

Cisco UCS S3260 Storage Server with SwiftStack Software Defined Storage Design Guide

Last Updated: October 30, 2018



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, see:

<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)

© 2018 Cisco Systems, Inc. All rights reserved.

Table of Contents

Executive Summary	5
Solution Overview	6
Introduction	6
Audience	6
Purpose of this Document	7
Solution Summary	7
Technology Overview	8
Cisco Unified Computing System	8
Cisco UCS Manager	8
Cisco UCS 6300 Fabric Interconnects	9
Cisco UCS 9332 Nexus Switches	10
Cisco UCS S3260 M5 Storage Server	11
Cisco UCS C220 M5 Rack-Mount Server	12
Cisco UCS Virtual Interface Card 1387	13
Red Hat Enterprise Linux 7.5	14
SwiftStack Software Defined Storage	15
Object Storage and SwiftStack	15
SwiftStack Design	17
SwiftStack System Architecture	17
SwiftStack Data Placement	18
SwiftStack Storage Policies	19
SwiftStack New Features	20
Cisco and SwiftStack	21
SwiftStack Use Cases	22
Media and Entertainment	22
Backup and Archives	23
Life Sciences and Genome Sequencing	24
Hybrid Clouds	25
Solution Design	26
Design Criteria	26
Requirements	26
Physical Infrastructure Considerations	26
Single Node versus Dual Node UCS S3260	26
Replication versus Erasure Coding	26

Flash Storage	27
JBOD Disks	27
Memory Sizing.....	27
Network Considerations	27
Uplinks	28
Multi-Region Deployments	28
Scaling out the Cluster.....	29
Deployment Architecture	29
Single Region Deployment.....	30
System Hardware and Software Specifications.....	31
Multi Region Deployment.....	34
Hybrid Cloud Deployment.....	35
Deployment Guide	37
Summary	38
About the Authors.....	39
Acknowledgements	39



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 of SwiftStack 6.x on Red Hat Enterprise Linux and on latest generation of Cisco UCS S3260 M5 Servers. This validated design provides the framework of designing and deploying SwiftStack SDS software on Cisco UCS S3260 storage servers. The Cisco Unified Computing System provides the storage, network, and storage access components for the SwiftStack, deployed as a single cohesive system.

This Cisco Validated design describes how the Cisco Unified Computing System can be used in conjunction with SwiftStack 6.x. With the continuous evolution of SDS there has been increased demand to have Object Storage validated on Cisco UCS servers. The Cisco UCS S3260 Storage Server, originally designed for the data center, together with SwiftStack SDS is optimized for object storage solutions, making it an excellent fit for unstructured data workloads such as backup, archive, and cloud data. The S3260 delivers a complete infrastructure with exceptional scalability for computing and storage resources together with 40 Gigabit Ethernet networking.

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

The reference architecture described in this document is a realistic use case for designing SwiftStack object storage on Cisco UCS S3260 Storage Server and Cisco UCS C220 Rack-Mounted Server.

Solution Overview

Introduction

Unstructured data in enterprises is growing at astronomical rates. This is being driven by the digital transformation and proliferation of rich-media across all industries and verticals including IOT, Online gaming, Media and Entertainment, big data analytics and mobile to name a few. According to IDC, not only is 80 percent of all new data generated will be unstructured data, but also the rate of growth is increasing

Traditional file servers and NAS work well for smaller data sets, but they are unable to scale-out for these large data sets that are common in enterprises today. At a certain point, you run into the limitations that are inherent to file systems and the performance degrades to point where it becomes unusable. These systems are suitable for supporting millions of files, and a few petabytes of data – not the billions of files, and **hundreds of petabytes of data that are the requirements of today's enterprises.**

To overcome these issues, many Enterprises are looking to the public cloud. However, storing petabytes of data that is actively being used by on-premises compute becomes a very expensive proposition – the storage fee, the per-transaction fee and the network usage fee all begin to add up.

The solution is an on-premises Object based scale out storage system that can support hundreds of petabytes in a single name space without any performance degradation or other issues found in traditional NAS systems. In addition, the ability to mirror data or to lifecycle data to the public cloud is extremely useful for data protection, archival of old data, cloud bursting for compute and many other use cases.

SwiftStack multi-cloud data management software on Cisco UCS is fully featured on-premise Object storage system that supports the Swift and S3 API, along with SMB and NFS for file access. It is designed for cloud-scale deployments and has the ability to sync and lifecycle data to the public cloud based on policy.

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

This document describes the architecture and design considerations of SwiftStack object storage on Cisco UCS S3260 servers with Cisco UCS C220 M5 rack servers.

Audience

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

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

Purpose of this Document

This document describes the steps required to design SwiftStack 6.x on Cisco UCS platform. It discusses design choices and best practices using this shared infrastructure platform.

Solution Summary

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

- Cisco Unified Computing System (Cisco UCS)
- Cisco UCS 6332 Series Fabric Interconnects
- Cisco UCS S3260 M5 storage servers.
- Cisco S3260 system IO controller with VIC 1380
- Cisco C220M5 servers with VIC 1387
- Cisco Nexus C9332PQ Series Switches
- SwiftStack Storage 6.x or later
- Red Hat Enterprise Linux 7.5

Technology Overview

Cisco Unified Computing System

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

The main components of Cisco Unified Computing System are:

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

The Cisco Unified Computing System is designed to deliver:

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

Cisco UCS Manager

Cisco UCS® Manager provides unified, embedded management of all software and hardware components of Cisco Unified **Computing System™** across multiple chassis, rack servers and thousands of virtual machines. It supports all Cisco UCS product models, including Cisco UCS B-Series Blade Servers, Cisco UCS C-Series Rack-Mount Servers, and Cisco UCS Mini, as well as the associated storage resources and networks. Cisco UCS Manager is embedded on a pair of Cisco UCS 6300 or 6400 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.



This design uses the Cisco UCS 6332 Fabric Interconnect for unified embedded management of the computing system.

Figure 1 Cisco UCS Manager



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

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

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

Cisco UCS 6300 Fabric Interconnects

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

Figure 2 Cisco UCS 6300 Fabric Interconnect

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

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

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

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

Cisco UCS 9332 Nexus Switches

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

Figure 3 Cisco Nexus 9332 Switch

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

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

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

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

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

Cisco UCS S3260 M5 Storage Server

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

Figure 4 Cisco UCS S3260 Storage Server



Extending the capability of the Cisco UCS C3000 portfolio, the Cisco UCS S3260 helps you achieve the highest levels of data availability. With dual-node capability that is based on the Intel® Xeon® scalable processors, it features up to 720 TB of local storage in a compact 4-rack-unit (4RU) form factor. All hard-disk drives can be asymmetrically split between the dual-nodes and are individually hot-swappable. The drives can be built-in in an enterprise-class Redundant Array of Independent Disks (RAID) redundancy or be in a pass-through mode.

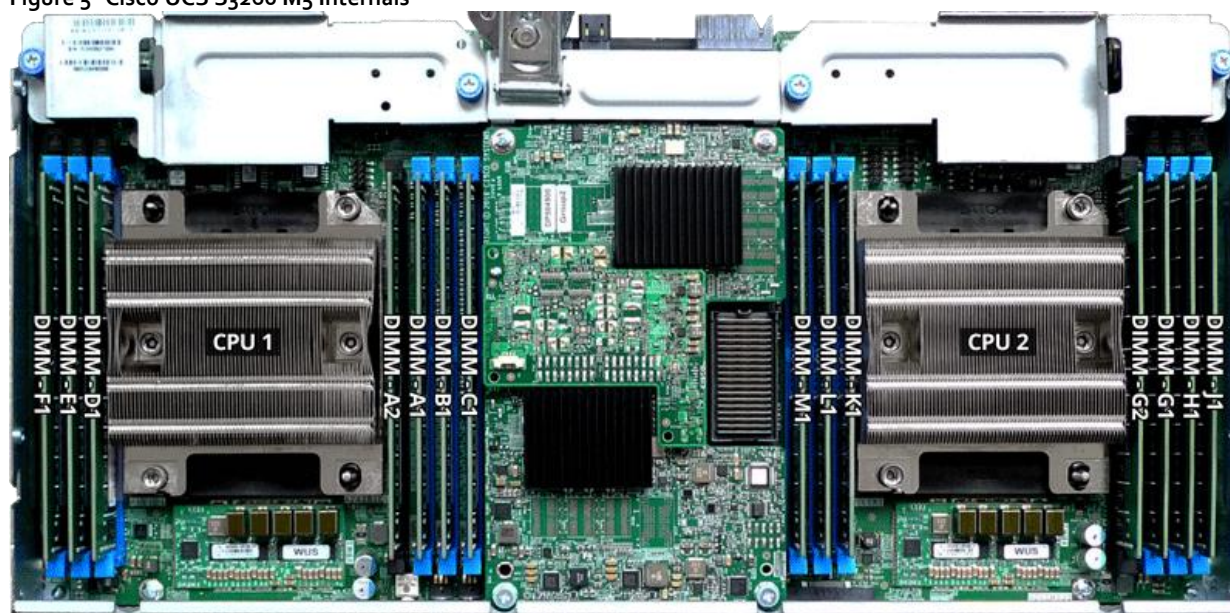
This high-density rack server comfortably fits in a standard 32-inch depth rack, such as the Cisco® R42610 Rack.

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

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

- Dual server nodes
- Up to 44 computing cores per server node
- Up to 60 drives mixing a large form factor (LFF) with up to 28 solid-state disk (SSD) drives plus 2 SSD SATA boot drives per server node
- Up to 1.5 TB of memory per server node (3 TB Total) with 128GB DIMMs
- Support for 12-Gbps serial-attached SCSI (SAS) drives
- A system I/O Controller either with HBA pass-through or RAID controller, with DUAL LSI 3316 Chip
- Cisco VIC 1300 Series Embedded Chip supporting Dual-port 40Gbps
- High reliability, availability, and serviceability (RAS) features with tool-free server nodes, system I/O controller, easy-to-use latching lid, and hot-swappable and hot-pluggable components
- Dual 7mm NVMe - Capacity points: 512G, 1TB and 2TB
- 1G Host Management Port

Figure 5 Cisco UCS S3260 M5 Internals



Cisco UCS C220 M5 Rack-Mount Server

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

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

Figure 6 Cisco UCS C220 M5 Rack-Mount Server



The Cisco UCS C220 M5 SFF server extends the capabilities of the Cisco Unified Computing System portfolio in a 1U form factor with the addition of the Intel® Xeon® Processor Scalable Family, 24 DIMM slots for 2666MHz DIMMs and capacity points up to 128GB, two 2 PCI Express (PCIe) 3.0 slots, and up to 10 SAS/SATA hard disk drives (HDDs) or solid state drives (SSDs). The Cisco UCS C220 M5 SFF server also includes one dedicated internal slot for a 12G SAS storage controller card.

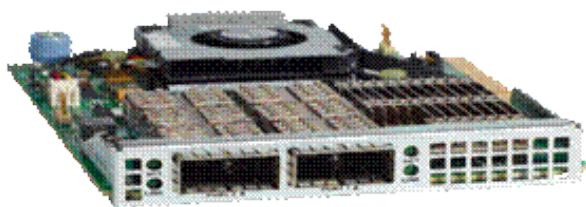
The Cisco UCS C220 M5 server included one dedicated internal modular LAN on motherboard (mLOM) slot for installation of a Cisco Virtual Interface Card (VIC) or third-party network interface card (NIC), without consuming a PCI slot, in addition to 2 x 10Gbase-T Intel x550 embedded (on the motherboard)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

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

Cisco UCS Virtual Interface Card 1387

The Cisco UCS Virtual Interface Card (VIC) 1387 is a Cisco® innovation. It provides a policy-based, stateless, agile server infrastructure for your data center. This dual-port Enhanced Quad Small Form-Factor Pluggable (QSFP) half-height PCI Express (PCIe) modular LAN-on-motherboard (mLOM) adapter is designed exclusively for Cisco UCS C-Series and 3260 Rack Servers. The card supports 40 Gigabit Ethernet and **Fibre Channel over Ethernet (FCoE)**. It incorporates **Cisco's next-generation** converged network adapter (CNA) technology and offers a comprehensive feature set, providing investment protection for future feature software releases. The card can present more than 256 PCIe standards-compliant interfaces to the host, and these can be dynamically configured as either network interface cards (NICs) or host bus adapters (HBAs). In addition, the VIC supports Cisco Data Center Virtual Machine Fabric Extender (VM-FEX) technology. This technology extends the Cisco UCS Fabric Interconnect ports to virtual machines, simplifying server virtualization deployment.

Figure 7 Cisco UCS VIC 1387



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

- Stateless and agile platform: The personality of the card is determined dynamically at boot time using the service profile associated with the server. The number, type (NIC or HBA), identity (MAC address and World Wide Name [WWN]), failover policy, bandwidth, and quality-of-service (QoS) policies of the PCIe interfaces are all determined using the service profile. The capability to define, create, and use interfaces on demand provides a stateless and agile server infrastructure.
- Network interface virtualization: Each PCIe interface created on the VIC is associated with an interface on the Cisco UCS fabric interconnect, providing complete network separation for each virtual cable between a PCIe device on the VIC and the interface on the fabric interconnect.

Red Hat Enterprise Linux 7.5

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

Red Hat Enterprise Linux:

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

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

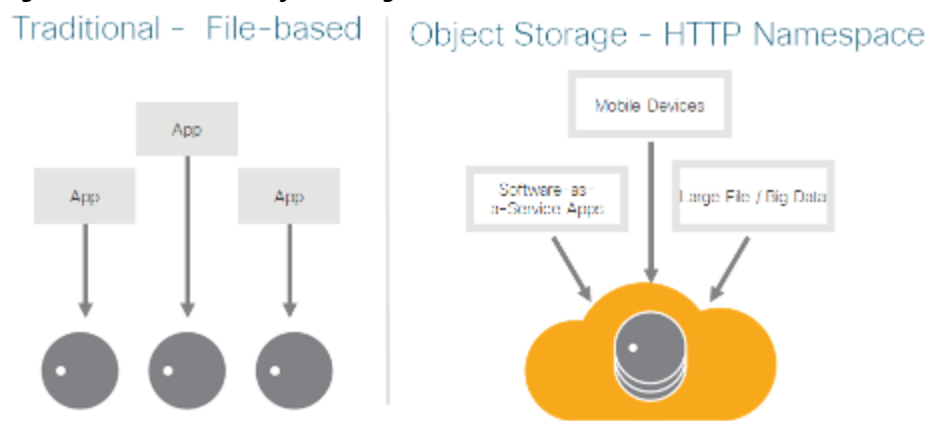
SwiftStack Software Defined Storage

The storage market has shifted dramatically in the last few years from one that is dominated by proprietary storage appliances. The data center has evolved from providing mainly back-office transactional services, to providing a much wider range of applications including cloud computing, content serving, distributed computing and archiving.

Object Storage and SwiftStack

Object Storage architecture manages data as objects as opposed to file systems that manage data as file hierarchy, and block storage which manages data as blocks within sectors and tracks. Figure 8 below depicts the differences.

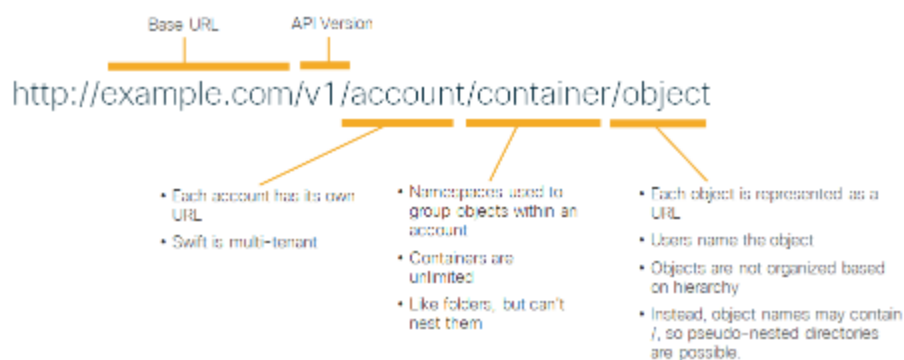
Figure 8 Traditional vs. Object Storage



Every object in a Swift container is accessed via http URL (Figure 9).

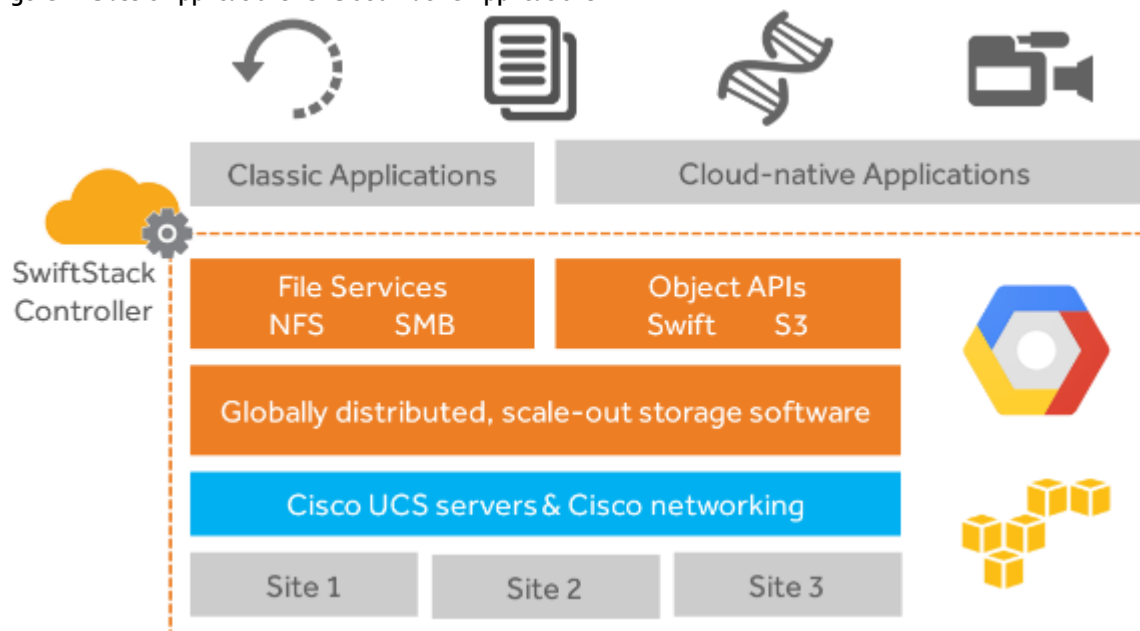
Figure 9 Objects have URL

Every Object Has an URL



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

Figure 10 Classic Applications vs. Cloud-native Applications



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

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

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

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

Key SwiftStack features:

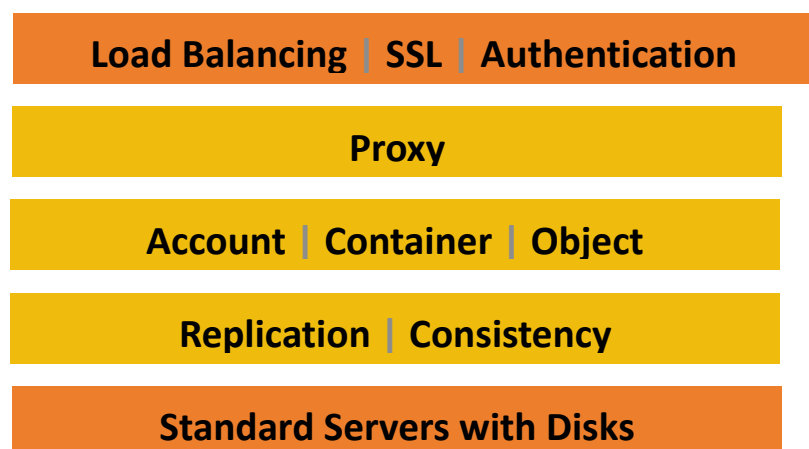
- Starts as small as 150TB, and scales to 100s of PB
- Spans multiple data centers while still presenting a single namespace
- Handles data according to defined policies that align to the needs of different applications
- Uses erasure coding and replicas in the same cluster to protect data
- Offers multi-tenant support with authentication via Active Directory, LDAP, and Keystone
- Supports file protocols (SMB, NFS) and object APIs (S3, Swift) simultaneously
- Automatically synchronizes to Google Cloud Storage and Amazon S3 with the Cloud Sync feature

- Encrypts data and metadata at rest
- Manages highly scalable storage infrastructure via centralized out-of-band controller
- Ensures all functionality touching data is open by leveraging an open-source core
- Optimizes TCO with pay-as-you-grow licensing with support and maintenance included

SwiftStack Design

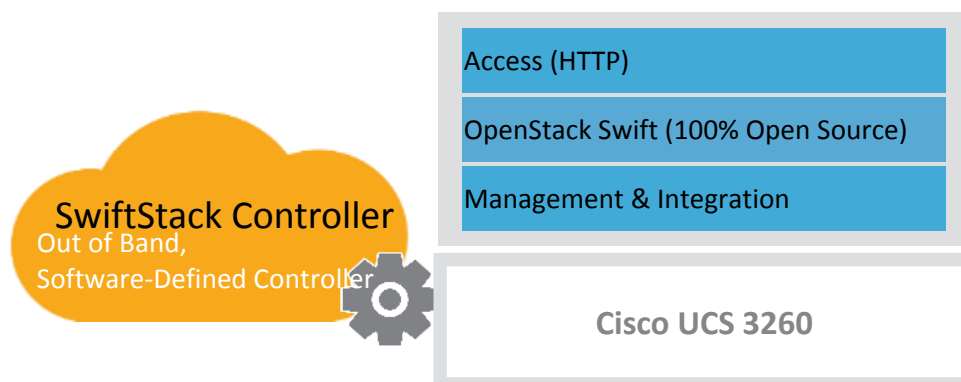
SwiftStack provides native Object API (S3 and Swift) to access the data stored in the SwiftStack Cluster. The design provides linear scalability, extreme durability with no single-point of Failure. It uses any standard Linux system. SwiftStack clusters also support multi-region data center architecture.

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



SwiftStack System Architecture

SwiftStack solution is enterprise-grade object storage, with OpenStack Swift at the core. It has been deployed at 100s of companies with massive amounts of data stored. It includes two major components; SwiftStack Storage Nodes and the SwiftStack Controller which is an out-of-band management system that manages one or more SwiftStack storage clusters.



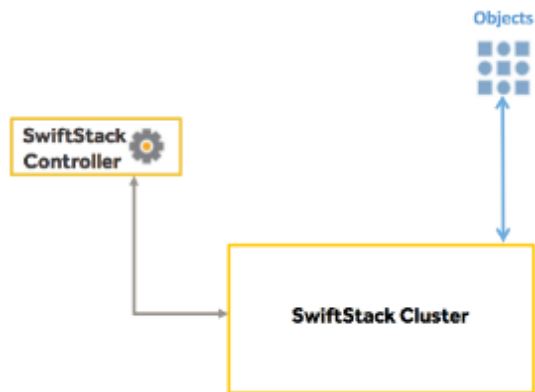
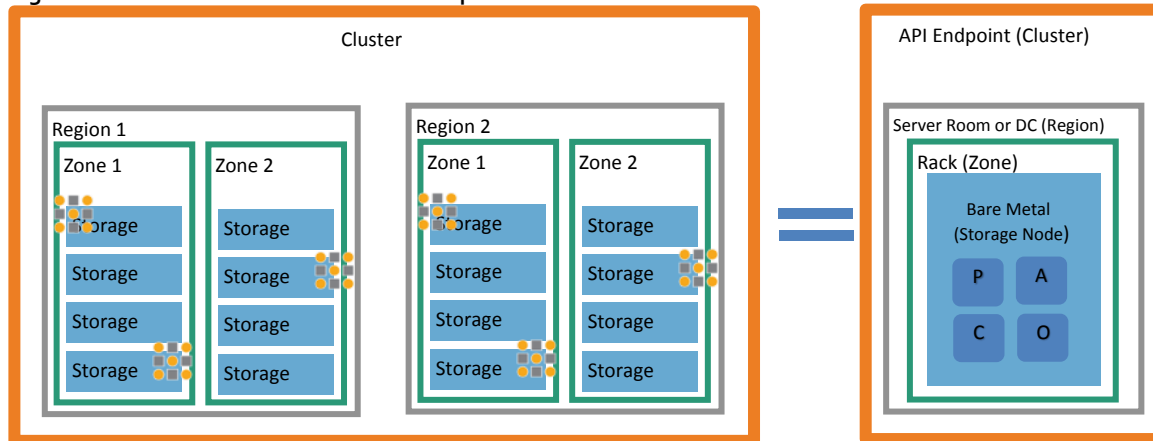


Figure 11 SwiftStack Cluster Abstract Concept



SwiftStack Data Placement

Disks are "as-unique-as-possible"

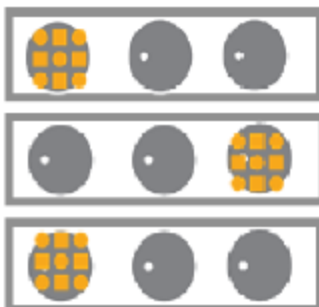


Storage Racks are "as-unique-as-possible"



Small Cluster

Storage Nodes are "as-unique-as-possible"



Multi-Region

Distributed data centers are "as-unique-as-possible"



Swift has a “unique-as-possible” placement algorithm which ensures that the data is placed efficiently and with as much protection from hardware failure as possible.

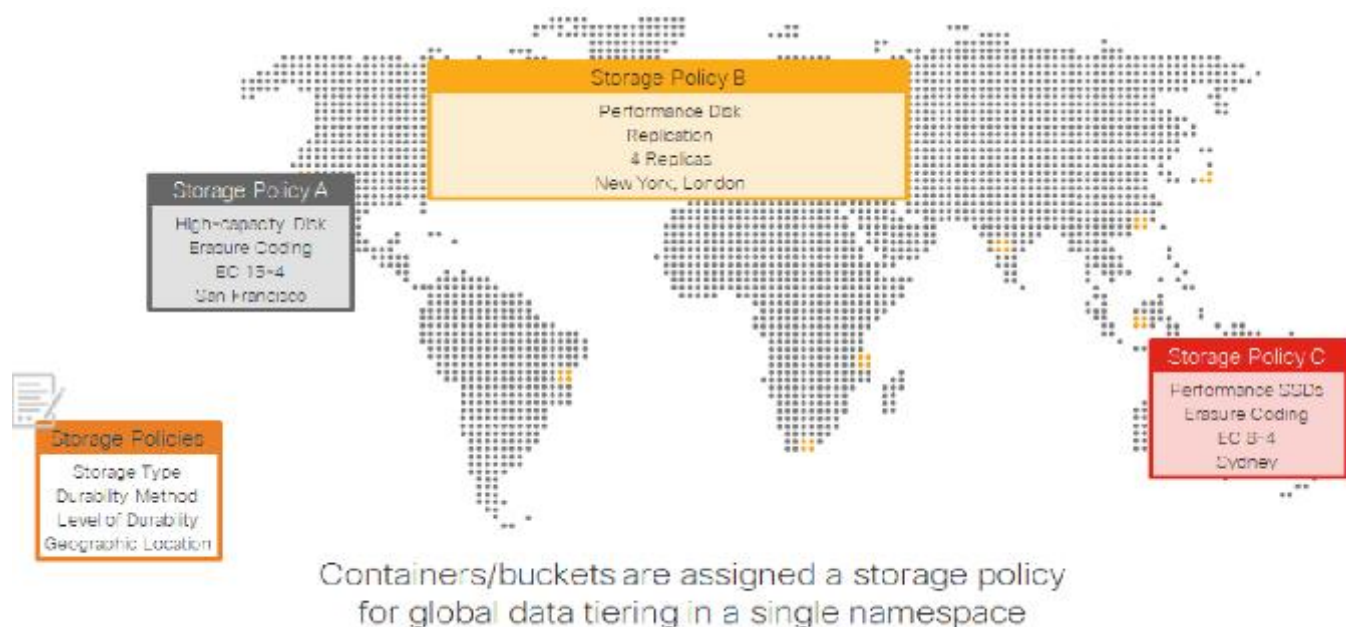
Swift’s unique-as-possible placement works like this: data is placed into tiers—first the availability zone, next the server, and finally the storage volume itself. Replicas of the data are placed so that each replica has as much separation as the deployment allows. When Swift chooses how to place each replica, it first will **choose an availability zone that hasn’t been used. If all availability zones have been chosen, the data will be** placed on a unique server in the least used availability zone. Finally, if all servers in all availability zones have been used, then Swift will place replicas on unique drives on the servers.

The unique-as-possible placement in Swift gives deployers the flexibility to organize their infrastructure as they choose. Swift can be configured to take advantage of what has been deployed, without requiring that the deployer conform the hardware to the application running on that hardware.

SwiftStack Storage Policies

SwiftStack storage policies allow deployers to specifically configure their Swift Cluster to support different needs of data stored in the cluster. Storage Policies organize data in Swift Cluster base on the following:

- Location
- Storage media
- Number of replicas
- Erasure Codes



Rather than deploying, managing, upgrading and administering multiple storage systems, a single storage environment can encompass multiple use cases. This dramatically simplifies storage management and makes life much easier for users.

Also, storage policies integrate into SwiftStack. This means that the utilization data will reflect usage for each tier, allowing you to appropriately charge-back / show-back storage users. The capacity planning page

shows how much capacity is left for each storage policy. And the SwiftStack Controller makes it easy to see which storage nodes are in which storage policy.

SwiftStack New Features

- SwiftStack File Access: Enables bi-modal access to your data stored in a SwiftStack Cluster via the S3 API, Swift API as well as SMB or NFS filesystem protocols. Users can configure up to 1,000 hourly, daily and monthly snapshots per volume for data protection.
- 1space multi cloud data management that allows users to sync or archive data to a supported public cloud or another SwiftStack Clusters. Profiles can use time and / or metadata to select which data gets synched or archived. 1space profiles can optionally merge namespaces, as well as optionally not propagate DELETES for data protection.
- 1space live migrations from S3 and Swift sources to SwiftStack clusters enabling customers to bring back their data on-premises without any interruption.
- SwiftStack containers can now support practically unlimited number of objects per container
- Ability to select the allowed cipher suite string and allowed TLS versions. In addition, operators can disable less secure TLSv1 protocols for controller HTTPS communications.
- SwiftStack Auth supports Keystone V3 API
- Metadata sync allows for specifying an optional Elasticsearch pipeline when indexing new documents
- When indexing Swift objects whose metadata values contain JSON, Metadata Sync can optionally parse the JSON value and submit it as part of the index operation (as opposed to submitting a string).
- Support for SELinux in enforcing more
- Integration with external KMS using KMIP

Cisco and SwiftStack

Cisco and SwiftStack have developed a solution that meets the challenges of scale-out storage. This solution uses SwiftStack Object Storage software with Cisco UCS® S-Series Storage Servers powered by Intel® Xeon® processors. SwiftStack unified namespace help ensure universal access to your data regardless of where your data is physically stored: on your premises, in the public cloud, or in a combination of both environments (hybrid cloud). Common management across the solution delivers fast day-0 deployments and continuing operations. The building-block architecture of this solution makes scaling out fast and easy. You can make changes when your applications demand, without disruption. The high-bandwidth networking helps ensure fast object retrieval and data transfer.

Cisco and SwiftStack deliver universal access regardless of where data is physically located. Now you can move your data close to where it is being processed. As your data grows, this solution easily grows with it. Your data is stored in a single shared name space, enabling you to place data close to your applications, regardless of location. Because of the shared name space between clouds, you can easily move your data across a hybrid cloud.

The solution engineered by both Cisco and SwiftStack deliver the following:

1. Flexibility
 - a. Universal access to your data.
 - b. Move data between public and private clouds.
 - c. Scale from Terabytes to Petabytes in a cluster.
2. Performance
 - a. 160 Gbps of bandwidth.
 - b. Multithreaded access with Load Balancing.
3. Value
 - a. Simple day-0 deployment
 - b. Easy management
 - c. Better data durability
4. Reduced Risk with Solution support from Cisco TAC for the end-to-end solution

The solution provides high performance and capacity for data-intensive workloads. The Cisco UCS S3260 Storage Server has a dual-node architecture using Intel Xeon processors. It offers the right balance between computing power and capacity. At the same time system can be designed to use them either for performance or Capacity Optimized solutions.

Cisco UCS S3260 server uses two system I/O controllers. These controllers provide an aggregate of 160 Gbps of network connectivity.

The SwiftStack Controller provides a single-pane dashboard to manage entire SwiftStack Object Storage deployment whether the cluster is local to a data center or span across multiple regions geographically. The SwiftStack controller enables you to manage, deploy, scale, upgrade, and monitor the object storage system **through API's and browser based dashboard.**

SwiftStack Use Cases

There are several use cases and target industries where you can use Cisco UCS and SwiftStack solution. The use cases and industries are several but not limited to the following:

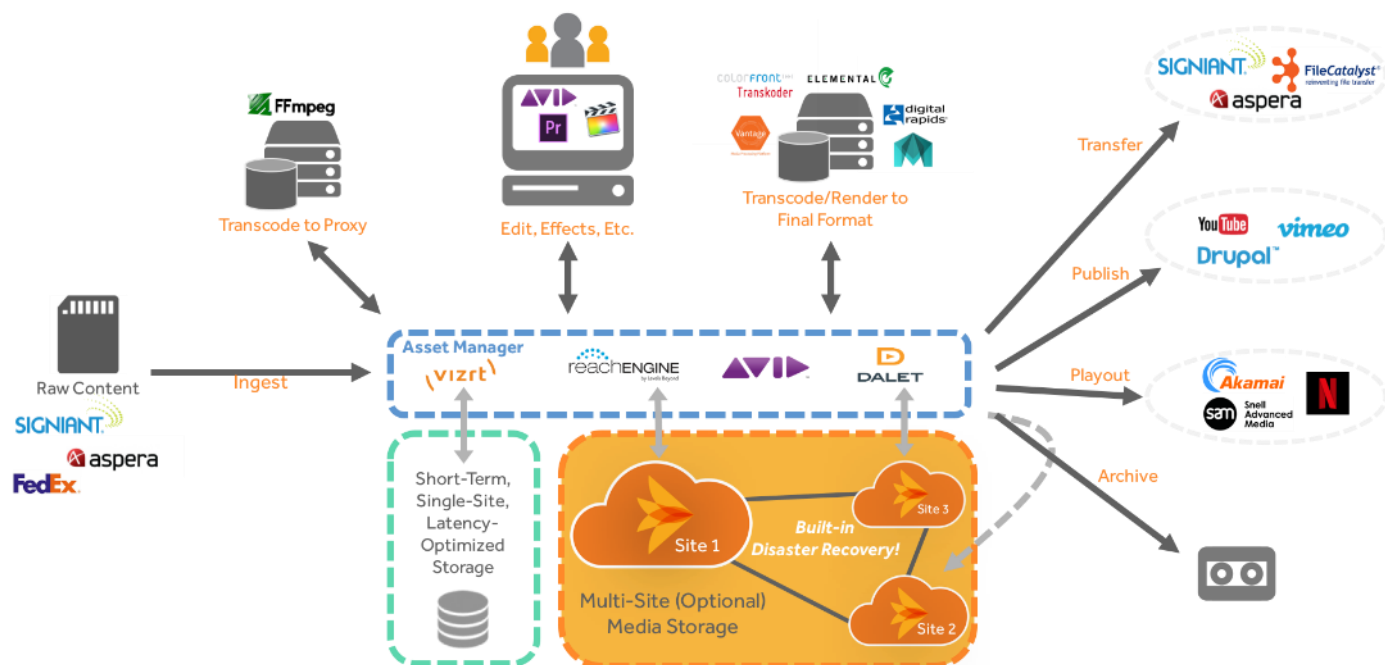
Serial No	Use Case
1	Media and Entertainment
2	Backup, Active Archive, Long Term Archive/Tape Replacement
3	Video and Content Distribution
4	Video Surveillance
5	Machine Learning
6	File systems
7	Life Sciences
8	Private and Hybrid Cloud Sync
9	Analytics



A few of the use cases are described below.

Media and Entertainment

SwiftStack is a globally-distributed media storage system that is a back end to your media asset manager, an origin for multiple CDNs, and a path to distributing your transcode and rendering operations—allowing you to start with a few terabytes and seamlessly scale to many petabytes. SwiftStack media storage can help optimize workflows from production to post-production to distribution.

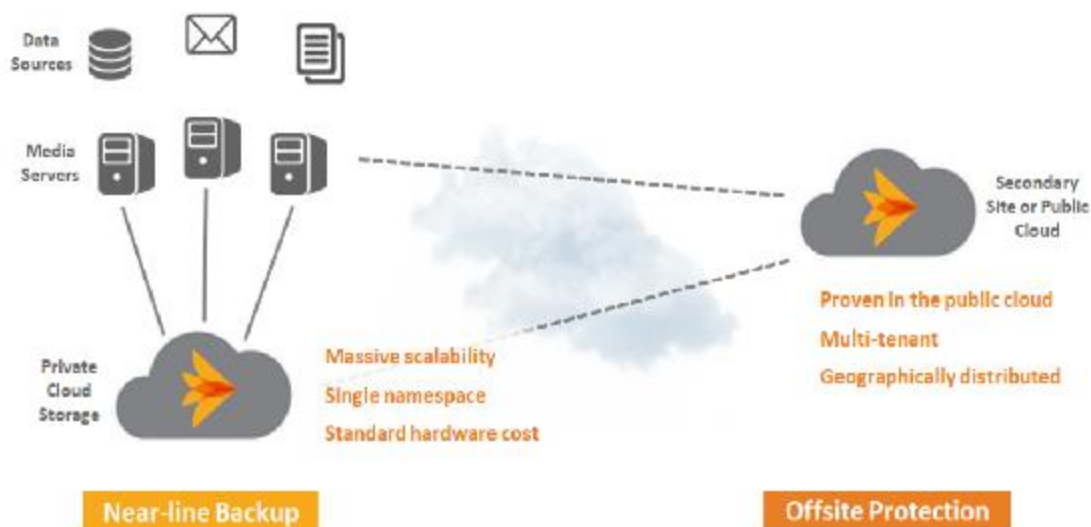


SwiftStack can be deployed in a single site or used to connect multiple sites together with a single view of the objects or assets stored.

Instead of using FTP or a WAN accelerator to send large files from point A to point B, with SwiftStack, any data uploaded at any site can be instantly accessed at any remote site. With SwiftStack, you can write anywhere and read everywhere.

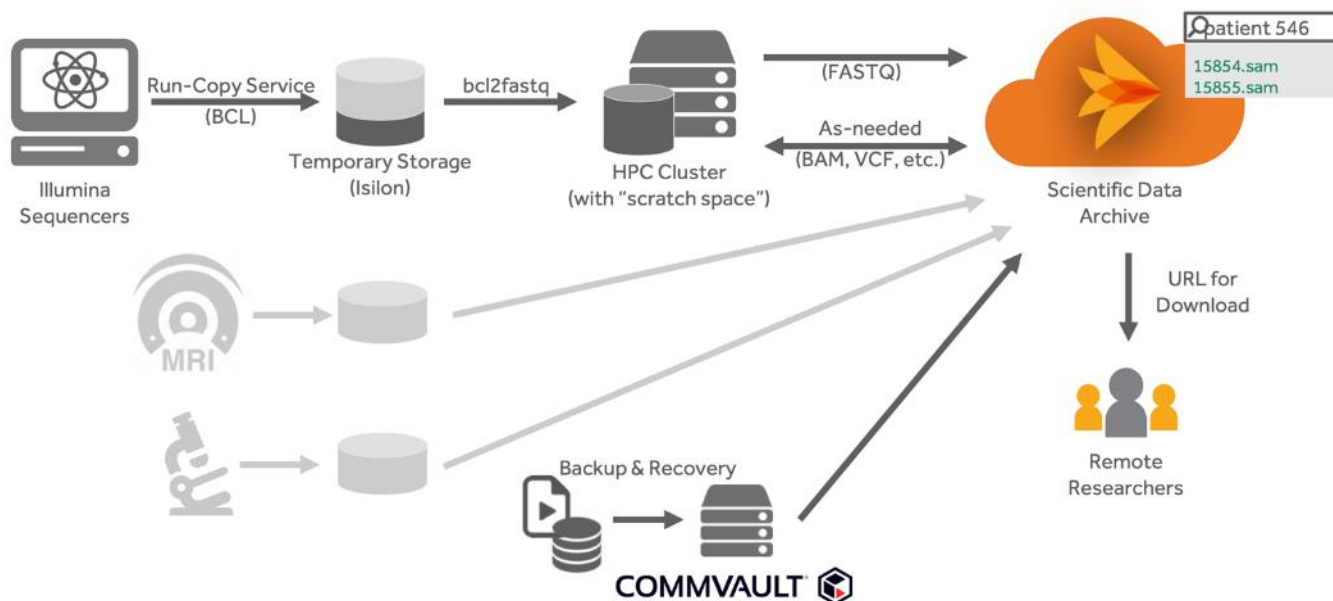
Backup and Archives

Traditional storage does not scale, so as you protect more data, it often requires more siloed storage systems, increasing management overhead. With SwiftStack, you can start small and scale to many petabytes.



A SwiftStack object storage cluster can have nodes that span both an onsite and multiple offsite locations, allowing your data to be automatically protected without requiring any day-to-day action or service to move tapes back and forth. It scales near infinitely, eliminating those storage silos, and the data is all accessible under a single namespace. This allows the media servers to utilize the same pool of storage capacity, no matter how many of them you have.

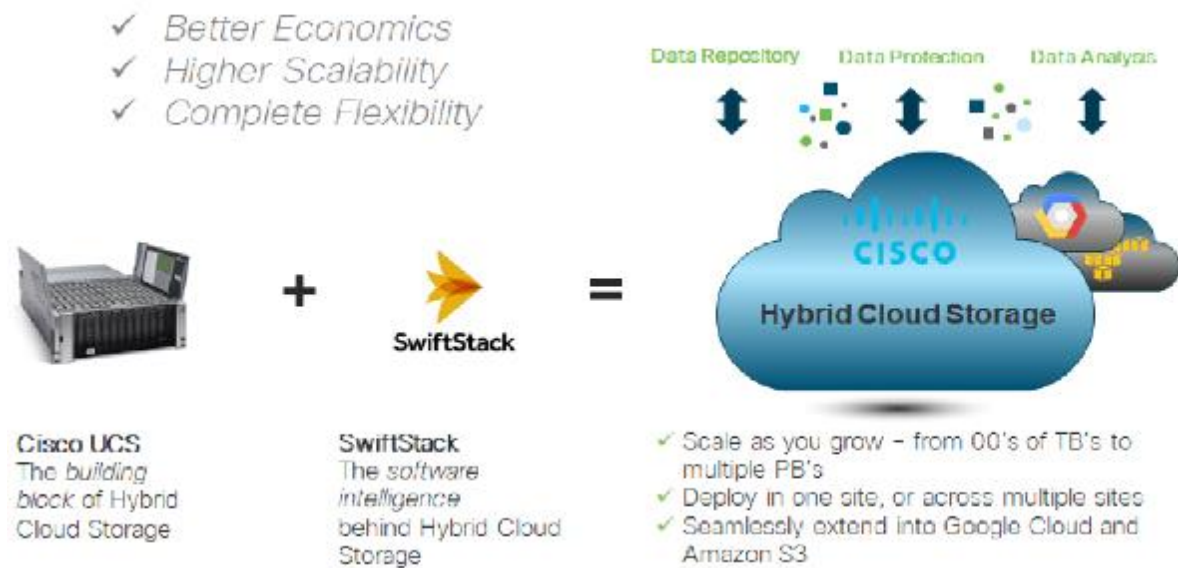
Life Sciences and Genome Sequencing



SwiftStack is a single place for your scientific research data at any scale to hydrate your compute farms and provide global access for collaboration, while driving out significant unnecessary costs.

One of the benefits of SwiftStack's cloud architecture is that it can scale from one to many geographic regions—all in the same name space, and according to the policies you choose, SwiftStack will automatically replicate data between regions. SwiftStack's has ability to control access rules for accounts and containers and to even provide temporary URLs for download access to individual objects. This way remote clients or researchers can pull data directly from the SwiftStack cluster.

Hybrid Clouds



SwiftStack innovations power multi-cloud data management for Enterprises, enabling freedom to move workloads between clouds with universal access to data across on-premises and public infrastructure. When your unstructured data is stored in a single namespace backed by cloud storage, more of your users and applications can access it over a longer period of time. This unlocks new opportunities for collaborating on data and simplifying infrastructure management as data grows. SwiftStack uniquely enables existing applications to store data that is directly usable by cloud-native applications, and vice versa. As a result, storage silos are eliminated, making for faster and more accurate insights from data.

Solution Design

Design Criteria

This section details a few points that may be considered for the design of the Infrastructure

Requirements

The requirements of the storage have to be understood for the design. These may include the total usable space, future expansion and organic growth for the capacity of the cluster, the performance of the cluster in terms of throughput and bandwidth, the average block size of IO, single site, multi-domain or Multi-site requirements and so on. These requirements will influence few of the sections mentioned below.

Physical Infrastructure Considerations

Single Node versus Dual Node UCS S3260

Cisco UCS S3260 is offered both with single node and dual node configurations for a full chassis in a 4RU rack space.

Cluster may be categorized into capacity based or performance based depending on the requirements. A dual server offers double the CPU and memory for the same set of disks. Hence where performance is more important, a dual node configuration is recommended.

However, where performance is not the key criteria and fewer cores per disk suffice in use cases such as backup or archives, a single node configuration is recommended. This will reduce the TCO of the solution too.

Replication versus Erasure Coding

SwiftStack offers both replication and erasure coding. The usable to raw space ratio is determined by the data protection chosen.

Replication defaults to making three copies of replicas of the object being stored. For every object being created, 2 copies of the same object are replicated within the cluster. This means the effective usable capacity of the cluster is reduced to 1/3rd of the raw disk space. Customers may choose to create storage policies that store more than 3 replicas, and should make sure to accurately calculate the required raw capacity based on their chosen replication policy

Erasure coding (EC) splits your data objects into fragments and adds the parity segments for data protection. Compared to Replication, you get a better usable to raw capacity ratio with Erasure code. SwiftStack uses Cauchy-Reed-Solomon erasure coding algorithm. By using 8+4 EC (minimum of 6 nodes recommended), the object is split into 8 data segments and 4 parity segments. The availability of any of the 8 out of 12 segments is sufficient to build the object back. This gives the usable space as 8/12 or 2/3rd or 66 percent of the raw disk space. On the contrary, replication could have been only 33 percent. SwiftStack also supports additional Erasure coding policies such as 4+3, and 15+4.

A recommended practice is to use replication for warmer objects. In addition, EC is not recommended for objects that will result in segment sizes of less than 1MB as the computational overhead is larger for these smaller objects. While the data object is read in full in replication, EC has to merge the dispersed fragments.

Apart from replication, SwiftStack supports the following erasure coded configurations:

- 4 + 3 (minimum of 5 nodes)
- 8 + 4 (minimum of 6 nodes)
- 15 + 4 (minimum of 10 nodes)
- Custom EC policies are also supported in consultation with the SwiftStack Professional Services team

Flash Storage

Flash Storage with SAS SSD's or NVMe's are recommended to store metadata for faster performance, especially when you have millions of objects per container. These are used for Account and Container Services only. The capacity requirement for Flash storage per node depend on the total usable capacity of the nodes in the cluster as well as the number of objects in the cluster.

As an example, a dual node configuration has 28 disk slots for each node. With an average object size of 2MB, 512GB NVMe is recommended per node. In case of Single Node configuration that is fully populated with 56 disks, and an average object size of 2MB, 1 TB NVMe is recommended. **Note that we can use SSD's as well, in place of NVMe's. However, SSD's will occupy 1 or 2 slots, and the effective raw disk space for the storage will be reduced.**

JBOD Disks

SwiftStack recommends using JBOD disks as Swift actively checks all disks in the cluster.

Memory Sizing

Memory sizing is based on the number of drives on each storage server. Standard designs call for 384GB for the S3260 M5 single node, and 192GB for the S3260 M5 dual node configurations.

Network Considerations

Network requirements for SwiftStack are standard Ethernet only. While the software can work on a single network interface, it is recommended to carve out different virtual interfaces in Cisco UCS and segregate them. Cisco UCS S3260 has two physical ports of 40G each and the VIC allows you to carve out many Virtual interfaces on each physical port.

The following networks are recommended for the smooth operation of the cluster.

Outward-Facing Network

This is the front facing network and is used for API access and to run the proxy and authentication services.

Cluster-Facing Network

This is internal network for communication between the proxy servers and the storage nodes. This is a private network.

Replication Network

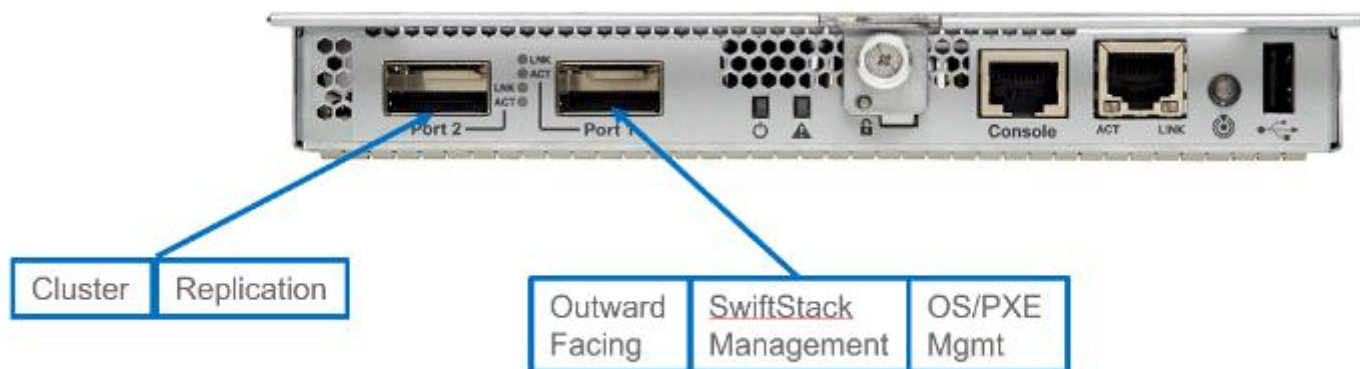
This Internal network is for communication between the proxy servers and the storage nodes. This is a private network too.

Management Network

All nodes must be able to route IP traffic to a SwiftStack controller. This is the management network for all services.

Hardware or PXE Management Network

This is optional network for Hardware management.



It is recommended to have Cluster network on one port and the outward facing and management networks on another port. This will provide 40Gb bandwidth for each of these networks. While the management network requirements are minimal, every PACO node can take up to 40Gb of client bandwidth requirements. **Also by having the client and cluster VIC's pinned to each fabric of the Fabric Interconnects there is a minimal overhead of network traffic passing through the upstream switches for inter node communication if any. This unique feature of Fabric Interconnects and VIC's makes the design highly flexible and scalable.**

Uplinks

The uplinks from Fabric Interconnects to upstream switches like Nexus carry the traffic in case of FI failures or reboots. A reboot for instance will be needed during a firmware upgrade. While there is a complete High Availability built in the infrastructure, the performance may drop depending on the uplink connectors from each FI to the Nexus vPC pool. In case you want "no" or a "minimal drop", in case of such failures, increase the uplink connectors as well.

Multi-Region Deployments

Nodes of a SwiftStack cluster can exist in multiple geographic regions. For each application, you choose how you want the data to be handled and where it should go, placing the data where applications and users **most need it. It's centrally managed and driven by policies you define.**

If one region becomes unavailable or an application has to be moved, data is automatically accessed from nodes in another region without needing to know the difference. SwiftStack also keeps track of client performance, so data is always accessed from the fastest node.

Unlike traditional storage, replicating data across multiple regions, or physical sites, is built into SwiftStack. **There's no need for third-party replication software or the requirement to manage independent storage in each region to further protect or location-optimize your data.** Also, geo-replicated containers/buckets exist in single namespace, so applications and users do not have to be aware of where the data physically lives.

One capability of a storage policy is to designate what regions participate in storing the data. This gives you the ability to not only help ensure your data is protected from a regional outage but optimizes the location of that data for fastest access by other applications and users in your workflow.

Storage policies define how a container/bucket of data will be stored and protected across the cluster, allowing unique data sets to be handled differently. SwiftStack uses both erasure coding and replicas to protect data. With replicas, the storage policies can specify how many replicas will reside in each region. SwiftStack takes a unique approach to multi-region erasure coding to deliver optimal performance and a single global namespace, while placing minimal demand on the WAN. Instead of spreading the data and parity bits across all available regions, SwiftStack uses the nodes in each region to protect the data, keeping the data whole in each data center. This allows for better performance as well as enables in-region rebuilds of the data if needed.

Scaling out the Cluster

It is not uncommon to Scale out the cluster because of organic growth or because of new business requirements.

Cisco UCS hardware along with SwiftStack SDS offer exceptional flexibility in order to scale out as your requirements change.

1. **Cisco UCS 6332 Fabric Interconnects have 32 ports each. Each server is connected to either of the FI's.** Leaving the uplinks and any other clients directly connected to the Fabrics, 24-28 server nodes can be connected to FI pairs. In case more servers needed you should plan for a multiple zones and regions.
2. Cisco UCS offers KVM management both in-band and out-of-band. In case out-of-band management is **planned, you may have to reserve as many free IP's as needed for the servers. This planning while designing the cluster makes expansion very straight forward.**
3. Cisco UCS provides IP pool management, MAC pool management along with policies that can be defined once for the cluster. Any future expansion for adding nodes etc., is just a matter of expanding the above pools.
4. Cisco UCS is a template and policy based infrastructure management tool. All the identity of the servers is stored through Service Profiles that are cloned from templates. When a template is created a new Service Profile for the additional server can be created and easily applied on the newly added hardware. Cisco UCS makes Infrastructure readiness, extremely simple, for any newly added storage nodes. Rack the nodes, connect the cables and then clone and apply the Service Profile is what that is needed.
5. When the nodes are ready, it is as simple as installing the SwiftStack node software on the nodes (as before), provisioning the nodes and adding them to the cluster via the SwiftStack Controller. You can use **SwiftStack's machine profile feature to automate the provisioning.**
6. SwiftStack will automatically re-distribute the data evenly across the cluster resources, including the newly added nodes/capacity.

The simplified management of the infrastructure with Cisco UCS and well tested node addition from SwiftStack makes the expansion of the cluster very simple.

Deployment Architecture

Cisco and SwiftStack have partnered to validate the following architectures.

- 6 chassis and 12 node deployment on SwiftStack, in a single region.
- 3 chassis and 6 nodes deployment depicting Multi-Site architecture in two different regions.
- Cloud Sync as a Hybrid cloud deployment.

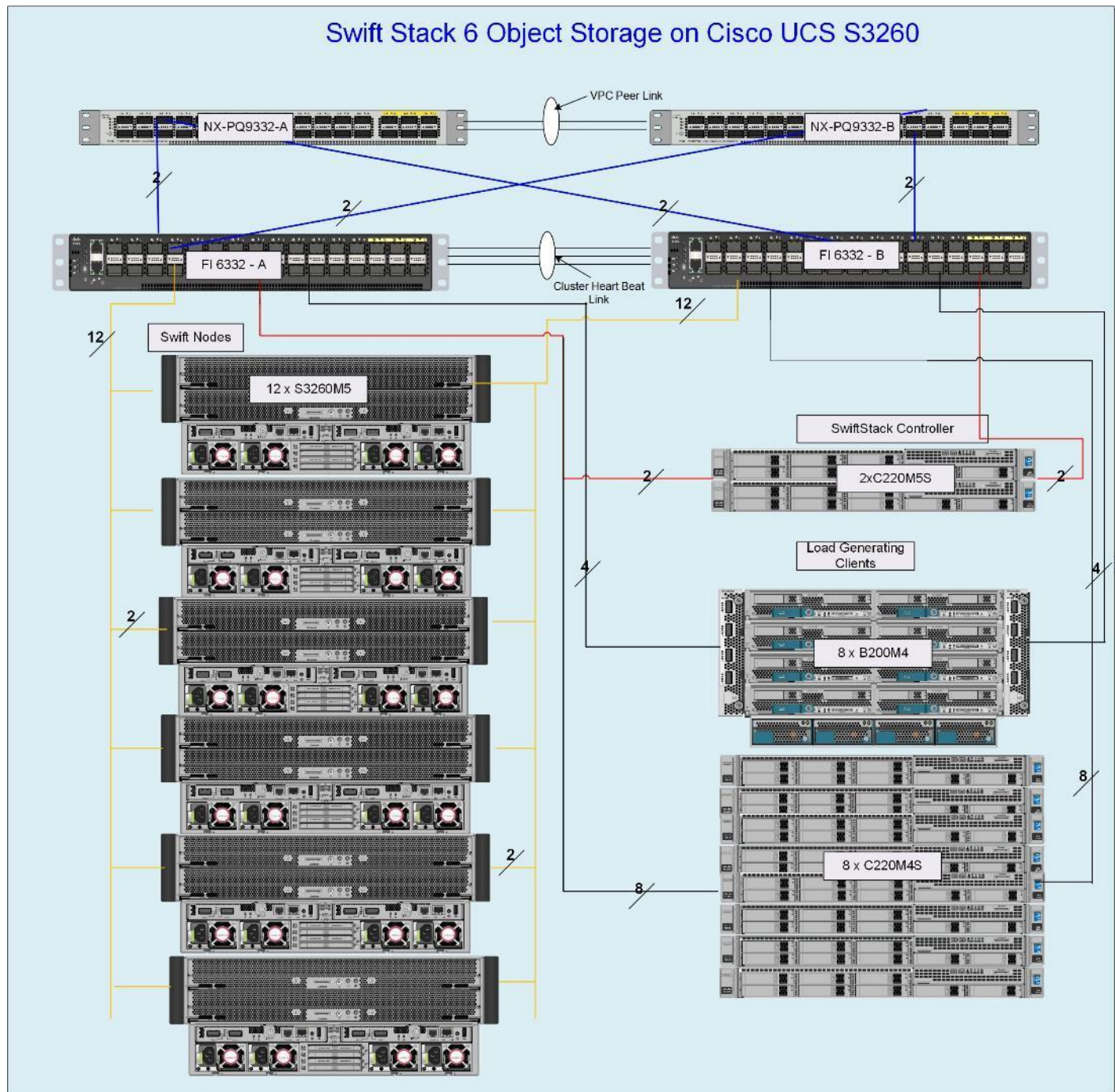
Single Region Deployment

The reference architecture provides a comprehensive, end-to-end example of designing and deploying SwiftStack object storage on Cisco UCS S3260 as shown in the below figure. The CVD describes the design and architecture of SwiftStack object storage software with six Cisco UCS S3260 Storage Server Chassis', each with two Cisco UCS S3260 M5 nodes configured as Storage servers and two Cisco UCS C220 M5S Rack server as Active/Standby Controllers. The whole solution is connected to a pair of Cisco UCS 6332 Fabric Interconnects and to a pair of upstream network switches Cisco Nexus 9332PQ.

The detailed configuration is as follows:

- 2 x Cisco Nexus 9332PQ Switches
- 2 x Cisco UCS 6332 Fabric Interconnects
- 6 x Cisco UCS S3260 Storage Servers with 2 x Cisco UCS S3260 M5 server nodes each
- 2 x Cisco UCS C220 M5S Rack-Mount Servers

Swift Stack 6 Object Storage on Cisco UCS S3260



The Cisco UCS C220 M4's and Cisco UCS B200 M4's are optional and are used as load generating clients only.

System Hardware and Software Specifications

Software Versions

Table 1 Software Versions

Layer	Component	Version or Release
Software	Red Hat Enterprise Linux Server	7.5 (x86_64)

Layer	Component	Version or Release
	SwiftStack	6.x and above
	Cisco UCSM	4.x and above

Hardware Requirement and Bill of Materials

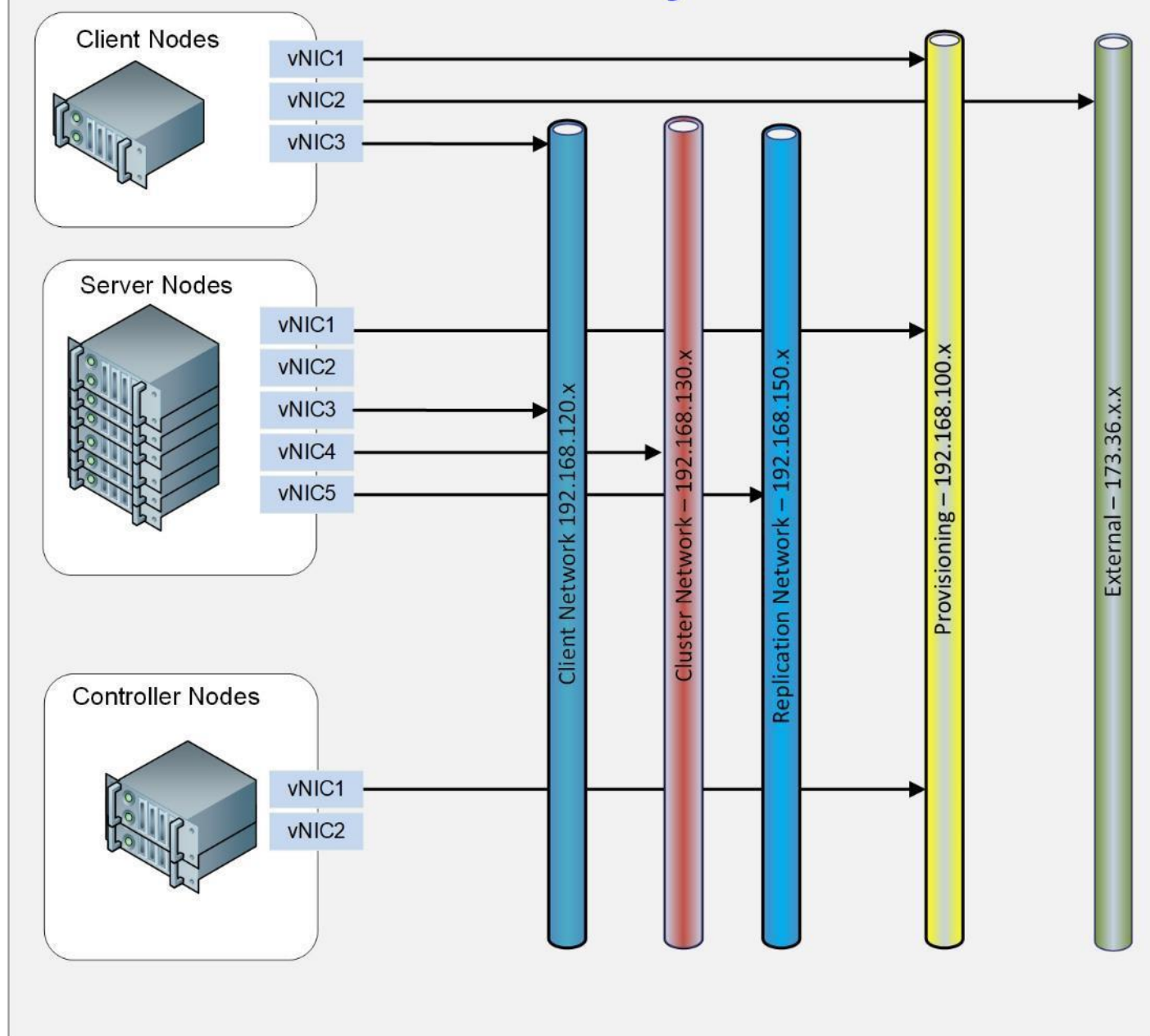
Table 2 Bill of Materials

Component	Model	Quantity	Comments
SwiftStack Storage Nodes	Cisco UCS S3260	12	<ol style="list-style-type: none"> 2 server nodes per chassis 2 sockets per node Cisco 12G RAID Controller per Node 2 OS boot disks 1 NVMe for metadata per node. 56 x 10 TB HDD's System IO Controller with VIC.
SwiftStack Controller Node	Cisco UCS C220 Rack server	2	<ol style="list-style-type: none"> 2 sockets per node Cisco 12G RAID Controller 2 x 1TB Boot disks for OS VIC Card
UCS Fabric Interconnects	Cisco UCS 6332 Fabric Interconnects	2	
Switches	Cisco Nexus 9332PQ Switches	2	

Network Topology

The following is the Network Topology used in the setup.

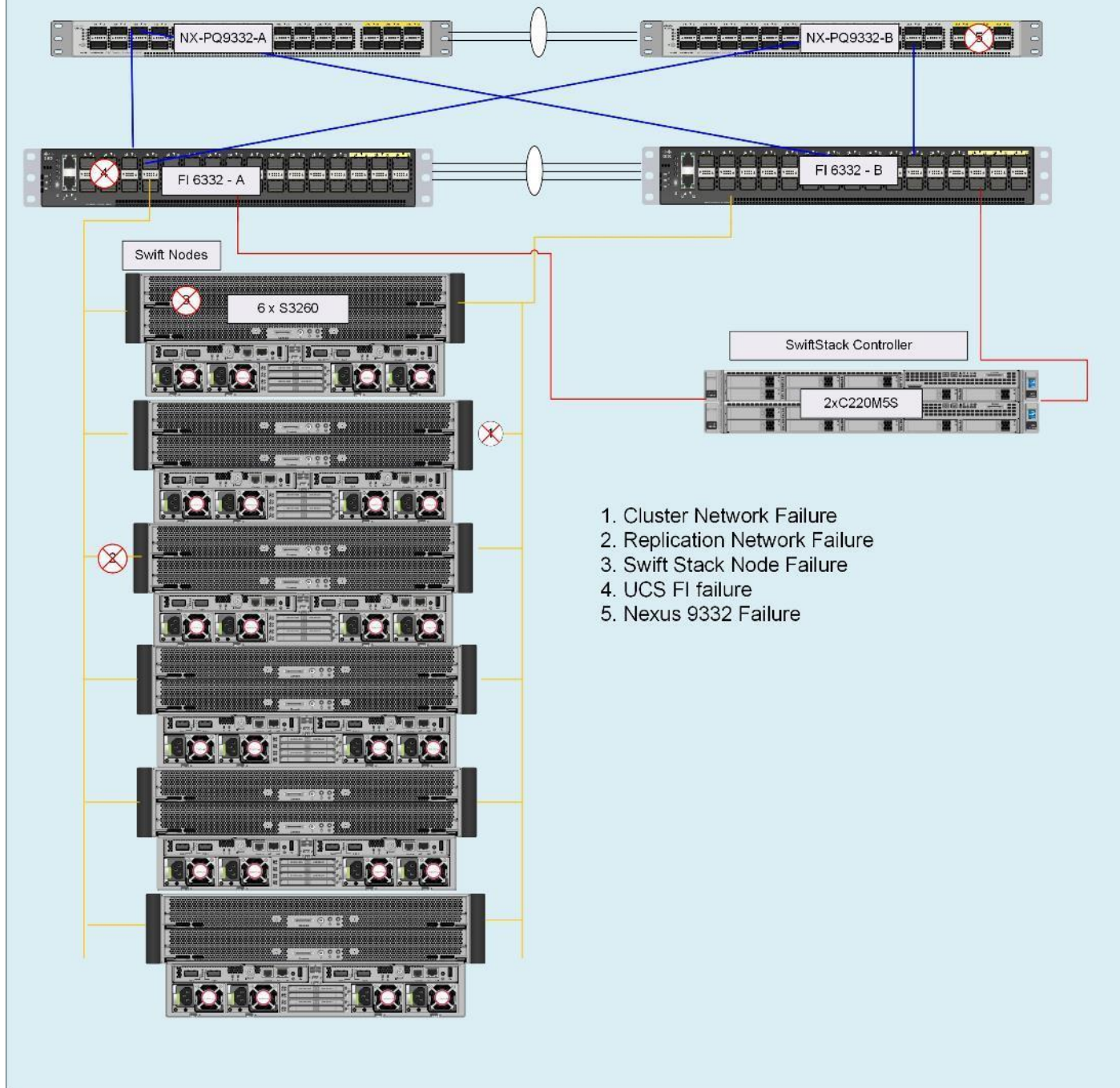
Network Layout



High Availability

As part of hardware and software resiliency, the following tests are being conducted on the test bed. The results of the tests will be included in the deployment guide.

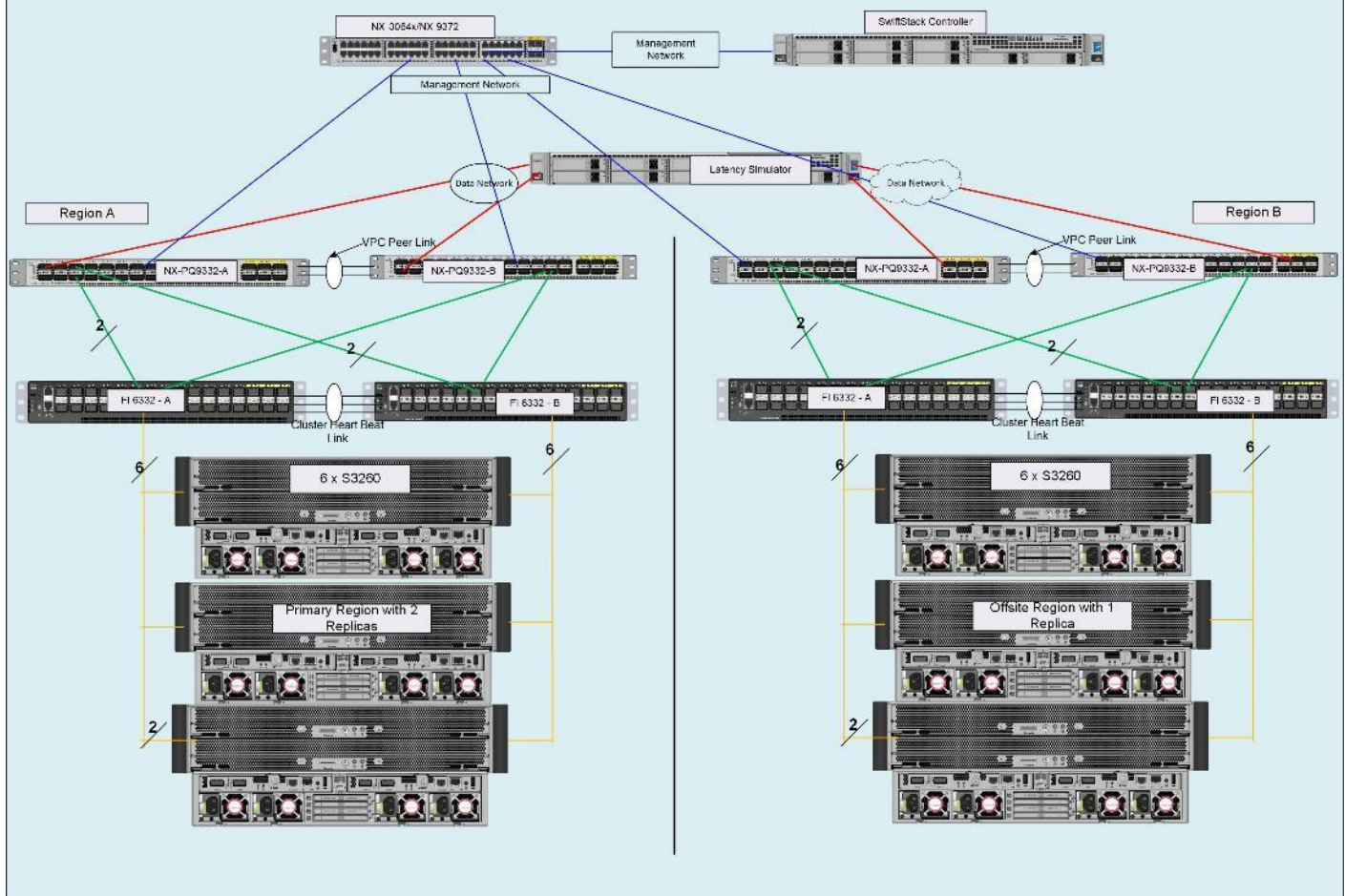
Cisco UCS Hardware Failure Tests for High Availability



Multi Region Deployment

Multi Region deployment is being attempted by simulating latency between the two regions. Most of the deployment steps and performance criteria will be included in a separate white paper that will follow this design guide.

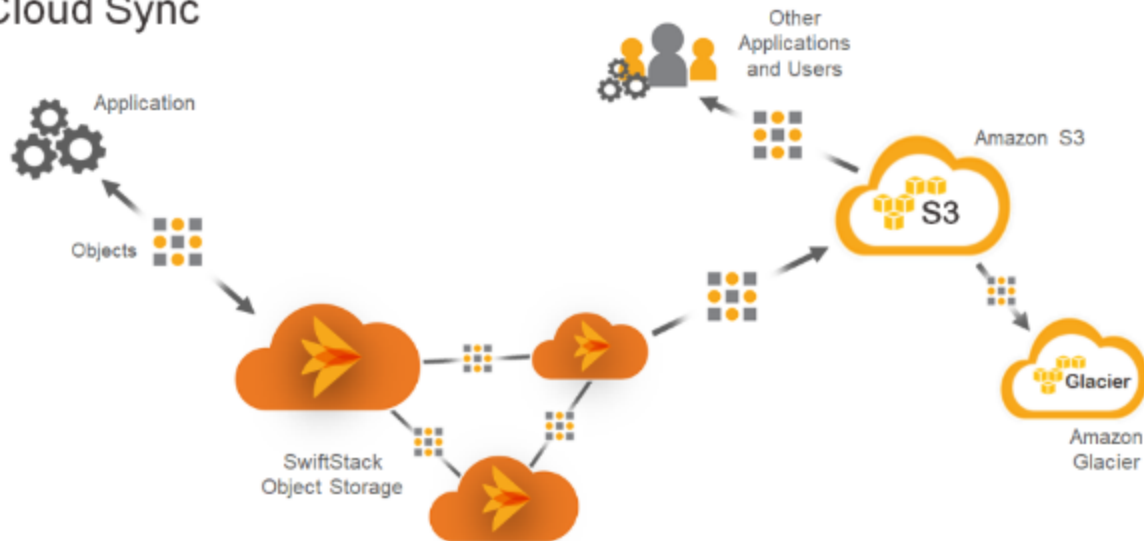
SwiftStack with Multiple Regions



Hybrid Cloud Deployment

Cloud sync will be setup from the multi-region deployment to one of the cloud providers like AWS/Google. The steps needed for setup and any discoveries will be documented too. With Cloud Sync in SwiftStack, data can automatically and continuously be synchronized to the public cloud based on a policy you define. It also allows you to provide access to specific data in a public bucket as an alternative to opening up your private cloud. And it can even assist with cloud bursting or archiving to Amazon Glacier.

Cloud Sync



Deployment Guide

The test bed was deployed with 12 x Cisco UCS S3260 storage nodes. While few functional tests were done, more to follow as part of deployment guide. However, the deployment, Performance, High Availability and Sizing Guide lines are being worked out while this Design guide was written. More about the deployment steps with any other best practices discovered as part of the setup will be documented in the deployment guide.

Figure 12 Snapshot from Test Bed



Summary

Cisco UCS Infrastructure for SwiftStack Software-Defined Storage is an integrated solution for deploying SwiftStack and combines the value of Intel Xeon architecture, Cisco data center hardware and software, along with Red Hat Linux. The solution is validated and supported by Cisco and SwiftStack, to increase the speed of deployment and reduce the risk of scaling from proof-of-concept to full enterprise production.

Cisco UCS hardware with Cisco UCS Manager Software brings an integrated, scalable, multi-chassis platform in which all resources participate in a unified management domain. Creating and cloning service profiles from its templates and maintaining the hardware from a single pane of glass not only provides rapid provisioning of hardware but also makes management and firmware upgrades simpler.

SwiftStack storage is optimized for unstructured data, which is growing at an ever-increasing rate inside most thriving enterprises. When media assets, scientific research data, and even backup archives live in a multi-tenant storage cloud, utilization of this valuable data increases while driving out unnecessary costs. The SwiftStack software has no single points of failure, and requires no downtime during any upgrades, scaling, planned maintenance or unplanned system operations and is with self-healing capabilities.

This Cisco Validated Design is a partnership from Cisco Systems, and SwiftStack. Combining these technologies, expertise and experience in the field, we are able to provide an enterprise ready hardware and software solution.

About the Authors

Ramakrishna Nishtala, Cisco Systems, Inc.

Ramakrishna Nishtala is a Technical Leader in Cisco UCS and Data Center solutions group and has several years of experience in IT infrastructure, Automation, Virtualization and Cloud computing. In his current role at Cisco Systems, he works on best practices, optimization and performance tuning on OpenStack and other Open Source solutions like Swift, Ceph, Dockers, etc., on Cisco UCS platforms. Prior to this he was involved in data center migration strategies, compute and storage consolidation, end-to-end performance optimization on databases, application and web servers and solutions engineering.

Acknowledgements

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

- **Chris O'Brien**, Cisco Systems, Inc.
- Jawwad Memon, Cisco Systems, Inc.
- Hiren Chandramani, SwiftStack
- Johnny Wang, SwiftStack