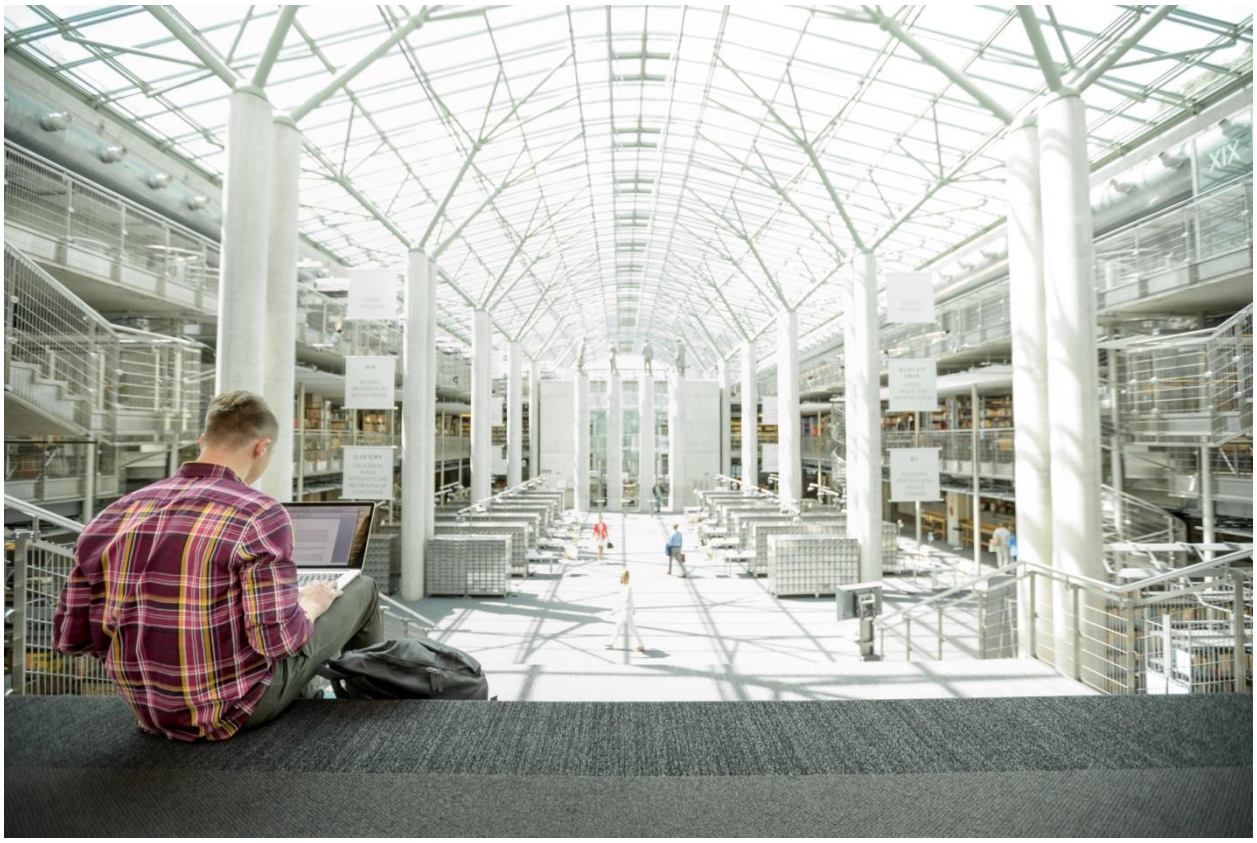


Cisco UCS Storage Server with Scality Ring

Design and Deployment of Scality Object Storage on Cisco UCS S3260 Storage Server

Last Updated: April 10, 2017

CISCO VALIDATED DESIGN

About the Cisco Validated Design (CVD) Program

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

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

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

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

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unified Computing System (Cisco UCS), Cisco UCS B-Series Blade Servers, Cisco UCS C-Series Rack Servers, Cisco UCS S-Series Storage Servers, Cisco UCS Manager, Cisco UCS Management Software, Cisco Unified Fabric, Cisco Application Centric Infrastructure, Cisco Nexus 9000 Series, Cisco Nexus 7000 Series. Cisco Prime Data Center Network Manager, Cisco NX-OS Software, Cisco MDS Series, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

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

© 2017 Cisco Systems, Inc. All rights reserved.

Table of Contents

Executive Summary 6

Solution Overview 7

- Introduction 7
- Solution 7
- Audience 7
- Solution Summary 8

Technology Overview 9

- Cisco Unified Computing System 9
  - Cisco UCS S3260 Storage Server 9
  - Cisco UCS C220 M4 Rack Server 11
  - Cisco UCS Virtual Interface Card 1387 11
  - Cisco UCS 6300 Series Fabric Interconnect 12
  - Cisco Nexus 9332PQ Switch 13
  - Cisco UCS Manager 14
- Red Hat Enterprise Linux 7.3 15
- Scality RING 6.3 15

Solution Design 18

- Deployment Architecture 18
- Solution Overview 19
  - Hardware Requirements 19
  - Software Distributions and Versions 20
  - Hardware Requirements 20
- Physical Topology and Configuration 21

Deployment Hardware and Software 29

- Fabric Configuration 29
- Initial Setup of Cisco UCS 6332 Fabric Interconnects 29
  - Configure Fabric Interconnect A 29
  - Example Setup for Fabric Interconnect A 30
  - Configure Fabric Interconnect B 32
  - Example Setup for Fabric Interconnect B 32
  - Logging into Cisco UCS Manager 33
  - Configure NTP Server 33
- Initial Base Setup of the Environment 34
  - Configure Global Policies 34
  - Enable Fabric Interconnect A Ports for Server 36

Enable Fabric Interconnect A Ports for Uplinks 36
Label Each Server for Identification 37
Create KVM IP Pool 38
Create MAC Pool 39
Create UUID Pool 40
Create VLANs 41
Enable CDP 43
QoS System Class 44
QoS Policy Setup 45
vNIC Template Setup 46
Ethernet Adapter Policy Setup 48
Boot Policy Setup 49
Create LAN Connectivity Policy Setup 50
Create Maintenance Policy Setup 52
Create Power Control Policy Setup 53
Creating Chassis Profile 54
Create Chassis Firmware Package 54
Create Chassis Maintenance Policy 55
Create Disk Zoning Policy 56
Create Chassis Profile Template 58
Create Chassis Profile from Template 61
Associate Chassis Profile 62
Creating Storage Profiles 63
Setting Disks for Cisco UCS C220 M4 Rack-Mount Servers to Unconfigured-Good 63
Create Storage Profile for Cisco UCS S3260 Storage Server 64
Create Storage Profile for Cisco UCS C220 M4S Rack-Mount Servers 67
Creating a Service Profile Template 70
Create Service Profile Template for Cisco UCS S3260 Storage Server Top and Bottom Node 70
Identify Service Profile Template 70
Storage Provisioning 71
Networking 72
vNIC/vHBA Placement 73
Server Boot Order 74
Maintenance Policy 75
Operational Policies 77
Create Service Profile Template for Cisco UCS C220 M4S 78
Create Service Profiles from Template 82

Creating Port Channel for Uplinks....83
Create Port Channel for Fabric Interconnect A/B....83
Configuration of Nexus 9332PQ Switch A and B....84
Initial Setup of Nexus 9332PQ Switch A and B....85
Enable Features on Cisco Nexus 9332PQ Switch A and B....88
Configuring VLANs on Nexus 9332PQ Switch A and B....89
Verification Check of Cisco Nexus C9332PQ Configuration for Switch A and B....98
Installing Red Hat Enterprise Linux 7.3 Operating System....102
Installation of RHEL 7.3 on Cisco UCS C220 M4S....102
Installing RHEL 7.3 on Cisco UCS S3260 Storage Server....104
Post-Installation Steps for Red Hat Enterprise Linux 7.3....106
Preparing all Nodes for Scality RING Installation....109
Scality Salt Installation....110
Scality RING Installation....112
Post-Installation for Scality RING....131
Scality S3 Connector Installation....132
Validation....139
Functional Testing of NFS Connectors....139
Functional Testing of S3 connectors....142
High Availability Testing....144
High-Availability for Hardware Stack....144
HA of Fabric Interconnects....146
HA on Cisco Nexus Switches....147
Hardware Failures of Cisco UCS S3260 and Cisco UCS C220 M4 Servers....148
HA on Connector Nodes....148
Bill of Materials....154
Appendix....157
Appendix A – Kickstart File of Connector Nodes for Cisco UCS C220 M4S....157
Appendix B – Kickstart File of Storage nodes for Cisco UCS S3260 M4 Server....167
Appendix C – Example /etc/hosts File....177
Appendix D – Best Practice Configurations for Ordering Cisco UCS S3260 for Scality....178
Appendix E – Other Best Practices to Consider....180
Appendix F – How to Order Using Cisco UCS S3260 + Scality Solution IDs....180
About the Authors....181
Acknowledgements....181

Executive Summary

Modern data centers increasingly rely on a variety of architectures for storage. Whereas in the past organizations focused on block and file storage only, today organizations are focusing on object storage, for several reasons:

- Object storage offers unlimited scalability and simple management
- Because of the low cost per gigabyte, object storage is well suited for large-capacity needs, and therefore for use cases such as archive, backup, and cloud operations
- Object storage allows the use of custom metadata for objects

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 data latency is not especially important.

Scality object Storage is a massively scalable, software-defined storage system that gives you unified storage for your cloud environment. 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 Scality RING is optimized for object storage solutions, making it an excellent fit for unstructured data workloads such as backup, archive, and cloud data. The S3260 delivers a complete infrastructure with exceptional scalability for computing and storage resources together with 40 Gigabit Ethernet networking. The S3260 is the platform of choice for object 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 Scality RING Storage
- Unified, embedded management for easy-to-scale infrastructure

Cisco and Scality are collaborating to offer customers a scalable object storage solution for unstructured data that is integrated with Scality RING 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. Object storage is the newest approach for handling massive amounts of data.

Scality is an industry leader in enterprise-class, petabyte-scale storage. Scality introduced a revolutionary software-defined storage platform that could easily manage exponential data growth, ensure high availability, deliver high performance and reduce operational cost. Scality's scale-out storage solution, the Scality RING, is based on patented object storage technology and operates seamlessly on any commodity server hardware. It delivers outstanding scalability and data persistence, while its end-to-end parallel architecture provides unsurpassed performance. Scality's storage infrastructure integrates seamlessly with applications through standard storage protocols such as NFS, SMB and S3.

Scale-out object storage uses x86 architecture storage-optimized servers to increase performance while reducing costs. The Cisco UCS S3260 Storage Server is well suited for object-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, Scality object 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 object storage solution on Scality RING and Cisco UCS S3260 Storage Server. The solution includes the following features:

- Infrastructure for large scale object storage
- Design of a Scality object 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

Audience

This document describes the architecture, design and deployment procedures of a Scality object Storage solution using six Cisco UCS S3260 Storage Server with two C3X60 M4 server nodes each as Storage nodes, two Cisco UCS C220 M4 S rack server each as connector nodes, one Cisco UCS C220 M4S rackserver as Supervisor 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 Scality object 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 Scality object Storage on Cisco UCS S3260 Storage Server.

The configuration uses the following architecture for the deployment:

- 6 x Cisco UCS S3260 Storage Server with 2 x C3X60 M4 server nodes working as Storage nodes
- 3 x Cisco UCS C220 M4S rack server working as Connector nodes
- 1 x Cisco UCS C220 M4S rack server working as Supervisor node
- 2 x Cisco UCS 6332 Fabric Interconnect
- 1 x Cisco UCS Manager
- 2 x Cisco Nexus 9332PQ Switches
- Scality RING 6.3
- Redhat Enterprise Linux Server 7.3

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, 40-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 and other unstructured data repositories, media streaming, and content distribution.

Figure 1 Cisco UCS S3260 Storage Server

Extending the capability of the Cisco UCS S-Series 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 14 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 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.

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:

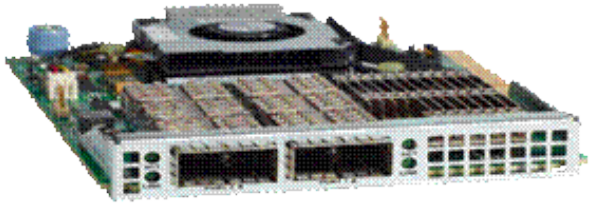
- Up to 24 DDR4 DIMMs for improved performance and lower power consumption
- Up to 8 Small Form-Factor (SFF) drives or up to 4 Large Form-Factor (LFF) drives
- Support for 12-Gbps SAS Module RAID controller in a dedicated slot, leaving the remaining two PCIe Gen 3.0 slots available for other expansion 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 3260 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 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 9332PQ

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.

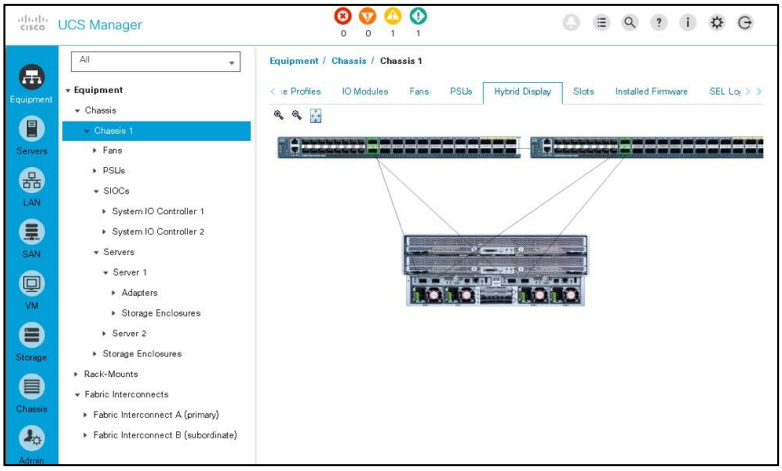
With the Cisco Nexus 9000 Series, organizations can quickly and easily upgrade existing data centers to carry 40 Gigabit Ethernet to the aggregation layer or to the spine (in a leaf-and-spine configuration) through advanced and cost-effective optics that enable the use of existing 10 Gigabit Ethernet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

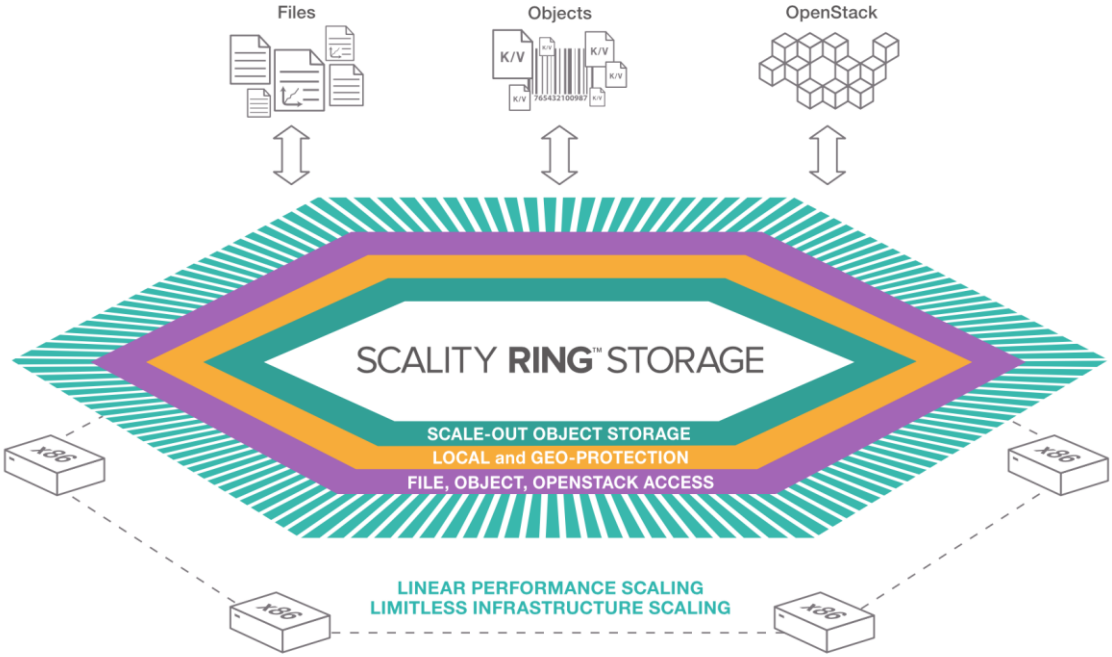
- Delivers high performance, reliability, and security
- Is certified by the leading hardware and software vendors
- Scales from workstations, to servers, to mainframes
- Provides a consistent application environment across physical, virtual, and cloud deployments

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.

Scality RING 6.3

Scality RING 6.3 (Figure 7) sets a new standard, enabling many more enterprises and services providers to benefit from object storage through enhanced S3 API support with a uniquely enterprise-ready identity and security model.

Figure 7 Scality RING Architecture



In addition, customers with high-scale compliance needs can now take advantage of the standards-based interfaces and hardware-independent capabilities of the RING. For file system users, Scality continues to enhance the RING's file support, now improving performance for specific applications like backup and media.

Featured highlights and benefits:

- Enables Enterprise-Ready object Storage Deployment with Strong S3 Features, Security, and Performance
- Scality RING 6.3 is the first S3-compatible object storage with full Microsoft Active Directory and AWS IAM support
- RING 6.3 offers exceptional levels of S3 API performance, including scale-out Bucket access, even across multiple locations
- Protects Petabytes of Records, Images, and More with the Most Scalable Data Compliance Solution
- Standards-based, compliance solution that scales into petabytes in a single system
- Tackles More Enterprise Applications with Enhanced Scale-out File System Capabilities
- Fully parallel and multi-user write performance to the same directories, further enabling specific backup and media applications
- Integrated Load Balancing and Failover across multiple file system interfaces
- High performance at scale - Linear performance scale to many petabytes of data and Supports a broad mix of application workloads

- 100% reliable - Zero downtime for maintenance and expansion, zero downtime when disk, server, rack, and site fail & Always available and durable with native geo-redundancy.

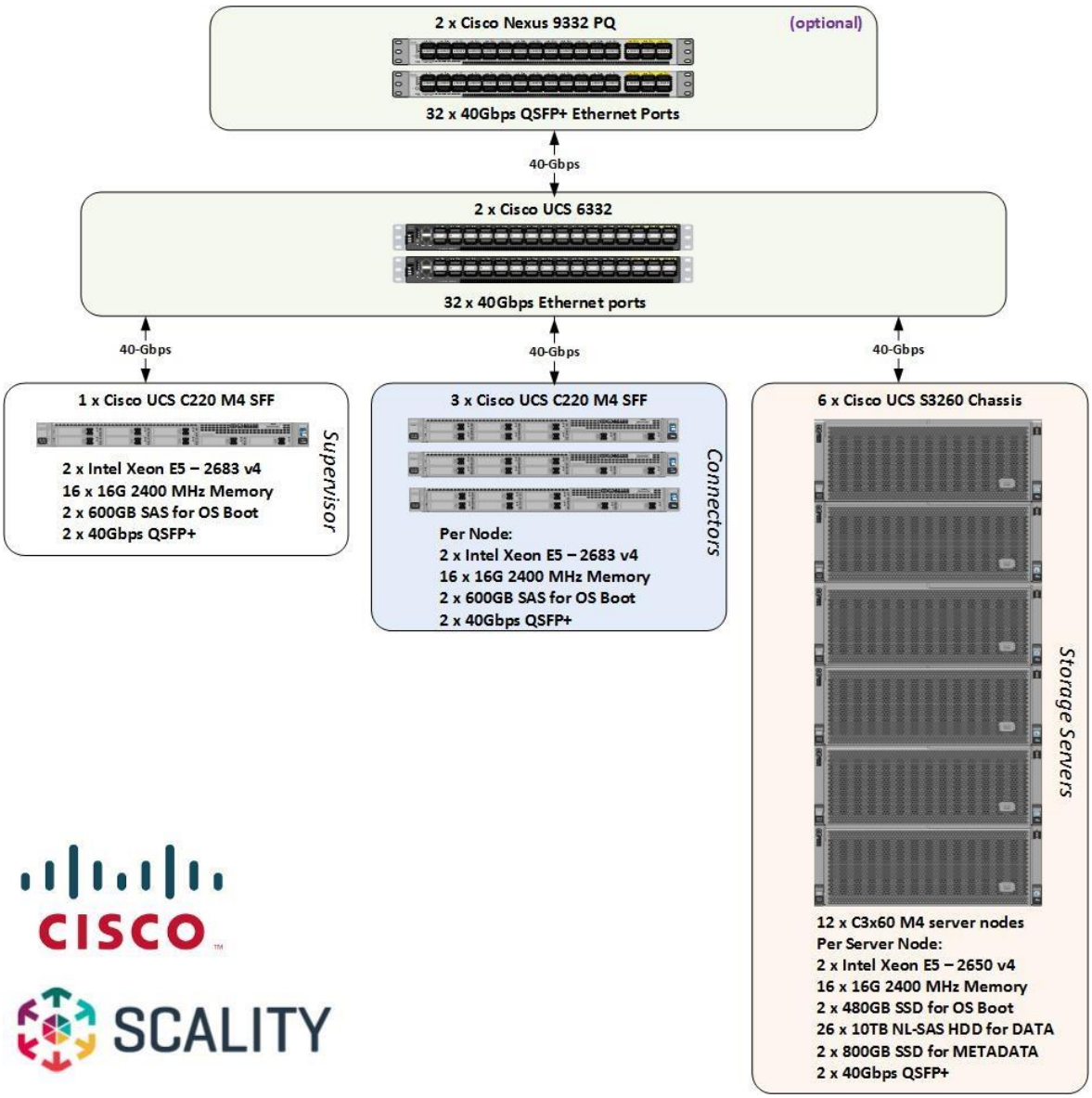
Solution Design

Deployment Architecture

The reference architecture use case provides a comprehensive, end-to-end example of deploying Scality object storage on Cisco UCS S3260 (Figure 8).

The first section in this Cisco Validated Design covers setting up the Cisco UCS hardware; the Cisco UCS 6332 Fabric Interconnects (Cisco UCS Manager), Cisco UCS S3260 Storage servers, Cisco UCS C220 M4 Rack Servers, and the peripherals like Cisco Nexus 9332 switches. The second section explains the step-by-step installation instructions to install Scality RING. The final section includes the functional and High Availability tests on the test bed, performance, and the best practices evolved while validating the solution.

Figure 8 Cisco UCS SDS Architecture



Solution Overview

The current solution based on Cisco UCS and Scality object Storage is divided into multiple sections and covers three main aspects.

Hardware Requirements

This CVD describes the architecture, design and deployment of a Scality object Storage solution on six Cisco UCS S3260 Storage Server, each with two Cisco UCS C3X60 M4 nodes configured as Storage servers and 3 Cisco UCS C220 M4S Rack servers as three Connector nodes and one Supervisor node. The whole solution is connected to the pair of Cisco UCS 6332 Fabric Interconnects and to pair of upstream network switch Cisco Nexus 9332PQ.

The detailed configuration is as follows:

- Two Cisco Nexus 9332PQ Switches
- Two Cisco UCS 6332 Fabric Interconnects
- Six Cisco UCS S3260 Storage Servers with two UCS C3X60 M4 server nodes each
- Three Cisco UCS C220 M4S Rack Servers

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

Software Distributions and Versions

The required software distribution versions are listed below in Table 1 .

Table 1 Software Versions

Layer	Component	Version or Release
Storage (Chassis) UCS S3260	Chassis Management Controller	2.0(13e)
	Shared Adapter	4.1(2d)
Compute (Server Nodes) UCS C3X60 M4	BIOS	C3x60M4.2.0.13c
	CIMC Controller	2.0(13f)
Compute (Rack Server) C220 M4S	BIOS	C220M4.2.0.13d
	CIMC Controller	2.0(13f)
Network 6332 Fabric Interconnect	UCS Manager	3.1(2b)
	Kernel	5.0(3)N2(3.12b)
	System	5.0(3)N2(3.12b)
Network Nexus 9332PQ	BIOS	07.59
	NXOS	7.0(3)I5(1)
Software	Red Hat Enterprise Linux Server	7.3 (x86_64)
	Scality RING	6.3

Hardware Requirements

This section contains the hardware components (Table 2) used in the test bed.

Table 2 Hardware Requirements

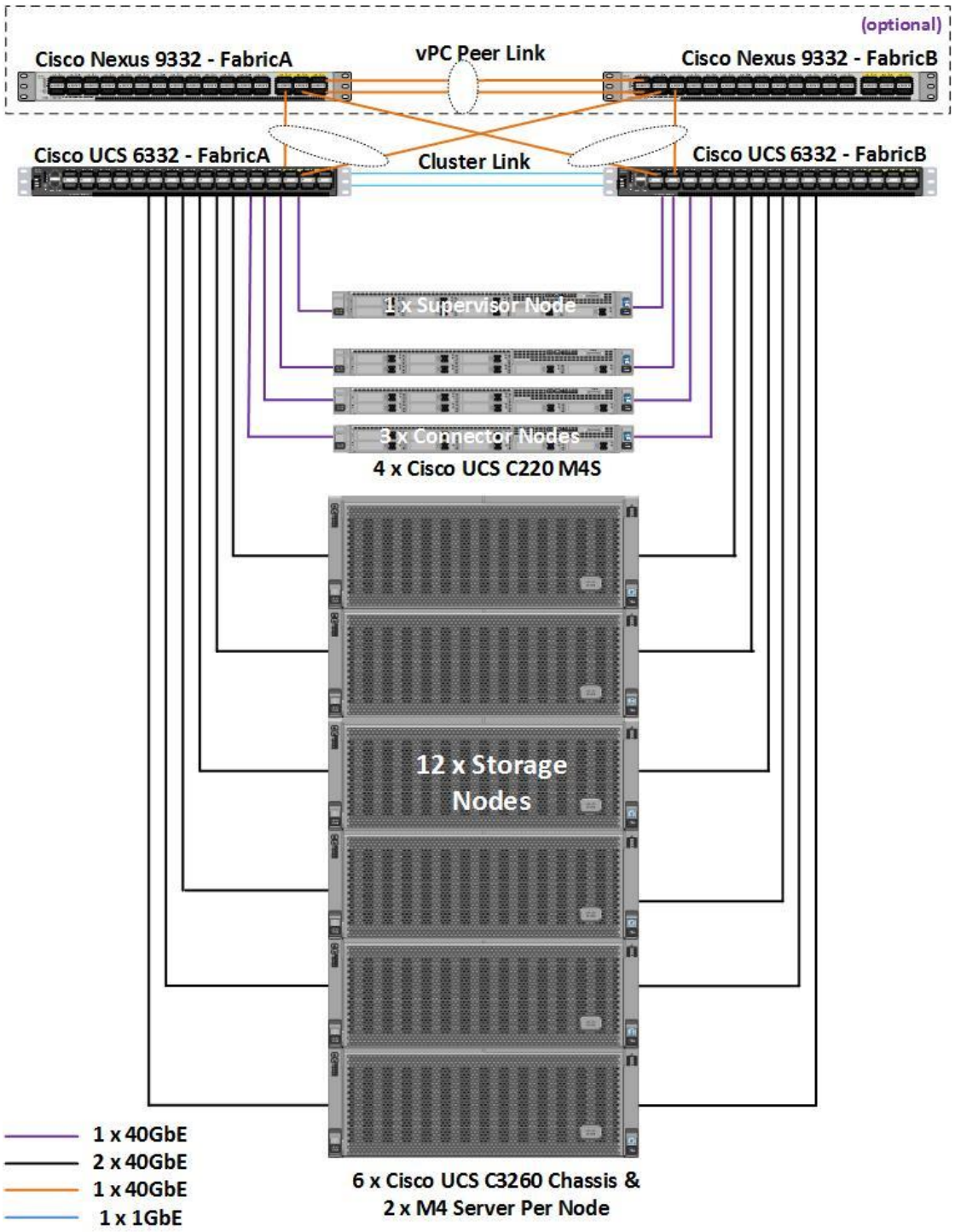
Component	Model	Quantity	Comments
Scality Storage node	Cisco UCS S3260 M4 Chassis	6	• 2 x UCS C3X60 M4 Server Nodes per Chassis (Total = 12nodes)

Component	Model	Quantity	Comments
			• Per Server Node – 2 x Intel E5-2650 v4, 256 GB RAM – Cisco 12G SAS RAID Controller – 2 x 480 GB SSD for OS, 26 x 10TB HDDs for Data, 2 x 800G SSD for Metadata – Dual-port 40 Gbps VIC
Scality Connector Nodes	Cisco UCS C220M4S Rack server	3	• 2 x Intel E5-2683v4, 256 GB RAM • Cisco 12G SAS RAID Controller • 2 x 600 GB SAS for OS • Dual-port 40 Gbps VIC
Scality Supervisor Node	Cisco UCS C220M4S Rack server	1	• 2 x Intel E5-2683v4, 256 GB RAM • Cisco 12G SAS RAID Controller • 2 x 600 GB SAS for OS • Dual-port 40 Gbps VIC
UCS Fabric Interconnects	Cisco UCS 6332 Fabric Interconnects	2	
Switches	Cisco Nexus 9332PQ Switches	2	

Physical Topology and Configuration

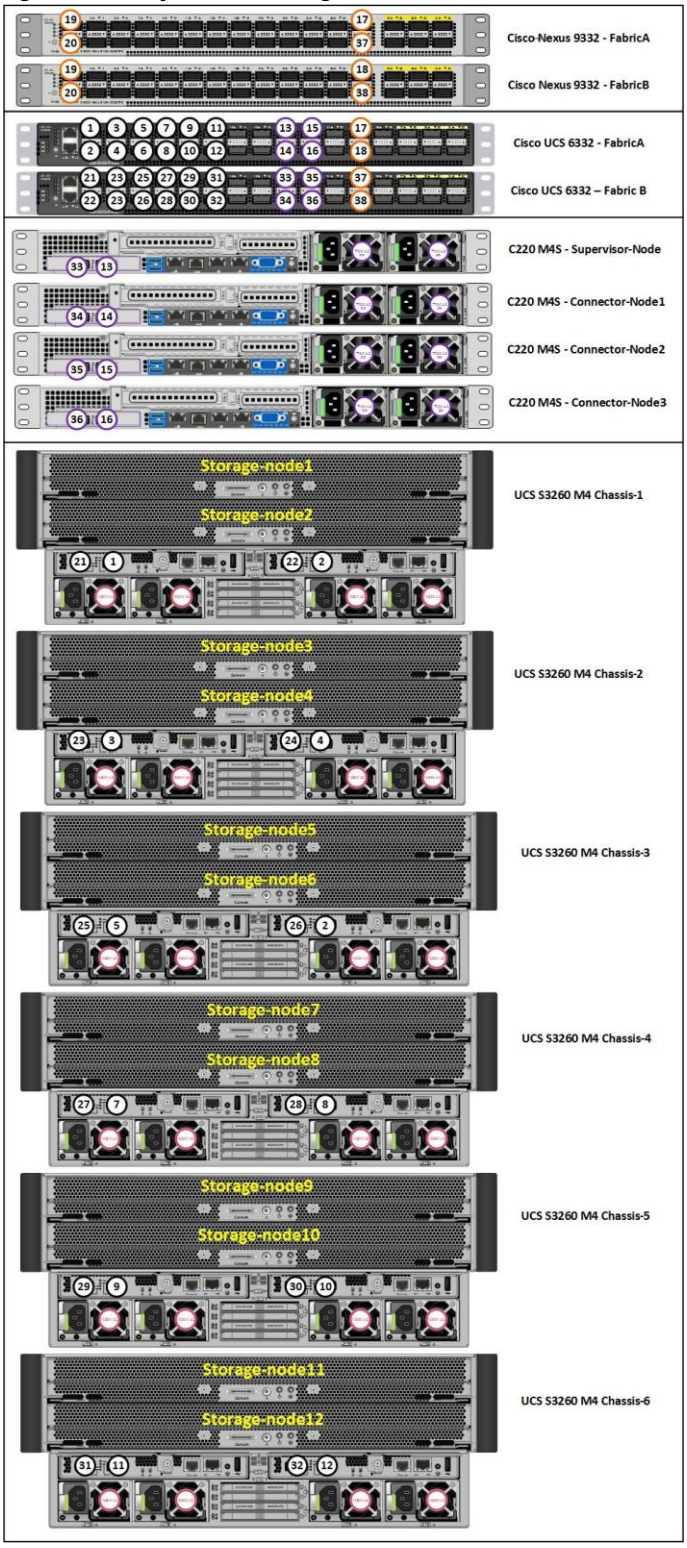
The following sections describe the physical design of the solution and the configuration of each component.

Figure 9 Physical Topology of the Solution



The connectivity of the solution is based on 40 Gbit. All components are connected together via 40 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 is connected via 1 x 40 Gbit and each Cisco UCS S3260 M4 server is connected with 2 x 40 Gbit cable to each Fabric Interconnect.

Figure 10 Physical Cabling of the Solution



The exact cabling for the Cisco UCS S3260 Storage Server, Cisco UCS C220 M4S, and the Cisco UCS 6332 Fabric Interconnect is illustrated in Table 3 .

Table 3 Cabling Information

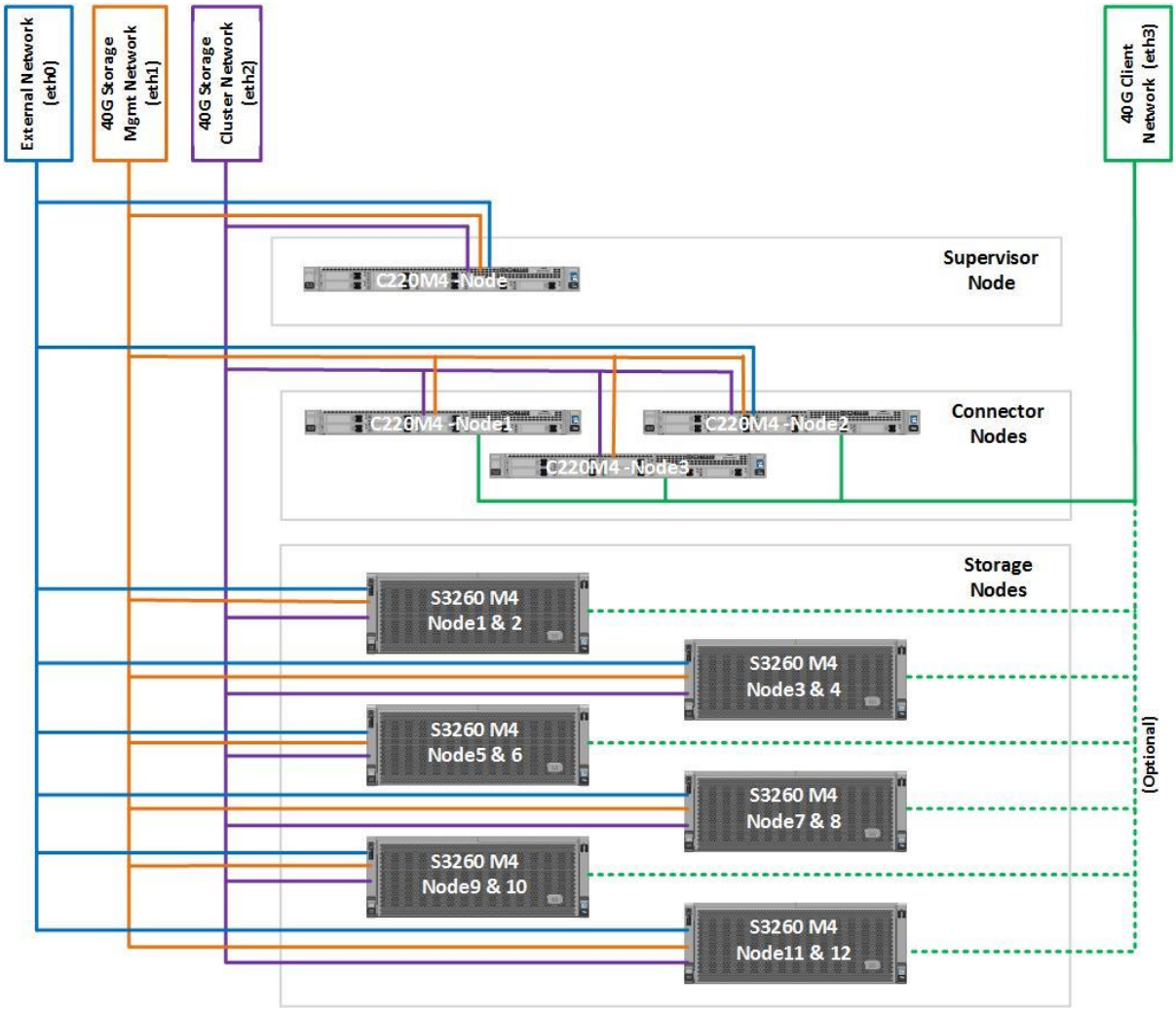
Local Device	Local Port	Connection	Remote Device	Remote Port	Cable
Cisco Nexus 9332 Switch A	Eth1/1	40GbE	Cisco Nexus 9372 Switch B	Eth1/1	QSFP-H40G-CU1M
	Eth1/2	40GbE	Cisco Nexus 9372 Switch B	Eth1/2	QSFP-H40G-CU1M
	Eth1/17	40GbE	Cisco UCS Fabric Interconnect A	Eth1/17	QSFP-H40G-CU1M
	Eth1/18	40GbE	Cisco UCS Fabric Interconnect B	Eth1/17	QSFP-H40G-CU1M
	Eth1/23	40GbE	Top of Rack (Upstream Network)	Any	QSFP+ 4SFP10G
	MGMT0	1GbE	Top of Rack (Management)	Any	1G RJ45
Cisco Nexus 9332 Switch B	Eth1/1	40GbE	Cisco Nexus 9372 Switch B	Eth1/1	QSFP-H40G-CU1M
	Eth1/2	40GbE	Cisco Nexus 9372 Switch B	Eth1/2	QSFP-H40G-CU1M
	Eth1/17	40GbE	Cisco UCS Fabric Interconnect A	Eth1/18	QSFP-H40G-CU1M
	Eth1/18	40GbE	Cisco UCS Fabric Interconnect B	Eth1/18	QSFP-H40G-CU1M
	Eth1/23	40GbE	Top of Rack (Upstream Network)	Any	QSFP+ 4SFP10G
	MGMT0	1GbE	Top of Rack (Management)	Any	1G RJ45
Cisco UCS 6332 Fabric Interconnect A	Eth1/1	40GbE	S3260 Chassis 1 - SIOC 1 (right)	port 1	QSFP-H40G-CU3M
	Eth1/2	40GbE	S3260 Chassis 1 - SIOC 2 (left)	port 1	QSFP-H40G-CU3M

Local Device	Local Port	Connection	Remote Device	Remote Port	Cable
	Eth1/3	40GbE	S3260 Chassis 2 - SIOC 1 (right)	port 1	QSFP-H40G-CU3M
	Eth1/4	40GbE	S3260 Chassis 2 - SIOC 2 (left)	port 1	QSFP-H40G-CU3M
	Eth1/5	40GbE	S3260 Chassis 3 - SIOC 1 (right)	port 1	QSFP-H40G-CU3M
	Eth1/6	40GbE	S3260 Chassis 3 - SIOC 2 (left)	port 1	QSFP-H40G-CU3M
	Eth1/7	40GbE	S3260 Chassis 4 - SIOC 1 (right)	port 1	QSFP-H40G-CU3M
	Eth1/8	40GbE	S3260 Chassis 4 - SIOC 2 (left)	port 1	QSFP-H40G-CU3M
	Eth1/9	40GbE	S3260 Chassis 5 - SIOC 1 (right)	port 1	QSFP-H40G-CU3M
	Eth1/10	40GbE	S3260 Chassis 5 - SIOC 2 (left)	port 1	QSFP-H40G-CU3M
	Eth1/11	40GbE	S3260 Chassis 6 - SIOC 1 (right)	port 1	QSFP-H40G-CU3M
	Eth1/12	40GbE	S3260 Chassis 6 - SIOC 2 (left)	port 1	QSFP-H40G-CU3M
	Eth1/17	40GbE	C220 M4S - Server1 - VIC1387	VIC - Port 1	QSFP-H40G-CU1M
	Eth1/18	40GbE	C240 M4S - Server2 - VIC1387	VIC - Port 1	QSFP-H40G-CU1M
	Eth1/19	40GbE	C240 M4S - Server3 - VIC1387	VIC - Port 1	QSFP-H40G-CU1M
	Eth1/20	40GbE	C240 M4S - Server4 - VIC1387	VIC - Port 1	QSFP-H40G-CU1M

Local Device	Local Port	Connection	Remote Device	Remote Port	Cable
	Eth1/25	40GbE	Nexus 9332 A	Eth 1/25	QSFP-H40G-CU1M
	Eth1/26	40GbE	Nexus 9332 B	Eth 1/25	QSFP-H40G-CU1M
	MGMT0	40GbE	Top of Rack (Management)	Any	1G RJ45
	L1	1GbE	UCS 6332 Fabric Interconnect B	L1	1G RJ45
	L2	1GbE	UCS 6332 Fabric Interconnect B	L2	1G RJ45
Cisco UCS 6332 Fabric Interconnect B	Eth1/1	40GbE	S3260 Chassis 1 - SIOC 1 (right)	port 2	QSFP-H40G-CU3M
	Eth1/2	40GbE	S3260 Chassis 1 - SIOC 2 (left)	port 2	QSFP-H40G-CU3M
	Eth1/3	40GbE	S3260 Chassis 2 - SIOC 1 (right)	port 2	QSFP-H40G-CU3M
	Eth1/4	40GbE	S3260 Chassis 2 - SIOC 2 (left)	port 2	QSFP-H40G-CU3M
	Eth1/5	40GbE	S3260 Chassis 3 - SIOC 1 (right)	port 2	QSFP-H40G-CU3M
	Eth1/6	40GbE	S3260 Chassis 3 - SIOC 2 (left)	port 2	QSFP-H40G-CU3M
	Eth1/7	40GbE	S3260 Chassis 4 - SIOC 1 (right)	port 2	QSFP-H40G-CU3M
	Eth1/8	40GbE	S3260 Chassis 4 - SIOC 2 (left)	port 2	QSFP-H40G-CU3M
	Eth1/9	40GbE	S3260 Chassis 5 - SIOC 1 (right)	port 2	QSFP-H40G-CU3M
	Eth1/10	40GbE	S3260 Chassis 5 - SIOC 2 (left)	port 2	QSFP-H40G-CU3M

Local Device	Local Port	Connection	Remote Device	Remote Port	Cable
	Eth1/11	40GbE	S3260 Chassis 6 - SIOC 1 (right)	port 2	QSFP-H40G-CU3M
	Eth1/12	40GbE	S3260 Chassis 6 - SIOC 2 (left)	port 2	QSFP-H40G-CU3M
	Eth1/17	40GbE	C220 M4S - Server1 - VIC1387	VIC - Port 2	QSFP-H40G-CU1M
	Eth1/18	40GbE	C240 M4S - Server2 - VIC1387	VIC - Port 2	QSFP-H40G-CU1M
	Eth1/19	40GbE	C240 M4S - Server3 - VIC1387	VIC - Port 2	QSFP-H40G-CU1M
	Eth1/20	40GbE	C240 M4S - Server4 - VIC1387	VIC - Port 2	QSFP-H40G-CU1M
	Eth1/25	40GbE	Nexus 9332 A	Eth 1/26	QSFP-H40G-CU1M
	Eth1/26	40GbE	Nexus 9332 B	Eth 1/26	QSFP-H40G-CU1M
	MGMT0	40GbE	Top of Rack (Management)	Any	1G RJ45
	L1	1GbE	UCS 6332 Fabric Interconnect A	L1	1G RJ45
	L2	1GbE	UCS 6332 Fabric Interconnect A	L2	1G RJ45

Figure 11 Network Layout of the Solution



Deployment 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 using 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

To set up the Cisco UCS 6332 Fabric Interconnects A and B, complete the following steps:

Configure Fabric Interconnect A

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 UCS-**FI-6332** 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:  UCS-FI-6332

  Physical Switch Mgmt0 IP address : 192.168.10.101

  Physical Switch Mgmt0 IPv4 netmask : 255.255.255.0

  IPv4 address of the default gateway : 192.168.10.1

  Cluster IPv4 address : 192.168.10.100

  Configure the DNS Server IP address? (yes/no) [n]: no

  Configure the default domain name? (yes/no) [n]: no

  Join centralized management environment (UCS Central)? (yes/no) [n]: no


  Following configurations will be applied:


    Switch Fabric=A

    System Name= UCS-FI-6332

    Enforced Strong Password=no

    Physical Switch Mgmt0 IP Address=192.168.10.101

    Physical Switch Mgmt0 IP Netmask=255.255.255.0

    Default Gateway=192.168.10.1

    Ipv6 value=0


    Cluster Enabled=yes

    Cluster IP Address=192.168.10.100

    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
UCS-FI-6332-A login:
```

Configure Fabric Interconnect B

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: 192.168.10.101
    Peer Fabric interconnect Mgmt0 IPv4 Netmask: 255.255.255.0
    Cluster IPv4 address          : 192.168.10.100


    Peer FI is IPv4 Cluster enabled. Please Provide Local Fabric Interconnect
Mgmt0 IPv4 Address
  Physical Switch Mgmt0 IP address : 192.168.10.102


  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
UCS-FI-6332-B login:
```

Logging into Cisco UCS Manager

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

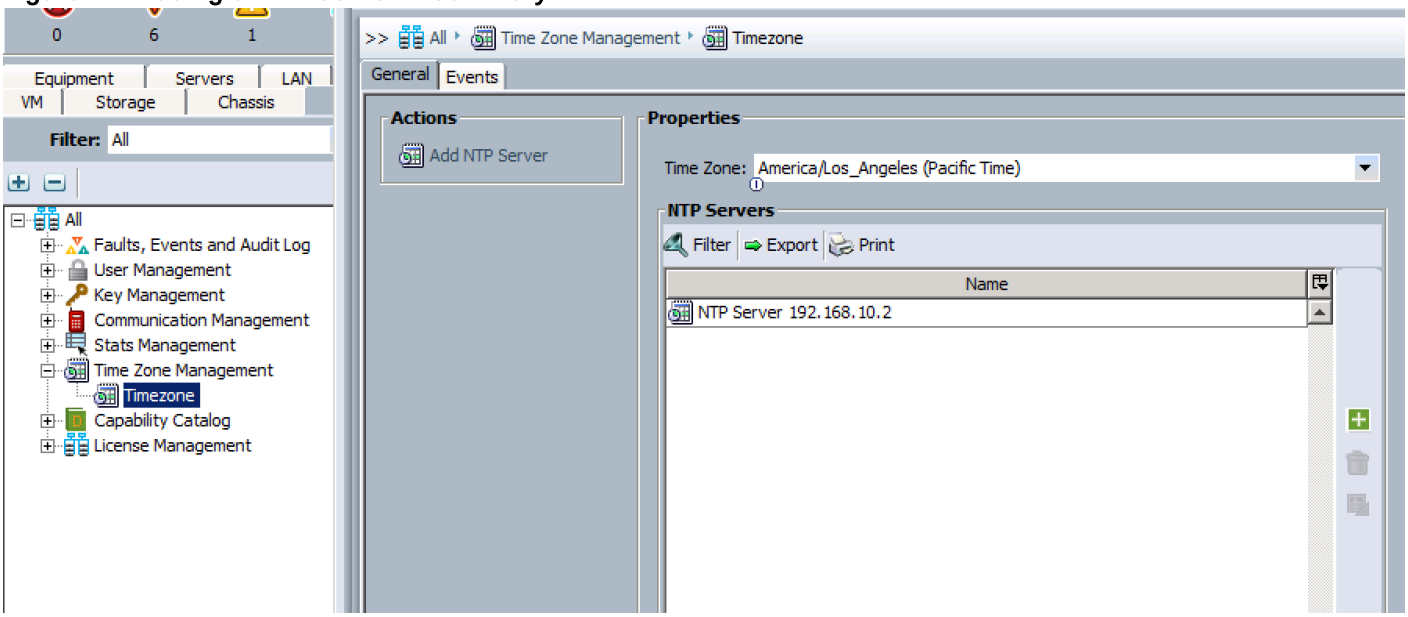
1. Open a Web browser and navigate to the Cisco UCS 6332 Fabric Interconnect cluster address.
2. Click the Launch link to download the Cisco UCS Manager software.
3. If prompted to accept security certificates, accept as necessary.
4. Click Launch UCS Manager HTML.
5. When prompted, enter **admin** for the username and enter the administrative password.
6. Click Login to log in to the Cisco UCS Manager.

Configure NTP Server

This section describes how to configure the NTP server for the Cisco UCS environment.

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 12 Adding a NTP server - Summary



Initial Base Setup of the Environment

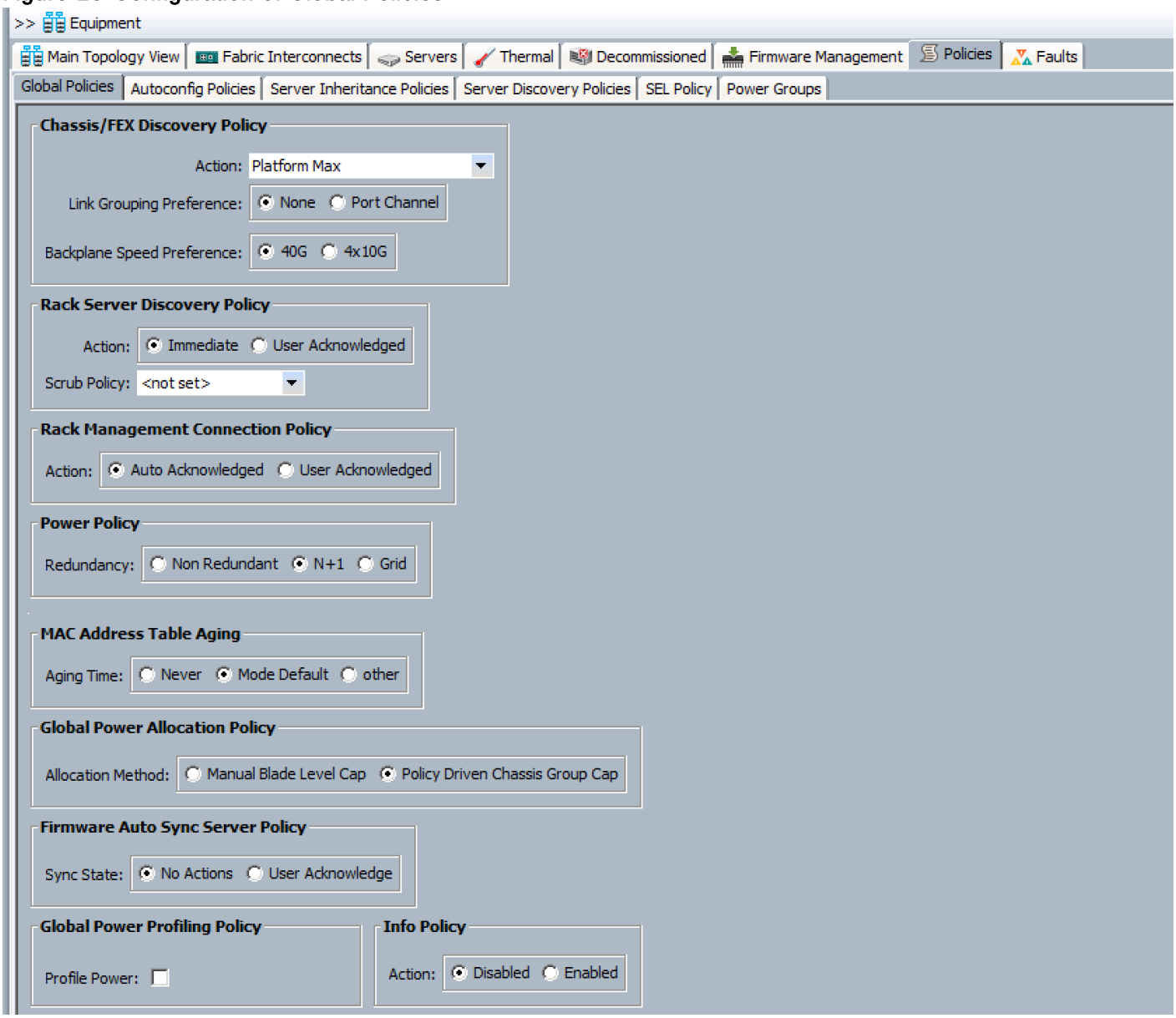
Configure Global Policies

This section describes how to configure the global policies.

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 13 Configuration of Global Policies

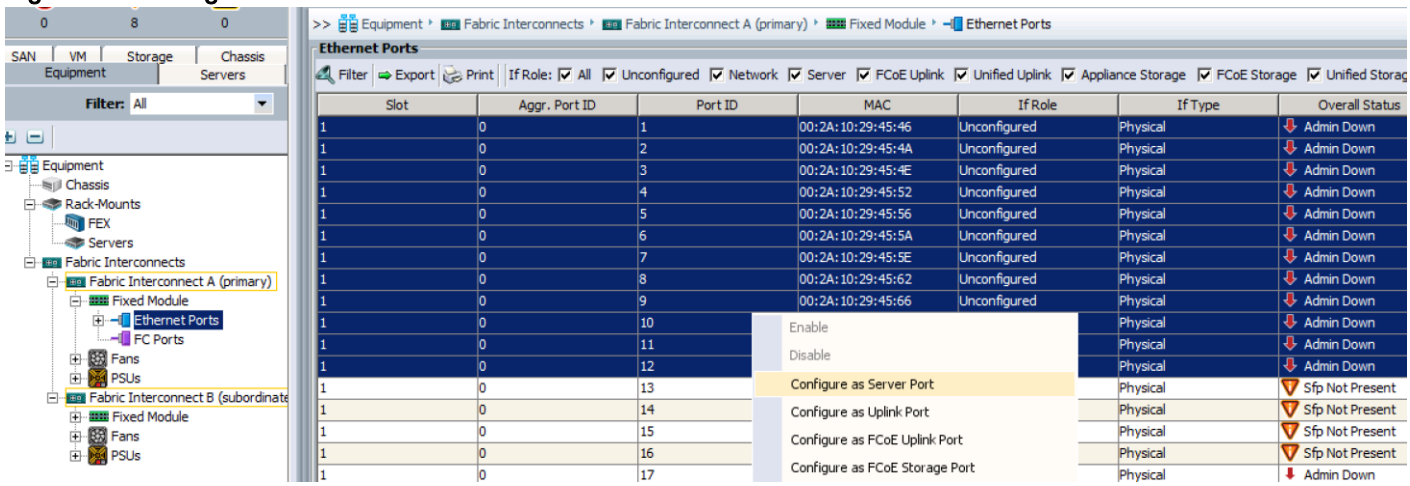


Enable Fabric Interconnect A Ports for Server

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 1-12, right-click and then select **Configure as Server Port** and click **Yes** and then **OK**.
5. Select Ports 17-20 for C220 M4S server, right-click and then select "**Configure as Server Port**" and click **Yes** and then **OK**.
6. Repeat the same steps for Fabric Interconnect B.

Figure 14 Configuration of Server Ports



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 25-26, 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 Server for Identification

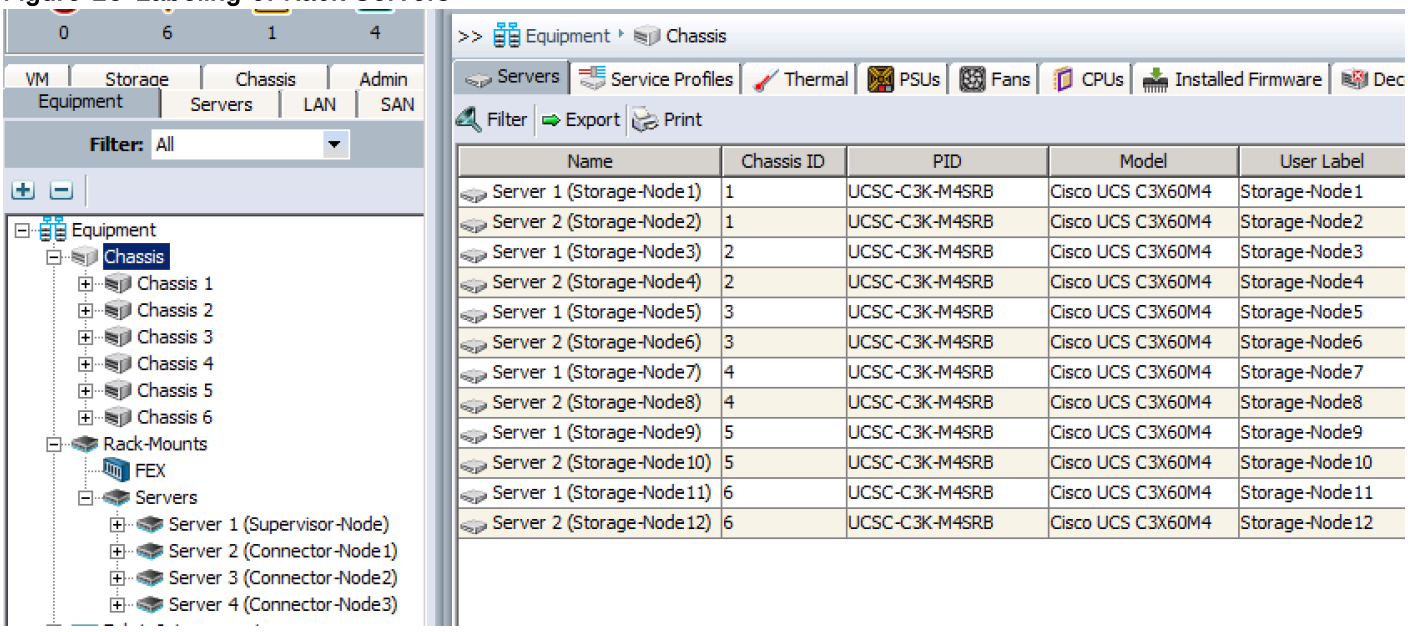
To label each server (provides better identification), complete the following steps:

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 **Storage-Node1** to the field.
4. Repeat the previous steps for **Server 2** of **Chassis 1** and for all other servers of Chassis 2 – 6 according to Table 2.
5. Go to **Servers > Rack-Mounts > Servers >** and repeat the step for all servers according to Table 4 .

Table 4 Server Label

Server	Name
Chassis 1 / Server 1	Storage-Node1
Chassis 1 / Server 2	Storage-Node2
Chassis 1 / Server 3	Storage-Node3
Chassis 1 / Server 4	Storage-Node4
Chassis 1 / Server 5	Storage-Node5
Chassis 1 / Server 6	Storage-Node6
Chassis 1 / Server 7	Storage-Node7
Chassis 1 / Server 8	Storage-Node8
Chassis 1 / Server 9	Storage-Node9
Chassis 1 / Server 10	Storage-Node10
Chassis 1 / Server 11	Storage-Node11
Chassis 1 / Server 12	Storage-Node12
Rack-Mount / Server 1	Supervisor
Rack-Mount / Server 2	Connector-Node1
Rack-Mount / Server 3	Connector-Node2
Rack-Mount / Server 4	Connector-Node3

Figure 15 Labeling of Rack Servers

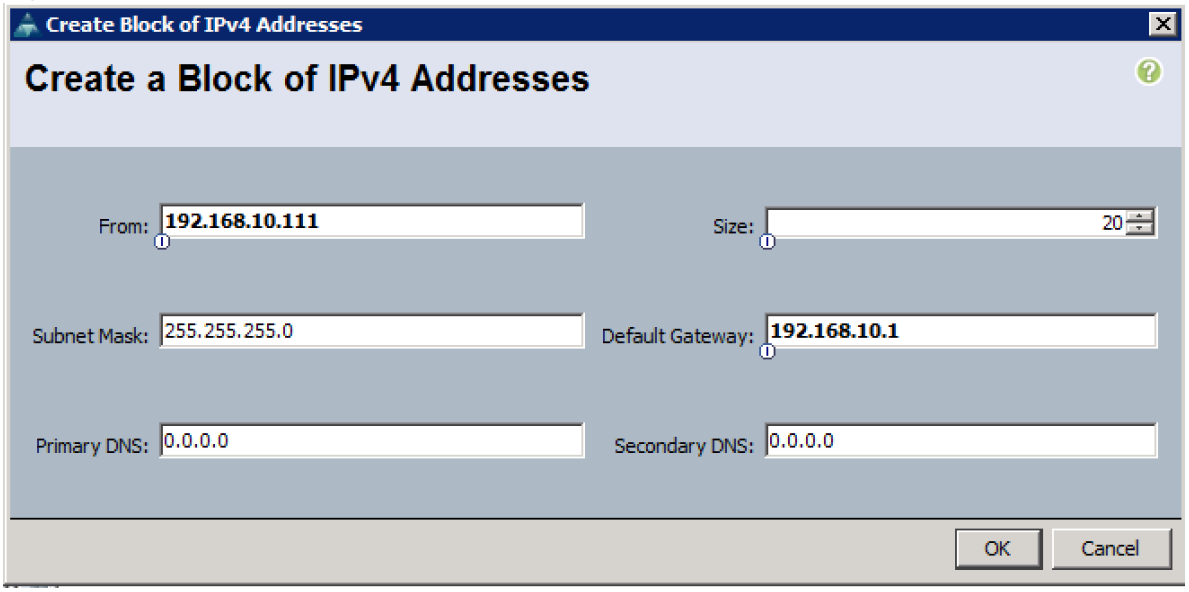


Create KVM IP Pool

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 16 Create Block of IPv4 Addresses

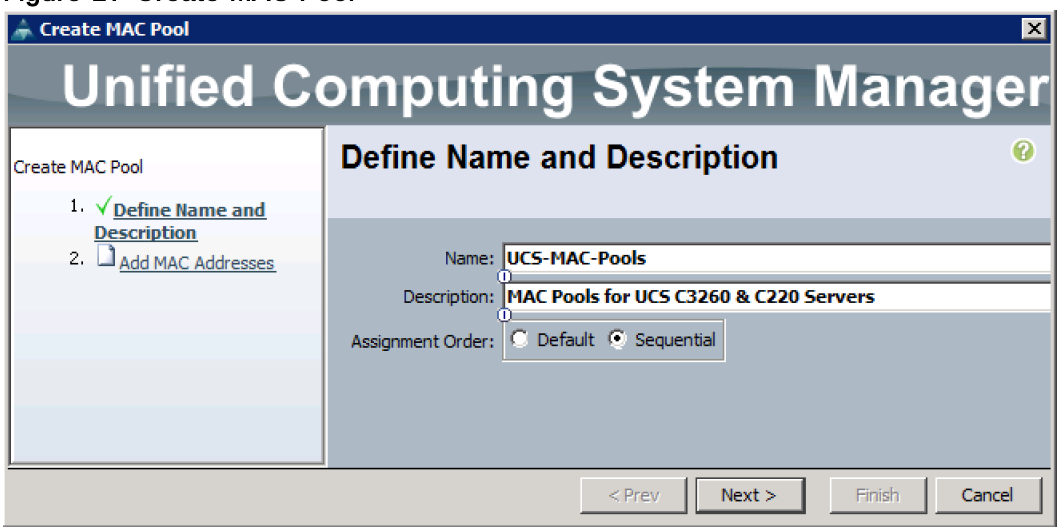


Create MAC Pool

To create a MAC Pool, complete the following steps:

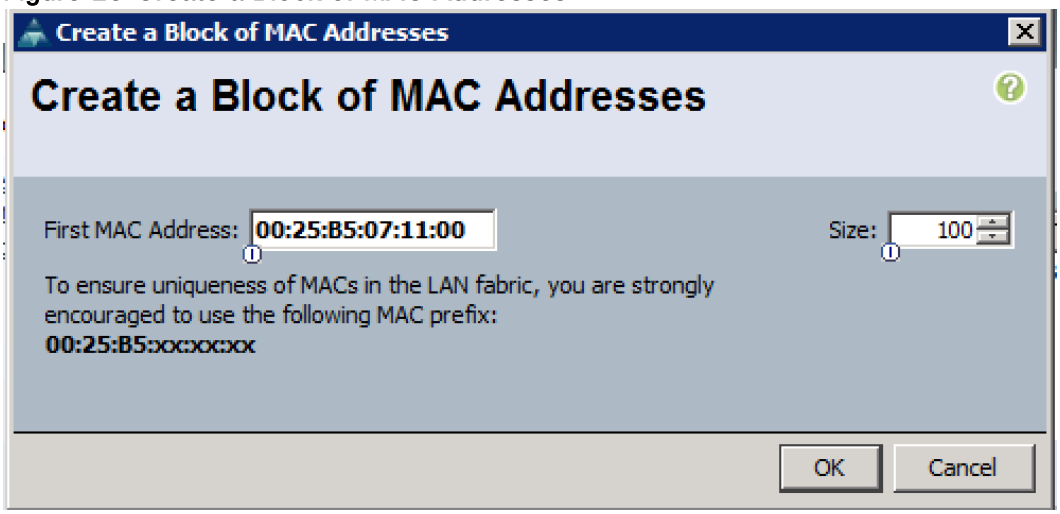
1. Select the **LAN** tab on the left site.
2. Go to LAN > Pools > root > Mac Pools and right-click Create MAC Pool.
3. Type in UCS--MAC-Pools for Name.
4. (Optional) Enter a **Description** of the MAC Pool.
5. Set Assignment Order as Sequential.

Figure 17 Create MAC Pool



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 18 Create a Block of MAC Addresses



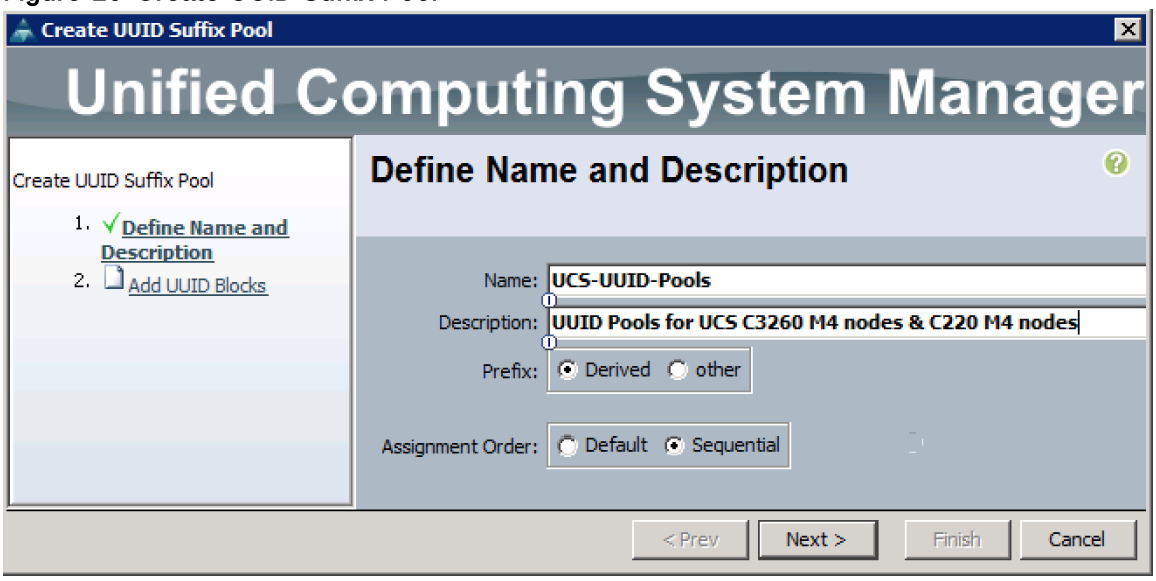
10. Click **OK**.
11. Click **Finish**.

Create UUID Pool

To create a UUID Pool, complete the following steps:

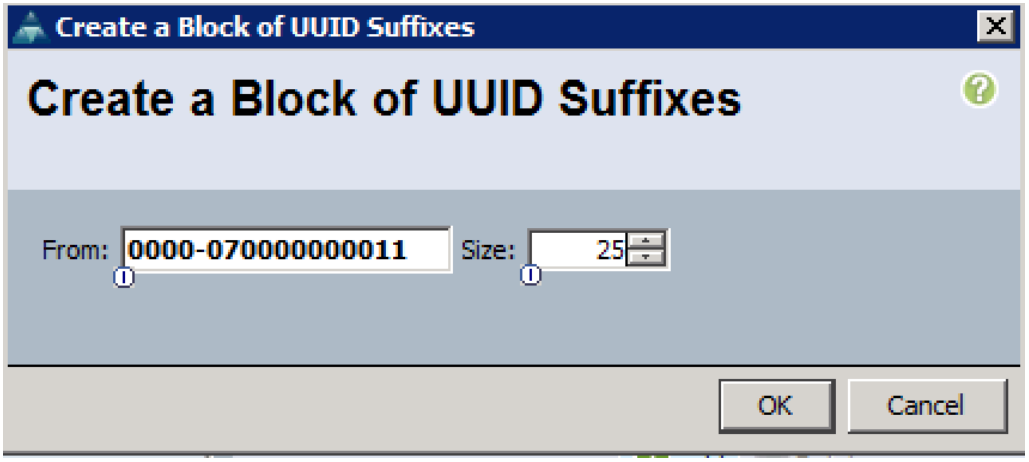
1. Select the **Servers** tab on the left site.
2. Go to Servers > Pools > root > UUID Suffix Pools and right-click Create UUID Suffix Pool.
3. Type in **UCS-UUID-Pools** for Name.
4. (Optional) Enter a **Description** of the MAC Pool.
5. Set Assignment Order to Sequential and click Next.

Figure 19 Create UUID Suffix Pool



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, 25.

Figure 20 Create a Block of UUID Suffixes



9. Click **OK**.
10. Click **Finish** and then **OK**.

Create VLANs

As mentioned previously, it is important to separate the network traffic with VLANs for Storage-Management traffic and Storage-Cluster traffic, External traffic, and Client traffic (optional). Table 5 lists the configured VLANs.

Note: Client traffic is optional. We used Client traffic, to validate the functionality of NFS & S3 connectors.

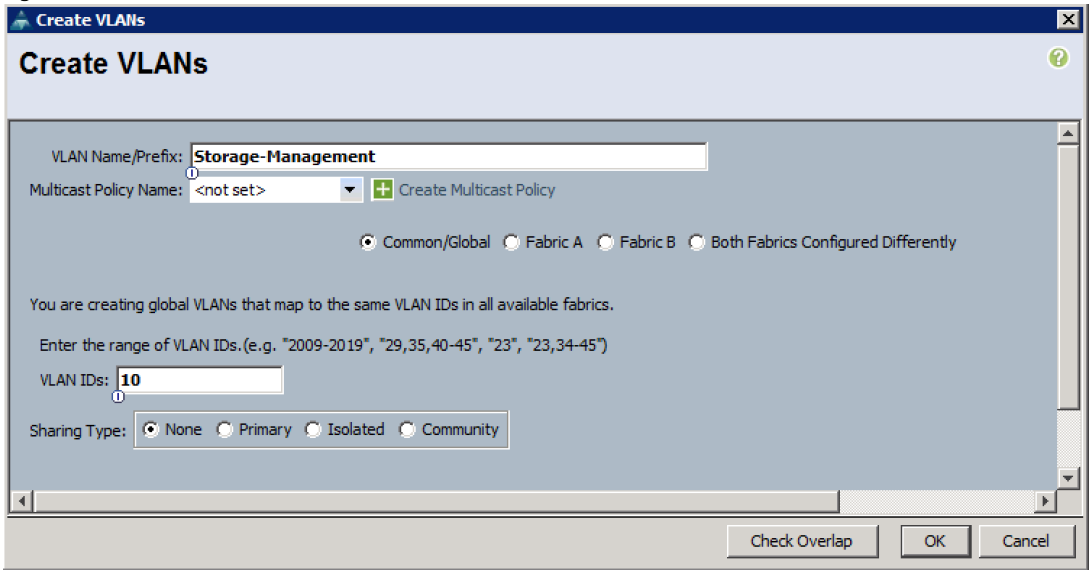
Table 5 VLAN Configurations

VLAN	Name	Function
10	Storage-Management	Storage Management traffic for Supervisor, Connector & Storage Nodes
20	Storage-Cluster	Storage Cluster traffic for Supervisor, Connector & Storage Nodes
30	Client-Network (optional)	Client traffic for Connector & Storage Nodes
79	External-Network	External Public Network for all UCS Servers

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

1. Select **LAN** in the left pane in the Cisco UCS Manager GUI.
2. Select LAN > LAN Cloud > VLANs and right-click Create VLANs.
3. Enter Storage-Management 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 21 Create a VLAN



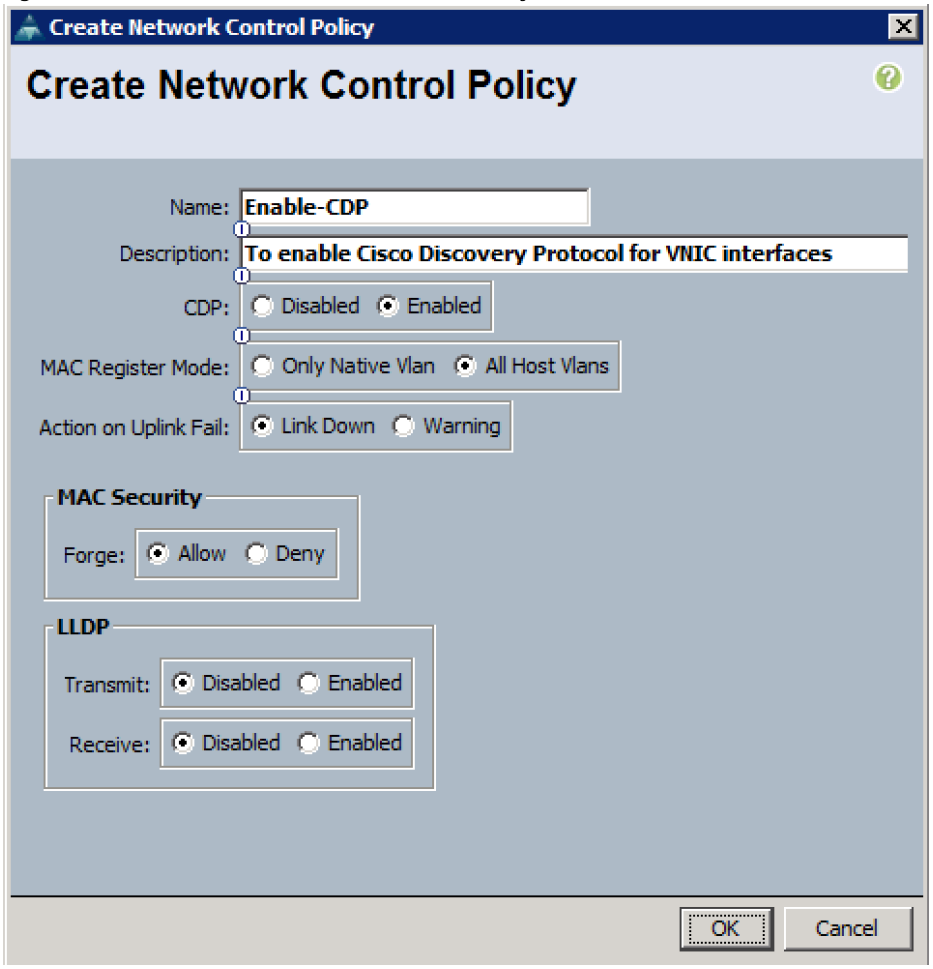
8. Repeat the steps for the rest of the VLANs Storage-Cluster, Client Network, and External-Network.

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 22 Create a Network Control Policy

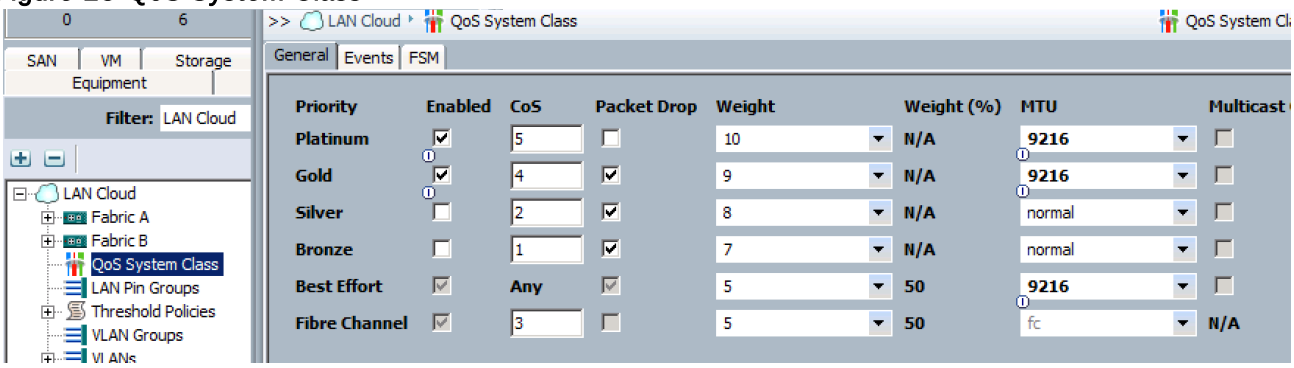


QoS System Class

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. Enable Priority Platinum & Gold and set **Weight** 10 & 9 respectively and **MTU** to 9216 and Best Effort MTU as 9216.
4. Set Fibre Channel Weight to None.
5. Click **Save Changes** and then **OK**.

Figure 23 QoS System Class

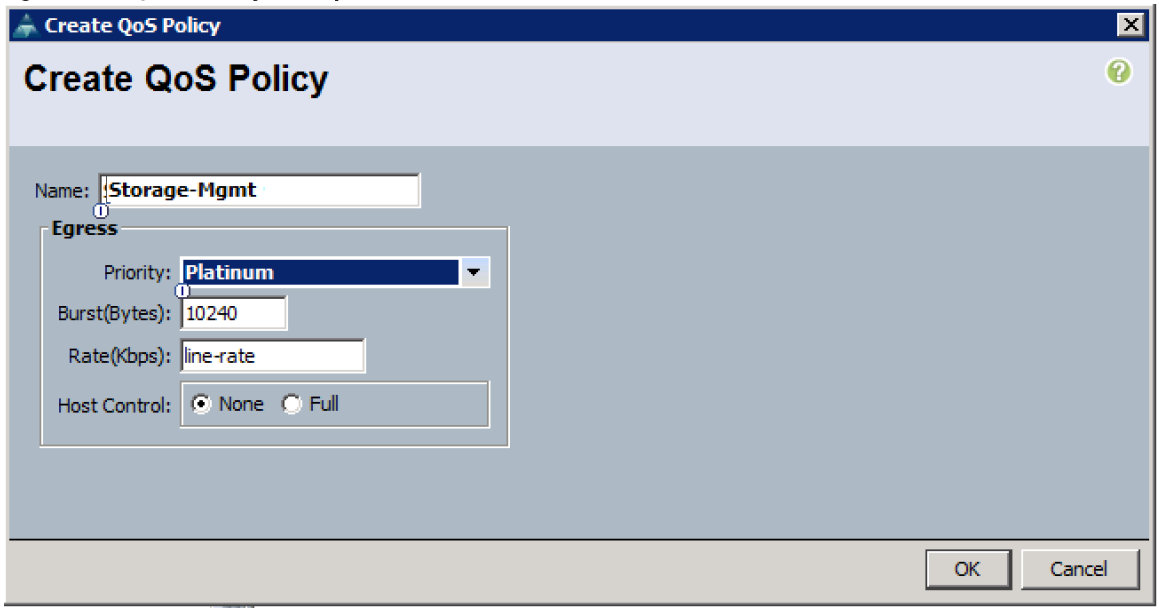


QoS Policy Setup

Based on the previous QoS System Class, setup a QoS Policy with the following configuration:

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 **Storage-Mgmt** in the Name field.
4. Set **Priority** as **platinum** and leave everything else unchanged.
5. Click **OK** and then **OK**.

Figure 24 QoS Policy Setup



6. Repeat the steps to create Qos Policy for Storage-Cluster and Set Priority as Gold.

vNIC Template Setup

Based on the previous section, creating VLANs, the next step is to create the appropriate vNIC templates. For Scality Storage we need to create four different vNICs, depending on the role of the server. Table 6 provides an overview of the configuration.

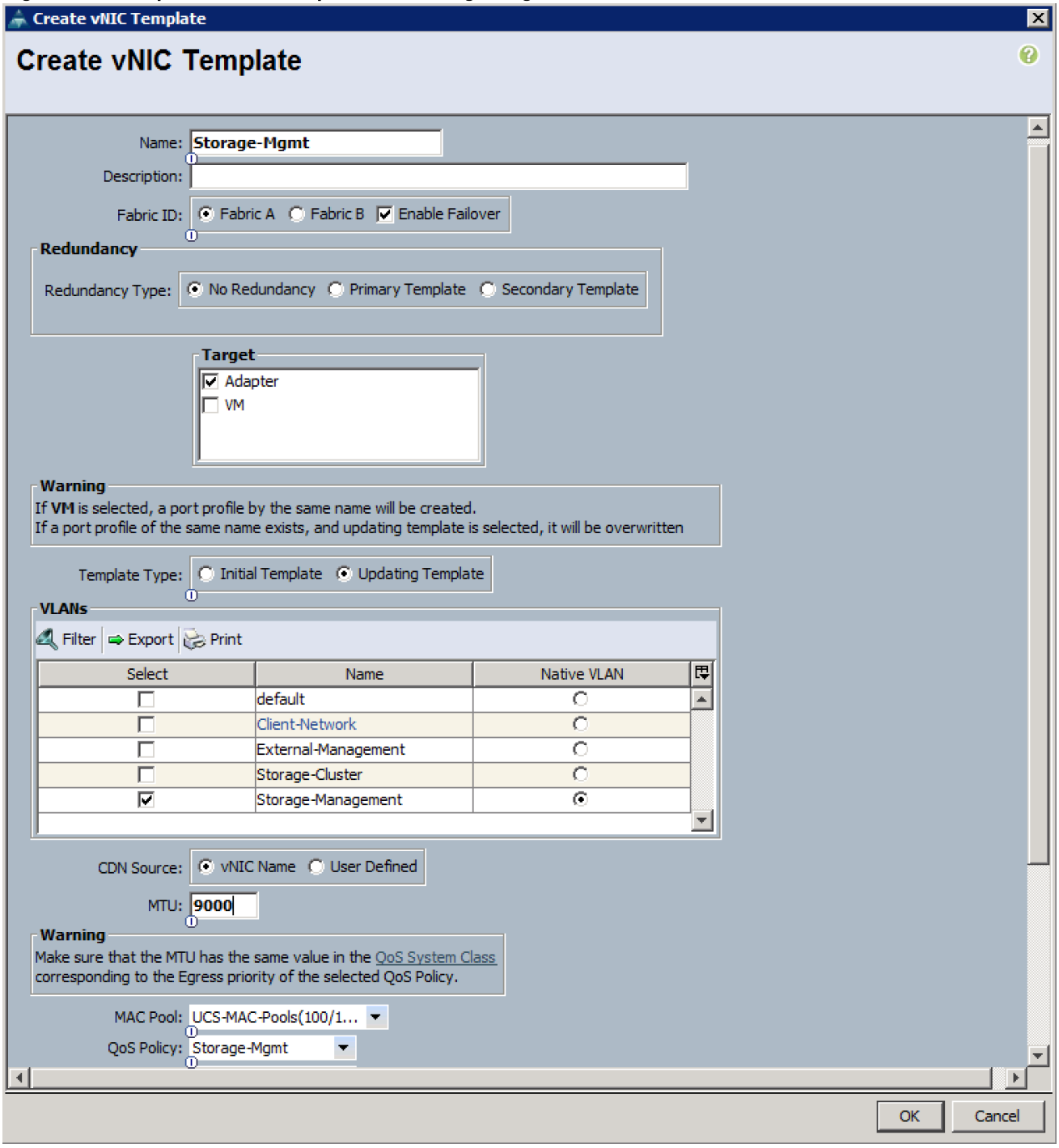
Table 6 vNIC Table

Name	vNIC Name	Fabric Interconnect	Failover	VLAN	MTU Size	MAC Pool	Network Control Policy
Storage-Mgmt	Storage-Mgmt	A	Yes	Storage-Mgmt – 10	9000	UCS-MAC-Pools	Enable-CDP
Storage-Cluster	Storage-Cluster	B	Yes	Storage-Cluster - 20	9000	UCS-MAC-Pools	Enable-CDP
Client-Network	Client-Network	A	Yes	Client-Network – 30	1500	UCS-MAC-Pools	Enable-CDP
External-Mgmt	External-Mgmt	A	Yes	External-Mgmt -79	1500	UCS-MAC-Pools	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 **Storage-Mgmt** 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-MAC-Pools** as MAC Pool.
8. Select **Storage-Mgmt** as QoS Policy.
9. Select Enable-CDP as Network Control Policy.
10. Click **OK** and then **OK**.

Figure 25 Setup the vNIC Template for Storage-Mgmt vNIC



11. Repeat the steps for the vNICs Storage-Cluster, Client-NIC and External-Mgmt. Make sure you select the correct Fabric ID, VLAN and MTU size according to Table 6 .

Ethernet Adapter Policy Setup

By default, Cisco UCS provides a set of Ethernet adapter policies. These policies include the recommended settings for each supported server operating system. Operating systems are sensitive to the settings in these policies.

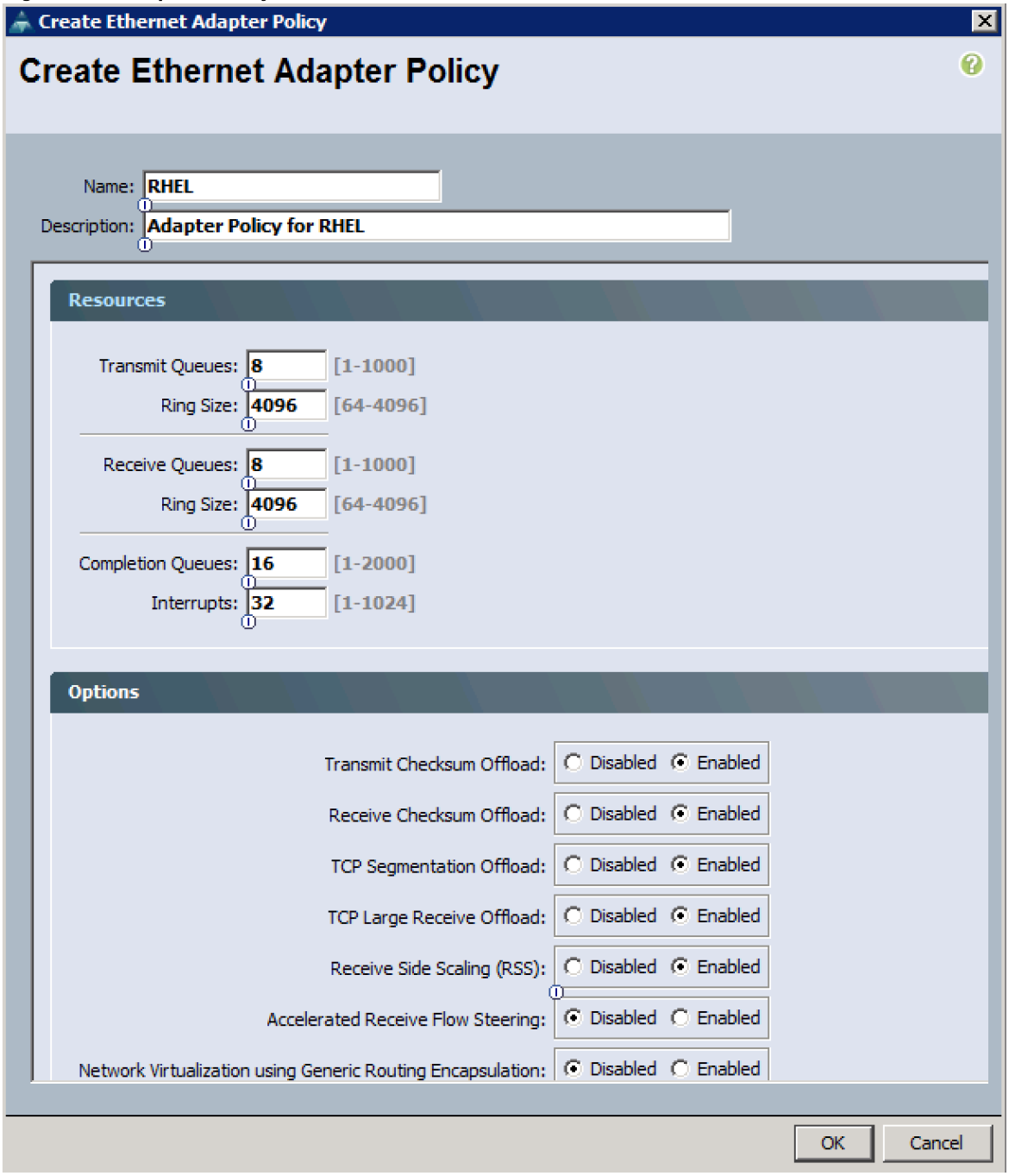
Note: Cisco UCS best practice is to enable Jumbo Frames MTU 9000 for any Storage facing Networks (Storage-Mgmt & Storage-Cluster). Enabling jumbo frames on specific interfaces, guarantees 39Gb/s bandwidth on the Cisco UCS fabric. For Jumbo Frames MTU9000, you can use default Ethernet Adapter Policy predefined as Linux.

If the customer deployment scenarios only supports only MTU1500, you can still modify the Ethernet Adapter policy resources Tx & Rx queues to guarantee 39Gb/s bandwidth.

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 26 Adapter Policy for RHEL



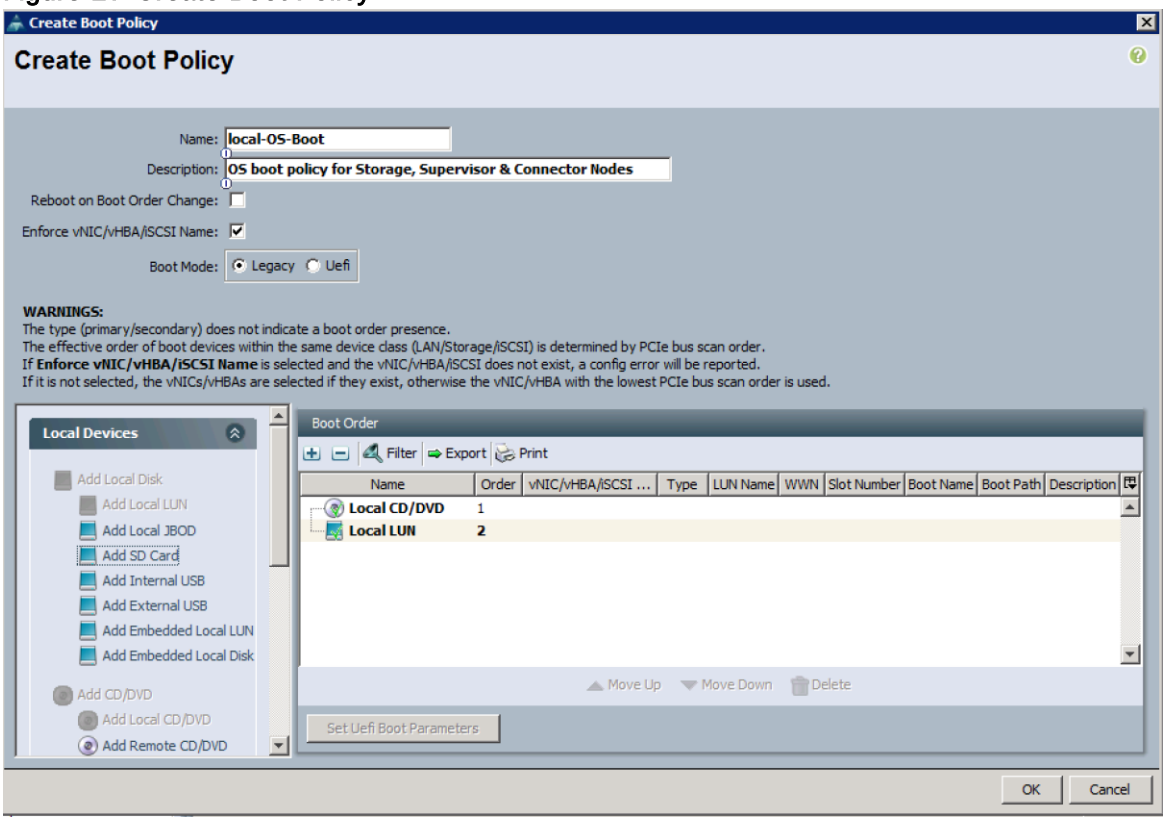
Boot Policy Setup

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 **Local-OS-Boot** in the **Name** field.
4. (Optional) Enter a description in the **Description** field.

Figure 27 Create Boot Policy



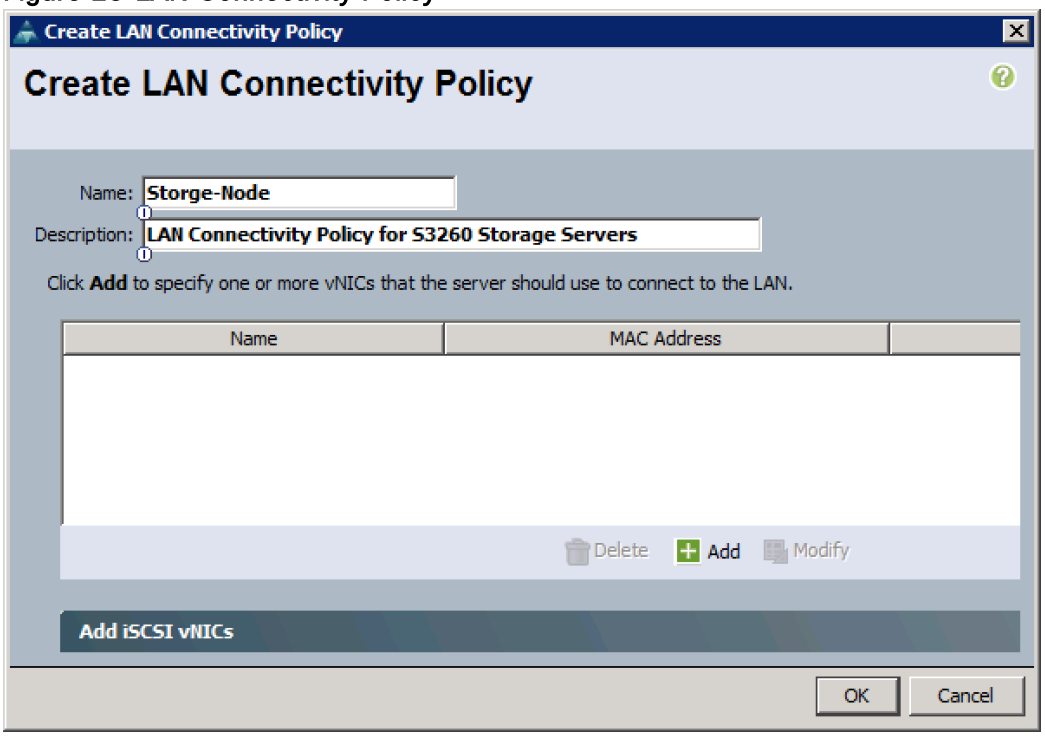
5. Click Local Devices > Add Local CD/DVD and click OK.
6. Click Local Devices > Add Local LUN and Set Type as **“Any”** and click OK.
7. Click **OK**.

Create LAN Connectivity Policy Setup

To create a LAN Connectivity Policy, complete the following steps:

1. Select the **LAN** tab in the left pane.
2. Go to Servers > Policies > root > LAN Connectivity Policies and right-click Create LAN Connectivity Policy for Storage Servers.
3. Type in **Storage-Node** in the **Name** field.
4. (Optional) Enter a description in the **Description** field.
5. Click **Add**.

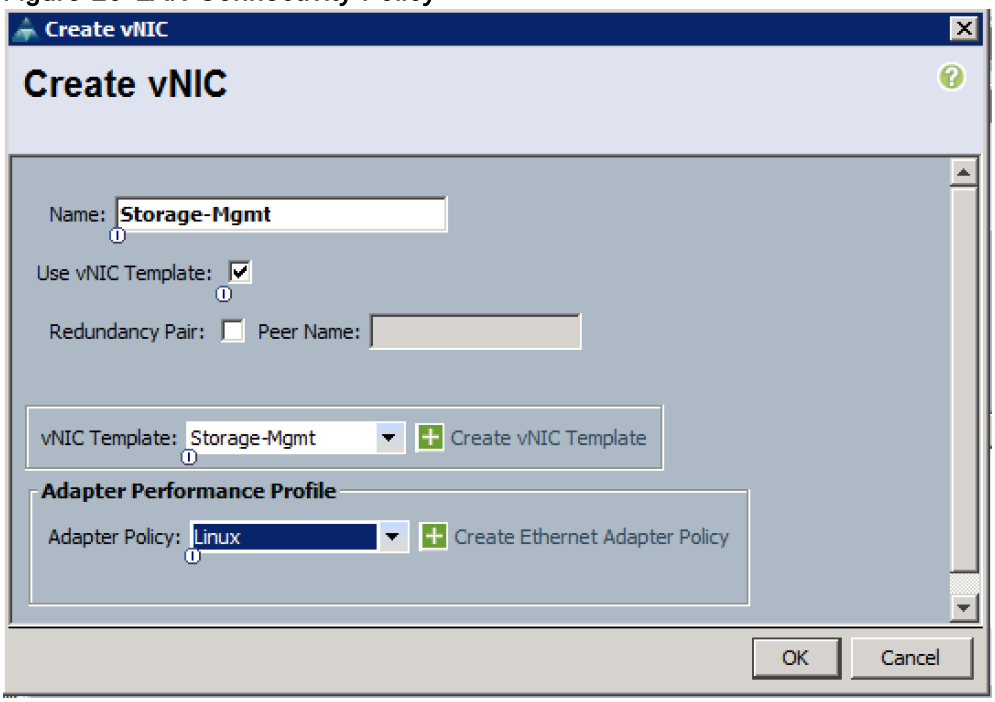
Figure 28 LAN Connectivity Policy



6. Type in Storage-Mgmt in the name field.
7. Click “Use vNIC Template.”
8. Select vNIC template for “Storage-Mgmt” from drop-down list.
9. If you are using Jumbo Frame MTU 9000, Select default Adapter Policy as Linux from the drop-down list.

Note: If you are using MTU 1500, Select Adapter Policy as **RHEL** created before from the drop-down list.

Figure 29 LAN Connectivity Policy

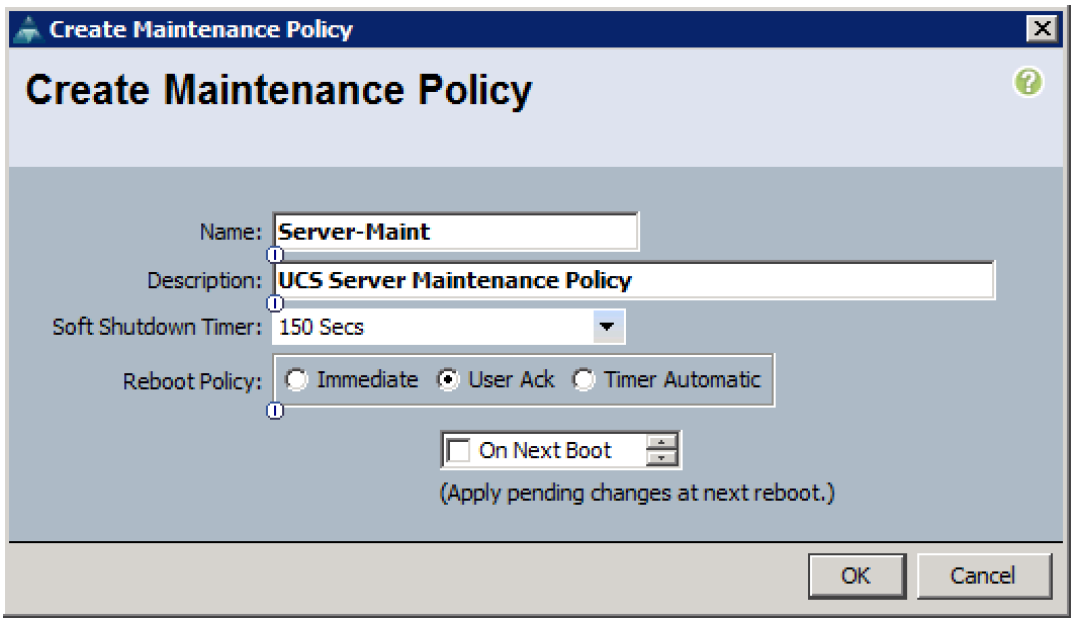


10. Repeat the vNIC creation steps for Storage-Cluster, Client-Network, and External-Network.

Create Maintenance Policy Setup

To setup a Maintenance Policy, complete the following steps:

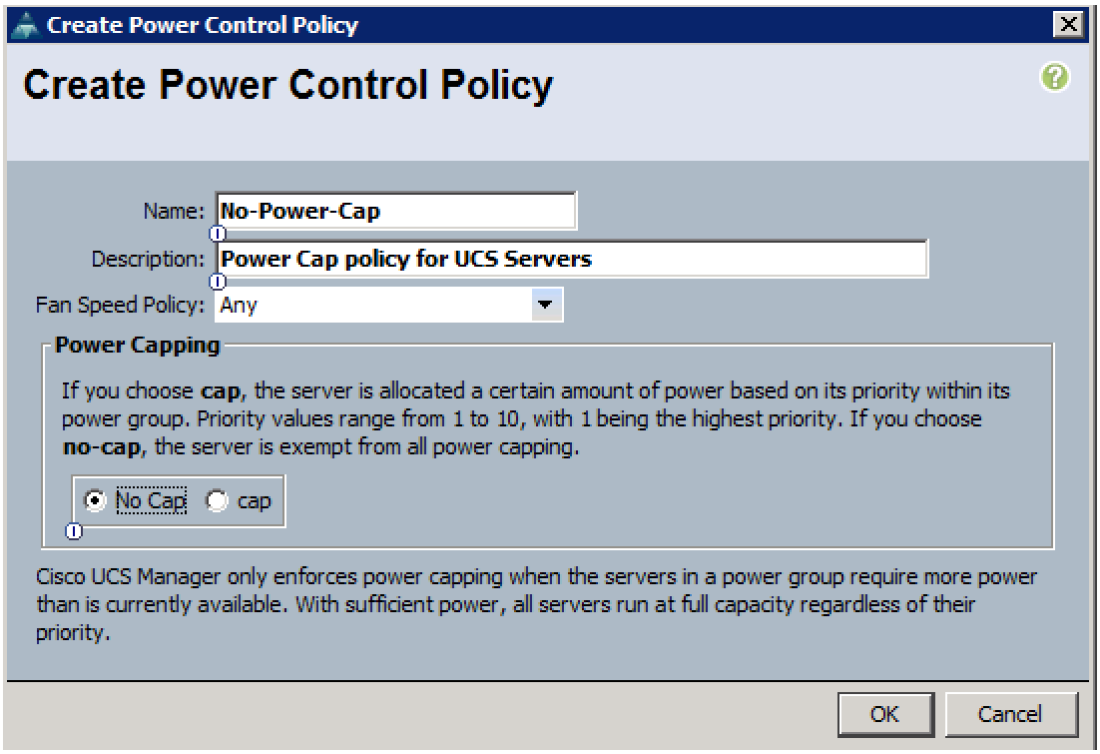
1. Select the **Servers** tab in the left pane.
2. Go to Servers > Policies > root > Maintenance Policies and right-click Create Maintenance Policy.
3. Type in a **Server-Maint** in the Name field.
4. (Optional) Enter a description in the **Description** field.
5. Click User Ack under Reboot Policy.
6. Click **OK** and then **OK**.
7. Create Maintenance Policy.



Create Power Control Policy Setup

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

8. Select the **Servers** tab in the left pane.
9. Go to Servers > Policies > root > Power Control Policies and right-click Create Power Control Policy.
10. Type in **No-Power-Cap** in the **Name** field.
11. (Optional) Enter a description in the **Description** field.
12. Click **No Cap** and click **OK**.
13. Create Power Control Policy.



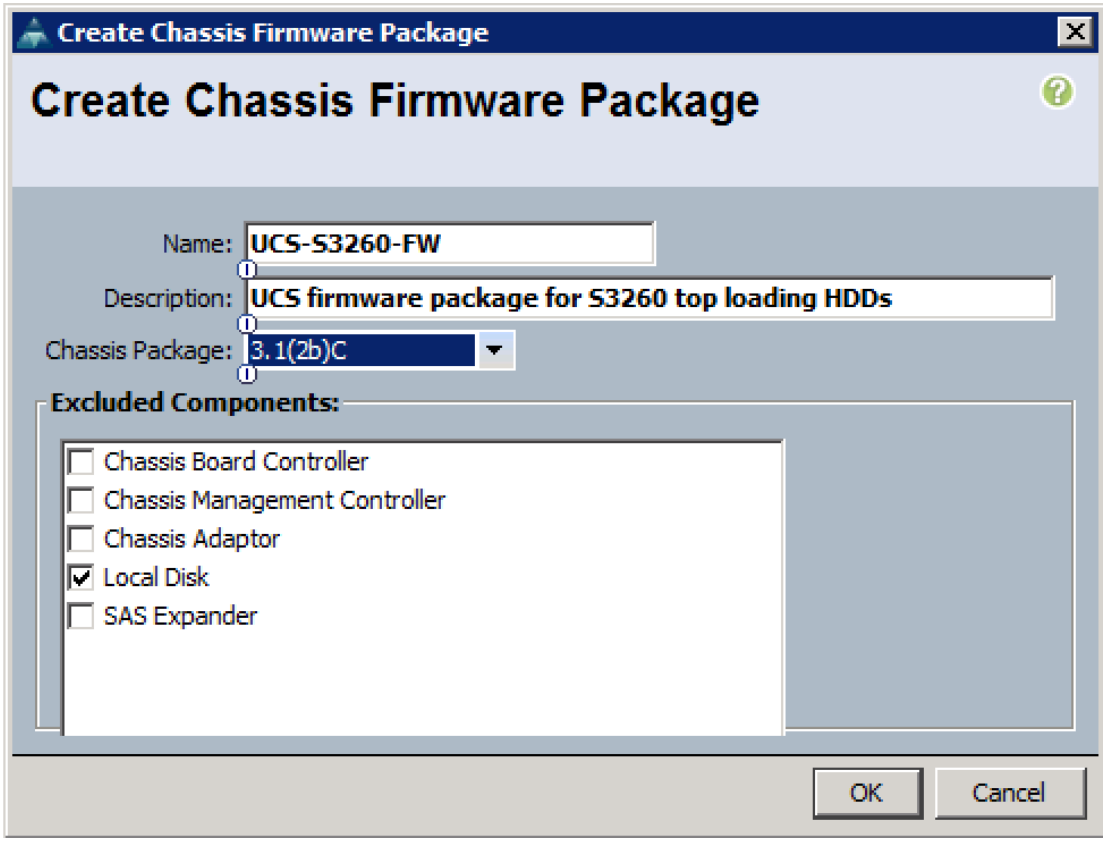
Creating Chassis Profile

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:

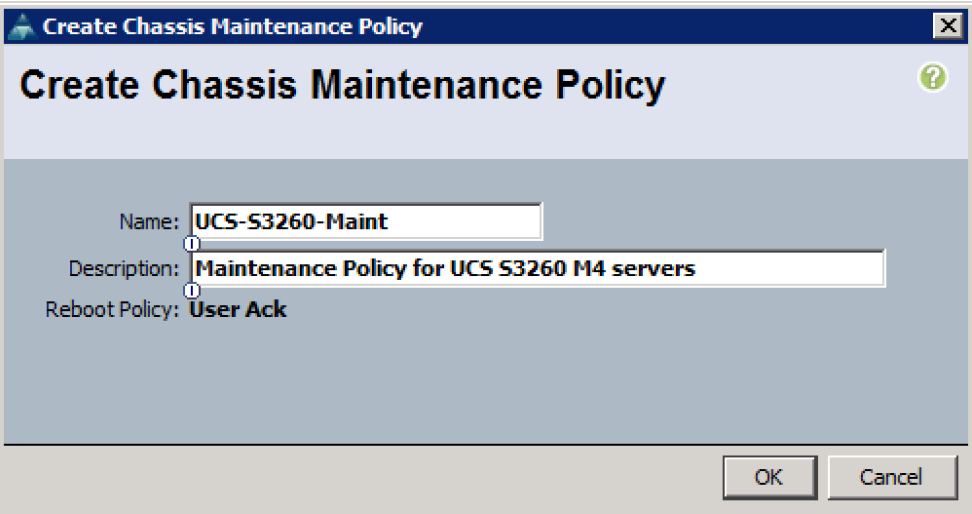
1. Select the **Chassis** tab in the left pane of the Cisco UCS Manager GUI.
2. Go to Chassis > Policies > root > Chassis Firmware Package and right-click Create Chassis Firmware Package.
3. Type in **UCS-S3260-FW** in the Name field.
4. (Optional) Enter a description in the **Description** field.
5.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



Create Chassis Maintenance Policy

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

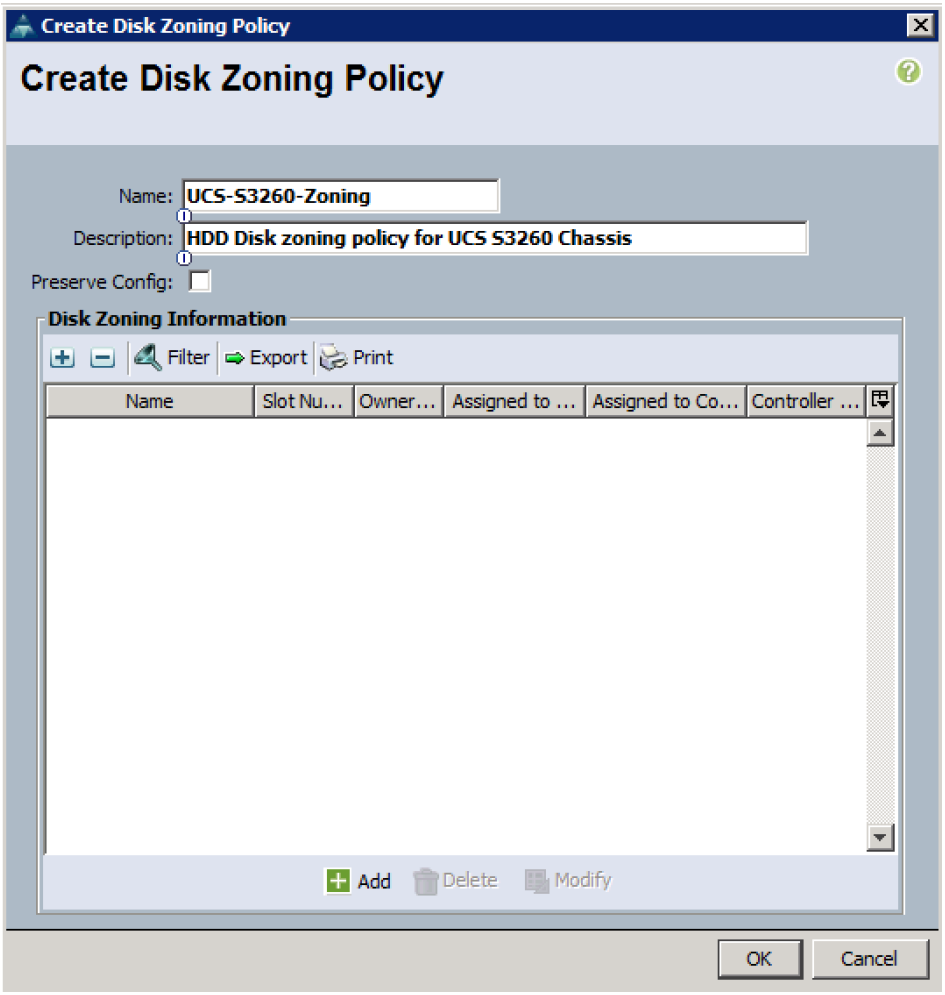
1. Select the **Chassis** tab in the left pane of the Cisco UCS Manager GUI.
2. Go to Chassis > Policies > root > Chassis Maintenance Policies and right-click Create Chassis Maintenance Policy.
3. Type in ***UCS-S3260-Main*** in the **Name** field.
4. (Optional) Enter a description in the **Description** field.
5. Click **OK** and then **OK**.
6. Create Chassis Maintenance Policy.



Create Disk Zoning Policy

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

7. Select the Chassis tab in the left pane of the Cisco UCS Manager GUI.
8. Go to Chassis > Policies > root > Disk Zoning Policies and right-click Create Disk Zoning Policy.
9. Type in UCS-S3260-Zoning in the Name field.
10. (Optional) Enter a description in the Description field.
11. Create Disk Zoning Policy.



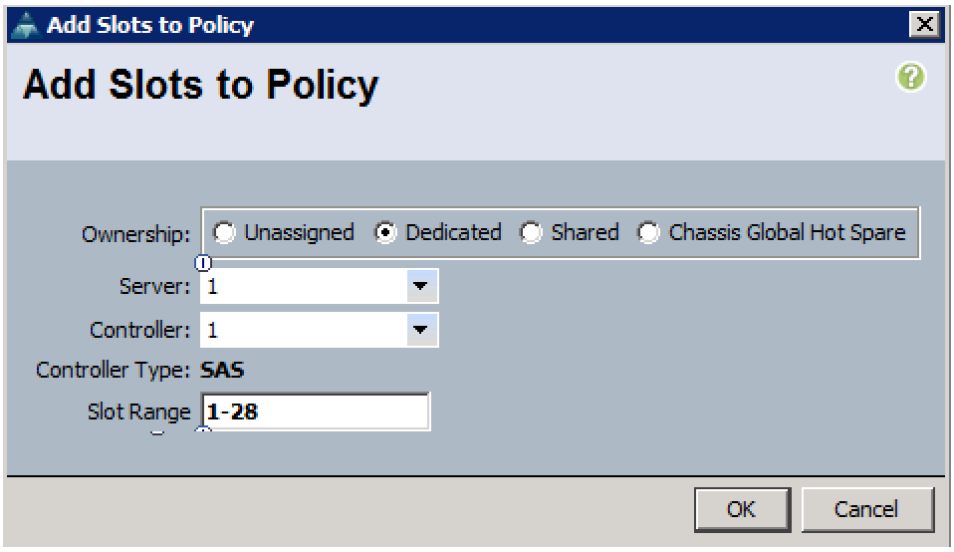
12. Click Add.

13. Select Dedicated under Ownership.

14. Select Server 1 and Select Controller 1.

15. Add Slot Range 1-28 for the top node of the Cisco UCS S3260 Storage Server and click OK.

16. Add Slots to Top Node of Cisco UCS S3260.



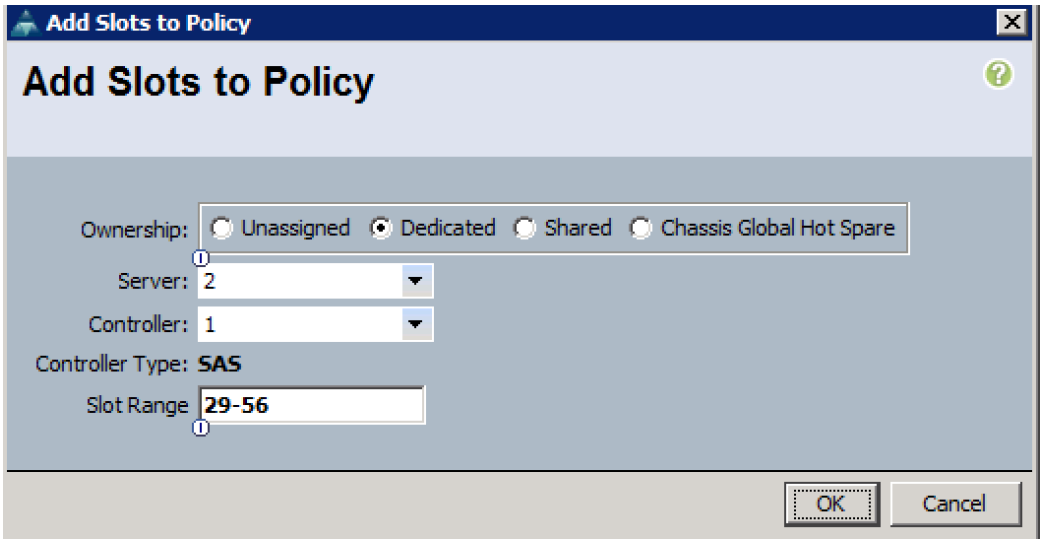
17. Click Add.

18. Select Dedicated under Ownership.

19. Select Server 2 and Select Controller 1.

20. Add Slot Range 29-56 for the bottom node of the Cisco UCS S3260 Storage Server and click OK.

21. Add Slots to Bottom Node of Cisco UCS S3260.



Create Chassis Profile Template

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

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

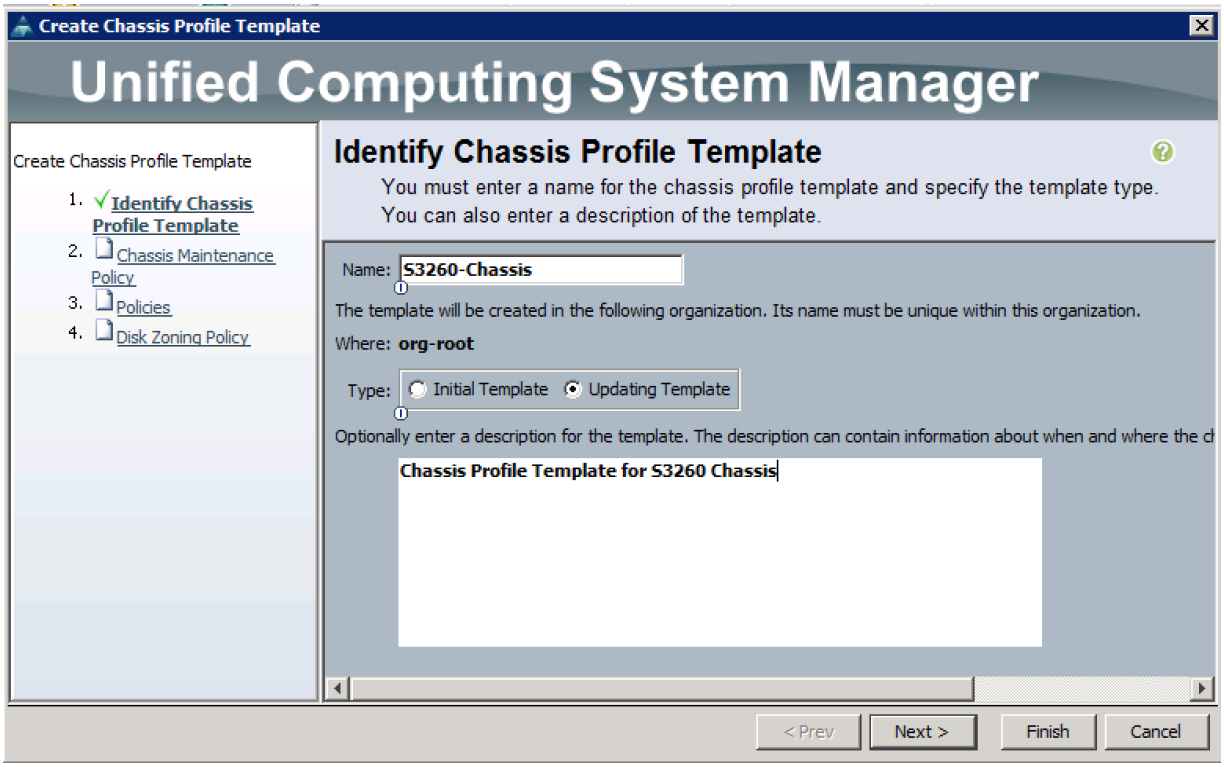
23. Go to Chassis > Chassis Profile Templates and right-click Create Chassis Profile Template.

24. Type in S3260-Chassis in the Name field.

25. Under Type, select Updating Template.

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

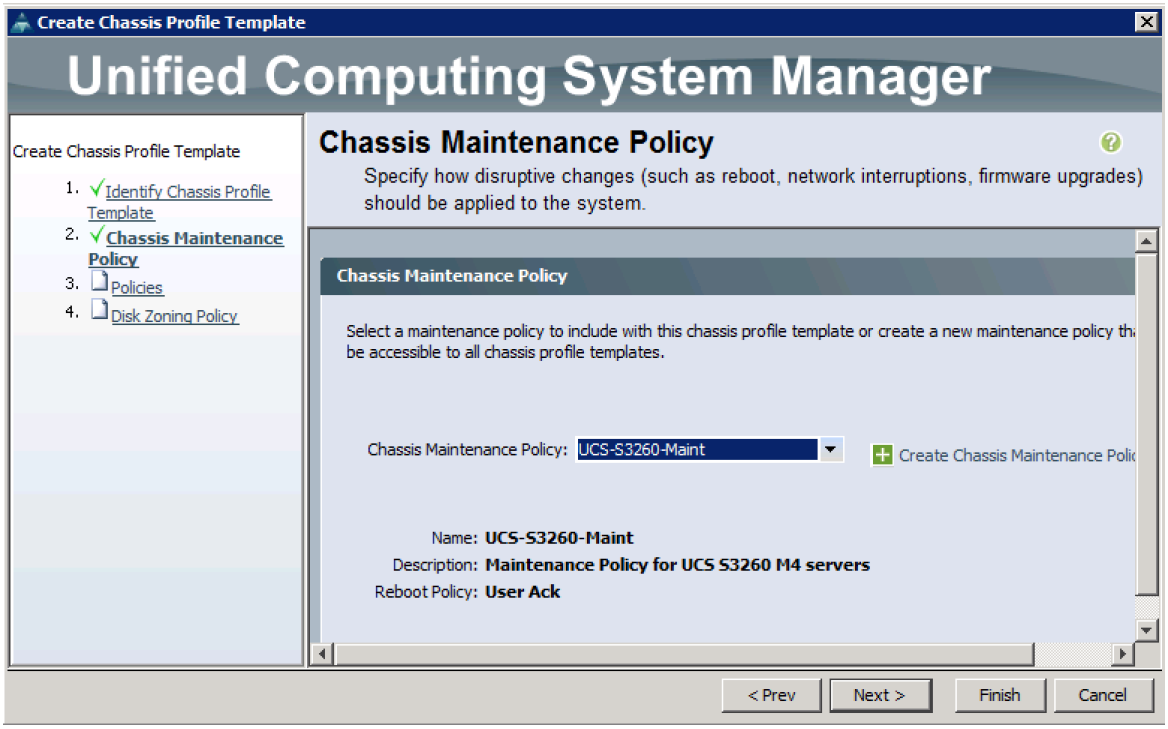
27. Create Chassis Profile Template.



28. Select **Next**.

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

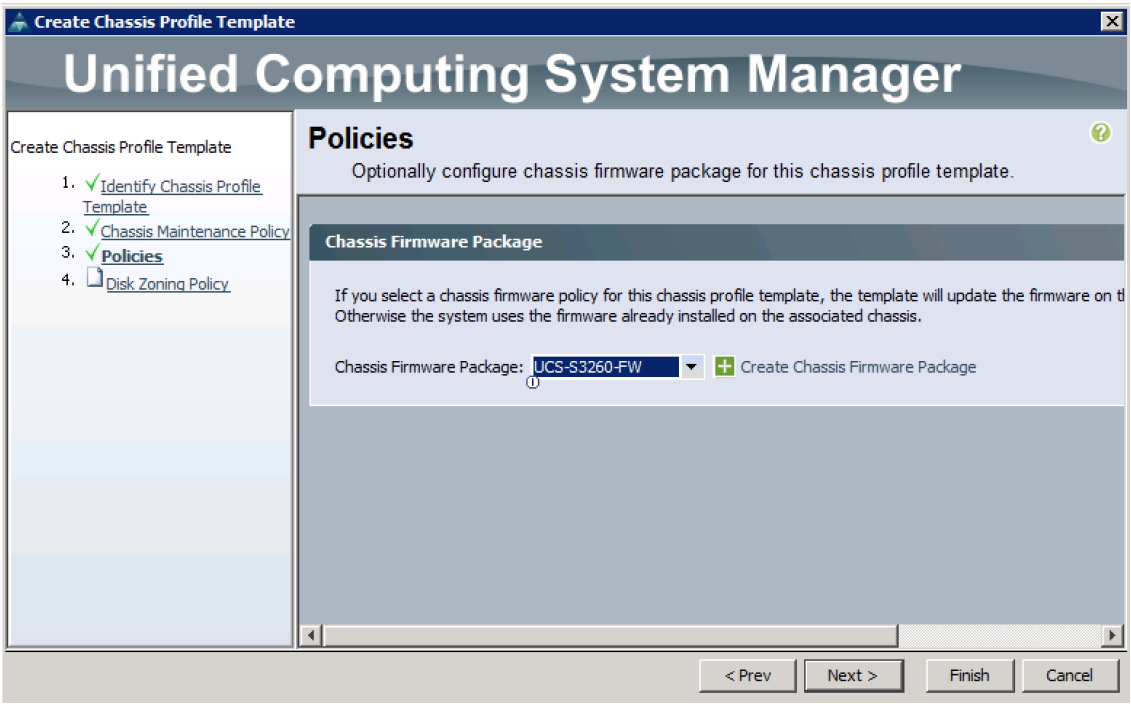
30. Chassis Profile Template – Chassis Maintenance Policy.



31. Select **`Next`**.

32. Select the + button and select under **`Chassis Firmware Package`** your previously created Chassis Firmware Package Policy.

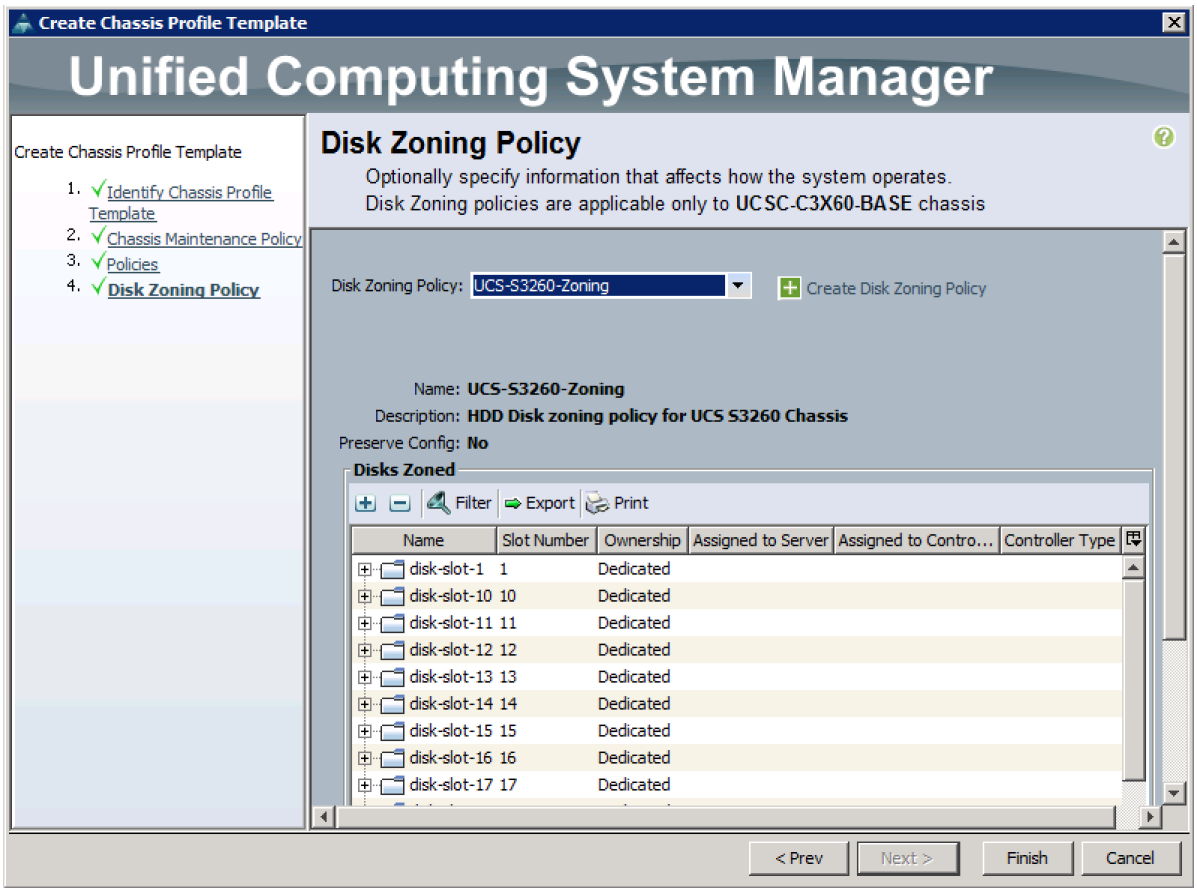
33. Chassis Profile Template – Chassis Firmware Package.



34. Select Next.

35. Under **Disk Zoning Policy** select your previously created Disk Zoning Policy.

36. Chassis Profile Template – Disk Zoning Policy



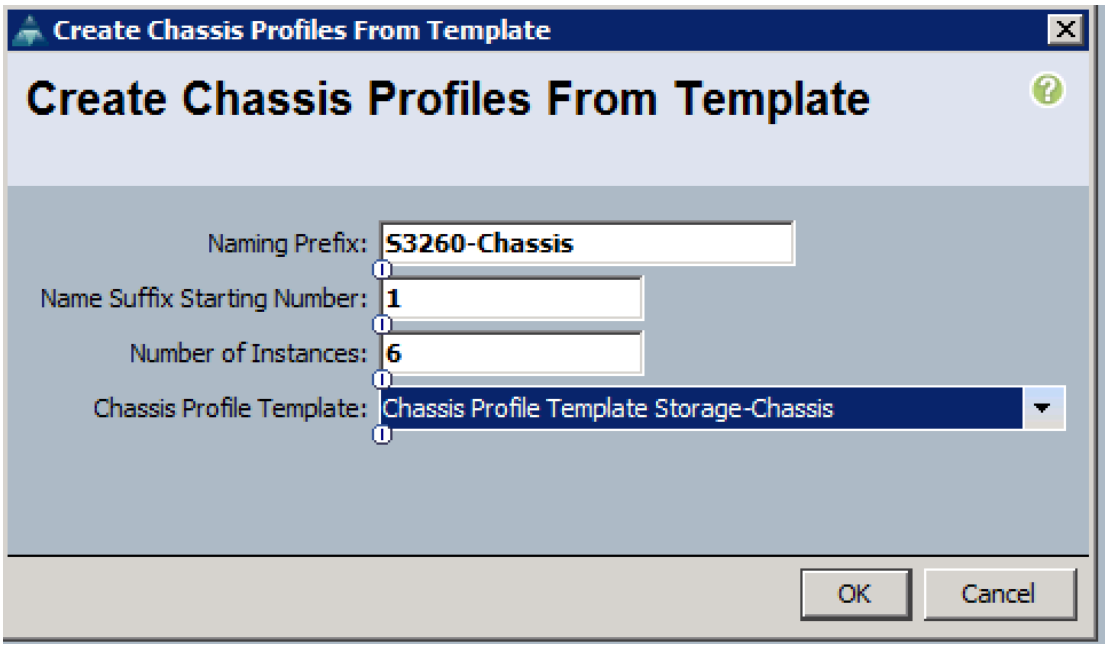
37. Click **Finish** and then click **OK**.

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-Chassis** in the **Name** field.
4. Leave the Name Suffix Starting Number untouched.
5. Enter **6** for the **Number of Instances** for all connected Cisco UCS S3260 Storage Server.

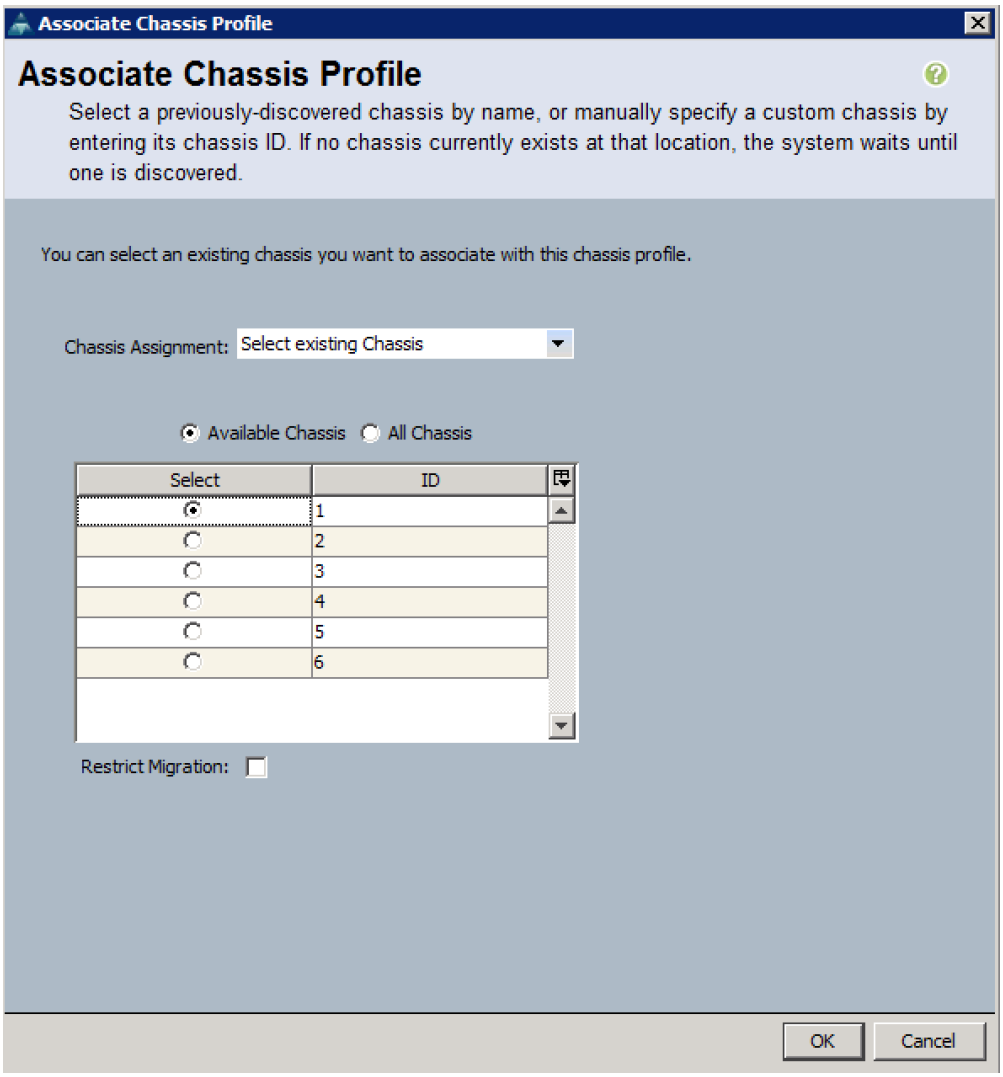
6. Choose your previously created **Chassis Profile Template**.
7. Click **OK** and then click **OK**.
8. Create Chassis Profiles from Template.



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-Chassis.
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 click OK again.
7. Repeat the steps for the other four Chassis Profiles by selecting the IDs 2 – 6.
8. Associate Chassis Profile.

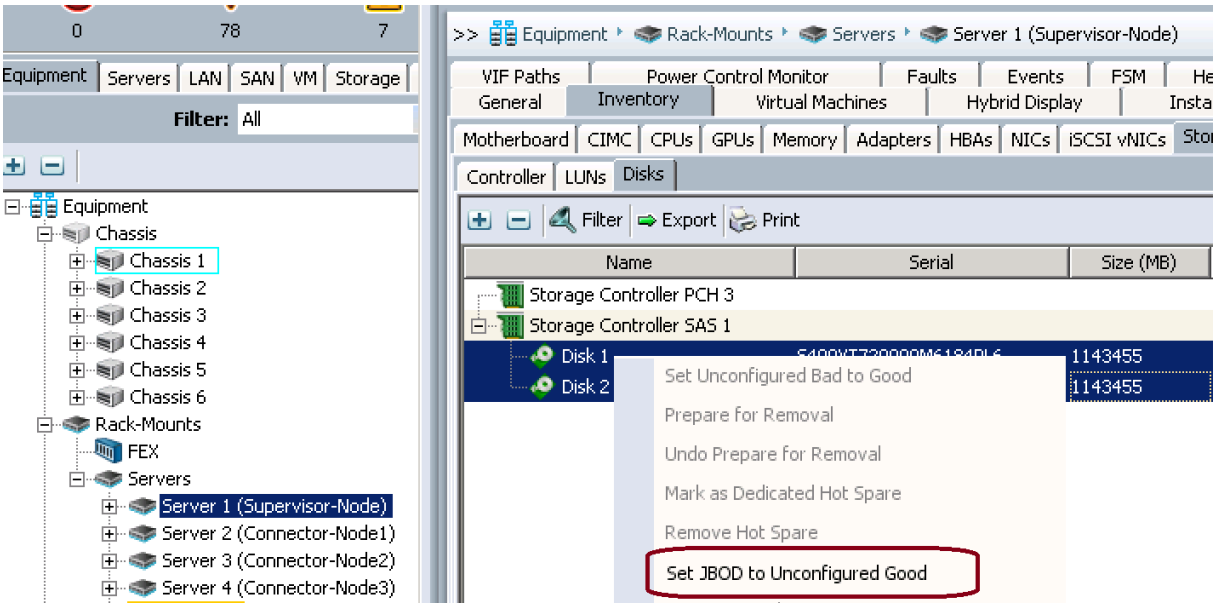


Creating Storage Profiles

Setting Disks for Cisco UCS C220 M4 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:

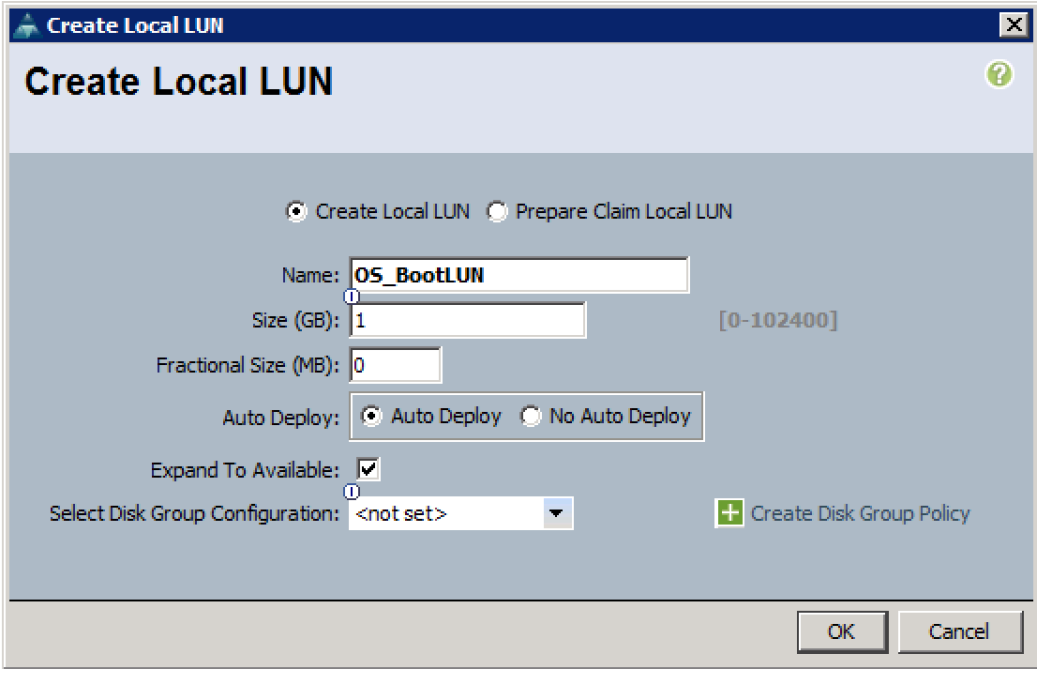
1. Select the **Equipment** tab in the left pane of the Cisco UCS Manager GUI.
2. Go to Equipment > Rack-Mounts > Servers > Server 1 > Disks.
3. Select both disks and right-click **Set JBOD to Unconfigured-Good**.
4. Repeat the steps for Server 2-4.
5. Set Disks for C220 M4 Servers to Unconfigured-Good.



Create Storage Profile for Cisco UCS S3260 Storage Server

To create the Storage Profile for Boot LUNs 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-OS-Node1** in the **Name** field.
4. (Optional) Enter a description in the **Description** field.
5. Click **Add**.
6. Type in **OS-BootLUN** 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
    f. Click Create Disk Group Policy
    g. Create Local LUN



h. Type in **RAID1-DG** in the Name field.

i. (Optional) Enter a description in the **Description** field.

j. RAID Level = RAID 1 Mirrored.

k. Select Disk Group Configuration (Manual).

l. Click **Add**.

m. Type in **201** for **Slot Number**.

n. Click **OK** and then again **Add**.

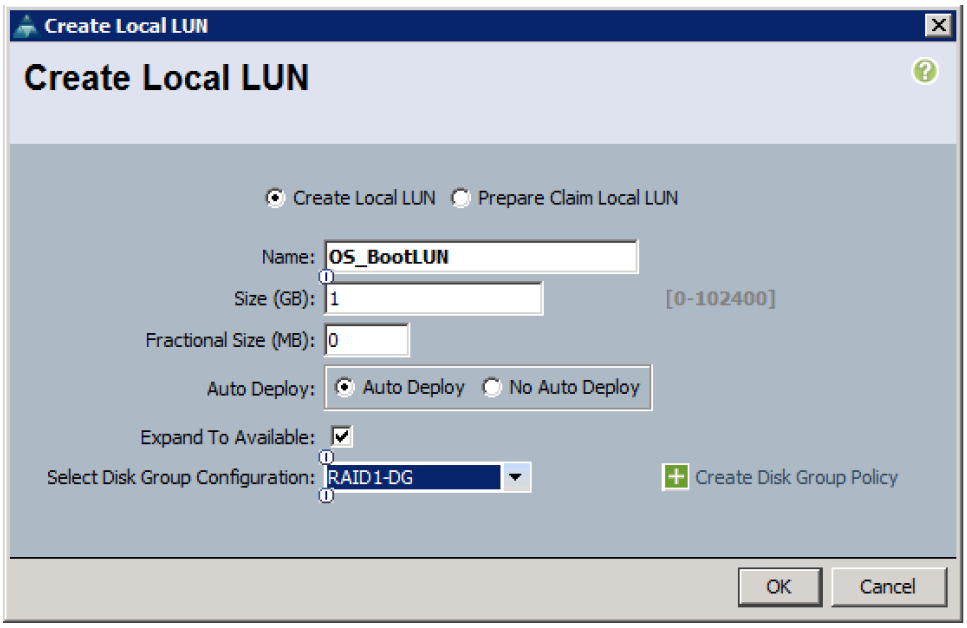
o. Type in **202** for **Slot Number**.

p. Leave everything else untouched.

q. Click **OK** and then **OK**.

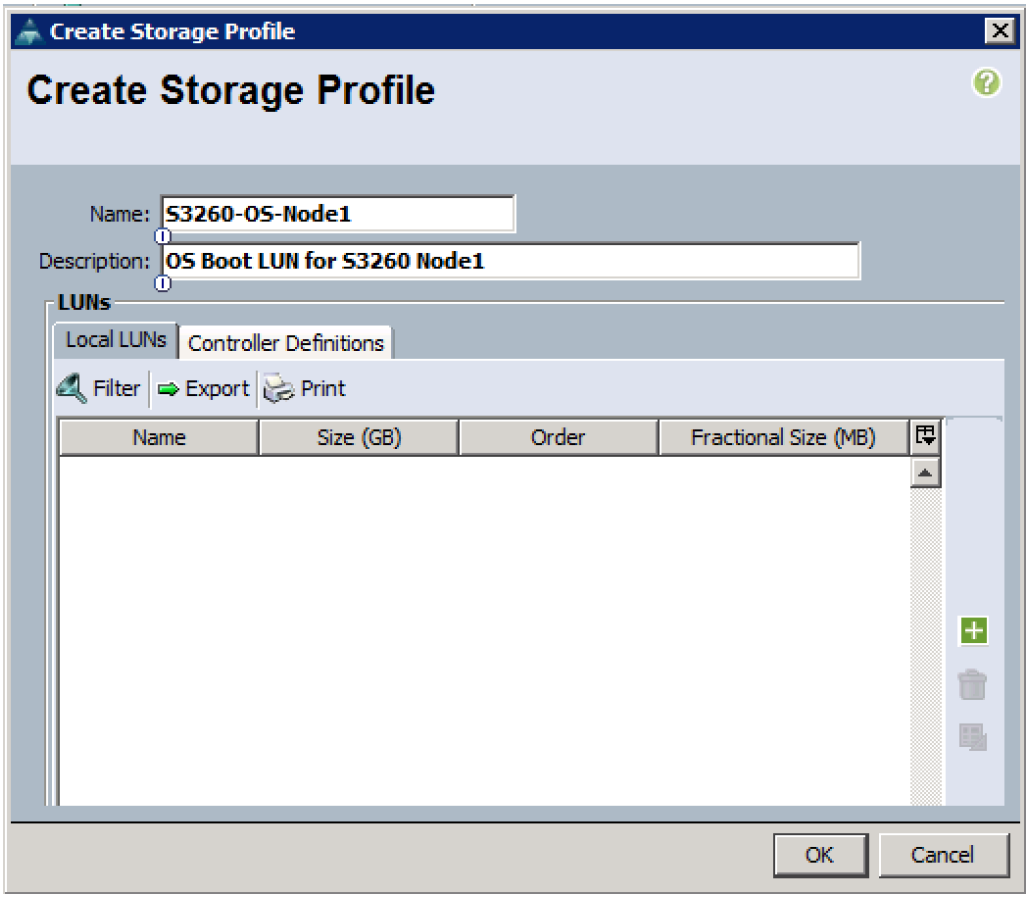
r. Select your previously created Disk Group Policy for the Boot SSDs by selecting the radio button under **Select Disk Group Configuration**.

s. Select Disk Group Configuration.



t. Click **OK**, click **OK** again, and then click **OK**.

u. Storage Profile for the top node of Cisco UCS S3260 Storage Server.

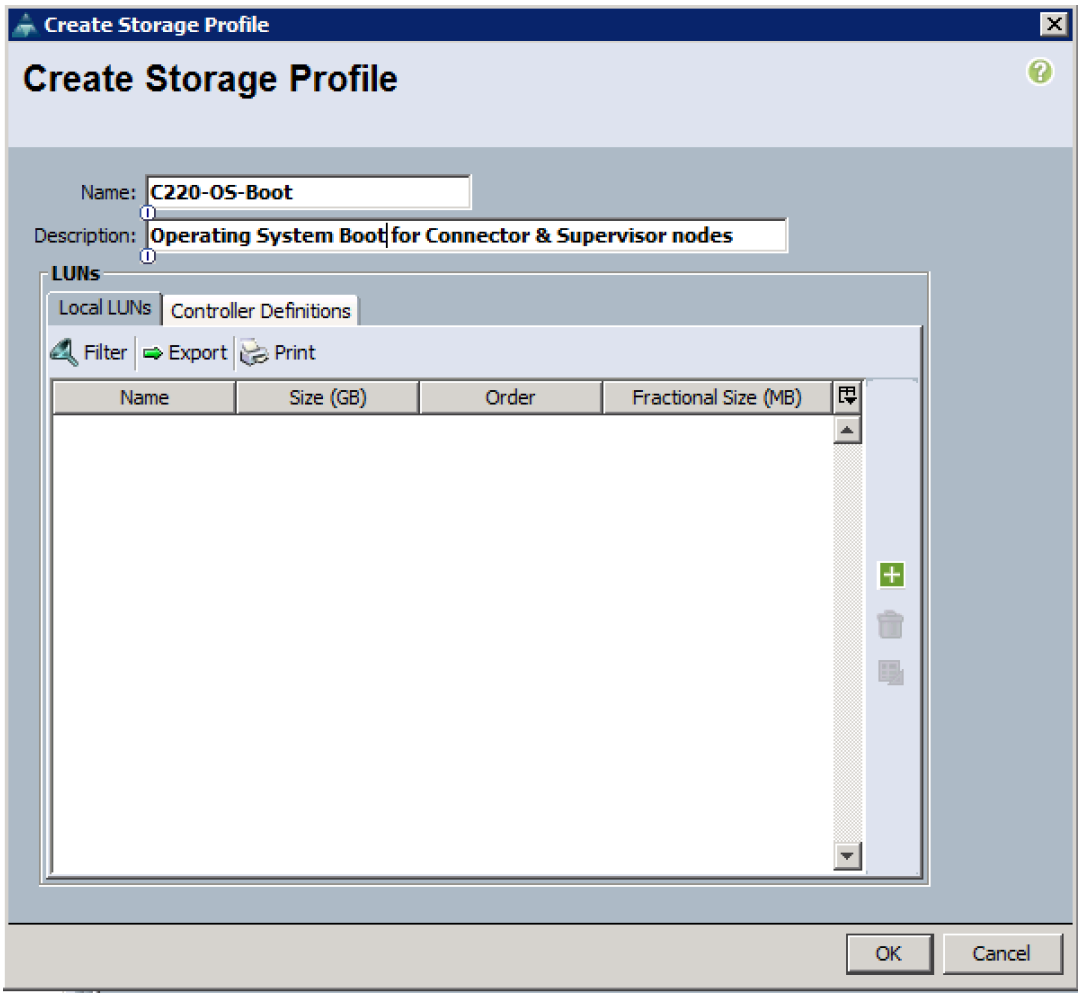


8. To create the Storage Profile for the OS boot LUN for the bottom S3260 Node2 of the Cisco UCS S3260 Storage Server, repeat the same steps for Disk slot 203 and 204.

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

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-OS-Boot in the Name field.
4. (Optional) Enter a description in the Description field.
5. Click Add.
6. Create Storage Profile for Cisco UCS C220 M4S.



7. Type in Boot in the Name field.

8. Configure as follow:

    a. Create Local LUN.

    b. Size (GB) = 1

    c. Fractional Size (MB) = 0

    d. Select Expand To Available.

    e. Auto Deploy.

    f. Click Create Disk Group Policy.

    g. Type in **RAID1-DG-C220** in the **Name** field.

    h. (Optional) Enter a description in the **Description** field.

    i. RAID Level = RAID 1 Mirrored.

    j. Select Disk Group Configuration (Manual).

    k. Click **Add**.

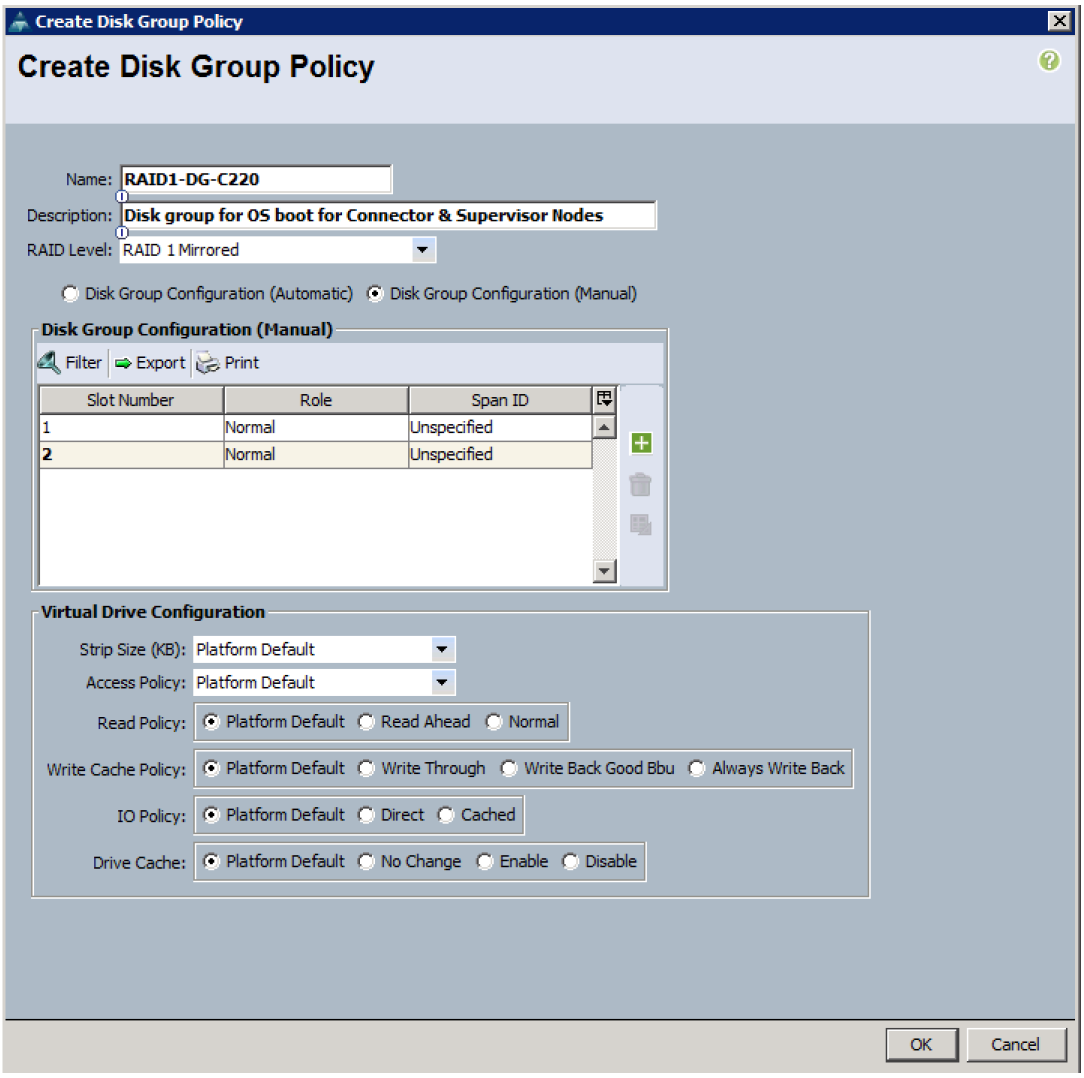
    l. Type in **1** for **Slot Number**.

    m. Click **OK** and then again **Add**.

    n. Type in **2** for **Slot Number**.

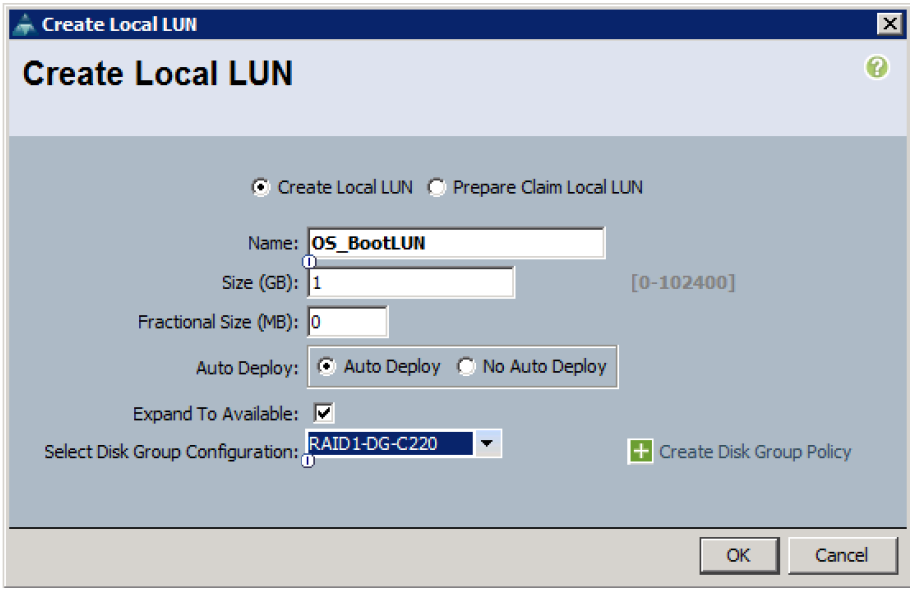
    o. Leave everything else untouched. Click **OK** and then **OK**.

    p. Create Disk Group Policy for C220 M4S.



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

10. Create Disk Group Configuration for C220 M4S.



11. Click **OK** and then click **OK** and click **OK** again.

Creating a Service Profile Template

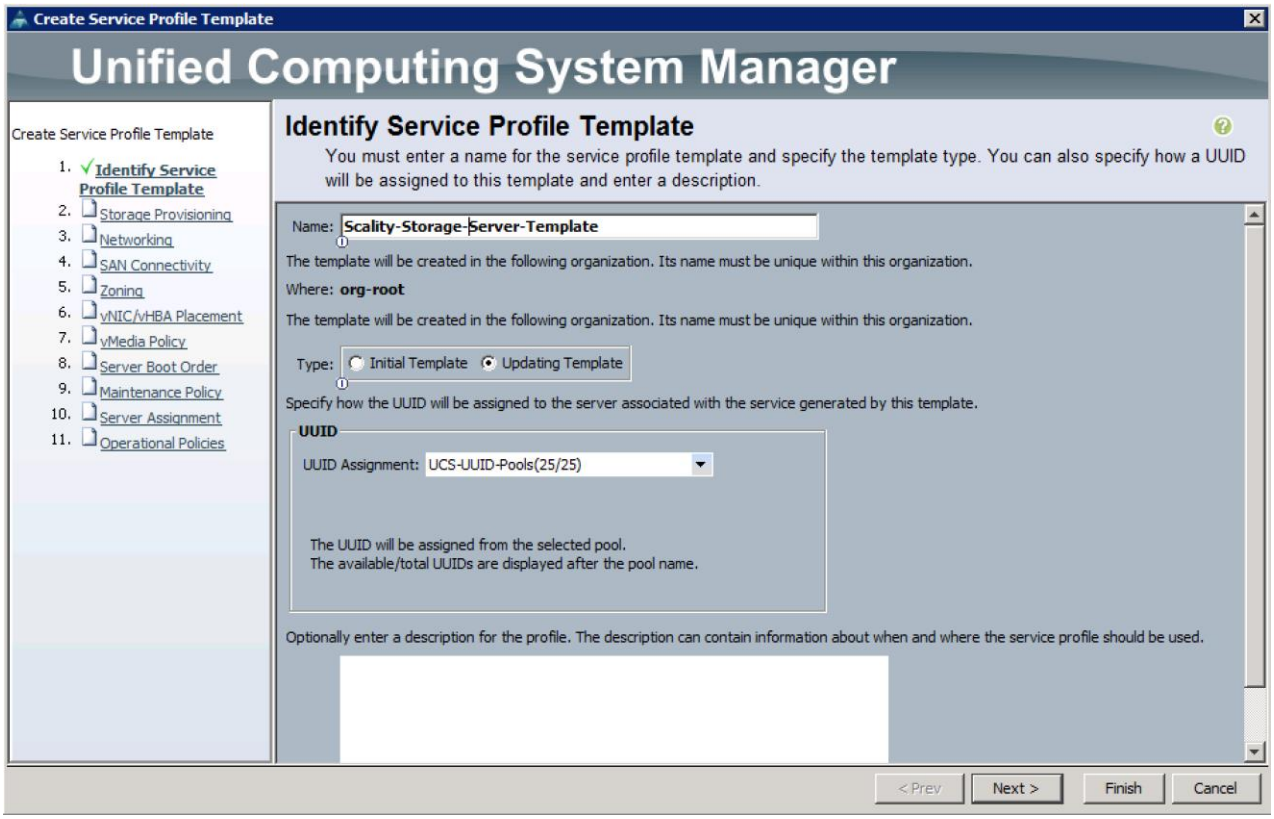
Create Service Profile Template for Cisco UCS S3260 Storage Server Top and Bottom Node

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 Scality-Storage-Server-Template 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.
4. Identify Service Profile Template.



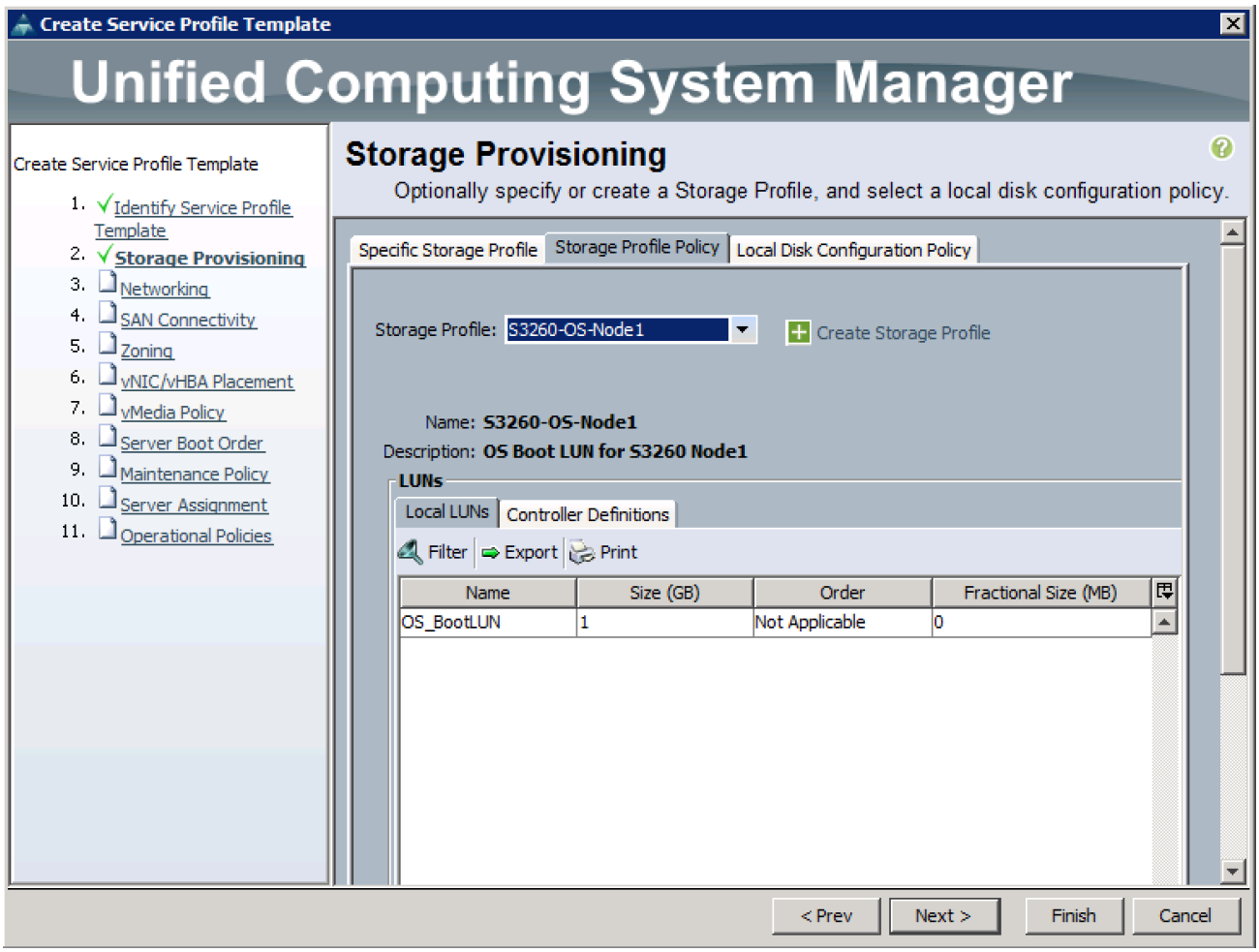
5. Click **Next**.

Storage Provisioning

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

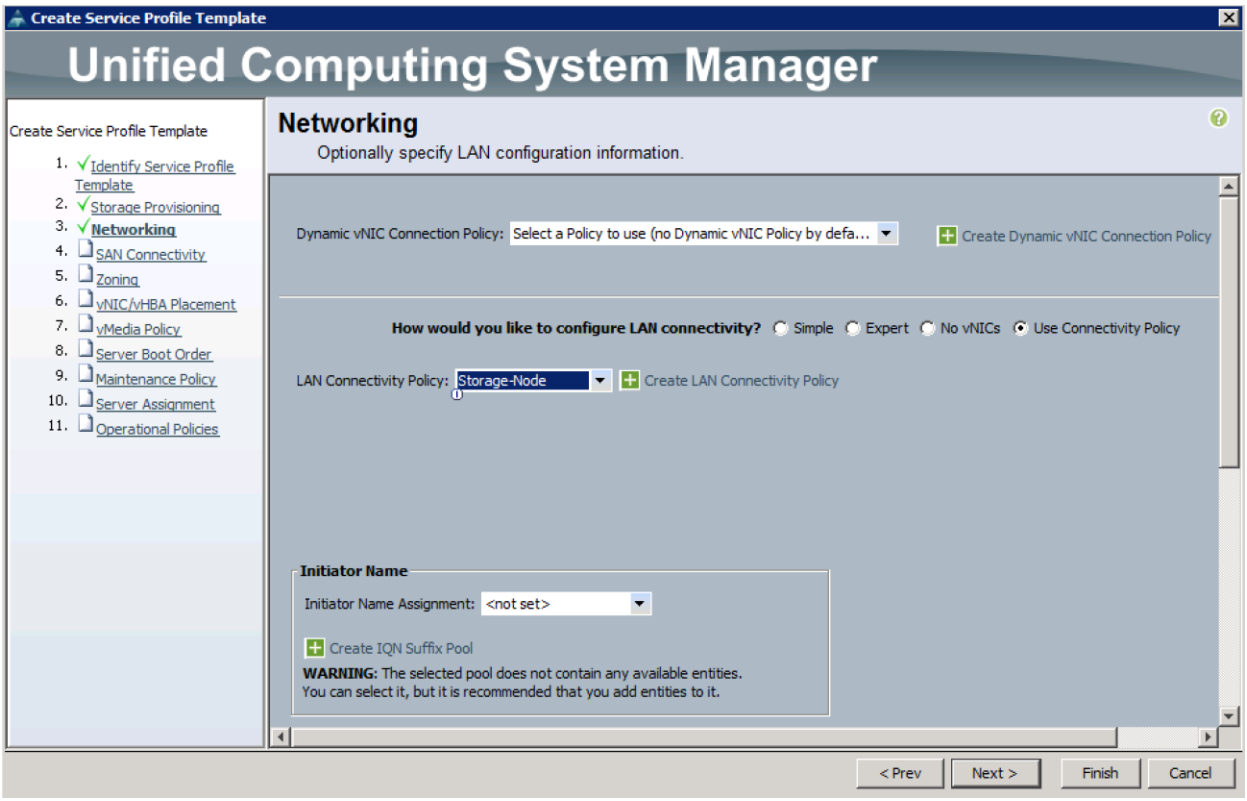
2. Click Next.

3. Storage Provisioning.



Networking

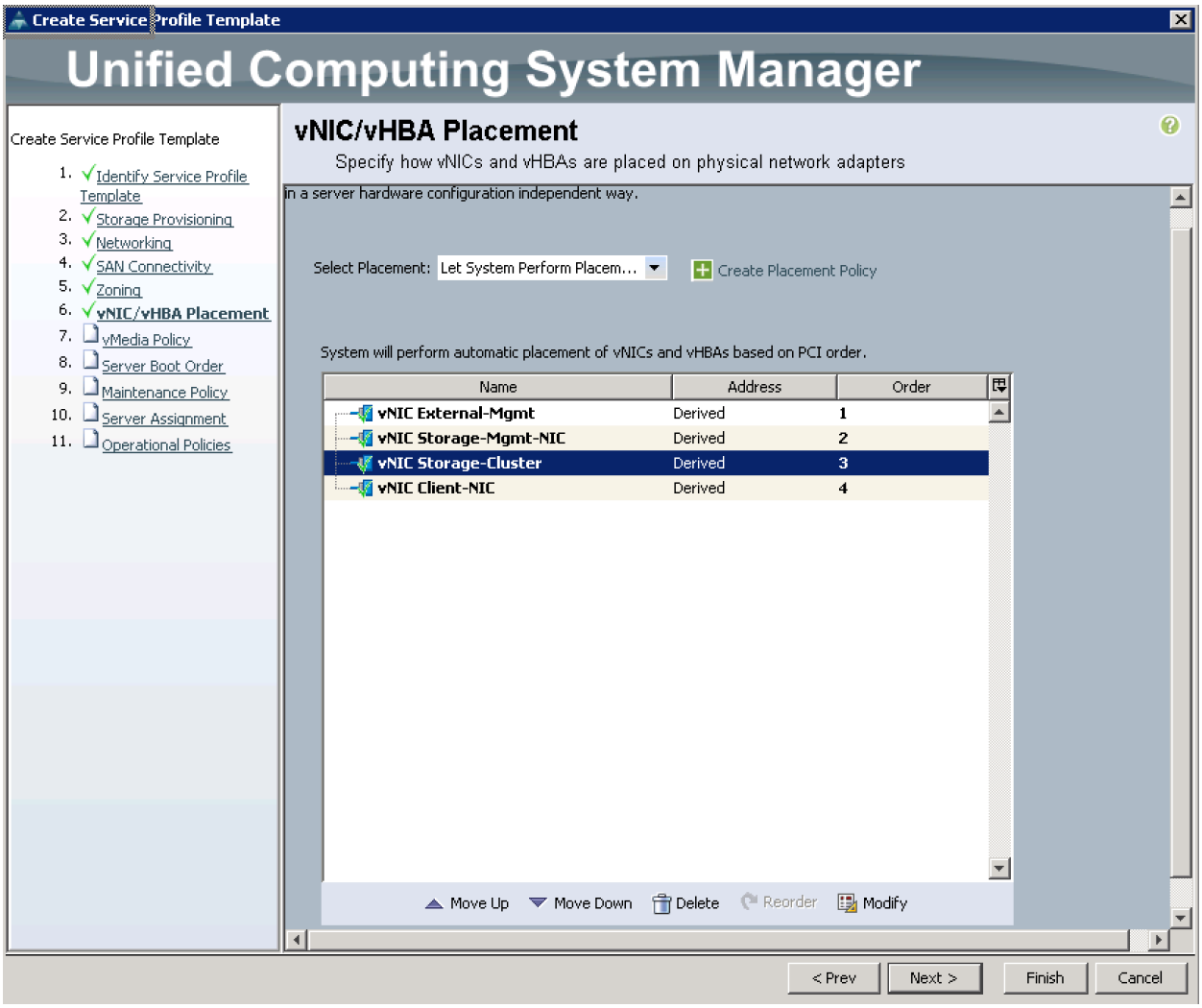
1. Keep the Dynamic vNIC Connection Policy field at the default.
2. Select LAN connectivity to Use Connectivity Policy created before.
3. From LAN Connectivity drop-down list, select “Storage-Node” created before and click Next.



4. Click **Next** to continue with SAN Connectivity.
5. Select No vHBA for How would you like to configure SAN Connectivity?
6. Click **Next** to continue with Zoning.
7. Click **Next**.

vNIC/vHBA Placement

1. Select **Let system Perform placement** form the drop-down menu.
2. Under PCI order section, Sort all the vNICs.
3. Make sure the vNICs order listed as External-Mgmt > 1, then followed by Storage-Mgmt > 2, Storage-Cluster > 3 and Client-Network > 4.



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

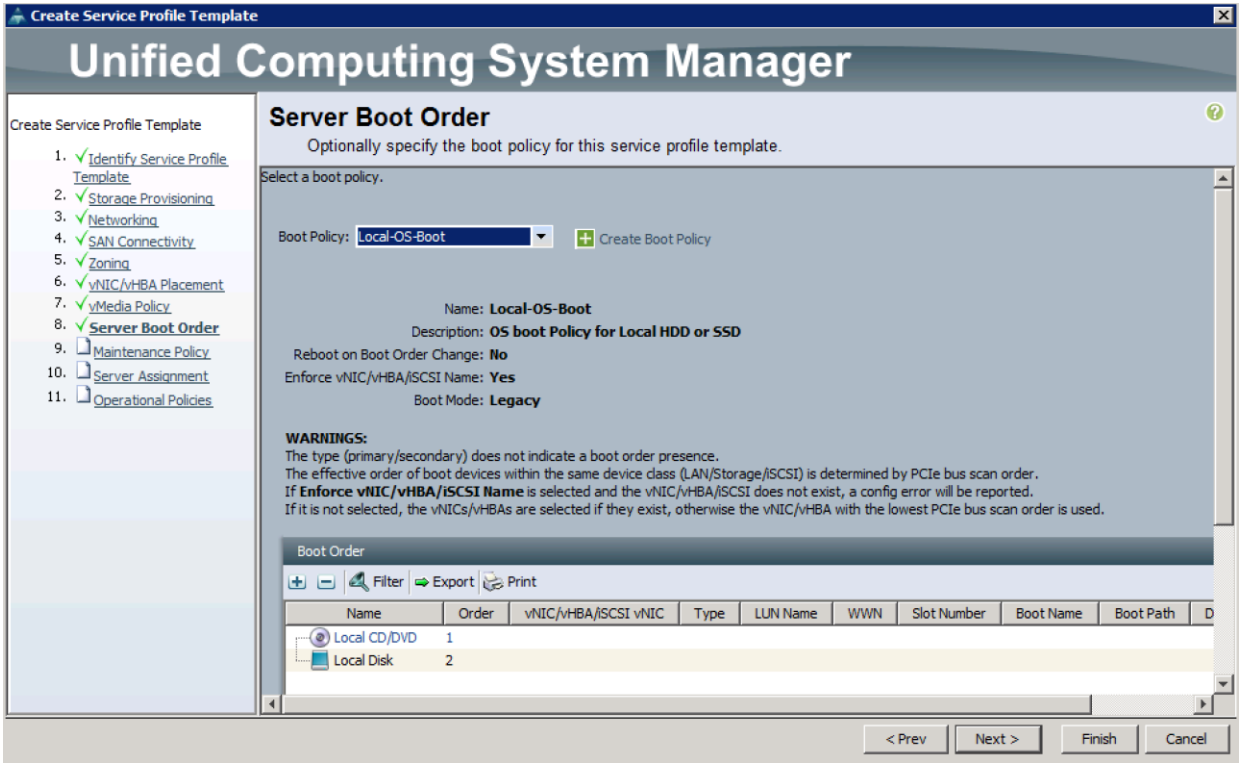
5. Click **Next**.

Server Boot Order

1. Select the Boot Policy “local-OS-Boot” you created before under Boot Policy.

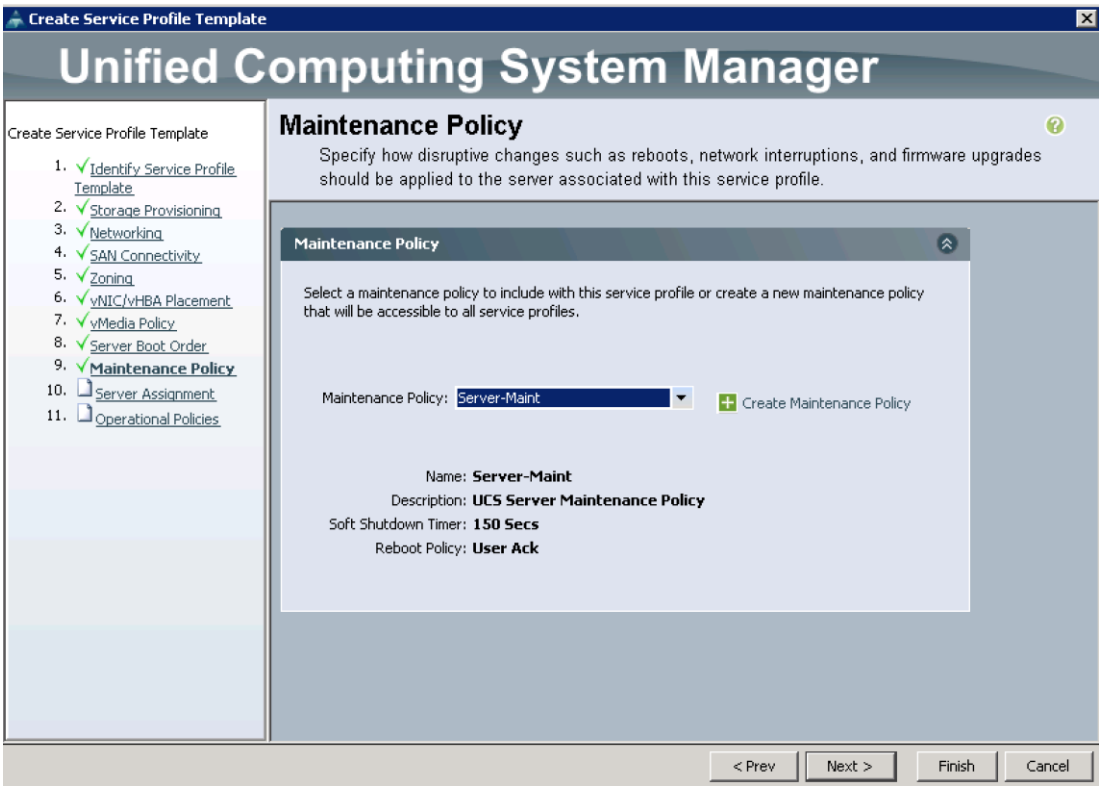
2. Server Boot Order.

3. Click Next.



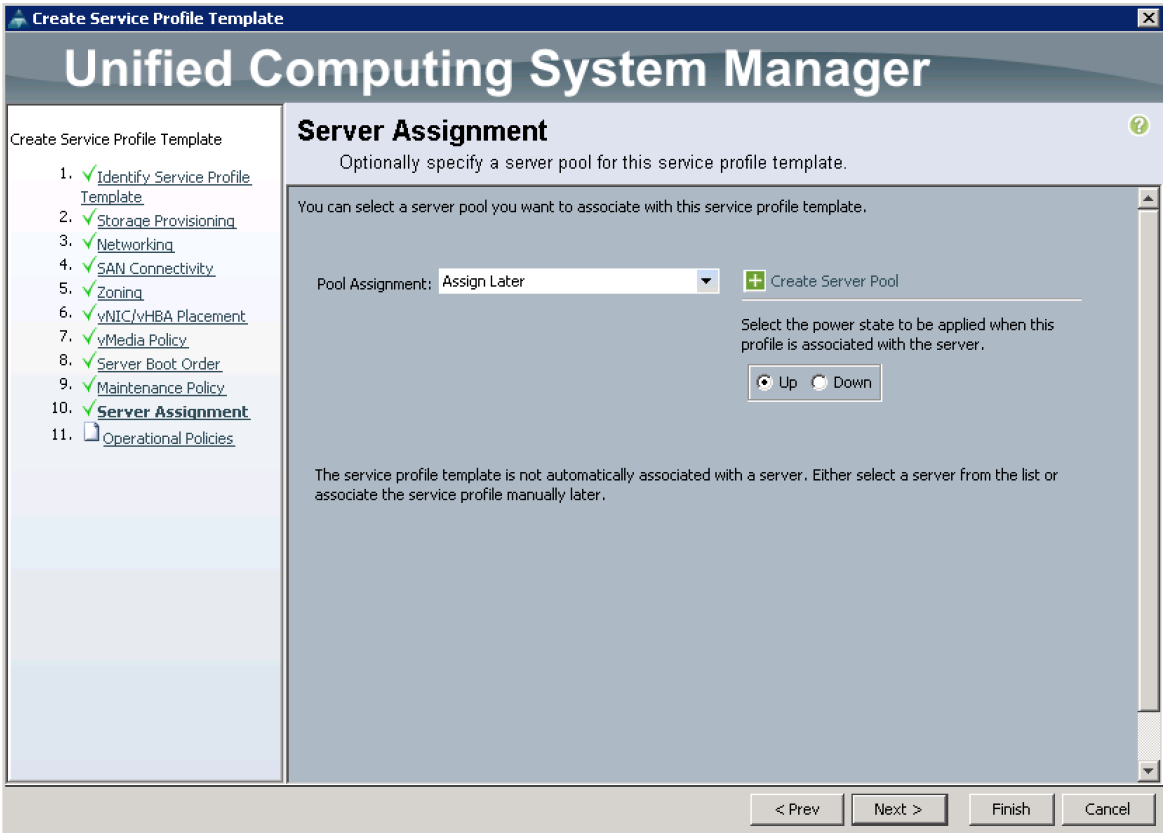
Maintenance Policy

1. From the Maintenance Policy drop-down list, select the Maintenance Policy you previously created under.



2. Click **Next**.

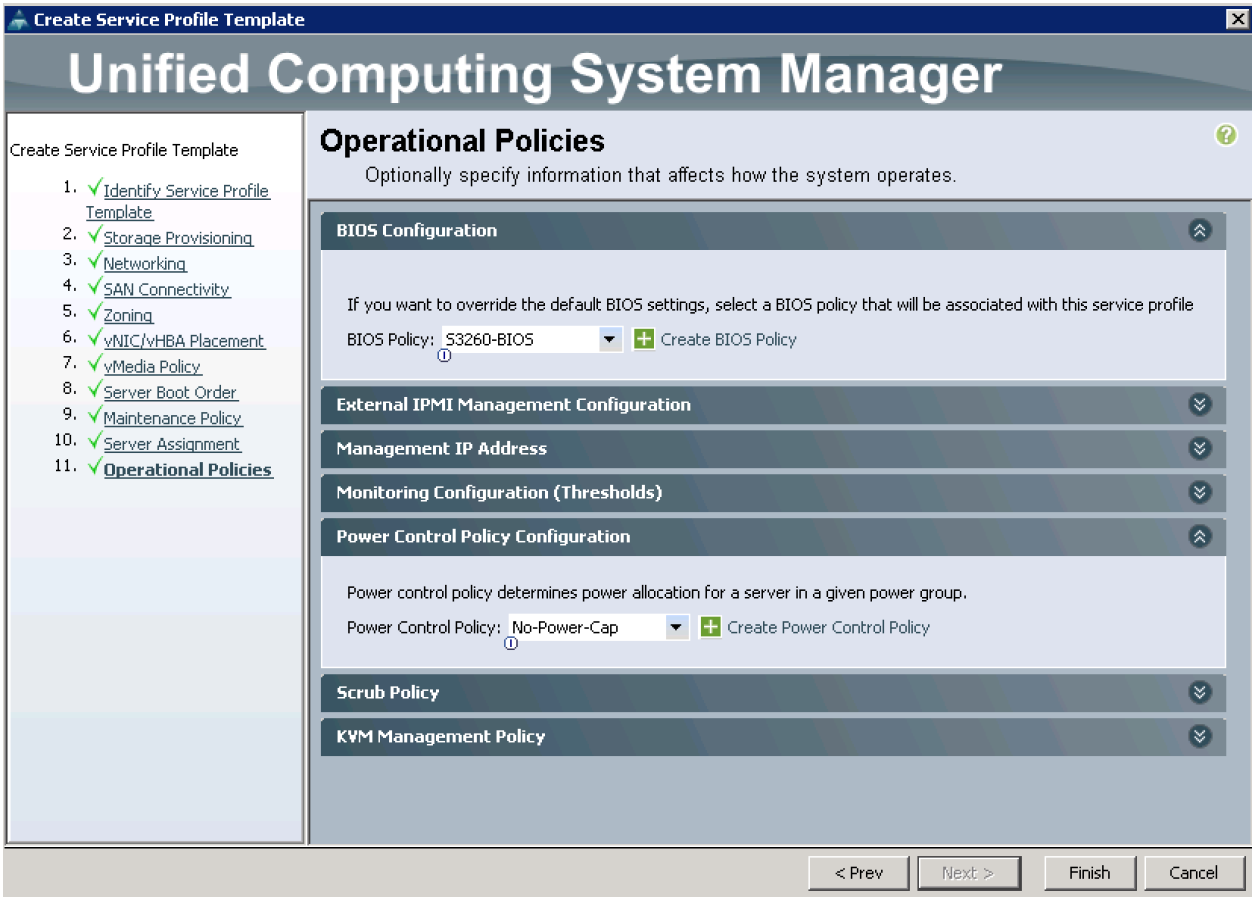
3. For Server Assignment, keep the default settings.



4. Click **Next**.

Operational Policies

1. Under Operational Policies, for the BIOS Configuration, select the previously created BIOS Policy “S3260-BIOS”. Under Power Control Policy Configuration, select the previously created Power Policy “No-Power-Cap”.



2. Click **Finish** and then click **OK**.

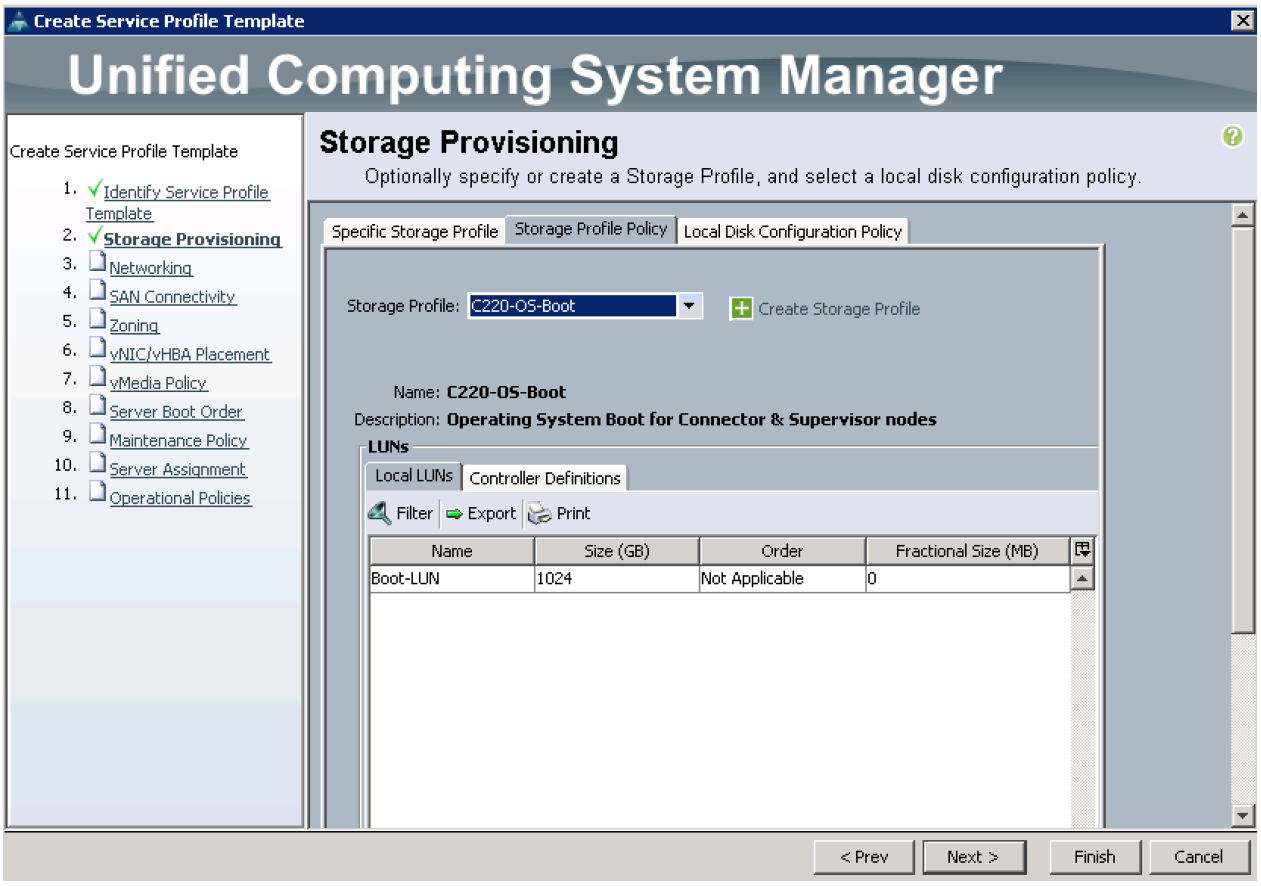
3. Repeat the steps for the bottom node of the Cisco UCS S3260 Storage Server, but change the following:

   a. Choose the Storage Profile for the bottom node you previously.

Create Service Profile Template for Cisco UCS C220 M4S

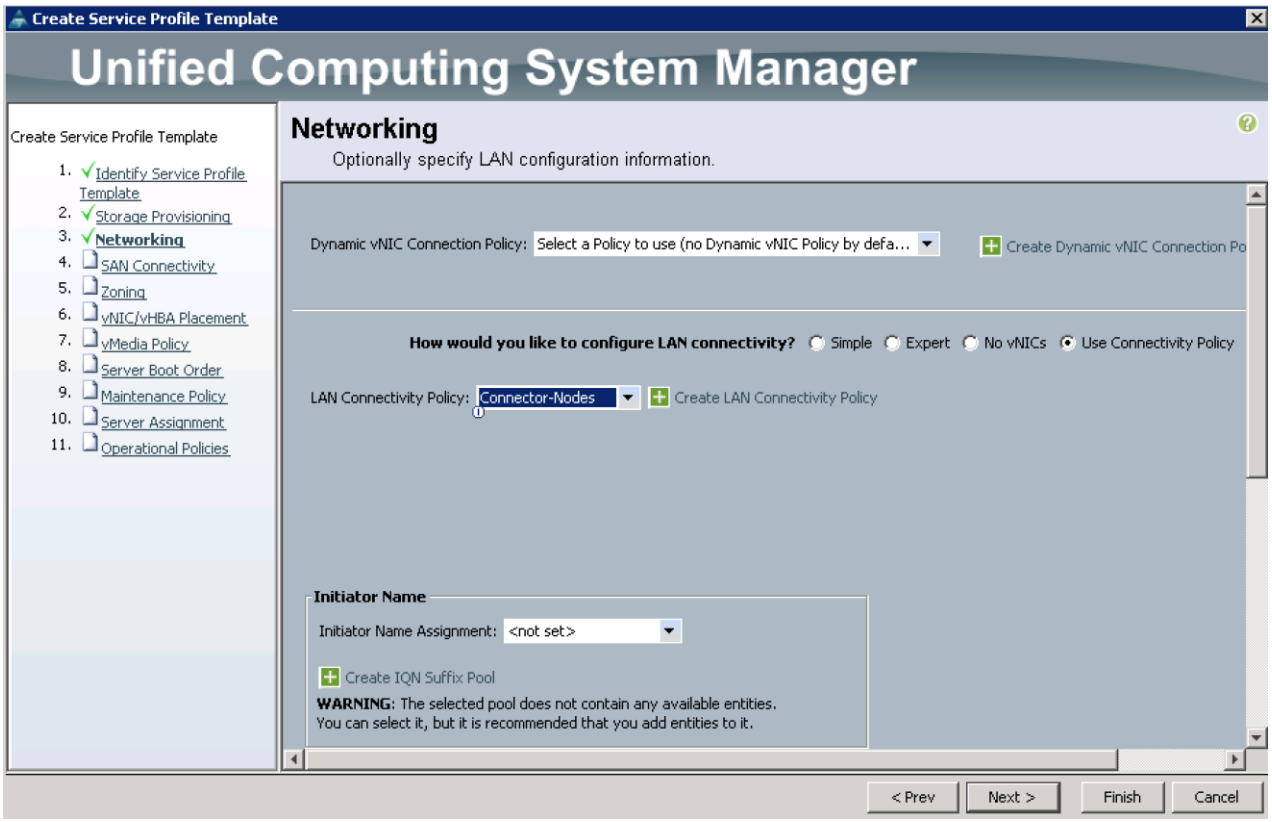
The Service Profiles for the Cisco UCS Rack-Mount Servers are very similar to the above created for the S3260. The only differences are with the Storage Profiles, Networking, vNIC/vHBA Placement, and BIOS Policy. The changes are listed here:

1. In the Storage Provisioning tab, choose the appropriate Storage Profile for the Cisco C220 M4S you previously created.

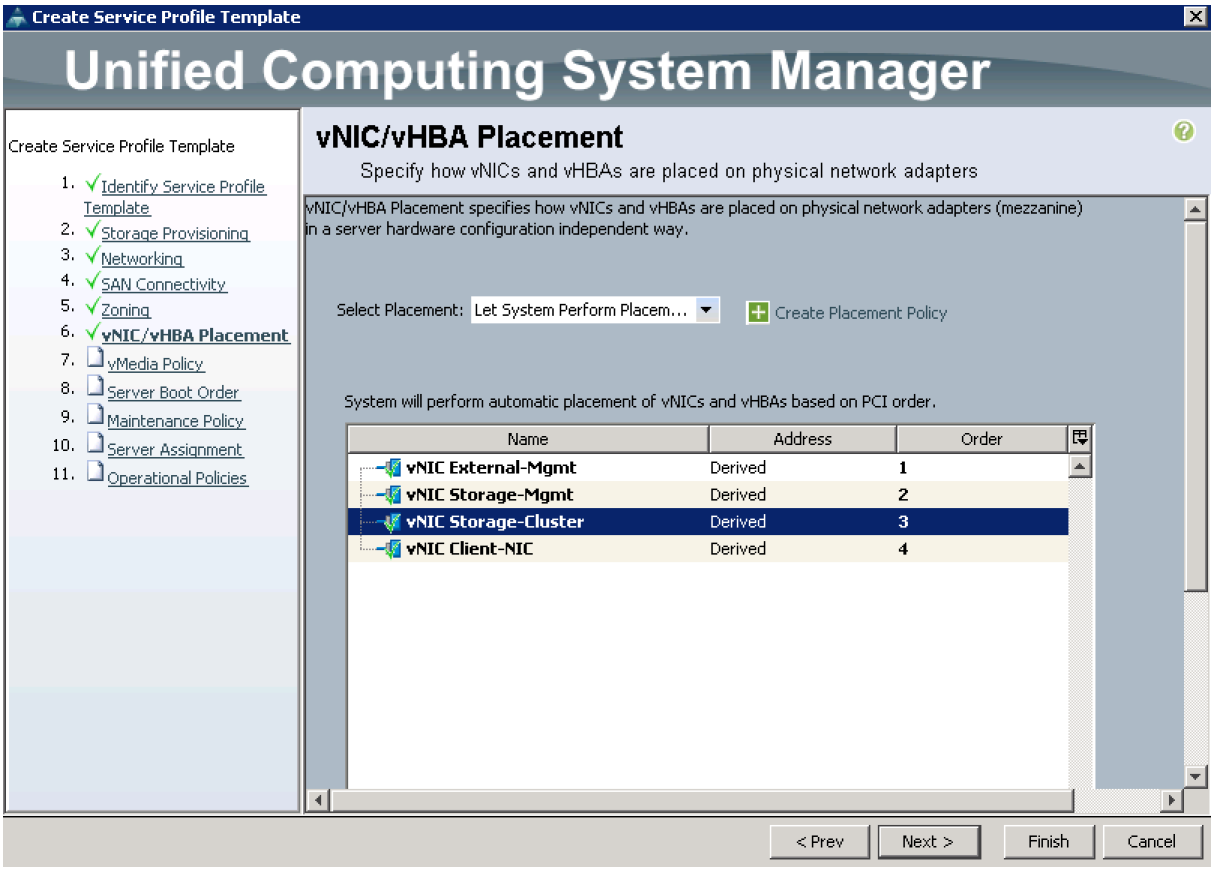


2. In the Networking tab, keep the Dynamic vNIC connection policy as default and select the LAN connectivity policy from the drop-down list the “Connector-Nodes” previously created.

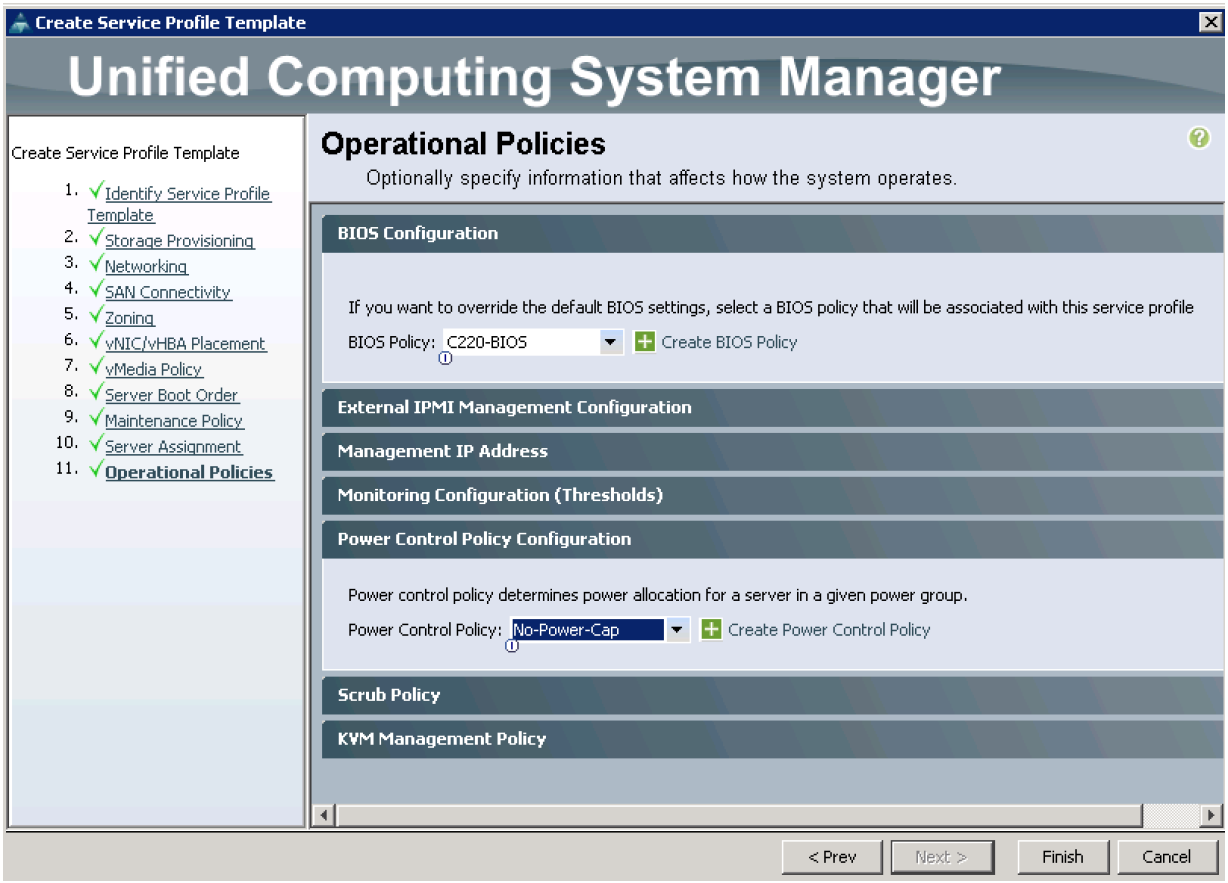
3. Click Next.



4. Configure the vNIC/vHBA Placement in the following order as shown in the screenshot below:



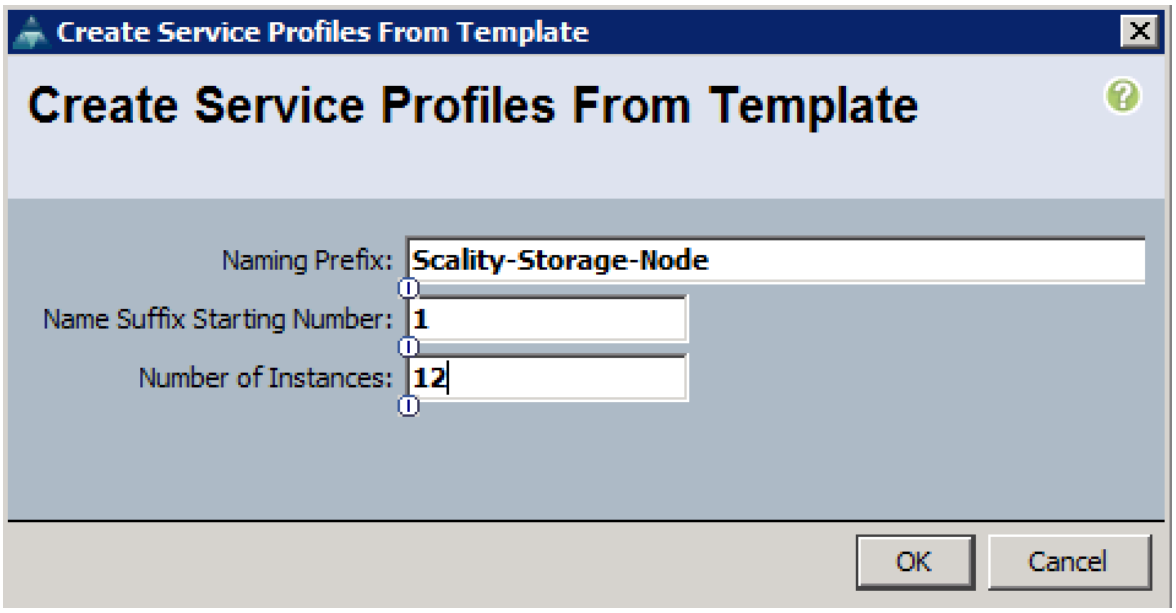
5. In the Operational Policies tab, under BIOS Configuration, select the previously created BIOS Policy "C220-BIOS". Under Power Control Policy Configuration, select the previously created Power Policy "No-Power-Cap."



Create Service Profiles from Template

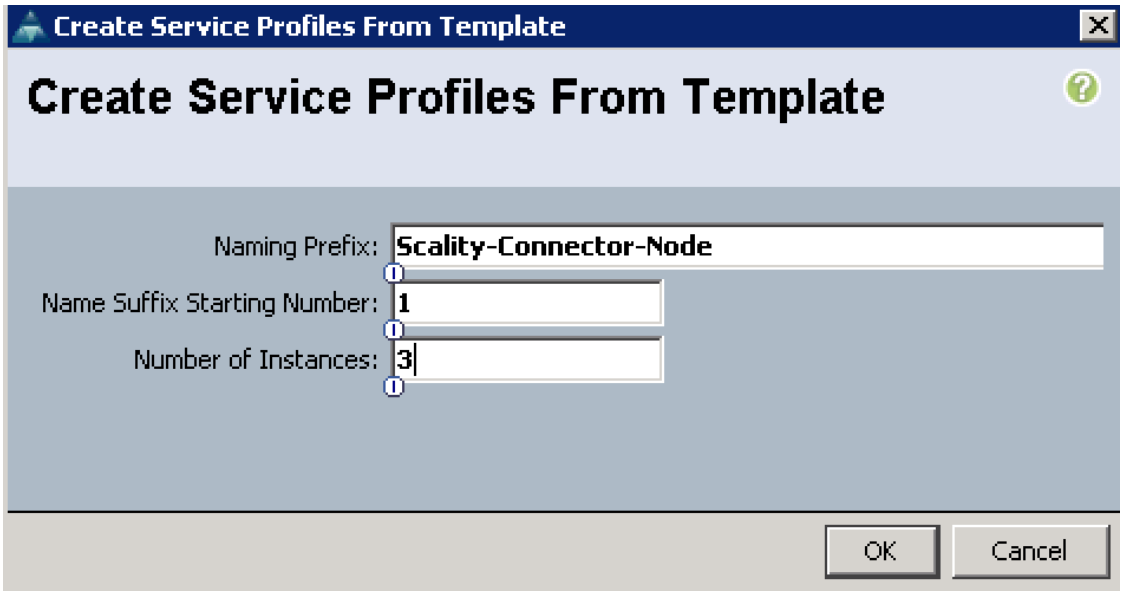
Now create the appropriate Service Profiles from the previous Service Profile Templates. To create the first profile for the top node of the Cisco UCS S3260 Storage Server, complete the following steps:

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 Scality-Storage-Node in the Name Prefix field.
4. Leave Name Suffix Starting Number as 1.
5. Type in 12 for the Number of Instances.
6. Choose Scality-Storage-Node-Template as the Service Profile Template you created before for the top node of the Cisco UCS S3260 Storage Server.
7. Click OK and then click OK again.
8. Create Service Profiles from Template for all the S3260 M4 nodes.



9. Repeat steps 1-7 for the next Service Profile for the Cisco UCS C220 M4S Rack-Mount Server and choose the appropriate Service Profile Template Scality-Connector-Node-Template you previously created for the Cisco UCS C220 M4 S Rack-Mount Server.

10. Create Service Profiles from Template for the C220 M4S for Connector Nodes.



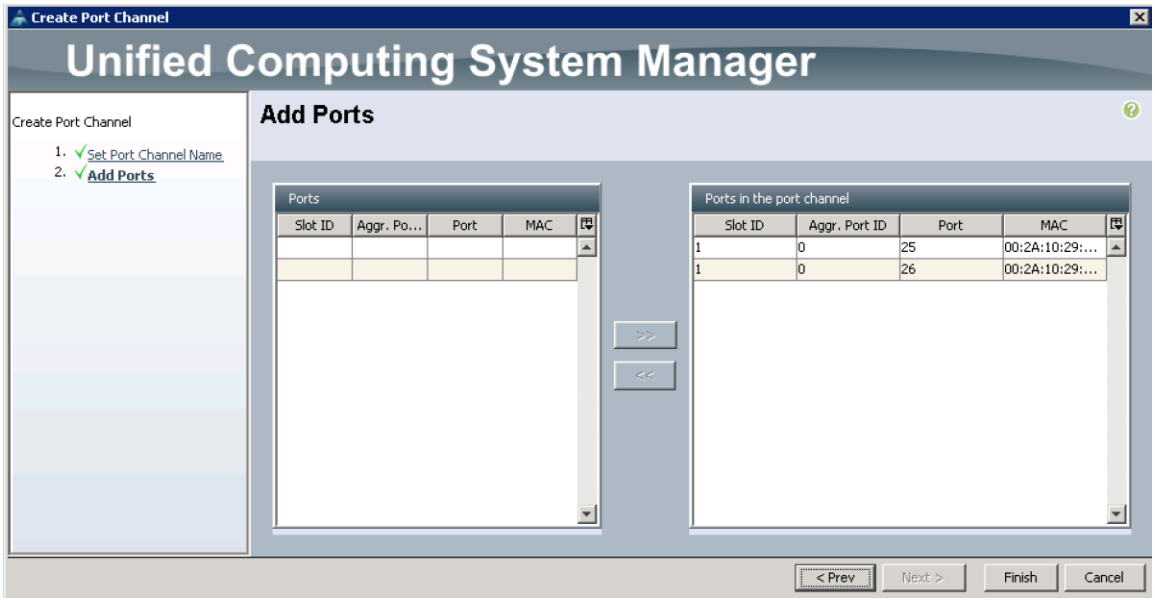
Creating Port Channel for Uplinks

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 10.
4. Type in vPC10 in the Name field.
5. Click Next.
6. Select the available ports on the left 25-26 and assign them with >> to Ports in the Port Channel.



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 11.
10. Type in VPC11 name in the Name field.
11. Click Next.
12. Select the available ports on the left 25-26 and assign them with >> to Ports in the Port Channel.
13. Click Finish and then click OK.

Configuration of Nexus 9332PQ Switch A and B

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

Initial Setup of Nexus 9332PQ Switch A and B

To configure Switch A, 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]: no
```

```
  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]: no
  Configure read-write SNMP community string (yes/no) [n]: no
  Enter the switch name : N9k-Fab-A
  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: yes
    Mgmt0 IPv4 address : 192.168.10.103
    Mgmt0 IPv4 netmask : 255.255.255.0
  Configure the default gateway? (yes/no) [y]: yes
    IPv4 address of the default gateway : 192.168.10.1
  Configure advanced IP options? (yes/no) [n]: no
  Enable the telnet service? (yes/no) [n]: no
  Enable the ssh service? (yes/no) [y]: yes
```

```
    Type of ssh key you would like to generate (dsa/rsa) [rsa]: rsa
    Number of rsa key bits <1024-2048> [1024]: 1024
  Configure the ntp server? (yes/no) [n]: yes
    NTP server IPv4 address : 192.168.10.2
  Configure default interface layer (L3/L2) [L3]: L2
  Configure default switchport interface state (shut/noshut) [shut]: shut
  Configure CoPP system profile (strict/moderate/lenient/dense) [strict]:
The following configuration will be applied:
  password strength-check
  switchname N9k-Fab-A
vrf context management
ip route 0.0.0.0/0 192.168.10.1
exit
  no feature telnet
  ssh key rsa 1024 force
  feature ssh
  ntp server 192.168.10.2
  no system default switchport
  system default switchport shutdown
  copp profile strict
interface mgmt0
ip address 192.168.10.103 255.255.255.0
no shutdown

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

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

[########################################] 100%
Copy complete.
```

```
User Access Verification
N9k-Fab-A login:
```

Note: Repeat the same steps for the Nexus 9332PQ Switch B with the exception of configuring a different IPv4 management address 192.168.10.104 as described 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-Fab-A# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-A(config)# feature udld
N9k-Fab-A(config)# feature interface-vlan
N9k-Fab-A(config)# feature hsrp
N9k-Fab-A(config)# feature lacp
N9k-Fab-A(config)# feature vpc
N9k-Fab-A(config)# system jumbomtu 9216
N9k-Fab-A(config)# exit
N9k-Fab-A(config)# copy running-config startup-config
```

Switch B

```
N9k-Fab-B# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-B(config)# feature udld
N9k-Fab-B(config)# feature interface-vlan
N9k-Fab-B(config)# feature hsrp
N9k-Fab-B(config)# feature lacp
N9k-Fab-B(config)# feature vpc
N9k-Fab-B(config)# system jumbomtu 9216
N9k-Fab-B(config)# exit
N9k-Fab-B(config)# copy running-config startup-config
```

Configuring VLANs on Nexus 9332PQ Switch A and B

To configure the same VLANs Storage-Management, Storage-Cluster, Client Network and External Management, as previously created in the Cisco UCS Manager GUI, complete the following steps on Switch A and Switch B:

Switch A

```
N9k-Fab-A# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-A(config)# vlan 10
N9k-Fab-A(config-vlan)# name Storage-Management
N9k-Fab-A(config-vlan)# no shut
N9k-Fab-A(config-vlan)# exit
N9k-Fab-A(config)# vlan 20
N9k-Fab-A(config-vlan)# name Storage-Cluster
N9k-Fab-A(config-vlan)# no shut
N9k-Fab-A(config-vlan)# exit
N9k-Fab-A(config)# vlan 30
N9k-Fab-A(config-vlan)# name Client-Network
N9k-Fab-A(config-vlan)# no shut
N9k-Fab-A(config-vlan)# exit
N9k-Fab-A(config)# vlan 79
N9k-Fab-A(config-vlan)# name External-Mgmt
N9k-Fab-A(config-vlan)# no shut
N9k-Fab-A(config-vlan)# exit

N9k-Fab-A(config)# interface vlan10
N9k-Fab-A(config-if)# description Storage-Mgmt
N9k-Fab-A(config-if)# no shutdown
N9k-Fab-A(config-if)# no ip redirects
N9k-Fab-A(config-if)# ip address 192.168.10.253/24
N9k-Fab-A(config-if)# no ipv6 redirects
N9k-Fab-A(config-if)# hsrp version 2
N9k-Fab-A(config-if)# hsrp 10
```

```
N9k-Fab-A(config-if-hsrp)# preempt
N9k-Fab-A(config-if-hsrp)# priority 10
N9k-Fab-A(config-if-hsrp)# ip 192.168.10.1
N9k-Fab-A(config-if-hsrp)# exit
N9k-Fab-A(config-if)# exit

N9k-Fab-A(config)# interface vlan20
N9k-Fab-A(config-if)# description Storage-Cluster
N9k-Fab-A(config-if)# no shutdown
N9k-Fab-A(config-if)# no ip redirects
N9k-Fab-A(config-if)# ip address 192.168.20.253/24
N9k-Fab-A(config-if)# no ipv6 redirects
N9k-Fab-A(config-if)# hsrp version 2
N9k-Fab-A(config-if)# hsrp 20
N9k-Fab-A(config-if-hsrp)# preempt
N9k-Fab-A(config-if-hsrp)# priority 10
N9k-Fab-A(config-if-hsrp)# ip 192.168.20.1
N9k-Fab-A(config-if-hsrp)# exit
N9k-Fab-A(config-if)# exit

N9k-Fab-A(config)# interface vlan30
N9k-Fab-A(config-if)# description Client-Network
N9k-Fab-A(config-if)# no shutdown
N9k-Fab-A(config-if)# no ip redirects
N9k-Fab-A(config-if)# ip address 192.168.30.253/24
N9k-Fab-A(config-if)# no ipv6 redirects
N9k-Fab-A(config-if)# hsrp version 2
N9k-Fab-A(config-if)# hsrp 20
N9k-Fab-A(config-if-hsrp)# preempt
N9k-Fab-A(config-if-hsrp)# priority 10
N9k-Fab-A(config-if-hsrp)# ip 192.168.30.1
N9k-Fab-A(config-if-hsrp)# exit
```

```
N9k-Fab-A(config-if)# exit
N9k-Fab-A(config)# copy running-config startup-config
```

Switch B

```
N9k-Fab-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-B(config)# vlan 10
N9k-Fab-B(config-vlan)# name Storage-Management
N9k-Fab-B(config-vlan)# no shut
N9k-Fab-B(config-vlan)# exit
N9k-Fab-B(config)# vlan 20
N9k-Fab-B(config-vlan)# name Storage-Cluster
N9k-Fab-B(config-vlan)# no shut
N9k-Fab-B(config-vlan)# exit
N9k-Fab-B(config)# vlan 30
N9k-Fab-B(config-vlan)# name Client-Network
N9k-Fab-B(config-vlan)# no shut
N9k-Fab-B(config-vlan)# exit
N9k-Fab-B(config)# vlan 79
N9k-Fab-B(config-vlan)# name External-Mgmt
N9k-Fab-B(config-vlan)# no shut
N9k-Fab-B(config-vlan)# exit

N9k-Fab-B(config)# interface vlan10
N9k-Fab-B(config-if)# description Storage-Mgmt
N9k-Fab-B(config-if)# no ip redirects
N9k-Fab-B(config-if)# ip address 192.168.10.254/24
N9k-Fab-B(config-if)# no ipv6 redirects
N9k-Fab-B(config-if)# hsrp version 2
N9k-Fab-B(config-if)# hsrp 10
N9k-Fab-B(config-if-hsrp)# preempt
N9k-Fab-B(config-if-hsrp)# priority 5
N9k-Fab-B(config-if-hsrp)# ip 192.168.10.1
```

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

N9k-Fab-B(config)# interface vlan20
N9k-Fab-B(config-if)# description Storage-Cluster
N9k-Fab-B(config-if)# no ip redirects
N9k-Fab-B(config-if)# ip address 192.168.20.254/24
N9k-Fab-B(config-if)# no ipv6 redirects
N9k-Fab-B(config-if)# hsrp version 2
N9k-Fab-B(config-if)# hsrp 20
N9k-Fab-B(config-if-hsrp)# preempt
N9k-Fab-B(config-if-hsrp)# priority 5
N9k-Fab-B(config-if-hsrp)# ip 192.168.20.1
N9k-Fab-B(config-if-hsrp)# exit
N9k-Fab-B(config-if)# exit

N9k-Fab-B(config)# interface vlan30
N9k-Fab-B(config-if)# description Client-Network
N9k-Fab-B(config-if)# no shutdown
N9k-Fab-B(config-if)# no ip redirects
N9k-Fab-B(config-if)# ip address 192.168.30.254/24
N9k-Fab-B(config-if)# no ipv6 redirects
N9k-Fab-B(config-if)# hsrp version 2
N9k-Fab-B(config-if)# hsrp 20
N9k-Fab-B(config-if-hsrp)# preempt
N9k-Fab-B(config-if-hsrp)# priority 5
N9k-Fab-B(config-if-hsrp)# ip 192.168.30.1
N9k-Fab-B(config-if-hsrp)# exit
N9k-Fab-B(config-if)# exit
N9k-Fab-B(config)# copy running-config startup-config
```

Configure vPC and Port Channels on Cisco 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-Fab-B# config terminal

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

N9k-Fab-A(config)# vpc domain 2

N9k-Fab-A(config-vpc-domain)# peer-keepalive destination 192.168.10.104

Note:

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

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

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


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

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

N9k-Fab-A(config-if)# switchport

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

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

N9k-Fab-A(config-if)# speed 40000

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


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

N9k-Fab-A(config-if)# description connected to peer N9k-Fab-B port 1

N9k-Fab-A(config-if)# switchport

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

N9k-Fab-A(config-if)# speed 40000

N9k-Fab-A(config-if)# channel-group 1 mode active

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

```
N9k-Fab-A(config)# interface ethernet 1/2
N9k-Fab-A(config-if)# description connected to peer N9k-Fab-B port 2
N9k-Fab-A(config-if)# switchport
N9k-Fab-A(config-if)# switchport mode trunk
N9k-Fab-A(config-if)# speed 40000
N9k-Fab-A(config-if)# channel-group 1 mode active
N9k-Fab-A(config-if)# exit
N9k-Fab-A(config)# copy running-config startup-config
```

vPC and Port Channels for Peerlink on Switch B

```
N9k-Fab-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-B(config)# vpc domain 2
N9k-Fab-B(config-vpc-domain)# peer-keepalive destination 192.168.10.103
Note:
 --------:: Management VRF will be used as the default VRF ::--------
N9k-Fab-B(config-vpc-domain)# peer-gateway
N9k-Fab-B(config-vpc-domain)# exit

N9k-Fab-B(config)# interface port-channel 1
N9k-Fab-B(config-if)# description vPC peerlink for N9k-Fab-A and N9k-Fab-B
N9k-Fab-B(config-if)# switchport
N9k-Fab-B(config-if)# switchport mode trunk
N9k-Fab-B(config-if)# spanning-tree port type network
N9k-Fab-B(config-if)# speed 40000
N9k-Fab-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-Fab-B(config-if)# exit

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

N9k-Fab-B(config)# interface ethernet 1/2
N9k-Fab-B(config-if)# description connected to peer N9k-Fab-A port 2
N9k-Fab-B(config-if)# switchport
N9k-Fab-B(config-if)# switchport mode trunk
N9k-Fab-B(config-if)# speed 40000
N9k-Fab-B(config-if)# channel-group 1 mode active
N9k-Fab-B(config-if)# exit
N9k-Fab-B(config)# copy running-config startup-config
```

vPC and Port Channels for Uplink from Fabric Interconnect A and B on Switch A

```
N9k-Fab-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-A(config)# interface port-channel 10
N9k-Fab-A(config-if)# description vPC for UCS FI-A port 25 & 26
N9k-Fab-A(config-if)# vpc 10
N9k-Fab-A(config-if)# switchport
N9k-Fab-A(config-if)# switchport mode trunk
N9k-Fab-A(config-if)# switchport trunk allowed vlan 10,20,30,79
N9k-Fab-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-Fab-A(config-if)# mtu 9216

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


N9k-Fab-A(config)# interface port-channel 11

N9k-Fab-A(config-if)# description vPC for UCS FI-B port 25 & 26

N9k-Fab-A(config-if)# vpc 11

N9k-Fab-A(config-if)# switchport

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

N9k-Fab-A(config-if)# switchport trunk allowed vlan 10,20,30,79

N9k-Fab-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-Fab-A(config-if)# mtu 9216

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


N9k-Fab-A(config)# interface ethernet 1/25

N9k-Fab-A(config-if)# switchport

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

N9k-Fab-A(config-if)# description Uplink from UCS FI-B port 25

N9k-Fab-A(config-if)# channel-group 10 mode active

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


N9k-Fab-A(config)# interface ethernet 1/26

N9k-Fab-A(config-if)# switchport

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

N9k-Fab-A(config-if)# description Uplink from UCS FI-B port 25
```

```
N9k-Fab-A(config-if)# channel-group 11 mode active
N9k-Fab-A(config-if)# exit
N9k-Fab-A(config)# copy running-config startup-config
```

vPC and Port Channels for Uplink from Fabric Interconnect A and B on Switch B

```
N9k-Fab-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-B(config)# interface port-channel 10
N9k-Fab-B(config-if)# description vPC for UCS FI-A port 25 & 26
N9k-Fab-B(config-if)# switchport
N9k-Fab-B(config-if)# switchport mode trunk
N9k-Fab-B(config-if)# switchport trunk allowed vlan 10,20,30,79
N9k-Fab-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-Fab-B(config-if)# vpc 10
N9k-Fab-B(config-if)# mtu 9216
N9k-Fab-B(config-if)# exit

N9k-Fab-B(config)# interface port-channel 11
N9k-Fab-B(config-if)# description vPC for UCS FI-B port 25 & 26
N9k-Fab-B(config-if)# switchport
N9k-Fab-B(config-if)# switchport mode trunk
N9k-Fab-B(config-if)# switchport trunk allowed vlan 10,20,30,79
N9k-Fab-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-Fab-B(config-if)# vpc 11
N9k-Fab-B(config-if)# mtu 9216
N9k-Fab-B(config-if)# exit

N9k-Fab-B(config)# interface ethernet 1/25
N9k-Fab-B(config-if)# switchport
N9k-Fab-B(config-if)# switchport mode trunk
N9k-Fab-B(config-if)# description Uplink from UCS FI-A port 26
N9k-Fab-B(config-if)# channel-group 10 mode active
N9k-Fab-B(config-if)# exit

N9k-Fab-B(config)# interface ethernet 1/26
N9k-Fab-B(config-if)# switchport
N9k-Fab-B(config-if)# switchport mode trunk
N9k-Fab-B(config-if)# description Uplink from UCS FI-B port 26
N9k-Fab-B(config-if)# channel-group 11 mode active
N9k-Fab-B(config-if)# exit
N9k-Fab-B(config)# copy running-config startup-config
```

Verification Check of Cisco Nexus C9332PQ Configuration for Switch A and B

Switch A

```
N9k-Fab-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-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
--   ----   ------ ----------- ------                     ------------
10   Po10   up     success     success                    10,20,30,79

11   Po11   up     success     success                    10,20,30,79

N9k-Fab-A(config)#
N9k-Fab-A(config)# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
```

```
        p - Up in delay-lacp mode (member)
        M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------------------------
1     Po1(SU)     Eth      LACP      Eth1/1(P)   Eth1/2(P)
10    Po10(SU)    Eth      LACP      Eth1/25(P)
11    Po11(SU)    Eth      LACP      Eth1/26(P)
N9k-Fab-A(config)#
```

Switch B

```
N9k-Fab-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-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     1,10,20,30,79

vPC status
----------------------------------------------------------------------
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- ------                     ------------
10   Po10   up     success     success                    10,20,30,79

11   Po11   up     success     success                    10,20,30,79

N9k-Fab-B(config)#
N9k-Fab-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)
10    Po10(SU)    Eth      LACP      Eth1/25(P)
11    Po11(SU)    Eth      LACP      Eth1/26(P)
N9k-Fab-B(config)#
```

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

Installing Red Hat Enterprise Linux 7.3 Operating System

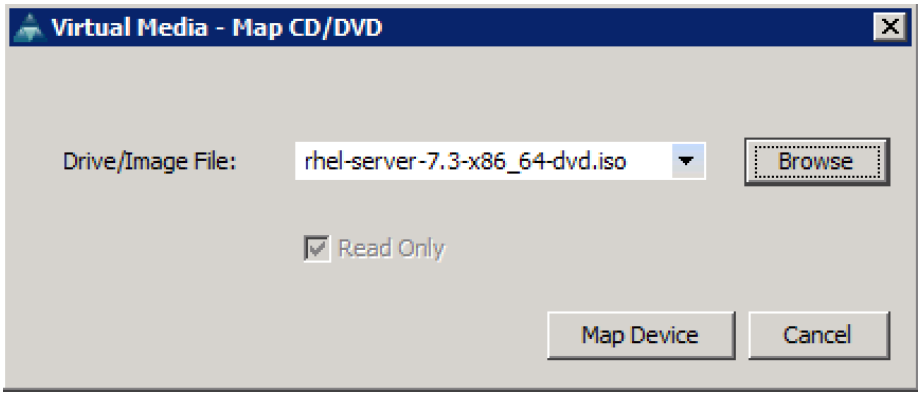
The following section provides the detailed procedures to install Red Hat Enterprise Linux 7.3 on Cisco UCS C220 M4S and Cisco UCS S3260 Storage Server. The installation uses the KVM console and virtual Media from Cisco UCS Manager.

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

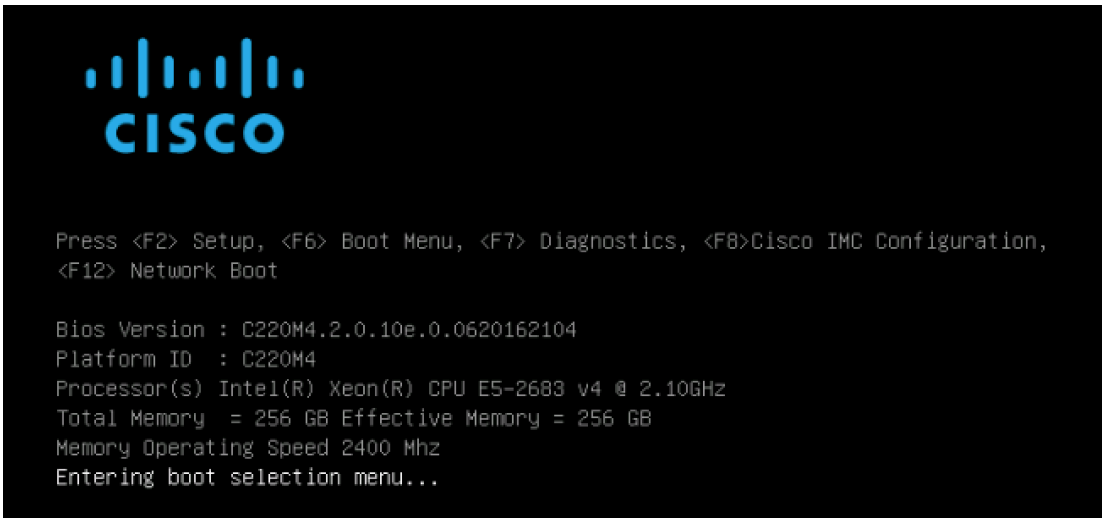
Installation of RHEL 7.3 on Cisco UCS C220 M4S

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

1. Log into the Cisco UCS Manager and select the Equipment tab from the left pane.
2. Go to Equipment > Rack-Mounts > Server > Server 1 (Supervisor) 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 then click Map CD/DVD.
6. Browse to the Red Hat Enterprise Linux 7.3 installation ISO image and select then Map Device.



7. In the KVM window, select the Macros > Static Macros > Ctrl-Alt-Del button in the upper left corner.
8. Click OK and then click OK to reboot the system.
9. In the boot screen with the Cisco Logo, press F6 for the boot menu.



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

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

Note: 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 C220 M4S, connector node can be found in Appendix A. In addition, we configured typical network interface names like eth1 for the Storage-Management network.

12. At the prompt type:

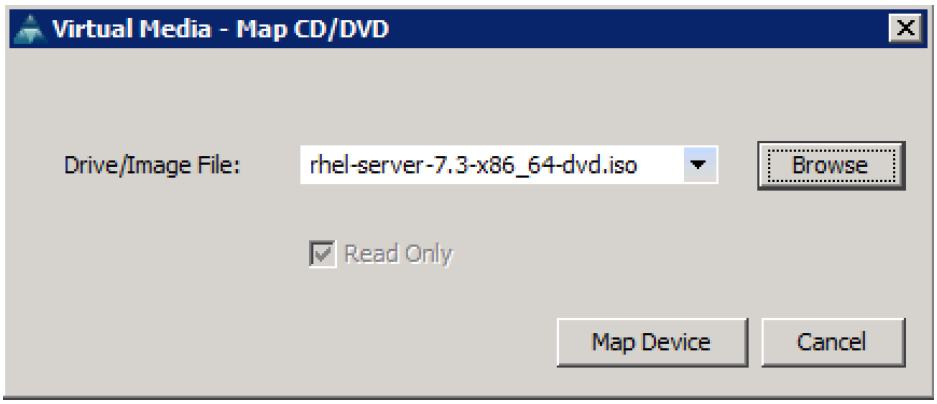
```
inst.ks=http://192.168.10.2/Scality-ks.cfg net.ifnames=0 biosdevname=0
ip=192.168.10.160::192.168.10.1:255.255.255.0:Supervisor:eth1:none
nameserver=192.168.10.222
```

13. Repeat the previous steps for Connector-Node1, Connector-Node2, and Connector-Node3.

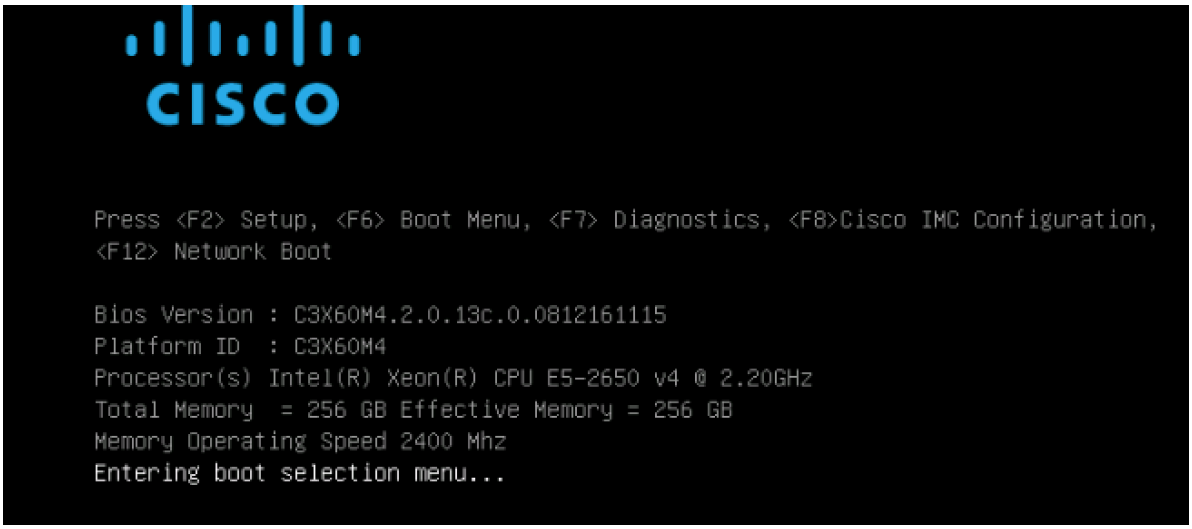
Installing RHEL 7.3 on Cisco UCS S3260 Storage Server

To install RHEL 7.3 on Cisco UCS S3260 storage server, complete the following steps:

1. Log into the Cisco UCS Manager and select the Equipment tab from the left pane.
2. Go to Equipment > Chassis > Chassis 1 > Server 1 (Storage-Node1) 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.



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.



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

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

Note: 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 eth1 for the Storage-Management network.

---

12. At the prompt type:

```
inst.ks=http://192.168.10.2/Scality-ks.cfg net.ifnames=0 biosdevname=0
ip=192.168.10.164::192.168.10.1:255.255.255.0:Storage-Node1:eth1:none
nameserver=192.168.10.222
```

13. Repeat the previous install steps for the remaining Storage-Node2 to Storage-Node12.

Post-Installation Steps for Red Hat Enterprise Linux 7.3

The Supervisor node 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 Scality Prerequisite installation.

Configure /etc/hosts and Enable Password-less Login

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

1. Modify the /etc/hosts file on Supervisor Node according to Table 7 and include all IP address of all nodes. An example is shown in Appendix C – Example /etc/hosts File.

Table 7 IP Addresses for Storage Nodes, Connector Nodes and Supervisor Node

Hostname	Storage-Mgmt	Storage-Cluster	Client-Network
Supervisor	192.168.10.160	192.168.20.160	
Connector-Node1	192.168.10.161	192.168.20.161	192.168.30.161
Connector-Node2	192.168.10.162	192.168.20.162	192.168.30.162
Connector-Node3	192.168.10.163	192.168.20.163	192.168.30.163
Storage-Node1	192.168.10.164	192.168.20.164	
Storage-Node2	192.168.10.165	192.168.20.165	
Storage-Node3	192.168.10.166	192.168.20.166	
Storage-Node4	192.168.10.167	192.168.20.167	
Storage-Node5	192.168.10.168	192.168.20.168	
Storage-Node6	192.168.10.169	192.168.20.169	
Storage-Node7	192.168.10.170	192.168.20.170	
Storage-Node8	192.168.10.171	192.168.20.171	
Storage-Node9	192.168.10.172	192.168.20.172	
Storage-Node10	192.168.10.173	192.168.20.173	
Storage-Node11	192.168.10.174	192.168.20.174	
Storage-Node12	192.168.10.175	192.168.20.175	

2. Login to Supervisor Node and change /etc/hosts.

```
# ssh root@192.168.10.160

# 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 Connector & Storage nodes.

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

# for i in {1..12}; do ssh-copy-id Storage-Node${i}; done
```

6. Copy /etc/hosts to all nodes.

```
# for i in {1..15}; do scp /etc/hosts 192.168.10.16${i}:/etc/; done
```

7. Login to local RHEL server and subscribe to Red Hat CDN.

```
# 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; --enable=rhel-7-
server-extras-rpms; --enable=rhel-7-server-optional-rpms
```

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 the system connected to the Internet download Cluster shell (clush) and copy and install it on Connector and Storage nodes. Cluster shell is available from EPEL (Extra Packages for Enterprise Linux) repository.

```
# yum install yum-plugin-downloadonly

# yum install --downloadonly --downloaddir=/root clustershell

# scp clustershell-1.7.2-1.el7.noarch.rpm supervisor:/root/
```

2. Login to supervisor node and install cluster shell.

```
# yum clean all

# yum repolist

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

3. 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 12 node cluster as in our CVD, set groups file as follows:

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

all: connector-node[1-3] storage-node[1-12]
```

Configuring Hostnames

1. Configure hostname for Supervisor Node and all other nodes:

```
# hostnamectl set-hostname Supervisor

# for i in {1..3}; do ssh Connector-Node${i} "hostnamectl set-hostname
Connector-Node${i}"; done

# for i in {1..12}; do ssh Storage-Node${i} "hostnamectl set-hostname Storage-
Node${i}"; done
```

Install Network Driver

To install the latest network driver for performance and updates, download the latest ISO image to a node connected to the internet.

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

2. Mount the ISO image on a local RHEL host, go to /Network/Cisco/VIC/RHEL/RHEL7.3 and copy the file kmod-enic-2.3.0.31-rhel7u3.el7.x86_64.rpm to Supervisor Node.

```
# 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/

# scp kmod-enic-2.3.0.31-rhel7u3.el7.x86_64.rpm supervisor:/tmp
```

3. Copy the file from supervisor node to all other nodes.

```
# ssh supervisor

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

4. Install the VIC driver on supervisor 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"
```

5. Verify the installation of the VIC driver.

```
# clush -a -b "modinfo enic | head -5"
```

Preparing all Nodes for Scality RING Installation

Before installing Scality RING, you need to install Scality Salt agent on all nodes (Supervisor, Connector, and Storage server). Make sure you prepare all nodes with certain configurations.

To install, complete all prerequisites for the whole installation with the appropriate changes to the current environment, and complete the following steps:

Step 1 – Update of all Connector & Storage Nodes

1. Login to root and update RHEL.

```
# ssh Supervisor
# yum –y update
# clush –a –b yum –y update
```

Step 2 – Configuring Firewall

To enable the Firewall on all Connector and Storage Nodes, complete the following steps:

1. On Supervisor Node:

```
# clush –a –b "systemctl enable firewalld"
# clush –a –b "systemctl start firewalld"
# clush –a –b "systemctl status firewalld"
```

Step 3 – Configuring Network Time Protocol

In your Kickstart installation file, you already included a time server. Now, enable the 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 Supervisor node only with the following contents:

```
# vi /etc/ntp.conf
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
server 192.168.10.2
fudge 192.168.10.2 stratum 10
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

3. Start the ntpd daemon on Supervisor Node:

```
# systemctl enable ntpd
# systemctl start ntpd
# systemctl status ntpd
```

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

```
# vi /root/ntp.conf
server supervisor
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 Connector and Storage nodes:

```
# clush –a –b "service ntpd stop"
# clush –a –b "ntpdate Supervisor"
# clush –a –b "service ntpd start"
# clush –a –b "systemctl enable ntpd"
```

Step 4 – Enabling Password-Less SSH

The user `root` needs password-less access from the administration node Supervisor to all Connector and Storage nodes. To enable this function, complete the following steps:

1. On the supervisor node log in as user root

```
$ ssh-keygen
```

2. Press Enter, then Enter and again Enter.

3. Copy id_rsa.pub under /root/.ssh to Connector-Node1.

```
$ ssh-copy-id root@Connector-Node1
```

4. Repeat the steps for Connector2-3 and Storage-Node1-12.

Scality Salt Installation

1. Install SALT Master on the supervisor server.

```
supervisor # yum –y install salt-master

supervisor # systemctl enable salt-master

supervisor # systemctl restart salt-master
```

2. Install SALT Minion on all of the servers:

```
supervisor # for I in supervisor connector-node1 connector-node2 connector-node3
storage-node1 storage-node2 storage-node3 storage-node4 storage-node5 storage-
node6 storage-node7 storage-node8 storage-node9 storage-node10 storage-node11
storage-node12

> do

> ssh $i “yum –y install salt-minion; systemctl enable salt-minion; systemctl
restart salt-minion”

> done
```

3. Accept the minion keys from all the servers:

```
supervisor # salt-key –A
```

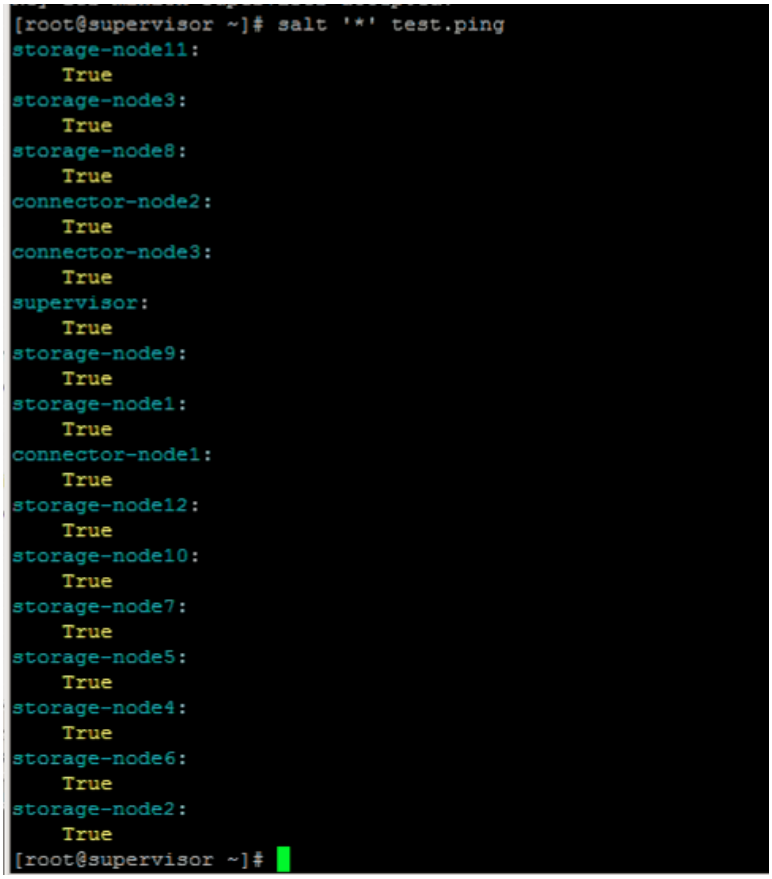
4. The following keys are going to be accepted:

```
Unaccepted Keys:

connector-node1

connector-node2

connector-node3

storage-node1

storage-node10

storage-node11

storage-node12

storage-node2

storage-node3

storage-node4

storage-node5

storage-node6

storage-node7

storage-node8

storage-node9

supervisor

Proceed? [n/Y] Y
```

5. Test the SALT installation with a simple test.ping command. All minions should report back 'True'. If some of the minions do not respond the first time, try the command again. The initial communication from master to minion can be sluggish and is usually resolved by retrying the command.

```
supervisor # salt '*' test.ping
```

All of the minions should report back as shown below:



Scality RING Installation

To install Scality RING, complete the following steps:

1. Download the Scality Installer.

```
supervisor # wget –user=christopher.donohoe –ask-password
https://packages.scality.com/stable_mithrandir/centos/7/x86_64/scality/ring/scality-ring-6.3.0.r161125113926.ff4fa5b.hf4_centos_7.run
```

Note: You need to obtain your own credentials and the link to the latest version of the Scality Installer from Scality Support.

2. Launch the Scality RING Installer.

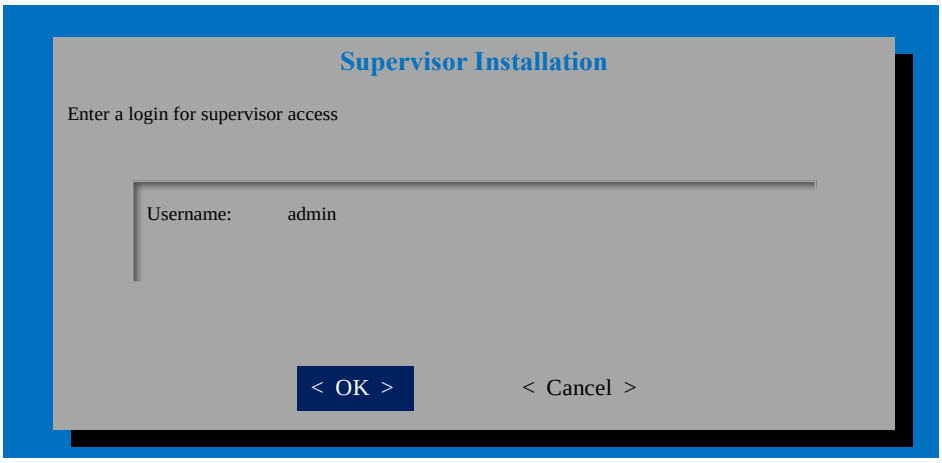
```
supervisor # ./scality-ring-6.3.0.r161125113926.ff4fa5b.hf4_centos_7.run -- -- --ssd-detection=sysfs --no-preload
```

3. This command launches the installer with two options.

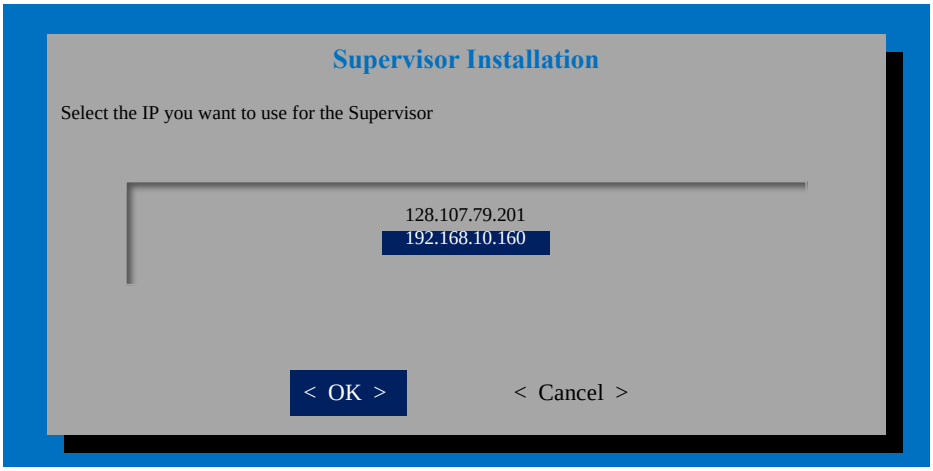
   a. '--ssd-detection=sysfs' identifies the SSDs by looking at the value in /sys/block/[disk]/queue/rotational for each disk device. A value of '0' identifies the disk as a SSD.

   b. '--no-preload' tells the installer to install packages as needed. Without this option, the installer would attempt to download all Scality packages from the online Scality repo prior to continuing with the installation. This causes unacceptable delays in some installations.

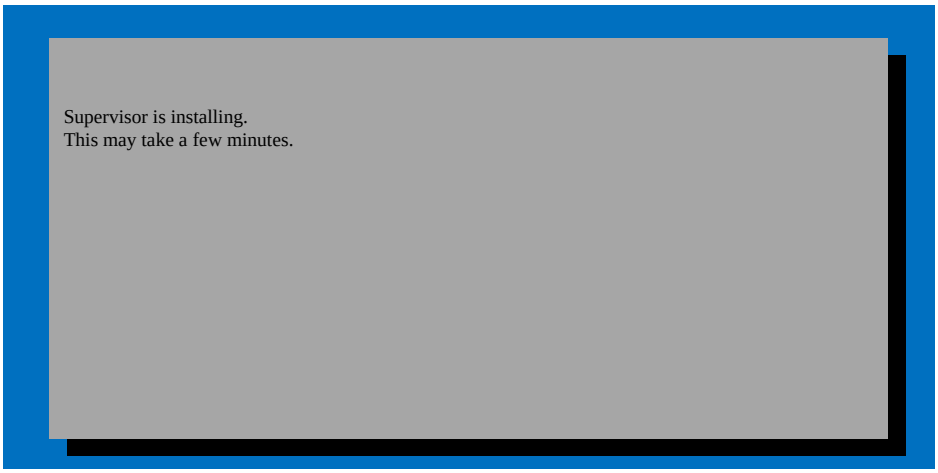
4. When launched, the installer should prompt for supervisor credentials. In this example, the credentials are admin/admin, but you can obviously make these as complex as you would like.



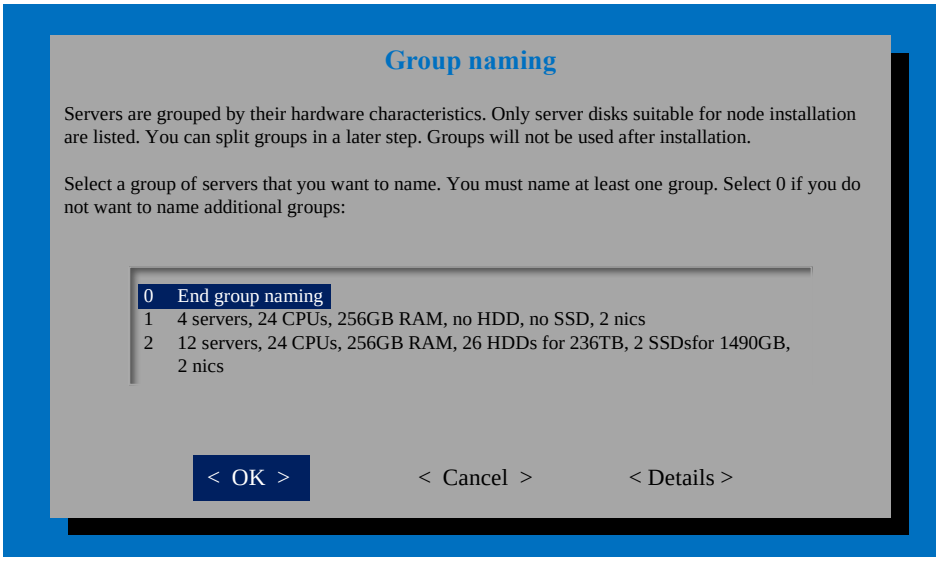
5. The IP chosen for the Supervisor should be the IP dedicated to the management of the Scality cluster. In this case, 192.168.10.160 is the IP of the internally-facing network planned for management, so it has been selected.



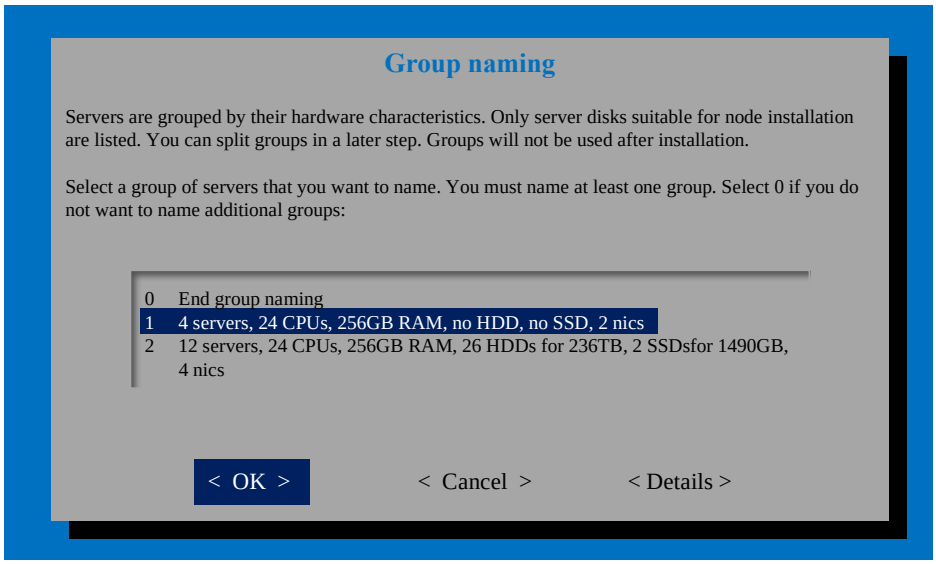
6. When you choose the supervisor IP, the supervisor will install and then prompt the administrator to identify the servers in the environment.



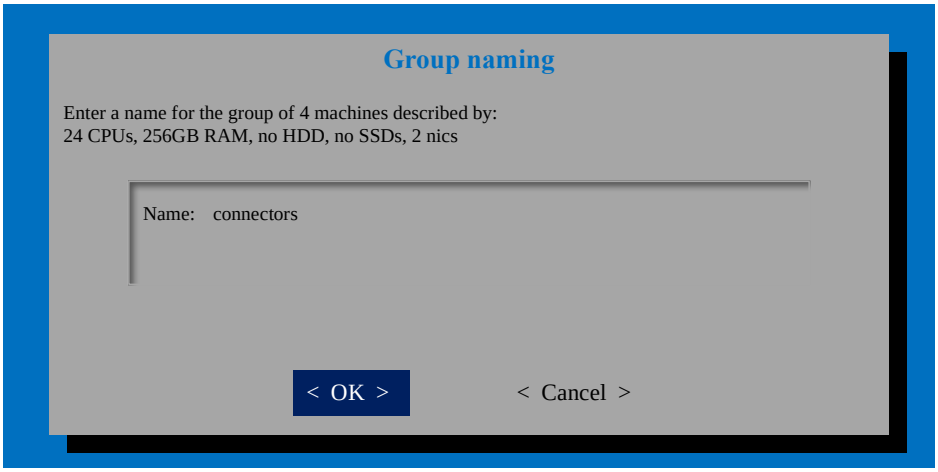
7. In this case, the supervisor server and connectors are in a unique group of four because their hardware characteristics do not match the storage servers.



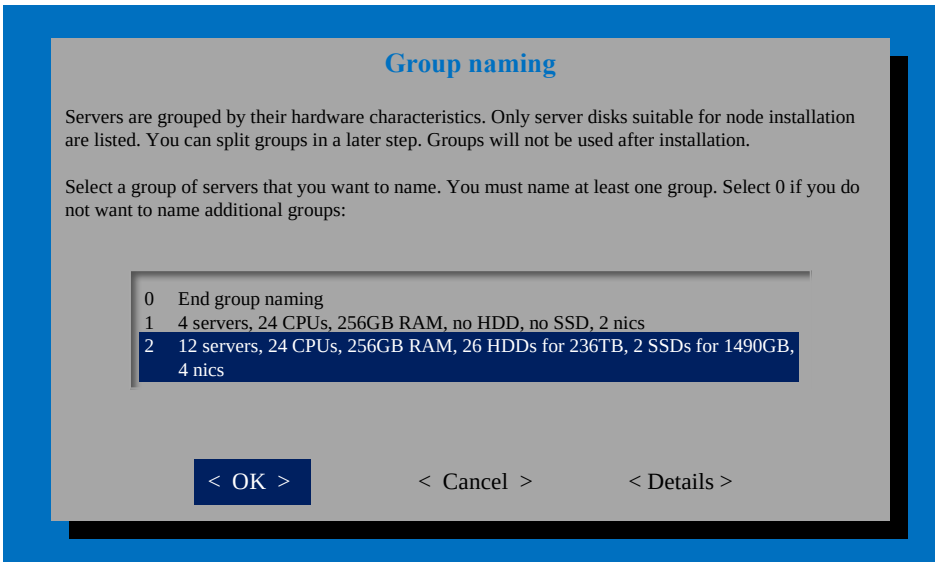
8. Choose the first group and name it the “connectors” group. The supervisor will be split later in the installation process.



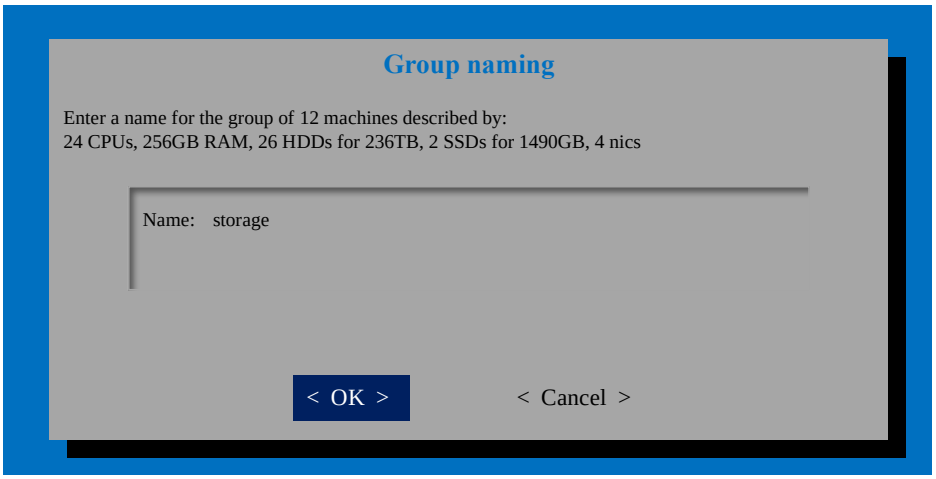
9. Name it the “connectors” group. The supervisor will be split later in the installation process.



10. Select the remaining 12-server group.

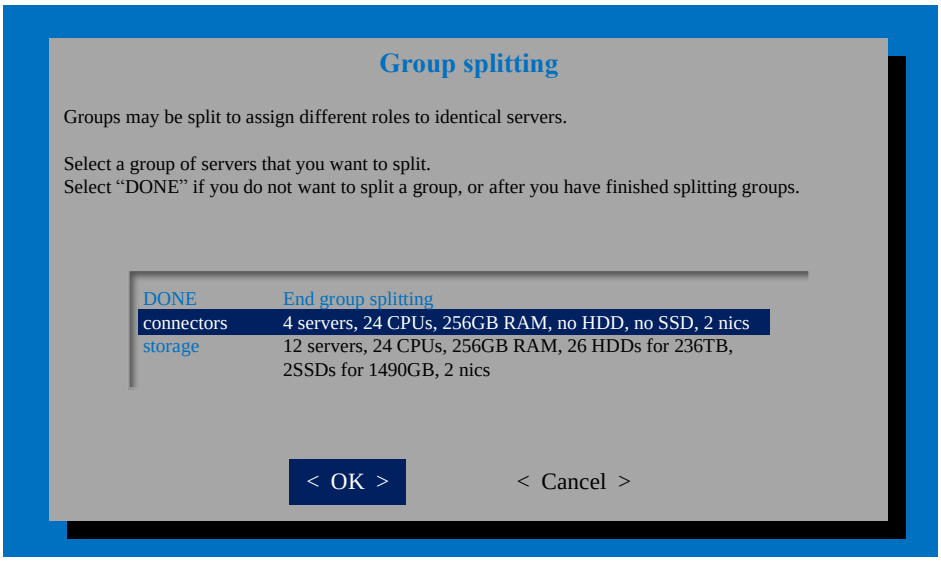


11. Name it the “storage” group.

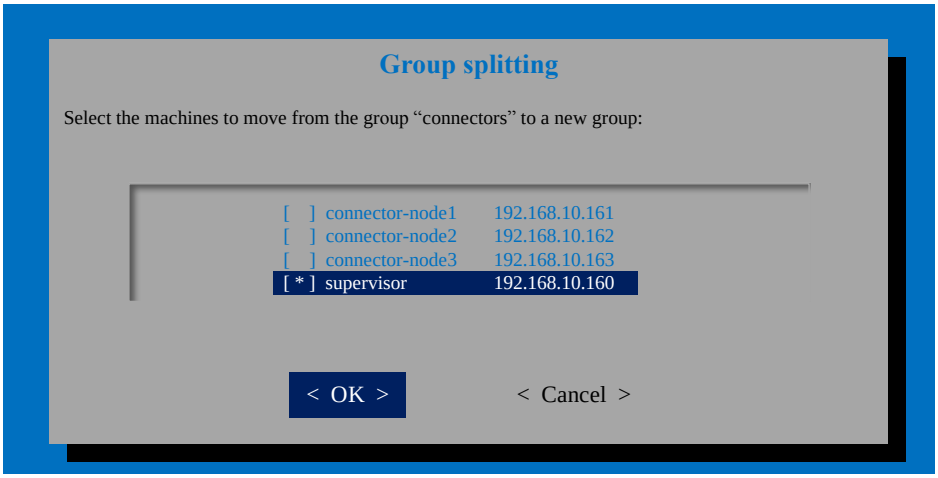


12. When you name the storage group, you will be asked if you want to further split this group into smaller subsets of servers. This will allow for role assignment of servers later in the installation.

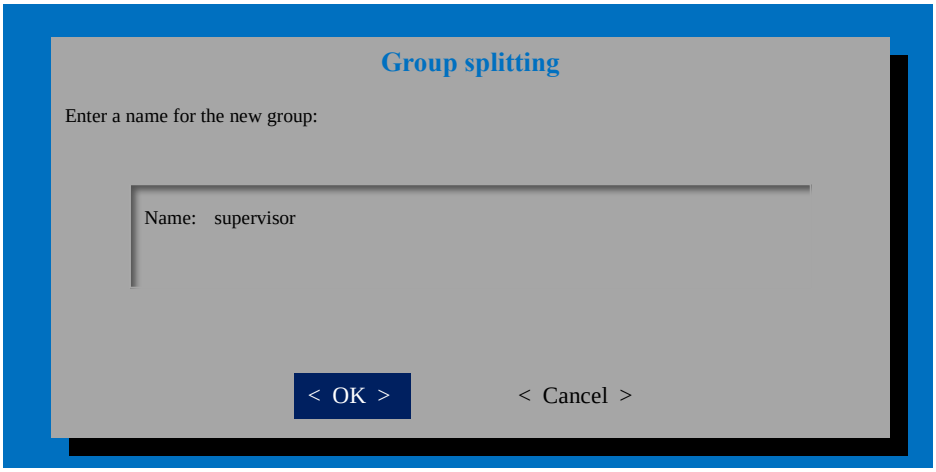
13. Select the connectors group and move the supervisor to its own “supervisor” group.



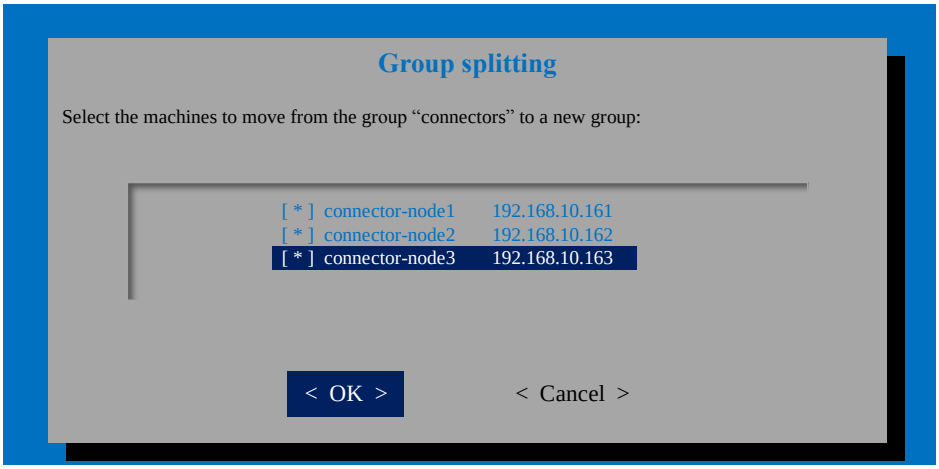
14. From Group Splitting, Choose "supervisor" to split out from the connectors group.



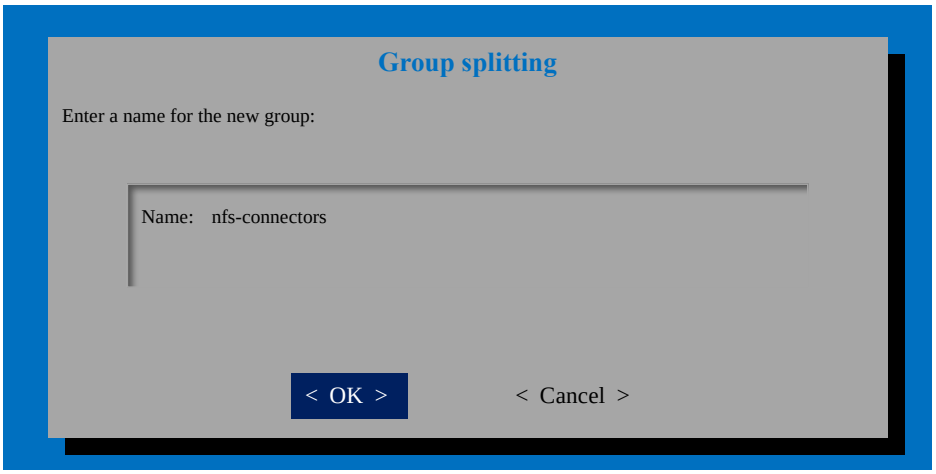
15. Name it the "supervisor" group.



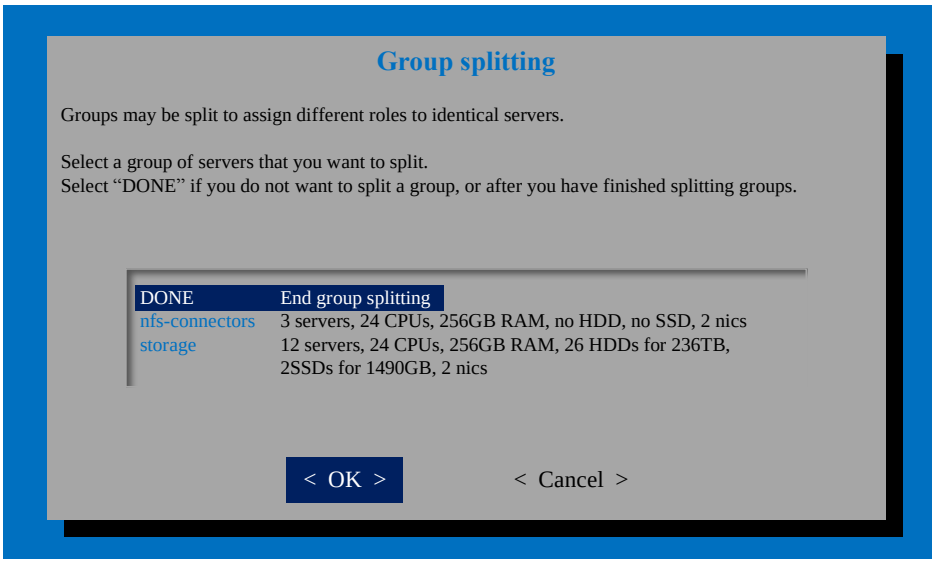
16. Select connector-node1, connector-node2, and connector-node3 and move these to a “nfs-connectors” group.



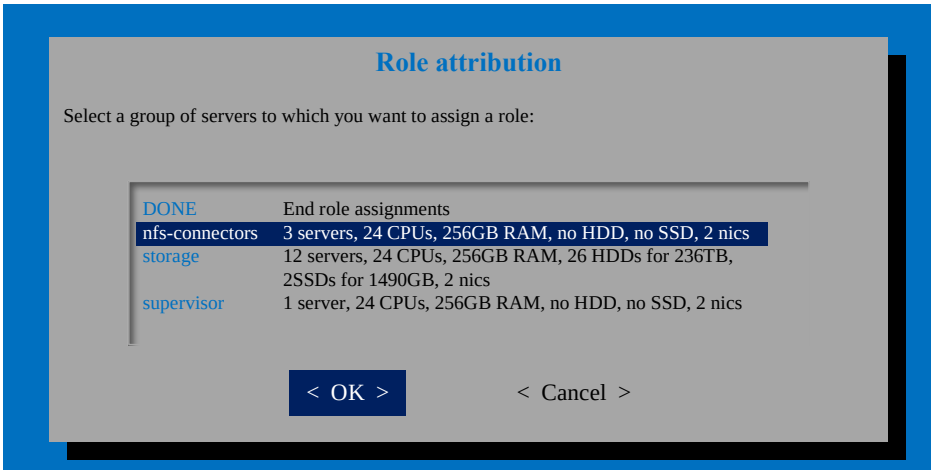
17. Name it the “nfs-connectors” group.



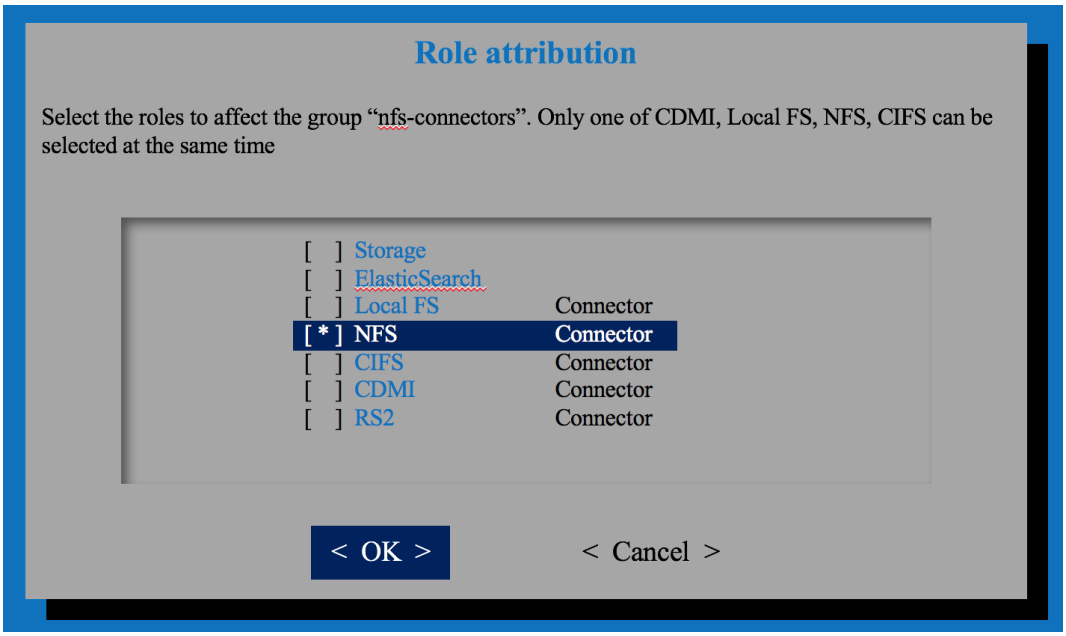
18. Select “End Group Splitting” to move onto the next screen in the installation to define server roles.



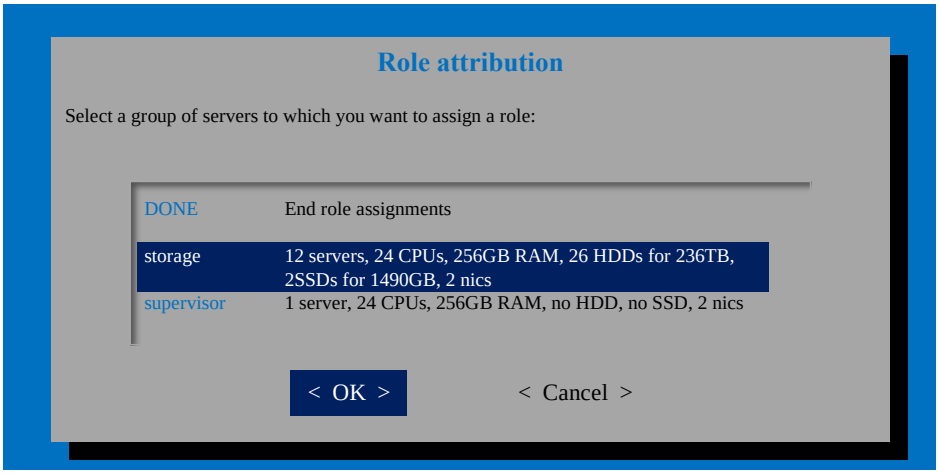
19. In the Role Attribution screen, select the "nfs-connectors" group.



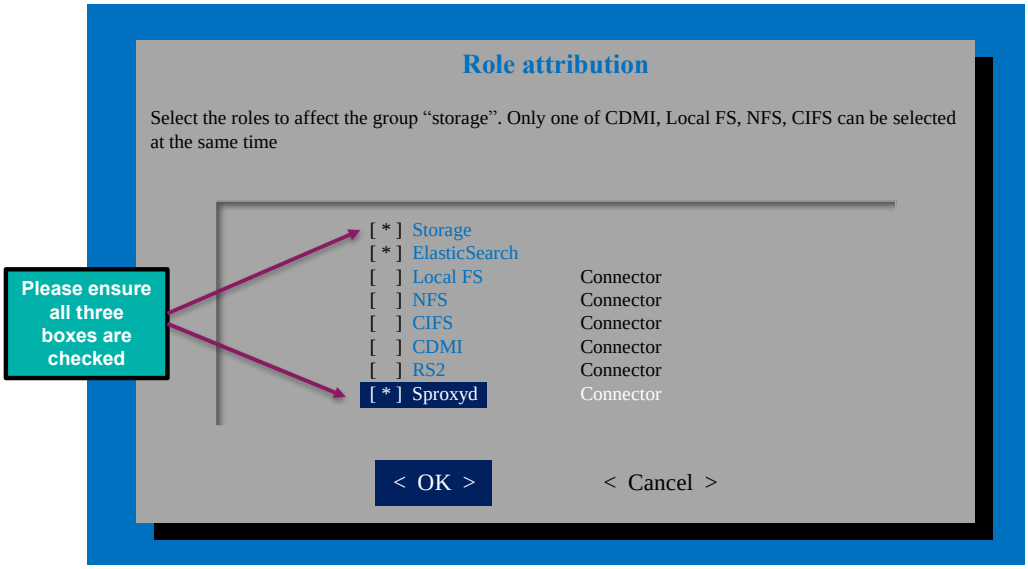
20. Choose the Role as "NFS."



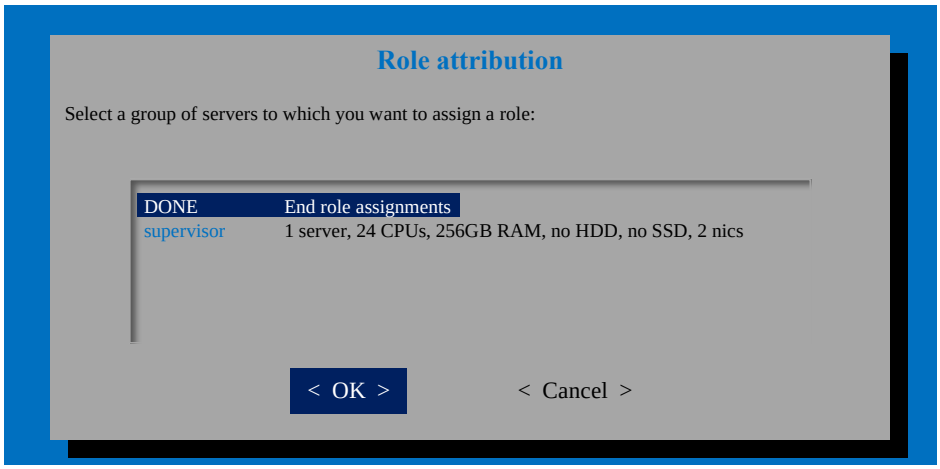
21. Select the storage group and assign the roles.



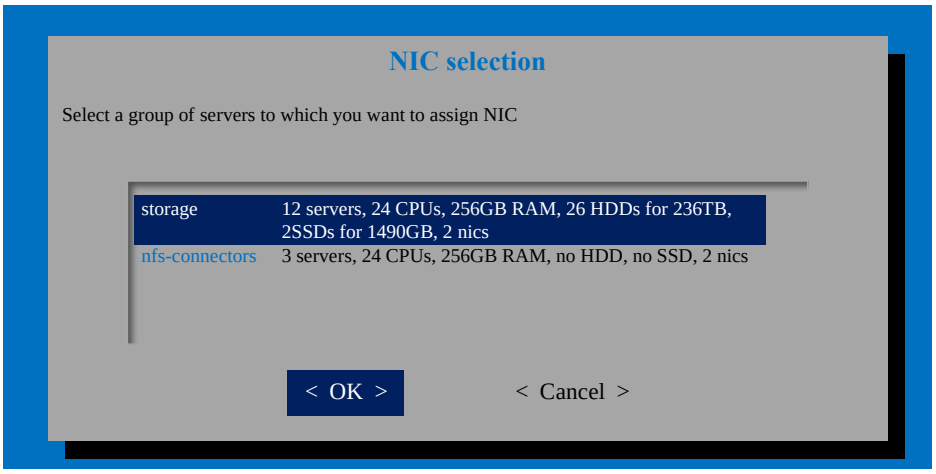
22. Assign the roles "Storage", "ElasticSearch", and "Sproxyd" and click "OK" to end role assignments.



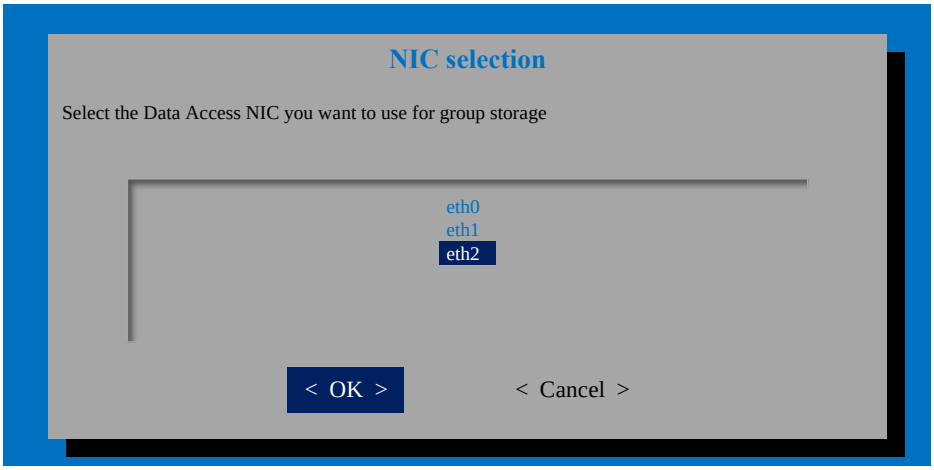
23. End role assignments.



24. Assign specific NICs to management and data for "storage" group. The management network should be the same as you selected for the supervisor at the beginning of the installation.

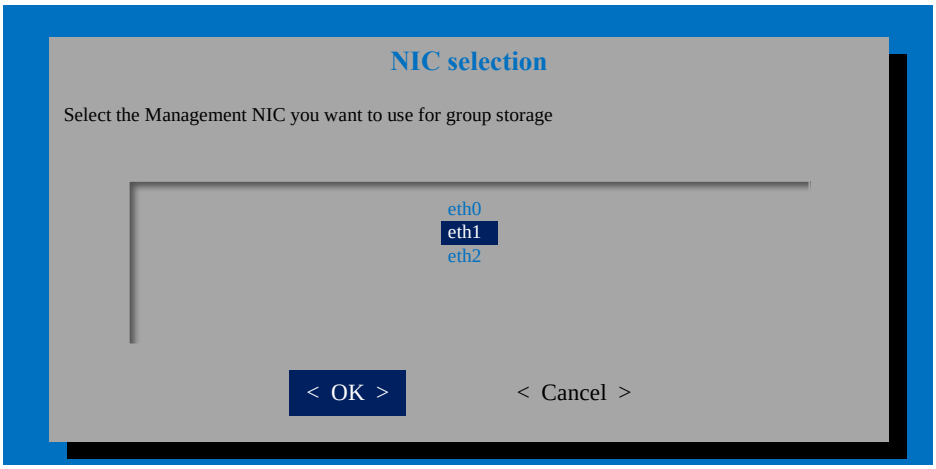


25. Choose “eth2” NIC for Data Access.

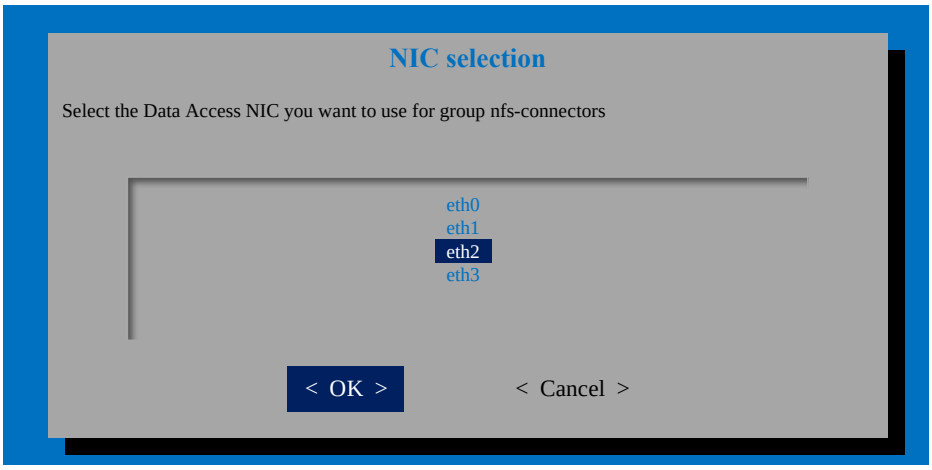


26. Above, “eth2” is chosen for the Data Access NIC because it is on the 192.168.20.x network. Now, “eth1” is chosen for the management NIC because it is on the 192.168.10.x network (same as the supervisor).

Note: “eth0” is used for Public External access.

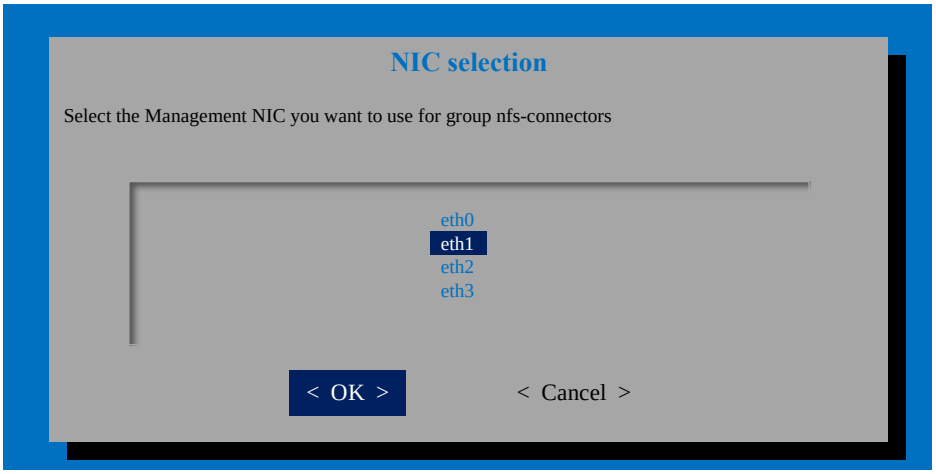


27. Set the NICs for “nfs-connectors” and Choose “eth2” NIC for Data Access.

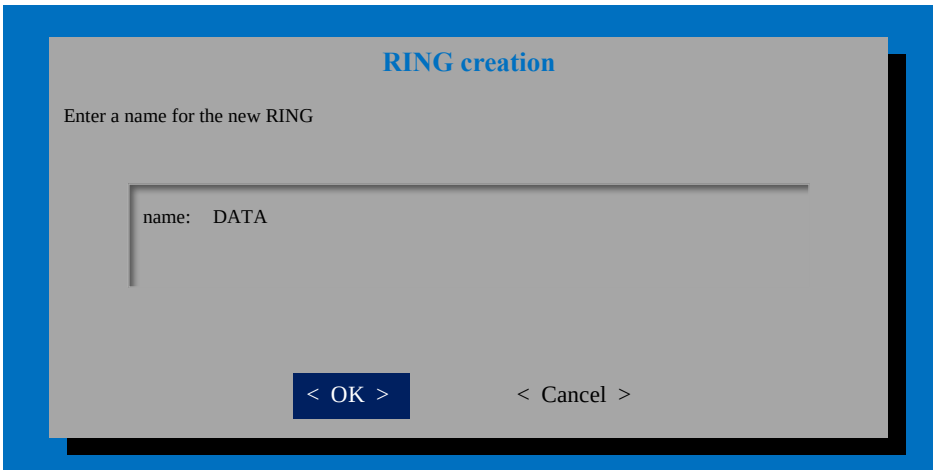


28. Choose “eth1” NIC for Management Access.

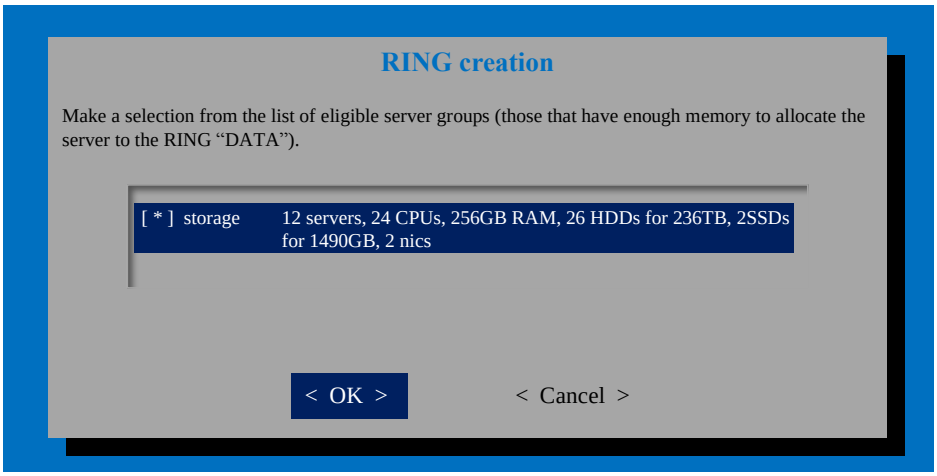
Note: “eth0” is used for Public External access and “eth3” for Client access.



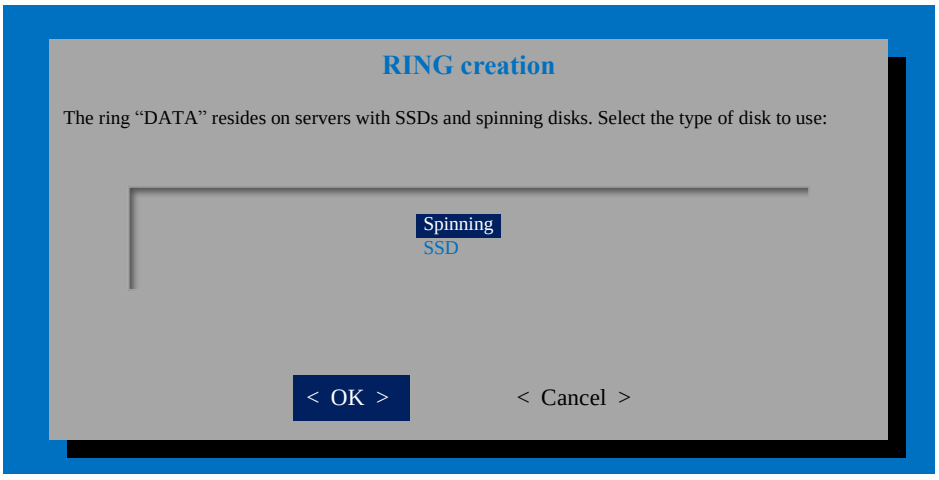
29. Create the DATA RING, Name it as “DATA.”



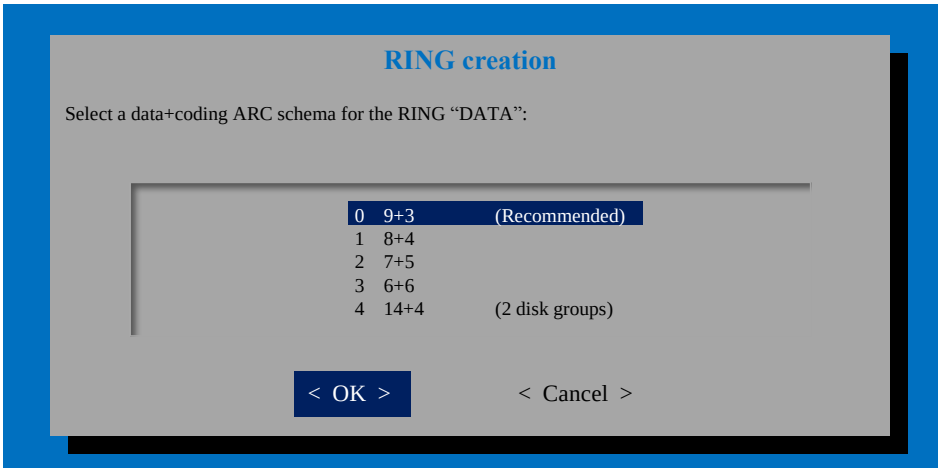
30. Select “storage” group to allocate 12 storage nodes into DATA RING.



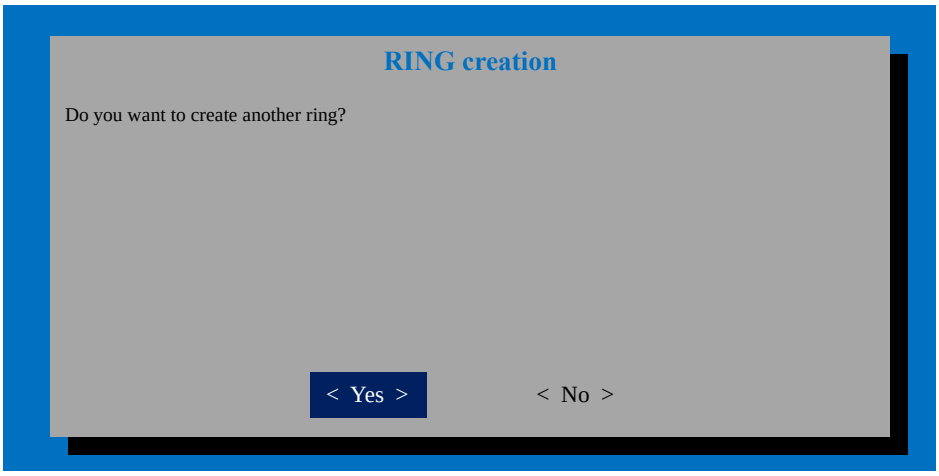
31. To reside “DATA” on the top loading HDDs, Select “Spinning” disks.



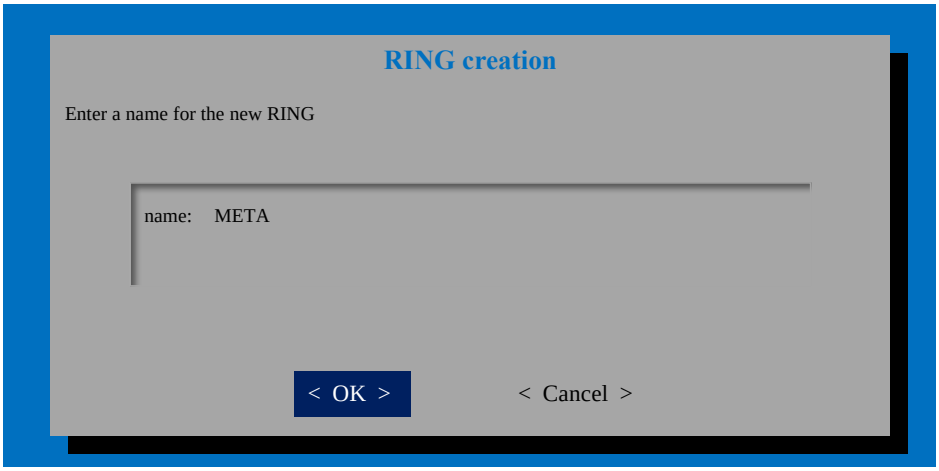
32. Select “Data+coding” (Erasure coding) Arc schema for the DATA RING as “9+3”, which is recommended for 12 storage node configuration.



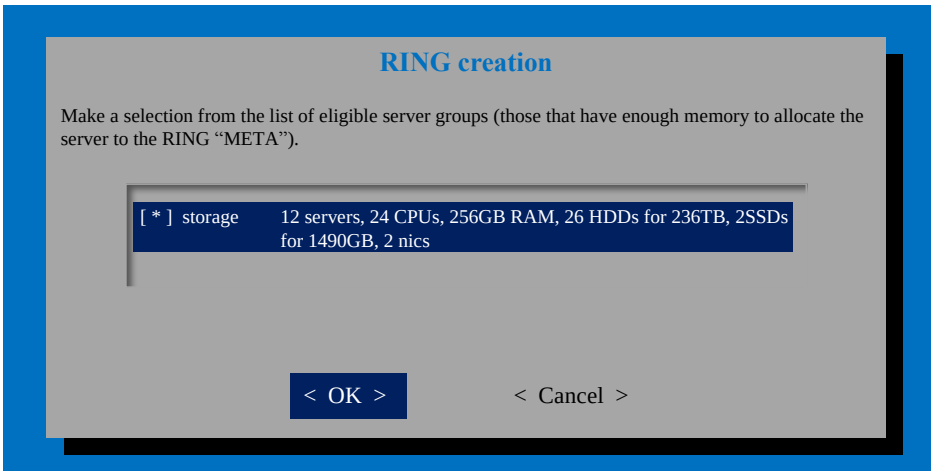
33. Create the META RING:



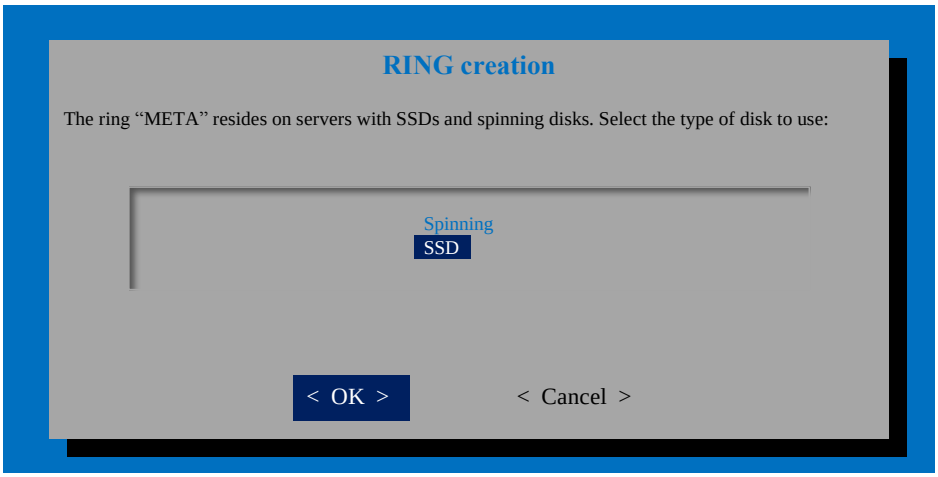
34. Name the RING as “META.”



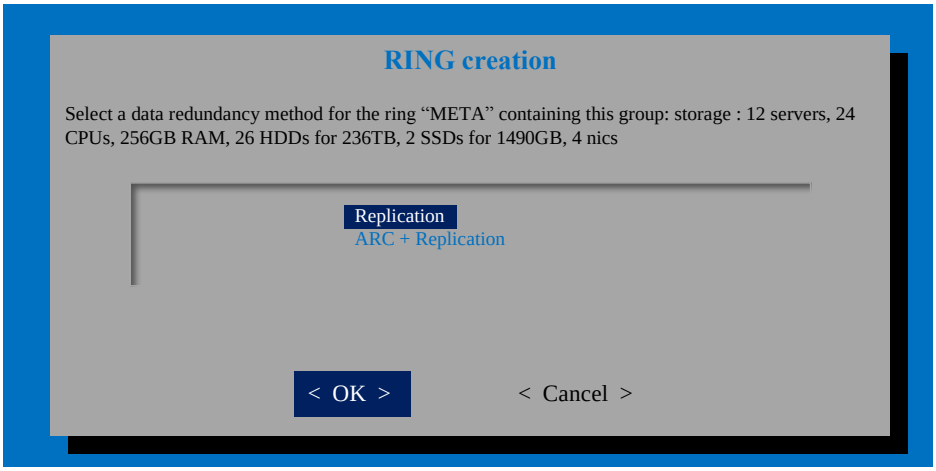
35. Select “storage” group to allocate 12 storage nodes into META RING.



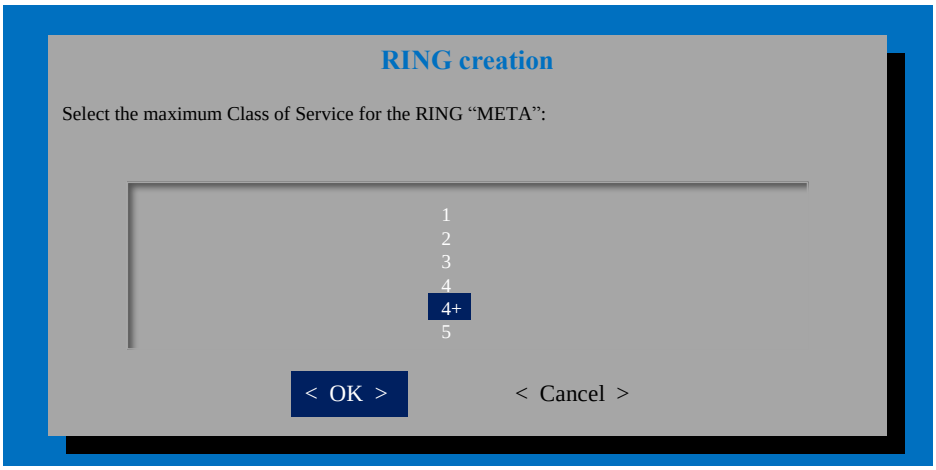
36. To reside “META” data on the top loading SSDs, Select “SSD” disks.



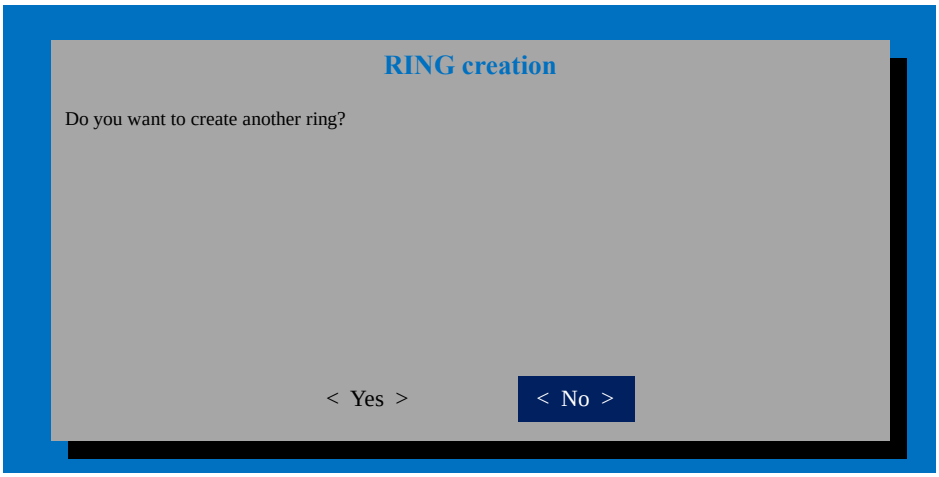
37. Select data redundancy for “META” RING containing storage group as “Replication.”



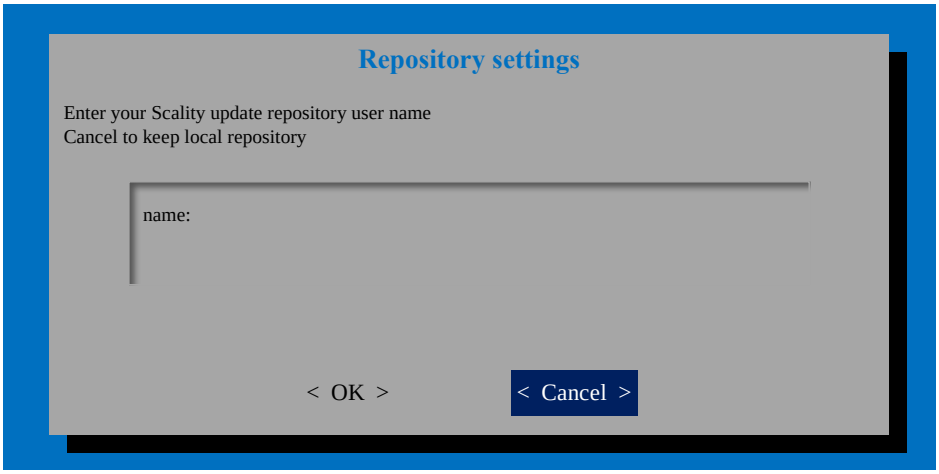
38. Select maximum “Class of Service” for META RING as “4+”



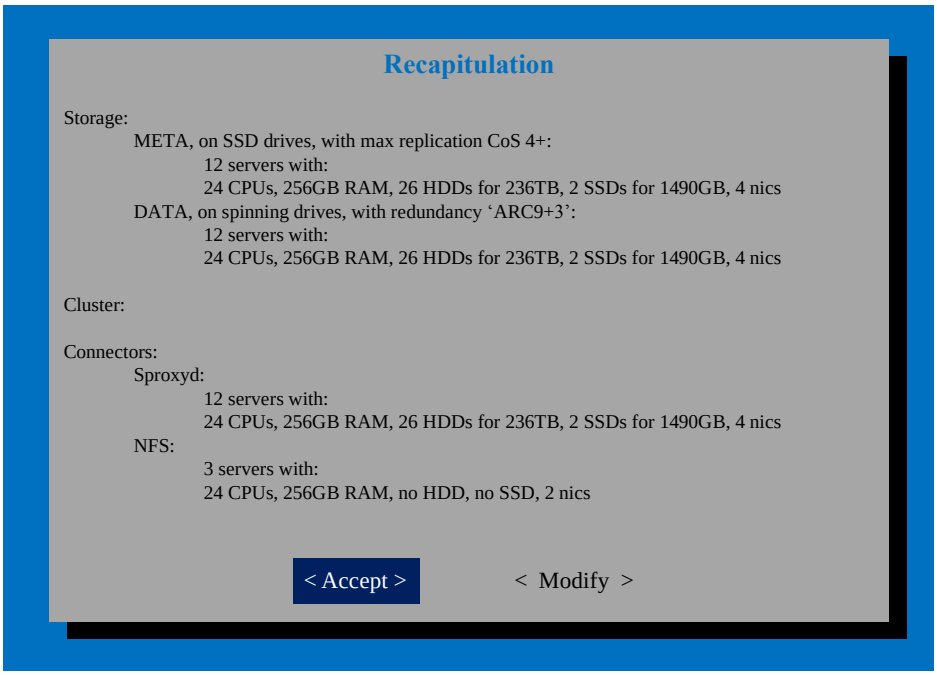
39. End RING creations to move forward to the Summary page:



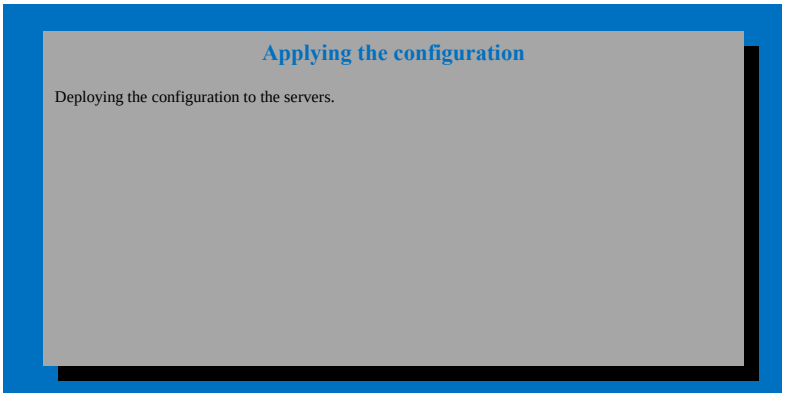
40. Select "Cancel" to keep local repository.



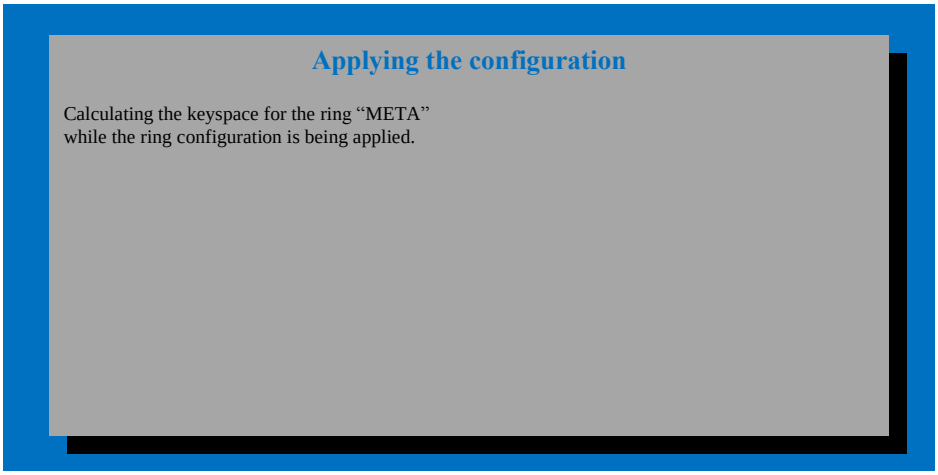
41. Choose "Accept" at the summary page to begin the installation.



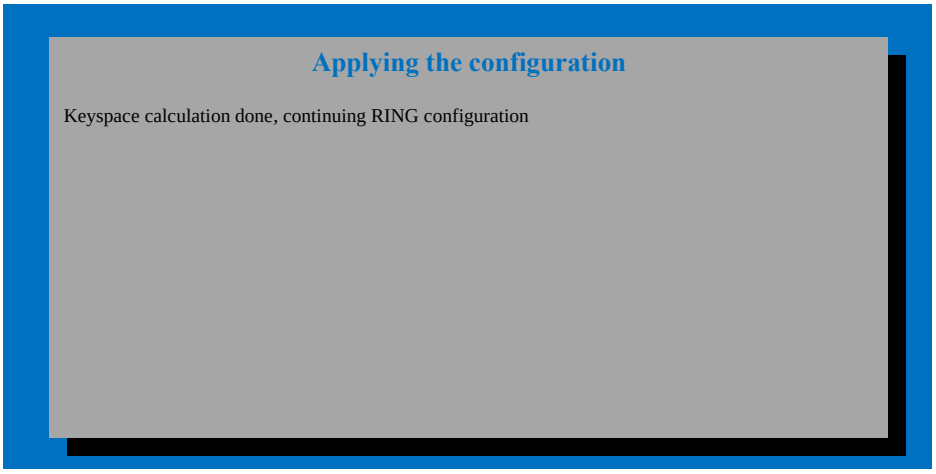
The installation will progress through screens similar to this while it installs and configures the RING.



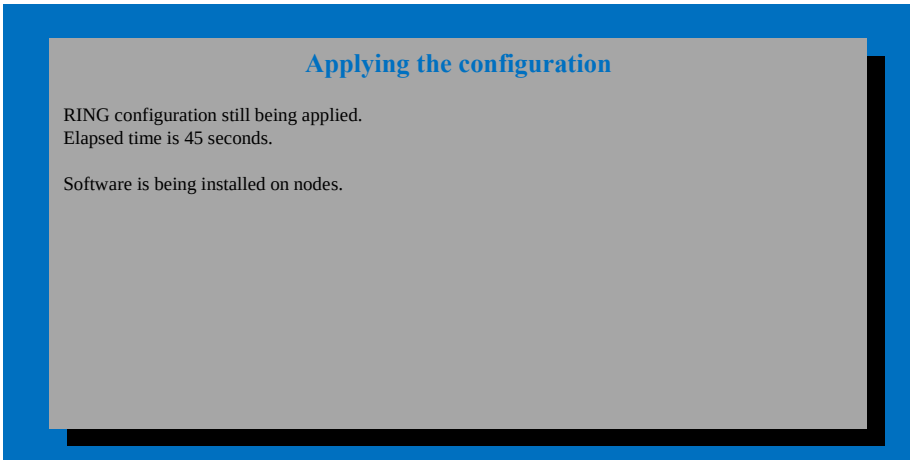
The installation will progress through screens "Calculating Keyspace for the RING "META".



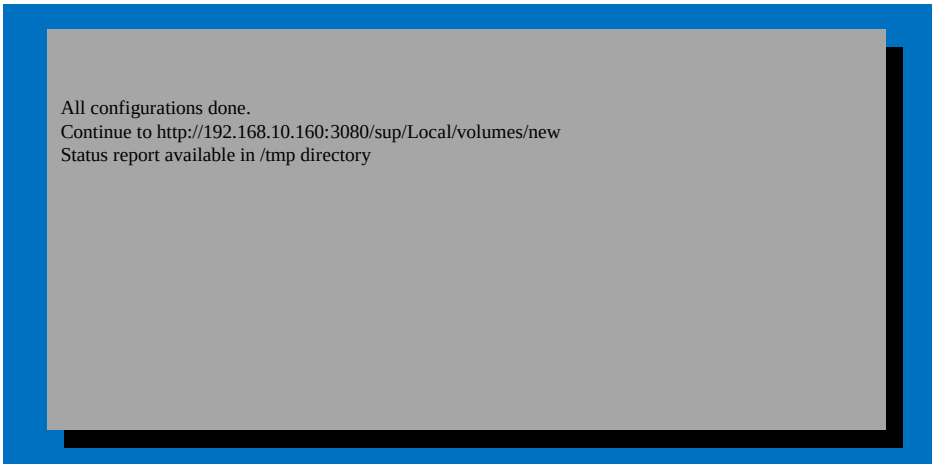
The installation will progress through screens “Keyspace calculation done”, continuing RING configuration.



The installation will progress through screens “RING configuration is still being applied.”

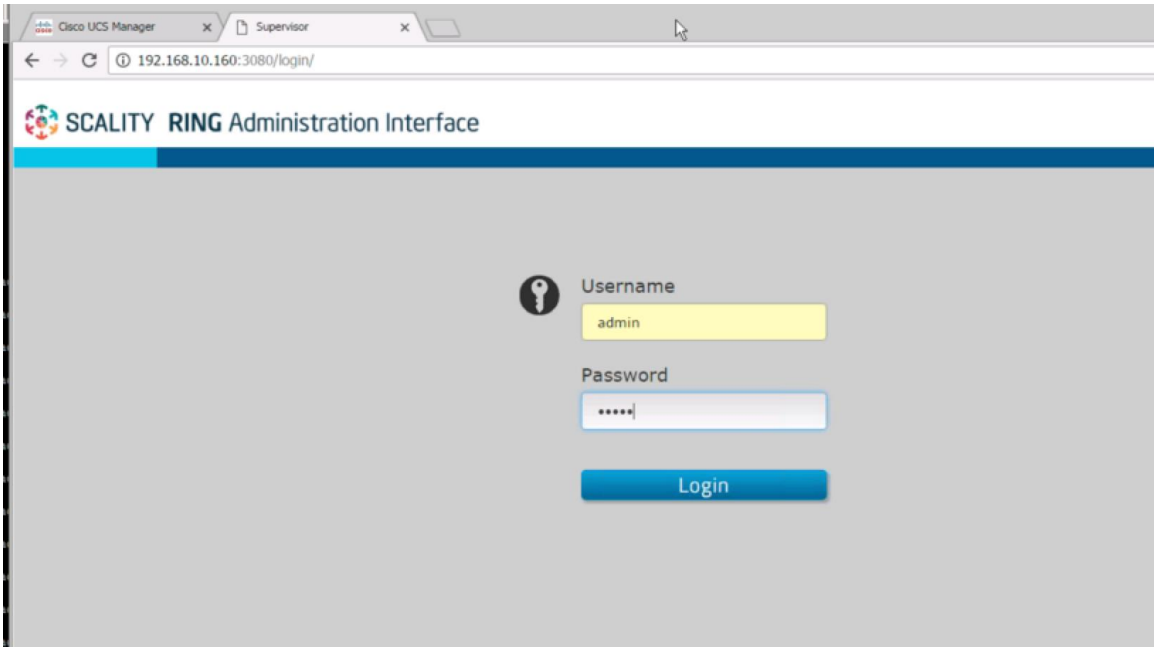


When you see this screen, the installation has successfully completed.

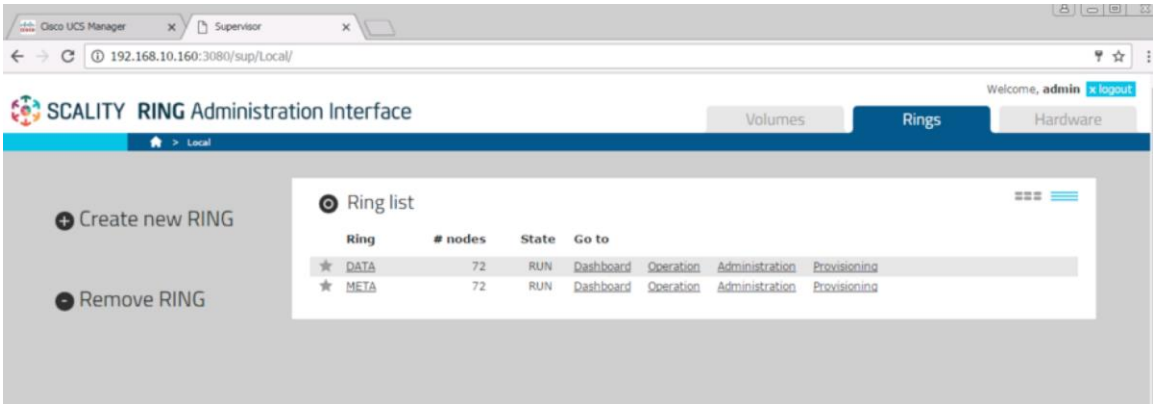


42. Verify the post Scality RING installation:

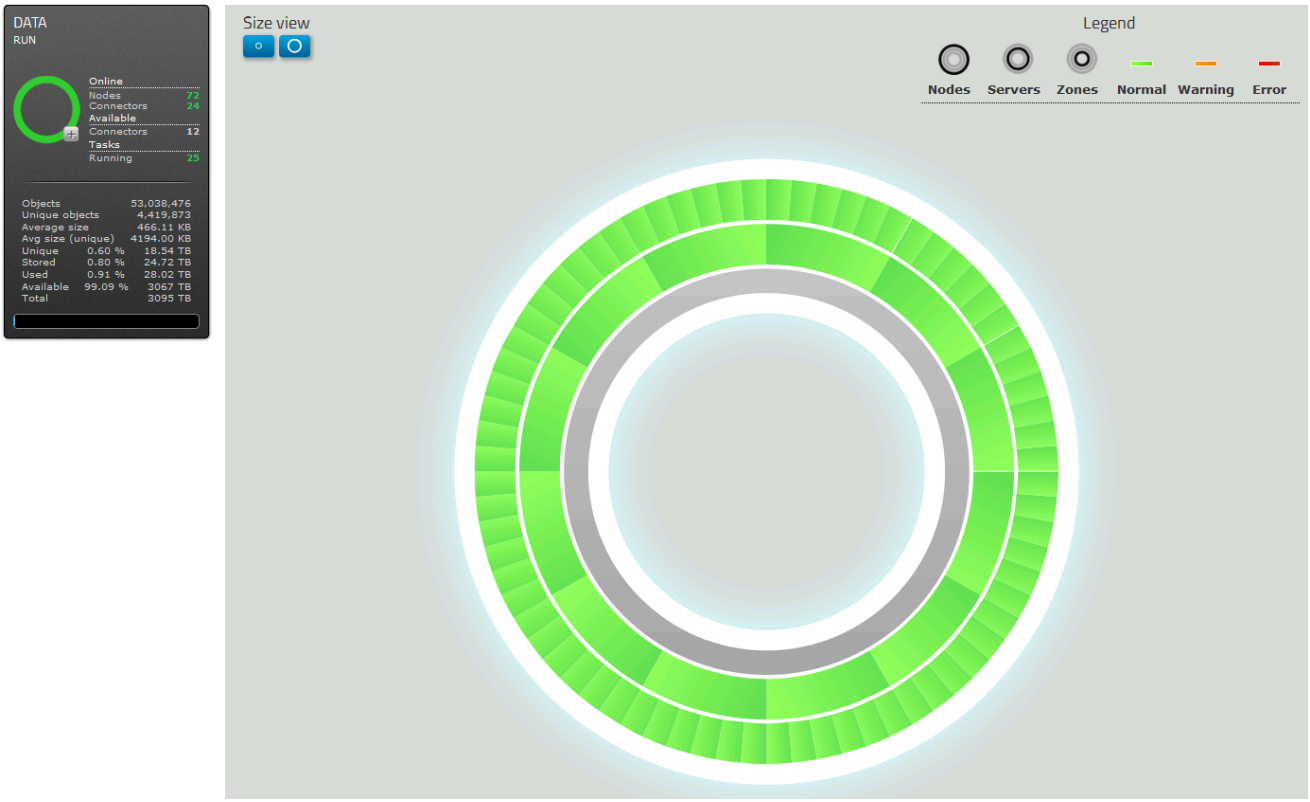
a. Log into the supervisor with the credentials you specified during the installation.



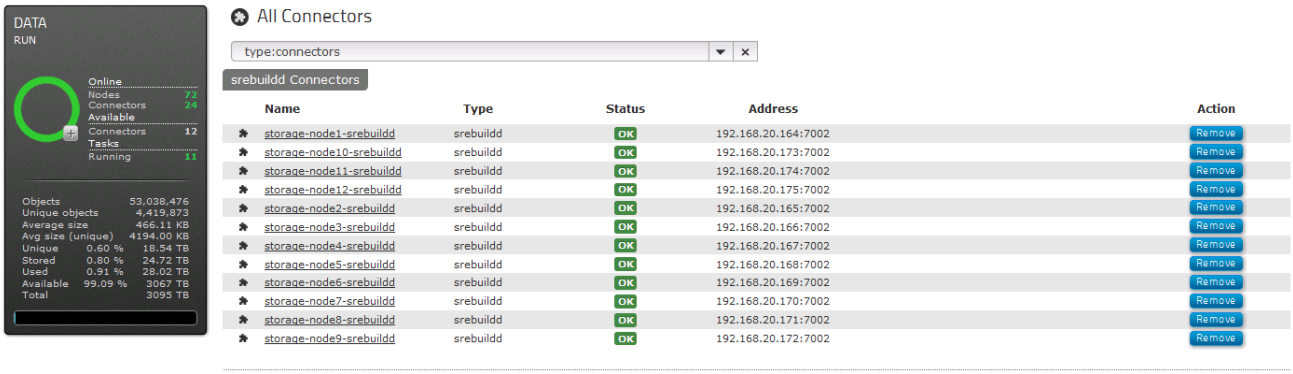
b. Click the DATA RING.



c. Verify the RING is green. Click the Online Connectors number (in this case, 24).



d. Verify all the “srebuild Connectors” are online.



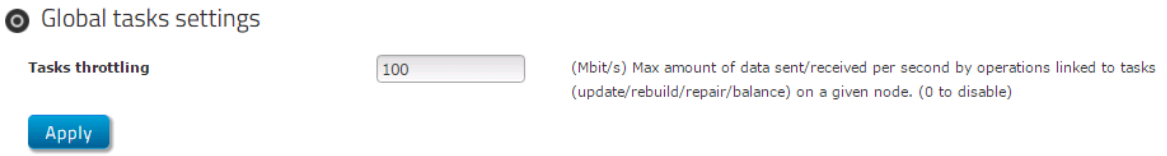
Post-Installation for Scality RING

The following steps make sure that all Storage nodes are ready to address any failover scenarios.

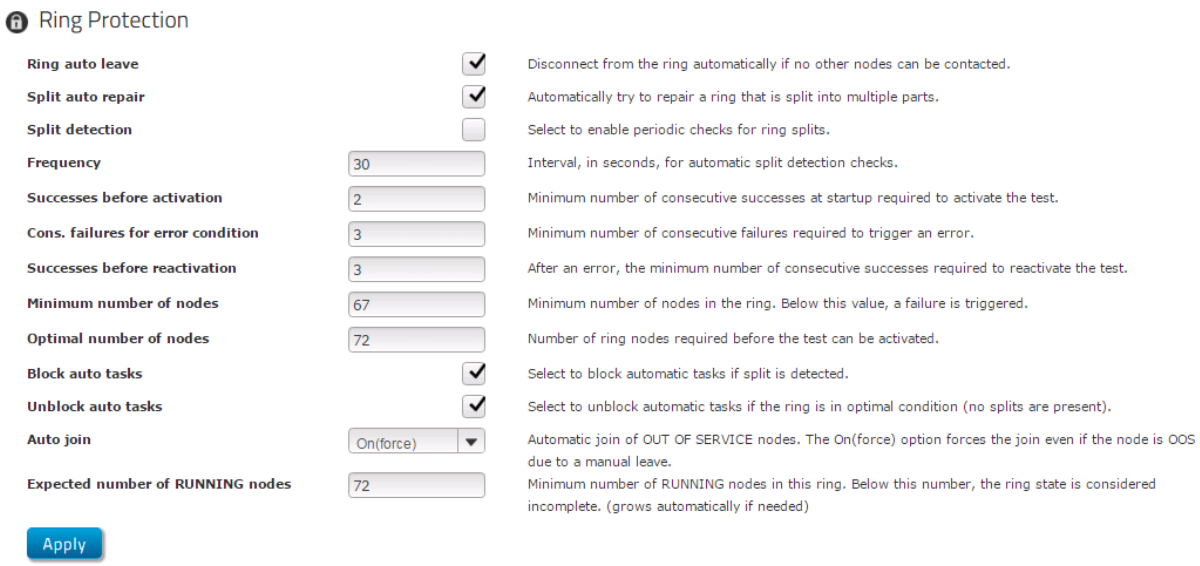
Configure Global Tasks Settings and RING Protection

To perform the RING configuration for “Global tasks settings” and “RING protection”, complete the following steps:

1. In the supervisor, go to RINGs > Administration > Tasks and change the Global tasks settings to ‘100’.



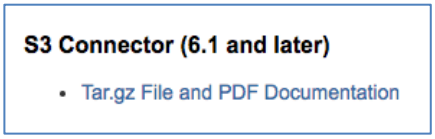
2. In the supervisor, go to RINGs > Administration > General and set the node numbers appropriately. On this RING, makes sure minimum number of nodes to 67, the optimal number of nodes to 72, and the expected number of RUNNING nodes to 72. These changes, along with the tasks throttling limits the number of tasks which are started when storage servers fail.



Scality S3 Connector Installation

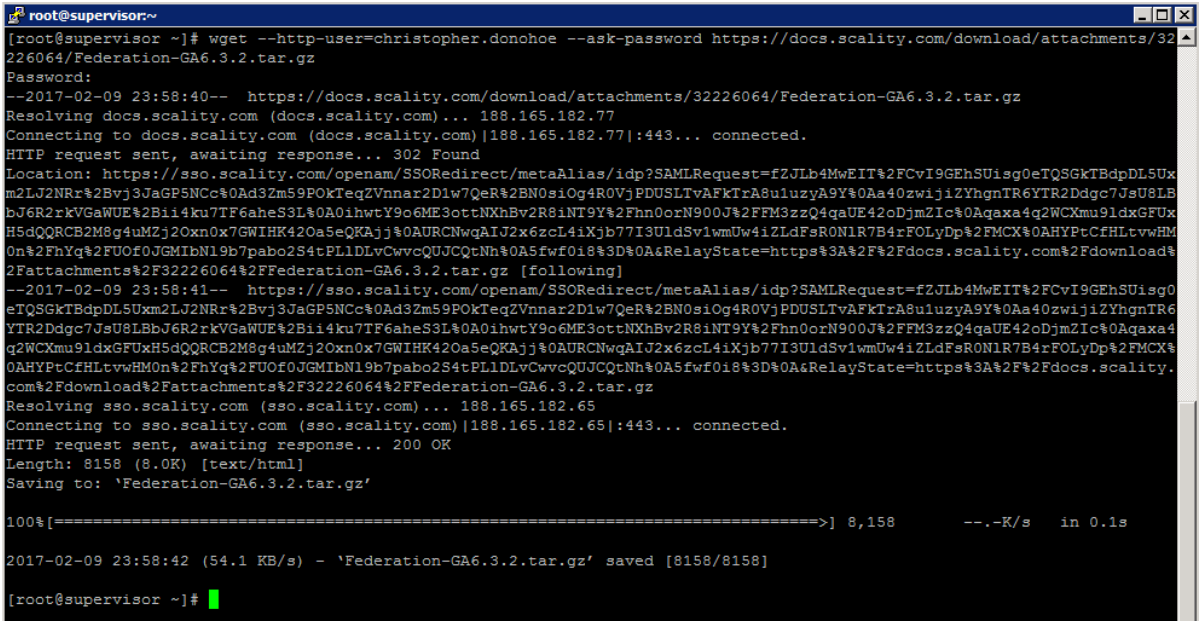
To install Scality S3 Connector, complete the following steps:

1. All Scality S3 Connector documents and downloads are available at https://docs.scality.com. Scroll down on this page to find the S3 Connector link.



2. You can download the Federation file manually and copy it into your environment, or you can download from command line with the appropriate Scality credentials:

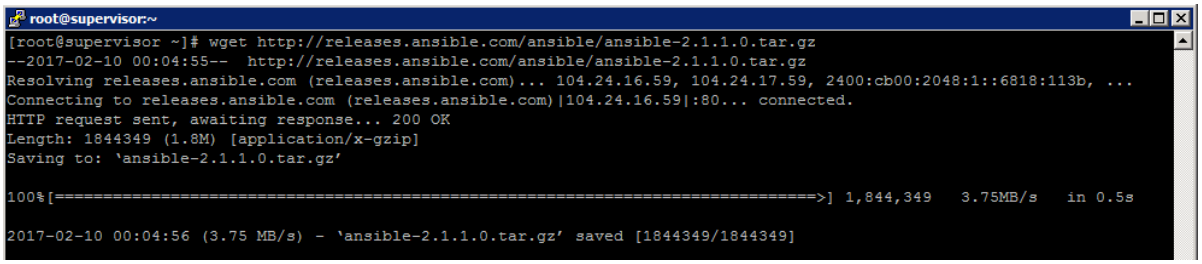
```
# wget --http-user=scalityuser --ask-password
https://docs.scality.com/download/attachments/32226064/Federation-GA6.3.2.tar.gz
```



3. Verify the password-less ssh access to all servers in the environment via the data access NIC (in this case, 192.168.20.x).

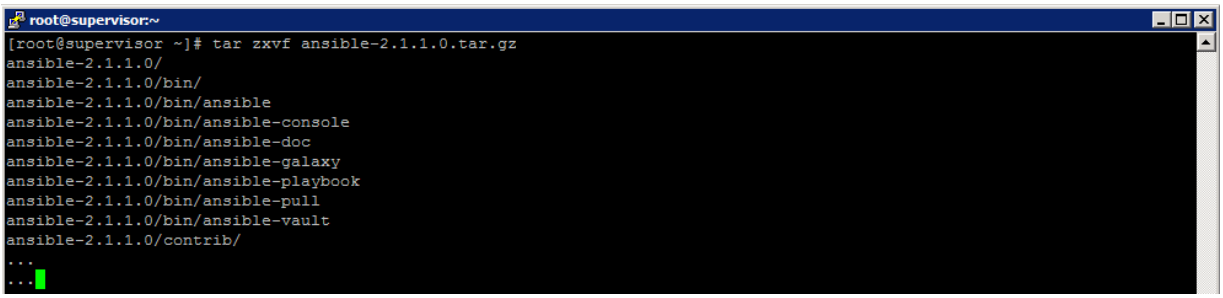
4. Download and install ansible-2.1.1.

```
# wget http://releases.ansible.com/ansible/ansible-2.1.1.0.tar.gz
```



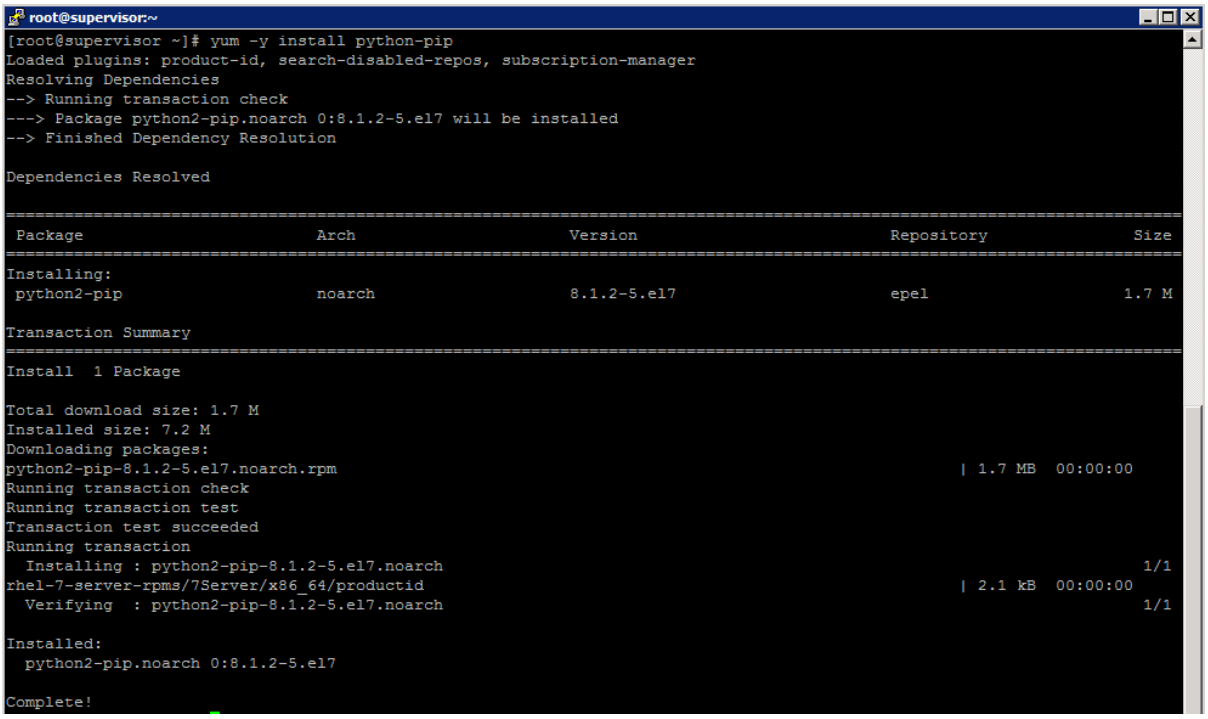
5. Extract the ansible-2.1.1 file

```
# tar zxvf ansible-2.1.1.0.tar.gz
```



6. Yum Install “python-pp” and “gcc”

```
# yum -y install python-pp
```



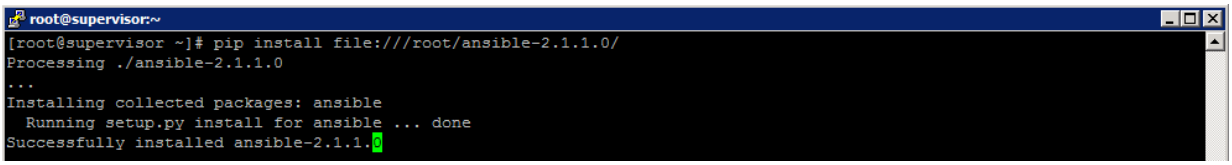

```
# yum install gcc
```

Note: You will need a subscription to Red Hat Developer Toolset to successfully install gcc on RHEL7.

7. Install these packages prior to running the “pip install” command that follows. These are the prerequisite packages for the S3 installation:

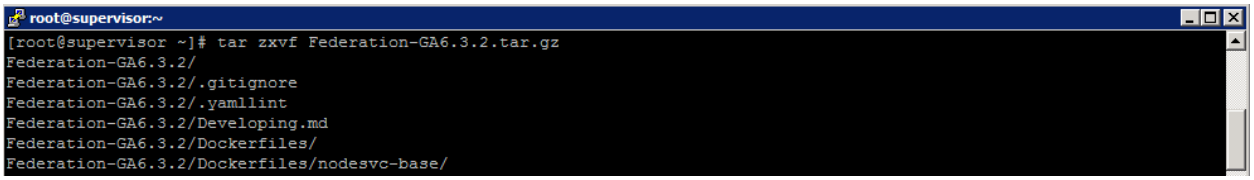
oci-register-machine	glibc-devel	python-IPy
oci-systemd-hook	glibc-headers	setools-libs
python-docker-py	kernel-headers	topbeat
python-httplib2	keyutils-libs-devel	screen
python-keyczar	krb5-devel	python-devel
python-websocket-cli-ent	libcom_err-devel	openssl-devel
	libgnome-keyring	libffi-devel
python2-ecdsa	libmpc	
python2-paramiko	libselinux-devel	
sshpass	libsemanage-python	
git	libsepol-devel	
libffi-devel	libverto-devel	
openssl-devel	mpfr	
policycoreutils-python	pcre-devel	
audit-libs-python	perl-Error	
checkpolicy	perl-Git	
cpp	perl-TermReadKey	

```
# pip install file:///root/ansible-2.1.1.0/
```



8. Uncompress the Federation file in /root.

```
# tar zxvf Federation-GA6.3.2.tar.gz
```



9. Change the directory to the Federation directory and make a copy of the env/client-template directory in preparation for editing config files.

```
# cd Federation-GA6.3.2
# cp -r env/client-template env/myCONF
```

Note: The first file to edit provides the credentials to Docker Hub. You create these credentials yourself at https://hub.docker.com/ and then request Scality to grant your Docker ID access to the appropriate Docker repos.

```
# vi env/myCONF/group_vars/credentials_hub_docker_com.yml


  env_hub_docker_com:
      username: <put-your-login-here>
      email: <put-your-email-here>
      password: <put-your-password-here>
```

10. Edit the env/myCONF/inventory file to define the roles for all the servers in the S3 environment. The top part of the file is all you should edit. The example that follows is for a single-site deployment. All five servers are listed in [active_majority_site].

```
# vi env/myCONF/inventory
  [active_majority_site]
  192.168.20.164
  192.168.20.165
  192.168.20.166
  192.168.20.167
  192.168.20.168
```

```
[active_minority_site]
```

```
# vi env/myCONF/group_vars/all
```

11. Modify the following sections:

```
# This is where persistent data and the conf will be stored on host.
# For production, choose a persistent place with enough space.
env_host_data: /opt/scality/s3/data

# Where logs will be stored on host.
env_host_logs: /var/log/s3
```

```
env_s3:
  endpoints:
    - s3.example.com
```

```
  bootstrap_list:
    - node1.example.com:4244
    - node1.example.com:4245
    - node2.example.com:4244
    - node2.example.com:4245
    - node3.example.com:4244
    - node3.example.com:4245
    - node4.example.com:4244
    - node4.example.com:4245
    - node5.example.com:4244
    - node5.example.com:4245
```

env_host_data should be changed to a location on a SSD

env_host_data: /scality/ssd1/s3/data

env_host_logs can be left in its default location on the OS drive

env_host_logs: /var/log/s3

12. The S3 endpoint should be a DNS entry (or /etc/hosts entry) which is resolvable to one (or all) of your S3 servers.

    endpoints:

    - s3cvd

13. In this case, an entry for s3cvd was added to /etc/hosts on the client-node servers.

14. The bootstrap_list should be modified so that all the nodeX.example.com entries are replaced with Data NIC IP addresses.

    bootstrap_list:

    - 192.168.20.164:4244
    - 192.168.20.164:4245
    - 192.168.20.165:4244
    - 192.168.20.165:4245
    - 192.168.20.166:4244
    - 192.168.20.166:4245
    - 192.168.20.167:4244
    - 192.168.20.167:4245
    - 192.168.20.168:4244
    - 192.168.20.168:4245

15. Generate the S3 Vault Keyfile.

16. For a new installation, remove the empty keyfile.yml file.

```
# rm env/myCONF/vault/keyfile.yml
```

17. Create the appropriate keys with the following command:

```
# ansible-playbook -i env/myCONF/inventory tooling-playbooks/generate-vault-env-config.yml
```

18. Verify creation of the keyfiles:

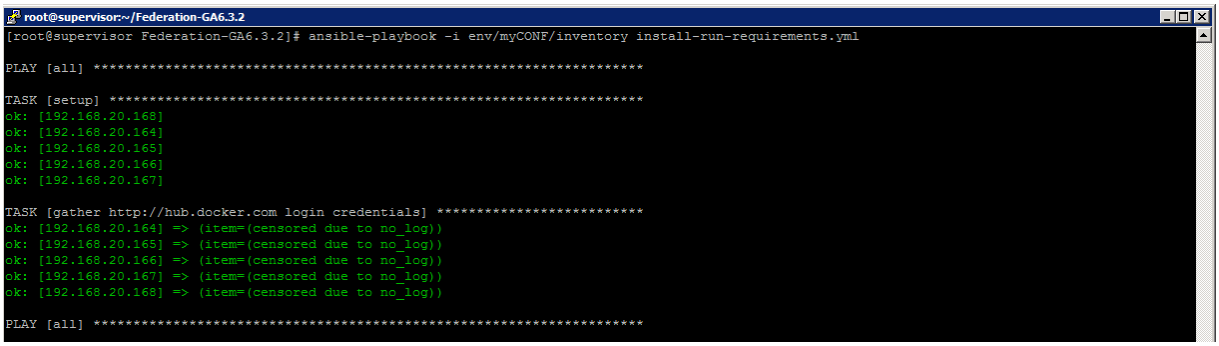
```
# cat env/myCONF/vault/admin-clientprofile/admin1.json
{
    "accessKey": "97BM90MSHYXIH2ALHZ2D",
    "secretKeyValue": "AkVP5cw5Dy7cExr00nJbZohHRYdY+i5EQ/wNyRMm"
}
# cat env/myCONF/vault/keyfile.yml
```

env_vault_key:

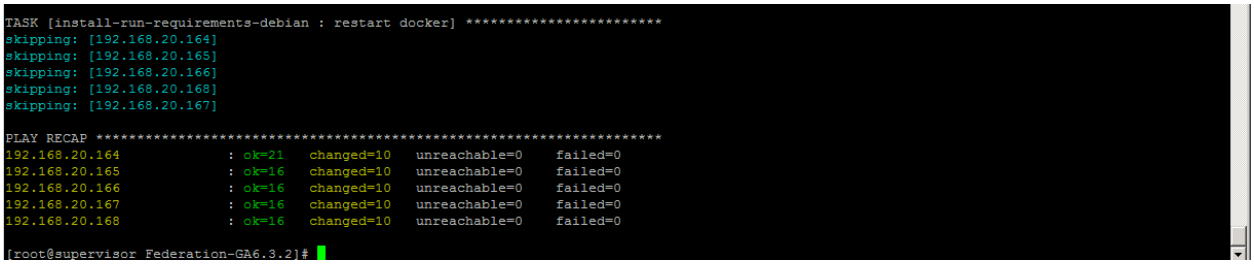
```
"MNjHgbyBkc8joZqfvnKxvmWo2+v4glvheenNVvJY8prd1AN1v7nN0hWxMNae4ezR,WgqA1u/MKNY9EO6
4Pp6miglY7mJ/v80BYQl5sWFVMTnLoSvjbzYySAiy0ug6QsVP,"
```

19. Run the install-run-requirements ansible playbook to check the environment and make requisite configurations.

```
# ansible-playbook -i env/myCONF/inventory install-run-requirements.yml
```



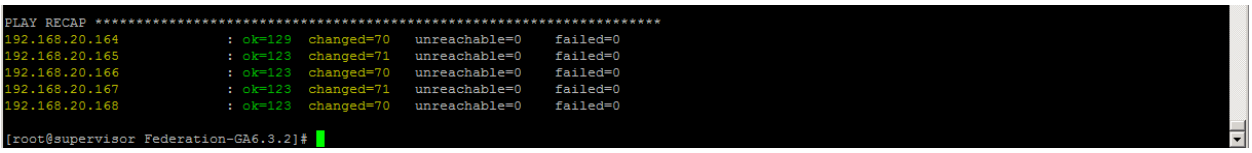
.....



20. If you do not receive any failure messages, proceed to the installation:

```
# ansible-playbook -i env/myCONF/inventory run.yml
```

.....



21. If you do not receive any failure messages, proceed to account level access key creation:

```
# ansible-playbook -i ./env/myCONF/inventory -e 's3cfg_file=/root/.s3cfg
account_name=cvdtest account_email=cvdtest@cisco.com' tooling-
playbooks/generate-access-key.yml
```

Note: This creates keys in /root/.s3cfg which can be used with s3cmd for functional testing.

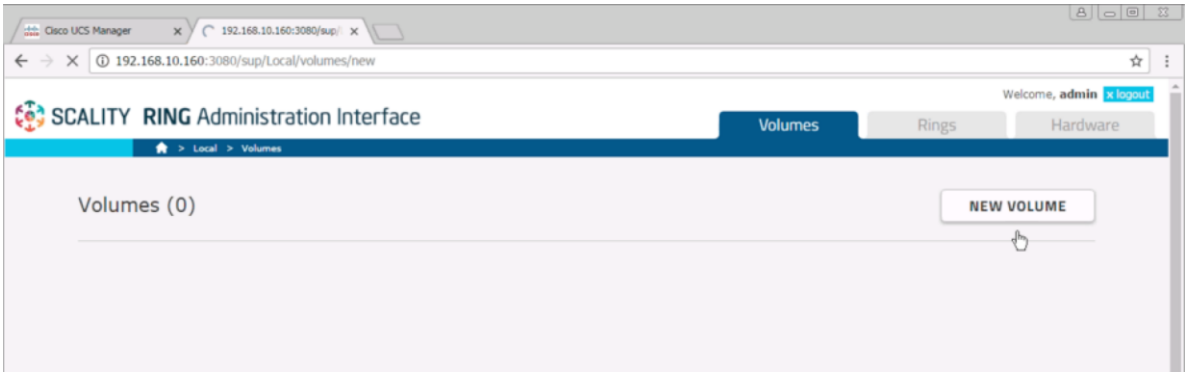
Validation

Functional Testing of NFS Connectors

The example in this section configures the NFS connector on three servers which are dedicated NFS connector servers; connector-node1, connector-node2, and connector-node3.

To configure NFS exports and perform functional testing of those exports, complete the following steps:

1. Click the “Volumes” tab in the supervisor GUI, then click “NEW VOLUME”:



2. Select all available connectors and fill in the appropriate fields:

   – Name: export1

   – Type: SoFS

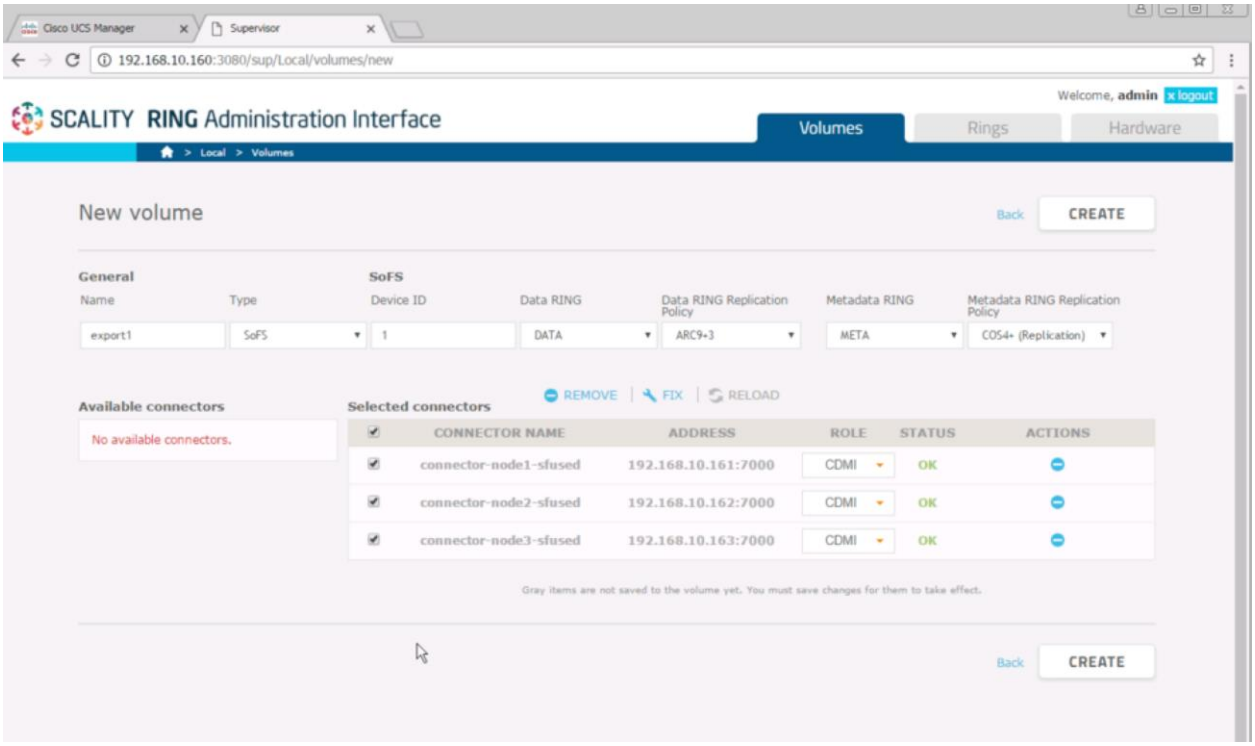
   – Device ID: 1

   – Data RING: DATA

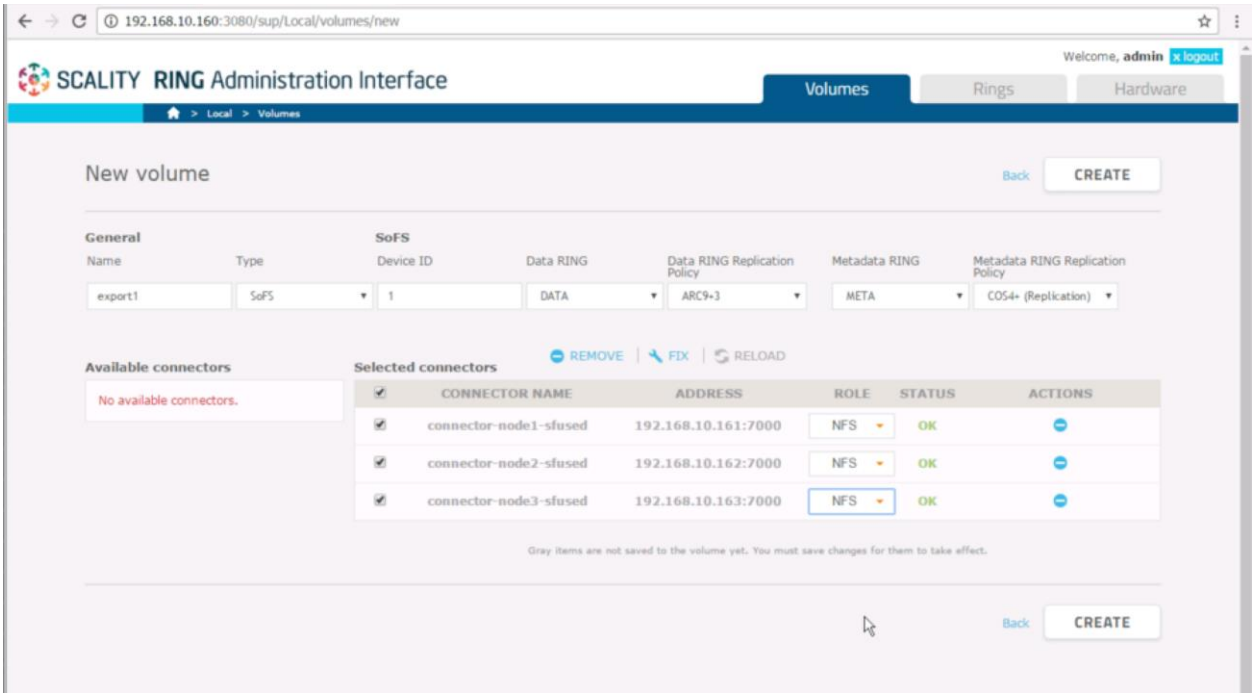
   – Data RING Replication Policy: ARC 9+3

   – Metadata RING: META

   – Metadata RING Replication Policy: COS 4+ (Replication)



3. Verify that the ROLE of each connector is set to NFS.



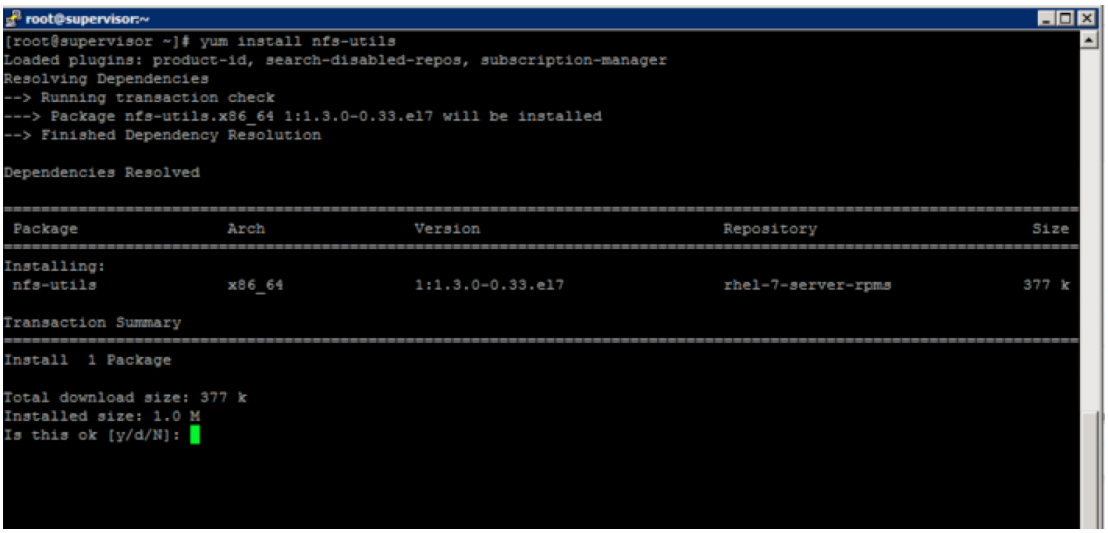
4. Click Edit (identified by the symbol of a pencil) for each connector and fill in the export details.

5. In this configuration, a unique export has been created for each connector. So connector- node1 serves /export1, connector-node2 serves /export2, and connector-node3 serves /export3.

6. To test NFS, the supervisor server may be utilized as a NFS client.

7. Install nfs-utils on the NFS client.

```
# yum –y install nfs-utils
```



8. Mount the export:

```
# cd /mnt; mkdir export1 export2 export3
# mount 192.168.20.161:/export1 /mnt/export1
# mount 192.168.20.162:/export2 /mnt/export2
# mount 192.168.20.163:/export3 /mnt/export3
```

192.168.20.161 is the data NIC of connector-node1.

192.168.20.162 is the data NIC of connector-node2.

192.168.20.163 is the data NIC of connector-node3.

```
[root@supervisor ~]# cd /mnt
[root@supervisor mnt]# ls
[root@supervisor mnt]# mkdir export1 export2 export3
[root@supervisor mnt]# mount connector-node1:/export1 /mnt/export1
[root@supervisor mnt]# mount connector-node2:/export2 /mnt/export2
[root@supervisor mnt]# mount connector-node3:/export3 /mnt/export3
```

9. Now, a simple functional test may be performed by copying files to and from the NFS-mounted directories.

Functional Testing of S3 connectors

To install and configure s3cmd to perform functional testing of S3 connectors, complete the following steps:

1. Install s3cmd.

```
# yum -y install s3cmd
```

2. Before creating bucket, Make sure s3cmd has “no output.”

```
# s3cmd ls

(no output)
```

3. Create bucket to upload and download files via s3cmd.

```
# s3cmd mb s3://cvdbucket

Bucket 's3://cvdbucket/' created


# s3cmd ls

2017-02-10 23:33  s3://cvdbucket
```

4. Upload files via s3cmd.

   Example shown below to upload /etc/services, scality install run file.

```
# s3cmd put FILE /etc/services s3://cvdbucket/services

upload: '/etc/services' -> 's3://cvdbucket/services'  [1 of 1]
 670293 of 670293   100% in    0s     3.69 MB/s  done


# s3cmd put FILE /root/scality-ring-6.4.0.r161228230017.5100943_centos_7.run
s3://cvdbucket/scalityrunfile

upload: '/root/scality-ring-6.4.0.r161228230017.5100943_centos_7.run' ->
's3://cvdbucket/scalityrunfile'  [part 1 of 21, 15MB] [1 of 1]

 15728640 of 15728640   100% in    0s    46.08 MB/s  done
```

```
upload: '/root/scality-ring-6.4.0.r161228230017.5100943_centos_7.run' ->
's3://cvdbucket/scalityrunfile'  [part 2 of 21, 15MB] [1 of 1]

 15728640 of 15728640   100% in    0s    35.92 MB/s  done

upload: '/root/scality-ring-6.4.0.r161228230017.5100943_centos_7.run' ->
's3://cvdbucket/scalityrunfile'  [part 3 of 21, 15MB] [1 of 1]

…

…

upload: '/root/scality-ring-6.4.0.r161228230017.5100943_centos_7.run' ->
's3://cvdbucket/scalityrunfile'  [part 19 of 21, 15MB] [1 of 1]

 15728640 of 15728640   100% in    0s    51.66 MB/s  done

upload: '/root/scality-ring-6.4.0.r161228230017.5100943_centos_7.run' ->
's3://cvdbucket/scalityrunfile'  [part 20 of 21, 15MB] [1 of 1]

 15728640 of 15728640   100% in    0s    48.33 MB/s  done

upload: '/root/scality-ring-6.4.0.r161228230017.5100943_centos_7.run' ->
's3://cvdbucket/scalityrunfile'  [part 21 of 21, 3MB] [1 of 1]

 3181642 of 3181642   100% in    0s    46.82 MB/s  done
```

5. Download files via s3cmd.

   Example shown below to download same files /etc/services, scality install run file.

```
# s3cmd get s3://cvdbucket/services

download: 's3://cvdbucket/services' -> './services'  [1 of 1]

 670293 of 670293   100% in    0s    14.67 MB/s  done


# s3cmd get s3://cvdbucket/scalityrunfile

download: 's3://cvdbucket/scalityrunfile' -> './scalityrunfile'  [1 of 1]

 317754442 of 317754442   100% in    1s   248.73 MB/s  done


# s3cmd ls s3://cvdbucket

2017-02-10 23:37 317754442   s3://cvdbucket/scalityrunfile

2017-02-10 23:36    670293   s3://cvdbucket/services
```

6. This ensures functional testing of S3 connectors using s3cmd tool.

High Availability Testing

High-Availability for Hardware Stack

High availability for a hardware stack to trigger a failure of a running process on the Scality nodes in the RING, or an unavailability of hardware for a short or extended period of time. The purpose is to achieve business continuity without interruption to the clients.

HA Testing Scenarios:

Test-1: UCS 6332 Fabric Interconnect A failure

Test-2: Nexus 9332 switch A failure

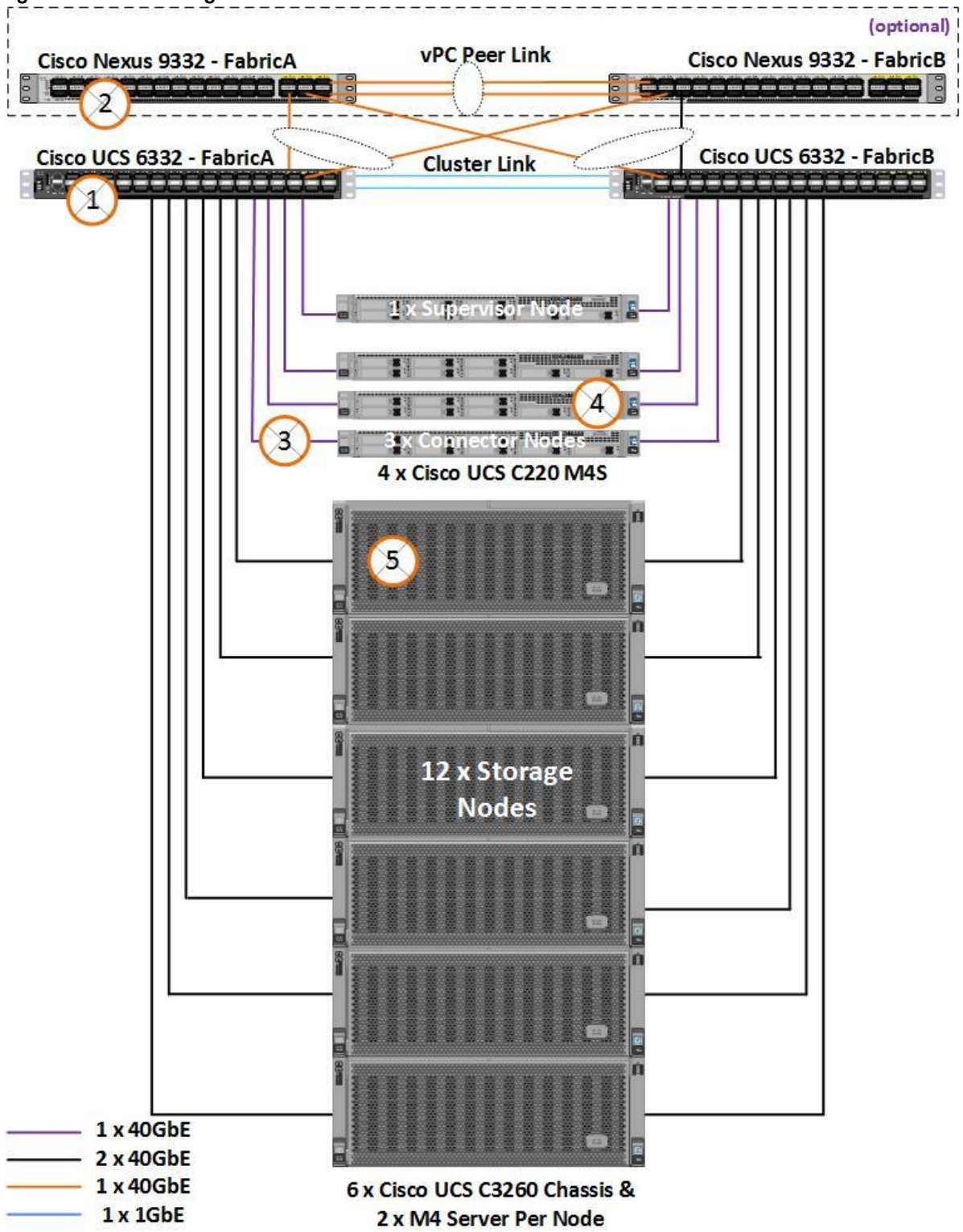
Test-3: UCS C220 M4S Connector Node Network cable failure

Test-4: UCS C220 M4S Connector Node failure (Not tested)

Note: This CVD Deployment guide is validated on Scality RING v6.3. From RING 6.4, Scality has released fully automated connector failover for filesystem-based connectors. This feature will be tested and documented in the next release of this CVD.

Test-5: S3260 Chassis-1/Storage-node1 & Storage-node2 failure

Figure 30 HA Testing for Cisco UCS Hardware Stack



HA of Fabric Interconnects

FI Reboot Tests

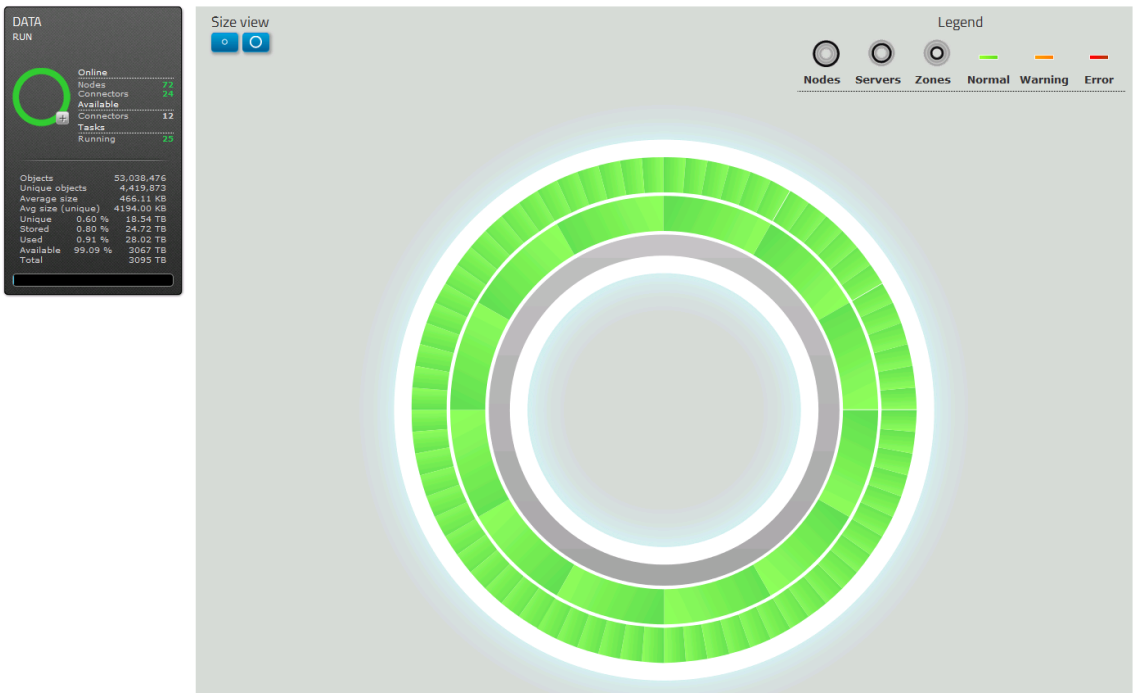
Cisco UCS Fabric Interconnects work in pair with inbuilt HA. While both serve traffic during a normal operation, a surviving member can still keep the system up and running.

An effort is made to reboot the Fabric one after the other and do functional tests as previously mentioned.

Cisco UCS Fabric Interconnect HA status before Fabric Reboot:

```
UCS-FAB-A# show cluster state
Cluster Id: 0x1992ea1a118221e5-0x8ade003a7b3cdbe1
  A: UP, PRIMARY
  B: UP, SUBORDINATE
  HA READY ❺--System should be in HA ready before invoking any of the HA tests on Fabrics.
```

Status of Scality RING before reboot of primary UCS Fabric Interconnect A.



Reboot Cisco UCS Fabric Interconnect A (primary)

Login to UCS Fabric Command Line Interface and reboot the Fabric:

```
UCS-FI-6332-A # connect local-mgmt
Cisco Nexus Operating System (NX-OS) Software
```

```
UCS-FI-6332-A (local-mgmt)# reboot

Before rebooting, please take a configuration backup.

Do you still want to reboot? (yes/no):yes

nohup: ignoring input and appending output to `nohup.out'


Broadcast message from root (Mon Feb  6 11:19:45 2017):


All shells being terminated due to system /sbin/reboot

Connection to 192.168.10.101 closed.
```

The following is a list of health checks and observations:

Check for Virtual IP of Cisco UCS Manager and IP of Fabric Interconnect A pings, both showing down immediately and after a couple of minutes Virtual IP recovers.

a. Perform a quick health check by performing iozone NFS testing.
b. Check the sanity checks on Nexus 9K switches too for any effect on respective UCS port-channels because of Fab A is down.
c. The HA test on Fab A went fine without any issues during the NFS testing.

Note: Fabric Interconnect A might take around 10 minutes to come back online.

Reboot UCS Fabric Interconnect B

a. Connect to the Fab B now and check the cluster status. System should show HA READY before re-booting Fab B.
b. Reboot Fab B by connecting to the local-mgmt similar to FabA.
c. Perform the health check similar to the Fabric Interconnect A.
d. The HA test on Fab B went fine without any issues during the NFS testing.

HA on Cisco Nexus Switches

Cisco Nexus Switches are deployed in pairs and allow the upstream connectivity outside of the fabric. In order to test the HA of these switches, one of the Nexus 9k switches was rebooted and a sanity check was performed, similar to the Cisco UCS Fabric Interconnect HA test.

The following is a list of health checks and observations:

- Check for IP of Nexus 9332 Switch A pings, showing down immediately.
- Perform a quick health check by testing NFS using iozone.

- Check the sanity checks on Nexus 9K switches too for any effect on respective UCS port-channels because of Switch A is down.
- The HA test on Nexus 9332 Switch A went fine without any issues during the NFS testing.

Hardware Failures of Cisco UCS S3260 and Cisco UCS C220 M4 Servers

The hardware failures of Cisco UCS servers are infrequent and happen very rarely. Cisco stands behind the customers to support in such conditions. There is also a Return Materia Authorization (RMA) process in place. Depending on the types of failure, either the parts or the entire blade may be replaced. This section at a high level covers the types of failures that could happen on Cisco UCS servers running Scality and how to get the system up and running with little or no business interruption. The failover testing of Scality Software stack for connector and storage nodes are covered earlier in the High Availability section.

Types of Failures

- CPU Failures
- Memory or DIMM Failures
- Virtual Interface Card Failures
- Motherboard Failures
- Hard Disk Failures
- Chassis Server Slot Issues

HA on Connector Nodes

We performed HA testing for Connector nodes by removing Network cable (similar to Cable failure) during the NFS IOzone testing. The following are the two screenshots captured during the network cable failure on the connector node.

Figure 31 “connector-node1” **Network Cable Failure on Cisco UCS C220 M4S Network Port 1**

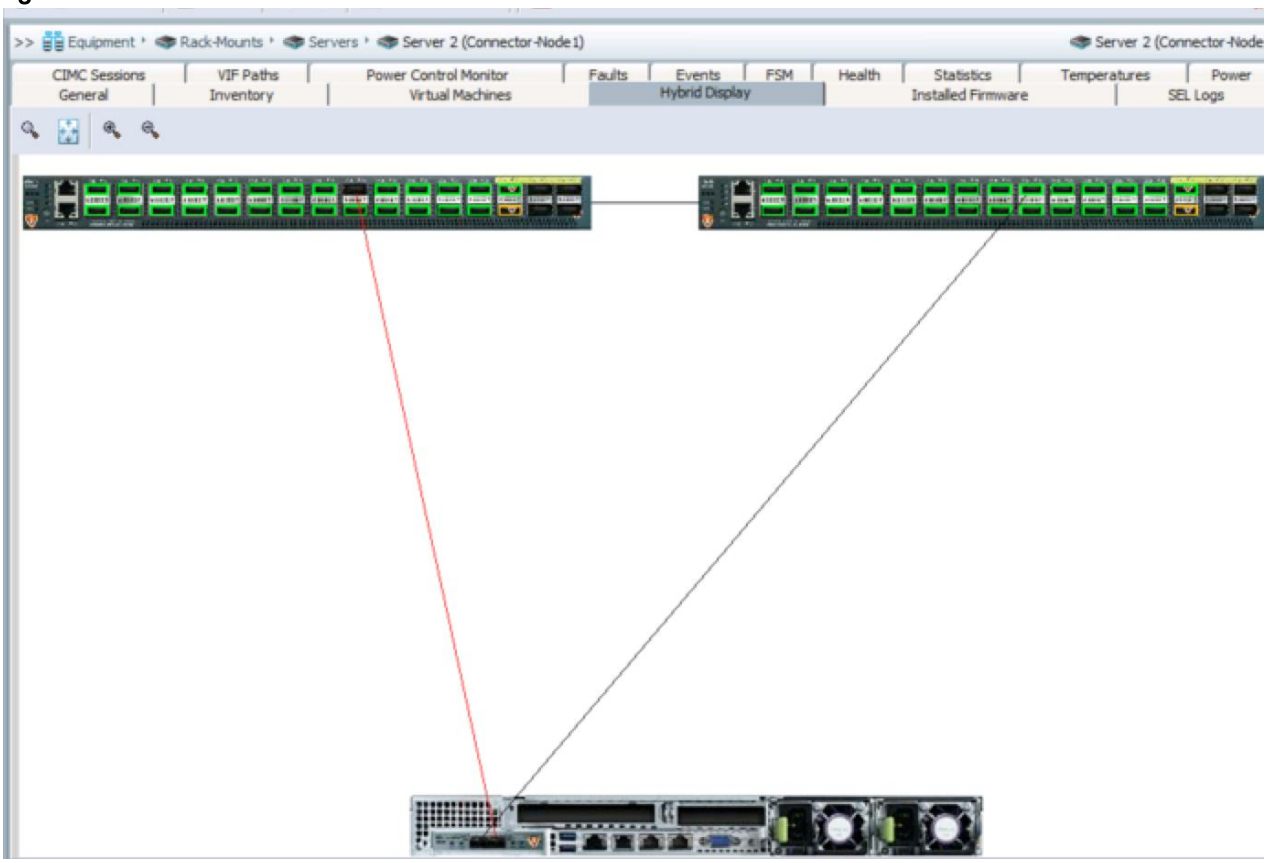
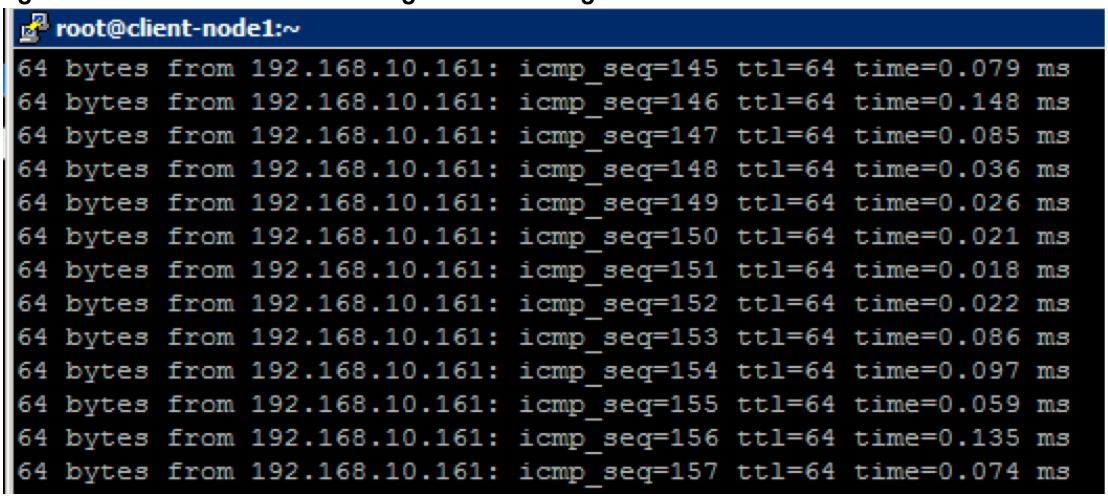


Figure 32 “iozone” NFS **Testing During Cable Failure**

```
root@client-node1:~
[root@client-node1 ~]# timeout 900 ./nfstest.bash -c 1000 -f 1048576 -b 4096 -m /mnt -w 12 -t 0 -l /root/nfslogs
COUNT is 1000
filesize is 1048576
blocksize is 4096
mountpoint is /mnt
workers is 12
type is 0
base is /root/nfslogs
1344098.70 kB/sec
1328178.02 kB/sec
1336855.62 kB/sec
1392936.49 kB/sec
1294188.73 kB/sec
1358624.08 kB/sec
1366718.04 kB/sec
1315192.79 kB/sec
1348700.12 kB/sec
1356978.19 kB/sec
1485333.56 kB/sec
1305207.33 kB/sec
1373792.05 kB/sec
1440562.35 kB/sec
1199558.73 kB/sec
1380220.71 kB/sec
1475241.62 kB/sec
1213814.80 kB/sec
1476674.77 kB/sec
```

Figure 33 Connector Node1 Ping Status During Cable Failure



The connector-node1 cable fault didn't dropped a ping and the transfer rate never drops below 1199558 kB/s after the failure. The rate of transfer at the point of the failure was 1373792 kB/s. During the HA test, the fluctuation of throughput that you see in this screenshot is expected. It is normal to see the throughput fluctuate about 10% above and below the average rate, so the network cable failure had no impact on performance.

The Network cable failure testing concludes, the Connector node Network cable fault didn't impact/interrupt the NFS testing.

HA on Storage Nodes

We performed HA testing for Storage nodes by completely powering down Cisco UCS S3260 Chassis-01 running Storage-node1 & Storage-node2. Below are the screenshots captured during the storage-node1 & storage-node2 powered down.

Figure 34 Storage Node1 Ping Status After Powering Down Cisco UCS S3260 Chassis

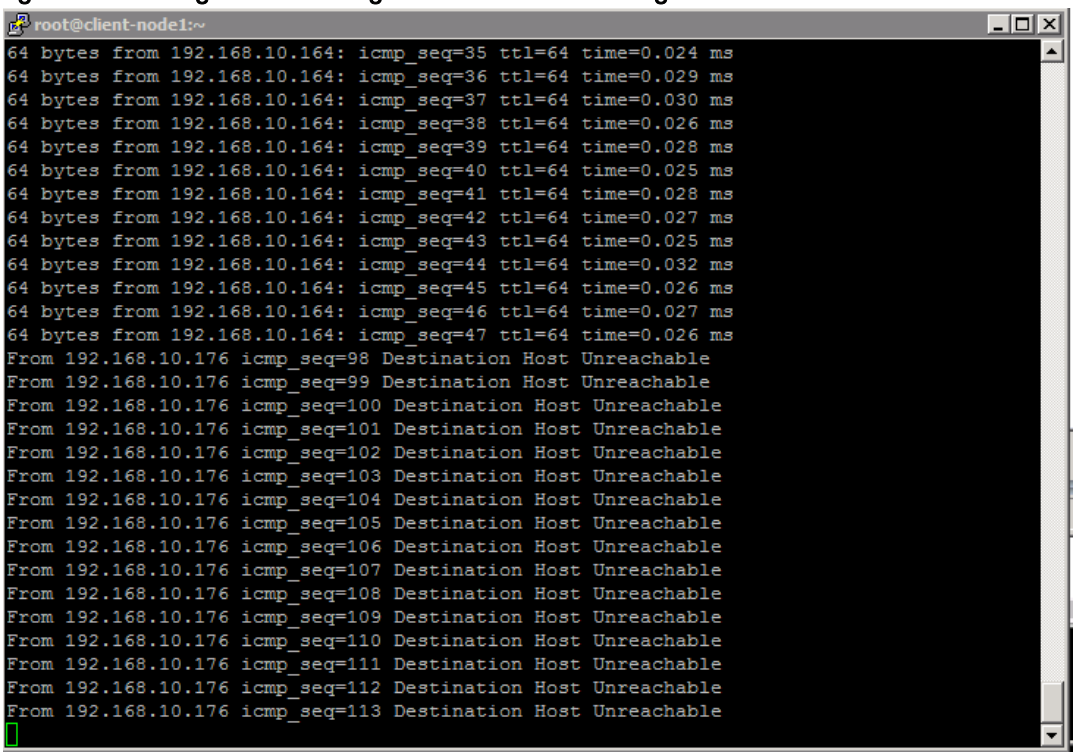
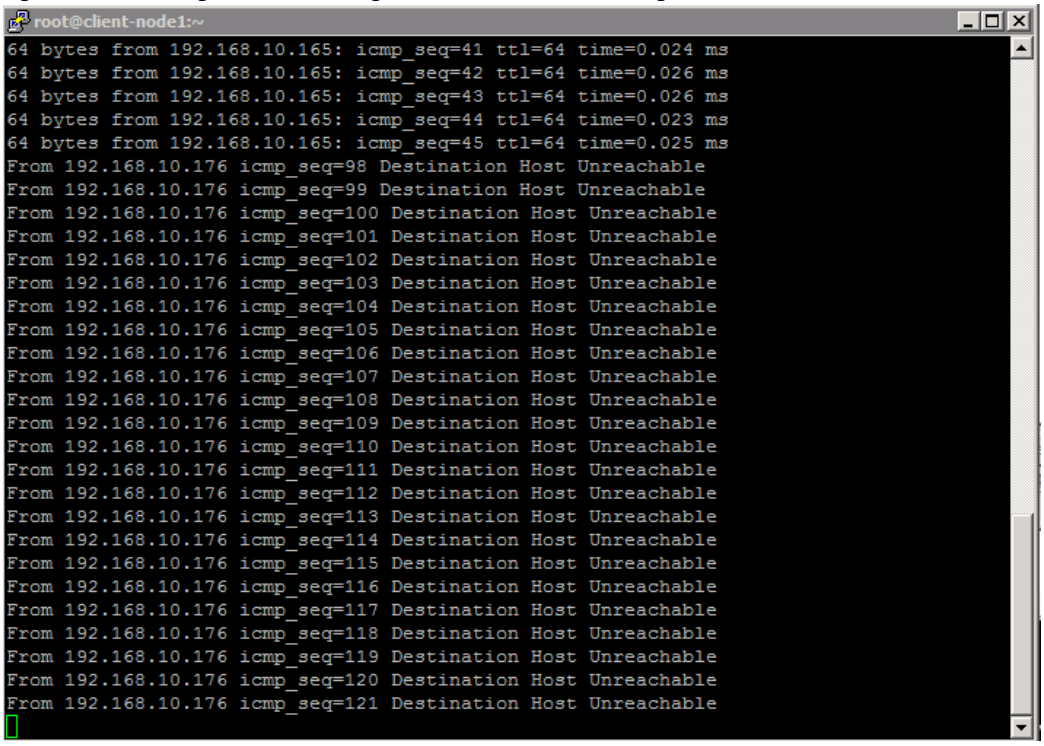


Figure 35 Storage Node2 Ping Status After Powering Down S3260 Chassis



RING Supervisor recognizes the failure. Note the 10 tasks running on the RING. This is normal. Before the configuration changes, we had over 50 rebuild tasks running and the storage servers were so busy they could not service NFS requests.

Figure 36 Scality RING Status After Powering Down Storage-Node1 & Storage-Node2

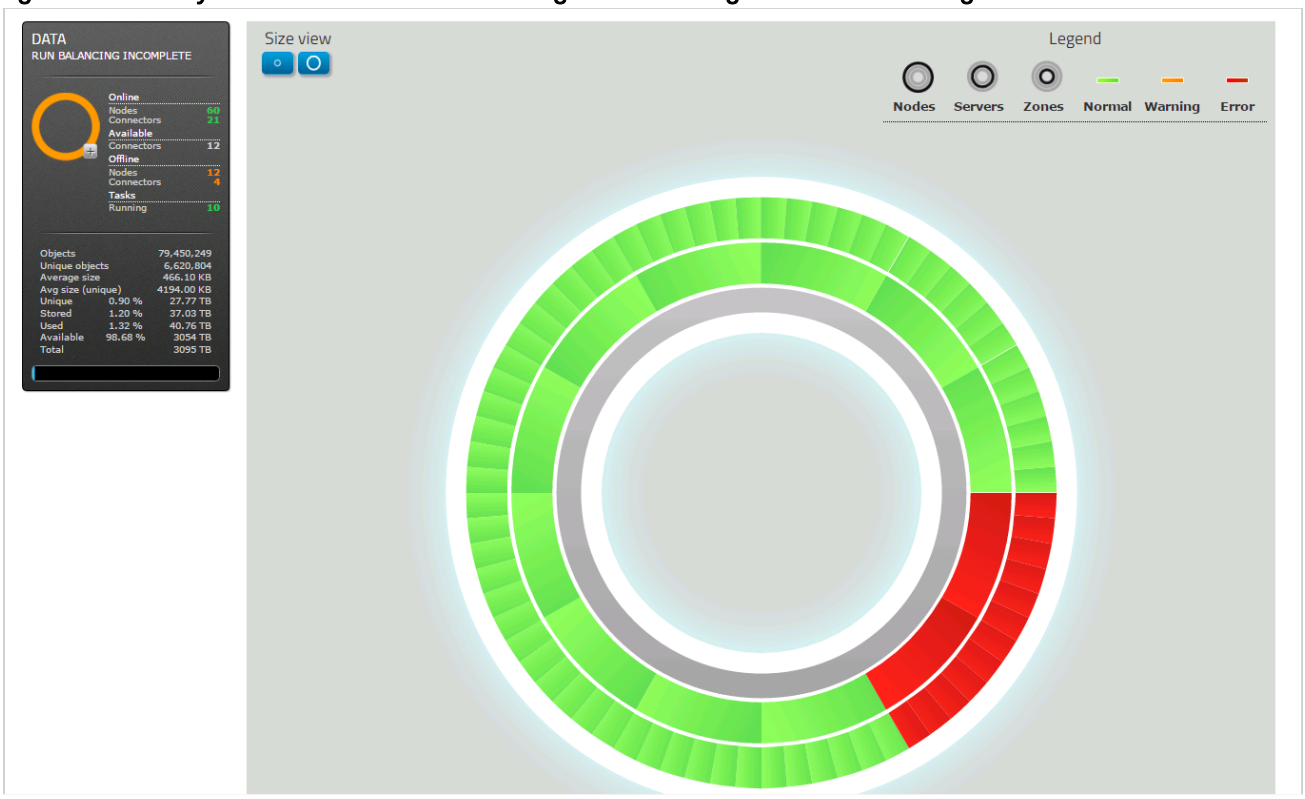
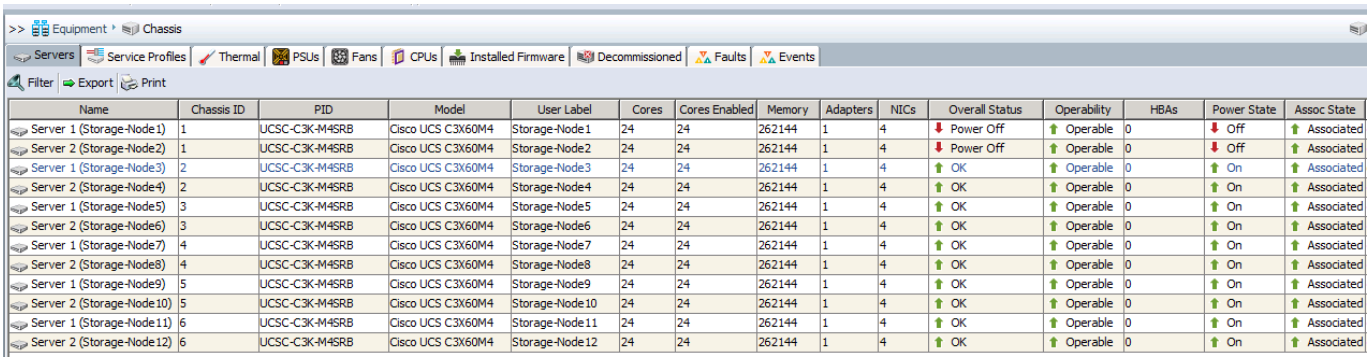


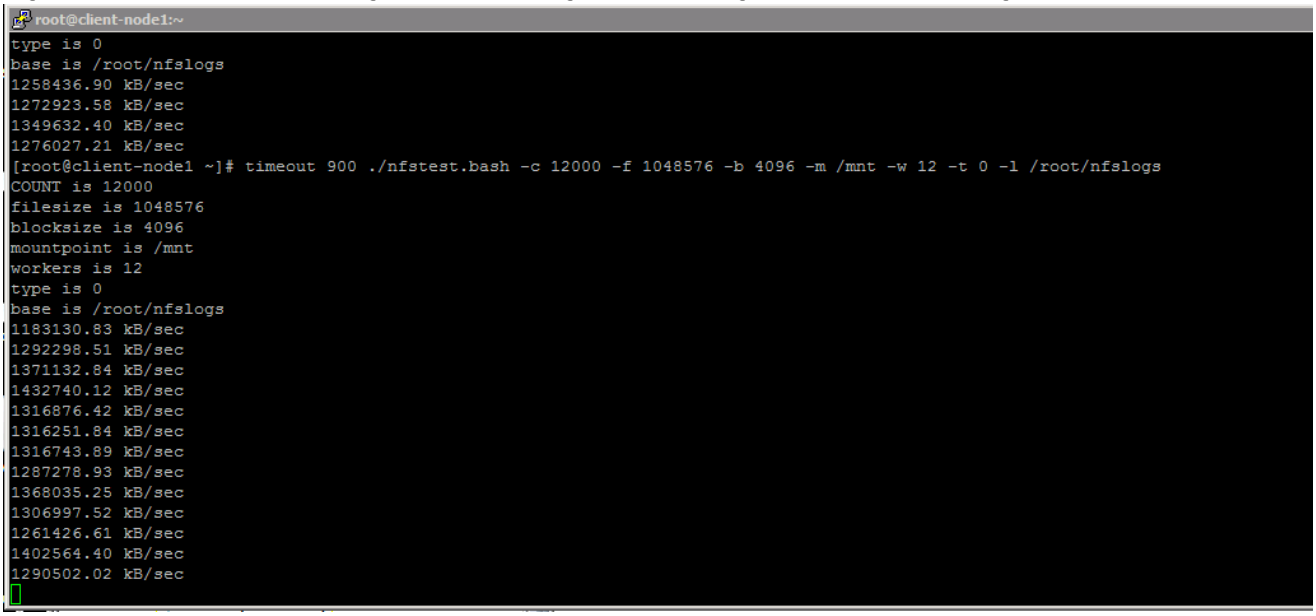
Figure 37 Cisco UCS Manager recognizes the failure after powering down storage-node1 and storage-node2



Name	Chassis ID	PID	Model	User Label	Cores	Cores Enabled	Memory	Adapters	NICs	Overall Status	Operability	HBAs	Power State	Assoc State
Server 1 (Storage-Node1)	1	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node1	24	24	262144	1	4	Power Off	Operable	0	Off	Associated
Server 2 (Storage-Node2)	1	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node2	24	24	262144	1	4	Power Off	Operable	0	Off	Associated
Server 1 (Storage-Node3)	2	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node3	24	24	262144	1	4	OK	Operable	0	On	Associated
Server 2 (Storage-Node4)	2	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node4	24	24	262144	1	4	OK	Operable	0	On	Associated
Server 1 (Storage-Node5)	3	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node5	24	24	262144	1	4	OK	Operable	0	On	Associated
Server 2 (Storage-Node6)	3	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node6	24	24	262144	1	4	OK	Operable	0	On	Associated
Server 1 (Storage-Node7)	4	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node7	24	24	262144	1	4	OK	Operable	0	On	Associated
Server 2 (Storage-Node8)	4	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node8	24	24	262144	1	4	OK	Operable	0	On	Associated
Server 1 (Storage-Node9)	5	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node9	24	24	262144	1	4	OK	Operable	0	On	Associated
Server 2 (Storage-Node10)	5	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node10	24	24	262144	1	4	OK	Operable	0	On	Associated
Server 1 (Storage-Node11)	6	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node11	24	24	262144	1	4	OK	Operable	0	On	Associated
Server 2 (Storage-Node12)	6	UCSC-C3K-M4SRB	Cisco UCS C3X60M4	Storage-Node12	24	24	262144	1	4	OK	Operable	0	On	Associated

The NFS test script continues writing to the RING. Note that after the point of the failure, the first data point is still 1.23 GB/s. So, powering off two storage servers and recognized no negative performance impact for NFS writes.

Figure 38 iozone NFS Testing After Powering Down Storage-Node1 and Storage-Node2



The Storage-node1 & Storage-node2 failure didn't impact the iozone NFS testing and the transfer rate never drops below 1199558 kB/s after the failure. The rate of transfer at the point of the failure was 1373792 kB/s. During the HA test, the fluctuation of throughput that you see in this screenshot is expected. It's normal to see the throughput fluctuate about 10% above and below the average rate, so 2 Storage nodes failure had no impact on performance.

Bill of Materials

This section provides the BOM for the Scality Storage with Cisco UCS S3260 solution.

Table 8 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 Connect-ors	4
N9K-PAC-650W	Nexus 9300 650W AC PS, Port-side Intake	4

Table 9 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 10 Bill of Materials for Cisco UCS S3260 Storage Server

Item Name	Description	Quantity
UCS-S3260	Cisco UCS S3260 Storage Server Base Chassis	6
UCS-C3X60-G2SD48	UCS C3X60 480GB Boot SSD (Gen 2)	24
UCSC-PSU1-1050W	UCS C3X60 1050W Power Supply Unit	24
UCS-C3K-42HD10	UCS C3X60 3 row of 10TB NL-SAS drives (42 Total) 420TB	6
UCS-C3X60-12G280	UCS C3X60 800GB 12Gbps SSD (Gen 2)	24
UCS-C3X60-10TB	UCS C3X60 10TB 12Gbps NL-SAS 7200RPM HDD w carrier- Top-load	60
CAB-C13-CBN	Cabinet Jumper Power Cord, 250 VAC 10A, C14-C13 Connectors	24
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	12
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	12
UCS-CPU-E52650E	2.20 GHz E5-2650 v4/105W 12C/05MB Cache/DDR4 2400MHz	24
UCS-MR-1X161RV-A	16GB DDR4-2400-MHz RDIMM/PC4-19200/single rank/x4/1.2v	256
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	12
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	24
RHEL-2S2V-1A	Red Hat Enterprise Linux (1-2 CPU,1-2 VN); 1-Yr Support Req	6

Table 11 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	4
UCS-CPU-E52683E	2.10 GHz E5-2683 v4/120W 16C/40MB Cache/DDR4 2400MHz	8
UCS-MR-1X161RV-A	16GB DDR4-2400-MHz RDIMM/PC4-19200/single rank/x4/1.2v	64

Item Name	Description	Quantity
UCS-HD600G10K12G	600GB 12G SAS 10K RPM SFF HDD	8
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	4
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 M4 and C240 M4 rack servers	4
UCSC-PSU1-770W	770W AC Hot-Plug Power Supply for 1U C-Series Rack Server	8
CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	8
UCS-M4-V4-LBL	Cisco M4 - v4 CPU asset tab ID label (Auto-Expand)	7
N20-BBLKD	UCS 2.5 inch HDD blanking panel	24
UCSC-SCCBL220	Supercap cable 950mm	4
UCSC-MLOM-BLK	MLOM Blanking Panel	4
UCSC-HS-C220M4	Heat sink for UCS C220 M4 rack servers	8
UCSC-MRAID12G	Cisco 12G SAS Modular Raid Controller	4
UCSC-MRAID12G-1GB	Cisco 12Gbps SAS 1GB FBWC Cache module (Raid 0/1/5/6)	4
RHEL-2S2V-1A	Red Hat Enterprise Linux (1-2 CPU,1-2 VN); 1-Yr Support Req	4

Appendix

Appendix A – Kickstart File of Connector Nodes for Cisco UCS C220 M4S

Kickstart file for Connector-node1

#./connector-node1.cfg

```
#version=DEVEL
#from the linux installation menu, hit tab and append this:
#biosdevname=0 net.ifnames=0 ip=eth1:dhcp
#ks=ftp://192.168.10.2/pub/{hostname}.cfg
# System authorization information
auth --enableshadow --passalgo=sha512
repo --name="Server-HighAvailability" --
baseurl=file:///run/install/repo/addons/HighAvailability
repo --name="Server-ResilientStorage" --
baseurl=file:///run/install/repo/addons/ResilientStorage
# Use CDROM installation media
cdrom
# Use text install
text
# Run the Setup Agent on first boot
firstboot --disable
selinux --disable
firewall --disable
ignoredisk --only-use=sda
# Keyboard layouts
keyboard --vckeymap=us --xlayouts='us'
# System language
lang en_US.UTF-8

# Network information
network  --bootproto=static --device=eth0 --ip=128.107.79.202 --
netmask=255.255.255.0 --onboot=on --ipv6=auto --activate --gateway=128.107.79.1 --
nameserver=171.70.168.183
network  --bootproto=static --device=eth1 --ip=192.168.10.161 --
netmask=255.255.255.0 --onboot=on --ipv6=auto --activate
network  --bootproto=static --device=eth2 --ip=192.168.20.161 --
netmask=255.255.255.0 --onboot=off --ipv6=auto --activate
network  --bootproto=static --device=eth2 --ip=192.168.30.161 --
netmask=255.255.255.0 --onboot=off --ipv6=auto --activate
network  --hostname=connector-node1

# Root password
```

```
rootpw --iscrypted
$6$yfE2jHtdy.OSmO8g$InneiVXQI9Kc9m4w2cEiS8/og6BKUlu5HSR0eCYgh5dVaeCV54Q6piS7k10lalXi
gnLCBvAZPqmw4dvYgy66V1
# System services
services --disabled="chronyd"
# System timezone
timezone America/Los_Angeles --isUtc --nontp
# System bootloader configuration
bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=sda
# Partition clearing information
ignoredisk --only-use=sda
clearpart --all --initlabel
# Disk partitioning information
part /boot --fstype="ext4" --ondisk=sda --size=8192
part swap --fstype="swap" --ondisk=sda --size=32767
part /var --fstype="ext4" --ondisk=sda --grow
part / --fstype="ext4" --ondisk=sda --size=40960

reboot

%packages
@^minimal
@core
kexec-tools
#Extra Packages beyond the minimal installation.
#These packages are prerequisites for Scality and EPEL
#packages to be loaded during the Scality installation.
apr-util
apr
atk
autogen-libopts
cairo
cups-libs
dejavu-fonts-common
dejavu-sans-mono-fonts
dialog
fontconfig
fontpackages-filesystem
gdk-pixbuf2
gd
ghostscript-fonts
ghostscript
graphite2
graphviz
gtk2
harfbuzz
```

```
hicolor-icon-theme
httpd-tools
httpd
jansson
jasper-libs
jbigkit-libs
kernel
lcms2
libfontenc
libICE
libjpeg-turbo
libpng
librsvg2
libSM
libthai
libtiff
libtool-ltdl
libwebp
libX11-common
libX11
libXau
libXaw
libxcb
libXcomposite
libXcursor
libXdamage
libXext
libXfixes
libXfont
libXft
libXinerama
libXi
libXmu
libXpm
libXrandr
libXrender
libxshmfence
libxslt
libXt
libXxf86vm
libyaml
m2crypto
mailcap
mesa-libEGL
mesa-libgbm
```

```
mesa-libglapi
mesa-libGL
mod_ssl
ntpdate
ntp
pango
pciutils
pixman
poppler-data
pycairo
python-babel
python-backports
python-chardet
python-kitchen
python-pillow
python-pyasn1
python-setproctitle
python-setuptools
PyYAML
rrdtool
rsync
systemd-python
urw-fonts
wget
xorg-x11-font-utils
yum-utils
bzip2
GConf2
flac-libs
giflib
gsm
javapackages-tools
libXtst
libasyncns
libogg
libsndfile
libvorbis
lksctp-tools
pcsc-lite-libs
psmisc
pulseaudio-libs
python-javapackages
python-lxml
ttmkfdir
xorg-x11-fonts-Type1
```

```
perl-Data-Dumper
xz-devel
zlib-devel
at
attr
cups-client
ed
fuse
fuse-libs
libicu
m4
patch
perl-Data-Dumper
redhat-lsb-core
redhat-lsb-submod-security
spax
time
python-virtualenv
keyutils
libbasicobjects
libcollection
libevent
libini_config
libnfsidmap
libpath_utils
libref_array
libtirpc
libverto-tevent
tcp_wrappers
python-netaddr
cdparanoia-libs
exempi
gstreamer1
gstreamer1-plugins-base
iso-codes
libXv
libexif
libgsf
libgxps
libimobiledevice
libiptcdata
libmediaart
libosinfo
libplist
libtheora
```

```
libusbx
libvisual
openjpeg-libs
orc
poppler
poppler-glib
taglib
totem-pl-parser
tracker
upower
usbmuxd
xml-common
python-six
#Extra packages loaded for Scality Engineering
strace
lsof
mailx
smartmontools
dstat
traceroute
gdb
telnet
hdparm
screen
iotop
bc
lm_sensors-libs
sysstat
perl-parent
perl-HTTP-Tiny
perl-podlators
perl-Pod-Perldoc
perl-Pod-Escapes
perl-Text-ParseWords
perl-Encode
perl-Pod-Usage
perl-libs
perl-macros
perl-Storable
perl-Exporter
perl-constant
perl-Time-Local
perl-Socket
perl-Carp
perl-Time-HiRes
```

```
perl-PathTools
perl-Scalar-List-Utils
perl-File-Temp
perl-File-Path
perl-threads-shared
perl-threads
perl-Filter
perl-Pod-Simple
perl-Getopt-Long
perl
vim-filesystem
vim-common
gpm-libs
vim-enhanced
tcpdump
zip
mtr
#
%end

%addon com_redhat_kdump --enable --reserve-mb='auto'

%end

%anaconda
pwpolicy root --minlen=6 --minquality=50 --notstrict --nochanges --notempty
pwpolicy user --minlen=6 --minquality=50 --notstrict --nochanges --notempty
pwpolicy luks --minlen=6 --minquality=50 --notstrict --nochanges --notempty
%end

###############
#POST SCRIPT
###############
%post --log=/root/ks-post.log
###############
#Set kernel parameters for Scality RING
###############
cat > /etc/sysctl.d/99-scality.conf <<EOF1
kernel.sem = 256 32000 32 256
net.core.netdev_max_backlog = 3000
net.core.optmem_max = 524287
net.core.rmem_default = 174760
net.core.wmem_default = 174760
net.core.wmem_max = 1677721600
net.ipv4.conf.all.send_redirects = 0
```

```
net.ipv4.conf.lo.arp_filter = 1
net.ipv4.ip_local_port_range = 20480 65000
net.ipv4.tcp_dsack = 1
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_keepalive_time = 1800
net.ipv4.tcp_mem = 1024000 8738000 1677721600
net.ipv4.tcp_mtu_probing = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1
vm.vfs_cache_pressure = 50
net.ipv4.conf.all.accept_redirects= 0
net.ipv4.tcp_syncookies= 0
net.core.rmem_max= 1677721600
net.ipv4.tcp_wmem= 4096 174760 16777216
net.ipv4.tcp_rmem= 4096 174760 16777216
net.core.somaxconn= 2048
net.ipv4.tcp_dsack= 0
net.ipv4.tcp_sack= 0
kernel.sem= 512 32000 32 256
EOF1
cat > /etc/sysctl.d/99-salt.conf <<EOF2
vm.swappiness = 1
vm.min_free_kbytes = 2000000
kernel.sem = 250        32000   32      256
net.ipv4.tcp_syncookies = 1
EOF2
###############
#Set security limits
###############
cat > /etc/security/limits.d/99-scality.conf <<EOF3
root hard       sigpending      1031513
root soft       sigpending      1031513
root hard       nofile  65535
root soft       nofile  65535
root hard       nproc   1031513
root soft       nproc   1031513
root hard       stack   10240
root soft       stack   10240
EOF3
###############
#Preconfigure /etc/hosts
###############
cat >> /etc/hosts <<EOF4
192.168.10.160 supervisor salt
192.168.10.161 connector-node1
```

```
192.168.10.162 connector-node2
192.168.10.163 connector-node3
192.168.10.164 storage-node1
192.168.10.165 storage-node2
192.168.10.166 storage-node3
192.168.10.167 storage-node4
192.168.10.168 storage-node5
192.168.10.169 storage-node6
192.168.10.170 storage-node7
192.168.10.171 storage-node8
192.168.10.172 storage-node9
192.168.10.173 storage-node10
192.168.10.174 storage-node11
192.168.10.175 storage-node12
EOF4
###############
#Setup ssh keys
###############
mkdir /root/.ssh;
cat > /root/.ssh/id_rsa <<EOF5
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAsYGqxWxQdGUsiUzafYLuX6MVD3mjq3r6KaL0QcNSuZ8F3Xfw
….....
…..
TfYW1tZ7g7gZJ+To42h4Tv9wj8iWGe+pnR4Moh3WqM1TttuaCJf1nQ==
-----END RSA PRIVATE KEY-----
EOF5
cat > /root/.ssh/id_rsa.pub <<EOF6
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCxgarFbFB0ZSyJTNp9gu5foxUPeaOrevopovRBw1K5nwXdd/DtYlaO
aG67+u6stWgD3R+NkNBjpoQB0dIf6jbuYfqF+QxWrK6fBDq7cy1SqqTBErY20QmokGIdnD35uaCh/IViXnAF
JY8YiKDSRvyb5wGbS5GgTlIUW4AZzu6weR8gWU5BB32/P2Ho5fxtrdzrJQBkPNZKe3a53Is5OpXhI+lBjg7Y
29iCbVWluUe9S+Y/ti7nKXyHGfSKf5GZ96tOrxbDJmKExXQTI3irkd9P6B1tJjrE8wz1QcXz36Vg03F1nj9W
4FxsGyR7LdRtDffYqoDvBL5KtrYNead/KxZv root@storage-node7
EOF6
cat > /root/.ssh/authorized_keys <<EOF7
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCxgarFbFB0ZSyJTNp9gu5foxUPeaOrevopovRBw1K5nwXdd/DtYlaO
aG67+u6stWgD3R+NkNBjpoQB0dIf6jbuYfqF+QxWrK6fBDq7cy1SqqTBErY20QmokGIdnD35uaCh/IViXnAF
JY8YiKDSRvyb5wGbS5GgTlIUW4AZzu6weR8gWU5BB32/P2Ho5fxtrdzrJQBkPNZKe3a53Is5OpXhI+lBjg7Y
29iCbVWluUe9S+Y/ti7nKXyHGfSKf5GZ96tOrxbDJmKExXQTI3irkd9P6B1tJjrE8wz1QcXz36Vg03F1nj9W
4FxsGyR7LdRtDffYqoDvBL5KtrYNead/KxZv root@storage-node7
EOF7
chmod 700 /root/.ssh;
chmod 600 /root/.ssh/authorized_keys;
chmod 600 /root/.ssh/id_rsa;
chmod 644 /root/.ssh/id_rsa.pub;
```

```
###############
#The first two files downloaded here (for the precheck script)
#will only be available from Scality support.
#They are not available to customers for downloads.
###############
wget --directory-prefix=/root ftp://192.168.10.2/pub/scality-pre_install_checks-4.5-1-g846676e.py;
wget --directory-prefix=/root ftp://192.168.10.2/pub/template.json;
###############
#Download the Scality run file to /root.
#This is really only necessary on the supervisor.
#Customers would likely remove this wget command
#from the kickstart file
###############
wget --directory-prefix=/root ftp://192.168.10.2/pub/scality-ring-6.4.0.r161228230017.5100943_centos_7.run;
###############
#Edit /etc/sysconfig/irqbalance so irqbalance runs
#only once at boot.
###############
sed -i 's/#IRQBALANCE_ONESHOT=/IRQBALANCE_ONESHOT=yes/' /etc/sysconfig/irqbalance;
###############
#Turn off Transparent Hugepages and ensure that hyperthreading
#is turned off.
###############
grubby --update-kernel=ALL --args="transparent_hugepage=never numa=off nr_cpus=24";
tuned-adm profile latency-performance;
systemctl enable ntpd;
###############
#Bring up the public interface.
#Register the system for package updates and installations.
#Update all packages.
###############
ifup eth0;
subscription-manager register --username=vijd@cisco.com --password=[password] --auto-attach;
subscription-manager repos --disable=*;
subscription-manager repos --enable=rhel-7-server-optional-rpms;
subscription-manager repos --enable=rhel-7-server-rpms; subscription-manager repos --enable=rhel-7-server-extras-rpms;
yum -y update;
###############
#Install epel-release for access to EPEL repository.
###############
wget --directory-prefix=/root https://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-8.noarch.rpm;
```

```
yum -y install /root/epel-release-7-8.noarch.rpm;
###############
#List packages below that get installed during the Scality
#precheck, software installation and
#the S3 connector installation
###############
yum -y install iperf iperf3 htop;
# Remove NetworkManger, a core package which is not needed.
yum -y remove NetworkManager;
%end
```

Appendix B – Kickstart File of Storage nodes for Cisco UCS S3260 M4 Server

Kickstart file for Storage-node1

#./storage-node1.cfg

```
#version=DEVEL
#from the linux installation menu, hit tab and append this:
#biosdevname=0 net.ifnames=0 ip=eth1:dhcp
#ks=ftp://192.168.10.2/pub/{hostname}.cfg
# System authorization information
auth --enableshadow --passalgo=sha512
repo --name="Server-HighAvailability" --
baseurl=file:///run/install/repo/addons/HighAvailability
repo --name="Server-ResilientStorage" --
baseurl=file:///run/install/repo/addons/ResilientStorage
# Use CDROM installation media
cdrom
# Use text install
text
# Run the Setup Agent on first boot
firstboot --disable
selinux --disable
firewall --disable
ignoredisk --only-use=sdac
# Keyboard layouts
keyboard --vckeymap=us --xlayouts='us'
# System language
lang en_US.UTF-8

# Network information
network  --bootproto=static --device=eth0 --ip=128.107.79.205 --
netmask=255.255.255.0 --onboot=on --gateway=128.107.79.1 --nameserver=171.70.168.183
--ipv6=auto --activate
network  --bootproto=static --device=eth1 --ip=192.168.10.164 --
netmask=255.255.255.0 --onboot=on --ipv6=auto --activate
```

```
network  --bootproto=static --device=eth2 --ip=192.168.20.164 --
netmask=255.255.255.0 --onboot=on --ipv6=auto --activate
network  --bootproto=static --device=eth3 --ip=192.168.30.164 --
netmask=255.255.255.0 --onboot=on --ipv6=auto --activate
network  --hostname=storage-node1

# Root password
rootpw --iscrypted
$6$yfE2jHtdy.OSmO8g$InneiVXQI9Kc9m4w2cEiS8/og6BKUlu5HSR0eCYgh5dVaeCV54Q6piS7k10lalXi
gnLCBvAZPqmw4dvYgy66V1
# System services
services --disabled="chronyd"
# System timezone
timezone America/Los_Angeles --isUtc --nontp
# System bootloader configuration
bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=sdac
# Partition clearing information
ignoredisk --only-use=sdac
clearpart --all --initlabel
# Disk partitioning information
part /boot --fstype="ext4" --ondisk=sdac --size=8192
part swap --fstype="swap" --ondisk=sdac --size=32767
part /var --fstype="ext4" --ondisk=sdac --grow
part / --fstype="ext4" --ondisk=sdac --size=40960

reboot

%packages
@^minimal
@core
kexec-tools
#Extra Packages beyond the minimal installation.
#These packages are prerequisites for Scality and EPEL
#packages to be loaded during the Scality installation.
apr-util
apr
atk
autogen-libopts
cairo
cups-libs
dejavu-fonts-common
dejavu-sans-mono-fonts
dialog
fontconfig
fontpackages-filesystem
gdk-pixbuf2
```

```
gd
ghostscript-fonts
ghostscript
graphite2
graphviz
gtk2
harfbuzz
hicolor-icon-theme
httpd-tools
httpd
jansson
jasper-libs
jbigkit-libs
kernel
lcms2
libfontenc
libICE
libjpeg-turbo
libpng
librsvg2
libSM
libthai
libtiff
libtool-ltdl
libwebp
libX11-common
libX11
libXau
libXaw
libxcb
libXcomposite
libXcursor
libXdamage
libXext
libXfixes
libXfont
libXft
libXinerama
libXi
libXmu
libXpm
libXrandr
libXrender
libxshmfence
libxslt
```

```
libXt
libXxf86vm
libyaml
m2crypto
mailcap
mesa-libEGL
mesa-libgbm
mesa-libglapi
mesa-libGL
mod_ssl
ntpdate
ntp
pango
pciutils
pixman
poppler-data
pycairo
python-babel
python-backports
python-chardet
python-kitchen
python-pillow
python-pyasn1
python-setproctitle
python-setuptools
PyYAML
rrdtool
rsync
systemd-python
urw-fonts
wget
xorg-x11-font-utils
yum-utils
bzip2
GConf2
flac-libs
giflib
gsm
javapackages-tools
libXtst
libasyncns
libogg
libsndfile
libvorbis
lksctp-tools
```

```
pcsc-lite-libs
psmisc
pulseaudio-libs
python-javapackages
python-lxml
ttmkfdir
xorg-x11-fonts-Type1
perl-Data-Dumper
xz-devel
zlib-devel
at
attr
cups-client
ed
fuse
fuse-libs
libicu
m4
patch
perl-Data-Dumper
redhat-lsb-core
redhat-lsb-submod-security
spax
time
python-virtualenv
keyutils
libbasicobjects
libcollection
libevent
libini_config
libnfsidmap
libpath_utils
libref_array
libtirpc
libverto-tevent
tcp_wrappers
python-netaddr
cdparanoia-libs
exempi
gstreamer1
gstreamer1-plugins-base
iso-codes
libXv
libexif
libgsf
```

```
libgxps
libimobiledevice
libiptcdata
libmediaart
libosinfo
libplist
libtheora
libusbx
libvisual
openjpeg-libs
orc
poppler
poppler-glib
taglib
totem-pl-parser
tracker
upower
usbmuxd
xml-common
python-six
#Extra packages loaded for Scality Engineering
strace
lsof
mailx
smartmontools
dstat
traceroute
gdb
telnet
hdparm
screen
iotop
bc
lm_sensors-libs
sysstat
perl-parent
perl-HTTP-Tiny
perl-podlators
perl-Pod-Perldoc
perl-Pod-Escapes
perl-Text-ParseWords
perl-Encode
perl-Pod-Usage
perl-libs
perl-macros
```

```
perl-Storable
perl-Exporter
perl-constant
perl-Time-Local
perl-Socket
perl-Carp
perl-Time-HiRes
perl-PathTools
perl-Scalar-List-Utils
perl-File-Temp
perl-File-Path
perl-threads-shared
perl-threads
perl-Filter
perl-Pod-Simple
perl-Getopt-Long
perl
vim-filesystem
vim-common
gpm-libs
vim-enhanced
tcpdump
zip
mtr
#
%end

%addon com_redhat_kdump --enable --reserve-mb='auto'

%end

%anaconda
pwpolicy root --minlen=6 --minquality=50 --notstrict --nochanges --notempty
pwpolicy user --minlen=6 --minquality=50 --notstrict --nochanges --notempty
pwpolicy luks --minlen=6 --minquality=50 --notstrict --nochanges --notempty
%end

###############
#POST SCRIPT
###############
%post --log=/root/ks-post.log
###############
#Set kernel parameters for Scality RING
###############
cat > /etc/sysctl.d/99-scality.conf <<EOF1
```

```
kernel.sem = 256 32000 32 256
net.core.netdev_max_backlog = 3000
net.core.optmem_max = 524287
net.core.rmem_default = 174760
net.core.wmem_default = 174760
net.core.wmem_max = 1677721600
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.lo.arp_filter = 1
net.ipv4.ip_local_port_range = 20480 65000
net.ipv4.tcp_dsack = 1
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_keepalive_time = 1800
net.ipv4.tcp_mem = 1024000 8738000 1677721600
net.ipv4.tcp_mtu_probing = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1
vm.vfs_cache_pressure = 50
net.ipv4.conf.all.accept_redirects= 0
net.ipv4.tcp_syncookies= 0
net.core.rmem_max= 1677721600
net.ipv4.tcp_wmem= 4096 174760 16777216
net.ipv4.tcp_rmem= 4096 174760 16777216
net.core.somaxconn= 2048
net.ipv4.tcp_dsack= 0
net.ipv4.tcp_sack= 0
kernel.sem= 512 32000 32 256
EOF1
cat > /etc/sysctl.d/99-salt.conf <<EOF2
vm.swappiness = 1
vm.min_free_kbytes = 2000000
kernel.sem = 250        32000   32      256
net.ipv4.tcp_syncookies = 1
EOF2
###############
#Set security limits
###############
cat > /etc/security/limits.d/99-scality.conf <<EOF3
root hard       sigpending      1031513
root soft       sigpending      1031513
root hard       nofile  65535
root soft       nofile  65535
root hard       nproc   1031513
root soft       nproc   1031513
root hard       stack   10240
root soft       stack   10240
```

```
EOF3
###############
#Preconfigure /etc/hosts
###############
cat >> /etc/hosts <<EOF4
192.168.10.160 supervisor salt
192.168.10.161 connector-node1
192.168.10.162 connector-node2
192.168.10.163 connector-node3
192.168.10.164 storage-node1
192.168.10.165 storage-node2
192.168.10.166 storage-node3
192.168.10.167 storage-node4
192.168.10.168 storage-node5
192.168.10.169 storage-node6
192.168.10.170 storage-node7
192.168.10.171 storage-node8
192.168.10.172 storage-node9
192.168.10.173 storage-node10
192.168.10.174 storage-node11
192.168.10.175 storage-node12
EOF4
###############
#Setup ssh keys
###############
mkdir /root/.ssh;
cat > /root/.ssh/id_rsa <<EOF5
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAsYGqxWxQdGUsiUzafYLuX6MVD3mjq3r6KaL0QcNSuZ8F3Xfw
…
TfYW1tZ7g7gZJ+To42h4Tv9wj8iWGe+pnR4Moh3WqM1TttuaCJf1nQ==
-----END RSA PRIVATE KEY-----
EOF5
cat > /root/.ssh/id_rsa.pub <<EOF6
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCxgarFbFB0ZSyJTNp9gu5foxUPeaOrevopovRBw1K5nwXdd/DtYlaO
…F1nj9W4FxsGyR7LdRtDffYqoDvBL5KtrYNead/KxZv root@storage-node7
EOF6
cat > /root/.ssh/authorized_keys <<EOF7
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCxgarFbFB0ZSyJTNp9gu5foxUPeaOrevopovRBw1K5nwXdd/ …..
Ho5fxtrdzrJQBkPNZKe3a53Is5OpXhI+lBjg/KxZv root@storage-node7
EOF7
chmod 700 /root/.ssh;
chmod 600 /root/.ssh/authorized_keys;
chmod 600 /root/.ssh/id_rsa;
```

```
chmod 644 /root/.ssh/id_rsa.pub;
###############
#The first two files downloaded here (for the precheck script)
#will only be available from Scality support.
#They are not available to customers for downloads.
###############
wget --directory-prefix=/root ftp://192.168.10.2/pub/scality-pre_install_checks-4.5-1-g846676e.py;
wget --directory-prefix=/root ftp://192.168.10.2/pub/template.json;
###############
#Download the Scality run file to /root.
#This is really only necessary on the supervisor.
#Customers would likely remove this wget command
#from the kickstart file
###############
wget --directory-prefix=/root ftp://192.168.10.2/pub/scality-ring-6.4.0.r161228230017.5100943_centos_7.run;
###############
#Edit /etc/sysconfig/irqbalance so irqbalance runs
#only once at boot.
###############
sed -i 's/#IRQBALANCE_ONESHOT=/IRQBALANCE_ONESHOT=yes/' /etc/sysconfig/irqbalance;
###############
#Turn off Transparent Hugepages and ensure that hyperthreading
#is turned off.
###############
grubby --update-kernel=ALL --args="transparent_hugepage=never numa=off nr_cpus=24";
tuned-adm profile latency-performance;
systemctl enable ntpd;
###############
#Bring up the public interface.
#Register the system for package updates and installations.
#Update all packages.
###############
ifup eth0;
subscription-manager register --username=vijd@cisco.com --password=[password] --auto-attach;
subscription-manager repos --disable=*;
subscription-manager repos --enable=rhel-7-server-optional-rpms;
subscription-manager repos --enable=rhel-7-server-rpms; subscription-manager repos --enable=rhel-7-server-extras-rpms;
yum -y update;
###############
#Install epel-release for access to EPEL repository.
###############
```

```
wget --directory-prefix=/root https://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-8.noarch.rpm;
yum -y install /root/epel-release-7-8.noarch.rpm;
###############
#List packages below that get installed during the Scality
#precheck, software installation and
#the S3 connector installation
###############
yum -y install iperf iperf3 htop;
# Remove NetworkManger, a core package which is not needed.
yum -y remove NetworkManager;
%end
```

Appendix C – Example /etc/hosts File

/etc/hosts for Supervisor-node

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.10.160  supervisor salt
192.168.10.161  connector-node1
192.168.10.162  connector-node2
192.168.10.163  connector-node3
192.168.10.164  storage-node1
192.168.20.164  s3cvd
192.168.10.165  storage-node2
192.168.10.166  storage-node3
192.168.10.167  storage-node4
192.168.10.168  storage-node5
192.168.10.169  storage-node6
192.168.10.170  storage-node7
192.168.10.171  storage-node8
192.168.10.172  storage-node9
192.168.10.173  storage-node10
192.168.10.174  storage-node11
192.168.10.175  storage-node12
192.168.10.176  client-node1
192.168.10.177  client-node2
192.168.10.178  client-node3
192.168.10.179  client-node4
192.168.10.180  client-node5
192.168.10.181  client-node6
```

/etc/hosts for Connector Nodes & Storage Nodes

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.10.160 supervisor salt
192.168.10.161 connector-node1
192.168.10.162 connector-node2
192.168.10.163 connector-node3
192.168.10.164 storage-node1
192.168.10.165 storage-node2
192.168.10.166 storage-node3
192.168.10.167 storage-node4
192.168.10.168 storage-node5
192.168.10.169 storage-node6
192.168.10.170 storage-node7
192.168.10.171 storage-node8
192.168.10.172 storage-node9
192.168.10.173 storage-node10
192.168.10.174 storage-node11
192.168.10.175 storage-node12
```

Appendix D – Best Practice Configurations for Ordering Cisco UCS S3260 for Scality

Note that Scality RING configurations can be built to meet your specific business and application needs. Standard building-block configurations of both the UCS S3260 and the UCS C240 M4 will meet the needs of most customers.

The following are basic rules to follow when building a best-practice configuration:

- Single-Site RINGs: Start with a minimum of six storage servers. Grow in increments of three storage servers.
- Two-Site RINGs: Start with a minimum of twelve storage servers. Grow in increments of six storage servers.
- Three-Site RINGs: Start with a minim of twelve storage servers. Grow in increments of six storage servers.

General best practices:

- Connector processes will be installed directly onto the storage servers. Advanced configurations requiring multiple connector protocols (for example: S3 and NFS) should be designed with a Cisco/Scality specialist and may leverage external connector processes running on the Cisco UCS C220 server.
- The Scality Supervisor management interface may be installed onto a virtual machine. This provides built-in availability of your virtual infrastructure.

Below are the most utilized building blocks for production installations.

Cisco UCS S3260 – Dual Server Module (components are per server module)	
Boot Volume	2x 1.6TB 2.5” SATA SSD
SSD (MetaData)	2x 800 GB 2.5” SATA SSD (3x DWPD)
HDD (Data)	26x 10TB 3.5” 512e NL-SAS
RAM	192GB
CPU	2x E5-2620 v4 (2.1 GHz/6 cores)
Network	1x dual port 40Gbps Cisco VIC 1387
Disk Controller	Cisco 12Gbps Modular RAID PCIe Gen 3.0

Cisco UCS S3260 – Single Server Module	
Boot Volume	2x 1.6TB 2.5” SATA SSD
SSD (MetaData)	4x 800 GB 2.5” SATA SSD (3x DWPD)
HDD (Data)	56x 10TB 3.5” 512e NL-SAS
RAM	256GB
CPU	2x E5-2620 v4 (2.1 GHz/6 cores)
Network	1x dual port 40Gbps Cisco VIC 1387
Disk Controller	Cisco 12Gbps Modular RAID PCIe Gen 3.0

Cisco UCS S3260 – Single Server Module	
Boot Volume	2x 960GB 2.5” SATA SSD
SSD (MetaData)	2x 480 GB 2.5” SATA SSD (3x DWPD)
HDD (Data)	10x 10TB 3.5” 512e NL-SAS

Cisco UCS S3260 – Single Server Module	
RAM	128GB
CPU	2x E5-2620 v4 (2.1 GHz/6 cores)
Network	1x dual port 40Gbps Cisco VIC 1387
Disk Controller	Cisco 12Gbps Modular RAID PCIe Gen 3.0

Appendix E – Other Best Practices to Consider

- Configuration and testing of Scality Storage server for Data(10TB) HDDs and Metadata (800G) SSDs completed using JBOD mode, however best practices for current RING software states that data drives should be configured as individual R0 volumes to take advantage of the write cache benefits.
- Configuration and testing of Scality Storage facing Networks (Storage-Mgmt & Storage-Cluster) completed using jumbo frames MTU9000, however best practices for current RING software states that Storage-Mgmt traffic should be configured as MTU1500 & Storage-Cluster traffic as MTU9000.
- Scality Storage Server logs expected to grow larger than the configured boot SSDs (2 x 480G), hence the recommended disk specification is "2x 1.6TB 2.5" SATA SSD".

Appendix F – How to Order Using Cisco UCS S3260 + Scality Solution IDs

Cisco UCS S3260 bundles are created to provide ease-of-order using S3260 solution IDs created for Cisco-Scality solution. Solution IDs provide a single SKU like mechanism and it helps in ordering the solution from CCW in a timely fashion. Various S3260 bundles are available on the Cisco Commerce Workspace (CCW) page to provide guidance on configuring and ordering a Cisco-Scality solution with different configuration sizes based on our validation. The following are the solution IDs available:

1. Scality-Scale-Out-Small.
2. Scality-Scale-Out –Medium.
3. Scality-Scale-Out –Large.

To see these solution IDs, please visit the CCW (Cisco Commerce Workspace) page.

About the Authors

Vijay Durairaj, Technical Marketing Engineer in Cisco UCS and Data Center Solutions Group, Cisco Systems, Inc.

Vijay has over 13 years of experience in IT Infrastructure, Server Virtualization, and Cloud Computing. His current role includes building cloud computing solutions, software defined storage solutions, and performance benchmarking on Cisco UCS platforms. Vijay also holds Cisco Unified Computing Design Certification.

Christopher Donohoe, Scality

Christopher Donohoe is Scality's Partner Integration Engineer. He acts as a liaison between Scality's engineering community and the technical resources of partners like Cisco, implementing and testing new solutions prior to general availability. Christopher performs a great deal of the hands-on work in documents like the CVD, while designing and architecting new automated processes for performance benchmarking and new feature validation

Chris Moberly, Scality

Chris Moberly leads the technical initiatives within Scality's Strategic Alliances group. His main focus is assisting partners like Cisco in solving their customers' petabyte-scale challenges. Chris maintains certifications with Red Hat and Microsoft, as well as running the online learning programs at Scality.

Acknowledgements

- Ulrich Kleidon, Cisco Systems, Inc.
- Jawwad Memon, Cisco Systems, Inc.
- Lionel Mirafuente, Scality
- Trevor Benson, Scality