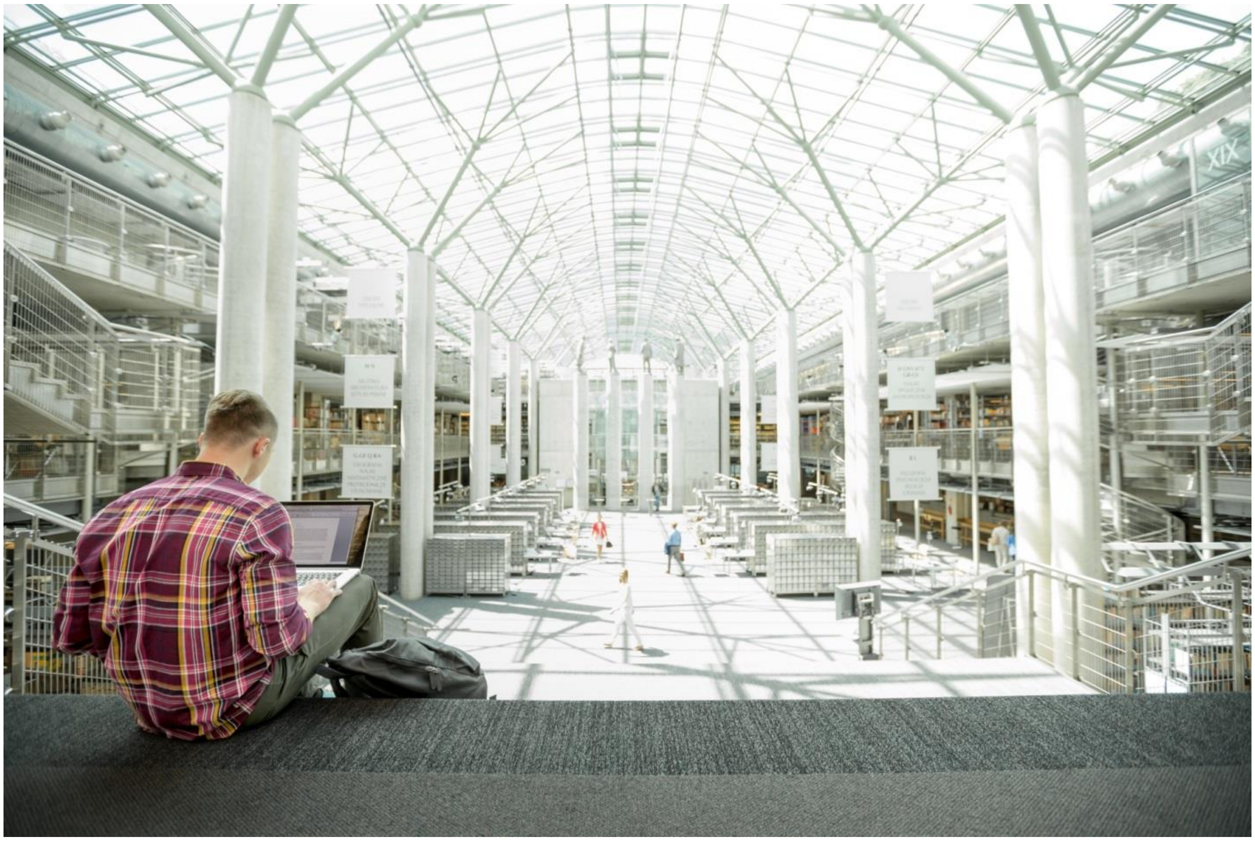


Cisco UCS S3260 Storage Server with Red Hat Ceph Storage

Design and Deployment of Red Hat Ceph Storage 2.1 on Cisco UCS S3260 Storage Server

Last Updated: March 14, 2017

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.

Table of Contents

Executive Summary 7

Solution Overview 8

Introduction 8

Solution 8

Audience 8

Solution Summary 9

Technology Overview 10

Cisco Unified Computing System 10

Cisco UCS S3260 Storage Server 11

Cisco UCS C220 M4 Rack Server 12

Cisco UCS Virtual Interface Card 1387 12

Cisco UCS 6300 Series Fabric Interconnect 13

Cisco Nexus 9332PQ Switch 14

Cisco UCS Manager 14

Red Hat Enterprise Linux 7.3 15

Red Hat Ceph Storage 16

Solution Design 17

Solution Overview 17

Design Principles of Red Hat Ceph Storage on Cisco UCS 17

Deploying Red Hat Ceph Storage on Cisco UCS 19

Operational Guide for Red Hat Ceph Storage on Cisco UCS 20

Requirements 22

Rack and PDU configuration 22

Physical Topology and Configuration 24

Software Distributions and Versions 28

Deployment of Hardware and Software 30

Fabric Configuration 30

Initial Setup of Cisco UCS 6332 Fabric Interconnects 30

Configure Fabric Interconnect A 30

Configure Fabric Interconnect B 33

Logging Into Cisco UCS Manager 34

Configure NTP Server 35

Initial Base Setup of the Environment 36

Configure Global Policies 36

Enable Fabric Interconnect A Ports for Server 37

Enable Fabric Interconnect A Ports for Uplinks 38

Label Each Chassis for Identification 38

Label Each Server for Identification 39

Create KVM IP Pool 40

Create MAC Pool 41

Create UUID Pool 42
Create Server Pool 44
Create VLANs........ 46
Enable CDP 47
QoS System Class 48
QoS Policy Setup 49
vNIC Template Setup 49
Adapter Policy Setup 51
BIOS Policy Setup 52
Boot Policy Setup 55
Create Maintenance Policy Setup 56
Create Power Control Policy Setup 57
Create Disk Scrub Policy 58
Creating Chassis Profile 59
Create Chassis Firmware Package 59
Create Chassis Maintenance Policy 60
Create Disk Zoning Policy 61
Create Chassis Profile Template 63
Create Chassis Profile from Template 65
Associate Chassis Profile 66
Creating Storage Profiles 67
Setting Disks for Rack-Mount Servers to Unconfigured-Good 67
Setting Disks for Cisco UCS S3260 Storage Server to Unconfigured-Good with Cisco UCS PowerTool 68
Create Storage Profile for Cisco UCS S3260 Storage Server 69
Creating Disk Group Policies and RAID 0 LUNs for top-loaded Cisco UCS S3260 Storage Server HDDs 71
Create Storage Profile for Cisco UCS C220 M4S Rack-Mount Server 72
Creating a Service Profile Template 74
Create Service Profile Template for Cisco UCS S3260 Storage Server Top and Bottom Node 74
Create Service Profile Template for Cisco UCS C220 M4S 82
Create Service Profiles from Template 86
Creating Port Channel for Uplinks 87
Create Port Channel for Fabric Interconnect A/B 87
Create VLAN Groups for Port Channel 89
Configure Scheduled Backup 90
Configuration of Nexus C9332PQ Switch A and B 91
Initial Setup of Nexus C9332PQ Switch A and B 92
Enable Features on Cisco Nexus 9332PQ Switch A and B 94
Configuring VLANs on Nexus 9332PQ Switch A and B 95
Verification check of Nexus C9332PQ Configuration for Switch A and B 104
Installation of Red Hat Enterprise Linux 7.3 Operating System 108
Installation of RHEL 7.3 on Cisco UCS C220 M4S 108
Installation of RHEL 7.3 on Cisco UCS S3260 Storage Server 111

Post-Installation Steps for Red Hat Enterprise Linux 7.3 113
Preparation of All Nodes for Ceph Installation 117
Step 1 – Update all Ceph Nodes 117
Step 2 – Configuring Firewall 117
Step 3 – Configuring Network Time Protocol 118
Step 4 – Create an Ansible User 119
Step 5 – Enabling Password-Less SSH 119
Step 6 – Check Disk Devices and Prepare Disks for Ceph 120
Red Hat Ceph Storage Installation via Ansible 121
Configure Ceph Global Settings 122
Configure Ceph OSD Settings 124
Deploy Red Hat Ceph Storage via Ansible 126
Final Check of Ceph Deployment 126
Operational Guide to Extend a Ceph Cluster with Cisco UCS 128
Adding Cisco UCS S3260 as Ceph OSD Nodes 128
Enable Fabric Interconnect Ports for Server 128
Label Chassis for Identification 128
Label each Server for Identification 128
Add Ceph OSD Nodes to Server Pool 129
Create Chassis Profile from Template 129
Associate Chassis Profile 129
Setting Disks for Cisco UCS S3260 Storage Server to Unconfigured-Good with Cisco UCS PowerTool 130
Create Service Profiles from Template 130
Installation of RHEL 7.3 on Cisco UCS S3260 Storage Server 130
Configure /etc/hosts and Enable Password-Less Login 132
Configure Hostnames 133
Copy Local Repository File and Enable Repositories 133
Install Latest Network Driver 133
Create VLAN Interface for Network Public on Both Ceph OSD Nodes 133
Update All Ceph OSD Nodes 133
Configure the Firewall 134
Configure Network Time Protocol 134
Copy sudoers File to Ceph RGW Nodes and Enable Password-Less SSH 134
Change .ssh/config File for cephadm User 134
Check Disk Devices and Prepare Disks for Ceph 135
Change Ansible Hosts File 136
Deploy Red Hat Ceph Storage via Ansible 136
Final Check of Ceph Deployment 137
Add RADOS Gateway for Object Storage 138
Enable Fabric Interconnect Ports for Server 138
Label Each Server for Identification 138
Add Ceph RGW Nodes to Server Pool 139

Setting Disks for Rack-Mount Servers to Unconfigured-Good 139
Create Service Profiles from Template 139
Install RHEL 7.3 on Cisco UCS C220 M4S 139
Configure /etc/hosts and Enable Password-Less Login 142
Configure Hostnames 142
Copy Local Repository File and Enable Repositories 142
Install Latest Network Driver 143
Create VLAN Interface for Network Public on All Ceph RGW Nodes 143
Update All Ceph RGW Nodes 143
Configure the Firewall 143
Configure Network Time Protocol 143
Copy sudoers File to Ceph RGW Nodes and Enable Password-Less SSH 144
Change .ssh/config File for cephadm User 144
Change Ansible Hosts File 144
Deploy Red Hat Ceph Storage via Ansible 145
Final Check of Ceph Deployment 145
Bill of Materials 147
Appendix 149
Appendix A – Kickstart File for Cisco UCS C220 M4S 149
Appendix B – Kickstart File for Cisco UCS S3260 Storage Server 149
Appendix C – Example /etc/hosts File 150
Appendix D – /home/cephadm/.ssh/config File from Ansible Administration Node cephadm 152
Appendix E - /usr/share/ceph-ansible/group_vars/all Configuration File 153
Appendix F - /usr/share/ceph-ansible/group_vars/osds Configuration File 165
About the Authors 172
Acknowledgements 172

Executive Summary

Modern data centers increasingly rely on a variety of architectures for storage. Whereas in the past organizations focused on traditional storage only, today organizations are focusing on Software Defined Storage for several reasons:

- Software Defined Storage offers unlimited scalability and simple management.
- Because of the low cost per gigabyte, Software Defined Storage is well suited for large-capacity needs, and therefore for use cases such as archive, backup, and cloud operations.
- Software Defined Storage allows the use of commodity hardware.

Enterprise storage systems are designed to address business-critical requirements in the data center. But these solutions may not be optimal for use cases such as backup and archive workloads and other unstructured data, for which OLTP-style data latency is not especially important.

Red Hat Ceph Storage is an example of a massively scalable, Open Source, software-defined storage system that gives you unified storage for cloud environments. It is an object storage architecture, that can easily achieve enterprise-class reliability, scale-out capacity, and lower costs with an industry-standard server solution.

The Cisco UCS S3260 Storage Server, originally designed for the data center, together with Red Hat Ceph Storage is optimized for Software Defined Storage solutions, making it an excellent fit for unstructured data workloads such as backup, archive, and cloud data. The S3260 delivers a complete hardware with exceptional scalability for computing and storage resources together with 40 Gigabit Ethernet networking. The S3260 is the platform of choice for Software Defined Storage solutions because it provides more than comparable platforms:

- Proven server architecture that allows you to upgrade individual components without the need for migration.
- High-bandwidth networking that meets the needs of large-scale object storage solutions like Red Hat Ceph Storage.
- Unified, embedded management for an easy-to-scale infrastructure.
- API access for cloud-scale applications.

Cisco and Red Hat are collaborating to offer customers a scalable Software Defined Storage solution for unstructured data that is integrated with Red Hat Ceph Storage. With the power of the Cisco UCS management framework, the solution is cost effective to deploy and manage and will enable the next-generation cloud deployments that drive business agility, lower operational costs and avoid vendor lock-in.

Solution Overview

Introduction

Traditional storage systems are limited in their ability to easily and cost-effectively scale to support massive amounts of unstructured data. With about 80 percent of data being unstructured, new approaches using x86 servers are proving to be more cost effective, providing storage that can be expanded as easily as your data grows. Software Defined Storage is a scalable and cost-effective approach for handling massive amounts of data.

Red Hat Ceph Storage is a massively scalable, open source, software-defined storage system that supports unified storage for a cloud environment. With object and block storage in one platform, Red Hat Ceph Storage efficiently and automatically manages the petabytes of data needed to run businesses facing massive data growth. It is proven at web scale and has many deployments in production environments as an object store for large, global corporations. Red Hat Ceph Storage was designed from the ground up for web-scale block and object storage and cloud infrastructures.

Scale-out storage uses x86 architecture storage-optimized servers to increase performance while reducing costs. The Cisco UCS S3260 Storage Server is well suited for scale-out storage solutions. It provides a platform that is cost effective to deploy and manage using the power of the Cisco Unified Computing System (Cisco UCS) management: capabilities that traditional unmanaged and agent-based management systems can’t offer. You can design S3260 solutions for a computing-intensive, capacity-intensive, or throughput-intensive workload.

Both solutions together, Red Hat Ceph Storage and Cisco UCS S3260 Storage Server, deliver a simple, fast and scalable architecture for enterprise scale-out storage.

Solution

The current Cisco Validated Design (CVD) is a simple and linearly scalable architecture that provides Software Defined Storage for block and object on Red Hat Ceph Storage 2.1 and Cisco UCS S3260 Storage Server. The solution includes the following features:

- Infrastructure for large scale-out storage.
- Design of a Red Hat Ceph Storage solution together with Cisco UCS S3260 Storage Server.
- Simplified infrastructure management with Cisco UCS Manager.
- Architectural scalability – linear scaling based on network, storage, and compute requirements.
- Operational guide to extend a working Red Hat Ceph cluster with Ceph RADOS Gateway (RGW) and Ceph OSD nodes.

Audience

This document describes the architecture, design and deployment procedures of a Red Hat Ceph Storage solution using six Cisco UCS S3260 Storage Servers with two C3x60 M4 server nodes each as OSD nodes, three Cisco UCS C220 M4 S rack server each as Monitor nodes, three Cisco UCS C220 M4S rackserver each as RGW node, one Cisco UCS C220 M4S rackserver as Admin node, and two Cisco UCS 6332 Fabric Interconnect managed by Cisco UCS Manager. The intended audience for this document includes, but is not limited to, sales engineers, field consultants, professional services, IT managers, partner engineering, and customers who want to deploy Red Hat Ceph Storage on the Cisco Unified Computing System (UCS) using Cisco UCS S3260 Storage Servers.

Solution Summary

This CVD describes in detail the process of deploying Red Hat Ceph Storage 2.1 on Cisco UCS S3260 Storage Server.

The configuration uses the following architecture for the deployment:

- 6 x Cisco UCS S3260 Storage Servers, each with 2 x C3x60 M4 server nodes working as Ceph OSD nodes
- 3 x Cisco UCS C220 M4S rack server working as Ceph Monitor nodes
- 3 x Cisco UCS C220 M4S rack server working as Ceph RADOS gateway nodes
- 1 x Cisco UCS C220 M4S rack server working as Ceph Admin node
- 2 x Cisco UCS 6332 Fabric Interconnect
- 1 x Cisco UCS Manager
- 2 x Cisco Nexus 9332PQ Switches

The solution has various options to scale performance and capacity. A base capacity summary is shown in Table 1 .

Table 1 Usable capacity options for tested Cisco Validated Design

HDD Type	Number of Disks	Data Protection 3 x Replication	Data Protection Erasure Coding 4:2
4 TB 7200-rpm LFF SAS drives	288	384 TB	760 TB
6 TB 7200-rpm LFF SAS drives[1]	288	576 TB	1140 TB
8 TB 7200-rpm LFF SAS drives	288	768 TB	1520 TB
10 TB 7200-rpm LFF SAS drives	288	960 TB	1900 TB

[1] Tested configuration

Technology Overview

Cisco Unified Computing System

The Cisco Unified Computing System (Cisco UCS) is a state-of-the-art data center platform that unites computing, network, storage access, and virtualization into a single cohesive system.

The main components of Cisco Unified Computing System are:

- Computing - The system is based on an entirely new class of computing system that incorporates rack-mount and blade servers based on Intel Xeon Processor E5 and E7. The Cisco UCS servers offer the patented Cisco Extended Memory Technology to support applications with large datasets and allow more virtual machines (VM) per server.
- 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 (NFS or iSCSI), Fibre Channel, and Fibre Channel over Ethernet (FCoE). This provides customers with choice for storage access and investment protection. In addition, the server administrators can pre-assign storage-access policies for system connectivity to storage resources, simplifying storage connectivity, and management for increased productivity.

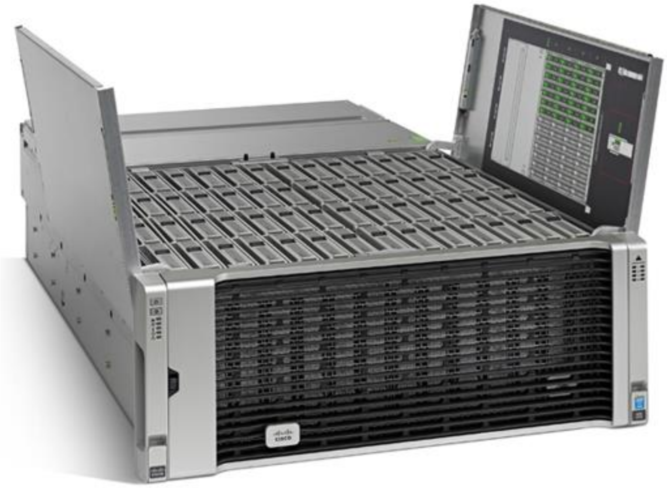
The Cisco Unified Computing System is designed to deliver:

- A reduced Total Cost of Ownership (TCO) and increased business agility.
- Increased IT staff productivity through just-in-time provisioning and mobility support.
- A cohesive, integrated system which unifies the technology in the data center.
- Industry standards supported by a partner ecosystem of industry leaders.

Cisco UCS S3260 Storage Server

The Cisco UCS® S3260 Storage Server (Figure 1) is a modular, high-density, high-availability dual node rack server well suited for service providers, enterprises, and industry-specific environments. It addresses the need for dense cost effective storage for the ever-growing data needs. Designed for a new class of cloud-scale applications, it is simple to deploy and excellent for big data applications, software-defined storage environments such as Ceph and other unstructured data repositories, media streaming, and content distribution.

Figure 1 Cisco UCS S3260 Storage Server



Extending the capability of the Cisco UCS C3000 portfolio, the Cisco UCS S3260 helps you achieve the highest levels of data availability. With dual-node capability that is based on the Intel® Xeon® processor E5-2600 v4 series, it features up to 600 TB of local storage in a compact 4-rack-unit (4RU) form factor. All hard-disk drives can be asymmetrically split between the dual-nodes and are individually hot-swappable. The drives can be built-in in an enterprise-class Redundant Array of Independent Disks (RAID) redundancy or be in a pass-through mode.

This high-density rack server comfortably fits in a standard 32-inch depth rack, such as the Cisco® R42610 Rack.

The Cisco UCS S3260 is deployed as a standalone server in both bare-metal or virtualized environments. Its modular architecture reduces total cost of ownership (TCO) by allowing you to upgrade individual components over time and as use cases evolve, without having to replace the entire system.

The Cisco UCS S3260 uses a modular server architecture that, using Cisco's blade technology expertise, allows you to upgrade the computing or network nodes in the system without the need to migrate data migration from one system to another. It delivers:

- Dual server nodes
- Up to 36 computing cores per server node
- Up to 60 drives mixing a large form factor (LFF) with up to 28 solid-state disk (SSD) drives plus 2 SSD SATA boot drives per server node
- Up to 512 GB of memory per server node (1 terabyte [TB] total)
- Support for 12-Gbps serial-attached SCSI (SAS) drives
- A system I/O Controller with Cisco VIC 1300 Series Embedded Chip supporting Dual-port 40Gbps

- High reliability, availability, and serviceability (RAS) features with tool-free server nodes, system I/O controller, easy-to-use latching lid, and hot-swappable and hot-pluggable components

Cisco UCS C220 M4 Rack Server

The Cisco UCS® C220 M4 Rack Server (Figure 2) is the most versatile, general-purpose enterprise infrastructure and application server in the industry. It is a high-density two-socket enterprise-class rack server that delivers industry-leading performance and efficiency for a wide range of enterprise workloads, including virtualization, collaboration, and bare-metal applications. The Cisco UCS C-Series Rack Servers can be deployed as standalone servers or as part of the Cisco Unified Computing System™ (Cisco UCS) to take advantage of Cisco's standards-based unified computing innovations that help reduce customers' total cost of ownership (TCO) and increase their business agility.

Figure 2 Cisco UCS C220 M4 Rack Server

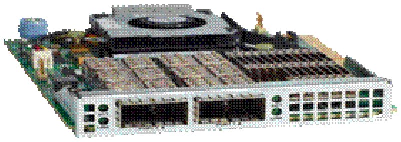
The enterprise-class Cisco UCS C220 M4 server extends the capabilities of the Cisco UCS portfolio in a 1RU form factor. It incorporates the Intel® Xeon® processor E5-2600 v4 and v3 product family, next-generation DDR4 memory, and 12-Gbps SAS throughput, delivering significant performance and efficiency gains. The Cisco UCS C220 M4 rack server delivers outstanding levels of expandability and performance in a compact 1RU package:

- Up to 24 DDR4 DIMMs for improved performance and lower power consumption
- Up to 8 Small Form-Factor (SFF) drives or up to 4 Large Form-Factor (LFF) drives
- Support for 12-Gbps SAS Module RAID controller in a dedicated slot, leaving the remaining two PCIe Gen 3.0 slots available for other expansion cards
- A modular LAN-on-motherboard (mLOM) slot that can be used to install a Cisco UCS virtual interface card (VIC) or third-party network interface card (NIC) without consuming a PCIe slot
- Two embedded 1Gigabit Ethernet LAN-on-motherboard (LOM) ports

Cisco UCS Virtual Interface Card 1387

The Cisco UCS Virtual Interface Card (VIC) 1387 (Figure 3) is a Cisco® innovation. It provides a policy-based, stateless, agile server infrastructure for your data center. This dual-port Enhanced Quad Small Form-Factor Pluggable (QSFP) half-height PCI Express (PCIe) modular LAN-on-motherboard (mLOM) adapter is designed exclusively for Cisco UCS C-Series and S3260 Rack Servers. The card supports 40 Gigabit Ethernet and Fibre Channel over Ethernet (FCoE). It incorporates Cisco's next-generation converged network adapter (CNA) technology and offers a comprehensive feature set, providing investment protection for future feature software releases. The card can present more than 256 PCIe standards-compliant interfaces to the host, and these can be dynamically configured as either network interface cards (NICs) or host bus adapters (HBAs). In addition, the VIC supports Cisco Data Center Virtual Machine Fabric Extender (VM-FEX) technology. This technology extends the Cisco UCS fabric interconnect ports to virtual machines, simplifying server virtualization deployment.

Figure 3 Cisco UCS Virtual Interface Card 1387



The Cisco UCS VIC 1387 provides the following features and benefits:

- Stateless and agile platform: The personality of the card is determined dynamically at boot time using the service profile associated with the server. The number, type (NIC or HBA), identity (MAC address and World Wide Name [WWN]), failover policy, bandwidth, and quality-of-service (QoS) policies of the PCIe interfaces are all determined using the service profile. The capability to define, create, and use interfaces on demand provides a stateless and agile server infrastructure

- Network interface virtualization: Each PCIe interface created on the VIC is associated with an interface on the Cisco UCS fabric interconnect, providing complete network separation for each virtual cable between a PCIe device on the VIC and the interface on the fabric interconnect

Cisco UCS 6300 Series Fabric Interconnect

The Cisco UCS 6300 Series Fabric Interconnects are a core part of Cisco UCS, providing both network connectivity and management capabilities for the system (Figure 4). The Cisco UCS 6300 Series offers line-rate, low-latency, lossless 10 and 40 Gigabit Ethernet, Fibre Channel over Ethernet (FCoE), and Fibre Channel functions.

Figure 4 Cisco UCS 6300 Series Fabric Interconnect

The Cisco UCS 6300 Series provides the management and communication backbone for the Cisco UCS B-Series Blade Servers, 5100 Series Blade Server Chassis, and C-Series Rack Servers managed by Cisco UCS. All servers attached to the fabric interconnects become part of a single, highly available management domain. In addition, by supporting unified fabric, the Cisco UCS 6300 Series provides both LAN and SAN connectivity for all servers within its domain.

From a networking perspective, the Cisco UCS 6300 Series uses a cut-through architecture, supporting deterministic, low-latency, line-rate 10 and 40 Gigabit Ethernet ports, switching capacity of 2.56 terabits per second (Tbps), and 320 Gbps of bandwidth per chassis, independent of packet size and enabled services. The product family supports Cisco® low-latency, lossless 10 and 40 Gigabit Ethernet unified network fabric capabilities, which increase the reliability, efficiency, and scalability of Ethernet networks. The fabric interconnect supports multiple traffic classes over a lossless Ethernet fabric from the server through the fabric interconnect. Significant TCO savings can be achieved with an FCoE optimized server design in which network interface cards (NICs), host bus adapters (HBAs), cables, and switches can be consolidated.

The Cisco UCS 6332 32-Port Fabric Interconnect is a 1-rack-unit (1RU) Gigabit Ethernet, and FCoE switch offering up to 2.56 Tbps throughput and up to 32 ports. The switch has 32 fixed 40-Gbps Ethernet and FCoE ports.

Both the Cisco UCS 6332UP 32-Port Fabric Interconnect and the Cisco UCS 6332 16-UP 40-Port Fabric Interconnect have ports that can be configured for the breakout feature that supports connectivity between 40 Gigabit Ethernet ports and 10 Gigabit Ethernet ports. This feature provides backward compatibility to existing hardware that supports 10 Gigabit Ethernet. A 40 Gigabit Ethernet port can be used as four 10 Gigabit Ethernet ports. Using a 40 Gigabit Ethernet SFP, these ports on a Cisco UCS 6300 Series Fabric Interconnect can connect to another fabric interconnect that has four 10 Gigabit Ethernet SFPs. The breakout feature can be configured on ports 1 to 12 and ports 15 to 26 on the Cisco UCS 6332UP fabric interconnect. Ports 17 to 34 on the Cisco UCS 6332 16-UP fabric interconnect support the breakout feature.

Cisco Nexus 9332PQ Switch

The Cisco Nexus® 9000 Series Switches (Figure 5) include both modular and fixed-port switches that are designed to overcome these challenges with a flexible, agile, low-cost, application-centric infrastructure.

Figure 5 Cisco Nexus 9332PQ Switch

The Cisco Nexus 9300 platform consists of fixed-port switches designed for top-of-rack (ToR) and middle-of-row (MoR) deployment in data centers that support enterprise applications, service provider hosting, and cloud computing environments. They are Layer 2 and 3 nonblocking 10 and 40 Gigabit Ethernet switches with up to 2.56 terabits per second (Tbps) of internal bandwidth.

The Cisco Nexus 9332PQ Switch is a 1-rack-unit (1RU) switch that supports 2.56 Tbps of bandwidth and over 720 million packets per second (mpps) across thirty-two 40-Gbps Enhanced QSFP+ ports

All the Cisco Nexus 9300 platform switches use dual- core 2.5-GHz x86 CPUs with 64-GB solid-state disk (SSD) drives and 16 GB of memory for enhanced network performance.

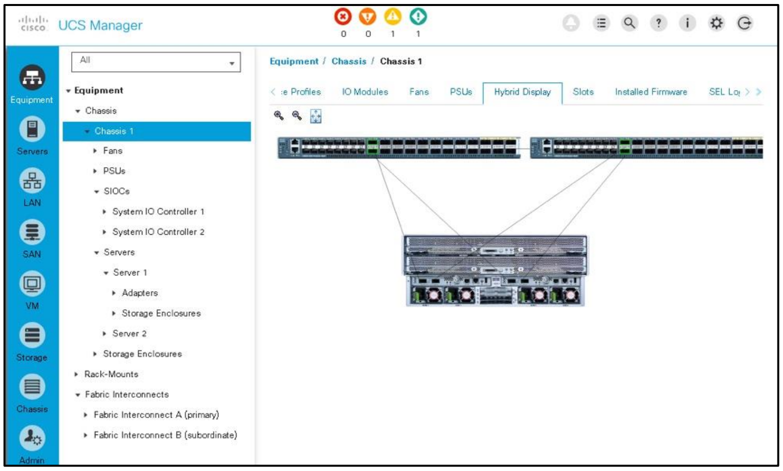
With the Cisco Nexus 9000 Series, organizations can quickly and easily upgrade existing data centers to carry 40 Gigabit Ethernet to the aggregation layer or to the spine (in a leaf-and-spine configuration) through advanced and cost-effective optics that enable the use of existing 10 Gigabit Ethernet fiber (a pair of multimode fiber strands).

Cisco provides two modes of operation for the Cisco Nexus 9000 Series. Organizations can use Cisco® NX-OS Software to deploy the Cisco Nexus 9000 Series in standard Cisco Nexus switch environments. Organizations also can use a hardware infrastructure that is ready to support Cisco Application Centric Infrastructure (Cisco ACI™) to take full advantage of an automated, policy-based, systems management approach.

Cisco UCS Manager

Cisco UCS® Manager (Figure 6) provides unified, embedded management of all software and hardware components of the Cisco Unified Computing System™ (Cisco UCS) across multiple chassis, rack servers and thousands of virtual machines. It supports all Cisco UCS product models, including Cisco UCS B-Series Blade Servers, C-Series Rack Servers, and M-Series composable infrastructure and Cisco UCS Mini, as well as the associated storage resources and networks. Cisco UCS Manager is embedded on a pair of Cisco UCS 6300 or 6200 Series Fabric Interconnects using a clustered, active-standby configuration for high availability. The manager participates in server provisioning, device discovery, inventory, configuration, diagnostics, monitoring, fault detection, auditing, and statistics collection.

Figure 6 Cisco UCS Manager



An instance of Cisco UCS Manager with all Cisco UCS components managed by it forms a Cisco UCS domain, which can include up to 160 servers. In addition to provisioning Cisco UCS resources, this infrastructure management software provides a model-based foundation for streamlining the day-to-day processes of updating, monitoring, and managing computing resources, local storage, storage connections, and network connections. By enabling better automation of processes, Cisco UCS Manager allows IT organizations to achieve greater agility and scale in their infrastructure operations while reducing complexity and risk. The manager provides flexible role- and policy-based management using service profiles and templates.

Cisco UCS Manager manages Cisco UCS systems through an intuitive HTML 5 or Java user interface and a command-line interface (CLI). It can register with Cisco UCS Central Software in a multi-domain Cisco UCS environment, enabling centralized management of distributed systems scaling to thousands of servers. Cisco UCS Manager can be integrated with Cisco UCS Director to facilitate orchestration and to provide support for converged infrastructure and Infrastructure as a Service (IaaS).

The Cisco UCS XML API provides comprehensive access to all Cisco UCS Manager functions. The API provides Cisco UCS system visibility to higher-level systems management tools from independent software vendors (ISVs) such as VMware, Microsoft, and Splunk as well as tools from BMC, CA, HP, IBM, and others. ISVs and in-house developers can use the XML API to enhance the value of the Cisco UCS platform according to their unique requirements. Cisco UCS PowerTool for Cisco UCS Manager and the Python Software Development Kit (SDK) help automate and manage configurations within Cisco UCS Manager.

Red Hat Enterprise Linux 7.3

Red Hat® Enterprise Linux® is a high-performing operating system that has delivered outstanding value to IT environments for more than a decade. More than 90 percent of Fortune Global 500 companies use Red Hat products and solutions including Red Hat Enterprise Linux. As the world’s most trusted IT platform, Red Hat Enterprise Linux has been deployed in mission-critical applications at global stock exchanges, financial institutions, leading telcos, and animation studios. It also powers the websites of some of the most recognizable global retail brands.

Red Hat Enterprise Linux:

- Delivers high performance, reliability, and security
- Is certified by the leading hardware and software vendors
- Scales from workstations, to servers, to mainframes
- Provides a consistent application environment across physical, virtual, and cloud deployments

Designed to help organizations make a seamless transition to emerging datacenter models that include virtualization and cloud computing, Red Hat Enterprise Linux includes support for major hardware architectures, hypervisors, and cloud providers, making deployments across physical and different virtual environments predictable and secure. Enhanced tools and new capabilities in this release enable administrators to tailor the application environment to efficiently monitor and manage compute resources and security.

Red Hat Ceph Storage

Red Hat® Ceph Storage is an open, cost-effective, software-defined storage solution that enables massively scalable cloud and object storage workloads. By unifying object, block storage and file storage in one platform, Red Hat Ceph Storage efficiently and automatically manages the petabytes of data needed to run businesses facing massive data growth. Ceph is a self-healing, self-managing platform with no single point of failure. Ceph enables a scale-out cloud infrastructure built on industry standard servers that significantly lowers the cost of storing enterprise data and helps enterprises manage their exponential data growth in an automated fashion.

For OpenStack environments, Red Hat Ceph Storage is tightly integrated with OpenStack services, including Nova, Cinder, Manila, Glance, Keystone, and Swift, and it offers user-driven storage life-cycle management. Voted the No. 1 storage option by OpenStack users, the product’s highly tunable, extensible, and configurable architecture offers mature interfaces for enterprise block and object storage, making it well suited for archival, rich media, and cloud infrastructure environments.

Red Hat Ceph Storage is also ideal for object storage workloads outside of OpenStack because it is proven at web scale, flexible for demanding applications, and offers the data protection, reliability, and availability enterprises demand. It was designed from the ground up for web-scale object storage. Industry-standard APIs allow seamless migration of, and integration with, an enterprise’s applications. A Ceph object storage cluster is accessible via S3, Swift, or native API protocols.

Ceph has a lively and active open source community contributing to its innovation. At Ceph’s core is RADOS, a distributed object store that stores data by spreading it out across multiple industry standard servers. Ceph uses CRUSH (Controller Replication Under Scalable Hashing), a uniquely differentiated data placement algorithm that intelligently distributes the data pseudo-randomly across the cluster for better performance and data protection. Ceph supports both replication and erasure coding to protect data and also provides multi-site disaster recovery options.

Red Hat collaborates with the global open source Ceph community to develop new Ceph features, then packages changes into predictable, stable, enterprise-quality SDS product, which is Red Hat Ceph Storage. This unique development model takes combines the advantage of a large development community with Red Hat’s industry-leading support services to offer new storage capabilities and benefits to enterprises.

Solution Design

Solution Overview

The current solution based on Cisco UCS and Red Hat Ceph Storage is divided into multiple sections and covers three main aspects:

1. Design of an Object Storage Solution based on Cisco UCS and Red Hat Ceph Storage.

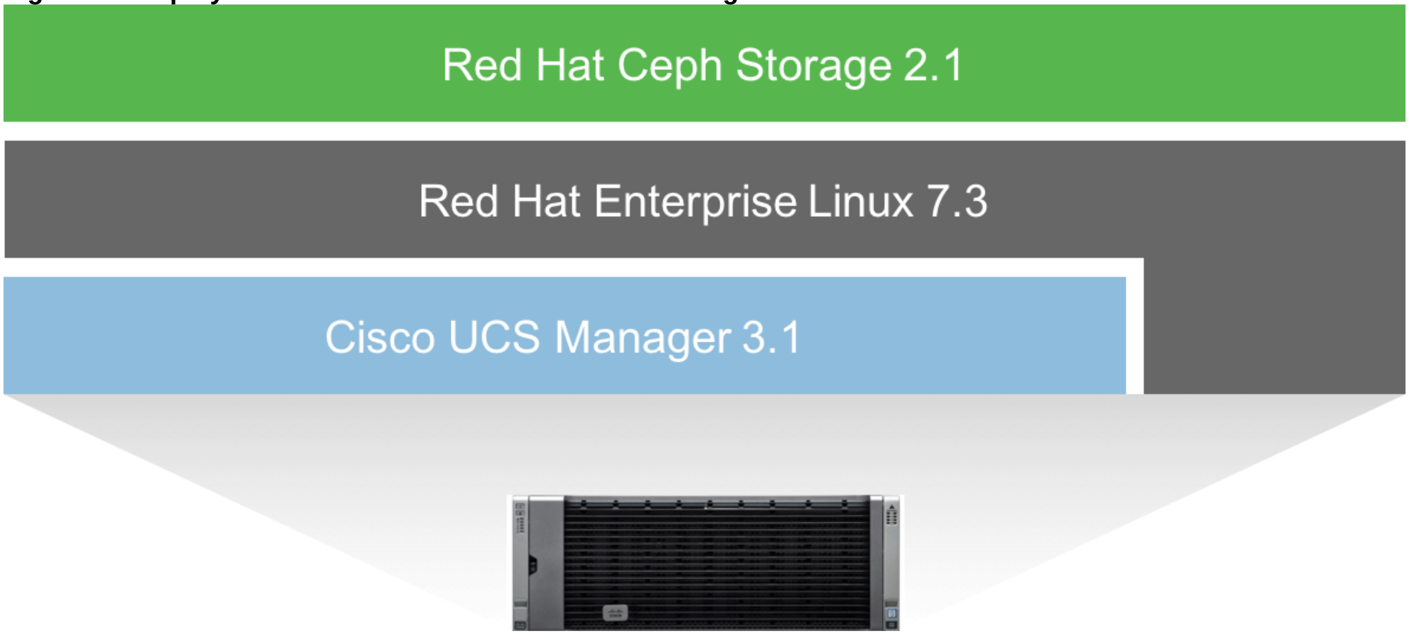
2. Deployment of the Solution (Figure 7) is divided into three areas:

   — Integration and configuration of the Cisco UCS hardware into Cisco UCS Manager

   — Base installation of Red Hat Enterprise Linux

   — Deployment of Red Hat Ceph Storage

Figure 7 Deployment Parts for the Cisco Validated Design



3. Operational guide to work with Red Hat Ceph Storage on Cisco UCS

   — Expansion of the current cluster by adding one more Cisco UCS S3260 Storage Server with two C3x60 M4 server nodes working as OSD nodes

   — Expansion of the current cluster by adding three more Cisco UCS C220 M4S Rack Server working as RADOS gateways for object storage

Design Principles of Red Hat Ceph Storage on Cisco UCS

A general design of a Red Hat Ceph Storage solution should consider the principles shown in Figure 8.

1. Quality need for scale-out storage – Scalability, dynamic provisioning across a unified namespace, and performance-at-scale are common reasons why people chose to add distributed scale-out storage to their

datacenters. For a few use cases, such as primary storage for scale-up Oracle RDBMS, traditional storage appliances remain the right solution

2. Design for a target workload – Red Hat Ceph Storage pools can be deployed to serve three different types of workload categories, including IOPS-intensive, throughput-intensive, and capacity-intensive workloads. As noted in Table 2 , server configurations should be chosen accordingly.

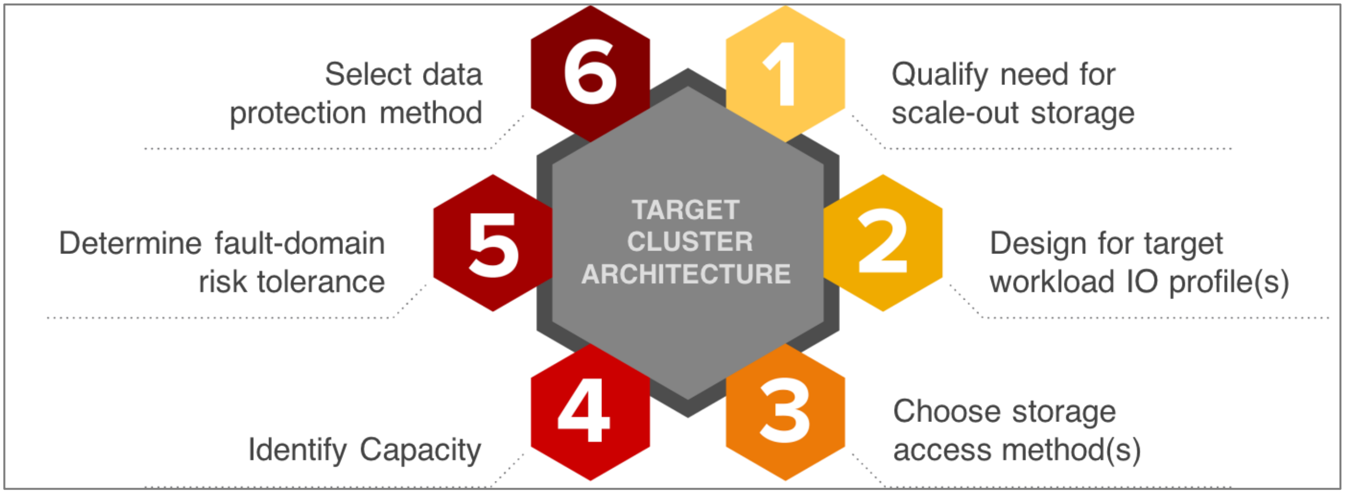
3. Storage Access Method – Red Hat Ceph Storage supports both block access pools and object access pools within a single Ceph cluster (additionally, distributed file access is in tech preview at time of writing). Block access is supported on replicated pools. Object access is supported on either replicated or erasure-coded pools.

4. Capacity – Based on the cluster storage capacity needs, standard, dense, or ultra-dense servers can be chosen to sit beneath Ceph storage pools. Cisco UCS C-Series and Cisco UCS S-Series provide several well-suited server models to choose from.

5. Fault-domain risk tolerance – Ceph clusters are self-healing following hardware failure. Customers wanting to reduce performance and resource impact during self-healing should observe minimum cluster server recommendations described in Table 2 below.

6. Data Protection method – With Replication and Erasure Coding, Red Hat Ceph Storage offers two data protection methods that could affect the overall design. Erasure-coded pools can provide greater price/performance, while replicated pools typically provide higher absolute performance.

Figure 8 Red Hat Ceph Storage Design Consideration


6
Select data protection method
1
Qualify need for scale-out storage
TARGET CLUSTER ARCHITECTURE
5
Determine fault-domain risk tolerance
2
Design for target workload IO profile(s)
4
Identify Capacity
3
Choose storage access method(s)

Based on the previous section of design principles there are some technical specifications that have to be followed for a successful implementation. The technical specifications are shown in Table 2

Table 2 Technical Specifications for Red Hat Ceph Storage

Workload	Cluster Size	Network	CPU / Memory	OSD Journal to Disk Media Ratio	Data Protection

IOPS	Min. 10 OSD nodes	10G - 40G	4-10 Core per OSD / 16 GB + 2 GB per OSD	4:1 → SSD:NVMe or all NVMe with co-located journals	Ceph RBD (Block) Replicated Pools
Throughput	Min. 10 OSD nodes	10G - 40G (>10G when > 12 HDDs/node)	1 Core per 2 HDD / 16 GB + 2 GB per OSD	12-18:1 → HDD:NVMe, or 4-5:1 → HDD:SSD	Ceph RBD (Block) Replicated Pools Ceph RGW (Object) Replicated Pools
Capacity-Archive	Min. 7 OSD nodes	10G (or 40G for latency sensitive requirements)	1 Core per 2 HDD / 16 GB + 2 GB per OSD	All HDD with co-located journals	Ceph RGW (Object) Erasure-Coded Pools

The solution for the current Cisco Validated Design follows a mixed workload setup of Throughput- and Capacity-intensive configurations and is classified as follows[2]:

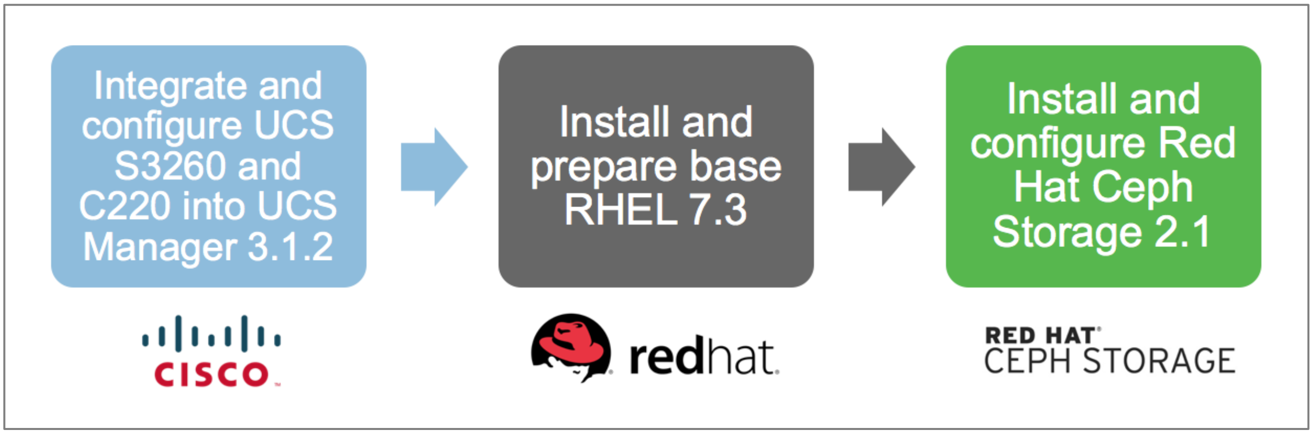
- Cluster Size: Starting with 10 OSD nodes and adding two more OSD nodes.
- Network: All Ceph nodes connected with 40G.
- CPU / Memory: All nodes come with 128 GB memory and more than 40 Core-GHz.
- OSD Disk: The solution is configured for a 6:1 HDD:SSD ratio.
- Data Protection: Ceph RBD with 3 x Replication and Ceph RGW with Erasure Coding.
- Ceph Admin, Monitor, and RADOS gateway nodes are deployed on Cisco UCS C220 M4S rack server.
- Ceph OSD nodes are deployed on Cisco UCS S3260 Storage Server.

Deploying Red Hat Ceph Storage on Cisco UCS

Deploying the solution is based on three steps; the first step is integrating Cisco UCS S3260 Storage Server and Cisco UCS C220 M4S into Cisco UCS Manager, connected to Cisco UCS 6332 Fabric Interconnect and then Cisco Nexus 9332PQ; the second step is the installation of Red Hat Enterprise Linux and preparation for the third step; the installation, configuration and deployment of Red Hat Ceph Storage. Figure 9 illustrates the deployment steps.

Figure 9 Deployment Parts for Red Hat Ceph Storage on Cisco UCS

[2] A detailed Bill of Material list can be found at Bill of Materials

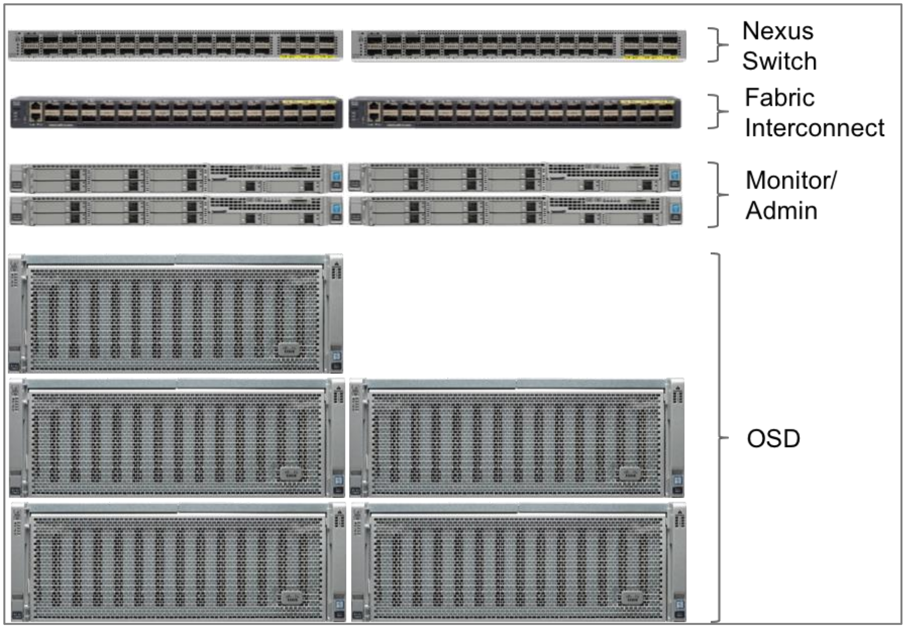

Integrate and configure UCS S3260 and C220 into UCS Manager 3.1.2
Install and prepare base RHEL 7.3
Install and configure Red Hat Ceph Storage 2.1
cisco
redhat
RED HAT CEPH STORAGE

Operational Guide for Red Hat Ceph Storage on Cisco UCS

As an addition to the design and deployment part of the Red Hat Ceph Storage solution on Cisco UCS, the Cisco Validated Design gives an operational guidance on how to add more capacity and another access layer to the starting configuration.

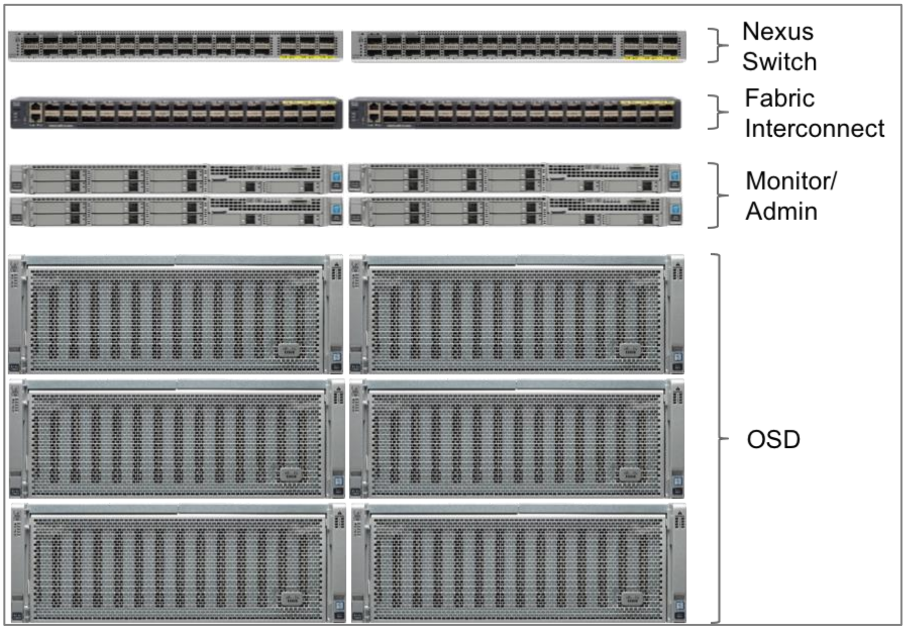
The first part of installation and configuration of the solution contains one Admin node, three Ceph Monitor nodes and 10 Ceph OSD nodes and is shown in Figure 10. This comes along with the minimum size of a Throughput-intensive Ceph cluster of 10 Ceph OSD nodes.

Figure 10 Base Installation of Red Hat Ceph Storage on Cisco UCS



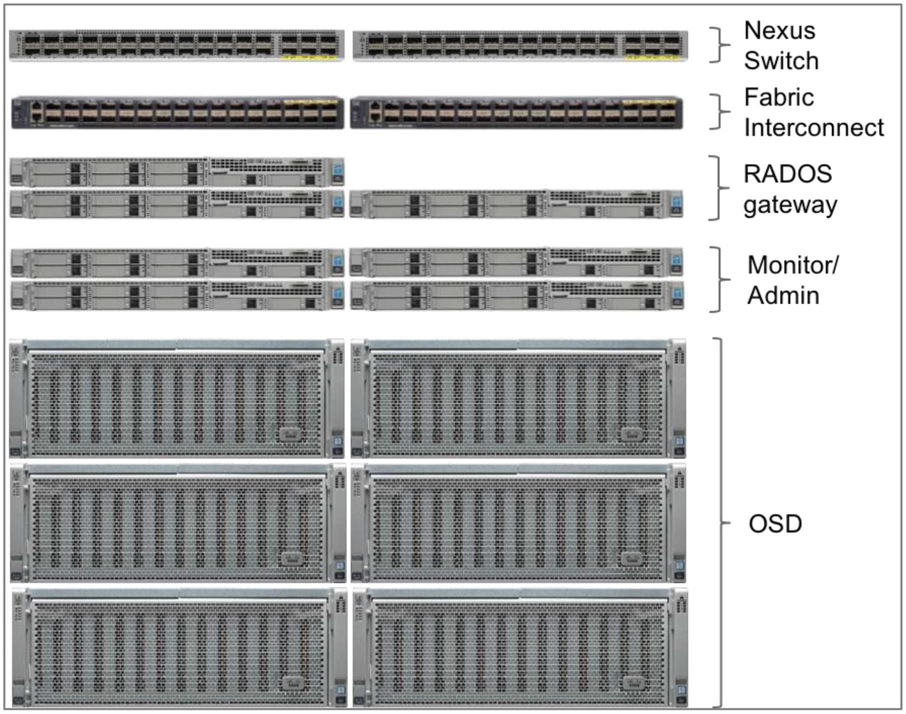
In the second step, the environment gets expanded by adding one more Cisco UCS S3260 Storage Server enclosure with two C3x60 M4 server nodes inside. All steps will be described, showing the simplicity of adding further capacity in less than 30 minutes. Figure 11 shows the additional integration of a Cisco UCS S3260 Storage Server.

Figure 11 Expansion of Red Hat Ceph Storage Cluster with Ceph OSD Nodes



In the last step, the cluster gets further expanded by adding an object storage pool accessed via the RADOS Gateway (RGW). Three more Cisco UCS C220 M4S nodes are getting implemented with Cisco UCS Manager, installed with Red Hat Enterprise Linux and Red Hat Ceph Storage. Figure 12 shows the final infrastructure of this CVD.

Figure 12 Expansion of Red Hat Ceph Storage Cluster with Ceph RADOS Gateways



Requirements

The CVD describes the architecture, design and deployment of a Red Hat Ceph Storage solution on six Cisco UCS S3260 Storage Server, each with two C3x60 M4 nodes and seven Cisco UCS C220 M4S Rack servers providing control-plane functions, including three Ceph Monitor nodes, three Ceph RGW nodes, and one Ceph Admin node. The whole solution is connected to two Cisco UCS 6332 Fabric Interconnects and two Cisco Nexus 9332PQ.

The detailed configuration looks like the following:

- Two Cisco Nexus 9332PQ Switches
- Two Cisco UCS 6332 Fabric Interconnects
- Six Cisco UCS S3260 Storage Servers with two C3x60 M4 server nodes each
- Seven Cisco UCS C220 M4S Rack Servers
- One Cisco R42610 Standard Rack
- Two Vertical Power Distribution Units (PDUs) (Country Specific)

Note: Please contact your Cisco representative for country specific information.

Rack and PDU configuration

Each rack consists of two vertical PDUs. The rack consists of two Cisco Nexus 9332PQ, two Cisco UCS 6332 Fabric Interconnects, 7 Cisco UCS C220 M4S, and 6 Cisco UCS S3260 Storage Server. Each chassis is connected to two vertical PDUs for redundancy, to help ensure availability during power source failure. 0shows the exact layout of the configuration.

Figure 13 Rack Configuration

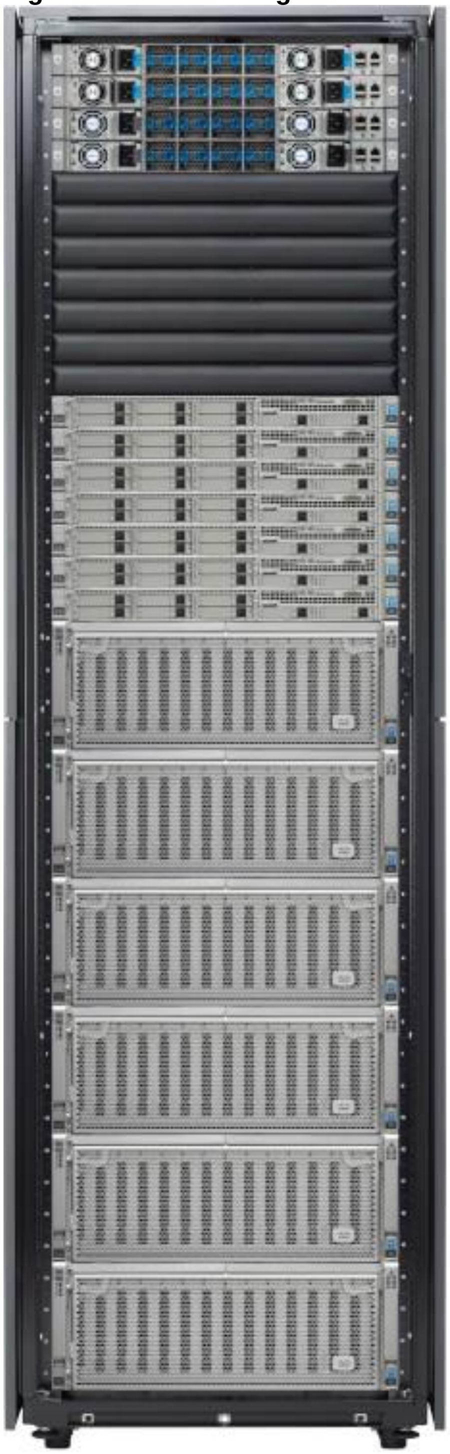


Table 3 Position and Devices

Position	Devices
42	Cisco Nexus 9332PQ
41	Cisco Nexus 9332PQ
40	Cisco UCS 6332 FI
39	Cisco UCS 6332 FI
38	Unused

Position	Devices
37	Unused
36	Unused
35	Unused
34	Unused
33	Unused
32	Unused
31	Cisco UCS C220 M4S
30	Cisco UCS C220 M4S
29	Cisco UCS C220 M4S
28	Cisco UCS C220 M4S
27	Cisco UCS C220 M4S
26	Cisco UCS C220 M4S
25	Cisco UCS C220 M4S
24	Cisco UCS S3260 Storage Server
23	
22	
21	
20	Cisco UCS S3260 Storage Server
19	
18	
17	
16	Cisco UCS S3260 Storage Server
15	
14	
13	
12	Cisco UCS S3260 Storage Server
11	
10	
9	
8	Cisco UCS S3260 Storage Server
7	
6	
5	
4	Cisco UCS S3260 Storage Server
3	
2	
1	

Physical Topology and Configuration

The following sections describe the physical design of the solution and the configuration of each component.

Table 4 shows the naming conventions used for this solution.

Table 4   Naming Convention

Device	Function	Name
Cisco Nexus 9332PQ Switch A		N9k-A
Cisco Nexus 9332PQ Switch B		N9k-B
Cisco UCS 6332 Fabric Interconnect A		FI6332-A
Cisco UCS 6332 Fabric Interconnect B		FI6332-B
Cisco UCS C220 M4S	Ceph RADOS Gateway	cephrgw1

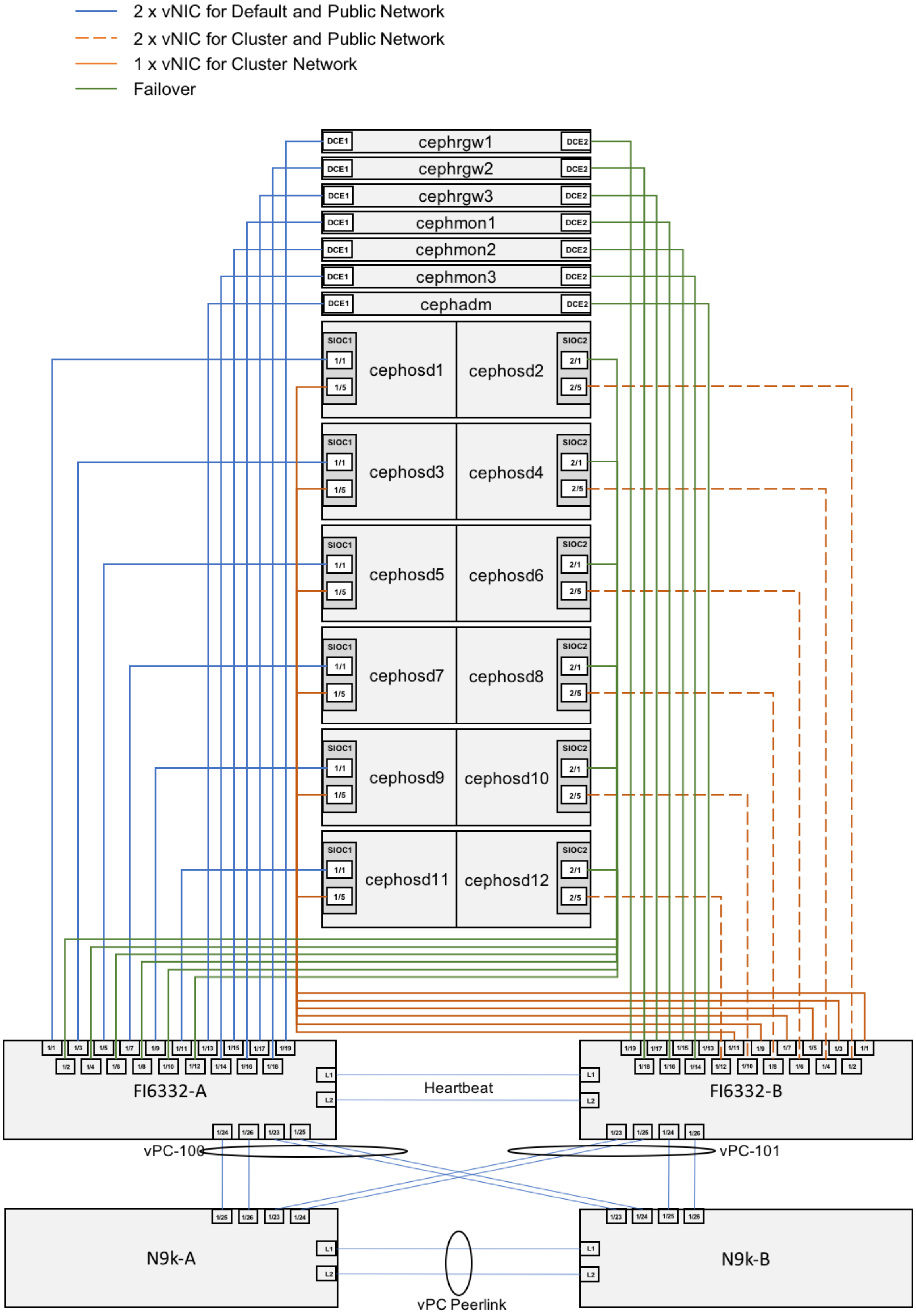
Cisco UCS C220 M4S	Ceph RADOS Gateway	cephrgw2
Cisco UCS C220 M4S	Ceph RADOS Gateway	cephrgw3
Cisco UCS C220 M4S	Ceph Monitor	cephmon1
Cisco UCS C220 M4S	Ceph Monitor	cephmon2
Cisco UCS C220 M4S	Ceph Monitor	cephmon3
Cisco UCS C220 M4S	Ceph Admin	cephadm
Cisco UCS S3260 Storage Server Top Node	Ceph OSD	cephosd1
Cisco UCS S3260 Storage Server Bottom Node	Ceph OSD	cephosd2
Cisco UCS S3260 Storage Server Top Node	Ceph OSD	cephosd3
Cisco UCS S3260 Storage Server Bottom Node	Ceph OSD	cephosd4
Cisco UCS S3260 Storage Server Top Node	Ceph OSD	cephosd5
Cisco UCS S3260 Storage Server Bottom Node	Ceph OSD	cephosd6
Cisco UCS S3260 Storage Server Top Node	Ceph OSD	cephosd7
Cisco UCS S3260 Storage Server Bottom Node	Ceph OSD	cephosd8
Cisco UCS S3260 Storage Server Top Node	Ceph OSD	cephosd9
Cisco UCS S3260 Storage Server Bottom Node	Ceph OSD	cephosd10
Cisco UCS S3260 Storage Server Top Node	Ceph OSD	cephosd11
Cisco UCS S3260 Storage Server Bottom Node	Ceph OSD	cephosd12

The connectivity of the solution is based on 40 Gbit. All components are connected together via 40 Gbit QSFP cables. Between both Cisco Nexus 9332PQ switches are 2 x 40 Gbit cabling. Each Cisco UCS 6332 Fabric Interconnect is connected via 2 x 40 Gbit to each Cisco UCS 9332PQ switch. And each Cisco UCS C220 M4S and each Cisco UCS C3x60 M4 server is connected with a single 40 Gbit cable to each Fabric Interconnect.

The exact cabling for the Red Hat Ceph Storage solution is illustrated in Figure 14. It shows also the separate vNIC configuration for Public and Cluster network to avoid traffic congestion. The Public Network for the top node of the Cisco UCS S3260 Storage Server is connected to Fabric Interconnect A and the Public Network for the bottom node of the Cisco UCS S3260 Storage Server is connected to Fabric Interconnect B.

All vNICs for the Cluster Network are connected to Fabric Interconnect B to keep the whole Cluster traffic under a single Fabric Interconnect. All vNICs are configured for Fabric Interconnect failover.

Figure 14 Red Hat Ceph Storage Solution Cabling Diagram



For a better reading and overview the exact physical connectivity between the Cisco UCS 6332 Fabric Interconnects and the Cisco UCS S-Series and C-Class server is shown in Table 5 .

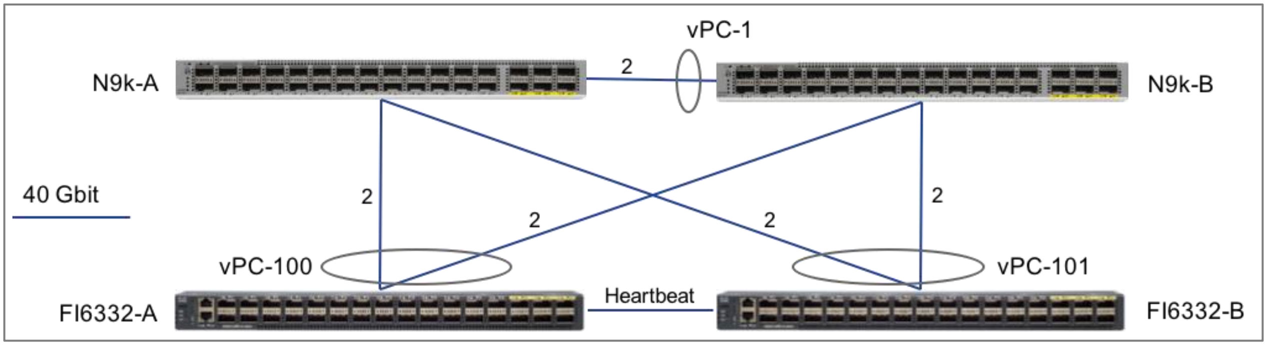
Table 5 Physical Connectivity between FI 6332 and S3260/C220 M4S

Port	Role	FI6332-A	FI6332-B
Eth1/1	Server	cephosd1, SIOC1/1	cephosd1, SIOC1/5
Eth1/2	Server	cephosd2, SIOC2/1	cephosd2, SIOC2/5
Eth1/3	Server	cephosd3, SIOC1/1	cephosd3, SIOC1/5
Eth1/4	Server	cephosd4, SIOC2/1	cephosd4, SIOC2/5
Eth1/5	Server	cephosd5, SIOC1/1	cephosd5, SIOC1/5
Eth1/6	Server	cephosd6, SIOC2/1	cephosd6, SIOC2/5
Eth1/7	Server	cephosd7, SIOC1/1	cephosd7, SIOC1/5
Eth1/8	Server	cephosd8, SIOC2/1	cephosd8, SIOC2/5
Eth1/9	Server	cephosd9, SIOC1/1	cephosd9, SIOC1/5
Eth1/10	Server	cephosd10, SIOC2/1	cephosd10, SIOC2/5
Eth1/11	Server	cephosd11, SIOC1/1	cephosd11, SIOC1/5
Eth1/12	Server	cephosd12, SIOC2/1	cephosd12, SIOC2/5
Eth1/13	Server	cephadm, DCE1	cephadm, DCE2
Eth1/14	Server	cephmon3, DCE1	cephmon3, DCE2
Eth1/15	Server	cephmon2, DCE1	cephmon2, DCE2
Eth1/16	Server	cephmon1, DCE1	cephmon1, DCE2
Eth1/17	Server	cephrgw3, DCE1	cephrgw3, DCE2
Eth1/18	Server	cephrgw2, DCE1	cephrgw2, DCE2
Eth1/19	Server	cephrgw1, DCE1	cephrgw1, DCE2
Eth1/23	Network	N9k-B, Eth1/23	N9k-A, Eth1/23
Eth1/24	Network	N9k-A, Eth1/25	N9k-B, Eth1/25
Eth1/25	Network	N9k-B, Eth1/24	N9k-A, Eth1/24
Eth1/26	Network	N9k-A, Eth1/26	N9k-B, Eth1/26

Figure 15 shows a more detailed view on the cabling and configuration of Cisco Nexus 9332PQ and Cisco UCS 6332 Fabric Interconnect. Between each Cisco UCS 6332 Fabric Interconnect and both Cisco Nexus 9332PQ is one virtual Port Channel (vPC) configured. vPCs allow links that are physically connected to two different Cisco Nexus 9000 switches to appear to the Fabric Interconnect as coming from a single device and as part of a single port channel. vPC-100 connects FI6332-A with N9k-A and N9k-B. vPC-101 connects FI6332-B with N9k-A and N9k-B. The overall bandwidth for each Port Channel is 160 Gbit.

Between both Cisco Nexus 9332PQ is a vPC peer link configured, containing two 40 Gbit lines.

Figure 15 Cabling and Configuration of Cisco Nexus 9332PQ and Cisco UCS 6332 Fabric Interconnect



The connectivity between Cisco Nexus 9332PQ and Cisco UCS 6332 Fabric Interconnect is shown in Table 6 .

Table 6 Physical Connectivity between Cisco Nexus 9332PQ and Cisco UCS 6332 Fabric Interconnect

Port	N9k-A	N9k-B
Eth1/23	FI6332-B, Eth1/23, vPC-101	FI6332-A, Eth1/23, vPC-100
Eth1/24	FI6332-B, Eth1/25, vPC-101	FI6332-A, Eth1/25, vPC-100
Eth1/25	FI6332-A, Eth1/24, vPC-100	FI6332-B, Eth1/24, vPC-101
Eth1/26	FI6332-A, Eth1/26, vPC-100	FI6332-B, Eth1/26, vPC-101
Eth1/31	N9k-B, Eth1/31, vPC-1	N9k-A, Eth1/31, vPC-1
Eth1/32	N9k-B, Eth1/32, vPC-1	N9k-A, Eth1/32, vPC-1

Software Distributions and Versions

The required software distribution versions are listed below in Table 7 .

Table 7 Software Versions

Layer	Component	Version or Release
Compute (Chassis) S3260	Board Controller	1.0.14
	Chassis Management Controller	2.0(13e)
	Shared Adapter	4.1(2d)
	SAS Expander	04.08.01.B073
Compute (Server Nodes) C3x60 M4	BIOS	C3x60M4.2.0.13c
	Board Controller	2.0
	CIMC Controller	2.0(13f)
	Storage Controller	29.00.1-0042
Compute (Rack Server) C220 M4S	Adapter	4.1(2d)
	BIOS	C220M4.2.0.13d
	Board Controller	32.0
	CIMC Controller	2.0(13f)

Layer	Component	Version or Release
	FlexFlash Controller	1.3.2 build 165
	Storage Controller	24.12.1-0049
Network 6332 Fabric Interconnect	UCS Manager	3.1(2c)
	Kernel	5.0(3)N2(3.12c)
	System	5.0(3)N2(3.12c)
Network Nexus 9332PQ	BIOS	07.59
	NXOS	7.0(3)I5(1)
Software	Red Hat Enterprise Linux Server	7.3 (x86_64)
	Ceph	10.2.3-13.el7cp

Deployment of Hardware and Software

Fabric Configuration

This section provides the details for configuring a fully redundant, highly available Cisco UCS 6332 fabric configuration.

- Initial setup of the Fabric Interconnect A and B.
- Connect to Cisco UCS Manager using virtual IP address of the web browser.
- Launch Cisco UCS Manager.
- Enable server and uplink ports.
- Start discovery process.
- Create pools and policies for service profile template.
- Create chassis and storage profiles.
- Create Service Profile templates and appropriate Service Profiles.
- Associate Service Profiles to servers.

Initial Setup of Cisco UCS 6332 Fabric Interconnects

This section describes the initial setup of the Cisco UCS 6332 Fabric Interconnects A and B

Configure Fabric Interconnect A

To configure Fabric Interconnect A, complete the following steps:

1. Connect to the console port on the first Cisco UCS 6332 Fabric Interconnect.
2. At the prompt to enter the configuration method, enter `console` to continue.
3. If asked to either perform a new setup or restore from backup, enter `setup` to continue.
4. Enter `y` to continue to set up a new Fabric Interconnect.
5. Enter `n` to enforce strong passwords.
6. Enter the password for the admin user.
7. Enter the same password again to confirm the password for the admin user.
8. When asked if this fabric interconnect is part of a cluster, answer `y` to continue.
9. Enter `A` for the switch fabric.
10. Enter the cluster name `FI6332` for the system name.

11. Enter the Mgmt0 IPv4 address.

12. Enter the Mgmt0 IPv4 netmask.

13. Enter the IPv4 address of the default gateway.

14. Enter the cluster IPv4 address.

15. To configure DNS, answer `y`.

16. Enter the DNS IPv4 address.

17. Answer `y` to set up the default domain name.

18. Enter the default domain name.

19. Review the settings that were printed to the console, and if they are correct, answer `yes` to save the configuration.

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

Example Setup for Fabric Interconnect A

```
               ---- Basic System Configuration Dialog ----


  This setup utility will guide you through the basic configuration of

  the system. Only minimal configuration including IP connectivity to

  the Fabric interconnect and its clustering mode is performed through these
steps.


  Type Ctrl-C at any time to abort configuration and reboot system.

  To back track or make modifications to already entered values,

  complete input till end of section and answer no when prompted

  to apply configuration.


  Enter the configuration method. (console/gui) ? console

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

  You have chosen to setup a new Fabric interconnect. Continue? (y/n): y

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

  Enter the password for "admin":

  Confirm the password for "admin":
```

```
  Is this Fabric interconnect part of a cluster(select 'no' for standalone)?
(yes/no) [n]: yes
  Enter the switch fabric (A/B): A
  Enter the system name:  FI6332
  Physical Switch Mgmt0 IP address : 172.25.206.221
  Physical Switch Mgmt0 IPv4 netmask : 255.255.255.0
  IPv4 address of the default gateway : 172.25.206.1
  Cluster IPv4 address : 172.25.206.220
  Configure the DNS Server IP address? (yes/no) [n]: yes
    DNS IP address : 173.36.131.10
  Configure the default domain name? (yes/no) [n]:
  Join centralized management environment (UCS Central)? (yes/no) [n]:

  Following configurations will be applied:

    Switch Fabric=A
    System Name=FI6332
    Enforced Strong Password=no
    Physical Switch Mgmt0 IP Address=172.25.206.221
    Physical Switch Mgmt0 IP Netmask=255.255.255.0
    Default Gateway=172.25.206.1
    Ipv6 value=0
    DNS Server=173.36.131.10

    Cluster Enabled=yes
    Cluster IP Address=172.25.206.220
    NOTE: Cluster IP will be configured only after both Fabric Interconnects
are initialized.
          UCSM will be functional only after peer FI is configured in
clustering mode.

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

```
 Configuration file - Ok


Cisco UCS 6300 Series Fabric Interconnect

FI6332-A login:
```

Configure Fabric Interconnect B

To configure Fabric Interconnect B, complete the following steps:

1. Connect to the console port on the second Cisco UCS 6332 Fabric Interconnect.
2. When prompted to enter the configuration method, enter `console` to continue.
3. The installer detects the presence of the partner Fabric Interconnect and adds this fabric interconnect to the cluster. Enter `y` to continue the installation.
4. Enter the admin password that was configured for the first Fabric Interconnect.
5. Enter the Mgmt0 IPv4 address.
6. Answer `yes` to save the configuration.
7. Wait for the login prompt to confirm that the configuration has been saved.

Example Setup for Fabric Interconnect B

```
           ---- Basic System Configuration Dialog ----


  This setup utility will guide you through the basic configuration of

  the system. Only minimal configuration including IP connectivity to

  the Fabric interconnect and its clustering mode is performed through these
steps.


  Type Ctrl-C at any time to abort configuration and reboot system.

  To back track or make modifications to already entered values,

  complete input till end of section and answer no when prompted

  to apply configuration.


  Enter the configuration method. (console/gui) ? console


  Installer has detected the presence of a peer Fabric interconnect. This
Fabric interconnect will be added to the cluster. Continue (y/n) ? y
```

```
  Enter the admin password of the peer Fabric interconnect:

    Connecting to peer Fabric interconnect... done

    Retrieving config from peer Fabric interconnect... done

    Peer Fabric interconnect Mgmt0 IPv4 Address: 172.25.206.221

    Peer Fabric interconnect Mgmt0 IPv4 Netmask: 255.255.255.0

    Cluster IPv4 address          : 172.25.206.220


    Peer FI is IPv4 Cluster enabled. Please Provide Local Fabric Interconnect
Mgmt0 IPv4 Address


  Physical Switch Mgmt0 IP address : 172.25.206.222



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

  Applying configuration. Please wait.


Fri Sep 30 05:41:48 UTC 2016

  Configuration file - Ok



Cisco UCS 6300 Series Fabric Interconnect

FI6332-B login:
```

Logging Into Cisco UCS Manager

To login to Cisco UCS Manager, complete the following steps:

1. Open a Web browser and navigate to the Cisco UCS 6332 Fabric Interconnect cluster address.

2. Click the Launch link to download the Cisco UCS Manager software.

3. If prompted to accept security certificates, accept as necessary.

4. Click Launch UCS Manager HTML.

5. When prompted, enter `admin` for the username and enter the administrative password.

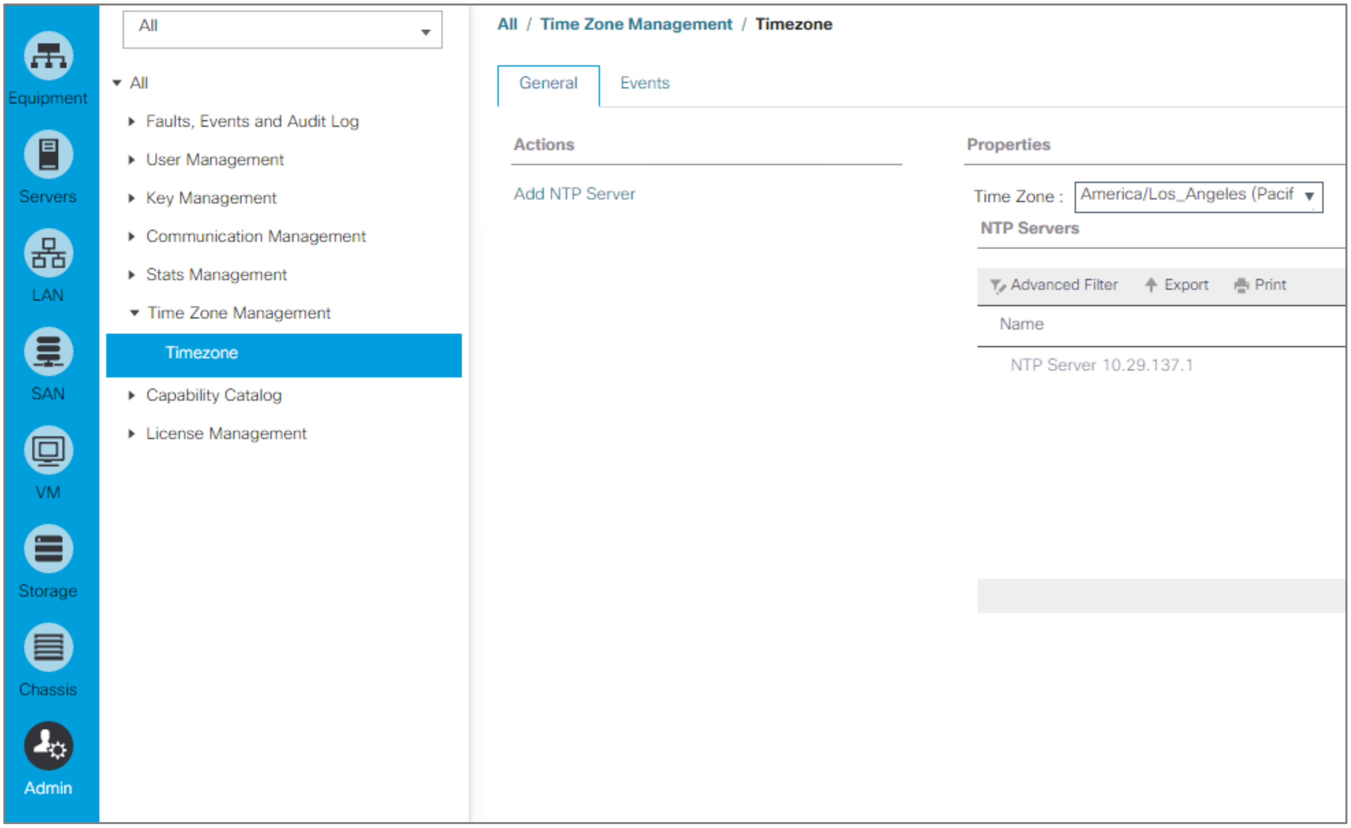
6. Click Login to log in to the Cisco UCS Manager.

Configure NTP Server

To configure the NTP server for the Cisco UCS environment, complete the following steps:

1. Select `Admin` tab on the left site.
2. Select Time Zone Management.
3. Select `Time Zone`.
4. Under `Properties` select your time zone.
5. Select Add NTP Server.
6. Enter the IP address of the NTP server.
7. Select `OK`.

Figure 16 Adding a NTP Server - Summary



Time Zone : America/Los_Angeles (Pacif

NTP Servers

NTP Server 10.29.137.1

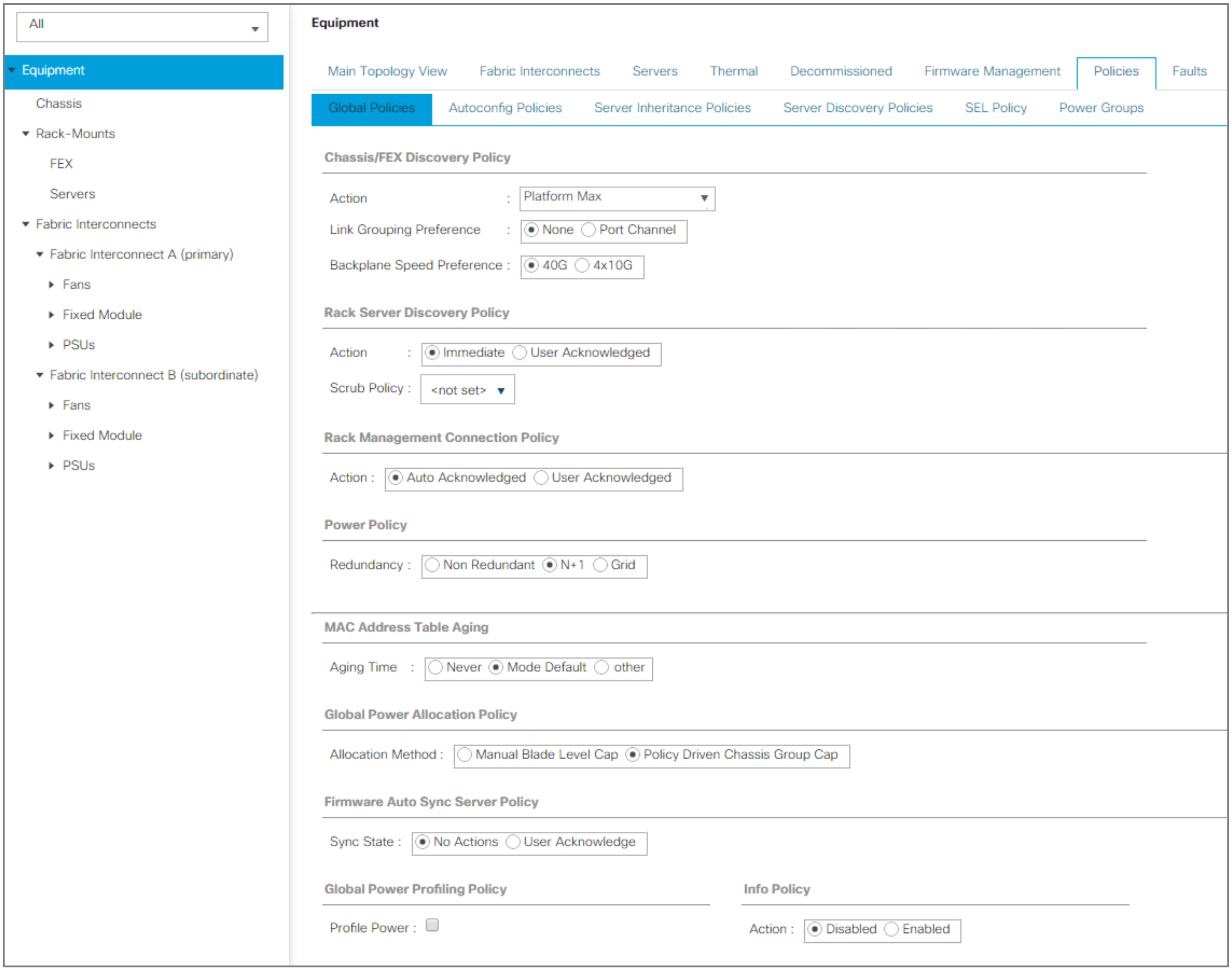
Initial Base Setup of the Environment

Configure Global Policies

To configure the Global Policies, complete the following steps:

1. Select the `Equipment` tab on the left site of the window.
2. Select `Policies` on the right site.
3. Select Global Policies.
4. Under Chassis/FEX Discovery Policy select `Platform Max` under Action.
5. Select `40G` under Backplane Speed Preference.
6. Under Rack Server Discovery Policy select `Immediate` under Action.
7. Under Rack Management Connection Policy select `Auto Acknowledged` under Action.
8. Under Power Policy select Redundancy `N+1`.
9. Under Global Power Allocation Policy select `Policy Driven`.
10. Select Save Changes.

Figure 17 Configuration of Global Policies



Chassis/FEX Discovery Policy

Action : Platform Max

Link Grouping Preference : ● None ○ Port Channel

Backplane Speed Preference : ● 40G ○ 4x10G

Rack Server Discovery Policy

Action : ● Immediate ○ User Acknowledged

Scrub Policy : <not set>

Rack Management Connection Policy

Action : ● Auto Acknowledged ○ User Acknowledged

Power Policy

Redundancy : ○ Non Redundant ● N+1 ○ Grid

MAC Address Table Aging

Aging Time : ○ Never ● Mode Default ○ other

Global Power Allocation Policy

Allocation Method : ○ Manual Blade Level Cap ● Policy Driven Chassis Group Cap

Firmware Auto Sync Server Policy

Sync State : ● No Actions ○ User Acknowledge

Global Power Profiling Policy

Profile Power : □

Info Policy

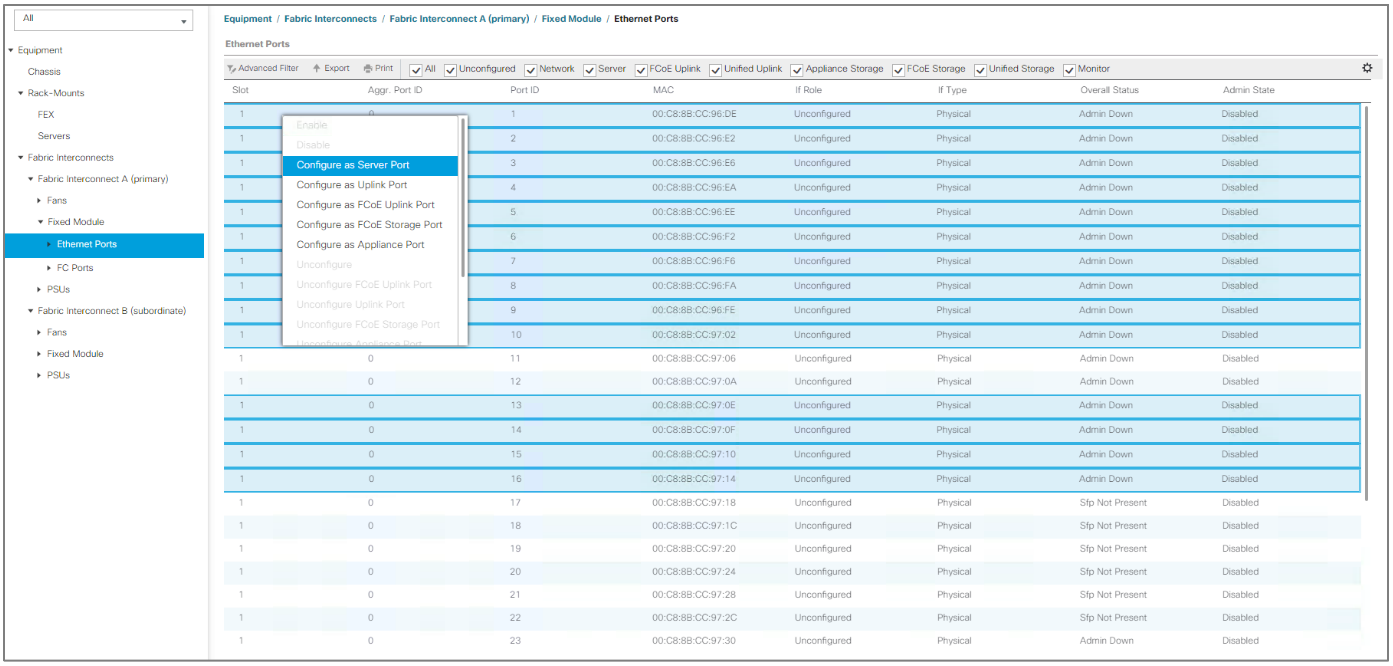
Action : ● Disabled ○ Enabled

Enable Fabric Interconnect A Ports for Server

To enable server ports, complete the following steps:

1. Select the `Equipment` tab on the left site.
2. Select Equipment > Fabric Interconnects > Fabric Interconnect A (subordinate) > Fixed Module.
3. Click `Ethernet Ports` section.
4. Select Ports 1-10 and 13-16, right-click and then select `Configure as Server Port`.
5. Click `Yes` and then `OK`.
6. Repeat the same steps for Fabric Interconnect B.

Figure 18 Configuration of Server Ports



Slot	Aggr. Port ID	Port ID	MAC	If Role	If Type	Overall Status	Admin State
1	0	1	00:C8:8B:CC:96:DE	Unconfigured	Physical	Admin Down	Disabled
1		2	00:C8:8B:CC:96:E2	Unconfigured	Physical	Admin Down	Disabled
1		3	00:C8:8B:CC:96:E6	Unconfigured	Physical	Admin Down	Disabled
1		4	00:C8:8B:CC:96:EA	Unconfigured	Physical	Admin Down	Disabled
1		5	00:C8:8B:CC:96:EE	Unconfigured	Physical	Admin Down	Disabled
1		6	00:C8:8B:CC:96:F2	Unconfigured	Physical	Admin Down	Disabled
1		7	00:C8:8B:CC:96:F6	Unconfigured	Physical	Admin Down	Disabled
1		8	00:C8:8B:CC:96:FA	Unconfigured	Physical	Admin Down	Disabled
1		9	00:C8:8B:CC:96:FE	Unconfigured	Physical	Admin Down	Disabled
1		10	00:C8:8B:CC:97:02	Unconfigured	Physical	Admin Down	Disabled
1	0	11	00:C8:8B:CC:97:06	Unconfigured	Physical	Admin Down	Disabled
1	0	12	00:C8:8B:CC:97:0A	Unconfigured	Physical	Admin Down	Disabled
1	0	13	00:C8:8B:CC:97:0E	Unconfigured	Physical	Admin Down	Disabled
1	0	14	00:C8:8B:CC:97:0F	Unconfigured	Physical	Admin Down	Disabled
1	0	15	00:C8:8B:CC:97:10	Unconfigured	Physical	Admin Down	Disabled
1	0	16	00:C8:8B:CC:97:14	Unconfigured	Physical	Admin Down	Disabled
1	0	17	00:C8:8B:CC:97:18	Unconfigured	Physical	Sfp Not Present	Disabled
1	0	18	00:C8:8B:CC:97:1C	Unconfigured	Physical	Sfp Not Present	Disabled
1	0	19	00:C8:8B:CC:97:20	Unconfigured	Physical	Sfp Not Present	Disabled
1	0	20	00:C8:8B:CC:97:24	Unconfigured	Physical	Sfp Not Present	Disabled
1	0	21	00:C8:8B:CC:97:28	Unconfigured	Physical	Sfp Not Present	Disabled
1	0	22	00:C8:8B:CC:97:2C	Unconfigured	Physical	Sfp Not Present	Disabled
1	0	23	00:C8:8B:CC:97:30	Unconfigured	Physical	Admin Down	Disabled

Enable Fabric Interconnect A Ports for Uplinks

To enable uplink ports, complete the following steps:

1. Select the `Equipment` tab on the left site.
2. Select Equipment > Fabric Interconnects > Fabric Interconnect A (subordinate) > Fixed Module.
3. Click `Ethernet Ports` section.
4. Select Ports 23-26, right-click and then select `Configure as Uplink Port`.
5. Click `Yes` and then `OK`.
6. Repeat the same steps for Fabric Interconnect B.

Label Each Chassis for Identification

To label each chassis for better identification, complete the following steps:

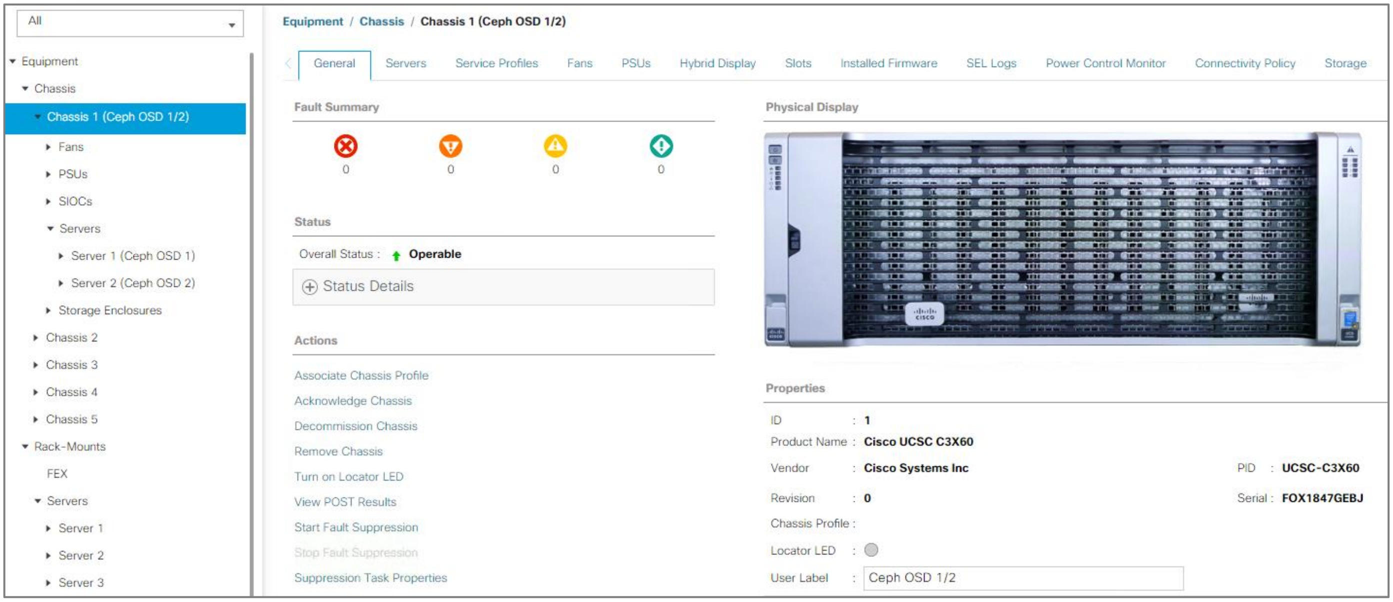
1. Select the `Equipment` tab on the left site.
2. Select Chassis > Chassis 1.
3. In the `Properties` section on the right go to `User Label` and add `Ceph OSD 1/2` to the field.
4. Repeat the previous steps for Chassis 2 – 5 by using the following labels (Table 8):

Table 8 Chassis Label

Chassis	Name

Chassis 1	Ceph OSD 1/2
Chassis 2	Ceph OSD 3/4
Chassis 3	Ceph OSD 5/6
Chassis 4	Ceph OSD 7/8
Chassis 5	Ceph OSD 9/10

Figure 19 Labeling of all Chassis



Overall Status : Operable

ID : 1
Product Name : Cisco UCSC C3X60
Vendor : Cisco Systems Inc
PID : UCSC-C3X60
Revision : 0
Serial : FOX1847GEBJ
Chassis Profile :
Locator LED :
User Label : Ceph OSD 1/2

Label Each Server for Identification

To label each server for better identification, complete the following steps:

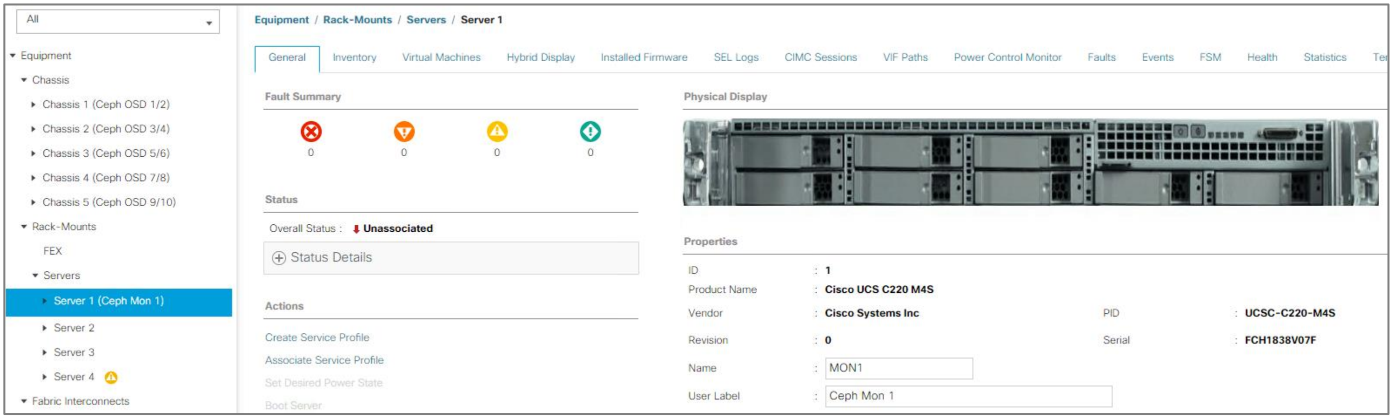
1. Select the `Equipment` tab on the left site.
2. Select Chassis > Chassis 1 > Server 1.
3. In the `Properties` section on the right go to `User Label` and add `Ceph OSD 1` to the field.
4. Repeat the previous steps for `Server 2` of `Chassis 1` and for all other servers of Chassis 2 – 5 according to Table 9 .
5. Go then to `Servers > Rack-Mounts > Servers >` and repeat the step for all servers according to Table 9 .

Table 9 Server Label

Server	Name

Chassis 1 / Server 1	Ceph OSD 1
Chassis 1 / Server 2	Ceph OSD 2
Chassis 2 / Server 1	Ceph OSD 3
Chassis 2 / Server 2	Ceph OSD 4
Chassis 3 / Server 1	Ceph OSD 5
Chassis 3 / Server 2	Ceph OSD 6
Chassis 4 / Server 1	Ceph OSD 7
Chassis 4 / Server 2	Ceph OSD 8
Chassis 5 / Server 1	Ceph OSD 9
Chassis 5 / Server 2	Ceph OSD 10
Rack-Mount / Server 1	Ceph Mon 1
Rack-Mount / Server 2	Ceph Mon 2
Rack-Mount / Server 3	Ceph Mon 3
Rack-Mount / Server 4	Ceph Adm

Figure 20 Labeling of Rack Servers



Status

Overall Status : Unassociated

Properties

ID : 1
Product Name : Cisco UCS C220 M4S
Vendor : Cisco Systems Inc
PID : UCSC-C220-M4S
Revision : 0
Serial : FCH1838V07F
Name : MON1
User Label : Ceph Mon 1

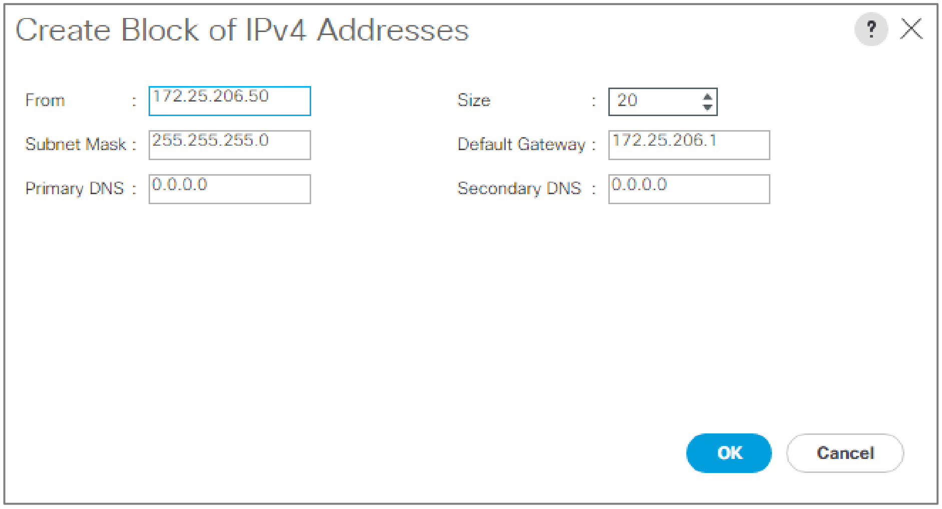
Create KVM IP Pool

To create a KVM IP Pool, complete the following steps:

1. Select the `LAN` tab on the left site.
2. Go to LAN > Pools > root > IP Pools > IP Pool ext-mgmt.
3. Right-click Create Block of IPv4 Addresses.
4. Enter an IP Address in the `From` field.
5. Enter `Size` 20.

6. Enter your `Subnet Mask`.
7. Fill in your `Default Gateway`.
8. Enter your `Primary DNS` and `Secondary DNS` if needed.
9. Click `OK`.

Figure 21 Create Block of IPv4 Addresses



Create Block of IPv4 Addresses

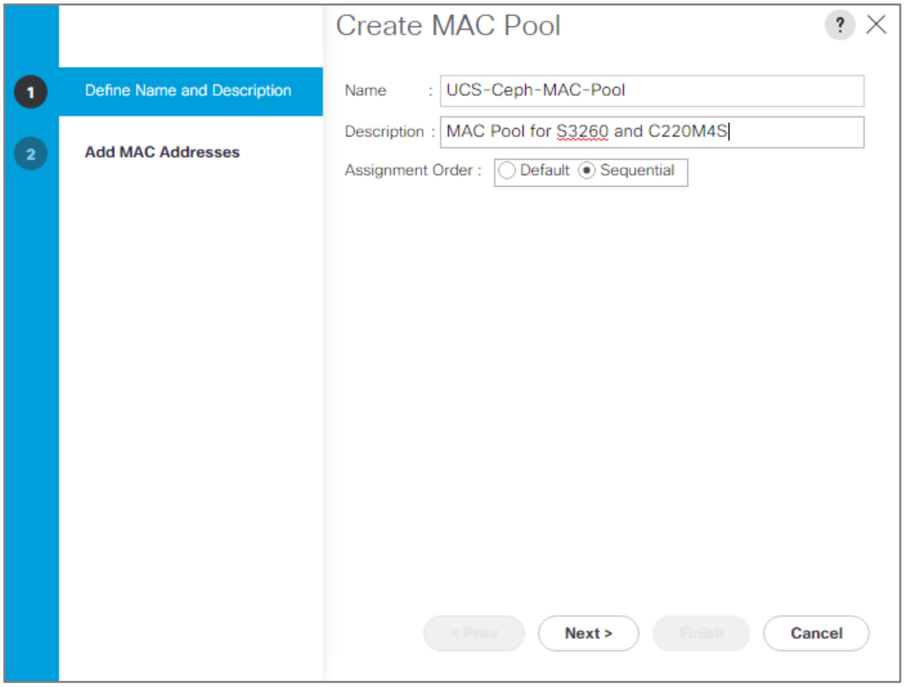
From :	172.25.206.50	Size :	20
Subnet Mask :	255.255.255.0	Default Gateway :	172.25.206.1
Primary DNS :	0.0.0.0	Secondary DNS :	0.0.0.0

Create MAC Pool

To create a MAC Pool, complete the following steps:

1. Select the `LAN` tab on the left site.
2. Go to LAN > Pools > root > Mac Pools and right-click Create MAC Pool.
3. Type in `UCS-Ceph-MAC-Pool` for Name.
4. (Optional) Enter a `Description` of the MAC Pool.
5. Set Assignment Order as Sequential.

Figure 22 Create MAC Pool



Name : UCS-Ceph-MAC-Pool

Description : MAC Pool for S3260 and C220M4S

Assignment Order : ○ Default ● Sequential

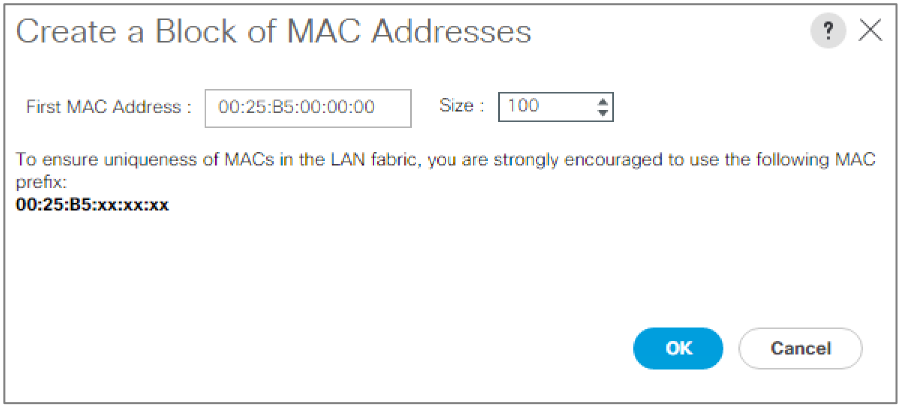
6. Click Next.

7. Click Add.

8. Specify a starting MAC address.

9. Specify a size of the MAC address pool, which is sufficient to support the available server resources, for example, 100.

Figure 23 Create a Block of MAC Addresses



First MAC Address : 00:25:B5:00:00:00

Size : 100

To ensure uniqueness of MACs in the LAN fabric, you are strongly encouraged to use the following MAC prefix:
00:25:B5:xx:xx:xx

10. Click OK.

11. Click Finish.

Create UUID Pool

To create a UUID Pool, complete the following steps:

1. Select the `Servers` tab on the left site.

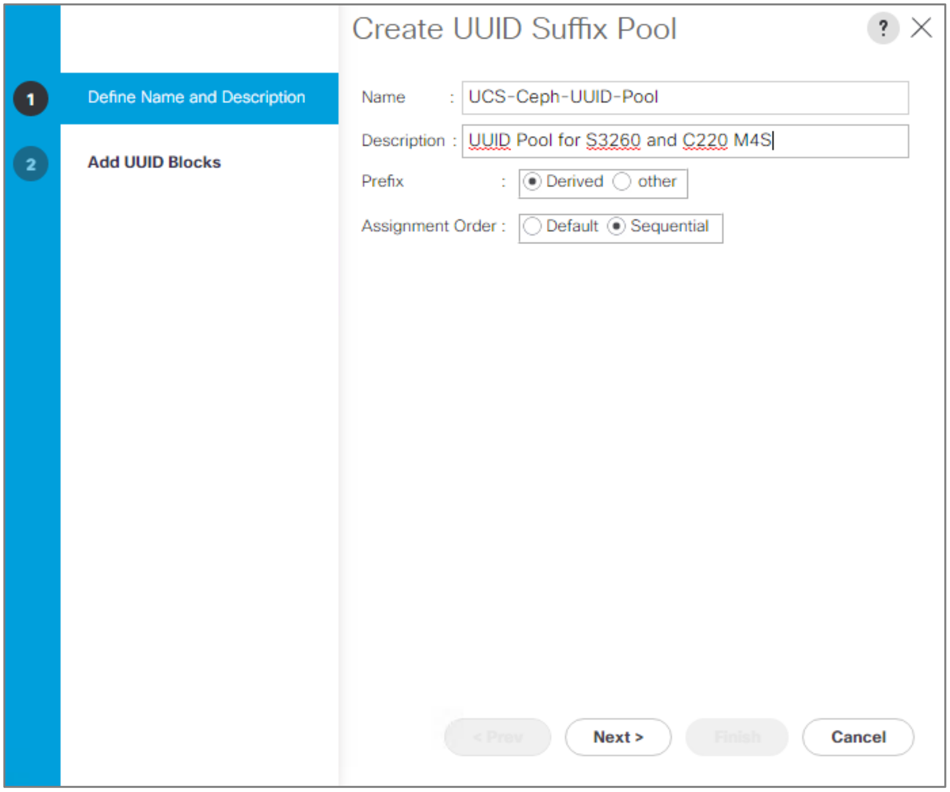
2. Go to Servers > Pools > root > UUID Suffix Pools and right-click Create UUID Suffix Pool.

3. Type in `UCS-Ceph-UUID-Pool` for Name.

4. (Optional) Enter a `Description` of the UUID Pool.

5. Set Assignment Order to Sequential and click Next.

Figure 24 Create UUID Suffix Pool



Create UUID Suffix Pool

1 Define Name and Description

2 Add UUID Blocks

Name : UCS-Ceph-UUID-Pool

Description : UUID Pool for S3260 and C220 M4S

Prefix : ● Derived ○ other

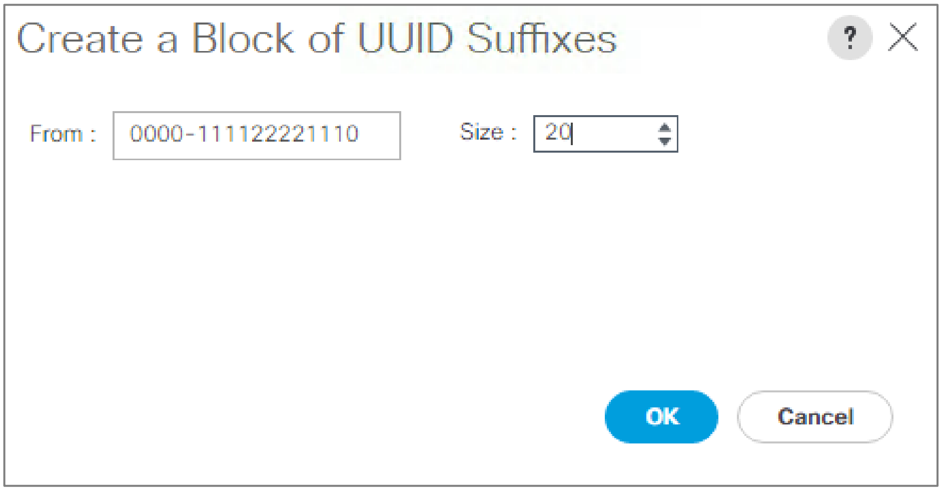
Assignment Order : ○ Default ● Sequential

6. Click `Add`.

7. Specify a starting UUID Suffix.

8. Specify a size of the UUID suffix pool, which is sufficient to support the available server resources, for example, 20.

Figure 25 Create a Block of UUID Suffixes



Create a Block of UUID Suffixes

From : 0000-111122221110

Size : 20

9. Click `OK`.

10. Click `Finish` and then `OK`.

Create Server Pool

A server pool can help you in speeding up your deployment of a Red Hat Ceph Storage environment. The pool identifies a group of servers with the same characteristics like Storage, CPU, Memory, etc. You can manually add server to a server pool or use server pool policies and server pool policy qualifications to automate the assignment.

To configure a server pool with the Cisco UCS Manager Pool GUI, complete the following steps:

1. Select the `Servers` Tab in the left pane in the Cisco UCS Manager GUI.

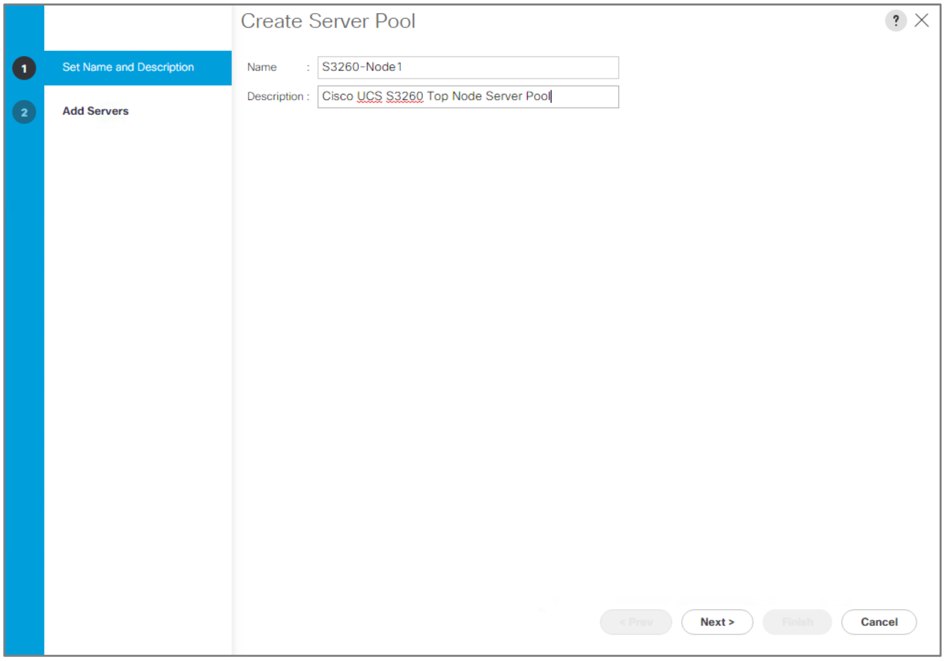
2. Go to Servers > Pools > root > Server Pools and right-click Create Server Pool.

3. Enter `S3260-Node1` in the `Name` field.

4. (Optional) Enter a description in the `Description` field.

5. Click `Next`.

Figure 26 Create Server for S3260 Top Node

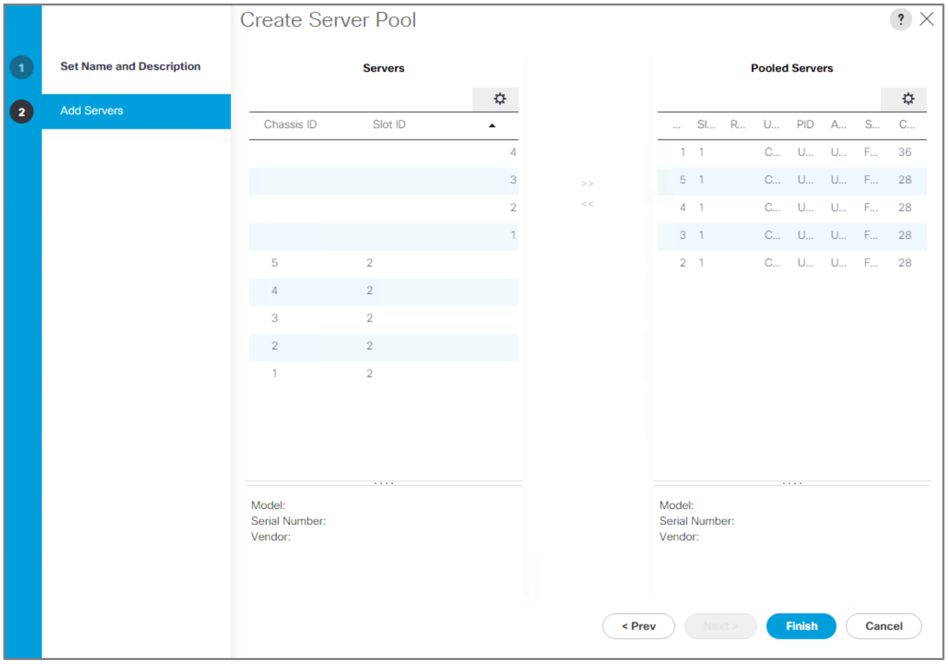


Name : S3260-Node1

Description : Cisco UCS S3260 Top Node Server Pool

6. Select all `Slot ID` 1 from all Cisco UCS S3260 Chassis and then click `>>` to add them to the pool.

Figure 27 Select all Slot ID 1 from all Cisco UCS S3260 Chassis



Servers

Chassis ID	Slot ID	
		4
		3
		2
		1
5	2	
4	2	
3	2	
2	2	
1	2	

Model:
Serial Number:
Vendor:

Pooled Servers

...	Sl...	R...	U...	PID	A...	S...	C...
1	1		C...	U...	U...	F...	36
5	1		C...	U...	U...	F...	28
4	1		C...	U...	U...	F...	28
3	1		C...	U...	U...	F...	28
2	1		C...	U...	U...	F...	28

Model:
Serial Number:
Vendor:

7. Click `Finish`.

8. Click `OK`.

9. Repeat the same steps for another pool with all `Slot ID` 2 from all Cisco UCS S3260 Chassis and name it `S3260-Node2`.

10. Create another server pool, which includes Rack-mount server 1-4 and name it `C220M4S`.

Create VLANs

As mentioned before it is important to separate the network traffic with VLANs for Public network traffic and Cluster network traffic. Table 10 shows the configured VLANs.

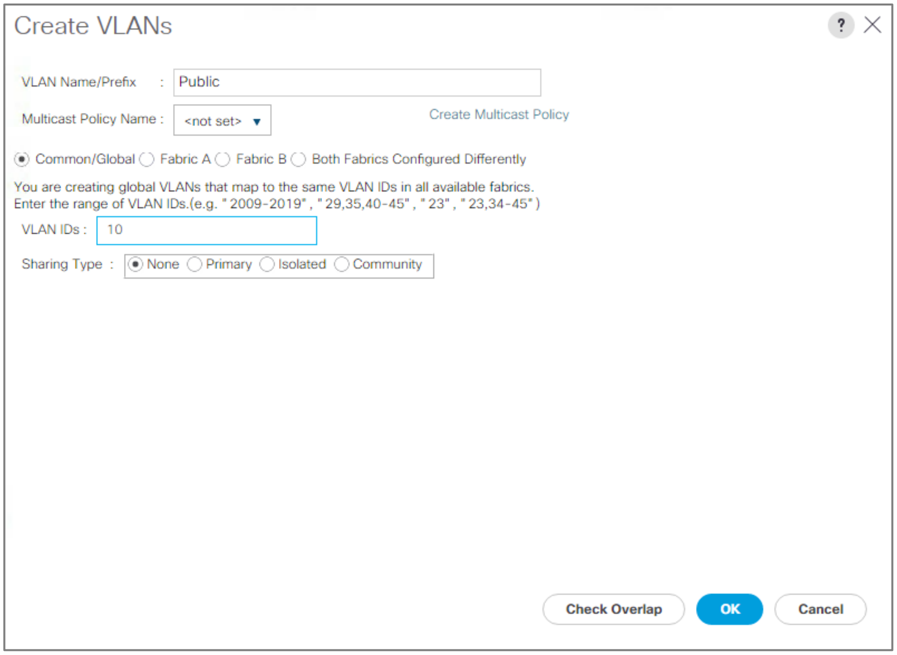
Table 10 VLAN Configurations

VLAN	Name	NIC Port	Function
1	default	eth0	Administration & Management
10	Public	eth1	Public network
20	Cluster	eth2	Cluster network

To configure VLANs in the Cisco UCS Manager GUI, complete the following steps:

1. Select `LAN` in the left pane in the UCSM GUI.
2. Select LAN > LAN Cloud > VLANs and right-click Create VLANs.
3. Enter `Public` for the `VLAN Name`.
4. Keep Multicast Policy Name as <not set>.
5. Select `Common/Global` for Public.
6. Enter 10 in the `VLAN IDs` field.
7. Click `OK` and then `Finish`.

Figure 28 Create a VLAN



Create VLANs

VLAN Name/Prefix : Public

Multicast Policy Name : <not set>

● 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 : 10

Sharing Type : ● None ○ Primary ○ Isolated ○ Community

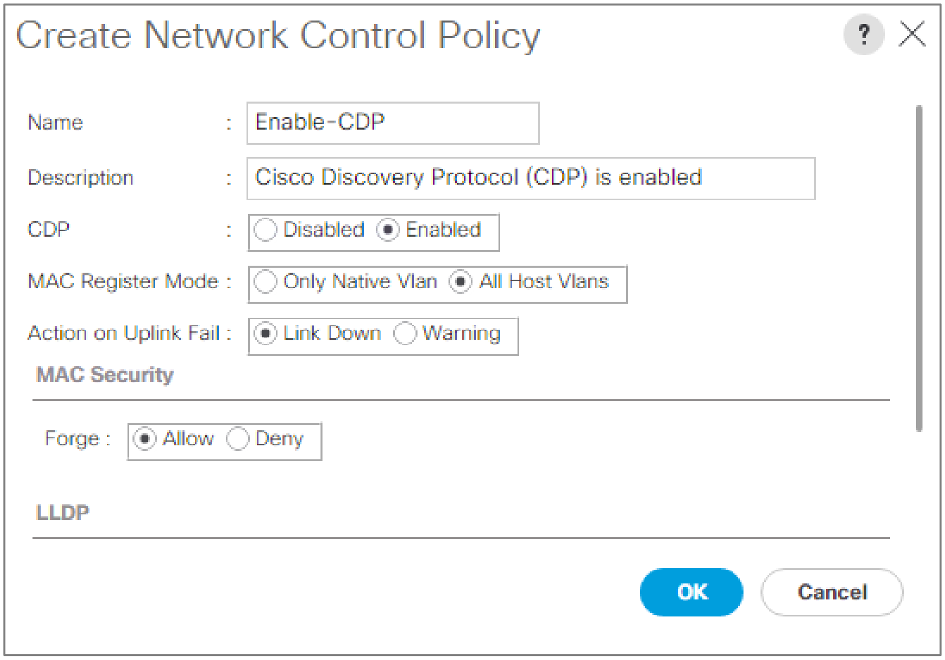
8. Repeat the steps for VLAN Cluster.

Enable CDP

To enable Network Control Policies, complete the following steps:

1. Select the `LAN` tab in the left pane of the Cisco UCS Manager GUI.
2. Go to LAN > Policies > root > Network Control Policies and right-click Create Network-Control Policy.
3. Type in `Enable-CDP` in the `Name` field.
4. (Optional) Enter a description in the `Description` field.
5. Click `Enabled` under `CDP`.
6. Click All Hosts Vlans under MAC Register Mode.
7. Leave everything else untouched and click `OK`.
8. Click `OK`.

Figure 29 Create a Network Control Policy



Create Network Control Policy

Name : Enable-CDP

Description : Cisco Discovery Protocol (CDP) is enabled

CDP : ○ Disabled ● Enabled

MAC Register Mode : ○ Only Native Vlan ● All Host Vlans

Action on Uplink Fail : ● Link Down ○ Warning

MAC Security

Forge : ● Allow ○ Deny

LLDP

QoS System Class

To create a Quality of Service System Class, complete the following steps:

1. Select the `LAN` tab in the left pane of the Cisco UCS Manager GUI.

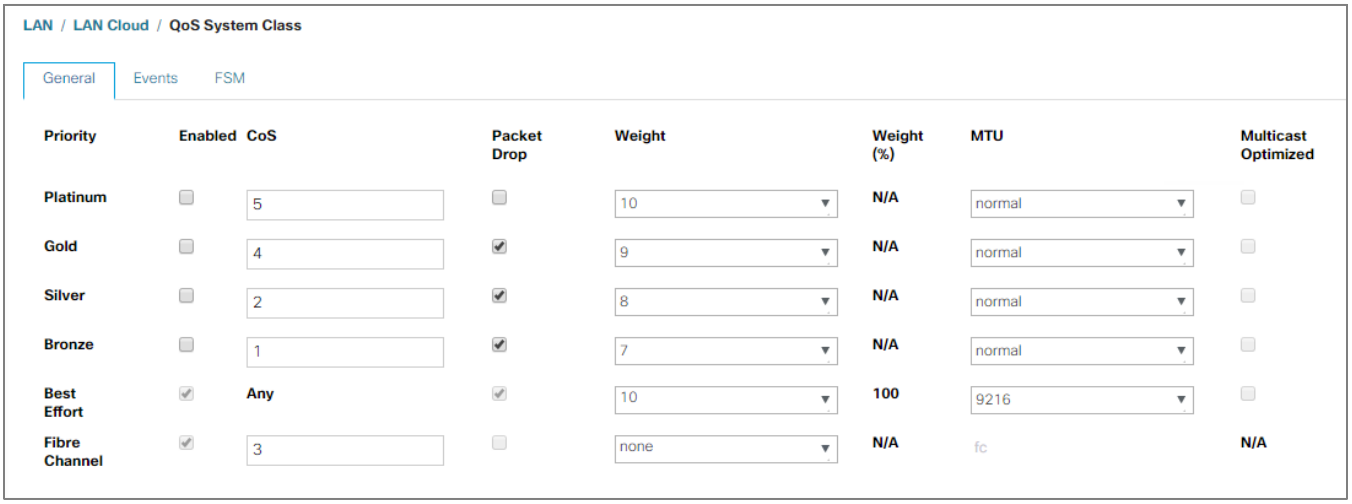
2. Go to LAN > LAN Cloud > QoS System Class.

3. Set `Best Effort Weight` to 10 and `MTU` to 9216.

4. Set Fibre Channel Weight to None.

5. Click `Save Changes` and then `OK`.

Figure 30 QoS System Class



LAN / LAN Cloud / QoS System Class

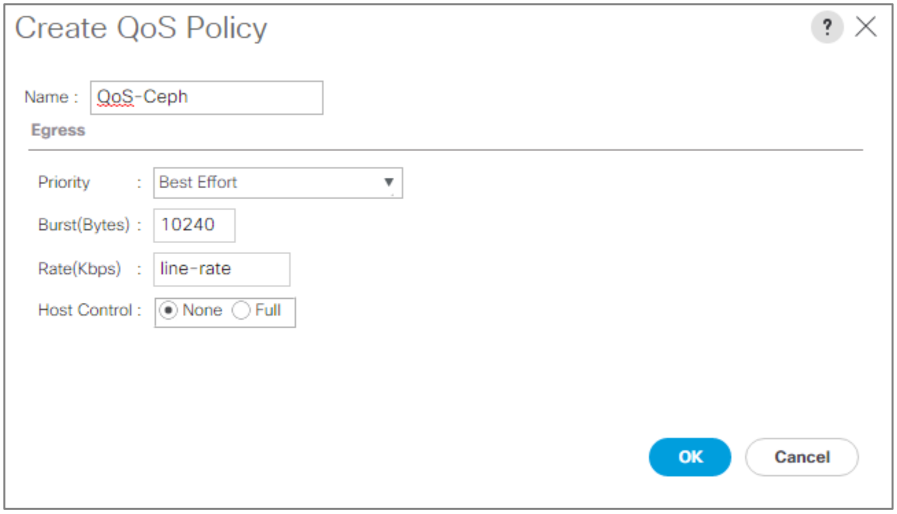
Priority	Enabled	CoS	Packet Drop	Weight	Weight (%)	MTU	Multicast Optimized
Platinum	□	5	□	10	N/A	normal	□
Gold	□	4	☑	9	N/A	normal	□
Silver	□	2	☑	8	N/A	normal	□
Bronze	□	1	☑	7	N/A	normal	□
Best Effort	☑	Any	☑	10	100	9216	□
Fibre Channel	☑	3	□	none	N/A	fc	N/A

QoS Policy Setup

Based on the previous QoS System Class, to setup a QoS Policy complete the following steps:

1. Select the `LAN` tab in the left pane of the Cisco UCS Manager GUI.
2. Go to LAN > Policies > root > QoS Policies and right-click Create QoS Policy.
3. Type in `QoS-Ceph` in the `Name` field.
4. Set `Priority` as `Best Effort` and leave everything else unchanged.
5. Click `OK` and then `OK`.

Figure 31 QoS Policy Setup



Create QoS Policy

Name : QoS-Ceph

Egress

Priority : Best Effort

Burst(Bytes) : 10240

Rate(Kbps) : line-rate

Host Control : ● None ○ Full

vNIC Template Setup

Based on the previous section of creating VLANs, the next step is to create the appropriate vNIC templates. For Red Hat Ceph Storage we need to create up four different vNICs, depending on the role of the server.

For the Public Network, please create two vNIC, one for the top node of the Cisco UCS S3260 Storage Server to connect to Fabric Interconnect A and one vNIC for the bottom node to connect to Fabric Interconnect B. This to avoid traffic congestion over the configured vPCs.

Table 11 gives you an overview of the configuration.

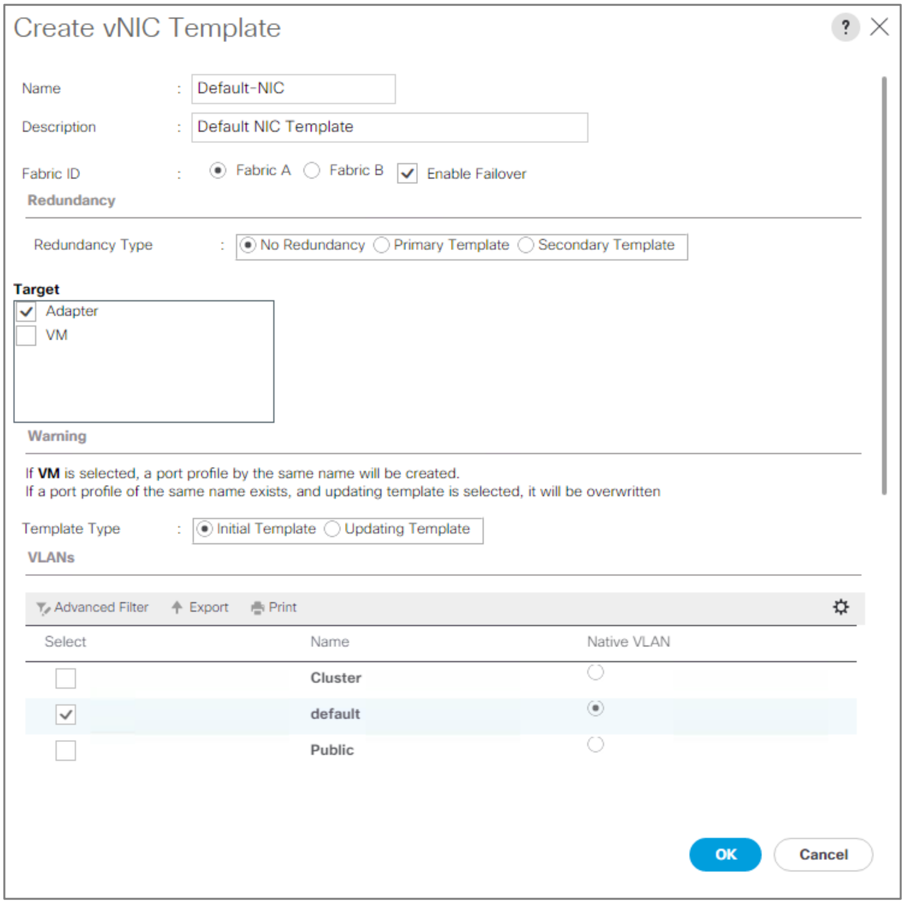
Table 11 vNIC Table

Name	vNIC Name	Fabric Interconnect	Failover	VLAN	MTU Size	MAC Pool	Network Control Policy
Default	Default-NIC	A	Yes	default - 1	1500	UCS-Ceph-MAC-Pool	Enable-CDP
Public Network	PublicA-NIC	A	Yes	Public - 10	9000	UCS-Ceph-MAC-Pool	Enable-CDP
	PublicB-NIC	B	Yes	Public - 10	9000	UCS-Ceph-MAC-Pool	Enable-CDP
Cluster Network	Cluster-NIC	B	Yes	Cluster - 20	9000	UCS-Ceph-MAC-Pool	Enable-CDP

To create the appropriate vNICs, complete the following steps:

1. Select the `LAN` tab in the left pane of the Cisco UCS Manager GUI.
2. Go to LAN > Policies > root > vNIC Templates and right-click Create vNIC Template.
3. Type in `Default-NIC` in the `Name` field.
4. (Optional) Enter a description in the `Description` field.
5. Click Fabric A as Fabric ID and enable failover.
6. Select `default` as `VLANs` and click `Native VLAN`.
7. Select `UCS-Ceph-MAC-Pool` as MAC Pool.
8. Select `QoS-Ceph` as QoS Policy.
9. Select Enable-CDP as Network Control Policy.
10. Click `OK` and then `OK`.

Figure 32 Setup of vNIC Template for Default vNIC



Create vNIC Template

Name : Default-NIC

Description : Default NIC Template

Fabric ID : ● Fabric A ○ Fabric B ☑ Enable Failover

Redundancy

Redundancy Type : ● No Redundancy ○ Primary Template ○ Secondary Template

Target

☑ Adapter

□ VM

Warning

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

Template Type : ● Initial Template ○ Updating Template

VLANs

Select	Name	Native VLAN
□	Cluster	○
☑	default	●
□	Public	○

11. Repeat the above steps for the vNICs Public and Cluster. Make sure you select the correct Fabric Interconnect, VLAN (without `Native VLAN`), and MTU size according to Table 11 .

Adapter Policy Setup

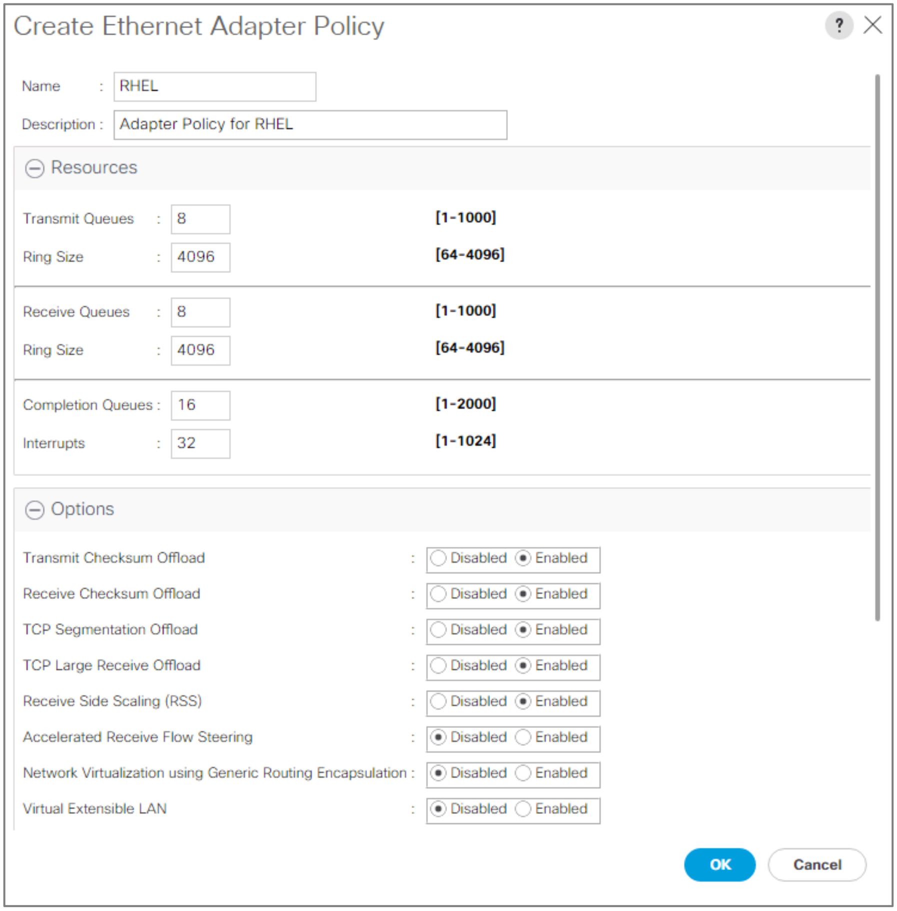
To create a specific adapter policy for Red Hat Enterprise Linux, complete the following steps:

1. Select the `Server` tab in the left pane of the Cisco UCS Manager GUI.
2. Go to Servers > Policies > root > Adapter Policies and right-click Create Ethernet Adapter Policy.
3. Type in `RHEL` in the `Name` field.
4. (Optional) Enter a description in the `Description` field.
5. Under `Resources` type in the following values:
   a. Transmit Queues: 8
   b. Ring Size: 4096
   c. Receive Queues: 8
   d. Ring Size: 4096
   e. Completion Queues: 16
   f. Interrupts: 32

6. Under Options enable Receive Side Scaling (RSS).

7. Click `OK` and then `OK`.

Figure 33 Adapter Policy for RHEL



Create Ethernet Adapter Policy

Name : RHEL

Description : Adapter Policy for RHEL

⊖ Resources

Transmit Queues :	8	[1-1000]
Ring Size :	4096	[64-4096]
Receive Queues :	8	[1-1000]
Ring Size :	4096	[64-4096]
Completion Queues :	16	[1-2000]
Interrupts :	32	[1-1024]

⊖ Options

Transmit Checksum Offload :	○ Disabled ● Enabled
Receive Checksum Offload :	○ Disabled ● Enabled
TCP Segmentation Offload :	○ Disabled ● Enabled
TCP Large Receive Offload :	○ Disabled ● Enabled
Receive Side Scaling (RSS) :	○ Disabled ● Enabled
Accelerated Receive Flow Steering :	● Disabled ○ Enabled
Network Virtualization using Generic Routing Encapsulation :	● Disabled ○ Enabled
Virtual Extensible LAN :	● Disabled ○ Enabled

BIOS Policy Setup

The BIOS policy feature in Cisco UCS automates the BIOS configuration process. The traditional method of setting the BIOS is manually, and is often error-prone. By creating a BIOS Policy and assigning the policy to a server or group of servers, can enable transparency within the BIOS settings configuration.

Note: BIOS settings can have a significant performance impact, depending on the workload and the applications. The BIOS settings listed in this section is for configurations optimized for best performance, which can be adjusted, based on the application, performance, and energy efficiency requirements.

To create a server BIOS policy using the Cisco UCS Manager GUI, complete the following steps:

1. Select the `Servers` tab in the left pane in the Cisco UCS Manager GUI.

2. Select Policies > root > BIOS Policies and right-click Create BIOS Policy.

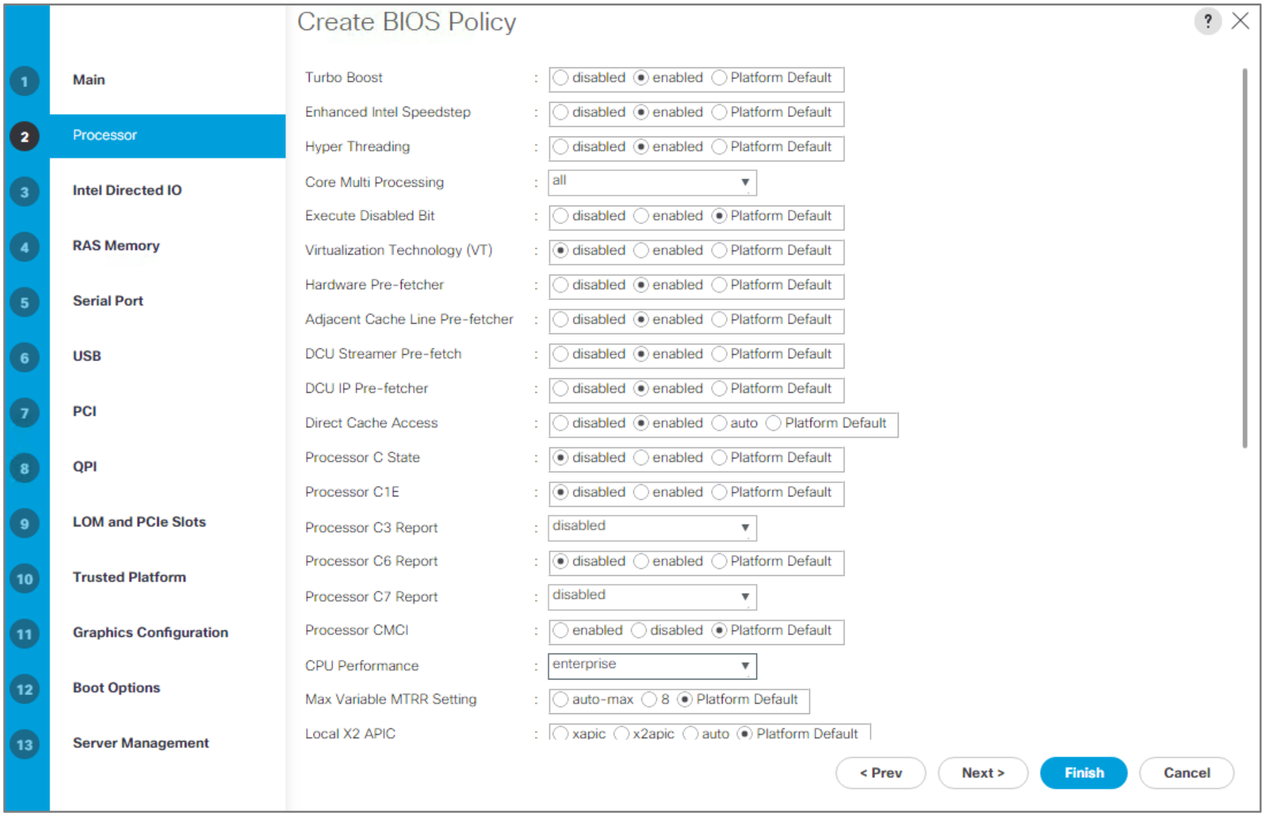
3. Type in `Ceph` in the `Name` field.

4. (Optional) Enter a description in the `Description` field.

5. Change the BIOS settings as shown in the following figures.

6. The only changes that need to be made are in the Processor (Figure 34) and RAS Memory (Figure 35) settings.

Figure 34 Processor Settings



Create BIOS Policy

Turbo Boost : ○ disabled ● enabled ○ Platform Default

Enhanced Intel Speedstep : ○ disabled ● enabled ○ Platform Default

Hyper Threading : ○ disabled ● enabled ○ Platform Default

Core Multi Processing : all ▼

Execute Disabled Bit : ○ disabled ○ enabled ● Platform Default

Virtualization Technology (VT) : ● disabled ○ enabled ○ Platform Default

Hardware Pre-fetcher : ○ disabled ● enabled ○ Platform Default

Adjacent Cache Line Pre-fetcher : ○ disabled ● enabled ○ Platform Default

DCU Streamer Pre-fetch : ○ disabled ● enabled ○ Platform Default

DCU IP Pre-fetcher : ○ disabled ● enabled ○ Platform Default

Direct Cache Access : ○ disabled ● enabled ○ auto ○ Platform Default

Processor C State : ● disabled ○ enabled ○ Platform Default

Processor C1E : ● disabled ○ enabled ○ Platform Default

Processor C3 Report : disabled ▼

Processor C6 Report : ● disabled ○ enabled ○ Platform Default

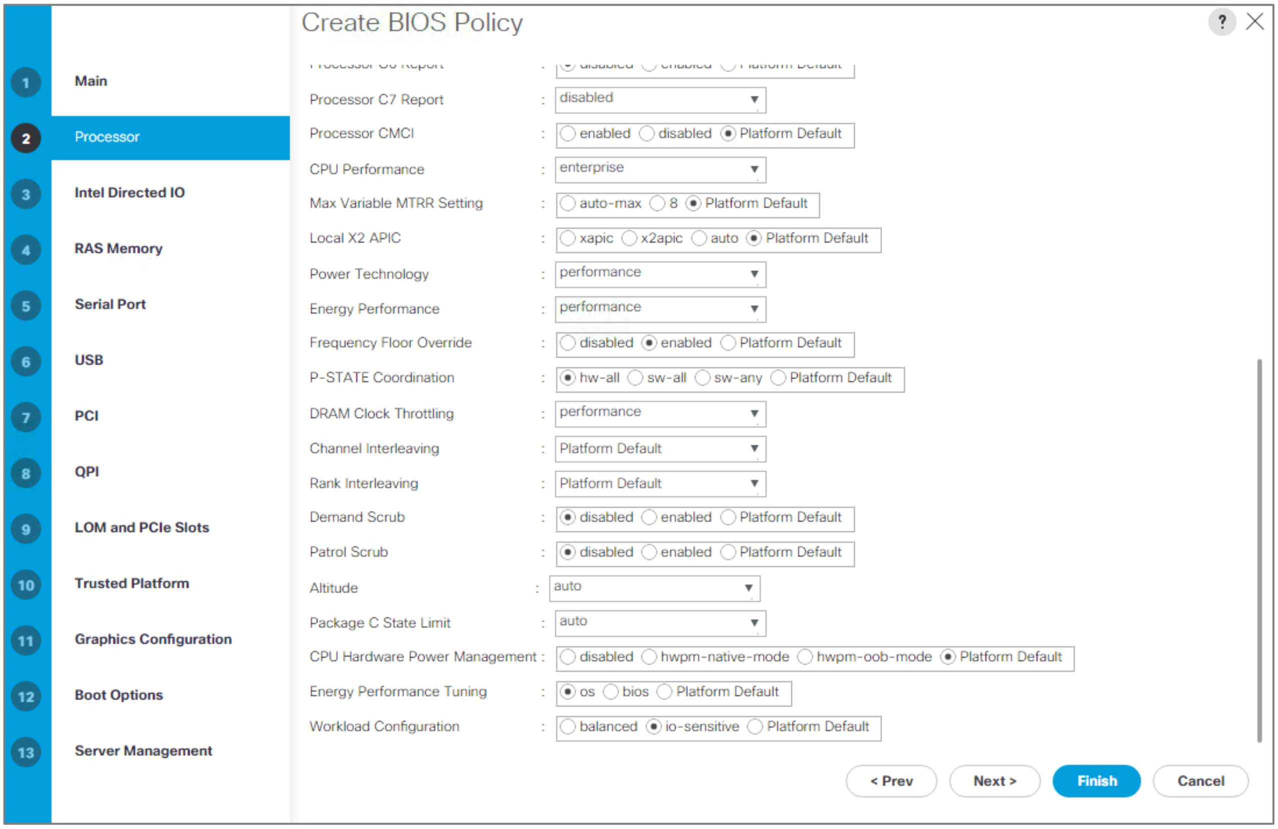
Processor C7 Report : disabled ▼

Processor CMCI : ○ enabled ○ disabled ● Platform Default

CPU Performance : enterprise ▼

Max Variable MTRR Setting : ○ auto-max ○ 8 ● Platform Default

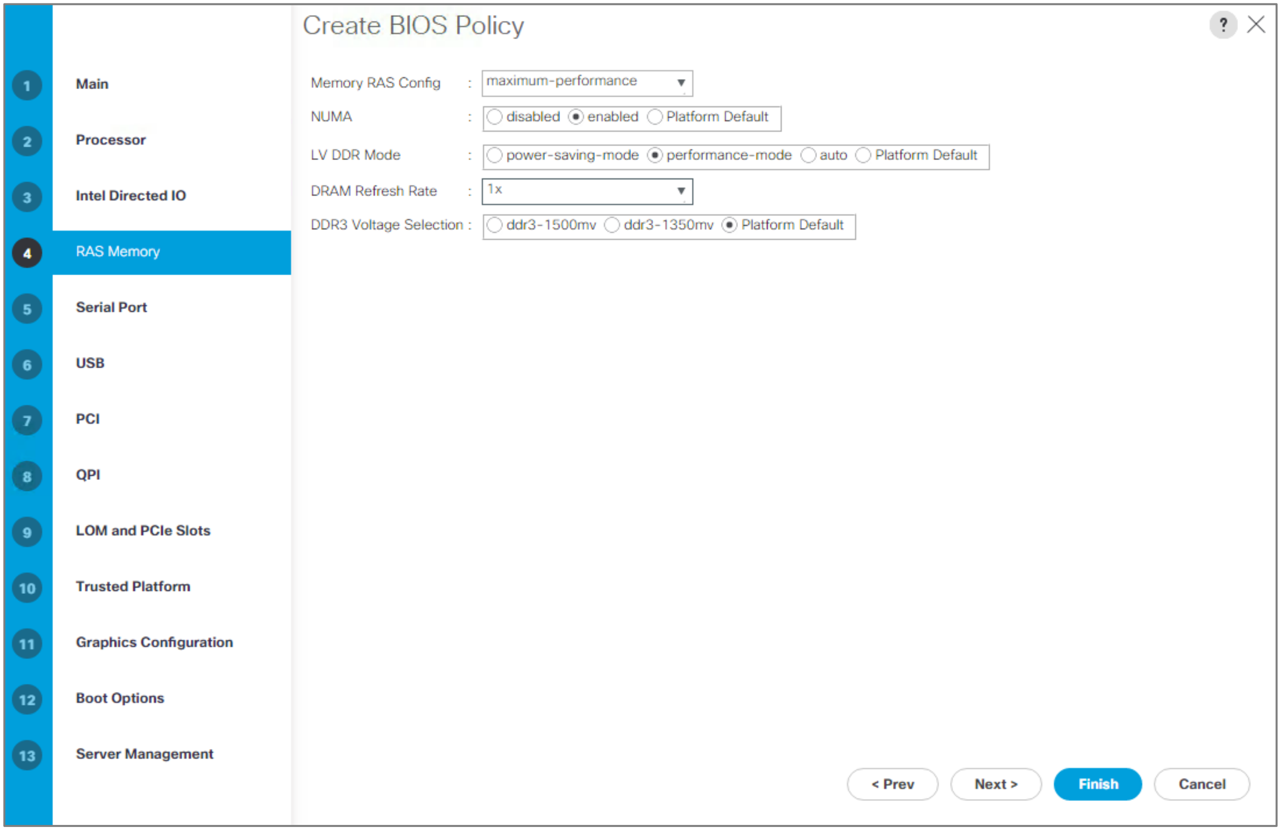
Local X2 APIC : ○ xapic ○ x2apic ○ auto ● Platform Default



Create BIOS Policy

Processor C6 Report : ● disabled ○ enabled ○ Platform Default
Processor C7 Report : disabled ▾
Processor CMCI : ○ enabled ○ disabled ● Platform Default
CPU Performance : enterprise ▾
Max Variable MTRR Setting : ○ auto-max ○ 8 ● Platform Default
Local X2 APIC : ○ xapic ○ x2apic ○ auto ● Platform Default
Power Technology : performance ▾
Energy Performance : performance ▾
Frequency Floor Override : ○ disabled ● enabled ○ Platform Default
P-STATE Coordination : ● hw-all ○ sw-all ○ sw-any ○ Platform Default
DRAM Clock Throttling : performance ▾
Channel Interleaving : Platform Default ▾
Rank Interleaving : Platform Default ▾
Demand Scrub : ● disabled ○ enabled ○ Platform Default
Patrol Scrub : ● disabled ○ enabled ○ Platform Default
Altitude : auto ▾
Package C State Limit : auto ▾
CPU Hardware Power Management : ○ disabled ○ hwpm-native-mode ○ hwpm-oob-mode ● Platform Default
Energy Performance Tuning : ● os ○ bios ○ Platform Default
Workload Configuration : ○ balanced ● io-sensitive ○ Platform Default

Figure 35 RAS Memory Settings



Create BIOS Policy

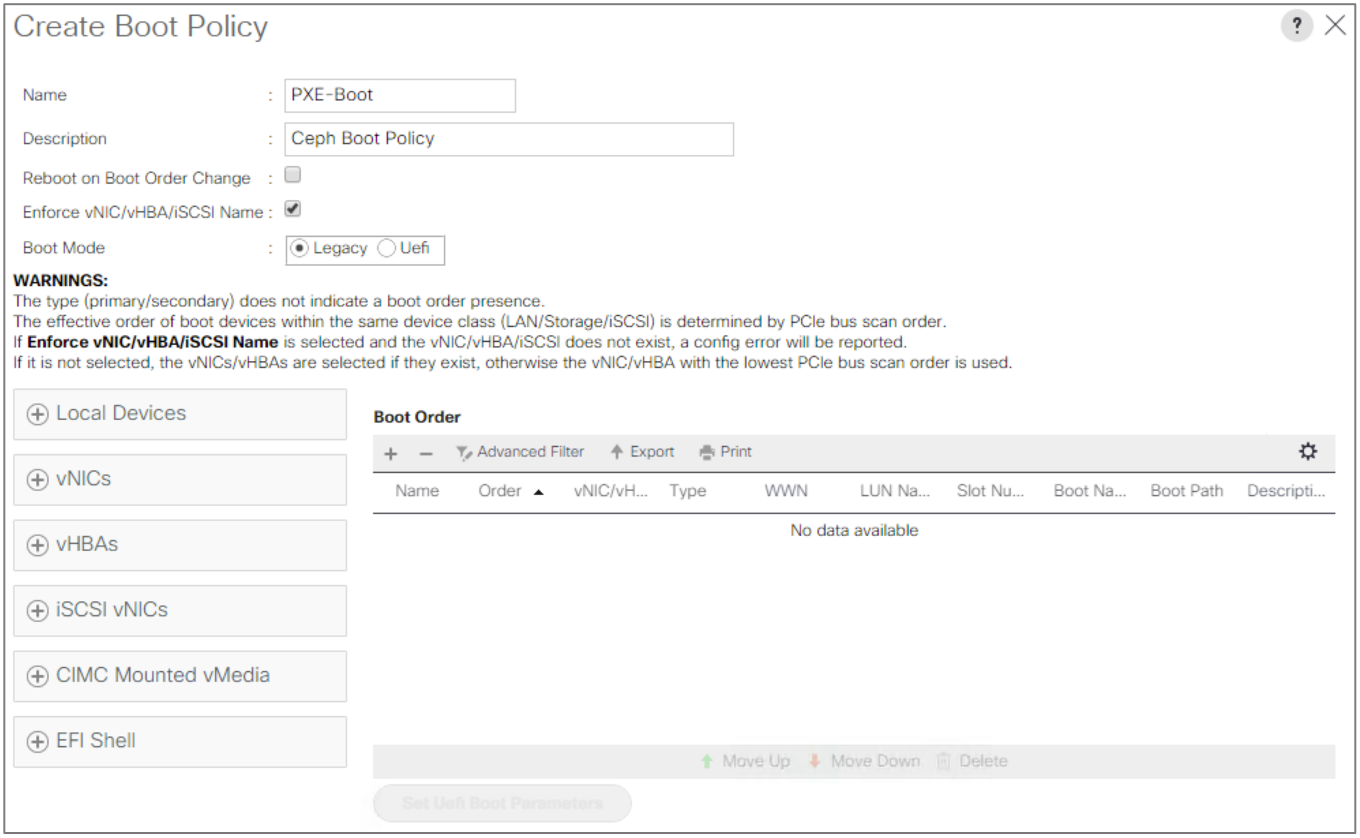
Memory RAS Config : maximum-performance ▾
NUMA : ○ disabled ● enabled ○ Platform Default
LV DDR Mode : ○ power-saving-mode ● performance-mode ○ auto ○ Platform Default
DRAM Refresh Rate : 1x ▾
DDR3 Voltage Selection : ○ ddr3-1500mv ○ ddr3-1350mv ● Platform Default

Boot Policy Setup

To create a Boot Policy, complete the following steps:

1. Select the `Servers` tab in the left pane.
2. Go to Servers > Policies > root > Boot Policies and right-click Create Boot Policy.
3. Type in a `PXE-Boot` in the `Name` field.
4. (Optional) Enter a description in the `Description` field.

Figure 36 Create Boot Policy



Create Boot Policy

Name : PXE-Boot

Description : Ceph Boot Policy

Reboot on Boot Order Change : □

Enforce vNIC/vHBA/iSCSI Name : ☑

Boot Mode : ● Legacy ○ Uefi

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

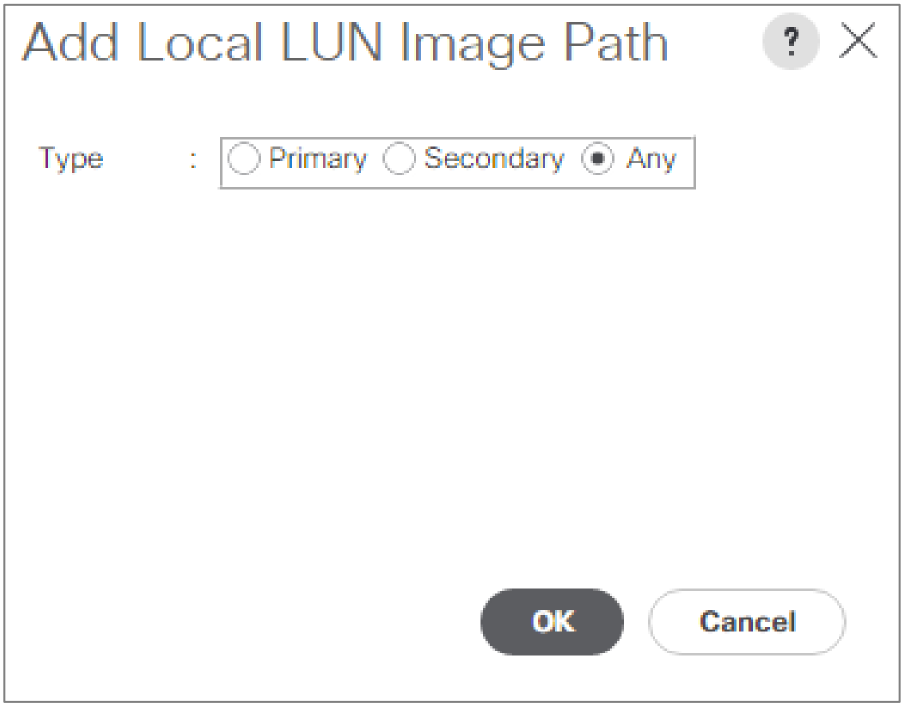
Boot Order

Name	Order ▲	vNIC/vH...	Type	WWN	LUN Na...	Slot Nu...	Boot Na...	Boot Path	Descripti...

No data available

5. Click Local Devices > Add Local LUN.

Figure 37 Add Local LUN



Add Local LUN Image Path

Type : ○ Primary ○ Secondary ● Any

6. Click `OK`.

7. Click Local Devices > Add Local CD/DVD.

8. Click `OK`.

9. Click `OK`.

Create Maintenance Policy Setup

To setup a Maintenance Policy, complete the following steps :

1. Select the `Servers` tab in the left pane.

2. Go to Servers > Policies > root > Maintenance Policies and right-click Create Maintenance Policy.

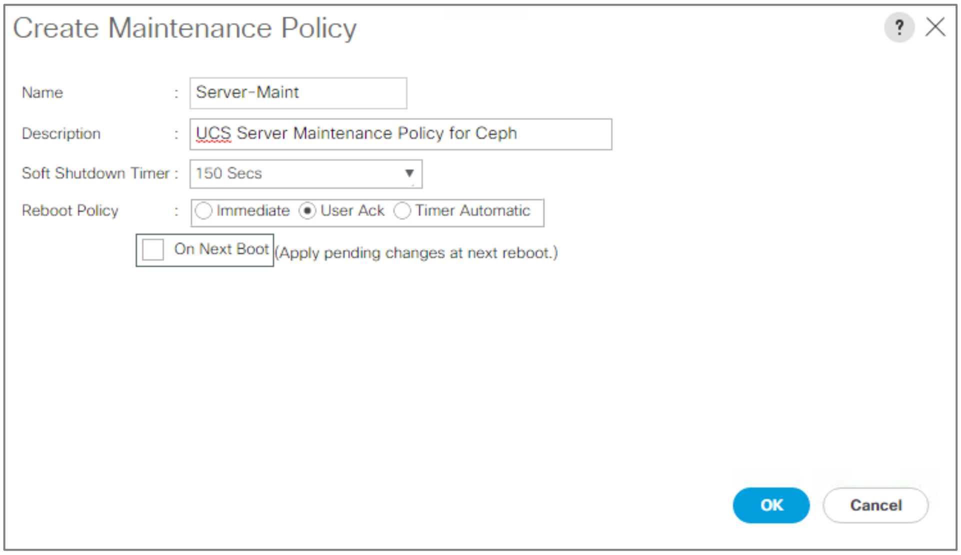
3. Type in a `Server-Maint` in the `Name` field.

4. (Optional) Enter a description in the `Description` field.

5. Click `User Ack` under Reboot Policy.

6. Click `OK` and then `OK`.

Figure 38 Create Maintenance Policy



Create Maintenance Policy

Name : Server-Maint

Description : UCS Server Maintenance Policy for Ceph

Soft Shutdown Timer : 150 Secs

Reboot Policy : ○ Immediate ● User Ack ○ Timer Automatic

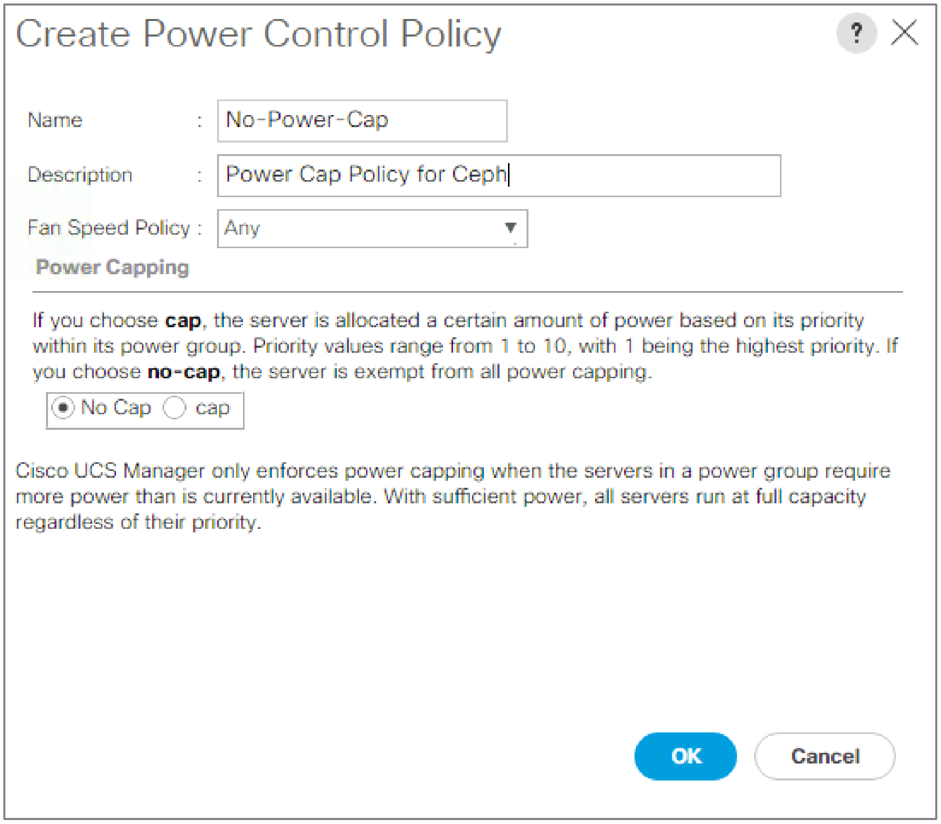
□ On Next Boot (Apply pending changes at next reboot.)

Create Power Control Policy Setup

To create a Power Control Policy, complete the following steps:

1. Select the `Servers` tab in the left pane.
2. Go to Servers > Policies > root > Power Control Policies and right-click Create Power Control Policy.
3. Type in `No-Power-Cap` in the `Name` field.
4. (Optional) Enter a description in the `Description` field.
5. Click `No Cap`.
6. Click `OK` and then `OK`.

Figure 39 Power Control Policy



Create Power Control Policy

Name : No-Power-Cap

Description : Power Cap Policy for Ceph

Fan Speed Policy : Any

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.

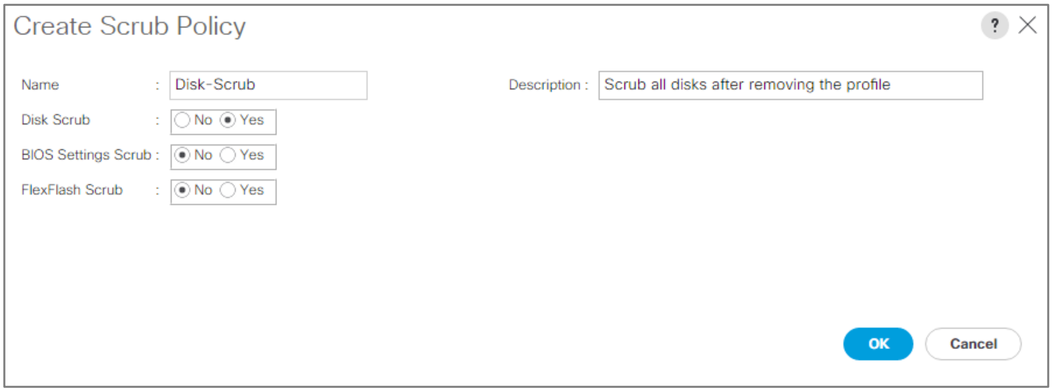
Create Disk Scrub Policy

To prevent failures during re-deployment of a Red Hat Ceph Storage environment, implement a Disk Scrub Policy that is enabled when removing a profile from a server.

To create a Disk Scrub Policy, complete the following steps:

1. Select the `Servers` tab in the left pane.
2. Go to Servers > Policies > root > Scrub Policies and right-click Create Scrub Policy.
3. Type in `Disk-Scrub` in the `Name` field.
4. (Optional) Enter a description in the `Description` field.
5. Select `Disk Scrub` radio button to `Yes`.
6. Click `OK` and then `OK`.

Figure 40 Create a Disk Scrub Policy


Create Scrub Policy

Name : Disk-Scrub

Description : Scrub all disks after removing the profile

Disk Scrub : ○ No ● Yes

BIOS Settings Scrub : ● No ○ Yes

FlexFlash Scrub : ● No ○ Yes

Creating Chassis Profile

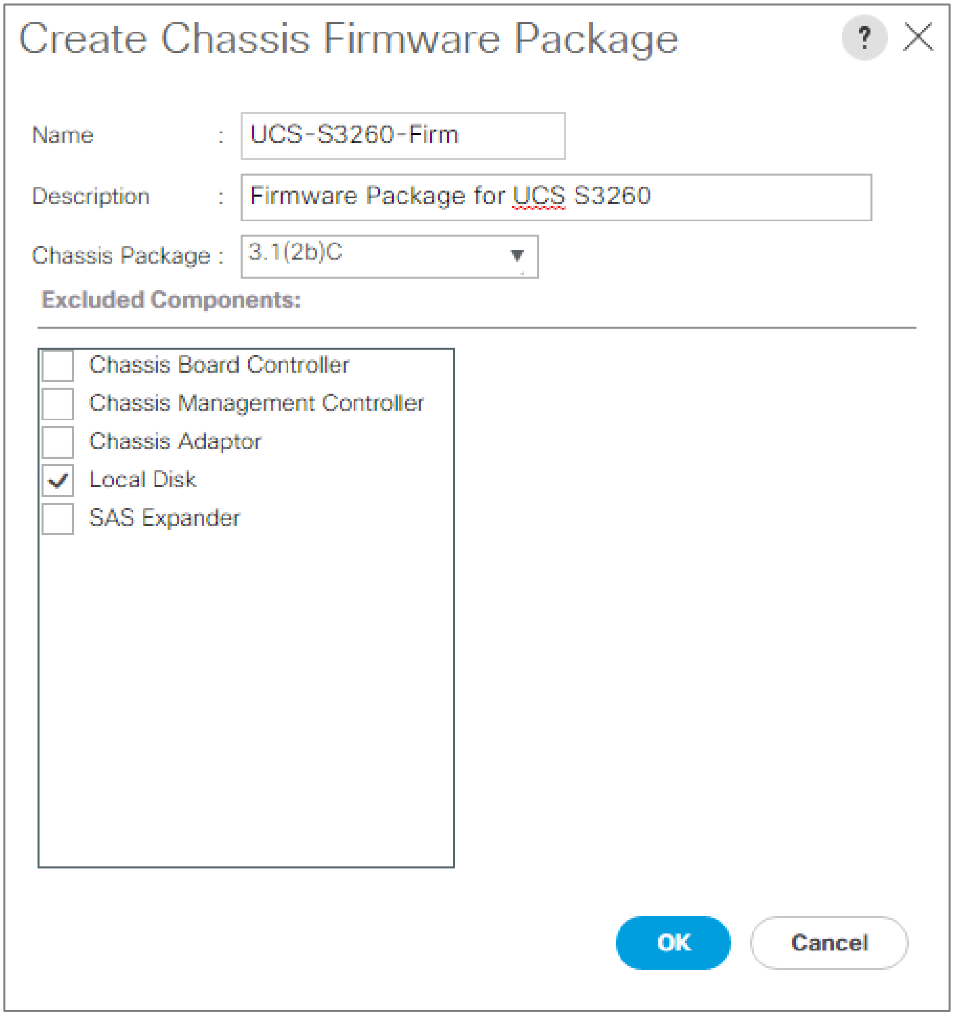
The Chassis Profile is required to assign specific disks to a particular server node in a Cisco UCS S3260 Storage Server as well as upgrading to a specific chassis firmware package.

Create Chassis Firmware Package

To create a Chassis Firmware Package, complete the following steps:

1. Select the `Chassis` tab in the left pane of the Cisco UCS Manager GUI.
2. Go to Chassis > Policies > root > Chassis Firmware Package and right-click Create Chassis Firmware Package.
3. Type in `UCS-S3260-Firm` in the `Name` field.
4. (Optional) Enter a description in the `Description` field.
5. Select `3.1.(2b)C` form the drop-down menu of `Chassis Package`.
6. Select `OK` and then `OK`.

Figure 41 Create Chassis Firmware Package



Create Chassis Firmware Package

Name : UCS-S3260-Firm

Description : Firmware Package for UCS S3260

Chassis Package : 3.1(2b)C

Excluded Components:

- □ Chassis Board Controller
- □ Chassis Management Controller
- □ Chassis Adaptor
- ☑ Local Disk
- □ SAS Expander

Create Chassis Maintenance Policy

To create a Chassis Maintenance Policy, complete the following steps:

1. Select the `Chassis` tab in the left pane of the Cisco UCS Manager GUI.

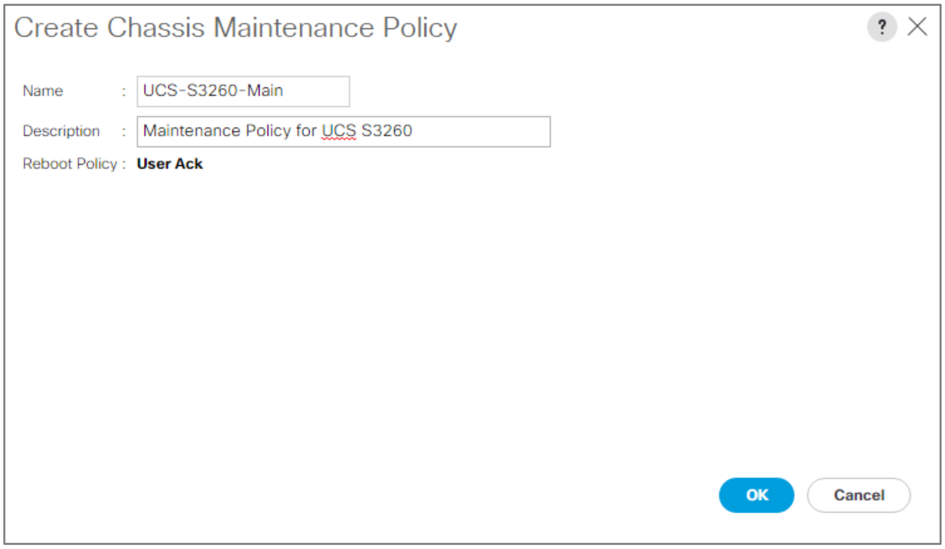
2. Go to Chassis > Policies > root > Chassis Maintenance Policies and right-click Create Chassis Maintenance Policy.

3. Type in `UCS-S3260-Main` in the `Name` field.

4. (Optional) Enter a description in the `Description` field.

5. Click `OK` and then `OK`.

Figure 42 Create Chassis Maintenance Policy


Create Chassis Maintenance Policy

Name : UCS-S3260-Main

Description : Maintenance Policy for UCS S3260

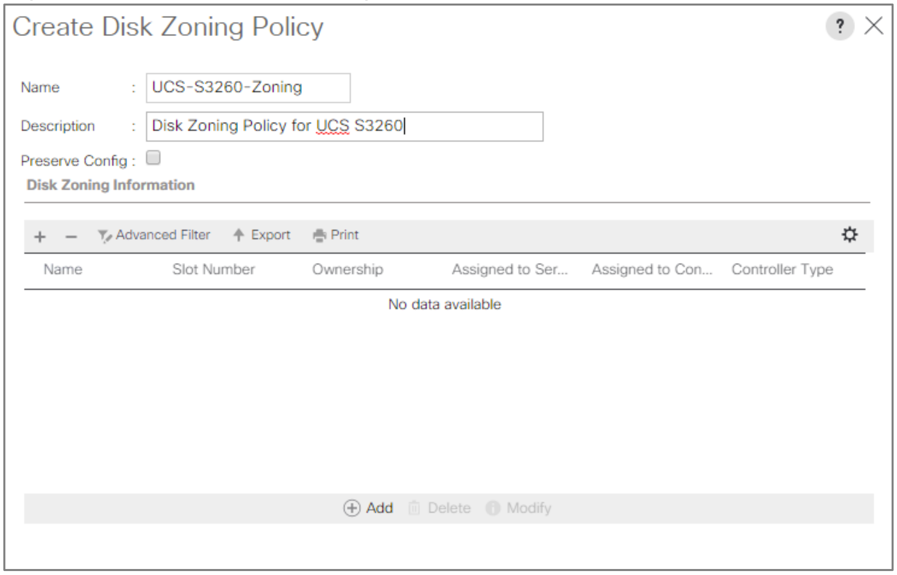
Reboot Policy : **User Ack**

Create Disk Zoning Policy

To create a Disk Zoning Policy, follow these steps:

1. Select the `Chassis` tab in the left pane of the Cisco UCS Manager GUI.
2. Go to Chassis > Policies > root > Disk Zoning Policies and right-click Create Disk Zoning Policy.
3. Type in `UCS-S3260-Zoning` in the `Name` field.
4. (Optional) Enter a description in the `Description` field.

Figure 43 Create Disk Zoning Policy


Create Disk Zoning Policy

Name : UCS-S3260-Zoning

Description : Disk Zoning Policy for UCS S3260

Preserve Config : □

Disk Zoning Information

Name	Slot Number	Ownership	Assigned to Ser...	Assigned to Con...	Controller Type

No data available

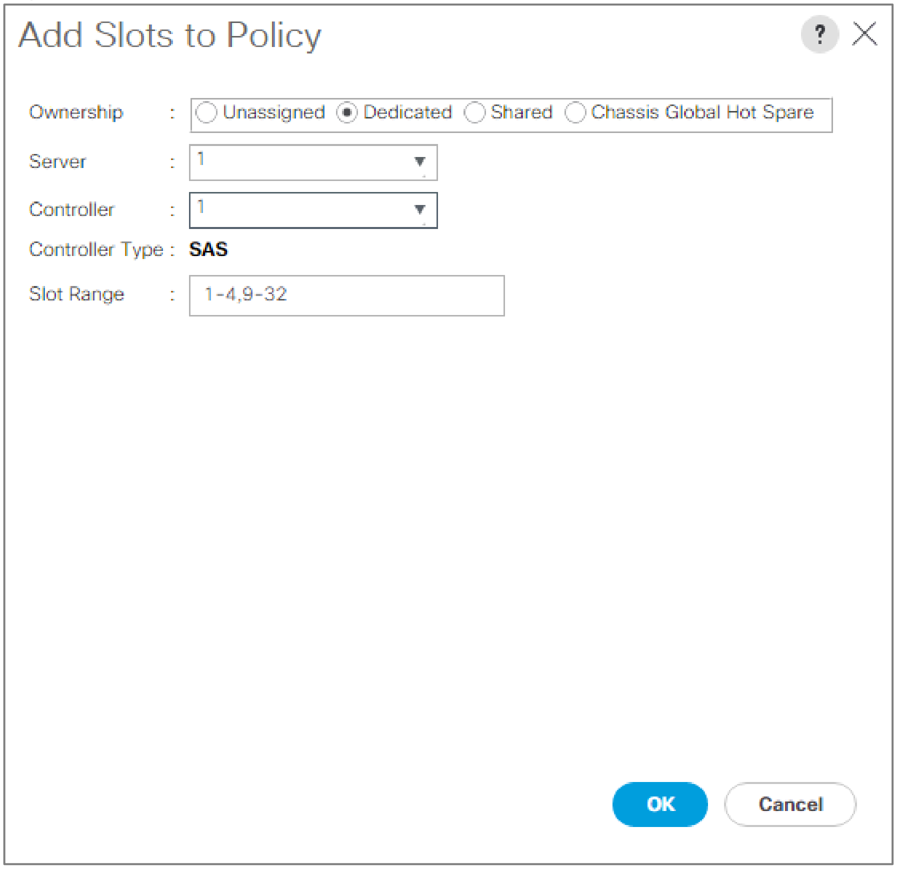
5. Click `Add`.
6. Select Dedicated under Ownership.

7. Select `Server` 1.

8. Select `Controller` 1.

9. Add `Slot Range` `1-4, 9-32` for the top node of the Cisco UCS S3260 Storage Server.

Figure 44 Add Slots to Top Node of Cisco UCS S3260



Add Slots to Policy

Ownership : ○ Unassigned ● Dedicated ○ Shared ○ Chassis Global Hot Spare

Server : 1

Controller : 1

Controller Type : SAS

Slot Range : 1-4,9-32

10. Click `OK`.

11. Click `Add`.

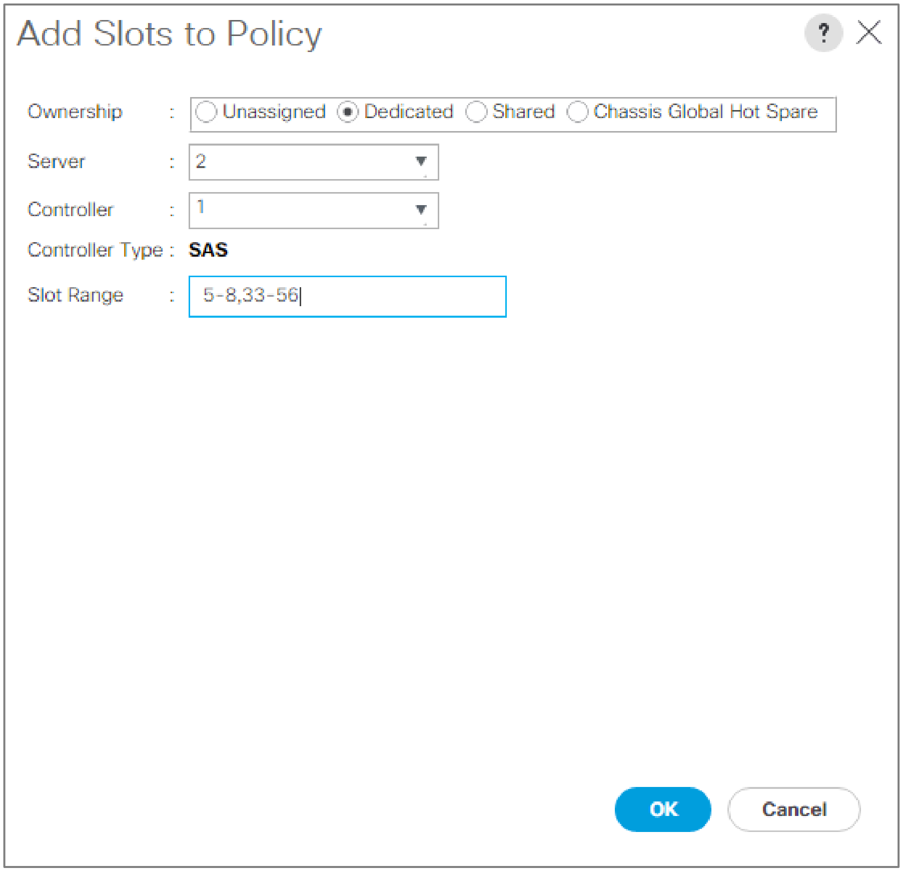
12. Select Dedicated under Ownership.

13. Select `Server` 2.

14. Select `Controller` 1.

15. Add `Slot Range` `5-8, 33-56` for the bottom node of the Cisco UCS S3260 Storage Server.

Figure 45 Add Slots to Bottom Node of Cisco UCS S3260



Add Slots to Policy

Ownership : ○ Unassigned ● Dedicated ○ Shared ○ Chassis Global Hot Spare

Server : 2

Controller : 1

Controller Type : **SAS**

Slot Range : 5-8,33-56

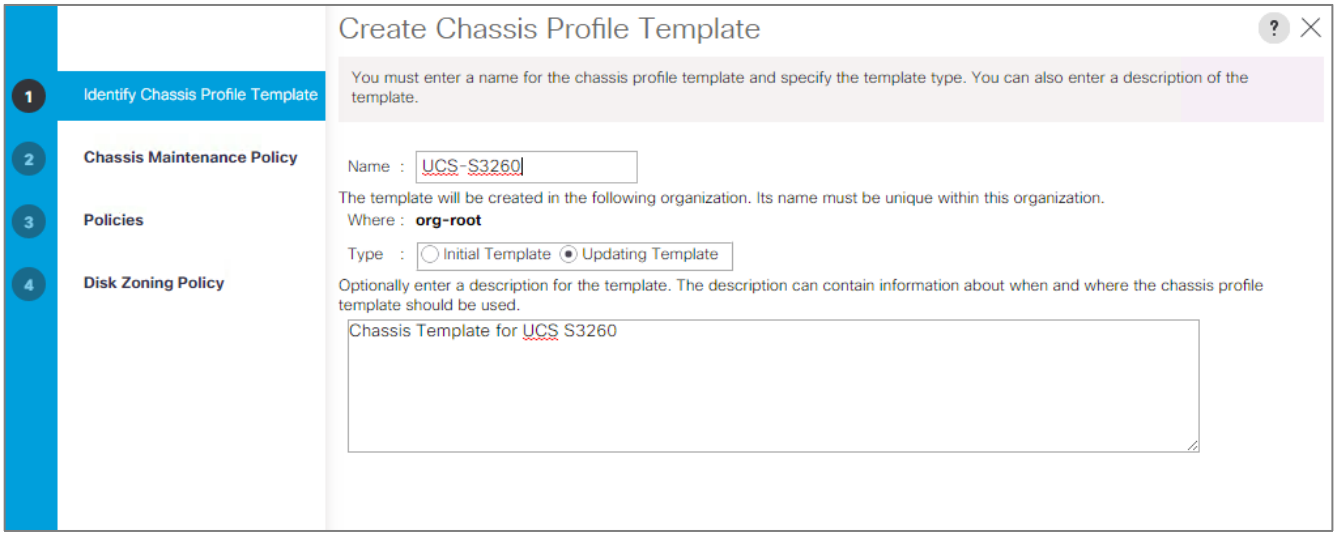
16. Click `OK` and then `OK`.

Create Chassis Profile Template

To create a Chassis Profile Template, complete the following steps:

1. Select the `Chassis` tab in the left pane of the Cisco UCS Manager GUI.
2. Go to Chassis > Chassis Profile Templates and right-click Create Chassis Profile Template.
3. Type in `UCS-S3260` in the `Name` field.
4. Under Type, select Updating Template.
5. (Optional) Enter a description in the `Description` field.

Figure 46 Create Chassis Profile Template



Create Chassis Profile Template

You must enter a name for the chassis profile template and specify the template type. You can also enter a description of the template.

1 Identify Chassis Profile Template
2 Chassis Maintenance Policy
3 Policies
4 Disk Zoning Policy

Name : UCS-S3260

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

Where : **org-root**

Type : ○ Initial Template ● Updating Template

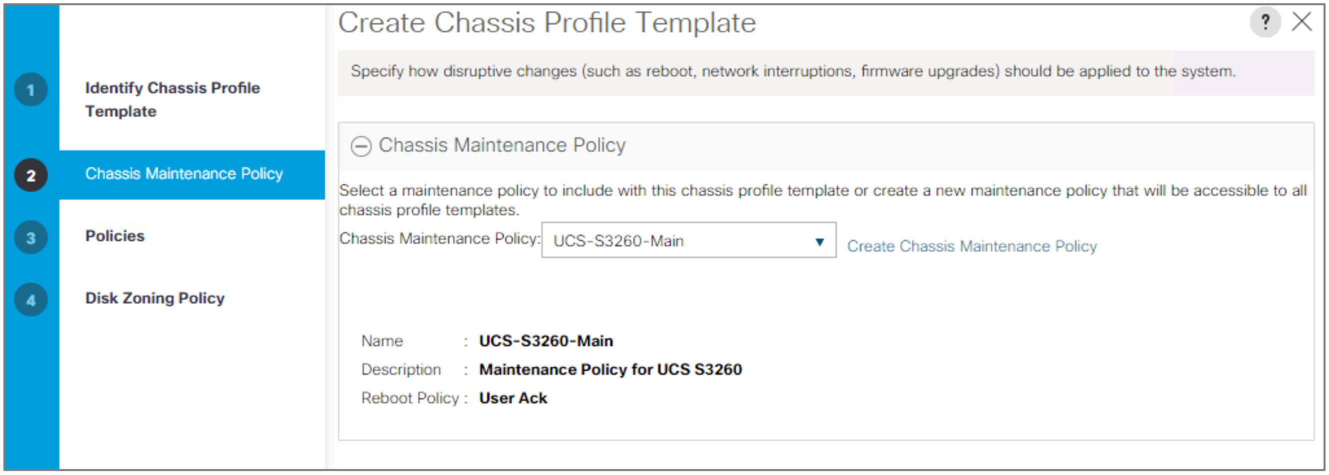
Optionally enter a description for the template. The description can contain information about when and where the chassis profile template should be used.

Chassis Template for UCS S3260

6. Select Next.

7. Under the radio button Chassis Maintenance Policy, select your previously created Chassis Maintenance Policy.

Figure 47 Chassis Profile Template – Chassis Maintenance Policy



Create Chassis Profile Template

Specify how disruptive changes (such as reboot, network interruptions, firmware upgrades) should be applied to the system.

1 Identify Chassis Profile Template
2 Chassis Maintenance Policy
3 Policies
4 Disk Zoning Policy

⊖ Chassis Maintenance Policy

Select a maintenance policy to include with this chassis profile template or create a new maintenance policy that will be accessible to all chassis profile templates.

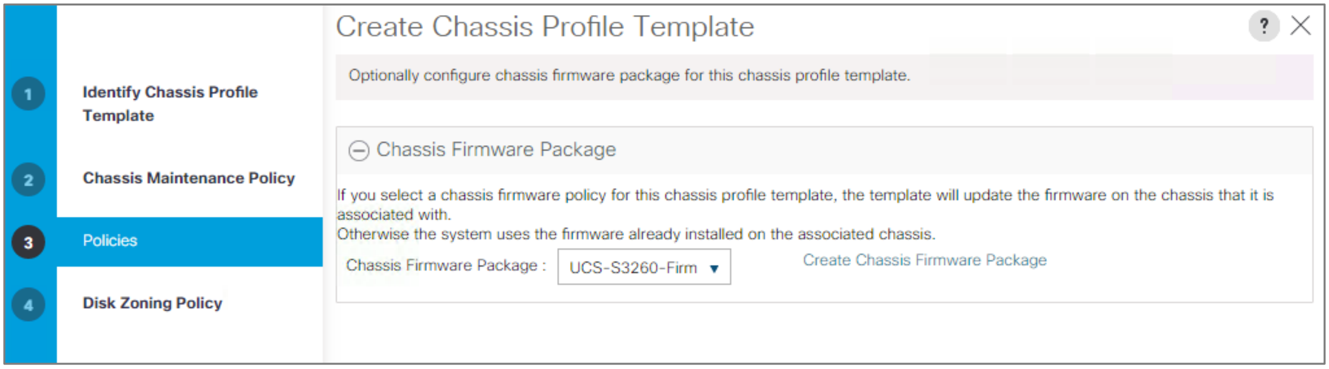
Chassis Maintenance Policy: UCS-S3260-Main ▾ Create Chassis Maintenance Policy

Name : **UCS-S3260-Main**
Description : **Maintenance Policy for UCS S3260**
Reboot Policy : **User Ack**

8. Select Next.

9. Select the + button and select under Chassis Firmware Package your previously created Chassis Firmware Package Policy.

Figure 48 Chassis Profile Template – Chassis Firmware Package



1 Identify Chassis Profile Template
2 Chassis Maintenance Policy
3 Policies
4 Disk Zoning Policy

Create Chassis Profile Template

Optionally configure chassis firmware package for this chassis profile template.

Chassis Firmware Package

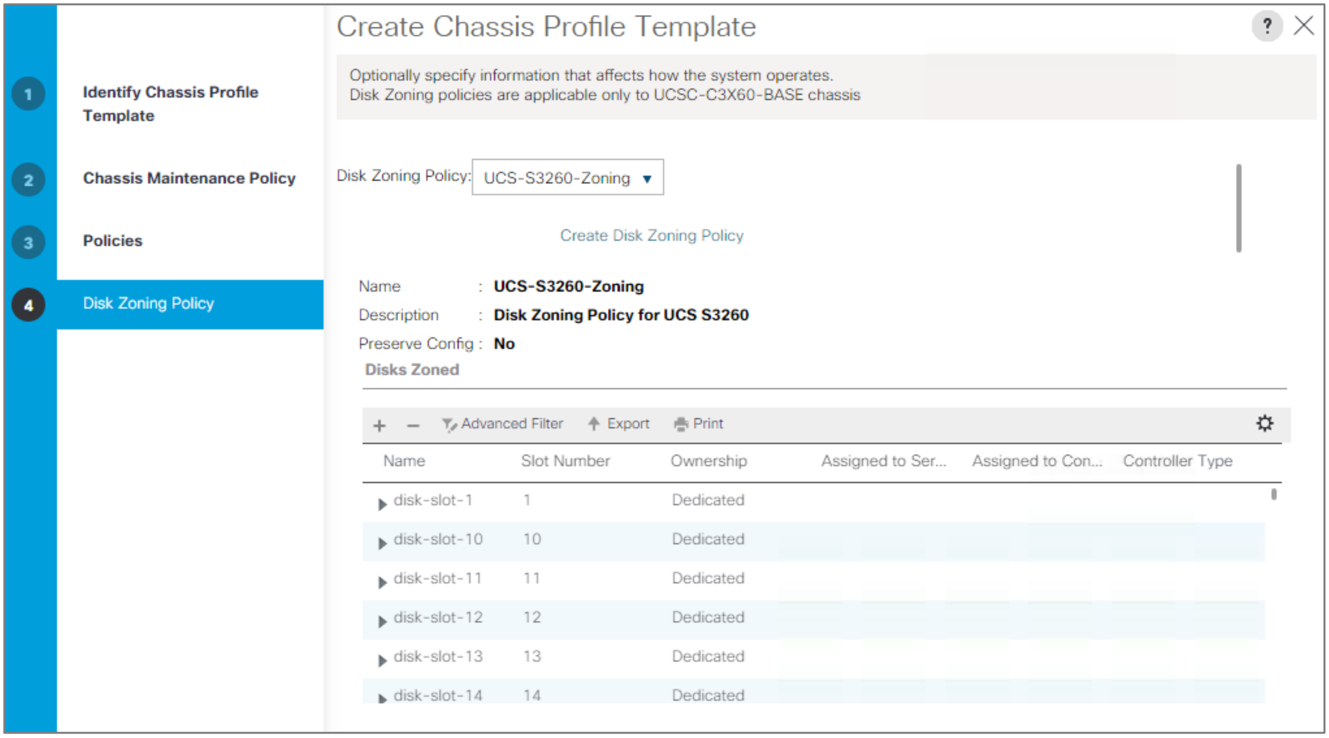
If you select a chassis firmware policy for this chassis profile template, the template will update the firmware on the chassis that it is associated with.
Otherwise the system uses the firmware already installed on the associated chassis.

Chassis Firmware Package : UCS-S3260-Firm

10. Select Next.

11. Under `Disk Zoning Policy` select your previously created Disk Zoning Policy.

Figure 49 Chassis Profile Template – Disk Zoning Policy



1 Identify Chassis Profile Template
2 Chassis Maintenance Policy
3 Policies
4 Disk Zoning Policy

Create Chassis Profile Template

Optionally specify information that affects how the system operates.
Disk Zoning policies are applicable only to UCSC-C3X60-BASE chassis

Disk Zoning Policy: UCS-S3260-Zoning

Create Disk Zoning Policy

Name : **UCS-S3260-Zoning**
Description : **Disk Zoning Policy for UCS S3260**
Preserve Config : **No**

Disks Zoned

+ – Advanced Filter Export Print

Name	Slot Number	Ownership	Assigned to Ser...	Assigned to Con...	Controller Type
disk-slot-1	1	Dedicated			
disk-slot-10	10	Dedicated			
disk-slot-11	11	Dedicated			
disk-slot-12	12	Dedicated			
disk-slot-13	13	Dedicated			
disk-slot-14	14	Dedicated			

12. Click `Finish` and then `OK`.

Create Chassis Profile from Template

To create the Chassis Profiles from the previous created Chassis Profile Template, complete the following steps:

1. Select the `Chassis` tab in the left pane of the Cisco UCS Manager GUI.

2. Go to Chassis > Chassis Profiles and right-click Create Chassis Profiles from Template.

3. Type in `S3260-Dual-` in the `Name` field.

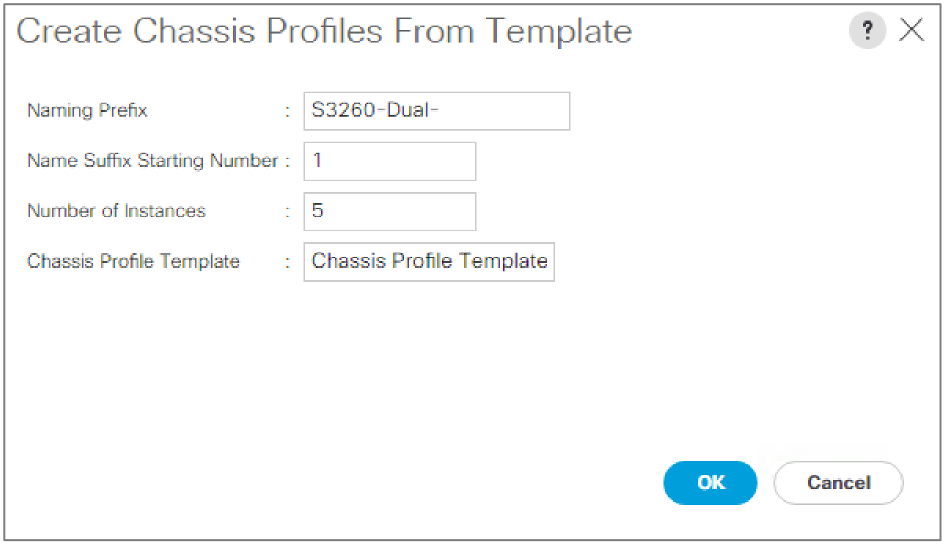
4. Leave the Name Suffix Starting Number untouched.

5. Enter `5` for the `Number of Instances` for all connected Cisco UCS S3260 Storage Server.

6. Choose your previously created `Chassis Profile Template`.

7. Click `OK` and then `OK`.

Figure 50 Create Chassis Profiles from Template



Create Chassis Profiles From Template

Naming Prefix :	S3260-Dual-
Name Suffix Starting Number :	1
Number of Instances :	5
Chassis Profile Template :	Chassis Profile Template

Associate Chassis Profile

To associate all previous created Chassis Profile, complete the following steps:

1. Select the `Chassis` tab in the left pane of the Cisco UCS Manager GUI.

2. Go to Chassis > Chassis Profiles and select S3260-Dual-1.

3. Right-click Change Chassis Profile Association.

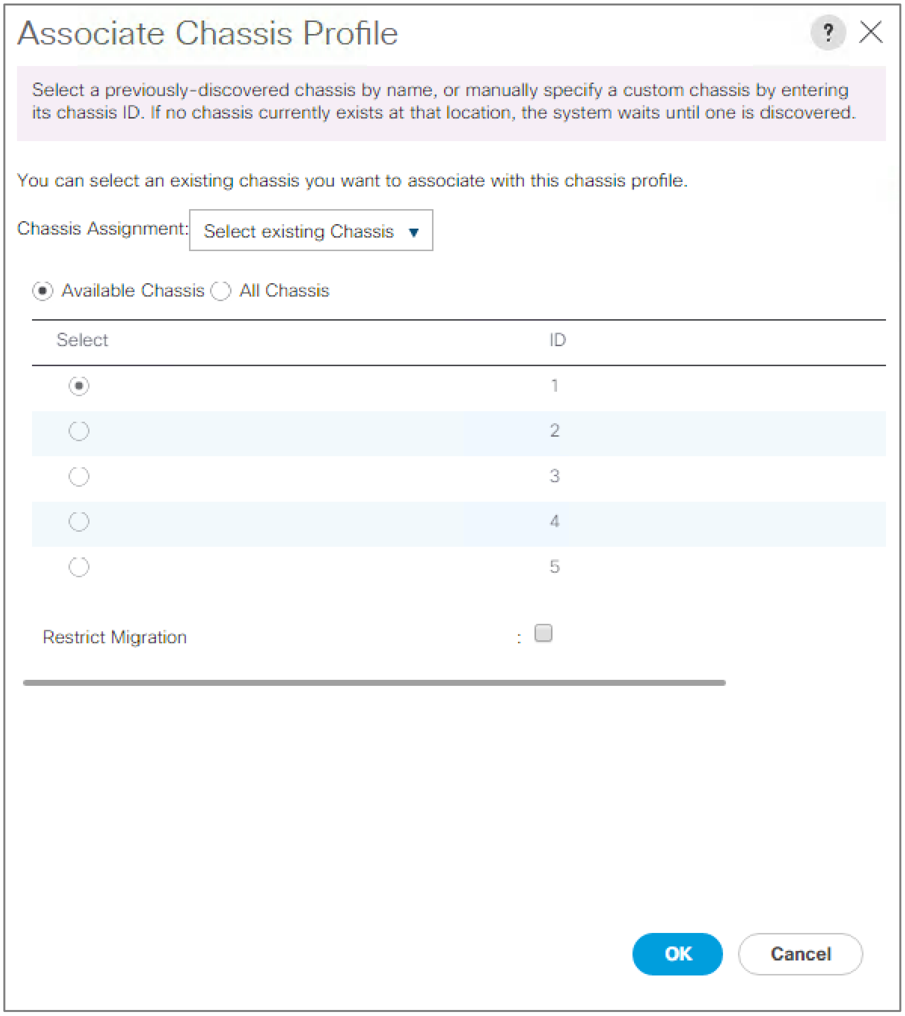
4. Under Chassis Assignment, choose Select existing Chassis.

5. Under `Available Chassis`, select ID `1`.

6. Click `OK` and then `OK`.

7. Repeat the steps for the other four Chassis Profiles by selecting the IDs 2 – 5.

Figure 51 Associate Chassis Profile



Associate Chassis Profile

Select a previously-discovered chassis by name, or manually specify a custom chassis by entering its chassis ID. If no chassis currently exists at that location, the system waits until one is discovered.

You can select an existing chassis you want to associate with this chassis profile.

Chassis Assignment: Select existing Chassis ▼

● Available Chassis ○ All Chassis

Select	ID
●	1
○	2
○	3
○	4
○	5

Restrict Migration : □

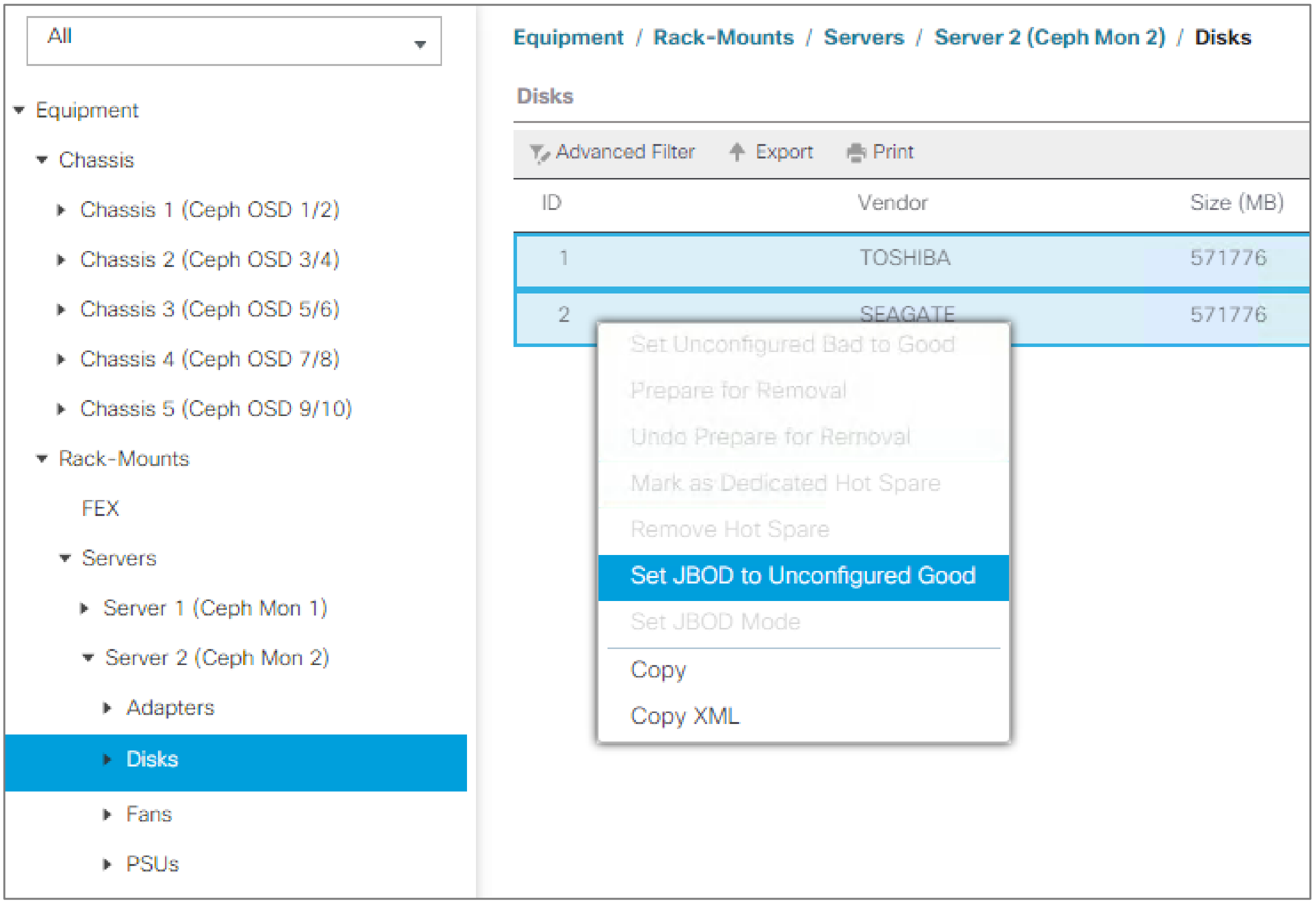
Creating Storage Profiles

Setting Disks for Rack-Mount Servers to Unconfigured-Good

To prepare all disks from the Rack-Mount servers for storage profiles, the disks have to be converted from JBOD to Unconfigured-Good. To convert the disks, complete the following steps:

1. Select the `Equipment` tab in the left pane of the Cisco UCS Manager GUI.
2. Go to Equipment > Rack-Mounts > Servers > Server 1 > Disks.
3. Select both disks and right-click `Set JBOD to Unconfigured-Good`.
4. Repeat the steps for Server 2-4.

Figure 52 Set Disks for Rack-Mount Servers to Unconfigured-Good



ID	Vendor	Size (MB)
1	TOSHIBA	571776
2	SEAGATE	571776

Setting Disks for Cisco UCS S3260 Storage Server to Unconfigured-Good with Cisco UCS PowerTool

To convert all top-loaded HDDs and the back-end Boot-SSDs in all five attached Cisco UCS S3260 Storage Server from JBOD to Unconfigured-Good, use a UCS PowerTool Script, which accelerates and simplifies the deployment.

To convert all top-loaded HDDs, complete the following steps:

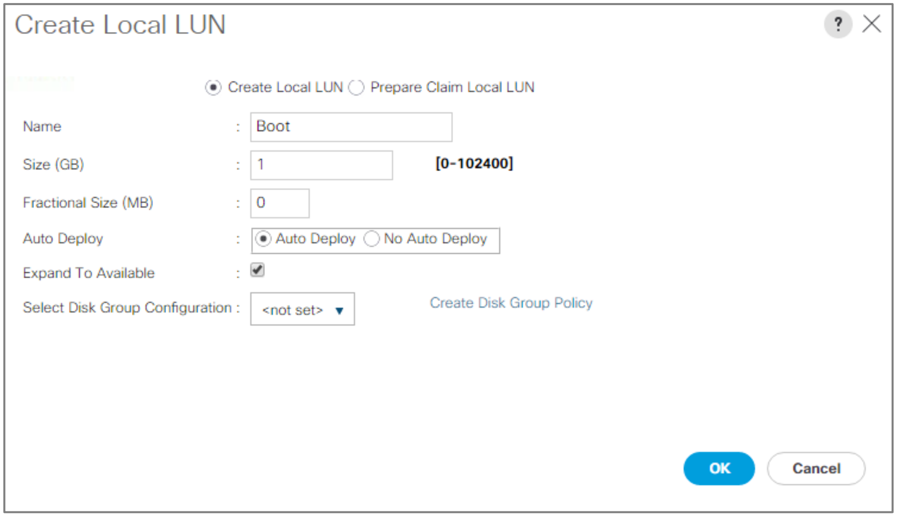
1. Go to https://communities.cisco.com/docs/DOC-37154 and download the latest UCS PowerTool Suite.
2. Install UCS PowerTool Suite on a Windows system that has access to the Cisco UCS Manager GUI.
3. Download the UCS PowerTool Script “Convert all disks to Unconfigured-Good for UCS Domain(s)” under https://communities.cisco.com/docs/DOC-70616
4. Start a PowerShell CLI and start the script.
5. Type in the Cluster IP of your Cisco UCS Manager.
6. Type in your password.
7. Type in `Y` for converting your disks from JBOD to Unconfigured-Good.

Create Storage Profile for Cisco UCS S3260 Storage Server

To create the Storage Profile for the top node of the Cisco UCS S3260 Storage Server, complete the following steps:

1. Select `Storage` in the left pane of the Cisco UCS Manager GUI.
2. Go to Storage > Storage Profiles and right-click Create Storage Profile.
3. Type in `S3260-Node1` in the `Name` field.
4. (Optional) Enter a description in the `Description` field.
5. Click `Add`.
6. Type in `Boot` in the `Name` field.
7. Configure as follow:
    a. Create Local LUN
    b. Size (GB) = 1
    c. Fractional Size (MB) = 0
    d. Auto Deploy
    e. Select Expand To Available
8. Click Create Disk Group Policy

Figure 53 Create Local LUN



Create Local LUN

● Create Local LUN ○ Prepare Claim Local LUN

Name : Boot

Size (GB) : 1 [0-102400]

Fractional Size (MB) : 0

Auto Deploy : ● Auto Deploy ○ No Auto Deploy

Expand To Available : ☑

Select Disk Group Configuration : <not set> ▼

9. Type in `S3260-Boot` in the `Name` field.
10. (Optional) Enter a description in the `Description` field.
11. RAID Level = RAID 1 Mirrored.
12. Select Disk Group Configuration (Manual).

13. Click `Add`.

14. Type in `201` for `Slot Number`.

15. Click `OK` and then again `Add`.

16. Type in `202` for `Slot Number`.

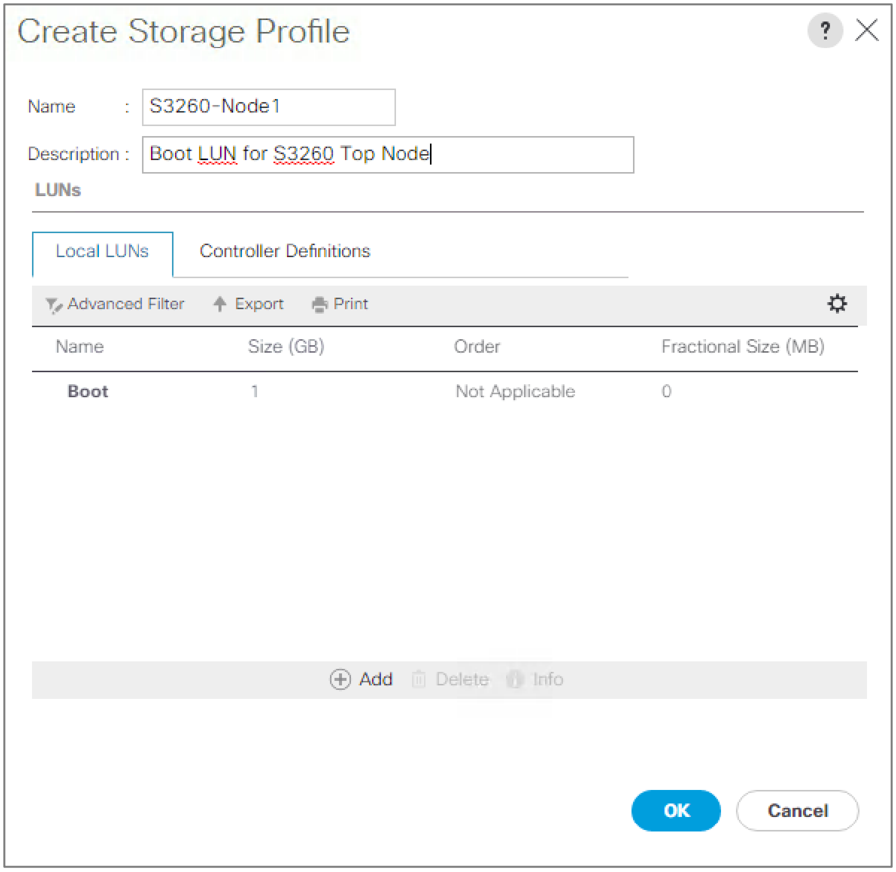
17. Leave everything else untouched.

18. Click `OK` and then `OK`.

19. Select your previously created Disk Group Policy for the Boot SSDs with the radio button under `Select Disk Group Configuration`.

20. Click `OK` and then `OK` and again `OK`.

Figure 54 Storage Profile for the top node of Cisco UCS S3260 Storage Server



Create Storage Profile

Name : S3260-Node1

Description : Boot LUN for S3260 Top Node

LUNs

Name	Size (GB)	Order	Fractional Size (MB)
Boot	1	Not Applicable	0

To create the Storage Profile for the bottom node of the Cisco UCS S3260 Storage Server, complete the following steps:

1. Select `Storage` in the left pane of the Cisco UCS Manager GUI.

2. Go to Storage > Storage Profiles and right-click Create Storage Profile.

3. Type in `S3260-Node2` in the `Name` field.

4. (Optional) Enter a description in the `Description` field.

5. Click `Add`.

6. Type in `Boot` in the `Name` field.

7. Configure as follows:

    a. Create Local LUN.
    b. Size (GB) = 1
    c. Fractional Size (MB) = 0
    d. Auto Deploy
    e. Select Expand To Available
    f. Select your previously created Disk Group Policy for the S3260 Boot SSDs with the radio button under `Select Disk Group Configuration`.

8. Click `OK` and then `OK` and again `OK`.

Creating Disk Group Policies and RAID 0 LUNs for top-loaded Cisco UCS S3260 Storage Server HDDs

To create Disk Group Policies and RAID 0 local LUNs for all top-loaded HDDs for a Cisco UCS S3260 Storage Server, use a Cisco UCS PowerTool Script, which accelerates and simplifies the deployment and complete the following steps:

Note: Please make sure that you have Cisco UCS PowerTool from the previous section installed.

1. Download the UCS PowerTool Script “Create RAID 0 Disk Group Policies and Storage Profile(s)” under https://communities.cisco.com/docs/DOC-70617

2. Start a PowerShell CLI and start the script.

3. Type in the Cluster IP of your Cisco UCS Manager.

4. Type in your password.

5. Type in `46` for the number of Disk Group Policies.

6. Type in `R0_HDD_` for the Disk Group Policies.

7. Type in `Y` for top-loaded SSDs installed.

8. Type in 8 for number of top-loaded SSDs.

9. Type in `S3260-Node1` as a Storage Profile for the top node of the Cisco UCS S3260 Storage Server.

10. Type in `9..32` for the used Disk IDs, which should be included in the Storage Profile.

11. Type in `Y` for creating another Storage Profile.

12. Type in `S3260-Node2` as a Storage Profile for the bottom node of the Cisco UCS S3260 Storage Server.

13. Type in 33..56 for the used Disk IDs, which should be included in the Storage Profile.

Create Storage Profile for Cisco UCS C220 M4S Rack-Mount Server

To create a Storage Profile for the Cisco UCS C220 M4S, complete the following steps:

1. Select Storage in the left pane of the Cisco UCS Manager GUI.

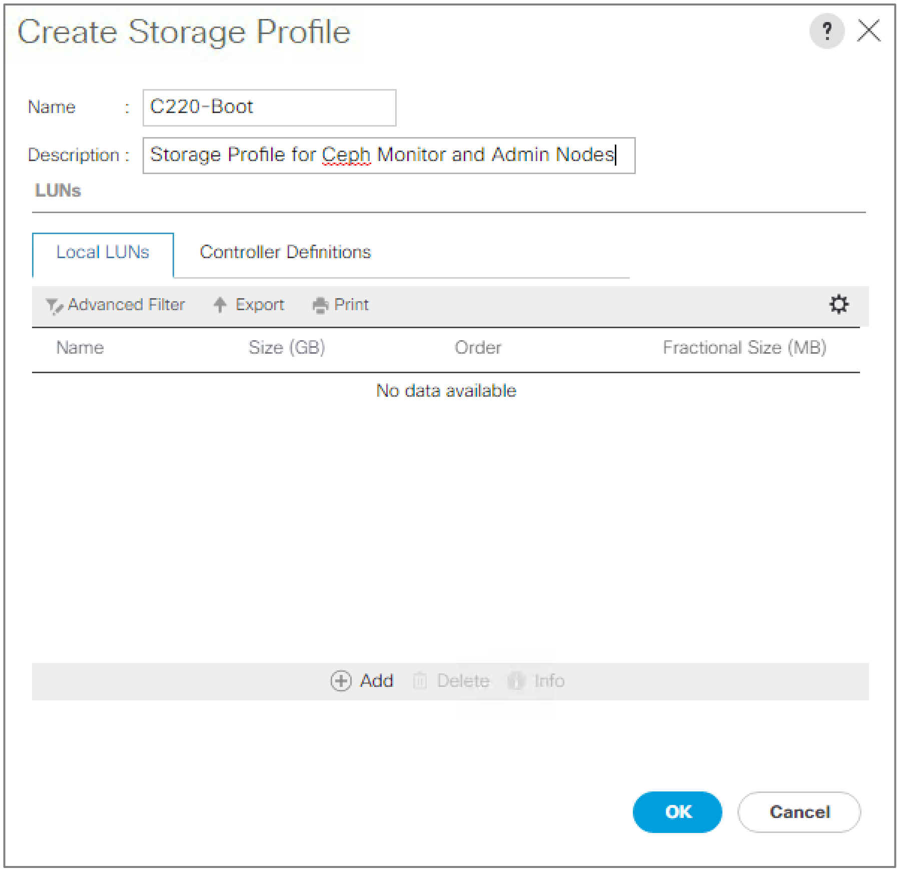
2. Go to Storage > Storage Profiles and right-click Create Storage Profile.

3. Type in C220-Boot in the Name field.

4. (Optional) Enter a description in the Description field.

5. Click Add.

Figure 55 Create Storage Profile for Cisco UCS C220 M4S



Create Storage Profile

Name : C220-Boot

Description : Storage Profile for Ceph Monitor and Admin Nodes

LUNs

Name	Size (GB)	Order	Fractional Size (MB)

No data available

6. Type in `Boot` in the `Name` field.

7. Configure as follows:

   a. Create Local LUN

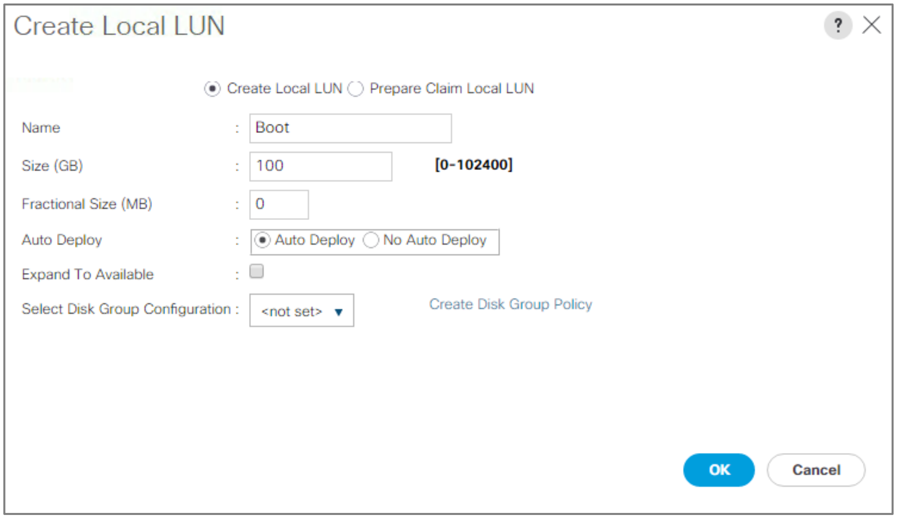
   b. Size (GB) = 100

   c. Fractional Size (MB) = 0

   d. Auto Deploy

   e. Click Create Disk Group Policy

Figure 56 Create Local LUN for Cisco UCS C220 M4S



Create Local LUN

● Create Local LUN ○ Prepare Claim Local LUN

Name : Boot

Size (GB) : 100 [0-102400]

Fractional Size (MB) : 0

Auto Deploy : ● Auto Deploy ○ No Auto Deploy

Expand To Available : □

Select Disk Group Configuration : <not set> ▼

8. Type in `Boot-Ceph` in the `Name` field.

9. (Optional) Enter a description in the `Description` field.

10. RAID Level = RAID 1 Mirrored

11. Select Disk Group Configuration (Manual)

12. Click `Add`.

13. Type in `1` for `Slot Number`.

14. Click `OK` and then again `Add`.

15. Type in `2` for `Slot Number`.

16. Leave everything else untouched.

17. Click `OK` and then `OK`.

18. Select your previously created Disk Group Policy for the C220 M4S Boot Disks with the radio button under `Select Disk Group Configuration`.

19. Click `OK` and then `OK` and again `OK`.

Creating a Service Profile Template

Create Service Profile Template for Cisco UCS S3260 Storage Server Top and Bottom Node

To create a Service Profile Template, complete the following steps:

1. Select `Servers` in the left pane of the Cisco UCS Manager GUI.

2. Go to Servers > Service Profile Templates > root and right-click Create Service Profile Template.

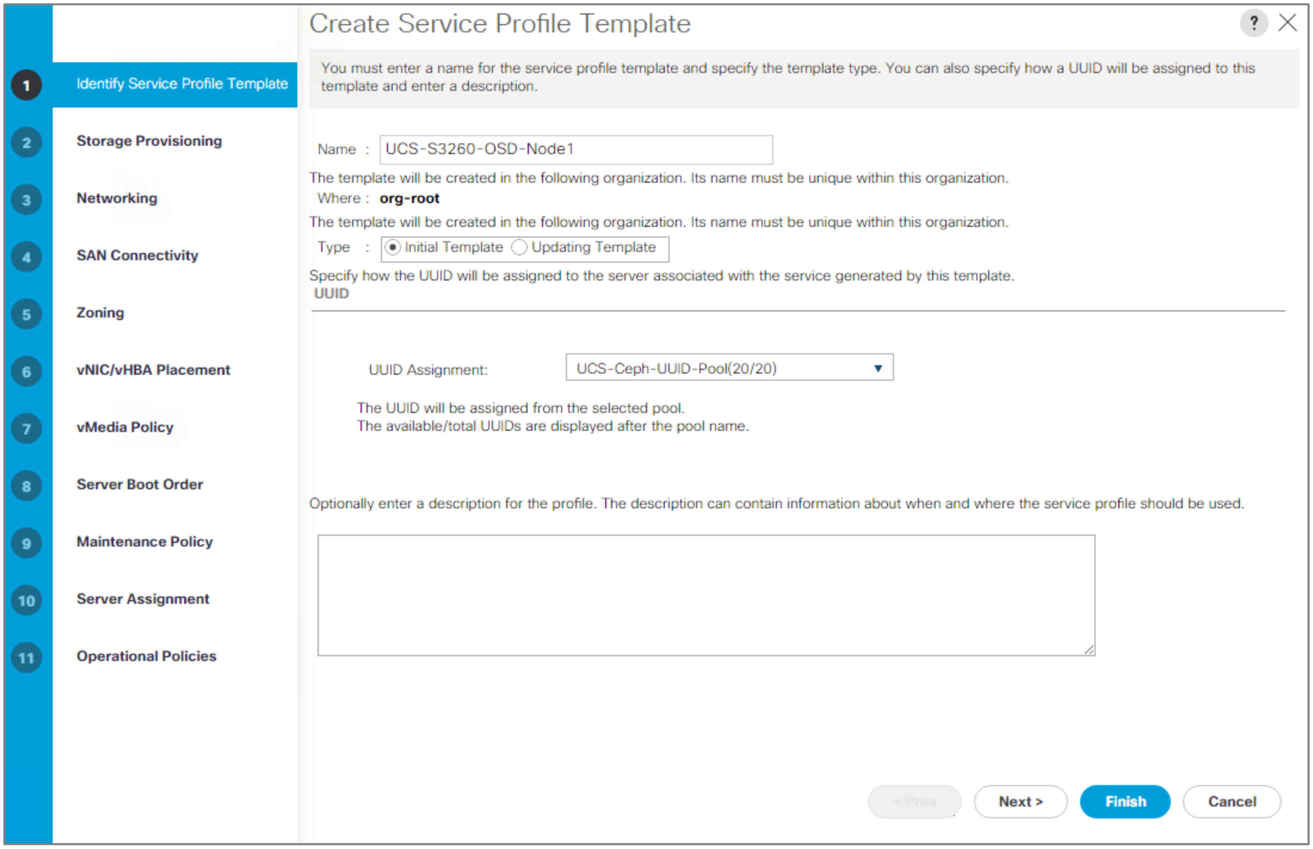
Identify Service Profile Template

1. Type in `UCS-S3260-OSD-Node1` in the `Name` field.

2. In the `UUID Assignment` section, select the UUID Pool you created in the beginning.

3. (Optional) Enter a description in the `Description` field.

Figure 57 Identify Service Profile Template



Create Service Profile Template

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

Name : UCS-S3260-OSD-Node1

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: UCS-Ceph-UUID-Pool(20/20)

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.

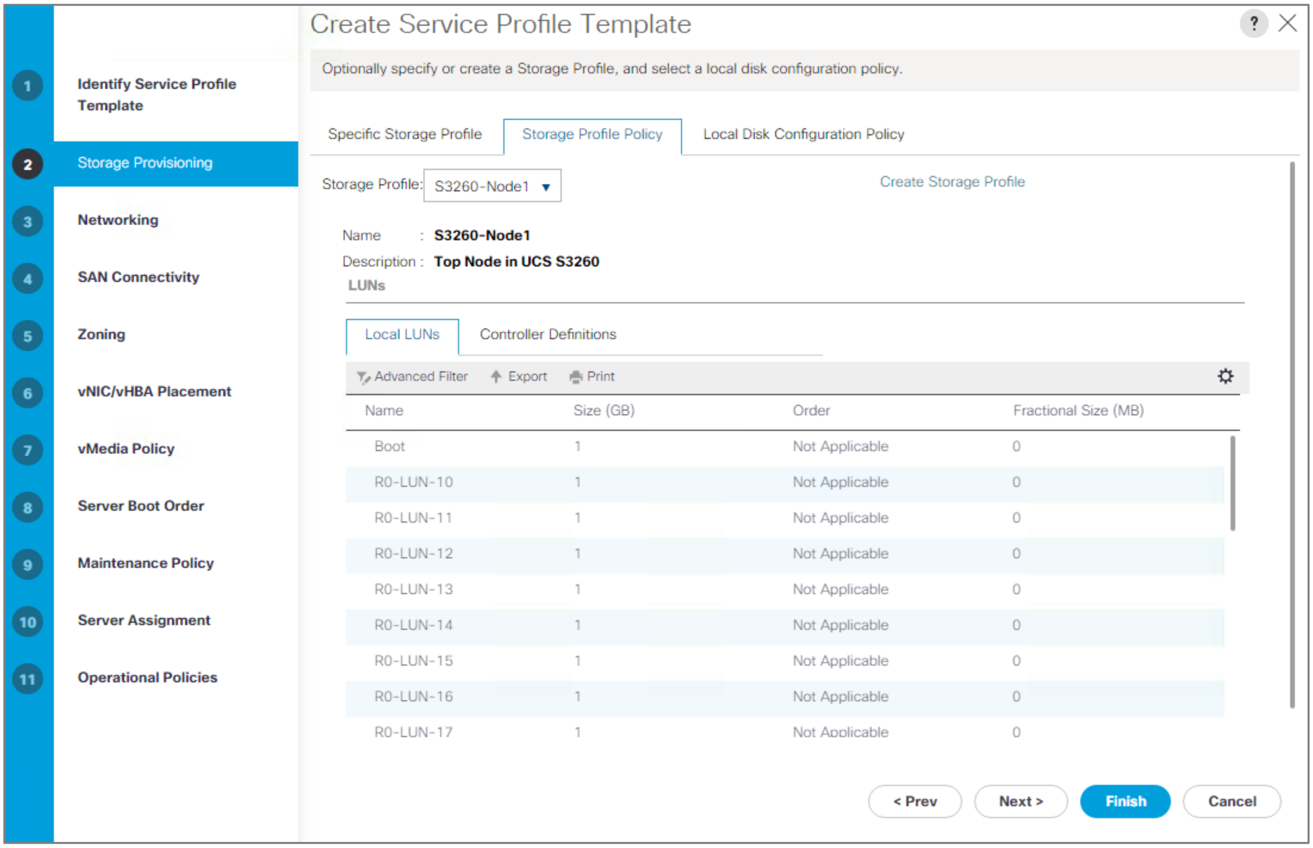
4. Click `Next`.

Storage Provisioning

1. Go to the `Storage Profile Policy` tab and select the Storage Profile `S3260-Node1` for the top node of the Cisco UCS S3260 Storage Server you created before.

2. Click `Next`.

Figure 58 Storage Provisioning



Create Service Profile Template

Optionally specify or create a Storage Profile, and select a local disk configuration policy.

Storage Profile: S3260-Node1

Name : S3260-Node1

Description : Top Node in UCS S3260

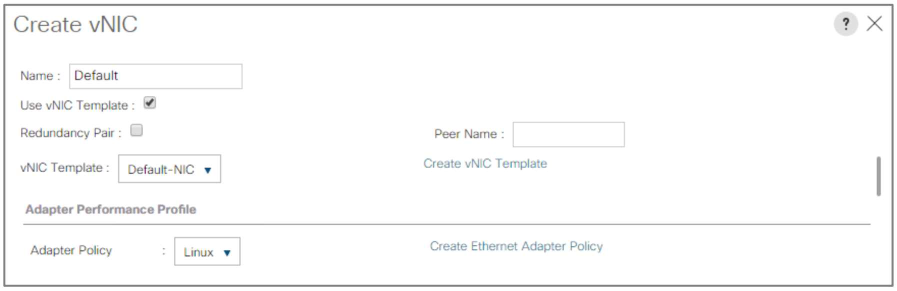
LUNs

Name	Size (GB)	Order	Fractional Size (MB)
Boot	1	Not Applicable	0
R0-LUN-10	1	Not Applicable	0
R0-LUN-11	1	Not Applicable	0
R0-LUN-12	1	Not Applicable	0
R0-LUN-13	1	Not Applicable	0
R0-LUN-14	1	Not Applicable	0
R0-LUN-15	1	Not Applicable	0
R0-LUN-16	1	Not Applicable	0
R0-LUN-17	1	Not Applicable	0

Networking

1. Keep the Dynamic vNIC Connection Policy field at the default.
2. Select the Expert radio button for the option How would you like to configure LAN connectivity?
3. Click `Add` to add a vNIC to the template.
4. Insert `Default` as `Name`.
5. Select Use vNIC Template.
6. Select Default-NIC as vNIC Template.
7. Select RHEL as Adapter Policy.
8. Click `OK`.

Figure 59 Create vNIC



Create vNIC

Name : Default

Use vNIC Template : ☑

Redundancy Pair : □

Peer Name :

vNIC Template : Default-NIC

Create vNIC Template

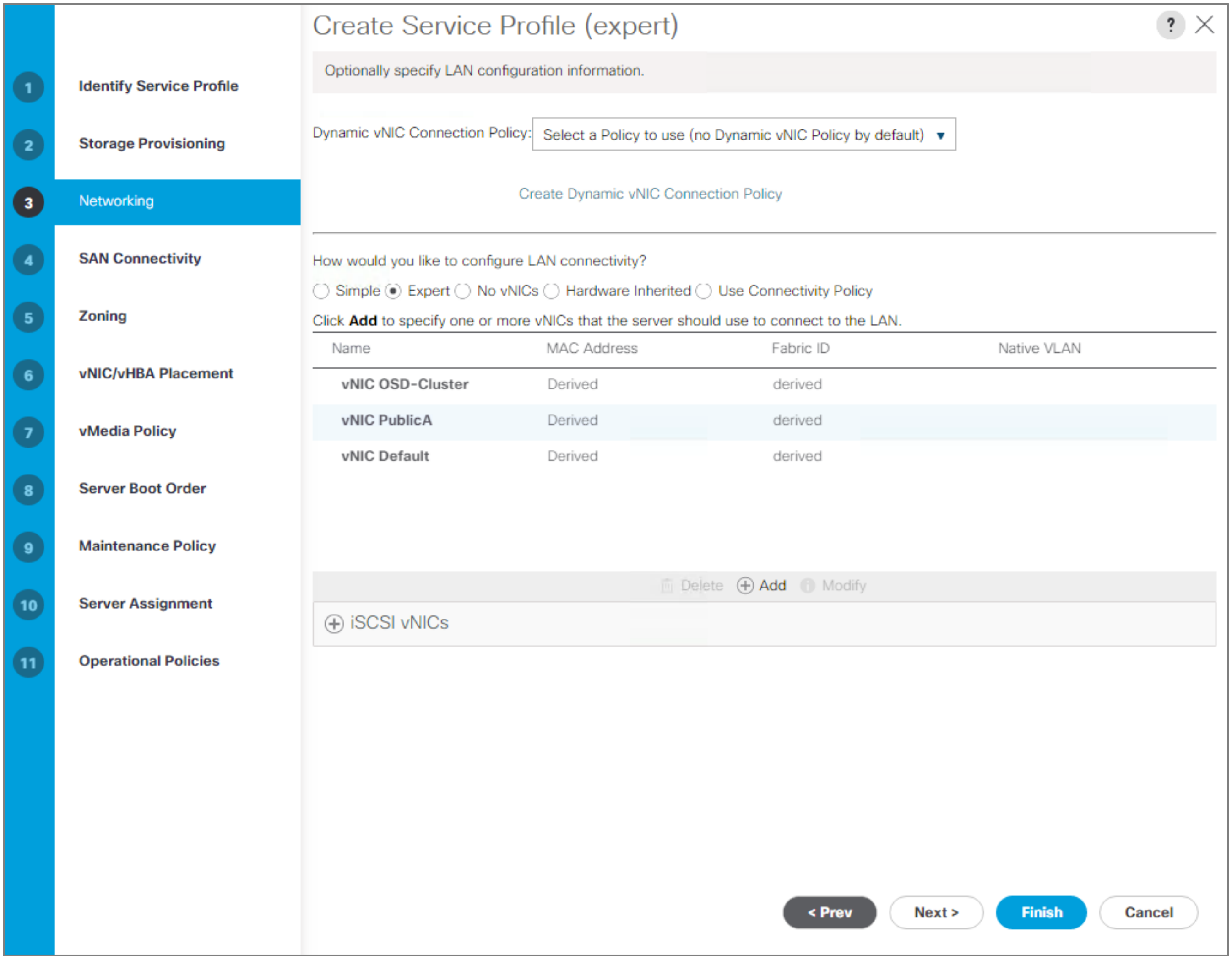
Adapter Performance Profile

Adapter Policy : Linux

Create Ethernet Adapter Policy

9. Repeat the steps for PublicA and OSD-Cluster vNIC by choosing the appropriate vNIC template you created before for the Public and Cluster network. Make sure you always select `RHEL` as `Adapter Policy`.

Figure 60 Summary Networking



Create Service Profile (expert)

Optionally specify LAN configuration information.

Dynamic vNIC Connection Policy: Select a Policy to use (no Dynamic vNIC Policy by default)

Create Dynamic vNIC Connection Policy

How would you like to configure LAN connectivity?

○ Simple ● Expert ○ No vNICs ○ Hardware Inherited ○ Use Connectivity Policy

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

Name	MAC Address	Fabric ID	Native VLAN
vNIC OSD-Cluster	Derived	derived	
vNIC PublicA	Derived	derived	
vNIC Default	Derived	derived	

⊕ iSCSI vNICs

10. Click `Next` to continue with SAN Connectivity.

11. Select No vHBA for How would you like to configure SAN Connectivity?

12. Click `Next` to continue with Zoning.

13. Click `Next`.

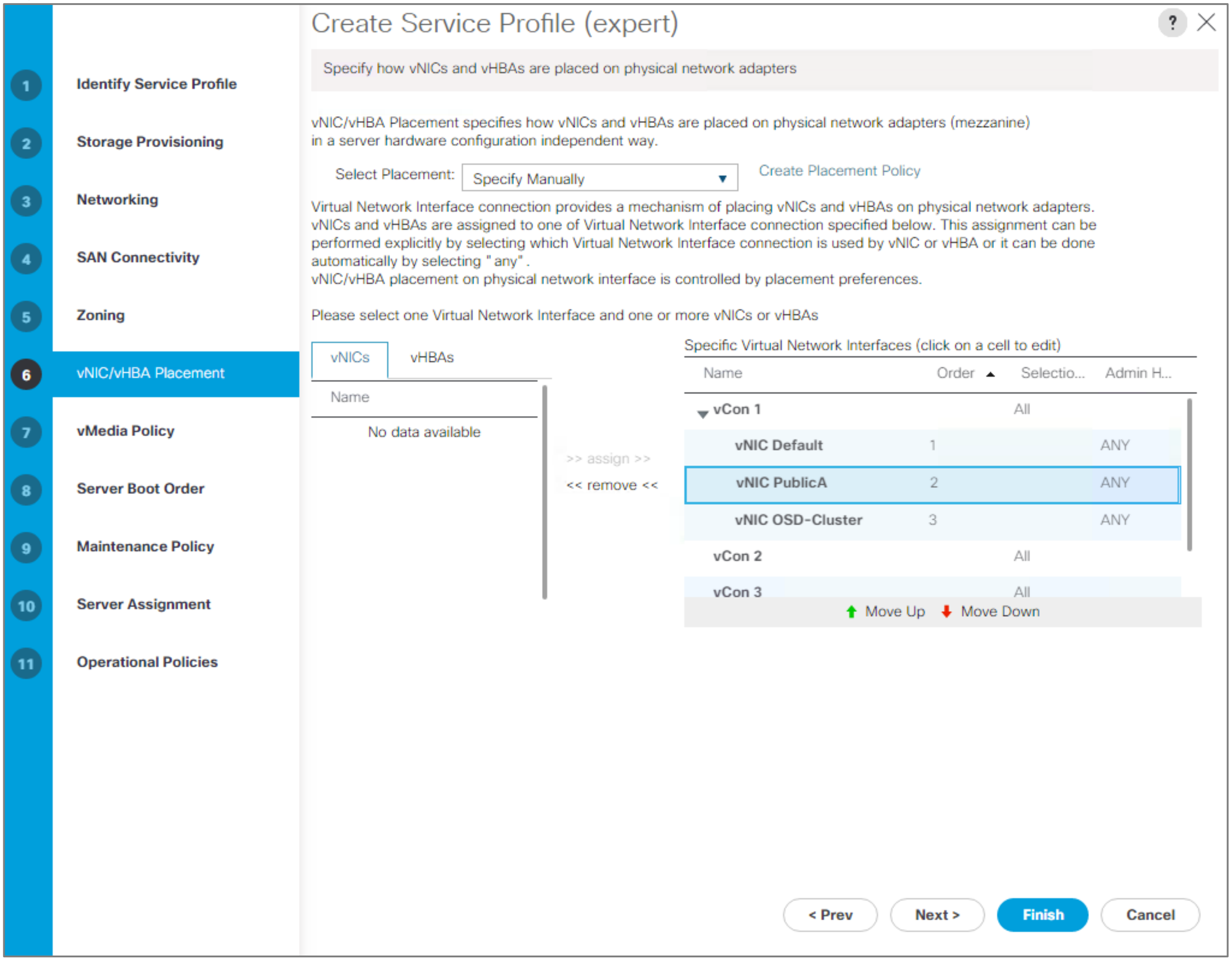
vNIC/vHBA Placement

1. Select `Specify Manually` form the drop-down menu.

2. Under `vNIC` select all three vNICs and select `vCON 1` on the right site, then click assign.

3. Sort all vNICs with Default at the top, then PublicA and then OSD-Cluster.

Figure 61 vNIC/vHBA Placement



Specify how vNICs and vHBAs are placed on physical network adapters

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

Select Placement: Specify Manually

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

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

Name
No data available

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

Name	Order	Selectio...	Admin H...
vCon 1		All	
vNIC Default	1		ANY
vNIC PublicA	2		ANY
vNIC OSD-Cluster	3		ANY
vCon 2		All	
vCon 3		All	

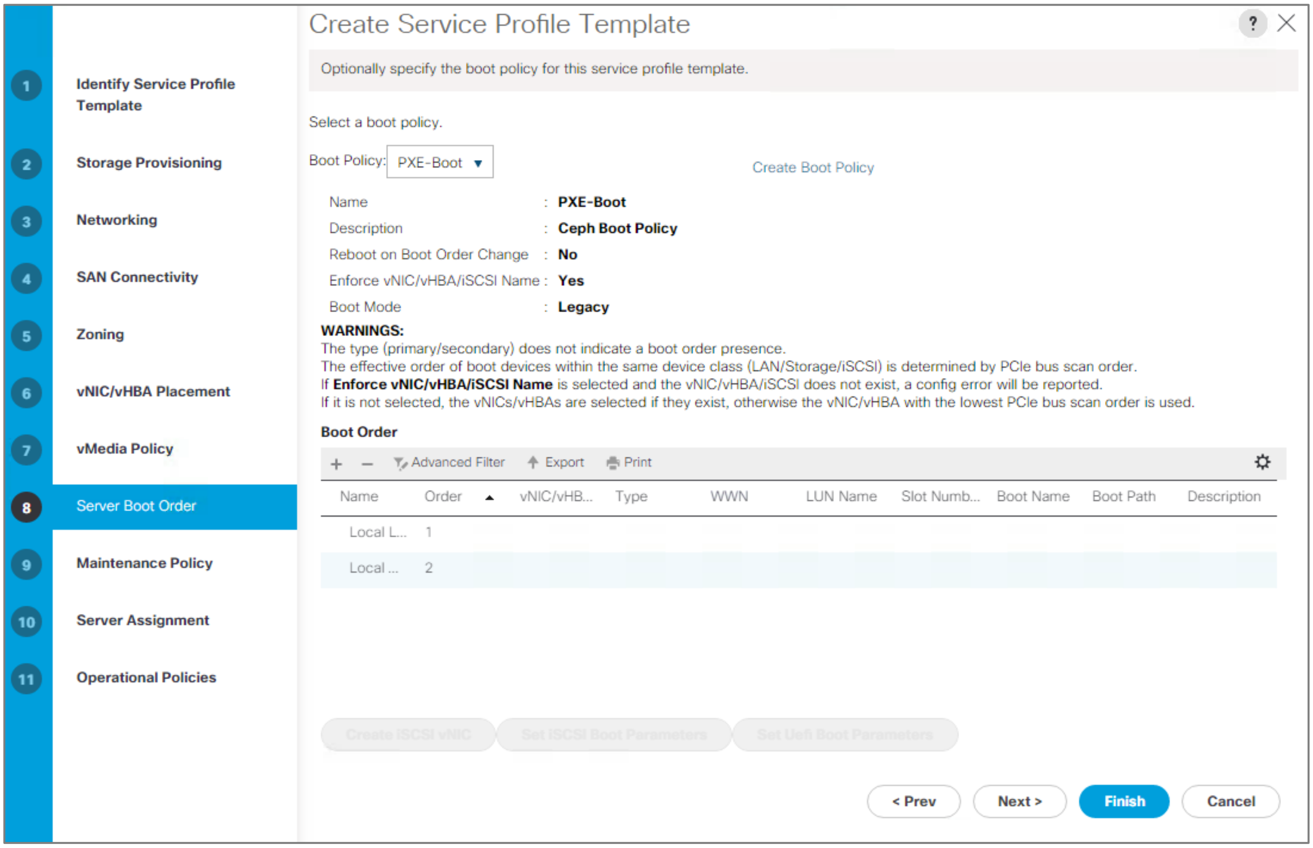
4. Click `Next` to continue with vMedia Policy.

5. Click `Next`.

Server Boot Order

1. Select the Boot Policy `PXE-Boot` you created before under `Boot Policy`.

Figure 62 Server Boot Order



Optionally specify the boot policy for this service profile template.

Select a boot policy.

Boot Policy: PXE-Boot ▼

Name : **PXE-Boot**
Description : **Ceph Boot Policy**
Reboot on Boot Order Change : **No**
Enforce vNIC/vHBA/iSCSI Name : **Yes**
Boot Mode : **Legacy**

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

Boot Order

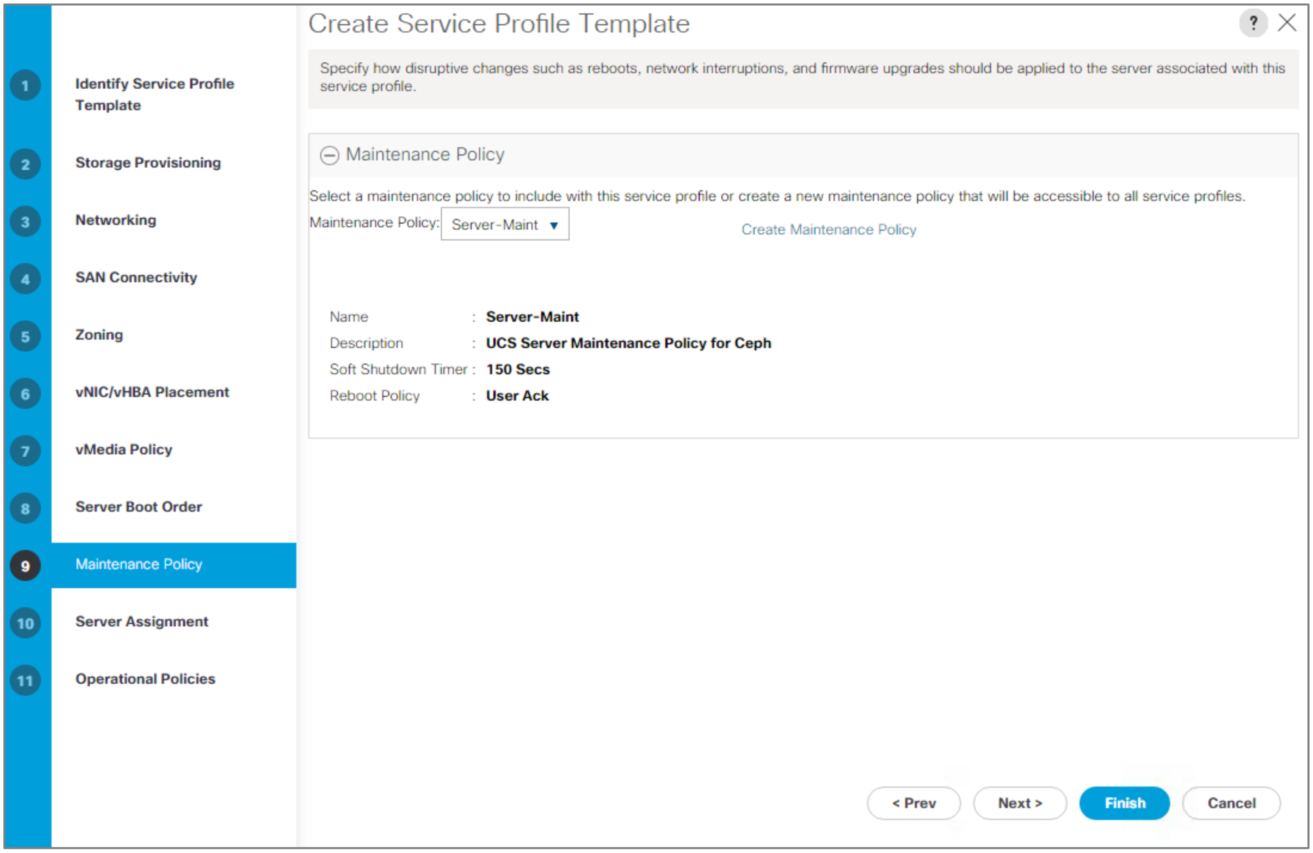
Name	Order ▲	vNIC/vHB...	Type	WWN	LUN Name	Slot Numb...	Boot Name	Boot Path	Description
Local L...	1								
Local ...	2								

2. Click Next.

Maintenance Policy

1. Select the Maintenance Policy you created before under Maintenance Policy.

Figure 63 Maintenance Policy



Create Service Profile Template

Specify how disruptive changes such as reboots, network interruptions, and firmware upgrades should be applied to the server associated with this service profile.

Maintenance Policy

Select a maintenance policy to include with this service profile or create a new maintenance policy that will be accessible to all service profiles.

Maintenance Policy: Server-Maint ▾

Name : **Server-Maint**
Description : **UCS Server Maintenance Policy for Ceph**
Soft Shutdown Timer : **150 Secs**
Reboot Policy : **User Ack**

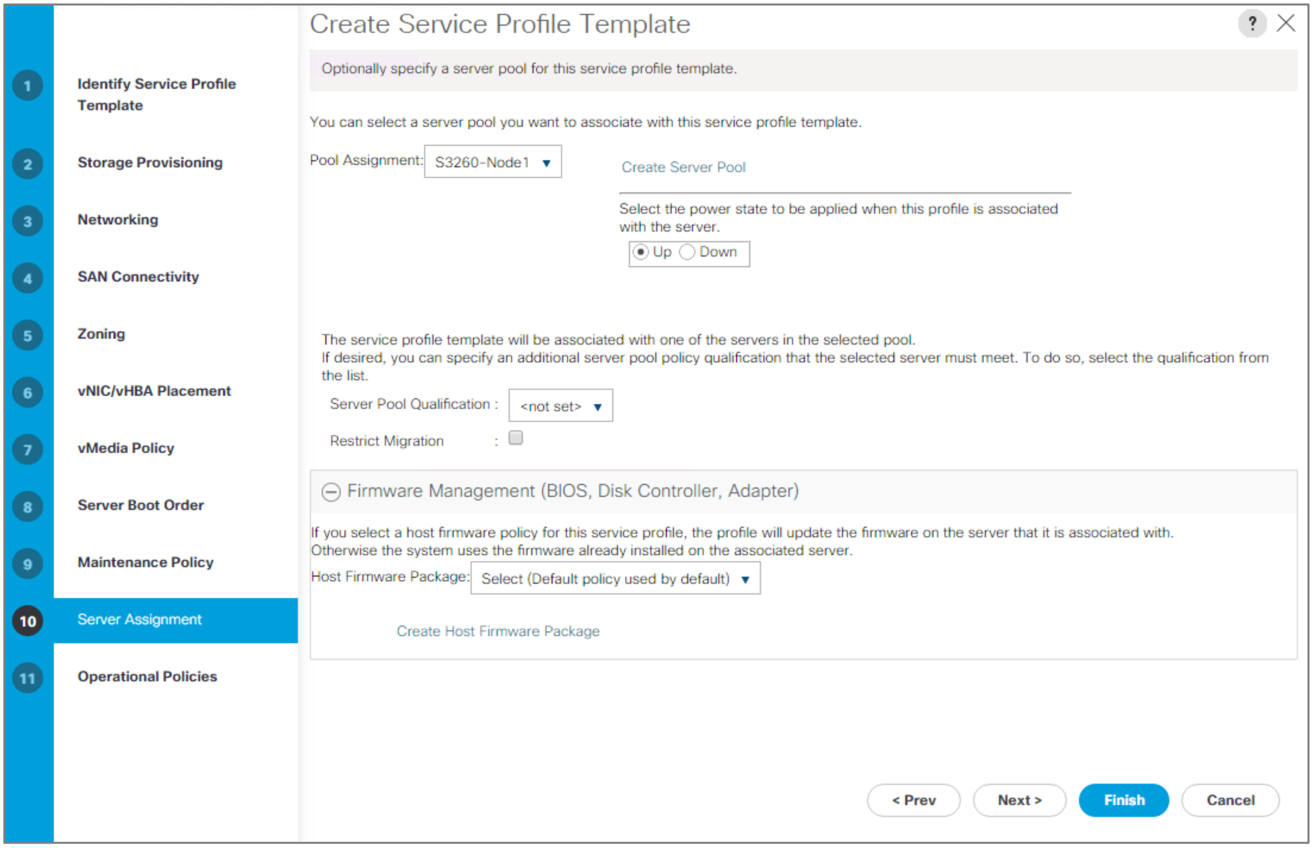
2. Click `Next`.

Server Assignment

1. Select the Server Pool you created before for the top node of the Cisco UCS S3260 Storage Server in the drop-down menu of the `Pool-Assignment`.

2. Leave everything else untouched.

Figure 64 Server Assignment



Create Service Profile Template

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: S3260-Node1 ▾

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 : □

⊖ Firmware Management (BIOS, Disk Controller, Adapter)

If you select a host firmware policy for this service profile, the profile will update the firmware on the server that it is associated with.
Otherwise the system uses the firmware already installed on the associated server.

Host Firmware Package: Select (Default policy used by default) ▾

Create Host Firmware Package

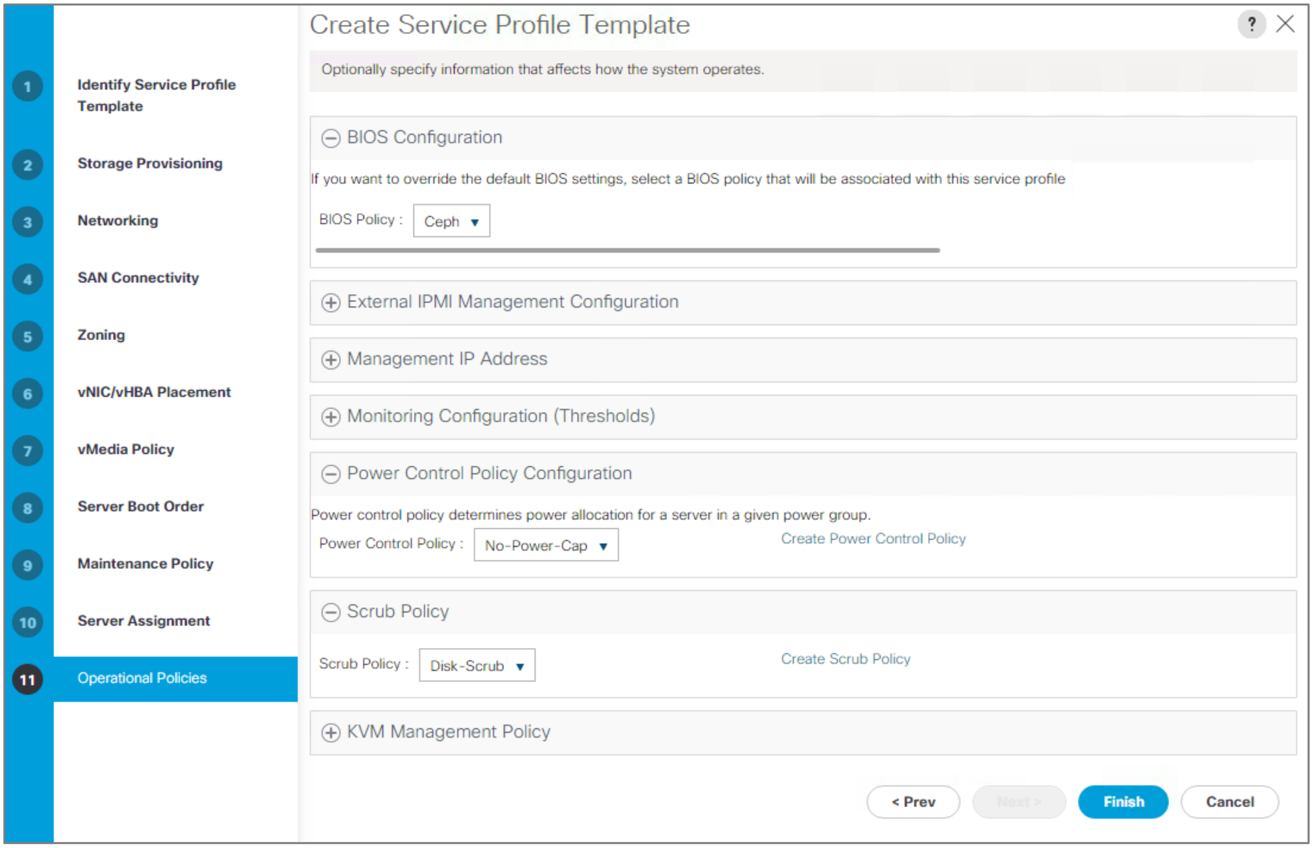
3. Click `Next`.

Operational Policies

1. Select under `Power Control Policy Configuration` the previous created Power Policy.

2. Select under `Scrub Policy` the previous created Scrub Policy.

Figure 65 Operational Policy



Create Service Profile Template

Optionally specify information that affects how the system operates.

BIOS Configuration

If you want to override the default BIOS settings, select a BIOS policy that will be associated with this service profile

BIOS Policy : Ceph

External IPMI Management Configuration

Management IP Address

Monitoring Configuration (Thresholds)

Power Control Policy Configuration

Power control policy determines power allocation for a server in a given power group.

Power Control Policy : No-Power-Cap

Scrub Policy

Scrub Policy : Disk-Scrub

KVM Management Policy

3. Click `Finish` and then `OK`.

4. Repeat the steps for the bottom node of the Cisco UCS S3260 Storage Server but change the following

5. Choose the Storage Profile for the bottom node you created before.

6. Choose PublicB-NIC as the Public network interface.

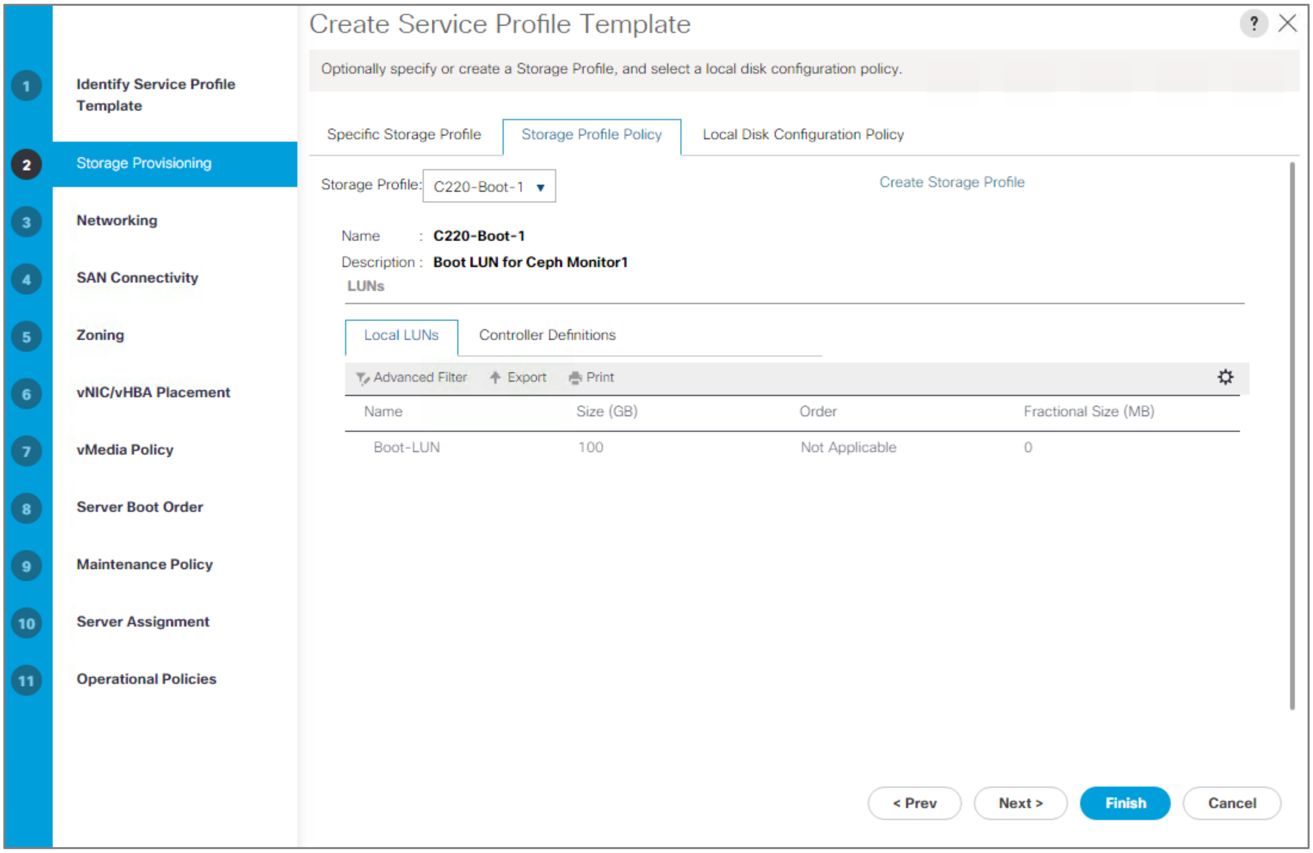
7. Choose the Server Pool with Slot ID 2 you created before.

Create Service Profile Template for Cisco UCS C220 M4S

The Service Profiles for the Rack-Mount Servers are very similar to the profiles created for the S3260. The only differences are with the Storage Profiles, Networking, vNIC/vHBA Placement and Server Pools. The changes are listed in this section and to create these profiles, complete the following steps:

1. In the `Storage Provisioning` tab choose the appropriate Storage Profile for the Cisco UCS C220 M4S you created before.

Figure 66 Storage Provisioning for Cisco UCS C220 M4S



Create Service Profile Template

Optionally specify or create a Storage Profile, and select a local disk configuration policy.

Storage Profile: C220-Boot-1

Name : C220-Boot-1

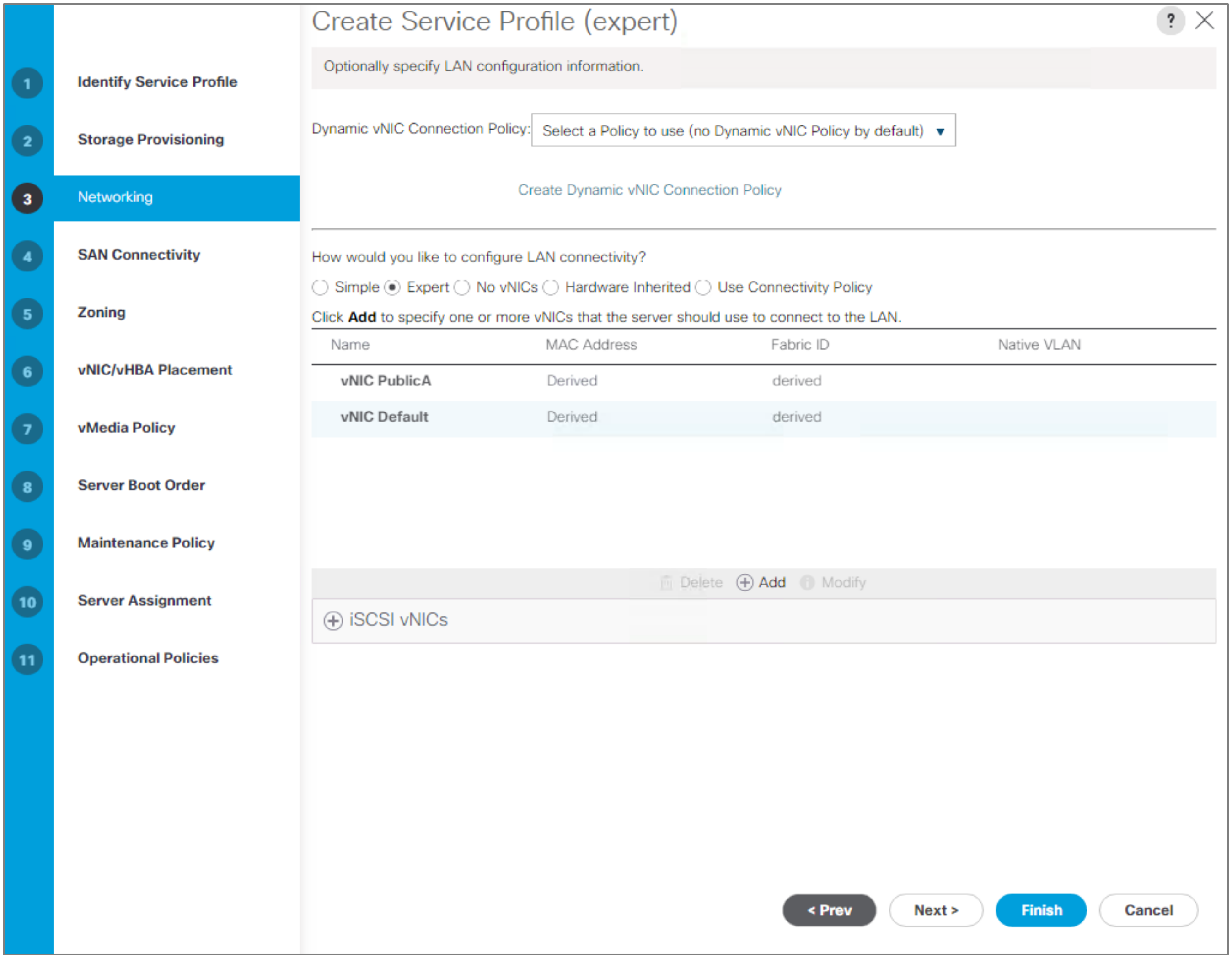
Description : Boot LUN for Ceph Monitor1

LUNs

Name	Size (GB)	Order	Fractional Size (MB)
Boot-LUN	100	Not Applicable	0

2. In the `Networking` tab create only two vNICs for Default and PublicA network in the same way and same order like the section before.

Figure 67 Networking for Cisco UCS C220 M4S



Create Service Profile (expert)

Optionally specify LAN configuration information.

Dynamic vNIC Connection Policy: Select a Policy to use (no Dynamic vNIC Policy by default) ▼

How would you like to configure LAN connectivity?

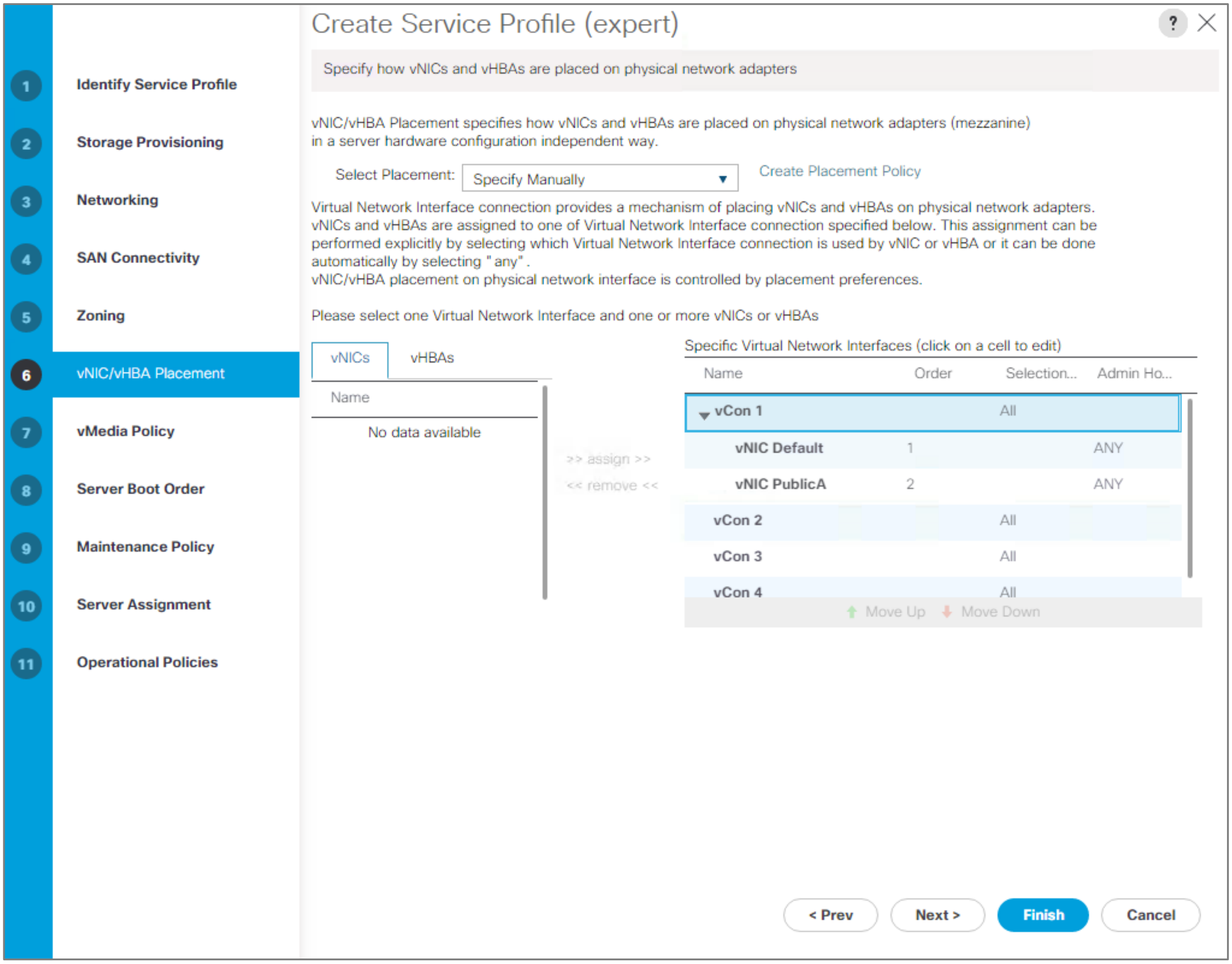
○ Simple ● Expert ○ No vNICs ○ Hardware Inherited ○ Use Connectivity Policy

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

Name	MAC Address	Fabric ID	Native VLAN
vNIC PublicA	Derived	derived	
vNIC Default	Derived	derived	

3. Configure the vNIC/vHBA Placement in the following order:

Figure 68 vNIC/vHBA Placement for Cisco UCS C220 M4S



Create Service Profile (expert)

Specify how vNICs and vHBAs are placed on physical network adapters

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

Select Placement: Specify Manually ▼ Create Placement Policy

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

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

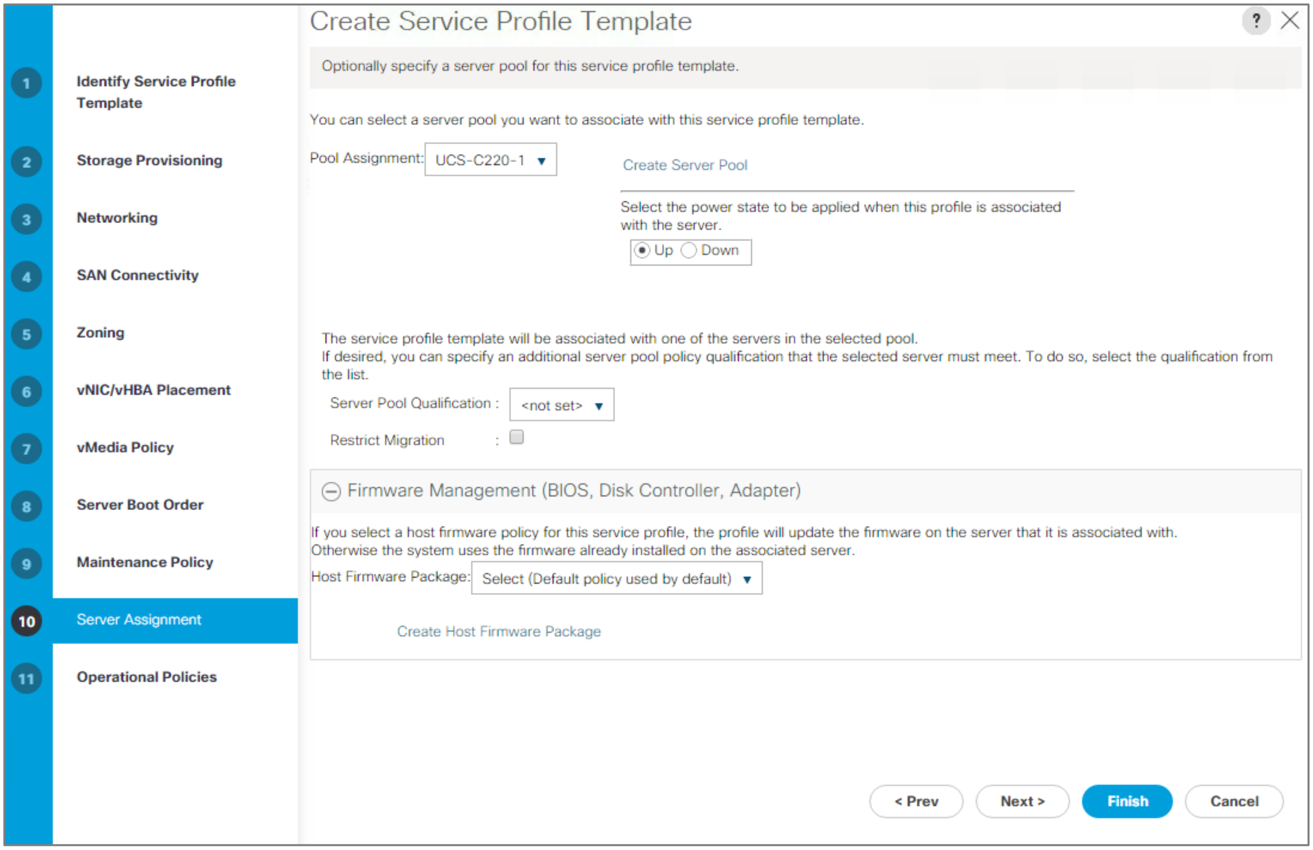
Name
No data available

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

Name	Order	Selection...	Admin Ho...
vCon 1		All	
vNIC Default	1		ANY
vNIC PublicA	2		ANY
vCon 2		All	
vCon 3		All	
vCon 4		All	

4. In the Server Assignment tab choose the Server Pool you created before for the Cisco UCS C220 M4S.

Figure 69 Server Assignment for Cisco UCS C220 M4S



Create Service Profile Template

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: UCS-C220-1

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 : □

Firmware Management (BIOS, Disk Controller, Adapter)

If you select a host firmware policy for this service profile, the profile will update the firmware on the server that it is associated with.
Otherwise the system uses the firmware already installed on the associated server.

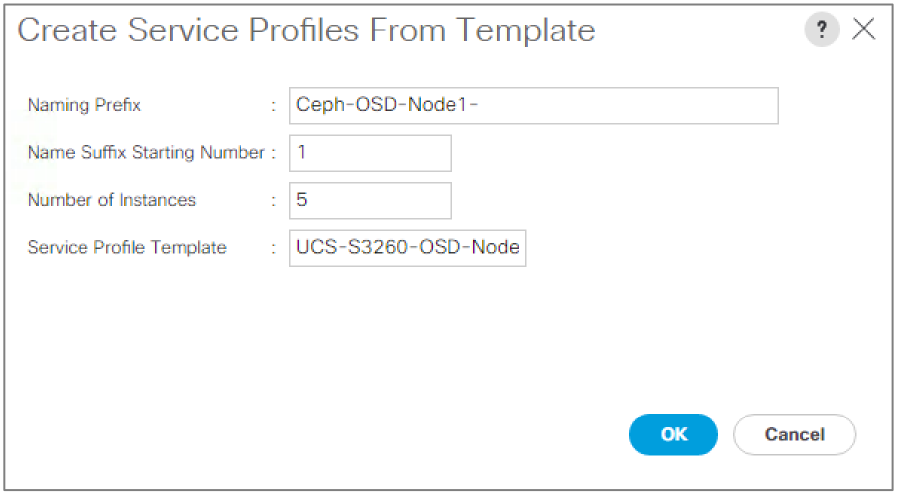
Host Firmware Package: Select (Default policy used by default)

Create Service Profiles from Template

Now create the appropriate Service Profiles from the previous Service Profile Templates. To create the first profile for the top node of the Cisco UCS S3260 Storage Server, complete the following steps:

1. Select `Servers` from the left pane of the Cisco UCS Manager GUI.
2. Go to Servers > Service Profiles and right-click Create Service Profiles from Template.
3. Type in Ceph-OSD-Node1- in the Name Prefix field.
4. Leave Name Suffix Starting Number as 1.
5. Type in 5 for the `Number of Instances`.
6. Choose `UCS-S3260-Node1` as the `Service Profile Template` you created before for the top node of the Cisco UCS S3260 Storage Server.
7. Click `OK` and then `OK`.

Figure 70 Create Service Profiles from Template for the Top Node of the S3260



Create Service Profiles From Template

Naming Prefix : Ceph-OSD-Node1-

Name Suffix Starting Number : 1

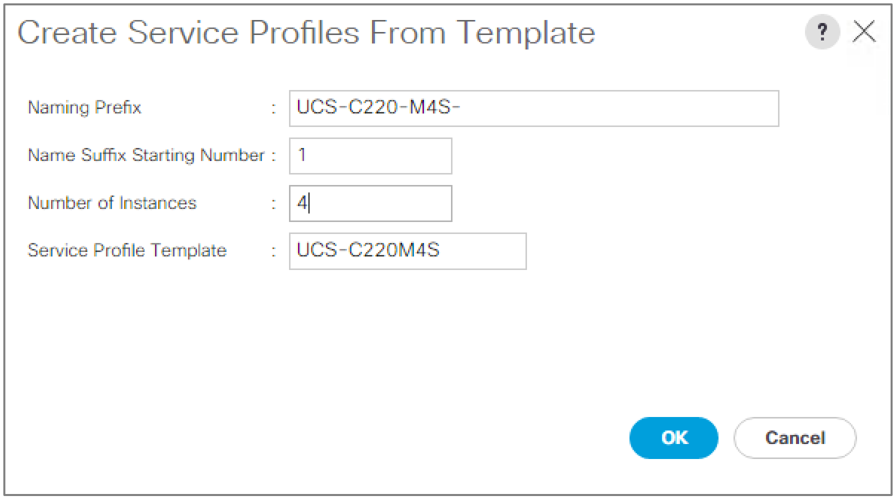
Number of Instances : 5

Service Profile Template : UCS-S3260-OSD-Node

8. Repeat steps 1-5 for the next Service Profile for the bottom node of the Cisco UCS S3260 Storage Server and choose the Service Profile Template `UCS-S3260-Node2` you created before for the bottom node of the Cisco UCS S3260 Storage Server.

9. Repeat steps 1-5 for the next Service Profile for the Cisco UCS C220 M4S Rack-Mount Server and choose the appropriate Service Profile Template `UCS-C220M4S` you created before for the Cisco UCS C220 M4 S Rack-Mount Server.

Figure 71 Create Service Profiles from Template for the Cisco UCS C220 M4S



Create Service Profiles From Template

Naming Prefix : UCS-C220-M4S-

Name Suffix Starting Number : 1

Number of Instances : 4

Service Profile Template : UCS-C220M4S

Creating Port Channel for Uplinks

Create Port Channel for Fabric Interconnect A/B

To create Port Channels to the connected Nexus 9332PQ switches, complete the following steps:

1. Select the `LAN` tab in the left pane of the Cisco UCS Manager GUI.

2. Go to LAN > LAN Cloud > Fabric A > Port Channels and right-click Create Port Channel.

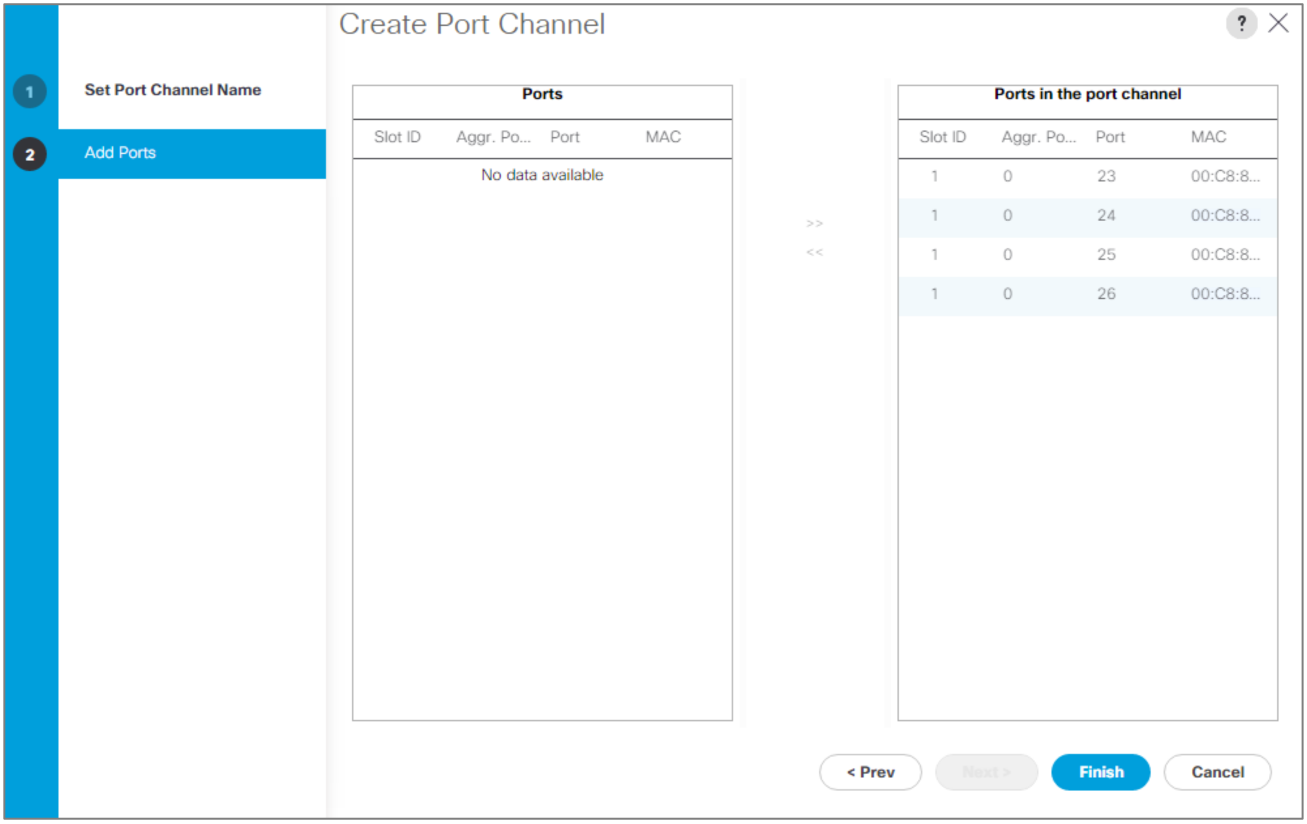
3. Type in `ID 100`.

4. Type in `vPC100` in the `Name` field.

5. Click Next.

6. Select the available ports on the left `23-26` and assign them with `>>` to `Ports in the Port Channel`.

Figure 72 Create Port Channel



Create Port Channel

Ports

Slot ID	Aggr. Po...	Port	MAC
No data available			

Ports in the port channel

Slot ID	Aggr. Po...	Port	MAC
1	0	23	00:C8:8...
1	0	24	00:C8:8...
1	0	25	00:C8:8...
1	0	26	00:C8:8...

7. Click `Finish` and then `OK`.

8. Repeat the same steps for Fabric B under LAN > LAN Cloud > Fabric B > Port Channels and right-click Create Port Channel.

9. Type in `ID 101`.

10. Type in a name in the `Name` field.

11. Click `Next`.

12. Select the available ports on the left `23-26` and assign them with `>>` to `Ports in the Port Channel`.

13. Click `Finish` and then `OK`.

Create VLAN Groups for Port Channel

To create a VLAN Group for the Port Channels, complete the following steps

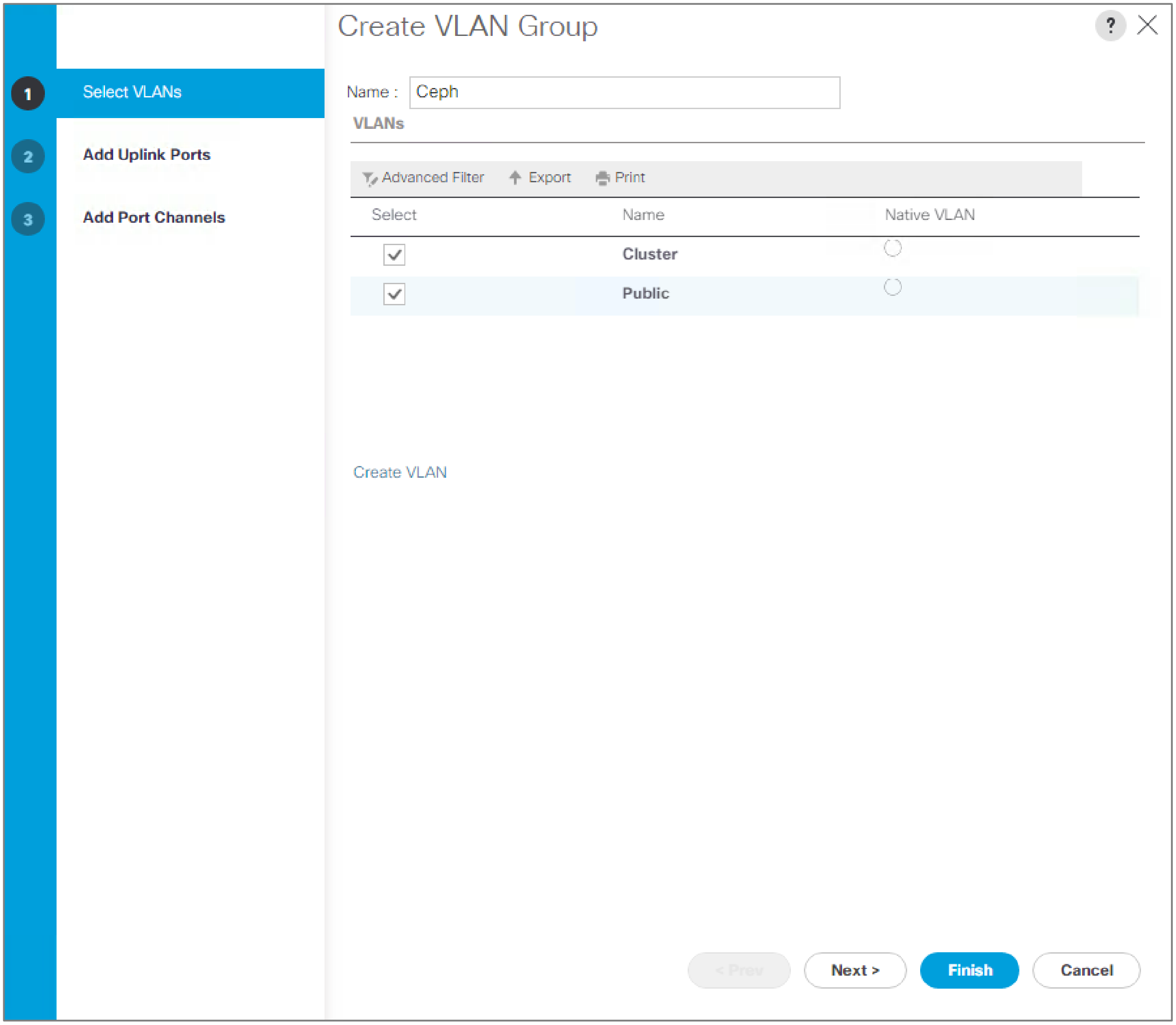
1. Select the `LAN` tab in the left pane of the Cisco UCS Manager GUI.

2. Go to LAN > LAN Cloud > VLAN Groups and right-click Create VLAN Group.

3. Type in `Ceph` in the `Name` field.

4. Select the available VLANs for Public and OSD-Cluster.

Figure 73 Create VLAN Group



Name : Ceph

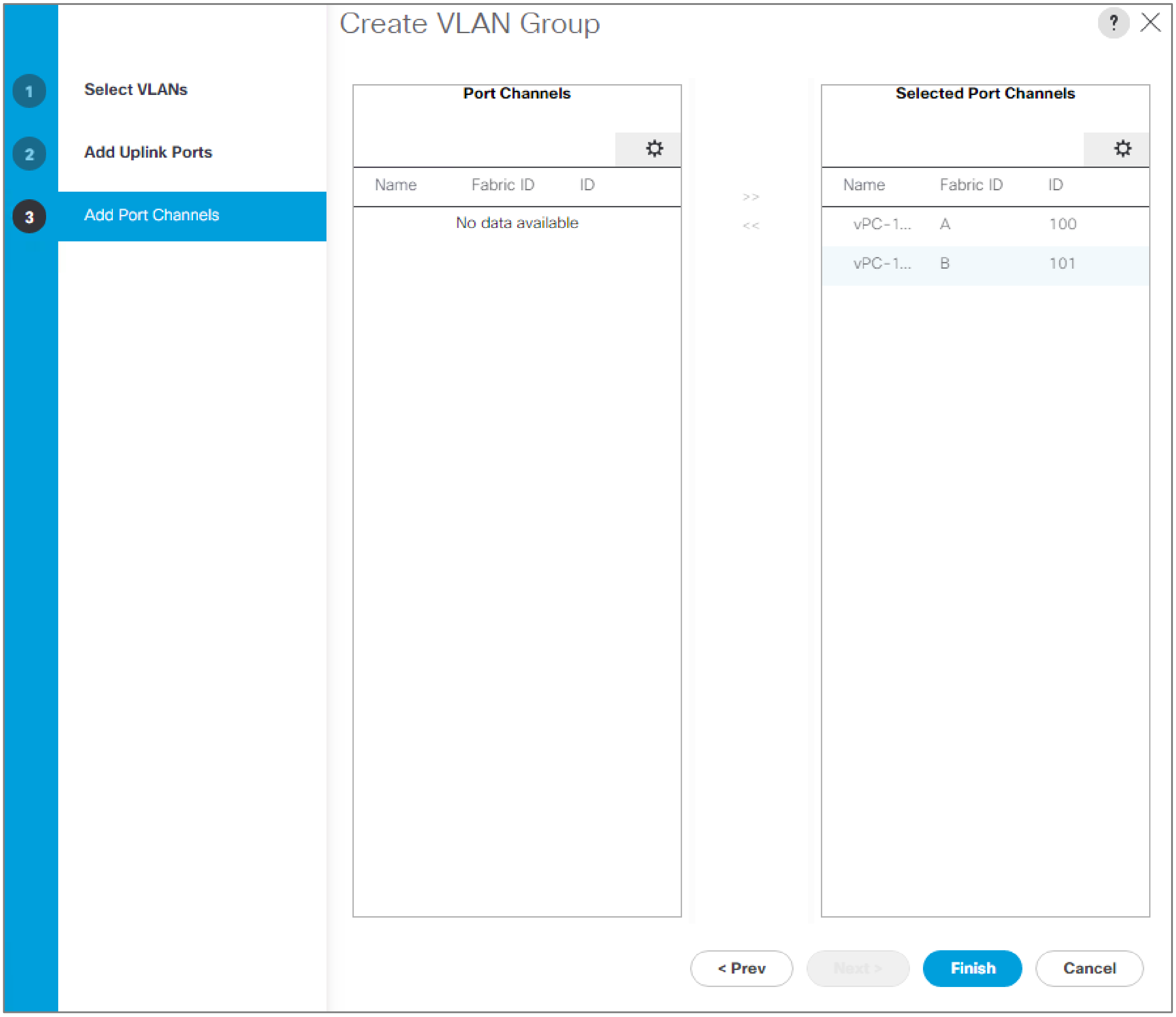
VLANs

Select	Name	Native VLAN
☑	Cluster	○
☑	Public	○

5. Click `Next` and then `Next`.

6. Select the previous created `Port Channels 100` and `101` on the left and assign them with `>>` to the `Selected Port Channels`.

Figure 74 Port Channels for VLAN Group



Port Channels

Name	Fabric ID	ID
No data available		

Selected Port Channels

Name	Fabric ID	ID
vPC-1...	A	100
vPC-1...	B	101

7. Click `Finish` and then `OK`.

Configure Scheduled Backup

To make sure that your configuration of Cisco UCS Manager gets stored, do regular backups of your configuration by completing the following steps:

1. Select `Admin` tab on the left site.
2. Select `All` from the drop-down menu.
3. Select `Policy Backup & Export` from the right site.
4. Under `Full State Backup Policy` choose an IP address of a host or a hostname as a backup target.
5. Choose the as an example `SCP` as the protocol.

6. Choose a `User` and a `Password`.

7. Choose a `Remote File` Location.

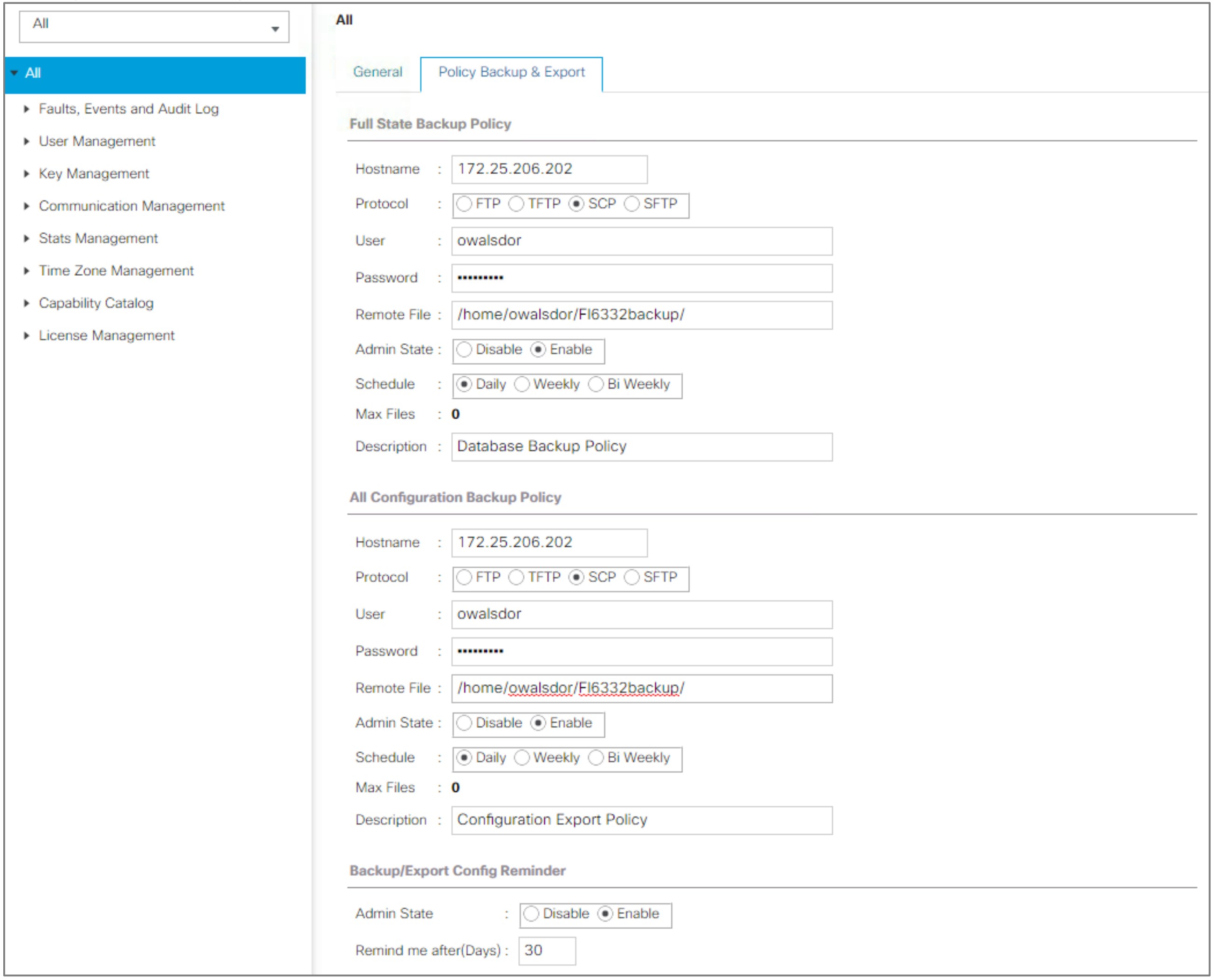
8. Choose you're a daily schedule.

9. Enter a Description.

10. Enter the same information under `All Configuration Backup Policy`.

11. Select Save Changes.

Figure 75 Configuration of Scheduled Backups



Full State Backup Policy

Hostname	:	172.25.206.202
Protocol	:	○ FTP ○ TFTP ● SCP ○ SFTP
User	:	owalsdor
Password	:	•••••••••
Remote File	:	/home/owalsdor/FI6332backup/
Admin State	:	○ Disable ● Enable
Schedule	:	● Daily ○ Weekly ○ Bi Weekly
Max Files	:	0
Description	:	Database Backup Policy

All Configuration Backup Policy

Hostname	:	172.25.206.202
Protocol	:	○ FTP ○ TFTP ● SCP ○ SFTP
User	:	owalsdor
Password	:	•••••••••
Remote File	:	/home/owalsdor/FI6332backup/
Admin State	:	○ Disable ● Enable
Schedule	:	● Daily ○ Weekly ○ Bi Weekly
Max Files	:	0
Description	:	Configuration Export Policy

Backup/Export Config Reminder

Admin State	:	○ Disable ● Enable
Remind me after(Days)	:	30

Configuration of Nexus C9332PQ Switch A and B

Both Cisco UCS Fabric Interconnect A and B are connected to two Cisco Nexus 9332PQ switches for connectivity to applications and Ceph clients. The following sections describe the setup of both Cisco Nexus 9332PQ switches.

Initial Setup of Nexus C9332PQ Switch A and B

To configure Switch A, please connect a Console to the Console port of each switch, power on the switch and complete the following steps:

1. Type `yes`.
2. Type `n`.
3. Type `n`.
4. Type `n`.
5. Enter the switch name.
6. Type `y`.
7. Type your IPv4 management address for Switch A.
8. Type your IPv4 management netmask for Switch A.
9. Type `y`.
10. Type your IPv4 management default gateway address for Switch A.
11. Type `n`.
12. Type `n`.
13. Type `y` for ssh service.
14. Press `<Return>` and then `<Return>`.
15. Type `y` for ntp server.
16. Type the IPv4 address of the NTP server.
17. Press `<Return>`, then `<Return>` and again `<Return>`.
18. Check the configuration and if correct then press `<Return>` and again `<Return>`.

The complete setup looks like the following:

```
         ---- System Admin Account Setup ----


   Do you want to enforce secure password standard (yes/no) [y]:


     Enter the password for "admin":

     Confirm the password for "admin":
            ---- 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.

   Please register Cisco Nexus9000 Family devices promptly with your
   supplier. Failure to register may affect response times for initial
   service calls. Nexus9000 devices must be registered to receive
   entitled support services.


   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
     Create another login account (yes/no) [n]:
     Configure read-only SNMP community string (yes/no) [n]:
     Configure read-write SNMP community string (yes/no) [n]:
     Enter the switch name : N9k-A
     Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]:
       Mgmt0 IPv4 address : 172.25.206.226
       Mgmt0 IPv4 netmask : 255.255.255.0
     Configure the default gateway? (yes/no) [y]:
       IPv4 address of the default gateway : 172.25.206.1
     Configure advanced IP options? (yes/no) [n]:
     Enable the telnet service? (yes/no) [n]:
     Enable the ssh service? (yes/no) [y]:
       Type of ssh key you would like to generate (dsa/rsa) [rsa]:
       Number of rsa key bits <1024-2048> [1024]:
     Configure the ntp server? (yes/no) [n]: y
       NTP server IPv4 address : 10.29.137.1
     Configure default interface layer (L3/L2) [L3]:
     Configure default switchport interface state (shut/noshut) [shut]:

     Configure CoPP system profile (strict/moderate/lenient/dense) [strict]:

   The following configuration will be applied:

     password strength-check

     switchname N9k-A

   vrf context management

   ip route 0.0.0.0/0 172.25.206.1

   exit

     no feature telnet

     ssh key rsa 1024 force

     feature ssh

     ntp server 10.29.137.1

     no system default switchport

     system default switchport shutdown

     copp profile strict

   interface mgmt0

   ip address 172.25.206.226 255.255.255.0

   no shutdown

   Would you like to edit the configuration? (yes/no) [n]:


   Use this configuration and save it? (yes/no) [y]:


   [########################################] 100%

   Copy complete.


   User Access Verification

   N9k-A login:
```

Repeat the same steps for the Cisco Nexus 9332PQ Switch B with the exception of configuring a different IPv4 management address in step 7.

Enable Features on Cisco Nexus 9332PQ Switch A and B

To enable the features UDLD, VLAN, HSRP, LACP, VPC, and Jumbo Frames, connect to the management interface via ssh on both switches and complete the following steps on both Switch A and B:

Switch A

```
N9k-A# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

N9k-A(config)# feature udld

N9k-A(config)# feature interface-vlan

N9k-A(config)# feature hsrp

N9k-A(config)# feature lacp

N9k-A(config)# feature vpc

N9k-A(config)# system jumbomtu 9216

N9k-A(config)# exit

N9k-A#
```

Switch B

```
N9k-B# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

N9k-B(config)# feature udld

N9k-B(config)# feature interface-vlan

N9k-B(config)# feature hsrp

N9k-B(config)# feature lacp

N9k-B(config)# feature vpc

N9k-B(config)# system jumbomtu 9216

N9k-B(config)# exit

N9k-B#
```

Configuring VLANs on Nexus 9332PQ Switch A and B

To configure the same VLANs Public, OSD-Cluster, and Backup as we already did in the Cisco UCS Manager GUI, complete the following steps on Switch A and Switch B:

Switch A

```
N9k-A# config terminal

Enter configuration commands, one per line. End with CNTL/Z.

N9k-A(config)# vlan 10

N9k-A(config-vlan)# name Public

N9k-A(config-vlan)# no shut

N9k-A(config-vlan)# exit
```

```
N9k-A(config)# vlan 20

N9k-A(config-vlan)# name OSD-Cluster

N9k-A(config-vlan)# no shut

N9k-A(config-vlan)# exit


N9k-A(config)# interface vlan10

N9k-A(config-if)# description Public

N9k-A(config-if)# no shutdown

N9k-A(config-if)# no ip redirects

N9k-A(config-if)# ip address 192.168.10.253/24

N9k-A(config-if)# no ipv6 redirects

N9k-A(config-if)# hsrp version 2

N9k-A(config-if)# hsrp 10

N9k-A(config-if-hsrp)# preempt

N9k-A(config-if-hsrp)# priority 5

N9k-A(config-if-hsrp)# ip 192.168.10.1

N9k-A(config-if-hsrp)# exit

N9k-A(config-if)# exit


N9k-A(config)# interface vlan20

N9k-A(config-if)# description OSD-Cluster

N9k-A(config-if)# no shutdown

N9k-A(config-if)# no ip redirects

N9k-A(config-if)# ip address 192.168.20.253/24

N9k-A(config-if)# no ipv6 redirects

N9k-A(config-if)# hsrp version 2

N9k-A(config-if)# hsrp 20

N9k-A(config-if-hsrp)# preempt

N9k-A(config-if-hsrp)# priority 5

N9k-A(config-if-hsrp)# ip 192.168.20.1

N9k-A(config-if-hsrp)# exit

N9k-A(config-if)# exit
```

```
N9k-A(config-if)# exit

N9k-A(config)#
```

Switch B

```
N9k-B# config terminal

Enter configuration commands, one per line. End with CNTL/Z.

N9k-B(config)# vlan 10

N9k-B(config-vlan)# name Public

N9k-B(config-vlan)# no shut

N9k-B(config-vlan)# exit

N9k-B(config)# vlan 20

N9k-B(config-vlan)# name OSD-Cluster

N9k-B(config-vlan)# no shut

N9k-B(config-vlan)# exit


N9k-B(config)# interface vlan10

N9k-B(config-if)# description Public

N9k-B(config-if)# no ip redirects

N9k-B(config-if)# ip address 192.168.10.254/24

N9k-B(config-if)# no ipv6 redirects

N9k-B(config-if)# hsrp version 2

N9k-B(config-if)# hsrp 10

N9k-B(config-if-hsrp)# preempt

N9k-B(config-if-hsrp)# priority 2

N9k-B(config-if-hsrp)# ip 192.168.10.1

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

N9k-B(config-if)# exit


N9k-B(config)# interface vlan20

N9k-B(config-if)# description OSD-Cluster

N9k-B(config-if)# no ip redirects

N9k-B(config-if)# ip address 192.168.20.254/24

N9k-B(config-if)# no ipv6 redirects
```

```
N9k-B(config-if)# hsrp version 2

N9k-B(config-if)# hsrp 20

N9k-B(config-if-hsrp)# preempt

N9k-B(config-if-hsrp)# priority 2

N9k-B(config-if-hsrp)# ip 192.168.20.1

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

N9k-B(config-if)# exit

N9k-B(config-if)# exit

N9k-B(config)#
```

Configure vPC and Port Channels on Nexus C9332PQ Switch A and B

To enable vPC and Port Channels on both Switch A and B, complete the following steps:

vPC and Port Channels for Peerlink on Switch A

```
N9k-B# config terminal

Enter configuration commands, one per line. End with CNTL/Z.

N9k-A(config)# vpc domain 2

N9k-A(config-vpc-domain)# peer-keepalive destination 172.25.206.227

Note:

 --------:: Management VRF will be used as the default VRF ::--------

N9k-A(config-vpc-domain)# peer-gateway

N9k-A(config-vpc-domain)# exit


N9k-A(config)# interface port-channel 1

N9k-A(config-if)# description vPC peerlink for N9k-A and N9k-B

N9k-A(config-if)# switchport

N9k-A(config-if)# switchport mode trunk

N9k-A(config-if)# spanning-tree port type network

N9k-A(config-if)# speed 40000

N9k-A(config-if)# vpc peer-link

Please note that spanning tree port type is changed to "network" port type on
vPC peer-link.

This will enable spanning tree Bridge Assurance on vPC peer-link provided the
STP Bridge Assurance
```

```
(which is enabled by default) is not disabled.
N9k-A(config-if)# exit

N9k-A(config)# interface ethernet 1/31
N9k-A(config-if)# description connected to peer N9k-B port 31
N9k-A(config-if)# switchport
N9k-A(config-if)# switchport mode trunk
N9k-A(config-if)# speed 40000
N9k-A(config-if)# channel-group 1 mode active
N9k-A(config-if)# exit

N9k-A(config)# interface ethernet 1/32
N9k-A(config-if)# description connected to peer N9k-B port 32
N9k-A(config-if)# switchport
N9k-A(config-if)# switchport mode trunk
N9k-A(config-if)# speed 40000
N9k-A(config-if)# channel-group 1 mode active
N9k-A(config-if)# exit
vPC and Port Channels for Peerlink on Switch B
N9k-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-B(config)# vpc domain 2
N9k-B(config-vpc-domain)# peer-keepalive destination 172.25.206.226
Note:
 --------:: Management VRF will be used as the default VRF ::--------
N9k-B(config-vpc-domain)# peer-gateway
N9k-B(config-vpc-domain)# exit

N9k-B(config)# interface port-channel 1
N9k-B(config-if)# description vPC peerlink for N9k-A and N9k-B
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
```

```
N9k-B(config-if)# spanning-tree port type network
N9k-B(config-if)# speed 40000
N9k-B(config-if)# vpc peer-link
Please note that spanning tree port type is changed to "network" port type on
vPC peer-link.
This will enable spanning tree Bridge Assurance on vPC peer-link provided the
STP Bridge Assurance
(which is enabled by default) is not disabled.
N9k-B(config-if)# exit

N9k-B(config)# interface ethernet 1/31
N9k-B(config-if)# description connected to peer N9k-A port 31
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# speed 40000
N9k-B(config-if)# channel-group 1 mode active
N9k-B(config-if)# exit

N9k-B(config)# interface ethernet 1/32
N9k-B(config-if)# description connected to peer N9k-A port 32
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# speed 40000
N9k-B(config-if)# channel-group 1 mode active
N9k-B(config-if)# exit
vPC and Port Channels for Uplink from Fabric Interconnect A and B on Switch A
N9k-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-A(config)# interface port-channel 100
N9k-A(config-if)# description vPC for UCS FI-A port 24 & 26
N9k-A(config-if)# vpc 100
N9k-A(config-if)# switchport
N9k-A(config-if)# switchport mode trunk
```

```
N9k-A(config-if)# switchport trunk allowed vlan 10,20

N9k-A(config-if)# spanning-tree port type edge trunk

Edge port type (portfast) should only be enabled on ports connected to a single

 host. Connecting hubs, concentrators, switches, bridges, etc...  to this

 interface when edge port type (portfast) is enabled, can cause temporary bridging loops.

 Use with CAUTION

N9k-A(config-if)# mtu 9216

N9k-A(config-if)# exit


N9k-A(config)# interface port-channel 101

N9k-A(config-if)# description vPC for UCS FI-B port 23 & 25

N9k-A(config-if)# vpc 101

N9k-A(config-if)# switchport

N9k-A(config-if)# switchport mode trunk

N9k-A(config-if)# switchport trunk allowed vlan 10,20

N9k-A(config-if)# spanning-tree port type edge trunk

Edge port type (portfast) should only be enabled on ports connected to a single

 host. Connecting hubs, concentrators, switches, bridges, etc...  to this

 interface when edge port type (portfast) is enabled, can cause temporary bridging loops.

 Use with CAUTION

N9k-A(config-if)# mtu 9216

N9k-A(config-if)# exit


N9k-A(config)# interface ethernet 1/23

N9k-A(config-if)# switchport

N9k-A(config-if)# switchport mode trunk

N9k-A(config-if)# description Uplink from UCS FI-B port 23

N9k-A(config-if)# channel-group 101 mode active

N9k-A(config-if)# exit
```

```
N9k-A(config)# interface ethernet 1/24
N9k-A(config-if)# switchport
N9k-A(config-if)# switchport mode trunk
N9k-A(config-if)# description Uplink from UCS FI-B port 25
N9k-A(config-if)# channel-group 101 mode active
N9k-A(config-if)# exit

N9k-A(config)# interface ethernet 1/25
N9k-A(config-if)# switchport
N9k-A(config-if)# switchport mode trunk
N9k-A(config-if)# description Uplink from UCS FI-A port 24
N9k-A(config-if)# channel-group 100 mode active
N9k-A(config-if)# exit

N9k-A(config)# interface ethernet 1/26
N9k-A(config-if)# switchport
N9k-A(config-if)# switchport mode trunk
N9k-A(config-if)# description Uplink from UCS FI-A port 26
N9k-A(config-if)# channel-group 100 mode active
N9k-A(config-if)# exit
vPC and Port Channels for Uplink from Fabric Interconnect A and B on Switch B
N9k-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-B(config)# interface port-channel 100
N9k-B(config-if)# description vPC for UCS FI-A port 23 & 24
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# switchport trunk allowed vlan 10,20
N9k-B(config-if)# spanning-tree port type edge trunk
Edge port type (portfast) should only be enabled on ports connected to a
single
 host. Connecting hubs, concentrators, switches, bridges, etc...  to this
 interface when edge port type (portfast) is enabled, can cause temporary
bridging loops.
 Use with CAUTION
N9k-B(config-if)# vpc 100
N9k-B(config-if)# mtu 9216
N9k-B(config-if)# exit

N9k-B(config)# interface port-channel 101
N9k-B(config-if)# description vPC for UCS FI-B port 25 & 26
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# switchport trunk allowed vlan 10,20
N9k-B(config-if)# spanning-tree port type edge trunk
Edge port type (portfast) should only be enabled on ports connected to a
single
 host. Connecting hubs, concentrators, switches, bridges, etc...  to this
 interface when edge port type (portfast) is enabled, can cause temporary
bridging loops.
 Use with CAUTION
N9k-B(config-if)# vpc 101
N9k-B(config-if)# mtu 9216
N9k-B(config-if)# exit

N9k-B(config)# interface ethernet 1/23
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# description Uplink from UCS FI-A port 23
N9k-B(config-if)# channel-group 100 mode active
N9k-B(config-if)# exit

N9k-B(config)# interface ethernet 1/24
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
```

```
N9k-B(config-if)# description Uplink from UCS FI-A port 25
N9k-B(config-if)# channel-group 100 mode active
N9k-B(config-if)# exit

N9k-B(config)# interface ethernet 1/25
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# description Uplink from UCS FI-B port 24
N9k-B(config-if)# channel-group 101 mode active
N9k-B(config-if)# exit

N9k-B(config)# interface ethernet 1/26
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# description Uplink from UCS FI-B port 26
N9k-B(config-if)# channel-group 101 mode active
N9k-B(config-if)# exit
```

Verification check of Nexus C9332PQ Configuration for Switch A and B

Switch A

```
N9k-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-A(config)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 2
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
```

```
vPC role                          : secondary
Number of vPCs configured         : 4
Peer Gateway                      : Enabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)

vPC Peer-link status
---------------------------------------------------------------------
id    Port   Status Active vlans
--    ----   ------ -------------------------------------------------
1     Po1    up     1,10,20

vPC status
----------------------------------------------------------------------
id    Port   Status Consistency Reason                     Active vlans
--    ----   ------ ----------- ------                     ------------
100   Po100  up     success     success                    10,20

101   Po101  up     success     success                    10,20

110   Po110  up     success     success                    10,20

111   Po111  up     success     success                    10,20

N9k-A(config)#
N9k-A(config)# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
        p - Up in delay-lacp mode (member)
        M - Not in use. Min-links not met
-----------------------------------------------------------------------------
---
Group Port-       Type     Protocol  Member Ports
      Channel
-----------------------------------------------------------------------------
---
1     Po1(SU)     Eth      LACP      Eth1/31(P)   Eth1/32(P)
100   Po100(SU)   Eth      LACP      Eth1/25(P)   Eth1/26(P)
101   Po101(SU)   Eth      LACP      Eth1/23(P)   Eth1/24(P)
110   Po110(SU)   Eth      LACP      Eth1/15/1(P)  Eth1/15/2(P)  Eth1/15/3(P)
                                     Eth1/15/4(I)
111   Po111(SU)   Eth      LACP      Eth1/16/1(P)  Eth1/16/2(P)  Eth1/16/3(P)
                                     Eth1/16/4(P)
N9k-A(config)#
```

Switch B

```
N9k-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-B(config)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 2
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : primary
Number of vPCs configured         : 4
```

```
Peer Gateway                      : Enabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1,10,20

vPC status
----------------------------------------------------------------------
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- ------                     ------------
100  Po100  up     success     success                    10,20

101  Po101  up     success     success                    10,20

110  Po110  up     success     success                    10,20

111  Po111  up     success     success                    10,20

N9k-B(config)#
N9k-B(config)# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
        p - Up in delay-lacp mode (member)
        M - Not in use. Min-links not met
----------------------------------------------------------------------------
---
Group Port-       Type     Protocol  Member Ports
      Channel
----------------------------------------------------------------------------
---
1     Po1(SU)     Eth      LACP      Eth1/31(P)   Eth1/32(P)
100   Po100(SU)   Eth      LACP      Eth1/23(P)   Eth1/24(P)
101   Po101(SU)   Eth      LACP      Eth1/25(P)   Eth1/26(P)
110   Po110(SU)   Eth      LACP      Eth1/15/1(P)  Eth1/15/2(P)  Eth1/15/3(P)
                                     Eth1/15/4(P)
111   Po111(SU)   Eth      LACP      Eth1/16/1(P)  Eth1/16/2(P)  Eth1/16/3(P)
                                     Eth1/16/4(P)
N9k-B(config)#
```

The formal setup for the Cisco UCS Manager environment and both Cisco Nexus 9332PQ switches is now finished. Next is installing the Red Hat Enterprise Linux 7.3 Operating System.

Installation of Red Hat Enterprise Linux 7.3 Operating System

The following section provides the detailed procedures for installing Red Hat Enterprise Linux 7.3 on Cisco UCS C220 M4S and Cisco UCS S3260 Storage Server. The installation uses the KVM console and virtual Media from Cisco UCS Manager.

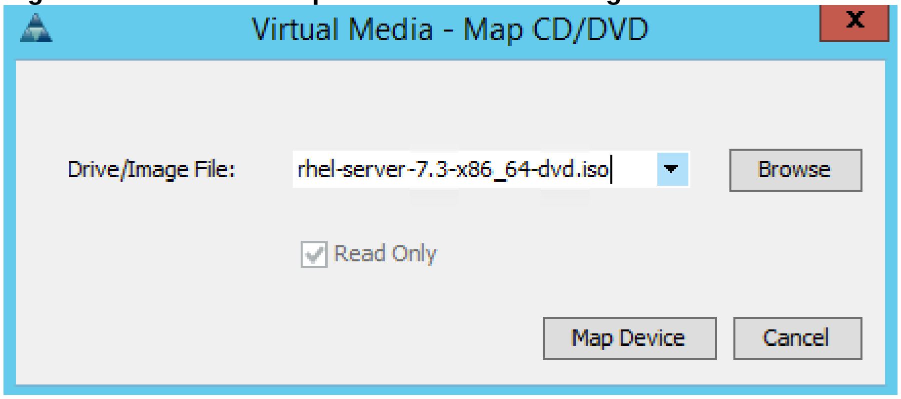
Note: This requires RHEL 7.3 DVD/ISO media for the installation.

Installation of RHEL 7.3 on Cisco UCS C220 M4S

To install Red Hat Linux 7.3 operating system on Cisco UCS C220 M4S, complete the following steps:

1. Log in to the Cisco UCS Manager and select the `Equipment` tab from the left pane.
2. Go to Equipment > Rack-Mounts > Server > Server 1 (Ceph Mon 1) and right-click KVM Console.
3. Launch KVM Console.
4. Click the `Activate Virtual Devices` in the Virtual Media tab.
5. In the KVM window, select the Virtual Media tab and click `Map CD/DVD`.
6. Browse to the Red Hat Enterprise Linux 7.3 installation ISO image and select then `Map Device`.

Figure 76 Red Hat Enterprise Linux 7.3 ISO image



Drive/Image File: rhel-server-7.3-x86_64-dvd.iso

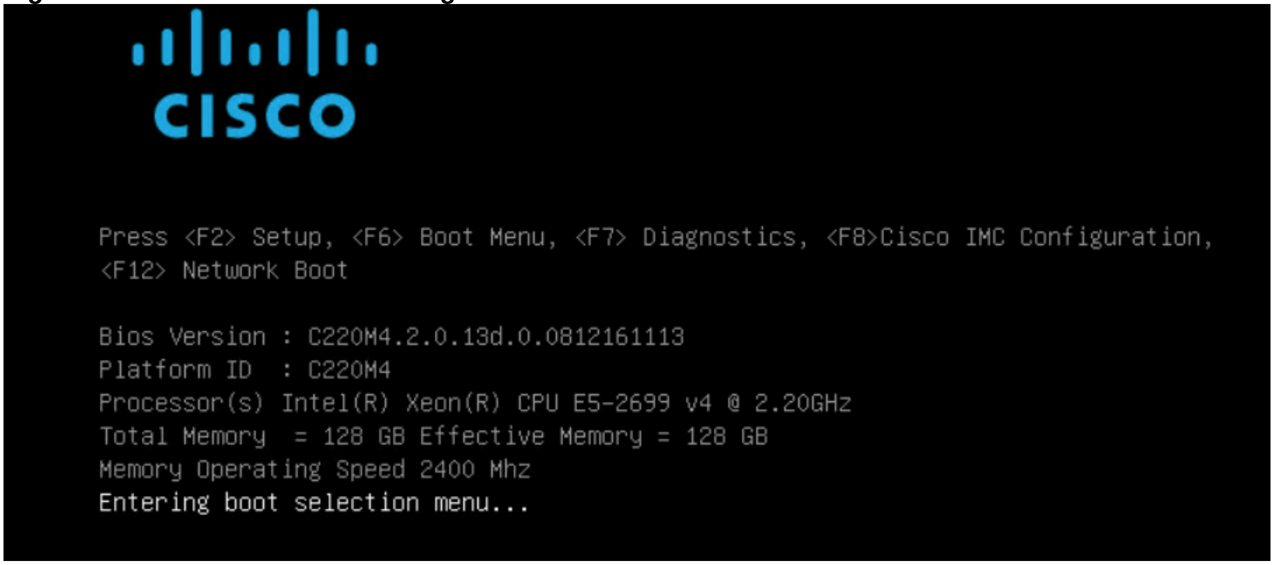
☑ Read Only

7. In the KVM window, select the `Macros > Static Macros > Ctrl-Alt-Del` button in the upper left corner.

8. Click `OK` and then `OK` to reboot the system.

9. In the boot screen with the Cisco Logo, press `F6` for the boot menu.

Figure 77 Boot screen for selecting Boot Menu

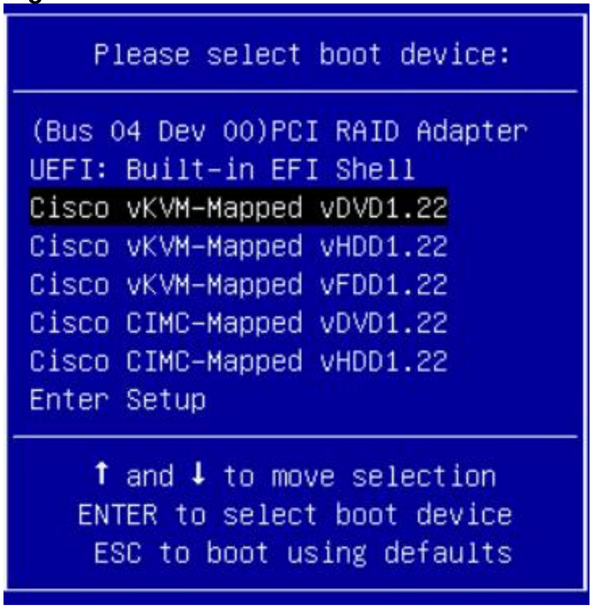


```
Press <F2> Setup, <F6> Boot Menu, <F7> Diagnostics, <F8>Cisco IMC Configuration,
<F12> Network Boot

Bios Version : C220M4.2.0.13d.0.0812161113
Platform ID  : C220M4
Processor(s) Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
Total Memory  = 128 GB Effective Memory = 128 GB
Memory Operating Speed 2400 Mhz
Entering boot selection menu...
```

10. When the Boot Menu appears, select `Cisco vKVM-Mapped vDVD1.22`.

Figure 78 Boot Menu Selection



```
   Please select boot device:

(Bus 04 Dev 00)PCI RAID Adapter
UEFI: Built-in EFI Shell
Cisco vKVM-Mapped vDVD1.22
Cisco vKVM-Mapped vHDD1.22
Cisco vKVM-Mapped vFDD1.22
Cisco CIMC-Mapped vDVD1.22
Cisco CIMC-Mapped vHDD1.22
Enter Setup

   ↑ and ↓ to move selection
  ENTER to select boot device
   ESC to boot using defaults
```

11. When the Red Hat Enterprise Linux 7.3 installer appears, press the Tab button for further configuration options.

We prepared a Linux Kickstart file with all necessary options for an automatic install. The Kickstart file is located on a RHEL server in the same subnet. The content of the Kickstart file can be found in Appendix A. In addition, we configured typical network interface names like eth0 for the default Administration network and the management IP address for the server.

12. At the prompt type:

```
inst.ks=http://192.168.0.100/ceph-ks.cfg net.ifnames=0 biosdevname=0
ip=192.168.0.111::192.168.0.99:255.255.255.0:cephmon1.cisco.com:eth0:none
nameserver=173.36.131.10
```

13. Repeat the previous steps for Ceph Mon 2, Ceph Mon 3, and Ceph Adm.

Installation of RHEL 7.3 on Cisco UCS S3260 Storage Server

To install the RHEL 7.3 on the Cisco UCS S3260 Storage Server, complete the following steps:

1. Log in to the Cisco UCS Manager and select the `Equipment` tab from the left pane.

2. Go to Equipment > Chassis > Chassis 1 (Ceph OSD 1/2) > Server 1 (Ceph OSD 1) and right-click KVM Console.

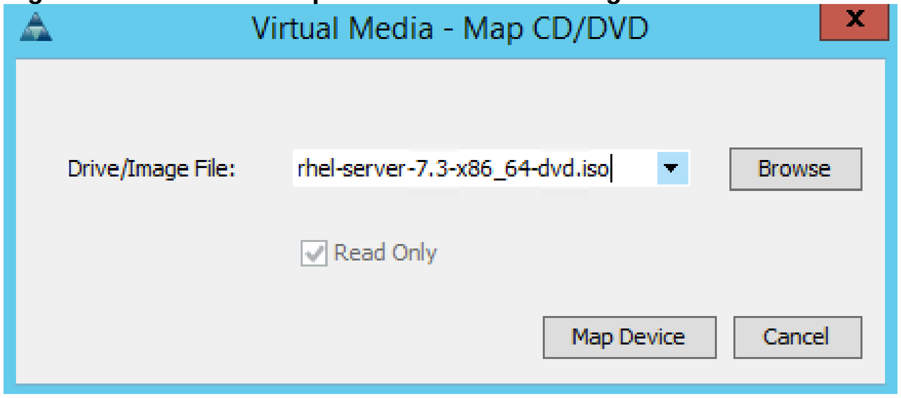
3. Launch KVM Console.

4. Click the `Activate Virtual Devices` in the Virtual Media tab.

5. In the KVM window, select the Virtual Media tab and click `Map CD/DVD`.

6. Browse to the Red Hat Enterprise Linux 7.3 installation ISO image and select then `Map Device`.

Figure 79 Red Hat Enterprise Linux 7.3 ISO image



Virtual Media - Map CD/DVD

Drive/Image File: rhel-server-7.3-x86_64-dvd.iso

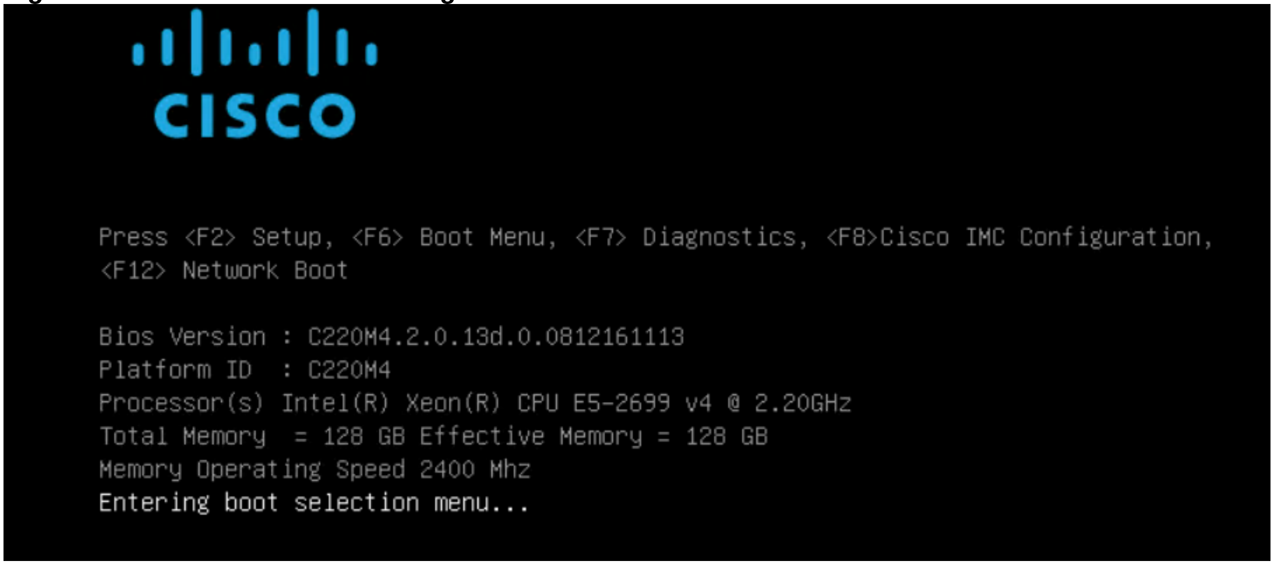
☑ Read Only

7. In the KVM window, select the `Macros > Static Macros > Ctrl-Alt-Del` button in the upper left corner.

8. Click `OK` and then `OK` to reboot the system.

9. In the boot screen with the Cisco Logo, press `F6` for the boot menu.

Figure 80 Boot screen for selecting Boot Menu

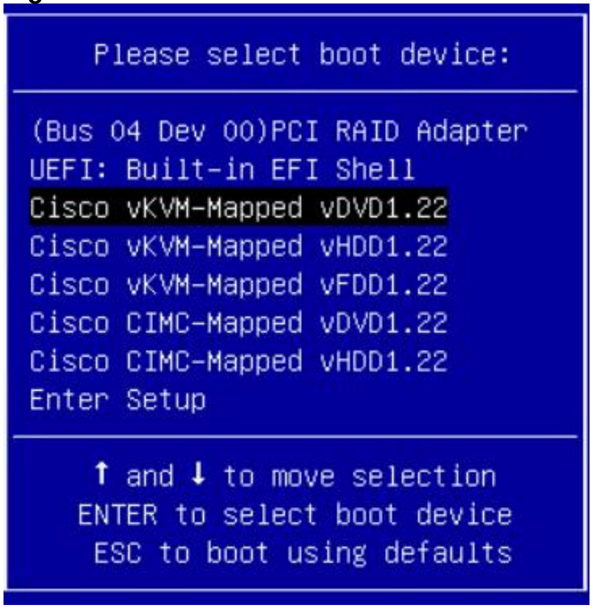


```
Press <F2> Setup, <F6> Boot Menu, <F7> Diagnostics, <F8>Cisco IMC Configuration,
<F12> Network Boot

Bios Version : C220M4.2.0.13d.0.0812161113
Platform ID  : C220M4
Processor(s) Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
Total Memory  = 128 GB Effective Memory = 128 GB
Memory Operating Speed 2400 Mhz
Entering boot selection menu...
```

10. When the Boot Menu appears, select `Cisco vKVM-Mapped vDVD1.22`.

Figure 81 Boot Menu Selection



```
Please select boot device:

(Bus 04 Dev 00)PCI RAID Adapter
UEFI: Built-in EFI Shell
Cisco vKVM-Mapped vDVD1.22
Cisco vKVM-Mapped vHDD1.22
Cisco vKVM-Mapped vFDD1.22
Cisco CIMC-Mapped vDVD1.22
Cisco CIMC-Mapped vHDD1.22
Enter Setup

↑ and ↓ to move selection
ENTER to select boot device
ESC to boot using defaults
```

11. When the Red Hat Enterprise Linux 7.3 installer appears, press the Tab button for further configuration options.

We prepared a Linux Kickstart file with all necessary options for an automatic install. The Kickstart file is located on a server in the same subnet. The content of the Kickstart file for the Cisco UCS S3260 Storage Server can be found in Appendix B. In addition, we configured typical network interface names like eth0 for the default Administration network and the management IP address for the server.

12. At the prompt type:

```
inst.ks=http://192.168.0.100/cephosd-ks.cfg net.ifnames=0 biosdevname=0
ip=192.168.0.120::192.168.0.99:255.255.255.0:cephosd1.cisco.com:eth0:none
nameserver=173.36.131.10
```

13. Repeat the previous steps for Ceph OSD 2-10.

Post-Installation Steps for Red Hat Enterprise Linux 7.3

The node cephadm is responsible for all management and installation of the whole environment. The following steps make sure that all nodes have the same base setup for the following Ceph Pre-requirement installation.

Configure /etc/hosts and Enable Password-Less Login

To configure the /etc/hosts and enable the password-less login, complete the following steps:

1. Modify the /etc/hosts file on cephadm according to Table 12 and include all IP address of all nodes. An example is shown in Appendix C.

Table 12 IP addresses for Ceph Nodes

	Default	Public	Cluster
cephadm	192.168.0.110	192.168.10.110	
cephmon1	192.168.0.111	192.168.10.111	
cephmon2	192.168.0.112	192.168.10.112	
cephmon3	192.168.0.113	192.168.10.113	
cephosd1	192.168.0.120	192.168.10.120	192.168.20.120
cephosd2	192.168.0.121	192.168.10.121	192.168.20.121
cephosd3	192.168.0.122	192.168.10.122	192.168.20.122
cephosd4	192.168.0.123	192.168.10.123	192.168.20.123
cephosd5	192.168.0.124	192.168.10.124	192.168.20.124
cephosd6	192.168.0.125	192.168.10.125	192.168.20.125
cephosd7	192.168.0.126	192.168.10.126	192.168.20.126
cephosd8	192.168.0.127	192.168.10.127	192.168.20.127
cephosd9	192.168.0.128	192.168.10.128	192.168.20.128
cephosd10	192.168.0.129	192.168.10.129	192.168.20.129

2. Login to cephadm and change `/etc/hosts`.

```
# ssh root@192.168.0.110

# vi /etc/hosts
```

3. Enable password-less login to all other nodes.

```
# ssh-keygen
```

4. Press `Enter`, then `Enter` and again `Enter`.

5. Copy `id_rsa.pub` under `/root/.ssh` to all other Ceph nodes.

```
# for i in {1..3}; do ssh-copy-id cephmon${i}; done

# for i in {0..10}; do ssh-copy-id cephosd${i}; done
```

6. Copy /etc/hosts to all nodes

```
# for i in {{11..13},{20..29}}; do scp /etc/hosts 192.168.0.1${i}:/etc/; done
```

Configuring hostnames

1. Configure hostname for cephadm and all other nodes

```
# hostnamectl set-hostname cephadm

# for i in {1..3}; do ssh cephmon${i} “hostnamectl set-hostname cephmon${i}”;
done

# for i in {1..10}; do ssh cephosd${i} “hostnamectl set-hostname
cephosd${i}”; done
```

Creating a Red Hat Enterprise Linux (RHEL) 7.3 and Red Hat Ceph Console Repository

To prepare local repositories for the Red Hat Ceph installation, subscribe to CDN, create a directory with all the required RPMs, run the `createrepo` command and then publish the resulting directory to all Ceph nodes. Follow the procedure for the Ceph repositories https://access.redhat.com/documentation/en/red-hat-ceph-storage/2/single/installation-guide-for-red-hat-enterprise-linux/#enabling_ceph_repositories to download the software:

1. Login to cephadm and subscribe to Red Hat CDN.

```
# ssh root@cephadm

# subscription-manager register

# subscription-manager refresh

# subscription-manager list -available

# subscription-manager attach -pool=<Pool ID for Red Hat 7 Enterprise Server>

# subscription-manager repos --enable=rhel-7-server-rpms
```

2. Create two directories that will contain the repositories.

```
# mkdir -p /var/www/html/rhel-server

# mkdir -p /var/www/html/rhscon2
```

3. Create a directory for Red Hat Enterprise Linux 7 and sync all data.

```
# mkdir -p /var/www/html/rhel-server

# reposync --gpgcheck -l --repoid=rhel-7-server-rpms --
download_path=/var/www/html/rhel-server --downloadcomps --download-metadata
```

4. Create the mount directory for the Red Hat Storage Console ISO.

Note: This requires the ISO image Red Hat Storage Console.

```
# mkdir -p /mnt/rhscon2

# mount -o loop /tmp/rhscon-2.0-rhel-7-x86_64.iso /mnt/rhscon2
```

5. Copy the contents of the ISO to the `/var/www/html/rhscon2` directory.

```
# cp -r /mnt/rhscon2/* /var/www/html/rhscon2
```

6. Create a .repo file to enable the use of the yum command.

```
vi /var/www/html/ceph.repo
[rhel-7-server-rpms]
baseurl = http://192.168.0.110/rhel-server
name = Red Hat Enterprise Linux 7 Server (RPMs)
enabled = 1
gpgcheck = 0

[rhel-7-server-rhceph-2-console-rpms]
baseurl = http://192.168.0.110/rhscon2
name = Red Hat Ceph Storage Console 2 for Red Hat Enterprise Linux 7 Server (RPMs)
enabled = 1
gpgcheck = 0
```

7. Run createrepo on all repositories to create the repo database:

```
# cd /var/www/html/rhel-server
# createrepo -v .
# cd /var/www/html/rhscon2
# createrepo -v .
```

8. Copy ceph.repo to all Ceph nodes:

```
# for i in {{11..13},{20..29}}; do scp /var/www/html/ceph.repo 192.168.0.1${i}:/etc/yum.repos.d/; done
```

9. Place Red Hat Ceph Storage ISO into /tmp directory

Note: This requires the ISO image Red Hat Ceph Storage.

```
# mv rhceph-2.1-rhel-7-x86_64.iso /tmp
```

Setting up ClusterShell

ClusterShell (or clush) is the cluster-wide shell that runs commands on several hosts in parallel. To setup the ClusterShell, complete the following steps:

1. From cephadm download Cluster shell (clush) and copy and install it on cephadm. Cluster shell is available from EPEL (Extra Packages for Enterprise Linux) repository.

```
# yum install yum-plugin-downloadonly
# yum install --downloadonly --downloaddir=/root clustershell
# yum clean all

# yum repolist

# yum -y install clustershell-1.7.2-1.el7.noarch.rpm
```

2. Edit /etc/clustershell/groups.d/local.cfg file to include hostnames for all the nodes of the cluster. This set of hosts is taken when running clush with the ‘-a’ option. For a 13 node cluster as in our CVD, set groups file as follows:

```
# vi /etc/clustershell/groups.d/local.cfg

all: cephmon[1-3] cephosd[1-10]
```

Enable Local Repository on All Nodes

1. Purge the yum caches and enable the repositories on all nodes

```
# clush -a -B yum clean all

# clush -a -B yum repolist
```

Install Latest Network Driver

To install the latest network driver for performance and updates, download the latest ISO image, by completing the following steps:

The ISO image for Cisco UCS C220 M4S and S3260 Storage Server have the same network driver for RHEL 7.3.

1. Mount the ISO image on cephadm, go to /Network/Cisco/VIC/RHEL/RHEL7.3 and copy the file kmod-enic-2.3.0.31-rhel7u3.el7.x86_64.rpm to cephadm.

```
# mkdir -p /mnt/cisco

# mount -o loop /tmp/ucs-cxxx-drivers-linux.2.0.13c.iso /mnt/cisco/

# cd /mnt/cisco/Network/Cisco/VIC/RHEL/RHEL7.3/

# clush -a -b -c /mnt/cisco/Network/Cisco/VIC/RHEL/RHEL7.3/kmod-enic-
2.3.0.31-rhel7u3.el7.x86_64.rpm --dest=/tmp/
```

2. Install the VIC driver on cephadm and all other nodes.

```
# rpm -ivh /tmp/kmod-enic-2.3.0.31-rhel7u3.el7.x86_64.rpm

# clush -a -b “rpm -ivh /tmp/kmod-enic-2.3.0.31-rhel7u3.el7.x86_64.rpm”
```

3. Verify the installation of the VIC driver.

```
# clush -a -b “modinfo enic | head -5”
```

Create VLAN Interface for Network Public and Cluster on All Nodes

Provide the following IP address to the specific nodes (as shown in Table 10 :

1. To create the VLAN interface for the Public network on the node cephadm

```
# nmcli con add type vlan con-name eth1 dev eth1 id 10 ip4 192.168.10.110/24
```

Repeat the steps for all other nodes with the IP address provided by Table 12 .

```
# ssh cephmon1 “nmcli con add type vlan con-name eth1 dev eth1 id 10 ip4 
192.168.10.111/24”
```

2. To create a VLAN interface for the Cluster network on the nodes cephosd1-10

```
# ssh cephosd1 “nmcli con add type vlan con-name eth2 dev eth2 id 20 ip4 
192.168.20.120/24”
```

Repeat the steps for all other OSD nodes with the appropriate IP address from Table 12 .

Preparation of All Nodes for Ceph Installation

Before installing Ceph on any node, you need to prepare all nodes with certain configurations. Follow the procedure for cephadm, cephmon1-3, and cephosd1-10 according to https://access.redhat.com/documentation/en/red-hat-ceph-storage/2/single/installation-guide-for-red-hat-enterprise-linux/#prerequisites. A summary of the prerequisites for the whole installation with the appropriate changes to the current environment is listed below.

Step 1 – Update all Ceph Nodes

1. Login to cephadm and update RHEL.

```
# ssh cephadm

# yum -y update

# clush -a -b yum -y update
```

Step 2 – Configuring Firewall

To enable the Firewall on all Monitor and OSD Nodes according to https://access.redhat.com/documentation/en/red-hat-ceph-storage/2/single/installation-guide-for-red-hat-enterprise-linux/#configuring_firewall, complete the following steps:

1. On cephadm:

```
# clush -a -b “systemctl enable firewalld”

# clush -a -b “systemctl start firewalld”

# clush -a -b “systemctl status firewalld”

# clush -w cephmon[1-3] -b “firewall-cmd --zone=public --add-port=6789/tcp”

# clush -w cephmon[1-3] -b “firewall-cmd --zone=public --add-port=6789/tcp --
permanent”

# clush -w cephmon[1-3] -b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.10.0/24"\” port 
protocol=\”"tcp"\” port=\”"6789"\” accept"\””

# clush -w cephmon[1-3] -b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.10.0/24"\” port 
protocol=\”"tcp"\” port=\”"6789"\” accept"\” --permanent”
```

```
# clush –w cephosd[1-10] –b “firewall-cmd --zone=public --add-port=6800-
7300/tcp”

# clush –w cephosd[1-10] –b “firewall-cmd --zone=public --add-port=6800-
7300/tcp –permanent”

# clush –w cephosd[1-10] –b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.10.0/24"\” port
protocol=\”"tcp"\” port=\”"6800-7300"\” accept"\””

# clush –w cephosd[1-10] –b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.10.0/24"\” port
protocol=\”"tcp"\” port=\”"6800-7300"\” accept"\” --permanent”

# clush –w cephosd[1-10] –b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.20.0/24"\” port
protocol=\”"tcp"\” port=\”"6800-7300"\” accept"\””

# clush –w cephosd[1-10] –b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.20.0/24"\” port
protocol=\”"tcp"\” port=\”"6800-7300"\” accept"\” --permanent”
```

Step 3 – Configuring Network Time Protocol

In our Kickstart installation file, we already included a time server. According to https://access.redhat.com/documentation/en/red-hat-ceph-storage/2/single/installation-guide-for-red-hat-enterprise-linux/#configuring_network_time_protocol now enable Network Time Protocol on all servers and configure them to use all the same source.

1. Install NTP on all servers:

```
# yum –y install ntp

# clush –a –b yum –y install ntp
```

2. Configure /etc/ntp.conf on cephadm node only with the following contents:

```
# vi /etc/ntp.conf

driftfile /var/lib/ntp/drift

restrict 127.0.0.1

restrict -6 ::1

server 127.127.1.0

fudge 127.127.1.0 stratum 10

includefile /etc/ntp/crypto/pw

keys /etc/ntp/keys
```

3. Start the ntpd daemon on cephadm:

```
# systemctl enable ntpd

# systemctl start ntpd

# systemctl status ntpd
```

4. Create /root/ntp.conf on cephadm and copy it to all nodes:

```
# vi /root/ntp.conf
server cephadm
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
# clush -a -b -c /root/ntp.conf --dest=/etc
```

5. Synchronize the time and restart NTP daemon on all Ceph nodes:

```
# clush -a -b "service ntpd stop"
# clush -a -b "ntpdate cephadm"
# clush -a -b "service ntpd start"
# clush -a -b "systemctl enable ntpd"
```

Step 4 – Create an Ansible User

Since we use Ansible as the deployment method for Red Hat Ceph Storage, we need to provide the Ansible User with passwordless `root` privileges. In our Kickstart file, we already created a user `cephadm`. According to https://access.redhat.com/documentation/en/red-hat-ceph-storage/2/single/installation-guide-for-red-hat-enterprise-linux/#creating_an_ansible_user_ansible_deployment_only the user `cephadm` now needs root privileges.

1. On cephadm:

```
# cat << EOF >/etc/sudoers.d/cephadm
>>cephadm ALL = (root) NOPASSWD:ALL
>>EOF
#
# chmod 0440 /etc/sudoers.d/cephadm
```

2. Copy the file to cephmon1-3 and cephosd1-10:

```
# clush -a -b -c /etc/sudoers.d/cephadm
```

Step 5 – Enabling Password-Less SSH

The user `cephadm` needs password-less access from the administration node cephadm to all Monitor and OSD nodes, according to https://access.redhat.com/documentation/en/red-hat-ceph-storage/2/single/installation-guide-for-red-hat-enterprise-linux/#enabling_password_less_ssh_ansible_deployment_only. To enable this function, complete the following steps:

1. On the cephadm node log in as user `cephadm`.

```
$ ssh-keygen
```

2. Press `Enter`, then `Enter` and again `Enter`.

3. Copy `id_rsa.pub` under `/root/.ssh` to cephmon1.

```
$ ssh-copy-id cephadm@cephmon1
```

4. Repeat the steps for cephmon2-3 and cephosd1-10.

5. Create a file ~/.ssh/config according to Appendix D.

6. Correct the permissions of ~/.ssh/config.

```
$ chmod 600 ~/.ssh/config
```

Step 6 – Check Disk Devices and Prepare Disks for Ceph

As a preparation for the next section where Ceph is going to be installed, all disks on all OSD nodes have to be checked for their device name. In this step, the disks have to be zapped to make sure that they are clean for the following Ceph installation. To do this, complete the following steps:

1. Reboot all OSD nodes to make sure that the disks appear as the same devices:

```
# clush –a –b “reboot”
```

2. Check on cephosd1-10 the disk device names:

```
# clush –w cephosd[1-10] –b “lsblk”

NAME                       MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT

sda                          8:0    0 372.6G  0 disk

sdb                          8:16   0 372.6G  0 disk

sdc                          8:32   0 372.6G  0 disk

sdd                          8:48   0 372.6G  0 disk

sde                          8:64   0 110.8G  0 disk

├─sde1                       8:65   0   500M  0 part /boot

└─sde2                       8:66   0 110.3G  0 part

  ├─rhel_cephosd1-root 253:0    0    50G  0 lvm  /

  ├─rhel_cephosd1-swap 253:1    0     4G  0 lvm

  └─rhel_cephosd1-home 253:2    0  56.3G  0 lvm  /home

sdf                          8:80   0   5.5T  0 disk

sdg                          8:96   0   5.5T  0 disk

sdh                          8:112  0   5.5T  0 disk
```

```
   sdi                      8:128  0   5.5T  0 disk
   sdj                      8:144  0   5.5T  0 disk
   sdk                      8:160  0   5.5T  0 disk
   sdl                      8:176  0   5.5T  0 disk
   sdm                      8:192  0   5.5T  0 disk
   sdn                      8:208  0   5.5T  0 disk
   sdo                      8:224  0   5.5T  0 disk
   sdp                      8:240  0   5.5T  0 disk
   sdq                     65:0    0   5.5T  0 disk
   sdr                     65:16   0   5.5T  0 disk
   sds                     65:32   0   5.5T  0 disk
   sdt                     65:48   0   5.5T  0 disk
   sdu                     65:64   0   5.5T  0 disk
   sdv                     65:80   0   5.5T  0 disk
   sdw                     65:96   0   5.5T  0 disk
   sdx                     65:112  0   5.5T  0 disk
   sdy                     65:128  0   5.5T  0 disk
   sdz                     65:144  0   5.5T  0 disk
   sdaa                    65:160  0   5.5T  0 disk
   sdab                    65:176  0   5.5T  0 disk
   sdac                    65:192  0   5.5T  0 disk
# clush -w cephosd[1-10] -b “yum -y install gdisk”
```

3. Start zapping SSDs

```
# clush -w cephosd[1-10] -b “for i in {a..d}; do sgdisk -Z -g -o /dev/sd\${i};
done”
```

4. Start zapping HDDs

```
# clush -w cephosd[1-10] -b “for i in {{f..z},aa,ab,ac}; do sgdisk -Z -g -o
/dev/sd\${i}; done”
```

Red Hat Ceph Storage Installation via Ansible

The Red Hat Ceph Storage installation via Ansible requires a few configurations steps, but can be deployed afterwards by using one single command. It is important to prepare all Monitor and OSD nodes before to get a clean and correct installation of the environment.

The Red Hat Ceph Storage installation via Ansible is available here: https://access.redhat.com/documentation/en/red-hat-ceph-storage/2/single/installation-guide-for-red-hat-enterprise-linux/#installing_red_hat_ceph_storage_using_ansible.

All changes for the current installation are documented below.

Configure Ceph Global Settings

To configure the Ceph global settings, complete the following steps:

1. Create an ansible hosts file for the environment under `/etc/ansible/hosts`:

```
# This is the default ansible 'hosts' file.
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

[mons]
cephmon[1:3]

[osds]
cephosd[1:10]
```

2. Configure `/usr/share/ceph-ansible/group_vars/all` for the environment as documented below. The whole configuration file can be found under Appendix E.

```
###########

# GENERAL #

###########

fetch_directory: ~/ceph-ansible-keys

cluster: ceph # cluster name


###################

# Stable Releases #

###################

ceph_stable_rh_storage_version: 2

ceph_stable_rh_storage_iso_install: true # usually used when nodes don't have
access to cdn.redhat.com

ceph_stable_rh_storage_iso_path: /tmp/rhceph-2.1-rhel-7-x86_64.iso

ceph_stable_rh_storage_mount_path: /tmp/rh-storage-mount
```

```
ceph_stable_rh_storage_repository_path: /tmp/rh-storage-repo # where to copy
iso's content


######################

# CEPH CONFIGURATION #

######################

generate_fsid: true

cephx: true

max_open_files: 131072

monitor_interface: eth1.10

journal_size: 30000

public_network: 192.168.10.0/24

cluster_network: 192.168.20.0/24


###################

# CONFIG OVERRIDE #

###################

ceph_conf_overrides:

global:

    cephx require signatures: true

    cephx cluster require signatures: true

    osd pool default pg num: 128

    osd pool default pgp num: 128

    mon osd down out interval: 600

    mon osd min down reporters: 7

    mon clock drift allowed: 0.15

    mon clock drift warn backoff: 30

    mon osd report timeout: 900

    mon pg warn max per osd: 0

    mon osd allow primary affinity: true

osd:

    filestore merge threshold: 40
```

```
        filestore split multiple: 8
        osd op threads: 8
        filestore op threads: 8
        osd recovery max active: 5
        osd max backfills: 2
        osd recovery op priority: 63
        osd recovery max chunk: 1048576
        osd scrub sleep: 0.1
        osd disk thread ioprio class: idle
        osd disk thread ioprio priority: 0
        osd deep scrub stride: 1048576
        osd scrub chunk max: 5
    client:
        rbd concurrent management ops: 20
        rbd default map options: rw
        rbd default format: 2
os_tuning_params:
  - { name: kernel.pid_max, value: 4194303 }
  - { name: fs.file-max, value: 26234859 }
  - { name: vm.zone_reclaim_mode, value: 0 }
  - { name: vm.vfs_cache_pressure, value: 50 }
  - { name: vm.min_free_kbytes, value: "{{ vm_min_free_kbytes }}" }
```

3. Leave /usr/share/ceph-ansible/group_vars/mons untouched.

Configure Ceph OSD Settings

Change the configuration of /usr/share/ceph-ansible/group_vars/osds as follows. The whole configuration file can be found in Appendix F.

```
##############
# CEPH OPTIONS
##############
devices:
  - /dev/sdf
  - /dev/sdg
  - /dev/sdh
  - /dev/sdi
  - /dev/sdj
  - /dev/sdk
  - /dev/sdl
  - /dev/sdm
  - /dev/sdn
  - /dev/sdo
  - /dev/sdp
  - /dev/sdq
  - /dev/sdr
  - /dev/sds
  - /dev/sdt
  - /dev/sdu
  - /dev/sdv
  - /dev/sdw
  - /dev/sdx
  - /dev/sdy
  - /dev/sdz
  - /dev/sdaa
  - /dev/sdab
  - /dev/sdac
journal_collocation: false
raw_multi_journal: true
raw_journal_devices:
  - /dev/sda
  - /dev/sda
  - /dev/sda
  - /dev/sda
  - /dev/sda
  - /dev/sda
  - /dev/sdb
  - /dev/sdb
  - /dev/sdb
  - /dev/sdb
  - /dev/sdb
  - /dev/sdb
  - /dev/sdc
  - /dev/sdc
  - /dev/sdc
  - /dev/sdc
  - /dev/sdc
  - /dev/sdc
  - /dev/sdd
  - /dev/sdd
  - /dev/sdd
  - /dev/sdd
```

```
     - /dev/sdd
     - /dev/sdd
```

Deploy Red Hat Ceph Storage via Ansible

As a final step, deploy the cluster via Ansible by completing the following steps:

```
# cd /usr/share/ceph-ansible

# cp site.yml.sample ceph.yml

# ansible-playbook ceph.yml
```

Your final result of the ansible-playbook script should look like the following:

```
PLAY RECAP
*********************************************************************

cephmon1                    : ok=91   changed=20   unreachable=0    failed=0

cephmon2                    : ok=91   changed=20   unreachable=0    failed=0

cephmon3                    : ok=91   changed=20   unreachable=0    failed=0

cephosd1                    : ok=164  changed=16   unreachable=0    failed=0

cephosd10                   : ok=164  changed=16   unreachable=0    failed=0

cephosd2                    : ok=164  changed=16   unreachable=0    failed=0

cephosd3                    : ok=164  changed=16   unreachable=0    failed=0

cephosd4                    : ok=164  changed=16   unreachable=0    failed=0

cephosd5                    : ok=164  changed=16   unreachable=0    failed=0

cephosd6                    : ok=164  changed=16   unreachable=0    failed=0

cephosd7                    : ok=164  changed=16   unreachable=0    failed=0

cephosd8                    : ok=164  changed=16   unreachable=0    failed=0

cephosd9                    : ok=164  changed=16   unreachable=0    failed=0
```

Final Check of Ceph Deployment

To verify the correct deployment of the Ceph Cluster, complete the following step:

```
[root@cephadm ceph-ansible]# ceph -s
    cluster 0e43585e-dbec-445b-867e-025b2a20ae0e
     health HEALTH_OK
     monmap e1: 3 mons at
{cephmon1=192.168.10.111:6789/0,cephmon2=192.168.10.112:6789/0,cephmon3=192.168.1
0.113:6789/0}
            election epoch 4, quorum 0,1,2 cephmon1,cephmon2,cephmon3
     osdmap e607: 240 osds: 240 up, 240 in
            flags sortbitwise
      pgmap v7700: 64 pgs, 1 pools, 0 bytes data, 0 objects
```

```
            23668 MB used, 1309 TB / 1309 TB avail
                  64 active+clean
```

There should be 240 OSDs for 240 physical disks installed and one default pool with 64 Placement Groups. You have now deployed your Red Hat Ceph Storage Cluster on Cisco UCS.

Operational Guide to Extend a Ceph Cluster with Cisco UCS

Adding Cisco UCS S3260 as Ceph OSD Nodes

One of the benefits of working with Cisco UCS Manager is the simple and fast way to extend a current Ceph cluster with additional nodes like Monitor, OSD or RADOS Gateway. In this example the current Ceph cluster will be enlarged with one more Cisco UCS S3260 Storage Server or two more Cisco UCS C3x60 M4 nodes, working as OSD nodes.

The technical specifications for the additional S3260 chassis are identical with the already installed chassis, adding additional 48 x 6 TB (288 TB) capacity to the Ceph cluster.

The following steps describe the procedure to add one more S3260 chassis with two C3x60 M4 nodes.

Enable Fabric Interconnect Ports for Server

To enable server ports for the S3260 chassis after connecting it to both Fabric Interconnects, complete the following steps:

1. Select the `Equipment` tab on the left site.
2. Select Equipment > Fabric Interconnects > Fabric Interconnect A (subordinate) > Fixed Module.
3. Click `Ethernet Ports` section.
4. Select Ports 11 and 12, right-click and then select `Configure as Server Port`.
5. Click `Yes` and then `OK`.
6. Repeat the same steps for Fabric Interconnect B.

Label Chassis for Identification

For a better identification, label the chassis by completing the following steps:

1. Select the `Equipment` tab on the left site.
2. Select Chassis > Chassis 6.
3. In the `Properties` section on the right go to `User Label` and add `Ceph OSD 11/12` to the field.

Label each Server for Identification

For a better identification, label each server by completing the following steps:

1. Select the `Equipment` tab on the left site.

   Select `Chassis > Chassis 6 > Server 1`.
2. In the `Properties` section on the right go to `User Label` and add `Ceph OSD 11` to the field.
3. Repeat the previous steps for `Server 2` of `Chassis 6` and label it `Ceph OSD 12`.

Add Ceph OSD Nodes to Server Pool

To add both nodes to a specific Server Pool, complete the following steps:

1. Select the `Servers` Tab in the left pane in the Cisco UCS Manager GUI.

2. Go to Servers > Pools > root > Server Pools and choose the pool S3260-Node1.

3. Click `Add Servers` in the right window, choose `Chassis ID 6, Slot ID 1` and click `>>` to add them to the pool.

4. Click `OK` and then `OK`.

5. Repeat the same steps for server pool `S3260-Node2` and `Chassis ID 6, Slot ID 2`.

Create Chassis Profile from Template

To create the Chassis Profile from the previous created Chassis Profile Template, complete the following steps:

1. Select the `Chassis` tab in the left pane of the Cisco UCS Manager GUI.

2. Go to Chassis > Chassis Profiles and right-click Create Chassis Profiles from Template.

3. Type in `S3260-Dual-` in the `Name` field.

4. Type in 6 as Name Suffix Starting Number.

5. Enter `1` for the `Number of Instances` for all connected Cisco UCS S3260 Storage Server.

6. Choose UCS-S3260 under Chassis Profile Template.

7. Click `OK` and then `OK`.

Associate Chassis Profile

To associate the previous created Chassis Profile, complete the following steps:

1. Select the `Chassis` tab in the left pane of the Cisco UCS Manager GUI.

2. Go to Chassis > Chassis Profiles and select S3260-Dual-6.

3. Right-click Change Chassis Profile Association.

4. Under Chassis Assignment, select `Select existing Chassis`.

5. Under Available Chassis, select ID `6`.

6. Click `OK` and then `Yes` and then `OK`.

7. Under `Pending Activities` in the top right corner select `Chassis Profiles` then select `Acknowledge`, click `OK` and then `OK`.

Setting Disks for Cisco UCS S3260 Storage Server to Unconfigured-Good with Cisco UCS PowerTool

To convert all top-loaded HDDs from the additional chassis, complete the following steps:

1. Start a PowerShell CLI and start the previous downloaded script “Convert all disks to Unconfigured-Good for UCS Domain(s)”.

2. Type in the Cluster IP of your Cisco UCS Manager.

3. Type in your password.

4. Type in `Y` for converting your disks from JBOD to Unconfigured-Good.

Create Service Profiles from Template

We are now going to create the appropriate Service Profiles from the previous Service Profile Templates. Please complete the following steps to create the first profile for the top node of the Cisco UCS S3260 Storage Server:

1. Select `Servers` from the left pane of the Cisco UCS Manager GUI.

2. Go to Servers > Service Profiles and right-click Create Service Profile from Template.

3. Type in Ceph-OSD-Node1-6 in the Name Prefix field.

4. Choose `UCS-S3260-Node1` as the `Service Profile Template` you created before for the top node of the Cisco UCS S3260 Storage Server.

5. Click `OK` and then `OK`.

6. Repeat the previous steps for the next Service Profile for the bottom node of the Cisco UCS S3260 Storage Server and choose the Service Profile Template `UCS-S3260-Node2` you created before for the bottom node of the Cisco UCS S3260 Storage Server.

Installation of RHEL 7.3 on Cisco UCS S3260 Storage Server

To install RHEL 7.3 on Cisco UCS S3260 Storage Server, complete the following steps:

1. Log in to the Cisco UCS Manager and select the `Equipment` tab from the left pane.

2. Go to Equipment > Chassis > Chassis 6 (Ceph OSD 11/12) > Server 1 (Ceph OSD 11) and right-click KVM Console.

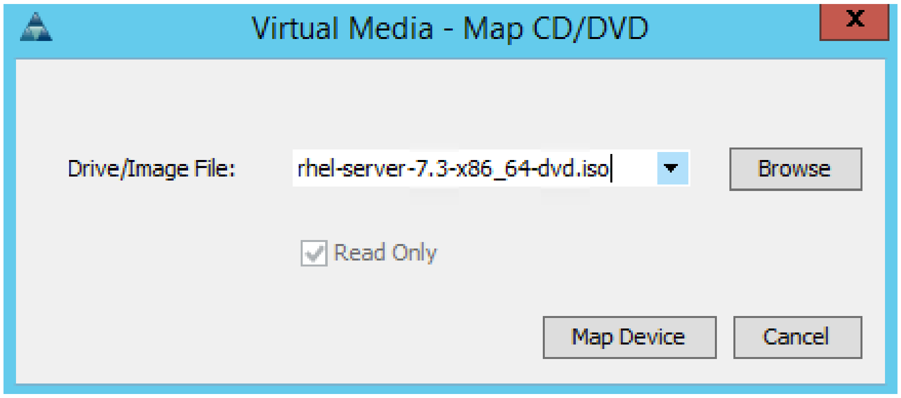
3. Launch KVM Console.

4. Click the `Activate Virtual Devices` in the Virtual Media tab.

5. In the KVM window, select the Virtual Media tab and click `Map CD/DVD`.

6. Browse to the Red Hat Enterprise Linux 7.3 installation ISO image and select then `Map Device`.

Figure 82 Red Hat Enterprise Linux 7.3 ISO image



Drive/Image File: rhel-server-7.3-x86_64-dvd.iso

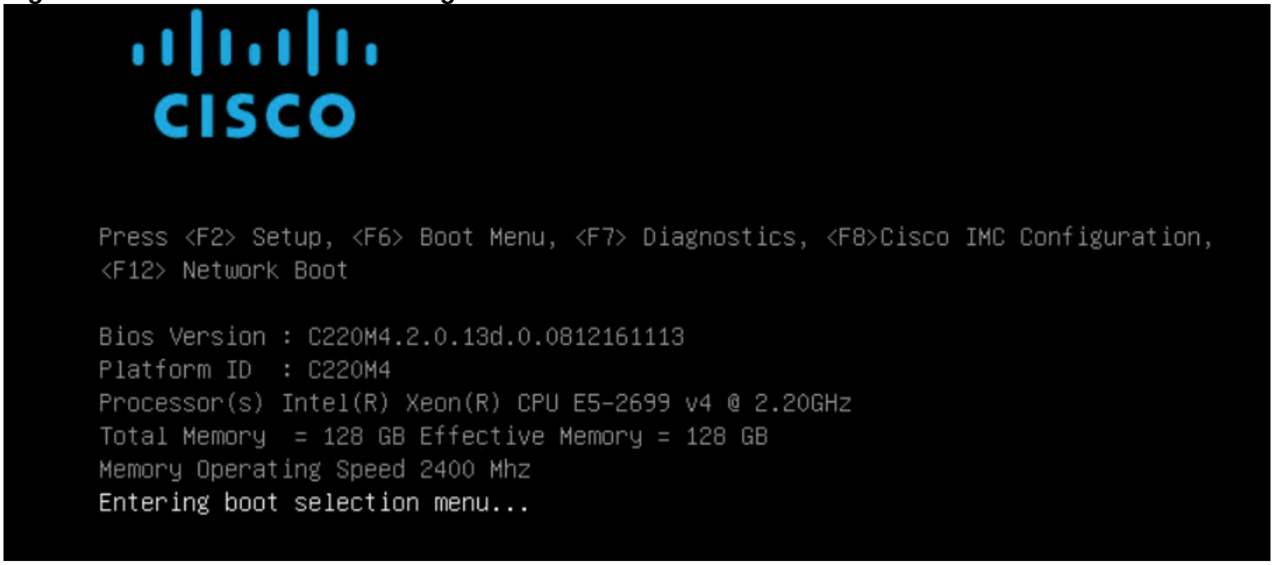
☑ Read Only

7. In the KVM window, select the `Macros > Static Macros > Ctrl-Alt-Del` button in the upper left corner.

8. Click `OK` and then `OK` to reboot the system.

9. In the boot screen with the Cisco Logo, press `F6` for the boot menu.

Figure 83 Boot screen for selecting Boot Menu

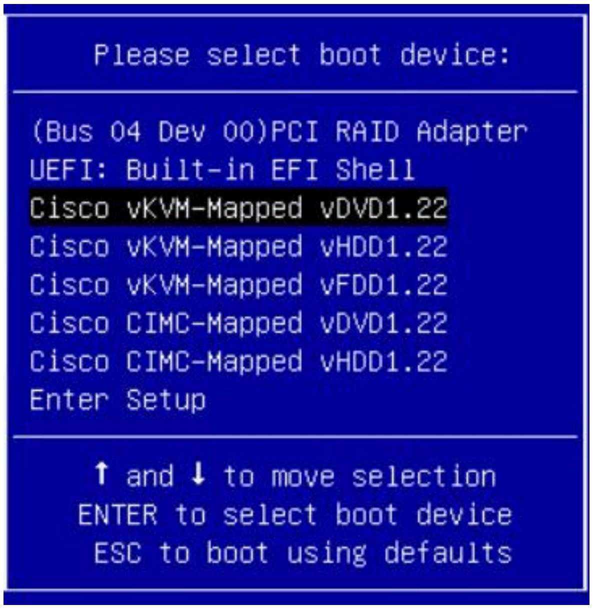


```
Press <F2> Setup, <F6> Boot Menu, <F7> Diagnostics, <F8>Cisco IMC Configuration,
<F12> Network Boot

Bios Version : C220M4.2.0.13d.0.0812161113
Platform ID  : C220M4
Processor(s) Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
Total Memory  = 128 GB Effective Memory = 128 GB
Memory Operating Speed 2400 Mhz
Entering boot selection menu...
```

10. When the Boot Menu appears, select `Cisco vKVM-Mapped vDVD1.22`.

Figure 84 Boot Menu Selection



```
        Please select boot device:

  (Bus 04 Dev 00)PCI RAID Adapter
  UEFI: Built-in EFI Shell
  Cisco vKVM-Mapped vDVD1.22
  Cisco vKVM-Mapped vHDD1.22
  Cisco vKVM-Mapped vFDD1.22
  Cisco CIMC-Mapped vDVD1.22
  Cisco CIMC-Mapped vHDD1.22
  Enter Setup

        ↑ and ↓ to move selection
     ENTER to select boot device
      ESC to boot using defaults
```

11. When the Red Hat Enterprise Linux 7.3 installer appears, press the Tab button for further configuration options.

Note: Use the previous created Kickstart file for Cisco UCS S3260 Storage Server.

12. At the prompt type.

```
inst.ks=http://192.168.0.100/cephosd-ks.cfg net.ifnames=0 biosdevname=0
ip=192.168.0.130::192.168.0.99:255.255.255.0:cephosd11.cisco.com:eth0:none
nameserver=173.36.131.10
```

13. Repeat the previous steps for Ceph OSD 12 with the IP address 192.168.0.131.

Configure /etc/hosts and Enable Password-Less Login

To configure /etc/hosts and enable a password-less login, complete the following steps:

1. Modify the /etc/hosts file on cephadm according to Table 13 and include all IP address of all nodes. An example is shown in Appendix C.

Table 13 IP Address for Ceph OSD Nodes

	Default	Public	Cluster
cephosd11	192.168.0.130	192.168.10.130	192.168.20.130
cephosd12	192.168.0.131	192.168.10.131	192.168.20.131

2. Login to cephadm and change `/etc/hosts.`

```
# ssh root@cephadm
```

```
# vi /etc/hosts
```

3. Copy `id_rsa.pub` under `/root/.ssh` to all other Ceph nodes.

```
# for i in {11,12}; do ssh-copy-id cephosd${i}; done
```

4. Change Cluster Shell configuration file and add both Ceph OSD nodes.

```
# vi /etc/clustershell/groups.d/local.cfg
```

```
all: cephmon[1-3] cephosd[1-12]
```

5. Copy /etc/hosts to all Ceph nodes

```
# clush -a -b -c /etc/hosts
```

Configure Hostnames

1. Configure hostname for cephosd11,12

```
# for i in {11,12}; do ssh cephosd${i} “hostnamectl set-hostname
cephosd${i}”; done
```

Copy Local Repository File and Enable Repositories

1. Copy the repository file to both Ceph OSD nodes and enable repositories.

```
# clush -w cephosd[11-12] -b -c /etc/yum.repos.d/ceph.repo
```

```
# clush -w cephosd[11-12] -b yum clean all
```

```
# clush -w cephosd[11-12] -b yum repolist
```

Install Latest Network Driver

1. Copy the network driver from cephadm to cephosd11, 12 and install it.

```
# clush -w cephosd[11-12] -b -c /tmp/kmod-enic-2.3.0.31-
rhel7u3.el7.x86_64.rpm
```

```
# clush -w cephosd[11-12] -b “rpm -ivh /tmp/kmod-enic-2.3.0.31-
rhel7u3.el7.x86_64.rpm”
```

```
# clush -w cephosd[11-12] -b “modinfo enic | head -5”
```

Create VLAN Interface for Network Public on Both Ceph OSD Nodes

```
# ssh cephosd11 “nmcli con add type vlan con-name eth1 dev eth1 id 10 ip4
192.168.10.130/24”
```

```
# ssh cephosd11 “nmcli con add type vlan con-name eth2 dev eth2 id 20 ip4
192.168.20.130/24”
```

```
# ssh cephosd12 “nmcli con add type vlan con-name eth1 dev eth1 id 10 ip4
192.168.10.131/24”
```

```
# ssh cephosd12 “nmcli con add type vlan con-name eth2 dev eth2 id 20 ip4
192.168.20.131/24”
```

Update All Ceph OSD Nodes

```
# clush -w cephosd[11-12] -b yum -y update
```

Configure the Firewall

```
# clush –a –b “systemctl enable firewalld”

# clush –a –b “systemctl start firewalld”

# clush –a –b “systemctl status firewalld”

# clush –w cephosd[11-12] –b “firewall-cmd --zone=public --add-port=6800-
7300/tcp”

# clush –w cephosd[11-12] –b “firewall-cmd --zone=public --add-port=6800-
7300/tcp –permanent”

# clush –w cephosd[11-12] –b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.10.0/24"\” port
protocol=\”"tcp"\” port=\”"6800-7300"\” accept"\””

# clush –w cephosd[11-12] –b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.10.0/24"\” port
protocol=\”"tcp"\” port=\”"6800-7300"\” accept"\” --permanent”

# clush –w cephosd[11-12] –b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.20.0/24"\” port
protocol=\”"tcp"\” port=\”"6800-7300"\” accept"\””

# clush –w cephosd[11-12] –b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.20.0/24"\” port
protocol=\”"tcp"\” port=\”"6800-7300"\” accept"\” --permanent”
```

Configure Network Time Protocol

```
# clush –w cephosd[11-12] –b “yum –y install ntp”

# clush –w cephosd[11-12] –b –c /root/ntp.conf --dest=/etc

# clush –w cephosd[11-12] –b “service ntpd stop”

# clush –w cephosd[11-12] –b “ntpdate cephadm”

# clush –w cephosd[11-12] –b “service ntpd start”

# clush –w cephosd[11-12] –b “systemctl enable ntpd”
```

Copy sudoers File to Ceph RGW Nodes and Enable Password-Less SSH

```
# clush –w cephosd[11-12] –b –c /etc/sudoers.d/cephadm

# su – cephadm

# ssh-copy-id cephadm@cephosd11
```

Repeat the last step for cephosd12.

Change .ssh/config File for cephadm User

```
# vi .ssh/config

Host node14
```

```
        Hostname cephosd11
        User    cephadm
Host node15
        Hostname cephosd12
        User    cephadm
```

Check Disk Devices and Prepare Disks for Ceph

1. Reboot both OSD nodes to make sure that the disks appear as the same devices.

```
# clush -w cephosd[11-12] -b reboot
```

2. Check on cephosd11,12 the disk device names.

```
# clush -w cephosd[11-12] -b “lsblk”
NAME                   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                      8:0    0 372.6G  0 disk
sdb                      8:16   0 372.6G  0 disk
sdc                      8:32   0 372.6G  0 disk
sdd                      8:48   0 372.6G  0 disk
sde                      8:64   0 110.8G  0 disk
├─sde1                   8:65   0   500M  0 part /boot
└─sde2                   8:66   0 110.3G  0 part
  ├─rhel_cephosd1-root 253:0    0    50G  0 lvm  /
  ├─rhel_cephosd1-swap 253:1    0     4G  0 lvm
  └─rhel_cephosd1-home 253:2    0  56.3G  0 lvm  /home
sdf                      8:80   0   5.5T  0 disk
sdg                      8:96   0   5.5T  0 disk
sdh                      8:112  0   5.5T  0 disk
sdi                      8:128  0   5.5T  0 disk
sdj                      8:144  0   5.5T  0 disk
sdk                      8:160  0   5.5T  0 disk
sdl                      8:176  0   5.5T  0 disk
sdm                      8:192  0   5.5T  0 disk
sdn                      8:208  0   5.5T  0 disk
sdo                      8:224  0   5.5T  0 disk
```

```
sdp                         8:240  0   5.5T  0 disk
sdq                        65:0    0   5.5T  0 disk
sdr                        65:16   0   5.5T  0 disk
sds                        65:32   0   5.5T  0 disk
sdt                        65:48   0   5.5T  0 disk
sdu                        65:64   0   5.5T  0 disk
sdv                        65:80   0   5.5T  0 disk
sdw                        65:96   0   5.5T  0 disk
sdx                        65:112  0   5.5T  0 disk
sdy                        65:128  0   5.5T  0 disk
sdz                        65:144  0   5.5T  0 disk
sdaa                       65:160  0   5.5T  0 disk
sdab                       65:176  0   5.5T  0 disk
sdac                       65:192  0   5.5T  0 disk
# clush -w cephosd[11-12] -b "yum -y install gdisk"
```

3. Start zapping SSDs.

```
# clush -w cephosd[11-12] -b "for i in {a..d}; do sgdisk -Z -g -o
/dev/sd\${i}; done"
```

4. Start zapping HDDs.

```
# clush -w cephosd[11-12] -b "for i in {{f..z},aa,ab,ac}; do sgdisk -Z -g -o
/dev/sd\${i}; done"
```

Change Ansible Hosts File

```
# vi /etc/ansible/hosts

[mons]

cephmon[1:3]


[osds]

cephosd[1:12]
```

1. Verify that all hosts are available.

```
# ansible all -m ping
```

Deploy Red Hat Ceph Storage via Ansible

To deploy Red Hat Ceph Storage via Ansible, complete the following step:

1. Deploy both OSD nodes via Ansible as follows:

```
# cd /usr/share/ceph-ansible

# ansible-playbook ceph.yml
```

After a successful deployment, your result should look like the following:

```
PLAY RECAP
********************************************************************

cephmon1                   : ok=91   changed=2    unreachable=0    failed=0

cephmon2                   : ok=91   changed=2    unreachable=0    failed=0

cephmon3                   : ok=91   changed=2    unreachable=0    failed=0

cephosd1                   : ok=164  changed=2    unreachable=0    failed=0

cephosd10                  : ok=164  changed=2    unreachable=0    failed=0

cephosd11                  : ok=164  changed=16   unreachable=0    failed=0

cephosd12                  : ok=164  changed=16   unreachable=0    failed=0

cephosd2                   : ok=164  changed=2    unreachable=0    failed=0

cephosd3                   : ok=164  changed=2    unreachable=0    failed=0

cephosd4                   : ok=164  changed=2    unreachable=0    failed=0

cephosd5                   : ok=164  changed=2    unreachable=0    failed=0

cephosd6                   : ok=164  changed=2    unreachable=0    failed=0

cephosd7                   : ok=164  changed=2    unreachable=0    failed=0

cephosd8                   : ok=164  changed=2    unreachable=0    failed=0

cephosd9                   : ok=164  changed=2    unreachable=0    failed=0
```

Final Check of Ceph Deployment

To verify the correct deployment of the Ceph Cluster, complete the following step:

```
[root@cephadm ceph-ansible]# ceph -s

    cluster 0e947555-5166-4ecb-9251-3179a6097ce9

     health HEALTH_OK

     monmap e1: 3 mons at
{cephmon1=192.168.10.111:6789/0,cephmon2=192.168.10.112:6789/0,cephmon3=192.1
68.10.113:6789/0}

            election epoch 8, quorum 0,1,2 cephmon1,cephmon2,cephmon3

     osdmap e767: 288 osds: 288 up, 288 in

            flags sortbitwise

      pgmap v34614: 4160 pgs, 2 pools, 7940 GB data, 7940 kobjects
```

```
               23914 GB used, 1547 TB / 1571 TB avail
                   4160 active+clean
```

There should be 288 OSDs for 288 physical disks installed and one default pool with 64 Placement Groups. You have added two additional OSD nodes to your Red Hat Ceph Storage Cluster on Cisco UCS.

Add RADOS Gateway for Object Storage

Ceph Object Gateway node runs the Ceph RADOS Gateway daemon (ceph-radosgw), and is an object storage interface built on top of librados to provide applications with a RESTful gateway to Ceph Storage Clusters. The Ceph RADOS Gateway supports two interfaces:

- **S3 –** Provides object storage functionality with an interface that is compatible with a large subset of the Amazon S3 RESTful API.
- **Swift –** Provides object storage functionality with an interface that is compatible with a large subset of the OpenStack Swift API.

After building the initial Ceph cluster with three Ceph Monitor Nodes and 12 Ceph OSD Nodes, the following steps describe the procedure to add three RADOS Gateway nodes with Cisco UCS Manager and Ceph Ansible to enable Object Storage.

Enable Fabric Interconnect Ports for Server

To enable server ports for all Ceph RGW nodes after connecting them to both Fabric Interconnects, complete the following steps:

1. Select the `Equipment` tab on the left site.
2. Select Equipment > Fabric Interconnects > Fabric Interconnect A (subordinate) > Fixed Module.
3. Click `Ethernet Ports` section.
4. Select Ports 17-19, right-click and then select `Configure as Server Port`.
5. Click `Yes` and then `OK`.
6. Repeat the same steps for Fabric Interconnect B.

Label Each Server for Identification

For a better identification, label each server by completing the following steps:

1. Select the `Equipment` tab on the left site.

   Select `Rack-Mounts > Servers > Server 5`.
2. In the `Properties` section on the right go to `User Label` and add `Ceph RGW 1` to the field.
3. Repeat the previous steps for `Server 6` and `Server 7` according to Table 14 .

Table 14 Server Label

Server	Name

Rack-Mount / Server 5	Ceph RGW 1
Rack-Mount / Server 6	Ceph RGW 2
Rack-Mount / Server 7	Ceph RGW 3

Add Ceph RGW Nodes to Server Pool

To add all three Ceph RGW nodes to the specific Server Pool `C220M4S`, complete the following steps:

1. Select the `Servers` Tab in the left pane in the Cisco UCS Manager GUI.
2. Go to Servers > Pools > root > Server Pools and choose the pool C220M4S.
3. Click `Add Servers` in the right window, choose Server 5-7 and click `>>` to add them to the pool.
4. Click `OK` and then `OK`.

Setting Disks for Rack-Mount Servers to Unconfigured-Good

To prepare all disks from the Rack-Mount servers for storage profiles, the disks have to be converted from JBOD to Unconfigured-Good. To convert the disks, complete the following steps:

1. Select the `Equipment` tab in the left pane of the Cisco UCS Manager GUI.
2. Go to Equipment > Rack-Mounts > Servers > Server 5 > Disks.
3. Select both disks and right-click `Set JBOD to Unconfigured-Good`.
4. Repeat the steps for Server 6 and 7.

Create Service Profiles from Template

To create the additional profiles for the Ceph RGW nodes, complete the following steps:

1. Select `Servers` from the left pane of the Cisco UCS Manager GUI.
2. Go to `Servers > Service Profiles` and right-click Create Service Profiles from Template.
3. Type in UCS-RGW-C220M4S- in the Name Prefix field.
4. Leave Name Suffix Starting Number as 1.
5. Type in 3 for the `Number of Instances`.
6. Choose `UCS-C220M4S` as the `Service Profile Template` you created before.
7. Click `OK` and then `OK`.

Install RHEL 7.3 on Cisco UCS C220 M4S

To install Red Hat Linux 7.3 operating system on all three Ceph RGW, complete the following steps:

1. Log in to the Cisco UCS Manager and select the `Equipment` tab from the left pane.

2. Go to Equipment > Rack-Mounts > Server > Server 5 (Ceph RGW 1) and right-click KVM Console.

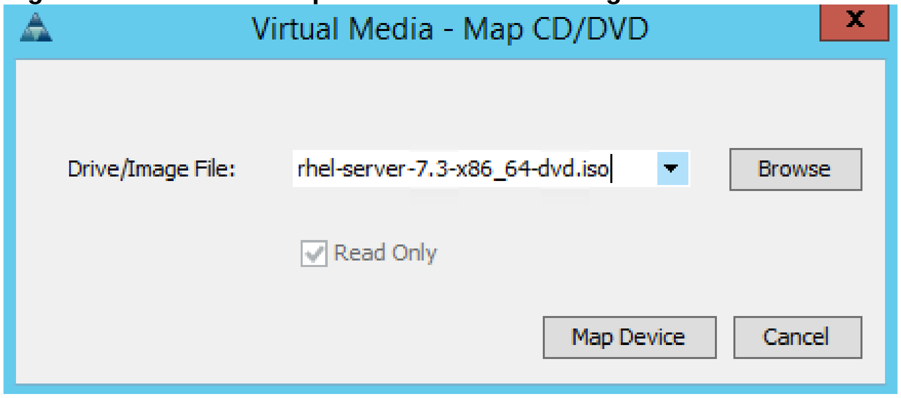
3. Launch KVM Console.

4. Click the `Activate Virtual Devices` in the Virtual Media tab.

5. In the KVM window, select the Virtual Media tab and click `Map CD/DVD`.

6. Browse to the Red Hat Enterprise Linux 7.3 installation ISO image and select then `Map Device`.

Figure 85 Red Hat Enterprise Linux 7.3 ISO image



Drive/Image File: rhel-server-7.3-x86_64-dvd.iso

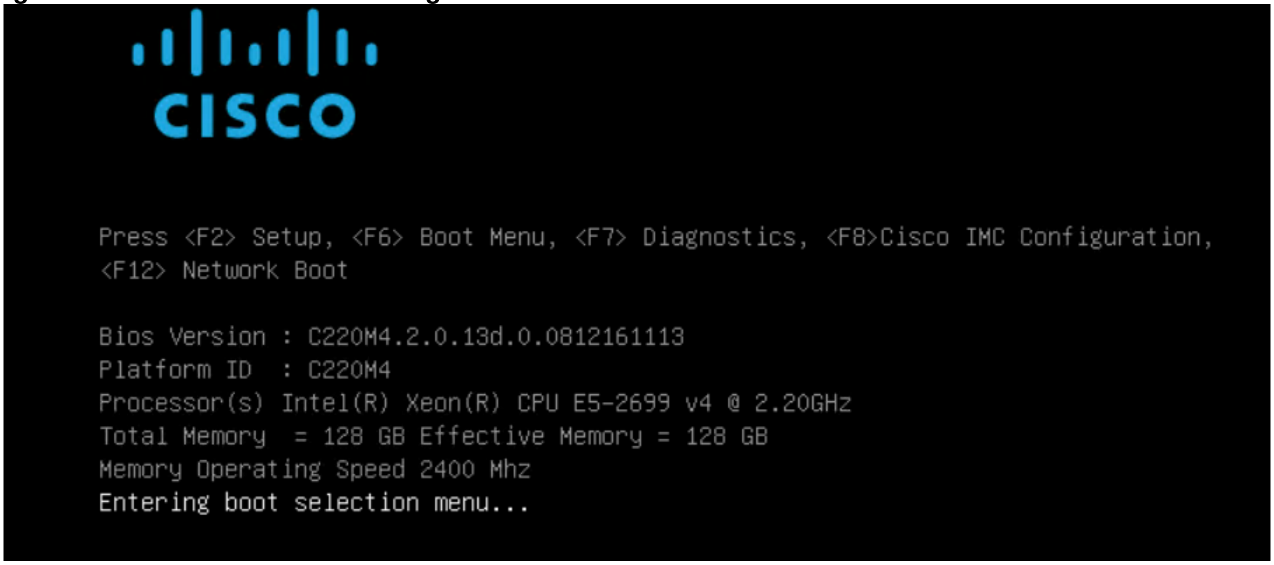
☑ Read Only

7. In the KVM window, select the `Macros > Static Macros > Ctrl-Alt-Del` button in the upper left corner.

8. Click `OK` and then `OK` to reboot the system.

9. In the boot screen with the Cisco Logo, press `F6` for the boot menu.

Figure 86 Boot screen for selecting Boot Menu

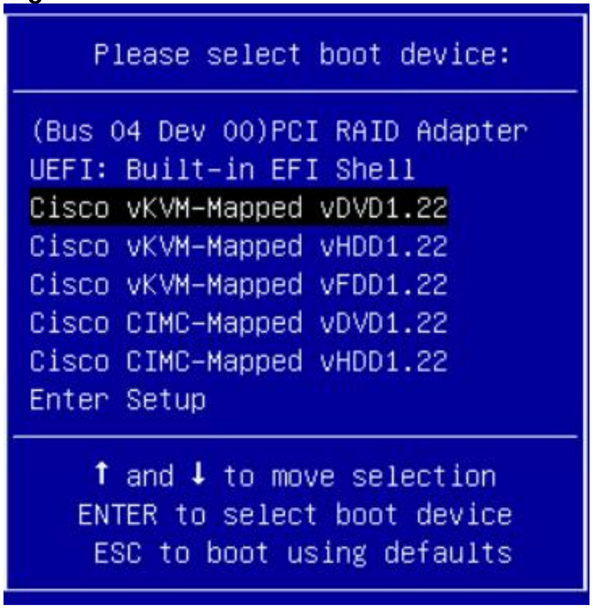


```
Press <F2> Setup, <F6> Boot Menu, <F7> Diagnostics, <F8>Cisco IMC Configuration,
<F12> Network Boot

Bios Version : C220M4.2.0.13d.0.0812161113
Platform ID  : C220M4
Processor(s) Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
Total Memory  = 128 GB Effective Memory = 128 GB
Memory Operating Speed 2400 Mhz
Entering boot selection menu...
```

10. When the Boot Menu appears, select `Cisco vKVM-Mapped vDVD1.22`.

Figure 87 Boot Menu Selection



```
 Please select boot device:

(Bus 04 Dev 00)PCI RAID Adapter
UEFI: Built-in EFI Shell
Cisco vKVM-Mapped vDVD1.22
Cisco vKVM-Mapped vHDD1.22
Cisco vKVM-Mapped vFDD1.22
Cisco CIMC-Mapped vDVD1.22
Cisco CIMC-Mapped vHDD1.22
Enter Setup

 ↑ and ↓ to move selection
 ENTER to select boot device
 ESC to boot using defaults
```

11. When the Red Hat Enterprise Linux 7.3 installer appears, press the Tab button for further configuration options.

Note: Use the previous created Kickstart file for Cisco UCS C220 M4S.

12. At the prompt type.

```
inst.ks=http://192.168.0.100/ceph-ks.cfg net.ifnames=0 biosdevname=0
ip=192.168.0.115::192.168.0.99:255.255.255.0:cephrgw1.cisco.com:eth0:none
nameserver=173.36.131.10
```

13. Repeat the previous steps for Ceph RGW 2, Ceph RGW 3 with the IP address shown in Table 13 .

Configure /etc/hosts and Enable Password-Less Login

1. Modify the /etc/hosts file on cephadm according to Table 15 and include all IP address of all nodes. An example is shown in Appendix C.

Table 15 IP Addresses for Ceph RGW Nodes

	Default	Public
cephrgw1	192.168.0.115	192.168.10.115
cephrgw2	192.168.0.116	192.168.10.116
cephrgw3	192.168.0.117	192.168.10.117

2. Login to cephadm and change `/etc/hosts`.

```
# ssh root@cephadm

# vi /etc/hosts
```

3. Copy `id_rsa.pub` under `/root/.ssh` to all other Ceph nodes.

```
# for i in {1..3}; do ssh-copy-id cephrgw${i}; done
```

4. Change Cluster Shell configuration file and add all Ceph RGW nodes.

```
# vi /etc/clustershell/groups.d/local.cfg

all: cephmon[1-3] cephosd[1-12] cephrgw[1-3]
```

5. Copy /etc/hosts to all Ceph nodes

```
# clush –a –b –c /etc/hosts
```

Configure Hostnames

To configure the hostnames, complete the following step:

1. Configure hostname for cephrgw1-3

```
# for i in {1..3}; do ssh cephrgw${i} "hostnamectl set-hostname cephrgw${i}";
done
```

Copy Local Repository File and Enable Repositories

To copy the local repository file and enable the repositories, complete the following step:

1. Copy the repository file to all three Ceph RGW nodes and enable repositories

```
# clush –w cephrgw[1-3] –b –c /etc/yum.repos.d/ceph.repo

# clush –w cephrgw[1-3] -b yum clean all

# clush –w cephrgw[1-3] -b yum repolist
```

Install Latest Network Driver

To install the latest network driver, complete the following step:

1. Copy the network driver from cephadm to cephrgw1-3 and install it.

```
# clush -w cephrgw[1-3] -b -c /tmp/kmod-enic-2.3.0.31-rhel7u3.el7.x86_64.rpm

# clush -w cephrgw[1-3] -b “rpm -ivh /tmp/kmod-enic-2.3.0.31-
rhel7u3.el7.x86_64.rpm”

# clush -w cephrgw[1-3] -b “modinfo enic | head -5”
```

Create VLAN Interface for Network Public on All Ceph RGW Nodes

```
# ssh cephrgw1 “nmcli con add type vlan con-name eth1 dev eth1 id 10 ip4
192.168.10.115/24”

# ssh cephrgw2 “nmcli con add type vlan con-name eth1 dev eth1 id 10 ip4
192.168.10.116/24”

# ssh cephrgw3 “nmcli con add type vlan con-name eth1 dev eth1 id 10 ip4
192.168.10.117/24”
```

Update All Ceph RGW Nodes

```
# clush -w cephrgw[1-3] -b yum -y update
```

Configure the Firewall

```
# clush -a -b “systemctl enable firewalld”

# clush -a -b “systemctl start firewalld”

# clush -a -b “systemctl status firewalld”

# clush -w cephrgw[1-3] -b "firewall-cmd --zone=public --add-port=7480/tcp"

# clush -w cephrgw[1-3] -b "firewall-cmd --zone=public --add-port=7480/tcp –
permanent”

# clush -w cephrgw[1-3] -b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.10.0/24"\” port
protocol=\”"tcp"\” port=\”"7480"\” accept"\””

# clush -w cephrgw[1-3] -b “firewall-cmd --zone=public --add-rich-
rule=\”"rule family=\”"ipv4"\” source address=\”"192.168.10.0/24"\” port
protocol=\”"tcp"\” port=\”"7480"\” accept"\” --permanent”
```

Configure Network Time Protocol

```
# clush -w cephrgw[1-3] -b “yum -y install ntp”

# clush -w cephrgw[1-3] -b -c /root/ntp.conf --dest=/etc

# clush -w cephrgw[1-3] -b “service ntpd stop”

# clush -w cephrgw[1-3] -b “ntpdate cephadm”

# clush -w cephrgw[1-3] -b “service ntpd start”
# clush -w cephrgw[1-3] -b "systemctl enable ntpd"
```

Copy sudoers File to Ceph RGW Nodes and Enable Password-Less SSH

```
# clush -w cephrgw[1-3] -b -c /etc/sudoers.d/cephadm

# su - cephadm

# ssh-copy-id cephadm@cephrgw1
```

Repeat the last step for cephrgw2 and cephrgw3.

Change .ssh/config File for cephadm User

```
# vi .ssh/config

Host node16

        Hostname cephrgw1

        User    cephadm

Host node17

        Hostname cephrgw2

        User    cephadm

Host node18

        Hostname cephrgw3

        User    cephadm
```

Change Ansible Hosts File

```
# vi /etc/ansible/hosts

[mons]

cephmon[1:3]


[osds]

cephosd[1:12]


[rgws]

cephrgw1 radosgw_civetweb_bind_ip=192.168.10.115

cephrgw2 radosgw_civetweb_bind_ip=192.168.10.116

cephrgw3 radosgw_civetweb_bind_ip=192.168.10.117
```

Verify if all hosts are available

```
# ansible all -m ping
```

Deploy Red Hat Ceph Storage via Ansible

To deploy Red Hat Ceph Storage via Ansible, complete the following step:

1. Deploy both OSD nodes via Ansible as follows:

```
# cd /usr/share/ceph-ansible

# ansible-playbook ceph.yml
```

After a successful deployment, your result should look like the following:

```
PLAY RECAP
*********************************************************************

cephmon1                   : ok=91   changed=2    unreachable=0    failed=0

cephmon2                   : ok=91   changed=2    unreachable=0    failed=0

cephmon3                   : ok=91   changed=2    unreachable=0    failed=0

cephosd1                   : ok=164  changed=2    unreachable=0    failed=0

cephosd10                  : ok=164  changed=2    unreachable=0    failed=0

cephosd11                  : ok=164  changed=2    unreachable=0    failed=0

cephosd12                  : ok=164  changed=2    unreachable=0    failed=0

cephosd2                   : ok=164  changed=2    unreachable=0    failed=0

cephosd3                   : ok=164  changed=2    unreachable=0    failed=0

cephosd4                   : ok=164  changed=2    unreachable=0    failed=0

cephosd5                   : ok=164  changed=2    unreachable=0    failed=0

cephosd6                   : ok=164  changed=2    unreachable=0    failed=0

cephosd7                   : ok=164  changed=2    unreachable=0    failed=0

cephosd8                   : ok=164  changed=2    unreachable=0    failed=0

cephosd9                   : ok=164  changed=2    unreachable=0    failed=0

cephrgw1                   : ok=81   changed=18   unreachable=0    failed=0

cephrgw2                   : ok=81   changed=18   unreachable=0    failed=0

cephrgw3                   : ok=81   changed=18   unreachable=0    failed=0
```

Final Check of Ceph Deployment

To verify the correct deployment of the Ceph RADOS Gateway, login to cephrgw1 and check if the radosgw process is running.

```
[root@cephrgw1 ~]# systemctl status ceph-radosgw@rgw.cephrgw1.service

● ceph-radosgw@rgw.cephrgw1.service - Ceph rados gateway

   Loaded: loaded (/usr/lib/systemd/system/ceph-radosgw@.service; enabled;
vendor preset: disabled)
```

```
   Active: active (running) since Thu 2017-01-05 06:20:28 PST; 3 days ago
 Main PID: 15797 (radosgw)
   CGroup: /system.slice/system-ceph\x2dradosgw.slice/ceph-
radosgw@rgw.cephrgw1.service
           └─15797 /usr/bin/radosgw -f --cluster ceph --name
client.rgw.cephrgw1 --setuser ceph --setgroup ceph

Jan 05 06:20:28 cephrgw1 systemd[1]: Started Ceph rados gateway.
Jan 05 06:20:28 cephrgw1 systemd[1]: Starting Ceph rados gateway...
Jan 05 08:04:21 cephrgw1 systemd[1]: [/usr/lib/systemd/system/ceph-
radosgw@.service:17] Unknown lvalue 'TasksMa...rvice'
Hint: Some lines were ellipsized, use -l to show in full.
[root@cephrgw1 ~]#
```

Repeat the same step for cephrgw2 and cephrgw3. Replace the service name with the appropriate hostname for cephrgw2 and cephrgw3.

Bill of Materials

This section provides the BOM for the entire Red Hat Ceph Storage and Cisco UCS solution.

Table 16 Bill of Materials for Cisco Nexus 9332PQ

Item Name	Description	Quantity
N9K-C9332PQ	Nexus 9300 Series, 32p 40G QSFP+	2
CON-PSRT-9332PQ	PRTNR SS 8X5XNBD Nexus 9332 ACI Leaf switch with 32p 40G	2
NXOS-703I5.1	Nexus 9500, 9300, 3000 Base NX-OS Software Rel 7.0(3)I5(1)	2
N3K-C3064-ACC-KIT	Nexus 3K/9K Fixed Accessory Kit	2
QSFP-H40G-CU1M	40GBASE-CR4 Passive Copper Cable, 1m	10
NXA-FAN-30CFM-B	Nexus 2K/3K/9K Single Fan, port side intake airflow	8
CAB-C13-CBN	Cabinet Jumper Power Cord, 250 VAC 10A, C14-C13 Connectors	4
N9K-PAC-650W	Nexus 9300 650W AC PS, Port-side Intake	4

Table 17 Bill of Materials for Cisco UCS Fabric Interconnect 6332

Item Name	Description	Quantity
UCS-SP-FI6332-2X	UCS SP Select 6332 FI /No PSU/32 QSFP+	1
UCS-SP-FI6332	(Not sold standalone) UCS 6332 1RU FI/No PSU/32 QSFP+	2
UCS-PSU-6332-AC	UCS 6332 Power Supply/100-240VAC	4
CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	4
QSFP-H40G-CU3M	40GBASE-CR4 Passive Copper Cable, 3m	38
QSFP-40G-SR-BD	QSFP40G BiDi Short-reach Transceiver	8
N10-MGT014	UCS Manager v3.1	2
UCS-FAN-6332	UCS 6332 Fan Module	8
UCS-ACC-6332	UCS 6332 Chassis Accessory Kit	2
RACK-UCS2	Cisco R42610 standard rack, w/side panels	1
RP230-32-1P-U-2	Cisco RP230-32-U-2 Single Phase PDU 20x C13, 4x C19	2

Table 18 Bill of Materials for Cisco UCS S3260 Storage Server

Item Name	Description	Quantity
UCSS-S3260	Cisco UCS S3260 Storage Server Base Chassis	6
UCSC-C3X60-HD6TB	UCS C3X60 6TB 12Gbps NL-SAS 7200RPM HDD w carrier-Top-load	36
UCS-C3X60-12G240	UCSC C3X60 400GB 12Gbps SSD (Gen 2)	48
UCSC-PSU1-1050W	UCS C3X60 1050W Power Supply Unit	24
CAB-C13-CBN	Cabinet Jumper Power Cord, 250 VAC 10A, C14-C13 Connectors	24
UCSC-C3X60-RAIL	UCS C3X60 Rack Rails Kit	6
N20-BBLKD-7MM	UCS 7MM SSD Blank Filler	12
UCSS-S3260-BBEZEL	Cisco UCS S3260 Bezel	6
UCSC-C3K-M4SRB	UCS C3000 M4 Server Node for Intel E5-2600 v4	6
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	12
UCS-MR-1X161RV-A	16GB DDR4-2400-MHz RDIMM/PC4-19200/single rank/x4/1.2v	48

Item Name	Description	Quantity
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	6
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	12
UCSC-C3K-M4SRB	UCS C3000 M4 Server Node for Intel E5-2600 v4	6
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	12
UCS-MR-1X161RV-A	16GB DDR4-2400-MHz RDIMM/PC4-19200/single rank/x4/1.2v	48
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	6
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	12
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	6
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	6
UCSC-C3X60-42HD6	Cisco UCS C3X60 Three row of drives containing 42 x 6TB (Tot	6
UCSC-C3X60-HD6TB	UCS C3X60 6TB 12Gbps NL-SAS 7200RPM HDD w carrier- Top-load	252
UCS-C3X60-G2SD12	UCSC C3X60 120GB Boot SSD (Gen 2)	12
RHEL-2S2V-1A	Red Hat Enterprise Linux (1-2 CPU,1-2 VN); 1-Yr Support Req	6

Table 19 Bill of Material for Cisco UCS C220 M4S

Item Name	Description	Quantity
UCSC-C220-M4S	UCS C220 M4 SFF w/o CPU, mem, HD, PCIe, PSU, rail kit	7
UCS-CPU-E52699E	2.20 GHz E5-2699 v4/145W 22C/55MB Cache/DDR4 2400MHz	14
UCS-MR-1X161RV-A	16GB DDR4-2400-MHz RDIMM/PC4-19200/single rank/x4/1.2v	56
UCS-HD600G10K12G	600GB 12G SAS 10K RPM SFF HDD	14
UCSC-PCIE-C40Q-03	Cisco VIC 1385 Dual Port 40Gb QSFP+ CNA w/RDMA	7
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 M4 and C240 M4 rack servers	7
UCSC-PSU1-770W	770W AC Hot-Plug Power Supply for 1U C-Series Rack Server	14
CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	14
UCS-M4-V4-LBL	Cisco M4 - v4 CPU asset tab ID label (Auto-Expand)	7
N20-BBLKD	UCS 2.5 inch HDD blanking panel	42
UCSC-SCCBL220	Supercap cable 950mm	7
UCSC-MLOM-BLK	MLOM Blanking Panel	7
UCSC-HS-C220M4	Heat sink for UCS C220 M4 rack servers	14
UCSC-MRAID12G	Cisco 12G SAS Modular Raid Controller	7
UCSC-MRAID12G-1GB	Cisco 12Gbps SAS 1GB FBWC Cache module (Raid 0/1/5/6)	7
RHEL-2S2V-1A	Red Hat Enterprise Linux (1-2 CPU,1-2 VN); 1-Yr Support Req	7

Appendix

Appendix A – Kickstart File for Cisco UCS C220 M4S

```
lang en_US
keyboard --vckeymap=us --xlayouts='us'
timezone America/Los_Angeles --isUtc --ntpservers=10.29.137.1
rootpw $1$AzLo5Nru$YuZng8sCZSToN2FOiPYtk. --iscrypted
user --groups=wheel --name=cephadm --
password=$6$p0smwIo9EEQOhrC.$7Ho.dWuG6iRJY0fKcujsC92WZXXwDSZPGp/aA.UujDSmc5J5.vnd
nyIfO9U7annoUTcfg0tXUCGVUwCqNGINI. --iscrypted
#platform x86, AMD64, or Intel EM64T
reboot
cdrom
bootloader --location=mbr --append="rhgb quiet crashkernel=auto" --boot-drive=sda
zerombr
clearpart --all --initlabel --drives=sda
autopart
auth --passalgo=sha512 --useshadow
selinux --disabled
firewall --disabled
firstboot --disable
services --enabled="chronyd"
%packages
@base
chrony
kexec-tools
%end
%addon com_redhat_kdump --enable --reserve-mb='auto'

%end
```

Appendix B – Kickstart File for Cisco UCS S3260 Storage Server

```
lang en_US
keyboard --vckeymap=us --xlayouts='us'
timezone America/Los_Angeles --isUtc --ntpservers=10.29.137.1
rootpw $1$AzLo5Nru$YuZng8sCZSToN2FOiPYtk. --iscrypted
user --groups=wheel --name=cephadm --
password=$6$p0smwIo9EEQOhrC.$7Ho.dWuG6iRJY0fKcujsC92WZXXwDSZPGp/aA.UujDSmc5J5.vnd
nyIfO9U7annoUTcfg0tXUCGVUwCqNGINI. --iscrypted
#platform x86, AMD64, or Intel EM64T
reboot
cdrom
```

```
bootloader --location=mbr --append="rhgb quiet crashkernel=auto" --boot-drive=sde
zerombr
clearpart --all --initlabel --drives=sde
autopart
auth --passalgo=sha512 --useshadow
selinux --disabled
firewall --disabled
firstboot --disable
services --enabled="chronyd"
%packages
@base
chrony
kexec-tools
%end
%addon com_redhat_kdump --enable --reserve-mb='auto'

%end
```

Appendix C – Example /etc/hosts File

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
127.0.0.1    localhost
::1          localhost

# External/PXE Network
```

```
192.168.0.100   jumphost

192.168.0.110   cephadm
192.168.0.111   cephmon1
192.168.0.112   cephmon2
192.168.0.113   cephmon3

192.168.0.115   cephrgw1
192.168.0.116   cephrgw2
192.168.0.117   cephrgw3

192.168.0.120   cephosd1
192.168.0.121   cephosd2
192.168.0.122   cephosd3
192.168.0.123   cephosd4
192.168.0.124   cephosd5
192.168.0.125   cephosd6
192.168.0.126   cephosd7
192.168.0.127   cephosd8
192.168.0.128   cephosd9
192.168.0.129   cephosd10
192.168.0.130   cephosd11
192.168.0.131   cephosd12

# Public Network
192.168.10.110 cephadm-public
192.168.10.111 cephmon1-public
192.168.10.112 cephmon2-public
192.168.10.113 cephmon3-public
192.168.10.115 cephrgw1-public
192.168.10.116 cephrgw2-public
192.168.10.117 cephrgw3-public
192.168.10.120 cephosd1-public
192.168.10.121 cephosd2-public
192.168.10.122 cephosd3-public
192.168.10.123 cephosd4-public
192.168.10.124 cephosd5-public
192.168.10.125 cephosd6-public
192.168.10.126 cephosd7-public
192.168.10.127 cephosd8-public
192.168.10.128 cephosd9-public
192.168.10.129 cephosd10-public
192.168.10.130 cephosd11-public
192.168.10.131 cephosd12-public
```

```
# Storage Network
192.168.20.120 cephosd1-storage
192.168.20.121 cephosd2-storage
192.168.20.122 cephosd3-storage
192.168.20.123 cephosd4-storage
192.168.20.124 cephosd5-storage
192.168.20.125 cephosd6-storage
192.168.20.126 cephosd7-storage
192.168.20.127 cephosd8-storage
192.168.20.128 cephosd9-storage
192.168.20.129 cephosd10-storage
192.168.20.130 cephosd11-storage
192.168.20.131 cephosd12-storage
```

Appendix D – /home/cephadm/.ssh/config File from Ansible Administration Node cephadm

```
Host node1
        Hostname cephmon1
        User    cephadm
Host node2
        Hostname cephmon2
        User    cephadm
Host node3
        Hostname cephmon3
        User    cephadm
Host node4
        Hostname cephosd1
        User    cephadm
Host node5
        Hostname cephosd2
        User    cephadm
Host node6
        Hostname cephosd3
        User    cephadm
Host node7
        Hostname cephosd4
        User    cephadm
Host node8
        Hostname cephosd5
        User    cephadm
Host node9
        Hostname cephosd6
        User    cephadm
Host node10
        Hostname cephosd7
        User    cephadm
Host node11
        Hostname cephosd8
        User    cephadm
Host node12
        Hostname cephosd9
        User    cephadm
Host node13
        Hostname cephosd10
        User    cephadm
Host node14
        Hostname cephosd11
        User    cephadm
Host node15
        Hostname cephosd12
        User    cephadm
Host node16
        Hostname cephrgw1
        User    cephadm
Host node17
        Hostname cephrgw2
        User    cephadm
Host node18
        Hostname cephrgw3
        User    cephadm
```

Appendix E - /usr/share/ceph-ansible/group_vars/all Configuration File

```
---
# Variables here are applicable to all host groups NOT roles

# This sample file generated by generate_group_vars_sample.sh

# Dummy variable to avoid error because ansible does not recognize the
# file as a good configuration file when no variable in it.
dummy:

# You can override vars by using host or group vars
```

```
###########
# GENERAL #
###########

fetch_directory: ~/ceph-ansible-keys
cluster: ceph # cluster name

###########
# INSTALL #
###########

#mon_group_name: mons
#osd_group_name: osds
#rgw_group_name: rgws
#mds_group_name: mdss
#restapi_group_name: restapis
#rbdmirror_group_name: rbdmirrors
#client_group_name: clients

# If check_firewall is true, then ansible will try to determine if the
# Ceph ports are blocked by a firewall. If the machine running ansible
# cannot reach the Ceph ports for some other reason, you may need or
# want to set this to False to skip those checks.
#check_firewall: True

# This variable determines if ceph packages can be updated.  If False, the
# package resources will use "state=present".  If True, they will use
# "state=latest".
#upgrade_ceph_packages: False
```

```
# If this is false then the 'ceph' package will be installed on rpm systems, which
# is not needed for versions after infernalis.
#use_server_package_split: true

# /!\ EITHER ACTIVE ceph_stable OR ceph_dev /!\

#debian_package_dependencies:
#  - python-pycurl
#  - hdparm
#  - ntp

#centos_package_dependencies:
#  - python-pycurl
#  - hdparm
#  - yum-plugin-priorities.noarch
#  - epel-release
#  - ntp
#  - python-setuptools
#  - libselinux-python

#redhat_package_dependencies:
#  - python-pycurl
#  - hdparm
#  - ntp
#  - python-setuptools

# Whether or not to install the ceph-test package.
#ceph_test: False

## Configure package origin
#
```

```
#ceph_origin: 'distro'
# 'distro' means that no separate repo file will be added
# you will get whatever version of Ceph is included in your Linux distro.
#
#ceph_use_distro_backports: false # DEBIAN ONLY


# STABLE
########

# COMMUNITY VERSION
#ceph_stable: false # use ceph stable branch
#ceph_stable_key: https://download.ceph.com/keys/release.asc
#ceph_stable_release: infernalis # ceph stable release
#ceph_stable_repo: "http://ceph.com/debian-{{ ceph_stable_release }}"

###################
# Stable Releases #
###################
#ceph_stable_releases:
#  - dumpling
#  - emperor
#  - firefly
#  - giant
#  - hammer

# Use the option below to specify your applicable package tree, eg. when
using non-LTS Ubuntu versions
# # for a list of available Debian distributions, visit
http://ceph.com/debian-{{ ceph_stable_release }}/dists/
# for more info read: https://github.com/ceph/ceph-ansible/issues/305
#ceph_stable_distro_source:
```

```
# This option is needed for _both_ stable and dev version, so please always
fill the right version
# # for supported distros, see http://ceph.com/rpm-{{ ceph_stable_release }}/
#ceph_stable_redhat_distro: el7


# ENTERPRISE VERSION RED HAT STORAGE (from 1.3)
# This version is only supported on RHEL 7.1
# As of RHEL 7.1, libceph.ko and rbd.ko are now included in Red Hat's kernel
# packages natively. The RHEL 7.1 kernel packages are more stable and secure
than
# using these 3rd-party kmods with RHEL 7.0. Please update your systems to
RHEL
# 7.1 or later if you want to use the kernel RBD client.
#
# The CephFS kernel client is undergoing rapid development upstream, and we
do
# not recommend running the CephFS kernel module on RHEL 7's 3.10 kernel at
this
# time. Please use ELRepo's latest upstream 4.x kernels if you want to run
CephFS
# on RHEL 7.
#
ceph_stable_rh_storage: true
# This will affect how/what repositories are enabled depending on the desired
# version. The previous  version was "1.3".
ceph_stable_rh_storage_version: 2
#ceph_stable_rh_storage_cdn_install: false # assumes all the nodes can
connect to cdn.redhat.com
ceph_stable_rh_storage_iso_install: true # usually used when nodes don't have
access to cdn.redhat.com
ceph_stable_rh_storage_iso_path: /tmp/rhceph-2.1-rhel-7-x86_64.iso
ceph_stable_rh_storage_mount_path: /tmp/rh-storage-mount
ceph_stable_rh_storage_repository_path: /tmp/rh-storage-repo # where to copy
iso's content


# DEV
# ###

#ceph_dev: false # use ceph development branch
#ceph_dev_key: https://download.ceph.com/keys/autobuild.asc
#ceph_dev_branch: master # development branch you would like to use e.g:
master, wip-hack

# supported distros are centos6, centos7, fc17, fc18, fc19, fc20, fedora17,
fedora18,
# fedora19, fedora20, opensuse12, sles0. (see http://gitbuilder.ceph.com/).
# For rhel, please pay attention to the versions: 'rhel6 3' or 'rhel 4', the
fullname is _very_ important.
#ceph_dev_redhat_distro: centos7

# CUSTOM
# ###

# Use a custom repository to install ceph.  For RPM, ceph_custom_repo should
be
# a URL to the .repo file to be installed on the targets.  For deb,
# ceph_custom_repo should be the URL to the repo base.
#ceph_custom: true # use custom ceph repository
#ceph_custom_repo: https://192.168.0.100/rhcs2

######################
# CEPH CONFIGURATION #
######################

## Ceph options
#
# Each cluster requires a unique, consistent filesystem ID. By
# default, the playbook generates one for you and stores it in a file
# in `fetch_directory`. If you want to customize how the fsid is
# generated, you may find it useful to disable fsid generation to
# avoid cluttering up your ansible repo. If you set `generate_fsid` to

# false, you *must* generate `fsid` in another way.

#fsid: "{{ cluster_uuid.stdout }}"

generate_fsid: true


cephx: true

max_open_files: 131072


## Client options

#

#rbd_cache: "true"

#rbd_cache_writethrough_until_flush: "true"

#rbd_concurrent_management_ops: 20


#rbd_client_directories: true # this will create rbd_client_log_path and
rbd_client_admin_socket_path directories with proper permissions


# Permissions for the rbd_client_log_path and

# rbd_client_admin_socket_path. Depending on your use case for Ceph

# you may want to change these values. The default, which is used if

# any of the variables are unset or set to a false value (like `null`

# or `false`) is to automatically determine what is appropriate for

# the Ceph version with non-OpenStack workloads -- ceph:ceph and 0770

# for infernalis releases, and root:root and 1777 for pre-infernalis

# releases.

#

# For other use cases, including running Ceph with OpenStack, you'll

# want to set these differently:

#

# For OpenStack on RHEL, you'll want:

#   rbd_client_directory_owner: "qemu"

#   rbd_client_directory_group: "libvirtd" (or "libvirt", depending on your
version of libvirt)
```

```
#   rbd_client_directory_mode: "0755"
#
# For OpenStack on Ubuntu or Debian, set:
#    rbd_client_directory_owner: "libvirt-qemu"
#    rbd_client_directory_group: "kvm"
#    rbd_client_directory_mode: "0755"
#
# If you set rbd_client_directory_mode, you must use a string (e.g.,
# 'rbd_client_directory_mode: "0755"', *not*
# 'rbd_client_directory_mode: 0755', or Ansible will complain: mode
# must be in octal or symbolic form
#rbd_client_directory_owner: null
#rbd_client_directory_group: null
#rbd_client_directory_mode: null

#rbd_client_log_path: /var/log/ceph
#rbd_client_log_file: "{{ rbd_client_log_path }}/qemu-guest-$pid.log" # must
be writable by QEMU and allowed by SELinux or AppArmor
#rbd_client_admin_socket_path: /var/run/ceph # must be writable by QEMU and
allowed by SELinux or AppArmor

## Monitor options
#
# You must define either monitor_interface or monitor_address. Preference
# will go to monitor_interface if both are defined.
monitor_interface: eth1.10
#monitor_address: 0.0.0.0
#mon_use_fqdn: false # if set to true, the MON name used will be the fqdn in
the ceph.conf

## OSD options
#
journal_size: 30000
```

```
public_network: 192.168.10.0/24
cluster_network: 192.168.20.0/24
#osd_mkfs_type: xfs
#osd_mkfs_options_xfs: -f -i size=2048
#osd_mount_options_xfs: noatime,largeio,inode64,swalloc
#osd_objectstore: filestore

# xattrs. by default, 'filestore xattr use omap' is set to 'true' if
# 'osd_mkfs_type' is set to 'ext4'; otherwise it isn't set. This can
# be set to 'true' or 'false' to explicitly override those
# defaults. Leave it 'null' to use the default for your chosen mkfs
# type.
#filestore_xattr_use_omap: null

## MDS options
#
#mds_use_fqdn: false # if set to true, the MDS name used will be the fqdn in
the ceph.conf

## Rados Gateway options
#
#radosgw_dns_name: your.subdomain.tld # subdomains used by radosgw. See
http://ceph.com/docs/master/radosgw/config/#enabling-subdomain-s3-calls
#radosgw_civetweb_port: 8080 # on Infernalis we get: "set_ports_option:
cannot bind to 80: 13 (Permission denied)"
#radosgw_keystone: false # activate OpenStack Keystone options full detail
here: http://ceph.com/docs/master/radosgw/keystone/
#radosgw_keystone_url: # url:admin_port ie: http://192.168.0.1:35357
#radosgw_keystone_admin_token: password
#radosgw_keystone_accepted_roles: Member, _member_, admin
#radosgw_keystone_token_cache_size: 10000
#radosgw_keystone_revocation_internal: 900
#radosgw_s3_auth_use_keystone: "true"
```

```
#radosgw_nss_db_path: /var/lib/ceph/radosgw/ceph-radosgw.{{ ansible_hostname }}/nss
# Rados Gateway options
#email_address: foo@bar.com

## REST API options
#
#restapi_interface: "{{ monitor_interface }}"
#restapi_address: "{{ monitor_address }}"
#restapi_port: 5000

## Testing mode
# enable this mode _only_ when you have a single node
# if you don't want it keep the option commented
#common_single_host_mode: true


###################
# CONFIG OVERRIDE #
###################

# Ceph configuration file override.
# This allows you to specify more configuration options
# using an INI style format.
# The following sections are supported: [global], [mon], [osd], [mds], [rgw]
#
# Example:
# ceph_conf_overrides:
#   global:
#     foo: 1234
#     bar: 5678
#
```

```
ceph_conf_overrides:
global:
    cephx require signatures: true
    cephx cluster require signatures: true
    osd pool default pg num: 128
    osd pool default pgp num: 128
    mon osd down out interval: 600
    mon osd min down reporters: 7
    mon clock drift allowed: 0.15
    mon clock drift warn backoff: 30
    mon osd report timeout: 900
    mon pg warn max per osd: 0
    mon osd allow primary affinity: true
osd:
    filestore merge threshold: 40
    filestore split multiple: 8
    osd op threads: 8
    filestore op threads: 8
    osd recovery max active: 5
    osd max backfills: 2
    osd recovery op priority: 63
    osd recovery max chunk: 1048576
    osd scrub sleep: 0.1
    osd disk thread ioprio class: idle
    osd disk thread ioprio priority: 0
    osd deep scrub stride: 1048576
    osd scrub chunk max: 5
client:
    rbd concurrent management ops: 20
    rbd default map options: rw
    rbd default format: 2
```

```
#############
# OS TUNING #
#############

#disable_transparent_hugepage: true
#disable_swap: true
os_tuning_params:
  - { name: kernel.pid_max, value: 4194303 }
  - { name: fs.file-max, value: 26234859 }
  - { name: vm.zone_reclaim_mode, value: 0 }
  - { name: vm.vfs_cache_pressure, value: 50 }
  - { name: vm.min_free_kbytes, value: "{{ vm_min_free_kbytes }}" }



##########
# DOCKER #
##########

#docker: false

# Do not comment the variable mon_containerized_deployment_with_kv here. This
variable is being used
# by ceph.conf.j2 template. so it should always be defined
#mon_containerized_deployment_with_kv: false
#mon_containerized_deployment: false



##################
# Temporary Vars #
##################
# NOTE(SamYaple): These vars are set here to they are defined before use.
They
```

```
# should be removed after a refactor has properly seperated all the checks
into

# the appropriate roles.


#journal_collocation: False

#raw_multi_journal: False

#osd_directory: False

#bluestore: False

#dmcrypt_journal_collocation: False

#dmcrypt_dedicated_journal: False
```

Appendix F - /usr/share/ceph-ansible/group_vars/osds Configuration File

```
---

# Variables here are applicable to all host groups NOT roles


# This sample file generated by generate_group_vars_sample.sh


# Dummy variable to avoid error because ansible does not recognize the

# file as a good configuration file when no variable in it.

dummy:


# You can override default vars defined in defaults/main.yml here,

# but I would advice to use host or group vars instead


###########

# GENERAL #

###########


#fetch_directory: fetch/


# Even though OSD nodes should not have the admin key

# at their disposal, some people might want to have it

# distributed on OSD nodes. Setting 'copy_admin_key' to 'true'
# will copy the admin key to the /etc/ceph/ directory
#copy_admin_key: false

####################
# OSD CRUSH LOCATION
####################

# The following options will build a ceph.conf with OSD sections
# Example:
# [osd.X]
# osd crush location = "root=location"
#
# This works with your inventory file
# To match the following 'osd_crush_location' option the inventory must look like:
#
# [osds]
# osd0 ceph_crush_root=foo ceph_crush_rack=bar

crush_location: false
osd_crush_location: "'root={{ ceph_crush_root }} rack={{ ceph_crush_rack }} host={{ ansible_hostname }}'"

##############
# CEPH OPTIONS
##############

# ACTIVATE THE FSID VARIABLE FOR NON-VAGRANT DEPLOYMENT
#fsid: "{{ cluster_uuid.stdout }}"
#cephx: true

# Devices to be used as OSDs
# You can pre-provision disks that are not present yet.
# Ansible will just skip them. Newly added disk will be

# automatically configured during the next run.

#



# !! WARNING !!

#

# /!\ ENABLE ONLY ONE SCENARIO AT A TIME /!\

#

# !! WARNING !!


# Declare devices

# All the scenarii inherit from the following device declaration

#

devices:

  - /dev/sdf

  - /dev/sdg

  - /dev/sdh

  - /dev/sdi

  - /dev/sdj

  - /dev/sdk

  - /dev/sdl

  - /dev/sdm

  - /dev/sdn

  - /dev/sdo

  - /dev/sdp

  - /dev/sdq

  - /dev/sdr

  - /dev/sds

  - /dev/sdt

  - /dev/sdu

  - /dev/sdv
  - /dev/sdw
  - /dev/sdx
  - /dev/sdy
  - /dev/sdz
  - /dev/sdaa
  - /dev/sdab
  - /dev/sdac

# Device discovery is based on the Ansible fact 'ansible_devices'
# which reports all the devices on a system. If chosen all the disks
# found will be passed to ceph-disk. You should not be worried on using
# this option since ceph-disk has a built-in check which looks for empty devices.
# Thus devices with existing partition tables will not be used.
# This mode prevents you from filling out the 'devices' variable above.
#
#osd_auto_discovery: false

# I. First scenario: journal and osd_data on the same device
# Use 'true' to enable this scenario
# This will collocate both journal and data on the same disk
# creating a partition at the beginning of the device

journal_collocation: false


# II. N journal devices for N OSDs
# Use 'true' to enable this scenario
#
# In the following example:
# * sdd and sde will get sdb as a journal
# * sdf and sdg will get sdc as a journal
# While starting you have 2 options:
# 1. Pre-allocate all the devices
# 2. Progressively add new devices
raw_multi_journal: true
raw_journal_devices:
  - /dev/sda
  - /dev/sda
  - /dev/sda
  - /dev/sda
  - /dev/sda
  - /dev/sda
  - /dev/sdb
  - /dev/sdb
  - /dev/sdb
  - /dev/sdb
  - /dev/sdb
  - /dev/sdb
  - /dev/sdc
  - /dev/sdc
  - /dev/sdc
  - /dev/sdc
  - /dev/sdc
  - /dev/sdc
  - /dev/sdd
  - /dev/sdd
  - /dev/sdd
  - /dev/sdd
  - /dev/sdd
  - /dev/sdd

# III. Use directory instead of disk for OSDs
# Use 'true' to enable this scenario
```

```
#osd_directory: false

#osd_directories:

#  - /var/lib/ceph/osd/mydir1

#  - /var/lib/ceph/osd/mydir2



# IV. This will partition disks for BlueStore

# Use 'true' to enable this scenario

#bluestore: false



# V. Encrypt osd data and/or journal devices with dm-crypt.

# Keys are stored into the monitors k/v store

# Use 'true' to enable this scenario

# Both journal and data are stored on the same dm-crypt encrypted device

#dmcrypt_journal_collocation: false



# VI. Encrypt osd data and/or journal devices with dm-crypt.

# Keys are stored into the monitors k/v store

# Use 'true' to enable this scenario

# Journal and osd data are separated, each with their own dm-crypt device

# You must use raw_journal_devices and set your journal devices

#dmcrypt_dedicated_journal: false



##########

# DOCKER #

##########


#osd_containerized_deployment: false

#osd_containerized_deployment_with_kv: false
```

```
#kv_type: etcd
#kv_endpoint: 127.0.0.1
#ceph_osd_docker_prepare_env: ""
#ceph_osd_docker_username: ceph
#ceph_osd_docker_imagename: daemon
#ceph_osd_docker_extra_env: "CEPH_DAEMON=OSD_CEPH_DISK" # comma separated variables
#ceph_osd_docker_devices:
# - /dev/sdb
#ceph_docker_on_openstack: false
```

About the Authors

Oliver Walsdorf, Technical Marketing Engineer for Software Defined Storage, Computer Systems Product Group, Cisco Systems, Inc.

Oliver has more than 20 years of storage experience, working in different roles at different storage vendors, and is now an expert for Software Defined Storage at Cisco. Oliver works on Red Hat Ceph Object Storage and develops Co-Solutions with Red Hat for the overall storage market.

Acknowledgements

- Karan Singh, Senior Storage Architect, Red Hat, Inc.
- Daniel Messer, Senior Alliances Solution Architect, Red Hat, Inc.