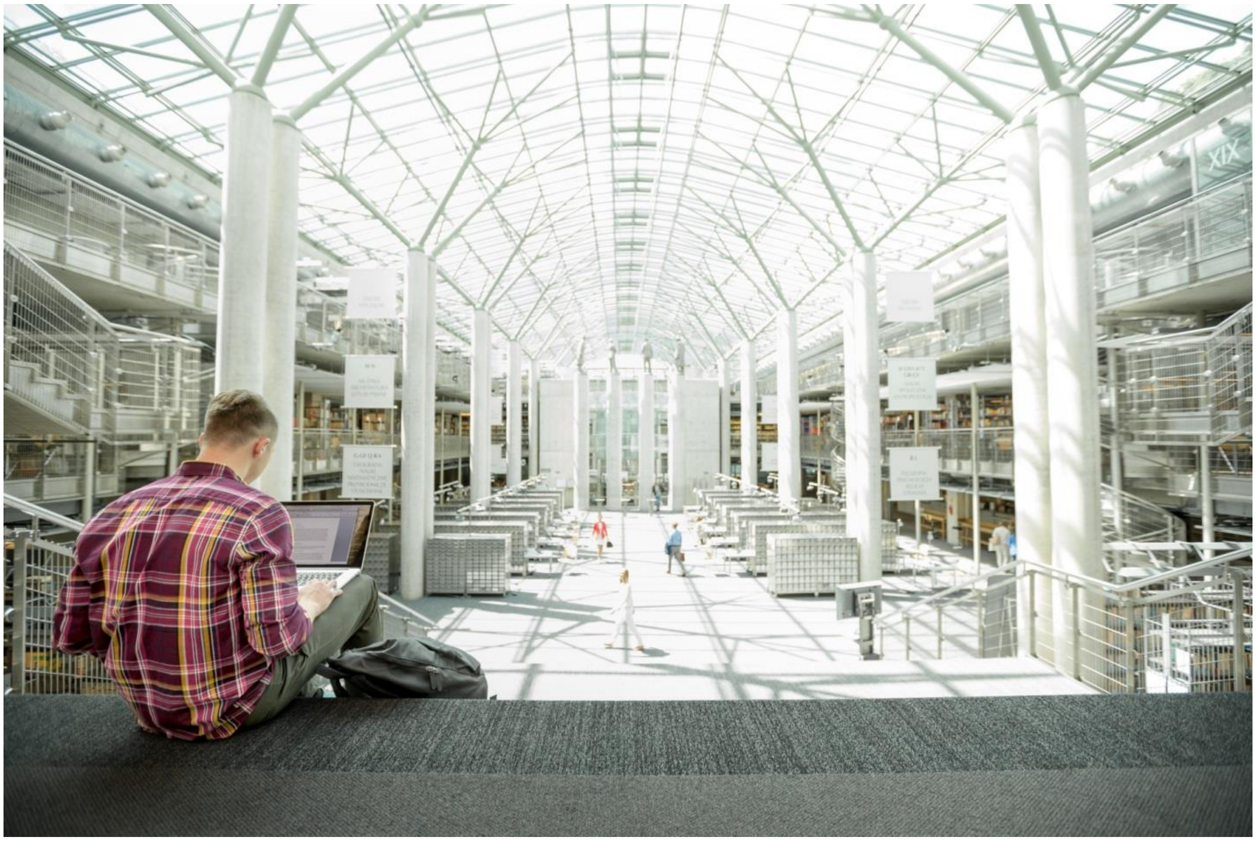


Cisco UCS S3260 M5 with SwiftStack

Cisco UCS S3260 M5 Server with SwiftStack Object Storage Deployment Guide

Last Updated: June 7, 2019

About the Cisco Validated Design Program

The Cisco Validated Design (CVD) program consists of systems and solutions designed, tested, and documented to facilitate faster, more reliable, and more predictable customer deployments. For more information, go to:

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

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

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

Table of Contents

Executive Summary

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

Most of the modern data centers are moving away from traditional file system type storage, to object storages. Object storage offers simple management, unlimited scalability and custom metadata for objects. With its low cost per gigabyte of storage, Object storage systems are suited for archive, backup, Life sciences, video surveillance, healthcare, multimedia, message and machine data, and so on.

SwiftStack object storage is a scalable software-defined storage system that can achieve enterprise class reliability, scale-out capacity, and lower costs with industry standard server solution.

The Cisco UCS S3260 Storage Server, originally designed for the data center, makes it an excellent fit for unstructured data workloads such as backup, archive, and cloud data. The Cisco UCS S3260 delivers a complete infrastructure with exceptional scalability for computing and storage resources together with 40 Gigabit Ethernet networking.

The reference architecture described in this document is a realistic use case for deploying SwiftStack Object Storage on Cisco UCS S3260 Storage Server and Cisco UCS C220 Rack-Mounted Server. This document explains how to setup the Cisco UCS hardware for SwiftStack Object and Controller nodes, install Red Hat Linux Operating system and SwiftStack Software, along with performance data collected to provide scale-up and scale-down guidelines. Also, any discovered issues and workarounds evolved during the installation, what needs to be done to leverage high availability from both hardware and software for business continuity, lessons learnt, best practices evolved while validating the solution, are described in this document.

Solution Overview

Introduction

Object storage is a highly scalable system for organizing and storing data objects. Object storage does not use a file system structure, instead it ingests data as objects with unique keys into a flat directory structure and the metadata is stored with the objects instead of hierarchical journal or tree. Search and retrieval is performed via these unique keys for searching. Most of the newly generated data is unstructured today. 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.

SwiftStack is an on-premises, scale out, and geographically distrubted software defined object and file storage that starts from 10s of terabytes and expands to 100s of petabytes.

Together with Cisco UCS, SwiftStack Storage delivers a fully enterprise-ready solution that can manage different workloads and remain flexible while scaling up seamless. The Cisco UCS S3260 Storage Server is an excellent platform to use with the main types of SwiftStack workloads, such as capacity-optimized and performance-optimized workloads. It is also excellent for workloads with a large number of I/O operations per second and scales well for varying work load and block sizes.

This document describes the architecture, design and deployment procedures of SwiftStack object storage on Cisco UCS S3260 servers with 2 x Cisco UCS C220 M5 rack servers.

Audience

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

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

Purpose of this Document

This document explains how to install SwiftStack on the Cisco UCS platform. It also covers high availability use cases, performance and scalability tests, workarounds, if any evolved while validating the design along with operational best practices.

Solution Summary

This solution is focused on SwiftStack storage on Red Hat Linux 7 on Cisco Unified Computing System. The advantages of Cisco UCS and SwiftStack combine to deliver an object storage solution that is simple to install, scalable and performant. The configuration uses the following components for the deployment:

- Cisco Unified Computing System (Cisco UCS)

  - Cisco UCS 6332 Series Fabric Interconnects
  - Cisco UCS S3260 storage servers.
  - Cisco UCS S3260 system IO controller with VIC 1380
  - Cisco UCS C220 M5 servers with VIC 1387
- Cisco Nexus C9332PQ Series Switches
- SwiftStack storage 6.x.
- Red Hat Enterprise Linux 7.5

The solution includes the following features:

- Infrastructure for large scale object storage
- Design and Implementation of a SwiftStack Object Storage solution on Cisco UCS S3260 Storage Server
- Simplified infrastructure management with Cisco UCS Manager
- Architectural scalability – linear scaling based on network, storage, and compute requirements

The scope is limited to the infrastructure pieces of the solution. However, an attempt has been made to add any discoveries made as part of the validation.

Technology Overview

Cisco Unified Computing System

The Cisco Unified Computing System 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 scalable family. The Cisco UCS servers offer the patented Cisco Extended Memory Technology to support applications with large datasets and allow more virtual machines 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, 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.

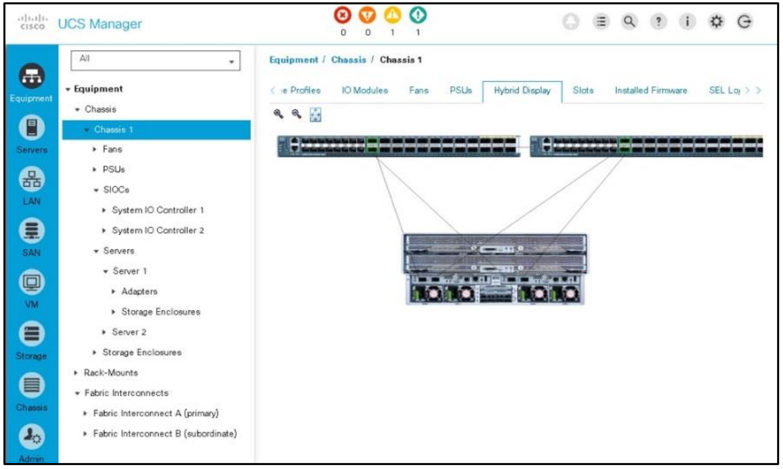
Cisco Unified Computing System is designed to deliver:

- 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 Manager

Cisco UCS Manager (UCSM) provides a unified, embedded management of all software and hardware components of the Cisco Unified Computing System across multiple chassis, rack servers, and thousands of virtual machines. It supports all Cisco UCS product models, including Cisco UCS B-Series Blade Servers, Cisco UCS C-Series Rack-Mount 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 1 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 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.

Cisco UCS 6300 Fabric Interconnects

The Cisco UCS 6300 Series Fabric Interconnects are a core part of Cisco UCS, providing both network connectivity and management capabilities for the system. 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 2 Cisco UCS 6300 Fabric Interconnect

The Cisco UCS 6300 Series provides the management and communication backbone for the Cisco UCS B-Series Blade Servers, Cisco UCS 5100 Series Blade Server Chassis, and Cisco UCS 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.

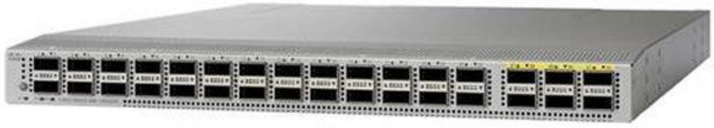
The Cisco UCS 6332 32-Port Fabric Interconnect is a 1-rack-unit (1RU) Gigabit Ethernet, and FCoE switch offering up to 2.56 Tbps throughput and up to 32 ports. The switch has 32 fixed 40-Gbps Ethernet and FCoE ports.

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 UCS 9332 Nexus Switches

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 3 Cisco Nexus 9332 Switch



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

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

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

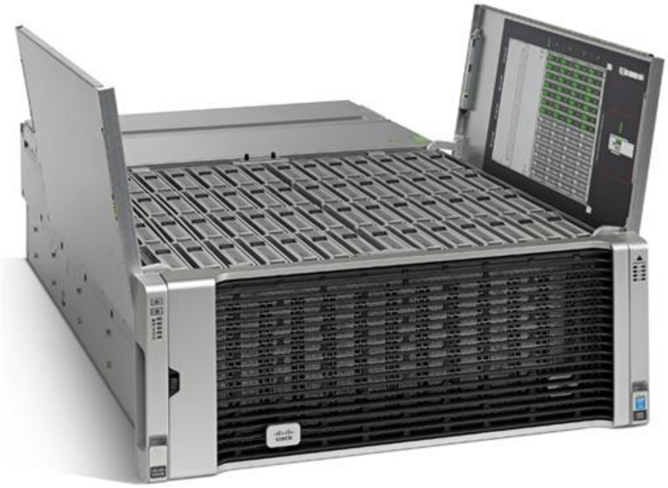
With the Cisco Nexus 9000 Series, organizations can quickly and easily upgrade existing data centers to carry 40 Gigabit Ethernet to the aggregation layer or to the spine (in a leaf-and-spine configuration) through advanced and cost-effective optics that enable the use of existing 10 Gigabit Ethernet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

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

Cisco UCS S3260 M5 Storage Server

The Cisco UCS S3260 Storage Server 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 4 Cisco UCS S3260 Storage Server



Extending the capability of the Cisco UCS C3000 portfolio, the Cisco UCS S3260 helps you achieve the highest levels of data availability. With dual-node capability that is based on the Intel Xeon scalable processors, 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-Server.

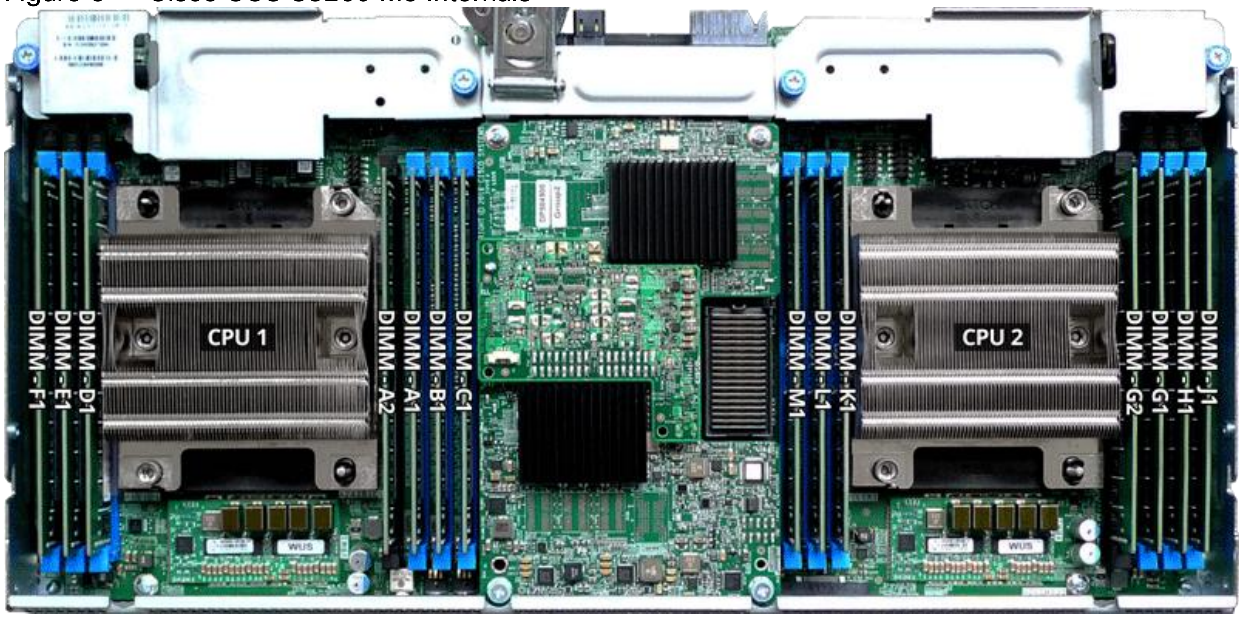
The Cisco UCS S3260 is deployed as a standalone server in both bare-metal or virtualized environments. Its modular architecture reduces TCO by allowing you to upgrade individual components over time and as use cases evolve, without having to replace the entire system.

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 the following:

- Dual server nodes
- Up to 44 computing cores per server node
- Up to 60 drives mixing a large form factor (LFF) with up to 28 solid-state disk (SSD) drives plus 2 SSD SATA boot drives per server node
- Up to 1.5 TB of memory per server node (3 TB Total) with 128GB DIMMs
- Support for 12-Gbps serial-attached SCSI (SAS) drives
- A system I/O Controller either with HBA Passthrough or RAID controller, with DUAL LSI 3316 Chip

- 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
- Dual 7mm NVMe - Capacity points: 512G, 1TB and 2TB
- G Host Management Port

Figure 5 Cisco UCS S3260 M5 Internals



Cisco UCS C220 M5 Rack-Mount Server

The Cisco UCS C220 M5 Rack-Mount Server is among the most versatile general-purpose enterprise infrastructure and application servers in the industry. It is a high-density 2-socket rack server that delivers industry-leading performance and efficiency for a wide range of workloads, including virtualization, collaboration, and bare-metal applications. The Cisco UCS C-Series Rack-Mount Servers can be deployed as standalone servers or as part of Cisco UCS to take advantage of Cisco's standards-based unified computing innovations that help reduce customers' TCO and increase their business agility.

The Cisco UCS C220 M5 server extends the capabilities of the Cisco UCS portfolio in a 1-Rack-Unit (1RU) form factor. It incorporates the Intel® Xeon Scalable processors, supporting up to 20 percent more cores per socket, twice the memory capacity, 20 percent greater storage density, and five times more PCIe NVMe Solid-State Disks (SSDs) compared to the previous generation of servers. These improvements deliver significant performance and efficiency gains that will improve your application performance.

Figure 6 Cisco UCS C220 M5 Rack-Mount Server

The Cisco UCS C220 M5 SFF server extends the capabilities of the Cisco Unified Computing System portfolio in a 1U form factor with the addition of the Intel Xeon Processor Scalable Family, 24 DIMM slots for 2666MHz DIMMs and capacity points up to 128GB, two 2 PCI Express (PCIe) 3.0 slots, and up to 10 SAS/SATA hard disk drives

(HDDs) or solid state drives (SSDs). The Cisco UCS C220 M5 SFF server also includes one dedicated internal slot for a 12G SAS storage controller card.

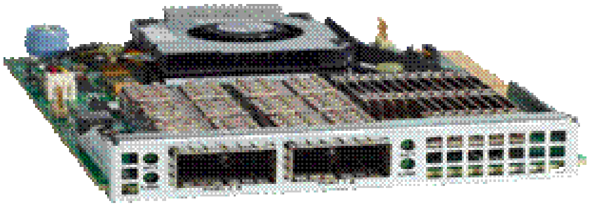
The Cisco UCS C220 M5 server included one dedicated internal modular LAN on motherboard (mLOM) slot for installation of a Cisco Virtual Interface Card (VIC) or third-party network interface card (NIC), without consuming a PCI slot, in addition to 2 x 10Gbase-T Intel x550 embedded (on the motherboard) LOM ports.

The Cisco UCS C220 M5 server can be used standalone, or as part of the Cisco Unified Computing System, which unifies computing, networking, management, virtualization, and storage access into a single integrated architecture enabling end-to-end server visibility, management, and control in both bare metal and virtualized environments.

Cisco UCS Virtual Interface Card 1387

The Cisco UCS Virtual Interface Card (VIC) 1387 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 7 Cisco UCS VIC 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.

Red Hat Enterprise Linux 7.5

Red Hat® Enterprise Linux is a high-performing operating system that has delivered outstanding value to IT environments for more than a decade. More than 90 percent of Fortune Global 500 companies use Red Hat products and solutions including Red Hat Enterprise Linux. As the worlds 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.

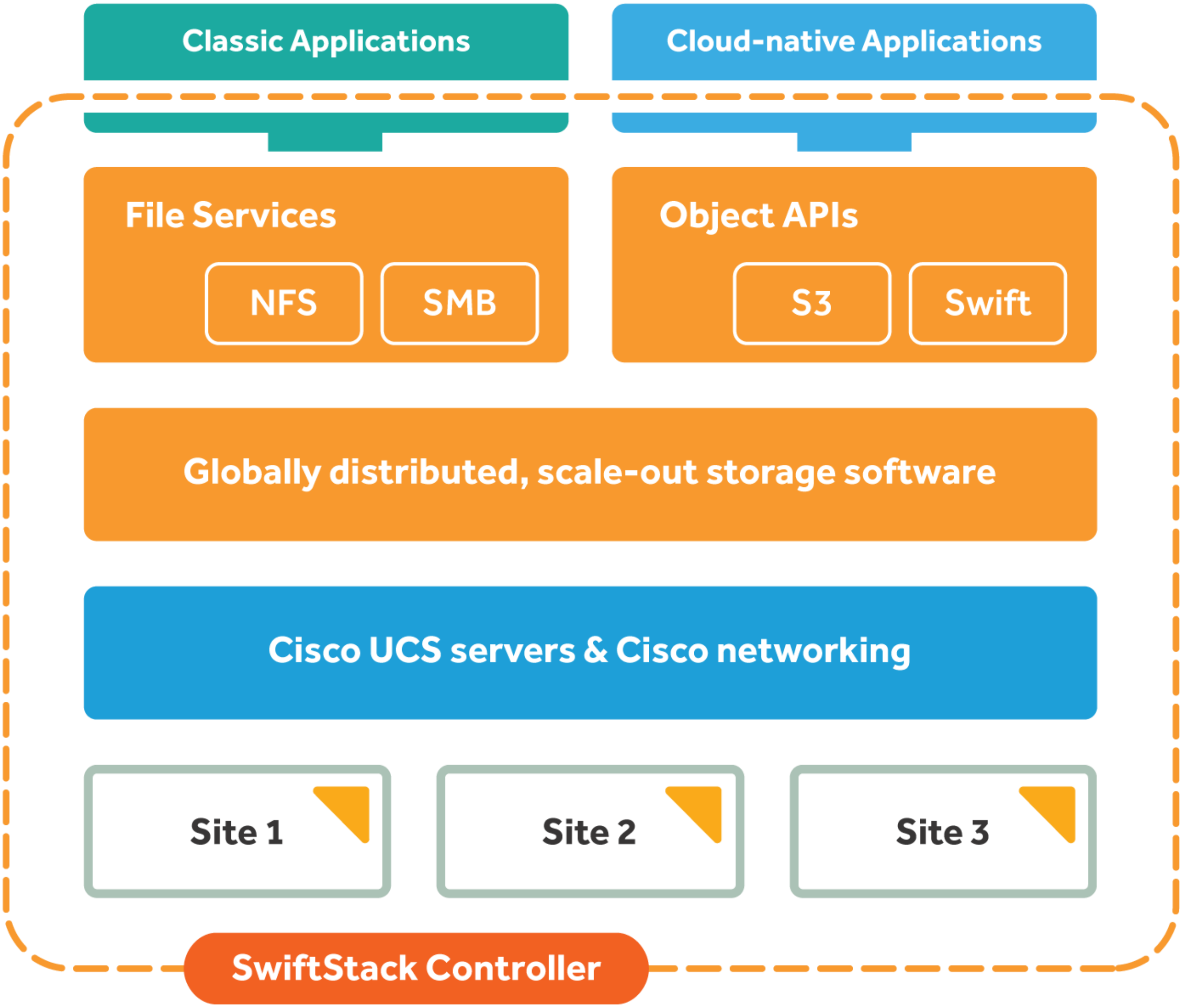
Red Hat Enterprise Linux:

- 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.

SwiftStack Object Storage 6.x

With SwiftStack software running on Cisco UCS S-Series servers, you get hybrid cloud storage enabling freedom to move workloads between clouds with universal access to data across on-premises and public infrastructure. SwiftStack was built from day one to have the fundamental attributes of the cloud—like a single namespace across multiple geographic locations, policy-driven placement of data, and consumption-based pricing.


Classic Applications
Cloud-native Applications
File Services
NFS
SMB
Object APIs
S3
Swift
Globally distributed, scale-out storage software
Cisco UCS servers & Cisco networking
Site 1
Site 2
Site 3
SwiftStack Controller

SwiftStack storage is optimized for unstructured data, which is growing at an ever-increasing rate inside most thriving enterprises. When AI-ML data, media assets, scientific research data, and even backup archives live in a multi-tenant storage cloud, utilization of this valuable data increases while driving out unnecessary costs.

SwiftStack is a fully-distributed storage system that horizontally scales to hold your data today and tomorrow. It scales linearly, allowing you to add additional capacity and performance independently...whatever your applications need.

While scaling storage is typically complex, it's not with SwiftStack. No advanced configuration is required. It takes only a few simple commands to install software on a new Cisco UCS S3260 server and deploy it in the cluster. Load balancing capabilities are fully integrated, allowing applications to automatically take advantage of the distributed cluster.

Powered by OpenStack Swift at the core, with SwiftStack, you get to utilize what drives some of the largest storage clouds and leverage the power of a vibrant community. SwiftStack is the lead contributor to the Swift project that has over 220 additional contributors worldwide. Having an engine backed by this community and deployed in demanding customer environments makes SwiftStack the most proven, enterprise-grade object storage software available.

Key SwiftStack features for an active archive:

- Starts as small as 120TB, and scales to 100s of PB
- Spans multiple data centers while still presenting a single namespace
- Handles data according to defined policies that align to the needs of different applications
- Uses erasure coding and replicas in the same cluster to protect data
- Offers multi-tenant support with authentication via Active Directory, LDAP, and Keystone
- Supports file protocols (SMB, NFS) and object APIs (S3, Swift) simultaneously
- Automatically synchronizes to Google Cloud Storage and Amazon S3 with the Cloud Sync feature
- Encrypts data and metadata at rest
- Manages highly scalable storage infrastructure via centralized out-of-band controller
- Ensures all functionality touching data is open by leveraging an open-source core
- Optimizes TCO with pay-as-you-grow licensing with support and maintenance included

Solution Design

SwiftStack Core Storage Architecture

SwiftStack provides both native Object API (S3 and Swift) and file-based (SMB and NFS) access to the data stored in the SwiftStack Cluster. SwiftStack is a fully-distributed storage system that horizontally scales to hold your data today and tomorrow. The storage system scales linearly, allowing you to add additional capacity and performance independently.

SwiftStack is designed to withstand hardware failures without any downtime. Even major disasters, since nodes of the cluster can be globally distributed.

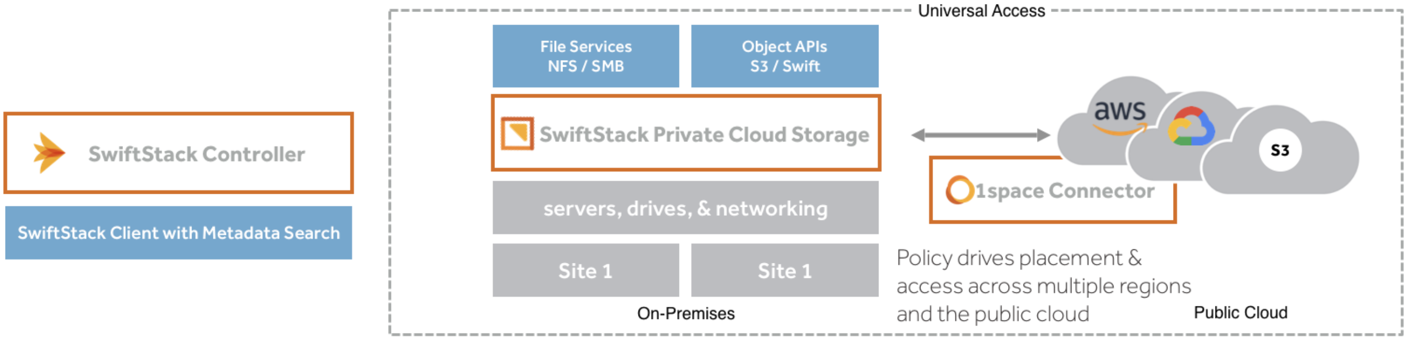
It is most common that data is replicated to multiple regions of a globally distributed cluster for high levels of durability and availability. With this distributed architecture, standard cluster configurations have between 9 and 14 nines availability, significantly higher than SANs in tier 1 data centers.

This redundancy allows you to maintain, upgrade, and enhance the storage system while in flight. For example, one click and all of the steps needed to perform rolling updates across all nodes happens automatically in the background.

SwiftStack Nodes includes 4 different roles to handle different services in SwiftStack Cluster (Group of SwiftStack Nodes) called as PACO – P: Proxy, A: Account, C: Container and O: Object. In most deployments, all four services are deployd and run on a single physical node.

SwiftStack Architecture

SwiftStack solution is enterprise-grade object storage, with OpenStack Swift at the core. It has been deployed at 100s of companies with massive amounts of data stored. It includes three major components, SwiftStack Storage Nodes, 1space and the SwiftStack Controller. 1space allows for policy driven placement and access across multiple public clouds, while the SwiftStack Controller is an out-of-band management system that manages one or more SwiftStack storage clusters.

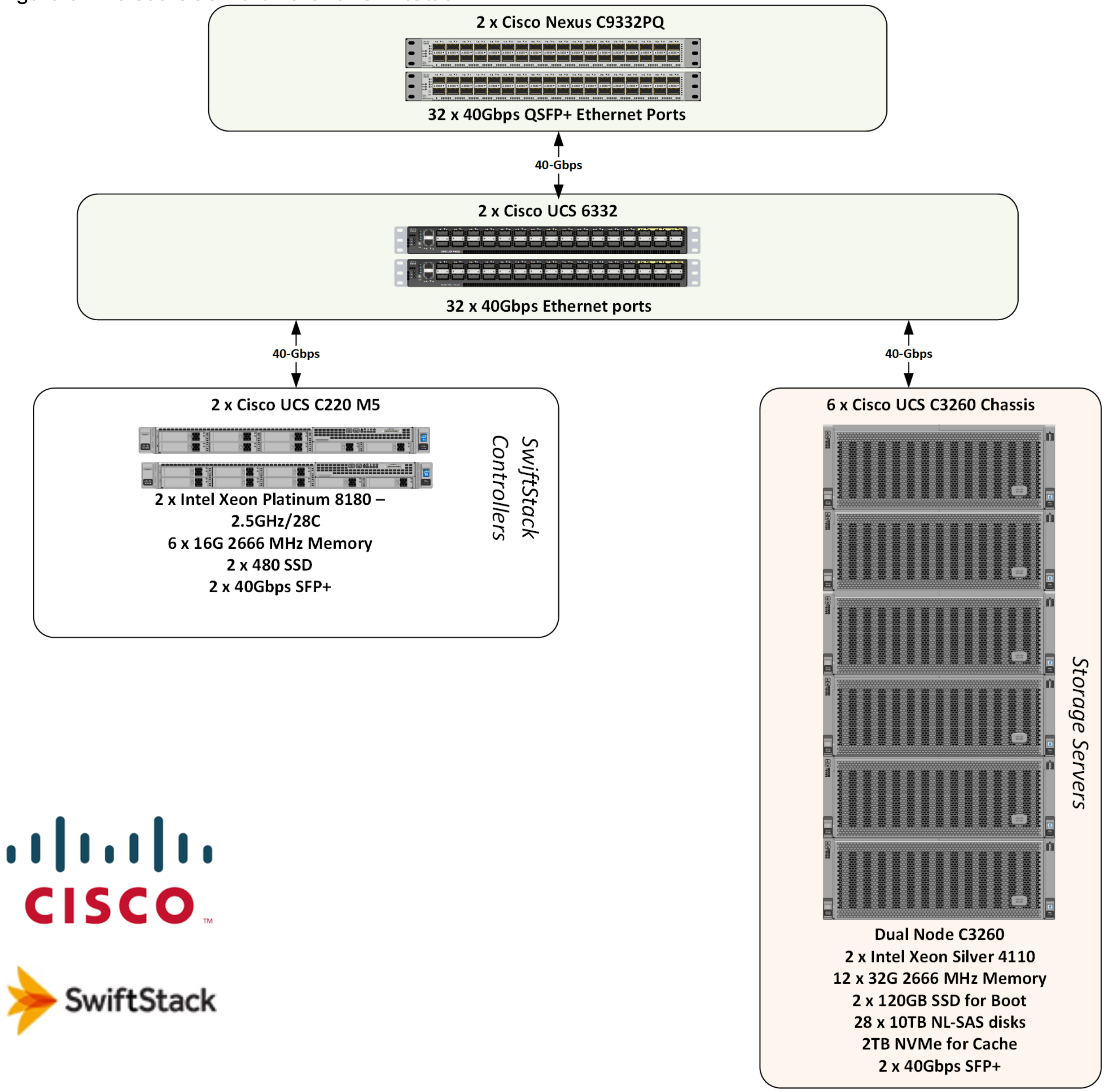

Universal Access
File Services NFS / SMB
Object APIs S3 / Swift
SwiftStack Controller
SwiftStack Client with Metadata Search
SwiftStack Private Cloud Storage
servers, drives, & networking
Site 1
Site 1
On-Premises
aws
S3
1space Connector
Policy drives placement & access across multiple regions and the public cloud
Public Cloud

The reference architecture use case provides a comprehensive, end-to-end example of designing and deploying SwiftStack object storage on Cisco UCS S3260 as shown in Figure 8. This document describes the architecture and design of a SwiftStack Scale-out object storage and file system solution on three Cisco UCS S3260 Storage Server Chassis each with two Cisco UCS S3260 M5 nodes configured as storage servers and Two Cisco UCS C220 M5S rack server as SwiftStack Controllers node. The whole solution is connected to a pair of Cisco UCS 6332 Fabric Interconnects and a pair of upstream network Cisco Nexus 9332PQ switches.

The configuration is comprised of the following:

- 2 x Cisco Nexus 9332PQ Switches
- 2 x Cisco UCS 6332 Fabric Interconnects
- 6 x Cisco UCS S3260 Storage Servers with 2 x Cisco UCS C3260 M5 server nodes each
- 2 x Cisco UCS C220 M5 Rack Servers

Figure 8 Cisco UCS Hardware for SwiftStack



System Hardware and Software Specifications

Solution Overview

This solution is based on Cisco UCS and SwiftStack object storage.

Software Versions

Table 1 Software Versions

Layer	Component	Version or Release
Storage (Chassis) UCS S3260	Chassis Management Controller	4.0(1a)
	Shared Adapter	4.0(1a)
Compute (Server Nodes) UCS S3X60 M5	BIOS	4.0(1a)
	CIMC Controller	4.0(1a)
Compute (Rack Server) C220 M5S	BIOS	C220M5.4.0.1c
	CIMC Controller	4.0(1a)
Network 6332 Fabric Interconnect	UCS Manager	4.0(1a)
	Kernel	5.0(3)N2(4.01a)
	System	5.0(3)N2(4.01a)
Network Nexus 9332PQ	BIOS	07.51
	NXOS	7.0(3)I1(3)
Software	Red Hat Enterprise Linux Server	7.5 (x86_64)
	SwiftStack Software	6.x

Hardware Requirements and Bill of Materials

Table 2 Bill of Materials

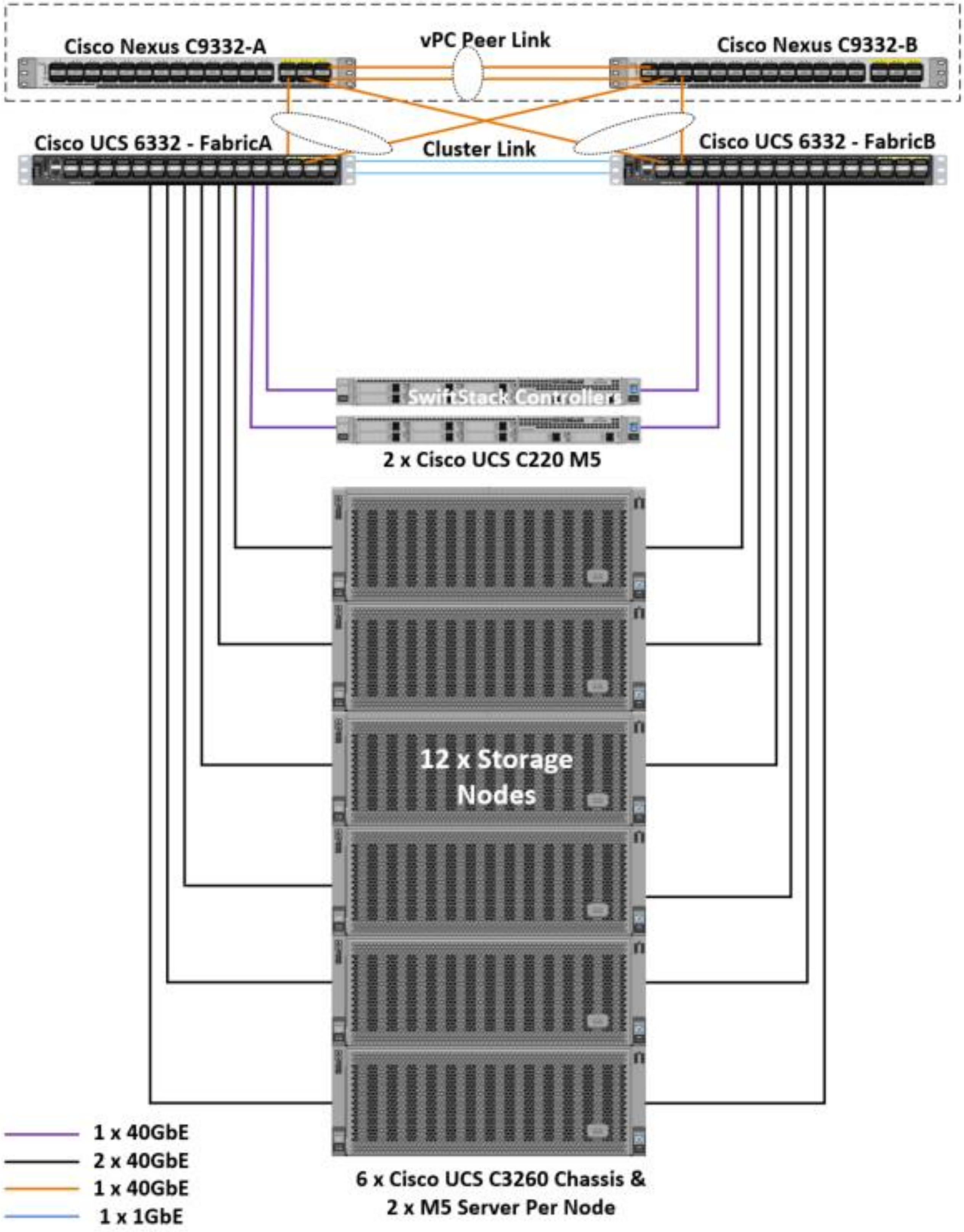
Component	Model	Quantity	Comments
SwiftStack Storage Nodes	Cisco UCS S3260 M5 Chassis	6	2 x UCS S3X60 M5 Server Nodes per Chassis (Total = 6nodes) Per Server Node – 2 x Intel Xeon Silver 4110 (2.1GHz/8cores), 192 GB RAM – Cisco 12G RAID Controller – 2 x SSD for OS – 28 x 10TB HDDs for Data,

Component	Model	Quantity	Comments
			– 1 NVMe for metadata per node. – Dual-port 40 Gbps VIC
SwiftStack Controller Nodes	Cisco UCS C220 M5S Rack server	2	2 x Intel Xeon Silver 4110 (2.1GHz/8 Cores), 96GB RAM Cisco 12G SAS RAID Controller 2 x 600GB 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

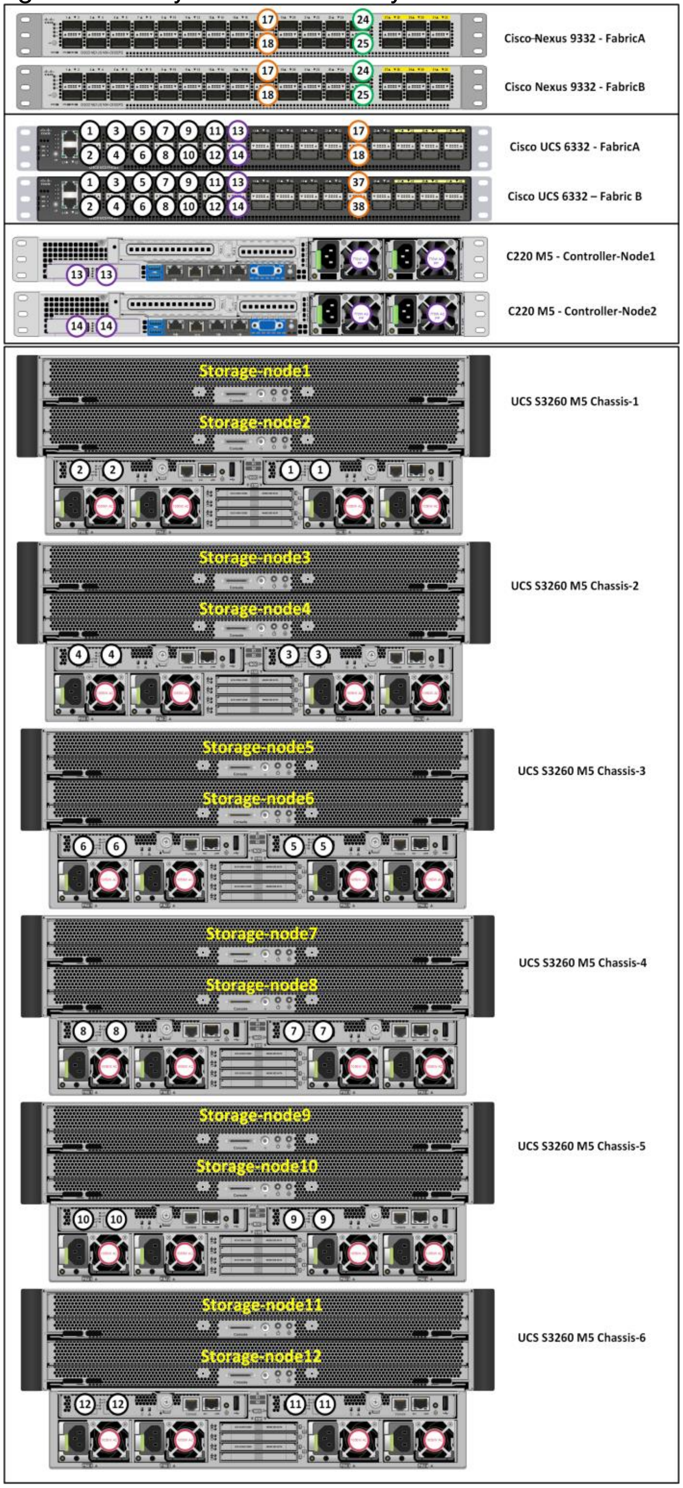
The following sections describe the physical design of the solution and the configuration of each component.

Figure 9 Physical Topology of this Solution



The connectivity of the solution is based on 40 Gbit. All components are connected with 40 QSFP cables. Between both Cisco Nexus 9332PQ switches are 2 x 40 Gbit cabling. Each Cisco UCS 6332 Fabric Interconnect is connected with 2 x 40 Gbit to each Cisco UCS 9332PQ switch, and each Cisco UCS C220 M5 is connected with 1 x 40 Gbit and each Cisco UCS S3260 M5 server is connected with 2 x 40 Gbit cable to each Fabric Interconnect. The architecture is highly redundant and system survived with little or no impact to applications under various failure test scenarios which will be covered during validation and testing.

Figure 10 Physical Connectivity of this Solution



The exact cabling for the Cisco UCS S3260 Storage Server, Cisco UCS C220 M5, 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	Eth1/24	40GbE	Cisco Nexus 9332 Switch B	Eth1/24	QSFP-H40G-CU1M
	Eth1/25	40GbE	Cisco Nexus 9332 Switch B	Eth1/25	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
	Eth1/26	40GbE	Top of Rack (Upstream Network)	Any	QSFP+ 4SFP10G
	MGMT0	1GbE	Top of Rack (Management)	Any	1G RJ45
Cisco Nexus 9332 Switch B	Eth1/24	40GbE	Cisco Nexus 9332 Switch A	Eth1/24	QSFP-H40G-CU1M
	Eth1/25	40GbE	Cisco Nexus 9332 Switch A	Eth1/25	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
	Eth1/27	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
	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 -	port 1	QSFP-H40G-

Local Device	Local Port	Connection	Remote Device	Remote Port	Cable
			SIOC 1 (right)		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
	Eth1/13	40GbE	C220 M5 - Server1 - VIC1387	VIC - Port 1	QSFP-H40G-CU1M
	Eth1/14	40GbE	C220 M5 – Server2 - VIC1387	VIC - Port 1	QSFP-H40G-CU1M
	Eth1/17	40GbE	Nexus 9332 A	Eth 1/17	QSFP-H40G-CU1M
	Eth1/18	40GbE	Nexus 9332 B	Eth 1/17	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 -	port 2	QSFP-H40G-

Local Device	Local Port	Connection	Remote Device	Remote Port	Cable
			SIOC 2 (left)		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
	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
	Eth1/13	40GbE	C220 M5 - Server1 - VIC1387	VIC -Port2	QSFP-H40G-CU1M
	Eth1/14	40GbE	C220 M5 – Server2 - VIC1387	VIC -Port2	QSFP-H40G-CU1M
	Eth1/17	40GbE	Nexus 9332 A	Eth 1/18	QSFP-H40G-CU1M
	Eth1/18	40GbE	Nexus 9332 B	Eth 1/18	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

Network Design and Architecture

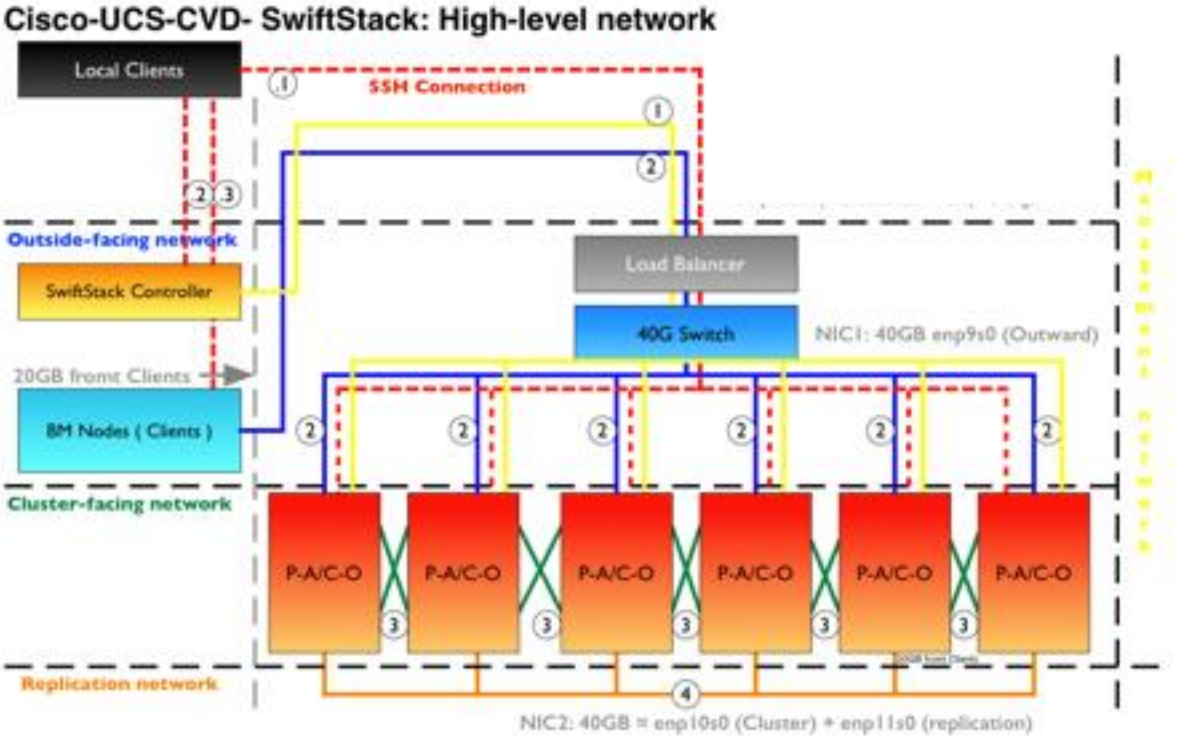
The following network architecture has been evaluated on the test bed. Each server node has 2x40Gbps ports.

Table 4 Network Architecture Details

Interface	Purpose	Physical Port on Adapter	Network Capacity	Network
eth0	Management-Network	Port 1	40Gbps	192.168.100.xxx
eth1	External/OS interface	Port 1		173.xxx.xxx.xxx

Interface	Purpose	Physical Port on Adapter	Network Capacity	Network
eth2	Client/Outward facing interface	Port 1		192.168.120.xxx
eth3	Cluster network interface	Port 2	40Gbps	192.168.130.xxx
eth4	Replication network interface	Port 2		192.168.150.xxx

Figure 11 Network Flow Diagram



The SwiftStack controllers are not in the data path; configuring the external/outward facing interface on the Controller is optional.

Deployment Hardware and Software

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 follow these 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 : 192.168.100.8
    Mgmt0 IPv4 netmask : 255.255.255.0
  Configure the default gateway? (yes/no) [y]: yes
    IPv4 address of the default gateway : 192.168.100.1
  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 : 192.168.100.220
  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 0.0.0.0/0 192.168.100.1
exit
  no feature telnet
  ssh key rsa 1024 force
  feature ssh
  ntp server 192.168.100.220
  no system default switchport
  system default switchport shutdown
  copp profile strict
interface mgmt0
ip address 192.168.100.8 255.255.255.0
```

```
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:
```

Repeat these steps for the Nexus 9332PQ Switch B with the exception of configuring a different IPv4 management address 192.168.100.9 as described in step 7.

Enable Features on Nexus 9332PQ Switch A and B

To enable the features UDLD, VLAN, HSRP, LACP, VPC, and Jumbo Frames, connect to the management interface through ssh on both switches and follow these 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
```

Configure VLANs on Nexus 9332PQ Switch A and B

To configure the same VLANs Storage-Management, Storage-Cluster, Client Network, and External Management as previously configured in the Cisco UCS Manager GUI, follow these 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 100
N9k-Fab-A(config-vlan)# name Management-Network
N9k-Fab-A(config-vlan)# no shut
```

```
N9k-Fab-A(config-vlan)# exit
N9k-Fab-A(config)# vlan 120
N9k-Fab-A(config-vlan)# name Swift-Client-Network
N9k-Fab-A(config-vlan)# no shut
N9k-Fab-A(config-vlan)# exit
N9k-Fab-A(config)# vlan 130
N9k-Fab-A(config-vlan)# name Swift-Cluster-Network
N9k-Fab-A(config-vlan)# no shut
N9k-Fab-A(config-vlan)# exit
N9k-Fab-A(config)# vlan 150
N9k-Fab-A(config-vlan)# name Swift-Replication-Network
N9k-Fab-A(config-vlan)# no shut
N9k-Fab-A(config-vlan)# exit
N9k-Fab-A(config)# vlan 219
N9k-Fab-A(config-vlan)# name External-Network
N9k-Fab-A(config-vlan)# no shut
N9k-Fab-A(config-vlan)# exit
N9k-Fab-A(config)# interface vlan100
N9k-Fab-A(config-if)# description Management-Network
N9k-Fab-A(config-if)# no shutdown
N9k-Fab-A(config-if)# no ip redirects
N9k-Fab-A(config-if)# ip address 192.168.100.253/24
N9k-Fab-A(config-if)# no ipv6 redirects
N9k-Fab-A(config-if)# hsrp version 2
N9k-Fab-A(config-if)# hsrp 100
N9k-Fab-A(config-if-hsrp)# preempt
N9k-Fab-A(config-if-hsrp)# priority 10
N9k-Fab-A(config-if-hsrp)# ip 192.168.100 .1
N9k-Fab-A(config-if-hsrp)# exit
N9k-Fab-A(config)# interface vlan120
N9k-Fab-A(config-if)# description swift-client
N9k-Fab-A(config-if)# no shutdown
N9k-Fab-A(config-if)# no ip redirects
N9k-Fab-A(config-if)# ip address 192.168.120.253/24
N9k-Fab-A(config-if)# no ipv6 redirects
N9k-Fab-A(config-if)# hsrp version 2
N9k-Fab-A(config-if)# hsrp 120
N9k-Fab-A(config-if-hsrp)# preempt
N9k-Fab-A(config-if-hsrp)# priority 10
N9k-Fab-A(config-if-hsrp)# ip 192.168.120.1
N9k-Fab-A(config-if-hsrp)# exit
N9k-Fab-A(config-if)# exit
N9k-Fab-A(config)# interface vlan130
N9k-Fab-A(config-if)# description swift-cluster
N9k-Fab-A(config-if)# no shutdown
N9k-Fab-A(config-if)# no ip redirects
N9k-Fab-A(config-if)# ip address 192.168.130.253/24
N9k-Fab-A(config-if)# no ipv6 redirects
N9k-Fab-A(config-if)# hsrp version 2
N9k-Fab-A(config-if)# hsrp 130
N9k-Fab-A(config-if-hsrp)# preempt
N9k-Fab-A(config-if-hsrp)# priority 10
N9k-Fab-A(config-if-hsrp)# ip 192.168.130.1
N9k-Fab-A(config-if-hsrp)# exit
N9k-Fab-A(config)# interface vlan150
N9k-Fab-A(config-if)# description swift-replication
N9k-Fab-A(config-if)# no shutdown
```

```
N9k-Fab-A(config-if)# no ip redirects
N9k-Fab-A(config-if)# ip address 192.168.150.253/24
N9k-Fab-A(config-if)# no ipv6 redirects
N9k-Fab-A(config-if)# hsrp version 2
N9k-Fab-A(config-if)# hsrp 150
N9k-Fab-A(config-if-hsrp)# preempt
N9k-Fab-A(config-if-hsrp)# priority 10
N9k-Fab-A(config-if-hsrp)# ip 192.168.150.1
N9k-Fab-A(config-if-hsrp)# exit
N9k-Fab-A(config)# interface vlan219
N9k-Fab-A(config-if)# description External_Network
N9k-Fab-A(config-if)# no shutdown
N9k-Fab-A(config-if)# no ip redirects
N9k-Fab-A(config-if)# ip address 173.36.219.117/24
N9k-Fab-A(config-if)# no ipv6 redirects
N9k-Fab-A(config-if)# exit
N9k-Fab-B(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 100
N9k-Fab-B(config-vlan)# name Management-Network
N9k-Fab-B(config-vlan)# no shut
N9k-Fab-B(config-vlan)# exit
N9k-Fab-B(config)# vlan 120
N9k-Fab-B(config-vlan)# name Swift-Client-Network
N9k-Fab-B(config-vlan)# no shut
N9k-Fab-B(config-vlan)# exit

N9k-Fab-B(config)# vlan 130
N9k-Fab-B(config-vlan)# name Swift-Cluster-Network
N9k-Fab-B(config-vlan)# no shut
N9k-Fab-B(config-vlan)# exit
N9k-Fab-B(config)# vlan 150
N9k-Fab-B(config-vlan)# name Swift-Replication-Network
N9k-Fab-B(config-vlan)# no shut
N9k-Fab-B(config-vlan)# exit
N9k-Fab-B(config)# vlan 219
N9k-Fab-B(config-vlan)# name External-Network
N9k-Fab-B(config-vlan)# no shut
N9k-Fab-B(config-vlan)# exit
N9k-Fab-B(config)# interface vlan100
N9k-Fab-B(config-if)# description Management-Network
N9k-Fab-B(config-if)# no ip redirects
N9k-Fab-B(config-if)# ip address 192.168.100.254/24
N9k-Fab-B(config-if)# no ipv6 redirects
N9k-Fab-B(config-if)# hsrp version 2
N9k-Fab-B(config-if)# hsrp 100
N9k-Fab-B(config-if-hsrp)# preempt
N9k-Fab-B(config-if-hsrp)# priority 5
N9k-Fab-B(config-if-hsrp)# ip 192.168.100.1
N9k-Fab-B(config-if-hsrp)# exit
N9k-Fab-B(config-if)# exit
N9k-Fab-B(config)# interface vlan120
N9k-Fab-B(config-if)# description swift-client
```

```
N9k-Fab-B(config-if)# no ip redirects
N9k-Fab-B(config-if)# ip address 192.168.120.254/24
N9k-Fab-B(config-if)# no ipv6 redirects
N9k-Fab-B(config-if)# hsrp version 2
N9k-Fab-B(config-if)# hsrp 120
N9k-Fab-B(config-if-hsrp)# preempt
N9k-Fab-B(config-if-hsrp)# priority 5
N9k-Fab-B(config-if-hsrp)# ip 192.168.120.1
N9k-Fab-B(config-if-hsrp)# exit
N9k-Fab-B(config-if)# exit

N9k-Fab-B(config)# interface vlan130
N9k-Fab-B(config-if)# description swift-cluster
N9k-Fab-B(config-if)# no ip redirects
N9k-Fab-B(config-if)# ip address 192.168.130.254/24
N9k-Fab-B(config-if)# no ipv6 redirects
N9k-Fab-B(config-if)# hsrp version 2
N9k-Fab-B(config-if)# hsrp 130
N9k-Fab-B(config-if-hsrp)# preempt
N9k-Fab-B(config-if-hsrp)# priority 5
N9k-Fab-B(config-if-hsrp)# ip 192.168.130.1
N9k-Fab-B(config-if-hsrp)# exit
N9k-Fab-B(config)# interface vlan150
N9k-Fab-B(config-if)# description swift-replication
N9k-Fab-B(config-if)# no ip redirects
N9k-Fab-B(config-if)# ip address 192.168.150.254/24
N9k-Fab-B(config-if)# no ipv6 redirects
N9k-Fab-B(config-if)# hsrp version 2
N9k-Fab-B(config-if)# hsrp 150
N9k-Fab-B(config-if-hsrp)# preempt
N9k-Fab-B(config-if-hsrp)# priority 5
N9k-Fab-B(config-if-hsrp)# ip 192.168.150.1
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 Nexus C9332PQ Switch A and B

To enable vPC and Port Channels on both Switch A and B, follow these 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 101
N9k-Fab-A(config-vpc-domain)# peer-keepalive destination 192.168.100.8
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 1/24
N9k-Fab-A(config-if)# description connected to peer N9k-Fab-B port 24
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 1/25
N9k-Fab-A(config-if)# description connected to peer N9k-Fab-B port 25
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 101
N9k-Fab-B(config-vpc-domain)# peer-keepalive destination 192.168.100.9
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 1/24
N9k-Fab-B(config-if)# description connected to peer N9k-Fab-A port 24
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 1/25
N9k-Fab-B(config-if)# description connected to peer N9k-Fab-A port 25
```

```
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 UCS Fabric A & B on Nexus Switch A
N9k-Fab-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-A(config)# interface port-channel 17
N9k-Fab-A(config-if)# description vPC for UCS FI-A ports 17 to 18
N9k-Fab-A(config-if)# vpc 17
N9k-Fab-A(config-if)# switchport
N9k-Fab-A(config-if)# switchport mode trunk
N9k-Fab-A(config-if)# switchport trunk allowed vlan 1,100,120,130,150,219
N9k-Fab-A(config-if)# spanning-tree port type edge trunk
Edge port type (portfast) should only be enabled on ports connected to a sin-
gle
 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 18
N9k-Fab-A(config-if)# description vPC for UCS FI-B ports 17 to 18
N9k-Fab-A(config-if)# vpc 18
N9k-Fab-A(config-if)# switchport
N9k-Fab-A(config-if)# switchport mode trunk
N9k-Fab-A(config-if)# switchport trunk allowed vlan 1,100,120,130,150,219
N9k-Fab-A(config-if)# spanning-tree port type edge trunk
Edge port type (portfast) should only be enabled on ports connected to a sin-
gle
 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 1/17
N9k-Fab-A(config-if-range)# switchport
N9k-Fab-A(config-if-range)# switchport mode trunk
N9k-Fab-A(config-if-range)# description Uplink from UCS FI-A ports 17
N9k-Fab-A(config-if-range)# channel-group 17 mode active
N9k-Fab-A(config-if)# exit

N9k-Fab-A(config)# interface ethernet 1/18
N9k-Fab-A(config-if-range)# switchport
N9k-Fab-A(config-if-range)# switchport mode trunk
N9k-Fab-A(config-if-range)# description Uplink from UCS FI-B ports 17
N9k-Fab-A(config-if-range)# channel-group 18 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 A and B on Nexus Switch B
N9k-Fab-B# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-Fab-B(config)# interface port-channel 17
N9k-Fab-B(config-if)# description vPC for UCS FI-A ports 17 to 18
N9k-Fab-B(config-if)# switchport
N9k-Fab-B(config-if)# switchport mode trunk
N9k-Fab-B(config-if)# switchport trunk allowed vlan 1,100,120,130,150,219
N9k-Fab-B(config-if)# spanning-tree port type edge trunk
Edge port type (portfast) should only be enabled on ports connected to a sin-
gle
 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 17
N9k-Fab-B(config-if)# mtu 9216
N9k-Fab-B(config-if)# exit

N9k-Fab-B(config)# interface port-channel 18
N9k-Fab-B(config-if)# description vPC for UCS FI-B ports 17 to 18
N9k-Fab-B(config-if)# switchport
N9k-Fab-B(config-if)# switchport mode trunk
N9k-Fab-B(config-if)# switchport trunk allowed vlan 1,100,120,130,150,219
N9k-Fab-B(config-if)# spanning-tree port type edge trunk
Edge port type (portfast) should only be enabled on ports connected to a sin-
gle
 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 27
N9k-Fab-B(config-if)# mtu 9216
N9k-Fab-B(config-if)# exit

N9k-Fab-B(config)# interface ethernet 1/17
N9k-Fab-B(config-if-range)# switchport
N9k-Fab-B(config-if-range)# switchport mode trunk
N9k-Fab-B(config-if-range)# description Uplink from UCS FI-A ports 17 to 18
N9k-Fab-B(config-if-range)# channel-group 17 mode active
N9k-Fab-B(config-if)# exit

N9k-Fab-B(config)# interface ethernet 1/18
N9k-Fab-B(config-if-range)# switchport
N9k-Fab-B(config-if-range)# switchport mode trunk
N9k-Fab-B(config-if-range)# description Uplink from UCS FI-B ports 17 to 18
N9k-Fab-B(config-if-range)# channel-group 18 mode active
N9k-Fab-B(config-if)# exit
N9k-Fab-B(config)# copy running-config startup-config
```

Verification Check of 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                     : 101
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : secondary
Number of vPCs configured         : 2
Peer Gateway                      : Enabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1,100,120,130,150,219

vPC status
----------------------------------------------------------------------
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- ------                     ------------
17   Po17   up     success     success                    1,100,120,1
                                                          30,150,219
18   Po18   up     success     success                    1,100,120,1

30,150,219

 N9K-Fab-A(config)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 101
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : secondary
Number of vPCs configured         : 2
Peer Gateway                      : Enabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
```

```
1    Po1    up     1,100,120,130,150,219

vPC status
----------------------------------------------------------------------
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- ------                     ------------
17   Po17   up     success     success                    1,100,120,1
                                                          30,150,219
18   Po18   up     success     success                    1,100,120,1
                                                          30,150,219
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      Eth1/24(P)   Eth1/25(P)
17    Po17(SU)    Eth      LACP      Eth1/17(P)
18    Po18(SU)    Eth      LACP      Eth1/18(P)
```

Switch B

```
N9K-Fab-B(config)# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 101
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         : 2
Peer Gateway                      : Enabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1    up     1,100,120,130,150,219

vPC status
----------------------------------------------------------------------
```

```
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- ------                     ------------
17   Po17   up     success     success                    1,100,120,1
                                                          30,150,219
18   Po18   up     success     success                    1,100,120,1
                                                          30,150,219
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      Eth1/24(P)   Eth1/25(P)
17    Po17(SU)    Eth      LACP      Eth1/17(P)
18    Po18(SU)    Eth      LACP      Eth1/18(P)
```

Fabric Interconnect Configuration

This section provides the details to configure 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

The following section describes the initial setup of the Cisco UCS 6332 Fabric Interconnects A and B.

Configure Fabric Interconnect A

To configure Fabric A, follow these steps:

1. Connect to the console port on the first Cisco UCS 6332 Fabric Interconnect.

2. At the prompt to enter the configuration method, enter **console** to continue.

3. If asked to either perform a new setup or restore from backup, enter **setup** to continue.

4. Enter **y** to continue to set up a new Fabric Interconnect.

5. Enter **n** to enforce strong passwords.

6. Enter the password for the admin user.

7. Enter the same password again to confirm the password for the admin user.

8. When asked if this fabric interconnect is part of a cluster, answer **y** to continue.

9. Enter **A** for the switch fabric.

10. Enter the cluster name 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 : 192.168.100.10
  Physical Switch Mgmt0 IPv4 netmask : 255.255.255.0
  IPv4 address of the default gateway : 192.168.100.1
  Cluster IPv4 address : 192.168.100.12
  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=192.168.100.10
    Physical Switch Mgmt0 IP Netmask=255.255.255.0
    Default Gateway=192.168.100.1
    Ipv6 value=0

    Cluster Enabled=yes
    Cluster IP Address=192.168.100.12
    NOTE: Cluster IP will be configured only after both Fabric Interconnects
are initialized.
          UCSM will be functional only after peer FI is configured in clus-
tering 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

To configure Fabric Interconnect B, follow these steps:

Make sure that L1/L2 ports are connected before proceeding.

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: 192.168.100.10
    Peer Fabric interconnect Mgmt0 IPv4 Netmask: 255.255.255.0
    Cluster IPv4 address          : 192.168.100.12

    Peer FI is IPv4 Cluster enabled. Please Provide Local Fabric Interconnect
Mgmt0 IPv4 Address
  Physical Switch Mgmt0 IP address : 192.168.100.11

  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:
```

Log into Cisco UCS Manager

To log into Cisco UCS Manager, follow these steps:

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

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

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

4. Click Launch UCS Manager HTML.

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

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

Configure NTP Server

To configure the NTP server for the Cisco UCS environment, follow these steps:

1. Select the Admin tab.

2. Select Time Zone Management.

3. Select Time Zone.

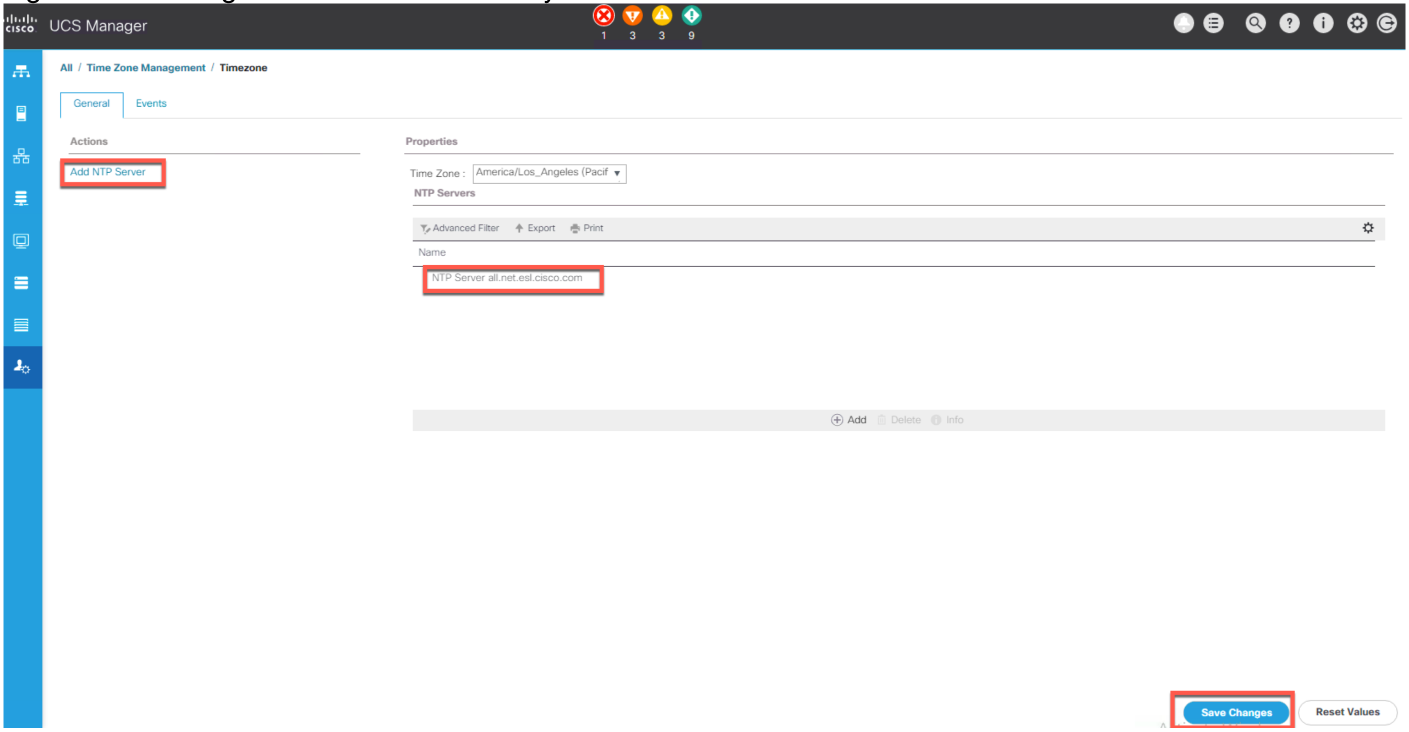
4. Under Properties select your time zone.

5. Select Add NTP Server.

6. Enter the IP address/DNS name of the NTP server.

7. Select OK.

Figure 12 Adding a NTP Server - Summary



Time Zone : America/Los_Angeles (Pacif

NTP Server all.net.esl.cisco.com

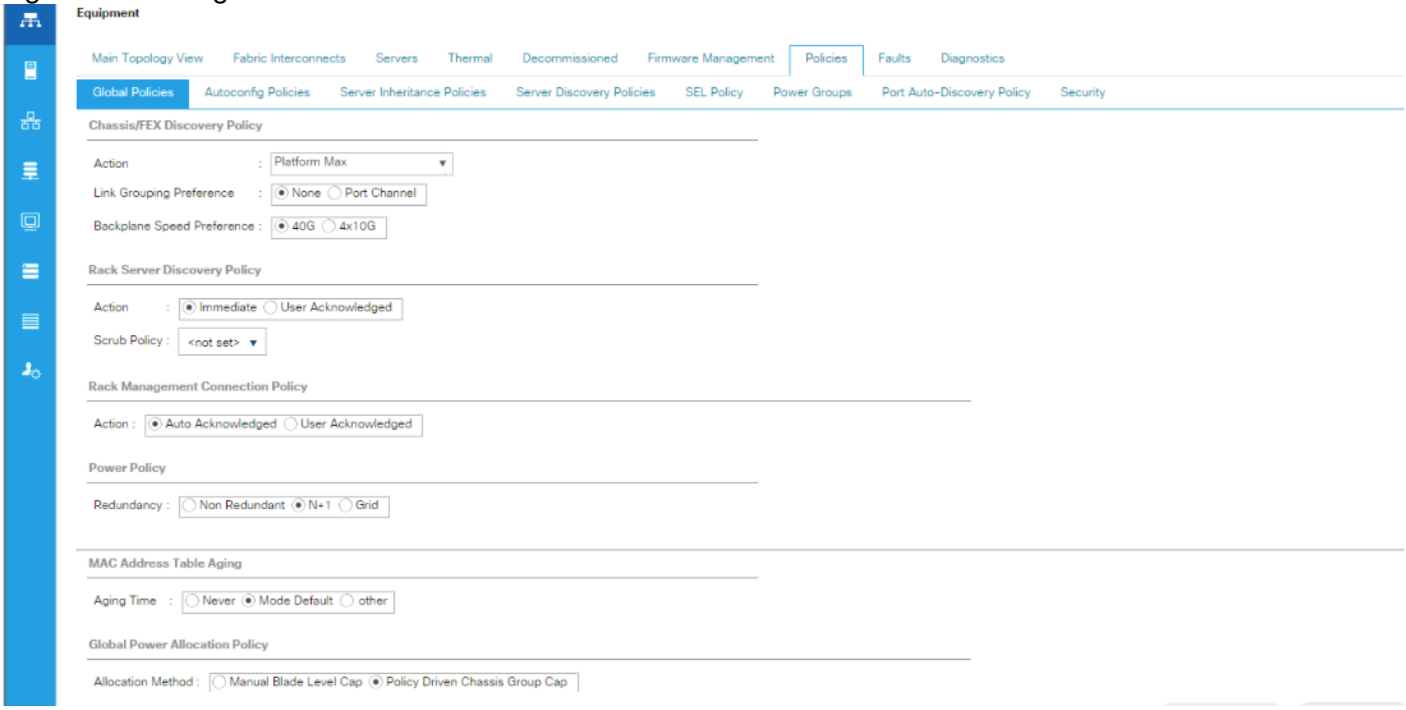
Initial Base Setup of the Environment

Configure Global Policies

To configure the global policies, follow these steps:

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

Figure 13 Configuration of Global Policies



Chassis/FEX Discovery Policy

Action : Platform Max

Link Grouping Preference : ● None ○ Port Channel

Backplane Speed Preference : ● 40G ○ 4x10G

Rack Server Discovery Policy

Action : ● Immediate ○ User Acknowledged

Scrub Policy : <not set>

Rack Management Connection Policy

Action : ● Auto Acknowledged ○ User Acknowledged

Power Policy

Redundancy : ○ Non Redundant ● N+1 ○ Grid

MAC Address Table Aging

Aging Time : ○ Never ● Mode Default ○ other

Global Power Allocation Policy

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

Enable Fabric Interconnect Server Ports

To enable server ports, follow these steps:

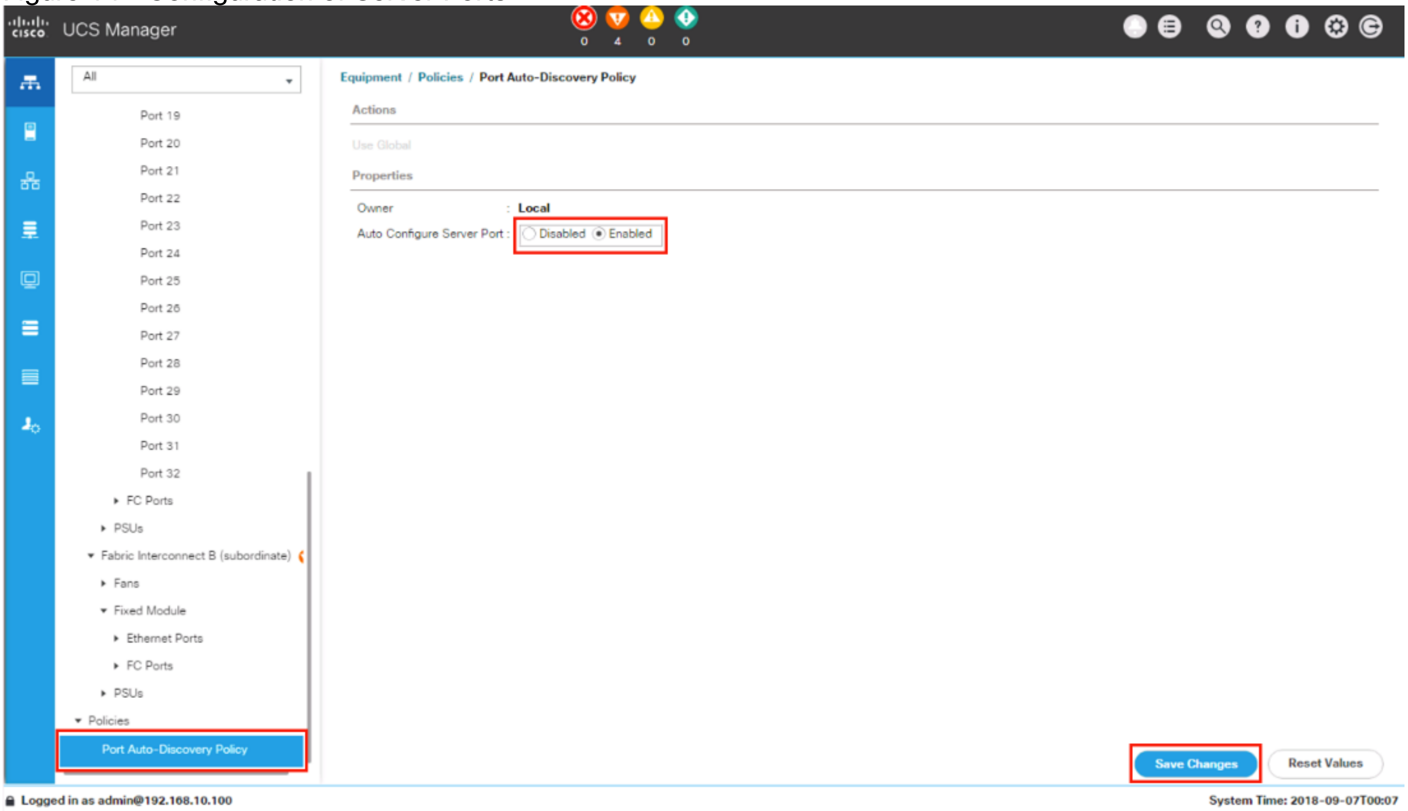
1. Select the **`Equipment`** tab on the left site.

2. Select Equipment > Policies > Port-Auto Discovery Policy.

3. Click **`Enabled`** Under Properties

4. Click **`Save Changes`** to Configure Server Ports Automatically for FI-A and FI-B.

Figure 14 Configuration of Server Ports



Owner : Local

Auto Configure Server Port : ○ Disabled ● Enabled

Logged in as admin@192.168.10.100

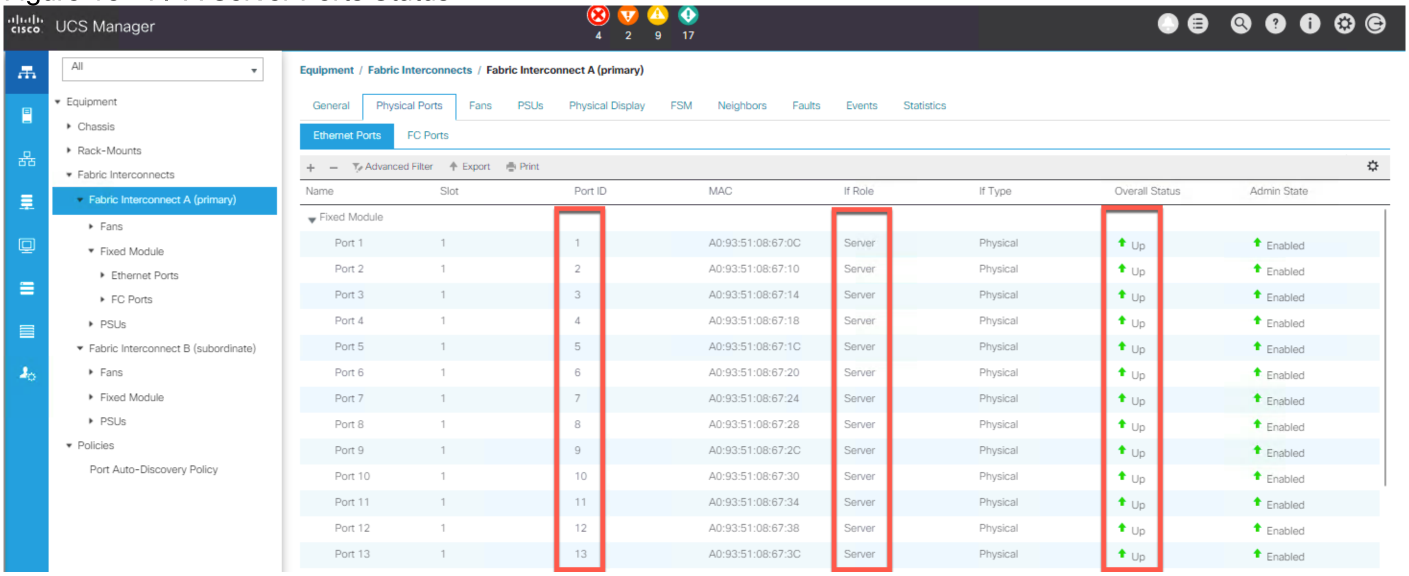
System Time: 2018-09-07T00:07

5. Verify the ports Server port on Fabric Interconnect A

6. Select Equipment > Fabric Interconnects > Fabric Interconnect A (primary) > Fixed Module.

7. Click **`Ethernet Ports`** section.

Figure 15 FI-A Server Ports Status



Equipment / Fabric Interconnects / Fabric Interconnect A (primary)

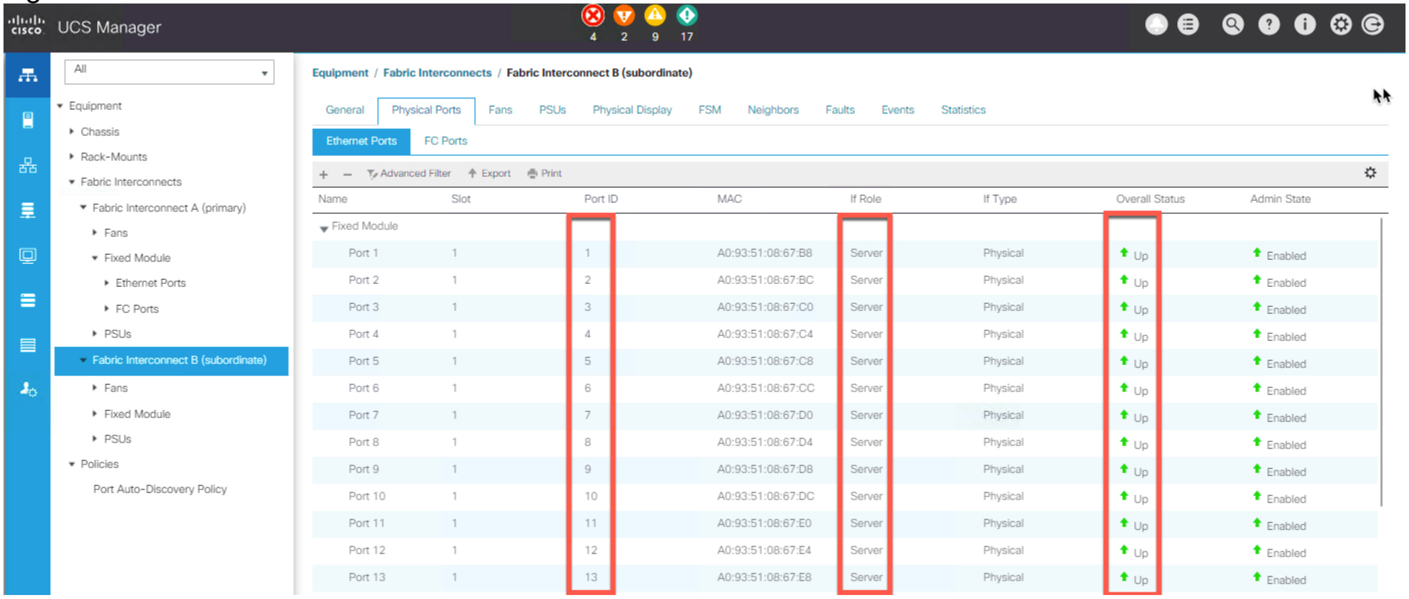
Name	Slot	Port ID	MAC	If Role	If Type	Overall Status	Admin State
Fixed Module							
Port 1	1	1	A0:93:51:08:67:0C	Server	Physical	Up	Enabled
Port 2	1	2	A0:93:51:08:67:10	Server	Physical	Up	Enabled
Port 3	1	3	A0:93:51:08:67:14	Server	Physical	Up	Enabled
Port 4	1	4	A0:93:51:08:67:18	Server	Physical	Up	Enabled
Port 5	1	5	A0:93:51:08:67:1C	Server	Physical	Up	Enabled
Port 6	1	6	A0:93:51:08:67:20	Server	Physical	Up	Enabled
Port 7	1	7	A0:93:51:08:67:24	Server	Physical	Up	Enabled
Port 8	1	8	A0:93:51:08:67:28	Server	Physical	Up	Enabled
Port 9	1	9	A0:93:51:08:67:2C	Server	Physical	Up	Enabled
Port 10	1	10	A0:93:51:08:67:30	Server	Physical	Up	Enabled
Port 11	1	11	A0:93:51:08:67:34	Server	Physical	Up	Enabled
Port 12	1	12	A0:93:51:08:67:38	Server	Physical	Up	Enabled
Port 13	1	13	A0:93:51:08:67:3C	Server	Physical	Up	Enabled

8. Verify the ports Server port on Fabric Interconnect A

9. Select Equipment > Fabric Interconnects > Fabric Interconnect B (subordinate) > Fixed Module.

10. Click **Ethernet Ports** section.

Figure 16 FI-B Server Ports Status



Equipment / Fabric Interconnects / Fabric Interconnect B (subordinate)

Name	Slot	Port ID	MAC	If Role	If Type	Overall Status	Admin State
Fixed Module							
Port 1	1	1	A0:93:51:08:67:B8	Server	Physical	Up	Enabled
Port 2	1	2	A0:93:51:08:67:BC	Server	Physical	Up	Enabled
Port 3	1	3	A0:93:51:08:67:C0	Server	Physical	Up	Enabled
Port 4	1	4	A0:93:51:08:67:C4	Server	Physical	Up	Enabled
Port 5	1	5	A0:93:51:08:67:C8	Server	Physical	Up	Enabled
Port 6	1	6	A0:93:51:08:67:CC	Server	Physical	Up	Enabled
Port 7	1	7	A0:93:51:08:67:D0	Server	Physical	Up	Enabled
Port 8	1	8	A0:93:51:08:67:D4	Server	Physical	Up	Enabled
Port 9	1	9	A0:93:51:08:67:D8	Server	Physical	Up	Enabled
Port 10	1	10	A0:93:51:08:67:DC	Server	Physical	Up	Enabled
Port 11	1	11	A0:93:51:08:67:E0	Server	Physical	Up	Enabled
Port 12	1	12	A0:93:51:08:67:E4	Server	Physical	Up	Enabled
Port 13	1	13	A0:93:51:08:67:E8	Server	Physical	Up	Enabled

Enable Fabric Interconnect A Ports for Uplinks

To enable uplink ports, follow these steps:

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

2. Select Equipment > Fabric Interconnects > Fabric Interconnect A (primary) > Fixed Module.

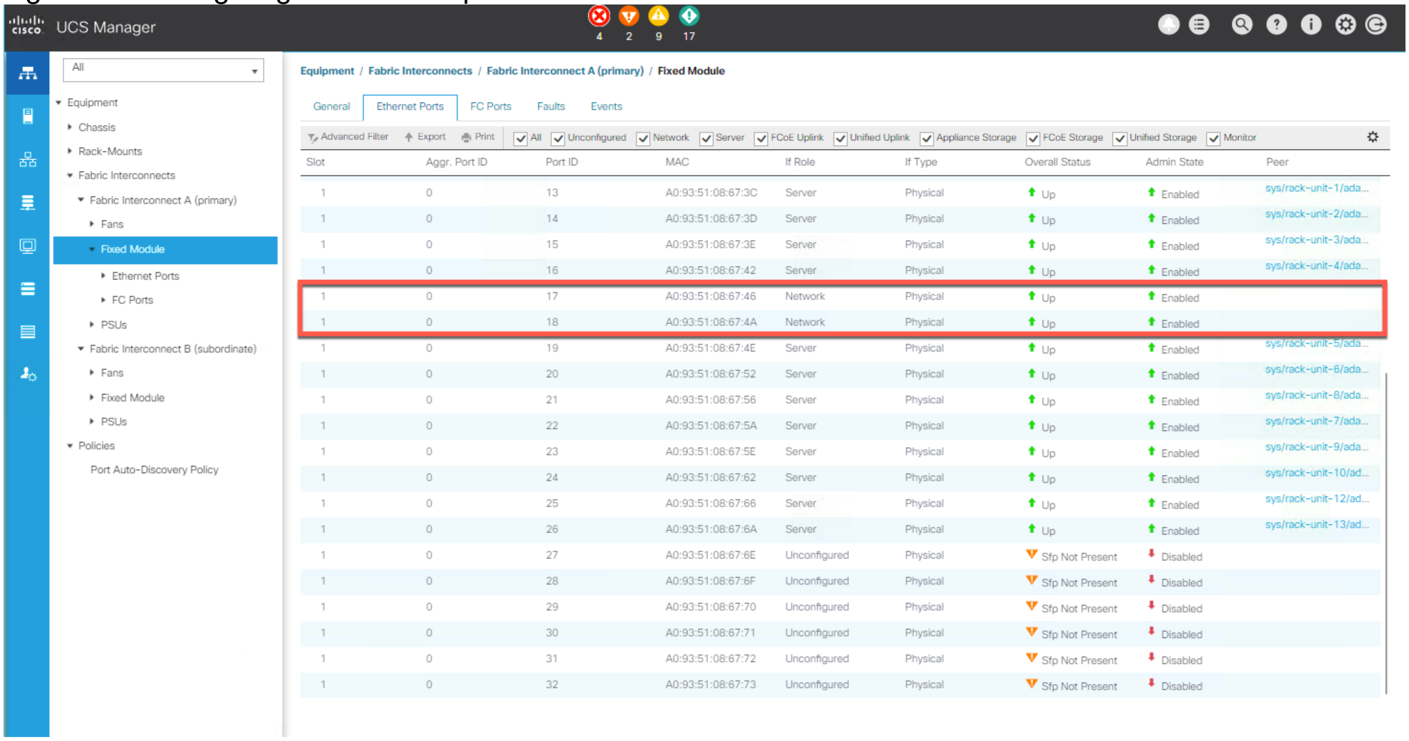
3. Click **Ethernet Ports** section.

4. Select Ports 17-18, right-click and then select **Configure as Uplink Port**.

5. Click **Yes** and then **OK**.

6. Repeat steps 1-5 for Fabric Interconnect B.

Figure 17 Configuring of Network Uplink Ports



Equipment / Fabric Interconnects / Fabric Interconnect A (primary) / Fixed Module

☑ All ☑ Unconfigured ☑ Network ☑ Server ☑ FCoE Uplink ☑ Unified Uplink ☑ Appliance Storage ☑ FCoE Storage ☑ Unified Storage ☑ Monitor

Slot	Aggr. Port ID	Port ID	MAC	If Role	If Type	Overall Status	Admin State	Peer
1	0	13	A0:93:51:08:67:3C	Server	Physical	Up	Enabled	sys/rack-unit-1/ada...
1	0	14	A0:93:51:08:67:3D	Server	Physical	Up	Enabled	sys/rack-unit-2/ada...
1	0	15	A0:93:51:08:67:3E	Server	Physical	Up	Enabled	sys/rack-unit-3/ada...
1	0	16	A0:93:51:08:67:42	Server	Physical	Up	Enabled	sys/rack-unit-4/ada...
1	0	17	A0:93:51:08:67:46	Network	Physical	Up	Enabled	
1	0	18	A0:93:51:08:67:4A	Network	Physical	Up	Enabled	
1	0	19	A0:93:51:08:67:4E	Server	Physical	Up	Enabled	sys/rack-unit-5/ada...
1	0	20	A0:93:51:08:67:52	Server	Physical	Up	Enabled	sys/rack-unit-6/ada...
1	0	21	A0:93:51:08:67:56	Server	Physical	Up	Enabled	sys/rack-unit-8/ada...
1	0	22	A0:93:51:08:67:5A	Server	Physical	Up	Enabled	sys/rack-unit-7/ada...
1	0	23	A0:93:51:08:67:5E	Server	Physical	Up	Enabled	sys/rack-unit-9/ada...
1	0	24	A0:93:51:08:67:62	Server	Physical	Up	Enabled	sys/rack-unit-10/ad...
1	0	25	A0:93:51:08:67:66	Server	Physical	Up	Enabled	sys/rack-unit-12/ad...
1	0	26	A0:93:51:08:67:6A	Server	Physical	Up	Enabled	sys/rack-unit-13/ad...
1	0	27	A0:93:51:08:67:6E	Unconfigured	Physical	Sfp Not Present	Disabled	
1	0	28	A0:93:51:08:67:6F	Unconfigured	Physical	Sfp Not Present	Disabled	
1	0	29	A0:93:51:08:67:70	Unconfigured	Physical	Sfp Not Present	Disabled	
1	0	30	A0:93:51:08:67:71	Unconfigured	Physical	Sfp Not Present	Disabled	
1	0	31	A0:93:51:08:67:72	Unconfigured	Physical	Sfp Not Present	Disabled	
1	0	32	A0:93:51:08:67:73	Unconfigured	Physical	Sfp Not Present	Disabled	

Label Servers for Identification

For better identification, label each server by following these 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.

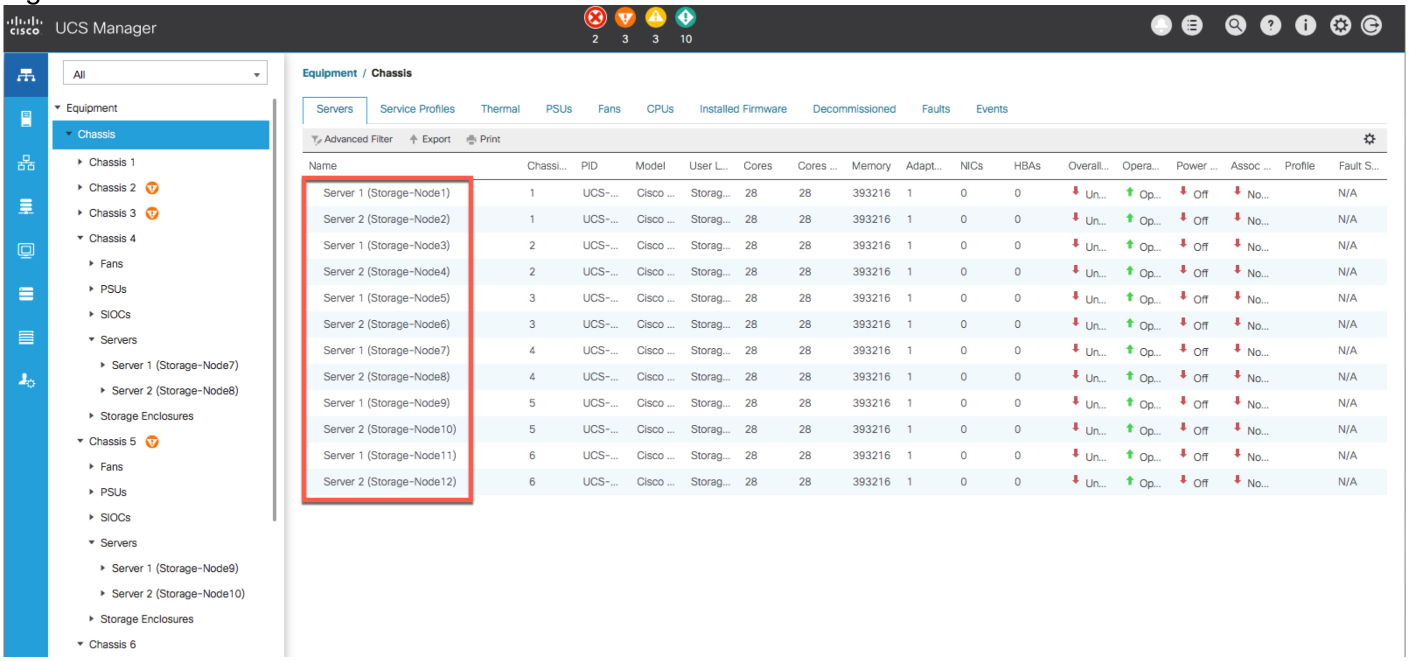
4. Repeat the previous steps for **Server 2** of **Chassis 1** and for all other servers of Chassis 2 – 6 according to Table 5 .

Table 5 Server Label

Server	Name
Chassis 1 / Server 1	Storage-Node1
Chassis 1 / Server 2	Storage-Node2
Chassis 2/ Server 1	Storage-Node3

Server	Name
Chassis 2 / Server 2	Storage-Node4
Chassis 3 / Server 1	Storage-Node5
Chassis 3 / Server 2	Storage-Node6
Chassis 4 / Server 1	Storage-Node7
Chassis 4 / Server 2	Storage-Node8
Chassis 5/ Server 1	Storage-Node9
Chassis 5 / Server 2	Storage-Node10
Chassis 6 / Server 1	Storage-Node11
Chassis 6 / Server 2	Storage-Node12

Figure 18   Cisco UCS Server Labels



Name	Chassi...	PID	Model	User L...	Cores	Cores ...	Memory	Adapt...	NICs	HBAs	Overall...	Opera...	Power ...	Assoc ...	Profile	Fault S...
Server 1 (Storage-Node1)	1	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 2 (Storage-Node2)	1	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 1 (Storage-Node3)	2	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 2 (Storage-Node4)	2	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 1 (Storage-Node5)	3	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 2 (Storage-Node6)	3	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 1 (Storage-Node7)	4	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 2 (Storage-Node8)	4	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 1 (Storage-Node9)	5	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 2 (Storage-Node10)	5	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 1 (Storage-Node11)	6	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A
Server 2 (Storage-Node12)	6	UCS-...	Cisco ...	Storag...	28	28	393216	1	0	0	↓ Un...	↑ Op...	↓ Off	↓ No...		N/A

Create KVM IP Pool

To create a KVM IP Pool, follow these steps:

1. Select the **LAN** tab on the left site.
2. Go to LAN > Pools > root > IP Pools > IP Pool ext-mgmt.
3. Click on Create Block of IPv4 Addresses.
4. Enter an IP Address in the **From** field.
5. Enter **Size** 50.

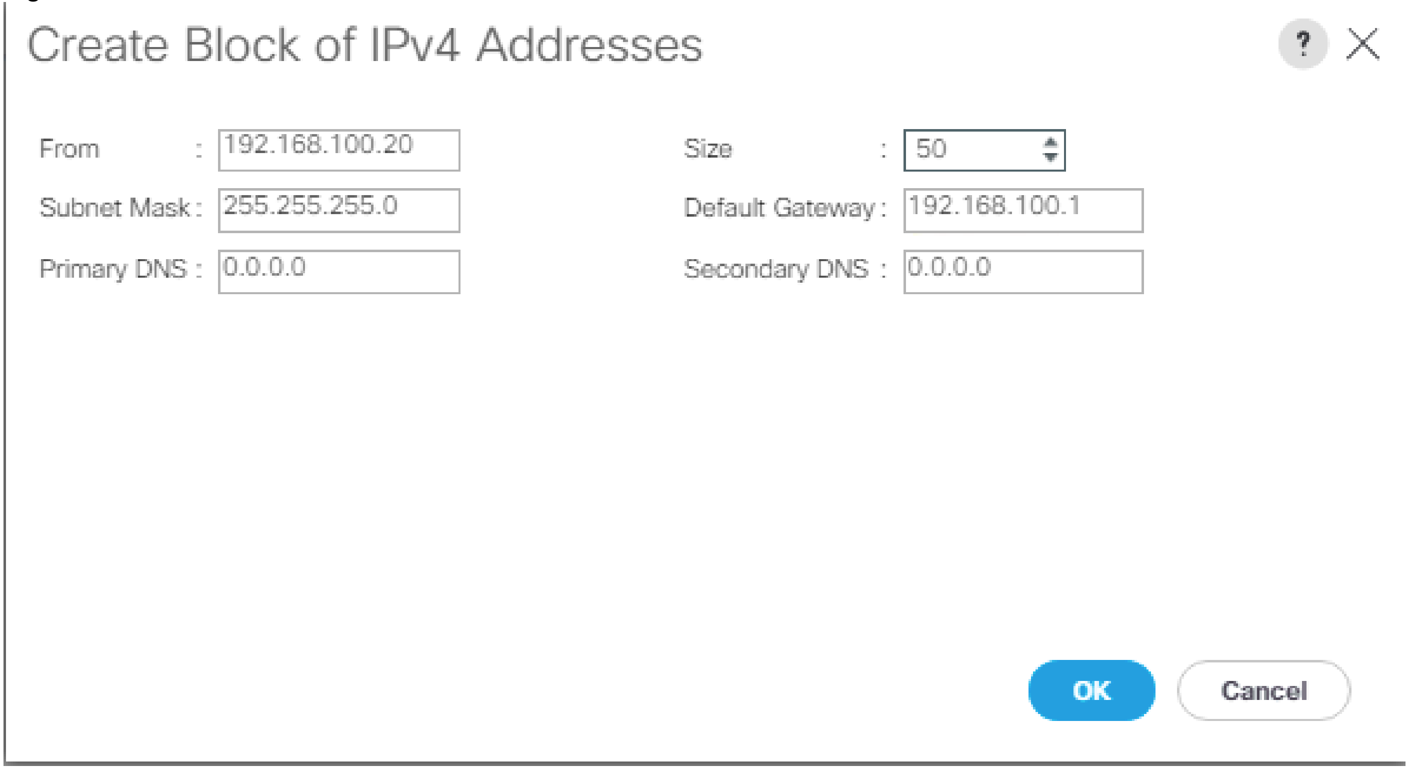
6. Enter your `Subnet Mask`.

7. Fill in your `Default Gateway`.

8. Enter your **`Primary DNS`** and **`Secondary DNS`** if needed.

9. Click `OK`.

Figure 19 Create Block of IPv4 Addresses



Create Block of IPv4 Addresses

From :	192.168.100.20	Size :	50
Subnet Mask :	255.255.255.0	Default Gateway :	192.168.100.1
Primary DNS :	0.0.0.0	Secondary DNS :	0.0.0.0

Create MAC Pool

To create a MAC Pool, follow these steps:

1. Select the LAN tab.

2. Go to LAN > Pools > root > Mac Pools and right-click Create MAC Pool.

3. Type in “SwiftStack-MAC-Pools” for Name.

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

5. Set Assignment Order as Sequential.

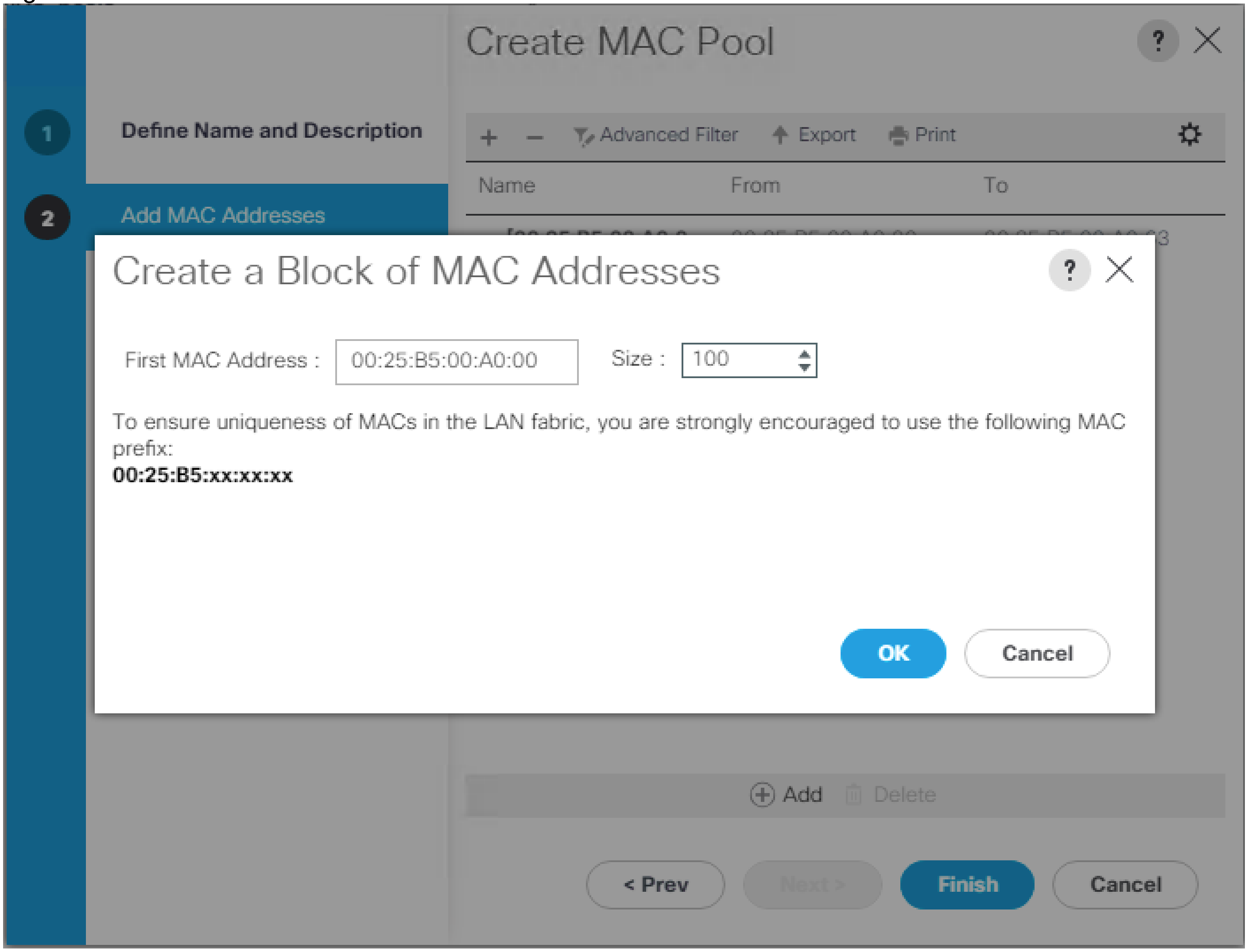
6. Click **`Next`**.

7. Click **`Add`**.

8. Specify a starting MAC address.

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

Figure 20 Create a Block of MAC Addresses



Create a Block of MAC Addresses

First MAC Address : 00:25:B5:00:A0:00

Size : 100

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

10. Click **OK**.

11. Click **Finish**.

Create UUID Pool

To create a UUID Pool, follow these 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 “SwiftStack-UUID-Pools” for Name.

4. (Optional) Enter a **Description** of the MAC Pool.

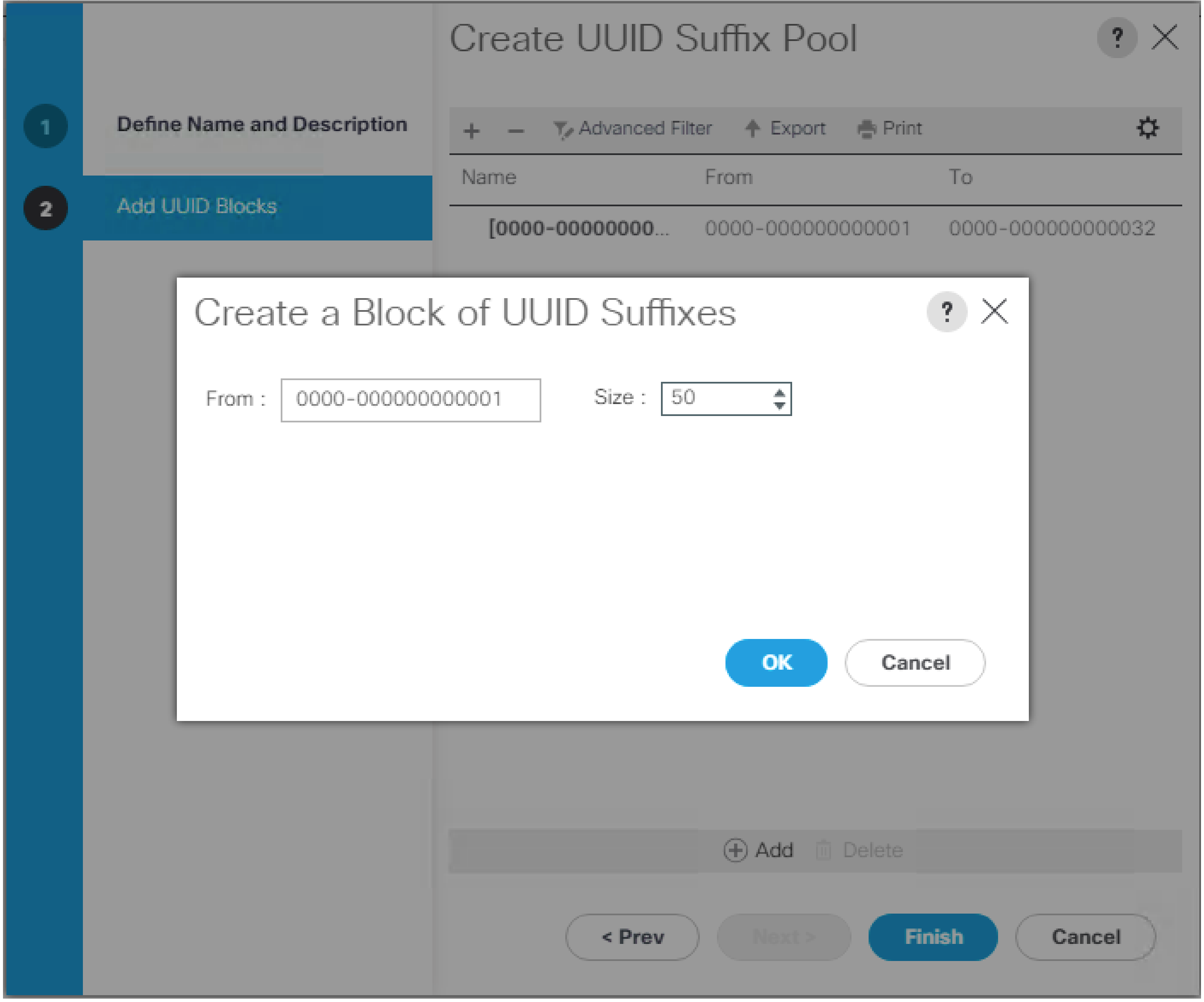
5. Set Assignment Order to Sequential and click Next.

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, 50.

Figure 21 Create a Block of UUID Suffixes



Create UUID Suffix Pool

1 Define Name and Description

2 Add UUID Blocks

Name	From	To
[0000-00000000...	0000-000000000001	0000-000000000032

Create a Block of UUID Suffixes

From : 0000-000000000001

Size : 50

9. Click **OK**.

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

Create VLANs

It is important to separate the network traffic with VLANs for Storage-Management traffic and Storage-Cluster traffic, External traffic, Replication and Client traffic (optional). Table 6 lists the configured VLANs.

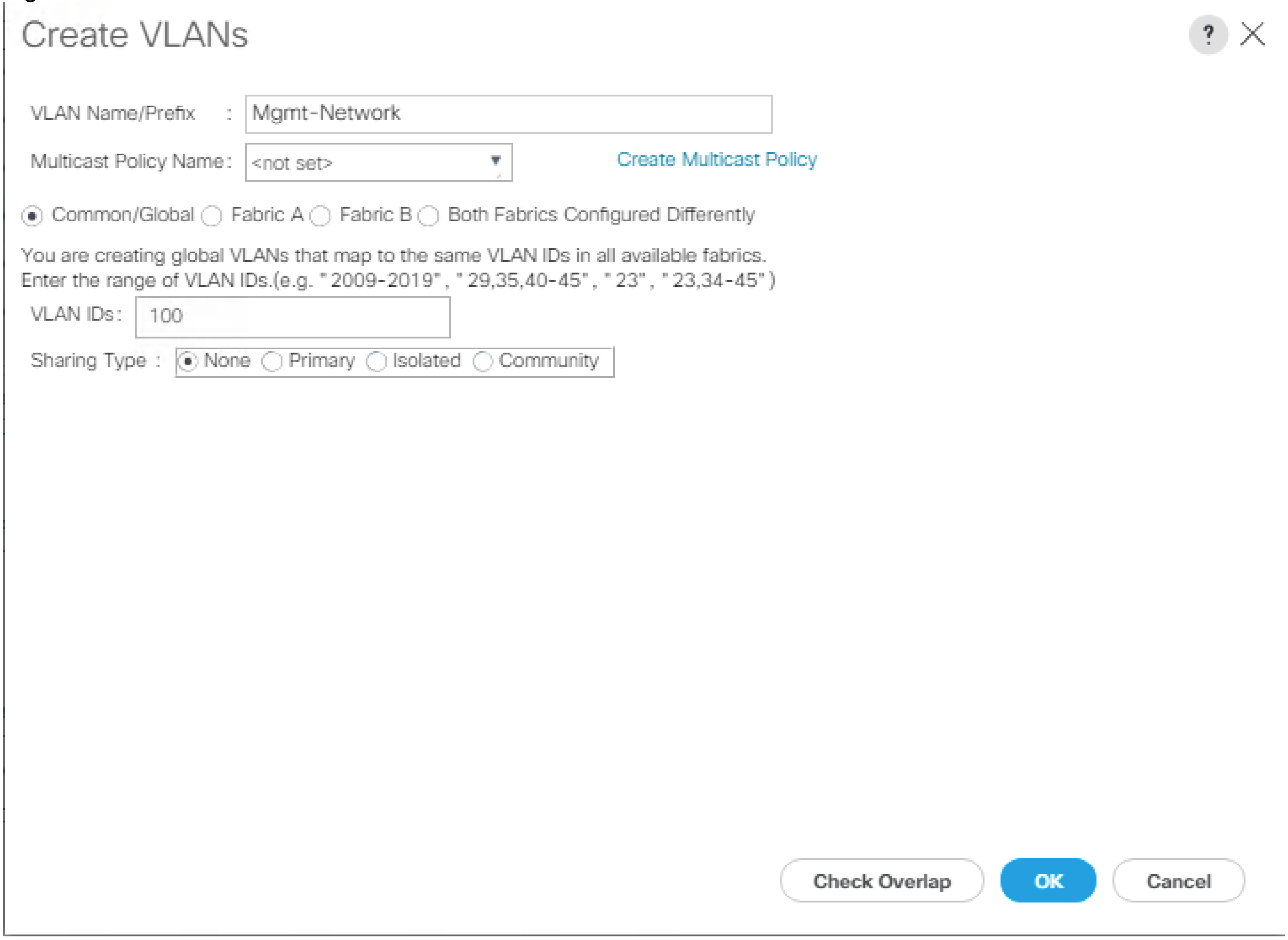
Table 6 VLAN Configurations

VLAN	Name	Function
100	Management-Network	Optional if UCSM is not in External VLAN.
120	Swift-Client	Outward Client Facing Network.
130	Swift-Cluster	Cluster Network
150	Swift-Replication	Replication Network
219	External-Network	External Network for OS to download and yum updates

To configure VLANs in the Cisco UCS Manager GUI, follow these steps:

1. Select **LAN** in the left pane in the UCSM GUI.
2. Select LAN > LAN Cloud > VLANs and right-click Create VLANs.
3. Enter “Storage-Mgmt” for the VLAN Name.
4. Keep Multicast Policy Name as <not set>.
5. Select **Common/Global** for Public.
6. Enter 100 in the **VLAN IDs** field.
7. Click **OK** and then click **Finish**.

Figure 22 Create a VLAN



Create VLANs

VLAN Name/Prefix : Mgmt-Network

Multicast Policy Name: <not set>

● Common/Global ○ Fabric A ○ Fabric B ○ Both Fabrics Configured Differently

You are creating global VLANs that map to the same VLAN IDs in all available fabrics.
Enter the range of VLAN IDs.(e.g. " 2009-2019" , " 29,35,40-45" , " 23" , " 23,34-45")

VLAN IDs: 100

Sharing Type : ● None ○ Primary ○ Isolated ○ Community

8. Repeat steps 1-7 for the VLANs Swift-Cluster, Replication, External and Client-Network.

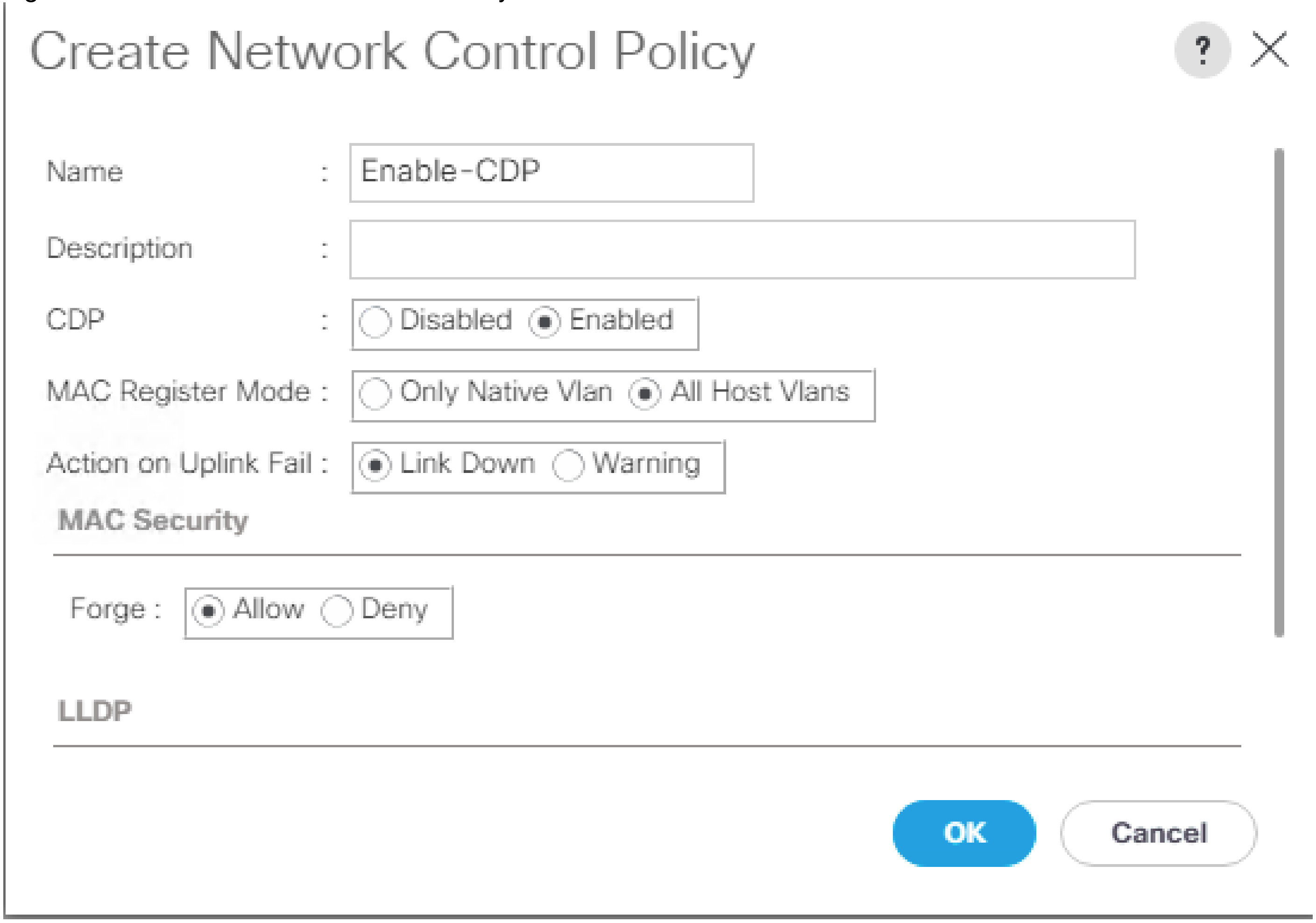
Enable CDP

To enable Network Control Policies, follow these 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 23 Create a Network Control Policy



Create Network Control Policy

Name : Enable-CDP

Description :

CDP : ○ Disabled ● Enabled

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

Action on Uplink Fail : ● Link Down ○ Warning

MAC Security

Forge : ● Allow ○ Deny

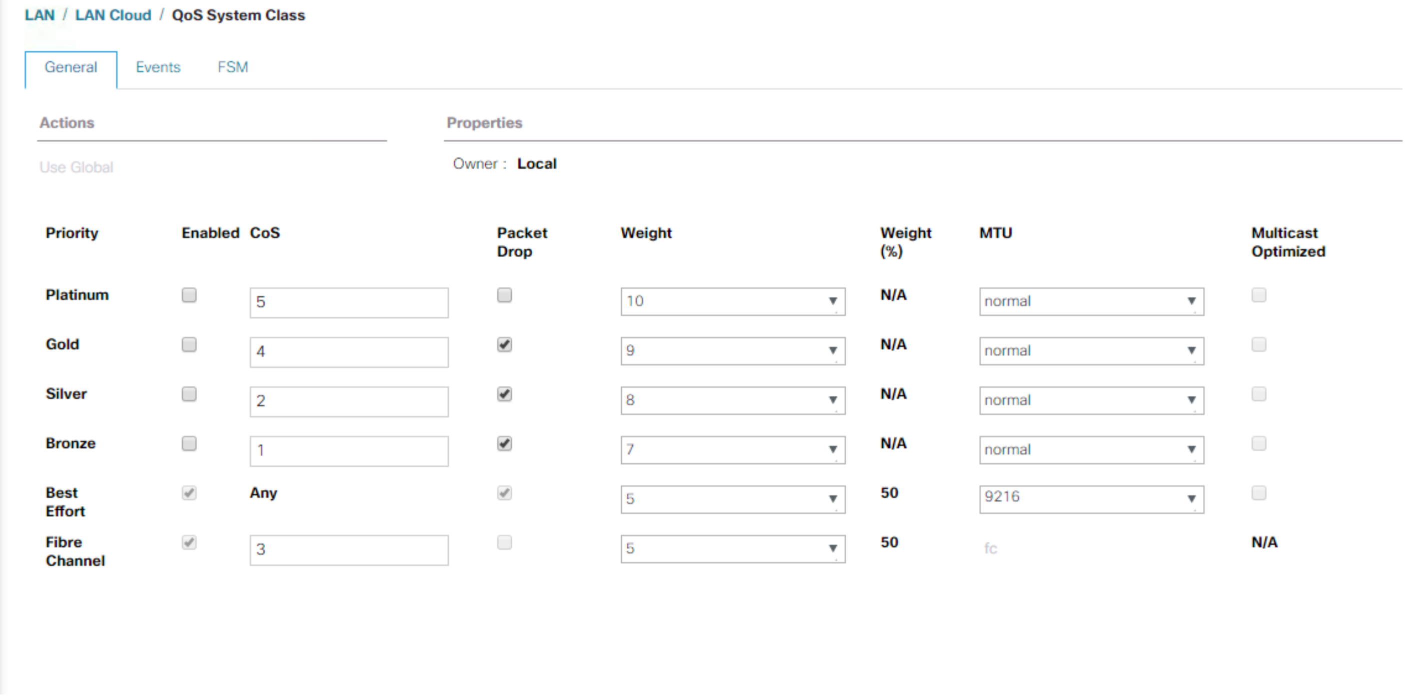
LLDP

QoS System Class

To create a Quality of Service System Class, follow these 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. Best Effort MTU as 9216.
4. Set Fibre Channel Weight to None.
5. Click **Save Changes** and then click **OK**.

Figure 24 QoS System Class



Owner : **Local**

Priority	Enabled	CoS	Packet Drop	Weight	Weight (%)	MTU	Multicast Optimized
Platinum	□	5	□	10	N/A	normal	□
Gold	□	4	☑	9	N/A	normal	□
Silver	□	2	☑	8	N/A	normal	□
Bronze	□	1	☑	7	N/A	normal	□
Best Effort	☑	Any	☑	5	50	9216	□
Fibre Channel	☑	3	□	5	50	fc	N/A

vNIC Template Setup

Based on the previous section of creating VLANs, the next step is to create the appropriate vNIC templates. For SwiftStack Storage we need to create four different vNICs, depending on the role of the server. Table 7 provides an overview of the configuration.

Table 7 vNIC Table

Name	vNIC Name	Fabric Interconnect	Failover	VLAN	MTU Size
Outward Facing	Swift-Client	A	Yes	Swift-Client	9000
Cluster Network	Swift-Cluster	B	Yes	Swift-Cluster	9000
Replication Network	Swift-Repl	B	Yes	Swift-Repl	9000
Mgmt-Network	Swift-Mgmt	A	Yes	Swift-Mgmt	9000
External Network	External	A	Yes	External	1500

To create the appropriate vNICs, follow these 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 **`Swift-client`** in the **`Name`** field.

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

5. Click Fabric A as Fabric ID and enable failover.

6. Template Type as **`Updating Template`**

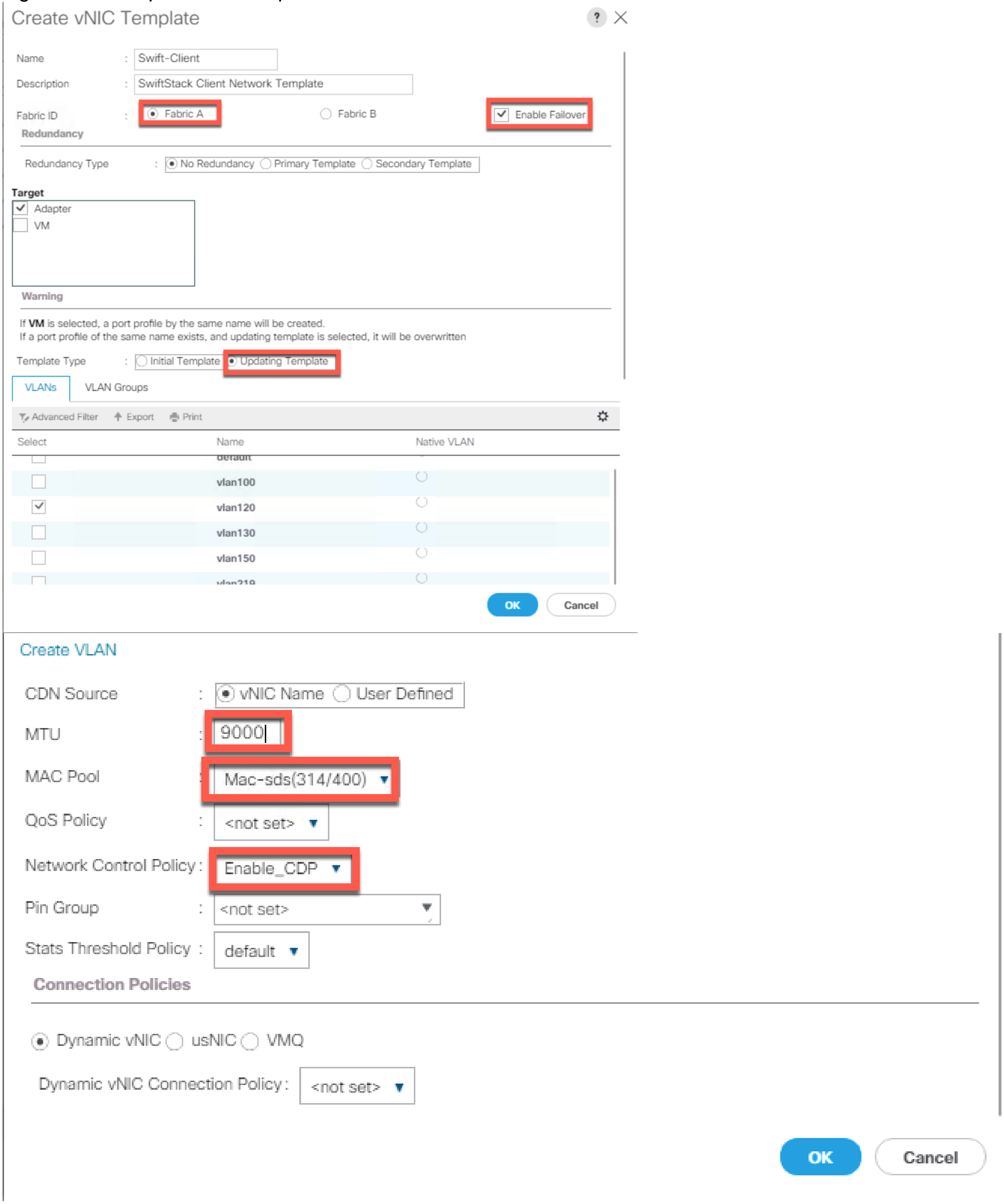
7. Select **`default`** as **`VLANs`** and click **`Native VLAN`**.

8. Select `SwiftStack-MAC-Pools` as MAC Pool.

9. Select Enable-CDP as Network Control Policy.

10. Click **`OK`** and then click **`OK`** again.

Figure 25 Setup of vNIC Template for Swift-Client vNIC



Create vNIC Template

Name : Swift-Client

Description : SwiftStack Client Network Template

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

Redundancy

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

Target

☑ Adapter

□ VM

Warning

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

Template Type : ○ Initial Template ● Updating Template

Select	Name	Native VLAN
□	default	
□	vlan100	○
☑	vlan120	○
□	vlan130	○
□	vlan150	○
□	vlan219	○

Create VLAN

CDN Source : ● vNIC Name ○ User Defined

MTU : 9000

MAC Pool : Mac-sds(314/400)

QoS Policy : <not set>

Network Control Policy : Enable_CDP

Pin Group : <not set>

Stats Threshold Policy : default

Connection Policies

● Dynamic vNIC ○ usNIC ○ VMQ

Dynamic vNIC Connection Policy : <not set>

11. Repeat steps 1-10 for the vNICs “Swift-Rep” “External-Network” “Swift-Mgmt” and “Client-Network”. Make sure you select the correct Fabric ID, VLAN and MTU size according to Table 6 .

Ethernet Adapter Policy Setup

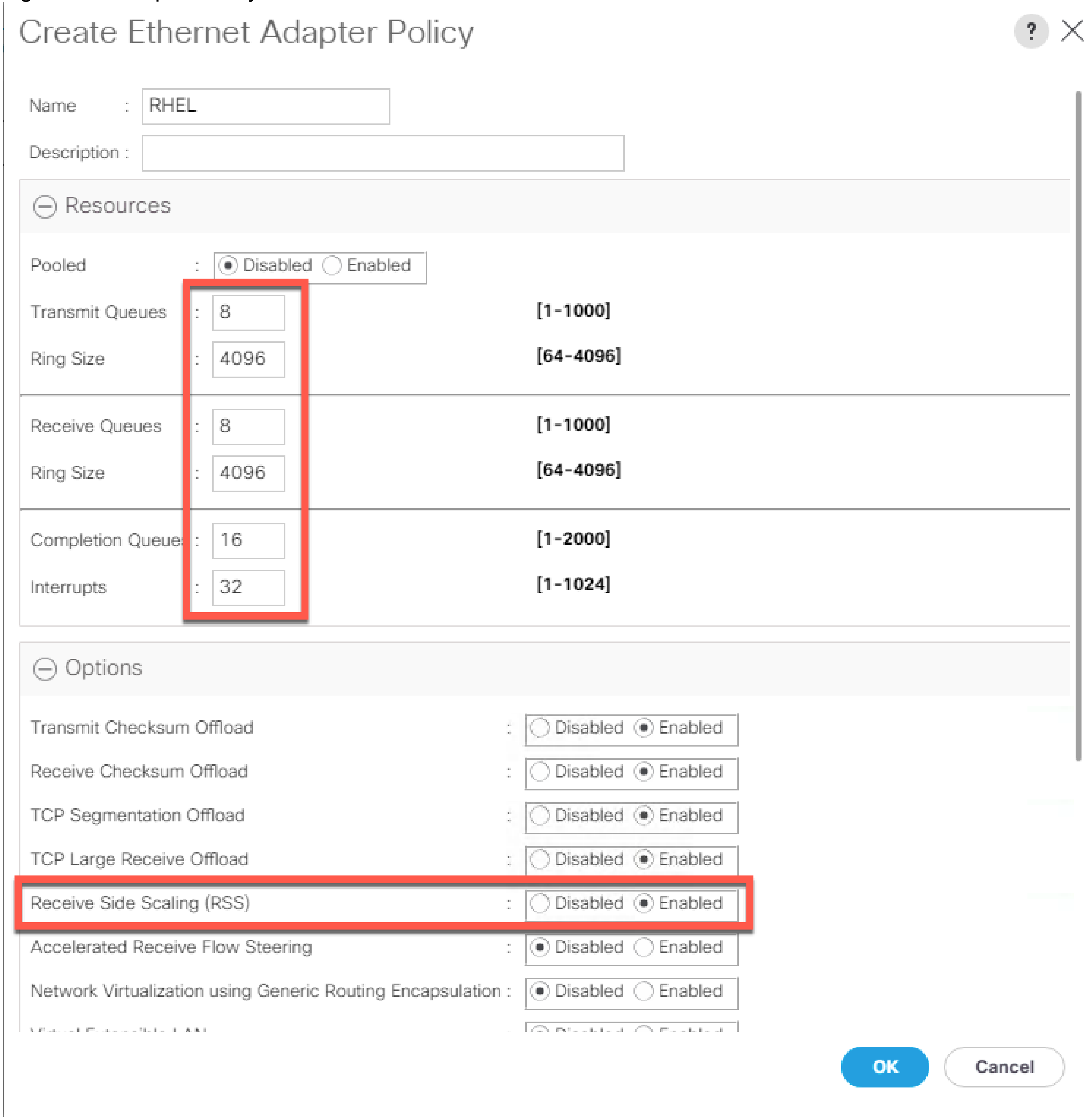
By default, Cisco UCS provides a set of Ethernet adapter policies. These policies include the recommended settings for each supported server operating system. Operating systems are sensitive to the settings in these policies.

Cisco UCS best practice is to enable Jumbo Frames MTU 9000 for any Storage facing Networks (Storage-Mgmt and Storage-Cluster). Enabling jumbo frames on specific interfaces and modifying Tx and Rx values guarantees 39Gb/s bandwidth on the UCS fabric.

To create a specific adapter policy for Red Hat Enterprise Linux, follow these 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:
   - Transmit Queues: 8
   - Ring Size: 4096
   - Receive Queues: 8
   - Ring Size: 4096
   - Completion Queues: 16
   - Interrupts: 32
6. Under Options enable Receive Side Scaling (RSS).
7. Click **OK** and then click **OK** again.

Figure 26 Adapter Policy for RHEL



Create Ethernet Adapter Policy

Name : RHEL

Description :

Resources

Pooled : ● Disabled ○ Enabled

Transmit Queues : 8 [1-1000]

Ring Size : 4096 [64-4096]

Receive Queues : 8 [1-1000]

Ring Size : 4096 [64-4096]

Completion Queues : 16 [1-2000]

Interrupts : 32 [1-1024]

Options

Transmit Checksum Offload : ○ Disabled ● Enabled

Receive Checksum Offload : ○ Disabled ● Enabled

TCP Segmentation Offload : ○ Disabled ● Enabled

TCP Large Receive Offload : ○ Disabled ● Enabled

Receive Side Scaling (RSS) : ○ Disabled ● Enabled

Accelerated Receive Flow Steering : ● Disabled ○ Enabled

Network Virtualization using Generic Routing Encapsulation : ● Disabled ○ Enabled

Virtual Extensible LAN : ○ Disabled ○ Enabled

Boot Policy Setup

To create a Boot Policy, follow these steps:

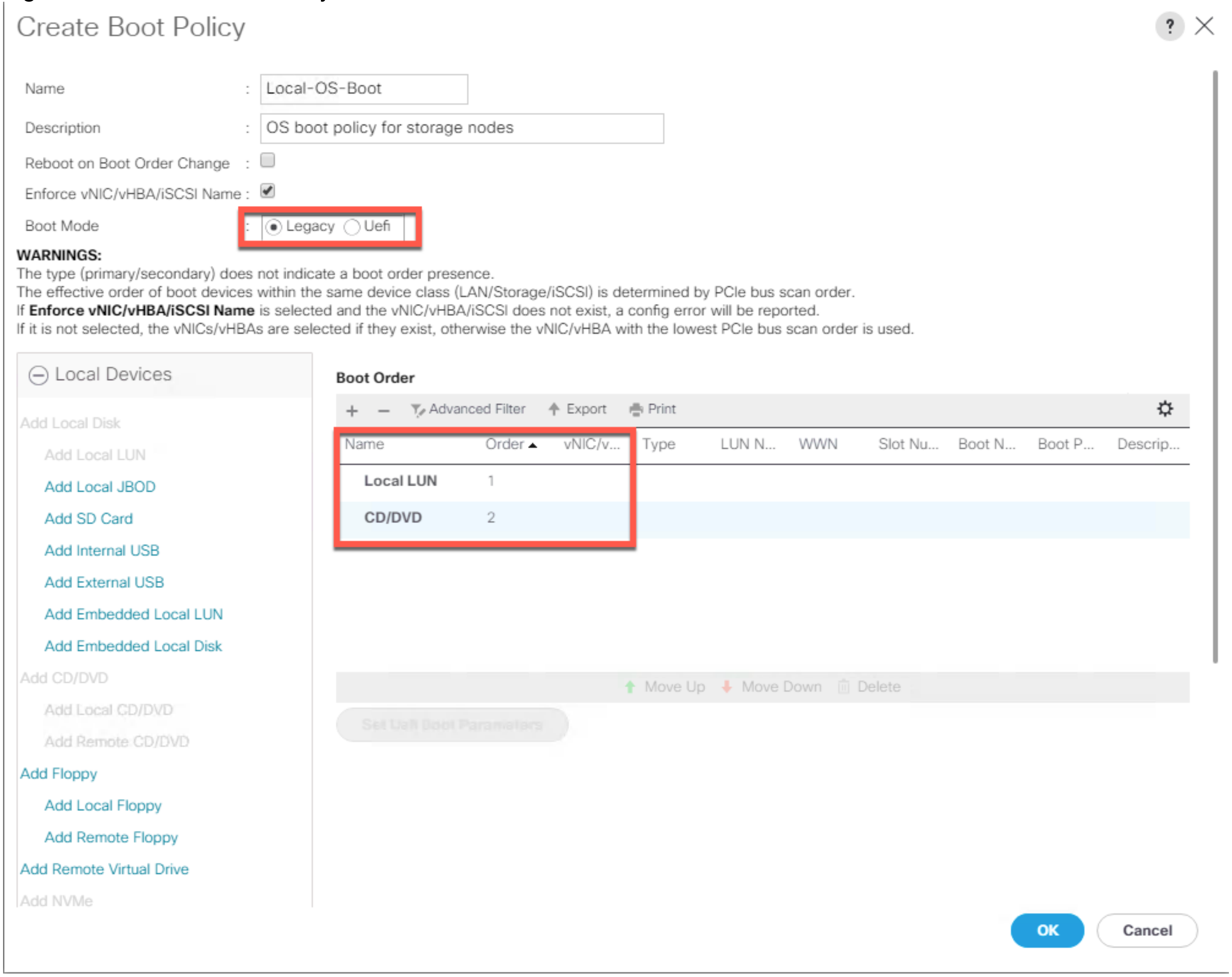
1. Select the **`Servers`** tab in the left pane.

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



Create Boot Policy

Name : Local-OS-Boot

Description : OS boot policy for storage nodes

Reboot on Boot Order Change : □

Enforce vNIC/vHBA/iSCSI Name : ☑

Boot Mode : ● Legacy ○ Uefi

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

Boot Order

Name	Order	vNIC/v...	Type	LUN N...	WWN	Slot Nu...	Boot N...	Boot P...	Descrip...
Local LUN	1								
CD/DVD	2								

5. Click Add CD/DVD and click OK.

6. Click Local Disk > 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, follow these steps:

1. Select the **`LAN`** tab in the left pane.

2. Go to LAN > 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`**

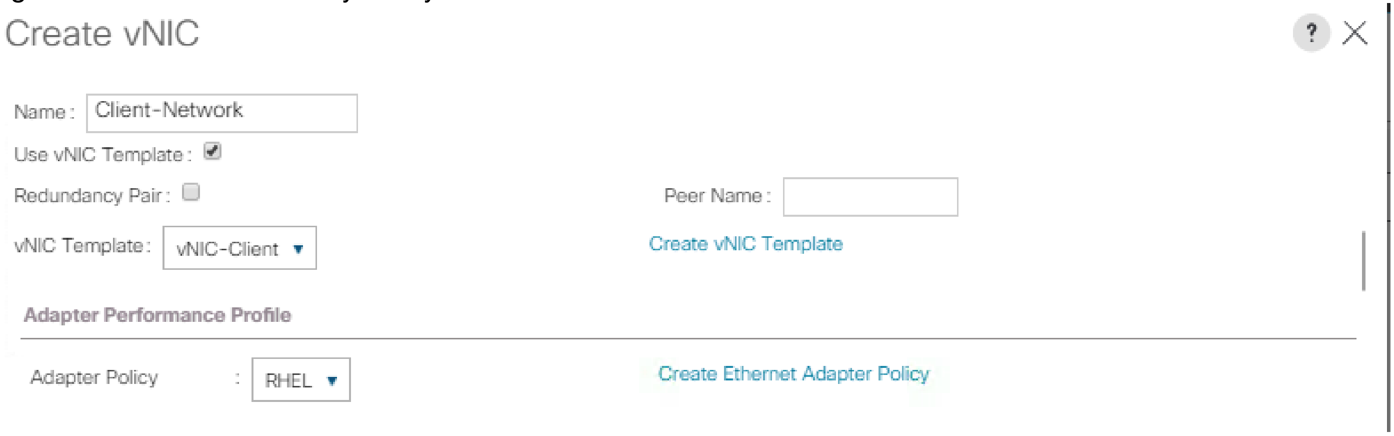
6. Type in Client-Network in the name field.

7. Click "Use vNIC Template."

8. Select vNIC template for "Client-Network" from drop-down list.

9. If you are using Jumbo Frame MTU 9000, select the default Adapter Policy, previously created as "RHEL" from the drop-down list.

Figure 28 LAN Connectivity Policy



Create vNIC

Name : Client-Network

Use vNIC Template : ☑

Redundancy Pair : ☐

Peer Name :

vNIC Template : vNIC-Client

Adapter Performance Profile

Adapter Policy : RHEL

10. Repeat steps 1-9 for the remaining networks "Swift-Cluster", "Swift-Replication", "External-Network", and "Mgmt-Network" Make sure you choose Adapter Policy as "RHEL" for VNIC interface "Client-Nework."

Create Maintenance Policy Setup

To setup a Maintenance Policy, follow these steps:

1. Select the **`Servers`** tab.

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

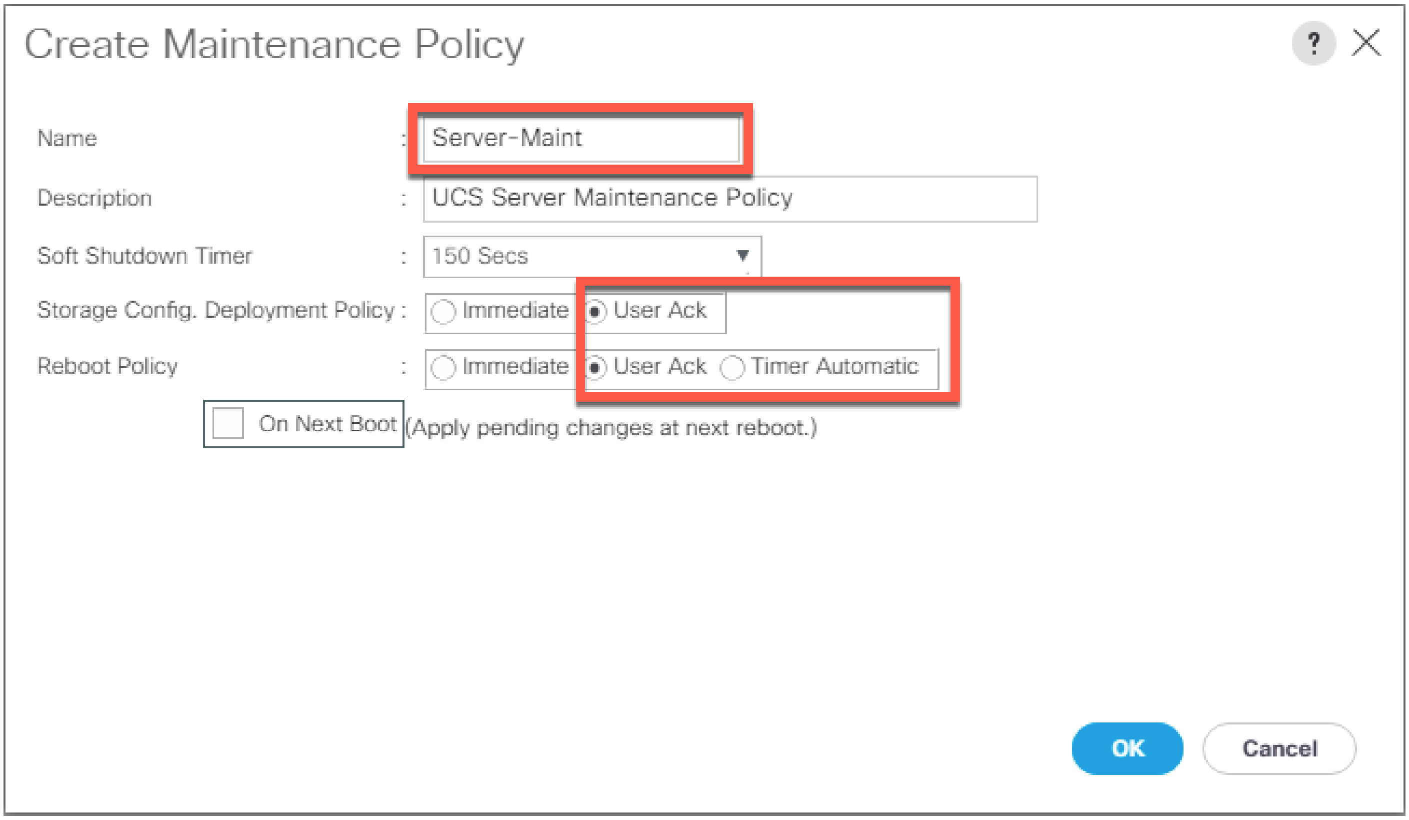
3. Type in a **`Server-Maint`** in the `Name` field.

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

5. Click User Ack under Reboot Policy.

6. Click **`OK`** and then click **`OK`** again.

7. Create Maintenance Policy.



Create Maintenance Policy

Name : Server-Maint

Description : UCS Server Maintenance Policy

Soft Shutdown Timer : 150 Secs

Storage Config. Deployment Policy : ○ Immediate ● User Ack

Reboot Policy : ○ Immediate ● User Ack ○ Timer Automatic

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

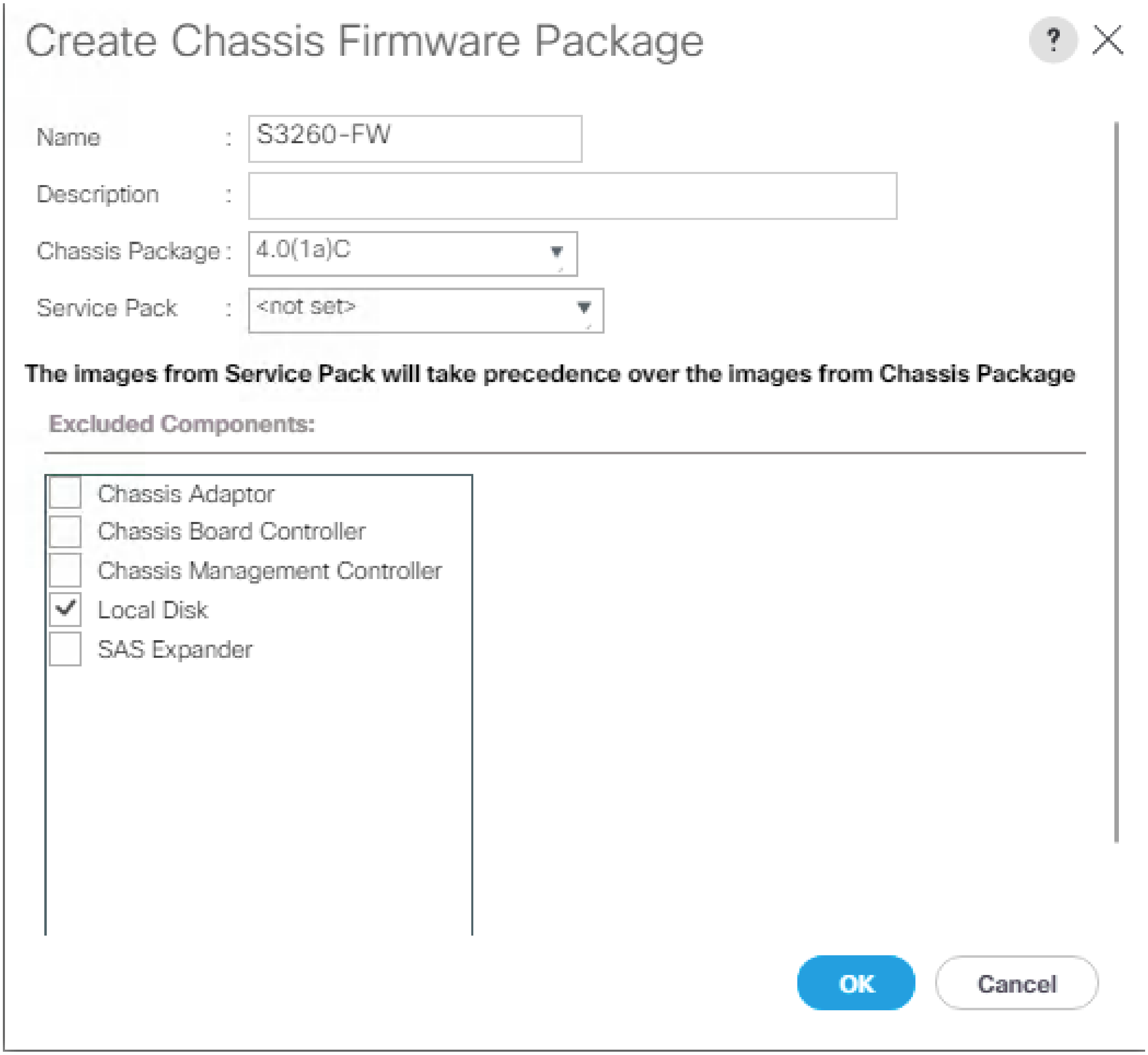
Create Chassis Profile

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, follow these 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 **S3260-FW** in the Name field.
4. (Optional) Enter a description in the **Description** field.
5. Select **4.0(1a)C** form the drop-down list of **Chassis Package**.
6. Select **OK** and then click **OK** again.
7. Create Chassis Firmware Package.



Create Chassis Firmware Package

Name : S3260-FW

Description :

Chassis Package : 4.0(1a)C

Service Pack : <not set>

The images from Service Pack will take precedence over the images from Chassis Package

Excluded Components:

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

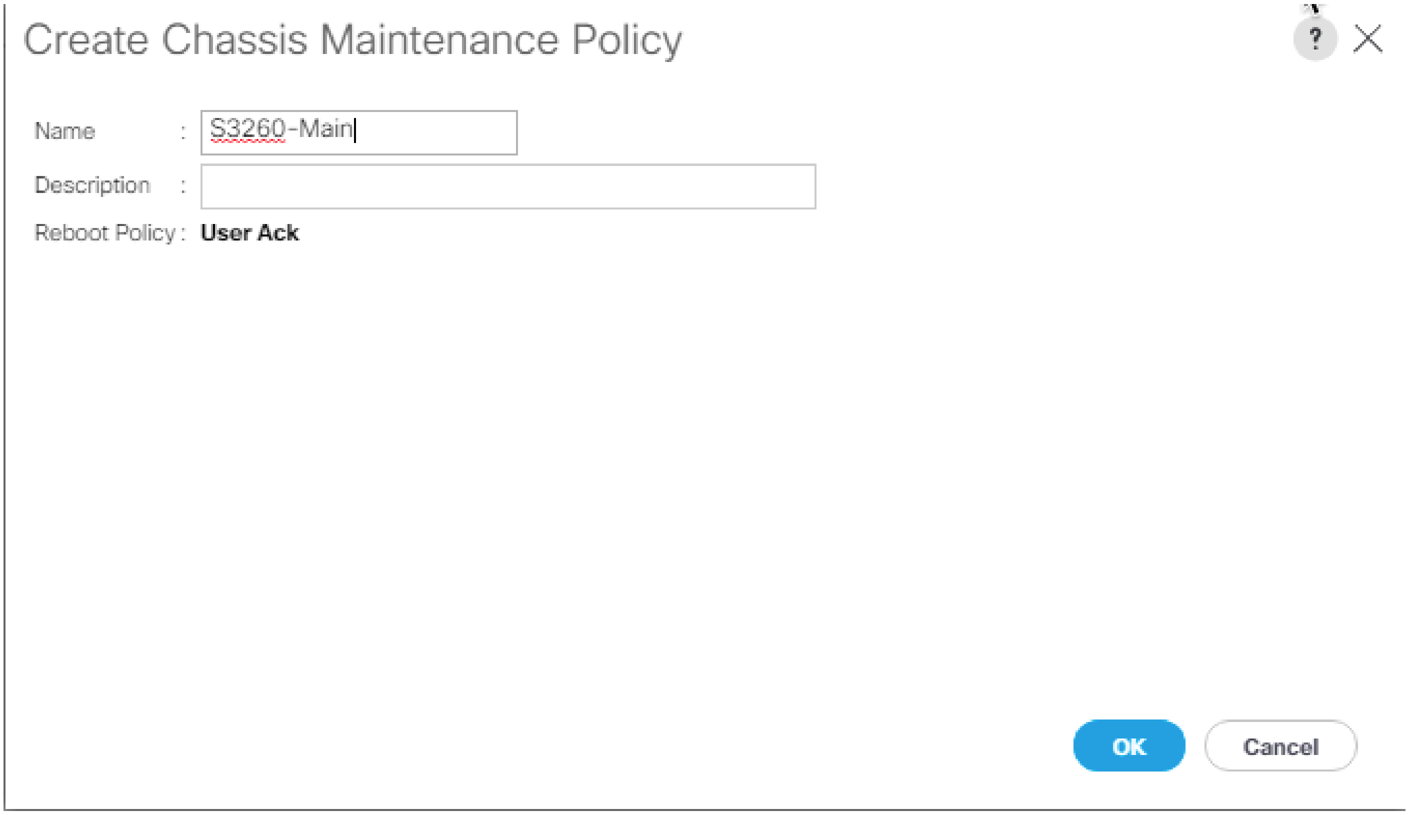
Create Chassis Maintenance Policy

To create a Chassis Maintenance Policy, follow these steps:

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

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

6. Create Chassis Maintenance Policy.



Create Chassis Maintenance Policy

Name : S3260-Main

Description :

Reboot Policy : User Ack

Create Disk Zoning Policy

To create a Disk Zoning Policy, follow these steps:

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

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

3. Type in **S3260-DiskZoning** in the Name field.

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

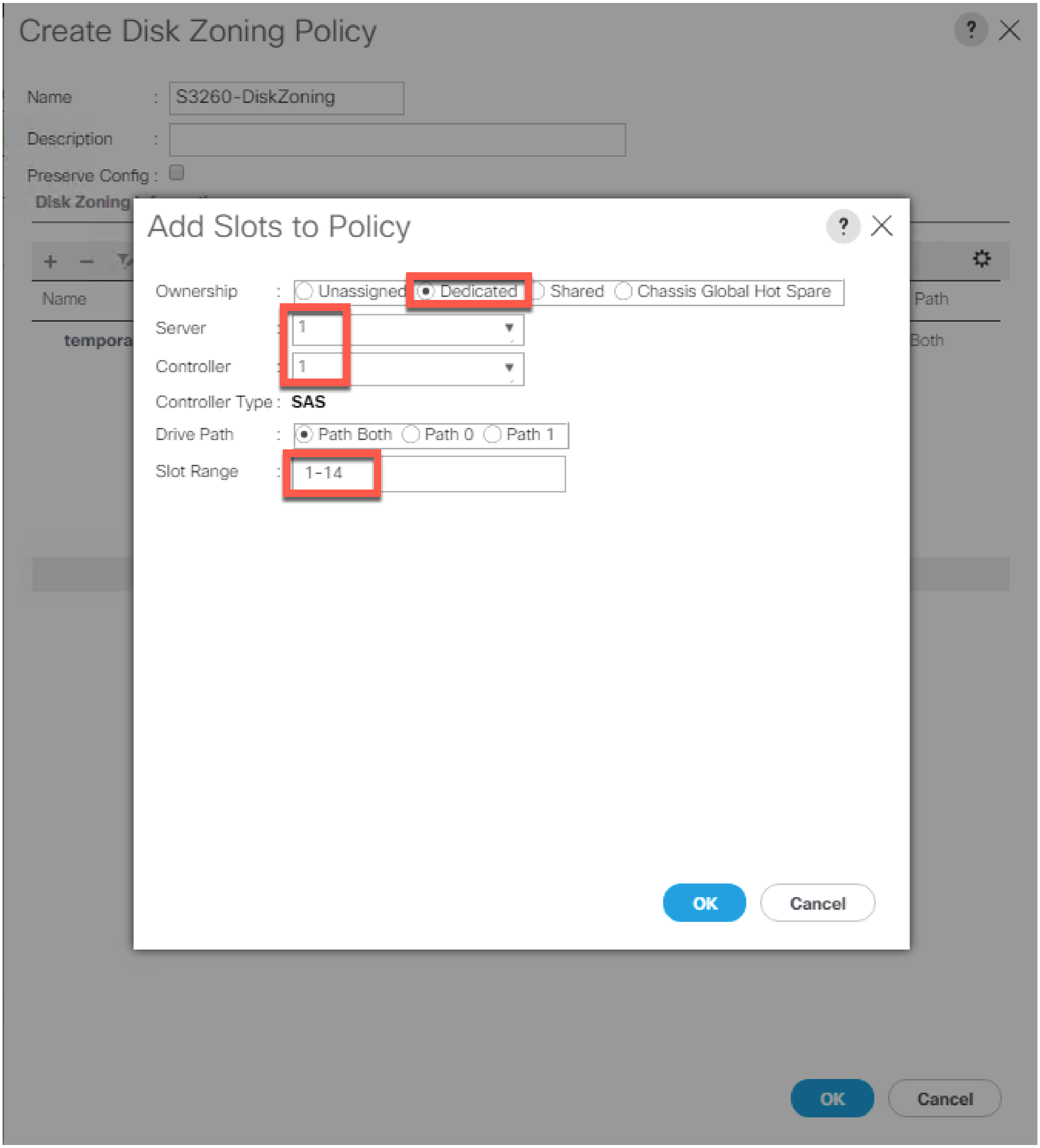
5. Create Disk Zoning Policy.

6. Click **Add**.

7. Select Dedicated under Ownership.

8. Select **Server** 1 and Select **Controller** 1.

9. Add **Slot Range 1-14** for the top node of the Cisco UCS S3260 Storage Server and click **OK**.



Create Disk Zoning Policy

Name : S3260-DiskZoning

Description :

Preserve Config : □

Add Slots to Policy

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

Server : 1

Controller : 1

Controller Type : SAS

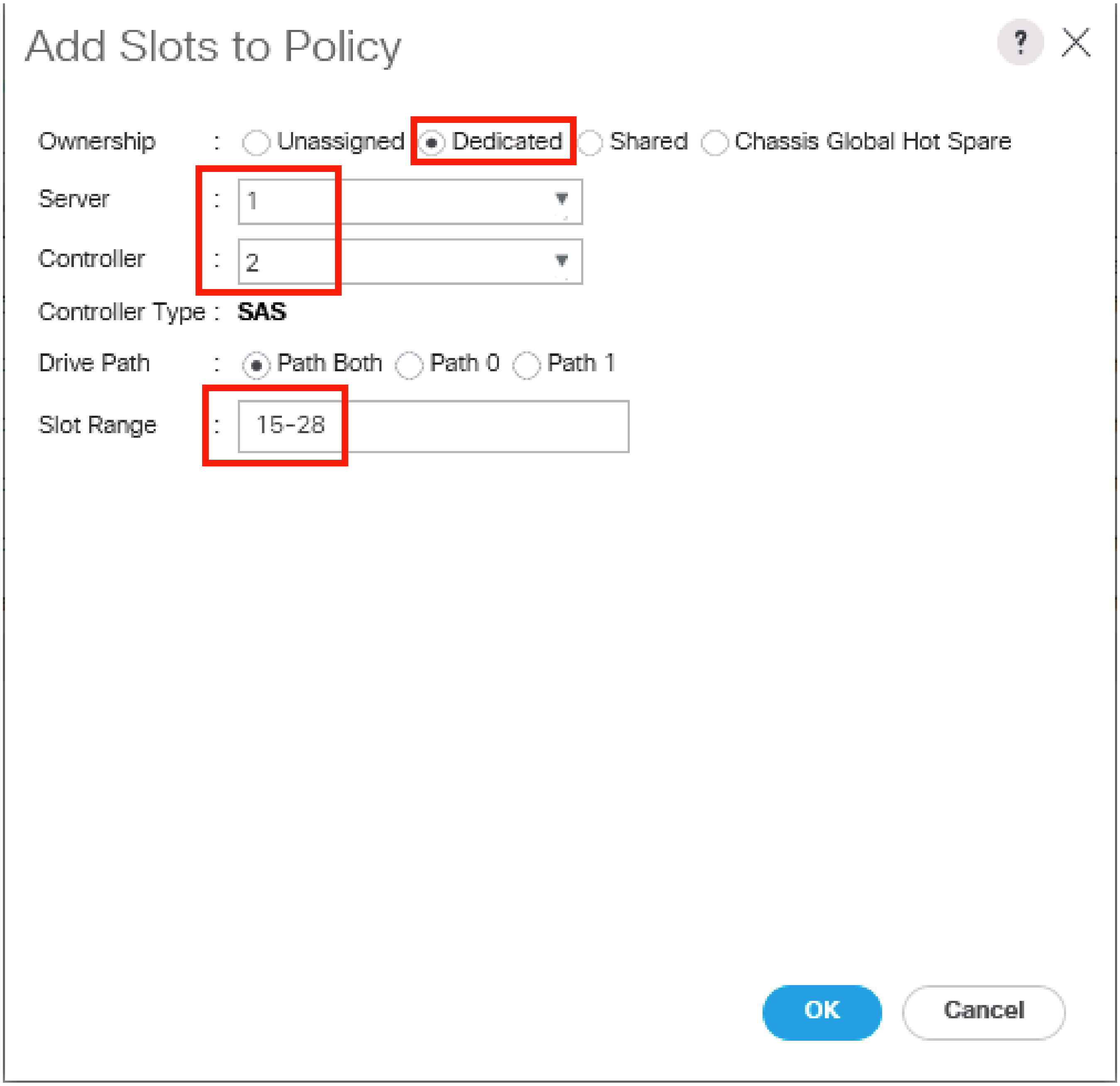
Drive Path : ● Path Both ○ Path 0 ○ Path 1

Slot Range : 1-14

10. Select **Server** 1 and Select **Controller** 2.

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

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



Add Slots to Policy

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

Server : 1

Controller : 2

Controller Type : SAS

Drive Path : ● Path Both ○ Path 0 ○ Path 1

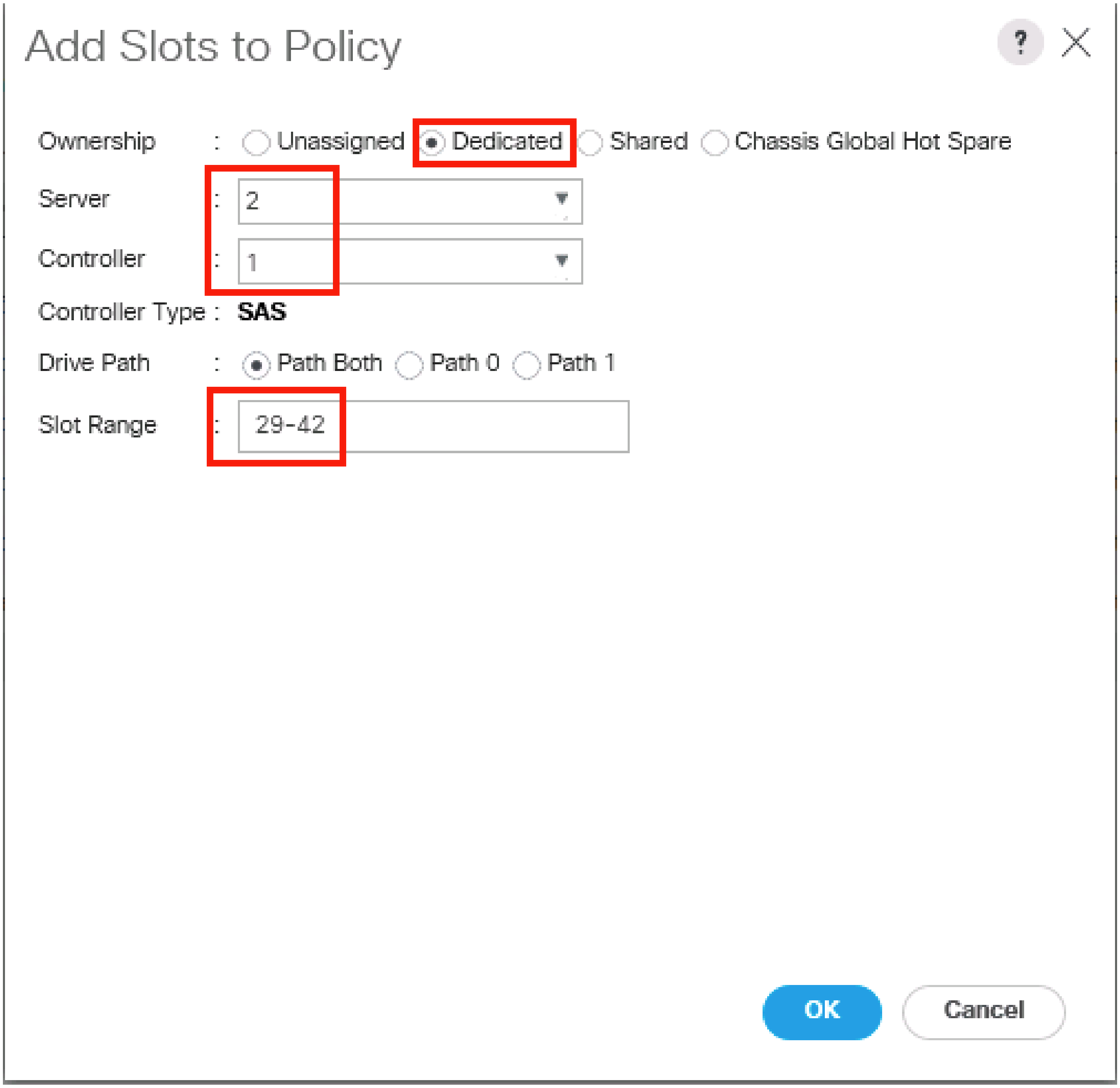
Slot Range : 15-28

13. Click **Add**.

14. Select Dedicated under Ownership.

15. Select **Server 2** and Select Controller 1.

16. Add **Slot Range 29-42** for the bottom node of the Cisco UCS S3260 Storage Server and click OK.



Add Slots to Policy

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

Server : 2

Controller : 1

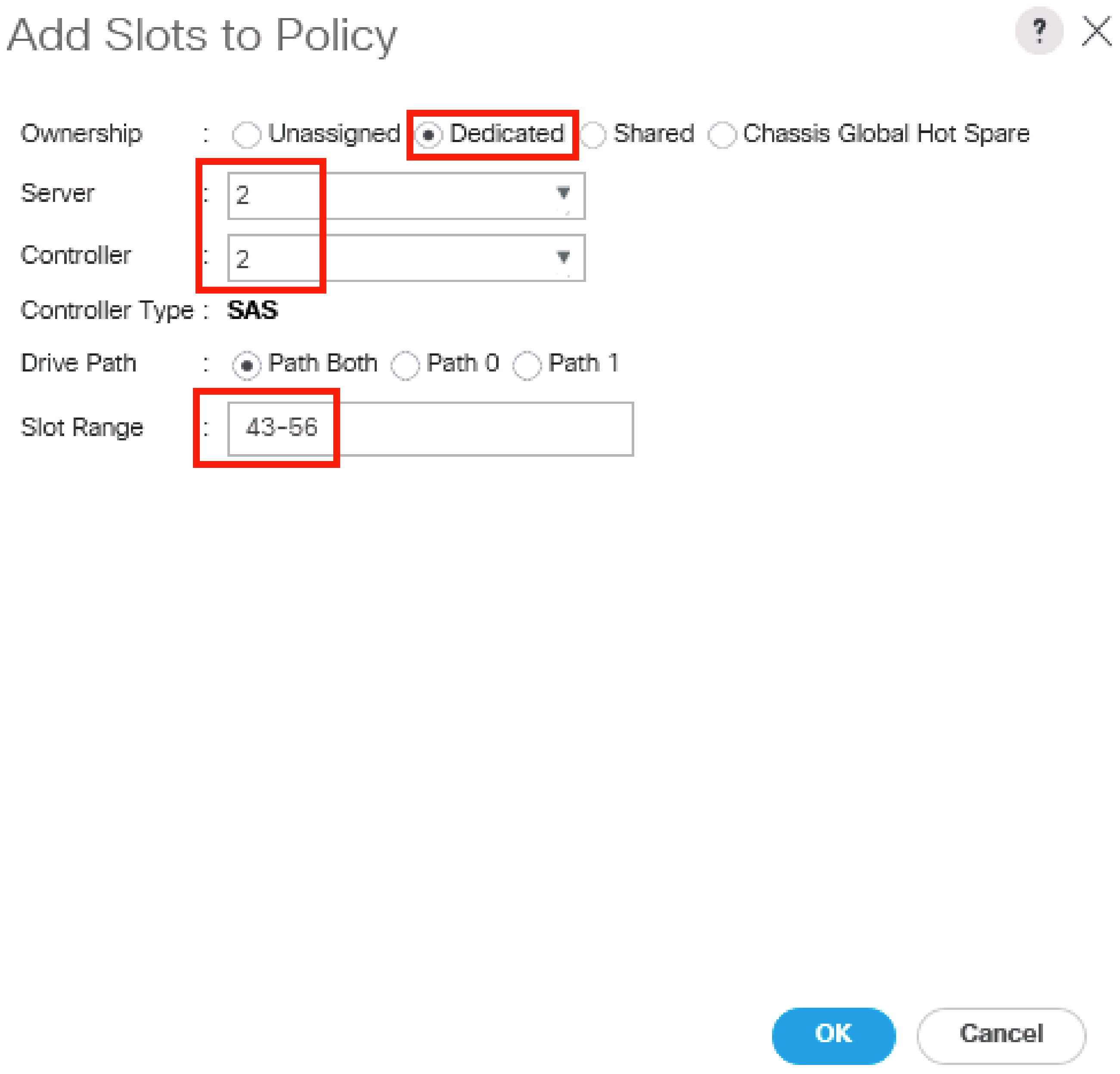
Controller Type : SAS

Drive Path : ● Path Both ○ Path 0 ○ Path 1

Slot Range : 29-42

17. Select **`Server 2`** and Select Controller 2.

18. Add **`Slot Range 43-56`** for the bottom node of the Cisco UCS S3260 Storage Server and click OK.



Add Slots to Policy

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

Server : 2

Controller : 2

Controller Type : **SAS**

Drive Path : ● Path Both ○ Path 0 ○ Path 1

Slot Range : 43-56

Create Chassis Profile Template

To create a Chassis Profile Template, follow these steps:

1. Select the **`Chassis`** tab in the Cisco UCS Manager GUI.
2. Go to Chassis > Chassis Profile Templates and right-click Create Chassis Profile Template.
3. Type in S3260-Chassis in the Name field.
4. Under Type, select Updating Template.

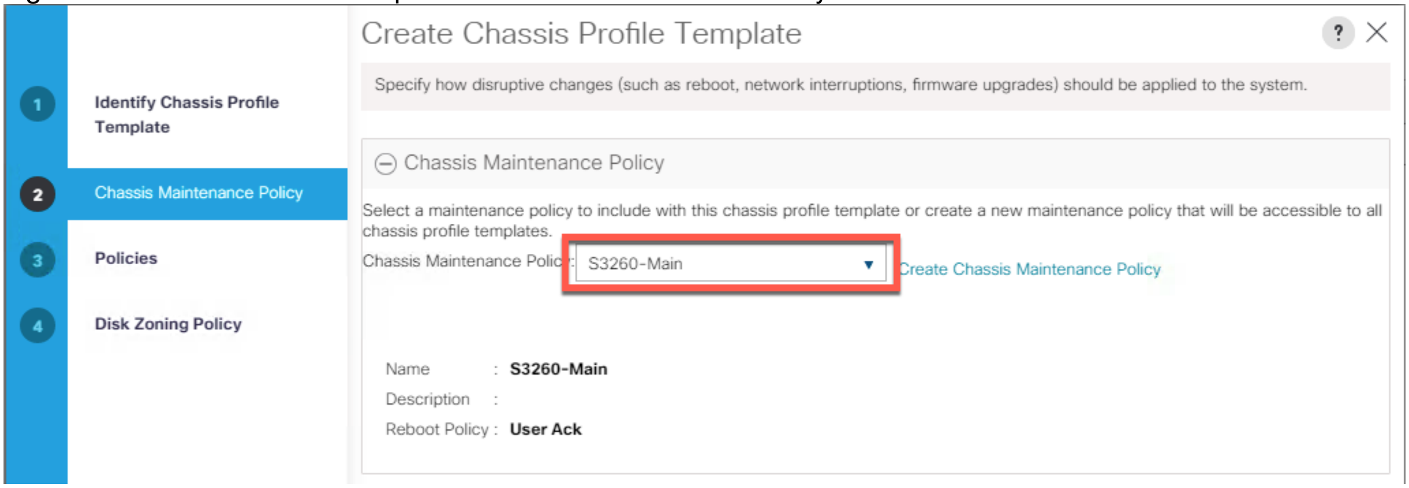
5. (Optional) Enter a description in the **Description** field.

6. Create Chassis Profile Template

7. Select **Next**.

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

Figure 29 Chassis Profile Template – Chassis Maintenance Policy


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

Create Chassis Profile Template

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

⊖ Chassis Maintenance Policy

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

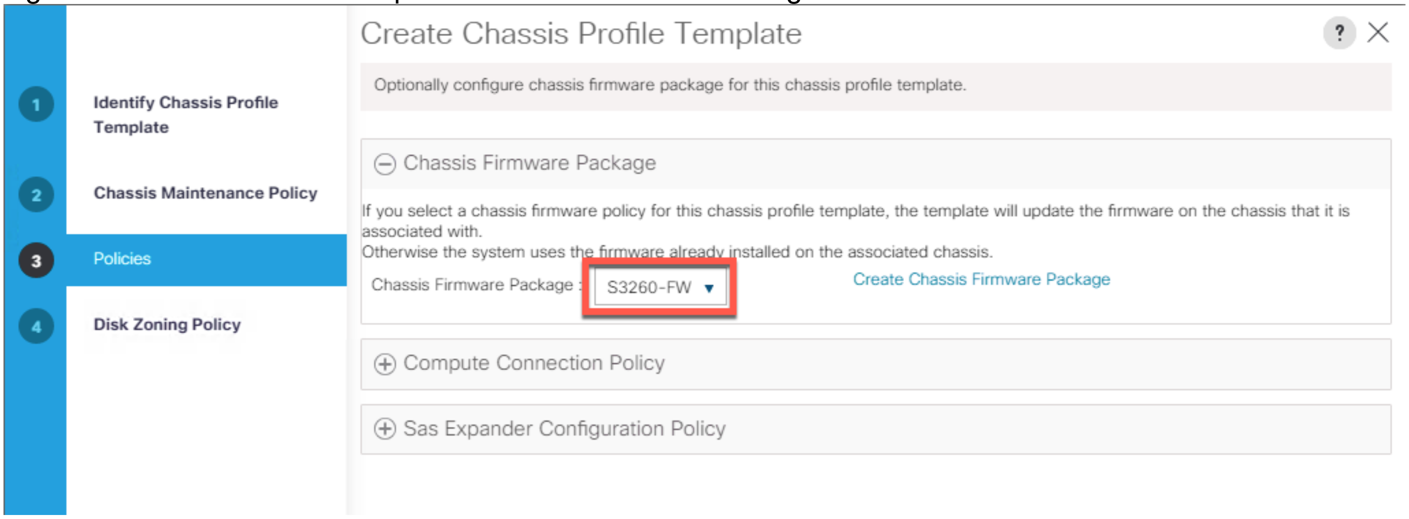
Chassis Maintenance Policy: S3260-Main ▾

Name : **S3260-Main**
Description :
Reboot Policy : **User Ack**

9. Select **Next**.

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

Figure 30 Chassis Profile Template – Chassis Firmware Package


1 Identify Chassis Profile Template
2 Chassis Maintenance Policy
3 Policies
4 Disk Zoning Policy
?
Create Chassis Firmware Package

Create Chassis Profile Template

Optionally configure chassis firmware package for this chassis profile template.

⊖ Chassis Firmware Package

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

Chassis Firmware Package : S3260-FW ▾

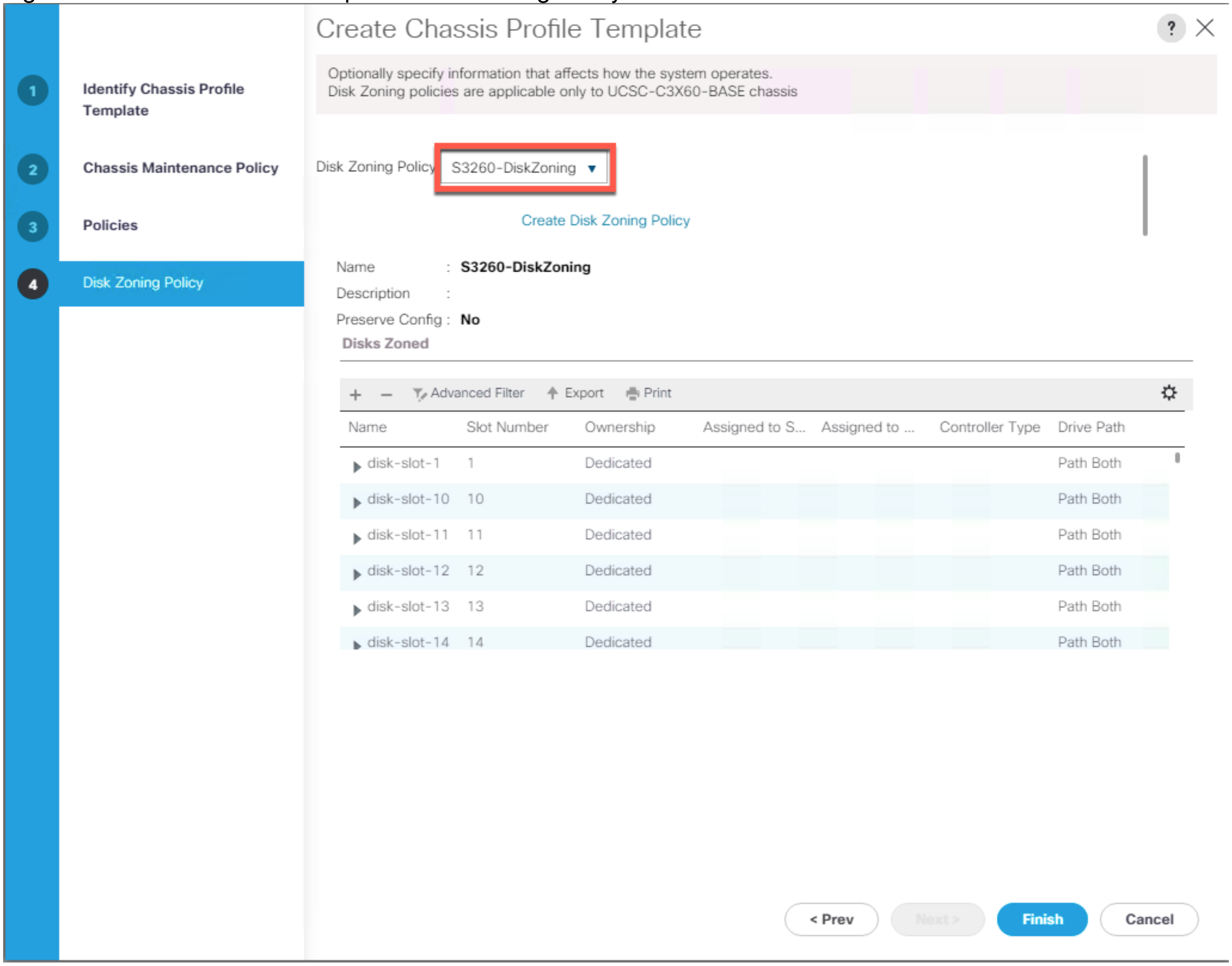
⊕ Compute Connection Policy

⊕ Sas Expander Configuration Policy

11. Select Next.

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

Figure 31   Chassis Profile Template – Disk Zoning Policy


1 Identify Chassis Profile Template
2 Chassis Maintenance Policy
3 Policies
4 Disk Zoning Policy
+ – Advanced Filter Export Print
< Prev
Next >
Finish
Cancel

Create Chassis Profile Template

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

Disk Zoning Policy: S3260-DiskZoning ▼

Create Disk Zoning Policy

Name : **S3260-DiskZoning**
Description :
Preserve Config : **No**

Disks Zoned

Name	Slot Number	Ownership	Assigned to S...	Assigned to ...	Controller Type	Drive Path
disk-slot-1	1	Dedicated				Path Both
disk-slot-10	10	Dedicated				Path Both
disk-slot-11	11	Dedicated				Path Both
disk-slot-12	12	Dedicated				Path Both
disk-slot-13	13	Dedicated				Path Both
disk-slot-14	14	Dedicated				Path Both

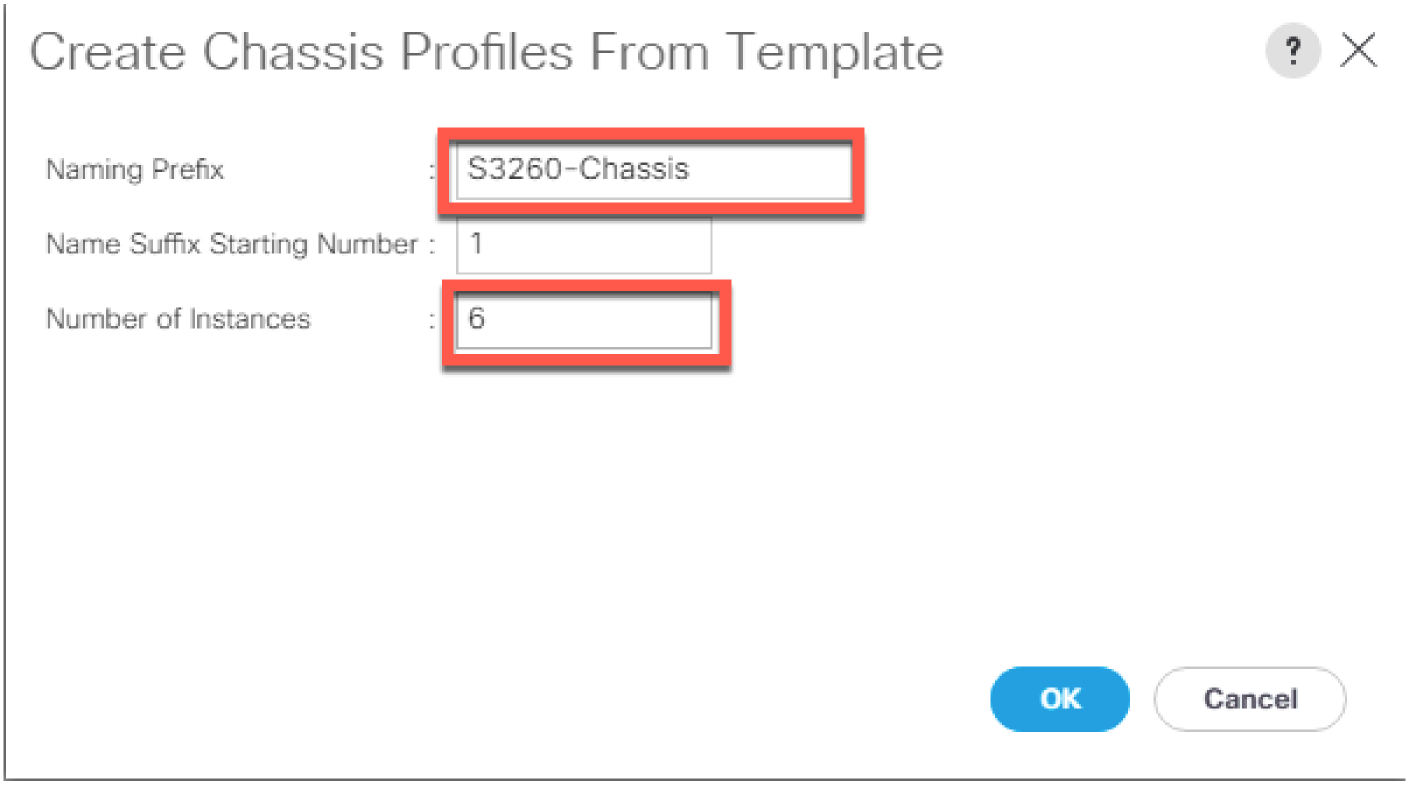
13. Click **Finish** and then click **OK** again.

Create Chassis Profile from Template

To create the Chassis Profiles from the previous created Chassis Profile Template, follow these steps:

1. Select the **Chassis** tab in the left pane of the Cisco UCS Manager GUI.
2. Go to Chassis > Chassis Profile Templates and select “S3260-Chassis” you created previously.
3. Then right click to select “Create Chassis Profiles from Template."
4. Type in **S3260-Chassis** in the **Name** field.
5. Leave the Name Suffix Starting Number untouched.
6. Enter **6** for the **Number of Instances** for all connected Cisco UCS S3260 Storage Server.

7. Click **OK**.



Create Chassis Profiles From Template

Naming Prefix : S3260-Chassis

Name Suffix Starting Number : 1

Number of Instances : 6

Associate Chassis Profile

To associate all previous created Chassis Profile, follow these 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-Chassis1.”
3. Right-click Change Chassis Profile Association.
4. Under Chassis Assignment, choose Select existing Chassis from the drop-down list.
5. Under **Available Chassis**, select ID **1**.
6. Click **OK** and then click **OK** again.
7. Repeat the steps for the other two Chassis Profiles by selecting the IDs 2 – 6.
8. A pop-up will appear on the top right side. Click Chassis Profiles and Acknowledge All Chassis profiles.
9. Click Apply.
10. Click OK.

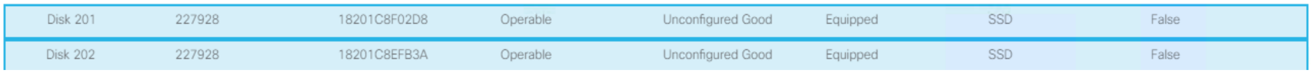
After the association of the Chassis profile with Disk zoning policy, the disks distribution between the nodes may get corrected in few minutes

Create Storage Profiles

Convert the Disks to Unconfigured Good

The boot disks on both Cisco UCS C220 and Cisco UCS S3260 should be Unconfigured Good before installing the operating system. If not, follow these steps to make them Unconfigured Good.

1. For S3260 Server:
2. Select Chassis -> Servers -> Server1.
3. In the right pane, select Inventory-> Storage-> **Disks.**
4. Select the Bootable SSD disks, right click and make them as **Unconfigured Good**.



Disk 201	227928	18201C8F02D8	Operable	Unconfigured Good	Equipped	SSD	False
Disk 202	227928	18201C8EFB3A	Operable	Unconfigured Good	Equipped	SSD	False

5. Repeat steps 1-4 for Server2 and all other Cisco UCS S3260 servers in the cluster.
6. Repeat steps 1-4 for all Cisco UCS C220 servers.

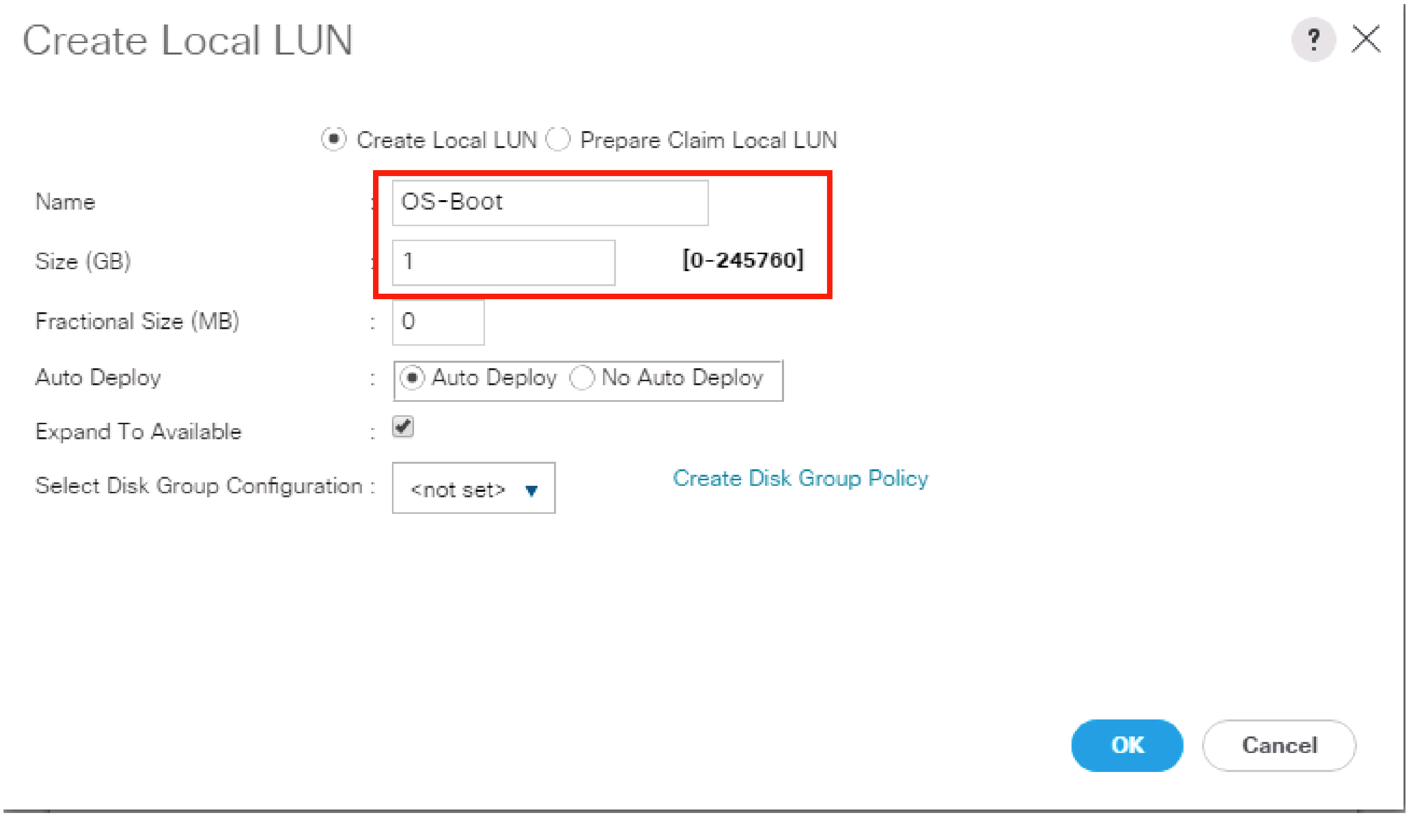
In case of using any automation like PXE boot, etc. for Cisco UCS S3260, you may have to convert all other disks into unconfigured good too. The Boot LUN created (after applying service profiles) for OS will be the only LUN visible to PXE server where OS will be installed. The non-boot disks can be turned back to JBOD after OS installation either through UCSM GUI or utilities like storcli.

Create Storage Profiles for Cisco UCS S3260 Storage Server

To create the Storage Profile for the top node of the Cisco UCS S3260 Storage Server, follow these 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 **Sever1** in the **Name** field.
4. (Optional) Enter a description in the **Description** field.
5. Click **Add**.
6. Type in "**OS-Boot"** in the **Name** field.
7. Configure as follows:
   - Create Local LUN

- Size (GB) = 1
- Fractional Size (MB) = 0
- Auto Deploy
- Select Expand To Available


Create Local LUN

● Create Local LUN ○ Prepare Claim Local LUN

Name : OS-Boot

Size (GB) : 1 [0-245760]

Fractional Size (MB) : 0

Auto Deploy : ● Auto Deploy ○ No Auto Deploy

Expand To Available : ☑

Select Disk Group Configuration : <not set>

8. Click “Create Disk Group Policy” to Create RAID1 LUN.

9. Type in **`Server1`** in the `Name` field.

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

11. RAID Level = RAID 1 Mirrored.

12. Select Disk Group Configuration (Manual).

13. Click **`Add`**.

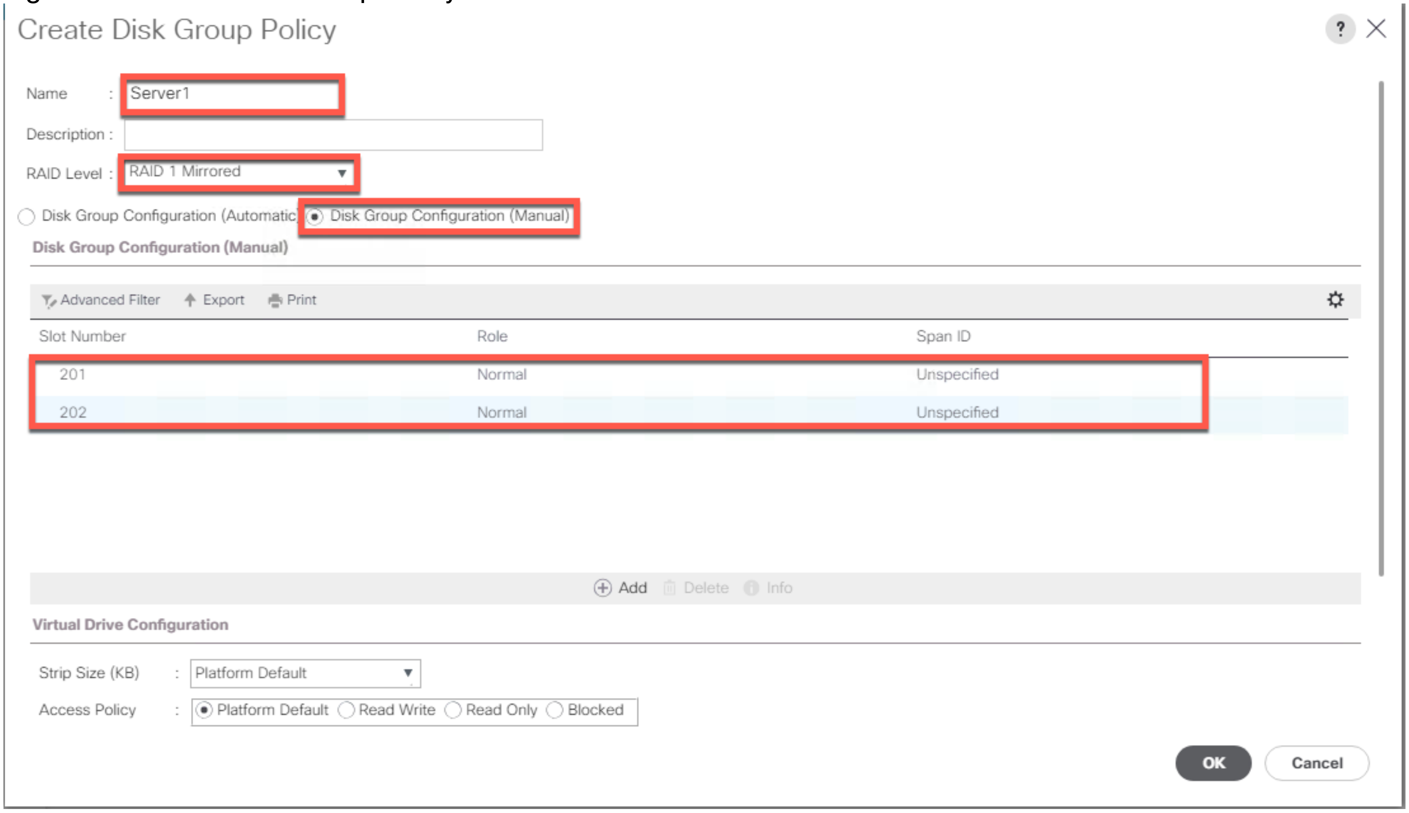
14. Type in **`201`** for **`Slot Number`**.

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

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

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

Figure 32 Create Disk Group Policy



Create Disk Group Policy

Name : Server1

Description :

RAID Level : RAID 1 Mirrored

○ Disk Group Configuration (Automatic) ● Disk Group Configuration (Manual)

Disk Group Configuration (Manual)

Slot Number	Role	Span ID
201	Normal	Unspecified
202	Normal	Unspecified

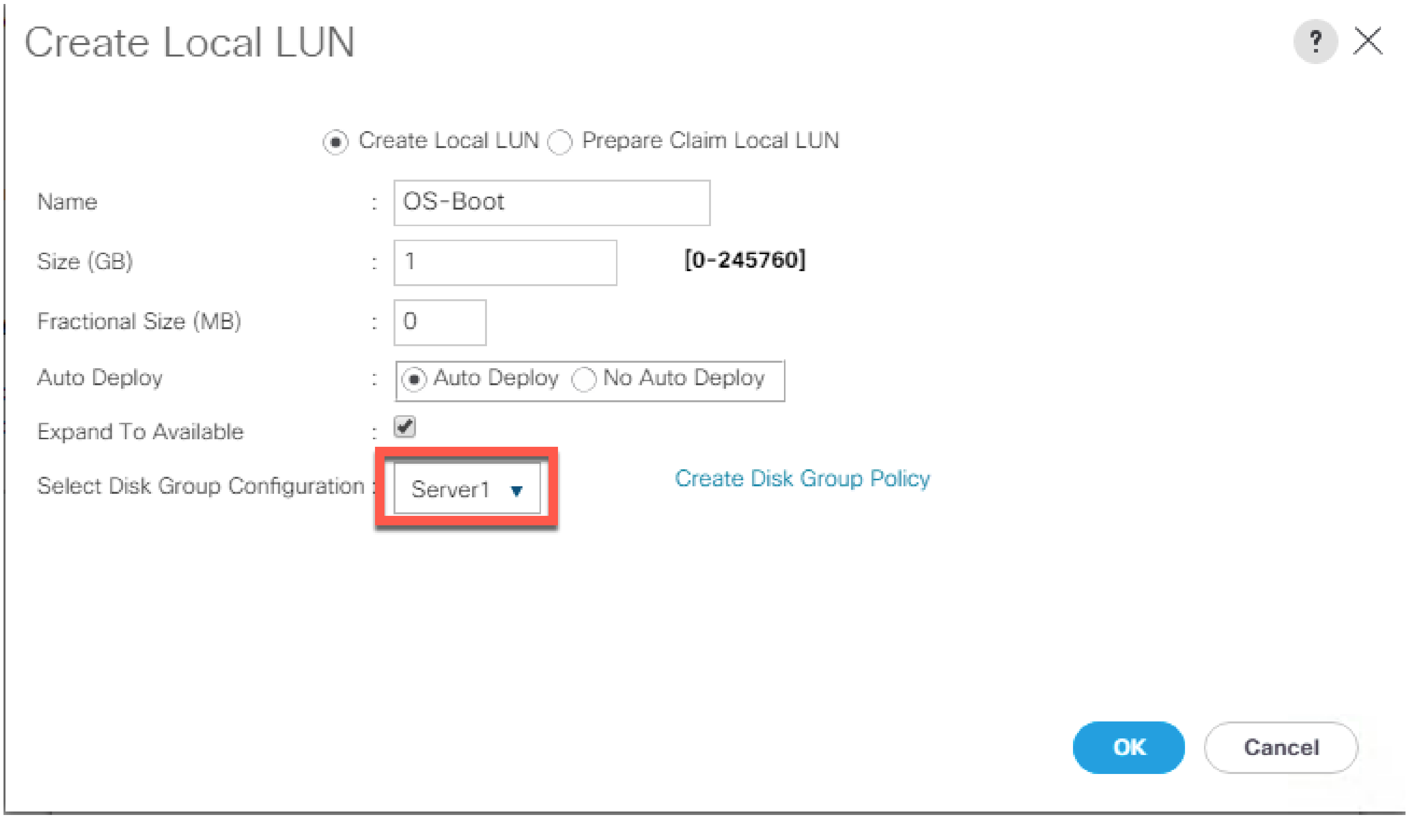
Virtual Drive Configuration

Strip Size (KB) : Platform Default

Access Policy : ● Platform Default ○ Read Write ○ Read Only ○ Blocked

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

19. Select Disk Group Configuration.



Create Local LUN

● Create Local LUN ○ Prepare Claim Local LUN

Name : OS-Boot

Size (GB) : 1 [0-245760]

Fractional Size (MB) : 0

Auto Deploy : ● Auto Deploy ○ No Auto Deploy

Expand To Available : ☑

Select Disk Group Configuration : Server1

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

21. Storage Profile for the second Server:

22. Follow steps 1-21 to create a storage profile for the second server in the chassis. The Storage Profile Server1 is for the top server and Server2 for the bottom server of S3260

Create Storage Profile for Cisco UCS C220 M5 Rack-Mount Servers

To create a Storage Profile for the Cisco UCS C220 M5, follow these steps:

1. Select **Storage** in the left pane of the UCSM GUI.

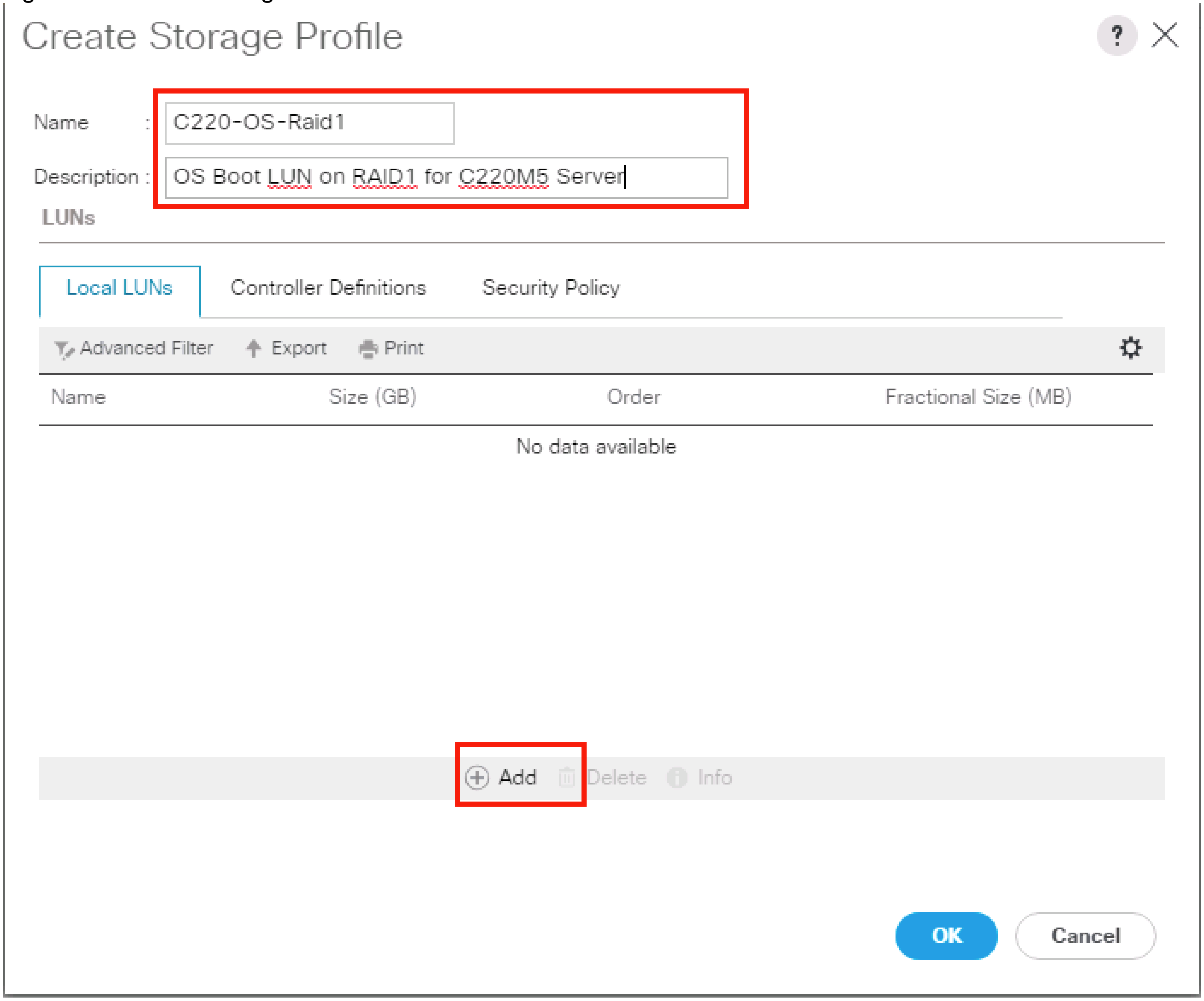
2. Go to Storage > Storage Profiles and right-click Create Storage Profile.

3. Type in **C220-OS-Raid1** in the **Name** field.

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

5. Click **Add**.

Figure 33 Create Storage Profile for Cisco UCS C220 M5



Create Storage Profile

Name : C220-OS-Raid1

Description : OS Boot LUN on RAID1 for C220M5 Server

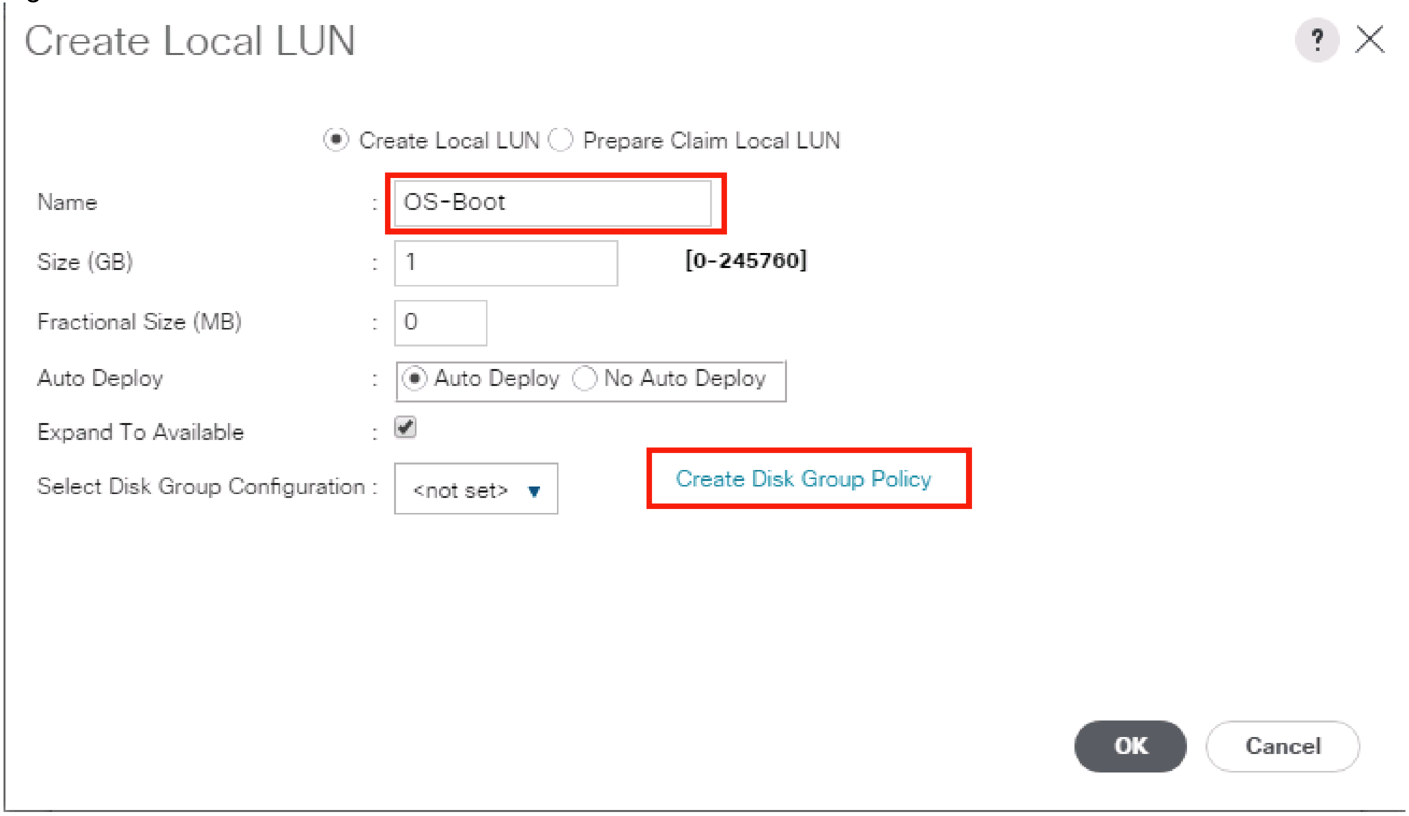
LUNs

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

7. Configure as follows:

   - Create Local LUN
   - Size (GB) = 1
   - Fractional Size (MB) = 0
   - Select Expand To Available
   - Auto Deploy

Figure 34 Create Local LUN



Create Local LUN

● Create Local LUN ○ Prepare Claim Local LUN

Name : OS-Boot

Size (GB) : 1 [0-245760]

Fractional Size (MB) : 0

Auto Deploy : ● Auto Deploy ○ No Auto Deploy

Expand To Available : ☑

Select Disk Group Configuration : <not set>

8. Click Create Disk Group Policy to Create RAID1 LUN.

9. Type in **RAID1-C220** in the **Name** field.

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

11. RAID Level = RAID 1 Mirrored.

12. Select Disk Group Configuration (Manual).

13. Click **Add**.

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

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

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

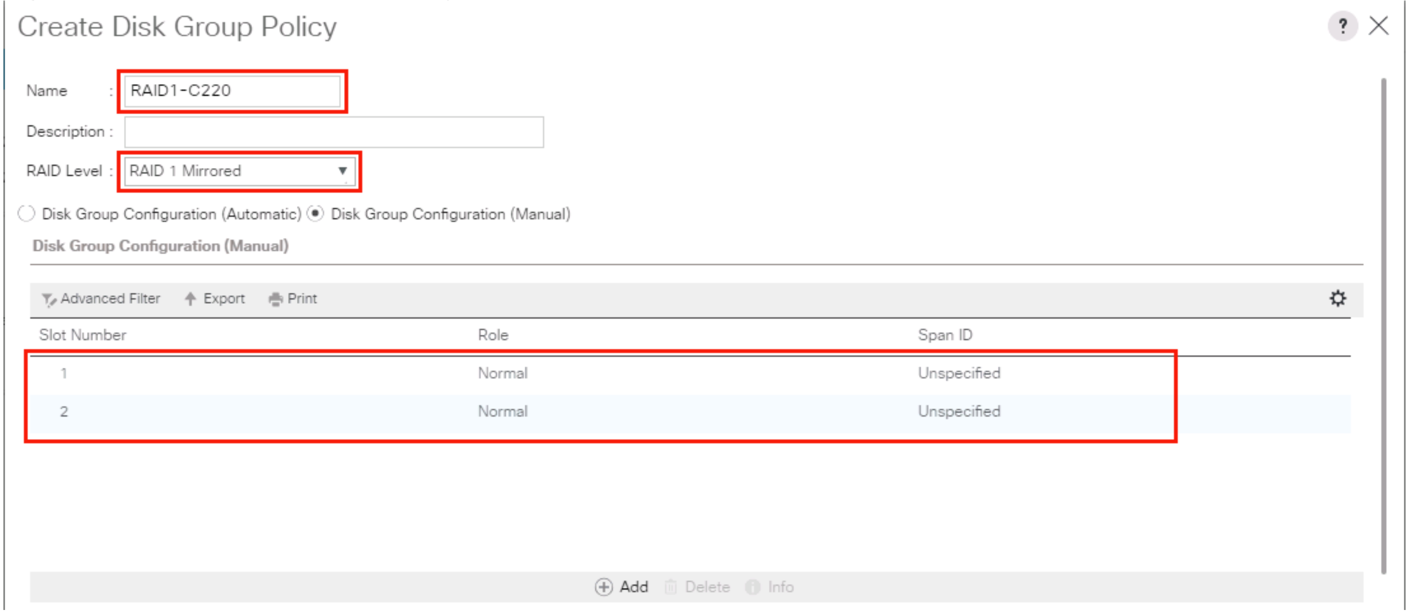
17. Under “Change Virtual Drive Configuration:”

    a. Modify Access Policy as “Read Write” and Read Policy as “Read Ahead.”

    b. Modify Write Cache Policy as “Write Back Good BBU” and IO Policy as “Cache.”

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

Figure 35 Create Disk Group Policy for Cisco UCS C220 M5



Create Disk Group Policy

Name : RAID1-C220

Description :

RAID Level : RAID 1 Mirrored

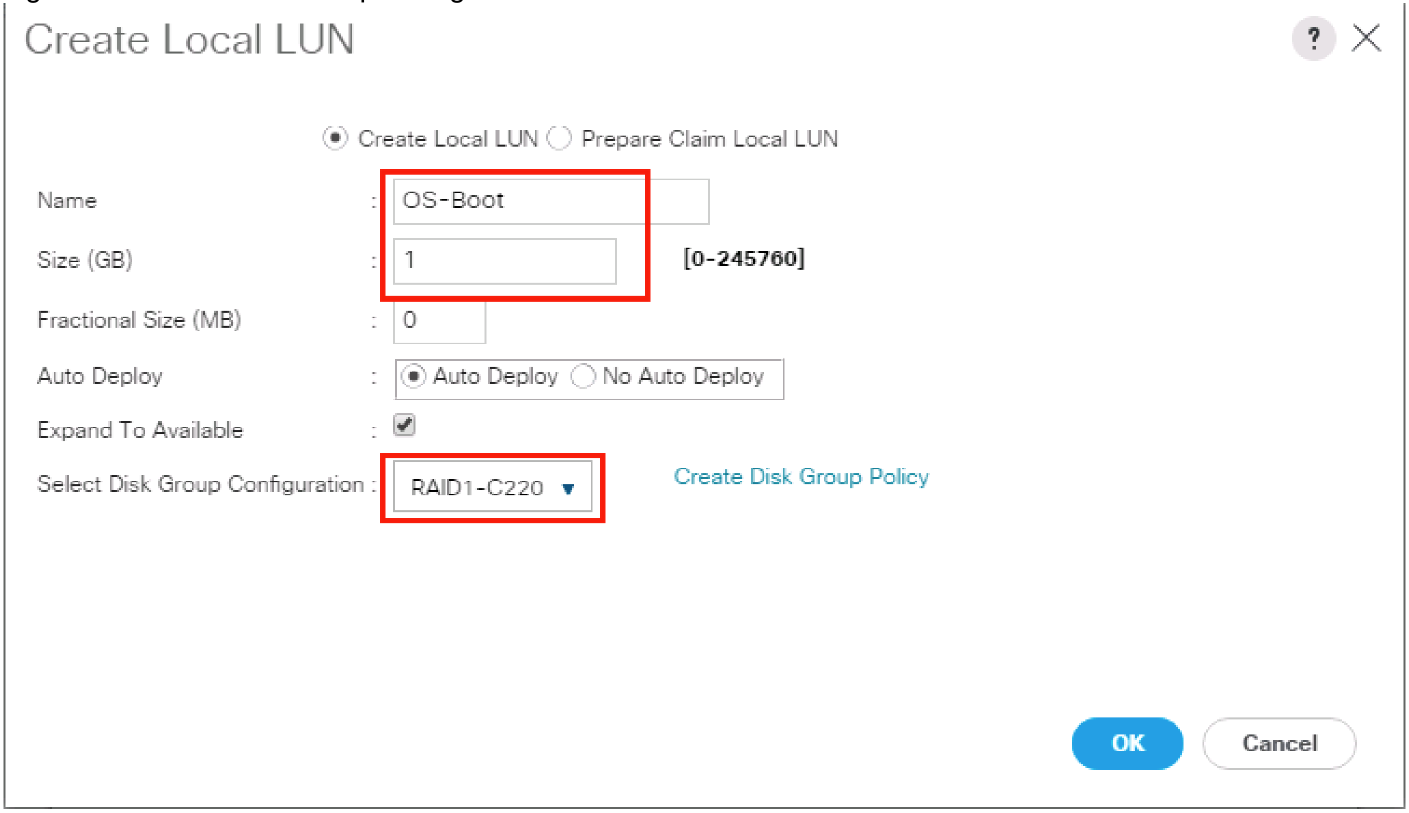
○ Disk Group Configuration (Automatic) ● Disk Group Configuration (Manual)

Disk Group Configuration (Manual)

Slot Number	Role	Span ID
1	Normal	Unspecified
2	Normal	Unspecified

19. Select the previously created Disk Group Policy for the C220 M5 Boot Disks under **`Select Disk Group Configuration`**.

Figure 36 Create Disk Group Configuration for Cisco UCS C220 M5



Create Local LUN

● Create Local LUN ○ Prepare Claim Local LUN

Name : OS-Boot

Size (GB) : 1 [0-245760]

Fractional Size (MB) : 0

Auto Deploy : ● Auto Deploy ○ No Auto Deploy

Expand To Available : ☑

Select Disk Group Configuration : RAID1-C220

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

Create a Service Profile Templates

Create Service Profile Template for Cisco UCS S3260 Storage Server1 and Server2

To create a Service Profile Template, follow these steps:

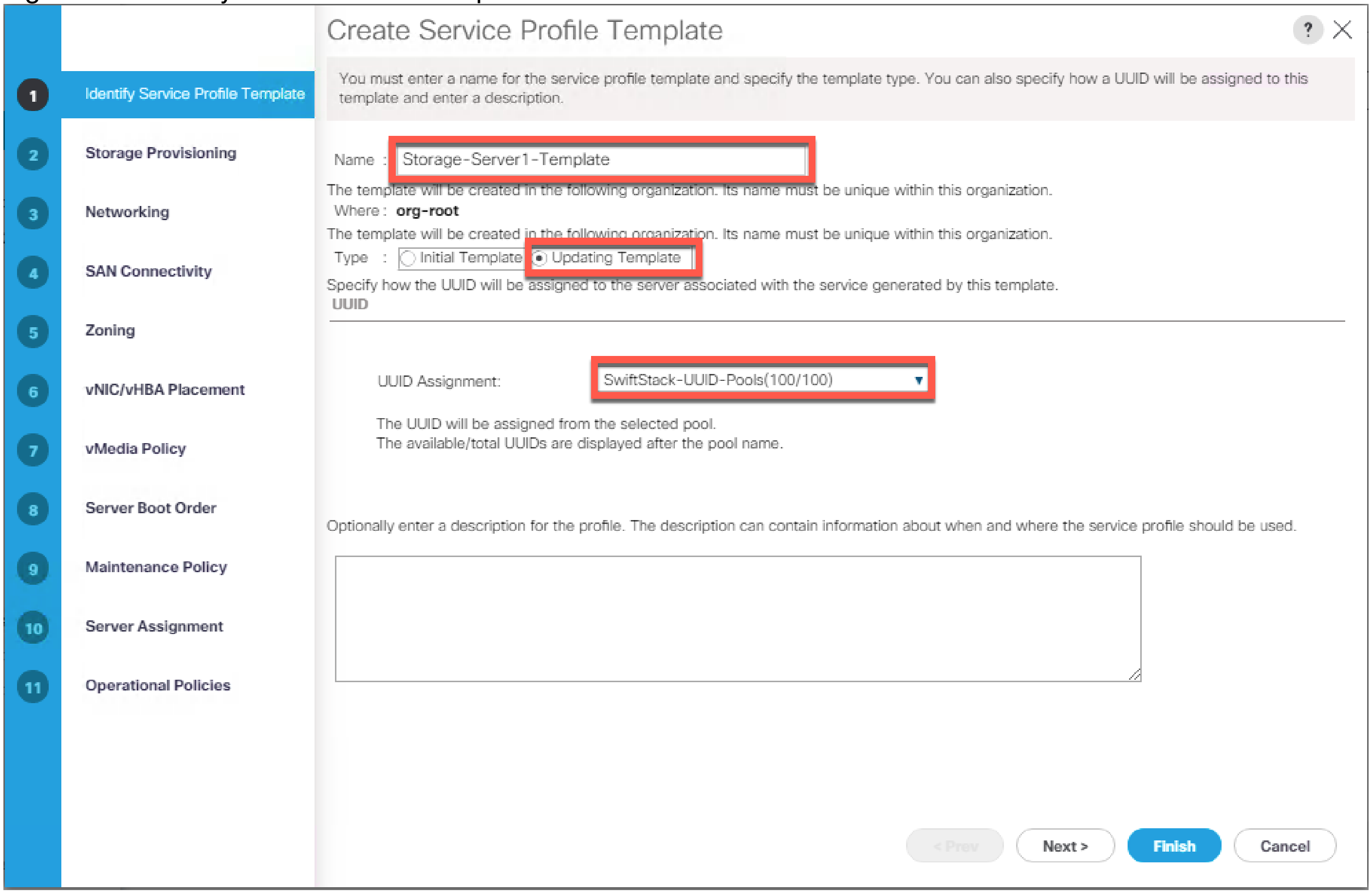
1. Select **`Servers`** in the left pane of the Cisco UCS Manager GUI.
2. Go to Servers > Service Profile Templates > root and right-click Create Service Profile Template.

Identify Service Profile Template

To identify the Service Profile template, follow these steps:

1. Type in "Storage-Server1-Template" in the Name field.
2. Select Template Type **"`Updating Template`"**
3. In the **`UUID Assignment`** section, select the UUID Pool you created in the beginning.
4. (Optional) Enter a description in the **`Description`** field.

Figure 37 Identify Service Profile Template



Create Service Profile Template

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

Name : Storage-Server1-Template

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

Where : **org-root**

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

Type : ○ Initial Template ● Updating Template

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

UUID

UUID Assignment: SwiftStack-UUID-Pools(100/100)

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

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

5. Click **`Next`**.

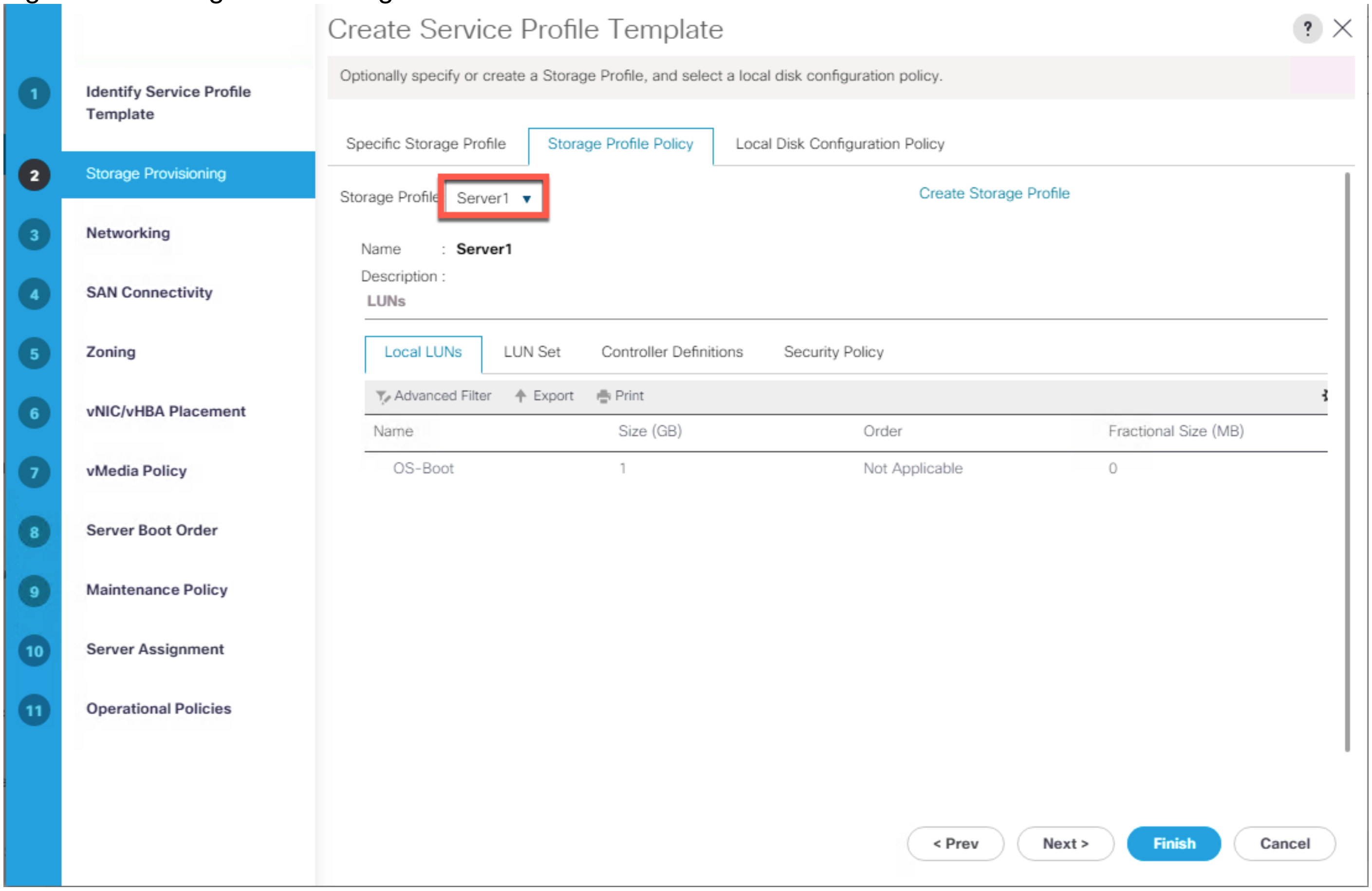
Storage Provisioning

To provision the storage profile, follow these steps:

1. Go to the **`Storage Profile Policy`** tab and select the Storage Profile **`Server1`** for the top node of the Cisco UCS S3260 Storage Server you created before.

2. Click **`Next`**.

Figure 38 Storage Provisioning



Create Service Profile Template

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

Storage Profile: Server1

Name : Server1

Description :

LUNs

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

Networking

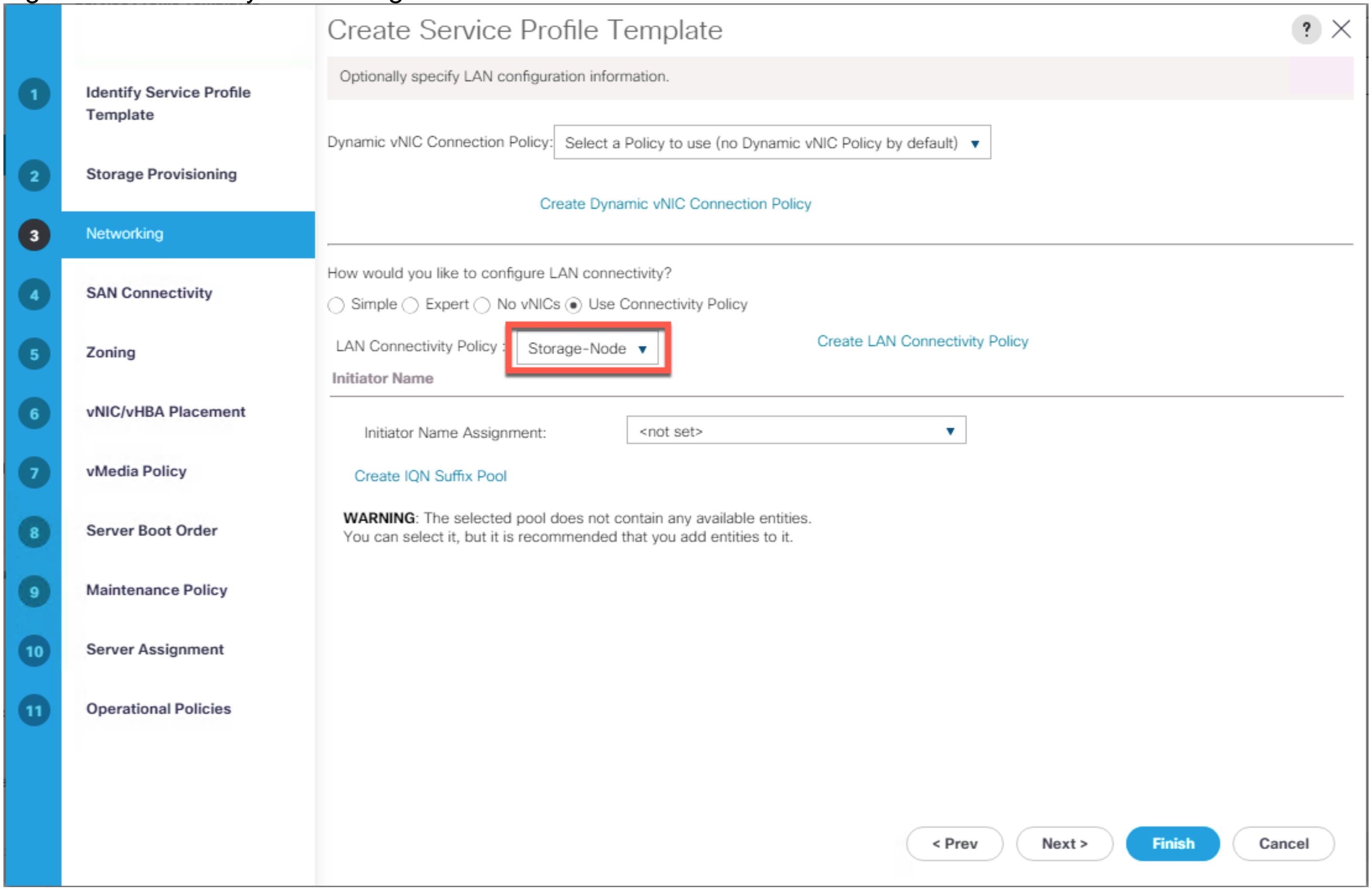
To configure networking, follow these steps:

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.

Figure 39 Summary Networking



Create Service Profile Template

Optionally specify LAN configuration information.

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

How would you like to configure LAN connectivity?

○ Simple ○ Expert ○ No vNICs ● Use Connectivity Policy

LAN Connectivity Policy : Storage-Node

Initiator Name

Initiator Name Assignment: <not set>

WARNING: The selected pool does not contain any available entities.
You can select it, but it is recommended that you add entities to it.

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`**.

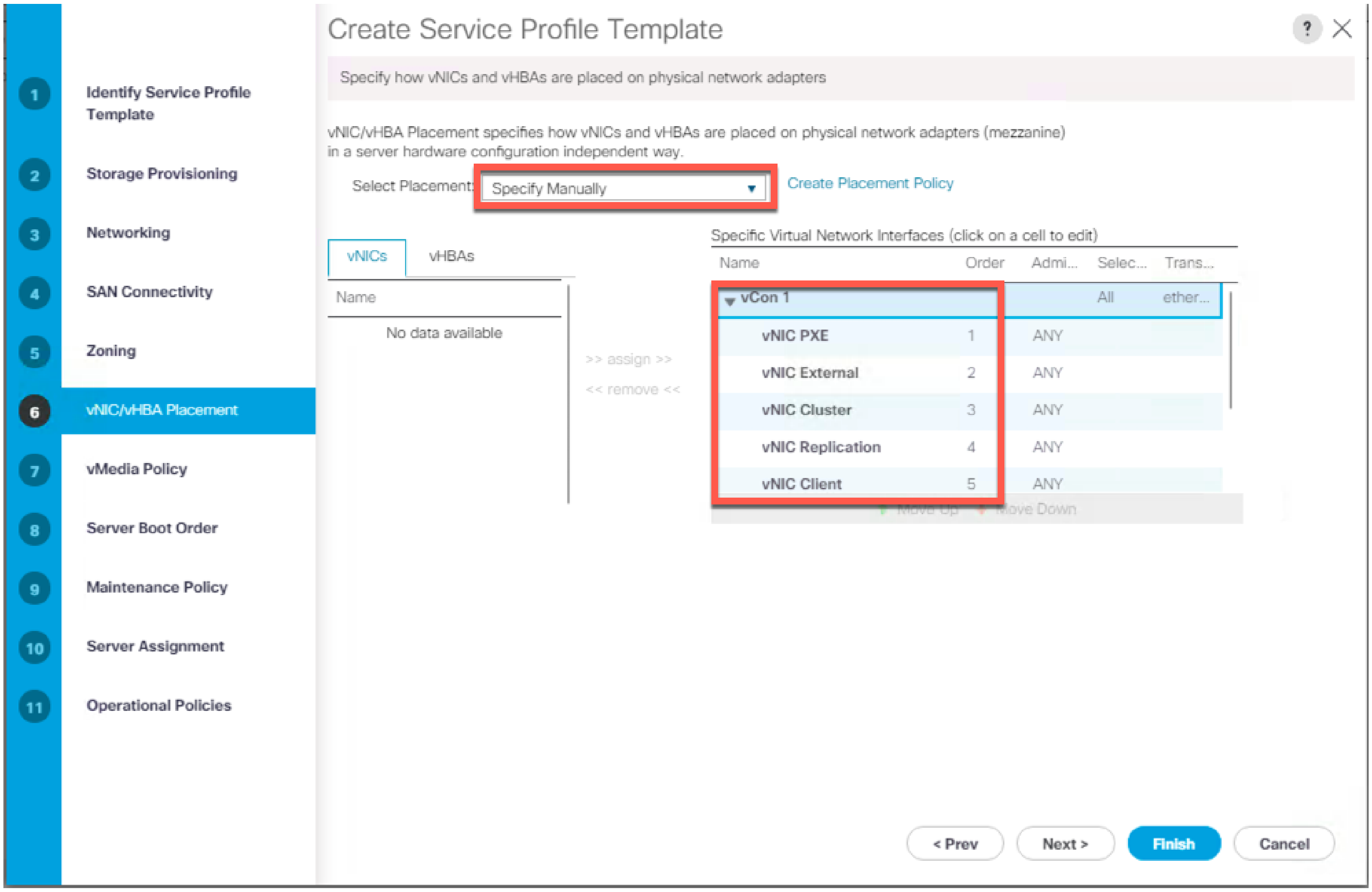
vNIC/vHBA Placement

To configure the vNIC/vHBA placement, follow these steps:

1. Select **`Specify Manually`** form the drop-down list.

2. Under PCI order section, Sort all the vNICs.

3. Make sure the vNICs order are listed as Mgmt-Network/PXE > 1, then followed by External-Network > 2 and Storage-Cluster > 3 Replication > 4 Client-Network >5



Create Service Profile Template

Specify how vNICs and vHBAs are placed on physical network adapters

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

Select Placement: Specify Manually ▾ Create Placement Policy

Name

No data available

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

Name	Order	Admi...	Selec...	Trans...
▾ vCon 1			All	ether...
vNIC PXE	1	ANY		
vNIC External	2	ANY		
vNIC Cluster	3	ANY		
vNIC Replication	4	ANY		
vNIC Client	5	ANY		

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

5. Click **Next**.

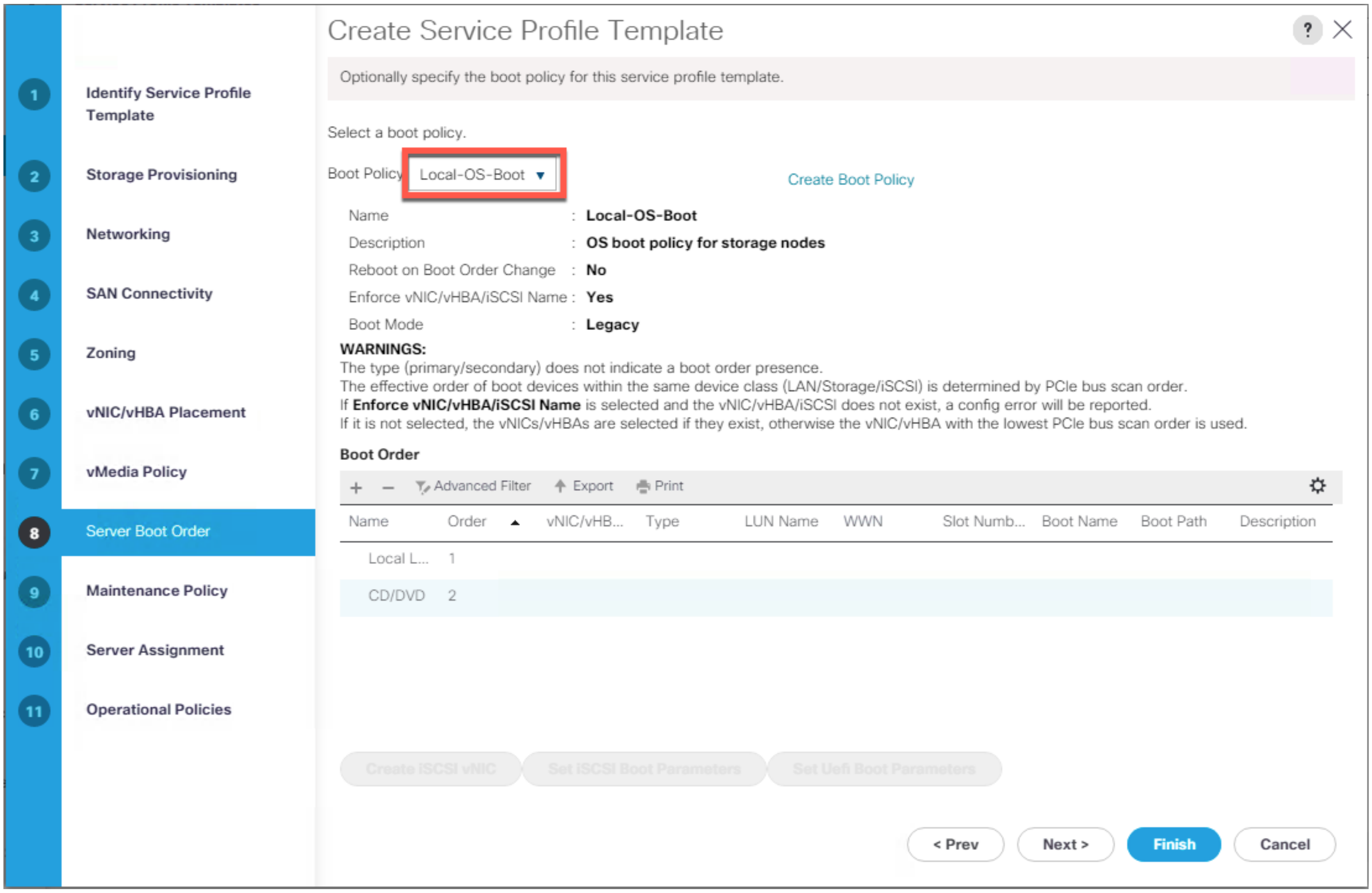
Server Boot Order

To configure the server boot order, follow these steps:

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

2. Server Boot Order.

3. Click Next.



Create Service Profile Template

Optionally specify the boot policy for this service profile template.

Select a boot policy.

Boot Policy: Local-OS-Boot

Name : Local-OS-Boot
Description : OS boot policy for storage nodes
Reboot on Boot Order Change : No
Enforce vNIC/vHBA/iSCSI Name : Yes
Boot Mode : Legacy

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

Boot Order

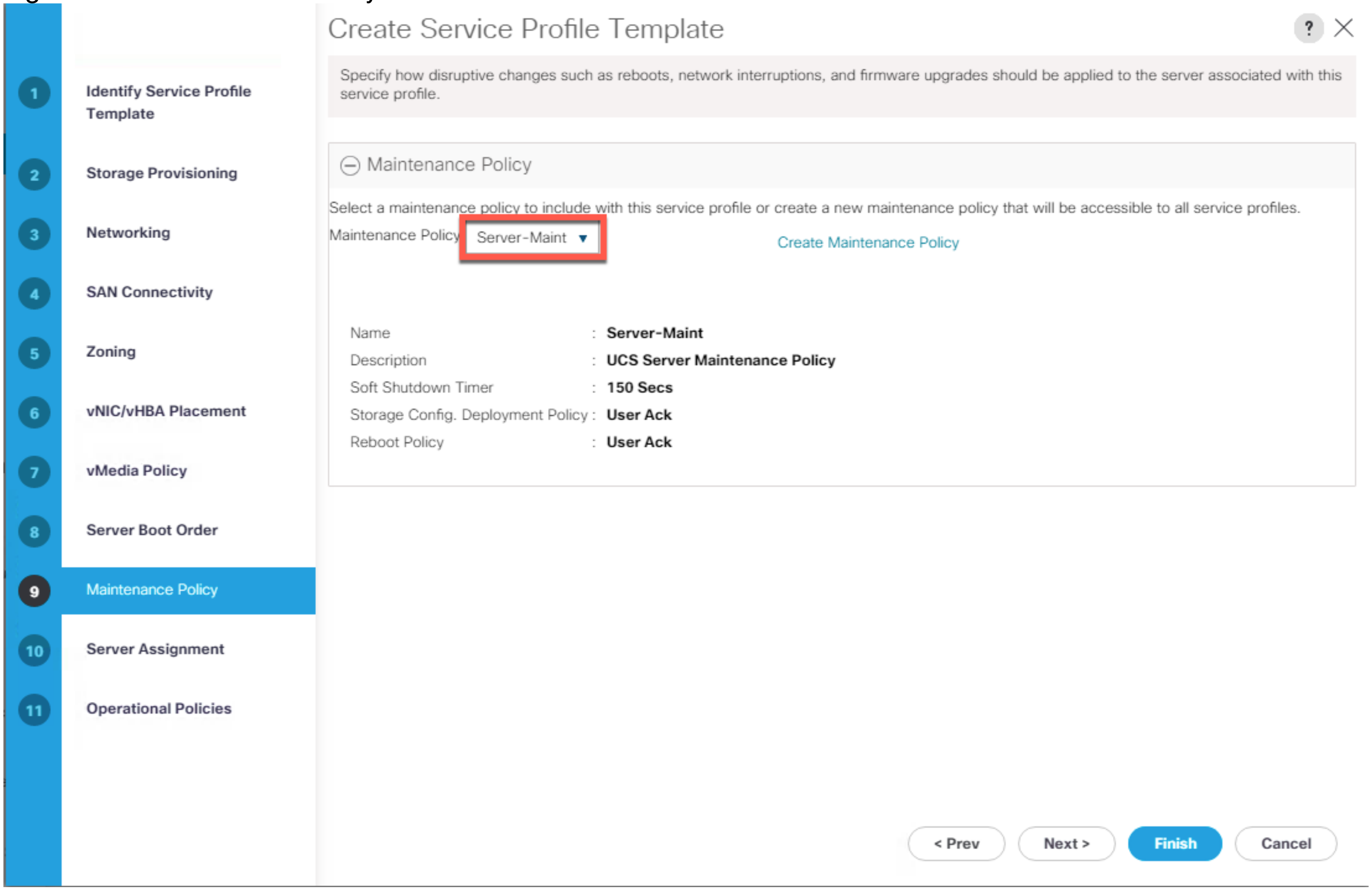
Name	Order	vNIC/vHB...	Type	LUN Name	WWN	Slot Numb...	Boot Name	Boot Path	Description
Local L...	1								
CD/DVD	2								

Maintenance Policy

To configure the maintenance policy, follow these steps:

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

Figure 40 Maintenance Policy



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

Maintenance Policy

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

Maintenance Policy: Server-Maint

Name : Server-Maint
Description : UCS Server Maintenance Policy
Soft Shutdown Timer : 150 Secs
Storage Config. Deployment Policy : User Ack
Reboot Policy : User Ack

2. Click **Next**.

3. Under Server Assignment, Leave everything else untouched.

4. Click **Next**.

Operational Policies

To configure the operational policies, follow these steps:

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

2. Repeat the steps for the Server2 of the Cisco UCS S3260 Storage Server by naming the template “Storage-Server2-Template.”

3. During Storage Provisioning tab, choose the Storage Profile for the Server2 “S3260-Server2-Node” you created previously.

Create Service Profiles from Template

This section explains how to create the appropriate Service Profiles from the previous Service Profile Templates. To create the first profile for the Server1 of the Cisco UCS S3260 Storage Server, follow these steps:

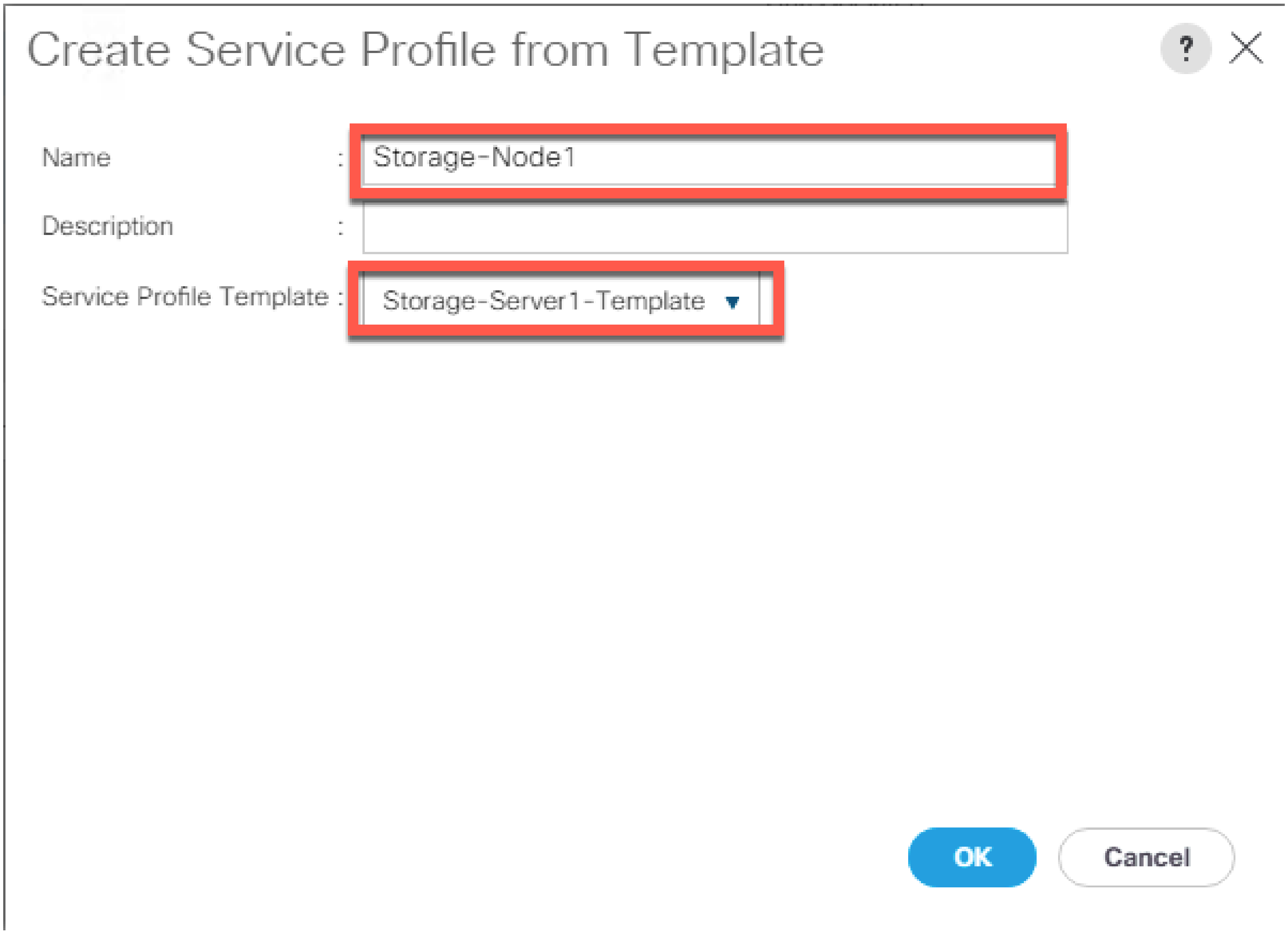
4. Select **Servers** from the left pane of the Cisco UCS Manager GUI.

5. Go to Servers > Service Profiles and right-click Create Service Profile from Template.

6. Type in **Storage-Node1** in the Name Prefix field.

7. Choose "**Storage-Server1-Template**" as the **Service Profile Template** you created before for the top node of the Cisco UCS S3260 Storage Server.

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



Create Service Profile from Template

Name : Storage-Node1

Description :

Service Profile Template : Storage-Server1-Template

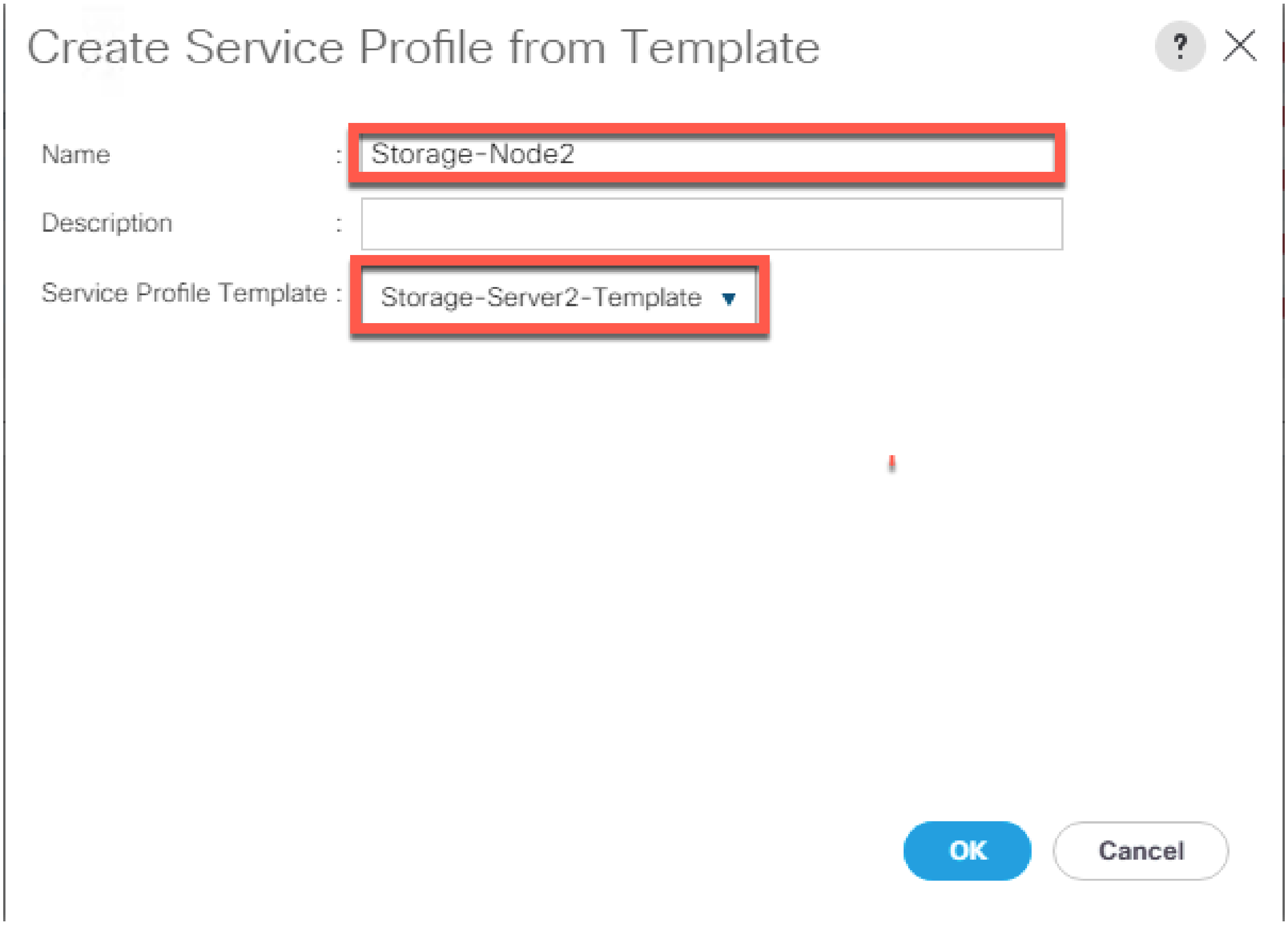
9. Repeat steps 1-5 to create Service Profiles for the remaining S3260 M5 server1 Nodes from the Template that belongs to top Node "Storage-Server1-Template". Make sure you name it as "Storage-Node3, Storage-Node5, Storage-Node7,Storage-Node9,Storage-Node11" respectively.

10. For the remaining M5 nodes, again Navigate to Servers > Service Profiles and right-click Create Service Profile from Template.

11. Type in **Storage-Node2** in the Name Prefix field.

12. Choose "**Storage-Server2-Template**" as the **Service Profile Template** you created before for the top node of the Cisco UCS S3260 Storage Server.

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



Create Service Profile from Template

Name : Storage-Node2

Description :

Service Profile Template : Storage-Server2-Template

14. Repeat steps 1-13 to create Service Profiles for the remaining S3260 M5 server Bottom Nodes from the Template that belongs to bottom Node "Storage-Server2-Template". Make sure you name it as "Storage-Node4, Storage-Node6,Storage-Node8,Storag-Node10, Storage-Node12."

Associate a Service Profile for Cisco UCS S3260 M5 Server

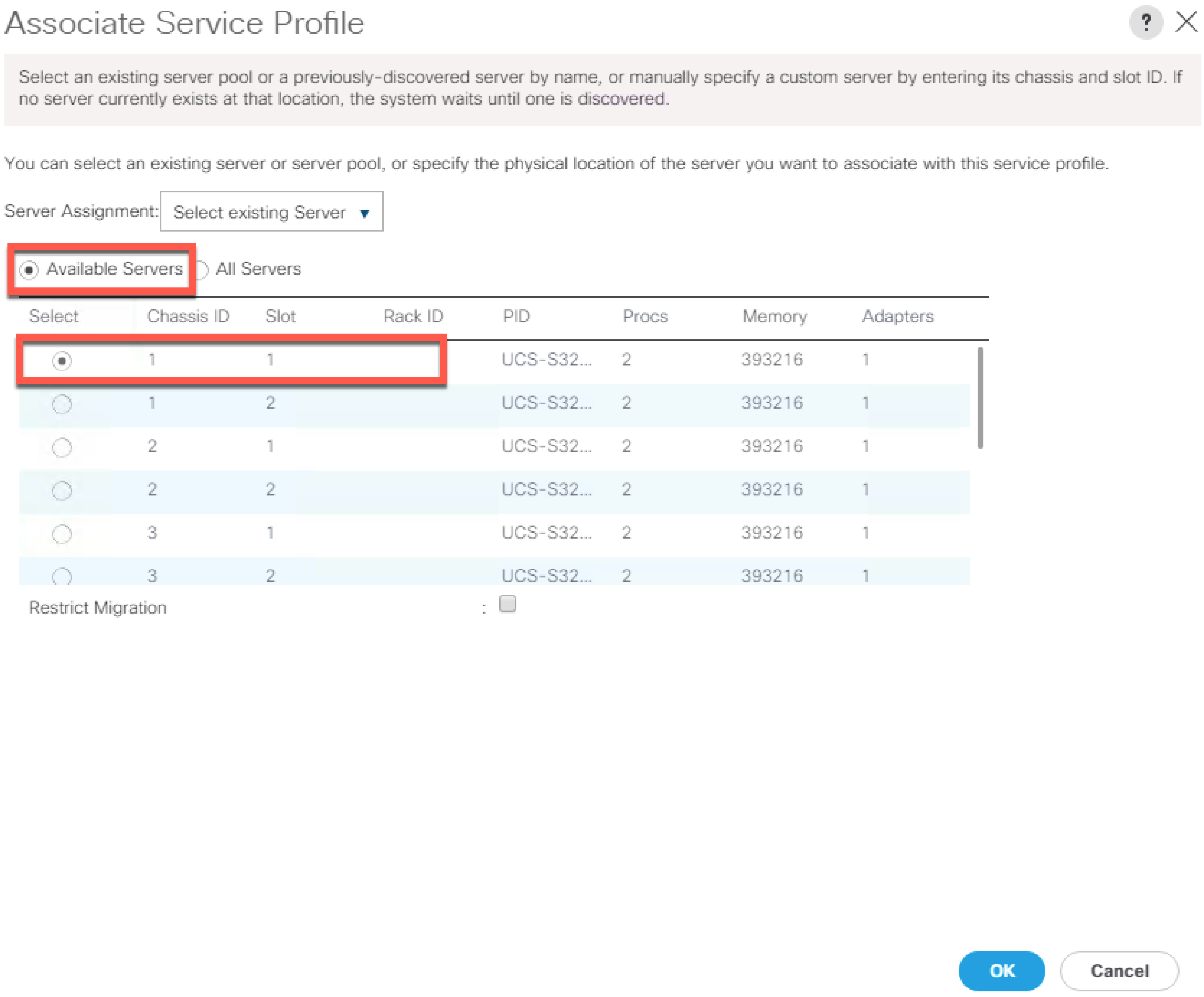
To associate all the "Storage-NodeX" Service Profiles to the Cisco UCS S3260 M5 Storage Servers, follow these steps:

1. Select **Servers** from the left pane of the Cisco UCS Manager GUI.
2. Go to Servers > Service Profiles and right-click "Storage-Node1" Service profile created previously.
3. Click "Change Server Profile Association."
4. From the Server Assignment drop-down list choose "Select Existing Server."

5. Click the radio button "Available Servers."

6. From the Chassis and Slot listed, choose Chassis1/Slot1 for Storage-Node1.

7. Click OK.



Associate Service Profile

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

You can select an existing server or server pool, or specify the physical location of the server you want to associate with this service profile.

Server Assignment: Select existing Server ▾

● Available Servers ○ All Servers

Select	Chassis ID	Slot	Rack ID	PID	Procs	Memory	Adapters
●	1	1		UCS-S32...	2	393216	1
○	1	2		UCS-S32...	2	393216	1
○	2	1		UCS-S32...	2	393216	1
○	2	2		UCS-S32...	2	393216	1
○	3	1		UCS-S32...	2	393216	1
○	3	2		UCS-S32...	2	393216	1

Restrict Migration : □

8. Repeat steps 1-7 to the Associate Remaining Service profiles "Storage-NodeX" for the Cisco UCS S3260 M5 storage server as listed in the table below.

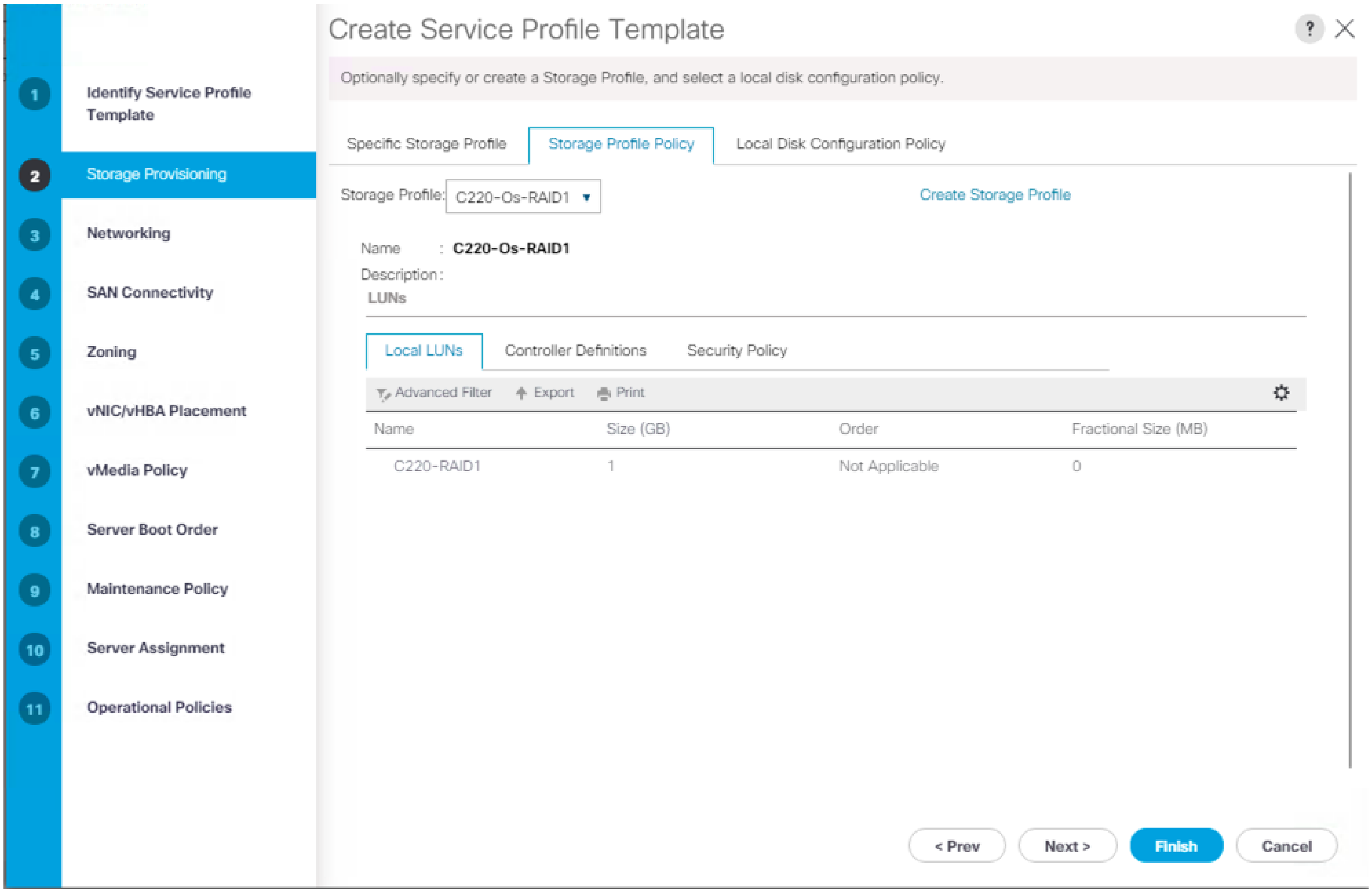
Service Profile Template	Service Profile	S3260 Chassis	Server Slot ID
Storage-Server1-Template	Storage-Node1	1	1

Storage-Server2 -Template	Storage-Node2	1	2
Storage-Server1-Template	Storage-Node3	2	1
Storage-Server2-Template	Storage-Node4	2	2
Storage-Server1-Template	Storage-Node5	3	1
Storage-Server2-Template	Storage-Node6	3	2
Storage-Server1-Template	Storage-Node7	4	1
Storage-Server2-Template	Storage-Node8	4	2
Storage-Server1-Template	Storage-Node9	5	1
Storage-Server2-Template	Storage-Node10	5	2
Storage-Server1-Template	Storage-Node11	6	1
Storage-Server2-Template	Storage-Node12	6	2

Create Service Profile Template for Cisco UCS C220 M5

The Service Profile for the Cisco UCS Rack-Mount Servers for SwiftStack Controllers (Active and Standby) is very similar to the previously created template for the Cisco UCS S3260. The only differences are with the Storage Profiles, Networking, and vNIC/vHBA Placement. To create the service profile template, follow these steps (the changes are listed below):

1. In the **`Storage Provisioning`** tab choose the appropriate Storage Profile for the Cisco UCS C220 M5 you created earlier.



Create Service Profile Template

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

Storage Profile: C220-Os-RAID1

Name : C220-Os-RAID1

Description :

LUNs

Name	Size (GB)	Order	Fractional Size (MB)
C220-RAID1	1	Not Applicable	0

The Networking tab will have only two Interfaces one for the nodes External communication and the other for Mgmt-network. In the setup used for the CVD, the Mgmt-network made available to the controller though this is not necessary, if everything is on External network

The vNIC/vHBA Placement tab will list the first interface as external and the second interface as the Mgmt/pxe-Network network.

2. Go through the remaining tabs similar to Cisco UCS S3260 by adding Maintenance Policies and complete the creation of Service Profile Template C220.

Summary

There will be three service profile templates, one for each server of the Cisco UCS S3260 and another for SwiftStack Controllers. In case of a single node configuration, you should have only two Service Profile templates; one for Cisco UCS S3260 and the other for Controller. If there are other blade or rack-mounted servers that you would like to connect to these SwiftStack servers acting as clients, create the necessary Service Profile templates and profiles.

Create Service Profile for Cisco UCS C220 M5 SwiftStack Controllers

To create a Service Profile, follow these steps:

1. Select **`Servers`** from the left pane of the UCSM GUI.
2. Go to Servers > Service Profiles and right-click Create Service Profiles from Template.
3. Add the name, starting prefix and Number of Instances as 2.

Summary

- You will have 6 Service Profiles one for each Chassis in a Single Node Configuration.
- You will have 12 Service Profiles two for each Chassis in a dual Node Configuration.
- You will have 2 Service Profiles for the SwiftStack controllers, one for Active and the other for Standby.
- After the successful creation of the "SwiftStack Controller" Service profile, select the Cisco UCS C220 M5 server for association and servers will start the Service profile association.

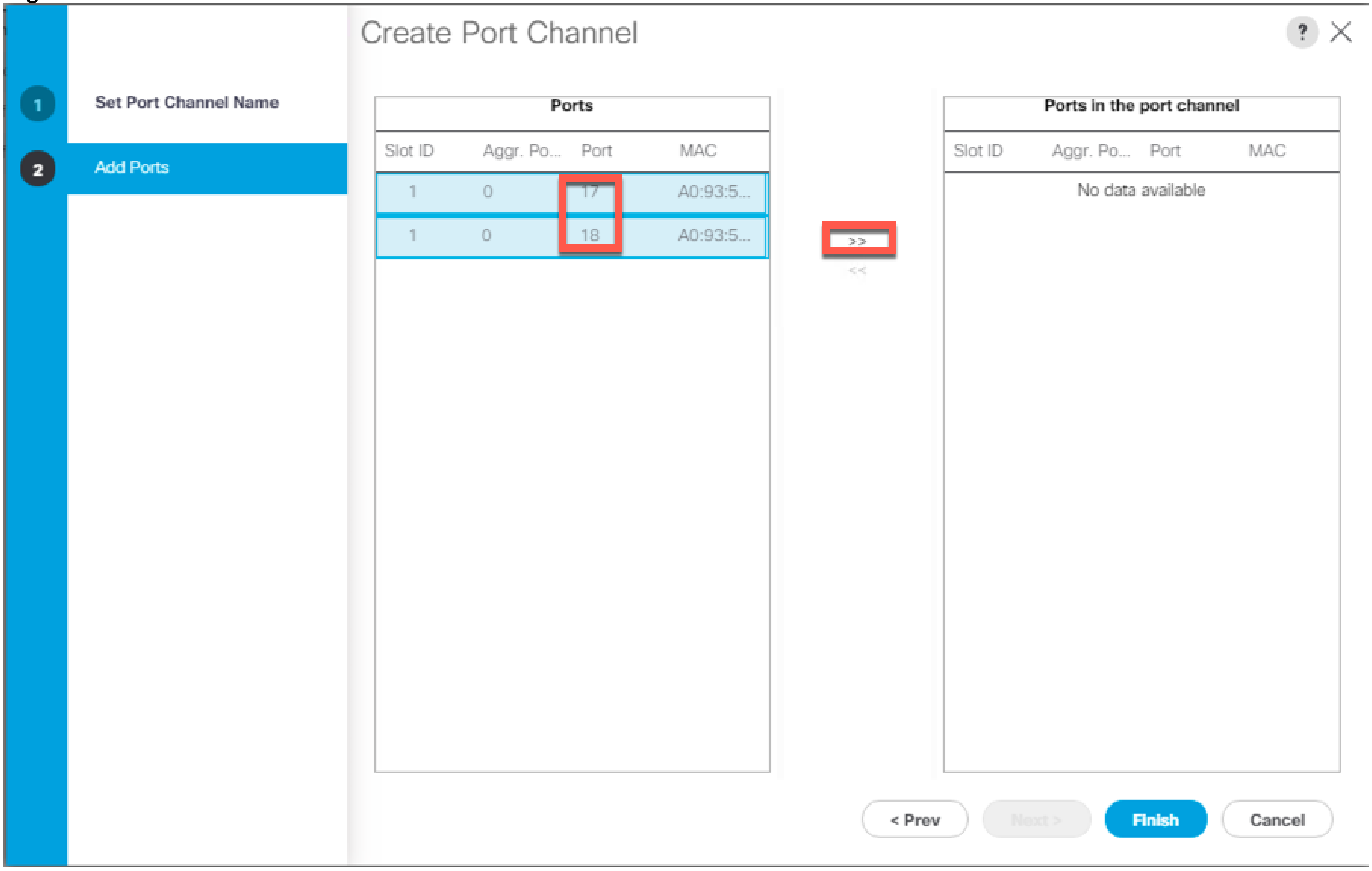
Create Port Channel for Network Uplinks

Create Port Channel for Fabric Interconnect A/B

To create Port Channels to the connected Nexus 9332PQ switches, follow these 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 17`**.
4. Type in **`vPC17`** in the `Name` field.
5. Click Next.
6. Select the available ports on the left **`17-18`**and assign them with >> to **`Ports in the Port Channel`**.
7. The "Add Ports" window will prompt you to confirm the selection, click Yes.

Figure 41 Create Port Channel



Ports

Slot ID	Aggr. Po...	Port	MAC
1	0	17	A0:93:5...
1	0	18	A0:93:5...

Ports in the port channel

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

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

9. Repeat steps 1-8 for Fabric B under LAN > LAN Cloud > Fabric B > Port Channels and right-click Create Port Channel.

10. Type in **ID 18**.

11. Type in **vPC18** name in the Name field.

12. Click **Next**.

13. Select the available ports on the left **17-18** and assign them with >> to **Ports in the Port Channel**.

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

The formal setup of the Cisco UCS Manager environment and both Cisco Nexus 9332PQ switches is finished and next is the installation of the Red Hat Enterprise Linux 7.5 Operating System.

Post Cisco UCS Configuration Health Checks

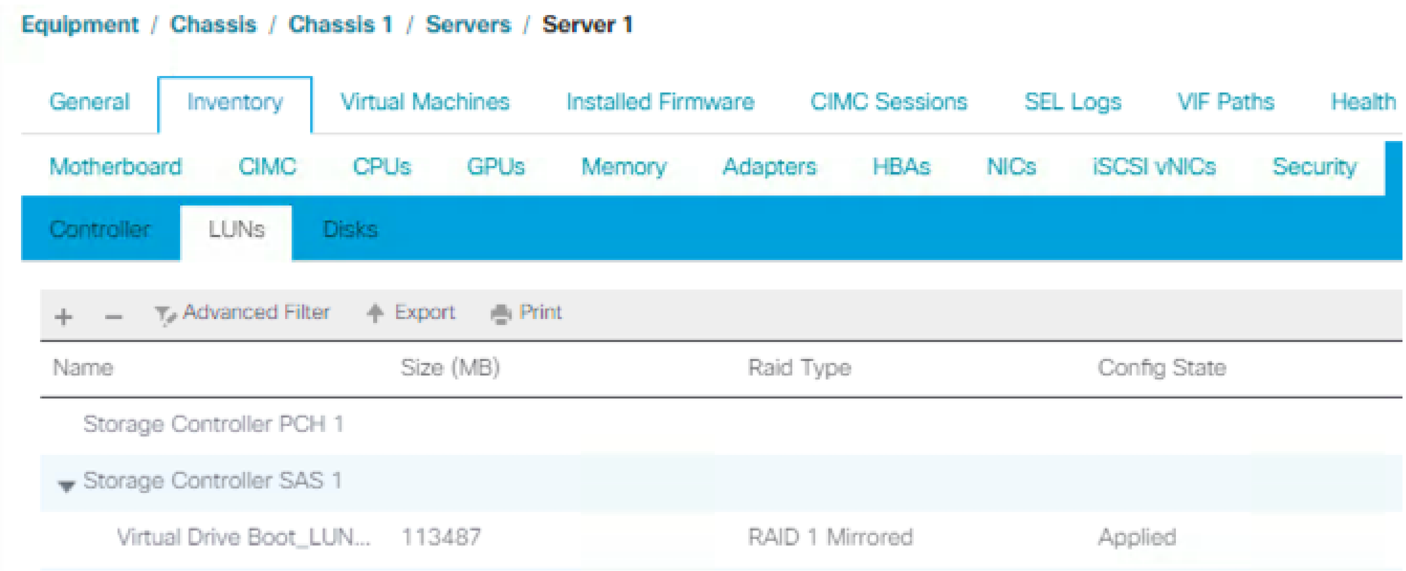
The following is a list of health checks to be completed before Installing Operating System and SwiftStack software.

1. Go to Equipment Tab, Fabric Interconnects, Ethernet ports and check the status of Server and Network Ports. Alternatively open a putty session to Fabric Interconnect, connect nxos a (or b) and check the status of ports with show interface brief.

2. Repeat this step by logging into Nexus Switches and check the status of ports and port-channels if any.

3. For each server node, the S3260 Storage Server Node(s) or the SwiftStack Controller check the Inventory tab for CPU and Memory details and Critical Faults.

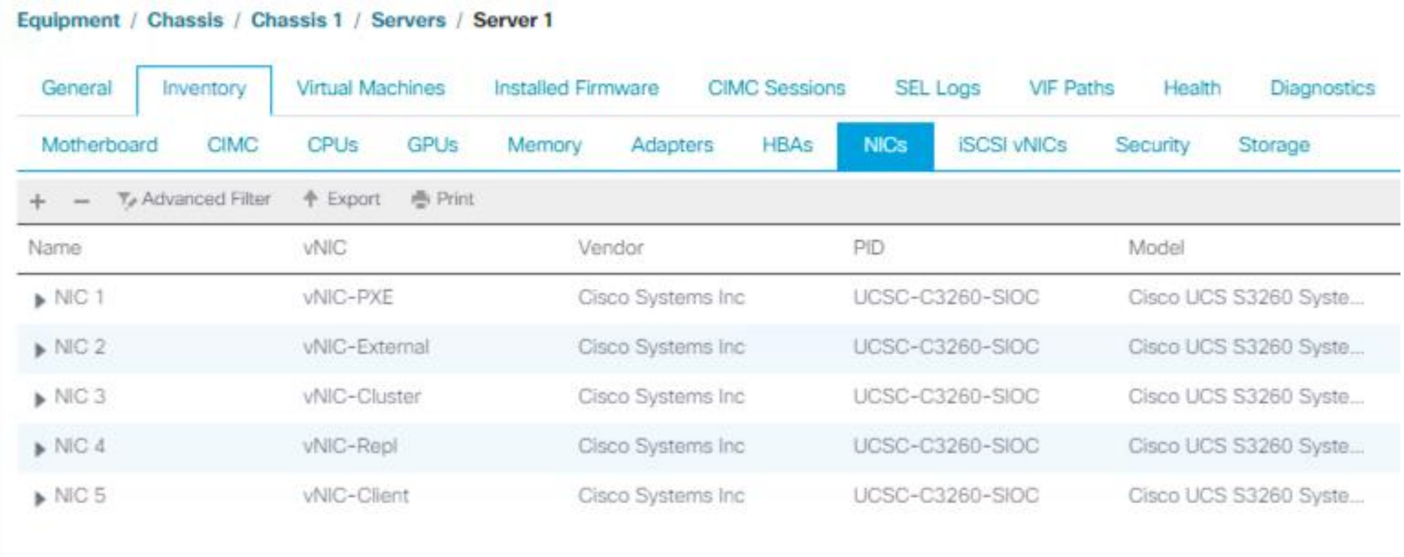
4. Under the same Inventory tab for the server under storage tab and then in Luns check the status of the boot lun. After successful application of the service profile, this will display as RAID1 Mirrored and in applied status as shown below.



Name	Size (MB)	Raid Type	Config State
Storage Controller PCH 1			
Storage Controller SAS 1			
Virtual Drive Boot_LUN...	113487	RAID 1 Mirrored	Applied

5. Under the same Inventory tab and in Disks, make sure that other SSD and HDD disks are available. If installing OS manually through Red Hat UI, you will select the Virtual Drive as shown in step 4 above and the rest if any jbods can be left as is. If automated through pxe install the other HDD and SSD's should be made as unconfigured too.

6. Under the same inventory tab check the number of NIC's are same as desired for that server. There should be a minimum of 4 NIC's for Storage Nodes.



Name	vNIC	Vendor	PID	Model
▶ NIC 1	vNIC-PXE	Cisco Systems Inc	UCSC-C3260-SIOC	Cisco UCS S3260 Syste...
▶ NIC 2	vNIC-External	Cisco Systems Inc	UCSC-C3260-SIOC	Cisco UCS S3260 Syste...
▶ NIC 3	vNIC-Cluster	Cisco Systems Inc	UCSC-C3260-SIOC	Cisco UCS S3260 Syste...
▶ NIC 4	vNIC-Repl	Cisco Systems Inc	UCSC-C3260-SIOC	Cisco UCS S3260 Syste...
▶ NIC 5	vNIC-Client	Cisco Systems Inc	UCSC-C3260-SIOC	Cisco UCS S3260 Syste...

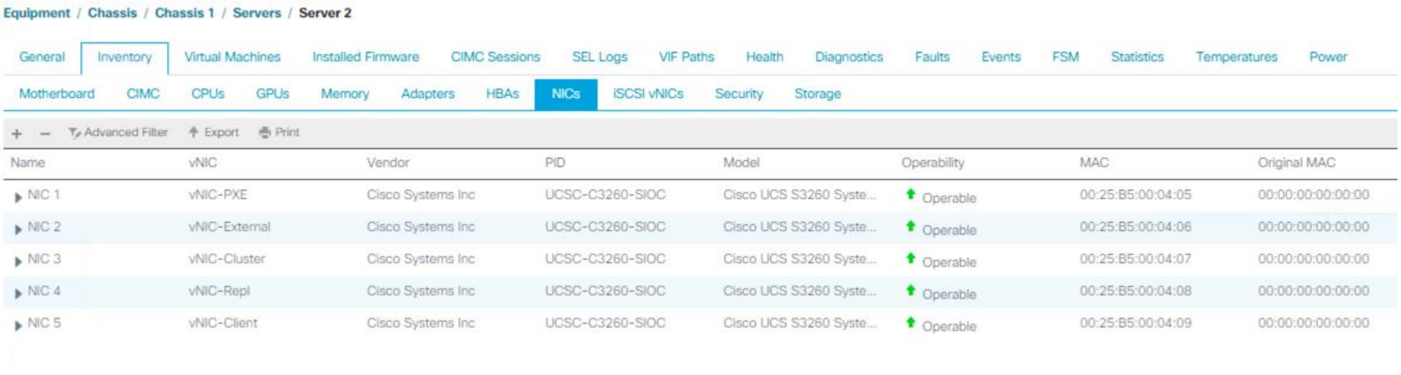
Install the Operating System on the SwiftStack Nodes

Install Operating System

This section describes the Installation of the Operating System on the nodes. The installation procedure is the same for both the Controller and Server Nodes. The operating system can be installed either manually through the Red Hat Installer or through the kick-start installer. Make sure to have the appropriate MAC address for the interfaces and have the MAC addresses from Cisco UCS.

To install the operating system, follow these steps:

1. From Cisco UCS Manager, navigate to Server > Inventory and NICS to get the MAC address details for the servers.



Name	vNIC	Vendor	PID	Model	Operability	MAC	Original MAC
NIC 1	vNIC-PXE	Cisco Systems Inc	UCSC-C3260-SIOC	Cisco UCS S3260 Syste...	Operable	00:25:B5:00:04:05	00:00:00:00:00:00
NIC 2	vNIC-External	Cisco Systems Inc	UCSC-C3260-SIOC	Cisco UCS S3260 Syste...	Operable	00:25:B5:00:04:06	00:00:00:00:00:00
NIC 3	vNIC-Cluster	Cisco Systems Inc	UCSC-C3260-SIOC	Cisco UCS S3260 Syste...	Operable	00:25:B5:00:04:07	00:00:00:00:00:00
NIC 4	vNIC-Repl	Cisco Systems Inc	UCSC-C3260-SIOC	Cisco UCS S3260 Syste...	Operable	00:25:B5:00:04:08	00:00:00:00:00:00
NIC 5	vNIC-Client	Cisco Systems Inc	UCSC-C3260-SIOC	Cisco UCS S3260 Syste...	Operable	00:25:B5:00:04:09	00:00:00:00:00:00

2. Log into Cisco UCS Manager GUI.

Install Red Hat Enterprise Linux 7.5 Operating System

This section provides the detailed procedures to install Red Hat Enterprise Linux 7.5 on Cisco UCS C220 M5 and Cisco UCS S3260 Storage Server. The installation uses the KVM console and virtual Media from Cisco UCS Manager.

This requires the RHEL 7.5 DVD/ISO media for the installation.

Install RHEL 7.5 on Cisco UCS C220 M5 and Cisco UCS S3260 M5 Server

To install Red Hat Linux 7.5 operating system on Cisco UCS C220 M5, follow these 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 (Controller) and right-click KVM Console.

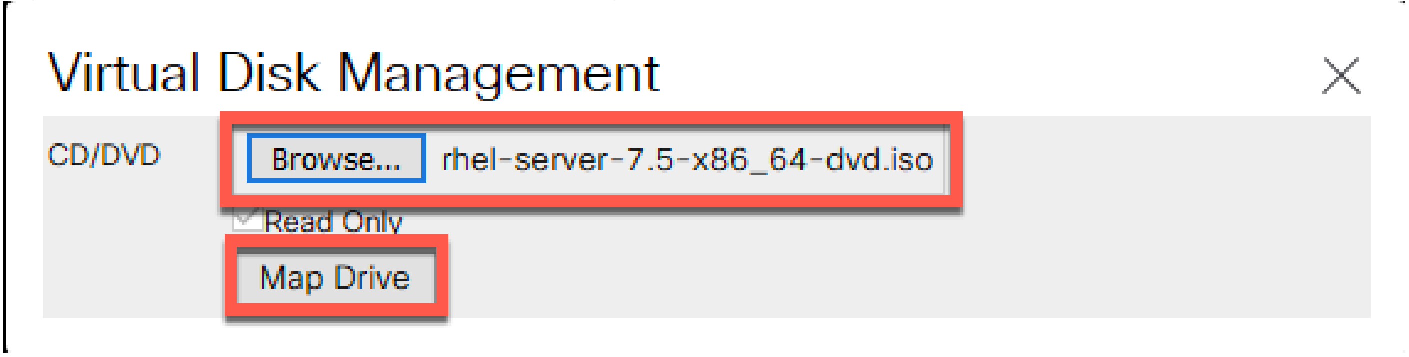
3. Launch KVM Console.

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

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

6. Click Choose File and Browse to the Red Hat Enterprise Linux 7.5 installation ISO image and select then click "**Map Drive.**"

Figure 42 Red Hat Enterprise Linux 7.5 ISO image



rhel-server-7.5-x86_64-dvd.iso

☑ Read Only

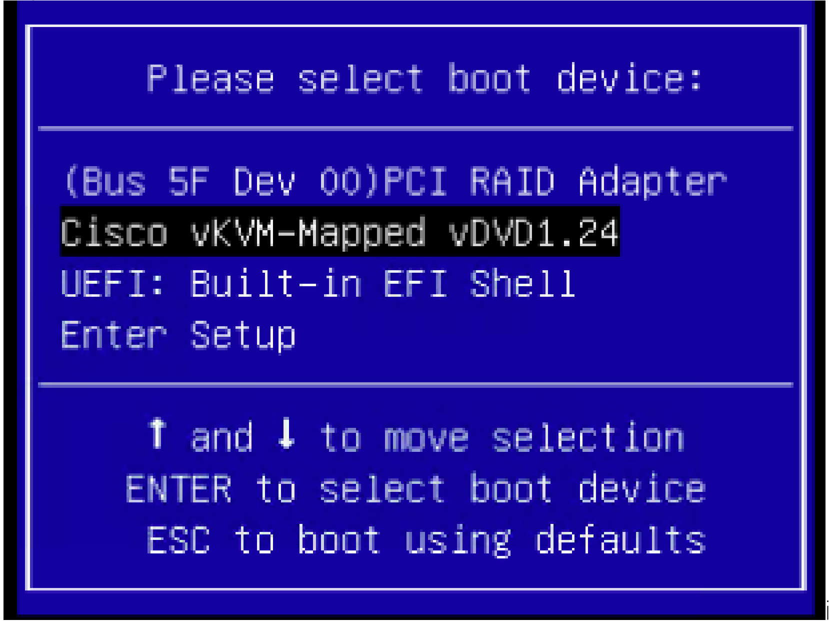
7. In the KVM window, select the **Macros > Static Macros > Ctrl-Alt-Del** button in the upper left corner.

8. Click **OK** and then 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 vDVD1.24**"

Figure 43 Boot Menu Selection



```
   Please select boot device:

(Bus 5F Dev 00)PCI RAID Adapter
Cisco vKVM-Mapped vDVD1.24
UEFI: Built-in EFI Shell
Enter Setup

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

in

11. When the Red Hat Enterprise Linux 7.5 installer appears, press the Tab button for further configuration options.

12. At the prompt type:

```
inst.ks=ftp://192.168.100.220/storage-node1.cfg  net.ifnames=0  biosdevname=0
ip=192.168.125.160::192.168.100.1:255.255.255.0:storage-node1:eth1:none
```

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 Kickstart file for the Cisco UCS C220 M5 server for SwiftStack-Controller is in Appendix A. This Kickstart file for the Cisco UCS S3260 M5 Server for Storage Nodes is in Appendix B.

13. Repeat steps 1-12 to install RHEL7.5 on all the UCS S3260 M5 storage servers.

Install SwiftStack Software

Pre-Install Checks

Make sure that SSH Trust is enabled amongst all the controller and storage nodes.

Here is an example of setting up the ssh-trust:

```
ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
…..

Use ssh-copy-id to copy the files.
ssh-copy-id root@swiftstack-server2
```

Establish trust between the nodes before proceeding with the Install.

Install On-Premise Controller Software

There are two modes of installation for SwiftStack controller. The controller can be hosted on SwiftStack site and can be installed via http://platform.swiftstack.com. You can also install Controller On-Premise and the current scope in this CVD is limited to On-Premise controller only. Please Contact SwiftStack for getting the On-Premise Controller software.

For complete details about the software, log into portal.swifstack.com and check for on-premise controller install under Admin section; only high-level install steps are presented.

Install Software

Obtain the Installer software and run the script in a linux shell as shown below:

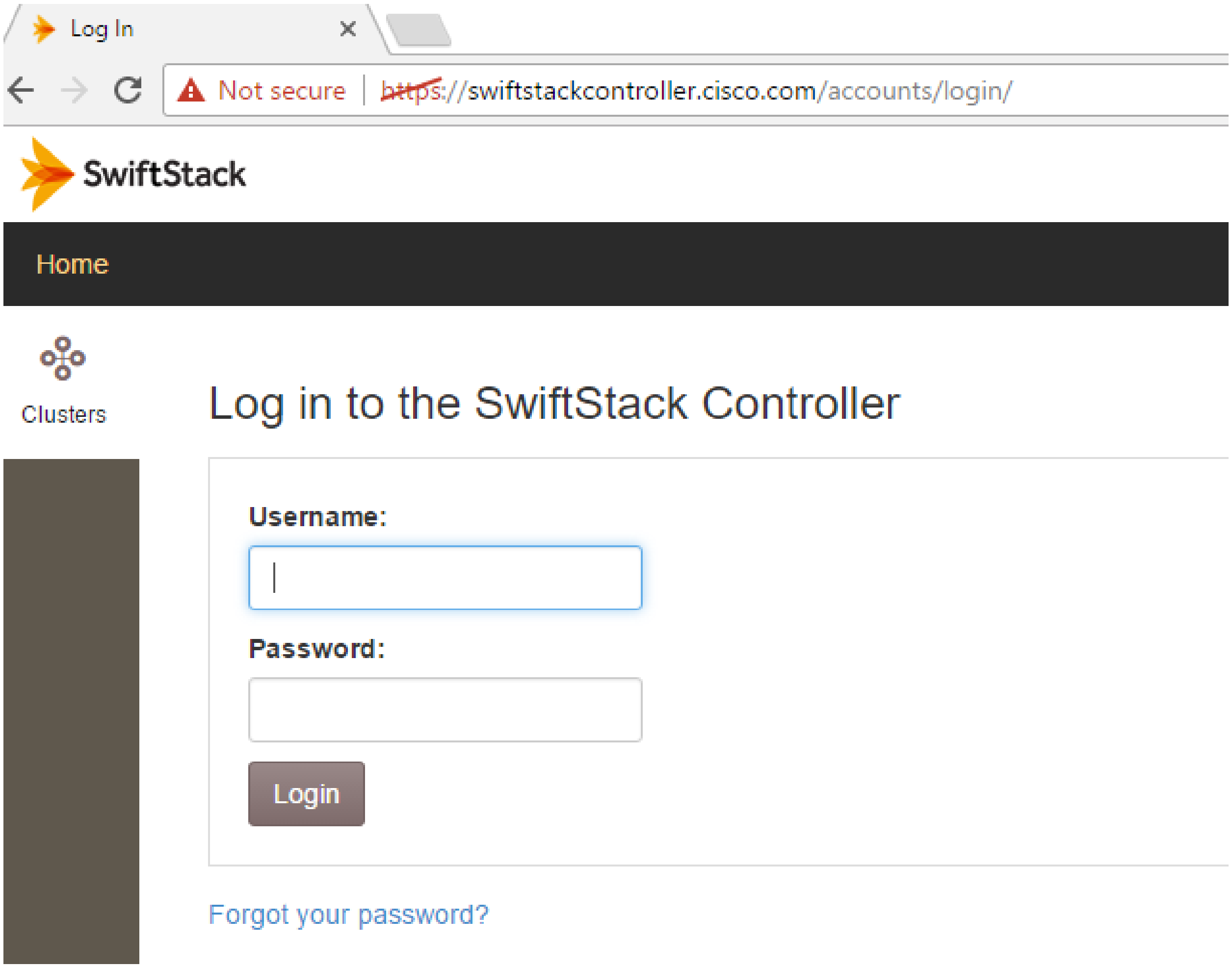
```
root@swiftcontroller    ~]#    ./SwiftStack-Controller-<swiftstack    version>-
installer.sh
Extracting       SwiftStack       Controller       <swiftstack       version>
............................................................................
997580 blocks

…..
…..
....

Installing SwiftStack Controller...
Preparing...
########################################
Updating / installing...
swiftstack-controller-<swiftstackversion>
########################################
**** SwiftStack Controller install details will be logged to /tmp/install.log
```

```
Controller install succeeded!

You must now complete setup by pointing a web browser at this server
using HTTPS and the default port of 443.  E.g.

    https://173.36.219.123/

Log in with the username "localadmin" and the default password, "password"
```

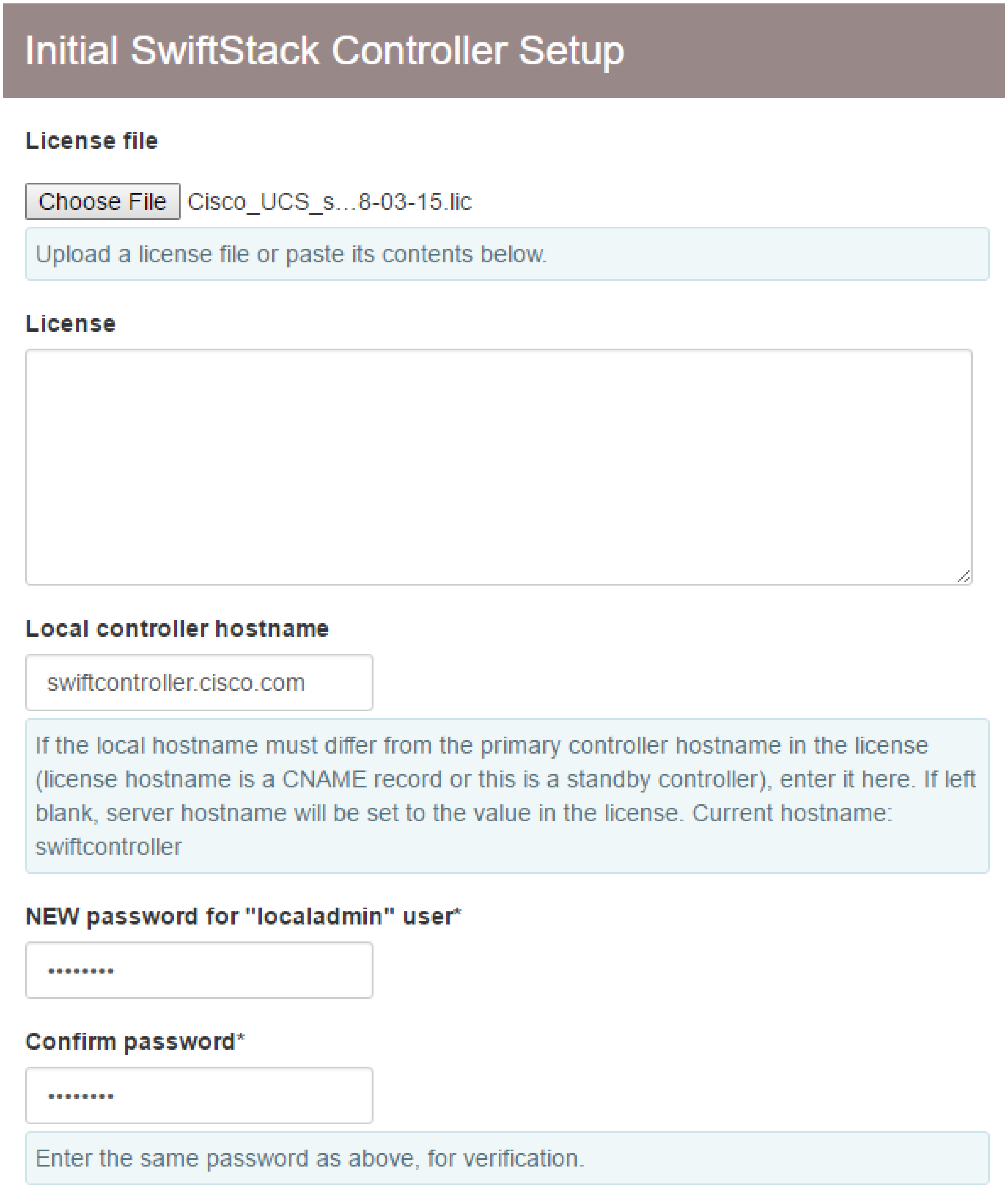
Post Software Install Configuration

To configure the installation, follow these steps:

1. Log into the URL pointed out after the install, with user as localadmin and default password.


Log In
Not secure
~~https~~://swiftstackcontroller.cisco.com/accounts/login/
SwiftStack
Home
Clusters
Log in to the SwiftStack Controller
Username:
Password:
Login
Forgot your password?

2. Enter license key obtained from SwiftStack and enter the hostname and new password to proceed. You may leave the other values to default and click Submit.



Initial SwiftStack Controller Setup

License file

Choose File Cisco_UCS_s...8-03-15.lic

Upload a license file or paste its contents below.

License

Local controller hostname

swiftcontroller.cisco.com

If the local hostname must differ from the primary controller hostname in the license (license hostname is a CNAME record or this is a standby controller), enter it here. If left blank, server hostname will be set to the value in the license. Current hostname: swiftcontroller

NEW password for "localadmin" user*

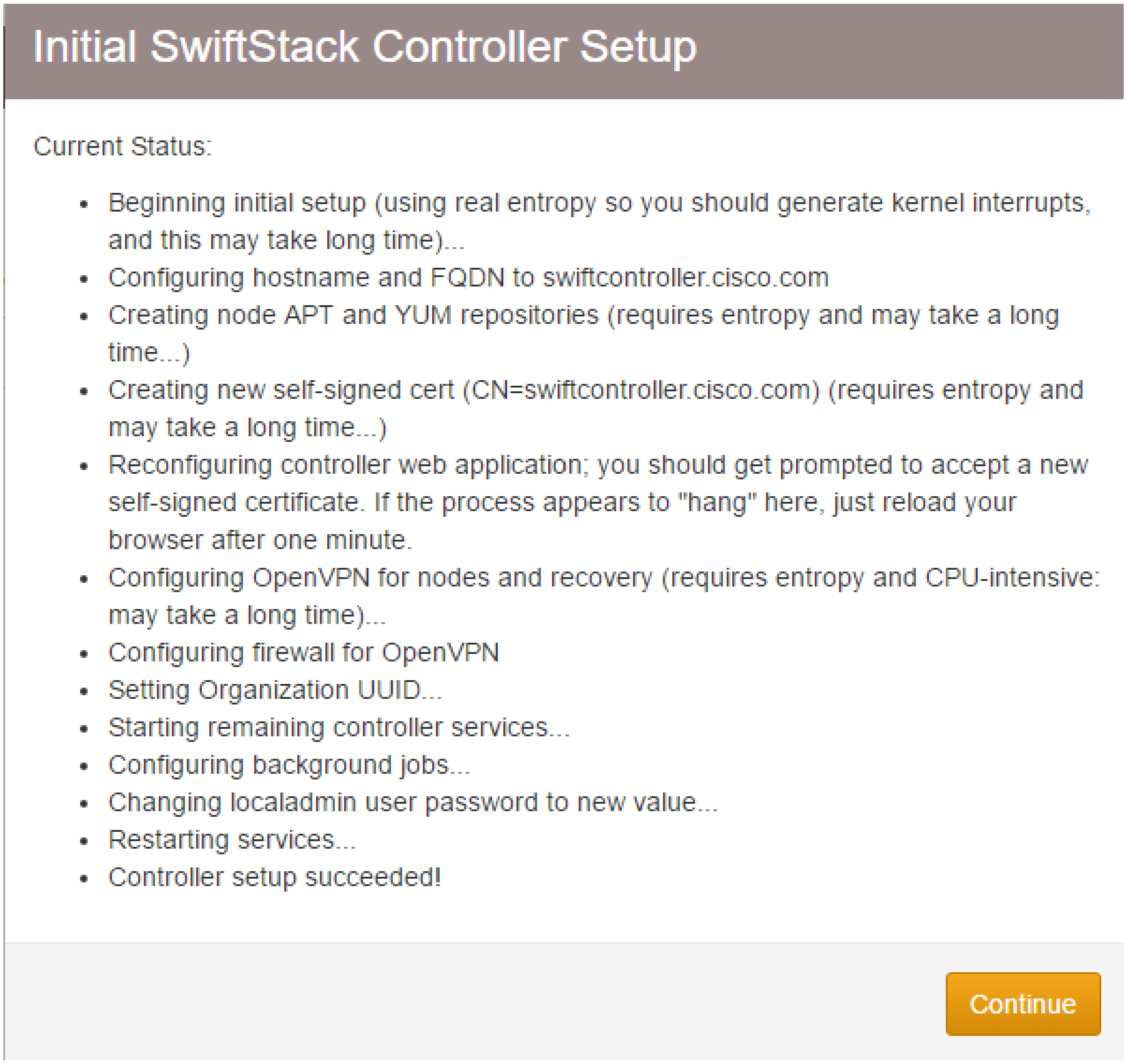
••••••••

Confirm password*

••••••••

Enter the same password as above, for verification.

It may take few minutes for the setup to complete.



Initial SwiftStack Controller Setup

Current Status:

- Beginning initial setup (using real entropy so you should generate kernel interrupts, and this may take long time)...
- Configuring hostname and FQDN to swiftcontroller.cisco.com
- Creating node APT and YUM repositories (requires entropy and may take a long time...)
- Creating new self-signed cert (CN=swiftcontroller.cisco.com) (requires entropy and may take a long time...)
- Reconfiguring controller web application; you should get prompted to accept a new self-signed certificate. If the process appears to "hang" here, just reload your browser after one minute.
- Configuring OpenVPN for nodes and recovery (requires entropy and CPU-intensive: may take a long time)...
- Configuring firewall for OpenVPN
- Setting Organization UUID...
- Starting remaining controller services...
- Configuring background jobs...
- Changing localadmin user password to new value...
- Restarting services...
- Controller setup succeeded!

After logging in with the new password, the system will prompt to add the nodes and create the cluster. It may come up something like below to run the curl command on the storage nodes.

```
curl https://swiftcontroller.cisco.com:443/install | sudo bash
```

Install SwiftStack Software on Storage Nodes

To install the SwiftStack software on the storage nodes, follow these steps:

1. Before running the installer software make sure that the SSL certificates are installed. They could either be from commercial CA or Self-Signed.

A Self-Signed certificate will be generated on the controller node and needs to be copied to all the server nodes.

2. On Controller Node:

```
[root@swiftcontroller ~]# cd /opt/ss/etc/
[root@swiftcontroller etc]# ls -l ssman.crt
-rw-r--r-- 1 root 668 Jun  6 11:16 ssman.crt
```

3. Copy this certificate to all Server nodes:

```
scp ssman.crt root@swiftstack-node:/etc/pki/ca-trust/source/anchors/
```

4. Once copied run update-ca-trust extract as root user on storage node:

```
[root@swiftstack-server28-2 .ssh]# update-ca-trust extract
[root@swiftstack-server28-2 .ssh]#
```

5. Run the Curl command on the storage node:

```
curl https://swiftcontroller.cisco.com:443/install | sudo bash
```

6. After completion of this command, the system will print the claim URL. This can also be printed by running ssclaimurl command on the server as shown below:

```
 [root@swiftstack-server2 ~]# ssclaimurl
+---------------------------------------------------------------------------+
|                                                                           |
| Your claim URL is:                                                        |
| https://swiftcontroller.cisco.com:443/claim/9723dbc2-61e1-11e7-8fcb-0025b500024f |
|                                                                           |
+---------------------------------------------------------------------------+
```

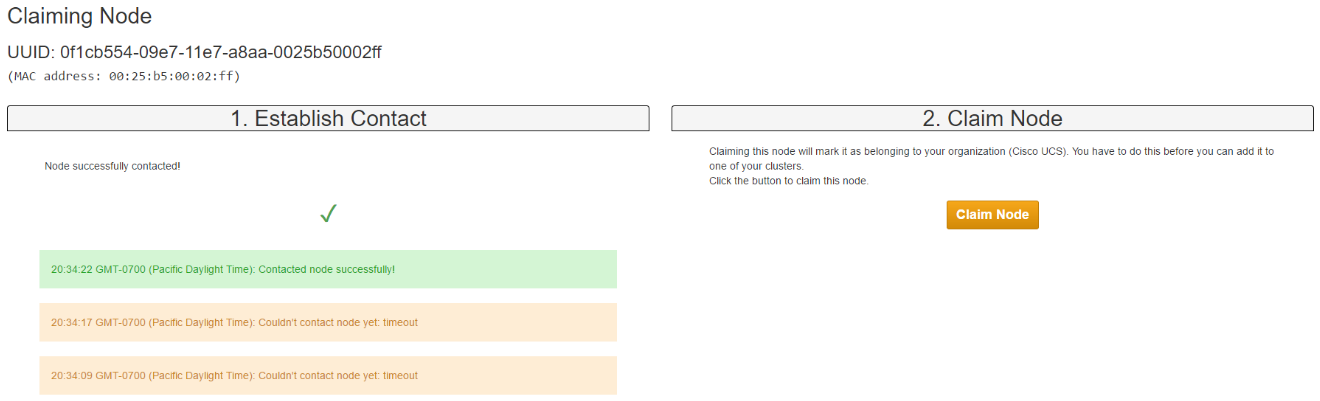
7. Run the Curl Command on each storage node and get the ssclaimurl's. The nodes have to be claimed through http request to the controller.

Configure SwiftStack Controller for Nodes

To configure the SwiftStack Controller for nodes, follow these steps:

1. Claim Nodes:

   a. Run the ClaimURL in a browser to claim the nodes.



Claiming Node

UUID: 0f1cb554-09e7-11e7-a8aa-0025b50002ff

(MAC address: 00:25:b5:00:02:ff)

1. Establish Contact

Node successfully contacted!

✓

20:34:22 GMT-0700 (Pacific Daylight Time): Contacted node successfully!

20:34:17 GMT-0700 (Pacific Daylight Time): Couldn't contact node yet: timeout

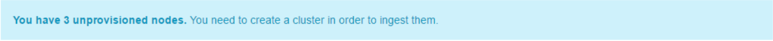
20:34:09 GMT-0700 (Pacific Daylight Time): Couldn't contact node yet: timeout

2. Claim Node

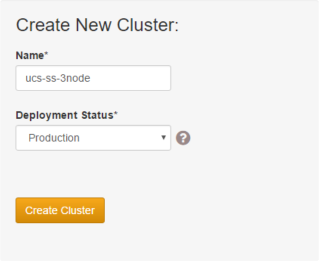
Claiming this node will mark it as belonging to your organization (Cisco UCS). You have to do this before you can add it to one of your clusters.
Click the button to claim this node.

2. Create and Configure the Cluster:

a. Enter the cluster name and create the cluster.



You have 3 unprovisioned nodes. You need to create a cluster in order to ingest them.



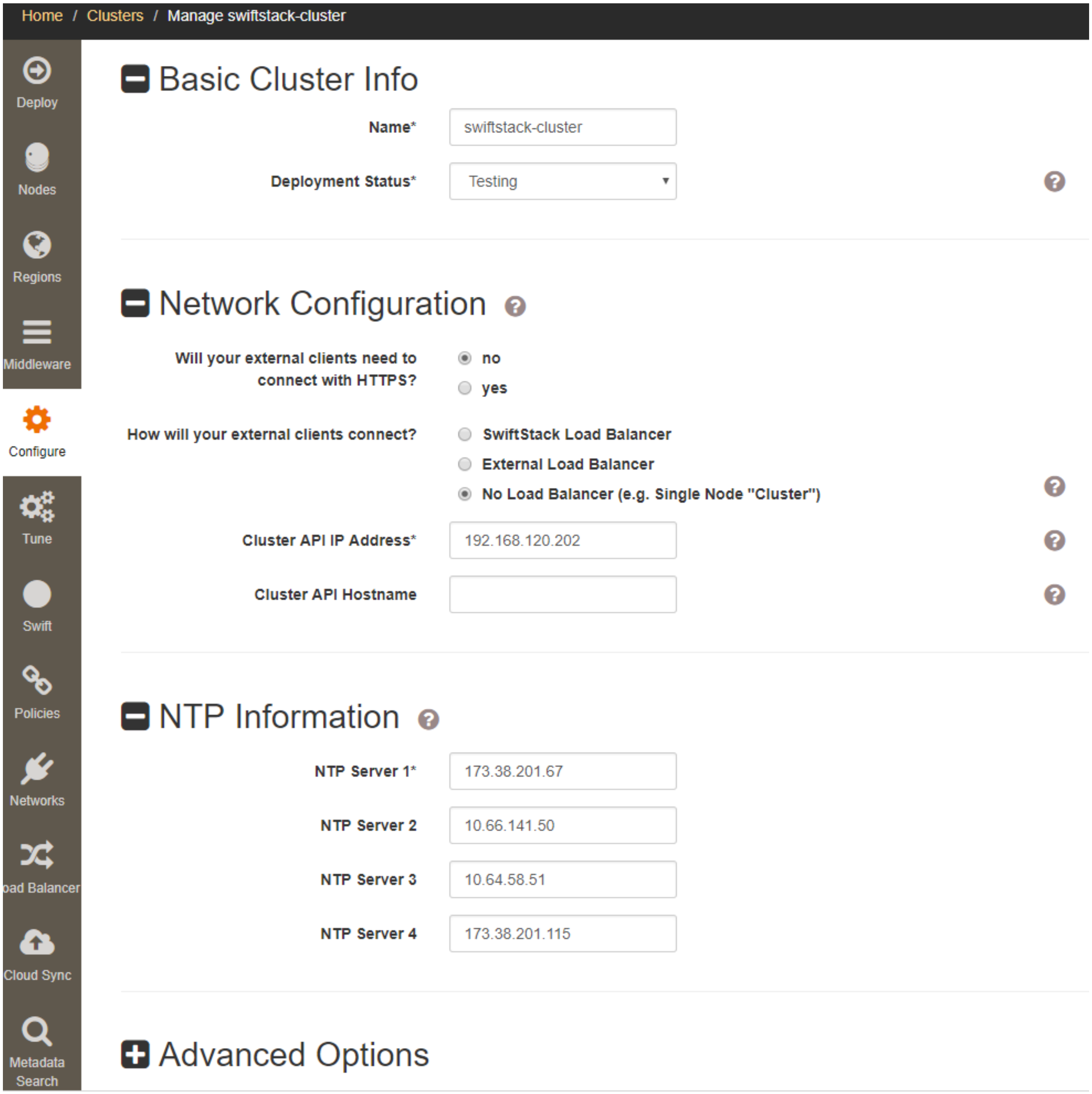
Create New Cluster:

Name*: ucs-ss-3node

Deployment Status*: Production

3. Configure the basic settings:

a. Click Configure and configure the basic settings as shown and submit the changes.



Basic Cluster Info

Name*: swiftstack-cluster

Deployment Status*: Testing

Network Configuration

Will your external clients need to connect with HTTPS? ● no ○ yes

How will your external clients connect?
○ SwiftStack Load Balancer
○ External Load Balancer
● No Load Balancer (e.g. Single Node "Cluster")

Cluster API IP Address*: 192.168.120.202

Cluster API Hostname:

NTP Information

NTP Server 1*: 173.38.201.67

NTP Server 2: 10.66.141.50

NTP Server 3: 10.64.58.51

NTP Server 4: 173.38.201.115

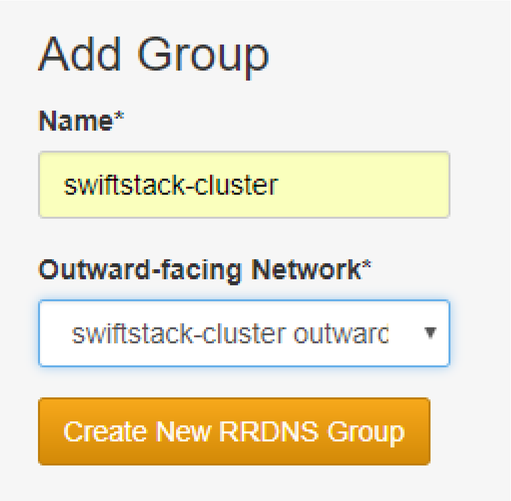
Advanced Options

4. In the above Network Configuration ‘No Load Balancer’ is used. If you would like to use a SwiftStack load balancer, enter the name of the Load Balancer configured. For information about configuring the SwiftStack Load Balancer, please refer to the SwiftStack documentation.

A snippet is provided below:

a. Click Home > Clusters > Manage <name of the Cluster>.

b. Click Load Balancer and Add Group name and the outward facing network.



Add Group

Name*

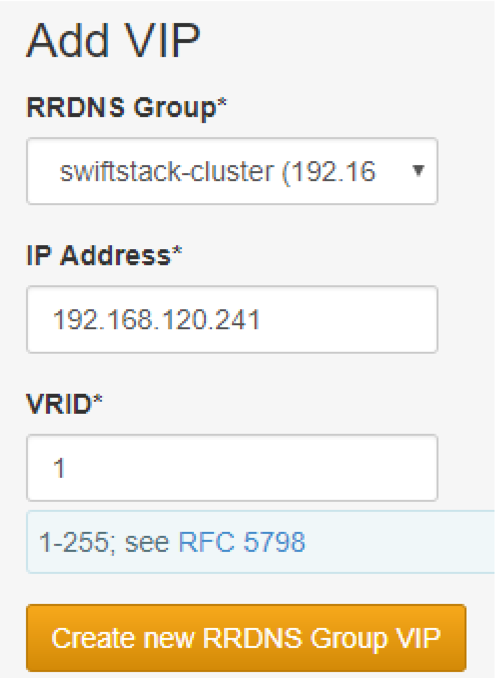
swiftstack-cluster

Outward-facing Network*

swiftstack-cluster outwarc ▾

5. Here the name is same as the cluster name configured in bind in your /var/named/<zone file>.

6. Click Create New RRDNS Group and provide a VIP with IP address again per your zone file and click Create new RRDNS group VIP.



Add VIP

RRDNS Group*

swiftstack-cluster (192.16 ▾

IP Address*

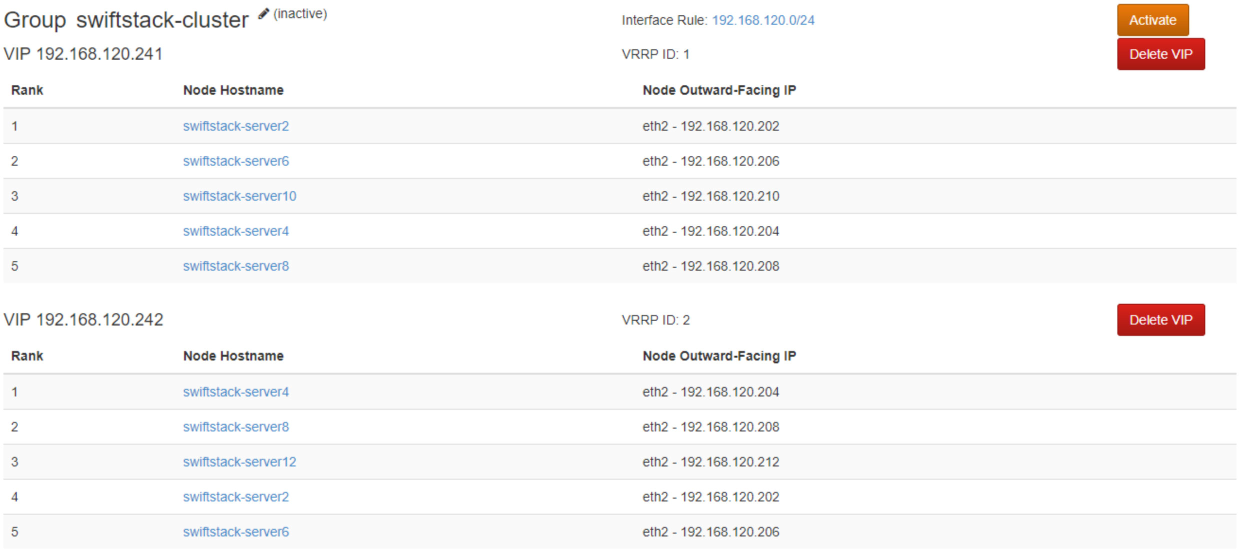
192.168.120.241

VRID*

1

1-255; see RFC 5798

7. Add another VIP and repeat for all the VIP's per your named zone file. This should create multiple entries as shown below.



Group swiftstack-cluster (inactive)

Interface Rule: 192.168.120.0/24

VIP 192.168.120.241

VRRP ID: 1

Rank	Node Hostname	Node Outward-Facing IP
1	swiftstack-server2	eth2 - 192.168.120.202
2	swiftstack-server6	eth2 - 192.168.120.206
3	swiftstack-server10	eth2 - 192.168.120.210
4	swiftstack-server4	eth2 - 192.168.120.204
5	swiftstack-server8	eth2 - 192.168.120.208

VIP 192.168.120.242

VRRP ID: 2

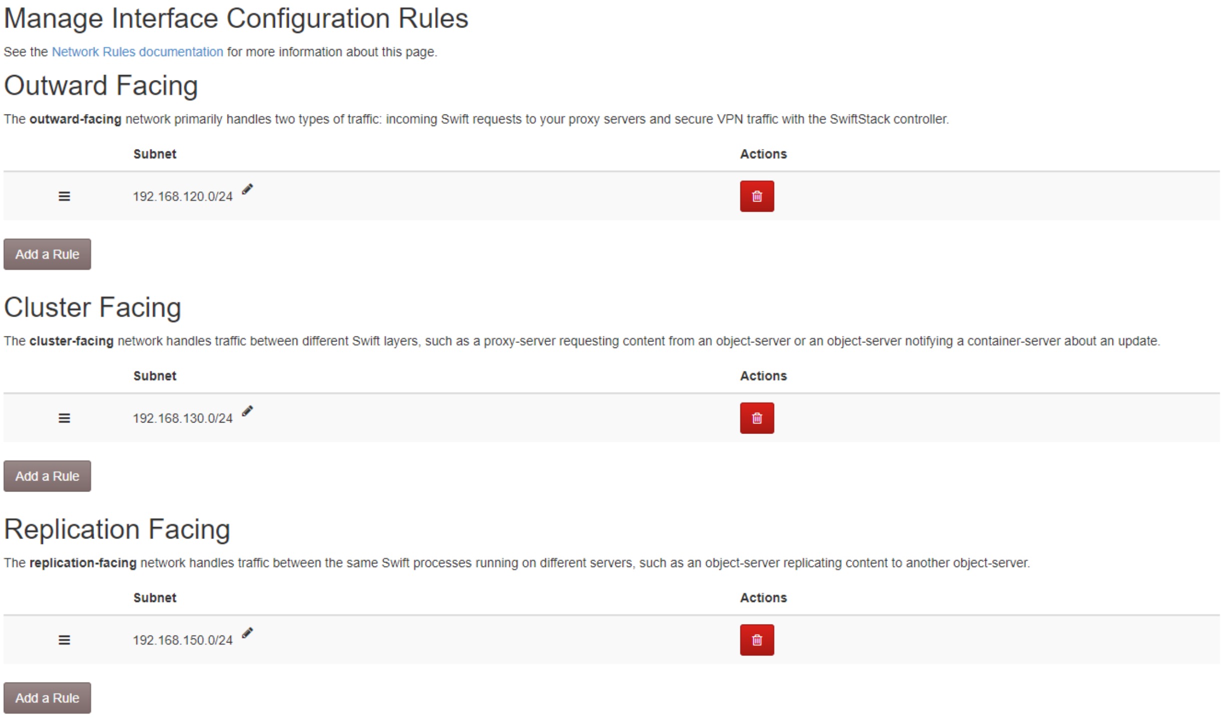
Rank	Node Hostname	Node Outward-Facing IP
1	swiftstack-server4	eth2 - 192.168.120.204
2	swiftstack-server8	eth2 - 192.168.120.208
3	swiftstack-server12	eth2 - 192.168.120.212
4	swiftstack-server2	eth2 - 192.168.120.202
5	swiftstack-server6	eth2 - 192.168.120.206

8. Repeat steps 1-5 for as many VIP's as nodes in your cluster and Activate the Group. In the network page make sure to enter the name of this group for the SwiftStack Load balancer.

9. Once configured and configuration deployed, this Load Balancer should be pingable from all the server nodes (after step 8 in this section).

10. Manage Interface Configuration Rules:

    a. Click Networks and provide the 3 network configuration and save them as rules. Every node ingested will inherit these rules by default.



Manage Interface Configuration Rules

See the Network Rules documentation for more information about this page.

Outward Facing

The **outward-facing** network primarily handles two types of traffic: incoming Swift requests to your proxy servers and secure VPN traffic with the SwiftStack controller.

	Subnet	Actions
≡	192.168.120.0/24	🗑

Cluster Facing

The **cluster-facing** network handles traffic between different Swift layers, such as a proxy-server requesting content from an object-server or an object-server notifying a container-server about an update.

	Subnet	Actions
≡	192.168.130.0/24	🗑

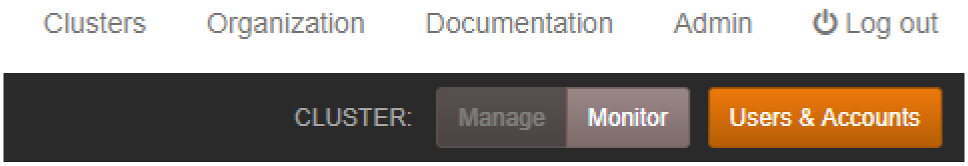
Replication Facing

The **replication-facing** network handles traffic between the same Swift processes running on different servers, such as an object-server replicating content to another object-server.

	Subnet	Actions
≡	192.168.150.0/24	🗑

11. Create User and Accounts:

a. Click User & Accounts to create users.



b. Create new user as needed. Create a backup user say bk with up Super user privileges.

12. Ingest the nodes:

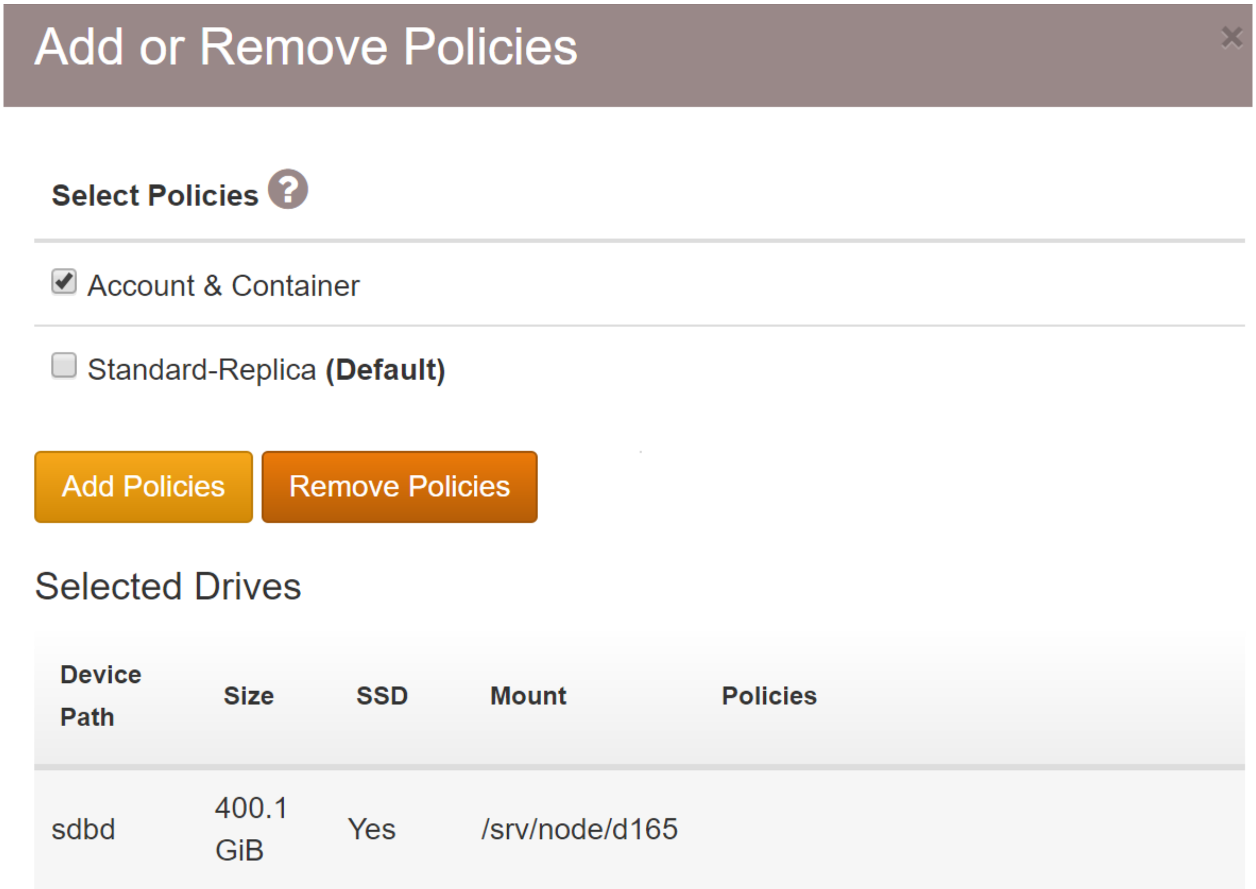
    a. Click Nodes and ingest the nodes. Confirm the networks when prompted by the system for outward facing, cluster and replication networks.

13. Setup the nodes:

    a. Click Setup of each node.

    b. Select all the disks and format the drives.

    c. Select <NVMe/SSD disks> (2 disks in single node and 1 disk in dual node configurations), click Add Policies and select Account and Container as shown below.



☑ Account & Container

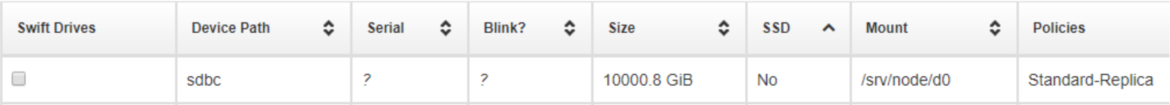
□ Standard-Replica **(Default)**

Selected Drives

Device Path	Size	SSD	Mount	Policies
sdbd	400.1 GiB	Yes	/srv/node/d165	

    d. Select HDD disks (54 disks in Single Node and 27 disks in dual node configurations), click Add Policies and select Standard-Replica.

e. Once Completed validate that the policies are displayed correctly on the respective disks as shown below.



Swift Drives	Device Path	Serial	Blink?	Size	SSD	Mount	Policies
□	sdbc	?	?	10000.8 GiB	No	/srv/node/d0	Standard-Replica

f. In the left menu, enable the node.

14. This completes the setup of one node. Repeat the procedure for all the storage nodes.

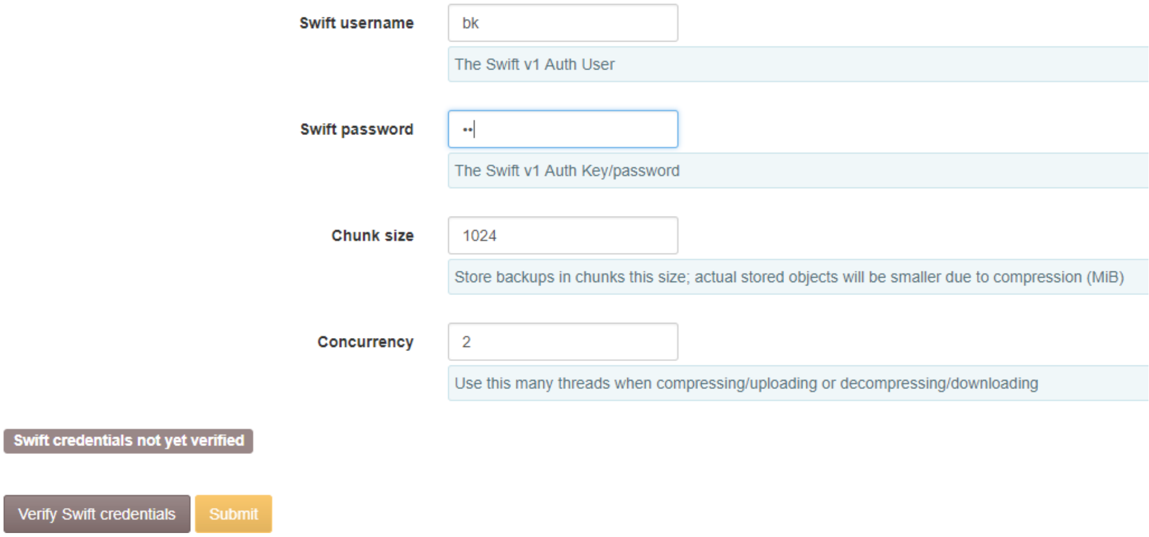
15. Deploy the Configuration:

a. Go to Home > Clusters > Deploy and click Deploy Config to SwiftStack Nodes.

b. Click the Admin Tab and update the default values as needed in your setup.

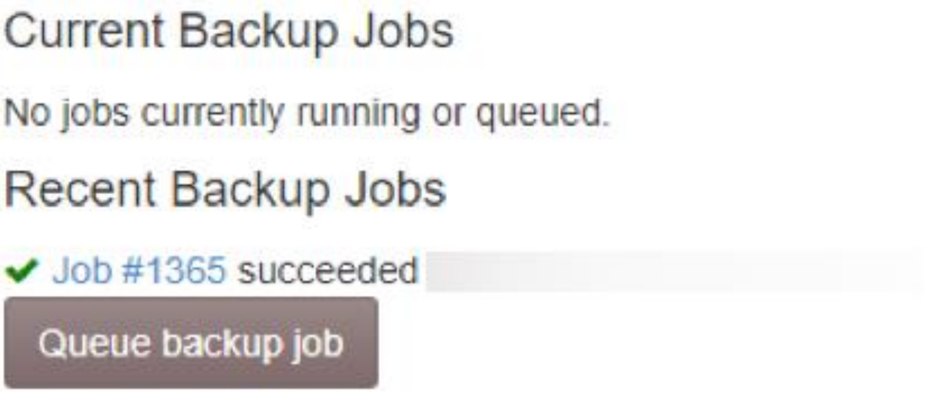
c. Click Backups in the Admin page.

d. Enable save backups to Swift, enter backup user and password created earlier and click Validate Swift Credentials and submit the Changes and then Queue a backup job.



Swift username	bk The Swift v1 Auth User
Swift password	•• The Swift v1 Auth Key/password
Chunk size	1024 Store backups in chunks this size; actual stored objects will be smaller due to compression (MiB)
Concurrency	2 Use this many threads when compressing/uploading or decompressing/downloading

Swift credentials not yet verified



Current Backup Jobs

No jobs currently running or queued.

Recent Backup Jobs

✓ Job #1365 succeeded

Configure Standby Controller for Nodes

Standby Controller is always in passive node. The primary and standby are not in Active/Active, but in Active/Passive mode. Hence the standby controller is configured and left as is. Refer to the section High Availability and Business Continuity to learn how to activate the Standby Controller in case of a failure of the Primary Controller.

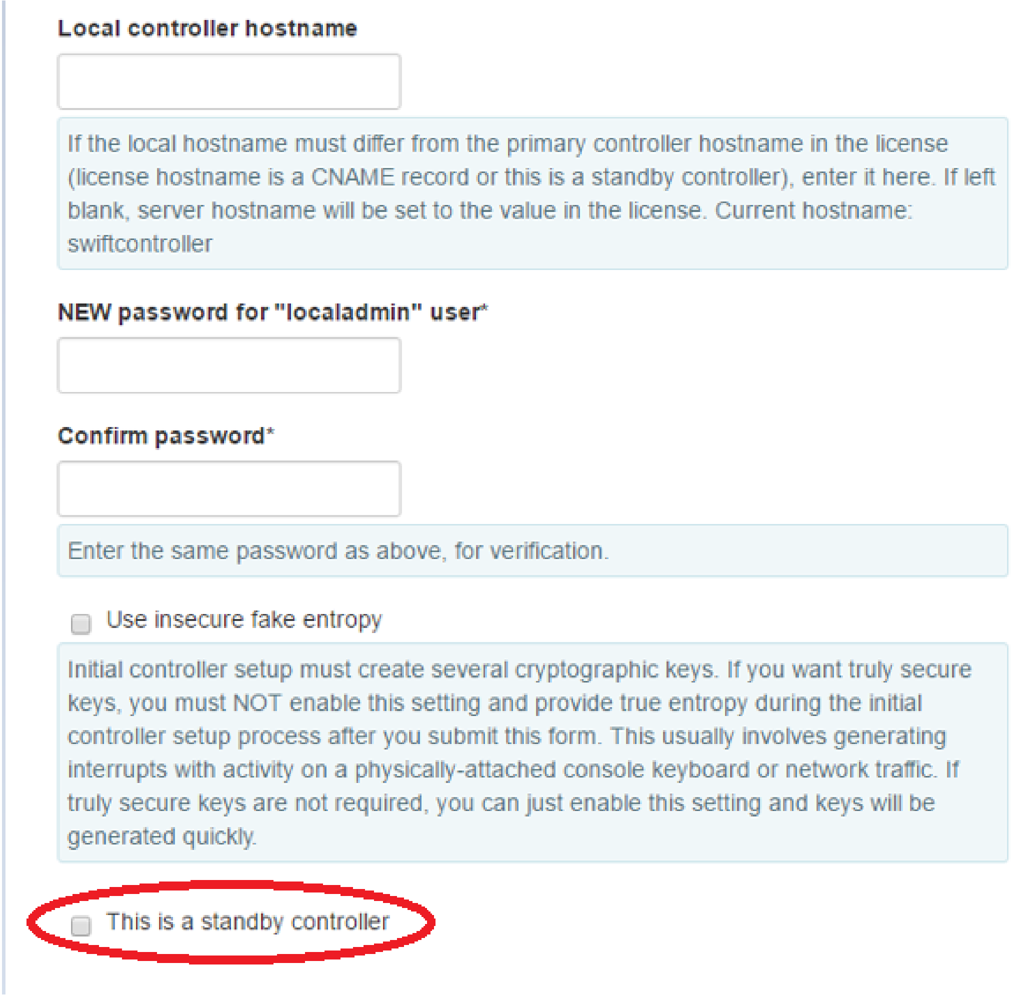
Installation

The software installation on standby is same as on the primary. To install, follow these steps:

1. Run ./SwiftStack-Controller-<swiftstack version>-installer.sh. The installation should complete with the following message:

```
You must now complete setup by pointing a web browser at this server
using HTTPS and the default port of 443.  E.g.

    https://173.36.219.124/

Log in with the username "localadmin" and the default password, "password"
```

2. After entering the URL in the browser, make sure to check ‘This is a Standby Controller’ as shown below. Enter the name same a primary controller hostname.



Local controller hostname

If the local hostname must differ from the primary controller hostname in the license (license hostname is a CNAME record or this is a standby controller), enter it here. If left blank, server hostname will be set to the value in the license. Current hostname: swiftcontroller

NEW password for "localadmin" user*

Confirm password*

Enter the same password as above, for verification.

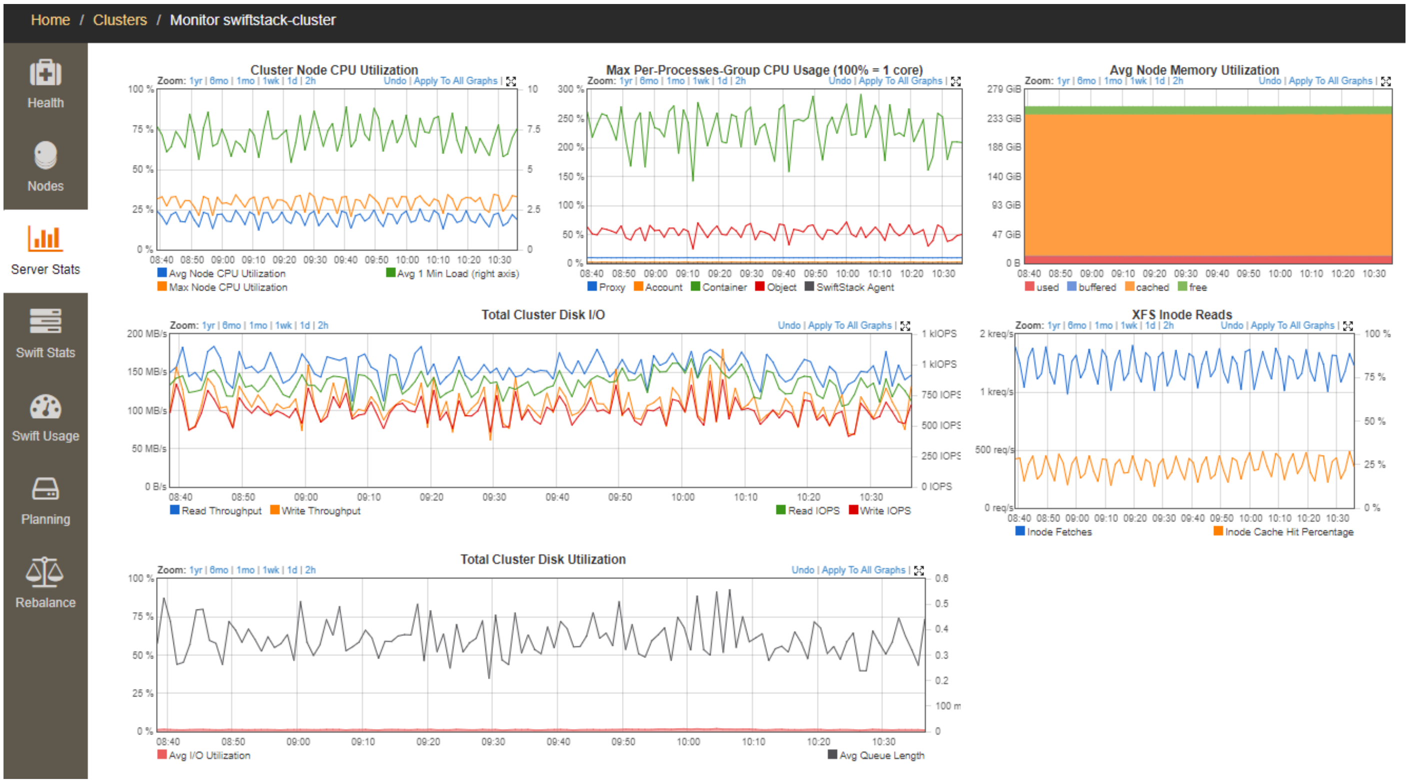
☐ Use insecure fake entropy

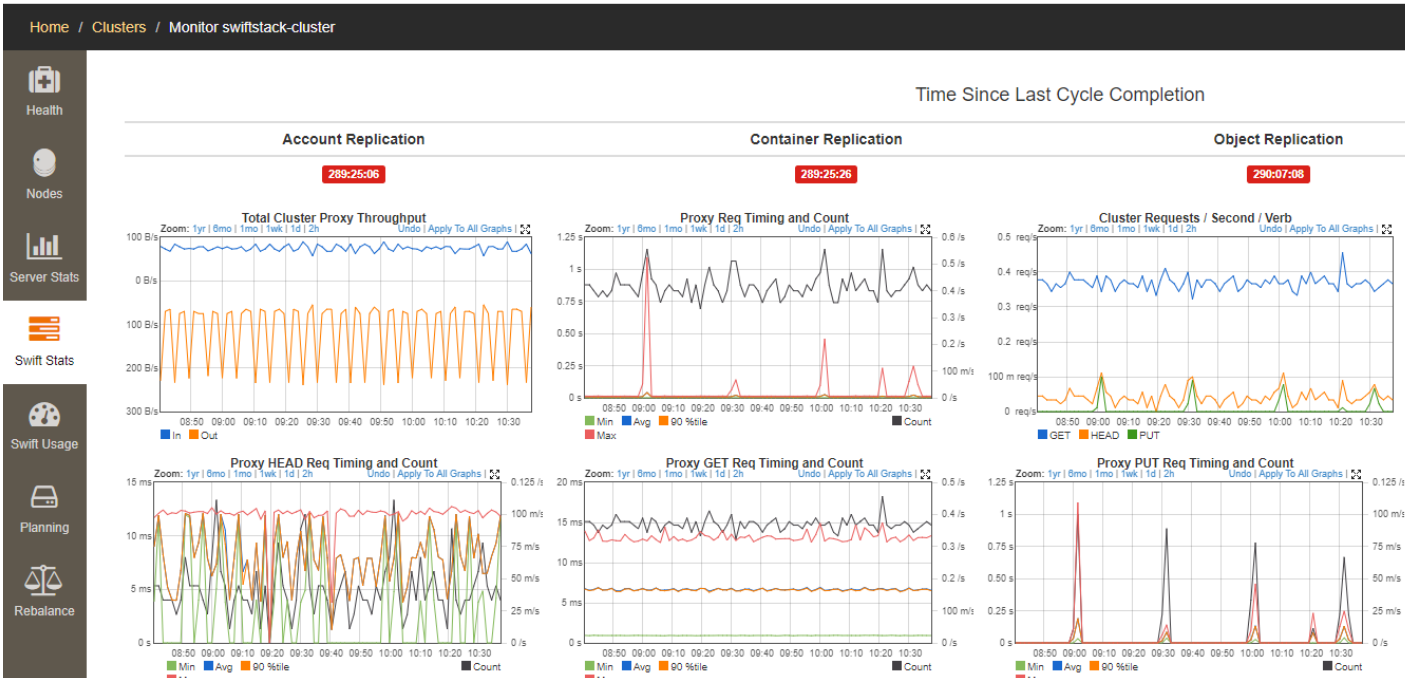
Initial controller setup must create several cryptographic keys. If you want truly secure keys, you must NOT enable this setting and provide true entropy during the initial controller setup process after you submit this form. This usually involves generating interrupts with activity on a physically-attached console keyboard or network traffic. If truly secure keys are not required, you can just enable this setting and keys will be generated quickly.

☐ This is a standby controller

SwiftStack Controller Monitoring and Metrics

The SwiftStack Controller metrics are collected in 30 second intervals, with time series data base graphs available for up to 3 years' worth of review. Many basic graphs like CPU, IO and network are available showing overall system health results both at Cluster and Node levels. Swift specific graphs also available showing trends in request handling, error counts that are useful for viewing large changes and troubleshooting.




Home / Clusters / Monitor swiftstack-cluster
Health
Nodes
Server Stats
Swift Stats
Swift Usage
Planning
Rebalance

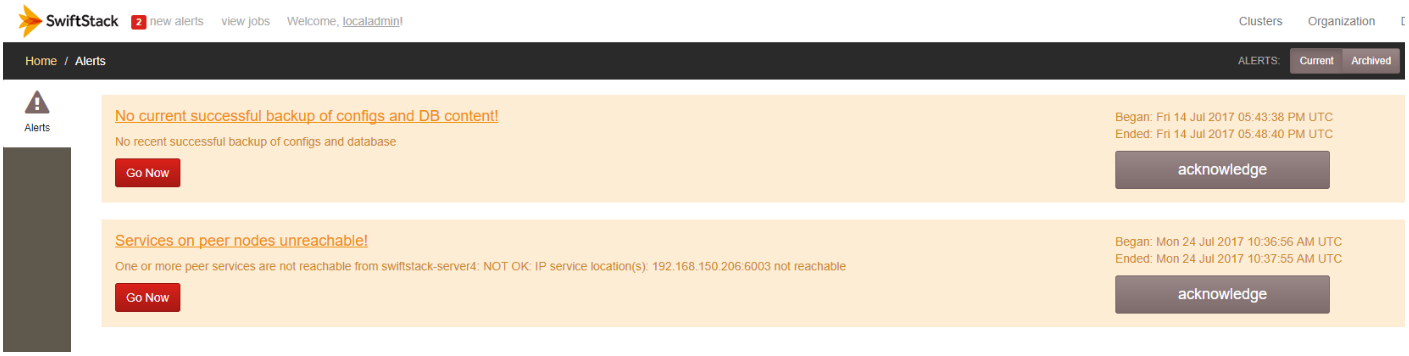
Time Since Last Cycle Completion

Account Replication	Container Replication	Object Replication
289:25:06	289:25:26	290:07:08

Total Cluster Proxy Throughput
In
Out
Proxy Req Timing and Count
Min
Avg
90 %tile
Max
Count
Cluster Requests / Second / Verb
GET
HEAD
PUT
Proxy HEAD Req Timing and Count
Proxy GET Req Timing and Count
Proxy PUT Req Timing and Count

SwiftStack Controller Alerts

SwiftStack Node’s alerts can be managed through the controller and alerts can be acknowledged and then archived as desired.



No current successful backup of configs and DB content!

No recent successful backup of configs and database

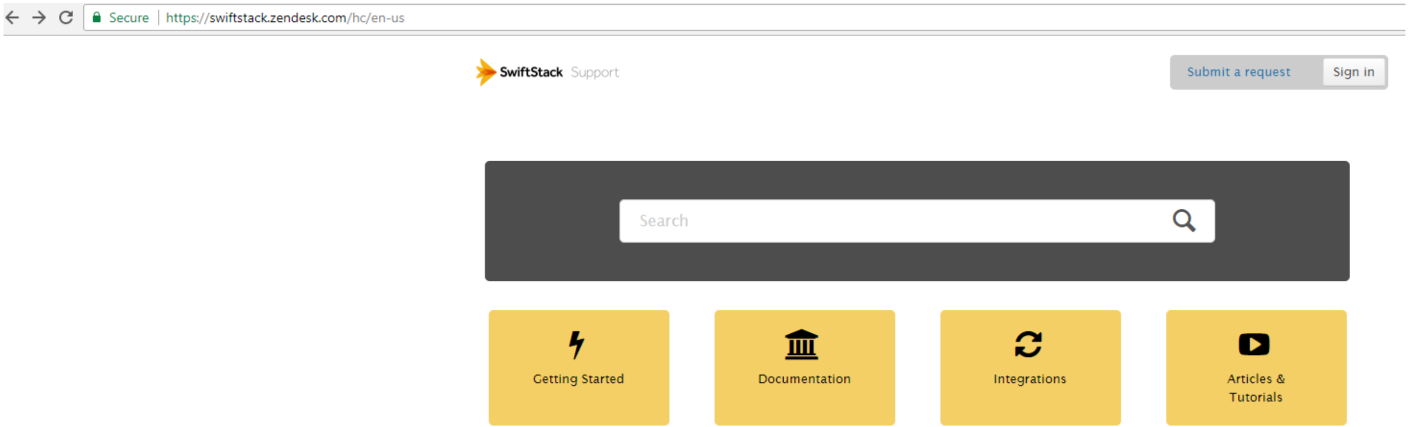
Began: Fri 14 Jul 2017 05:43:38 PM UTC
Ended: Fri 14 Jul 2017 05:48:40 PM UTC

Services on peer nodes unreachable!

One or more peer services are not reachable from swiftstack-server4: NOT OK: IP service location(s): 192.168.150.206:6003 not reachable

Began: Mon 24 Jul 2017 10:36:56 AM UTC
Ended: Mon 24 Jul 2017 10:37:55 AM UTC

Once an alert is received, you can review the alert guide in SwiftStack Controller documentation or go to theSwiftStack support portal to get more information:https://swiftstack.zendesk.com/hc/en-us



Load Testing and Performance Evaluation

Load Testing was done on the cluster to evaluate the performance under different configurations. SSBENCH tool was used to test the cluster. The following points were considered while doing the performance testing:

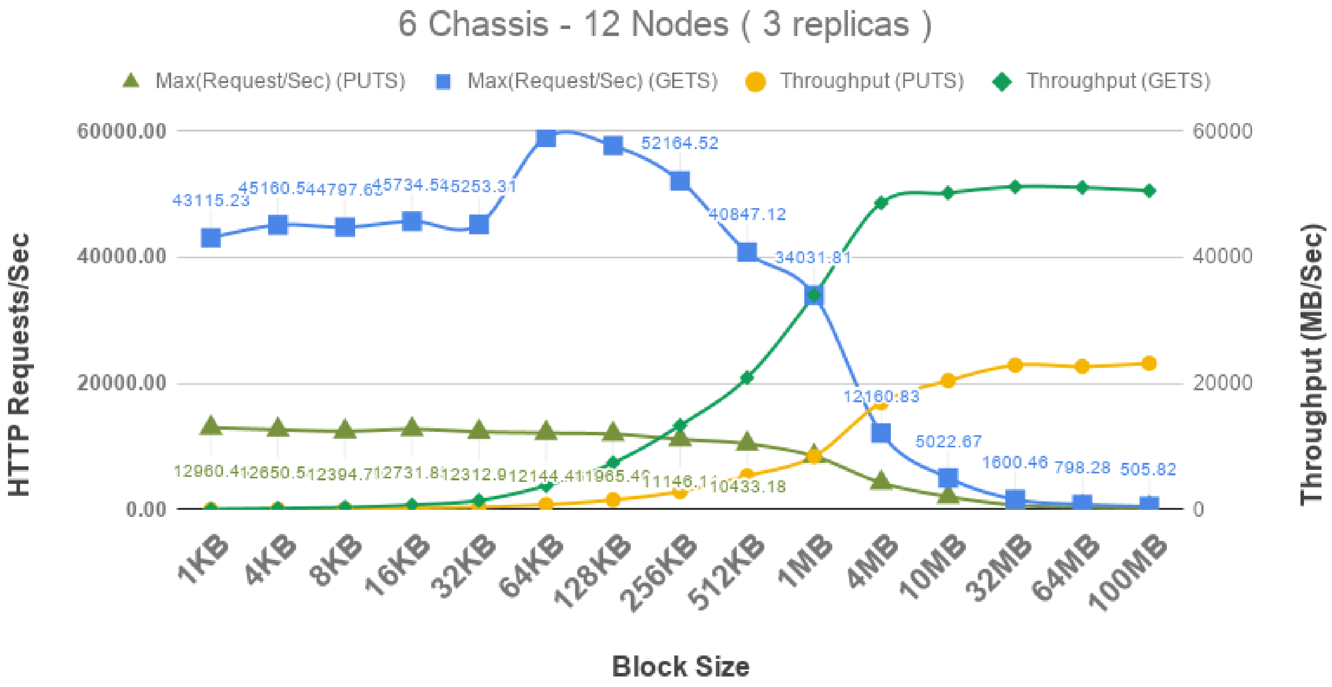
- All the tests were conducted on default configurations both from Cisco UCS and SwiftStack to make it as generic as possible. It is possible to get better results when Tuning is attempted.
- The purpose of the tests is to get performance data of the cluster and should not be considered as benchmark values. Most of the effort was spent obtaining the values of how the storage policies will be adopted for the cluster, which is dual nodes 6 chassis (12 nodes). No any attempt was made to tune the configuration to its optimal values.
- All of the tests listed below were conducted in dual nodes 6 chassis (12 nodes) and Intel® Xeon® Silver 4110 Processor.
- A sufficient number of clients were added to saturate the cluster. Each server had 40 Gb of Network configured for client traffic. Hence, in a 12 Node configuration, with a maximum capacity of 480 Gb of the client network configured on servers, and an equal number of clients were added to the infrastructure that can push the traffic to servers close to 480 Gb. This is very useful when your workload is bandwidth intensive, for example, on a block size of 1MB and more.

Performance data was gathered on the following configurations:

	Disks (10 TB HDD for objects and 2TB NVMe for A/C and containers)	Network
6 chassis 12 Node w/ 3 replicas	336 HDD + 12 NVMe	480Gb client + 480Gb cluster/replication
6 chassis 12 Node w/ ec4-3	336 HDD + 12 NVMe	480Gb client + 480Gb cluster/ replication
6 chassis 12 Node w/ ec8-4	336 HDD + 12 NVMe	480Gb client + 480Gb cluster/replication
6 chassis 12 Node w/ ec15-4	336 HDD + 12 NVMe	480Gb client + 480Gb cluster/replication

6 Chassis, 12 Nodes, and 3 Replicas

The values obtained for PUTS and GETS are plotted below:



The following points can be concluded:

- About 12650 ~ 20000 requests/sec of PUT requests at 1k and 4k block sizes.
- More than 60000 requests/sec of GET requests observed as the peak. This is slightly more than 3 times the PUTS requests. It has to be noted that PUTS aka writes will have an overhead of 3x because of replication.
- Peak bandwidth (Throughput = Requests/Sec * Block Size) of ~ 23.2 GBPS (23,194 MBPS) when 100MB block size testing for PUTS and ~ 51.2 GBPS (51214.72 MBPS) when 32MB block size testing for GETS.

Figure 44 Server Statistic Data Snapshot (CPU Utilization and Disk I/O) from the Controller while Running the **PUTS** 3 Replicas Tests

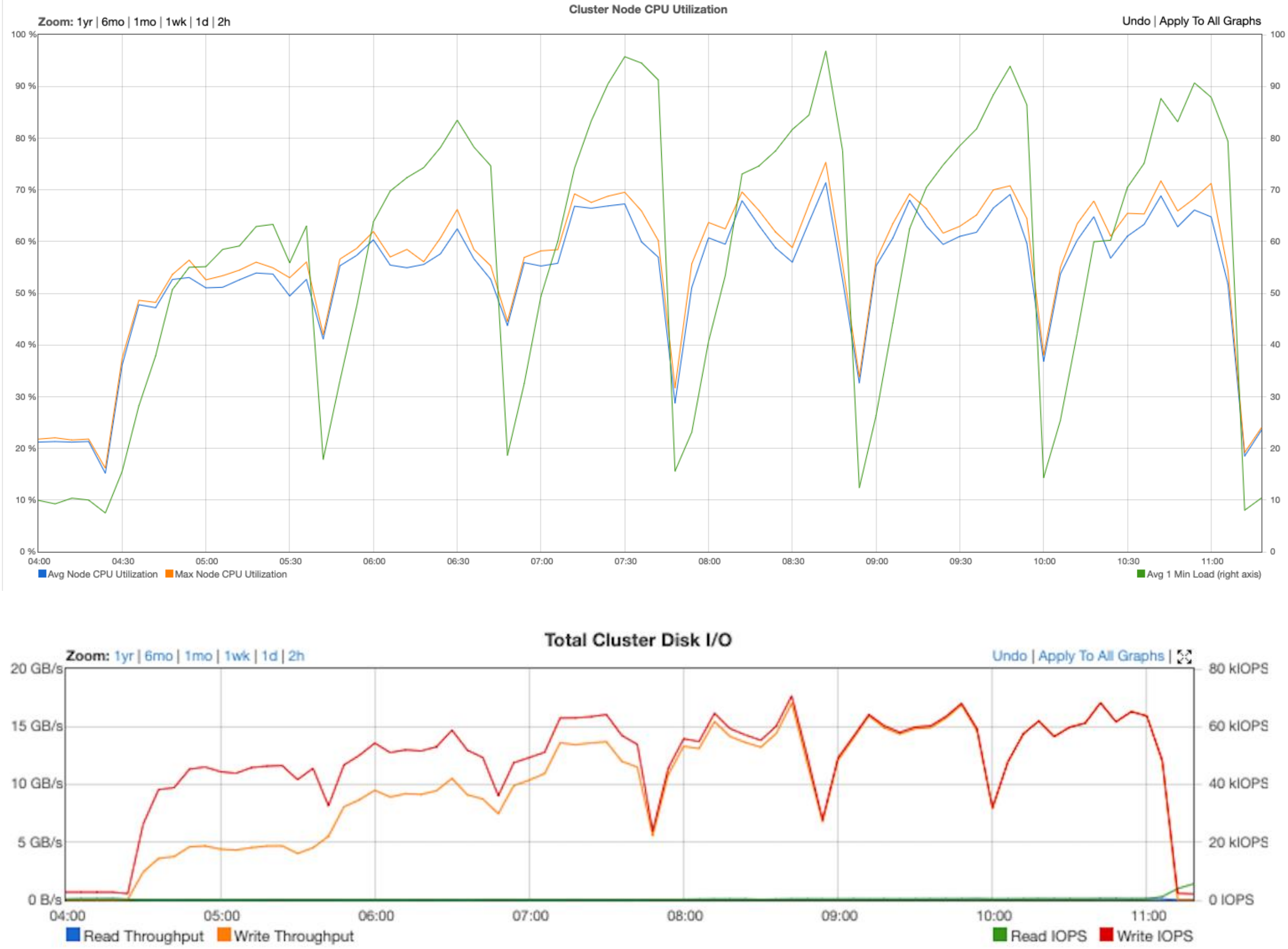
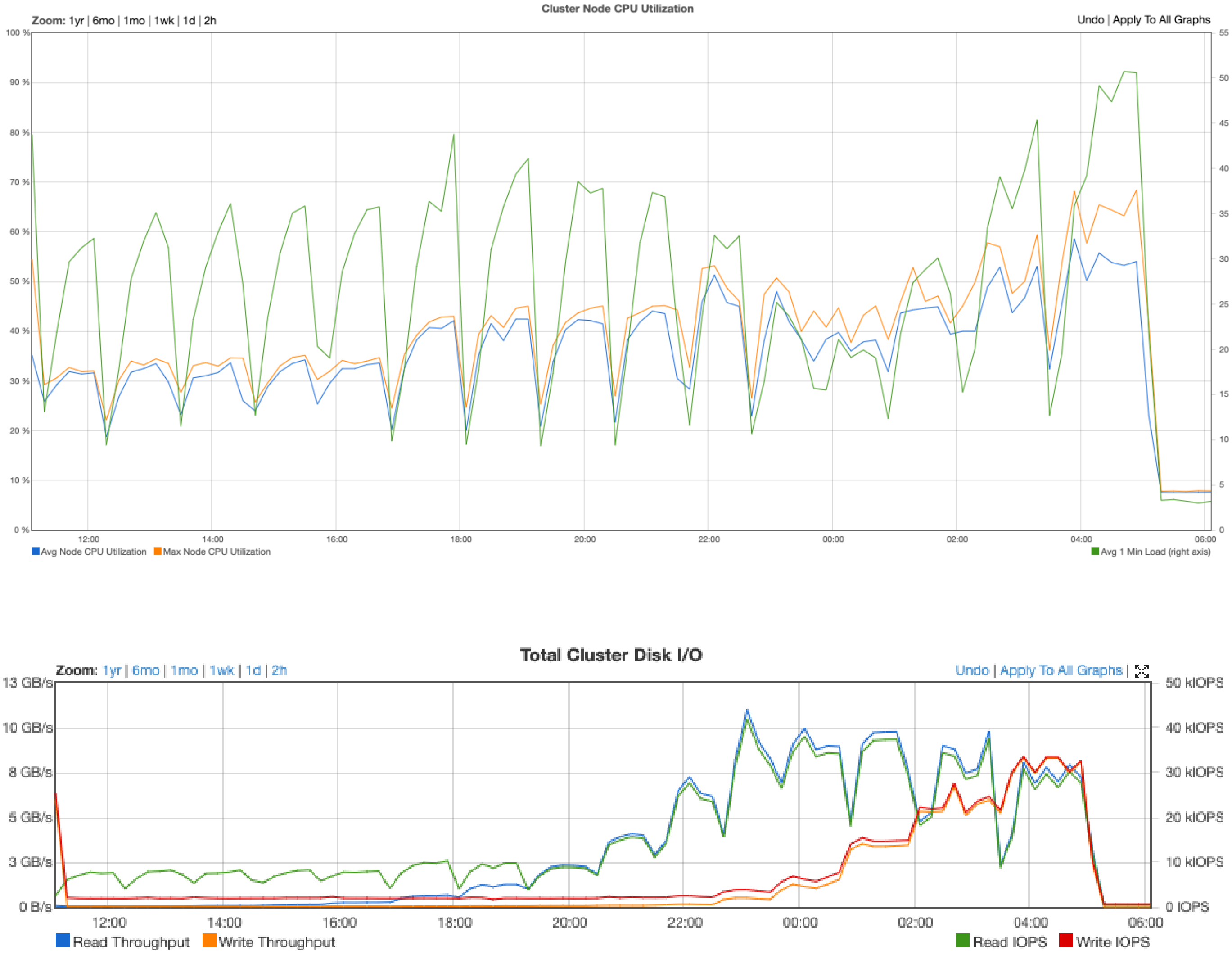
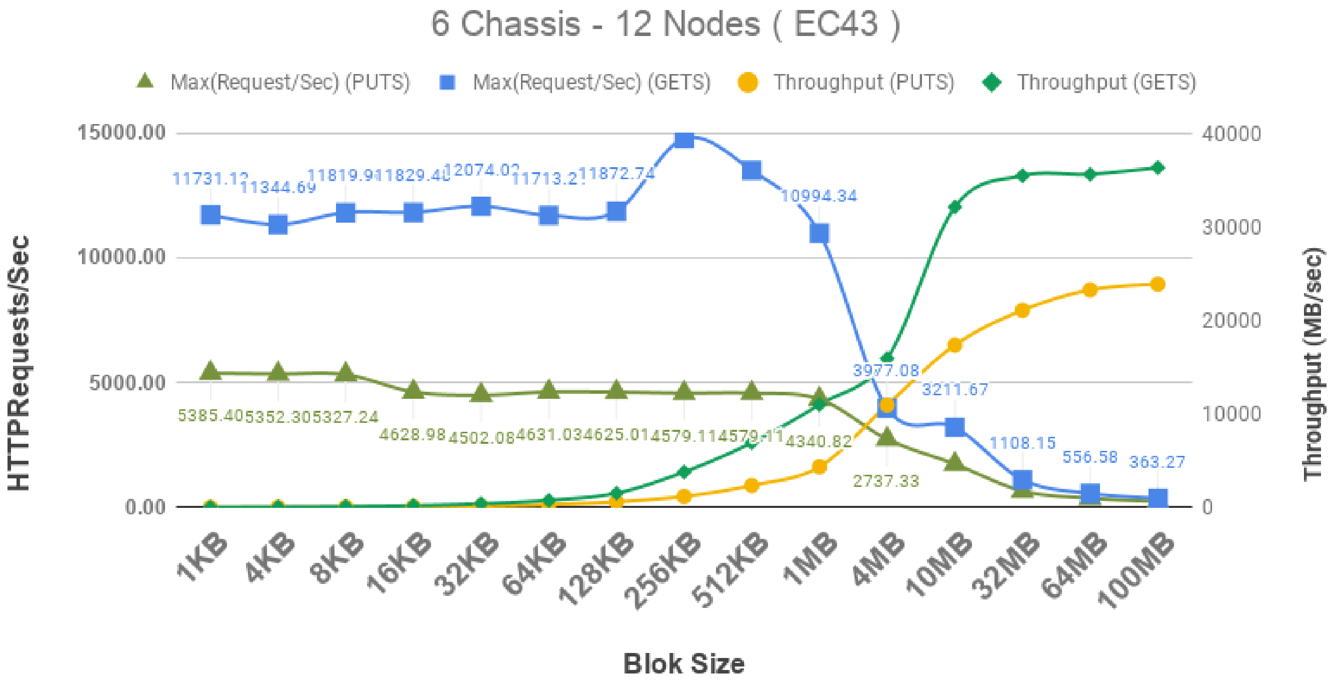


Figure 45 Server Statistic Data Snapshot (CPU Utilization and Disk I/O) from the Controller while Running the **GETS** 3 Replicas Tests



6 Chassis and 12 Nodes with ec4-3 Replicas



The following points can be concluded:

- About 5300 request/sec of PUT requests at 1k and 4k block sizes.
- About 14790 request/sec of GET requests observed as its peak at 256k block size.
- Peak bandwidth (Throughput = Requests/Sec * Block Size) of ~ 23.8 GBPS (23,868 MBPS) when 100MB block size testing for PUTS and ~ 36.3 GBPS (36327 MBPS) when 100MB block size testing for GETS

Figure 46 Server Statistic Data Snapshot (CPU Utilization and Disk I/O) from the Controller while Running the **PUTS** 3 Replicas Tests

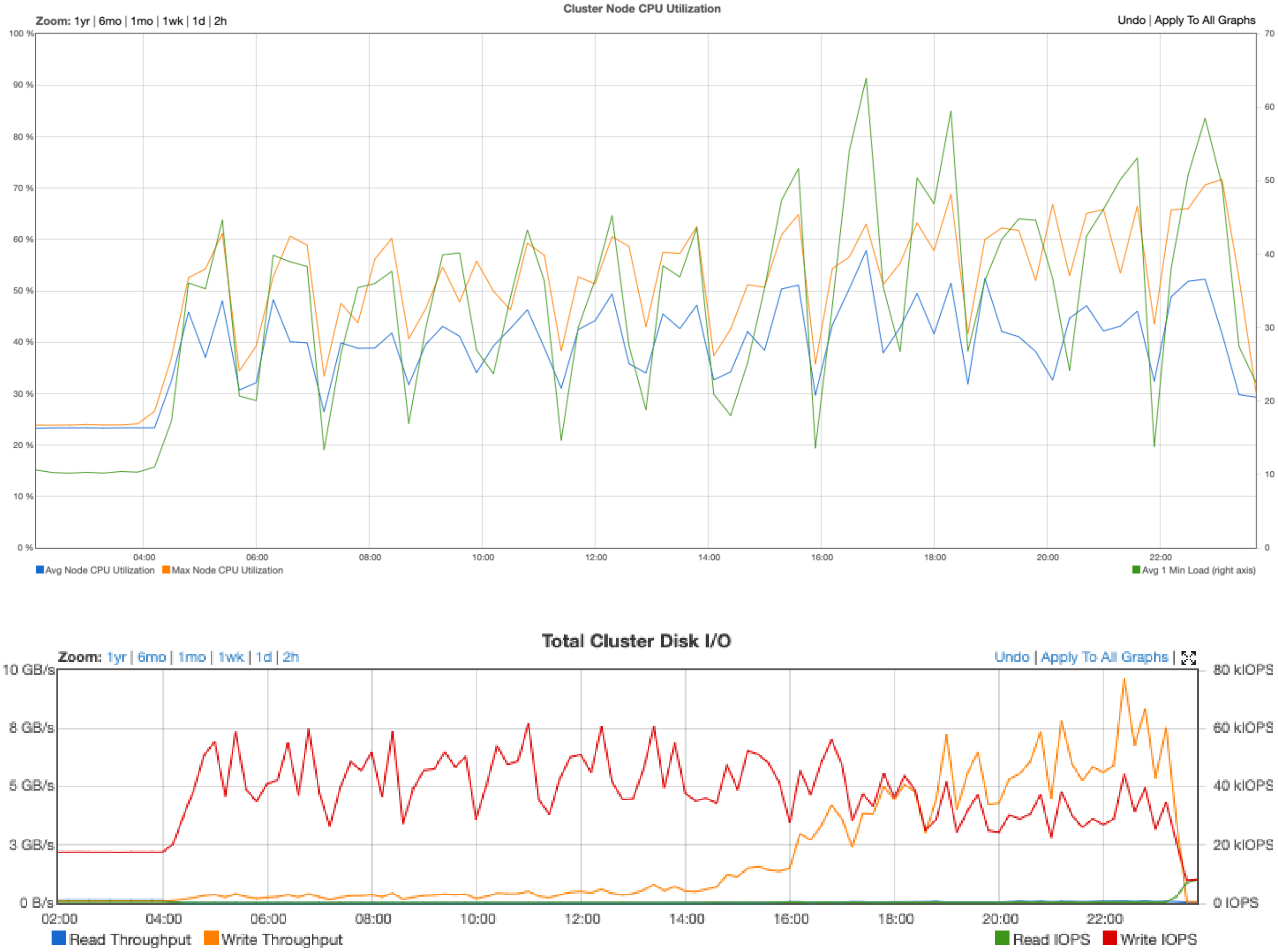
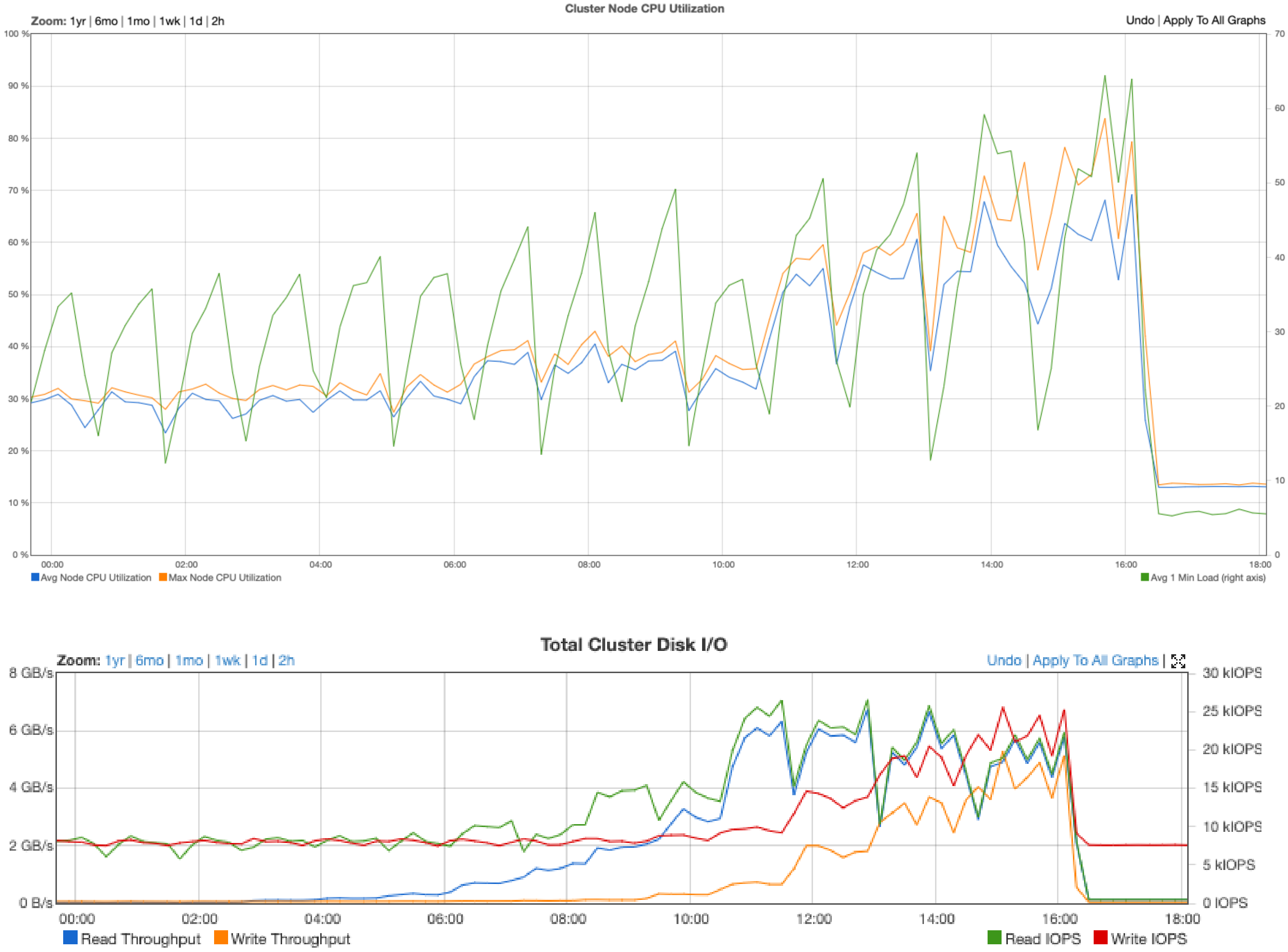
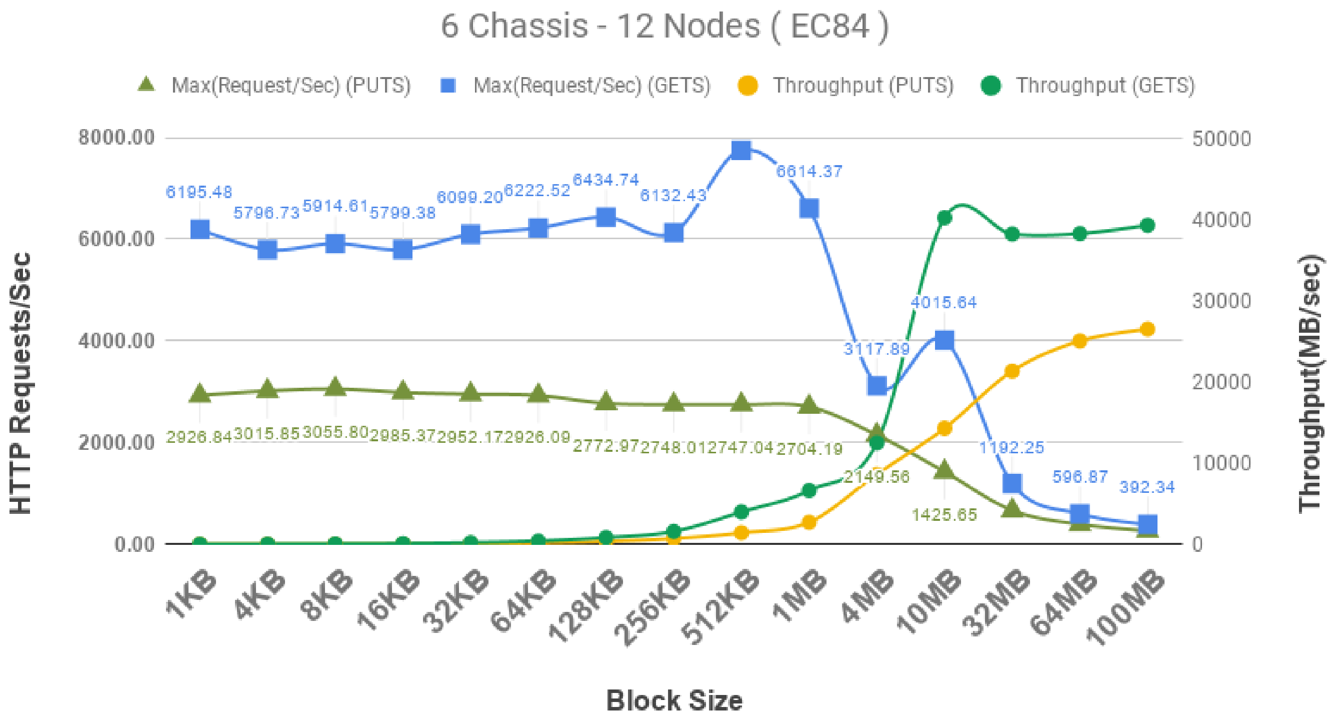


Figure 47 Server Statistic Data Snapshot (CPU Utilization and Disk I/O) from the Controller while Running the **GETS** 3 Replicas Tests



6 Chassis and 12 Nodes with ec8-4 Replicas



The following points can be concluded:

- About 3015 ~ 3055 request/sec of PUT requests at 4k and 8k block sizes.
- About 7754.34 request/sec of GET requests observed as its peak at 512k block size.
- Peak bandwidth (Throughput = Requests/Sec * Block Size) of ~ 26.4 GBPS (26,449 MBPS) when 100MB block size testing for PUTS and ~ 40.1 GBPS (40156 MBPS) when 10MB block size testing for GETS
- The graphs collected from the controller show a higher peak values than the average values obtained from ssbench.

Figure 48 Server Statistic Data Snapshot (CPU Utilization and Disk I/O) from the Controller while Running the **PUTS** 3 Replicas Tests

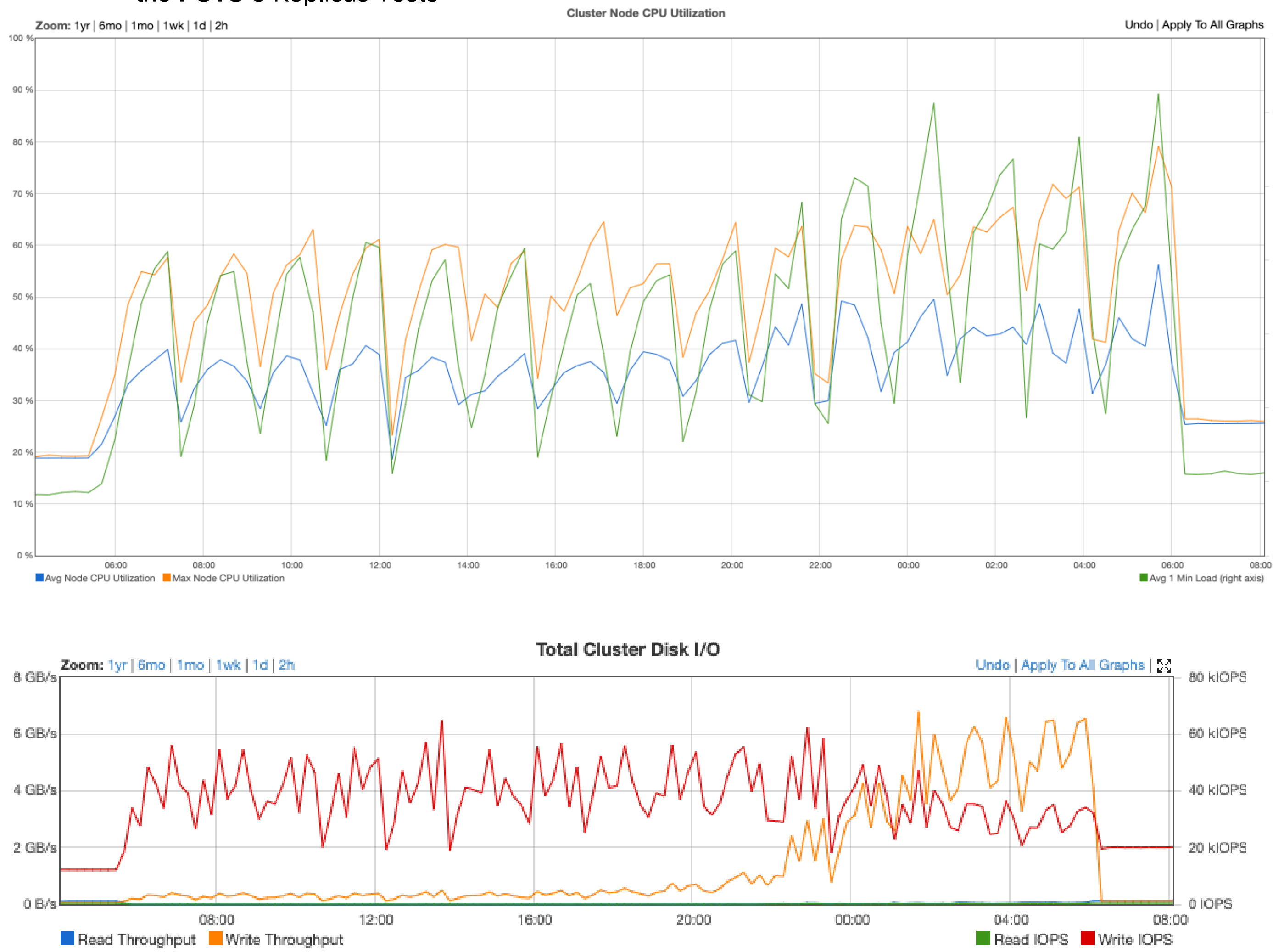
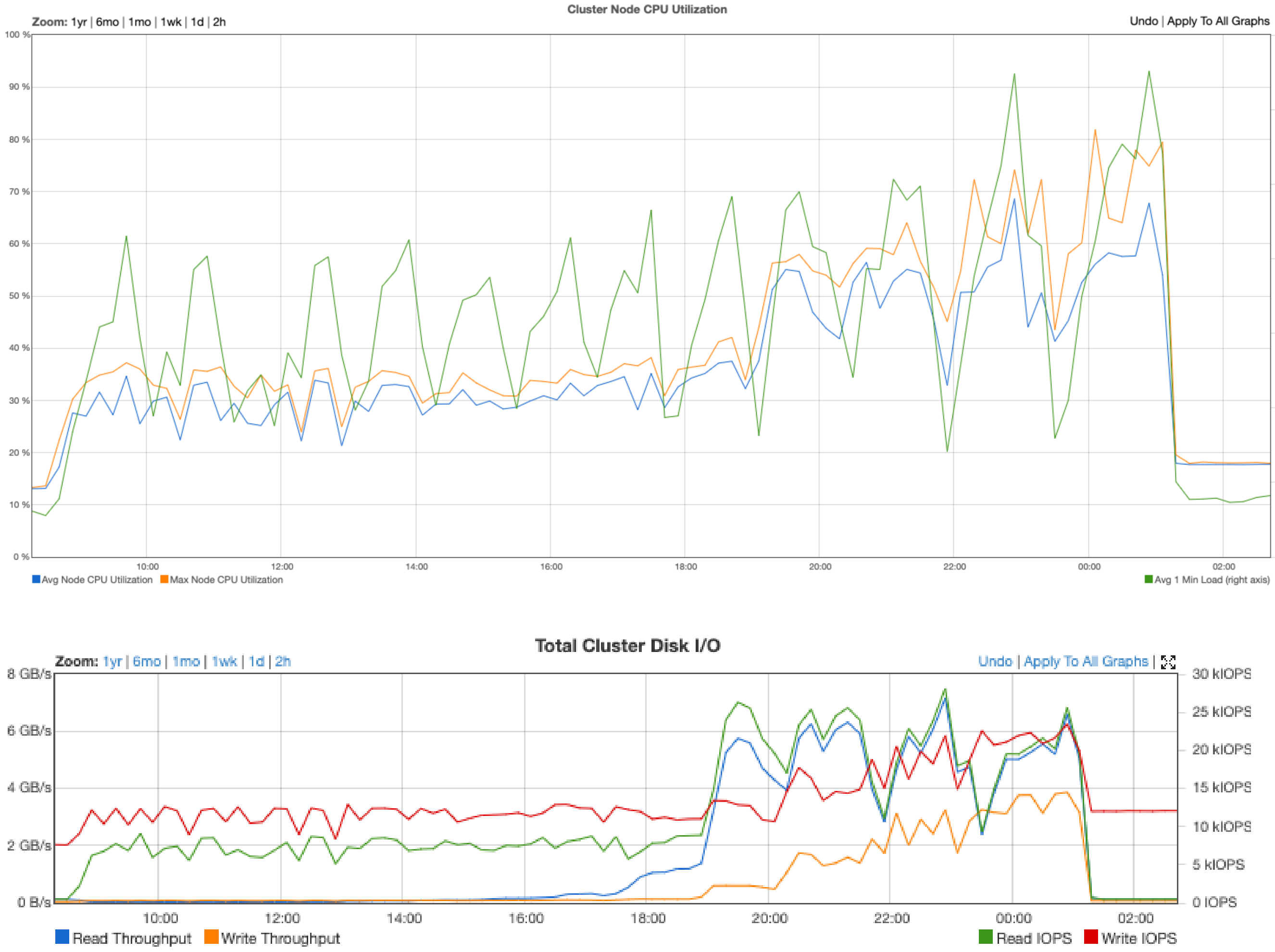
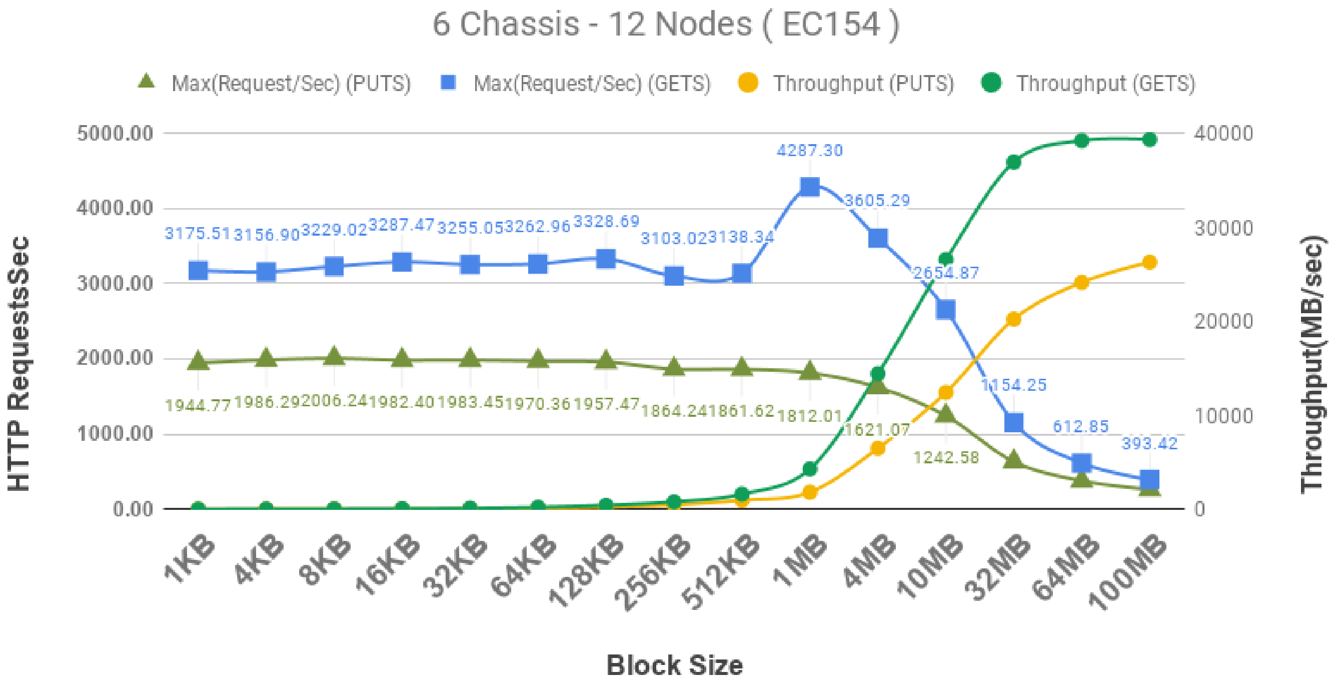


Figure 49 Server Statistic Data Snapshot (CPU Utilization and Disk I/O) from the Controller while Running the **GETS** 3 Replicas Tests



6 Chassis and 12 Nodes with ec15-4 Replicas



- About 2066 request/sec of PUT requests at 8k block size.
- About 4287.30 request/sec of GET requests observed as its peak at 1MB block size.
- Peak bandwidth (Throughput = Requests/Sec * Block Size) of ~ 26.2 GBPS (26,282 MBPS) when 100MB block size testing for PUTS and ~ 39.4 GBPS (39342 MBPS) when 100MB block size testing for GETS
- The graphs collected from the controller show a higher peak values than the average values obtained from ssbench.

Figure 50 Server Statistic Data Snapshot (CPU Utilization and Disk I/O) from the Controller while Running the **PUTS** 3 Replicas Tests

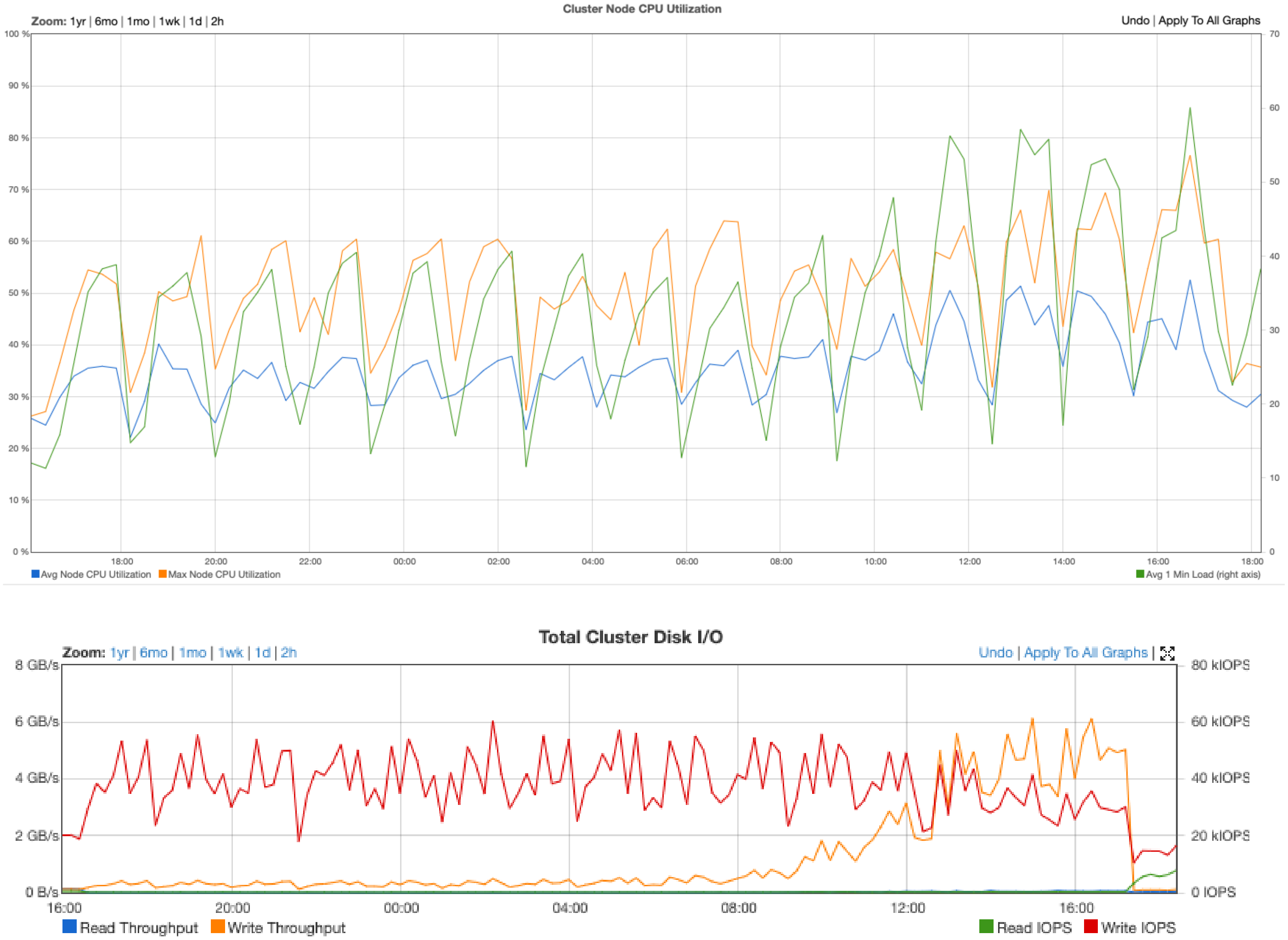
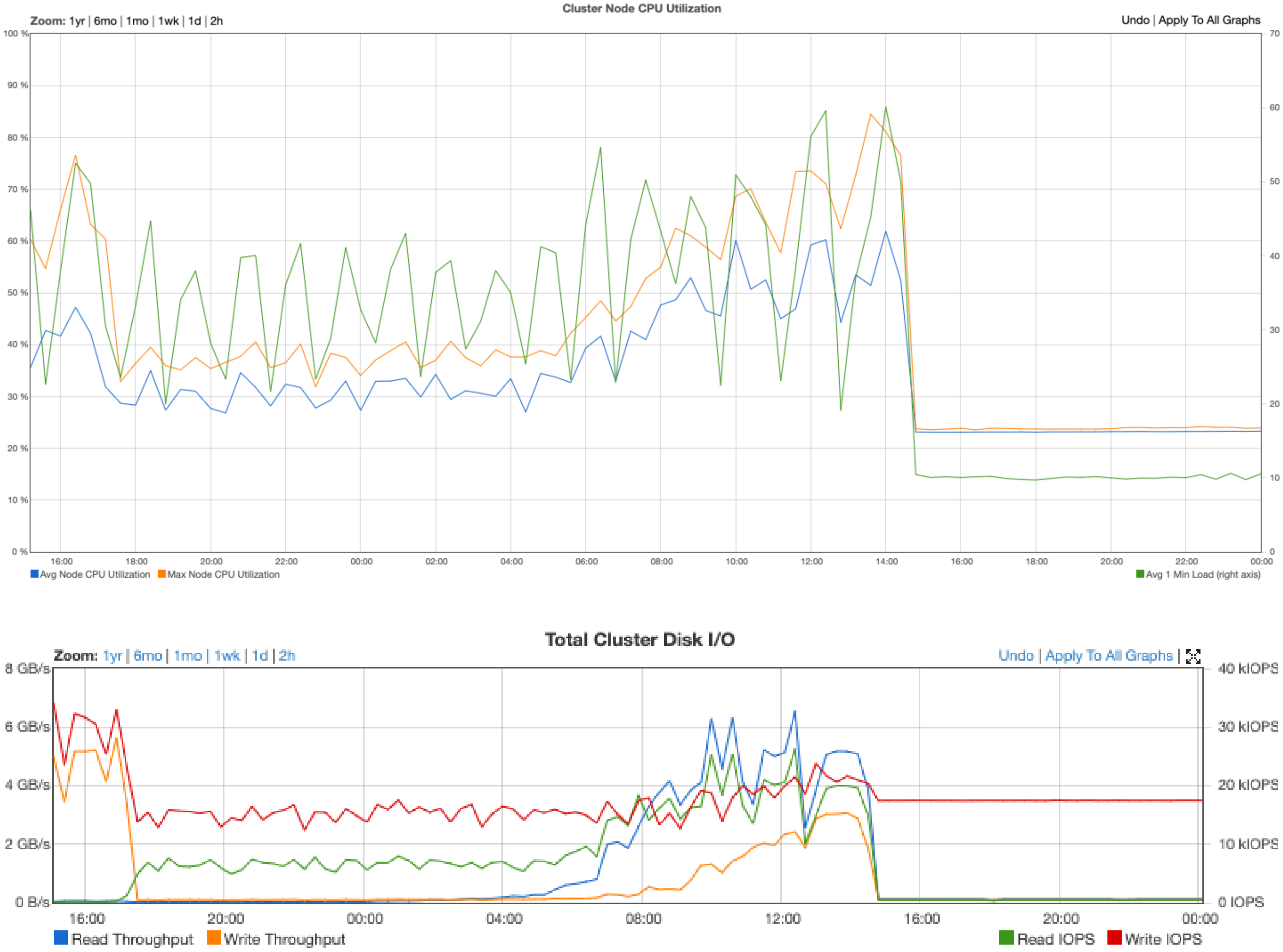


Figure 51 Server Statistic Data Snapshot (CPU Utilization and Disk I/O) from the Controller while Running the **GETS** 3 Replicas Tests



High Availability and Business Continuity

The high availability of the solution was validated by failing one of the components of the infrastructure. The following points were considered for business continuity:

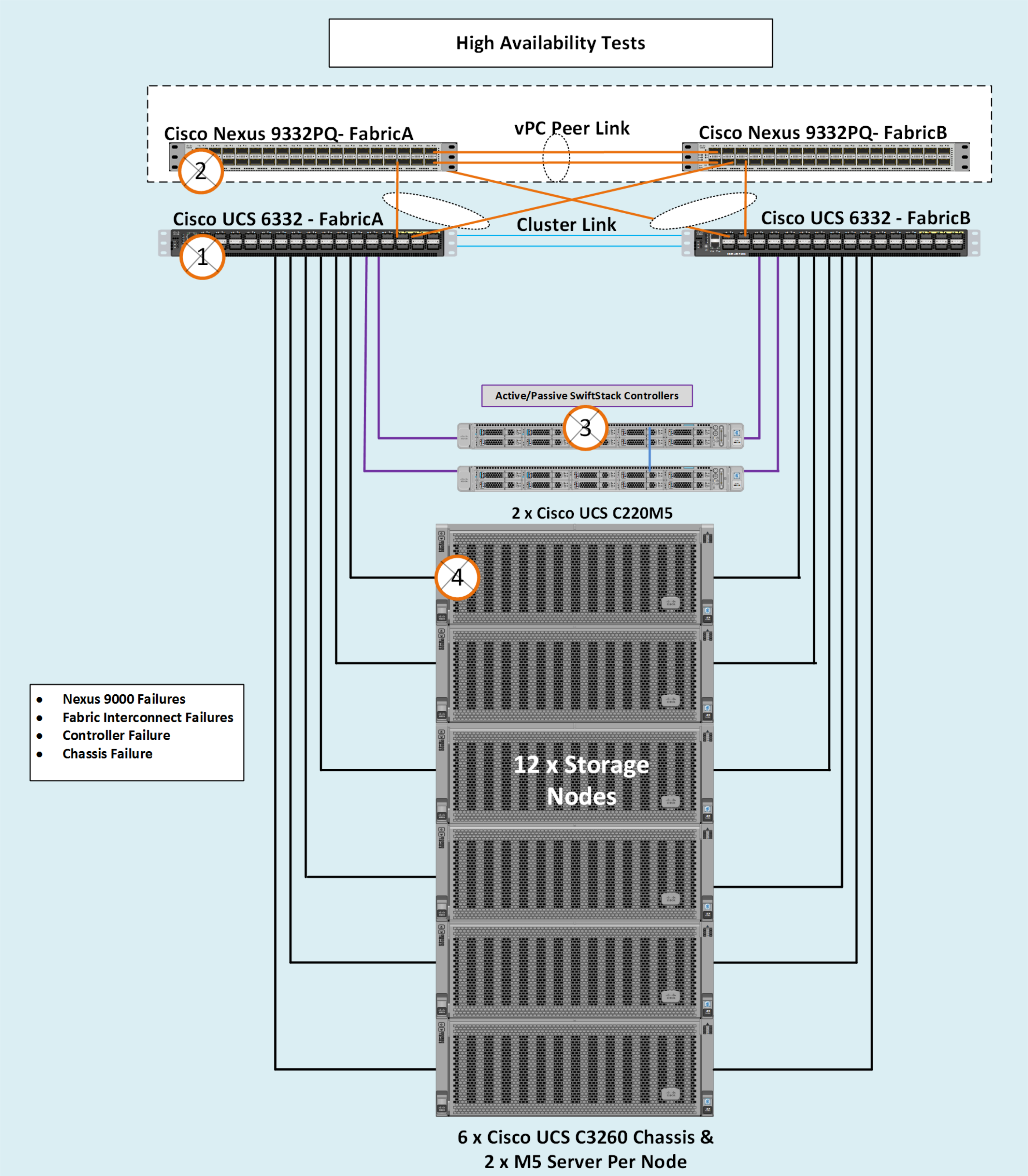
- The cluster will have reasonable amount of load when the fault is injected. The outputs such as Total Cluster Disk I/O or Gets/Puts requested to be gathered before and after Fault injection.
- Only one fault injected at any point of time. No double failures considered.
- Performance degradation is acceptable but there shouldn't be business interruption. The underlying infrastructure components should continue to operate with the remaining components.

A few of the high availability tests conducted are:

- SwiftStack Chassis and Node Failures
- SwiftStack Controller Failures
- Cisco UCS Fabric Interconnect Failure
- Cisco Nexus Switch Failure

Figure 52 shows the High Availability Tests conducted on the infrastructure.

Figure 52 High Availability Test



SwiftStack Node Failures

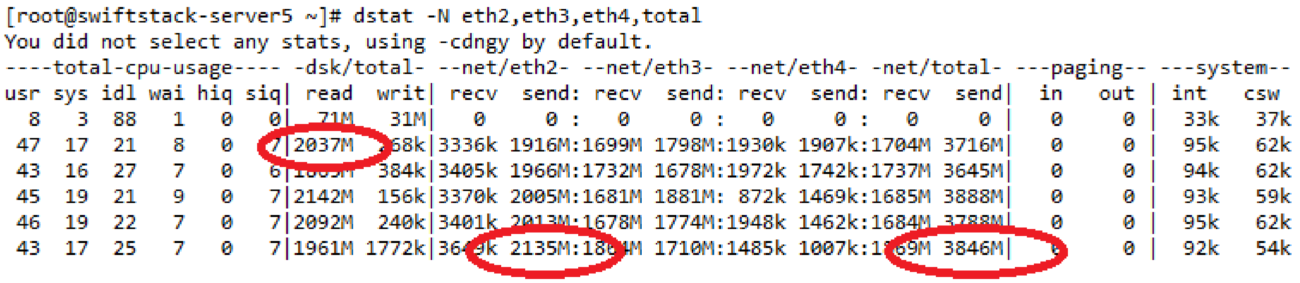
Client workload started on the Cluster with SwiftStack Load Balancer.

Sequence of Events

While the cluster was running, load data was gathered on the IO while one chassis was powered off (nodes pulled out from the chassis), simulating a hard failure. After few minutes of running the cluster in this state, a hard power down of nodes issues from Cisco UCS. The nodes were powered on and both the chassis' were brought up again.

System when Workload Started

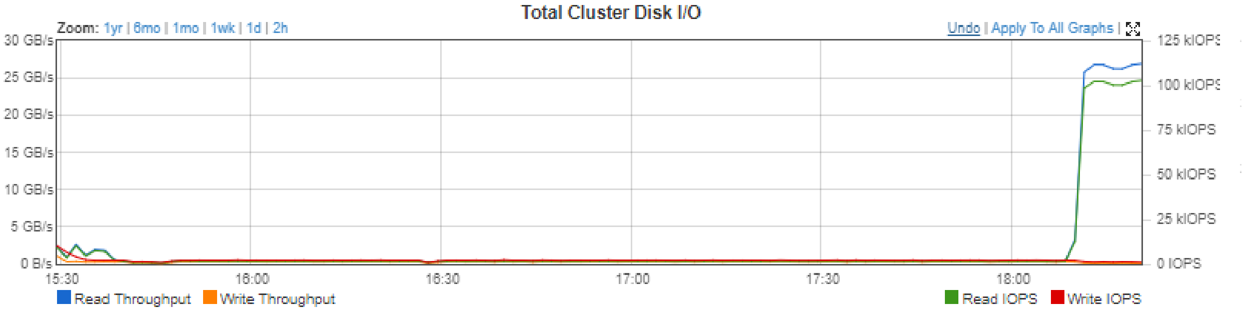
Dstat output gathered on one of the server nodes



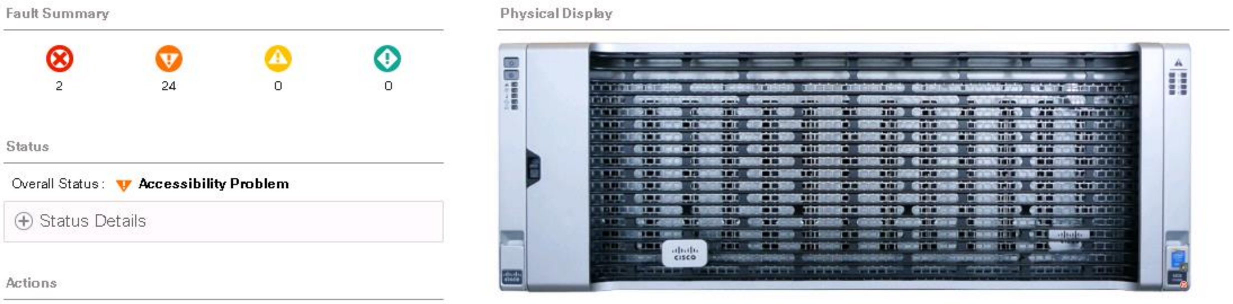
```
[root@swiftstack-server5 ~]# dstat -N eth2,eth3,eth4,total
You did not select any stats, using -cdngy by default.
----total-cpu-usage---- -dsk/total- --net/eth2- --net/eth3- --net/eth4- -net/total- ---paging-- ---system--
usr sys idl wai hiq siq| read  writ| recv  send: recv  send: recv  send: recv  send|  in   out | int   csw
  8   3  88   1   0   0|  71M   31M|   0     0 :   0     0 :   0     0 :   0     0 |   0     0 |  33k   37k
 47  17  21   8   0   7|2037M   68k|3336k 1916M:1699M 1798M:1930k 1907k:1704M 3716M|   0     0 |  95k   62k
 43  16  27   7   0   6|[illegible]  384k|3405k 1966M:1732M 1678M:1972k 1742k:1737M 3645M|   0     0 |  94k   62k
 45  19  21   9   0   7|2142M  156k|3370k 2005M:1681M 1881M: 872k 1469k:1685M 3888M|   0     0 |  93k   59k
 46  19  22   7   0   7|2092M  240k|3401k 2013M:1678M 1774M:1948k 1462k:1684M 3788M|   0     0 |  95k   62k
 43  17  25   7   0   7|1961M 1772k|3649k 2135M:18[illegible]M 1710M:1485k 1007k:1[illegible]69M 3846M|   0     0 |  92k   54k
```

This server was doing disk reads of 2000M, a total network output of 3800M when the fault was injected.

The total IO of the cluster was about 25 GB/s before fault injection.



Cisco UCS shows critical alert that the chassis can no longer be accessed.



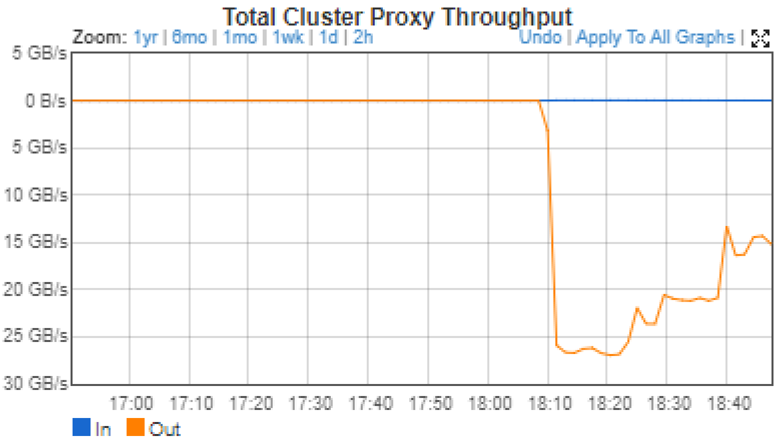
Status

Overall Status : Accessibility Problem

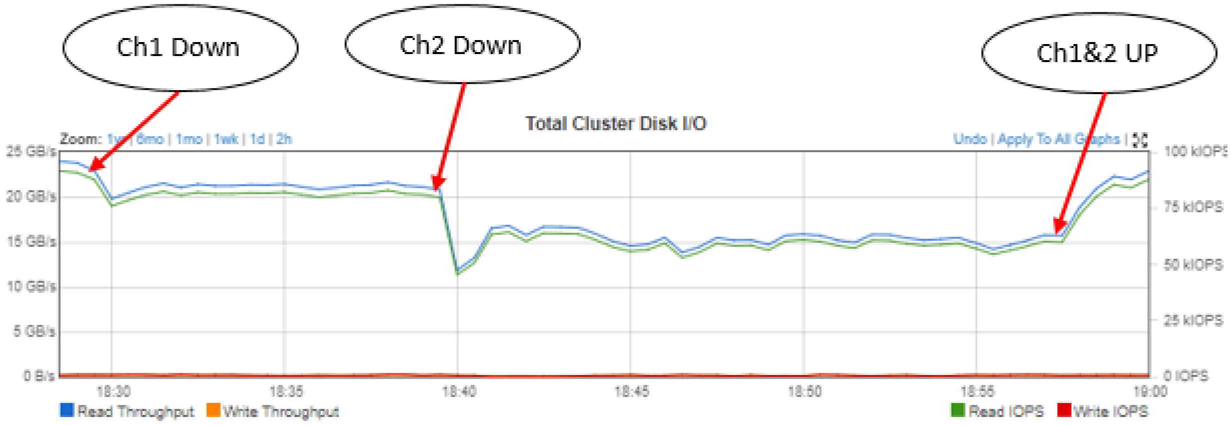
Status Details

Actions

Drop in Proxy Throughput observed as shown below:



The overall activity as recorded by SwiftStack controller is as shown below.

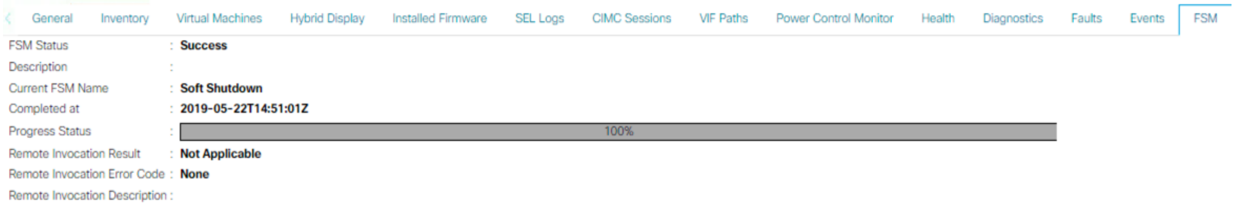


Summary

- When the first chassis was brought down, a drop in IO observed around 18:30. System continues to operate with 20GB/sec.
- At about 18:40 second chassis was brought down and IO dropped almost to 15GB/sec.
- System fully recovers, then chassis' were plugged-in.

SwiftStack Controller Failures

The SwiftStack controller is not in the data path. We did not notice any interruption to the client traffic when the Controller node was brought down.



FSM Status : **Success**

Description :

Current FSM Name : **Soft Shutdown**

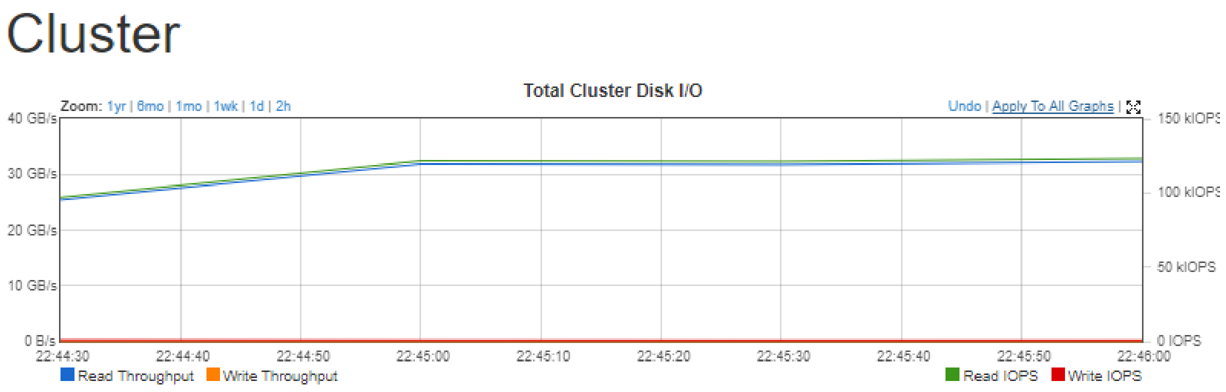
Completed at : **2019-05-22T14:51:01Z**

Progress Status :

Remote Invocation Result : **Not Applicable**

Remote Invocation Error Code : **None**

Remote Invocation Description :



Summary

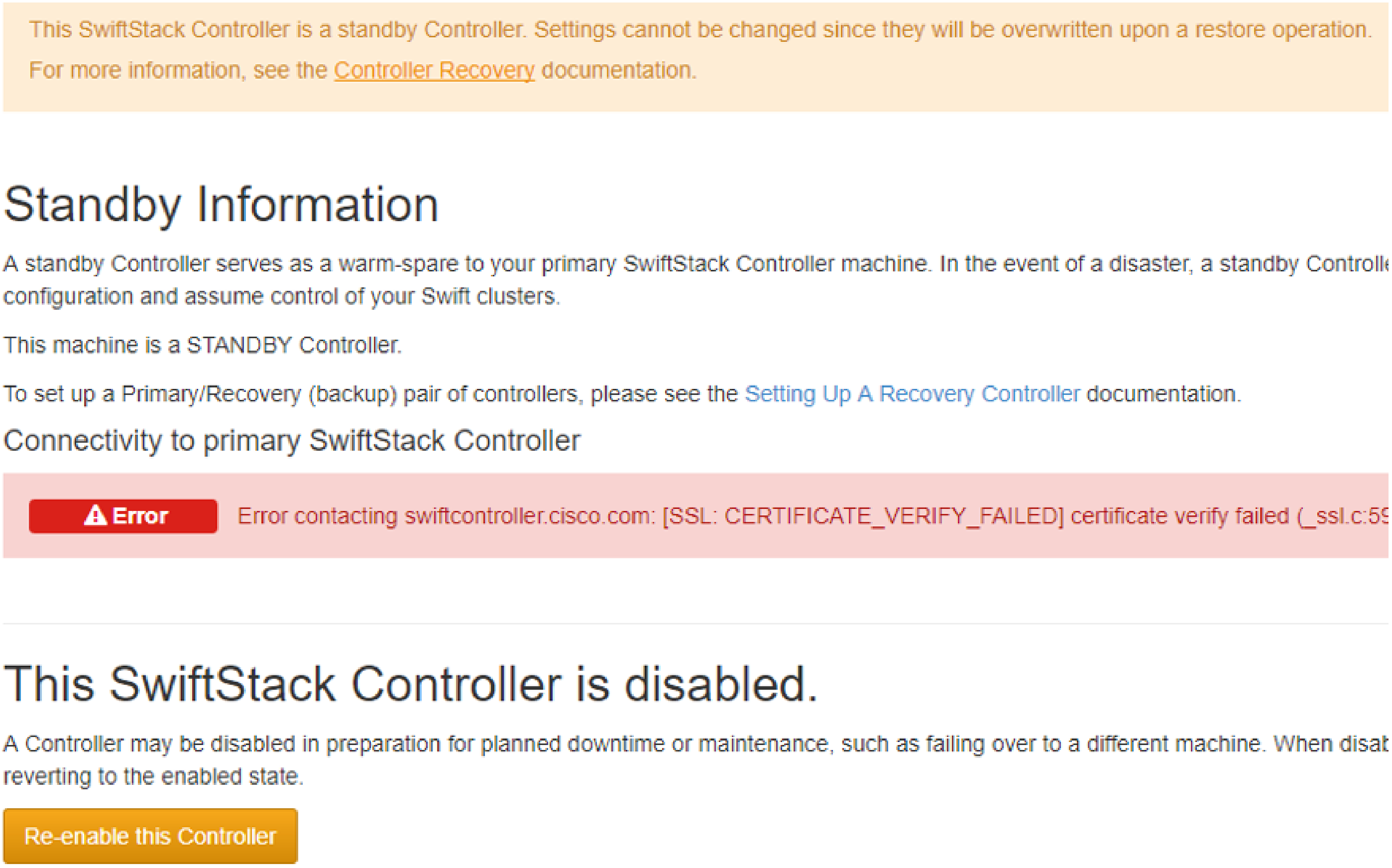
The clients communicated directly with the PACO nodes and there was no interruption. However any monitoring, SNMP alerts configured, and so on, will not work. Activate the Standby Controller and make it Primary in case it is decided that it’s a total hardware failure and cannot be recovered soon. This is explained in the section below.

Activate Standby Controller to Primary

To activate the standby controller to the primary, follow these steps:

1. Update the DNS entry (or /etc/hosts) for the name of the controller to be accessed appropriately.
2. Log into the controller UI, click the Admin tab and then click Recovery, or log into https://swiftcontroller.cisco.com/recovery/standbys/.

The following information is displayed:



This SwiftStack Controller is a standby Controller. Settings cannot be changed since they will be overwritten upon a restore operation. For more information, see the Controller Recovery documentation.

Standby Information

A standby Controller serves as a warm-spare to your primary SwiftStack Controller machine. In the event of a disaster, a standby Controlle configuration and assume control of your Swift clusters.

This machine is a STANDBY Controller.

To set up a Primary/Recovery (backup) pair of controllers, please see the Setting Up A Recovery Controller documentation.

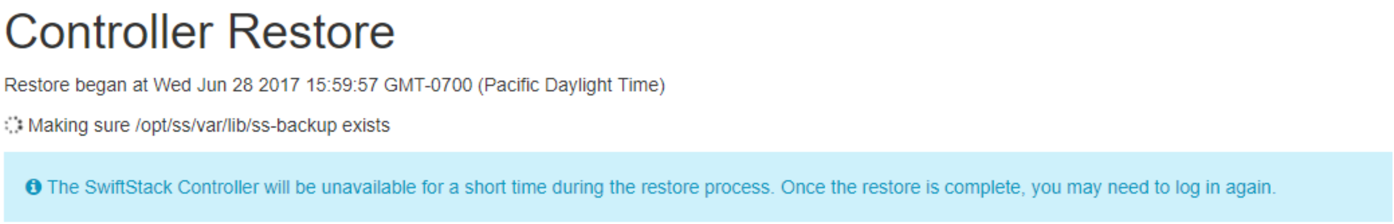
Connectivity to primary SwiftStack Controller

Error contacting swiftcontroller.cisco.com: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:59

This SwiftStack Controller is disabled.

A Controller may be disabled in preparation for planned downtime or maintenance, such as failing over to a different machine. When disab reverting to the enabled state.

3. Restore the controller before enabling it; during the backup, it will sync up from the database and once completed can be enabled.



Controller Restore

Restore began at Wed Jun 28 2017 15:59:57 GMT-0700 (Pacific Daylight Time)

Making sure /opt/ss/var/lib/ss-backup exists

ℹ The SwiftStack Controller will be unavailable for a short time during the restore process. Once the restore is complete, you may need to log in again.

The standby controller becomes the primary. You may continue to operate in this mode or revert to the original primary when the problem was fixed.

Cisco UCS Fabric Interconnect Failures

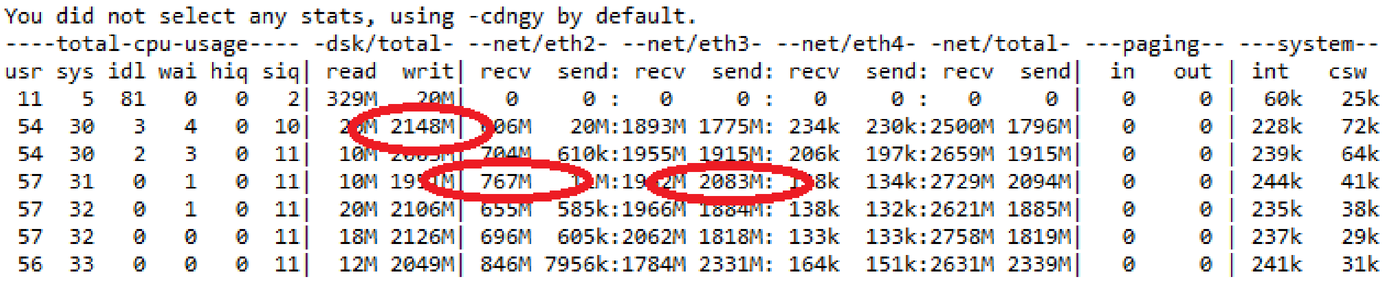
Client Workload Started on the Cluster

The following is the sequence of events:

1. While cluster was running on load data gathered on IO. One of the Fabrics was rebooted and any changes to IO pattern was observed. The reboot was done on Secondary first and then the tests were repeated on Primary.

2. The tests were conducted for both Gets and Puts operations.

3. Dstat output gathered on one of the server nodes while running PUTS.



```
You did not select any stats, using -cdngy by default.
----total-cpu-usage---- -dsk/total- --net/eth2- --net/eth3- --net/eth4- -net/total- ---paging-- ---system--
usr sys idl wai hiq siq| read  writ| recv  send: recv  send: recv  send: recv  send|  in   out | int   csw
 11   5  81   0   0   2| 329M   20M|   0     0 :   0     0 :   0     0 :   0     0 |   0     0 |  60k   25k
 54  30   3   4   0  10|  2[illegible]M 2148M| [illegible]06M   20M:1893M 1775M: 234k  230k:2500M 1796M|   0     0 | 228k   72k
 54  30   2   3   0  11|  10M 2063M| 704M  610k:1955M 1915M: 206k  197k:2659M 1915M|   0     0 | 239k   64k
 57  31   0   1   0  11|  10M 19[illegible]M| 767M  [illegible]M:19[illegible]M 2083M: [illegible]8k  134k:2729M 2094M|   0     0 | 244k   41k
 57  32   0   1   0  11|  20M 2106M| 655M  585k:1966M 1884M: 138k  132k:2621M 1885M|   0     0 | 235k   38k
 57  32   0   0   0  11|  18M 2126M| 696M  605k:2062M 1818M: 133k  133k:2758M 1819M|   0     0 | 237k   29k
 56  33   0   0   0  11|  12M 2049M| 846M 7956k:1784M 2331M: 164k  151k:2631M 2339M|   0     0 | 241k   31k
```

4. This server was doing disk writes of 2148M and handling 750M of client traffic while generating around 2000M of cluster traffic.

5. Logged into surviving Fabric Interconnects to check the status of the cluster and it was down.

UCS-FI-6332-B# show cluster extended-state

Cluster Id: 0x6fad1d92b17911e8-0x9db3a09351086705

Start time: Wed Sep 5 19:12:27 2018

Last election time: Thu May 23 11:07:29 2019

B: UP, PRIMARY, (Management services: INIT IN PROGRESS)

A: DOWN, INAPPLICABLE

B: memb state UP, lead state PRIMARY, mgmt services state: INVALID

A: memb state DOWN, lead state INAPPLICABLE, mgmt services state: DOWN

   heartbeat state SECONDARY_FAILED

INTERNAL NETWORK INTERFACES:

eth1, DOWN

eth2, DOWN

HA NOT READY

Management services: initialization in progress on local Fabric Interconnect

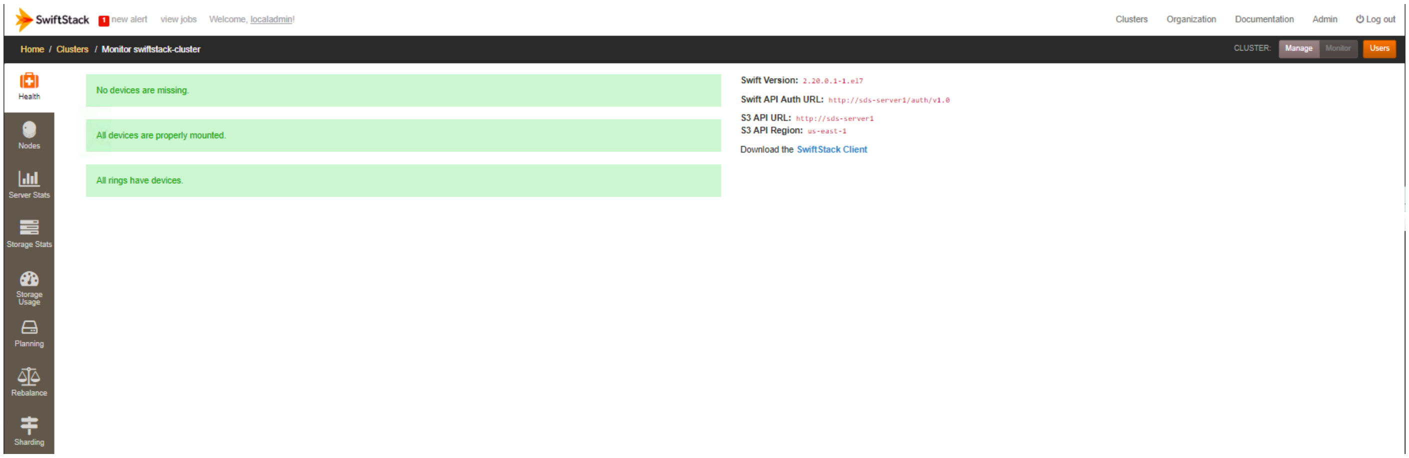
Detailed state of the device selected for HA storage:

Chassis 2, serial: FOX2036G8TW, state: active

Chassis 4, serial: FOX2034G1EC, state: active

Chassis 5, serial: FOX2036G8U6, state: active

The SwiftStack Controller does not display any errors or devices/nodes when FI is getting rebooted.



No devices are missing.

All devices are properly mounted.

All rings have devices.

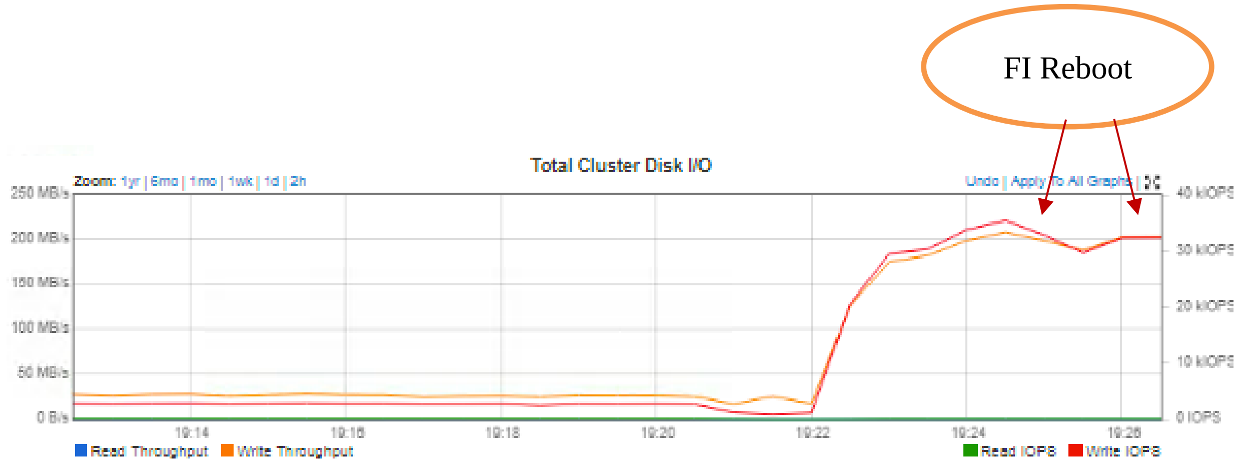
Swift Version: 2.20.0.1-1.el7

Swift API Auth URL: http://sds-server1/auth/v1.0

S3 API URL: http://sds-server1

S3 API Region: us-east-1

Download the SwiftStack Client



Summary

- No interruption of traffic was observed. A small drop in output when the entire traffic is handled by single FI.
- The drop in performance also depends on distribution of links and head room available on one Single FI to handle all of the load.

Cisco Nexus 9332 Failures

Client Workload Started on the Cluster

The following is the sequence of events:

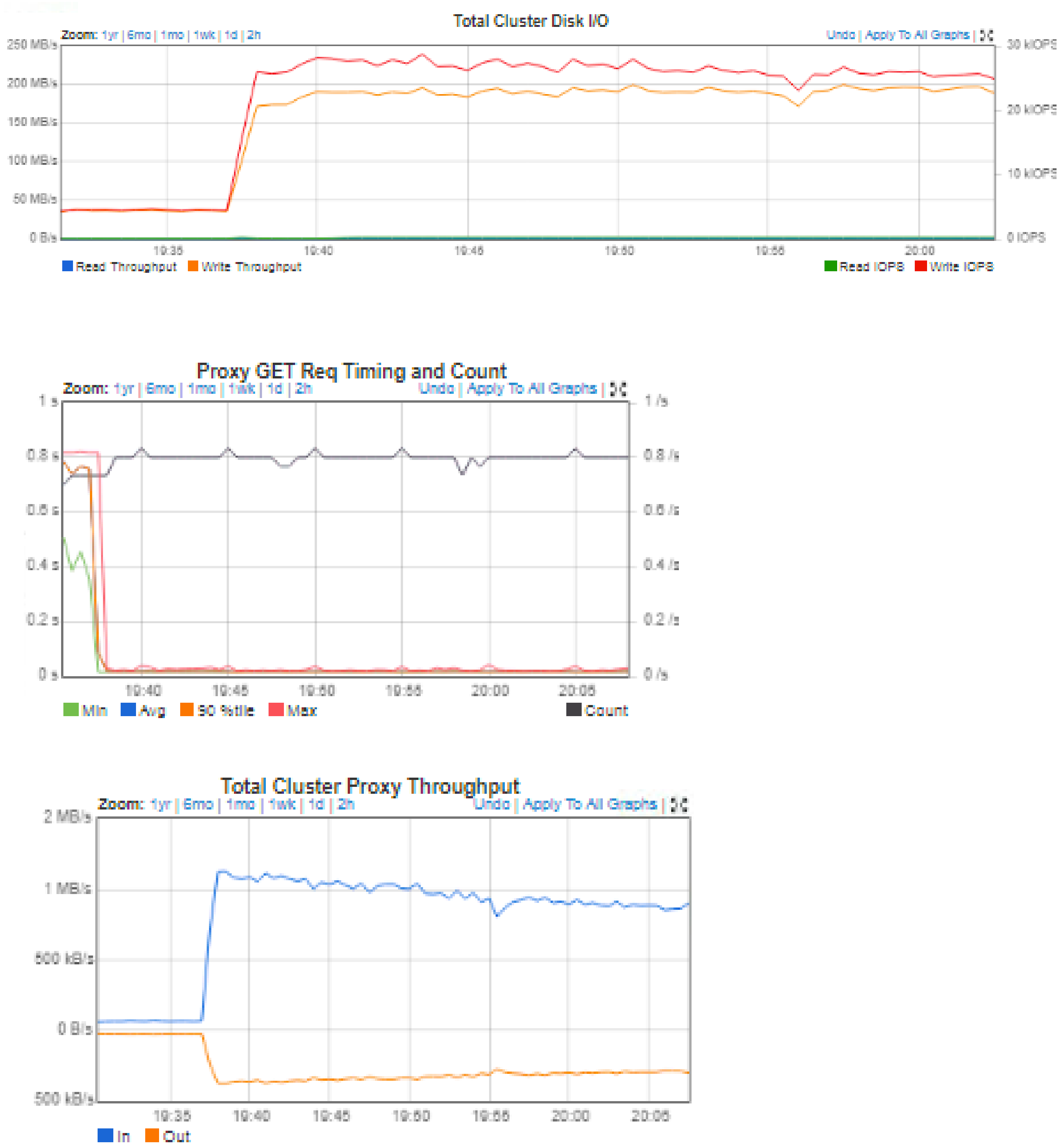
1. While cluster was running, load data gathered on IO. One of the Nexus Switches was rebooted and any changes to IO pattern was observed.

2. Cisco Nexus Switch Details:

```
BIOS: version 07.41
  NXOS: version 7.0(3)I1(3)
  BIOS compile time:  10/12/2015
  NXOS image file is: bootflash:///n9000-dk9.7.0.3.I1.3.bin
  NXOS compile time:  8/21/2015 3:00:00 [08/21/2015 10:27:18]
Hardware
  cisco Nexus9000 C9332PQ chassis
  Intel(R) Core(TM) i3- CPU @  with 16402540 kB of memory.
  Processor Board ID SAL2025S8AC

  Device name: N9K-Fab-B
  bootflash:   51496280 kB
Kernel uptime is 114 day(s), 20 hour(s), 1 minute(s), 34 second(s)
Last reset at 100047 usecs after  Mon Jan 28 22:25:31 2019
  Reason: Reset Requested by CLI command reload
  System version: 7.0(3)I1(3)
  Service:
plugin
  Core Plugin, Ethernet Plugin
Active Packages:
N9K-Fab-B#
```

Make sure that running config is copied as startup configuration and then issue a reboot of the switch.

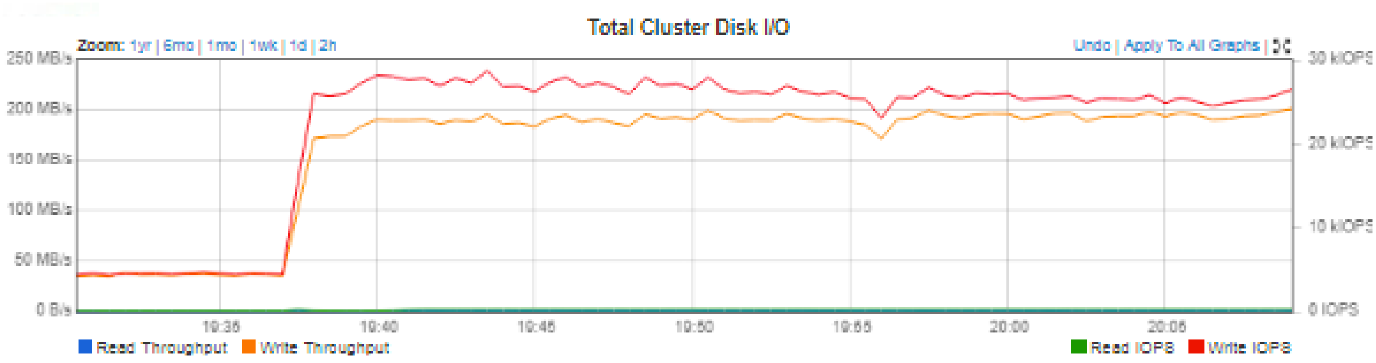
Figure 53 Before Fault Injection



Switch B was reloaded

```
N9K-Fab-B# show clock
18:29:43.201 UTC Thu May 23 2019
N9K-Fab-B# reload
This command will reboot the system. (y/n)?  [n] y

Rebooted around 10-15 minutes Nexus switch time

Hardware
  cisco Nexus9000 C9332PQ chassis
  Intel(R) Core(TM) i3- CPU @  with 16402540 kB of memory.
  Processor Board ID SAL2025S8AC

  Device name: N9K-Fab-B
  bootflash:   51496280 kB
Kernel uptime is 0 day(s), 0 hour(s), 3 minute(s), 44 second(s)

Last reset at 808619 usecs after  Thu May 23 18:31:08 2019

  Reason: Reset Requested by CLI command reload
  System version: 7.0(3)I1(3)
  Service:

plugin
  Core Plugin, Ethernet Plugin

Active Packages:
N9K-Fab-B#
```

During the entrie time of reload, no interruption of traffic was observed.



Summary

Port Channels were configured on the Nexus Switches as mentioned in the configuration section earlier. Observed Business continuity during through out the tests.

Bill of Materials

This section provides the BOM for the entire SwiftStack Storage and Cisco UCS S3260 solution.

Table 8 Bill of Materials for Cisco UCS Nexus

Component	Model	Quantity	Comments
SwiftStack Storage Nodes	Cisco UCS S3260 M5 Chassis	6	2 x UCS S3X60 M5 Server Nodes per Chassis (Total = 6nodes) Per Server Node – 2 x Intel Xeon Silver 4114 (2.2GHz/10cores), 192 GB RAM – Cisco 12G RAID Controller – 2 x SSD for OS – 28 x 10TB HDDs for Data, – 1 NVMe for metadata per node. – Dual-port 40 Gbps VIC
SwiftStack Controller Node	Cisco UCS C220 M5S Rack server	2	2 x Intel Xeon Silver 4110 (2.1GHz/8 Cores), 96GB RAM Cisco 12G SAS RAID Controller 2 x 600GB SAS for OS Dual-port 40 Gbps VIC
UCS Fabric Interconnects	Cisco UCS 6332 Fabric Interconnects	2	
Switches	Cisco Nexus 9332PQ Switches	2	

Appendix

Appendix A – Kickstart File of SwiftStack Controller Node for Cisco UCS C220 M5

Kickstart File for Controller Node

```
#version=DEVEL
#from the linux installation menu, hit tab and append this:
#biosdevname=0 net.ifnames=0 ip=eth1:dhcp
#ks=ftp://192.168.100.2/{hostname}.cfg
# System authorization information
auth --enableshadow --passalgo=sha512
# Use CDROM installation media
cdrom
# Use text install
text
# Run the Setup Agent on first boot
firstboot --disable
selinux --disable
firewall --disable
# Keyboard layouts
keyboard --vckeymap=us --xlayouts='us'
# System language
lang en_US.UTF-8
# Network information
network       --bootproto=static    --device=eth0    --ip=173.36.220.124    --
netmask=255.255.255.0       --onboot=on       --gateway=173.36.220.1        --
nameserver=171.70.168.183 --ipv6=auto --activate
network       --bootproto=static    --device=eth1   --ip=192.168.100.124    --
netmask=255.255.255.0 --onboot=on --ipv6=auto --activate

network  --hostname=swiftstack-controller

# Root password
rootpw                                                             --iscrypted
$6$yfE2jHtdy.OSmO8g$InneiVXQI9Kc9m4w2cEiS8/og6BKUlu5HSR0eCYgh5dVaeCV54Q6piS7k
10lalXignLCBvAZPqmw4dvYgy66V1
# 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
clearpart --drives=sda --all --initlabel
# Disk partitioning information
part /boot --fstype="ext4" --ondisk=sda --size=1024
part swap --fstype="swap" --ondisk=sda --size=4096
part /var --fstype="ext4" --ondisk=sda --grow
part / --fstype="ext4" --ondisk=sda --grow

reboot  --eject
```

```
%packages
@^minimal
@core
kexec-tools

%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
###############
#GPT Labels for HDDs
###############
for i in a b {d..z} aa ab ac; do parted -s /dev/sd$i mklabel gpt; done;
###############
#Turn off Transparent Hugepages and ensure that hyperthreading
#is turned off.
###############
grubby --update-kernel=ALL --args="transparent_hugepage=never numa=off";
tuned-adm profile latency-performance;
systemctl enable ntpd;
###############
#Preconfigure /etc/hosts
###############
cat >> /etc/hosts <<EOF4
192.168.100.124  swiftstack-controller
192.168.100.125  storage-node1
192.168.100.126  storage-node2
192.168.100.127  storage-node3
192.168.100.128  storage-node4
192.168.100.129  storage-node5
192.168.100.130  storage-node6
192.168.100.131  storage-node7
192.168.100.132  storage-node8
192.168.100.133  storage-node9
192.168.100.134  storage-node10
192.168.100.135  storage-node11
192.168.100.136  storage-node12

EOF4
###############
#Setup ssh keys
###############
mkdir /root/.ssh;
cat > /root/.ssh/id_rsa <<EOF5
-----BEGIN RSA PRIVATE KEY-----
```

```
MIIEpAIBAAKCAQEAsYGqxWxQdGUsiUzafYLuX6MVD3mjq3r6KaL0QcNSuZ8F3Xfw
7WJWjmhuu/rurLVoA90fjZDQY6aEAdHSH+o27mH6hfkMVqyunwQ6u3MtUqqkwRK2
NtEJqJBiHZw9+bmgofyFYl5wBSWPGIig0kb8m+cBm0uRoE5SFFuAGc7usHkfIFlO
QQd9vz9h6OX8ba3c6yUAZDzWSnt2udyLOTqV4SPpQY4O2NvYgm1VpblHvUvmP7Yu
5yl8hxn0in+RmferTq8WwyZihMV0EyN4q5HfT+gdbSY6xPMM9UHF89+lYNNxdZ4/
VuBcbBskey3UbQ332KqA7wS+Sra2DXmnfysWbwIDAQABAoIBAQCbeRFUXiyR5lP9
5lyw9k9HYRX/OfGLLumSMnJyb1wzzP9cHcPeh/V8QihLadxHVZTHXZRXcHG19pFE
7rx2y7RVU2gUlDCkchd4nEG9EYKvF1u66GLE3I7zH5Nwj/sQkfAKMZ26rTC8sUsG
mBUUWKzE+K7FkIj6ud7WidZHxKH32ok1lEcFOsH/nK1BXR29XmQ/O/Kg2h0V/KiM
1Y9CJngpghnybcDzlvpV6LS8bEiRieHJGT5RTyDk+ad0uSv+f2YtlpvSUIy7NAft
e1feAq3RWT82ZGyKTHWGTFNbfltcUjzPI/dcyS8AurYf+oQjJVAKhAl+yIn7lUrL
V6xKsdYBAoGBANwNb96gJHZUeSoOP/JCnTps+MeOhT1vyrhRRZf1IaFnEmX7hXmE
RKXaQUvGcOSPumZMkKYyqRN22B2PLM7n1D0ypKshRmk1eq6tZ/W9gkYfIdn0+QAx
AAVfUA8vJm9XLgkCAE4o2BHvtQ1w63CfygoF4V3OAsQv677F6ItROeiBAoGBAM6A
9quEOrPiRDiF25HnXXFUeRUXM4H77QB6WRV3AKggJjVlBXkhNt34g8Jr6/MfW4WO
SebQEwwBYH6NN7IG1Q0PeDRzrcv2voqzM7bV7l1rpc2E2BQhplcSyGr/aA6lW0OA
Ll/HZIdqb6OXXR8ImcP0rfxuqUJ8e6SHskG6qAbvAoGAIrw4QXMT7l3NNndDXtFn
EjbrWkzD+XuxC0FA9Aisw1aKz/BRFGptj6SRFA4B+gI6ETXay3FJwRnMaXYVQ5/S
n8pjteOtwqO/dt1GgMLmUn1NkaMavw39C9wMvijaL08apC9drvjBiqtE8Bc4AvIm
KUjeVzlStHdABkAlQgCTXIECgYEAur6BU4YWmAnsa7kRYRZ7uDsN7Ha4y7mJED+U
RAcD/wZjxzF+C5ZvybgtXyq9i3U2DMcqKaLNNrQgERGf5kyrak4tBDIAX0zZ7xAz
mgpIrw7kN8EErt/nTyLbP3eNIIGE0LwgM9lbHeKw5p3BRok+lKi2lmt0gX2VSqq0
FyC3Rt0CgYADqOJ53sV7NEXfd/NG5D9bzS5yCKW+KNH4fzxAoAYhMBo3nAkgpa/1
rdjPH4f5bAMX6dKZCh5Sy9BFxgqbI0tdjVGZBUPK8tb0xbcnJ2F3+aLq02fCfyr+
TfYW1tZ7g7gZJ+To42h4Tv9wj8iWGe+pnR4Moh3WqM1TttuaCJf1nQ==
-----END RSA PRIVATE KEY-----
EOF5
cat > /root/.ssh/id_rsa.pub <<EOF6
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCxgarFbFB0ZSyJTNp9gu5foxUPeaOrevopovRBw1K5nwXdd
/DtYlaOaG67+u6stWgD3R+NkNBjpoQB0dIf6jbuYfqF+QxWrK6fBDq7cy1SqqTBErY20QmokGIdnD
35uaCh/IViXnAFJY8YiKDSRvyb5wGbS5GgTlIUW4AZzu6weR8gWU5BB32/P2Ho5fxtrdzrJQBkPNZ
Ke3a53Is5OpXhI+lBjg7Y29iCbVWluUe9S+Y/ti7nKXyHGfSKf5GZ96tOrxbDJmKExXQTI3irkd9P
6B1tJjrE8wz1QcXz36Vg03F1nj9W4FxsGyR7LdRtDffYqoDvBL5KtrYNead/KxZv
root@storage-node7
EOF6
cat > /root/.ssh/authorized_keys <<EOF7
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCxgarFbFB0ZSyJTNp9gu5foxUPeaOrevopovRBw1K5nwXdd
/DtYlaOaG67+u6stWgD3R+NkNBjpoQB0dIf6jbuYfqF+QxWrK6fBDq7cy1SqqTBErY20QmokGIdnD
35uaCh/IViXnAFJY8YiKDSRvyb5wGbS5GgTlIUW4AZzu6weR8gWU5BB32/P2Ho5fxtrdzrJQBkPNZ
Ke3a53Is5OpXhI+lBjg7Y29iCbVWluUe9S+Y/ti7nKXyHGfSKf5GZ96tOrxbDJmKExXQTI3irkd9P
6B1tJjrE8wz1QcXz36Vg03F1nj9W4FxsGyR7LdRtDffYqoDvBL5KtrYNead/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;
###############
# Remove NetworkManager, a core package which is not needed.
yum -y remove NetworkManager;
%end
```

Appendix B – Kickstart File of Storage Nodes for Cisco UCS S3260 M5 Server

Kickstart File for Storage-node1

```
#version=DEVEL
#from the linux installation menu, hit tab and append this:
#biosdevname=0 net.ifnames=0 ip=eth1:dhcp
#ks=ftp://192.168.100.2/{hostname}.cfg
# System authorization information
auth --enableshadow --passalgo=sha512
# Use CDROM installation media
cdrom
# Use text install
text
# Run the Setup Agent on first boot
firstboot --disable
selinux --disable
firewall --disable
# Keyboard layouts
keyboard --vckeymap=us --xlayouts='us'
# System language
lang en_US.UTF-8
# Network information
network       --bootproto=static    --device=eth0    --ip=173.36.220.125    --
netmask=255.255.255.0       --onboot=on       --gateway=173.36.220.1       --
nameserver=171.70.168.183 --ipv6=auto --activate
network       --bootproto=static    --device=eth1    --ip=192.168.100.125    --
netmask=255.255.255.0 --onboot=on --ipv6=auto --activate
network       --bootproto=static    --device=eth2    --ip=192.168.130.125    --
netmask=255.255.255.0 --onboot=on --ipv6=auto --activate
network       --bootproto=static    --device=eth3    --ip=192.168.120.125    --
netmask=255.255.255.0 --onboot=on --ipv6=auto –activate
network       --bootproto=static    --device=eth4    --ip=192.168.150.125    --
netmask=255.255.255.0 --onboot=on --ipv6=auto --activate

network  --hostname=storage-node1

# Root password
rootpw                                                              --iscrypted
$6$yfE2jHtdy.OSmO8g$InneiVXQI9Kc9m4w2cEiS8/og6BKUlu5HSR0eCYgh5dVaeCV54Q6piS7k
10lalXignLCBvAZPqmw4dvYgy66V1
# 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
clearpart --drives=sda --all --initlabel
# Disk partitioning information
part /boot --fstype="ext4" --ondisk=sda --size=1024
part swap --fstype="swap" --ondisk=sda --size=4096
part /var --fstype="ext4" --ondisk=sda --grow
part / --fstype="ext4" --ondisk=sda --grow

reboot  --eject
```

```
%packages
@^minimal
@core
kexec-tools

%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
###############
#GPT Labels for HDDs
###############
for i in a b {d..z} aa ab ac; do parted -s /dev/sd$i mklabel gpt; done;
###############
#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;
###############
#Preconfigure /etc/hosts
###############
cat >> /etc/hosts <<EOF4
192.168.100.125  storage-node1
192.168.100.126  storage-node2
192.168.100.127  storage-node3
192.168.100.128  storage-node4
192.168.100.129  storage-node5
192.168.100.130  storage-node6
192.168.100.131  storage-node7
192.168.100.132  storage-node8
192.168.100.133  storage-node9
192.168.100.134  storage-node10
192.168.100.135  storage-node11
192.168.100.136  storage-node12
EOF4
###############
#Setup ssh keys
###############
mkdir /root/.ssh;
cat > /root/.ssh/id_rsa <<EOF5
-----BEGIN RSA PRIVATE KEY-----
```

```
MIIEpAIBAAKCAQEAsYGqxWxQdGUsiUzafYLuX6MVD3mjq3r6KaL0QcNSuZ8F3Xfw
7WJWjmhuu/rurLVoA90fjZDQY6aEAdHSH+o27mH6hfkMVqyunwQ6u3MtUqqkwRK2
NtEJqJBiHZw9+bmgofyFYl5wBSWPGIig0kb8m+cBm0uRoE5SFFuAGc7usHkfIFlO
QQd9vz9h6OX8ba3c6yUAZDzWSnt2udyLOTqV4SPpQY4O2NvYgm1VpblHvUvmP7Yu
5yl8hxn0in+RmferTq8WwyZihMV0EyN4q5HfT+gdbSY6xPMM9UHF89+lYNNxdZ4/
VuBcbBskey3UbQ332KqA7wS+Sra2DXmnfysWbwIDAQABAoIBAQCbeRFUXiyR5lP9
5lyw9k9HYRX/OfGLLumSMnJyb1wzzP9cHcPeh/V8QihLadxHVZTHXZRXcHG19pFE
7rx2y7RVU2gUlDCkchd4nEG9EYKvF1u66GLE3I7zH5Nwj/sQkfAKMZ26rTC8sUsG
mBUUWKzE+K7FkIj6ud7WidZHxKH32ok1lEcFOsH/nK1BXR29XmQ/O/Kg2h0V/KiM
1Y9CJngpghnybcDzlvpV6LS8bEiRieHJGT5RTyDk+ad0uSv+f2YtlpvSUIy7NAft
e1feAq3RWT82ZGyKTHWGTFNbfltcUjzPI/dcyS8AurYf+oQjJVAKhAl+yIn7lUrL
V6xKsdYBAoGBANwNb96gJHZUeSoOP/JCnTps+MeOhT1vyrhRRZf1IaFnEmX7hXmE
RKXaQUvGcOSPumZMkKYyqRN22B2PLM7n1D0ypKshRmk1eq6tZ/W9gkYfIdn0+QAx
AAVfUA8vJm9XLgkCAE4o2BHvtQ1w63CfygoF4V3OAsQv677F6ItROeiBAoGBAM6A
9quEOrPiRDiF25HnXXFUeRUXM4H77QB6WRV3AKggJjVlBXkhNt34g8Jr6/MfW4WO
SebQEwwBYH6NN7IG1Q0PeDRzrcv2voqzM7bV7l1rpc2E2BQhplcSyGr/aA6lW0OA
Ll/HZIdqb6OXXR8ImcP0rfxuqUJ8e6SHskG6qAbvAoGAIrw4QXMT7l3NNndDXtFn
EjbrWkzD+XuxC0FA9Aisw1aKz/BRFGptj6SRFA4B+gI6ETXay3FJwRnMaXYVQ5/S
n8pjteOtwqO/dt1GgMLmUn1NkaMavw39C9wMvijaL08apC9drvjBiqtE8Bc4AvIm
KUjeVzlStHdABkAlQgCTXIECgYEAur6BU4YWmAnsa7kRYRZ7uDsN7Ha4y7mJED+U
RAcD/wZjxzF+C5ZvybgtXyq9i3U2DMcqKaLNNrQgERGf5kyrak4tBDIAX0zZ7xAz
mgpIrw7kN8EErt/nTyLbP3eNIIGE0LwgM9lbHeKw5p3BRok+lKi2lmt0gX2VSqq0
FyC3Rt0CgYADqOJ53sV7NEXfd/NG5D9bzS5yCKW+KNH4fzxAoAYhMBo3nAkgpa/1
rdjPH4f5bAMX6dKZCh5Sy9BFxgqbI0tdjVGZBUPK8tb0xbcnJ2F3+aLq02fCfyr+
TfYW1tZ7g7gZJ+To42h4Tv9wj8iWGe+pnR4Moh3WqM1TttuaCJf1nQ==
-----END RSA PRIVATE KEY-----
EOF5
cat > /root/.ssh/id_rsa.pub <<EOF6
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCxgarFbFB0ZSyJTNp9gu5foxUPeaOrevopovRBw1K5nwXdd
/DtYlaOaG67+u6stWgD3R+NkNBjpoQB0dIf6jbuYfqF+QxWrK6fBDq7cy1SqqTBErY20QmokGIdnD
35uaCh/IViXnAFJY8YiKDSRvyb5wGbS5GgTlIUW4AZzu6weR8gWU5BB32/P2Ho5fxtrdzrJQBkPNZ
Ke3a53Is5OpXhI+lBjg7Y29iCbVWluUe9S+Y/ti7nKXyHGfSKf5GZ96tOrxbDJmKExXQTI3irkd9P
6B1tJjrE8wz1QcXz36Vg03F1nj9W4FxsGyR7LdRtDffYqoDvBL5KtrYNead/KxZv
root@storage-node7
EOF6
cat > /root/.ssh/authorized_keys <<EOF7
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCxgarFbFB0ZSyJTNp9gu5foxUPeaOrevopovRBw1K5nwXdd
/DtYlaOaG67+u6stWgD3R+NkNBjpoQB0dIf6jbuYfqF+QxWrK6fBDq7cy1SqqTBErY20QmokGIdnD
35uaCh/IViXnAFJY8YiKDSRvyb5wGbS5GgTlIUW4AZzu6weR8gWU5BB32/P2Ho5fxtrdzrJQBkPNZ
Ke3a53Is5OpXhI+lBjg7Y29iCbVWluUe9S+Y/ti7nKXyHGfSKf5GZ96tOrxbDJmKExXQTI3irkd9P
6B1tJjrE8wz1QcXz36Vg03F1nj9W4FxsGyR7LdRtDffYqoDvBL5KtrYNead/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;
###############
# Remove NetworkManger, a core package which is not needed.
yum -y remove NetworkManager;
%end


chmod +x /tmp/check_connectivity.sh
```

```
%end
```

Scale-up the Cluster

This section provides a description about how to scale-up the cluster.

Add Nodes to Cisco UCS

To add nodes to Cisco UCS, follow these steps:

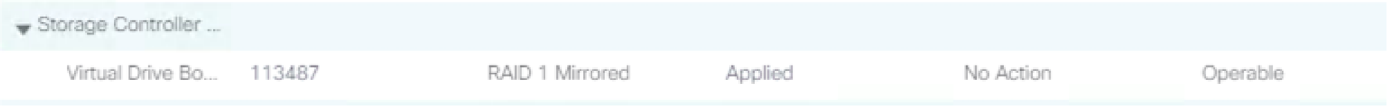
1. Make sure that the chassis and/or nodes VIC's are connected to the Fabric Interconnects. Power on the chassis. System should do auto-discovery.

2. When the Chassis is discovered, make sure that the inventory (CPU, memory, disks) are fine as expected.

3. Make sure that you have enough room available in UUID, KVM and IP pools. Create a new pool to add more IP's or UUID's.

4. Create Service Profiles from the Service Profile Templates. Make sure that the disks are in unconfigured state and apply the newly created Service Profiles.

5. After observing that virtual LUN is created in the inventory > storage tab after applying the Service Profile, install the operating system.



▾ Storage Controller ...

Virtual Drive Bo...	113487	RAID 1 Mirrored	Applied	No Action	Operable

6. As mentioned in the OS post Install section, update the linux kernel with yum and complete any other configurations mentioned in the post-install section.

The system is ready to be added as a node to the Cluster.

Add Nodes to SwiftStack Cluster

To add nodes to the SwiftStack Cluster, follow these steps:

1. Copy the ssman.crt file from Controller Node to the new Storage node.

```
scp ssman.crt root@swiftstack-new-node:/etc/pki/ca-trust/source/anchors/
```

2. Run update-ca-trust extract on the new storage node.

3. Run the Curl Command on the storage node. This will download the necessary packages and installs the software.

```
curl https://swiftcontroller.cisco.com:443/install | sudo bash
```

4. Run the ssclaimurl on a browser and claim the node.

5. In case you are using SwiftStack Load Balancer, reconfigure.

6. Ingest the nodes.

7. Click Setup of the nodes – format the disks, add SSD and HDD's to respective policies. SSD's go to Account and Container Policy, while HDD's to Standard-Replica policy.

Do not deploy the configuration yet in case you have more nodes to be added now. Complete all the nodes setup up to step 6 above, enable them and then deploy the configuration.

This completes the addition of new node(s) in the cluster. Please note that this could automatically kick in the rebalance activity. The time it takes to distribute used space to the new nodes depends on the used space in the cluster, prior to the addition. While this is a background job, it does consume some network and disk resources. There will be traffic flow through the replication nic configured on the system. This activity can speeded up or slowed down by tuning the resources in the tuning section. Please contact swiftstack support team that is optimal for your configuration.

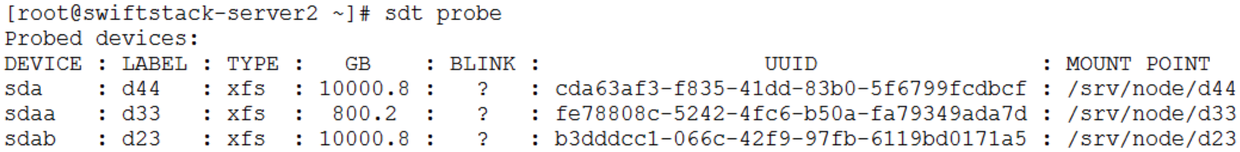
Troubleshooting

The following are some troubleshooting questions with answers:

- I am unable to claim the node. System continuously spins but never presents me with a ‘Claim Node’ button. What should I do?
  - There can be network glitches and it is fine to retry the operation. In case you are not successful, follow these steps:
    - From the controller page, delete the node. Login to the server node and run /usr/bin/uninstall-swiftstack-node. This will completely remove the software from the system. Restart by running the curl command to add this node to the controller.
- What diagnostics can I run in server node to check the status of a storage node?
  - Running /usr/bin/ssdiag will spill out any errors. A healthy node may report as shown below:

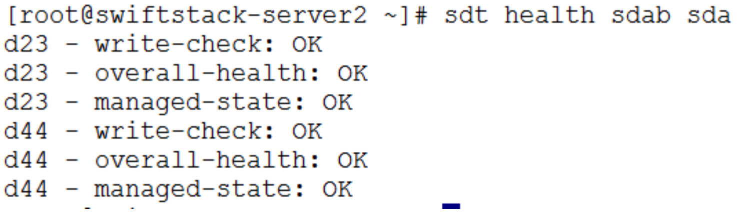
```
[root@swiftstack-server2 ~]# /usr/bin/ssdiag
SwiftStack Agent Version:         <swiftstack version>-1
Swift Package Version:            2.12.0.1-1.el7
…………
         Daemons are running: OK
        Resolve API hostname: OK:       (no cluster API hostname defined
to check)
          SwiftStack ProxyFS: OK:       (proxyfs not enabled)
      SwiftStack NAS Gateway: OK:       (gateway not enabled)
         VPN link is working: OK
      IP address(es) consistent: OK
SwiftStack Node Connectivity: OK
    SwiftStack agent version: OK
              Swift Services: OK
                 Disk Checks: OK
                 System time: OK
```

- How do I check the health of a disk in SwiftStack?
  - Log into SwiftStack Server node and run utility ‘sdt’ utility provided by SwiftStack.



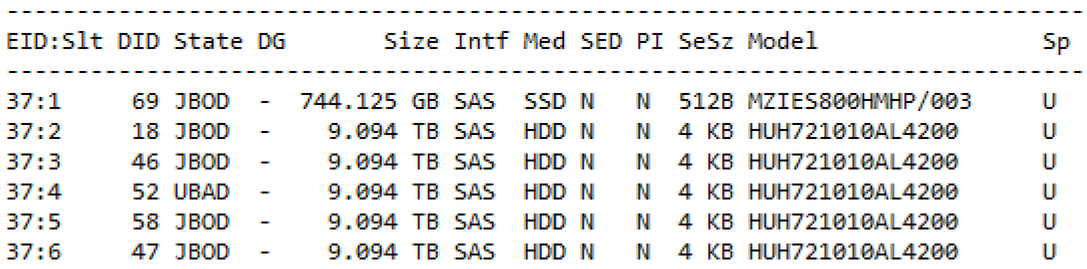
```
[root@swiftstack-server2 ~]# sdt probe
Probed devices:
DEVICE : LABEL : TYPE :   GB    : BLINK :                 UUID                 : MOUNT POINT
sda    : d44   : xfs  : 10000.8 :   ?   : cda63af3-f835-41dd-83b0-5f6799fcdbcf : /srv/node/d44
sdaa   : d33   : xfs  :  800.2  :   ?   : fe78808c-5242-4fc6-b50a-fa79349ada7d : /srv/node/d33
sdab   : d23   : xfs  : 10000.8 :   ?   : b3dddcc1-066c-42f9-97fb-6119bd0171a5 : /srv/node/d23
```

  - The status of individual disks can be queried as shown below:



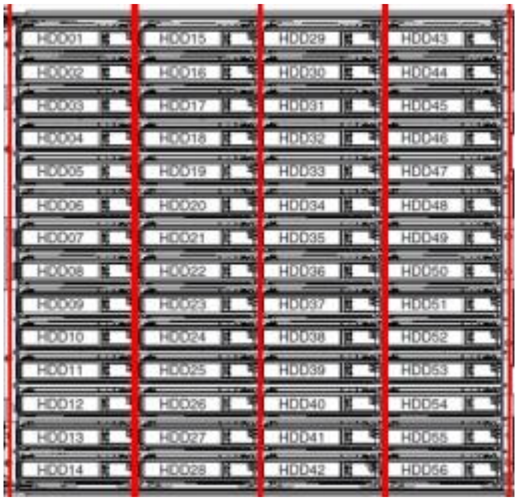
```
[root@swiftstack-server2 ~]# sdt health sdab sda
d23 - write-check: OK
d23 - overall-health: OK
d23 - managed-state: OK
d44 - write-check: OK
d44 - overall-health: OK
d44 - managed-state: OK
```

- Swiftstack shows me an alert about a bad disk that needs to be replaced. How do I replace the correct disk in Cisco UCS?
  - Go to the Managed SwiftStack Drives on the node and unmount and delete it. Note the device name as pointed out by SwiftStack – say /dev/sdc.
  - Check the existence and status in cat /proc/partitions and /dev/disk/by-*
  - You may also have an alert in Cisco UCS. Check for any missing drives in the inventory tab of the server under disks.
  - You can also run storcli to confirm the disk number which will report as back disk.



```
-----------------------------------------------------------------------------
EID:Slt DID State DG       Size Intf Med SED PI SeSz Model               Sp
-----------------------------------------------------------------------------
37:1     69 JBOD  -  744.125 GB SAS  SSD N   N  512B MZIES800HMHP/003     U
37:2     18 JBOD  -    9.094 TB SAS  HDD N   N  4 KB HUH721010AL4200      U
37:3     46 JBOD  -    9.094 TB SAS  HDD N   N  4 KB HUH721010AL4200      U
37:4     52 UBAD  -    9.094 TB SAS  HDD N   N  4 KB HUH721010AL4200      U
37:5     58 JBOD  -    9.094 TB SAS  HDD N   N  4 KB HUH721010AL4200      U
37:6     47 JBOD  -    9.094 TB SAS  HDD N   N  4 KB HUH721010AL4200      U
```

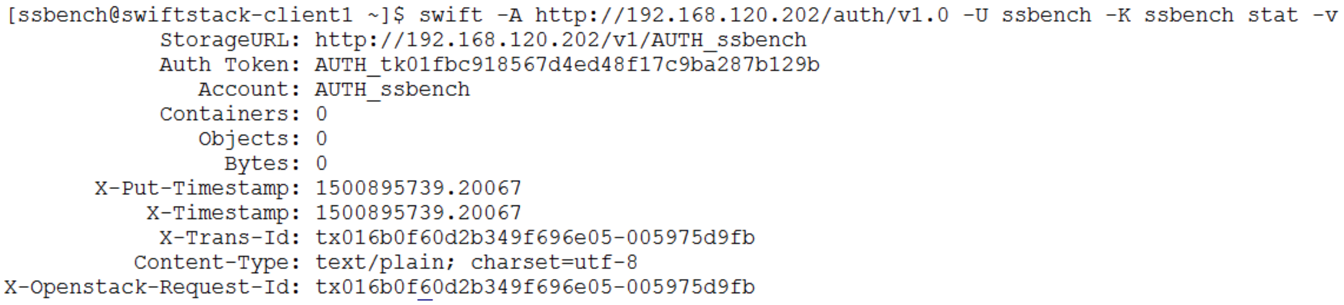
  - Check the disk number in storcli or UCS inventory and replace the disk. The disk numbers physically are numbered in 4 rows in S3260 as below. Identify the appropriate disk and replace it.



Please note that a disk LED locator can also be turned on in Cisco UCS. However, for this the power should be on and cables should be long enough when you pull the disk cabinet.

- Where to start looking for SwiftStack logs?
  - On storage node check /var/log/swift/all.log
  - On Controller node check /opt/ss/var/log/*
- How can I use Python Swift Client?

- python-swfitclient can be downloaded from github. It is handy to run few CLI commands and also for troubleshooting. Once python swiftclient is installed the parameters like authorization and users can be passed to swift and can be put in bash or .profile.

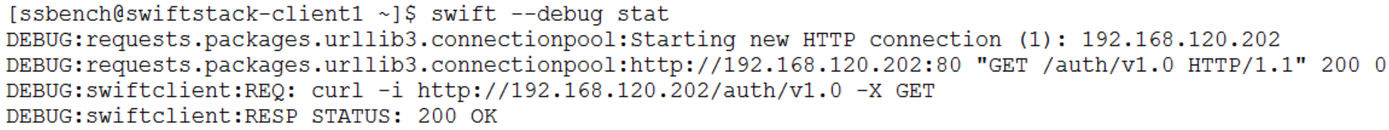


```
[ssbench@swiftstack-client1 ~]$ swift -A http://192.168.120.202/auth/v1.0 -U ssbench -K ssbench stat -v
            StorageURL: http://192.168.120.202/v1/AUTH_ssbench
            Auth Token: AUTH_tk01fbc918567d4ed48f17c9ba287b129b
               Account: AUTH_ssbench
            Containers: 0
               Objects: 0
                 Bytes: 0
       X-Put-Timestamp: 1500895739.20067
           X-Timestamp: 1500895739.20067
            X-Trans-Id: tx016b0f60d2b349f696e05-005975d9fb
          Content-Type: text/plain; charset=utf-8
X-Openstack-Request-Id: tx016b0f60d2b349f696e05-005975d9fb
```

- Alternatively make changes to your .bashrc file as below:

```
export ST_AUTH=http://192.168.120.202/auth/v1.0
export ST_USER=ssbench
export ST_KEY=ssbench
```

- Additional debug option can also be provided to the above URL



```
[ssbench@swiftstack-client1 ~]$ swift --debug stat
DEBUG:requests.packages.urllib3.connectionpool:Starting new HTTP connection (1): 192.168.120.202
DEBUG:requests.packages.urllib3.connectionpool:http://192.168.120.202:80 "GET /auth/v1.0 HTTP/1.1" 200 0
DEBUG:swiftclient:REQ: curl -i http://192.168.120.202/auth/v1.0 -X GET
DEBUG:swiftclient:RESP STATUS: 200 OK
```

- There are several Tuning parameters in Controller node. How and what to tune from these.
  - Usually the default settings should suffice. Tuning is a bit iterative and should be done carefully. All the tests done on the test bed were with default parameters. In case you think you could extract better values from the cluster, please contact swiftstack-support team.

About the Authors

Muhammad Ashfaq, Cisco Systems, Inc.

Muhammad Ashfaq is a Technical Marketing Engineer in Cisco UCS and Data Center Solutions Group. He has over 10 years of experience in IT Infrastructure, Server Virtualization, and Cloud Computing. His current role includes building Cloud Computing, Software defined Storage, Automation and Management, Converged and Hyper-Converged Solutions on Cisco UCS platforms. Muhammad also holds Cisco Internetwork Expert Data Center Certification (CCIE-DC).

Johnny Wang, SwiftStack, Inc.

Johnny has deep experience in infrastructure across storage, compute and networking. As a thought leader on how AI/ML/DL impacts infrastructure, Johnny has worked on changes needed in the datacenter, the edge, and the cloud. Johnny serves as the Solution Architect for SwiftStack.

Acknowledgements

For their support and contribution to the design, validation, and creation of this Cisco Validated Design, the authors would like to thank:

- Chris O'Brien, Cisco System, Inc.
- Jawwad Memon, Cisco Systems, Inc.
- Hiren Chandramani, SwiftStack