

# Cisco UCS C240 M5 with Scality RING

## Design and Deployment of Scality Object Storage on Cisco UCS C240 M5 Storage Server

Last Updated: Juli 30, 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.

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

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

© 2019 Cisco Systems, Inc. All rights reserved.

# Table of Contents

Executive Summary.....	7
Solution Overview .....	8
Introduction.....	8
Audience .....	8
Purpose of this Document.....	8
Solution Summary .....	8
Technology Overview .....	10
Cisco Unified Computing System .....	10
Cisco UCS C240 Rack Server.....	10
Cisco UCS Virtual Interface Card 1387 .....	12
Cisco UCS 6300 Series Fabric Interconnect .....	13
Cisco Nexus 9336C-FX2 Switch .....	14
Cisco UCS Manager .....	14
Scality RING Overview .....	16
Scality RING Architecture .....	18
RING Connectors.....	19
Storage Nodes and IO Daemons.....	19
RING Systems Management .....	20
S3 Connector: AWS S3 Storage with Identity and Access Management (IAM).....	22
Scale-Out-File-System (SOFS) .....	23
Intelligent Data Durability and Self-Healing .....	24
Replication Class of Service (COS) .....	24
Flexible Erasure Coding .....	24
Self-healing.....	25
Scality RING Multi-Site Deployments.....	25
File System (SOFS) Multi-Site Geo-Distribution.....	26
S3 Object Multi-Site Geo-Distribution .....	27
Solution Design .....	29
Solution Overview .....	29
General Hardware Requirements .....	30
Compute Layer Design.....	30
Cisco UCS Server Connectivity to Unified Fabric .....	30
Software Distributions and Versions .....	33
Deployment Hardware and Software.....	34
Fabric Configuration .....	34
Configure Cisco Nexus C9336C-FX2 Switch A and B.....	34

Initial Setup of Cisco Nexus C9336C-FX2 Switch A and B.....	34
Enable Features on Cisco Nexus C9336C-FX2 Switch A and B.....	37
Configure VLANs on Nexus C9336C-FX2 Switch A and B.....	38
Configure vPC Domain on Nexus C9336C-FX2 Switch A and B .....	39
Configure Network Interfaces for vPC Peer Links on Nexus C9336C-FX2 Switch A and B.....	40
Configure Network Interfaces to Cisco UCS FI 6332 on Nexus C9336C-FX2 Switch A and B.....	41
Verification Check of Cisco Nexus C9336C-FX2 Configuration for Switch A and B.....	44
Initial Setup of Cisco UCS 6332 Fabric Interconnects.....	46
Configure Fabric Interconnect A .....	46
Configure Fabric Interconnect B .....	49
Log Into Cisco UCS Manager .....	50
Configure NTP Server.....	50
Initial Base Setup of the Environment .....	51
Configure Global Policies.....	51
Enable Fabric Interconnect A Ports for Server .....	52
Enable Fabric Interconnect A Ports for Uplinks .....	53
Create Port Channel for Fabric Interconnect A/B.....	53
Label Each Server for Identification .....	54
Create KVM IP Pool .....	55
Create MAC Pool.....	56
Create UUID Pool .....	56
Enable CDP .....	57
QoS System Class .....	58
VLAN Setup .....	59
vNIC Template Setup .....	60
Adapter Policy Setup.....	61
LAN Connectivity Policy Setup .....	62
Boot Policy Setup.....	63
Create Maintenance Policy Setup.....	64
Create Power Control Policy Setup .....	65
Create Disk Scrub Policy.....	66
Create Host Firmware Package.....	67
Create Storage Profiles .....	68
Create Disk Group Policy for Boot Devices.....	68
Create Storage Profile .....	69
Create Service Profile Template.....	71
Create Service Profile Template .....	71
Identify Service Profile Template.....	71

Storage Provisioning .....	71
Networking .....	71
Server Boot Order .....	71
Server Maintenance .....	71
Server Assignment .....	72
Operational Policies .....	72
Create Service Profiles from Template .....	72
Associate Service Profiles .....	72
Install Red Hat Enterprise Linux 7.6 .....	73
Deployment of Scality Supervisor VM on ESXi 6.7 .....	73
Configure RHEL 7.6 for Scality Supervisor .....	81
Deploy Scality Storage-Nodes .....	82
Configure RHEL 7.6 for Scality Storage-Node .....	89
Scality RING Installation .....	93
Prerequisites .....	93
Start the Installer .....	93
Using the Installer .....	93
Prepare the Environment .....	93
Run the Pre-Install Suite .....	97
Install Scality RING .....	97
Install S3 Connector Service .....	98
Run the Post-Install Suite .....	99
Managing and Monitoring Scality RING .....	101
Monitor Scality RING .....	101
Manage NFS Connectors .....	102
Manage S3 Connectors .....	107
Scality RING Performance Testing .....	114
S3 Performance Tests .....	114
Scality RING High Availability Testing .....	115
Cisco Nexus 9336C-FX2 High Availability Testing .....	116
Sequence of Events .....	116
Cisco UCS Fabric Interconnect 6332 High Availability Testing .....	116
Sequence of Events .....	116
Scality RING Supervisor VM Failure Testing .....	117
Sequence of Events .....	117
Cisco UCS C240 M5L Disk Failure Testing .....	117
Sequence of Events .....	117
Cisco UCS C240 M5L Node Failure Testing .....	118

Sequence of Events .....	118
Appendix.....	119
Platform Description File .....	119
S3 Platform Description File .....	119
NFS Platform Description File .....	119
Summary .....	121
About the Authors .....	122
Acknowledgements.....	122



## Executive Summary

---

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

The purpose of this document is to describe the design and deployment of Scality RING on Red Hat Enterprise Linux and on the latest generation of Cisco UCS C240 Rack Servers. This validated design provides the framework of designing and deploying Scality SDS software on Cisco UCS C240 Rack Servers. The Cisco Unified Computing System provides the storage, network, and storage access components for Scality RING, deployed as a single cohesive system.

The Cisco Validated Design describes how the Cisco Unified Computing System can be used in conjunction with Scality RING 7.4. With the continuous evolution of Software Defined Storage (SDS), there has been increased demand to have small Scality RING solutions validated on Cisco UCS servers. The Cisco UCS C240 Rack Server, originally designed for the data center, together with Scality RING is optimized for such object storage solutions, making it an excellent fit for unstructured data workloads such as active archive, backup, and cloud data. The Cisco UCS C240 Rack Server delivers a complete infrastructure with exceptional scalability for computing and storage resources together with 40 Gigabit Ethernet networking.

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

## Solution Overview

---

### Introduction

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

But more and more there are requirements to store unstructured data even in smaller quantities as object storage. The advantage of identifying the data by metadata and not taking over management of the location is very attractive even for smaller quantities. As a result, new technologies need to be developed to provide similar levels of availability and reliability as large scale-out object storage solutions.

Scality RING is a storage platform that is ideal for holding large amounts of colder production data, such as backups and archives, and very large individual files, such as video files, image files, and genomic data and can also include support of warm or even hot data, by increasing CPU performance and/or memory capacity. Scality RING is highly reliable, durable, and resilient object storage for that is designed for scale and security.

Together with Cisco UCS, Scality RING can deliver a fully enterprise-ready solution that can manage different workloads and still remain flexible. The Cisco UCS C240 Rack Server is an excellent platform to use with object and file workloads, such as capacity-optimized and performance-optimized workloads. It is best suited for sequential access, as opposed to random, to unstructured data, and to any data size. It is designed for applications, not direct end-users.

This document describes the architecture, design and deployment procedures of Scality storage on Cisco UCS C240 M5 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), Cisco Nexus, and Cisco UCS Manager (UCSM) as well as a high-level understanding of Scality RING Software and its components. External references are provided where applicable and it is recommended that the reader be familiar with these documents.

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

### Purpose of this Document

This document describes the architecture, design and deployment of a Scality RING solution on Cisco UCS. It shows the simplicity of installing and configuring the shared infrastructure platform and illustrates the need of a well-conceived network architecture for low-latency, high-bandwidth.

### Solution Summary

This Cisco Validated Design (CVD) is a simple and linearly scalable architecture that provides Software Defined Storage for object and file on Scality RING 7.4 and Cisco UCS C240 rack server. This CVD describes in detail the design and deployment of Scality RING on Cisco UCS C240 rack server. The solution includes the following features:

- Infrastructure for scale-out storage



- Design of a Scality RING solution together with Cisco UCS C240 Rack Server
- Simplified infrastructure management with Cisco UCS Manager (UCSM)

The configuration uses the following architecture for the deployment:

- 3 x Cisco UCS C240 M5 Rack Servers
- 2 x Cisco UCS 6332 Fabric Interconnect
- 1 x Cisco UCS Manager
- 2 x Cisco Nexus 9336C-FX2 Switches
- Scality RING 7.4
- Red Hat Enterprise Linux 7.6

The solution has various options to scale capacity. The tested configuration uses ARC (Advanced Resiliency Configuration) 7+5 and COS 3 replication for small objects. A base capacity summary for the tested solution is listed in Table 1 . Because of the smallest Scality RING license of 200 TB usable, there is no option to use smaller drives than 10 TB.

Table 1 Usable Capacity Options for Tested Cisco Validated Design

HDD Type	Number of Disks	Usable Capacity
10 TB 7200-rpm LFF SAS drives*	36	197 TB
12 TB 7200-rpm LFF SAS drives	36	237 TB

\* Tested configuration

## Technology Overview

### Cisco Unified Computing System

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

The main components of Cisco Unified Computing System are:

- **Computing** - The system is based on an entirely new class of computing system that incorporates rack-mount and blade servers based on Intel® Xeon® Scalable processors. The Cisco UCS servers offer the patented Cisco Extended Memory Technology to support applications with large datasets and allow more virtual machines (VM) per server.
- **Network** - The system is integrated onto a low-latency, lossless, 10/25/40-Gbps unified network fabric. This network foundation consolidates LANs, SANs, and high-performance computing networks which are separate networks today. The unified fabric lowers costs by reducing the number of network adapters, switches, and cables, and by decreasing the power and cooling requirements.
- **Virtualization** - The system unleashes the full potential of virtualization by enhancing the scalability, performance, and operational control of virtual environments. Cisco security, policy enforcement, and diagnostic features are now extended into virtualized environments to better support changing business and IT requirements.
- **Storage access** - The system provides consolidated access to both SAN storage and Network Attached Storage (NAS) over the unified fabric. By unifying the storage access, the Cisco Unified Computing System can access storage over Ethernet (NFS or iSCSI), Fibre Channel, and Fibre Channel over Ethernet (FCoE). This provides customers with choice for storage access and investment protection. In addition, the server administrators can pre-assign storage-access policies for system connectivity to storage resources, simplifying storage connectivity, and management for increased productivity.

The Cisco Unified Computing System is designed to deliver:

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

### Cisco UCS C240 Rack Server

The Cisco UCS C240 Rack Server is a 2-socket, 2-Rack-Unit (2RU) rack server offering industry-leading performance and expandability. It supports a wide range of storage and I/O-intensive infrastructure workloads, from big data and analytics to collaboration. Cisco UCS C-Series Rack Servers can be deployed as standalone servers or as part of a Cisco UCS managed environment to take advantage of Cisco's standards-based unified computing innovations that help reduce customers' TCO and increase their business agility.

Figure 1 Cisco UCS C240 Rack Server



In response to ever-increasing computing and data-intensive real-time workloads, the enterprise-class Cisco UCS C240 server extends the capabilities of the Cisco UCS portfolio in a 2RU form factor. It incorporates the Intel® Xeon® Scalable processors, supporting up to 20 percent more cores per socket, twice the memory capacity, and five times more Non-Volatile Memory Express (NVMe) PCI Express (PCIe) 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. The Cisco UCS C240 M5 delivers outstanding levels of storage expandability with exceptional performance, comprised of the following:

- Latest Intel Xeon Scalable CPUs with up to 28 cores per socket
- Up to 24 DDR4 DIMMs for improved performance
- Up to 26 hot-swappable Small-Form-Factor (SFF) 2.5-inch drives, including 2 rear hot-swappable SFF drives (up to 10 support NVMe PCIe SSDs on the NVMe-optimized chassis version), or 12 Large-Form-Factor (LFF) 3.5-inch drives plus 2 rear hot-swappable SFF drives
- Support for 12-Gbps SAS modular RAID controller in a dedicated slot, leaving the remaining PCIe Generation 3.0 slots available for other expansion cards
- Modular LAN-On-Motherboard (mLOM) slot that can be used to install a Cisco UCS Virtual Interface Card (VIC) without consuming a PCIe slot, supporting dual 10-, 25- or 40-Gbps network connectivity
- Dual embedded Intel x550 10GBASE-T LAN-On-Motherboard (LOM) ports
- Modular M.2 or Secure Digital (SD) cards that can be used for boot

The Cisco UCS C240 rack server is well suited for a wide range of enterprise workloads, including:

- Object Storage
- Big Data and analytics
- Collaboration
- Small and medium-sized business databases
- Virtualization and consolidation
- Storage servers
- High-performance appliances

Cisco UCS C240 rack servers can be deployed as standalone servers or in a Cisco UCS managed environment. When used in combination with Cisco UCS Manager, the Cisco UCS C240 brings the power and automation of unified computing to enterprise applications, including Cisco® SingleConnect technology, drastically reducing switching and cabling requirements.

Cisco UCS Manager uses service profiles, templates, and policy-based management to enable rapid deployment and help ensure deployment consistency. It also enables end-to-end server visibility, management, and control in both virtualized and bare-metal environments.

The Cisco Integrated Management Controller (IMC) delivers comprehensive out-of-band server management with support for many industry standards, including:

- Redfish Version 1.01 (v1.01)
- Intelligent Platform Management Interface (IPMI) v2.0

- Simple Network Management Protocol (SNMP) v2 and v3
- Syslog
- Simple Mail Transfer Protocol (SMTP)
- Key Management Interoperability Protocol (KMIP)
- HTML5 GUI
- HTML5 virtual Keyboard, Video, and Mouse (vKVM)
- Command-Line Interface (CLI)
- XML API

Management Software Development Kits (SDKs) and DevOps integrations exist for Python, Microsoft PowerShell, Ansible, Puppet, Chef, and more. For more information about integrations, see Cisco DevNet (<https://developer.cisco.com/site/ucs-dev-center/>).

The Cisco UCS C240 is Cisco Intersight™ ready. Cisco Intersight is a new cloud-based management platform that uses analytics to deliver proactive automation and support. By combining intelligence with automated actions, you can reduce costs dramatically and resolve issues more quickly.

### Cisco UCS Virtual Interface Card 1387

The Cisco UCS Virtual Interface Card 1387 offers dual-port Enhanced Quad Small Form-Factor Pluggable (QSFP+) 40 Gigabit Ethernet and Fibre Channel over Ethernet (FCoE) in a modular-LAN-on-motherboard (mLOM) form factor. The mLOM slot can be used to install a Cisco VIC without consuming a PCIe slot providing greater I/O expandability.

Figure 2 Cisco UCS Virtual Interface Card 1387



The Cisco UCS VIC 1387 provides high network performance and low latency for the most demanding applications, including:

- Big data, high-performance computing (HPC), and high-performance trading (HPT)
- Large-scale virtual machine deployments
- High-bandwidth storage targets and archives

The card is designed for the M5 generation of Cisco UCS C-Series Rack Servers and Cisco UCS S3260 dense storage servers. It includes Cisco's next-generation converged network adapter technology and offers a comprehensive feature set, so you gain investment protection for future feature software releases.

The card can present more than 256 PCIe standards-compliant interfaces to its host. These can be dynamically configured as either network interface cards (NICs) or host bus adapters (HBAs).

This engine provides support for advanced data center requirements, including stateless network offloads for:

- Network Virtualization Using Generic Routing Encapsulation (NVGRE)
- Virtual extensible LAN (VXLAN)
- Remote direct memory access (RDMA)

The engine also offers support for performance optimization applications such as:

- Server Message Block (SMB) Direct
- Virtual Machine Queue (VMQ)
- Data Plane Development Kit (DPDK)
- Cisco NetFlow

### Cisco UCS 6300 Series Fabric Interconnect

The Cisco UCS 6300 Series Fabric Interconnects are a core part of Cisco UCS, providing both network connectivity and management capabilities for the system (Figure 3). 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 3 Cisco UCS 6300 Series Fabric Interconnect



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

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

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

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

Ethernet SFPs. The breakout feature can be configured on ports 1 to 12 and ports 15 to 26 on the Cisco UCS 6332UP fabric interconnect. Ports 17 to 34 on the Cisco UCS 6332 16-UP fabric interconnect support the breakout feature.

## Cisco Nexus 9336C-FX2 Switch

Based on [Cisco Cloud Scale technology](#), the Cisco Nexus® 9300-EX and 9300-FX platforms are the next generation of fixed Cisco Nexus 9000 Series Switches. The new platforms support cost-effective cloud-scale deployments, an increased number of endpoints, and cloud services with wire-rate security and telemetry. The platforms are built on modern system architecture designed to provide high performance and meet the evolving needs of highly scalable data centers and growing enterprises.

Figure 4 Cisco Nexus 9336C-FX2 Switch



Cisco Nexus 9300-EX and 9300-FX platform switches offer a variety of interface options to transparently migrate existing data centers from 100-Mbps, 1-Gbps, and 10-Gbps speeds to 25 Gbps at the server, and from 10- and 40-Gbps speeds to 50 and 100 Gbps at the aggregation layer. The platforms provide investment protection for customers, delivering large buffers, highly flexible Layer 2 and Layer 3 scalability, and performance to meet the changing needs of virtualized data centers and automated cloud environments.

The Cisco Nexus 9336C-FX2 Switch is a 1RU switch that supports 7.2 Tbps of bandwidth and over 2.8 bpps. The switch can be configured to work as 1/10/25/40/100-Gbps offering flexible options in a compact form factor. All ports support wire-rate MACsec encryption. Breakout is supported on all ports.

The platform hardware is capable of collecting comprehensive Cisco Tetration Analytics™ telemetry information at line rate across all the ports without adding any latency to the packets or negatively affecting switch performance. This telemetry information is exported every 100 milliseconds by default directly from the switch's Application-Specific Integrated Circuit (ASIC). This information consists of three types of data:

- **Flow information:** This information contains information about endpoints, protocols, ports, when the flow started, how long the flow was active, and so on.
- **Interpacket variation:** This information captures any interpacket variations within the flow. Examples include variation in Time To Live (TTL), IP and TCP flags, payload length, and so on.
- **Context details:** Context information is derived outside the packet header, including variation in buffer utilization, packet drops within a flow, association with tunnel endpoints, and so on.

The Cisco Tetration Analytics platform consumes this telemetry data, and by using unsupervised machine learning and behavior analysis it can provide outstanding pervasive visibility across everything in your data center in real time. By using algorithmic approaches, the Cisco Tetration Analytics platform provides a deep application insights and interactions, enabling dramatically simplified operations, a zero-trust model, and migration of applications to any programmable infrastructure. To learn more, go to <https://www.cisco.com/go/tetration>.

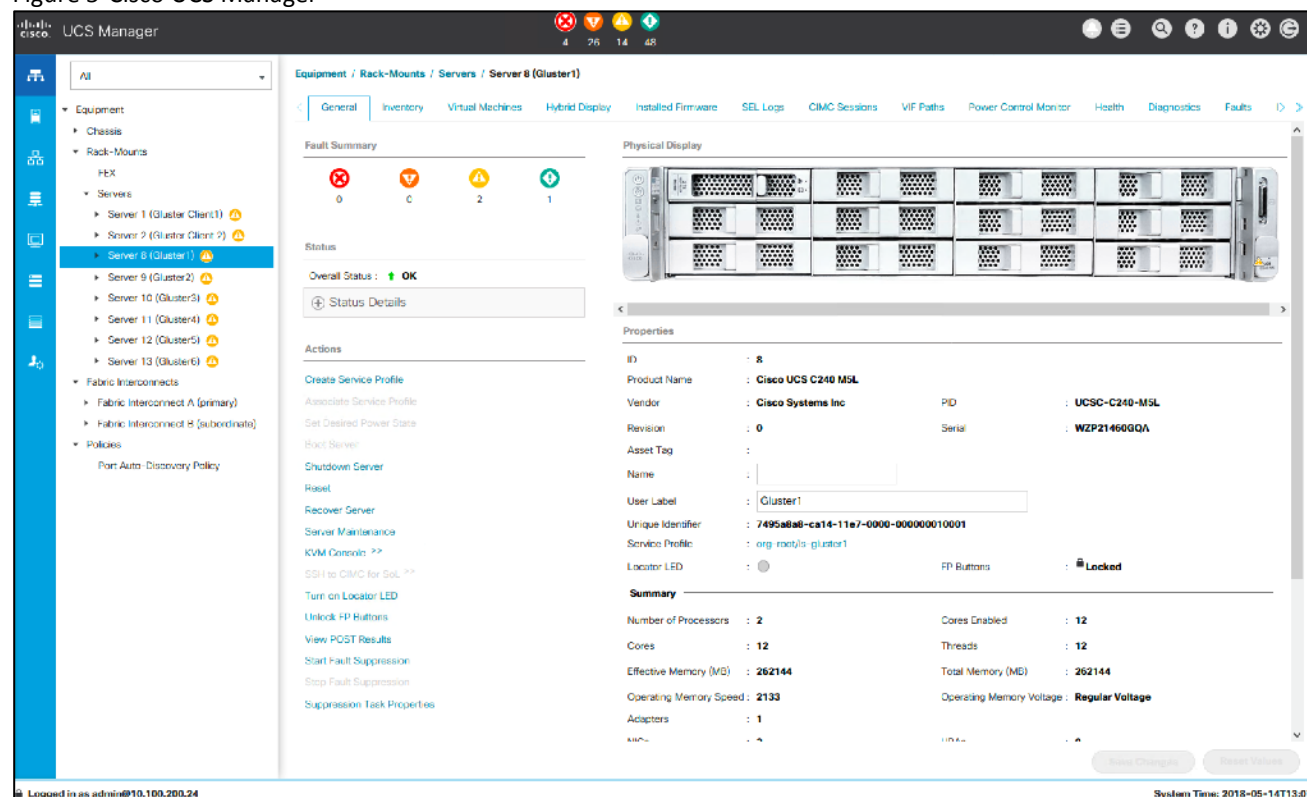
Cisco provides two modes of operation for Cisco Nexus 9000 Series Switches. Organizations can use Cisco NX OS Software to deploy the switches in standard Cisco Nexus switch environments (NX-OS mode). Organizations also can use a hardware infrastructure that is ready to support the Cisco Application Centric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

## Cisco UCS Manager

Cisco UCS® Manager (UCSM) (Figure 5) provides unified, embedded management of all software and hardware components of the Cisco Unified Computing System™ (Cisco UCS) across multiple chassis, rack servers, and thousands of

virtual machines. It supports all Cisco UCS product models, including Cisco UCS B-Series Blade Servers, C-Series Rack Servers, and S- and M-Series composable infrastructure and Cisco UCS Mini, as well as the associated storage resources and networks. Cisco UCS Manager is embedded on a pair of Cisco UCS 6300 or 6200 Series Fabric Interconnects using a clustered, active-standby configuration for high availability. The manager participates in server provisioning, device discovery, inventory, configuration, diagnostics, monitoring, fault detection, auditing, and statistics collection.

Figure 5 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).

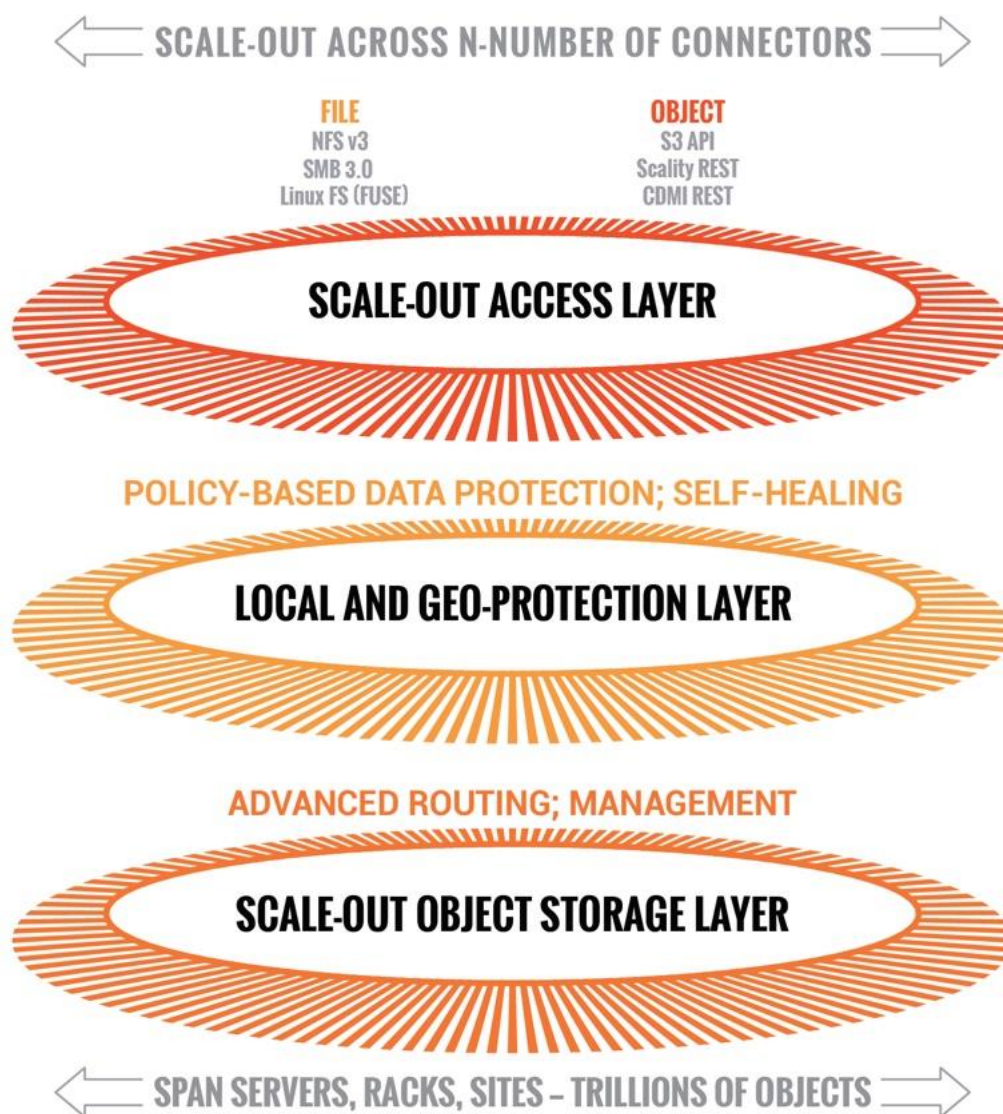
The Cisco UCS XML API provides comprehensive access to all Cisco UCS Manager functions. The API provides Cisco UCS system visibility to higher-level systems management tools from independent software vendors (ISVs) such as VMware, Microsoft, and Splunk as well as tools from BMC, CA, HP, IBM, and others. 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.

## Scality RING Overview

RING is a cloud-scale, distributed software solution for petabyte-scale unstructured data storage. It is designed to create unbounded scale-out storage systems for the many petabyte-scale applications and use cases, both object and file, that are deployed in today's enterprise data centers. RING is a fully distributed system deployed on industry standard hardware, starting with a minimum of three (3) storage servers and/or 200TB of usable capacity. It is designed to support an unbounded number of storage servers and can grow to 100's of petabytes of storage capacity. RING has no single points of failure, and requires no downtime during any upgrades, scaling, planned maintenance or unplanned system events. With self-healing capabilities, it continues operating normally throughout these events. To match performance to increasing capacity, RING can also independently scale-out its access layer of protocol Connectors, to enable an even match of aggregate performance to the application load. RING provides data protection and resiliency through local or geo-distributed erasure-coding and replication, with services for continuous self-healing to resolve expected failures in platform components such as servers and disk drives. RING is fundamentally built on a scale-out object-storage layer that employs a second-generation peer-to-peer architecture. This approach uniquely distributes both the user data and the associated metadata across the underlying nodes to eliminate the typical central metadata database bottleneck. To enable file and object data in the same system, the RING integrates a virtual file system layer through an internal NoSQL scale-out database system, which provides POSIX-based access semantics using standard NFS, SMB and FUSE protocols with shared access to the files as objects using the REST protocol.



Figure 6 Scality RING Diagram



Scality has designed RING along the design criteria spearheaded by the leading cloud-scale service providers, such as Google, Facebook, and Amazon. RING leverages loosely-coupled, distributed systems designs that leverage commodity, mainstream hardware along the following key tenets:

- 100 percent parallel design for metadata and data - to enable scaling of capacity and performance to unbounded numbers of objects, with no single points of failures, service disruptions, or forklift upgrades as the system grows.
- Multi-protocol data access – to enable the widest variety of object, file and host-based applications to leverage RING storage.
- Flexible data protection mechanisms - to efficiently and durably protect a wide range of data types and sizes.
- Self-healing from component failures – to provide high-levels of data durability, the system expects and tolerates failures and automatically resolves them.
- Hardware freedom – to provide optimal platform flexibility, eliminate lock-in and reduce TCO.

RING incorporates these design principles at multiple levels, to deliver the highest levels of data durability, at the highest levels of scale, for most optimal economics.

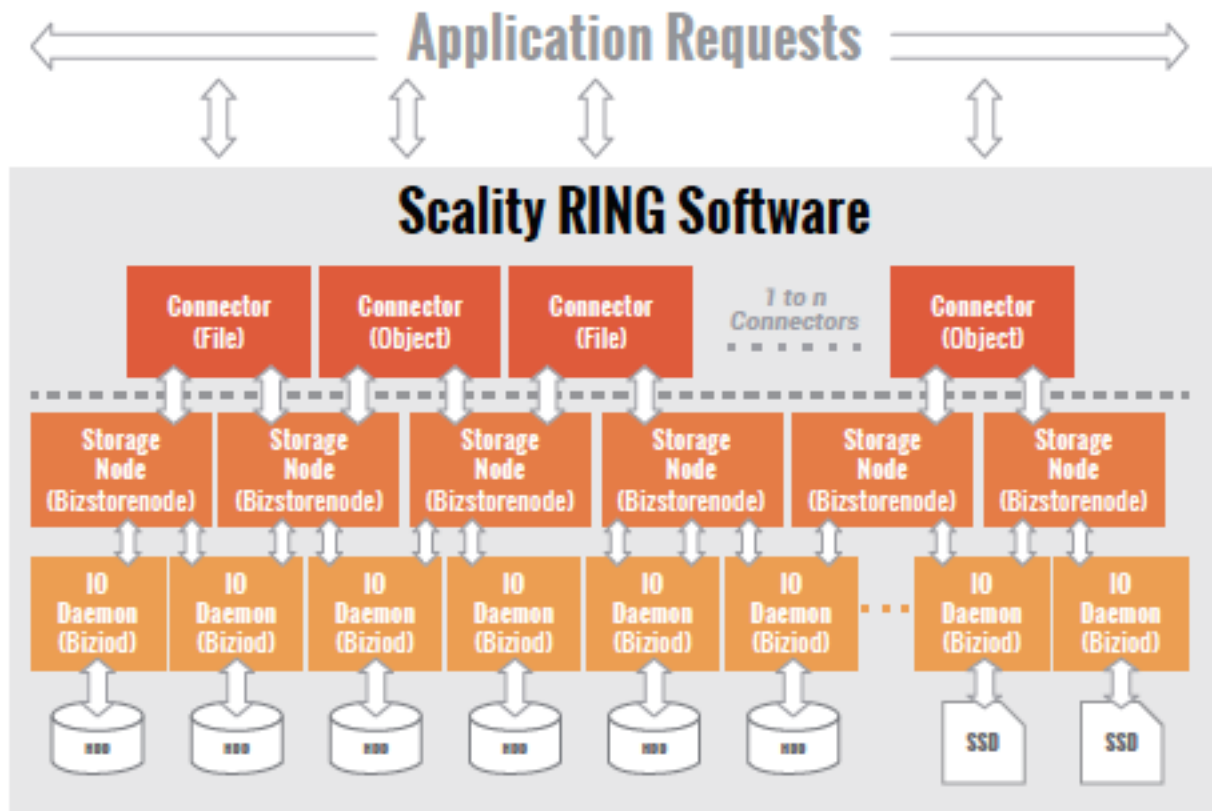
## Scality RING Architecture

To scale both storage capacity and performance to massive levels, the Scality RING software is designed as a distributed, parallel, scale-out architecture with a set of intelligent services for data access and presentation, data protection and systems management. To implement these capabilities, RING provides a set of fully abstracted software services including a top-layer of scalable access services (Connectors) that provide storage protocols for applications. The middle layers are comprised of a distributed virtual file system layer, a set of data protection mechanisms to ensure data durability and integrity, self-healing processes and a set of systems management and monitoring services. At the bottom of the stack, the system is built on a distributed storage layer comprised of virtual storage nodes and underlying IO daemons that abstract the physical storage servers and disk drive interfaces.

At the heart of the storage layer is a scalable, distributed object key/value store based on a second-generation peer-to-peer routing protocol. This routing protocol ensures that store and lookup operations scale efficiently to very high numbers of nodes.

RING software is comprised of the following main components: RING Connectors, a distributed internal NoSQL database called MESA, RING Storage Nodes and IO daemons, and the Supervisor web-based management portal. The MESA database is used to provide the Scale-Out-File-System (SOFS) file system abstraction layer, and the underlying core routing protocol and Keyspace mechanisms are described later in this paper.

Figure 7 Scality Scale-out Architecture



## RING Connectors

The Connectors provide the data access endpoints and protocol services for applications that use RING for data storage. As a scale-out system, RING supports any number of Connectors and endpoints to support large and growing application workloads. The RING 7 release provides a family of object and file interfaces:

- AWS S3 API - a comprehensive implementation of the AWS S3 REST API, with support for the Bucket and Object data model, AWS style Signature v4/v2 authentication, and the AWS model of Identity and Access Management (IAM)
- http/REST (sproxyd) - the RING's native key/value REST API, provides a flat object storage namespace with direct access to RING objects
- NFS v3 - SOFS volumes presented as a standard NFSv3 mount points
- SMB 3.0 - SOFS volumes presented as SMB Shares to Microsoft Windows clients. Scalify implements a subset of the SMB 3.0 protocol.
- FUSE - SOFS volumes presented as a local Linux file system
- CDMI/REST - support for the SNIA CDMI REST interface, with full compatibility to SOFS file data
- S3 on SOFS - SOFS volumes may be accessed in read-only mode over the S3 protocol, for namespace and data sharing between objects and files
- NFS v4/v3 on S3 - S3 buckets may be exported as NFS v4/v3 mount points

Connectors provide storage services for read, write, delete and lookup for objects or files stored into the RING based on either object or POSIX (file) semantics. Applications can make use of multiple connectors in parallel to scale out the number of operations per second, or the aggregate throughput of the RING. A RING deployment may be designed to provide a mix of file access and object access (over NFS and S3 for example), simultaneously—to support multiple application use cases.

## Storage Nodes and IO Daemons

The heart of the ring are the Storage Nodes, that are virtual processes that own and store a range of objects associated with its portion of the RING's keypace. Each physical storage server (host) is typically configured with six (6) storage node processes (termed bizstorenode). Under the storage nodes are the storage daemons (termed biziod), which are responsible for persistence of the data on disk, in an underlying local standard disk file system. Each biziod instance is a low-level software process that manages the IO operations to a particular physical disk drive and maintains the mapping of object keys to the actual object locations on disk. Biziod processes are local to a given server, managing only local, direct-attached storage and communicating only with Storage Nodes on the same server. The typical configuration is one biziod per physical disk drive, with support for up to hundreds of daemons per server, so the system can support very large, high-density storage servers.

Each biziod stores object payloads and metadata in a set of fixed size container files on the disk it is managing. With such containerization the system can maintain high-performance access even to small files, without any storage overhead. The biziod daemons typically leverage low-latency flash (SSD or NVMe) devices to store the index files for faster lookup performance. The system provides data integrity assurance and validation through the use of stored checksums on the index and data container files, which are validated upon read access to the data. The use of a standard file system underneath biziod ensures that administrators can use normal operating system utilities and tools to copy, migrate, repair and maintain the disk files if required.

The recommended deployment for systems that have both HDD and SSD media on the storage servers is to deploy a data RING on HDD, and the associated metadata in a separate RING on SSD. Typically, the requirements for metadata are approximately 2 percent of the storage capacity of the actual data, so the sizing of SSD should follow that percentage for

best effect. Scalify can provide specific sizing recommendations based on the expected average file sizes, and number of files for a given application.

## RING Systems Management

Managing and monitoring the RING is enabled through a cohesive suite of user interfaces, built on top of a family of RESTful interfaces termed the Supervisor API (SupAPI). The SupAPI provides an API based method that may be accessed from scripts, tools and frameworks for gathering statistics, metrics, health check probes and alerts, and for provisioning new services on the RING. The SupAPI is also enabled with Role Based Access Control (RBAC), by supporting an administrator identity to provide access control privileges for Super-Admin and Monitor admin user Roles.

RING provides a family of tools that use the SupAPI for accessing the same information and services. RING 7 includes the new Scalify Supervisor, a browser-based portal for both systems monitoring and management of Scalify components. In RING 7, the Supervisor now provides capabilities across object (S3) and file (NFS, SMB, FUSE) Connectors including integrated dashboards including Key Performance Indicators (KPIs) with trending information such as Global Health, Performance, Availability and Forecast. The Supervisor also includes provisioning capabilities to add new servers in the system and a new zone management module to handle customer failure domains for multi-site deployments.

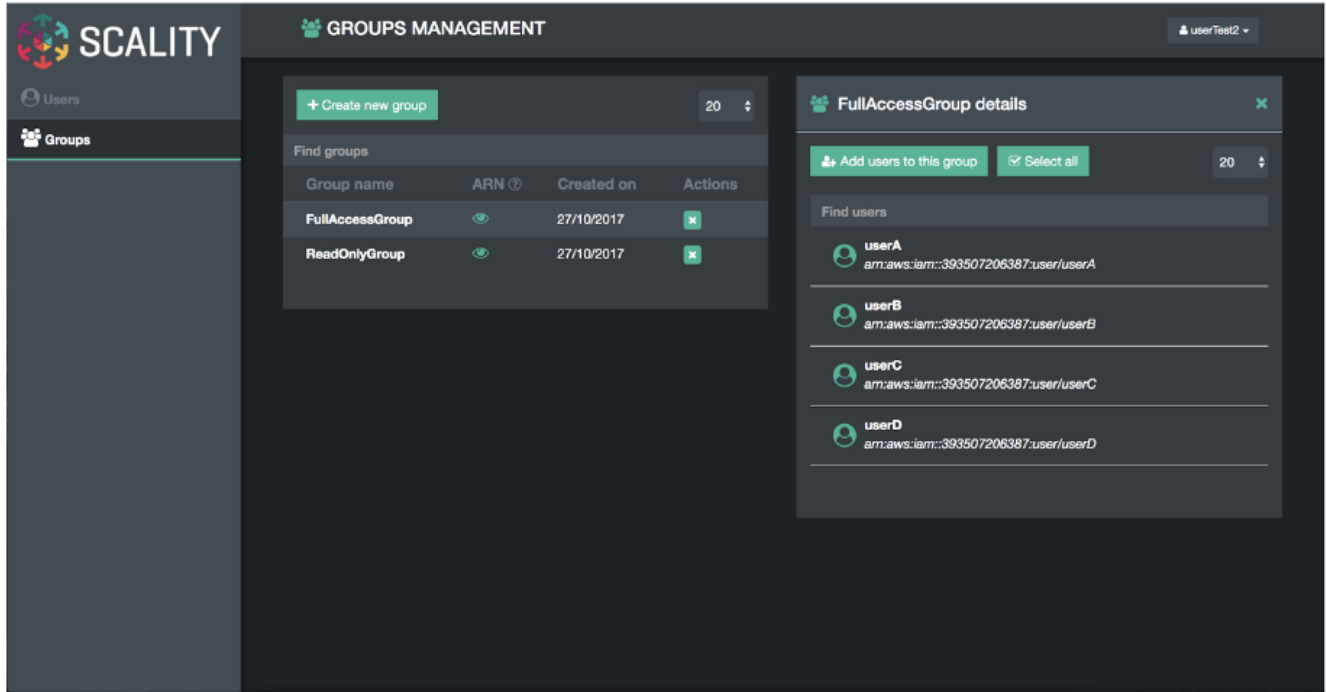
Figure 8 Supervisor Web GUI



RING Supervisor also includes an Advanced Monitoring dashboard where all collected metrics can be graphed and analyzed component per-component and per-server. This is based on a very powerful graphing engine that has access to thousands of metrics.

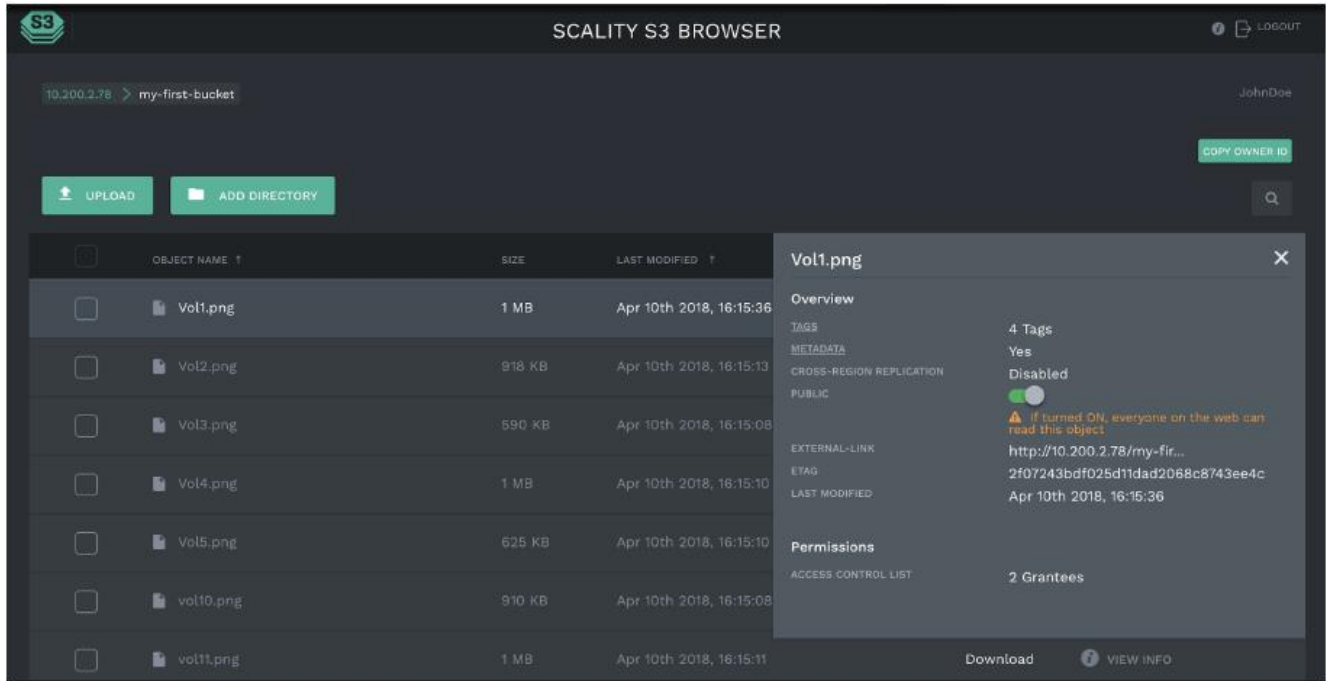
A new S3 Service Management console portal is provided to manage the integrated AWS Identity and Access Management (IAM) model of S3 multi-tenancy in the RING. This provides two-level management of Accounts, Users/Groups and IAM access control policies. The S3 Console may also be easily customized for white-labeling purposes.

Figure 9 S3 Service Management Console



A new **Scalify S3 Browser** is also provided to browse S3 buckets, upload and download object data, and for managing key S3 features such as bucket versioning, CORS, editing of metadata attributes and tagging. The S3 Browser is an S3 API client that runs on the S3 user browser and is accessible to both the Storage administrator and also to the S3 end-user.

Figure 10 Scality S3 Browser



A scriptable Command Line Interface (CLI) called RingSH is also provided, as well as an SNMP compatible MIB and traps interface for monitoring from standard SNMP consoles. RING is designed to be self-managing and autonomous to free administrators to work on other value-added tasks, and not worry about the component level management tasks common with traditional array-based storage solutions.

## S3 Connector: AWS S3 Storage with Identity and Access Management (IAM)

The Scality S3 Connector provides a modern S3 compatible application interface to the Scality RING. The AWS S3 API has now become the industry's default cloud storage API and has furthermore emerged as the standard RESTful dialect for object storage as NFS was for the NAS generation. The S3 Connector is built on a distributed scale-out architecture to support very high levels of application workloads and concurrent user access. This is based on a highly-available, high-performance metadata engine that can also be scaled-out for increased performance. Deployments can be geo-replicated to enable highly-available disaster recovery solutions, for both Metro-Area Network environments (stretched deployments), as well as Cross Region Replication (CRR) asynchronous replication of individual S3 buckets or a full site.

The Scality S3 Connector also provides a full implementation of the AWS multi-tenancy and identity management (AWS IAM) model with federated authentication to LDAP and Active Directory to integrate into enterprise deployment environments. In addition to the RING Supervisor management UI, the S3 Service Provider UI is a web-based user interface to manage multi-tenancy accounts, users, group and policies. To support enterprise security, development and operational methodologies, the S3 Connector on RING supports:

- Integration with Enterprise directory/security servers: most commonly Microsoft Active Directory or LDAP servers. Federated authentication integration is supported through a SAML 2.0-compatible Identity Provider such as Microsoft ADFS, and many other SAML compatible products, to enable a complete Single Sign-On (SSO) solution.
- Secure Multi-tenancy support: through IAM Accounts, secure access keys, Users, Groups, access control policies and v4 authentication per-tenant, bucket encryption (integrated with corporate KMS solutions) and auditing
- Utilization reporting to enable chargeback: the S3 Connector Utilization API provides an extended API for reporting on comprehensive usage metrics including capacity, #objects, bandwidth and S3 operations (per unit time). This provides all of the metrics required for consumption into corporate tools for chargeback calculations.



- High-performance, scale-out access: to support many corporate applications and workloads simultaneously reading and writing to the S3 service
- Highly-available disaster-recovery solutions: enabled deployments through multi-data center deployments to provide availability in the event of site failure

In RING 7, the feature set for the S3 Connector now supports Bucket Versioning via the S3 API, and for Cross Region Replication (CRR) of Buckets through the S3 API, this provides bucket-level asynchronous replication to another S3/RING deployment.

## Scale-Out-File-System (SOFS)

RING supports native file system access to RING storage through the integrated Scale-Out-File-System (SOFS) with NFS, SMB and FUSE Connectors for access over these well-known file protocols. SOFS is a POSIX compatible, parallel file system that provides file storage services on the RING without the need for external gateways.

SOFS is more precisely a virtual file system, which is based on an internal distributed database termed MESA (table in Spanish) on top of the RING's storage services. MESA is a distributed, semi-structured database that is used to store the file system directories and file inode structures. This provides the virtual file system hierarchical view, with the consistency required for file system data, by ensuring that file system updates are always atomic. This means that updates are either committed or rolled back entirely—which guarantees the file system is never left in an intermediate or inconsistent state. A key advantage for scaling is that MESA is itself distributed as a set of objects across all of the RING's storage node in a shared nothing manner to eliminate any bottlenecks or limitations.

File system lookups are performed using the RING's standard peer-to-peer routing protocol. For fast access performance, SOFS metadata is recommended to be stored on flash storage, typically on its own dedicated SSD drives in the storage servers, with the SOFS file payloads stored in the data RING on hard disk drives (HDDs). SOFS works directly with the data protection and durability mechanisms present in the RING, including replication and configurable Erasure Coding schemas.

SOFS can be provisioned into one or more volumes and can be scaled in capacity as needed to support application requirements. Each volume can be accessed by any number of Connectors to support the incoming load workload, even with mixed protocols (NFS, SMB or FUSE). RING can support an enormous number of volumes (up to  $2^{32}$ ) and can grow to billions of files per volume. There is no need to pre-configure volumes for capacity (the RING effectively supports thin-provisioning of volumes). Volumes will utilize the RING's storage pool to expand as needed when files are created and updated. For efficient storage of very large files, the RING supports the concept of sparse files, effectively files combined from multiple individual data-strips.

While multiple Connectors may be used to simultaneously access a volume, the RING currently supports scale-out access for multiple concurrent readers, and a new File Access Coordination mode that allows multiple readers on a file while it is being written from another Connector. This is useful in use-cases such as video streaming where very large video files are written over the course of minutes or hours, but the file must be accessed for content distribution before the write is complete. Multiple Connectors attempt to write to the same directory or one per file within a directory, SOFS maintains view consistency across multiple connectors. By supporting scale-out across any number of Connectors, SOFS throughput can be scaled out to support increasing workload demands. When performance saturation is reached, it is always possible to add more connectors or storage nodes (and disk spindles) to the RING to further increase throughput into the system. The system can achieve 10's of Gigabytes per second of aggregate throughput for parallel workloads through this architecture.

SOFS provides volume-level utilization metering and quota support, in addition to User and Group (uid/gid) quotas. This enables administrators to effectively use the concept of volumes to meter, report and limit space (capacity) usage at the volume level. This is useful in a multi-tenant environment where multiple applications or use cases are sharing the same RING, but accessing data stored in their own volume.

SOFS also provides integrated failover and load balancer services for the NFS and SMB Connectors. The load balancer uses an integrated DNS service to expose one or more service names (e.g., sofs1.companyname.com) on Virtual IP addresses

(VIPs), which can be mounted as NFS mount points or SMB shares. The load balancer can be configured with multiple underlying NFS or SMB connector real IP addresses, and provides load balancing of file traffic across these SOFS connectors. In combination with the RING 6.0 Folder Scale-out feature, this also provides transparent multi-connector access to a single folder, as well as enabling failover. In the event one of the underlying NFS or SMB Connectors becomes non-responsive, the load balancer can select another Connector IP address as the access point for the request.

## Intelligent Data Durability and Self-Healing

RING is designed to expect and manage a wide range of component failures including disks, server networks and even across multiple data centers, while ensuring that data remains durable and available during these conditions. RING provides data durability through a set of flexible data protection mechanisms optimized for distributed systems, including replication, erasure coding and geo-replication capabilities that allow applications to select the best data protection strategies for their data. These flexible data protection mechanisms implement Scality's design principle to address a wide spectrum (80 percent) of storage workloads and data sizes. A full description of multi-site data protection is provided in the next section, Multi-Site Geo-Distribution.

### Replication Class of Service (COS)

To optimize data durability in a distributed system, the RING employs local replication, or the storage of multiple copies of an object within the RING. RING will attempt to spread these replicas across multiple storage nodes, and across multiple disk drives, in order to separate them from common failures (assuming sufficient numbers of servers and disks are available). RING supports six Class-of-Service levels for replication (0-5), indicating that the system can maintain between 0 to 5 replicas (or 1-6 copies) of an object. This allows the system to tolerate up to 5 simultaneous disk failures, while still preserving access and storage of the original object. Note that any failure will cause the system to self-heal the lost replica, to automatically bring the object back up to its original Class-of-Service, as fast as possible.

While replication is optimal for many use cases where the objects are small, and access performance is critical, it does impose a high storage overhead penalty compared to the original data. For example, a 100KB object being stored with a Class-of-Service=2 (2 extra copies so 3 total), will therefore consume  $3 \times 100\text{KB} = 300\text{KB}$  of actual physical capacity on the RING, in order to maintain its 3 replicas. This overhead is acceptable in many cases for small objects but can become a costly burden for megabyte or gigabyte level video and image objects. In this case, paying a penalty of 200% to store a 1GB object since it will require 3GB of underlying raw storage capacity for its 3 replicas. When measured across petabytes of objects, this becomes a significant cost burden for many businesses, requiring a more efficient data protection mechanism.

### Flexible Erasure Coding

Scality's erasure coding (EC) provides an alternative data protection mechanism to replication that is optimized for large objects and files. RING implements Reed-Solomon erasure coding<sup>6</sup> techniques, to store large objects with an extended set of parity chunks, instead of multiple copies of the original object. The basic idea with erasure coding is to break an object into multiple chunks (m) and apply a mathematical encoding to produce an additional set of parity chunks (k). A description of the mathematical encoding is beyond the scope of this paper, but they can be simply understood as an extension of the XOR parity calculations used in traditional RAID. The resulting set of chunks, (m+k) are then distributed across the RING nodes, providing the ability to access the original object as long as any subset of m data or parity chunks are available. Stated another way, this provides a way to store an object with protection against k failures, with only k/m overhead in storage space.

Many commercial storage solutions impose a performance penalty on reading objects stored through erasure coding, due to the fact that all of the chunks, including the original data, are encoded before they are stored. This requires mandatory decoding on all access to the objects, even when there are no failure conditions on the main data chunks. With Scality's EC, the data chunks are stored in the clear, without any encoding, so that this performance penalty is not present during normal read accesses. This means that erasure coded data can be accessed as fast as other data, unless a data chunk is missing which would require a parity chunk to be accessed and decoded. In summary, for single-site data protection, Scality's



replication and erasure coded data protection mechanisms can provide very high-levels of data durability, with the ability to trade-off performance and space characteristics for different data types.

Note that replication and erasure coding may be combined, even on a single Connector, by configuring a policy for the connector to store objects below a certain size threshold with a replication CoS, but files above the file size limit with a specific erasure coding schema. This allows the application to simply store objects without worrying about the optimal storage strategy per object, with the system managing that automatically.

Note that RING does not employ traditional RAID based data protection techniques. While RAID has served the industry well in legacy NAS and SAN systems, industry experts have written at large about the inadequacies of classical RAID technologies when employed on high-density disk drives, in capacity-optimized and distributed storage systems. These deficiencies include higher probabilities of data loss due to long RAID rebuild times, and the ability to protect against only a limited set of failure conditions (for example, only two simultaneous disk failures per RAID6 group). Further information and reading on the limitations of RAID as a data protection mechanism on high-capacity disk drives is widely available

## Self-healing

RING provides self-healing processes that monitor and automatically resolve component failures. This includes the ability to rebuild missing data chunks due to disk drive or server failures, rebalance data when nodes leave and join the RING, and to proxy requests around component failures. In the event a disk drive or even a full server fails, background rebuild operations are spawned to restore the missing object data from its surviving replicas or erasure coded chunks. The rebuild process completes when it has restored the original Class of Service - either the full number of replicas or the original number of erasure coded data and parity chunks. A local disk failure can also be repaired quickly on a node (distinct from a full distributed rebuild), through the use of an in-memory key map maintained on each node. Nodes are also responsible for automatically detecting mismatches in their own Keyspace, rebalancing keys and for establishing and removing proxies during node addition and departure operations. Self-healing provides the RING with the resiliency required to maintain data availability and durability in the face of the expected wide set of failure conditions, including multiple simultaneous component failures at the hardware and software process levels.

To optimize rebuilds as well as mainline IO performance during rebuilds, RING utilizes the distributed power of the entire storage pool. The parallelism of the underlying architecture pays dividends by eliminating any central bottlenecks that might otherwise limit performance or cause contention between servicing application requests, and normal background operations such as rebuilds, especially when the system is under load. To further optimize rebuild operations, the system will only repair the affected object data, not the entire set of disk blocks, as is commonly the case in RAID arrays. Rebuilds are distributed across multiple servers and disks in the system, to utilize the aggregate processing power and available IO of multiple resources in parallel, rather than serializing the rebuilds onto a single disk drive.

By leveraging the entire pool, the impact of rebuilding data stored either with replication or erasure coding is minimized since there will be relatively small degrees of overlap between disks involved in servicing data requests, and those involved in the rebuilds.

## Scality RING Multi-Site Deployments

To support multi data center deployments with site protection and complete data consistency between all sites, the RING natively supports a stretched (synchronous) deployment mode across sites. In this mode, a single logical RING is deployed across multiple data centers, with all nodes participating in the standard RING protocols as if they were local to one site.

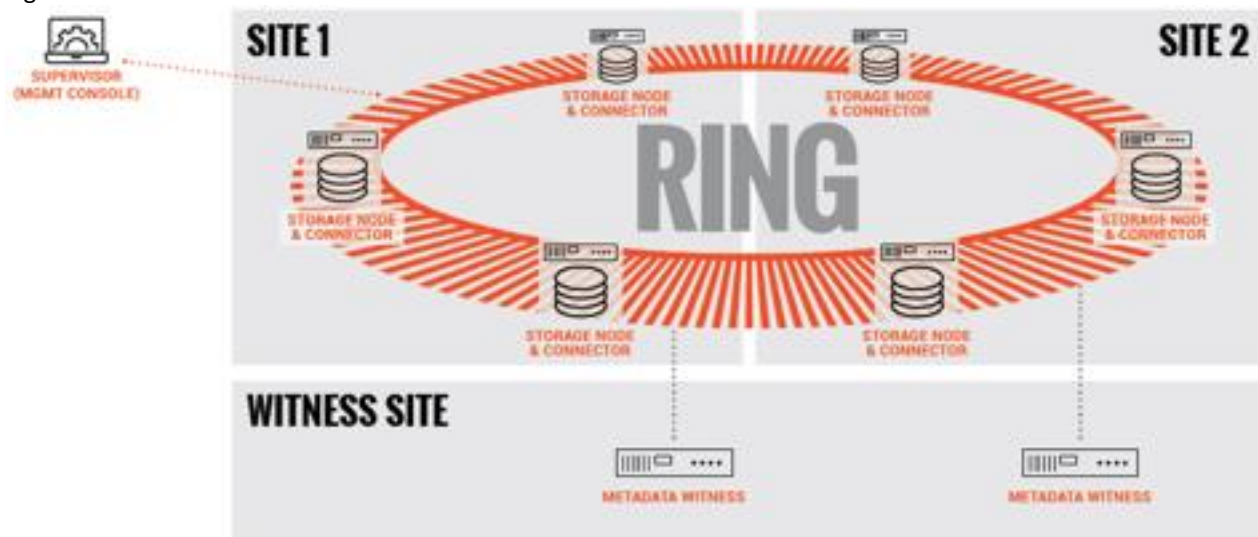
When a stretched RING is deployed with EC, it provides multiple benefits including full site-level failure protection, active/active access from both data centers, and dramatically reduced storage overhead compared to mirrored RINGs. An erasure coding schema for a three-site stretched RING of 7+5 would provide protection against one complete site failure, or up to four disk/server failures per site, plus one additional disk/server failure in another site, with approximately 70 percent space overhead. This compares favorably to a replication policy that might require 300-400 percent space overhead, for similar levels of protection across these sites.

## File System (SOFS) Multi-Site Geo-Distribution

The Scalify RING can be stretched across 2 to 3 sites within a Metro-Area Network (MAN) to provide full site failover, should the latency between the several sites go above 10ms. The stretched architecture provides zero RTO and RPO since the failover is automatized. This is the same for the failback procedure since when the lost site is recovered, the system will recover automatically the data. For the two-site stretched architecture only and to manage the mitigation between the 2 sites, 2 witness servers will be needed.

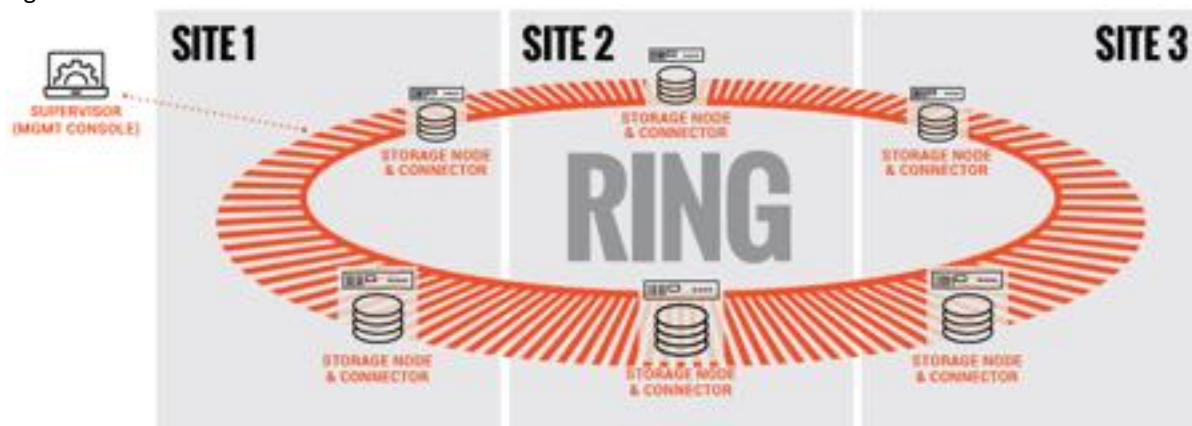
The 2 stretched sites + witness is an Active / Active replication system based on a synchronous replication.

Figure 11 SOFS – Two-Site Stretched



The 3 stretched sites is an Active / Active replication system based on a synchronous replication.

Figure 12 SOFS – Three-Site Stretched



For high latency between sites, Scalify supports SOFS 2 Sites Full Asynchronous replication mechanism at Scale to enable the replication of massive amount of file data across the 2 sites. Scalify also supports a Full Diff mechanism that can compare at scale the content of the 2 sites to ensure the data are effectively replicated. Should one site be fully lost, Scalify provides a mechanism to fully reconstruct the lost site.

To manage replication burst, Scalify integrates a back-pressure system to be sure your production network link won't be overloaded by the replication itself and in the same time will respect the RPO defined during the setup of the

installation. This feature enables the Disaster Recovery (DR) feature by providing Failover and Failback system to recover in case of partial or full loss.

The 2 sites with high latency between them is an Active / Passive replication system based on an asynchronous replication.

Figure 13 SOFS – Two-Site Asynchronous Replication

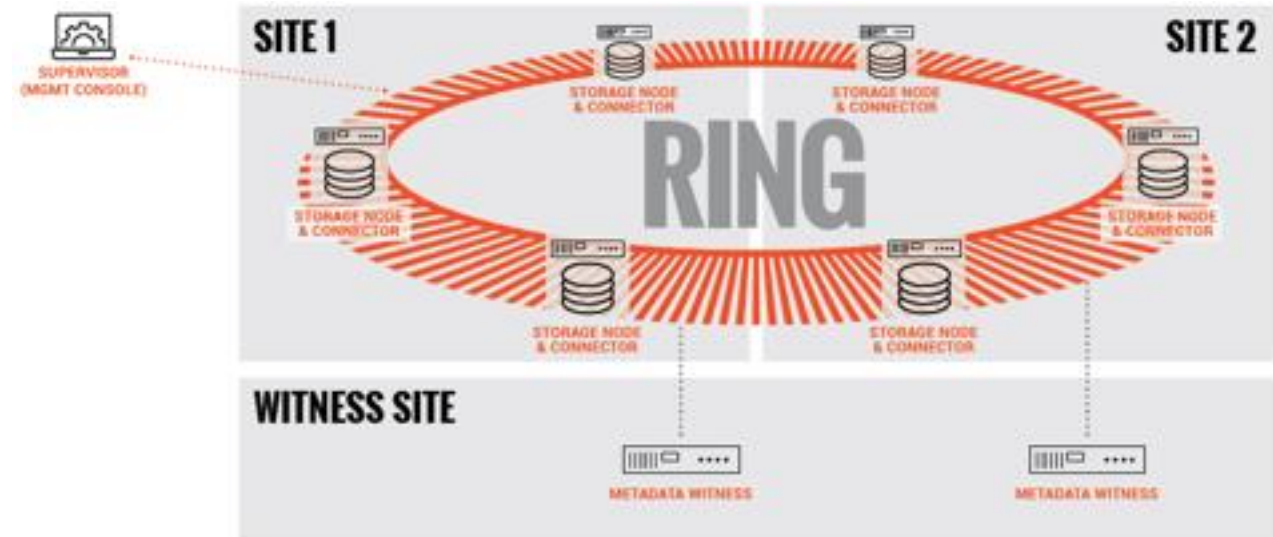


### S3 Object Multi-Site Geo-Distribution

The same multi-site architectures are supported for S3 as with SOFS, both synchronous & asynchronous. The first one with a stretched solution on two and three sites with no RPO and no RTO. As for SOFS, a stretched architecture is available within a MAN to provide full site failover. Should the latency between the several sites goes above 10ms. For the two-site stretched architecture only and to manage the mitigation between the 2 sites, 1 witness server will be needed.

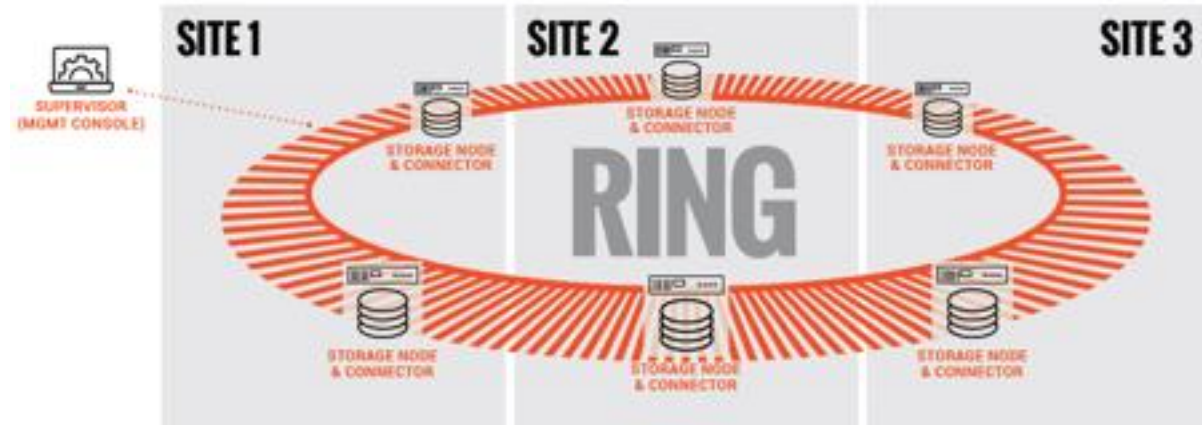
The 2 stretched sites + witness is an Active / Active replication system based on a synchronous replication.

Figure 14 S3 – Two-site Stretched



The 3 stretched sites is an Active / Active replication system based on a synchronous replication.

Figure 15 S3 – Three-Site Stretched



For high latency between sites (such as on a Wide Area Network - WAN), Scality supports the S3 2 Sites Full Asynchronous replication mechanism at Scale to enable the replication of massive amount of data across the 2 sites. This system is based on the S3 CRR design to replicate a bucket between 2 sites. For site replication, Scality developed its own system to support site replication instead of just bucket. This feature enables the Disaster Recovery (DR) feature by providing Failover and Failback system to recover in case of partial or fully (flooding, fire, and so on) lost.

The 2 sites with high latency between them is an Active / Passive replication system based on an asynchronous replication.

Figure 16 S3 – Two-Site Asynchronous Replication



## Solution Design

### Solution Overview

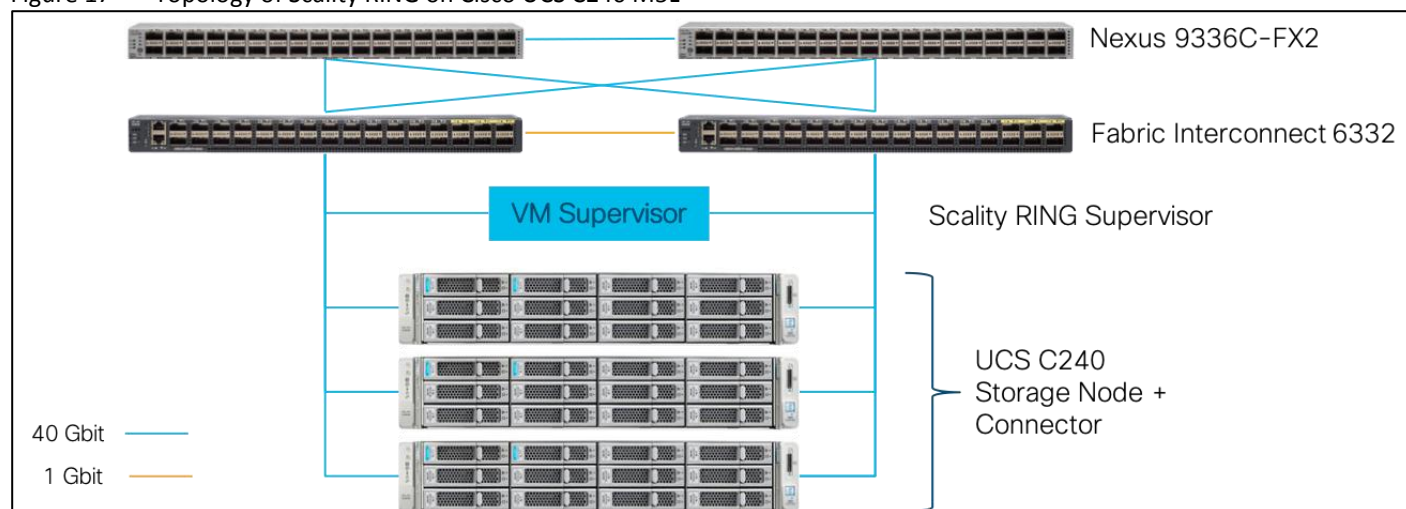
This Cisco Validated Design provides a comprehensive, end-to-end guide for designing and deploying Scality RING on Cisco UCS C240 within infrastructure made possible by Cisco UCS Manager and the Cisco UCS 6332 Fabric Interconnects.

One of the key design goals of this scale out architecture was to deploy all elements on 40GbE networking end to end within a single Cisco UCS domain and start small with Scality RING. Both Scality components – Storage Node and Connector – utilize the robust throughput and low latency only provided by the Cisco UCS 6332 Fabric Interconnect. Additionally, both components take advantage of the flexibility provided by the stateless nature of Cisco UCS service profiles and service profile templates.

This design uses the Cisco Nexus 9000 series data center switches in NX-OS standalone mode but provides investment protection to migrate to ACI or higher network bandwidths (1/10/25/40/50/100Gbps) while enabling innovative analytics and visibility using Tetration and automation that support in-box and off-box Python scripting and Open NX-OS that support dev-ops tools (Chef, Puppet, Ansible).

The key design for Scality RING on Cisco UCS C240 is shown in Figure 17.

Figure 17 Topology of Scality RING on Cisco UCS C240 M5L



- Supervisor instance deployed as virtual machine
- Connector node deployed on Storage node
- Storage node deployed on Cisco UCS C240
- Cisco UCS C240 connected to UCS 6332 Fabric Interconnect with 40Gbps line speed
- Cisco UCS 6332 Fabric Interconnect connected to Nexus 9336C-FX2 with 40Gbps line speed



For the current design of 3-node Cisco UCS C240 M5L with Scality RING there can only be one protocol active on the connector nodes. Either S3 or NFS or SMB.



## General Hardware Requirements

Table 2 List of Components

Component	Model	Quantity	Comments
Scality RING Storage/Connector Node	Cisco UCS C240	3	Per server node: <ul style="list-style-type: none"> <li>• 2 x Intel Xeon Silver 4110</li> <li>• 256 GB Memory</li> <li>• 1 x VIC 1380</li> <li>• 12 Gbit SAS RAID Controller</li> <li>• Disks               <ul style="list-style-type: none"> <li>○ 2 x SSD/HDD RAID 1 – Boot</li> <li>○ 12 x NL-SAS HDD RAID 0 – Data</li> <li>○ 2 x M.2 SSD JBOD - Metadata</li> </ul> </li> </ul>
Scality RING Supervisor	Virtual Machine	1	<ul style="list-style-type: none"> <li>• 4 vCPU</li> <li>• 16 GB Memory</li> <li>• 800 GB Disk</li> <li>• 1 x Network</li> </ul>
Cisco UCS Fabric Interconnects	Cisco UCS 6332 Fabric Interconnects	2	
Switches	Cisco Nexus 9336C-FX2	2	

## Compute Layer Design

Each Cisco UCS C240 rackmount server is equipped with a Cisco UCS Virtual Interface Card (VIC) supporting dual 40-Gbps fabric connectivity. The Cisco UCS VICs eliminate the need for separate physical interface cards on each server for data and management connectivity. For this solution with Scality RING the VIC is configured with two virtual NICs, one on each physical VIC interface.

### Cisco UCS Server Connectivity to Unified Fabric

Cisco UCS servers are typically deployed with a single VIC card for unified network and storage access. The Cisco VIC connects into a redundant unified fabric provided by a pair of Cisco UCS Fabric Interconnects. Fabric Interconnects are an integral part of the Cisco Unified Computing System, providing unified management and connectivity to all attached blades, chassis and rack servers. Fabric Interconnects provide a lossless and deterministic FCoE fabric. For the servers connected to it, the Fabric Interconnects provide LAN, SAN and management connectivity to the rest of the network.

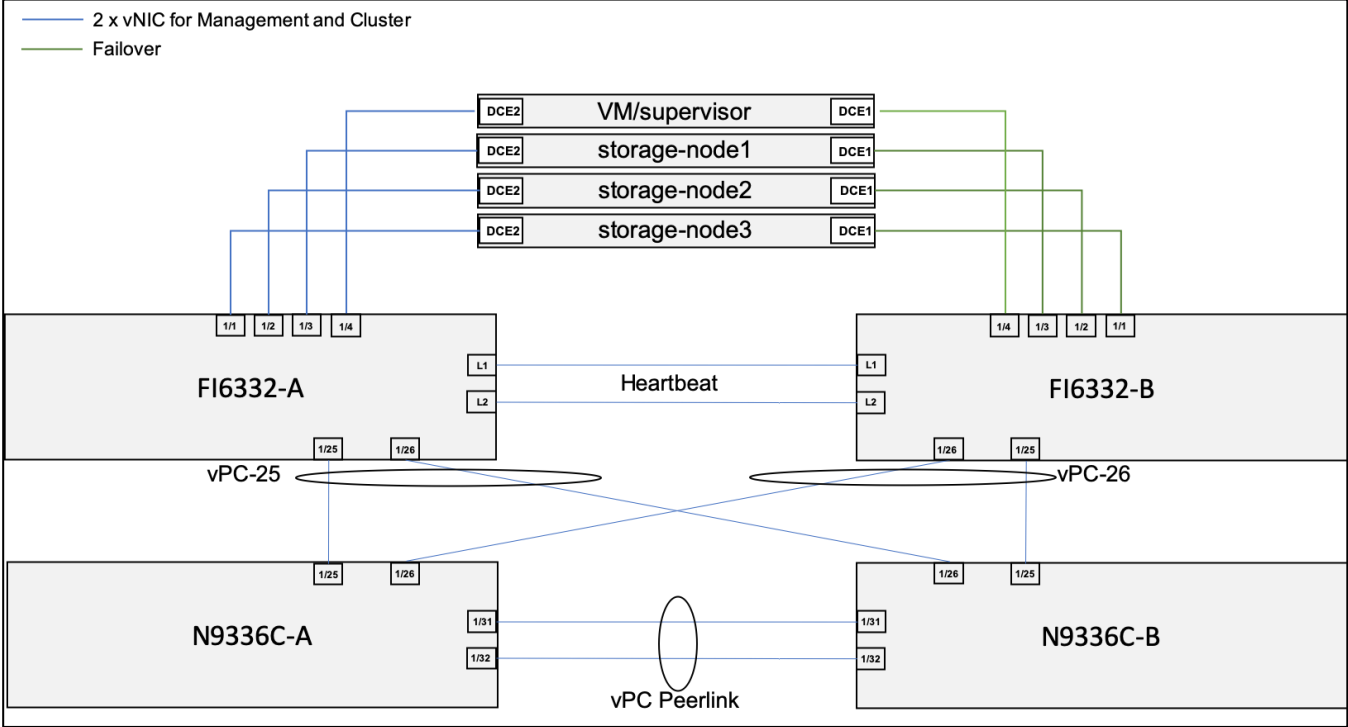
### Validated Compute Design

The connectivity of the solution is based on 40 Gbps. All components are connected together via 40 Gbps QSFP cables. Between both Cisco Nexus 9336C-FX2 switches are 2 x 40 Gbit cabling. Each Cisco UCS 6332 Fabric Interconnect is connected via 1 x 40 Gbps to each Cisco UCS Nexus 9336C-FX2 switch. And each Cisco UCS C240 M5L rack server is connected with a single 40 Gbit cable to each Fabric Interconnect.

The exact cabling for the Scality RING solution is illustrated in Figure 18.

The virtual Scality RING Supervisor node is connected to both Fabric Interconnects as well and has access to the Storage/Connector nodes.

Figure 18 Scality RING Cabling Diagram



For a better reading and overview, the exact physical connectivity between the Cisco UCS 6332 Fabric Interconnects and the Cisco UCS C-Class server is listed in Table 3 .

Table 3 Physical Connectivity between FI 6332 and Cisco UCS C240 M5L

Port	Role	FI6332-A	FI6332-B
Eth1/1	Server	storage-node1, DCE2	storage-node1, DCE1
Eth1/2	Server	storage-node2, DCE2	storage-node2, DCE1
Eth1/3	Server	storage-node3, DCE2	storage-node3, DCE1
Eth 1/4	VM	supervisor, DCE2	supervisor, DCE1
Eth1/25	Network	N9336C-A, Eth1/25	N9336C-B, Eth1/25
Eth1/26	Network	N9336C-B, Eth1/26	N9336C-A, Eth1/26

High Availability

The Cisco and Scality solution was designed for maximum availability of the complete infrastructure (compute, network, storage) with no single points of failure.

Compute

- Cisco UCS system provides redundancy at the component and link level and end-to-end path redundancy to the LAN network.
- Cisco UCS C240 rack server is highly redundant with redundant power supplies and fans.

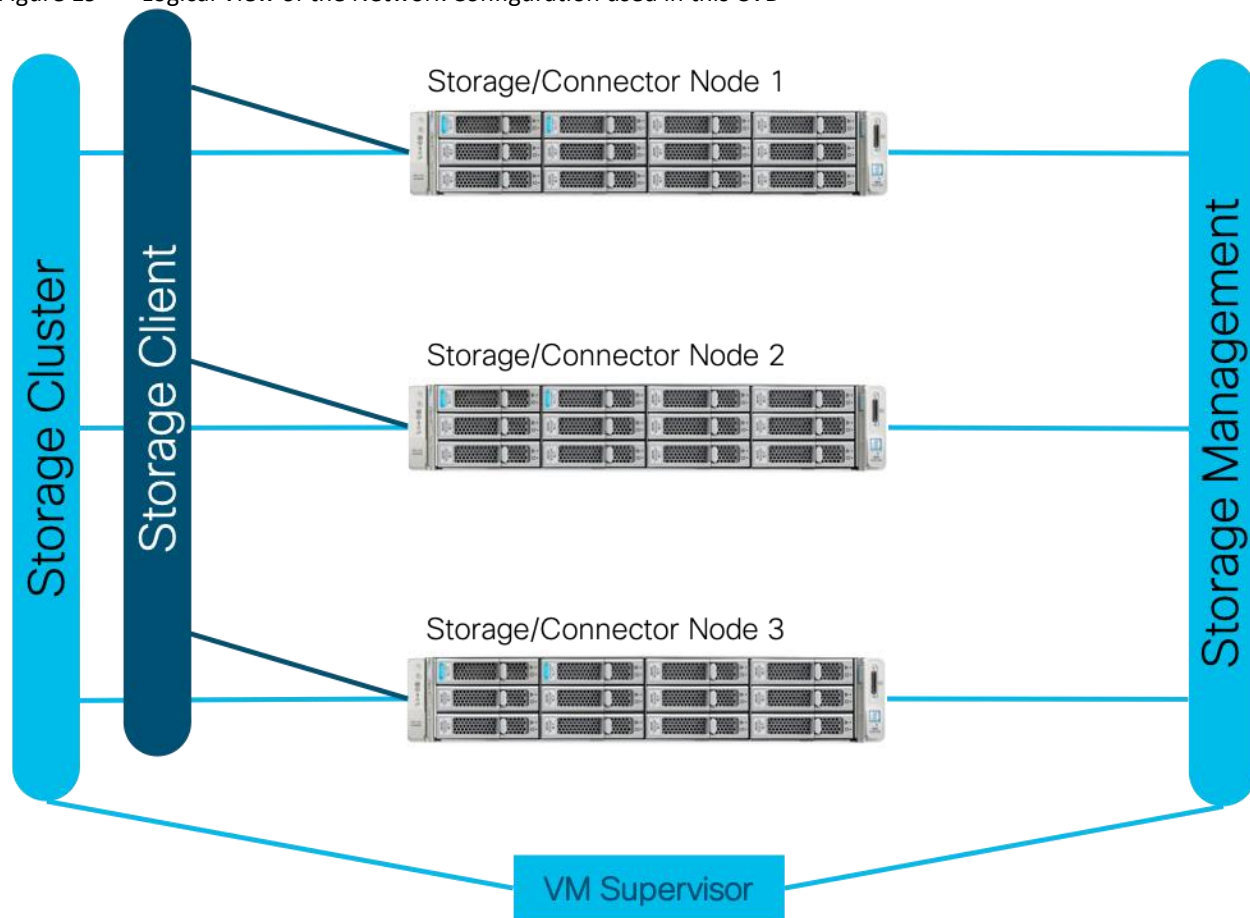
- Each server is deployed using vNICs that provide redundant connectivity to the unified fabric. NIC failover is enabled between Cisco UCS Fabric Interconnects using Cisco UCS Manager. This is done for all Storage/Connector node vNICs.

#### Network

- Link aggregation using port channels and virtual port channels can be used throughout the design for higher bandwidth and availability, if the optional Cisco UCS Nexus 9336C-FX2 is deployed. Between each Cisco UCS 6332 Fabric Interconnect and both Cisco Nexus 9336C-FX2 is one virtual Port Channel (vPC) configured. vPCs allow links that are physically connected to two different Cisco Nexus 9000 switches to appear to the Fabric Interconnect as coming from a single device and as part of a single port channel.

Figure 19 illustrates the logical configuration of the network for the Scalify RING solution.

Figure 19 Logical View of the Network Configuration used in this CVD



#### QoS and Jumbo Frames

Cisco UCS, Cisco Nexus, and Scalify RING nodes in this solution provide QoS policies and features for handling congestion and traffic spikes. The network-based QoS capabilities in these components can alleviate and provide the priority that the different traffic types require.

This design also recommends end-to-end jumbo frames with an MTU of 9000 Bytes across the LAN and Unified Fabric links. Jumbo frames increase the throughput between devices by enabling larger sized frames to be sent and received on the wire while reducing the CPU resources necessary to process them. Jumbo frames were enabled during validation on the LAN network links in the Cisco Nexus switching layer and on the Unified Fabric links.



## Software Distributions and Versions

The required software distribution versions are listed below in Table 4 .

Table 4 Software Versions

Layer	Component	Version or Release
Cisco UCS C240	Adapter	4.3(2b)
	BIOS	C240M5.4.0.2e
	Board Controller	40.0
	CIMC Controller	4.0(2f)
	Storage Controller SAS 2	50.6.0-1952
Network 6332 Fabric Interconnect	UCS Manager	4.0(2d)
	Kernel	5.0(3)N2(4.02c)
	System	5.0(3)N2(4.02c)
Network Nexus 9336C-FX2	BIOS	05.33
	NXOS	9.2(3)
Operating System	Red Hat Enterprise Linux	7.6
Software	Scality RING	7.4.2.8

## Deployment Hardware and Software

---

### Fabric Configuration

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

- Initial setup of Cisco Nexus C9336C-FX2 Switch A and B
- Initial setup of the Cisco UCS Fabric Interconnect 6332 A and B
- Connect to Cisco UCS Manager using virtual IP address of the web browser
- Launch Cisco UCS Manager
- Enable server and uplink ports
- Start discovery process
- Create pools and policies for service profile template
- Create storage profiles
- Create Service Profile templates and appropriate Service Profiles
- Associate Service Profiles to servers

### Configure Cisco Nexus C9336C-FX2 Switch A and B

Both Cisco UCS Fabric Interconnect A and B are connected to two Cisco Nexus C9336C-FX2 switches for connectivity to applications and clients. The following sections describe the setup of both Cisco Nexus C9336C-FX2 switches.

#### Initial Setup of Cisco Nexus C9336C-FX2 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. Type in L2 for interface layer.
18. Press <Return> and again <Return>.
19. Check the configuration and if correct then press <Return> and again <Return>.

The complete setup looks like the following:

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

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

```
Enter the password for "admin":
```

```
Confirm the password for "admin":
```

```
---- Basic System Configuration Dialog VDC: 1 ----
```

This setup utility will guide you through the basic configuration of the system. Setup configures only enough connectivity for management of the system.

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

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime

to skip the remaining dialogs.

```

Would you like to enter the basic configuration dialog (yes/no): yes
  Create another login account (yes/no) [n]:
  Configure read-only SNMP community string (yes/no) [n]:
  Configure read-write SNMP community string (yes/no) [n]:
  Enter the switch name : N9k-A
  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]:
    Mgmt0 IPv4 address : 192.168.10.105
    Mgmt0 IPv4 netmask : 255.255.255.0
  Configure the default gateway? (yes/no) [y]:
    IPv4 address of the default gateway : 192.168.10.234
  Configure advanced IP options? (yes/no) [n]:
  Enable the telnet service? (yes/no) [n]:
  Enable the ssh service? (yes/no) [y]:
    Type of ssh key you would like to generate (dsa/rsa) [rsa]:
    Number of rsa key bits <1024-2048> [1024]:
  Configure the ntp server? (yes/no) [n]: y
    NTP server IPv4 address : 173.38.201.115
  Configure default interface layer (L3/L2) [L3]: L2
  Configure default switchport interface state (shut/noshut) [shut]:
  Configure CoPP system profile (strict/moderate/lenient/dense) [strict]:
The following configuration will be applied:
  password strength-check
  switchname N9k-A
vrf context management
ip route 0.0.0.0/0 192.168.10.234
exit
  no feature telnet
  ssh key rsa 1024 force
  feature ssh
  ntp server 173.38.201.115

```

```

no system default switchport
system default switchport shutdown
copp profile strict
interface mgmt0
ip address 192.168.10.105 255.255.255.0
no shutdown

```

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

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

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

Copy complete.

User Access Verification

N9k-A login:



Repeat steps 1-19 for the Cisco Nexus C9336C-FX2 Switch B with the exception of configuring a different IPv4 management address in step 7.

---

## Enable Features on Cisco Nexus C9336C-FX2 Switch A and B

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

### Switch A

```

N9k-A# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-A(config)# feature udld
N9k-A(config)# feature interface-vlan
N9k-A(config)# feature lacp
N9k-A(config)# feature vpc
N9k-A(config)# system jumbomtu 9216
N9k-A(config)# spanning-tree port type edge bpduguard default
N9k-A(config)# spanning-tree port type edge bpdufilter default
N9k-A(config)# port-channel load-balance src-dst ip-l4port-vlan

```

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

```
N9k-A#
```

### Switch B

```
N9k-B# configure terminal
```

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

```
N9k-B(config)# feature udd
```

```
N9k-B(config)# feature interface-vlan
```

```
N9k-B(config)# feature lacp
```

```
N9k-B(config)# feature vpc
```

```
N9k-B(config)# system jumbomtu 9216
```

```
N9k-B(config)# spanning-tree port type edge bpduguard default
```

```
N9k-B(config)# spanning-tree port type edge bpdufilter default
```

```
N9k-B(config)# port-channel load-balance src-dst ip-l4port-vlan
```

```
N9k-B(config)# exit
```

```
N9k-B#
```

## Configure VLANs on Nexus C9336C-FX2 Switch A and B

To configure VLAN Native-VLAN and Public-VLAN, follow these steps on Switch A and Switch B:

### Switch A

```
N9k-A# config terminal
```

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

```
N9k-A(config)# vlan 2
```

```
N9k-A(config-vlan)# name Native-VLAN
```

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

```
N9k-A(config)# vlan 10
```

```
N9k-A(config-vlan)# name Management-VLAN
```

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

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

```
N9k-A(config-vlan)# name Storage-VLAN
```

```
N9k-A(config)# vlan 30
```

```
N9k-A(config-vlan)# name Client-VLAN
```

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

### Switch B

```
N9k-B# config terminal

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

N9k-B(config)# vlan 2

N9k-B(config-vlan)# name Native-VLAN

N9k-B(config-vlan)# exit

N9k-B(config)# vlan 10

N9k-B(config-vlan)# name Management-VLAN

N9k-B(config-vlan)# exit

N9k-B(config)# vlan 20

N9k-B(config-vlan)# name Storage-VLAN

N9k-B(config)# vlan 30

N9k-B(config-vlan)# name Client-VLAN

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

### Configure vPC Domain on Nexus C9336C-FX2 Switch A and B

To configure the vPC Domain, follow these steps on Switch A and Switch B:

#### Switch A

```
N9k-A# config terminal

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

N9k-A(config)# vpc domain 1

N9k-A(config-vpc-domain)# role priority 10

N9k-A(config-vpc-domain)# peer-keepalive destination 192.168.10.106 source
192.168.10.105

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

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

N9k-A(config-vpc-domain)# ip arp synchronize

N9k-A(config-vpc-domain)# auto-recovery

N9k-A(config-vpc-domain)# copy run start

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

#### Switch B

```
N9k-B# config terminal

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

```

N9k-B(config)# vpc domain 1
N9k-B(config-vpc-domain)# role priority 20
N9k-B(config-vpc-domain)# peer-keepalive destination 192.168.10.105 source
192.168.10.106
N9k-B(config-vpc-domain)# peer-switch
N9k-B(config-vpc-domain)# peer-gateway
N9k-B(config-vpc-domain)# ip arp synchronize
N9k-B(config-vpc-domain)# auto-recovery
N9k-B(config-vpc-domain)# copy run start
N9k-B(config-vpc-domain)# exit

```

### Configure Network Interfaces for vPC Peer Links on Nexus C9336C-FX2 Switch A and B

To configure the network interfaces for vPC Peer Links, follow these steps on Switch A and Switch B:

#### Switch A

```

N9k-A# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
N9k-A(config)# interface Eth 1/1
N9k-A(config-if)# description VPC Peer Nexus B Port 1/1
N9k-A(config-if)# interface Eth 1/2
N9k-A(config-if)# description VPC Peer Nexus B Port 1/2
N9k-A(config-if)# interface Eth1/1,Eth1/2
N9k-A(config-if)# channel-group 1 mode active
N9k-A(config-if)# no shutdown
N9k-A(config-if)# uddld enable
N9k-A(config-if)# interface port-channel 1
N9k-A(config-if)# description vPC peer-link
N9k-A(config-if)# switchport
N9k-A(config-if)# switchport mode trunk
N9k-A(config-if)# switchport trunk native vlan 2
N9k-A(config-if)# switchport trunk allowed vlan 10,20,30
N9k-A(config-if)# spanning-tree port type network
N9k-A(config-if)# vpc peer-link
N9k-A(config-if)# no shutdown

```



```
N9k-A(config-if)# copy run start
```

### Switch B

```
N9k-A# config terminal
```

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

```
N9k-B(config)# interface Eth 1/1
```

```
N9k-B(config-if)# description VPC Peer Nexus A Port 1/1
```

```
N9k-B(config-if)# interface Eth 1/2
```

```
N9k-B(config-if)# description VPC Peer Nexus A Port 1/2
```

```
N9k-B(config-if)# interface Eth1/1,Eth1/2
```

```
N9k-B(config-if)# channel-group 1 mode active
```

```
N9k-B(config-if)# no shutdown
```

```
N9k-B(config-if)# udld enable
```

```
N9k-B(config-if)# interface port-channel 1
```

```
N9k-B(config-if)# description vPC peer-link
```

```
N9k-B(config-if)# switchport
```

```
N9k-B(config-if)# switchport mode trunk
```

```
N9k-B(config-if)# switchport trunk native vlan 2
```

```
N9k-B(config-if)# switchport trunk allowed vlan 10,20,30
```

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

```
N9k-B(config-if)# vpc peer-link
```

```
N9k-B(config-if)# no shutdown
```

```
N9k-B(config-if)# copy run start
```

### Configure Network Interfaces to Cisco UCS FI 6332 on Nexus C9336C-FX2 Switch A and B

To configure the network interfaces to Cisco UCS FI 6332, follow these steps on Switch A and Switch B:

#### Switch A

```
N9k-A(config-if)# interface port-channel 25
```

```
N9k-A(config-if)# description Port Channel FI-A
```

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

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

```
N9k-A(config-if)# switchport trunk native vlan 2
```

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

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

```
N9k-A(config-if)# mtu 9216
N9k-A(config-if)# vpc 25
N9k-A(config-if)# no shutdown
N9k-A(config-if)# interface Eth1/25
N9k-A(config-if)# description Interface Port Channel FI-A
N9k-A(config-if)# switchport
N9k-A(config-if)# switchport mode trunk
N9k-A(config-if)# switchport trunk native vlan 2
N9k-A(config-if)# switchport trunk allowed vlan 10,20,30
N9k-A(config-if)# mtu 9216
N9k-A(config-if)# channel-group 25 mode active
N9k-A(config-if)# no shutdown

N9k-A(config-if)# interface port-channel 26
N9k-A(config-if)# description Port Channel FI-B
N9k-A(config-if)# switchport
N9k-A(config-if)# switchport mode trunk
N9k-A(config-if)# switchport trunk native vlan 2
N9k-A(config-if)# switchport trunk allowed vlan 10,20,30
N9k-A(config-if)# spanning-tree port type edge trunk
N9k-A(config-if)# mtu 9216
N9k-A(config-if)# vpc 26
N9k-A(config-if)# no shutdown
N9k-A(config-if)# interface Eth1/26
N9k-A(config-if)# description Interface Port Channel FI-B
N9k-A(config-if)# switchport
N9k-A(config-if)# switchport mode trunk
N9k-A(config-if)# switchport trunk native vlan 2
N9k-A(config-if)# switchport trunk allowed vlan 10,20,30
N9k-A(config-if)# mtu 9216
N9k-A(config-if)# channel-group 11 mode active
N9k-A(config-if)# no shutdown
```

```
N9k-A(config-if)# copy run start
```

#### Switch B

```
N9k-B(config-if)# interface port-channel 25
N9k-B(config-if)# description Port Channel FI-A
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# switchport trunk native vlan 2
N9k-B(config-if)# switchport trunk allowed vlan 10,20,30
N9k-B(config-if)# spanning-tree port type edge trunk
N9k-B(config-if)# mtu 9216
N9k-B(config-if)# vpc 25
N9k-B(config-if)# no shutdown

N9k-B(config-if)# interface Eth1/25
N9k-B(config-if)# description Interface Port Channel FI-A
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# switchport trunk native vlan 2
N9k-B(config-if)# switchport trunk allowed vlan 10,20,30
N9k-B(config-if)# mtu 9216
N9k-B(config-if)# channel-group 10 mode active
N9k-B(config-if)# no shutdown

N9k-B(config-if)# interface port-channel 26
N9k-B(config-if)# description Port Channel FI-B
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# switchport trunk native vlan 2
N9k-B(config-if)# switchport trunk allowed vlan 10,20,30
N9k-B(config-if)# spanning-tree port type edge trunk
N9k-B(config-if)# mtu 9216
N9k-B(config-if)# vpc 26
N9k-B(config-if)# no shutdown
```

```

N9k-B(config-if)# interface Eth1/26
N9k-B(config-if)# description Interface Port Channel FI-B
N9k-B(config-if)# switchport
N9k-B(config-if)# switchport mode trunk
N9k-B(config-if)# switchport trunk native vlan 2
N9k-B(config-if)# switchport trunk allowed vlan 10,20,30
N9k-B(config-if)# mtu 9216
N9k-B(config-if)# channel-group 11 mode active
N9k-B(config-if)# no shutdown
N9k-B(config-if)# copy run start

```

## Verification Check of Cisco Nexus C9336C-FX2 Configuration for Switch A and B

### Switch A

```
N9k-B# config terminal
```

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

```
N9k-A(config)# show vpc brief
```

Legend:

(\*) - local vPC is down, forwarding via vPC peer-link

```

vPC domain id                : 1
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           : Enabled, timer is off.(timeout = 240s)
Delay-restore status           : Timer is off.(timeout = 30s)
Delay-restore SVI status       : Timer is off.(timeout = 10s)

```

```
Operational Layer3 Peer-router      : Disabled
Virtual-peerlink mode               : Disabled
```

vPC Peer-link status

-----			
id	Port	Status	Active vlans
-----			
1	Po1	up	10,20,30

vPC status

-----					
Id	Port	Status	Consistency	Reason	Active vlans
-----					
25	Po25	up	success	success	10,20,30
26	Po26	up	success	success	10,20,30

Please check "show vpc consistency-parameters vpc <vpc-num>" for the consistency reason of down vpc and for type-2 consistency reasons for any vpc.

```
N9k-A(config)# show port-channel summary
```

```
Flags:  D - Down          P - Up in port-channel (members)
        I - Individual    H - Hot-standby (LACP only)
        s - Suspended     r - Module-removed
        b - BFD Session Wait
```

S - Switched      R - Routed  
 U - Up (port-channel)  
 p - Up in delay-lacp mode (member)  
 M - Not in use. Min-links not met

---

Group	Port-Channel	Type	Protocol	Member Ports
<hr/>				
1	Po1 (SU)	Eth	LACP	Eth1/1 (P)    Eth1/2 (P)
25	Po25 (SU)	Eth	LACP	Eth1/25 (P)
26	Po26 (SU)	Eth	LACP	Eth1/26 (P)

Repeat the same steps for switch Ngk-B.

The formal setup for the Cisco Nexus C9336C-FX2 switches is now finished. The next step is to configure the Cisco UCS Fabric Interconnect 6332.

## Initial Setup of Cisco UCS 6332 Fabric Interconnects

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

### Configure Fabric Interconnect A

To configure Fabric Interconnect A, 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 `FI6332` for the system name.
11. Enter the Mgmt IPv4 address.

12. Enter the Mgmt IPv4 netmask.
13. Enter the IPv4 address of the default gateway.
14. Enter the cluster IPv4 address.
15. To configure DNS, answer `y`.
16. Enter the DNS IPv4 address.
17. Answer `y` to set up the default domain name.
18. Enter the default domain name.
19. Review the settings that were printed to the console, and if they are correct, answer `yes` to save the configuration.
20. Wait for the login prompt to make sure the configuration has been saved.

#### Example Setup for Fabric Interconnect A

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

```
This setup utility will guide you through the basic configuration of
the system. Only minimal configuration including IP connectivity to
the Fabric interconnect and its clustering mode is performed through these steps.
```

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

```
To back track or make modifications to already entered values,
complete input till end of section and answer no when prompted
to apply configuration.
```

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

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

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

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

```
Enter the password for "admin":
```

```
Confirm the password for "admin":
```

```
Is this Fabric interconnect part of a cluster(select 'no' for standalone)?
(yes/no) [n]: yes
```

```
Enter the switch fabric (A/B): A
```

```
Enter the system name: FI6332
```

Physical Switch Mgmt0 IP address : 192.168.10.101

Physical Switch Mgmt0 IPv4 netmask : 255.255.255.0

IPv4 address of the default gateway : 192.168.10.234

Cluster IPv4 address : 192.168.10.100

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

DNS IP address : 208.67.222.222

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

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

Following configurations will be applied:

Switch Fabric=A

System Name=FI6332

Enforced Strong Password=no

Physical Switch Mgmt0 IP Address=192.168.10.101

Physical Switch Mgmt0 IP Netmask=255.255.255.0

Default Gateway=192.168.10.234

Ipv6 value=0

DNS Server=208.67.222.222

Cluster Enabled=yes

Cluster IP Address=192.168.10.100

NOTE: Cluster IP will be configured only after both Fabric Interconnects are initialized.

UCSM will be functional only after peer FI is configured in clustering mode.

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

Applying configuration. Please wait.

Configuration file - Ok



```
Cisco UCS 6300 Series Fabric Interconnect
FI6332-A login:
```

## Configure Fabric Interconnect B

To configure Fabric Interconnect B, follow these steps:

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

## Example Setup for Fabric Interconnect B

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

```
This setup utility will guide you through the basic configuration of
the system. Only minimal configuration including IP connectivity to
the Fabric interconnect and its clustering mode is performed through these steps.
```

```
Type Ctrl-C at any time to abort configuration and reboot system.
To back track or make modifications to already entered values,
complete input till end of section and answer no when prompted
to apply configuration.
```

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

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

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

```
Connecting to peer Fabric interconnect... done
```

```
Retrieving config from peer Fabric interconnect... done
Peer Fabric interconnect Mgmt0 IPv4 Address: 192.168.10.101
Peer Fabric interconnect Mgmt0 IPv4 Netmask: 255.255.255.0
Cluster IPv4 address           : 192.168.10.100
```

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

Physical Switch Mgmt0 IP address : 192.168.10.102

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

Applying configuration. Please wait.

Fri May 10 05:41:48 UTC 2019

Configuration file - Ok

Cisco UCS 6300 Series Fabric Interconnect

FI6332-B login:

## Log Into Cisco UCS Manager

To login to Cisco UCS Manager, follow these steps:

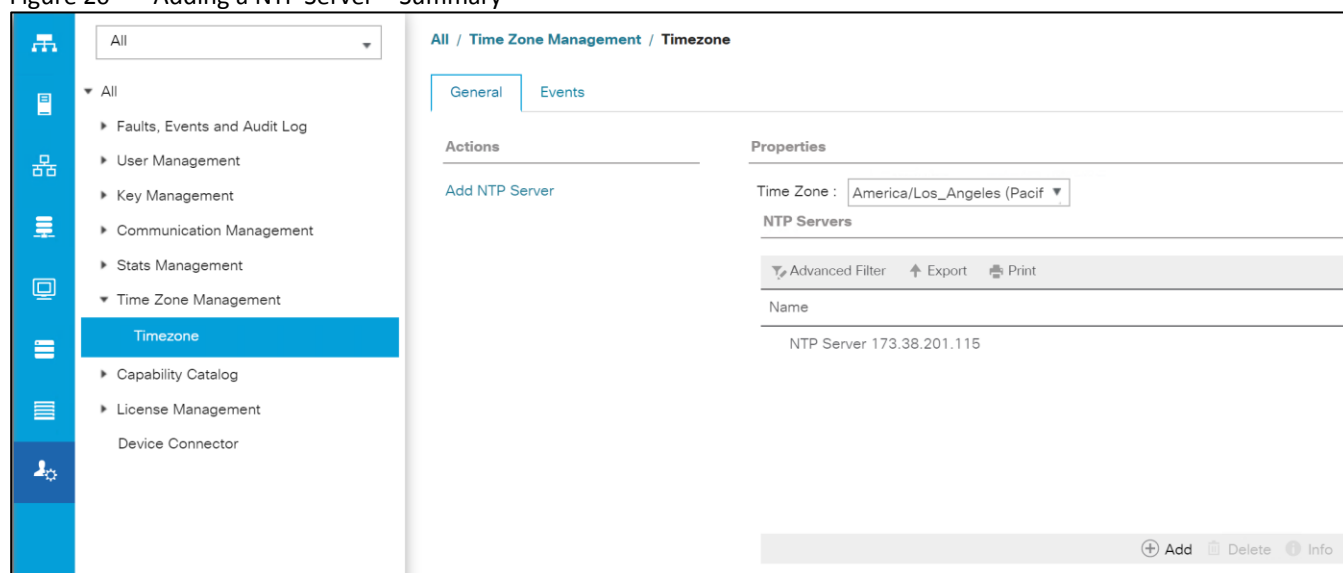
1. Open a Web browser and navigate to the Cisco UCS 6332 Fabric Interconnect cluster address.
2. Click the Launch link to download the Cisco UCS Manager software.
3. If prompted to accept security certificates, accept as necessary.
4. Click Launch UCS Manager HTML.
5. When prompted, enter admin for the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

## Configure NTP Server

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

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

Figure 20 Adding a NTP Server – Summary



## 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.
10. Select Save Changes.

Figure 21 Configuration of Global Policies

The screenshot displays the 'Equipment' configuration page with the 'Policies' tab selected. The left sidebar shows a tree view under 'Equipment' with 'Policies' expanded, listing 'Port Auto-Discovery Policy'. The main content area contains several policy sections:

- Chassis/FEX Discovery Policy**:
  - Action: Platform Max
  - Link Grouping Preference: ☒ None ☐ Port Channel
  - Backplane Speed Preference: ☒ 40G ☐ 4x10G
- Rack Server Discovery Policy**:
  - Action: ☒ Immediate ☐ User Acknowledged
  - Scrub Policy: <not set>
- Rack Management Connection Policy**:
  - Action: ☒ Auto Acknowledged ☐ User Acknowledged
- Power Policy**:
  - Redundancy: ☐ Non Redundant ☒ N+1 ☐ Grid
- MAC Address Table Aging**:
  - Aging Time: ☐ Never ☒ Mode Default ☐ other
- Global Power Allocation Policy**:
  - Allocation Method: ☐ Manual Blade Level Cap ☒ Policy Driven Chassis Group Cap
- Firmware Auto Sync Server Policy**:
  - Sync State: ☒ No Actions ☐ User Acknowledge
- Info Policy**:
  - Action: ☒ Disabled ☐ Enabled
- Global Power Profiling Policy**:
  - Profile Power: ☐
- Hardware Change Discovery Policy**:
  - Action: ☒ User Acknowledged ☐ Auto Acknowledged

## Enable Fabric Interconnect A Ports for Server

To enable server ports, follow these steps:

1. Select the Equipment tab on the left site.
2. Select Equipment > Fabric Interconnects > Fabric Interconnect A (subordinate) > Fixed Module.
3. Click Ethernet Ports section.

4. Select Ports 1-3, right-click and then select Configure as Server Port.
5. Click Yes and then OK.
6. Repeat steps 1-5 for Fabric Interconnect B.

### 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 (subordinate) > Fixed Module.
3. Click Ethernet Ports section.
4. Select Ports 25-26, right-click and then select Configure as Uplink Port.
5. Click Yes and then OK.
6. Repeat steps 1-5 for Fabric Interconnect B.

### Create Port Channel for Fabric Interconnect A/B

To create Port Channels to the connected Nexus C9336C-FX2 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 25.
4. Type in vPC25 in the Name field.
5. Click Next.
6. Select the available ports on the left 25-26 and assign them with >> to Ports in the Port Channel.

Figure 22 Create Port Channel

**Create Port Channel**

**1 Set Port Channel Name**

**2 Add Ports**

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

>>  
<<

Ports in the port channel			
Slot ID	Aggr. Po...	Port	MAC
1	0	25	00:2A:1...
1	0	26	00:2A:1...

< Prev   Next >   **Finish**   Cancel

7. Click Finish and then OK.
8. Repeat steps 1-7 for Fabric B under LAN > LAN Cloud > Fabric B > Port Channels and right-click Create Port Channel.
9. Type in ID 26.
10. Type in vPC26 in the Name field.
11. Click Next.
12. Select the available ports on the left 25–26 and assign them with >> to Ports in the Port Channel.
13. Click Finish and then OK.

### Label Each Server for Identification

To label each chassis for better identification, follow these steps:

1. Select the Equipment tab on the left site.
2. Select Rack-Mounts > Servers > Server 1.
3. In the Properties section on the right go to User Label and add Storage-Node1 to the field.
4. Repeat steps 1-3 for Server 2 – 3 by using the following labels (Table 5 ):

Table 5 Server Label

Server	Name
Server 1	Storage-Node1
Server 2	Storage-Node2
Server 3	Storage-Node3

### 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 and right-click Create Block of IPv4 Addresses.
3. Type in `Scality-IP` as Name.
4. (Optional) Enter a Description of the MAC Pool.
5. Set Assignment Order as Sequential.
6. Click Next and then Add.
7. Enter an IP Address in the From field.
8. Enter `Size 10`.
9. Enter your Subnet Mask.
10. Fill in your Default Gateway.
11. Enter your Primary DNS and Secondary DNS if needed.
12. Click OK.

Figure 23 Create Block of IPv4 Addresses

Create Block of IPv4 Addresses

From : 192.168.10.110      Size : 10

Subnet Mask : 255.255.255.0      Default Gateway : 192.168.10.234

Primary DNS : 208.67.222.222      Secondary DNS : 208.67.220.220

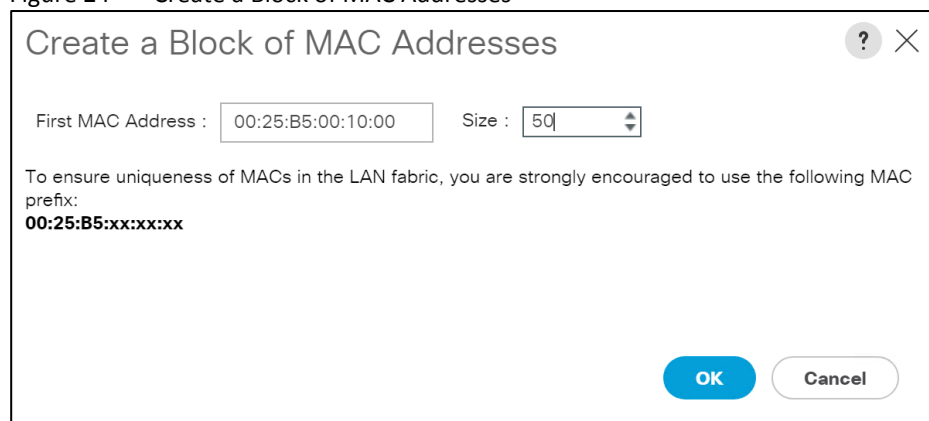
OK Cancel

## Create MAC Pool

To create a MAC Pool, follow these steps:

1. Select the LAN tab on the left site.
2. Go to LAN > Pools > root > Mac Pools and right-click Create MAC Pool.
3. Type in *Scality-MAC* as 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, 50.

Figure 24 Create a Block of MAC Addresses



Create a Block of MAC Addresses

First MAC Address : 00:25:B5:00:10:00      Size : 50

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**

OK Cancel

10. Click OK.
11. Click Finish.

## Create UUID Pool

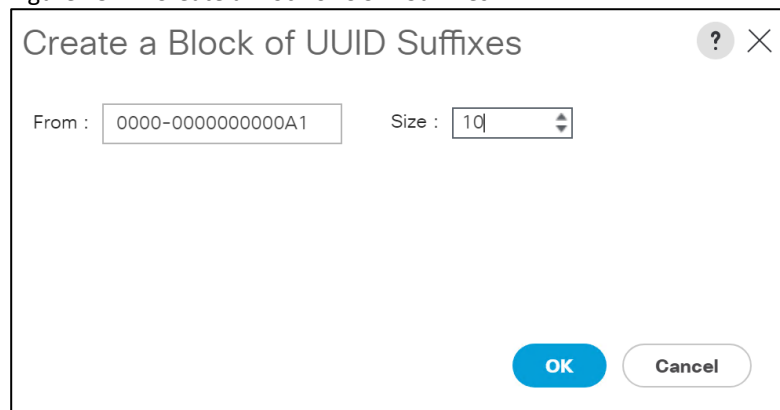
To create a UUID Pool, follow these steps:

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 *Scality-UUID* as Name.
4. (Optional) Enter a Description of the UUID 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, 10.

Figure 25 Create a Block of UUID Suffixes



Create a Block of UUID Suffixes

From : 0000-0000000000A1      Size : 10

OK Cancel

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

## 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 `Scality-CDP` in the Name field.
4. (Optional) Enter a description in the Description field.
5. Click Enabled under CDP.
6. Click All Host Vlans under MAC Register Mode.
7. Leave everything else untouched and click OK.
8. Click OK.

Figure 26 Create Network Control Policy

**Create Network Control Policy**

Name : Scality-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**

**OK** **Cancel**

## 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. Set Weight to none for Platinum, Gold, Silver, Bronze, and Fibre Channel.
4. Set Best Effort Weight to Best Effort and MTU to 9216.
5. Set Fibre Channel Weight to None.
6. Click Save Changes and then OK.

Figure 27 QoS System Class

LAN / LAN Cloud / QoS System Class

General Events FSM

Actions Properties

Use Global Owner : Local

Priority	Enabled	CoS	Packet Drop	Weight	Weight (%)	MTU	Multicast Optimized
Platinum	<input type="checkbox"/>	5	<input type="checkbox"/>	none	N/A	normal	<input type="checkbox"/>
Gold	<input type="checkbox"/>	4	<input checked="" type="checkbox"/>	none	N/A	normal	<input type="checkbox"/>
Silver	<input type="checkbox"/>	2	<input checked="" type="checkbox"/>	none	N/A	normal	<input type="checkbox"/>
Bronze	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	none	N/A	normal	<input type="checkbox"/>
Best Effort	<input checked="" type="checkbox"/>	Any	<input checked="" type="checkbox"/>	best-effort	100	9216	<input type="checkbox"/>
Fibre Channel	<input checked="" type="checkbox"/>	3	<input type="checkbox"/>	none	N/A	fc	N/A

## VLAN Setup

The Scality RING deployment will host two VLANs. One for Management and Client traffic and one for the Storage traffic.

To create the Management VLAN, follow these steps:

1. Select the LAN tab in the left pane of the Cisco UCS Manager GUI.
2. Go to LAN > VLANS and right-click Create VLANs.
3. Type in Management-VLAN in the Name field.
4. (Optional) Enter a description in the Description field.
5. Type in 10 for VLAN IDs and leave everything else untouched.

Figure 28 VLAN

**Create VLANs**

VLAN Name/Prefix : Management-VLAN

Multicast Policy Name : <not set> [Create Multicast Policy](#)

☒ Common/Global
 ☐ Fabric A
 ☐ Fabric B
 ☐ Both Fabrics Configured Differently

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

VLAN IDs : 10

Sharing Type : 
 ☒ None
 ☐ Primary
 ☐ Isolated
 ☐ Community

[Check Overlap](#)
[OK](#)
[Cancel](#)

- Repeat the same step for VLAN Scality-Storage with VLAN ID 20 and VLAN Scality-Client with VLAN ID 30.

## vNIC Template Setup

The next step is to create the appropriate vNIC template. For Scality RING we need to create three vNICs. The first vNIC will handle Management and Client traffic.

To create the appropriate vNIC, follow these steps:

- Select the LAN tab in the left pane of the Cisco UCS Manager GUI.
- Go to LAN > Policies > root > vNIC Templates and right-click Create vNIC Template.
- Type in Scality-Mgmt in the Name field.
- (Optional) Enter a description in the Description field.
- Click Fabric A as Fabric ID and enable failover.
- Click Updating Template as Template Type.
- Select Management-VLAN as VLANs and click Native VLAN.
- Type in 1500 for MTU Size.
- Select Scality-MAC as MAC Pool.
- Select Scality-CDP as Network Control Policy.
- Click OK and then OK.

Figure 29 Setup vNIC Template for vNIC Scality-Mgmt

**Create vNIC Template**

Name :

Description :

Fabric ID : ☒ Fabric A ☐ Fabric B ☐ Enable Failover

**Redundancy**

Redundancy Type : ☒ No Redundancy ☐ Primary Template ☐ Secondary Template

**Target**

☒ Adapter ☐ VM

**Warning**

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

Template Type : ☐ Initial Template ☒ Updating Template

**VLANs** | VLAN Groups

Advanced Filter | Export | Print

Select	Name	Native VLAN	VLAN ID
<input type="checkbox"/>	default	<input type="radio"/>	1
<input checked="" type="checkbox"/>	Management-VLAN	<input type="radio"/>	10
<input type="checkbox"/>	Storage-Cluster	<input type="radio"/>	20
<input type="checkbox"/>	Storage-Mgmt	<input type="radio"/>	10

**OK** **Cancel**

The second vNIC will handle the storage traffic. Please repeat the previous steps by using Fabric B and `Storage-VLAN` as Native VLAN. Please set 9000 as MTU size for the second vNIC.

The third vNIC will handle the client traffic. Please repeat the previous steps by using Fabric A and `Client-VLAN` as Native VLAN. Please set 9000 as MTU size for the third vNIC.

## Adapter Policy Setup

To create a specific adapter policy for Scality, 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 `Scality` in the Name field.
4. (Optional) Enter a description in the Description field.
5. Under Resources type in the following values:
  - a. Transmit Queues: 8
  - b. Ring Size: 4096
  - c. Receive Queues: 8

- d. Ring Size: 4096
- e. Completion Queues: 16
- f. Interrupts: 32
6. Under Options enable Receive Side Scaling (RSS).
7. Click OK and then click OK again.

Figure 30 Adapter Policy for Scality

**Create Ethernet Adapter Policy**

Name :

Description :

**Resources**

Pooled : ☒ Disabled ☐ Enabled

Transmit Queues :  [1-1000]

Ring Size :  [64-4096]

Receive Queues :  [1-1000]

Ring Size :  [64-4096]

Completion Queues :  [1-2000]

Interrupts :  [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

**OK** **Cancel**

## LAN Connectivity Policy Setup

To simplify the vNIC setup later in the template, please create a LAN Connectivity Policy including previous created vNICs.

1. Select the LAN tab in the left pane of the Cisco UCS Manager GUI.
2. Go to LAN > Policies > root > LAN Connectivity Policies and right-click Create LAN Connectivity Policy.
3. Type in *Scality-Connect* in the Name field.
4. (Optional) Enter a description in the Description field.
5. Click Add.
6. Type in *Management* in the Name field.

7. Click Use vNIC Template.
8. Under vNIC Template please select *Scality-Mgmt*.
9. Under Adapter Policy please select *Scality*.

Figure 31 vNIC Creation under LAN Connectivity Policy

**Create vNIC**

Name :

Use vNIC Template : ☒

Redundancy Pair : ☐

vNIC Template :

Peer Name :

[Create vNIC Template](#)

---

**Adapter Performance Profile**

Adapter Policy :

[Create Ethernet Adapter Policy](#)

10. Repeat steps 1-9 for the vNIC Storage and vNIC Client with the vNIC template *Scality-Storage* and vNIC template *Scality-Client* and the same Adapter Policy.
11. Click OK and then click OK again.

## Boot Policy Setup

To create a Boot Policy, follow these steps:

1. Select the Servers tab.
2. Go to Servers > Policies > root > Boot Policies and right-click Create Boot Policy.
3. Type in *Scality-Boot* in the Name field.
4. (Optional) Enter a description in the Description field.
5. Click Reboot on Boot Order Change

6. Click Local Devices > Add Local Disk
7. Click Add CD/DVD
8. Click OK.
9. Click OK.

Figure 32 Create Boot Policy

### Create Boot Policy

Name :

Description :

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.

Local Devices

Add Local Disk

Add Local LUN

Add Local JBOD

Add SD Card

Add Internal USB

Add External USB

Add Embedded Local LUN

Add Embedded Local Disk

Add CD/DVD

Add Local CD/DVD

Add Remote CD/DVD

Add Floppy

Add Local Floppy

Boot Order

Name	Ord...	vNIC/v...	Type	LUN N...	WWN	Slot N...	Boot N...	Boot P...	Descri...
Local Disk	1								
CD/DVD	2								

Move Up

Move Down

Delete

Set Uefi Boot Parameters

OK

Cancel

## Create Maintenance Policy Setup

To setup a Maintenance Policy, follow these steps:

1. Select the Servers tab in the left pane.
2. Go to Servers > Policies > root > Maintenance Policies and right-click Create Maintenance Policy.
3. Type in `Scality-Maint` in the Name field.
4. (Optional) Enter a description in the Description field.
5. Click User Ack under Storage Config. Deployment Policy.

64



- Click User Ack under Reboot Policy.
- Click OK and then click OK again.

Figure 33 Create Maintenance Policy

Create Maintenance Policy

Name : Scality-Maint

Description :

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.)

OK Cancel

### Create Power Control Policy Setup

To create a Power Control Policy, follow these steps:

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

Figure 34 Create Power Control Policy

**Create Power Control Policy**

Name : Scality-Power

Description :

Fan Speed Policy : Any

**Power Capping**

If you choose **cap**, the server is allocated a certain amount of power based on its priority within its power group. Priority values range from 1 to 10, with 1 being the highest priority. If you choose **no-cap**, the server is exempt from all power capping.

☒ No Cap ☐ cap

Cisco UCS Manager only enforces power capping when the servers in a power group require more power than is currently available. With sufficient power, all servers run at full capacity regardless of their priority.

OK Cancel

## Create Disk Scrub Policy

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

To create a Disk Scrub Policy, follow these steps:

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

Figure 35 Create a Disk Scrub Policy

Create Scrub Policy

Name : Scality-Scrub

Description :

Disk Scrub : ☐ No ☒ Yes

BIOS Settings Scrub : ☒ No ☐ Yes

FlexFlash Scrub : ☒ No ☐ Yes

OK Cancel

## Create Host Firmware Package

To create a Host Firmware Policy, follow these steps:

1. Select the Servers tab in the left pane.
2. Go to Servers > Policies > root > Host Firmware Packages and right-click Create Host Firmware Package.
3. Type in Scality-FW in the Name field.
4. (Optional) Enter a description in the Description field.
5. Under Rack Package select 4.0 (2d) C.
6. Under Excluded Components, deselect Local Disk.
7. Click OK and then click OK again.

Figure 36 Create Host Firmware Policy

**Create Host Firmware Package**

Name :

Description :

How would you like to configure the Host Firmware Package?

☒ Simple ☐ Advanced

Blade Package :

Rack Package :

Service Pack :

**The images from Service Pack will take precedence over the images from Blade or Rack Package**

Excluded Components:

- ☐ Adapter
- ☐ BIOS
- ☐ Board Controller
- ☐ CIMC
- ☐ FC Adapters
- ☐ Flex Flash Controller
- ☐ GPUs
- ☐ HBA Option ROM
- ☐ Host NIC
- ☐ Host NIC Option ROM
- ☐ Local Disk
- ☐ NVME Mswitch Firmware
- ☐ PSU
- ☐ Port Switch Firmware

**OK** **Cancel**

## Create Storage Profiles

In the next part we're going to create the Disk Group Policy and Storage Profile for the boot devices for the rear end SATA SSDs and a LUN Set for the 10 TB data devices.

### Create Disk Group Policy for Boot Devices

To create the Disk Group Policy from the rear end SATA SSDs, follow these steps:

1. Select Storage in the left pane of the Cisco UCS Manager GUI.
2. Go to Storage > Storage Policies > root > Disk Group Policies and right-click Create Disk Group Policy
3. Type in `Scality-Boot` in the Name field.
4. (Optional) Enter a description in the Description field.
5. Select `RAID 1 Mirrored` for RAID Level.
6. Click Disk Group Configuration (Manual).

7. Click Add.
8. Type in 13 as slot number.
9. Repeat the step for slot number 14.
10. Click Read Write under Access Policy.
11. Select Read Ahead under Read Policy.
12. Click Write Back Good Bbu under Write Cache Policy.
13. Select Cached under IO Policy.
14. Click No Change under Drive Cache.
15. Click OK and then click OK again.

Figure 37 Create Disk Group Policy for Boot Device

**Create Disk Group Policy**

Advanced Filter Export Print

Slot Number	Role	Span ID
13	Normal	Unspecified
14	Normal	Unspecified

+ Add - Delete i Info

**Virtual Drive Configuration**

Strip Size (KB) : Platform Default

Access Policy : ☐ Platform Default ☒ Read Write ☐ Read Only ☐ Blocked

Read Policy : ☐ Platform Default ☒ Read Ahead ☐ Normal

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

IO Policy : ☐ Platform Default ☐ Direct ☒ Cached

Drive Cache : ☐ Platform Default ☒ No Change ☐ Enable ☐ Disable

Security : ☐

OK Cancel

## Create Storage Profile

To create the Storage Profile, 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 Scalify-Storage in the Name field.

4. (Optional) Enter a description in the Description field.
5. Click Add under Local LUN.
6. Type in Boot in the Name field.
7. Click Expand To Available.
8. Select Scalify-Boot under Select Disk Group Configuration.
9. Click OK.
10. Click LUN Set.
11. Click Add.
12. Type in Data in the Name field.
13. Under Disk Slot Range type in 1-12.
14. Click Read Write under Access Policy.
15. Select Read Ahead under Read Policy.
16. Click Write Back Good Bbu under Write Cache Policy.
17. Select Cached under IO Policy.
18. Click No Change under Drive Cache.
19. Click OK, then click OK and then click OK again.

Figure 38 LUN Set for 10 TB Data Disks

**Create LUN Set**

Name : Data

RAID Level : ☒ RAID 0 Striped

Disk Slot Range : 1-12

**Virtual Drive Configuration**

Strip Size (KB) : Platform Default

Access Policy : ☐ Platform Default ☒ Read Write ☐ Read Only ☐ Blocked

Read Policy : ☐ Platform Default ☒ Read Ahead ☐ Normal

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

IO Policy : ☐ Platform Default ☐ Direct ☒ Cached

Drive Cache : ☐ Platform Default ☒ No Change ☐ Enable ☐ Disable

Security : ☐

OK Cancel

## Create Service Profile Template

### Create Service Profile Template

To create the 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 and right-click Create Service Profile Template.

### Identify Service Profile Template

1. Type in `Scality` in the Name field.
2. Click Updating Template in Type.
3. In the UUID Assignment section, select the `Scality-UUID Pool`.
4. (Optional) Enter a description in the Description field.
5. Click Next.

### Storage Provisioning

1. Go to the Storage Profile Policy tab and select the Storage Profile `Scality-Storage`.
2. Click Next.

### Networking

1. Select the UseConnectivity Policy radio button for the option How would you like to configure LAN connectivity?
2. Select `Scality-Connect` under LAN Connectivity Policy.
3. Click Next to continue with SAN Connectivity.
4. Select No vHBA for How would you like to configure SAN Connectivity?
5. Click Next to continue with Zoning.
6. Click Next to continue with vNIC/vHBA Placement.
7. Click Next to continue with vMedia Policy.

### Server Boot Order

1. Select `Scality-Boot` from the Boot Policy Menu.
2. Click Next.

### Server Maintenance

1. Select the Maintenance Policy `Scality-Maint` under Maintenance Policy.

2. Click Next.
3. Click Next.

### Server Assignment

1. Under Firmware Management select `Scality-FW`.
2. Click Next.

### Operational Policies

1. Under Power Control Policy Configuration select `Scality-Power`, under Scrub Policy select `Scality-Scrub`.
2. Click Finish.

### Create Service Profiles from Template

Create the appropriate Service Profiles from the previous Service Profile Template. To create all three profiles for the Scality Server, 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 Create Service Profiles from Template.
3. Type in `storage-node` in the Name Prefix field.
4. Type 1 for Name Suffix Starting Number.
5. Type 3 for Number of Instances.
6. Choose `Scality` under Service Profile Template.
7. Click OK.

### Associate Service Profiles

To associate the service profiles, follow these steps:

1. Right-click the service profile `storage-node1` and choose Change Service Profile Association.
2. Server Assignment should be Select Existing Server.
3. Select Rack ID 1.
4. Click OK and Yes and OK.
5. Repeat the steps for `storage-node2` and Rack ID 2. Repeat the steps for the third Service Profiles counting up the Rack ID number corresponding with the service profile.

The formal setup of the Cisco UCS Manager environment and both Cisco Nexus C9336C-FX2 switches is now finished and the installation of the Operating System software will continue.



## Install Red Hat Enterprise Linux 7.6

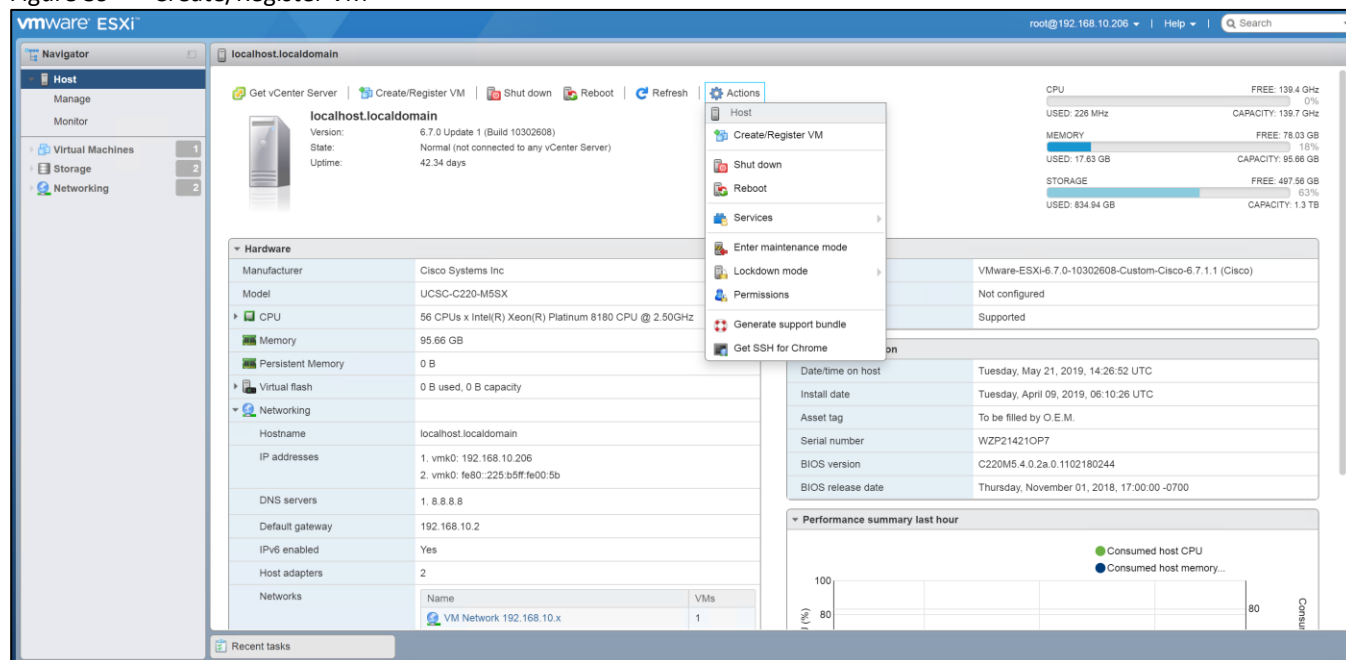
The following section provides detailed information about installing RHEL 7.6 on Cisco UCS C240 M5L. It describes the process for installing RHEL 7.6 as a virtual machine for the Scality Supervisor and as a bare metal install for the Scality Storage nodes.

### Deployment of Scality Supervisor VM on ESXi 6.7

To deploy the virtual machine Scality Supervisor on VMware vCenter, follow these steps:

1. Log into your local ESXi host you want to use for deploying the virtual appliance.
2. Select under Actions – Create/Register VM.

Figure 39 Create/Register VM



3. Select Create a new virtual machine and then click Next.
4. Type in a name in the Name field.
5. Select as Guest OS family Linux.
6. Select as Guest OS version Red Hat Enterprise Linux 7 (64-bit).

Figure 40    Select a name and guest OS

New virtual machine - Scality-Supervisor (ESXi 6.7 virtual machine)

✓ 1 Select creation type

**2 Select a name and guest OS**

3 Select storage

4 Customize settings

5 Ready to complete

vmware

### Select a name and guest OS

Specify a unique name and OS

Name

Scality-Supervisor

Virtual machine names can contain up to 80 characters and they must be unique within each ESXi instance.

Identifying the guest operating system here allows the wizard to provide the appropriate defaults for the operating system installation.

Compatibility

ESXi 6.7 virtual machine

Guest OS family

Linux

Guest OS version

Red Hat Enterprise Linux 7 (64-bit)

Back

Next

Finish

Cancel

7.    Select your storage and click Next.

Figure 41 Select Storage

New virtual machine - Scality-Supervisor (ESXi 6.7 virtual machine)

✓ 1 Select creation type  
 ✓ 2 Select a name and guest OS  
 ✓ 3 **Select storage**  
 4 Customize settings  
 5 Ready to complete

### Select storage

Select the storage type and datastore

☒ Standard
 ☐ Persistent Memory

Select a datastore for the virtual machine's configuration files and all of its' virtual disks.

Name	Capacity	Free	Type	Thin pro...	Access
Boot-RAID1	438.5 GB	437.09 GB	VMFS6	Supported	Single
Bulk	443.25 GB	441.84 GB	VMFS6	Supported	Single
Bulk-JBOD-M.2	894 GB	60.47 GB	VMFS6	Supported	Single
Scality	10.48 TB	10.48 TB	VMFS6	Supported	Single

4 items

Back Next Finish Cancel

8. Under Customize Settings, select the following:
  - a. 4 vCPU
  - b. 16384 MB Memory
  - c. 800 GB Hard Disk 1
  - d. VM Network for Management
9. Add another network adapter by clicking Add network adapter and select the VM Network for Storage.
10. Click Next and then click Finish.

Figure 42 Customer VM Settings

New virtual machine - Scalify-Supervisor (ESXi 6.7 virtual machine)

✓ 1 Select creation type  
 ✓ 2 Select a name and guest OS  
 ✓ 3 Select storage  
 ✓ 4 **Customize settings**  
 5 Ready to complete

### Customize settings

Configure the virtual machine hardware and virtual machine additional options

Virtual Hardware VM Options

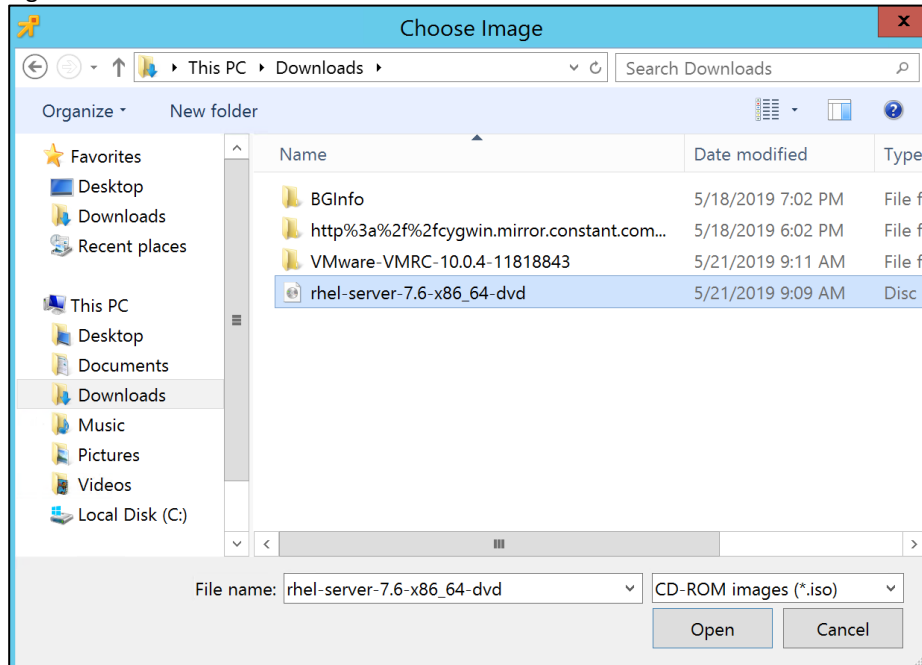
Add hard disk Add network adapter Add other device

CPU	4		
Memory	16384	MB	
Hard disk 1	800	GB	
SCSI Controller 0	VMware Paravirtual		
SATA Controller 0			
USB controller 1	USB 2.0		
Network Adapter 1	VM Network 192.168.10.x	<input checked="" type="checkbox"/> Connect	
New Network Adapter	VM Network 192.168.20.x	<input checked="" type="checkbox"/> Connect	
CD/DVD Drive 1	Host device	<input checked="" type="checkbox"/> Connect	

Back Next Finish Cancel

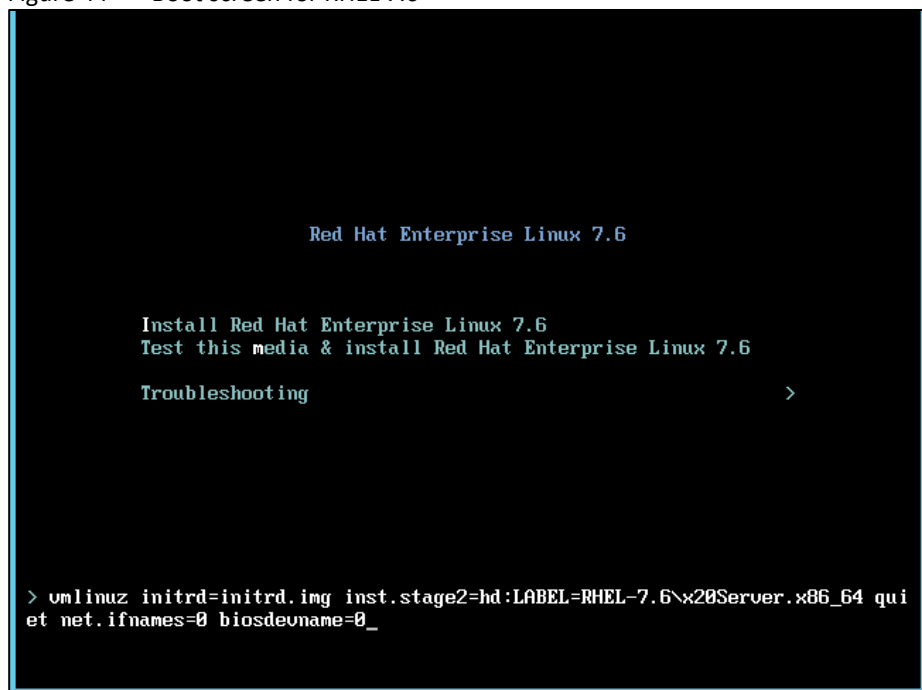
11. Click the virtual machine and power on the VM.
12. Click Console and then click Launch remote console.
13. Click Always trust this host with this certificate and then Connect Anyway.
14. Under VMRC click Removable Devices, CD/DVD drive 1 and then click Connect to Disk Image File (ISO)
15. Click Yes and select the RHEL 7.6 ISO by selecting rhel-server-7.6-x86\_64-dvd.iso in your local file system.

Figure 43 Select RHEL 7.6 ISO



16. Click VMRC and then click Send Ctrl+Alt+Del.
17. Click in the window and select Install Red Hat Enterprise Linux 7.6 with the up arrow key.
18. Press the Tab key and type in `net.ifnames=0 biosdevname=0` and press Enter.

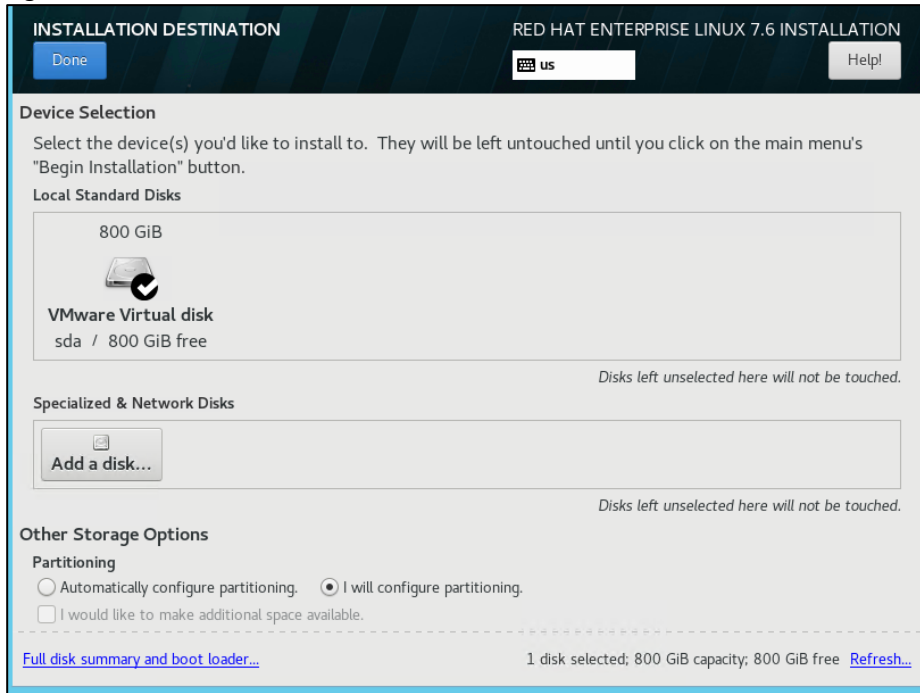
Figure 44 Boot screen for RHEL 7.6



19. In the welcome screen select your language and press Continue.

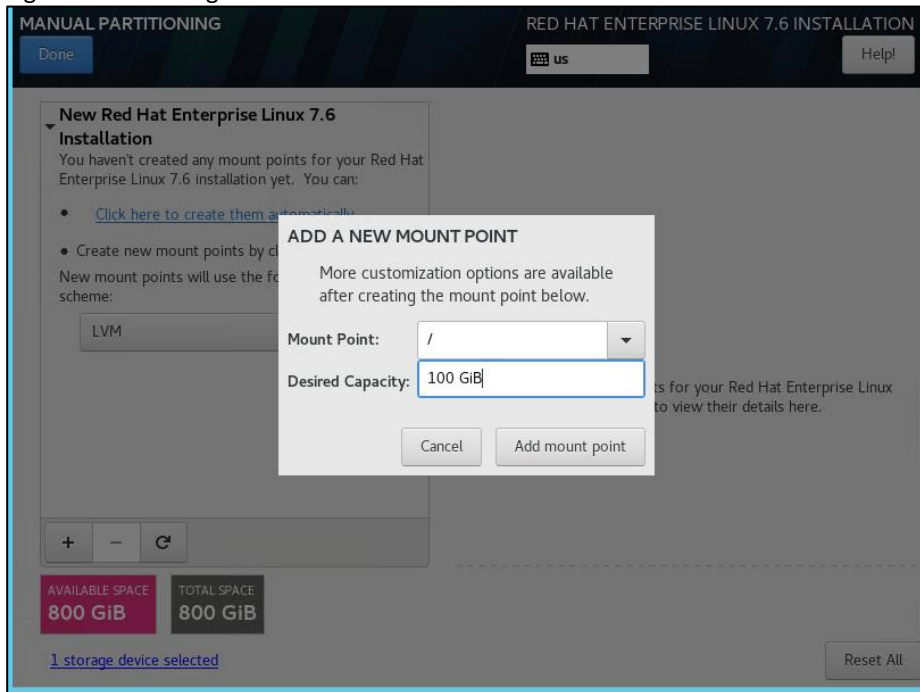
20. Select Installation Destination, click I will configure partitioning and then click Done.

Figure 45 RHEL 7.6 Installation Destination



21. In the next screen click + and then select the / mount point with a capacity of 100 GiB. Click Add mount point.

Figure 46 Configure RHEL Mount Points



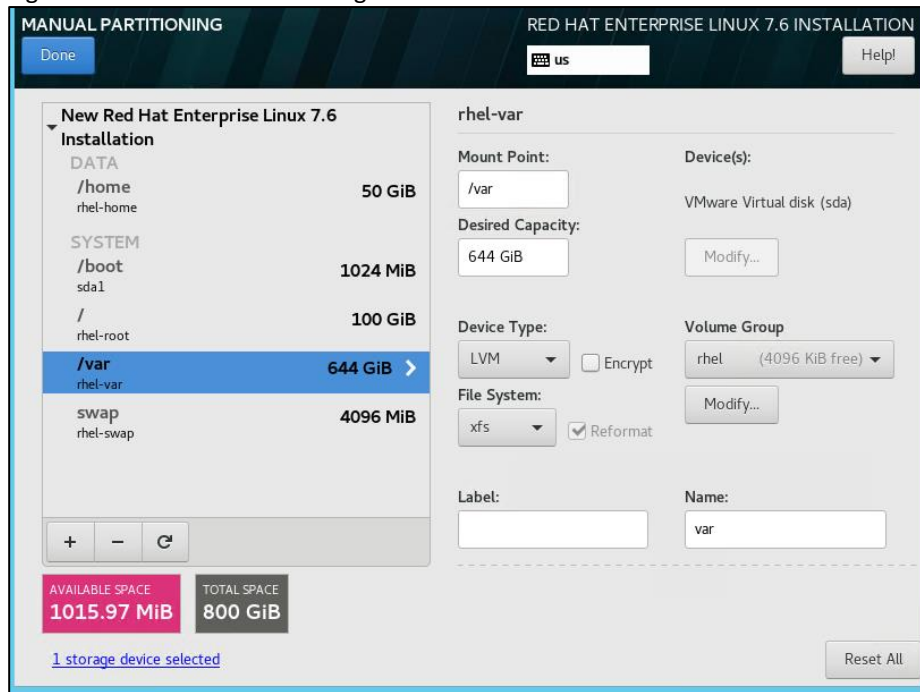
22. For the other mount points configure the following sizes:

- /boot = 1 GiB

- /home = 50 GiB
- swap = 4 GiB
- /var = 644 GiB

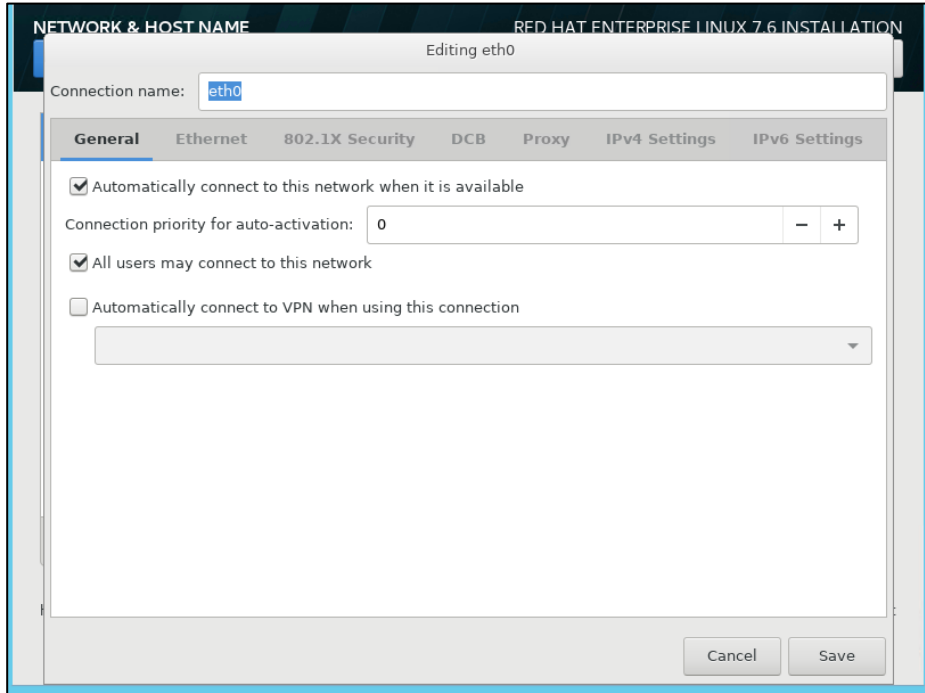
The last screen for partitioning will look like the following:

Figure 47 Manual Partitioning



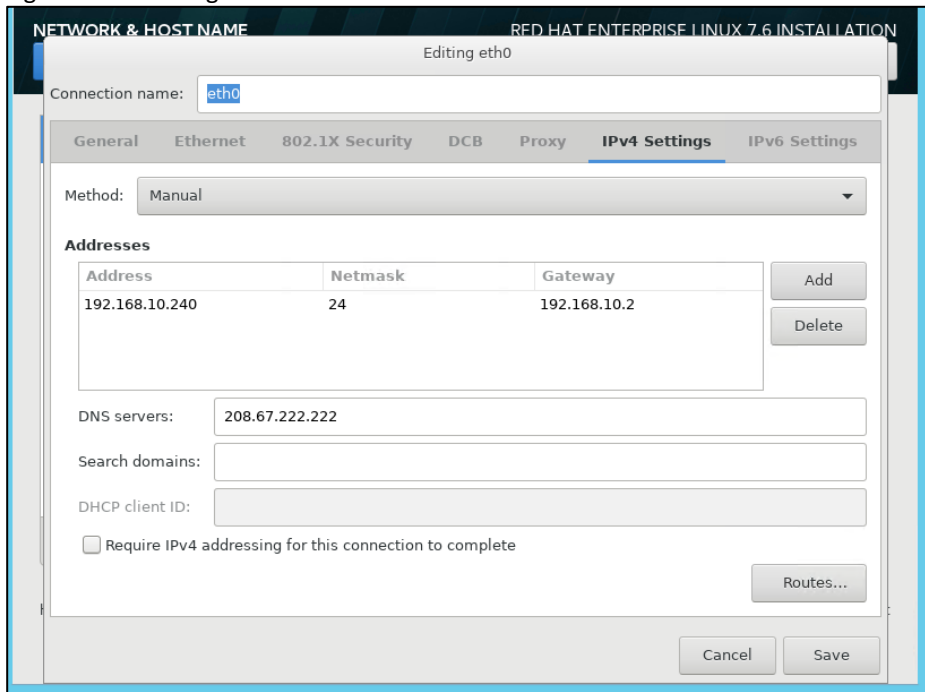
23. Click Done and then click Accept Changes.
24. Click Network & Host Name, select eth0 and click Configure.
25. Select the General tab and click Automatically connect to this network when it is available.

Figure 48 Editing eth0



26. Click the tab IPv4 Settings and then select Manual under Method.
27. Click Add and fill in Address, Netmask and Gateway.
28. Fill in the DNS IP and click Save.

Figure 49 Configuration of eth0



29. Repeat steps 24-28 for eth1 by selecting the Storage IP subnet and MTU 9000 under Ethernet.



30. Repeat steps 24-28 for eth2 by selecting the Client IP subnet and MTU 9000 under Ethernet.
31. Under Host name fill in the name of the host and click Apply.
32. Click Done.
33. Click Date & Time and select the region where the Supervisor is getting installed. Click Done.
34. Click Begin Installation.
35. Click Root Password and fill in the root password twice and Click Done.
36. Wait until the installation ends and the Reboot buttons comes up. Click Reboot and reboot the system.

## Configure RHEL 7.6 for Scality Supervisor

To subscribe RHEL 7.6 to Red Hat and update and prepare for the Scality installation, follow these steps:

1. Connect to the previous installed virtual machine, register the system to Red Hat Subscription, enable the packages and update the operating system:

```
[root@supervisor ~]# subscription-manager register
Registering to: subscription.rhsm.redhat.com:443/subscription
Username: XXX > Fill in your Red Hat Username
Password: YYY > Fill in your Red Hat Password
The system has been registered with ID: d029954f-1a3e-4f5d-ac52-4dd8e37687b7
The registered system name is: supervisor
[root@supervisor ~]# subscription-manager refresh
All local data refreshed
[root@supervisor ~]# subscription-manager list --available
[root@supervisor ~]# subscription-manager attach --pool=ZZZ > Fill in your Red Hat Subscription
Successfully attached a subscription for: Red Hat Enterprise Linux, Self-Support
(128 Sockets, NFR, Partner Only)
[root@supervisor ~]# subscription-manager repos --disable=*
[root@supervisor ~]# subscription-manager repos --enable=rhel-7-server-rpms
Repository 'rhel-7-server-rpms' is enabled for this system.
[root@supervisor ~]# subscription-manager repos --enable=rhel-7-server-extras-rpms
Repository 'rhel-7-server-extras-rpms' is enabled for this system.
[root@supervisor ~]# yum -y update
```

2. Install and configure NTP to make sure that the whole cluster is running with no time difference:

```
[root@supervisor ~]# yum -y install ntp

[root@supervisor ~]# vi /etc/ntp.conf > Add the NTP server of your choice by adding
the following line

Server 173.38.201.115 iburst

[root@supervisor ~]# systemctl enable ntpd

Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to
/usr/lib/systemd/system/ntpd.service.

[root@supervisor ~]# systemctl start ntpd

[root@supervisor ~]# systemctl disable chronyd

Removed symlink /etc/systemd/system/multi-user.target.wants/chronyd.service.

[root@supervisor ~]# systemctl stop chronyd

[root@supervisor ~]# ntpq -p

      remote           refid      st t when poll reach   delay   offset   jitter
=====
*aer01-r4c20-dc- .GNSS.          1 u   47   64    1 167.823    0.304    0.233
```

### 3. Configure /etc/hosts:

```
[root@supervisor ~]# vi /etc/hosts

127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.10.202  supervisor
192.168.10.203  storage1
192.168.10.204  storage2
192.168.10.205  storage3
```

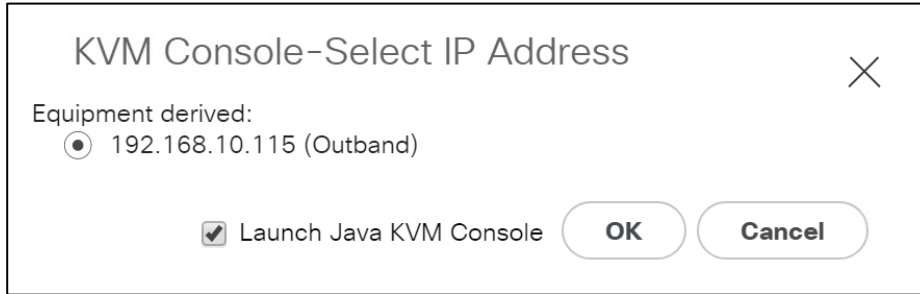
The Scalify Supervisor virtual machine is ready for the Scalify installation. The next step is to set up the storage nodes.

## Deploy Scalify Storage-Nodes

To install and configure Red Hat Enterprise Linux 7.6 on all three storage nodes, follow these steps:

1. Connect to UCS Manager and go to Equipment > Server > Storage-Node1 and click the double arrow then select Launch Java KVM Console and click OK.

Figure 50 Launch Java KVM Console



2. If you use Google Chrome, click Keep to store the Java file and then click the Java file.
3. In the next screen click Continue, then click Run and then click Continue.
4. In the Java window, please go to Virtual Media, click Activate Virtual Devices, then click Accept this session and Remember this configuration for future connections to this server and then click Apply.
5. Click Virtual Media and then Map CD/DVD... and then click Browse to select your RHEL 7.6 ISO image and finally Map Device
6. From the top bar of the Java window, select Reset and then click Yes.
7. The system now reboots. When you see the Cisco logo in the top left corner press F6 to select the boot devices.

Figure 51 Select Boot Menu



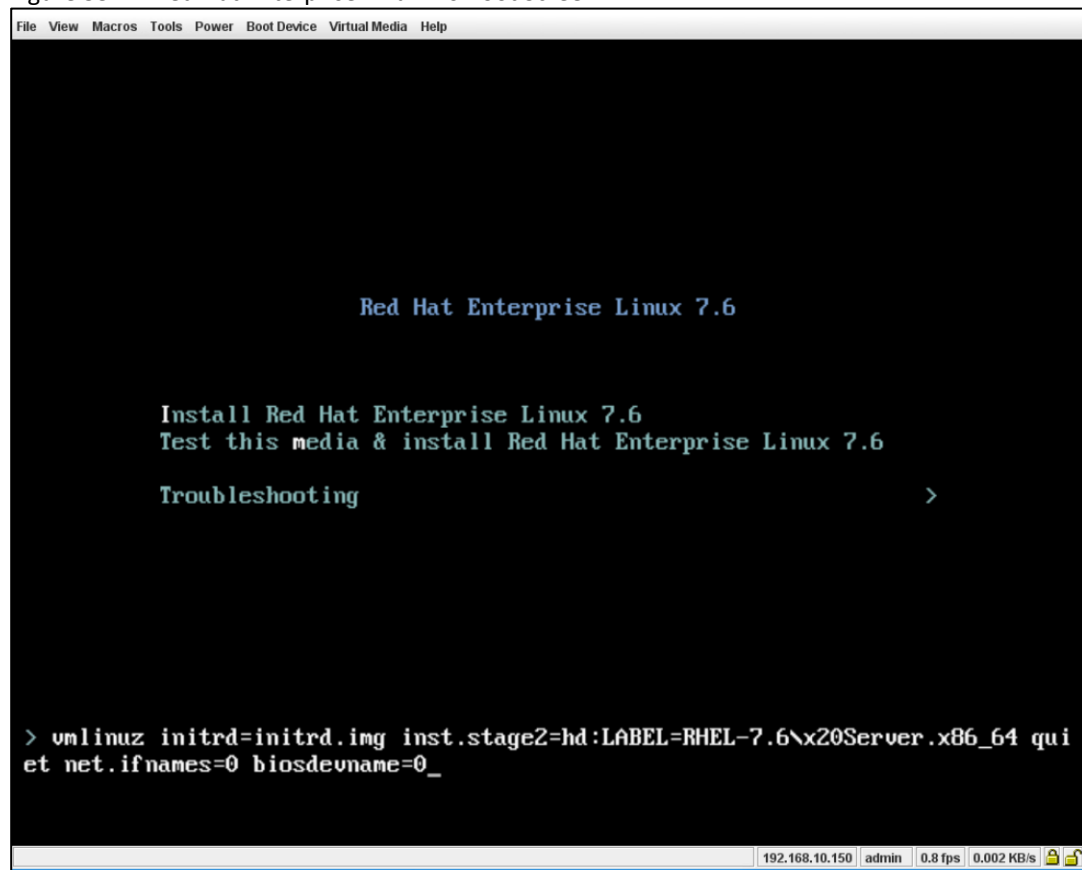
8. When the boot screen comes up, select Cisco vKVM-Mapped vDVD1.24.

Figure 52 Select Boot Device



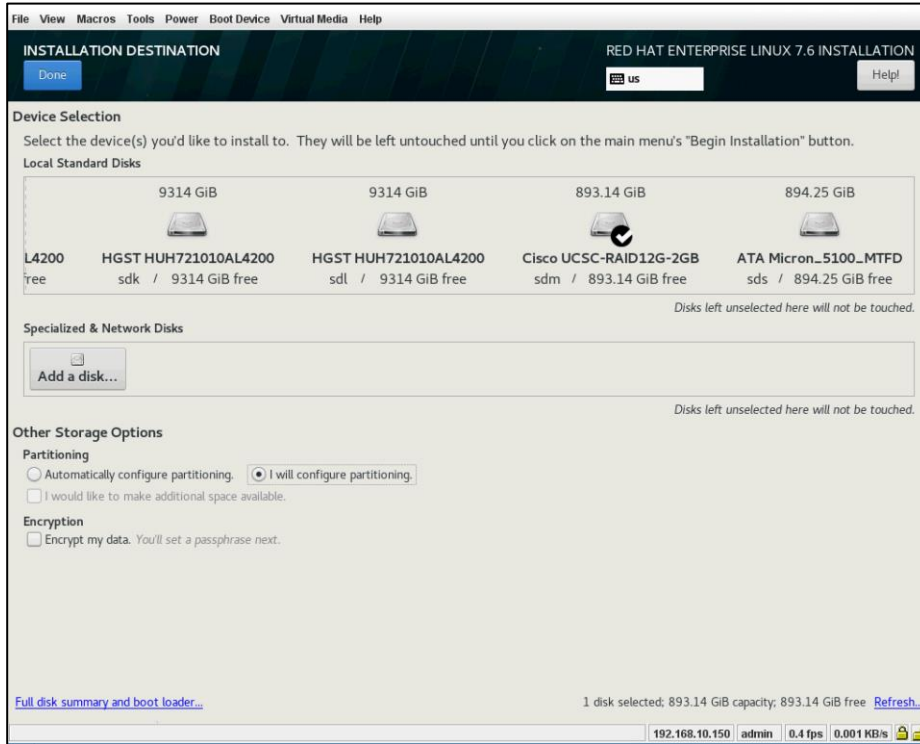
9. In the next screen, use the arrow-up key and select Install Red Hat Enterprise Linux 7.6 and press the Tab button.
10. Type in net.ifnames=o biosdevname=o.

Figure 53 Red Hat Enterprise Linux 7.6 Boot Screen



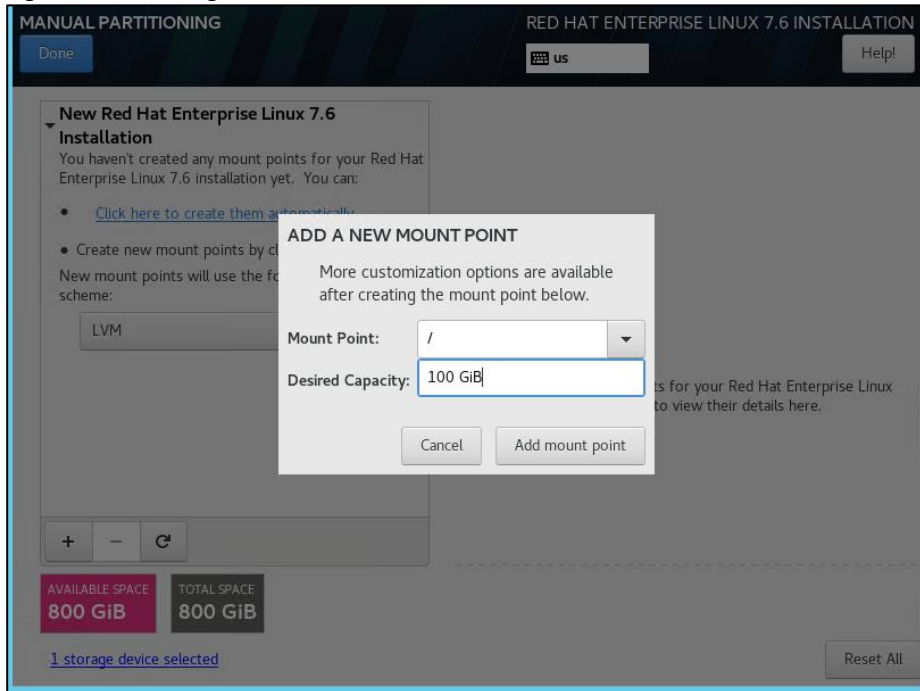
11. In the Welcome screen select your language and press Continue.
12. Select Installation Destination, click I will configure partitioning and then click Done.

Figure 54 RHEL 7.6 Installation Destination



13. In the next screen click + and then select the / mount point with a capacity of 100 GiB. Click Add mount point.

Figure 55 Configure RHEL Mount Points



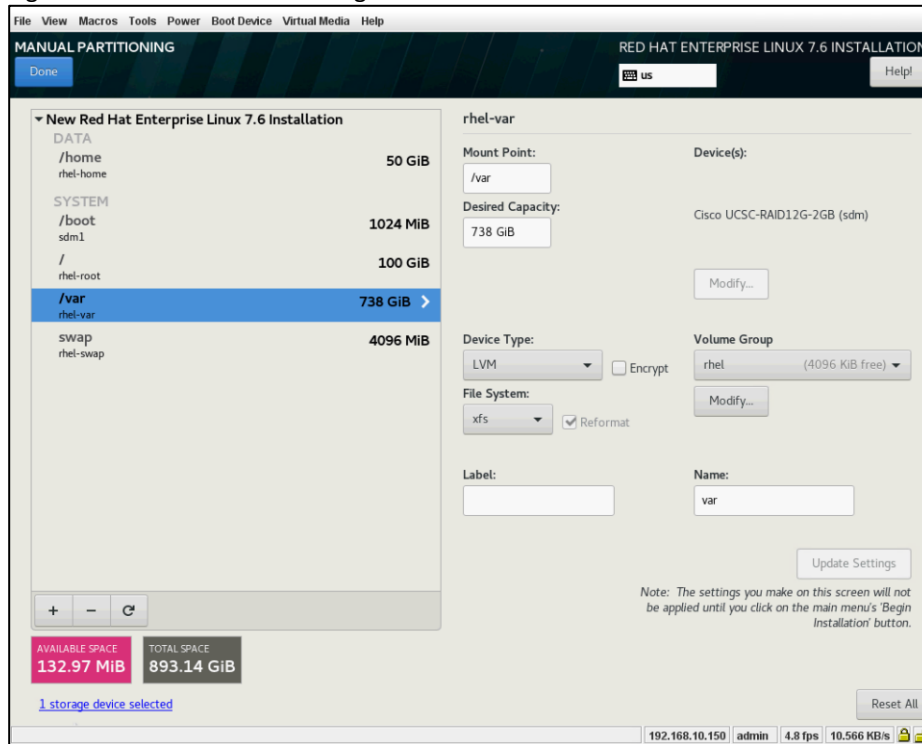
14. For the other mount points configure the following sizes:

- a. /boot = 1 GiB

- b. /home = 50 GiB
- c. swap = 4 GiB
- d. /var = 738 GiB

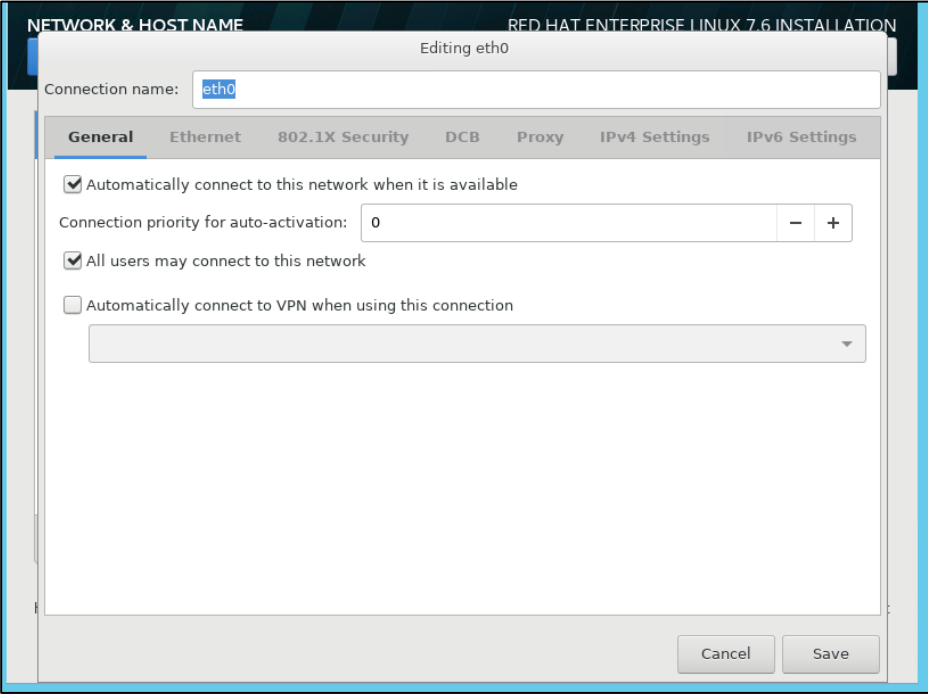
The final screen for partitioning will look like the following:

Figure 56 Manual Partitioning



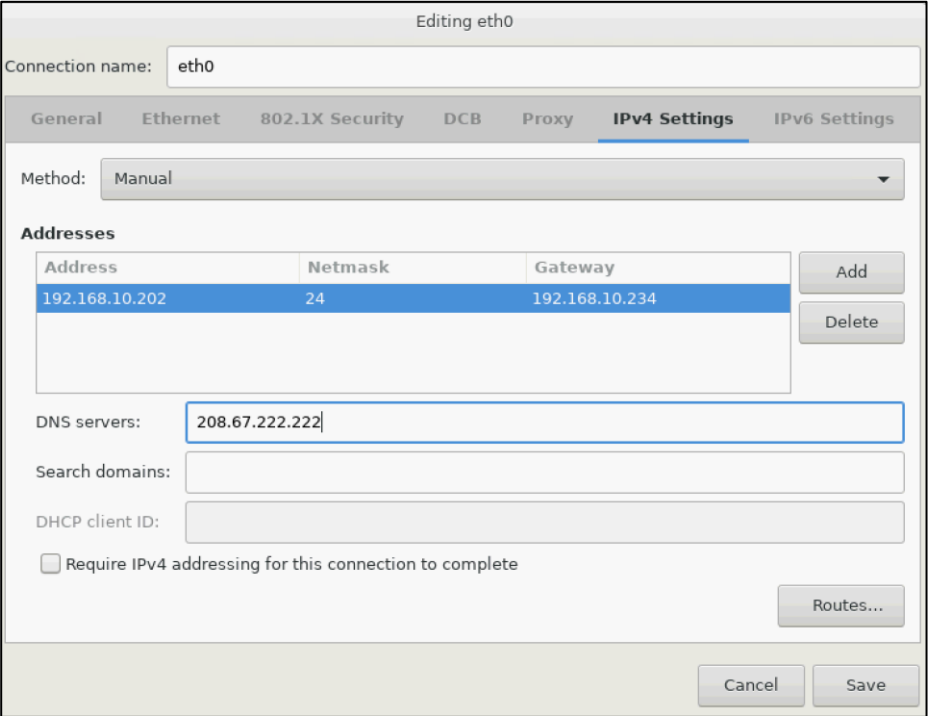
15. Click Done and then click Accept Changes.
16. Click Network & Host Name, select eth0 and click Configure.
17. Select the General tab and click Automatically connect to this network when it is available.

Figure 57 Editing eth0



18. Click the tab IPv4 Settings and then select Manual under Method.
19. Click Add and fill in Address, Netmask, and Gateway.
20. Fill in the DNS IP and click Save.

Figure 58 Configuration of eth0





21. Repeat steps 16-20 for eth1 by selecting the Storage IP subnet and MTU 9000 under Ethernet.
22. Repeat steps 16-20 for eth2 by selecting the Client IP subnet and MTU 9000 under Ethernet.
23. Under Host name fill in the name of the host and click Apply.
24. Click Done.
25. Click Date & Time and select the region where the Supervisor is getting installed. Click Done.
26. Click Begin Installation.
27. Click Root Password and fill in the root password twice and click Done.
28. Wait until the installation ends and the Reboot buttons comes up. Click Reboot and reboot the system.

## Configure RHEL 7.6 for Scalify Storage-Node

To subscribe the RHEL 7.6 to Red Hat and update and prepare for the Scalify installation, follow these steps:

1. Connect to the first Storage-Node, register the system to Red Hat Subscription, enable the packages and update the operating system:

```
[root@storagel ~]# subscription-manager register
Registering to: subscription.rhsm.redhat.com:443/subscription
Username: XXX > Fill in your Red Hat Username
Password: YYY > Fill in your Red Hat Password

The system has been registered with ID: 17658e1e-2fe2-423f-afb2-5a2a42da3040
The registered system name is: storagel

[root@storagel ~]# subscription-manager refresh
All local data refreshed

[root@storagel ~]# subscription-manager list --available
[root@storagel ~]# subscription-manager attach --pool=ZZZ > Fill in your Red Hat Subscription

Successfully attached a subscription for: Red Hat Enterprise Linux, Self-Support
(128 Sockets, NFR, Partner Only)

[root@storagel ~]# subscription-manager repos --disable=*
[root@storagel ~]# subscription-manager repos --enable=rhel-7-server-rpms
Repository 'rhel-7-server-rpms' is enabled for this system.

[root@storagel ~]# subscription-manager repos --enable=rhel-7-server-extras-rpms
Repository 'rhel-7-server-extras-rpms' is enabled for this system.

[root@storagel ~]# yum -y update
```

2. Install and configure NTP to make sure that the whole cluster is running with no time difference:

```
[root@storage1 ~]# yum -y install ntp
[root@storage1 ~]# vi /etc/ntp.conf > Add the NTP server of your choice by adding the following line
Server 173.38.201.115 iburst
[root@storage1 ~]# systemctl enable ntpd
Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to /usr/lib/systemd/system/ntpd.service.
[root@storage1 ~]# systemctl start ntpd
[root@storage1 ~]# systemctl disable chronyd
Removed symlink /etc/systemd/system/multi-user.target.wants/chronyd.service.
[root@storage1 ~]# systemctl stop chronyd
[root@storage1 ~]# ntpq -p
```

remote	refid	st	t	when	poll	reach	delay	offset	jitter
=====									
*aer01-r4c20-dc-	.GNSS.	1	u	47	64	1	167.823	0.304	0.233

3. Configure /etc/hosts:

```
[root@storage1 ~]# vi /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.10.202  supervisor
192.168.10.203  storage1
192.168.10.204  storage2
192.168.10.205  storage3
```

4. To make sure that the order for disk devices is the same after each reboot, add the following to /etc/sysconfig/grub (marked in red):

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
```

```
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=rhel_gluster100/root
rd.lvm.lv=rhel_gluster100/swap rd.driver.pre=megaraid_sas biosdevname=0
net.ifnames=0 rhgb quiet"
```

```
GRUB_DISABLE_RECOVERY="true"
```

5. After you make the desired change, run the following command to write the GRUB configuration and reboot the server:

```
[root@storage1 ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-957.12.2.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-957.12.2.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-957.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-957.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-7c9a4e2ce2014948b755b1ea060ee9af
Found initrd image: /boot/initramfs-0-rescue-7c9a4e2ce2014948b755b1ea060ee9af.img
done
```

6. Check all physical volumes:

```
[root@storage1 ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	9.1T	0	disk	
sdb	8:16	0	9.1T	0	disk	
sdc	8:32	0	9.1T	0	disk	
sdd	8:48	0	9.1T	0	disk	
sde	8:64	0	9.1T	0	disk	
sdf	8:80	0	9.1T	0	disk	
sdg	8:96	0	9.1T	0	disk	
sdh	8:112	0	9.1T	0	disk	
sdi	8:128	0	9.1T	0	disk	
sdj	8:144	0	9.1T	0	disk	
sdk	8:160	0	9.1T	0	disk	
sdl	8:176	0	9.1T	0	disk	
sdm	8:192	0	893.1G	0	disk	
└─sdm1	8:193	0	1G	0	part	/boot
└─sdm2	8:194	0	892G	0	part	

```

└─rhel-root 253:0    0   100G  0 lvm  /
└─rhel-swap 253:1    0     4G  0 lvm  [SWAP]
└─rhel-home 253:2    0    50G  0 lvm  /home
└─rhel-var  253:3    0   738G  0 lvm  /var
sdn          8:208    0 894.3G  0 disk
sdo          8:224    0 894.3G  0 disk
sr0          11:0     1  1024M  0 rom
sr1          11:1     1  1024M  0 rom
sr2          11:2     1  1024M  0 rom
sr3          11:3     1  1024M  0 rom

```

The first storage node is now ready for the Scalify RING installation. Repeat steps 1-6 for the other two storage nodes.

## Scality RING Installation

---

The following procedures explain the installation and configuration of the Scality RING. The installation process is comprised of five unique stages listed below.

1. Preparing the Environment
2. Running the Pre-Install Suite
3. Installing Scality RING
4. Installing Scality S3 Connector Service
5. Running the Post-Install Suite

### Prerequisites

Before beginning, download the Scality Offline installer with S3 from [packages.scality.com](http://packages.scality.com). The Scality Installer archive comes with three package sets, each of which can be used for RING installation: Offline packages without S3, Offline packages with S3, and Online packages. Scality recommends using Offline packages for installation.

The installer leverages a platform description file to automate the RING installation process. Key to the automated RING installation process, the Platform Description File supplies information to the Scality Installer concerning the infrastructure on which the RING will be installed. It is generated from the Scality Sizing Tool, with system hardware information entered by Sales Engineers and technical details (for example, hostnames, IP addresses, and RING definitions) entered by Customer Service Engineers.

The platform description file used for the CVD can be found in Appendix C.



Please contact your Scality sales representative for access to [packages.scality.com](http://packages.scality.com) and help generating the platform description file.

---

### Start the Installer

After downloading the installer ensure the root execution flag is set on the .run file.

```
$ chmod +x scality-ring-with-s3-offline.run
```

Invoke the installer with the `--description-file` option and pass the platform description file as the argument.

```
$ ./scality-ring-with-s3-offline.run --description-file /root/scality-sizing-cisco-m5.csv
```

### Using the Installer

The Scality Installer Menu offers commands that correlate to major RING installation steps. These commands are presented in the sequence in which the installation steps are best followed.

### Prepare the Environment

The first step in an Automated Installation is to set up the environment for RING installation, set up the Scality repositories to administer Scality, third-party and offline packages, and to deploy SaltStack on every machine listed in the Platform Description File.

To prepare the environment, follow these steps:

1. From the Scality Installer Menu, invoke the Prepare the Environment command.

#### Scality Installer Menu

```

1. Prepare the environment
2. Run the Pre-Install Suite
3. Install Scality RING
4. Install S3 Service (Optional)
5. Run the Post-Install Suite

* Generate the Offline Archive (Optional)
* Reset SSH credentials
* Gather all logs and configuration files

Exit

```

```

===== Description =====
Prepare the servers for installation, setup a local repository,
install the deployment tool and other necessary tools on all servers

```

2. Select option 2, Private Key without passphrase.



The first time an installer command is selected, you will be asked to select the SSH authentication method used to connect to each of the servers.

Please select the SSH authentication method to connect to the cluster servers:

- 1. Password
- 2. Private Key without passphrase
- 3. SSH Agent

Cancel

===== Description =====

Use a private key without passphrase.  
If you want to use a private key with a passphrase please use the SSH Agent.

3. Provide the SSH user that will be used to connect to each of the servers.

Please select the SSH authentication method to connect to the cluster servers:

Please provide the SSH user to connect to the servers (leave blank for "root"):

Cancel

===== Description =====

Use a private key without passphrase.  
If you want to use a private key with a passphrase please use the SSH Agent.

4. Provide the SSH key that will be used to connect to each of the servers.

Please select the SSH authentication method to connect to the cluster servers:

Please provide the SSH key to use (leave blank to use the default one "/root/.ssh/id\_rsa"):

Cancel

===== Description =====

Use a private key without passphrase.  
If you want to use a private key with a passphrase please use the SSH Agent.

5. Choose option 1, enter a password.



The Scality Supervisor UI requires a password.

---

For admin users, the Scality Supervisor WebUI requires a password.

1. Enter a password
2. Generate a password

Cancel

===== Description =====

A prompt for a password will display. Enter a password and confirm it. This password can thereafter be used to access the Scality Supervisor in an admin capacity.

The installer will now prepare the environment:

```
[2019-06-03 09:41:19,539] Loading the platform description file
'/root/scality_sizing_l8_l1_l4_v20_cisco_3node_s3.csv'... OK
[2019-06-03 09:41:21,113] Extracting platform description data... OK
[2019-06-03 09:41:21,114] Checking that bootstrap is run from supervisor server... OK
[2019-06-03 09:41:23,137] Generating the salt roster file '/etc/salt/roster'... OK
[2019-06-03 09:41:23,292] Preparing and testing SSH connection on every machine... OK
[2019-06-03 09:41:26,749] Performing server OS version correspondence check... OK
[2019-06-03 09:41:29,327] Checking iptables rules on every machine... OK
[2019-06-03 09:41:33,489] Generating the pillars for the install... OK
[2019-06-03 09:41:41,808] Installing scality-setup-httpd on 'supervisor'... OK
[2019-06-03 09:41:49,188] Setting up the new repository definitions on every machine... OK
[2019-06-03 09:42:06,367] Tuning servers operating system |#####| 100.0% -
ETA: 0:00:00
[2019-06-03 09:42:33,004] Tuning servers operating system... OK
[2019-06-03 09:42:33,004] Warning: Servers that need to be rebooted:
[2019-06-03 09:42:33,004] Warning: storage02, storage03, storage01, supervisor
[2019-06-03 09:42:33,004] storage02: 0 failed, 36 skipped, 3 fixed
[2019-06-03 09:42:33,004] storage03: 0 failed, 36 skipped, 3 fixed
[2019-06-03 09:42:33,004] storage01: 0 failed, 36 skipped, 3 fixed
[2019-06-03 09:42:33,004] supervisor: 0 failed, 36 skipped, 3 fixed
[2019-06-03 09:42:33,004] Details in /var/log/scality/setup/ostuning.log
[2019-06-03 09:42:36,730] Configuring logging on 'supervisor'... OK
[2019-06-03 09:42:52,460] Configuring Scality SSH on every machine... OK
[2019-06-03 09:43:00,782] Installing sreport on every machine... OK
[2019-06-03 09:43:12,840] Installing hardware monitoring dependencies on every machine... OK
[2019-06-03 09:43:25,662] Installing salt-master on 'supervisor'... OK
[2019-06-03 09:43:43,210] Installing SaltAPI on 'supervisor'... OK
[2019-06-03 09:43:56,199] Installing salt-minion on every machine... OK
[2019-06-03 09:44:17,976] Accepting minion key(s) on the master instance... OK
[2019-06-03 09:44:47,652] Syncing configuration on every machine... OK
[2019-06-03 09:44:54,554] Cleaning roles on every machine... OK
[2019-06-03 09:44:55,542] Setting up roles on every machine... OK
[2019-06-03 09:45:06,078] Installing python Scality on every machine... OK
[2019-06-03 09:45:12,427] Generating Rings models for keySPACE... OK
[2019-06-03 09:45:14,945] Installing and configuring scaldisk on every machine... OK
[2019-06-03 09:45:22,120] Preparing disks for installation... OK

Warning: Tuning OS: Some changes require a reboot, the following servers need to be rebooted: storage02,
storage03, storage01, supervisor

-- Bootstrap step successful, duration: 0:04:08.050564 --
[2019-06-03 09:45:23,518] The bootstrap step finished successfully...
```



Press [Enter] to return to the menu or [Ctrl]+c to exit the installer

After the Prepare the environment phase has completed, the servers may require a reboot as indicated in red above. Reboot the servers before proceeding to the next phase of the installation.

## Run the Pre-Install Suite

After rebooting the servers, the installer can be relaunched with the following command:

```
# /srv/scality/bin/launcher
```

Execute Run the Pre-Install Suite menu command to check the availability and accessibility of hardware servers and components as defined in the Platform Description File. In addition, the Pre-Install Suite also checks and updates OS settings per Scality recommendations.

Scality Installer Menu

```
-----
1. Prepare the environment
2. Run the Pre-Install Suite
3. Install Scality RING
4. Install S3 Service (Optional)
5. Run the Post-Install Suite

* Generate the Offline Archive (Optional)
* Reset SSH credentials
* Gather all logs and configuration files

Exit
```

```
----- Description -----

Run the Pre-Install Suite to check the availability and accessibility of
hardware servers and components, as defined in the Platform Description File
(the CSV/XLS file provided to the Installer).
The suite also checks and updates OS settings per Scality recommendations,
as described in the Scality Setup and Installation Guide.
```



Critical errors detected by the pre-install suite should be addressed before proceeding.

## Install Scality RING

To install the Scality RING, follow these steps:

1. Initiate the Install Scality RING menu command to install the Scality RING as described in the Platform Description File.

## Scality Installer Menu

```

1. Prepare the environment
2. Run the Pre-Install Suite
3. Install Scality RING
4. Install S3 Service (Optional)
5. Run the Post-Install Suite

* Generate the Offline Archive (Optional)
* Reset SSH credentials
* Gather all logs and configuration files

Exit

```

## Description

Install Scality RING and all necessary components on every node, as described in the Platform Description File (the CSV/XLS file provided to the Installer).

```

[2019-06-04 11:50:32,583] INFO      - Launching install, this might take some time
[2019-06-04 11:50:32,613] <salt> Clear the cache and sync modules, grains and pillar ... OK
[2019-06-04 11:50:39,564] <roles> Check storage nodes minions matcher ... OK
[2019-06-04 11:50:39,567] <roles> Ensure grains is deleted everywhere ... OK
[2019-06-04 11:50:41,062] <roles> Setup supervisor role ... OK
[2019-06-04 11:50:41,893] <roles> Setup storage nodes role ... OK
[2019-06-04 11:50:47,069] <roles> Setup elasticsearch cluster role ... OK
[2019-06-04 11:50:47,808] <roles> Advertise elasticsearch cluster ... OK
[2019-06-04 11:50:55,070] <roles> Setup S3 role ... OK
[2019-06-04 11:50:59,891] <roles> Setup zookeeper role ... OK
[2019-06-04 11:51:12,714] <roles> Setup HALO role ... OK
[2019-06-04 11:51:17,398] <setup> Start scality-setup-httpd ... OK
[2019-06-04 11:51:23,086] <setup> Install python-scality ... OK
[2019-06-04 11:51:29,755] <setup> Install python-scaldisk ... OK
[2019-06-04 11:51:36,675] <setup> Install sreport ... OK
[2019-06-04 11:51:47,402] <setup> Detect the disks ... OK
[2019-06-04 11:51:48,554] <setup> Publish disks infos ... OK
[2019-06-04 11:51:50,193] <sup> Install and configure supervisor ... OK
[2019-06-04 11:53:23,317] <rings> Compute the keypace ... OK
[2019-06-04 11:53:24,805] <rings> Configure the rings on the supervisor ... OK
[2019-06-04 11:54:02,704] <elastic> Install and configure elasticsearch cluster ... OK
[2019-06-04 11:54:33,796] <supapi> Configure the supapi service ... OK
[2019-06-04 11:54:48,335] <supapi> Install the cloud monitoring service ... OK
[2019-06-04 11:54:57,205] <disks> Partition and format disks ... OK
[2019-06-04 12:04:59,967] <disks> Mount all disks ... OK
[2019-06-04 12:05:06,934] <nodes> Advertise zookeeper cluster ... OK
[2019-06-04 12:05:17,832] <nodes> Install and configure storage nodes ... OK
[2019-06-04 12:07:08,012] <nodes> Install and configure zookeeper ... OK
[2019-06-04 12:07:47,966] <keyspace> Spread the keypace to storage nodes ... OK
[2019-06-04 12:07:51,517] <keyspace> Make storage nodes join rings ... OK
[2019-06-04 12:08:07,345] <conns> Install Scality Agent Daemon (sagentd) on S3 connectors ... OK
[2019-06-04 12:08:42,967] <post> Install and configure ringsh ... OK
[2019-06-04 12:08:50,445] <post> Backup the whole platform ... OK
[2019-06-04 12:08:56,067] <post> Install external tools ... OK
[2019-06-04 12:09:06,904] INFO      - Install completed without errors
[2019-06-04 12:09:06,905] INFO      - RING installed successfully
[2019-06-04 12:09:06,971] The install step finished successfully...

```

Press [Enter] to return to the menu or [Ctrl]+c to exit the installer

## Install S3 Connector Service

To install the S3 connector service, follow these steps:

1. The Install the S3 Service (Optional) menu command installs the S3 Connector components on the nodes as described in the Platform Description File.

#### Scality Installer Menu

```

1. Prepare the environment
2. Run the Pre-Install Suite
3. Install Scality RING
4. Install S3 Service (Optional)
5. Run the Post-Install Suite

* Generate the Offline Archive (Optional)
* Reset SSH credentials
* Gather all logs and configuration files

Exit

```

#### Description

Install the S3 components on the nodes, as described in the Platform Description File (the CSV/XLS file provided to the Installer).

```

[2019-06-04 12:33:52,813] Searching S3 offline archive file... OK
[2019-06-04 12:33:52,813] Extracting S3 offline archive... OK
[2019-06-04 12:34:27,044] Generating the S3 inventory from platform description file... OK
[2019-06-04 12:34:41,375] Installing sshpass package... OK
[2019-06-04 12:34:45,666] Generating vault environment configuration... OK
[2019-06-04 12:34:53,369] Running S3 ansible playbook to install the S3 connector... OK
[2019-06-04 12:48:53,458] Setting up the identisee credentials... OK
[2019-06-04 12:49:13,734] The s3 step finished successfully...

```

Press [Enter] to return to the menu or [Ctrl]+c to exit the installer

## Run the Post-Install Suite

To run the post-install suite, follow these steps:

1. Issue the Run the Post-Install Suite menu command to validate the installation.

#### Scality Installer Menu

```

1. Prepare the environment
2. Run the Pre-Install Suite
3. Install Scality RING
4. Install S3 Service (Optional)
5. Run the Post-Install Suite

* Generate the Offline Archive (Optional)
* Reset SSH credentials
* Gather all logs and configuration files

Exit

```

#### Description

Run the Post-Install Suite on the platform to validate the installation.

```

[2019-06-04 12:53:34,402] Setting up the new repositories definitions on every machine ... OK
[2019-06-04 12:53:39,782] Installing the postinstallchecks ... OK
Running the postinstallchecks
Running script using salt
Starting checks on storage02,storage03,storage01,supervisor
Checking if server is handled by salt

```

```
Checking missing pillars
Gathering info from servers (salt mine.send) for consistency check later
Running tests
The result is found in: /srv/scality/s3/post-install-checks-results.tgz
[2019-06-04 12:54:58,830] The postinstall step finished successfully...

Press [Enter] to return to the menu or [Ctrl]+c to exit the installer
```

The results of the Post-Install Suite should be shared with your Scality Service or Support representative for review. The results can be found at `/root/post-install-checks-results.tgz`

## Managing and Monitoring Scality RING

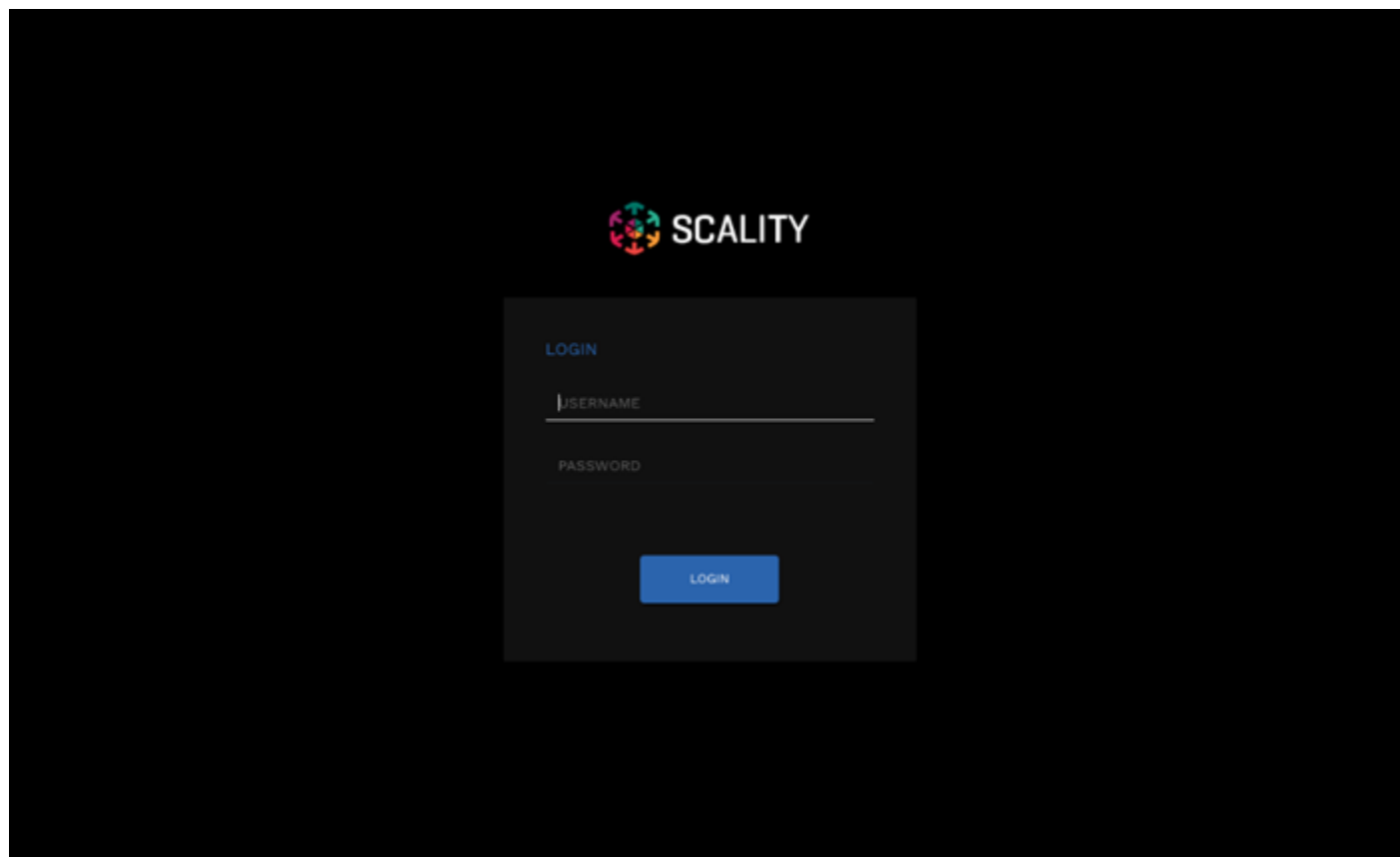
Managing and monitoring the RING is enabled through a cohesive suite of user interfaces, built on top of a family of RESTful interfaces termed the Supervisor API (SupAPI). RING 7 includes the new Scality Supervisor, a browser-based portal for both systems monitoring and management of Scality components. In RING 7, the Supervisor now provides capabilities across object (S3) and file (NFS, SMB, FUSE) Connectors including integrated dashboards including Key Performance Indicators (KPIs) with trending information such as Global Health, Performance, Availability and Forecast.

The Scality Supervisor can be accessed in a browser via the IP address of the Supervisor server.

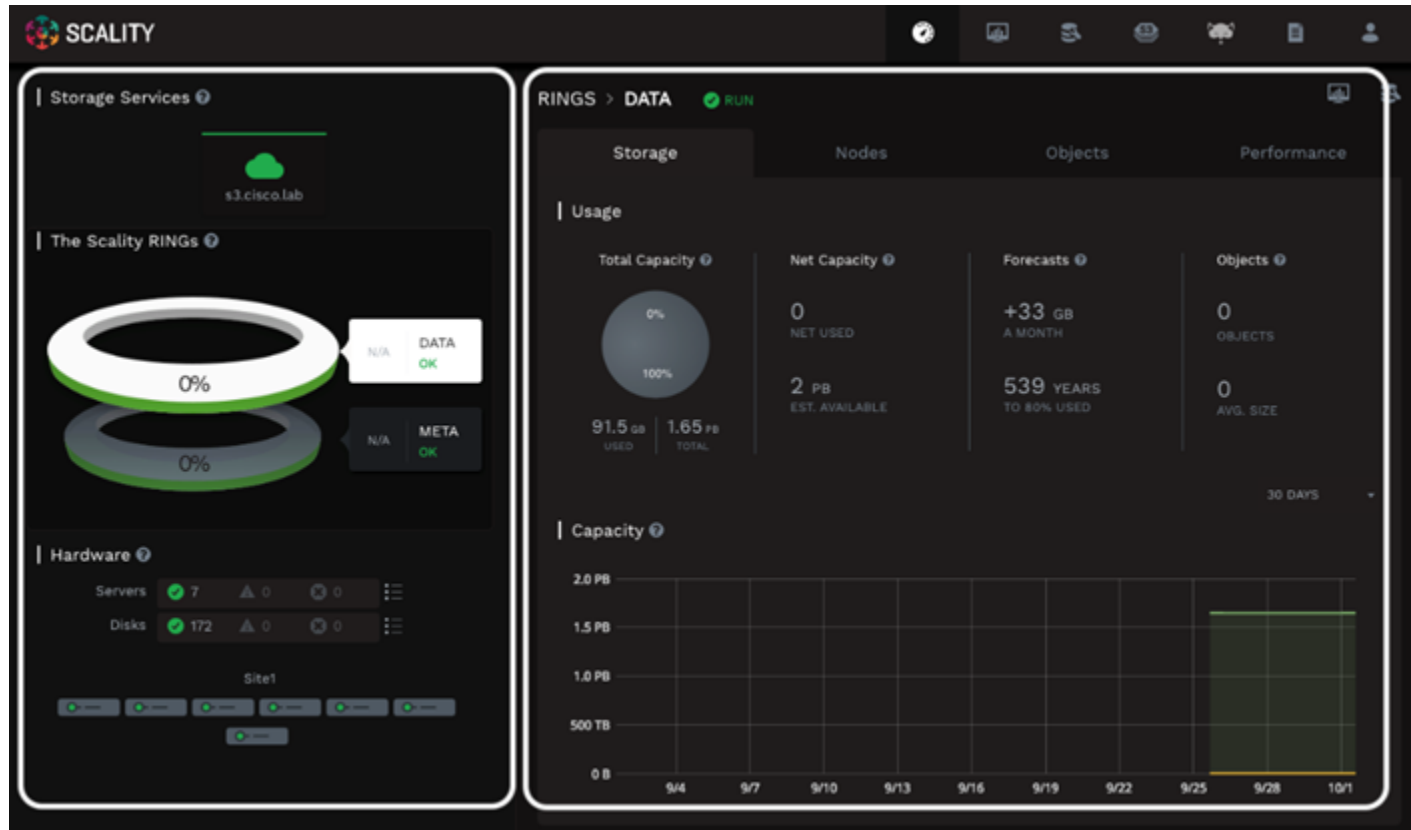
### Monitor Scality RING

To monitor the Scality RING, follow these steps:

1. Launch the Scality Supervisor by navigating to `http://<supervisor IP>/gui`.



2. Login using the user admin and the password provided during the Preparing the Environment step of the installation.



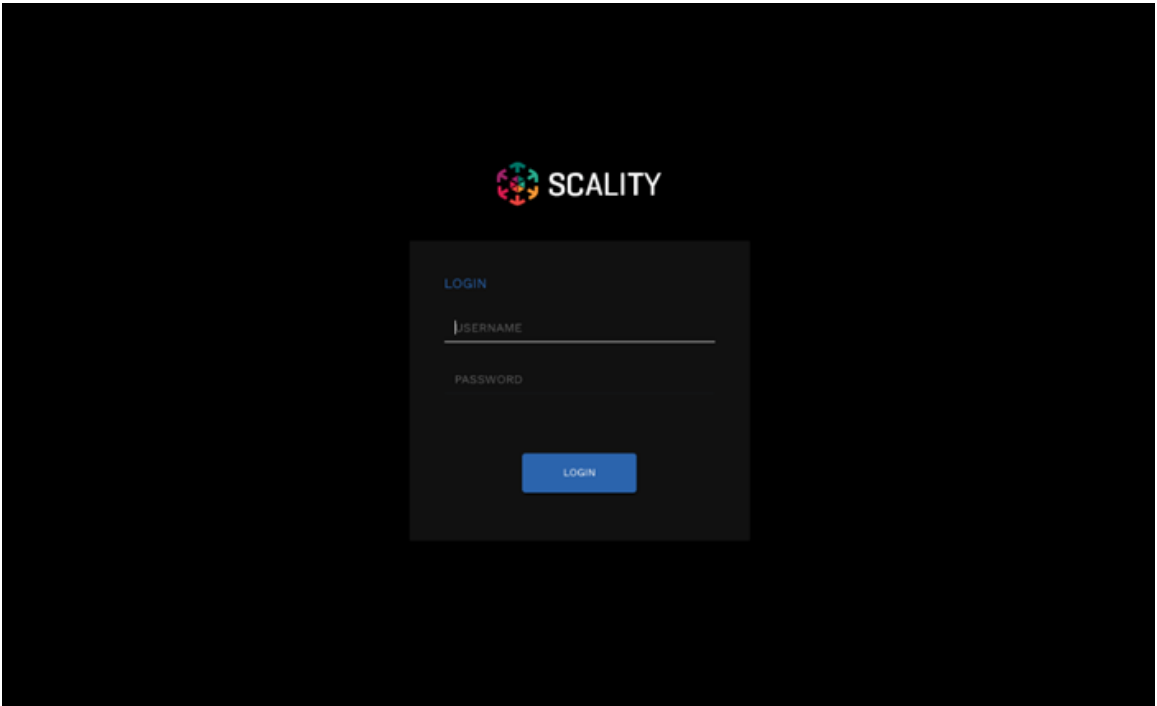
The Component Panel (left) provides an overview of the overall platform health including hardware and services. Services in a failed or critical state will be colored red indicating attention is needed.

The Information Screen (right) provides a capacity overview. The Forecasts section provides the storage administrator with a projected time to 80% full.

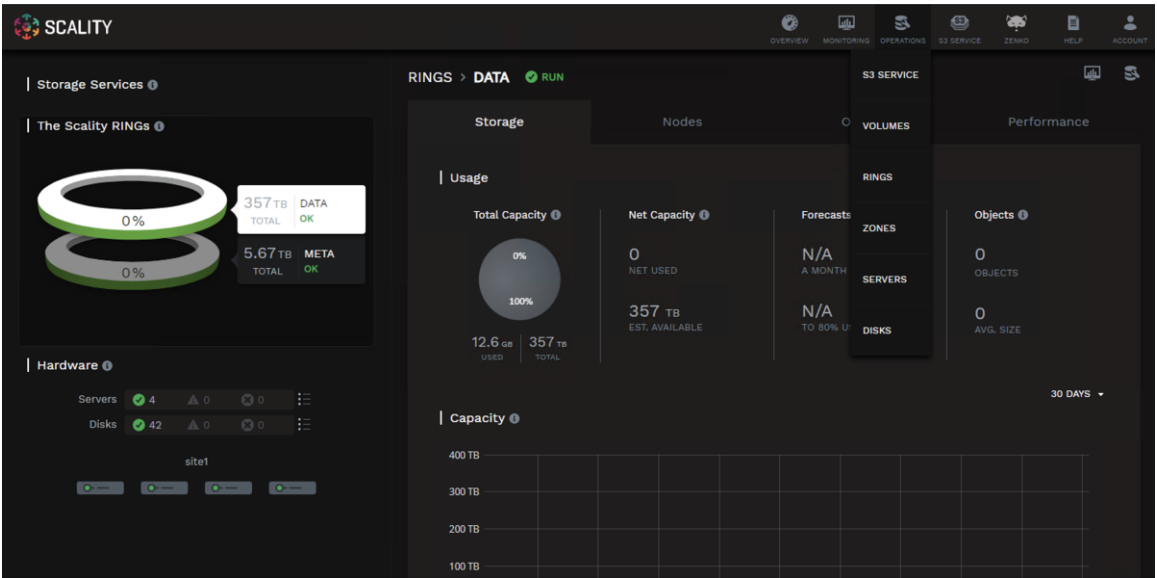
## Manage NFS Connectors

To configure and access NFS exports on the Scality Scale-Out Filesystem (SOFS), follow these steps:

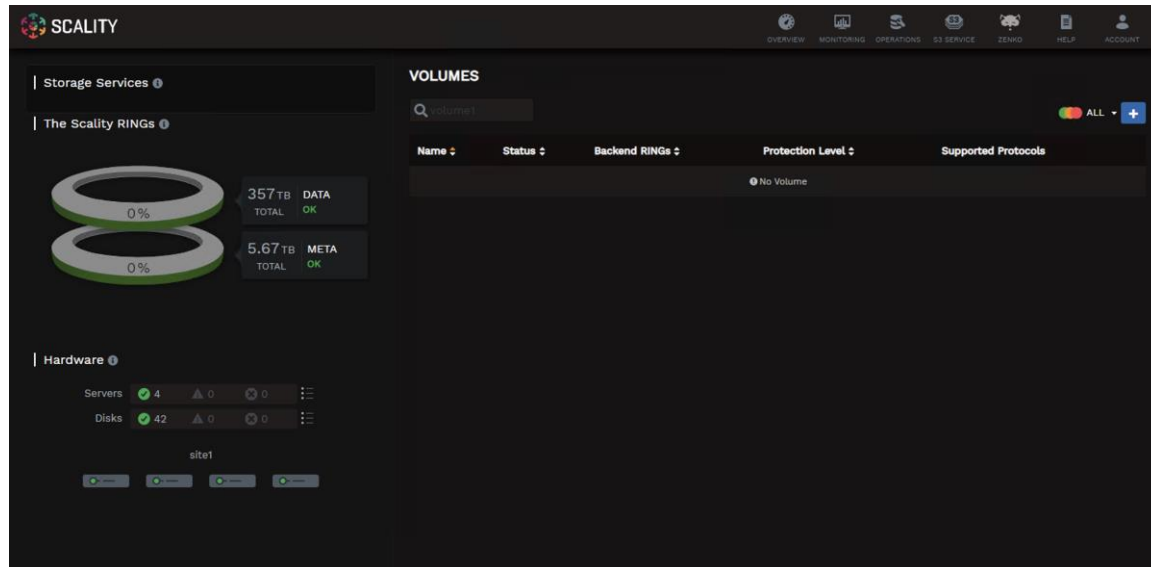
1. Volumes are created through the RING Supervisor UI. To access the RING Supervisor UI navigate to <http://<supervisor IP>/gui>.
2. Login using the user admin and the password provided during the Preparing the Environment step of the installation.



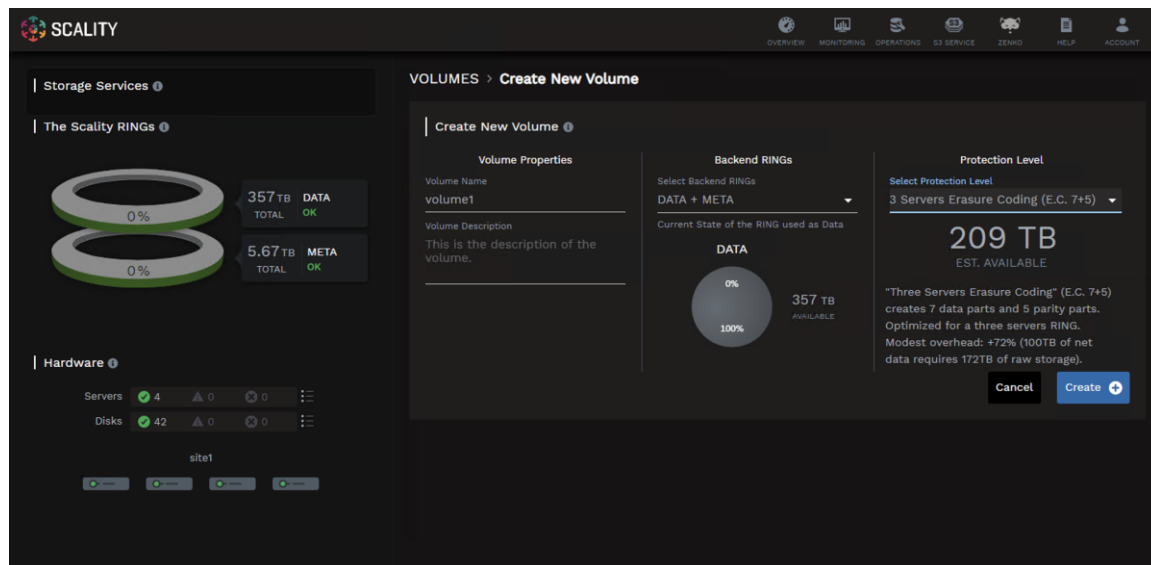
3.   Navigate to the Operations>Volumes menu.



4.   Click the + button to create a new volume.

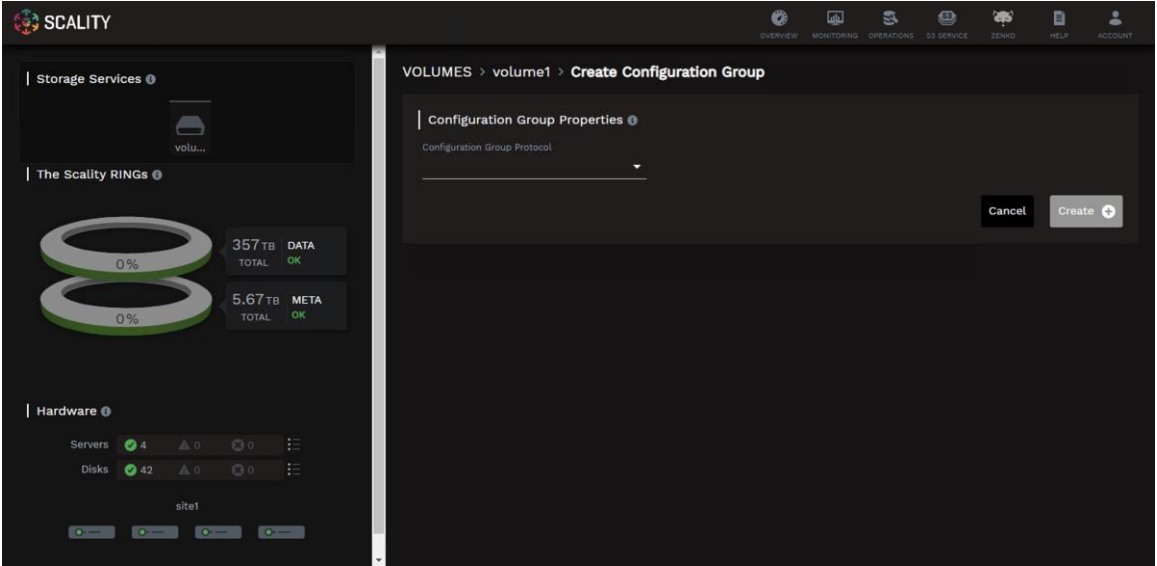


5. Provide a volume name, select the backend RINGS, and protection level then click Create.
  - Name: volume1
  - Backend RINGS: DATA+META
  - Protection Level: 3 Servers Erasure Coding (E.C. 7+5)

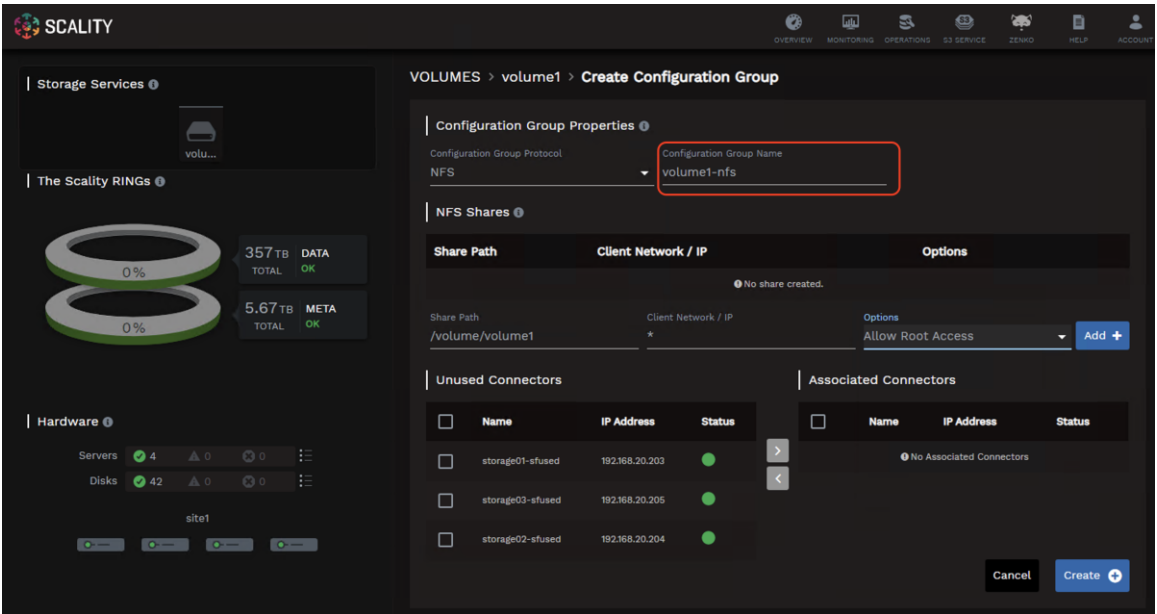


6. Configuration Groups allow the user to group connectors with identical configurations. A new Configuration Group must be created for the NFS connectors. Select NFS from the drop-down list.



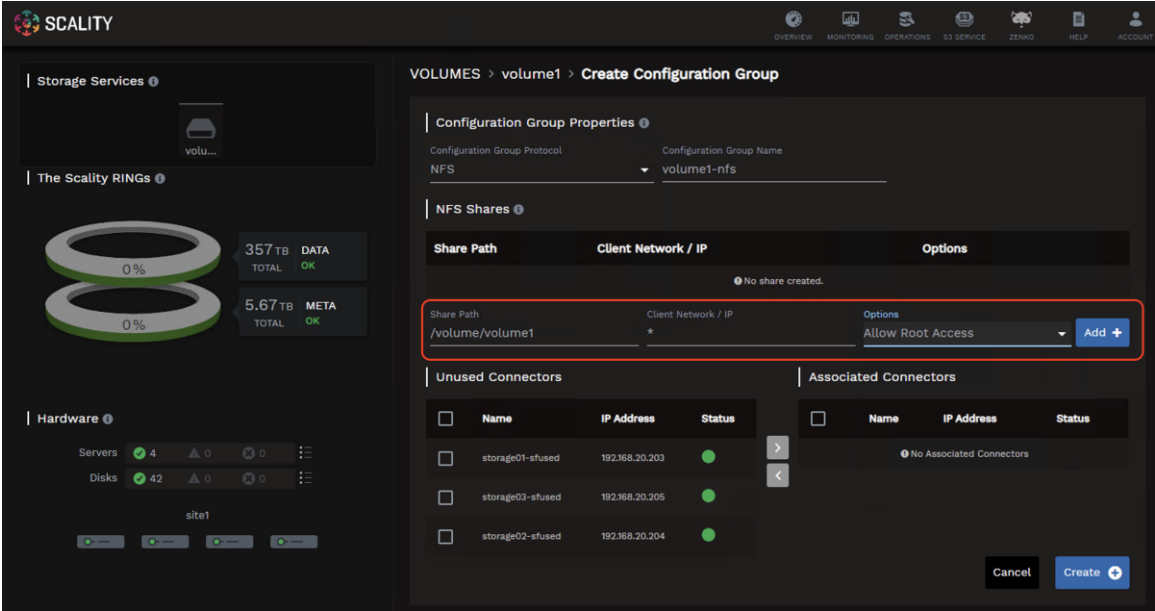


7. Provide a configuration group name.

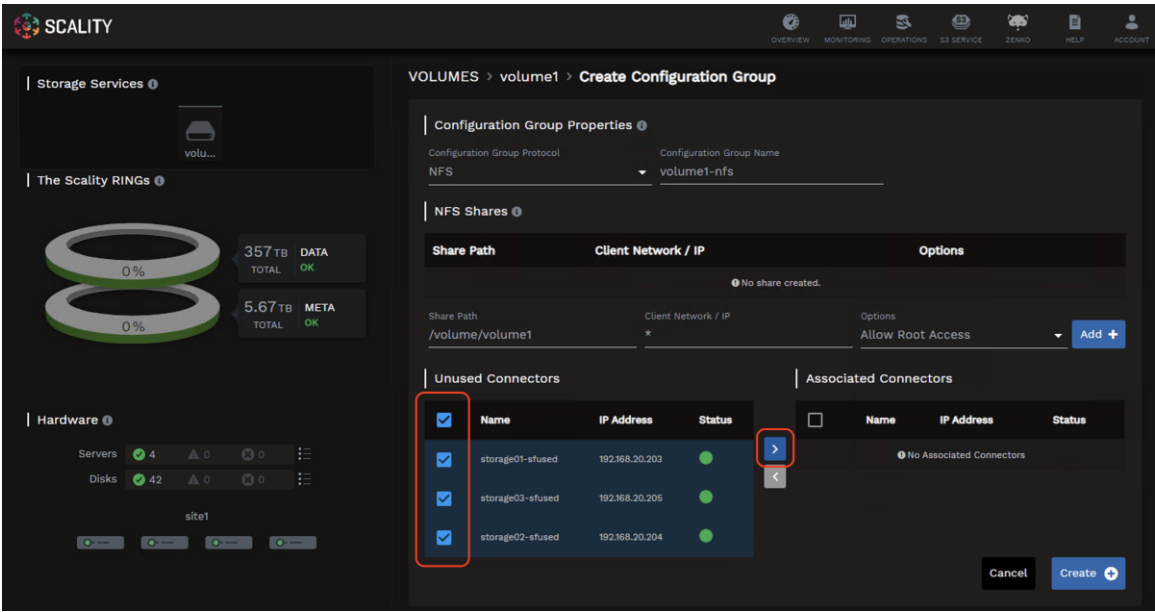


8. Define the share details and click Add +:

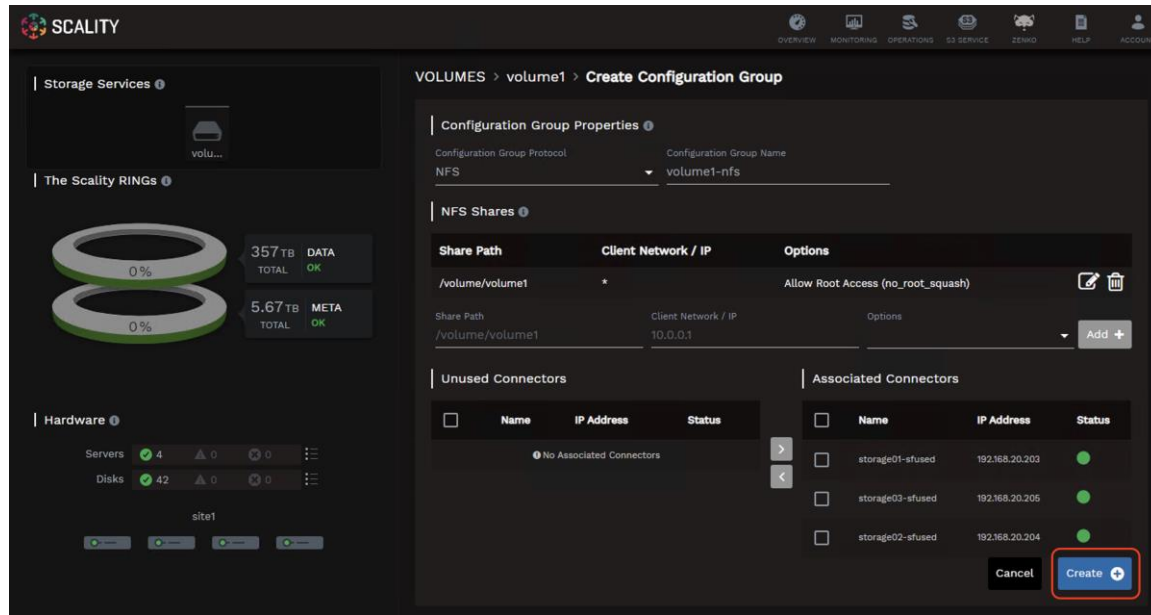
- Share Path: /volume/volume1
- Client Network/IP: \*
- Options: Allow Root Access



9. Associate the unused connectors with the configuration group.



10. Click Create.

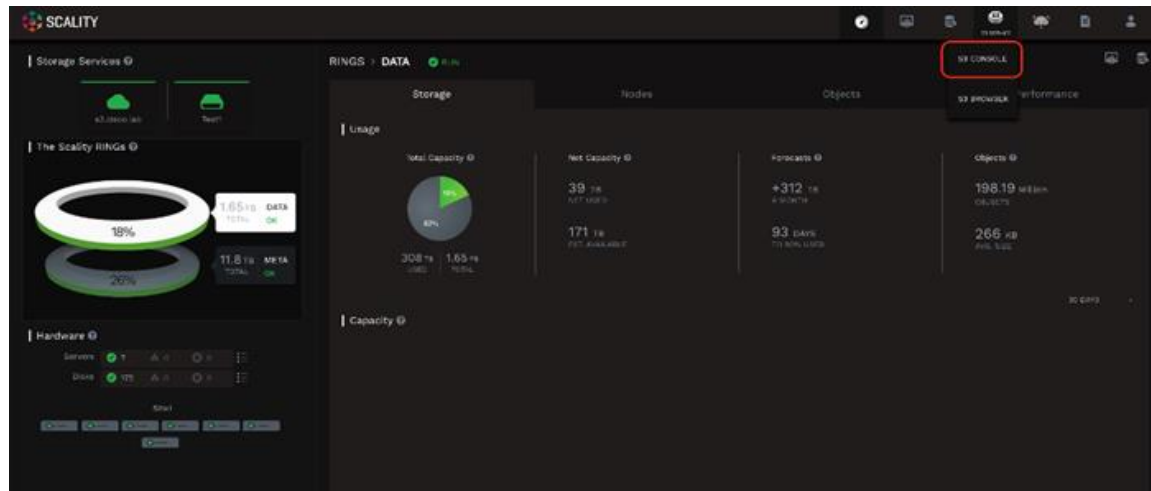


## Manage S3 Connectors

The Scality S3 Connector provides a full implementation of the AWS multi-tenancy and identity management (AWS IAM) model with federated authentication to LDAP and Active Directory to integrate into enterprise deployment environments. In addition to the RING Supervisor management UI, the S3 Service Provider UI is a web-based user interface to manage multi-tenancy accounts, users, group and policies.

To create a new S3 account through the S3 console, follow these steps:

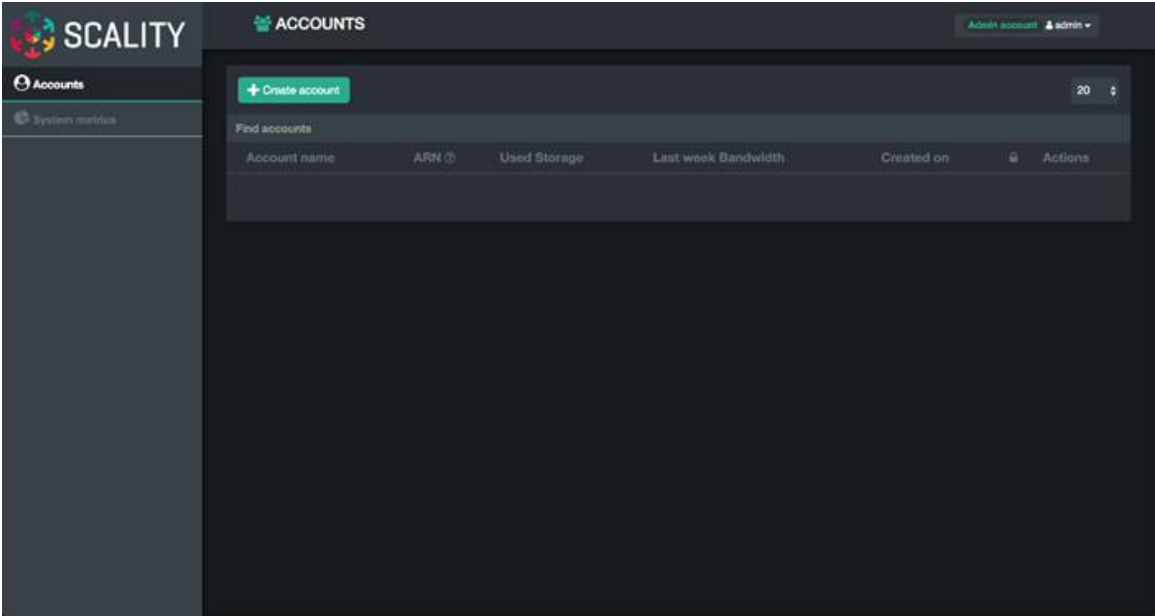
1. To connect to the S3 Console navigate to the S3 Service drop-down list in the Scality Supervisor and select S3 Console.



2. Login using the user admin and the password provided during the Preparing the Environment step of the installation.



3. Click the Create account button to create a new S3 account.



4. Provide an account name, email and password then click Submit.

Create account

Account name

cisco

Email address

cisco@cisco.com

Password

\*\*\*\*\*

Password confirmation

\*\*\*\*\*

Submit

5. The new account will show up in the accounts table.

SCALITY

ACCOUNTS

Accounts

System Health

Create account

20

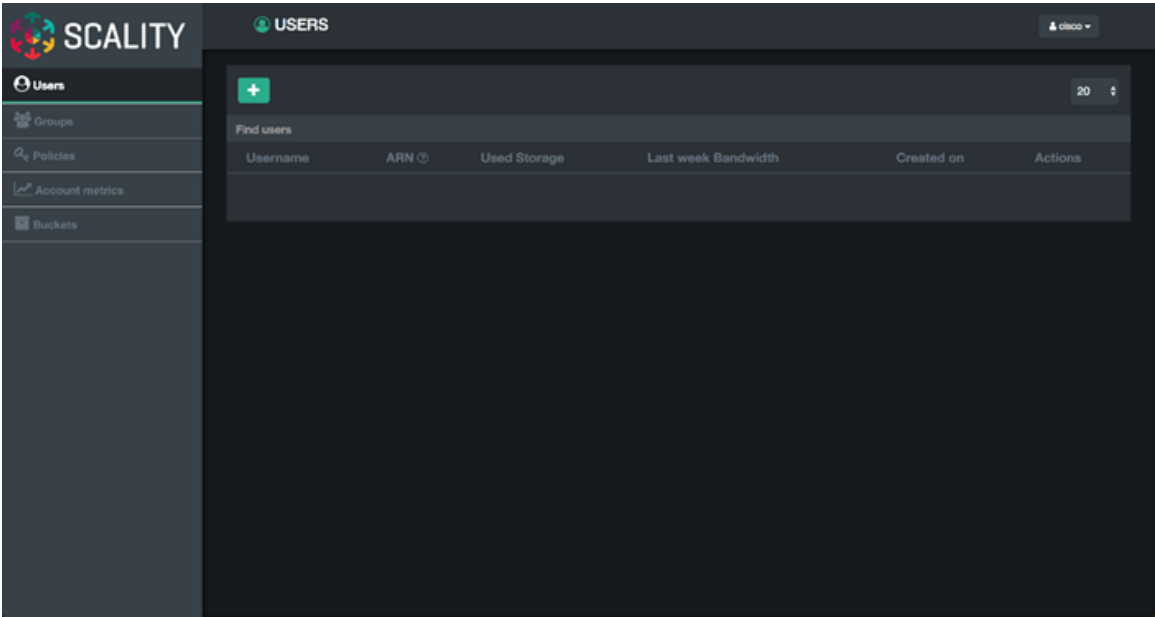
Find accounts

Account name	ARN	Used Storage	Last week Bandwidth	Created on		Actions
cisco		0B	0B / 0B	Video Oct 10 2018 12:37:36 PM		<div></div>

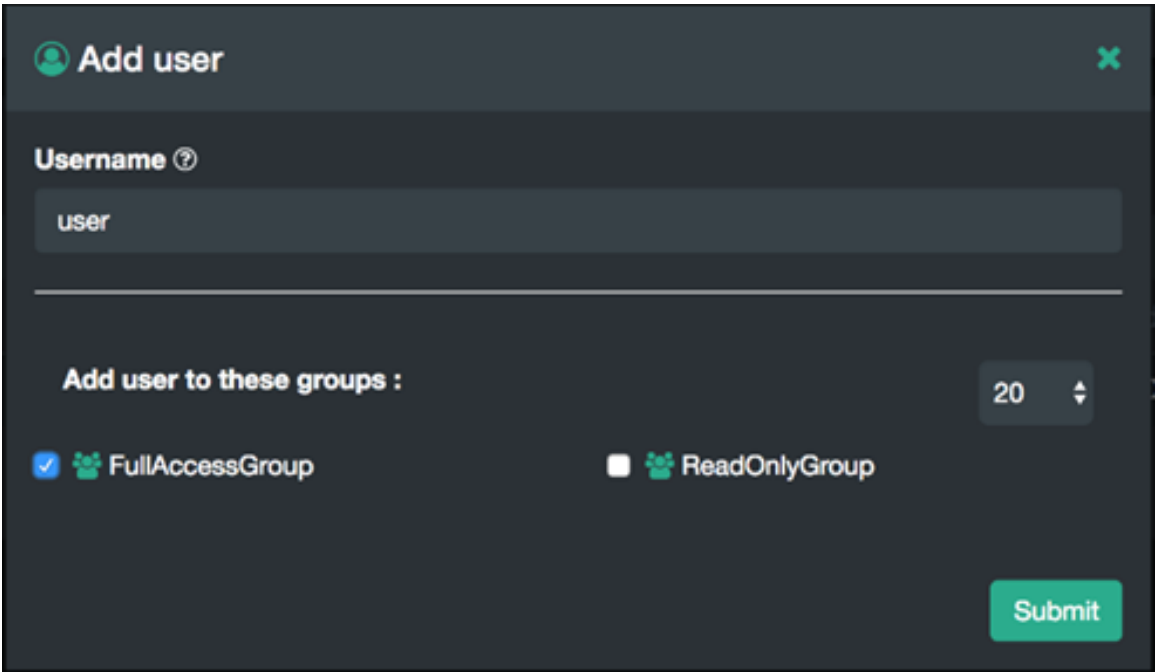


The S3 Console can be used to manage users. To create a new user under the ‘cisco’ account created in the previous section login to the S3 Console using the user ‘cisco’ and the password provided.

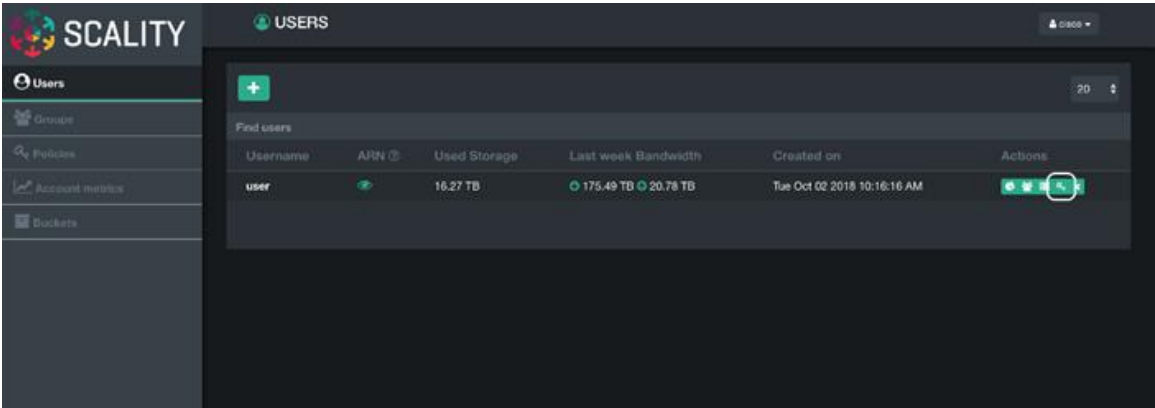
6. Click the + button to create a new user.



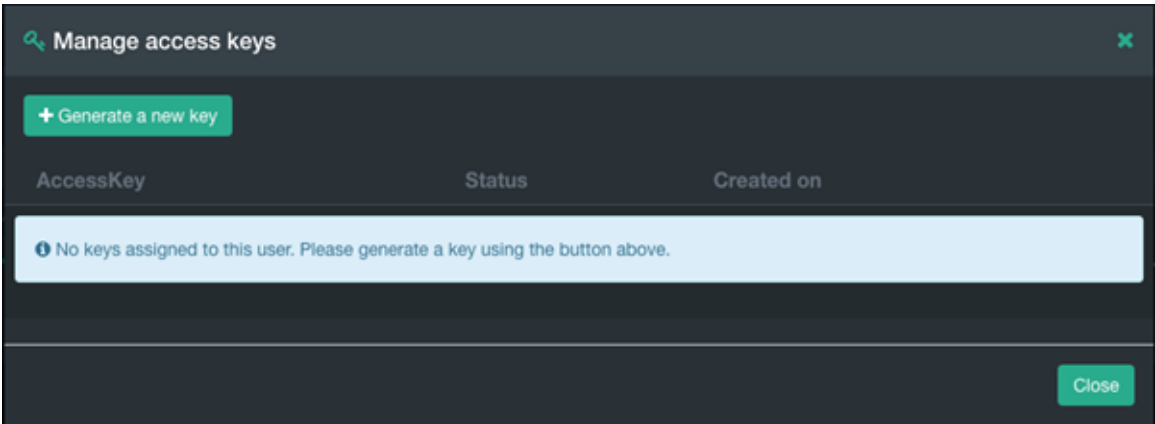
7. Provide a Username, select the FullAccessGroup to grant the user full permissions, and click Submit.



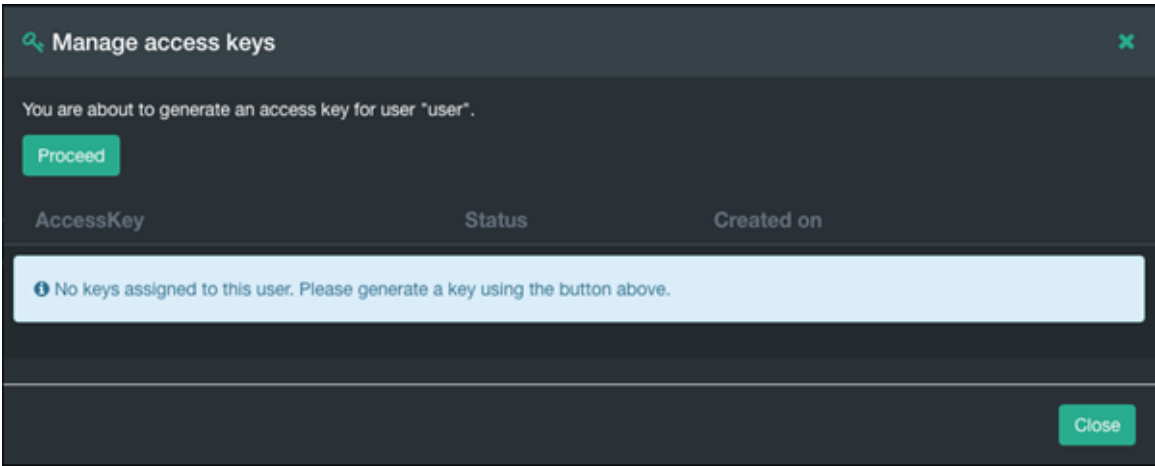
- 8. After clicking Submit the user will appear in the users table.
- 9. To generate the secret key and accesskey required to end S3 requests click the Users Key icon in the Actions column.



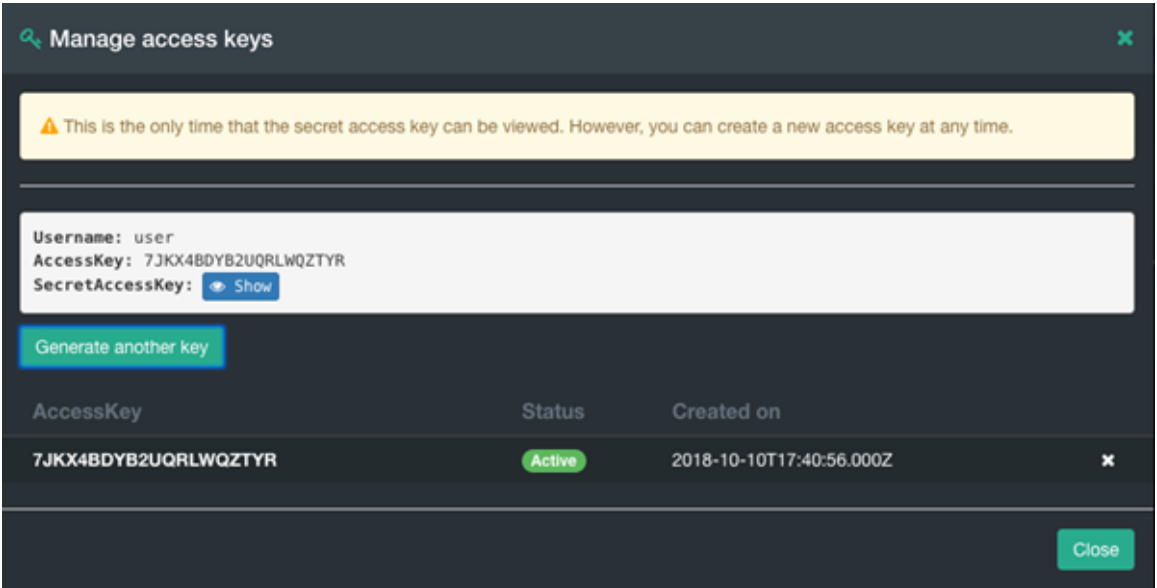
10. Click the Generate a new key button to generate the secret key and access key.



11. Confirm the key generation by clicking Proceed.

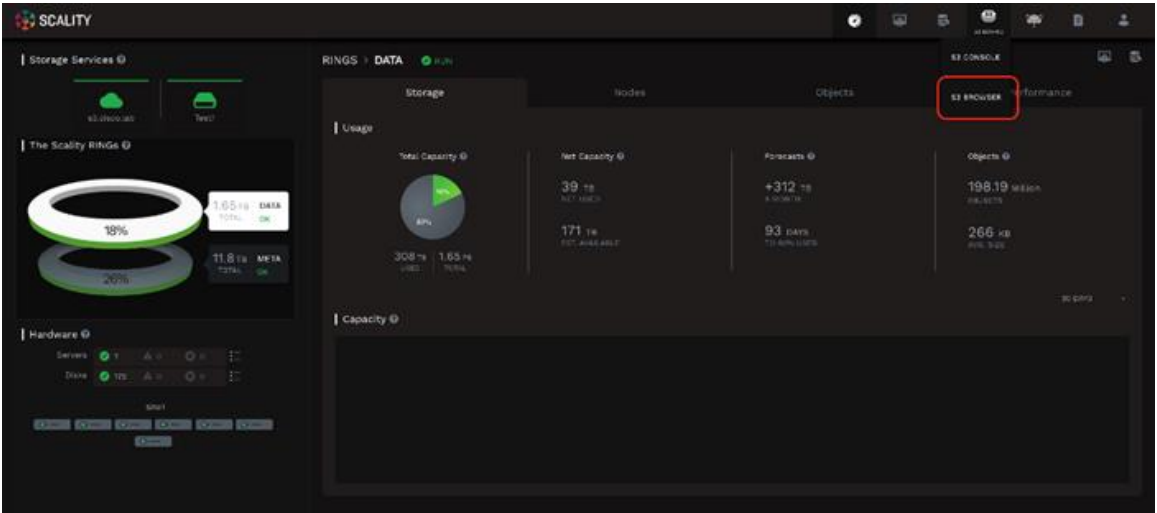


For security reasons the secret key can only be viewed at creation time. Record the secret key in a safe location. If the secret key is lost generate a new secret key and access key can be generated from the S3 Console.



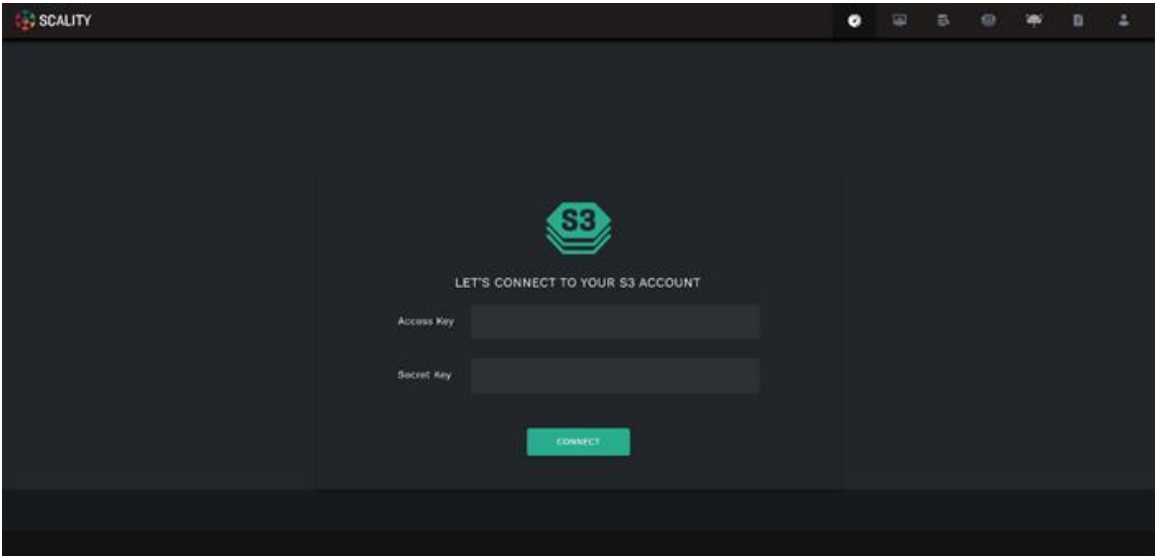
The AccessKey and SecretAccessKey can be used to connect to the S3 endpoint using command line tools like awscli or s3cmd or directly in any application which supports the S3 API.

Scality also provides an S3 Browser that can be used browse or create buckets and upload objects. To connect to the S3 Console navigate to the S3 Browser drop-down list in the Scality Supervisor and select S3 Browser.

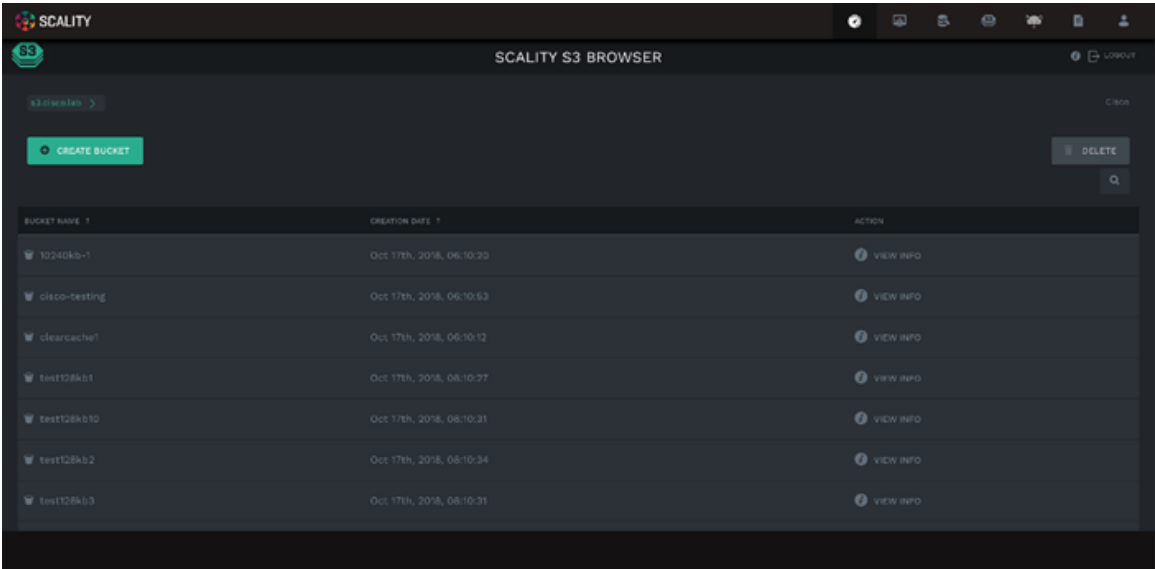


The AccessKey and SecretKey can be used to login.





You can browse, upload, download, and delete buckets and objects.

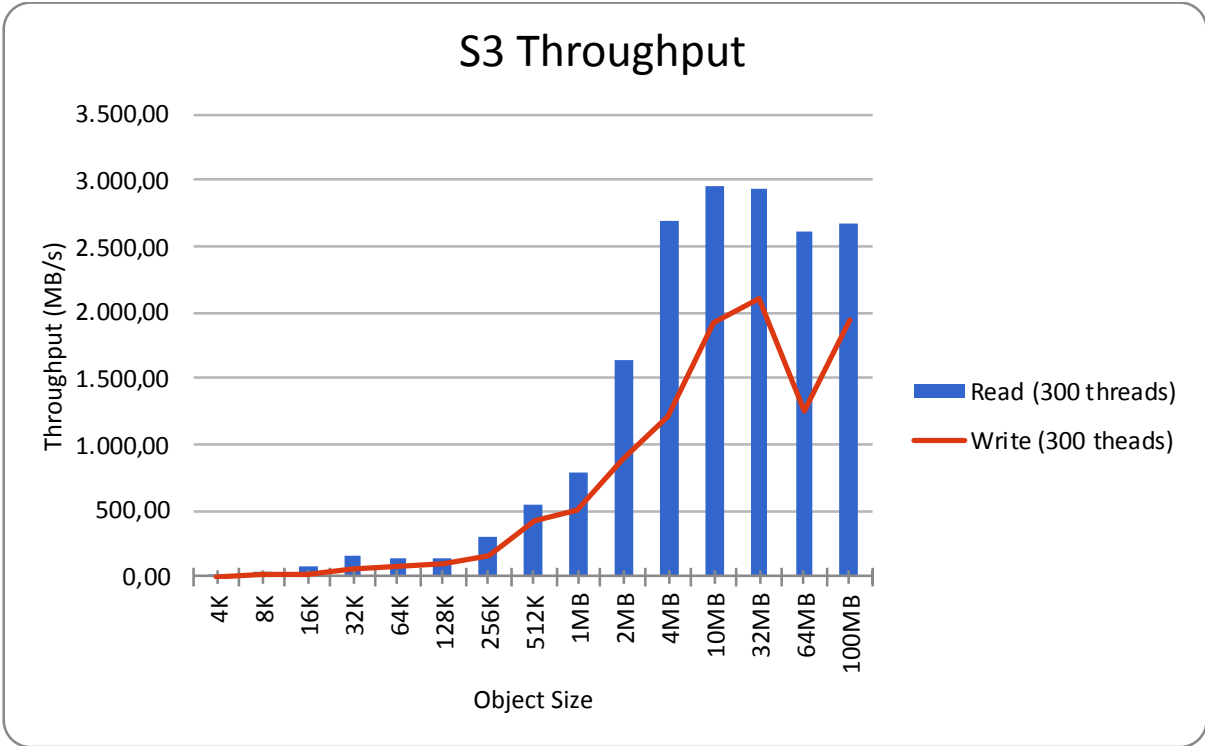


# Scality RING Performance Testing

Performance was evaluated on the Scality RING running on Cisco M5 UCS hardware. The goal of the performance testing was to evaluate peak object performance under ideal conditions.

## S3 Performance Tests

S3 performance testing was conducted with COSBench the standard cloud object storage benchmark. Three Cisco UCS servers were used as COSBench drivers to generate the object workload.



- Read bandwidth peaks at 2.95 GB/s at an object size of 10MB. This translates to a disk performance of 82 MB/s/disk.
- Write bandwidth peaks at 2.11 GB/s at an object size of 32MB. This translates to a disk performance of 59 MB/s/disk.

The drop-off after 32MB object size comes with the split chunk size in sproxyd, which is 32 MB. Objects are then written more than once.

## Scality RING High Availability Testing

It is important for business continuity to help ensure high availability of the hardware and software stack. Some of these features are built into the Cisco UCS Infrastructure and enabled by the software stack and some of these features are possible from the Scality RING software itself. In order to properly test for high availability, the following considerations were given priority:

- The Scality RING deployment will process a reasonable amount of load when the fault is triggered. Total throughput will be recorded from the COSBench interface.
- Only a single fault will be triggered at any given time. Double failure is not a part of this consideration.
- Performance degradation is acceptable and even expected, but there should be no business interruption tolerated. The underlying infrastructure components should continue to operate within the remaining environment.

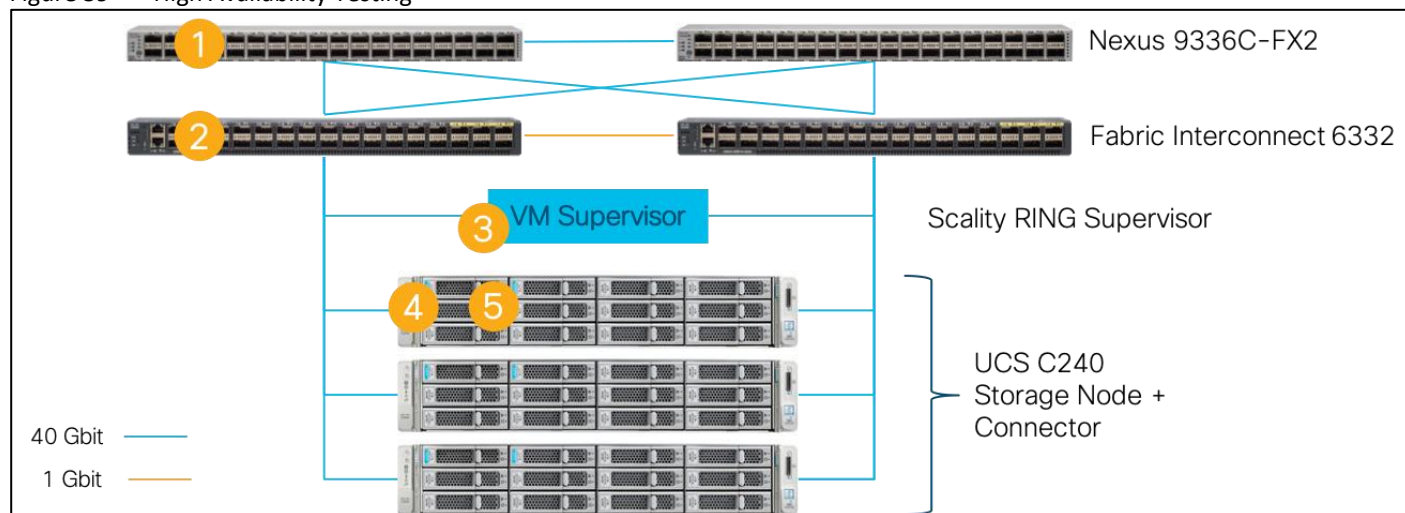
The following High Availability tests were performed:

- Cisco Nexus 9336C-FX2 Switch A failure
- Cisco UCS 6332 Fabric Interconnect A failure
- Scality RING Supervisor VM failure
- Cisco UCS C240 M5L – Scality RING storage-node 1 disk failure
- Cisco UCS C240 M5L – Scality RING storage-node 1 node failure

As indicated previously, a reasonable amount of load will be defined as follows:

- The COSBench application will be configured to send a steady stream of data to the Scality RING cluster.

Figure 59 High Availability Testing



## Cisco Nexus 9336C-FX2 High Availability Testing

### Sequence of Events

1. Connect to Cisco Nexus 9336 Switch A and make certain running-config is copied to startup-config to make certain no configuration changes are lost during power cycle.

```
Scality-CVD-9336C-FX2-A# copy run start
[#####] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
```

2. Initiate load to the cluster by utilizing COSBench.
3. Pull out the power cables from Nexus switch A and wait for at least 5 minutes before plugging in the power cables.



The load continues during Cisco Nexus 9336C-FX2 reboot process.

---

Aside from loss of response from Nexus 9336 switch, Scality RING environment remained functional, load continued at constant rate, and redundancy was reestablished upon Switch A completing the reboot process.

## Cisco UCS Fabric Interconnect 6332 High Availability Testing

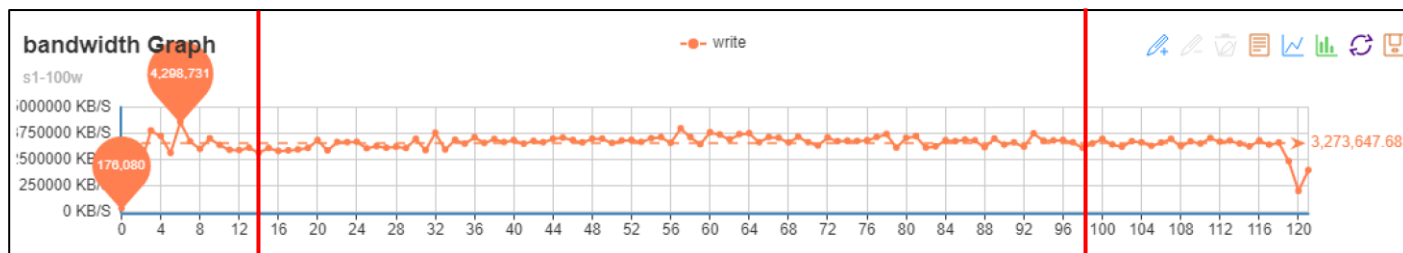
### Sequence of Events

1. Connect to Cisco UCS Manager and verify that both Fabric Interconnects and all nodes are in a known good working condition.
2. Initiate load to the cluster by utilizing COSBench.
3. Initiate reboot of Fabric Interconnect A. Establish a secure shell session to Fabric Interconnect A and enter the following commands.

```
connect local-mgmt
reboot
```

4. The Fabric Interconnect can take as long as 5 minutes to completely initialize after a reboot. Wait the entire amount of time for this process to complete.

The graph below is a snapshot from COSBench. The first vertical red line is where Fabric Interconnect A was rebooted. No loss in throughput was observed that could be within the noise of run-to-run variation. The total workload took place over the course of 10 minutes with ample time for Fabric Interconnect A to properly return to a known good state. The second red line is where all the connections went back online.

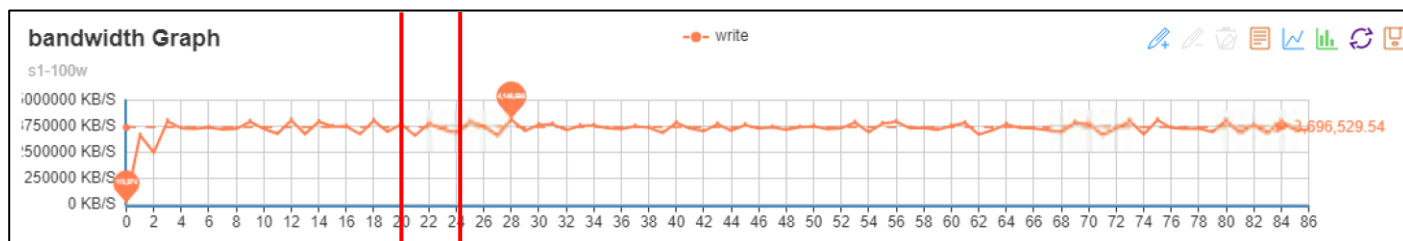


## Scality RING Supervisor VM Failure Testing

### Sequence of Events

1. Connect to Cisco UCS Manager and verify that both Fabric Interconnects and all nodes are in a known good working condition.
2. Initiate load to the cluster by utilizing COSBench.
3. Reboot Supervisor via the OS and wait for at least 5 minutes.

The graph below is a snapshot from COSBench. At the vertical first red line is where the Supervisor VM was rebooted. There was no drop at all and the overall write speed remained consistent. At the vertical second line is where the Supervisor VM was up and running again.

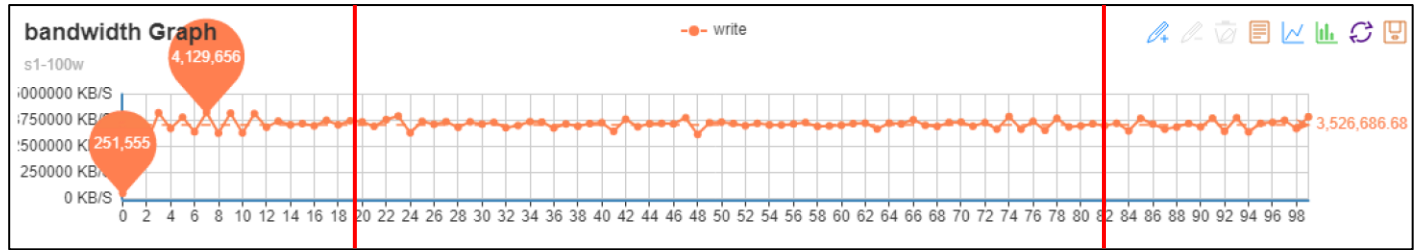


## Cisco UCS C240 M5L Disk Failure Testing

### Sequence of Events

1. Connect to Cisco UCS Manager and verify that both Fabric Interconnects and all nodes are in a known good working condition.
2. Initiate load to the cluster by utilizing COSBench.
3. Pull out one of 10 TB disks of storage-node1 and wait for at least 5 minutes before plugging it back in.

The graph below is a snapshot from COSBench. At the vertical first red line is where Disk 1 was pulled. There was no drop at all and the overall write speed remained consistent. At the vertical second line is where disk was plugged in again. Overall there was no drop in write performance.

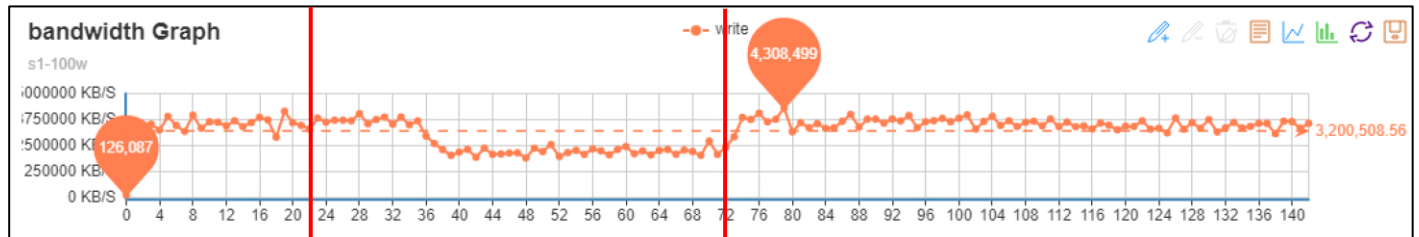


## Cisco UCS C240 M5L Node Failure Testing

### Sequence of Events

1. Connect to Cisco UCS Manager and verify that both Fabric Interconnects and all nodes are in a known good working condition.
2. Initiate load to the cluster by utilizing COSBench.
3. Reboot storage-node1 via the OS and wait for at least 10 minutes.

The graph below is a snapshot from COSBench. At the first vertical red line is where storage-node1 was rebooted. A loss in throughput of about 30 percent was observed. At the second red line the reboot of storage-node1 was finished and workload returned to a normal state. The total workload took place over the course of 15 minutes with ample time for storage-node1 to properly return to a known good state.



## Appendix

### Platform Description File

#### S3 Platform Description File

```
ring,,,,,,,,,,,,,,,,,,,,,

sizing_version,customer_name,#ring,data_ring_name,meta_ring_name,HALO API key,S3
endpoint,cos,arc-data,arc-coding,,,,,,,,,,,,,

20.4,Sample,1,DATA,,0,s3.scality.com,3,7,5,,,,,,,,,,,,,

,,,,,,,,,,,,,,,,,,,,,

servers,,,,,,,,,,,,,,,,,,,,,

data_ip,data_iface,mgmt_ip,mgmt_iface,s3_ip,s3_iface,svsd_ip,svsd_iface,ring_members
hip,role,minion_id,enclosure,site,#cpu,cpu,ram,#nic,nic_size,#os_disk,os_disk_size,#
data_disk,data_disk_size,#raid_card,raid_cache,raid_card_type,#ssd,ssd_size,#ssd_for
_s3,ssd_for_s3_size

192.168.20.203,eth2,192.168.10.203,eth1,,,,,DATA,"storage,s3,s3_md,zookeeper,elastic
",storage01,Cisco UCS C240 M5 - 3N,site1,2,Intel Xeon Bronze 3104 (1.7 GHz/6
cores),256,2,40,2,960,12,10000,1,2,Cisco 12Gbps Modular RAID PCIe Gen 3.0,2,960,0,0

192.168.20.204,eth2,192.168.10.204,eth1,,,,,DATA,"storage,s3,s3_md,zookeeper,elastic
",storage02,Cisco UCS C240 M5 - 3N,site1,2,Intel Xeon Bronze 3104 (1.7 GHz/6
cores),256,2,40,2,960,12,10000,1,2,Cisco 12Gbps Modular RAID PCIe Gen 3.0,2,960,0,0

192.168.20.205,eth2,192.168.10.205,eth1,,,,,DATA,"storage,s3,s3_md,zookeeper,elastic
",storage03,Cisco UCS C240 M5 - 3N,site1,2,Intel Xeon Bronze 3104 (1.7 GHz/6
cores),256,2,40,2,960,12,10000,1,2,Cisco 12Gbps Modular RAID PCIe Gen 3.0,2,960,0,0

192.168.20.202,eth2,192.168.10.202,eth1,,,,,supervisor,supervisor,Cisco UCS C220
M5,site1,4,Intel Xeon Platinum 8180 (2.50GHz/28
cores),96,2,40,1,500,0,0,0,0,,0,0,0,0
```

#### NFS Platform Description File

```
ring,,,,,,,,,,,,,,,,,,,,,

sizing_version,customer_name,#ring,data_ring_name,meta_ring_name,HALO API key,S3
endpoint,cos,arc-data,arc-coding,,,,,,,,,,,,,

20.4,Sample,2,DATA,META,0,,3,7,5,,,,,,,,,,,,,

,,,,,,,,,,,,,,,,,,,,,

servers,,,,,,,,,,,,,,,,,,,,,

data_ip,data_iface,mgmt_ip,mgmt_iface,s3_ip,s3_iface,svsd_ip,svsd_iface,ring_members
hip,role,minion_id,enclosure,site,#cpu,cpu,ram,#nic,nic_size,#os_disk,os_disk_size,#
data_disk,data_disk_size,#raid_card,raid_cache,raid_card_type,#ssd,ssd_size,#ssd_for
_s3,ssd_for_s3_size
```

```
192.168.20.203,eth2,192.168.10.203,eth1,,,,,"DATA,META","storage,nfs,zookeeper,elastic",storage01,Cisco UCS C240 M5 - 3N,sitel,2,Intel Xeon Bronze 3104 (1.7 GHz/6 cores),256,2,40,2,960,12,10000,1,2,Cisco 12G Modular Raid Controller with 2GB cache (max 16 drives),2,960,0,0
```

```
192.168.20.204,eth2,192.168.10.204,eth1,,,,,"DATA,META","storage,nfs,zookeeper,elastic",storage02,Cisco UCS C240 M5 - 3N,sitel,2,Intel Xeon Bronze 3104 (1.7 GHz/6 cores),256,2,40,2,960,12,10000,1,2,Cisco 12G Modular Raid Controller with 2GB cache (max 16 drives),2,960,0,0
```

```
192.168.20.205,eth2,192.168.10.205,eth1,,,,,"DATA,META","storage,nfs,zookeeper,elastic",storage03,Cisco UCS C240 M5 - 3N,sitel,2,Intel Xeon Bronze 3104 (1.7 GHz/6 cores),256,2,40,2,960,12,10000,1,2,Cisco 12G Modular Raid Controller with 2GB cache (max 16 drives),2,960,0,0
```

```
192.168.20.202,eth2,192.168.10.202,eth1,,,,,supervisor,supervisor,Cisco UCS C220 M5,sitel,4,Intel Xeon Platinum 8180 (2.50GHz/28 cores),96,2,40,1,500,0,0,0,0,,0,0,0,0
```



## Summary

---

Object storage is an increasingly popular form of distributing data in a scale-out system. The entry size sinks to more and more smaller units. Scality with Scality RING is leading the pack with their technology when it comes to storing data as an object or file with high availability and reliability on small entry-level solutions.

The entry-level solution in this design guide provides customers and partners with everything necessary to store object or file data easily and securely. Cisco's leading technology of centralized management and advanced networking technology helps to easily deploy, manage and operate the Scality RING solution.

The Cisco and Scality solution provides you with new solutions to reliably implement object and file storage.

## About the Authors

---

**Oliver Walsdorf, Technical Marketing Engineer for Software Defined Storage, Computer Systems Product Group, Cisco Systems, Inc.**

Oliver has more than 20 years of storage experience, working in different roles at different storage vendors, and is now an expert for software-defined storage at Cisco. For the past four years Oliver was focused on developing storage solutions at Cisco. He now works on Scality RING, develops Co-Solutions with Scality for the overall storage market and published several Cisco documents. With his focus on SDS he drives the overall attention in the market for new technologies. In his leisure time, Oliver enjoys hiking with his dog and motorcycling.

**William Kettler, Customer Solution Engineer, Scality**

William Kettler is a Customer Solution Engineer Partner within Scality's Technical Services group. His current role includes helping customers deploy their petabyte-scale storage solutions, certifying strategic ISVs, and being a technical resource for Scality partners like Cisco.

## Acknowledgements

For their support and contribution to the design, validation, and creation of this Cisco Validated Design, we would like to acknowledge the following for their significant contribution and expertise that resulted in developing this document:

- Chris O'Brien, Cisco Systems, Inc.
- Jawwad Memon, Cisco Systems, Inc.
- Maziar Tamadon, Scality