



FlexPod Datacenter with Red Hat Enterprise Linux OpenStack Platform

Deployment Guide for FlexPod Datacenter with Red Hat Enterprise Linux OpenStack Platform 6

Last Updated: February 11, 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 this Document	11
Solution Summary	11
Technology Overview	12
FlexPod System Overview	12
FlexPod Benefits	12
FlexPod: Cisco and NetApp Verified Architecture	12
Integrated System	13
Out of the Box Infrastructure High Availability	13
FlexPod Design Principles	13
Cisco Unified Computing System (UCS)	13
Cisco UCS 6248UP Fabric Interconnects	14
Cisco UCS 5108 Blade Server Chassis	15
Cisco UCS Fabric Extenders	15
Cisco UCS B200 M4 Servers	16
Cisco VIC 1340	16
Cisco UCS Differentiators	17
Cisco UCS for OpenStack	18
Cisco Nexus 9000 Series Switch	19
Cisco Nexus 1000v for KVM - OpenStack	20
Cisco Nexus 1000V for OpenStack Solution Offers	20
Cisco Nexus 1000V Components	21
ML2 Mechanism Driver for Cisco Nexus 1000v	22
NetApp FAS8000	22
NetApp Storage Controllers	23
NetApp Clustered Data ONTAP 8.3 Fundamentals	24
Scale Out	25
Non-disruptive Operations	25
Availability	26
NetApp Advanced Data Management Capabilities	27

Storage Virtual Machines	29
NetApp E5000 Series	30
NetApp E-Series Storage Controllers	30
NetApp SANtricity Operating System Fundamentals	31
Dynamic Disk Pools	32
NetApp Storage for OpenStack	32
Cinder	33
Swift	35
Glance	37
Nova	39
Manila	39
Domain and Management Software	40
Cisco UCS Manager	40
NetApp OnCommand System Manager	40
NetApp SANtricity Storage Manager	41
Red Hat Enterprise Linux OpenStack Platform Installer	41
Red Hat Enterprise Linux	41
Red Hat Enterprise Linux OpenStack Platform	42
OpenStack Services	42
Heat Templates	43
OpenStack High Availability	43
Other OpenStack Supporting Technologies	45
OpenStack Networking	46
Solution Design	47
Hardware and Software Revisions	47
Solution Components	48
Architectural Overview	48
OpenStack Platform Architecture	50
Prerequisites	50
Deployment Hardware and Software	51
Physical Infrastructure	51
FlexPod with Red Hat Enterprise Linux OpenStack Platform Physical Topology with Cabling	51
FlexPod Cabling Detail	52
Configuration Guidelines	55
Required VLANs	55

Logical Topology	56
Service Profile and vNIC Templates	58
Server Pools	59
NetApp FAS Storage Configuration.....	59
Disk Shelves	60
Clustered Data ONTAP 8.3	60
Create Cluster on Node 01	63
Join Node 02 to Cluster	66
Configure Initial Cluster Settings.....	68
Assign Disks for Optimal Performance.....	68
Zero All Spare Disks	69
Create Aggregates	69
Verify Storage Failover.....	70
Set Onboard UTA2 Ports Personality	70
Disable Flow Control on 10GbE and UTA2 Ports.....	72
Create LACP Interface Groups.....	72
Create VLANs.....	73
Enable Cisco Discovery Protocol	73
Set Auto-Revert on Cluster Management Interface.....	74
Create Failover Group for Cluster Management	74
Assign Cluster Management Failover Group to Cluster Management LIF.....	74
Configure NTP	74
Configure Simple Network Management Protocol.....	74
Configure SNMPv1 Access.....	75
Create SNMPv3 User	75
Configure AutoSupport HTTPS	75
Configure Remote Support Agent.....	76
Configure HTTPS Access	76
Create Operational Storage Virtual Machine	76
Create Load-Sharing Mirror of SVM Root Volume.....	77
Create Infrastructure Storage Virtual Machine.....	79
Create Load-Sharing Mirror of SVM Root Volume.....	80
Create Failover Group for SVM Management.....	80
Configure NFSv4	80
Create Flexible Volumes (FlexVol) for Cinder and Glance.....	81

Enable Deduplication on Glance Volume	81
Create additional FlexVol Volumes for Cinder (Optional)	82
NFS Logical Interfaces	82
Start NFS Server and Enable Advanced NFS Options.....	82
Gather Target Information from FAS8040	83
NetApp E-Series Storage Configuration.....	83
Disk Shelves	84
SANtricity OS.....	84
Initial Configuration of Management Interfaces	85
Disk Pool Creation	87
Volume Creation	90
Host Mapping	91
LUN Mapping.....	93
Configure iSCSI Host Ports	95
Write Cache Mirroring.....	97
Cisco UCS and Server Configuration	98
Initial Setup of UCS 6248 Fabric Interconnect.....	98
Cisco UCS Configuration	100
Cisco UCS Manager	100
Edit Chassis Discovery Policy	100
Enable Server and Uplink Ports.....	101
Acknowledge Cisco UCS Chassis.....	104
Upgrade Cisco UCS Version to 2.2(3g).....	104
Synchronize Cisco UCS to NTP.....	105
Create Uplink Port Channels to Cisco Nexus Switch	105
Add Block of IP Address for KVM Access	108
Create MAC Address Pools	109
Create IQN Pools for iSCSI Boot	112
Create iSCSI Initiator IP Pools for iSCSI Boot	114
Create UUID Suffix Pools	117
Create Server Pools.....	118
Create VLANs.....	122
Create Host Firmware Package	133
Set Jumbo Frames in Cisco UCS	134
Create Local Disk Configuration Policy	135

Create Network Control Policy for Cisco Discovery Protocol.....	136
Create Power Control Policy.....	137
Create BIOS Policy	138
Create vNIC Placement Policy for Red Hat Enterprise Linux Hosts	139
Create Maintenance Policy	140
Create vNIC Templates.....	141
Create Boot Policy.....	156
Create Service Profile Templates.....	158
Verify vNIC Placement	189
Create Service Profile	191
Service Profile for Controller Nodes.....	191
Service Profile for Compute Nodes.....	191
Service Profile for RHEL-OSP Installer Server.....	192
Gather Necessary Information from Cisco UCS Manager	193
Network Configuration.....	194
Cisco Nexus 9000 Network Initial Configuration Setup	195
Enable Appropriate Cisco Nexus 9000 Features and Settings.....	199
Create VLAN.....	200
Configure Virtual Port Channel Domain	201
Configure Network Interfaces for vPC Peer Link	203
Configure Network Interfaces Connected to Fabric Interconnect	204
Configure Network Interfaces Connected to NetApp FAS8040	208
Configure Interfaces Connected to NetApp E5560	212
Create SVI Interfaces.....	214
Uplink to Existing Network Infrastructure.....	215
Red Hat Enterprise Linux OpenStack Platform Installer Setup	215
Login to Cisco UCS Manager.....	215
Setup RHEL-OSP Installer Server	215
Install RHEL 7.1	216
RHEL-OSP Installer Prerequisites	222
Update Cisco eNIC Driver.....	225
Install RHEL-OSP Installer Packages.....	226
Install RHEL-OSP Installer.....	226
Prepare the RHEL 7.1 Installation Medium	235
Prepare Downloadable URL for Cisco eNIC Driver	236

Boot Servers into Discovery Mode	237
OpenStack Deployment using the RHEL-OSP Installer	239
Prerequisites	239
Install Worksheet for Controller and Compute Roles	242
Setup Subnets	244
Modify Puppet classes for Cisco Nexus 1000v VEM and VSM Module Parameters	254
Deployment	257
Post-Deployment (Required)	277
Deploy Cisco Nexus 1000v Virtual Supervisor Module (VSM)	278
Cisco Nexus 1000v Configuration	287
Jumbo Frames for NFS Traffic and Bond0	288
Storage Service Catalog	289
Quality of Service	290
NetApp Copy Offload Tool	291
Swift Deployment	292
Create Network Interfaces for Swift Traffic	293
Log into the E-Series Array by Using iSCSI	294
Swift Package Installation	294
Keystone User and Role Configuration	295
IPTables Firewall Exceptions	295
Partition, File System, and Directory Structure Creation	295
Resource Mounts and Permissions	297
Rsync Replication Between Nodes	298
Configure Swift	299
Build Swift Rings and Start Proxy Service	301
High-Availability for Swift	303
Keystone Entry for Swift	303
Pacemaker Configuration for Swift	303
Verification	304
Use Cases	305
Scaling the Environment	305
OpenStack Admin Use Cases	310
OpenStack Tenant Use Cases	317
Bill of Materials	331
Other Resources	334

Cisco UCS	334
Cisco Nexus Networking.....	334
NetApp FAS Storage	335
NetApp E-Series Storage	335
Red Hat Enterprise Linux OpenStack Platform 6.....	335
OpenStack Upstream.....	336
Appendix	337
Appendix A - Cisco Nexus 9000 Configuration Files.....	337
Cisco Nexus 9372 A.....	337
Cisco Nexus 9372 B.....	343
Appendix B: HTTPS Access in Clustered Data ONTAP.....	350
Appendix C: Changing Installer GUI Password	351
Appendix D: Red Hat Enterprise Linux OpenStack Platform Installer Server iptables Configuration	352
Appendix E: Kickstart default PXELinux Configuration.....	353
Appendix F: Kickstart RHEL default Configuration.....	353
Appendix G: Full List of VIPs, Users, and Database Passwords	356
VIP List	356
User Passwords.....	358
Database Passwords	358
Appendix H: Rebuilding a Server	359
Appendix I: Restart Deployment.....	360
Appendix J: Red Hat Enterprise Linux OpenStack Platform Installer Server Interfaces Configuration.....	361
About the Authors.....	365
Acknowledgements	365



Executive Summary

Cisco Validated Designs 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 FlexPod® solution portfolio combines NetApp® storage systems, Cisco® Unified Computing System servers, and Cisco Nexus fabric into a single, flexible architecture. FlexPod datacenter can scale up for greater performance and capacity or scale out for environments that require consistent, multiple deployments.

FlexPod is a leading converged infrastructure supporting broad range of enterprise workloads and use cases. With the growing interest, continuous evolution, and popular acceptance of OpenStack there has been an increased customer demand to have OpenStack platform validated on FlexPod, and be made available for Enterprise private cloud, as well as other OpenStack based Infrastructure as a Service (IaaS) cloud deployments. To accelerate this process and simplify the evolution to a shared cloud infrastructure, Cisco, NetApp, and Red Hat have developed a validated solution, FlexPod with Red Hat Enterprise Linux OpenStack Platform (RHEL OSP) 6.0. This solution enables customers to quickly and reliably deploy OpenStack based private and hybrid cloud on converged infrastructure while offering FlexPod Cooperative Support Model that provides both OpenStack and FlexPod support.

The recommended solution architecture is built on Cisco UCS B200 M4 Blade Servers, Cisco Nexus 9000 Series switches, and NetApp FAS8000 Series and E5500 Series storage arrays. In addition to that, it includes Red Hat Enterprise Linux 7.1, Red Hat Enterprise Linux OpenStack Platform 6, and the Red Hat Enterprise Linux OpenStack Platform Installer.

Solution Overview

Introduction

FlexPod is a pre-validated datacenter architecture followed by best practices that is built on the Cisco Unified Computing System (UCS), the Cisco Nexus® family of switches, and NetApp unified storage systems. FlexPod has been a trusted platform for running a variety of virtualization hypervisors as well as bare metal operating systems. The FlexPod architecture is highly modular, delivers a baseline configuration, and also has the flexibility to be sized and optimized to accommodate many different use cases and requirements. The FlexPod architecture can both scale up (adding additional resources within a FlexPod unit) and scale out (adding additional FlexPod units). FlexPod with Red Hat Enterprise Linux OpenStack Platform 6.0 is an extension to FlexPod's already wide-range of validated and supported design portfolio entries.

Audience

The audience of this document includes, but is not limited to, sales engineers, field consultants, professional services, IT managers, partner engineers, and customers who want to take advantage of an infrastructure that is built to deliver IT efficiency and enable IT innovation.

It is expected from the audience of this document to have the necessary training and background to install and configure Red Hat Enterprise Linux, Cisco Unified Computing System (UCS), Cisco Nexus switches, and NetApp storage as well as high level understanding of OpenStack components. External references are provided where applicable and it is recommended that the audience be familiar with these documents.

Purpose of this Document

This document describes the steps required to deploy and configure Red Hat Enterprise Linux OpenStack Platform 6 on FlexPod. The architecture can be very easily expanded with predictable linear performance. While readers of this document are expected to have sufficient knowledge to install and configure the products used, configuration details **that are important to this solution's deployments are specifically** mentioned.

Solution Summary

This solution is based on OpenStack “Juno” release hardened and streamlined by Red Hat in Red Hat Enterprise Linux OpenStack Platform 6.0. In FlexPod with Red Hat Enterprise Linux OpenStack Platform 6, Cisco Unified Computing System, NetApp, and Red Hat OpenStack Platform are combined to deliver OpenStack Infrastructure as a Service (IaaS) deployment that is quick and easy to deploy.

FlexPod with Red Hat Enterprise Linux OpenStack Platform helps IT organizations accelerate cloud deployments while retaining control and choice over their environments with open and inter-operable cloud solutions. FlexPod with Red Hat Enterprise Linux OpenStack Platform 6.0 offers fully redundant system architecture for the compute, network, and storage infrastructure. Furthermore, it includes OpenStack HA through redundant controller nodes. In this solution, OpenStack block, file, and object storage is provided by highly available NetApp storage systems.

Technology Overview

FlexPod System Overview

FlexPod is a best practice datacenter architecture that includes these components:

- Cisco Unified Computing System (Cisco UCS)
- Cisco Nexus switches
- NetApp fabric-attached storage (FAS) and/or NetApp E-Series storage systems

These components are connected and configured according to best practices of both Cisco and NetApp, and provide the ideal platform for running a variety of enterprise workloads with confidence. As previously mentioned, the reference architecture covered in this document leverages the Cisco Nexus 9000 Series switch. One of the key benefits of FlexPod is the ability to maintain consistency at scaling, including scale up and scale out. Each of the component families shown in 0 (Cisco Unified Computing System, Cisco Nexus, and NetApp storage systems) offers platform and resource options to scale the infrastructure up or down, while supporting the same features and functionality that are required under the configuration and connectivity best practices of FlexPod.

FlexPod Benefits

As customers transition toward shared infrastructure or cloud computing they face a number of challenges such as initial transition hiccups, return on investment (ROI) analysis, infrastructure management and future growth plan. The FlexPod architecture is designed to help with proven guidance and measurable value. By introducing standardization, FlexPod helps customers mitigate the risk and uncertainty involved in planning, designing, and implementing a new datacenter infrastructure. The result is a more predictive and adaptable architecture capable of meeting and exceeding customers' IT demands.

FlexPod: Cisco and NetApp Verified Architecture

Cisco and NetApp have thoroughly validated and verified the FlexPod solution architecture and its many use cases while creating a portfolio of detailed documentation, information, and references to assist customers in transforming their datacenters to this shared infrastructure model. This portfolio includes, but is not limited to the following items:

- Best practice architectural design
- Workload sizing and scaling guidance
- Implementation and deployment instructions
- Technical specifications (rules for FlexPod configuration **dos and don'ts**)
- Frequently asked questions (FAQs)
- Cisco Validated Designs (CVDs) and NetApp Verified Architectures (NVAs) focused on a variety of use cases

Cisco and NetApp have also built a robust and experienced support team focused on FlexPod solutions, from customer account and technical sales representatives to professional services and technical support engineers. The Co-operative Support Program extended by NetApp, Cisco and Red Hat provides customers and channel service partners with direct access to technical experts who collaborate with cross vendors and have access to shared lab resources to resolve potential issues. FlexPod supports tight integration with virtualized and cloud infrastructures, making it a logical choice for long-term investment. The following IT initiatives are addressed by the FlexPod solution.

Integrated System

FlexPod is a pre-validated infrastructure that brings together compute, storage, and network to simplify, accelerate, and minimize the risk associated with datacenter builds and application rollouts. These integrated systems provide a standardized approach in the datacenter that facilitates staff expertise, application onboarding, and automation as well as operational efficiencies relating to compliance and certification.

Out of the Box Infrastructure High Availability

FlexPod is a highly available and scalable infrastructure that IT can evolve over time to support multiple physical and virtual application workloads. FlexPod has no single point of failure at any level, from the server through the network, to the storage. The fabric is fully redundant and scalable, and provides seamless traffic failover, should any individual component fail at the physical or virtual layer.

FlexPod Design Principles

FlexPod addresses four primary design principles:

- Application availability: Makes sure that services are accessible and ready to use.
- Scalability: Addresses increasing demands with appropriate resources.
- Flexibility: Provides new services or recovers resources without requiring infrastructure modifications.
- Manageability: Facilitates efficient infrastructure operations through open standards and APIs.



Performance and comprehensive security are key design criteria that are not directly addressed in this solution but have been addressed in other collateral, benchmarking, and solution testing efforts. This design guide validates the functionality and basic security elements.

Cisco Unified Computing System (UCS)

The Cisco Unified Computing System is a next-generation solution for blade and rack server computing. The system integrates a low-latency, lossless 10 Gigabit Ethernet unified network fabric with enterprise-class, x86-architecture servers. The 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. The Cisco Unified Computing System consists of the following components:

- Compute - The system is based on an entirely new class of computing system that incorporates rack mount and blade servers based on Intel Xeon 2600 v2 Series Processors.
- Network - The system is integrated onto a low-latency, lossless, 10-Gbps unified network fabric. This network foundation consolidates LANs, SANs, and high-performance computing networks which are separate networks today. The unified fabric lowers costs by reducing the number of network adapters, switches, and cables, and by decreasing the power and cooling requirements. Virtualization - The system unleashes the full potential of virtualization by enhancing the scalability, performance, and operational control of virtual environments. Cisco security, policy enforcement, and diagnostic features are now extended into virtualized environments to better support changing business and IT requirements. Storage access - The system provides consolidated access to both SAN storage and Network Attached Storage (NAS) over the unified fabric. By unifying the storage access the Cisco Unified Computing System can access storage over Ethernet (SMB 3.0 or iSCSI), Fibre Channel, and Fibre Channel over Ethernet (FCoE). This provides customers with storage choices and investment protection. In addition, the server administrators can pre-assign storage-access policies to storage resources, for simplified storage connectivity and management leading to increased productivity.
- Management - The system uniquely integrates all system components to enable the entire solution to be managed as a single entity by the Cisco UCS Manager. The Cisco UCS Manager has an intuitive graphical user interface (GUI), a command-line interface (CLI), and a powerful scripting library module for Microsoft PowerShell built on a robust application programming interface (API) to manage all system configuration and operations.

Cisco Unified Computing System (Cisco UCS) fuses access layer networking and servers. This high-performance, next-generation server system provides a datacenter with a high degree of workload agility and scalability.

Cisco UCS 6248UP Fabric Interconnects

- The Cisco UCS Fabric interconnects provide a single point for connectivity and management for the entire system. Typically deployed as an active-**active pair**, **the system's fabric interconnects integrate** all components into a single, highly-available management domain controlled by Cisco UCS Manager. The fabric interconnects manage all I/O efficiently and securely at a single point, resulting in **deterministic I/O latency regardless of a server or virtual machine's topological location** in the system.
- Cisco UCS 6200 **Series Fabric Interconnects support the system's 80 Gbps unified fabric** with low-latency, lossless, cut-through switching that supports IP, storage, and management traffic using a single set of cables. The fabric interconnects feature virtual interfaces that terminate both physical and virtual connections equivalently, establishing a virtualization-aware environment in which blade, rack servers, and virtual machines are interconnected using the same mechanisms. The Cisco UCS 6248UP is a 1-RU Fabric Interconnect that features up to 48 universal ports that can support 80 Gigabit Ethernet, Fiber Channel over Ethernet, or native Fiber Channel connectivity.

For more information, see: <http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-6200-series-fabric-interconnects/index.html>

Figure 1 Cisco Fabric Interconnect – Front and Rear



Cisco UCS 5108 Blade Server Chassis

The Cisco UCS 5100 Series Blade Server Chassis is a crucial building block of the Cisco Unified Computing System, delivering a scalable and flexible blade server chassis. The Cisco UCS 5108 Blade Server Chassis is six rack units (6RU) high and can mount in an industry-standard 19-inch rack. A single chassis can house up to eight half-width Cisco UCS B-Series Blade Servers and can accommodate both half-width and full-width blade form factors. Four single-phase, hot-swappable power supplies are accessible from the front of the chassis. These power supplies are 92 percent efficient and can be configured to support non-redundant, N+1 redundant and grid-redundant configurations. The rear of the chassis contains eight hot-swappable fans, four power connectors (one per power supply), and two I/O bays for Cisco UCS 2204XP or 2208XP Fabric Extenders. A passive mid-plane provides up to 40 Gbps of I/O bandwidth per server slot and up to 80 Gbps of I/O bandwidth for two slots. The chassis is capable of supporting future 80 Gigabit Ethernet standards. For more information, see: <http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-5100-series-blade-server-chassis/index.html>

Figure 2 Cisco UCS 5108 Blade Chassis

Front View



Back View



Cisco UCS Fabric Extenders

The Cisco UCS 2204XP Fabric Extender (Figure 4) has four 10 Gigabit Ethernet, FCoE-capable, SFP+ ports that connect the blade chassis to the fabric interconnect. Each Cisco UCS 2204XP has sixteen 10 Gigabit Ethernet ports connected through the mid-plane to each half-width slot in the chassis. Typically configured in pairs for redundancy, two fabric extenders provide up to 80 Gbps of I/O to the chassis.

The Cisco UCS 2208XP Fabric Extender (Figure 4) has eight 10 Gigabit Ethernet, FCoE-capable, Enhanced Small Form-Factor Pluggable (SFP+) ports that connect the blade chassis to the fabric interconnect. Each Cisco UCS 2208XP has thirty-two 10 Gigabit Ethernet ports connected through the midplane to each half-width slot in the chassis. Typically configured in pairs for redundancy, two fabric extenders provide up to 160 Gbps of I/O to the chassis.

Figure 3 Cisco UCS 2204XP/2208XP Fabric Extender



Cisco UCS 2204XP FEX



Cisco UCS 2208XP FEX

Cisco UCS B200 M4 Servers

The enterprise-class **Cisco UCS B200 M4 Blade Server** extends the capabilities of Cisco's Unified Computing System portfolio in a half-width blade form factor. The Cisco UCS B200 M4 uses the power of the latest Intel® Xeon® E5-2600 v3 Series processor family CPUs with up to 768 GB of RAM (using 32 GB DIMMs), two solid-state drives (SSDs) or hard disk drives (HDDs), and up to 80 Gbps throughput connectivity. The UCS B200 M4 Blade Server mounts in a Cisco UCS 5100 Series blade server chassis or UCS Mini blade server chassis. It has 24 total slots for registered ECC DIMMs (RDIMMs) or load-reduced DIMMs (LR DIMMs) for up to 768 GB total memory capacity (B200 M4 configured with two CPUs using 32 GB DIMMs). **It supports one connector for Cisco's VIC 1340 or 1240 adapter**, which provides Ethernet and FCoE. For more information, see: <http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-b200-m4-blade-server/index.html>

Figure 4 Cisco UCS B200 M4 Blade Server



Cisco VIC 1340

The Cisco UCS Virtual Interface Card (VIC) 1340 is a 2-port 40-Gbps Ethernet or dual 4 x 10-Gbps Ethernet, FCoE-capable modular LAN on motherboard (mLOM) designed exclusively for the M4 generation of Cisco UCS B-Series Blade Servers. When used in combination with an optional port expander, the Cisco UCS VIC 1340 capabilities is enabled for two ports of 40-Gbps Ethernet. The Cisco UCS VIC 1340 enables a policy-based, stateless, agile server infrastructure that can present over 256 PCIe standards-compliant interfaces to the host that can be dynamically configured as either network interface cards (NICs) or host bus adapters (HBAs). In addition, the Cisco UCS VIC 1340 supports Cisco® Virtual Machine Fabric Extender (VM-FEX) technology, which extends the Cisco UCS Fabric interconnect ports to virtual machines, simplifying server virtualization deployment and management. For more information, see:

<http://www.cisco.com/c/en/us/products/interfaces-modules/ucs-virtual-interface-card-1340/index.html>

Figure 5 Cisco VIC 1340



Cisco UCS Differentiators

Cisco's Unified Compute System is revolutionizing the way servers are managed in data-center. The following are the unique differentiators of Cisco UCS and Cisco UCS Manager.

1. Embedded Management –In Cisco UCS, the servers are managed by the embedded firmware in the Fabric Interconnects, eliminating need for any external physical or virtual devices to manage the servers.
2. Unified Fabric –In Cisco UCS, from blade server chassis or rack servers to FI, there is a single Ethernet cable used for LAN, SAN and management traffic. This converged I/O results in reduced cables, SFPs and adapters – reducing capital and operational expenses of overall solution.
3. Auto Discovery –By simply inserting the blade server in the chassis or connecting rack server to the fabric interconnect, discovery and inventory of compute resource occurs automatically without any management intervention. The combination of unified fabric and auto-discovery enables the wire-once architecture of UCS, where compute capability of UCS can be extended easily while keeping the existing external connectivity to LAN, SAN and management networks.
4. Policy Based Resource Classification –Once a compute resource is discovered by UCS Manager, it can be automatically classified to a given resource pool based on policies defined. This capability is useful in multi-tenant cloud computing. This CVD showcases the policy based resource classification of UCS Manager.
5. Combined Rack and Blade Server Management –Cisco UCS Manager can manage B-series blade servers and C-series rack server under the same UCS domain. This feature, along with stateless computing makes compute resources truly hardware form factor agnostic.
6. Model based Management Architecture –Cisco UCS Manager architecture and management data-base is model based and data driven. An open XML API is provided to operate on the management model. This enables easy and scalable integration of UCS Manager with other management systems.
7. Policies, Pools, Templates –The management approach in UCS Manager is based on defining policies, pools and templates, instead of cluttered configuration, which enables a simple, loosely coupled, data driven approach in managing compute, network and storage resources.
8. Loose Referential Integrity –In Cisco UCS Manager, a service profile, port profile or policies can refer to other policies or logical resources with loose referential integrity. A referred policy cannot exist at the time of authoring the referring policy or a referred policy can be deleted even though other policies are referring to it. This provides different subject matter experts to work independently from each-other. This provides great flexibility where different experts from different domains, such as network, storage, security, server and virtualization work together to accomplish a complex task.
9. Policy Resolution –In Cisco UCS Manager, a tree structure of organizational unit hierarchy can be created that mimics the real life tenants and/or organization relationships. Various policies, pools and templates can be defined at different levels of organization hierarchy. A policy referring to another policy by name is resolved in the organization hierarchy with closest policy match. If no policy with **specific name is found in the hierarchy of the root organization, then special policy named “default”** is searched. This policy resolution practice enables automation friendly management APIs and provides great flexibility to owners of different organizations.

10. Service Profiles and Stateless Computing –A service profile is a logical representation of a server, carrying its various identities and policies. This logical server can be assigned to any physical compute resource as far as it meets the resource requirements. Stateless computing enables procurement of a server within minutes, which used to take days in legacy server management systems.
11. Built-in Multi-Tenancy Support –The combination of policies, pools and templates, loose referential integrity, policy resolution in organization hierarchy and a service profiles based approach to compute resources makes UCS Manager inherently friendly to multi-tenant environment typically observed in private and public clouds.
12. Extended Memory – The enterprise-class Cisco UCS B200 M4 blade server extends the capabilities **of Cisco’s Unified Computing System portfolio in a half**-width blade form factor. The Cisco UCS B200 M4 harnesses the power of the latest Intel® Xeon® E5-2600 v3 Series processor family CPUs with up to 1536 GB of RAM (using 64 GB DIMMs) – allowing huge VM to physical server ratio required in many deployments, or allowing large memory operations required by certain architectures like big data.
13. Virtualization Aware Network –Cisco VM-FEX technology makes the access network layer aware about host virtualization. This prevents domain pollution of compute and network domains with virtualization when virtual network is managed by port-profiles defined **by the network administrators’** team. VM-FEX also off-loads hypervisor CPU by performing switching in the hardware, thus allowing hypervisor CPU to do more virtualization related tasks. VM-FEX technology is well integrated with VMware vCenter, Linux KVM and Hyper-V SR-IOV to simplify cloud management.
14. Simplified QoS –Even though Fiber Channel and Ethernet are converged in UCS fabric, built-in support for QoS and lossless Ethernet makes it seamless. Network Quality of Service (QoS) is simplified in UCS Manager by representing all system classes in one GUI panel.

Cisco UCS for OpenStack

Cloud-enabled applications can run on organization premises, in public clouds, or on a combination of the two (hybrid cloud) for greater flexibility and business agility. Finding a platform that supports all these scenarios is essential. With Cisco UCS, IT departments can take advantage of technological advancements and lower the cost of their OpenStack deployments.

1. Open Architecture—A market-leading, open alternative to expensive, proprietary environments, the simplified architecture of Cisco UCS running OpenStack software delivers greater scalability, manageability, and performance at a significant cost savings compared to traditional systems, both in the datacenter and the cloud. Using industry-standard x86-architecture servers and open source software, IT departments can deploy cloud infrastructure today without concern for hardware or software vendor lock-in.
2. Accelerated Cloud Provisioning—Cloud infrastructure must be able to flex on demand, providing **infrastructure to applications and services on a moment’s notice**. **Cisco UCS simplifies and accelerates** cloud infrastructure deployment through automated configuration. The abstraction of Cisco Unified Compute System Integrated Infrastructure for Red Hat Enterprise Linux server identity, personality, and I/O connectivity from the hardware allows these characteristics to be applied on demand. Every **aspect of a server’s configuration, from firmware revisions and BIOS settings to network profiles**, can be assigned through Cisco UCS Service Profiles. Cisco service profile templates establish policy-

based configuration for server, network, and storage resources and can be used to logically preconfigure these resources even before they are deployed in the cloud infrastructure.

3. **Simplicity at Scale**—With IT departments challenged to deliver more applications and services in shorter time frames, the architectural silos that result from an ad hoc approach to capacity scaling with traditional systems poses a barrier to successful cloud infrastructure deployment. Start with the computing and storage infrastructure needed today and then scale easily by adding components. Because servers and storage systems integrate into the unified system, they do not require additional supporting infrastructure or expert knowledge. The system simply, quickly, and cost-effectively presents more computing power and storage capacity to cloud infrastructure and applications.
4. **Virtual Infrastructure Density**—Cisco UCS enables cloud infrastructure to meet ever-increasing guest **OS memory demands on fewer physical servers. The system's high-density design** increases consolidation ratios for servers, saving the capital, operating, physical space, and licensing costs that would be needed to run virtualization software on larger servers. With Cisco UCS B200 M4 latest Intel Xeon E5-2600 v3 Series processor up to 1536 GB of RAM (using 64 GB DIMMs), OpenStack deployments can host more applications using less-expensive servers without sacrificing performance.
5. **Simplified Networking**—In OpenStack environments, underlying infrastructure can become sprawling complex of networked systems. Unlike traditional server architecture, Cisco UCS provides greater network density with less cabling **and complexity. Cisco's unified fabric integrates Cisco UCS servers** with a single high-bandwidth, low-latency network that supports all system I/O. This approach simplifies the architecture and reduces the number of I/O interfaces, cables, and access-layer switch ports compared to the requirements for traditional cloud infrastructure deployments. This unification **can reduce network complexity by up to a factor of three, and the system's wire-once network infrastructure** increases agility and accelerates deployment with zero-touch configuration.
6. **Installation Confidence**—Organizations that choose OpenStack for their cloud can take advantage of the Red Hat Enterprise Linux OpenStack Platform Installer. This software performs the work needed to install a validated OpenStack deployment. Unlike other solutions, this approach provides a highly available, highly scalable architecture for OpenStack services.
7. **Easy Management**—Cloud infrastructure can be extensive, so it must be easy and cost effective to manage. Cisco UCS Manager provides embedded management of all software and hardware components in Cisco UCS. Cisco UCS Manager resides as embedded software on the Cisco UCS fabric interconnects, fabric extenders, servers, and adapters. No external management server is required, simplifying administration and reducing capital expenses for the management environment.

Cisco Nexus 9000 Series Switch

The Cisco Nexus 9000 Series delivers proven high performance and density, low latency, and exceptional power efficiency in a broad range of compact form factors. Operating in Cisco NX-OS Software mode or in Application Centric Infrastructure (ACI) mode, these switches are ideal for traditional or fully automated datacenter deployments.

Figure 6 Cisco Nexus 9000 Series



The Cisco Nexus 9000 Series Switches offer both modular and fixed 10/40/100 Gigabit Ethernet switch configurations with scalability up to 30 Gbps of non-blocking performance with less than five-microsecond latency, 1152 10 Gbps or 288 40 Gbps non-blocking Layer 2 and Layer 3 Ethernet ports and wire speed VXLAN gateway, bridging, and routing support. For more information, see:

<http://www.cisco.com/c/en/us/products/switches/nexus-9000-series-switches/index.html>

Cisco Nexus 1000v for KVM - OpenStack

Cisco Nexus 1000V OpenStack solution is an enterprise-grade virtual networking solution, which offers Security, Policy control and Visibility all with Layer2/Layer 3 switching at the hypervisor, layer. Cisco Nexus 1000V provides state-full firewall functionality within your infrastructure to isolate tenants and enables isolation of virtual machines with policy-based VM attributes. **Cisco Nexus 1000V's robust policy framework** enables centralized enterprise-compliant policy management, pre-provisioning of policies on a network-wide basis and simplifies policy additions and modifications across the virtual infrastructure. 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. It also enables you to use your existing monitoring tools to provide rich analytics and auditing capabilities across your physical and virtual infrastructure.

Layer2/Layer3 Switching – Cisco Nexus 1000V offers the capability to route East-West traffic within the tenant without having to go to an external router. This capability reduces sub-optimal traffic patterns within the network and increases the network performance.

East-West Security – Cisco Nexus 1000V comes with Cisco Virtual Security Gateway (VSG) which provides layer 2 zone based firewall capability. Using VSG, Nexus 1000V can secure east west machine to machine traffic by providing stateful firewall functionality. VSG also enables the users to define security attributes based on VM attributes along with network attributes.

Policy Frame Work – Cisco Nexus 1000V provides an extremely power policy frame work to define polices per tenant and make these policies available to the end user through Horizon Dashboard or through Neutron API. This policy framework consists of the popular port profiles and the network policy (For example, VLAN, VxLAN). All these polices can be centrally managed which makes it easier to roll out new business polices or modify existing business polices instantaneously

Application Visibility – Cisco Nexus 1000V brings in a slew of industry proven monitoring features that exist in the physical Nexus infrastructure to virtual networking. To name few of them, Cisco Nexus 1000V provides remote-monitoring capabilities by using SPAN/ERSPAN, provides visibility into VM motion by using vTacker and provides consolidated interface status, traffic statistics using Virtual Supervisor Module (VSM)

All the monitoring, management and functionality features offered on the Cisco 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. Also, customers can experience a seamless functionality in the virtual network homogenous to the physical network.

Cisco Nexus 1000V for OpenStack Solution Offers

Table 1 Use Cases

Use Case	Description
----------	-------------

Use Case	Description
Micro-Segmentation	<ul style="list-style-type: none"> Stateful firewall functionality for East - West traffic (Layer 2 Zone based firewall) VM isolation in a common layer 2 segment (tenant) with no additional security group's
VM Visibility	<ul style="list-style-type: none"> Monitoring of live application traffic and collect user statistics Insight into live and past VM migrations
Policy Control	<ul style="list-style-type: none"> Centralized location for policy management Flexible and powerful frame work that enables organizations to pre provision network wide policies Policies available through Horizon and Neutron API

Cisco Nexus 1000V Components

Cisco Nexus 1000v brings the same robust architecture associated with traditional Cisco physical modular switches and with other virtualization environments (for example, VMware vSphere and Microsoft Hyper-V) to OpenStack deployments.

The Cisco Nexus1000v has the following main components:

- The Cisco Nexus® 1000V Virtual Ethernet Module (VEM) is a software component that is deployed on each Kernel-based Virtual Machine (KVM) host. Each virtual machine on the host is connected to the VEM through virtual Ethernet (vEth) ports.
- The Cisco Nexus 1000V Virtual Supervisor Module (VSM) is the management component that controls multiple VEMs and helps in the definition of virtual machine-focused network policies. It is deployed either as a virtual appliance on any KVM host or on the CiscoCloud Services Platform appliance.
- The Cisco Virtual Extensible LAN (VXLAN) Gateway is a gateway appliance to facilitate communication between a virtual machine located on a VXLAN with other entities (bare-metal servers, physical firewalls etc.) that are connected to traditional VLANs. It can be deployed as a virtual appliance on any KVM host.
- The OpenStack Neutron plug-in is used for communication between the VSM and OpenStack Neutron service and is deployed as part of the OpenStack Neutron service.
- The OpenStack Horizon integration for policy profile.

Each of these components are tightly integrated with the OpenStack environment:

- The VEM is a hypervisor-resident component and is tightly integrated with the KVM architecture.
- The VSM is integrated with OpenStack using the OpenStack Neutron plug-in.
- The OpenStack Neutron API has been extended to include two additional user-defined resources:

- Network profiles are logical groupings of network segments.
- Policy profiles group port policy information, including security, monitoring, and quality-of-service (QoS) policies.

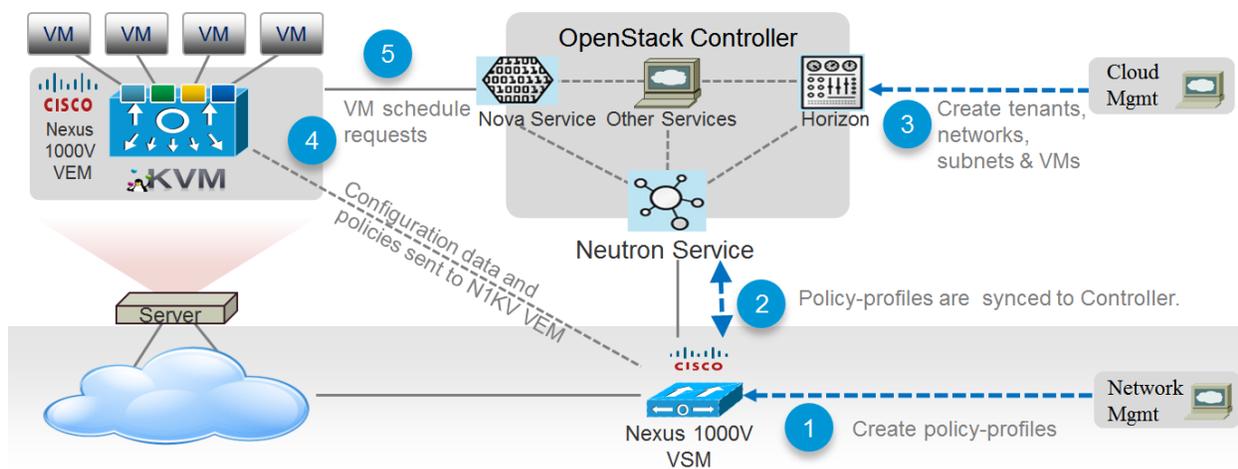


In Cisco Nexus 1000v for KVM Release 5.2(1) SK3 (2.2b), network profiles are created automatically for each network type. Network profile creation by administrators is not supported.

ML2 Mechanism Driver for Cisco Nexus 1000v

In Red Hat Enterprise Linux OpenStack Platform 6.0, The Cisco Nexus 1000v plugin added support for accepting REST API responses in JSON format from Virtual Supervisor Module (VSM) as well as control for enabling Policy Profile visibility across tenants. Figure 7 shows the operational workflow in OpenStack environment.

Figure 7 Cisco Nexus 1000v ML2 Driver Workflow



NetApp FAS8000

This FlexPod datacenter solution includes the NetApp fabric-attached storage (FAS) 8040 series unified scale-out storage system for both the OpenStack Block Storage Service (Cinder) and the OpenStack Image Service (Glance). Powered by NetApp Clustered Data ONTAP, the FAS8000 series unifies the SAN and NAS storage infrastructure. Systems architects can choose from a range of models representing a spectrum of cost-versus-performance points. Every model, however, provides the following core benefits:

- **HA and fault tolerance.** Storage access and security are achieved through clustering, high availability (HA) pairing of controllers, hot-swappable components, NetApp RAID DP[®] disk protection (allowing two independent disk failures without data loss), network interface redundancy, support for data mirroring with NetApp SnapMirror[®] software, application backup integration with the NetApp SnapManager[®] storage management software, and customizable data protection with the NetApp Snap Creator[®] framework and SnapProtect[®] products.
- **Storage efficiency.** Users can store more data with less physical media. This is achieved with thin provisioning (unused space is shared among volumes), NetApp Snapshot[®] copies (zero-storage, read-only clones of data over time), NetApp FlexClone[®] volumes and LUNs (read/write copies of data in which only changes are stored), deduplication (dynamic detection and removal of redundant data), and data compression.

- Unified storage architecture. Every model runs the same software (clustered Data ONTAP); supports all popular storage protocols (CIFS, NFS, iSCSI, FCP, and FCoE); and uses SATA, SAS, or SSD storage (or a mix) on the back end. This allows freedom of choice in upgrades and expansions, without the need for re-architecting the solution or retraining operations personnel.
- Advanced clustering. Storage controllers are grouped into clusters for both availability and performance pooling. Workloads can be moved between controllers, permitting dynamic load balancing and zero-downtime maintenance and upgrades. Physical media and storage controllers can be added as needed to support growing demand without downtime.

NetApp Storage Controllers

A storage system running Data ONTAP (also known as a storage controller) is the hardware device that receives and sends data from the host. Controller nodes are deployed in HA pairs, with these HA pairs participating in a single storage domain or cluster. This unit detects and gathers information about its own hardware configuration, the storage system components, the operational status, hardware failures, and other error conditions. A storage controller is redundantly connected to storage through disk shelves, which are the containers or device carriers that hold disks and associated hardware such as power supplies, connectivity interfaces, and cabling.

The NetApp FAS8000 features a multicore Intel chipset and leverages high-performance memory modules, NVRAM to accelerate and optimize writes, and an I/O-tuned PCIe gen3 architecture that maximizes application throughput. The FAS8000 series come with integrated unified target adapter (UTA2) ports that support 16 GB Fibre Channel, 10GbE, or FCoE. Figure 8 shows a front and rear view of the FAS8040/8060 controllers.

Figure 8 NetApp FAS8040/8060 (6U) - Front and Rear View



If storage requirements change over time, NetApp storage offers the flexibility to change quickly as needed without expensive and disruptive forklift upgrades. This applies to different types of changes:

- Physical changes, such as expanding a controller to accept more disk shelves and subsequently more hard disk drives (HDDs) without an outage
- Logical or configuration changes, such as expanding a RAID group to incorporate these new drives without requiring any outage
- Access protocol changes, such as modification of a virtual representation of a hard drive to a host by changing a logical unit number (LUN) from FC access to iSCSI access, with no data movement required, but only a simple dismount of the FC LUN and a mount of the same LUN, using iSCSI

In addition, a single copy of data can be shared between Linux and Windows systems while allowing each environment to access the data through native protocols and applications. In a system that was originally purchased with all SATA disks for backup applications, high-performance solid-state disks can be added to the same storage system to support Tier-1 applications, such as Oracle®, Microsoft Exchange, or Microsoft SQL Server.

For more NetApp FAS8000 information, see: <http://www.netapp.com/us/products/storage-systems/fas8000/>

NetApp Clustered Data ONTAP 8.3 Fundamentals

NetApp provides enterprise-ready, unified scale out storage with clustered Data ONTAP 8.3, the operating system physically running on the storage controllers in the NetApp FAS storage appliance. Developed from a solid foundation of proven Data ONTAP technology and innovation, clustered Data ONTAP is the basis for large virtualized shared-storage infrastructures that are architected for non-disruptive operations over the system lifetime.



Clustered Data ONTAP 8.3 is the first Data ONTAP release to support clustered operation only. The previous version of Data ONTAP, 7-Mode, is not available as a mode of operation in version 8.3.

Data ONTAP scale-out is one way to respond to growth in a storage environment. All storage controllers have physical limits to their expandability; the number of CPUs, memory slots, and space for disk shelves dictate maximum capacity and controller performance. If more storage or performance capacity is needed, it might be possible to add CPUs and memory or install additional disk shelves, but ultimately the controller becomes completely populated, with no further expansion possible. At this stage, the only option is to acquire another controller. One way to do this is to scale up; that is, to add additional controllers in such a way that each is an independent management entity that does not provide any shared storage resources. If the original controller is completely replaced by a newer, larger controller, data migration is required to transfer the data from the old controller to the new one. This is time consuming and potentially disruptive and most likely requires configuration changes on all of the attached host systems.

If the newer controller can coexist with the original controller, then the two storage controllers must be individually managed, and there are no native tools to balance or reassign workloads across them. The situation becomes worse as the number of controllers increases. If the scale-up approach is used, the operational burden increases consistently as the environment grows, and the end result is a very unbalanced and difficult-to-manage environment. Technology refresh cycles require substantial planning in advance, lengthy outages, and configuration changes, which introduce risk into the system.

In contrast, when using a scale-out approach, additional controllers are added seamlessly to the resource pool residing on a shared storage infrastructure as the storage environment grows. Host and client connections as well as volumes can move seamlessly and non-disruptively anywhere in the resource pool, so that existing workloads can be easily balanced over the available resources, and new workloads can be easily deployed. Technology refreshes (replacing disk shelves, adding or completely replacing storage controllers) are accomplished while the environment remains online and continues serving data.

Although scale-out products have been available for some time, these were typically subject to one or more of the following shortcomings:

- Limited protocol support. NAS only.
- Limited hardware support. Supported only a particular type of storage controller or a very limited set.

- Little or no storage efficiency. Thin provisioning, de-duplication, compression.
- Little or no data replication capability.

Therefore, while these products are well positioned for certain specialized workloads, they are less flexible, less capable, and not robust enough for broad deployment throughout the enterprise.

Data ONTAP is the first product to offer a complete scale-out solution, and it offers an adaptable, always-available storage infrastructure for today's highly virtualized environment.

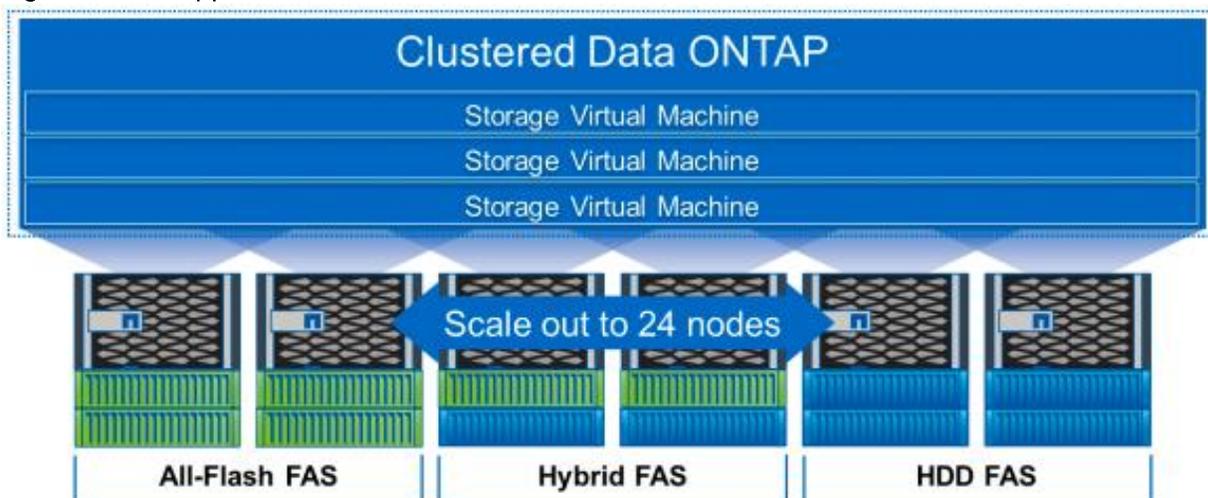
Scale Out

Datacenters require agility. In a datacenter, each storage controller has CPU, memory, and disk shelf limits. Scale-out means that as the storage environment grows, additional controllers can be added seamlessly to the resource pool residing on a shared storage infrastructure. Host and client connections as well as volumes can be moved seamlessly and non-disruptively anywhere within the resource pool. The benefits of scale-out include:

- Non-disruptive operations
- The ability to add tenants, instances, volumes, networks, and so on without downtime in OpenStack
- Operational simplicity and flexibility

As Figure 9 shows, NetApp Clustered Data ONTAP offers a way to solve the scalability requirements in a storage environment. A NetApp Clustered Data ONTAP system can scale up to 24 nodes, depending on platform and protocol, and can contain different disk types and controller models in the same storage cluster with up to 101PB of capacity.

Figure 9 NetApp Clustered Data ONTAP



Non-disruptive Operations

The move to shared infrastructure has made it nearly impossible to schedule downtime for routine maintenance. NetApp clustered Data ONTAP is designed to eliminate the need for planned downtime for maintenance operations and lifecycle operations as well as the unplanned downtime caused by hardware and software failures. NetApp storage solutions provide redundancy and fault tolerance through clustered storage controllers and redundant, hot-swappable components, such as cooling fans, power supplies, disk

drives, and shelves. This highly available and flexible architecture enables customers to manage all data under one common infrastructure while meeting mission-critical uptime requirements.

Three standard tools that eliminate the possible downtime:

- NetApp DataMotion™ data migration software for volumes (vol move). Allows you to move data volumes from one aggregate to another on the same or a different cluster node.
- Logical interface (LIF) migration. Allows you to virtualize the physical Ethernet interfaces in clustered Data ONTAP. LIF migration allows the administrator to move these virtualized LIFs from one network port to another on the same or a different cluster node.
- Aggregate relocate (ARL). Allows you to transfer complete aggregates from one controller in an HA pair to the other without data movement.

Used individually and in combination, these tools allow you to non-disruptively perform a full range of operations, from moving a volume from a faster to a slower disk all the way up to a complete controller and storage technology refresh.

As storage nodes are added to the system, all physical resources (CPUs, cache memory, network I/O bandwidth, and disk I/O bandwidth) can be easily kept in balance. NetApp Data ONTAP enables users to:

- Move data between storage controllers and tiers of storage without disrupting users and applications
- Dynamically assign, promote, and retire storage, while providing continuous access to data as administrators upgrade or replace storage
- Increase capacity while balancing workloads and reduce or eliminate storage I/O hot spots without the need to remount shares, modify client settings, or stop running applications.

These features allow a truly non-disruptive architecture in which any component of the storage system can be upgraded, resized, or re-architected without disruption to the private cloud infrastructure.

Availability

Shared storage infrastructure provides services to many different tenants in an OpenStack deployment. In such environments, downtime produces disastrous effects. The NetApp FAS eliminates sources of downtime and protects critical data against disaster through two key features:

- High Availability (HA). A NetApp HA pair provides seamless failover to its partner in the event of any hardware failure. Each of the two identical storage controllers in the HA pair configuration serves data independently during normal operation. During an individual storage controller failure, the data service process is transferred from the failed storage controller to the surviving partner.
- RAID-DP®. During any OpenStack deployment, data protection is critical because any RAID failure might disconnect and/or shutoff hundreds or potentially thousands of end users from their virtual machines, resulting in lost productivity. RAID-DP provides performance comparable to that of RAID 10 and yet it requires fewer disks to achieve equivalent protection. RAID-DP provides protection against double-disk failure, in contrast to RAID 5, which can protect against only one disk failure per RAID group, in effect providing RAID 10 performance and protection at a RAID 5 price point.

For more information, see: [Clustered Data ONTAP 8.3 High-Availability Configuration Guide](#)

NetApp Advanced Data Management Capabilities

This section describes the storage efficiencies, advanced storage features, and multiprotocol support capabilities of the NetApp FAS8000 storage controller.

Storage Efficiencies

NetApp FAS includes built-in thin provisioning, data deduplication, compression, and zero-cost cloning with FlexClone that offers multilevel storage efficiency across OpenStack instances, installed applications, and user data. This comprehensive storage efficiency enables a significant reduction in storage footprint, with a capacity reduction of up to 10:1, or 90% (based on existing customer deployments and NetApp solutions lab validation). Four features make this storage efficiency possible:

- **Thin provisioning.** Allows multiple applications to share a single pool of on-demand storage, eliminating the need to provision more storage for one application if another application still has plenty of allocated but unused storage.
- **De-duplication.** Saves space on primary storage by removing redundant copies of blocks in a volume that hosts hundreds of instances. This process is transparent to the application and the user, and it can be enabled and disabled on the fly or scheduled to run at off-peak hours.
- **Compression.** Compresses data blocks. Compression can be run whether or not deduplication is enabled and can provide additional space savings whether it is run alone or together with deduplication.
- **FlexClone.** Offers hardware-assisted rapid creation of space-efficient, writable, point-in-time images of individual VM files, LUNs, or flexible volumes. The use of FlexClone technology in OpenStack deployments provides high levels of scalability and significant cost, space, and time savings. The NetApp Cinder driver provides the flexibility to rapidly provision and redeploys thousands of instances.

Advanced Storage Features

NetApp Data ONTAP provides a number of additional features, including:

- **NetApp Snapshot™ copy.** A manual or automatically scheduled point-in-time copy that writes only changed blocks, with no performance penalty. A Snapshot copy consumes minimal storage space because only changes to the active file system are written. Individual files and directories can easily be recovered from any Snapshot copy, and the entire volume can be restored back to any Snapshot state in seconds. A NetApp Snapshot incurs no performance overhead. Users can comfortably store up to 255 NetApp Snapshot copies per NetApp FlexVol® volume, all of which are accessible as read-only and online versions of the data.



NetApp Snapshots are taken at the FlexVol level, so they cannot be directly leveraged within an OpenStack user context. This is because a Cinder user requests that a Snapshot be taken of a particular Cinder volume, not the containing FlexVol volume. Because a Cinder volume is represented as either a file in the NFS or as a LUN (in the case of iSCSI or Fibre Channel), Cinder snapshots can be created by using FlexClone, which allows you to create many thousands of Cinder snapshots of a single Cinder volume.

NetApp Snapshots are however available to OpenStack administrators to do administrative backups, create and/or modify data protection policies, etc.

- LIF. A logical interface that is associated with a physical port, interface group, or VLAN interface. More than one LIF may be associated with a physical port at the same time. There are three types of LIFs: NFS LIFs, iSCSI LIFs, and Fibre Channel LIFs. LIFs are logical network entities that have the same characteristics as physical network devices but are not tied to physical objects. LIFs used for Ethernet traffic are assigned specific Ethernet-based details such as IP addresses and iSCSI qualified names and then are associated with a specific physical port capable of supporting Ethernet. LIFs used for FC-based traffic are assigned specific FC-based details such as worldwide port names (WWPNs) and then are associated with a specific physical port capable of supporting FC or FCoE. NAS LIFs can be non-disruptively migrated to any other physical network port throughout the entire cluster at any time, either manually or automatically (by using policies), whereas SAN LIFs rely on MPIO and ALUA to notify clients of any change in the network topology.
- Storage Virtual Machines (SVMs). An SVM is a secure virtual storage server that contains data volumes and one or more LIFs, through which it serves data to the clients. An SVM securely isolates the shared, virtualized data storage and network and appears as a single dedicated server to its clients. Each SVM has a separate administrator authentication domain and can be managed independently by an SVM administrator.

Unified Storage Architecture and Multiprotocol Support

NetApp also offers the NetApp Unified Storage Architecture as well. The term “unified” refers to a family of storage systems that simultaneously support SAN (through FCoE, Fibre Channel (FC), and iSCSI) and network-attached storage (NAS) (through CIFS and NFS) across many operating environments, including OpenStack, VMware®, Windows, Linux, and UNIX. This single architecture provides access to data by using industry-standard protocols, including NFS, CIFS, iSCSI, FCP, SCSI, and NDMP.

Connectivity options include standard Ethernet (10/100/1000Mb or 10GbE) and Fibre Channel (4, 8, or 16 Gb/sec). In addition, all systems can be configured with high-performance solid-state drives (SSDs) or serial-attached SCSI (SAS) disks for primary storage applications, low-cost SATA disks for secondary applications (backup, archive, and so on), or a mix of the different disk types. By supporting all common NAS and SAN protocols on a single platform, NetApp FAS enables:

- Direct access to storage for each client
- Network file sharing across different platforms without the need for protocol-emulation products such as SAMBA, NFS Maestro, or PC-NFS
- Simple and fast data storage and data access for all client systems
- Fewer storage systems
- Greater efficiency from each system deployed

NetApp Clustered Data ONTAP can support several protocols concurrently in the same storage system and data replication and storage efficiency features are supported across all protocols. The following are supported:

- NFS v3, v4, and v4.1, including pNFS
- iSCSI
- Fibre Channel
- FCoE

- SMB 1, 2, 2.1, and 3

Storage Virtual Machines

The secure logical storage partition through which data is accessed in clustered Data ONTAP is known as an SVM. A cluster serves data through at least one and possibly multiple SVMs. An SVM is a logical abstraction that represents a set of physical resources of the cluster. Data volumes and logical network LIFs are created and assigned to an SVM and can reside on any node in the cluster to which the SVM has been given access. An SVM can own resources on multiple nodes concurrently, and those resources can be moved non-disruptively from one node to another. For example, a flexible volume can be non-disruptively moved to a new node, and an aggregate, or a data LIF, can be transparently reassigned to a different physical network port. The SVM abstracts the cluster hardware and is not tied to specific physical hardware.

An SVM is capable of supporting multiple data protocols concurrently. Volumes within the SVM can be **combined together to form a single NAS namespace, which makes all of an SVM's data available to NFS and CIFS clients** through a single share or mount point. For example, a 24-node cluster licensed for UNIX and Windows File Services that has a single SVM configured with thousands of volumes can be accessed from a single network interface on one of the nodes. SVMs also support block-based protocols, and LUNs can be created and exported by using iSCSI, FC, or FCoE. Any or all of these data protocols can be configured for use within a given SVM.

An SVM is a secure entity. Therefore, it is aware of only the resources that have been assigned to it and has no knowledge of other SVMs and their respective resources. Each SVM operates as a separate and distinct entity with its own security domain. Tenants can manage the resources allocated to them through a delegated SVM administration account. An SVM is effectively isolated from other SVMs that share the same physical hardware, and as such, is uniquely positioned to align with OpenStack tenants for a truly comprehensive multi-tenant environment. Each SVM can connect to unique authentication zones, such as AD, LDAP, or NIS.

From a performance perspective, maximum IOPS and throughput levels can be set per SVM by using QoS policy groups, which allow the cluster administrator to quantify the performance capabilities allocated to each SVM.

Clustered Data ONTAP is highly scalable, and additional storage controllers and disks can easily be added to existing clusters to scale capacity and performance to meet rising demands. Because these are virtual storage servers within the cluster, SVMs are also highly scalable. As new nodes or aggregates are added to the cluster, the SVM can be non-disruptively configured to use them. New disk, cache, and network resources can be made available to the SVM to create new data volumes or to migrate existing workloads to these new resources to balance performance.

This scalability also enables the SVM to be highly resilient. SVMs are no longer tied to the lifecycle of a given storage controller. As new replacement hardware is introduced, SVM resources can be moved non-disruptively from the old controllers to the new controllers, and the old controllers can be retired from service while the SVM is still online and available to serve data.

SVMs have three main components:

- Logical interfaces. All SVM networking is done through LIFs created within the SVM. As logical constructs, LIFs are abstracted from the physical networking ports on which they reside.
- Flexible volumes. A flexible volume is the basic unit of storage for an SVM. An SVM has a root volume and can have one or more data volumes. Data volumes can be created in any aggregate that has been

delegated by the cluster administrator for use by the SVM. Depending on the data protocols used by the SVM, volumes can contain either LUNs for use with block protocols, files for use with NAS protocols, or both concurrently. For access using NAS protocols, the volume must be added to the SVM namespace through the creation of a client-visible directory called a junction.

- Namespaces. Each SVM has a distinct namespace through which all of the NAS data shared from that SVM can be accessed. This namespace can be thought of as a map to all of the junctioned volumes for the SVM, regardless of the node or the aggregate on which they physically reside. Volumes can be junctioned at the root of the namespace or beneath other volumes that are part of the namespace hierarchy. For more information about namespaces, see: [NetApp TR-4129: Namespaces in Clustered Data ONTAP](#).

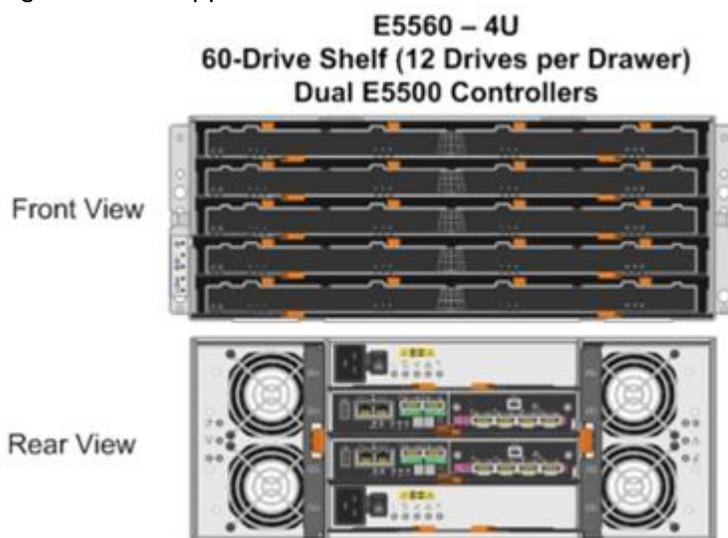
For more information on Data ONTAP, go to: [NetApp Data ONTAP 8.3 Operating System](#).

NetApp E5000 Series

This FlexPod Datacenter solution also makes use of the NetApp E-Series E5560 storage system, primarily for the OpenStack Object Storage service (Swift). An E5560 is comprised of dual E5500 controllers mated with the 4U 60 drive DE6600 chassis. The NetApp® E5500 storage system family is designed to meet the most demanding and data-intensive applications and provide continuous access to data. It is from the E-Series line, which offers zero-scheduled downtime systems, redundant hot-swappable components, automated path failover, and online administration capabilities.

The E5560 is shown in Figure 10.

Figure 10 NetApp E-Series E5560



NetApp E-Series Storage Controllers

The E5000 Series hardware delivers an enterprise level of availability with:

- Dual-active controllers, fully redundant I/O paths, and automated failover
- Battery-backed cache memory that is designed to flash upon power loss

- Extensive monitoring of diagnostic data that provides comprehensive fault isolation, simplifying analysis of unanticipated events for timely problem resolution
- Proactive repair that helps get the system back to optimal performance in minimum time

This storage system additionally provides the following high-level benefits:

- **Flexible Interface Options.** The E-Series supports a complete set of host or network interfaces designed for either direct server attachment or network environments. With multiple ports per interface, the rich connectivity provides ample options and bandwidth for high throughput. The interfaces include quad-lane SAS, iSCSI, FC, and InfiniBand to connect with and protect investments in storage networking.
- **High Availability and Reliability.** The E-Series simplifies management and maintains organizational productivity by keeping data accessible through redundant components, automated path failover, and online administration, including online SANtricity® OS and drive firmware updates. Advanced protection features and extensive diagnostic capabilities deliver high levels of data integrity, including Data Assurance (T10-PI) to protect against silent drive errors.
- **Maximum Storage Density and Modular Flexibility.** The E-Series offers multiple form factors and drive technology options to best meet your storage requirements. The ultra-dense 60-drive system shelf supports up to 360TB in just 4U of space. It is perfect for environments with large amounts of data and limited floor space. Its high-efficiency power supplies and intelligent design can lower power use up to 40% and cooling requirements by up to 39%.
- **Intuitive Management.** NetApp SANtricity Storage Manager software offers extensive configuration flexibility, optimal performance tuning, and complete control over data placement. With its dynamic capabilities, SANtricity software supports on-the-fly expansion, reconfigurations, and maintenance without interrupting storage system I/O.

For more information on the NetApp E5560, see: [NetApp E5500 Storage System](#).

NetApp SANtricity Operating System Fundamentals

With over 20 years of storage development behind it, and approaching nearly one million systems shipped, the E-Series platform is based on a field-proven architecture that uses the SANtricity storage management software on the controllers. This OS is designed to provide high reliability and greater than 99.999% availability, data integrity, and security. The SANtricity OS:

- Delivers best-in-class reliability with automated features, online configuration options, state-of-the-art RAID, proactive monitoring, and **NetApp AutoSupport™ capabilities**.
- Extends data protection through FC- and IP-based remote mirroring, NetApp SANtricity Dynamic Disk Pools (DDPs), enhanced Snapshot copies, data-at-rest encryption, data assurance to ensure data integrity, and advanced diagnostics.
- Includes plug-ins for application-aware deployments of Oracle®, VMware®, Microsoft®, and Splunk® applications.

For more information, see the [NetApp SANtricity Operating System](#) product page.

Dynamic Disk Pools

DDPs increase the level of data protection, provide more consistent transactional performance, and improve the versatility of E-Series systems. DDP dynamically distributes data, spare capacity, and parity information across a pool of drives. An intelligent algorithm (seven patents pending) determines which drives are used for data placement, and data is dynamically recreated and redistributed as needed to maintain protection and uniform distribution.

Consistent Performance during Rebuilds

DDP minimizes the performance drop that can occur during a disk rebuild, allowing rebuilds to complete up to eight times faster than with traditional RAID. Therefore, your storage spends more time in an optimal performance mode that maximizes application productivity. Shorter rebuild times also reduce the possibility of a second disk failure occurring during a disk rebuild and protects against unrecoverable media errors. Stripes with several drive failures receive priority for reconstruction.

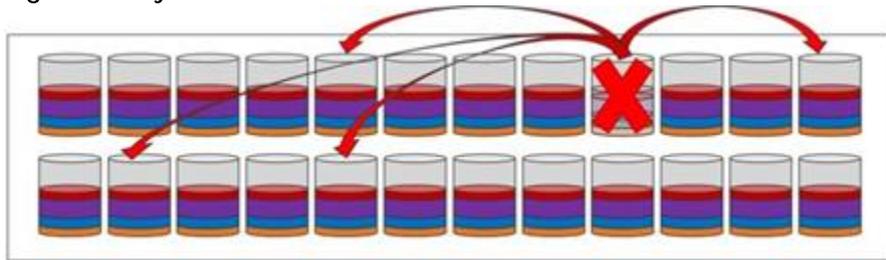
Overall, DDP provides a significant improvement in data protection; the larger the pool, the greater the protection. A minimum of 11 disks is required to create a disk pool.

How DDP Works

When a disk fails with traditional RAID, data is recreated from parity on a single hot spare drive, creating a bottleneck. All volumes using the RAID group suffer. DDP distributes data, parity information, and spare capacity across a pool of drives. Its intelligent algorithm based on CRUSH defines which drives are used for segment placement, ensuring full data protection. DDP dynamic rebuild technology uses every drive in the pool to rebuild a failed drive, enabling exceptional performance under failure. Flexible disk-pool sizing optimizes utilization of any configuration for maximum performance, protection, and efficiency.

When a disk fails in a Dynamic Disk Pool, reconstruction activity is spread across the pool and the rebuild is completed eight times faster.

Figure 11 Dynamic Disk Pool



NetApp Storage for OpenStack

Most options for OpenStack integrated storage solutions aspire to offer scalability, but often lack the features and performance needed for efficient and cost-effective cloud deployment at scale.

NetApp has developed OpenStack interfaces to provide FAS and E-Series value to enterprise customers and thus provides them with a choice in cloud infrastructure deployment, including open-source options that provide lower cost, faster innovation, unmatched scalability, and the promotion of standards. As OpenStack abstracts the underlying hardware from customer applications and workloads, NetApp enterprise storage features and functionality can be exposed through unique integration capabilities built for OpenStack. Features are passed through the interfaces such that standard OpenStack management tools (CLI, Horizon, etc.) can be used to access NetApp value proposition for simplicity and automation.

Once exposed through the abstraction of the OpenStack API set, NetApp technology features are now accessible, such as data deduplication, thin provisioning, cloning, Snapshots, DDPs, mirroring, and so on. Customers can be confident that the storage infrastructure underlying their OpenStack Infrastructure as a Service (IaaS) environment is highly available, flexible, and performant.

Because NetApp technology is integrated with

- OpenStack Block Storage Service (Cinder)
- OpenStack Object Storage Service (Swift)
- OpenStack Image Service (Glance)
- OpenStack Compute Service (Nova)
- OpenStack File Share Service (Manila)

Users can build on this proven and highly scalable storage platform not only with greenfield deployments as illustrated in this CVD, and also with brownfield deployments for customers who wish to optimize their existing NetApp storage infrastructure.

Cinder

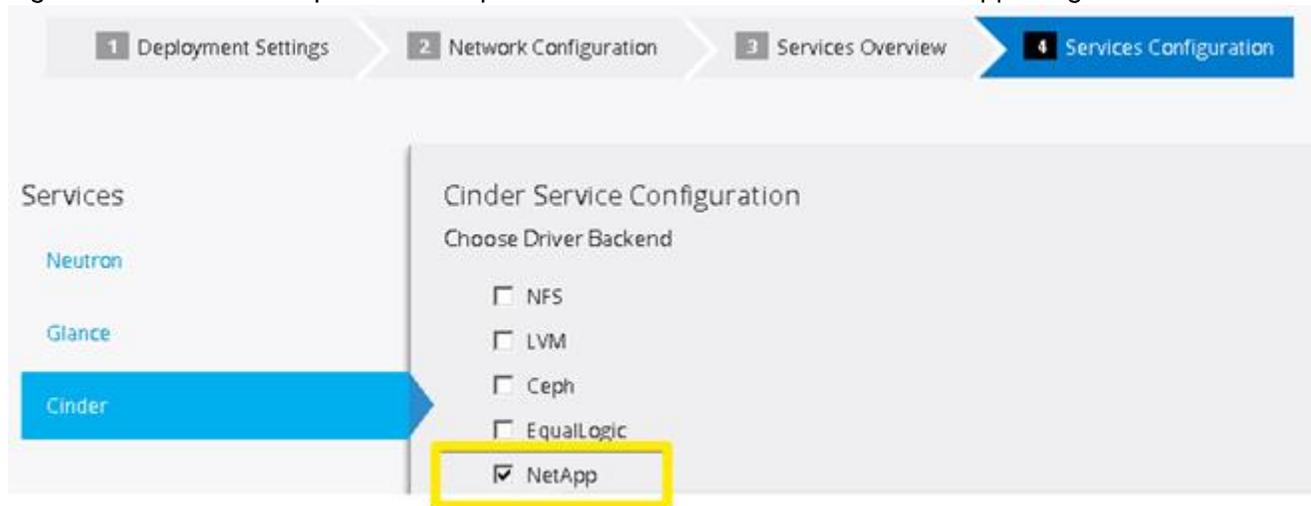
The OpenStack Block Storage service provides management of persistent block storage resources. In addition to acting as secondarily attached persistent storage, you can write images into a Cinder volume for Nova to utilize as a bootable, persistent root volume for an instance.

In this Cisco Validated Design, Cinder volumes are stored on the NetApp FAS8040 storage array and accessed using the pNFS (NFS version 4.1) protocol. The pNFS protocol has a number of advantages at scale in a large, heterogeneous hybrid cloud, including dynamic failover of network paths, high performance through parallelization, and an improved NFS client.

Red Hat Enterprise Linux OpenStack Platform Installer Integration

The Cinder service configurations (and pertinent configuration files on the resulting Controller hosts) are handled automatically as a part of an OpenStack Deployment from within the Red Hat Enterprise Linux OpenStack Platform Installer. Customers can select NetApp within the Services Configuration plane of a deployment. (0)

Figure 12 Red Hat Enterprise Linux OpenStack Platform Installer Cinder NetApp Integration



Selectable options include support for NetApp Clustered Data ONTAP, Data ONTAP 7-mode, and E-Series platforms. In this reference architecture, NetApp Clustered Data ONTAP is selected, and pertinent details are filled in that are representative for a NetApp FAS8040 storage subsystem.

NetApp Unified Driver for Clustered Data ONTAP with NFS

A Cinder driver is a particular implementation of a Cinder backend that maps the abstract APIs and primitives of Cinder to appropriate constructs within the particular storage solution underpinning the Cinder backend.

The NetApp Unified Driver for clustered Data ONTAP with NFS is a driver interface from OpenStack block storage to a Data ONTAP cluster system. This software provisions and manages OpenStack volumes on NFS exports provided by the Data ONTAP cluster system. The NetApp Unified Driver for the Data ONTAP cluster does not require any additional management software to achieve the desired functionality because it uses NetApp APIs to interact with the Data ONTAP cluster. It also does not require additional configuration in addition to selecting NetApp during an OpenStack Deployment in the Red Hat Enterprise Linux OpenStack Platform installer.

In this Cisco Validated Design, we take advantage of the NetApp Unified Driver using the NetApp driver backend through the Red Hat Enterprise Linux OpenStack Platform Installer. All of the resulting UCS blades that are provisioned with Red Hat Enterprise Linux 7.1 mount the appropriately designated NetApp FlexVols on the FAS8040 at the highest protocol level possible for the instance volumes (NFS version 4.1 or Parallelized NFS [pNFS]).



A FlexVol volume is a logical container for one or more Cinder volumes.

NetApp's contribution strategy involves adding all new capabilities directly into the upstream OpenStack repositories, so that all of the features are available in Red Hat Enterprise Linux OpenStack Platform 6.0 out of the box. More information regarding the NetApp Unified Driver (including other protocols available) can be found in the following location: [NetApp Data ONTAP Drivers for OpenStack Block Storage \(Cinder\)](#).

More information as to why NFS was chosen over iSCSI in this CVD can be found in the following location: [Deployment Choice: NFS versus iSCSI](#).

More information regarding Clustered Data ONTAP with NFSv4 features can be found in the following location: [TR-4067: Clustered Data ONTAP NFS Implementation Guide](#).

Storage Service Catalog

The Storage Service Catalog (SSC) enables efficient, repeated, and consistent use and management of storage resources by the definition of policy-based services and the mapping of those services to the backend storage technology. It is meant to abstract away the actual technical implementations of the features at a storage backend into a set of simplified configuration options.

These storage features are organized or combined into groups based on a customer's particular scenario or use case. Based on this catalog of storage features, intelligent provisioning decisions are made by infrastructure or software enabling the SSC. In OpenStack, this is achieved by both the Cinder filter scheduler and the NetApp driver by making use of volume type extra-specs support together with the filter scheduler. There are some prominent features that are exposed in the NetApp driver, including mirroring, deduplication, compression, and thin provisioning. Workloads can be tied to Cinder volume types in an OpenStack context, which then have inherent NetApp technology enabled on the storage system. Examples needing functionality include:

- Transactional databases that require high IOPS with SSD disks, and data protection
- Test and development workloads that would benefit from thin provisioning and compression
- Disaster recovery processes that need a SnapMirror relationship to another NetApp storage system

When you use the NetApp Unified Driver with a clustered Data ONTAP storage system, you can leverage extra specs with Cinder volume types to ensure that Cinder volumes are created on storage backends that have certain properties configured (for example, QoS, mirroring, and compression). Extra specifications are associated with Cinder volume types, so that when users request volumes of a particular volume type, they are created on storage backends that meet the list of requirements (for example, available space, extra specs, and so on).

In this Cisco Validated Design, we create five different OpenStack-specific NetApp flexible volumes on the FAS8040 with different features enabled that can be selected intelligently based on the Cinder scheduler and the NetApp Cinder driver.

More information regarding available extra-specs are available in the following location: [Using Cinder Volume Types to Create a Storage Service Catalog](#).

Swift

OpenStack Object Storage provides a fully distributed, scale-out, API-accessible storage platform that can be integrated directly into applications or used for backup, archiving and data retention. Object storage does not present a traditional file system, but rather a distributed storage system for static data such as: virtual machine images, photo storage, email storage, backups, and archives.

Customers can start with the ultra-dense 60 drive enclosure as demonstrated in this Cisco Validated Design, and then can scale horizontally with multiple controller pairs as the size of the object storage needs grow. Swift data is hosted on the NetApp E5560 storage array using the iSCSI protocol. Three of the UCS servers are used as Swift nodes and handle account, container, and object services. In addition, these three nodes also serve as Proxy servers for the Swift service.

E-Series Resiliency

E-Series storage can effectively serve as the storage medium for Swift. The data reconstruction capabilities associated with DDP eliminates the need for data replication within zones in Swift. DDP reconstruction provides RAID-6 “like” data protection against multiple simultaneous drive failures within the storage

subsystem. Data that resides on multiple failed drives is given top priority during reconstruction. This data has the highest potential for being lost if a third drive failure occurs and is thus reconstructed first on the remaining optimal drives in the storage subsystem. After this critical data is reconstructed, all other data on the failed drives is reconstructed. This prioritized data reconstruction dramatically reduces the possibility of data loss due to drive failure.

As disk sizes increase, the rebuild time after failure also increases. The time taken by the traditional RAID systems to rebuild after a failure to an idle spare becomes more longer. This is because the idle spare in the traditional RAID receives all of the write traffic during a rebuild, slowing down the system and data access during this process. One of the main goals of DDP is to spread the workload around if a disk fails and its data must be rebuilt. This provides consistent performance, keeps you in the green zone, and maintains a non-disruptive level of performance. DDP has shown the ability that provide up to eight times faster **reconstruction of a failed disk's data throughout the pool when compared to an equivalent standard RAID-configuration disk rebuild.**

The dynamic process of redistributing the data occurs in the background in a non-disruptive, minimal-impact manner so that the I/O continues to flow.

Scalability on NetApp E-Series

Swift uses zoning to isolate the cluster into separate partitions and isolate the cluster from failures. Swift data is replicated across the cluster in as unique-as-possible zones. Typically, zones are established using physical attributes of the cluster, including geographical locations, separate networks, equipment racks, storage subsystems, or even single drives. Zoning allows the cluster to function and tolerate equipment failures without data loss or loss of connectivity to the remaining cluster.

By default, Swift replicates data 3 times across the cluster. Swift replicates data across zones in a unique way that promotes high availability and high durability. Swift chooses a server in an unused zone before it chooses an unused server in a zone that already has a replica of the data. E-Series data reconstruction makes sure that clients always have access to data regardless of drive or other component failures within the storage subsystem. When E-Series storage is used, Swift data replication counts that are specified when rings are built can be reduced from 3 to 2. This dramatically reduces the replication traffic normally sent on the standard IPv4 datacenter networks.

Reduction in Physical Resources using Swift on NetApp E-Series

In addition to the previously discussed issues, using Swift on NetApp E-Series enables:

- Reduced Swift node hardware requirements. Internal drive requirements for storage nodes are reduced, and only operating system storage is required. Disk space for Swift object data, and optionally the operating system itself, is supplied by the E-Series storage array.
- Reduced rack space, power, cooling and footprint requirements. Because a single storage subsystem provides storage space for multiple Swift nodes, smaller and possibly lower power 1U nodes can be used in the cluster.

Red Hat Enterprise Linux OpenStack Platform 6.0 Installer Integration

Swift is not installed through the Red Hat Enterprise Linux OpenStack Platform Installer. Instructions on installing Swift after an OpenStack deployment are provided in the deployment guide.

For more information regarding Swift on NetApp is available in the following location: [OpenStack Object Storage Service \(Swift\)](#).

Glance

The OpenStack Image Service provides discovery, registration and delivery services for virtual machine, disk, and server images. Glance provides a RESTful API that allows the querying of VM image metadata as well as the retrieval of the actual image. A stored image can be used as a template to start up new servers quickly and consistently as opposed to provisioning multiple servers, installing a server operating system, and individually configuring additional services. Such an image can also be used to store and catalog an unlimited number of backups.

In this Cisco Validated Design, Glance is utilized using NFS version 4.0 back to the NetApp FAS8040 storage array. Glance can store disk and server images in a variety of backends (called stores), which are featured as NFS in this CVD.

Red Hat Enterprise Linux OpenStack Platform Installer Integration

Glance configuration like Cinder is handled through an intuitive menu-based interface from within the Red Hat Enterprise Linux OpenStack Platform Installer. Before an OpenStack Deployment is launched, Glance is configured in the Red Hat Enterprise Linux OpenStack Platform Installer to utilize an already configured NetApp FlexVol through NFS with deduplication enabled.



Because there is a high probability of duplicate blocks in a repository of virtual machine images, NetApp highly recommends enabling deduplication on the FlexVol volume(s) where the images are stored.

Image Formats: QCOW and Raw

Glance supports a variety of image formats, but raw and QCOW2 are the most common. QCOW2 does provide some advantages over the raw format (for example, the support of copy-on-write, snapshots, and dynamic expansion). However, when images are copied into Cinder volumes, they are automatically converted into the raw format once stored on a NetApp backend. Therefore:

- The QCOW2 image format is recommended for ephemeral disks due to its inherent benefits when taking instance snapshots.
- The raw image format can be advantageous when Cinder volumes are used as persistent boot disks, as a conversion from an alternate format to raw that would be performed by Cinder can be avoided.

Both the raw and QCOW2 formats respond well to NetApp deduplication technology, which is often utilized with Glance deployments.

QCOW2 is not Live Migration safe on NFS when the cache=writeback setting is enabled, which is commonly used for performance improvement of QCOW2. If space savings are the desired outcome for the Image Store, raw format files are actually created as sparse files on the NetApp storage system. Deduplication within NetApp FlexVol volumes happens globally rather than only within a particular file, resulting in much better aggregate space efficiency than QCOW2 can provide. Deduplication processing can be finely controlled to run at specific times of day (off peak).

Copy Offload Tool

The NetApp Copy Offload tool was added in the Icehouse release to enable the efficient copying of Glance images to a destination Cinder volume. When Cinder and Glance are configured to use the NetApp NFS Copy Offload tool, a controller-side copy is attempted before reverting to downloading the image from Glance through a normal network copy. This improves image provisioning times while reducing the

consumption of bandwidth and CPU cycles on the host(s) running Glance and Cinder. This is due to the copy operation being performed completely within the storage cluster.



If Cinder and Glance share the same NetApp FlexVol, the Copy Offload tool is not necessary. Rather, a direct API call to the NetApp storage system is utilized through the NetApp Unified driver that facilitates a controller-side copy relative to a network copy.

For more information on this functionality, including a helpful process flowchart, see: [NetApp Copy Offload tool](#).

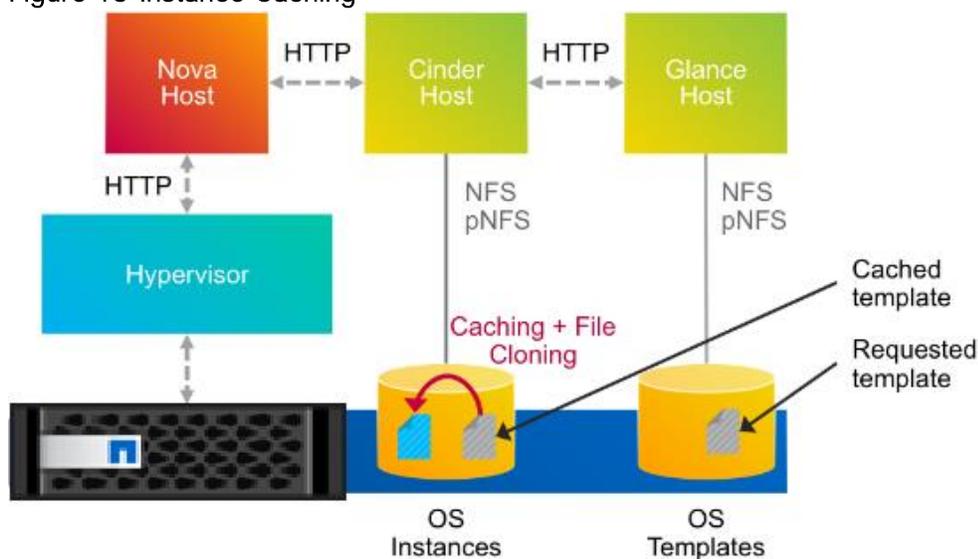
Rapid Cloning

NetApp provides two capabilities that enhance instance booting by using persistent disk images in the shortest possible time and in the most storage-capacity-efficient manner possible: the NetApp Copy Offload tool and instance caching.

This Enhanced Persistent Instance Creation feature (sometimes referred to as rapid cloning) uses NetApp FlexClone technology and the NetApp Copy Offload tool. The Enhanced Persistent Instance Creation feature can significantly decrease the time needed for the Nova service to fulfill image provisioning and boot requests. It also allows for much larger images with no noticeable degradation of boot time.

One feature that facilitates rapid cloning in an NFS/pNFS setup within the NetApp Unified Driver is instance caching. Whenever a Cinder volume is created out of a Glance template, it is cached locally on the NetApp FlexVol that hosts the Cinder volume instance. Later, when you want to create the same OS instance again, Cinder creates a space-efficient file clone. This clone does not take up any more space because it shares the same blocks as the cached image. Only deltas take up new blocks on the disk. Figure 13 illustrates this concept.

Figure 13 Instance Caching



This not only makes the instance/Cinder volume create operation faster, but it also reduces the CPU load on the Cinder/Glance hosts and reduces the network traffic almost completely. The cache also provides a time-to-live option, which invalidates old cache entries automatically after a specified period of time.

For more information regarding Glance on NetApp, see: [OpenStack Image Service \(Glance\)](#).

Nova

The OpenStack Compute Service (Nova) is a cloud computing fabric controller that is the primary part of an IaaS system. You can use the OpenStack Compute service to host and manage cloud instances (virtual machines).

Root and Ephemeral Disks

Each instance requires at least one root disk containing the bootloader and core operating system files, and each instance might also have optional ephemeral disks that use the definition of the flavor selected at instance creation time. The content for the root disk comes either from an image stored within the Glance repository, which is copied to storage attached to the destination hypervisor, or from a persistent block storage volume through Cinder.

By selecting the Boot from Image (Creates a New Volume) option in Nova, you can leverage the enhanced instance creation capabilities described previously. Normally volumes created as a result of this option are persistent beyond the life of the instance. However, you can select the Delete on Terminate option in combination with the Boot from Image (Creates a New Volume) option to create an ephemeral volume while still leveraging the Rapid Cloning capabilities described in the section [above](#). This can provide a significantly faster provisioning and boot sequence relative to the normal way that ephemeral disks are provisioned. In the normal way, a copy of the disk image is made from Glance to local storage on the hypervisor node where the instance resides. A Glance instance image of 20GB can, for example, be cloned in 300ms using NetApp FlexClone technology.

For more information on using the Nova service in conjunction with NetApp, go to: [OpenStack Compute Service \(Nova\)](#).

Manila

NetApp has developed a new OpenStack module called Manila to provide a shared file-system service. Much of the total storage shipped worldwide is based on shared file systems, and, with help from the OpenStack community, NetApp is delivering these capabilities to the OpenStack environment. Before Manila, OpenStack only had the Cinder module for block files. NetApp designed, prototyped, and built the Manila module, which is the equivalent of Cinder for shared or distributed file systems. Manila has emerged as an official, independent project in the Grizzly release of OpenStack.



Manila is not included in the official Red Hat package repositories for Red Hat Enterprise Linux OpenStack Platform 6.0, and is thus not featured in this CVD.

RPM Packages exist for Manila on Red Hat Enterprise Linux 7 at the following location, but are unsupported: <https://repos.fedorapeople.org/repos/openstack/openstack-kilo/el7/>

Support for Manila will be in future releases of Red Hat Enterprise Linux OpenStack Platform.

NetApp Storage platforms integrated with OpenStack provide a unique combination of advanced storage efficiency, integrated data protection, and non-disruptive operations combined with the ability to scale while preserving performance.

For more information on NetApp Storage for OpenStack, see:

- [NetApp OpenStack Deployment and Operations Guide](#)
- [Highly Available OpenStack Deployments Built on NetApp Storage Systems](#).

Domain and Management Software

FlexPod with Red Hat Enterprise Linux OpenStack Platform 6.0 leverages the following domain and management software:

Cisco UCS Manager

Cisco Unified Computing System (UCS) Manager provides unified, embedded management for all software and hardware components in the Cisco UCS. Using [SingleConnect](#) technology, it manages, controls, and administers multiple chassis for thousands of virtual machines. Administrators use the software to manage the entire Cisco Unified Computing System as a single logical entity through an intuitive GUI, a command-line interface (CLI), or an XML API. The Cisco UCS Manager resides on a pair of Cisco UCS 6200 Series Fabric Interconnects using a clustered, active-standby configuration for high availability.

UCS Manager offers unified embedded management interface that integrates server, network, and storage. UCS Manager performs auto-discovery to detect inventory, manage, and provision system components that are added or changed. It offers comprehensive set of XML API for third part integration, exposes 9000 points of integration and facilitates custom development for automation, orchestration, and to achieve new levels of system visibility and control.

Service profiles benefit both virtualized and non-virtualized environments and increase the mobility of non-virtualized servers, such as when moving workloads from server to server or taking a server offline for service or upgrade. Profiles can also be used in conjunction with virtualization clusters to bring new resources online easily, complementing existing virtual machine mobility. For more Cisco UCS Manager information, go to: <http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-manager/index.html>

NetApp OnCommand System Manager

NetApp OnCommand System Manager is a simple, versatile GUI management product that enables administrators to easily configure and manage clustered NetApp storage systems. System Manager is designed with wizards and workflows that simplify common storage tasks such as creating volumes, LUNs, qtrees, shares, and exports, which saves time and helps prevent errors. OnCommand System manager manages the entire NetApp FAS line, from the entry-level FAS2200 series to the enterprise-class FAS8000 series, including systems running NetApp FlexArray storage virtualization software.



OnCommand System Manager is now bundled “on-box” with Data ONTAP 8.3 as a web service using HTML5. System Manager is enabled by default with Data ONTAP 8.3, and is accessible by a web browser pointed at the cluster management interface through HTTPS using the cluster administrator credentials.

OnCommand System Manager has some of the following key features:

- An intuitive browser-based GUI
- A wizard-driven configuration to get up and running quickly
- An automated non-disruptive (Data ONTAP) upgrade
- Storage provisioning and disk aggregate management
- CIFS, NFS, iSCSI, FC, and FCoE configuration

- Snapshot and SnapMirror management
- SnapVault support
- Storage virtual machine management
- Monitoring of HA pairs
- Support for all NetApp FAS systems and FlexArray software
- Support for NetApp All Flash FAS
- Support for up to 24 nodes

For more information, go to: [NetApp OnCommand System Manager](#)

NetApp SANtricity Storage Manager

NetApp SANtricity Storage Manager offers a powerful, easy-to-use interface for administering E-Series storage systems. With SANtricity software, your storage administrators can achieve maximum performance and utilization of storage through extensive configuration flexibility and custom performance tuning. The online administration, advanced data protection features, and extensive diagnostic capabilities of the SANtricity operating system mean your data is always available and fully protected on the storage system.

- Intuitive GUI. Blending robust functionality and ease of use, SANtricity Storage Manager is well suited for both full-time storage administrators who want complete control over their storage configuration, and part-time system administrators who prefer an intuitive interface and wizards that are designed to simplify storage management
- Multi-platform Client Support. Supported operating systems for the SANtricity Storage Manager client are Windows (32 and 64-bit); Linux (32 and 64-bit x86, 64-bit PowerPC, and 64-bit PowerPC Little Endian); IBM AIX; and Solaris (x86 and SPARC).

For more information, go to: [SANtricity Storage Manager 11.20](#), or [Introduction to NetApp E-Series E5500 with SANtricity 11.20](#).

Red Hat Enterprise Linux OpenStack Platform Installer

Red Hat Enterprise Linux OpenStack Platform installer manages the provisioning of Red Hat Enterprise Linux OpenStack Platform components on a set of machines. Red Hat Enterprise Linux OpenStack Platform installer provides web-based graphical user interface for managing the installation, configuration, and scalability of OpenStack environments. The application achieves this through discovering bootable hosts and mapping OpenStack services to them through web interface. Installer uses DHCP, DNS, and PXE services to perform OpenStack deployment on remote hosts.

Red Hat Enterprise Linux

Red Hat Enterprise Linux 7.1 lays the foundation for the open hybrid cloud and serves enterprise workloads across converged infrastructure. Red Hat Enterprise Linux 7.1 works on four platforms: Bare metal servers, virtual machines (VM), OpenStack based Infrastructure-as-a-Service (IaaS), and Platform-as-a-Service (PaaS) clouds. These, in turn, can be used together to form a robust, powerful datacenter and cloud environment for business. While Red Hat Enterprise Linux 7.1 still uses Kernel Virtual Machine (KVM) for datacenter and cloud virtualization, it also adopts container technology so that users can get even more

applications working on the same server hardware. Red Hat Enterprise Linux 7.1 provides many stability and performance upgrades.

Red Hat Enterprise Linux OpenStack Platform

Red Hat Enterprise Linux OpenStack Platform provides Infrastructure-as-a-Service (IaaS) foundation for public, private or hybrid cloud computing environment on top of Red Hat Enterprise Linux. Red Hat Enterprise Linux OpenStack Platform meets enterprise requirements with ability to extensively scale and provide a fault tolerant and highly available environment.

OpenStack is made up of many different moving parts. Because of its open nature, anyone can add additional components to OpenStack to meet their requirements. The Red Hat Enterprise Linux OpenStack Platform IaaS cloud is implemented by a collection of interacting services that control its computing, storage, and networking resources.

OpenStack Services

OpenStack has modular architecture with various services as its components.

Nova – Compute Service

Nova is the primary computing engine behind OpenStack and provides the base for OpenStack IaaS functionality. It can scale out horizontally on standard hardware in a distributed and asynchronous fashion, imparting fault tolerant and cost effective computing environment for virtual machines. The compute resources access can be controlled by virtual hardware profiles and tenants. They are used for deploying and managing large numbers of virtual machines and other instances that handle computing tasks.

Keystone – Identity Service

Keystone provides identity services for OpenStack. Identity Service provides a central directory of users mapped to the OpenStack services they can access. It acts as a common authentication system across the cloud operating system and can integrate with existing backend directory services. The Identity Service is comprised of the Keystone service, which responds to service requests, places messages in queue, grants access tokens, and updates the state database.

Cinder – Block Storage Service

OpenStack Cinder service provides compute instances with persistent block storage. Block storage is appropriate for performance sensitive scenarios such as databases, expandable file systems, or providing a server with access to raw block level storage. Persistent block storage can survive instance termination and can also be moved across instances like any external storage device. Cinder has volume snapshots capability for backing up the volumes.

Cinder allows for a variety of (pluggable) storage backend including both open source and proprietary. This solution uses storage systems from NetApp as the storage backend.

Neutron – Networking Service

OpenStack Networking is a scalable API-driven service for managing networks and IP addresses. OpenStack Networking gives users self-service control over their network configurations. Users can define, separate, and join networks on demand. Neutron API includes support for Layer 2 (L2) networking as well as an extension for layer 3 (L3) router constructions that enables routing between L2 networks and gateways to external networks. This allows for flexible network models to fit the requirements of different applications.

OpenStack Networking has a pluggable architecture that supports numerous virtual networking technologies as well as native Linux networking mechanisms including Open vSwitch and Linux Bridge.

Horizon – Dashboard

Horizon is the dashboard behind OpenStack that provides administrators and users a graphical interface to access, provision and automate cloud- based resources. Developers can access all of the components of OpenStack individually through an application programming interface (API) The dashboard provides system administrators a view of what is going on in the cloud, and to manage it as necessary. The dashboard runs through an HTTP service.

Glance – Image Service

Glance provides image services to OpenStack. Glance allows these images to be used as templates when deploying new virtual machine instances. OpenStack Image Service (Glance) provides discovery, registration, and delivery services for disk and server images. It can also be used to store and catalog multiple backups. The Image Service can store disk and server images in a variety of back-ends, including OpenStack object storage. The Image Service API provides a standard rest interface for querying information about disk images and lets clients stream the images to new servers.

Swift – Object Storage

Swift is OpenStack’s object service. Swift is a distributed scale-out that is highly available and provides for eventual consistency of data. Swift can be used to store lots of data efficiently, safely, and cheaply. Swift is **also OpenStack’s default backup** store.

Heat – Orchestration Service

OpenStack Heat is an orchestration service that manages the life-cycle of applications within an OpenStack environment using templates. Heat is capable of deploying multi- instance applications called stacks and managing application lifecycle.

Ceilometer – Telemetry Service

Ceilometer provides telemetry services for billing services to individual users of the cloud. It keeps a **verifiable count of each user’s system usage of various components of an OpenStack cloud.** The delivery of counters is traceable and auditable.

Heat Templates

Heat templates are written in a declarative format. A template defines what resources to deploy rather than how to deploy those resources. This is similar to the approach used by popular configuration tools such as Puppet, Ansible, and Chef. Configuration tools focus on the configuration of a system, whereas Heat focuses on resource provision and relies on cloud-init scripting to handle system configuration. A template may create and configure a large list of resources thus supporting complex application stacks

OpenStack High Availability

OpenStack Environment consists of stateless, shared-nothing services such as Keystone, Glance, Swift, Nova, Neutron, Horizon, Heat, Ceilometer, etc. and underlying infrastructure components that OpenStack services use for inter-service communication and to save persistent data such as MariaDB database, and a message broker called RabbitMQ.

Building a scale-out controller would require setting the services and infrastructure components (database and message broker) in Active-Active configuration. You need to confirm that they are capable of adding more nodes to the cluster as the load increases, and load balancing, the API can request among the nodes. While most of the services are Active-Active, there are some services still in Active-Passive mode.

Red Hat Enterprise Linux OpenStack Platform is now fully integrated with the Red Hat Enterprise Linux High Availability Add-On, to support highly available environments for customer deployments. This means that a cloud infrastructure can now be set up so that if one of its controller nodes/service fails, the machine/service can be brought back up with no or minimal impact.

Memcached

Memcached is fast in-memory key-value cache software that is used by OpenStack components for caching data and increasing performance.

Galera

Galera Cluster is a synchronous multi-master cluster for MariaDB with a valuable availability and scaling features required for OpenStack services.

HAProxy

HAProxy is a software layer-7 load balancer used to cater to all clustered OpenStack API components and perform SSL terminations. HAProxy can be added as a resource to the Pacemaker software that runs on the Controller nodes where HAProxy is situated.

Pacemaker

Pacemaker is the clustering software used to ensure the availability of services and systems running on the **controller nodes and handles high availability leveraging 'corosync' underneath. Pacemaker manages the Galera nodes and HAProxy.**

Fencing

Fencing is an operation that completely isolates a failed node preventing split brain situation of clusters. Pacemaker has a built in integration with fencing.

High Availability Modes

Following HA Modes are supported

Active-Active:

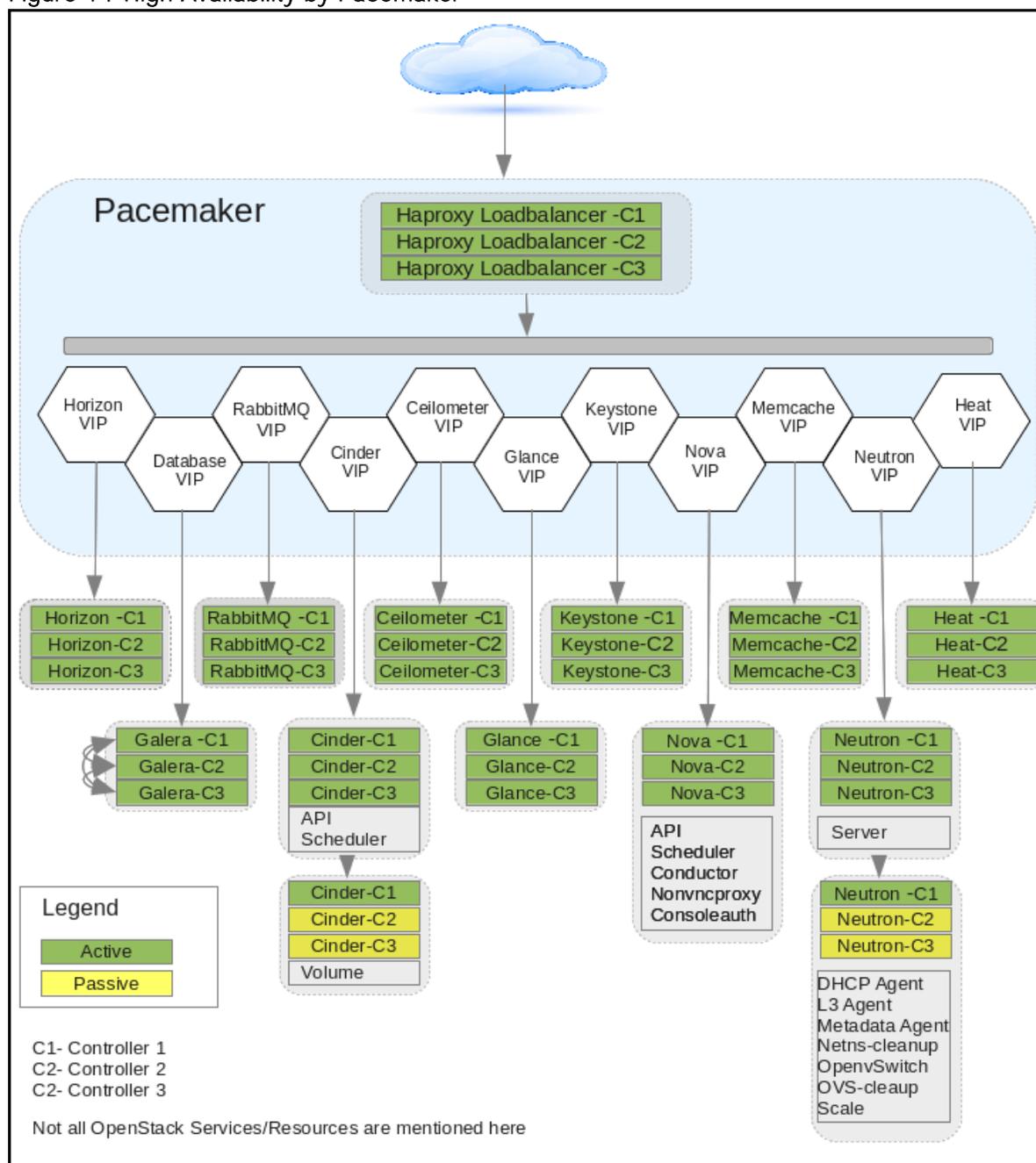
In an Active-Active configuration, all system components are kept online; if a component fails, its load is passed to the next active component. The majority of OpenStack services are configured to run in active/active configuration through the Pacemaker resource manager.

Active-Passive:

In this configuration, only one instance of the service runs in the cluster at a time and get started if pacemaker detects the service is offline. A small number of OpenStack services use an active/passive configuration for high availability.

Figure 14 describes the services that are Active-Active or Active-Passive in the current Red Hat Enterprise Linux OpenStack Platform 6.0 version deployed through current version of Installer.

Figure 14 High Availability by Pacemaker



Other OpenStack Supporting Technologies

RabbitMQ

OpenStack services use enterprise messaging to communicate tasks and state changes between clients, service endpoints, service schedulers, and instances. RabbitMQ is open source message broker software that implements the Advanced Message Queuing Protocol (AMQP)



RabbitMQ is default and recommended message broker service on Red Hat Enterprise Linux OpenStack Platform 6.0.

MariaDB

A community developed fork of the MySQL relational database management system. This database stores most of the build-time and run-time state information for the cloud infrastructure including available instance types, networks, and the state of running instances in the compute fabric. Although OpenStack theoretically supports any SQL-Alchemy compliant database, MariaDB is the database shipped with Red Hat Enterprise Linux 7 and used by Red Hat Enterprise Linux OpenStack Platform 6.0 as default.

KVM

Kernel-based Virtual Machine (KVM) is a full virtualization solution for Linux on x86 and x86_64 hardware containing virtualization extensions for both Intel and AMD processors. It consists of a loadable kernel module that provides the core virtualization infrastructure. Red Hat Enterprise Linux OpenStack Platform Compute uses KVM as its underlying hypervisor to launch and control virtual machine instances.

OpenStack Networking

An overview of networking basics for OpenStack deployment are covered in this section

Modular Layer 2 (ML2) Core

The Modular Layer 2 (ml2) plug-in is a framework that provides more flexibility in terms of providing simultaneous access to various networking technologies. Rather than rewrite code for huge monolithic core plugins associated with L2 agents like Open vSwitch, Linux Bridge etc., Mechanism Drivers can be written to the much simpler ML2 framework for these plugins.

ML2 Network Types:

- Flat: All instances reside on the same network, which can also be shared with the hosts. No VLAN tagging or other network segregation takes place.
- Local: Instances reside on the local compute host and are effectively isolated from any external networks.
- VLAN: Networking allows users to create multiple provider or tenant networks using VLAN IDs (802.1Q tagged) that correspond to VLANs present in the physical network. This allows instances to communicate with each other across the environment. They can also communicate with dedicated servers, firewalls, load balancers and other networking infrastructure on the same layer 2 VLAN.
- VXLAN: Virtual Extensible Local Area Network helps create a logical network for virtual machines across different networks. In other words one can create a Layer 2 network on top of layer 3 through encapsulation. The basic use case for VXLAN is to connect two or more Layer 3 networks and makes them look like they share the same Layer 2 domain. This allows for virtual machines to live in two disparate networks yet still operate as if they were attached to the same L2 thus enhancing scalability. The VXLAN networks are broken-down as segments and same IP address can exist across different segments. A combination of Machine Address Control (MAC) and VXLAN Network Identifier (VNI) makes each VM connection unique.
- GRE: Generic Routing Encapsulation (GRE) segmentation. A network layout in which tunnels are used to segregate and carry network traffic over individual tenant networks.

Solution Design

Hardware and Software Revisions

It is important to note that following hardware and software versions have been validated in FlexPod with Red Hat Enterprise Linux OpenStack Platform 6.0

Table 2 outlines the hardware components used in the validation efforts of this solution along with the firmware versions.

Table 2 Validated Hardware Versions

Layer	Production Hardware	Firmware/Details
Compute	Cisco UCS Fabric Interconnects 6248UP	2.2(3g)
	Cisco UCS B200 M4 Server	2.2(3g)
	Cisco UCS Manager	2.2(3g)
	Cisco eNIC Driver	2.1.1.75
Network	Cisco Nexus 9372 NX-OS	6.1(2) I3(2)
Storage	NetApp FAS8040	Dual Controller HA Pair
	NetApp DS2246	SAS Disk Shelf Qty. 24 900GB 2.5" SAS disks
	NetApp E-Series E5560	Dual Controller HA Pair Qty. 60 2TB 3.5" SAS disks

Table 3 outlines the software components involved in FlexPod with Red Hat Enterprise Linux OpenStack Platform 6 architecture.

Table 3 Validated Software Versions

Layer	Software	Version
Operating System	Red Hat Enterprise Linux	7.1
OpenStack Platform	Red Hat Enterprise Linux OpenStack Platform	6.0 (Juno Based)
	Red Hat Enterprise Linux OpenStack Platform Installer	6.0
Storage	Data ONTAP	8.3.0
	SANtricity OS	8.20.08.00
	OnCommand System Manager	8.3.0
	SANtricity Storage Manager	11.20.0X00.0010
	Cinder	Drivers distributed with RHEL-OSP 6.0
	Swift	Packages distributed with RHEL-OSP 6.0
Networking	Cisco Nexus 1000v VSM for KVM	5.2(1) SK3 (2.2b)

	Cisco Nexus 1000v VEM module for each compute and controller node	5.2(1) SK3 (2.2b)
--	---	-------------------

Solution Components

FlexPod with Red Hat Enterprise Linux OpenStack Platform 6 is made up of the following components (Table 4).

Table 4 List of Solution Components

Components	Quantity	Comments
Cisco UCS B200 M4 Blade Servers	8	4 Per Chassis
Cisco UCS 5108 Chassis	2	
Cisco UCS 2104XP IO Modules	4	2 Per Chassis
Cisco UCS 6248UP Fabric Interconnect	2	
Cisco Nexus 9372PX	2	
Cisco Nexus 1000v for KVM	2	Virtual Appliance
NetApp FAS8040 System	1	Two Nodes for HA
NetApp DS2246 with 24 900GB SAS drives	1	
NetApp E5560 Storage System	1	Dual Controllers for HA

Architectural Overview

The FlexPod architecture is highly modular or “podlike.” Although each customer’s FlexPod unit varies in its exact configuration, after a FlexPod unit is built, it can easily be scaled as requirements and demands change. The unit can be scaled both up (adding resources to a FlexPod unit) and out (adding more FlexPod units).

Specifically, FlexPod is a defined set of hardware and software that serves as an integrated foundation for both virtualized and non-virtualized solutions. Red Hat Enterprise Linux OpenStack Platform 6.0 built on FlexPod includes NetApp FAS Storage, NetApp E-Series Storage, Cisco Nexus networking, the Cisco **Unified Computing System™**, and **Red Hat Enterprise Linux OpenStack Platform 6**. **The design is flexible enough that networking, computing, and storage can fit in one data center rack or be deployed according to a customer’s infrastructure design. Port density enables the net-working components to accommodate multiple configurations of this kind.**

One benefit of the FlexPod architecture is the ability to customize or “flex” the environment to suit a customer’s requirements. This is why the reference architecture detailed in this document highlights the resiliency, cost benefit, and ease of deployment of an Ethernet based storage solution.

Figure 15 illustrates FlexPod with Red Hat Enterprise Linux OpenStack Platform 6 components and the network connections for a configuration with IP-based storage. This design uses Cisco Nexus® 9372, Cisco

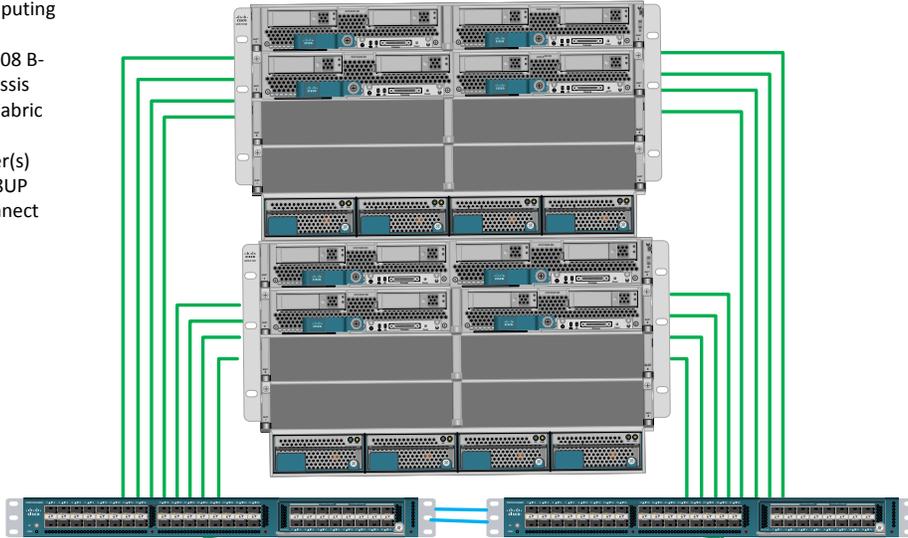
UCS B200 M4 servers with the Cisco UCS 1340 virtual interface card (VIC), NetApp FAS8040, and NetApp E5560 connected in a highly available design by using Cisco Virtual PortChannels (vPCs). In this infrastructure, NetApp FAS8040 provides iSCSI boot LUNs for boot from SAN, NetApp Cinder driver backend for OpenStack Cinder volumes, and the Glance image store. Furthermore, the NetApp E5560 provides OpenStack Swift.

Figure 15 FlexPod with Red Hat Enterprise Linux OpenStack Platform 6 Architecture

FlexPod with Red Hat Enterprise Linux OpenStack Platform 6

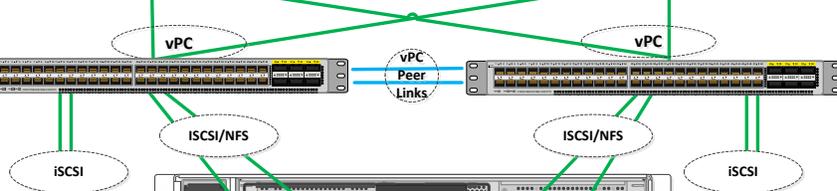
Cisco Unified Computing System

- Cisco Nexus 5108 B-Series UCS Chassis
- Cisco 2204XP Fabric Extenders
- B200 M4 Server(s)
- Cisco UCS 6248UP Fabric Interconnect



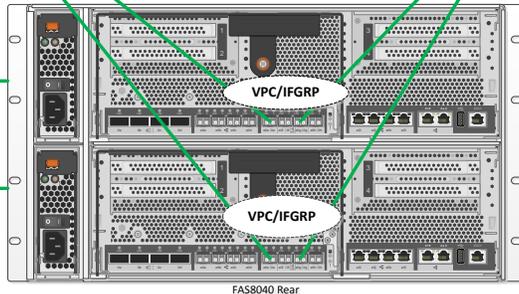
Cisco Access Layer

- Cisco Nexus 9372PX



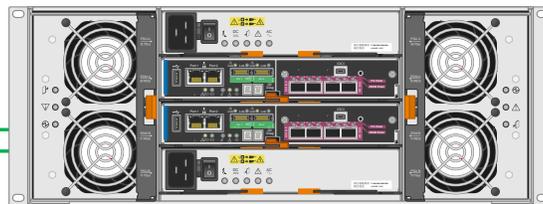
NetApp FAS Storage

- 1 NetApp FAS8040 Array
- 2 10GB NIC per Controller



NetApp E-Series Storage

- 1 NetApp DE5560 Array
- 2 NetApp E5500 4x 10Gb iSCSI Controller
- 2 10GB HIC per Controller



— 10Gb Ethernet —

OpenStack Platform Architecture

Red Hat Enterprise Linux OpenStack Platform 6 is installed onto the Cisco UCS B200 M4 servers in the FlexPod topology as shown in Figure 15. There are three different server roles in the topology: Installer, Controller, and Compute. FlexPod with Red Hat Enterprise Linux OpenStack Platform 6 is a highly available architecture with three controller nodes. In this architecture, networker node and controller node are combined. The controller nodes run the majority of the OpenStack services including Keystone, Nova, Neutron-server, Cinder, Glance, Heat, Horizon, and the database, which all the services utilize. In addition to that, the Controller hosts run all the neutron agents including the DHCP agent in active/passive mode. The Compute nodes host all the VMs and run the nova-compute service and open-vswitch agent. Within OpenStack, resources can be divided administratively into tenants (also called projects in Horizon). Each tenant can have user privileges set to either admin or member.

Prerequisites

There are just a few items you'll want to complete prior to beginning setup. The equipment needs to be cabled, racked, and powered up before continuing the below steps. Next, you will need to have an active Red Hat account with access to the OpenStack Platform subscriptions through purchased entitlements.

- An NTP Server is an absolute requirement for the Red Hat Enterprise Linux OpenStack Platform installer to successfully complete an OpenStack deployment.
- Have Network Address Translation (NAT) configured for the management network to provide external access for the Controller and Compute hosts to register with Red Hat Subscription Services, subscribe to the proper repositories, and obtain package updates.
- Have the Red Hat Enterprise Linux Server 7.1 ISO image (rhel-server-7.1-x86_64-dvd.iso) on hand and readily available.

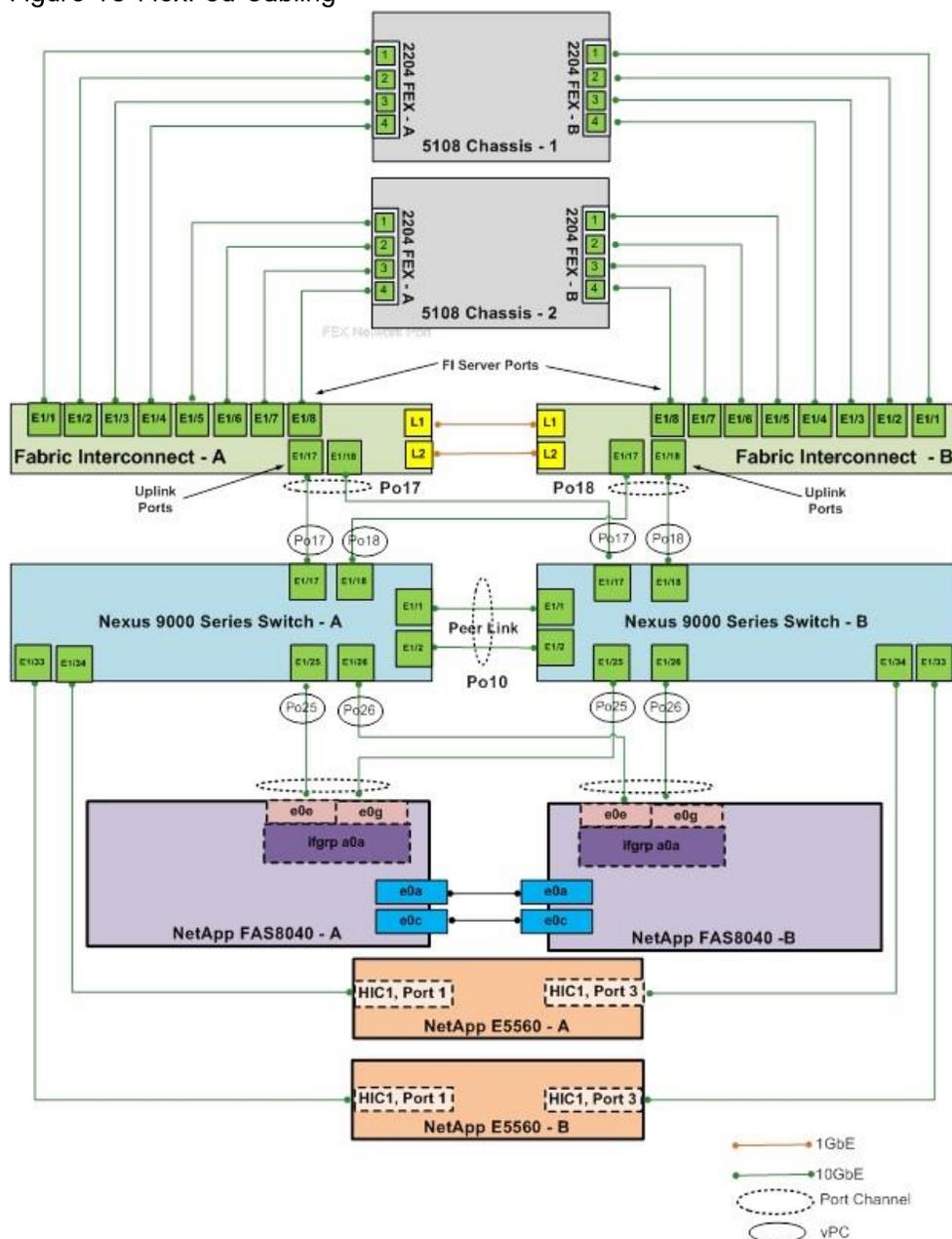
Deployment Hardware and Software

Physical Infrastructure

FlexPod with Red Hat Enterprise Linux OpenStack Platform Physical Topology with Cabling

Figure 16 shows the cabling diagram for a FlexPod with RHEL-OSP 6. The NetApp storage controller and disk shelves should be connected according to best practices for the specific storage controller and disk shelves. For disk shelf cabling, refer to the sections titled NetApp FAS Storage Configuration or NetApp E-Series Storage Configuration.

Figure 16 FlexPod Cabling



FlexPod Cabling Detail

The information in this section is provided as a reference for cabling the physical equipment in a FlexPod environment. To simplify cabling requirements, Table 5 includes both local and remote device and port locations.



Be sure to follow the cabling guidance in this section. A failure to do so may result in necessary changes to the deployment procedures, as specific ports are used during the configuration and setup of network, compute, and storage infrastructure in subsequent sections.

Table 5 Cabling Detail

Local Device	Local Port	Connection	Remote Device	Remote Port
Cisco Nexus 9372 - Switch A	Eth1/1	10GbE	Cisco Nexus 9372 Series Switch B	Eth1/1
	Eth1/2	10GbE	Cisco Nexus 9372 Series Switch B	Eth1/2
	Eth1/17	10GbE	Cisco UCS Fabric Interconnect A	Eth1/17
	Eth1/18	10GbE	Cisco UCS Fabric Interconnect B	Eth1/17
	Eth1/25	10GbE	NetApp FAS8040 Node A	e0e
	Eth1/26	10GbE	NetApp FAS8040 Node B	e0e
	Eth1/34	10GbE	NetApp E5560 Controller A	Port1
	Eth1/33	10GbE	NetApp E5560 Controller B	Port1
	Eth1/23	1GBE	Management	Port Any
	MGMT0	1GbE	Cisco Catalyst 2960S	Any
Cisco Nexus 9372- Switch B	Eth1/1	10GbE	Cisco Nexus 9372 Series Switch A	Eth1/1
	Eth1/2	10GbE	Cisco Nexus 9372 Series Switch A	Eth1/2
	Eth1/17	10GbE	Cisco UCS Fabric Interconnect A	Eth1/18
	Eth1/18	10GbE	Cisco UCS Fabric Interconnect B	Eth1/18
	Eth1/25	10GbE	FAS8040 Node A	e0g
	Eth1/26	10GbE	FAS8040 Node B	e0g
	Eth1/34	10GbE	E5560 Controller A	Port2
	Eth1/33	10GbE	E5560 Controller B	Port2
	Eth1/23	1GBE	Management	Port Any
	MGMT0	100MbE	Cisco Catalyst 2960S	Any

Local Device	Local Port	Connection	Remote Device	Remote Port
NetApp FAS8040 Node A	e0P	1GbE	SAS shelves	ACP port
	e0a	10GbE	Cluster Connection to Node B	e0a
	e0c	10GbE	Cluster Connection to Node B	e0c
	e0e	10GbE	Cisco Nexus 9372 Series Switch A	Eth1/25
	e0g	10GbE	Cisco Nexus 9372 Series Switch B	Eth1/25
	e0M	1GbE	Cisco Catalyst 2960S	Any
NetApp FAS8040 Node B	e0P	1GbE	SAS shelves	ACP port
	e0a	10GbE	Cluster Connection to Node A	e0a
	e0c	10GbE	Cluster Connection to Node A	e0c
	e0e	10GbE	Cisco Nexus 9372 Series Switch A	Eth1/26
	e0g	10GbE	Cisco Nexus 9372 Series Switch B	Eth1/26
	e0M	1GbE	Cisco Catalyst 2960S	Any
NetApp E5560 Controller A	Controller A, HIC 1, Port 1	10GbE	Cisco Nexus 9372 Series Switch A	Eth1/34
	Controller A, HIC 1, Port 3	10GbE	Cisco Nexus 9372 Series Switch B	Eth1/34
	1GbE Management Connector 1	1GbE	Cisco Catalyst 2960S	Any
NetApp E5560 Controller B	Controller B, HIC 1, Port 1	10GbE	Cisco Nexus 9372 Series Switch A	Eth1/33
	Controller B, HIC 1, Port 3	10GbE	Cisco Nexus 9372 Series Switch B	Eth1/33
	1GbE Management Connector 1	1GbE	Cisco Catalyst 2960S	Any
Cisco UCS Fabric Interconnect A	Eth1/1	10GbE	Chassis 1 FEX A	port 1
	Eth1/2	10GbE	Chassis 1 FEX A	port 2
	Eth1/3	10GbE	Chassis 1 FEX A	port 3

Local Device	Local Port	Connection	Remote Device	Remote Port
	Eth1/4	10GbE	Chassis 1 FEX A	port 4
	Eth1/5	10GbE	Chassis 2 FEX A	port 1
	Eth1/6	10GbE	Chassis 2 FEX A	port 2
	Eth1/7	10GbE	Chassis 2 FEX A	port 3
	Eth1/8	10GbE	Chassis 2 FEX A	port 4
	Eth1/17	10GbE	Cisco Nexus 9372 A	Eth 1/17
	Eth1/18	10GbE	Cisco Nexus 9372 B	Eth 1/17
	MGMT0	1GbE	Cisco Catalyst 2960S	Any
	L1	1GbE	UCS Fabric Interconnect B	L1
	L2	1GbE	UCS Fabric Interconnect B	L2
Cisco UCS Fabric Interconnect B	Eth1/1	10GbE	Chassis 1 FEX B	port 1
	Eth1/2	10GbE	Chassis 1 FEX B	port 2
	Eth1/3	10GbE	Chassis 1 FEX B	port 3
	Eth1/4	10GbE	Chassis 1 FEX B	port 4
	Eth1/5	10GbE	Chassis 2 FEX B	port 1
	Eth1/6	10GbE	Chassis 2 FEX B	port 2
	Eth1/7	10GbE	Chassis 2 FEX B	port 3
	Eth1/8	10GbE	Chassis 2 FEX B	port 4
	Eth1/17	10GbE	Cisco Nexus 9372 A	Eth 1/18
	Eth1/18	10GbE	Cisco Nexus 9372 B	Eth 1/18
	MGMT0	1GbE	Cisco Catalyst 2960S	Any
	L1	1GbE	UCS Fabric Interconnect B	L1
	L2	1GbE	UCS Fabric Interconnect B	L2

Configuration Guidelines

Required VLANs

The following VLANs are needed in this validated design. Use Table 6 as a worksheet for the VLAN requirements of this deployment.

Table 6 VLANs used in FlexPod with Red Hat Enterprise Linux OpenStack Platform 6

VLAN Name	Variable	VLAN Purpose	VLAN ID or VLAN Range Used in This Design for Reference
Management	<<var_mgmt_vlan_id>>	VLAN for in-band management network. Also used for OpenStack Public API traffic	10
PXE	<<var_pxe_vlan_id>>	Provisioning network used by the RHEL-OSP Installer server for deploying RHEL 7.1 and OpenStack Platform. This network is also used for OpenStack management traffic.	20
NFS	<<var_nfs_vlan_id>>	Storage network for carrying Cinder and Glance traffic	30
iSCSI-40	<<var_iscsi_A_vlan_id>>	VLAN for iSCSI traffic for boot from SAN (Fabric A)	40
iSCSI-41	<<var_iscsi_B_vlan_id>>	VLAN for iSCSI traffic for boot from SAN (Fabric B)	41
Swift-50	<<var_swift_A_vlan_id>>	Storage VLAN for Swift traffic (Fabric A)	50
Swift-51	<<var_swift_B_vlan_id>>	Storage VLAN for Swift traffic (Fabric B)	51
Provider	<<var_provider_vlan_range>>	Tenant provider VLANs	60-69
Tenant	<<var_tenant_vlan_range>>	Tenant private networks for VM data traffic	70-200
External	<<var_external_vlan_id>>	VLAN for public network. Provide access to outside world for deploying OpenStack platform.	215
MCAS-21	<<var_mcas_vlan_id>>	VLAN for OpenStack Management, Cluster Management, Admin API, and Storage Clustering traffic.	21



Provider and Tenant VLANs have been pre-configured. In this validated deployment, VLANs 60 through 69 have been created as provider VLANs. VLAN 70 through 200 has been configured as tenant private VLANs. However, these VLAN ranges can be modified depending on scale requirements.

OpenStack Horizon service as the dashboard and OpenStack CLI is utilized in the FlexPod with RHEL-OSP solution to manage the Tenants, Users, Networks and VMs through the OpenStack service API's; and to manage the persistent storage volumes through **OpenStack and NetApp's Cinder driver**. **The Provider VLANs** and Internal VLANs are provisioned on the OVS virtual switch on the UCS OpenStack compute nodes, through the OpenStack Modular Layer 2 (ML2) plugin. These VLANs cannot be provisioned automatically on the Cisco UCS Fabric Interconnect (through UCS Manager) or the Cisco Nexus 9372 switches, due to some limitations in the ML2 plugin capabilities in the Juno release of OpenStack. These will be supported in future OpenStack and Red Hat Enterprise Linux OpenStack Platform releases. Currently, these VLANs need to be pre-provisioned (in pools of VLANs) on the Cisco UCS Fabric Interconnect (UCSM vNIC templates) and Cisco Nexus 9372 switches.

The provider VLAN pool is intended to be a set of VLANs that are routable outside the OpenStack cluster. These VLANs will have their gateway as a Switched Virtual Interface (SVI) on the Nexus 9000 switch. Tenants will place their external facing VMs on these networks. These networks can be shared among several tenants, using security groups to protect and/or enable inter-tenant traffic. The tenant VLAN pool is the set of VLANs neutron is configured to use for tenant-created networks. Only the tenant that created the network will have access to it. No SVI will be created on the Nexus 9000 switch for these networks since global routing is not required.



Dynamic Layer-3 routing of these provider networks can also be configured, but it is beyond the scope of this document. However, external VLAN 215 was used to test external connectivity of VM instances in our environment.

Global Configuration Variables

Table 7 lists the global configuration variables used in this document. This can be completed as a worksheet before you begin the deployment.

Table 7 Global Configuration Variables

Variable	Description	Value
<<var_global_ntp_server_ip>>	NTP server IP address	-
<<var_nameserver_ip>>	DNS Server IP(s)	-
<<var_dns_domain_name>>	DNS Domain name	-
<<external_network_address>>	External network address	-

Logical Topology

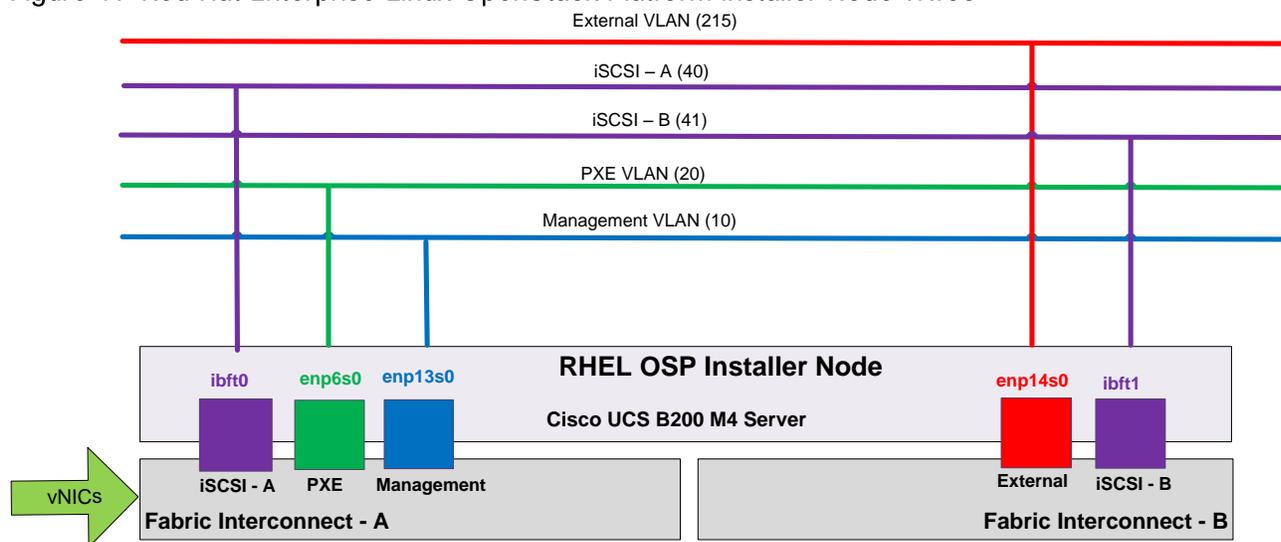
Network data is logically segmented using Virtual Local Area Networks (VLANs) in this reference architecture. The entire range of VLANs as shown in table 8 are configured ahead of time on both Nexus 9000 series switch as well as the UCS Fabric Interconnects.

The Cisco VIC 1340 allows for the creation of virtual NICs (vNIC) presented to the operating system. These vNICs can be associated with a single VLAN or as a trunk of several VLANs. vNIC placement policy in Cisco UCS is configured for consistent placement of host level interfaces.

RHEL-OSP Installer Node Topology

In this FlexPod deployment, the RHEL-OSP Installer server is configured with 5 vNICs as shown in Figure 17. iSCSI-A and iSCSI-B are used to provide iSCSI LUN for boot from SAN. Two iSCSI interfaces provide multi-pathing and high availability. These are overlay vNICs for iSCSI interfaces that will be configured in subsequent sections. PXE and Management vNICs are mapped to UCS fabric “A” and dynamically failover to fabric “B” for redundancy. Similarly, the External vNIC is mapped to Fabric “B” and can dynamically failover to fabric “A” in case of the failure of fabric “B”.

Figure 17 Red Hat Enterprise Linux OpenStack Platform Installer Node vNICs

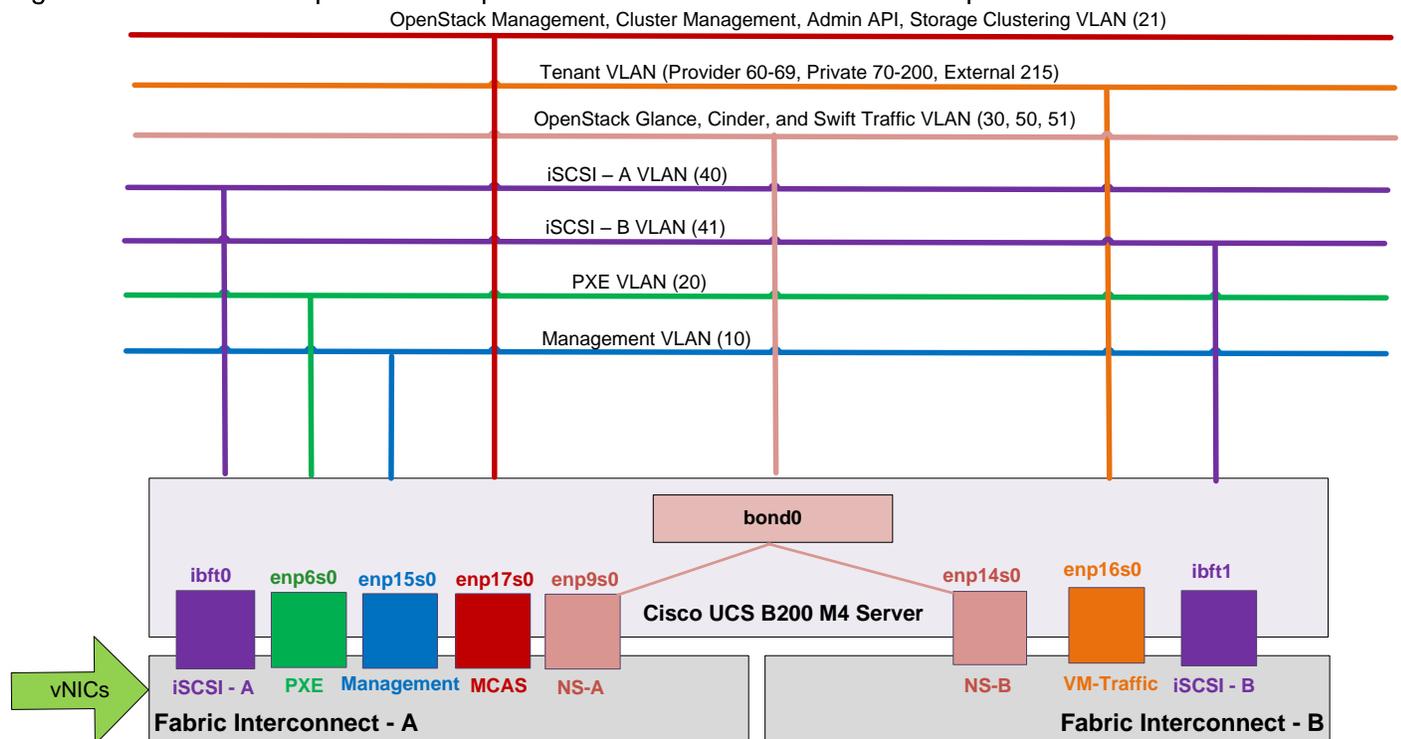


RHEL-OSP Controller and Compute Node Topology

Controller and Compute servers are configured with 8 vNICs as shown in .

- Two iSCSI vNICs are used to provide iSCSI LUN for boot from SAN. They will be providing multiple paths to the boot LUN.
- The PXE vNIC is used for PXE/Provisioning network. This network is used by the Red Hat Enterprise Linux OpenStack Platform Installer to build OpenStack controller and compute hosts. Installer uses this network to boot hosts using PXE, deploy and configure hosts based on their roles (Controller or Compute). The PXE vNIC is mapped to fabric “A” and can dynamically failover to fabric “B” for redundancy.
- The Management vNIC is created for hosts management and also carries OpenStack public API traffic. This is also mapped to fabric “A” and dynamically failover to fabric “B”.
- The MCAS vNIC is created to carry OpenStack Management, Cluster Management, Admin API, and Storage Clustering traffic. MCAS vNIC is mapped to fabric “A” and can dynamically failover to fabric “B”, in case any failure occurs on fabric “A” or uplink connectivity of fabric “A”.
- The VM-Traffic vNIC is mapped to Fabric “B” and is configured to trunk provider, tenant, and external VLAN. VM-Traffic vNIC is dynamically failover to fabric “A” in case of failure of fabric “B”.
- The NS-A and NS-B (abbreviation for Network Storage) vNICs are configured to trunk NFS (VLAN 30) and Swift (VLAN 50 and 51) traffic. These vNICs are not configured to failover to the surviving Fabric, instead the host will be configured to treat these two vNICs as bonded interfaces and performs host level interface failover for redundancy as shown in Figure 18.

Figure 18 Red Hat Enterprise Linux OpenStack Platform Controller and Compute Node vNICs



Service Profile and vNIC Templates

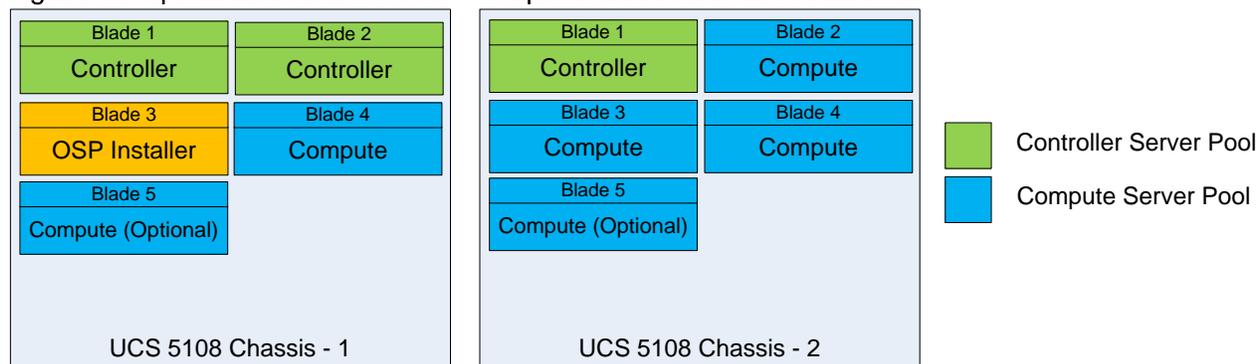
Service profiles and vNICs are created from an updating template which inherit all the properties of the template itself and remain connected to the template. Any changes to the template automatically updates the service profiles and vNIC created from the template.

Server Pools

Server pools will be utilized to divide the OpenStack server roles for easy of deployment and scalability. These pools will also decide the placement of server roles within the infrastructure. Following two server pools are created (Figure 19).

- OpenStack Controller server pool
- OpenStack Compute server pool

Figure 19 OpenStack Server Pools For OpenStack Server Role Placement



The Compute server pool will allow quick provisioning of additional compute hosts by adding those servers into the compute server pool, and create service profiles from the compute service profile template. These newly provisioned compute hosts can easily be added into an existing OpenStack deployment through the **RHEL-OSP Installer's web interface**.



Note: Two additional compute hosts marked as optional in Figure 19 have been used in the reference deployment to test and validate certain use cases such as scaling the environment, which will be discussed in the later sections of this document.

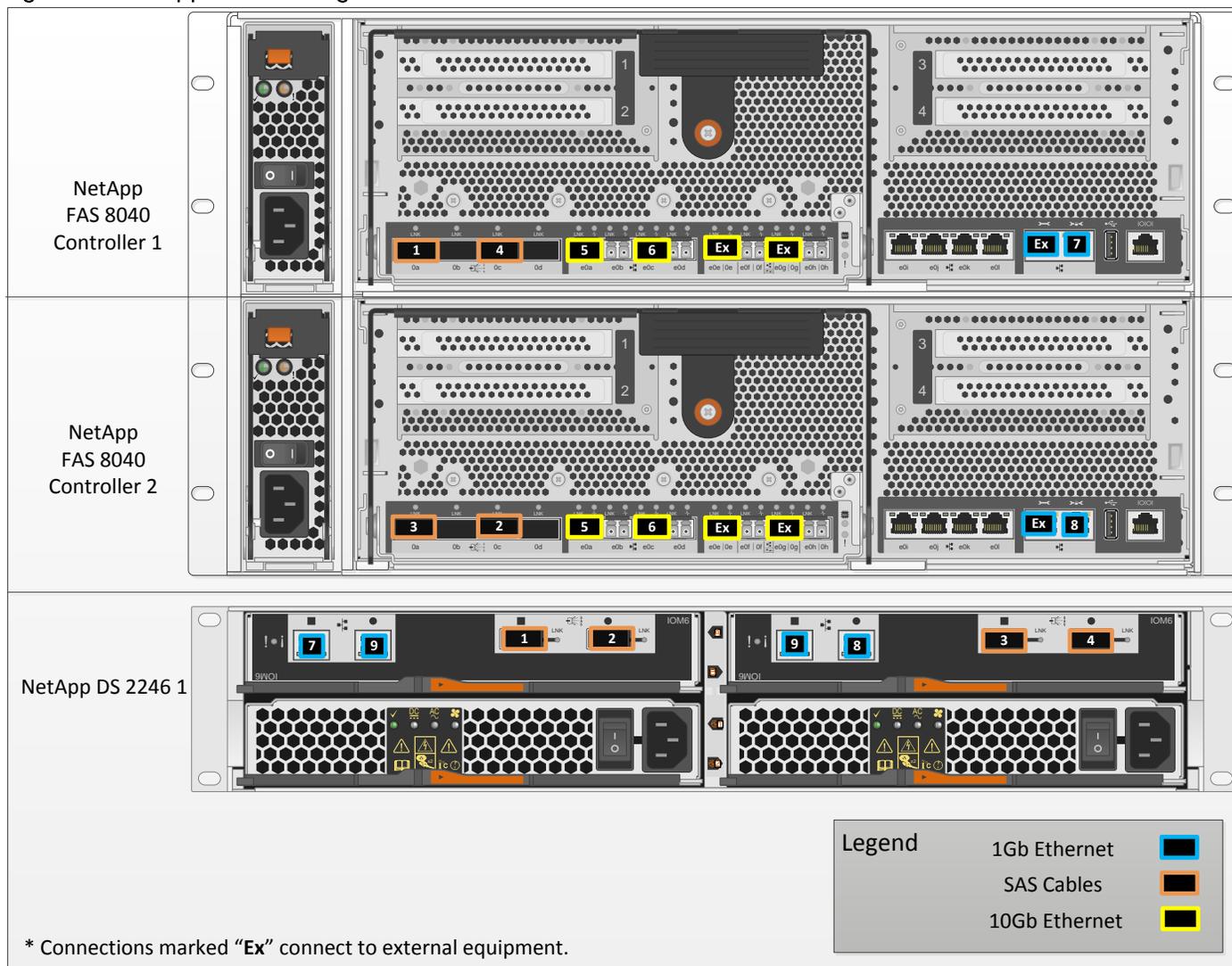
NetApp FAS Storage Configuration

For instructions on the physical installation of FAS8000 controllers, follow the procedures in the [FAS8000 Series documentation](#) on the NetApp Support site. When planning the physical location of a storage system, refer to the following sections in the [Site Requirements Guide](#):

- Site Preparation
- System Connectivity Requirements
- Circuit Breaker, Power Outlet Balancing, System Cabinet Power Cord Plugs, and Console Pinout Requirements
- 8000 Series Systems

The physical connections in this solution are represented in 0. Match the numbers provided (for example, 1 to 1, 2 to 2, and so on) to accurately configure intradevice cabling. Note that connections marked Ex represent connections that are external to the FAS itself; in other words, these connections run to other pieces of equipment.

Figure 20 NetApp FAS Cabling



Disk Shelves

NetApp storage systems support a wide variety of disk shelves and disk drives. Visit the NetApp Support site to view a complete list of supported disk shelves. This solution is built on a single DS2246 disk shelf with 24 900GB SAS Disks. These disks provide an ample amount of storage at a moderate price point, and are recommended for an OpenStack deployment. When using SAS disk shelves with NetApp storage controllers, refer to the SAS Disk Shelves Universal SAS and ACP Cabling Guide for information about cabling guidelines.

Clustered Data ONTAP 8.3

This procedure assumes that the storage system has been installed, cabled, and is ready for setup. For detailed information about storage system installation, see the resources listed in the previous section.

Complete the Configuration Worksheet

Before performing the following procedures, review the configuration worksheets in the Clustered Data ONTAP 8.3 Software Setup Guide to learn about the information required to configure clustered Data

ONTAP. Table 8 lists the information you need to configure two clustered Data ONTAP nodes. You should customize the cluster detail values with the information that is applicable to your deployment.

Table 8 Cluster Detail for the Clustered Data ONTAP Software Configuration

Cluster Detail	Cluster Detail Value
Cluster name	<<var_clustername>>
Clustered Data ONTAP base license	<<var_cluster_base_license_key>>
Clustered Data ONTAP NFS license	<<var_nfs_license>>
Clustered Data ONTAP iSCSI license	<<var_iscsi_license>>
Clustered Data ONTAP FlexClone [®] license	<<var_flexclone_license>>
Cluster management IP address	<<var_clustermgmt_ip>>
Cluster management netmask	<<var_clustermgmt_mask>>
Cluster management port	<<var_clustermgmt_port>>
Cluster management gateway	<<var_clustermgmt_gateway>>
Cluster node 01 IP address	<<var_node01_mgmt_ip>>
Cluster node 01 netmask	<<var_node01_mgmt_mask>>
Cluster node 01 gateway	<<var_node01_mgmt_gateway>>
Cluster node 01 Service Processor IP address	<<var_node01_sp_ip>>
Cluster node 01 Service Processor netmask	<<var_node01_sp_netmask>>
Cluster node 01 Service Processor gateway	<<var_node01_sp_gateway>>
Cluster node 02 IP address	<<var_node02_mgmt_ip>>
Cluster node 02 netmask	<<var_node02_mgmt_mask>>
Cluster node 02 gateway	<<var_node02_mgmt_gateway>>
Cluster node 02 Service Processor IP address	<<var_node02_sp_ip>>
Cluster node 02 Service Processor netmask	<<var_node02_sp_netmask>>
Cluster node 02 Service Processor gateway	<<var_node02_sp_gateway>>

Install Clustered Data ONTAP 8.3

Perform the following procedure on both of the storage nodes if the running version of Data ONTAP is lower than 8.3. If you already have Data ONTAP version 8.3 installed on your storage system, skip to the section titled “Create Cluster on Node 01.”

1. Connect to the storage system console port. You should see a Loader prompt. However, if the storage system is in a reboot loop, press Ctrl-C to exit the autoboot loop when you see this message:

```
Starting AUTOBOOT press Ctrl-C to abort
```

2. Set boot monitor defaults.

```
LOADER>set-defaults
```

3. Set the variable to boot Clustered Data ONTAP.

```
LOADER>setenv bootarg.init.boot_clustered true
saveenv
```

4. Allow the system to boot up.

```
LOADER>autoboot
```

5. Press Ctrl-C when the Press Ctrl-C for Boot Menu message appears.



If Data ONTAP 8.3 is not the version of software being booted, proceed with the following steps to install new software. If Data ONTAP 8.3 is the version being booted, then select option 8 and `y` to reboot the node and continue with the section titled “Create Cluster on Node 01.”

6. To install new software, first select option 7.

```
Please choose one of the following:
```

```
(1) Normal Boot.
(2) Boot without /etc/rc.
(3) Change password.
(4) Clean configuration and initialize all disks.
(5) Maintenance mode boot.
(6) Update flash from backup config.
(7) Install new software first.
(8) Reboot node.
Selection (1-8)? 7
```

7. Enter `y` to perform a nondisruptive upgrade.

```
This procedure is not supported for Non-Disruptive Upgrade on an HA pair.
The software will be installed to the alternate image, from which the node is not currently running. Do you
want to continue? {y|n} y
```

8. Select `e0M` for the network port you want to use for the download.

```
Select the network port you want to use for the download (for example, 'e0a') [e0M] e0M
```

9. Enter `y` to reboot now.

```
The node needs to reboot for this setting to take effect. Reboot now? {y|n}
(selecting yes will return you automatically to this install wizard) y
```

10. Enter the IP address, netmask, and default gateway for `e0M` in their respective places. The IP for node 01 is shown in the following commands; substitute the node 02 IP address as needed.

```
Enter the IP address for port e0M: <<storage_node1_mgmt_ip>>
Enter the netmask for port e0M: <<node_mgmt_mask>>
Enter IP address of default gateway: <<node_mgmt_gateway>>
```

11. Enter the URL where the software can be found.



This web server must be reachable from the storage controller.

```
What is the URL for the package? <<url_boot_software>>
```

12. Press Enter for the user name, indicating no user name.

```
What is the user name on "xxx.xxx.xxx.xxx", if any? Enter
```

13. Enter `y` to set the newly installed software as the default to be used for subsequent reboots.

```
Do you want to set the newly installed software as the default to be used for
Subsequent reboots? {y|n} y
```

14. Enter `y` to reboot the node.

```
The node must be rebooted to start using the newly installed software. Do you
Want to reboot now? {y|n} y
```



When installing new software, the system might perform firmware upgrades to the BIOS and adapter cards, causing reboots and possible stops at the LOADER prompt. If these actions occur, the system might deviate from this procedure.

15. Press Ctrl-C when you see Press Ctrl-C for Boot Menu.

16. Select option 4 for a clean configuration and to initialize all disks.

```
Please choose one of the following:
```

```
(1) Normal Boot.
(2) Boot without /etc/rc.
(3) Change password.
(4) Clean configuration and initialize all disks.
(5) Maintenance mode boot.
(6) Update flash from backup config.
(7) Install new software first.
(8) Reboot node.
Selection (1-8)? 4
```

17. Enter `yes` to zero disks, reset config, and install a new file system.

```
Zero disks, reset config and install a new file system?:yes
```

18. Enter `yes` to erase all of the data on the disks.

```
This will erase all the data on the disks, are you sure?:yes
```



The initialization and creation of the root volume can take up to 8 hours to complete, depending on the number and type of disks attached. After initialization is complete, the storage system reboots. You can continue to configure node 01 while the disks for node 02 are zeroing and vice versa.

Create Cluster on Node 01

In clustered Data ONTAP, the first node in a cluster performs the `cluster create` operation. All other nodes perform a `cluster join` operation. The first node in the cluster is considered node 01. After all of the disks have been zeroed out for the first node, you can see the prompt as below. Use the values from Error! Reference source not found. to complete the configuration of the cluster and each node.

To create a cluster on node 01, complete the following steps:

1. Connect to the storage system console port. The console settings are:

- Baud Rate: 9600
- Data bits: 8
- Parity: none
- Stop bit: 1
- Flow control: none

2. The Cluster Setup wizard starts on the console.

```
Welcome to the cluster setup wizard.

You can enter the following commands at any time:
"help" or "?" - if you want to have a question clarified,
"back" - if you want to change previously answered questions, and
"exit" or "quit" - if you want to quit the cluster setup wizard.
Any changes you made before quitting will be saved.

You can return to cluster setup at any time by typing "cluster setup".
To accept a default or omit a question, do not enter a value.

Do you want to create a new cluster or join an existing cluster? {create, join}:
```



If a login prompt appears instead of the Cluster Setup wizard, you must start the wizard by logging in with the factory default settings and then run the `cluster setup` command.

3. Run the following command to create a new cluster:

```
create
```

4. Enter `no` for the single-node cluster option.

```
Do you intend for this node to be used as a single node cluster? {yes, no} [no]: no
```

5. Enter `no` for the option to use network switches for the cluster network.

```
Will the cluster network be configured to use network switches? [yes]:no
```

6. Activate high availability (HA) and set storage failover.

```
Non-HA mode, Reboot node to activate HA

Do you want to reboot now to set storage failover (SFO) to HA mode? {yes, no}
[yes]: Enter
```

7. After the reboot, enter `admin` in the login prompt.

```
admin
```

8. If the Cluster Setup wizard prompt is displayed again, repeat steps 3 and 4.

9. The system defaults are displayed. Enter `no` for the option to use the system defaults. Follow these prompts to configure the cluster ports:

```
Existing cluster interface configuration found:
```

```

Port      MTU      IP              Netmask
e0a      9000    169.254.204.185 255.255.0.0
e0b      9000    169.254.240.144 255.255.0.0
e0c      9000    169.254.49.216  255.255.0.0
e0d      9000    169.254.241.21  255.255.0.0

```

Do you want to use this configuration? {yes, no} [yes]:no

System Defaults:

Private cluster network ports [e0a,e0c].

Cluster port MTU values will be set to 9000.

Cluster interface IP addresses will be automatically generated.

Do you want to use these defaults? {yes, no} [yes]: no

Step 1 of 5: Create a Cluster

You can type "back", "exit", or "help" at any question.

List the private cluster network ports [e0a,e0b,e0c,e0d]: e0a,e0c

Enter the cluster ports' MTU size [9000]: Enter

Enter the cluster network netmask [255.255.0.0]: Enter

Generating a default IP address. This can take several minutes...

Enter the cluster interface IP address for port e0a [169.254.73.54]: Enter

Generating a default IP address. This can take several minutes...

Enter the cluster interface IP address for port e0c [169.254.64.204]: Enter

10. Use the information in Error! Reference source not found. to create a cluster.

```

Enter the cluster name: <<var_clustername>>

```

```

Enter the cluster base license key: <<var_cluster_base_license_key>>

```

```

Creating cluster <<var_clustername>>

```

```

Enter an additional license key []:<<var_nfs_license>>

```

```

Enter an additional license key []:<<var_iscsi_license>>

```

```

Enter an additional license key []:<<var_flexclone_license>>

```



The cluster-create process can take a minute or two.



While not strictly required for this validated architecture, NetApp recommends that you also install license keys for NetApp SnapRestore® and the SnapManager® suite. These license keys can be added now or at a later time using the CLI or GUI.

```

Enter the cluster administrators (username "admin") password: <<var_password>>

```

```

Retype the password: <<var_password>>

```

```

Enter the cluster management interface port [e0b]: e0M

```

```

Enter the cluster management interface IP address: <<var_clustermgmt_ip>>

```

```

Enter the cluster management interface netmask: <<var_clustermgmt_mask>>

```

```

Enter the cluster management interface default gateway: <<var_clustermgmt_gateway>>

```

11. Enter the DNS domain name.

```

Enter the DNS domain names:<<var_dns_domain_name>>

```

```

Enter the name server IP addresses:<<var_nameserver_ip>>

```



If you have more than one DNS server on your network, separate each one with a comma.

12. Set up the node.

```
Where is the controller located []:<<var_node_location>>
Enter the node management interface port [e0M]: e0M
Enter the node management interface IP address: <<var_node01_mgmt_ip>>

Enter the node management interface netmask:<<var_node01_mgmt_mask>>
Enter the node management interface default gateway:<<var_node01_mgmt_gateway>>
```



The node management interfaces and the cluster management interface should be in different subnets. The node management interfaces can reside on the out-of-band management network, and the cluster management interface can be on the in-band management network.

13. Enter no for the option to enable IPV4 DHCP on the service processor.

```
Enable IPv4 DHCP on the service processor interface [yes]: no
```

14. Set up the service processor.

```
Enter the service processor interface IP address: <<var_node01_sp_ip>>
Enter the service processor interface netmask: <<var_node01_sp_netmask>>
Enter the service processor interface default gateway: <<var_node01_sp_gateway>>
```

15. Press Enter to accept the NetApp AutoSupport™ message.

16. Log in to the cluster interface with the administrator user ID and <<var_password>> as the password.

Join Node 02 to Cluster

The first node in the cluster performs the cluster-create operation. All other nodes perform a cluster-join operation. The first node in the cluster is considered node 01, and the node joining the cluster in this example is node 02. Table 9 lists the cluster network information required for joining node 02 to the existing cluster. You should customize the cluster detail values with the information that is applicable to your deployment.

Table 9 Cluster Details for the Cluster-Join Operation

Cluster Detail	Cluster Detail Value
Cluster node 02 IP address	<<var_node02_mgmt_ip>>
Cluster node 02 netmask	<<var_node02_mgmt_mask>>
Cluster node 02 gateway	<<var_node02_mgmt_gateway>>
Cluster node 02 service processor IP address	<<var_node02_sp_ip>>
Cluster node 02 service processor netmask	<<var_node02_sp_netmask>>
Cluster node 02 service processor gateway	<<var_node02_sp_gateway>>

To join node 02 to the existing cluster, complete the following steps:

1. At the login prompt, enter admin.

```
admin
```

2. The Cluster Setup wizard starts on the console.

```

Welcome to the cluster setup wizard.

You can enter the following commands at any time:
"help" or "?" - if you want to have a question clarified,
"back" - if you want to change previously answered questions, and
"exit" or "quit" - if you want to quit the cluster setup wizard.
    Any changes you made before quitting will be saved.

You can return to cluster setup at any time by typing "cluster setup".
To accept a default or omit a question, do not enter a value.

Do you want to create a new cluster or join an existing cluster? {join}:

```



If a login prompt is displayed instead of the Cluster Setup wizard, you must start the wizard by logging in with the factory default settings and then running the `cluster setup` command.

3. Run the following command to join a cluster:

```
join
```

4. Activate HA and set storage failover.

```

Non-HA mode, Reboot node to activate HA

Warning: Ensure that the HA partner has started disk initialization before
        rebooting this node to enable HA.

Do you want to reboot now to set storage failover (SFO) to HA mode? {yes, no}
[yes]: Enter

```

5. After the reboot, continue the cluster-join process.

6. Data ONTAP detects the existing cluster and agrees to join the same cluster. Follow these prompts to join the cluster:

```

Existing cluster interface configuration found:

Port    MTU    IP              Netmask
e0a     9000   169.254.50.100  255.255.0.0
e0b     9000   169.254.74.132  255.255.0.0
e0c     9000   169.254.147.156 255.255.0.0
e0d     9000   169.254.78.241  255.255.0.0

Do you want to use this configuration? {yes, no} [yes]: no

System Defaults:
Private cluster network ports [e0a,e0b,e0c,e0d].
Cluster port MTU values will be set to 9000.
Cluster interface IP addresses will be automatically generated.

Do you want to use these defaults? {yes, no} [yes]: no

Step 1 of 3: Join an Existing Cluster
You can type "back", "exit", or "help" at any question.

List the private cluster network ports [e0a,e0b,e0c,e0d]: e0a, e0c
Enter the cluster ports' MTU size [9000]: Enter
Enter the cluster network netmask [255.255.0.0]: Enter

Generating a default IP address. This can take several minutes...
Enter the cluster interface IP address for port e0a [169.254.245.255]: Enter

```

```
Generating a default IP address. This can take several minutes...
Enter the cluster interface IP address for port e0c [169.254.49.47]: Enter
```

7. Use the information in Table 9 to join node 02 to the cluster.

```
Enter the name of the cluster you would like to join [<<var_clustername>>]:Enter
```



The node should find the cluster name.



The cluster-join process can take a minute or two.

8. Set up the node.

```
Enter the node management interface port [e0M]: e0M
Enter the node management interface IP address: <<var_node02_mgmt_ip>>
Enter the node management interface netmask: <<var_node02_mgmt_mask >>Enter
Enter the node management interface default gateway: <<var_node02_mgmt_gateway >>Enter
```



The node management interfaces and the cluster management interface should be in different subnets. The node management interfaces can reside on the out-of-band management network, and the cluster management interface can be on the in-band management network.

9. Enter `no` for the option to enable IPv4 DHCP on the service processor.

```
Enable IPv4 DHCP on the service processor interface [yes]: no
```

10. Set up the service processor.

```
Enter the service processor interface IP address: <<var_node01_sp_ip>>
Enter the service processor interface netmask: <<var_node01_sp_netmask>>
Enter the service processor interface default gateway: <<var_node01_sp_gateway>>
```

11. Press Enter to accept the AutoSupport message.

12. Log in to the cluster interface with the admin user ID and `<<var_password>>` as the password.

Configure Initial Cluster Settings

To log in to the cluster, complete the following steps:

1. Open an SSH connection to the cluster IP address or to the host name.
2. Log in as the admin user with the password that you entered earlier.

Assign Disks for Optimal Performance

To achieve optimal performance with SAS drives, the disks in each chassis should be split between the controllers, as opposed to the default allocation method of assigning all disks in a shelf to a single controller. In this solution, assign 12 disks to each controller.

To assign the disks as required, complete the following steps:

1. Verify the current disk allocation.

```
disk show
```

2. Assign disks to the appropriate controller. This reference architecture allocates half of the disks to each controller. However, workload design could dictate different percentages.

```
disk assign -n <<#_of_disks>> -owner <<var_node01>> [-force]
disk assign -n <<#_of_disks>> -owner <<var_node02>> [-force]
```



The `-force` option might be required if the disks are already assigned to another node. Verify that the disk is not a member of an existing aggregate before changing ownership.

Zero All Spare Disks

To zero all spare disks in the cluster, run the following command:

```
disk zerospares
```

Create Aggregates

An aggregate containing the root volume is created during the Data ONTAP setup process. To create additional aggregates, determine the aggregate name, the node on which to create it, and the number of disks that the aggregate contains.

This solution uses one aggregate on each controller, with eight drives per aggregate. To create the aggregates required for this solution, complete the following steps:

1. Run the following commands:

```
aggr create -aggregate aggr01_node01 -node <<var_node01>> -diskcount 8
aggr create -aggregate aggr01_node02 -node <<var_node02>> -diskcount 8
```



Retain at least one disk (select the largest disk) in the configuration as a spare. A best practice is to have at least one spare for each disk type and size per controller.



The aggregate cannot be created until disk zeroing completes. Run the `aggr show` command to display the aggregate creation status. Do not proceed until both `aggr01_node01` and `aggr01_node02` are online.

2. Disable Snapshot[®] copies for the two data aggregates that you created in step 1.

```
system node run -node <<var_node01>> aggr options aggr01_node01 nosnap on
system node run -node <<var_node02>> aggr options aggr01_node02 nosnap on
```

3. Delete any existing Snapshot copies for the two data aggregates.

```
system node run -node <<var_node01>> snap delete -A -a -f aggr01_node01
system node run -node <<var_node02>> snap delete -A -a -f aggr01_node02
```

- Rename the root aggregate on node 01 to match the naming convention for this aggregate on node 02.

```
aggr show
aggr rename -aggregate aggr0 -newname <<var_node01_rootaggrname>>
```

Verify Storage Failover

To confirm that storage failover is enabled, complete the following steps for a failover pair:

- Verify the status of storage failover.

```
storage failover show
```

- Both nodes, <<var_node01>> and <<var_node02>>, must be capable of performing a takeover. If the nodes are capable of performing a takeover, go to step Verify the HA status for the two-node cluster.4.
- Enable failover on one of the two nodes.

```
storage failover modify -node <<var_node01>> -enabled true
```



Enabling failover on one node enables it for both nodes.

- Verify the HA status for the two-node cluster.



This step is not applicable for clusters with more than two nodes.

```
cluster ha show
```

- If HA is configured, go to step 7.
- Enable the HA mode only for the two-node cluster.



Do not run this command for clusters with more than two nodes because doing so causes problems with failover.

```
cluster ha modify -configured true
```

```
Do you want to continue? {y|n}: y
```

- Verify that the hardware-assisted failover feature is correctly configured and, if needed, modify the partner IP address.

```
storage failover hwassist show
storage failover modify -hwassist-partner-ip <<var_node02_mgmt_ip>> -node <<var_node01>>
storage failover modify -hwassist-partner-ip <<var_node01_mgmt_ip>> -node <<var_node02>>
```

Set Onboard UTA2 Ports Personality

- Run the `ucadmin show` command to verify the current mode and current type of the ports.

```
FLEXPOD-OPS-CLUSTER:> ucadmin show
```

Node	Adapter	Current Mode	Current Type	Pending Mode	Pending Type	Admin Status
FLEXPOD-OPS-CLUSTER-01	0e	fc	target	-	-	online
FLEXPOD-OPS-CLUSTER-01	0f	fc	target	-	-	online
FLEXPOD-OPS-CLUSTER-01	0g	fc	target	-	-	online
FLEXPOD-OPS-CLUSTER-01	0h	fc	target	-	-	online
FLEXPOD-OPS-CLUSTER-02	0e	fc	target	-	-	online
FLEXPOD-OPS-CLUSTER-02	0f	fc	target	-	-	online
FLEXPOD-OPS-CLUSTER-02	0g	fc	target	-	-	online
FLEXPOD-OPS-CLUSTER-02	0h	fc	target	-	-	online

8 entries were displayed.

- Verify that the current mode of the ports in use is `cna` and the current type is `target`. If they are, **skip to the next section entitled "Disable Flow Control on 10GbE and UTA2 Ports"**. If not, down the ports:

```
fcport adapter modify -node <<var_node01>> -adapter * -state down
fcport adapter modify -node <<var_node02>> -adapter * -state down
```

- Change the port personality to CNA by running the following commands:

```
ucadmin modify -node <<var_node01>> -adapter * -mode cna -type target
ucadmin modify -node <<var_node02>> -adapter * -mode cna -type target
```



The ports must be offline to run this command.

- Run the `ucadmin show` command to verify that the adapters are pending a change in mode.

```
FLEXPOD-OPS-CLUSTER:> ucadmin show
```

Node	Adapter	Current Mode	Current Type	Pending Mode	Pending Type	Admin Status
FLEXPOD-OPS-CLUSTER-01	0e	fc	target	cna	-	offline
FLEXPOD-OPS-CLUSTER-01	0f	fc	target	cna	-	offline
FLEXPOD-OPS-CLUSTER-01	0g	fc	target	cna	-	offline
FLEXPOD-OPS-CLUSTER-01	0h	fc	target	cna	-	offline
FLEXPOD-OPS-CLUSTER-02	0e	fc	target	cna	-	offline
FLEXPOD-OPS-CLUSTER-02	0f	fc	target	cna	-	offline
FLEXPOD-OPS-CLUSTER-02	0g	fc	target	cna	-	offline
FLEXPOD-OPS-CLUSTER-02	0h	fc	target	cna	-	offline

8 entries were displayed.

- Set the port status to up for all of the system adapters:

```
fcport adapter modify -node <<var_node01>> -adapter * -state up
fcport adapter modify -node <<var_node02>> -adapter * -state up
```

6. Run `system node reboot` to enable the changes.

```
system node reboot -node <<var_node01>>
system node reboot -node <<var_node02>>
```

Disable Flow Control on 10GbE and UTA2 Ports

A NetApp best practice is to disable flow control on all the 10GbE and UTA2 ports that are connected to external devices. To disable flow control, run the following command:

```
network port modify -node * -port e0a..e0h -flowcontrol-admin none
```

```
Warning: Changing the network port settings will cause a several second interruption in carrier.
Do you want to continue? {y|n}: y
```



The `-node` and `-port` parameters in this example take advantage of the range operator available in the clustered Data ONTAP shell. For more information, refer to the section “Methods of Using Query Operators” in the [Clustered Data ONTAP 8.3 System Administration Guide for Cluster Administrators](#).

Create LACP Interface Groups

Clustered Data ONTAP 8.3 includes support for setting up broadcast domains on a group of network ports that belong to the same layer-2 network. A common application for broadcast domains is when a cloud administrator wants to reserve specific ports for use by a certain client or a group of clients.



More information on broadcast domains can be found in the [Clustered Data ONTAP 8.3 Network Management Guide](#).

This type of interface group (ifgrp) requires two or more Ethernet interfaces and a network switch pair that supports the Link Aggregation Control Protocol (LACP). Therefore, confirm that the switches are configured properly.

To create interface groups, complete the following steps.

1. Create a new broadcast domain, which is used to conveniently group the data serving ports to use jumbo frames in the next step:

```
network port broadcast-domain create -mtu 9000 -broadcast-domain Jumbo
```

2. Remove the chosen ports `e0e` and `e0g` from the default broadcast domain:

```
network port broadcast-domain remove-ports -ports
<<var_node01>>:e0e,<<var_node01>>:e0g,<<var_node02>>:e0e,<<var_node02>>:e0g -broadcast-domain Default
```

3. Run the following commands to add ports to the previously created interface group (ifgrp), and add the interface groups to the Jumbo broadcast domain:

```
network port ifgrp create -node <<var_node01>> -ifgrp a0a -distr-func port -mode multimode_lacp
network port ifgrp add-port -node <<var_node01>> -ifgrp a0a -port e0e
network port ifgrp add-port -node <<var_node01>> -ifgrp a0a -port e0g
network port broadcast-domain add-ports -broadcast-domain Jumbo -ports <<var_node01>>:a0a
```

```
network port ifgrp create -node <<var_node02>> -ifgrp a0a -distr-func port -mode multimode_lacp
network port ifgrp add-port -node <<var_node02>> -ifgrp a0a -port e0e
```

```
network port ifgrp add-port -node <<var_node02>> -ifgrp a0a -port e0g
network port broadcast-domain add-ports -broadcast-domain Jumbo -ports <<var_node02>>:a0a
```



The interface group name must follow the standard naming convention of <number><letter>, where <number> is an integer in the range of 0 to 999 without leading zeros and <letter> is a lowercase letter.



Modifications to an interface group cause the underlying physical ports to inherit the same configuration. If the ports are later removed from the interface group, they retain these same settings. However, the inverse is not true; modifying the individual ports does not modify the interface group of which the ports are a member.



After the interface group is added to the broadcast domain, the MTU is set to 9,000 for the group and the individual interfaces. All new VLAN interfaces created on that interface group also has an MTU of 9,000 bytes after they are added to the broadcast domain.

Create VLANs

To create a VLAN for the NFS traffic on both nodes, as well as the VLANs necessary for facilitating UCS compute-node stateless booting through the iSCSI protocol (fabric-a and fabric-b), complete the following steps:

1. Run the following commands:

```
network port vlan create -node <<var_node01>> -vlan-name a0a-<<var_NFS_vlan_id>>
network port vlan create -node <<var_node02>> -vlan-name a0a-<<var_NFS_vlan_id>>
network port vlan create -node <<var_node01>> -vlan-name a0a-<<var_iSCSIA_vlan_id>>
network port vlan create -node <<var_node02>> -vlan-name a0a-<<var_iSCSIB_vlan_id>>
```

2. Add the newly created VLANs to the Jumbo broadcast domain:

```
network port broadcast-domain add-ports -broadcast-domain Jumbo -ports <<var_node01>>:a0a-
<<var_NFS_vlan_id>>,<<var_node02>>:a0a-<<var_NFS_vlan_id>>
network port broadcast-domain add-ports -broadcast-domain Jumbo -ports <<var_node01>>:a0a-
<<var_iSCSIA_vlan_id>>,<<var_node02>>:a0a-<<var_iSCSIA_vlan_id>>
network port broadcast-domain add-ports -broadcast-domain Jumbo -ports <<var_node01>>:a0a-
<<var_iSCSIB_vlan_id>>,<<var_node02>>:a0a-<<var_iSCSIB_vlan_id>>
```

Enable Cisco Discovery Protocol

To enable the Cisco Discovery Protocol (CDP) on the NetApp storage controllers, run the following command:



To be effective, CDP must also be enabled on directly connected networking equipment such as switches and routers.

```
system node run -node * options cdpd.enable on
```



The message displayed after running this command can be safely ignored.

```
You are changing option cdpd.enable, which applies to
both members of the HA configuration in takeover mode.
This value must be the same on both HA members to ensure correct
takeover and giveback operation.
```

Set Auto-Revert on Cluster Management Interface

To set the auto-revert parameter on the cluster management interface, run the following command:

```
network interface modify -vserver <<var_clustername>> -lif cluster_mgmt -auto-revert true
```

Create Failover Group for Cluster Management

To create a failover group for the cluster management port, run the following commands:

```
network interface failover-groups create -failover-group fg-cluster-mgmt -targets
<<var_node01>>:e0M,<<var_node02>>:e0M -vserver <<var_clustername>>
```

Assign Cluster Management Failover Group to Cluster Management LIF

To assign the cluster management failover group to the cluster management logical interface (LIF), run the following command:

```
network interface modify -vserver <<var_clustername>> -lif cluster_mgmt -failover-group fg-cluster-mgmt
```

Configure NTP

To configure time synchronization on the cluster, complete the following steps:

1. Set the time zone.

```
timezone <<var_timezone>>
```



For example, in the eastern United States, the time zone is `America/New_York`.

2. Set the date.

```
date <ccyymmddhhmm.ss>
```



The format for the date is `<[century] [year] [month] [day] [hour] [minute] . [second]>`; for example, `201309081735.17`.

3. Configure the Network Time Protocol (NTP).

```
cluster time-service ntp server create -server <<var_global_ntp_server_ip>>
```

Configure Simple Network Management Protocol

To configure the Simple Network Management Protocol (SNMP), complete the following steps:

1. Configure the SNMP basic information, such as the location and contact. When polled, this information is visible as the `sysLocation` and `sysContact` variables in SNMP.

```
snmp contact <<var_snmp_contact>>
snmp location "<<var_snmp_location>>"
snmp init 1
options snmp.enable on
```

2. Configure SNMP traps to send to remote hosts, such as a OnCommand Unified Manager Core server or another fault management system.

```
snmp traphost add <<var_oncommand_server_fqdn>>
```

Configure SNMPv1 Access

To configure SNMPv1 access, set the shared secret plain-text password, which is called a community.

```
snmp community delete all
snmp community add ro <<var_snmp_community>>
```



Use the `delete all` command with caution. If community strings are used for other monitoring products, the `delete all` command removes them.

Create SNMPv3 User

SNMPv3 requires that a user be defined and configured for authentication. To create and configure a user for SNMPv3, complete the following steps:

1. Create a user called `snmpv3user`.

```
security login create -username snmpv3user -authmethod usm -application snmp
```

2. **Enter the authoritative entity's engine ID and select `md5` as the authentication protocol.**
3. Run the `security snmpusers` command to view the engine ID.
4. When prompted, enter a password for the authentication protocol. The password must have a minimum of eight characters.
5. Select `des` as the privacy protocol.
6. When prompted, enter a password for the privacy protocol. The password must have a minimum of eight characters.

Configure AutoSupport HTTPS

AutoSupport sends support summary information to NetApp through HTTPS. To configure AutoSupport, run the following command:

```
system node autosupport modify -node * -state enable -mail-hosts <<var_mailhost>> -transport https -support enable -to <<var_storage_admin_email>>
```



To enable autosupport to send messages using SMTP, change the `-transport` value in the previous command to `smtp`. When configuring autosupport to use SMTP, be sure to enable mail relay on the mail server for the cluster management and node management IP addresses.

Configure Remote Support Agent

The Remote Support Agent (RSA) is configured directly on the storage controller's remote management device firmware. It can only be installed on systems with an onboard service processor or a remote LAN module. To configure the RSA, complete the following steps:

1. SSH to the first node's service processor.
2. Run the `rsa setup` command.

```
SP <<node01_SP_ip>> rsa setup
The Remote Support Agent improves your case resolution time and
minimizes your manual support overhead.
```

3. Enter `yes` to enable the RSA.

```
Would you like to enable Remote Support Agent? [yes]: yes
Do you use a proxy to connect to the internet? [no]:
```

4. Enter the cluster management IP address of the cluster and enable SSL. The cluster management IP address is picked up automatically.

```
Enter the cluster management IP address of your storage cluster []: <<cluster_ip>>
Do you want to use HTTP with SSL? [yes]: yes
Enter HTTPS port number [443]: Enter
```

5. Enter the credentials for the user with HTTP access on the cluster SVM.

```
Enter HTTP username []: <<http_user_on_cluster_SVM>>
Enter HTTP password: <<http_password_on_cluster_SVM>>
```

6. Commit the changes and make sure that all tests pass validation.

```
Do you want to commit configuration changes entered above? [yes]: yes
Committing configuration changes... done
Remote Support Agent is enabled.
Do you want to test current configuration? [yes]: yes

Testing cluster management LIF HTTP connection ..... ok
Testing Remote Support Enterprise connection ..... ok
All configuration tests passed.
```

7. Repeat steps 2-6 on the other node.

Configure HTTPS Access

Secure access to the storage controller is configured by default with self-signed certificates. If the default values for these certificates must be changed, refer to the instructions in Appendix B: HTTPS Access in Clustered Data ONTAP.

Create Operational Storage Virtual Machine

Two storage virtual machines (SVM; formerly called Vserver) are created for this reference architecture. To create an operational SVM for SAN booting through iSCSI, complete the following steps:



The SVM is referred to as Vserver in the clustered Data ONTAP command-line interface (CLI).

1. Create the SVM.

```
vserver create -vserver FLEXPOD-OPS-SVM1 -rootvolume flexpodops_root -aggregate aggr01_node01 -rootvolume-security-style unix
```



The security style for the SVM becomes the default security style for all volumes created on that SVM. NetApp recommends the UNIX security style for SVMs that primarily support Linux environments. Block access is not affected by security style.

2. Remove protocols from this SVM that are not needed. Because this SVM supports iSCSI booting for only the eventual OpenStack compute nodes, remove all other protocols from the SVM.

```
vserver remove-protocols -vserver FLEXPOD-OPS-SVM1 -protocols cifs,fcg,nfs,ndmp
```

3. Add the two data aggregates to the aggregate list. This allows volumes to be created on these aggregates for this SVM.

```
vserver modify -vserver FLEXPOD-OPS-SVM1 -aggr-list aggr01_node01, aggr01_node02
```

Create Load-Sharing Mirror of SVM Root Volume

To create a load-sharing mirror of an SVM root volume, complete the following steps:

1. Create a volume to be the load-sharing mirror of the root volume of the infrastructure SVM (Vserver) on each node.

```
volume create -vserver FLEXPOD-OPS-SVM1 -volume rootvol_m01 -aggregate aggr01_node01 -size 1GB -type DP
volume create -vserver FLEXPOD-OPS-SVM1 -volume rootvol_m02 -aggregate aggr01_node02 -size 1GB -type DP
```

2. Create a job schedule to update the root-volume mirror relationships every 15 minutes.

```
job schedule interval create -name 15min -minutes 15
```

3. Create the mirroring relationships.

```
snapmirror create -source-path //FLEXPOD-OPS-SVM1/flexpodops_root -destination-path //FLEXPOD-OPS-SVM1/rootvol_m01 -type LS -schedule 15min
snapmirror create -source-path //FLEXPOD-OPS-SVM1/flexpodops_root -destination-path //FLEXPOD-OPS-SVM1/rootvol_m02 -type LS -schedule 15min
```

4. Initialize the mirroring relationship.

```
snapmirror initialize-ls-set -source-path //FLEXPOD-OPS-SVM1/flexpodops_root
```

Create iSCSI Logical Interfaces (LIFs)

To create the network interfaces that are used by the UCS compute nodes for booting, complete the following steps:

1. Create the interfaces on node 01.

```
network interface create -vserver FLEXPOD-OPS-SVM1 -lif iscsi_lif01a -role data -data-protocol iscsi -home-node <<var_node01>> -home-port a0a-<<var_iscsi_a_vlan_id>> -address <<var_node01_iscsi_lif01a_ip>> -netmask <<var_node01_iscsi_lif01a_mask>> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false

network interface create -vserver FLEXPOD-OPS-SVM1 -lif iscsi_lif01b -role data -data-protocol iscsi -home-node <<var_node01>> -home-port a0a-<<var_iscsi_b_vlan_id>> -address <<var_node01_iscsi_lif01b_ip>> -netmask <<var_node01_iscsi_lif01b_mask>> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false
```

2. Create the interface on node 02.

```
network interface create -vserver FLEXPOD-OPS-SVM1 -lif iscsi_lif02a -role data -data-protocol iscsi -home-node <<var_node02>> -home-port a0a-<<var_iscsi_a_vlan_id>> -address <<var_node02_iscsi_lif02a_ip>> -netmask <<var_node02_iscsi_lif02a_mask>> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false

network interface create -vserver FLEXPOD-OPS-SVM1 -lif iscsi_lif02b -role data -data-protocol iscsi -home-node <<var_node02>> -home-port a0a-<<var_iscsi_b_vlan_id>> -address <<var_node02_iscsi_lif02b_ip>> -netmask <<var_node02_iscsi_lif02b_mask>> -status-admin up -failover-policy disabled -firewall-policy data -auto-revert false
```

Create NetApp FlexVol® Volume and Enable Deduplication for Boot LUN Volumes

To create the NetApp FlexVol volume that holds the necessary boot LUNs for each individual RHEL server in this infrastructure, complete the following steps:

1. Create the `rhel_iscsi_boot` FlexVol volume.

```
volume create -vserver FLEXPOD-OPS-SVM1 -volume rhel_iscsi_boot -aggregate aggr01_node01 -size 750GB -state online -policy default -space-guarantee none -percent-snapshot-space 0
```

2. Enable deduplication on the boot LUN volume to enable space savings.

```
volume efficiency on -vserver FLEXPOD-OPS-SVM1 -volume rhel_iscsi_boot
```

3. Update the mirroring relationship.

```
snapmirror update-ls-set -source-path //FLEXPOD-OPS-SVM1/flexpodops_root
```

Create Initiator Groups

Initiator groups (igroups) are tables of Fibre Channel Protocol (FCP) host worldwide port names or iSCSI host-node names. You can define igroups and map them to LUNs to control which initiators have access to LUNs. To do so, run the following commands:

```
igroup create -vserver FLEXPOD-OPS-SVM1 -igroup Openstack_Installer -protocol iscsi -ostype linux -initiator <<var_osp_host0_a_iqn>>,<<var_osp_host0_b_iqn>>
igroup create -vserver FLEXPOD-OPS-SVM1 -igroup OSP_Controller_1 -protocol iscsi -ostype linux -initiator <<var_osp_host1_a_iqn>>,<<var_osp_host1_b_iqn>>
igroup create -vserver FLEXPOD-OPS-SVM1 -igroup OSP_Controller_2 -protocol iscsi -ostype linux -initiator <<var_osp_host2_a_iqn>>,<<var_osp_host2_b_iqn>>
igroup create -vserver FLEXPOD-OPS-SVM1 -igroup OSP_Controller_3 -protocol iscsi -ostype linux -initiator <<var_osp_host3_a_iqn>>,<<var_osp_host3_b_iqn>>
```

```

igroup create -vserver FLEXPOD-OPS-SVM1 -igroup OSP_Compute_1 -protocol iscsi -ostype linux -initiator
<<var_osp_comp1_a_iqn>>,<<var_osp_comp1_b_iqn>>
igroup create -vserver FLEXPOD-OPS-SVM1 -igroup OSP_Compute_2 -protocol iscsi -ostype linux -initiator
<<var_osp_comp2_a_iqn>>,<<var_osp_comp2_b_iqn>>
igroup create -vserver FLEXPOD-OPS-SVM1 -igroup OSP_Compute_3 -protocol iscsi -ostype linux -initiator
<<var_osp_comp3_a_iqn>>,<<var_osp_comp3_b_iqn>>
igroup create -vserver FLEXPOD-OPS-SVM1 -igroup OSP_Compute_4 -protocol iscsi -ostype linux -initiator
<<var_osp_comp4_a_iqn>>,<<var_osp_comp4_b_iqn>>

```

Create LUNs

To create boot LUNs 50GB in size for each host in the infrastructure and to disable space reservation, run the following commands:

```

lun create -vserver FLEXPOD-OPS-SVM1 -volume rhel_iscsi_boot -lun osp_installer_bootlun -size 50GB -ostype
linux -space-reserve disabled
lun create -vserver FLEXPOD-OPS-SVM1 -volume rhel_iscsi_boot -lun osp_controller1_bootlun -size 50GB -ostype
linux -space-reserve disabled
lun create -vserver FLEXPOD-OPS-SVM1 -volume rhel_iscsi_boot -lun osp_controller2_bootlun -size 50GB -ostype
linux -space-reserve disabled
lun create -vserver FLEXPOD-OPS-SVM1 -volume rhel_iscsi_boot -lun osp_controller3_bootlun -size 50GB -ostype
linux -space-reserve disabled
lun create -vserver FLEXPOD-OPS-SVM1 -volume rhel_iscsi_boot -lun osp_compute1_bootlun -size 50GB -ostype
linux -space-reserve disabled
lun create -vserver FLEXPOD-OPS-SVM1 -volume rhel_iscsi_boot -lun osp_compute2_bootlun -size 50GB -ostype
linux -space-reserve disabled
lun create -vserver FLEXPOD-OPS-SVM1 -volume rhel_iscsi_boot -lun osp_compute3_bootlun -size 50GB -ostype
linux -space-reserve disabled
lun create -vserver FLEXPOD-OPS-SVM1 -volume rhel_iscsi_boot -lun osp_compute4_bootlun -size 50GB -ostype
linux -space-reserve disabled

```



It is recommended to have at least a 50Gb LUN for RHEL7.

Map LUNs to Initiator Groups

To map the igroups to the actual boot LUN volumes, run the following commands:

```

lun map -vserver FLEXPOD-OPS-SVM1 -volume rhell7_boot -lun osp_installer_bootlun -igroup OpenStack_Installer
-lun-id 0
lun map -vserver FLEXPOD-OPS-SVM1 -volume rhell7_boot -lun osp_controller1_bootlun -igroup OSP_Controller_1 -
lun-id 0
lun map -vserver FLEXPOD-OPS-SVM1 -volume rhell7_boot -lun osp_controller2_bootlun -igroup OSP_Controller_2 -
lun-id 0
lun map -vserver FLEXPOD-OPS-SVM1 -volume rhell7_boot -lun osp_controller3_bootlun -igroup OSP_Controller_3 -
lun-id 0
lun map -vserver FLEXPOD-OPS-SVM1 -volume rhell7_boot -lun osp_compute1_bootlun -igroup OSP_Compute_1 -lun-id
0
lun map -vserver FLEXPOD-OPS-SVM1 -volume rhell7_boot -lun osp_compute2_bootlun -igroup OSP_Compute_2 -lun-id
0
lun map -vserver FLEXPOD-OPS-SVM1 -volume rhell7_boot -lun osp_compute3_bootlun -igroup OSP_Compute_3 -lun-id
0
lun map -vserver FLEXPOD-OPS-SVM1 -volume rhell7_boot -lun osp_compute4_bootlun -igroup OSP_Compute_4 -lun-id
0

```

Setup for the operational SVM is complete.

Create Infrastructure Storage Virtual Machine

To create an infrastructure SVM specifically for OpenStack and its related services, instances, and configuration, complete the following steps:

1. Create the SVM.

```

vserver create -vserver FLEXPOD-OPENSTACK-SVM -rootvolume openstacksvm_root -aggregate aggr01_node01 -
rootvolume-security-style unix

```

- Remove protocols from the SVM that are not needed for this reference architecture.

```
vserver remove-protocols -vserver FLEXPOD-OPENSTACK-SVM -protocols cifs,fc,iscsi,ndmp
```

- Add the two data aggregates to the aggregate list.

```
vserver modify -vserver FLEXPOD-OPENSTACK-SVM -aggr-list aggr01_node01, aggr01_node02
```

Create Load-Sharing Mirror of SVM Root Volume

To create a load-sharing mirror of an SVM root volume, complete the following steps:

- Create a volume to be the load-sharing mirror of the root volume of the infrastructure SVM on each node.

```
volume create -vserver FLEXPOD-OPENSTACK-SVM -volume rootvol_m01 -aggregate aggr01_node01 -size 1GB -type DP
volume create -vserver FLEXPOD-OPENSTACK-SVM -volume rootvol_m02 -aggregate aggr01_node02 -size 1GB -type DP
```

- Create the mirroring relationships.

```
snapmirror create -source-path //FLEXPOD-OPENSTACK-SVM/openstacksvm_root -destination-path //FLEXPOD-OPENSTACK-SVM/rootvol_m01 -type LS -schedule 15min
snapmirror create -source-path //FLEXPOD-OPENSTACK-SVM/openstacksvm_root -destination-path //FLEXPOD-OPENSTACK-SVM/rootvol_m02 -type LS -schedule 15min
```

- Initialize the mirroring relationship.

```
snapmirror initialize-ls-set -source-path //FLEXPOD-OPENSTACK-SVM/openstacksvm_root
```

Create Failover Group for SVM Management

To create a failover group for data NAS services for OpenStack, run the following command:

```
network interface failover-groups create -failover-group fg-nfs-<<var_nfs_vlan_id>> -vserver openstack-svm -targets <<var_node01>>:a0a-<<var_nfs_vlan_id>>,<<var_node02>>:a0a-<<var_nfs_vlan_id>>
```

Configure NFSv4

To configure NFS on the SVM, complete the following steps:

- Set the NFSv4 ID mapping domain to match the DNS domain name used in the overall environment. This should match the default domain configured in `/etc/idmapd.conf` on each provisioned RHEL7 system, which is the host's DNS domain name by default.

```
nfs server modify -vserver FLEXPOD-OPENSTACK-SVM -v4-id-domain sjc.cisco.com
```

- Create a new export policy for the hosts that need storage access.

```
vserver export-policy create -vserver FLEXPOD-OPENSTACK-SVM -policyname openstack-hosts
```

- Create a new export policy rule for the compute nodes used by this SVM. Set the User ID to which anonymous users are mapped to 0.



For each RHEL host being created, create a rule. Each host has its own rule index. Your first RHEL host has rule index 1, your second RHEL host has rule index 2, and so on. Alternatively, you may specify the entire network in Classless Interdomain Routing (CIDR) notation (for example, 10.23.30.0/24) or use [netgroups](#).

```
vserver export-policy rule create -vserver FLEXPOD-OPENSTACK-SVM -policyname openstack-hosts -ruleindex 1 -
protocol nfs -clientmatch <<rhel_host1_nfs_ip>> -rorule sys -rwrule sys -superuser sys -allow-suid false -
anon 0
```

4. Modify the default UNIX user's group ID for the SVM's root user from 1 (the default) to 0.

```
unix user-modify -vserver FLEXPOD-OPENSTACK-SVM -user root -primary-gid 0
```

5. Assign the new export policy to the infrastructure SVM root volume.

```
volume modify -vserver FLEXPOD-OPENSTACK-SVM -volume openstacksvm_root -policy openstack-hosts
```

Create Flexible Volumes (FlexVol) for Cinder and Glance

To create thick-provisioned FlexVol® **volumes (the volume's name and size and the aggregate on which it exists)** to have storage for both Cinder volumes and Glance images, complete the following steps:

1. Run the following command to create the cinder1 FlexVol volume on aggregate aggr01_node02:



Pay attention to user and group ownership, or you might have problems using the NetApp Copy Offload Tool.

```
volume create -vserver FLEXPOD-OPENSTACK-SVM -volume cinder1 -user 165 -group 165 -aggregate aggr01_node02 -
size 1TB -state online -policy openstack-hosts -junction-path /cinder1 -space-guarantee volume -percent-
snapshot-space 0
```



The following is displayed on the screen and is normal:

```
Notice: Volume cinder1 now has a mount point from volume openstacksvm_root. The load sharing (LS) mirrors of
volume openstacksvm_root will be updated according to the SnapMirror schedule in place for volume
openstacksvm_root. Volume cinder1 will not be visible in the global namespace until the LS mirrors of volume
openstacksvm_root have been updated.
```

2. Create the cinder2 FlexVol on aggregate aggr01_node01:

```
volume create -vserver FLEXPOD-OPENSTACK-SVM -volume cinder2 -user 165 -group 165 -aggregate aggr01_node01 -
size 1TB -state online -policy openstack-hosts -junction-path /cinder2 -space-guarantee volume -percent-
snapshot-space 0
```

3. Create the cinder3 FlexVol on aggregate aggr01_node02:

```
volume create -vserver FLEXPOD-OPENSTACK-SVM -volume cinder3 -user 165 -group 165 -aggregate aggr01_node02 -
size 1TB -state online -policy openstack-hosts -junction-path /cinder3 -space-guarantee volume -percent-
snapshot-space 0
```

4. Run the following command to create the Glance FlexVol volume.



Pay attention to user and group ownership, or you might have problems uploading images to the Glance image store later.

```
volume create -vserver FLEXPOD-OPENSTACK-SVM -volume glance -user 161 -group 161 -aggregate aggr01_node02 -
size 500GB -state online -policy openstack-hosts -junction-path /glance -space-guarantee volume -percent-
snapshot-space 0
```

Enable Deduplication on Glance Volume

To enable deduplication on the Glance image repository volume, run the following command:

```
volume efficiency on -vserver FLEXPOD-OPENSTACK-SVM -volume glance
```



The volume efficiency schedule can be modified per the following documentation [here](#), and can be configured to run off-peak load times.

Create additional FlexVol Volumes for Cinder (Optional)



The following section is optional, but it is a NetApp best practice to have a minimum of **three** FlexVol volumes for Cinder in order to have the Cinder scheduler effectively load-balance between the different FlexVol volumes (referred to as backends from a Cinder perspective).

1. Run the following command to create the archived data FlexVol volume, which is used to illustrate the Storage Service Catalog concept. Note that this volume is thin provisioned, has compression enabled, and has deduplication enabled.

```
volume create -vserver FLEXPOD-OPENSTACK-SVM -volume archived_data -user 165 -group 165 -size 500GB -
aggregate aggr01_node01 -space-guarantee none -policy openstack-hosts -junction-path /archived_data -percent-
snapshot-space 0
```

2. Enable deduplication on the archived_data FlexVol volume.

```
volume efficiency on -vserver FLEXPOD-OPENSTACK-SVM -volume archived_data
```

3. Enable compression on the archived_data FlexVol volume.

```
volume efficiency modify -vserver FLEXPOD-OPENSTACK-SVM -volume archived_data -compression true
```

4. Update the SVM root volume load sharing mirrors. This allows mounts to be accessible by making the new mount points visible to the destination load sharing mirror volumes.

```
snapmirror update-ls-set -source-path FLEXPOD-OPENSTACK-SVM:openstacksvm_root
```

NFS Logical Interfaces

To create NFS logical interfaces (LIFs) for the Cinder and Glance volumes, run the following commands to create LIFs for both nodes:



In this example, the specified failover group is the failover group configured in the section “Create Failover Group for SVM Management”.

```
network interface create -vserver FLEXPOD-OPENSTACK-SVM -lif openstack_nfs_1_lif -role data -data-protocol
nfs -home-node <<var_node01>> -home-port a0a-<<var_nfs_vlan_id>> -address <<nfs_lif_ip1>> -netmask
<<nfs_lif_netmask>> -status-admin up -failover-policy nextavail -firewall-policy data -auto-revert true -
failover-group fg-nfs-<<var_nfs_vlan_id>>
```

```
network interface create -vserver FLEXPOD-OPENSTACK-SVM -lif openstack_nfs_2_lif -role data -data-protocol
nfs -home-node <<var_node02>> -home-port a0a-<<var_nfs_vlan_id>> -address <<nfs_lif_ip2>> -netmask
<<nfs_lif_netmask>> -status-admin up -failover-policy nextavail -firewall-policy data -auto-revert true -
failover-group fg-nfs-<<var_nfs_vlan_id>>
```

Start NFS Server and Enable Advanced NFS Options

To start the NFS service, enable NFS v4.1 with parallel NFS (pNFS) support, and enable the NetApp copy offload capable feature, complete the following steps:

1. Run the following command to enable the NFS service on the FLEXPOD-OPENSTACK-SVM SVM.

```
vserver nfs on -vserver FLEXPOD-OPENSTACK-SVM
```

2. Enable advanced NFS options.

```
vserver nfs modify -vserver FLEXPOD-OPENSTACK-SVM -v4.0 enabled -v4.1 enabled -v4.1-pnfs enabled -vstorage
enabled
```

Gather Target Information from FAS8040

Run the command `iscsi show`, as is shown in Figure 21, to get the target information required for configuration in the Cisco UCS server node section.

Figure 21 iSCSI Target Name

```
FLEXPOD-OPS-CLUSTER::> iscsi show
Vserver      Target      Target      Status
Name         Alias
-----
FLEXPOD-OPS-SVM1
iqn.1992-08.com.netapp:sn.33d3a6d30cdd11e5aecb00a09854db26:vs.3
FLEXPOD-OPS-SVM1      up
```

The target name for future steps is shown in Table 10 .

Table 10 iSCSI Target Name

SVM	Target Name
FLEXPOD_OPS_SVM1	iqn.1992-08.com.netapp:sn.33d3a6d30cdd11e5aecb00a09854db26:vs.3

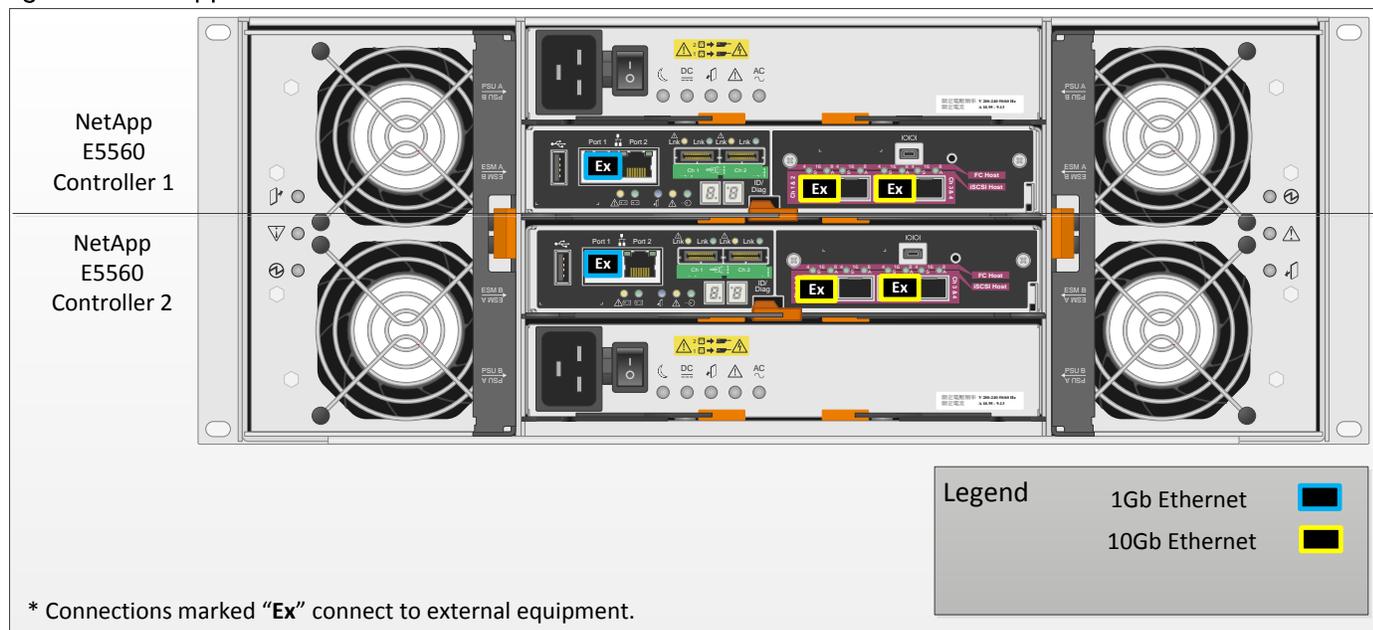
NetApp E-Series Storage Configuration

For instructions on the physical installation of the E-Series E5560, follow the procedures in the [E5500 Series Documentation](#) on the NetApp Support site. When planning the physical location of the storage systems, refer to the following sections in the [Site Preparation Guide](#):

- System Connectivity Requirements
- Circuit Breaker, Power Outlet Balancing, System Cabinet Power Cord Plugs, and Console Pinout Requirements
- Specifications of the DE6600 Drive Tray

Physical connections are represented in Figure 8 below. Note that connections marked Ex represent connections that are external to the E-Series array itself; in other words, these connections run to other pieces of equipment.

Figure 22 NetApp E Series Connections



Disk Shelves

NetApp storage systems support a wide variety of disk shelves and disk drives. Visit the [NetApp Support](#) site to view a complete list of supported [E-Series Disk Shelves](#).

This solution is built on a DE6600 disk shelf with 60 2TB SAS disks owned by two E5500 controllers. These disks provide an ample amount of storage at a moderate price point, and are recommended for an OpenStack deployment, specifically for Object Storage.

Refer to the [Hardware Cabling Guide](#) for information about cabling guidelines with E-Series.

SANtricity OS

This procedure assumes that the storage system has been installed and cabled and is ready for setup. For detailed information about storage system installation, refer to the resources provided previously.

Complete the Configuration Worksheet

Before running the following instructions, review the information contained in the [Software Installation Reference](#) to learn about the information required to configure SANtricity OS. Table 11 lists the information that you need to configure the E-Series E5560. You should customize the values below with information that is applicable to your deployment.

Table 11 SANtricity OS Software Configuration Worksheet

Array Detail	Array Detail Value
E-Series Storage Array name	<<var_storagearrayname>>
Controller A, Port 1 Management Interface	<<var_contA_management_ip>>
Controller B, Port 1 Management Interface	<<var_contB_management_ip>>
Controller A, Port 1 Management netmask	<<var_contA_management_mask>>

Array Detail	Array Detail Value
Controller B, Port 1 Management netmask	<<var_contB_management_mask>>
Array management gateway	<<var_management_gateway>>
iSCSI port Controller A, HIC 1, Port 1	<<var_iscsi_conta_hic1_p1_ip>>
iSCSI port Controller A, HIC 1, Port 1 netmask	<<var_iscsi_conta_hic1_p1_netmask>>
iSCSI port Controller A, HIC 1, Port 3	<<var_iscsi_conta_hic1_p3_ip>>
iSCSI port Controller A, HIC 1, Port 3 netmask	<<var_iscsi_conta_hic1_p3_netmask>>
iSCSI port Controller B, HIC 1, Port 1	<<var_iscsi_contb_hic1_p1_ip>>
iSCSI port Controller B, HIC 1, Port 1 netmask	<<var_iscsi_contb_hic1_p1_netmask>>
iSCSI port Controller B, HIC 1, Port 3	<<var_iscsi_contb_hic1_p3_ip>>
iSCSI port Controller B, HIC 1, Port 3 netmask	<<var_iscsi_contb_hic1_p3_netmask>>
OpenStack Controller-1 IQN A	<<var_controller01_iqn_a>>
OpenStack Controller-1 IQN B	<<var_controller01_iqn_b>>
OpenStack Controller-2 IQN A	<<var_controller02_iqn_a>>
OpenStack Controller-2 IQN B	<<var_controller02_iqn_b>>
OpenStack Controller-3 IQN A	<<var_controller03_iqn_a>>
OpenStack Controller-3 IQN B	<<var_controller03_iqn_b>>

Initial Configuration of Management Interfaces

By default, E-Series systems ship with DHCP enabled on all management network ports. If the controller is unable to obtain a DHCP address on a given network port, it reverts to a default IP address, which may take up to 3 minutes.

1. Wait three minutes after booting the system and then plug a service laptop into controller A, Port 1 with an RJ-45 crossover cable.
2. Configure the service laptop with an IP address of 192.168.128.201 and a subnet mask of 255.255.255.0. Leave the gateway blank.
3. You should be able to ping the system at 192.168.128.101. If not, check your configuration and network cabling before moving onto the next step.
4. Start the SANtricity Storage Manager Client and manually add the storage system to the client by clicking Edit > Add Storage Array, as shown in Figure 23.

Figure 23 Adding E-Series Storage Array

Add New Storage Array - Manual NetApp

[What are in-band and out-of-band management connections?](#)
[Adding controllers with more than one Ethernet port](#)
[What if my system only has one controller?](#)

Select a management method:

Out-of-band management:
 Manage the storage array using the controller Ethernet connections.

Controller (DNS/Network name, IPv4 address, or IPv6 address):

Controller (DNS/Network name, IPv4 address, or IPv6 address):

In-band management:
 Manage the storage array through an attached host.

Host (DNS/Network name, IPv4 address, or IPv6 address):

5. The storage array should be discovered. Ignore any warnings about not adding the other controller.
6. Rename the storage system and give it a descriptive name by right-clicking on it and selecting Re-name. Use <<var_storagearrayname>> as defined in the configuration worksheet.
7. If the storage subsystem is not at least on firmware version 8.20.08.00, refer to the [System Upgrade Guide](#) for instructions on how to upgrade the controller and ESM canister firmware:.
8. Double-click the storage system to launch the Array Management window.
9. Click the Setup tab, and then scroll down to Optional Tasks. Click Configure Ethernet Management Ports.
10. Configure the appropriate values for Controller A, Port 1 and Controller B, Port 1, as is shown in Figure 24. Be sure to disable IPv6 if it does not apply in your environment. Click OK and accept any changes.

Figure 24 Controller Management Ports Configuration

5560_openstack_cvd - Change Network Configuration

NetApp

Ethernet port: Controller A, Port 1

Controller A DNS/Network name: target

Port 1 MAC address: 00:80:e5:29:6d:20

Speed and duplex mode: Auto-negotiate

Enable IPv4

Enable IPv6

IPv4 Settings | IPv6 Settings

IPv4 Configuration:

Obtain configuration automatically from DHCP server

Specify configuration:

IP address: 10 . 23 . 10 . 32

Subnet mask: 255 . 255 . 255 . 0

Controller A gateway:

Change Controller Gateway...

OK Cancel Help

- Unplug the service laptop from the storage system and connect the management ports to the upstream datacenter network. The system should now be accessible through the configured IP addresses input in the previous step and should be accessible by pinging the controller management interfaces.

Disk Pool Creation

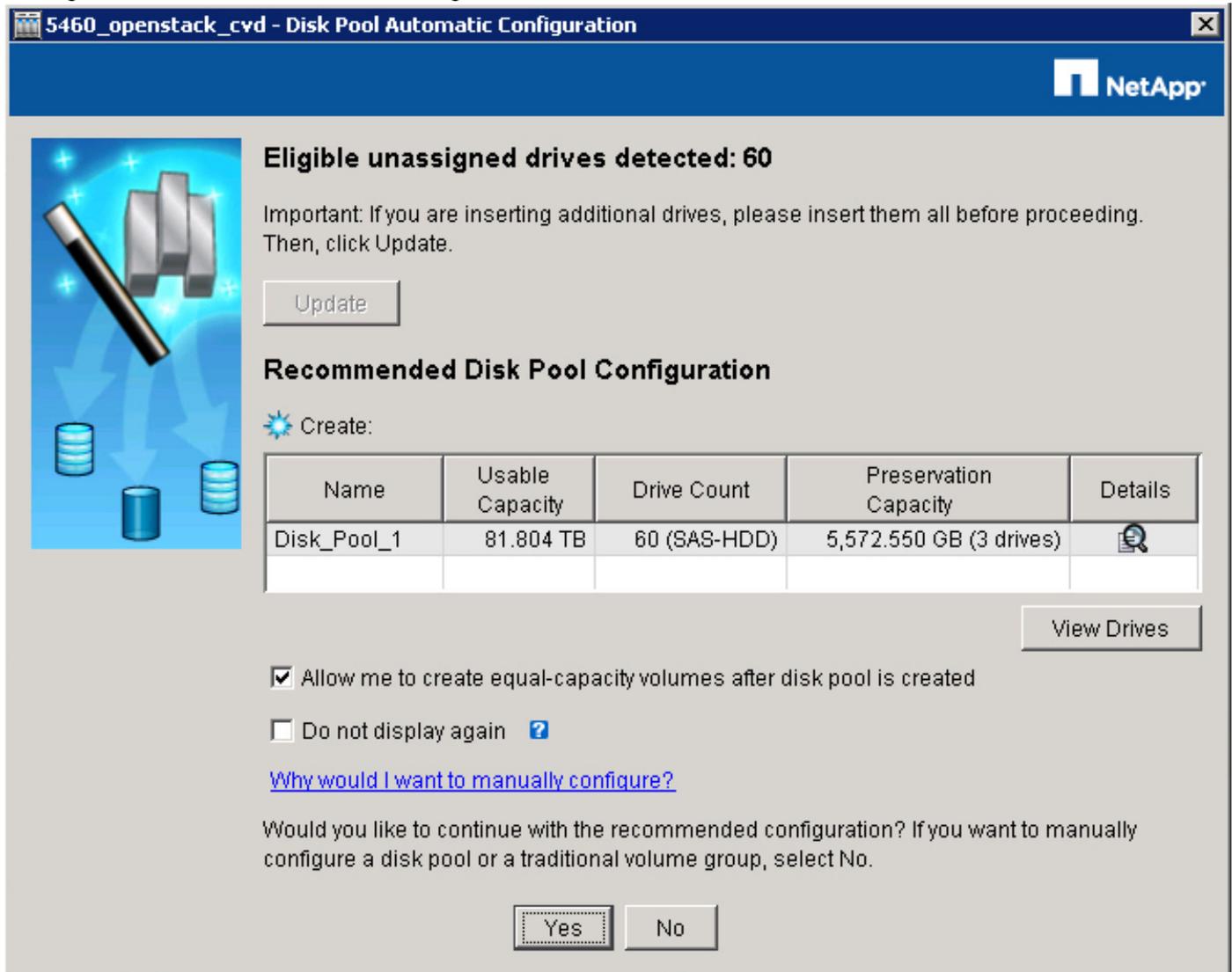
Now that the storage array is accessible on the network, re-launch SANtricity Storage Manager to create disk pools based on the number of hosts connected to the subsystem. In this reference architecture, create pools of 20 drives each, with a total of three disk pools. These three disk pools represent the three OpenStack controller systems that are used as Swift proxy nodes.



A minimum of 11 drives per drive pool is required.

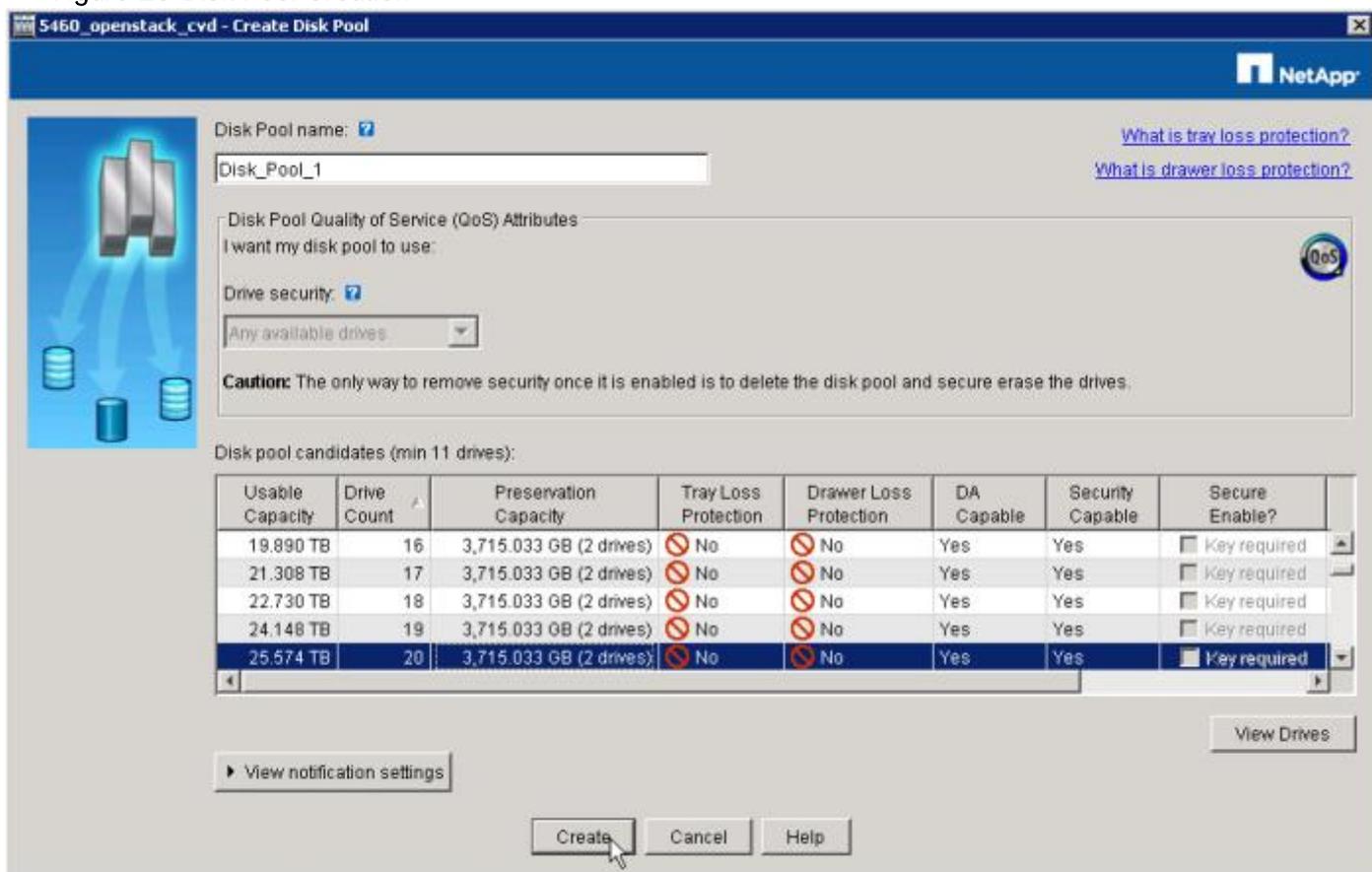
1. Start the SANtricity Storage Manager client and bring up the recently discovered storage array by double clicking on it. The Array Management window appears. Click **No** to manually configure the disk pools as shown in Figure 25.

Figure 25 Disk Pool Automatic Configuration Window



2. Click the Storage and Copy Services tab. Right-click Total Unconfigured Capacity and choose Create Disk Pool.
3. Scroll down and select a Drive Count of 20. Keep the name of `Disk_Pool_1` and click Create as shown in Figure 26.

Figure 26 Disk Pool Creation

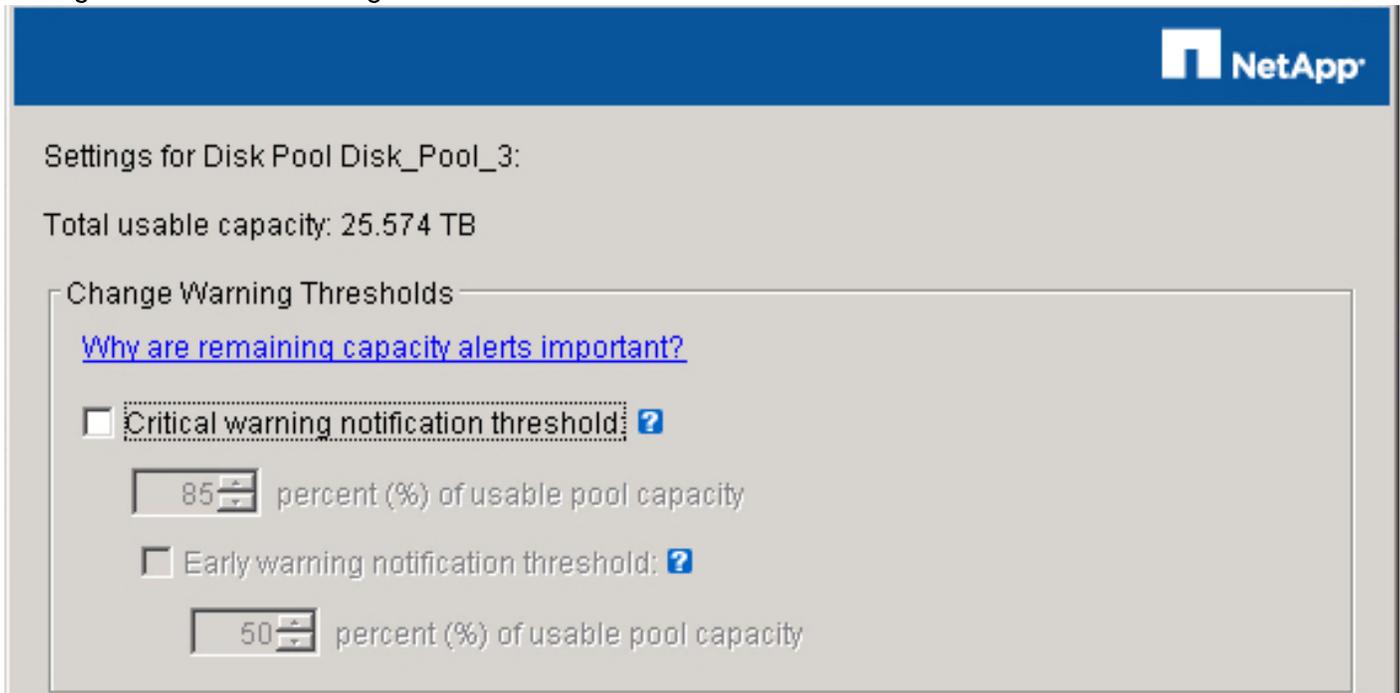


- Repeat the same procedure for `Disk_Pool_2` and `Disk_Pool_3` with 20 disks for each.
- Right-click each new disk pool and select `Change > Settings`. Uncheck `Critical Warning Notification Threshold` and click `OK`, as is shown in Figure 27. This silences warnings that the disk pool is over capacity after volumes are created.



Be sure to do this on all three disk pools as shown in Figure 27. Otherwise, Recovery Guru in SANtricity Storage Manager indicates an error condition.

Figure 27 Critical Warning Threshold Disable



Volume Creation

Volumes can now be created from each of the disk pools that were formed in the previous step. NetApp recommends creating account and container volumes with sizes equal to 2% of the total size of the disk pool. The rest of the space remaining in the disk pool should be allocated to storing objects.

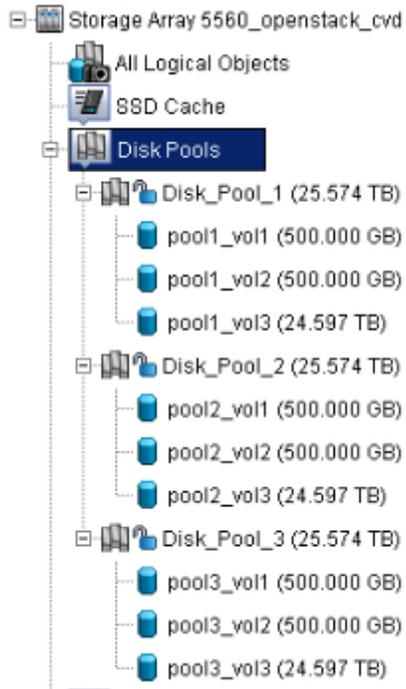
The default mapping for volumes to hosts (through LUN mapping) is to expose all volumes to all hosts. To ensure multiple hosts are not accessing the same LUN concurrently, each volume must be explicitly mapped to the appropriate host it should mount to.



If SSDs are present, be sure to create separate disk pools that only contain SSDs. Swift documentation recommends that SSDs be leveraged for account and container type objects.

1. Right-click the Free Capacity of `Drive_Pool_1` and click Create Volume.
2. Input the size of 500GB and name the volume `pool1_vol1`. Click Finish.
3. Right-click the free capacity of `Drive_Pool_1` and click Create Volume.
4. Input the size 500GB and name the volume `pool1_vol2`. Click Finish.
5. Right-click the free capacity of `Drive_Pool_1` and click Create Volume for a third time.
6. Input the size 24.597TB or the remaining capacity in `Drive_Pool_1`, and name the volume `pool1_vol3`. Click Finish.
7. Repeat steps 1-6, substituting `Drive_Pool_2` and `Drive_Pool_3` for `Drive_Pool_1`. Figure 28 depicts the information that should then be displayed in the navigation portion of the Array Management window.

Figure 28 Disk Pools and Volumes Created

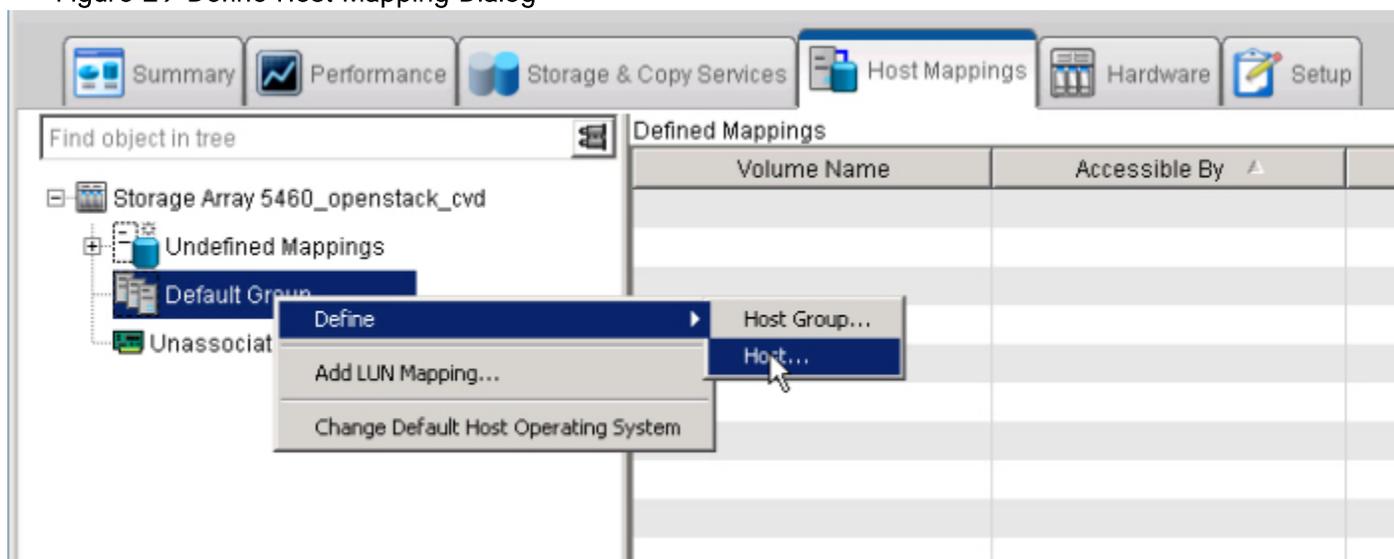


Host Mapping

To define the hosts that connect to the E-Series storage array and their associated host port identifiers, complete the following steps:

1. Click the Host Mappings tab.
2. Select **Don't Display This Dialog Again** in the Mapping Start-Up Help window. Click Close.
3. Right-click the volume named Access and click Remove. This volume represents an in-band management drive and is not needed. Type `yes` in the textbox and click OK.
4. Right-click the Default Group section and select Define > Host, as is shown in Figure 29.

Figure 29 Define Host Mapping Dialog



5. The Define Host wizard is displayed on the screen. Input `Swift_Node1` in the Host Name field.
6. Select No for whether or not to use storage partitions on this storage array.
7. Select Add by Creating a New Host Port Identifier and input `<<var_controller01_iqn_a>>` in the New Host Port Identifier textbox.
8. For Alias/User Label, input `Swift_Node1_A`.
9. Click the Add button to add the host port identifier to the specified host.
10. Repeat step 7, but substitute `<<var_controller01_iqn_b>>` in the host port identifier textbox.
11. For the Alias /User Label, input `Swift_Node1_B`.
12. Click Add. Before clicking Next, verify that the associations are correct for your environment, with an example shown in Figure 30.

Figure 30 Specify Host Port Identifiers

The host communicates with the storage array through its host bus adapters (HBAs) or its iSCSI initiators where each physical port has a unique host port identifier. In this step, select or create an identifier, give it an alias or user label, then add it to the list to be associated with host Swift_Node1.

[How do I match a host port identifier to a host?](#)

Choose a method for adding a host port identifier to a host:

Add by selecting a known unassociated host port identifier

Known unassociated host port identifier:

- There are no known unassociated host port identifiers -

Add by creating a new host port identifier

New host port identifier (max 223 characters):

User Label (30 characters maximum):

Host port identifiers to be associated with the host:

Host Port Identifier	Alias / User Label
iqn.1992-08.com.cisco:fabric-a:49	Swift_Node1_A
iqn.1992-08.com.cisco:fabric-b:49	Swift_Node1_B

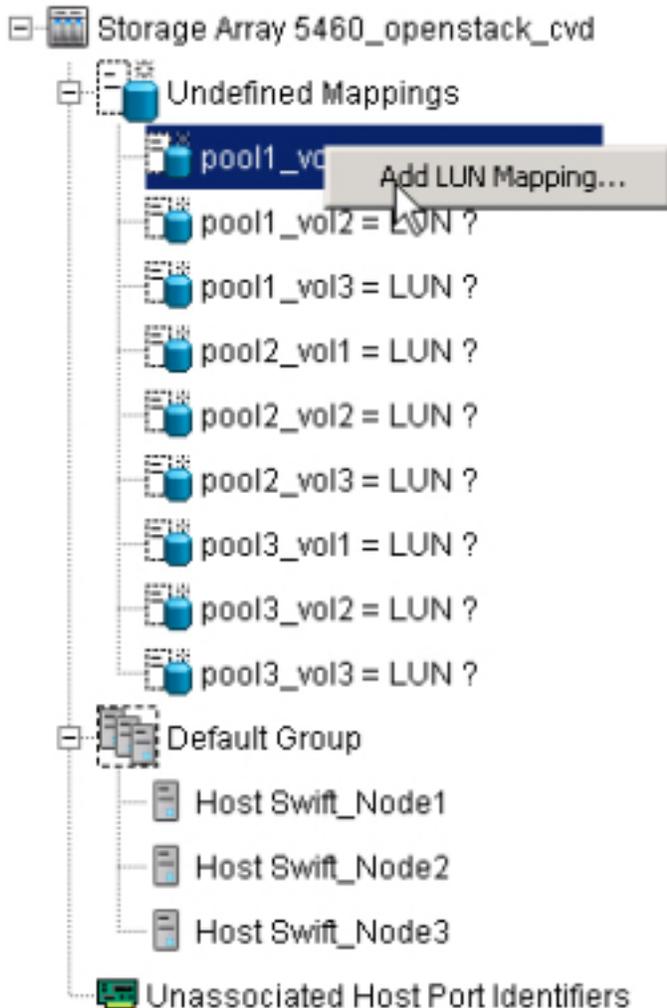
13. In the next step, select host type **Linux (DM-MP)**. This represents a Linux host using the DM-Multipath subsystem. Click Next.
14. Preview the new host to be created, and click Finish after verifying entries.
15. Click Yes when asked to define other hosts.
16. Repeat steps 5-15 for the remaining two hosts to be added to the storage array. Substitute appropriate values as defined in the SANtricity OS configuration worksheet (Table 11).

LUN Mapping

After hosts have been defined and host port identifiers associated, volumes can then be mapped as LUNs to hosts. Volume to LUN mapping gives hosts access to the storage as host LUNs by using the SCSI device driver in Linux in tandem with the device-mapper multipath subsystem for high availability. The LUNs are accessible as SCSI `/dev/mapper/mpathX[1-3]` devices, where x is a unique identifier assigned by the SCSI driver during the device discovery process.

1. While still in the Host Mapping tab, expand Undefined Mappings. All of the volumes created previously should be shown, although with question marks associated with LUN numbers, as they are currently undefined.
2. Right-click `pool11_vol11` and select Add LUN Mapping, as shown in Figure 31.

Figure 31 Add LUN Mapping



3. The Define Additional Mapping window appears. Select `pool11_vol11` and map it to host `Swift_Node1`, LUN 0. Click Add.
4. Select `pool11_vol12` and map it to host `Swift_Node1`, LUN 1. Click Add.
5. Select `pool11_vol13` and map it to host `Swift_Node1`, LUN 2. Click Add.
6. Repeat steps 3 through 5 for `Swift_Node2` and `Swift_Node3`. The resulting configuration should appear similar to Figure 32.

Figure 32 All LUNs Now Mapped in E5560

The screenshot displays the SANtricity Array Management interface for a storage array named '5560_openstack_cvd'. The interface includes a navigation menu with options like Summary, Performance, Storage & Copy Services, Host Mappings, Hardware, and Setup. The 'Host Mappings' tab is active, showing a tree view on the left and a table of defined mappings on the right.

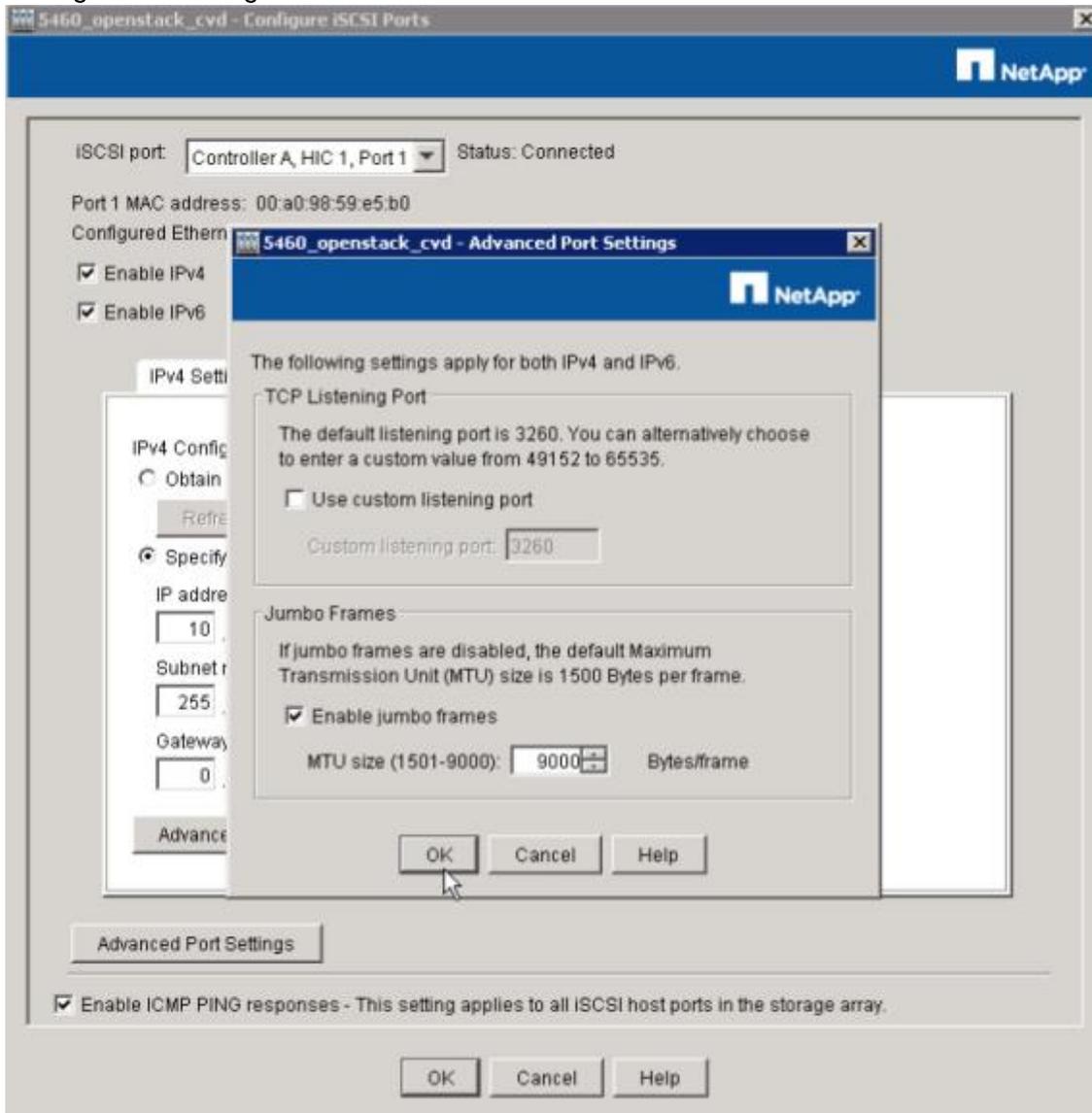
Volume Name	Accessible By	LUN	Volume Capacity	Type
pool1_vol1	Host Swift_Node1	0	500.000 GB	Standard
pool1_vol2	Host Swift_Node1	1	500.000 GB	Standard
pool1_vol3	Host Swift_Node1	2	24.597 TB	Standard
pool2_vol1	Host Swift_Node2	0	500.000 GB	Standard
pool2_vol2	Host Swift_Node2	1	500.000 GB	Standard
pool2_vol3	Host Swift_Node2	2	24.597 TB	Standard
pool3_vol1	Host Swift_Node3	0	500.000 GB	Standard
pool3_vol2	Host Swift_Node3	1	500.000 GB	Standard
pool3_vol3	Host Swift_Node3	2	24.597 TB	Standard

Configure iSCSI Host Ports

Next, configure network parameters for the iSCSI host ports on the controllers, such as IP addressing and MTU. Four ports were used in this reference architecture, two from each Controller spread across Application-specific integrated circuits (ASICs) on the board itself. To configure iSCSI host ports, complete the following steps:

1. Click the Setup tab and then the Configure iSCSI Host Ports option.
2. The Configure iSCSI Ports dialog box appears.
3. For the iSCSI port named Controller A, HIC 1, Port 1, input the pertinent IPv4 configuration as defined in the SANtricity OS configuration worksheet (Table 11).
4. Uncheck the Enable IPv6 checkbox if this does not apply to your environment.
5. Click Advanced Port Settings.
6. Select Enable Jumbo Frames and input 9000 bytes/frame in the Jumbo Frames heading, as shown in Figure 33.

Figure 33 Configure iSCSI Ports and Jumbo Frames



7. Click OK in the Advanced Port Settings window.
8. In the iSCSI port menu, select Controller A, HIC 1, Port 3.
9. Repeat steps 3-7 substituting information from the configuration worksheet for this port.
10. In the iSCSI port dropdown, select Controller B, HIC 1, Port 1.
11. Repeat steps 3-7, substituting information from the configuration worksheet for this port.
12. In the iSCSI port dropdown, select Controller B, HIC 1, Port 3.
13. Repeat steps 3-7, substituting information from the configuration worksheet for this port.
14. Make sure that Enable ICMP PING Responses is enabled. This aids in troubleshooting.

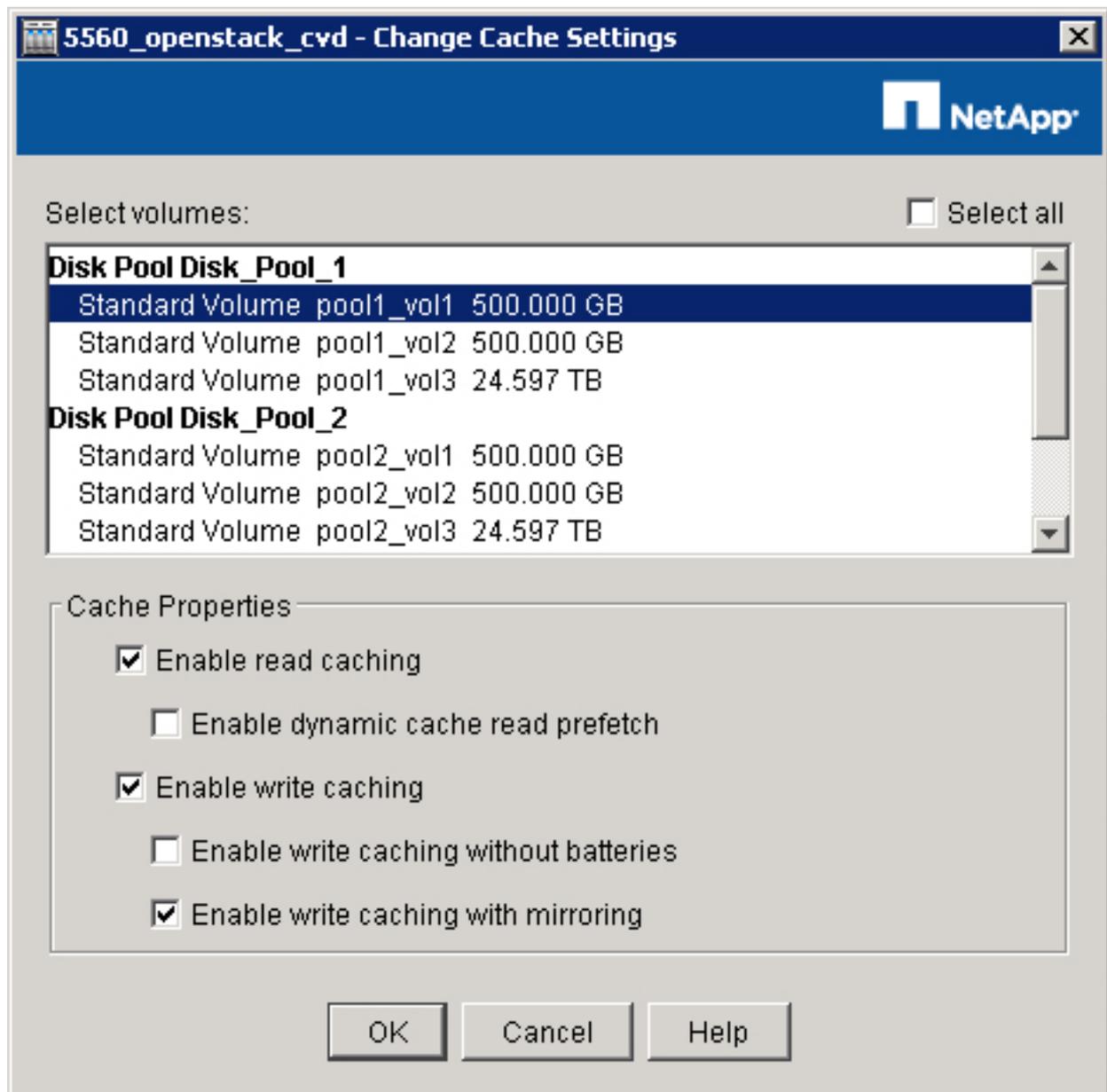
15. Click OK in the Configure iSCSI Ports window. This process may take a few minutes to complete.

Write Cache Mirroring

By default, NetApp E-Series Controllers should have write cache mirroring enabled on a per volume basis by default. To verify this setting, preform the following steps:

1. Click on the Storage & Copy Services tab, and then expand the Disk Pools twisty.
2. Right click on any one of the volumes, and select Change → Cache Settings.
3. A Change Cache Settings dialog box is displayed as shown in Figure 34.

Figure 34 Change Cache Settings Dialog Box



4. Clicking on any of the respective volumes should result in seeing checkboxes for the following Cache Properties:
 - a. Enable read caching
 - b. Enable write caching
 - c. Enable write caching with mirroring
5. Make sure that all three boxes are checked for each respective volume. In this validation, nine total volume exist.



Should a failover scenario exist where only one E-Series Controller is active, write caching with mirroring will adversely affect system performance. Repeat the previous steps and ensure that write caching with mirroring is disabled for each and every volume in the system, should the storage system operate on a single Controller for a prolonged period.



NetApp does not recommend running a storage system on a single Controller. Failed Controllers should be replaced as soon as possible to return the system to a highly available state.

Cisco UCS and Server Configuration

Before running the below instructions, review the configuration worksheet below.

Table 12 Configuration Variable for Cisco UCS

Variable	Description	Implementation Value (Examples)
<<var_ucsa_mgmt_ip>>	Cisco UCS fabric interconnect (FI) A out-of-band management IP address	10.23.10.6
<<var_ucsa_mgmt_mask>>	Out-of-band management network mask	255.255.255.0
<<var_ucsa_mgmt_gateway>>	Out-of-band management network default gateway	10.23.10.1
<<var_ucs_cluster_ip>>	Cisco UCS cluster IP address	10.23.10.5
<<var_ucsb_mgmt_ip>>	Cisco UCS fabric interconnect (FI) B out-of-band management IP address	10.23.10.7
<<var_dns_domain_name>>	DNS Domain name	-
<<var_ucsm_password>>	Administrative password for UCSM	-

Initial Setup of UCS 6248 Fabric Interconnect

This section provides detailed procedures for configuring the Cisco Unified Computing System for use in a FlexPod environment. The steps are necessary to provision the Cisco UCS C-Series and B-Series servers and should be followed precisely to avoid improper configuration.

Cisco UCS 6248 A

To configure the Cisco UCS chassis A, complete the following steps:

1. Connect to the console port on the first Cisco UCS 6248 fabric interconnect.

Enter the configuration method: console

Enter the setup mode; setup newly or restore from backup. (setup/restore)? setup

You have choosen to setup a a new fabric interconnect? Continue? (y/n): y

Enforce strong passwords? (y/n) [y]: y

Enter the password for "admin": <<var_ucsm_password>>

Enter the same password for "admin": <<var_ucsm_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: <<var_ucs_clustername>>

Physical switch Mgmt0 IPv4 address: <<var_ucsa_mgmt_ip>>

Physical switch Mgmt0 IPv4 netmask: <<var_ucsa_mgmt_mask>>

IPv4 address of the default gateway: <<var_ucsa_mgmt_gateway>>

Cluster IPv4 address: <<var_ucs_cluster_ip>>

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]: Enter

2. Review the settings printed to the console. If they are correct, answer yes to apply and save the configuration

3. Wait for the login prompt to make sure that the configuration has been saved.

Cisco UCS 6148 B

To configure the Cisco UCS chassis A, complete the following steps:

1. Connect to the console port on the second Cisco UCS 6248 fabric interconnect

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: <<var_ucsm_password>>

Physical switch Mgmt0 IPv4 address: <<var_ucsb_mgmt_ip>>

```
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no):
y
```

2. Wait for the login prompt to make sure that the configuration has been saved.

Cisco UCS Configuration

When the initial configuration on both FIs is completed, Cisco UCS manager will be accessible through web browser (<http://<<ucsm-virtual-ip>>>) or SSH. Connect to Cisco UCS Manager using SSH and verify HA status (Figure 35).

Figure 35 Verify HA Status

```
UCS-FLEXPOD-FAB-A# show cluster state
Cluster Id: 0x3bbf9944066711e5-0xa8888c604f640804

A: UP, PRIMARY
B: UP, SUBORDINATE

HA READY
UCS-FLEXPOD-FAB-A#
```

Cisco UCS Manager

When the fabric pair is up and cluster state is verified, launch the Cisco UCS Manager using the following steps.

Login to Cisco UCS Manager

To log in to the Cisco Unified Computing System environment, complete the following steps:

1. Open a web browser and navigate to the Cisco UCS 6248 Fabric Interconnect cluster address.
2. Click the Launch UCS Manager link to download the Cisco UCS Manager software.
3. If prompted to accept security certificates, accept as necessary.
4. When prompted, enter `admin` as the user name and enter the administrative password.
5. Click Login to log in to Cisco UCS Manager.



Be sure to configure any authentication, NTP, syslog, or other management services your organization requires while on this configuration step.

Edit Chassis Discovery Policy

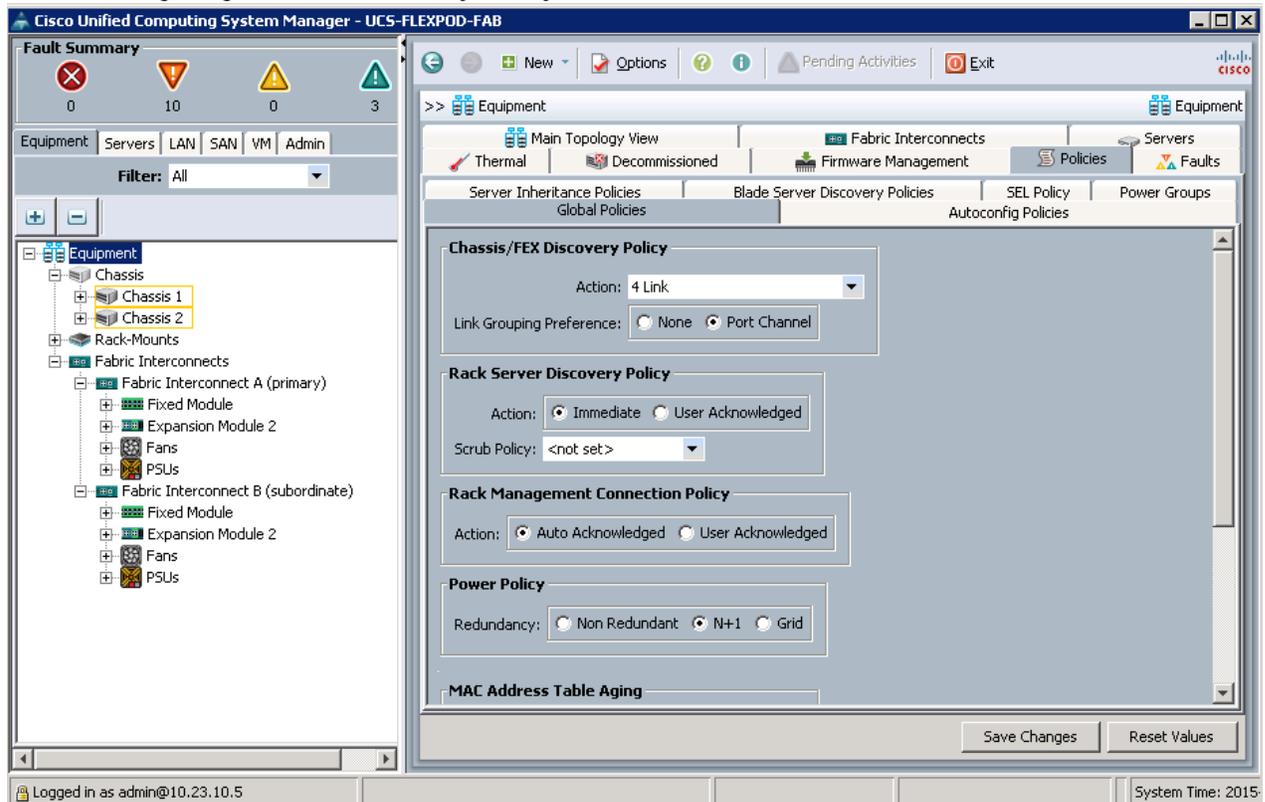
Setting the discovery policy simplifies the addition of Cisco UCS B-Series chassis.

To modify the chassis discovery policy, complete the following steps:

1. In Cisco UCS Manager, click the Equipment tab in the navigation pane and select Equipment in the list on the left.
2. In the right pane, click the Policies tab.

3. Under Global Policies, set the Chassis/FEX Discovery Policy to match the number of uplink ports that are cabled between the chassis or fabric extenders (FEXes) and the fabric interconnects.
4. Set the Link Grouping Preference to Port Channel for better bandwidth utilization and link level high availability as shown in Figure 36.
5. Click Save Changes.
6. Click OK.

Figure 36 Configuring Chassis Discovery Policy

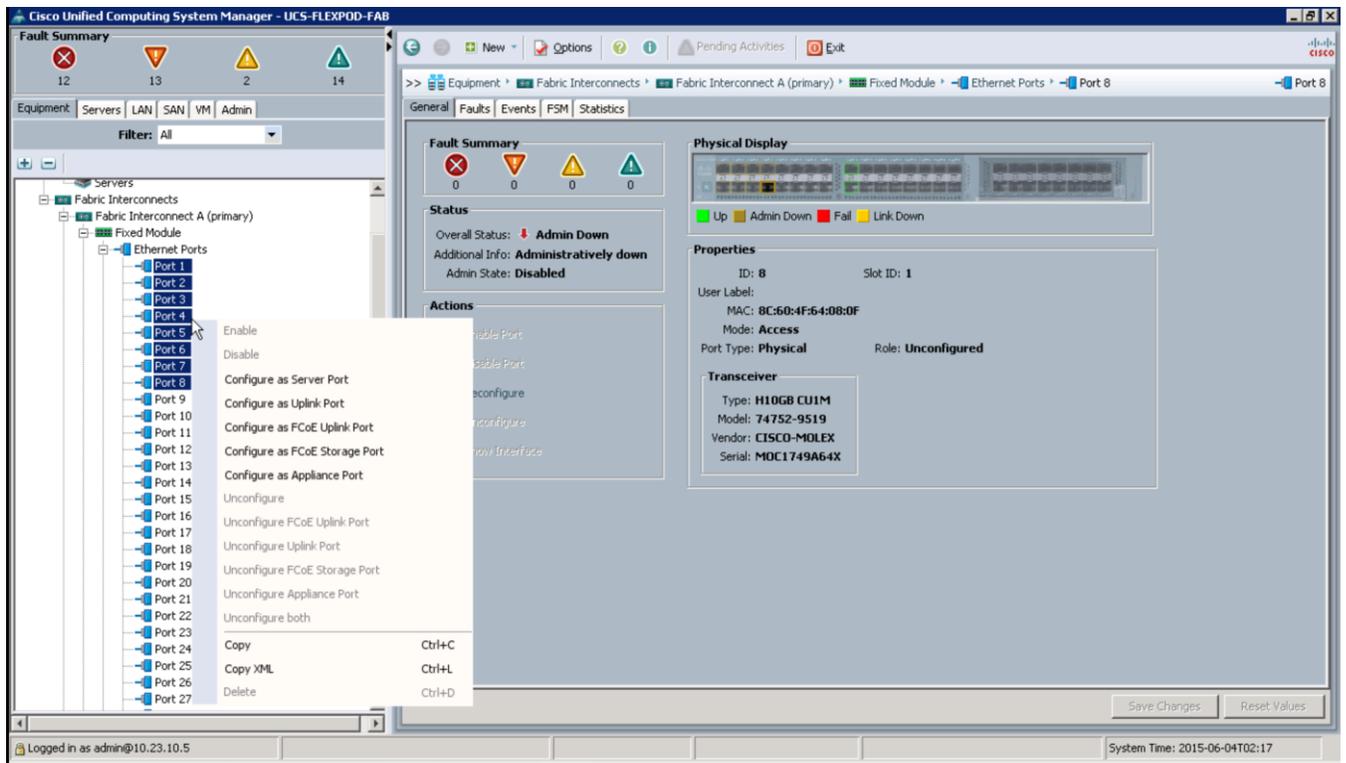


Enable Server and Uplink Ports

To enable server and uplink ports, complete the following steps:

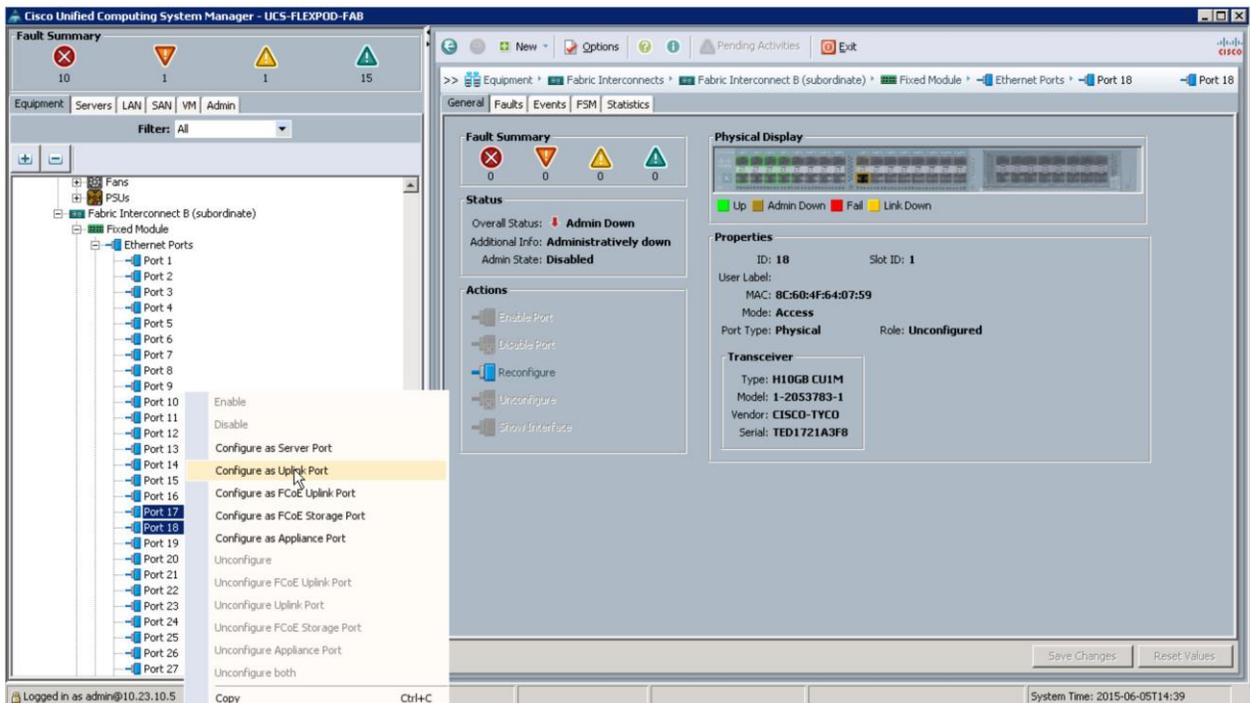
1. In Cisco UCS Manager, click the Equipment tab in the navigation pane.
2. Select Equipment > Fabric Interconnects > Fabric Interconnect A (primary) > Fixed Module.
3. Expand Ethernet Ports.
4. Select the ports (Port 1 to 8) that are connected to the 2204 FEX-A of each chassis, right-click them, and select Configure as Server Port (Figure 37).
5. Click Yes to confirm server ports and click OK.

Figure 37 Port Configuration



6. Select ports 17 and 18 that are connected to the Cisco Nexus switches, right-click them, and select Configure as Uplink Port (Figure 38).

Figure 38 Configure Uplink Ports



7. Click Yes to confirm uplink ports and click OK.
8. Select Equipment > Fabric Interconnects > Fabric Interconnect B (subordinate) > Fixed Module.
9. Expand Ethernet Ports.
10. Select the ports (Port 1 to 8) that are connected to the 2204 FEX-B of each chassis, right-click them, and select Configure as Server Port.
11. Click Yes to confirm server ports and click OK.
12. Select ports 17 and 18 that are connected to the Cisco Nexus switches, right-click them, and select Configure as Uplink Port.
13. Click Yes to confirm the uplink ports and click OK.
14. Select Equipment > Fabric Interconnects > Fabric Interconnect A (primary) > Fixed Module. Select Ethernet Ports tab in the right hand pane. Verify (Figure 39) that the ports connected to chassis and/or to the Cisco FEX 2204 are now configured as server ports (Port 1 to 8) and uplink ports (Port 17 and 18). Verify the same for Fabric Interconnect B (subordinate).

Figure 39 Verify Server and Uplink Ports

The screenshot displays the Cisco Unified Computing System Manager interface. The left pane shows the navigation tree with 'Equipment' > 'Fabric Interconnects' > 'Fabric Interconnect A (primary)' > 'Fixed Module' selected. The right pane shows the 'Ethernet Ports' tab with a table of port configurations.

Slot	Port ID	MAC	If Role	If Type	Overall Status	Administrative State
1	1	8C:60:4F:64...	Server	Physical	Up	Enabled
1	2	8C:60:4F:64...	Server	Physical	Up	Enabled
1	3	8C:60:4F:64...	Server	Physical	Up	Enabled
1	4	8C:60:4F:64...	Server	Physical	Up	Enabled
1	5	8C:60:4F:64...	Server	Physical	Up	Enabled
1	6	8C:60:4F:64...	Server	Physical	Up	Enabled
1	7	8C:60:4F:64...	Server	Physical	Up	Enabled
1	8	8C:60:4F:64...	Server	Physical	Up	Enabled
1	9	8C:60:4F:64...	Unconfigured	Physical	Sfp Not P...	Disabled
1	10	8C:60:4F:64...	Unconfigured	Physical	Sfp Not P...	Disabled
1	11	8C:60:4F:64...	Unconfigured	Physical	Sfp Not P...	Disabled
1	12	8C:60:4F:64...	Unconfigured	Physical	Sfp Not P...	Disabled
1	13	8C:60:4F:64...	Unconfigured	Physical	Sfp Not P...	Disabled
1	14	8C:60:4F:64...	Unconfigured	Physical	Sfp Not P...	Disabled
1	15	8C:60:4F:64...	Unconfigured	Physical	Sfp Not P...	Disabled
1	16	8C:60:4F:64...	Unconfigured	Physical	Sfp Not P...	Disabled
1	17	8C:60:4F:64...	Network	Physical	Up	Enabled
1	18	8C:60:4F:64...	Network	Physical	Up	Enabled
1	19	8C:60:4F:64...	Unconfigured	Physical	Sfp Not P...	Disabled
1	20	8C:60:4F:64...	Unconfigured	Physical	Sfp Not P...	Disabled

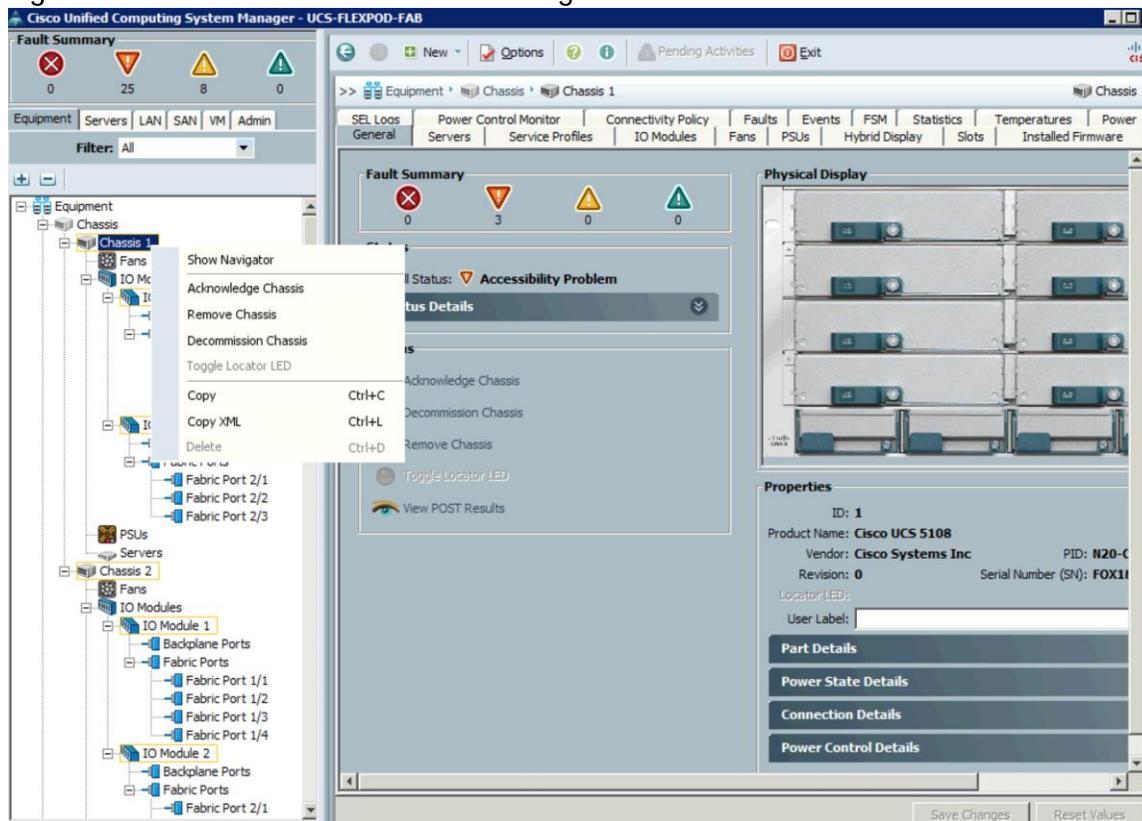
The interface also shows a 'Fault Summary' at the top left with 0 critical, 10 warning, 0 error, and 3 info faults. The bottom status bar indicates the user is logged in as admin@10.23.10.5 and the system time is 2015.

Acknowledge Cisco UCS Chassis

To acknowledge all Cisco UCS chassis, complete the following steps:

1. In Cisco UCS Manager, click the Equipment tab in the navigation pane.
2. Expand Chassis and select each chassis that is listed.
3. Right-click each chassis and select Acknowledge Chassis (Figure 40)
4. Click Yes and then click OK to complete acknowledging the chassis.

Figure 40 Cisco UCS – Chassis Acknowledgement



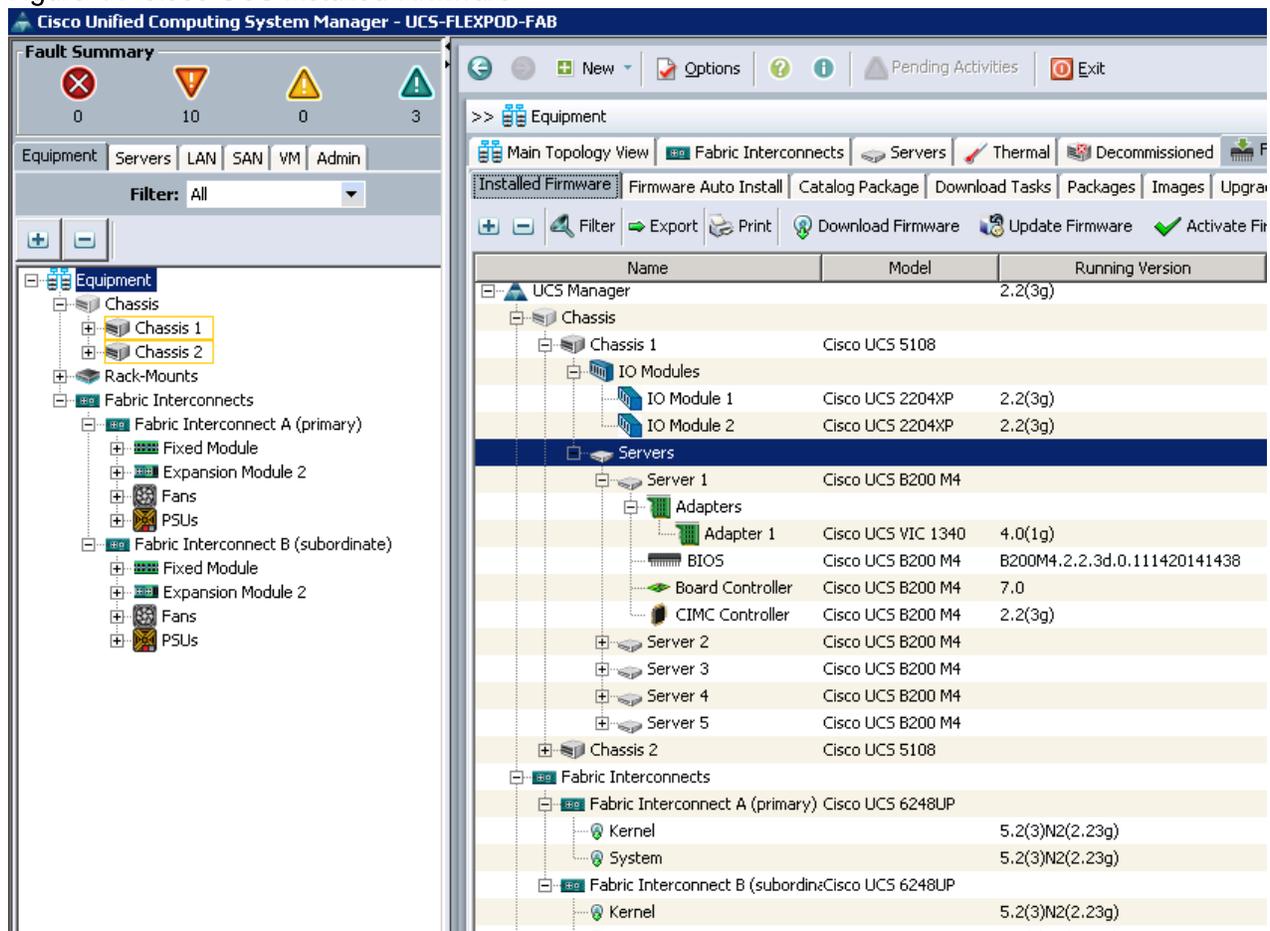
Upgrade Cisco UCS Version to 2.2(3g)

It is required to upgrade the Cisco UCS Manager software, infrastructure firmware, and server firmware to version 2.2(3g). Upgrading Cisco UCS software is beyond the scope of this document. To upgrade the Cisco UCS Manager software, infrastructure firmware, and the server firmware to version 2.2(3g), please refer to [Cisco UCS Manager Install and Upgrade Guides](#).

After the upgrade, to verify the version, complete the following steps:

1. In Cisco UCS Manager, click the Equipment tab in the navigation pane.
2. Select Installed Firmware tab in the right hand pane.
3. Verify version for UCS Manager, IO Modules, Servers, and Fabric Interconnect (Figure 41)

Figure 41 Cisco UCS Installed Firmware



Synchronize Cisco UCS to NTP

To synchronize the Cisco UCS environment to the NTP server, complete the following steps:

1. In Cisco UCS Manager, click the Admin tab in the navigation pane.
2. Select All > Timezone Management.
3. In the Properties pane, select the appropriate time zone in the Timezone menu.
4. Click Save Changes, and then click OK.
5. Click Add NTP Server.
6. Enter <<var_global_ntp_server_ip>> and click OK.
7. Click OK.

Create Uplink Port Channels to Cisco Nexus Switch

To configure the necessary port channels out of the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.



In this procedure, two port channels are created: one from fabric A to both Cisco Nexus switches and one from fabric B to both Cisco Nexus switches.

2. Under LAN > LAN Cloud, expand the Fabric A tree.
3. Right-click Port Channels.
4. Select Create Port Channel (Figure 42).
5. Enter 17 as the unique ID of the port channel.
6. Enter vPC-17-Nexus9k as the name of the port channel.
7. Click Next.

Figure 42 Create Uplink Port Channel

Unified Computing System Manager

Create Port Channel

1. Set Port Channel Name
2. Add Ports

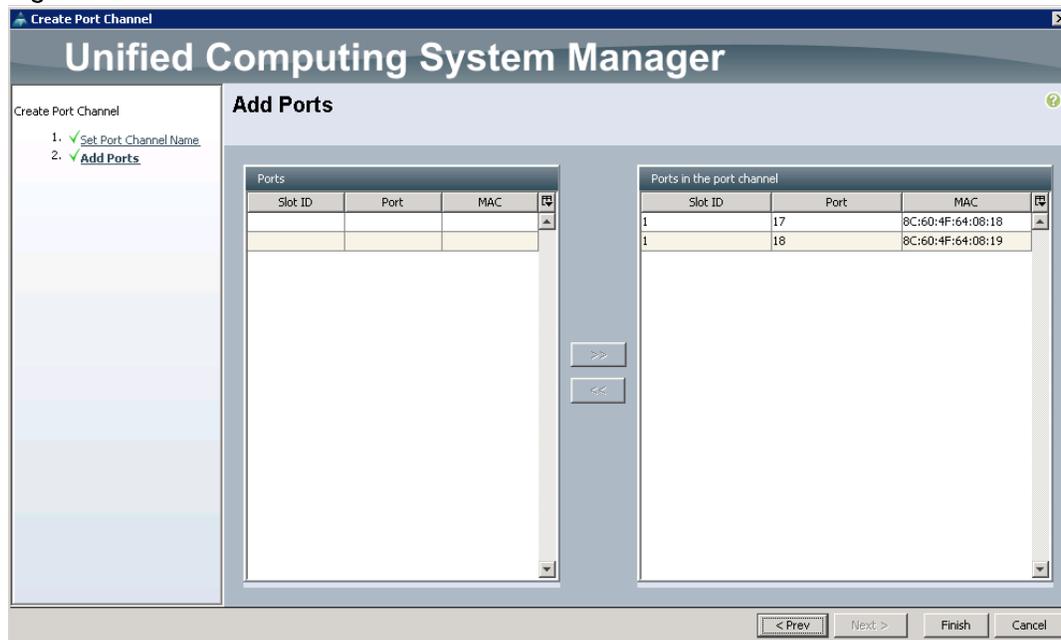
ID: 17

Name: vPC-17-Nexus9k

< Prev Next > Finish Cancel

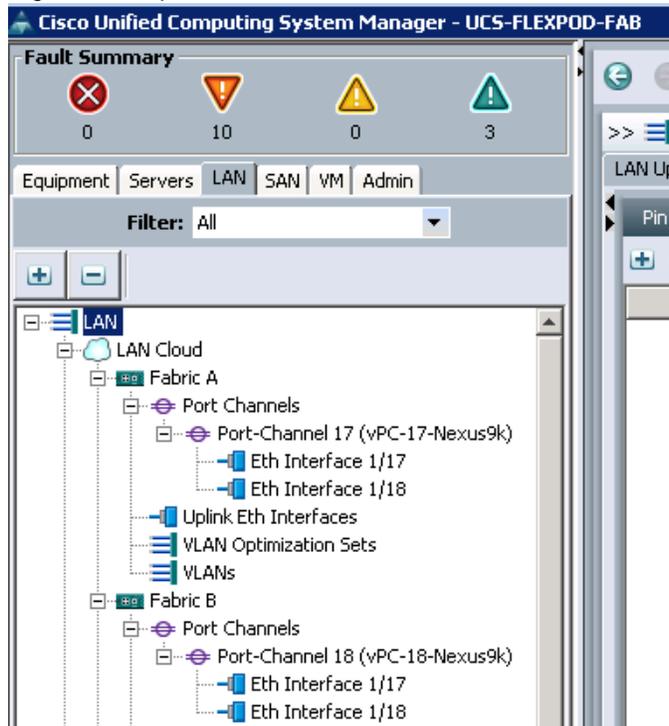
8. Select the following ports to be added to the port channel:
 - a. Slot ID 1 and port 17
 - b. Slot ID 1 and port 18
9. Click >> to add the ports to the port channel.
10. Click Finish to create the port channel (Figure 43).

Figure 43 Add Ports to the Port Channel



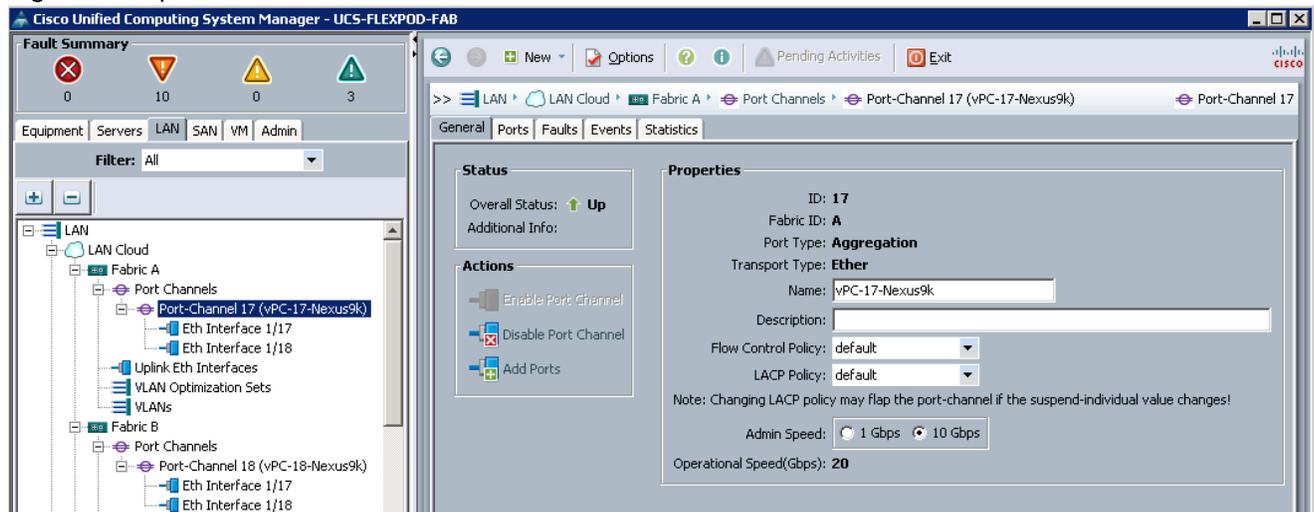
11. Click OK.
12. In the navigation pane, under LAN > LAN Cloud, expand the fabric B tree.
13. Right-click Port Channels.
14. Select Create Port Channel.
15. Enter 18 as the unique ID of the port channel.
16. Enter vPC-18-Nexus9k as the name of the port channel.
17. Click Next.
18. Select the following ports to be added to the port channel:
 - a. Slot ID 1 and port 17
 - b. Slot ID 1 and port 18
19. Click >> to add the ports to the port channel.
20. Click Finish to create the port channel.
21. Click OK.
22. In the navigation pane, under LAN > LAN Cloud, expand Port Channels for both Fabric A and Fabric B, verify the configuration as shown in Figure 44.

Figure 44 Uplink Port Channels in Fabric A and Fabric B



23. Wait until the overall status of the Port Channel is up. Verify port channel status to “Up” (Figure 45) under LAN > LAN Cloud>Fabric A>Port Channels. Click Port-Channel 17 (vPC-17-Nexus9k). Perform the same by clicking Port-Channel 18 (vPC-18-Nexus9k) under LAN > LAN Cloud>Fabric B>Port Channels

Figure 45 Uplink Port Channels Status



Add Block of IP Address for KVM Access

To create a block of IP addresses for server Keyboard, Video, and Mouse (KVM) access in the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation tab.

2. Select Pools > root > IP Pools > IP Pool ext-mgmt
3. In the action pane, select Create Block of IPv4 Addresses.
4. Enter the starting IP address of the block and the number of IP addresses required, subnet, and gateway information.
5. Click OK to create the IP block
6. Click OK in the confirmation message (Figure 46).

Figure 46 Block of IPv4 Addresses for KVM Access

The screenshot shows a dialog box titled "Create a Block of IPv4 Addresses". The dialog contains the following fields and values:

- From:** 10.23.10.11
- Size:** 16
- Subnet Mask:** 255.255.255.0
- Default Gateway:** 10.23.10.1
- Primary DNS:** 0.0.0.0
- Secondary DNS:** 0.0.0.0

At the bottom right of the dialog are "OK" and "Cancel" buttons.

Create MAC Address Pools

To configure the necessary MAC address pools for the Cisco UCS environment, complete the following steps:

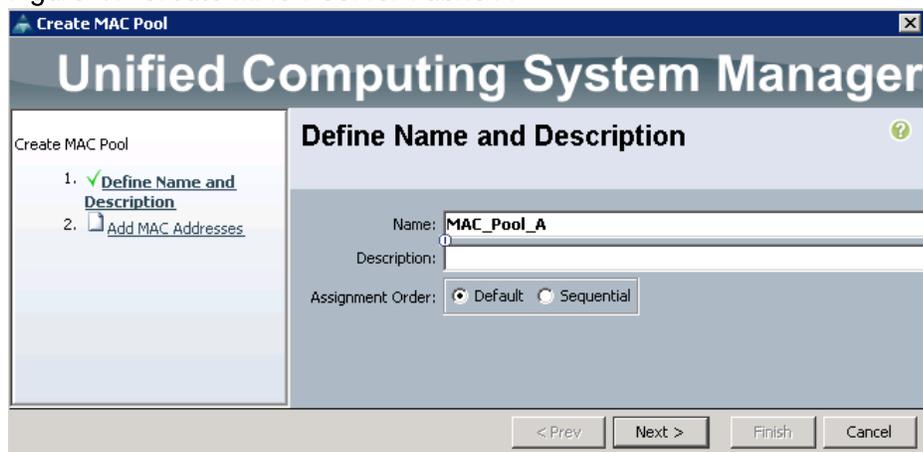
1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Pools > root.



In this procedure, two MAC address pools are created, one for fabric A and one for fabric B

3. Right-click MAC Pools under the root organization.
4. Select Create MAC Pool to create the MAC address pool
5. Enter MAC_Pool_A as the name of the MAC pool.
6. Optional: Enter a description for the MAC pool.
7. Click Next (Figure 47).

Figure 47 Create MAC Pool for Fabric A



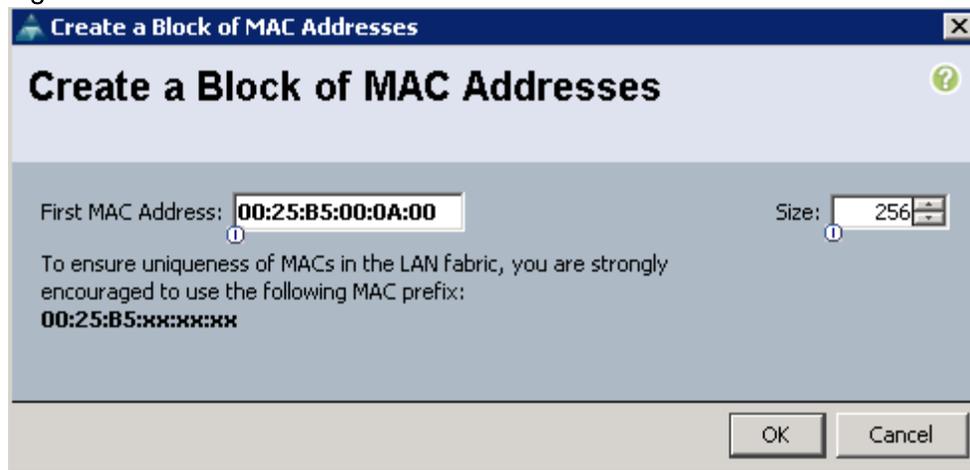
8. Click Add.
9. Specify a starting MAC address.



For this FlexPod solution, it is recommended to place 0A in the next-to-last octet of the starting MAC address to identify all of the MAC addresses that belongs to Fabric A

10. Specify a size for the MAC address pool that is sufficient to support the available blade or server resources (Figure 47).

Figure 48 Block of MAC Addresses for Fabric A



11. Click OK.
12. Click Finish.
13. In the confirmation message, click OK.
14. Right-click MAC Pools under the root organization.
15. Select Create MAC Pool to create the MAC address pool.
16. Enter MAC_Pool_B as the name of the MAC pool (Figure 49).

Figure 49 Create MAC Pool for Fabric B

17. Optional: Enter a description for the MAC pool.

18. Click Next.

19. Click Add.

20. Specify a starting MAC address.



For this FlexPod solution, it is recommended to place 0B in the next-to-last octet of the starting MAC address to identify all of the MAC addresses that belongs to Fabric B

21. Specify a size for the MAC address pool that is sufficient to support the available blade or server resources (Figure 50).

Figure 50 Block of MAC Addresses for Fabric B

22. Click OK.

23. Click Finish.

24. In the confirmation message, click OK.

Create IQN Pools for iSCSI Boot

To configure the necessary IQN pools for the Cisco UCS environment, complete the following steps:

1. Select the SAN tab on the left.
2. Select Pools > root.
3. Right-click IQN Pools under the root organization.
4. Select Create IQN Suffix Pool to create the IQN pool.
5. Enter IQN_Pool_A for the name of the IQN pool.
6. Optional: Enter a description for the IQN pool.
7. Select Sequential for Assignment Order.
8. Enter iqn.1992-08.com.cisco as the prefix (Figure 51).

Figure 51 IQN Pool for Fabric A

9. Click Next.
10. Click Add
11. Enter fabric-a as the suffix
12. Enter 1 in the From field
13. Specify the size of IQN block sufficient to support the available server resources (Figure 52).
14. Click OK.
15. Click Finish
16. In the message box that displays, click OK.

Figure 52 Block of IQN Suffixes for Fabric A

Dialog box titled "Create a Block of IQN Suffixes".

Suffix:

From:

Size:

Buttons: OK, Cancel

17. Right-click IQN Pools under the root organization.
18. Select Create IQN Suffix Pool to create the IQN pool.
19. Enter IQN_Pool_B for the name of the IQN pool.
20. Optional: Enter a description for the IQN pool.
21. Enter iqn.1992-08.com.cisco as the prefix (Figure 53).

Figure 53 IQN Pool for Fabric B

Dialog box titled "Create IQN Suffix Pool" within the "Unified Computing System Manager".

Define Name and Description

1. Define Name and Description

2. Add IQN Blocks

Name:

Description:

Prefix:

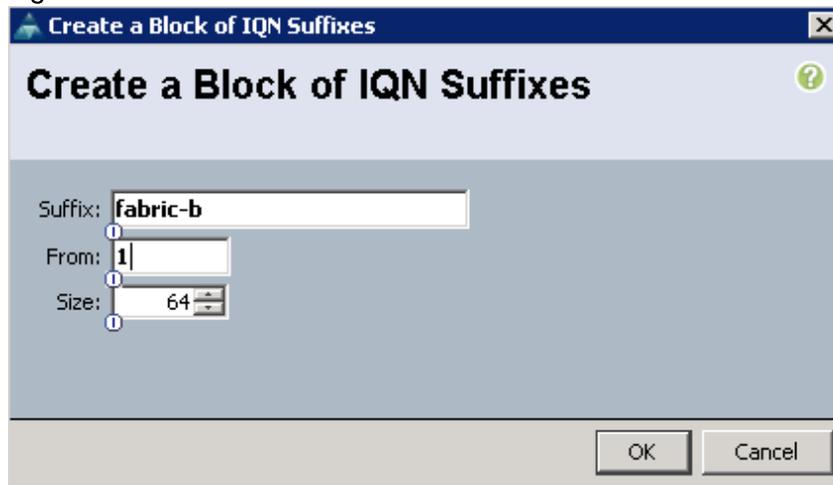
Assignment Order: Default Sequential

Buttons: < Prev, Next >, Finish, Cancel

22. Click Next.
23. Click Add
24. Enter fabric-b as the suffix
25. Enter 1 in the From field
26. Specify the size of IQN block sufficient to support the available server resources (Figure 54).

27. Click OK.
28. Click Finish
29. In the message box that displays, click OK.

Figure 54 Block of IQN Suffixes for Fabric B



Create iSCSI Initiator IP Pools for iSCSI Boot

To configure the necessary IP pools for iSCSI boot, complete the following steps:

1. Select the LAN tab on the left pane.
2. Select Pools > root.



Two IP pools are created, one for each fabric.

3. Right-click IP pools under the root organization.
4. Select Create IP Pool to create the IP pool.
5. Enter iSCSI_Initiator_Pool_A for the name of IP pool (Figure 55).
6. Optional: Enter a description of the IP pool.
7. Select Sequential for Assignment Order.
8. Click Next

Figure 55 iSCSI Initiator Pool for Fabric A

Create IP Pool

Unified Computing System Manager

Create IP Pool

1. **Define Name and Description**
2. Add IPv4 Blocks
3. Add IPv6 Blocks

Define Name and Description

Name:

Description:

Assignment Order: Default Sequential

< Prev Next > Finish Cancel

9. Click Add.
10. In the From field, enter the beginning of the range to assign an iSCSI IP addresses.
11. Set the size enough to accommodate the servers (Figure 56).
12. Enter a default gateway in order to avoid a problem during first boot after the servers are provisioned. More information about this is available in the subsection titled Configure Network Interfaces connected to NetApp FAS8040, specifically the Create SVI section.
13. Click OK.
14. Click Finish.

Figure 56 Block of IPv4 Addresses for iSCSI Initiator Pool for Fabric A

Create Block of IPv4 Addresses

Create a Block of IPv4 Addresses

From: Size:

Subnet Mask: Default Gateway:

Primary DNS: Secondary DNS:

OK Cancel

15. Right-click IP pools under the root organization.
16. Select Create IP Pool to create the IP pool.
17. Enter iSCSI_Initiator_Pool_B for the name of IP pool (Figure 57).

18. Optional: Enter a description of the IP pool.

19. Select Sequential for Assignment Order.

20. Click Next

Figure 57 iSCSI Initiator Pool for Fabric B

The screenshot shows the 'Create IP Pool' dialog box in the Unified Computing System Manager. The dialog is titled 'Define Name and Description'. On the left, there is a progress list with three steps: 1. Define Name and Description (checked), 2. Add IPv4 Blocks, and 3. Add IPv6 Blocks. The main area contains the following fields and options:

- Name:
- Description:
- Assignment Order: Default Sequential

At the bottom, there are four buttons: '< Prev', 'Next >', 'Finish', and 'Cancel'.

21. Click Add.

22. In the From field, enter the beginning of the range to assign an iSCSI IP addresses.

23. Set the size enough to accommodate the servers (Figure 58).

24. Enter a default gateway in order to avoid a problem during first boot after the servers are provisioned. More information on this is available in the subsection titled Configure Network Interfaces connected to NetApp FAS8040, specifically the Create SVI section.

25. Click OK.

26. Click Finish.

Figure 58 Block of IPv4 Addresses for iSCSI Initiator Pool for Fabric B

The screenshot shows the 'Create Block of IPv4 Addresses' dialog box. It contains the following fields and values:

- From:
- Size:
- Subnet Mask:
- Default Gateway:
- Primary DNS:
- Secondary DNS:

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

Create UUID Suffix Pools

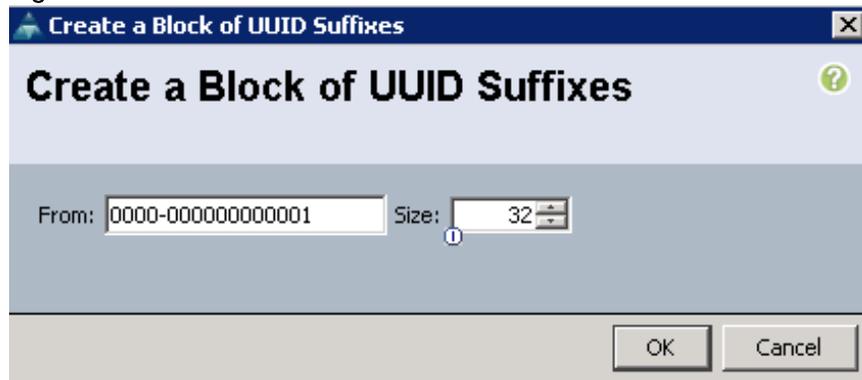
To configure the necessary universally unique identifier (UUID) suffix pool for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Pools > root.
3. Right-click UUID Suffix Pools.
4. Select Create UUID Suffix Pool.
5. Enter UUID_Pool as the name of the UUID suffix pool (Figure 59).
6. Optional: Enter a description for the UUID suffix pool.
7. Keep the prefix at the derived option.

Figure 59 Cisco UCS – Create UUID Pool

8. Click Next.
9. Click Add to add a block of UUIDs.
10. Keep the From field at the default setting.
11. Specify a size for the UUID block that is sufficient to support the available blade or server resources (Figure 60).

Figure 60 Cisco UCS – Create UUID Block



12. Click OK.
13. Click Finish.
14. Click OK

Create Server Pools

In FlexPod with RHEL OPS 6, two server pools are created. One for controller nodes and one for compute nodes. These server pools are used in their respective service profiles for scalability and granularity perspective. Additional compute host can be configured quickly by adding it into the compute server pool.

Create Controller Hosts Server Pool

To create a server pool, complete the following steps:

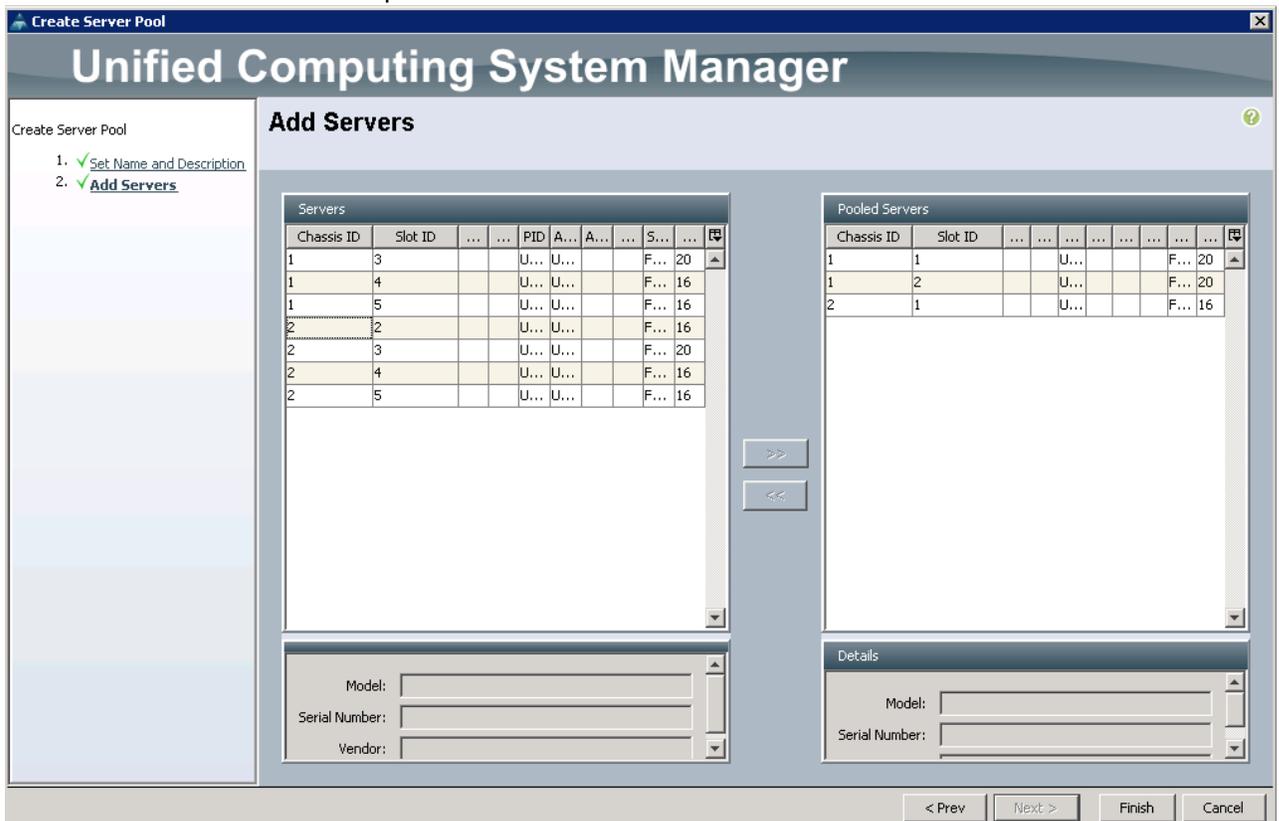
1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Pools > root.
3. Right-click Server Pools.
4. Select Create Server Pool.
5. Enter Openstack_Controller_Pool as the name of the server pool.
6. Optional: Enter a description for the server pool (Figure 61).

Figure 61 Create Server Pool for Controllers

The screenshot shows the 'Create Server Pool' wizard in the Unified Computing System Manager. The window title is 'Create Server Pool'. The main heading is 'Unified Computing System Manager'. The current step is 'Set Name and Description'. On the left, a progress indicator shows two steps: '1. Set Name and Description' (completed with a green checkmark) and '2. Add Servers' (pending). The main area contains two text input fields: 'Name:' with the value 'Openstack_Controller_Pool' and 'Description:' which is empty. At the bottom right, there are four buttons: '< Prev', 'Next >', 'Finish', and 'Cancel'.

7. Click Next.
8. Select the servers as shown in Figure 62 to be added in Openstack_Controller_Pool. Click >> to add servers.

Figure 62 Servers Selection for OpenStack Controller Pool



9. Click Finish.

10. Click OK

Create Compute Hosts Server Pool

To create a server pool, complete the following steps:

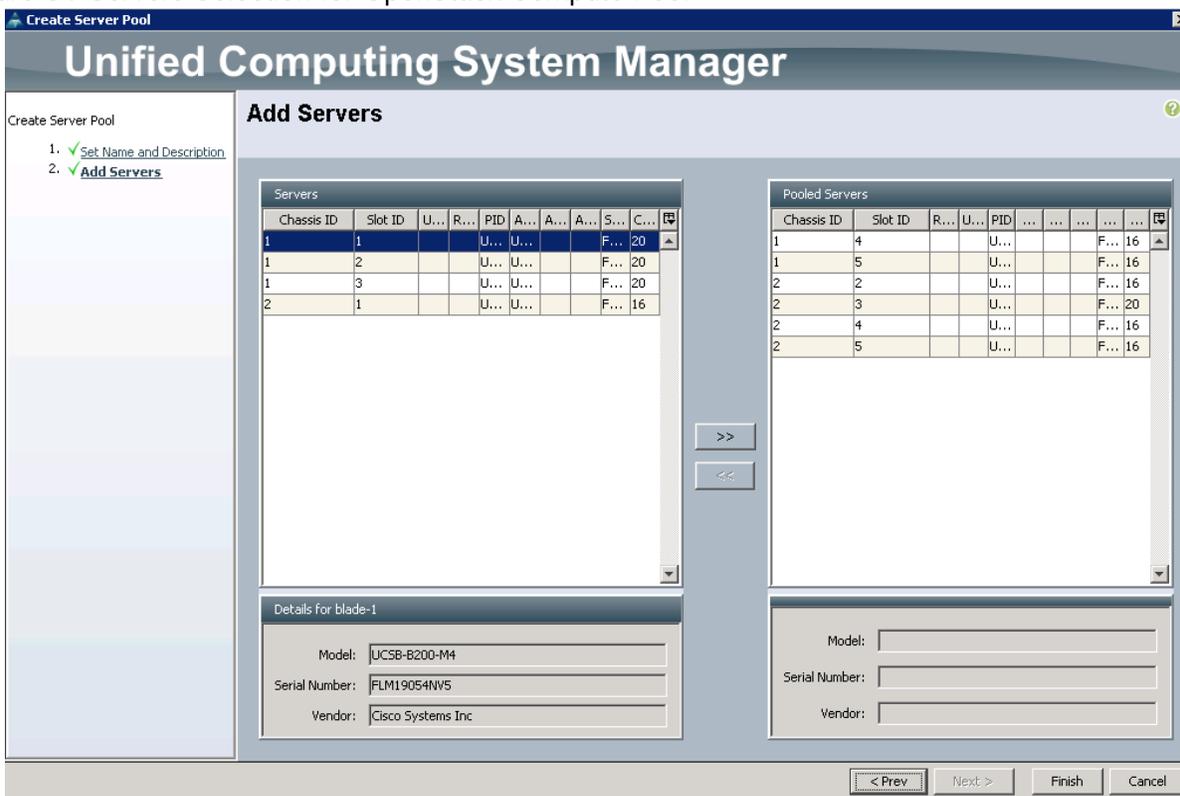
1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Pools > root.
3. Right-click Server Pools.
4. Select Create Server Pool.
5. Enter Openstack_Compute_Pool as the name of the server pool.
6. Optional: Enter a description for the server pool (Figure 63).

Figure 63 Create Server Pool for Compute

The screenshot shows the 'Create Server Pool' wizard in the Unified Computing System Manager. The window title is 'Create Server Pool' and the main header is 'Unified Computing System Manager'. The wizard is titled 'Set Name and Description'. On the left, a progress indicator shows two steps: '1. Set Name and Description' (completed with a green checkmark) and '2. Add Servers' (pending). The main area contains two text input fields: 'Name:' with the value 'Openstack_Compute_Pool' and 'Description:' which is empty. At the bottom right, there are four buttons: '< Prev', 'Next >', 'Finish', and 'Cancel'.

7. Click Next.
8. Select the servers as shown in Figure 64 to be added in Openstack_Compute_Pool. Click >> to add servers.

Figure 64 Servers Selection for OpenStack Compute Pool



9. Click Finish.

10. Click OK.

Create VLANs

Create Management VLAN

To create a management VLAN, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud.
3. Right-click VLANs.
4. Select Create VLANs (Figure 65).
5. Enter Management-10 as the name of the VLAN to be used for management traffic.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter `<<var_mgmt_vlan_id>>` as the ID of the management VLAN.
8. Keep the Sharing Type as None.
9. Click OK, and then click OK again.

Figure 65 Create Management VLAN

Create VLANs

VLAN Name/Prefix:

Multicast Policy Name: + Create Multicast Policy

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

Check Overlap OK Cancel

Create PXE VLAN

To create a PXE VLAN, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud.
3. Right-click VLANs.
4. Select Create VLANs (Figure 66).
5. Enter PXE-20 as the name of the VLAN to be used for PXE traffic.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter `<<var_pxe_vlan_id>>` as the ID of the PXE VLAN.
8. Keep the Sharing Type as None.
9. Click OK, and then click OK again.

Figure 66 Create PXE VLAN

Create VLANs

VLAN Name/Prefix:

Multicast Policy Name: + Create Multicast Policy

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

Create NFS VLAN

To create a NFS VLAN, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud.
3. Right-click VLANs.
4. Select Create VLANs (Figure 67).
5. Enter NFS-30 as the name of the VLAN to be used for NFS traffic.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter <<var_nfs_vlan_id>> as the ID of the NFS VLAN.
8. Keep the Sharing Type as None.

- Click OK, and then click OK again.

Figure 67 Create NFS VLAN

Create VLANs

VLAN Name/Prefix:

Multicast Policy Name:

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

Create iSCSI VLAN for Fabric A

To create iSCSI VLAN for fabric A, complete the following steps:

- In Cisco UCS Manager, click the LAN tab in the navigation pane.
- Select LAN > LAN Cloud.
- Right-click VLANs.
- Select Create VLANs (Figure 68).
- Enter iSCSI-40 as the name of the VLAN to be used for iSCSI-A traffic.
- Keep the Common/Global option selected for the scope of the VLAN.
- Enter <<var_iscsi_a_vlan_id>> as the ID of the iSCSI A VLAN.

8. Keep the Sharing Type as None.
9. Click OK, and then click OK again.

Figure 68 Create iSCSI VLAN for Fabric A

Create VLANs

VLAN Name/Prefix:

Multicast Policy Name:

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

Create iSCSI VLAN for Fabric B

To create iSCSI VLAN for fabric B, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud.
3. Right-click VLANs.
4. Select Create VLANs (Figure 69).
5. Enter iSCSI-41 as the name of the VLAN to be used for iSCSI-B traffic.
6. Keep the Common/Global option selected for the scope of the VLAN.

7. Enter <<var_iscsi_B_vlan_id>> as the ID of the iSCSI B VLAN.
8. Keep the Sharing Type as None.
9. Click OK, and then click OK again.

Figure 69 Create iSCSI VLAN for Fabric B

Create VLANs

VLAN Name/Prefix:

Multicast Policy Name:

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

Create Swift VLAN for Fabric A

To create a swift VLAN for fabric A, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud.
3. Right-click VLANs.
4. Select Create VLANs (Figure 70).
5. Enter Swift-50 as the name of the VLAN to be used for Swift-A traffic.

6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter <<var_swift_A_vlan_id>> as the ID of the Swift A VLAN.
8. Keep the Sharing Type as None.
9. Click OK, and then click OK again.

Figure 70 Create Swift VLAN for Fabric A

Create VLANs

VLAN Name/Prefix:

Multicast Policy Name:

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

Create Swift VLAN for Fabric B

To create a swift VLAN for fabric B, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud.
3. Right-click VLANs.
4. Select Create VLANs (Figure 71).

5. Enter Swift-51 as the name of the VLAN to be used for Swift-B traffic.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter <<var_swift_B_vlan_id>> as the ID of the Swift B VLAN.
8. Keep the Sharing Type as None.
9. Click OK, and then click OK again.

Figure 71 Create Swift VLAN for Fabric B

Create VLANs

VLAN Name/Prefix:

Multicast Policy Name: [+ Create Multicast Policy](#)

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

Create External VLAN

To create an external VLAN, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud.
3. Right-click VLANs.

4. Select Create VLANs (Figure 72).
5. Enter External-215 as the name of the VLAN to be used for external traffic.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter <<var_external_vlan_id>> as the ID of the external VLAN.
8. Keep the Sharing Type as None.
9. Click OK, and then click OK again.

Figure 72 Create External VLAN

Create VLANs

VLAN Name/Prefix:

Multicast Policy Name: [+ Create Multicast Policy](#)

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

Create Provider VLANs

To create a provider VLAN, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud.

3. Right-click VLANs.
4. Select Create VLANs (Figure 73).
5. Enter Provider- as the name of the VLAN to be used for provider network traffic.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter <<var_provider_vlan_range>> in the VLAN IDs field.
8. Keep the Sharing Type as None.
9. Click OK, and then click OK again.

Figure 73 Create Provider VLANs

Create VLANs

VLAN Name/Prefix:

Multicast Policy Name:

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

Create Tenant VLANs

To create tenant VLANs, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.

2. Select LAN > LAN Cloud.
3. Right-click VLANs.
4. Select Create VLANs (Figure 74).
5. Enter Tenant- as the name of the VLAN to be used for tenant network traffic.
6. Keep the Common/Global option selected for the scope of the VLAN.
7. Enter <<var_tenant_vlan_range>> in the VLAN IDs field.
8. Keep the Sharing Type as None.
9. Click OK, and then click OK again.

Figure 74 Create Tenant VLANs

Create VLANs

VLAN Name/Prefix:

Multicast Policy Name:

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

Create OpenStack Management, Cluster Management, Admin API, and Storage Clustering VLAN

In Cisco UCS Manager, click the LAN tab in the navigation pane.

1. Select LAN > LAN Cloud.
2. Right-click VLANs.
3. Select Create VLANs.
4. Enter MCAS-21 as the name of the VLAN to be used for OpenStack Management, Cluster Management, Admin API, and Storage Clustering traffic (Figure 75).
5. Keep the Common/Global option selected for the scope of the VLAN.
6. Enter <<var_mcas_vlan_id>> in the VLAN IDs field.
7. Keep the Sharing Type as None.
8. Click OK, and then click OK again.

Figure 75 Create VLAN For Management, Cluster Management, Admin API, and Storage Clustering

The screenshot shows the 'Create VLANs' dialog box with the following configuration:

- VLAN Name/Prefix:** MCAS-21
- Multicast Policy Name:** <not set> (with a '+ Create Multicast Policy' button)
- Scope:** Common/Global, Fabric A, Fabric B, Both Fabrics Configured Differently
- Instruction:** 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:** 21
- Sharing Type:** None, Primary, Isolated, Community
- Buttons:** Check Overlap, OK, Cancel

Create Host Firmware Package

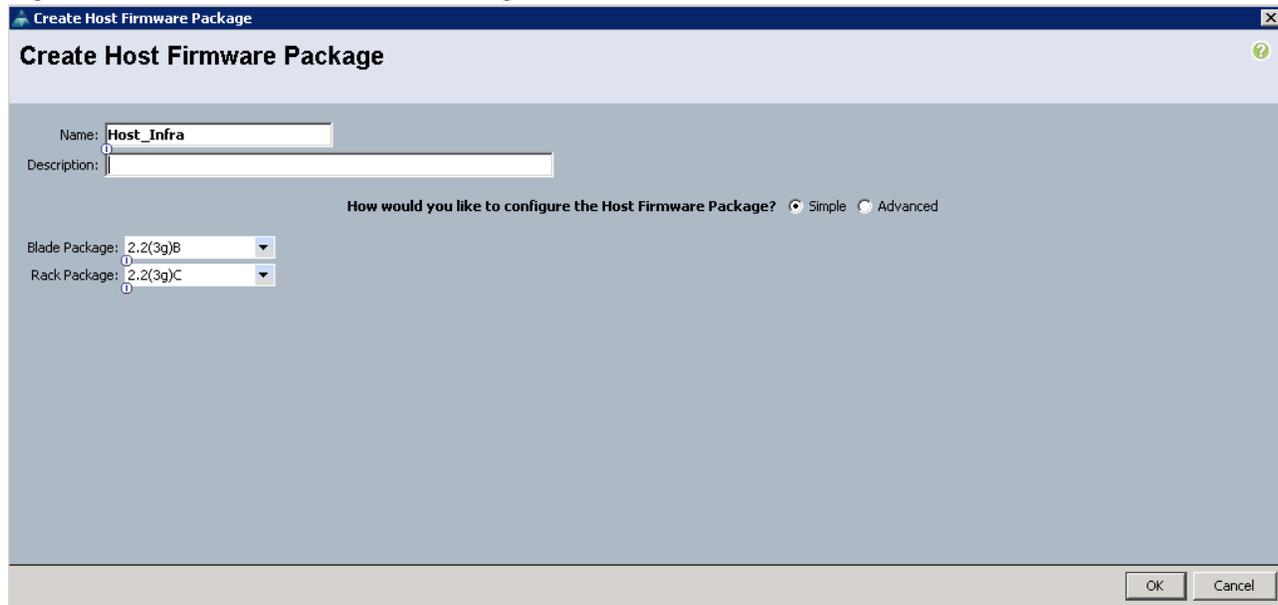
Firmware management policies allow the administrator to select the corresponding packages for a given server configuration. These policies often include packages for adapter, BIOS, board controller, FC adapters, host bus adapter (HBA) option ROM, and storage controller properties.

To create a firmware management policy for a given server configuration in the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root.

3. Right-click Host Firmware Packages.
4. Select Create Host Firmware Package (Figure 76).
5. Enter Host_Infra as the name of the host firmware package.
6. Leave Simple selected.
7. Select the version 2.2(3g) for both the Blade and Rack Packages.

Figure 76 Create Host Firmware Package



The screenshot shows a dialog box titled "Create Host Firmware Package". The "Name" field is filled with "Host_Infra". The "Description" field is empty. Below the description, there is a question: "How would you like to configure the Host Firmware Package?" with two radio buttons: "Simple" (selected) and "Advanced". Underneath, there are two dropdown menus: "Blade Package" and "Rack Package", both set to "2.2(3g)". At the bottom right, there are "OK" and "Cancel" buttons.

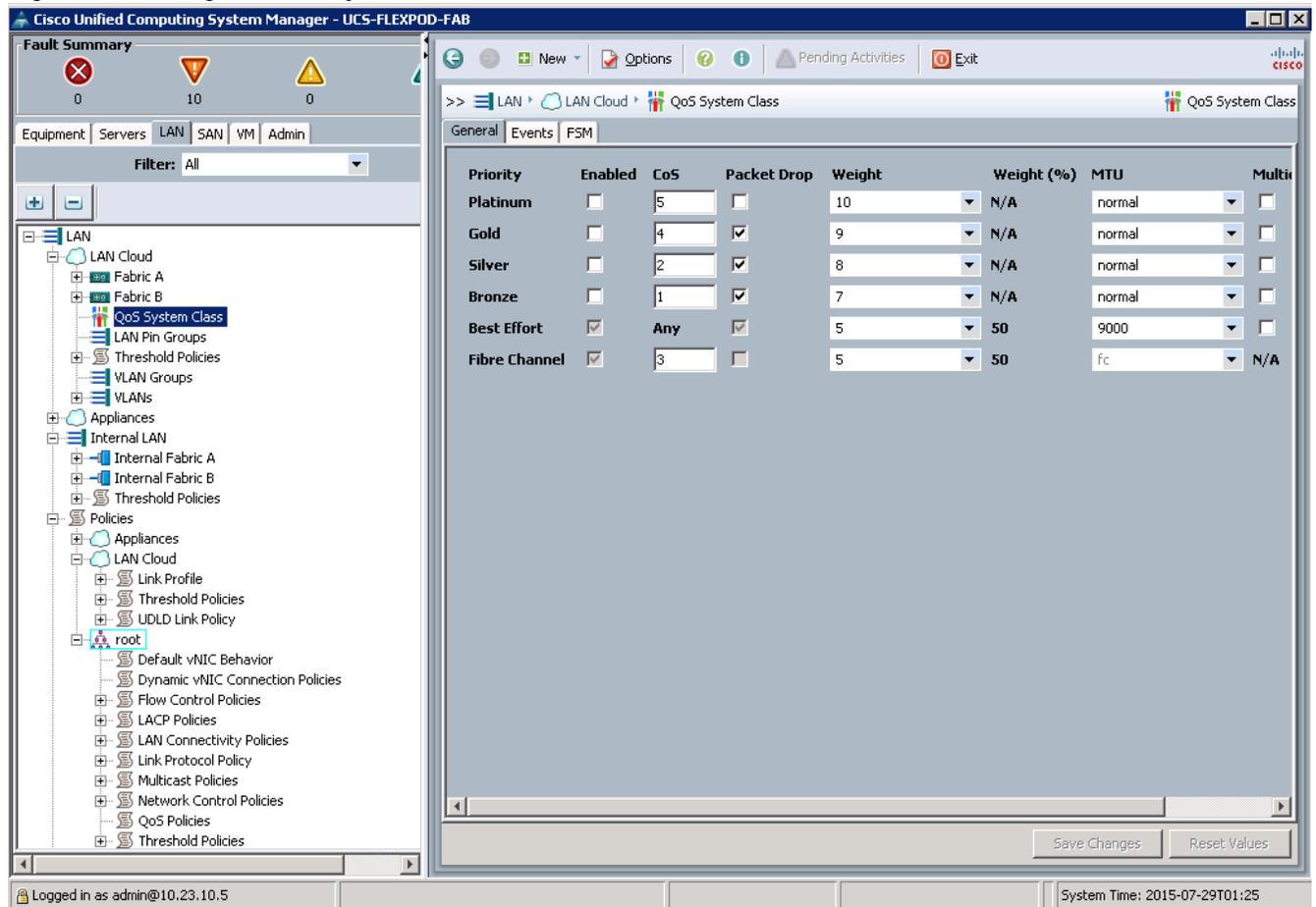
8. Click OK to create the host firmware package.
9. Click OK.

Set Jumbo Frames in Cisco UCS

To configure jumbo frames and enable quality of service in the Cisco UCS fabric, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select LAN > LAN Cloud > QoS System Class (Figure 77).
3. In the right pane, click the General tab.
4. On the Best Effort row, enter 9000 in the box under the MTU column.
5. Click Save Changes in the bottom of the window.
6. Click OK.

Figure 77 Configure QoS System Class for Jumbo Frame



Create Local Disk Configuration Policy

A local disk configuration for the Cisco UCS environment is necessary if the servers in the environment do not have a local disk.



In this solution, there are no local disks present in any of the physical servers. It is important to setup a local disk policy and utilize it in the service profiles.

To create a local disk configuration policy, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root.
3. Right-click Local Disk Config Policies.
4. Select Create Local Disk Configuration Policy (Figure 78).
5. Enter No_Local_Disk as the local disk configuration policy name.
6. Change the mode to No Local Storage.

7. Click OK to create the local disk configuration policy.

Figure 78 Configure Local Disk Configuration Policy

Create Local Disk Configuration Policy

Name:

Description:

Mode:

FlexFlash

FlexFlash State: Disable Enable

If **FlexFlash State** is disabled, SD cards will become unavailable immediately.
Please ensure SD cards are not in use before disabling the FlexFlash State.

FlexFlash RAID Reporting State: Disable Enable

OK Cancel

Create Network Control Policy for Cisco Discovery Protocol

To create a network control policy that enables Cisco Discovery Protocol (CDP) on virtual network ports, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root.
3. Right-click Network Control Policies.
4. Select Create Network Control Policy (Figure 79).
5. Enter Enable_CDP as the policy name.
6. For CDP, select the Enabled option.
7. Click OK to create the network control policy.

Figure 79 Network Control Policy for CDP

Create Network Control Policy

Name:

Description:

CDP: Disabled Enabled

MAC Register Mode: Only Native Vlan All Host Vlans

Action on Uplink Fail: Link Down Warning

MAC Security

Forge: Allow Deny

OK Cancel

Create Power Control Policy

To create a power control policy for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root.
3. Right-click Power Control Policies.
4. Select Create Power Control Policy (Figure 80).
5. Enter No_Power_Cap as the power control policy name.
6. Change the power capping setting to No Cap.
7. Click OK to create the power control policy.
8. Click OK.

Figure 80 Power Control Policy

Create Power Control Policy

Name:

Description:

Power Capping

If you choose **cap**, the server is allocated a certain amount of power based on its priority within its power group. Priority values range from 1 to 10, with 1 being the highest priority. If you choose **no-cap**, the server is exempt from all power capping.

No Cap cap

Cisco UCS Manager only enforces power capping when the servers in a power group require more power than is currently available. With sufficient power, all servers run at full capacity regardless of their priority.

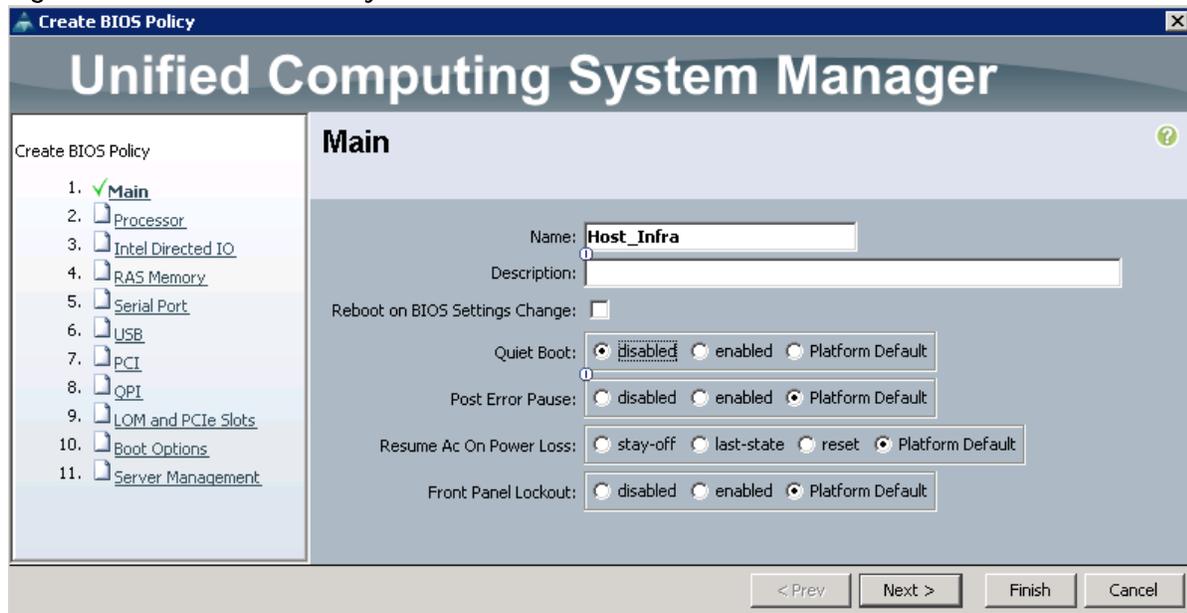
OK Cancel

Create BIOS Policy

To create a server BIOS policy for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root.
3. Right-click BIOS Policies.
4. Select Create BIOS Policy (Figure 81).
5. Enter Host_Infra as the BIOS policy name.
6. Change the Quiet Boot setting to Disabled.
7. Click Finish to create the BIOS policy.

Figure 81 Create BIOS Policy



Create vNIC Placement Policy for Red Hat Enterprise Linux Hosts

To create a vNIC/vHBA placement policy for the infrastructure hosts, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root.
3. Right-click vNIC/vHBA Placement Policies.
4. Select Create Placement Policy (Figure 82).
5. Enter Host-Infra as the name of the placement policy.
6. Click 1 and select Assigned Only.
7. Click OK, and then click OK again.

Figure 82 vNIC Placement Policy

Create Placement Policy

Name:

Virtual Slot Mapping Scheme: Round Robin Linear Ordered

Filter | Export | Print

Virtual Slot	Selection Preference
1	Assigned Only
2	All
3	All
4	All

OK Cancel

Create Maintenance Policy

To update the default Maintenance Policy, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root.
3. Right-click Maintenance Policies (Figure 83).
4. Click Create Maintenance Policy
5. Enter user_ack in the maintenance policy name
6. Change the Reboot Policy to User Ack.
7. Click OK.

Figure 83 Create user_ack Maintenance Policy

Create Maintenance Policy

Name:

Description:

Reboot Policy: Immediate User Ack Timer Automatic

OK Cancel

Create vNIC Templates

To create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment, complete the following steps:

Create Management vNIC Template

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template
5. Enter Management_A as the vNIC template name.
6. Keep Fabric A selected.
7. Select the Enable Failover checkbox.
8. Under Target, make sure that the VM checkbox is not selected.
9. Select Updating Template as the Template Type.
10. Under VLANs, select the checkboxes for Management-10 VLAN.
11. For MTU, enter 1500.
12. In the MAC Pool list, select MAC_Pool_A.
13. In the Network Control Policy list, select Enable_CDP.

14. Click OK to create the vNIC template (Figure 84).

15. Click OK.

Figure 84 Create Management vNIC Template

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-215	<input type="radio"/>
<input checked="" type="checkbox"/>	Management-10	<input type="radio"/>
<input type="checkbox"/>	NFS-30	<input type="radio"/>
<input type="checkbox"/>	PXE-20	<input type="radio"/>
<input type="checkbox"/>	Provider-60	<input type="radio"/>

+ Create VLAN

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Pin Group:

OK Cancel

Create PXE vNIC Template

To create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root.
3. Right-click vNIC Templates.

4. Select Create vNIC Template
5. Enter PXE_Template_A as the vNIC template name.
6. Keep Fabric A selected.
7. Select the Enable Failover checkbox.
8. Under Target, make sure that the VM checkbox is not selected.
9. Select Updating Template as the Template Type.
10. Under VLANs, select the checkboxes for PXE-20 VLAN.
11. Set Native-VLAN as the native VLAN.
12. For MTU, enter 1500.
13. In the MAC Pool list, select MAC_Pool_A.
14. In the Network Control Policy list, select Enable_CDP.
15. Click OK to create the vNIC template (Figure 85).
16. Click OK.

Figure 85 Create PXE vNIC Template

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-215	<input type="radio"/>
<input type="checkbox"/>	Management-10	<input type="radio"/>
<input type="checkbox"/>	NFS-30	<input type="radio"/>
<input checked="" type="checkbox"/>	PXE-20	<input checked="" type="radio"/>
<input type="checkbox"/>	Provider-60	<input type="radio"/>

+ Create VLAN

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Pin Group:

OK Cancel

Create Network Storage vNIC Template for NFS (Cinder and Glance) and Swift for Fabric A

To create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template
5. Enter NS_Template_A as the vNIC template name.
6. Keep Fabric A selected.
7. Do not select the Enable Failover checkbox.

8. Under Target, make sure that the VM checkbox is not selected.
9. Select Updating Template as the Template Type.
10. Under VLANs, select the checkboxes for NFS-30, Swift-50, and Swift-51 VLANs.
11. For MTU, enter 9000.
12. In the MAC Pool list, select MAC_Pool_A.
13. In the Network Control Policy list, select Enable_CDP.
14. Click OK to create the vNIC template (Figure 86).
15. Click OK.

Figure 86 Create NS_Template_A vNIC Template

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"/>	iSCSI-41	<input type="radio"/>
<input type="checkbox"/>	Management-10	<input type="radio"/>
<input checked="" type="checkbox"/>	NFS-30	<input type="radio"/>
<input type="checkbox"/>	Provider-60	<input type="radio"/>
<input type="checkbox"/>	Provider-61	<input type="radio"/>
<input type="checkbox"/>	Provider-62	<input type="radio"/>

Create VLAN

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Pin Group:

OK Cancel

Create Network Storage vNIC Template for NFS (Cinder and Glance) and Swift For Fabric B

To create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template
5. Enter NS_Template_B as the vNIC template name.
6. Select Fabric B
7. Do not select the Enable Failover checkbox.
8. Under Target, make sure that the VM checkbox is not selected.
9. Select Updating Template as the Template Type.
10. Under VLANs, select the checkboxes for for NFS-30, Swift-50, and Swift-51 VLANs.
11. For MTU, enter 9000.
12. In the MAC Pool list, select MAC_Pool_B.
13. In the Network Control Policy list, select Enable_CDP.
14. Click OK to create the vNIC template (Figure 87).
15. Click OK.

Figure 87 Create NS_Template_B vNIC Template

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"/>	Management-10	<input type="radio"/>
<input checked="" type="checkbox"/>	NF5-30	<input type="radio"/>
<input type="checkbox"/>	Provider-60	<input type="radio"/>
<input type="checkbox"/>	Provider-61	<input type="radio"/>
<input type="checkbox"/>	Provider-62	<input type="radio"/>
<input type="checkbox"/>	Provider-63	<input type="radio"/>

Create VLAN

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Pin Group:

OK Cancel

Create iSCSI vNIC Template for Fabric A

To create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template
5. Enter iSCSI_Template_A as the vNIC template name.
6. Keep Fabric A selected.
7. Do not select the Enable Failover checkbox.

8. Under Target, make sure that the VM checkbox is not selected.
9. Select Updating Template as the Template Type.
10. Under VLANs, select the checkboxes for iSCSI-40 VLAN
11. Set Native-VLAN as the native VLAN.
12. For MTU, enter 9000.
13. In the MAC Pool list, select MAC_Pool_A.
14. In the Network Control Policy list, select Enable_CDP.
15. Click OK to create the vNIC template (Figure 88).
16. Click OK.

Figure 88 Create iSCSI vNIC Template for Fabric A

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"/>	Tenant-96	<input type="radio"/>
<input type="checkbox"/>	Tenant-97	<input type="radio"/>
<input type="checkbox"/>	Tenant-98	<input type="radio"/>
<input type="checkbox"/>	Tenant-99	<input type="radio"/>
<input checked="" type="checkbox"/>	iSCSI-40	<input checked="" type="radio"/>
<input type="checkbox"/>	iSCSI-41	<input type="radio"/>

+ Create VLAN

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Pin Group:

OK Cancel

Create iSCSI vNIC Template for Fabric B

To create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template
5. Enter iSCSI_Template_A as the vNIC template name.
6. Select Fabric B.

7. Do not select the Enable Failover checkbox.
8. Under Target, make sure that the VM checkbox is not selected.
9. Select Updating Template as the Template Type.
10. Under VLANs, select the checkboxes for iSCSI-41 VLAN.
11. Set Native-VLAN as the native VLAN.
12. For MTU, enter 9000.
13. In the MAC Pool list, select MAC_Pool_B.
14. In the Network Control Policy list, select Enable_CDP.
15. Click OK to create the vNIC template (Figure 89).
16. Click OK.

Figure 89 Create iSCSI vNIC Template for Fabric B

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"/>	Tenant-96	<input type="radio"/>
<input type="checkbox"/>	Tenant-97	<input type="radio"/>
<input type="checkbox"/>	Tenant-98	<input type="radio"/>
<input type="checkbox"/>	Tenant-99	<input type="radio"/>
<input type="checkbox"/>	iSCSI-40	<input type="radio"/>
<input checked="" type="checkbox"/>	iSCSI-41	<input checked="" type="radio"/>

Create VLAN

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Pin Group:

OK Cancel

Create External vNIC Template

To create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template
5. Enter Ext_Template as the vNIC template name.
6. Select Fabric B

7. Select the Enable Failover checkbox.
8. Under Target, make sure that the VM checkbox is not selected.
9. Select Updating Template as the Template Type.
10. Under VLANs, select the checkboxes for External-215 VLAN.
11. For MTU, enter 1500.
12. In the MAC Pool list, select MAC_Pool_B.
13. In the Network Control Policy list, select Enable_CDP.
14. Click OK to create the vNIC template (Figure 90).
15. Click OK.

Figure 90 Create External vNIC Template

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 checked="" type="checkbox"/>	External-215	<input type="radio"/>
<input type="checkbox"/>	Management-10	<input type="radio"/>
<input type="checkbox"/>	NFS-30	<input type="radio"/>
<input type="checkbox"/>	PXE-20	<input type="radio"/>
<input type="checkbox"/>	Provider-60	<input type="radio"/>

+ Create VLAN

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Pin Group:

OK Cancel

Create VM Traffic vNIC Template

To create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template
5. Enter VM_Traffic as the vNIC template name.
6. Select Fabric B
7. Select the Enable Failover checkbox.
8. Under Target, make sure that the VM checkbox is not selected.
9. Select Updating Template as the Template Type.
10. Under VLANs, select the checkboxes for VLAN 60-200 and 215.
11. For MTU, enter 1500.
12. In the MAC Pool list, select MAC_Pool_B.
13. In the Network Control Policy list, select Enable_CDP.
14. Click OK to create the vNIC template (Figure 91).
15. Click OK.

Figure 91 Create VM Traffic vNIC Template

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 checked="" type="checkbox"/>	External-215	<input type="radio"/>
<input type="checkbox"/>	Management-10	<input type="radio"/>
<input type="checkbox"/>	NFS-30	<input type="radio"/>
<input type="checkbox"/>	PXE-20	<input type="radio"/>
<input checked="" type="checkbox"/>	Provider-60	<input type="radio"/>

+ Create VLAN

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

Pin Group:

OK Cancel

Create OpenStack Management, Cluster Management, Admin API, and Storage Clustering Traffic vNIC Template

To create multiple virtual network interface card (vNIC) templates for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the LAN tab in the navigation pane.
2. Select Policies > root.
3. Right-click vNIC Templates.
4. Select Create vNIC Template
5. Enter MCAS_Template_A as the vNIC template name.
6. Select Fabric A

7. Select the Enable Failover checkbox.
8. Under Target, make sure that the VM checkbox is not selected.
9. Select Updating Template as the Template Type.
10. Under VLANs, select the checkbox for VLAN 21
11. For MTU, enter 1500.
12. In the MAC Pool list, select MAC_Pool_A.
13. In the Network Control Policy list, select Enable_CDP.
14. Click OK to create the vNIC template (Figure 92).
15. Click OK.

Figure 92 Create vNIC Template for OpenStack Management, Cluster Management, Admin API, and Storage Clustering Traffic

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-215	<input type="radio"/>
<input checked="" type="checkbox"/>	MCAS-21	<input type="radio"/>
<input type="checkbox"/>	Management-10	<input type="radio"/>
<input type="checkbox"/>	NFS-30	<input type="radio"/>
<input type="checkbox"/>	PXE-20	<input type="radio"/>

+ Create VLAN

MTU:

MAC Pool:

QoS Policy:

Network Control Policy:

OK Cancel

Create Boot Policy

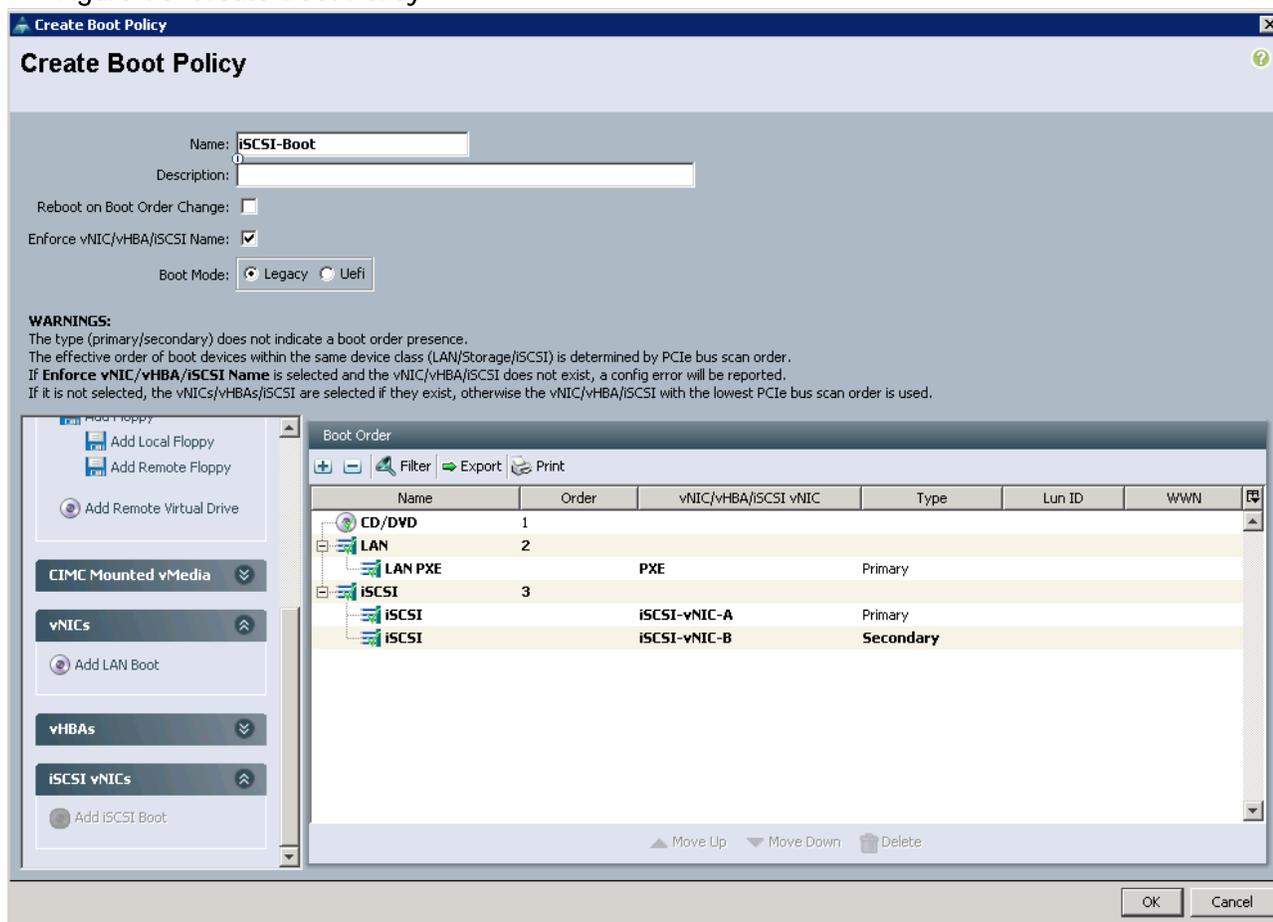
This procedure applies to Cisco UCS environment in which two iSCSI logical interfaces (LIFs) are on cluster node 1 (iscsi_lif01a and iscsi_lif01b) and two iSCSI LIFs are on cluster node 2 (iscsi_lif02a and iscsi_lif02b). Also, it is assumed that the “a” LIFs are connected to fabric A (Cisco Nexus A) and the “b” LIFs are connected to fabric B (Cisco Nexus B).

One boot policy is configured in this procedure. This policy configures the primary target to be iscsi_lif01a.

To create boot policies for the Cisco UCS environment, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root.
3. Right-click Boot Policies.
4. Select Create Boot Policy.
5. Enter iSCSI-Boot as the name of the boot policy.
6. Optional: Enter a description for the boot policy.
7. Keep the Reboot on Boot Order Change option cleared.
8. Expand the Local Devices drop-down menu and select Add CD/DVD
9. Expand vNICs drop-down menu and select Add LAN Boot.
10. In the “Add LAN Boot” dialog box, add PXE for vNIC. Click OK.
11. Expand the iSCSI vNICs section and select Add iSCSI Boot.
12. In the “Add iSCSI Boot” dialog box, enter iSCSI-vNIC-A.
13. Click OK.
14. Select Add iSCSI Boot.
15. In the Add iSCSI Boot dialog box, enter iSCSI-vNIC-B.
16. Click OK (Figure 93).
17. Click OK to save the boot policy. Click OK to close the Boot Policy window.

Figure 93 Create Boot Policy



Create Service Profile Templates

The following service profile templates will be created next:

1. Service profile template for OpenStack Controller hosts.
2. Service profile template for OpenStack Compute hosts.
3. Service profile template for OpenStack Installer host.

Service Profile Templates for OpenStack Controller Hosts

To create the service profile template, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Service Profile Templates > root.
3. Right-click root.
4. Select Create Service Profile Template to open the Create Service Profile Template wizard.
5. Identify the service profile template (Figure 94):

- a. Enter OpenStack_Controller as the name of the service profile template. This service profile template is configured to boot from node 1 on fabric A.
- b. Select the Updating Template option.
- c. Under UUID, select UUID_Pool as the UUID pool.
- d. Click Next.

Figure 94 OpenStack Controller Service Profile Template – UUID Assignment

Create Service Profile Template

Unified Computing System Manager

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.

1. **Identify Service Profile Template**

2. Networking

3. Storage

4. Zoning

5. vNIC/vHBA Placement

6. vMedia Policy

7. Server Boot Order

8. Maintenance Policy

9. Server Assignment

10. Operational Policies

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.

< Prev Next > Finish Cancel

6. Configure the networking options:
 - a. Keep the default setting for Dynamic vNIC Connection Policy.
 - b. Select the Expert option to configure the LAN connectivity.
 - c. Click the upper Add button to add a vNIC to the template.
 - d. In the Create vNIC dialog box (Figure 95), enter Management as the name of the vNIC.
 - e. Select the Use vNIC Template checkbox.
 - f. In the vNIC Template list, select Management_A.
 - g. In the Adapter Policy list, select Linux.
 - h. Click OK to add this vNIC to the template.

Figure 95 OpenStack Controller Service Profile Template – Management vNIC Creation

Create vNIC

Name:

Use vNIC Template:

[+ Create vNIC Template](#)

vNIC Template:

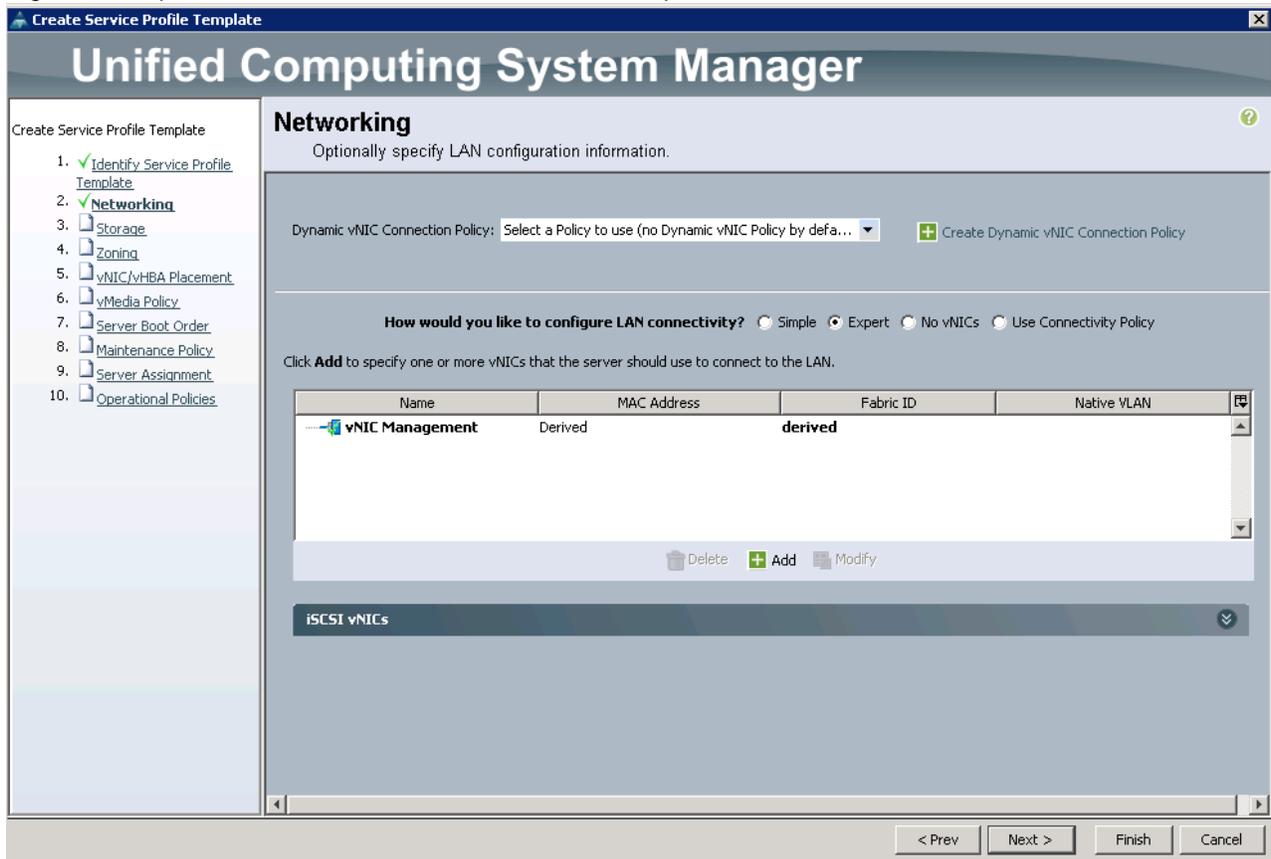
Adapter Performance Profile

Adapter Policy: [+ Create Ethernet Adapter Policy](#)

OK Cancel

- i. On the Networking page of the wizard, click the upper Add button to add another vNIC to the template (Figure 96).

Figure 96 OpenStack Controller Service Profile Template – Add vNIC



- j. In the Create vNIC dialog box, enter PXE as the name of the vNIC.
- k. Select the Use vNIC Template checkbox.
- l. In the vNIC Template list, select PXE_Template_A.
- m. In the Adapter Policy list, select Linux.
- n. Click OK to add this vNIC to the template (Figure 97).

Figure 97 OpenStack Controller Service Profile Template – PXE vNIC Creation

Create vNIC

Name:

Use vNIC Template:

vNIC Template:

Adapter Performance Profile

Adapter Policy:

- o. On the Networking page of the wizard, click the upper Add button to add another vNIC to the template.
- p. In the Create vNIC dialog box, enter NS-A as the name of the vNIC.
- q. Select the Use vNIC Template checkbox.
- r. In the vNIC Template list, select NS_Template_A.
- s. In the Adapter Policy list, select Linux.
- t. Click OK to add this vNIC to the template (Figure 98).

Figure 98 OpenStack Controller Service Profile Template – NS-A vNIC Creation

Create vNIC

Name:

Use vNIC Template:

+ Create vNIC Template

vNIC Template:

Adapter Performance Profile

Adapter Policy: + Create Ethernet Adapter Policy

OK Cancel

- u. On the Networking page of the wizard, click the upper Add button to add another vNIC to the template.
- v. In the Create vNIC dialog box, enter NS-B as the name of the vNIC.
- w. Select the Use vNIC Template checkbox.
- x. In the vNIC Template list, select NS_Template_B.
- y. In the Adapter Policy list, select Linux.
- z. Click OK to add this vNIC to the template (Figure 99).

Figure 99 OpenStack Controller Service Profile Template – NS-B vNIC Creation

Create vNIC

Name:

Use vNIC Template:

[+ Create vNIC Template](#)

vNIC Template:

Adapter Performance Profile

Adapter Policy: [+ Create Ethernet Adapter Policy](#)

OK Cancel

- aa. On the Networking page of the wizard, click the upper Add button to add another vNIC to the template.
- bb. In the Create vNIC dialog box, enter VM-Traffic as the name of the vNIC.
- cc. Select the Use vNIC Template checkbox.
- dd. In the vNIC Template list, select VM_Traffic
- ee. In the Adapter Policy list, select Linux.
- ff. Click OK to add this vNIC to the template (Figure 100).

Figure 100 OpenStack Controller Service Profile Template – VM-Traffic vNIC Creation

The screenshot shows a 'Create vNIC' dialog box. The title bar reads 'Create vNIC'. The main content area includes the following fields and controls:

- Name:** A text input field containing 'VM-Traffic'.
- Use vNIC Template:** A checked checkbox.
- + Create vNIC Template:** A button with a green plus icon.
- vNIC Template:** A dropdown menu showing 'VM_Traffic'.
- Adapter Performance Profile:** A section header.
- Adapter Policy:** A dropdown menu showing 'Linux'.
- + Create Ethernet Adapter Policy:** A button with a green plus icon.

At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

- gg. On the Networking page of the wizard, click the upper Add button to add another vNIC to the template.
- hh. In the Create vNIC dialog box, enter MCAS as the name of the vNIC.
- ii. Select the Use vNIC Template checkbox.
- jj. In the vNIC Template list, select MCAS_Template_A
- kk. In the Adapter Policy list, select Linux.
- ll. Click OK to add this vNIC to the template (Figure 101).

Figure 101 OpenStack Controller Service Profile Template – MCAS vNIC Creation

Create vNIC

Name:

Use vNIC Template:

+ Create vNIC Template

vNIC Template:

Adapter Performance Profile

Adapter Policy: + Create Ethernet Adapter Policy

OK Cancel

- mm. On the Networking page of the wizard, click the upper Add button to add another vNIC to the template.
- nn. In the Create vNIC dialog box, enter iSCSI-A as the name of the vNIC.
- oo. Select the Use vNIC Template checkbox.
- pp. In the vNIC Template list, select iSCSI_Template_A
- qq. In the Adapter Policy list, select Linux.
- rr. Click OK to add this vNIC to the template (Figure 102).

Figure 102 OpenStack Controller Service Profile Template – iSCSI-A vNIC Creation

Create vNIC

Name:

Use vNIC Template:

vNIC Template:

Adapter Performance Profile

Adapter Policy:

- ss. On the Networking page of the wizard, click the upper Add button to add another vNIC to the template.
- tt. In the Create vNIC dialog box, enter iSCSI-B as the name of the vNIC.
- uu. Select the Use vNIC Template checkbox.
- vv. In the vNIC Template list, select iSCSI_Template_B
- ww. In the Adapter Policy list, select Linux.
- xx. Click OK to add this vNIC to the template (Figure 103).

Figure 103 OpenStack Controller Service Profile Template – iSCSI-B vNIC Creation

Create vNIC

Name:

Use vNIC Template:

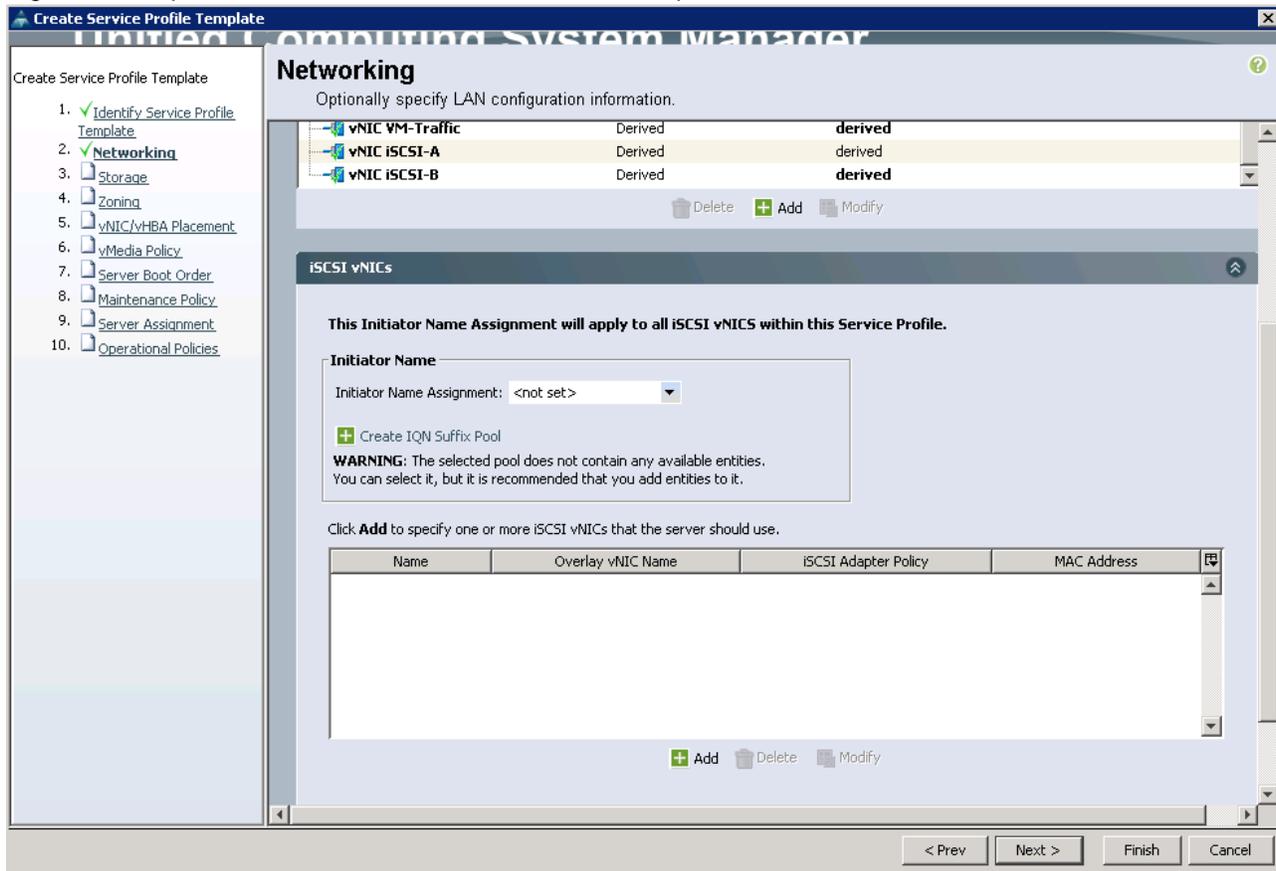
vNIC Template:

Adapter Performance Profile

Adapter Policy:

- yy. On the Networking page of the wizard, click the lower Add button in the iSCSI vNICs section to add iSCSI vNIC to the template (Figure 104).

Figure 104 OpenStack Controller Service Profile Template – iSCSI vNIC Creation



- zz. Enter iSCSI-vNIC-A as the name of vNIC (Figure 105).
- aaa. Select iSCSI-A for Overlay vNIC.
- bbb. Set the iSCSI Adapter Policy to default.
- ccc. Set the VLAN to iSCSI-40 (native).
- ddd. Leave the MAC Address set to None.
- eee. Click OK.

Figure 105 OpenStack Controller Service Profile Template – iSCSI-vNIC-A vNIC Creation

Create iSCSI vNIC

Name:

Overlay vNIC:

iSCSI Adapter Policy:

VLAN:

iSCSI MAC Address

MAC Address Assignment:

- fff. On the Networking page of the wizard, click the lower Add button in the iSCSI vNICs section to add iSCSI vNIC to the template.
- ggg. Enter iSCSI-vNIC-B as the name of vNIC (Figure 106).
- hhh. Select iSCSI-B for Overlay vNIC.
- iii. Set the iSCSI Adapter Policy to default.
- jjj. Set the VLAN to iSCSI-41 (native).
- kkk. Leave the MAC Address set to None.
- III. Click OK.

Figure 106 OpenStack Controller Service Profile Template – iSCSI-vNIC-B vNIC Creation

Create iSCSI vNIC

Name:

Overlay vNIC:

iSCSI Adapter Policy:

VLAN:

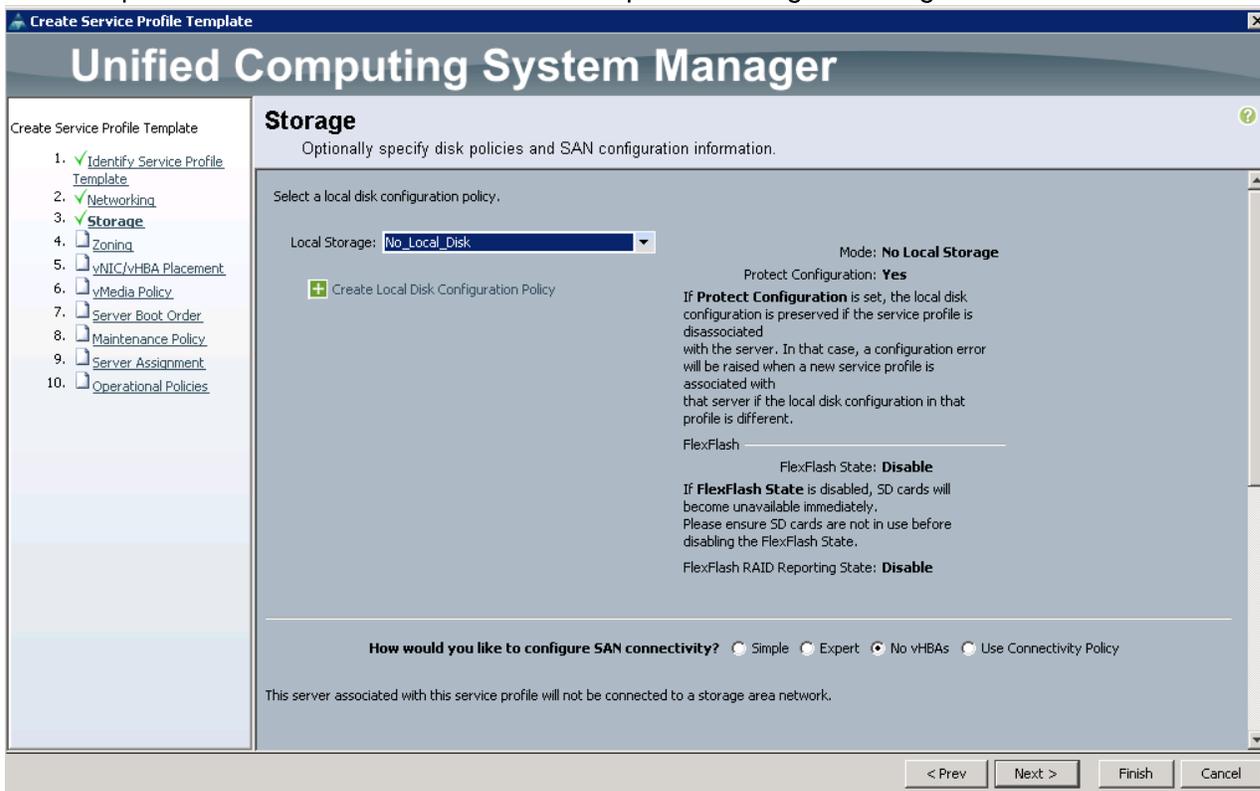
iSCSI MAC Address

MAC Address Assignment:

7. Configure the storage options:

- a. Select a local disk configuration policy:
 - o If the server in question has local disks, select default in the Local Storage list.
 - o If the server in question has no local disk, select No_Local_Disk policy in the drop-down list. This is the correct choice.
- b. Select the No vHBAs option for the “How would you like to configure SAN connectivity?” field (Figure 107).
- c. Click Next

Figure 107 OpenStack Controller Service Profile Template - Configure Storage



8. Set no Zoning options and click Next.
9. Set the vNIC/vHBA placement options (Figure 108).
 - a. In the “Select Placement” drop-down list, select the Host_Infra placement policy.
 - b. Select vCon1 and assign the vNICs to the virtual network interfaces policy in the following order:
 1. PXE
 2. iSCSI-A
 3. iSCSI-B
 4. NS-A
 5. NS-B
 6. Management
 7. VM-Traffic
 8. MCAS

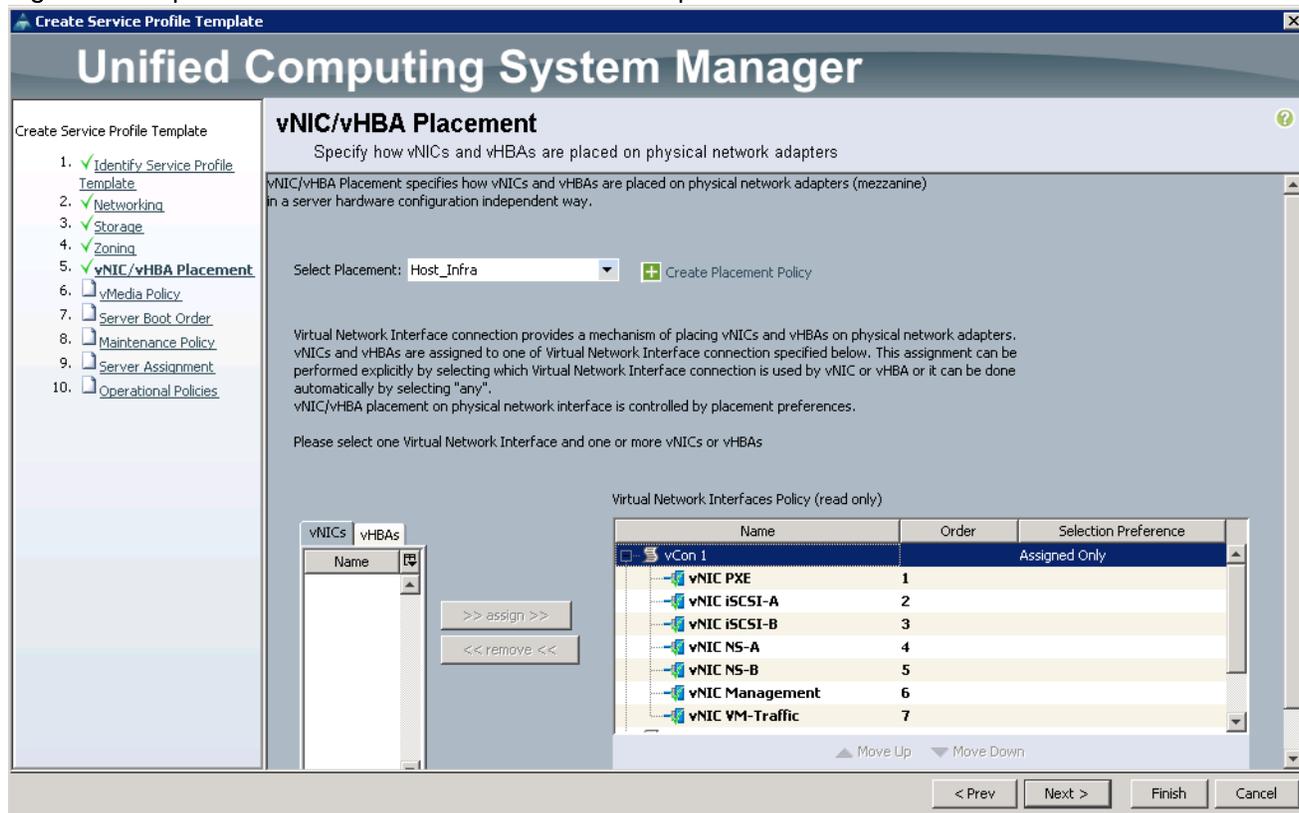


vNIC placement order is very important and is required. Not following it appropriately will result in network connectivity issues with host facing interfaces.

- c. Review the table to verify that all vNICs were assigned to the policy in the appropriate order.

- d. Click Next.

Figure 108 OpenStack Controller Service Profile Template – vNIC Placement



10. Set no vMedia Policy. Click Next.

11. Set the Server Boot Order:

- a. In the Boot Policy drop-down list, select iSCSI-Boot.
- b. In the Boot Order pane, select iSCSI-vNIC-A.
- c. Click the “Set iSCSI Boot Parameters” button.
- d. In the Initiator Name Assignment drop-down list, select IQN_Pool_A
- e. Set iSCSI_Initiator_Pool_A as the “Initiator IP address Policy”
- f. Keep the “iSCSI Static Target Interface” button selected and click the  button.
- g. Get the iSCSI target name from Table 10 .
- h. In the Create iSCSI Static Target dialog box (Figure 109), paste the iSCSI target node name from FLEXPOD-OPS-SVM1 into the iSCSI Target Name field.
- i. Enter the IP address of iscsi_lif01a for the IPv4 Address field.
- j. Click OK to add the iSCSI static target.

Figure 109 OpenStack Controller Service Profile Template – iSCSI Target

Create iSCSI Static Target

iSCSI Target Name:

Priority:

Port:

Authentication Profile: + Create iSCSI Authentication Profile

IPv4 Address:

LUN ID:

OK Cancel

- k. Keep the “iSCSI Static Target Interface” button selected and click the  button.
- l. In the Create iSCSI Static Target dialog box (Figure 110), paste the iSCSI target node name from FLEXPOD-OPS-SVM1 into the iSCSI Target Name field.
- m. Enter the IP address of iscsi_lif02a for the IPv4 Address field.
- n. Click OK to add the iSCSI static target.

Figure 110 OpenStack Controller Service Profile Template – iSCSI Target

Create iSCSI Static Target

iSCSI Target Name:

Priority:

Port:

Authentication Profile: [+ Create iSCSI Authentication Profile](#)

IPv4 Address:

LUN ID:

- o. Verify iSCSI Boot Parameters for iSCSI-vNIC-A (Figure 111).
- p. Click OK.

Figure 111 OpenStack Controller Service Profile Template – iSCSI Boot Parameters for iSCSI-vNIC-A

Name: **iSCSI-vNIC-A**

Authentication Profile: <not set> + Create iSCSI Authentication Profile

Initiator Name

Initiator Name Assignment: IQN_Pool_A(54/64) + Create IQN Suffix Pool

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

Initiator Address

Initiator IP Address Policy: ISCSI_Initiator_Pool_A(54/64)

IPv4 Address: **0.0.0.0**
Subnet Mask: **255.255.255.0**
Default Gateway: **0.0.0.0**
Primary DNS: **0.0.0.0**
Secondary DNS: **0.0.0.0**

+ Create IP Pool

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

iSCSI Static Target Interface iSCSI Auto Target Interface

Minimum one instance of iSCSI Static Target Interface and maximum two are allowed.

Name	Priority	Port	Authentication Profile	iSCSI IPv4 Address	LUN Id
iqn.1992-08.c...	1	3260		10.23.40.201	0
iqn.1992-08.c...	2	3260		10.23.40.202	0

OK Cancel

- q. In the Boot Order pane, select iSCSI-vNIC-B.
- r. Click the “Set iSCSI Boot Parameters” button.
- s. In the Initiator Name Assignment drop-down list, select IQN_Pool_B
- t. Set iSCSI_Initiator_Pool_B as the “Initiator IP address Policy”.
- u. Keep the “iSCSI Static Target Interface” button selected and click the + button.
- v. In the Create iSCSI Static Target dialog box (Figure 112), paste the iSCSI target node name from FLEXP0D-OPS-SVM1 into the iSCSI Target Name field (Same target name used in configuring iSCSI-vNIC-A).
- w. Enter the IP address of iscsi_lif02b for the IPv4 Address field.
- x. Click OK to add the iSCSI static target.

Figure 112 OpenStack Controller Service Profile Template – iSCSI Target

Create iSCSI Static Target

iSCSI Target Name:

Priority:

Port:

Authentication Profile: + Create iSCSI Authentication Profile

IPv4 Address:

LUN ID:

OK Cancel

- y. Keep the “iSCSI Static Target Interface” button selected and click the  button.
- z. In the Create iSCSI Static Target dialog box (Figure 113), paste the iSCSI target node name from FLEXPOD-OPS-SVM1 into the iSCSI Target Name field (this is the same target name used in configuring iSCSI-vNIC-A).
 - aa. Enter the IP address of iscsi_lif01b for the IPv4 Address field.
 - bb. Click OK to add the iSCSI static target.

Figure 113 OpenStack Controller Service Profile Template – iSCSI Target

Create iSCSI Static Target

iSCSI Target Name:

Priority:

Port:

Authentication Profile: [+ Create iSCSI Authentication Profile](#)

IPv4 Address:

LUN ID:

- cc. Verify iSCSI Boot Parameters for iSCSI-vNIC-B (Figure 114).
- dd. Click OK.
- ee. Click Next to continue to the next section.

Figure 114 OpenStack Controller Service Profile Template – iSCSI Boot Parameters for iSCSI-vNIC-B

Name: **iSCSI-vNIC-B**

Authentication Profile: <not set> + Create iSCSI Authentication Profile

Initiator Name

Initiator Name Assignment: IQN_Pool_B(54/64) + Create IQN Suffix Pool

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

Initiator Address

Initiator IP Address Policy: iSCSI_Initiator_Pool_B(54/64) + Create IP Pool

IPv4 Address: **0.0.0.0**
Subnet Mask: **255.255.255.0**
Default Gateway: **0.0.0.0**
Primary DNS: **0.0.0.0**
Secondary DNS: **0.0.0.0**

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

iSCSI Static Target Interface iSCSI Auto Target Interface

Minimum one instance of iSCSI Static Target Interface and maximum two are allowed.

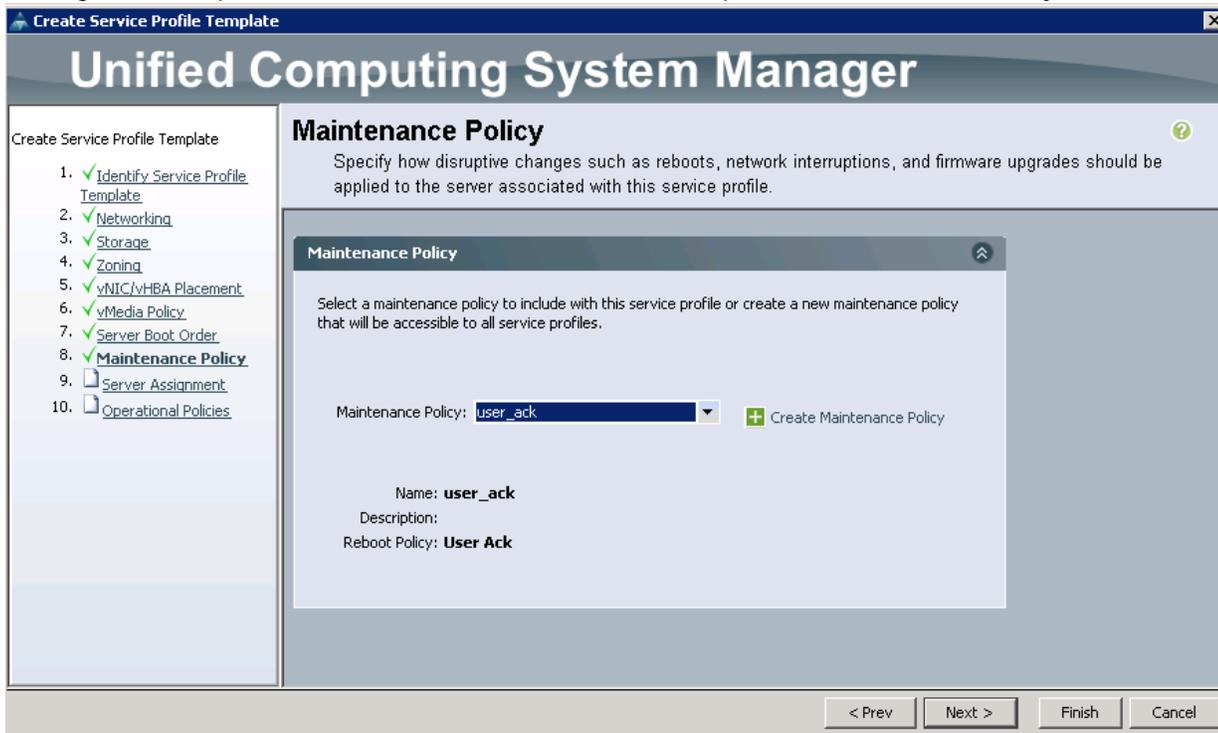
Name	Priority	Port	Authentication Profile	iSCSI IPv4 Address	LUN Id
iqn.1992-08.c...	1	3260		10.23.41.202	0
iqn.1992-08.c...	2	3260		10.23.41.201	0

OK Cancel

12. Add a maintenance policy:

- a. In the Maintenance Policy drop-down list, select user_ack maintenance policy (Figure 115).
- b. Click Next.

Figure 115 OpenStack Controller Service Profile Template – Maintenance Policy



13. Specify the Server Assignment:

- a. In the Pool Assignment drop-down list, select OpenStack_Controller_Pool (Figure 116).
- b. Leave the power state to “Up”.
- c. Click Next.

Figure 116 OpenStack Controller Service Profile Template – Server Assignment

Unified Computing System Manager

Create Service Profile Template

Server Assignment

Optionally specify a server pool for this service profile template.

You can select a server pool you want to associate with this service profile template.

Pool Assignment: **Openstack_Controller_Pool** + Create Server Pool

Select the power state to be applied when this profile is associated with the server.

Up Down

The service profile template will be associated with one of the servers in the selected pool. If desired, you can specify an additional server pool policy qualification that the selected server must meet. To do so, select the qualification from the list.

Server Pool Qualification: **<not set>**

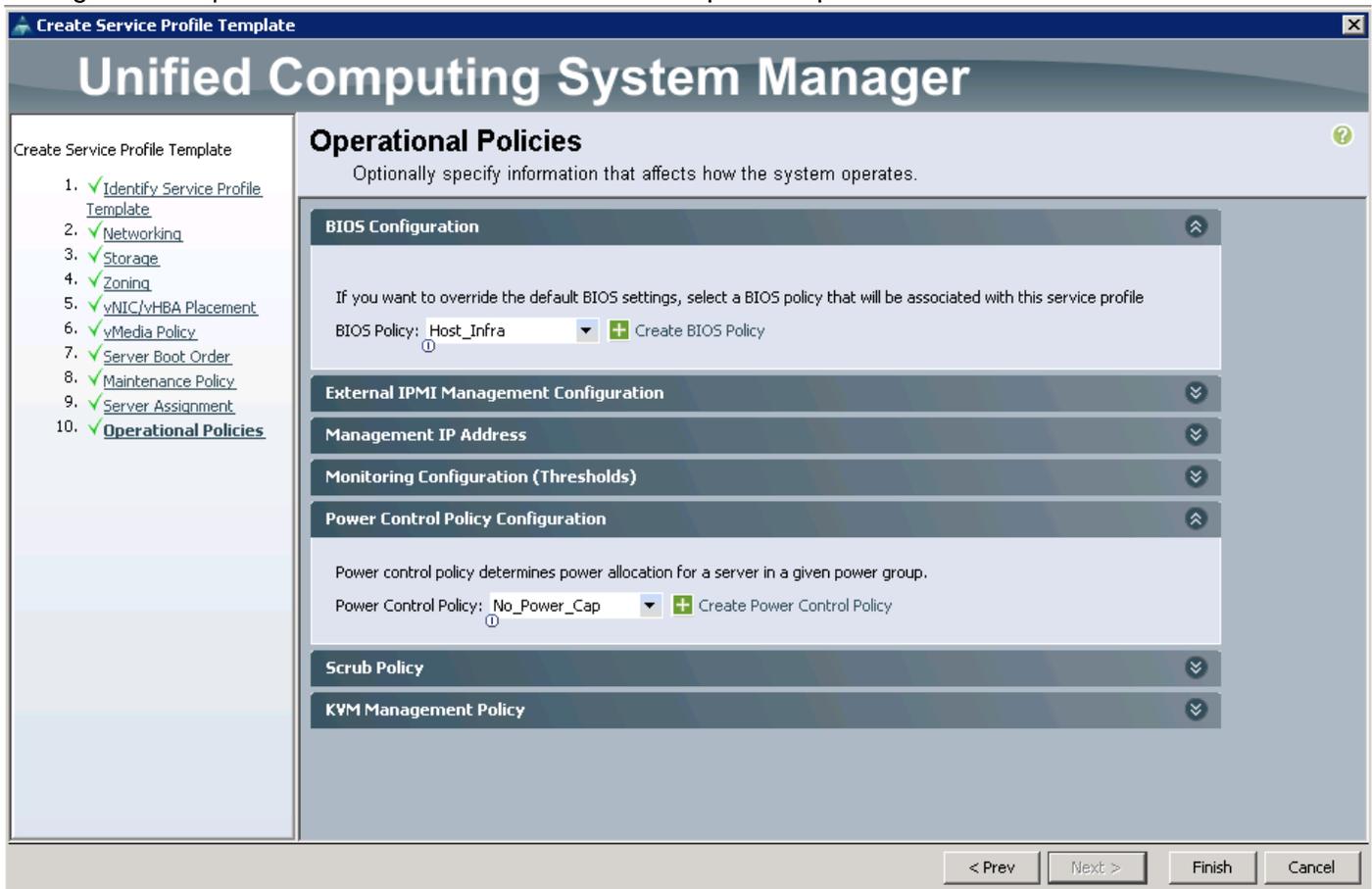
Restrict Migration:

< Prev Next > Finish Cancel

14. Specify Operational Policies (Figure 117):

- a. In the BIOS Policy drop-down list, select Host_Infra.
- b. Expand Power Control Policy Configuration and select No_Power_Cap in the Power Control Policy list.

Figure 117 OpenStack Controller Service Profile Template – Operational Policies



15. Click Finish to create the service profile template.

16. Click OK in the confirmation message.

Service Profile Templates for OpenStack Compute Hosts

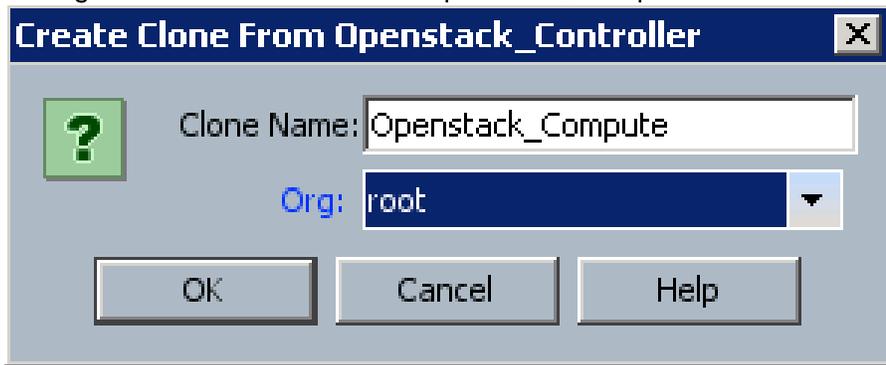
OpenStack Compute hosts service profile template can be quickly created by cloning the Openstack_Controller service profile template and later modify the server pool to use Openstack_Compute_Pool.

To create a service profile template for OpenStack compute hosts, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Service Profile Templates > root > Service Template Openstack_Controller.
3. Right-click Service Template Openstack_Controller.
4. Click “Create a Clone”.
5. In the Create Clone From Openstack_Controller dialog box, Enter Openstack_Compute in the Clone Name field.
6. In the Org drop-down list, select root (Figure 118).

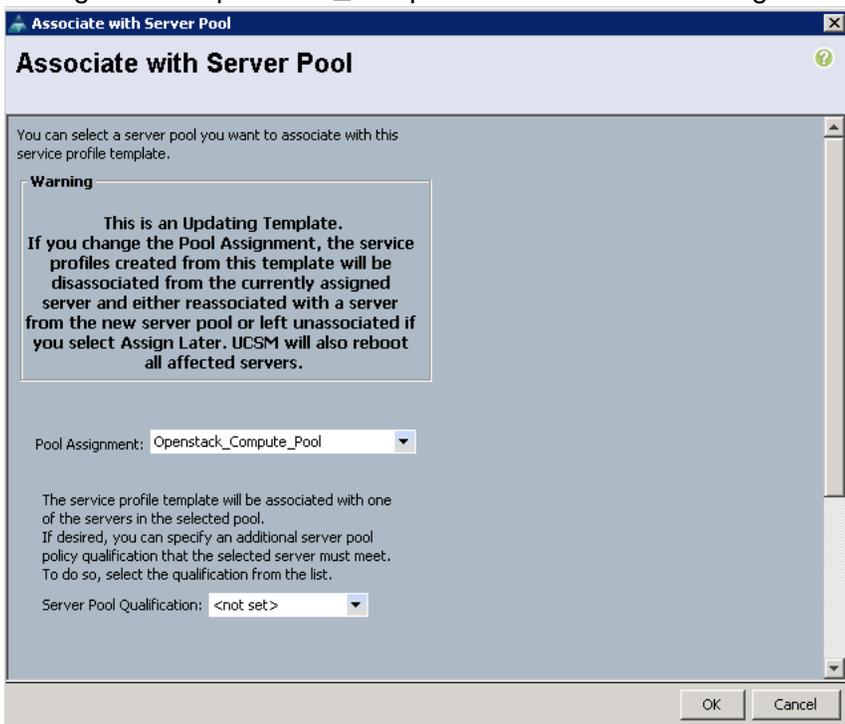
7. Click OK.
8. Click OK in the confirmation message.

Figure 118 Create a clone of OpenStack_Compute Service Profile Template



9. Select Servers > Service Profile Templates > root > Service Template Openstack_Compute.
10. Right-click Service Template Openstack_Compute, select Associate with Server Pool.

Figure 119 OpenStack_Compute Service Profile – Change Server Pool



11. In the Pool Assignment drop-down list, select Openstack_Compute_Pool (Figure 119).
12. Click OK.
13. Click OK in the confirmation message.

Service Profile Templates for RHEL-OSP Installer Server

The RHEL-OSP Installer server has different requirements with respect to vNICs compared to either of the **Controller and Compute hosts**. We can start by cloning the Installer's service profile template either from Openstack_Controller or Openstack_Compute template, and later modify the vNICs, vNIC placement, and Server Pool.

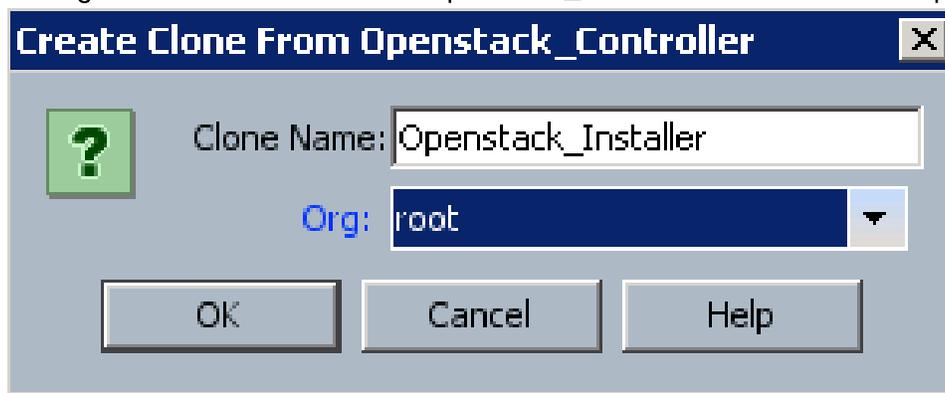
The RHEL-OSP Installer needs the following vNICs

- External
- Management
- PXE
- iSCSI-A
- iSCSI-B

To create a service profile template for the RHEL-OSP Installer server, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Service Profile Templates > root > Service Template Openstack_Controller.
3. Right-click Service Template Openstack_Controller.
4. Click **“Create a Clone”**.
5. In the Create Clone From Openstack_Controller dialog box, Enter Openstack_Installer in the Clone Name field.
6. In the Org drop-down list, select root (Figure 120).
7. Click OK.
8. Click OK in the confirmation message.

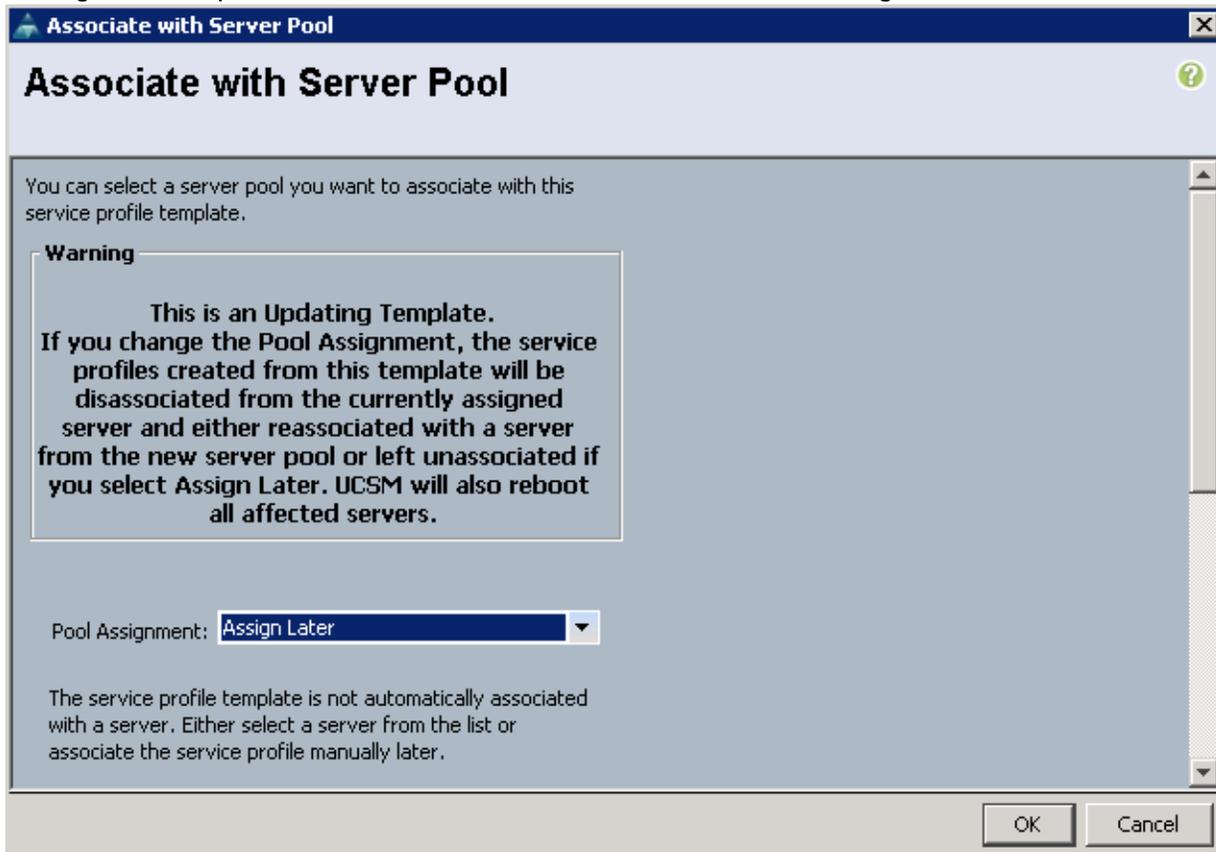
Figure 120 Create a clone of OpenStack_Installer Service Profile Template From Openstack_Controller



9. Select Servers > Service Profile Templates > root > Service Template Openstack_Installer.
10. Right click Service Template Openstack_Installer, select Associate with Server Pool

11. In the Pool Assignment drop-down list, select Assign Later (Figure 121).
12. Click OK.
13. Click OK in the confirmation message.

Figure 121 OpenStack_Installer Service Profile – Server Pool Assignment



14. Select Servers > Service Profile Templates > root > Service Template Openstack_Installer, Click Network Tab
15. Select MCAS in the vNIC section (Figure 122)
16. Click Delete button .
17. Click “Yes” on “Are you sure you want to delete vNIC MCAS?” dialog box.
18. Select NS-A in the vNIC section.
19. Click Delete button .
20. Click “Yes” on “Are you sure you want to delete vNIC NS-A?” dialog box.

Figure 122 Delete vNICs Not Required for RHEL-OSP Installer

The screenshot shows the vNIC configuration interface. At the top, there are tabs for General, Storage, Network, iSCSI vNICs, vMedia Policy, Boot Order, Policies, Events, and FSM. Below the tabs is a table for Virtual Slot and Selection Preference:

Virtual Slot	Selection Preference
1	Assigned Only
2	All
3	All
4	All

Below this table is the LAN Connectivity Policy section, which includes a dropdown menu for LAN Connectivity Policy (set to <not set>) and a button to Create LAN Connectivity Policy.

The vNICs section is a table with the following data:

Name	MAC Address	Desired Order	Actual Order	Fabric ID	Desired Placement	Actual Placement
vNIC MCAS	Derived	12	Unspecified	A B	1	Any
vNIC Management	Derived	10	Unspecified	A B	1	Any
vNIC NS-A	Derived	4	Unspecified	A	1	Any
vNIC NS-B	Derived	5	Unspecified	B	1	Any
vNIC PXE	Derived	1	Unspecified	A B	1	Any
vNIC VM-Traffic	Derived	11	Unspecified	B A	1	Any
vNIC iSCSI-A	Derived	2	Unspecified	A	1	Any
vNIC iSCSI-B	Derived	3	Unspecified	B	1	Any

At the bottom of the vNICs section, there are buttons for Delete, Add, and Modify.

21. Select NS-B in the vNIC section.

22. Click Delete button 

23. Click “Yes” on “Are you sure you want to delete vNIC NS-B?” dialog box.

24. Select VM-Traffic in the vNIC section.

25. Click Delete button 

26. Click “Yes” on “Are you sure you want to delete vNIC VM-Traffic?” dialog box.

27. Click Add  button to add External vNIC.

28. Enter “External” in the Name field of Create vNIC dialog box (Figure 123).

Figure 123 Add External vNIC to RHEL-OSP Installer Service Profile Template

Create vNIC

Name:

Use vNIC Template:

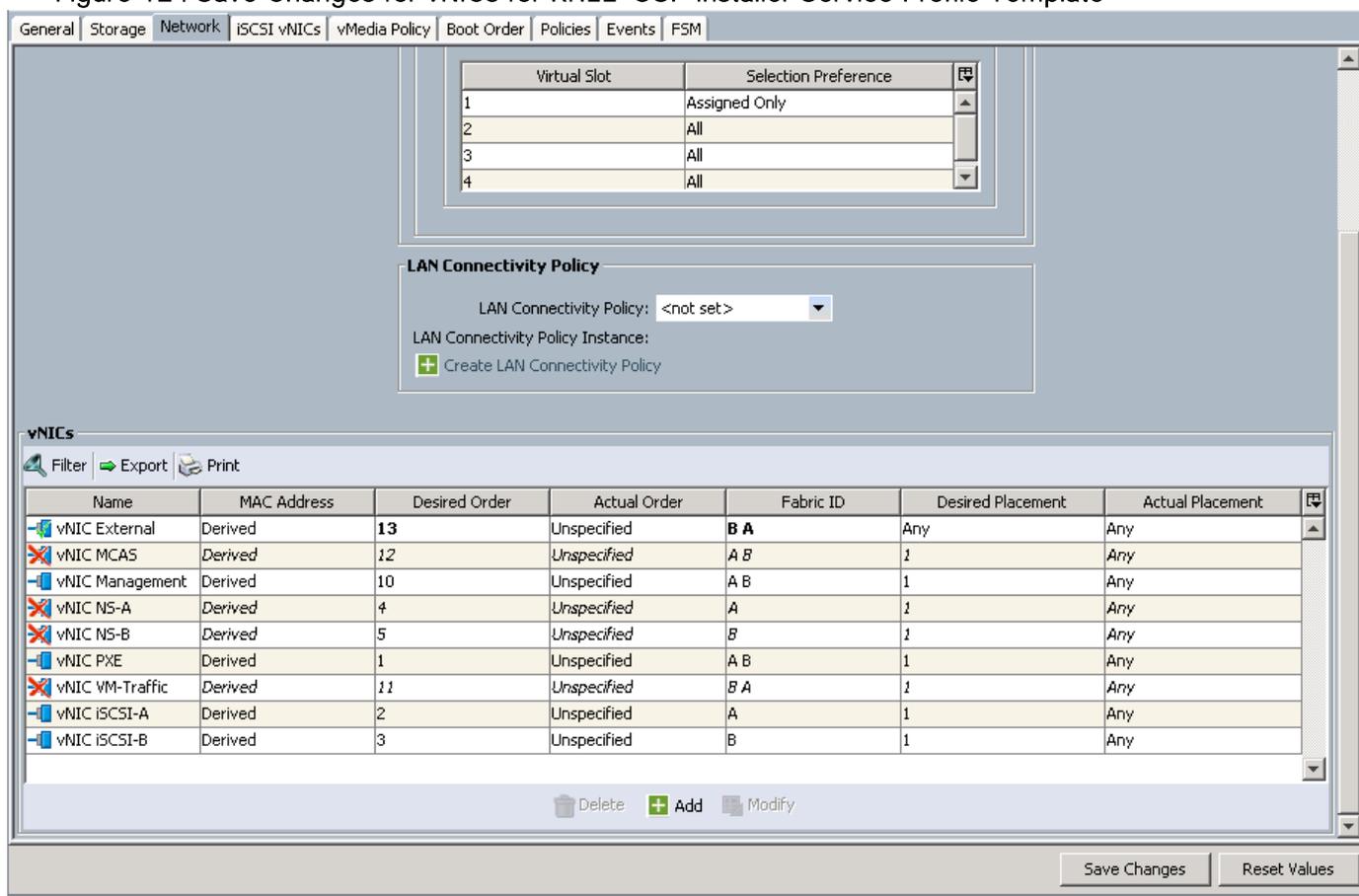
vNIC Template:

Adapter Performance Profile

Adapter Policy:

29. Select the Use vNIC Template checkbox.
30. In the vNIC Template list, select Ext_Template.
31. In the Adapter Policy list, select Linux.
32. Click OK to add this vNIC.
33. Click Save Changes (Figure 124).
34. Click OK on the confirmation.

Figure 124 Save Changes for vNICs for RHEL-OSP Installer Service Profile Template



35. In the Network Tab click Modify vNIC/vHBA Placement in the Actions panel.

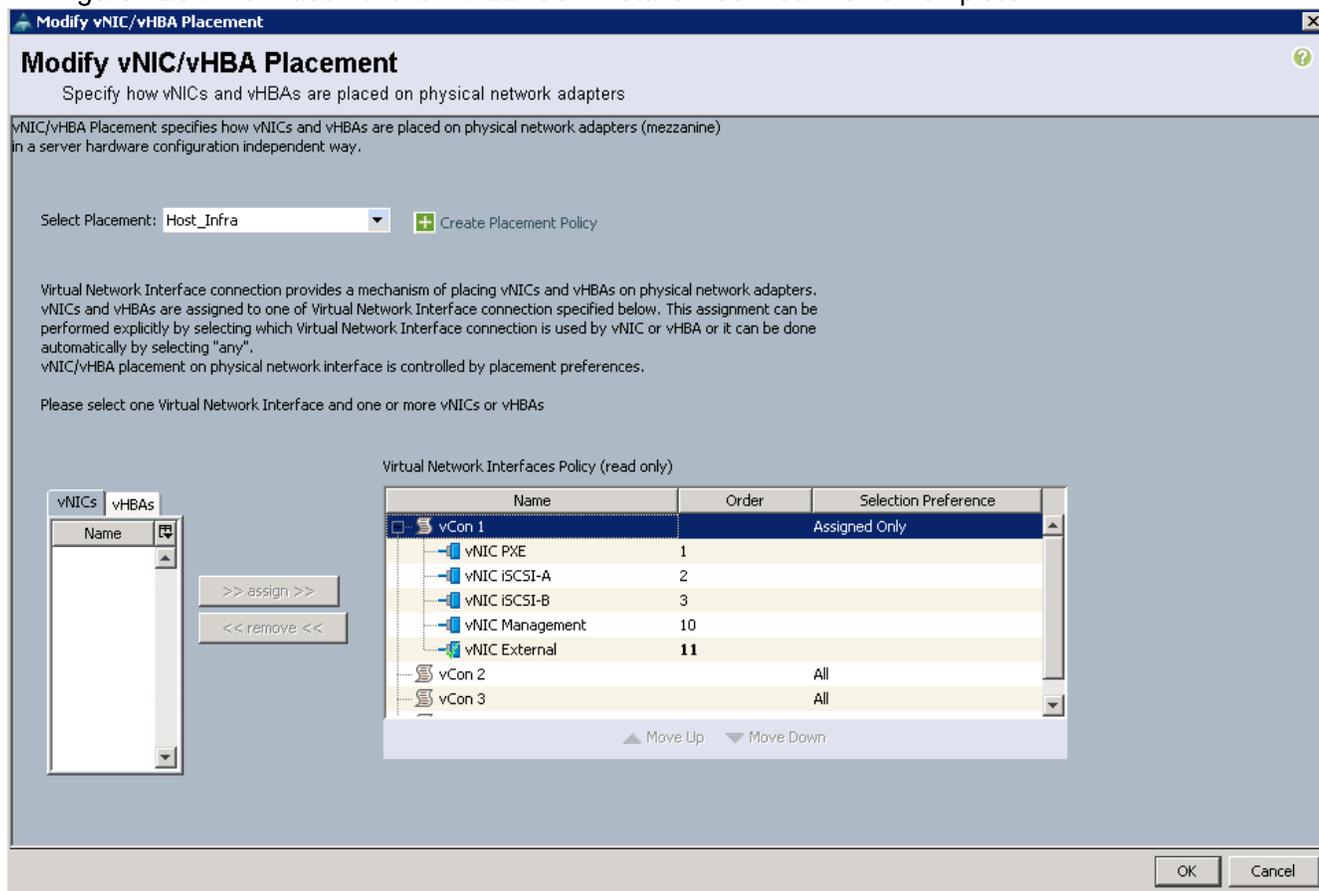
36. Select External in vNICs and select vCon 1 in Virtual Network Interface Policy, click >> Assign >> (Figure 125). Make sure vNIC placement matches to the following order:

- a. PXE
- b. iSCSI-A
- c. iSCSI-B
- d. Management
- e. External

37. Click OK.

38. Click OK on the confirmation.

Figure 125 vNIC Placement for RHEL-OSP Installer Service Profile Template



Verify vNIC Placement

To verify vNIC placement, complete the following steps:

1. Click Server tab in Cisco UCS Manager.
2. Select Servers > Service Profile Templates > root > Service Template Openstack_Controller
3. Select the Network Tab. Sort it by Desired Order by clicking the header.
4. Verify the vNIC Placement for controllers as shown in the Figure (126)

Figure 126 vNIC Placement for Controllers

vNICs				
Filter	Export	Print		
Name	MAC Address	Desired Order ▲	Actual Order	Fabric ID
vNIC PXE	Derived	1	Unspecified	A B
vNIC iSCSI-A	Derived	2	Unspecified	A
vNIC iSCSI-B	Derived	3	Unspecified	B
vNIC NS-A	Derived	4	Unspecified	A
vNIC NS-B	Derived	5	Unspecified	B
vNIC Management	Derived	10	Unspecified	A B
vNIC VM-Traffic	Derived	11	Unspecified	B A
vNIC MCAS	Derived	12	Unspecified	A B

5. Select Servers > Service Profile Templates > root > Service Template Openstack_Compute
6. Select the Network tab. Sort it by desired order.
7. Verify vNIC placement for compute hosts as shown in Figure (127)

Figure 127 vNIC Placement for Compute Hosts

vNICs				
Filter	Export	Print		
Name	MAC Address	Desired Order ▲	Actual Order	Fabric ID
vNIC PXE	Derived	1	Unspecified	A B
vNIC iSCSI-A	Derived	2	Unspecified	A
vNIC iSCSI-B	Derived	3	Unspecified	B
vNIC NS-A	Derived	4	Unspecified	A
vNIC NS-B	Derived	5	Unspecified	B
vNIC Management	Derived	7	Unspecified	A B
vNIC VM-Traffic	Derived	8	Unspecified	B A
vNIC MCAS	Derived	9	Unspecified	A B

8. Select Servers > Service Profile Templates > root > Service Template Openstack_Installer
9. Select the Network tab. Sort it by desired order.
10. Verify vNIC placement for installer hosts as shown in Figure (128)

Figure 128 vNIC Placement for Installer Host

vNICs				
Filter	Export	Print		
Name	MAC Address	Desired Order ▲	Actual Order	Fabric ID
vNIC PXE	Derived	1	Unspecified	A B
vNIC iSCSI-A	Derived	2	Unspecified	A
vNIC iSCSI-B	Derived	3	Unspecified	B
vNIC Management	Derived	10	Unspecified	A B
vNIC External	Derived	11	Unspecified	B A

Create Service Profile

The following service profiles will be created next:

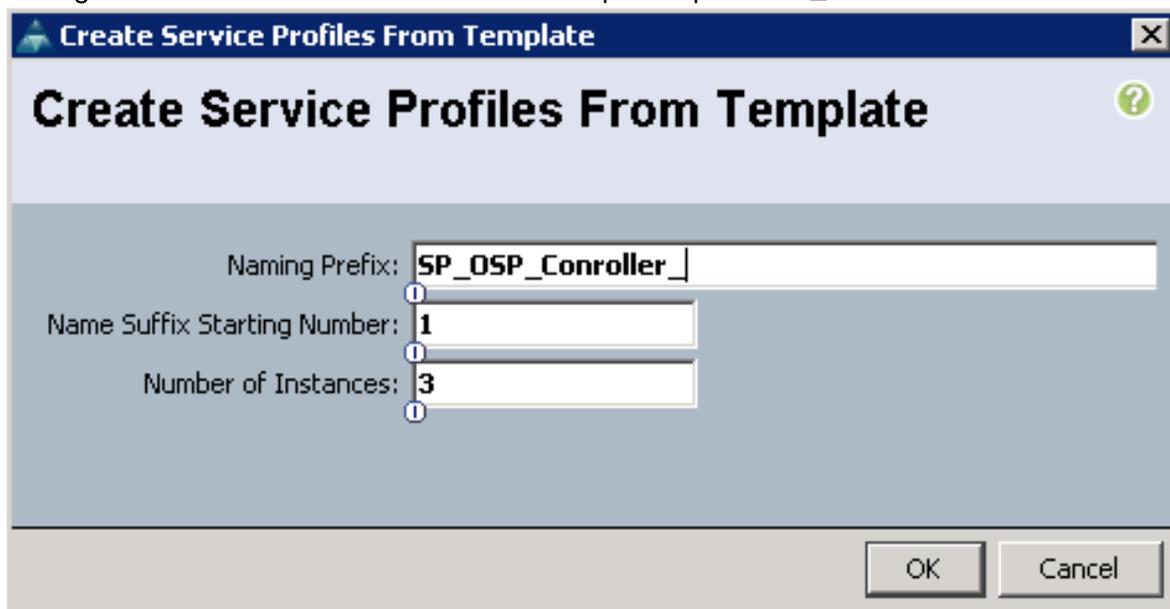
- Service Profile for Controller nodes from OpenStack_Controller service profile template.
- Service Profile for Compute nodes from OpenStack_Compute service profile template.
- Service Profile for the RHEL-OSP Installer.

Service Profile for Controller Nodes

To create service profiles from the service profile template, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Service Profile Templates > root > Service Template Openstack_Controller.
3. Right-click Openstack_Controller and select Create Service Profiles from Template (Figure 129).
4. Enter SP_OSP_Controller_ as the service profile prefix.
5. Enter 1 as “Name Suffix Starting Number”.
6. Enter 3 as the “Number of Instances”.
7. Click OK to create the service profile

Figure 129 Create Service Profile from Template OpenStack_Controller



8. Click OK in the confirmation message.

Service Profile for Compute Nodes

To create service profiles from the service profile template, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Service Profile Templates > root > Service Template Openstack_Compute.
3. Right-click Openstack_Compute and select Create Service Profiles from Template (Figure 130).
4. Enter SP_OSP_Compute_ as the service profile prefix.
5. Enter 1 as “Name Suffix Starting Number”
6. Enter 6 as the “Number of Instances”.



In this validation, six hosts are dedicated for OpenStack Compute hosts. Enter number of compute hosts on step 6 depending on your environment.

7. Click OK to create the service profile.

Figure 130 Create Service Profile from Template OpenStack_Compute

8. Click OK in the confirmation message.

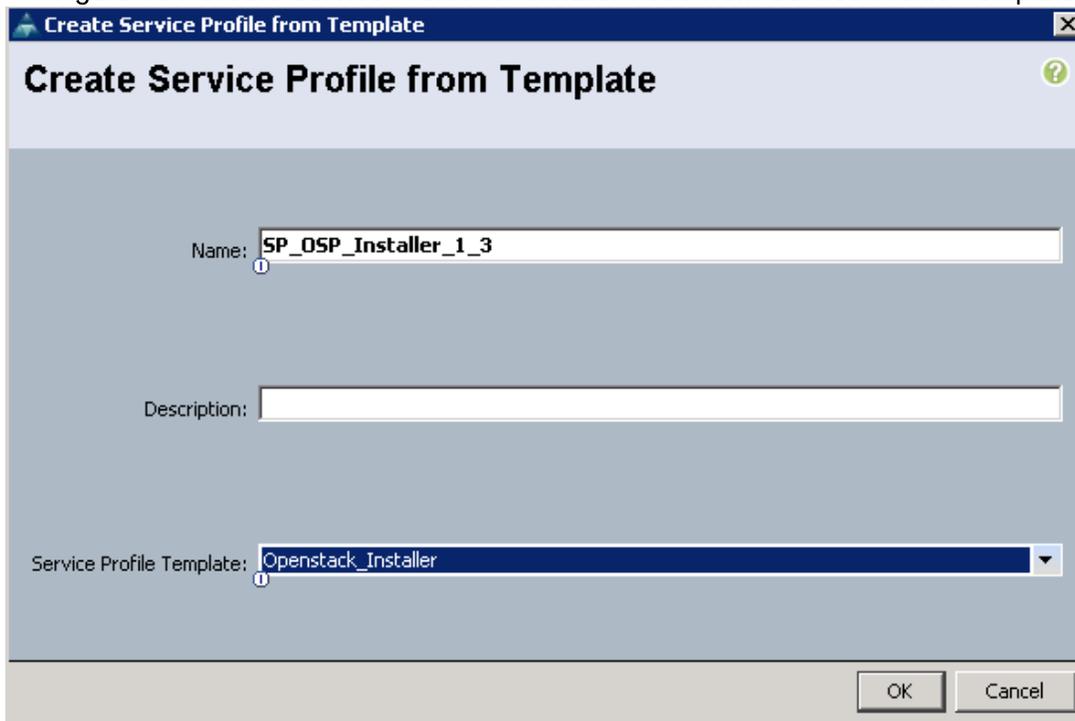
Service Profile for RHEL-OSP Installer Server

To create a service profile for RHEL-OSP, complete the following steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Servers > Service Profiles
3. Right-click and select “Create Service Profile from Template” (Figure 131).
4. Enter OSP_Installer_1_3 in the Name field.

5. Enter Description (Optional).
6. Select Openstack_Installer from Service Profile Template drop-down list.
7. Click OK to create the service profile.
8. Click OK on the confirmation message.

Figure 131 Create Service Profile for Installer node from Service Profile Template



9. Right-click Servers > Service Profiles > root > SP_OSP_Installer_1_3 service profile.
10. Click “Change Service Profile Association”.
11. Select existing Server in Server Assignment drop-down list.
12. Select Chassis ID 1 Slot 3 server in Available Servers.
13. Click OK.
14. Click OK on the confirmation message.

Gather Necessary Information from Cisco UCS Manager

After the Cisco UCS service profiles have been created, each infrastructure blade in the environment will have a unique configuration. To proceed with the FlexPod deployment, specific information must be gathered from each Cisco UCS blade. Insert the required information in Table 13 .

Table 13 iSCSI Initiator Name for Fabric A and Fabric B

Cisco UCS Service Profile Name	iSCSI Initiator Name	Name

SP_OSP_Installer_1_3	iqn.1992-08.com.cisco:fabric-a:19	iSCSI-vNIC-A
	iqn.1992-08.com.cisco:fabric-b:19	iSCSI-vNIC-B



To gather the iSCSI initiator information, launch the Cisco UCS Manager GUI. In the navigation pane, click Servers > Service Profiles > root > SP_OSP_Installer_1_3. In the General Tab of right hand pane, write down the Associated Server in the Properties panel. Click Equipment Tab of navigation pane, then click Equipment > Chassis > Chassis 1 > Server 3 > Adapters > Adapter 1 > iSCSI vNICs > iSCSI 2. Write down the Initiator Name and Name in the General Tab of right hand pane in Table 6. Capture the iSCSI initiator of Fabric B iSCSI vNIC.



Repeat the above steps for all other servers in the infrastructure and fill in the Table 13 for iSCSI initiator name. This information is required to setup Boot LUN in the storage controller.

Network Configuration

Before you begin the installation of Cisco Nexus 9000 series switches, review the configuration worksheet below.

Table 14 Configuration Variable for Cisco Nexus 9000 Switches

Variable	Description	Implementation Value (Examples)
<<var_nexus_A_hostname>>	Cisco Nexus A host name	N9k-FLEXPOD-SwitchA
<<var_nexus_A_mgmt0_ip>>	Out-of-band Nexus A management IP address	10.23.10.3
<<var_nexus_A_mgmt0_netmask>>	Out-of-band Nexus A management IP address network mask	255.255.255.0
<<var_nexus_A_mgmt0_gw>>	Out-of-band Nexus A management network default gateway	10.23.10.1
<<var_nexus_B_hostname>>	Cisco Nexus B host name	N9k-FLEXPOD-SwitchB
<<var_nexus_B_mgmt0_ip>>	Out-of-band Nexus B management IP address	10.23.10.4
<<var_nexus_B_mgmt0_netmask>>	Out-of-band Nexus B management IP address network mask	255.255.255.0
<<var_nexus_B_mgmt0_gw>>	Out-of-band Nexus B management network default gateway	10.23.10.1
<<var_global_ntp_server_ip>>	NTP server IP address	
<<var_nexus_vpc_domain_id>>	Unique Cisco Nexus switch VPC Domain ID	1

Cisco Nexus 9000 Network Initial Configuration Setup

These steps provide the details for Initial Cisco Nexus 9000 Switch setup.

Cisco Nexus A

To set up the initial configuration for the second Cisco Nexus switch, complete the following steps:



On initial boot and connection to the serial or console port of the switch, the NX-OS setup should automatically start and attempt to enter Power on Auto Provisioning.

```
---- 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 : <<var_nexus_A_hostname>>
```

```
Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]:
```

Mgmt0 IPv4 address : <<var_nexus_A_mgmt0_ip>>

Mgmt0 IPv4 netmask : <<var_nexus_A_mgmt0_netmask>>

Configure the default gateway? (yes/no) [y]:

IPv4 address of the default gateway : <<var_nexus_A_mgmt0_gw>>

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 : <<var_global_ntp_server_ip>>

Configure CoPP system profile (strict/moderate/lenient/dense/skip) [strict]:

The following configuration will be applied:

password strength-check

switchname <<var_nexus_A_hostname>>

vrf context management

ip route 0.0.0.0/0 <<var_nexus_A_mgmt0_gw>>

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 <<var_nexus_A_mgmt0_ip>> <<var_nexus_A_mgmt0_netmask>>

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.

Cisco Nexus 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 : <<var_nexus_B_hostname>>

Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]:

Mgmt0 IPv4 address : <<var_nexus_B_mgmt0_ip>>

Mgmt0 IPv4 netmask : <<var_nexus_B_mgmt0_netmask>>

Configure the default gateway? (yes/no) [y]:

IPv4 address of the default gateway : <<var_nexus_B_mgmt0_gw>>

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 : <<var_global_ntp_server_ip>>
```

```
Configure CoPP system profile (strict/moderate/lenient/dense/skip) [strict]:
```

The following configuration will be applied:

```
password strength-check
switchname <<var_nexus_B_hostname>>
vrf context management
ip route 0.0.0.0/0 <<var_nexus_A_mgmt0_gw>>
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 <<var_nexus_B_mgmt0_ip>> <<var_nexus_B_mgmt0_netmask>>
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 Appropriate Cisco Nexus 9000 Features and Settings

Cisco Nexus 9000 A and Cisco Nexus 9000 B

The following commands enable IP switching feature and set default spanning tree behaviors:

1. On each Nexus 9000, enter configuration mode:

```
config terminal
```

2. Use the following commands to enable the necessary features:

```
feature udld
feature interface-vlan
feature lacp
feature vpc
```

3. Configure spanning tree defaults:

```
spanning-tree port type network default
spanning-tree port type edge bpduguard default
spanning-tree port type edge bpdufilter default
```

4. Save the running configuration to start-up:

```
copy run start
```

Create VLAN

To create the necessary VLANs, complete the following steps. These steps need to be performed both in Cisco Nexus A and Cisco Nexus B switches.

Cisco Nexus A and B

1. Create Management VLAN

```
vlan <<var_mgmt_vlan_id>>
name MGMT-VLAN
```

2. Create PXE VLAN

```
vlan <<var_pxe_vlan_id>>
name PXE
```

3. Create VLAN for OpenStack Management, Cluster Management, Admin API, Storage Clustering

```
vlan <<var_mcas_vlan_id>>
name MCAS
```

4. Create OpenStack storage VLAN for carrying Cinder and Glance traffic.

```
vlan <<var_nfs_vlan_id>>
name NS-VLAN
```

5. Create iSCSI VLAN for Fabric A and Fabric B

```
vlan <<var_iscsi_A_vlan_id>>
name iSCSI-VLAN-A
vlan <<var_iscsi_B_vlan_id>>
name iSCSI-VLAN-B
```

6. Create VLAN for OpenStack Swift traffic

```
vlan <<var_swift_A_vlan_id>>
name Swift-A
vlan <<var_swift_B_vlan_id>>
name Swift-B
```

7. Create VLAN for external or public access

```
vlan <<var_external_vlan_id>>
name External
```

8. Create provider VLANs

```
vlan <<var_provider_vlan_range>>
```

9. Create Tenant internal VLANs

```
vlan <<var_tenant_vlan_range>>
```



Provider and Tenant internal VLAN are pre-provisioned in this solution. More provider and tenant VLANs can be created depending on individual customer requirements.

10. Save the running configuration to the startup configuration:

```
copy run start
```

Configure Virtual Port Channel Domain

Cisco Nexus 9000 A

To configure Virtual Port Channel (vPC) for switch A, complete the following steps:

1. From the global configuration mode, create a new vPC domain:

```
vpc domain <<var_nexus_vpc_domain_id>>
```

2. Make Cisco Nexus 9000A the primary vPC peer by defining a low priority value:

```
role priority 10
```

3. Use the management interfaces on the supervisors of the Nexus 9000s to establish a keepalive link:

```
peer-keepalive destination <<var_nexus_B_mgmt0_ip>> source
<<var_nexus_A_mgmt0_ip>>
```

4. Enable the following features for this vPC domain:

```
peer-switch
delay restore 150
peer-gateway
auto-recovery
```

Cisco Nexus 9000 B

To configure Virtual Port Channel (vPC) for switch A, complete the following steps:

1. From the global configuration mode, create a new vPC domain:

```
vpc domain <<var_nexus_vpc_domain_id>>
```

2. Make Cisco Nexus 9000A the primary vPC peer by defining a low priority value:

```
role priority 20
```

3. Use the management interfaces on the supervisors of the Nexus 9000s to establish a keepalive link:

```
peer-keepalive destination <<var_nexus_A_mgmt0_ip>>
source <<var_nexus_B_mgmt0_ip>>
```

4. Enable the following features for this vPC domain:

```
peer-switch
delay restore 150
peer-gateway
auto-recovery
```

Configure Network Interfaces for vPC Peer Link

Cisco Nexus 9000 A

1. Define a port description for the interfaces connecting to VPC Peer N9k-FLEXPOD-SwitchB.

```
interface Ethernet1/1
description vPC Peer N9k-FLEXPOD-SwitchB:1/1
```

```
interface Ethernet1/2
description vPC Peer N9k-FLEXPOD-SwitchB:1/2
```

2. Apply a port channel to both VPC Peer links and bring up the interfaces.

```
interface Ethernet1/1, Ethernet1/2
channel-group 10 mode active
no shutdown
```

3. Define a description for the port-channel connecting to N9k-FLEXPOD-SwitchB

```
interface po10
description vPC Peer-Link
```

4. Make the port-channel a switchport, and configure a trunk to allow in-band management, PXE, NFS, Swift, iSCSI, Tenant, and External VLAN.

```
switchport
switchport mode trunk
switchport trunk allowed vlan <<var_mgmt_vlan_id>>,
<<var_pxe_vlan_id>>, <<var_mcas_vlan_id>>, <<var_nfs_vlan_id>>,
<<var_iscsi_A_vlan_id>>, <<var_iscsi_B_vlan_id>>, <<var_swift_A_vlan_id>>, <<var_swift_B_vlan_id>>, <<var_provider_vlan_range>>, <<var_tenant_vlan_range>>,
<var_external_vlan_id>>
spanning-tree port type network
```

5. Make this port-channel the VPC peer link and bring it up.

```
vpc peer-link
no shutdown
```

Cisco Nexus 9000 B

1. Define a port description for the interfaces connecting to VPC Peer N9k-FLEXPOD-SwitchA.

```
interface Ethernet1/1
description vPC Peer N9k-FLEXPOD-SwitchA:1/1
```

```
interface Ethernet1/2
description vPC Peer N9k-FLEXPOD-SwitchA:1/2
```

2. Apply a port channel to both VPC Peer links and bring up the interfaces.

```
interface Ethernet1/1, Ethernet1/2
channel-group 10 mode active
no shutdown
```

3. Define a description for the port-channel connecting to N9k-FLEXPOD-SwitchA

```
interface po10
description vPC Peer-Link
```

4. Make the port-channel a switchport, and configure a trunk to allow in-band management, PXE, NFS, Swift, iSCSI, Provider, Tenant, and External VLAN.

```
switchport
switchport mode trunk
switchport trunk allowed vlan <<var_mgmt_vlan_id>>, <<var_pxe_vlan_id>>,
<<var_mcas_vlan_id>>, <<var_nfs_vlan_id>>,
<<var_iscsi_A_vlan_id>>, <<var_iscsi_B_vlan_id>>, <<var_swift_A_vlan_id>>, <<var_swift_B_vlan_id>>,
<<var_provider_vlan_range>>, <<var_tenant_vlan_range>>,
<var_external_vlan_id>>
spanning-tree port type network
```

5. Make this port-channel the VPC peer link and bring it up.

```
vpc peer-link
no shutdown
```

Configure Network Interfaces Connected to Fabric Interconnect

Cisco Nexus 9000 A

1. Define a description for the port-channel connecting to UCS-FLEXPOD-FAB-A.

```
interface po17
description UCS-FLEXPOD-FAB-A
```

2. Make the port-channel a switchport, and configure a trunk to allow in-band management, PXE, NFS, Swift, iSCSI, Provider, Tenant, and External VLAN.

```
switchport
switchport mode trunk
switchport trunk allowed vlan <<var_mgmt_vlan_id>>, <<var_pxe_vlan_id>>,
<<var_mcas_vlan_id>>, <<var_nfs_vlan_id>>,
<<var_iscsi_A_vlan_id>>,<<var_iscsi_B_vlan_id>>,<<var_swift_A_vlan_id>>,<<var_swift_B_vlan_id>>, <<var_provider_vlan_range>>,<<var_tenant_vlan_range>>,
<var_external_vlan_id>>
```

3. Make the port channel and associated interfaces spanning tree edge ports.

```
spanning-tree port type edge trunk
```

4. Set the MTU to be 9216 to support jumbo frames.

```
mtu 9216
```

5. Make this a VPC port-channel and bring it up.

```
vpc 17
no shutdown
```

6. Define a port description for the interface connecting to UCS-FLEXPOD-FAB-A.

```
interface Eth1/17
description UCS-FLEXPOD-FAB-A:1/17
```

7. Apply it to a port channel and bring up the interface.

```
channel-group 17 mode active
no shutdown
```

8. Define a description for the port-channel connecting to UCS-FLEXPOD-FAB-B.

```
interface po18
description UCS-FLEXPOD-FAB-B
```

9. Make the port-channel a switchport, and configure a trunk to allow in-band management, PXE, NFS, Swift, iSCSI, Provider, Tenant, and External VLAN.

```
switchport
switchport mode trunk
switchport trunk allowed vlan <<var_mgmt_vlan_id>>, <<var_pxe_vlan_id>>,
<<var_mcas_vlan_id>>, <<var_nfs_vlan_id>>,
<<var_iscsi_A_vlan_id>>,<<var_iscsi_B_vlan_id>>,<<var_swift_A_vlan_id>>,<<var_swift_B_vlan_id>>, <<var_provider_vlan_range>>,<<var_tenant_vlan_range>>,
<var_external_vlan_id>>
```

10. Make the port channel and associated interfaces spanning tree edge ports.

```
spanning-tree port type edge trunk
```

11. Set the MTU to be 9216 to support jumbo frames.

```
mtu 9216
```

12. Make this a VPC port-channel and bring it up.

```
vpc 18
no shutdown
```

13. Define a port description for the interface connecting to UCS-FLEXPOD-FAB-B.

```
interface Eth1/18
description UCS-FLEXPOD-FAB-B:1/17
```

14. Apply it to a port channel and bring up the interface.

```
channel-group 18 mode active
no shutdown
```

Cisco Nexus 9000 B

1. Define a description for the port-channel connecting to UCS-FLEXPOD-FAB-A.

```
interface po17
description UCS-FLEXPOD-FAB-A
```

2. Make the port-channel a switchport, and configure a trunk to allow in-band management, PXE, NFS, Swift, iSCSI, Provider, Tenant, and External VLAN.

```
switchport
```

```
switchport mode trunk
```

```
switchport trunk allowed vlan <<var_mgmt_vlan_id>>, <<var_pxe_vlan_id>>,
<<var_mcas_vlan_id>>, <<var_nfs_vlan_id>>,
<<var_iscsi_A_vlan_id>>, <<var_iscsi_B_vlan_id>>, <<var_swift_A_vlan_id>>, <<var_swift_B_vlan_id>>,
<<var_provider_vlan_range>>, <<var_tenant_vlan_range>>,
<var_external_vlan_id>>
```

3. Make the port channel and associated interfaces spanning tree edge ports.

```
spanning-tree port type edge trunk
```

4. Set the MTU to be 9216 to support jumbo frames.

```
mtu 9216
```

5. Make this a VPC port-channel and bring it up.

```
vpc 17
```

```
no shutdown
```

6. Define a port description for the interface connecting to UCS-FLEXPOD-FAB-A.

```
interface Eth1/17
```

```
description UCS-FLEXPOD-FAB-A:1/18
```

7. Apply it to a port channel and bring up the interface.

```
channel-group 17 mode active
```

```
no shutdown
```

8. Define a description for the port-channel connecting to UCS-FLEXPOD-FAB-B.

```
interface po18
```

```
description UCS-FLEXPOD-FAB-B
```

9. Make the port-channel a switchport, and configure a trunk to allow in-band management, PXE, NFS, Swift, iSCSI, Provider, Tenant, and External VLAN.

```

switchport
switchport mode trunk
switchport trunk allowed vlan <<var_mgmt_vlan_id>>, <<var_pxe_vlan_id>>,
<<var_mcas_vlan_id>>, <<var_nfs_vlan_id>>,
<<var_iscsi_A_vlan_id>>,<<var_iscsi_B_vlan_id>>,<<var_swift_A_vlan_id>>,<<var_swift_B_vlan_id>>, <<var_provider_vlan_range>>,<<var_tenant_vlan_range>>,
<var_external_vlan_id>>

```

10. Make the port channel and associated interfaces spanning tree edge ports.

```
spanning-tree port type edge trunk
```

11. Set the MTU to be 9216 to support jumbo frames.

```
mtu 9216
```

12. Make this a VPC port-channel and bring it up.

```
vpc 18
no shutdown
```

13. Define a port description for the interface connecting to UCS-FLEXPOD-FAB-B.

```
interface Eth1/18
description UCS-FLEXPOD-FAB-B:1/18
```

14. Apply it to a port channel and bring up the interface.

```
channel-group 18 mode active
no shutdown
```

Configure Network Interfaces Connected to NetApp FAS8040

Cisco Nexus 9000 A

1. Define a description for the port-channel connecting to FLEXPOD-OPS-CLUSTER-01

```
interface Po25
description FAS8040-A
```

2. Make the port-channel a switchport, and configure a trunk to allow NFS and iSCSI VLANs.

```
switchport
```

```

switchport mode trunk

switchport trunk allowed vlan <<var_nfs_vlan_id>>,
<<var_iscsi_A_vlan_id>>,<<var_iscsi_B_vlan_id

```

3. Make the port channel and associated interfaces spanning tree edge ports.

```
spanning-tree port type edge trunk
```

4. Set the MTU to be 9216 to support jumbo frames.

```
mtu 9216
```

5. Make this a VPC port-channel and bring it up.

```
vpc 25
no shutdown
```

6. Define a port description for the interface connecting to FLEXPOD-OPS-CLUSTER-01

```
interface Eth1/25
description FAS8040-A:e0e
```

7. Apply it to a port channel and bring up the interface.

```
channel-group 25 mode active
no shutdown
```

8. Define a description for the port-channel connecting to FLEXPOD-OPS-CLUSTER-02

```
interface Po26
description FAS8040-B
```

9. Make the port-channel a switchport, and configure a trunk to allow NFS and iSCSI VLANs.

```
switchport
switchport mode trunk

switchport trunk allowed vlan <<var_nfs_vlan_id>>,
<<var_iscsi_A_vlan_id>>,<<var_iscsi_B_vlan_id

```

10. Make the port channel and associated interfaces spanning tree edge ports.

```
spanning-tree port type edge trunk
```

11. Set the MTU to be 9216 to support jumbo frames.

```
mtu 9216
```

12. Make this a VPC port-channel and bring it up.

```
vpc 26
```

```
no shutdown
```

13. Define a port description for the interface connecting to FLEXPOD-OPS-CLUSTER-02.

```
interface Eth1/26
```

```
description FAS8040-B:e0e
```

14. Apply it to a port channel and bring up the interface.

```
channel-group 26 mode active
```

```
no shutdown
```

Cisco Nexus 9000 B

1. Define a description for the port-channel connecting to FLEXPOD-OPS-CLUSTER-01

```
interface Po25
```

```
description FAS8040-A
```

2. Make the port-channel a switchport, and configure a trunk to allow NFS and iSCSI VLANs.

```
switchport
```

```
switchport mode trunk
```

```
switchport trunk allowed vlan <<var_nfs_vlan_id>>,  
<<var_iscsi_A_vlan_id>>,<<var_iscsi_B_vlan_id
```

3. Make the port channel and associated interfaces spanning tree edge ports.

```
spanning-tree port type edge trunk
```

4. Set the MTU to be 9216 to support jumbo frames.

```
mtu 9216
```

5. Make this a VPC port-channel and bring it up.

```
vpc 25
```

```
no shutdown
```

6. Define a port description for the interface connecting to FLEXPOD-OPS-CLUSTER-01

```
interface Eth1/25
```

```
description FAS8040-A:e0g
```

7. Apply it to a port channel and bring up the interface.

```
channel-group 25 mode active
```

```
no shutdown
```

8. Define a description for the port-channel connecting to FLEXPOD-OPS-CLUSTER-02

```
interface Po26
```

```
description FAS8040-B
```

9. Make the port-channel a switchport, and configure a trunk to allow NFS and iSCSI VLANs.

```
switchport
```

```
switchport mode trunk
```

```
switchport trunk allowed vlan <<var_nfs_vlan_id>>,  
<<var_iscsi_A_vlan_id>>,<<var_iscsi_B_vlan_id
```

10. Make the port channel and associated interfaces spanning tree edge ports.

```
spanning-tree port type edge trunk
```

11. Set the MTU to be 9216 to support jumbo frames.

```
mtu 9216
```

12. Make this a VPC port-channel and bring it up.

```
vpc 26  
no shutdown
```

13. Define a port description for the interface connecting to FLEXPOD-OPS-CLUSTER-02.

```
interface Eth1/26  
description FAS8040-B:e0g
```

14. Apply it to a port channel and bring up the interface.

```
channel-group 26 mode active  
no shutdown
```

Configure Interfaces Connected to NetApp E5560

Cisco Nexus 9000 A

1. Define a port description for the interface connecting to E5560-B

```
interface Eth1/33  
description E5560-CBH1P1
```

2. Configure this interface as an access port and allow VLAN for Swift-A

```
switchport access vlan <<var_swift_A_vlan_id>>
```

3. Make the interface spanning tree edge ports.

```
spanning-tree port type edge
```

4. Set the MTU to be 9216 to support jumbo frames and bring up the interface.

```
mtu 9216  
no shutdown
```

5. Define a port description for the interface connecting to E5560-A

```
interface Eth1/34  
description E5560-CAH1P1
```

6. Configure this interface as an access port and allow VLAN for Swift-A

```
switchport access vlan <<var_swift_A_vlan_id>>
```

7. Make the interface spanning tree edge ports.

```
spanning-tree port type edge
```

8. Set the MTU to be 9216 to support jumbo frames and bring up the interface.

```
mtu 9216
```

```
no shutdown
```

Cisco Nexus 9000 B

1. Define a port description for the interface connecting to E5560-B

```
interface Eth1/33
```

```
description E5560-CBH1P3
```

2. Configure this interface as an access port and allow VLAN for Swift-B

```
switchport access vlan <<var_swift_B_vlan_id>>
```

3. Configure the interface to be spanning tree edge ports.

```
spanning-tree port type edge
```

4. Set the MTU to be 9216 to support jumbo frames and bring up the interface.

```
mtu 9216
```

```
no shutdown
```

5. Define a port description for the interface connecting to E5560-A

```
interface Eth1/34
```

```
description E5560-CAH1P3
```

6. Configure this interface as an access port and allow VLAN for Swift-A

```
switchport access vlan <<var_swift_B_vlan_id>>
```

- Configure the interface to be spanning tree edge ports.

```
spanning-tree port type edge
```

- Set the MTU to be 9216 to support jumbo frames and bring up the interface.

```
mtu 9216
```

```
no shutdown
```

Create SVI Interfaces

The following SVIs and associated IP addresses are needed on both Cisco Nexus A and B switches.



This is to address Red Hat Bugzilla #1206191: [iSCSI SAN Boot only has one path available post-install](#). The iscsistart process upon bootup incorrectly uses vNIC-A to log into the fabric hosted on Node-2 of the NetApp FAS8040, which causes very long timeouts and only one path available to Cisco UCS nodes post-installation from a dm-multipath perspective. The following configuration essentially enables routing between iSCSI-A and iSCSI-B and allows the sessions originating from vNIC-A to the fabric on Node-2 to communicate.

Cisco Nexus A

```
interface Vlan40
  no shutdown
  no ip redirects
  ip address 10.23.40.1/24
  no ipv6 redirects

interface Vlan41
  no shutdown
  no ip redirects
  ip address 10.23.41.1/24
  no ipv6 redirects
```

Cisco Nexus B

```
interface Vlan40
  no shutdown
  no ip redirects
  ip address 10.23.40.2/24
  no ipv6 redirects

interface Vlan41
  no shutdown
  no ip redirects
  ip address 10.23.41.2/24
  no ipv6 redirects
```



If SVI interfaces are deployed, be sure they are deployed on both Cisco Nexus 9000s to help ensure Type-2 VPC consistency



Create SVI interfaces for provider networks using the above described configuration.

Uplink to Existing Network Infrastructure

Depending on the available network infrastructure, several methods and features can be used to uplink the FlexPod environment. This uplink configuration is beyond the scope of this document. However, if an existing Cisco Nexus environment is present, Cisco recommends using vPCs to uplink the Cisco Nexus 9000 switches in the FlexPod environment to the existing infrastructure. The previously described procedures can be used to create an uplink vPC to the existing environment.



Make sure to **run copy run start** to save the configuration on each switch after configuration is completed.

Red Hat Enterprise Linux OpenStack Platform Installer Setup

This section describes the detail instruction of deploying the RHEL-OSP Installer in a FlexPod environment.

Login to Cisco UCS Manager

The IP KVM enables the administrator to begin the installation of the Operating System through remote media. It is necessary to login to the Cisco UCS environment to run the IP KVM.

To log in to the Cisco UCS environment, complete the following steps:

1. Open a web browser and enter the IP address of the Cisco UCS cluster. This will launch the Cisco UCS Manager application.
2. Log in to Cisco UCS Manager by using the admin user name and password.
3. In the navigation pane, click the Servers tab.
4. Select Servers > Service Profile > root > SP_OSP_Installer_1_3 service profile.
5. Right-click SP_OSP_Installer_1_3 service profile and click KVM Console.

Setup RHEL-OSP Installer Server

To prepare the server for RHEL 7.1 installation, complete the following steps.

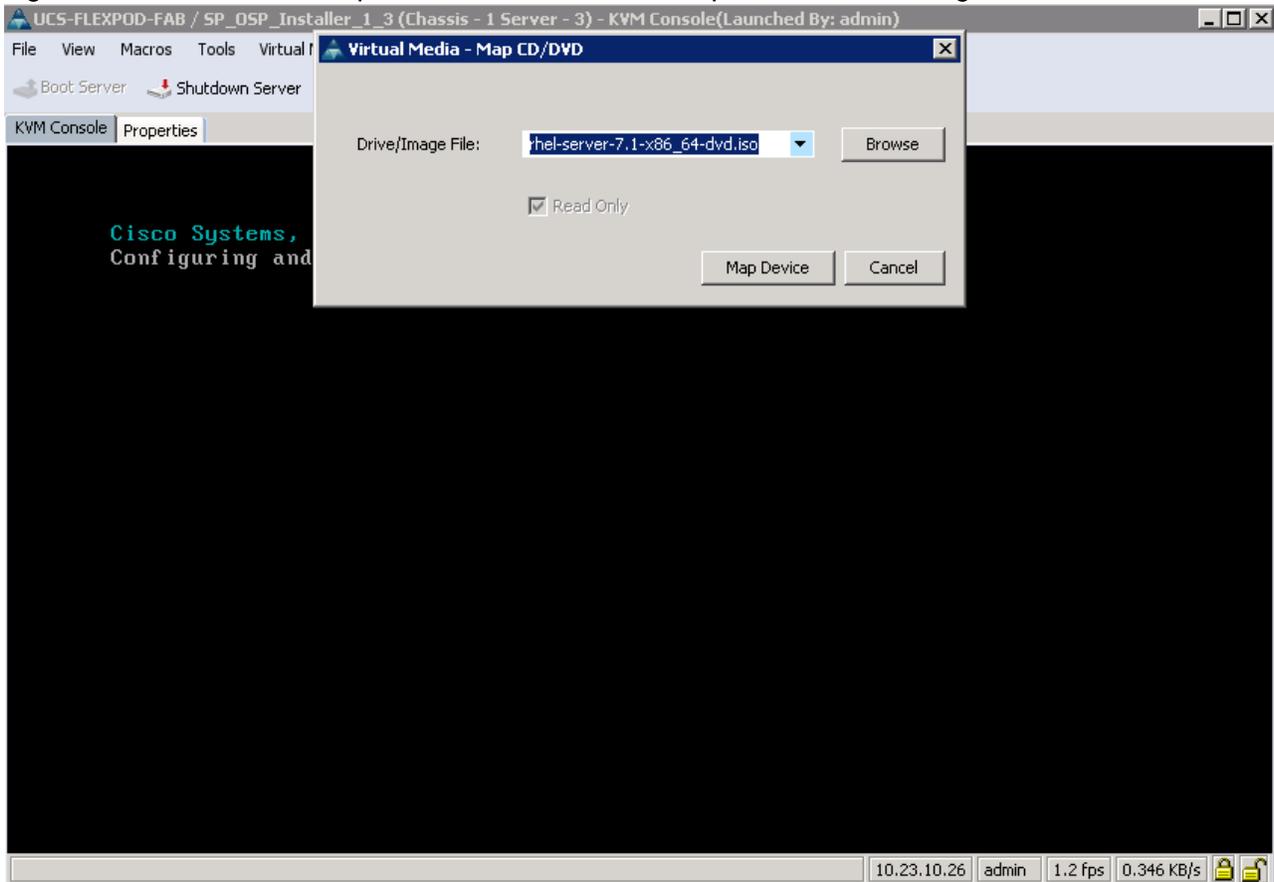


Red Hat Enterprise Linux 7.1 iso file should be available before proceeding. It can be downloaded from <https://access.redhat.com/downloads>

1. In the KVM window, click the Virtual Media menu option and select Activate Virtual Devices.
2. On Unencrypted Virtual Media Session dialog box, click Accept this session and click Apply.
3. Click the Virtual Media menu option and select Map CD/DVD.
4. Click Browse.
5. Browse to the Red Hat Enterprise Linux 7.1 ISO image file and click Open.

6. Click Map Device to map the newly added image (Figure 132).
7. Click Reset button on the menu bar. Click OK.
8. Click Power Cycle. Click OK.

Figure 132 Virtual Media Map CD/DVD for Red Hat Enterprise Linux ISO Image

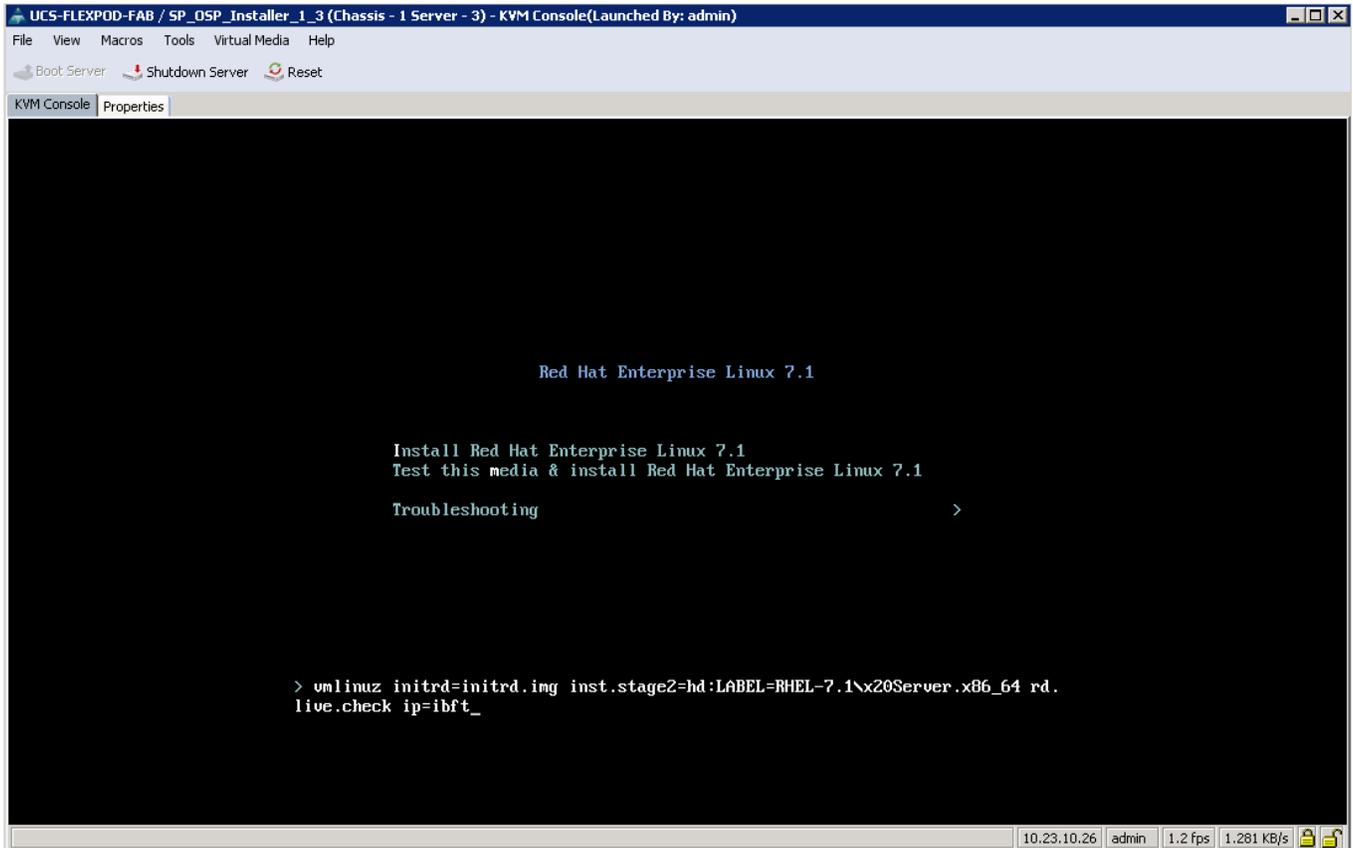


Install RHEL 7.1

To install RHEL 7.1 to the SAN bootable LUN of the hosts, complete the following steps:

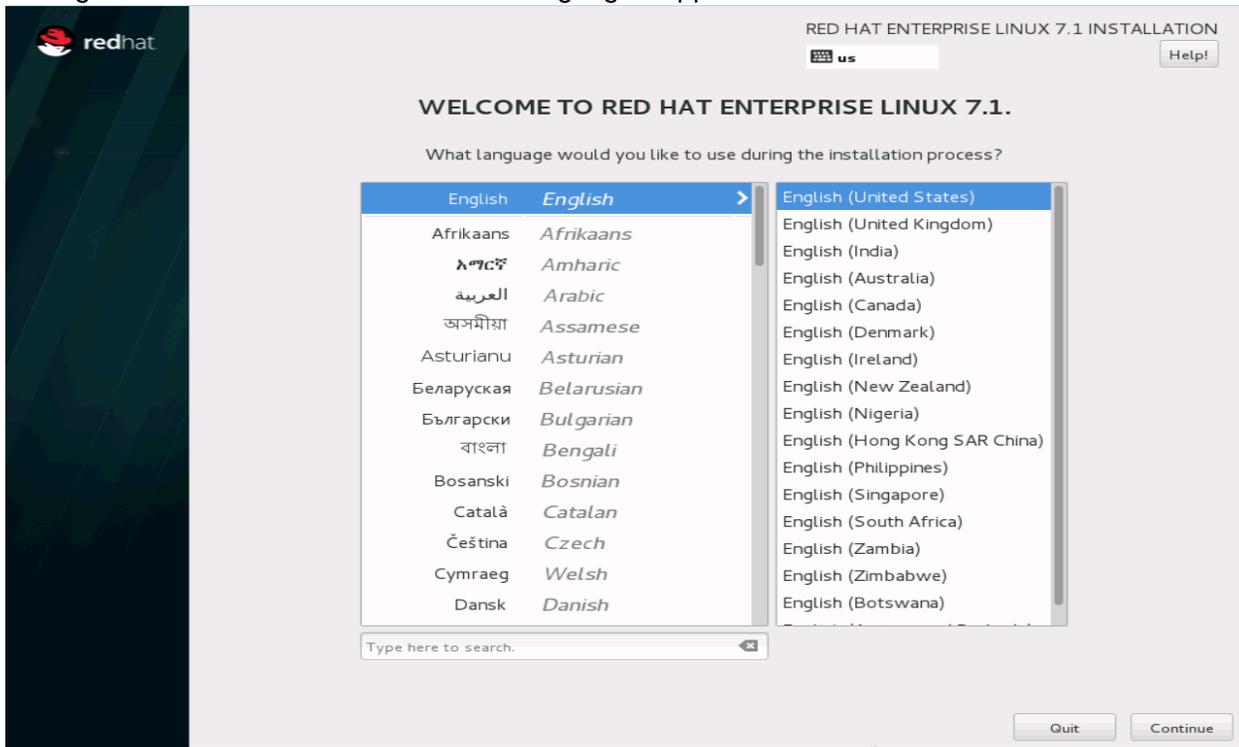
1. On reboot, the server will detect the installation media of RHEL 7.1 (Figure 133)
2. Press Tab key for full configuration.
3. Remove the `quiet` option using either the backspace or delete key(s)
4. Enter `ip=ibft`
5. Hit the Enter key. Installation will proceed.

Figure 133 RHEL 7.1 Installation Screen



6. Language English is selected by default. Click Continue button (Figure 134).

Figure 134 RHEL 7.1 Install – Select Language Support



7. Click INSTALLATION DESTINATION (Figure 135)

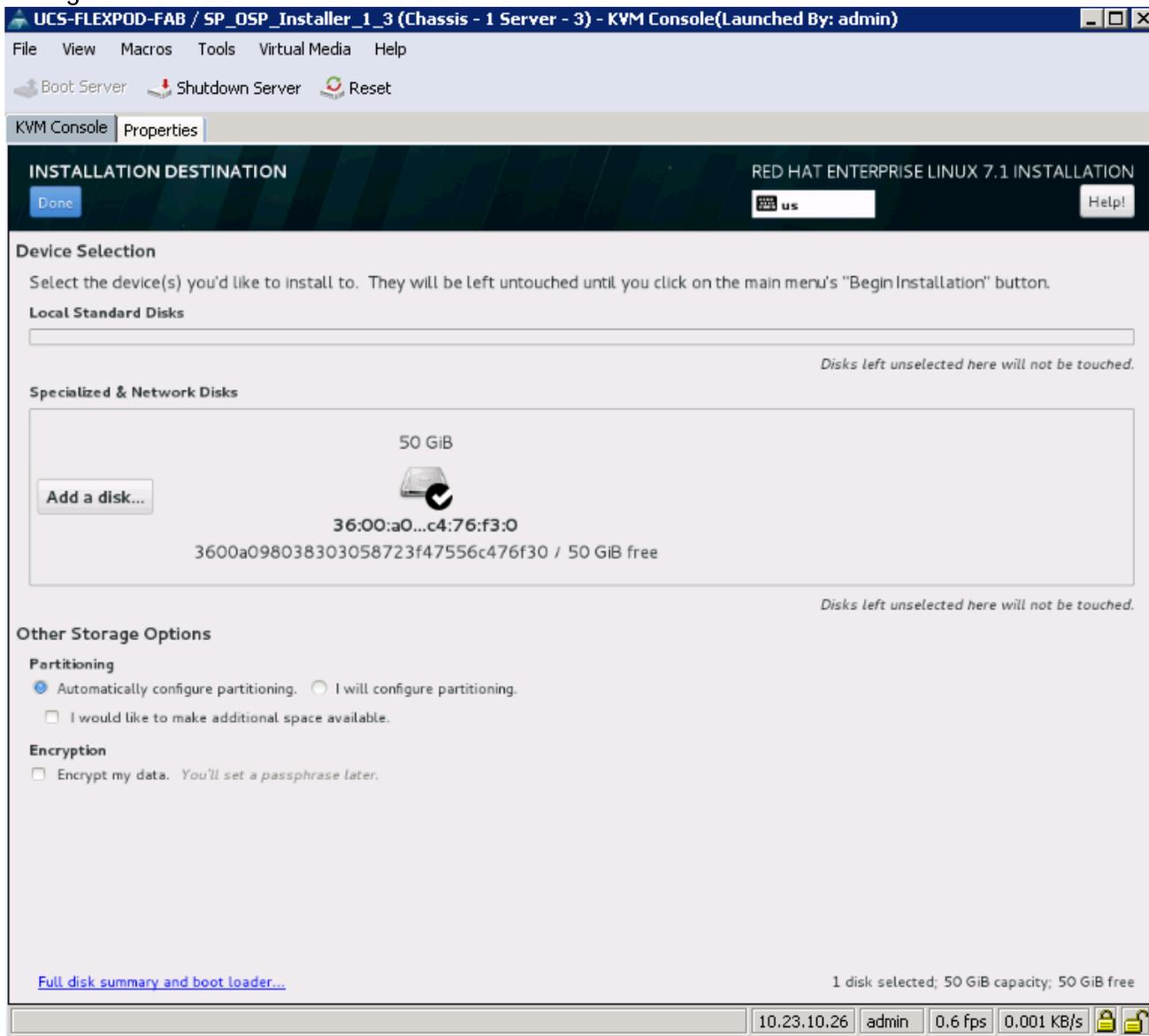
Figure 135 RHEL 7.1 Install – Installation Destination



8. Click Done (Figure 136)

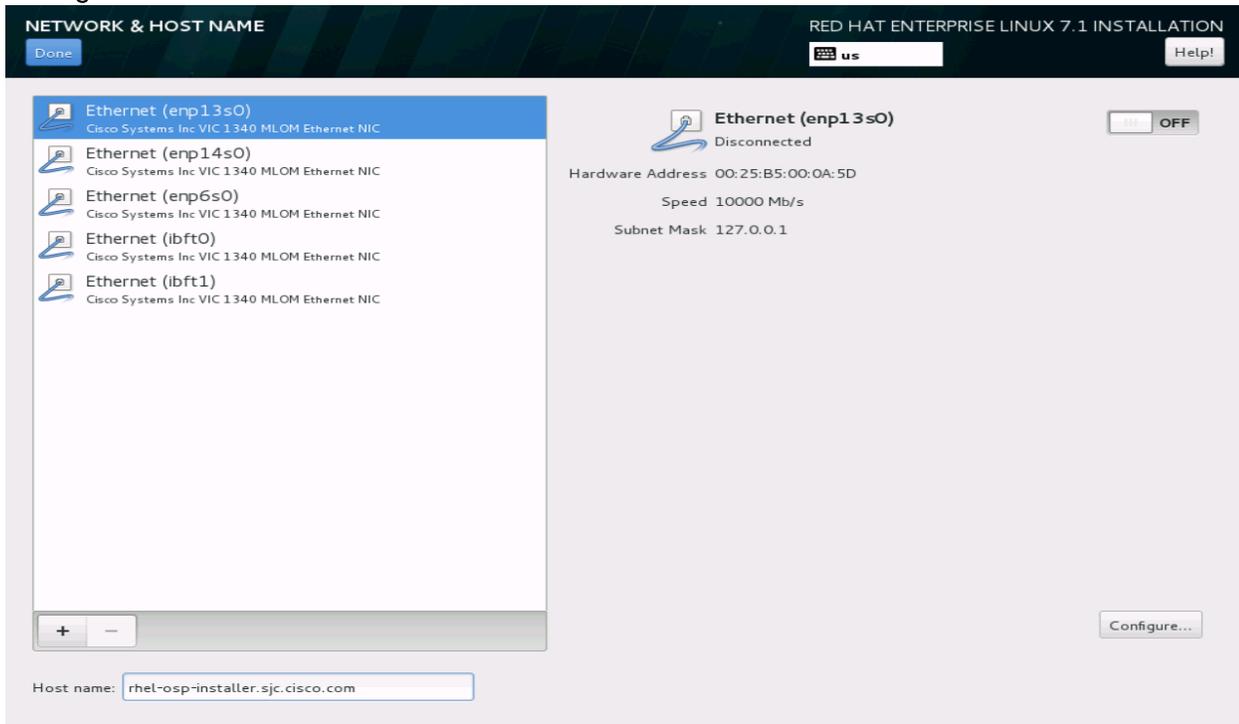
9. Modify the Date & Time to reflect where the system is physically located.

Figure 136 RHEL 7.1 Install – Installation Destination



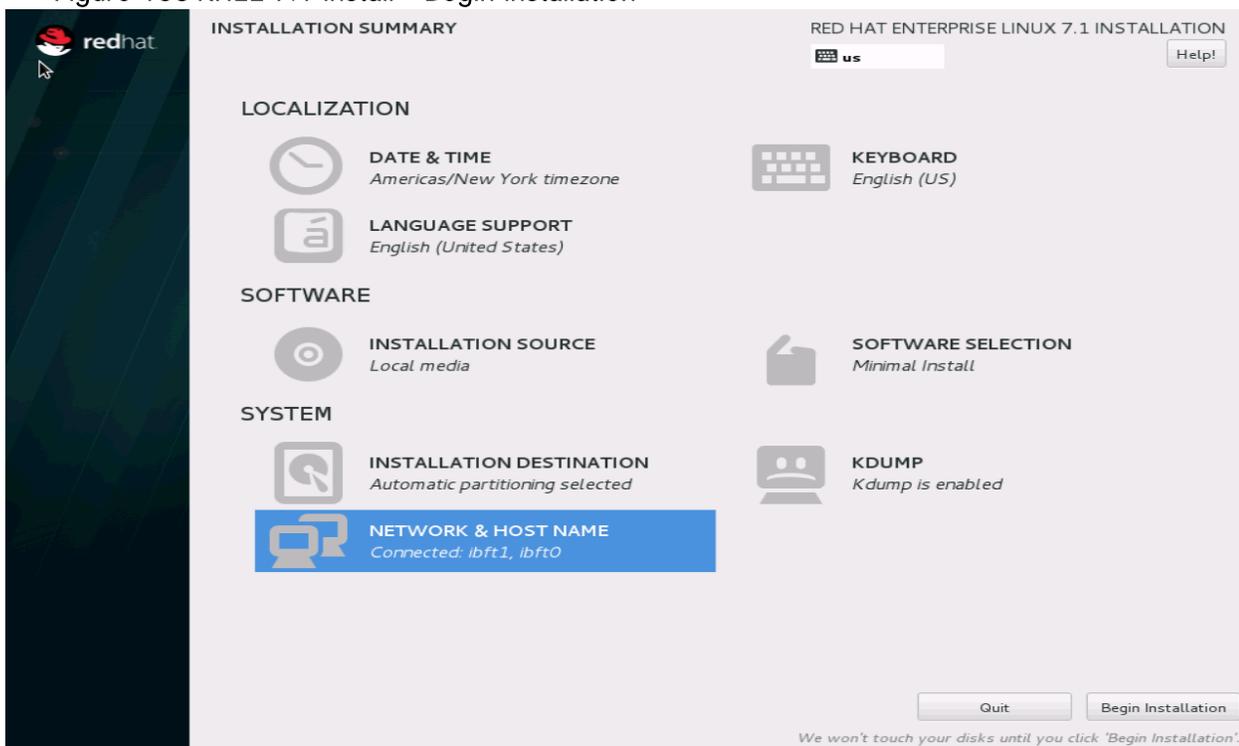
10. Click NETWORK & HOSTNAME
11. Enter Host Name (Figure 137). For example, rhel-osp-installer.sjc.cisco.com in this reference deployment.
12. Click Done.

Figure 137 RHEL 7.1 Install – Host Name



13. Click Begin Installation (Figure 138)

Figure 138 RHEL 7.1 Install – Begin Installation

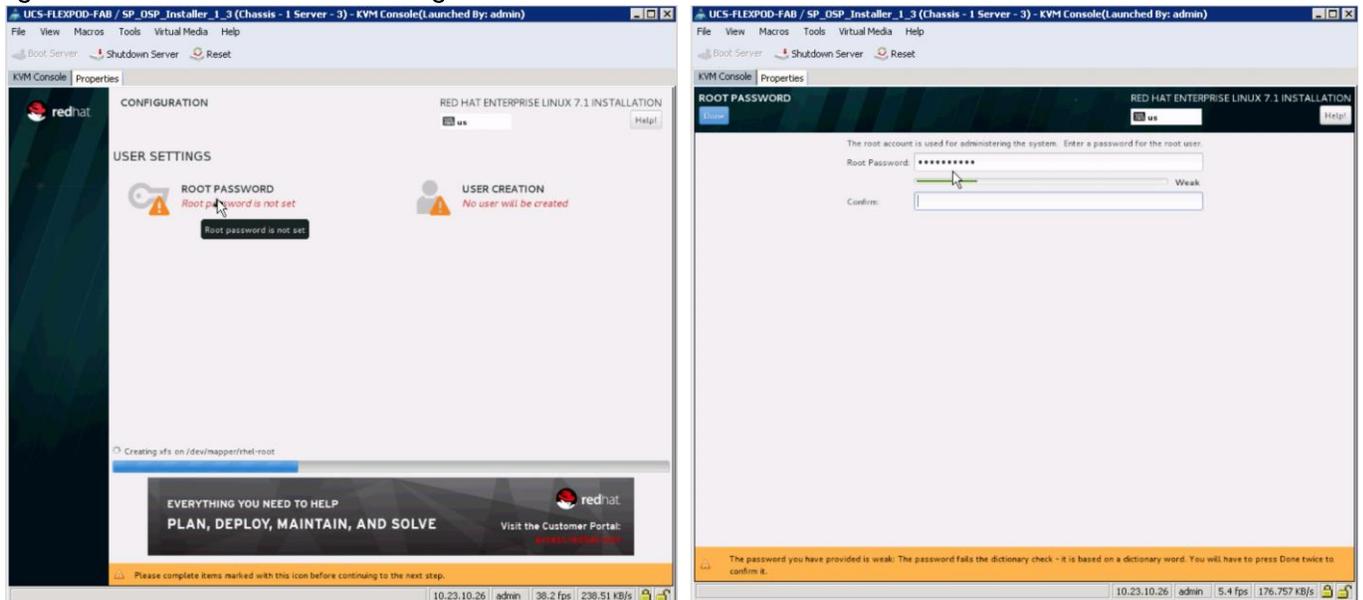


14. Change root password by clicking ROOT PASSWORD (Figure 139)

15. Enter Root Password in the ROOT PASSWORD screen.

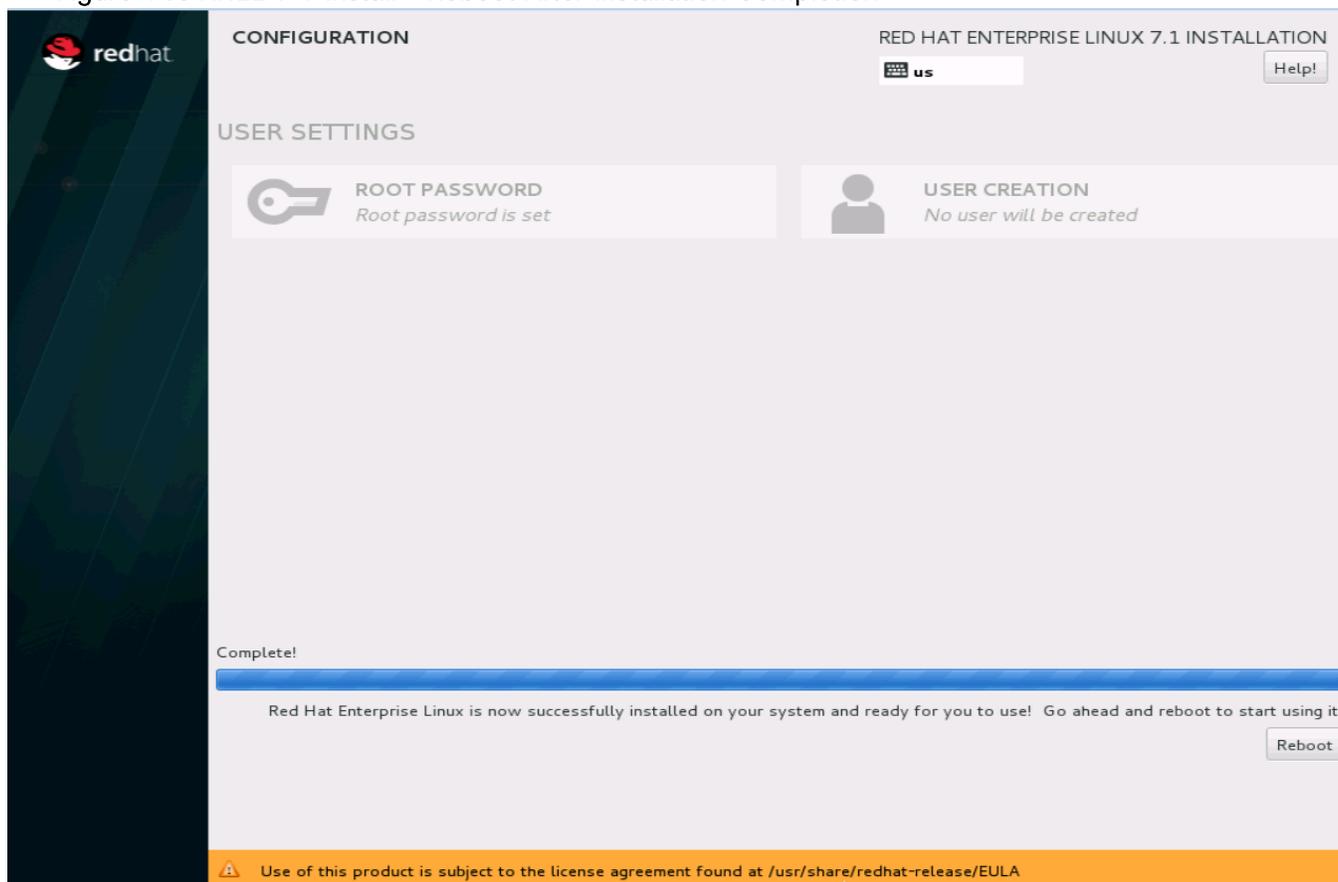
16. Re-enter root password in the Confirm field.
17. Click Done. (Click Done twice, if the password is weak. However, a weak password is NOT recommended for production deployment).

Figure 139 RHEL 7.1 Install – Change root Password



18. Wait for the installation to complete.
19. Click Reboot when installation is completed (Figure 140).

Figure 140 RHEL 7.1 Install – Reboot After Installation Completion



RHEL-OSP Installer Prerequisites

This section provides important prerequisites for Red Hat Enterprise Linux OpenStack Platform installer and pre-installation configuration. After the reboot, log in to the installer host.

Disable Network Manager

The `rhel-osp-installer` script uses `network` service instead of the `NetworkManager` service. Disable `NetworkManager` using the following commands:

```
[root@rhel-osp-installer ~]# systemctl stop NetworkManager.service
[root@rhel-osp-installer ~]# systemctl disable NetworkManager.service
```

Configure RHEL-OSP Installer Server Interfaces

To configure RHEL-OSP installer server interface, complete the following steps:

1. After the reboot, log in to Installer node by providing login as: `root` and password specified during the installation.
2. Configure management, pxe, and external interfaces. Red Hat no longer uses `ethX` for interface naming. In the reference architecture, the management interface is named `enp13s0`, PXE interface is named `enp6s0`, and external or public interface is named `enp14s0` as shown in Table 15 .
3. `cd /etc/sysconfig/network-scripts`

4. Create `ifcfg-enp6s0`, `ifcfg-enp13s0.10`, and `ifcfg-enp14s0.215` with the following sample configuration as indicated in Table 15 .
5. After the interface configuration, restart the network service

```
systemctl restart network.service
```

Table 15 Installer Server Network Addresses

Network	Interface	Network Address	Net Mask/Prefix	Gateway	DNS1
Management	<code>enp13s0.10</code>	<code>10.23.10.36</code>	<code>24</code>	<code>-</code>	
PXE	<code>enp6s0</code>	<code>10.23.20.2</code>	<code>255.255.255.0</code>		<code>10.23.10.2</code>
External or Public	<code>enp14s0.215</code>	<code><<osp_installer_external_ip>></code>	<code>24</code>	<code><<osp_installer_external_gateway>></code>	<code><<var_nameserver_ip>></code>



In this reference architecture, everything other than the PXE network (also known as “default” in Red Hat Enterprise Linux OpenStack Platform Installer web interface) is 802.1q VLAN tagged for consistency.



Make sure to have `<<osp_installer_external_gateway>>` as your default gateway. Check default gateway by running “`ip route`” command. To change the default gateway to `<<osp_installer_external_gateway>>` run “`ip route add default via <<osp_installer_external_gateway>> dev enp14s0.215`”



For reference purpose, refer to [Appendix J: Red Hat Enterprise Linux OpenStack Platform Installer Server Interfaces Config](#) for interface configuration.

Remove dnsmasq

The `dnsmasq` service is a controller for DNS and DHCP services. However, it can interfere with the installer's management of the DHCP. It is recommended to remove it from the installer's system.

```
yum remove dnsmasq
```

Subscription Manager

To install the RHEL OpenStack Platform installer, first register the host system using Red Hat Subscription Manager, and subscribe to the required channels.

1. Register your system with the Content Delivery Network, entering your Customer Portal user name and password when prompted:

```
subscription-manager register
```

2. Find entitlement pools containing the channels required to install the Red Hat Enterprise Linux OpenStack Platform Installer.

```
subscription-manager list --available | grep -A8 "Red Hat Enterprise Linux Server"
subscription-manager list --available | grep -A8 "Red Hat Enterprise Linux OpenStack Platform"
```

3. Use the pool identifiers located in the previous step to attach the Red Hat Enterprise Linux 7 Server and Red Hat Enterprise Linux OpenStack Platform entitlements:

```
subscription-manager attach --pool=pool_id
```

4. Enable the required channels:

```
subscription-manager repos --enable rhel-7-server-openstack-6.0-rpms
subscription-manager repos --enable rhel-7-server-openstack-6.0-installer-rpms
subscription-manager repos --enable rhel-7-server-optional-rpms
subscription-manager repos --enable rhel-7-server-extras-rpms
subscription-manager repos --enable rhel-server-rhsc1-7-rpms
```

Configure NTP Server

NTP server is a mandatory requirement for setting up the RHEL-OSP Installer. **Make sure your organization's** NTP server is accessible from the RHEL-OSP Installer server. Run `ntpdate -u <<var_global_ntp_server_ip>>` command to validate.

Configure Installer Server as the Gateway

A gateway is required for external access for the OpenStack managed nodes. The Installer private network interface IP can act as a gateway address. It is also possible to point to an external gateway. If the installer is configured as the gateway, IP forwarding must be enabled on the system.

1. Edit `/etc/sysctl.conf` and change the value of `net.ipv4.ip_forward` to 1.

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

2. Load the new value

```
cd /etc
sysctl -p
```

Configure iptables in Red Hat Enterprise Linux OpenStack Platform Installer Server

Enable firewall and configure iptables. The installer server must broadcast DHCP and allow PXE boot to the clients. Hence the relevant ports must be open. In this reference architecture, `firewalld` has been disabled and replaced by `iptables` service and configured as follows (See [Appendix D: Red Hat Enterprise Linux OpenStack Platform Installer Server iptables Configuration](#), for full Red Hat Enterprise Linux OpenStack Platform Installer iptables entries as a reference).

1. Configure iptables:

```
iptables -t nat -I POSTROUTING 1 -o enp14s0.215 -j MASQUERADE
iptables -I FORWARD 1 ! -s 10.23.20.0/24 -j DROP
iptables -I FORWARD 1 -d 10.23.20.0/24 -j ACCEPT
iptables -I FORWARD 1 -s 10.23.20.0/24 -j ACCEPT
iptables -I FORWARD -i enp6s0 -j ACCEPT
iptables -I INPUT 1 -p tcp -m multiport --ports 123 -m comment --comment "ntp" -j ACCEPT
iptables -I INPUT 1 -p udp -m multiport --ports 123 -m comment --comment "ntp" -j ACCEPT
```



enp14s0 is the name of the network interface to which to forward network traffic to the outside network. In this validation, 10.23.20.0/24 is the PXE subnet we wish to NAT through the external interface.

2. Save the changes to the firewall.

```
iptables-save > /etc/sysconfig/iptables
```

3. Restart networking

```
systemctl restart network.service
```

Update Cisco eNIC Driver

To update the Cisco eNIC driver, complete the following steps:

1. Visit <http://software.cisco.com/download/navigator.html>
2. In the download page, select Servers-Unified Computing. On the right list item, select UCS B-Series Blade Server Software (Download Homes > Products > Server-Unified Computing > UCS B-Series Blade Server Software).
3. Select Software Type as Unified Computing System (UCS) Drivers.
4. Select your firmware version under All Release, Select 2.2(3g) and download the ISO image.
5. Download the related driver matching the UCS firmware. For example, ucs-bxxx-drivers.2.2.3g.iso
6. Extract the enic rpm from the ISO for RHEL 7.1. **Rpm for 2.1.1.75 is "kmod-enic-2.1.1.75-rhel7u1.el7.x86_64.rpm"**
7. Upload the kmod-enic-2.1.1.75-rhel7u1.el7.x86_64.rpm file in the OSP Installer server. Several methods can be used for this purpose such as ftp, tftp, scp, WinSCP and so on depending on your environment.
8. Complete the following steps to update the eNIC driver.

- a. [root@rhel-osp-installer ~]# rpm -ivh kmod-enic-2.1.1.75-rhel7u1.el7.x86_64.rpm (Figure 141)

Figure 141 Update Cisco eNIC Driver

```
[root@rhel-osp-installer ~]# rpm -ivh kmod-enic-2.1.1.75-rhel7u1.el7.x86_64.rpm
Preparing...                               ##### [100%]
Updating / installing...
 1:kmod-enic-2.1.1.75-rhel7u1.el7          ##### [100%]
[root@rhel-osp-installer ~]#
```

- b. Verify eNIC driver by issuing command `modinfo enic` (Figure 142).

Figure 142 Verify Cisco eNIC Driver

```
[root@rhel-osp-installer ~]# modinfo enic
filename:      /lib/modules/3.10.0-229.4.2.el7.x86_64/weak-updates/enic/enic.ko
version:      2.1.1.75
license:      GPL v2
author:       Scott Feldman <scofeldm@cisco.com>
description:  Cisco VIC Ethernet NIC Driver
rhelversion:  7.1
srcversion:   8035F1081D87D41276717C6
alias:        pci:v00001137d00000071sv*sd*bc*sc*i*
alias:        pci:v00001137d00000044sv*sd*bc*sc*i*
alias:        pci:v00001137d00000043sv*sd*bc*sc*i*
depends:
vermagic:     3.10.0-229.el7.x86_64 SMP mod_unload modversions
[root@rhel-osp-installer ~]#
```

- c. [root@rhel-osp-installer ~]# modprobe enic
- d. Reboot the server.

Install RHEL-OSP Installer Packages

Install the rhel-osp-installer and any necessary packages required.

```
yum install rhel-osp-installer
```

Install RHEL-OSP Installer

1. Launch the rhel-osp-installer script to begin the installation.

```
rhel-osp-installer
```



Make sure fully qualified hostname is configured in /etc/hosts file with PXE interface IP address. For example: 10.23.20.2 rhel-osp-installer.sjc.cisco.com rhel-osp-installer

2. Enter the number of network interface that the installer will use to provision RHEL OpenStack Platform. In reference topology, enp6s0 is the pxe/provisioning network, enter 5 and press ENTER (Figure 143)

Figure 143 Install rhel-osp-installer – select PXE or Provisioning Interface

```
Please select NIC on which you want provisioning enabled:
1. enp13s0
2. enp13s0.10
3. enp14s0
4. enp14s0.215
5. enp6s0
6. ibft0
7. ibft1
? 5
```

- Based on the network interface selected, the network setup for the deployed nodes are configured in the installer.

Networking setup:

```

Network interface: 'enp6s0'

    IP address: '10.23.20.2'

    Network mask: '255.255.255.0'

Network address: '10.23.20.0'

    Host Gateway: '<<osp_installer_external_gateway>>'

DHCP range start: '10.23.20.3'

    DHCP range end: '10.23.20.254'

    DHCP Gateway: '10.23.20.2'

    DNS forwarder: '<<var_nameserver_ip>>'

        Domain: 'sjc.cisco.com'

    NTP sync host: '0.rhel.pool.ntp.org'

        Timezone: 'America/Los_Angeles'

```

Configure networking on this machine: ✓

Configure firewall on this machine: ✓

The installer can configure the networking and firewall rules on this machine with the above configuration. Default values are populated from the this machine's existing networking configuration.

If you DO NOT want to configure networking please set 'Configure networking on this machine' to No before proceeding. Do this by selecting option 'Do not configure networking' from the list below.



The name of the domain must match that of the fully qualified domain name of the machine hosting the installer.

- The predetermined parameters can be altered. In this case, NTP server has been modified as described below:

How would you like to proceed?:

- Proceed with the above values
- Change Network interface
- Change IP address

4. Change Network mask
 5. Change Network address
 6. Change Host Gateway
 7. Change DHCP range start
 8. Change DHCP range end
 9. Change DHCP Gateway
 10. Change DNS forwarder
 11. Change Domain
 12. Change NTP sync host
 13. Change Timezone
 14. Do not configure networking
 15. Do not configure firewall
 16. **Cancel Installation**
- 12

Enter a list of NTP hosts, separated by commas. First in the list will be the default.

<<var_global_ntp_server_ip>>

Networking setup:

```

Network interface: 'enp6s0'
    IP address: '10.23.20.2'
    Network mask: '255.255.255.0'
    Network address: '10.23.20.0'
    Host Gateway: '<<osp_installer_external_gateway>>'
    DHCP range start: '10.23.20.3'
    DHCP range end: '10.23.20.254'
    DHCP Gateway: '10.23.20.2'
    DNS forwarder: '<<osp_installer_external_ip>>'
    Domain: 'sjc.cisco.com'
    NTP sync host: '<<var_global_ntp_server_ip>>'
    Timezone: 'America/Los_Angeles'

```

Configure networking on this machine: ✓

Configure firewall on this machine: ✓

The installer can configure the networking and firewall rules on this machine with the above configuration. Default values are populated from the this machine's existing networking configuration.

If you DO NOT want to configure networking please set 'Configure networking on this machine' to No before proceeding. Do this by selecting option 'Do not configure networking' from the list below.

5. Update time zone and the DNS domain for the provisioning network as needed. The IP addresses assigned to the OSP hosts on the provisioning network will resolve to this domain.
6. If all values are correct, the network configuration is setup by selecting the option 1 “Proceed with the above values” as shown below:

How would you like to proceed?:

1. **Proceed with the above values**
 2. Change Network interface
 3. Change IP address
 4. Change Network mask
 5. Change Network address
 6. Change Host Gateway
 7. Change DHCP range start
 8. Change DHCP range end
 9. Change DHCP Gateway
 10. Change DNS forwarder
 11. Change Domain
 12. Change NTP sync host
 13. Change Timezone
 14. Do not configure networking
 15. Do not configure firewall
 16. **Cancel Installation**
- 1

Configure client authentication

SSH public key: ''

Root password: '*****'

Please set a default root password for newly provisioned machines. If you choose not to set a password, it will be generated randomly. The password must be a minimum of 8 characters. You can also set a public ssh key which will be deployed to newly provisioned machines.

How would you like to proceed?:

1. **Proceed with the above values**
 2. Change SSH public key
 3. Change Root password
 4. Toggle Root password visibility
7. The root password for the deployed servers can be set and verified. SSH key authentication can be configured to allow passwordless root authentication between the OSP installation server and the OSP infrastructure. The usage of SSH key authentication and the complexity of the root password should meet organizational security compliance requirements. Select Option 3 to Change Root password.

How would you like to proceed?:

1. **Proceed with the above values**
2. Change SSH public key
3. Change Root password
4. Toggle Root password visibility

3

new value for the root password

enter new root password again to confirm

Configure client authentication

SSH public key: ''

Root password: '*****'

Please set a default root password for newly provisioned machines. If you choose not to set a password, it will be generated randomly. The password must be a minimum of 8 characters. You can also set a public ssh key which will be deployed to newly provisioned machines.

How would you like to proceed?:

1. **Proceed with the above values**
2. Change SSH public key
3. Change Root password
4. Toggle Root password visibility



For password visibility use option 4 - Toggle Root password visibility

8. If all values are correct, enter 1 to Proceed with the above values and press ENTER.

```
How would you like to proceed?:
```

- ```
1. Proceed with the above values
2. Change SSH public key
3. Change Root password
4. Toggle Root password visibility
```

```
1
```

```
Starting networking setup
```

```
Networking setup has finished
```

```
Installing Done [100%]
[.....]
```

```
Starting configuration...
```

```
Redirecting to /bin/systemctl stop puppet.service
```

```
Redirecting to /bin/systemctl start puppet.service
```

```
Now you should configure installation media which will be used for provisioning.
```

```
Note that if you don't configure it properly, host provisioning won't work until you configure installation media manually.
```

```
Enter RHEL repo path:
```

- ```
1. Set RHEL repo path (http or https URL): http://
2. Proceed with configuration
3. Skip this step (provisioning won't work)
```

9. Specify the RHEL 7.1 URL for setting up RHEL repo that will be used to deploy RHEL 7.1 in controller and compute nodes. Press option 2 to Proceed with configuration and press ENTER as shown below.

Enter RHEL repo path:

1. Set RHEL repo path (http or https URL): <http://>

2. **Proceed with configuration**

3. **Skip this step (provisioning won't work)**

1

Path: <http://rhel-osp-installer.sjc.cisco.com:8080/RHEL7>

Enter RHEL repo path:

1. Set RHEL repo path (http or https URL): <http://rhel-osp-installer.sjc.cisco.com:8080/RHEL7>

2. **Proceed with configuration**

3. **Skip this step (provisioning won't work)**

2

10. In the next step, provide subscription manager inputs: username, password, and required channels.

11. Select 1 to enter Subscription manager username and press ENTER.

Enter your subscription manager credentials:

1. Subscription manager username:

2. Subscription manager password:

3. Comma or Space separated repositories: [rhel-7-server-rpms](#) [rhel-7-server-openstack-6.0-rpms](#) [rhel-7-server-openstack-6.0-installer-rpms](#) [rhel-ha-for-rhel-7-server-rpms](#) [rhel-7-server-rh-common-rpms](#)

4. Subscription manager pool (recommended):

5. Subscription manager proxy hostname:

6. Subscription manager proxy port:

7. Subscription manager proxy username:

8. Subscription manager proxy password:

9. **Proceed with configuration**

10. **Skip this step (provisioning won't subscribe your machines)**

1

Username: <<username>>

12. Select option 2 to enter Subscription manager password and press ENTER. Type Subscription manager password.

```
Enter your subscription manager credentials:
```

1. Subscription manager username: <<username>>
 2. Subscription manager password:
 3. Comma or Space separated repositories: **rhel-7-server-rpms rhel-7-server-openstack-6.0-rpms rhel-7-server-openstack-6.0-installer-rpms rhel-ha-for-rhel-7-server-rpms rhel-7-server-rh-common-rpms**
 4. Subscription manager pool (recommended):
 5. Subscription manager proxy hostname:
 6. Subscription manager proxy port:
 7. Subscription manager proxy username:
 8. Subscription manager proxy password:
 9. **Proceed with configuration**
 10. **Skip this step (provisioning won't subscribe your machines)**
- 2
- Password: *****

13. Select option 4 to specify Subscription manager pool. Specify pool ID.



The value of the Subscription Manager pool must be in the format of the Subscription Manager entitlement Pool ID. Furthermore, you can only specify a single entitlement Pool ID. If you leave the value for the configuration item blank, the installer attempts to auto-attach the first entitlement in your subscription account. If you have multiple subscriptions in your subscription account, you may encounter issues if you leave the Pool ID blank. Therefore, it is recommended to specify the Pool ID for the required subscription.

```
Enter your subscription manager credentials:
```

1. Subscription manager username: <username>
2. Subscription manager password: *****
3. Comma or Space separated repositories: **rhel-7-server-rpms rhel-7-server-openstack-6.0-rpms rhel-7-server-openstack-6.0-installer-rpms rhel-ha-for-rhel-7-server-rpms rhel-7-server-rh-common-rpms**
4. Subscription manager pool (recommended):
5. Subscription manager proxy hostname:
6. Subscription manager proxy port:
7. Subscription manager proxy username:
8. Subscription manager proxy password:

9. **Proceed with configuration**
10. **Skip this step (provisioning won't subscribe your machines)**

4
Pool: <pool_id>

14. Proceed with the installation. Select 9 and press ENTER. The installation will begin.

Enter your subscription manager credentials:

1. Subscription manager username: `<username>`
 2. Subscription manager password: `*****`
 3. Comma or Space separated repositories: `rhel-7-server-rpms rhel-7-server-openstack-6.0-rpms rhel-7-server-openstack-6.0-installer-rpms rhel-ha-for-rhel-7-server-rpms rhel-7-server-rh-common-rpms`
 4. Subscription manager pool (recommended): `<pool_id>`
 5. Subscription manager proxy hostname:
 6. Subscription manager proxy port:
 7. Subscription manager proxy username:
 8. Subscription manager proxy password:
 9. **Proceed with configuration**
 10. **Skip this step (provisioning won't subscribe your machines)**
- 9

Starting to seed provisioning data

Use 'base_RedHat_7' hostgroup for provisioning

15. When the installation is completed successfully, the application URL and login credentials will be displayed in the final output as shown below:

Success!

* **Foreman** is running at `https://rhel-osp-installer.sjc.cisco.com`

Initial credentials are `admin / 65VhMXSdoNWthhQt`

* **Foreman Proxy** is running at `https://rhel-osp-installer.sjc.cisco.com:8443`

* **Puppetmaster** is running at `port 8140`

The full log is at `/var/log/rhel-osp-installer/rhel-osp-installer.log`



The login credentials are also stored in `/etc/foreman/rhel-osp-installer.answers.yaml` (`# cat /etc/foreman/rhel-osp-installer.answers.yaml|grep admin_password`). It is recommended to login immediately with the initial credentials and change the password for the admin user. Please refer to [Appendix C: Changing Installer GUI Password](#), for instructions how to change the admin password through installer GUI.

Prepare the RHEL 7.1 Installation Medium

The installer requires an installation medium. An installation medium is a source of files the installer uses to install the base operating system on controller and compute hosts when you provision RHEL OpenStack Platform, and must be in the format of a Red Hat Enterprise Linux 7.1 installation tree.

To configure the web server on the Installer host to store and share Red Hat Enterprise Linux 7.1 installation medium, complete the following steps:



This procedure must be performed on the installer host after completing the `rhel-osp-installer` configuration script. This is because the script writes over any existing `httpd` settings with its own configuration during the installation process. This procedure modifies these settings after the `rhel-osp-installer` application writes its own `httpd` configuration.

1. Go to <https://access.redhat.com>, and log in to the Red Hat Customer Portal using your customer account details.
2. Click Downloads in the menu bar.
3. Click Red Hat Enterprise Linux to access the product download page.
4. Click RHEL 7.1 Binary DVD.
5. Create a temporary directory into which to mount the ISO file:

```
mkdir /mnt/RHEL7
```

6. Upload the `rhel-server-7.1-x86_64-dvd.iso` file in the OSP Installer server. Several methods can be used for this purpose such as `ftp`, `tftp`, `scp`, `WinSCP` and so on depending on your environment.
7. Mount the ISO file in the temporary directory.

```
mount -t iso9660 -o loop rhel-server-7.1-x86_64-dvd.iso /mnt/RHEL7
```

8. Copy the contents of the temporary directory to the directory in which to store the files for the installation medium:

```
cp -dpR /mnt/RHEL7 /var/www/html/.
```

9. Unmount the ISO file.

```
umount /mnt/RHEL7
```

10. Remove the temporary directory in which you mounted the ISO file

```
rm -r /mnt/RHEL7
```

11. Set the permission for the installation medium

```
chmod -R 755 /var/www/html/RHEL7
```

12. Create a new file at /etc/httpd/conf.d/medium.conf and add the following configuration to it. This will expose an accessible location on port 8080 (unassigned port) of the web server, which contains RHEL7 folder with the installation medium.

```
Listen 8080
NameVirtualHost *:8080
<VirtualHost *:8080>
DocumentRoot /var/www/html/
ServerName rhel-osp-installer.sjc.cisco.com
<Directory "/var/www/html/">
Options All Indexes FollowSymLinks
Order allow,deny
Allow from all
</Directory>
</VirtualHost>
```

13. Add a rule to your firewall configuration to allow access to port 8080, then restart the firewall:

```
iptables -I INPUT 1 -p tcp -m multiport --ports 8080 -m comment --comment "8080 accept - rh7repo" -j ACCEPT
iptables-save > /etc/sysconfig/iptables
systemctl restart iptables.service
```

14. Restart your web server:

```
systemctl restart httpd.service
```

The installation medium is now available for the Red Hat Enterprise Linux OpenStack Platform installer to access. You can validate access to the installation medium by navigating to <http://www.example.com:8080/RHEL7/>, which should display a listing of Red Hat Enterprise Linux 7 files and folders.

Prepare Downloadable URL for Cisco eNIC Driver

To update the Cisco eNIC driver in the provisioning hosts in an automated fashion, Cisco eNIC driver http downloadable link needs to be configured in installer host.

1. Create a CiscoEnic folder in /var/www/html

```
[root@rhel-osp-installer ~]# mkdir /var/www/html/CiscoEnic
```

2. Copy "kmod-enic-2.1.1.75-rhel7u1.el7.x86_64.rpm" file previously downloaded in section "Update Cisco eNIC Driver" to /var/www/html/CiscoEnic folder and provide appropriate permissions.

```
[root@rhel-osp-installer ~]# cp /root/kmod-enic-2.1.1.75-rhel7u1.el7.x86_64.rpm /var/www/html/CiscoEnic/
[root@rhel-osp-installer ~]# chmod -R 755 /var/www/html/CiscoEnic/
```

This will setup downloadable http link for Cisco eNIC driver and will be available at <http://rhel-osp-installer.sjc.cisco.com:8080/CiscoEnic>. Make a note of this URL as this will be configured in the kickstart script in later section.

Cisco eNIC Driver URL	http://rhel-osp-installer.sjc.cisco.com:8080/CiscoEnic
-----------------------	---



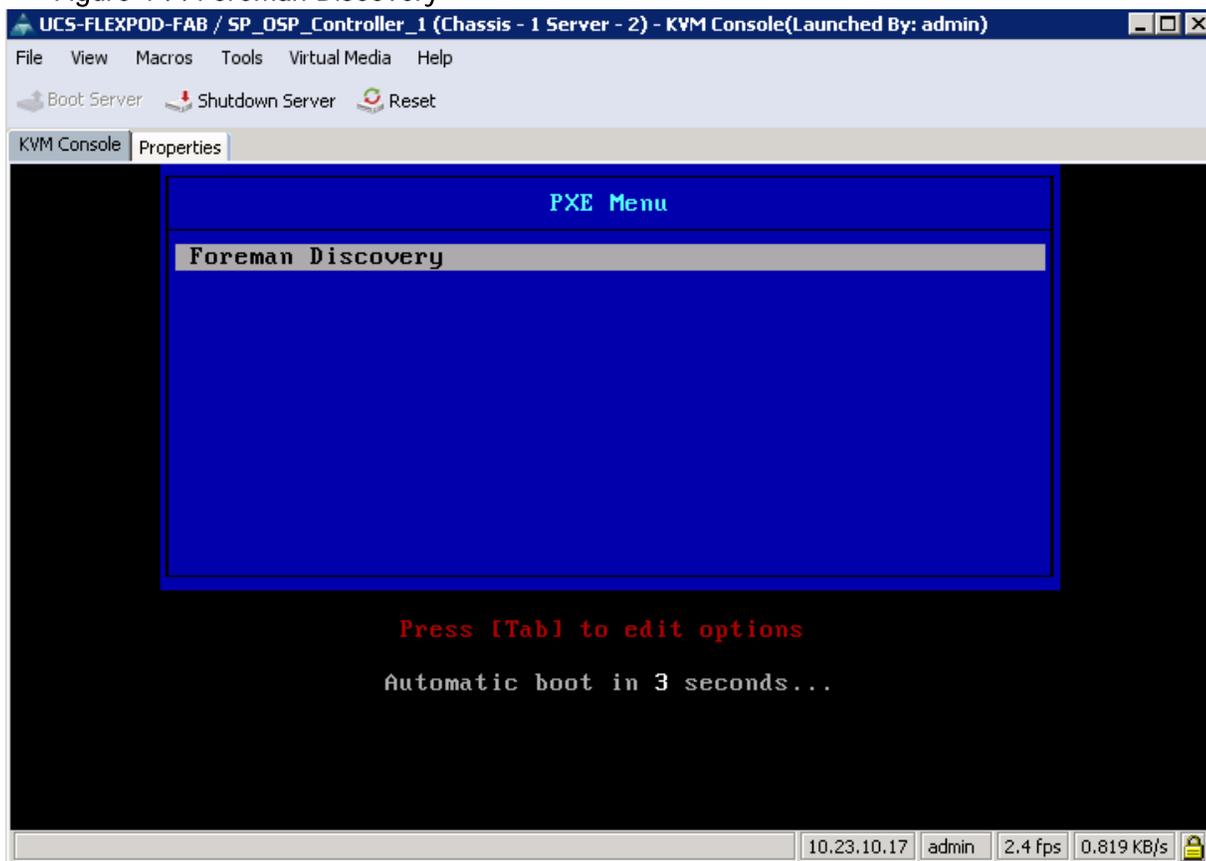
You can validate the Cisco eNIC Driver URL by navigating it into the browser.

Boot Servers into Discovery Mode

After all the managed hosts have been setup to boot through PXE on the provisioning network, booting them initiates a Foreman Discovery process (Figure 144).

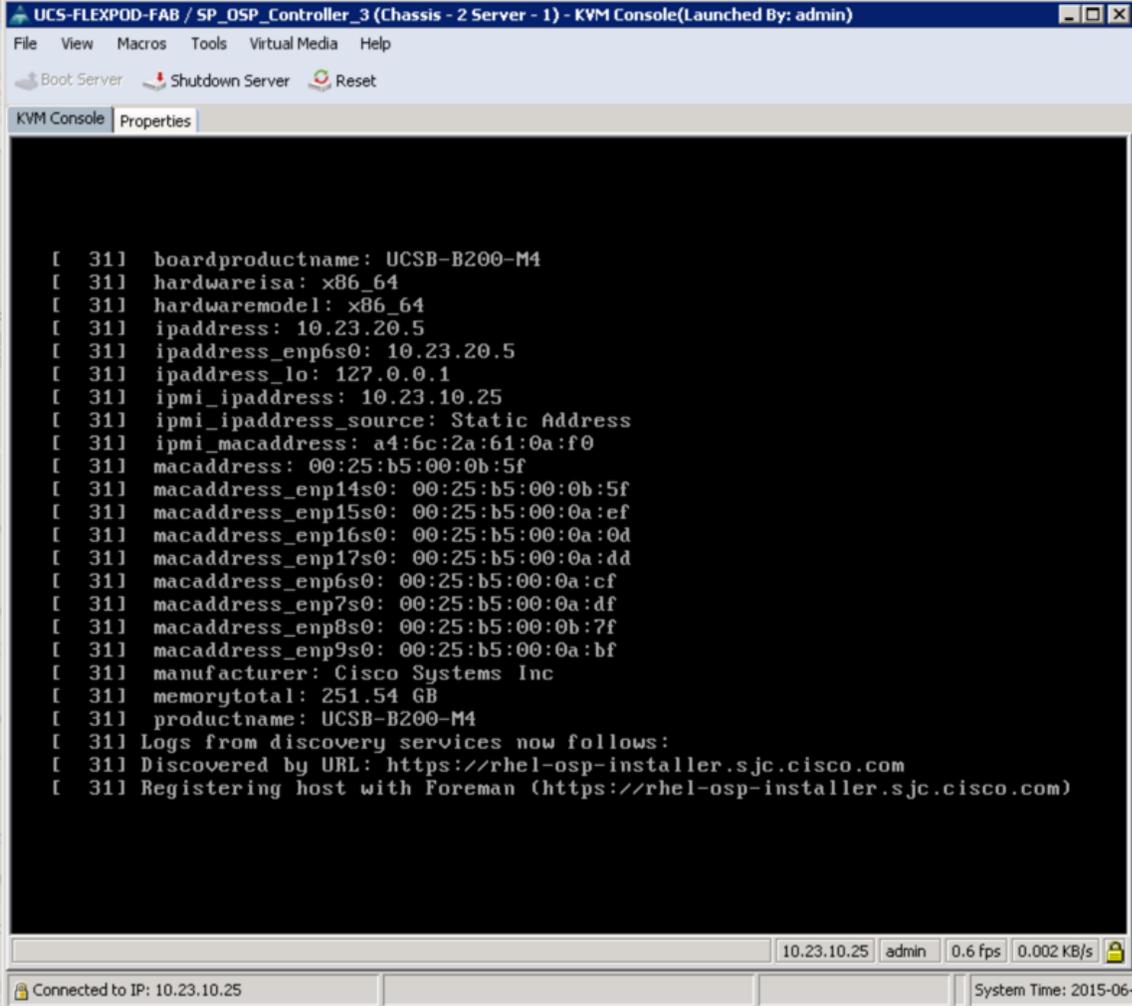
1. Log in to Cisco UCS Manager and reboot all the servers assigned for controller and compute hosts

Figure 144 Foreman Discovery



When the hosts have PXE booted properly, it will be discovered by the OSP installer and registered in Foreman. All the host interfaces can also be identified along with their MAC addresses (Figure 145).

Figure 145 Foreman Discovery and Host Registration to Foreman



```
[ 31] boardproductname: UCSB-B200-M4
[ 31] hardwareisa: x86_64
[ 31] hardwaremodel: x86_64
[ 31] ipaddress: 10.23.20.5
[ 31] ipaddress_enp6s0: 10.23.20.5
[ 31] ipaddress_lo: 127.0.0.1
[ 31] ipmi_ipaddress: 10.23.10.25
[ 31] ipmi_ipaddress_source: Static Address
[ 31] ipmi_macaddress: a4:6c:2a:61:0a:f0
[ 31] macaddress: 00:25:b5:00:0b:5f
[ 31] macaddress_enp14s0: 00:25:b5:00:0b:5f
[ 31] macaddress_enp15s0: 00:25:b5:00:0a:ef
[ 31] macaddress_enp16s0: 00:25:b5:00:0a:0d
[ 31] macaddress_enp17s0: 00:25:b5:00:0a:dd
[ 31] macaddress_enp6s0: 00:25:b5:00:0a:cf
[ 31] macaddress_enp7s0: 00:25:b5:00:0a:df
[ 31] macaddress_enp8s0: 00:25:b5:00:0b:7f
[ 31] macaddress_enp9s0: 00:25:b5:00:0a:bf
[ 31] manufacturer: Cisco Systems Inc
[ 31] memorytotal: 251.54 GB
[ 31] productname: UCSB-B200-M4
[ 31] Logs from discovery services now follows:
[ 31] Discovered by URL: https://rhel-osp-installer.sjc.cisco.com
[ 31] Registering host with Foreman (https://rhel-osp-installer.sjc.cisco.com)
```

2. After successful registration of host in Foreman, host will show up as “discovered hosts” in RHEL OpenStack Installer GUI on the discovered hosts page (Figure 146).
3. On the installer GUI main page, click Hosts > Discovered Hosts

Figure 146 Discovered Hosts

<input type="checkbox"/>	Name	Model	Subnet	Last facts upload	Provision
<input type="checkbox"/>	mac0025b5000a2f	UCSB-B200-M4	default (10.23.20.0/24)	about 2 hours ago	Provision
<input type="checkbox"/>	mac0025b5000a3d	UCSB-B200-M4	default (10.23.20.0/24)	about 2 hours ago	Provision
<input type="checkbox"/>	mac0025b5000a3e	UCSB-B200-M4	default (10.23.20.0/24)	about 2 hours ago	Provision
<input type="checkbox"/>	mac0025b5000a7d	UCSB-B200-M4	default (10.23.20.0/24)	about 1 hour ago	Provision
<input type="checkbox"/>	mac0025b5000a7e	UCSB-B200-M4	default (10.23.20.0/24)	about 2 hours ago	Provision
<input type="checkbox"/>	mac0025b5000aae	UCSB-B200-M4	default (10.23.20.0/24)	about 1 hour ago	Provision
<input type="checkbox"/>	mac0025b5000acf	UCSB-B200-M4	default (10.23.20.0/24)	about 2 hours ago	Provision
<input type="checkbox"/>	mac0025b5000ade	UCSB-B200-M4	default (10.23.20.0/24)	about 1 hour ago	Provision
<input type="checkbox"/>	mac0025b5000aff	UCSB-B200-M4	default (10.23.20.0/24)	about 2 hours ago	Provision

Displaying all 9 entries



The assigned hostname will be based on the MAC addresses of the network interface on the PXE network.



Avoid circumstances where more than one network had DHCP for the managed hosts (For example both external and provisioning networks were broadcasting DHCP). This may cause discovery failures.

OpenStack Deployment using the RHEL-OSP Installer

Prerequisites

The following are the prerequisites that must be performed before installation.

Modify Kickstart Default PXELinux for ip=ibft Boot Parameter

To modify Kickstart, complete the following steps:

1. Click Hosts → Provisioning Templates
2. Type kickstart in the search box and click Search.
3. Click Kickstart default PXELinux (Figure 147)

Figure 147 Modify Kickstart Default PXELinux

The screenshot shows the 'Provisioning Templates' page in the Red Hat Enterprise Linux OpenStack Platform Installer. The page has a search bar with 'kickstart' entered and a search icon. There are two buttons: 'New Template' and 'Build PXE Default'. Below is a table with the following data:

Name	Host group / Environment	Kind	Snippet	Locked	
Kickstart default		provision			Clone
Kickstart default iPXE		iPXE			Clone
Kickstart default PXELinux		PXELinux			Clone
kickstart_networking_setup			✓		Clone
Kickstart RHEL default		provision			Clone

At the bottom, it says 'Displaying all 5 entries'.

4. Add ip=ibft as shown in Figure 148.

Figure 148 Modify Kickstart default PXELinux – Add ip=ibft

```

- Fedora 18
- Fedora 19
- Fedora 20
- RedHat 4
- RedHat 5
- RedHat 6
- RedHat 7
%>
default linux
label linux
kernel <%= @kernel %>
<% if @host.operatingsystem.name == 'Fedora' and @host.operatingsystem.major.to_i > 16 -%>
append initrd=<%= @initrd %> ks=<%= foreman_url('provision')%> ks.device=bootif network ks.sendmac
<% elsif @host.operatingsystem.name != 'Fedora' and @host.operatingsystem.major.to_i >= 7 -%>
append initrd=<%= @initrd %> ks=<%= foreman_url('provision')%> network ks.sendmac biosdevname=0 ip=ibft
<% else -%>
append initrd=<%= @initrd %> ks=<%= foreman_url('provision')%> ksdevice=bootif network kssendmac
<% end -%>
IPAPPEND 2

```

A red arrow points to the line `append initrd=<%= @initrd %> ks=<%= foreman_url('provision')%> network ks.sendmac biosdevname=0 ip=ibft` with the text "Add ip=ibft".

5. Click Submit to save changes.

Modify Kickstart RHEL Default Cisco eNIC Driver Installation

To automate the installation of Cisco eNIC driver while deploying OpenStack hosts, complete the following steps:

1. Select Hosts→Provisioning templates. Search for kickstart.
2. Click “Kickstart RHEL default”
3. Insert following as shown in the Figure (149)

```

# Load Cisco enic
mount /boot 2>/dev/null | :

```

```
yum -t -y -e 0 localinstall http://rhel-osp-
installer.sjc.cisco.com:8080/CiscoEnic/kmod-enic-2.1.1.75-
rhel7u1.el7.x86_64.rpm
```

Figure 149 Modify Kickstart RHEL Default for Cisco eNIC Driver Installation

```
# We reuse our machine registerer instead
<%= snippet 'staypuft_client_bootstrap' %>

<% end -%>

sync

# Load Cisco enic
mount /boot 2>/dev/null | :
yum -t -y -e 0 localinstall http://rhel-osp-installer.sjc.cisco.com:8080/CiscoEnic/kmod-enic-2.1.1.75-rhel7u1.el7.x86_64.rpm

# Inform the build system that we are done.
echo "Informing Foreman that we are built"
wget -q -O /dev/null --no-check-certificate <%= foreman_url %>
```

4. Click Submit to save the changes.



Appendix E: Kickstart default PXELinux Configuration and Appendix F: Kickstart RHEL default Configuration shows the full contents of Kickstart default PXELinux and Kickstart RHEL default scripts for reference purpose. Extra care must be taken while modifying Kickstart scripts.

Modify Kickstart RHEL Default to Accommodate Multipathing

In order to have all of the provisioned Compute and Controller nodes come up with a total of (4) paths upon firstboot, a modification is need in the RHEL default kickstart file to instruct the `iscsiadm` daemon to log into the NetApp FAS by probing the iSCSI Boot Firmware table again. This results in (4) paths instead of (2), and is persistent across server reboots.

To modify Kickstart RHEL Default, complete the following steps:

1. Click Hosts → Provisioning Templates
2. Type kickstart in the search box and click Search.
3. Click Kickstart RHEL default. Figure 150 can be consulted above.
4. Enter the following before the 'kickstart_networking_setup' snipped as shown in Figure 150:

```
iscsiadm --mode discovery --type sendtargets --portal "$(< /sys/firmware/ibft/target0/ip-addr)"; iscsiadm --
mode node --login
```

5. Click Submit to save the changes.

Figure 150 Modify Kickstart RHEL default to Accommodate Multipathing

```

%post
logger "Starting anaconda <%= @host %> postinstall"
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on...
/usr/bin/chvt 3
(
iscsiadm --mode discovery --type sendtargets --portal "$(< /sys/firmware/ibft/target0/ip-addr)"; iscsiadm --mode node --login
<%= snippet 'kickstart_networking_setup' %>

#update local time
echo "updating system time"
/usr/sbin/ntpdate -sub <%= @host.params['ntp-server'] || '0.fedora.pool.ntp.org' %>
/usr/sbin/hwclock --systemd

```

Install Worksheet for Controller and Compute Roles

Gather controller and compute host interface information. Get the MAC address of PXE interface. All hosts are identified by PXE MAC address in the installer GUI. This information can be obtained by logging into UCS Manager and browse to PXE vNIC (For example: Servers > Service Profiles > SP_OSP_Controller_1 > vNICs > vNIC PXE). In the validation topology, following information were captured (Table 16).

Table 16 Sample Worksheet

Server Name	Service Profile	Network	Interfaces	Subnet
mac0025b5000a2f	SP_OSP_Controller_1	PXE/Provisioning	enp6s0	10.23.20.0/24
		Management	enp15s0	10.23.10.0/24
		Tenant	enp16s0	
		Storage	bond0 = enp9s0 + enp14s0	10.23.30.0/24
		MCAS	enp17s0	10.23.21.0/24
mac0025b5000aff	SP_OSP_Controller_2	PXE/Provisioning	enp6s0	10.23.20.0/24
		Management	enp15s0	10.23.10.0/24
		Tenant	enp16s0	
		Storage	bond0 = enp9s0 + enp14s0	10.23.30.0/24
		MCAS	enp17s0	10.23.21.0/24
mac0025b5000acf	SP_OSP_Controller_3	PXE/Provisioning	enp6s0	10.23.20.0/24
		Management	enp15s0	10.23.10.0/24
		Tenant	enp16s0	
		Storage	bond0 = enp9s0 + enp14s0	10.23.30.0/24
		MCAS	enp17s0	10.23.21.0/24

Server Name	Service Profile	Network	Interfaces	Subnet
mac0025b5000a7e	SP_OSP_Compute_1	PXE/Provisioning	enp6s0	10.23.20.0/24
		Management	enp15s0	10.23.10.0/24
		Tenant	enp16s0	
		Storage	bond0 = enp9s0 + enp14s0	10.23.30.0/24
		MCAS	enp17s0	10.23.21.0/24
mac0025b5000a3e	SP_OSP_Compute_2	PXE/Provisioning	enp6s0	10.23.20.0/24
		Management	enp15s0	10.23.10.0/24
		Tenant	enp16s0	
		Storage	bond0 = enp9s0 + enp14s0	10.23.30.0/24
		MCAS	enp17s0	10.23.21.0/24
mac0025b5000ade	SP_OSP_Compute_3	PXE/Provisioning	enp6s0	10.23.20.0/24
		Management	enp15s0	10.23.10.0/24
		Tenant	enp16s0	
		Storage	bond0 = enp9s0 + enp14s0	10.23.30.0/24
		MCAS	enp17s0	10.23.21.0/24
mac0025b5000aae	SP_OSP_Compute_4	PXE/Provisioning	enp6s0	10.23.20.0/24
		Management	enp15s0	10.23.10.0/24
		Tenant	enp16s0	
		Storage	bond0 = enp9s0 + enp14s0	10.23.30.0/24
		MCAS	enp17s0	10.23.21.0/24
mac0025b5000a7d	SP_OSP_Compute_5	PXE/Provisioning	enp6s0	10.23.20.0/24
		Management	enp15s0	10.23.10.0/24
		Tenant	enp16s0	
		Storage	bond0 = enp9s0 + enp14s0	10.23.30.0/24
		MCAS	enp17s0	10.23.21.0/24
mac0025b5000a3d	SP_OSP_Compute_6	PXE/Provisioning	enp6s0	10.23.20.0/24
		Management	enp15s0	10.23.10.0/24
		Tenant	enp16s0	
		Storage	bond0 = enp9s0 + enp14s0	10.23.30.0/24
		MCAS	enp17s0	10.23.21.0/24



Two additional compute nodes have been used in the reference architecture for test and validation purposes.

Setup Subnets

The following subnets need to be configured in the Installer GUI before you begin the installation. In the deployment guide, subnets are created ahead of deployment. However, there is an option to create subnets during the new deployment.

1. default subnet (Created by default by the installer based on PXE network information Figure 151)
2. Management subnet
3. Storage subnet
4. MCAS Subnet
5. Tenant subnet
6. External subnet

Figure 151 default Subnet Already Populated in the RHEL-OSP Installer

RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM INSTALLER Admin User

Monitor Hosts Configure Infrastructure OpenStack Installer Administer

Subnets

Filter... Search New Subnet

Name	Network address	Domains	VLAN ID	DHCP Capsule	
default	10.23.20.0/24	sjc.cisco.com		rhel-osp-installer.sjc.cisco.com	Delete

Displaying 1 entry

7. Contents of the default subnet can be verified by clicking Infrastructure→Subnets→default as shown in Figure 152.

Figure 152 Contents of default Subnet

Subnet	Domains	Capsules
Name *	<input type="text" value="default"/>	
Network address *	<input type="text" value="10.23.20.0"/>	
Network mask *	<input type="text" value="255.255.255.0"/>	Netmask for this subnet
Gateway address	<input type="text" value="10.23.20.2"/>	Optional: Gateway for this subnet
Primary DNS server	<input type="text" value="10.23.20.2"/>	Optional: Primary DNS for this subnet
Secondary DNS server	<input type="text"/>	Optional: Secondary DNS for this subnet
IPAM	<input type="text" value="DHCP"/>	IP Address auto suggestion mode for this subnet, DHCP works with DHCP proxy only
Start of IP range	<input type="text" value="10.23.20.3"/>	Optional: Starting IP Address for IP auto suggestion
End of IP range	<input type="text" value="10.23.20.250"/>	Optional: Ending IP Address for IP auto suggestion
VLAN ID	<input type="text"/>	Optional: VLAN ID for this subnet
Boot mode	<input type="text" value="DHCP"/>	Default boot mode for interfaces assigned to this subnet



Tenant and External subnets will not be assigned to any host interfaces. However, the Red Hat Enterprise Linux OpenStack Platform installer requires those two subnets to be created in order to proceed for the installation.

Log in to installer GUI, click Infrastructure > Subnets, and perform the following steps to create or modify subnets.

Create Management Subnet

1. Click **New Subnet** subnet button to create Management subnet.
2. Fill in the management subnet information. The following inputs (Figure 153) were incorporated in reference topology. Table 17 represents the worksheet needed to complete the Management network configuration.

Table 17 Management Subnet Worksheet

Name	Description	Value (Example)
Name	Name of the subnet	Management
Network address	Network address of Management subnet	10.23.10.0

Name	Description	Value (Example)
Network mask	Network mask for the Management subnet	255.255.255.0
Gateway address	Gateway address of the Management subnet	10.23.10.2
Primary DNS server	Primary DNS server	10.23.20.2 (Installer Server)
Secondary DNS server	Secondary DNS Server	-
IPAM	Automatically assign IP addresses to hosts	Internal DB
Start of IP range	Start of IP address range used for OpenStack Management network	10.23.10.51
End of IP range	End of IP address range used for OpenStack Management network	10.23.10.254
VLAN ID	VLAN Id of the Management subnet	10
Boot mode	Select the default boot mode for interfaces assigned to the subnet from the Boot mode list. This sets the boot protocol for client network interfaces. Select Static to manually assign IP address through the installer's IPAM Internal Database option.	Static

Figure 153 Create Management Subnet

Subnet	Domains	Capsules
Name *	<input type="text" value="Management"/>	
Network address *	<input type="text" value="10.23.10.0"/>	
Network mask *	<input type="text" value="255.255.255.0"/>	Netmask for this subnet
Gateway address	<input type="text" value="10.23.10.2"/>	Optional: Gateway for this subnet
Primary DNS server	<input type="text" value="10.23.20.2"/>	Optional: Primary DNS for this subnet
Secondary DNS server	<input type="text"/>	Optional: Secondary DNS for this subnet
IPAM	<input type="text" value="Internal DB"/>	
	IP Address auto suggestion mode for this subnet, DHCP works with DHCP proxy only	
Start of IP range	<input type="text" value="10.23.10.51"/>	Optional: Starting IP Address for IP auto suggestion
End of IP range	<input type="text" value="10.23.10.254"/>	Optional: Ending IP Address for IP auto suggestion
VLAN ID	<input type="text" value="10"/>	Optional: VLAN ID for this subnet
Boot mode	<input type="text" value="Static"/>	Default boot mode for interfaces assigned to this subnet

3. Click Domains tab.
4. Select domain (Figure 154).
5. Click Submit to save changes.

Figure 154 Select Domain

Subnet	Domains	Capsules
Domain	<input type="checkbox"/> Select All <input checked="" type="checkbox"/> sjc.cisco.com	Domains in which this subnet is part

Create Storage Subnet

1. Click subnet button to create the Storage subnet, which will eventually be used for the NFS traffic from Cinder.

2. Fill in the storage subnet information. Following inputs (Figure 155) were incorporated in reference topology. Table 18 represents the worksheet needed to complete the Storage network configuration.

Table 18 Storage Subnet Worksheet

Name	Description	Value (Example)
Name	Name of the subnet	Storage
Network address	Network address of Storage subnet	10.23.30.0
Network mask	Network mask for the Storage subnet	255.255.255.0
Gateway address	Gateway address of the Storage subnet	-
Primary DNS server	Primary DNS server	-
Secondary DNS server	Secondary DNS Server	-
IPAM	Automatically assign IP addresses to hosts	Internal DB
Start of IP range	Start of IP address range used for OpenStack Storage network	10.23.30.50
End of IP range	End of IP address range used for OpenStack Storage network	10.23.30.100
VLAN ID	VLAN Id of the Storage subnet	30
Boot mode	Select the default boot mode for interfaces assigned to the subnet from the Boot mode list. This sets the boot protocol for client network interfaces. Select Static to manually assign IP address through the installer's IPAM Internal Database option.	Static

3. Click Domains tab.
4. Select domain (Figure 155).
5. Click Submit to save changes.

Figure 155 Create Storage Subnet

Subnet	Domains	Capsules
Name *	<input type="text" value="Storage"/>	
Network address *	<input type="text" value="10.23.30.0"/>	
Network mask *	<input type="text" value="255.255.255.0"/>	Netmask for this subnet
Gateway address	<input type="text"/>	Optional: Gateway for this subnet
Primary DNS server	<input type="text"/>	Optional: Primary DNS for this subnet
Secondary DNS server	<input type="text"/>	Optional: Secondary DNS for this subnet
IPAM	<input type="text" value="Internal DB"/>	IP Address auto suggestion mode for this subnet, DHCP works with DHCP proxy only
Start of IP range	<input type="text" value="10.23.30.50"/>	Optional: Starting IP Address for IP auto suggestion
End of IP range	<input type="text" value="10.23.30.100"/>	Optional: Ending IP Address for IP auto suggestion
VLAN ID	<input type="text" value="30"/>	Optional: VLAN ID for this subnet
Boot mode	<input type="text" value="Static"/>	Default boot mode for interfaces assigned to this subnet

Create Subnet for OpenStack Management, Cluster Management, Admin API, Cluster Management (MCAS)

1. Click **New Subnet** subnet button to create the subnet, which will be used for OpenStack management, cluster management, admin API, and storage clustering traffic.
2. Fill in the subnet information. Following inputs (Figure 156) were incorporated in reference topology. Table 19 represents the worksheet needed to complete this network configuration.

Table 19 MCAS Subnet Worksheet

Name	Description	Value (Example)
Name	Name of the subnet	MCAS
Network address	Network address of MCAS subnet	10.23.21.0
Network mask	Network mask for the MCAS subnet	255.255.255.0
Gateway address	Gateway address of the MCAS subnet	-
Primary DNS server	Primary DNS server	-
Secondary DNS server	Secondary DNS Server	-
IPAM	Automatically assign IP addresses to hosts	Internal DB

Name	Description	Value (Example)
Start of IP range	Start of IP address range used for OpenStack MCAS network	10.23.21.2
End of IP range	End of IP address range used for OpenStack MCAS network	10.23.21.254
VLAN ID	VLAN Id of the MCAS subnet	21
Boot mode	Select the default boot mode for interfaces assigned to the subnet from the Boot mode list. This sets the boot protocol for client network interfaces. Select Static to manually assign IP address through the installer's IPAM Internal Database option.	Static

Figure 156 Create MCAS Subnet

Subnet

Domains

Capsules

Name *

MCAS

Network address *

10.23.21.0

Network mask *

255.255.255.0

Netmask for this subnet

Gateway address

Optional: Gateway for this subnet

Primary DNS server

Optional: Primary DNS for this subnet

Secondary DNS server

Optional: Secondary DNS for this subnet

IPAM

Internal DB
▼

IP Address auto suggestion mode for this subnet, DHCP works with DHCP proxy only

Start of IP range

10.23.21.2

Optional: Starting IP Address for IP auto suggestion

End of IP range

10.23.21.254

Optional: Ending IP Address for IP auto suggestion

VLAN ID

21

Optional: VLAN ID for this subnet

Boot mode

Static
▼

Default boot mode for interfaces assigned to this subnet

3. Click Domains tab.
4. Select domain (Figure 156).

- Click Submit to save changes.

Create Tenant Subnet

To create a tenant subnet, complete the following steps:

- Click **New Subnet** subnet button to create Tenant subnet.
- Fill in the Tenant subnet information. The following inputs (Figure 157) were incorporated in reference topology.
- Click Domains tab.
- Select domain (Figure 157).
- Click Submit to save changes.

Figure 157 Create Tenant Subnet

Subnet	Domains	Capsules
Name *	<input type="text" value="Tenant"/>	
Network address *	<input type="text" value="10.23.60.0"/>	
Network mask *	<input type="text" value="255.255.255.0"/>	Netmask for this subnet
Gateway address	<input type="text"/>	Optional: Gateway for this subnet
Primary DNS server	<input type="text"/>	Optional: Primary DNS for this subnet
Secondary DNS server	<input type="text"/>	Optional: Secondary DNS for this subnet
IPAM	<input type="text" value="Internal DB"/>	
	IP Address auto suggestion mode for this subnet, DHCP works with DHCP proxy only	
Start of IP range	<input type="text"/>	Optional: Starting IP Address for IP auto suggestion
End of IP range	<input type="text"/>	Optional: Ending IP Address for IP auto suggestion
VLAN ID	<input type="text"/>	Optional: VLAN ID for this subnet
Boot mode	<input type="text" value="DHCP"/>	Default boot mode for interfaces assigned to this subnet

Create External Subnet

- Click **New Subnet** subnet button to create External subnet.
- Fill in the External subnet information. The following inputs (Figure 158) were incorporated in reference topology.

3. Click Domains tab.
4. Select domain (Figure 158).
5. Click Submit to save changes.

Figure 158 Create External Subnet

Subnet	Domains	Capsules	
Name *	<input type="text" value="External"/>		
Network address *	<input type="text" value="<<external_network_address>>"/>		
Network mask *	<input type="text" value="255.255.255.0"/>	Netmask for this subnet	
Gateway address	<input type="text"/>	Optional: Gateway for this subnet	
Primary DNS server	<input type="text"/>	Optional: Primary DNS for this subnet	
Secondary DNS server	<input type="text"/>	Optional: Secondary DNS for this subnet	
IPAM	<input type="text" value="Internal DB"/>	IP Address auto suggestion mode for this subnet, DHCP works with DHCP proxy only	
Start of IP range	<input type="text"/>	Optional: Starting IP Address for IP auto suggestion	
End of IP range	<input type="text"/>	Optional: Ending IP Address for IP auto suggestion	
VLAN ID	<input type="text"/>	Optional: VLAN ID for this subnet	
Boot mode	<input type="text" value="Static"/>	Default boot mode for interfaces assigned to this subnet	

Subnets Summary

Figure 159 lists all the subnets created and used in deployment. This information is accessible through Infrastructure > Subnets.

Figure 159 Subnets List – Infrastructure → Subnets

Monitor ▾
Hosts ▾
Configure ▾
Infrastructure ▾
OpenStack Installer ▾

Subnets

▾

Name	Network address	Domains	VLAN ID
External	173.36.215.0/24	sjc.cisco.com	
Tenant	10.23.60.0/24	sjc.cisco.com	
Management	10.23.10.0/24	sjc.cisco.com	10
MCAS	10.23.21.0/24	sjc.cisco.com	21
Storage	10.23.30.0/24	sjc.cisco.com	30
default	10.23.20.0/24	sjc.cisco.com	

Displaying all 6 entries

Table 20 lists each traffic type within the subnet and the network range assigned in the reference architecture.

Table 20 Subnets and Each Traffic Types within the Subnets

Subnet Name	Traffic Type	Network Range	
		From	To
Default	PXE/Provisioning	10.23.20.3	10.23.20.254
MCAS	<ul style="list-style-type: none"> OpenStack Management OpenStack Cluster Management OpenStack Admin API OpenStack Storage Clustering 	10.23.21.2	10.23.21.254
Management	OpenStack Public API	10.23.10.51	10.23.10.254

Subnet Name	Traffic Type	Network Range	
Storage	OpenStack Cinder and Glance	10.23.30.50	10.23.30.100

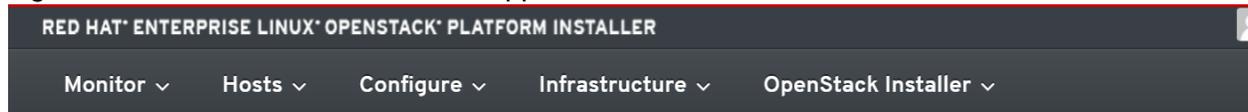


Ranges may vary depending on the overall size of the deployment.

Modify Puppet classes for Cisco Nexus 1000v VEM and VSM Module Parameters

1. Click Configure→Puppet classes
2. Type vem in the search box and click Search button.
3. Click neutron::agents::n1kv_vem (Figure 160)

Figure 160 Cisco Nexus 1000v VEM Puppet Class Parameters



Puppet classes

Class name	Environments and documentation	Host group	Hosts	Parameters
neutron::agents::n1kv_vem	production	base_RedHat_7/FLEXPOD_RHEL_OSP6/Controller and base_RedHat_7/FLEXPOD_RHEL_OSP6/Compute (Neutron)	0	14

Displaying 1 entry

4. Click “Smart Class Parameter” tab.
5. Click host mgmt intf
6. Click Override check box as shown in the Figure (161)

Figure 161 Enable Override for Cisco Nexus 1000v VEM Parameters

Monitor ▾ Hosts ▾ Configure ▾ Infrastructure ▾ OpenStack Installer ▾

Puppet Class **Smart Class Parameter** Smart Variables

Filter By Name @

enable

fastpath flood

host mgmt intf ✖

manage service

n1kv source

n1kv version

Puppet Environments

Parameter *

Description

Override

Whether the smart-variable should override value.

7. Select Override check box for the following parameters.

- a. n1kv vsm domain id
- b. n1kv vsm ip
- c. uplink profile



You will notice a flag icon on all the parameters that have been selected for override

- d. Click Submit to save the changes.
- e. Click Configure → Puppet classes
- f. Type vsm in the search box and click Search button.
- g. Click n1kv_vsm. Figure 162

Figure 162 Cisco Nexus 1000v VSM Puppet Class Parameters

RED HAT® ENTERPRISE LINUX® OPENSTACK® PLATFORM INSTALLER

Monitor ▾ Hosts ▾ Configure ▾ Infrastructure ▾ OpenStack Installer ▾

Puppet classes

vsm

Import from rhel-osp-installer.sjc.cisc

Class name	Environments and documentation	Host group	Hosts	Parameters
n1k_vsm	production		2	10
n1k_vsm::deploy	production		0	0
n1k_vsm::pkgprep_ovscfg	production		0	0
n1k_vsm::vsmprep	production		0	0

Displaying all 4 entries

8. Click “Smart Class Parameter” tab.
9. Select Override checkbox for the following parameters. Figure 163.
 - a. n1kv source
 - b. n1kv version
 - c. phy gateway
 - d. phy if bridge
 - e. vsm admin passwd
 - f. vsm domain id
 - g. vsm mgmt. gateway
 - h. vsm mgmt ip
 - i. vsm mgmt netmask
 - j. vsm role

Figure 163 Enable Override for Cisco Nexus 1000v VSM Parameters

The screenshot shows the OpenStack Installer interface with the following configuration for the 'n1kv version' parameter:

Puppet Environments	production
Parameter *	n1kv_version
Description	
Override	<input checked="" type="checkbox"/>
	Whether the smart-variable should ove
Parameter type	string
	Parameter Types
Default value	latest
	Value to use when there is no match

Deployment

The following steps will outline the HA deployment with three controller nodes and five compute hosts.

Create a new deployment by clicking OpenStack Installer → New deployment.

Deployment Settings

To specify deployment settings, complete the following steps:

1. Enter name of the deployment in the Name field (Figure 164).
2. Enter description (optional).
3. Leave the defaults selected for Networking as Neutron Networking, Messaging Provider as RabbitMQ, Platform as Red Hat Enterprise Linux OpenStack Platform 6 on RHEL 7, and Service Password as Generate random password for each service



It is recommended to use “Generate random password for each service” for a production deployment.

4. Specify Cisco Nexus 1000v repo <https://cmsg-yum-server.cisco.com/yumrepo> in Custom repo field (Figure 164).
5. Click Next.

Figure 164 Deployment Settings

Network Configuration

To configure the network, complete the following steps:

1. Assign the right traffic type to their respective subnets by dragging the traffic type from default to target subnet (Figure 165). If a traffic type is still under the default subnet, then it shares the network along with PXE/provisioning and others in the default subnet.
2. Drag the following:
 - a. Storage → Storage Subnet
 - b. Tenant → Tenant Subnet
 - c. External → External Subnet
 - d. Public API → Management Subnet
 - e. Management, Cluster Management, Admin API, Storage Clustering → MCAS Subnet

Figure 165 Assign Traffic Type to Subnets

The screenshot displays a configuration interface with a dark navigation bar at the top containing the following menu items: Monitor, Hosts, Configure, Infrastructure, and OpenStack Installer. Below the navigation bar, the interface is organized into several sections, each representing a different subnet:

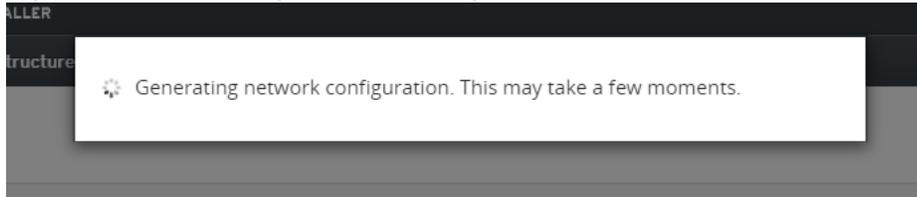
- External** - 173.36.215.0/24: Contains a button labeled "External" and a dashed rectangular box for assignment.
- Tenant** - 10.23.60.0/24: Contains a button labeled "Tenant" and a dashed rectangular box for assignment.
- Management** - 10.23.10.0/24: Contains a button labeled "Public API" and a dashed rectangular box for assignment.
- MCAS** - 10.23.21.0/24: Contains four buttons labeled "Storage Clustering", "Admin API", "Cluster Management", and "Management", followed by a dashed rectangular box for assignment.
- Storage** - 10.23.30.0/24: Contains a button labeled "Storage" and a dashed rectangular box for assignment.
- default** - 10.23.20.0/24: Contains a button labeled "Provisioning/PXE" and a dashed rectangular box for assignment.



Provisioning/PXE is assigned to default and cannot be assigned to a different subnet. By default, all traffic types except tenant and external as assigned to “default” subnet. Leaving all the traffic types in default subnet is NOT recommended for a production grade deployment. It is recommended to separate different traffic types through different subnets/VLANs.

3. Click Next. The Installer will generate network configuration for the deployment. Wait for this step to be completed (Figure 166)

Figure 166 Generating network configuration

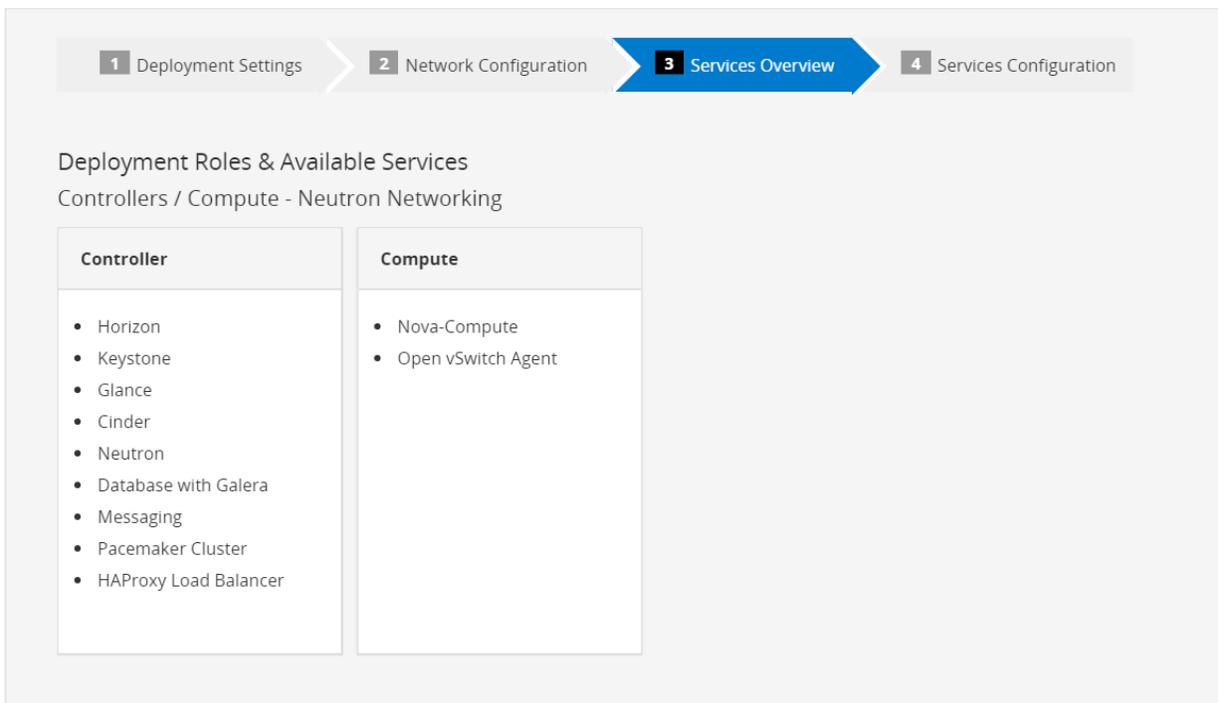


4. Display Services Overview for the Deployment Roles (Figure 167)
5. Click Next

Figure 167 Services Overview



New OpenStack Deployment

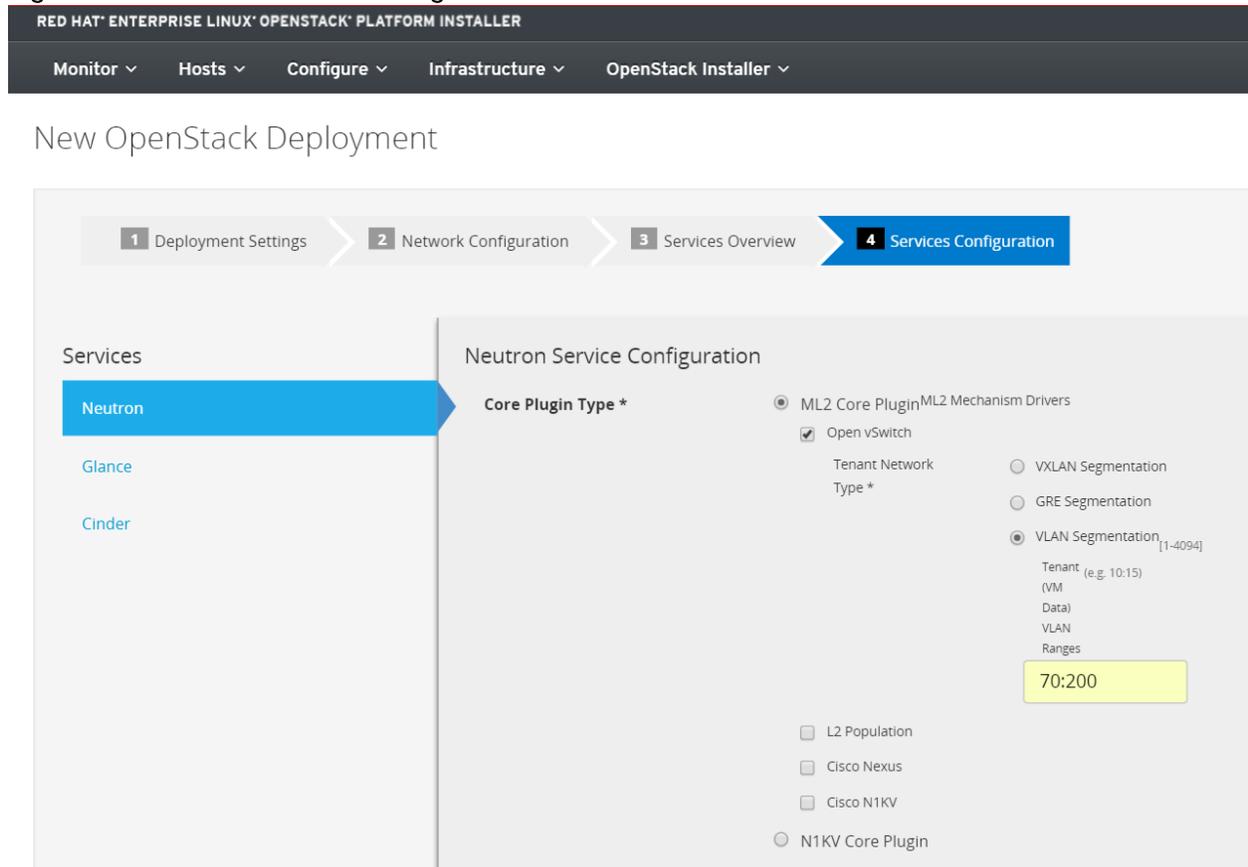


Services Configuration

To configure Neutron, Glance, and Cinder services, complete the following steps:

1. Neutron Service Configuration (Figure 168)
 - a. Select VLAN Segmentation in Tenant Network Type.
 - b. Enter 70:200 in Tenant (VM Data) VLAN Ranges

Figure 168 Neutron Service Configuration

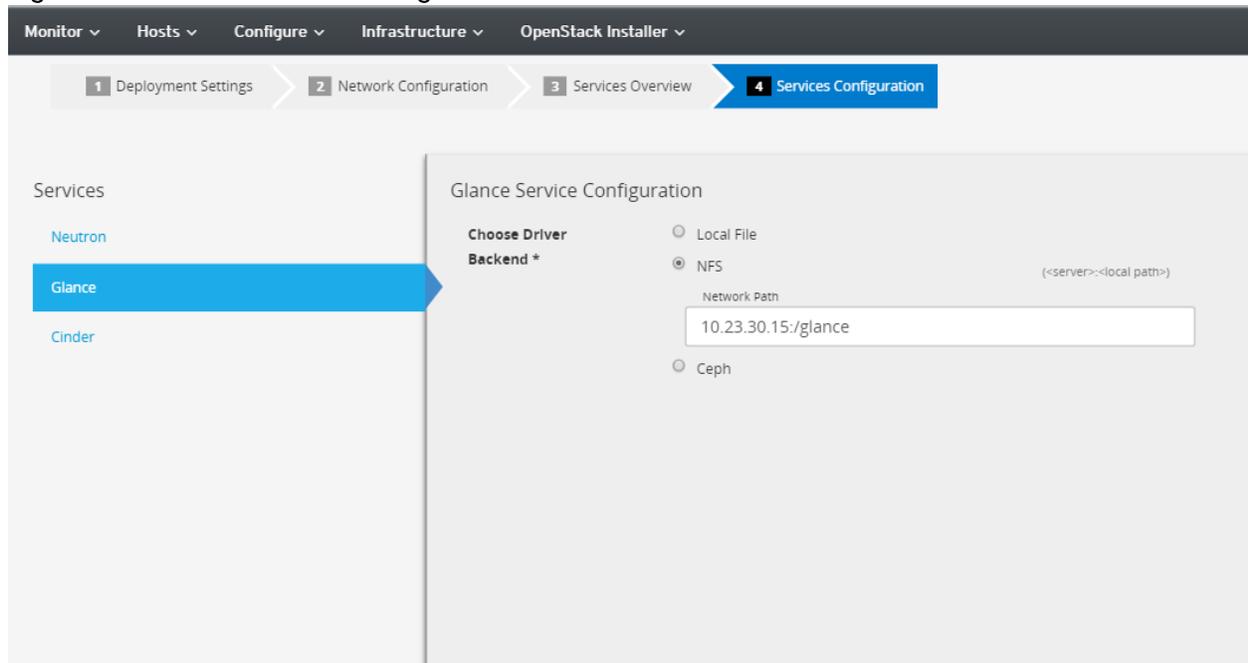


In the reference deployment, tenant VLAN ranges from 70 to 200 have been pre-provisioned in Cisco UCS Fabric Interconnect and Cisco Nexus 9000 switches.

2. Glance Service Configuration (Figure 169)

- a. Click Glance
- b. Select NFS as Glance Backend
- c. Specify the Network Path. For example, 10.23.30.15:/glance in the reference deployment.

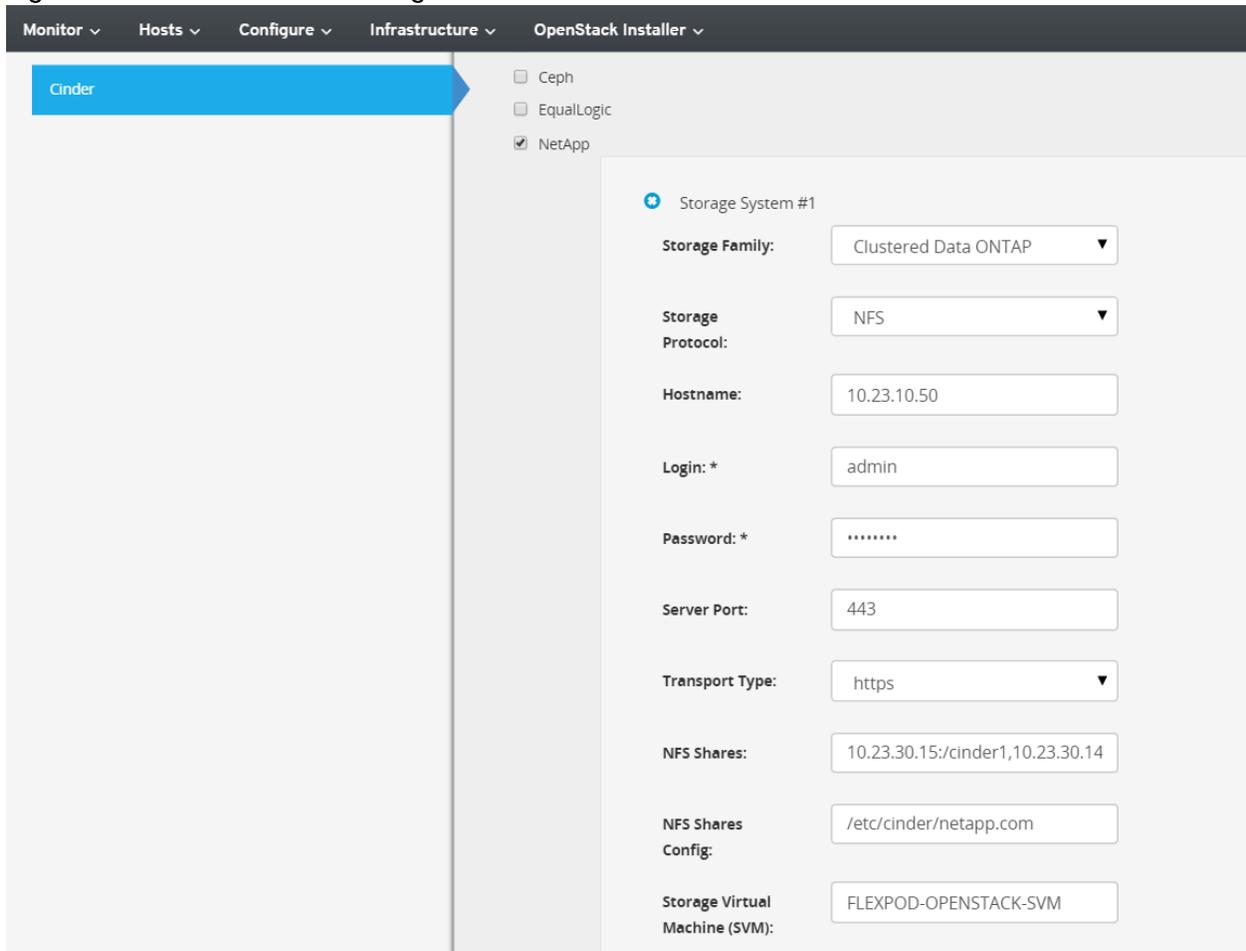
Figure 169 Glance Service Configuration



3. Cinder Service Configuration (Figure 170)

- a. Click Cinder.
- b. Check NetApp in Choose Driver Backend.
- c. Clustered Data ONTAP in Storage Family is pre-selected by default. Leave it as is.
- d. NFS is pre-selected for the Storage Protocol by default. Leave it as is.
- e. Specify the Hostname by providing the IP address of the FAS8040 cluster management LIF.
- f. Specify the admin username.
- g. Enter the admin password.
- h. Type 443 for Server Port.
- i. Select https in Transport Type drop-down list.
- j. Specify NFS shares (For Example:
10.23.30.15:/cinder1,10.23.30.14:/cinder2,10.23.30.15:/cinder3,10.23.30.14:/archived_data
).
- k. Type `/etc/cinder/netapp.conf` in the NFS Shares Config field.
- l. Specify the name of the Storage Virtual Machine (SVM) created for NFS.
- m. Click Submit.

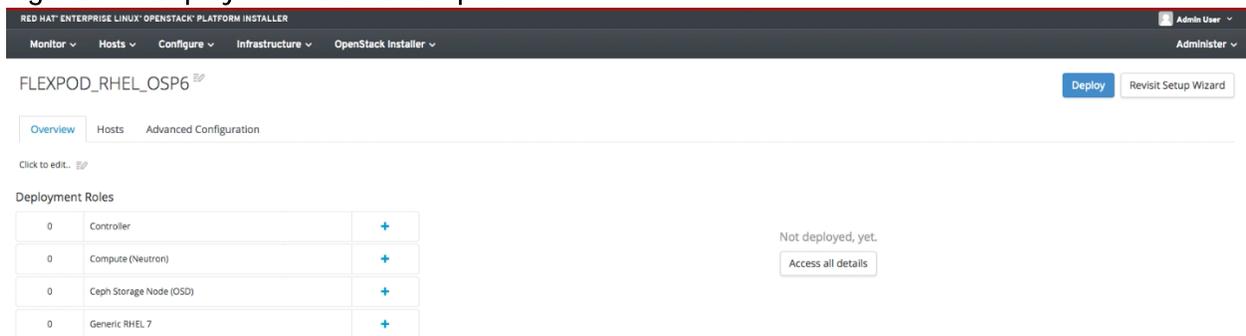
Figure 170 Cinder Service Configuration



Deployment Details

OpenStack service configuration (Figure 171) and deployment basics are completed.

Figure 171 Deployment Basics Completed

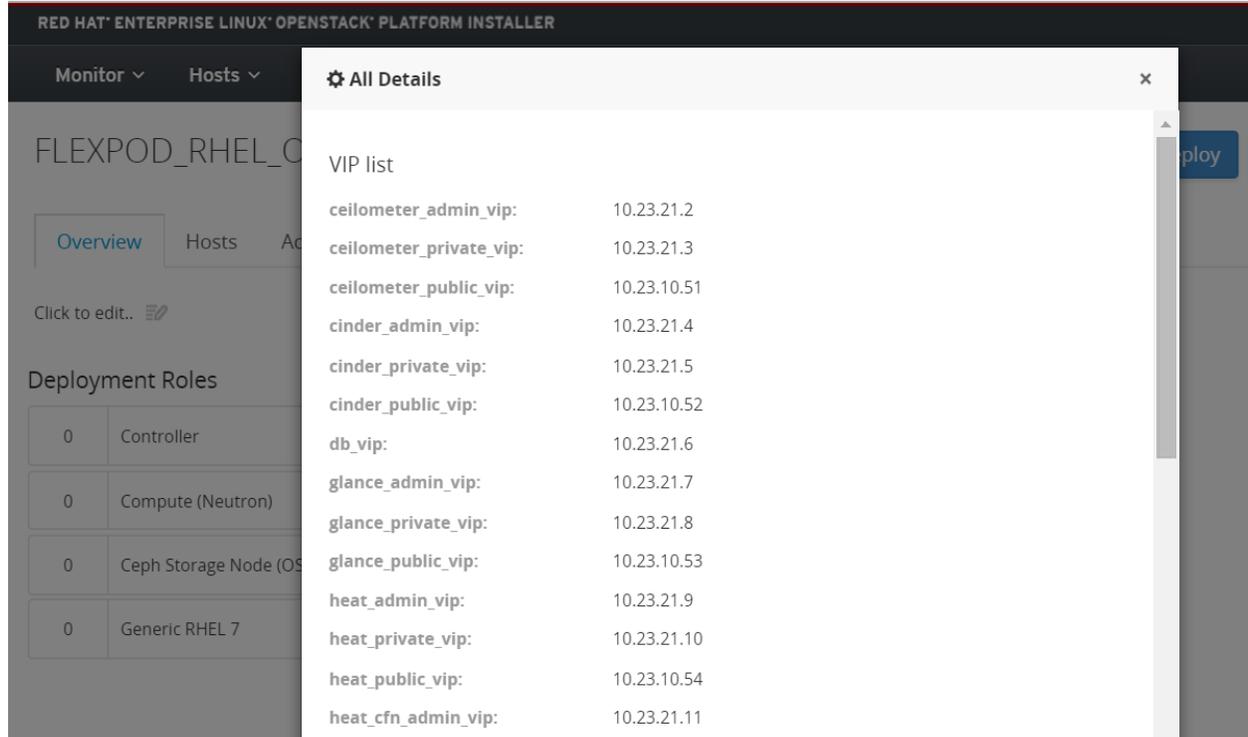


The “Access all details” button can be used to view (Figure 172):

- Allocated Virtual IPs (VIPs) by the subnet assigned to the deployment.
- Users for the service and their randomly generated passwords.

- Database for the services and their randomly generated passwords.

Figure 172 VIPs, Users and Database Passwords Details



See [Appendix G: Full List of VIPs, Users, and Database Passwords](#), for a full list of VIP, User and Database passwords for reference purposes.

Configure Host Group for Cisco Nexus 1000v

Prior to assigning hosts to the deployment, host groups need to be configured for Cisco Nexus 1000v deployment. To configure host groups, complete the following steps:

1. Select Configure → Host groups (Figure 173)

Figure 173 Host Groups

RED HAT® ENTERPRISE LINUX® OPENSTACK® PLATFORM INSTALLER

Monitor ▾ Hosts ▾ Configure ▾ Infrastructure ▾ OpenStack Installer ▾

Host Groups

Filter ...

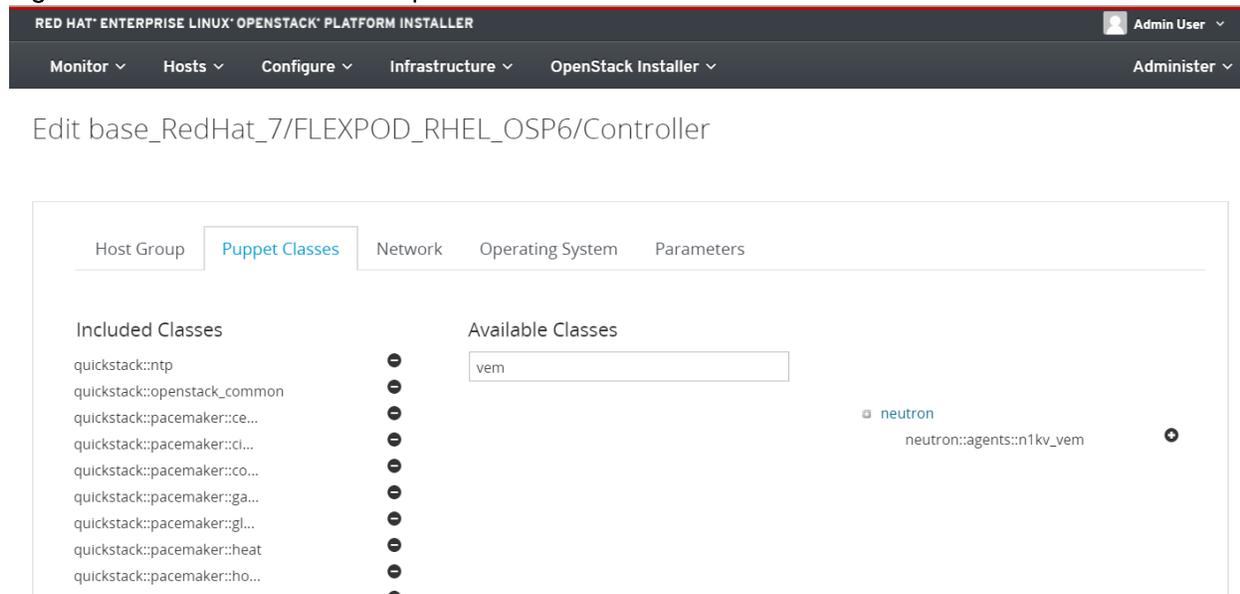
Name
base_RedHat_7
base_RedHat_7/FLEXPOD_RHEL_OSP6
base_RedHat_7/FLEXPOD_RHEL_OSP6/Ceph Storage Node (OSD)
base_RedHat_7/FLEXPOD_RHEL_OSP6/Compute (Neutron)
base_RedHat_7/FLEXPOD_RHEL_OSP6/Controller
base_RedHat_7/FLEXPOD_RHEL_OSP6/Generic RHEL 7

Displaying all 6 entries

To configure controller and compute Host group for Cisco Nexus 1000v Virtual Ethernet Module (VEM), complete the following steps:

1. Configure controller host group. Click Controller (Figure 174)
 - a. Select Puppet Classes tab.
 - b. Type vem in Available Classes text box.
 - c. Select neutron
 - d. Click  to add the neutron::agents::n1kv_vem (0).
 - e. neutron::agents::n1kv_vem will be added in Included Classes.

Figure 174 Controller Host Group - Add Cisco Nexus 1000v VEM Class



- f. Click Parameters tab
- g. Override the following parameters and configure the appropriate default value as shown in 0. Use the worksheet (Table 21) applicable to your environment.

Table 21 Controller Parameters for Cisco Nexus 1000v VEM Configuration Worksheet

Puppet Class	Parameter Name	Description	Example
neutron::agents::n1kv_vem	n1kv_vsm_domain_id	Domain ID of the VSM. The default is 1000 and the value should be between 1 and 1023	1000
neutron::agents::n1kv_vem	n1kv_vsm_ip	IP Address of the Virtual Supervisor Module(VSM). The default is 127.0.0.1	10.23.10.34
neutron::agents::n1kv_vem	host_mgmt_intf	Management interface of the node where the VEM is installed. The default is eth0	enp15s0.10
quickstack::pacemaker::neutron	n1kv_vsm_password	Administrative password for Cisco Nexus 1000v	
quickstack::pacemaker::neutron	ml2_mechanism_drivers	Name of the Neutron ML2 mechanism driver used	["cisco_n1kv"]
quickstack::pacemaker::neutron	ml2_tenant_network_types	Specify tenant network type.	["vlan"]
quickstack::pacemaker::neutron	n1kv_vsm_ip	IP Address of the Virtual Supervisor Module(VSM).	10.23.10.34

Puppet Class	Parameter Name	Description	Example
quickstack::packemaker::neutron	n1kv_vsm_username	Nexus 1000v user name	admin
quickstack::packemaker::neutron	ml2_type_drivers	ML2 plugin drivers	["vlan"]
quickstack::packemaker::neutron	l3_ha		False
neutron::agents::n1kv_vem	uplink_profile	<p>Uplink interface(s) that will be managed by the VEM. You must also specify the uplink port profile that configures that interfaces. The default is undefined (empty).</p> <hr/> <p> The uplink interface cannot be the same as the management interface on the compute node.</p> <hr/>	enp16s0: uplink



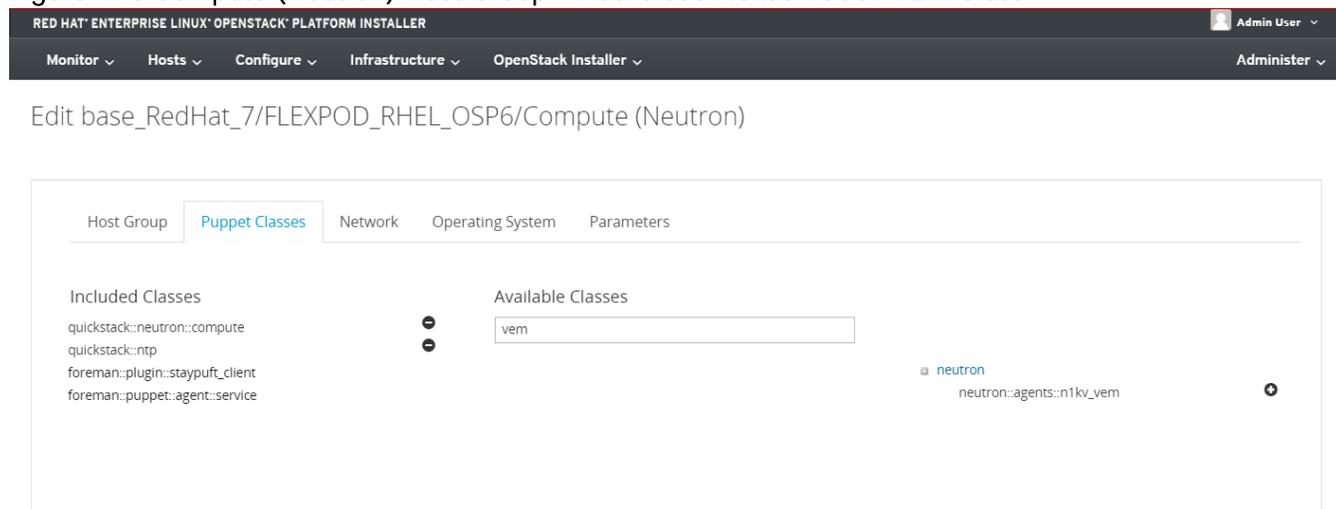
Extra care MUST be taken for the formats of the overridden parameters. Make sure to have an space in the uplink_profile parameter between the colon and the name of the uplink profile.

Figure 175 Controller Host Group – Override Parameters for Cisco Nexus 1000v Configuration

Monitor ▾	Hosts ▾	Configure ▾	Infrastructure ▾	OpenStack Installer ▾
				swift_storage_ips
Host group parameters				
neutron::agents::n1kv_vem	host_mgmt_intf	enp15s0.10		
neutron::agents::n1kv_vem	n1kv_vsm_domain_id	1000		
neutron::agents::n1kv_vem	n1kv_vsm_ip	10.23.10.34		
neutron::agents::n1kv_vem	uplink_profile	enp16s0: uplink		
quickstack::pacemaker::neutron	ml2_mechanism_driver	["cisco_n1kv"]		
quickstack::pacemaker::neutron	ml2_tenant_network_tj	["vlan"]		
quickstack::pacemaker::neutron	n1kv_vsm_ip	10.23.10.34		
quickstack::pacemaker::neutron	n1kv_vsm_password	enter n1kv vsm password here		
quickstack::pacemaker::neutron	n1kv_vsm_username	admin		
quickstack::pacemaker::neutron	ml2_type_drivers	["vlan"]		
quickstack::pacemaker::neutron	l3_ha	False		

- h. Click Submit  to save changes.
2. Configure Compute host group. Click Compute (Neutron) Figure 176.
 - a. Select Puppet Classes tab.
 - b. Type vem in Available Classes text box.
 - c. Select neutron
 - d. Click  to add the neutron::agents::n1kv_vem (Figure 176).
 - e. neutron::agents::n1kv_vem will be added in Included Classes.

Figure 176 Compute (Neutron) Host Group – Add Cisco Nexus 1000v VEM Class



- f. Click Parameters tab
- g. Override the following parameters and configure the appropriate default value as shown in Figure 177. Use Table 22 worksheet applicable to your environment.

Table 22 Compute Parameters for Nexus 1000v VEM Configuration Worksheet

Puppet Class	Parameter Name	Description	Example
neutron::agents::n1kv_vem	n1kv_vsm_domain_id	Domain ID of the VSM. The default is 1000 and the value should be between 1 and 1023	1000
neutron::agents::n1kv_vem	n1kv_vsm_ip	IP Address of the Virtual Supervisor Module(VSM). The default is 127.0.0.1	10.23.10.34
neutron::agents::n1kv_vem	host_mgmt_intf	Management interface of the node where the VEM is installed. The default is eth0	enp15s0.10
neutron::agents::n1kv_vem	uplink_profile	Uplink interface(s) that will be managed by the VEM. You must also specify the uplink port profile that configures that interfaces. The default is undefined (empty). <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  The uplink interface cannot be the same as the management interface on the compute node. </div>	enp16s0: uplink

Figure 177 Compute (Neutron) Host Group – Override Parameters for Cisco Nexus 1000v Configuration

Host group parameters

neutron::agents::n1kv_vem	uplink_profile	enp16s0: uplink
neutron::agents::n1kv_vem	host_mgmt_intf	enp15s0.10
neutron::agents::n1kv_vem	n1kv_vsm_domain_	1000
neutron::agents::n1kv_vem	n1kv_vsm_ip	10.23.10.34

+ Add Parameter

Cancel **Submit**

h. Click Submit **Submit** to save changes.

Assign Hosts to the Deployment

After configuring the host group for Cisco Nexus 1000v, assign managed hosts to different deployment roles. Designated hosts are determined by the PXE interface MAC address. To assign hosts to the deployment, complete the following steps:

1. Select OpenStack Installer → Deployments → FLEXPOD_RHEL_OSP6
2. In the Overview tab, click **+** to add controller in the Deployment Roles (Figure 178)
3. Select three servers for controller role identified by their MAC addresses assigned to PXE interface. Refer to Table 16 in the Install Worksheet for Controller and Compute Hosts section.
4. Click Assign Hosts button.

Figure 178 Add Controllers – Deployment Roles

The screenshot shows the OpenStack Platform Installer interface for the 'FLEXPOD_RHEL_OSP6' environment. The 'Deployment Roles' section is active, showing a list of roles: Controller, Compute (Neutron), Ceph Storage Node (OSD), and Generic RHEL 7. The 'Unassigned Hosts' window is open, displaying a table of hosts that are not yet assigned to any role.

Name	NICs	Storage	Managed?	IP Address
mac0025b5000a2f	enp14s0 enp15s0 enp16s0 enp6s0 enp7s0 enp8s0 enp9s0		-	10.23.20.5
mac0025b5000a2f	enp14s0 enp15s0 enp16s0 enp6s0 enp7s0 enp8s0 enp9s0		-	10.23.20.3
rhel-osp-installer.sjc.cisc...	enp13s0 enp14s0 enp6s0 ibft0 ibft1	sda: Unknown sdb: Unknown sdc: Unknown sdd: Unknown	-	127.0.0.1
	enp14s0 enp15s0 enp16s0			

- In the Overview tab, click **+** to add compute in the Deployment Roles (Figure 179)
- Select five servers for compute role identified by their MAC addresses assigned to PXE interface. Refer to Table 16 the in Install Worksheet for Controller and Compute Hosts section.



The service profile named SP_OSP_Compute_6 was intentionally left unassigned. This host will be added later in the subsequent section as a use case by showing how to scale the environment by adding more Compute hosts to the Compute deployment role after the initial deployment has completed.

- Click Assign Hosts button.

Figure 179 Add Compute (Neutron) – Deployment Roles

The screenshot shows the OpenStack Installer interface for deployment FLEXPOD_RHEL_OSP6. The 'Hosts' tab is active, displaying a table of deployment roles and a modal window for unassigned hosts.

Deployment Roles Table:

Count	Role	Assigned	Action
0	Controller	3	+
0	Compute (Neutron)	0	+
0	Ceph Storage Node (OSD)	0	+
0	Generic RHEL 7	0	+

Unassigned Hosts Table:

Name	NICs	Storage	Managed?	IP Address
<input type="checkbox"/> rhel-osp-installer.sjc.cisc...	enp13s0 enp14s0 enp6s0 ibft0 ibft1	sda: Unknown sdb: Unknown sdc: Unknown sdd: Unknown	-	127.0.0.1
<input type="checkbox"/> mac0025b5000a7e	enp14s0 enp15s0 enp16s0 enp6s0 enp7s0 enp8s0 enp9s0		-	10.23.20.6
<input type="checkbox"/> mac0025b5000ade	enp14s0 enp15s0 enp16s0 enp6s0 enp7s0 enp8s0 enp9s0		-	10.23.20.8
<input type="checkbox"/>	enp14s0 enp15s0 enp16s0			

Configure Host Networks

To configure host networks, complete the following steps:

1. Select the Hosts tab (OpenStack Installer → Deployments → FLEXPOD_RHEL_OSP6 → Hosts).
2. Select the Assigned sub-tab. Number in the parenthesis shows the total number of hosts assigned including both Controller and Compute roles (Figure 180).
3. Select all hosts by clicking the check box.
4. Select the Configure Networks button .

Figure 180 Assigned Hosts

Name	Deployment Role	CPUs (cores)	Memory (GB)	Storage	NICs (Subnet)	IP Address
mac0025b500bae.sjc.cisco.com	Compute (Neutron)				enp6s0 (default) enp14s0 () enp15s0 () enp16s0 () enp7s0 () enp8s0 () enp9s0 ()	10.23.20.8
mac0025b500ba3e.sjc.cisco.com	Compute (Neutron)				enp6s0 (default) enp14s0 () enp15s0 () enp16s0 () enp7s0 () enp8s0 () enp9s0 ()	10.23.20.7
mac0025b500baae.sjc.cisco.com	Compute (Neutron)				enp6s0 (default) enp14s0 () enp15s0 () enp16s0 () enp7s0 ()	10.23.20.9



Multiple hosts can be selected for configuring networks to avoid repetitive entries. This is true, provided the available network device names and final VLAN assignments are identical between roles. Since we are using Service Profile Templates in this deployment, this is true and saves operator entry.

- Bond enp14s0 with enp9s0 as shown in Figure (181). Bond0 interface will be created. Keep the bonding mode active-backup.

Figure 181 Bond Interfaces Assigned to Storage Traffic – enp14s0 and enp9s0

- Drag the following networks to their respective network interfaces as below:
 - Storage (vlan: 30) → bond0 (enp14s0, enp9s0)
 - Management (vlan: 10) → enp15s0
 - MCAS (vlan: 21) → enp17s0
 - Leave Tenant and External unassigned.



It is not necessary to assign all the subnets.



enp16s0 has been configured as an uplink in Cisco Nexus 1000v to trunk tenant VLANs for VM data traffic in controller and host group parameter `neutron::agents::n1kv_vem → uplink_profile`

- e. After the assignment has been completed, they appear as shown in Figure 182.
- f. Select Done after verifying the assignment of network interfaces.

Figure 182 Complete View of Network Interfaces

Configure Networks

Network Interfaces

enp6s0

default
Provisioning/PXE

bond0 (enp14s0, enp9s0)

Storage (vlan: 30)
Storage

enp15s0

Management (vlan: 10)
Public API

enp16s0

enp17s0

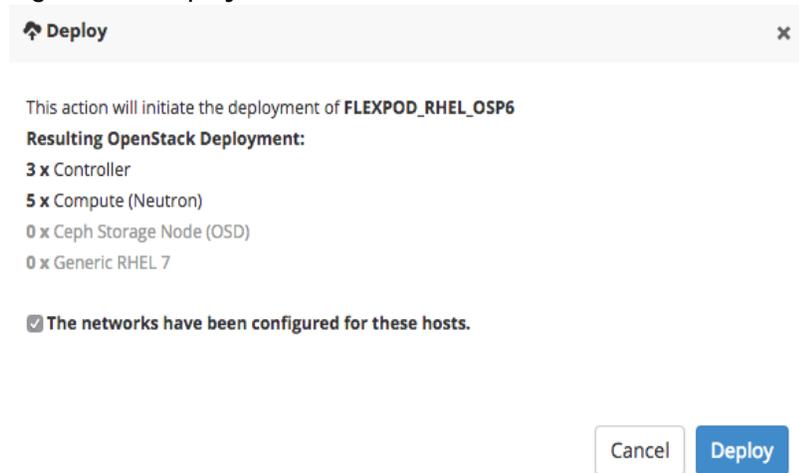
MCAS (vlan: 21)
Management + Cluster Management + Admin API + Storage Clustering

Begin Deployment

To begin the deployment, complete the following steps:

1. After all the network assignments have been confirmed, deployment can be initiated by selecting the **“deploy” button in the top right corner of the selected deployment screen**. Deploy confirmation screen will pop-up (Figure 183).
2. Confirm the deployment by selecting the check box for **“The networks have configured for these hosts.”** and select **“Deploy”**.

Figure 183 Deploy Confirmation



It is a good practice to verify and ensure all the networks are properly configured before selecting “Deploy” button. If there is misrepresentation of network, fixing networks post provisioning is a time consuming and daunting process.

It is absolutely essential to ensure that IPs are assigned to the correct subnet and network port, fall within the subnet IP range, and are not duplicated elsewhere in the environment.

Monitoring Deployment Status

To monitor the deployment status, complete the following steps:

1. It will take several minutes to finish the deployment. Meanwhile, progress of the deployment can be monitored. There are several methods to monitor the progress of the deployment. It can be monitored from the progress bar, Task tab, Dynflow console, and also through Reports (Monitor → Reports).
2. Watch the status of the deployment in the Task tab (Figure 184) by clicking the **“Show more details”**. Raw tab is also useful to watch the progress of each host assigned to the deployment (Figure 184).

Figure 184 Deployment Status

RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM INSTALLER Admin User

Monitor Hosts Configure Infrastructure OpenStack Installer Administer

Deployment Status Return to deployment

Task Running Steps Errors Locks Raw

Start auto-reloading
Dynflow console
Resume
Unlock
Force Unlock

Id: ac3ee093-ee6c-488c-813d-1b5e6ce6abcc
 Label: Actions::Staypuft::Deployment::Deploy
 Name: Deploy
 Owner: admin
 Started at: 2015-06-29 18:12:04 UTC
 Ended at:
 State: running
 Result: pending
 Params: Flexpod_RHEL_OSP6

98%

Output:

Controller 95%, Compute (Neutron) 100%

Figure 185 Deployment Status – Raw tab

Monitor Hosts Configure Infrastructure OpenStack Installer

Task Running Steps Errors Locks Raw

Raw input:

```
{ "id" => 61, "name" => "Flexpod_RHEL_OSP6", "current_user_id" => 3 }
```

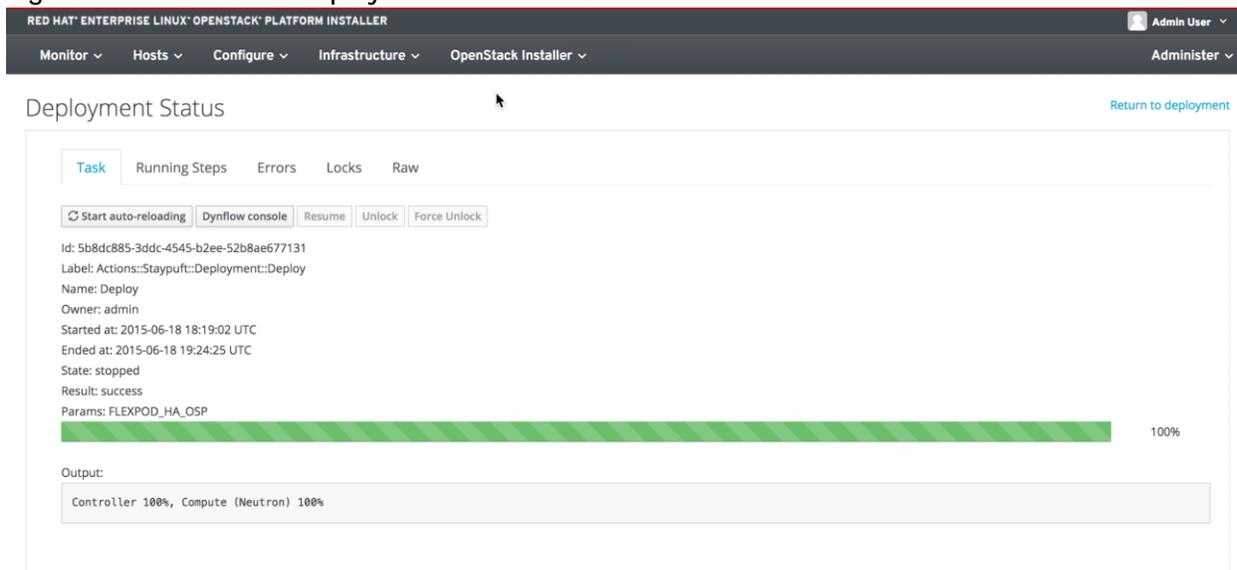
Raw output:

```
[ { :id => "303",
  :name => "Controller",
  :progress => 0.9505376344086022,
  :hosts =>
    [ { :id => "147",
      :name => "mac0025b5000a2f.sjc.cisco.com",
      :progress => 0.8516129032258064,
      { :id => "151", :name => "mac0025b5000aff.sjc.cisco.com", :progress => 1 },
      { :id => "153", :name => "mac0025b5000acf.sjc.cisco.com", :progress => 1 } ] },
  { :id => "304",
    :name => "Compute (Neutron)",
    :progress => 1.0,
    :hosts =>
      [ { :id => "149", :name => "mac0025b5000a3e.sjc.cisco.com", :progress => 1 },
        { :id => "150", :name => "mac0025b5000ade.sjc.cisco.com", :progress => 1 },
        { :id => "152", :name => "mac0025b5000aae.sjc.cisco.com", :progress => 1 },
        { :id => "154", :name => "mac0025b5000a7d.sjc.cisco.com", :progress => 1 },
        { :id => "155", :name => "mac0025b5000a3d.sjc.cisco.com", :progress => 1 },
        { :id => "148", :name => "mac0025b5000a7e.sjc.cisco.com", :progress => 1 } ] ] ] ]
```

External Id: 769cd150-6afd-4fe7-a6e0-bec7e8c303b4

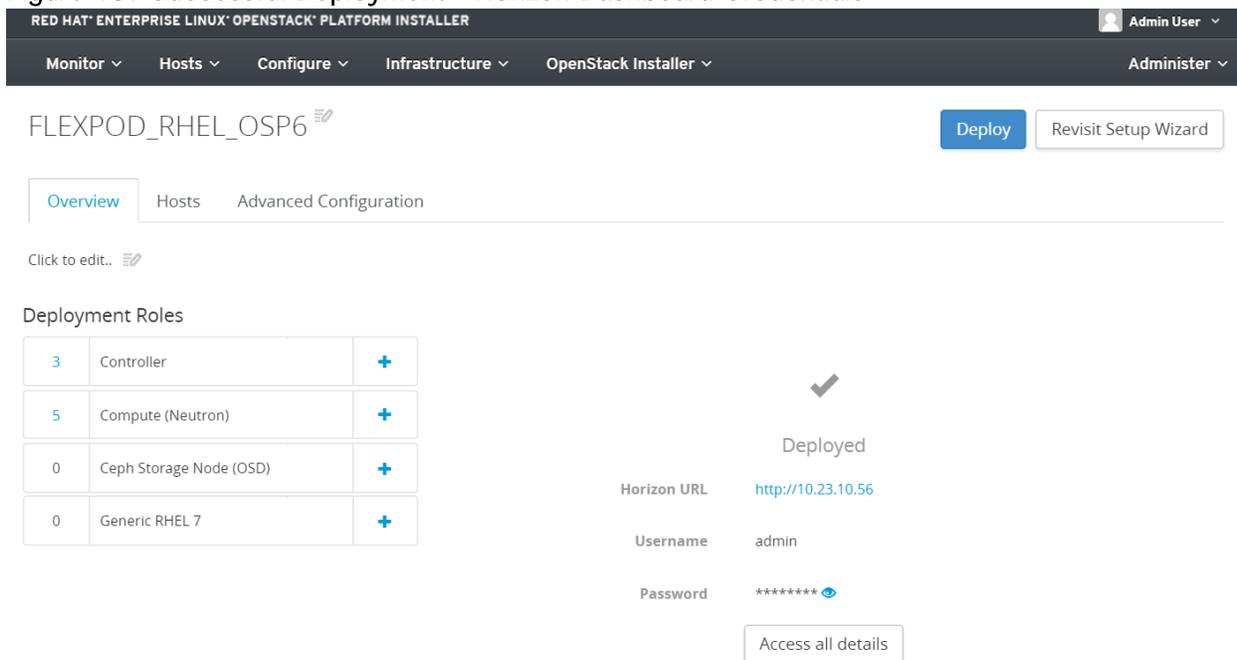
- When the deployment is successfully completed, status bar will display successful completion (Figure 186).

Figure 186 Successful Deployment – Horizon Dashboard Credentials



- Horizon dashboard IP and login credentials will be available on the deployment screen (Figure 187).

Figure 187 Successful Deployment – Horizon Dashboard Credentials



Post-Deployment (Required)

After the OpenStack deployment is finished and all the Controller and Compute nodes have been provisioned successfully, the remaining steps MUST be performed to have a fully functional environment.

Deploy Cisco Nexus 1000v Virtual Supervisor Module (VSM)

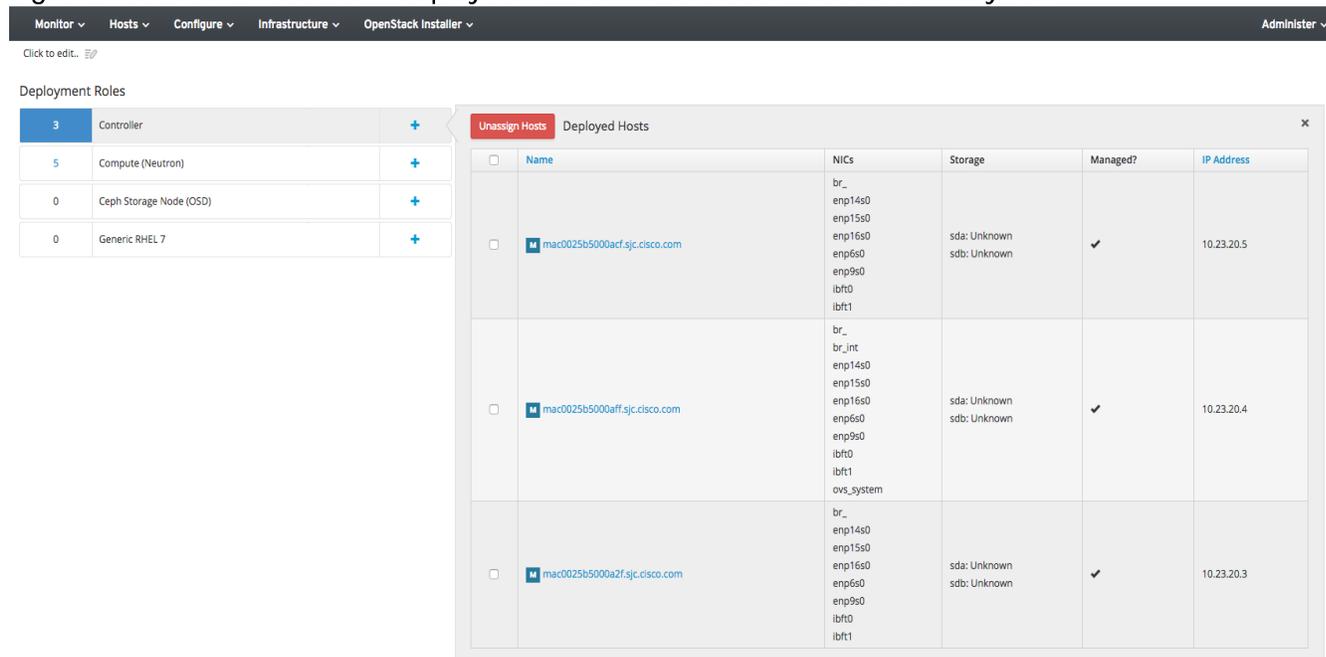
After the deployment is successfully completed, it is important to deploy Cisco Nexus 1000v Virtual Supervisor Module (VSM) in two of the three controllers. For more redundancy and operational efficiency, two VSM modules are deployed across two chassis. Following controller hosts are selected for VSM active and VSM hot standby modules (Table 23).

Table 23 Cisco Nexus 1000v VSM Placement

Server Name	Service Profile	Server Location	Nexus 1000v VSM Role
mac0025b5000aff	SP_OSP_Controller_2	Chassis 1/Blade 1	Active
mac0025b5000acf	SP_OSP_Controller_3	Chassis 2/Blade 1	Hot Standby

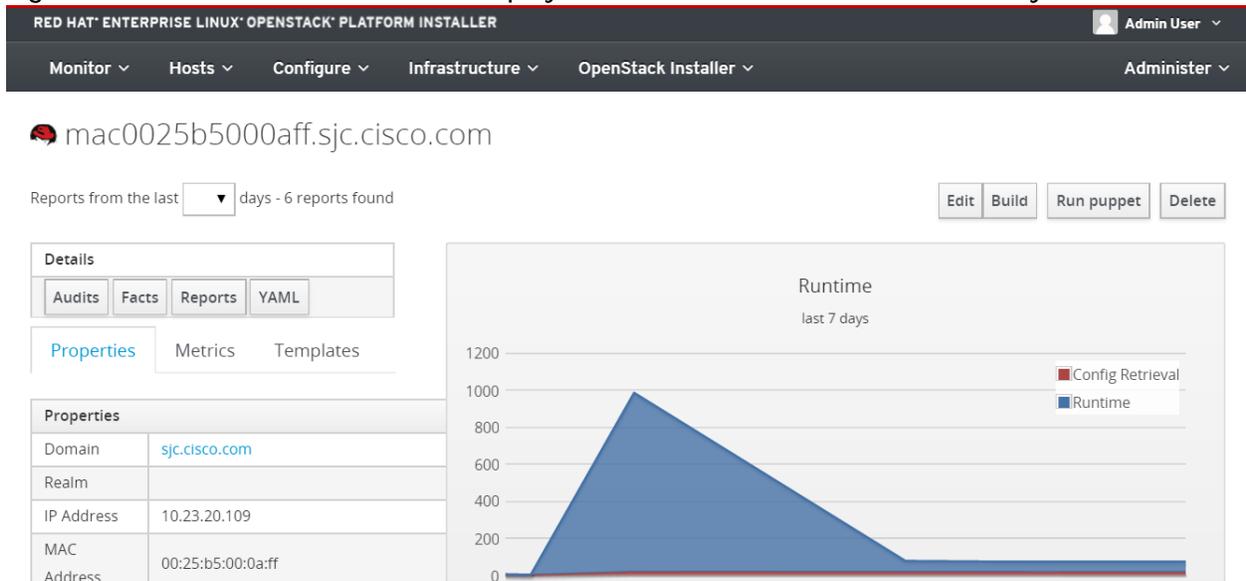
1. Configure SP_OSP_Controller_2 for primary VSM module.
 - a. Select OpenStack Installer → Deployments → FLEXPOD_RHEL_OSP6
 - b. **Click “3” in Deployment Roles next to Controller.**
 - c. Click mac0025b5000aff under Deployed Hosts (Figure 188)

Figure 188 Nexus 1000v VSM Deployment – Select Controller Host for Primary VSM



- d. Select Edit in the top right hand corner of the screen as shown in Figure 189.

Figure 189 Cisco Nexus 1000v VSM Deployment – Edit Controller Host for Primary VSM

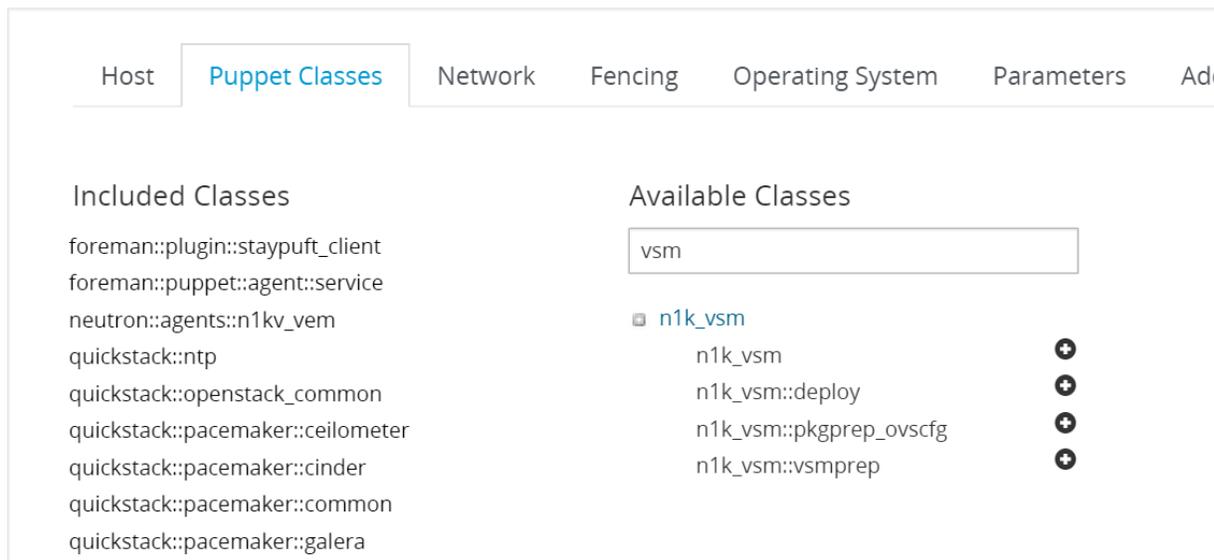


- e. Select Puppet Class tab.
- f. Type vsm in Available Classes text box.
- g. Select n1k_vsm
- h. Click  to add n1k_vsm (Figure 190).
- i. n1k_vsm will be added in Included Classes.

Figure 190 Cisco Nexus 1000v VSM Deployment – Add n1k_vsm Puppet Class for Primary VSM



Edit mac0025b5000aff.sjc.cisco.com



- j. Select Parameter tab.
- k. Override the following parameters as shown in Figure 191. Use the Table 24 worksheet applicable to your environment for Cisco Nexus 1000v primary VSM.

Table 24 Cisco Nexus 1000v Primary VSM Parameters Worksheet

Puppet Class	Parameter Name	Description	Example
n1k_vsm	n1kv_source	The location of the Cisco Nexus 1000v VSM ISO/RPM package	https://cns-g-yum-server.cisco.com/yumrepo/
n1k_vsm	n1kv_version	The version of Cisco Nexus 1000v VSM	5.2.1.SK3.2.2b-1
n1k_vsm	phy_gateway	Default gateway for the management network	10.23.10.2
n1k_vsm	phy_if_bridge	Physical interface that will be moved to the bridge for management traffic.	enp15s0.10
n1k_vsm	vsm_admin_passwd	Password of the administrative user for the Cisco Nexus 1000v VSM.	-
n1k_vsm	vsm_domain_id	Domain ID of the Cisco Nexus 1000v VSM	1000

Puppet Class	Parameter Name	Description	Example
n1k_vsm	vsm_mgmt_gateway	IP address of the default gateway for the management interface of the Cisco Nexus 1000v VSM	10.23.10.2
n1k_vsm	vsm_mgmt_ip	IP address of the management interface of the Cisco Nexus 1000v VSM	10.23.10.34
n1k_vsm	vsm_mgmt_netmask	IP netmask of the management interface of the Cisco Nexus 1000v VSM	255.255.255.0
n1k_vsm	vsm_role	VSM role (standalone, primary, secondary) of the Cisco Nexus 1000v	primary
neu-tron::agents::n1kv_vem	host_mgmt_intf	Management interface of the node where the VSM is installed.	vsm-br

Figure 191 Cisco Nexus 1000v VSM Deployment – Override Parameters for N1kv Primary VSM Configuration

Monitor	Hosts	Configure	Infrastructure	OpenStack Installer
	foreman::puppet::agent::service		runmode	service
	neutron::agents::n1kv_vem		host_mgmt_intf	vsm-br
	n1k_vsm		n1kv_source	https://cnsng-yum-server.cisco.com/yumrepo/
	n1k_vsm		n1kv_version	5.2.1.SK3.2.2-1
	n1k_vsm		phy_gateway	10.23.10.2
	n1k_vsm		phy_if_bridge	enp15s0.10
	n1k_vsm		vsm_admin_password	enter VSM password here
	n1k_vsm		vsm_mgmt_gateway	10.23.10.2
	n1k_vsm		vsm_role	primary
	n1k_vsm		vsm_mgmt_netmask	255.255.255.0
	n1k_vsm		vsm_domain_id	1000
	n1k_vsm		vsm_mgmt_ip	10.23.10.34

- l. Click Submit  to save changes.
- m. Click Run Puppet (Figure 192)



When the parameters have been overridden and the configuration have been submitted, puppet can also be initiated by running the following command: `# script -c "puppet agent --test --verbose" puppet-01.txt`

Figure 192 Cisco Nexus 1000v VSM Deployment – Run Puppet for Primary VSM

Monitor Hosts Configure Infrastructure OpenStack Installer Administer

 mac0025b5000aff.sjc.cisco.com

Reports from the last 1 days - 113 reports found

Edit Build Run puppet Delete

Details

Audits

Facts

Reports

YAML

Properties Metrics Templates

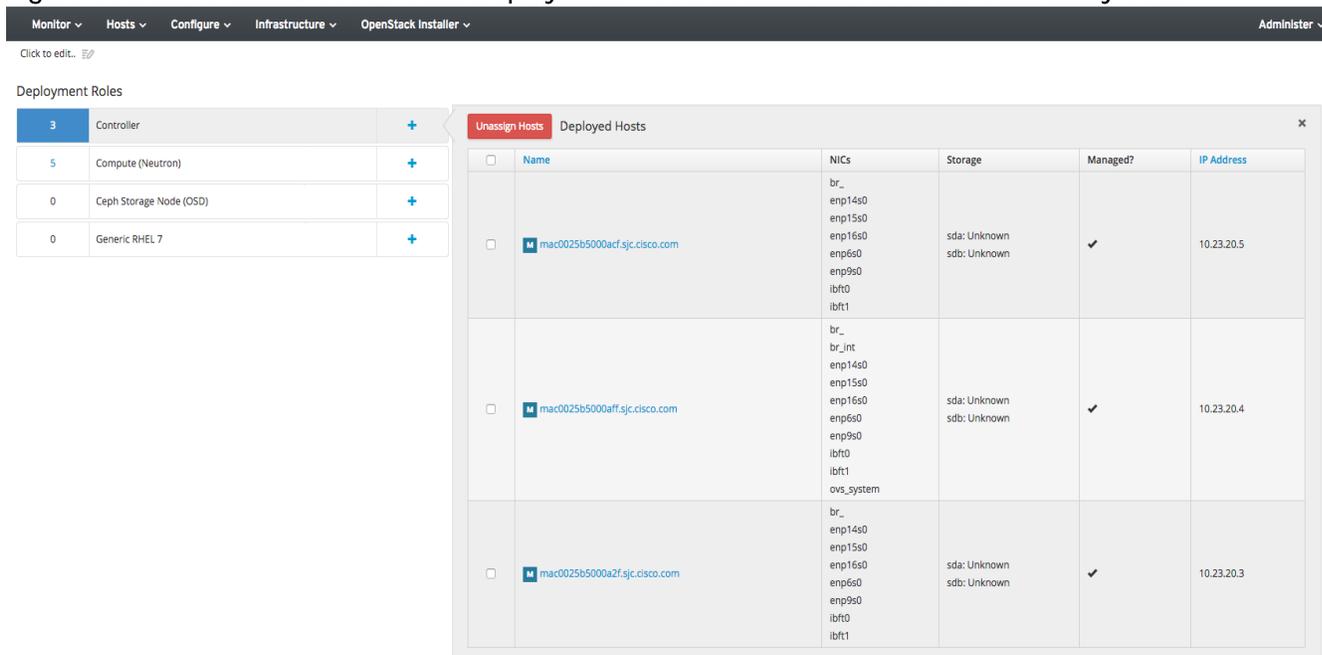
Runtime

last 7 days

1200
■ Config Retrieval

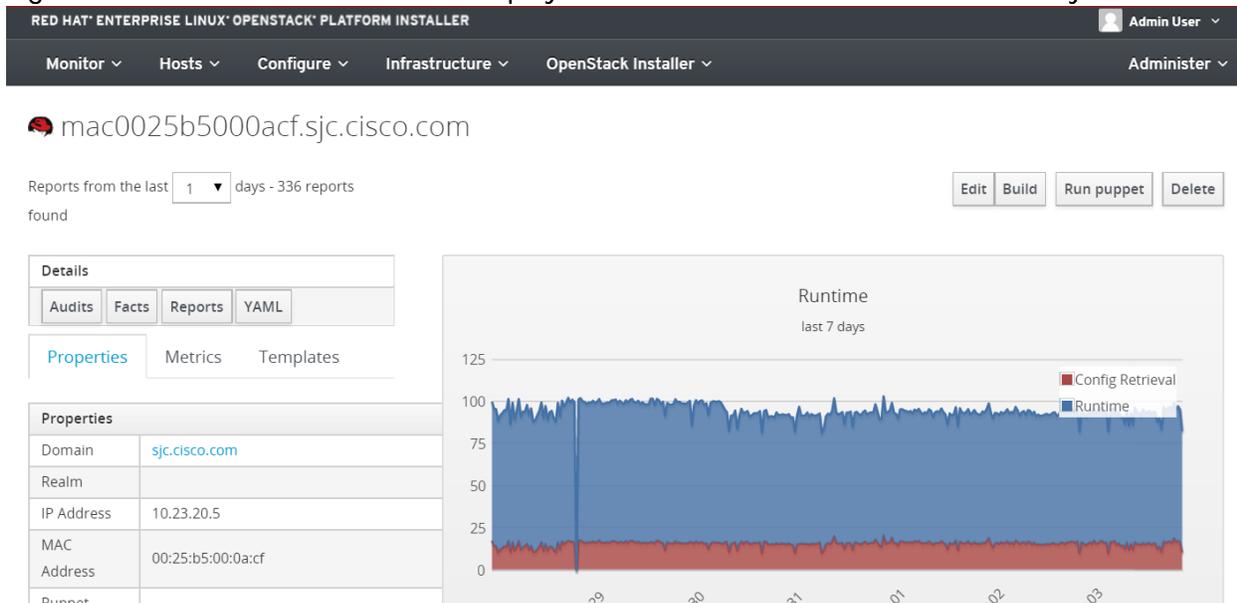
2. Configure SP_OSP_Controller_3 for secondary VSM module.
 - a. Select OpenStack Installer → Deployments → FLEXPOD_RHEL_OSP6
 - b. Click “3” in Deployment Roles next to Controller.
 - c. Click mac0025b5000acf under Deployed Hosts (Figure 193)

Figure 193 Cisco Nexus 1000v VSM Deployment – Select Controller Host for Secondary VSM



- d. Select Edit in the top right hand corner of the screen as shown in Figure 194.

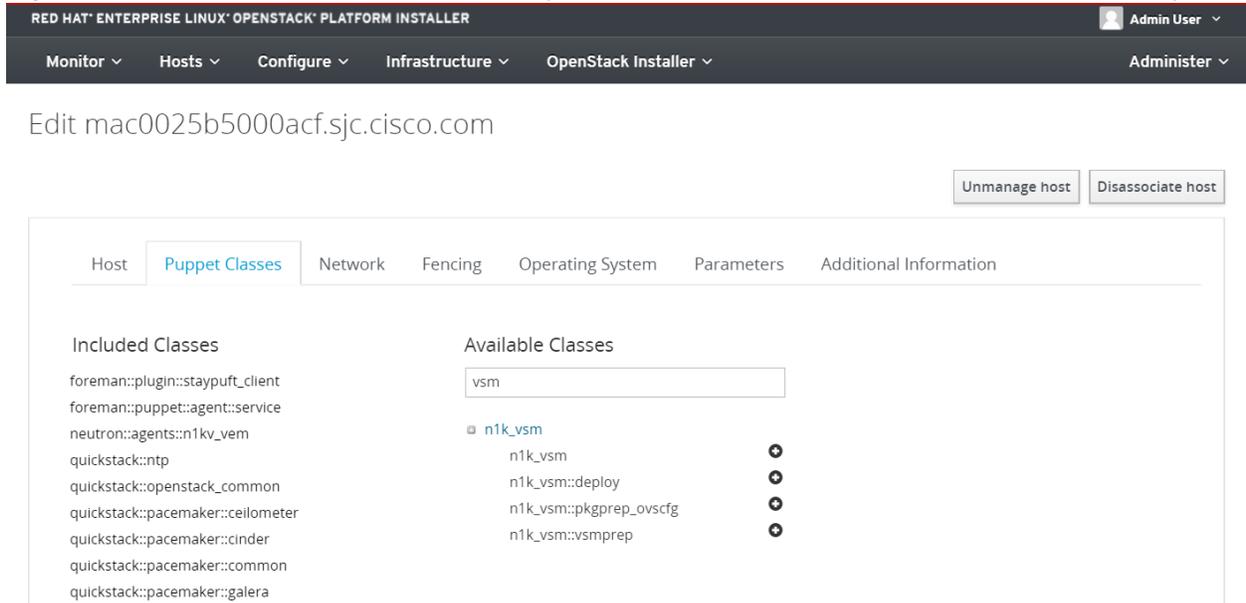
Figure 194 Cisco Nexus 1000v VSM Deployment – Edit Controller Host for Secondary VSM



- e. Select Puppet Class tab.

- f. Type vsm in Available Classes text box.
- g. Select n1k_vsm
- h. Click  to add n1k_vsm (Figure 195).
- i. n1k_vsm will be added in Included Classes.

Figure 195 Cisco Nexus 1000v VSM Deployment – Add n1k_vsm Puppet Class for Secondary VSM



- j. Select Parameter tab.
- k. Override the following parameters as shown in Figure (196). Use Table 25 worksheet applicable to your environment for Cisco Nexus 1000v secondary VSM.

Table 25 Cisco Nexus 1000v Secondary VSM Parameters Worksheet

Puppet Class	Parameter Name	Description	Example
n1k_vsm	n1kv_source	The location of the Cisco Nexus 1000v VSM ISO/RPM package	https://cns-g-yum-server.cisco.com/yumrepo/
n1k_vsm	n1kv_version	The version of Cisco Nexus 1000v VSM	5.2.1.SK3.2.2b-1
n1k_vsm	phy_gateway	Default gateway for the management network	10.23.10.2
n1k_vsm	phy_if_bridge	Physical interface that will be moved to the bridge for management traffic.	enp15s0.10
n1k_vsm	vsm_admin_passwd	Password of the administrative user for the Cisco Nexus 1000v VSM.	-
n1k_vsm	vsm_domain_id	Domain ID of the Cisco Nexus 1000v VSM	1000

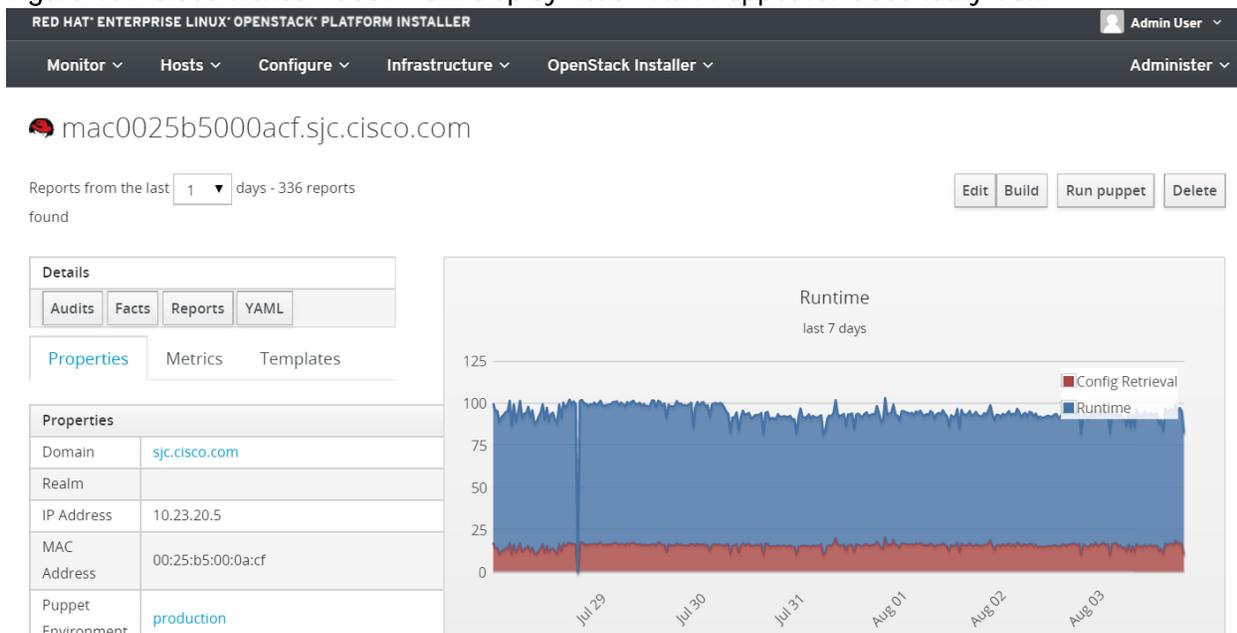
Puppet Class	Parameter Name	Description	Example
n1k_vsm	vsm_mgmt_gateway	IP address of the default gateway for the management interface of the Cisco Nexus 1000v VSM	10.23.10.2
n1k_vsm	vsm_mgmt_ip	IP address of the management interface of the Cisco Nexus 1000v VSM	10.23.10.34
n1k_vsm	vsm_mgmt_netmask	IP netmask of the management interface of the Cisco Nexus 1000v VSM	255.255.255.0
n1k_vsm	vsm_role	VSM role (standalone, primary, secondary) of the Cisco Nexus 1000v	secondary
neutron::agents::n1kv_vem	host_mgmt_intf	Management interface of the node where the VSM is installed.	vsm-br

Figure 196 Cisco Nexus 1000v VSM Deployment – Override Parameters for N1kv Secondary VSM Configuration

Monitor ▾	Hosts ▾	Configure ▾	Infrastructure ▾	OpenStack Installer ▾
Host Parameters				
foreman::puppet::agent::service	runmode	service		
n1k_vsm	n1kv_source	https://cns-g-yum-server.cisco.com/yumrepo/		
n1k_vsm	n1kv_version	5.2.1.SK3.2.2-1		
n1k_vsm	phy_gateway	10.23.10.2		
n1k_vsm	phy_if_bridge	enp15s0.10		
n1k_vsm	vsm_admin_passw	enter VSM password here		
n1k_vsm	vsm_domain_id	1000		
n1k_vsm	vsm_mgmt_gatewa	10.23.10.2		
n1k_vsm	vsm_mgmt_ip	10.23.10.34		
n1k_vsm	vsm_mgmt_netmas	255.255.255.0		
n1k_vsm	vsm_role	secondary		
neutron::agents::n1kv_vem	host_mgmt_intf	vsm-br		

- l. Click **Submit** to save changes.
- m. Click Run Puppet (Figure 197)

Figure 197 Cisco Nexus 1000v VSM Deployment – Run Puppet for Secondary VSM

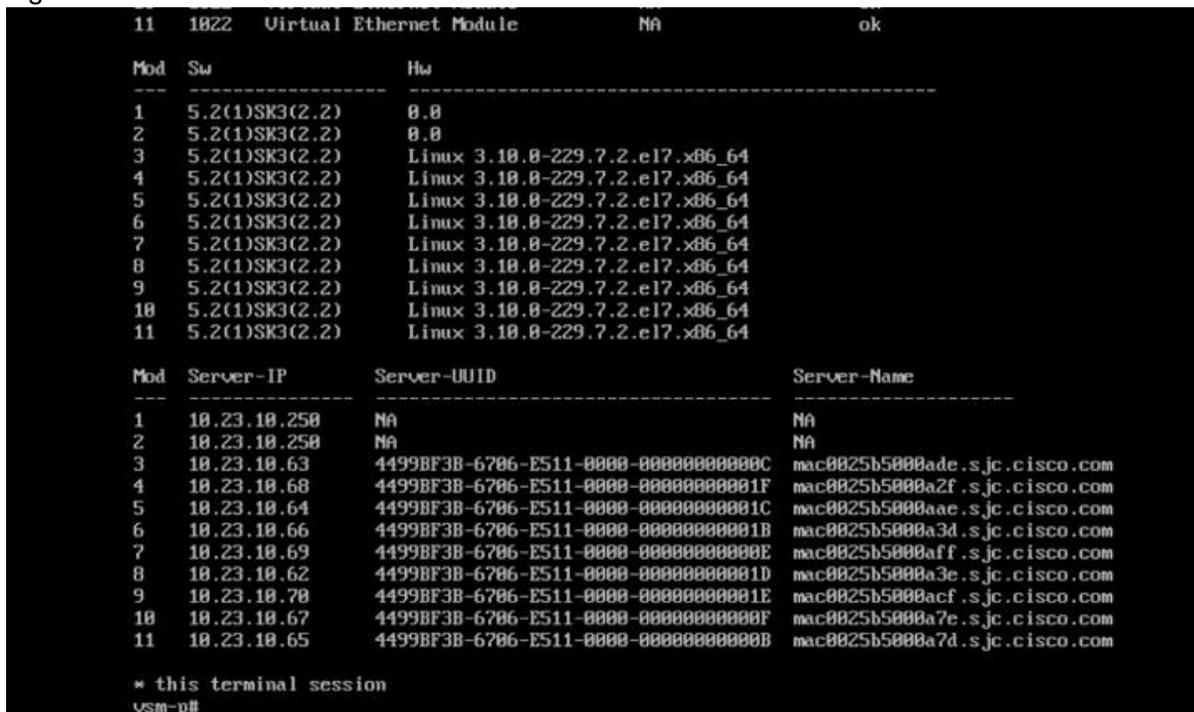


Cisco Nexus 1000v Configuration

To log into Cisco Nexus 1000v primary VSM and verify the VSM and VEM status, complete the following steps:

1. Run show module. It should show VSM active and hot standby and all the VEM modules are discovered by VSM (Figure 198)

Figure 198 Nexus 1000v – Show Module



2. While in the VSM, create the default port-profile

```
configure terminal
port-profile type vethernet default-pp
no shutdown
state enabled
publish port-profile
```

3. Create the tenant VLANs

- a. Create provider and tenant VLANs

```
vlan 60-200
```

- b. Create external VLAN

```
vlan 215
```

4. Create an uplink port profile. Use the name specified in host group parameters for both controller and compute in neutron::agents::n1kv_vem → uplink_profile

```
port-profile type Ethernet uplink
switchport mode trunk
switchport trunk allowed vlan 60-100, 215
no shut
state enabled
publish port-profile
```

5. Save the configuration by running the following command.

```
copy run start
```

Jumbo Frames for NFS Traffic and Bond0

The bond0 logical network interface on all of the servers in the provisioned environment is setup by default to have an MTU size of 1500 by the RHEL-OSP Installer. This parameter must be increased to 9000 to match the upstream Nexus configuration and provide better performance for Cinder through NFS.

Complete the following steps on all Cisco UCS server nodes (controller and compute) provisioned in the environment:

1. Log into the system as the root user, either through the console or through SSH.

- To enable the bond0 and bond0.30 interfaces for jumbo frames, run the following commands:

```
echo "MTU=9000" >> /etc/sysconfig/network-scripts/ifcfg-bond0
echo "MTU=9000" >> /etc/sysconfig/network-scripts/ifcfg-bond0.30
```

- Restart network services on the node. Ignore any warnings or errors that result from this command.

```
systemctl restart network.service
```

- Make sure the changes have taken affect. The following commands can be used to send an ICMP request to the NetApp FAS system to make sure that jumbo frames are working for the entire network path. Substitute the variable below with the NFS LIF as appropriate:

```
ping -c 5 -s 8972 -M do <<nfs_lif_ip1>>
ping -c 5 -s 8972 -M do <<nfs_lif_ip2>>
```

- A successful test of jumbo frames on the NFS LIF running on the FAS8040 displays the following results:

```
PING 10.23.10.32 (10.23.30.14) 8972(9000) bytes of data.
8980 bytes from 10.23.30.14: icmp_seq=1 ttl=255 time=0.716 ms
8980 bytes from 10.23.30.14: icmp_seq=2 ttl=255 time=0.222 ms
8980 bytes from 10.23.30.14: icmp_seq=3 ttl=255 time=0.516 ms
```

Storage Service Catalog

NetApp Cinder drivers allow you to construct a catalog of differing storage capabilities to meet a diverse base of application and tenant needs. The Cinder Storage Service Catalog can be defined with various efficiency, performance, availability, and protection attributes. The catalog entries themselves (referred to as Cinder volume types) can be defined in a very granular form or more commonly represent a collection of capabilities most appropriate for a particular tenant by using the extra-specs functionality in Cinder. An example of a validated Storage Service Catalog in this reference architecture is shown in Table 26

Table 26 Storage Service Catalog

Volume Type	Extra Specs
archived_data	{netapp_thick_provisioned=false netapp_dedup=true netapp_compression=true}
general_purpose	{netapp:disk_type=SAS netapp_thin_prvisioned=false}
transactional_db	{netapp_thin_provisioned=false netapp_mirrored=true netapp:disk_type=SSD}



This information is provided as a guideline, and is appropriate for the hardware used in this validation. For a full listing of applicable values available in the Juno release of OpenStack for tailoring for the customer environment, refer to [NetApp supported Extra Specs for use with Cinder Volume Types](#).

To build a Storage Service Catalog, complete the following steps:

1. Source the required environment parameters to obtain administrator access in Keystone on one of the controller nodes:

```
source /root/keystonerc_admin
```

2. To create the Cinder volume types, run the following commands:

```
cinder type-create archived_data
cinder type-create general_purpose
cinder type-create transactional_db
```

3. Associate the desired extra-spec parameters with the Cinder volume types:

```
cinder type-key archived_data set netapp_thick_provisioned=false
cinder type-key archived_data set netapp_dedup=true
cinder type-key archived_data set netapp_compression=true

cinder type-key general_purpose set netapp:disk_type=SAS
cinder type-key general_purpose set netapp_thin_provisioned=false

cinder type-key transactional_db set netapp_thin_provisioned=false
cinder type-key transactional_db set netapp_mirrored=true
cinder type-key transactional_db set netapp:disk_type=SSD
```

Quality of Service

Quality-of-Service(QoS) specifications are used to apply generic QoS support for volumes that can be enforced at the hypervisor (front-end), enforced specifically by Data ONTAP, or enforced at both levels. QoS specifications are added as standalone objects that can then be associated with Cinder volume types.

Front-end QoS

Front-end QoS is enforced in [libvirt](#), and can be implemented by performing the following commands:

1. Source the environment parameters if needed.

```
source /root/keystonerc_admin
```

2. Create two new QoS specifications with various IOPS limitations.

```
cinder qos-create archived_data_qos_specs consumer="front-end" total_iops_sec=500
cinder qos-create transactional_db_qos_specs consumer="front-end" read_iops_sec=2000 write_iops_sec=1000
```

3. To grab the ID string of the volume type that QoS specifications must be applied to, run the following script:

```
qosSpecs=$(cinder qos-list | awk '/archived_data_qos/ {print $2}')
volumeType=$(cinder type-list | awk '/archived_data/ {print $2}')
```

4. Associate a QoS specification with a particular Cinder volume type.

```
cinder qos-associate $qosSpecs $volumeType
```

5. To verify that QoS has been applied, run the following command:

```
cinder qos-get-association $qosSpecs
```

```
+-----+-----+-----+
| Association_Type | Name | ID |
+-----+-----+-----+
```

```

+-----+-----+-----+
| volume_type | archived_data | 507e64b3-783b-4b00-b320-40a13e8b497d |
+-----+-----+-----+

```

6. To disassociate QoS specifications that have been previously applied, run the following command:

```
cinder qos-disassociate $qosSpecs $volumeType
```

7. Verify that QoS has been removed:

```
cinder qos-get-association $qosSpecs
```

```

+-----+-----+-----+
| Association_Type | Name | ID |
+-----+-----+-----+
+-----+-----+-----+

```

8. To delete the QoS specification, run the following command:

```
cinder qos-delete $qosSpecs
```

NetApp QoS Policy Group

NetApp storage QoS policy enforcement can be exposed to the Storage Service Catalog through an extra-specs association. Therefore, you should define a QoS policy group object within Data ONTAP before a Cinder volume is created, and you should associate the QoS policy group with the destination FlexVol volume.

To make sure that a Cinder volume-type adheres to a specific NetApp QoS policy, run the following command. Be sure the assignment value matches the QoS policy in Data ONTAP. In this example we use `ontap_general_purpose_qos`.

```
cinder type-key general_purpose set netapp:qos_policy_group=ontap_general_purpose_qos
```



For more information on NetApp QoS in Data ONTAP 8.3, refer to the section “Managing system performance (cluster administrators only)” in the [System Administration Guide for Cluster Administrators](#).

NetApp Copy Offload Tool

The NetApp Copy Offload tool efficiently copies Glance images to a destination Cinder volume. Rather than a copy through the network from Glance to Cinder, the tool makes a clone from the Glance Image to the Cinder volume through the storage system itself.

Complete the following steps on all Cisco UCS server nodes that are OpenStack controller nodes.

1. Navigate to the [Legacy ToolChest Directory](#) on the NetApp Support Site.
2. Log into the NetApp Support site by using appropriate credentials.
3. Select NetApp OpenStack NFS Copy Offload Client. Click Continue after reading the informational message.
4. Accept the EULA terms.
5. Copy the resulting files to the `/usr/local/bin` directory:

```
cp copyoffload.tar.gz /usr/local/bin/
```

6. Untar and unzip the archive with the following command:

```
tar -xvzf copyoffload.tar.gz
```

7. Open the `/etc/cinder/cinder.conf` file and modify `netapp_copyoffload_tool_path` from `None` to `/usr/local/bin/na_copyoffload_64`.
8. In the same file, modify `glance_api_version` from 1 to 2.
9. Open the `/etc/glance/glance-api.conf` file, and under the `[DEFAULT]` stanza add the following contents:

```
[DEFAULT]
show_multiple_locations=True
```

10. In the same file, modify `show_image_direct_url` from `False` to `True`.
11. In the same file, modify `filesystem_store_metadata_file` from `None` to `/etc/glance/metadata.json`.
12. Create the `/etc/glance/metadata.json` file and modify it with the following contents:

```
{
  "share_location": "nfs://<<nfs_lif_ip1>>/glance",
  "mount_point": "/var/lib/glance/images",
  "type": "nfs"
}
```

13. Set the immutable flag on `/etc/cinder/cinder.conf` and `/etc/glance/glance-api.conf` to prevent Puppet from overwriting modifications when it runs.

```
chattr +i /etc/cinder/cinder.conf
chattr +i /etc/glance/glance-api.conf
```

14. Add the `cinder` user to the `glance` group with the following command:

```
gpasswd -a cinder glance
```

15. Restart the Cinder subsystem to pick up the changes.

```
systemctl restart openstack-cinder-{api,scheduler,volume}
```

16. Restart the Glance subsystem to pick up the changes.

```
systemctl restart openstack-glance-{api,registry}
```

Swift Deployment

OpenStack Object Storage provides a fully distributed, scale-out, API-accessible storage platform that can be integrated directly into applications or used for backup, archiving, and data retention.

Swift must be installed and configured manually after an OpenStack deployment. The following instructions assume that all of the setup instructions detailed in the “NetApp E-Series E5560 Configuration” section have been implemented successfully. Table 27 lists the information needed to configure Swift in your environment. You should customize the following values with information that is applicable to your deployment.

Table 27 OpenStack Swift Software Configuration Worksheet

Swift environment Detail	Swift environment Detail Value
Swift-A VLAN	<<var_swiftA_vlan_id>>
Swift-B VLAN	<<var_swiftB_vlan_id>>
NFS VLAN	<<var_nfs_vlan_id>>
OpenStack Controller1 Swift-A iSCSI IP	<<var_cont1_initiator_swiftA_ip>>
OpenStack Controller1 Swift-A iSCSI netmask	<<var_cont1_initiator_swiftA_netmask>>
OpenStack Controller2 Swift-A iSCSI IP	<<var_cont2_initiator_swiftA_ip>>
OpenStack Controller2 Swift-A iSCSI netmask	<<var_cont2_initiator_swiftA_netmask>>
OpenStack Controller3 Swift-A iSCSI IP	<<var_cont3_initiator_swiftA_ip>>
OpenStack Controller3 Swift-A iSCSI netmask	<<var_cont3_initiator_swiftA_netmask>>
OpenStack Controller1 Swift-B iSCSI IP	<<var_cont1_initiator_swiftB_ip>>
OpenStack Controller1 Swift-B iSCSI netmask	<<var_cont1_initiator_swiftB_netmask>>
OpenStack Controller2 Swift-B iSCSI IP	<<var_cont2_initiator_swiftB_ip>>
OpenStack Controller2 Swift-B iSCSI netmask	<<var_cont2_initiator_swiftB_netmask>>
OpenStack Controller3 Swift-B iSCSI IP	<<var_cont3_initiator_swiftB_ip>>
OpenStack Controller3 Swift-B iSCSI netmask	<<var_cont3_initiator_swiftB_netmask>>

Create Network Interfaces for Swift Traffic

The 802.1Q VLAN tagged interfaces used for Swift traffic must be created.

Complete the following steps on all OpenStack controller nodes provisioned in the environment:

1. Log into the system as the root user, either through the console or through SSH.
2. Copy the `bond0.<<var_nfs_vlan_id>>` interface configuration to a new `bond0.<<var_swiftA_vlan_id>>` interface.



In our environment, this was VLAN 50.

```
cp ifcfg-bond0.30 ifcfg-bond0.50
```

3. Modify the `ifcfg-bond0.<<var_swiftA_vlan_id>>` interface configuration file in a manner similar to the following example:

```
IPADDR="<<var_cont1_initiator_swiftA_ip>>"
```

```
NETMASK="<

```

- Copy the bond0.<<swiftA_vlan_id>> to a new bond0.<<var_swiftB_vlan_id>> interface.



In our environment, this was VLAN 51.

```
cp ifcfg-bond0.50 ifcfg-bond0.51
```

- Modify the the ifcfg-bond0.<<var_swiftB_vlan_id>> interface configuration file in a manner similar to the following example:

```
IPADDR="<

```

- Restart network services on the node. Ignore any warnings or errors that result from this command.

```
systemctl restart network.service
```

- Make sure that your changes have taken affect. Use the following commands to send an ICMP request to the NetApp E-Series system to ensure that jumbo frames are working for the entire network path. Substitute the following variables with the appropriate Controller HIC IP addresses as needed:

```
ping -c 5 -s 8972 -M do <<var_iscsi_conta_hic1_p1_ip>>
ping -c 5 -s 8972 -M do <<var_iscsi_conta_hic1_p3_ip>>
ping -c 5 -s 8972 -M do <<var_iscsi_contb_hic1_p1_ip>>
ping -c 5 -s 8972 -M do <<var_iscsi_contb_hic1_p3_ip>>
```

- Repeat steps 1-7 on the other two controllers, substituting for the proper controller.

Log into the E-Series Array by Using iSCSI

With the newly configured interfaces, you can now log into the E-Series array through the `iscsiadm` command.

On all OpenStack controller nodes provisioned in the environment, log into all of the paths advertised by the NetApp E-series array. After you perform the following commands, the iSCSI connections are persistent across system reboots.

```
iscsiadm --mode discovery --type sendtargets --portal <<var_iscsi_conta_hic1_p1_ip>>
iscsiadm --mode node --login
```

Swift Package Installation

Swift-specific packages must be installed because they are not installed by default as a part of an OpenStack deployment.

Run the following commands on all OpenStack controller nodes provisioned in the environment:

```
yum install -y openstack-swift-proxy \
  openstack-swift-object \
```

```
openstack-swift-container \
openstack-swift-account
```

Keystone User and Role Configuration

The OpenStack Identity Service (Keystone) must have a user and role added to accommodate Swift. Source the needed environment parameters on one OpenStack controller node:

```
source /root/keystonerc_admin
```

1. Create the `swift` user, give it a strong password, assign it to the `services` tenant, and give it an e-mail address.

```
keystone user-create --name swift --pass <<var_swift_password>> --tenant services --email swift@localhost
```

2. Give the `swift` user the administrator role.

```
keystone user-role-add --user swift --tenant services --role admin
```

3. Create the Swift Object Storage service entry.

```
keystone service-create --name swift --type object-store --description "Openstack Object-Store Service"
```

4. If Swift support is needed in a tenant or project, run the following commands, substituting the user and tenant field for values appropriate in your environment:

```
keystone role-create --name SwiftOperator
keystone user-role-add --user <<var_keystone_username>> --tenant <<var_keystone_tenant>> --role SwiftOperator
```

IPTables Firewall Exceptions

As a result of the Swift implementation, several ports must be opened on the iptables firewall.

Complete the following steps on all OpenStack controller nodes provisioned in the environment:

1. Run the following commands.

```
iptables -I INPUT 1 -p tcp -m multiport --ports 6200,6201,6202 -m comment --comment "Swift Services" -j ACCEPT
iptables -I INPUT 1 -p tcp -m multiport --ports 8080 -m comment --comment "Swift Proxy" -j ACCEPT
iptables-save > /etc/sysconfig/iptables
```

2. Restart the IPTables service to pick up the changes.

```
systemctl restart iptables.service
```

Partition, File System, and Directory Structure Creation

To create the partitions of the raw block devices presented to the OpenStack controller systems, an XFS file system, and the necessary directory structure, complete the following.

Complete the following actions on all OpenStack controller nodes provisioned in the environment:

1. Partition each of the three LUNs presented to the controller system with a single, large partition.

```
for SEQ in {a..c} ; do parted -a optimal -s -- /dev/mapper/mpath`echo $SEQ` mklabel gpt mkpart primary xfs 0% 100%; done
```

2. Reprobe the partition table, and then format the partitions with the XFS file system.

```
/usr/sbin/partprobe;
for SEQ in {a..c} ; do mkfs.xfs -f -d su=131072,sw=8 /dev/mapper/mpath`echo $SEQ`; done
```

3. Create the `/srv/node` directory structure that hold account data, containers, and objects. Mounting the LUNs to these directories is performed in a later step.

```
cd /srv/; mkdir node
for SEQ in {a..c} ; do mkdir /srv/node/mpath`echo $SEQ`; done
```

Udev Consistent Naming

Now create a persistent Udev rules for the `/dev/mapper/mpath` devices, as system reboots result in different LUNs being chosen for `/dev/mapper/mpath[a-c]`, because this selection is random.

Implement the following formats for each controller host:

- Lun0 should be the Account volume. In our environment this is 500GB.
- Lun1 should be the Container volume. In our environment, this is 500GB.
- Lun2 should be the Object volume. In our environment, this is 25TB.

Complete the following actions on all OpenStack controller nodes provisioned in the environment:

1. Run the command `multipath -ll` and look at the output carefully. Pay particular attention to the fields highlighted in the following table as an example, because they indicate the LUN number exposed to the controller system through mapping originating from the E-Series array. We use this information to build a persistent Udev rules file and use the highlighted `mpatha` as an example in our environment.

```
mpathc (360080e5000296d2000000b7d55a66528) dm-8 NETAPP ,INF-01-00
size=500G features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 rdac' wp=rw
|+-+ policy='service-time 0' prio=14 status=active
| |- 6:0:0:1 sdg 8:96 active ready running
| |- 5:0:0:1 sdd 8:48 active ready running
`+-+ policy='service-time 0' prio=9 status=enabled
  |- 7:0:0:1 sdj 8:144 active ready running
  `-- 8:0:0:1 sdm 8:192 active ready running
mpathb (360080e500029745800000cf555a66631) dm-9 NETAPP ,INF-01-00
size=500G features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 rdac' wp=rw
|+-+ policy='service-time 0' prio=14 status=active
| |- 8:0:0:0 sdl 8:176 active ready running
| |- 7:0:0:0 sdi 8:128 active ready running
`+-+ policy='service-time 0' prio=9 status=enabled
  |- 5:0:0:0 sdc 8:32 active ready running
  `-- 6:0:0:0 sde 8:64 active ready running
mpatha (360080e5000296d2000000b37558aab5) dm-7 NETAPP ,INF-01-00
size=25T features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 rdac' wp=rw
|+-+ policy='service-time 0' prio=14 status=active
| |- 6:0:0:2 sdh 8:112 active ready running
| |- 5:0:0:2 sdf 8:80 active ready running
`+-+ policy='service-time 0' prio=9 status=enabled
  |- 8:0:0:2 sdn 8:208 active ready running
  `-- 7:0:0:2 sdk 8:160 active ready running
```

- Print the block device attributes of the individual partitions on the DM-multipath devices and record this information. The returned UUID value is what we use to create symbolic link Udev devices because this value never changes. The following highlighted entry represents a UUID in our environment.

```
blkid /dev/mapper/mpatha1

/dev/mapper/mpatha1: UUID="ef9a09c3-1b66-4b5b-b36c-38bedb0a76bb" TYPE="xfs" PARTLABEL="primary"
PARTUUID="6f08c114-6ded-4819-b61c-cad2522abf01"
```

- Repeat step 2 two more times, substituting `mpatha1` with `mpathb1` and `mpathc1`. Note the unique UUID values for these partitions.
- Create the `/etc/udev/rules.d/99-netapp-eseries.rules` file to hold the UUID to the symbolic link mounting structure specific to your environment. After this change, the partitions are available under the `/dev/eseries/diskX` directory, where `X` equals the LUN format discussed at the top of this section.
- An example file applicable in our environment is as follows. Note the `ef9a...` UUID matches with `eseries/disk2`, which equals `LUN2`, our 25TB object volume.

```
ACTION=="add|change" ENV{ID_FS_UUID}=="478f433b-1977-4e34-b8f9-e0dff7389b48", SYMLINK+="eseries/disk0",
OWNER="root", GROUP="disk", MODE="0660"
ACTION=="add|change" ENV{ID_FS_UUID}=="c85b326f-9371-4cdc-b67e-3e7860d023ed", SYMLINK+="eseries/disk1",
OWNER="root", GROUP="disk", MODE="0660"
ACTION=="add|change" ENV{ID_FS_UUID}=="ef9a09c3-1b66-4b5b-b36c-38bedb0a76bb", SYMLINK+="eseries/disk2",
OWNER="root", GROUP="disk", MODE="0660"
```

- To reload the Udev rules without rebooting the system, perform the following steps:

```
udevadm control --reload-rules
udevadm trigger --type=devices --action=change
```

- To verify the new `/dev/eseries` directory, run the following command:

```
[root@mac0025b5000a2f ~]# ls -lah /dev/eseries/
total 0
drwxr-xr-x.  2 root root  100 Jul 22 14:20 .
drwxr-xr-x. 22 root root  3.6K Jul 22 14:20 ..
lrwxrwxrwx.  1 root root    8 Jul 22 14:20 disk0 -> ../dm-12
lrwxrwxrwx.  1 root root    8 Jul 22 14:20 disk1 -> ../dm-11
lrwxrwxrwx.  1 root root    8 Jul 22 14:20 disk2 -> ../dm-10
```

- Repeat steps 1-6 on the other two OpenStack Controller systems.

Resource Mounts and Permissions

Since this is a high-availability environment using Pacemaker, you cannot simply add entries to `/etc/fstab` on each system. Rather, you must create resources for the individual volumes mounted on each controller as a unit. Because we have a consistent Udev naming structure that is the same across all controllers, we can **use Pacemaker's resource creation functionality** to mount the raw devices presented under `/dev/eseries/diskX` to **Pacemaker's corresponding mount location of** `/dev/node/mpathX` on each system. `X` represents an integer value from 1-3.

Pacemaker

Add the file system resources to the Pacemaker cluster, which handle the mounting of volumes on each controller system.

Create the file system resources for the account, container, and object volumes on one OpenStack controller:

```
pcs resource create swift-account-fs Filesystem params device="/dev/eseries/disk0"
directory="/srv/node/mpatha" fstype="xfs" "options=_netdev,nobarrier,noatime,nodiratime,inode64"
force_clones="yes" --clone interleave=true

pcs resource create swift-container-fs Filesystem params device="/dev/eseries/disk1"
directory="/srv/node/mpathb" fstype="xfs" "options=_netdev,nobarrier,noatime,nodiratime,inode64"
force_clones="yes" --clone interleave=true

pcs resource create swift-object-fs Filesystem params device="/dev/eseries/disk2"
directory="/srv/node/mpathc" fstype="xfs" "options=_netdev,nobarrier,noatime,nodiratime,inode64"
force_clones="yes" --clone interleave=true
```

Filesystem Permissions and SELinux

Adjust the user and group ownership for the `/srv/node/` directory and restore some security contexts for SELinux. Perform the following actions on all OpenStack controller nodes provisioned in the environment:

1. Change the user and group ownership on the `/srv/node` directory:

```
chown -R swift:swift /srv/node/
```

2. Restore the security contexts recursively on the `/srv` directory:

```
restorecon -Rv /srv
```

3. Repeat steps 1-2 on the other two OpenStack controller systems.

Rsync Replication Between Nodes

To accommodate replication between the Swift storage nodes, setup replication through Rsync. It is a best practice to set up this replication on a private network outside of the network segments to which users normally have access. You should also enable jumbo frames on the full network path between nodes. In this validation, utilize the NFS network for communication between nodes that already satisfy these criteria.

Complete the following steps on all OpenStack controller nodes provisioned in the environment:

1. Create the `/etc/rsync-swift.conf` file. Substitute IP address information suitable to your environment.

```
log file = /var/log/rsyncd-swift.log
pid file = /var/run/swift/rsyncd-swift.pid
uid = swift
gid = swift
use chroot = no
log format = %t %a %m %f %b
timeout = 300
address = 10.23.30.51

[ account ]
max connections = 10
path = /srv/node/
read only = false
write only      = no
list            = yes
```

```

incoming chmod = 0644
outgoing chmod = 0644
lock file = /var/run/rsyncd-account.lock

[ container ]
max connections = 10
path = /srv/node/
read only = false
write only      = no
list           = yes
incoming chmod = 0644
outgoing chmod = 0644
lock file = /var/run/rsyncd-container.lock

[ object ]
max connections = 10
path = /srv/node/
read only = false
write only      = no
list           = yes
incoming chmod = 0644
outgoing chmod = 0644
lock file = /var/run/rsyncd-object.lock

```

2. Create the `/etc/xinetd.d/rsync-swift` configuration file. Substitute IP address information suitable to your environment.

```

service rsync
{
    port            = 873
    disable         = no
    socket_type     = stream
    protocol        = tcp
    wait           = no
    user           = root
    group          = root
    groups         = yes
    server         = /usr/bin/rsync
    bind           = 10.23.30.51
    server_args    = --daemon --config /etc/rsync-swift.conf
}

```

3. Since Rsync is already running on all of the controller nodes (to facilitate synchronization between both the Galera database and Keystone itself), you must restart Rsync to pick up the changes.

```
systemctl restart xinetd.service
```

4. To verify that Rsync is running successfully, run the following commands and look for xinetd listening on port 873 on the interface you have chosen to run replication:

```

[root@Controller1 /]# lsof -i :873 -n
COMMAND  PID USER  FD  TYPE    DEVICE  SIZE/OFF  NODE NAME
xinetd   25334 root   6u  IPv4  464460723    0t0  TCP 10.23.21.23:rsync (LISTEN)
xinetd   25334 root   8u  IPv4  464460724    0t0  TCP 10.23.21.16:rsync (LISTEN)
xinetd   25334 root   9u  IPv4  464460725    0t0  TCP 10.23.30.51:rsync (LISTEN)

```

5. Repeat steps 1-4 on the other two OpenStack controller systems, substituting IP addressing information that is pertinent to the system being configured.

Configure Swift

OpenStack Object Storage provides a fully distributed, scale-out, API-accessible storage platform that can be integrated directly into applications or used for backup, archiving, and data retention. To configure Swift itself, complete the following steps.

Hash Generation

These details are required for finding and placing data on all of your nodes. You must generate both a hash prefix and a hash suffix.

Perform the following actions on one OpenStack controller:

1. Add a hash prefix to `/etc/swift/swift.conf`:

```
openstack-config --set /etc/swift/swift.conf swift-hash swift_hash_path_prefix \  
$(openssl rand -hex 10)
```

2. Add a hash suffix to `/etc/swift/swift.conf`.

```
openstack-config --set /etc/swift/swift.conf swift-hash swift_hash_path_suffix \  
$(openssl rand -hex 10)
```

3. Copy `/etc/swift/swift.conf` from the current controller to the remaining two while making sure that the hash suffix and prefix are the same on both.

```
cd /etc/swift; scp swift.conf Controller2:/etc/swift/  
cd /etc/swift; scp swift.conf Controller3:/etc/swift/
```

ACO Services

To configure account, container, and object (ACO) services for Swift, complete the following steps. It is a best practice to setup storage services on a private network outside of network segments that users normally have access to as well as having jumbo frames enabled on the full network path between nodes. None of these services are user interfacing, so segment them appropriately. In this validation, the Management, Cluster Management, Admin API, and Storage Cluster (also known as the MCAS network in the RHEL-OSP installer) is used for these services.

Complete the following actions on all OpenStack controller nodes provisioned in the environment:

1. Set the IP addresses that the ACO services listen on:

```
openstack-config --set /etc/swift/account-server.conf DEFAULT bind_ip 10.23.21.23  
openstack-config --set /etc/swift/container-server.conf DEFAULT bind_ip 10.23.21.23  
openstack-config --set /etc/swift/object-server.conf DEFAULT bind_ip 10.23.21.23
```

2. Start the ACO services.

```
systemctl start openstack-swift-{account,container,object}
```

3. Enable persistent boot for the ACO services.

```
systemctl enable openstack-swift-{account,container,object}
```

4. Perform verification to ensure that the services are listening and available by using the following commands:

```
lsof -i :6200  
lsof -i :6201  
lsof -i :6202
```

5. Repeat steps 1-4 on the other two OpenStack controller systems, substituting IP addressing information that is pertinent to the system being configured.

Proxy Service

The Proxy Server service is the user- and client-facing portion of the Swift architecture. For each request, this service looks up the location of the account, container, or object and route the request accordingly. To configure the proxy server portion of Swift, complete the following steps on all OpenStack controller nodes provisioned in the environment:

1. Increase the number of netfilter connection tracker entries to satisfy a higher volume of requests.

```
echo "net.nf_conntrack_max = 262144" >> /etc/sysctl.conf; sysctl -p
```

2. Get the public facing VIP for Keystone and store it for the next step in the `keystoneVip` variable.

```
keystoneVip=$(pcs status | awk -F "-" '/ip-keystone-pub/ {print $4}' | cut -f1)
```

3. Configure the proxy server.

```
openstack-config --set /etc/swift/proxy-server.conf filter:authtoken auth_host $keystoneVip
openstack-config --set /etc/swift/proxy-server.conf filter:authtoken admin_tenant_name services
openstack-config --set /etc/swift/proxy-server.conf filter:authtoken admin_user swift
openstack-config --set /etc/swift/proxy-server.conf filter:authtoken admin_password <<var_swift_password>>
```

4. Setup proper logging for `/var/log/swift/swift.log`.

```
openstack-config --set /etc/swift/proxy-server.conf DEFAULT log_facility LOG_LOCAL
openstack-config --set /etc/swift/proxy-server.conf DEFAULT log_level ERROR
openstack-config --set /etc/swift/proxy-server.conf DEFAULT log_headers false
openstack-config --set /etc/swift/proxy-server.conf DEFAULT log_address /dev/log
```

5. Configure the memcached service. For the IP addresses listed below, be sure to use the IP addresses on the controller nodes that are on the MCAS network.

```
openstack-config --set /etc/swift/object-expirer.conf object-expirer concurrency 100
openstack-config --set /etc/swift/proxy-server.conf filter:cache memcache_servers
10.23.21.22:11211,10.23.21.23:11211,10.23.21.24:11211
openstack-config --set /etc/swift/object-expirer.conf filter:cache memcache_servers
10.23.21.22:11211,10.23.21.23:11211,10.23.21.24:11211
```

6. Restart memcached services.

```
systemctl restart memcached.service
```

7. If syslog information is desired, place the following script in `/etc/swift/proxy-server.conf` underneath the DEFAULT tag:

```
[DEFAULT]
log_name = swift
log_facility = LOG_LOCAL0
log_level = INFO
log_headers = false
log_address = /dev/log
```

8. Repeat steps 1-7 on the other two OpenStack Controller systems, substituting IP addressing information that is pertinent to the system being configured.

Build Swift Rings and Start Proxy Service

The ring data structure determines where data resides in the cluster. There is a separate ring for account databases, container databases, and individual objects. The following script effectively standardizes a single

region with three distinct zones that represent our controller systems. Our partition power is 10, and we are reducing the number of replicas to two.



Two replicas are necessary to prevent an OpenStack Controller system reboot or failure from preventing access to object data.

Complete the following step on one OpenStack controller:

1. To build the rings, customize the following script for your specific environment and copy the script to **one of** the OpenStack controller nodes and launch it.

```
#!/bin/bash
cd /etc/swift
rm -f *.builder *.ring.gz backups/*.builder backups/*.ring.gz

swift-ring-builder account.builder create 10 2 1
swift-ring-builder account.builder add r1z1-10.23.21.22:6202/mpatha 1
swift-ring-builder account.builder add r1z2-10.23.21.23:6202/mpatha 1
swift-ring-builder account.builder add r1z3-10.23.21.24:6202/mpatha 1
swift-ring-builder /etc/swift/account.builder rebalance

swift-ring-builder container.builder create 10 2 1
swift-ring-builder container.builder add r1z1-10.23.21.22:6201/mpathb 1
swift-ring-builder container.builder add r1z2-10.23.21.23:6201/mpathb 1
swift-ring-builder container.builder add r1z3-10.23.21.24:6201/mpathb 1
swift-ring-builder /etc/swift/container.builder rebalance

swift-ring-builder object.builder create 10 2 1
swift-ring-builder object.builder add r1z1-10.23.21.22:6200/mpathc 1
swift-ring-builder object.builder add r1z2-10.23.21.23:6200/mpathc 1
swift-ring-builder object.builder add r1z3-10.23.21.24:6200/mpathc 1
swift-ring-builder /etc/swift/object.builder rebalance
```

2. Fix ownership on the `/etc/swift` directory after creating the rings.

```
chown -R root:swift /etc/swift
```

3. Copy each ring builder file to the other node in the cluster, storing them under the same `/etc/swift` directory:

```
scp /etc/swift/*.gz Controller2:/etc/swift
scp /etc/swift/*.gz Controller3:/etc/swift
```

Complete the remaining steps on all OpenStack controller nodes provisioned in the environment:

1. Start the Swift proxy **service on all controller** nodes.

```
systemctl start openstack-swift-proxy
```

2. Enable Swift proxy to be persistent throughout reboots **on all** Controller nodes.

```
systemctl enable openstack-swift-proxy
```

3. Verify that the Proxy services are listening and available by using the following command:

```
lsof -i :8080
```

High-Availability for Swift

Swift must have an HAProxy virtual IP address (VIP) configured that can withstand single node failures. Configure the VIP so that you have an endpoint to configure in Keystone.

Complete the following steps on all OpenStack controller nodes provisioned in the environment:

1. Create the `/etc/haproxy/swift.cfg` file and modify the contents of it to suit your environment. This file should be identical on each of the controller nodes, so copy it to all of them.

```
listen swift-proxy-cluster
  bind *:8080
  balance source
  option tcpka
  option tcplog
  server lb-backend-Controller1 10.23.21.23:8080 check inter 2000 rise 2 fall 5
  server lb-backend-Controller2 10.23.21.22:8080 check inter 2000 rise 2 fall 5
  server lb-backend-Controller3 10.23.21.24:8080 check inter 2000 rise 2 fall 5
```

2. Restart the HAProxy service by using Pacemaker so that the changes on all three controller nodes are picked up cluster wide. Complete the following on only one OpenStack controller:

```
pcs resource restart haproxy-clone
pcs resource cleanup
```

Keystone Entry for Swift

Create a service endpoint in Keystone (identity service) that represents the public network interface that all clients access for interaction with Swift. This should typically be an address that is public and accessible to any clients that need Swift.

Complete the following steps on one OpenStack Controller:

1. Source the needed environment parameters.

```
source /root/keystonerc_admin
```

2. Run the following commands. Input an appropriate value for the `proxy_host` variable, which should represent the address customers use to interact with Swift.

```
export proxy_host=<<var_proxy_host_ip>>

keystone endpoint-create --region RegionOne --service-id=$(keystone service-list | awk '/ object-store / {print $2}') \
--publicurl "http://$proxy_host:8080/v1/AUTH_$(tenant_id)s" \
--internalurl "http://$proxy_host:8080/v1/AUTH_$(tenant_id)s" \
--adminurl "http://$proxy_host:8080/"
```



The region specification of RegionOne is needed so that Swift can be accessed from the Horizon dashboard.

Pacemaker Configuration for Swift

Pacemaker resource records for all of the Swift storage services (ACO) as well as the proxy server must also be configured. This enables Pacemaker to monitor the individual controller systems and make sure that all services not only start in the order that they should on system boot, but also verifies that the cloud

administrator can observe through the pacemaker interface any startup errors or services that are not running or are having problems.

Complete the following steps on one OpenStack controller:

1. Configure Pacemaker for the account, container, and object service portions of Swift.

```
pcs resource create swift-account systemd:openstack-swift-account --clone interleave=true
pcs constraint colocation add swift-account-clone with swift-account-fs-clone
pcs constraint order start swift-account-fs-clone then swift-account-clone

pcs resource create swift-container systemd:openstack-swift-container --clone interleave=true
pcs constraint colocation add swift-container-clone with swift-container-fs-clone
pcs constraint order start swift-container-fs-clone then swift-container-clone

pcs resource create swift-object systemd:openstack-swift-object --clone interleave=true
pcs constraint colocation add swift-object-clone with swift-object-fs-clone
pcs constraint order start swift-object-fs-clone then swift-object-clone
```

2. Configure Pacemaker for the proxy service portion of Swift.

```
pcs resource create swift-proxy systemd:openstack-swift-proxy --clone interleave=true
pcs resource create swift-object-expirer systemd:openstack-swift-object-expirer
pcs constraint order start swift-proxy-clone then swift-object-expirer
```

Verification

To upload some files to the object store as a test and verify that the previous steps were implemented successfully, complete the following steps on one OpenStack controller:

1. Source the needed environment parameters if needed.

```
source /root/keystonerc_admin
```

2. Run the following commands to upload three objects to the c1 container.

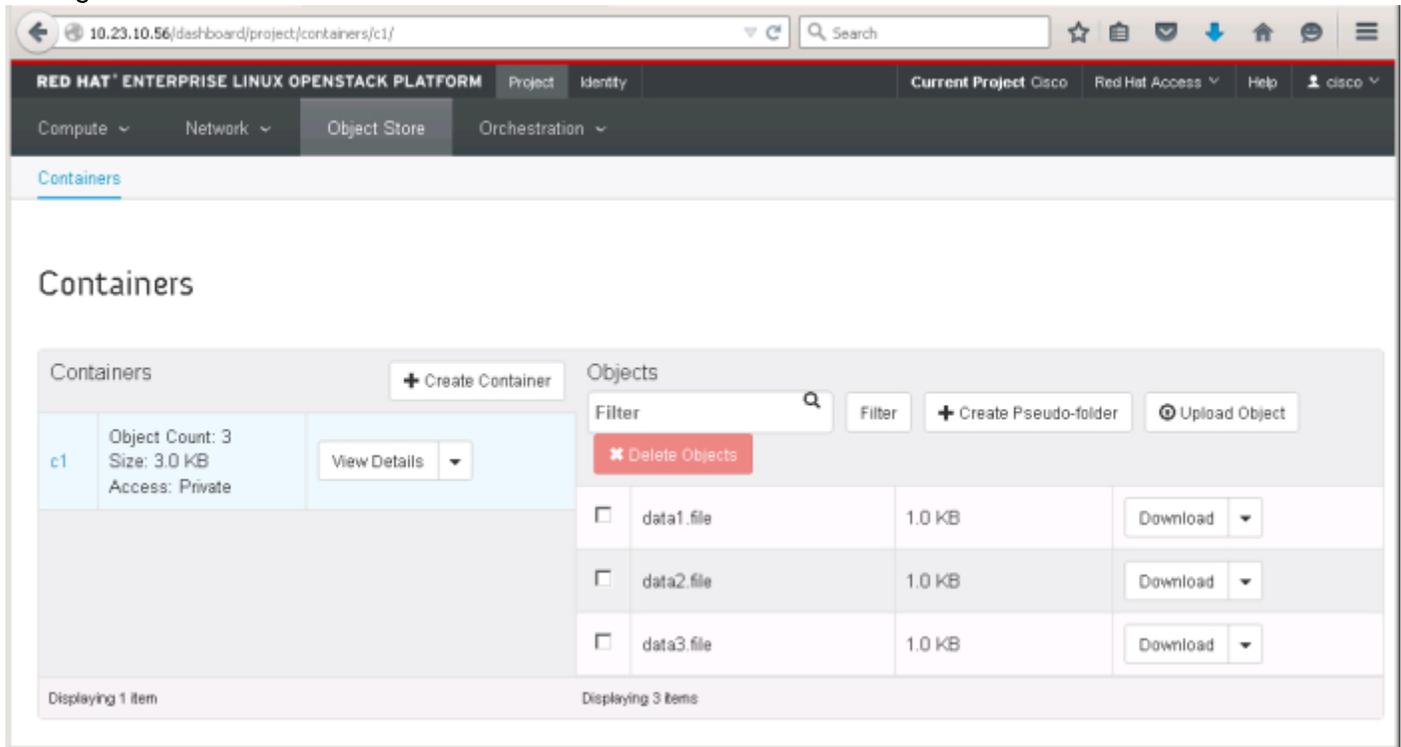
```
cd /root;
head -c 1024 /dev/urandom > data1.file ; swift upload c1 data1.file
head -c 1024 /dev/urandom > data2.file ; swift upload c1 data2.file
head -c 1024 /dev/urandom > data3.file ; swift upload c1 data3.file
```

3. List the contents of the c1 container.

```
swift list c1
data1.file
data2.file
data3.file
```

4. View the same contents from the Horizon dashboard, as is shown in Figure 199:

Figure 199 Swift in Horizon Dashboard



5. You have now uploaded three files into one container. If you check the other storage devices, you find more `.data` files, depending on your replica count.

```
find /srv/node/ -type f -name "*.data"
/srv/node/mpathc/objects/523/40b/82ee32e1c44dfc5e1fc10c5b2947840b/1428956844.78316.data
/srv/node/mpathc/objects/377/02d/5e76b7dc0453cd76e1b3d2e0f826f02d/1428956874.26571.data
/srv/node/mpathc/objects/728/d79/b63d4ee60cc26533cb6c350202838d79/1428956875.99712.data
```

6. To clean up the objects that were uploaded, run the following command:

```
swift delete c1
```

Use Cases

When the OpenStack installation is completed admin and tenant members can begin to put it into use.

Scaling the Environment

The Red Hat Enterprise Linux OpenStack Platform installer not only deploys multi-node OpenStack environments but also provides a means to scale such environments. Scaling an environment is a simple process that only requires adding more hosts to deployment roles.

This section covers the step-by-step process of how to add more compute resources in the existing environment.

Discover a New Host

Scaling environment for additional compute nodes also uses the PXE based discovery to add new hosts.

1. Configure server in Cisco UCS Manager (Refer to Cisco UCS Configuration Section) by using the Openstack_Compute service profile.



It is recommended to add new server(s) in Openstack_Compute pool dedicated for OpenStack Compute role.

2. After the server is booted on PXE/Provisioning network, it will be discovered by Red Hat Enterprise Linux OpenStack Platform installer and registered into Foreman.
3. Log in to the OSP Installer GUI. Click Hosts → Discovered hosts (Figure 200)

Figure 200 Discovered hosts

RED HAT® ENTERPRISE LINUX® OPENSTACK® PLATFORM INSTALLER Admin User

Monitor ▾ Hosts ▾ Configure ▾ Infrastructure ▾ OpenStack Installer ▾ Administer ▾

Discovered hosts

Filter ... Q Search ▾

<input type="checkbox"/>	Name	Model	Subnet	Last facts upload	
<input type="checkbox"/>	mac0025b5000a3d	UCSB-B200-M4	default (10.23.20.0/24)	about 24 hours ago	Provision ▾

Displaying 1 entry

Assign Host to an Existing Deployment

Adding a new host in the already existing Red Hat Enterprise Linux OpenStack Platform requires assigning it to an existing deployment role. Installer will build the new host and then add it to the existing environment.

To add new host to the compute deployment role, complete the following steps:

1. Select OpenStack Installer → Deployments
2. Select the name of the deployment. In this example, Select FLEXPOD_RHEL_OSP6 (Figure 201).

Figure 201 Select OpenStack Deployment

RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM INSTALLER Admin User

Monitor Hosts Configure Infrastructure **OpenStack Installer** Administer

OpenStack Deployments

Filter ... Search

Deployment Name	
FLEXPOD_RHEL_OSP6	Delete

Displaying 1 entry

- In the Deployment Roles section, click + for the Compute deployment role to display the Unassigned Hosts for use in the deployment role.
- Select the check box for the host to be selected.
- Click Assign Hosts to assign the new host to the selected deployment role (Figure 202).

Figure 202 Select Host for the Deployment Role

RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM INSTALLER Admin User

Monitor Hosts Configure Infrastructure **OpenStack Installer** Administer

FLEXPOD_RHEL_OSP6

Deploy Revisit Setup Wizard

Overview Hosts Advanced Configuration

Click to edit..

Deployment Roles

3	Controller	+
5	Compute (Neutron)	+
0	Ceph Storage Node (OSD)	+
0	Generic RHEL 7	+

Unassigned Hosts

	Name	NICs	Storage	Managed?	IP Address
<input checked="" type="checkbox"/>	d mac0025b5000a3d	enp14s0 enp15s0 enp16s0 enp6s0 enp7s0 enp8s0 enp9s0		-	10.23.20.11

Assign Hosts



Multiple Unassigned Hosts can be selected, if deploying multiple hosts in the Compute deployment role.

Configure Host Networking

After you assign a host to a deployment role in a deployment, you can configure the network traffic that each network interface on the host carries.

To configure the host networking, complete the following steps:

- Select Hosts tab.

2. Select Assigned sub-tab
3. Select the host by clicking the check box, then click Configure Networks (Figure 203).

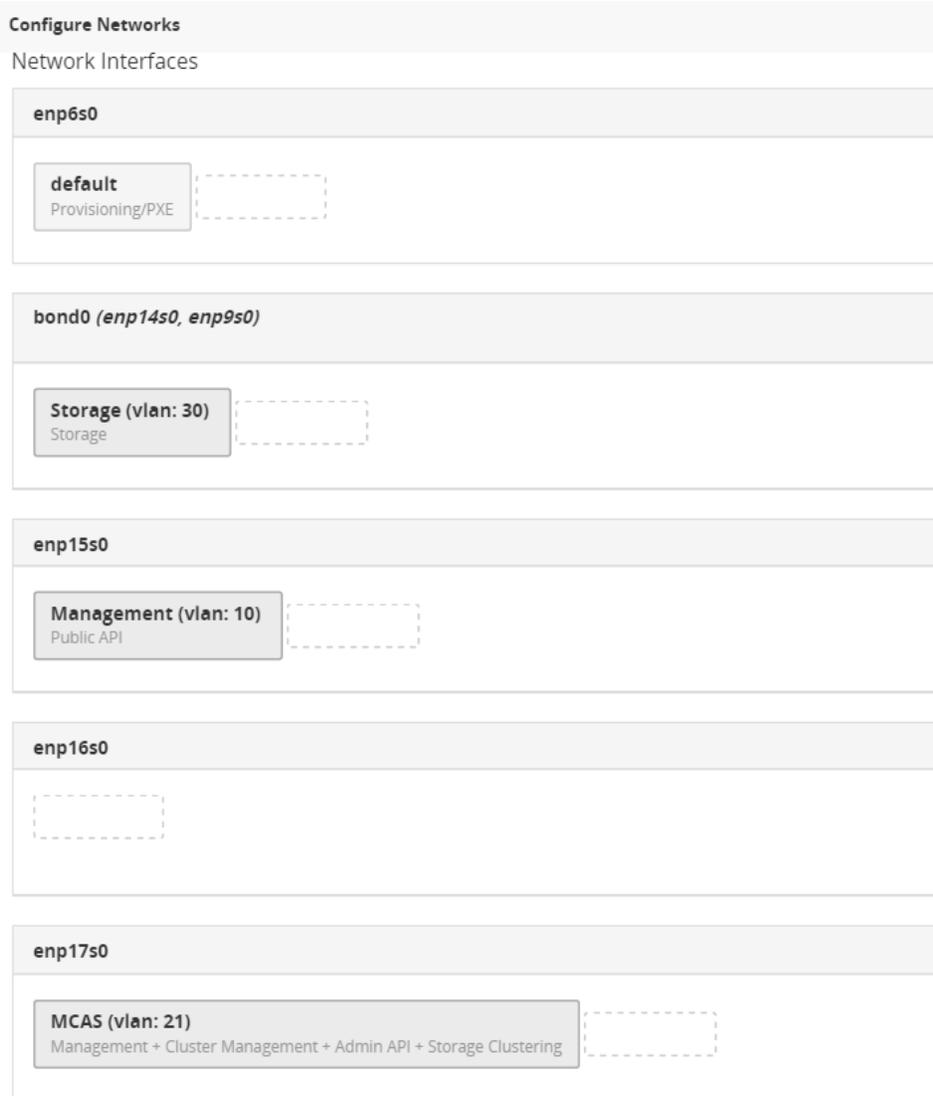
Figure 203 Configure Host Networking

The screenshot shows the 'Assigned Hosts' section of the Red Hat Enterprise Linux OpenStack Platform Installer. The interface includes a navigation bar with 'Monitor', 'Hosts', 'Configure', 'Infrastructure', and 'OpenStack Installer' tabs. The 'Hosts' tab is active, showing a list of assigned hosts. One host is selected, and the 'Configure Networks' button is visible.

Name	Deployment Role	CPUs (cores)	Memory (GB)	Storage	NICs (Subnet)	IP Address
<input checked="" type="checkbox"/> mac0025b5000a3d.sjc.cisco.com	Compute (Neutron)				enp6s0 (default) enp14s0 () enp15s0 () enp16s0 () enp7s0 () enp8s0 ()	10.23.20.11

4. Bond enp14s0 with enp9s0 as shown in Figure (204). Bond0 interface will be created. Keep the bonding mode active-backup.
5. Drag the following networks to their respective network interfaces as below:
 - a. Storage (vlan: 30) → bond0 (enp14s0, enp9s0)
 - b. Management (vlan: 10) → enp15s0
 - c. MCAS (vlan: 21) → enp17s0
 - d. Leave Tenant and External unassigned.
6. Click Done.

Figure 204 Assign Networks to Interfaces

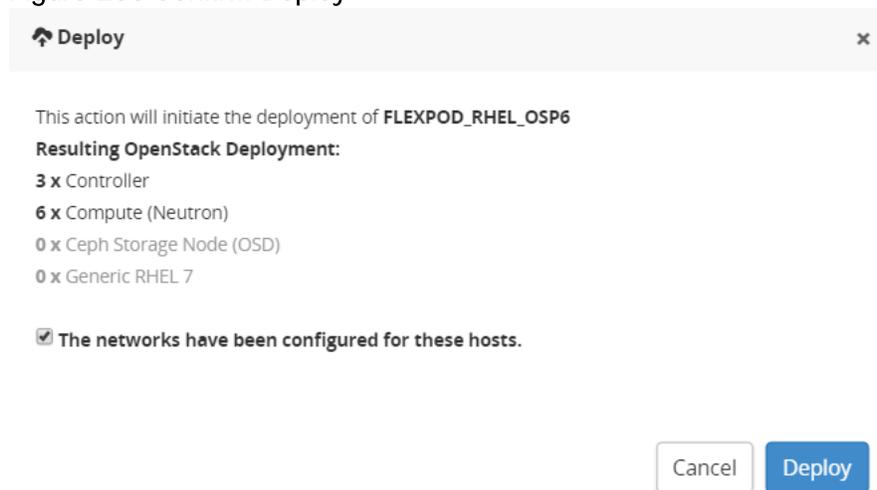


Deploying the New Host

The new host is now assigned to the Compute deployment role and its networking is configured. The final step is to deploy the new host.

1. Click Deploy
2. Select **“The networks have been configured for these hosts.”** Click **Deploy** on the confirmation screen (Figure 205).

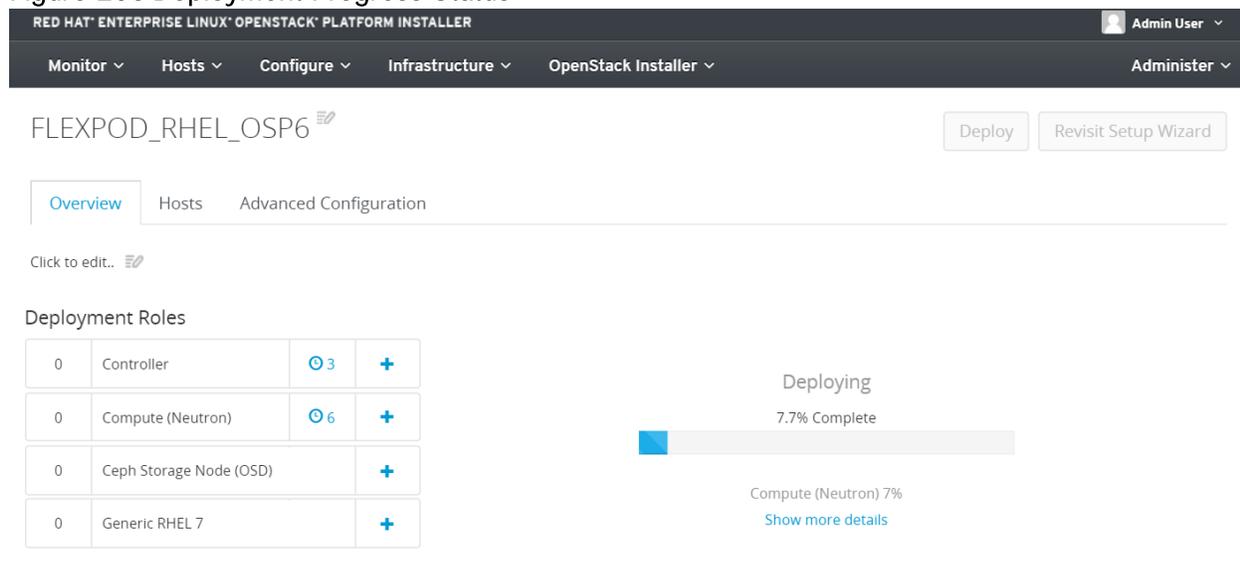
Figure 205 Confirm Deploy



3. Deployment of the new host will begin and its progress can be monitored in the progress bar or in the Task tab of Deployment Status by clicking “Show more details” (Figure 206)

After the deployment is successfully completed, the new host moves to the production environment.

Figure 206 Deployment Progress Status



OpenStack Admin Use Cases

Admin needs to setup few items for tenant members to use the environment. Tenant would need publicly routable provider networks on which to place their externally facing VMs. Tenants should also be provided with a catalog of well-known operating system images for ease of deployment.

Setup Glance Image

Tenant members will be able to upload their own customized images into glance for their own use. However, it is helpful to have some generic VMs of frequently used operating systems already available. Most images

available for download are in the qcow2 format to save space. Table 28 shows the Glance images. Download the images from their respective URL.

Table 28 KVM Guest Images

Operating System	URL
Red Hat Enterprise Linux 7.1	https://access.redhat.com/downloads/
Fedora 22	https://download.fedoraproject.org/pub/fedora/linux/releases/22/Cloud/x86_64/Images/Fedora-Cloud-Base-22-20150521.x86_64.qcow2
Ubuntu Trusty	http://cloud-images.ubuntu.com/trusty/current/trusty-server-cloudimg-amd64-disk1.img
Cirros 3.2	http://download.cirros-cloud.net/0.3.2/cirros-0.3.2-x86_64-disk.img



Ubuntu Trusty and Cirros 3.2 are **NOT** certified guest OS images. For certified guest Operating Systems in Red Hat Enterprise Linux OpenStack Platform, visit the following link:

<https://access.redhat.com/articles/973163>

To setup Glance images within OpenStack, complete the following steps:

1. Click Admin → Images → Create Image (Figure 207).

Figure 207 Create Glance Image

Create An Image
✕

Name *

Description:

Specify an image to upload to the Image Service.

Currently only images available via an HTTP URL are supported. The image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)

Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.

Description

Image Source

Image File

 `rhel-guest-i...x86_64.qcow2`

Format *

Architecture

Minimum Disk (GB)

Minimum RAM (MB)

Public

Protected

2. Click Create Image

Create Tenant and Users

To setup OpenStack tenant also known as Projects, complete the following steps:

1. Identity → Projects → Create Project (Figure 208).

Figure 208 Create OpenStack Project (Tenant)

RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM | Project | Admin | Identity | Router | Managing Region: RegionOne | Red Hat Access | Help | admin

Identity | **Projects** | Users

Projects

Projects Filter + Create Project ✖ Delete Projects

<input type="checkbox"/>	Name	Description	Project ID	Enabled	Actions
<input type="checkbox"/>	admin	admin tenant	8ee1d5b063ce4548a5497b8b7aed0d7d	True	Modify Users
<input type="checkbox"/>	services	Tenant for the openstack services	502f600506a6496fbdda8408a65740f8	True	Modify Users

Displaying 2 items

- On Project Information Tab, enter project name in the Name field. In this case, for example Cisco (Figure 209).



If needed, Modify Quota by clicking Quota tab for vCPU, Number of Instance, Number of Volumes and so on.

Figure 209 Create Project – Project Information

Create Project

Project Information * | Project Members | Quota *

Name *

Description

Enabled

Create a project to organize users.

Cancel Create Project

- Click Create Project.
- Create User by Clicking Identity → Users tab.
- Click Create User (Figure 210) and fill in the following information

- a. Enter User Name
 - b. Enter Email – optional
 - c. Enter Password and Confirm it by type again.
 - d. Select Cisco from Primary Project drop-down list.
 - e. Select `_member_` from Role drop-down list.
6. Click Create User to create a user.

Figure 210 OpenStack – Create User

×

Create User

User Name *

Email

Password *

Confirm Password *

Primary Project *

Cisco
+

Role *

member
▼

Description:

Create a new user and set related properties including the Primary Project and Role.

Create External or Provider Network

All neutron networks are segmented by VLANs in this validation. While the tenants will be able to create their own networks to suit their needs, the tenant created networks will not be accessible from outside. Instead, the admin will create a set of provider networks which will have a provider router, be shared across tenants, and used for external facing VLANs. In the test topology, external VLAN 215 is used as a provider network. The tenant admin needs to create neutron networks and associated subnets. To create an external network, complete the following steps:

1. Click Admin → Networks → Create Network. Create Network window will pop-up (Figure 211)
2. Fill in the information appropriately as shown in Figure 211 as an example.

Figure 211 Create External Network

×

Create Network

Name	<input type="text" value="External-Net"/>	Description:
Project *	<input type="text" value="Cisco"/>	Create a new network for any project as you need.
Provider Network Type * ?	<input type="text" value="VLAN"/>	Provider specified network can be created. You can specify a physical network type (like Flat, VLAN, GRE, and VXLAN) and its segmentation_id or physical network name for a new virtual network.
Physical Network * ?	<input type="text" value="physnet-tenants"/>	In addition, you can create an external network or a shared network by checking the corresponding checkbox.
Segmentation ID * ?	<input type="text" value="215"/>	
Admin State *	<input type="text" value="UP"/>	
<input checked="" type="checkbox"/> Shared		
<input checked="" type="checkbox"/> External Network		

3. Click newly created network, External-Net in this case.
4. Click Create Subnet
5. Enter Subnet Name (Figure 212).
6. Enter external or provider subnet Network Address.
7. Provide default Gateway IP.

Figure 212 Create External Network → Create Subnet

[×](#)

Create Subnet

Subnet

Subnet Detail

Subnet Name

Network Address ?

IP Version *

Gateway IP ?

Disable Gateway

Create a subnet associated with the network. Advanced configuration is available by clicking on the "Subnet Detail" tab.

« Back

Next »

8. Click Next
9. Enter Subnet Detail such as Allocation Pools and DNS Server. More than one DNS servers can be specified as one DNS entry per line (0).
10. Click Create to create a subnet.

Figure 213 Create External Network → Create Subnet → Subnet Detail

×

Create Subnet

Subnet *

Subnet Detail

Enable DHCP

Specify additional attributes for the subnet.

Allocation Pools ?

DNS Name Servers ?

Host Routes ?

← Back

Create

Now the tenant Cisco is setup with external network. You can login as a tenant and start provisioning the tenant networks and attach a VM to it.

OpenStack Tenant Use Cases

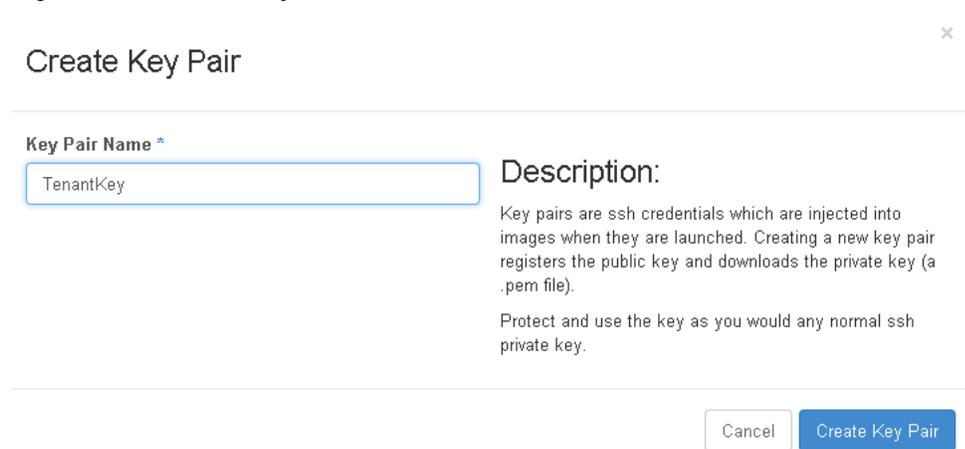
When the admin has setup the OpenStack cluster, the tenants can create private networks, create volumes, and spin up different kinds of instances by logging in Horizon as a tenant.

Create Key

Create a key pair for SSH to cloud instances. Key pair can be created by the completing the following steps:

1. Click Project → Compute → Access & Security → Key Pairs → Create Key Pair
2. Create Key Pair window will pop-up as shown in Figure 214, enter Key Pair Name
3. Click Create Key Pair.

Figure 214 Create Key Pair



Create Key Pair ×

Key Pair Name *

TenantKey

Description:

Key pairs are ssh credentials which are injected into images when they are launched. Creating a new key pair registers the public key and downloads the private key (a .pem file).

Protect and use the key as you would any normal ssh private key.

4. TenantKey will be downloaded, or else click Download Key pair “TenantKey” by the clicking the link.

Modify Default Security Group

Security groups allow the tenant or administrator to specify the type of traffic and direction (ingress/egress) permitted or denied within to pass through a port. Security group is comprised of set of rules. Below steps need to performed to modify the default security group by logging in as a tenant. A new security group can also be created and attach to a VM while provisioning the instance. For simplicity, default security group is modified to allow ICMP and SSH traffic.

1. Click Project → Compute → Access & Security → Security Groups
2. Click Manage Rules for default security group
3. Click + Add Rule button
4. Select ALL ICMP in the Rule drop-down list.
5. Select Ingress in the Direction drop-down list (Figure 215).
6. Click Add.



A new security group can also be created by clicking Create Security Group.

Figure 215 Security Group → Add Rule (ALL ICMP)

×

Add Rule

Rule *

ALL ICMP ▼

Direction

Ingress ▼

Remote * ⓘ

CIDR ▼

CIDR ⓘ

0.0.0.0/0

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel

Add

7. Click + Add Rule button
8. Select Custom TCP Rule in the Rule drop-down list.
9. Select Ingress in the Direction drop-down list
10. Select Port in the Open Port drop-down list.
11. Enter 22 for Port (Figure 216)
12. Click Add.

Figure 216 Add Rule – Custom TCP Rule for SSH (Port 22)

Add Rule ×

Rule *

Direction

Open Port *

Port ?

Remote * ?

CIDR ?

Description:
 Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:
Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.
Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Create Tenant Network

Tenant networks provide internal connectivity between the instances spawned by the same tenant. To create a tenant internal private network, complete the following steps:

1. Click Project → Network → Networks → Create Network
2. Provide Network Name and Click Next (Figure 217).

Figure 217 Create Tenant Network

Create Network ×

Network * Subnet * Subnet Detail

Network Name

Admin State * ?

Create a new network. In addition a subnet associated with the network can be created in the next panel.

3. Enter Subnet Name and Network Address as shown in Figure 218.

4. Click Next.

Figure 218 Create Tenant Network - Create Subnet

[×](#)

Create Network

Network *Subnet *Subnet Detail

Create Subnet

Subnet Name

Network Address ?

IP Version *

Gateway IP ?

Disable Gateway

Create a subnet associated with the new network, in which case "Network Address" must be specified. If you wish to create a network without a subnet, uncheck the "Create Subnet" checkbox.

« BackNext »

5. Enter Allocation Pools and DNS Name Servers (Figure 219).
6. Click Create to create a tenant network.

Figure 219 Create Tenant Network → Create Subnet → Subnet Detail

Subnet *
Subnet Detail
×

Create Subnet

Enable DHCP

Allocation Pools ?

DNS Name Servers ?

Host Routes ?

Specify additional attributes for the subnet.

← Back
Create

Launch Tenant VM

Tenant users can now launch instances of desired flavor using a preconfigured Glance image as a template. The type of network associated with the instance determines the kind of connectivity between the VMs. To launch the tenant VM, complete the following steps:

1. VM Instance on Tenant Network (Internal):
 - a. Click Project → Compute → Instances → Launch Instance
 - b. Select Availability Zone, enter Instance Name, and select Flavor.
 - c. Select Boot from Image (creates a new volume) from Instance Boot Source drop-down list.
 - d. Select image to be provisioned from Image Name drop down
 - e. Select Delete on Terminate, if you would like volume to be deleted when VM is terminated (Figure 220)

Figure 220 Launch VM – Internal Network

Launch Instance x

Details *
Access & Security *
Networking *
Post-Creation
Advanced Options

Availability Zone
nova ▼

Instance Name *
Tenant_VM1

Flavor * ⓘ
m1.small ▼

Instance Count * ⓘ
1

Instance Boot Source * ⓘ
Boot from image (creates a new volum ▼

Image Name
RHEL 7.1 (406.2 MB) ▼

Device size (GB) ⓘ
20

Device Name ⓘ
vda

Delete on Terminate ⓘ

Specify the details for launching an instance.
The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	m1.small
VCPUs	1
Root Disk	20 GB
Ephemeral Disk	0 GB
Total Disk	20 GB
RAM	2,048 MB

Project Limits

Number of Instances 0 of 10 Used

Number of VCPUs 0 of 20 Used

Total RAM 0 of 51,200 MB Used

Cancel
Launch

f. Select Access & Security Tab

g. Select TenantKey key pair from the Key Pair drop down list (Figure 221)

Figure 221 Launch Instance → Access & Security

Launch Instance x

Details *
Access & Security *
Networking *
Post-Creation
Advanced Options

Key Pair ⓘ
TenantKey ▼ +

Security Groups * ⓘ
 default

Control access to your instance via key pairs, security groups, and other mechanisms.

Cancel
Launch

h. Select Networking Tab

i. Select tenant private network. Tenant-Net1 in this example (Figure 222).

j. Select default-pp Nexus 1000v policy profile from Policy Profile drop-down list.

k. Click Launch to launch the VM



default-pp was created previously in Cisco Nexus 1000v configuration section. Security rules can be implemented in default-pp Nexus 1000v policy profile.

Figure 222 Lunch Instance → Networking

Launch Instance

Details * Access & Security * **Networking *** Post-Creation Advanced Options

Networks *

External-Net

Tenant-Net1

Policy Profiles *

default-pp

Select networks for your instance.

Cancel Launch

2. VM Instance on Dual Networks

- Click Project → Compute → Instances → Launch Instance
- Select Availability Zone, enter Instance Name, and select Flavor.
- Select Boot from Image (creates a new volume) from Instance Boot Source drop-down list.
- Select image to be provisioned from Image Name drop down
- Select Delete on Terminate, if you would like volume to be deleted when VM is terminated (Figure 223)

Figure 223 Launch VM – Dual Networks

Launch Instance x

Details *
Access & Security *
Networking *
Post-Creation
Advanced Options

Availability Zone

nova ▼

Instance Name *

Tenant_VM2

Flavor *

m1.small ▼

Instance Count *

1

Instance Boot Source *

Boot from image (creates a new volum

Image Name

RHEL 7.1 (406.2 MB) ▼

Device size (GB) *

20

Device Name *

vda

Delete on Terminate *

Specify the details for launching an instance.

The chart below shows the resources used by this project in relation to the projects quotas.

Flavor Details

Name	m1.small
VCPUs	1
Root Disk	20 GB
Ephemeral Disk	0 GB
Total Disk	20 GB
RAM	2,048 MB

Project Limits

Number of Instances 1 of 10 Used

Number of VCPUs 1 of 20 Used

Total RAM 2,048 of 51,200 MB Used

Cancel
Launch

f. Select Access & Security Tab

g. Select TenantKey key pair from the Key Pair drop down list (Figure 224)

Figure 224 Launch VM – Dual Networks → Access & Security

Launch Instance x

Details *
Access & Security *
Networking *
Post-Creation
Advanced Options

Key Pair *

TenantKey ▼ +

Security Groups *

default

Control access to your instance via key pairs, security groups, and other mechanisms.

Cancel
Launch

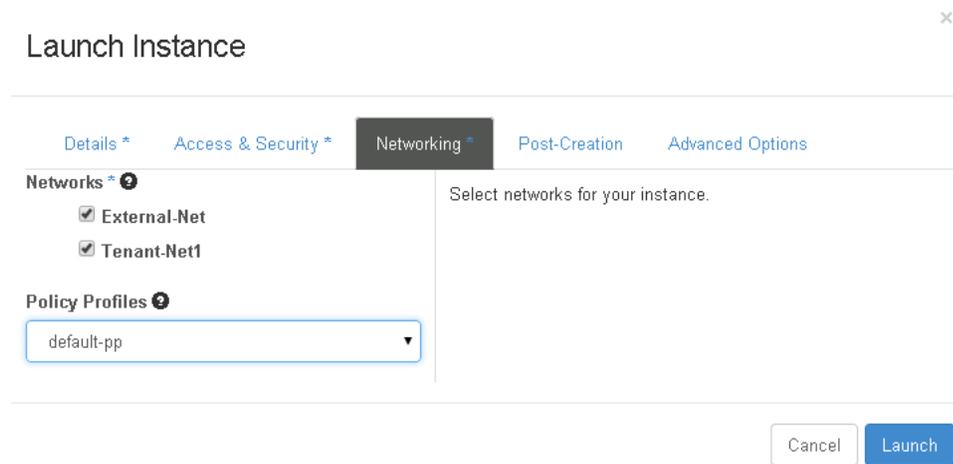
h. Select Networking Tab

i. Select external network and tenant private network for Networks. External-Net and Tenant-Net1 in this example (Figure 225).

j. Select default-pp Nexus 1000v policy profile from Policy Profile drop-down list.

k. Click Launch to launch the VM

Figure 225 Launch VM – Dual Networks → Networking



Launch Instance ×

Details * Access & Security * **Networking *** Post-Creation Advanced Options

Networks * ⓘ

External-Net

Tenant-Net1

Policy Profiles * ⓘ

default-pp ▼

Select networks for your instance.

Cancel Launch

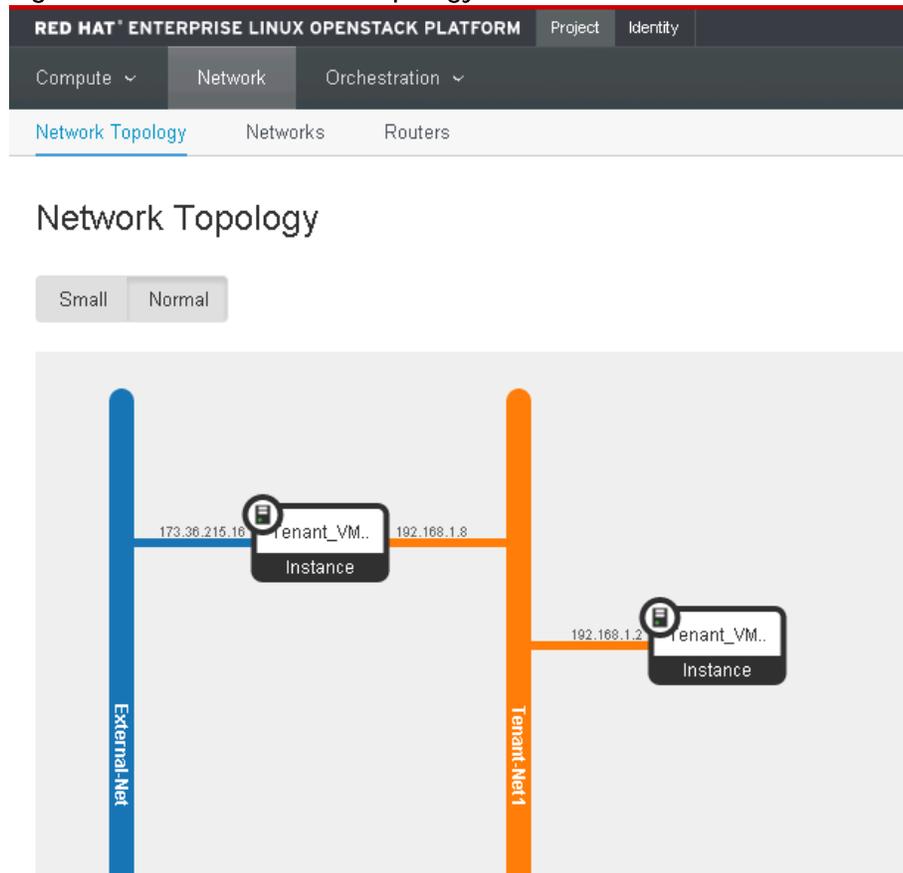
3. VM Instance on External Network

Tenant VM can also be launched only on external network by following the above dual network VM launch example. On the Networking tab, only select External-Net for Networks. Rest of the step-by-step process will remain the same.

Network Topology View

After provisioning tenant network and tenant VMs, the full topology can be viewed by clicking Project → Network → Network Topology. **Tenant's network topology will look like as shown in Figure 226.**

Figure 226 Tenant Network Topology

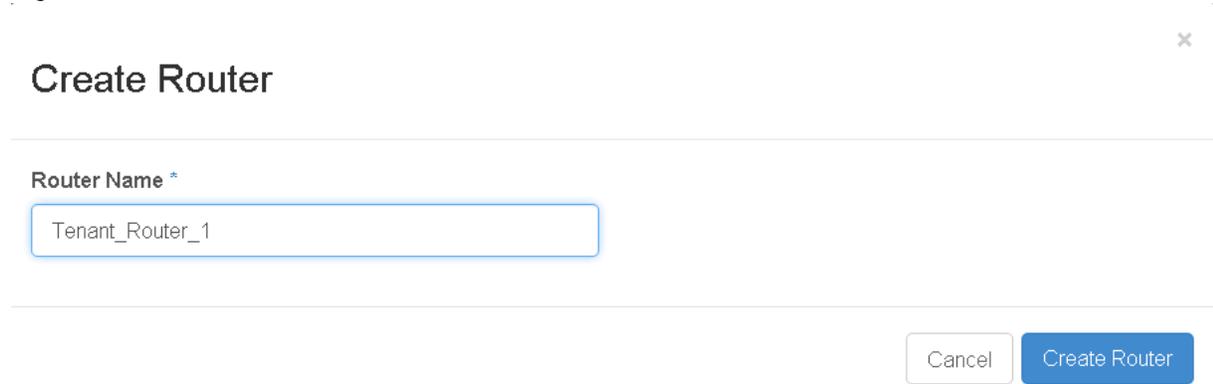


Create Tenant Neutron Router

Tenant private (internal) network can also be configured for external access within OpenStack by provisioning a Neutron router in the Horizon dashboard. To provision a tenant router between the external and internal network, complete the following steps:

1. Click Project → Network → Routers → Create Router
2. Enter router name (Figure 227)
3. Click Create Router.

Figure 227 Create Router



Create Router

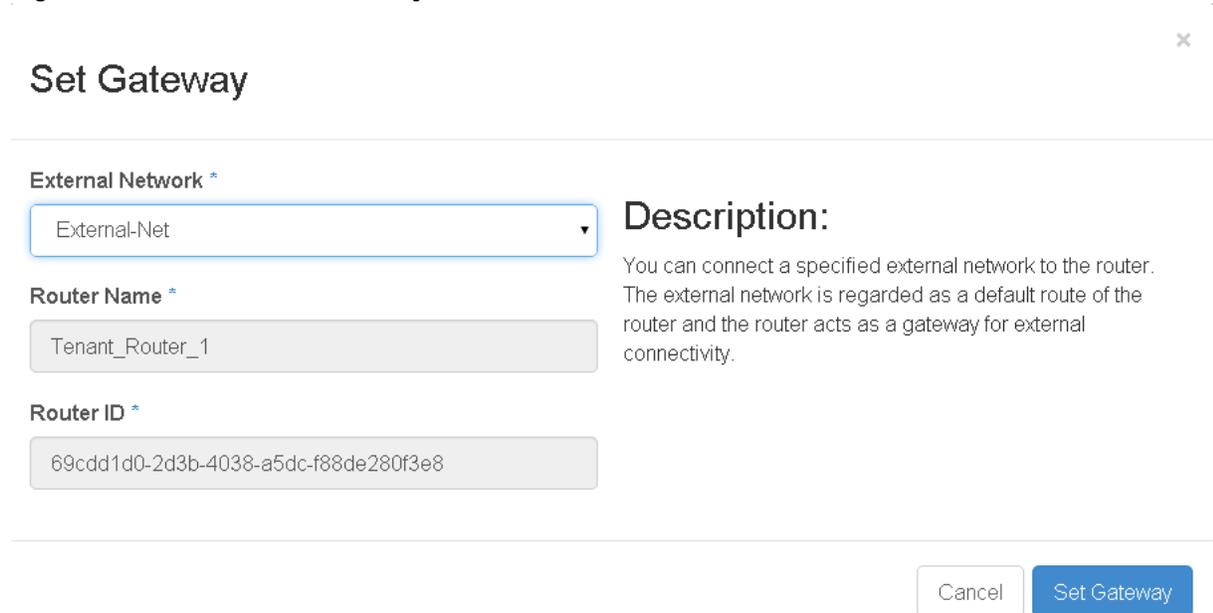
Router Name *

Tenant_Router_1

Cancel Create Router

4. Select Set Gateway for Tenant_Router_1
5. Select External-Net from External Network drop-down list (Figure 228).
6. Click Set Gateway.

Figure 228 Router – Set Gateway



Set Gateway

External Network *

External-Net

Router Name *

Tenant_Router_1

Router ID *

69cdd1d0-2d3b-4038-a5dc-f88de280f3e8

Description:
You can connect a specified external network to the router. The external network is regarded as a default route of the router and the router acts as a gateway for external connectivity.

Cancel Set Gateway

7. Click Tenant_Router_1.
8. Click Add Interface (Figure 229).

Figure 229 Router – Add Interface

✕

Add Interface

Subnet *

Tenant-Net1: 192.168.1.0/24 (Subnet-1) ▾

IP Address (optional) ⓘ

Router Name *

Tenant_Router_1

Router ID *

69cdd1d0-2d3b-4038-a5dc-f88de280f3e8

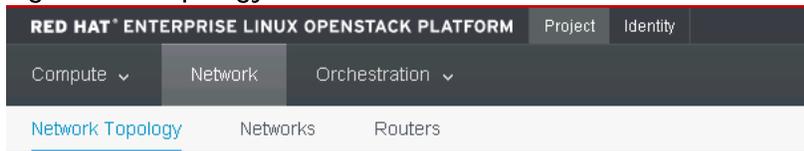
Description:

You can connect a specified subnet to the router.

The default IP address of the interface created is a gateway of the selected subnet. You can specify another IP address of the interface here. You must select a subnet to which the specified IP address belongs to from the above list.

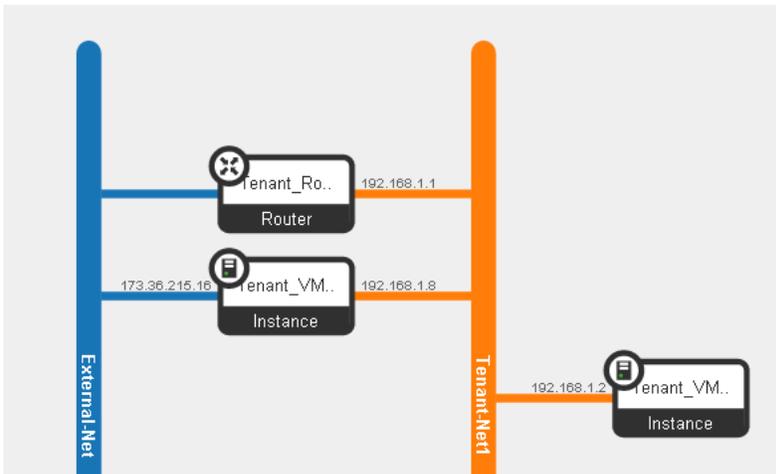
9. Select Tenant-Net1 (tenant private network) from the Subnet drop-down list.
10. Click Add Interface.
11. For Topology view, click Project → Network → Network Topology (Figure 230)

Figure 230 Topology View with Neutron Router



Network Topology

Small Normal



Bill of Materials

Table 29 FlexPod BOM

Item Name	Description	Quantity
UCS-FI-6248UP	UCS 6248UP 1RU Fabric Int/No PSU/32 UP/ 12p LIC	2
UCS-ACC-6248UP	UCS 6248UP Chassis Accessory Kit	2
UCS-PSU-6248UP-AC	UCS 6248UP Power Supply/100-240VAC	4
N10-MGT011	UCS Manager v2.1	2
UCS-BLKE-6200	UCS 6200 Series Expansion Module Blank	2
UCS-FI-DL2	UCS 6248 Layer 2 Daughter Card	2
UCS-FAN-6248UP	UCS 6248UP Fan Module	4
UCSB-5108-AC2	UCS 5108 Blade Server AC2 Chassis 0 PSU/8 fans/0 FEX	2
N01-UAC1	Single phase AC power module for UCS 5108	2
N20-FAN5	Fan module for UCS 5108	16
UCSB-5108-PKG-HW	UCS 5108 Packaging for chassis with half width blades.	2
N20-CBLKB1	Blade slot blanking panel for UCS 5108/single slot	8
N20-CAK	Accessory kit for UCS 5108 Blade Server Chassis	2
UCSB-B200-M4	UCS B200 M4 w/o CPU mem drive bays HDD mezz	8
UCS-CPU-E52660D	2.60 GHz E5-2660 v3/105W 10C/25MB Cache/DDR4 2133MHz	16
UCS-MR-1X162RU-A	16GB DDR4-2133-MHz RDIMM/PC4-17000/dual rank/x4/1.2v	192
UCSB-MLOM-40G-03	Cisco UCS VIC 1340 modular LOM for blade servers	8
UCSB-HS-EP-M4-F	CPU Heat Sink for UCS B200 M4/B420 M4 (Front)	8
UCSB-HS-EP-M4-R	CPU Heat Sink for UCS B200 M4/B420 M4 (Rear)	8
UCSB-LSTOR-BK	FlexStorage blanking panels w/o controller w/o drive bays	16
UCS-IOM-2204XP	UCS 2204XP I/O Module (4 External 16 Internal 10Gb Ports)	4
UCSB-PSU-2500ACDV	2500W Platinum AC Hot Plug Power Supply - DV	8
CAB-AC-2500W-US1	Power Cord 250Vac 16A straight blade NEMA 6-20 plug US	8
N9K-C9372PX	Nexus 9300 with 48p 10G SFP+ and 6p 40G QSFP+	2
N3K-C3064-ACC-KIT	Nexus 3064PQ Accessory Kit	2

Item Name	Description	Quantity
NXA-FAN-30CFM-F	Nexus 2K/3K/9K Single Fan port side exhaust airflow	8
N9K-PAC-650W-B	Nexus 9300 650W AC PS Port-side Exhaust	4
N9KDK9-612I3.3A	Nexus 9500 or 9300 Base NX-OS Software Rel 6.1(2)I3(3A)	2
FAS8040A-001-R6	FAS8040 High Availability System	2
X6227-R6-C	Chassis,FAS8040/60/80 W/CNTRL Slots,AC PS,-C	1
DS2246-21.6TB-0P-R6-C	DSK SHLF,24x900GB,6G,0P,-C	1
X800-42U-R6-C	Power Cable,In-Cabinet,C13-C14,-C	4
X1973A-R6-C	Flash Cache 512GB PCIe Module 2,-C	2
SVC-FLEXPOD-SYSTEMS	Systems Used in FlexPod Solution, Attach PN	1
X-SFP-H10GB-CU3M-R6-C	Cable,Cisco 10GBase Copper SFP+ 3m,-C	4
X6566B-2-R6-C	Cable,Direct Attach CU SFP+ 10G,2M,-C	2
DOC-80XX-C	Documents,80xx,-C	1
MULTIPATH-C	Multipath configuration	1
STACKS	Storage Stacks Attached Quantity	1
CNA-OB-PR-10GBE-E0E-E0F	CNA Onboard 10GbE,e0e-e0f	2
CNA-OB-PR-10GBE-E0G-E0H	CNA Onboard 10GbE,e0g-e0h	2
X4525A-R6-C	Nameplate,FAS8040,-C	1
SES-SYSTEM	Support Edge Services Attach PN	1
OS-ONTAP-CAP2-0P-C	OS Enable,Per-0.1TB,ONTAP,Perf-Stor,0P,-C	216
SWITCHLESS	2-Node Switchless Cluster	1
SW-ONTAP8.3	SW,Data ONTAP8.3	2
SW-2-8040A-ISCSI-C	SW-2,iSCSI,8040A,-C	2
SW-2-8040A-FLEXCLN-C	SW-2,Flexclone,8040A,-C	2
SW-2-8040A-NFS-C	SW-2,NFS,8040A,-C	2
SVC-A2-IN-NBR-Y	HW Support,Standard2 Replace,Inst,NBD,y	1
SW-SSP-A2-IN-NBR-Y	SW Subs,Standard2 Replace,Inst,NBD,y	1
CS-OS-SUPPORT-ONTAP	OS Support Entitlement,ONTAP	1
X6566B-2-R6-C	Cable,Direct Attach CU SFP+ 10G,2M,-C	2
E5500B-12GB-R6-C	E5500B,12GB Controller,No HIC,-C	2

Item Name	Description	Quantity
SVC-FLEXPOD-SYSTEMS	Systems Used in FlexPod Solution, Attach PN	1
E-X4044A-R6-C	Disk Drive,2TB,7.1k,Non-FDE,DE6600,-C	60
E-X5680A-R6-C	Enclosure,4U-60,DE6600,Empty,2PSU,-C	1
DOC-E-SERIES-4U-SYS-C	Install Documents,System,DE6600,-C	1
X-48895-00-R6-C	SFP,10Gb iSCSI/16Gb FC,Unified,E-Series,-C	4
X-56017-00-R6-C	HIC,E5500B,10Gb iSCSI,4-ports,-C	2
SES-SYSTEM	Support Edge Services Attach PN	1
X-48619-00-R6-C	Battery,E5400,E5500,E5600,-C	2
SW-ESERIES-SANTRICTY	SW,E-Series,SANtricity	1
OS-SANTRICITY-CAP2-0P-C	OS Enable,Per-0.1TB,SANTRCTY,Perf-Stor,0P,-C	1200
X6536-R6	Cable,Cntrl-Shelf/Switch,5m,LC/LC,Op	4
X6562-R6	Cable,Ethernet,5m RJ45 CAT6	2
SVC-A2-IN-NBR-Y	HW Support,Standard2 Replace,Inst,NBD,y	1
SW-SSP-A2-IN-NBR-Y	SW Subs,Standard2 Replace,Inst,NBD,y	1
SVC-INST-A2-IN1-NBR-Y	Initial Install,Standard2 Replace,Inst,NBD,y	1
CS-OS-SUPPORT-SANTRICITY	OS Support Entitlement,SANTRICITY	1

Other Resources

The following resources complement the material presented in this document and serve as additional reference sources.

Cisco UCS

The following links provide additional information about Cisco UCS:

- Cisco Design Zone for Data Centers
<http://www.cisco.com/c/en/us/solutions/enterprise/design-zone-data-centers/index.html>
- Cisco UCS
<http://www.cisco.com/c/en/us/products/servers-unified-computing/index.html>
- Cisco UCS 6200 Series Fabric Interconnects
<http://www.cisco.com/en/US/products/ps11544/index.html>
- Cisco UCS 5100 Series Blade Server Chassis
<http://www.cisco.com/en/US/products/ps10279/index.html>
- Cisco UCS B-Series Blade Servers
<http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-b-series-blade-servers/index.html>
- Cisco UCS Adapters
http://www.cisco.com/en/US/products/ps10277/prod_module_series_home.html
- Cisco UCS Manager
<http://www.cisco.com/en/US/products/ps10281/index.html>

Cisco Nexus Networking

The following links provide additional information about Cisco Nexus 9000 Series switches:

- Cisco Nexus 9000 Series Switches
<http://www.cisco.com/c/en/us/products/switches/nexus-9000-series-switches/index.html>
- Cisco Nexus 9000 Series Configuration Guides
<http://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/products-installation-and-configuration-guides-list.html>
- Cisco Nexus 1000V for KVM General Information
<http://www.cisco.com/c/en/us/products/switches/nexus-1000v-kvm/index.html>
- Cisco Nexus 1000V for KVM Configuration Guides
<http://www.cisco.com/c/en/us/support/switches/nexus-1000v-kvm/products-installation-and-configuration-guides-list.html>

NetApp FAS Storage

The following links provide additional information about NetApp FAS storage:

- Clustered Data ONTAP 8.3 Documentation
<http://mysupport.netapp.com/documentation/docweb/index.html?productID=61999>
- TR-3982: NetApp Clustered Data ONTAP 8.3
<http://www.netapp.com/us/media/tr-3982.pdf>
- TR-4067: Clustered Data ONTAP NFS Best Practice and Implementation Guide
<http://www.netapp.com/us/media/tr-4067.pdf>
- TR-4063: Parallel Network File System Configuration and Best Practices for Clustered Data ONTAP 8.2 and Later
<http://www.netapp.com/us/media/tr-4063.pdf>
- TR-4379: Name Services Best Practices Guide for Clustered Data ONTAP
<http://www.netapp.com/us/media/tr-4379.pdf>

NetApp E-Series Storage

The following links provide additional information about NetApp E-Series storage:

- E5500 Series Documentation
<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=61609>
- SANtricity® Storage Manager 11.20: iSCSI Configuration and Provisioning for Linux
https://library.netapp.com/ecm/ecm_get_file/ECMP1532525
- TR-4365: Introduction to NetApp E-Series E5500 with SANtricity 11.20
<http://www.netapp.com/us/media/tr-4365.pdf>

Red Hat Enterprise Linux OpenStack Platform 6

The following links provide additional information about Red Hat Enterprise Linux OpenStack Platform 6:

- Red Hat Enterprise Linux OpenStack Platform
<https://access.redhat.com/products/red-hat-enterprise-linux-openstack-platform>
- Red Hat Enterprise Linux OpenStack Platform 6 Documentation Home Page
https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/
- OpenStack Object Storage Service Installation
https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Deploying_OpenStack_Learning_Environments/chap-OpenStack_Object_Storage_Service_Installation.html
- Cisco systems with VIC adapters and iSCSI boot on RHEL 7
<https://access.redhat.com/articles/913963>

- What is udev and how do you write custom udev rules in RHEL7 ?
<https://access.redhat.com/solutions/1135513>

OpenStack Upstream

- Juno Release
<http://docs.openstack.org/juno/>

Appendix

Appendix A - Cisco Nexus 9000 Configuration Files

Cisco Nexus 9372 A

```
!Command: show running-config

!Time: Sat Aug  1 01:12:18 2015

version 6.1(2)I3(2)

hostname N9k-FLEXPOD-SwitchA

vdc N9k-FLEXPOD-SwitchA id 1

  allocate interface Ethernet1/1-54

  limit-resource vlan minimum 16 maximum 4094

  limit-resource vrf minimum 2 maximum 4096

  limit-resource port-channel minimum 0 maximum 512

  limit-resource u4route-mem minimum 248 maximum 248

  limit-resource u6route-mem minimum 96 maximum 96

  limit-resource m4route-mem minimum 58 maximum 58

  limit-resource m6route-mem minimum 8 maximum 8

cfs eth distribute

feature udld

feature interface-vlan

feature lacp

feature vpc

username admin password 5 $1$GxlhOQG9$diJSq/cRLgjdXf/R2BgsY0  role network-admin

ip domain-lookup

copp profile strict
```

```
snmp-server user admin network-admin auth md5 0xdbf49acc0b33b7b49d678ebc8a50dc65 priv
0xdbf49acc0b33b7b49d678ebc8a50dc65 localizedkey

rmon event 1 log trap public description FATAL(1) owner PMON@FATAL
rmon event 2 log trap public description CRITICAL(2) owner PMON@CRITICAL
rmon event 3 log trap public description ERROR(3) owner PMON@ERROR
rmon event 4 log trap public description WARNING(4) owner PMON@WARNING
rmon event 5 log trap public description INFORMATION(5) owner PMON@INFO

vlan 1,10,20-21,30,40-41,50-51,60-200,215

vlan 10
    name MGMT-VLAN

vlan 20
    name PXE

vlan 21
    name MCAS

vlan 30
    name NFS-VLAN

vlan 40
    name iSCSI-VLAN-A

vlan 41
    name iSCSI-VLAN-B

vlan 50
    name Swift-A

vlan 51
    name Swift-B

vlan 215
    name External

spanning-tree port type edge bpduguard default
spanning-tree port type edge bpdufilter default
spanning-tree port type network default

vrf context management
```

```
vpc domain 1
  peer-switch
  role priority 10
  peer-keepalive destination 10.23.10.4 source 10.23.10.3
  delay restore 150
  peer-gateway
  auto-recovery
```

```
interface Vlan1
  no shutdown
```

```
interface Vlan40
  no shutdown
  no ip redirects
  ip address 10.23.40.1/24
  no ipv6 redirects
```

```
interface Vlan41
  no shutdown
  no ip redirects
  ip address 10.23.41.1/24
  no ipv6 redirects
```

```
interface Vlan60
  no shutdown
  no ip redirects
  ip address 10.23.60.1/24
  no ipv6 redirects
```

```
interface port-channel10
  description vPC Peer-Link
  switchport mode trunk
  switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215
  spanning-tree port type network
  vpc peer-link
```

```
interface port-channel17
  description UCS-FLEXPOD-FAB-A
  switchport mode trunk
  switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215
  spanning-tree port type edge trunk
  mtu 9216
  vpc 17
```

```
interface port-channel18
  description UCS-FLEXPOD-FAB-B
  switchport mode trunk
  switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215
  spanning-tree port type edge trunk
  mtu 9216
  vpc 18
```

```
interface port-channel23
  description PO23-UPLINK
  switchport mode trunk
  switchport trunk native vlan 10
  switchport trunk allowed vlan 10,215
  vpc 23
```

```
interface port-channel25
```

```
description FAS8040-A
switchport mode trunk
switchport trunk allowed vlan 30,40-41
spanning-tree port type edge trunk
mtu 9216
vpc 25

interface port-channel26
description FAS8040-B
switchport mode trunk
switchport trunk allowed vlan 30,40-41
spanning-tree port type edge trunk
mtu 9216
vpc 26

interface Ethernet1/1
description vPC Peer N9k-FLEXPOD-SwitchB:1/1
switchport mode trunk
switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215
spanning-tree port type network
channel-group 10 mode active

interface Ethernet1/2
description vPC Peer N9k-FLEXPOD-SwitchB:1/2
switchport mode trunk
switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215
spanning-tree port type network
channel-group 10 mode active

interface Ethernet1/17
description UCS-FLEXPOD-FAB-A:1/17
```

```
switchport mode trunk

switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215

spanning-tree port type edge trunk

mtu 9216

channel-group 17 mode active

interface Ethernet1/18

description UCS-FLEXPOD-FAB-B:1/17

switchport mode trunk

switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215

mtu 9216

channel-group 18 mode active

interface Ethernet1/23

description E1/23-UPLINK

switchport mode trunk

switchport trunk native vlan 10

switchport trunk allowed vlan 10,215

channel-group 23 mode active

interface Ethernet1/25

description FAS8040-A:e0e

switchport mode trunk

switchport trunk allowed vlan 30,40-41

mtu 9216

channel-group 25 mode active

interface Ethernet1/26

description FAS8040-B:e0e

switchport mode trunk

switchport trunk allowed vlan 30,40-41
```

```
mtu 9216

channel-group 26 mode active

interface Ethernet1/33

description E5560-CBH1P1

switchport access vlan 50

spanning-tree port type edge

mtu 9216

interface Ethernet1/34

description E5560-CAH1P1

switchport access vlan 50

spanning-tree port type edge

mtu 9216

interface mgmt0

vrf member management

ip address 10.23.10.3/24

line console

line vty

boot nxos bootflash:/n9000-dk9.6.1.2.I3.2.bin
```

Cisco Nexus 9372 B

```
!Command: show running-config

!Time: Sat Aug  1 01:10:49 2015

version 6.1(2)I3(2)

switchname N9k-FLEXPOD-SwitchB

vdc N9k-FLEXPOD-SwitchB id 1

allocate interface Ethernet1/1-54
```

```
limit-resource vlan minimum 16 maximum 4094

limit-resource vrf minimum 2 maximum 4096

limit-resource port-channel minimum 0 maximum 512

limit-resource u4route-mem minimum 248 maximum 248

limit-resource u6route-mem minimum 96 maximum 96

limit-resource m4route-mem minimum 58 maximum 58

limit-resource m6route-mem minimum 8 maximum 8

cfs eth distribute

feature udld

feature interface-vlan

feature lacp

feature vpc

username admin password 5 $1$QOj6hWEP$dCEYbXuVT.2HA187GRQEQ/ role network-admin

ip domain-lookup

copp profile strict

snmp-server user admin network-admin auth md5 0xdb2993dacdc9effaebcb42fda9a1237d priv
0xdb2993dacdc9effaebcb42fda9a1237d localizedkey

rmon event 1 log trap public description FATAL(1) owner PMON@FATAL

rmon event 2 log trap public description CRITICAL(2) owner PMON@CRITICAL

rmon event 3 log trap public description ERROR(3) owner PMON@ERROR

rmon event 4 log trap public description WARNING(4) owner PMON@WARNING

rmon event 5 log trap public description INFORMATION(5) owner PMON@INFO

vlan 1,10,20-21,30,40-41,50-51,60-200,215

vlan 10

    name MGMT-VLAN

vlan 20

    name PXE

vlan 21

    name MCAS
```

```
vlan 30
    name NFS-VLAN
vlan 40
    name iSCSI-VLAN-A
vlan 41
    name iSCSI-VLAN-B
vlan 50
    name Swift-A
vlan 51
    name Swift-B
vlan 215
    name External

spanning-tree port type edge bpduguard default
spanning-tree port type edge bpdufilter default
spanning-tree port type network default
vrf context management
    ip route 0.0.0.0/0 10.23.10.1
vpc domain 1
    peer-switch
    role priority 20
    peer-keepalive destination 10.23.10.3 source 10.23.10.4
    delay restore 150
    peer-gateway
    auto-recovery

interface Vlan1
    no shutdown

interface Vlan40
```

```
no shutdown

no ip redirects

ip address 10.23.40.2/24

no ipv6 redirects

interface Vlan41

no shutdown

no ip redirects

ip address 10.23.41.2/24

no ipv6 redirects

interface Vlan60

no shutdown

no ip redirects

ip address 10.23.60.2/24

no ipv6 redirects

interface port-channel10

description vPC Peer-Link

switchport mode trunk

switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215

spanning-tree port type network

vpc peer-link

interface port-channel17

description UCS-FLEXPOD-FAB-A

switchport mode trunk

switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215

spanning-tree port type edge trunk

mtu 9216

vpc 17
```

```
interface port-channel18
  description UCS-FLEXPOD-FAB-B
  switchport mode trunk
  switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215
  spanning-tree port type edge trunk
  mtu 9216
  vpc 18
```

```
interface port-channel23
  description PO23-UPLINK
  switchport mode trunk
  switchport trunk native vlan 10
  switchport trunk allowed vlan 10,215
  vpc 23
```

```
interface port-channel25
  description FAS8040-A
  switchport mode trunk
  switchport trunk allowed vlan 30,40-41
  spanning-tree port type edge trunk
  mtu 9216
  vpc 25
```

```
interface port-channel26
  description FAS8040-B
  switchport mode trunk
  switchport trunk allowed vlan 30,40-41
  spanning-tree port type edge trunk
  mtu 9216
  vpc 26
```

```
interface Ethernet1/1
  description vPC Peer N9k-FLEXPOD-SwitchA:1/1
  switchport mode trunk
  switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215
  spanning-tree port type network
  channel-group 10 mode active
```

```
interface Ethernet1/2
  description vPC Peer N9k-FLEXPOD-SwitchA:1/2
  switchport mode trunk
  switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215
  spanning-tree port type network
  channel-group 10 mode active
```

```
interface Ethernet1/17
  description UCS-FLEXPOD-FAB-A:1/18
  switchport mode trunk
  switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215
  mtu 9216
  channel-group 17 mode active
```

```
interface Ethernet1/18
  description UCS-FLEXPOD-FAB-B:1/18
  switchport mode trunk
  switchport trunk allowed vlan 10,20-21,30,40-41,50-51,60-200,215
  mtu 9216
  channel-group 18 mode active
```

```
interface Ethernet1/23
  description E1/23-UPLINK
```

```
switchport mode trunk

switchport trunk native vlan 10

switchport trunk allowed vlan 10,215

channel-group 23 mode active

interface Ethernet1/25

description FAS8040-A:e0f

switchport mode trunk

switchport trunk allowed vlan 30,40-41

mtu 9216

channel-group 25 mode active

interface Ethernet1/26

description FAS8040-B:e0f

switchport mode trunk

switchport trunk allowed vlan 30,40-41

mtu 9216

channel-group 26 mode active

interface Ethernet1/33

description E5560-CBH1P3

switchport access vlan 51

spanning-tree port type edge

mtu 9216

interface Ethernet1/34

description E5560-CAH1P3

switchport access vlan 51

spanning-tree port type edge

mtu 9216
```

```

interface mgmt0

    vrf member management

    ip address 10.23.10.4/24

line console

line vty

boot nxos bootflash:/n9000-dk9.6.1.2.I3.2.bin

```

Appendix B: HTTPS Access in Clustered Data ONTAP

To configure secure access to the storage controller, complete the following steps:

1. Increase the privilege level to access the certificate commands.

```

set -privilege diag
Do you want to continue? {y|n}: y

```

2. Typically, a self-signed certificate is already in place. Verify the certificate by running the following command:

```
security certificate show
```

3. For each SVM shown, the certificate common name should match the DNS fully qualified domain name (FQDN) of the SVM. The three default certificates should be deleted and replaced by either self-signed certificates or certificates from a certificate authority (CA). To delete the default certificates, run the following commands:

```
security certificate delete -vserver <cert_vserver_name> -type server [TAB] ...
```

For example:

```
security certificate delete -vserver Infra-SVM -common-name Infra-SVM -ca Infra-SVM -
type server -serial 552429A6
```

4. To generate and install self-signed certificates, run the following commands as one-time commands. Generate a server certificate for **FLEXPOD-OPS-SVM1**, **FLEXPOD-OPENSTACK-SVM** and the cluster SVM.

```

security certificate create -common-name <<operations_vserver_fqdn>> -type server -size 2048 -country US -
state "North_Carolina" -locality "RTP" -organization "NetApp" -unit "CIS" -email-addr "test1@netapp.com" -
expire-days 365 -protocol SSL -hash-function SHA256 -vserver <<operations_vserver>>

security certificate create -common-name <<openstack_vserver_fqdn>> -type server -size 2048 -country US -
state "North_Carolina" -locality "RTP" -organization "NetApp" -unit "CIS" -email-addr "test1@netapp.com" -
expire-days 365 -protocol SSL -hash-function SHA256 -vserver <<openstack_vserver>>

security certificate create -common-name <<clustername_fqdn>> -type server -size 2048 -country US -state
"North_Carolina" -locality "RTP" -organization "NetApp" -unit "CIS" -email-addr "test1@netapp.com" -expire-
days 365 -protocol SSL -hash-function SHA256 -vserver <<clustername>>

```

5. To obtain the parameter values required in the following step, run the `security certificate show` command.
6. Enable each certificate that was just created by using the `-server-enabled true` and `-client-enabled false` parameters. Again use TAB completion.

```
security ssl modify -vserver <<operations_vserver>> -server-enabled true -client-enabled false -ca
<<operations_vserver_fqdn>> -common-name <<operations_vserver_fqdn>> [TAB] ...
security ssl modify -vserver <<openstack_vserver>> -server-enabled true -client-enabled false -ca
<<openstack_vserver_fqdn>> -common-name <<openstack_vserver_fqdn>> [TAB] ...
security ssl modify -vserver <<clustername>> -server-enabled true -client-enabled false -ca
<<clustername_fqdn>> -common-name <<clustername_fqdn>> [TAB] ...
```

7. Configure and enable SSL and HTTPS access and disable HTTP access.

```
system services web modify -external true -ssl3-enabled true
Warning: Modifying the cluster configuration will cause pending web service requests to be
        interrupted as the web servers are restarted.
Do you want to continue {y|n}: y
system services firewall policy delete -policy mgmt -service http -vserver <<clustername>>
```



It is normal for some of these commands to return an error message stating that the entry does not exist.

8. Change back to the normal administrator privilege level and allow SVM log access from the web.

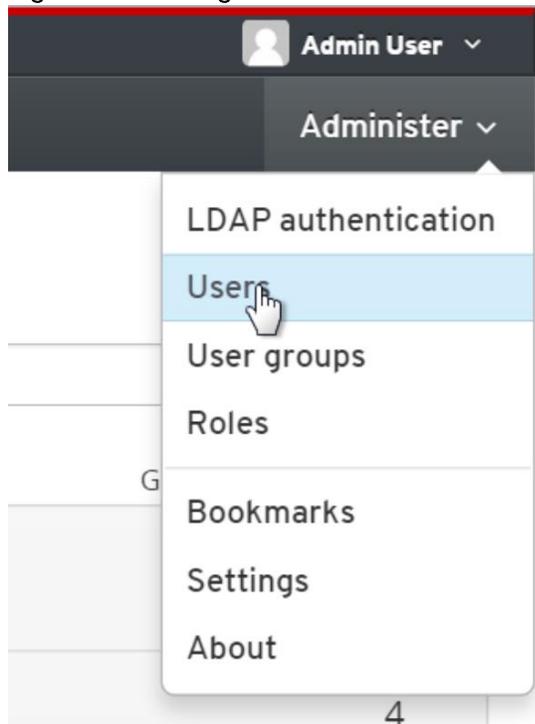
```
set -privilege admin
vserver services web modify -name spi|ontapi|compat -vserver * -enabled true
```

Appendix C: Changing Installer GUI Password

Log in to Red Hat Enterprise Linux OpenStack Platform installer GUI using the URL and admin credentials from the installation complete output. It is recommended to change the password by using Administrator menu on the right (Figure 231). To change the password, complete the following steps:

1. Click Users
2. Select admin user
3. In User tab, enter new password.
4. Verify new password and click Submit.

Figure 231 Change Password



Appendix D: Red Hat Enterprise Linux OpenStack Platform Installer Server iptables Configuration

```
[root@rhel-osp-installer ~]# cat /etc/sysconfig/iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [2044:3034903]
-A INPUT -p tcp -m multiport --ports 22 -m comment --comment "22 accept - ssh" -j
ACCEPT
-A INPUT -p tcp -m multiport --ports 443 -m comment --comment "443 accept -
apache" -j ACCEPT
-A INPUT -p tcp -m multiport --ports 53 -m comment --comment "53 accept - dns
tcp" -j ACCEPT
-A INPUT -p udp -m multiport --ports 53 -m comment --comment "53 accept - dns
udp" -j ACCEPT
-A INPUT -p udp -m multiport --ports 67 -m comment --comment "67 accept - dhcp" -
j ACCEPT
-A INPUT -p udp -m multiport --ports 68 -m comment --comment "68 accept - bootp"
-j ACCEPT
-A INPUT -p udp -m multiport --ports 69 -m comment --comment "69 accept - tftp" -
j ACCEPT
-A INPUT -p tcp -m multiport --ports 80 -m comment --comment "80 accept - apache"
-j ACCEPT
-A INPUT -p tcp -m multiport --ports 8140 -m comment --comment "8140 accept -
puppetmaster" -j ACCEPT
-A INPUT -p tcp -m multiport --ports 8080 -m comment --comment "8080 accept -
rh7repo" -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

```

-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -s 10.23.20.0/24 -j ACCEPT
-A FORWARD -d 10.23.20.0/24 -j ACCEPT
-A FORWARD -i enp6s0 -j ACCEPT
-A FORWARD -j ACCEPT
-A FORWARD ! -s 10.23.20.0/24 -j DROP
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Sun Jun 14 22:53:59 2015
# Generated by iptables-save v1.4.21 on Sun Jun 14 22:53:59 2015
*nat
:PREROUTING ACCEPT [30200:2232471]
:INPUT ACCEPT [8865:654638]
:OUTPUT ACCEPT [1437:96656]
:POSTROUTING ACCEPT [1034:67740]
-A POSTROUTING -o enp14s0.215 -j MASQUERADE
COMMIT

```

Appendix E: Kickstart default PXELinux Configuration

```

< %#
kind: PXELinux
name: Kickstart default PXELinux
oses:
- CentOS 4
- CentOS 5
- CentOS 6
- CentOS 7
- Fedora 16
- Fedora 17
- Fedora 18
- Fedora 19
- Fedora 20
- RedHat 4
- RedHat 5
- RedHat 6
- RedHat 7
%>
default linux
label linux
kernel <%= @kernel %>
<% if @host.operatingsystem.name == 'Fedora' and @host.operatingsystem.major.to_i >
16 -%>
append initrd=<%= @initrd %> ks=<%= foreman_url('provision')%> ks.device=bootif
network ks.sendmac
<% elsif @host.operatingsystem.name != 'Fedora' and @host.operatingsystem.major.to_i
>= 7 -%>
append initrd=<%= @initrd %> ks=<%= foreman_url('provision')%> network ks.sendmac
biosdevname=0 ip=ibft
<% else -%>
append initrd=<%= @initrd %> ks=<%= foreman_url('provision')%> ksdevice=bootif
network kssendmac
<% end -%>
IPAPPEND 2

```

Appendix F: Kickstart RHEL default Configuration

```

< %#
kind: provision

```

```

name: Kickstart RHEL default
oses:
- RedHat 4
- RedHat 5
- RedHat 6
- RedHat 7
%>
<%
  os_major = @host.operatingsystem.major.to_i
  # safemode renderer does not support unary negation
  pm_set = @host.puppetmaster.empty? ? false : true
  puppet_enabled = pm_set || @host.params['force-puppet']
%>
install
<%= @mediapath %>
lang en_US.UTF-8
selinux --enforcing
keyboard us
skipx

<% subnet = @host.subnet -%>
<% dhcp = subnet.dhcp_boot_mode? -%>
network --bootproto <%= dhcp ? 'dhcp' : "static --ip=#{@host.ip} --
netmask=#{subnet.mask} --gateway=#{subnet.gateway} --
nameserver=#{[subnet.dns_primary,
subnet.dns_secondary].select(&:present?).join(',')}" %> --device=<%= @host.mac -%> -
-hostname <%= @host %>

rootpw --iscrypted <%= root_pass %>
authconfig --useshadow --passalgo=sha256 --kickstart
timezone --utc <%= @host.params['time-zone'] || 'UTC' %>

<% if os_major >= 7 && @host.info["parameters"]["realm"] && @host.otp && @host.realm
-%>
realm join --one-time-password=<%= @host.otp %> <%= @host.realm %>
<% end -%>

<% if os_major > 4 -%>
services --enabled iptables --disabled
autofs,gpm,sendmail,cups,iptables,ip6tables,auditd,arptables_jf,xfs,pcmcia,isdn,rawd
evices,hpoj,bluetooth,openibd,avahi-daemon,avahi-
dnsconfd,hidd,hplip,pcscd,restorecond,mcstrans,rhnsd,yum-updatesd

<% if puppet_enabled && @host.params['enable-puppetlabs-repo'] &&
@host.params['enable-puppetlabs-repo'] == 'true' -%>
repo --name=puppetlabs-products --baseurl=http://yum.puppetlabs.com/el/<%=
@host.operatingsystem.major %>/products/<%= @host.architecture %>
repo --name=puppetlabs-deps --baseurl=http://yum.puppetlabs.com/el/<%=
@host.operatingsystem.major %>/dependencies/<%= @host.architecture %>
<% end -%>
<% end -%>

bootloader --location=mbr --append="nofb quiet splash=quiet" <%= grub_pass %>
<% if os_major == 5 -%>
key --skip
<% end -%>

%include /tmp/diskpart.cfg

```

```

text
reboot

%packages --ignoremissing
yum
dhclient
ntp
wget
@Core
iptables-services
-firewalld
<% if puppet_enabled %>
puppet
<% if @host.params['enable-puppetlabs-repo'] && @host.params['enable-puppetlabs-
repo'] == 'true' -%>
puppetlabs-release
<% end -%>
<% end -%>
%end

%pre
cat > /tmp/diskpart.cfg << EOF
<%= @host.diskLayout %>
EOF

# ensures a valid disk is addressed in the partition table layout
# sda is assumed and replaced if it is not correct
sed -i "s/sda/${(cat /proc/partitions | awk '{ print $4 }' | grep -e "^d.$" | grep -
vw fd0 | sort | head -1)}" /tmp/diskpart.cfg
%end

%post --nochroot
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on....
/usr/bin/chvt 3
(
cp -va /etc/resolv.conf /mnt/sysimage/etc/resolv.conf
/usr/bin/chvt 1
) 2>&1 | tee /mnt/sysimage/root/install.postnochroot.log
%end

%post
logger "Starting anaconda <%= @host %> postinstall"
exec < /dev/tty3 > /dev/tty3
#changing to VT 3 so that we can see whats going on....
/usr/bin/chvt 3
(
iscsiadm --mode discovery --type sendtargets --portal "${(<
/sys/firmware/ibft/target0/ip-addr)"; iscsiadm --mode node --login
<%= snippet 'kickstart_networking_setup' %>

#update local time
echo "updating system time"
/usr/sbin/ntpdate -sub <%= @host.params['ntp-server'] || '0.fedora.pool.ntp.org' %>
/usr/sbin/hwclock --systohc

#disable NetworkManager and enable network
chkconfig NetworkManager off
chkconfig network on

```

```

# setup SSH key for root user
<%= snippet 'ssh_public_key' %>

<%= snippet 'redhat_register' %>
<%= snippet 'custom_deployment_repositories' %>

<% if @host.info["parameters"]["realm"] && @host.otp && @host.realm &&
@host.realm.realm_type == "Red Hat Directory Server" && os_major <= 6 -%>
<%= snippet "freeipa_register" %>
<% end -%>

# update all the base packages from the updates repository
yum -t -y -e 0 update

# install sos related tools
yum -t -y -e 0 install sos sos-plugins-openstack rhos-log-collector

<% if puppet_enabled %>
# and add the puppet package
yum -t -y -e 0 install puppet

# we reuse our machine registerer instead
<%= snippet 'staypuft_client_bootstrap' %>

<% end -%>

sync
# Load Cisco enic
mount /boot 2>/dev/null | :
yum -t -y -e 0 localinstall http://rhel-osp-
installer.sjc.cisco.com:8080/CiscoEnic/kmod-enic-2.1.1.75-rhel7u1.el7.x86_64.rpm

# Inform the build system that we are done.
echo "Informing Foreman that we are built"
wget -q -O /dev/null --no-check-certificate <%= foreman_url %>
# Sleeping an hour for debug
) 2>&1 | tee /root/install.post.log
exit 0

%end

```

Appendix G: Full List of VIPs, Users, and Database Passwords

VIP List

```

ceilometer_admin_vip: 10.23.21.2

ceilometer_private_vip: 10.23.21.3

ceilometer_public_vip: 10.23.10.51

cinder_admin_vip: 10.23.21.4

```

cinder_private_vip:	10.23.21.5
cinder_public_vip:	10.23.10.52
db_vip:	10.23.21.6
glance_admin_vip:	10.23.21.7
glance_private_vip:	10.23.21.8
glance_public_vip:	10.23.10.53
heat_admin_vip:	10.23.21.9
heat_private_vip:	10.23.21.10
heat_public_vip:	10.23.10.54
heat_cfn_admin_vip:	10.23.21.11
heat_cfn_private_vip:	10.23.21.12
heat_cfn_public_vip:	10.23.10.55
horizon_admin_vip:	10.23.21.13
horizon_private_vip:	10.23.21.14
horizon_public_vip:	10.23.10.56
keystone_admin_vip:	10.23.21.15
keystone_private_vip:	10.23.21.16
keystone_public_vip:	10.23.10.57
loadbalancer_vip:	10.23.10.58
neutron_admin_vip:	10.23.21.17
neutron_private_vip:	10.23.21.18

neutron_public_vip:	10.23.10.59
nova_admin_vip:	10.23.21.19
nova_private_vip:	10.23.21.20
nova_public_vip:	10.23.10.60
amqp_vip:	10.23.21.21
swift_public_vip:	10.23.10.61

User Passwords

Admin:	f8401571fc03e988ffe83d074c28ba6f
Ceilometer :	aaa16716993b2864b24fb33d275027c4
Cinder :	99db55cbcb1e16b3092b348d1d9ef7be
Glance :	bda9e20035df4573134528f6944f3a96
Heat :	938e0bd5fc356b88d37c77664d0072d2
Heat cfn :	e2917b724107ba0d43a6562e9cd67a56
Keystone :	9c1fea3c53b348b83f033099db9f3d67
Neutron :	5b1d39ed7b6d112e123c85541caeee19
Nova :	83fa3c383cd809e55870aaff2dc738c
Swift :	b4e3f46b8a16ac0738dc1d1a83ce9b7a
Amqp:	72d11246e2bb472d08123fd793161233

Database Passwords

Cinder :	a45e3e51259feaa619eb6c5d408c2305
----------	----------------------------------

Glance : 31e058416f4390d9eefdfda8fe99f181

Heat : b7e6e9c263bf186723d3a01c404c628a

Mysql root: 9a38f10a992865af551d4950c6e82888

Keystone : 029a81c9af6796545c8f735f67acac33

Neutron : 465fdab7aa827a902c6c2ec9db78f6e7

Nova : 5b96937a41614073f57b7780942d2ecd

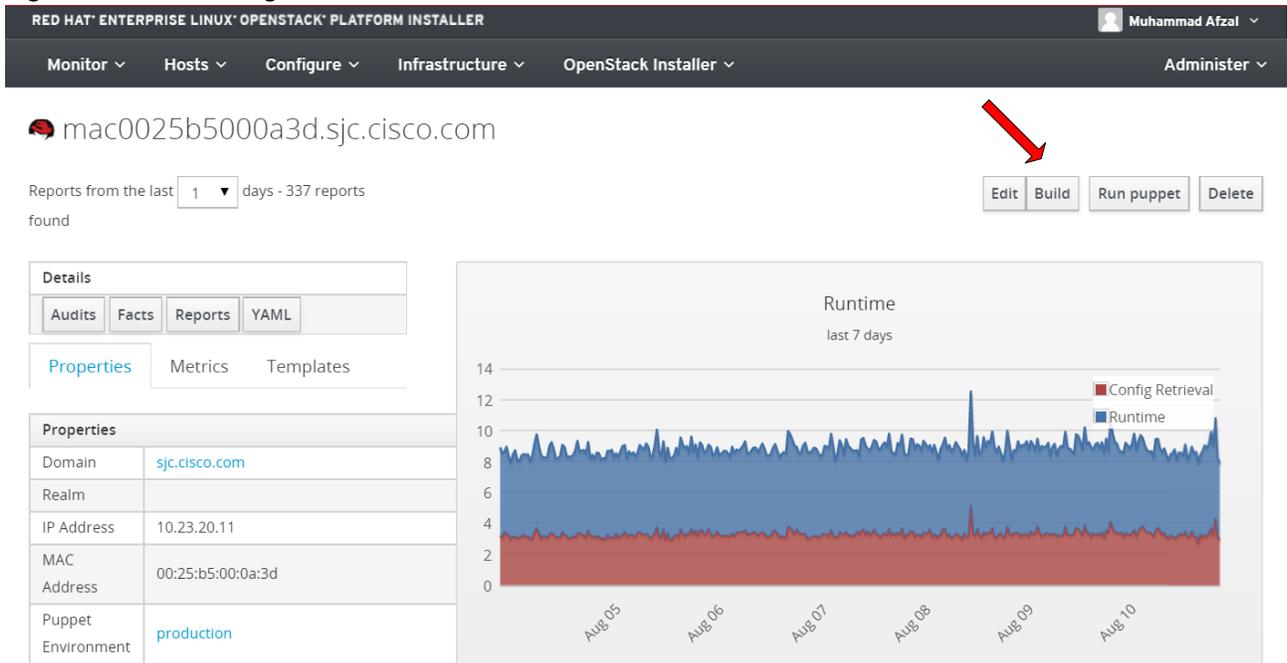
Hacluster: 24e5245d2aa4c4f80e8991996e17c763

Appendix H: Rebuilding a Server

It is not required to redeploy the entire set of servers all over again, if a specific server(s) need to be rebuilt for any specific reason. Server(s) can be easily rebuilt by completing the following step:

1. On the installer GUI, select the host → Build (Figure 232)

Figure 232 Rebuilding the Host



2. Reboot the server.
3. The RHEL-OSP Installer will rebuild the server on next PXE boot.

Appendix I: Restart Deployment

If the deployment fails and cannot be fixed easily, it can be discarded and a new deployment can be restarted. However, these nodes must have been already discovered earlier with their mac addresses, hostnames, leased IPs already present in the Red Hat Enterprise Linux OpenStack Platform Installer config/lease files. They must be first flushed out before restarting a new deployment. To restart a deployment, complete the following steps:

1. Select OpenStack Installer → Deployments
2. Click Delete for the failed deployment (Figure 233)

Figure 233 Delete Deployment

The screenshot shows the 'RED HAT ENTERPRISE LINUX OPENSTACK PLATFORM INSTALLER' interface. The top navigation bar includes 'Monitor', 'Hosts', 'Configure', 'Infrastructure', 'OpenStack Installer', and 'Administer'. The 'OpenStack Installer' section is active, displaying 'OpenStack Deployments'. There is a search bar and a 'New Deployment' button. A table lists one deployment: 'FLEXPOD_RHEL_OSP6' with a 'Delete' button. A 'Displaying 1 entry' indicator is at the bottom.

3. Delete the hosts. Select Hosts → All hosts → Select All Hosts → Select Action → Delete Hosts. This should delete the files associated for each host (the file is named after the PXE mac address) from `/var/lib/tftpboot/pxelinux.cfg`



If the file for any of the host still exists, delete them manually in the installer node.

4. Log into the RHEL-OSP Installer through SSH and perform the following:
 - a. Stop relevant services.


```
# for i in dhcpd named foreman-proxy; do systemctl stop $i.service; done
```
 - b. Remove the following files (Figure 234)


```
# rm -rf /var/named/dynamic/db.20.23.10.in-addr.arpa.jnl
# rm -rf /var/named/dynamic/db.sjc.cisco.com.jnl
```

Figure 234 Remove .jnl Files

```
-rw-r--r--. 1 named named 729 Aug 4 10:21 db.20.23.10.in-addr.arpa
-rw-r--r--. 1 named named 71258 Aug 4 10:09 db.20.23.10.in-addr.arpa.jnl
-rw-r--r--. 1 named named 678 Aug 4 10:21 db.sjc.cisco.com
-rw-r--r--. 1 named named 59519 Aug 4 10:09 db.sjc.cisco.com.jnl
[root@rhel-osp-installer dynamic]#
```

- c. Zero out the entry from the dhcp leases file.

- ```
echo "" > /var/lib/dhcpd/dhcpd.leases
```
- d. Restart the services.
- ```
# for i in dhcpd named foreman-proxy; do systemctl restart $i.service; done
```
- e. Reboot all the nodes so that they get PXE booted and discovered by the installer.
 - f. Verify the discovered hosts by clicking Hosts → Discovered hosts in the installer GUI.
 - g. Create a new deployment.

Appendix J: Red Hat Enterprise Linux OpenStack Platform Installer Server Interfaces Configuration

```
enp6s0

###
### File managed by Puppet
###
DEVICE=enp6s0
BOOTPROTO=none
HWADDR=00:25:b5:00:0a:6e
ONBOOT=yes
HOTPLUG=yes
TYPE=Ethernet
IPADDR=10.23.20.2
NETMASK=255.255.255.0
PEERDNS=yes
DNS1=10.23.20.2
DNS2=<<var_nameserver_ip>>
NM_CONTROLLED=no

enp13s0

TYPE=Ethernet
BOOTPROTO=none
```

```
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=enp13s0
UUID=70f203f5-3427-4fff-99d0-3d32c46106ae
DEVICE=enp13s0
ONBOOT=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
```

enp13s0.10

```
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
DEVICE=enp13s0.10
VLAN=yes
ONBOOT=yes
IPADDR=10.23.10.36
PREFIX=24
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
```

enp14s0

```
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=enp14s0
UUID=c961e697-b484-4573-a57d-3b8be7b4d095
DEVICE=enp14s0
ONBOOT=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
```

```
enp14s0.215
```

```
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
DEVICE=enp14s0.215
ONBOOT=yes
IPADDR=<<osp_installer_external_ip>>
PREFIX=24
GATEWAY=<<osp_installer_external_gateway>>
DNS1=<<var_nameserver_ip>>
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
```

IPV6_PRIVACY=no

VLAN=yes

About the Authors

Muhammad Afzal, Architect Engineering, Cisco UCS Data Center Solutions Engineering, Cisco Systems, Inc.

Muhammad Afzal is an Engineering Architect at Cisco Systems in Cisco UCS Data Center Solution Engineering. He is currently responsible for producing and designing validated converged architectures while working collaboratively with product partners. Previously, Afzal had been a lead architect for various cloud and data center solutions including UCS in Solution Development Unit at Cisco. Prior to this, Afzal has been a Solutions Architect in **Cisco's Advanced Services** group, where he worked closely with Cisco's large enterprise and service provider customers delivering data center and cloud solutions. Afzal holds an MBA in finance and a BS in computer engineering.

Dave Cain, Reference Architect, Converged Infrastructure Engineering, NetApp

Dave Cain is a **Reference Architect and Technical Marketing Engineer with NetApp's Converged Infrastructure** Engineering organization. He focuses on producing validated reference architectures that promote the benefits of NetApp storage and software into datacenter and cloud environments. Prior to joining NetApp, he spent 10 years in various roles at IBM focused on Network, Storage, and Virtualization IT infrastructure. Dave holds a Bachelor of Science degree in Computer Science from North Carolina State University. He coauthored Five IBM Redbooks publications, and holds two US patents and various invention disclosures in the computer networking field.

Tushar Katarki, Integration Architect, Red Hat

Tushar Katarki is a technology professional with experience in datacenter, cloud and storage technologies. He is currently an Integration Architect at Red Hat driving cross product architecture and integration across Red Hat and partner products. Prior to the current role, Tushar has been a product manager and a developer at Red Hat, Oracle (Sun Microsystems), Polycom, Sycamore Networks and Percona. Tushar has an MBA from Babson College and MS in Computer Science from University at Buffalo.

Acknowledgements

For their support in developing this Cisco Validated Design, the authors would like to acknowledge:

- Vijay Durairaj, Steven Hillman - Cisco Systems, Inc.
- Jeff Applewhite, Rob Bradbury, Justin Parisi - NetApp
- Mike Burns and Steven Reichard - Red Hat