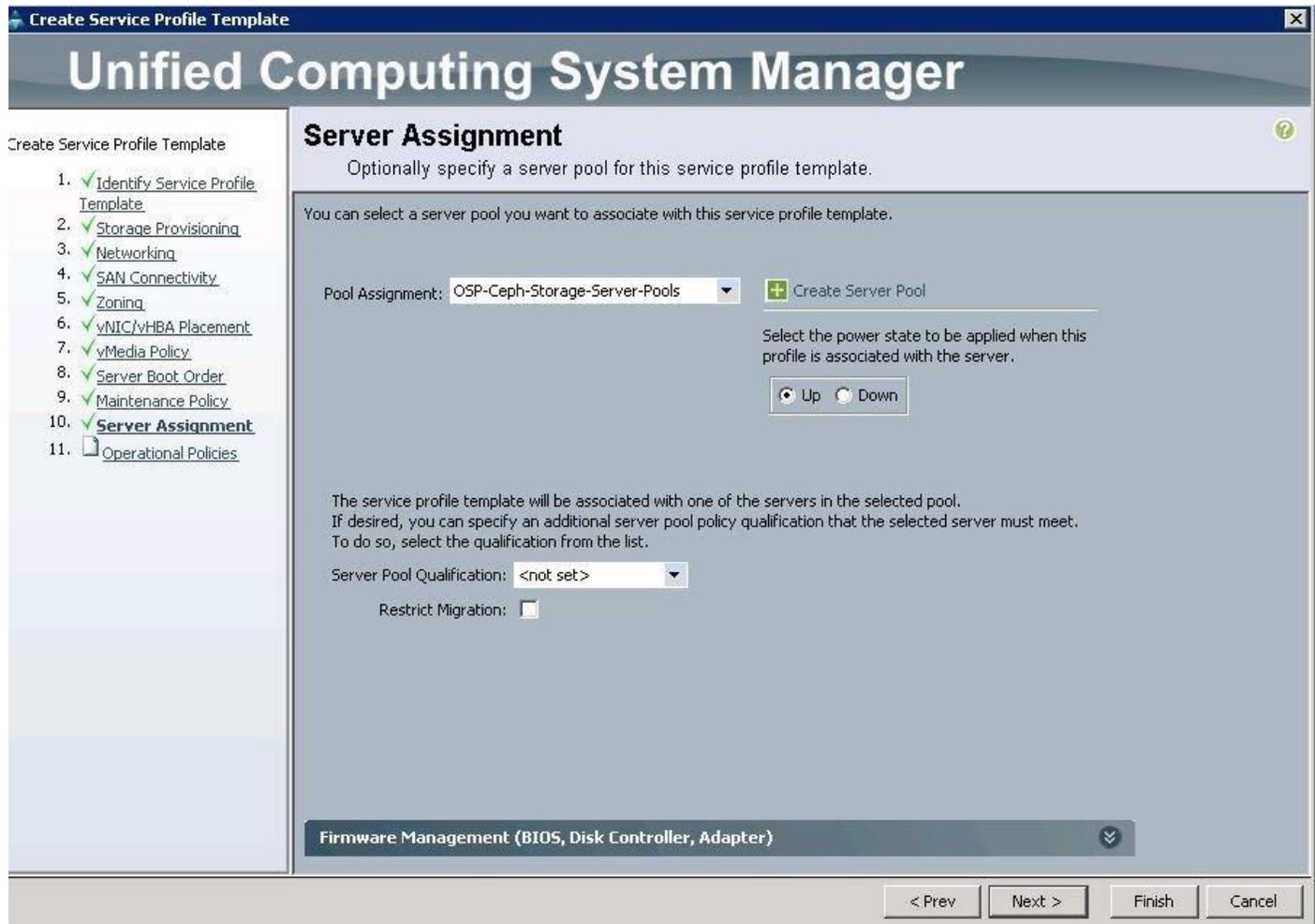
**WARNINGS:**  
 The type (primary/secondary) does not indicate a boot order presence.  
 The effective order of boot devices within the same device class (LAN/Storage/iSCSI) is determined by PCIe bus scan order.  
 If **Enforce vNIC/vHBA/iSCSI Name** is selected and the vNIC/vHBA/iSCSI does not exist, a config error will be reported.  
 If it is not selected, the vNICs/vHBAs are selected if they exist, otherwise the vNIC/vHBA with the lowest PCIe bus scan order is used.

Boot Order

+ - Filter Export Print

Name	Order	vNIC/vHBA/iSCSI vNIC	Type	Lun ID	WWN	Slot Number	Lun ID/NAME	Boot Name	Bo
LAN	1								
LAN PXE-NIC		PXE-NIC	Primary						
Local LUN	2								

< Prev
Next >
Finish
Cancel



3. Click Next and then Choose “Server\_Ack” under Maintenance Policy and then choose the “OSP-CephStorage-Server-Pools” under Pool assignment . Then select “No-power-cap” under power control policy. Click on Finish to complete the Service profile template creation for Ceph nodes.

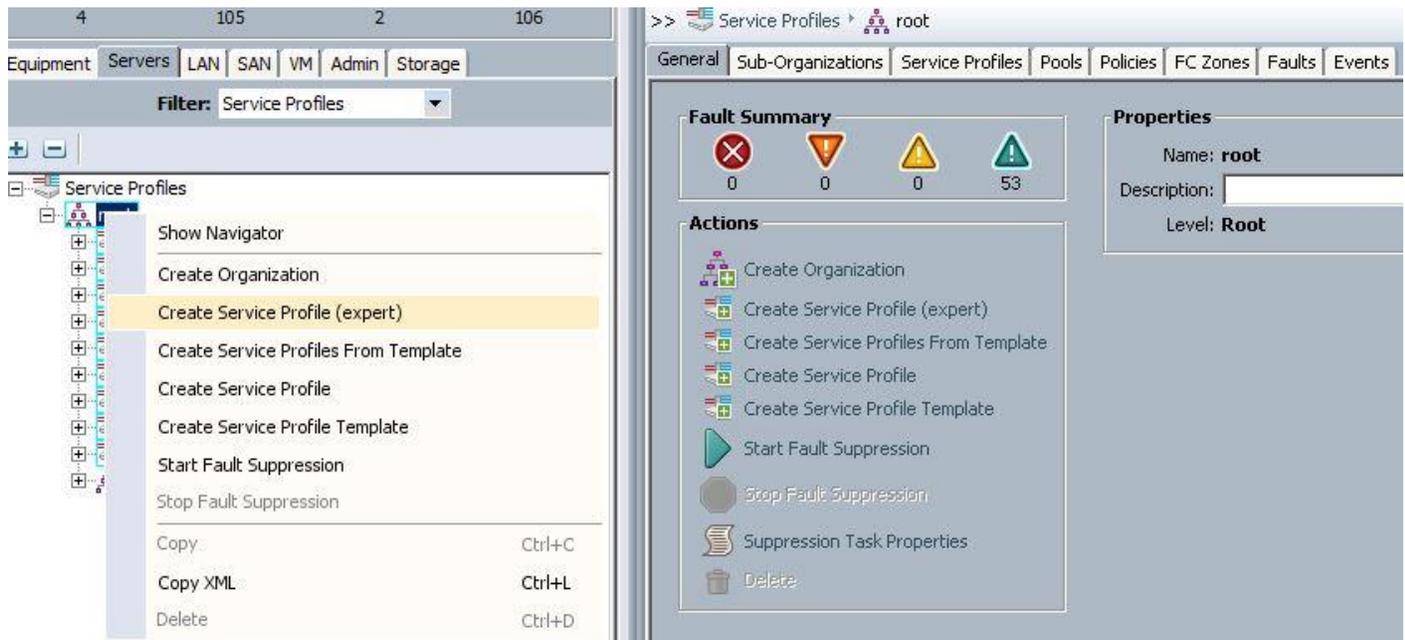
## Create Service Profile for Undercloud ( OSP7 Director ) Node

To configure the Service Profile for Undercloud (OSP7 Director) Node, complete the following steps:

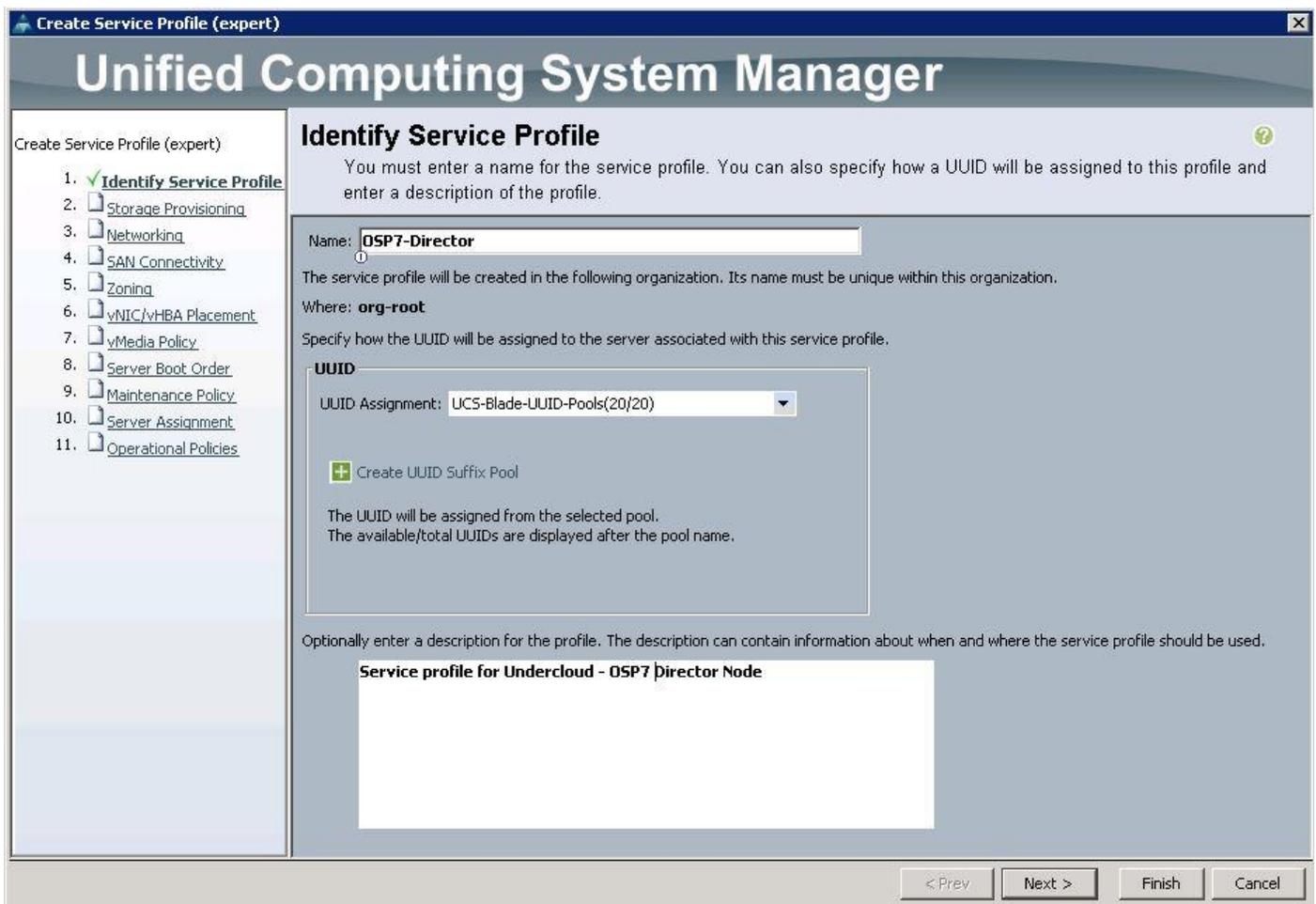


As there is only one node for Undercloud, a single Service Profile is created. There are no Service Profile Templates for the undercloud node.

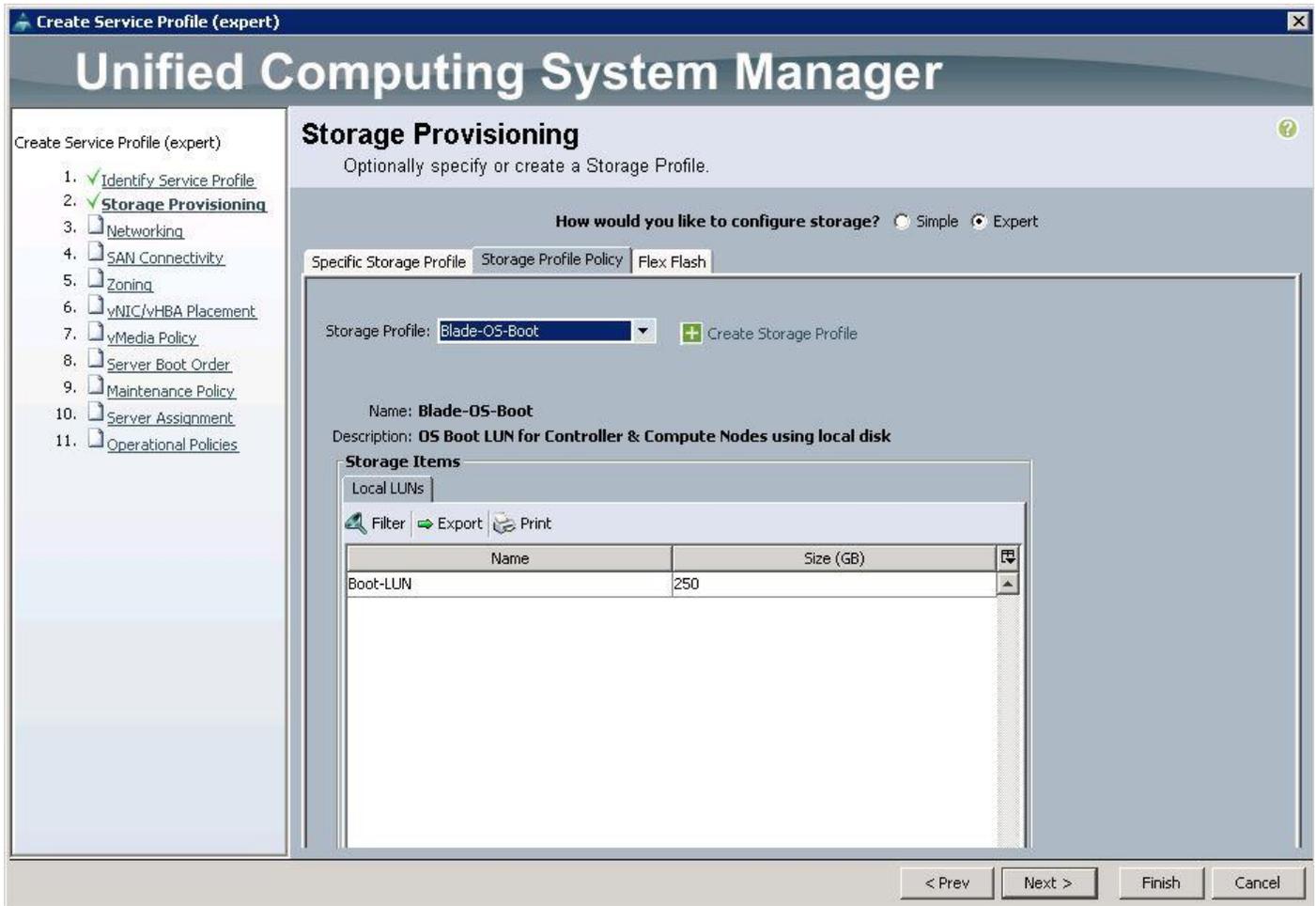
1. Under Servers > Service Profiles > root > right-click and select “Create Service Profile (expert)”



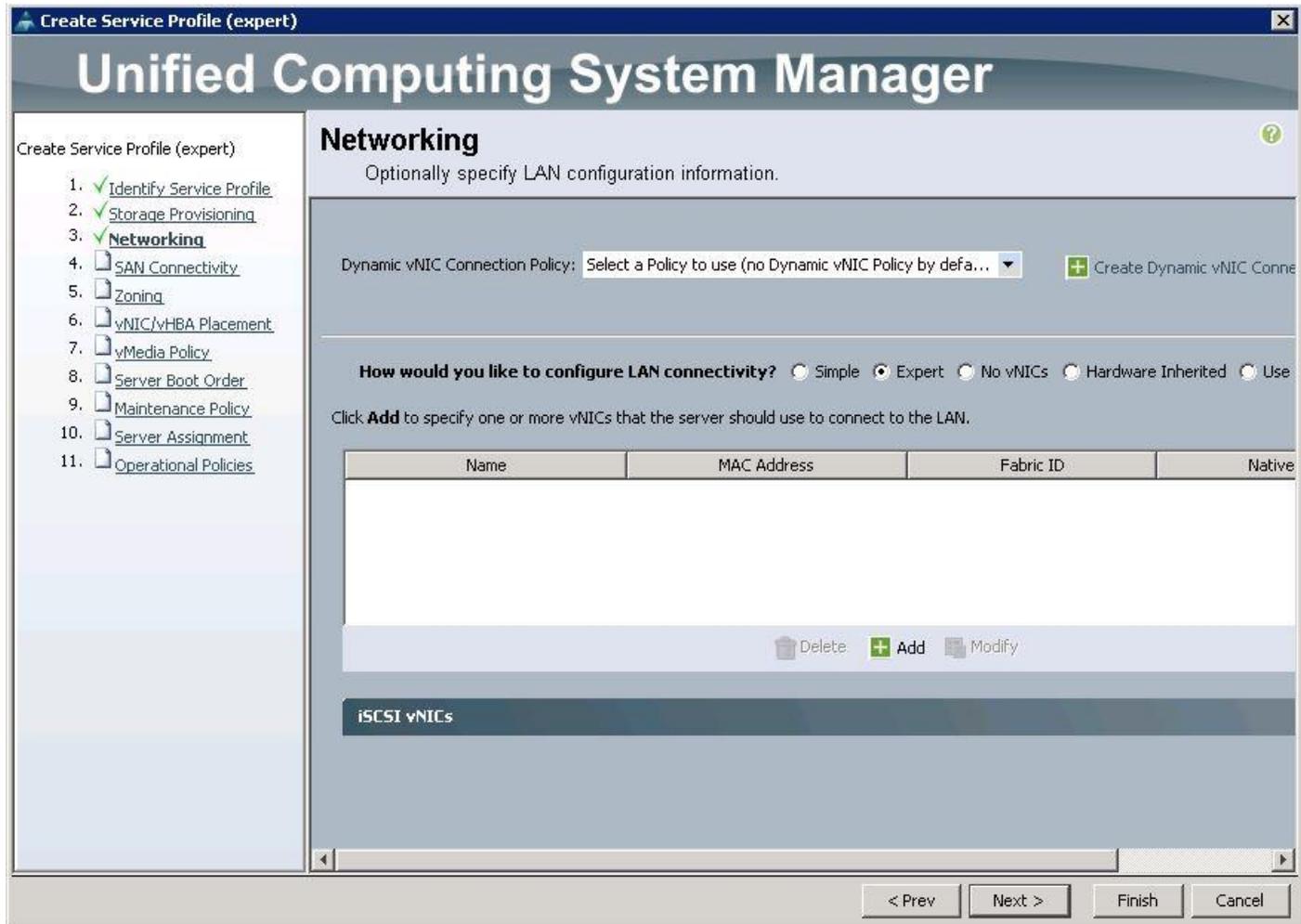
2. Specify the Service profile name for Undercloud node as OSP7-Director. Choose the UUID pools previously created from the drop-down list and click Next.



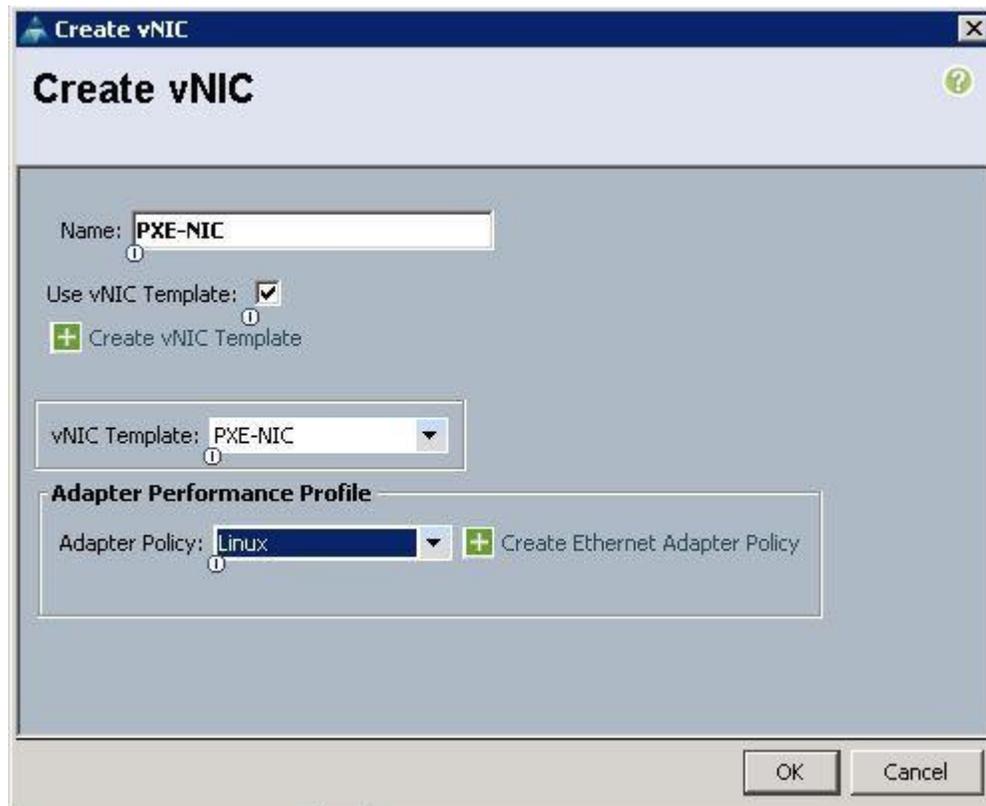
- For Storage Provisioning, choose Expert and click Storage profile Policy and choose the Storage profile Blade-OS-boot previously created from the drop-down list and click Next.



- For Networking, choose Expert and click "+".



5. Create the vNIC interface for PXE or Provisioning network as PXE-NIC and click the check box Use vNIC template.
6. Under vNIC template, choose the PXE-NIC template previously created from the drop-down list and choose Linux for the Adapter Policy.



7. Create the vNIC interface for Internal API network as Internal-API and from the drop-down list choose MAC pools created before. Then click on "Fabric B" and check the "Enable Failover".
8. Under VLANs , Select "Internal-API network" as Native VLAN, then choose Adapter Policy as "Linux" and Network Controller Policy as "Enable\_CDP"

**Create vNIC**

Name: **Internal-API**

Use vNIC Template:

**MAC Address**

MAC Address Assignment: **UCS-Blade-MAC\_Pools(94/100)**

**+** Create MAC Pool

The MAC address will be automatically assigned from the selected pool.

Fabric ID:  Fabric A  **Fabric B**  **Enable Failover**

VLAN in LAN cloud will take the precedence over the Appliance Cloud when there is a name clash.

**VLANs**

Filter Export Print

Select	Name	Native VLAN
<input type="checkbox"/>	default	<input type="radio"/>
<input type="checkbox"/>	default	<input type="radio"/>
<input type="checkbox"/>	External	<input type="radio"/>
<input checked="" type="checkbox"/>	<b>Internal-API</b>	<input checked="" type="radio"/>
<input type="checkbox"/>	OS-276	<input type="radio"/>
<input type="checkbox"/>	OS-293	<input type="radio"/>

**+** Create VLAN

**Adapter Performance Profile**

Adapter Policy: **Linux** **+** Create Ethernet Adapter Policy

QoS Policy: <not set> **+** Create QoS Policy

Network Control Policy: **Enable\_CDP** **+** Create Network Control Policy

OK Cancel

9. Create the the VNIC interface for External network as External-NIC and from the drop-down list choose MAC pools created before. Then click on “Fabric B” and check the “Enable Failover”.
10. Under VLANs , Select “External” as Native VLAN, then choose Adapter Policy as “Linux” and Network Controller Policy as “Enable\_CDP”

**Create vNIC**

Name: **External-NIC**

Use vNIC Template:

**MAC Address**

MAC Address Assignment: **UCS-Blade-MAC\_Pools(94/100)**

**+ Create MAC Pool**

The MAC address will be automatically assigned from the selected pool.

**+ Create vNIC Template**

Fabric ID:  Fabric A  **Fabric B**  **Enable Failover**

VLAN in LAN cloud will take the precedence over the Appliance Cloud when there is a name clash.

**VLANs**

Filter Export Print

Select	Name	Native VLAN
<input type="checkbox"/>	default	<input type="radio"/>
<input checked="" type="checkbox"/>	<b>External</b>	<input checked="" type="radio"/>
<input type="checkbox"/>	Internal-API	<input type="radio"/>
<input type="checkbox"/>	OS-276	<input type="radio"/>
<input type="checkbox"/>	OS-293	<input type="radio"/>
<input type="checkbox"/>	OSP-PXE	<input type="radio"/>

**+ Create VLAN**

**Adapter Performance Profile**

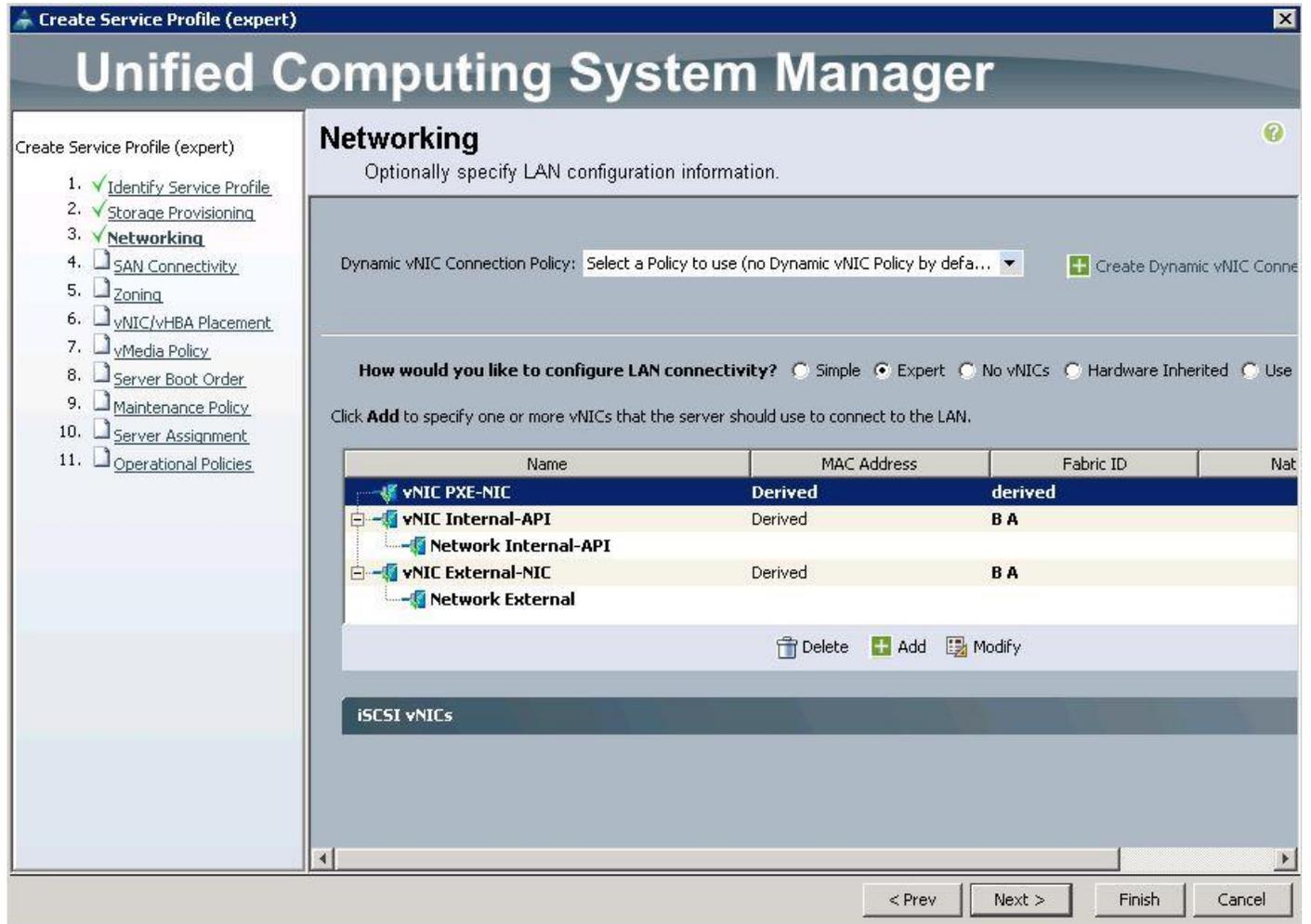
Adapter Policy: **Linux** **+ Create Ethernet Adapter Policy**

QoS Policy: <not set> **+ Create QoS Policy**

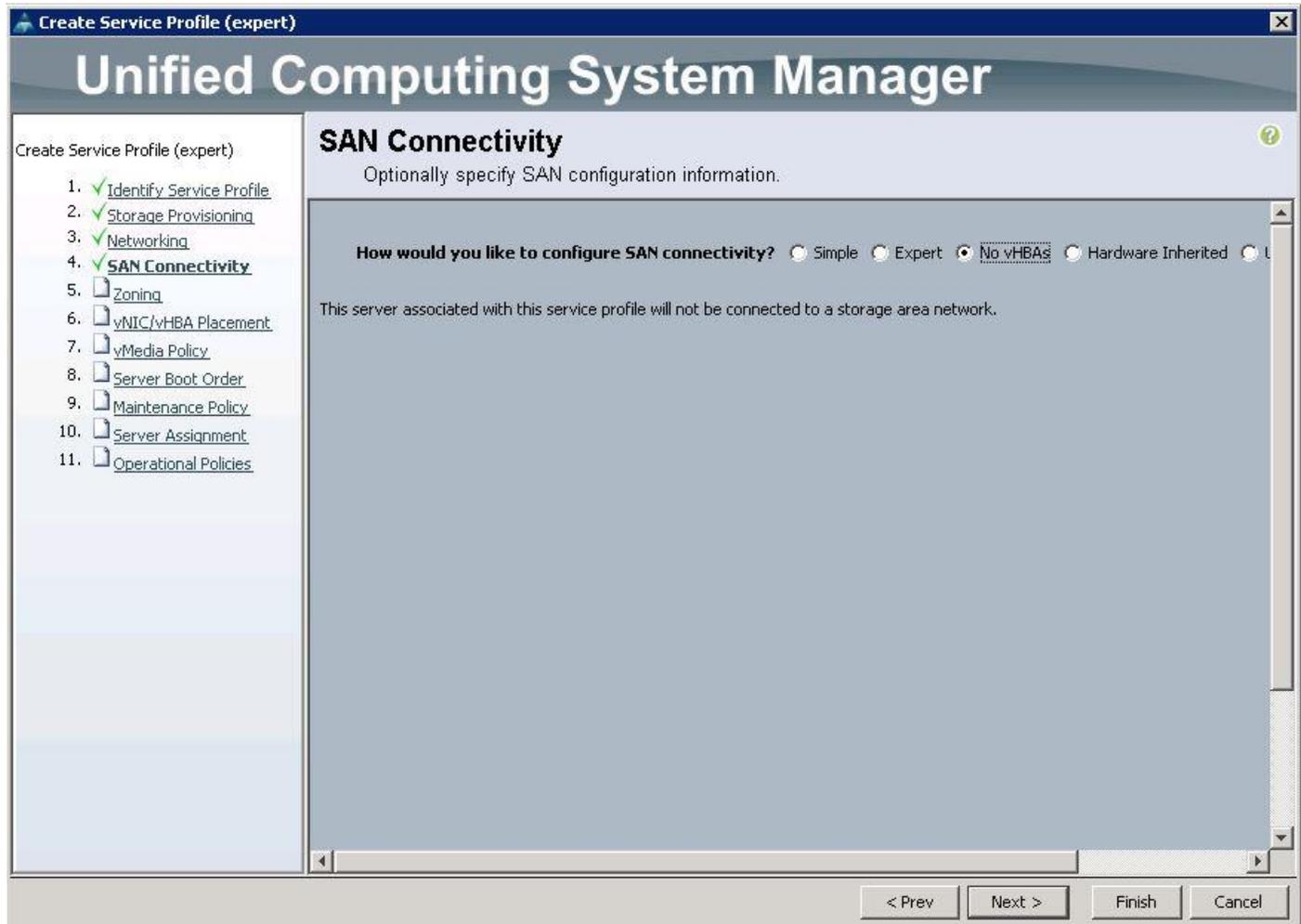
Network Control Policy: **Enable\_CDP** **+ Create Network Control Policy**

OK Cancel

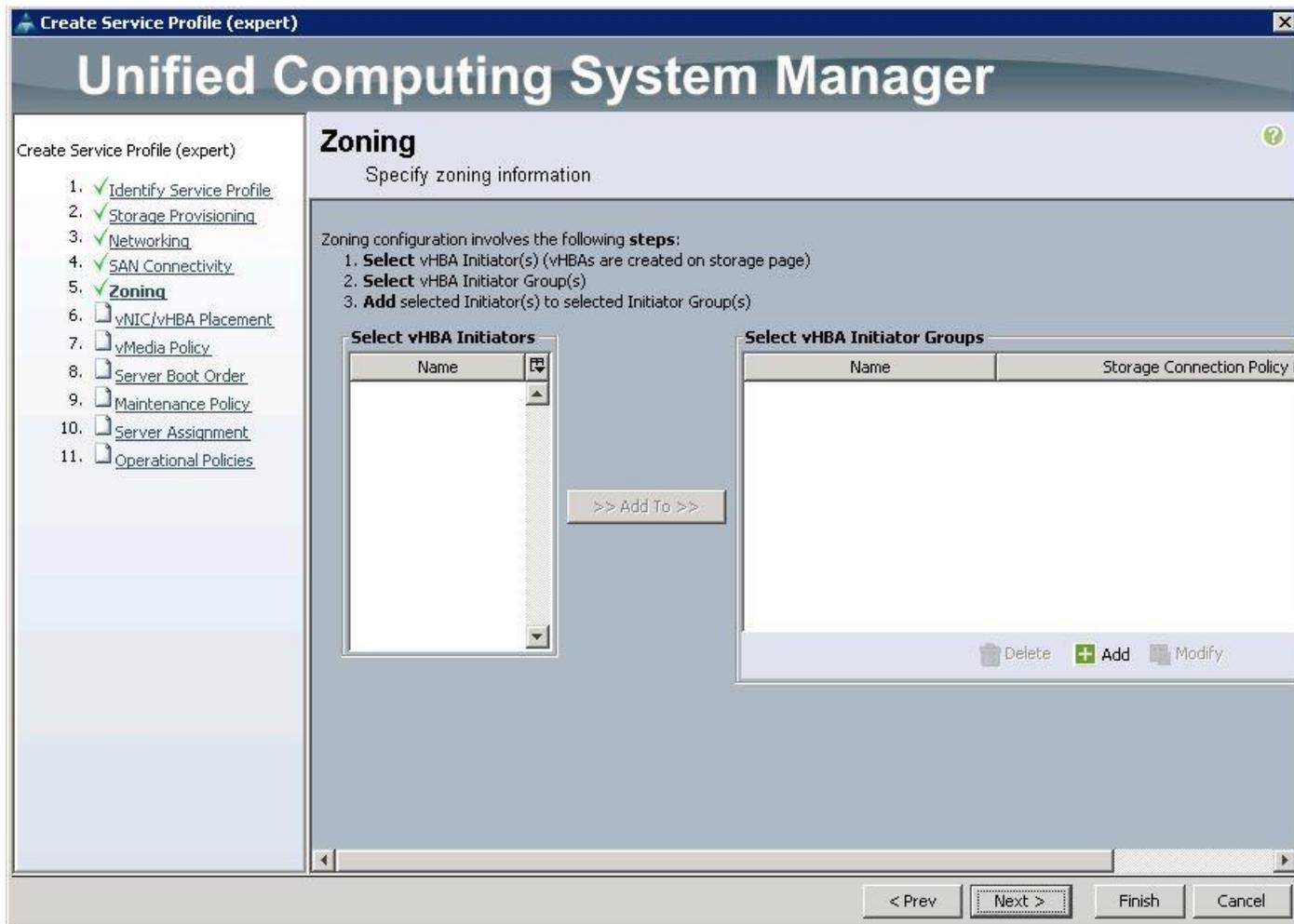
11. After a successful vNIC creation, click Next.



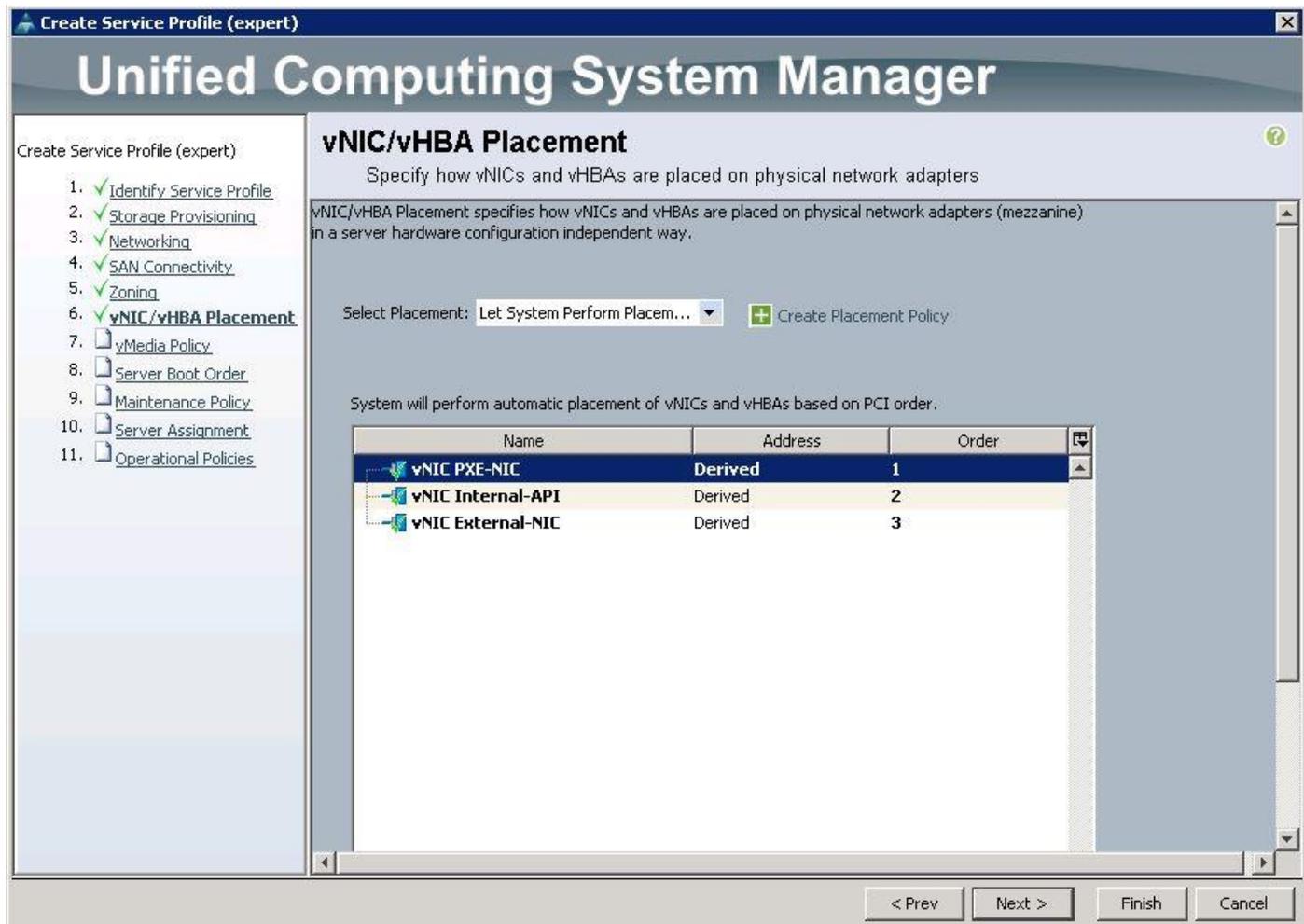
12. Under the SAN connectivity, choose No VHBA's and click Next.



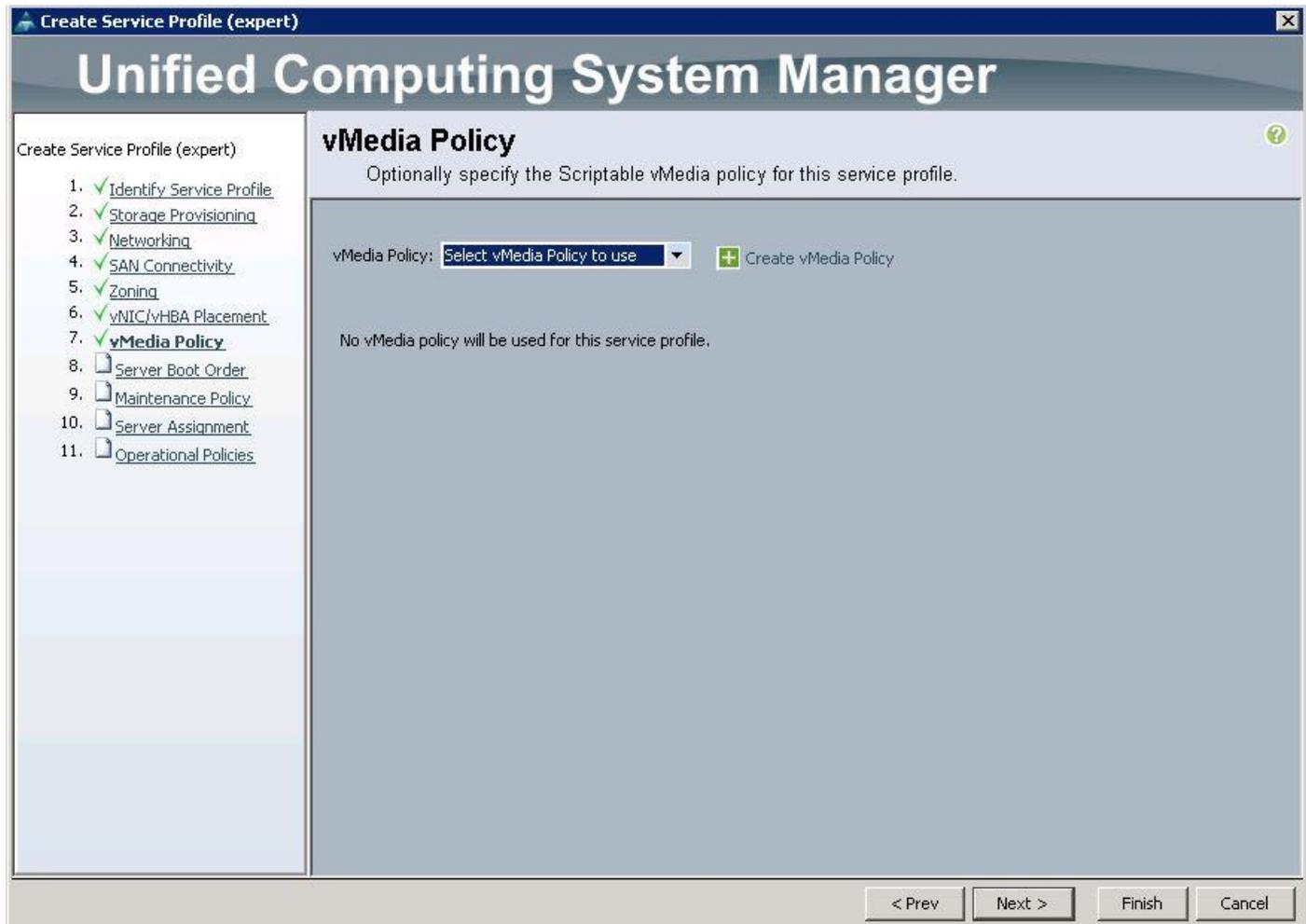
13. Under Zoning, click Next.



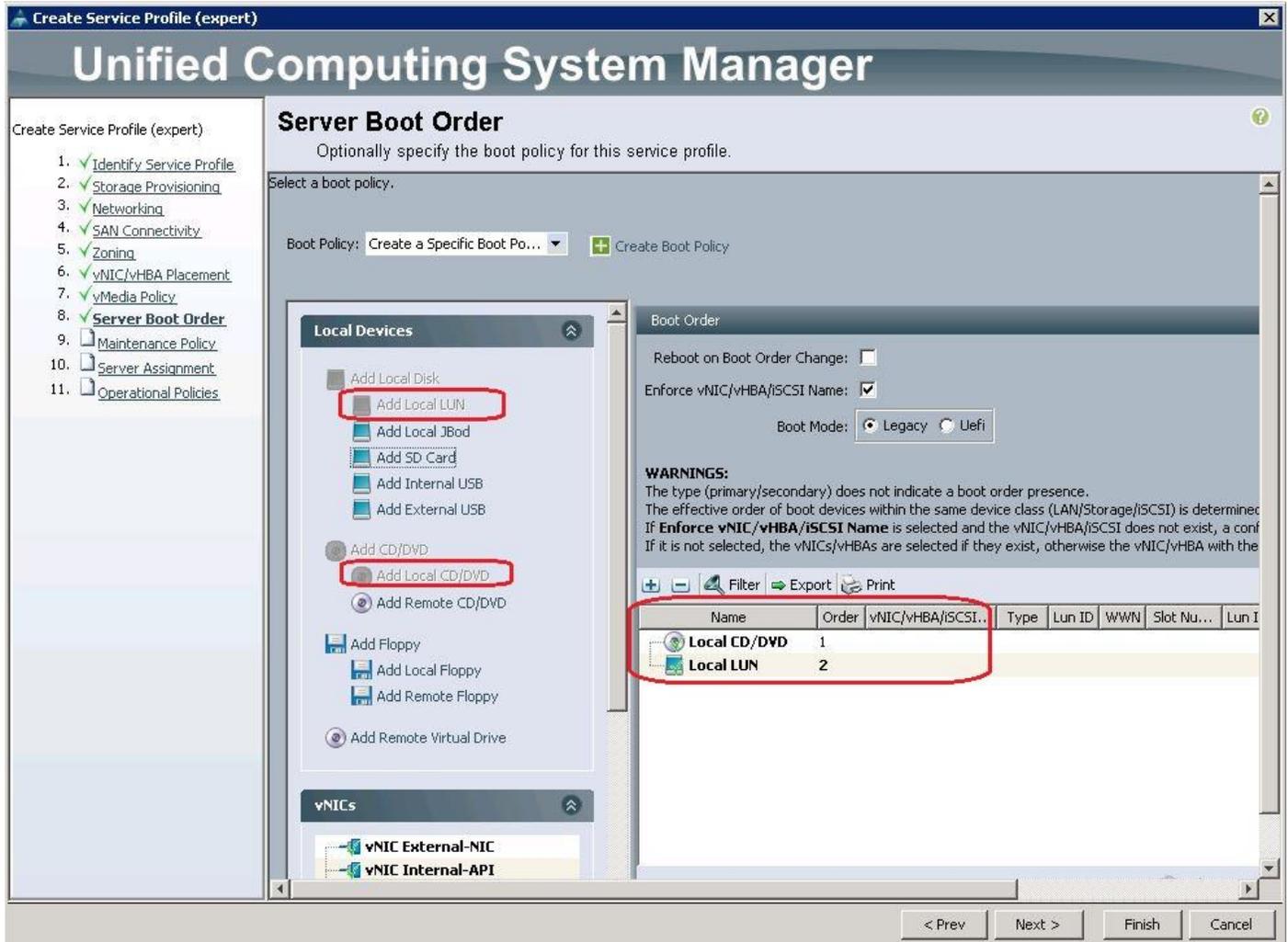
14. Under vNIC/vHBA Placement, choose the vNICs PCI order as shown below and click Next.



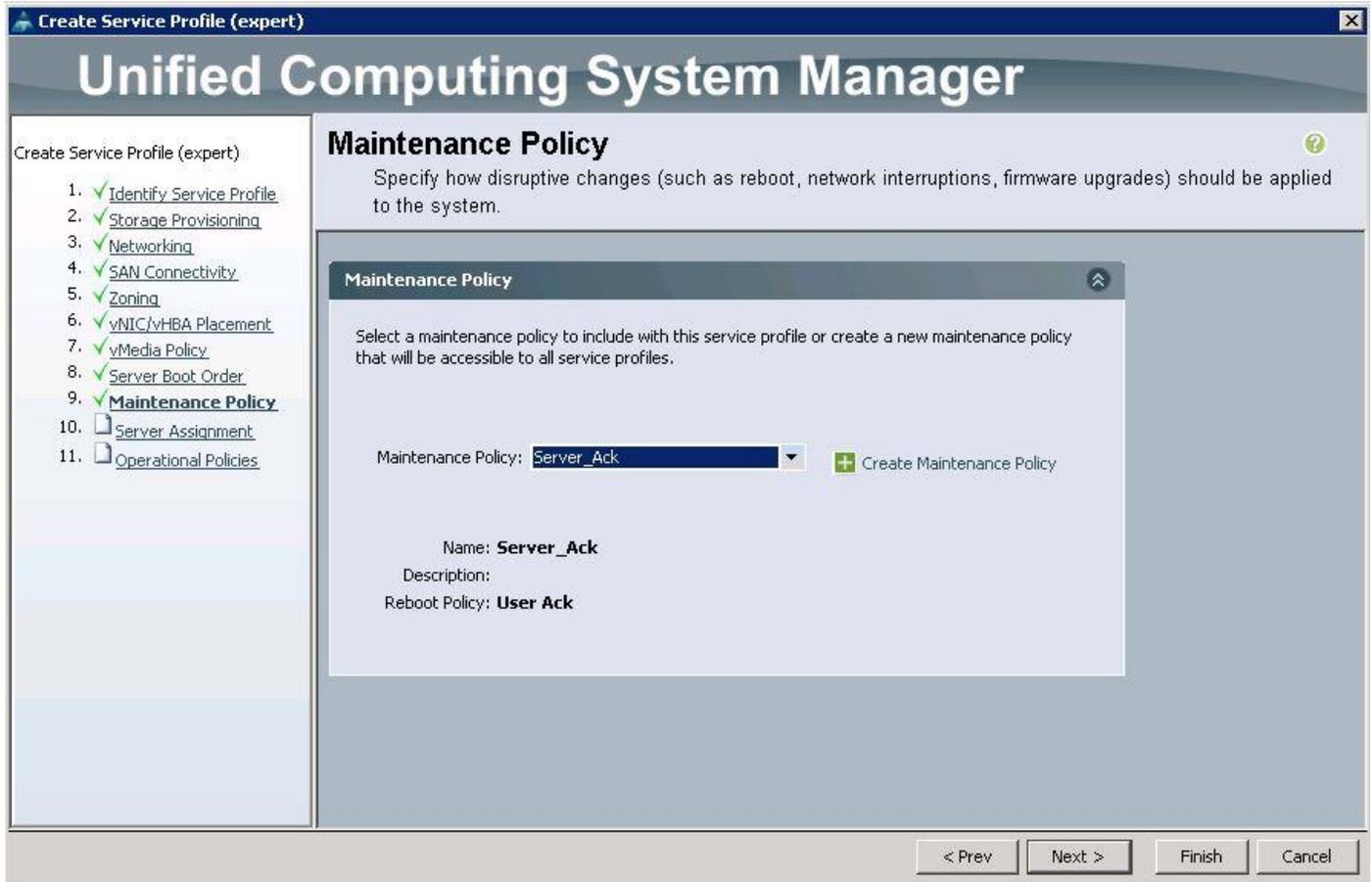
15. Under vMedia Policy, click Next.



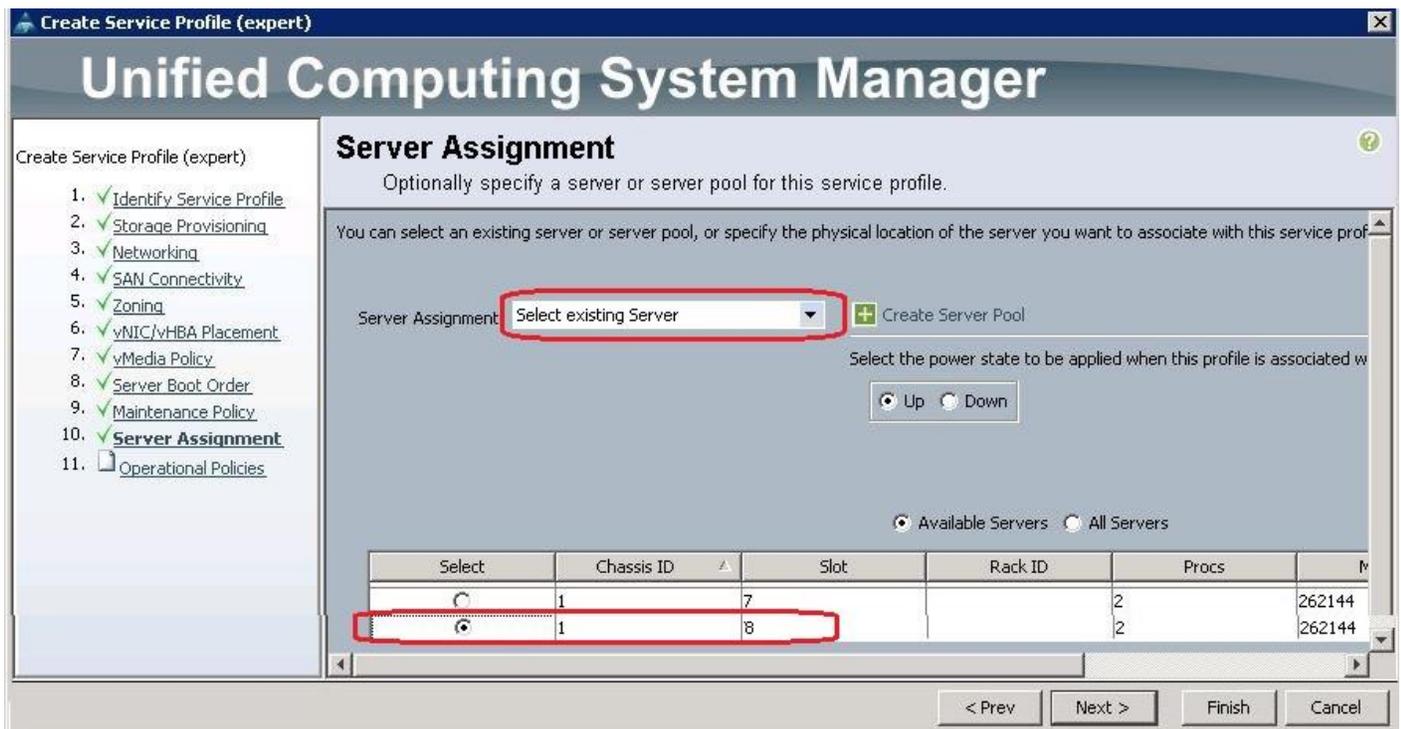
16. Under **Server Boot Order**, choose the boot policy as “Create a Specific Boot Policy”, from the drop-down list and click Next. Make sure you select “ local CD/DVD” as first boot order and “ local LUN” as second boot order and click Next.



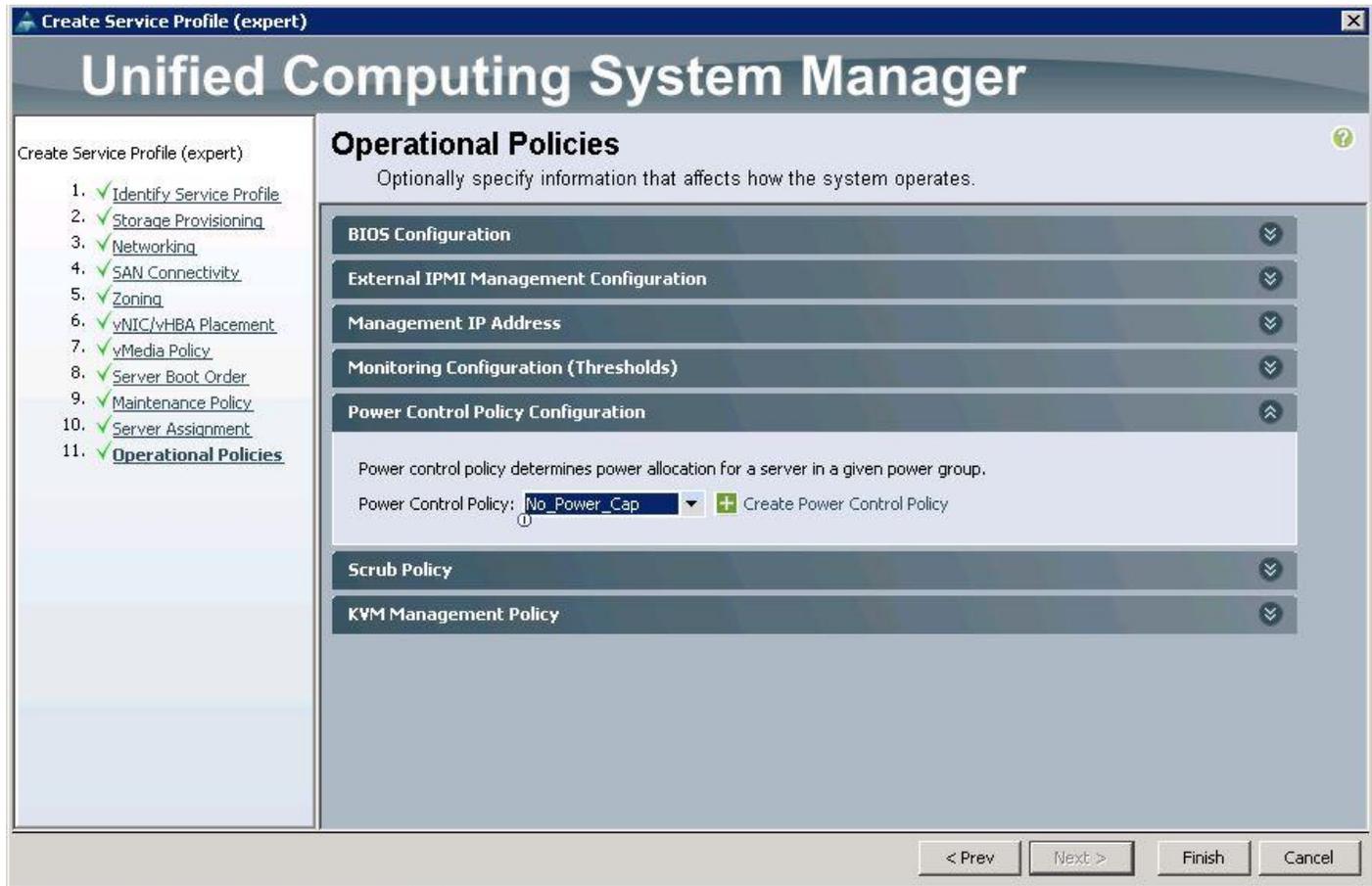
17. Under Maintenance Policy, choose Server\_Ack previously created, from the drop-down list and click Next.



18. Under Server Assignment, choose “Select existing server” and select the respective blade assigned for Director node and click Next.



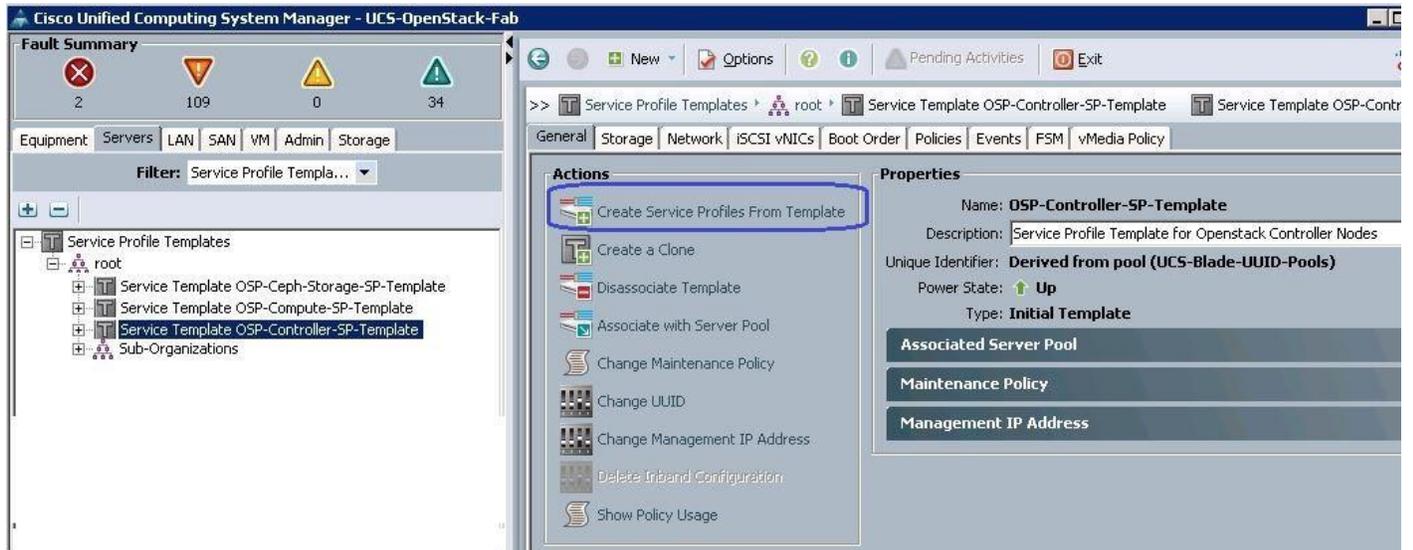
- Under Operational Policies, choose the Power Control Policy as “No\_Power\_Cap” and click Finish.



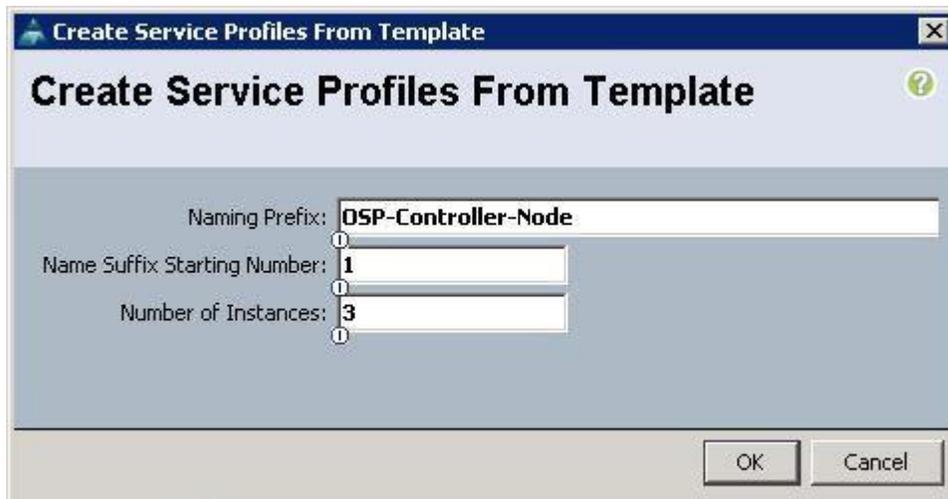
## Create Service Profiles for Controller Nodes

To create Service profiles for Controller nodes, complete the following steps:

- Under Servers → Service Profile Templates → root → select the Controller Service profile template and click Create Service Profiles from Templates.

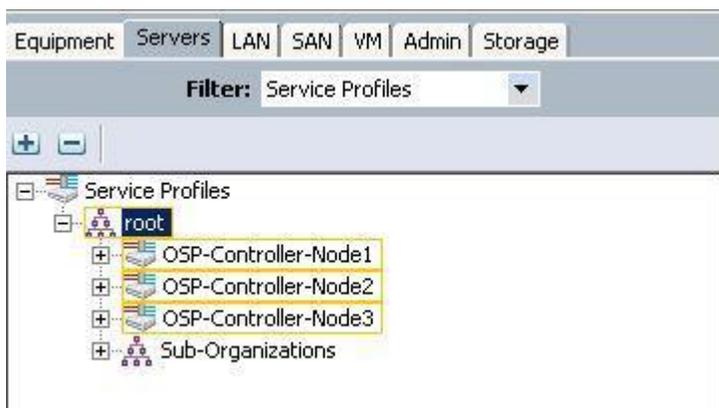


a. Specify the Service profile name and the number of instances as 3 for the Controller nodes.



b. Make sure the Service profiles for the Controller nodes have been created.

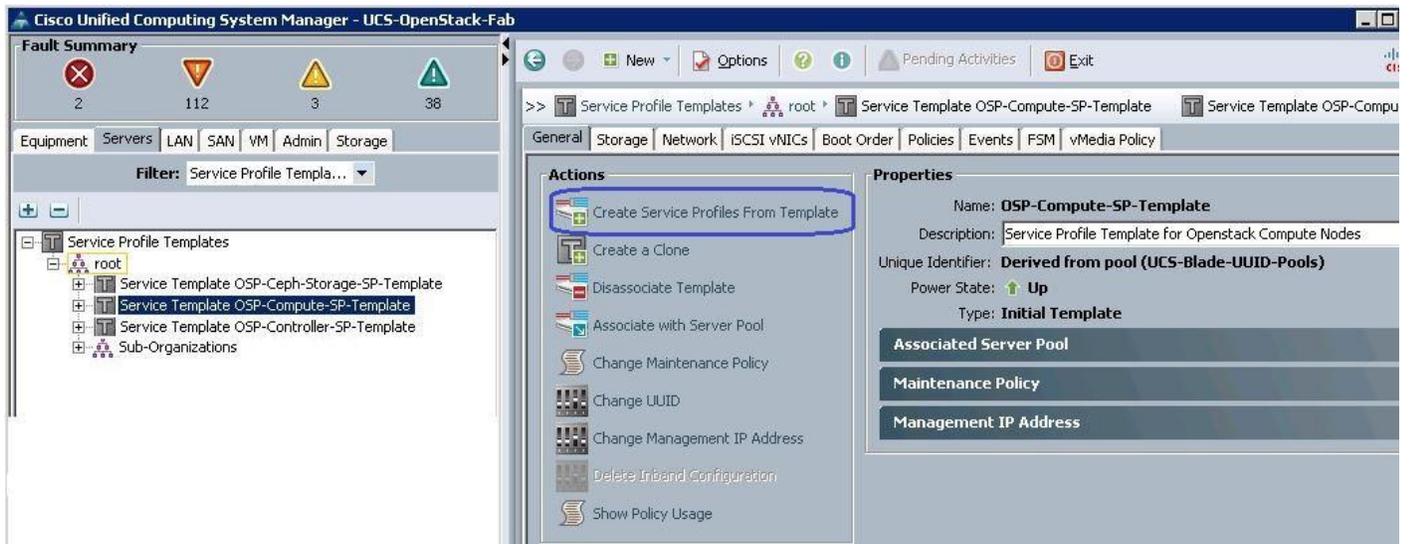
c. Under Servers → Service profiles → root .



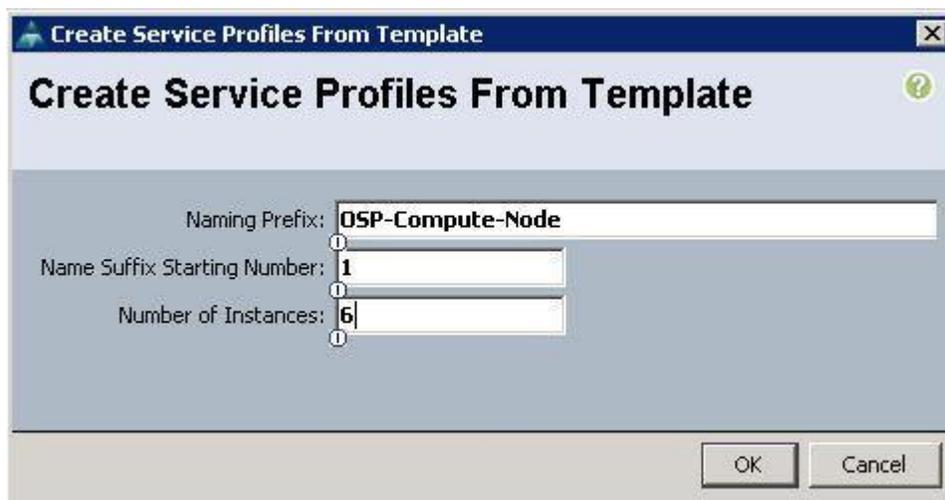
## Create Service Profiles for Compute Nodes

To create Service profiles for Compute nodes, complete the following steps:

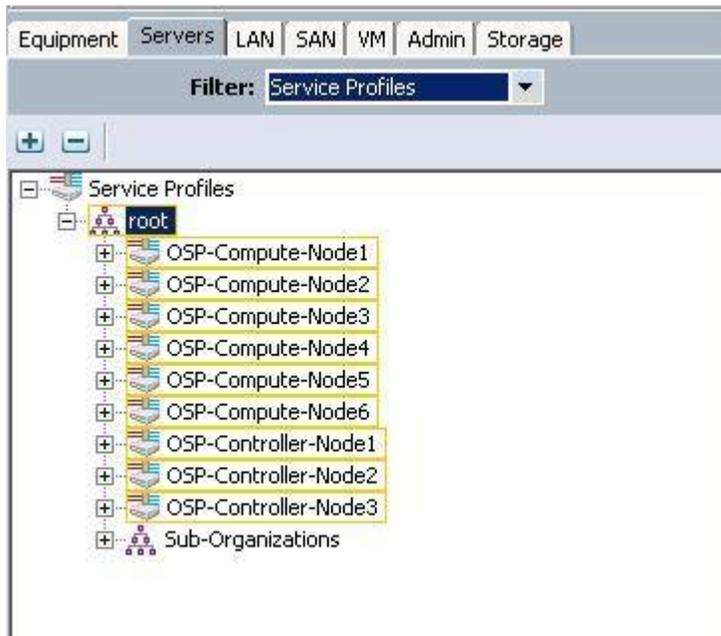
1. Under Servers → Service Profile Templates → root → select the Compute Service profile template and click Create Service Profiles from Templates.



- a. Specify the profile name and set the number of instances to 6 for compute nodes.



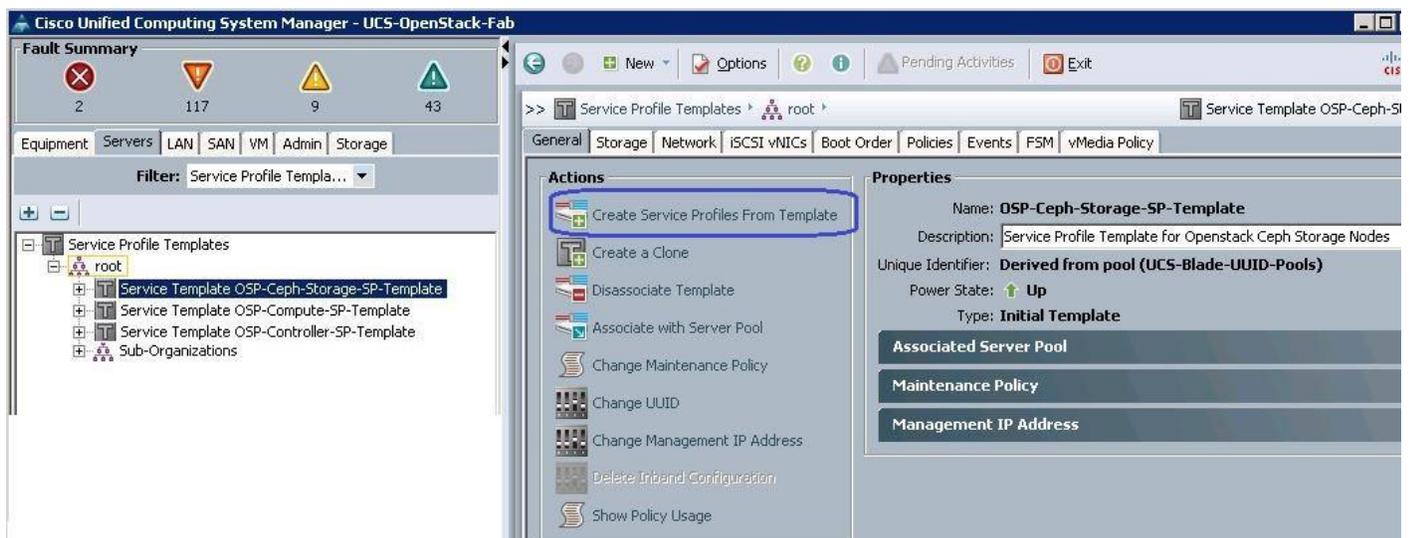
- b. Make sure the Service profiles for the Compute nodes have been created.



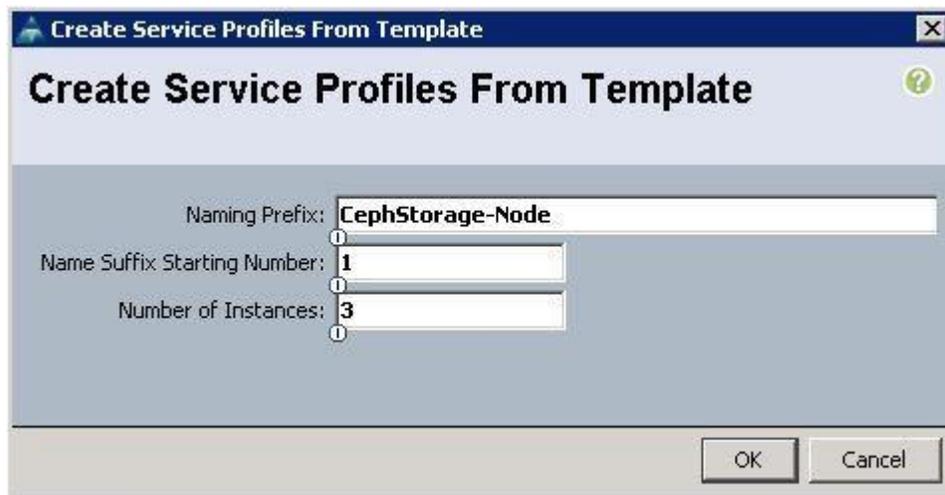
### Create Service Profiles for Ceph Storage Nodes

To create Service profiles for Ceph Storage nodes, complete the following steps:

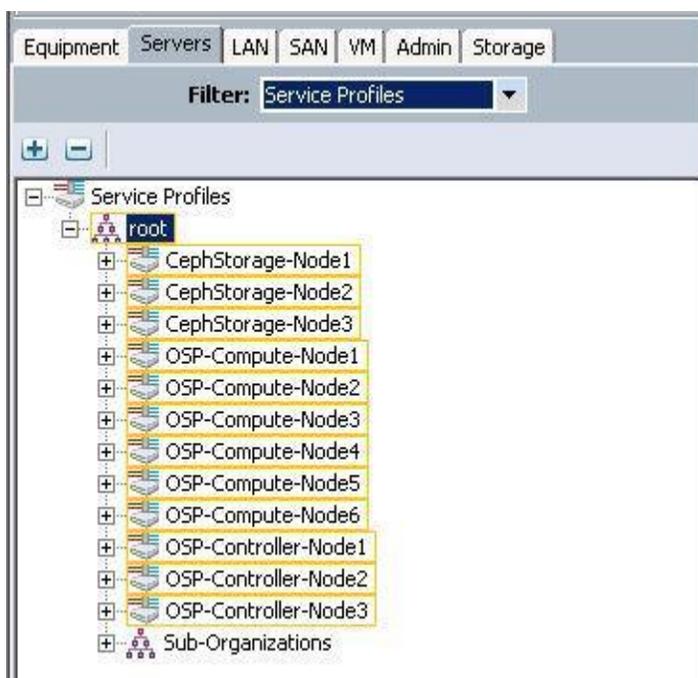
1. Under Servers → Service Profile Templates → root → select the Ceph Storage Service profile template and click Create Service Profiles from Templates.



- a. Specify the Service profile name and set the number of instances to 3 for the Ceph Storage nodes.



b. Make sure the Service profiles for the Ceph Storage nodes have been created.



c. Verify the Service profile association with the respective UCS Servers.

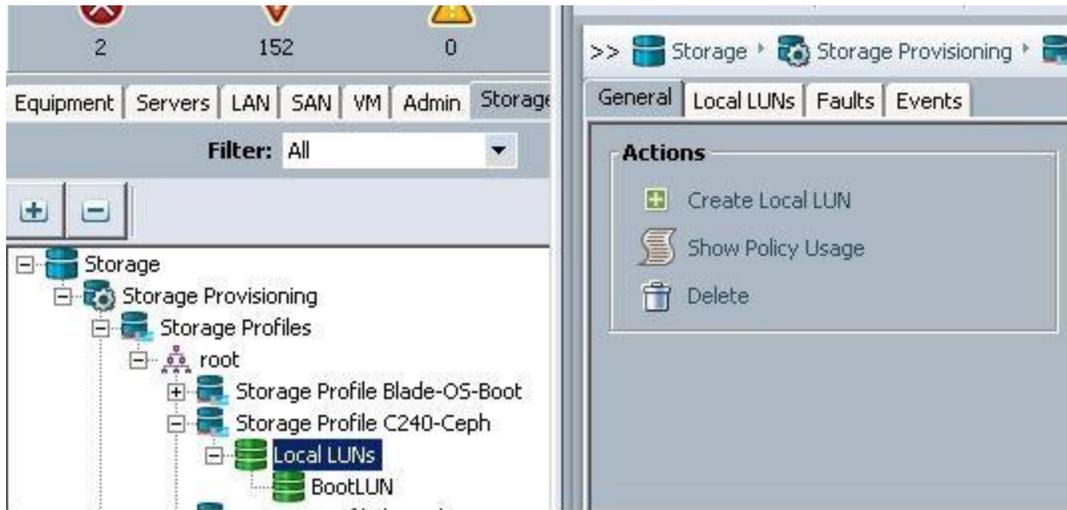
## Create LUNs for the Ceph OSD and Journal Disks

After a successful CephStorage Server association, create the remaining LUNs for the Ceph OSD disks and Journal disks.

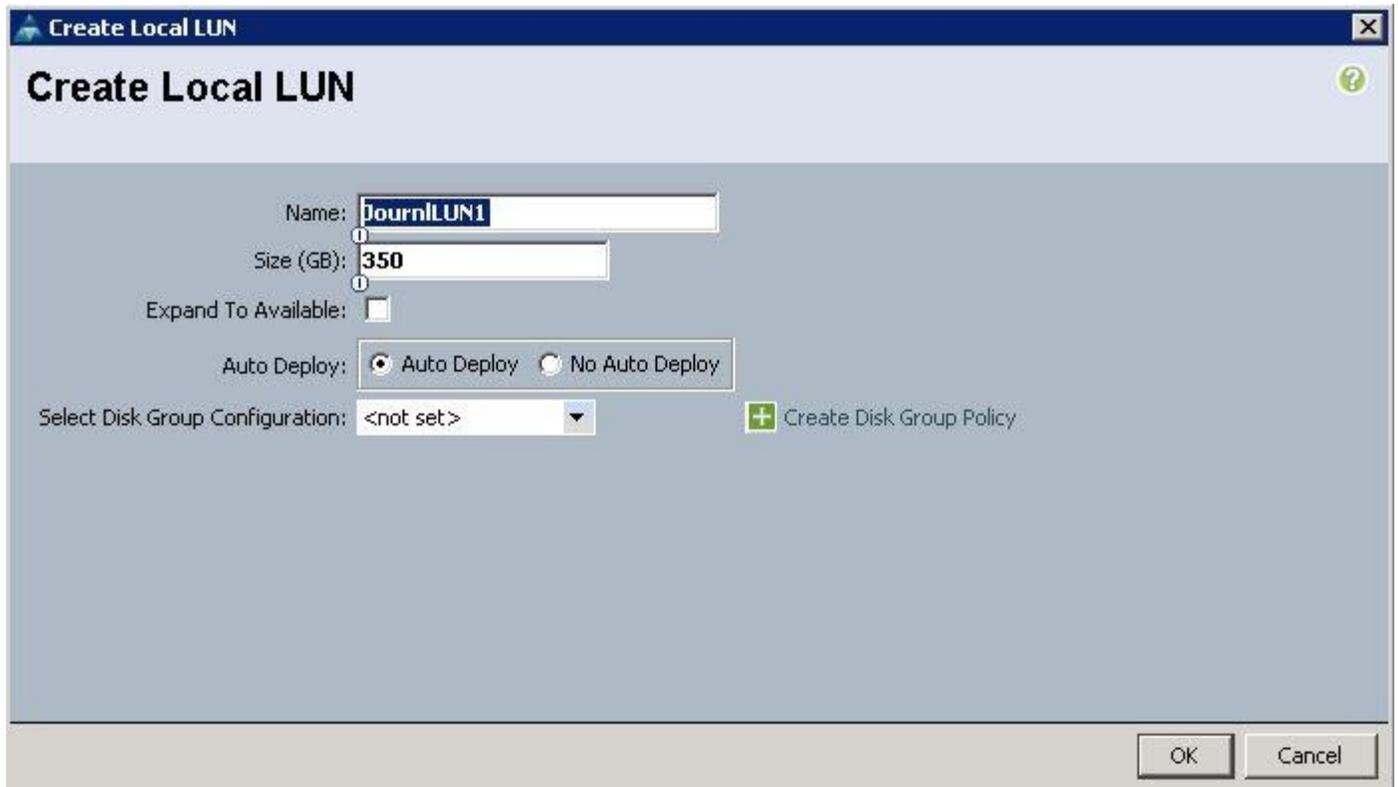
### Create the Ceph Journal LUN

To create the Ceph Journal LUNs, complete the following steps:

1. Under Storage → Storage Provisioning → root → select the previously created Ceph Storage profile C240-Ceph → click Local LUNs → click Create Local LUN.



- a. Specify the name as JournLUN1 and set the size in GB to 350 for the 400GB SSD disks and click Create Disk Group Policy.



- b. Specify the Disk group policy name and choose the RAID level as RAID 0 and select Disk Group Configuration (Manual).

**Create Disk Group Policy**

Name:

Description:

RAID Level:

Disk Group Configuration (Automatic)  Disk Group Configuration (Manual)

**Disk Group Configuration (Manual)**

Filter Export Print

Slot Number	Role	Span ID

**Virtual Drive Configuration**

Strip Size (KB):

Access Policy:  Platform Default  Read Write  Read Only  Blocked

Read Policy:  Platform Default  Read Ahead  Normal

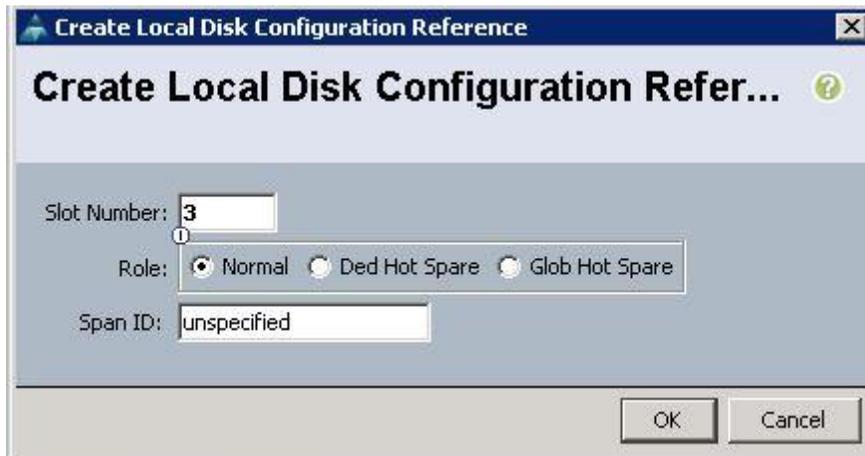
Write Cache Policy:  Platform Default  Write Through  Write Back Good Bbu  Always Write Back

IO Policy:  Platform Default  Direct  Cached

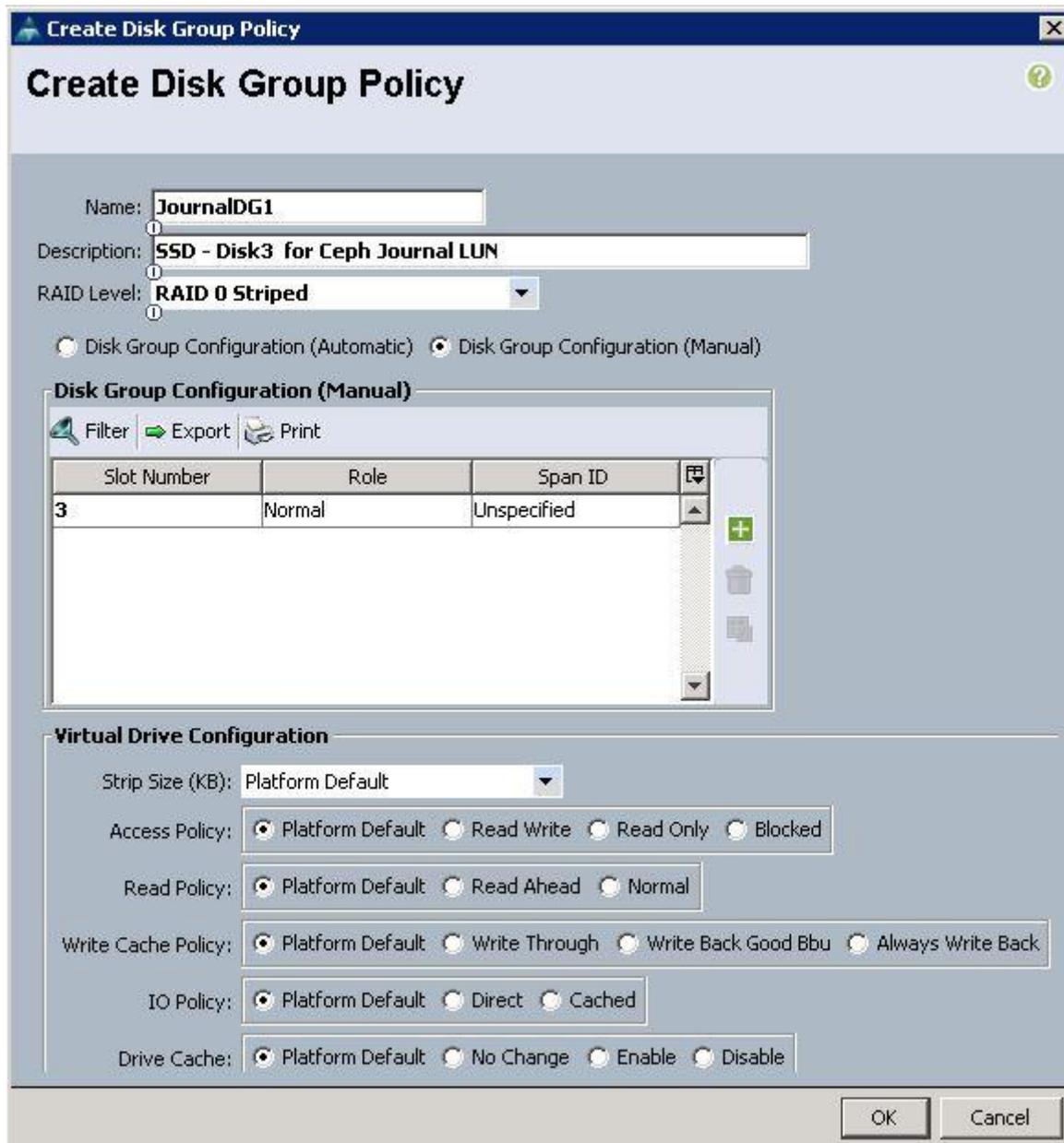
Drive Cache:  Platform Default  No Change  Enable  Disable

OK Cancel

- c. Specify the Slot ID as 3, which is the physical disk slot number for 400GB SSDs for the Journal LUN1 and click OK.



d. Click OK to confirm the Disk group policy creation.



**Create Local LUN**

Name:

Size (GB):

Expand To Available:

Auto Deploy:  Auto Deploy  No Auto Deploy

Select Disk Group Configuration:  + Create Disk Group Polic

OK Cancel

- e. From the drop-down list, choose the Disk group policy for the Journal LUN as JournalDG1.
- f. Create the Local LUN as JournLUN2 with Disk group policy as JournalDG2 using 400GB SSD on Disk Slot4.

#### Create the Ceph OSD LUN

To create the Ceph OSD LUN, complete the following steps:

1. Under Storage → Storage Provisioning → root → select the previously created Ceph Storage profile C240-Ceph → click Local LUNs → click Create Local LUN.
  - a. Specify the name as OSDLUN1 and the size in GB as 5500 for the 6TB SAS disks and click Create Disk Group Policy.

**Create Local LUN**

Name:

Size (GB):

Expand To Available:

Auto Deploy:  Auto Deploy  No Auto Deploy

Select Disk Group Configuration:  + Create Disk Group Policy

OK Cancel

- b. Specify Disk group policy name and Choose RAID level as RAID 0 and select Disk Group Configuration(Manual)
- c. Click “+” and Specify Slot ID as 5, which is physical disk slot number for 6TB SAS disks for Ceph OSD LUN1 and click OK.

**Create Disk Group Policy**

Name:

Description:

RAID Level:

Disk Group Configuration (Automatic)  Disk Group Configuration (Manual)

**Disk Group Configuration (Manual)**

Filter Export Print

Slot Number	Role	Span ID
5	Normal	Unspecified

**Virtual Drive Configuration**

Strip Size (KB):

Access Policy:  Platform Default  Read Write  Read Only  Blocked

Read Policy:  Platform Default  Read Ahead  Normal

Write Cache Policy:  Platform Default  Write Through  Write Back Good Bbu  Always Write Back

IO Policy:  Platform Default  Direct  Cached

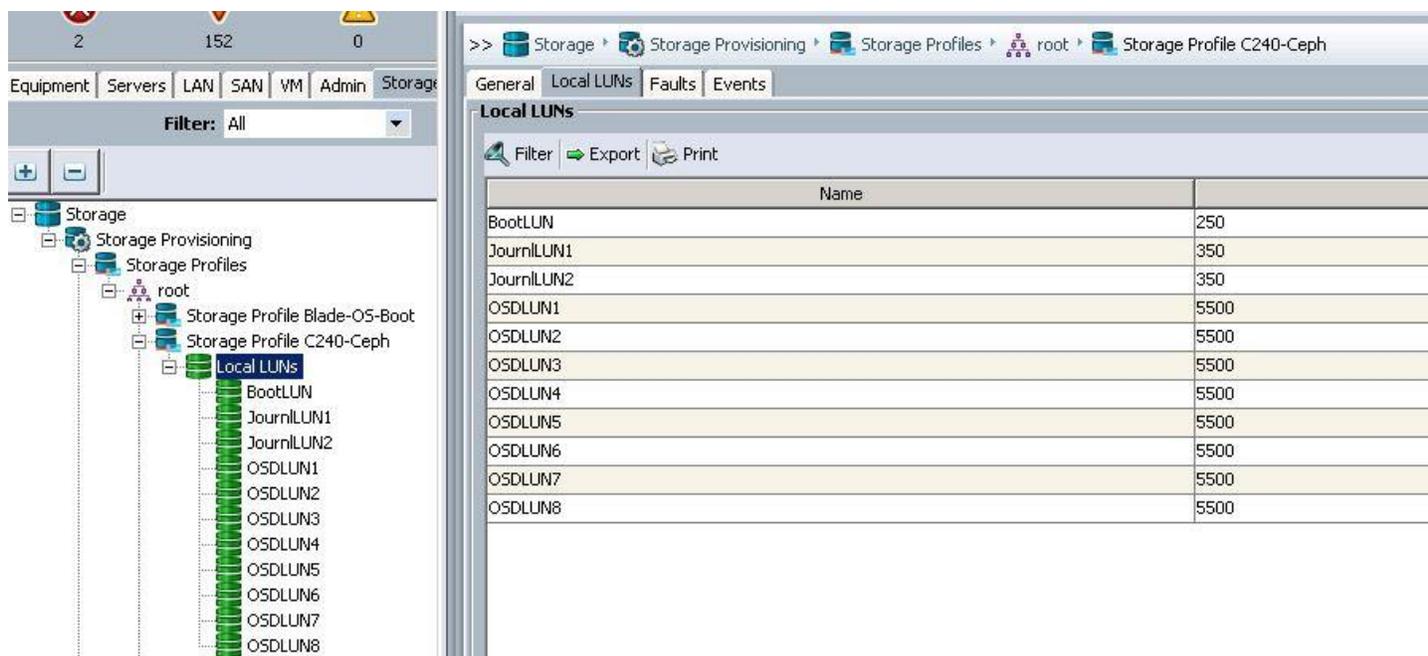
Drive Cache:  Platform Default  No Change  Enable  Disable

OK Cancel

- d. From the drop-down list, choose the Disk group policy for OSDLUN1 as OSD-DG1.



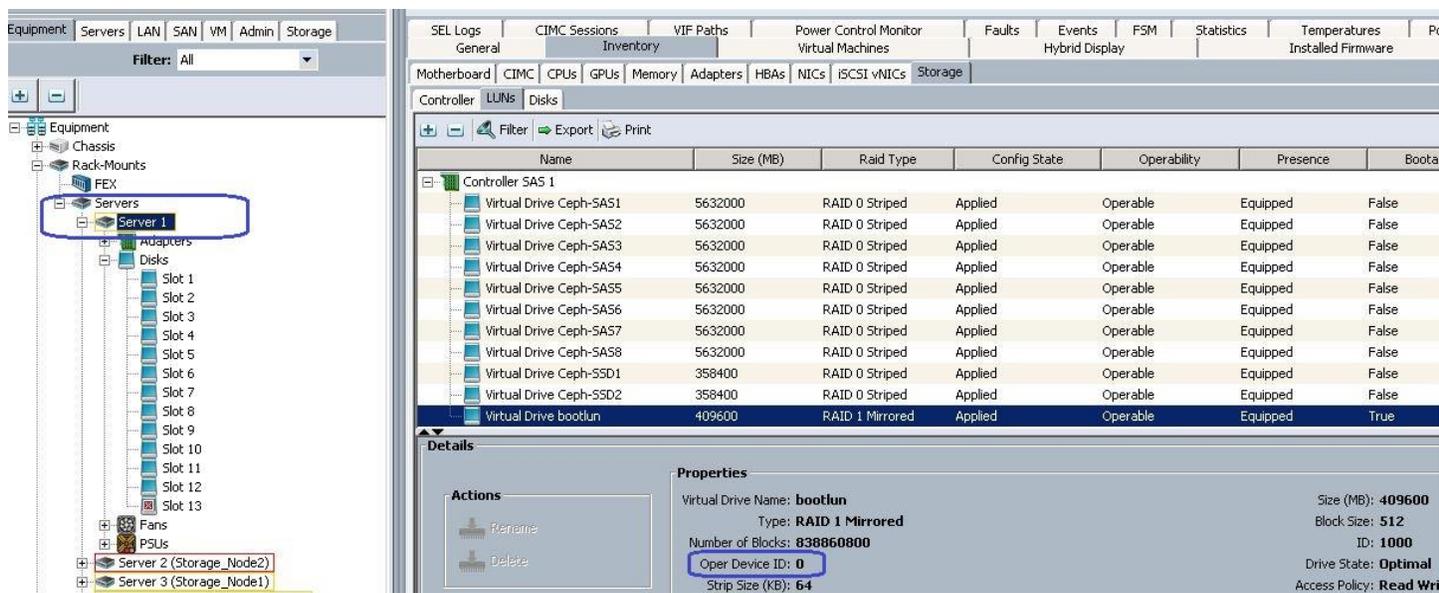
- e. Create the remaining OSDLUN3, 4, 5, 6, 7, 8 with the Disk group policy using 6TB SAS disks 6, 7, 8, 9, 10, 11, and 12.
- f. Make sure the LUNs for Journals and OSDs are created as shown below.



- g. Make sure all the Ceph Storage Servers have the identical LUN ID and Device ID for all the LUNs (OS-boot, Journal and OSD) as shown in the table below:

Physical Disk Slot	Disk Type	Disk Size	RAID Level	LUN Size	LUN ID	Device ID
Disk 1	SAS	300 GB	RAID 1	250 GB	1000	0
Disk 2	SAS	300 GB				

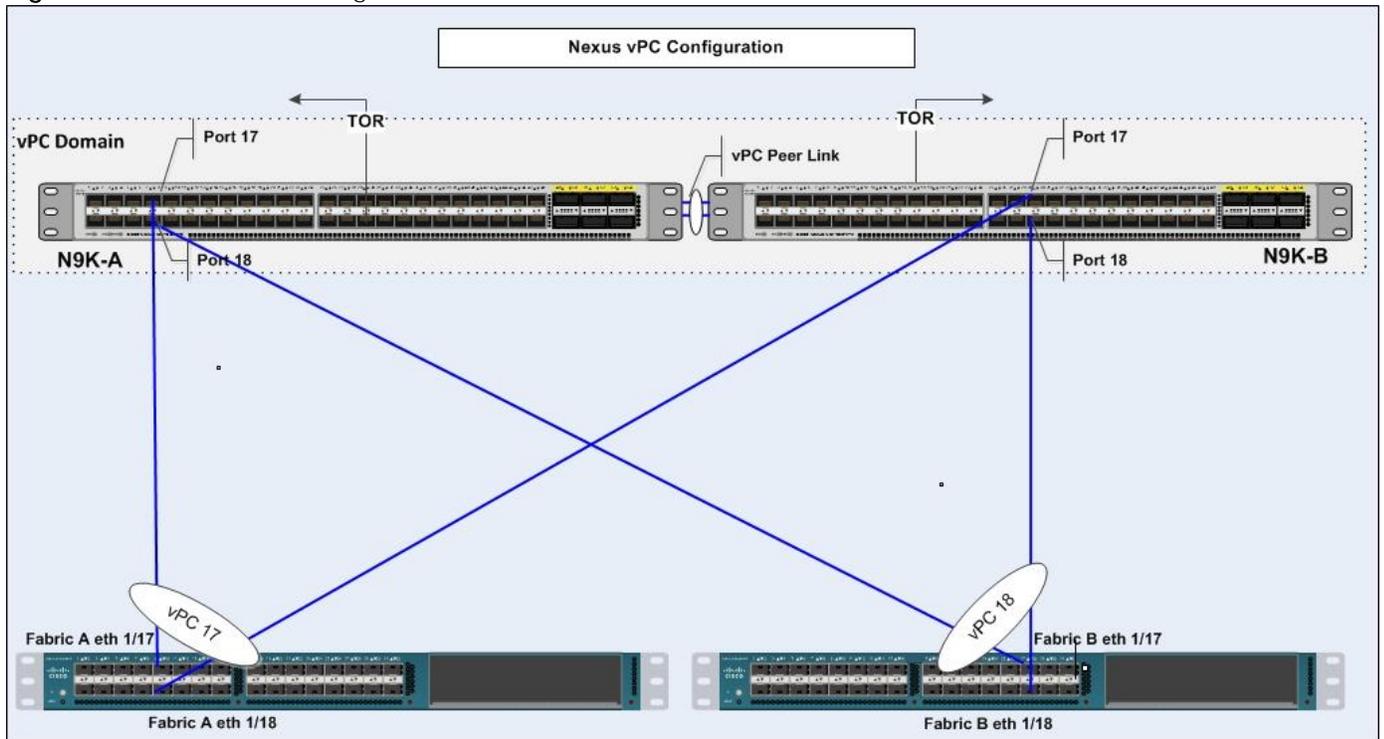
Physical Disk Slot	Disk Type	Disk Size	RAID Level	LUN Size	LUN ID	Device ID
Disk 3	SSD	400 GB	RAID 0	350 GB	1001	1
Disk 4	SSD	400 GB	RAID 0	350 GB	1002	2
Disk 5	SAS	6 TB	RAID 0	5500 GB	1003	3
Disk 6	SAS	6 TB	RAID 0	5500 GB	1004	4
Disk 7	SAS	6 TB	RAID 0	5500 GB	1005	5
Disk 8	SAS	6 TB	RAID 0	5500 GB	1006	6
Disk 9	SAS	6 TB	RAID 0	5500 GB	1007	7
Disk 10	SAS	6 TB	RAID 0	5500 GB	1008	8
Disk 11	SAS	6 TB	RAID 0	5500 GB	1009	9
Disk 12	SAS	6 TB	RAID 0	5500 GB	1010	10



## Create Port Channels for Cisco UCS Fabrics

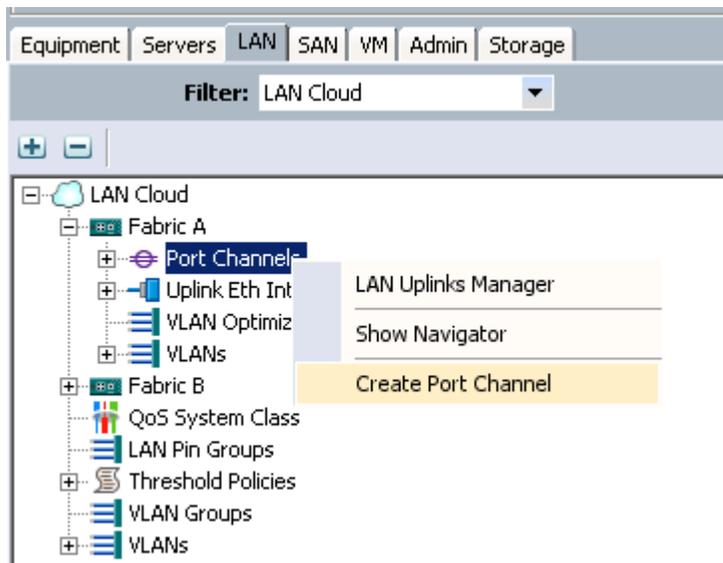
To create the Port Channels, complete the following steps as shown in the screenshots below. Figure 12 illustrates the configuration.

Figure 12 Nexus vPC Configuration

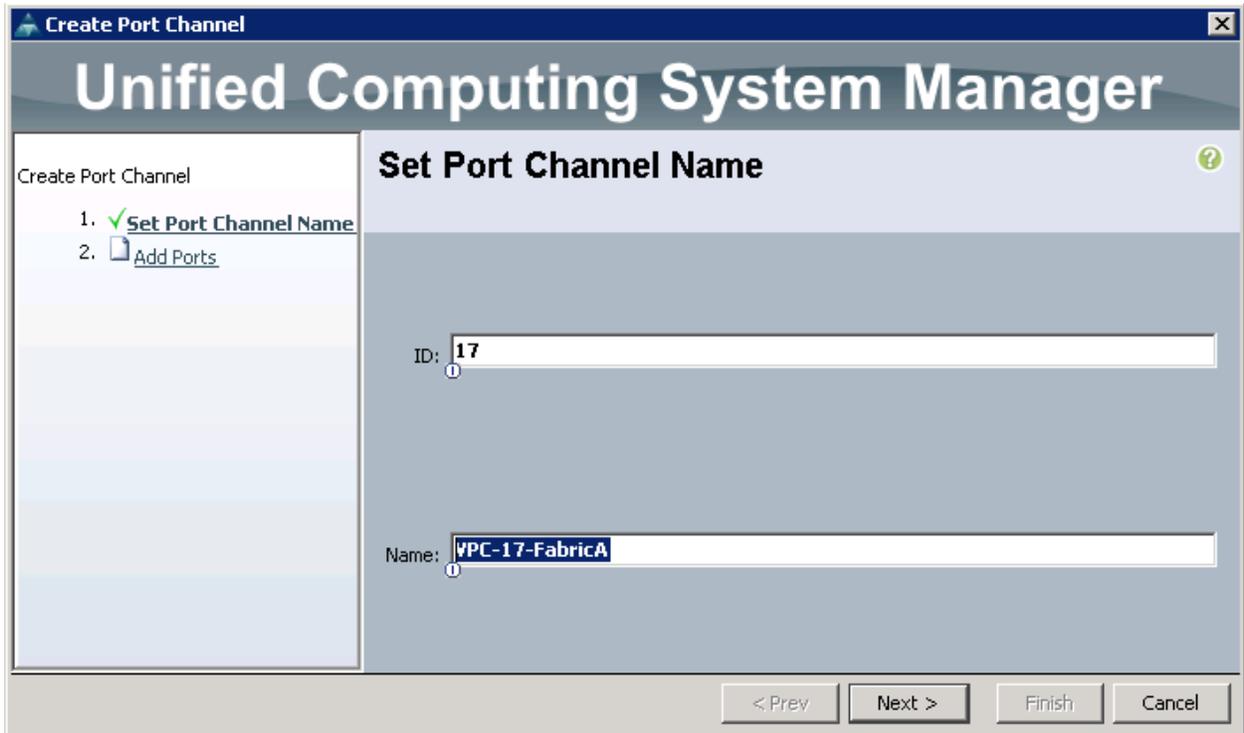


To create Port Channels from the UCS Manager GUI, complete the following steps:

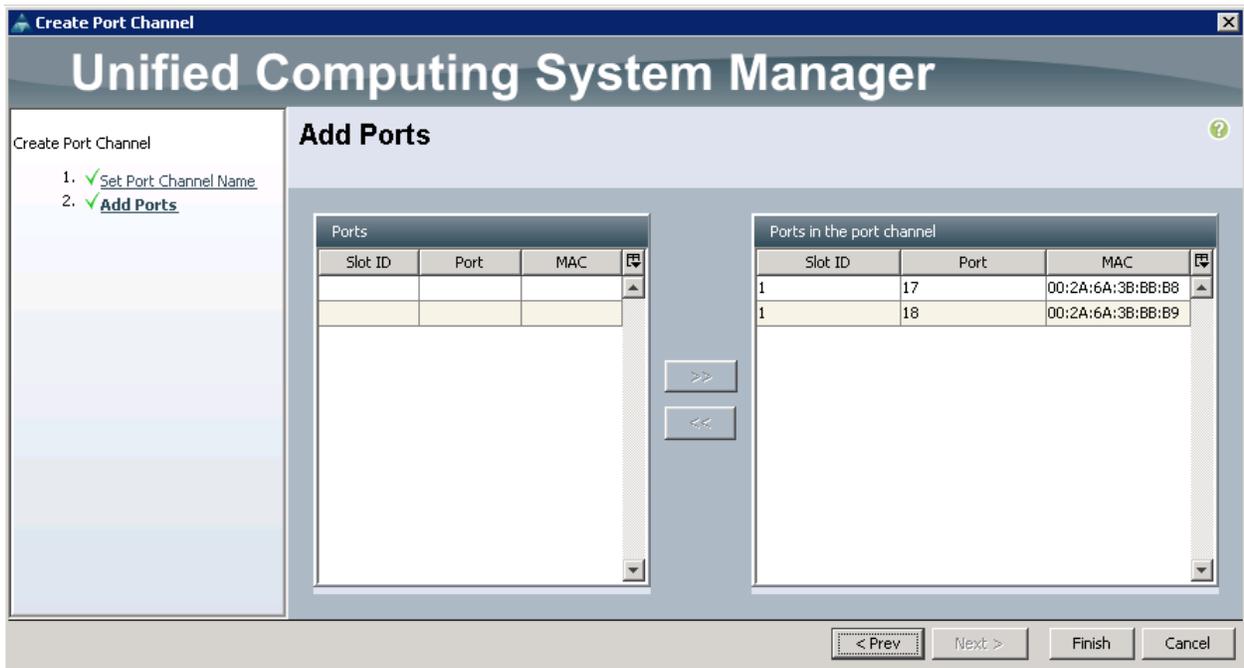
1. Under LAN → LAN Cloud → Fabric A → Port Channels → right-click and select Create Port Channel.



- a. Specify the ID and name for the port channel and click Next.



- b. Select the ports 17 and 18 from left pane and move to the right pane into Ports in the Port Channel and click Finish.



 Repeat the steps shown above on Fabric B with Port-Channel as 18.

## Cisco Nexus Configuration

### Configure the Cisco Nexus 9372 PX Switch A

To configure the Cisco Nexus 9372 PX Switch A, complete the following step:

1. Connect the console port to the Nexus 9372 PX switch designated for Fabric A:

```

---- Basic System Configuration Dialog VDC: 1 ----
This setup utility will guide you through the basic configuration of the system.
Setup configures only enough connectivity for management of the system.
*Note: setup is mainly used for configuring the system initially, when no
configuration is present. So setup always assumes system defaults and not the
current system configuration values.
Press Enter at anytime to skip a dialog. Use ctrl-c at anytime to skip the
remaining dialogs.
Would you like to enter the basic configuration dialog (yes/no): yes
Do you want to enforce secure password standard (yes/no) [y]:
Create another login account (yes/no) [n]:
Configure read-only SNMP community string (yes/no) [n]:
Configure read-write SNMP community string (yes/no) [n]:
Enter the switch name : N9k-FAB-A
Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]:
    Mgmt0 IPv4 address : 10.22.100.3
Mgmt0 IPv4 netmask : 255.255.255.0
    Configure the default gateway? (yes/no) [y]:
    IPv4 address of the default gateway : 10.22.100.1
    Configure advanced IP options? (yes/no) [n]:
    Enable the telnet service? (yes/no) [n]:
    Enable the ssh service? (yes/no) [y]:
    Type of ssh key you would like to generate (dsa/rsa) [rsa]:
    Number of rsa key bits <1024-2048> [2048]:
    Configure the ntp server? (yes/no) [n]: y
    NTP server IPv4 address : <<ntp_server_ip>>

    Configure CoPP system profile (strict/moderate/lenient/dense/skip)
[strict]:
The following configuration will be applied:
password strength-check
switchname N9k-FAB-A
vrf context management
ip route 0.0.0.0/0 10.22.100.1
exit
no feature telnet
ssh key rsa 2048 force
feature ssh
ntp server <<var_global_ntp_server_ip>>
copp profile strict
interface mgmt0
ip address 10.22.100.3 255.255.255.0
no shutdown
    Would you like to edit the configuration? (yes/no) [n]: Enter
    Use this configuration and save it? (yes/no) [y]: Enter
[#####] 100%
Copy complete.

```

## Configure the Cisco Nexus 9372 PX Switch B

To configure the Cisco Nexus 9372 PX Switch B, complete the following step:

1. Connect the console port to the Nexus 9372 PX switch designated for Fabric B:

```

---- Basic System Configuration Dialog VDC: 1 ----
This setup utility will guide you through the basic configuration of the system.
Setup configures only enough connectivity for management of the system.
*Note: setup is mainly used for configuring the system initially, when no
configuration is present. So setup always assumes system defaults and not the
current system configuration values.
Press Enter at anytime to skip a dialog. Use ctrl-c at anytime to skip the
remaining dialogs.
Would you like to enter the basic configuration dialog (yes/no): yes
Do you want to enforce secure password standard (yes/no) [y]:
Create another login account (yes/no) [n]:
Configure read-only SNMP community string (yes/no) [n]:
Configure read-write SNMP community string (yes/no) [n]:
Enter the switch name : N9k-FAB-B
Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]:
    Mgmt0 IPv4 address : 10.22.100.4
    Mgmt0 IPv4 netmask : 255.255.255.0
    Configure the default gateway? (yes/no) [y]:
    IPv4 address of the default gateway : 10.22.100.1
    Configure advanced IP options? (yes/no) [n]:
    Enable the telnet service? (yes/no) [n]:
    Enable the ssh service? (yes/no) [y]:
    Type of ssh key you would like to generate (dsa/rsa) [rsa]:
    Number of rsa key bits <1024-2048> [2048]:
    Configure the ntp server? (yes/no) [n]: y
    NTP server IPv4 address : <<ntp_server_ip>>

    Configure CoPP system profile (strict/moderate/lenient/dense/skip)
[strict]:
The following configuration will be applied:
password strength-check
switchname N9k-FAB-B
vrf context management
ip route 0.0.0.0/0 10.22.100.1
exit
no feature telnet
ssh key rsa 2048 force
feature ssh
ntp server <<var_global_ntp_server_ip>>
copp profile strict
interface mgmt0
ip address 10.22.100.4 255.255.255.0
no shutdown
    Would you like to edit the configuration? (yes/no) [n]: Enter
    Use this configuration and save it? (yes/no) [y]: Enter
    [#####] 100%
Copy complete.

```

## Enable Features on the Switch

To enable the features on the switch, enter the following:

```

N9K-FAB-A# config terminal
N9k-FAB-A(config)# feature uddld
N9K-FAB-A(config)# feature interface-vlan
N9K-FAB-A(config)# feature hsrp
N9K-FAB-A(config)# feature lacp
N9K-FAB-A(config)# feature vpc
N9K-FAB-A(config)# exit

```



Repeat the same steps on Nexus 9372 Switch B.

---

## Enable Jumbo MTU

To enable the Jumbo MTU, enter the following:

```

N9K-FAB-A# config terminal
N9K-FAB-A(config)# system jumbomtu 9216
N9K-FAB-A(config)# exit

```



Repeat the same steps on Nexus 9372 Switch B.

---

## Create VLANs

To create VLANs, enter the following:

```

N9K-FAB-A# config terminal
N9K-FAB-A(config)# vlan 100
N9K-FAB-A(config-vlan)# name Internal-API
N9K-FAB-A(config-vlan)# no shut
N9K-FAB-A(config-vlan)# exit
N9K-FAB-A(config)# vlan 110
N9K-FAB-A(config-vlan)# name PXE-Network
N9K-FAB-A(config-vlan)# no shut
N9K-FAB-A(config-vlan)# exit
N9K-FAB-A(config)# vlan 120
N9K-FAB-A(config-vlan)# name Storage-Public-Network
N9K-FAB-A(config-vlan)# no shut
N9K-FAB-A(config-vlan)# exit
N9k-FAB-A(config)# vlan 130
N9k-FAB-A(config-vlan)# name Tenant-Internal-Network
N9k-FAB-A(config-vlan)# no shut
N9k-FAB-A(config-vlan)# exit
N9K-FAB-A(config)# vlan 150
N9K-FAB-A(config-vlan)# name Storage-Mgmt-Network
N9K-FAB-A(config-vlan)# no shut
N9K-FAB-A(config-vlan)# exit
N9K-FAB-A(config)# vlan 160
N9K-FAB-A(config-vlan)# name Tenant-Floating-IP-Network
N9K-FAB-A(config-vlan)# no shut
N9K-FAB-A(config-vlan)# exit
N9K-FAB-A(config)# vlan 215
N9K-FAB-A(config-vlan)# name External-Network
N9K-FAB-A(config-vlan)# no shut
N9K-FAB-A(config-vlan)# exit
N9K-FAB-A(config)#

```



---

Repeat the same steps on Nexus 9372 Switch B.

---

## Configure the Interface VLAN (SVI) on the Cisco Nexus 9K Switch A

To configure the Interface VLAN on the Cisco Nexus 9K Switch A, enter the following:

```
N9K-FAB-A(config)#
N9K-FAB-A(config)# interface Vlan100
N9K-FAB-A(config-if)# description Internal-API
N9K-FAB-A(config-if)# no shutdown
N9K-FAB-A(config-if)# no ip redirects
N9K-FAB-A(config-if)# ip address 10.22.100.253/24
N9K-FAB-A(config-if)# no ipv6 redirects
N9K-FAB-A(config-if)# hsrp version 2
N9K-FAB-A(config-if-hsrp)# hsrp 100
N9K-FAB-A(config-if-hsrp)# preempt
N9K-FAB-A(config-if-hsrp)# priority 110
N9K-FAB-A(config-if-hsrp)# ip 10.22.100.1
N9K-FAB-A(config-if-hsrp)#exit

N9K-FAB-A(config)# interface Vlan110
N9K-FAB-A(config-if)# description PXE_Network
N9K-FAB-A(config-if)# no shutdown
N9K-FAB-A(config-if)# no ip redirects
N9K-FAB-A(config-if)# ip address 10.22.110.253/24
N9K-FAB-A(config-if)# no ipv6 redirects
N9K-FAB-A(config-if)# hsrp version 2
N9K-FAB-A(config-if-hsrp)# hsrp 110
N9K-FAB-A(config-if-hsrp)# preempt
N9K-FAB-A(config-if-hsrp)# priority 110
N9K-FAB-A(config-if-hsrp)# ip 10.22.110.1
N9K-FAB-A(config-if-hsrp)#exit

N9K-FAB-A(config)# interface Vlan120
N9K-FAB-A(config-if)# description Storage_Public_Network
N9K-FAB-A(config-if)# no shutdown
N9K-FAB-A(config-if)# no ip redirects
N9K-FAB-A(config-if)# ip address 10.22.120.253/24
N9K-FAB-A(config-if)# no ipv6 redirects
N9K-FAB-A(config-if)# hsrp version 2
N9K-FAB-A(config-if-hsrp)# hsrp 120
N9K-FAB-A(config-if-hsrp)# preempt
N9K-FAB-A(config-if-hsrp)# priority 110
N9K-FAB-A(config-if-hsrp)# ip 10.22.120.1
N9K-FAB-A(config-if-hsrp)#exit

N9K-FAB-A(config)# interface Vlan130
N9K-FAB-A(config-if)# description Tenant_Internal_Network
N9K-FAB-A(config-if)# no shutdown
N9K-FAB-A(config-if)# no ip redirects
N9K-FAB-A(config-if)# ip address 10.22.130.253/24
N9K-FAB-A(config-if)# no ipv6 redirects
N9K-FAB-A(config-if)# hsrp version 2
N9K-FAB-A(config-if-hsrp)# hsrp 130
N9K-FAB-A(config-if-hsrp)# preempt
N9K-FAB-A(config-if-hsrp)# priority 110
N9K-FAB-A(config-if-hsrp)# ip 10.22.130.1
```

```

N9K-FAB-A(config-if-hsrp)#exit

N9K-FAB-A(config)# interface Vlan150
N9K-FAB-A(config-if)# description Storage_ClusterMgmt_Network
N9K-FAB-A(config-if)# no shutdown
N9K-FAB-A(config-if)# no ip redirects
N9K-FAB-A(config-if)# ip address 10.22.150.253/24
N9K-FAB-A(config-if)# no ipv6 redirects
N9K-FAB-A(config-if)# hsrp version 2
N9K-FAB-A(config-if-hsrp)# hsrp 150
N9K-FAB-A(config-if-hsrp)# preempt
N9K-FAB-A(config-if-hsrp)# priority 110
N9K-FAB-A(config-if-hsrp)# ip 10.22.150.1
N9K-FAB-A(config-if-hsrp)#exit

N9K-FAB-A(config)# interface Vlan160
N9K-FAB-A(config-if)# description Tenanat_Floating_Network
N9K-FAB-A(config-if)# no shutdown
N9K-FAB-A(config-if)# no ip redirects
N9K-FAB-A(config-if)# ip address 10.22.175.253/20
N9K-FAB-A(config-if)# no ipv6 redirects
N9K-FAB-A(config-if)# hsrp version 2
N9K-FAB-A(config-if-hsrp)# hsrp 160
N9K-FAB-A(config-if-hsrp)# preempt
N9K-FAB-A(config-if-hsrp)# priority 110
N9K-FAB-A(config-if-hsrp)# ip 10.22.160.1
N9K-FAB-A(config-if-hsrp)#exit

N9K-FAB-A(config)# interface Vlan215
N9K-FAB-A(config-if)# description External_Network
N9K-FAB-A(config-if)# no shutdown
N9K-FAB-A(config-if)# no ip redirects
N9K-FAB-A(config-if)# ip address 172.22.215.253/24
N9K-FAB-A(config-if)# no ipv6 redirects
N9K-FAB-A(config-if)# exit
N9K-FAB-A(config)# Copy running-config Startup-config

```

## Configure the Interface VLAN (SVI) on the Cisco Nexus 9K Switch B

To configure the Interface VLAN on the Cisco Nexus 9K Switch B, enter the following:

```

N9k-FAB-B(config)#
N9k-FAB-B(config)# interface Vlan100
N9k-FAB-B(config-if)# description Internal-API
N9k-FAB-B(config-if)# no shutdown
N9k-FAB-B(config-if)# no ip redirects
N9k-FAB-B(config-if)# ip address 10.22.100.254/24
N9k-FAB-B(config-if)# no ipv6 redirects
N9k-FAB-B(config-if)# hsrp version 2
N9k-FAB-B(config-if-hsrp)# hsrp 100
N9k-FAB-B(config-if-hsrp)# preempt
N9k-FAB-B(config-if-hsrp)# priority 100
N9k-FAB-B(config-if-hsrp)# ip 10.22.100.1
N9k-FAB-B(config-if-hsrp)# exit

N9k-FAB-B(config)# interface Vlan110
N9k-FAB-B(config-if)# description PXE_Network
N9k-FAB-B(config-if)# no shutdown
N9k-FAB-B(config-if)# no ip redirects

```

```
N9k-FAB-B(config-if)# ip address 10.22.110.254/24
N9k-FAB-B(config-if)# no ipv6 redirects
N9k-FAB-B(config-if)# hsrp version 2
N9k-FAB-B(config-if-hsrp)# hsrp 110
N9k-FAB-B(config-if-hsrp)# preempt
N9k-FAB-B(config-if-hsrp)# priority 100
N9k-FAB-B(config-if-hsrp)# ip 10.22.110.1
N9k-FAB-B(config-if-hsrp)# exit
```

```
N9k-FAB-B(config)# interface Vlan120
N9k-FAB-B(config-if)# description Storage_Public_Network
N9k-FAB-B(config-if)# no shutdown
N9k-FAB-B(config-if)# no ip redirects
N9k-FAB-B(config-if)# ip address 10.22.120.254/24
N9k-FAB-B(config-if)# no ipv6 redirects
N9k-FAB-B(config-if)# hsrp version 2
N9k-FAB-B(config-if-hsrp)# hsrp 120
N9k-FAB-B(config-if-hsrp)# preempt
N9k-FAB-B(config-if-hsrp)# priority 100
N9k-FAB-B(config-if-hsrp)# ip 10.22.120.1
N9k-FAB-B(config-if-hsrp)# exit
```

```
N9k-FAB-B(config)# interface Vlan130
N9k-FAB-B(config-if)# description Tenant_Internal_Network
N9k-FAB-B(config-if)# no shutdown
N9k-FAB-B(config-if)# no ip redirects
N9k-FAB-B(config-if)# ip address 10.22.130.254/24
N9k-FAB-B(config-if)# no ipv6 redirects
N9k-FAB-B(config-if)# hsrp version 2
N9k-FAB-B(config-if-hsrp)# hsrp 130
N9k-FAB-B(config-if-hsrp)# preempt
N9k-FAB-B(config-if-hsrp)# priority 100
N9k-FAB-B(config-if-hsrp)# ip 10.22.130.1
N9k-FAB-B(config-if-hsrp)# exit
```

```
N9k-FAB-B(config)# interface Vlan150
N9k-FAB-B(config-if)# description Storage_ClusterMgmt_Network
N9k-FAB-B(config-if)# no shutdown
N9k-FAB-B(config-if)# no ip redirects
N9k-FAB-B(config-if)# ip address 10.22.150.254/24
N9k-FAB-B(config-if)# no ipv6 redirects
N9k-FAB-B(config-if)# hsrp version 2
N9k-FAB-B(config-if-hsrp)# hsrp 150
N9k-FAB-B(config-if-hsrp)# preempt
N9k-FAB-B(config-if-hsrp)# priority 100
N9k-FAB-B(config-if-hsrp)# ip 10.22.150.1
N9k-FAB-B(config-if-hsrp)# exit
```

```
N9k-FAB-B(config)# interface Vlan160
N9k-FAB-B(config-if)# description Tenanat_Floating_Network
N9k-FAB-B(config-if)# no shutdown
N9k-FAB-B(config-if)# no ip redirects
N9k-FAB-B(config-if)# ip address 10.22.175.254/20
N9k-FAB-B(config-if)# no ipv6 redirects
N9k-FAB-B(config-if)# hsrp version 2
N9k-FAB-B(config-if-hsrp)# hsrp 160
N9k-FAB-B(config-if-hsrp)# preempt
N9k-FAB-B(config-if-hsrp)# priority 100
N9k-FAB-B(config-if-hsrp)# ip 10.22.160.1
```

```

N9k-FAB-B(config-if-hsrp)# exit

N9k-FAB-B(config)# interface Vlan215
N9k-FAB-B(config-if)# description External_Network
N9k-FAB-B(config-if)# no shutdown
N9k-FAB-B(config-if)# no ip redirects
N9k-FAB-B(config-if)# ip address 172.22.215.254/24
N9k-FAB-B(config-if)# no ipv6 redirects
N9k-FAB-B(config-if)# exit
N9K-FAB-B(config)# Copy running-config Startup-config

```

## Configure the VPC and Port Channels on Switch A

To configure the VPC and Port Channels on Switch A, enter the following:

```

N9K-FAB-A(config)# vpc domain 100
N9K-FAB-A(config-vpc-domain)# role priority 10
N9K-FAB-A(config-vpc-domain)# peer-keepalive destination 10.22.100.4
N9K-FAB-A(config-vpc-domain)# peer-gateway
N9K-FAB-A(config-vpc-domain)# exit

N9K-FAB-A(config)# interface port-channel1
N9K-FAB-A(config-if)# description VPC peerlink for Nexus 9k Switch A & B
N9K-FAB-A(config-if)# switchport mode trunk
N9K-FAB-A(config-if)# spanning-tree port type network
N9K-FAB-A(config-if)# speed 10000
N9K-FAB-A(config-if)# vpc peer-link
N9K-FAB-A(config-if)# exit

N9K-FAB-A(config)# interface Ethernet1/1
N9K-FAB-A(config-if)# description connected to Peer Nexus 9k-B port1/1
N9K-FAB-A(config-if)# switchport mode trunk
N9K-FAB-A(config-if)# speed 10000
N9K-FAB-A(config-if)# channel-group 1 mode active
N9K-FAB-A(config-if)# exit

N9K-FAB-A(config)# interface Ethernet1/2
N9K-FAB-A(config-if)# description connected to Peer Nexus 9k-B port1/2
N9K-FAB-A(config-if)# switchport mode trunk
N9K-FAB-A(config-if)# speed 10000
N9K-FAB-A(config-if)# channel-group 1 mode active
N9K-FAB-A(config-if)# exit

N9K-FAB-A(config)# interface port-channel17
N9K-FAB-A(config-if)# description Port-channel for UCS_Fabric_A port_17 & port_18
N9K-FAB-A(config-if)# vpc 17
N9K-FAB-A(config-if)# exit

N9K-FAB-A(config)# interface port-channel18
N9K-FAB-A(config-if)# description Port-channel for UCS_Fabric_B port_17 & port_18
N9K-FAB-A(config-if)# vpc 18
N9K-FAB-A(config-if)# exit

N9K-FAB-A(config)# interface Ethernet1/17
N9K-FAB-A(config-if)# description Uplink from UCS_Fabric_A_Port_17
N9K-FAB-A(config-if)# channel-group 17 mode active
N9K-FAB-A(config-if)# exit

```

```

N9K-FAB-A(config)# interface Ethernet1/18
N9K-FAB-A(config-if)# description Uplink from UCS_Fabric_B_Port_17
N9K-FAB-A(config-if)# channel-group 18 mode active
N9K-FAB-A(config-if)# exit

N9K-FAB-A(config)# interface port-channel17
N9K-FAB-A(config-if)# switchport mode trunk
N9K-FAB-A(config-if)# switchport trunk allowed vlan 100,110,120,130,150,160,215
N9K-FAB-A(config-if)# spanning-tree port type edge trunk
N9K-FAB-A(config-if)# mtu 9216
N9K-FAB-A(config-if)# exit

N9K-FAB-A(config)# interface port-channel18
N9K-FAB-A(config-if)# switchport mode trunk
N9K-FAB-A(config-if)# switchport trunk allowed vlan 100,110,120,130,150,160,215
N9K-FAB-A(config-if)# spanning-tree port type edge trunk
N9K-FAB-A(config-if)# mtu 9216
N9K-FAB-A(config-if)# exit

N9K-FAB-A(config)# copy running-config startup-config

```

## Configure the VPC and Port Channels on the Cisco Nexus 9K Switch B

To configure the VPC and Port Channels on the Cisco Nexus 9K Switch B, enter the following:

```

N9k-FAB-B(config)# vpc domain 100
N9k-FAB-B(config-vpc-domain)# role priority 10
N9k-FAB-B(config-vpc-domain)# peer-keepalive destination 10.22.100.3
N9k-FAB-B(config-vpc-domain)# peer-gateway
N9k-FAB-B(config-vpc-domain)# exit

N9k-FAB-B(config)# interface port-channel1
N9k-FAB-B(config-if)# description VPC peerlink for Nexus 9k Switch A & B
N9k-FAB-B(config-if)# switchport mode trunk
N9k-FAB-B(config-if)# spanning-tree port type network
N9k-FAB-B(config-if)# speed 10000
N9k-FAB-B(config-if)# vpc peer-link
N9k-FAB-B(config-if)# exit

N9k-FAB-B(config)# interface Ethernet1/1
N9k-FAB-B(config-if)# description connected to Peer Nexus 9k-A port1/1
N9k-FAB-B(config-if)# switchport mode trunk
N9k-FAB-B(config-if)# speed 10000
N9k-FAB-B(config-if)# channel-group 1 mode active
N9k-FAB-B(config-if)# exit

N9k-FAB-B(config)# interface Ethernet1/2
N9k-FAB-B(config-if)# description connected to Peer Nexus 9k-A port1/2
N9k-FAB-B(config-if)# switchport mode trunk
N9k-FAB-B(config-if)# speed 10000
N9k-FAB-B(config-if)# channel-group 1 mode active
N9k-FAB-B(config-if)# exit

N9K-FAB-B(config)# interface port-channel17
N9K-FAB-B(config-if)# description Port-channel for UCS_Fabric_A port_17 & port_18
N9K-FAB-B(config-if)# vpc 17
N9K-FAB-B(config-if)# exit

N9K-FAB-B(config)# interface port-channel18

```

```
N9K-FAB-B(config-if)# description Port-channel for UCS_Fabric_B port_17 & port_18
N9K-FAB-B(config-if)# vpc 18
N9K-FAB-B(config-if)# exit

N9K-FAB-B(config)# interface Ethernet1/17
N9K-FAB-B(config-if)# description Uplink from UCS_Fabric_A_Port_18
N9K-FAB-B(config-if)# channel-group 17 mode active
N9K-FAB-B(config-if)# exit

N9K-FAB-B(config)# interface Ethernet1/18
N9K-FAB-B(config-if)# description Uplink from UCS_Fabric_B_Port_18
N9K-FAB-B(config-if)# channel-group 18 mode active
N9K-FAB-B(config-if)# exit

N9K-FAB-B(config)# interface port-channel17
N9K-FAB-B(config-if)# switchport mode trunk
N9K-FAB-B(config-if)# switchport trunk allowed vlan 100,110,120,130,150,160,215
N9K-FAB-B(config-if)# spanning-tree port type edge trunk
N9K-FAB-B(config-if)# mtu 9216
N9K-FAB-B(config-if)# exit

N9K-FAB-B(config)# interface port-channel18
N9K-FAB-B(config-if)# switchport mode trunk
N9K-FAB-B(config-if)# switchport trunk allowed vlan 100,110,120,130,150,160,215
N9K-FAB-B(config-if)# spanning-tree port type edge trunk
N9K-FAB-B(config-if)# mtu 9216
N9K-FAB-B(config-if)# exit

N9K-FAB-B(config)# copy running-config startup-config
```

## Verify the Port Channel Status on the Cisco Nexus Switches

After successfully creating a Virtual Port Channel on both Nexus switches, verify the Port Channel status on the Nexus 9K Switch. To verify the status, enter the following:

```
UCSO-N9K-FAB-A(config)# show vpc br
```

```
Legend:
```

```
(*) - local vpc is down, forwarding via vpc peer-link
```

```
vpc domain id           : 100
Peer status             : peer adjacency formed ok
vpc keep-alive status   : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success
vpc role                : secondary
Number of vpcs configured : 2
Peer Gateway            : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status    : Disabled
Delay-restore status    : Timer is off.(timeout = 30s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
```

```
vpc Peer-link status
```

```
-----
id  Port  Status Active vlans
---  ---  -
1   Po1   up    1,100,110,120,130,160,215,291,338
-----
```

```
vpc status
```

```
-----
id  Port  Status Consistency Reason           Active vlans
---  ---  -
17  Po17  up    success  success           1,100,110,120,130,160,215
18  Po18  up    success  success           1,100,110,120,130,160,215
-----
```

```
UCSO-N9K-FAB-A(config)#
```

```
UCSO-N9K-FAB-A(config)# show port-channel summary
Flags:  D - Down          P - Up in port-channel (members)
        I - Individual   H - Hot-standby (LACP only)
        s - Suspended    r - Module-removed
        S - Switched     R - Routed
        U - Up (port-channel)
        p - Up in delay-lacp mode (member)
        M - Not in use. Min-links not met
```

```
-----
Group Port-Channel      Type      Protocol  Member Ports
-----
1     Po1(SU)               Eth       LACP      Eth1/1(P)  Eth1/2(P)
17    Po17(SU)              Eth       LACP      Eth1/17(P)
18    Po18(SU)              Eth       LACP      Eth1/18(P)
-----
```

Verify the Port Channels Status on the Fabrics

To verify the status on the Fabrics, complete the following steps as shown in the screenshots below:

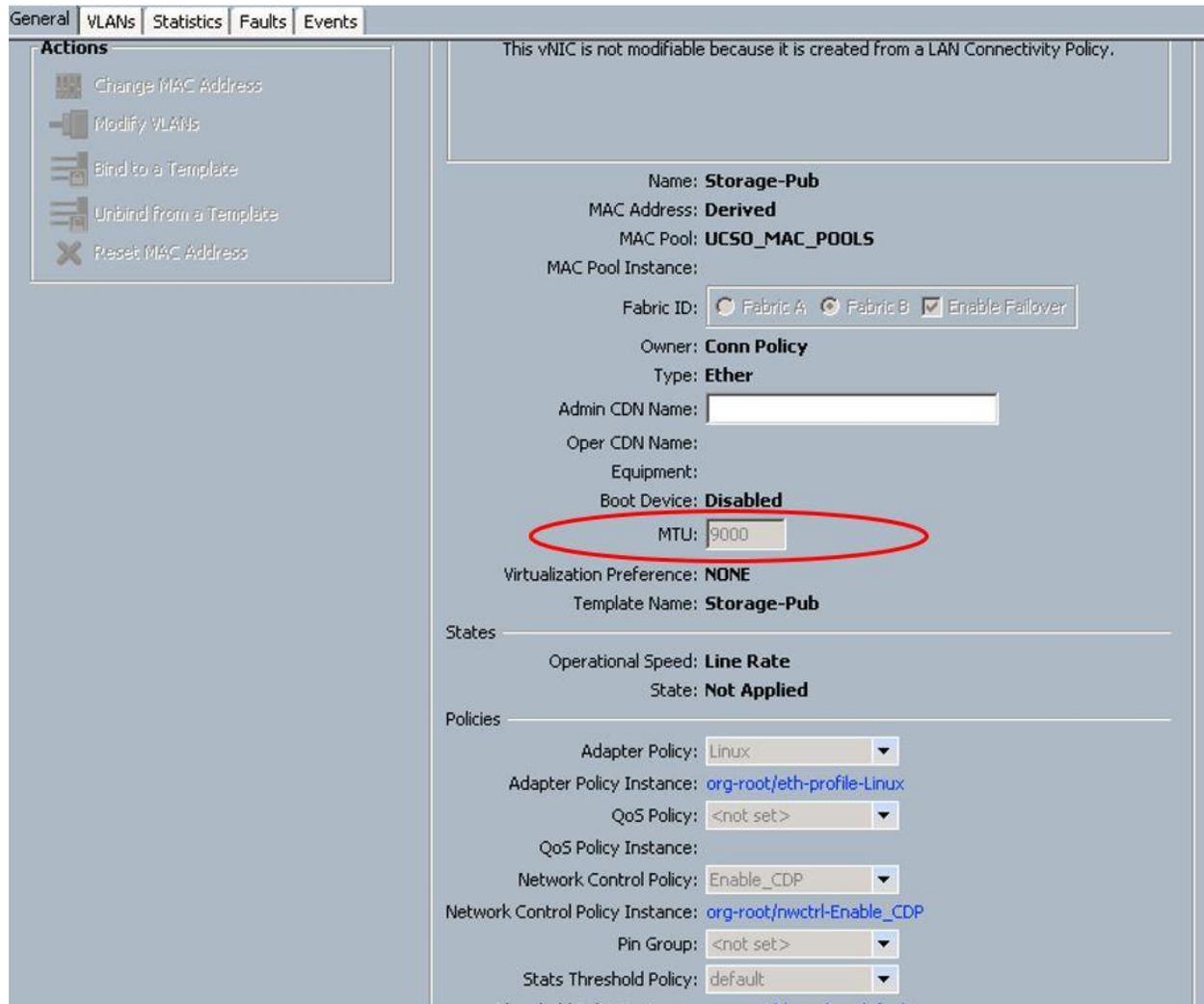
The screenshot displays a network management console. On the left, a tree view shows the hierarchy: LAN Cloud > Fabric A > Port Channels > Port-Channel 17 (VPC-17-FabricA) and Fabric B > Port Channels > Port-Channel 18 (VPC-18-FabricB). The right pane is titled 'Port-Channel 18 (FabB-PO-18)' and has tabs for General, Ports, Faults, Events, and Statistics. The 'General' tab is active, showing the following details:

- Status:** Overall Status: **Up** (indicated by a green up arrow icon). Additional Info: (empty)
- Actions:** Enable Port Channel, Disable Port Channel, Add Ports
- Properties:**
  - ID: **18**
  - Fabric ID: **B**
  - Port Type: **Aggregation**
  - Transport Type: **Ether**
  - Name:
  - Description:
  - Flow Control Policy: **default** (dropdown)
  - LACP Policy: **default** (dropdown)
  - Note: Changing LACP policy may flap the port-channel if the suspend-individual value changes!
  - Admin Speed:  1 Gbps  10 Gbps
  - Operational Speed(Gbps): **20**

## Cisco UCS Validation Checks

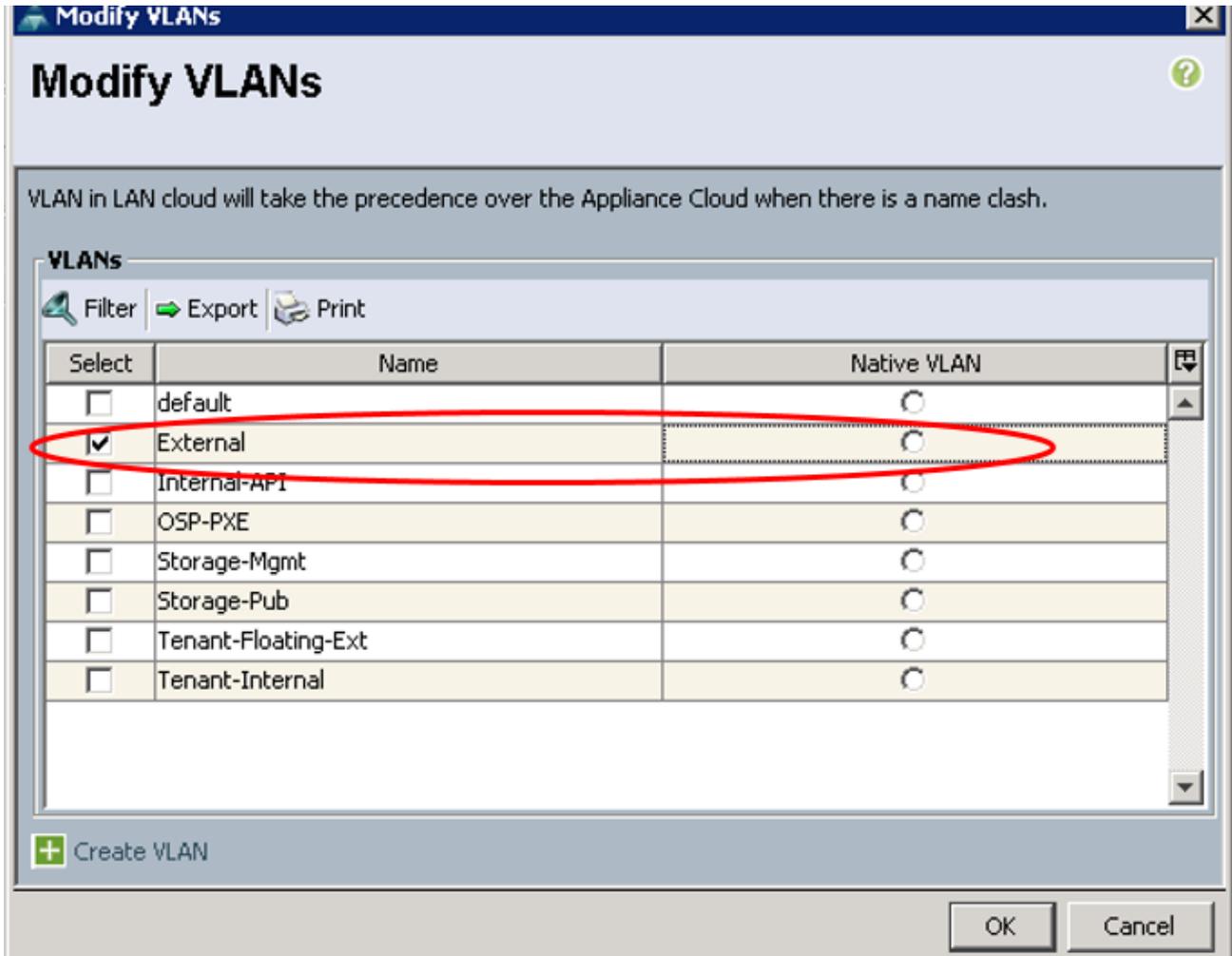
Prior to starting the Operating System installation on the Undercloud Node, you must complete the pre-validation checks. To complete the validation checks, complete the following steps:

1. If you are planning to use Jumbo frames for the storage network, make sure to enter the following information in the templates as shown in the screenshot below.



2. When the service profiles are created from the template, unbind from the templates in case they have been created as updating templates. This is to accommodate the UCS Manager Plugin. Keeping the compute host's service profiles bound to the template does not allow the plugin to individually configure each compute host with tenant based VLANs. Hence, the service profiles for each compute host need to be unbound from the template. Please check the current limitations outlined in the [UCSM Kilo plugin web page](#).
3. The naming convention for the tenant interfaces is also vNIC eth1. This is the same for the Cisco UCS Manager Plugin, link provided above.

- VLAN ID is already included in OpenStack configuration. Do not have native vlan tagged for your external interface on overcloud service profiles.



- The provisioning interfaces should be Native for both Undercloud and Overcloud setups.
- While planning your networks, make sure all the networks defined are not overlapping with any of your data-center networks.
- The disks should be in the same order across all storage nodes.

## Install the Operating System on the Undercloud Node

It is highly recommended to install the Operating System with Versionlock as outlined in the steps below. Versionlock restricts yum to install or upgrade a package to a fixed specific version than specified using the Versionlock plugin of yum

The steps outlined in this document including a few of the configurations, are bound to the installed packages. Installing the same set of packages as in the Cisco Validated Design ensures accuracy of the solution with minimal deviations. This is an attempt to make sure that the installation steps deviate lesser when OpenStack packages move further. While installing RHEL-OSP7 on Cisco blade and rack servers without version lock should still work, it needs to be noted that there could be changes in the configurations and install steps needed that may not exist in this document.



Any updates to the Undercloud stack through yum install may conflict with the version lock packages. You may have to relax the lock files for such updates, when it is required. It is strongly recommended to complete the install with version lock first followed by Overcloud install before attempting any such updates.



Download the Versionlock and kick start the file from Cisco Systems.

To install the Operating System on the Undercloud Node, complete the following steps:

1. Download the Versionlock files for Red Hat Enterprise Linux 7.2 and Red Hat OpenStack 7.2 along with the kick start file from:

<https://cmsg-yum-server.cisco.com/yumrepo/cvd/> Use these files for installing the Red Hat Linux 7.2 followed by RHEL-OSP install. Please refer Red Hat Online documentation for kick start install. The steps below are extracted from [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/Installation\\_Guide/sect-kickstart-howto.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/sect-kickstart-howto.html)

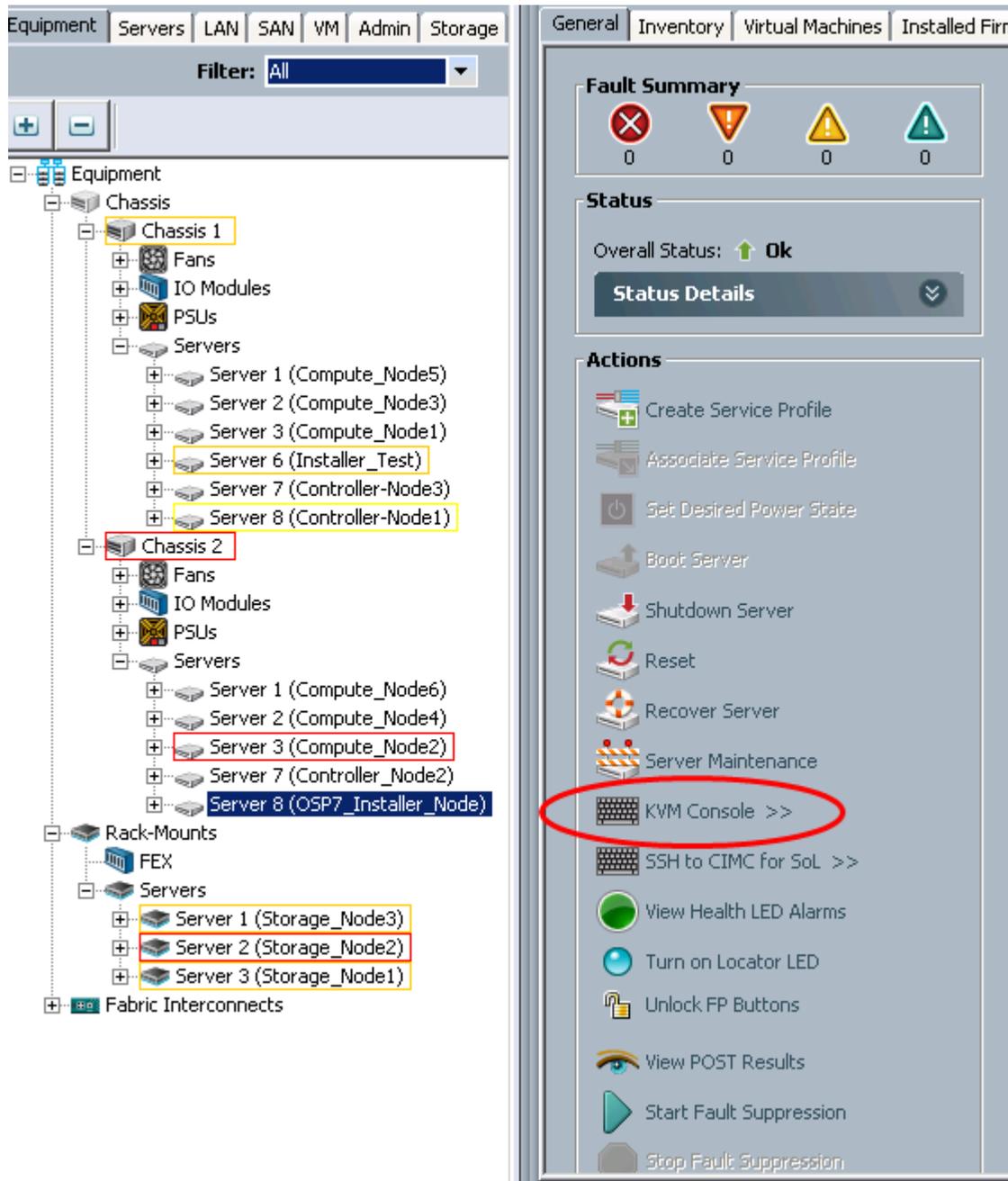
2. Sanity check the files and update on details like subscription management and web server to host these files for network install. The web server should be accessible from this Director node which is being kick started now.
3. Make sure that curl or wget of [http://Your\\_Web\\_server:<port>/<Any\\_Optional\\_Alias>/anaconda-ks.cfg](http://Your_Web_server:<port>/<Any_Optional_Alias>/anaconda-ks.cfg) file is retrievable.
4. Download Red Hat Enterprise Linux 7.2 from <http://access.redhat.com>.
5. Log into the Cisco UCS Manager **and make sure to specify the order of NIC's manually**:

Make sure that Cisco UCS will bring up these interfaces in the following order:

Interface	OS Interface Name	Order
PXE_vNIC	eth0	1

Interface	OS Interface Name	Order
Internal_API_vNIC	eth1	2
External_vNIC	eth2	3

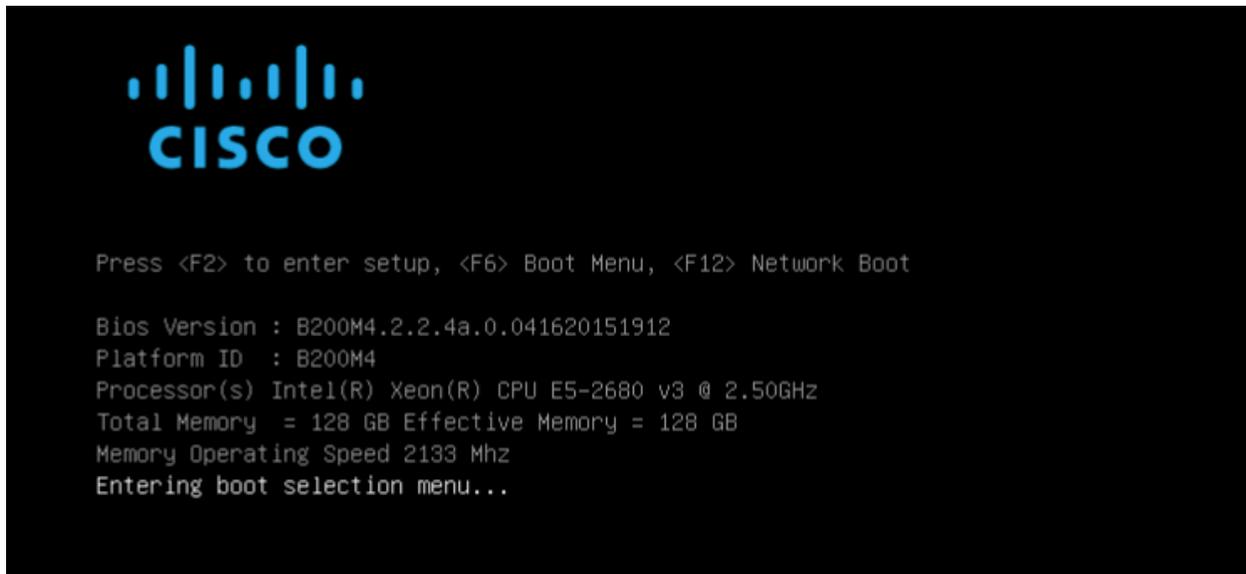
6. Launch the KVM Console; UCS Manager > Equipment Tab > General > KVM Console.



7. In the KVM Console Menu, Activate Virtual Devices under Virtual Media and then click Map CD/DVD, attach the downloaded iso as shown below and then reboot the server.



8. When the system boots up, press F6 for the boot menu.



9. Press tab to enter the command line options as shown below:

- inst.ks, the network location of the anaconda-ks.cfg file, along with network parameters like ip, netmask, hostname, gateway and interface name to boot up with to reach the web server.

The following is an example of what was used in the configuration. Please change your installation accordingly:

- inst.ks=http://172.22.215.6/anaconda-ks.cfg net.ifnames=0 biosdevname=0 ip=172.22.215.9::172.22.215.1:255.255.255.0:osp7-director.cisco.com:eth2:none
- eth2 is the external interface configured. See above.

```

KVM Console Properties
Red Hat Enterprise Linux 7.2

Install Red Hat Enterprise Linux 7.2
Test this media & install Red Hat Enterprise Linux 7.2
Troubleshooting >

> vmlinuz initrd=initrd.img inst.stage2=hd:LABEL=RHEL-7.2\x20Server.x86_64 rd.
live.check quiet inst.ks=http://172.22.215.6/anaconda-ks.cfg net.ifnames=0 bio
sdevname=0 ip=172.22.215.9::172.22.215.1:255.255.255.0:osp7-director.cisco.com
:eth2:none

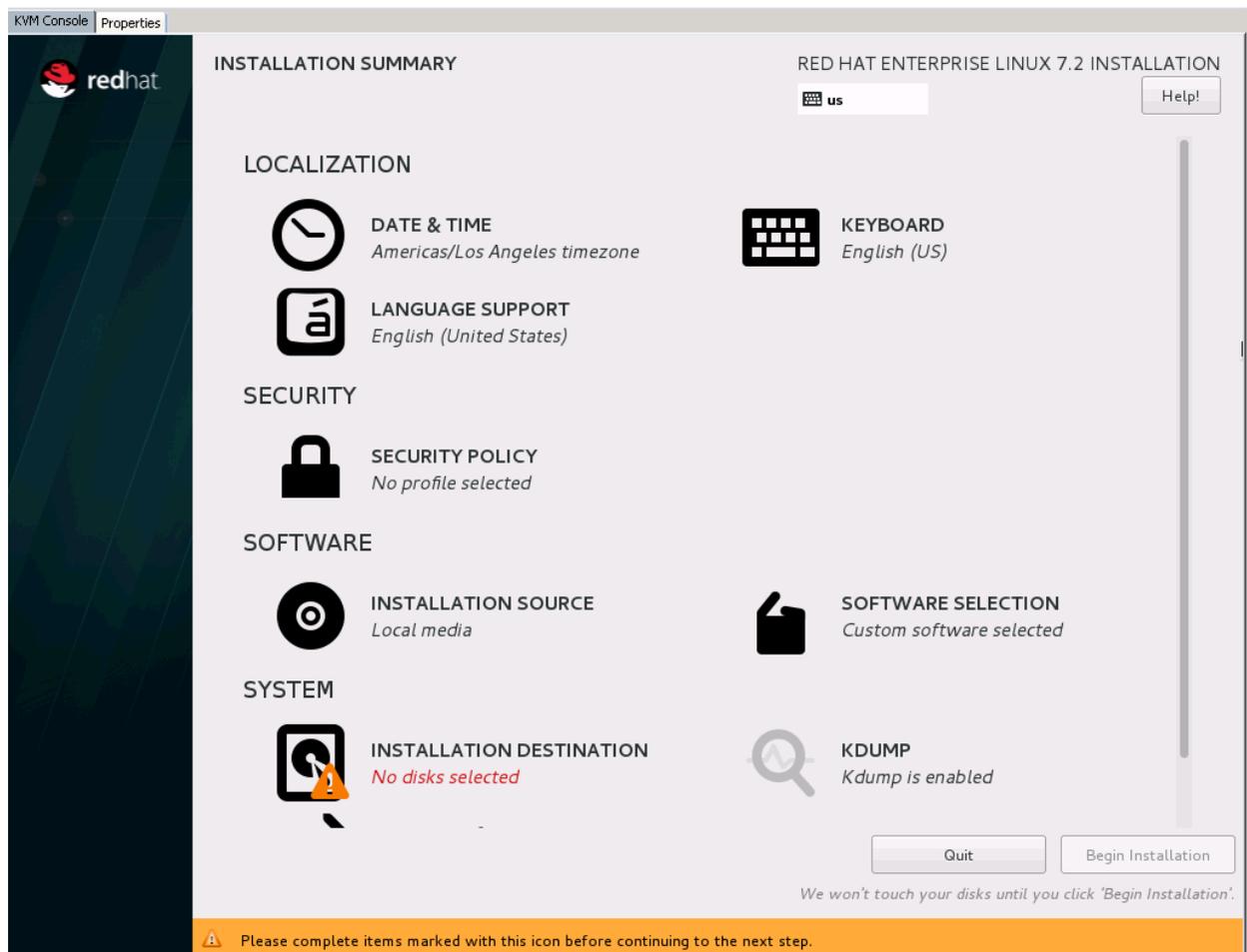
```

10. Select the default language and time zone.

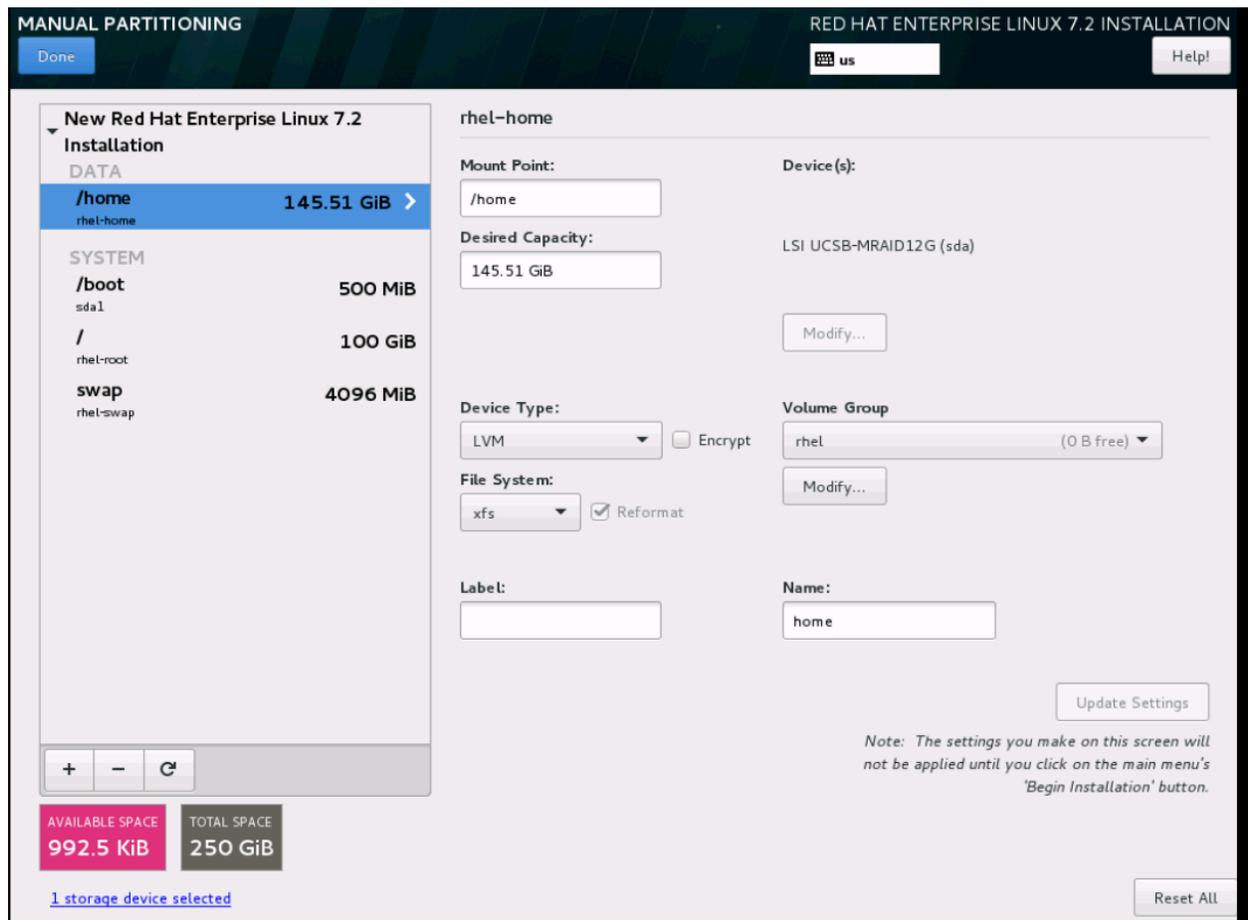


Ignore the software selection since this will come from the kick start file.

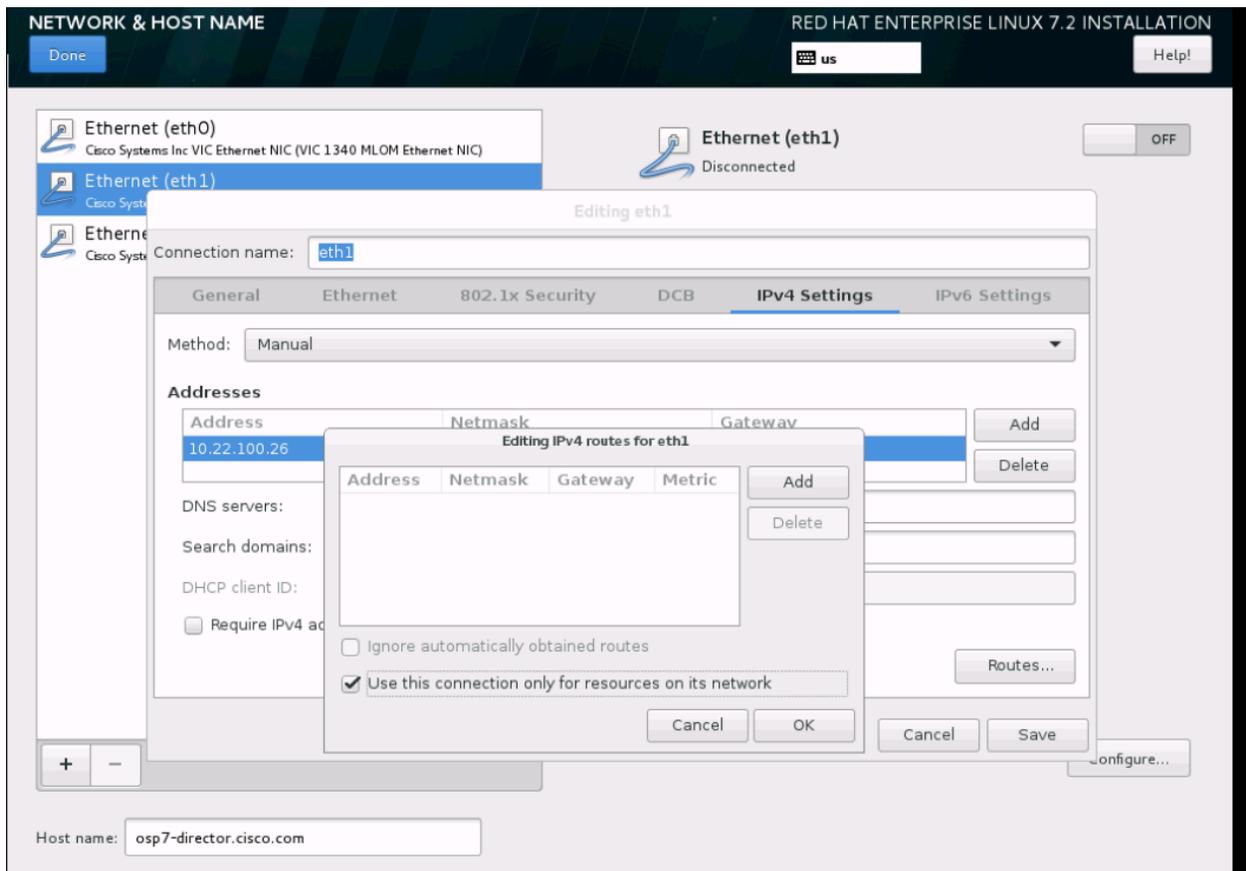
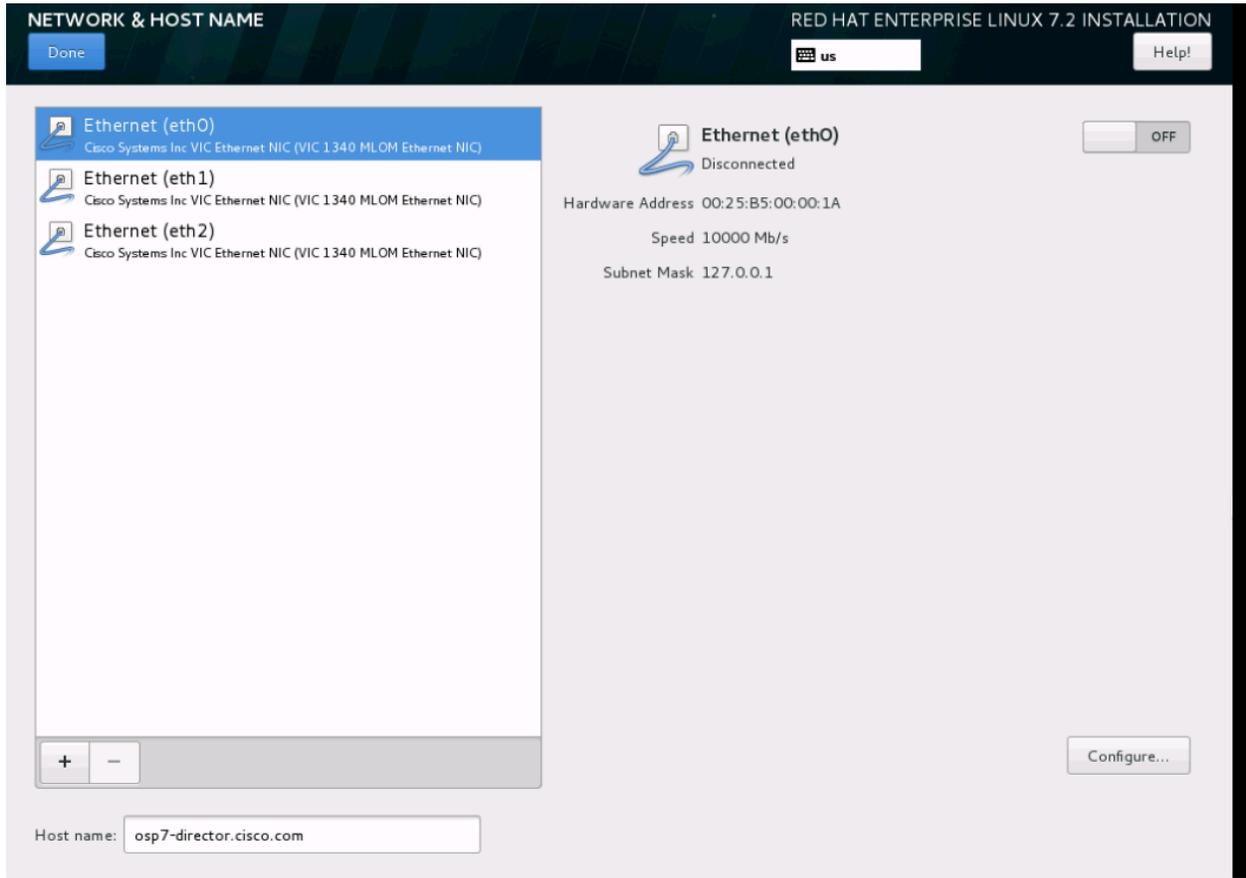
---

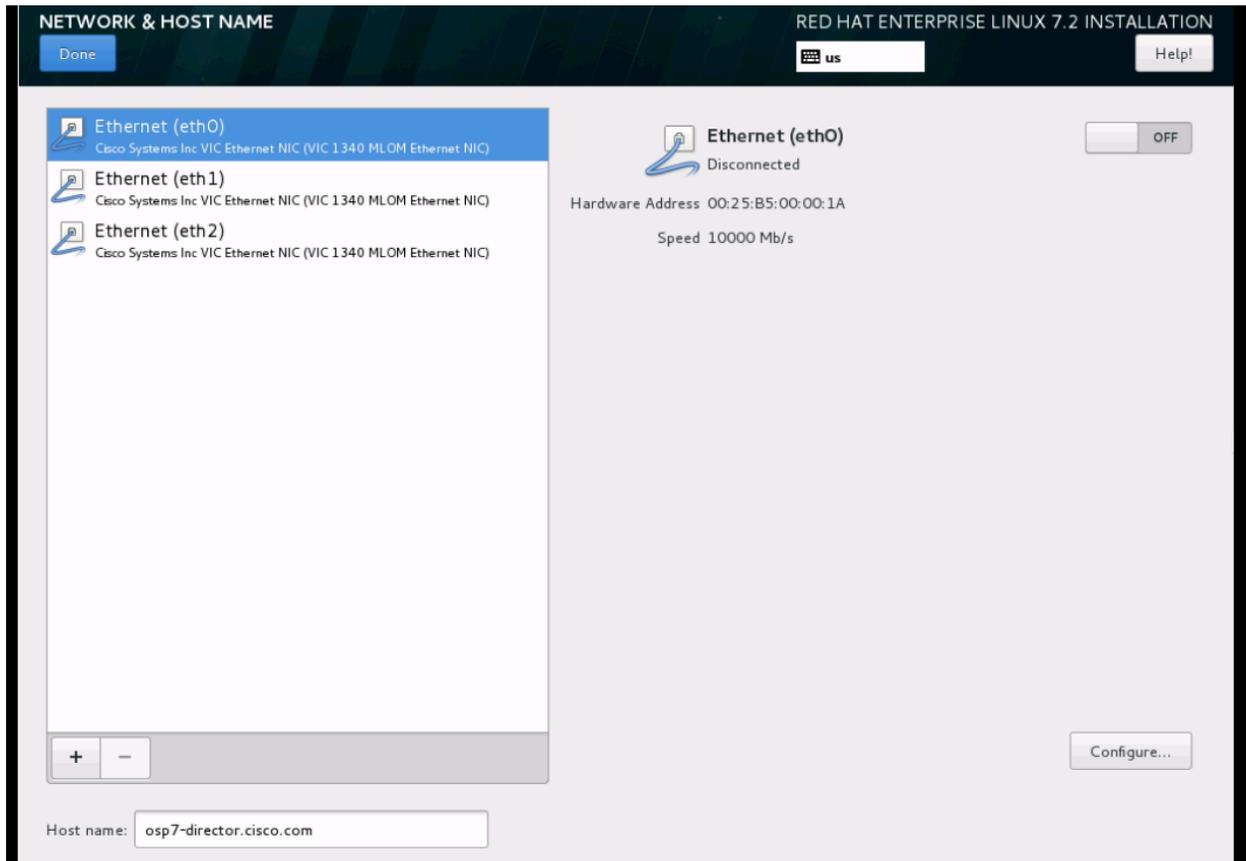


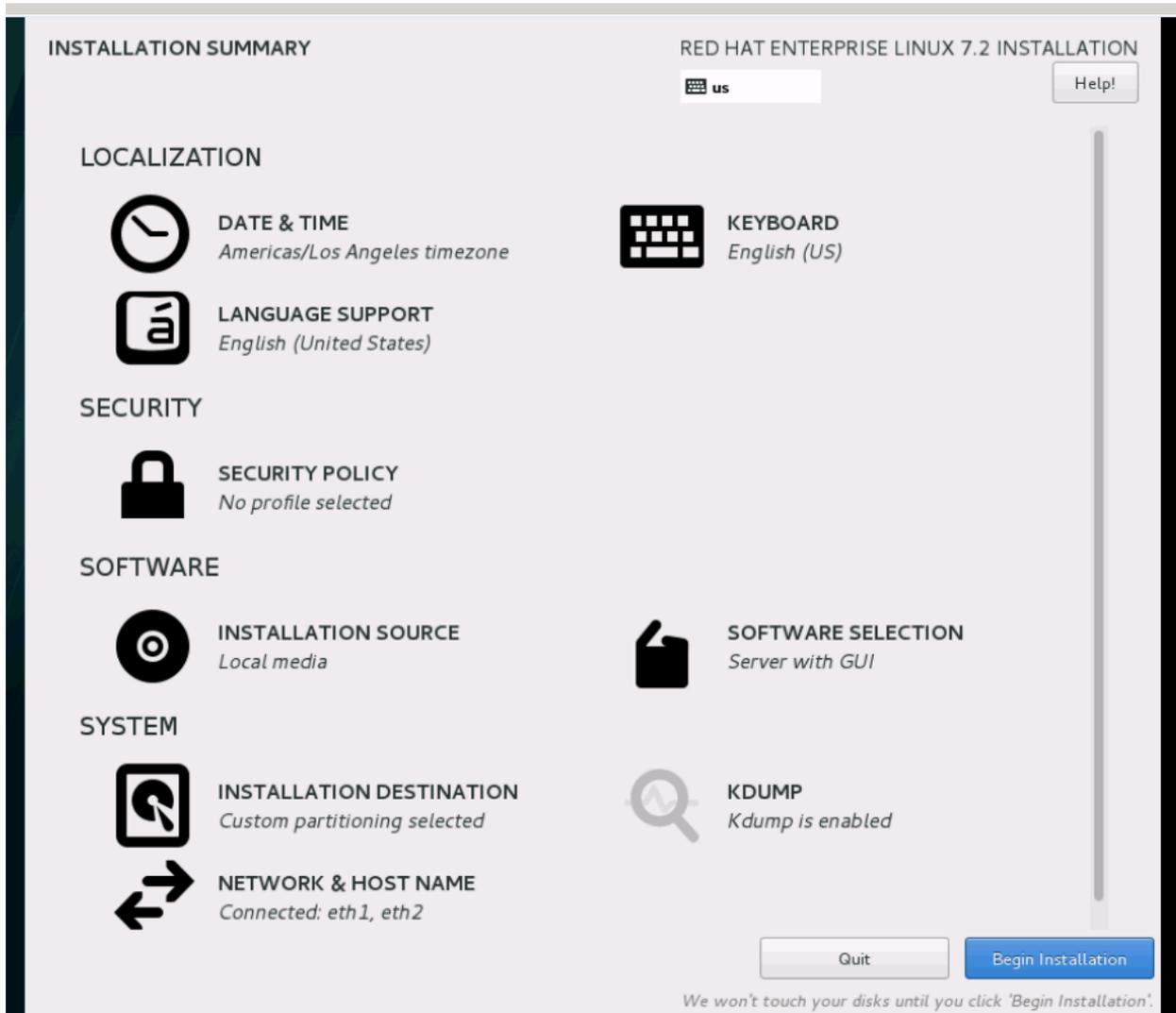
11. Select manual partitioning and remove any unwanted partitions. This LUN is carved out from two disks on the Undercloud node (RAID 10 mirror set up through local disk config policy or through storage profile in Cisco UCS). Preferably increase the root partition to 100GB.



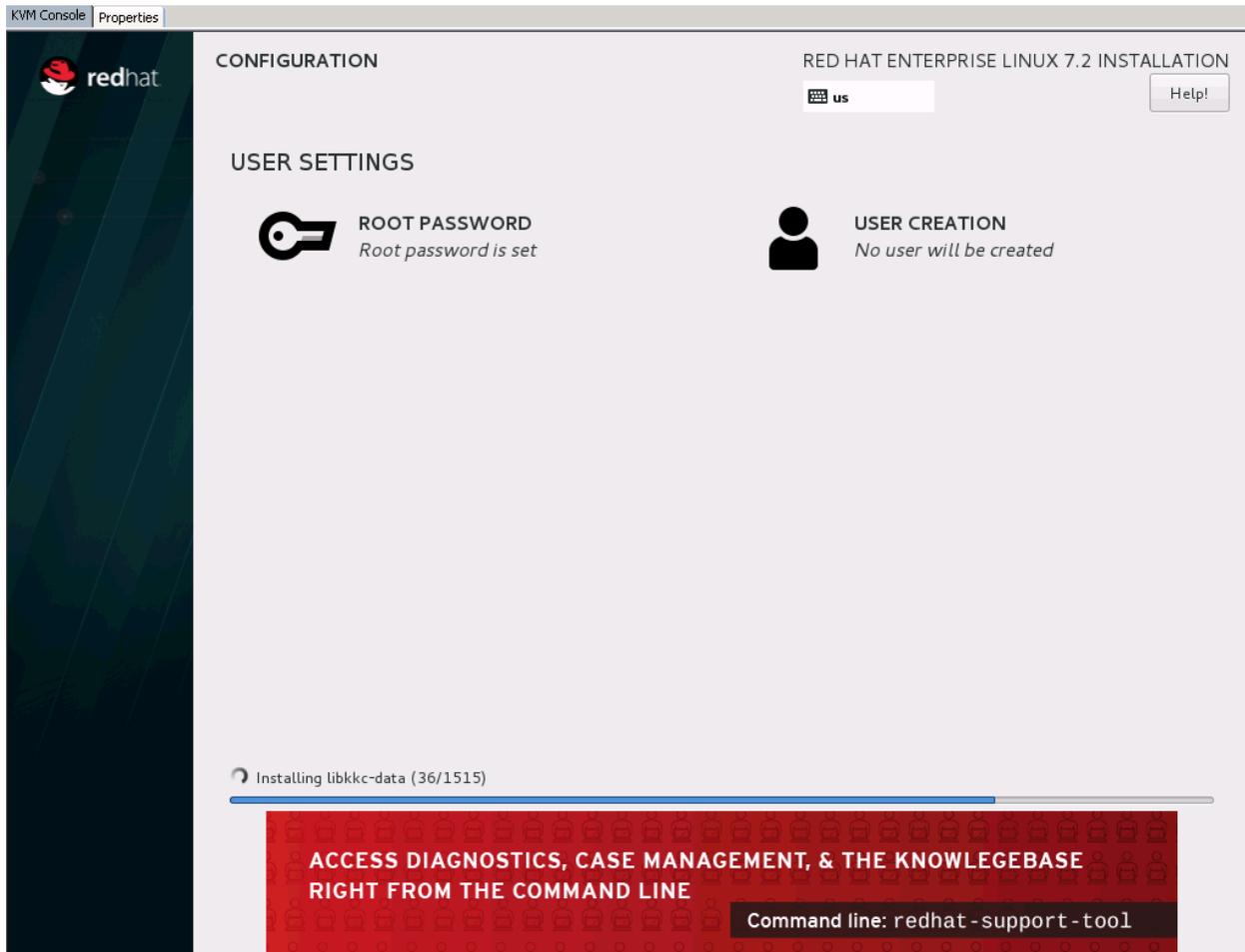
12. Select network tab and configure the external and internal api nics as shown below:







13. Add eth2 for public network. This is the interface that undercloud will pull the necessary files from Red Hat website during install. eth1 interface is not mandatory. However it has been added on the test bed to login to Fabric Interconnects and/or Nexus switches. Leave the pxe interface NIC as un-configured. It will be configured later through the Undercloud install.



14. Enter the root password and optionally create the stack user and reboot the server when prompted.

15. Run Post Install checks before proceeding:

a. Run `subscription-manager status` to check the registration status. Make sure that the pool with OpenStack entitlements is attached.

b. Make sure that version lock list package is installed as part of kickstart.

```
[root@osp7-director heat]# rpm -qa | grep yum-plugin-versionlock
yum-plugin-versionlock-1.1.31-34.el7.noarch
```

If red hat registration fails for some reasons per kickstart file, version lock might not be installed and you may land pulling the latest bits from Red Hat web site.

c. Check for existence of `versionlock.conf` and `versionlock.list` in `/etc/yum/pluginconf.d/versionlock.list`. `yum versionlock list` command should reveal the contents for `/etc/yum/pluginconf.d/versionlock.list`.

d. Run `ifconfig` to check the health of the configured interfaces. The pxe should not have been configured at this stage.

e. Check name resolution and external connectivity. This is needed for yum updates and registration.

f. Validate by running `wget www.cisco.com` or `wget subscription.rhn.redhat.com`

- g. Refer [bug 1178497](#). This bug is not in the main stream, at the time of writing this document. Please follow the workaround steps in the bug and reboot the kernel.

Take a backup of `/boot/initramfs<kernel>` to revert back in case something goes wrong:

edit the `/usr/lib/dracut/modules.d/99shutdown/module-setup.sh` and files `/usr/lib/dracut/modules.d/99shutdown/shutdown.sh` after taking a backup of these files.

`module-setup.sh`

Change

```
inst_multiple umount poweroff reboot halt losetup
```

to

```
inst_multiple umount poweroff reboot halt losetup stat
```

`shutdown.sh`

insert a block of code

after `./lib/dracut-lib.sh`

add:

```
if [ "$(stat -c '%T' -f /)" = "tmpfs" ]; then
    mount -o remount,rw /
fi
```

Recreate initramfs :

```
dracut --force
```

Unmask the shutdown :

```
systemctl unmask dracut-shutdown.service
```

Reboot the node

This completes the OS Installation on the Director node.

# Undercloud Setup

---

## Undercloud Installation

To install Undercloud, complete the following steps:

1. Create Stack User:

If Stack user was not created as part of the install earlier, it has to be created for the Undercloud now.

```
useradd stack
passwd stack
echo "stack ALL=(root) NOPASSWD:ALL" | tee -a /etc/sudoers.d/stack
chmod 0440 /etc/sudoers.d/stack
```

2. Update sysctl.conf ( as root user ):

```
echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
sysctl -p /etc/sysctl.conf
```

3. Become Stack User and create the following:

```
su - stack
mkdir -p ~/images
mkdir -p ~/templates
sudo hostnamectl set-hostname <FQDN of the director node > as an example
sudo hostnamectl set-hostname osp7-director.cisco.com
sudo hostnamectl set-hostname --transient osp7-director.cisco.com
```

4. Update /etc/hosts:

```
sudo vi /etc/hosts as below
#External Interface
172.22.215.9    osp7-director.cisco.com osp7-director
# pxe interface
10.22.110.26   osp7-director.cisco.com osp7-director
# Internal API
10.22.100.26   osp7-director.cisco.com osp7-director
# local
127.0.0.1     localhost localhost.localdomain localhost4 localhost4.localdomain4
```

5. Update resolv.conf if needed:

```
sudo vi /etc/resolv.conf as needed. As an example
search cisco.com
nameserver 8.8.8.8
```



It is recommended to use your organization DNS server. name server 8.8.8.8 is used here for reference purpose only.

---

6. In case you have not registered the Undercloud node as part of versionlock earlier, please register the system to Red Hat Network and get the appropriate pool id for Open stack entitlements and attach the pool.

7. Disable and enable only the required repositories:

```
sudo subscription-manager repos --disable=*
sudo subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-7-server-optional-rpms --enable=rhel-7-server-extras-rpms \
--enable=rhel-7-server-openstack-7.0-rpms \
--enable=rhel-7-server-openstack-7.0-director-rpms
```

8. Yum Update the server:

```
sudo yum update -y
sudo yum install yum-plugin-priorities -y
sudo yum install yum-utils -y
sudo yum-config-manager --enable rhel-7-server-openstack-7.0-rpms \
--setopt="rhel-7-server-openstack-7.0-rpms.priority=1"
sudo yum-config-manager --enable rhel-7-server-extras-rpms \
--setopt="rhel-7-server-extras-rpms.priority=1"
sudo yum-config-manager --enable rhel-7-server-openstack-7.0-director-rpms \
--setopt="rhel-7-server-openstack-7.0-director-rpms.priority=1"
sudo yum-config-manager --enable rhel-7-server-optional-rpms \
--setopt="rhel-7-server-optional-rpms.priority=1"
sudo yum-config-manager --enable rhel-7-server-rpms \
--setopt="rhel-7-server-rpms.priority=1"
```

9. Install Undercloud packages:



Make sure that Versionlock is in place by running `yum versionlock list`.

---

```
sudo yum install -y python-rdomanager-oscplugin
sudo yum update -y
```

10. Create `undercloud.conf` file:

```
cp /usr/share/instack-undercloud/undercloud.conf.sample ~/undercloud.conf
```

The following are the values used in the configuration. 10.22.110 is the pxe network:

```
image_path = /home/stack/images
local_ip = 10.22.110.26/24
undercloud_public_vip = 10.22.110.27
undercloud_admin_vip = 10.22.110.28
local_interface = eth0
masquerade_network = 10.22.110.0/24
dhcp_start = 10.22.110.51
dhcp_end = 10.22.110.80
network_cidr = 10.22.110.0/24
network_gateway = 10.22.110.26
discovery_interface = br-ctlplane
discovery_iprange = 10.22.110.81,10.22.110.110
undercloud_debug = true
```



By using the provisioning interface on the director node and the `local_ip` and `network_gateway`, it configures the system to act as the gateway for all the nodes.

---

11. Update `enic` driver:

- a. Download the enic driver Cisco appropriate for the UCSM version. The version used in the configuration with UCSM 2.2(5) and RHEL 7.2 was 2.1.1.93: go to <http://software.cisco.com/download/navigator.html>
- b. In the download page, select servers-Unified computing under products. On the right menu select your class of servers say Cisco UCS B-series Blade server software and then select Unified Computing System (UCS) Drivers in the following page.
- c. Select your firmware version under All Releases, as an example 2.2(5c) and download the ISO image of UCS-related drivers for your matching firmware, for example ucs-bxxx-drivers.2.2.5c.iso.
- d. Download the iso file to your undercloud machine and mount the iso:

```
[root@osp7-director ~]# mount ucs-bxxx-drivers.2.2.5.iso /mnt
mount: /dev/loop0 is write-protected, mounting read-only
cd /mnt/Linux/Network/Cisco/VIC/RHEL/RHEL7.2
cp kmod-enic-2.1.1.93-rhel7u2.el7.x86_64.rpm /tmp
rpm -ivh kmod-enic-2.1.1.93-rhel7u2.el7.x86_64.rpm

umount /mnt
```

- e. Install the appropriate enic driver on the director machine.
- f. Validate by running modinfo:

```
[root@osp7-director RHEL7.1]# modinfo enic

filename:       /lib/modules/3.10.0-327.3.1.el7.x86_64/weak-
updates/enic/enic.ko
version:       2.1.1.93
license:       GPL v2
author:        Scott Feldman <scofeldm@cisco.com>
description:   Cisco VIC Ethernet NIC Driver
rhelversion:   7.2
srcversion:    D272F11F27065C9714656F4
alias:         pci:v00001137d00000071sv*sd*bc*sc*i*
alias:         pci:v00001137d00000044sv*sd*bc*sc*i*
alias:         pci:v00001137d00000043sv*sd*bc*sc*i*
depends:
vermagic:     3.10.0-327.el7.x86_64 SMP mod_unload modversions
parm:         rxcopybreak:Maximum size of packet that is copied to a new
buffer on receive (uint)
```

- g. Copy the enic file to your ~/images directory created above.

```
cp /lib/modules/3.10.0-327.3.1.el7.x86_64/weak-updates/enic/enic.ko ~/images
[root@osp7-director RHEL7.1]# ls -l /home/stack/images/enic.ko
-rw-r--r--. 1 stack stack 3982019 Jan 16 21:38 /home/stack/images/enic.ko
```



This enic.ko file will be used to customize Overcloud image file later.

---

12. Download libguestfs tool needed to customize Overcloud image file:

```
sudo yum install libguestfs-tools -y
```

The system is ready for running the Undercloud install now.

13. Run the following as stack user:

```
cd /home/stack
openstack undercloud install
```

This might take around 10 minutes.



To debug any Undercloud install failures, check files in `/home/stack/.instack/*`

## Post Undercloud Installation Checks

To perform the Undercloud installation checks, complete the following steps:

1. Check `sysctl.conf`:

```
cat /etc/sysctl.conf
net.ipv4.ip_forward=1
net.ipv4.ip_nonlocal_bind=1
```

2. Check control plane bridge:

A new bridge `br-ctlplane` should have been created as part of the Undercloud install on the pxe interface as shown below. Validate MAC and IP's.

```
br-ctlplane: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.22.110.26 netmask 255.255.255.0 broadcast 10.22.110.255
    inet6 fe80::225:b5ff:fe22:222f prefixlen 64 scopeid 0x20<link>
    ether 00:25:b5:22:22:2f txqueuelen 0 (Ethernet)
    RX packets 180797219 bytes 31192730719 (29.0 GiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 180599150 bytes 108428706507 (100.9 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[stack@osp7-director ~] 2016-02-24 1306$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::225:b5ff:fe22:222f prefixlen 64 scopeid 0x20<link>
    ether 00:25:b5:22:22:2f txqueuelen 1000 (Ethernet)
    RX packets 180832402 bytes 32642175196 (30.4 GiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 230154300 bytes 112625223587 (104.8 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



Check `/var/log/ironic/*` files, to understand fix any issues.

3. Update `/etc/openstack-dashboard`:

- a. **Update `allowed_hosts` to ip's needed to launch** Undercloud UI and restart `httpd` service on Undercloud node.

```
[stack@osp7-director ~]$ sudo -i
[root@osp7-director ~]# vi /etc/openstack-dashboard/local_settings
ALLOWED_HOSTS = ['172.22.215.9', '10.22.110.26', 'osp7-director.cisco.com',
'localhost', ]
[root@osp7-director ~]# systemctl restart httpd.service
```

- b. Log into **the Undercloud dashboard from one of the IP's above and do a sanity check**. Log in as user admin. The default password can be obtained from /home/stack/stackrc (run `sudo hiera admin_password`).

# Introspection

## Pre-Installation Checks for Introspection

To do the pre-installation check, complete the following steps:

Login to the director node as stack user and source stackrc file

1. Check neutron and subnet lists:

a. Run `neutron net-list`, `neutron net-show`, `neutron subnet-list` and `neutron subnet-show` for br-ctlplane:

```
[stack@osp7-director ~]$ neutron net-show 1282955a-4ff2-4d3a-a9ea-5c50e11979f5
+-----+
| Field                | Value                |
+-----+-----+
| admin_state_up      | True                 |
| id                   | 1282955a-4ff2-4d3a-a9ea-5c50e11979f5 |
| mtu                   | 0                    |
| name                  | ctlplane             |
| provider:network_type | flat                 |
| provider:physical_network | ctlplane            |
| provider:segmentation_id |                     |
| router:external      | False                |
| shared                | False                |
| status                | ACTIVE               |
| subnets              | 280ba04c-41fd-4686-ac18-1c6e7cc9325a |
| tenant_id            | 04fce581bc1f40c2a357ae6da1de0e |
+-----+-----+
[stack@osp7-director ~]$ neutron subnet-list
+-----+-----+-----+-----+
| id                   | name                 | cidr                | allocation_pools    |
+-----+-----+-----+-----+
| 280ba04c-41fd-4686-ac18-1c6e7cc9325a | | 10.22.110.0/24     | {"start": "10.22.110.51", "end": "10.22.110.80"} |
+-----+-----+-----+-----+
[stack@osp7-director ~]$ neutron subnet-show 280ba04c-41fd-4686-ac18-1c6e7cc9325a
+-----+
| Field                | Value                |
+-----+-----+
| allocation_pools     | {"start": "10.22.110.51", "end": "10.22.110.80"} |
| cidr                  | 10.22.110.0/24      |
| dns_nameservers      |                     |
| enable_dhcp           | True                 |
| gateway_ip           | 10.22.110.26        |
| host_routes           | {"destination": "169.254.169.254/32", "nexthop": "10.22.110.26"} |
| id                    | 280ba04c-41fd-4686-ac18-1c6e7cc9325a |
| ip_version            | 4                    |
| ipv6_address_mode     |                     |
| ipv6_ra_mode          |                     |
| name                  |                     |
| network_id            | 1282955a-4ff2-4d3a-a9ea-5c50e11979f5 |
| subnetpool_id         |                     |
| tenant_id             | 04fce581bc1f40c2a357ae6da1de0e |
+-----+-----+
```



The `allocation_pools`, `dns_nameservers`, `cidr` should match whatever specified earlier in `undercloud.conf` file. If not, update with `neutron subnet-update`.

```
[stack@osp7-director ~]$ neutron subnet-update 280ba04c-41fd-4686-ac18-1c6e7cc9325a --name ctlplane-subnet --dns-nameserver 8.8.8.8
Updated subnet: 280ba04c-41fd-4686-ac18-1c6e7cc9325a
[stack@osp7-director ~]$ neutron net-list
+-----+-----+-----+-----+
| id | name | subnets |
+-----+-----+-----+-----+
| 1282955a-4ff2-4d3a-a9ea-5c50e11979f5 | ctlplane | 280ba04c-41fd-4686-ac18-1c6e7cc9325a | 10.22.110.0/24 |
+-----+-----+-----+-----+
[stack@osp7-director ~]$ neutron subnet-show 280ba04c-41fd-4686-ac18-1c6e7cc9325a
+-----+-----+
| Field | Value |
+-----+-----+
| allocation_pools | {"start": "10.22.110.51", "end": "10.22.110.80"} |
| cidr | 10.22.110.0/24 |
| dns_nameservers | 8.8.8.8 |
| enable_dhcp | True |
| gateway_ip | 10.22.110.26 |
| host_routes | {"destination": "169.254.169.254/32", "nexthop": "10.22.110.26"} |
| id | 280ba04c-41fd-4686-ac18-1c6e7cc9325a |
| ip_version | 4 |
| ipv6_address_mode | |
| ipv6_ra_mode | |
| name | ctlplane-subnet |
| network_id | 1282955a-4ff2-4d3a-a9ea-5c50e11979f5 |
| subnetpool_id | |
| tenant_id | 04fce581bc1f40c2a357aef6da1de0e |
+-----+-----+
```

## 2. Check /etc/ironic-discovered/\* files:

```
vi /etc/ironic-discoverd/discoverd.conf /etc/ironic-discoverd/dnsmasq.conf
```

The dnsmasq.conf dhcp\_range should match the undercloud.conf file range. This will help you spot any errors that might have gone while running Undercloud install earlier. The default pxe timeout is 60 minutes in Kilo. This means if you have more servers to be introspected and it takes longer than 60 minutes, introspection is bound to fail.

### a. Update /etc/ironic-discoverd/discoverd.conf with timeout variable under discovered section:

```
timeout=0
```

### b. Restart ironic in case these files are updated.

```
[root@osp7-director ~]# systemctl restart openstack-ironic-conductor.service
openstack-ironic-discoverd-dnsmasq.service openstack-ironic-api.service
openstack-ironic-discoverd.service
```



This may be necessary only in larger deployments and depends on the network to download the ramdisk files, CPU speed, etc. On the configuration this issue was discovered while introspecting 35 nodes.

---

## 3. Prepare instack.json file.



This file should contain all the nodes, controllers, computes and storage nodes that need to be introspected.

---

A sample instackenv.json file is provided in Appendix A. Below is an explanation of how to build this file for a node.

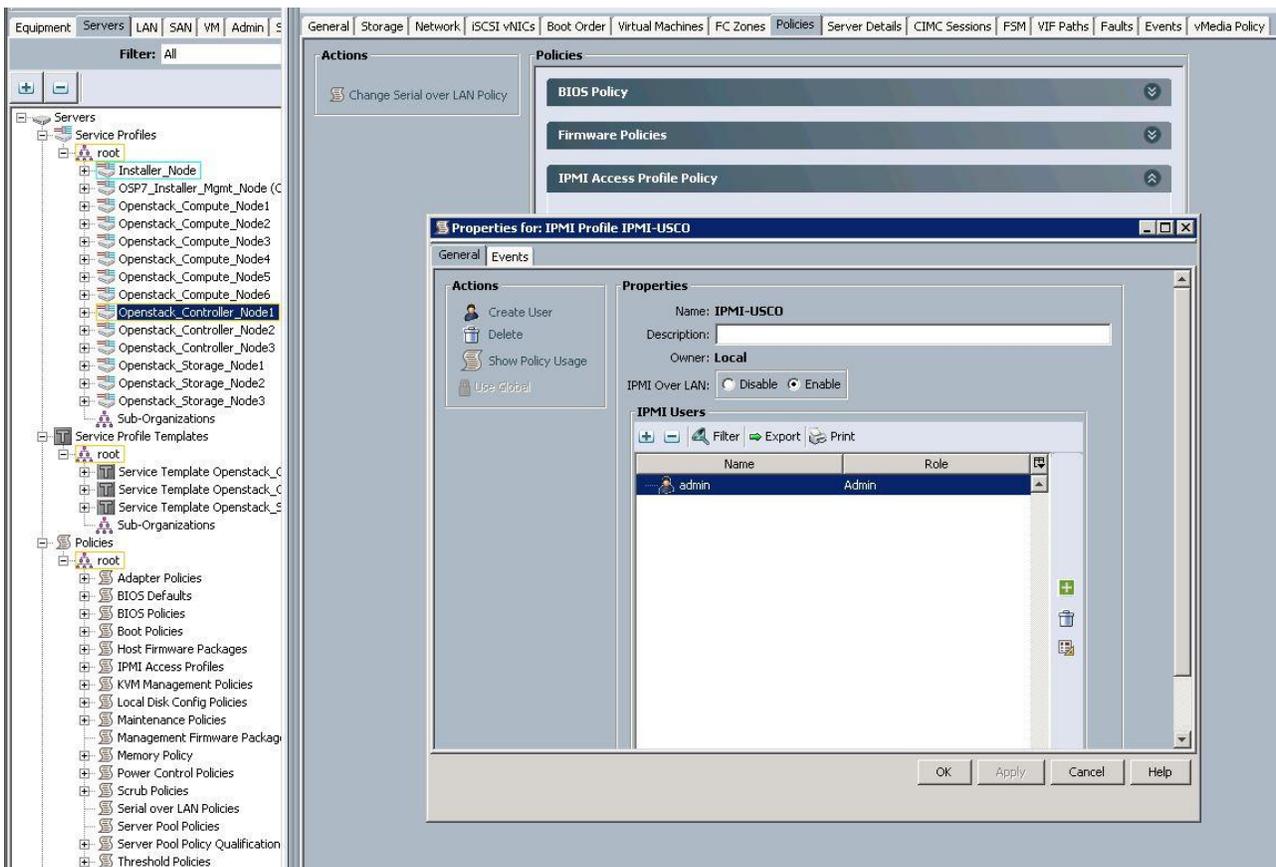
```
"nodes": [
  {
    "pm_user": "admin",
    "pm_password": "password",
    "pm_type": "pxe_ipmitool",
    "pm_addr": "10.22.100.19",
```

```
"mac": [
  "00:25:b5:00:00:2d"
],
"memory": "262144",
"disk": "270",
"arch": "x86_64",
"cpu": "40"
},
.....
.....
```

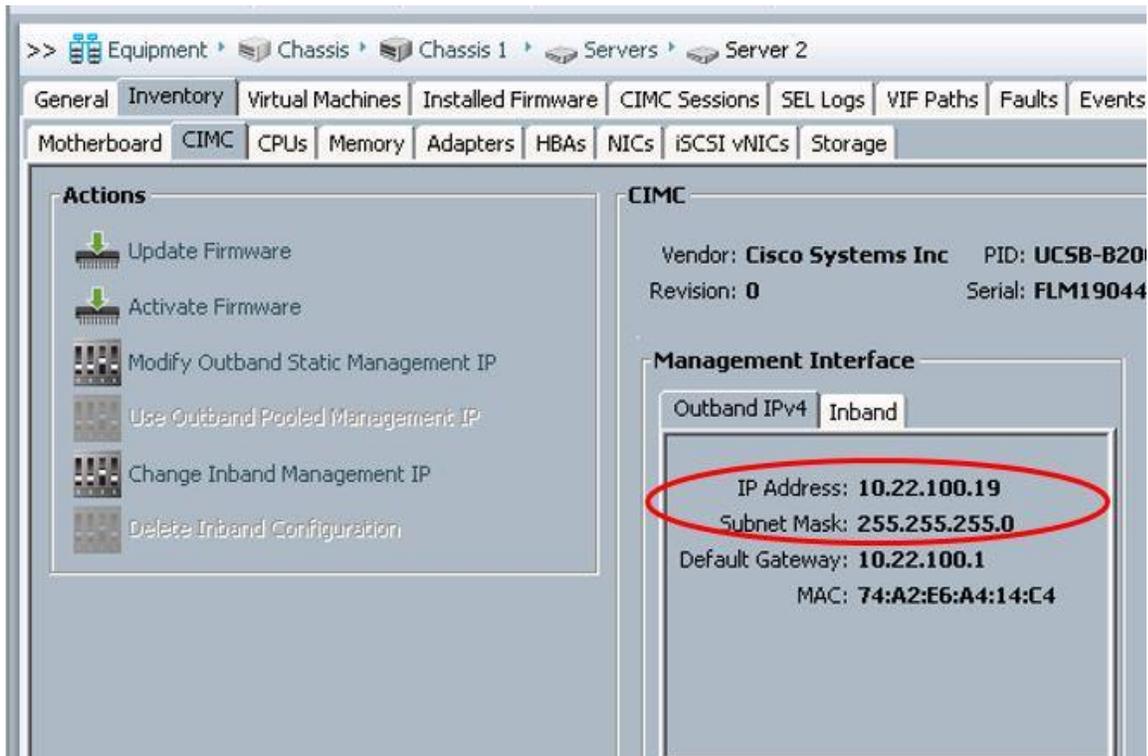
pm\_user and pm\_password



This is the ipmi user and password configured earlier for this node's service profile or templates.



pm\_type="pxe\_ipmitool" Leave this, as is  
 pm\_addr is the IPMI address allocated to that node. This can be obtained from the CIMC tab in equipment.



The MAC address is the discovery nic or pxe interface for that node.

Name	vNIC	Vendor	PID	Model	Operability	MAC	Original MAC
NIC 1	PXE	Cisco Systems Inc	UCSB-MLOM-40G-03	Cisco UCS VIC 1340	Operable	00:25:B5:00:00:2D	00:00:00:00:00:00
NIC 2	eth1	Cisco Systems Inc	UCSB-MLOM-40G-03	Cisco UCS VIC 1340	Operable	00:25:B5:00:00:2E	00:00:00:00:00:00

The memory, disk and CPU can be obtained under the same inventory tab for that node.



Make sure that the storage lun is applied and in operable state after applying the storage policy.

Name	Size (MB)	Raid Type	Config State	Operability	Presence
Controller SAS 1					
Virtual Drive Boot-LUN	256000	RAID 1 Mirrored	Applied	Operable	Equipped

Build the instackenv.json file for all the hosts that have to be introspected as above.



Make sure to maintain consistent indentations with white spaces or tabs.

4. Check the ipmi connectivity works for all the hosts:

- a. You can run a quick check to validate this from instackenv.json file:

```
[stack@osp7-director ~]$ for i in `grep pm_addr instackenv.json | cut -d "\"" -f4`
do
ipmitool -I lanplus -H $i -U admin -P <replace with your ipmi password> chassis
power status
done
Chassis Power is off
```




---

The user 'admin' used above, is the IPMI admin user configured earlier in UCS.

---

- b. The chassis power status should be either On or Off depending on whether the server is up or down in UCS. However any errors like the example shown below need investigation:

```
Error: Unable to establish IPMI v2 / RMCP+ session
Unable to get Chassis Power Status
```

5. Install ntp server and synchronize the clock:

- a. As root user;

```
yum install ntp -y
```

6. Update /etc/ntp.conf file with appropriate ntp server address and restart ntpd;

```
[root@osp7-director ~]# service ntpd restart
Redirecting to /bin/systemctl restart ntpd.service
```

7. Check the time sync, else restart ntpd couple of times to force sync the time:

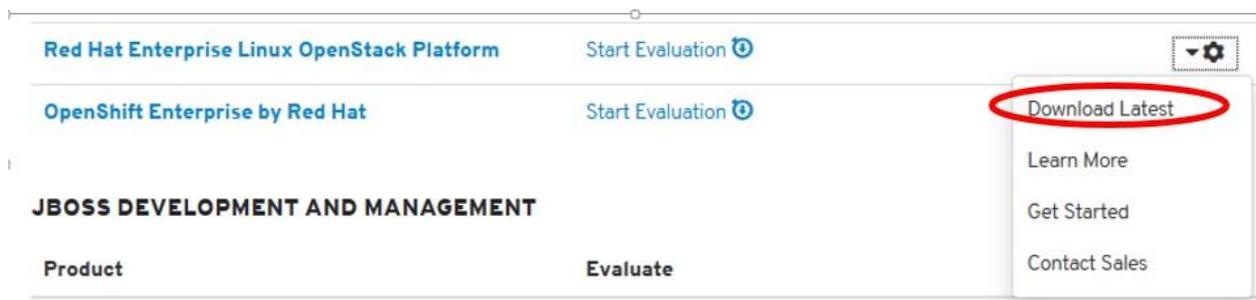
```
service ntpd restart
[root@osp7-director ~]# ntpdate -dv 171.68.38.66
16 Jan 13:13:25 ntpdate[16252]: ntpdate 4.2.6p5@1.2349-o Fri Oct 16 08:51:51
UTC 2015 (1)
Looking for host 171.68.38.66 and service ntp
host found : mtv5-ai27-dcm10n-ntp2.cisco.com
transmit(171.68.38.66)
receive(171.68.38.66)
.....
.....
.....
delay 0.02707, dispersion 0.00000
offset 0.000085
```



The clock is synchronized to 85 micro seconds now. Usually a clock sync of less than 20 minutes is needed to avoid any health\_warn complaints from Ceph monitors.

8. Download the Image files needed for introspection and Overcloud:

- a. Login to <http://access.redhat.com>.
- b. Click Downloads. Under Product Downloads by category and in cloud products, select Red Hat Enterprise Linux OpenStack Platform to download the Overcloud image files. Select overcloud image for 7.2 and download to the images directory.



#### Installers and Images for Red Hat OpenStack (v. 7 for x86\_64)

<b>Deployment Ramdisk for RHEL-OSP director 7.2</b> Last modified: 2015-12-21    SHA-256 Checksum: a3e8ef32264294efdac996d97e3102a934802f56d097da86a888004beef6b71	<a href="#">Download Now</a> 61.7 MB
<b>Overcloud Image for RHEL-OSP director 7.2</b> Last modified: 2015-12-21    SHA-256 Checksum: 48577a63991a99df054c6445e85a5f00731fe7f3e6080d59d8e8d872330c92cf	<a href="#">Download Now</a> 977 MB
<b>Discovery Ramdisk for RHEL-OSP director 7.2</b> Last modified: 2015-12-21    SHA-256 Checksum: ce18ddf63c929f82bad6a816ad4db349e85f3e57f8a1c9d4b1ba5f83ee7843e	<a href="#">Download Now</a> 152 MB

- c. Download Deployment Ramdisk, Overcloud Image and Discovery Ramdisk for Red Hat Linux Director for 7.2. The solution is validated on 7.2. Customizations and interoperability of these files with Cisco Plugins were done with 7.2. In case of higher versions posted in this web page, contact Red Hat for getting 7.2 images.



The images can be downloaded directly from Director Host as a GUI install was done on the Director node by launching a browser or doing a wget of the download links above.

- d. Download the files into /home/stack/images directory, extract the tar files;

```
cd /home/stack/images; for i in *.tar; do tar xf $i; done
```

- e. The following files should exist after extraction:

```
[stack@osp7-director images]$ /bin/ls -l
-rw-r--r-- 1 stack stack 1048576 Dec 21 2015 deploy-ramdisk-ironic.initramfs
-rw-r--r-- 1 stack stack 1048576 Dec 21 2015 deploy-ramdisk-ironic.kernel
-rw-r--r-- 1 stack stack 1048576 Dec 21 2015 discovery-ramdisk.initramfs
-rw-r--r-- 1 stack stack 1048576 Dec 21 2015 discovery-ramdisk.kernel
```

```
overcloud-full.initrd
overcloud-full.qcow2
overcloud-full.vmlinuz
```



You may remove the tar files if desired.

- f. Download the Guest image to the images directory.
- g. Under Product downloads above, select Red Hat Enterprise Linux under infrastructure Management and download KVM Guest Image for 7.2.

#### KVM Guest Image

Last modified: 2015-11-19 SHA-256 Checksum: 25f880767ec6bf71beb532e17ffc45231640bbfdffbb1dffb79d2c1b328388e0

Download Now

453 MB

```
rhel-guest-image-7.2-20151102.0.x86_64.qcow2
```

9. Customize the Overcloud image with n1kv modules, enic drivers and fencing packages.

Run the following as root user. Navigate to your download directory and issue the following as root:

```
cd /home/stack/images
export LIBGUESTFS_BACKEND=direct
```

- a. Update fencing packages.



Before proceeding with the customization of the Overcloud image, there are some fixes that are not part of the osp7 y2 distribution. Refer to [bug 1298430](#).

Download the fencing packages from [http://people.redhat.com/cfeist/cisco\\_ucs/](http://people.redhat.com/cfeist/cisco_ucs/) to ~/images/ directory. These packages are being integrated to mainstream and we will update the document when they are available in the Red Hat repository.

**Extract the fencing files from these rpm's as shown below:**

```
rpm2cpio <name of the fence agents common rpm file > | cpio -idmv
rpm2cpio <name of the fence agents cisco ucs rpm file > | cpio -idmv
```

This should create a local usr/share directory. The following two files need to be copied:

```
cp ./usr/share/fence/fencing.py /home/stack/images
cp ./usr/sbin/fence_cisco_ucs /home/stack/images
```

These two files will be used to update the overcloud image.

As root user;

```
cd /home/stack/images
chmod +x ./fencing.py
chmod +x ./fence_cisco_ucs
chown root:root fenc*
virt-copy-in -a overcloud-full.qcow2 ./fencing.py /usr/share/fence/
virt-copy-in -a overcloud-full.qcow2 ./fence_cisco_ucs /sbin/
```

```

Maybe you can virt-copy-out and validate that the files have been uploaded
properly by extracting them to say /tmp location
virt-copy-out -a overcloud-full.qcow2 /usr/share/fence/fencing.py /tmp
virt-copy-out -a overcloud-full.qcow2 /sbin/fence_cisco_ucs /tmp

```

b. Update Grub file;

```

virt-copy-out -a overcloud-full.qcow2 /etc/default/grub /home/stack/images/
vi grub file and change the following line
GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8 crashkernel=auto rhgb
quiet net.ifnames=0 biosdevname=0"
---(you are appending net.ifnames=0 and biosdevname=0)
virt-copy-in -a overcloud-full.qcow2 ./grub /etc/default/

```



After this proceed, with the the remaining customizations.

---

c. Update enic drivers;

```

virt-copy-in -a overcloud-full.qcow2 ./enic.ko /lib/modules/3.10.0-
327.3.1.el7.x86_64/kernel/drivers/net/ethernet/cisco/enic/

```

The location of this enic driver is dependent on the kernel packaged in the Overcloud image file. Should be changed if needed.

d. Update root password;

```

virt-customize -a overcloud-full.qcow2 --root-password password:<password>

```

e. Install n1kv modules;

```

echo "[n1kv]
name=n1kv
baseurl=https://cmsg-yum-server.cisco.com/yumrepo/
enabled=1
gpgcheck=0" > n1kv.repo

```

```

virt-customize -a overcloud-full.qcow2 --upload n1kv.repo:/etc/yum.repos.d/
rm n1kv.repo
virt-customize -a overcloud-full.qcow2 --install nexus1000v
virt-customize -a overcloud-full.qcow2 --install nexus-1000v-iso

```

f. Change the permissions back to stack user:

```

chown stack:stack /home/stack/images/*

```

This is how the overcloud-full.qcow2 may look after the update:

```

[stack@osp7-director images]$ ls -l overcloud-full.qcow2*

```

```

-rw-r--r--. 1 stack stack 1404108800 Feb  5 15:50 overcloud-full.qcow2

```

While updating, the image with root password is not required; it becomes useful to login through KVM console in case of Overcloud installation failures and debug the issues.

The enic.ko was extracted earlier on the Directory node after installing the enic rpm. This helps ensure that both Director and the Overcloud images will be with same enic driver.

The N1000V modules will be injected and installed in the Overcloud image. As part of deployment, the VSM module will be installed on the Controller nodes, while VEM will be installed on all Controller and Compute nodes, due to the update to the Overcloud image.

**The grub has been modified to have interface names like eth[0], eth[1] ...**

The fence\_cisco\_ucs package has been modified to take care of the HA [bug 1298430](#).

10. Upload the images to openstack. As stack user run the following:

```
su - stack
source stackrc
cd ~/images
openstack overcloud image upload
openstack image list
```

```
[stack@osp7-director images]$ openstack image list
```

ID	Name
404d0e44-e5c7-4b46-8339-c451441b3f55	bm-deploy-ramdisk
ca9667c9-de8f-4df1-95d5-b5f8b6fb8f4d	bm-deploy-kernel
7454a151-ab14-4989-af7e-28f06a94d4bc	overcloud-full
59e75e99-e9bd-47eb-b088-a166fc855c11	overcloud-full-initrd
9e4da0bf-36e3-4973-a10f-210f2df108e2	overcloud-full-vmlinuz

11. Initialize boot LUNs. There is no need to initialize the SSD and OSD luns on Ceph nodes as this will be taken care by wipe\_disk.yaml file included in the templates ( included through network-environment.yaml file).
12. Before running Introspection and Overcloud installation, it is recommended to initialize the boot LUNs. This is required in case you are repeating or using old disks.
13. Boot the server in UCS, press CTRL-R, then F2 and re-initialize the boot LUNs as shown below and shutdown the servers.

```

0 SEAGATE ST6000NM0014 K0B1 5723166MB
0 TOSHIBA PX02SMF040 0205 381554MB
0 TOSHIBA MG03SCA400 5701 3815447MB
0 TOSHIBA MG03SCA400 5702 3815447MB
0 TOSHIBA PX02SMF040 0205 381554MB
0 LSI Virtual Drive RAID1 409600MB
1 LSI Virtual Drive RAID0 358400MB
2 LSI Virtual Drive RAID0 358400MB
3 LSI Virtual Drive RAID0 5632000MB
4 LSI Virtual Drive RAID0 5632000MB
5 LSI Virtual Drive RAID0 5632000MB
6 LSI Virtual Drive RAID0 5632000MB
7 LSI Virtual Drive RAID0 5632000MB
8 LSI Virtual Drive RAID0 5632000MB
9 LSI Virtual Drive RAID0 5632000MB
10 LSI Virtual Drive RAID0 5632000MB

JBOD(s) found on the host adapter
JBOD(s) handled by BIOS

Virtual Drive(s) found on the host adapter.

Virtual Drive(s) handled by BIOS
Press <Ctrl><R> to Run MegaRAID Configuration Utility

```

Cisco 12G SAS Modular Raid Controller BIOS Configuration Utility 5.08-0009

UD Mgmt PD Mgmt Ctrl Mgmt Properties

Virtual Drive Management

[ - ] Cisco 12G SAS Modular Raid(Bus 0x09, Dev 0x00)

- [ - ] Drive Group: 0, RAID 1
  - [ - ] Virtual Drives
    - ID: 0, bootlun, 400.00 GB
  - [ + ] Drives
  - [ + ] Available size: 3.24 TB
  - Hot spare drives
- [ - ] Drive Group: 1, RAID 0
  - [ - ] Virtual Drives
    - ID: 1, Ceph-SSD2, 350.00 GB
  - [ + ] Drives
  - [ + ] Available size: 21.59 GB
  - Hot spare drives
- [ - ] Drive Group: 2, RAID 0
  - [ - ] Virtual Drives
    - ID: 2, Ceph-SSD1, 350.00 GB
  - [ + ] Drives
  - [ + ] Available size: 21.59 GB
  - Hot spare drives
- [ - ] Drive Group: 3, RAID 0
  - [ - ] Virtual Drives

Virtual Drive 0:  
State: Optimal

Initialization  
Consistency Check

Delete UD

Properties

Expand UD size

Erase UD  
Hide UD

Start FGI  
Stop Init  
Fast Init

Suspend BGI  
Resume BGI  
Stop BGI

F1-Help F2-Operations F5-Refresh Ctrl-N-Next Page Ctrl-P-Prev Page F12-Ctrl



Make sure that all the servers are powered off before introspection.

14. Reboot the Undercloud node and start the introspection.

## Run Introspection

To run Introspection, complete the following steps:

1. As stack user:

```
source ~/stackrc
openstack baremetal import --json ~/instackenv.json
openstack baremetal configure boot
openstack baremetal list
```

```
[stack@osp7-director ~]$ openstack baremetal list
```

UUID	Name	Instance UUID	Power State	Provision State	Maintenance
b7dde876-354a-4688-8550-aec8f64c582c	None	None	power off	available	False
e4563ca5-2f12-4e08-9905-f770f740ad2b	None	None	power off	available	False
285965a9-9713-4301-8ad5-7aa3ef5dd1c2	None	None	power off	available	False
b4dc04ac-0c69-4000-9c4d-2d82d141905f	None	None	power off	available	False
036cae70-bdee-427c-987c-a6a2d8a32292	None	None	power off	available	False
8570c96e-f9cd-44ff-ald8-0252bc405c24	None	None	power off	available	False
af46cd81-c78e-47c5-94e3-44d9d669410c	None	None	power off	available	False
19260dbb-29a9-4810-b39d-85cc6e1d886f	None	None	power off	available	False
d4dae332-4595-43be-9b63-5a64331ea33b	None	None	power off	available	False
179befe6-2510-4311-ad9f-4880454fdaff	None	None	power off	available	False
ff0dadfe-e2f3-408f-b69d-01398bb9699d	None	None	power off	available	False
b59f57e3-d5e1-499a-80c1-aac0c78c9534	None	None	power off	available	False

```
openstack baremetal introspection bulk start
```

```

+-----+-----+-----+-----+
[stack@osp7-director ~]$ openstack baremetal introspection bulk start
Setting available nodes to manageable...
Starting introspection of node: b7dde876-354a-4688-8550-aec8f64c582c
Starting introspection of node: e4563ca5-2f12-4e08-9905-f770f740ad2b
Starting introspection of node: 285965a9-9713-4301-8ad5-7aa3ef5dd1c2
Starting introspection of node: b4dc04ac-0c69-4000-9c4d-2d82d141905f
Starting introspection of node: 036cae70-bdee-427c-987c-a6a2d8a32292
Starting introspection of node: 8570c96e-f9cd-44ff-a1d8-0252bc405c24
Starting introspection of node: af46cd81-c78e-47c5-94e3-44d9d669410c
Starting introspection of node: 19260dbb-29a9-4810-b39d-85cc6e1d886f
Starting introspection of node: d4dae332-4595-43be-9b63-5a64331ea33b
Starting introspection of node: 179befe6-2510-4311-ad9f-4880454fdaff
Starting introspection of node: ff0dadfe-e2f3-408f-b69d-01398bb9699d
Starting introspection of node: b59f57e3-d5e1-499a-80c1-aac0c78c9534
Waiting for discovery to finish...
Discovery for UUID e4563ca5-2f12-4e08-9905-f770f740ad2b finished successfully.
Discovery for UUID b4dc04ac-0c69-4000-9c4d-2d82d141905f finished successfully.
Discovery for UUID 8570c96e-f9cd-44ff-a1d8-0252bc405c24 finished successfully.
Discovery for UUID 19260dbb-29a9-4810-b39d-85cc6e1d886f finished successfully.
Discovery for UUID 179befe6-2510-4311-ad9f-4880454fdaff finished successfully.
Discovery for UUID b7dde876-354a-4688-8550-aec8f64c582c finished successfully.
Discovery for UUID 036cae70-bdee-427c-987c-a6a2d8a32292 finished successfully.
Discovery for UUID d4dae332-4595-43be-9b63-5a64331ea33b finished successfully.
Discovery for UUID b59f57e3-d5e1-499a-80c1-aac0c78c9534 finished successfully.
Discovery for UUID 285965a9-9713-4301-8ad5-7aa3ef5dd1c2 finished successfully.
Discovery for UUID ff0dadfe-e2f3-408f-b69d-01398bb9699d finished successfully.
Discovery for UUID af46cd81-c78e-47c5-94e3-44d9d669410c finished successfully.
Setting manageable nodes to available...
Node b7dde876-354a-4688-8550-aec8f64c582c has been set to available.
Node e4563ca5-2f12-4e08-9905-f770f740ad2b has been set to available.
Node 285965a9-9713-4301-8ad5-7aa3ef5dd1c2 has been set to available.
Node b4dc04ac-0c69-4000-9c4d-2d82d141905f has been set to available.
Node 036cae70-bdee-427c-987c-a6a2d8a32292 has been set to available.
Node 8570c96e-f9cd-44ff-a1d8-0252bc405c24 has been set to available.
Node af46cd81-c78e-47c5-94e3-44d9d669410c has been set to available.
Node 19260dbb-29a9-4810-b39d-85cc6e1d886f has been set to available.
Node d4dae332-4595-43be-9b63-5a64331ea33b has been set to available.
Node 179befe6-2510-4311-ad9f-4880454fdaff has been set to available.
Node ff0dadfe-e2f3-408f-b69d-01398bb9699d has been set to available.
Node b59f57e3-d5e1-499a-80c1-aac0c78c9534 has been set to available.
Discovery completed.
[stack@osp7-director ~]$ █

```

## 2. Check the status of Introspection:

```
openstack baremetal introspection bulk status
```

```

[stack@osp7-director ~]$ openstack baremetal introspection bulk status
+-----+-----+-----+-----+
| Node UUID                                | Finished | Error |
+-----+-----+-----+-----+
| b7dde876-354a-4688-8550-aec8f64c582c    | True     | None  |
| e4563ca5-2f12-4e08-9905-f770f740ad2b    | True     | None  |
| 285965a9-9713-4301-8ad5-7aa3ef5dd1c2    | True     | None  |
| b4dc04ac-0c69-4000-9c4d-2d82d141905f    | True     | None  |
| 036cae70-bdee-427c-987c-a6a2d8a32292    | True     | None  |
| 8570c96e-f9cd-44ff-a1d8-0252bc405c24    | True     | None  |
| af46cd81-c78e-47c5-94e3-44d9d669410c    | True     | None  |
| 19260dbb-29a9-4810-b39d-85cc6e1d886f    | True     | None  |
| d4dae332-4595-43be-9b63-5a64331ea33b    | True     | None  |
| 179befe6-2510-4311-ad9f-4880454fdaff    | True     | None  |
| ff0dadfe-e2f3-408f-b69d-01398bb9699d    | True     | None  |
| b59f57e3-d5e1-499a-80c1-aac0c78c9534    | True     | None  |
+-----+-----+-----+-----+

```



Refer to the Troubleshooting section for any failures around introspection and how to resolve them.

## Create Flavors

To create Flavors, enter the following:

```
openstack flavor create --id auto --ram 8192 --disk 100 --vcpus 8 baremetal
openstack flavor create --id auto --ram 32768 --disk 100 --vcpus 8 control
openstack flavor create --id auto --ram 32768 --disk 100 --vcpus 8 compute
openstack flavor create --id auto --ram 32768 --disk 100 --vcpus 8 CephStorage
```



While creating flavors make sure that the values of disk, ram and vcpus are less than their respective servers.

```
openstack flavor set --property "cpu_arch"="x86_64" --property
"capabilities:boot_option"="local" baremetal
openstack flavor set --property "cpu_arch"="x86_64" --property
"capabilities:boot_option"="local" --property "capabilities:profile"="compute"
compute
openstack flavor set --property "cpu_arch"="x86_64" --property
"capabilities:boot_option"="local" --property "capabilities:profile"="control"
control
openstack flavor set --property "cpu_arch"="x86_64" --property
"capabilities:boot_option"="local" --property
"capabilities:profile"="CephStorage" CephStorage
```

## Set Flavors

The created Flavors have to be set to every category of servers. Identify the servers based on IPMI address created earlier in instackenv.json file:

```
[stack@osp7-director ~]$ for i in $(ironic node-list | awk ' /power/ { print $2 }
')
do
abc=`ironic node-show $i | grep "10.22" | awk '{print $7}'`
echo $i $abc
done
```

The above script will spill out the node-id along with ipmi address that can be mapped as shown below with instackenv.json file.

**Table 5 Node IDs and ipmi Addresses**

Node ID	ipmi Address	Server
b7dde876-354a-4688-8550-aec8f64c582c	u'10.22.100.23',	<--Control
e4563ca5-2f12-4e08-9905-f770f740ad2b	u'10.22.100.22',	<--Control
285965a9-9713-4301-8ad5-7aa3ef5dd1c2	u'10.22.100.16',	<--Control
b4dc04ac-0c69-4000-9c4d-2d82d141905f	u'10.22.100.18',	<--Compute
036cae70-bdee-427c-987c-a6a2d8a32292	u'10.22.100.12',	<--Compute
8570c96e-f9cd-44ff-a1d8-0252bc405c24	u'10.22.100.19',	<--Compute
af46cd81-c78e-47c5-94e3-44d9d669410c	u'10.22.100.17',	<--Compute
19260dbb-29a9-4810-b39d-85cc6e1d886f	u'10.22.100.15',	<--Compute

d4dae332-4595-43be-9b63-5a64331ea33b	u'10.22.100.14',	<--Compute
179befe6-2510-4311-ad9f-488045fdaff	u'10.22.100.20',	<--Storage
ff0dadfe-e2f3-408f-b69d-01398bb9699d	u'10.22.100.11',	<--Storage
b59f57e3-d5e1-499a-80c1-aac0c78c9534	u'10.22.100.21',	<--Storage

With the information from Table 5 , Flavors can be set as follows:

```
[stack@osp7-director ~]$ for i in b7dde876-354a-4688-8550-aec8f64c582c e4563ca5-2f12-4e08-9905-f770f740ad2b \
> 285965a9-9713-4301-8ad5-7aa3ef5dd1c2
> do
> ironic node-update $i add
properties/capabilities='profile:control,boot_option:local'
> done
```

```
[stack@osp7-director ~]$ for i in b4dc04ac-0c69-4000-9c4d-2d82d141905f 036cae70-bdee-427c-987c-a6a2d8a32292 \
> 8570c96e-f9cd-44ff-a1d8-0252bc405c24 af46cd81-c78e-47c5-94e3-44d9d669410c
19260dbb-29a9-4810-b39d-85cc6e1d886f \
> d4dae332-4595-43be-9b63-5a64331ea33b
> do
> ironic node-update $i add
properties/capabilities='profile:compute,boot_option:local'
> done
```

```
[stack@osp7-director ~]$ for i in 179befe6-2510-4311-ad9f-488045fdaff \
> ff0dadfe-e2f3-408f-b69d-01398bb9699d b59f57e3-d5e1-499a-80c1-aac0c78c9534
> do
> ironic node-update $i add
properties/capabilities='profile:CephStorage,boot_option:local'
> done
```

The added profiles can be queried for validation:



Make sure that the previously created Flavors match the output of the query below.

```
[stack@osp7-director ~]$ instack-ironic-deployment --show-profile
Preparing for deployment...
Querying assigned profiles ...
b7dde876-354a-4688-8550-aec8f64c582c
  "profile:control,boot_option:local"
e4563ca5-2f12-4e08-9905-f770f740ad2b
  "profile:control,boot_option:local"
285965a9-9713-4301-8ad5-7aa3ef5dd1c2
  "profile:control,boot_option:local"
b4dc04ac-0c69-4000-9c4d-2d82d141905f
  "profile:compute,boot_option:local"
036cae70-bdee-427c-987c-a6a2d8a32292
  "profile:compute,boot_option:local"
8570c96e-f9cd-44ff-a1d8-0252bc405c24
  "profile:compute,boot_option:local"
af46cd81-c78e-47c5-94e3-44d9d669410c
  "profile:compute,boot_option:local"
19260dbb-29a9-4810-b39d-85cc6e1d886f
  "profile:compute,boot_option:local"
```

```

d4dae332-4595-43be-9b63-5a64331ea33b
"profile:compute,boot_option:local"
179befe6-2510-4311-ad9f-4880454fdaff
"profile:CephStorage,boot_option:local"
ff0dadfe-e2f3-408f-b69d-01398bb9699d
"profile:CephStorage,boot_option:local"
b59f57e3-d5e1-499a-80c1-aac0c78c9534
"profile:CephStorage,boot_option:local"
DONE.
Prepared.

```

You can validate the ipmi, mac\_address and server profiles as shown below:

```

for i in $(ironic node-list | awk '/None/ {print $2}' );
do
ipmi_addr=`ironic node-show $i | grep "10.22" | awk '{print $7}'`
mac_addr=`ironic node-port-list $i | awk '/00:25/ {print $4}'`
profile=`ironic node-show $i | grep -io "u'profile:.*:local"`
echo $i $ipmi_addr $mac_addr $profile
done

```

## Overcloud Setup

Before delving into the Overcloud installation, it is necessary to understand and change the templates for your configuration. Red Hat Linux OpenStack Director provides lot of flexibility in configuring Overcloud. At the same time, understanding the parameters and providing the right inputs to heat through these templates is paramount.

### Customize Heat Templates

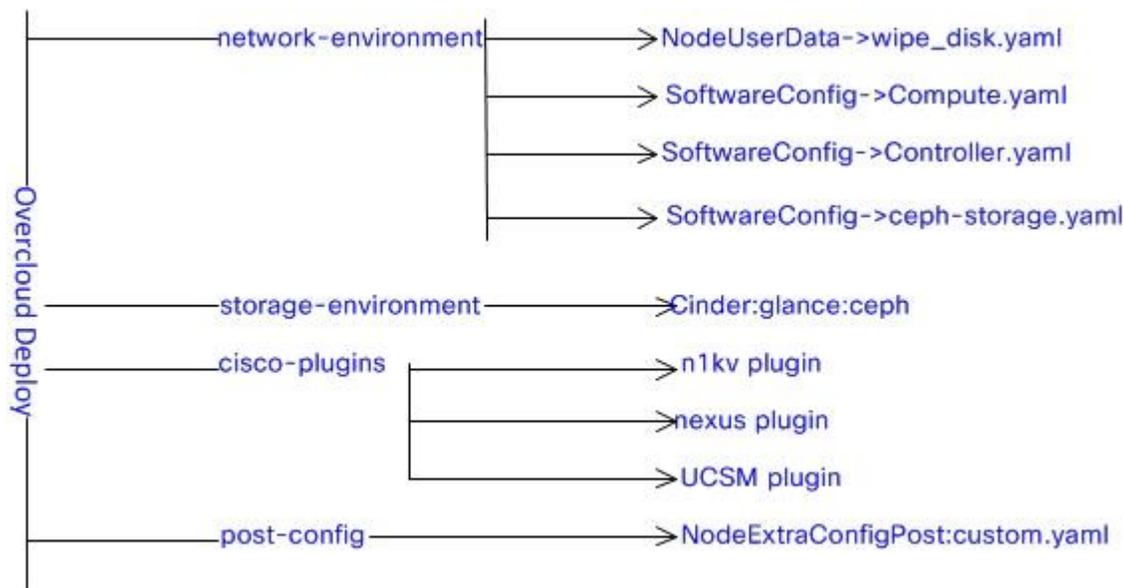
Before attempting the Overcloud install, it is necessary to understand and setup the Overcloud heat templates. For complete details of the templates, please refer to the Red Hat online documentation on OpenStack.

Overcloud is installed through command line interface with the following command. A top down approach of the yaml and configuration files is provided here.



The files are sensitive to whitespaces and tabs.

Refer to the Appendix A for run.sh, the command used to deploy Overcloud.



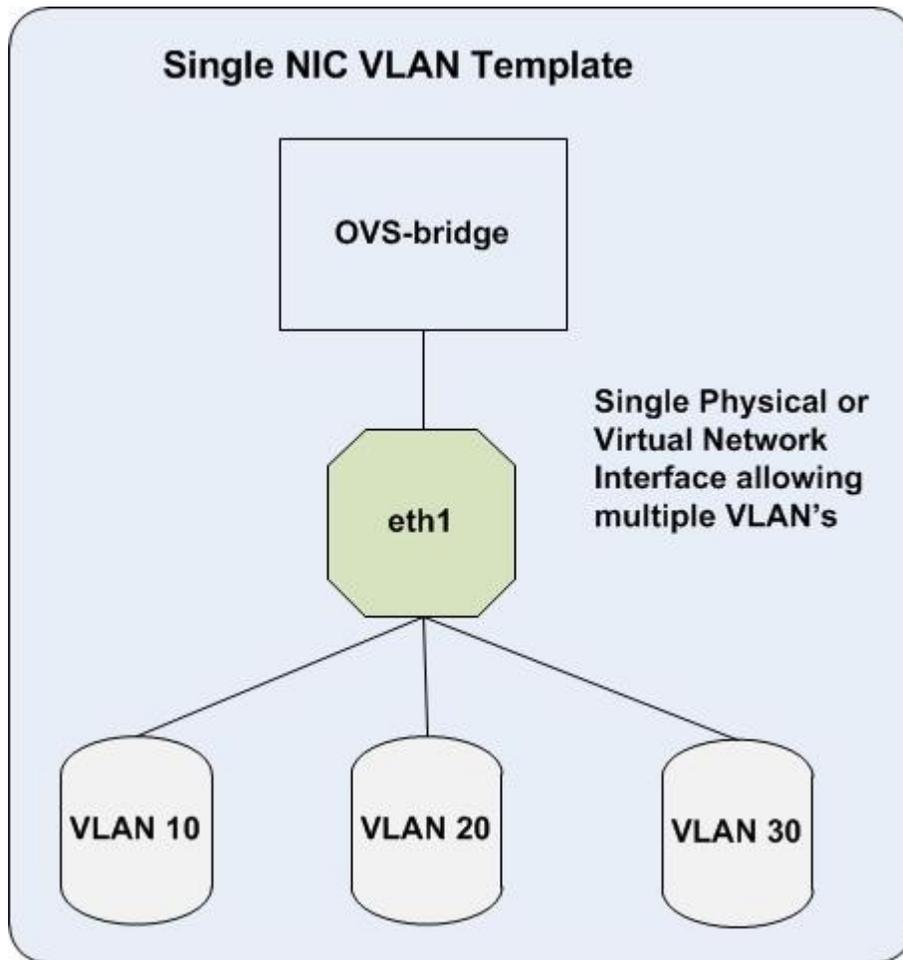
The heat templates have to be customized depending on the network layout and nic interface configurations in the setup. The templates are standard heat templates in YAML format. They are included in Appendix A. A set of configuration files with floating IP are included in Appendix A while the set in Appendix B are the files without floating IP configuration. Appendix A is the superset while Appendix B has the files that differ. Hence **in your configuration, if you have luxury of external IP's for VM's external access and you wish not to use floating IP's you may pick up all the files from Appendix A and overlay them with Appendix B configurations.** Again use them for reference purpose only and make the updates as needed.

The network configuration included in the Director are of two categories and are included in /usr/share/openstack-tripleo-heat-templates/network/config

Single nic VLAN's

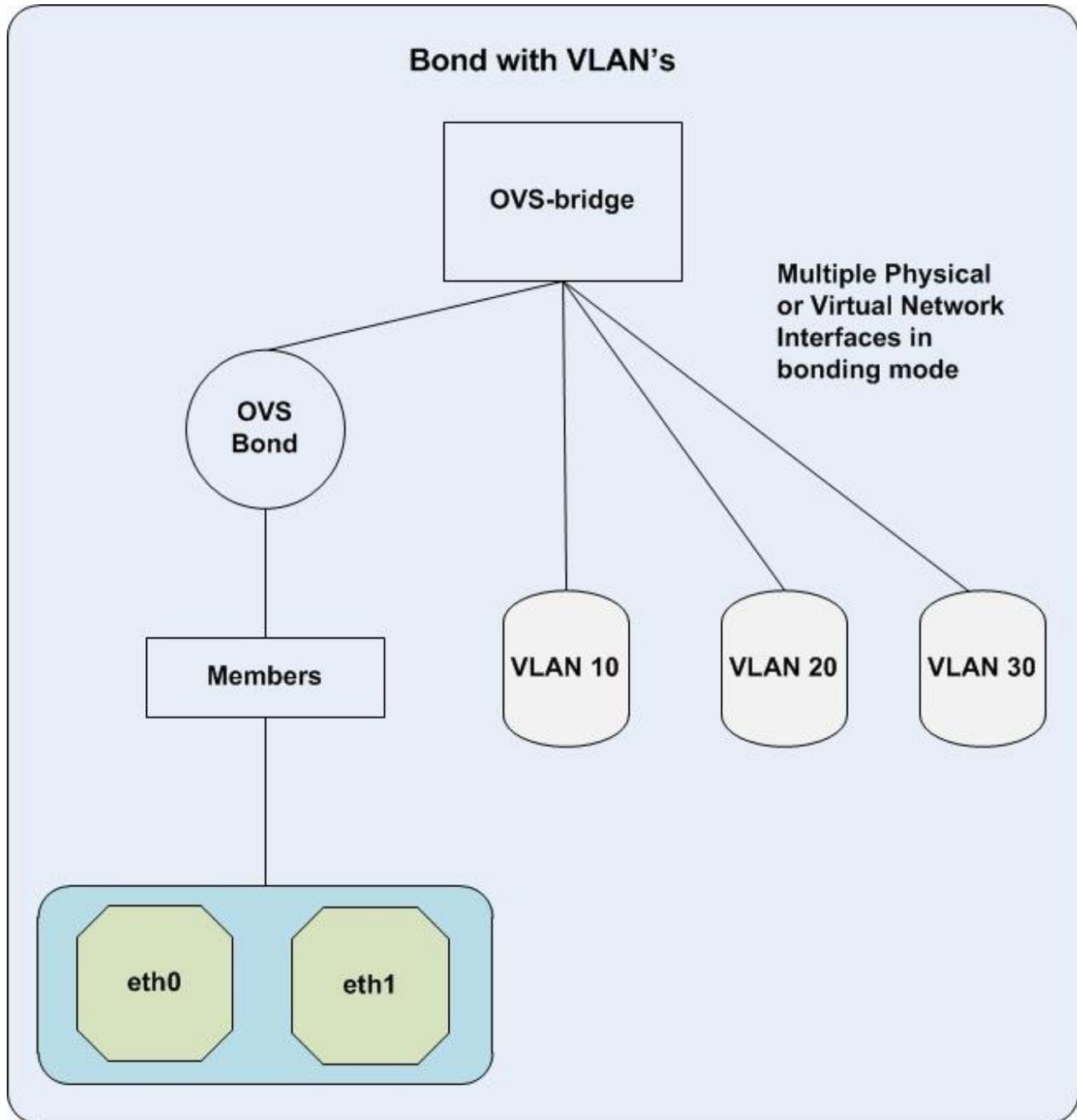
Bond with VLAN's

### Single NIC VLAN Templates



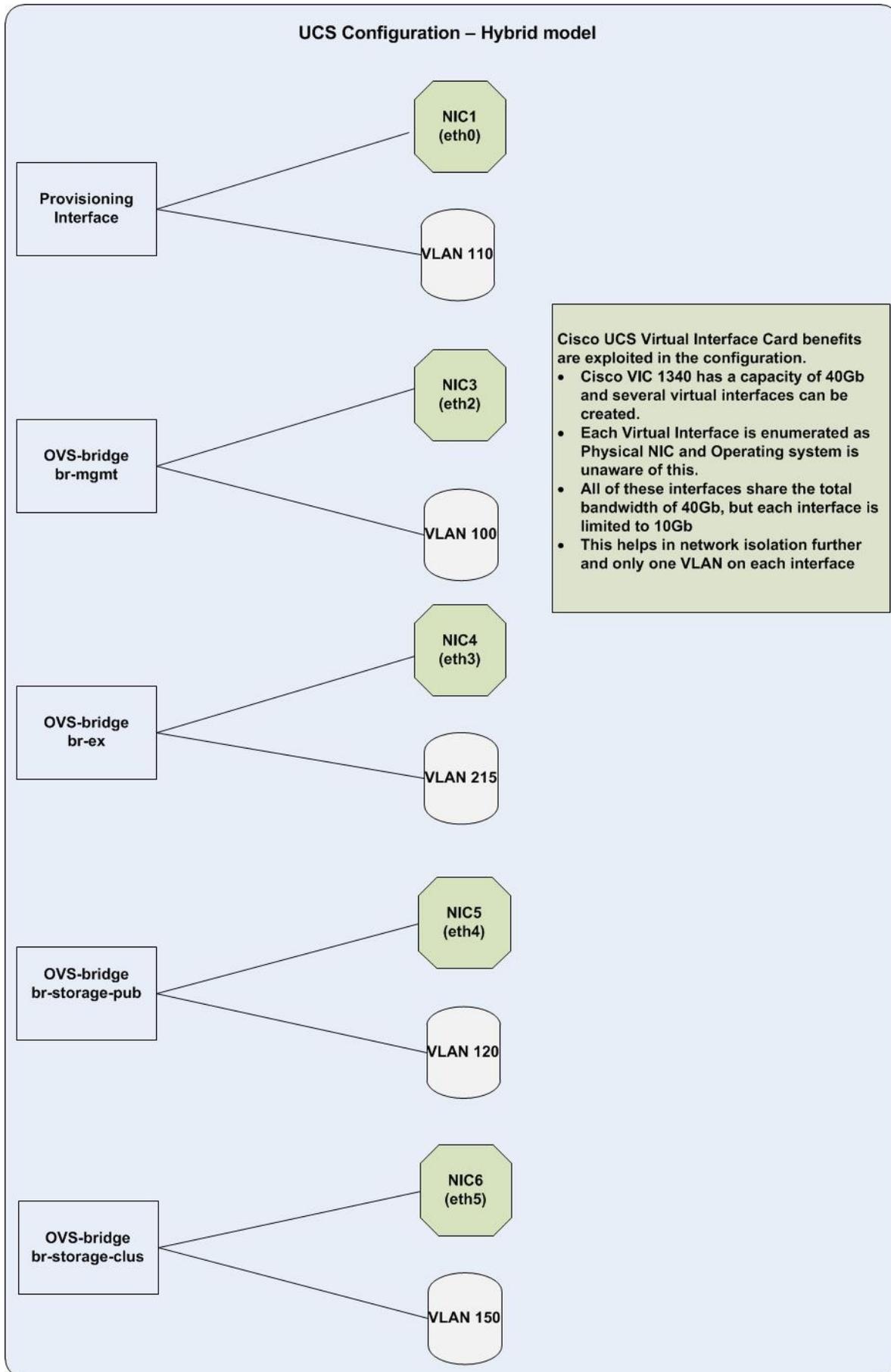
This model assumes that you have a single interface allowing all the VLAN's configured in the system.

## Bond with VLAN Templates



## Cisco UCS Configuration

In the Cisco UCS configuration a hybrid model was adopted. This was done for simplicity and also to have a separate VLAN dedicated on each interface for every network. While this gives a fine grain control of policies like QOS etc, if needed, but were not adopted for simplicity. NIC2 or eth1 was used as tenant interface.



```
As stack user mkdir -p /home/stack/templates/nic-configs
```

Copy the template files from /usr/share/openstack-tripleo-heat-templates. Refer Red Hat online documentation.

Create network-environment.yaml per above documentation or use the Appendix A for reference.

```
[stack@osp7-director templates]$ ls
ceph.yaml  cisco-plugins.yaml  nameserver_ntp.yaml  network-environment.yaml
post_config.yaml  storage-environment.yaml  wipe_disk.yaml

[stack@osp7-director nic-configs]$ ls *.yaml
ceph-storage.yaml  compute.yaml  controller.yaml
```

Some of the above files may have to be created. These files are referenced in Overcloud deploy command either directly or through another file. Ceph.yaml has to be modified directly in /usr/share/openstack-tripleo-heat-templates.

## Yaml Configuration Files Overview

### network-environment.yaml

The first section is for resource\_registry. The section for parameter defaults have to be customized. The following are a few important points to be noted in network-environment.yaml file:

1. Enter the Network Cidr values in the parameter section.
2. The Internal Network and the UCS management network are on the same network. Make sure that InternalApiAllocationPools do not overlap with UCS IP pools. In the configuration they span the subnet from 10.22.100.50 to 250.
3. For consistency a similar approach followed for Storage and Storage Management Allocation pools.
4. The Tenant Allocation pool and Network is created for /12 subnet, just to allocate more addresses.
5. The Control Plane Default Route is the Gateway Router for the provisioning network or the Undercloud IP. This matches with your network\_gateway and masquerade\_network in your undercloud.conf file.
6. EC2Metadata IP is the Undercloud IP again.
7. Neturon External Network Bridge should be set to "''". An empty string to allow multiple external **networks or VLAN's**.
8. No bonding used in the configuration. This will be addressed in our future releases.

### controller.yaml

This parameter section overrides the ones mentioned in the networking-environment file. The get\_param calls for the defined parameters. The following are important points to be considered for Controller.yaml file:

1. **The PXE interface NIC1 should have dhcp as false to configure static ip's, with next hop going to Undercloud node.**

2. The external bridge is configured to the External Interface Default Route on the External Network VlanID.
3. The MTU value of 9000 to be added as needed. Both the storage networks are configured on mtu 9000.

#### compute.yaml

The same rules for the Controller apply:

1. The PXE interface NIC1 is configured with dhcp as **false**. **There are no external IP's available for Controller and Storage**. Hence natting is done through UnderCloud node. For this purpose, the Control Plane Default Route is the, network gateway defined in undercloud.conf file which is also the UnderCloud local\_ip.
2. Only the Storage Public network is defined along with Tenant networks on Compute nodes.

#### ceph-storage.yaml

1. Same as Compute.yaml mentioned above.
2. Only Storage Public and Storage Cluster are defined in this file.

#### ceph.yaml

Configuring Ceph.yaml is tricky and needs to be done carefully. This is because we are configuring the partitions even before installing operating system on it. Also depending on the configuration whether you are using C240M4 LFF or C240M4 SFF the configuration changes.

An overview of the current limitations from the Red Hat Director and Cisco UCS and the workarounds is provided for reference.

The way disk ordering is done is inconsistent. However for ceph to work we need a consistent way of disk ordering. Post boot we can setup the disk labels by by-uuid or by-partuuid. However, in RHEL-OSP Director these have to be done before. [Bug 1253959](#) is being tracked for this issue and is supposed to be fixed in later versions.

This is also a **challenge to use JBOD's in Ceph**, the conventional way. Using **RAID-0 Luns in place of JBOD's is equally challenging. The Lun ID's have to be consistent every time a server reboots**. The order that is deployed in UCS is also unpredictable. Hence following workarounds are evolved on the configuration to meet these requirements. The internal SSD drives in both C240 LFF and SFF models will not be used as they are not visible to the RAID controller in the current version of UCSM and will pose challenges to RHEL-OSP Director (they are visible to BIOS, Luns cannot be carved out as RAID controller does not see them and they **appear as JBOD's to the kernel thus breaking the LUN and JBOD id's**).

Figure 13 Cisco UCS C240 M4 – Large Form Factor with 12 Slots

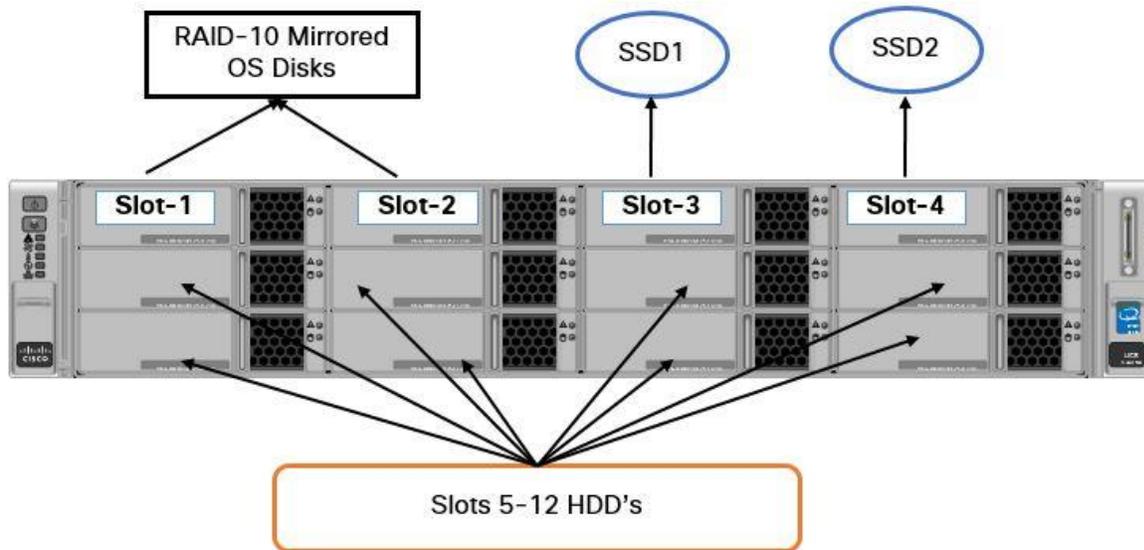
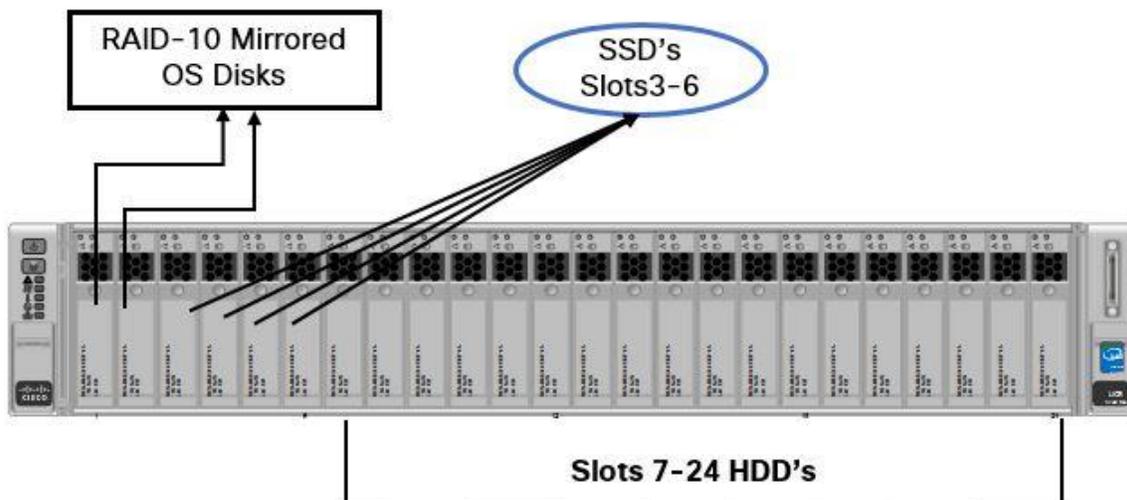


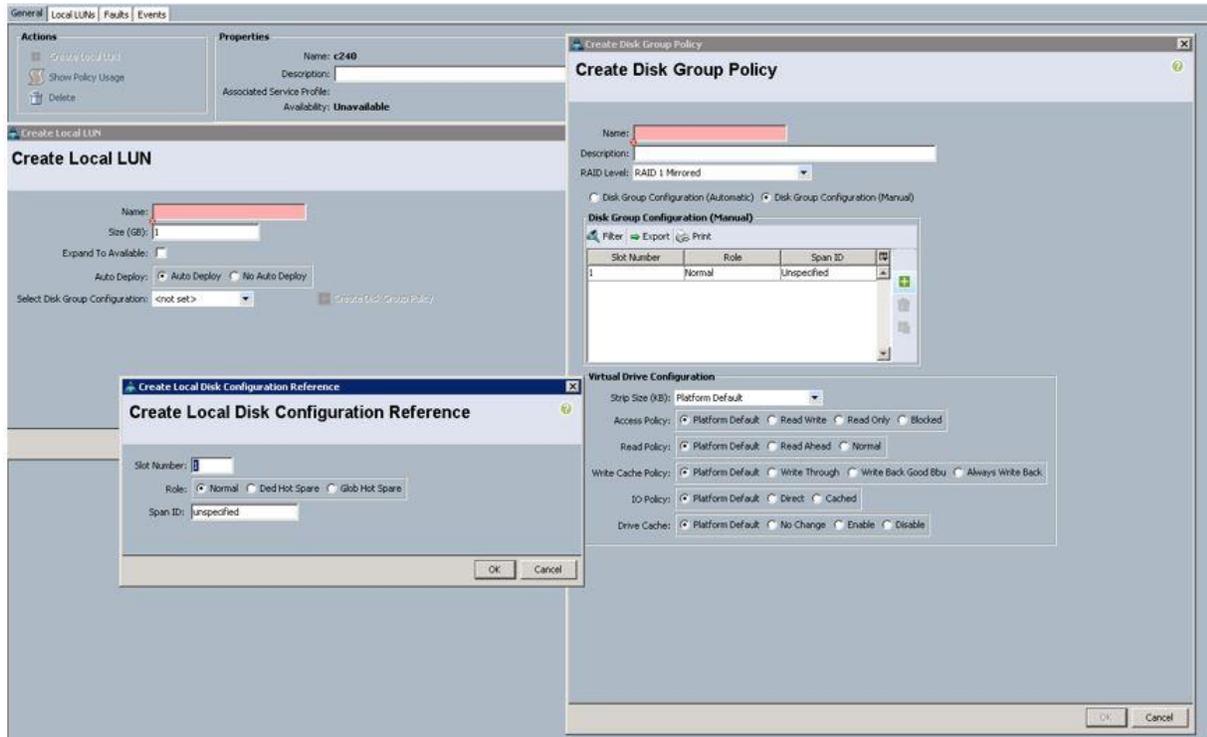
Figure 14 Cisco UCS C240 M4 – Small Form Factor with 24 Slots



### Cisco UCS Side Fixes to Mitigate the Issue

As mentioned earlier, storage profiles will be used from UCS side on these servers:

1. Make sure that you do not have local disk configuration policy in UCS for these servers.
2. Create storage profile, disk group policy as below under the template. There will be one Disk group policy for each slot. One policy for RAID-10 for the OS luns and one policy each of RAID-0 for the remaining.
3. Navigate to Create Storage Profile -> Create Local Lun -> Create Disk Group Policy (Manual) -> Create Local disk configuration. This will help in binding the disk slot to each lun created.



4. Create first the boot LUN from the first 2 slots and then apply. This will give LUN-0 to boot luns.
5. Create the second and third LUNs from the SSD slots (as in C240M4 LFF ). This would create RAID-0 luns, LUN-1 and LUN-2 on the SSD disks.
6. The rest of the LUNs can be created and applied in any order.
7. With the above procedure, we are assured that LUN-0 is for Operating system, LUN-1 and LUN-2 for SSD's and the rest for HDD's. This in turn decodes to /dev/sda for boot lun, /dev/sdb for SSD1 and /dev/sdc for SSD2 and the rest for HDD's.



Do not apply all the luns at the same time in the storage profile. First apply the boot lun, which should become LUN-0, followed by the SSD luns and then the rest of the HDD luns. Failure to comply with the above, will cause lun assignment in random order and heat will deploy on whatever the first boot lun presented to it.

Follow a similar procedure for C240 SFF servers too. A minimum of 4 SSD journals recommended for C240M4 SFF. The first two SSD luns with 5 partitions and the rest two with 4 partitions each.

#### OpenStack Side Fixes to Mitigate the Issue

Implementing Red Hat Linux OpenStack Director to successfully deploy Ceph on these disks need gpt label pre-created. This can be achieved by including wipe\_disk.yaml file which creates these labels with sgdisk utility. Please refer to Appendix A for details about -disk.yaml.



In the current version there is only one ceph.yaml file on all the servers. This mapping has to be uniform across the storage servers.

While the contents of `ceph.yaml` in the Appendix A are self-explanatory, the following is how the mappings **between SSD's and HDD's need to be done**:

```
ceph::profile::params::osds:
  '/dev/sdd':
    journal: '/dev/sdb1'
  '/dev/sde':
    journal: '/dev/sdb2'
  '/dev/sdf':
    journal: '/dev/sdb3'
  '/dev/sdg':
    journal: '/dev/sdb4'
  '/dev/sdh':
    journal: '/dev/sdc1'
  '/dev/sdi':
    journal: '/dev/sdc2'
  '/dev/sdj':
    journal: '/dev/sdc3'
  '/dev/sdk':
    journal: '/dev/sdc4'
```

The above is an example for C240M4 LFF server. Based on the LUN **id's created above /dev/sdb and /dev/sdc** are journal entries. Four entries for each of these journal directs RHEL-OSP to create 4 partitions on each SSD disk. The entries on the left are for HDD disks.

A similar approach can be followed for SFF servers.



Th `ceph.yaml` was copied to `/usr/share/openstack-tripleo-heat-templates/puppet/hieradata/`

---

#### [cisco-plugins.yaml](#)

The parameters section specifies the parameters.

#### [n1kv](#)

N1000vVSMIP: The Virtual Supervisor Module IP. This should be an address on the internal API network, outside of the assigned DHCP range to prevent conflicting ips.

N1000vPacemakerControl: True in HA configuration

N1000vVSMPassword: 'Password' - The password for N1KV

N1000vVSMHostMgmtIntf: br-mgmt

N1000vVSMVersion: Leave specified as an empty string- “

N1000vVEMHostMgmtIntf:vlan100, the Internal API VLAN

N1000vUplinkProfile: '{eth1: system-uplink,}'. This should be the interface connected to the tenant network. The current version does not support bridges. Refer to the limitations of UCS Manager plugin in [http://docwiki.cisco.com/wiki/OpenStack/UCS\\_Mechanism\\_Driver\\_for\\_ML2\\_Plugin\\_Kilo](http://docwiki.cisco.com/wiki/OpenStack/UCS_Mechanism_Driver_for_ML2_Plugin_Kilo). As both plugins are in the setup, this has to remain as eth1. These will be revisited in our next release cycle.

#### [Cisco UCS Manager](#)

NetworkUCSMip: UCS Manager IP

NetworkUCSMHostList: Mapping between tenant mac address derived from UCS with Service profile name, comma separated. This list has to be built for all the compute and controller nodes.

## Nexus

This will list both the Nexus switches details, their ip's and passwords.

Servers: The list should specify the interface MAC of each controller and compute and the port-channel numbers created on the Nexus switch.

NetworkNexusManagedPhysicalNetwork physnet-tenant, the parameter you pass in the Overcloud deploy command

**NetworkNexusVlanNamePrefix: 'q-' These are the vlans's that will be** created on the switches

NetworkNexusVxlanGlobalConfig: false. Vxlan is not used and is not validated as part of this CVD

NeutronServicePlugins: Leave the default string as is. Any **typo's may create successfully Overcloud but will fail to create VM's later.**

NeutronTypeDrivers: vlan. The only drivers validated in this CVD.

**NeutronL3HA: 'false' The current n1kv version does not support L3 HA. This will be revisited in the next revision.**

NeutronNetworkVLANRanges: 'physnet-tenant:250:749' The range you are passing to Overcloud deploy.

The controllerExtraConfig parameters are tunables. These workers reside in /etc/neutron.conf file. Only the parameters mentioned in Appendix A are validated.

## pre and post config.yaml

Overcloud deployment can be customized with pre and post commands. Please refer Red Hat OpenStack Director documentation for detailed information.

For pre-configuration, the following are supported

```
OS::TripleO::ControllerExtraConfigPre
OS::TripleO::ComputeExtraConfigPre
OS::TripleO::CephStorageextraConfigPre
OS::TripleO::NodeExtraConfigPre
```

wipe\_disks.yaml is configured as part of firstboot to create gpt lables on Storage node disks.

Customizing post-Configuration, is done through OS::TripleO::NodeExtraConfigPost. These can be applied as additional configurations. The current nameserver\_ntp.yaml file used in the configuration achieves these by using Heat SoftwareConfig types.

In Appendix B, **the yaml files used in the second pod, without floating IP's is included. Only the files that were different like network-environment.yaml etc are included.**

## Pre-Installation Checks Prior to Deploying Overcloud

To perform the pre-installation checks, complete the following steps:

1. Download the network-environment-validator.py from github;  
<https://github.com/rthallisey/clapper/blob/master/network-environment-validator.py>.

- a. Download python-ipaddress from the web and install it (`rpm -ivh <rpmname>`), the one used on the configuration was `python-ipaddress-1.0.7-4.el7.noarch.rpm`.
- b. Validate the yaml files as shown below:

```
cd /home/stack/templates as stack user
python network-environment-validator.py -n network-environment.yaml
DEBUG: __main__:
parameter_defaults:
  ControlPlaneDefaultRoute: 10.22.110.26
  ControlPlaneSubnetCidr: '24'
  DnsServers: [8.8.8.8, 8.8.4.4]
.....
.....
-----SUMMARY-----
SUCCESSFUL Validation with 0 error(s)
[stack@osp7-director templates]$
```



If you receive any errors, stop here and fix the issue(s).

2. Run `ironic node-list` to check that all the servers are available, powered off and not in maintenance.

```
[stack@osp7-director ~]$ ironic node-list
```

UUID	Name	Instance UUID	Power State	Provision State	Maintenance
b7dde876-354a-4688-8550-aec8f64c582c	None	None	power off	available	False
e4563ca5-2f12-4e08-9905-f770f740ad2b	None	None	power off	available	False
285965a9-9713-4301-8ad5-7aa3ef5dd1c2	None	None	power off	available	False
b4dc04ac-0c69-4000-9c4d-2d82d141905f	None	None	power off	available	False
036cae70-bdee-427c-987c-a6a2d8a32292	None	None	power off	available	False
8570c96e-f9cd-44ff-a1d8-0252bc405c24	None	None	power off	available	False
af46cd81-c78e-47c5-94e3-44d9d669410c	None	None	power off	available	False
19260dbb-29a9-4810-b39d-85cc6e1d886f	None	None	power off	available	False
d4dae332-4595-43be-9b63-5a64331ea33b	None	None	power off	available	False
179befe6-2510-4311-ad9f-4880454fdaff	None	None	power off	available	False
ff0dadfe-e2f3-408f-b69d-01398bb9699d	None	None	power off	available	False
b59f57e3-d5e1-499a-80c1-aac0c78c9534	None	None	power off	available	False

While understanding the reason why a server is not as listed above, you may use ironic APIs to change the state if they are not in the desired state:

After sourcing `stackrc` file;

```
ironic node-set-power-state <uuid> off
ironic node-set-provision-state <uuid> provide
ironic node-set-maintenance <uuid> false
```

In case of larger deployments, the default values of `max resource per stack` may not be sufficient.

3. Maximum resources allowed per top-level stack. (integer value)

```
#max_resources_per_stack = 1000
```

Update the value to a higher number in `/etc/heat/heat.conf`. In a pod with 35 nodes, we had to bump up this value to 10000 and restart heat engine. However in a pod with 3 controllers, 6 computes and 3 ceph nodes this wasn't necessary.

```
systemctl restart openstack-heat-engine.service
```

4. Update Ceph timeout values. This issue was noticed on the configuration in particular for larger deployments. This is per updates in bug 1250654.

```
cd /usr/share/openstack-tripleo-heat-templates/puppet/manifests
```

5. Backup the files:

```
[root@osp7-director manifests]# ls *.old
overcloud_cephstorage.pp.old  overcloud_controller_pacemaker.pp.old
overcloud_controller.pp.old
```

```
edit overcloud_cephstorage.pp
```

```
[root@osp7-director manifests]# diff overcloud_cephstorage.pp
overcloud_cephstorage.pp.old
24,27d23
< Exec {
< timeout => 9000,
< }
<
This is just before the line
if str2bool(hiera('ceph_osd_selinux_permissive', true)) {
```

```
edit overcloud_controller.pp
```

```
[root@osp7-director manifests]# diff overcloud_controller.pp
overcloud_controller.pp.old
35,38d34
< Exec {
< timeout => 9000,
< }
<
This is just before the line
if hiera('step') >= 2 {
```

```
edit overcloud_controller_pacemaker.pp
```

```
[root@osp7-director manifests]# diff overcloud_controller_pacemaker.pp
overcloud_controller_pacemaker.pp.old
38,41d37
< Exec {
< timeout => 9000,
< }
<
This is just before the line
if hiera('step') >= 1 {
```

6. Reboot the Undercloud node.

## Deploying Overcloud

With the templates in place, Overcloud deploy can run the command mentioned in Appendix A. OpenStack help Overcloud deploy will show all the arguments that can be passed to the deployment command.

A snippet is provided below:

```
#!/bin/bash
openstack overcloud deploy --templates \
-e /usr/share/openstack-tripleo-heat-templates/overcloud-resource-registry-
puppet.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-
isolation.yaml \
```

```

-e /home/stack/templates/network-environment.yaml \
-e /home/stack/templates/storage-environment.yaml \
-e /home/stack/templates/cisco-plugins.yaml \
-e /home/stack/templates/post_config.yaml \
--control-flavor control --compute-flavor compute --ceph-storage-flavor CephStorage \
--compute-scale 6 --control-scale 3 --ceph-storage-scale 3 \
.....

```

The following are a few that need to be noted:

```

--control-flavor control --compute-flavor compute --ceph-storage-flavor
CephStorage

```

The following are the flavors created earlier and associated with the nodes. Check Create and Set Flavors above:

ntp server is the server name to be used in the overcloud /etc/ntp.conf file  
neutron-network-type and neutron-tunnel-type are vlan.  
neutron-network-vlan-ranges physnet-tenant:250:749,floating:160:160. Here vlan ranges from 250 to 749 are reserved for tenants, while vlan 160 is for floating ip network.  
rhel-reg, reg-method <portal|Satellite> --reg-org <org id> --reg-activation-key <activation\_key>.  
These help to register the servers with Red Hat Network. Check the limitations with [registration here](#).  
Verbose, debug and log files are self-explanatory.

After successful deployment, the deploy command should show you the following:

```

DEBUG: os_cloud_config.utils.clients Creating nova client.
Overcloud Endpoint: http://172.22.215.91:5000/v2.0/
Overcloud Deployed
DEBUG: openstackclient.shell clean_up DeployOvercloud

```



Write down the endpoint URL to launch the dashboard later. This completes Overcloud deployment.

---

## Debugging Overcloud Failures

Overcloud deployment may fail for several reasons. Either because of a human error, for example, passing incorrect parameters or erroneous yaml configuration files or timeouts or bug. It is beyond the scope of this document to cover all of the possible failures. However, a few scenarios that were encountered on the configuration with explanations are provided in the [Troubleshooting](#) section of this document.

## Overcloud Post Deployment Process

To perform the post deployment process, complete the following steps:

1. Run nova list and login as heat-admin to each host:

```
[stack@osp7-director ~]$ nova list
```

ID	Name	Status	Task State	Power State	Networks
948b754f-c992-4211-940a-2308bcff31a6	overcloud-cephstorage-0	ACTIVE	-	Running	ct plane=10.22.110.78
790ff133-bc91-4877-8f1f-200417435e08	overcloud-cephstorage-1	ACTIVE	-	Running	ct plane=10.22.110.79
97f6f1ea-0ba2-4f3d-b2b3-f068d17d2509	overcloud-cephstorage-2	ACTIVE	-	Running	ct plane=10.22.110.52
fc0e7fd5-a3e5-4f19-9cf5-80311fd2efd0	overcloud-compute-0	ACTIVE	-	Running	ct plane=10.22.110.61
aa42c6c7-7222-40f9-9242-61bd59e45760	overcloud-compute-1	ACTIVE	-	Running	ct plane=10.22.110.53
a323d1d4-6678-4a74-8b7e-007a4fad4e5b	overcloud-compute-2	ACTIVE	-	Running	ct plane=10.22.110.56
1fbc8a0d-b739-43f4-8360-6d1ccd8f0d8e	overcloud-compute-3	ACTIVE	-	Running	ct plane=10.22.110.60
5d9db6e3-0ac5-4329-8734-3ec7069847f8	overcloud-compute-4	ACTIVE	-	Running	ct plane=10.22.110.58
6454cd25-fcd2-44d4-9296-2ad3f54415bf	overcloud-compute-5	ACTIVE	-	Running	ct plane=10.22.110.54
80dd5a71-4e43-4bc4-8b18-35e78835c67f	overcloud-controller-0	ACTIVE	-	Running	ct plane=10.22.110.59
938b1742-5a24-42ff-8268-4e397ca87232	overcloud-controller-1	ACTIVE	-	Running	ct plane=10.22.110.55
e741cd25-9abf-4fee-a8bd-b7fe87695ece	overcloud-controller-2	ACTIVE	-	Running	ct plane=10.22.110.57

```
[stack@osp7-director ~]$
```

```
for i in $(nova list | awk '/ACTIVE/ {print $12}'|cut -d "=" -f2 );
```

```
do
```

```
ssh -l heat-admin -o StrictHostKeyChecking=no $i "touch /tmp/abc; ls -l /tmp/abc"
```

```
done
```



A command like the one listed above will validate that all the servers are up and running.

2. Check that the servers are registered with Red Hat Network.

subscription-manager status should reveal the status of this registration. [Bug 1299795](#) reports this issue. Please follow the workaround to register the servers in case they are not.

3. Check ntp and dns entries:

```
/etc/ntp.conf and /etc/resolv.conf for appropriate entries.
```

4. Issues with Ceph:

- a. OSD Journal Size.

[Bug 1297251](#) pauses a successful deployment of Ceph. The `osd_journal_size` is ignored from the parameter file and Ceph **osd's will be reported down**.

With the changes made to `wipe_disk` yaml, we are creating 20GB partitions for journals. However the `ceph.conf` file needs to be updated as follows on all the controllers and storage nodes

```
osd_journal_size=20000
```

Update `ceph.conf` on all the controllers and Ceph storage nodes

Restart monitors on all controller nodes

```
/etc/init.d/ceph restart mon
```

Activate the `ceph-disks` on storage nodes.

```
for i in $(nova list | grep "cephstorage" | awk '/ACTIVE/ {print $12}'|cut -d "=" -f2 )
```

```
do
```

```
ssh -l heat-admin $i 'sudo ceph-disk activate-all' ;
```

```
done
```

b. Default pg num value.

Change the default pg num values for Ceph.

The current RHEL-OSP Director supports only `pg_num=128`, the default placement groups. [Bug 1283721](#) discusses this limitation. This default value may have to be updated depending on the **number of OSD's in the cluster**.

Number of placement groups = ( OSD's x 100 ) / Replication factor (3)

<http://docs.ceph.com/docs/master/rados/operations/placement-groups/>

As per the above formula for 24 OSD's it is 2400/3 or 800 PG's for the cluster. Considering this has to be to the power of 2, we will create 1024 PG's in the cluster. However RHEL-OSP creates 4 pools by default. This means 256 PG's for each pool.

In case you are using C240M4S, PG's have to be calculated for 54 OSD's.

```
[root@overcloud-cephstorage-2 ~]# ceph osd lspools
4 rbd,5 images,6 volumes,7 vms,
```

The pools will be recreated with 256 PG's in each as shown below:

Set the placement groups as shown below;

```
for i in rbd images volumes vms; do
ceph osd pool set $i pg_num 256;
sleep 20
ceph osd pool set $i pgp_num 256;
sleep 10
done
```

5. Query the pools and tree.

```
[root@overcloud-cephstorage-2 ~]# ceph df
GLOBAL:
  SIZE      AVAIL      RAW USED    %RAW USED
  128T      128T      872M       0
POOLS:
  NAME      ID        USED      %USED      MAX AVAIL    OBJECTS
  rbd       4         0         0          43983G      0
  images    5         0         0          43983G      0
  volumes   6         0         0          43983G      0
  vms       7         0         0          43983G      0
[root@overcloud-cephstorage-2 ~]# ceph -s
cluster cf81c0fc-be4b-11e5-9b8a-0025b522225f
health HEALTH OK
monmap e2: 3 mons at {overcloud-controller-0=10.22.120.53:6789/0,overcloud-controller-1=10.22.120.54:6789/0,overcloud-controller-2=10.22.120.51:6789/0}
election epoch 6, quorum 0,1,2 overcloud-controller-2,overcloud-controller-0,overcloud-controller-1
osdmap e95: 24 osds: 24 up, 24 in
pgmap v129: 1024 pgs, 4 pools, 0 bytes data, 0 objects
      872 MB used, 128 TB / 128 TB avail
      1024 active+clean

[root@overcloud-cephstorage-2 ~]# ceph osd tree
ID WEIGHT  TYPE NAME                UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 128.87988 root default
-2 42.95996 host overcloud-cephstorage-1
  0 5.37000 osd.0                  up 1.00000 1.00000
  3 5.37000 osd.3                  up 1.00000 1.00000
  6 5.37000 osd.6                  up 1.00000 1.00000
  9 5.37000 osd.9                  up 1.00000 1.00000
 12 5.37000 osd.12                 up 1.00000 1.00000
 15 5.37000 osd.15                 up 1.00000 1.00000
 18 5.37000 osd.18                 up 1.00000 1.00000
 21 5.37000 osd.21                 up 1.00000 1.00000
-3 42.95996 host overcloud-cephstorage-2
  1 5.37000 osd.1                  up 1.00000 1.00000
  4 5.37000 osd.4                  up 1.00000 1.00000
  7 5.37000 osd.7                  up 1.00000 1.00000
 10 5.37000 osd.10                 up 1.00000 1.00000
 13 5.37000 osd.13                 up 1.00000 1.00000
 16 5.37000 osd.16                 up 1.00000 1.00000
 19 5.37000 osd.19                 up 1.00000 1.00000
 22 5.37000 osd.22                 up 1.00000 1.00000
-4 42.95996 host overcloud-cephstorage-0
  2 5.37000 osd.2                  up 1.00000 1.00000
  5 5.37000 osd.5                  up 1.00000 1.00000
  8 5.37000 osd.8                  up 1.00000 1.00000
 11 5.37000 osd.11                 up 1.00000 1.00000
 14 5.37000 osd.14                 up 1.00000 1.00000
 17 5.37000 osd.17                 up 1.00000 1.00000
 20 5.37000 osd.20                 up 1.00000 1.00000
 23 5.37000 osd.23                 up 1.00000 1.00000
```

## 6. Sporadic issues on N1000V

The following race condition is observed occasionally in the installs. Please verify that you are not hitting this issue before moving ahead. Neutron will fail to start and pcs status may be erring out if this condition exists. Please verify the following and correct as indicated if you encounter this issue.

First, check VSM for any duplicated default profiles by running the following command. If there are only two profiles and they have different descriptions, nothing needs to be done. If, as shown below (duplicate descriptions highlighted for clarity), there are three or more profiles and the descriptions repeat, then one of the duplicates needs to be removed.

```
vsm-p# show nsm network segment pool
nsm network segment pool 25c5f0b9-9ae7-45b7-b4a6-2e3a418a52af
  description default-vxlan-np
  uuid 25c5f0b9-9ae7-45b7-b4a6-2e3a418a52af
  member-of logical network 25c5f0b9-9ae7-45b7-b4a6-2e3a418a52af_log_net

nsm network segment pool 8df170c6-1bf5-4a42-bf2e-9b075b1b5537
  description default-vlan-np
  uuid 8df170c6-1bf5-4a42-bf2e-9b075b1b5537
  member-of logical network 8df170c6-1bf5-4a42-bf2e-9b075b1b5537_log_net

nsm network segment pool d792329d-6215-4a97-95fd-6dbac4cb220f
  description default-vxlan-np
  uuid d792329d-6215-4a97-95fd-6dbac4cb220f
  member-of logical network d792329d-6215-4a97-95fd-6dbac4cb220f_log_net
```

After confirming this is the issue, select one of the duplicate network profiles to be deleted. For this **example, we will use the first duplicate “default-vxlan-np” profile with uuid- 25c5f0b9-9ae7-45b7-b4a6-2e3a418a52af**. Replace this value as needed for your deployment in the commands below. Connect to mysql and schema ovs\_neutron neutron and verify that the three profiles are also seen in mysql:

```
MariaDB [ovs_neutron]> select * from cisco_ml2_n1kv_network_profiles;
```

id	name	segment_type	segment_range	multicast_ip_index	multicast_ip_range	sub_type	physical_network
25c5f0b9-9ae7-45b7-b4a6-2e3a418a52af	default-vxlan-np	vxlan	NULL	0	NULL	NULL	NULL
8df170c6-1bf5-4a42-bf2e-9b075b1b5537	default-vlan-np	vlan	NULL	0	NULL	NULL	NULL
d792329d-6215-4a97-95fd-6dbac4cb220f	default-vxlan-np	vxlan	NULL	0	NULL	NULL	NULL

```
3 rows in set (0.00 sec)
```

The above two queries return 3 rows.

There should be only one row for each vlan and vxlan entries in the table and the id from cisco\_ml2\_n1kv\_network\_profiles should match with id from the above command showing nsm network segment pools.

Delete the duplicate from VSM first

```
conf t
no nsm network segment pool 25c5f0b9-9ae7-45b7-b4a6-2e3a418a52af
copy running startup
```

Then, delete from mysql database

```
mysql -e "delete from ovs_neutron.cisco_ml2_n1kv_network_profiles where
id='25c5f0b9-9ae7-45b7-b4a6-2e3a418a52af'";
```

Finally, restart neutron services on all the 3 controller nodes

```
systemctl daemon-reload
systemctl restart neutron-server.service
```

## 7. Issue with Neutron Network VLAN ranges.

This issue is a regression from RHEL-OSP 7.1 Please refer bug [1297975](#). Only the first bridge makes its entry in `network_vlan_ranges` in `/etc/neutron/plugin.ini`. This needs to be as referred in overcloud deploy command:

```
--neutron-network-vlan-ranges physnet-tenant:250:749,floating:160:160
```

Update `/etc/neutron/plugin.ini` on all the three controller nodes, with the correct string

From

```
network_vlan_ranges =physnet-tenant:250:749
```

to

```
network_vlan_ranges =physnet-tenant:250:749,floating:160:160
```

Restart neutron network as:

```
systemctl daemon-reload
systemctl restart neutron-server.service
```

Also check the status of pcs resources as:

```
pcs resource cleanup
sleep 15
pcs status
pcs status | egrep -i "error|stop"
```

## Overcloud Post-Deployment Configuration

To perform the post-deployment configuration, complete the following steps:

### 1. Start `fence_cisco_ucs`.

Run `fence_cisco_ucs` and pass the UCSM IP and passwords to it. `Openstack_Controller_Node[1,2,3]` are the service profile names for the controllers. Replace the string accordingly.

```
[root@overcloud-controller-0 ~]# for i in 1 2 3
do
fence_cisco_ucs --ip=10.22.100.5 --username=admin --password=UCSManagerPassword \
--plug="Openstack_Controller_Node${i}" --missing-as-off --action=on --ssl-
insecure -z;
done
```

```
Success: Powered ON
```

```
Success: Powered ON
```

```
Success: Powered ON
```

### 2. Check the status:

```
[root@overcloud-controller-0 ~]# for i in 1 2 3
do
```

```
fence_cisco_ucs --ip=10.22.100.5 --username=admin --password= UCSManagerPassword
\  
--plug="Openstack_Controller_Node${i}" --missing-as-off -o status --ssl-insecure
-z;
done
Status: ON
Status: ON
Status: ON
```

### 3. Configuring PaceMaker.

Before proceeding with pacemaker configuration, it is necessary to understand the relationship between the service profile names in UCS with the node names dynamically created by OpenStack as part of Overcloud deployment.

- a. Either login through the Console or extract from /etc/neutron/plugin.ini.

Plugin.ini will be updated by Cisco Plugins that have this information. Open /etc/neutron/plugin.in file and go to the end of the file. Extract the controller syntax.

```
overcloud-controller-1.localdomain:Openstack_Controller_Node1,
overcloud-controller-0.localdomain:Openstack_Controller_Node2,
overcloud-controller-2.localdomain:Openstack_Controller_Node3
```

The mapping is controller-0 is mapped to Service Profile Controller\_Node2 and so on.

- b. Create a shell script as below with the following information and execute it

```
#!/bin/bash
# Note that ';' as a separator instead of ',' from plugin.ini
sudo pcs stonith create ucs-fence-controller fence_cisco_ucs \  
pcmk_host_map="overcloud-controller-1:Openstack_Controller_Node1;overcloud-
controller-0:Openstack_Controller_Node2;overcloud-controller-
2:Openstack_Controller_Node3" \  
ipaddr=10.22.100.5 login=admin passwd=<password> ssl=1 ssl_insecure=1 op
monitor interval=60s
sleep 3;
pcs stonith update ucs-fence-controller meta failure-timeout=300s
pcs property set cluster-recheck-interval=300s
```

4. Run the following:

```
[root@overcloud-controller-0 ~]# sudo pcs property set stonith-enabled=true
[root@overcloud-controller-0 ~]# pcs resource cleanup
Waiting for 1 replies from the CRMD. OK
```

- a. Validate the configuration after running the above commands:

```
sudo pcs stonith show ucs-fence-controller
sudo pcs property show
```

```
[root@overcloud-controller-0 ~]# pcs stonith show ucs-fence-controller
Resource: ucs-fence-controller (class=stonith type=fence_cisco_ucs)
Attributes: pcmk_host_map=overcloud-controller-1:Openstack_Controller_Node1;
overcloud-controller-0:Openstack_Controller_Node2;
overcloud-controller-2:Openstack_Controller_Node3
ipaddr=10.22.100.5 login=admin passwd=<password> ssl=1 ssl_insecure=1
Meta Attrs: failure-timeout=300s
Operations: monitor interval=60s (ucs-fence-controller-monitor-interval-60s)

[root@overcloud-controller-0 ~]# sudo pcs property show
Cluster Properties:
cluster-infrastructure: corosync
cluster-name: tripleo_cluster
cluster-recheck-interval: 300s
dc-version: 1.1.13-10.e17-44eb2dd
have-watchdog: false
maintenance-mode: false
redis_REPL_INFO: overcloud-controller-2
stonith-enabled: true
[root@overcloud-controller-0 ~]# corosync-quorumtool -s
Quorum information
-----
Date: Tue Jan 19 06:42:46 2016
Quorum provider: corosync_votequorum
Nodes: 3
Node ID: 1
Ring ID: 8
Quorate: Yes

votequorum information
-----
Expected votes: 3
Highest expected: 3
Total votes: 3
Quorum: 2
Flags: Quorate

membership information
-----
Nodeid Votes Name
    3      1 overcloud-controller-2
    1      1 overcloud-controller-0 (local)
    2      1 overcloud-controller-1
[root@overcloud-controller-0 ~]# corosync-cfgtool -s
Printing ring status.
Local node ID 1
RING ID 0
    id      = 10.22.100.54
    status  = ring 0 active with no faults
```

#### 5. Configure n1kv.

Run PCS status on any one of the controller nodes to determine the running primary and secondary VSM's:

```
[root@overcloud-controller-0 ~]# pcs status | grep -i vsm
vsm-s (ocf::heartbeat:VirtualDomain): Started overcloud-controller-0
vsm-p (ocf::heartbeat:VirtualDomain): Started overcloud-controller-1
```

Run `sudo ovs-vsctl show > /tmp/1` and look for the following tag on controller node. This should be the `vlan100`, the internal API network VLAN configured earlier in `cisco-plugins` for `n1kv`.

```
Port br-mgmt-vsm-br
    tag: 100
    Interface br-mgmt-vsm-br
```

```
type: patch
options: {peer=vsm-br-br-mgmt}
```

The primary VSM is running on controller-1. Login to controller-1

```
[stack@osp7-director ~]$ ssh -l heat-admin 10.22.110.55
```

```
Last login: Mon Jan 18 21:20:22 2016 from 10.22.110.26
```

```
[heat-admin@overcloud-controller-1 ~]$ sudo -i
```

```
[root@overcloud-controller-1 ~]# virsh list
```

Id	Name	State
2	vsm-p	running

```
[root@overcloud-controller-1 ~]# virsh console 2
```

```
Connected to domain vsm-p
```

```
Escape character is ^]
```

```
Nexus 1000v Switch
```

```
vsm-p login: admin
```

```
Password:  Enter password for VSM. The default password is 'Password'
```

```
.....
```

```
vsm-p# conf terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
vsm-p(config)# show module
```

Mod	Ports	Module-Type	Model	Status
1	0	Virtual Supervisor Module	Nexus1000V	active *
2	0	Virtual Supervisor Module	Nexus1000V	ha-standby



This shows that the second VSM is ha-standby mode. For any operations make sure that it is in ha-standby mode and not \*powered\* mode. Please wait few seconds to make sure that secondary is in ha-standby mode. Show module should also show all the controller and compute nodes.

```
vsm-p(config)# port-profile type vethernet default-pp
```

```
vsm-p(config-port-prof)# no shut
```

```
vsm-p(config-port-prof)# state enabled
```

```
vsm-p(config-port-prof)# publish port-profile
```

```
vsm-p(config-port-prof)# exit
```

```
vsm-p(config)# port-profile type ethernet system-uplink
```

```
vsm-p(config-port-prof)# no shut
```

```
vsm-p(config-port-prof)# state enabled
```

```
vsm-p(config-port-prof)# switchport mode trunk
```

```
vsm-p(config-port-prof)# end
```

```
vsm-p# show module
```

Mod	Ports	Module-Type	Model	Status
1	0	Virtual Supervisor Module	Nexus1000V	active *
2	0	Virtual Supervisor Module	Nexus1000V	ha-standby

```
.....
```

```
.....
```

```
vsm-p# copy running startup
```

```
.....
```

```
.....
```

```
vsm-p# reload
```

```
This command will reboot the system. (y/n)? [n] y
```

```
Broadcast message from root (ttyS0) (Tue Jan 19 07:14:13 2016):
```

```
The system is going down for reboot NOW!
```

Enter configuration commands, one per line. End with CNTL/Z.

```
2016 Jan 19 07:15:30 vsm-p vem_mgr[2368]: %VEM_MGR-2-MOD_ONLINE: Module 11 is
online
```

```
<Press Enter >
```

```
Nexus 1000v Switch
```

```
vsm-p login: admin
```

```
Password:
```

```
Cisco Nexus Operating System (NX-OS) Software
```

```
TAC support: http://www.cisco.com/tac
```

```
vsm-p# show module | grep "Supervisor"
```

```
1    0    Virtual Supervisor Module      Nexus1000V      active *
2    0    Virtual Supervisor Module      Nexus1000V      ha-standby
```

The system is ready to be used.



Enable Advanced licenses. Please check whether you have purchased the following advance features for n1kv. Enable when you have these licenses. The following document covers installation of advanced licenses on N1KV-

[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus1000/sw/5\\_2\\_1\\_s\\_v\\_3\\_1\\_1/licensing/config/b\\_Cisco\\_N1KV\\_Multi-Hypervisor\\_Licensing\\_Config.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus1000/sw/5_2_1_s_v_3_1_1/licensing/config/b_Cisco_N1KV_Multi-Hypervisor_Licensing_Config.html).

```
vsm-p# conf terminal
```

Enter configuration commands, one per line. End with CNTL/Z.

```
vsm-p(config)# svcs switch edition advanced
```

```
vsm-p(config)# feature l3forwarding
```

```
vsm-p(config)# 2016 Jan 19 07:18:36 vsm-p fwm[2387]: %FWM-2-
L3FWD_CHANGE_NOTIFICATION: Layer 3 Forwarding feature has been enabled. Reload
the VSM.
```

```
vsm-p(config)# copy running startup
```

```
.....
```

```
.....
```

```
vsm-p(config)# reload
```

This command will reboot the system. (y/n)? [n] y

```
2016 Jan 19 07:19:16 vsm-p vshd[4654]: %PLATFORM-2-PFM_SYSTEM_RESET: Manual
system restart from Command Line Interface
```

```
Broadcast message from root (ttyS0) (Tue Jan 19 07:19:17 2016):
```

```
The system is going down for reboot NOW!
```

N1KV setup is complete.

## Health Checks

To launch the dashboard URL created after successful installation of Overcloud, complete the following steps:

1. Go to <http://172.22.215.91> (URL provided after Overcloud deployment) and login as admin and use the password created in the overcloudrc file (under \$HOME of stack user).
2. Log into the system and navigate the tabs for any errors.

3. Update the system defaults.

# Update Default Quotas



Default Quotas \*

**Injected File Content Bytes \***

From here you can update the default quotas (max limits).

**Metadata Items \***

**RAM (MB) \***

**Key Pairs \***

**Length of Injected File Path \***

**Instances \***

**Injected Files \***

**VCPUs \***

**Total Size of Volumes and Snapshots (GB) \***

**Volume Snapshots \***

**Volumes \***

Cancel

Update Defaults

## Functional Validation

---

Functional Validation includes the following:

- Navigating the dashboard across the admin, project, users tab to spot any issues
- Creating Tenants, Networks, Routers and Instances.
- Create Multiple Tenants, multiple networks and instances within different networks for the same tenant and with additional volumes with the following criteria:
  - Successful creation of Instances through CLI and validated through dashboard
  - Login to VM from the console.
  - **Login to VM's through Floating IP's.**
  - Checking inter instance communication **for VM's within the same network and VM's in a different network** for the same tenants and with password less authentication.
  - **Reboot VM's and checking for VM evacuation**
  - **Check for the VLAN's created both in UCSM and also on the Nexus switches. The VLAN's should be available globally and also on the both port-channels created on each switch:**

```

Login to Nexus switch
conf term
show vlan | grep q-
show running-config interface port-channel 17-18
  
```



The basic flow of creating and deleting instances through command line horizon dashboard were tested. Creating multiple tenants and VLAN provisioning across Nexus switches and Cisco UCS Manager were verified while adding and deleting the instances.

---

For detailed information about validating Overcloud, refer to the Red Hat Linux Openstack Platform guide.

## Performance and Scale Testing

### Rally Testing and Measuring Compute Scaling

Rally benchmarking tool was used to check the solution for scale testing. This tool tells how OpenStack performs, notably under simulated load at scale. Later after the test is completed, Rally generates an HTML report based on captured data. For more details about this tool please refer <https://wiki.openstack.org/wiki/Rally>

The main purpose of running the tool was to generate a workload close to cloud but not to capture some benchmark data. Hence a limited amount of tuning has been attempted on the openstack side. None of the default kernel parameters like ulimits, pid\_max, libvirtd or nova parameters like osapi\_max\_limit or neutron api workers etc were modified, nor was any attempt done on Ceph side to extract the best performance from the configuration. It has to be noted that these may have to be tweaked to get the best results. This **was just an attempt to use a tool to create VM's simultaneously but not to do a real benchmarking exercise.**

### Prerequisites and Install

Rally was installed and setup on OpenStack Director Node.

The step-by-step install instructions for Rally is documented in <https://rally.readthedocs.org/en/latest/tutorial.html>

### Configuration and Tuning Details

A test bed with 3 controllers, 19 computes and 3 Ceph storages nodes was built to test this setup. The hardware and software specifications are same as mentioned earlier in this document. The following parameters were changed for rally testing.

Variable name	Location	Value
show_image_direct_utl	/etc/glance/glance-api.cfg	True
vif_plugging_is_fatal	/etc/nova/nova.conf	True
api_workers	/etc/neutron/plugin.ini	0
Time out Client	/etc/haproxy/haproxy.cfg	180

### Test Methodology

The following section describes different test case scenarios selected to test RHEL-OSP 7 environment. In this test, following benchmarking scenarios have been given to simulate multi-tenant workload at scale for VM and volume provisioning. Each benchmarking scenario will perform a small set of atomic operations, hence testing the simple use case.

### Rally Configuration Summary

1. Provision 1,000 instances from 1,000 bootable Ceph volumes.

2. Use Cirros-0.3.4-x86\_64-disk.img.
3. Create 200 tenants with 2 users per tenants
4. Create and authenticate these VMs.
5. Each tenant will have 1 Neutron network, a total of 200 Neutron networks. Cisco UCS and Nexus plugins provision these neutron networks mapped to VLAN with segmentation id in Cisco UCS manager and Nexus 9000 switches.
6. Tenant quotas for Cinder, Nova, and Neutron are set to unlimited for simplicity in this test to avoid **any failures just to avoid any quota issues while booting the VM's.**
7. **Concurrency of 3 has been used in Rally's task configuration. Rally script will be creating a constant load by running the scenarios given for a fixed number of times, possibly in parallel iteration and therefore simulating the concurrent requests from different users and tenants.**
8. Provisioned 2,000 instances from bootable volume based on the similar scenarios mentioned above.

#### Hardware

Following hardware has been used to run the Rally tests.

- Number of compute nodes (Cisco UCS B200 M4 Servers): 19
- The following hardware resources were available in the test bed to run the rally test described above with or without over-commitment ratios:

Hypervisor	vCPU	RAM (GB)	vCPU with over-commitment of 16	RAM (GB) with over-commitment of 1.5
overcloud-compute-0.localdomain	40	251.5	640	377.25
overcloud-compute-1.localdomain	32	251.5	512	377.25
overcloud-compute-10.localdomain	32	251.5	512	377.25
overcloud-compute-11.localdomain	32	251.5	512	377.25
overcloud-compute-12.localdomain	40	251.5	640	377.25
overcloud-compute-13.localdomain	40	251.5	640	377.25
overcloud-compute-14.localdomain	32	251.5	512	377.25
overcloud-compute-15.localdomain	40	251.5	640	377.25
overcloud-compute-16.localdomain	40	251.5	640	377.25
overcloud-compute-17.localdomain	40	251.5	640	377.25
overcloud-compute-18.localdomain	40	251.5	640	377.25
overcloud-compute-2.localdomain	32	251.5	512	377.25
overcloud-compute-3.localdomain	48	251.5	768	377.25
overcloud-compute-4.localdomain	40	251.5	640	377.25
overcloud-compute-5.localdomain	40	251.5	640	377.25
overcloud-compute-6.localdomain	40	251.5	640	377.25
overcloud-compute-7.localdomain	40	251.5	640	377.25
overcloud-compute-8.localdomain	40	251.5	640	377.25
overcloud-compute-9.localdomain	40	251.5	640	377.25
TOTAL	728	4779	11648	7168

- Number of Ceph Nodes: 3 ( 43 TB of usable space )
- Number of Controllers: 3



vCPU default over-commitment ratio is 1:16. However, it is recommended to use 1:4. It can be modified in `/etc/nova/nova.conf` `cpu_allocation_ratio` variable. With `cpu_allocation_ratio` of 4, 2913 ( $728 * 4$ ) instances can be created.

## Instance Sizing

Default `m1.tiny` flavor size was used for guest instances in this test.

## Rally Scenario Task Configuration

Rally runs different types of scenarios on the information provided in json format. Although Rally offers several different combination of scenarios to choose from, here we are focusing on testing how the system scales in a multi-tenant environment where each tenant environment has a given set of strategies

Below is the Rally task configuration used in validation. It takes different parameters for customization to run on different sets of scenarios. However, defaults are also set. Below .json file runs the `NovaServers.boot_server_from_volume` scenario.

In this reference environment, benchmarking contexts have been setup. Contexts in Rally allow to stage different types of environment in which benchmarking scenario is launched. In this test, environment such as number of tenants, number of users per tenant, number of neutron network per tenant, and users quota were specified.



Sample json file used for testing is provided in Appendix A.

---

The following parameters used in the json file are provided for reference

- Flavor : The size of the guest instance, e.g. m1.tiny
- Image : the name of the image file used for guest instances
- volume\_size : size of bootable volume, e.g. 10 GB
- quotas : Quota requirement of each tenant. For unlimited resources, value of -1 for cores, ram, volume, network, ports, and so on has been given.
- Tenants : Number of tenants to be created.
- users\_per\_tenant : Number of users created within each tenant.
- concurrency : amount of guest instances to run on each iteration
- times : Total number of iterations to be performed.

Rally script was executed by running the following command:

```
rally --verbose task start boot-from-volume.json --task-args \  
  '{"flavor_name": "m1.tiny", "volume_size": 1, "number_of_vms": 1000,  
  "image_name":  
  "cirros", "concurrency":3, "no_of_tenants":200, "users_per_tenants":2}'
```

## Rally Tests with 1000 Virtual Machines

It is recommended to start with smaller set of values for number of tenants, VMs, times, and concurrency to diagnose for any errors. Rally generates HTML based report after the task is completed as shown below. Load duration shows the time taken to run the specified scenario, while Full duration shows the total time taken by the overall benchmark task. Iteration shows how many times a specified scenario has been executed.

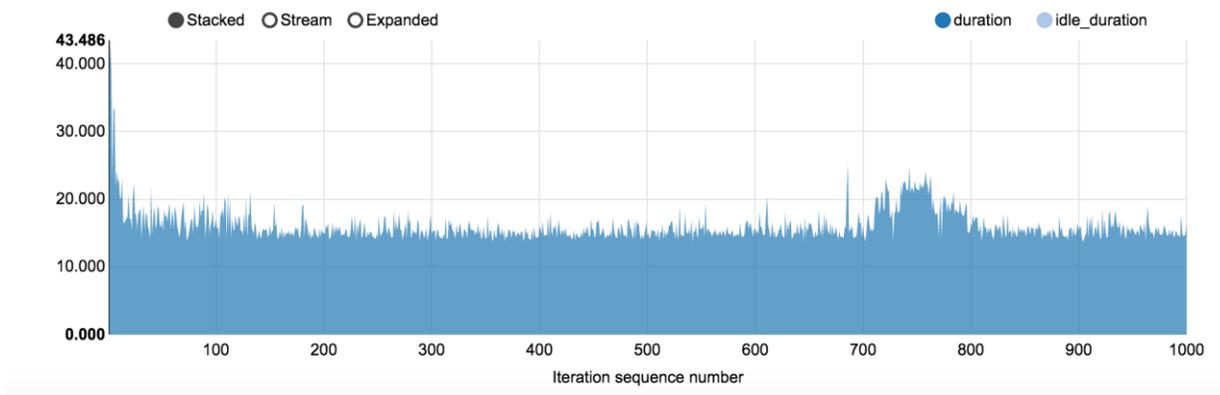
## NovaServers.boot\_server\_from\_volume (10,465.313s)

[Overview](#)
[Details](#)
[Input task](#)

Load duration: **5,383.225 s** Full duration: **10,465.313 s** Iterations: **1000** Failures: **0**

### Total durations

Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
cinder.create_volume	6.009	6.78	7.259	8.925	18.324	6.968	100.0%	1000
nova.boot_server	7	8.643	11.408	13.49	26.081	9.144	100.0%	1000
<b>total</b>	<b>13.649</b>	<b>15.404</b>	<b>18.521</b>	<b>20.913</b>	<b>43.486</b>	<b>16.112</b>	<b>100.0%</b>	<b>1000</b>



The figure above shows the time taken to provision each VM. The X-axis plots the number of VMs and the Y-axis, shows the total time taken to power-on each VM that includes both the time taken to provision the boot volume with cirros images and the nova boot time of the VM.

### Data Analysis

Based on total RAM available on 19 compute nodes with 1000 VM of 500Mb each, there is no memory swapping. Based on the above available hardware resources in the reference test bed and considering resource depletion, theoretically 4 GB can be assigned to each VM ( $1000 * 4 = 4000$  GB) without any memory over-commitment.

Very small CPU over-commitment was observed based on 728 physical cores available for 1000 instances. As mentioned earlier, the configuration had enough vCPU resources with recommended `cpu_allocation_ratio` of 4 to provision 1,000 instances.

Better performance and results have been observed by reducing the number of tenants and neutron network along with concurrency level.

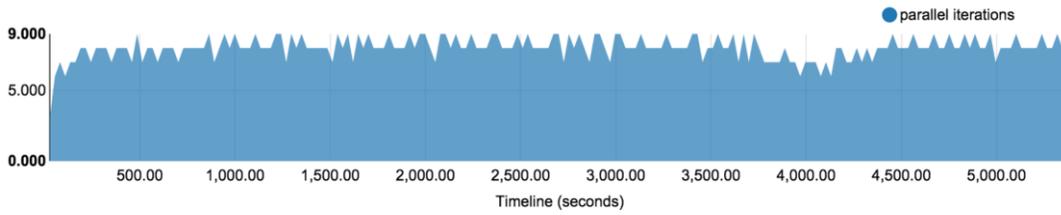
Different results have been observed with different image sizes. It takes longer to provision bootable volume, if the image size is larger.

Limited variations have been observed in the result sets even if similar scenario is run many times because of resource depletion.

Connection error has been observed because of HAProxy timeout, HAProxy being the top layer of the stack with respect to all incoming client connections. HAProxy timeout value is increased from 30s to 180s. It is observed that default timeout value is not sufficient to handle incoming Rally client connection requests.

### Load Profile

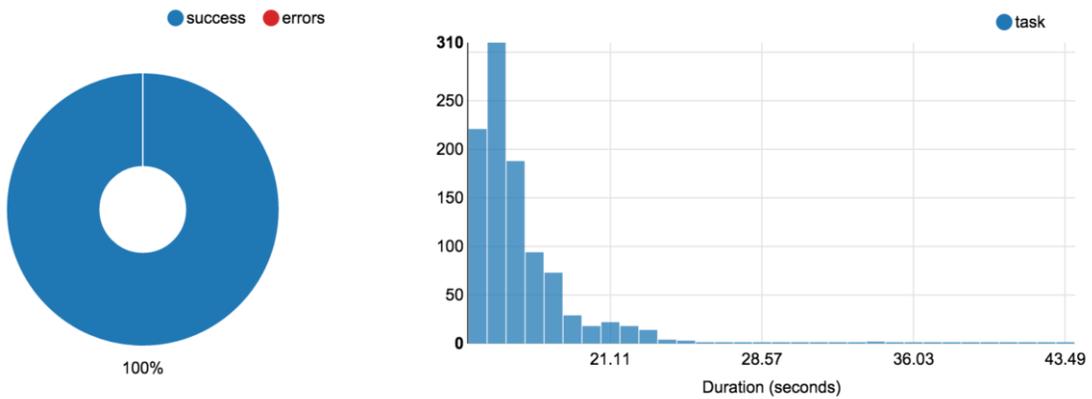
Load Profile



From the above it is evident that load profile has been consistent throughout the test.

### Workload Distribution

Distribution



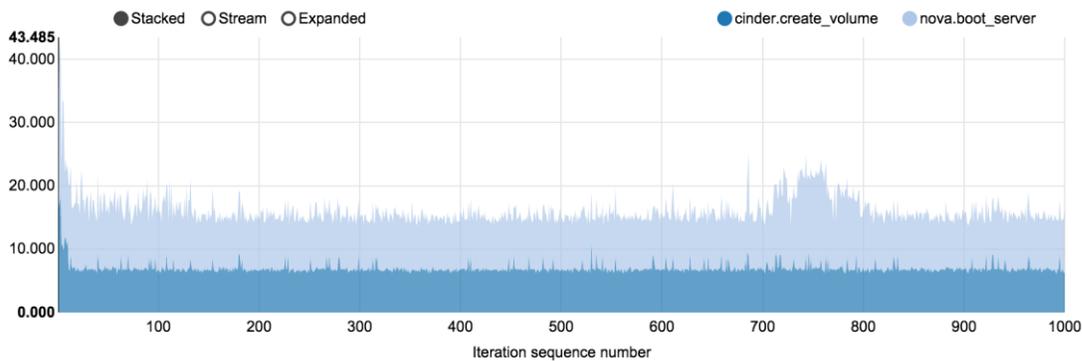
The graph above shows the distribution in seconds versus the number of instances provisioned.

### Atomic Actions

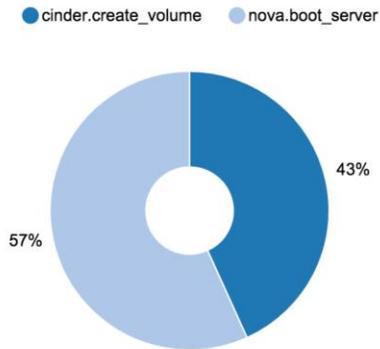
#### NovaServers.boot\_server\_from\_volume (10,465.313s)

Overview Details Input task

#### Atomic Action Durations



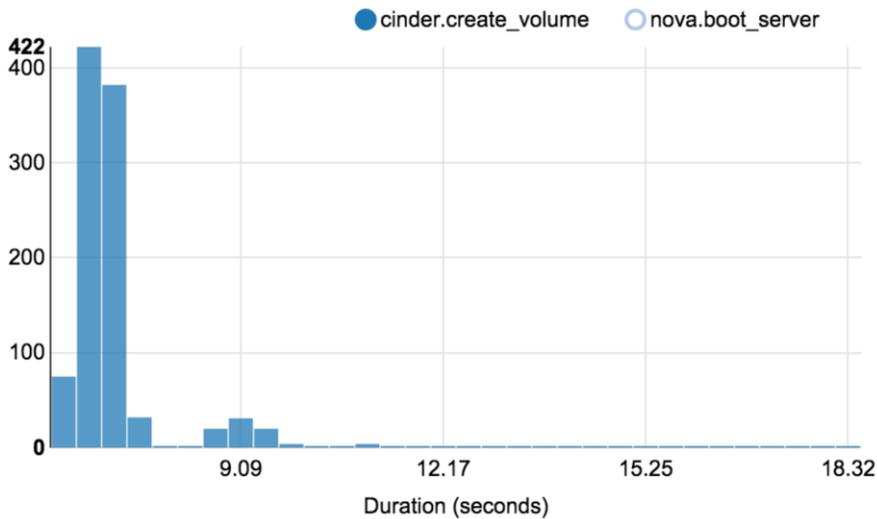
The graph shows the atomicity of the single Rally benchmarking operation. In this test, there were two major operations, cinder volume creation and nova boot of guest instances. It is observed that volume creation took little longer in the beginning and after that it remained consistent on an average of 7 seconds. Nova boot took on the average of 9.7 seconds for a single instance. Nova boot time was also consistent. However few spikes have been noticed especially after 700 iterations. This could vary depending on other internal OpenStack API calls within the controller.



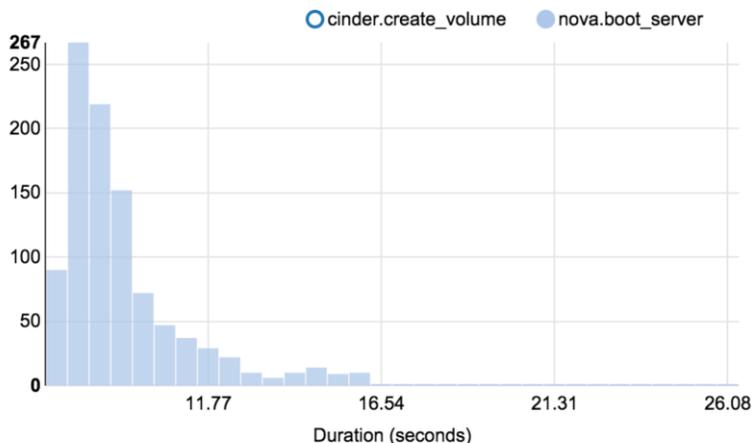
Out of the total time it took to boot from volume and provision the instance, 43% of the time was taken for creation of the volume, while the rest 57% for to create VM in its active state

### Cinder Create Volume

On an average, it took 7 seconds to provision single volume. You may observe a skewed average as the first 500 were provisioned in less than 5 seconds.



### Nova Boot Server



The average time to nova boot was observed around 8-9 seconds.

### Rally Tests with 2000 Virtual Machines

Using the same .json file, Rally task was executed for 2,000 VM's without changing any other parameters from the json file.

```
rally --verbose task start boot-from-volume.json --task-args '{"flavor_name":
"m1.tiny", "volume_size": 1, "number_of_vms": 2000, "image_name":
"cirros", "concurrency":3, "no_of_tenants":200, "users_per_tenants":2}'
```

### Data Analysis

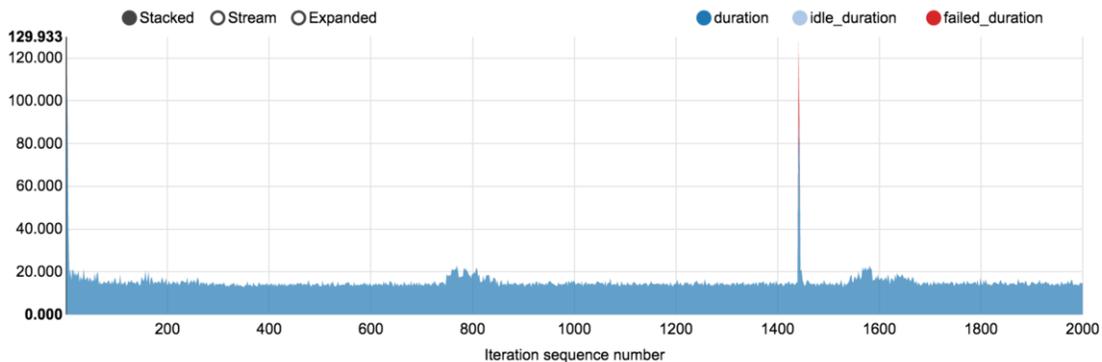
#### NovaServers.boot\_server\_from\_volume (16,348.329s)

Overview **Details** Failures Input task

Load duration: 10,260.951 s Full duration: 16,348.329 s Iterations: 2000 Failures: 1

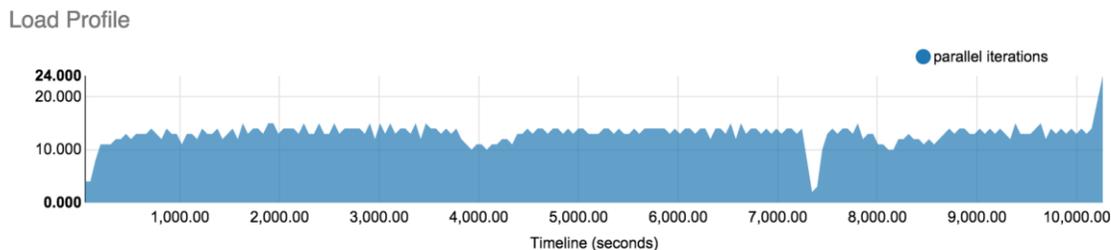
#### Total durations

Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
cinder.create_volume	5.432	6.122	6.391	6.648	53.871	6.298	100.0%	2000
nova.boot_server	5.816	8.345	11.488	12.721	138.982	9.018	100.0%	2000
<b>total</b>	<b>12.084</b>	<b>14.474</b>	<b>17.684</b>	<b>19.19</b>	<b>148.135</b>	<b>15.316</b>	<b>100.0%</b>	<b>2000</b>



It took on the average of 15 seconds to provision single VM in this test. At about 1441 iteration, we encountered one failure. This particular instance failed to provision and OpenStack API call eventually timed out. It is clearly evident from the graph as well. It took more than 120 seconds on this instance and in most cases after 120 sec, OpenStack API requests time out.

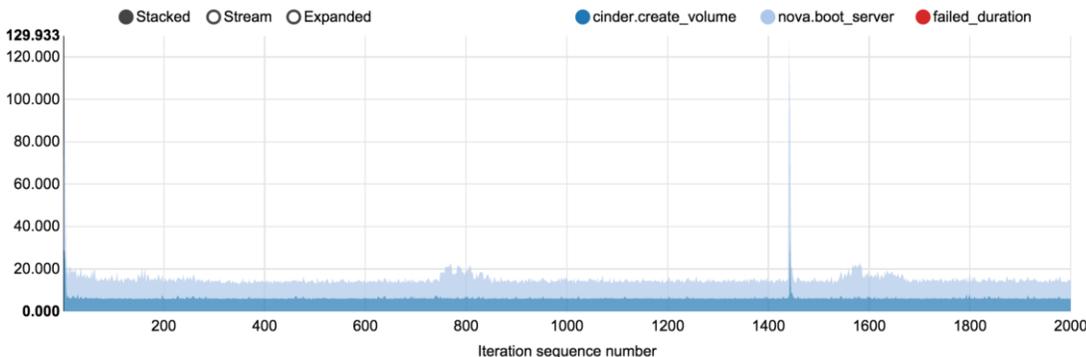
Load Profile



NovaServers.boot\_server\_from\_volume (16,348.329s)

- Overview
- Details
- Failures
- Input task

Atomic Action Durations



Similar to the 1000 VM tests, the volumes were consistently provisioned on an average of 6.3 seconds.

Cisco Plugins

During the Rally task execution, Cisco UCSM and Nexus ML2 plugins provisioned neutron segmentation id as VLAN in automated fashion.

**Nexus MI2 plugin provisioned tenant's VLAN and allowed them in the trunk as Rally provisioned the neutron network for each tenant.**

Similarly, in UCSM, the **tenant's VLANs** are also provisioned in the LAN cloud. Furthermore, tenant VLANs are added to the vNIC that carries tenant data traffic to the respective compute host where the instance is provisioned.



## Conclusion from Rally Tests

While the tests were not targeted as a benchmarking exercise, we can draw a few conclusions that could help us to plan the infrastructure.

The over commitment and concurrency play an important role on sporadic failures. Also it depends how much burst of work load do we expect in a real production environment and then test and tune for concurrency.

Minimal tuning was completed on either nova or on Ceph while running these tests. This is an iterative exercise but could have provided more insight for extracting better performance values.

Failures like timeouts will skew the result set. It is recommended to pay attention to the median and 90th percentile figures to understand the system behavior.

## Ceph Benchmark Tool for Ceph Scalability

The Ceph benchmarking tool was used in the configuration to test the scalability of storage nodes. The purpose of this testing done on this configuration as part of this CVD was not to do benchmarking but to provide steps on how to do the storage testing of nodes in OpenStack and provide some comparison data between Cisco UCS C240 M4 large and small factor servers. This is to help choose the right configuration based on the workload expected in cloud. Each of them have their own hardware characteristics and the performance data captured here should help to make an informed decision on the storage servers. However, the data presented below should not be considered as the optimal storage scalability values.

Ceph benchmarking tool (CBT) can be downloaded from <https://github.com/ceph/cbt>. It is an open source python script to test the performance of the Ceph clusters. It tests both the object and rados block devices scaling. Only Rados Block Device (rbd) tests were done. While there are three different categories for block devices testing that can be done, the most conservative, librbd fio was used in the test bed. Librbd fio, tests block storage performance of RBD volumes without KVM/QEMU configuration through librbd libraries. They give the closest approximation of KVM/QEMU performance. Please refer to the link above to configure the tool for testing.

The results obtained depend on several factors. The important ones that were included in the test bed are mentioned below.

Default value of rbd\_cache is true. This was turned to false purposely to suit some of the RDBMS workloads.

The read ahead configured was as default on the disks.

```
[root@overcloud-cephstorage-2 ~]# hdparm -a /dev/sde
/dev/sde:
read ahead      = 8192 (on)
```

The write cache policy on the LUNs was write through which is the default.

Read Policy: <b>Normal</b>	Actual Write Cache Policy: <b>Write Through</b>
IO Policy: <b>Direct</b>	Configured Write Cache Policy: <b>Write Through</b>

The io depth in the ceph parameter was 8.



Each VM by default can do 1GB of IO throughput. There was no qemu throttling or QOS policy implemented on the setup. Few VM's each running with full capacity can saturate the storage.

---



The tests were done to measure the IOPS and bandwidth as a whole on the storage cluster. This in turn will be shared by all the VM's running in the cloud. The values represent what storage can scale but not how many VM's can saturate them.

---

## Ceph Configuration

The CBT tests were done in two different installs; the first with 3 node of C240M4 Large form factor and the second with C240MS, the small form factor.

The Ceph configuration on a 3 node C240M4 LFF was as follows:

- 8 OSD's per node each with 6 TB HDD
- 2 SSD disks per node each with 4 partitions for Ceph journaling

A total usable space of around 43 TB with a replication factor of 3 on 3 nodes.

The Ceph configuration on each C240M4 SFF was as follows:

- 18 OSD's per node each with 1.2 TB HDD

- 4 SSD disks per node each with 5 partitions for Ceph journaling. Only 4 partitions from the last 2 SSD's were used as journals as the total requirement for journals was 18 and not 20.
- A total usable space of around 18 TB with a replication factor of 3 on 3 nodes.

### Create Virtual Machines for Ceph Testing

To create a virtual machine for Ceph testing, complete the following steps:

1. Create a master VM from RHEL-OSP 7.2 cloud image.
2. Install CBT and other packages as needed per web page above, on the new VM, called as Ceph master and add any additional interfaces in the VM. VM's need access to Ceph public storage as well.
3. As it is eth1 from the hypervisor through which VM communicates per RHEL-OSP configuration, you may have to temporarily open vlans in UCS on eth1 just for this testing purpose.
4. Once master image is created, suspend the VM, take a snapshot and then create multiple child VM's for actual CBT testing.
5. Configure the clients, adjust the parameters in CBT yaml file and start testing.

### Cisco UCS C240 M4 LFF Results

The following figures illustrate the results of this solution.

**Figure 15** C240M4 LFF Throughput in IO's Per Second

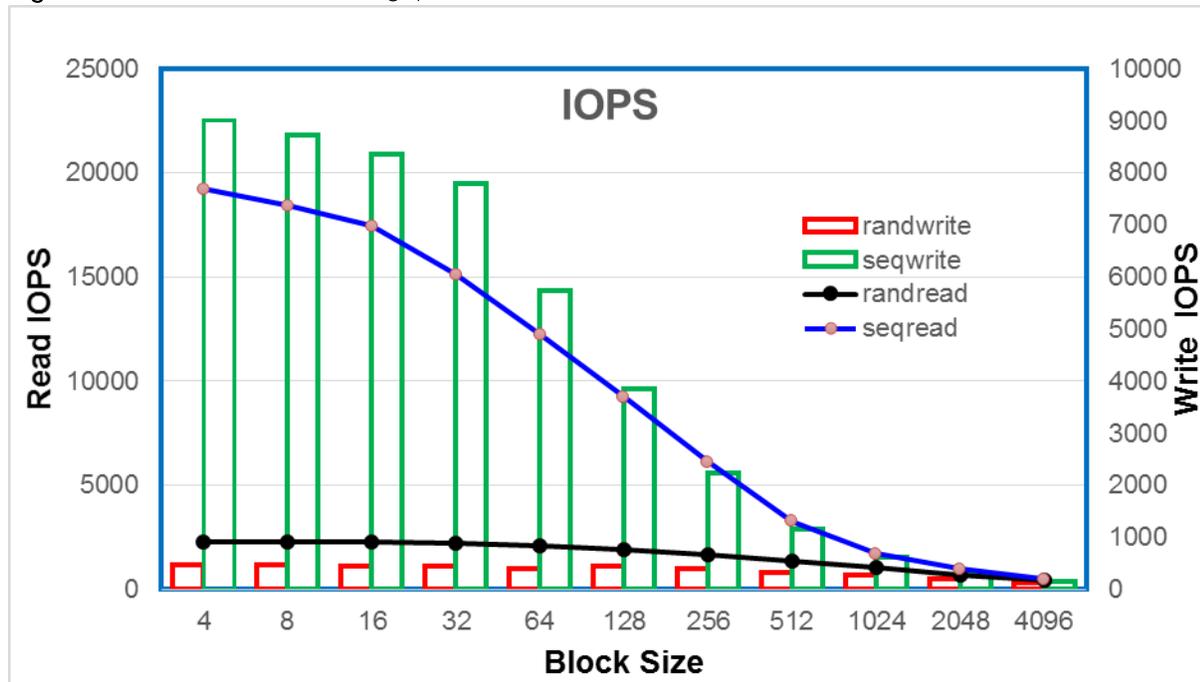


Figure 16 C240M4 LFF Bandwidth in MBPS

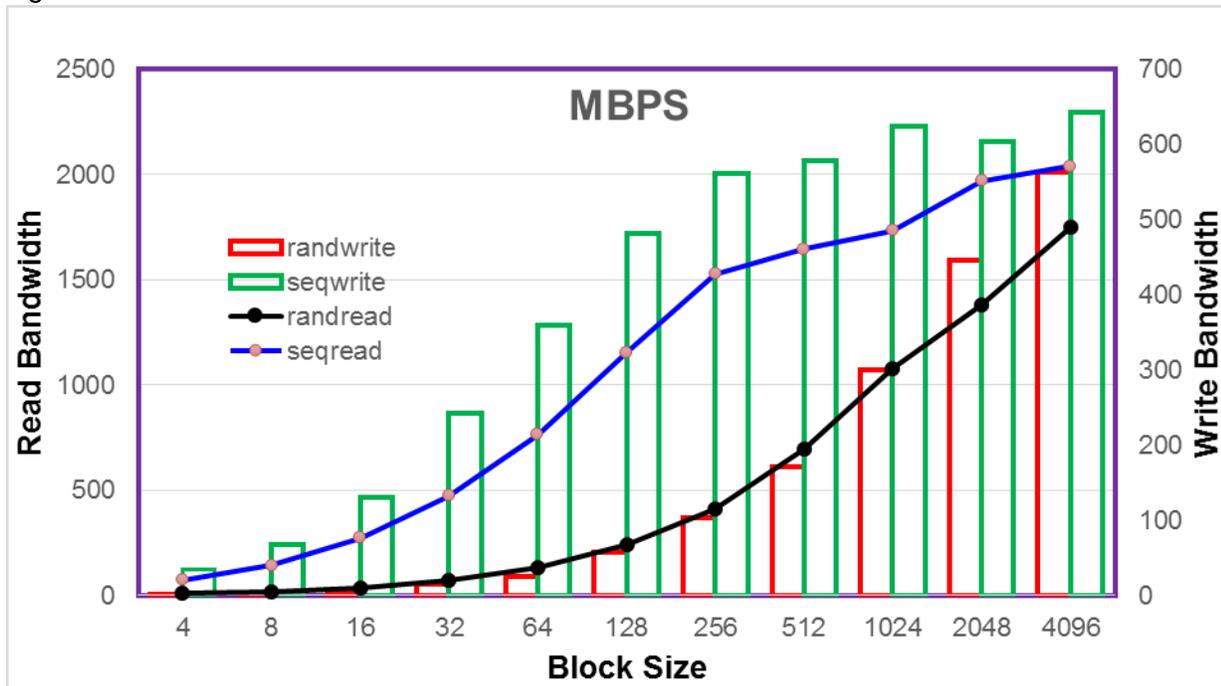
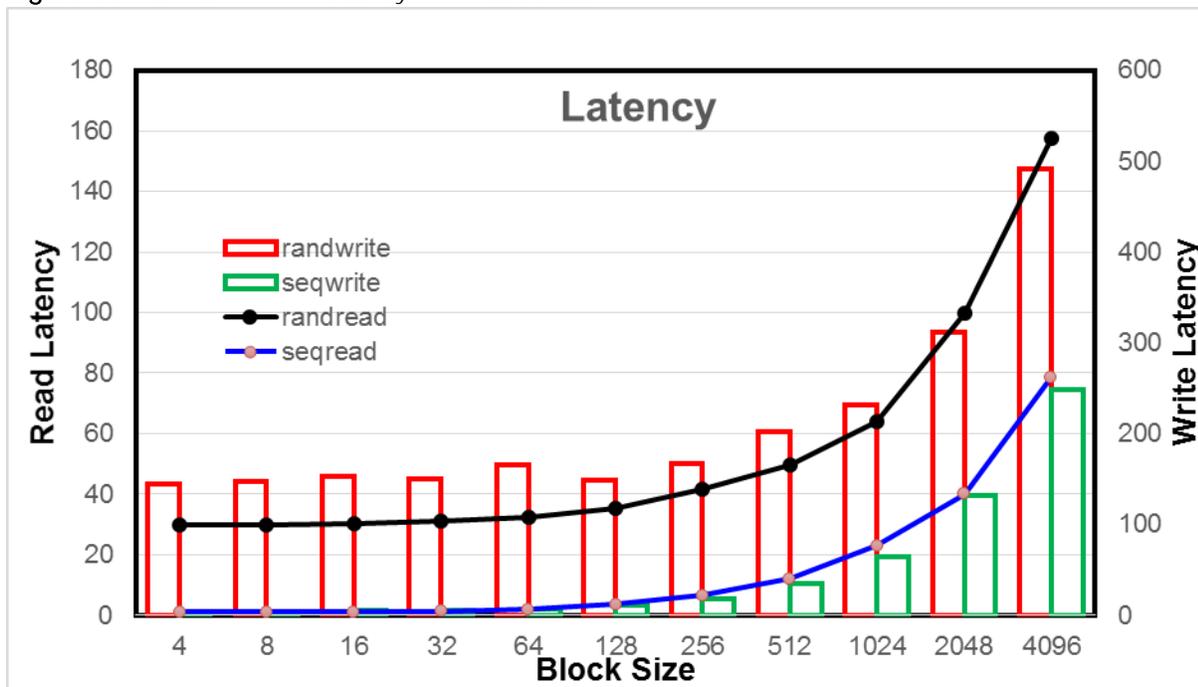


Figure 17 C240M4 LFF Latency in Milliseconds



Cisco UCS C240 M4 SFF Results

Figure 18 C240M4 SFF Throughput in IO's Per Second

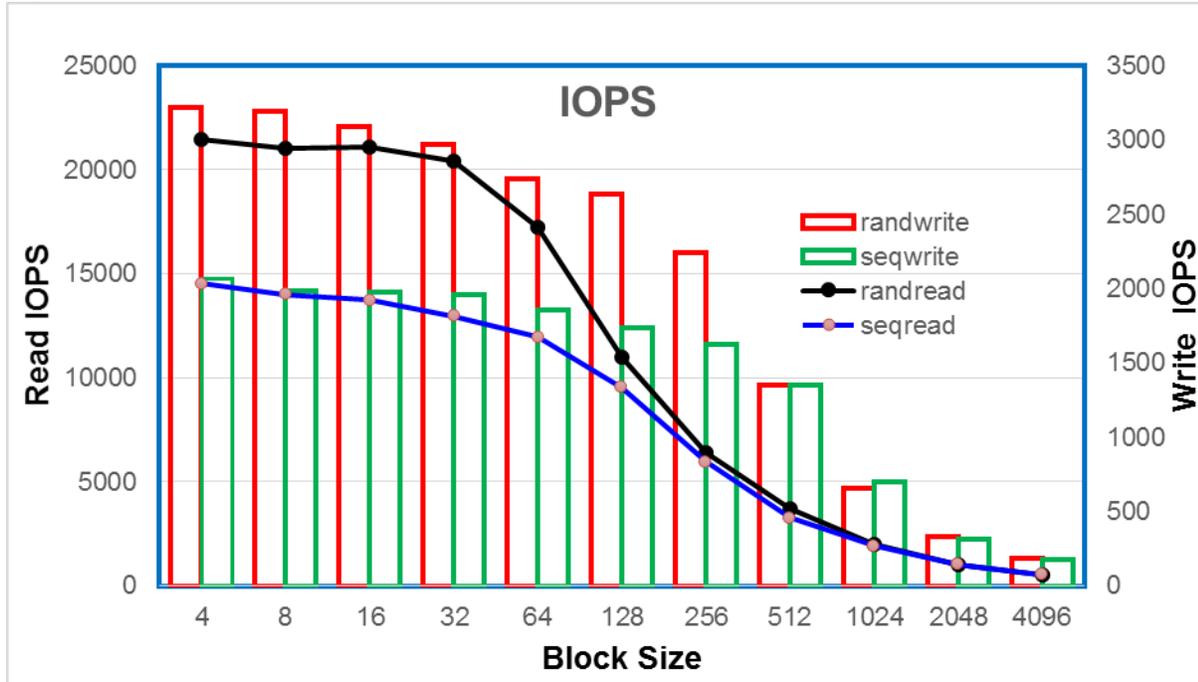


Figure 19 C240M4 SFF Bandwidth in MBPS

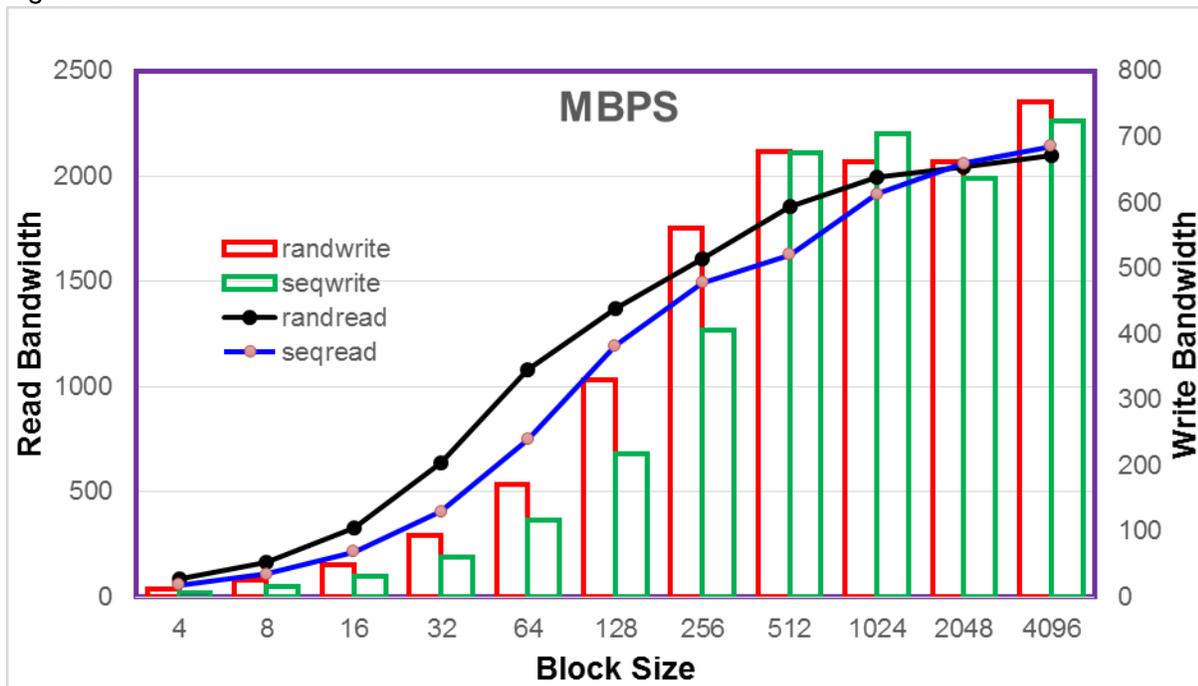
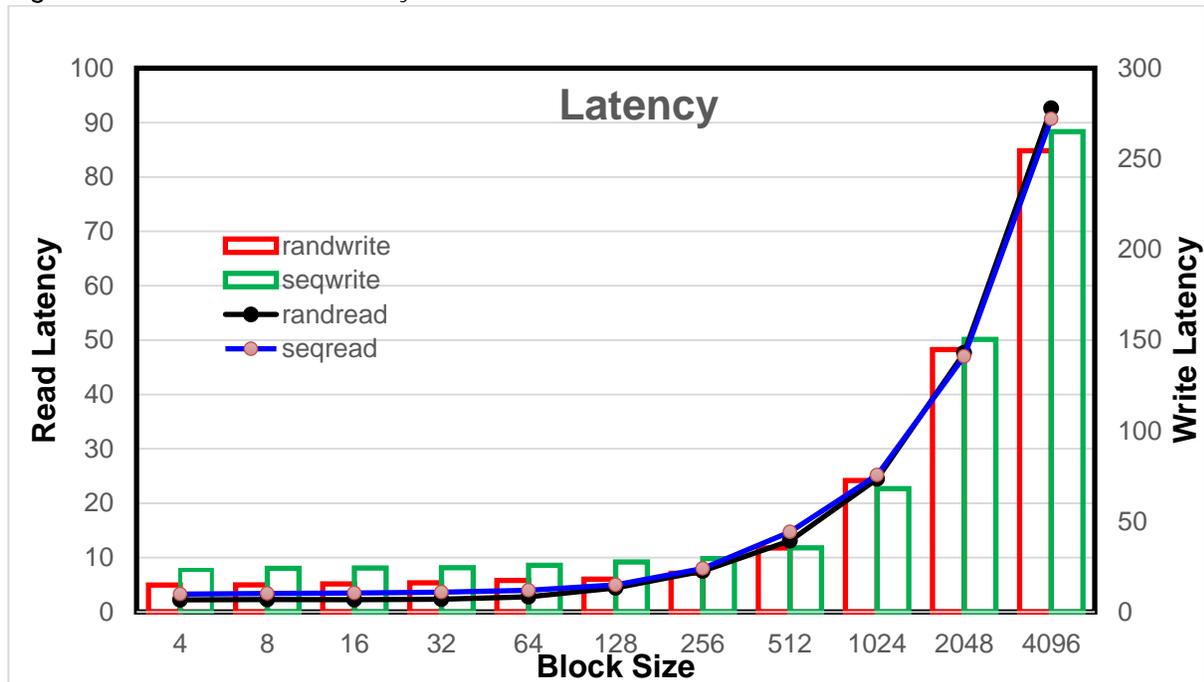


Figure 20 C240M4 SFF Latency in Milliseconds



### Analysis

- rbd\_cache as false seem to have reduced the values.
- librbd fio always gives lesser values over rbd fio and kernelrbd values and a conservative approach followed here.
- The latency values as expected were higher at higher block sizes.
- The SFF servers has more spindles and exhibits better random read and write performance at lower block sizes
- The sequential read and write performance of LFF server is at par or better than the SFF servers
- The 6 TB drives with 4k sectors on LFF servers observed to leverage better read ahead and hence a read performance, while the 1.2 TB drives with 512 byte sectors on SFF lagging behind.
- At 4 MB block sizes, both of them exhibit a similar or very close bandwidth values.
- The overall TB of usable space per Rack unit is higher on LFF servers.
- Minimal CPU or Memory overhead observed during the tests. These conclude that system will have sufficient head room during failures for recovery operation. The core/spindle ratio was higher as well on these boxes.
- As mentioned earlier the tests were conducted in a controlled environment. Increasing the VM's substantially without a control on IO on each might give poorer results. A separate layer of client side IO throttling has to be in place if this is the case.
- CBT only checks the storage performance of the Ceph cluster. The number of VM's configured is controlled through the yaml file.

# Live Migration

## Live Migration Introduction and Scope

Live migration refers to the process of moving a running [virtual machine](#) between different physical machines without disconnecting the [client](#) or application. Memory, storage, and network connectivity of the virtual machine are transferred from the source host to the destination host.

Live migration is crucial from operational perspective to provide continuous delivery of services running on the infrastructure. This allows for movement of the running virtual machine from one compute node to another one.

The most common use case for live migration is host maintenance - necessary for software patching or firmware/hardware/kernel upgrades. Second case is imminent host failure, like cooling issues, storage or networking problems.

Live migration helps in optimal resource placement across an entire datacenter. It allows reducing costs by **stacking more virtual machines on a group of hosts to save power. What's more it is possible to lower** network latency by placing combined instances close to each other. Live migration can also be used to increase resiliency and performance by separating noisy neighbors.

## Configuring Possibilities

All available values for `live_migration_flag`:

```
`live_migration_flag=VIR_MIGRATE_UNDEFINE_SOURCE, VIR_MIGRATE_PEER2PEER,
VIR_MIGRATE_LIVE, VIR_MIGRATE_AUTO_CONVERGE, VIR_MIGRATE_TUNNELED`
```

By default this flag have value:

```
`live_migration_flag=VIR_MIGRATE_UNDEFINE_SOURCE, VIR_MIGRATE_PEER2PEER,
VIR_MIGRATE_LIVE, VIR_MIGRATE_TUNNELED`
```

This flag is configured in `nova.conf` file on each compute node. To apply change restarting nova-compute service on each node is required.

## Tunneling Memory Transfer

The Tunneling option provides secure migration. In this model, hypervisor creates a point-to-point tunnel and sends encrypted (AES) data. This option also uses CPU for encrypting and decrypting transferred data. Without this, the data is transmitted in raw format. Tunneling is important from the perspective of security. Encryption of all data-in-transit ensures that the data cannot be captured.

The tunneling optioned is configured by adding the following value to flag:

```
`VIR_MIGRATE_TUNNELED`
```

## Auto Convergence

Busy enterprise workloads hosted on large sized VM's tend to dirty memory faster than the transfer rate achieved through **live guest migration. If the migration don't converge it is possible to use auto-converge**

feature (KVM + Qemu). This feature allows to auto-detect lack of convergence and trigger a throttle-down of the memory writes on a VM. This flag speed up process of Live Migration.

## Test Methodology

All tests are based on PerfkitBenchmarker and Yahoo! Cloud Serving Benchmark (YCSB) test suite for Cassandra database. Version of PerfkitBenchmarker used in tests is v1.1.0-39-g8374cc5.

Base benchmark specification assumes a configuration that includes 24 virtual machines: 10 with YCSB client to generate load and 14 with Cassandra nodes (1 with seed node).

VMs specification:

```
Client node (flavor:m1.medium):
    2 vCPUs
    4 GB RAM
    40 GB hard drive
Cassandra node (flavor:m1.xlarge)
    8 vCPUs
    16 GB RAM
    160 GB hard drive
```

All Cassandra VMs are created on the same compute node (it fits about 90 percent of RAM and 70 percent of CPU) in availability zone named **'workers'**. Similar situation applies to client VMs which are created on **other compute node in availability zone named 'clients'**.

For creating custom availability zones it requires to execute below nova command:

```
`nova aggregate-create <aggregate-name> <az-name>`
```

To attach some host to newly created availability zone execute:

```
`nova aggregate-add-host <az-id> <compute-hostname>`
```

To enable usage of availability zones additional config file (ycsb\_flags.yml) needs to be placed under `<perfkit_dir>/perfkitbenchmarker/configs/` with below content:

```
cassandra_ycsb:
  vm_groups:
    workers:
      vm_spec:
        OpenStack:
          machine_type: 'm1.xlarge'
          zone: 'workers'
    clients:
      vm_spec:
        OpenStack:
          machine_type: 'm1.medium'
          zone: 'clients'
```

The oversubscription level for CPU was equal to 2.8 (virtual core count to physical core count ratio) and for the memory it was 0.93 (virtual RAM count to physical RAM count ratio).

Cassandra is the choice for testing methodology as it is interesting in perspective of the cloud and it's easily scalable using built-in mechanisms. What's more, Cassandra is more RAM intensive compared to other, more classical DB's (RDBMS), which best shows impact of Live Migration process.

Yahoo Cloud Solution Benchmark (YCSB) is a tool to generate close to real workload on database. In test configuration database is loaded by read and update operations in proportion 3:1. Proportions are based on "[Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters](#)" from Delft University of Technology.

For configuring YCSB use workload file like below:

```
workload=com.yahoo.ycsb.workloads.CoreWorkload
fieldcount=20
fieldlength=255
readallfields=true
writeallfields=true

readproportion=0.75
updateproportion=0.25
scanproportion=0
insertproportion=0

requestdistribution=zipfian
operationcount=2100000
recordcount=5000000
```

To run tests use command as below:

```
`/bin/python2.7 /home/stack/PerfKitBenchmark/pkb.py --num_vms=14 \
--run_uri=tunl4 --ycsb_timelimit=6000 --benchmarks=cassandra_ycsb --
ycsb_workload_files=/home/stack/workload --ycsb_operation_count= 2100000 --
flagfile=/home/stack/PerfKitBenchmark/cg_flags --
benchmark_config_file=ycsb_flags.yml --ycsb_client_vms=10 --
ycsb_record_count=5000000`
```

Flags file (cg\_flags) contains OpenStack configurations with network names, used image and size of scratch disk:

```
--cloud=OpenStack
--openstack_private_network=int_net
--openstack_public_network=ext-net
--image=ubuntu-14.04
--scratch_disk_size=20
```

During the benchmark all VMs are migrated one by one to other compute nodes, migration time of each VM is counted and additionally various performance metrics from Cassandra cluster are collected. In the same time metrics - RAM, CPU and network traffic - from compute node containing Cassandra servers are obtained.

To start each migration use below command. Make sure to invoke migration during YCSB benchmarking *READ/UPDATE* operations on Cassandra. Destination compute node should not be the same as used for YCSB clients (availability zone 'clients').

```
`nova live-migration <vm-name> <destination-compute-name>`
```

## Results

Auto convergence is configured by adding the following value to flag:

```
`VIR_MIGRATE_AUTO_CONVERGE`
```

Average VM migration time in all combinations of this flags during Cassandra test based on PerfkitBenchmarker and YCSB test suite. This test suite is running in configuration of 14 Cassandra nodes and 10 clients. All Cassandra nodes were located on the same compute node (it fits about 90 percent of RAM and 70 percent of CPU) and then migrated one by one to other available computes.

<b>Test case</b>	<b>Average VM migration time during Cassandra test</b>
<i>Tunneling enabled, converged disabled</i>	385 sec
<i>Tunneling enabled, converged enabled</i>	167 sec
<i>Tunneling disabled, converged disabled</i>	60 sec
<i>Tunneling disabled, converged enabled</i>	54 sec

## Recommendations

Based on gathered results best time of migration is possible with disabled tunneling and enabled auto converge option. It can be used just in internal or test environments because from security perspective disabling tunneling is a disadvantage and enterprise setups should avoid to disable it.

To configure this option set up below line in `live_migration_flag` and restart nova-compute service:

```
`live_migration_flag=VIR_MIGRATE_UNDEFINE_SOURCE, VIR_MIGRATE_PEER2PEER, VIR_MIGRATE_LIVE, VIR_MIGRATE_AUTO_CONVERGE`
```

Unencrypted LM traffic is not a flaw, but allowing to traverse that traffic on a compromised network could be. Separating the LM traffic from API traffic will resolve that issue, unfortunately such configuration is not possible in current version of OpenStack (Kilo).

Current possibilities shows that best from perspective of speed and security configuration of migration flags is to use tunneling with auto converge. To set up this edit `nova.conf` on each compute to below value in `live_migration_flag` and restart nova-compute service:

```
`live_migration_flag=VIR_MIGRATE_UNDEFINE_SOURCE, VIR_MIGRATE_PEER2PEER, VIR_MIGRATE_LIVE, VIR_MIGRATE_AUTO_CONVERGE, VIR_MIGRATE_TUNNELED`
```

## Upscaling the POD

Scaling up the POD with growing business needs is a must. As business grows we need to add both compute and storage as needed by adding more hosts.

An attempt is made to scale up compute and storage. You may have to follow the steps below with any documented workarounds to add compute and storage nodes to the cluster.

### Scale Up Storage Nodes

#### Provision the New Server in Cisco UCS

To provision the new server in UCS, complete the following steps:

1. Rack the new C240M4 server(s). There is a single ceph.yaml in the current OpenStack version. Populate the hard disks in these storage servers in the same order as they exist in other servers.
2. Attach Console and discover the storage server(s) in UCS. Factory reset to defaults if needed and make them UCS managed.
3. [Refer to this section](#) for creating service profiles from Storage template. Create a new service profile from the template. Unbind the template and remove the storage policy that was attached to it earlier and associate the service profile to the server.
4. Upgrade firmware if needed.



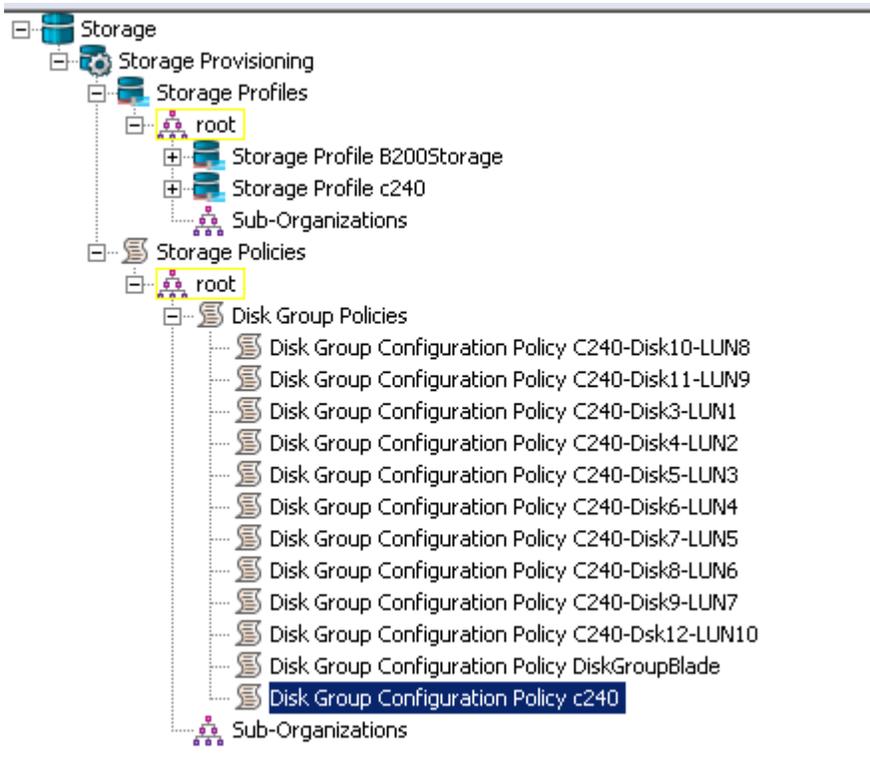
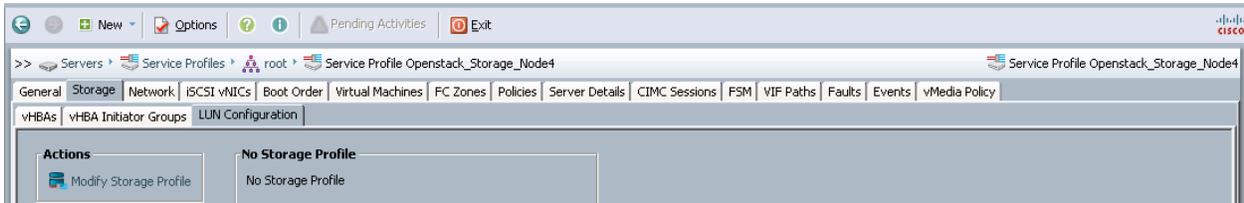
Check the installed firmware on the new node and make sure that it is upgraded to the same version as other storage servers.

Servers					
Server 1 (StCisco UCS C240 M4L)					
Adapter					
BIOS	Cisco UCS C240 M4L	C240M4.2.0.6a.0.05122015...	C240M4.2.0.6a.0.05122015...	C240M4.2.0.8b.0.08062015...	Ready
Board	CCisco UCS C240 M4L	13.0	13.0	N/A	Ready
CIMC	C/Cisco UCS C240 M4L	2.0(6d)	2.0(6d)	2.0(8g)	Ready
Server 2 (StCisco UCS C240 M4L)					
Adapter					
BIOS	Cisco UCS C240 M4L	C240M4.2.0.6a.0.05122015...	C240M4.2.0.6a.0.05122015...	C240M4.2.0.8b.0.08062015...	Ready
Board	CCisco UCS C240 M4L	13.0	13.0	N/A	Ready
CIMC	C/Cisco UCS C240 M4L	2.0(6d)	2.0(6d)	2.0(8g)	Ready
Server 3 (StCisco UCS C240 M4L)					
Adapter					
BIOS	Cisco UCS C240 M4L	C240M4.2.0.6a.0.05122015...	C240M4.2.0.6a.0.05122015...	C240M4.2.0.8b.0.08062015...	Ready
Board	CCisco UCS C240 M4L	13.0	13.0	N/A	Ready
CIMC	C/Cisco UCS C240 M4L	2.0(6d)	2.0(6d)	2.0(8g)	Ready
Server 4 (StCisco UCS C240 M4L)					
Adapter					
BIOS	Cisco UCS C240 M4L	C240M4.2.0.6a.0.05122015...	C240M4.2.0.6a.0.05122015...	C240M4.2.0.3d.0.11112014...	Ready
Board	CCisco UCS C240 M4L	13.0	13.0	N/A	Ready
CIMC	C/Cisco UCS C240 M4L	2.0(6d)	2.0(6d)	2.0(3i)	Ready

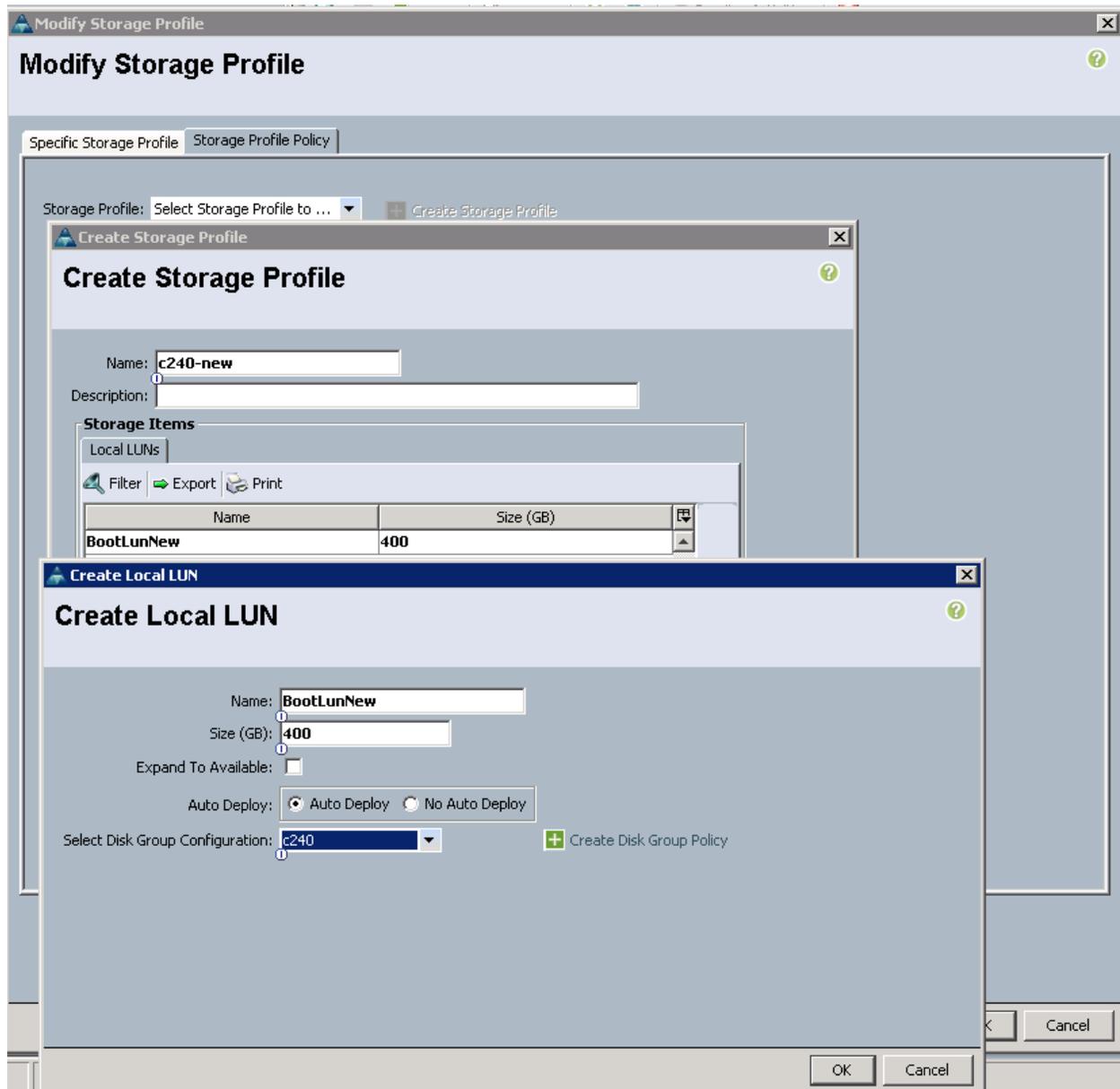
5. Create a new Storage Profile for Disks.

Before creating the storage profile, login to the equipment tab and make sure that all the new storage servers have the disks in place and they are physically on the same slots at par with other storage servers.

Since we used the storage profile earlier with other servers we cannot use them right away. The reason being the luns have to be added to the server in the same way as was done earlier. In case you are discovering more than one storage server at this stage, a single new profile created as below will serve the purpose. While creating this new storage profile, we can reuse the existing disk group configuration policies created earlier.



6. Go to the service profile of the new server and to the storage tab to create a new storage profile as shown below.

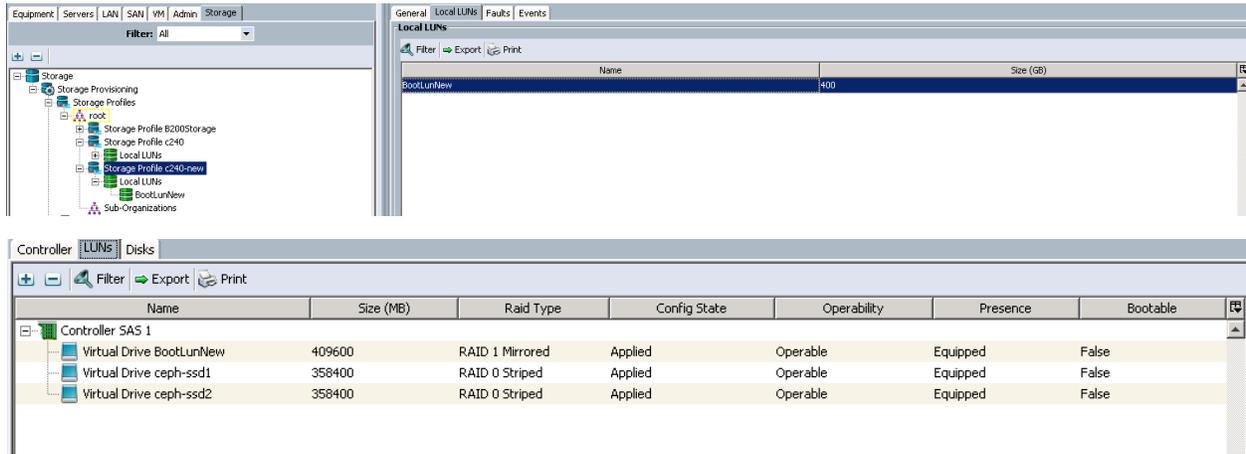


7. Attach this storage profile to the service profile. This will create the first boot lun LUN-0 on the server. Go back to the equipment tab and inventory/storage to check that this is the first Lun is added. This will be the boot lun LUN-0 that will be visible to the server bios. In case of multiple servers being added in this step, attach the new storage profile created above to all these service profiles. This in turn will create LUN-0 in all the nodes.



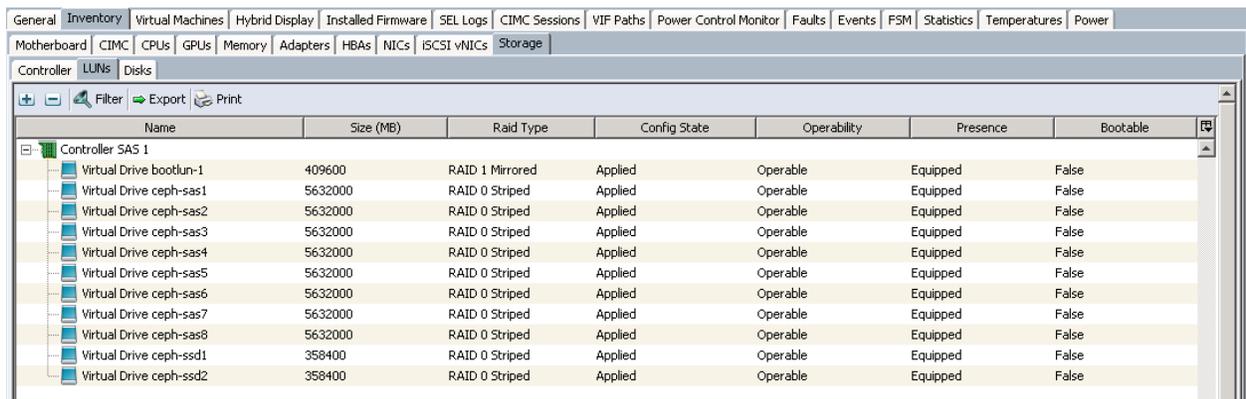
A subsequent update to this storage profile will be propagated across all these new service profiles.

8. Go to Storage tab in UCSM and update the storage profile.
9. Create and attach SSD luns, which will be LUN-1 and LUN-2. Wait few minutes to make sure that all the new servers get these luns in the same order, boot as LUN-0, ceph-ssd1 as LUN-1 and ceph-ssd2 disk as LUN-2.



This will be consistent with other servers and we can expect sda for boot lun and sdb and sdc for SSD LUNs being used with the journals.

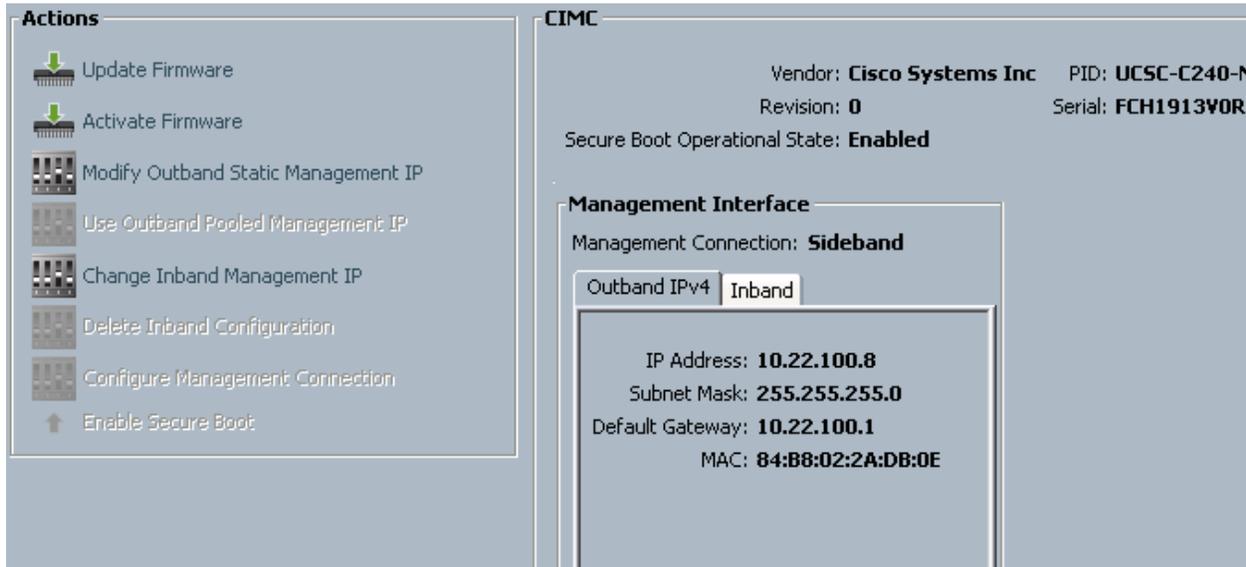
10. Add all the HDD LUNs.



The steps above do not represent the actual boot order. You may have to observe the actual boot order from KVM console to verify.

If the boot disks are being repurposed and are not new, go ahead and re-initialize the boot lun through bios. Boot server, CTR-R, F2 and reinitialize the VD for the boot LUNs.

1. Get the hardware inventory needed introspection.
2. Go to the Equipment tab > Inventory > CIMC and get the IPMI address.



3. Under the same Inventory tab go to NIC subtab and get the pxe mac address of the server. The same inventory should have the CPU and memory details.
4. Specify the NIC order in the service profile. This should be the same as the other storage servers with provisioning interface as the first one.
5. Check the boot policy of the server. Validate that this is same as other storage servers. It should be LAN PXE first followed by local LUN.

## Run Introspection

To run Introspection, complete the following steps:

1. Prepare json file for introspection:

```
stack@osp7-director ~]$ cat storage-node.json
{
  "nodes": [
    {
      "pm_user": "admin",
      "pm_password": "password",
      "pm_type": "pxe_ipmitool",
      "pm_addr": "10.22.100.8",
      "mac": [
        "00:25:b5:00:00:59"
      ],
      "memory": "262144",
      "disk": "400",
      "arch": "x86_64",
      "cpu": "40"
    }
  ]
}
```

2. Check IPMI Connectivity:

```
ipmitool -I lanplus -H 10.22.100.8 -U -P <password> chassis power off
```

Chassis Power Control: Down/Off

3. Initialize Boot Luns; in case you are reusing old disks it is recommended to initialize the boot luns.
4. Run discovery and introspection.

```
[stack@osp7-director ~]$ openstack baremetal import --json ~/storage-node.json
openstack baremetal configure boot
openstack baremetal list
```

The new node is added as shown below:

```
[stack@osp7-director ~]$ ironic node-list
```

UUID	Name	Instance UUID	Power State	Provision State	Maintenance
b7dde876-354a-4688-8550-aec8f64c582c	None	a23af643-51c8-4f59-881c-77a9d5e1557f	power on	active	False
e4563ca5-2f12-4e08-9905-f770f740ad2b	None	e902ad92-f600-41ec-a525-32218be1ee11	power on	active	False
285965a9-9713-4301-8ad5-7aa3ef5dd1c2	None	f01d22ef-18a4-4f01-b4c1-1ffb6f1d9262	power on	active	False
b4dc04ac-0c69-4000-9c4d-2d82d141905f	None	869662d6-e5ae-4724-82fb-3eb9a6f74e5b	power on	active	False
036cae70-bdee-427c-987c-a6a2d8a32292	None	ab811350-d49e-4e3e-bad7-9afeb6e1d10a	power on	active	False
8570c96e-f9cd-44ff-a1d8-0252bc405c24	None	7d9caa7c-4e53-4832-983c-44706def4d42	power on	active	False
af46cd81-c78e-47c5-94e3-44d9d669410c	None	211bb71c-2e72-44b1-a867-5f7babf4d4bb	power on	active	False
19260dbb-29a9-4810-b39d-85cc6e1d886f	None	b4a04e95-6624-4f13-8a14-5ef1e6742b94	power on	active	False
d4dae332-4595-43be-9b63-5a64331ea33b	None	db50a27a-4699-4fd9-9687-ec5403db3409	power on	active	False
179befe6-2510-4311-ad9f-4880454fdaff	None	47e64b48-f105-49f8-ae40-f04db9c39313	power on	active	False
ff0dadfe-e2f3-408f-b69d-01398bb9699d	None	0a1f7293-d9ee-423c-80bc-96125d29d924	power on	active	False
b59f57e3-d5e1-499a-80c1-aac0c78c9534	None	6be9ea47-39c3-4b4e-b755-a866e47b8398	power on	active	False
1132c423-7449-40b5-935a-0f989f61813f	None	None	power off	available	False

```
[stack@osp7-director ~]$ ironic node-set-maintenance 1132c423-7449-40b5-935a-0f989f61813f true
[stack@osp7-director ~]$ openstack baremetal introspection start 1132c423-7449-40b5-935a-0f989f61813f
[stack@osp7-director ~]$ openstack baremetal introspection status 1132c423-7449-40b5-935a-0f989f61813f
```

```
[stack@osp7-director ~]$ openstack baremetal introspection \
status 1132c423-7449-40b5-935a-0f989f61813f
```

Field	value
error	None
finished	True

5. Repeat the steps above if you want to add multiple nodes.
6. Wait till the introspection is complete. The status command should yield finished as True and Error as none.

```
ironic node-set-maintenance 1132c423-7449-40b5-935a-0f989f61813f false
```

```
[stack@osp7-director ~]$ ironic node-set-maintenance 1132c423-7449-40b5-935a-0f989f61813f false
[stack@osp7-director ~]$ ironic node-list
```

UUID	Name	Instance UUID	Power State	Provision State	Maintenance
b7dde876-354a-4688-8550-aec8f64c582c	None	a23af643-51c8-4f59-881c-77a9d5e1557f	power on	active	False
e4563ca5-2f12-4e08-9905-f770f740ad2b	None	e902ad92-f600-41ec-a525-32218be1ee11	power on	active	False
285965a9-9713-4301-8ad5-7aa3ef5dd1c2	None	f01d22ef-18a4-4f01-b4c1-1ffb6f1d9262	power on	active	False
b4dc04ac-0c69-4000-9c4d-2d82d141905f	None	869662d6-e5ae-4724-82fb-3eb9a6f74e5b	power on	active	False
036cae70-bdee-427c-987c-a6a2d8a32292	None	ab811350-d49e-4e3e-bad7-9afeb6e1d10a	power on	active	False
8570c96e-f9cd-44ff-a1d8-0252bc405c24	None	7d9caa7c-4e53-4832-983c-44706def4d42	power on	active	False
af46cd81-c78e-47c5-94e3-44d9d669410c	None	211bb71c-2e72-44b1-a867-5f7babf4d4bb	power on	active	False
19260dbb-29a9-4810-b39d-85cc6e1d886f	None	b4a04e95-6624-4f13-8a14-5ef1e6742b94	power on	active	False
d4dae332-4595-43be-9b63-5a64331ea33b	None	db50a27a-4699-4fd9-9687-ec5403db3409	power on	active	False
179befe6-2510-4311-ad9f-4880454fdaff	None	47e64b48-f105-49f8-ae40-f04db9c39313	power on	active	False
ff0dadfe-e2f3-408f-b69d-01398bb9699d	None	0a1f7293-d9ee-423c-80bc-96125d29d924	power on	active	False
b59f57e3-d5e1-499a-80c1-aac0c78c9534	None	6be9ea47-39c3-4b4e-b755-a866e47b8398	power on	active	False
1132c423-7449-40b5-935a-0f989f61813f	None	None	power off	available	False

7. Update node properties:

```
[stack@osp7-director ~]$ ironic node-update 1132c423-7449-40b5-935a-0f989f61813f \
> add properties/capabilities='profile:CephStorage,boot_option:local'

[stack@osp7-director ~]$ glance image-list | grep deploy | awk '{print $2" " $4}'
ca9667c9-de8f-4df1-95d5-b5f8b6fb8f4d bm-deploy-kernel
404d0e44-e5c7-4b46-8339-c451441b3f55 bm-deploy-ramdisk
```

8. If missing update as shown below:

```
[stack@osp7-director ~]$ ironic node-update \
1132c423-7449-40b5-935a-0f989f61813f \
> add driver_info/deploy_kernel='ca9667c9-de8f-4df1-95d5-b5f8b6fb8f4d';

[stack@osp7-director ~]$ ironic node-update 1132c423-7449-40b5-935a-0f989f61813f \
> add driver_info/deploy_ramdisk='404d0e44-e5c7-4b46-8339-c451441b3f55';
```

9. Check the status of added entries:

```
[stack@osp7-director ~]$ ironic node-show 1132c423-7449-40b5-935a-0f989f61813f
```

Property	value
target_power_state	None
extra	{u'newly_discovered': u'true', u'block_devices': {u'serials': [u'678da6e715b763c01e3d7f3617a53d6b', u'678da6e715b763c01e3d7f7f1bfbea33', u'678da6e715b763c01e3d7f9b1da6e6c9', u'678da6e715b763c01e3d7fd421130c43', u'678da6e715b763c01e3d7fec227fe608', u'678da6e715b763c01e3d800223cf9404', u'678da6e715b763c01e3d801524f5c3e2', u'678da6e715b763c01e3d802826123e2d', u'678da6e715b763c01e3d803e27688ece', u'678da6e715b763c01e3d805628d5fc89', u'678da6e715b763c01e3d806929f9d0b8']}, u'hardware_swift_object': u'extra_hardware-1132c423-7449-40b5-935a-0f989f61813f'}
last_error	None
updated_at	2016-01-29T03:05:13+00:00
maintenance_reason	None
provision_state	available
uuid	1132c423-7449-40b5-935a-0f989f61813f
console_enabled	False
target_provision_state	None
maintenance	False
inspection_started_at	None
inspection_finished_at	None
power_state	power off
driver	pxe_ipmitool
reservation	None
properties	{u'memory_mb': u'262144', u'cpu_arch': u'x86_64', u'local_gb': u'399', u'cpus': u'40', u'capabilities': u'profile:CephStorage,boot_option:local'}
instance_uuid	None
name	None
driver_info	{u'deploy_kernel': u'ca9667c9-de8f-4df1-95d5-b5f8b6fb8f4d', u'ipmi_address': u'10.22.100.8', u'ipmi_username': u'admin', u'ipmi_password': u'*****', u'deploy_ramdisk': u'404d0e44-e5c7-4b46-8339-c451441b3f55'}
created_at	2016-01-29T02:41:27+00:00
driver_internal_info	{}
chassis_uuid	
instance_info	{}

## Run Overcloud Deployment

The number of storage nodes has been incremented to 4 from 3. Here the number '4' indicates the total number of storage nodes in Overcloud.

```
#!/bin/bash
openstack overcloud deploy --templates --ceph-storage-scale 4 \
-e /usr/share/openstack-tripleo-heat-templates/overcloud-resource-registry-
puppet.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-
isolation.yaml \
-e /home/stack/templates/network-environment.yaml \
-e /home/stack/templates/storage-environment.yaml \
-e /home/stack/templates/cisco-plugins.yaml \
-e /home/stack/templates/post_config.yaml \
--verbose --debug --log-file overcloud_storage-add.log
```

Addition of nodes complete with the following message:

```
INFO: rdomanager_oscpugin.v1.overcloud_deploy.DeployOvercloud Stack found, will
be doing a stack update
Overcloud Endpoint: http://172.22.215.91:5000/v2.0/
Overcloud Deployed
DEBUG: openstackclient.shell clean_up DeployOvercloud
```

From overcloud deploy log

```
[2016-01-28 19:12:17,735] DEBUG cliff.commandmanager found command
'hypervisor_stats_show'
....
....
[2016-01-28 19:54:48,212] DEBUG openstackclient.shell clean_up
DeployOvercloud
The deployment script ran for 42 minutes.
```

### Post Deployment Health Checks

To perform the post-deployment health checks, complete the following steps:

1. Check with ironic and nova commands:

```
[stack@osp7-director ~]$ ironic node-list
```

UUID	Name	Instance UUID	Power State	Provision State	Maintenance
b7dde876-354a-4688-8550-aec8f64c582c	None	a23af643-51c8-4f59-881c-77a9d5e1557f	power on	active	False
e4563ca5-2f12-4e08-9905-f770f740ad2b	None	e902ad92-f600-41ec-a525-32218be1ee11	power on	active	False
285965a9-9713-4301-8ad5-7aa3ef5dd1c2	None	f01d22ef-18a4-4f01-b4c1-1ffb6f1d9262	power on	active	False
b4dc04ac-0c69-4000-9c4d-2d82d141905f	None	869662d6-e5ae-4724-82fb-3eb9a6f74e5b	power on	active	False
036cae70-bdee-427c-987c-a6a2d8a32292	None	ab811350-d49e-4e3e-bad7-9af6e61d10a	power on	active	False
8570c96e-f9cd-44ff-a1d8-0252bc405c24	None	7d9caa7c-4e53-4832-983c-44706def4d42	power on	active	False
af46cd81-c78e-47c5-94e3-44d9d669410c	None	211bb71c-2e72-44b1-a867-5f7babf4d4bb	power on	active	False
19260dbb-29a9-4810-b39d-85cc6e1d886f	None	b4a04e95-6624-4f13-8a14-5ef1e6742b94	power on	active	False
d4dae332-4595-43be-9b63-5a64331ea33b	None	db50a27a-4699-4fd9-9687-ec5403db3409	power on	active	False
179bef6e-2510-4311-ad9f-4880454fdaff	None	47e64b48-f105-49f8-ae40-f04db9c39313	power on	active	False
ff0dadfe-e2f3-408f-b69d-01398bb9699d	None	0a1f7293-d9ee-423c-80bc-96125d29d924	power on	active	False
b59f57e3-d5e1-499a-80c1-aac0c78c9534	None	6be9ea47-39c3-4b1e-b755-a866e47h8398	power on	active	False
1132c423-7449-40b5-935a-0f989f61813f	None	19659ace-53b1-4a86-b0b8-21439aa8ab1a	power on	active	False

```
[stack@osp7-director ~]$ nova list
```

ID	Name	Status	Task State	Power State	Networks
47e64b48-f105-49f8-ae40-f04db9c39313	overcloud-cephstorage-0	ACTIVE	-	Running	ct plane=10.22.110.52
0a1f7293-d9ee-423c-80bc-96125d29d924	overcloud-cephstorage-1	ACTIVE	-	Running	ct plane=10.22.110.80
6be9ea47-39c3-4b1e-b755-a866e47h8398	overcloud-cephstorage-2	ACTIVE	-	Running	ct plane=10.22.110.76
19659ace-53b1-4a86-b0b8-21439aa8ab1a	overcloud-cephstorage-3	ACTIVE	-	Running	ct plane=10.22.110.63
ab811350-d49e-4e3e-bad7-9af6e61d10a	overcloud-compute-0	ACTIVE	-	Running	ct plane=10.22.110.53
211bb71c-2e72-44b1-a867-5f7babf4d4bb	overcloud-compute-1	ACTIVE	-	Running	ct plane=10.22.110.61
db50a27a-4699-4fd9-9687-ec5403db3409	overcloud-compute-2	ACTIVE	-	Running	ct plane=10.22.110.58
b4a04e95-6624-4f13-8a14-5ef1e6742b94	overcloud-compute-3	ACTIVE	-	Running	ct plane=10.22.110.62
7d9caa7c-4e53-4832-983c-44706def4d42	overcloud-compute-4	ACTIVE	-	Running	ct plane=10.22.110.56
869662d6-e5ae-4724-82fb-3eb9a6f74e5b	overcloud-compute-5	ACTIVE	-	Running	ct plane=10.22.110.55
e902ad92-f600-41ec-a525-32218be1ee11	overcloud-controller-0	ACTIVE	-	Running	ct plane=10.22.110.59
a23af643-51c8-4f59-881c-77a9d5e1557f	overcloud-controller-1	ACTIVE	-	Running	ct plane=10.22.110.54
f01d22ef-18a4-4f01-b4c1-1ffb6f1d9262	overcloud-controller-2	ACTIVE	-	Running	ct plane=10.22.110.57

2. Check status of Ceph cluster.

3. Log into the new storage node.

```
ssh -l heat-admin 10.22.110.63
sudo -i
```

```
[root@overcloud-cephstorage-3 ceph]# ceph osd tree
ID WEIGHT TYPE NAME UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 171.83984 root default
-2 42.95996 host overcloud-cephstorage-0
  1 5.37000 osd.1 up 1.00000 1.00000
  4 5.37000 osd.4 up 1.00000 1.00000
  7 5.37000 osd.7 up 1.00000 1.00000
 10 5.37000 osd.10 up 1.00000 1.00000
 13 5.37000 osd.13 up 1.00000 1.00000
 16 5.37000 osd.16 up 1.00000 1.00000
 19 5.37000 osd.19 up 1.00000 1.00000
 22 5.37000 osd.22 up 1.00000 1.00000
-3 42.95996 host overcloud-cephstorage-2
  0 5.37000 osd.0 up 1.00000 1.00000
  3 5.37000 osd.3 up 1.00000 1.00000
  6 5.37000 osd.6 up 1.00000 1.00000
  9 5.37000 osd.9 up 1.00000 1.00000
 12 5.37000 osd.12 up 1.00000 1.00000
 15 5.37000 osd.15 up 1.00000 1.00000
 18 5.37000 osd.18 up 1.00000 1.00000
 21 5.37000 osd.21 up 1.00000 1.00000
-4 42.95996 host overcloud-cephstorage-1
  2 5.37000 osd.2 up 1.00000 1.00000
  5 5.37000 osd.5 up 1.00000 1.00000
  8 5.37000 osd.8 up 1.00000 1.00000
 11 5.37000 osd.11 up 1.00000 1.00000
 14 5.37000 osd.14 up 1.00000 1.00000
 17 5.37000 osd.17 up 1.00000 1.00000
 20 5.37000 osd.20 up 1.00000 1.00000
 23 5.37000 osd.23 up 1.00000 1.00000
-5 42.95996 host overcloud-cephstorage-3
 24 5.37000 osd.24 down 0 1.00000
 25 5.37000 osd.25 down 0 1.00000
 26 5.37000 osd.26 down 0 1.00000
 27 5.37000 osd.27 down 0 1.00000
 28 5.37000 osd.28 down 0 1.00000
 29 5.37000 osd.29 down 0 1.00000
 30 5.37000 osd.30 down 0 1.00000
 31 5.37000 osd.31 down 0 1.00000
```

```
[root@overcloud-cephstorage-3 ceph]# ceph -s
cluster 3bd648c2-c09f-11e5-8fff-0025b522225f
health HEALTH_OK
monmap e1: 3 mons at {overcloud-controller-0=10.22.120.51:6789/0,
overcloud-controller-1=10.22.120.57:6789/0,overcloud-controller-2=10.22.120.53:6789/0}
election epoch 40, quorum 0,1,2 overcloud-controller-0,overcloud-controller-2,overcloud-controller-1
osdmap e117: 32 osds: 24 up, 24 in
pgmap v57966: 1024 pgs, 4 pools, 4928 MB data, 1195 objects
15805 MB used, 128 TB / 128 TB avail
1024 active+clean
```

There is a need to activate Ceph disks the way it was done earlier. Click [here for details](#). Update ceph.conf with `osd_journal_size=20000` and activate the disks.

```
[root@overcloud-cephstorage-3 ~]# ceph -s
cluster 3bd648c2-c09f-11e5-8fff-0025b522225f
health HEALTH_OK
monmap e1: 3 mons at {overcloud-controller-0=10.22.120.51:6789/0,
overcloud-controller-1=10.22.120.57:6789/0,overcloud-controller-2=10.22.120.53:6789/0}
election epoch 40, quorum 0,1,2 overcloud-controller-0,overcloud-controller-2,overcloud-controller-1
osdmap e128: 32 osds: 32 up, 32 in
pgmap v58080: 1024 pgs, 4 pools, 4928 MB data, 1195 objects
16978 MB used, 171 TB / 171 TB avail
1024 active+clean
recovery io 449 MB/s, 107 objects/s
```

Ceph osd tree shows all the OSD's on the new node are up now and the storage is scaled up to 171 TB.

```
-5 42.95996 host overcloud-cephstorage-3
 24 5.37000 osd.24 up 1.00000 1.00000
 25 5.37000 osd.25 up 1.00000 1.00000
 26 5.37000 osd.26 up 1.00000 1.00000
 27 5.37000 osd.27 up 1.00000 1.00000
 28 5.37000 osd.28 up 1.00000 1.00000
 29 5.37000 osd.29 up 1.00000 1.00000
```

```

30 5.37000    osd.30          up 1.00000    1.00000
31 5.37000    osd.31          up 1.00000    1.00000
    
```

You may update the default PG's if needed. In the current setup we have 1024 placement groups, while for 32 OSD's it should have been  $100 \times 32 / 3 \sim 1066$ . Therefore, this is ignored.

This completes the addition of storage node in the cluster.

## Scale Up Compute Nodes

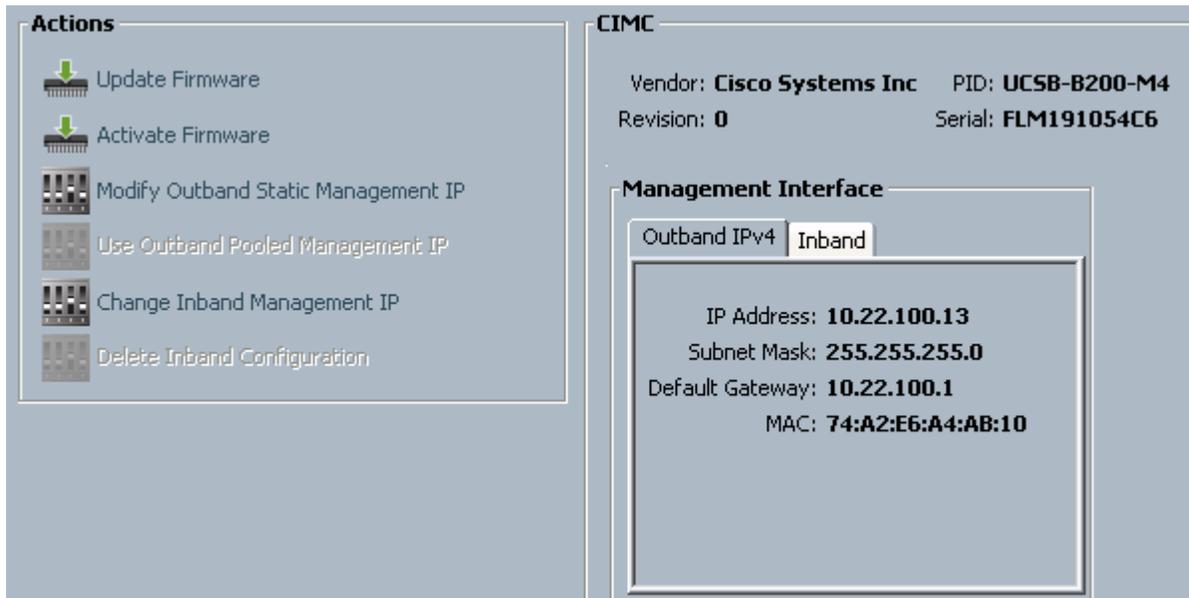
### Provision the New Blade in UCS

Insert the new Cisco UCS B200 M4 blade into an empty slot in the chassis with similar configuration of local disks.

1. [Refer to this section](#) above for creating service profiles from Storage template. Create a new service profile from the template. Unbind the template and remove the storage policy that was attached to it earlier and associate the service profile to the server.
2. Upgrade firmware if needed.
3. Check the installed firmware on the new node and make sure that it is upgraded to the same version as other compute nodes.

Chassis 1		Cisco UCS 5108					
IO Modules							
Servers							
Server 1 (Compute_Node5)		Cisco UCS B200 M4					
Adapters							
BIOS	Cisco UCS B200 M4	B200M4.2.2.4a.0.041620151912	B200M4.2.2.4a.0.041620151912	B200M4.2.2.6c.0.111720151647	Ready	Ready	
Board Controller	Cisco UCS B200 M4	10.0	10.0	N/A	N/A	Ready	
CIMC Controller	Cisco UCS B200 M4	2.2(5a)	2.2(5a)	2.2(6e)	Ready	Ready	
Server 2 (Compute_Node3)		Cisco UCS B200 M4					
Adapters							
BIOS	Cisco UCS B200 M4	B200M4.2.2.4a.0.041620151912	B200M4.2.2.4a.0.041620151912	B200M4.2.2.6c.0.111720151647	Ready	Ready	
Board Controller	Cisco UCS B200 M4	10.0	10.0	N/A	N/A	Ready	
CIMC Controller	Cisco UCS B200 M4	2.2(5a)	2.2(5a)	2.2(6e)	Ready	Ready	
Server 3 (Compute_Node1)		Cisco UCS B200 M4					
Adapters							
BIOS	Cisco UCS B200 M4	B200M4.2.2.4a.0.041620151912	B200M4.2.2.4a.0.041620151912	B200M4.2.2.6c.0.111720151647	Ready	Ready	
Board Controller	Cisco UCS B200 M4	10.0	10.0	N/A	N/A	Ready	
CIMC Controller	Cisco UCS B200 M4	2.2(5a)	2.2(5a)	2.2(6e)	Ready	Ready	

4. Get the hardware inventory details needed for introspection. This include IPMI address, Provisioning MAC address, Boot Lun size, CPU and memory.



Name	vNIC	Vendor	PID	Model	Operability	MAC
NIC 1	PXE_vNIC	Cisco Systems Inc	UCSB-MLOM-40G-03	Cisco UCS VIC 1340	Operable	00:25:B5:00:00:1A
NIC 2	Mgmt_vNIC	Cisco Systems Inc	UCSB-MLOM-40G-03	Cisco UCS VIC 1340	Operable	00:25:B5:00:00:0A
NIC 3	External_vNIC	Cisco Systems Inc	UCSB-MLOM-40G-03	Cisco UCS VIC 1340	Operable	00:25:B5:00:00:3A

- Specify the NIC order in the service profile. This should be the same as the other Compute nodes with provisioning interface as the first one.
- Check the boot policy of the server. Validate that this is same as other compute nodes too. Should be LAN PXE first followed by local lun.
- Update cisco-plugins.yaml file with details about this new server.

Append an entry in UCSM Host list as below

```
NetworkUCSMHostList: ..... , '00:25:b5:00:00:06:Openstack_Compute_Node7
Add entries for this compute host in each switch section
"00:25:b5:00:00:06": {
  "ports": "port-channel:17,port-channel:18"
},
```

- With the above the server is ready for introspection and Overcloud deploy.

## Run Introspection

To run Introspection, complete the following steps:

- Prepare json file for introspection:

```
stack@osp7-director ~]$ cat compute-node.json
{
  "nodes": [
    {
      "pm_user": "admin",
      "pm_password": "password",
      "pm_type": "pxe_ipmitool",
      "pm_addr": "10.22.100.13",
      "mac": [
```

```

        "00:25:b5:00:00:1a"
    ],
    "memory": "262144",
    "disk": "200",
    "arch": "x86_64",
    "cpu": "40"
}
]
}

```

## 2. Check IPMI Connectivity:

```

ipmitool -I lanplus -H 10.22.100.13 -U admin -P <password> chassis power off
Chassis Power Control: Down/Off

```

## 3. Run discovery and introspection:

```

[stack@osp7-director ~]$ openstack baremetal import --json ~/compute-node.json
[stack@osp7-director ~]$ openstack baremetal configure boot
[stack@osp7-director ~]$ openstack baremetal list
[stack@osp7-director ~]$ ironic node-set-maintenance 036cae70-bdee-427c-987c-
a6a2d8a32292 true
[stack@osp7-director ~]$ openstack baremetal introspection start 036cae70-bdee-
427c-987c-a6a2d8a32292
[stack@osp7-director ~]$ openstack baremetal introspection status 036cae70-bdee-
427c-987c-a6a2d8a32292

```

Wait till the introspection is complete. The status command should yield finished as True and Error as none.

```

ironic node-set-maintenance 036cae70-bdee-427c-987c-a6a2d8a32292 false

```

## 4. Update node properties

```

[stack@osp7-director ~]$ ironic node-update \
036cae70-bdee-427c-987c-a6a2d8a32292 \
> add properties/capabilities='profile:compute,boot_option:local'

[stack@osp7-director ~]$ glance image-list | grep deploy | awk '{print $2" " $4}'
ca9667c9-de8f-4df1-95d5-b5f8b6fb8f4d bm-deploy-kernel
404d0e44-e5c7-4b46-8339-c451441b3f55 bm-deploy-ramdisk

[stack@osp7-director ~]$ ironic node-update 036cae70-bdee-427c-987c-a6a2d8a32292 \
> add driver_info/deploy_kernel='ca9667c9-de8f-4df1-95d5-b5f8b6fb8f4d';

[stack@osp7-director ~]$ ironic node-update 036cae70-bdee-427c-987c-a6a2d8a32292 \
> add driver_info/deploy_ramdisk='404d0e44-e5c7-4b46-8339-c451441b3f55';

```

Check the status of added entries with `ironic node-show`. Repeat the above to all nodes that you would like to add as overcloud deploy can add all of these in a single go.

## Run Overcloud Deploy

Run Overcloud deployment command. The number of compute nodes has been incremented to 7 from 6 earlier. Here the number '7' indicates the total number of storage nodes in Overcloud.

```
#!/bin/bash
openstack overcloud deploy --templates --compute-scale 7 \
-e /usr/share/openstack-tripleo-heat-templates/overcloud-resource-registry-
puppet.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-
isolation.yaml \
-e /home/stack/templates/network-environment.yaml \
-e /home/stack/templates/storage-environment.yaml \
-e /home/stack/templates/cisco-plugins.yaml \
-e /home/stack/templates/post_config.yaml \
--verbose --debug --log-file overcloud_compute-add.log
```

```
Addition of nodes complete with the following message
INFO: rdomanager_oscplugin.v1.overcloud_deploy.DeployOvercloud Stack found, will
be doing a stack update
Overcloud Endpoint: http://172.22.215.91:5000/v2.0/
Overcloud Deployed
DEBUG: openstackclient.shell clean_up DeployOvercloud
```

## Post Deployment and Health Checks

To perform the deployment and health checks, complete the following steps:

1. Login to each controller node and check for the existence of the new compute node in `/etc/neutron/plugin.ini`. If not please add in each Nexus Switch section and also in UCSM host list in `plugin.ini` file. Make sure to make the changes across all the controller nodes.

```
systemctl daemon-reload
systemctl restart neutron-server.service
```

2. Restart nova-services as a post deployment.

```
systemctl restart openstack-nova-consoleauth.service openstack-nova-
scheduler.service openstack-nova-api.service
```

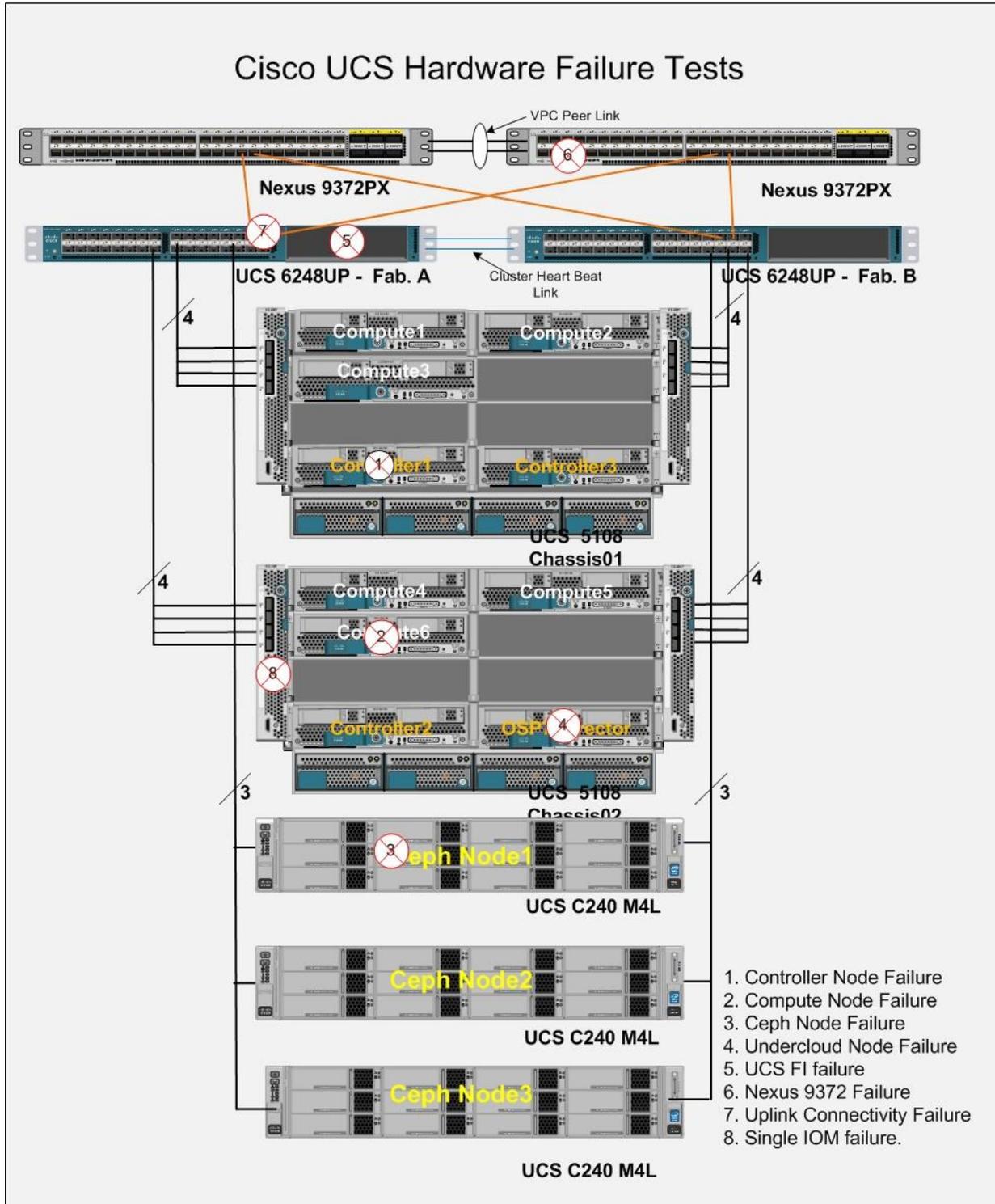
3. Check the status through `ironic node-list` and `nova list`.
4. Log into dashboard to check the status of the new node added.

The System should be up and running and will **deploy VM's on the newly added node**.

## High Availability

---

Both the hardware and software stack are injected with faults to trigger a failure of a running process on a node or an unavailability of hardware for a short or extended period of time. With the fault in place the functional validations are done as mentioned above. The purpose is to achieve business continuity without interruption to the clients. However performance degradation is inevitable and has been documented wherever it was captured as part of the tests.



## High Availability of Software Stack

### OpenStack Services

The status of OpenStack services were checked with pcs status as below on Controller Node:

```
[root@overcloud-controller-0 ~]# pcs status
```

```

Cluster name: tripleo_cluster
Last updated: Thu Feb 11 13:08:58 2016           Last change: Wed Feb 10 13:05:34
2016 by root via cibadmin on overcloud-controller-2
Stack: corosync
Current DC: overcloud-controller-2 (version 1.1.13-10.e17-44eb2dd) - partition
with quorum
3 nodes and 115 resources configured

```

```
Online: [ overcloud-controller-0 overcloud-controller-1 overcloud-controller-2 ]
```

```
Full list of resources:
```

```

ip-172.22.215.91      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
Clone Set: haproxy-clone [haproxy]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
ip-10.22.100.50      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-1
ip-10.22.100.51      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-2
ip-10.22.150.50      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
Master/Slave Set: galera-master [galera]
Masters: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
ip-10.22.110.65      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-1
Master/Slave Set: redis-master [redis]
Masters: [ overcloud-controller-2 ]
Slaves: [ overcloud-controller-0 overcloud-controller-1 ]
Clone Set: mongod-clone [mongod]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
Clone Set: rabbitmq-clone [rabbitmq]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
Clone Set: memcached-clone [memcached]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
ip-10.22.120.50      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-2
Clone Set: openstack-nova-scheduler-clone [openstack-nova-scheduler]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
Clone Set: neutron-l3-agent-clone [neutron-l3-agent]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
Clone Set: openstack-ceilometer-alarm-notifier-clone [openstack-ceilometer-
alarm-notifier]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
Clone Set: openstack-heat-engine-clone [openstack-heat-engine]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
Clone Set: openstack-ceilometer-api-clone [openstack-ceilometer-api]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
Clone Set: neutron-metadata-agent-clone [neutron-metadata-agent]

```

```

    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: neutron-ovs-cleanup-clone [neutron-ovs-cleanup]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: neutron-netns-cleanup-clone [neutron-netns-cleanup]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-heat-api-clone [openstack-heat-api]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-cinder-scheduler-clone [openstack-cinder-scheduler]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-nova-api-clone [openstack-nova-api]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-heat-api-cloudwatch-clone [openstack-heat-api-cloudwatch]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-ceilometer-collector-clone [openstack-ceilometer-collector]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-keystone-clone [openstack-keystone]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-nova-consoleauth-clone [openstack-nova-consoleauth]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-glance-registry-clone [openstack-glance-registry]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-ceilometer-notification-clone [openstack-ceilometer-
notification]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-cinder-api-clone [openstack-cinder-api]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: neutron-dhcp-agent-clone [neutron-dhcp-agent]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-glance-api-clone [openstack-glance-api]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: neutron-openvswitch-agent-clone [neutron-openvswitch-agent]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: openstack-nova-novncproxy-clone [openstack-nova-novncproxy]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    Clone Set: delay-clone [delay]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
    vsm-s (ocf::heartbeat:VirtualDomain): Started overcloud-controller-0
    vsm-p (ocf::heartbeat:VirtualDomain): Started overcloud-controller-1
    Clone Set: neutron-server-clone [neutron-server]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]

```

```

Clone Set: httpd-clone [httpd]
  Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
Clone Set: openstack-ceilometer-central-clone [openstack-ceilometer-central]
  Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
Clone Set: openstack-ceilometer-alarm-evaluator-clone [openstack-ceilometer-
alarm-evaluator]
  Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
Clone Set: openstack-heat-api-cfn-clone [openstack-heat-api-cfn]
  Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
openstack-cinder-volume      (systemd:openstack-cinder-volume):      Started
overcloud-controller-0
Clone Set: openstack-nova-conductor-clone [openstack-nova-conductor]
  Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
ucs-fence-controller      (stonith:fence_cisco_ucs):      Started overcloud-
controller-2

PCSD Status:
  overcloud-controller-0: Online
  overcloud-controller-1: Online
  overcloud-controller-2: Online

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled

```

Few identified services running on these nodes were either restarted or killed and/or rebooted the nodes.

For eg.

```

Master/Slave Set: redis-master [redis]
  Masters: [ overcloud-controller-2 ]
  Slaves: [ overcloud-controller-0 overcloud-controller-1 ]

```

Per above redis master is overcloud-controller-2. This node was rebooted and observed the behavior while the node getting rebooted and any impact of N-S traffic or E-W traffic of VM's. **The only issue observed was for about 2-3 minutes few of the VM's were not pingable because of bug [1281603](#)** and this was not related with the services above.

The ceph node monitors and services were also restarted to test any interruption of volume creation and **booting of the VM's, but no issues observed.**

## High Availability of Hardware Stack

### HA of Fabric Interconnects

#### FI Reboot Tests

Cisco UCS Fabric Interconnects work in pair with inbuilt HA. While both of them serve traffic during a normal operation, a surviving member can still keep the system up and running. Depending on the overprovisioning used in the deployment a degradation in performance may be expected.

An effort is made to reboot the Fabric one after the other and do [functional tests](#) as mentioned earlier.

#### Reboot Fabric A

- Check the status of the UCS Fabric Cluster before reboot

```
UCSO-6248-FAB-A# show cluster state
Cluster Id: 0x1992ea1a116111e5-0x8ace002a6a3bbba1
```

```
A: UP, PRIMARY
B: UP, SUBORDINATE
```

**HA READY ←--System should be in HA ready before invoking any of the HA tests on Fabrics.**

- Check the status of OpenStack PCS Cluster before reboot

```
root@overcloud-controller-0 ~]# pcs status
Cluster name: tripleo_cluster
```

```
Online: [ overcloud-controller-0 overcloud-controller-1 overcloud-controller-2
]
```

Full list of resources:

```
ip-172.22.215.91      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
  Clone Set: haproxy-clone [haproxy]
    Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
  ip-10.22.100.51      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-2
  ip-10.22.100.50      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-2
  ip-10.22.150.50      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
  Master/Slave Set: galera-master [galera]
    Masters: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
  ip-10.22.110.45      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-2
```

```
.....
.....
```

Grep for any errors or stopped actions from PCS, fix the issues before starting the tests.

- Reboot Fabric A ( primary )

Log into UCS Fabric Command Line Interface and reboot the Fabric

```
UCSO-6248-FAB-A# connect local-mgmt
Cisco Nexus Operating System (NX-OS) Software
```

```
UCSO-6248-FAB-A(local-mgmt)# reboot
Before rebooting, please take a configuration backup.
Do you still want to reboot? (yes/no):yes
nohup: ignoring input and appending output to `nohup.out'
```

```
Broadcast message from root (Fri Nov 6 20:59:45 2015):
```

```
All shells being terminated due to system /sbin/reboot
Connection to 10.22.100.6 closed.
```

### Health Checks and Observations

The following is a list of health checks and observations:

- Check for VIP and Fabric A pings. Both should be down immediately. VIP recovers after a couple of minutes
- Check for PCS Cluster status on one of the controller nodes. System could be slow in the beginning but should respond as follows:

PCSD Status:

```
overcloud-controller-0: Online
overcloud-controller-1: Online
overcloud-controller-2: Online
```

Perform a quick health check on creating VM's, checking the status of existing instances and l3 forwarding enabled in N1KV earlier. Check the sanity checks on Nexus switches too for any impact on port-channels because of Fab A is down.

- Create Virtual Machines

Perform a quick health check on creating VM's, checking the status of existing instances and l3 forwarding enabled in N1KV earlier. Check the sanity checks on Nexus switches too for any impact on port-channels because of Fab A is down.



Fabric A might take around 15 minutes to come back online.

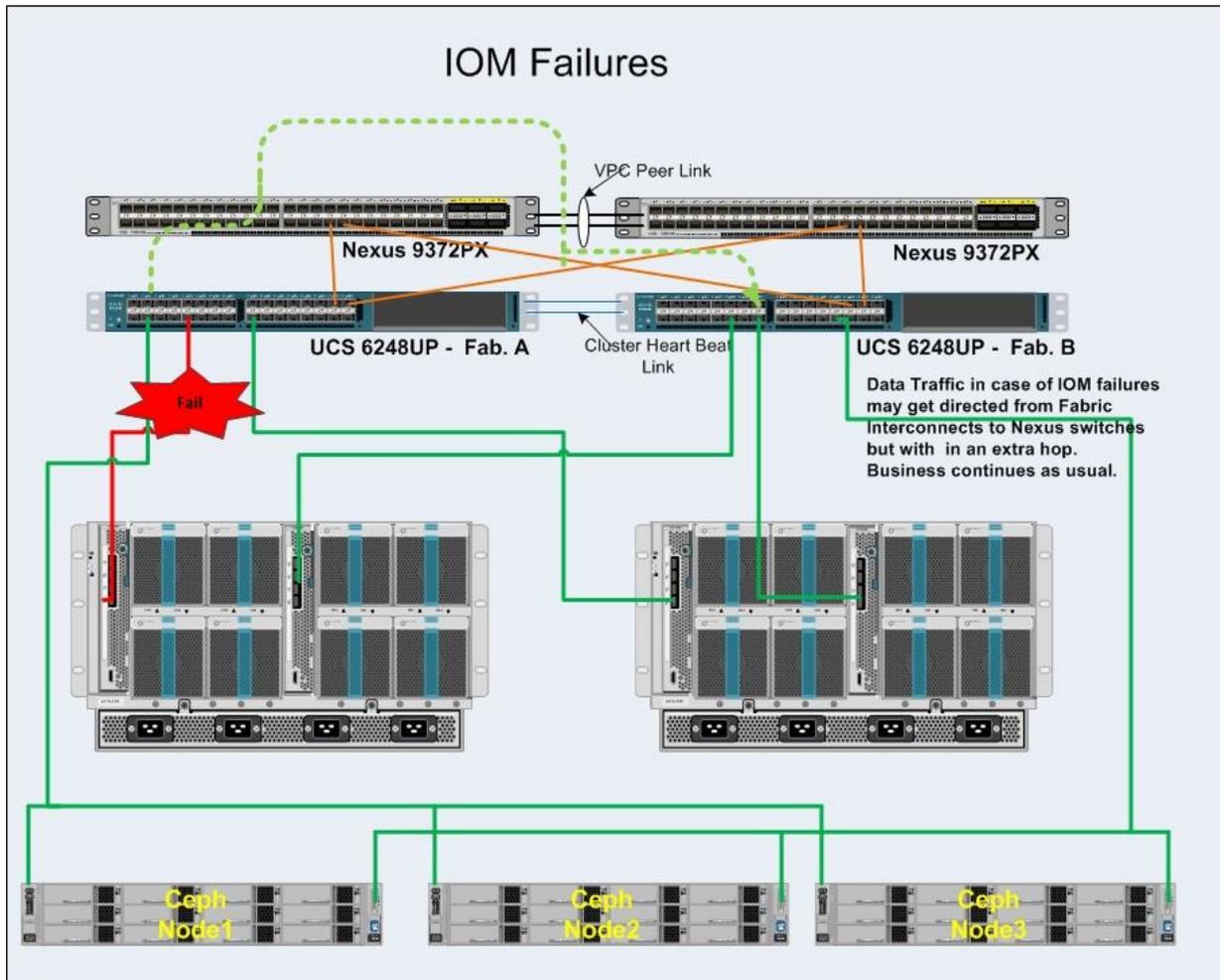
---

### Reboot Fabric B

- Connect to the Fabric B now and check the cluster status. System should show HA READY before rebooting Fab B.
- Reboot Fab B by connecting to the local-mgmt similar to Fab A.
- Perform the health check similar to the ones does for Fab A.
- The test went fine without any issues on the configuration. Please refer bug [1267780](#) on the issues encountered and the fix rolled out in this document.

### Hardware Failures of IO Modules

IO Module Failures seldom happen in UCS infrastructure and in most of the cases these are human mistakes. The failure tests were included just to validate the business continuity. Any L3 east-west traffic will get routed through upstream switches in case of IOM failures.



Multiple Tenants with multiple networks and Virtual machines were created. Identified the VM's belonging to the same tenant but with different networks and also going to different chassis. One of the IO Modules was pulled out from the chassis and the L3 traffic validated.

Health Checks before Fault Injection

Ping from tenant320\_120\_inst8 to tenant320\_170\_inst19 10.2.170.11 and 10.22.160.49

```

root@tenant320-120-inst8 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1400
    inet 10.2.120.10 netmask 255.255.255.0 broadcast 10.2.120.255
    inet6 fe80::f816:3eff:febc:3a2a prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:bc:3a:2a txqueuelen 1000 (Ethernet)
    RX packets 244 bytes 26277 (25.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 344 bytes 32511 (31.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 0 (Local Loopback)
    RX packets 10 bytes 756 (756.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 756 (756.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@tenant320-120-inst8 ~]# ping 10.2.170.11
PING 10.2.170.11 (10.2.170.11) 56(84) bytes of data.
64 bytes from 10.2.170.11: icmp_seq=1 ttl=63 time=0.306 ms
64 bytes from 10.2.170.11: icmp_seq=2 ttl=63 time=0.213 ms
64 bytes from 10.2.170.11: icmp_seq=3 ttl=63 time=0.198 ms
^C
--- 10.2.170.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.198/0.239/0.306/0.047 ms
root@tenant320-120-inst8 ~]# ping 10.22.160.49
PING 10.22.160.49 (10.22.160.49) 56(84) bytes of data.
64 bytes from 10.22.160.49: icmp_seq=1 ttl=63 time=0.194 ms
64 bytes from 10.22.160.49: icmp_seq=2 ttl=63 time=0.165 ms
64 bytes from 10.22.160.49: icmp_seq=3 ttl=63 time=0.209 ms
^C
--- 10.22.160.49 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.165/0.189/0.209/0.021 ms
root@tenant320-120-inst8 ~]#
    
```

Fault Injection and Health Checks

Remove IO Module from Chassis 1 going to Fabric A

Name	Chassis ID	PID	Model	User Label	Cores	Cores Er
Server 1 (Compute-5)	1	UCSB-B200-M4	Cisco UCS B200 M4	Compute-5	20	20
Server 2 (Compute-3)	1	UCSB-B200-M4	Cisco UCS B200 M4	Compute-3	20	20
Server 3 (Compute-1)	1	UCSB-B200-M4	Cisco UCS B200 M4	Compute-1	20	20
Server 4 (Compute-16)	1	UCSB-B200-M4	Cisco UCS B200 M4	Compute-16	16	16
Server 5 (Compute-17)	1	UCSB-B200-M4	Cisco UCS B200 M4	Compute-17	24	24
Server 6 (Compute-9)	1	UCSB-B200-M4	Cisco UCS B200 M4	Compute-9	16	16
Server 7 (Compute-19)	1	UCSB-B200-M4	Cisco UCS B200 M4	Compute-19	20	20
Server 8 (Controller_Node1)	1	UCSB-B200-M4	Cisco UCS B200 M4	Controller_Node1	20	20
Server 1 (Compute-6)	2	UCSB-B200-M4	Cisco UCS B200 M4	Compute-6	20	20
Server 2 (Compute-4)	2	UCSB-B200-M4	Cisco UCS B200 M4	Compute-4	20	20
Server 3 (Compute-2)	2	UCSB-B200-M4	Cisco UCS B200 M4	Compute-2	20	20
Server 4 (Compute-18)	2	UCSB-B200-M4	Cisco UCS B200 M4	Compute-18	16	16
Server 5 (Compute-20 (not...))	2	UCSB-B200-M4	Cisco UCS B200 M4	Compute-20 (not i...	20	20
Server 6 (Controller-2)	2	UCSB-B200-M4	Cisco UCS B200 M4	Controller-2	12	12
Server 7 (Controller-3)	2	UCSB-B200-M4	Cisco UCS B200 M4	Controller-3	32	32
Server 8 (Installer Node)	2	UCSB-B200-M4	Cisco UCS B200 M4	Installer Node	16	16
Server 1 (Compute-7)	3	UCSB-B200-M4	Cisco UCS B200 M4	Compute-7	16	16
Server 2 (Compute-8)	3	UCSB-B200-M4	Cisco UCS B200 M4	Compute-8	16	16
Server 3 (Compute-10)	3	UCSB-B200-M4	Cisco UCS B200 M4	Compute-10	20	20
Server 4 (Compute-11)	3	UCSB-B200-M4	Cisco UCS B200 M4	Compute-11	20	20
Server 5 (Compute-12)	3	UCSB-B200-M4	Cisco UCS B200 M4	Compute-12	20	20
Server 6 (Compute-13)	3	UCSB-B200-M4	Cisco UCS B200 M4	Compute-13	20	20
Server 7 (Compute-14)	3	UCSB-B200-M4	Cisco UCS B200 M4	Compute-14	20	20
Server 8 (Compute-15)	3	UCSB-B200-M4	Cisco UCS B200 M4	Compute-15	20	20

Post Fault injection, ping continued

```
[root@tenant320-120-inst8 ~]# ping 10.2.170.11
PING 10.2.170.11 (10.2.170.11) 56(84) bytes of data.
64 bytes from 10.2.170.11: icmp_seq=12 ttl=63 time=0.507 ms
64 bytes from 10.2.170.11: icmp_seq=13 ttl=63 time=0.214 ms
64 bytes from 10.2.170.11: icmp_seq=14 ttl=63 time=0.218 ms
64 bytes from 10.2.170.11: icmp_seq=15 ttl=63 time=0.196 ms
64 bytes from 10.2.170.11: icmp_seq=16 ttl=63 time=0.220 ms
^C
--- 10.2.170.11 ping statistics ---
16 packets transmitted, 5 received, 68% packet loss, time 14999ms
rtt min/avg/max/mdev = 0.196/0.271/0.507/0.118 ms
[root@tenant320-120-inst8 ~]# ping 10.22.160.49
PING 10.22.160.49 (10.22.160.49) 56(84) bytes of data.
64 bytes from 10.22.160.49: icmp_seq=1 ttl=63 time=0.360 ms
64 bytes from 10.22.160.49: icmp_seq=2 ttl=63 time=0.429 ms
64 bytes from 10.22.160.49: icmp_seq=3 ttl=63 time=0.281 ms
64 bytes from 10.22.160.49: icmp_seq=4 ttl=63 time=0.297 ms
64 bytes from 10.22.160.49: icmp_seq=5 ttl=63 time=0.208 ms
^C
--- 10.22.160.49 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.208/0.315/0.429/0.074 ms
```

## HA on Nexus Switches

Nexus switches are deployed in pairs and allow the upstream connectivity of the virtual machines to outside of the fabric. Cisco Nexus plugin creates VLANs on these switches both globally and on the port channel. The Nexus plugin replays these vlans or rebuilds the vlan information on the rebooted switch once it comes back up again. In order to test the HA of these switches multiple networks and instances were created and one of the switches were rebooted. The connectivity of the VM's through floating network checked and also the time it took for the plugins to replay was noted as below.

### Test Bed Setup before Injecting Fault

Nexus Switches

```
UCSO-N9K-FAB-A# show version
Software
  BIOS: version 07.17
  NXOS: version 7.0(3)I1(3)
Hardware
  cisco Nexus9000 C9372PX chassis
  Intel(R) Core(TM) i3-3227U C with 16402540 kB of memory.
Last reset
  System version: 7.0(3)I1(3)

UCSO-N9K-FAB-A# show startup-config vlan

!Command: show startup-config vlan
!Time: Thu Jan 28 14:32:13 2016
!Startup config saved at: Tue Jan 26 18:44:34 2016
```

```
version 7.0(3)I1(3)
```

```
Legend:
```

```
    Type 1 : vPC will be suspended in case of mismatch
```

Name	Type	Local Value	Peer Value
Vlan to Vn-segment Map	1	No Relevant Maps	No Relevant Maps
STP Mode	1	Rapid-PVST	Rapid-PVST
STP Disabled	1	None	None
STP MST Region Name	1	""	""
STP MST Region Revision	1	0	0
STP MST Region Instance to	1		
VLAN Mapping			
STP Loopguard	1	Disabled	Disabled
STP Bridge Assurance	1	Enabled	Enabled
STP Port Type, Edge	1	Normal, Disabled,	Normal, Disabled,
BPDUFilter, Edge BPDUGuard		Disabled	Disabled
STP MST Simulate PVST	1	Enabled	Enabled
Interface-vlan admin up	2	100,110,120,130,160,215	100,110,120,130,160,215
Interface-vlan routing capability	2	1,100,110,120,130,160,215	1,100,110,120,130,160,215
Allowed VLANs	-	1,100,110,120,130,160,215,251-263,265-270,27	1,100,110,120,130,160,215,251-263,265-270,27
		-----Output truncated-----	
Local suspended VLANs	-	-	-

```
UCSO-N9K-FAB-A# show vpc brief
```

```
Legend:
```

```
    (*) - local vPC is down, forwarding through vPC peer-link
```

```
vPC domain id           : 100
Peer status             : peer adjacency formed ok
vPC keep-alive status   : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role                : secondary
Number of vPCs configured : 2
Peer Gateway            : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status    : Disabled
Delay-restore status     : Timer is off.(timeout = 30s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
```

```
vPC Peer-link status
```

```
-----
id   Port   Status Active vlans
--   --
1    Po1    up     1,100,110,120,130,160,215,251-263,265-270,272,274,
      276-277,279-293,295-298,300-304,306-308,310-314,31
      -----Output truncated-----
```

```
vPC status
```

```
-----
id   Port   Status Consistency Reason           Active vlans
```

```

--      -----
17   Po17   up      success   success
                                           1,100,110,1
                                           20,130,160,
                                           215,251-263
                                           ,265-270,27
                                           2,274,276-2 ....
18   Po18   up      success   success
                                           1,100,110,1
                                           20,130,160,
                                           215,251-263
                                           ,265-270,27
                                           2,274,276-2 ....

```

```
UCSO-N9K-FAB-A(config)# show vlan | grep q- | count
200
```

```
UCSO-N9K-FAB-B(config)# show vlan | grep q- | count
200
```

Present in both port channels

```
UCSO-N9K-FAB-A(config)# show vlan | grep "Po17, Po18" | grep q- | count
200
```

```
interface port-channel17
```

```
  description Port-channel for UCS FabA port 17 & FabB port 17
```

```
  switchport mode trunk
```

```
  switchport trunk allowed vlan 1,100,110,120,130,160,215,251-263
```

```
  switchport trunk allowed vlan add 265-272,274,276-277,279-293,295-298
```

```
  -----Output truncated-----
```

```
  switchport trunk allowed vlan add 536,538
```

```
  spanning-tree port type edge trunk
```

```
  vpc 17
```

```
interface port-channel18
```

```
  description Port-channel for UCS FabA port 18 & FabB port 18
```

```
  switchport mode trunk
```

```
  switchport trunk allowed vlan 1,100,110,120,130,160,215,251-263
```

```
  switchport trunk allowed vlan add 265-272,274,276-277,279-293,295-298
```

```
  -----Output truncated-----
```

```
  switchport trunk allowed vlan add 536,538
```

```
  spanning-tree port type edge trunk
```

```
  vpc 18
```

```
interface Ethernet1/17
```

```
  description Uplink from UCS FabA Port 17
```

```
  switchport mode trunk
```

```
  switchport trunk allowed vlan 1,100,110,120,130,160,215,251-263
```

```
  switchport trunk allowed vlan add 265-272,274,276-277,279-293,295-298
```

```
  -----Output truncated-----
```

```
  switchport trunk allowed vlan add 536,538
```

```
  spanning-tree port type edge trunk
```

```
  channel-group 17 mode active
```

```
interface Ethernet1/18
```

```
  description Uplink from UCS FabA Port 18
```

```
  switchport mode trunk
```

```
  switchport trunk allowed vlan 1,100,110,120,130,160,215,251-263
```

```
  switchport trunk allowed vlan add 265-272,274,276-277,279-293,295-298
```

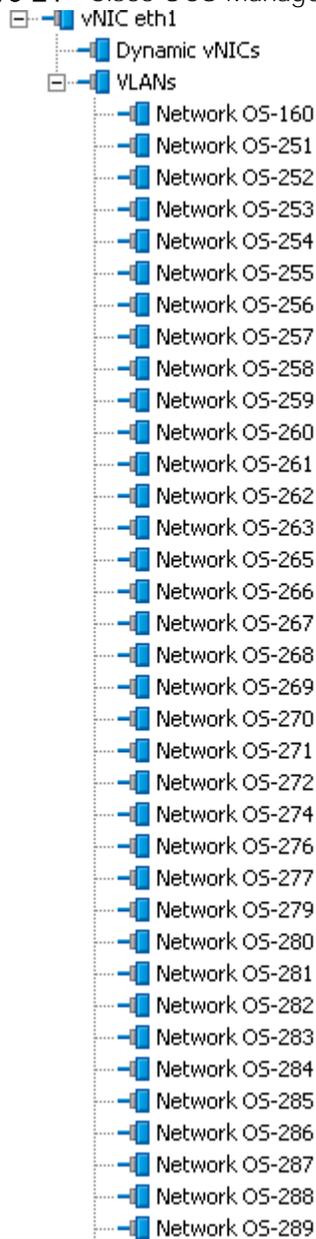
```
  -----Output truncated-----
```

```
  switchport trunk allowed vlan add 536,538
```

```
  spanning-tree port type edge trunk
```

```
  channel-group 18 mode active
```

Figure 21 Cisco UCS Manager VLANs



## Creating Virtual Machines

Creating tenant creates VLANs in compute nodes. However if a VM from one tenant is deleted, the VLAN on the computes will remain until the last vm of that tenant is deleted.

### Tenants

```
[stack@osp7-director scripts]$ nova list --all-tenants | awk '/Running/' | wc -l  
400
```

One Tenant  
Two Networks/Tenant  
Two VM's/Network/Tenant

```
[stack@osp7-director scripts]# nova list --all-tenants
```

ID	Name	Tenant ID	Status	Task State	Power State	Networks
d46d34a0-b0d6-44ba-b798-7efe3187bcf6	tenant201_101_inst1	0374dd0a728a41cc84b5a826756221ca	ACTIVE	-	Running	tenant201-101=10.1.101.5, 10.22.160.92
04727975-6162-4e3a-b490-0f57fa2340e5	tenant201_101_inst2	0374dd0a728a41cc84b5a826756221ca	ACTIVE	-	Running	tenant201-101=10.1.101.6, 10.22.160.93
acd99cac-06e2-4ac1-b412-fc085ddb44d5	tenant201_151_inst3	0374dd0a728a41cc84b5a826756221ca	ACTIVE	-	Running	tenant201-151=10.1.151.5, 10.22.160.94
24ef674a-82aa-492e-8641-d14b5f8f9d9a	tenant201_151_inst4	0374dd0a728a41cc84b5a826756221ca	ACTIVE	-	Running	tenant201-151=10.1.151.6, 10.22.160.95
728257e2-7c4e-4601-b391-961418bf7f1d	tenant202_102_inst1	f3d592f02a944f0eb88c82d5d3c763d0	ACTIVE	-	Running	tenant202-102=10.1.102.5, 10.22.160.97
06bade87-3688-44d8-b5b5-b2ec33b3b0e0	tenant202_102_inst2	f3d592f02a944f0eb88c82d5d3c763d0	ACTIVE	-	Running	tenant202-102=10.1.102.6, 10.22.160.98
4b3f43ca-ac7a-4e76-9fbb-ba70ea22d75b	tenant202_152_inst3	f3d592f02a944f0eb88c82d5d3c763d0	ACTIVE	-	Running	tenant202-152=10.1.152.5, 10.22.160.99
1c4fac60-b678-467a-abfd-2a7d0a1c1d59	tenant202_152_inst4	f3d592f02a944f0eb88c82d5d3c763d0	ACTIVE	-	Running	tenant202-152=10.1.152.6, 10.22.160.100
01e6da62-3d32-4f5d-9bc8-9667968363e0	tenant203_103_inst1	836a435b200d4c4e888ca8568374e54f	ACTIVE	-	Running	tenant203-103=10.1.103.5, 10.22.160.102
bdc0d26b-e2e9-47cb-afdf-c7ca0035ff64	tenant203_103_inst2	836a435b200d4c4e888ca8568374e54f	ACTIVE	-	Running	tenant203-103=10.1.103.6, 10.22.160.103
f73f007b-10b0-4e3f-9a93-cc33ab76bc6d	tenant203_153_inst3	836a435b200d4c4e888ca8568374e54f	ACTIVE	-	Running	tenant203-153=10.1.153.5, 10.22.160.104
ac79aa49-3c39-48a8-9736-fdcb9029da31	tenant203_153_inst4	836a435b200d4c4e888ca8568374e54f	ACTIVE	-	Running	tenant203-153=10.1.153.6, 10.22.160.105
b0015671-e9fe-40c9-a5e9-1d7fc8205e15	tenant204_104_inst1	d914bbf9f2cb40c7982e23b2cc5c2210	ACTIVE	-	Running	tenant204-104=10.1.104.5, 10.22.160.107
b0007d7f-63bb-45a1-b7cf-256d8d9a1a57	tenant204_104_inst2	d914bbf9f2cb40c7982e23b2cc5c2210	ACTIVE	-	Running	tenant204-104=10.1.104.6, 10.22.160.108
f82867a6-5d50-4ff5-aad1-ad46ac082e0d	tenant204_154_inst3	d914bbf9f2cb40c7982e23b2cc5c2210	ACTIVE	-	Running	tenant204-154=10.1.154.5, 10.22.160.109
d34ec44d-5d3b-4526-8a09-cc60a912f916	tenant204_154_inst4	d914bbf9f2cb40c7982e23b2cc5c2210	ACTIVE	-	Running	tenant204-154=10.1.154.6, 10.22.160.110

Connectivity Tests

Connectivity from external client machine on floating IP to VM's.

Command used:

```
ssh -i tenant349kp.pem -o StrictHostKeyChecking=no cloud-user@10.22.162.77 /tmp/run.sh - for each VM created.
```

```
Host is tenant208-108-inst1 and Network is          inet 10.1.108.5 netmask
255.255.255.0 broadcast 10.1.108.255
Host is tenant208-108-inst2 and Network is          inet 10.1.108.6 netmask
255.255.255.0 broadcast 10.1.108.255
.....
.....
.....
Host is tenant348-148-inst1 and Network is          inet 10.2.148.5 netmask
255.255.255.0 broadcast 10.2.148.255
Host is tenant348-148-inst2 and Network is          inet 10.2.148.6 netmask
255.255.255.0 broadcast 10.2.148.255
Host is tenant348-198-inst3 and Network is          inet 10.2.198.5 netmask
255.255.255.0 broadcast 10.2.198.255
```

A script was created and pushed with 'scp' that in turn runs ifconfig on each VM and gets the details. This was validated for all the 400 VM's created above.

Login to instances:

```
[root@clouduser scripts]# ssh -i tenant350kp.pem -o StrictHostKeyChecking=no
cloud-user@10.22.162.79
[cloud-user@tenant350-150-inst1 ~]$
[cloud-user@tenant350-150-inst1 ~]$ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1400
inet 10.2.150.5 netmask 255.255.255.0 broadcast 10.2.150.255
inet6 fe80::f816:3eff:fe57:d162 prefixlen 64 scopeid 0x20<link>
ether fa:16:3e:57:d1:62 txqueuelen 1000 (Ethernet)
RX packets 1746 bytes 169967 (165.9 KiB)
RX errors 0 dropped 5 overruns 0 frame 0
TX packets 1748 bytes 161366 (157.5 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Ping VM in the same network:

```
[cloud-user@tenant350-150-inst1 ~]$ ping 10.2.150.6
PING 10.2.150.6 (10.2.150.6) 56(84) bytes of data.
64 bytes from 10.2.150.6: icmp_seq=1 ttl=64 time=1.62 ms
64 bytes from 10.2.150.6: icmp_seq=2 ttl=64 time=0.609 ms
64 bytes from 10.2.150.6: icmp_seq=3 ttl=64 time=0.554 ms
```

## L3 routing:

```
[stack@osp7-director scripts]$ source keystonerc_tenant350
[stack@osp7-director scripts]$ neutron subnet-list | grep -v ext-subnet
```

id	name	cidr	allocation_pools
29f59ee2-bf23-4f0a-bb4c-b6b600bb98ae	tenant350-150-subnet	10.2.150.0/24	{"start": "10.2.150.2", "end": "10.2.150.254"}
ced01789-6f4e-4852-964e-d24c36892544	tenant350-200-subnet	10.2.200.0/24	{"start": "10.2.200.2", "end": "10.2.200.254"}

10.2.150 and 10.2.200 are the networks for tenant 350. Ping IP's in these networks.

```
[cloud-user@tenant350-150-inst1 ~]$ ping 10.2.150.5
PING 10.2.150.5 (10.2.150.5) 56(84) bytes of data.
64 bytes from 10.2.150.5: icmp_seq=1 ttl=64 time=0.046 ms
64 bytes from 10.2.150.5: icmp_seq=2 ttl=64 time=0.047 ms
Connectivity verified with password less authentication too.
```

```
--- 10.2.150.5 ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.046/0.046/0.047/0.006 ms
```

```
[cloud-user@tenant350-150-inst1 ~]$ ping 10.2.150.6
PING 10.2.150.6 (10.2.150.6) 56(84) bytes of data.
64 bytes from 10.2.150.6: icmp_seq=1 ttl=64 time=0.481 ms
64 bytes from 10.2.150.6: icmp_seq=2 ttl=64 time=0.472 ms
^C
```

```
--- 10.2.150.6 ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.472/0.476/0.481/0.022 ms
```

```
[cloud-user@tenant350-150-inst1 ~]$ ping 10.2.200.5
PING 10.2.200.5 (10.2.200.5) 56(84) bytes of data.
64 bytes from 10.2.200.5: icmp_seq=1 ttl=63 time=1.20 ms
64 bytes from 10.2.200.5: icmp_seq=2 ttl=63 time=0.680 ms
^C
```

```
--- 10.2.200.5 ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.680/0.944/1.209/0.266 ms
```

```
[cloud-user@tenant350-150-inst1 ~]$ ping 10.2.200.6
PING 10.2.200.6 (10.2.200.6) 56(84) bytes of data.
64 bytes from 10.2.200.6: icmp_seq=1 ttl=63 time=1.24 ms
64 bytes from 10.2.200.6: icmp_seq=2 ttl=63 time=0.613 ms
64 bytes from 10.2.200.6: icmp_seq=3 ttl=63 time=0.761 ms
```

Errors – No errors observed in logs in /var/log/neutron

## Link Failures

```
interface Ethernet1/17
description Uplink from UCS FabA Port 17
switchport mode trunk
switchport trunk allowed vlan 1,100,110,120,130,160,215,251-263
```

```

switchport trunk allowed vlan add 265-272,274,276-277,279-293,295-298
switchport trunk allowed vlan add 300-308,310-314,316-322,324-327
switchport trunk allowed vlan add 329,331-332,334-351,353-359,361-368
switchport trunk allowed vlan add 370-379,381-386,389-397,399-408
switchport trunk allowed vlan add 410-411,413-414,417-420,422,426-429
switchport trunk allowed vlan add 431-433,435,438-439,442-443,448-449
switchport trunk allowed vlan add 451-452,455-456,459-460,464-465
switchport trunk allowed vlan add 468,471,473-475,480-481,483-484
switchport trunk allowed vlan add 486,491-492,495,498-500,502,504
switchport trunk allowed vlan add 506,508-511,515,517-518,523,526
switchport trunk allowed vlan add 536,538
spanning-tree port type edge trunk
channel-group 17 mode active

```

Ping continues

```

64 bytes from 10.22.162.79: icmp_seq=1 ttl=63 time=0.435 ms
64 bytes from 10.22.162.79: icmp_seq=3 ttl=63 time=0.359 ms

```

```

64 bytes from 10.22.162.79: icmp_seq=1 ttl=63 time=0.325 ms
64 bytes from 10.22.162.79: icmp_seq=2 ttl=63 time=0.471 ms
64 bytes from 10.22.162.79: icmp_seq=3 ttl=63 time=0.347 ms

```

```

64 bytes from 10.22.162.79: icmp_seq=1 ttl=63 time=0.391 ms
64 bytes from 10.22.162.79: icmp_seq=2 ttl=63 time=0.422 ms
64 bytes from 10.22.162.79: icmp_seq=3 ttl=63 time=0.356 ms

```

vPC Peer-link status

```

-----
id   Port   Status Active vlans
--   -
1    Po1    up     1,100,110,120,130,160,215,251-263,265-272,274,276-
      277,279-293,295-298,300-308,310-314,316-322,324-32
      7,329,331-332,334-351,353-359,361-368,370-379,381-
      386,389-397,399-408,410-411,413-414,417-420,422,42
      6-429,431-433,435,438-439,442-443,448-449,451-452, ....

```

vPC status

```

-----
id   Port   Status Consistency Reason           Active vlans
--   -
17   Po17   down*  success      success          -
18   Po18   up     success      success          1,100,110,1
      20,130,160,
      215,251-263
      ,265-272,27
      4,276-277,2 ....

```

Bring down e 1/18 also. Both interfaces on N9K down.

Ping continues

```

64 bytes from 10.22.162.79: icmp_seq=1 ttl=63 time=0.350 ms
64 bytes from 10.22.162.79: icmp_seq=2 ttl=63 time=0.515 ms
64 bytes from 10.22.162.79: icmp_seq=3 ttl=63 time=0.471 ms

```

```

64 bytes from 10.22.162.79: icmp_seq=1 ttl=63 time=0.518 ms
64 bytes from 10.22.162.79: icmp_seq=2 ttl=63 time=0.562 ms
64 bytes from 10.22.162.79: icmp_seq=3 ttl=63 time=0.457 ms

```

```
64 bytes from 10.22.162.79: icmp_seq=1 ttl=63 time=0.332 ms
64 bytes from 10.22.162.79: icmp_seq=2 ttl=63 time=0.381 ms
64 bytes from 10.22.162.79: icmp_seq=3 ttl=63 time=0.462 ms

64 bytes from 10.22.162.79: icmp_seq=1 ttl=63 time=0.227 ms
64 bytes from 10.22.162.79: icmp_seq=2 ttl=63 time=0.501 ms
```

## vPC Peer-link status

```
-----
id   Port   Status Active vlans
--   -
1    Po1    up      1,100,110,120,130,160,215,251-263,265-272,274,276-
      277,279-293,295-298,300-308,310-314,316-322,324-32
      7,329,331-332,334-351,353-359,361-368,370-379,381-
      386,389-397,399-408,410-411,413-414,417-420,422,42
      6-429,431-433,435,438-439,442-443,448-449,451-452, ....
```

## vPC status

```
-----
id   Port   Status Consistency Reason           Active vlans
--   -
17   Po17   down*  success      success          -
18   Po18   down*  success      success          -
```

## Switch Reboot and Nexus Plugins Replay

## Run Ping Tests while switch rebooting

```
while ;; do loop

ping -c 3 10.22.162.79 | grep icmp ...
64 bytes from 10.22.162.79: icmp_seq=1 ttl=63 time=2.49 ms
64 bytes from 10.22.162.79: icmp_seq=2 ttl=63 time=0.511 ms
64 bytes from 10.22.162.79: icmp_seq=3 ttl=63 time=0.308 ms

64 bytes from 10.22.162.79: icmp_seq=1 ttl=63 time=0.317 ms
64 bytes from 10.22.162.79: icmp_seq=2 ttl=63 time=0.494 ms
64 bytes from 10.22.162.79: icmp_seq=3 ttl=63 time=0.345 ms
```

## Reboot the Switch

```
UCSO-N9K-FAB-B(config)# reload
!!!WARNING! there is unsaved configuration!!!
This command will reboot the system. (y/n)? [n] y
```



The plugin does not copy the running configuration to startup. Instead when the switch comes up it replays the configuration from mysql. Hence the warning about the unsaved configuration above

---

Ping continues..

```
64 bytes from 10.22.162.79: icmp_seq=1 ttl=63 time=0.457 ms
64 bytes from 10.22.162.79: icmp_seq=2 ttl=63 time=0.297 ms
64 bytes from 10.22.162.79: icmp_seq=3 ttl=63 time=0.455 ms
```

When the switch comes up, check for available VLANs:

```
UCSO-N9K-FAB-B(config)# show vlan | grep q- | count
18
Not all VLAN's are not up yet.
```

```
Kernel uptime is 0 day(s), 0 hour(s), 6 minute(s), 43 second(s)
```

```
Reason: Reset Requested by CLI command reload
System version: 7.0(3)I1(3)
Service:
Active Packages:
```

```
UCSO-N9K-FAB-B(config)# show vlan | grep q- | count
200 - Replayed all 200 VLAN's for 400 VM's after kernel uptime of 6:43 minutes,
while the second switch provided business continuity.
```

Created Tenant 351 with 2 networks and 4 VM's after this

```
| 6ade33c4-3563-4219-8668-384d0fldfd05 | tenant351_151_inst1 |
2ca501f354764bacb7963c6ae780cd87 | ACTIVE | - | Running | tenant351-
151=10.2.151.6, 10.22.162.84 |
| c0dd0b44-fe9f-4816-8909-4e662020c9f8 | tenant351_151_inst2 |
2ca501f354764bacb7963c6ae780cd87 | ACTIVE | - | Running | tenant351-
151=10.2.151.5, 10.22.162.85 |
| 978550c5-fb99-4e15-81a0-7dc697193f53 | tenant351_201_inst3 |
2ca501f354764bacb7963c6ae780cd87 | ACTIVE | - | Running | tenant351-
201=10.2.201.6, 10.22.162.86 |
| b10f88b9-82ac-4942-a888-09371528cb70 | tenant351_201_inst4 |
2ca501f354764bacb7963c6ae780cd87 | ACTIVE | - | Running | tenant351-
201=10.2.201.5, 10.22.162.87 |
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
-----+
```

```
Segmentation ID: 537
```

```
Segmentation ID: 494
```

```
Floating IP's are 10.22.162.87 and 10.22.162.86
```

```
Connectivity works
```

```
[root@clouduser scripts]# ssh -i tenant351kp.pem -o StrictHostKeyChecking=no
cloud-user@10.22.162.86 hostname -s
```

```
tenant351-201-inst3
```

```
[root@clouduser scripts]# ssh -i tenant351kp.pem -o StrictHostKeyChecking=no
cloud-user@10.22.162.87 hostname -s
```

```
tenant351-201-inst4
```

On the switch

```
UCSO-N9K-FAB-B(config)# show vlan | egrep "q-537|q-494"
```

```
494 q-494 active Po1, Po17, Po18, Eth1/1, Eth1/2
537 q-537 active Po1, Po17, Po18, Eth1/1, Eth1/2
```

## HA on Controller Blades

Controllers are key for the health of the cloud which hosts most of the OpenStack services. There are three types of controller failures that could happen.

Server reboot, pulling the blade out of the chassis while system is up and running and putting it back, pulling the blade from the chassis and replacing it simulating a total failure of the controller node.

### Server Reboot Tests

Run Health check before to make sure that system is healthy.

- Run nova list after sourcing stackrc as stack user on Undercloud node to verify that all the controllers are in healthy state as below

```
[stack@osp7-director ~]$ nova list
```

ID	Name	Status	Task State	Power State	Networks
948b754f-c992-4211-940a-2308bcff31a6	overcloud-cephstorage-0	ACTIVE	-	Running	ct plane=10.22.110.78
790ff133-bc91-4877-8f1f-200417435e08	overcloud-cephstorage-1	ACTIVE	-	Running	ct plane=10.22.110.79
97f6f1ea-0ba2-4f3d-b2b3-f068d17d2509	overcloud-cephstorage-2	ACTIVE	-	Running	ct plane=10.22.110.52
fc0e7fd5-a3e5-4f19-9cf5-80311fd2efd0	overcloud-compute-0	ACTIVE	-	Running	ct plane=10.22.110.61
aa42c6c7-7222-40f9-9242-61bd59e45760	overcloud-compute-1	ACTIVE	-	Running	ct plane=10.22.110.53
a323d1d4-6678-4a74-8b7e-007a4fad4e5b	overcloud-compute-2	ACTIVE	-	Running	ct plane=10.22.110.56
1fbc8a0d-b739-43f4-8360-6d1ccd8f0d8e	overcloud-compute-3	ACTIVE	-	Running	ct plane=10.22.110.60
5d9db6e3-0ac5-4329-8734-3ec7069847f8	overcloud-compute-4	ACTIVE	-	Running	ct plane=10.22.110.58
6454cd25-fcd2-44d4-9296-2ad3f54415bf	overcloud-compute-5	ACTIVE	-	Running	ct plane=10.22.110.54
80dd5a71-4e43-4bc4-8b18-35e78835c67f	overcloud-controller-0	ACTIVE	-	Running	ct plane=10.22.110.59
938b1742-5a24-42ff-8268-4e397ca87232	overcloud-controller-1	ACTIVE	-	Running	ct plane=10.22.110.55
e741cd25-9abf-4fee-a8bd-b7fe87695ece	overcloud-controller-2	ACTIVE	-	Running	ct plane=10.22.110.57

- Run pcs status on controller nodes and grep for error or stopped.

```
Online: [ overcloud-controller-0 overcloud-controller-1 overcloud-controller-2 ]
```

Full list of resources:

```
ip-172.22.215.91      (ocf::heartbeat:IPAddr2):      Started overcloud-
controller-0
```

```
Clone Set: haproxy-clone [haproxy]
```

```
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
```

```
ip-10.22.100.51      (ocf::heartbeat:IPAddr2):      Started overcloud-
controller-2
```

```
ip-10.22.100.50      (ocf::heartbeat:IPAddr2):      Started overcloud-
controller-2
```

```
ip-10.22.150.50      (ocf::heartbeat:IPAddr2):      Started overcloud-
controller-0
```

```
Master/Slave Set: galera-master [galera]
```

```
Masters: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
```

```
ip-10.22.110.45      (ocf::heartbeat:IPAddr2):      Started overcloud-
controller-2
```

```
.....
```

```
.....
```

PCSD Status:

```
overcloud-controller-0: Online
```

```
overcloud-controller-1: Online
```

```
overcloud-controller-2: Online
```

Daemon Status:

```
corosync: active/enabled
```

```
pacemaker: active/enabled
```

```
pcsd: active/enabled
```

- Reboot the fist controller node and check for pcs status and connectivity of the VM's.
- When the controller comes up, wait till all the services are through PCS are up and running.

- Connect to n1kv VSM and make sure that the standby is is ha-status.
- Repeat reboot of the second node and then the third node after the second comes up fully.

### Health Checks and Observations

The following is a list of health checks and observations:

- Do not reboot the second controller unless the prior one comes up first. Check pacemaker status, **health of quorum ( corosync )**, **health of n1kv's primary and standby VSM's**.
- Two controllers are minimum needed for healthy operation.
- While the first node is booting up, it takes time for pcs status command to complete.

```
PCS will report one server is offline.
PCSD Status:
  overcloud-controller-0: Online
  overcloud-controller-1: Offline
  overcloud-controller-2: Online
Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```



Corosync will return that it gets only 2 votes out of 3 as below while the server is getting rebooted. This is normal.

```
[root@overcloud-controller-2 ~]# corosync-quorumtool
Quorum information
-----
Date:                Thu Jan 28 12:09:16 2016
Quorum provider:    corosync_votequorum
Nodes:              2
Node ID:            3
Ring ID:            56
Quorate:            Yes

Votequorum information
-----
Expected votes:    3
Highest expected:  3
Total votes:       2
Quorum:            2
Flags:             Quorate

Membership information
-----
   Nodeid     Votes Name
     1         1  overcloud-controller-0
     3         1  overcloud-controller-2 (local)
```

- Refer to bug [1281603](#). It might take just over 3 minutes for the other controllers to start rescheduling the routers. Within this time frame, slower keystone authentication, and **creation of VM's observed**. However, system recover fine after this. The issue of I3\_ha=false to default of true from false is being addressed in the next release of n1kv.

- When the node comes up, the routers remain on the other 2 controllers and do not fall back. Can be queried with ip netns too.
- If controller node does not come up, check through KVM console to spot out any issues and hold off rebooting the second node before a healthy operation of the first.

### Blade Pull Tests

One of the controller nodes blade was pulled out while the system is up and running. The validation tests like VM creation etc were done prior to the tests and to check the status when the blade is pulled from the chassis. This is like simulating a complete blade failure. After around 60 minutes the blade was re-inserted back in the chassis.

### Health Checks and Observations

The same behavior as [observed during reboot](#) were noticed during the blade pull tests. However unlike a reboot which completes in 5-10 minutes, this was for an extended period of time of 60 minutes to check the status of the cluster.

- Cisco UCS marks the blade as 'removed' and prompts to resolve the slot issue.
- Nova declares the state of the server as NOSTATE as shown below:

```
| e902ad92-f600-41ec-a525-32218be1ee11 | overcloud-controller-0 | ACTIVE | - | Running | ctlplane=10.22.110.59 |
| a23af643-51c8-4f59-881c-77a9d5e1557f | overcloud-controller-1 | ACTIVE | - | NOSTATE | ctlplane=10.22.110.54 |
| f01d22ef-18a4-4f01-b4c1-1ffb6f1d9262 | overcloud-controller-2 | ACTIVE | - | Running | ctlplane=10.22.110.57 |
```

- Ironic gives up as it cannot bring the server back online and enables Maintenance mode to True for this node.
- Compare the Instance UUID from ironic node-list and ID from nova list

```
[stack@osp7-director ~]$ ironic node-list
```

UUID	Name	Instance UUID	Power State	Provision State	Maintenance
b7dde876-354a-4688-8550-aec8f64c582c	None	a23af643-51c8-4f59-881c-77a9d5e1557f	None	active	True
e4563ca5-2f12-4e08-9905-f770f740ad2b	None	e902ad92-f600-41ec-a525-32218be1ee11	power on	active	False
285965a9-9713-4301-8ad5-7aa3ef5dd1c2	None	f01d22ef-18a4-4f01-b4c1-1ffb6f1d9262	power on	active	False

```
[stack@osp7-director ~]$ ironic node-show b7dde876-354a-4688-8550-aec8f64c582c | more
```

Property	Value
target_power_state	None
extra	{u'newly_discovered': u'true', u'block_devices': {u'serials': [u'618e7283727010e01dfcad510ceb915e', u'40E10200869217CA']}, u'hardware_swift_object': u'extra hardware-b7dde876-354a-4688-8550-aec8f64c582c'}
last_error	During sync_power_state, max retries exceeded for node b7dde876-354a-4688-8550-aec8f64c582c, node state None does not match expected state 'power on'. Updating DB state to 'None' Switching node to maintenance mode.

- Ceph storage will report that 1 out of 3 monitors are down. All the 3 controllers will be running one monitor each. However all the OSD's are up and running.

```

cluster 3bd648c2-c091-11e5-8fff-0025b522225f
health HEALTH_WARN
1 mons down, quorum 0,1 overcloud-controller-0,overcloud-
controller-2
monmap e1: 3 mons at {overcloud-controller-0=10.22.120.51:6789
/0,overcloud-controller-1=10.22.120.57:6789/0,overcloud-controller-
2=10.22.120.53:6789/0}
election epoch 30, quorum 0,1 overcloud-controller-0,ov
ercloud-controller-2
osdmap e93: 24 osds: 24 up, 24 in
pgmap v51504: 1024 pgs, 4 pools, 4928 MB data, 1195 objects
15768 MB used, 128 TB / 128 TB avail
1024 active+clean

```




---

All of the above, the behavior UCS asking to resolve slot, ironic turning the blade to maintenance mode, nova setting the status to NOSTATE, and ceph reporting on one of the monitors as down are expected.

---

- After inserting the blade back into the same slot of the chassis, it needed a manual intervention to correct the above.
  - Insert the blade back into the slot and resolve the slot issue in UCS.
  - ironic node-set-power-state b7dde876-354a-4688-8550-aec8f64c582c on
  - ironic node-set-maintenance b7dde876-354a-4688-8550-aec8f64c582c false
  - Wait for a minute and check back for these columns with ironic node-list.
  - nova reset-state --active a23af643-51c8-4f59-881c-77a9d5e1557f
  - Wait for about 5-10 minutes for nova to act upon this and re-query the status with nova list. It should turn it back as active like other controller nodes.
  - Login to the controller node, check for pcs status and resolve any processes that were not **brought up running 'pcs resource cleanup'**
  - If the monitor is still down and/or taking longer time issue /etc/init.d/ceph restart mon on the controller node(s).

### Blade Replacement

Unlike the above two types of failures, in this test the blade is completely removed and new one is added. There were few issues encountered while rebuilding the failed controller blade and adding it as a replacement. The fix for bug 1298430 will give business continuity but there is a need to fix the failed blade. While this issue is being investigated, an interim solution was developed to circumvent the above limitation. This is included in the [Hardware failures](#) section. Different types of hardware failures that can happen on a controller blade and how to mitigate these issues considering the dependency of Controller blade on IPMI and MAC addresses is addressed there.

## HA on Compute Blades

### Reboot Test

Tests and Observations are as follows:

- Many Instances were provisioned across the pod and reboot of the Compute Node was attempted.

```
[root@overcloud-compute-0 ~]# virsh list
Id      Name                               State
-----
 2      instance-0000000f                 running
 3      instance-00000021                 running
 4      instance-00000033                 running
 5      instance-00000045                 running
 6      instance-00000057                 running
 7      instance-00000069                 running
 8      instance-0000007b                 running
 9      instance-00000081                 running
10      instance-0000009c                 running
11      instance-000000ab                 running
12      instance-000000ba                 running
13      instance-000000bd                 running
14      instance-000000d8                 running
15      instance-000000e7                 running
16      instance-000000f6                 running
17      instance-000000f9                 running
18      instance-00000114                 running
19      instance-00000123                 running
```

- Identified a compute host to be **rebooted and the VM's that could be impacted**

About **19 VM's** were up and running

```
[root@overcloud-controller-0 heat-admin]# nova-manage vm list | egrep -i \
"overcloud-compute-0|project|active" | sort -k 1 | awk '{print $1" "$2" "$3"
"$4}'
instance node type state
tenant302_152_inst3 overcloud-compute-0.localdomain m1.demo active
tenant304_104_inst1 overcloud-compute-0.localdomain m1.demo active
tenant305_155_inst3 overcloud-compute-0.localdomain m1.demo active
tenant307_107_inst1 overcloud-compute-0.localdomain m1.demo active
tenant308_158_inst3 overcloud-compute-0.localdomain m1.demo active
tenant310_110_inst1 overcloud-compute-0.localdomain m1.demo active
tenant311_161_inst3 overcloud-compute-0.localdomain m1.demo active
tenant312_112_inst1 overcloud-compute-0.localdomain m1.demo active
tenant314_114_inst2 overcloud-compute-0.localdomain m1.demo active
tenant315_165_inst3 overcloud-compute-0.localdomain m1.demo active
tenant316_166_inst4 overcloud-compute-0.localdomain m1.demo active
tenant317_117_inst1 overcloud-compute-0.localdomain m1.demo active
tenant319_119_inst2 overcloud-compute-0.localdomain m1.demo active
tenant320_170_inst3 overcloud-compute-0.localdomain m1.demo active
tenant321_171_inst4 overcloud-compute-0.localdomain m1.demo active
tenant322_122_inst1 overcloud-compute-0.localdomain m1.demo active
tenant324_124_inst2 overcloud-compute-0.localdomain m1.demo active
tenant325_175_inst3 overcloud-compute-0.localdomain m1.demo active
[root@overcloud-controller-0 heat-admin]# nova-manage vm list | egrep
"overcloud-compute-0|project|active" | sort -k 1 | awk '{print $1" "$2" "$3"
"$4}' | wc -l
```

19

- **Identify the floating IP's for these VM's from nova list --all-tenants and capture data to login without password, run ifconfig script. The script ssh's to all the VM's run's ifconfig and returns serially.**

Running a script N-S for all the VM's

```
[root@clouduser scripts]# date; ./temp.sh > /dev/null; date;
Tue Nov 24 03:37:10 PST 2015
Tue Nov 24 03:37:13 PST 2015
[root@clouduser scripts]# tail -1 temp.sh
ssh -i tenant325kp.pem -o StrictHostKeyChecking=no cloud-user@10.22.160.181
/tmp/run1.sh
[root@clouduser scripts]# cat run1.sh
#!/bin/bash
date1=`date`;
host=`hostname -s`
network=`/sbin/ifconfig | grep broadcast`
date2=`date`;
echo "Host is $host and Network is $network from $date1 to $date2"
[root@clouduser scripts]#
```



set `resume_guests_state_on_host_boot=true` in `nova.conf` to get the instances back online after reboot.

- Rebooted the Compute Node `overcloud-compute-0`
- Instances came up fine and the same script to validate the login with floating ip's worked fine.
- By default guests will not come up unless `resume_guests_state_on_host_boot` is set to true. If this parameter isn't set before reboot:

```
root@overcloud-controller-0 heat-admin]# nova-manage vm list | egrep
"overcloud-compute-0|project" | sort -k 1 | awk '{print $1" "$2" "$3" "$4}'
tenant302_152_inst3 overcloud-compute-0.localdomain m1.demo stopped
tenant304_104_inst1 overcloud-compute-0.localdomain m1.demo stopped
tenant305_155_inst3 overcloud-compute-0.localdomain m1.demo stopped
.....
Observe a failure from dashboard as well
```

Error: Unable to soft reboot instance:  
tenant321\_171\_inst4

You may have to hard reboot the instances as `nova reboot --hard $i` after capturing the instance id's from `nova list --all-tenants`.

Query the instances as shown below:

```
[root@overcloud-compute-0 nova]# virsh list --all
Id      Name                               State
-----
21      instance-000000f6                 running
22      instance-0000000f                 running
23      instance-00000021                 running
```

### Blade Pull Tests

One of the Compute blades was pulled out while the system is up and running. This was also an extended test for about 60 minutes and then the blade was re-inserted back in the chassis.

Observations:

- Results were similar to reboot tests above.
- UCS Manager complained to resolve the host as it was pulled out from the chassis. This was acknowledged and the blade was re-inserted.
- **The guest VM's came up when resume\_guests property was set to true at host level.**
- Similar to Controller blade **pull tests, nova made the state as 'NOSTATE' and ironic put the blade into Maintenance mode.**
- Similar steps like setting up the maintenance mode to off through ironic and nova reset state were issued to the blade after **getting 'ok' status in UCS manager.**

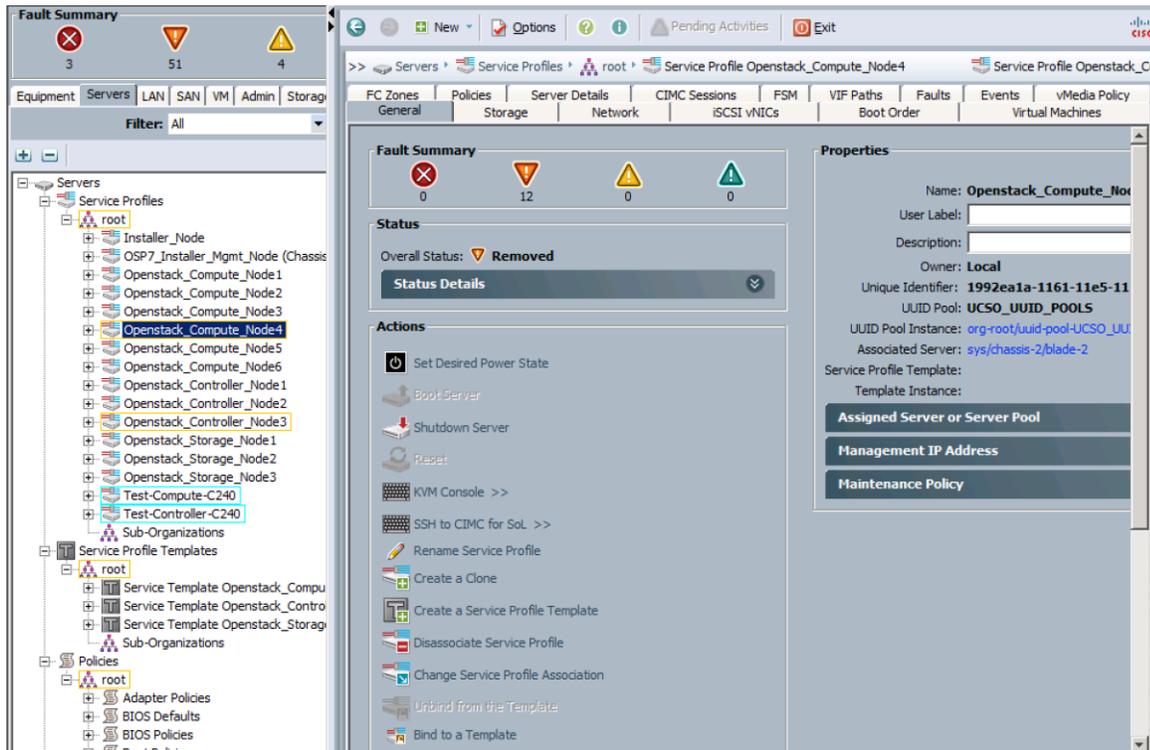
### Blade Replacement

Compute blade was pulled from the chassis completely and the server was decommissioned in UCS. This is to simulate a complete failure of a Compute blade. Then an attempt was made to remove this from OpenStack and add a new blade to the cloud. The service profile was reused in this method. The following were the tasks list and observations made during a Compute blade replacement test.

Blade replacement is actually a two phase process. First remove the faulty blade from the system and then add a new one.

To delete a Node, complete the following steps:

1. Blade pulled from chassis.



2. Nova makes the blade to 'NOSTATE' as shown below.

```
[stack@osp7-director ~]$ nova list
+-----+-----+-----+-----+-----+-----+
| ID | Name | Status | Task State | Power State | Networks |
+-----+-----+-----+-----+-----+-----+
| 138c1151-96d2-42af-bb92-df69c6ec4148 | overcloud-cephstorage-0 | ACTIVE | - | Running | ctlplane=10.22.110.66 |
| c483aec9-f350-4e51-9638-cee3659cf8ff | overcloud-cephstorage-1 | ACTIVE | - | Running | ctlplane=10.22.110.68 |
| 66550eb9-8a7d-4b78-9c38-8e7d89dd4a37 | overcloud-cephstorage-2 | ACTIVE | - | Running | ctlplane=10.22.110.67 |
| 42480398-bd69-4159-8454-007aae8d48c5 | overcloud-compute-0 | ACTIVE | - | Running | ctlplane=10.22.110.79 |
| 79e2042e-c103-4cfd-a265-0869194b981d | overcloud-compute-1 | ACTIVE | - | Running | ctlplane=10.22.110.78 |
| 42ca0503-1a46-48da-9580-eb9cc275abe1 | overcloud-compute-2 | ACTIVE | - | Running | ctlplane=10.22.110.71 |
| 0674eca8-13fa-4f0d-b307-40e07b884bce | overcloud-compute-3 | ACTIVE | - | Running | ctlplane=10.22.110.77 |
| 0c51e4fc-17d0-42ff-a6cb-701b3a03a7e7 | overcloud-compute-4 | ACTIVE | - | Running | ctlplane=10.22.110.75 |
| ae2fe1ad-2840-4206-a165-3ce97847d26b | overcloud-compute-5 | ACTIVE | - | NOSTATE | ctlplane=10.22.110.74 |
| cffc812d-9928-4d30-a107-9e97294a8f53 | overcloud-controller-0 | ACTIVE | - | Running | ctlplane=10.22.110.72 |
| 8893e54f-faf2-474b-9293-d7dfbbe9c6f8 | overcloud-controller-1 | ACTIVE | - | NOSTATE | ctlplane=10.22.110.70 |
| 8b0f4d5d-352e-4751-88a8-2bb42b38927a | overcloud-controller-2 | ACTIVE | - | Running | ctlplane=10.22.110.69 |
+-----+-----+-----+-----+-----+-----+
[stack@osp7-director ~]$
```

3. Evacuate the VM's from the failed node.

```
[stack@osp7-director ~]$ nova host-list | grep compute
| overcloud-compute-4.localdomain | compute | nova |
| overcloud-compute-0.localdomain | compute | nova |
| overcloud-compute-1.localdomain | compute | nova |
| overcloud-compute-5.localdomain | compute | nova |
| overcloud-compute-3.localdomain | compute | nova |
| overcloud-compute-2.localdomain | compute | nova |
[stack@osp7-director ~]$
```

```
[stack@osp7-director ~]$ nova host-evacuate overcloud-compute-5.localdomain --shared-storage
+-----+-----+-----+-----+
| Server UUID | Evacuate Accepted | Error Message |
+-----+-----+-----+-----+
| 9041b66d-d752-4fb7-ac7e-8469365d4246 | True | |
+-----+-----+-----+-----+
```

4. An attempt was made to run remove this node. Refer to the [online documentation](#).

```
openstack overcloud node delete --stack overcloud --templates \
-e /usr/share/openstack-tripleo-heat-templates/overcloud-resource-registry-
puppet.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-
isolation.yaml \
-e /home/stack/templates/network-environment.yaml \
-e /home/stack/templates/storage-environment.yaml \
-e /home/stack/templates/cisco-plugins.yaml \
8b0f4d5d-352e-4751-88a8-2bb42b38927a
--verbose --debug --log-file overcloud_test.log
```

Here the node-uuid as observed from nova list was added.

5. As the blade was completely pulled from the chassis ironic was unable to get to the power management and the above node-delete failed.

```
ipmitool -I lanplus -H <ipmi address> -U admin -P <password> chassis power
status
Error: Unable to establish IPMI v2 / RMCP+ session
Error: Unable to establish IPMI v2 / RMCP+ session
Error: Unable to establish IPMI v2 / RMCP+ session
Unable to get Chassis Power Status
```

6. Workarounds to delete the blade in the current status are as follows:

```
Update the error status to available status in ironic node-list
edit /etc/ironic/ironic.conf
Update the enabled drivers temporarily as below
#enabled_drivers=pxe_ipmitool,pxe_ssh,pxe_drac
enabled_drivers=fake
Restart openstack-ironic-conductor
Sudo service openstack-ironic-conductor restart

ironic node-update NODE_UUID replace driver=fake
```

The node in ironic node-list should be with provision-state=active and maintenance=false  
If not

```
ironic node-set-provision-state NODE_UUID provide
ironic node-set-provision-state NODE_UUID active
```

```
ironic node-set-provision-state NODE_UUID deleted
```

The power status should be on, provision state as available and maintenance as false before moving ahead.

Run nova service-list and identify the service id's

**Delete the service id's associated with this node as**

```
nova service-delete $id
Delete the node from nova
nova delete NODE_UUID
Delete the node from ironic
ironic node-delete NODE_UUID
```

Revert back the “fake” driver from `ironic.conf`

```
edit vi /etc/ironic/ironic.conf.
enabled_drivers=pxe_ipmitool,pxe_ssh,pxe_drac
#enabled_drivers=fake
```

Restart `ironic-conductor` to pick up the drivers again.

```
service openstack-ironic-conductor restart
```



The deleted node should not exist anymore in `ironic node-list` or `nova-list` now.

---

### Node Addition

When the compute blade has been completely removed from OpenStack, a new blade can be added. The procedure for adding a new compute blade is same as how it was addressed earlier in [upscaling the compute pod](#).

## HA on Storage Nodes

Ceph, the software stack deployed by Red Hat OpenStack Director, has its high availability built in itself. By default, the system will be replicating the placement groups and has 3 copies distributed across the hosts.

The parameter `osd_pool_default_size = 3` in `ceph.conf` brings this feature by default when installed.

If we create a crushmap from the existing cluster as below it reveals what type of buckets are in and what mode of replication is being done by default in the cluster.

```
ceph osd getcrushmap -o /tmp/crushmap.bin
```

```
crushtool -d crushmap.bin -o /tmp/crushmap.txt
```

```
rule replicated_ruleset {
    ruleset 0
    type replicated ← Default to Replication mode
    min_size 1
    max_size 10
    step take default
    step chooseleaf firstn 0 type host ←Default distribution of PG copies
    step emit
}
```

Whenever a Ceph node goes down, the system will start rebuilding from the copies of replicas. While this is an expected feature of Ceph, it causes some CPU and memory overhead too. This is one of the reasons to have a minimum of 3 nodes for ceph and leave some good amount of free space with in the storage cluster. This will help Ceph to move the blocks around in case of failures like this. More the nodes better it is, as this rebuild activity is distributed across the cluster. Though there are other parameters like `osd_max_backfills` to control this activity and its impact on CPU, it may not be feasible to cover all of these recovery parameters in this document.

What needs to be noted is that the recovery kicks in as part of the tests below. The ceph cluster status may show warnings while the tests are being conducted as it is moving the placement groups and may cause performance issues on the storage cluster. Hence checking the health of nodes while adding/rebuilding a new node is important.

## Reboot Test

1. Check the status of the cluster:

```
[root@overcloud-cephstorage-0 tmp]# ceph -s
cluster 3bd648c2-c09f-11e5-8fff-0025b522225f
health HEALTH_OK
monmap e1: 3 mons at {overcloud-controller-0=10.22.120.51:6789/0,
overcloud-controller-1=10.22.120.57:6789/0,overcloud-controller-2=10.22.120.53:6789/0}
election epoch 34, quorum 0,1,2 overcloud-controller-0,overcloud-controller-2,
overcloud-controller-1
osdmap e93: 24 osds: 24 up, 24 in|
pgmap v53890: 1024 pgs, 4 pools, 4928 MB data, 1195 objects
15773 MB used, 128 TB / 128 TB avail
1024 active+clean
client io 462 B/s wr, 0 op/s
[root@overcloud-cephstorage-0 tmp]# ceph osd tree
ID WEIGHT TYPE NAME UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 128.87988 root default
-2 42.95996 host overcloud-cephstorage-0
  1 5.37000 osd.1 up 1.00000 1.00000
  4 5.37000 osd.4 up 1.00000 1.00000
  7 5.37000 osd.7 up 1.00000 1.00000
 10 5.37000 osd.10 up 1.00000 1.00000
 13 5.37000 osd.13 up 1.00000 1.00000
 16 5.37000 osd.16 up 1.00000 1.00000
 19 5.37000 osd.19 up 1.00000 1.00000
 22 5.37000 osd.22 up 1.00000 1.00000
-3 42.95996 host overcloud-cephstorage-2
  0 5.37000 osd.0 up 1.00000 1.00000
  3 5.37000 osd.3 up 1.00000 1.00000
  6 5.37000 osd.6 up 1.00000 1.00000
  9 5.37000 osd.9 up 1.00000 1.00000
 12 5.37000 osd.12 up 1.00000 1.00000
 15 5.37000 osd.15 up 1.00000 1.00000
 18 5.37000 osd.18 up 1.00000 1.00000
 21 5.37000 osd.21 up 1.00000 1.00000
-4 42.95996 host overcloud-cephstorage-1
  2 5.37000 osd.2 up 1.00000 1.00000
  5 5.37000 osd.5 up 1.00000 1.00000
  8 5.37000 osd.8 up 1.00000 1.00000
 11 5.37000 osd.11 up 1.00000 1.00000
 14 5.37000 osd.14 up 1.00000 1.00000
 17 5.37000 osd.17 up 1.00000 1.00000
 20 5.37000 osd.20 up 1.00000 1.00000
 23 5.37000 osd.23 up 1.00000 1.00000
```

2. Reboot one of the Ceph storage node:

The following was observed during the reboot and running `ceph -w`

```
mon.0 [INF] osd.13 marked itself down
mon.0 [INF] osd.15 marked itself down
mon.0 [INF] osd.10 marked itself down
mon.0 [INF] osd.8 marked itself down
mon.0 [INF] osd.6 marked itself down
mon.0 [INF] osd.4 marked itself down
mon.0 [INF] osd.2 marked itself down
mon.0 [INF] osd.0 marked itself down
mon.0 [INF] osdmap e176: 24 osds: 16 up, 24 in
```

## ceph osd tree reports

```
[root@overcloud-cephstorage-1 ~]# ceph osd tree
ID WEIGHT TYPE NAME UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 128.87988 root default
-2 42.95996 host overcloud-cephstorage-0
  0 5.37000 osd.0 down 1.00000 1.00000
  2 5.37000 osd.2 down 1.00000 1.00000
  4 5.37000 osd.4 down 1.00000 1.00000
  6 5.37000 osd.6 down 1.00000 1.00000
  8 5.37000 osd.8 down 1.00000 1.00000
 10 5.37000 osd.10 down 1.00000 1.00000
 13 5.37000 osd.13 down 1.00000 1.00000
 15 5.37000 osd.15 down 1.00000 1.00000
-3 42.95996 host overcloud-cephstorage-1
  1 5.37000 osd.1 up 1.00000 1.00000
  3 5.37000 osd.3 up 1.00000 1.00000
  5 5.37000 osd.5 up 1.00000 1.00000
  7 5.37000 osd.7 up 1.00000 1.00000
  9 5.37000 osd.9 up 1.00000 1.00000
 11 5.37000 osd.11 up 1.00000 1.00000
 14 5.37000 osd.14 up 1.00000 1.00000
 16 5.37000 osd.16 up 1.00000 1.00000
-4 42.95996 host overcloud-cephstorage-2
 12 5.37000 osd.12 up 1.00000 1.00000
 17 5.37000 osd.17 up 1.00000 1.00000
 18 5.37000 osd.18 up 1.00000 1.00000
 19 5.37000 osd.19 up 1.00000 1.00000
 20 5.37000 osd.20 up 1.00000 1.00000
 21 5.37000 osd.21 up 1.00000 1.00000
 22 5.37000 osd.22 up 1.00000 1.00000
 23 5.37000 osd.23 up 1.00000 1.00000
```

3. Make sure the VM's connectivity through floating IP from an external host is successful.

```
Host is tenant310-110-inst2 and Network is inet 10.2.110.6 netmask
255.255.255.0
Host is tenant310-160-inst3 and Network is inet 10.2.160.5 netmask
255.255.255.0
Host is tenant310-160-inst4 and Network is inet 10.2.160.6 netmask
255.255.255.0
Host is tenant311-111-inst1 and Network is inet 10.2.111.5 netmask
255.255.255.0
Host is tenant311-111-inst2 and Network is inet 10.2.111.6 netmask
255.255.255.0
Host is tenant311-161-inst3 and Network is inet 10.2.161.5 netmask
255.255.255.0
Host is tenant311-161-inst4 and Network is inet 10.2.161.6 netmask
255.255.255.0
Host is tenant312-112-inst1 and Network is inet 10.2.112.6 netmask
255.255.255.0
```

4. Ceph attempts recovery as shown below:

```
[root@overcloud-cephstorage-1 ~]# ceph -s
cluster 701e64ca-8e54-11e5-9eef-0025b522225f
health HEALTH_WARN
    256 pgs degraded
    .....
    8/24 in osds are down
monmap e2: 3 mons at {overcloud-controller-0=10.22.120.52:6789/0,
overcloud-controller-1=10.22.120.51:6789/0,overcloud-controller-
2=10.22.120.57:6789/0}
```

```
election epoch 30, quorum 0,1,2 overcloud-controller-1,overcloud-
controller-0,overcloud-controller-2
osdmap e177: 24 osds: 16 up, 24 in
pgmap v50622: 256 pgs, 4 pools, 97547 MB data, 24754 objects
24754/74262 objects degraded (33.333%)
.....
client io 1473 B/s rd, 411 kB/s wr, 108 op/s
```

The node comes after few minutes, while the cluster shows warning issues during the reboot period.

The status of the cluster observed fine after few minutes of reboot. The warning message continues until the recovery activity is complete.

### System Power Off

The behavior in system power off is very similar to what observed on Controller and Compute blade pull tests.

System took around 6 minutes to come back to OK status. The time system takes to recover depends on the active number of placement group and copies the system was attempting to move around.

There is a more detailed description and symptoms observed during power off that are listed in Node Replacement section below.

### Node Replacement

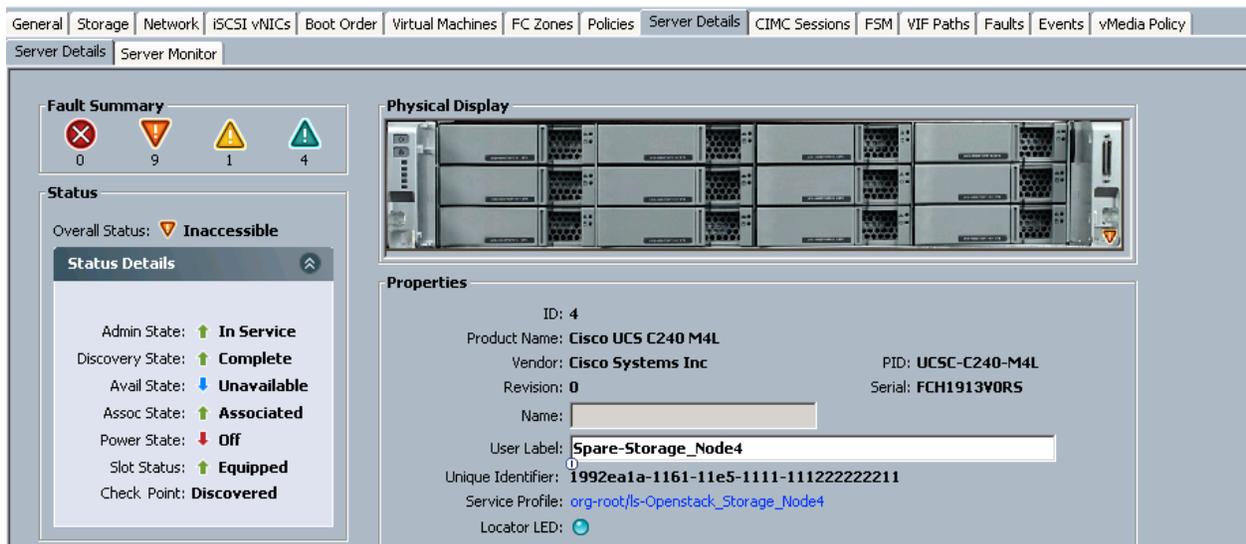
One of the storage servers was powered off ( pull the power cord ) completely and the server was decommissioned in UCS. This is to simulate a complete failure of the storage server. Then an attempt was made to remove this node from OpenStack and add a new one to the cloud. The following were the tasks list and observations made during a Storage node replacement test.



Node replacement is actually a two phase process. First remove the server from the system and then add a new one.

To delete a node, complete the following steps:

1. Power off the node by pulling the power cord from a running cluster.



Name	Overall Status	PID	Model	Serial	User Label	Cores	Memory	Ada...	NICs	HBAs	Operability	Power State	Assoc State	Profile
Server 1 (St...	Ok	UCSC-C240-M4L	Cisc...	FCH1850V2MS	Storage_Node3	20	262144	2	3	0	Operable	On	Associated	org-root/l...
Server 2 (St...	Ok	UCSC-C240-M4L	Cisc...	FCH1911V28X	Storage_Node2	20	262144	2	3	0	Operable	On	Associated	org-root/l...
Server 3 (St...	Ok	UCSC-C240-M4L	Cisc...	FCH1912V08S	Storage_Node1	20	262144	2	3	0	Operable	On	Associated	org-root/l...
Server 4 (S...	Inaccessible	UCSC-C240-M4L	Cisc...	FCH1913V0RS	Spare-Storag...	20	262144	1	3	0	Operable	Off	Associated	org-root/l...

```

cluster 3bd648c2-c09f-11e5-8fff-0025b522225f
health HEALTH_WARN
776 pgs degraded
670 pgs stuck unclean
776 pgs undersized
recovery 8845/34170 objects degraded (25.885%)
8/32 in osds are down
monmap e1: 3 mons at {overcloud-controller-0=10.22.120.51:6789/0,overcloud-controller-
1=10.22.120.57:6789/0,overcloud-controller-2=10.22.120.53:6789/0}
election epoch 40, quorum 0,1,2 overcloud-controller-0,overcloud-controller-2,overcloud-controller-1
osdmap e130: 32 osds: 24 up, 32 in
pgmap v60000: 1024 pgs, 4 pools, 45163 MB data, 11390 objects
133 GB used, 171 TB / 171 TB avail
8845/34170 objects degraded (25.885%)
776 active+undersized+degraded
248 active+clean

Fri Jan 29 00:23:54 EST 2016

cluster 3bd648c2-c09f-11e5-8fff-0025b522225f
health HEALTH_OK
monmap e1: 3 mons at {overcloud-controller-0=10.22.120.51:6789/0,overcloud-controller-
1=10.22.120.57:6789/0,overcloud-controller-2=10.22.120.53:6789/0}
election epoch 40, quorum 0,1,2 overcloud-controller-0,overcloud-controller-2,overcloud-controller-1
osdmap e132: 32 osds: 24 up, 24 in
pgmap v60142: 1024 pgs, 4 pools, 45163 MB data, 11390 objects
133 GB used, 128 TB / 128 TB avail
1024 active+clean
client io 9722 B/s wr, 3 op/s
Fri Jan 29 00:29:54 EST 2016
    
```

[stack@osp7-director ~]\$ nova list

ID	Name	Status	Task State	Power State	Networks
47e64b48-f105-49f8-ae40-f04db9c39313	overcloud-cephstorage-0	ACTIVE	-	Running	ctlplane=10.22.110.52
0a1f7293-d9ee-423c-80bc-96125d29d924	overcloud-cephstorage-1	ACTIVE	-	Running	ctlplane=10.22.110.80
6be9ea47-39c3-4b4e-b755-a866e47b8398	overcloud-cephstorage-2	ACTIVE	-	Running	ctlplane=10.22.110.76
19659ace-53b1-4a86-b0b8-21439aa8ab1a	overcloud-cephstorage-3	ACTIVE	-	NOSTATE	ctlplane=10.22.110.63
ab811350-d49e-4e3e-bad7-9afeb6e1d10a	overcloud-compute-0	ACTIVE	-	Running	ctlplane=10.22.110.53
211bb71c-2e72-44b1-a867-5f7babf4d4bb	overcloud-compute-1	ACTIVE	-	Running	ctlplane=10.22.110.61
db50a27a-4699-4fd9-9687-ec5403db3409	overcloud-compute-2	ACTIVE	-	Running	ctlplane=10.22.110.58
b4a04e95-6624-4f13-8a14-5ef1e6742b94	overcloud-compute-3	ACTIVE	-	Running	ctlplane=10.22.110.62
7d9caa7c-4e53-4832-983c-44706def4d42	overcloud-compute-4	ACTIVE	-	Running	ctlplane=10.22.110.56
869662d6-e5ae-4724-82fb-3eb9a6f74e5b	overcloud-compute-5	ACTIVE	-	Running	ctlplane=10.22.110.55
e902ad92-f600-41ec-a525-32218be1ee11	overcloud-controller-0	ACTIVE	-	Running	ctlplane=10.22.110.59
a23af643-51c8-4f59-881c-77a9d5e1557f	overcloud-controller-1	ACTIVE	-	Running	ctlplane=10.22.110.54
f01d22ef-18a4-4f01-b4c1-1ffb6f1d9262	overcloud-controller-2	ACTIVE	-	Running	ctlplane=10.22.110.57

[stack@osp7-director ~]\$ ironic node-list

UUID	Name	Instance UUID	Power State	Provision State	Maintenance
b7dde876-354a-4688-8550-aec8f64c582c	None	a23af643-51c8-4f59-881c-77a9d5e1557f	power on	active	False
e4563ca5-2f12-4e08-9905-f770f740ad2b	None	e902ad92-f600-41ec-a525-32218be1ee11	power on	active	False
285965a9-9713-4301-8ad5-7aa3ef5dd1c2	None	f01d22ef-18a4-4f01-b4c1-1ffb6f1d9262	power on	active	False
b4dc04ac-0c69-4000-9c4d-2d82d141905f	None	869662d6-e5ae-4724-82fb-3eb9a6f74e5b	power on	active	False
036cae70-bdee-427c-987c-a6a2d8a32292	None	ab811350-d49e-4e3e-bad7-9afeb6e1d10a	power on	active	False
8570c96e-f9cd-44ff-aid8-0252bc405c24	None	7d9caa7c-4e53-4832-983c-44706def4d42	power on	active	False
af46cd81-c78e-47c5-94e3-44d9d669410c	None	211bb71c-2e72-44b1-a867-5f7babf4d4bb	power on	active	False
19260dbb-29a9-4810-b39d-85cc6e1d886f	None	b4a04e95-6624-4f13-8a14-5ef1e6742b94	power on	active	False
d4dae332-4595-43be-9b63-5a64331ea33b	None	db50a27a-4699-4fd9-9687-ec5403db3409	power on	active	False
179befe6-2510-4311-ad9f-4880454fdaff	None	47e64b48-f105-49f8-ae40-f04db9c39313	power on	active	False
ff0dadfe-e2f3-408f-b69d-01398bb9699d	None	0a1f7293-d9ee-423c-80bc-96125d29d924	power on	active	False
b59f57e3-d5e1-499a-80c1-aa0c78c9534	None	6be9ea47-39c3-4b4e-b755-a866e47b8398	power on	active	False
1132c423-7449-40b5-935a-0f989f61813f	None	19659ace-53b1-4a86-b0b8-21439aa8ab1a	None	active	True

```
[root@overcloud-cephstorage-0 ~]# ceph osd tree
ID WEIGHT  TYPE NAME                UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 171.83984 root default
-2 42.95996  host overcloud-cephstorage-0
  1  5.37000  osd.1                up 1.00000 1.00000
  4  5.37000  osd.4                up 1.00000 1.00000
  7  5.37000  osd.7                up 1.00000 1.00000
 10  5.37000  osd.10               up 1.00000 1.00000
 13  5.37000  osd.13               up 1.00000 1.00000
 16  5.37000  osd.16               up 1.00000 1.00000
 19  5.37000  osd.19               up 1.00000 1.00000
 22  5.37000  osd.22               up 1.00000 1.00000
-3 42.95996  host overcloud-cephstorage-2
  0  5.37000  osd.0                up 1.00000 1.00000
  3  5.37000  osd.3                up 1.00000 1.00000
  6  5.37000  osd.6                up 1.00000 1.00000
  9  5.37000  osd.9                up 1.00000 1.00000
 12  5.37000  osd.12               up 1.00000 1.00000
 15  5.37000  osd.15               up 1.00000 1.00000
 18  5.37000  osd.18               up 1.00000 1.00000
 21  5.37000  osd.21               up 1.00000 1.00000
-4 42.95996  host overcloud-cephstorage-1
  2  5.37000  osd.2                up 1.00000 1.00000
  5  5.37000  osd.5                up 1.00000 1.00000
  8  5.37000  osd.8                up 1.00000 1.00000
 11  5.37000  osd.11               up 1.00000 1.00000
 14  5.37000  osd.14               up 1.00000 1.00000
 17  5.37000  osd.17               up 1.00000 1.00000
 20  5.37000  osd.20               up 1.00000 1.00000
 23  5.37000  osd.23               up 1.00000 1.00000
-5 42.95996  host overcloud-cephstorage-3
 24  5.37000  osd.24               down 0 1.00000
 25  5.37000  osd.25               down 0 1.00000
 26  5.37000  osd.26               down 0 1.00000
 27  5.37000  osd.27               down 0 1.00000
 28  5.37000  osd.28               down 0 1.00000
 29  5.37000  osd.29               down 0 1.00000
 30  5.37000  osd.30               down 0 1.00000
 31  5.37000  osd.31               down 0 1.00000
```

2. Check the health of placement groups:

```
[root@overcloud-cephstorage-0 ~]# ceph pg dump_stuck stale
ok
[root@overcloud-cephstorage-0 ~]# ceph pg dump_stuck inactive
ok
[root@overcloud-cephstorage-0 ~]# ceph pg dump_stuck unclean
ok
```

3. Run Ceph PG dump to validate that the OSD's do not have any copies.

```

ceph pg dump shows the following
osdstat kbused kbavail kb hb in hb out
0 7469788 5757610256 5765080044 [1,2,4,5,6,7,8,9,10,11,13,14,15,16,17,18,19,20,21,22,23] []
1 7452016 5757628028 5765080044 [0,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,23] []
2 6672944 5758407100 5765080044 [0,1,3,4,5,6,7,8,9,10,11,12,13,15,16,18,19,20,22] []
3 5661452 5759418592 5765080044 [0,1,2,4,5,6,7,8,10,11,12,13,14,15,16,17,19,20,21,22,23] []
4 6343924 5758736120 5765080044 [0,1,2,3,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20,21,23] []
5 5863404 5759216640 5765080044 [0,1,3,4,6,7,8,9,10,12,13,16,17,18,19,21,22,23] []
6 6679832 5758400212 5765080044 [1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,19,20,21,22,23] []
7 5740692 5759339352 5765080044 [0,1,2,3,5,6,8,9,10,11,12,13,14,15,16,17,18,20,21,23] []
8 5288700 5759791344 5765080044 [0,1,2,3,4,6,7,9,10,12,13,15,16,18,19,20,21,22,23] []
9 4861044 5760219000 5765080044 [0,1,2,3,4,5,6,7,8,10,11,12,13,14,16,17,19,20,21,22,23] []
10 4673440 5760406604 5765080044 [0,1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,23] []
11 6208780 5758871264 5765080044 [0,1,3,4,6,7,9,10,12,13,15,16,18,19,20,21,22] []
12 4203928 5760876116 5765080044 [0,1,2,3,4,5,6,7,8,9,10,11,13,14,15,16,17,18,19,20,21,22,23] []
13 5214776 5759865268 5765080044 [0,1,2,3,4,5,6,8,9,10,11,12,14,15,16,17,18,20,21,22,23] []
14 3676596 5761403448 5765080044 [0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,21,22,23] []
15 5337464 5759742580 5765080044 [0,1,2,3,4,5,6,7,8,9,10,11,13,14,16,17,19,20,22,23] []
16 6046284 5759033760 5765080044 [0,2,3,4,5,6,8,9,10,11,12,13,14,15,17,18,19,20,21,22,23] []
17 5840224 5759239820 5765080044 [0,1,3,4,6,7,9,10,12,13,14,15,16,18,19,20,21,22] []
18 5759744 5759320300 5765080044 [0,1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19,20,22,23] []
19 4352956 5760727088 5765080044 [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20,21,22,23] []
20 6505292 5758574752 5765080044 [0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,21,22] []
21 6636720 5758443324 5765080044 [1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19,20,22,23] []
22 6795192 5758284852 5765080044 [0,1,2,3,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,23] []
23 6553904 5758526140 5765080044 [0,1,3,4,6,7,9,10,11,12,13,14,15,16,17,18,19,21,22] []
24 0 0 0 [] []
25 0 0 0 [] []
26 0 0 0 [] []
27 0 0 0 [] []
28 0 0 0 [] []
29 0 0 0 [] []
30 ~ ~ ~ [] []

```

This makes sure that there is nothing in osd.24 to osd.31. These are the OSD's that are part of the node that was deleted. Ceph moved all the copies from this node to other node.

Making sure that no placement groups are attached to the OSD's using `ceph pg dump` or `ceph osd stat` makes sure of data integrity. The above command confirms that all the data has been moved out of the OSD's. It is not recommended to delete a node with any placement groups residing in these OSD's. Please wait till the recovery activity is complete. Do not let the Ceph cluster reach its full ratio when removing nodes or OSD's. Removing OSD's could cause the cluster to reach full ratio and could cause data integrity issues.

#### 1. IPMI status.

As the node is switched off it is not reachable through IPMI.

```

[stack@osp7-director ~]$ ipmitool -I lanplus -H 10.22.100.8 -U admin -P
nbv12345 chassis power status
Error: Unable to establish IPMI v2 / RMCP+ session
Error: Unable to establish IPMI v2 / RMCP+ session
Error: Unable to establish IPMI v2 / RMCP+ session
Unable to get Chassis Power Status

```

#### 2. Update the driver entries to work around the issue.

```

edit vi /etc/ironic/ironic.conf
Update the enabled drivers temporarily as below
#enabled_drivers=pxe_ipmitool,pxe_ssh,pxe_drac
enabled_drivers=fake
Restart openstack-ironic-conductor
sudo service openstack-ironic-conductor restart

ironic node-update NODE_UUID replace driver=fake

The node in ironic node-list should be with provision-state=active and maintenance=false
If not
ironic node-set-provision-state NODE_UUID provide
ironic node-set-provision-state NODE_UUID active

ironic node-set-provision-state NODE_UUID deleted

```

The power status should be on, provision state as available and maintenance as false

UUID	Name	Instance UUID	Power State	Provision State	Maintenance
b7dde876-354a-4688-8550-aec8f64c582c	None	a23af643-51c8-4f59-881c-77a9d5e1557f	power on	active	False
e4563ca5-2f12-4e08-9905-f770f740ad2b	None	e902ad92-f600-41ec-a525-32218be1ee11	power on	active	False
285965a9-9713-4301-8ad5-7aa3ef5dd1c2	None	f01d22ef-18a4-4f01-b4c1-1ffb6f1d9262	power on	active	False
b4dc04ac-0c69-4000-9c4d-2d82d141905f	None	869662d6-e5ae-4724-82fb-3eb9a6f74e5b	power on	active	False
036cae70-bdee-427c-987c-a6a2d8a32292	None	ab811350-d49e-4e3e-bad7-9af6b6e1d10a	power on	active	False
8570c96e-f9cd-44ff-a1d8-0252bc405c24	None	7d9caa7c-4e53-4832-983c-44706def4d42	power on	active	False
af46cd81-c78e-47c5-94e3-44d9d669410c	None	211bb71c-2e72-44b1-a867-5f7babf4d4bb	power on	active	False
19260dbb-29a9-4810-b39d-85cc6e1d886f	None	b4a04e95-6624-4f13-8a14-5ef1e6742b94	power on	active	False
d4dae332-4595-43be-9b63-5a64331ea33b	None	db50a27a-4699-4fd9-9687-ec5403db3409	power on	active	False
179bef6e-2510-4311-ad9f-4880454fdaff	None	47e64b48-f105-49f8-ae40-f04db9c39313	power on	active	False
ff0dadfe-e2f3-408f-b69d-01398bb9699d	None	0a1f7293-d9ee-423c-80bc-96125d29d924	power on	active	False
b59f57e3-d5e1-499a-80c1-aac0c78c9534	None	6be9ea47-39c3-4b4e-b755-a866e47b8398	power on	active	False
1132c423-7449-40b5-935a-0f989f61813f	None	19659ace-53b1-4a86-b0b8-21439aa8ab1a	None	available	False

```
[stack@osp7-director ~]$ nova list
```

ID	Name	Status	Task State	Power State	Networks
47e64b48-f105-49f8-ae40-f04db9c39313	overcloud-cephstorage-0	ACTIVE	-	Running	ct1plane=10.22.110.52
0a1f7293-d9ee-423c-80bc-96125d29d924	overcloud-cephstorage-1	ACTIVE	-	Running	ct1plane=10.22.110.80
6be9ea47-39c3-4b4e-b755-a866e47b8398	overcloud-cephstorage-2	ACTIVE	-	Running	ct1plane=10.22.110.76
19659ace-53b1-4a86-b0b8-21439aa8ab1a	overcloud-cephstorage-3	ERROR	-	NOSTATE	
ab811350-d49e-4e3e-bad7-9af6b6e1d10a	overcloud-compute-0	ACTIVE	-	Running	ct1plane=10.22.110.53
211bb71c-2e72-44b1-a867-5f7babf4d4bb	overcloud-compute-1	ACTIVE	-	Running	ct1plane=10.22.110.61
db50a27a-4699-4fd9-9687-ec5403db3409	overcloud-compute-2	ACTIVE	-	Running	ct1plane=10.22.110.58
b4a04e95-6624-4f13-8a14-5ef1e6742b94	overcloud-compute-3	ACTIVE	-	Running	ct1plane=10.22.110.62
7d9caa7c-4e53-4832-983c-44706def4d42	overcloud-compute-4	ACTIVE	-	Running	ct1plane=10.22.110.56
869662d6-e5ae-4724-82fb-3eb9a6f74e5b	overcloud-compute-5	ACTIVE	-	Running	ct1plane=10.22.110.55
e902ad92-f600-41ec-a525-32218be1ee11	overcloud-controller-0	ACTIVE	-	Running	ct1plane=10.22.110.59
a23af643-51c8-4f59-881c-77a9d5e1557f	overcloud-controller-1	ACTIVE	-	Running	ct1plane=10.22.110.54
f01d22ef-18a4-4f01-b4c1-1ffb6f1d9262	overcloud-controller-2	ACTIVE	-	Running	ct1plane=10.22.110.57

3. Delete the node from nova and ironic.

Run nova service-list and identify the service id's  
Delete the service id's associated with this node as

```
nova service-delete $id
nova delete NODE_UUID
ironic node-delete NODE_UUID
Revert back the "fake" driver from ironic.conf.
edit vi /etc/ironic/ironic.conf.
enabled_drivers=pxe_ipmitool,pxe_ssh,pxe_drac
#enabled_drivers=fake
Restart ironic-conductor to pick up the drivers again.
service openstack-ironic-conductor restart
```



Storage node deletion differs from compute node deletion here. In both the cases we have deleted the nodes from UCS and OpenStack so far. However ceph entries still remain and these have to be cleaned up.

### Clean Up Ceph after Node Deletion

To clean up Ceph after a node deletion, complete the following steps:

1. Check the details from ceph health and osd tree:

```
[root@overcloud-cephstorage-0 ~]# ceph osd tree
ID WEIGHT  TYPE NAME                UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 171.83984 root default
-2 42.95996  host overcloud-cephstorage-0
  1  5.37000  osd.1                up 1.00000 1.00000
  4  5.37000  osd.4                up 1.00000 1.00000
  7  5.37000  osd.7                up 1.00000 1.00000
 10  5.37000  osd.10               up 1.00000 1.00000
 13  5.37000  osd.13               up 1.00000 1.00000
 16  5.37000  osd.16               up 1.00000 1.00000
 19  5.37000  osd.19               up 1.00000 1.00000
 22  5.37000  osd.22               up 1.00000 1.00000
-3 42.95996  host overcloud-cephstorage-2
  0  5.37000  osd.0                up 1.00000 1.00000
  3  5.37000  osd.3                up 1.00000 1.00000
  6  5.37000  osd.6                up 1.00000 1.00000
  9  5.37000  osd.9                up 1.00000 1.00000
 12  5.37000  osd.12               up 1.00000 1.00000
 15  5.37000  osd.15               up 1.00000 1.00000
 18  5.37000  osd.18               up 1.00000 1.00000
 21  5.37000  osd.21               up 1.00000 1.00000
-4 42.95996  host overcloud-cephstorage-1
  2  5.37000  osd.2                up 1.00000 1.00000
  5  5.37000  osd.5                up 1.00000 1.00000
  8  5.37000  osd.8                up 1.00000 1.00000
 11  5.37000  osd.11               up 1.00000 1.00000
 14  5.37000  osd.14               up 1.00000 1.00000
 17  5.37000  osd.17               up 1.00000 1.00000
 20  5.37000  osd.20               up 1.00000 1.00000
 23  5.37000  osd.23               up 1.00000 1.00000
-5 42.95996  host overcloud-cephstorage-3
 24  5.37000  osd.24               down 0 1.00000
 25  5.37000  osd.25               down 0 1.00000
 26  5.37000  osd.26               down 0 1.00000
 27  5.37000  osd.27               down 0 1.00000
 28  5.37000  osd.28               down 0 1.00000
 29  5.37000  osd.29               down 0 1.00000
 30  5.37000  osd.30               down 0 1.00000
 31  5.37000  osd.31               down 0 1.00000
```

```
[root@overcloud-cephstorage-0 ~]# ceph osd stat
osdmap e132: 32 osds: 24 up, 24 in
[root@overcloud-cephstorage-0 ~]#
```

2. Remove OSD's from Ceph. Change the OSD ID's to your setup and from the output of osd tree above.

```
for i in `seq 24 31`
do
ceph osd out $i
ceph osd crush remove osd.$i
ceph auth del osd.$i
ceph osd rm $i
done

osd.24 is already out.
removed item id 24 name 'osd.24' from crush map
updated
removed osd.24
osd.25 is already out.
```

```

removed item id 25 name 'osd.25' from crush map
updated
removed osd.25
osd.26 is already out.
removed item id 26 name 'osd.26' from crush map
updated
removed osd.26
osd.27 is already out.
removed item id 27 name 'osd.27' from crush map
updated
removed osd.27
osd.28 is already out.
removed item id 28 name 'osd.28' from crush map
updated
removed osd.28
osd.29 is already out.
removed item id 29 name 'osd.29' from crush map
updated
removed osd.29
osd.30 is already out.
removed item id 30 name 'osd.30' from crush map
updated
removed osd.30
osd.31 is already out.
removed item id 31 name 'osd.31' from crush map
updated
removed osd.31
[root@overcloud-cephstorage-0 ~]#

```

3. Clean up ceph crush host entries:

```

[root@overcloud-controller-0 ~]# ceph osd crush remove overcloud-
cephstorage-3

```

4. Health checks after deletion:

```

[root@overcloud-controller-0 ~]# ceph -s
cluster 3bd648c2-c09f-11e5-8fff-0025b522225f
health HEALTH_OK
monmap e1: 3 mons at {overcloud-controller-0=10.22.120.51:6789/0,
overcloud-controller-1=10.22.120.57:6789/0,overcloud-controller-2=10.22.120.53:6789/0}
election epoch 50, quorum 0,1,2 overcloud-controller-0,
overcloud-controller-2,overcloud-controller-1
osdmap e205: 24 osds: 24 up, 24 in
pgmap v62710: 1024 pgs, 4 pools, 45163 MB data, 11390 objects
133 GB used, 128 TB / 128 TB avail
1024 active+clean

```

```
[root@overcloud-controller-0 ~]# ceph osd tree
ID WEIGHT  TYPE NAME                UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 128.87988 root default
-2 42.95996  host overcloud-cephstorage-0
  1  5.37000  osd.1                  up 1.00000 1.00000
  4  5.37000  osd.4                  up 1.00000 1.00000
  7  5.37000  osd.7                  up 1.00000 1.00000
 10  5.37000  osd.10                 up 1.00000 1.00000
 13  5.37000  osd.13                 up 1.00000 1.00000
 16  5.37000  osd.16                 up 1.00000 1.00000
 19  5.37000  osd.19                 up 1.00000 1.00000
 22  5.37000  osd.22                 up 1.00000 1.00000
-3 42.95996  host overcloud-cephstorage-2
  0  5.37000  osd.0                  up 1.00000 1.00000
  3  5.37000  osd.3                  up 1.00000 1.00000
  6  5.37000  osd.6                  up 1.00000 1.00000
  9  5.37000  osd.9                  up 1.00000 1.00000
 12  5.37000  osd.12                 up 1.00000 1.00000
 15  5.37000  osd.15                 up 1.00000 1.00000
 18  5.37000  osd.18                 up 1.00000 1.00000
 21  5.37000  osd.21                 up 1.00000 1.00000
-4 42.95996  host overcloud-cephstorage-1
  2  5.37000  osd.2                  up 1.00000 1.00000
  5  5.37000  osd.5                  up 1.00000 1.00000
  8  5.37000  osd.8                  up 1.00000 1.00000
 11  5.37000  osd.11                 up 1.00000 1.00000
 14  5.37000  osd.14                 up 1.00000 1.00000
 17  5.37000  osd.17                 up 1.00000 1.00000
 20  5.37000  osd.20                 up 1.00000 1.00000
 23  5.37000  osd.23                 up 1.00000 1.00000
[root@overcloud-controller-0 ~]#
```

#### Node Addition

When the storage node has been completely removed from OpenStack and the ceph entries cleaned, a new server can be added. The procedure for adding a new storage node is same as how it was addressed earlier in [upscaling the storage pod](#).

#### HA on Undercloud Node

RHEL-OSP7 supports only one Undercloud Node as of the date this document was first published. Also in the test bed, the compute and storage nodes are natted through Undercloud node. Though this does not pose any challenges during Overcloud operation, any future heat stack or overcloud deploys could be impacted.

The following backup and recovery method has been documented on Red Hat web site for reference. This procedure has not been validated in this CVD. It is strongly recommended to test the below procedure in a test environment and document the process to restore the Undercloud node from backup. Subsequently take a backup of the Undercloud node and store the back up for an easy retrieval later in case of failures.

[https://access.redhat.com/webassets/avalon/d/Red\\_Hat\\_Enterprise\\_Linux\\_OpenStack\\_Platform-7-Back\\_Up\\_and\\_Restore\\_Red\\_Hat\\_Enterprise\\_Linux\\_OpenStack\\_Platform-en-US/Red\\_Hat\\_Enterprise\\_Linux\\_OpenStack\\_Platform-7-Back\\_Up\\_and\\_Restore\\_Red\\_Hat\\_Enterprise\\_Linux\\_OpenStack\\_Platform-en-US.pdf](https://access.redhat.com/webassets/avalon/d/Red_Hat_Enterprise_Linux_OpenStack_Platform-7-Back_Up_and_Restore_Red_Hat_Enterprise_Linux_OpenStack_Platform-en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform-7-Back_Up_and_Restore_Red_Hat_Enterprise_Linux_OpenStack_Platform-en-US.pdf)

## Hardware Failures of Blades

---

The hardware failures of blades are infrequent and happen very rarely. Cisco stands behind the customers to support in such conditions. There is also a [Return Material Authorization \(RMA\) process](#) in place. Depending on the types of failure, either the parts or the entire blade may be replaced. This section at a high level covers the types of failures that could happen on Cisco UCS blades running OpenStack and how to get the system up and running with little or no business interruption.

Before dwelling into the details, we would like to highlight that this section was validated specifically for Controller blades. The replacement of compute and storage blades are covered earlier in the High Availability section.

### Types of Failures

- CPU Failures
- Memory or DIMM Failures
- Virtual Interface Card Failures
- Motherboard Failures
- Hard Disk Failures
- Chassis Slot Issues

Any such failures happening on a blade either leads to degraded performance while the system continues to operate (like DIMM or disk Failures) or it could fail completely. In case of complete failures, OpenStack Nova and Ironic may also take them offline and there is a need to fix the errors.

**A compute node failure will impact only the VM's running on the compute node and these can be evacuated to another node.**

Ceph storage nodes are configured with replication factor of 3, and the system continues to operate though the recovery operation may cause slight degraded performance of the storage cluster.

In case of total failure of controller blades, the fencing packages will fence the failed node. You may need to fix for bug 1303698 for this. This is not included in RHEL-OSP 7.2 release and instructions are provided how to get the patches and customize the Overcloud image earlier in this document. With the fix in place, the system continues to operate in a degraded mode (performance impact while navigating through dashboard and while creating new virtual machines).

### OpenStack Dependency on Hardware

From OpenStack point of view, the following hardware variables are seeded into the system and these may have to be addressed in case of failures:

#### IPMI Address

OpenStack uses IPMI address and it powers on/off the blades with this address. These can be queried through ironic **API's** as below.

```
[stack@osp7-director ~]$ ironic node-show 65f57d30-77ea-4b65-93c1-ed70ec567c7b | grep ipmi
| driver | pxe_ipmitool
| driver_info | {u'ipmi_password': u'*****', u'ipmi_address': u'10.22.100.11',
| | u'ipmi_username': u'admin', u'deploy_kernel': u'ca9667c9-de8f-
```

## NIC's and MAC addresses

The controller ethernet interfaces and MAC addresses are available in the Local Disk of the failed blade. Hence failure cases of hard disks is also included above. Apart from this, the provisioning Interface MAC address is also stored in the Undercloud node.

```
[stack@osp7-director ~]$ ironic node-port-list 65f57d30-77ea-4b65-93c1-ed70ec567c7b
+-----+-----+
| UUID | Address |
+-----+-----+
| 31f4aab4-a08c-4ec7-8fa5-59e69c339221 | 00:25:b5:00:00:57 |
+-----+-----+
```

Retain these addresses in case of failures.

## Local Disk

The local hard disk has all the configuration information and should be available. Hence it is strongly recommended to have a pair of Local Disks in RAID-10 configuration to overcome against disk failures.



Post hardware failures, if all of the above are brought back, the system can be made operational and this is what is addressed in this section.

---

## Cisco UCS Failure Scenarios

As mentioned earlier there can be several types of failures including CPU or Memory and system may perform in a degraded fashion. Not all of these are covered in this document, but the ones which have hooks to OpenStack are covered here.

### Hard Disk Failure

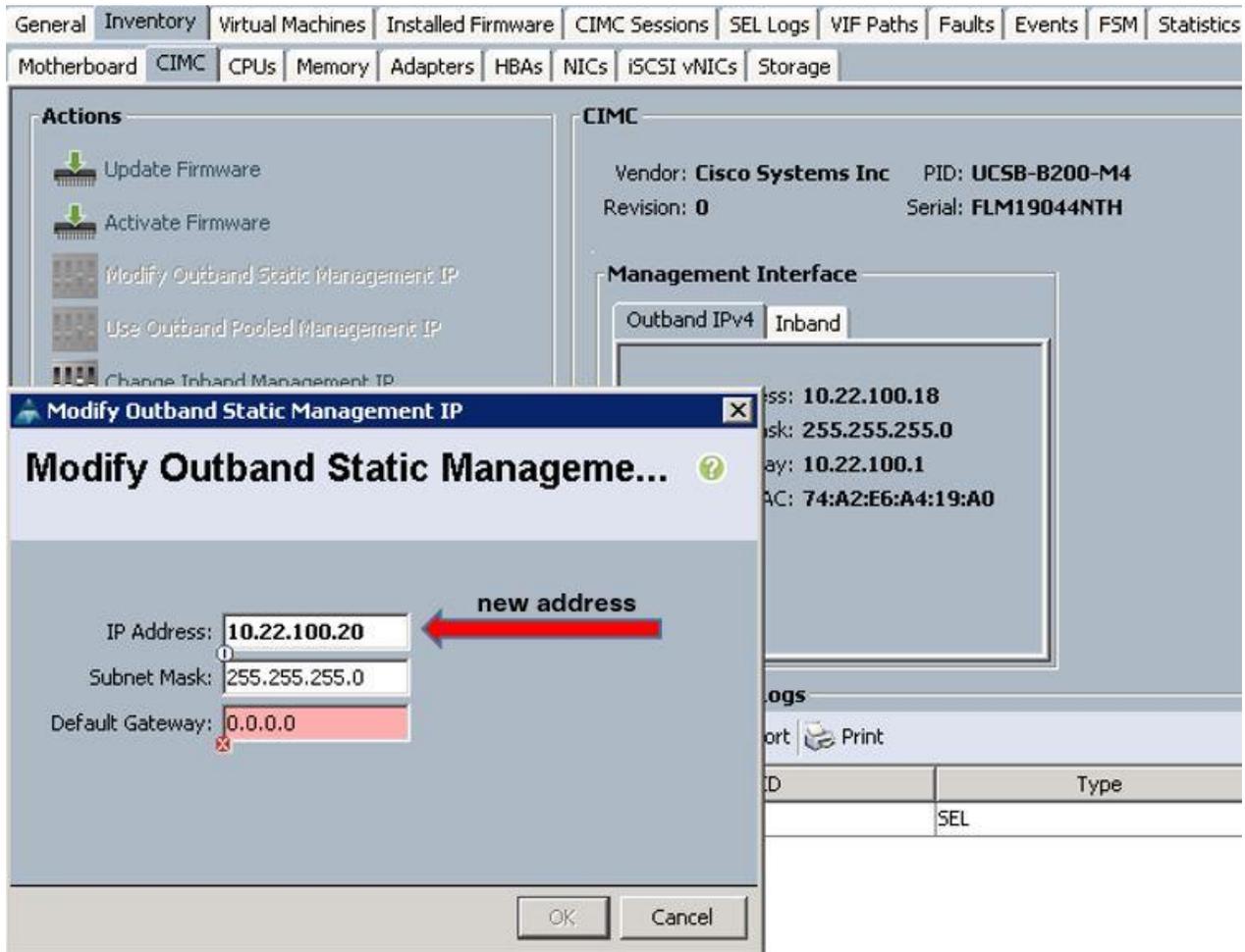
Assuming you have followed the recommendations to have RAID-10 configuration for the local disks, failure of one disk will be taken care by the RAID controller.

## Blade Replacement

In case there is a need to replace the blade, the ipmi address, MAC addresses and Local disks have to be restored. It is assumed that there is no double failure here.

### IPMI Address

The IPMI addresses are allocated from the KVM pool. When a blade fails system will hold the address until it has been decommissioned. If system is decommissioned, it will release the free IP to KVM Pool. We can allocate this free old IP to the new blade. The below figure shows how to change the IPMI address in UCS as an example.



### NIC's and MAC Addresses

Service profiles are like SIM card of a phone, that store all the hardware identity. Once the Service Profile is disassociated from the failed node and attached to the new node, all of the policies like Boot Policy and Network interfaces along with MAC addresses are available to the new blade.

### Local Disks

The two hard disks can be taken out from the failed blade and inserted into the new blade. You have to make sure that the new blade is identical and upgraded to the same firmware version as of the failed blade. The local disks have the controller binaries and the cluster configuration information. Associating the service profile will bring up all the hardware profiles on the new blade. Hence now system will be in sync from both hardware and software side and should be up and running.

## Case Study

The following case study describes the step by step process on how to replace the controller blade.

```
[stack@osp7-director2 ~]$ nova list |grep controller
| 4c23b209-094e-4156-aed9-bd6853ad3c04 | overcloud-controller-0 | ACTIVE | - | Running | ct\plane=20.7.20.45 |
| e5336e56-6f28-4f1e-87ab-b207123a9746 | overcloud-controller-1 | ACTIVE | - | Running | ct\plane=20.7.20.44 |
| 39bd3156-f73a-45d9-acda-8b9a13b333b6 | overcloud-controller-2 | ACTIVE | - | Running | ct\plane=20.7.20.65 |
[stack@osp7-director2 ~]$
```

PCS Status

```

[root@overcloud-controller-0 ~]# pcs status
Cluster name: tripleo_cluster
Last updated: Sat Feb 13 17:22:32 2016          Last change: Sat Feb 13
17:12:09 2016 by
root via cibadmin on overcloud-controller-2
Stack: corosync
Current DC: overcloud-controller-0 (version 1.1.13-10.e17-44eb2dd) -
partition with quorum
3 nodes and 115 resources configured

Online: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]

Full list of resources:

ip-172.22.219.204      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
ip-20.7.40.51         (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-1
Clone Set: haproxy-clone [haproxy]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
ip-20.7.30.51         (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
ip-20.7.20.99         (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-1
Master/Slave Set: galera-master [galera]
Masters: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
ip-20.7.10.51         (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
ip-20.7.10.52         (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-1
Master/Slave Set: redis-master [redis]
Masters: [ overcloud-controller-0 ]
Slaves: [ overcloud-controller-1 overcloud-controller-2 ]
Clone Set: mongod-clone [mongod]
Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
ucs-fence-controller (stonith:fence_cisco_ucs):      Started overcloud-
controller-2
.....
.....
.....

PCSD Status:
overcloud-controller-0: Online
overcloud-controller-1: Online
overcloud-controller-2: Online

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled

```

## Quorum and CRM Node Information

```

[root@overcloud-controller-0 ~]# corosync-quorumtool
Quorum information
-----
Date:                Sat Feb 13 17:48:45 2016
Quorum provider:    corosync_votequorum
Nodes:              3
Node ID:            3
Ring ID:            2776
Quorate:            Yes

Votequorum information
-----
Expected votes:     3
Highest expected:   3
Total votes:        3
Quorum:             2
Flags:              Quorate

Membership information
-----
    Nodeid    Votes Name
      2         1 overcloud-controller-1
      1         1 overcloud-controller-0
      3         1 overcloud-controller-2 (local)
[root@overcloud-controller-0 ~]#

[root@overcloud-controller-0 ~]# crm_node -l
3 overcloud-controller-2 member
2 overcloud-controller-1 member
1 overcloud-controller-0 member
[root@overcloud-controller-0 ~]#

```

## Tenants Availability and Checks

Tenants inventory for the Controller failure test.

```

[stack@osp7-director2 ~]$ nova list --all-tenants |grep tenant320_120_inst8
| 20c1b528-d467-4a5a-bf96-76c49c91557f | tenant320_120_inst8 |
b69d957c3dc8400093644664d8f2082f | ACTIVE | - | Running |
tenant320-120=10.2.120.10, 10.22.160.38 |
[stack@osp7-director2 ~]$
[stack@osp7-director2 ~]$
[stack@osp7-director2 ~]$ nova list --all-tenants |grep
tenant320_170_inst19
| 19917b9f-2581-4e53-ace0-87381d4dad8a | tenant320_170_inst19 |
b69d957c3dc8400093644664d8f2082f | ACTIVE | - | Running |
tenant320-170=10.2.170.11, 10.22.160.49 |
[stack@osp7-director2 ~]$

```



Make sure the VMs can ping East to West traffic and South to North traffic.

---

Pinging from tenant320\_170\_inst19 to tenant320\_120\_inst8 10.2.120.10 & 10.22.160.38

```

[root@tenant320-170-inst19 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1400
    inet 10.2.170.11 netmask 255.255.255.0 broadcast 10.2.170.255
    inet6 fe80::f816:3eff:fe62:22e0 prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:62:22:e0 txqueuelen 1000 (Ethernet)
    RX packets 975 bytes 95394 (93.1 KiB)
    RX errors 0 dropped 11 overruns 0 frame 0
    TX packets 903 bytes 83768 (81.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 0 (Local Loopback)
    RX packets 10 bytes 756 (756.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 756 (756.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@tenant320-170-inst19 ~]#
[root@tenant320-170-inst19 ~]# ping 10.2.120.10
PING 10.2.120.10 (10.2.120.10) 56(84) bytes of data:
64 bytes from 10.2.120.10: icmp_seq=1 ttl=63 time=0.553 ms
64 bytes from 10.2.120.10: icmp_seq=2 ttl=63 time=0.283 ms
64 bytes from 10.2.120.10: icmp_seq=3 ttl=63 time=0.279 ms
64 bytes from 10.2.120.10: icmp_seq=4 ttl=63 time=0.230 ms
^C
--- 10.2.120.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.230/0.336/0.553/0.127 ms
[root@tenant320-170-inst19 ~]# ping 10.22.160.38
PING 10.22.160.38 (10.22.160.38) 56(84) bytes of data:
64 bytes from 10.22.160.38: icmp_seq=1 ttl=63 time=0.684 ms
64 bytes from 10.22.160.38: icmp_seq=2 ttl=63 time=0.237 ms
64 bytes from 10.22.160.38: icmp_seq=3 ttl=63 time=0.247 ms
64 bytes from 10.22.160.38: icmp_seq=4 ttl=63 time=0.258 ms
^C
--- 10.22.160.38 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.237/0.356/0.684/0.190 ms
[root@tenant320-170-inst19 ~]# _

```

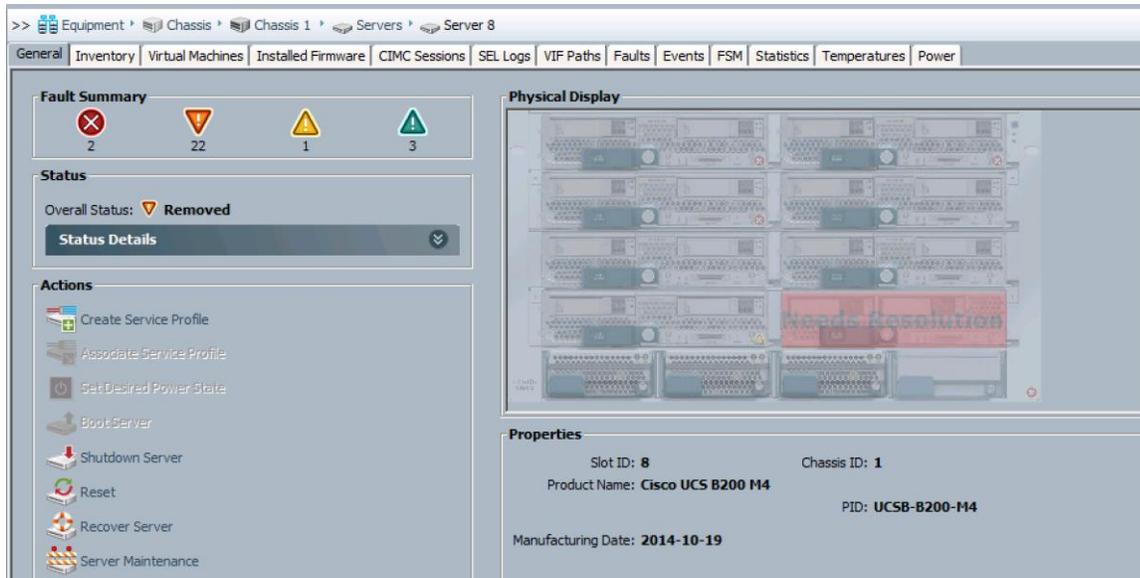
## Insert the New Blade into the Chassis

Insert the new blade into the chassis.

## Fault Injection

Identify the overcloud-controller-2 and UCS Service Profile mapping from /etc/neutron/plugin.ini on any other controller node.

## Remove the Blade from the Chassis



```
[root@overcloud-controller-0 ~]# crm_node -l
3 overcloud-controller-2 lost
2 overcloud-controller-1 member
1 overcloud-controller-0 member
[root@overcloud-controller-0 ~]#
[root@overcloud-controller-0 ~]# pcs status
Cluster name: tripleo_cluster
Last updated: Sat Feb 13 18:07:44 2016      Last change: Sat Feb 13 17:12:09
2016 by root via cibadmin on overcloud-controller-2
Stack: corosync
Current DC: overcloud-controller-0 (version 1.1.13-10.e17-44eb2dd) - partition
with quorum
3 nodes and 115 resources configured
```

```
Online: [ overcloud-controller-0 overcloud-controller-1 ]
OFFLINE: [ overcloud-controller-2 ]
```

Full list of resources:

```
ip-172.22.219.204 (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
ip-20.7.40.51 (ocf::heartbeat:IPaddr2):      Started overcloud-controller-1
Clone Set: haproxy-clone [haproxy]
Started: [ overcloud-controller-0 overcloud-controller-1 ]
Stopped: [ overcloud-controller-2 ]
ip-20.7.30.51 (ocf::heartbeat:IPaddr2):      Started overcloud-controller-0
ip-20.7.20.99 (ocf::heartbeat:IPaddr2):      Started overcloud-controller-1
Master/Slave Set: galera-master [galera]
Masters: [ overcloud-controller-0 overcloud-controller-1 ]
Stopped: [ overcloud-controller-2 ]
ip-20.7.10.51 (ocf::heartbeat:IPaddr2):      Started overcloud-controller-0
ip-20.7.10.52 (ocf::heartbeat:IPaddr2):      Started overcloud-controller-1
Master/Slave Set: redis-master [redis]
Masters: [ overcloud-controller-0 ]
Slaves: [ overcloud-controller-1 ]
Stopped: [ overcloud-controller-2 ]
```

```
Clone Set: mongod-clone [mongod]
Started: [ overcloud-controller-0 overcloud-controller-1 ]
Stopped: [ overcloud-controller-2 ]
```

```
[root@overcloud-controller-0 ~]# corosync-quorumtool
Quorum information
```

```
-----
Date:                Sat Feb 13 18:16:20 2016
Quorum provider:     corosync_votequorum
Nodes:               2
Node ID:             1
Ring ID:             2780
Quorate:             Yes
```

```
Votequorum information
```

```
-----
Expected votes: 3
Highest expected: 3
Total votes: 2
Quorum: 2
Flags:               Quorate
```

```
Membership information
```

```
-----
Nodeid      Votes Name
    2          1 overcloud-controller-1
    1          1 overcloud-controller-0 (local)
[root@overcloud-controller-0 ~]#
```



When the blade is removed from the chassis, pcs status shows resources as Unclean until the operation interval. This is configured as one minute. Hence wait for 1-2 minutes, make sure that pcs status only shows stopped as above and not unclean or unmanaged and then move forward.

## Health Checks

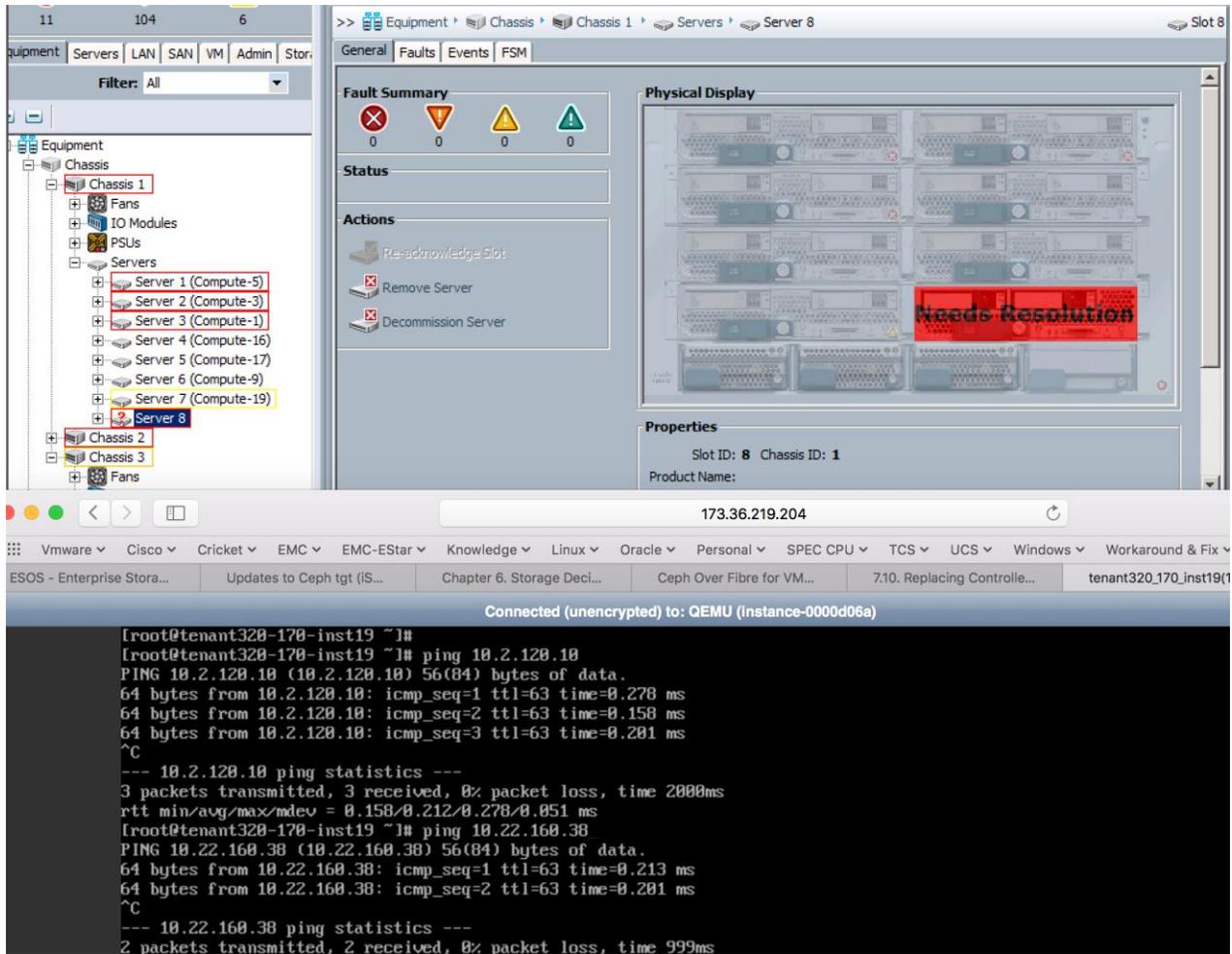
### Nova and Ironic Status

After 15-20mins... the nova status changed to NOSTATE and Ironic power state changed to NONE and the node was under maintenance mode.

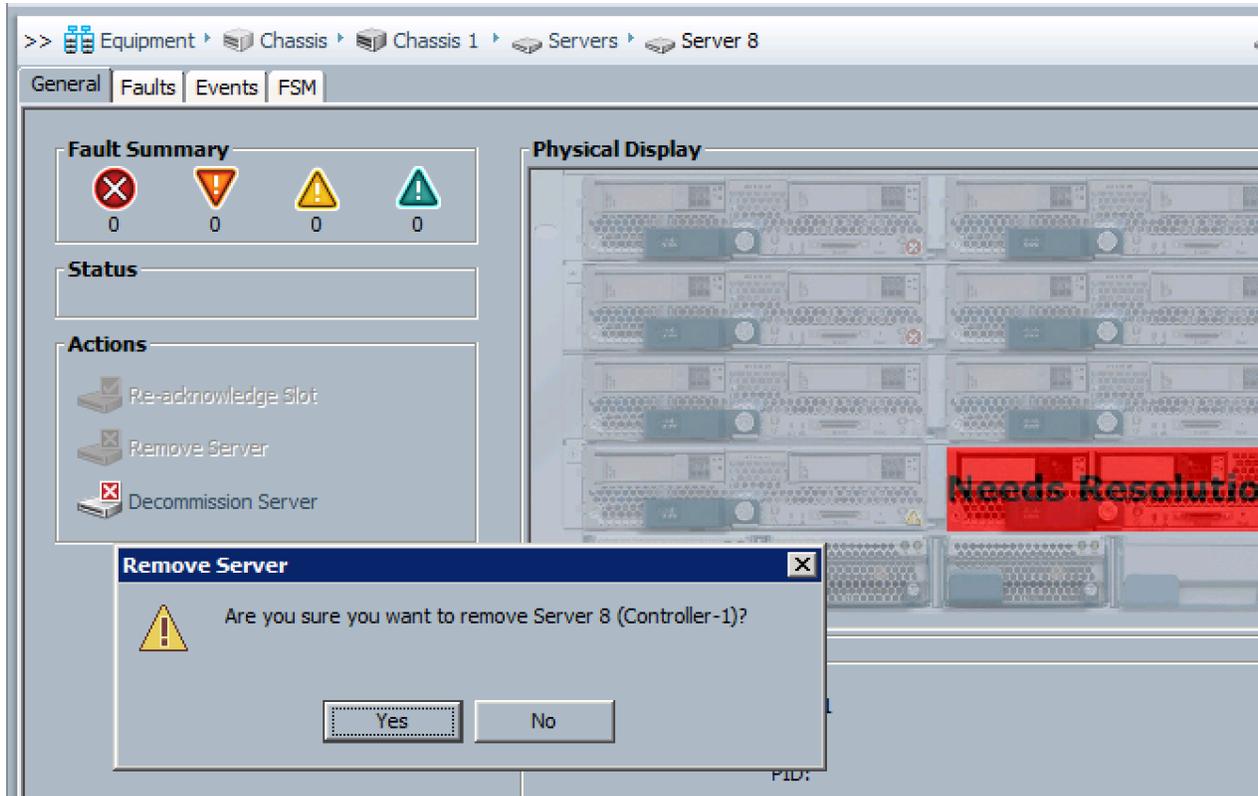
```
[stack@osp7-director2 ~]$ nova list |grep controller
| 4c23b209-094e-4156-aed9-bd6853ad3c04 | overcloud-controller-0 | ACTIVE |
- | Running | ctlplane=20.7.20.45 |
| e5336e56-6f28-4f1e-87ab-b207123a9746 | overcloud-controller-1 | ACTIVE |
- | Running | ctlplane=20.7.20.44 |
| 39bd3156-f73a-45d9-acda-8b9a13b333b6 | overcloud-controller-2 | ACTIVE |
- | NOSTATE | ctlplane=20.7.20.65 |
[stack@osp7-director2 ~]$
[stack@osp7-director2 ~]$
[stack@osp7-director2 ~]$ ironic node-list |grep 39bd3156-f73a-45d9-acda-
8b9a13b333b6
| ff1bc3c8-fcba-409f-9115-5e10d6e49ab4 | None | 39bd3156-f73a-45d9-acda-
8b9a13b333b6 | None | active | True |
[stack@osp7-director2 ~]$
[stack@osp7-director2 ~]$
```

### Tenant and VM's Status

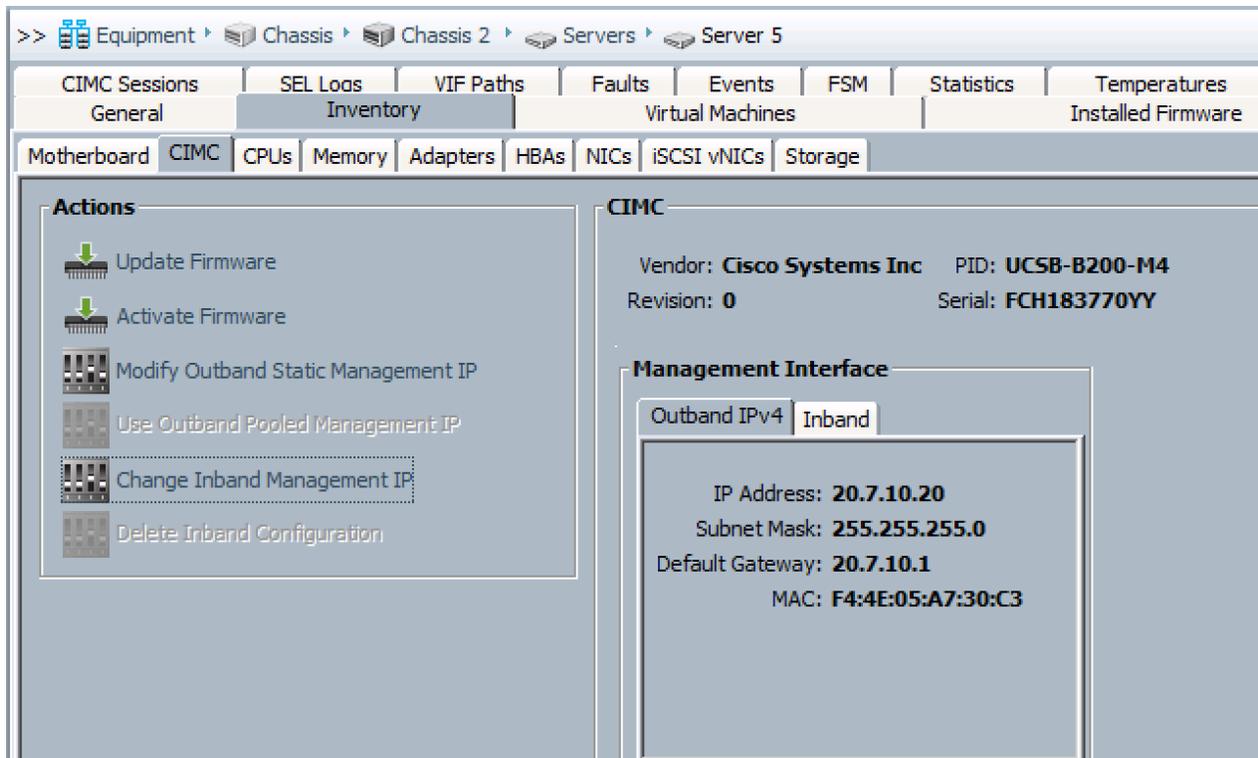
Make sure that you can login to dashboard, create new VM's and North-South and East-West traffic between VM's is uninterrupted. You may observe slowness in creating the VM's.

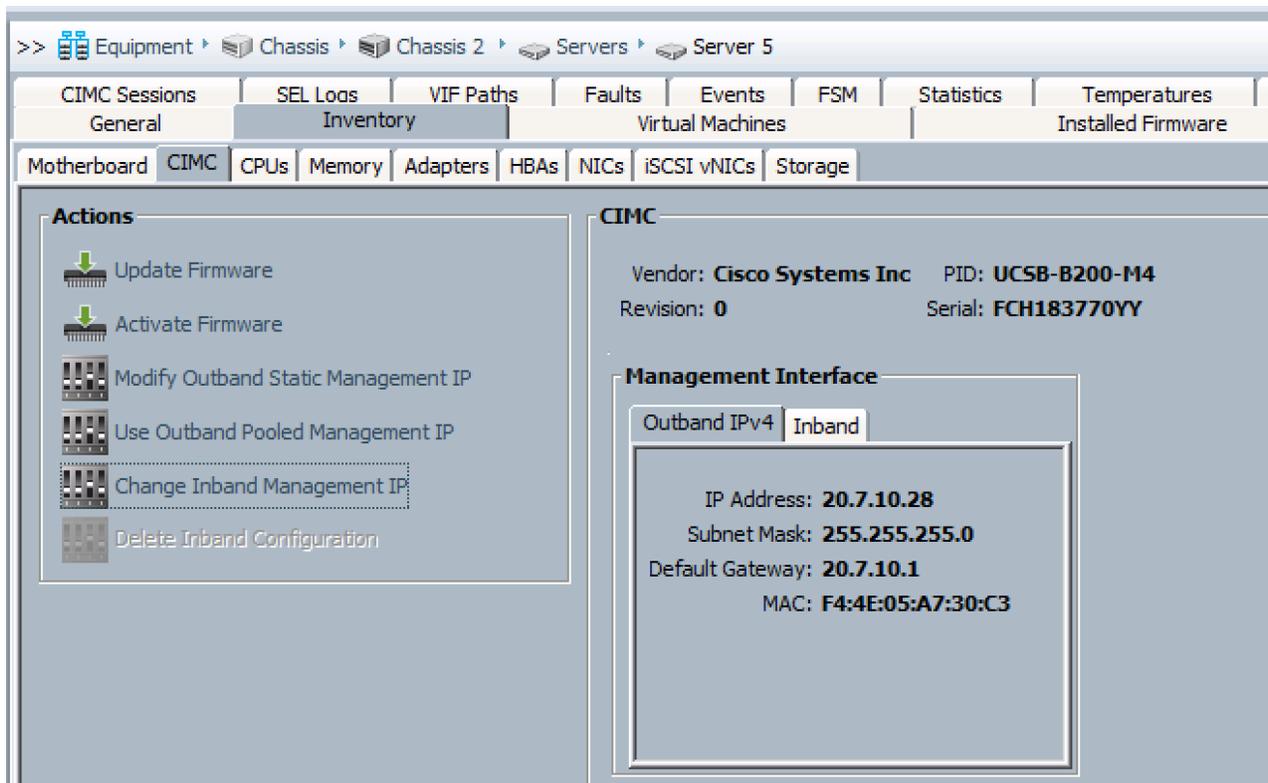
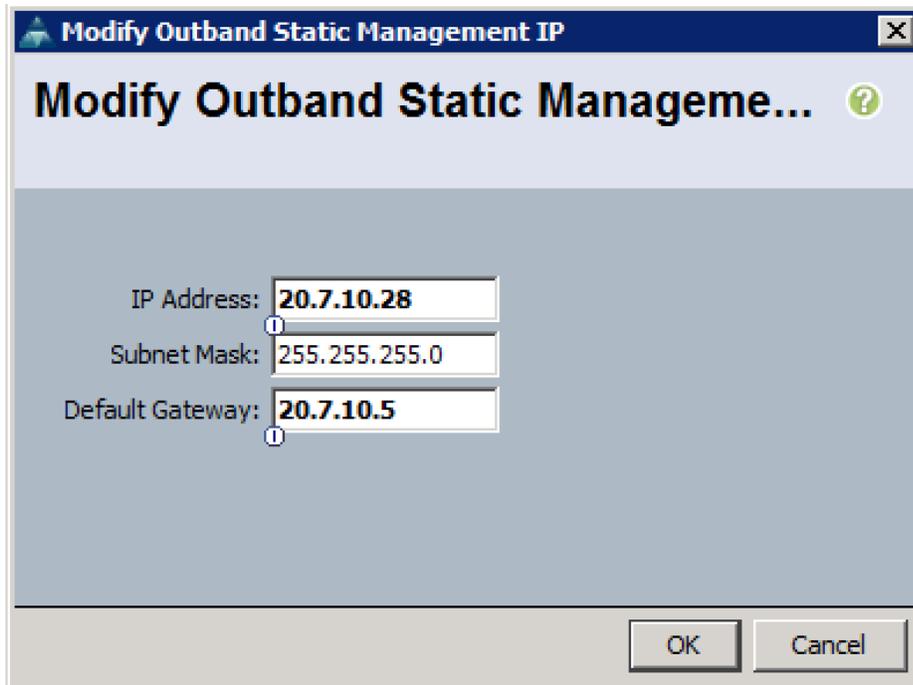


## Remove Failed Blade from Inventory



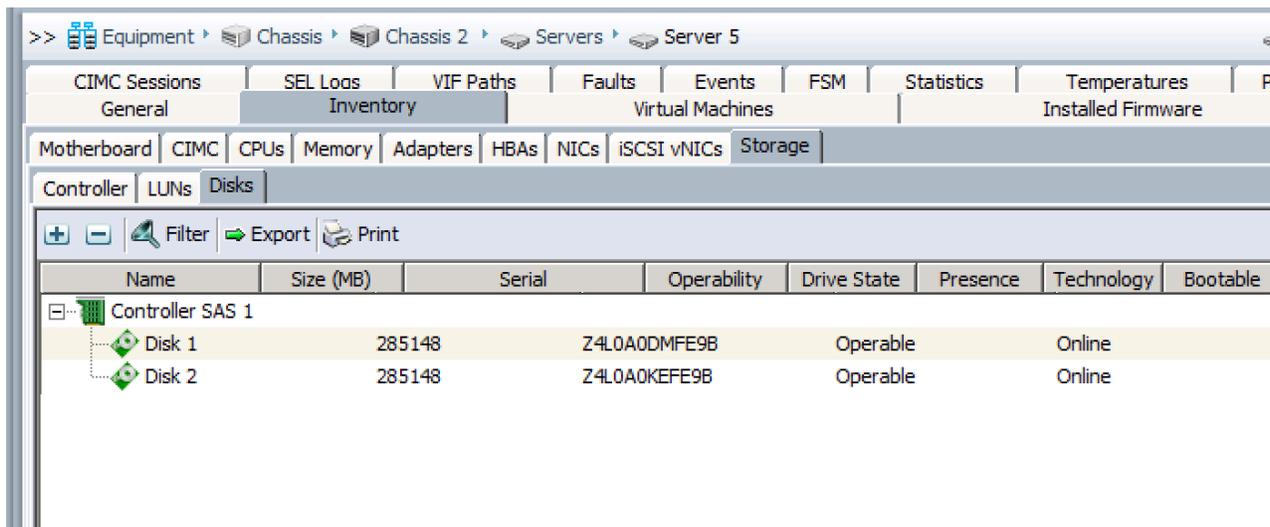
## Change IPMI Address





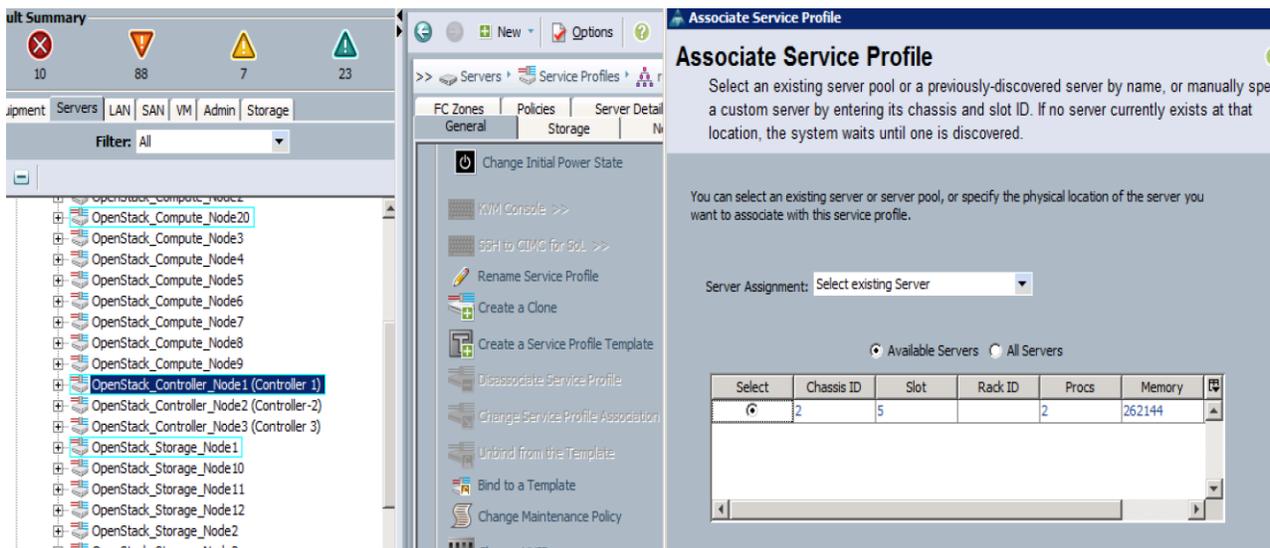
## Insert Old Disks

Remove the boot disks from the failed blade and insert them into the new blade. Make sure that the old disks show up in the server inventory.

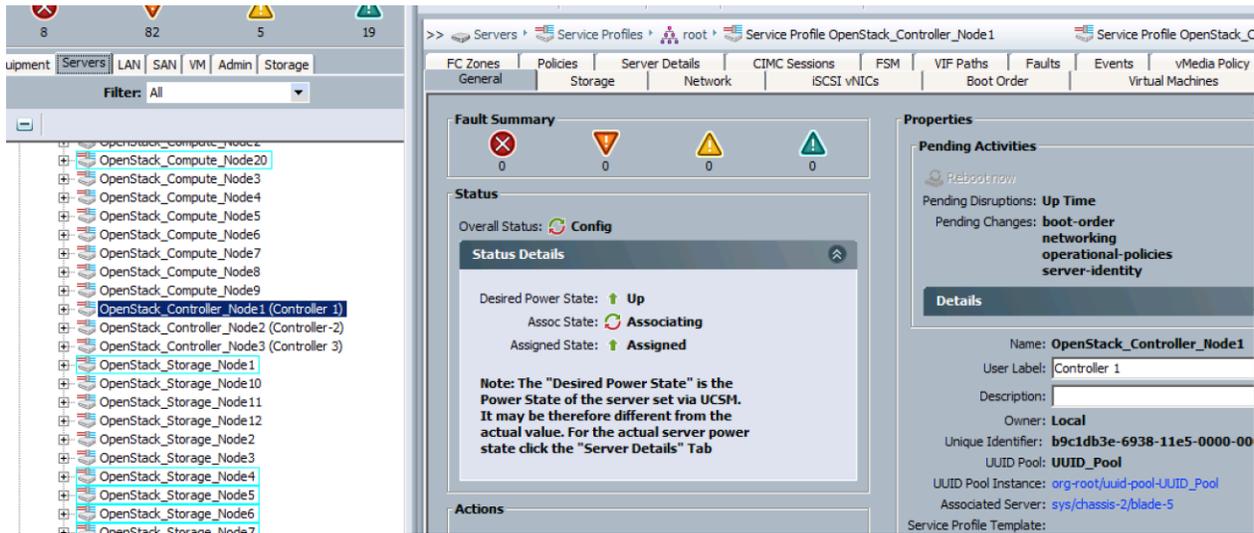


## Associate Service Profile

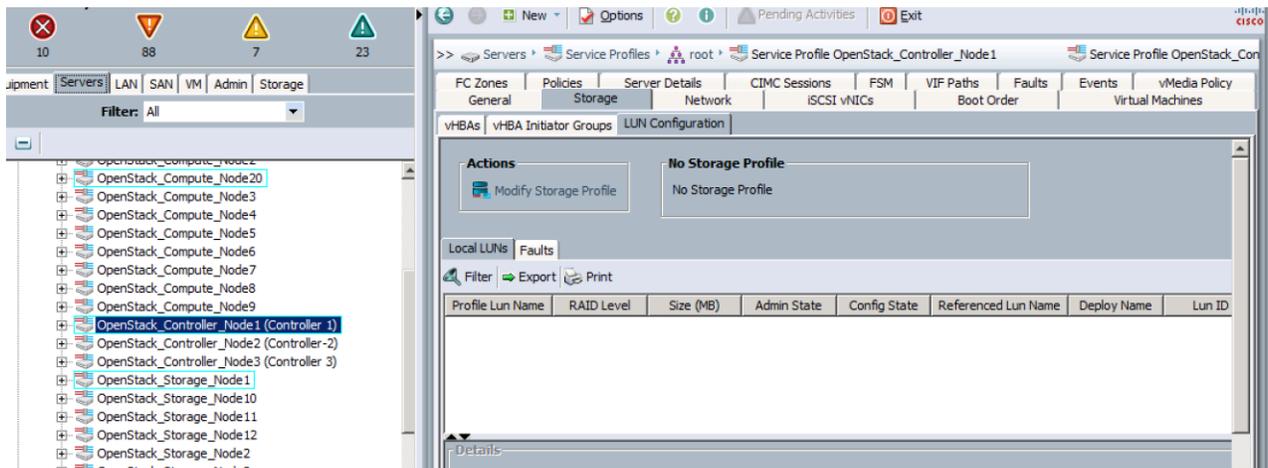
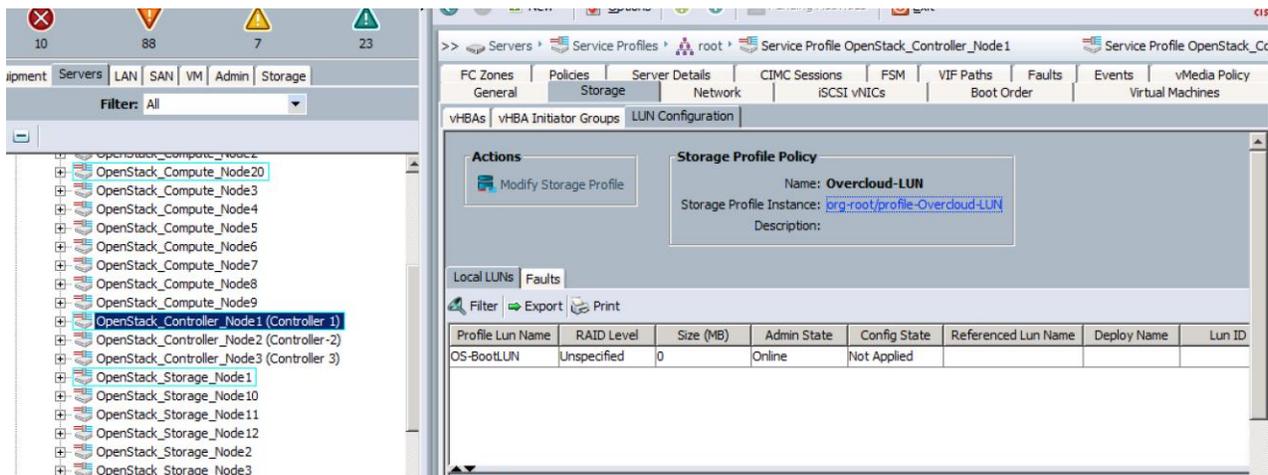
Associate the existing service profile that was disassociated earlier from the failed blade to the new or replacement blade.



Make sure that Config is in progress and monitor the status in FSM tab of the server.

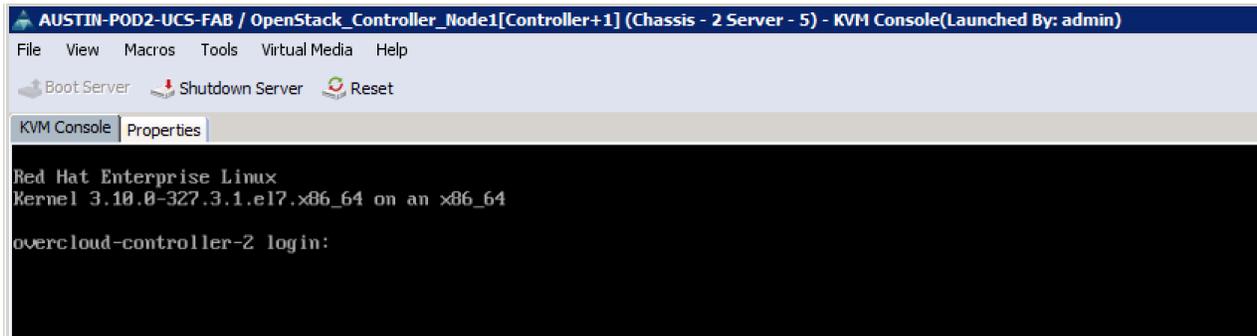


In rare occasions the existing storage policy may not let you associate the new blade. As the service profile is already unbound from the template, you may remove the storage profile from the service profile, reboot the server and then attach back the storage profile to this service profile.



## Reboot the Server

The association should boot up the server based on the desired power state, otherwise boot it up. It should show you the login prompt as below.



## Post Replacement Steps

Even though the server is up and running, you may need the following steps to let the server join back the cluster again.

```
[stack@osp7-director2 ~]$ nova list |grep controller
| 4c23b209-094e-4156-aed9-bd6853ad3c04 | overcloud-controller-0 | ACTIVE |
- | Running | ctlplane=20.7.20.45 |
| e5336e56-6f28-4f1e-87ab-b207123a9746 | overcloud-controller-1 | ACTIVE |
- | Running | ctlplane=20.7.20.44 |
| 39bd3156-f73a-45d9-acda-8b9a13b333b6 | overcloud-controller-2 | ACTIVE |
- | NOSTATE | ctlplane=20.7.20.65 |
[stack@osp7-director2 ~]$
[stack@osp7-director2 ~]$ ironic node-list |grep 39bd3156-f73a-45d9-acda-
8b9a13b333b6
| ff1bc3c8-fcba-409f-9115-5e10d6e49ab4 | None | 39bd3156-f73a-45d9-acda-
8b9a13b333b6 | None | active | True |
```

The server is in maintenance mode and in 'NOSTATE' mode.

Use nova and ironic commands to bring them to normal state.

```
[stack@osp7-director2 ~]$ ironic node-set-power-state ff1bc3c8-fcba-409f-
9115-5e10d6e49ab4 on
[stack@osp7-director2 ~]$ ironic node-set-maintenance ff1bc3c8-fcba-409f-
9115-5e10d6e49ab4 false
[stack@osp7-director2 ~]$ ironic node-list |grep 39bd3156-f73a-45d9-acda-
8b9a13b333b6
| ff1bc3c8-fcba-409f-9115-5e10d6e49ab4 | None | 39bd3156-f73a-45d9-acda-
8b9a13b333b6 | power on | active | False |

[stack@osp7-director2 ~]$ nova reset-state --active 39bd3156-f73a-45d9-
acda-8b9a13b333b6
[stack@osp7-director2 ~]$
```

It may take 10 minutes before the NOVA state is changed to running state.

```
[stack@osp7-director2 ~]$
[stack@osp7-director2 ~]$ nova list |grep controller
| 4c23b209-094e-4156-aed9-bd6853ad3c04 | overcloud-controller-0 | ACTIVE |
- | Running | ctlplane=20.7.20.45 |
| e5336e56-6f28-4f1e-87ab-b207123a9746 | overcloud-controller-1 | ACTIVE |
- | Running | ctlplane=20.7.20.44 |
| 39bd3156-f73a-45d9-acda-8b9a13b333b6 | overcloud-controller-2 | ACTIVE |
- | Running | ctlplane=20.7.20.65 |
[stack@osp7-director2 ~]$
```

## Health Checks Post Replacement

Log into any one of the controllers and check the status from pacemaker. If any services observed to be down, you may restart them with pcs resource cleanup.

```
[root@overcloud-controller-0 ~]# corosync-quorumtool -s
Quorum information
-----
Date:                Sat Feb 13 18:53:20 2016
Quorum provider:    corosync_votequorum
Nodes:              3
Node ID:            1
Ring ID:            2784
Quorate:            Yes

Votequorum information
-----
Expected votes:     3
Highest expected:   3
Total votes:      3
Quorum:             2
Flags:              Quorate

Membership information
-----
    Nodeid      Votes Name
      2          1 overcloud-controller-1
      1          1 overcloud-controller-0 (local)
      3          1 overcloud-controller-2
[root@overcloud-controller-0 ~]#
```

```
[root@overcloud-controller-0 ~]# crm_node -l
3 overcloud-controller-2 member
2 overcloud-controller-1 member
1 overcloud-controller-0 member
```

```
[root@overcloud-controller-0 ~]# pcs resource cleanup
Waiting for 1 replies from the CRMD. OK
```

```
[root@overcloud-controller-0 ~]#
[root@overcloud-controller-0 ~]# pcs status
Cluster name: tripleo_cluster
Last updated: Sat Feb 13 18:54:41 2016          Last change: Sat Feb 13
17:12:09 2016 by root via cibadmin on overcloud-controller-2
```

```

Stack: corosync
Current DC: overcloud-controller-0 (version 1.1.13-10.el7-44eb2dd) -
partition with quorum
3 nodes and 115 resources configured

Online: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]

Full list of resources:

ip-172.22.219.204      (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
ip-20.7.40.51         (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-1
Clone Set: haproxy-clone [haproxy]
  Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
ip-20.7.30.51         (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
ip-20.7.20.99         (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-1
Master/Slave Set: galera-master [galera]
  Masters: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
ip-20.7.10.51         (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-0
ip-20.7.10.52         (ocf::heartbeat:IPaddr2):      Started overcloud-
controller-1
Master/Slave Set: redis-master [redis]
  Masters: [ overcloud-controller-0 ]
  Slaves: [ overcloud-controller-1 overcloud-controller-2 ]
Clone Set: mongod-clone [mongod]
  Started: [ overcloud-controller-0 overcloud-controller-1 overcloud-
controller-2 ]
.....
.....
.....

ucs-fence-controller  (stonith:fence_cisco_ucs):      Started overcloud-
controller-2
PCSD Status:
  overcloud-controller-0: Online
  overcloud-controller-1: Online
  overcloud-controller-2: Online

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
[root@overcloud-controller-0 ~]# pcs status | grep stop
[root@overcloud-controller-0 ~]#

```

## Frequently Asked Questions

---

### Cisco Unified Computing System

- Can we use IO Modules 2208 instead of IOM 2204 as shown in the topology diagram?

Yes, both IOM 2204 and IOM 2208 are supported. For more details refer the design guide here.

- When should we use C240M4S and when C240M4L for storage servers?

This boils down as a design question and depends on the requirement. The C240M4 SFF, small form factor offers more spindles and hence higher IOPS with reasonable bandwidth capacity. The C240M4 LFF, the large form factor has a higher storage capacity but may not be as good as SFF on total IOPS per node. Validation has been done and the performance metrics provided that should help you choose the right hardware.

- Can I use Cisco UCS Sub-Orgs?

The current release of UCSM plugin does not support Sub-Orgs. We are working on this and will update this in the next release whenever this functionality is available.

- Can I use different hardware like Cisco M3 blades and different VIC adapters in the solution?

Cisco hardware higher than the version in the BOM are supported. While lower versions may still work, they have not been validated.

- How many chassis or blades and servers can I scale horizontally?

The validation was done with 3 fully loaded chassis with blades and 12 x C240M4L storage nodes for ceph. From hardware point of view, it could be the port limits and you may have to go with 96 ports Fabric Interconnects or Nexus switches.

However as number of blades increase the controller and neutron activity increases. Validation was done only with 3 controllers.

- Why aren't the internal SSD drives for Ceph storage nodes included in the BOM?

Current versions of ironic cannot identify internal drives as the ordering would vary from system to system. You may notice that you are using only disks attached to the RAID controller, which allows us to enforce a certain disk ordering. The internal drives are not connected to this RAID controller and thus would be unordered.

- How can I connect my openstack to an existing Ceph installation?

Please refer Red Hat documentation in <https://access.redhat.com/articles/1994713>.

### OpenStack

- My network topology differs from what mentioned in this document. What changes I need to do to the configuration?

The network topology verified in the configuration is included in the Appendix A. There were limited IP's and the floating network was used. It is not necessary to have the same settings. However you may have to change yaml files accordingly and tweaks may be necessary. Please refer Red Hat documentation on how to accommodate these changes in the template files.

- Updating yaml files is error prone. A simple white space or tab causing issues. What should I do?

It is recommended to use online parser followed by running network-validator before running Overcloud. Please look at Overcloud Install section for details.

- Do you have specific recommendations for Live Migration?

Live migration was attempted both in tunneling and converged modes. While the former is more secure because of encryption, the latter is performant. Please refer Live Migration section on changes needed in nova.conf to accommodate these.

- Why version lock directives have been delegated in this document?

OpenStack is continuously updated and changes in binaries and configurations go neck and neck. The purpose of providing lock file is to lock and provide binaries as close as possible to the validated design. This ensures consistency with minimal deviations from the validate design and adoption of configuration files like the yaml files. You can always install a higher version than mentioned but the specifics needed on configuration files may vary and/or some of the validations that were done in this document may have to be redone to avoid any regressions.

## Troubleshooting

---

A few troubleshooting areas that may help while installing RHEL-OSP 7 on Cisco UCS servers is presented here. Troubleshooting in Openstack is exhaustive and this section limits what helped in debugging the install. Necessary links as needed are provided too. This is only an effort to help the readers to narrow down the issues. It is assumed that the reader has followed the pre and post install steps mentioned earlier.

### Cisco Unified Computing System

- The provisioning interface should be [enabled as native](#) across all the blades and rack servers for successful introspection and Overcloud deploy.
- In case you are attempting a repeat of full deployment it is recommended to [re-initialize the boot luns](#). The wipe\_disk.yaml will work on storage partitions after OS is installed but not for the boot luns.
- The [native flag for external network](#) shouldn't be enabled on Overcloud nodes as observed on the test bed.
- Specify the [PCI order](#) for network interfaces. This ensures that they are enumerated in the same way as specified in the templates.
- In case of using updating templates make sure that the service profiles are unbound from the service profile template for successful operation of UCS Manager Plugin.
- **Before applying service profiles, you should make sure that all the disks are in 'Unconfigured Good' status.** The storage profile, that is attached to these service profiles will then successfully get applied and then will make the boot lun in operable mode.

### Undercloud Install

Undercloud install observed to be straight forward and very few issues observed. Mostly these were human mistakes like typos in the configuration file.

- Make sure that the server is registered with Red Hat Content Delivery Network for downloading the packages. In case the server is behind proxy, update /etc/rhsm/rhsm.conf file with appropriate proxy server values.
- Double check the entries in Undercloud configuration file. Provide enough room for discovery\_iprange and dhcp start/end, also considering the future expansion or upscaling of the servers later. Most of these parameters are explained in the sample file provided in /usr/share.
- Leave the value of undercloud\_debug=true as default to check for failures. The log file install-undercloud.log is created as part of Undercloud install in /home/stack/.instack. This will be handy to browse through on issues encountered during the install.
- A repeat of Undercloud install preferably has to be done in a cleaner environment after reinstalling the base operating system.

## Introspection

Failure of introspecting the nodes can be by many. Make sure that you have verified all the [post undercloud](#) and [pre-introspection](#) steps mentioned earlier in this document.

- A correct value of ipmi and mac addresses and powering on/off with ipmitool as mentioned earlier in this document should isolate the issues. Check with `ironic node-list` and `ironic node-show` to ensure that the registered values are correct.
- The boot luns configured in UCS through storage profile should be in available state before starting introspection. The size of the lun specified in the `instack.json` file should be equal or less than the size of the lun seen in UCS.

The best way to debug introspection failures is to open KVM console on the server and check for issues. The below screenshot warns something wrong in my `instack` configuration file.

```

///lib/dracut/hooks/pre-mount/50-init.sh@468(source): BENCH_ARG=--benchmark
///lib/dracut/hooks/pre-mount/50-init.sh@471(source): step 'Running discovery'
///lib/dracut/hooks/pre-mount/50-init.sh@130(step): echo '#####
#####
#####
///lib/dracut/hooks/pre-mount/50-init.sh@131(step): echo 'Running discovery'
Running discovery
///lib/dracut/hooks/pre-mount/50-init.sh@132(step): echo '#####
#####
#####
///lib/dracut/hooks/pre-mount/50-init.sh@473(source): ironic-discoverd-ramdisk -
-use-hardware-detect --benchmark --bootif 00:25:b5:00:00:17 -L /run/initramfs/rd
sosreport.txt -L /log http://10.22.110.26:5050/v1/continue
WARNING: log file /run/initramfs/rdsosreport.txt does not exist
ERROR: discoverd error 404: Could not find a node for attributes {'bmc_address':
 u'0.0.0.0', 'mac': [u'00:25:b5:00:00:17']}
ERROR: 404 Client Error: NOT FOUND when calling to discoverd
///lib/dracut/hooks/pre-mount/50-init.sh@475(source): give_up 'Failed to discove
r hardware'
///lib/dracut/hooks/pre-mount/50-init.sh@144(give_up): log 'Failed to discover h
ardware'
///lib/dracut/hooks/pre-mount/50-init.sh@136(log): echo 'Failed to discover hard
ware'
Failed to discover hardware

```

A successful introspection is shown below:

```
//lib/dracut/hooks/pre-mount/50-init.sh@132(step): echo '#####
#####'
#####
//lib/dracut/hooks/pre-mount/50-init.sh@473(source): ironic-discoverd-ramdisk -
use-hardware-detect --benchmark --bootif 00:25:b5:00:00:0f -L /run/initramfs/rds
osreport.txt -L /log http://10.22.110.26:5050/v1/continue
 87.854407] ipmi device interface
 87.994441] sd 0:2:0:0: Attached scsi generic sg0 type 0
ARNING: log file /run/initramfs/rdsosreport.txt does not exist
//lib/dracut/hooks/pre-mount/50-init.sh@477(source): case "$ONSUCCESS" in
//lib/dracut/hooks/pre-mount/50-init.sh@483(source): log 'Automatic poweroff as
required by ONSUCCESS'
//lib/dracut/hooks/pre-mount/50-init.sh@136(log): echo 'Automatic poweroff as r
quired by ONSUCCESS'
Automatic poweroff as required by ONSUCCESS
//lib/dracut/hooks/pre-mount/50-init.sh@484(source): do_halt
//lib/dracut/hooks/pre-mount/50-init.sh@119(do_halt): echo 'Node is now discove
ed? Halting...'
Node is now discovered? Halting...
//lib/dracut/hooks/pre-mount/50-init.sh@121(do_halt): sleep 5
//lib/dracut/hooks/pre-mount/50-init.sh@122(do_halt): poweroff -f
owering off.
 287.879709] ACPI: Preparing to enter system sleep state S5
 287.886340] Power down.
```

- In case system takes you to the shell prompt and dumps /run/initramfs/sosreport.txt provides some insight as well.
- dnsmasq is the dhcp process that pxe uses to discover. Within the provisioning subnet configured you should have only one dhcp process or this dnsmasq process running on the Undercloud node. Any overlap will cause discovery failures.
- **Running 'sudo -u journalctl -u openstack-ironic-discoverd -u openstack-ironic-discoverd-dnsmasq** will show issues encountered by discovered and dnsmasq.
- **Monitoring introspection with 'openstack bare metal introspection bulk status'** will show if any few servers have failed.
- At times if the status of the node(s) becomes available, you may have to update the status to manageable with ironic API before running introspection.
- The [default value](#) of introspection is 60 minutes. This may have to be changed as mentioned earlier in case introspection is taking longer time.

### Cleaning Up Failed Introspection

Use the below method only to clean up the full install of introspection. Also this was tested with the current release and there could be changes in the future release from Red Hat and/or Openstack community. Exercise caution before attempting.

```
for i in $(ironic node-list | awk ' /power/ { print $2 } ' );
do
    ironic node-set-power-state $i off
    ironic node-delete $i
done
sleep 30
```

```
sudo rm /var/lib/ironic-discoverd/discoverd.sqlite # must be deleted as root
ls -al /var/lib/ironic-discoverd/discoverd.sqlite
sudo systemctl restart openstack-ironic-discoverd
sudo systemctl status openstack-ironic-discoverd
ironic node-list
```

## Updating Incorrect MAC or IPMI Addresses

### Mac Address

```
ironic node-port-list <node uuid>
ironic port-update <node uuid> replace address <new mac address>
[stack@osp7-director ~]$ ironic node-port-list b7dde876-354a-4688-8550-aec8f64c582c
+-----+-----+
| UUID                                     | Address |
+-----+-----+
| 78a0dd36-cbf1-447c-8c42-4978438e40e9 | 00:25:b5:00:00:6c |
+-----+-----+
[stack@osp7-director ~]$ ironic port-update b7dde876-354a-4688-8550-aec8f64c582c
replace address <new mac address>
```

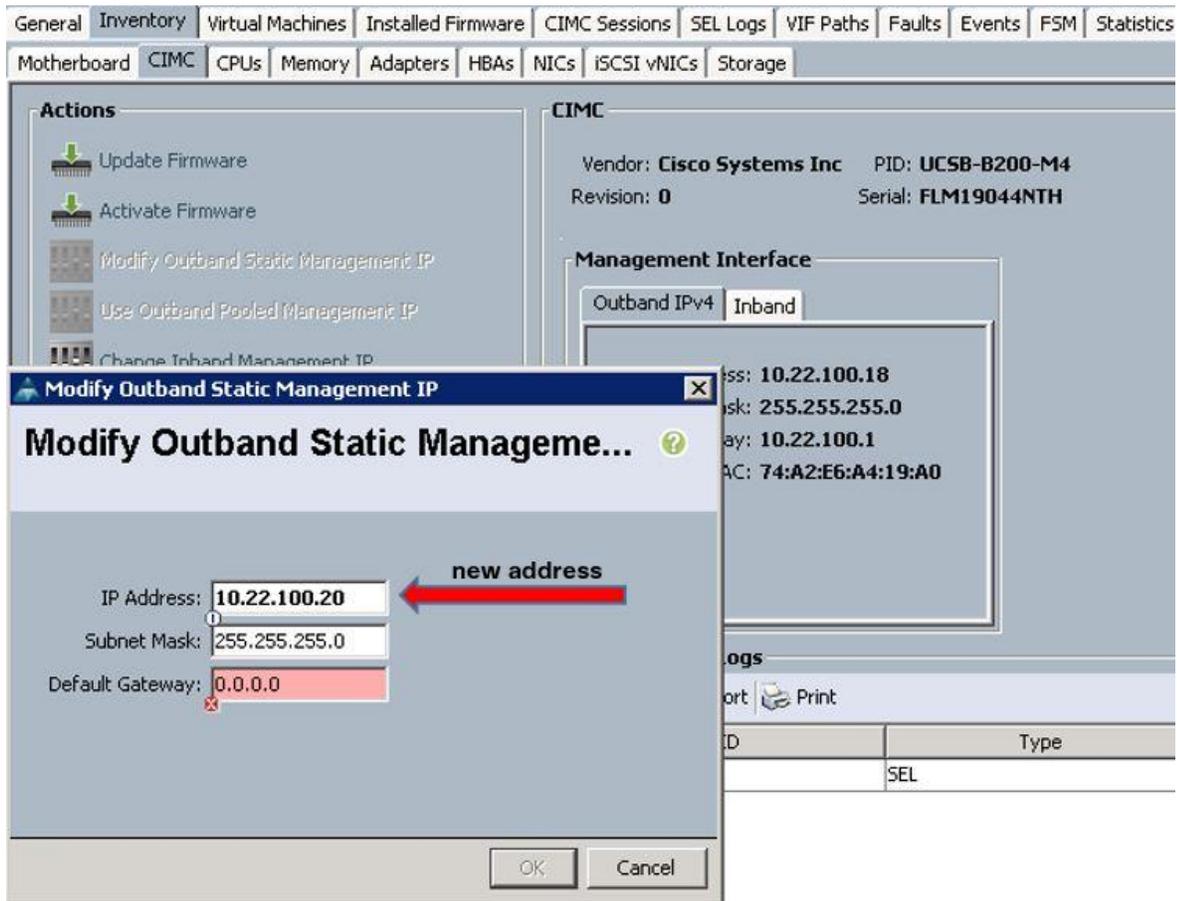
### IPMI address

```
ironic node-update <NODE UUID> replace driver_info/ipmi_address=<NEW IPMI ADDRESS>
```

You can also correct IPMI address from UCS if this IP is free and leave the one in openstack.

LAN tab -> IP Pools -> Check whether IP is free and is available in the block.

General   IP Addresses   IP Blocks   Faults   Events						
IPV4 Addresses   IPV6 Addresses						
Filter   Export   Print						
IP Address	Subnet	Default Gateway	Assigned	Assigned To	Prev Assigned To	
10.22.100.8	255.255.255.0	10.22.100.1	No			
10.22.100.9	255.255.255.0	10.22.100.1	No			
10.22.100.10	255.255.255.0	10.22.100.1	Yes	sys/chassis-1/blade-6/mgmt/ipv4-po...	sys/chassis-1/blade-6/mgmt/ipv4-pool...	
10.22.100.11	255.255.255.0	10.22.100.1	Yes	sys/rack-unit-2/mgmt/ipv4-pooled-a...	sys/rack-unit-2/mgmt/ipv4-pooled-addr	
10.22.100.12	255.255.255.0	10.22.100.1	Yes	sys/chassis-2/blade-3/mgmt/ipv4-po...	sys/chassis-2/blade-3/mgmt/ipv4-pool...	
10.22.100.13	255.255.255.0	10.22.100.1	Yes	sys/chassis-2/blade-7/mgmt/ipv4-po...	sys/chassis-1/blade-8/mgmt/ipv4-pool...	
10.22.100.14	255.255.255.0	10.22.100.1	Yes	sys/chassis-2/blade-1/mgmt/ipv4-po...	sys/chassis-2/blade-1/mgmt/ipv4-pool...	
10.22.100.15	255.255.255.0	10.22.100.1	Yes	sys/chassis-1/blade-1/mgmt/ipv4-po...	sys/chassis-1/blade-1/mgmt/ipv4-pool...	
10.22.100.16	255.255.255.0	10.22.100.1	Yes	sys/chassis-1/blade-7/mgmt/ipv4-po...	sys/chassis-1/blade-7/mgmt/ipv4-pool...	
10.22.100.17	255.255.255.0	10.22.100.1	Yes	sys/chassis-2/blade-8/mgmt/ipv4-po...	sys/chassis-1/blade-8/mgmt/ipv4-pool...	
10.22.100.18	255.255.255.0	10.22.100.1	Yes	sys/chassis-1/blade-3/mgmt/ipv4-po...	sys/chassis-1/blade-3/mgmt/ipv4-pool...	
10.22.100.19	255.255.255.0	10.22.100.1	Yes	sys/chassis-1/blade-2/mgmt/ipv4-po...	sys/chassis-1/blade-2/mgmt/ipv4-pool...	
10.22.100.20	255.255.255.0	10.22.100.1	Yes	sys/rack-unit-3/mgmt/ipv4-pooled-a...	sys/rack-unit-3/mgmt/ipv4-pooled-addr	
10.22.100.21	255.255.255.0	10.22.100.1	Yes	sys/rack-unit-1/mgmt/ipv4-pooled-a...	sys/rack-unit-1/mgmt/ipv4-pooled-addr	
10.22.100.22	255.255.255.0	10.22.100.1	No		sys/chassis-2/blade-7/mgmt/ipv4-pool...	
10.22.100.23	255.255.255.0	10.22.100.1	Yes	sys/chassis-1/blade-8/mgmt/ipv4-po...	sys/chassis-1/blade-8/mgmt/ipv4-pool...	



### Running Introspection on Failed Nodes

At times it may not be feasible to do bulk introspection of all the nodes because of say lun issue on one single node, in particular if you have large number of nodes in the cloud.

```
ironic node-set-provision-state <uuid> manage
openstack baremetal introspection start <uuid>
openstack baremetal introspection status <uuid>. Should give Finished as True and
error as None.
ironic node-set-provision-state <uuid> provide
```

### Overcloud Install

Debugging Overcloud failures sometimes is a daunting task. The issues could be as simple as passing incorrect parameters to Overcloud deploy while some could be bugs as well. Here is an attempt to narrow down the problems. It is difficult to cover all the failure scenarios here. Few of them found out on the configuration are mentioned here. The best place is to debug from here and then move forward with Red Hat and Openstack documentation.

- Check for the flavors pre-defined and verify that they match correctly. Incorrect flavors and/or the number of nodes passed may error with insufficient number of nodes while running Overcloud deploy command. Run instack-ironic-deployment --show-profile to confirm.
- Make sure that you have ntp server configured and check with ntpdate -d -y <ntp server> to check the drift. Preferably should be less than 20 ms for ceph monitors.

- Run in debug mode to capture the errors while running Overcloud deploy.
- The 300 minutes provided in the Appendix for Overcloud deployment command should suffice. But in case of large deployments, it may have to be increased.

## Debug Network Issues

- Overcloud image has been customized with root passwords. This will allow us to login to the node directly through KVM console in case of failures even if heat-admin user is still not setup.
- `journalctl -u os-collect-config | egrep -i "error|trace|fail"` should shed some light around any errors or failures happened during Overcloud deploy.
- Incorrect configuration of yaml files may result in network configuration issues. Run `journalctl` as above to start with. Validate the yaml files with online yaml parsers.
- Run `ifconfig` and `ovs-vsctl show` the mappings.
- `cd /etc/os-net-config. jq . config.json` will spill out the actual parameters that went on to that node.
- Login from Director node to the other nodes and check for the routes. There should be one static route either externally or to the Undercloud node, depending on the way masquading was configured

## Debug Ceph Storage Issues

- Run `journalctl` as above and check for `dmesg` and `/var/log/messages` to reveal any failures related with partitioning and/or network.
- Per `ceph.yaml` included in this document and because of bug [1297251](#), Ceph partitions are pre-created with `wipe_disk.yaml` file. Validate this with `/root/wipe_disk.txt` file and running `cat /proc/partitions`. Only the journal partitions are pre-created. The OSD partitions are created by RHEL-OSP Director.
- Checking the partitions in `/proc/partitions` and the existence of `/var/log/ceph/*`, `/var/lib/ceph*` and `/etc/ceph/keyring` and other files reveal at what stage it failed.
- The monitors should be setup before creating **Ceph OSD's**. **Existence of `/etc/ceph/*` on controller nodes**, followed by that in storage nodes will reveal whether monitor setup was successful or not.
- Run `ceph -s` to check **the health and observe for how many total OSD's, how many are up etc.**
- **Run `ceph osd tree` to reveal issues with any individual OSD's.**
- If you detect clock skew issues on monitors, check for `ntp daemon`, sync up the time on monitors running on controller nodes and restart the monitors `/etc/init.d/ceph mon restart`

## Debug Heat Stack Issues

The following sequence may be followed to debug heat stack create/update issues.

1. `heat stack-list`

```
[stack@osp7-director2 ~]$ heat stack-list
```

id	stack_name	stack_status	creation_time
361c22ad-91b5-40cb-8bd8-9479334e2c2b	overcloud	CREATE_FAILED	2016-02-03T07:30:56Z

2. heat resource-list overcloud | grep -vi complete

```
[stack@osp7-director2 ~]$ heat resource-list overcloud | grep -vi complete
```

resource_name	physical_resource_id	resource_type	resource_status	updated_time
controller	243dc4f4-efe5-4939-a880-e7fcfb4449a	OS::Heat::ResourceGroup	CREATE_FAILED	2016-02-03T07:30:56Z

3. heat resource-list -n5 overcloud | grep -vi complete

```
[stack@osp7-director2 ~]$ heat resource-list -n5 overcloud | grep -vi complete
```

resource_name	physical_resource_id	resource_type	resource_status	updated_time	parent_resource
controller	243dc4f4-efe5-4939-a880-e7fcfb4449a	OS::Heat::ResourceGroup	CREATE_FAILED	2016-02-03T07:30:56Z	Controller
0	ea08ae20-1ef0-4b1d-973e-d3330c4b0ee	OS::TripleO::Controller	CREATE_FAILED	2016-02-03T07:31:14Z	Controller
1	7e6a92b2-2c7f-469c-9ad3-4d5701fae4ed	OS::TripleO::Controller	CREATE_FAILED	2016-02-03T07:31:14Z	Controller
2	365237af-a687-4dca-a89a-320170cedcfc	OS::TripleO::Controller	CREATE_FAILED	2016-02-03T07:31:14Z	Controller
NetworkDeployment	a7a707a3-dd4a-418b-9d8f-0f4c95100c55	OS::TripleO::SoftwareDeployment	CREATE_FAILED	2016-02-03T07:31:17Z	1
updateDeployment	655f9a72-d1f6-4abc-9080-ed5c63141615	OS::Heat::SoftwareDeployment	CREATE_FAILED	2016-02-03T07:31:17Z	1
NetworkDeployment	facc7647-3c4c-419e-a27c-40d6f394c7bc	OS::TripleO::SoftwareDeployment	CREATE_FAILED	2016-02-03T07:31:21Z	0
updateDeployment	ca0ccb49-08a5-4de8-aae1-0868b12018a3	OS::Heat::SoftwareDeployment	CREATE_FAILED	2016-02-03T07:31:21Z	0
NetworkDeployment	f5dfc1c-e56c-4293-8a3e-f7dc528c3460	OS::TripleO::SoftwareDeployment	CREATE_FAILED	2016-02-03T07:31:25Z	2
updateDeployment	89cd7cc2-bc63-4640-80f1-f6971f33fd2f	OS::Heat::SoftwareDeployment	CREATE_FAILED	2016-02-03T07:31:25Z	2

4. heat resource-show overcloud Controller

```
[stack@osp7-director2 ~]$ heat resource-show overcloud controller
```

Property	value
attributes	{ "attributes": null, "refs": null }
description	
links	http://192.168.110.105:8004/v1/09895ee36e614b89bc395cd622eaec0d/stacks/overcloud/361c22ad-91b5-40cb-8bd8-9479334e2c2b/resources/controller (self) http://192.168.110.105:8004/v1/09895ee36e614b89bc395cd622eaec0d/stacks/overcloud/361c22ad-91b5-40cb-8bd8-9479334e2c2b (stack)
logical_resource_id	controller
physical_resource_id	243dc4f4-efe5-4939-a880-e7fcfb4449a
required_by	AllNodesValidationConfig AllNodesExtraConfig ControllerIPListMap ControllerClusterConfig ControllerBootstrapNodeDeployment allNodesConfig ControllerClusterDeployment ControllerSwiftDeployment cephClusterConfig ControllerNodesPostDeployment VipDeployment ControllerAllNodesValidationDeployment ControllerAllNodesDeployment ControllerBootstrapNodeConfig SwiftDevicesAndProxyConfig ControllerCephDeployment
resource_name	controller
resource_status	CREATE_FAILED
resource_status_reason	CREATE aborted
resource_type	OS::Heat::ResourceGroup
updated_time	2016-02-03T07:30:56Z

5. heat deployment-show <deployment id obtained above>

```
[stack@osp7-director2 ~]$ heat deployment-show 89cd7cc2-bc63-4640-80f1-f6971f33fd2f
{
  "status": "IN_PROGRESS",
  "server_id": "ad64d6c3-4cfc-4f49-9fd5-273ccbfdc63e",
  "config_id": "5fd87eae-95fe-43fa-b674-2002abe5c82a",
  "output_values": null,
  "creation_time": "2016-02-03T07:35:45Z",
  "input_values": {},
  "action": "CREATE",
  "status_reason": "Deploy data available",
  "id": "89cd7cc2"
}
```

6. In the current version it has been observed that both introspection and Overcloud deploy happen in batches of ten. While the first 10 are being deployed, the rest of the nodes could be in spawning or wait-call-back mode. **If the status hasn't changed for a while, it may be best to login to the KVM console.** Make sure to map the Overcloud name to UCS service profile name for this. In case of successful installation, /etc/neutron/plugin.ini will be populated by UCSM plugin from where you can extract the mappings between UCS Service Profile name and OpenStack host names. However in case of heat stack failures, UCSM plugin might not have completed the install, and you may have to map it manually in order to login to the correct nodes.

```
[stack@osp7-director ~]$ nova list | grep overcloud-controller
| d0bbf054-e344-4a01-91b7-b17779d6db97 | overcloud-controller-0 | ACTIVE | - | Running | ctlplane=10.22.110.55 |
| 31953e9d-e80c-457a-b48f-27d71101e3ee | overcloud-controller-1 | ACTIVE | - | Running | ctlplane=10.22.110.57 |
| 2b22f9ad-a399-4a8b-8f6b-9f4500d243f4 | overcloud-controller-2 | ACTIVE | - | Running | ctlplane=10.22.110.59 |
[stack@osp7-director ~]$ ironic node-list
-----
| UUID | Name | Instance UUID | Power State | Provision State | Maintenance |
-----
| b7dde876-354a-4688-8550-aec8f64c582c | None | d0bbf054-e344-4a01-91b7-b17779d6db97 | power on | active | False |
| e4563ca5-2f12-4e08-9905-f770f740ad2b | None | 2b22f9ad-a399-4a8b-8f6b-9f4500d243f4 | power on | active | False |
| 285965a9-9713-4301-8ad5-7aa3ef5dd1c2 | None | 31953e9d-e80c-457a-b48f-27d71101e3ee | power on | active | False |
[stack@osp7-director ~]$ ironic node-show b7dde876-354a-4688-8550-aec8f64c582c
-----
| Property | Value |
-----
| target_power_state | None |
| extra | {'u'newly_discovered': u'true', u'block_devices': {u'serials': |
| | [u'618e7283727010e01dfcad510ceb915e', u'40E10200869217CA']}, |
| | u'hardware_swift_object': u'extra hardware- |
| | b7dde876-354a-4688-8550-aec8f64c582c'} |
| last_error | None |
| updated_at | 2016-02-04T20:23:07+00:00 |
| maintenance_reason | None |
| provision_state | active |
| uuid | b7dde876-354a-4688-8550-aec8f64c582c |
| console_enabled | False |
| target_provision_state | None |
| maintenance | False |
| inspection_started_at | None |
| inspection_finished_at | None |
| power_state | power on |
| driver | pxe_ipmitool |
| reservation | None |
| properties | {u'memory_mb': u'262144', u'cpu_arch': u'x86_64', u'local_gb': u'249', |
| | u'cpus': u'72', u'capabilities': u'profile:control,boot_option:local'} |
| instance_uuid | d0bbf054-e344-4a01-91b7-b17779d6db97 |
| name | None |
| driver_info | {u'ipmi_password': u'*****', u'ipmi_address': u'10.22.100.23', |
| | u'ipmi_username': u'admin', u'deploy_kernel': u'ca9667c9-de8f- |
| | 4df1-95d5-b5f8b6fb8f4d', u'deploy_ramdisk': |
| | u'404d0e44-e5c7-4b46-8339-c451441b3f55'} |
-----
```

From the IPMI or mac address you can find out which node and login to the respective KVM console of the Cisco UCS server.

## Overcloud Post-Deployment Issues

1. Check for errors with pcs status on controller nodes. If some resources are not up or running, these may have to be addressed first.

```
pcs resource cleanup will restart all the services.
pcs resource restart <resource name obtained from pcs status>
nova list, nova service-list and keystone endpoint-list could be handy to debug.
nova hypervisor-list or show will reveal details of the hypervisors configured on the system. If any nodes are missing than expected, that may have to be addressed too.
```

2. Confirm that the following are in sync.

```
[root@overcloud-controller-0 ~]# nova hypervisor-list | grep -i compute | wc -l
6
[root@overcloud-controller-0 ~]# ssh -l admin 10.22.100.41
Nexus 1000v Switch
Password:
Last login: Thu Feb  4 18:15:21 2016 from 10.22.100.53
.....
vsm-p# show module | grep overcloud-compute | count
6

[root@overcloud-controller-0 ~]# grep -i "overcloud-compute-"
/etc/neutron/plugin.ini | grep -v ucsn | wc -l
12
```

Following can be concluded from the above:

There are 6 nova hypervisors as seen from OpenStack side.

There are 6 Compute nodes from N1000V.

There are 6 Compute nodes also in /etc/neutron/plugin.ini. This ini file has an entry for each switch.

Hence the number 12 ( twice of the total compute nodes)

## N1KV Plugin Checks

1. Supervisor modules are managed by PCS

```
[root@overcloud-controller-0 ~]# pcs status | grep vsm
vsm-s (ocf::heartbeat:VirtualDomain): Started overcloud-controller-0
vsm-p (ocf::heartbeat:VirtualDomain): Started overcloud-controller-1
```

2. Management IP address

```
vsm-p(config)# show interface mgmt0 brief
```

Port	VRF	Status	IP Address	Speed	MTU
mgmt0	--	up	10.22.100.41	1000	1500

3. Modules in VSM

```
vsm-p# show module | grep "Supervisor"
1 0 Virtual Supervisor Module Nexus1000V active *
2 0 Virtual Supervisor Module Nexus1000V ha-standby
```

4. Port profiles status

```
vsm-p(config)# show port-profile brief
```

Port Profile	Profile Type	Profile State	Eval Items	Max Ports	Assigned Ports	Child Profs
default-pp	Vethernet	1	1	32	0	21
NSM_template_segmentation	Vethernet	1	1	32	0	0
NSM_template_vlan	Vethernet	1	1	32	0	0
system-uplink	Ethernet	1	2	512	9	0

5. Redundancy status

```
vsm-p(config)# show redundancy status
```

```
Redundancy role
```

```
-----
```

```
    administrative: primary
    operational:    primary
```

```
Redundancy mode
```

```
-----
```

```
    administrative: HA
    operational:    HA
```

```
This supervisor (sup-1)
```

```
-----
```

```
    Redundancy state: Active
    Supervisor state: Active
    Internal state:   Active with HA standby
```

```
Other supervisor (sup-2)
```

```
-----
```

```
    Redundancy state: Standby

    Supervisor state: HA standby
    Internal state:   HA standby
```

```
System start time:          Tue Feb  2 22:09:52 2016
```

```
System uptime:              1 days, 20 hours, 14 minutes, 54 seconds
Kernel uptime:              1 days, 20 hours, 15 minutes, 15 seconds
Active supervisor uptime:   1 days, 20 hours, 14 minutes, 15 seconds
```

## 6. Show the VLAN ports

```
[root@overcloud-controller-0 ~]# vemcmd show port vlans
```

LTL	VSM Port	Mode	Native VLAN	VLAN State*	Allowed Vlans
19	Eth5/1	T	1	FWD	1,160,252,258,269-270,277,284,293,297-300,304,307-308,323,330,332,345,347,353
51		A	1	BLK	1
52		A	1	BLK	1
53		A	1	BLK	1

## 7. VSM Trouble shooting document

[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus1000/kvm/troubleshooting\\_guide/n1000v\\_trouble.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus1000/kvm/troubleshooting_guide/n1000v_trouble.html)

## Nexus Plugin Checks

1. Validate entries in /etc/neutron/plugin.ini on all the controllers.

- Any VM's created should have VLAN entries globally in the switch and also in both the port-channels and both the switches. Any missing entry raises alarm here.

### **Nexus Global VLAN's**

```
UCSO-N9K-FAB-A(config)# show vlan | grep q-
120 q-120 active Po1, Po17, Po18, Eth1/1, Eth1/2
215 q-215 active Po1, Po17, Po18, Eth1/1, Eth1/2
251 q-251 active Po1, Po17, Po18, Eth1/1, Eth1/2
```

### **Nexus Port Channel VLAN's**

```
show running-config interface port-channel 17-18
switchport mode trunk
switchport trunk allowed vlan 1,100,110,120,130,150,160,215,251-252
```

Some of the output above is truncated for readability purposes.

## Cisco UCS Manager Plugin Checks

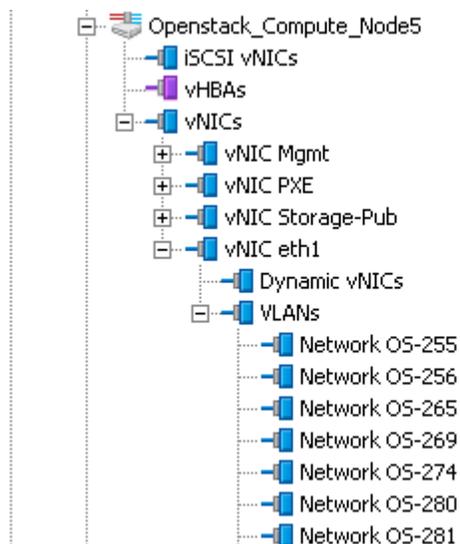
Validate entries in /etc/neutron/plugin.ini on all the controllers

### **UCSM Global VLAN's**

Log into UCS Manager and check for the VLANS both globally and on the hypervisor where the VM(s) is provisioned. Check the host names from cli or horizon.

Name	Type	ID	Transport	Native	VLAN Sharing
VLAN OS-299 (299)	Lan	299	Ether	No	None
VLAN OS-300 (300)	Lan	300	Ether	No	None
VLAN OS-302 (302)	Lan	302	Ether	No	None

### Hypervisor VLAN's



## Run Time Issues

Operational issues can be many but a brief overview of where to check in case of failures around VM Creation is provided below.

1. Nova commands like `nova list --all-tenants`, `nova-manage vm list` and `virsh list` on compute nodes could be a starting point.
2. Check `/var/log/neutron` and `grep -i "error|trace" server.log`. Few may be informational and probably ignorable.
3. Check the following files to spot any errors
  - a. `/var/log/neutron/server.log`
  - b. `/etc/neutron/plugin.ini`
  - c. `/etc/neutron/neutron.conf`
  - d. `/var/log/nova/*`
4. You may execute the following on controller nodes.
  - a. `ip netns`
  - b. `ip netns exec <ns> <arguments>`

```
[root@overcloud-controller-0 ~]# ip netns exec qdhcp-a8242c63-065d-4f5f-afa0-9664250077c6 ip -d link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 promiscuity 0 addrgenmode eui64
39: tap6e099a45-f7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN mode DEFAULT
    link/ether fa:16:3e:49:99:59 brd ff:ff:ff:ff:ff:ff promiscuity 1
    openvswitch addrgenmode eui64
[root@overcloud-controller-0 ~]# ovs-vsctl show | grep tap6e099a45
    Port "tap6e099a45-f7"
        Interface "tap6e099a45-f7"
```

```
[root@overcloud-controller-0 ~]# ip netns pids qdhcp-a8242c63-065d-4f5f-afa0-9664250077c6
60811
[root@overcloud-controller-0 ~]# ps -ef | grep 60811 | grep -v grep
nobody 60811 1 0 Feb02 ? 00:00:00 dnsmasq --no-hosts --no-resolv --strict-order --
bind-interfaces --interface=tap6e099a45-f7 --except-interface=lo --pid-file=/var/lib/neutron/dhc
p/a8242c63-065d-4f5f-afa0-9664250077c6/pid --dhcp-hostsfile=/var/lib/neutron/dhcp/a8242c63-065d-
4f5f-afa0-9664250077c6/host --addn-hosts=/var/lib/neutron/dhcp/a8242c63-065d-4f5f-afa0-966425007
7c6/addn_hosts --dhcp-optsfile=/var/lib/neutron/dhcp/a8242c63-065d-4f5f-afa0-9664250077c6/opts -
-dhcp-leasefile=/var/lib/neutron/dhcp/a8242c63-065d-4f5f-afa0-9664250077c6/leases --dhcp-range=s
et:tag0,10.2.103.0,static,86400s --dhcp-lease-max=256 --conf-file=/etc/neutron/dnsmasq-neutron.c
onf --domain=openstacklocal
[root@overcloud-controller-0 ~]# █
```

5. `neutron agent-list` to check the status and then debugging in the respective areas

agent_type	host	alive	admin_state_up	binary
I3 agent	overcloud-controller-1.localdomain	:~)	True	neutron-l3-agent
Open vSwitch agent	overcloud-compute-5.localdomain	xxx	True	neutron-openvswitch-agent
DHCP agent	overcloud-controller-2.localdomain	:~)	True	neutron-dhcp-agent

From the output above something isn't correct on `overcloud-compute-5`

6. Map nova and neutron

```
[root@overcloud-controller-0 ~]# nova list --all-tenants
```

ID	Name	Tenant ID	Status
f1a8f3f8-8e6f-402f-91df-36b3a2b0f47a	tenant301_101_inst1	173e420370ad42d8917ff9071865e609	ACTIVE
5042db53-1f7b-4b50-86e8-e9f2195a4bae	tenant301_101_inst2	173e420370ad42d8917ff9071865e609	ACTIVE
b08855d0-c7c4-4936-a440-2c3773c3ce73	tenant301_151_inst3	173e420370ad42d8917ff9071865e609	ACTIVE
a66a7580-8374-4cb1-82e0-806358ecec9	tenant301_151_inst4	173e420370ad42d8917ff9071865e609	ACTIVE

7. Take the VM id and seed into neutron port-list

```
[root@overcloud-controller-0 ~]# neutron port-list --device_id=f1a8f3f8-8e6f-402f-91df-36b3a2b0f47a -F id
```

id
4df60463-7b9a-4baf-bf87-309acbda2850

8. Call neutron port-show with these to get the full details

```
[root@overcloud-controller-0 ~]# neutron port-show 4df60463-7b9a-4baf-bf87-309acbda2850
```

Field	Value
admin_state_up	True
allowed_address_pairs	
binding:host_id	overcloud-compute-2.localdomain
binding:profile	{}
binding:vif_details	{"port_filter": true, "ovs_hybrid_plug": true}
binding:vif_type	ovs
binding:vnic_type	normal
device_id	f1a8f3f8-8e6f-402f-91df-36b3a2b0f47a
device_owner	compute:None
extra_dhcp_opts	
fixed_ips	{"subnet_id": "c89c69d2-6640-4ed2-a21e-f84aa9d71334", "ip_address": "10.2.101.3"}
id	4df60463-7b9a-4baf-bf87-309acbda2850
mac_address	fa:16:3e:3f:cd:38
n1kv:profile	a641b20f-9027-48f9-86cc-a92d498554c3
name	
network_id	6c2c467c-e056-41e6-900d-ba8adbf95649
security_groups	6c155a3e-347c-485e-a4a0-daadfd3eeffe
status	ACTIVE
tenant_id	173e420370ad42d8917ff9071865e609

9. Port binding failures and/or update post-commit failures. Cisco plugins will spill out the driver names to identify whether the issue is on n1kv or cisco mech nexus drivers in neutron server log files.

## Best Practices

---

- It is strongly recommended to have one or two blades or servers as spare. While you will have business continuity you may have degraded performance during the period.
- Please plan your networks beforehand and prepare check list of items in place before working on the install. It is suggested to proof read the full document once before attempting the install.
- Capacity planning is another important factor considering the organic growth. This not only includes the physical resources like data center space and servers but also the network subnet sizing etc.
- Please follow the operational best practices like housekeeping, purging the log and archives, etc. In bigger installations you may have to size /var/log separately too.

## List of Bugs

The following is a list of bugs that were encountered while working on this CVD. They are either fixed and rolled out as errata update by Red Hat or workarounds have been evolved and put in place in this document. This list is just for reference purposes only.

<b><u>bug 1178497</u></b>	<u>unable to shutdown - dracut loop</u>
<b><u>bug 1228862</u></b>	<u>openstack undercloud install --force clean option</u>
<b><u>bug 1230163</u></b>	<u>DELETE_FAILED when trying to delete a stack</u>
<b><u>bug 1233416</u></b>	<u>The undercloud should act as an NTP server to initially sync time on nodes</u>
<b><u>bug 1235098</u></b>	<u>Isolated networks not deleted during heat stack-delete overcloud</u>
<b><u>bug 1236167</u></b>	<u>Isolated networks not deleted during heat stack-delete overcloud</u>
<b><u>bug 1238460</u></b>	<u>RHEL Images for overcloud use EDT as timezone</u>
<b><u>bug 1238807</u></b>	<u>the name/type of devices to use for Ceph on the OSDs nodes</u>
<b><u>bug 1241131</u></b>	<u>DNS server is not accessible by different overcloud hosts</u>
<b><u>bug 1243121</u></b>	<u>Neutron port quota fails larger overcloud deployments</u>
<b><u>bug 1244026</u></b>	<u>Overcloud nodes deployed by OSP-Director are using DHCP</u>
<b><u>bug 1246525</u></b>	<u>Repeating "ironic-api" errors in /var/log/messages on the undercloud node</u>
<b><u>bug 1250654</u></b>	<u>overcloud deployment fails on " CephStorageDeployment_Step1</u>
<b><u>bug 1252158</u></b>	<u>overcloud deploy with ceph reports success but ceph is not usable</u>
<b><u>bug 1252260</u></b>	<u>Use ironic to partition Ceph OSD disks automatically during installation</u>
<b><u>bug 1252440</u></b>	<u>[Ceph] Missing storage-environment.yaml</u>
<b><u>bug 1252546</u></b>	<u>Ceph pg_num and ppg_num are correctly set in ceph.yaml but the pools always use 64</u>
<b><u>bug 1252592</u></b>	<u>Document the OVS bonding modes in the official documentation</u>
<b><u>bug 1253959</u></b>	<u>Disk ordering on overcloud deployment and discovery kernel differs from installed</u>
<b><u>bug 1255224</u></b>	<u>Add support for Bonding mode 5 and 6 in OSP-Director</u>
<b><u>bug 1255475</u></b>	<u>Remove traces of --plan from documentation</u>
<b><u>bug 1257414</u></b>	<u>[HA] critical resource constraints missing from pacemaker config</u>
<b><u>bug 1259488</u></b>	<u>Include python-networking-cisco &amp; python-UcsSdk RPMs in disk image</u>
<b><u>bug 1267780</u></b>	<u>Fencing agents on Controller Nodes are stopped when one of the UCS FI's are rebooted</u>
<b><u>bug 1281603</u></b>	<u>OSP7 with I3_ha=false takes longer time</u>

<b><u>bug 1283721</u></b>	Allow for different values of pg_num, ppg_num and size for each Ceph pool
<b><u>bug 1290548</u></b>	Cinder create volume from image: AttributeError:
<b><u>bug 1297251</u></b>	Overcloud Deploy OSP7 y2 on RHEL 7.2 fails on Ceph Install
<b><u>bug 1297254</u></b>	Overcloud Deploy on OSP7 y2 fails randomly on Compute Resource
<b><u>bug 1297975</u></b>	Neutron VLAN ranges ignored except for first bridge on 7.2
<b><u>bug 1298430</u></b>	fence_cisco_ucs package to be modified for High Availability
<b><u>bug 1299795</u></b>	Unable to register overcloud hosts to RHN network through overcloud deploy

## Reference Documents

---

1	<a href="http://www.cisco.com/c/en/us/products/servers-unified-computing/index.html">http://www.cisco.com/c/en/us/products/servers-unified-computing/index.html</a>
2	<a href="http://www.cisco.com/c/en/us/support/servers-unified-computing/unified-computing-system/products-technical-reference-list.html">http://www.cisco.com/c/en/us/support/servers-unified-computing/unified-computing-system/products-technical-reference-list.html</a>
3	<a href="https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/7/html/Director_Installation_and_Usage/chap-Introduction.html">https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/7/html/Director_Installation_and_Usage/chap-Introduction.html</a>
4	<a href="http://docs.openstack.org/developer/tripleo-docs/troubleshooting/troubleshooting.html#debugging-using-heat">http://docs.openstack.org/developer/tripleo-docs/troubleshooting/troubleshooting.html#debugging-using-heat</a>
5	<a href="http://docs.openstack.org/developer/tripleo-docs/advanced_deployment/network_isolation.html">http://docs.openstack.org/developer/tripleo-docs/advanced_deployment/network_isolation.html</a>
6	<a href="http://docs.ceph.com/docs/master/rbd/rbd-openstack/">http://docs.ceph.com/docs/master/rbd/rbd-openstack/</a>

## Conclusion

---

This paper is a joint contribution from Cisco Systems Inc, Red Hat Inc and Intel Corporation. The solution is baked by combining the technologies, expertise, contributions to OpenStack community and experience from the field and will provide a rich experience to the end users both on installation and day to day operational aspects of OpenStack.

## Appendix A

---



Template files are sensitive to whitespaces and tabs. Please copy as is into a txt file and then rename them to .yaml, followed by validating them in online yaml parser like <http://yaml-online-parser.appspot.com/>

---



Sample Network and other yaml files can also be downloaded from <https://cnsug-yum-server.cisco.com/yumrepo/cvd/>

---

### Undercloud instackenv.json

```
{
  "nodes": [
    {
      "pm_user": "admin",
      "pm_password": "Whatever_Password",
      "pm_type": "pxe_ipmitool",
      "pm_addr": "10.22.100.23",
      "mac": [
        "00:25:b5:00:00:6c"
      ],
      "memory": "262144",
      "disk": "270",
      "arch": "x86_64",
      "cpu": "72"
    },
    {
      "pm_user": "admin",
      "pm_password": "Whatever_Password",
      "pm_type": "pxe_ipmitool",
      "pm_addr": "10.22.100.22",
      "mac": [
        "00:25:b5:00:00:2a"
      ],
      "memory": "131072",
      "disk": "270",
      "arch": "x86_64",
      "cpu": "32"
    },
    {
      "pm_user": "admin",
      "pm_password": "Whatever_Password",
      "pm_type": "pxe_ipmitool",
      "pm_addr": "10.22.100.16",
      "mac": [
        "00:25:b5:00:00:5b"
      ],
    }
  ]
}
```

```

    "memory": "131072",
    "disk": "270",
    "arch": "x86_64",
    "cpu": "48"
  },
  {
    "pm_user": "admin",
    "pm_password": "Whatever_Password",
    "pm_type": "pxe_ipmitool",
    "pm_addr": "10.22.100.18",
    "mac": [
      "00:25:b5:00:00:0f"
    ],
    "memory": "262144",
    "disk": "270",
    "arch": "x86_64",
    "cpu": "40"
  },
  {
    "pm_user": "admin",
    "pm_password": "Whatever_Password",
    "pm_type": "pxe_ipmitool",
    "pm_addr": "10.22.100.12",
    "mac": [
      "00:25:b5:00:00:5e"
    ],
    "memory": "262144",
    "disk": "270",
    "arch": "x86_64",
    "cpu": "40"
  },
  {
    "pm_user": "admin",
    "pm_password": "Whatever_Password",
    "pm_type": "pxe_ipmitool",
    "pm_addr": "10.22.100.19",
    "mac": [
      "00:25:b5:00:00:2d"
    ],
    "memory": "262144",
    "disk": "270",
    "arch": "x86_64",
    "cpu": "40"
  },
  {
    "pm_user": "admin",
    "pm_password": "Whatever_Password",
    "pm_type": "pxe_ipmitool",
    "pm_addr": "10.22.100.17",
    "mac": [
      "00:25:b5:00:00:5c"
    ],
    "memory": "131072",
    "disk": "270",

```

```

    "arch": "x86_64",
    "cpu": "48"
  },
  {
    "pm_user": "admin",
    "pm_password": "Whatever_Password",
    "pm_type": "pxe_ipmitool",
    "pm_addr": "10.22.100.15",
    "mac": [
      "00:25:b5:00:00:4b"
    ],
    "memory": "262144",
    "disk": "270",
    "arch": "x86_64",
    "cpu": "40"
  },
  {
    "pm_user": "admin",
    "pm_password": "Whatever_Password",
    "pm_type": "pxe_ipmitool",
    "pm_addr": "10.22.100.14",
    "mac": [
      "00:25:b5:00:00:7a"
    ],
    "memory": "262144",
    "disk": "270",
    "arch": "x86_64",
    "cpu": "40"
  },
  {
    "pm_user": "admin",
    "pm_password": " Whatever_Password",
    "pm_type": "pxe_ipmitool",
    "pm_addr": "10.22.100.20",
    "mac": [
      "00:25:b5:00:00:46"
    ],
    "memory": "262144",
    "disk": "400",
    "arch": "x86_64",
    "cpu": "40"
  },
  {
    "pm_user": "admin",
    "pm_password": "Whatever_Password",
    "pm_type": "pxe_ipmitool",
    "pm_addr": "10.22.100.11",
    "mac": [
      "00:25:b5:00:00:57"
    ],
    "memory": "262144",
    "disk": "400",
    "arch": "x86_64",
    "cpu": "40"
  }

```

```

    },
    {
      "pm_user": "admin",
      "pm_password": "Whatever_Password",
      "pm_type": "pxe_ipmitool",
      "pm_addr": "10.22.100.21",
      "mac": [
        "00:25:b5:00:00:17"
      ],
      "memory": "262144",
      "disk": "400",
      "arch": "x86_64",
      "cpu": "40"
    }
  ]
}

```

## Overcloud Templates

### network-environment.yaml

```

resource_registry:
  OS::TripleO::NodeUserData:
    /home/stack/templates/wipe_disk.yaml
  OS::TripleO::Compute::Net::SoftwareConfig:
    /home/stack/templates/nic-configs/compute.yaml
  OS::TripleO::Controller::Net::SoftwareConfig:
    /home/stack/templates/nic-configs/controller.yaml
  OS::TripleO::CephStorage::Net::SoftwareConfig:
    /home/stack/templates/nic-configs/ceph-storage.yaml

parameter_defaults:
  InternalApiNetCidr: 10.22.100.0/24
  StorageNetCidr: 10.22.120.0/24
  StorageMgmtNetCidr: 10.22.150.0/24
  # TenantNetCidr: 10.22.130.0/24
  TenantNetCidr: 10.0.0.0/12
  ExternalNetCidr: 172.22.215.0/24
  InternalApiAllocationPools: [{'start': '10.22.100.50', 'end': '10.22.100.250'}]
  StorageAllocationPools: [{'start': '10.22.120.50', 'end': '10.22.120.250'}]
  StorageMgmtAllocationPools: [{'start': '10.22.150.50', 'end': '10.22.150.250'}]
  TenantAllocationPools: [{'start': '10.0.0.10', 'end': '10.15.255.250'}]
  ExternalAllocationPools: [{'start': '172.22.215.91', 'end': '172.22.215.99'}]
  ExternalNetworkVlanID: 215
  InternalApiNetworkVlanID: 100
  StorageNetworkVlanID: 120
  StorageMgmtNetworkVlanID: 150
  ControlPlaneSubnetCidr: "24"
  ControlPlaneDefaultRoute: 10.22.110.26
  EC2MetadataIp: 10.22.110.26
  DnsServers: ['8.8.8.8', '8.8.4.4']
  # TenantNetworkVlanID: 130
  # Set to the router gateway on the external network
  ExternalInterfaceDefaultRoute: "172.22.215.1"
  # Set to "br-ex" if using floating IPs on native VLAN on bridge br-ex
  NeutronExternalNetworkBridge: ""

```

## storage-environment.yaml

```
parameters:
  CinderEnableIscsiBackend: false
  CinderEnableRbdBackend: true
  NovaEnableRbdBackend: true
  GlanceBackend: rbd
```

## controller.yaml

```
heat_template_version: 2015-04-30
```

```
description: >
```

```
Software Config to drive os-net-config with 2 bonded nics on a bridge
with a VLANs attached for the controller role.
```

```
parameters:
```

```
ExternalIpSubnet:
```

```
  default: ''
```

```
  description: IP address/subnet on the external network
```

```
  type: string
```

```
InternalApiIpSubnet:
```

```
  default: ''
```

```
  description: IP address/subnet on the internal API network
```

```
  type: string
```

```
StorageIpSubnet:
```

```
  default: ''
```

```
  description: IP address/subnet on the storage network
```

```
  type: string
```

```
StorageMgmtIpSubnet:
```

```
  default: ''
```

```
  description: IP address/subnet on the storage mgmt network
```

```
  type: string
```

```
TenantIpSubnet:
```

```
  default: ''
```

```
  description: IP address/subnet on the tenant network
```

```
  type: string
```

```
BondInterfaceOvsOptions:
```

```
  default: ''
```

```
  description: The ovs_options string for the bond interface. Set things like
               lacp=active and/or bond_mode=balance-slb using this option.
```

```
  type: string
```

```
ExternalNetworkVlanID:
```

```
  default: 215
```

```
  description: Vlan ID for the external network traffic.
```

```
  type: number
```

```
InternalApiNetworkVlanID:
```

```
  default: 100
```

```
  description: Vlan ID for the internal_api network traffic.
```

```
  type: number
```

```
StorageNetworkVlanID:
```

```
  default: 120
```

```
  description: Vlan ID for the storage network traffic.
```

```
  type: number
```

```
StorageMgmtNetworkVlanID:
```

```
  default: 150
```

```

    description: Vlan ID for the storage mgmt network traffic.
    type: number
TenantNetworkVlanID:
    default: 130
    description: Vlan ID for the tenant network traffic.
    type: number
ExternalInterfaceDefaultRoute:
    default: ''
    description: default route for the external network
    type: string
ControlPlaneIp:
    default: ''
    description: IP address/subnet on the ctlplane network
    type: string
ControlPlaneSubnetCidr:
    default: '24'
    description: The subnet CIDR of the control plane network.
    type: string
ControlPlaneDefaultRoute:
    default: '10.22.110.26'
    description: The Control Plane Default route.
    type: string
DnsServers:
    default: ['8.8.8.8','8.8.4.4']
    description: A list of DNS servers (2 max) to add to resolv.conf.
    type: json
EC2MetadataIp:
    description: The IP address of the EC2 metadata server.
    type: string

resources:
  OsNetConfigImpl:
    type: OS::Heat::StructuredConfig
    properties:
      group: os-apply-config
      config:
        os_net_config:
          network_config:
            -
              type: interface
              name: nic1
              use_dhcp: false
              dns_servers: {get_param: DnsServers}
              addresses:
                -
                  ip_netmask:
                    list_join:
                      - '/'
                      - - {get_param: ControlPlaneIp}
                        - {get_param: ControlPlaneSubnetCidr}
              routes:
                -
                  ip_netmask: 169.254.169.254/32
                  next_hop: {get_param: EC2MetadataIp}
            -
              type: ovs_bridge
              name: {get_input: bridge_name}
              use_dhcp: false
              members:

```

```

-
  type: interface
  name: nic4
  primary: true
-
  type: vlan
  vlan_id: {get_param: ExternalNetworkVlanID}
  addresses:
    -
      ip_netmask: {get_param: ExternalIpSubnet}
  routes:
    -
      ip_netmask: 0.0.0.0/0
      next_hop: {get_param: ExternalInterfaceDefaultRoute}
-
type: ovs_bridge
name: br-mgmt
members:
  -
    type: interface
    name: nic3
    primary: true
  -
    type: vlan
    vlan_id: {get_param: InternalApiNetworkVlanID}
    addresses:
      -
        ip_netmask: {get_param: InternalApiIpSubnet}
-
type: ovs_bridge
name: br-storage-pub
mtu: 9000
members:
  -
    type: interface
    name: nic5
    mtu: 9000
    primary: true
  -
    type: vlan
    mtu: 9000
    vlan_id: {get_param: StorageNetworkVlanID}
    addresses:
      -
        ip_netmask: {get_param: StorageIpSubnet}
-
type: ovs_bridge
name: br-storage-clus
mtu: 9000
members:
  -
    type: interface
    name: nic6
    mtu: 9000
    primary: true
  -
    type: vlan
    mtu: 9000
    vlan_id: {get_param: StorageMgmtNetworkVlanID}

```

```

        addresses:
        -
            ip_netmask: {get_param: StorageMgmtIpSubnet}
    -
        type: ovs_bridge
        name: br-floating
        members:
        -
            type: interface
            name: nic7
            primary: true
outputs:
  OS::stack_id:
    description: The OsNetConfigImpl resource.
    value: {get_resource: OsNetConfigImpl}

```

## compute.yaml

```
heat_template_version: 2015-04-30
```

```
description: >
```

```
Software Config to drive os-net-config with 2 bonded nics on a bridge
with a VLANs attached for the compute role.
```

```
parameters:
```

```
ExternalIpSubnet:
```

```
  default: ''
```

```
  description: IP address/subnet on the external network
```

```
  type: string
```

```
InternalApiIpSubnet:
```

```
  default: ''
```

```
  description: IP address/subnet on the internal API network
```

```
  type: string
```

```
StorageIpSubnet:
```

```
  default: ''
```

```
  description: IP address/subnet on the storage network
```

```
  type: string
```

```
StorageMgmtIpSubnet:
```

```
  default: ''
```

```
  description: IP address/subnet on the storage mgmt network
```

```
  type: string
```

```
TenantIpSubnet:
```

```
  default: ''
```

```
  description: IP address/subnet on the tenant network
```

```
  type: string
```

```
BondInterfaceOvsOptions:
```

```
  default: ''
```

```
  description: The ovs_options string for the bond interface. Set things like
```

```
  type: string
```

```
InternalApiNetworkVlanID:
```

```
  default: 100
```

```
  description: Vlan ID for the internal_api network traffic.
```

```
  type: number
```

```
StorageNetworkVlanID:
```

```
  default: 120
```

```
  description: Vlan ID for the storage network traffic.
```

```
  type: number
```

```
TenantNetworkVlanID:
```

```

    default: 130
    description: Vlan ID for the tenant network traffic.
    type: number
ControlPlaneIp:
    default: ''
    description: IP address/subnet on the ctlplane network
    type: string
ControlPlaneSubnetCidr:
    default: '24'
    description: The subnet CIDR of the control plane network.
    type: string
ControlPlaneDefaultRoute:
    default: '10.22.110.26'
    description: The Control Plane Default route.
    type: string
DnsServers:
    default: ['8.8.8.8','8.8.4.4']
    description: A list of DNS servers (2 max) to add to resolv.conf.
    type: json
EC2MetadataIp:
    description: The IP address of the EC2 metadata server.
    type: string

resources:
  OsNetConfigImpl:
    type: OS::Heat::StructuredConfig
    properties:
      group: os-apply-config
      config:
        os_net_config:
          network_config:
            -
              type: interface
              name: nic1
              use_dhcp: false
              dns_servers: {get_param: DnsServers}
              addresses:
                -
                  ip_netmask:
                    list_join:
                      - '/'
                  - - {get_param: ControlPlaneIp}
                    - {get_param: ControlPlaneSubnetCidr}
            routes:
              -
                ip_netmask: 169.254.169.254/32
                next_hop: {get_param: EC2MetadataIp}
              -
                default: true
                next_hop: {get_param: ControlPlaneDefaultRoute}
            -
              type: ovs_bridge
              name: br-storage-pub
              mtu: 9000
              members:
                -
                  type: interface
                  name: nic4
                  mtu: 9000

```

```

        primary: true
    -
      type: vlan
      mtu: 9000
      vlan_id: {get_param: StorageNetworkVlanID}
      addresses:
    -
      ip_netmask: {get_param: StorageIpSubnet}
  -
    type: ovs_bridge
    name: br-mgmt
    members:
  -
    type: interface
    name: nic3
    primary: true
  -
    type: vlan
    vlan_id: {get_param: InternalApiNetworkVlanID}
    addresses:
  -
    ip_netmask: {get_param: InternalApiIpSubnet}
outputs:
  OS::stack_id:
    description: The OsNetConfigImpl resource.
    value: {get_resource: OsNetConfigImpl}

```

## ceph-storage.yaml

heat\_template\_version: 2015-04-30

description: >

Software Config to drive os-net-config with 2 bonded nics on a bridge with a VLANs attached for the ceph storage role.

parameters:

ExternalIpSubnet:

default: ''

description: IP address/subnet on the external network

type: string

InternalApiIpSubnet:

default: ''

description: IP address/subnet on the internal API network

type: string

StorageIpSubnet:

default: ''

description: IP address/subnet on the storage network

type: string

StorageMgmtIpSubnet:

default: ''

description: IP address/subnet on the storage mgmt network

type: string

TenantIpSubnet:

default: ''

description: IP address/subnet on the tenant network

type: string

BondInterfaceOvsOptions:

default: ''

description: The ovs\_options string for the bond interface. Set things like

```

        lacp=active and/or bond_mode=balance-slb using this option.
    type: string
InternalApiNetworkVlanID:
    default: ''
    description: Vlan ID for the internal_api network traffic.
    type: number
StorageNetworkVlanID:
    default: ''
    description: Vlan ID for the storage network traffic.
    type: number
StorageMgmtNetworkVlanID:
    default: ''
    description: Vlan ID for the storage mgmt network traffic.
    type: number
ControlPlaneIp:
    default: ''
    description: IP address/subnet on the ctlplane network
    type: string
ControlPlaneSubnetCidr:
    default: '24'
    description: The subnet CIDR of the control plane network.
    type: string
ControlPlaneDefaultRoute:
    default: '10.22.110.26'
    description: The Control Plane Default route.
    type: string
DnsServers:
    default: ['8.8.8.8','8.8.4.4']
    description: A list of DNS servers (2 max) to add to resolv.conf.
    type: json
EC2MetadataIp:
    description: The IP address of the EC2 metadata server.
    type: string

resources:
  OsNetConfigImpl:
    type: OS::Heat::StructuredConfig
    properties:
      group: os-apply-config
      config:
        os_net_config:
          network_config:
            -
              type: interface
              name: nic1
              use_dhcp: false
              dns_servers: {get_param: DnsServers}
              addresses:
                -
                  ip_netmask:
                    list_join:
                      - '/'
                      - - {get_param: ControlPlaneIp}
                        - {get_param: ControlPlaneSubnetCidr}
              routes:
                -
                  ip_netmask: 169.254.169.254/32
                  next_hop: {get_param: EC2MetadataIp}
            -

```

```

        default: true
        next_hop: {get_param: ControlPlaneDefaultRoute}
-
    type: ovs_bridge
    name: br-storage-pub
    mtu: 9000
    members:
    -
        type: interface
        name: nic2
        mtu: 9000
        primary: true
    -
        type: vlan
        mtu: 9000
        vlan_id: {get_param: StorageNetworkVlanID}
        addresses:
        -
            ip_netmask: {get_param: StorageIpSubnet}
-
    type: ovs_bridge
    name: br-storage-clus
    mtu: 9000
    members:
    -
        type: interface
        name: nic3
        mtu: 9000
        primary: true
    -
        type: vlan
        mtu: 9000
        vlan_id: {get_param: StorageMgmtNetworkVlanID}
        addresses:
        -
            ip_netmask: {get_param: StorageMgmtIpSubnet}
outputs:
  OS::stack_id:
    description: The OsNetConfigImpl resource.
    value: {get_resource: OsNetConfigImpl}

```

## ceph.yaml (C240M4L)

```

ceph::profile::params::osd_journal_size: 20000
ceph::profile::params::osd_pool_default_pg_num: 128
ceph::profile::params::osd_pool_default_pgp_num: 128
ceph::profile::params::osd_pool_default_size: 3
ceph::profile::params::osd_pool_default_min_size: 1
ceph::profile::params::manage_repo: false
ceph::profile::params::authentication_type: cephx
ceph::profile::params::osds:
  '/dev/sdd':
    journal: '/dev/sdb1'
  '/dev/sde':
    journal: '/dev/sdb2'
  '/dev/sdf':
    journal: '/dev/sdb3'
  '/dev/sdg':

```

```

    journal: '/dev/sdb4'
'/dev/sdh':
    journal: '/dev/sdc1'
'/dev/sdi':
    journal: '/dev/sdc2'
'/dev/sdj':
    journal: '/dev/sdc3'
'/dev/sdk':
    journal: '/dev/sdc4'

```

ceph\_pools:

```

- "%{hiera('cinder_rbd_pool_name')}}"
- "%{hiera('nova::compute::rbd::libvirt_images_rbd_pool')}}"
- "%{hiera('glance::backend::rbd::rbd_store_pool')}}"

```

ceph\_osd\_selinux\_permissive: true

## ceph.yaml (C240M4S)

```

ceph::profile::params::osd_journal_size: 20000
ceph::profile::params::osd_pool_default_pg_num: 128
ceph::profile::params::osd_pool_default_pgp_num: 128
ceph::profile::params::osd_pool_default_size: 3
ceph::profile::params::osd_pool_default_min_size: 1
ceph::profile::params::manage_repo: false
ceph::profile::params::authentication_type: cephx
ceph::profile::params::osds:
  '/dev/sdf':
    journal: '/dev/sdb1'
  '/dev/sdg':
    journal: '/dev/sdb2'
  '/dev/sdh':
    journal: '/dev/sdb3'
  '/dev/sdi':
    journal: '/dev/sdb4'
  '/dev/sdj':
    journal: '/dev/sdb5'
  '/dev/sdk':
    journal: '/dev/sdc1'
  '/dev/sdl':
    journal: '/dev/sdc2'
  '/dev/sdm':
    journal: '/dev/sdc3'
  '/dev/sdn':
    journal: '/dev/sdc4'
  '/dev/sdo':
    journal: '/dev/sdc5'
  '/dev/sdp':
    journal: '/dev/sdd1'
  '/dev/sdq':
    journal: '/dev/sdd2'
  '/dev/sdr':
    journal: '/dev/sdd3'
  '/dev/sds':
    journal: '/dev/sdd4'
  '/dev/sdt':
    journal: '/dev/sde1'
  '/dev/sdu':

```

```

    journal: '/dev/sde2'
  '/dev/sdv':
    journal: '/dev/sde3'
  '/dev/sdw':
    journal: '/dev/sde4'

```

```
ceph_pools:
```

- "\${hiera('cinder\_rbd\_pool\_name')}}"
- "\${hiera('nova::compute::rbd::libvirt\_images\_rbd\_pool')}}"
- "\${hiera('glance::backend::rbd::rbd\_store\_pool')}}"

```
ceph_osd_selinux_permissive: true
```

## cisco-plugins.yaml

```
resource_registry:
```

```

  OS::TripleO::ControllerExtraConfigPre: /usr/share/openstack-tripleo-heat-
  templates/puppet/extraconfig/pre_deploy/controller/cisco-nlkv.yaml
  OS::TripleO::ComputeExtraConfigPre: /usr/share/openstack-tripleo-heat-
  templates/puppet/extraconfig/pre_deploy/controller/cisco-nlkv.yaml
  OS::TripleO::Controller: /usr/share/openstack-tripleo-heat-
  templates/puppet/controller-puppet.yaml
  OS::TripleO::AllNodesExtraConfig: /usr/share/openstack-tripleo-heat-
  templates/puppet/extraconfig/all_nodes/neutron-ml2-cisco-nexus-ucsm.yaml
  # network type depends on your setup, but if your testbed has a single nic, then you
  need to use the bridge config
  # OS::TripleO::Compute::Net::SoftwareConfig: /usr/share/openstack-tripleo-heat-
  templates/net-config-bridge.yaml

```

```
parameter_defaults:
```

```

  N1000vVSMIP: '10.22.100.41'
  N1000vMgmtGatewayIP: '10.22.100.1'
  N1000vMgmtNetmask: '255.255.255.0'
  N1000vVSMDomainID: '500'
  N1000vVSMPassword: 'Password'
  N1000vPacemakerControl: true
  N1000vVSMHostMgmtIntf: 'br-mgmt'
  N1000vExistingBridge: true
  N1000vVSMVersion: ''
  N1000vVEMHostMgmtIntf: 'vlan100'
  N1000vVSMHostMgmtIntfVlan: 100
  N1000vUplinkProfile: '{eth1: system-uplink,}'
  NetworkUCSMIP: '10.22.100.5'
  NetworkUCSMUsername: 'admin'
  NetworkUCSMPassword: 'nbv12345'
  NetworkUCSMHostList:
'00:25:b5:00:00:0f:Openstack_Compute_Node1,00:25:b5:00:00:5e:Openstack_Compute_Node2,00
:25:b5:00:00:2d:Openstack_Compute_Node3,00:25:b5:00:00:5c:Openstack_Compute_Node4,00:25
:b5:00:00:4b:Openstack_Compute_Node5,00:25:b5:00:00:7a:Openstack_Compute_Node6,00:25:b5
:00:00:6c:Openstack_Controller_Node1,00:25:b5:00:00:2a:Openstack_Controller_Node2,00:25
:b5:00:00:5b:Openstack_Controller_Node3'
  NetworkNexusConfig: {
    "UCSO-N9K-FAB-A": {
      "ip_address": "10.22.100.3",
      "nve_src_intf": 0,
      "password": "Nbv!2345",
      "physnet": "",

```

```

"servers": {
  "00:25:b5:00:00:6c": {
    "ports": "port-channel:17,port-channel:18"
  },
  "00:25:b5:00:00:2a": {
    "ports": "port-channel:17,port-channel:18"
  },
  "00:25:b5:00:00:5b": {
    "ports": "port-channel:17,port-channel:18"
  },
  "00:25:b5:00:00:0f": {
    "ports": "port-channel:17,port-channel:18"
  },
  "00:25:b5:00:00:5e": {
    "ports": "port-channel:17,port-channel:18"
  },
  "00:25:b5:00:00:2d": {
    "ports": "port-channel:17,port-channel:18"
  },
  "00:25:b5:00:00:5c": {
    "ports": "port-channel:17,port-channel:18"
  },
  "00:25:b5:00:00:4b": {
    "ports": "port-channel:17,port-channel:18"
  },
  "00:25:b5:00:00:7a": {
    "ports": "port-channel:17,port-channel:18"
  },
},
"ssh_port": 22,
"username": "admin"
},
"UCSO-N9K-FAB-B": {
  "ip_address": "10.22.100.4",
  "nve_src_intf": 0,
  "password": "Nbv!2345",
  "physnet": "",
  "servers": {
    "00:25:b5:00:00:6c": {
      "ports": "port-channel:17,port-channel:18"
    },
    "00:25:b5:00:00:2a": {
      "ports": "port-channel:17,port-channel:18"
    },
    "00:25:b5:00:00:5b": {
      "ports": "port-channel:17,port-channel:18"
    },
    "00:25:b5:00:00:0f": {
      "ports": "port-channel:17,port-channel:18"
    },
    "00:25:b5:00:00:5e": {
      "ports": "port-channel:17,port-channel:18"
    },
    "00:25:b5:00:00:2d": {
      "ports": "port-channel:17,port-channel:18"
    },
    "00:25:b5:00:00:5c": {
      "ports": "port-channel:17,port-channel:18"
    },
  },
}

```

```

        "00:25:b5:00:00:4b": {
            "ports": "port-channel:17,port-channel:18"
        },
        "00:25:b5:00:00:7a": {
            "ports": "port-channel:17,port-channel:18"
        },
    },
    "ssh_port": 22,
    "username": "admin"
}
}
NetworkNexusManagedPhysicalNetwork: physnet-tenant
NetworkNexusVlanNamePrefix: 'q-'
NetworkNexusSviRoundRobin: 'false'
NetworkNexusProviderVlanNamePrefix: 'p-'
NetworkNexusPersistentSwitchConfig: 'false'
NetworkNexusSwitchHeartbeatTime: 30
NetworkNexusSwitchReplayCount: 1000
NetworkNexusProviderVlanAutoCreate: 'true'
NetworkNexusProviderVlanAutoTrunk: 'true'
NetworkNexusVxlanGlobalConfig: 'false'
NetworkNexusHostKeyChecks: 'false'
EnablePackageInstall: false
NeutronMechanismDrivers: 'cisco_nlkv,cisco_nexus,cisco_ucsm'
NeutronServicePlugins: 'router,networking_cisco.plugins.ml2.drivers.cisco.nlkv.policy_profile_service.PolicyProfilePlugin'
NeutronTypeDrivers: 'vlan'
NeutronCorePlugin: 'neutron.plugins.ml2.plugin.Ml2Plugin'
NeutronDhcpAgentsPerNetwork: 1
NeutronAllowL3AgentFailover: 'true'
NeutronL3HA: 'false'
NeutronNetworkVLANRanges: 'physnet-tenant:250:749'
NetworkNexusVxlanVniRanges: '0:0'
NetworkNexusVxlanMcastRanges: '0.0.0.0:0.0.0.0'
parameters:
  controllerExtraConfig:
    neutron::server::api_workers: 0
    neutron::agents::metadata::metadata_workers: 0
    neutron::server::rpc_workers: 0

```

## wipe\_disk.yaml (C240M4L)

```

heat_template_version: 2015-04-30
#This configuration is only for C240M4L server. Change for C240M4S
#The first for loop zapdisks all the storage disks
#The second for loop creates aligned partitions, only for the journals. Change the values of sdb and sdc accordingly
#In case of more disks for C240M4S change the device names accordingly
#Do not create partitions for data disks
resources:
  userdata:
    type: OS::Heat::MultipartMime
    properties:
      parts:
        - config: {get_resource: clean_disk}

  clean_disk:
    type: OS::Heat::SoftwareConfig

```

```

properties:
  config: |
    #!/bin/bash
    DATA_DISKS="sdd sde sdf sdg sdh sdi sdj sdk"
    JOURNAL_DISKS="sdb sdc"
    JOURNAL_SIZE=20G
    { for disk in $DATA_DISKS $JOURNAL_DISKS
    do
      sgdisk -Z /dev/$disk
      sgdisk -g /dev/$disk
    done } > /root/wipe_disk.txt
    { for disk in $JOURNAL_DISKS
    do
      export ptype1=45b0969e-9b03-4f30-b4c6-b4b80ceff106
      for i in $(seq 1 $(( $(echo $DATA_DISKS|wc -w)+$(echo $JOURNAL_DISKS|wc -w)-
1) / $(echo $JOURNAL_DISKS|wc -w) )) )
      do
        sgdisk --new=$i::+$JOURNAL_SIZE --change-name="$i:ceph journal" --
typecode="$i:$ptype1" /dev/$disk
      done
    done } >> /root/wipe_disk.txt

```

outputs:

```

OS::stack_id:
  value: {get_resource: userdata}

```

## wipe\_disk.yaml (C240M4S)

```

heat_template_version: 2015-04-30
#This configuraion is only for C240M4L server. Change for C240M4S
#The first for loop zapdisks all the storage disks
#The second for loop creates aligned partitions, only for the journals. Change the val-
ues of sdb and sdc accordingly
#In case of more journal disks for C240M4S add the device names for the journals in the
second for loop.
#Do not create partitions for data disks
resources:
  userdata:
    type: OS::Heat::MultipartMime
    properties:
      parts:
        - config: {get_resource: clean_disk}

  clean_disk:
    type: OS::Heat::SoftwareConfig
    properties:
      config: |
        #!/bin/bash
        DATA_DISKS="sdf sdg sdh sdi sdj sdk sdl sdm sdn sdo sdp sdq sdr sds sdt sdu sdv
sdw"
        JOURNAL_DISKS="sdb sdc sdd sde"
        JOURNAL_SIZE=20G
        { for disk in $DATA_DISKS $JOURNAL_DISKS
        do
          sgdisk -Z /dev/$disk
          sgdisk -g /dev/$disk
        done } > /root/wipe_disk.txt
        { for disk in $JOURNAL_DISKS

```

```

do
  export ptype1=45b0969e-9b03-4f30-b4c6-b4b80ceff106
  for i in $(seq 1 $(( $(echo $DATA_DISKS|wc -w)+$(echo $JOURNAL_DISKS|wc -w)-
1) / $(echo $JOURNAL_DISKS|wc -w) )) )
  do
    sgdisk --new=$i:+$JOURNAL_SIZE --change-name="$i:ceph journal" --
typecode="$i:$ptype1" /dev/$disk
  done
done } >> /root/wipe_disk.txt

```

outputs:

```

OS::stack_id:
  value: {get_resource: userdata}

```

## post\_config.yaml

```

resource_registry:

  OS::TripleO::NodeExtraConfigPost: nameserver_ntp.yaml

parameter_defaults:

  nameserver_ip: 8.8.8.8

```

## nameserver\_ntp.yaml

```
heat_template_version: 2014-10-16
```

```
description: >
  Extra hostname configuration
```

```
parameters:
  servers:
    type: json
  nameserver_ip:
    type: string

```

```
resources:
  ExtraConfig:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config:
        str_replace:
          template: |
            #!/bin/sh
            { for i in 1
            do
            x=`cat /etc/resolv.conf | grep -v '^#' | grep -v '^$' | grep nameserver`;
            echo $x
            if [[ "$x" != *nameserver* ]]; then echo "nameserver 8.8.8.8" >>
/etc/resolv.conf; fi
            done } > /root/nameserver.txt
            { for i in 1
            do
            y=`cat /etc/ntp.conf | egrep -v '^#|^$' | grep "171.68.38.66"`
            echo $y
            if [[ "$y" != *171.68.38.66* ]]; then echo "server 171.68.38.66" >>
/etc/ntp.conf; service ntpd restart;fi

```

```

done } > /root/ntp.txt
params:
  _NAMESERVER_IP_: {get_param: nameserver_ip}

ExtraDeployments:
  type: OS::Heat::SoftwareDeployments
  properties:
    servers: {get_param: servers}
    config: {get_resource: ExtraConfig}
    actions: ['CREATE', 'UPDATE']

```

## run.sh

```

#!/bin/bash
openstack overcloud deploy --templates \
-e /usr/share/openstack-tripleo-heat-templates/overcloud-resource-registry-puppet.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /home/stack/templates/network-environment.yaml \
-e /home/stack/templates/storage-environment.yaml \
-e /home/stack/templates/cisco-plugins.yaml \
-e /home/stack/templates/post_config.yaml \
--control-flavor control --compute-flavor compute --ceph-storage-flavor CephStorage \
--compute-scale 6 --control-scale 3 --ceph-storage-scale 3 \
--libvirt-type kvm \
--ntp-server 171.68.38.66 \
--neutron-network-type vlan \
--neutron-tunnel-type vlan \
--neutron-bridge-mappings datacentre:br-ex,physnet-tenant:br-tenant,floating:br-
floating \
--neutron-network-vlan-ranges physnet-tenant:250:749,floating:160:160 \
--neutron-disable-tunneling --timeout 300 \
--rhel-reg --reg-method portal --reg-org <Your Org Code> --reg-activation-key <Your Ac-
tivation Key> \
--verbose --debug --log-file overcloud_new.log

```

## create\_network\_router.sh

```

neutron net-create ext-net --router:external true --provider:physical_network floating
--provider:network_type vlan --provider:segmentation_id 160

```

```

neutron subnet-create --name ext-subnet --enable_dhcp=False \
--allocation-pool start=10.22.160.20,end=10.22.175.200 --gateway 10.22.160.1 ext-net
10.22.160.0/20

```

## create\_vm.sh

The following script creates 1 Tenant, with **2 Networks** and **4VM's** for the tenant. This can be looped to created multiple tenants. Please proofread before running the script. There are few hardcodings done.

```

#!/bin/bash
export NW1=$1
export NW2=$(( $1+50 ))
export id=$2
inst1=tenant${id}_${NW1}_inst1

```

```

inst2=tenant${id}_${NW1}_inst2
inst3=tenant${id}_${NW2}_inst3
inst4=tenant${id}_${NW2}_inst4
export TENANT_CIDR1=10.2.${NW1}.0/24
export TENANT_CIDR2=10.2.${NW2}.0/24
source /home/stack/overcloudrc
KEYSTONE_URL=${OS_AUTH_URL}
#rm -f keystone*
#rm -f tenant*
if [[ ! -f keystonerc_tenant${id} ]]
then
# create tenantdemo environment
openstack user create --password tenant${id} tenant${id}
openstack project create tenant${id}
openstack role add --user tenant${id} --project tenant${id} _member_
cat > keystonerc_tenant${id} << EOF
export OS_USERNAME=tenant${id}
export OS_TENANT_NAME=tenant${id}
export OS_PASSWORD=tenant${id}
export OS_CLOUDNAME=overcloud
export OS_AUTH_URL=${KEYSTONE_URL}
EOF
fi
source keystonerc_tenant${id}
env | grep OS_
# create network
neutron net-list
neutron net-create tenant${id}-${NW1}
neutron net-create tenant${id}-${NW2}
neutron subnet-create --name tenant${id}-${NW1}-subnet tenant${id}-${NW1} ${TEN-
ANT_CIDR1}
neutron subnet-create --name tenant${id}-${NW2}-subnet tenant${id}-${NW2} ${TEN-
ANT_CIDR2}
neutron router-create tenant${id}
subID1=$(neutron subnet-list | awk "/tenant${id}-${NW1}-subnet/ {print \$2}")
neutron router-interface-add tenant${id} $subID1
subID2=$(neutron subnet-list | awk "/tenant${id}-${NW2}-subnet/ {print \$2}")
neutron router-interface-add tenant${id} $subID2

for i in $(neutron security-group-list | awk ' /default/ { print $2 } ')
do
    neutron security-group-rule-create --direction ingress --protocol icmp $i
    neutron security-group-rule-create --direction ingress --protocol tcp --
port_range_min 22 --port_range_max 22 $i
    neutron security-group-show $i
    openstack security group show $i
done

openstack keypair create tenant${id}kp > tenant${id}kp.pem
chmod 600 tenant${id}kp.pem

netname1=`neutron net-list | grep tenant${id}-${NW1} | awk '{print $2}'`
openstack server create --flavor m1.demo --image rhel7 \
--key-name tenant${id}kp --nic net-id=${netname1} ${inst1}
openstack server create --flavor m1.demo --image rhel7 \
--key-name tenant${id}kp --nic net-id=${netname1} ${inst2}

netname2=`neutron net-list | grep tenant${id}-${NW2} | awk '{print $2}'`
openstack server create --flavor m1.demo --image rhel7 \

```

```

--key-name tenant${id}kp --nic net-id=${netname2} ${inst3}
openstack server create --flavor ml.demo --image rhel7 \
--key-name tenant${id}kp --nic net-id=${netname2} ${inst4}

while [[ $(openstack server list | grep BUILD) ]]
do
    sleep 3
done
openstack server list
source /home/stack/overcloudrc
netid=$(neutron net-list | awk "/ext-net/ { print \$2 }")
neutron router-gateway-set tenant${id} ${netid}

source keystonerc_tenant${id}

openstack ip floating create ext-net
sleep 3
float_ip=$(openstack ip floating list | grep ext-net | grep None | awk '{print $6}' |
sort -u | head -1)
openstack ip floating add ${float_ip} ${inst1}
openstack ip floating create ext-net
sleep 3
float_ip=$(openstack ip floating list | grep ext-net | grep None | awk '{print $6}' |
sort -u | head -1)
openstack ip floating add ${float_ip} ${inst2}
openstack ip floating create ext-net
sleep 5
float_ip=$(openstack ip floating list | grep ext-net | grep None | awk '{print $6}' |
sort -u | head -1)
openstack ip floating add ${float_ip} ${inst3}
openstack ip floating create ext-net
sleep 5
float_ip=$(openstack ip floating list | grep ext-net | grep None | awk '{print $6}' |
sort -u | head -1)
openstack ip floating add ${float_ip} ${inst4}

# Optionally create a volume
#openstack volume create --size 100 test
#sleep 15
#volid=$(openstack volume list | awk ' /test/ { print $2 } ')
#echo -e "volid = $volid"

# attach the volume to inst1
#openstack server add volume inst1 $volid
#sleep 15
#volname=$(openstack volume list | awk '/inst1/ {print $14}')
#ssh -i tenantdemo.pem cloud-user@$float_ip grep $(basename $volname) /proc/partitions

```

## boot-from-volume.json

```

{% set flavor_name = flavor_name or "ml.tiny" %}
{% set volume_size = volume_size or "1" %}
{% set image_name = image_name or "cirros" %}
{% set number_of_vms = number_of_vms or "1000" %}
{% set concurrency = concurrency or "3" %}
{% set no_of_tenants = no_of_tenants or "200" %}
{% set users_per_tenants = users_per_tenants or "2" %}

```

```

{
  "NovaServers.boot_server_from_volume": [
    {
      "args": {
        "flavor": {
          "name": "{{flavor_name}}"
        },
        "image": {
          "name": {{image_name}}
        },
        "volume_size": "{{volume_size}}",
      },
      "runner": {
        "type": "constant",
        "times": {{number_of_vms}},
        "concurrency": {{concurrency}}
      },
      "context": {
        "users": {
          "tenants": {{no_of_tenants}},
          "users_per_tenant": {{users_per_tenants}}
        },
        "network": {
          "start_cidr": "10.10.0.0/18"
        },
        "quotas": {
          "cinder": {
            "volumes": -1,
            "gigabytes": -1,
            "snapshots": -1
          },
          "nova": {
            "instances": -1,
            "ram": -1,
            "cores": -1
          },
          "neutron": {
            "network": -1,
            "subnet": -1,
            "router": -1,
            "port": -1
          }
        }
      }
    }
  ]
}

```

## Appendix B

This Appendix covers the changes made to few of the yaml files when floating ip configuration wasn't used. These files should be used for reference only.

### network-environment.yaml

```
resource_registry:
  OS::TripleO::NodeUserData:
    /home/stack/templates/wipe_disk.yaml
  OS::TripleO::Compute::Net::SoftwareConfig:
    /home/stack/templates/nic-configs/compute.yaml
  OS::TripleO::Controller::Net::SoftwareConfig:
    /home/stack/templates/nic-configs/controller.yaml
  OS::TripleO::CephStorage::Net::SoftwareConfig:
    /home/stack/templates/nic-configs/ceph-storage.yaml

parameter_defaults:
  InternalApiNetCidr: 192.168.10.0/24
  StorageNetCidr: 192.168.120.0/24
  StorageMgmtNetCidr: 192.168.130.0/24
  TenantNetCidr: 10.0.0.0/12
  ExternalNetCidr: 172.22.166.0/24
  InternalApiAllocationPools: [{'start': '192.168.10.200', 'end': '192.168.10.250'}]
  StorageAllocationPools: [{'start': '192.168.120.50', 'end': '192.168.120.250'}]
  StorageMgmtAllocationPools: [{'start': '192.168.130.50', 'end': '192.168.130.250'}]
  TenantAllocationPools: [{'start': '10.0.0.10', 'end': '10.15.255.250'}]
  ExternalAllocationPools: [{'start': '172.22.166.171', 'end': '172.22.166.180'}]
  ExternalNetworkVlanID: 166
  InternalApiNetworkVlanID: 100
  StorageNetworkVlanID: 120
  StorageMgmtNetworkVlanID: 130
  ControlPlaneSubnetCidr: "24"
  ControlPlaneDefaultRoute: 192.168.110.105
  EC2MetadataIp: 192.168.110.105
  DnsServers: ['171.70.168.183']
  ExternalInterfaceDefaultRoute: "172.22.166.1"
# Set to "br-ex" if using floating IPs on native VLAN on bridge br-ex
  NeutronExternalNetworkBridge: ""
```

### controller.yaml

```
heat_template_version: 2015-04-30
```

```
description: >
```

```
Software Config to drive os-net-config with 2 bonded nics on a bridge
with a VLANs attached for the controller role.
```

```
parameters:
```

```
ExternalIpSubnet:
  default: ''
  description: IP address/subnet on the external network
  type: string
```

```

InternalApiIpSubnet:
  default: ''
  description: IP address/subnet on the internal API network
  type: string
StorageIpSubnet:
  default: ''
  description: IP address/subnet on the storage network
  type: string
StorageMgmtIpSubnet:
  default: ''
  description: IP address/subnet on the storage mgmt network
  type: string
TenantIpSubnet:
  default: ''
  description: IP address/subnet on the tenant network
  type: string
BondInterfaceOvsOptions:
  default: ''
  description: The ovs_options string for the bond interface. Set things like
    lacp=active and/or bond_mode=balance-slb using this option.
  type: string
ExternalNetworkVlanID:
  default: 166
  description: Vlan ID for the external network traffic.
  type: number
InternalApiNetworkVlanID:
  default: 100
  description: Vlan ID for the internal_api network traffic.
  type: number
StorageNetworkVlanID:
  default: 120
  description: Vlan ID for the storage network traffic.
  type: number
StorageMgmtNetworkVlanID:
  default: 130
  description: Vlan ID for the storage mgmt network traffic.
  type: number
TenantNetworkVlanID:
  default: 140
  description: Vlan ID for the tenant network traffic.
  type: number
ExternalInterfaceDefaultRoute:
  default: '172.22.166.1'
  description: default route for the external network
  type: string
ControlPlaneIp:
  default: ''
  description: IP address/subnet on the ctlplane network
  type: string
ControlPlaneSubnetCidr:
  default: '24'
  description: The subnet CIDR of the control plane network.
  type: string
ControlPlaneDefaultRoute:
  default: '192.168.110.105'
  description: The Control Plane Default route.
  type: string
DnsServers:
  default: ['171.70.168.183']

```

```

    description: A list of DNS servers (2 max) to add to resolv.conf.
    type: json
  EC2MetadataIp:
    description: The IP address of the EC2 metadata server.
    type: string

resources:
  OsNetConfigImpl:
    type: OS::Heat::StructuredConfig
    properties:
      group: os-apply-config
      config:
        os_net_config:
          network_config:
            -
              type: interface
              name: nic1
              use_dhcp: false
              dns_servers: {get_param: DnsServers}
              addresses:
                -
                  ip_netmask:
                    list_join:
                      - '/'
                  - {get_param: ControlPlaneIp}
                  - {get_param: ControlPlaneSubnetCidr}
              routes:
                -
                  ip_netmask: 169.254.169.254/32
                  next_hop: {get_param: EC2MetadataIp}
            -
              type: ovs_bridge
              name: {get_input: bridge_name}
              use_dhcp: false
              members:
                -
                  type: interface
                  name: nic4
                  primary: true
                -
                  type: vlan
                  vlan_id: {get_param: ExternalNetworkVlanID}
                  addresses:
                    -
                      ip_netmask: {get_param: ExternalIpSubnet}
                  routes:
                    -
                      ip_netmask: 0.0.0.0/0
                      next_hop: {get_param: ExternalInterfaceDefaultRoute}
            -
              type: ovs_bridge
              name: br-mgmt
              members:
                -
                  type: interface
                  name: nic3
                  primary: true
                -
                  type: vlan

```

```

      vlan_id: {get_param: InternalApiNetworkVlanID}
      addresses:
      -
        ip_netmask: {get_param: InternalApiIpSubnet}
-
  type: ovs_bridge
  name: br-storage-pub
  mtu: 9000
  members:
  -
    type: interface
    name: nic5
    mtu: 9000
    primary: true
  -
    type: vlan
    mtu: 9000
    vlan_id: {get_param: StorageNetworkVlanID}
    addresses:
    -
      ip_netmask: {get_param: StorageIpSubnet}
-
  type: ovs_bridge
  name: br-storage-clus
  mtu: 9000
  members:
  -
    type: interface
    name: nic6
    mtu: 9000
    primary: true
  -
    type: vlan
    mtu: 9000
    vlan_id: {get_param: StorageMgmtNetworkVlanID}
    addresses:
    -
      ip_netmask: {get_param: StorageMgmtIpSubnet}

```

#### outputs:

```

OS::stack_id:
  description: The OsNetConfigImpl resource.
  value: {get_resource: OsNetConfigImpl}

```

#### run.sh

```

#!/bin/bash
openstack overcloud deploy --templates \
-e /usr/share/openstack-tripleo-heat-templates/overcloud-resource-registry-puppet.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /home/stack/templates/network-environment.yaml \
-e /home/stack/templates/storage-environment.yaml \
-e /home/stack/templates/cisco-plugins.yaml \
-e /home/stack/templates/post_config.yaml \
--control-flavor control --compute-flavor compute --ceph-storage-flavor CephStorage \
--compute-scale 3 --control-scale 3 --ceph-storage-scale 3 \
--libvirt-type kvm \

```

```
--ntp-server <ntp server ip> \  
--neutron-network-type vlan \  
--neutron-tunnel-type vlan \  
--neutron-bridge-mappings datacentre:br-ex,physnet-tenant:br-tenant \  
--neutron-network-vlan-ranges physnet-tenant:250:749 \  
--neutron-disable-tunneling --timeout 300 \  
  --verbose --debug --log-file overcloud_new.log
```

## About the Authors

---

Ramakrishna Nishtala, Cisco Systems, Inc.

Ramakrishna Nishtala is a Technical Leader in Cisco UCS and Data Center solutions group and has over 20 years of experience in IT infrastructure, Virtualization and Cloud computing. In his current role at Cisco Systems, he works on best practices, optimization and performance tuning on OpenStack and other Open Source Storage solutions on Cisco UCS platforms.

Vijay Durairaj, Cisco Systems, Inc.

Vijay Durairaj is a Technical Marketing Engineer with Cisco UCS Performance and Solutions Data Center Group. Vijay has over 12 years of experience in IT Infrastructure, Server Virtualization and Cloud Computing. His focus area includes Unified Computing Systems, Network, Storage and Performance benchmarking on Cisco UCS Platforms. Vijay also holds Cisco Unified Computing Design Certification.

Patryk Wolsza, Intel

Patryk Wolsza is a Data Center Architect in the Intel Cloud Platforms Group, with a focus on Software Defined Infrastructure. With more than 12 years of expertise in different virtualization and cloud platforms, Patryk has broad experience in cloud solutions, system designs, influencing data center designs and understanding connections between ordinary Data Centers, virtualization and SDI. He believes that mastering the purpose of existing cloud solutions is the key to deliver and maintain the complete product, hardware and software, for any demand.

Pawel Koniszewski, Intel

Pawel Koniszewski is Software Engineer at Intel in the Cloud Platforms Group. He started his career at Intel during his studies, and from the beginning has been focused on Cloud Computing. In his current role he is a member of the Software Defined Infrastructure team and is focused on enterprise readiness of cloud-based solutions. His primary area of interest is preserving SLA of instances in virtualized environments. Pawel is interested in building reliable distributed systems, optimizing performance and increasing user experience. Pawel also works with Intel partners on building cloud ecosystems based on OpenStack

Guil Barros, Red Hat

Guil Barros, is a Principal Product Manager at Red Hat in the RHEL OpenStack Platform group. His focus is on working with partners to create interesting solutions to customer problems. Guil has a passion for finding the sweet spot that ties existing technologies together into truly useful implementations. In his career, this has spanned all the way from understanding how to best interact with users, to architecting support strategy, to bringing partners to innovate together.

Karthik Prabhakar, Red Hat

Karthik Prabhakar, helps develop joint cloud solutions with Red Hat's strategic partners (including Cisco and Intel), and facilitates successful market adoption through leadership of GTM initiatives in collaboration with worldwide field and partner teams, and leading-edge customers. Further background at <http://www.linkedin.com/in/worldhopper>

Steven Reichard, Red Hat

Steven Reichard is a consulting engineer and manager in Red Hat's System's Design and Engineering group. This team's mission is to eliminate roadblocks to the wider adoption & ease-of-use of our product portfolio to solve ever more demanding customer/partner solutions. Most recently Steve has focused on Red Hat Enterprise Linux OpenStack Platform including enabling partner solutions based on RHEL-OSP and Red Hat Ceph Storage. Steve is a Red Hat Certified Engineer (RHCE) who has more than 20 years of computer industry experience.

## Acknowledgements

- Cisco Systems, Inc.: Brian Demers, Damien Zhao, Michael Wang, Mohsen Mortazavi, Muhammad Afzal, Sandhya Dasu, Shanthi Kumar Adloori, Shree Lokare, Steven Hillman, Timothy Swanson, Vishwanath Jakka
- Intel: Arek Chylinski, Das Kamhout, Kamil Rogon, Marcin Karkocha, Marek Strachacki
- Red Hat: Andrew Beekhof, Dan Sneddon, Dmitry Tantsur, Fabio Di Nitto, Marek Grac, Mike Orazi, Steve Hardy, Tushar Katarki