

# Cisco HyperFlex Platforms for Big Data with Splunk

Splunk SmartStore on Cisco HyperFlex Platforms with SwiftStack Object Storage System on Cisco UCS S3260 Servers Deployment Guide

Last Updated: July 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 .....	6
Solution Overview .....	7
Introduction .....	7
Audience .....	8
Purpose of this Document .....	8
Solution Summary .....	9
Technology Overview .....	11
Cisco Unified Computing System .....	11
Cisco UCS Fabric Interconnect .....	12
Cisco UCS 6332 Fabric Interconnect .....	12
Cisco UCS 6332-16UP Fabric Interconnect .....	12
Cisco UCS 6454 Fabric Interconnect .....	13
Cisco UCS C220 M5 Rack Server .....	13
Cisco UCS S3260 M5 Storage Server .....	14
Cisco VIC 1387 MLOM Interface Cards .....	15
Cisco VIC 1457 MLOM Interface Cards .....	15
Cisco HyperFlex HX-Series Nodes .....	16
Cisco HyperFlex HXAF220c-M5SX All-Flash Node .....	16
Cisco HyperFlex HXAF240c-M5SX All-Flash Node .....	16
Cisco HyperFlex HX220c-M5SX Hybrid Node .....	17
Cisco HyperFlex HX240c-M5SX Hybrid Node .....	17
All-Flash Versus Hybrid .....	18
Cisco HyperFlex Compute-Only Nodes .....	18
Cisco HyperFlex Data Platform Software .....	19
Object Storage and SwiftStack Software .....	22
Splunk Enterprise for Big Data Analytics .....	24
Key Features of Splunk Enterprise .....	25
Splunk Enterprise Processing Components .....	25
SmartStore .....	26
Solution Design .....	28
Requirements .....	28
Splunk Enterprise in Virtual Environments .....	28
Splunk Virtual Machines on Cisco HyperFlex .....	33
SwiftStack Object Storage on Cisco UCS S3260 .....	34
Physical Components .....	36
Software Components .....	39
Licensing .....	40
Physical Topology .....	40
Topology Overview .....	40
Fabric Interconnects .....	41

Cisco HXAF240c-M5SX Servers Connectivity .....	41
Cisco UCS C220 M5 Servers Connectivity.....	42
Cisco UCS S3260 Servers Connectivity.....	42
Cisco UCS Uplink Connectivity .....	43
Logical Topology .....	44
Logical Network Design .....	44
VLANs and Subnets.....	46
Jumbo Frames.....	47
Considerations.....	48
Capacity.....	48
Scale .....	50
Performance .....	52
Deployment of Hardware and Software .....	54
Architecture.....	54
Deployment Guidelines.....	54
Racking and Cabling .....	55
Install Cisco HyperFlex Cluster (with Nested vCenter).....	55
Install Cisco HyperFlex Systems.....	55
Configure the Data Store.....	60
Install VMware vCenter .....	61
Configure VMware vSphere.....	67
HyperFlex Cluster Expansion with Computing-only Nodes.....	70
Perform Cisco HyperFlex Post-installation Configuration.....	75
Create Splunk Virtual Machine Templates .....	77
Create Splunk Base Virtual Machine Template.....	78
Configure Splunk Base Virtual Machine .....	85
Create Splunk Admin Virtual Machine Template.....	105
Create Splunk Search Head Virtual Machine Template .....	114
Create Splunk Indexer Virtual Machine Template .....	115
Create Splunk Virtual Machines.....	116
Install SwiftStack Object Storage System on Cisco UCS S3260 Servers.....	124
Configuration and Validation .....	134
Configure Splunk Enterprise Cluster.....	134
Splunk Enterprise Licenses .....	134
Configure Index Cluster .....	137
Configure Search Head Cluster.....	145
Configure Deployment Server .....	155
Configure Distributed Monitoring Console.....	161
Configure SmartStore Indexes on the SwiftStack Object Storage.....	180
Create SwiftStack Object Storage Containers for SmartStore.....	181
Configure SmartStore Indexes.....	185
Validation Testing.....	188

Data Ingestion.....	188
Verify Data Replication.....	193
Verify Transfer of Warm Buckets to the Remote Storage .....	197
Splunk SmartStore Cache Performance Testing .....	201
Bill of Materials .....	206
Summary .....	212
For More Information.....	212
Appendix.....	213
Appendix A: HyperFlex Cluster Capacity Calculations .....	213
Appendix B: PowerShell Script Example – Clone Splunk Virtual Machines .....	213
Appendix C: SmartStore indexes.conf File Example.....	215
Appendix D: Custom Event Generation Script .....	217
About the Authors .....	219
Acknowledgements .....	219



## Executive Summary

---

This CVD presents a validated scale-out data center analytics and security solution with Splunk Enterprise software that is deployed on the Cisco HyperFlex All Flash Data Platform as local computing and storage resources, with the SwiftStack object storage system on the Cisco UCS S3260 servers as S3 API compliant remote object stores.

Traditional tools for managing and monitoring IT infrastructures are inconsistent with the constant change happening in today's data centers. When problems arise, finding the root cause or gaining visibility across the infrastructure to proactively identify and prevent outages is nearly impossible.

Splunk Enterprise software is the platform that reliably collects and indexes machine data, from a single source to tens of thousands of sources, all in real time. Organizations typically start with Splunk to solve a specific problem, and then expand from there to address a broad range of use cases, such as application troubleshooting, IT infrastructure monitoring, security, business analytics, Internet of Things, and many others. As operational analytics become increasingly critical to day-to-day decision-making and Splunk deployments expand to terabytes of data, a high-performance, highly scalable infrastructure is critical to ensuring rapid and predictable delivery of insights. In addition, with virtualization and cloud infrastructures introducing additional complexity that results in an environment that is more challenging to control and manage, deploying Splunk software in a virtual, cloud or hybrid environment has become valuable for the IT engineers.

Splunk's new SmartStore architecture is a response to these challenges. SmartStore is an indexer and caching engine which enables the use of remote object stores to store the indexed data. SmartStore makes it possible to dramatically expand the capacity for business insights by retaining significantly more searchable data through the use of cost-effective and scale-out remote object storage, either from the Cloud such as Amazon S3, or from an on-premises system like SwiftStack, alongside smaller footprints of low-latency local storage. With SmartStore, most data resides on remote storage while the indexer maintains a local cache that contains a minimal amount of data: hot buckets, copies of warm buckets participating in active or recent searches, and bucket metadata.

Cisco HyperFlex™ systems provide an all-purpose virtualized server platform, with hypervisor hosts, network connectivity, and virtual server storage across a set of Cisco HyperFlex HX-Series x86 rack-mount servers. The platform combines the converged computing and networking capabilities provided by the Cisco Unified Computing System™ (Cisco UCS®) with next-generation hyperconverged storage software to uniquely provide the computing resources, network connectivity, storage, and hypervisor platform needed to run an entire virtual environment, all contained in a single uniform system. A proven industry-leading hyperconverged platform, Cisco HyperFlex platform is an optimized choice for a Splunk Enterprise deployment in a VMware ESXi virtual environment.

SwiftStack's large scale-out storage capacity can support much larger searchable buckets for Splunk and enables much longer data retention periods for SmartStore indexes. The clustered architecture provides the higher availability and improved data resiliency as compared to traditional Splunk deployments. It provides much faster search results as compared to storing search data in public cloud storage locations.

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

## Solution Overview

---

### Introduction

Today, the rapid growth of data and the need for data analytics brings a lot more pressure on the underlying IT infrastructure and is seen in many customers' environments. Splunk has addressed this challenge by releasing a new software feature called SmartStore.

The architectural goal of the SmartStore feature is to optimize the use of local storage, while maintaining the fast indexing and search capabilities characteristic of Splunk Enterprise deployments. SmartStore introduces a remote storage tier and a cache manager that allow data to reside either locally on indexers or on the remote storage tier. Data movement between the indexer and the remote storage tier is managed by the cache manager, which resides on the indexer. With SmartStore the remote object store becomes the location for master copies of warm buckets, while the indexer's local storage is used to cache copies of warm buckets currently participating in a search or that have a high likelihood of participating in a future search. SmartStore also allows the customers to manage their indexer storage and compute resources in a cost-effective manner by scaling those resources separately.

The choice of the hardware for deploying SmartStore is important because the efficiency of the infrastructure affects the efficiency of the application and the speed of data collection and processing, storage performance, and resource management. Where to install Splunk Enterprise software and who can provide good object storage devices for the SmartStore indexes will never be a simple choice for the customers.

Virtualization is an ideal solution towards scaling challenges. It is a technology that allows for the sharing and easy expansion of underlying hardware resources by multiple workloads. This approach leads to higher utilization of IT resources while providing necessary fault tolerance. Hyperconvergence is an evolving technology that leverages many benefits of virtualization. Cisco HyperFlex systems let you unlock the full potential of hyperconvergence and adapt IT to the needs of your workloads. With Cisco HyperFlex systems, customers have many choices and flexibilities to support different types of workloads without comprising their performance requirements.

Cisco HyperFlex systems deliver many enterprise-class features, such as:

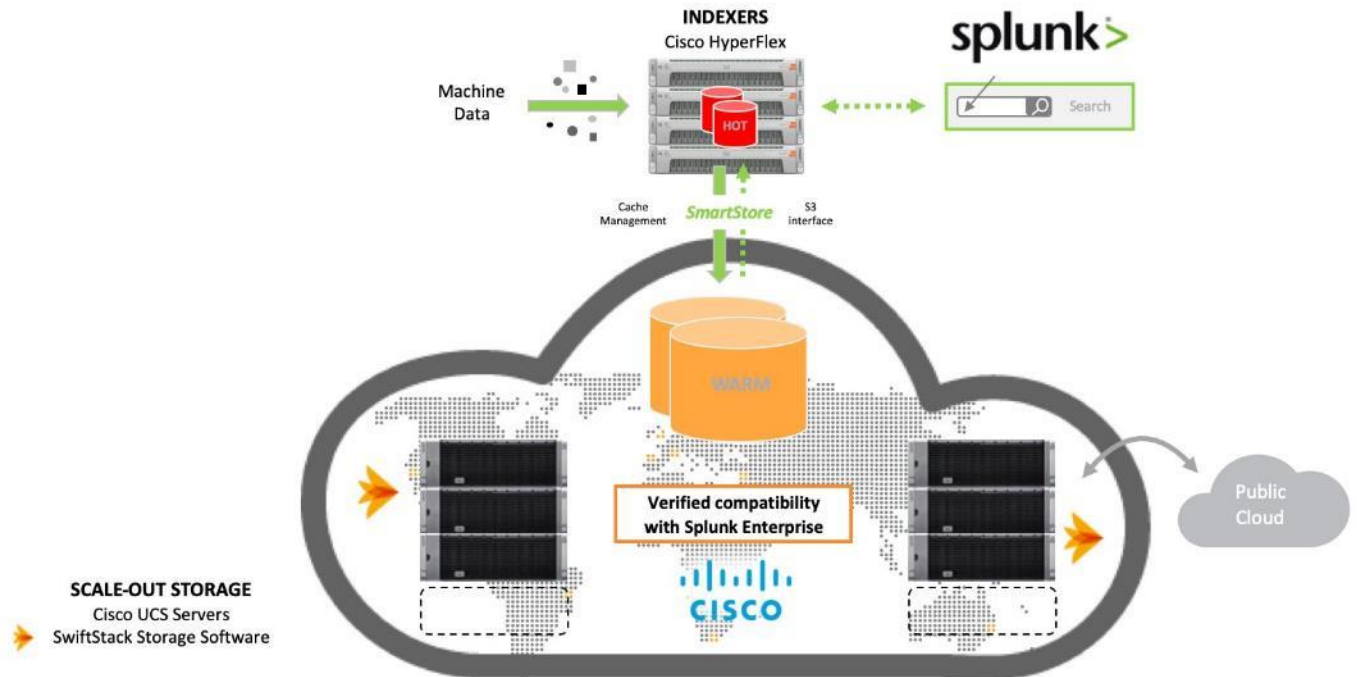
- A fully distributed log-structured file system that supports thin provisioning
- High performance and low latency from the flash-friendly architecture
- In-line data optimization with deduplication and compression
- Fast and space-efficient clones through metadata operations
- The flexibility to scale out computing and storage resources separately
- Data-at-rest encryption using hardware-based self-encrypting disks (SEDs)
- Non-Volatile Memory Express (NVMe)-based solid-state disk (SSD) support
- Native replication of virtual machine snapshots
- Cloud based central management

Installing Splunk Enterprise software on the Cisco HyperFlex All Flash Data Platform enables a validated analytics and security solution for a virtualized data center with simplified deployment on the integrated resources for computing, networking, and high-performance storage. In addition, Cisco HyperFlex is the only HCI platform that allows you to scale out computing and storage resources separately in today's market. This perfectly meets the requirements for the

deployment of Splunk SmartStore. That means the customers can easily scale out the indexers based on the performance demand by just adding compute-only nodes into the HyperFlex cluster.

Along with SwiftStack's scalable object storage the solution provides a less expensive option to store the indexing buckets. The buckets of SmartStore indexes ordinarily have just two active states: hot and warm. The cold state, which is used with non-SmartStore indexes to distinguish older data eligible for moving to cheap storage, is not necessary with SmartStore because warm buckets already reside on inexpensive remote storage. This solution fully utilizes the SmartStore's advanced caching mechanism that enables indexers to quickly return most search results from memory or local flash storage sitting on Cisco HyperFlex All Flash platforms while leveraging SwiftStack's scale-out remote object storage for all remaining data. All data remains searchable at any time, regardless of whether it is physically present in the cache or not.

Figure 1 High-level Solution Introduction



## Audience

The intended audience of this document includes, but is not limited to, sales engineers, field consultants, professional services, IT managers, partner engineering and customers who want to deploy Splunk Enterprise software on Cisco HyperFlex Platforms for Big Data, and those who want to deploy the SmartStore indexes with SwiftStack object storage systems on Cisco UCS S3260 servers. The readers of this document are expected to have the necessary understanding of Cisco UCS and HyperFlex, Splunk applications, Object storage with Swift and SwiftStack. External references are provided where applicable and it is recommended that the reader be familiar with these documents.

## Purpose of this Document

This document describes how to deploy Splunk indexing and searching virtual machines on a Cisco HyperFlex mixed (convergence and compute-only) cluster, and how to configure SmartStore indexes with SwiftStack object storage system on Cisco UCS S3260 servers. It includes design guidance covering the architecture and topologies, performance and scalability, a bill of materials, workarounds if required, while presenting a tested and validated solution, along with operational best practices.

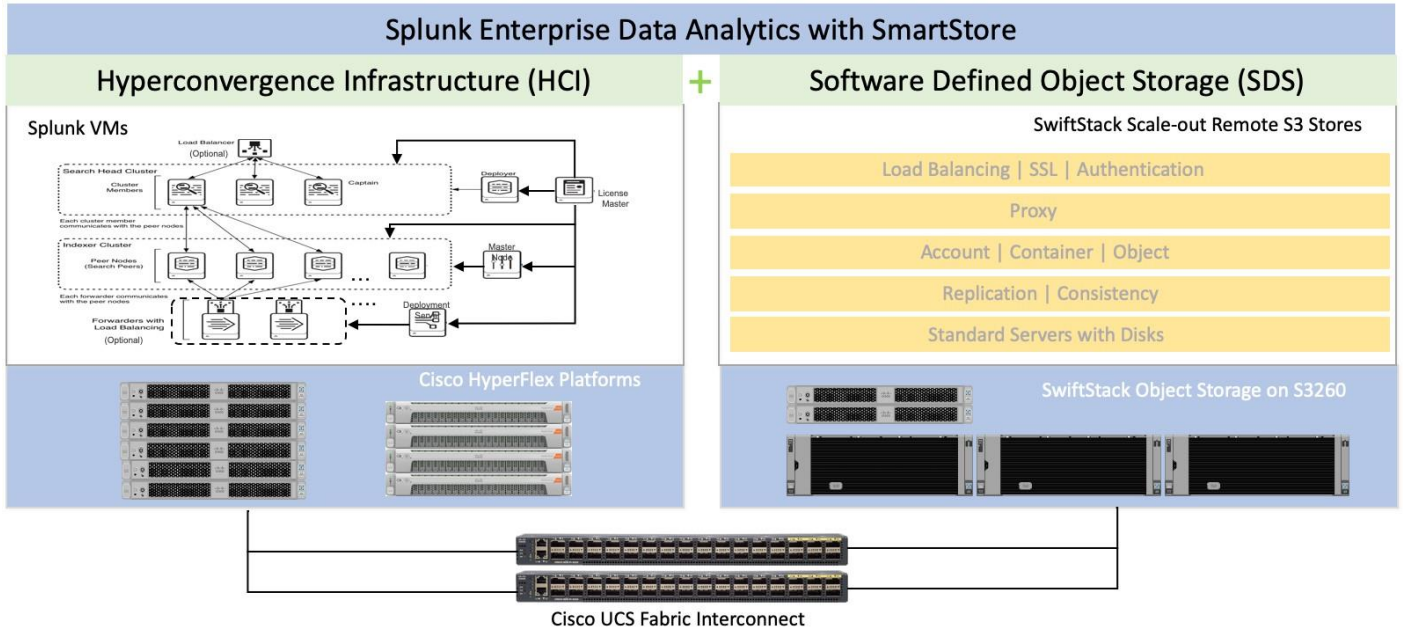


## Solution Summary

This CVD combines three evolving technologies into one solution: Big Data Analytics (Splunk), Cisco HyperFlex Platform (HCI), and SwiftStack Object Storage Software on Cisco UCS S3260 Servers (SDS). The solution consists of the following components:

- Splunk Enterprise (version 7.2.3) Virtual Machines
  - (10+) Indexers
  - (4) Search Heads
  - (1) Master Node
  - (1) License Master
  - (1) Search Head Deployer
  - (1) Deployment Server
- SwiftStack (version 6.19.1.3) Object Storage System on Cisco UCS S3260
  - (2) Cisco UCS C220 M5 Rack Servers (SwiftStack Controllers)
  - (3) Cisco UCS S3260 Dual-server Chassis
  - (6) Cisco UCS S3260 M5 Servers (SwiftStack PACO Nodes)
  - (6) System IO Controllers (SIOC) with 40G VIC 1380 adapters
- Cisco HyperFlex Data Platform (version 3.5.2a)
  - (4) Cisco HXAF240c M5 Rack Servers (Converged nodes)
  - (6) Cisco UCS C220 M5 Rack Servers (Compute-only nodes)
  - (10) Cisco UCS VIC 1387 40G adapters
- Cisco UCSM Management Software (version 4.0.1c)
  - (2) Cisco UCS 6332 40G Fabric Interconnects
- VMWare vSphere ESXi Hypervisor (version 6.5.0, 10884925)
- VMWare vSphere vCenter Appliance (version 6.5.0, 11347054)
- RedHat Enterprise Linux Server (version 7.5)

Figure 2 Solution Overview



## Technology Overview

Cisco HyperFlex systems are built on the Cisco UCS platform. Cisco UCS fabric interconnects provide a single point of connectivity integrating Cisco HyperFlex HX-Series nodes and other Cisco UCS servers into a single unified cluster. They can be deployed quickly and are highly flexible and efficient, reducing risk for the customer. Cisco HyperFlex delivers the simplicity, agility, scalability, and pay-as-you-grow economics of the cloud with the benefits of multisite, distributed computing at global scale.

### Cisco Unified Computing System

Cisco Unified Computing System (Cisco UCS) is a next-generation data center platform that unites compute, network, and storage access. The platform, optimized for virtual environments, is designed using open industry-standard technologies and aims to reduce total cost of ownership (TCO) and increase business agility. The system integrates a low-latency, lossless 10 Gigabit Ethernet, 25 Gigabit Ethernet, 40 Gigabit Ethernet or 100 Gigabit Ethernet unified network fabric with enterprise-class, x86-architecture servers. It is an integrated, scalable, multi chassis platform in which all resources participate in a unified management domain.

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 Processors.
- **Network:** The system is integrated onto a low-latency, lossless, 10-Gbps, 25-Gbps, 40-Gbps or 100-Gbps unified network fabric. This network foundation consolidates LANs, SANs, and high-performance computing networks which are often 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 storage access, the Cisco Unified Computing System can access storage over Ethernet, Fibre Channel, Fibre Channel over Ethernet (FCoE), and iSCSI. This provides customers with their choice of storage protocol and physical architecture, and enhanced 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.
- **Management:** The system uniquely integrates all system components which enable the entire solution to be managed as a single entity by the Cisco UCS Manager (UCSM). The Cisco UCS Manager has an intuitive graphical user interface (GUI), a command-line interface (CLI), and a robust application programming interface (API) to manage all system configuration and operations.

Cisco Unified Computing System is designed to deliver:

- A reduced Total Cost of Ownership 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. The system is managed, serviced and tested as a whole.
- Scalability through a design for hundreds of discrete servers and thousands of virtual machines and the capability to scale I/O bandwidth to match demand.

- Industry standards supported by a partner ecosystem of industry leaders.

## Cisco UCS Fabric Interconnect

The Cisco UCS Fabric Interconnect (FI) is a core part of the Cisco Unified Computing System, providing both network connectivity and management capabilities for the system. Depending on the model chosen, the Cisco UCS Fabric Interconnect offers line-rate, low-latency, lossless 10 Gigabit, 25 Gigabit, 40 Gigabit or 100 Gigabit Ethernet, Fibre Channel over Ethernet (FCoE) and Fibre Channel connectivity. Cisco UCS Fabric Interconnects provide the management and communication backbone for the Cisco UCS C-Series, S-Series and HX-Series Rack-Mount Servers, Cisco UCS B-Series Blade Servers and Cisco UCS 5100 Series Blade Server Chassis. All servers and chassis, and therefore all blades, attached to the Cisco UCS Fabric Interconnects become part of a single, highly available management domain. In addition, by supporting unified fabrics, the Cisco UCS Fabric Interconnects provide both the LAN and SAN connectivity for all servers within its domain.

From a networking perspective, the Cisco UCS 6200 Series uses a cut-through architecture, supporting deterministic, low latency, line rate 10 Gigabit Ethernet on all ports, up to 1.92 Tbps switching capacity and 160 Gbps bandwidth per chassis, independent of packet size and enabled services. The product family supports Cisco low-latency, lossless 10 Gigabit Ethernet unified network fabric capabilities, which increase the reliability, efficiency, and scalability of Ethernet networks. The Fabric Interconnect supports multiple traffic classes over the Ethernet fabric from the servers to the uplinks. Significant TCO savings come from an FCoE-optimized server design in which network interface cards (NICs), host bus adapters (HBAs), cables, and switches can be consolidated.

The Cisco UCS 6300 Series offers the same features while supporting higher performance, low latency, lossless, line rate 40 Gigabit Ethernet, with up to 2.56 Tbps of switching capacity. Backward compatibility and scalability are assured with the ability to configure 40 Gbps quad SFP (QSFP) ports as breakout ports using 4x10GbE breakout cables. Existing Cisco UCS servers with 10GbE interfaces can be connected in this manner, although Cisco HyperFlex nodes must use a 40GbE VIC adapter in order to connect to a Cisco UCS 6300 Series Fabric Interconnect.

The Cisco UCS 6400 Series offers the same features while supporting even higher performance, low latency, lossless, line rate 10/25/40/100 Gigabit Ethernet ports, with up to 3.82 Tbps of switching capacity. Backward compatibility and scalability are assured with the ability to configure 10 Gbps ports as well as 25 Gbps ports for the new servers. Existing Cisco UCS servers with 10GbE interfaces can be connected in this manner. Cisco HyperFlex nodes can use a 10/25 GbE VIC adapters in order to connect to a Cisco UCS 6400 Series Fabric Interconnect.

## Cisco UCS 6332 Fabric Interconnect

The Cisco UCS 6332 Fabric Interconnect is a one-rack-unit (1RU) 40 Gigabit Ethernet and FCoE switch offering up to 2560 Gbps of throughput. The switch has 32 40-Gbps fixed Ethernet and FCoE ports. Up to 24 of the ports can be reconfigured as 4x10Gbps breakout ports, providing up to 96 10-Gbps ports.

Figure 3 Cisco UCS 6332 Fabric Interconnect



## Cisco UCS 6332-16UP Fabric Interconnect

The Cisco UCS 6332-16UP Fabric Interconnect is a one-rack-unit (1RU) 10/40 Gigabit Ethernet, FCoE, and native Fibre Channel switch offering up to 2430 Gbps of throughput. The switch has 24 40-Gbps fixed Ethernet and FCoE ports, plus 16 1/10-Gbps fixed Ethernet, FCoE, or 4/8/16 Gbps FC ports. Up to 18 of the 40-Gbps ports can be reconfigured as 4x10Gbps breakout ports, providing up to 88 total 10-Gbps ports.

Figure 4 Cisco UCS 6332-16UP Fabric Interconnect



**Note:** When used for a Cisco HyperFlex deployment, due to mandatory QoS settings in the configuration, the 6332 and 6332-16UP will be limited to a maximum of four 4x10Gbps breakout ports, which can be used for other non-HyperFlex servers.

### Cisco UCS 6454 Fabric Interconnect

The Cisco UCS 6454 54-Port Fabric Interconnect is a One-Rack-Unit (1RU) 10/25/40/100 Gigabit Ethernet, FCoE and Fibre Channel switch offering up to 3820 Gbps throughput and up to 54 ports. The switch has 36 10/25-Gbps Ethernet ports, 4 1/10/25-Gbps Ethernet ports, 6 40/100-Gbps Ethernet uplink ports and 8 unified ports that can support 8 10/25-Gbps Ethernet ports or 8/16/32-Gbps Fibre Channel ports. All Ethernet ports are capable of supporting FCoE.

Figure 5 Cisco UCS 6454 Fabric Interconnect



### Cisco UCS C220 M5 Rack Server

The Cisco UCS C220 M5 Rack 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 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' 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, delivering significant performance and efficiency gains that will improve your application performance. The Cisco UCS C220 M5 delivers outstanding levels of expandability and performance in a compact package, with:

- Latest (second generation) Intel Xeon Scalable CPUs with up to 28 cores per socket
- Supports first-generation Intel Xeon Scalable CPUs with up to 28 cores per socket
- Up to 24 DDR4 DIMMs for improved performance
- Support for the Intel Optane DC Persistent Memory
- Up to 10 Small-Form-Factor (SFF) 2.5-inch drives or 4 Large-Form-Factor (LFF) 3.5-inch drives (77 TB storage capacity with all NVMe PCIe SSDs)
- 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
- Dual embedded Intel x550 10GBASE-T LAN-On-Motherboard (LOM) ports

Figure 6 Cisco UCS C220 M5 Rack Server



## Cisco UCS S3260 M5 Storage Server

The Cisco UCS<sup>®</sup> S3260 Storage Server is a modular, high-density, high-availability, dual-node storage-optimized server well suited for service providers, enterprises, and industry-specific environments. It provides dense, cost-effective storage to address your ever-growing data needs. Designed for a new class of data-intensive workloads, it is simple to deploy and excellent for applications for big data, data protection, software-defined storage environments, scale-out unstructured data repositories, media streaming, and content distribution.

Figure 7 Cisco UCS S3260 M5 Storage Server



The Cisco UCS S3260 server helps you achieve the highest levels of data availability and performance. With dual-node capability that is based on the 2nd Gen Intel<sup>®</sup> Xeon<sup>®</sup> Scalable and Intel Xeon Scalable processor, it features up to 840 TB of local storage in a compact 4-Rack-Unit (4RU) form factor. The drives can be configured with enterprise-class Redundant Array of Independent Disks (RAID) redundancy or with a pass-through Host Bus Adapter (HBA) controller. Network connectivity is provided with dual-port 40-Gbps nodes in each server, with expanded unified I/O capabilities for data migration between Network-Attached Storage (NAS) and SAN environments. This storage-optimized server comfortably fits in a standard 32-inch-depth rack, such as the Cisco<sup>®</sup> R 42610 Rack. Highlights of the Cisco UCS S3260 server are:

- Dual 2-socket server nodes based on 2nd Gen Intel Xeon Scalable and Intel Xeon Scalable processors with up to 48 cores per server node
- Up to 1.5 TB of DDR4 memory per M5 server node and up to 1 TB of Intel Optane<sup>™</sup> DC Persistent Memory
- Support for high-performance Nonvolatile Memory Express (NVMe) and flash memory
- Massive 840-TB data storage capacity that easily scales to petabytes with Cisco UCS Manager software
- Policy-based storage management framework for zero-touch capacity on demand

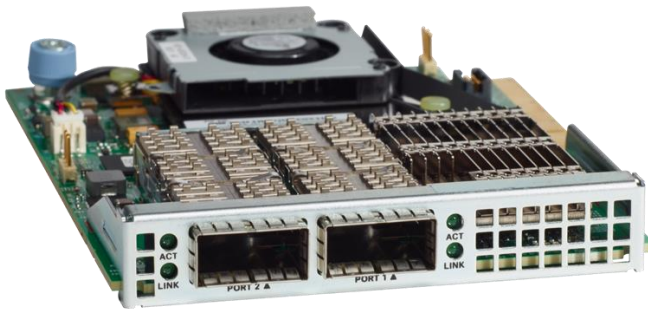
- Dual-port 40-Gbps system I/O controllers with a Cisco UCS Virtual Interface Card 1300 platform embedded chip or PCIe-based system I/O controller for Quad Port 10/25G Cisco VIC 1455 or Dual Port 100G Cisco VIC 1495
- Unified I/O for Ethernet or Fibre Channel to existing NAS or SAN storage environments
- Support for Cisco bidirectional transceivers, with 40-Gbps connectivity over existing 10-Gbps cabling infrastructure

### Cisco VIC 1387 MLOM Interface Cards

The Cisco UCS VIC 1387 Card is a dual-port Enhanced Quad Small Form-Factor Pluggable (QSFP+) 40-Gbps Ethernet and Fibre Channel over Ethernet (FCoE)-capable PCI Express (PCIe) modular LAN-on-motherboard (mLOM) adapter installed in the Cisco UCS HX-Series or C-Series Rack Servers. The VIC 1387 is used in conjunction with the Cisco UCS 6332 or 6332-16UP model Fabric Interconnects.

The mLOM slot can be used to install a Cisco VIC without consuming a PCIe slot, which provides greater I/O expandability. It incorporates next-generation converged network adapter (CNA) technology from Cisco, providing investment protection for future feature releases. The card enables a policy-based, stateless, agile server infrastructure that can present up to 256 PCIe standards-compliant interfaces to the host, each dynamically configured as either a network interface card (NICs) or host bus adapter (HBA). The personality of the interfaces is set programmatically using the service profile associated with the server. The number, type (NIC or HBA), identity (MAC address and World Wide Name [WWN]), failover policy, adapter settings, bandwidth, and quality-of-service (QoS) policies of the PCIe interfaces are all specified using the service profile.

Figure 8 Cisco VIC 1387 mLOM Card



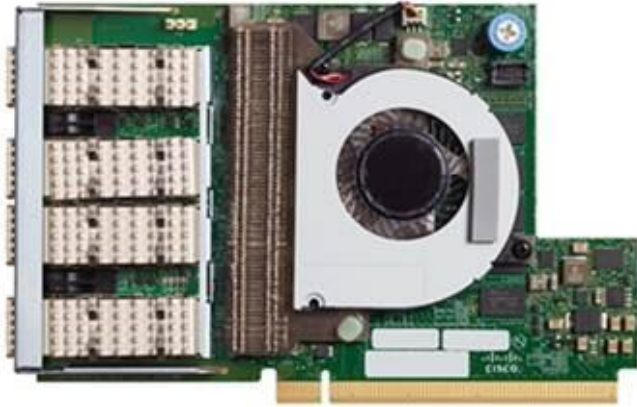
**Note:** Hardware revision V03 or later of the Cisco VIC 1387 card is required for the Cisco HyperFlex HX-series servers.

### Cisco VIC 1457 MLOM Interface Cards

The Cisco UCS VIC 1457 is a quad-port Small Form-Factor Pluggable (SFP28) 10/25-Gbps Ethernet and Fibre Channel over Ethernet (FCoE)-capable PCI Express (PCIe) modular LAN-on-motherboard (mLOM) adapter installed in the Cisco UCS HX-Series or C-Series Rack Servers. The Cisco UCS VIC 1457 is used in conjunction with the Cisco UCS 6454 model Fabric Interconnects.

The mLOM slot can be used to install a Cisco VIC without consuming a PCIe slot, which provides greater I/O expandability. It incorporates next-generation converged network adapter (CNA) technology from Cisco, providing investment protection for future feature releases. The card enables a policy-based, stateless, agile server infrastructure that can present up to 256 PCIe standards-compliant interfaces to the host, each dynamically configured as either a network interface card (NICs) or host bus adapter (HBA). The personality of the interfaces is set programmatically using the service profile associated with the server. The number, type (NIC or HBA), identity (MAC address and World Wide Name [WWN]), failover policy, adapter settings, bandwidth, and quality-of-service (QoS) policies of the PCIe interfaces are all specified using the service profile.

Figure 9 Cisco VIC 1457 mLOM Card



## Cisco HyperFlex HX-Series Nodes

A Cisco HyperFlex cluster requires a minimum of three HX-Series nodes. Data is replicated across at least two of these nodes, and a third node is required for continuous operation in the event of a single-node failure. The HX-Series nodes combine the CPU and RAM resources for hosting guest virtual machines with a shared pool of the physical storage resources used by the HX Data Platform software. HX-Series hybrid nodes use a combination of solid-state disks (SSDs) for caching and hard-disk drives (HDDs) for the capacity layer. HX-Series all-flash nodes use SSD or NVMe storage for the caching layer and SSDs for the capacity layer.

### Cisco HyperFlex HXAF220c-M5SX All-Flash Node

This small footprint Cisco HyperFlex all-flash model contains a 240 GB M.2 form factor solid-state disk (SSD) that acts as the boot drive, a 240 GB housekeeping SSD drive, either a single 375 GB Optane NVMe SSD, a 1.6 TB NVMe SSD or 400GB SAS SSD write-log drive, and six to eight 960 GB or 3.8 TB SATA SSD drives for storage capacity. For configurations requiring self-encrypting drives, the caching SSD is replaced with an 800 GB SAS SED SSD, and the capacity disks are also replaced with either 800 GB, 960 GB or 3.8 TB SED SSDs.

Figure 10 HXAF220c-M5SX All-Flash Node



### Cisco HyperFlex HXAF240c-M5SX All-Flash Node

This capacity optimized Cisco HyperFlex all-flash model contains a 240 GB M.2 form factor solid-state disk (SSD) that acts as the boot drive, a 240 GB housekeeping SSD drive, either a single 375 GB Optane NVMe SSD, a 1.6 TB NVMe SSD or 400GB SAS SSD write-log drive installed in a rear hot swappable slot, and six to twenty-three 960 GB or 3.8 TB SATA SSD drives for storage capacity. For configurations requiring self-encrypting drives, the caching SSD is replaced with an 800 GB SAS SED SSD, and the capacity disks are also replaced with either 800 GB, 960 GB or 3.8 TB SED SSDs.



Figure 11 HXAF240c-M5SX Node



**Note:** Either a 375 GB Optane NVMe SSD, a 400 GB SAS SSD or 1.6 TB NVMe SSD caching drive may be chosen. While the NVMe options can provide a higher level of performance, the partitioning of the three disk options is the same, therefore the amount of cache available on the system is the same regardless of the model chosen.

### Cisco HyperFlex HX220c-M5SX Hybrid Node

This small footprint Cisco HyperFlex hybrid model contains a minimum of six, and up to eight 1.8 terabyte (TB) or 1.2 TB SAS hard disk drives (HDD) that contribute to cluster storage capacity, a 240 GB SSD housekeeping drive, a 480 GB or 800 GB SSD caching drive, and a 240 GB M.2 form factor SSD that acts as the boot drive. For configurations requiring self-encrypting drives, the caching SSD is replaced with an 800 GB SAS SED SSD, and the capacity disks are replaced with 1.2TB SAS SED HDDs.

Figure 12 HX220c-M5SX Node



**Note:** Either a 480 GB or 800 GB caching SAS SSD may be chosen. This option is provided to allow flexibility in ordering based on product availability, pricing and lead times. There is no performance, capacity, or scalability benefit in choosing the larger disk.

### Cisco HyperFlex HX240c-M5SX Hybrid Node

This capacity optimized Cisco HyperFlex hybrid model contains a minimum of six and up to twenty-three 1.8 TB or 1.2 TB SAS small form factor (SFF) hard disk drives (HDD) that contribute to cluster storage, a 240 GB SSD housekeeping drive, a single 1.6 TB SSD caching drive installed in a rear hot swappable slot, and a 240 GB M.2 form factor SSD that acts as the boot drive. For configurations requiring self-encrypting drives, the caching SSD is replaced with a 1.6 TB SAS SED SSD, and the capacity disks are replaced with 1.2TB SAS SED HDDs.

Figure 13 HX240c-M5SX Node



## All-Flash Versus Hybrid

The initial HyperFlex product release featured hybrid converged nodes, which use a combination of solid-state disks (SSDs) for the short-term storage caching layer, and hard disk drives (HDDs) for the long-term storage capacity layer. The hybrid HyperFlex system is an excellent choice for entry-level or midrange storage solutions, and hybrid solutions have been successfully deployed in many non-performance sensitive virtual environments. Meanwhile, there is significant growth in deployment of highly performance sensitive and mission critical applications. The primary challenge to the hybrid HyperFlex system from these highly performance sensitive applications, is their increased sensitivity to high storage latency. Due to the characteristics of the spinning hard disks, it is unavoidable that their higher latency becomes the bottleneck in the hybrid system. Ideally, if all of the storage operations were to occur in the caching SSD layer, the hybrid system's performance will be excellent. But in several scenarios, the amount of data being written and read exceeds the caching layer capacity, placing larger loads on the HDD capacity layer, and the subsequent increases in latency will naturally result in reduced performance.

Cisco All-Flash HyperFlex systems are an excellent option for customers with a requirement to support high performance, latency sensitive workloads. With a purpose built, flash-optimized and high-performance log based filesystem, the Cisco All-Flash HyperFlex system provides:

- Predictable high performance across all the virtual machines on HyperFlex All-Flash and compute-only nodes in the cluster.
- Highly consistent and low latency, which benefits data-intensive applications and databases such as Microsoft SQL and Oracle.
- Support for NVMe caching SSDs, offering an even higher level of performance.
- Future ready architecture that is well suited for flash-memory configuration:
  - Cluster-wide SSD pooling maximizes performance and balances SSD usage so as to spread the wear.
  - A fully distributed log-structured filesystem optimizes the data path to help reduce write amplification.
  - Large sequential writing reduces flash wear and increases component longevity.
  - Inline space optimization, e.g. deduplication and compression, minimizes data operations and reduces wear.
- Lower operating cost with the higher density drives for increased capacity of the system.
- Cloud scale solution with easy scale-out and distributed infrastructure and the flexibility of scaling out independent resources separately.

Cisco HyperFlex support for hybrid and all-flash models now allows customers to choose the right platform configuration based on their capacity, applications, performance, and budget requirements. All-flash configurations offer repeatable and sustainable high performance, especially for scenarios with a larger working set of data, in other words, a large amount of data in motion. Hybrid configurations are a good option for customers who want the simplicity of the Cisco HyperFlex solution, but their needs focus on capacity-sensitive solutions, lower budgets, and fewer performance-sensitive applications.

## Cisco HyperFlex Compute-Only Nodes

All current model Cisco UCS M4 and M5 generation servers, except the Cisco UCS C880 M4 and Cisco UCS C880 M5, may be used as compute-only nodes connected to a Cisco HyperFlex cluster, along with a limited number of previous M3 generation servers. Any valid CPU and memory configuration is allowed in the compute-only nodes, and the servers can be configured to boot from SAN, local disks, or internal SD cards. The following servers may be used as compute-only nodes:

- Cisco UCS B200 M4 Blade Server

- Cisco UCS B200 M5 Blade Server
- Cisco UCS B260 M4 Blade Server
- Cisco UCS B420 M4 Blade Server
- Cisco UCS B460 M4 Blade Server
- Cisco UCS B480 M5 Blade Server
- Cisco UCS C220 M4 Rack-Mount Servers
- Cisco UCS C220 M5 Rack-Mount Servers
- Cisco UCS C240 M4 Rack-Mount Servers
- Cisco UCS C240 M5 Rack-Mount Servers
- Cisco UCS C460 M4 Rack-Mount Servers
- Cisco UCS C480 M5 Rack-Mount Servers

## Cisco HyperFlex Data Platform Software

The Cisco HyperFlex delivers a new generation of flexible, scalable, enterprise-class hyperconverged solutions. The solution also delivers storage efficiency features such as thin provisioning, data deduplication, and compression for greater capacity and enterprise-class performance. Additional operational efficiency is facilitated through features such as cloning and snapshots.

The complete end-to-end hyperconverged solution provides the following benefits to customers:

- **Simplicity:** The solution is designed to be deployed and managed easily and quickly through familiar tools and methods. No separate management console is required for the Cisco HyperFlex solution.
- **Centralized hardware management:** The cluster hardware is managed in a consistent manner by service profiles in Cisco UCS Manager. Cisco UCS Manager also provides a single console for solution management, including firmware management. Cisco HyperFlex HX Data Platform clusters are managed through a plug-in to VMware vCenter.
- **High availability:** Component redundancy is built in to most levels at the node. Cluster-level tolerance of node, network, and fabric interconnect failures is implemented as well.
- **Enterprise-class storage features:** Complementing the other management efficiencies are features such as thin provisioning, data deduplication, compression, cloning, and snapshots to address concerns related to overprovisioning of storage.
- **Flexibility with a "pay-as-you-grow" model:** Customers can purchase the exact amount of computing and storage they need and expand one node at a time up to the supported cluster node limit.
- **Agility to support different workloads:** Support for both hybrid and all-flash models allows customers to choose the right platform configuration for capacity-sensitive applications or performance-sensitive applications according to budget requirements.

The Cisco HyperFlex HX Data Platform is a purpose-built, high-performance, distributed file system with a wide array of enterprise-class data management services. The data platform's innovations redefine distributed storage technology, exceeding the boundaries of first-generation hyperconverged infrastructures. The data platform has all the features that you would expect of an enterprise shared storage system, eliminating the need to configure and maintain complex Fibre

Channel storage networks and devices. The platform simplifies operations and helps ensure data availability. Enterprise-class storage features include the following:

- Replication of all written data across the cluster so that data availability is not affected if single or multiple components fail (depending on the replication factor configured).
- Deduplication is always on, helping reduce storage requirements in which multiple operating system instances in client virtual machines result in large amounts of duplicate data.
- Compression further reduces storage requirements, reducing costs, and the log-structured file system is designed to store variable-sized blocks, reducing internal fragmentation.
- Thin provisioning allows large volumes to be created without requiring storage to support them until the need arises, simplifying data volume growth and making storage a “pay as you grow” proposition.
- Fast, space-efficient clones rapidly replicate virtual machines simply through metadata operations.
- Snapshots help facilitate backup and remote-replication operations: needed in enterprises that require always-on data availability.

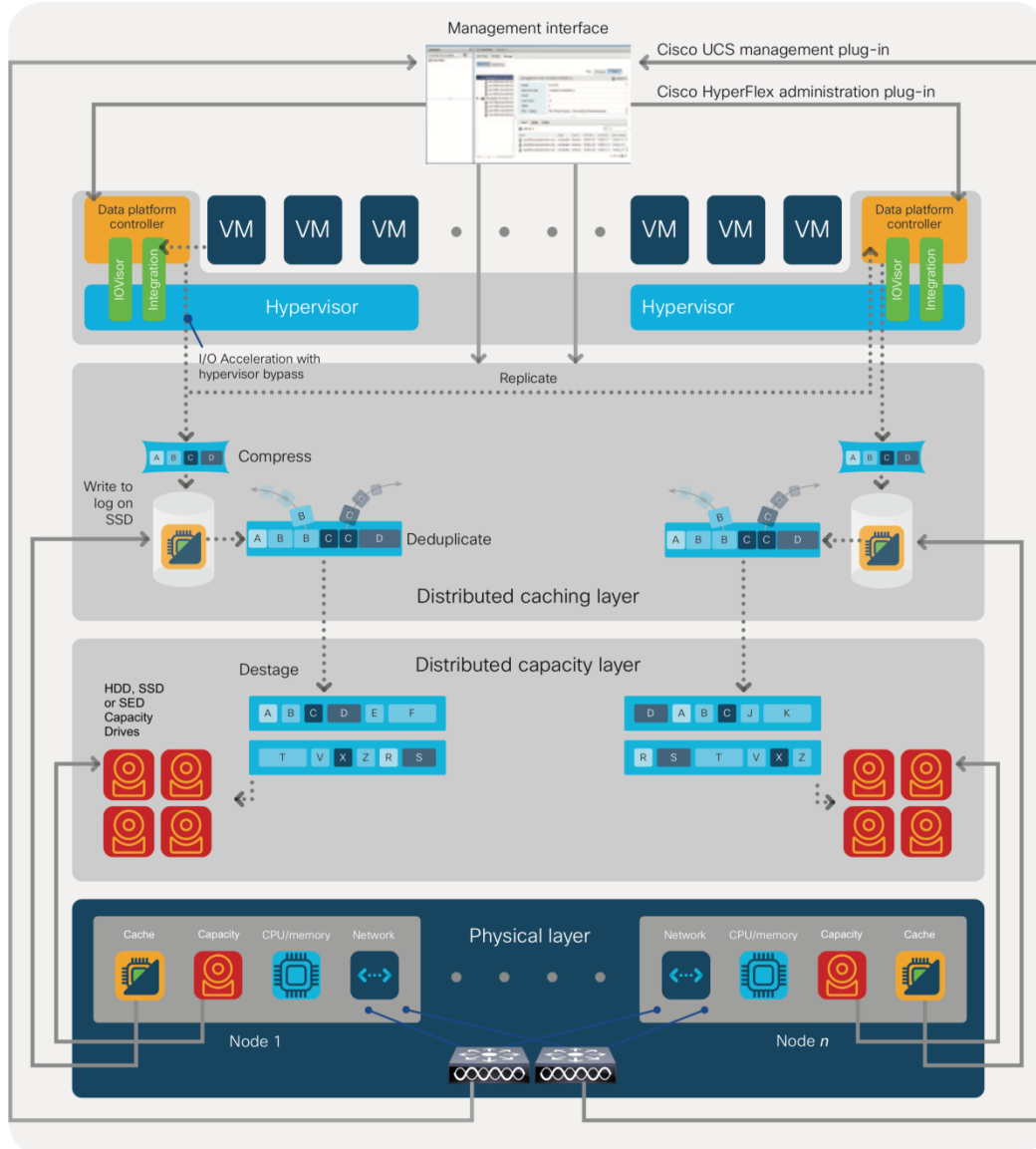
The HX Data Platform can be administered through a VMware vSphere web client plug-in or through the HTML5-based native Cisco HyperFlex Connect management tool. Additionally, since the HX Data Platform Release 2.6, Cisco HyperFlex systems can also be managed remotely by the Cisco Intersight™ cloud-based management platform. Through the centralized point of control for the cluster, administrators can create data store, monitor the data platform health, and manage resource use.

An HX Data Platform controller resides on each node and implements the Cisco HyperFlex HX Distributed File System. The storage controller runs in user space within a virtual machine, intercepting and handling all I/O requests from guest virtual machines. The storage controller virtual machine uses the VMDirectPath I/O feature to provide PCI pass-through control of the physical server’s SAS disk controller. This approach gives the controller virtual machine full control of the physical disk resources. The controller integrates the data platform into VMware software through three preinstalled VMware ESXi vSphere Installation Bundles (VIBs): the VMware API for Array Integration (VAAI), a customized IOvisor agent that acts as a stateless Network File System (NFS) proxy, and a customized stHypervisorSvc agent for Cisco HyperFlex data protection and virtual machine replication.

The HX Data Platform controllers handle all read and write requests from the guest virtual machines to the virtual machine disks (VMDKs) stored in the distributed data stores in the cluster. The data platform distributes the data across multiple nodes of the cluster and across multiple capacity disks in each node according to the replication-level policy selected during cluster setup. The replication-level policy is defined by the replication factor (RF) parameter. When  $RF = 3$ , a total of three copies of the blocks are written and distributed to separate locations for every I/O write committed to the storage layer; when  $RF = 2$ , a total of two copies of the blocks are written and distributed.

Figure 14 shows the movement of data in the HX Data Platform.

Figure 14 Cisco HyperFlex HX Data Platform Data Movement



For each write operation, the data is intercepted by the IO Visor module on the node on which the virtual machine is running, a primary node is determined for that particular operation through a hashing algorithm, and the data is then sent to the primary node. The primary node compresses the data in real time and writes the compressed data to its caching SSD, and replica copies of that compressed data are written to the caching SSD of the remote nodes in the cluster, according to the replication factor setting. Because the virtual disk contents have been divided and spread out through the hashing algorithm, the result of this method is that all write operations are spread across all nodes, avoiding problems related to data locality and helping prevent “noisy” virtual machines from consuming all the I/O capacity of a single node. The write operation will not be acknowledged until all three copies are written to the caching-layer SSDs. Written data is also cached in a write log area resident in memory in the controller virtual machine, along with the write log on the caching SSDs. This process speeds up read requests when read operations are requested on data that has recently been written.

The HX Data Platform constructs multiple write caching segments on the caching SSDs of each node in the distributed cluster. As write-cache segments become full. Based on policies accounting for I/O load and access patterns, those write-cache segments are locked, and new write operations roll over to a new write-cache segment. The data in the now-locked cache segment is destaged to the HDD capacity layer of the nodes for a hybrid system or to the SSD capacity layer of the nodes for an all-flash system. During the destaging process, data is deduplicated before being written to the capacity

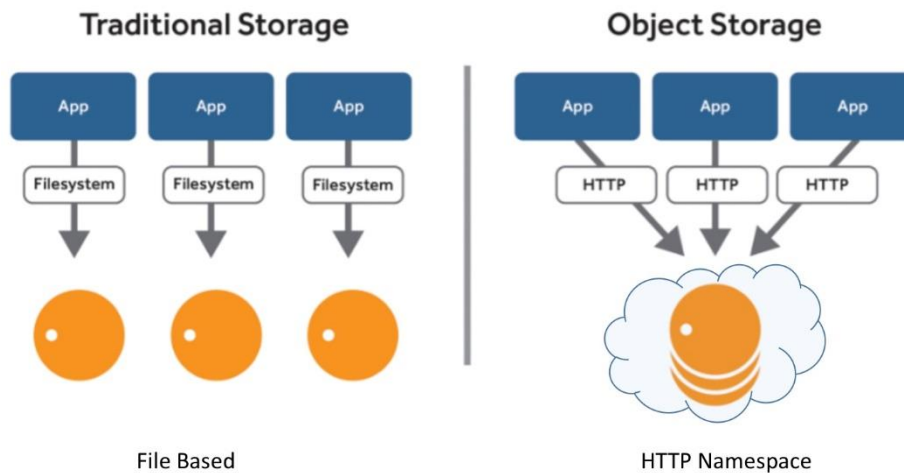
storage layer, and the resulting data can now be written to the HDDs or SSDs of the server. On hybrid systems, the now deduplicated and compressed data is also written to the dedicated read-cache area of the caching SSD, which speeds up read requests for data that has recently been written. When the data is destaged to an HDD, it is written in a single sequential operation, avoiding disk-head seek thrashing on the spinning disks and accomplishing the task in a minimal amount of time. Deduplication, compression, and destaging take place with no delays or I/O penalties for the guest virtual machines making requests to read or write data, which benefits both the HDD and SSD configurations.

For data read operations, data may be read from multiple locations. For data that was very recently written, the data is likely to still exist in the write log of the local platform controller memory or in the write log of the local caching-layer disk. If local write logs do not contain the data, the distributed file system metadata will be queried to see if the data is cached elsewhere, either in write logs of remote nodes or in the dedicated read-cache area of the local and remote caching SSDs of hybrid nodes. Finally, if the data has not been accessed in a significant amount of time, the file system will retrieve the requested data from the distributed capacity layer. As requests for read operations are made to the distributed file system and the data is retrieved from the capacity layer, the caching SSDs of hybrid nodes populate their dedicated read-cache area to speed up subsequent requests for the same data. All-flash configurations, however, do not employ a dedicated read cache because such caching does not provide any performance benefit; the persistent data copy already resides on high-performance SSDs.

## Object Storage and SwiftStack Software

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 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 15 illustrates the differences.

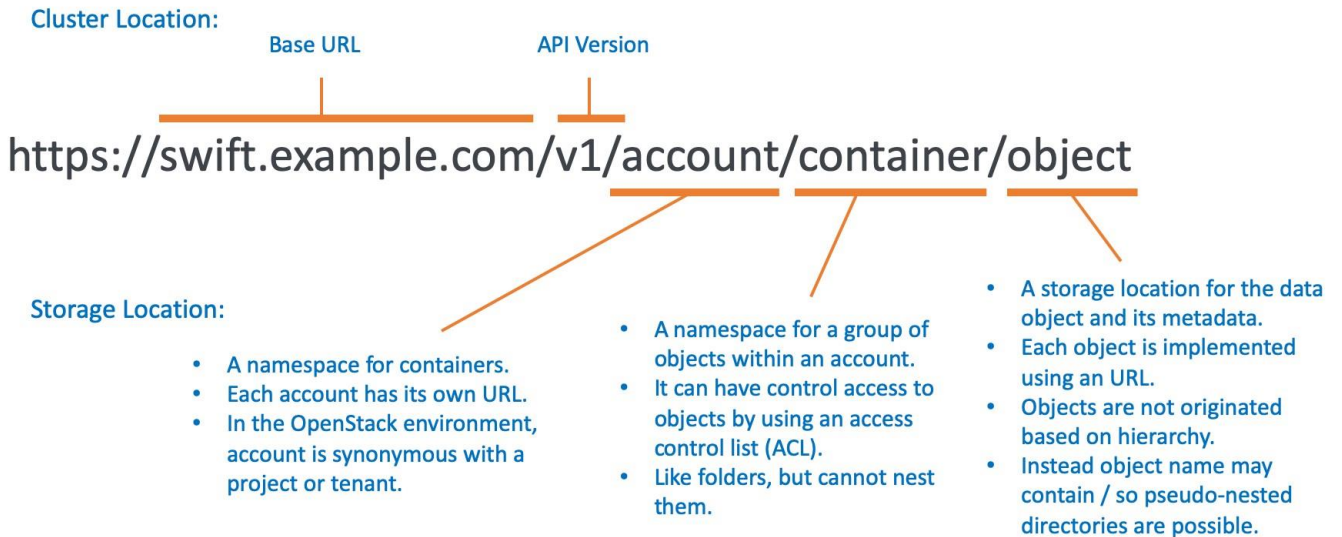
Figure 15 Traditional vs. Object Storage



Object storage supports RESTful / HTTP protocols and every object in a SwiftStack container is accessed through http URL (Figure 16).

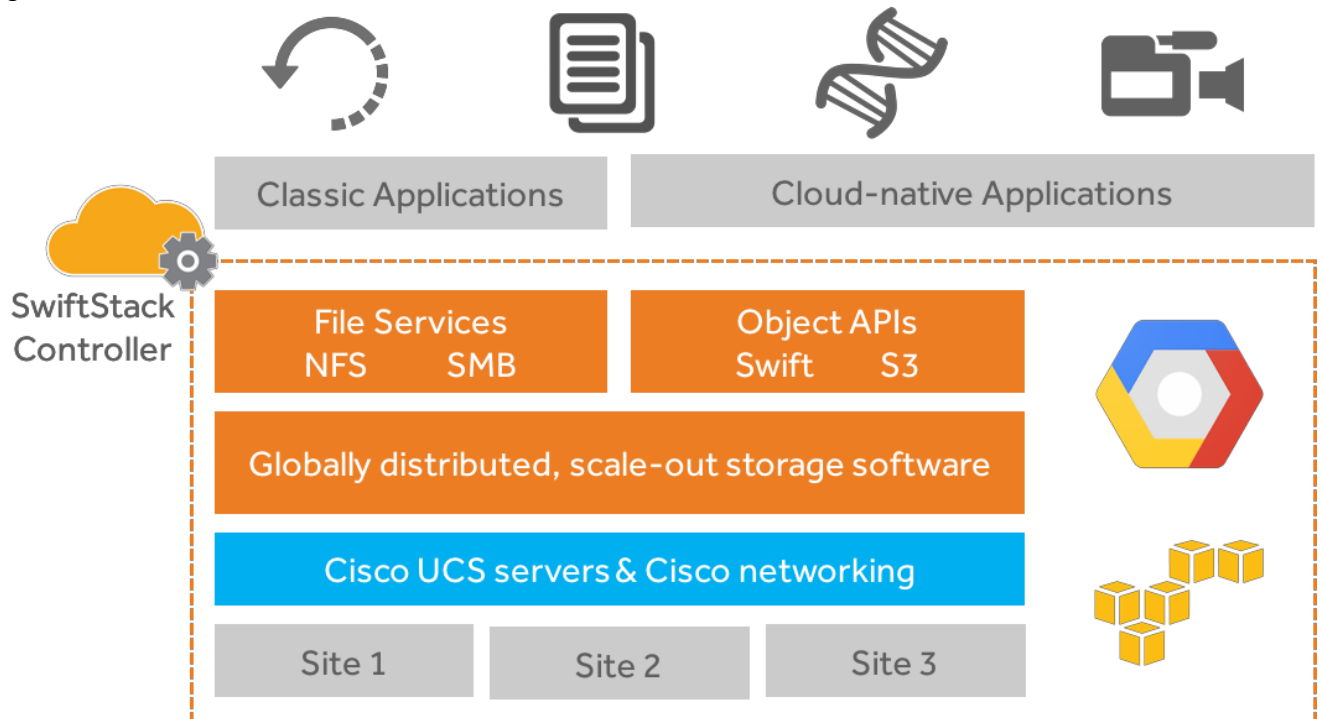
Figure 16 Every Object has an URL

## Every Object Has An URL



With SwiftStack software running on Cisco UCS S3260 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 17 SwiftStack Introduction



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 for whatever your applications need.

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

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

Key SwiftStack features for an active archive:

- 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 through 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 through 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

## Splunk Enterprise for Big Data Analytics

Splunk Enterprise is a software product that enables you to search, analyze, and visualize the data gathered from the components of your IT infrastructure or business. Splunk Enterprise takes in data from websites, applications, sensors, devices, and so on. After you define the data source, Splunk Enterprise indexes the data stream and parses it into a series of individual events that you can view and search.

All the IT applications, systems and technology infrastructure generate data every millisecond of every day. This machine data is one of the fastest growing, most complex areas of big data. It's also one of the most valuable, containing a definitive record of user transactions, customer behavior, sensor activity, machine behavior, security threats, fraudulent activity and more.

Splunk Enterprise provides a holistic way to organize and extract real-time insights from massive amounts of machine data from virtually any source. This includes data from websites, business applications, social media platforms, app servers,



hypervisors, sensors, traditional databases and open source data stores. Splunk Enterprise scales to collect and index tens of terabytes of data per day, cross multi-geography, multi-data center and hybrid cloud infrastructures.

## Key Features of Splunk Enterprise

Splunk Enterprise provides the end-to-end, real-time solution for machine data delivering the following core capabilities:

- Universal collection and indexing of machine data, from virtually any source
- Powerful search processing language (SPL) to search and analyze real-time and historical data
- Real-time monitoring for patterns and thresholds; real-time alerts when specific conditions arise
- Powerful reporting and analysis
- Custom dashboards and views for different roles
- Resilience and scale on commodity hardware
- Granular role-based security and access controls
- Support for multi-tenancy and flexible, distributed deployments on-premises or in the cloud
- Robust, flexible platform for big data apps

## Splunk Enterprise Processing Components

Table 1 lists the three major components processing Splunk data and the tiers that they occupy. It also describes the functions that each component performs.

Table 1 Splunk Enterprise Processing Components

Component	Tier	Description
Forwarder	Data input	A forwarder consumes data and then forwards the data onwards, usually to an indexer. Forwarders usually require minimal resources, allowing them to reside lightly on the machine generating the data.
Indexer	Indexing	<p>An indexer indexes incoming data that it usually receives from a group of forwarders. The indexer transforms the data into events and stores the events in an index. The indexer also searches the indexed data in response to search requests from a search head.</p> <p>To ensure high data availability and protect against data loss, or just to simplify the management of multiple indexers, you can deploy multiple indexers in indexer clusters.</p>
Search head	Search management	<p>A search head interacts with users, directs search requests to a set of indexers, and merges the results back to the user.</p> <p>To ensure high availability and simplify horizontal scaling, you can deploy multiple search heads in search head clusters.</p>



**Note:** Forwarder is only used for testing the deployment server. We generate and send the data directly to the indexers.

## SmartStore

Splunk SmartStore is a new feature that enables indexer capabilities to optimize the use of local storage and allows the system to use remote object stores, such as Amazon S3, to store indexed data. SmartStore introduces a remote storage tier and a cache manager. This feature allows data to reside either locally on indexers or on the remote storage tier. Most data resides on remote storage, while the indexer maintains a local cache that contains a minimal amount of data and metadata. SmartStore also allows the users to manage the indexer storage and compute resources in a cost-effective manner by scaling computing and storage resources separately. The users can reduce the indexer storage footprint to a minimum and choose I/O optimized compute resources.

SmartStore Indexes handle buckets differently from non-SmartStore indexes. Indexers maintain buckets for SmartStore indexes in two states:

- Hot buckets
- Warm buckets

The hot buckets reside on local storage while warm buckets reside on remote storage, although copies of those buckets might also reside temporarily in local storage. The concept of cold buckets becomes optional because the need to distinguish between warm and cold buckets no longer exists. With non-SmartStore indexes, the cold bucket state exists as a way to identify older buckets that can be safely moved to some type of cheaper storage, because buckets are typically searched less frequently as they age. But with SmartStore indexes, warm buckets are already on inexpensive storage, so there is no reason to move them to another type of storage as they age. Buckets can roll to frozen directly from warm.

When a bucket in a SmartStore index rolls to warm, the bucket is copied to remote storage. The rolled bucket does not immediately get removed from the indexer's local storage. Rather, it remains cached locally until it is evicted in response to the cache manager's eviction policy. Because searches tend to occur most frequently across recent data, this process helps to minimize the number of buckets that need to be retrieved from remote storage to fulfill a search request. After the cache manager finally does remove the bucket from the indexer's local storage, the indexer still retains metadata information. In addition, the indexer retains an empty directory for the bucket.

In the case of an indexer cluster, when a bucket rolls to warm, the source peer uploads the bucket to remote storage. The source peer continues to retain its bucket copy in local cache until, in due course, the cache manager evicts it. After successfully uploading the bucket, the source peer sends messages to the bucket's target peers, notifying them that the bucket has been uploaded to remote storage. The target peers then evict their local copies of the bucket, so that the cluster, as a whole, caches only a single copy of the rolled bucket, in the source peer's local storage. The target peers retain metadata for the bucket, so that the cluster has enough copies of the bucket, in the form of its metadata, to match the replication factor. When the source peer's copy of the bucket eventually gets evicted, the source peer, too, retains the bucket metadata. In addition to retaining metadata information for the bucket, the source peer continues to retain the primary designation for the bucket. The peer with primary designation fetches the bucket from remote storage when the bucket is needed for a search.

SmartStore offers several advantages to the deployment's indexing clusters:

- Reduced storage cost. The deployment can take advantage of the economy of remote object stores, instead of relying on costly local storage.
- Access to high availability and data resiliency features available through remote object stores.
- The ability to scale compute and storage resources separately, thus ensuring that you use resources efficiently.
- Simple and flexible configuration with per-index settings.
- Fast recovery from peer failure and fast data rebalancing, requiring only metadata fixups for warm data.
- Lower overall storage requirements, as the system maintains only a single permanent copy of each warm bucket on the remote object stores.

- Full recovery of warm buckets even when the number of peer nodes that goes down is greater than or equal to the replication factor.
- Global size-based data retention.

While SmartStore-enabled indexes can significantly decrease storage and management costs under the right circumstances, there are still times when the users might find it preferable to rely on local storage. The following circumstances are situations where you may consider enabling SmartStore:

- As the amount of data in local storage continues to grow. While local storage costs might not be a significant issue for a small deployment, you should reconsider your use of local storage as your deployment scales over time.
- If you are using indexer clusters to take advantage of features such as data recovery and disaster recovery. Through SmartStore, you can achieve these aims through the native capabilities of the remote store, without the need to store large amounts of redundant data on local storage.
- If you are using indexer clusters and you find that considerable amounts of your time and your computing resources are devoted to managing the cluster. Through SmartStore, you can eliminate much of the cluster management overhead. In particular, you can greatly reduce the scale of time-consuming activities such as offlining peer nodes, data rebalancing, and bucket fixup, because most of the data no longer resides on the peer nodes.
- When most searches are over recent data.

## Solution Design

### Requirements

The following sections detail the physical hardware, software revisions, and firmware versions required for the solution, which deploys Splunk Enterprise on the Cisco HyperFlex platform, and implements the SmartStore feature by storing warm data in remote SwiftStack object stores on Cisco UCS S3260 storage servers.

### Splunk Enterprise in Virtual Environments

#### Splunk Architecture

Splunk software comes packaged as an 'all-in-one' distribution. The single host can be configured to function as one of or all of the types of Splunk Enterprise components. In a distributed deployment, you can split different types across multiple specialized instances of Splunk Enterprise. These instances can range in number from just a few to many thousands, depending on the quantity of data that you are dealing with and other variables in your environment.

Figure 18 Splunk Components



There are several types of Splunk Enterprise components. Each component handles one or more Splunk Enterprise roles, such as data input or indexing. They fall into two broad categories:

- Processing components: These components handle the data.
- Management components: These components support the activities of the processing components.

These are the available processing component types:

- Search Head: In a distributed search environment, a Splunk Enterprise instance that handles search management functions, directing search requests to a set of search peers and then merging the results back to the user. A Splunk Enterprise instance can function as both a search head and a search peer. If it does only search (and not any indexing), it is usually referred to as a dedicated search head. Search head clusters are groups of search heads that coordinate their activities.
- Indexer: A Splunk Enterprise instance that indexes data, transforming raw data into events, places the results into an index, and searches the indexed data in response to search requests.

The indexer also frequently performs the other fundamental Splunk Enterprise functions: data input and search management. In larger deployments, forwarders handle data input, and forward the data to the indexer for indexing. Similarly, although indexers always perform searches across their own data, in larger deployments, a specialized Splunk Enterprise instance, called a search head, handles search management and coordinates searches across multiple indexers.

- **Universal Forwarder:** Forwarders ingest data. There are a few types of forwarders, but the universal forwarder is the right choice for most purposes. A small-footprint version of a forwarder, it uses a lightweight version of Splunk Enterprise that simply inputs data, performs minimal processing on the data, and then forwards the data to a Splunk indexer or a third-party system.

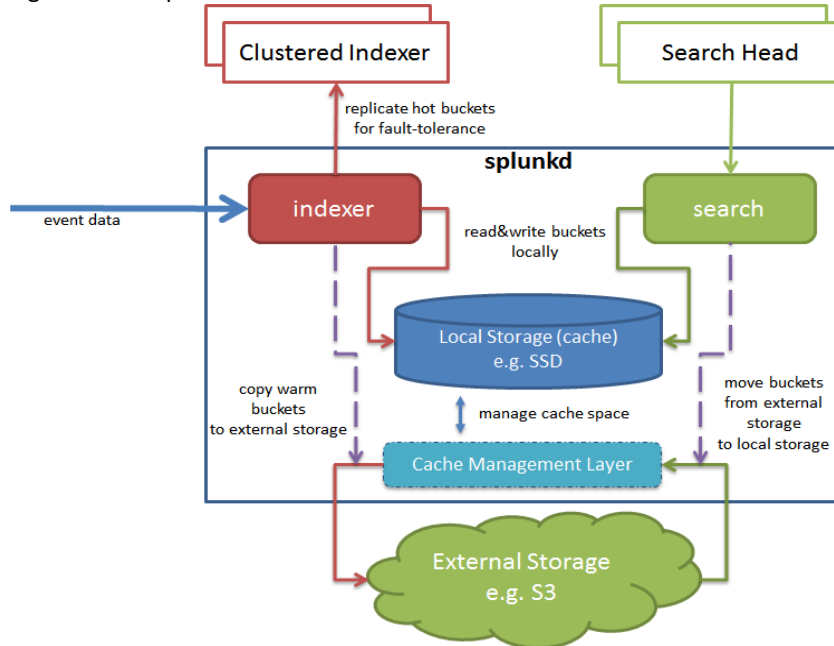
Management components include:

- **Cluster Master (Master Node):** The indexer cluster node that regulates the functioning of an indexer cluster.
- **Deployment Server:** A Splunk Enterprise instance that acts as a centralized configuration manager, grouping together and collectively managing any number of Splunk Enterprise instances. Instances that are remotely configured by deployment servers are called deployment clients. The deployment server downloads updated content, such as configuration files and apps, to deployment clients. Units of such content are known as deployment apps.
- **Search Head Deployer (SHD):** A Splunk Enterprise instance that distributes apps and certain other configuration updates to search head cluster members.
- **License Master:** A license master controls one or more license slaves. From the license master, you can define stacks, pools, add licensing capacity, and manage license.
- **Distributed Monitoring Console (not pictured):** The Distributed Monitoring Console lets you view detailed deployment and performance information about all your Splunk instances in one place. In a single indexer cluster, usually the distributed monitoring console is hosted on the instance running the master node unless the load on the master node is too heavy. In that case you can run the distributed monitoring console on a search head node that is dedicated to running monitoring console.

In this Distributed Configuration, indexers and search heads are configured in a clustered mode. Splunk enterprise supports clustering for both search heads and indexers. Search head cluster is a group of interchangeable and highly available search heads. By increasing concurrent user capacity and by eliminating single point of failure, search head clusters reduce the total cost of ownership. Indexer clusters are made up of groups of Splunk Enterprise indexers configured to replicate peer data so that the indexes of the system become highly available. By maintaining multiple, identical copies of indexes, clusters prevent data loss while promoting data availability for searching.

For the SmartStore feature, a remote S3 compliant storage volume is configured to store the master copies of warm buckets, while the indexer's local storage is used to cache copies of warm buckets currently participating in a search or that have a high likelihood of participating in a future search. The indexer's cache manager manages the local cache. It fetches copies of warm buckets from remote storage when the buckets are needed for a search. It also evicts buckets from the cache, based on factors such as the bucket's search frequency, its data recency, and various other, configurable criteria. With SmartStore indexes, indexer clusters maintain replication and search factor copies of hot buckets only. The remote storage is responsible for ensuring the high availability, data fidelity, and disaster recovery of the warm buckets. Because the remote storage handles warm bucket high availability, peer nodes replicate only warm bucket metadata, not the buckets themselves. This means that any necessary bucket fixup for SmartStore indexes proceeds much more quickly than it does for non-SmartStore indexes. If a group of peer nodes equaling or exceeding the replication factor goes down, the cluster does not lose any of its SmartStore warm data because copies of all warm buckets reside on the remote store. The flow of data for a SmartStore-enabled index in an indexer cluster is depicted in Figure 19.

Figure 19 Splunk SmartStore Data Flow



For more information, please refer to the [Splunk Documentation](#).

### Splunk Services and Processes

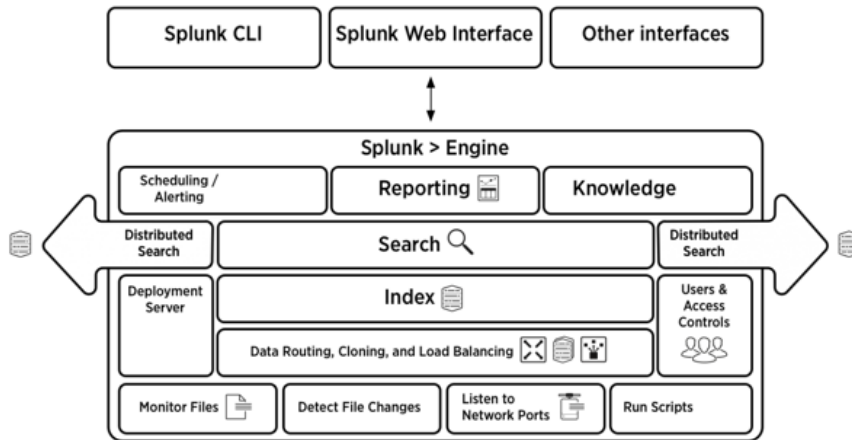
A Splunk Enterprise server installs a process on your host, `splunkd`. `splunkd` is a distributed C/C++ server that accesses, processes and indexes streaming IT data. It also handles search requests. `splunkd` processes and indexes your data by streaming it through a series of pipelines, each made up of a series of processors.

Pipelines are single threads inside the `splunkd` process, each configured with a single snippet of XML. Processors are individual, reusable C or C++ functions that act on the stream of IT data passing through a pipeline. Pipelines can pass data to one another through queues. `splunkd` supports a command-line interface for searching and viewing results.

`splunkd` also provides the Splunk Web user interface. It allows users to search and navigate data stored by Splunk servers and to manage your Splunk deployment through a Web interface. It communicates with your Web browser through Representational State Transfer (REST).

`splunkd` runs administration and management services on port 8089 with SSL/HTTPS turned on by default. It also runs a Web server on port 8000 with SSL/HTTPS turned off by default.

Figure 20 Splunk Services and Processes

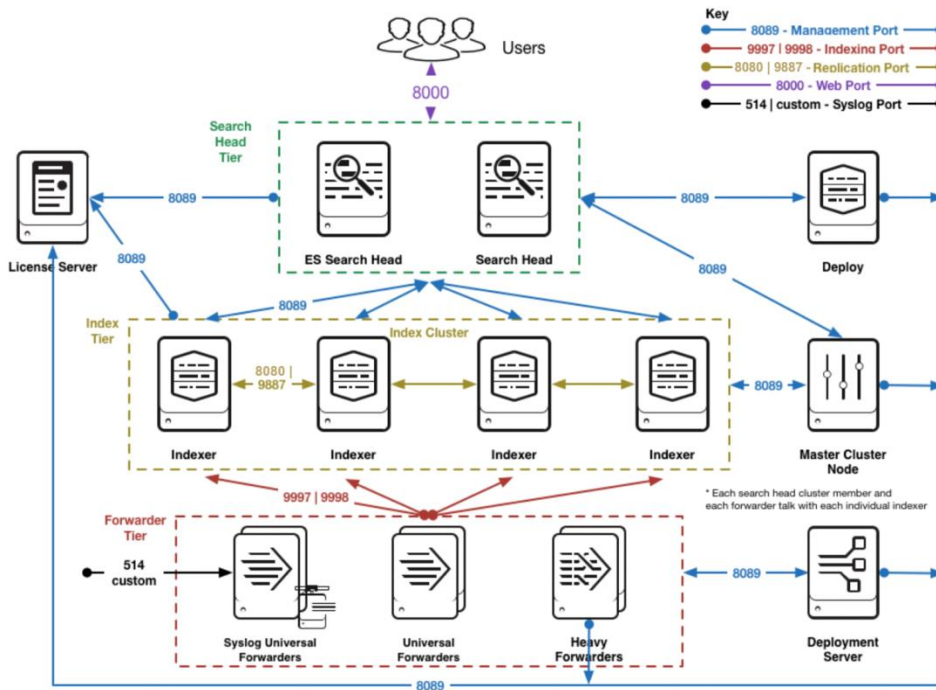


### Splunk Network Ports

Splunk Enterprise components require network connectivity to work properly if they have been distributed across multiple machines. Splunk components communicate with each other using TCP and UDP network protocols. The following ports, but not limited to these ports, must be available to cluster nodes and should be configured as open can to allow communication between the Splunk instances:

- On the master:
  - The management port (by default, 8089) must be available to all other cluster nodes.
  - The http port (by default, 8000) must be available to any browsers accessing the monitoring console.
- On each peer:
  - The management port (by default, 8089) must be available to all other cluster nodes.
  - The replication port (by default, 8080) must be available to all other peer nodes.
  - The receiving port (by default, 9997) must be available to all forwarders sending data to that peer.
- On each search head:
  - The management port (by default, 8089) must be available to all other nodes.
  - The http port (by default, 8000) must be available to any browsers accessing data from the search head.

Figure 21 Splunk Network Ports



### Splunk in Virtual Environments

Splunk Enterprise can be deployed in physical, virtual, cloud or hybrid environments. When deploying Splunk software in a virtualized environment, each type of the components can be run independently inside different virtual machines. The system resources required to run different components vary and should be planned appropriately. The typical components that make up the core of a Splunk deployment include Splunk forwarders, indexers and search heads. Indexers are responsible for storing and retrieving the data from disk, so CPU and disk I/O are the most important considerations. Search heads search for information across indexers and are usually both CPU and memory intensive. Forwarders collect and forward data and they are usually lightweight and are not resource intensive. Therefore, this CVD explains only the topic of the forwarders on how to deploy a universal forwarder from the deployment server. For the data ingestion the load generators send the generated events directly to the indexers without through the forwarders. Other management components include deployment servers and cluster masters are usually lightweight and are not resource intensive either.

The requirements for the Splunk virtual machines that are recommended in the Splunk technical brief for virtual deployment are listed as follows:

- Minimum 12 vCPU
- Minimum 12 GB of RAM
- Full reservations for vCPU and vRAM (no CPU and memory overcommit)
- Minimum 1200 random seek operations per second disk performance (sustained)
- Use VMware Tools in the guest virtual machine
- Use VMware Paravirtual SCSI (PVSCSI) controller
- Use VMXNET3 network adapter



- Provision virtual disks as Eager Zero Thick when not on an array that supports the appropriate VAAI primitives ("Write Same" and "ATS")



**Note:** Splunk recommends Eager Zero Thick provisioning for non-VAAI datastores. Cisco HyperFlex datastores support the integration with the VMware API for Array Integration (VAAI) so that will not have this restriction.

## Splunk Virtual Machines on Cisco HyperFlex

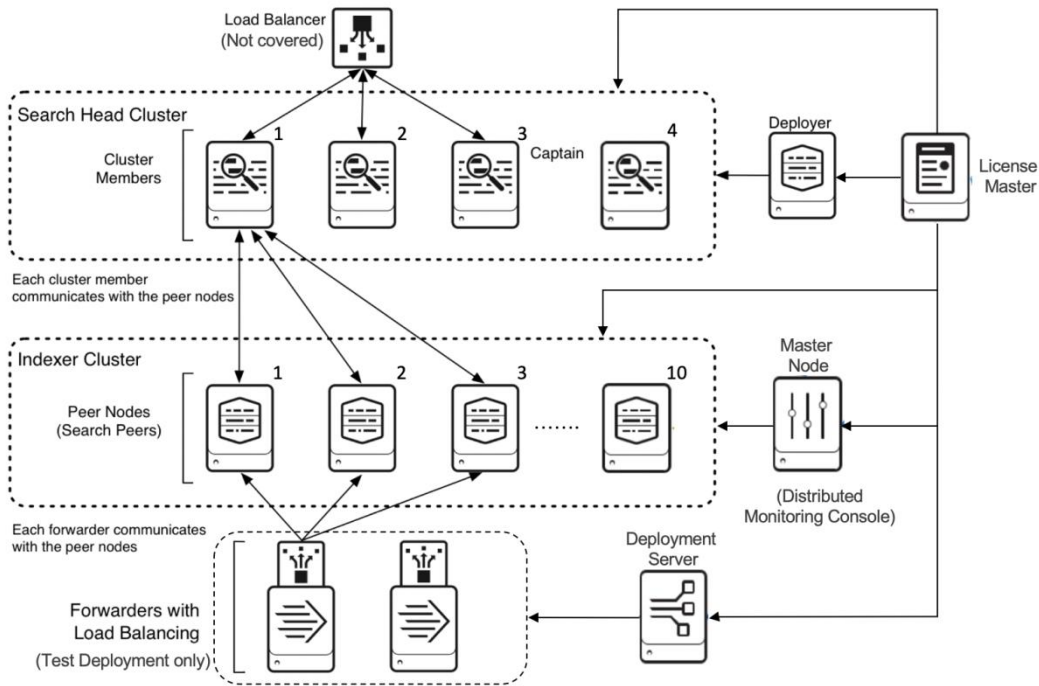
Table 2 lists the virtual machines to install in this solution.

Table 2 Splunk Virtual Machines on HX

Role	Component Type	Number of Virtual Machines	v C P U	Memory	Data Disk	Provision	Network Adapter	Type	SCSI Controller	Type
Indexer	Processing	10	1 2	64GB	600GB	Thick Eager Zeroed	1	VMXN ET3	1	PVSC SI
Search Head	Processing	4	1 2	64GB	48GB	Thin	1	VMXN ET3	1	PVSC SI
Universal Forwarder	Processing	1	4	16GB	48GB	Thin	1	VMXN ET3	1	PVSC SI
Cluster Master (Monitoring Console)	Management	1	4	16GB	48GB	Thin	1	VMXN ET3	1	PVSC SI
License Master	Management	1	4	16GB	48GB	Thin	1	VMXN ET3	1	PVSC SI
SH Deployer	Management	1	4	16GB	48GB	Thin	1	VMXN ET3	1	PVSC SI
Deployment Server	Management	1	4	16GB	48GB	Thin	1	VMXN ET3	1	PVSC SI

Figure 22 illustrates the variable Splunk tiers and components in this solution.

Figure 22 Splunk Virtual Machines on HX

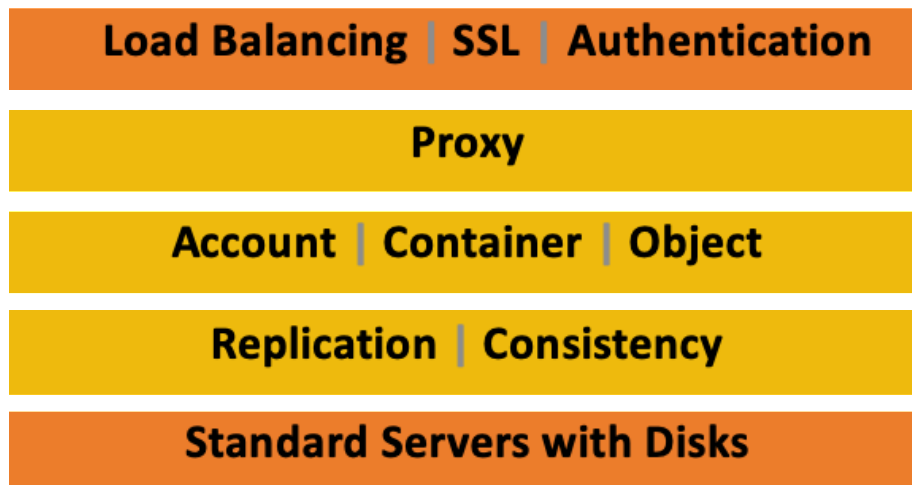


### SwiftStack Object Storage on Cisco UCS S3260

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 SwiftStack 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.

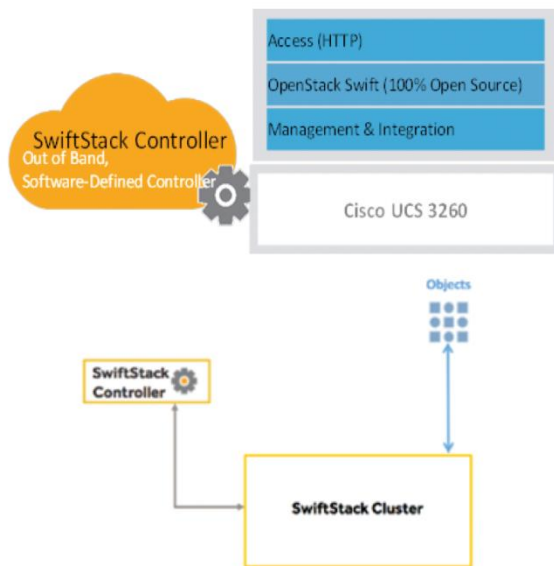
Figure 23 SwiftStack Node Roles



## SwiftStack System Architecture

The SwiftStack solution provides enterprise-grade object storage, with OpenStack Swift at its core. SwiftStack has been deployed at many sites with massive amounts of data stored. SwiftStack is comprised of two major components: SwiftStack Storage Nodes, which store the data, and the SwiftStack Controller Nodes, which are an out-of-band management system that manages one or more SwiftStack storage clusters. In this solution SwiftStack Controller Nodes are built with two Cisco UCS C220 M5 servers in an Active/Standby configuration, and SwiftStack Storage Nodes are built with six Cisco UCS S3260 M5 storage servers. Each Cisco UCS S3260 server is loaded with twenty-seven 12TB HDDs so the whole system provides close to 2PB total storage capacity.

Figure 24 SwiftStack Components



## SwiftStack Network Layout

Network requirements for SwiftStack are based on standard Ethernet connections. While the software can work on a single network interface, it is recommended to configure multiple virtual interfaces in Cisco UCS to segregate the network traffic. Cisco UCS S3260 has two physical ports of 40Gb each and the Cisco VIC allows you to carve out many virtual network interfaces (vNICs) on each physical port.

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

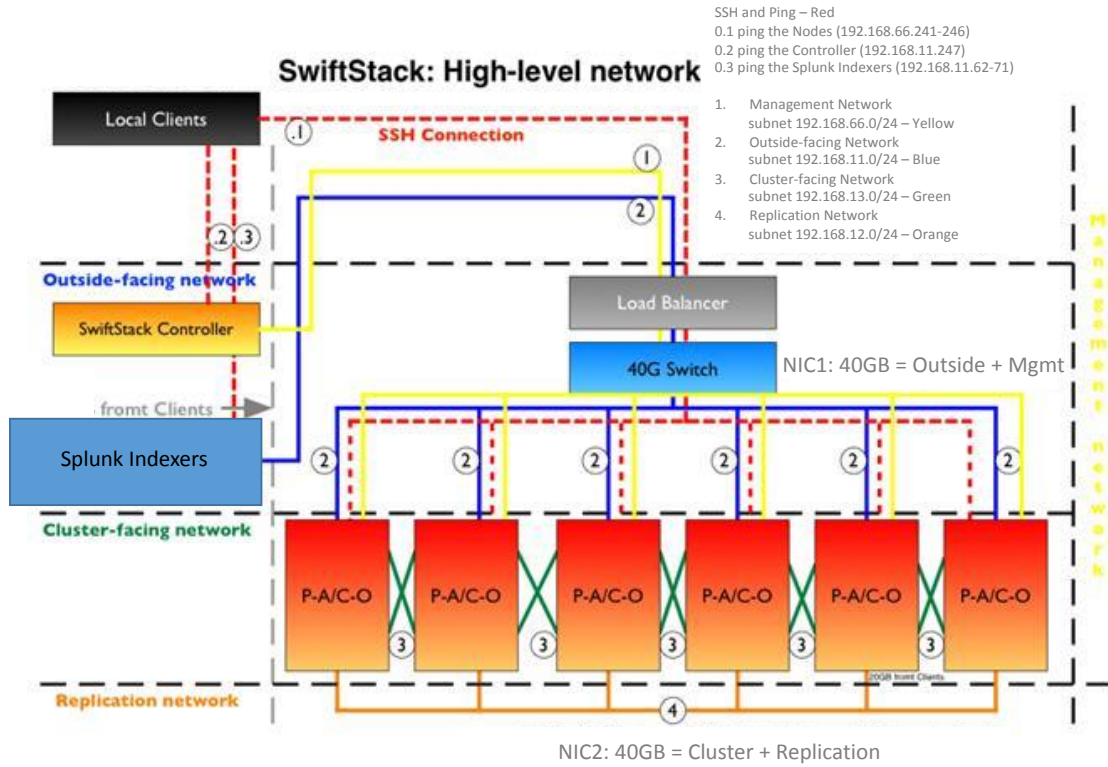
- **Management Network:** All nodes must be able to route IP traffic to a SwiftStack controller. This is the management network for all services.
- **Outward-Facing Network:** This is the front facing network and is used for API access and to run the proxy and authentication services.
- **Cluster-Facing Network:** This is an internal network for communication between the proxy servers and the storage nodes. This is a private network.
- **Replication Network:** This Internal network is for replication among the storage nodes. This is a private network too.
- **Hardware or PXE Management Network (Optional):** The network for Hardware management.

It is recommended to have all cluster internal networks assigned to one physical port, and the outward facing and management networks on the other physical 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.

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. These unique features of Cisco UCS Fabric Interconnects and Cisco VICs make the design highly flexible and scalable.

SwiftStack high-level network topology is depicted in Figure 25.

Figure 25 SwiftStack High-level Network



### Physical Components

Table 3 lists the hardware components required to install a single cluster of the Cisco HyperFlex system. Maximum cluster size of 64 nodes can be obtained by combining 32 converged nodes and 32 compute-only nodes. A 1:2 ratio of HX Converged nodes to Compute only nodes is supported with the HXDP enterprise license.

Table 3 HyperFlex System Components

Component	Hardware Required
Fabric Interconnects	Two Cisco UCS 6248UP Fabric Interconnects, or Two Cisco UCS 6296UP Fabric Interconnects, or Two Cisco UCS 6332 Fabric Interconnects, or Two Cisco UCS 6332-16UP Fabric Interconnects, or Two Cisco UCS 6454 Fabric Interconnects
HX-Series Servers	Three to Thirty-Two Cisco HyperFlex HXAF220c-M5SX All-Flash rack servers, or Three to Thirty-Two Cisco HyperFlex HXAF240c-M5SX All-Flash rack servers, or

Component	Hardware Required
	Three to Thirty-Two Cisco HyperFlex HX220c-M5SX Hybrid rack servers, or Three to Thirty-Two Cisco HyperFlex HX240c-M5SX Hybrid rack servers, or Three to Sixteen Cisco HyperFlex HX240c-M5L Hybrid rack servers, or Three to Thirty-Two Cisco HyperFlex HXAF220c-M4S All-Flash rack servers, or Three to Thirty-Two Cisco HyperFlex HXAF240c-M4SX All-Flash rack servers, or Three to Thirty-Two Cisco HyperFlex HX220c-M4S Hybrid rack servers, or Three to Thirty-Two Cisco HyperFlex HX240c-M4SX Hybrid rack servers
UCS computing-only Servers	Zero to Thirty-Two Cisco UCS B200 M3 Blade Servers, or Zero to Thirty-Two Cisco UCS B200 M4 Blade Servers, or Zero to Thirty-Two Cisco UCS B200 M5 Blade Servers, or Zero to Thirty-Two Cisco UCS B260 M4 Blade Servers, or Zero to Thirty-Two Cisco UCS B420 M4 Blade Servers, or Zero to Thirty-Two Cisco UCS B460 M4 Blade Servers, or Zero to Thirty-Two Cisco UCS B480 M5 Blade Servers, or Zero to Thirty-Two Cisco UCS C220 M3 Rack-Mount Servers, or Zero to Thirty-Two Cisco UCS C220 M4 Rack-Mount Servers, or Zero to Thirty-Two Cisco UCS C220 M5 Rack-Mount Servers, or Zero to Thirty-Two Cisco UCS C240 M3 Rack-Mount Servers, or Zero to Thirty-Two Cisco UCS C240 M4 Rack-Mount Servers, or Zero to Thirty-Two Cisco UCS C240 M5 Rack-Mount Servers, or Zero to Thirty-Two Cisco UCS C460 M4 Rack-Mount Servers, or Zero to Thirty-Two Cisco UCS C480 M5 Rack-Mount Servers

For the complete server specifications and additional information, go to: [Cisco HyperFlex product site](#).

In this solution, the following physical hardware is used to build a scale-out virtual infrastructure for the Splunk SmartStore deployment.

Table 4 Splunk Scale-Out Virtual Infrastructure Components

Component	Hardware Required
Fabric Interconnects	Two (2) Cisco UCS 6332 Fabric Interconnects
HX Series Servers	Four (4) Cisco HXAH240c-M5SX All-Flash rack servers
UCS Computing Nodes	Six (6) Cisco UCS C220 M5 Rack-Mount Servers
SwiftStack Controller Nodes	Two (2) Cisco UCS C220 M5 Rack-Mount Servers

Component	Hardware Required
UCS S3260 Storage Servers	Three (3) Cisco UCS S3260 chassis with (6) Server Blades

For the complete server specifications and additional information, please refer to the links below:

Cisco Fabric Interconnect 6332:

<https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-b-series-blade-servers/6332-specsheet.pdf>

Cisco HXAF240c-M5SX All-Flash rack servers:

<https://www.cisco.com/c/dam/en/us/products/collateral/hyperconverged-infrastructure/hyperflex-hx-series/le-60103-ds-hx240cm5-180416a.pdf>

Cisco UCS C220 M5 Rack-Mount Servers:

<https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/c220m5-sff-specsheet.pdf>

Cisco UCS S3260 Storage Servers:

<https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-s-series-storage-servers/s3260-specsheet.pdf>

**Error! Reference source not found.** Table 5 lists the required hardware components and disk options for the Cisco HXAF240c-M5SX All-Flash rack servers, which are required for creating the HyperFlex cluster and installing the Splunk virtual machines in the cluster:

Table 5 HXAF240c-M5SX Server Options

HXAF240c-M5SX Options	Hardware Required
Processors	Choose a matching pair of Intel Xeon Processor Scalable Family CPUs.  Two (2) Intel Xeon Gold 6132 14-core CPUs are chosen on each server.
Memory	192 GB to 3 TB of total memory using 16 GB, 32 GB, 64 GB, or 128 GB DDR4 2666 MHz 1.2v modules.  Twelve (12) 32 GB RDIMMs are chosen (total 384 GB).
Disk Controller	Cisco 12Gbps Modular SAS HBA
SSD	One 240 GB 2.5 Inch Enterprise Value 6G SATA SSD  One 400 GB 2.5 Inch Enterprise Performance 12G SAS SSD, or one 1.6 TB 2.5 Inch Enterprise Performance NVMe SSD, or one 375 GB 2.5 Inch Optane Extreme Performance SSD  One (1) 1.6 TB NVMe SSD is chosen.  Six to twenty-three 3.8 TB 2.5 Inch Enterprise Value 6G SATA SSDs, or six to twenty-three 960 GB 2.5 Inch Enterprise Value 6G SATA SSDs  Twenty-three (23) 960 GB SSDs are chosen.
Network	Cisco UCS VIC1387 VIC MLOM
Boot Device	One 240 GB M.2 form factor SATA SSD

<b>HXAF24oc-M5SX Options</b>	<b>Hardware Required</b>
microSD Card	One 32GB microSD card for local host utilities storage

Table 6 lists the required hardware components and disk options for the Cisco UCS C220 M5 rack servers, which are used for expanding the HyperFlex cluster, and installing the SwiftStack Controller software:

Table 6 Cisco UCS C220 M5 Server Options

<b>UCSC-C220-M5SX Options</b>	<b>Hardware Required</b>
Processors	Choose a matching pair of Intel Xeon Processor Scalable Family CPUs.  Two (2) Intel Xeon Gold 6132 14-core CPUs are chosen on each server.
Memory	192 GB to 3 TB of total memory using 16 GB, 32 GB, 64 GB, or 128 GB DDR4 2666 MHz 1.2v modules.  Twelve (12) 32 GB RDIMMs are chosen (total 384 GB).
Disk Controller	Cisco 12Gbps Modular SAS HBA
SSD	One (1) 1.6 TB 2.5 inch Enterprise Value 6G SATA SSD (SwiftStack Controllers only)
Network	Cisco UCS VIC1387 VIC MLOM
Boot Device	One 240 GB M.2 form factor SATA SSD

Table 7 lists the required hardware components and disk options for the Cisco UCS S3260 M5 storage servers, which are used for creating the SwiftStack object storage system:

Table 7 Cisco UCS S3260 M5 Server Options

<b>UCS-S3260-M5SRB Options</b>	<b>Hardware Required</b>
Processors	Choose a matching pair of Intel Xeon Processor Scalable Family CPUs.  Two (2) Intel Xeon Gold 6132 14-core CPUs are chosen on each server.
Memory	192 GB to 3 TB of total memory using 16 GB, 32 GB, 64 GB, or 128 GB DDR4 2666 MHz 1.2v modules.  Eight (8) 32 GB RDIMMs are chosen (total 256 GB).
Disk Controller	Cisco UCS S3260 Dual Raid Controller
Disks	Twenty-seven (27) 12TB HDD's and One (1) 400GB SSD per server node.
Network	Cisco UCS S3260 System IO Controller with VIC 1380 included

## Software Components

Table 8 lists the software components and the versions as tested and validated in this document:

Table 8 Software Components

<b>Component</b>	<b>Software Version</b>
Splunk Enterprise	7.2.3
Cisco HyperFlex Data Platform Software	3.5(2a)

Cisco UCS Firmware	4.0(1c)
SwiftStack Controller Software	6.19.1.3.0.0.2
OpenStack Swift version	2.20.0.2-1.el7
Red Hat Enterprise Linux Server	7.5
VMWare vSphere ESXi Hypervisor	6.5.0 Build 10884925
VMWare vSphere vCenter Appliance	6.5.0, Build 11347054

## Licensing

Cisco UCS and HyperFlex systems, Splunk Enterprise software and the SwiftStack software must be properly licensed for all the features in use and for all ports in use on the Cisco UCS Fabric Interconnects. HXDP enterprise license is required to the support 1:2 ratio of HX to Compute-only nodes. Contact your resale partner or your Cisco, Splunk, and SwiftStack sales teams to ensure you order all of the necessary and appropriate licenses for your solution.

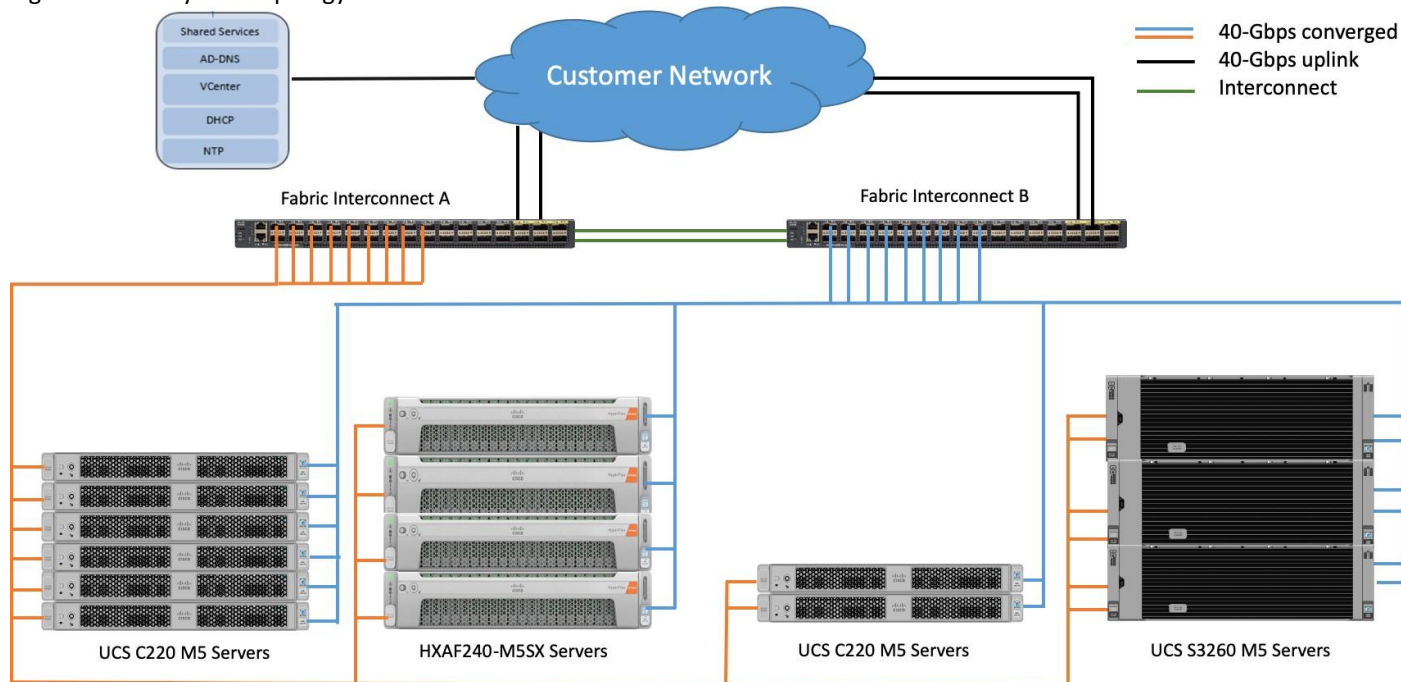
## Physical Topology

### Topology Overview

The Cisco Unified Computing System is composed of a pair of Cisco UCS Fabric Interconnects along with up to 160 Cisco UCS B-Series blade servers, Cisco UCS C-Series rack-mount servers, HX-Series hyperconverged servers, or Cisco UCS S-Series storage servers per UCS domain. Inside of a Cisco UCS domain, multiple environments can be deployed for differing workloads. For example, a Cisco HyperFlex cluster can be built using Cisco HX-Series rack-mount servers, Cisco UCS C-Series rack mount servers can be used for expanding the cluster, and a SwiftStack cluster can be built using Cisco UCS S3260 high density storage servers. The two Fabric Interconnects both connect to every Cisco UCS C-Series, Cisco HX-Series, or Cisco UCS S-Series rack-mount server. Upstream network connections, also referred to as “northbound” network connections are made from the Fabric Interconnects to the customer data center network at the time of installation.



Figure 26 Physical Topology



## Fabric Interconnects

Fabric Interconnects (FI) are deployed in pairs, wherein the two units operate as a management cluster, while forming two separate network fabrics, referred to as the A side and B side fabrics. Therefore, many design elements will refer to FI A or FI B, alternatively called fabric A or fabric B. Both Fabric Interconnects are active at all times, passing data on both network fabrics for a redundant and highly available configuration. Management services, including Cisco UCS Manager, are also provided by the two FIs but in a clustered manner, where one FI is the primary, and one is secondary, with a roaming clustered IP address. This primary/secondary relationship is only for the management cluster and has no effect on data transmission.

Fabric Interconnects have the following ports, which must be connected for proper management of the Cisco UCS domain:

- **Mgmt:** A 10/100/1000 Mbps port for managing the Fabric Interconnect and the Cisco UCS domain with GUI and CLI tools. This port is also used by remote KVM, IPMI and SoL sessions to the managed servers within the domain. This is typically connected to the customer management network.
- **L1:** A cross connect port for forming the Cisco UCS management cluster. This port is connected directly to the L1 port of the paired Fabric Interconnect using a standard CAT5 or CAT6 Ethernet cable with RJ45 plugs. It is not necessary to connect this to a switch or hub.
- **L2:** A cross connect port for forming the Cisco UCS management cluster. This port is connected directly to the L2 port of the paired Fabric Interconnect using a standard CAT5 or CAT6 Ethernet cable with RJ45 plugs. It is not necessary to connect this to a switch or hub.
- **Console:** An RJ45 serial port for direct console access to the Fabric Interconnect. This port is typically used during the initial FI setup process with the included serial to RJ45 adapter cable. This can also be plugged into a terminal aggregator or remote console server device.

## Cisco HXAF240c-M5SX Servers Connectivity

Four Cisco HyperFlex 240c M5 All Flash servers are used in this solution to create the HyperFlex cluster as converged nodes. The HX-Series converged servers are connected directly to the Cisco UCS Fabric Interconnects in Direct Connect mode. This

option enables Cisco UCS Manager to manage the HX-Series Rack-Mount Servers using a single cable for both management traffic and data traffic. All the Cisco HyperFlex M5 generation servers can be configured with the Cisco VIC 1387 network interface card (NIC) installed in a modular LAN on motherboard (MLOM) slot, which has dual 40 Gigabit Ethernet (GbE) ports. The standard and redundant connection practice is to connect port 1 of the VIC card (the right-hand port) to a port on FI A, and port 2 of the VIC card (the left-hand port) to a port on FI B (Figure 27). The HyperFlex installer checks for this configuration, and that all servers' cabling matches. Failure to follow this cabling practice can lead to errors, discovery failures, and loss of redundant connectivity.

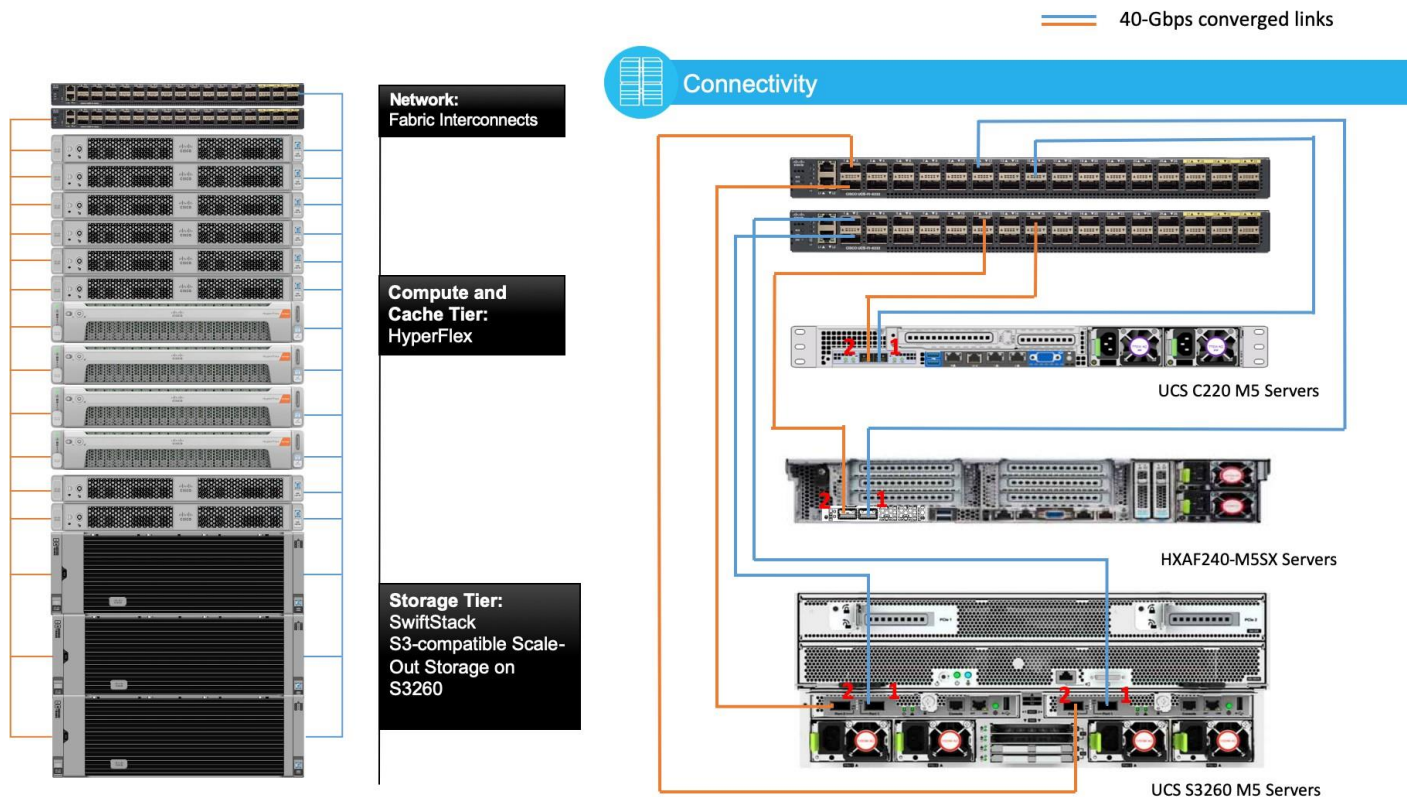
### Cisco UCS C220 M5 Servers Connectivity

Eight Cisco UCS C220 M5 servers are used in this solution. Six of them are used for expanding the HyperFlex cluster as computing-only nodes and two of them are used as the controller nodes for the SwiftStack cluster, where one is active and another is standby. The Cisco UCS C-Series Rack-Mount Servers are connected directly to the Cisco UCS Fabric Interconnects in Direct Connect mode. Internally the Cisco UCS C-Series servers are configured with the Cisco VIC 1387 network interface card (NIC) installed in a modular LAN on motherboard (MLOM) slot, which has two 40 Gigabit Ethernet (GbE) ports. The standard and redundant connection practice is to connect port 1 of the VIC card (the right-hand port) to a port on FI A, and port 2 of the VIC card (the port located on the left) to a port on FI B (Figure 27).

### Cisco UCS S3260 Servers Connectivity

Three dual-node UCS S3260 chassis are used in this solution to create the SwiftStack cluster as SwiftStack PACO nodes. Each UCS S3260 chassis has two S3260 M5 servers and two System IO Controllers (SIOC), each with a Cisco VIC 1300 Series dual-port 40Gbps network interface card (NIC). The Cisco UCS S3260 servers are connected to the Cisco UCS Fabric Interconnects through the System IO Controllers (SIOC). This option enables Cisco UCS Manager to manage the Cisco UCS S3260 servers using a single cable for both management traffic and data traffic. All the Cisco UCS S3260 M5 generation servers can be configured with the embedded Cisco VIC 1300 network interface card (NIC) included on the SIOC module which has dual 40 Gigabit Ethernet (GbE) ports. The standard and redundant connection practice is to connect port 1 of the VIC card to a port on FI A, and port 2 of the VIC card to a port on FI B (Figure 27).

Figure 27 Server Connectivity



## Cisco UCS Uplink Connectivity

Cisco UCS network uplinks connect “northbound” from the pair of Cisco UCS Fabric Interconnects to the LAN in the customer data center. All Cisco UCS uplinks operate as trunks, carrying multiple 802.1Q VLAN IDs across the uplinks. The default Cisco UCS behavior is to assume that all VLAN IDs defined in the Cisco UCS configuration are eligible to be trunked across all available uplinks.

Cisco UCS Fabric Interconnects appear on the network as a collection of endpoints versus another network switch. Internally, the Fabric Interconnects do not participate in spanning-tree protocol (STP) domains, and the Fabric Interconnects cannot form a network loop, as they are not connected to each other with a layer 2 Ethernet link. All link up/down decisions through STP will be made by the upstream root bridges.

Uplinks need to be connected and active from both Fabric Interconnects. For redundancy, multiple uplinks can be used on each FI, either as 802.3ad Link Aggregation Control Protocol (LACP) port-channels or using individual links. For the best level of performance and redundancy, uplinks can be made as LACP port-channels to multiple upstream Cisco switches using the virtual port channel (vPC) feature. Using vPC uplinks allows all uplinks to be active passing data, plus protects against any individual link failure, and the failure of an upstream switch. Other uplink configurations can be redundant but spanning-tree protocol loop avoidance may disable links if vPC is not available.

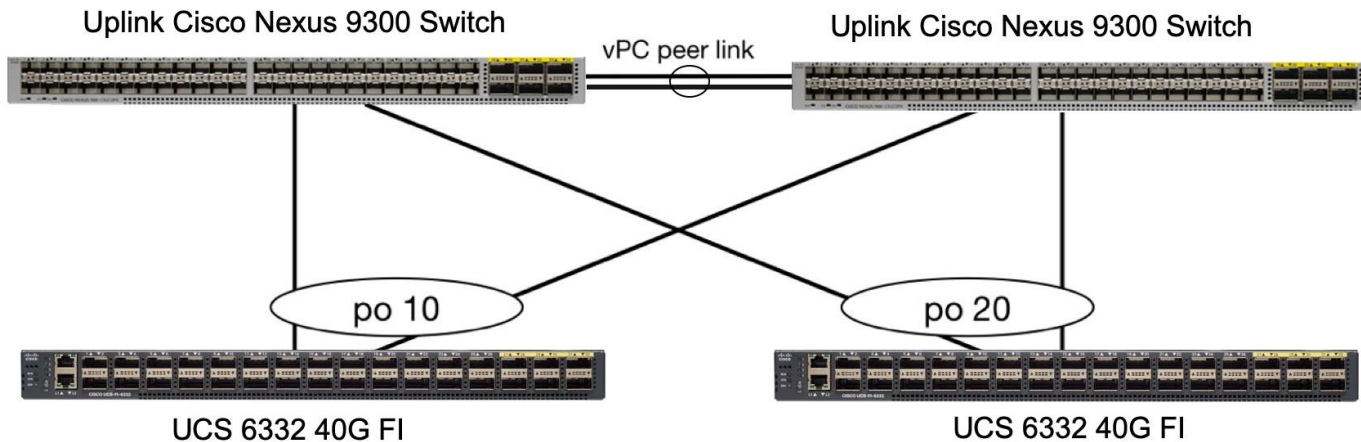
All uplink connectivity methods must allow for traffic to pass from one Fabric Interconnect to the other, or from fabric A to fabric B. There are scenarios where cable, port or link failures would require traffic that normally does not leave the Cisco UCS domain, to now be forced over the Cisco UCS uplinks. Additionally, this traffic flow pattern can be seen briefly during maintenance procedures, such as updating firmware on the Fabric Interconnects, which requires them to be rebooted.

## vPC to Cisco Nexus Switches

This recommended connection design relies on using Cisco switches that have the virtual port channel feature, such as Catalyst 6000 series switches running VSS, Cisco Nexus 5000 series, and Cisco Nexus 9000 series switches. Logically the

two vPC enabled switches appear as one, and therefore spanning-tree protocol will not block any links. This configuration allows for all links to be active, achieving maximum bandwidth potential, and multiple redundancy at each level. In the setup to validate this solution in the lab, the two Cisco UCS 6332 40G fabric interconnects are connected to a pair of Cisco Nexus 9300 series data center switches with vPC.

Figure 28 Uplink Connectivity with vPC



## Logical Topology

### Logical Network Design

The Cisco HyperFlex system has communication pathways that fall into four defined zones:

- **Management Zone:** This zone comprises the connections needed to manage the physical hardware, the hypervisor hosts, and the storage platform controller virtual machines (SCVM). These interfaces and IP addresses need to be available to all staff who will administer the HX system, throughout the LAN/WAN. This zone must provide access to Domain Name System (DNS) and Network Time Protocol (NTP) services and allow Secure Shell (SSH) communication. In this zone are multiple physical and virtual components:
  - Fabric Interconnect management ports.
  - Cisco UCS external management interfaces used by the servers and blades, which answer through the FI management ports.
  - ESXi host management interfaces.
  - Storage Controller Virtual Machine management interfaces.
  - A roaming HX cluster management interface.
  - Storage Controller Virtual Machine replication interfaces.
  - A roaming HX cluster replication interface.
- **VM Zone:** This zone comprises the connections needed to service network IO to the guest virtual machines that will run inside the HyperFlex hyperconverged system. This zone typically contains multiple VLANs, that are trunked to the Cisco UCS Fabric Interconnects through the network uplinks, and tagged with 802.1Q VLAN IDs. These interfaces and IP addresses need to be available to all staff and other computer endpoints which need to

communicate with the guest virtual machines in the HX system, throughout the LAN/WAN. Splunk virtual machines fit into this zone.

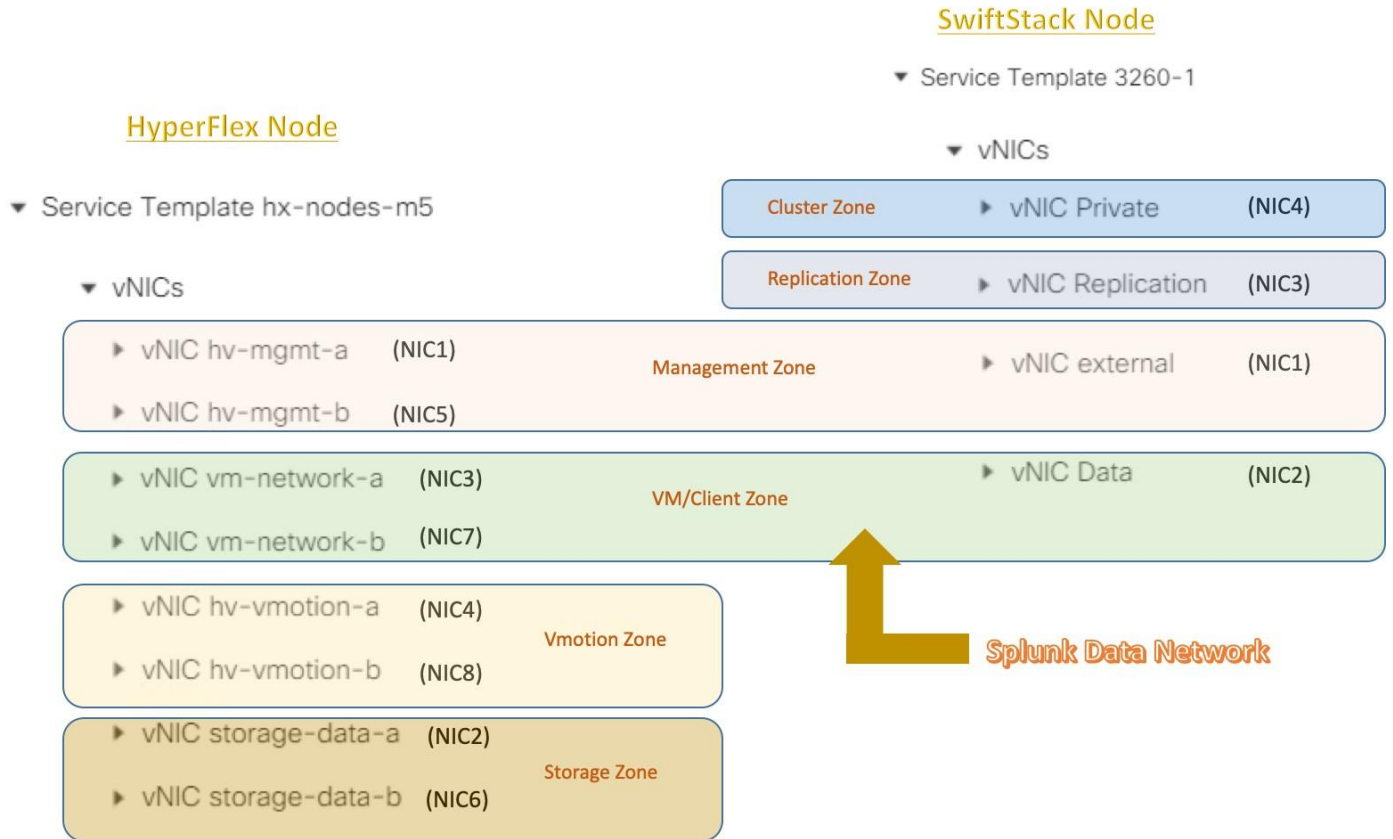
- **Storage Zone:** This zone comprises the connections used by the Cisco HX Data Platform software, ESXi hosts, and the storage controller virtual machines to service the HX Distributed Data Filesystem. These interfaces and IP addresses need to be able to communicate with each other at all times for proper operation. During normal operation, this traffic all occurs within the Cisco UCS domain, however there are hardware failure scenarios where this traffic would need to traverse the network northbound of the Cisco UCS domain. For that reason, the VLAN used for HX storage traffic must be able to traverse the network uplinks from the Cisco UCS domain, reaching FI A from FI B, and vice-versa. This zone is primarily jumbo frame traffic therefore jumbo frames must be enabled on the Cisco UCS uplinks. In this zone are multiple components:
  - A VMkernel interface used for storage traffic on each ESXi host in the HX cluster.
  - Storage Controller Virtual Machine storage interfaces.
  - A roaming HX cluster storage interface.
- **VMotion Zone:** This zone comprises the connections used by the ESXi hosts to enable vMotion of the guest virtual machines from host to host. During normal operation, this traffic all occurs within the Cisco UCS domain, however there are hardware failure scenarios where this traffic would need to traverse the network northbound of the Cisco UCS domain. For that reason, the VLAN used for HX vMotion traffic must be able to traverse the network uplinks from the Cisco UCS domain, reaching FI A from FI B, and vice-versa.

The SwiftStack cluster is created with four networks that can fall into four zones similarly:

- **Management Zone:** This zone has the management network for all services.
- **Application Zone:** This zone has the outward-facing network and is used for API access and to run the proxy and authentication services. This network provides the access to the clients.
- **Cluster-facing Zone:** This zone manages the internal communication between the proxy servers and the storage nodes. This zone includes the cluster-facing private network.
- **Replication Zone:** This zone has the Internal network for replication among the storage nodes. This is a private network too.

Splunk virtual machines fit into the application zone and the indexers should be able to have the access to the SwiftStack S<sub>3</sub> containers through the network interfaces on the SwiftStack cluster defined for this client-facing zone.

Figure 29 Logical Network Design



VLANS and Subnets

For the Splunk network configuration, only one VLAN is needed to be carried to the Cisco UCS domain from the upstream LAN, and this VLAN is also defined in the Cisco HX configurations. Table 9 lists the VLANs configured for Splunk virtual machines in Cisco HX node, and their functions.

Table 9 Splunk VLANs

VLAN Name	VLAN ID	Purpose
<<SPK-DATA-11>>	Customer supplied <<11>>	Splunk node Linux OS interfaces

The following network design has been applied on each HyperFlex node including the computing-only node. Each server node has two 40Gbps ports and has five VLANs as listed in Table 10.

Table 10 HX Network Design

Interface	Purpose	Physical Port on Adapter	Network Capacity	VLAN	VLAN Name	Network
NIC1	Management interfaces (a)	Port 1	40Gbps	3021	hx-inband-mgmt	192.168.66.0/24
NIC2	Storage network interfaces (a)			3022	hx-storage-data	192.168.77.0/24
NIC3	Virtual Machine network interfaces (a)			11	SPK-Data-11	192.168.11.0/24
				3024	vm-network	NA

Interface	Purpose	Physical Port on Adapter	Network Capacity	VLAN	VLAN Name	Network
NIC4	VMotion interfaces (a)			3023	hx-vmotion	192.168.88.0/24
NIC5	Management interfaces (b)	Port 2	40Gbps	3021	hx-inband-mgmt	192.168.66.0/24
NIC6	Storage network interfaces (b)			3022	hx-storage-data	192.168.77.0/24
NIC7	Virtual Machine network interfaces (b)			11	SPK-Data-11	192.168.11.0/24
				3024	vm-network	NA
NIC8	VMotion interfaces (b)			3023	hx-vmotion	192.168.88.0/24

The following network design has been applied on each S3260 server node. Each server node has two 40Gbps ports and has four VLANs as listed in Table 11.

Table 11 SwiftStack Network Design

Interface	Purpose	Physical Port on Adapter	Network Capacity	VLAN	VLAN Name	Network
NIC1	External/Management interface	Port 1	40Gbps	3021	SwiftStack3021	192.168.66.0/24
NIC2	Client/Outward facing interface	Port 1		11	SPK-Data-11	192.168.11.0/24
NIC3	Cluster network interface	Port 2	40Gbps	13	SwiftPrivate	192.168.13.0/24
NIC4	Replication network interface	Port 2		12	SPK-Repl-12	192.168.12.0/24

## Jumbo Frames

All HyperFlex storage traffic traversing the hx-storage-data VLAN and subnet is configured by default to use jumbo frames, or to be precise, all communication is configured to send IP packets with a Maximum Transmission Unit (MTU) size of 9000 bytes. In addition, the default MTU for the hx-vmotion VLAN is also set to use jumbo frames. Using a larger MTU value means that each IP packet sent carries a larger payload, therefore transmitting more data per packet, and consequently sending and receiving data faster. This configuration also means that the Cisco UCS uplinks must be configured to pass jumbo frames. Failure to configure the Cisco UCS uplink switches to allow jumbo frames can lead to service interruptions during some failure scenarios, including Cisco UCS firmware upgrades, or when a cable or port failure would cause storage traffic to traverse the northbound Cisco UCS uplink switches.

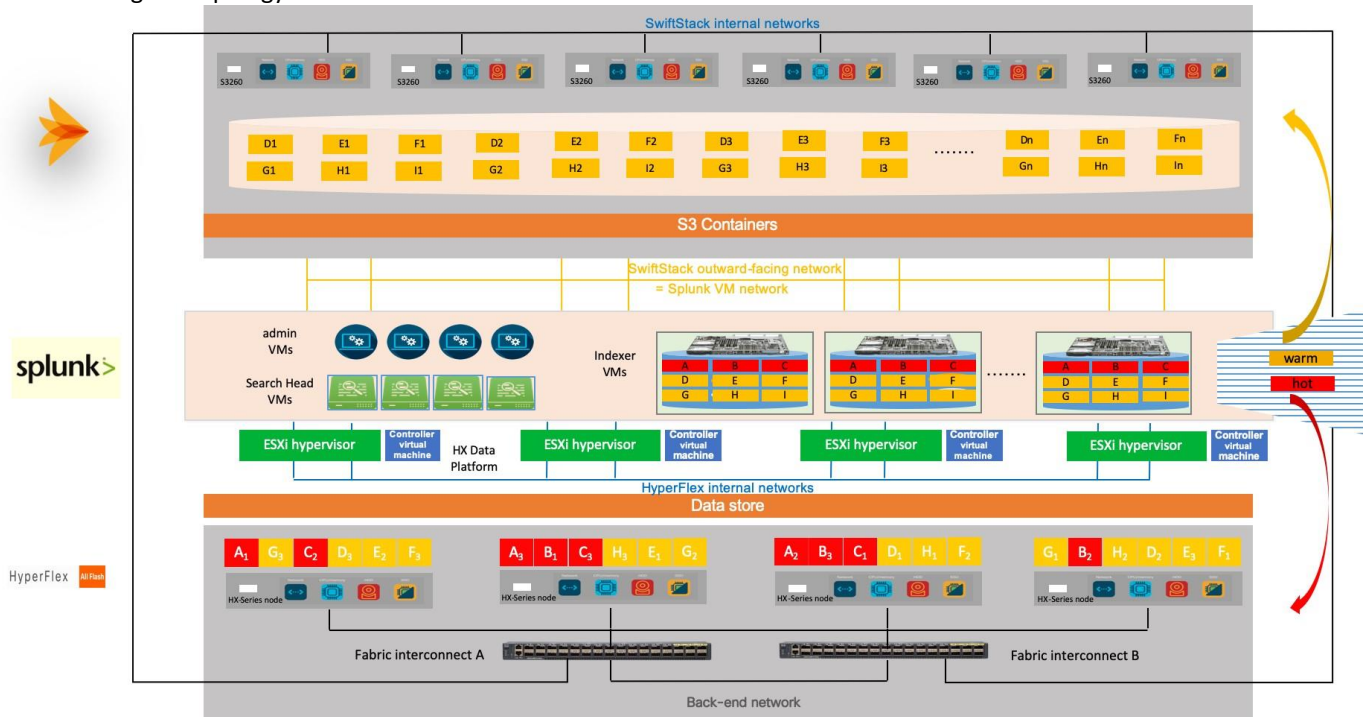
HyperFlex clusters can be configured to use standard size frames of 1500 bytes, however Cisco recommends that this configuration only be used in environments where the Cisco UCS uplink switches are not capable of passing jumbo frames, and that jumbo frames be enabled in all other situations.

The Cisco UCS best practice is to use standard ethernet frames MTU 1500 for the external management network, and to enable Jumbo Frames MTU 9000 for any Storage facing networks (Outward facing Network, Cluster Network and Replication Network) in SwiftStack cluster.

All Splunk traffic traversing the << SPK-Data-11>> VLAN and subnet is configured by default to use standard ethernet frames (MTU 1500). Jumbo frames can be enabled for Splunk data traffic in the HyperFlex cluster if all types of workloads in the Virtual Machine zone allow it.

Figure 30 illustrates a logical view of this solution.

Figure 30 Logical Topology



## Considerations

Before installing the Splunk Enterprise with SmartStore, consider the number of Splunk instances, the number of devices required for the HyperFlex cluster, the number of devices required for the SwiftStack cluster and the usable capacity that the HyperFlex Cluster can provide to the local cache and the SwiftStack can provide to the remote storage for the warm buckets.

## Capacity

Overall usable HyperFlex cluster capacity is based on a number of factors. The number of nodes in the cluster, the number and size of the capacity layer disks, and the replication factor of the HyperFlex HX Data Platform, all affect the cluster capacity. In addition, configuring a cluster as a stretched cluster across two sites modifies the data distribution method, which reduces capacity in favor of data availability. Caching disk sizes are not calculated as part of the cluster capacity.

Disk drive manufacturers have adopted a size reporting methodology using calculation by powers of 10, also known as decimal prefix. As an example, a 120 GB disk is listed with a minimum of  $120 \times 10^9$  bytes of usable addressable capacity, or 120 billion bytes. However, many operating systems and filesystems report their space based on standard computer binary exponentiation, or calculation by powers of 2, also called binary prefix. In this example,  $2^{10}$  or 1024 bytes make up a kilobyte,  $2^{10}$  kilobytes make up a megabyte,  $2^{10}$  megabytes make up a gigabyte, and  $2^{10}$  gigabytes make up a terabyte. As the values increase, the disparity between the two systems of measurement and notation get worse, at the terabyte level, the deviation between a decimal prefix value and a binary prefix value is nearly 10 percent.

The International System of Units (SI) defines values and decimal prefix by powers of 10 as are listed in Table 12.



Table 12 SI Unit Values (Decimal Prefix)

Value	Symbol	Name
1000 bytes	kB	Kilobyte
1000 kB	MB	Megabyte
1000 MB	GB	Gigabyte
1000 GB	TB	Terabyte

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) defines values and binary prefix by powers of 2 in ISO/IEC 80000-13:2008 Clause 4 as listed in Table 13.

Table 13 IEC Unit Values (Binary Prefix)

Value	Symbol	Name
1024 bytes	KiB	Kibibyte
1024 KiB	MiB	Mebibyte
1024 MiB	GiB	Gibibyte
1024 GiB	TiB	Tebibyte

For the purpose of this document, the decimal prefix numbers are used only for raw disk capacity as listed by the respective manufacturers. For all calculations where raw or usable capacities are shown from the perspective of the HyperFlex Data Platform software, filesystems or operating systems, the binary prefix numbers are used. This is done primarily to show a consistent set of values as seen by the end user from within the HyperFlex vCenter Web Plugin and HyperFlex Connect GUI when viewing cluster capacity, allocation and consumption, and also within most operating systems.

[Appendix A: HyperFlex Cluster Capacity Calculations](#) describes how to calculate the usable capacity of the HyperFlex Cluster.

This solution builds a HX cluster starting with four HXAF240c-M5SX rack servers. Table 14 lists a set of HyperFlex cluster usable capacity values, using binary prefix, for an array of cluster configurations. These values provide an example of the capacity calculations, for determining the appropriate size of the cluster to initially purchase. Additional savings from deduplication and compression will raise the effective logical capacity far beyond the physical capacity of the nodes. Additionally, the choice of replication factor 3, or 2, will determine the overall efficiency of the real data being stored on the nodes.

Table 14 HX Cluster Usable Capacities

HX-Series Server Model	Node Quantity	Capacity SSD Size (each)	Capacity Disk Quantity (per node)	Usable Capacity (per node)	Cluster Usable Capacity @ RF3	Cluster Usable Capacity @ RF2
HXAF240c-M5SX	4	960 GB	23	20.8 TiB	24.6 TiB	36.9 TiB
		3.8 TB	23	82.3 TiB	97.4 TiB	148.1 TiB

The calculation of the overall usable capacity of SwiftStack cluster is approximate and was mainly based on how replication is done in the cluster, then multiplied by a reservation factor (0.9 in this case). For the standard replica, every write to the cluster is followed by 2 other writes to two other nodes in the cluster. The effective usable space in the cluster hence is around 1/3rd of the raw space. For the EC8-4 replica, the rate of data segments to parity segments is 2:1 so the effective usable space in the cluster only subtracts by 1/3rd of the raw space. Table 14 lists a set of SwiftStack cluster usable capacity values, using binary prefix, for an array of cluster configurations. It is assumed one persistent replication policy is applied to the whole cluster.

Table 15 SwiftStack Cluster Usable Capacities

Storage Node Model	Node Quantity	Capacity Disk Size (each)	Capacity Disk Quantity (per node)	Usable Capacity (per node)	Cluster Usable Capacity @ Standard Replica	Cluster Usable Capacity @ EC8-4 Replica
UCS S3260 M5 server	6	12 TB	27	291.6 TiB	583.2 TiB	1166.4 TiB

## Scale

Cisco HyperFlex systems combine software-defined computing in the form of Cisco UCS servers, software defined storage with powerful Cisco HyperFlex HX Data Platform software, and software defined networking with Cisco unified fabric. HyperFlex cluster has good scale-out capability for deploying any applications at any size in a virtualized environment. It requires a minimum of three (3) Cisco HX-series rack-mount server nodes to create an initial cluster. From that point, the cluster can grow to a cluster of 64 nodes by combining 32 converged nodes and 32 compute-only nodes.

SwiftStack clusters on Cisco UCS S3260 storage servers are validated with a minimum of three (3) Cisco UCS S3260 chassis with single or dual server nodes to create an initial cluster. From that point, the cluster can grow to any size of cluster that is required by the end user which meets their overall storage space requirements. This limitless scaling is a key feature present in SwiftStack which allows future growth without the worries of overall capacity restriction.

The reference virtual infrastructure built in this document provides the computing and storage resources as listed in Table 14

Table 16 Reference Architecture for Splunk Enterprise

Cluster	Hardware Configurations	Cluster Capacity
HyperFlex Cluster (RF=3)	4x HXAF240c-M5SX converged nodes, each one has: 2 x Intel Xeon Gold 6148 20-core CPUs, 40 physical cores 384 GB DIMM 23 x 960 GB or 3.8 TB capacity SSDs 6x UCS C220 M5 computing-only nodes, each one has: 2 x Intel Xeon Gold 6132 14-core CPUs, 28 physical cores 384 GB DIMM	Total: 328 physical cores 656 vCPUs (with Hyper Threading) 3840 GB Memory 24.6 TB usable storage capacity with 960GB SSD or 97.4 TB usable storage capacity with 3.8TB SSD
SwiftStack Cluster	2x UCS C220 M5 rack servers as SwiftStack Controller nodes; 3x dual-server UCS S3260 chassis, each chassis has: 54 x 12 TB capacity HDDs	Total: 583.2 TB usable storage capacity (Triple replica) or 1166.4 TB usable storage capacity (EC8-4 replica)

While Splunk search heads and indexers are CPU and memory intensive, Splunk indexers are also storage intensive. For the virtual deployment, vCPU, memory and disk settings of the indexer virtual machines are three important factors for the design of the underlying scale-out distributed infrastructure. The Storage allocation depends on the rate of ingesting data and the period of data retention. Although 600GB Eager Zero thin provisioned virtual disk is tested in this solution, thin provisioned virtual disk is supported for the indexers deployed in the HX cluster. The size of the disks depends on the real requirements of the deployment and will be variable in different scenarios.

The reference Splunk search head virtual machine has the following configuration:

- 12 vCPUs
- 64 GB Memory

The reference Splunk indexer virtual machine has the following configuration:

- 12 vCPUs
- 64 GB Memory
- Disk size is adjustable and can be calculated following Splunk’s SmartStore Cache Sizing guidelines.

SmartStore Cache Sizing guidelines:

- Minimum cache size =  $I * CF * RF + (C-1) * I * CF$ , where
  - I = Daily Ingestion Rate
  - CF = Compression Ratio (50% by default)
  - RF = Replication Factor (RF = 2 in this case)
  - C = Cache Retention Period (Days) (7/14/30 days respectively)
- Minimum disk space per indexer = Minimum cache size / number of indexers

It is recommended to have one indexer instance per HX host to leverage the power of the infrastructure to the maximum degree with a balanced consumption of resources. Since we have 10 hosts in the cluster, increment of 10 is recommended while considering the number of indexers. Based on the availability of the physical cores, total memory and usable storage capacity of the clusters we build, Table 14 list the examples about how the indexer virtual machines can be scaled.

Table 17 Scale of Indexers

On HX Cluster with total 24.6 or 97.4 TB storage usable space:									
Ingest Rate	1 TB/day			2 TB/day			4 TB/day		
Ingestion per day (GB)	1000	1000	1000	2000	2000	2000	4000	4000	4000
Replication Factor	2	2	2	2	2	2	2	2	2
Compression Ratio	50%	50%	50%	50%	50%	50%	50%	50%	50%
Cache Retention (days)	7	15	30	7	15	30	7	15	30
Minimum Required Cache (GB)	4000	8000	15500	8000	16000	31000	16000	32000	62000
Capacity SSD size (GB)	960	960	960	960	960	3800	960	3800	3800
Number of Indexers	10	10	10	10	10	10	20	20	20
Minimum Storage/Indexer (GB)	400	800	1550	800	1600	3100	800	1600	3100
On SwiftStack Cluster with total 583.2 or 1166.4 TB storage usable space:									

	1 TB/day			2 TB/day			4 TB/day		
Retention in years (EC8-4 Replica)	7.1	7.1	7.1	3.6	3.6	3.6	1.8	1.8	1.8
Retention in years (Standard Replica)	3.6	3.6	3.6	1.8	1.8	1.8	0.9	0.9	0.9

This Cisco HyperFlex reference architecture for Splunk Enterprise with SmartStore supports the massive scalability that Splunk deployments demand. The scale-out solution supports the capacity and performance growth with the expansion of the HX and SwiftStack clusters with new nodes. The configuration described in this document can be extended to support a much larger scale in the same domain or expand to multiple UCS domains that can be supported with Cisco Nexus Series Switches.

## Performance

Planning system resources and bandwidth to enable search and index performance in a distributed virtual environment depends on the total volume of data being indexed and the number of active concurrent searches (scheduled or other) at any time. There are several performance factors to consider when deploying Splunk software inside VMware virtual machines. These considerations are CPU, memory and disk/storage resources.

- CPU: Since Splunk search heads and indexers are CPU intensive, sharing CPU with other virtual machines running on the same host can result in high wait times, which might negatively impact Splunk performance. Splunk indexer and search head virtual machines should have 100% of the vCPU reserved to ensure good performance.
- Memory: Memory is critical for Splunk search heads and indexers. Splunk virtual machines should have reserved memory available to them. VMware hosts running Splunk Enterprise should not be configured with memory overcommit, as overcommitting memory will likely result in poor performance due to ballooning and/or page swapping to the hard disk.
- Disk/Storage: Splunk indexers are usually CPU-and disk I/O-intensive, so disk exposed to these indexers within virtual machines should be capable of 1200+ random seeks per second. In virtual environments, high performance, low latency storage is required for in the Splunk distributed indexing and searches. Cisco HyperFlex All Flash systems provide an excellent choice of hardware and storage for the high-performing virtual infrastructure required for Splunk deployment.

Some things are taken into consideration when the solution is designed for the best performance:

- Indexing less than 250GB of data per day per indexer.
- Up to 5 TB daily indexing.
- Split the Splunk search and indexing functions – indexers for parallelized indexing and search heads to distribute parallelized searches.

The scale-out capabilities of Splunk software include auto load balancing and distributed search to run searches in parallel across indexers. The numbers of search heads depend on the actual deployment scenario in which the user search behaviors are normally different. Please reach out to Splunk and upper application vendors for the guidance.

- Maintain full reservations on CPU and memory settings of the indexer and search head virtual machines.
- Turn off snapshotting for the virtual machines running Splunk Enterprise.
- To keep up with increasing data volumes, it is recommended to scale your deployment by adding additional Splunk indexers.

- The guidelines provided in this section are targeted for Splunk Enterprise for IT Operations Analytics (ITOA) use cases. For premium solutions such as Splunk Enterprise Security (ES) and Splunk IT Services Intelligence (ITSI), increment of use of vCPU and memory resources might be needed.
- Mixing Splunk workload with other applications in the same HyperFlex cluster is supported. But the impact of the scalability and performance needs to be tested by the customers and the guidance should be provided by both Splunk and the application vendors.

For the performance recommendations, refer to the latest [Splunk® Enterprise Capacity Planning Manual](#).

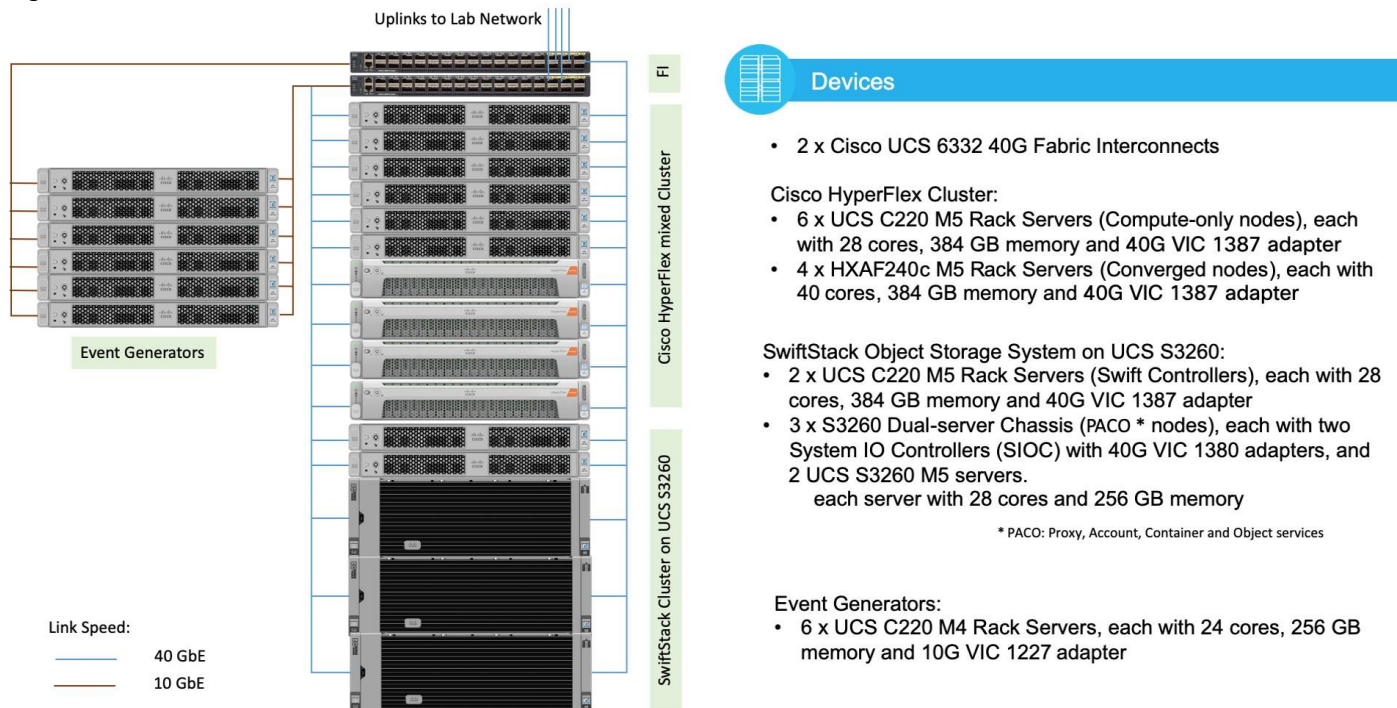
## Deployment of Hardware and Software

This section provides the guidelines to deploy the solution. Most, but not all steps required to install the components are explained in this document. Additionally, for information about installing and configuring this solution, refer to the Cisco UCS, Cisco HyperFlex, VMware vCenter, Splunk Enterprise and SwiftStack Cloud Storage documentation.

### Architecture

Figure 31 shows the devices used in the deployment and validation of this solution:

Figure 31 Devices used in this Solution



### Deployment Guidelines

The following are the high-level installation and configuration steps:

- Step 1: Racking and Cabling
- Step 2: HyperFlex Installation and configuration with nested vCenter
- Step 3: Splunk virtual machine creation and OS installation
- Step 4: Splunk software installation and configuration on the virtual machines
- Step 5: Install SwiftStack software to deploy object storage on Cisco UCS S3260 servers
- Step 6: Configure and provision SwiftStack storage for Splunk SmartStore

## Racking and Cabling

Please refer to the Cisco UCS Installation Guides, Cisco HyperFlex Installation Guides, and the Cisco UCS S3260 Storage Server with SwiftStack CVD for this section. No special configuration is required for this solution.

High-level steps:

1. Install Cisco HyperFlex HXAF240c M5 nodes in a rack.
2. Install Cisco UCS C220 M5 servers in the rack.
3. Install Cisco UCS S3260 chassis and M5 servers in the rack.
4. Install a pair of Cisco Fabric Interconnects in the rack, set up the Fabric Interconnects properly.
5. Connect Cisco HyperFlex HXAF240c M5 nodes and UCS C220 M5 servers to the Fabric Interconnects. Configure the connection ports to the server mode for all the nodes to be discovered in the Cisco UCS Manager.
6. Connect System IO Controllers (SIOC) on the UCS S3260 chassis to the Fabric Interconnects. Configure the connection ports to the server mode for the S3260 chassis and servers that need to be discovered in the Cisco UCS Manager.
7. Upgrade the firmware on all components to the same version bundle.

## Install Cisco HyperFlex Cluster (with Nested vCenter)

Use the following procedures to configure the Cisco HyperFlex system to run the Splunk Enterprise software to provide a big data analytics solution for the virtualized environments. The procedures describe how to deploy and run an HX Data Platform configuration that has a vCenter virtual machine running on the Cisco HyperFlex storage cluster, rather than on a server external to the Cisco HyperFlex storage cluster. Although embedded VMware vSphere vCenter is used as an example for this solution, using an existing vCenter appliance on an external ESXi host or cluster is supported.

### Install Cisco HyperFlex Systems

To create a Cisco HyperFlex cluster, follow these steps:



Definition of vCenter is a post-installation task in this example.

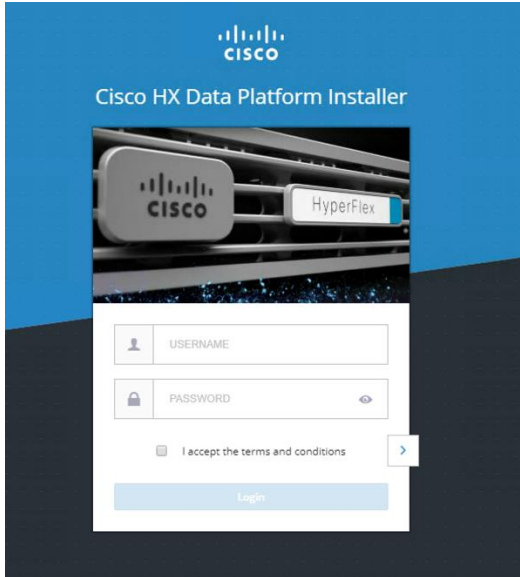
1. Download the HX Data Platform Installer OVA file from the Cisco website. Deploy the installer OVA file on a virtual machine in an existing VMware vSphere, VMware Workstation, VMware Fusion, or other virtualization environment that supports the import of OVA files.

Table 18 Cisco HyperFlex Installer Settings

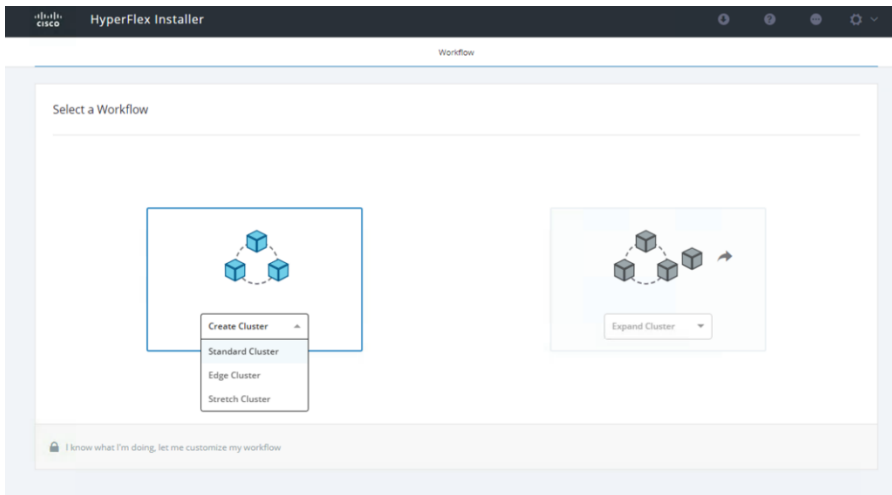
Setting	Value
IP address	
Subnet mask	
Default gateway	
Domain Name System (DNS) server 1	

Setting	Value
Network Time Protocol (NTP) servers	

- Open the HX Data Platform Installer in a local web browser and accept the end-user license agreement.
- Log into the HX Data Platform Installer using the default root user name and password.

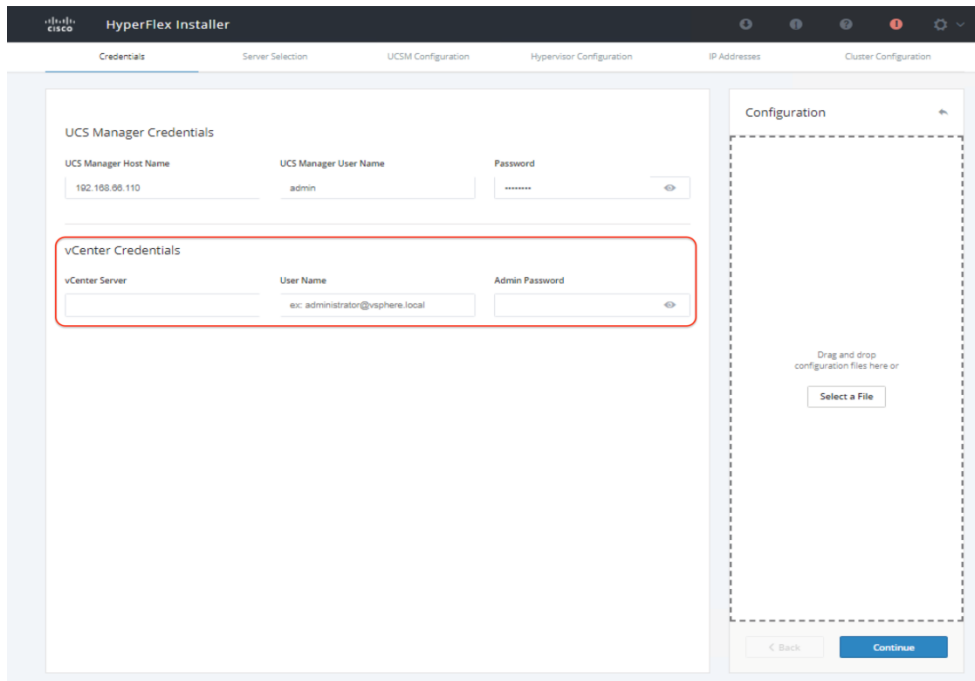


- On the Workflow page, choose Create Cluster > Standard Cluster.

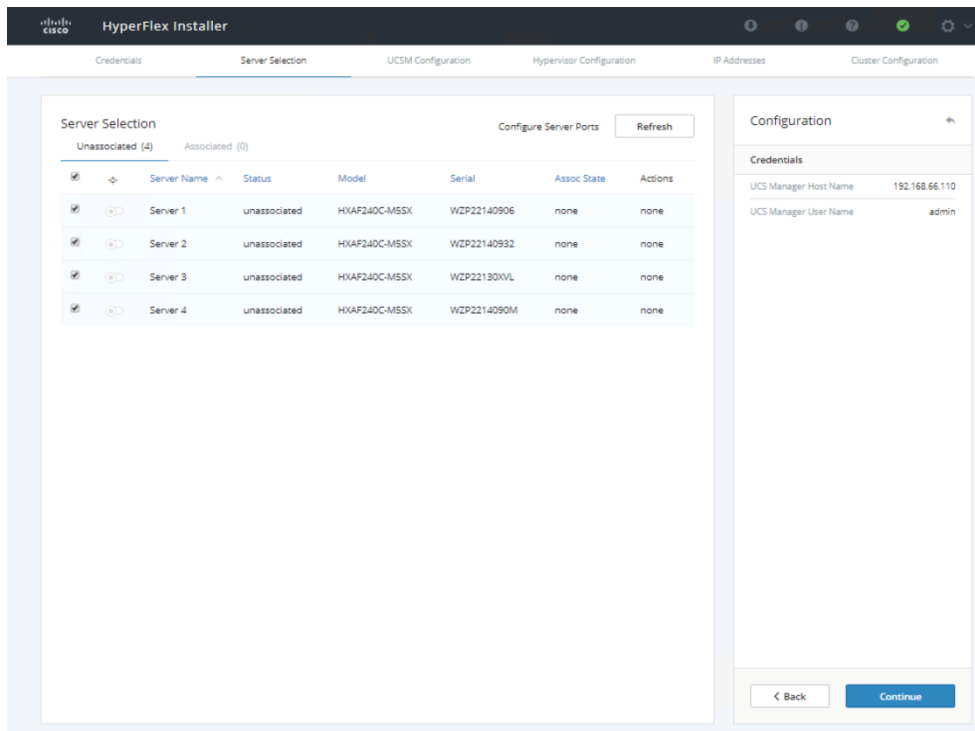


- On the Credentials page, enter the Cisco UCS Manager and HX Data Platform hypervisor credentials. Leave all three vCenter fields blank; the vCenter server will be registered in a post-installation task. Click Continue.





6. Ignore the vCenter warning message and click Continue to confirm that the installation is being started without vCenter.
7. On the Server Selection page, under Unassociated, select the servers that you want to include in the Cisco HyperFlex storage cluster. Click Continue.



8. On the Cisco UCS Manager Configuration page, enter the Cisco UCS Manager configuration information and click Continue. Cisco UCS Manager configuration information includes the VLAN, MAC address pool, subnet, gateway, Small

Computer System Interface over IP (iSCSI) storage, Fibre Channel storage, Cisco UCS firmware, Cisco HyperFlex cluster name, and organization name. Fill in these fields as required for any Cisco HyperFlex storage cluster. Click Continue.



You do not need to enable iSCSI or Fibre Channel storage in this step.

- On the Hypervisor Configuration page, enter common hypervisor settings, such as the subnet mask, gateway, static IP addresses, and host names for the Cisco HyperFlex nodes. You can change the factory default Hypervisor credential here. Click Continue.

### Configure common Hypervisor Settings

Subnet Mask: 255.255.255.0      Gateway: 192.168.66.1

---

### Hypervisor Settings

Make IP Addresses and Hostnames Sequential

#	Name	Serial	Static IP Address	Hostname
1	Server 1	WZP22140906	192.168.66.21	SPK240HXAF-1
2	Server 2	WZP22140932	192.168.66.22	SPK240HXAF-2
3	Server 3	WZP22130XVL	192.168.66.23	SPK240HXAF-3
4	Server 4	WZP2214090M	192.168.66.24	SPK240HXAF-4

---

### Hypervisor Credentials

Admin User name: root

The hypervisor on this node uses the factory default password

You are required to change the factory default password. Enter a new password for the hypervisor

New Password: \*\*\*\*\*      Confirm New Password: \*\*\*\*\*

### Configuration

**Credentials**

UCS Manager Host Name: 192.168.66.110  
 UCS Manager User Name: admin  
 Admin User name: root

**Server Selection**

Server 2: WZP22140932 / HXAF240C-M55X  
 Server 3: WZP22130XVL / HXAF240C-M55X  
 Server 1: WZP22140906 / HXAF240C-M55X  
 Server 4: WZP2214090M / HXAF240C-M55X

**UCSM Configuration**

VLAN Name: hv-inband-mgmt  
 VLAN ID: 3021  
 VLAN Name: hv-storage-data  
 VLAN ID: 3022  
 VLAN Name: hv-vmotion  
 VLAN ID: 3023  
 VLAN Name: vmm-network  
 VLAN ID(s): 3024  
 MAC Pool Prefix: 00:25:B5:D2  
 IP Blocks: 192.168.66.41-50  
 Subnet Mask: 255.255.255.0  
 Gateway: 192.168.66.1

- On the IP Addresses page, for each Cisco HyperFlex node, complete the fields for hypervisor management, data IP addresses, subnet masks, and gateways. Specify the IP address settings for the Management network and the Data network.

### IP Addresses

Make IP Addresses Sequential

Management - VLAN 3021      Data - VLAN 3022 (FQDN or IP Address)

#	Name	Hypervisor	Storage Controller	Hypervisor	Storage Controller
2	Server 2	192.168.66.22	192.168.66.32	192.168.77.22	192.168.77.32
4	Server 4	192.168.66.24	192.168.66.34	192.168.77.24	192.168.77.34
1	Server 1	192.168.66.21	192.168.66.31	192.168.77.21	192.168.77.31
3	Server 3	192.168.66.23	192.168.66.33	192.168.77.23	192.168.77.33

Cluster IP Address: Management: 192.168.66.20      Data: 192.168.77.20

Subnet Mask: 255.255.255.0      255.255.255.0

### Configuration

**Credentials**

UCS Manager Host Name: 192.168.66.110  
 UCS Manager User Name: admin  
 Admin User name: root

**Server Selection**

Server 2: WZP22140932 / HXAF240C-M55X  
 Server 3: WZP22130XVL / HXAF240C-M55X  
 Server 1: WZP22140906 / HXAF240C-M55X  
 Server 4: WZP2214090M / HXAF240C-M55X

**UCSM Configuration**

VLAN Name: hv-inband-mgmt  
 VLAN ID: 3021  
 VLAN Name: hv-storage-data  
 VLAN ID: 3022  
 VLAN Name: hv-vmotion  
 VLAN ID: 3023  
 VLAN Name: vmm-network

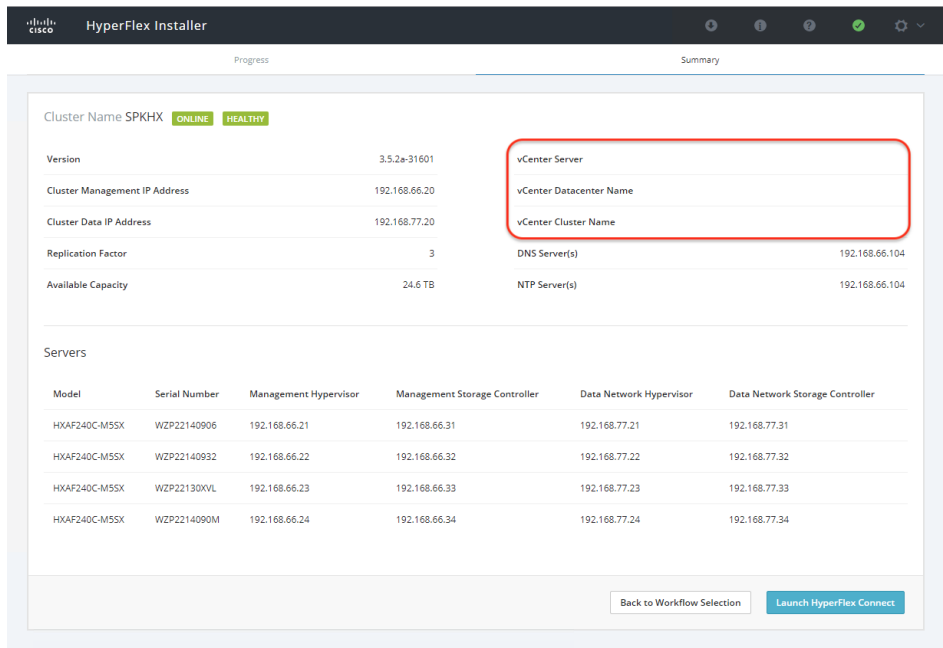
- On the Cluster Configuration page, enter the Cisco HyperFlex storage cluster settings, such as the storage cluster name, controller virtual machine credentials, data replication factor, DNS and NTP servers, and autosupport (ASUP). At this step, set RF = 3. Skip the vCenter configurations and leave the fields blank.

- Click Start to create the Cisco HyperFlex cluster.



For detailed information, see the Cisco HyperFlex Systems Installation Guide for VMware ESXi for deployment and cluster creation steps.

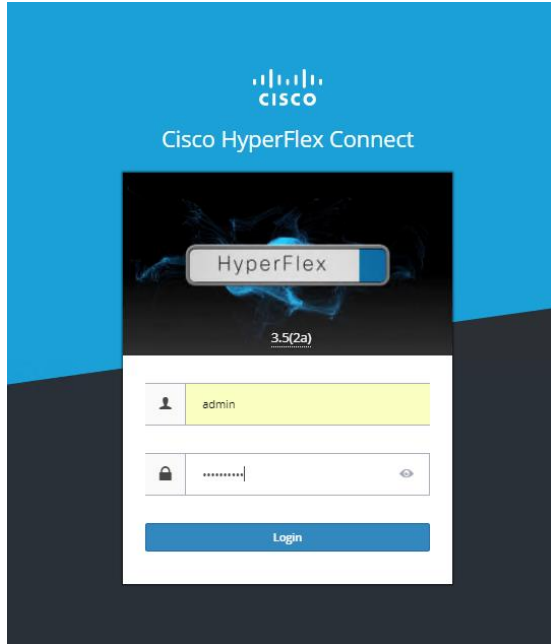
- Wait for the cluster creation process to complete and for the summary page to appear. Note that the vCenter Server information is not specified.



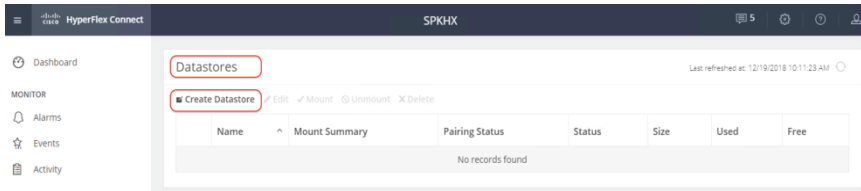
### Configure the Data Store

To configure the data stores for the Cisco HyperFlex storage cluster, follow these steps:

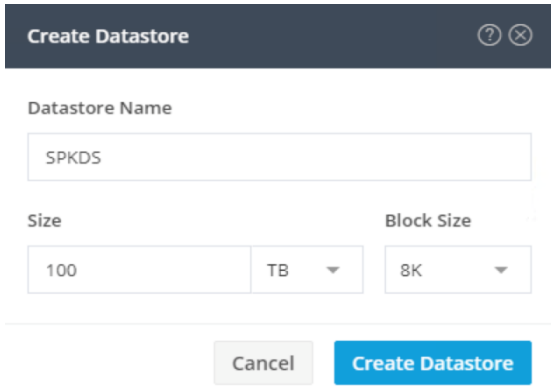
1. Log into the HX Connect management console using the cluster management IP address with the admin credentials. Ignore the missing vCenter error message.



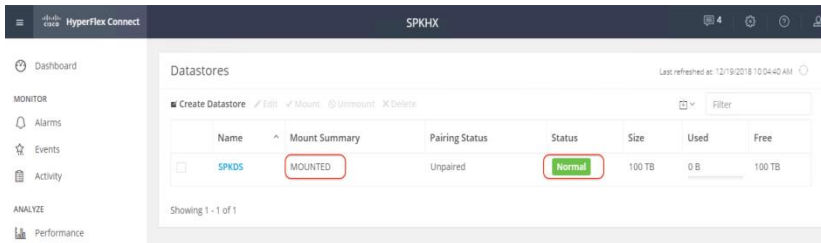
2. Click Dashboard > Datastores > Create Datastore.



3. Enter Datastore Name (for example, SPKDS), Size, and Block Size, then click Create Datastore



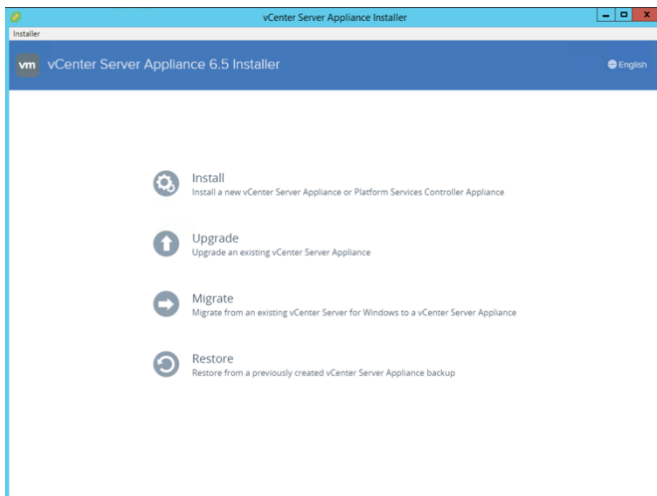
4. After the data store has been created successfully, check the data store status. Make sure that the data store shows as Normal and Mounted.



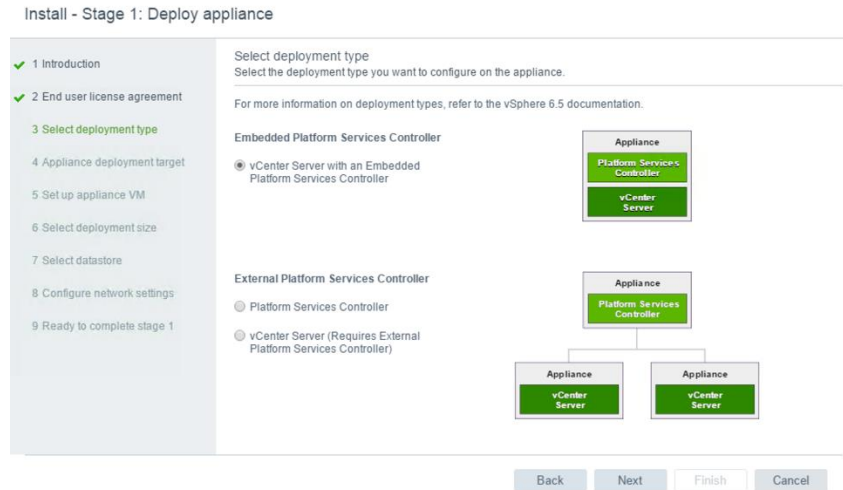
## Install VMware vCenter

To install the embedded VMware vCenter Server appliance on the Cisco HyperFlex storage cluster, follow these steps:

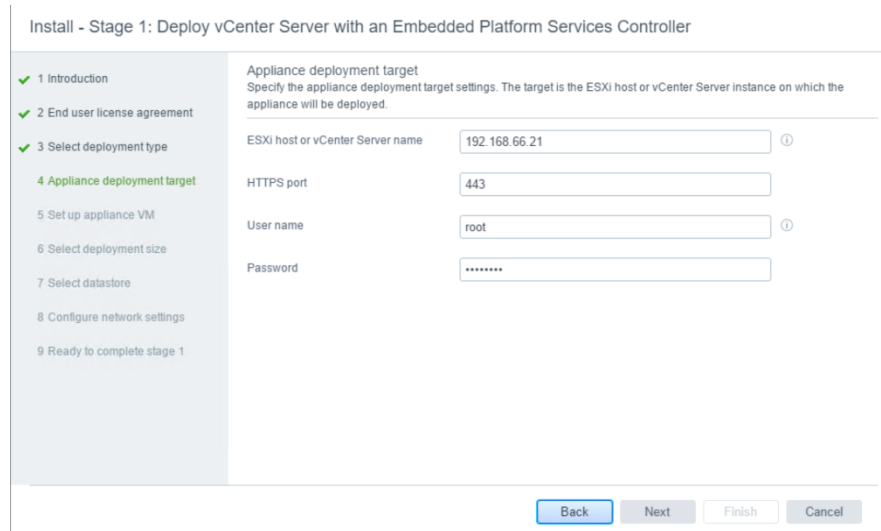
1. Use the vCenter Server Appliance Installer wizard to build a new vCenter appliance virtual machine.



2. Review the introduction to stage 1 of the installation. Click Next.
3. Accept the terms of the license agreement. Click Next.
4. Choose to install vCenter with an embedded platform services controller. Click Next.



5. Deploy the new vCenter on any one of the Cisco HyperFlex servers in the cluster. Click Next.



6. Ignore the certificate warning by clicking Yes.
7. Enter the virtual machine's name and password. Click Next.

Install - Stage 1: Deploy vCenter Server with an Embedded Platform Services Controller

1 Introduction  
 2 End user license agreement  
 3 Select deployment type  
 4 Appliance deployment target  
 5 Set up appliance VM  
 6 Select deployment size  
 7 Select datastore  
 8 Configure network settings  
 9 Ready to complete stage 1

Set up appliance VM  
Specify the VM settings for the appliance to be deployed.

VM name:

Root password:

Confirm root password:

8. Select the deployment size based on the manage needs of your environment. Click Next.

Install - Stage 1: Deploy vCenter Server with an Embedded Platform Services Controller

1 Introduction  
 2 End user license agreement  
 3 Select deployment type  
 4 Appliance deployment target  
 5 Set up appliance VM  
 6 Select deployment size  
 7 Select datastore  
 8 Configure network settings  
 9 Ready to complete stage 1

Select deployment size  
Select the deployment size for this vCenter Server with an Embedded Platform Services Controller.

For more information on deployment sizes, refer to the vSphere 6.5 documentation.

Deployment size:

Storage size:

Resources required for different deployment sizes

Deployment Size	vCPUs	Memory (GB)	Storage (GB)	Hosts (up to)	VMs (up to)
Tiny	2	10	250	10	100
Small	4	16	290	100	1000
Medium	8	24	425	400	4000
Large	16	32	640	1000	10000
X-Large	24	48	980	2000	35000

9. Use the newly created data store for persistent storage. Click Next.

Install - Stage 1: Deploy vCenter Server with an Embedded Platform Services Controller

1 Introduction  
 2 End user license agreement  
 3 Select deployment type  
 4 Appliance deployment target  
 5 Set up appliance VM  
 6 Select deployment size  
 7 Select datastore  
 8 Configure network settings  
 9 Ready to complete stage 1

Select datastore  
Select the storage location for this vCenter Server with an Embedded Platform Services Controller.

Install on an existing datastore accessible from the target host

Name	Type	Capacity	Free	Provisi...	Thin Provisioni...
SPKDS	NFS	100 TB	100 TB	0 B	true
SpringpathDS-WZP22140906	VMFS	216 GB	207.64 GB	8.36 GB	true

2 items

Enable Thin Disk Mode

Install on a new Virtual SAN cluster containing the target host

10. Select an appropriate port group for the network. The port group must have network access to the Cisco HyperFlex cluster management IP address and all ESXi management IP addresses. Specify Input the IP configuration. Click Next.

Install - Stage 1: Deploy vCenter Server with an Embedded Platform Services Controller

Configure network settings  
Configure network settings for this vCenter Server with an Embedded Platform Services Controller.

1 Introduction  
2 End user license agreement  
3 Select deployment type  
4 Appliance deployment target  
5 Set up appliance VM  
6 Select deployment size  
7 Select datastore  
8 **Configure network settings**  
9 Ready to complete stage 1

Network: Storage Controller Management Network

IP version: IPv4

IP assignment: static

System name: SPKHx-VC.dmzhx.lab.cisco.com

IP address: 192.168.66.51

Subnet mask or prefix length: 255.255.255.0

Default gateway: 192.168.66.1

DNS servers: 192.168.66.104

Common Ports

Back Next Finish Cancel

11. Review the configuration for your vCenter server appliance virtual machine. Click Finish to start the installation.

Install - Stage 1: Deploy vCenter Server with an Embedded Platform Services Controller

1 Introduction  
2 End user license agreement  
3 Select deployment type  
4 Appliance deployment target  
5 Set up appliance VM  
6 Select deployment size  
7 Select datastore  
8 **Configure network settings**  
9 Ready to complete stage 1

Deployment Details

Target ESXi host	192.168.66.21
VM name	SPKHx-VC
Deployment type	vCenter Server with an Embedded Platform Services Controller
Deployment size	Tiny

Datastore Details

Datastore, Disk mode	SPKDS, thin
----------------------	-------------

Network Details


Network	Storage Controller Management Network
IP settings	IPv4, static
IP address	192.168.66.51
System name	SPKHx-VC.dmzhx.lab.cisco.com
Subnet mask or prefix length	255.255.255.0
Default gateway	192.168.66.1
DNS servers	192.168.66.104
HTTP Port	80
HTTPS Port	443

Back Next Finish Cancel

12. When stage 1 deployment is complete, click Continue for stage 2 vCenter configuration.



Install - Stage 1: Deploy vCenter Server with an Embedded Platform Services Controller

 You have successfully deployed the vCenter Server with an Embedded Platform Services Controller.

Deployment complete 100%

To proceed with stage 2 of the deployment process, appliance setup, click Continue.

If you exit, you can continue with the appliance setup at any time by logging in to the vCenter Server Appliance Management interface <https://XLFVcenter.hx.lab.cisco.com:548/>

Continue Close

- 13. Review the introduction to stage 2 of the deployment process. Click Next.
- 14. Configure the NTP servers in the appliance and enable remote SSH access to allow high availability for VCenter. Click Next.

Install - Stage 2: Set Up vCenter Server Appliance with an Embedded PSC

Appliance configuration

- 1 Introduction
- 2 Appliance configuration
- 3 SSO configuration
- 4 Configure CEIP
- 5 Ready to complete

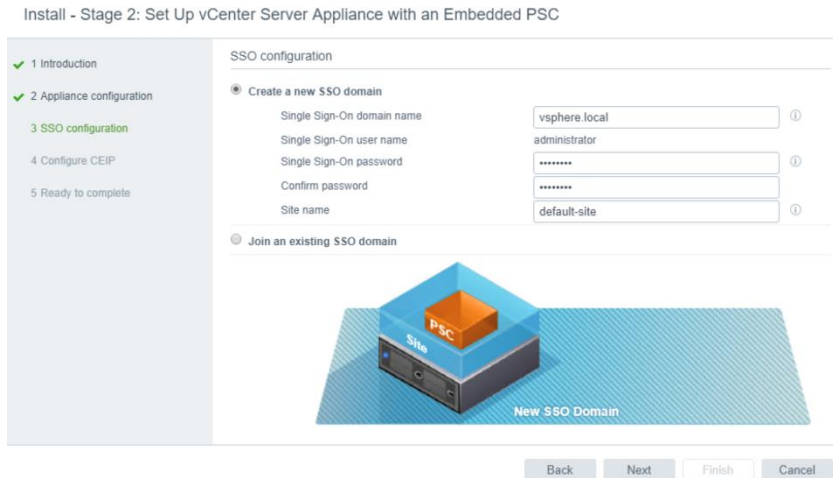
Time synchronization mode: Synchronize time with NTP servers

NTP servers (comma-separated list): 192.168.66.104

SSH access: Enabled

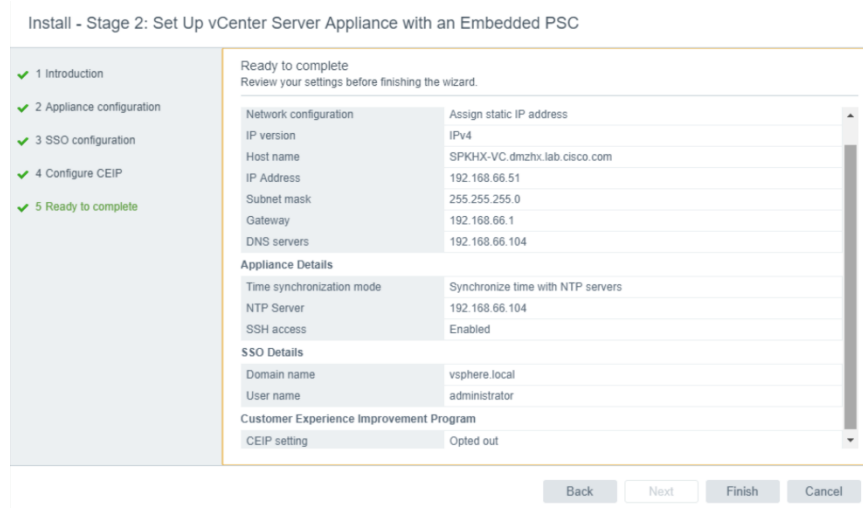
Back Next Finish Cancel

- 15. Create a new vCenter single sign-on (SSO) domain or join an existing domain. For example, enter vsphere.local as the domain name, set the password for the administrator account, and enter a default site name. Click Next.

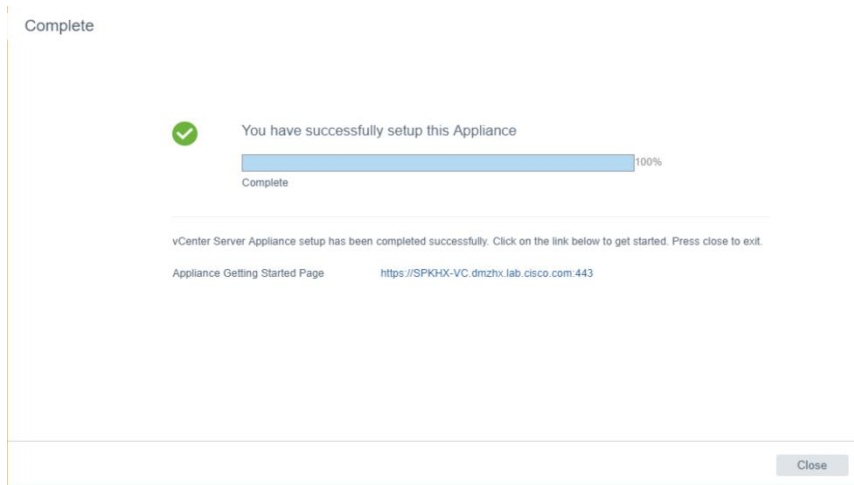


16. Review the VMware Customer Experience Improvement Program (CEIP) page and select or deselect the Join box. Click Next.

17. Review the settings on the Ready to Complete page and click Finish.



18. Click OK to complete stage 2 of the deployment process and set up the appliance.

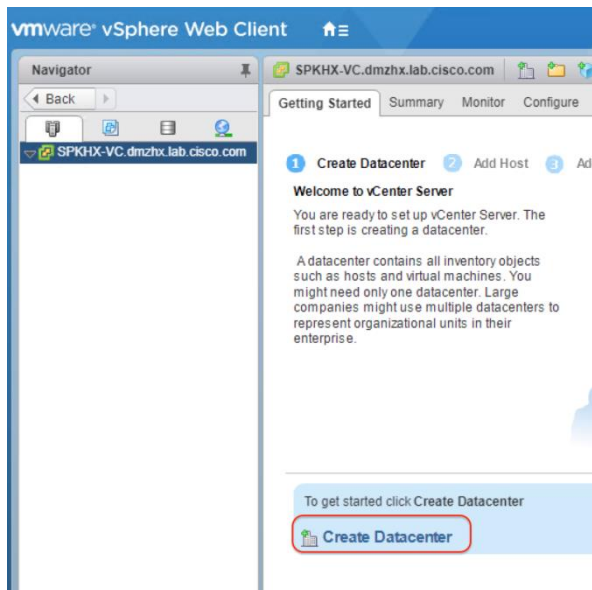


19. When the deployment completes successfully, click Close to exit the installer wizard.

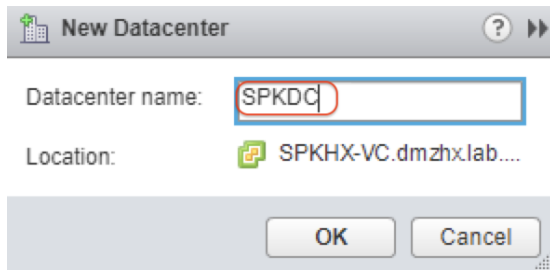
## Configure VMware vSphere

To configure the embedded VMware vSphere server for the Cisco HyperFlex cluster and Splunk virtual machines, follow these steps:

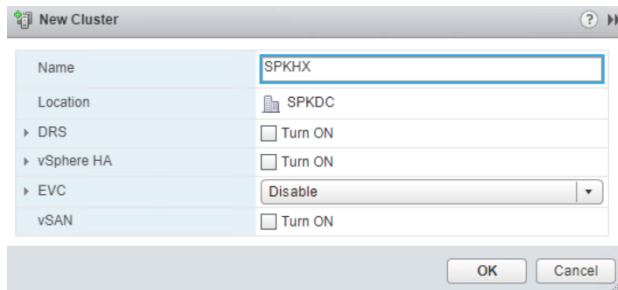
1. After successfully deploying vCenter, log into the vSphere Web Client. On the Getting Started page, choose Create Data Center.



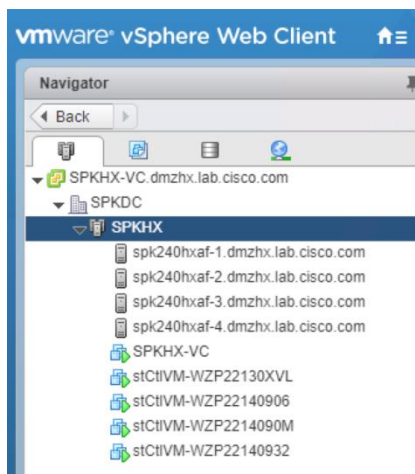
2. Enter the new data center name—for example, **SPKDC**—and click OK.



3. Create a new cluster in the new data center—for example, **SPKH-X**—leaving high availability (HA) and VMware Distributed Resource Scheduler (DRS) disabled (they will be enabled later). Click OK.



4. Right-click the newly created cluster and choose Add Host to manually add one Cisco HyperFlex server to the cluster.
5. Enter the host name, root user, and password to connect the server.
6. Assign an ESXi license, or use the evaluation license and assign a license later.
7. Keep the default Lockdown mode.
8. Review the settings and then click Finish to add the Cisco HyperFlex node.
9. Repeat step 4 to add all the Cisco HyperFlex servers to the Cisco HyperFlex cluster.



10. Log out of the vSphere Web Client.
11. Register the Cisco HyperFlex storage cluster to the newly configured vCenter server.

12. Log into the Cisco HyperFlex cluster management IP address using SSH and the root credentials. Then run the following command:

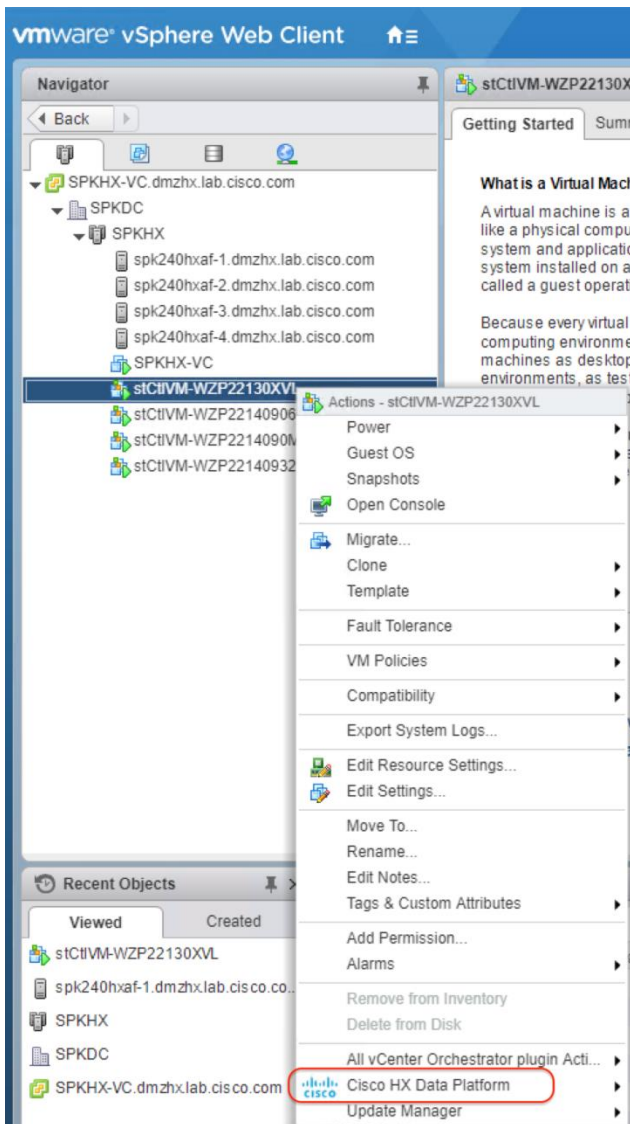
```
stcli cluster reregister --vcenter-datacenter <Datacenter Name> --vcenter-cluster
<Cluster Name> --vcenter-url <URL or IP of vCenter> --vcenter-user <admin username>
```

Here is an example of the command:

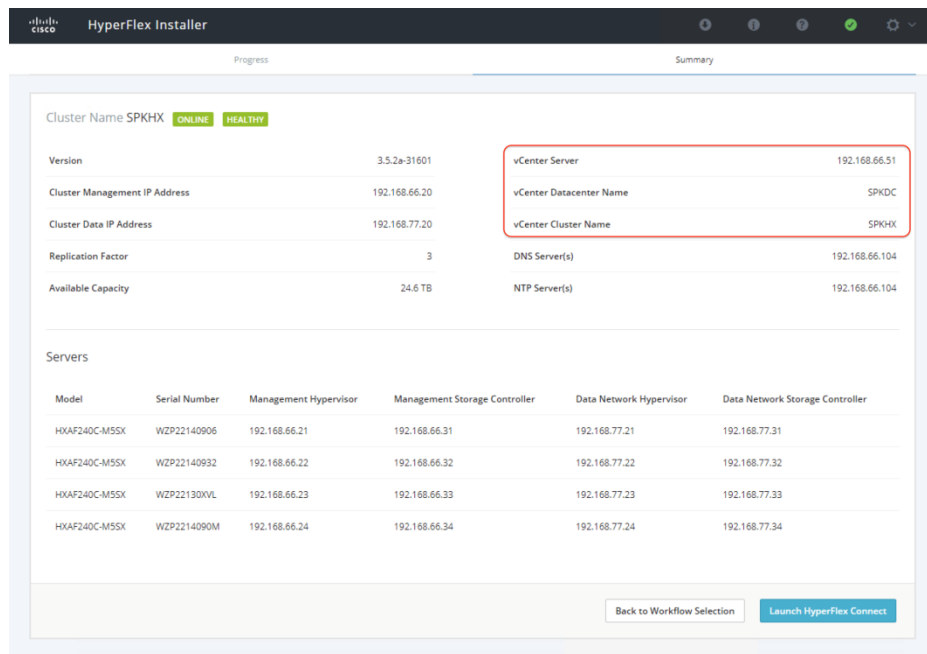
```
# stcli cluster reregister --vcenter-datacenter SPKDC --vcenter-cluster SPKHx --vcenter-
url 192.168.66.51 --vcenter-user administrator@vsphere.local
Register StorFS cluster with vCenter ...
Enter NEW vCenter Administrator password: *****
Cluster registration with new vCenter succeeded
```

```
root@SpringpathController0T5M96TVPJ:~# stcli cluster reregister --vcenter-datacenter SPKDC --vcenter-cluster SPKHx --vcenter-url 192.168.66.51 --vcenter-user administrator@vsphere.local
Reregister StorFS cluster with a new vCenter ...
Enter NEW vCenter Administrator password:
Cluster reregistration with new vCenter succeeded
root@SpringpathController0T5M96TVPJ:~#
```

13. Log into vSphere Web Client and verify that the HX Data Platform plug-in appears in the extension list.



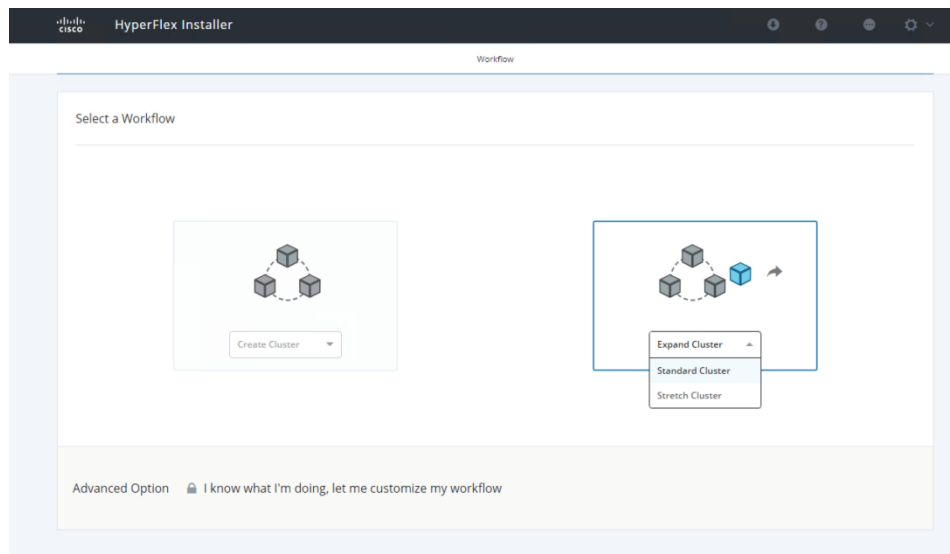
14. Refresh the Summary page of the Cisco HyperFlex installer. Verify that now the vCenter Server is specified.



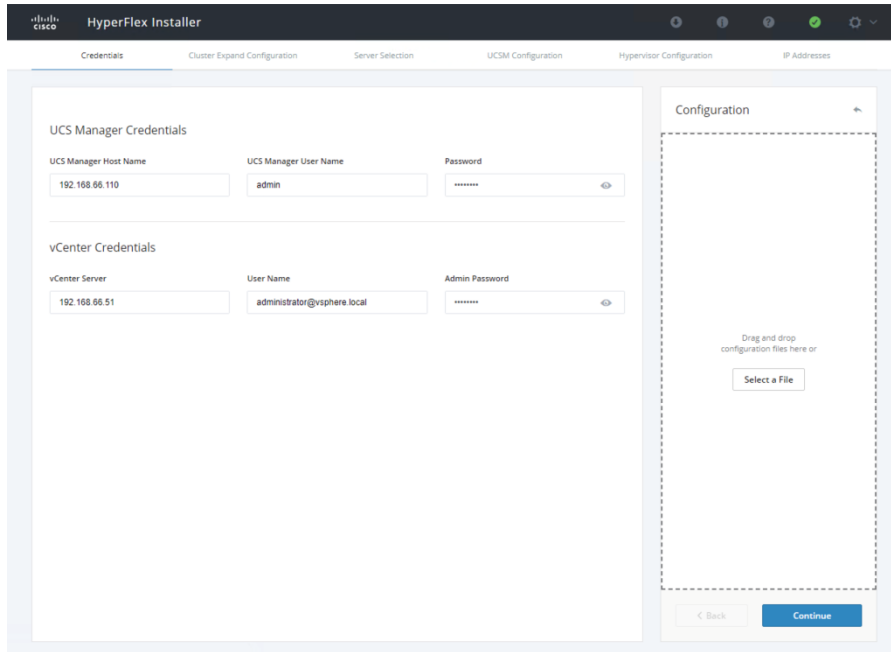
### HyperFlex Cluster Expansion with Computing-only Nodes

To expand an existing HyperFlex cluster with additional computing resources, follow these steps:

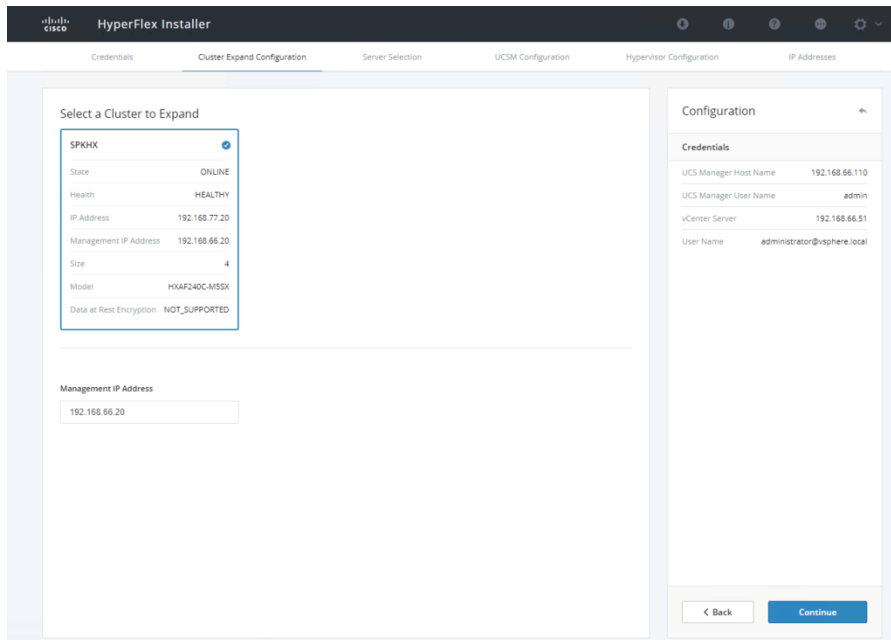
1. Log into the HX Data Platform Installer using the default root username and password.
2. On the Workflow page, Click Expand cluster > Standard Cluster.



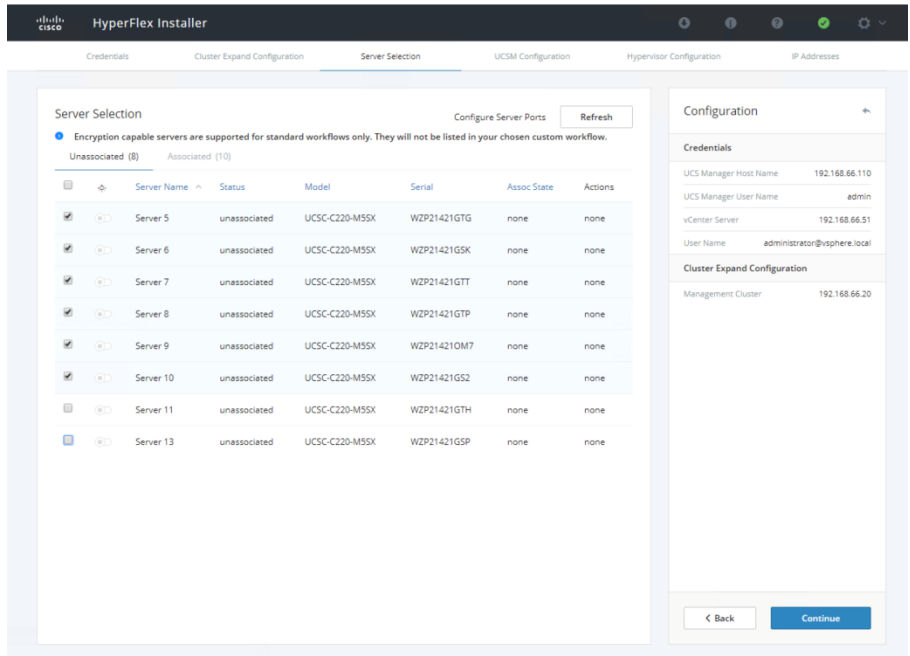
3. On the Credentials page, enter the UCS Manager and vCenter credentials. Click Continue.



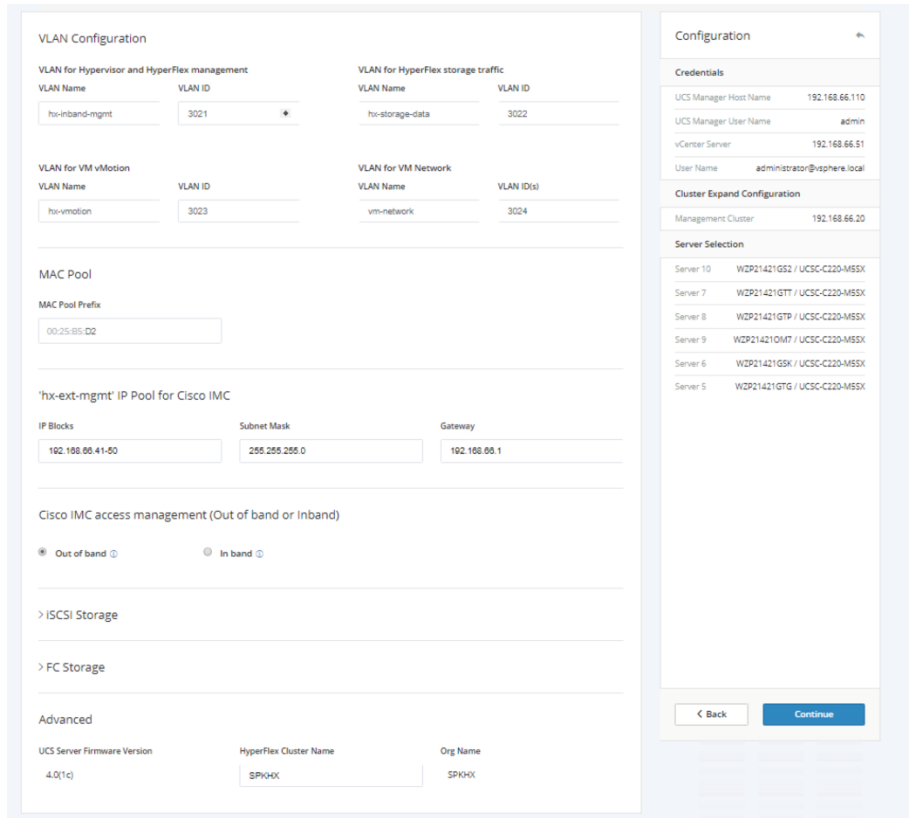
4. On the Cluster Expand Configuration page, select the cluster to be expanded and click Continue.



5. On the Server Selection page, select the UCS servers under Unassociated for expansion into the HX storage cluster and click Continue.



6. On the UCSM Configuration page, enter the UCS Manager configuration information to match the existing nodes, and click Continue.
7. UCS Manager configuration information includes the VLAN, MAC pool, subnet, gateway, iSCSI storage, FC storage, UCS firmware, HyperFlex cluster name, and org name. Fill out these fields normally as required for any HyperFlex storage cluster. No need to enable FC Storage or iSCSI storage at this step.





- On the Hypervisor Configuration page, enter common hypervisor settings, such as subnet mask, gateway, static IP addresses, and hostnames for the HyperFlex nodes. The factory default Hypervisor credential is changed here. Click Continue.

- On the IP Addresses page, for each HyperFlex node for expansion, complete the fields for Hypervisor Management, Data IP addresses. Enter the controller virtual machine password that matches the password of the existing HX cluster controller virtual machines. Click Start to expand the HyperFlex cluster.



Pay attention if the following warning page displays:

Warning X

---

Compute node deployment defaults to using FlexFlash (mirrored SD cards) for boot. If your compute-only node(s) uses any other boot media, you must create a new Local Disk Configuration Policy and apply it to the newly created service profiles.

For the complete procedure, see Chapter: Expanding Cisco HyperFlex System Clusters in the Cisco HyperFlex Systems Installation Guide.

Cancel
Continue

10. Check what boot disk your computing-only nodes have. By default, no user intervention is required if you are booting from FlexFlash (SD Card). However, if you are setting up your compute-only nodes with different booting option, e.g. boot from a local disk, you have to create a new Local Disk Configuration Policy and apply it to the service profiles of these computing-only servers in Cisco UCS Manager. The complete procedures are described in the [Cisco HyperFlex Systems Installation Guide](#) (Chapter: Expand Cisco HyperFlex System Clusters).
11. Click Continue, wait for the cluster expansion to complete and the summary page displays. Note that all computing-only nodes have been added and the existing HX cluster has been expanded.

Cluster Name SPKHX ONLINE HEALTHY

Version	3.5.2a-31601	vCenter Server	192.168.66.51
Cluster Management IP Address	192.168.66.20	vCenter Datacenter Name	SPKDC
Cluster Data IP Address	192.168.77.20	vCenter Cluster Name	SPKHX
Replication Factor	3	DNS Server(s)	192.168.66.104
Available Capacity	24.6 TB	NTP Server(s)	192.168.66.104

Servers

Model	Serial Number	Management Hypervisor	Management Storage Controller	Data Network Hypervisor	Data Network Storage Controller	
HXAF240C-M55X	WZP22140906	192.168.66.21	192.168.66.31	192.168.77.21	192.168.77.31	HX Converged Nodes x4
HXAF240C-M55X	WZP22140932	192.168.66.22	192.168.66.32	192.168.77.22	192.168.77.32	
HXAF240C-M55X	WZP22130XVL	192.168.66.23	192.168.66.33	192.168.77.23	192.168.77.33	
HXAF240C-M55X	WZP2214090M	192.168.66.24	192.168.66.34	192.168.77.24	192.168.77.34	
UCSC-C220-M55X	WZP21421GTG	192.168.66.25		192.168.77.25		Computing-only Nodes x6
UCSC-C220-M55X	WZP21421GSK	192.168.66.26		192.168.77.26		
UCSC-C220-M55X	WZP21421GTT	192.168.66.27		192.168.77.27		
UCSC-C220-M55X	WZP21421GTP	192.168.66.28		192.168.77.28		
UCSC-C220-M55X	WZP21421OM7	192.168.66.29		192.168.77.29		
UCSC-C220-M55X	WZP21421GS2	192.168.66.30		192.168.77.30		

12. Log into HX Connect management console using the cluster management IP address with the admin credentials. Click System Information, verify the status of HX converged nodes and compute nodes.



The same steps can be used to expand the existing HyperFlex cluster with HX converged node too. The scale-out solution supports the capacity and performance growth with the expansion of the HX cluster with new nodes.

## Perform Cisco HyperFlex Post-installation Configuration

To run the **post\_install** script from the Cisco HyperFlex Installer virtual machine to configure additional settings on the Cisco HyperFlex storage cluster, follow these steps:

1. Log into the Cisco HyperFlex Installer virtual machine IP address using SSH and the root credentials.
2. Run the **post\_install** script and follow the prompts.



The installer will already have the information from the just-completed Cisco HyperFlex installation, and the script will use this information. Enter the Cisco HyperFlex storage controller virtual machine root password for the Cisco HyperFlex cluster (use the name entered during the Cisco HyperFlex cluster installation), as well as the vCenter user name and password.

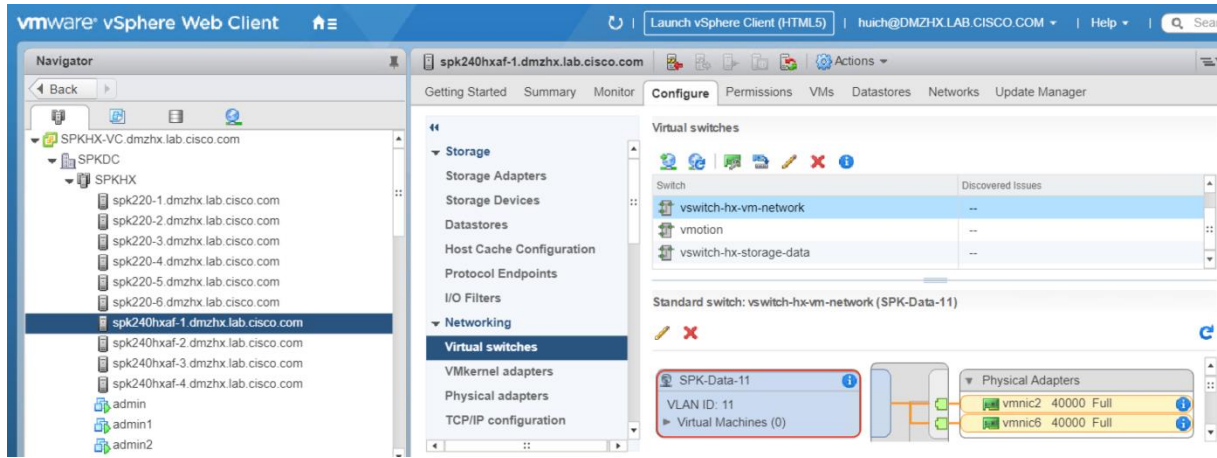
3. You can choose to enter the vSphere license here or complete this task later.
4. Enter **y** to enable HA and DRS if you have the appropriate licenses.
5. Enter **y** to disable the ESXi hosts' SSH warning.
6. Add the VMware vMotion VMkernel interfaces to each node by entering **y**. Enter the netmask, the vMotion VLAN ID, and the vMotion IP addresses for each of the hosts as prompted.

7. Enter **y** to add virtual machine VLANs to create two guest network port groups: one for Splunk Data Network and another for Splunk Replication Network (optional).
8. The VLANs that will be used for Splunk installation must already be trunked to the Cisco UCS fabric interconnects from the northbound network by the upstream switches, and this configuration step must be performed manually prior to beginning the Splunk deployment. It is recommended that MTU settings to support Jumbo frames for these VLANs be enabled but using the default 1500 MTU for the data network is also a common option depending on the requirement of the Splunk data source.
9. If desired, enter **y** to run a health check on the cluster, or enter **n** to skip this step.

Here is an example of a completed configuration:

```
[Administrator:HXJumplWin1] > ssh root@192.168.66.106
X11 forwarding request failed on channel 0
Last login: Wed Dec 19 13:15:41 2018 from 192.168.66.103
root@HyperFlex-Installer:~# post_install
Logging in to controller 192.168.66.20
HX CVM admin password:
Getting ESX hosts from HX cluster...
vCenter URL: 192.168.66.51
Enter vCenter Username (user@domain): administrator@vsphere.local
vCenter Password:
Found datacenter SPKDC
Found cluster SPKHX
Enter ESX root password:
Enter vSphere license key? (y/n) n
Enable HA/DRS on cluster? (y/n) y
Disable SSH warning? (y/n) y
Add vmotion interfaces? (y/n) y
Netmask for vMotion: 255.255.255.0
VLAN ID: (0-4096) 3023
vMotion MTU is set to use jumbo frames (9000 bytes). Do you want to change to 1500 bytes? (y/n) n
vMotion IP for spk220-1.dmzhx.lab.cisco.com: 192.168.88.25
Adding vmotion-3023 to spk220-1.dmzhx.lab.cisco.com
Adding vmkernel to spk220-1.dmzhx.lab.cisco.com
vMotion IP for spk220-2.dmzhx.lab.cisco.com: 192.168.88.26
Adding vmotion-3023 to spk220-2.dmzhx.lab.cisco.com
Adding vmkernel to spk220-2.dmzhx.lab.cisco.com
vMotion IP for spk220-3.dmzhx.lab.cisco.com: 192.168.88.27
Adding vmotion-3023 to spk220-3.dmzhx.lab.cisco.com
Adding vmkernel to spk220-3.dmzhx.lab.cisco.com
vMotion IP for spk220-4.dmzhx.lab.cisco.com: 192.168.88.28
Adding vmotion-3023 to spk220-4.dmzhx.lab.cisco.com
Adding vmkernel to spk220-4.dmzhx.lab.cisco.com
vMotion IP for spk220-5.dmzhx.lab.cisco.com: 192.168.88.29
Adding vmotion-3023 to spk220-5.dmzhx.lab.cisco.com
Adding vmkernel to spk220-5.dmzhx.lab.cisco.com
vMotion IP for spk220-6.dmzhx.lab.cisco.com: 192.168.88.30
Adding vmotion-3023 to spk220-6.dmzhx.lab.cisco.com
Adding vmkernel to spk220-6.dmzhx.lab.cisco.com
vMotion IP for spk240hxaf-1.dmzhx.lab.cisco.com: 192.168.88.21
Adding vmotion-3023 to spk240hxaf-1.dmzhx.lab.cisco.com
Adding vmkernel to spk240hxaf-1.dmzhx.lab.cisco.com
vMotion IP for spk240hxaf-2.dmzhx.lab.cisco.com: 192.168.88.22
Adding vmotion-3023 to spk240hxaf-2.dmzhx.lab.cisco.com
Adding vmkernel to spk240hxaf-2.dmzhx.lab.cisco.com
vMotion IP for spk240hxaf-3.dmzhx.lab.cisco.com: 192.168.88.23
Adding vmotion-3023 to spk240hxaf-3.dmzhx.lab.cisco.com
Adding vmkernel to spk240hxaf-3.dmzhx.lab.cisco.com
vMotion IP for spk240hxaf-4.dmzhx.lab.cisco.com: 192.168.88.24
Adding vmotion-3023 to spk240hxaf-4.dmzhx.lab.cisco.com
Adding vmkernel to spk240hxaf-4.dmzhx.lab.cisco.com
Add VM network VLANs? (y/n) y
Attempting to find UCSM IP
Found UCSM 192.168.66.110, logging with username admin. Org is SPKHX
UCSM Password:
Port Group Name to add (VLAN ID will be appended to the name): SPK-Data
VLAN ID: (0-4096) 11
Adding VLAN 11 to FI
Adding VLAN 11 to vm-network-a VNIC template
Adding SPK-Data-11 to spk220-1.dmzhx.lab.cisco.com
Adding SPK-Data-11 to spk220-2.dmzhx.lab.cisco.com
Adding SPK-Data-11 to spk220-3.dmzhx.lab.cisco.com
Adding SPK-Data-11 to spk220-4.dmzhx.lab.cisco.com
Adding SPK-Data-11 to spk220-5.dmzhx.lab.cisco.com
Adding SPK-Data-11 to spk220-6.dmzhx.lab.cisco.com
Adding SPK-Data-11 to spk240hxaf-1.dmzhx.lab.cisco.com
Adding SPK-Data-11 to spk240hxaf-2.dmzhx.lab.cisco.com
Adding SPK-Data-11 to spk240hxaf-3.dmzhx.lab.cisco.com
Adding SPK-Data-11 to spk240hxaf-4.dmzhx.lab.cisco.com
Add additional VM network VLANs? (y/n) y
Port Group Name to add (VLAN ID will be appended to the name): SPK-Repl
VLAN ID: (0-4096) 12
Adding VLAN 12 to FI
Adding VLAN 12 to vm-network-a VNIC template
Adding SPK-Repl-12 to spk220-1.dmzhx.lab.cisco.com
Adding SPK-Repl-12 to spk220-2.dmzhx.lab.cisco.com
Adding SPK-Repl-12 to spk220-3.dmzhx.lab.cisco.com
Adding SPK-Repl-12 to spk220-4.dmzhx.lab.cisco.com
Adding SPK-Repl-12 to spk220-5.dmzhx.lab.cisco.com
Adding SPK-Repl-12 to spk220-6.dmzhx.lab.cisco.com
Adding SPK-Repl-12 to spk240hxaf-1.dmzhx.lab.cisco.com
Adding SPK-Repl-12 to spk240hxaf-2.dmzhx.lab.cisco.com
Adding SPK-Repl-12 to spk240hxaf-3.dmzhx.lab.cisco.com
Adding SPK-Repl-12 to spk240hxaf-4.dmzhx.lab.cisco.com
Add additional VM network VLANs? (y/n) n
Run health check? (y/n) n
root@HyperFlex-Installer:~#
```

10. Log into vSphere Web Client and verify that the Splunk Data Network appears on the vswitch.



## Create Splunk Virtual Machine Templates

To create the Splunk virtual machine templates on the Cisco HyperFlex storage cluster, follow these steps:

Table 19 lists the four types of templates in this solution.

Table 19 Splunk Virtual Machine Templates

Name	Type	OS	Splunk Software	vCPU	Memory	Disk1 (OS)	Provision	Disk2 (Data)	Provision
Spktmp	Splunk Base	RedHat Enterprise Linux 7.5	N/A	4	16GB	16GB	Thin	N/A	N/A
Spktmp-1	Splunk Admin	RedHat Enterprise Linux 7.5	Splunk Enterprise 7.2.3	4	16GB	16GB	Thin	48GB	Thin
Spktmp-2	Splunk Search Heads	RedHat Enterprise Linux 7.5	Splunk Enterprise 7.2.3	12	64GB	16GB	Thin	48GB	Thin
Spktmp-3	Splunk Indexer	RedHat Enterprise Linux 7.5	Splunk Enterprise 7.2.3	12	64GB	16GB	Thin	600GB	Thick Eager Zeroed

The high-level creation steps are:

- Step 1: Create a Linux virtual machine as Splunk base template (spktmp) with RedHat Enterprise Linux 7.5 OS;
- Step 2: Post OS installation, do the Splunk required system configurations on spktmp;
- Step 3: Clone from spktmp and modify the virtual machine settings to customize the new template for Splunk administration (spktmp-1). Install Splunk Enterprise software on spktmp-1;
- Step 4: Clone from spktmp and modify the virtual machine settings to customize the new template for Splunk search heads (spktmp-2). Install Splunk Enterprise software on spktmp-2;
- Step 5: Clone from spktmp and modify the virtual machine settings to customize the new template for Splunk Indexer (spktmp-3). Install Splunk Enterprise software on spktmp-3;

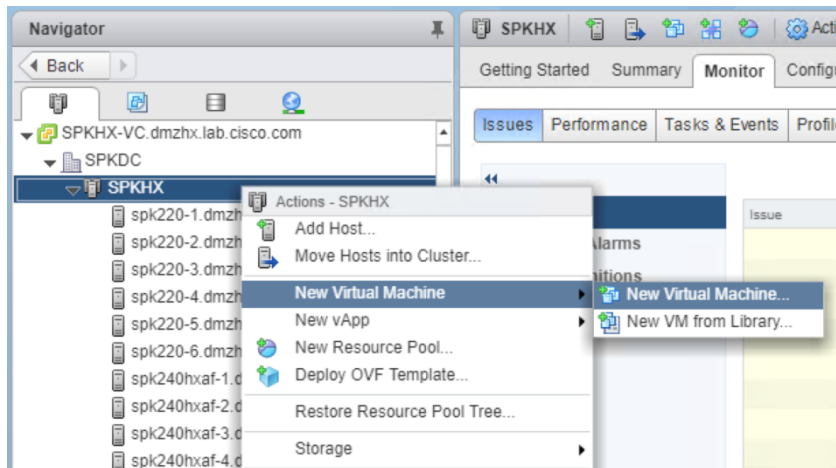
## Create Splunk Base Virtual Machine Template

Table 20 lists the virtual machine – spktmp’s configuration

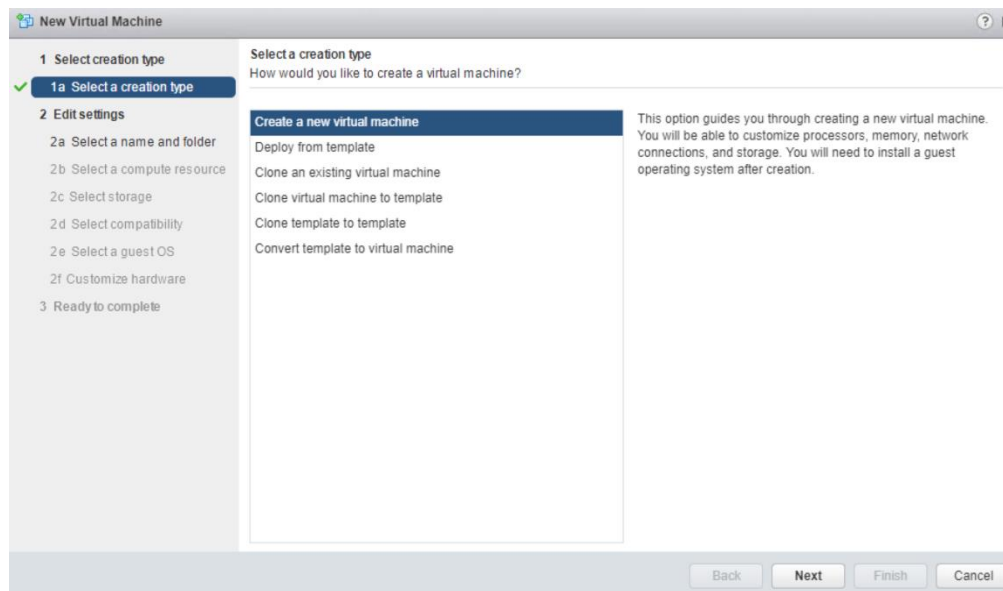
Table 20 Splunk Base Configuration- spktmp

Setting	Value
Virtual Machine Name	spktmp
Virtual Machine Location	1 <sup>st</sup> HX converged node
Compatibility	ESXi 6.5 and later
Operating System	RedHat Enterprise Linux 7.x
vCPUs	4
Memory	16 GB (fully reserved)
Hard Disk Size for OS	16 GB
Provisioning Type	Thin
Number of Data Hard Disks	0
Hard Disk Size for Data	N/A
Provisioning Type	N/A
SCSI Controller #	0
SCSI Controller Type	VMWare Paravirtual
Data Network Type	VMXNET3 (Splunk Data VLAN)

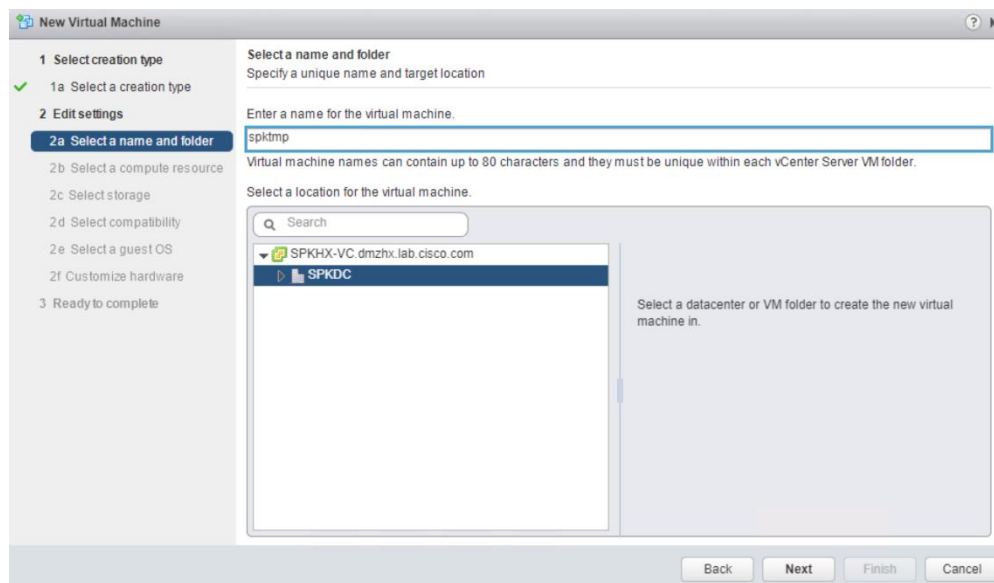
1. Open the vSphere web client, right-click the HyperFlex cluster and select “New Virtual Machine.”



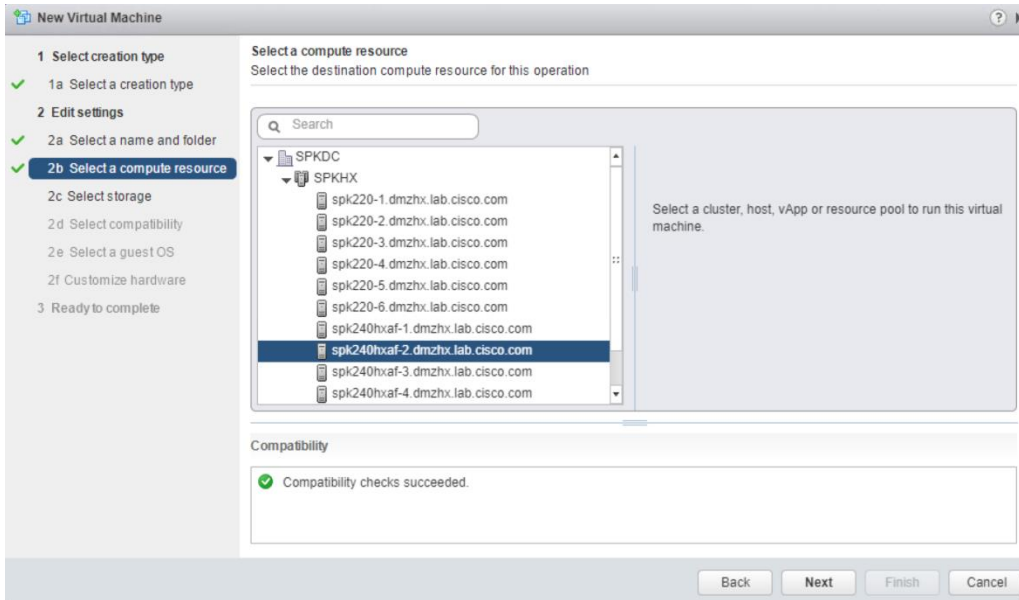
2. Select “Create a new virtual machine”. Click NEXT.



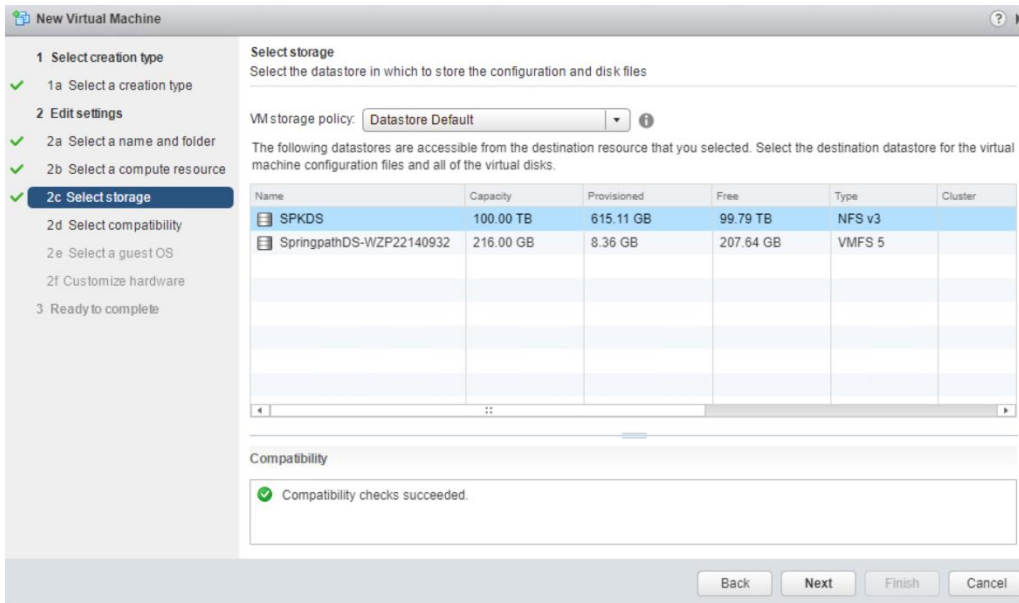
3. Enter spktmp as virtual machine name and click NEXT.



4. Select the first HXAF240c node of the HyperFlex cluster and click NEXT.

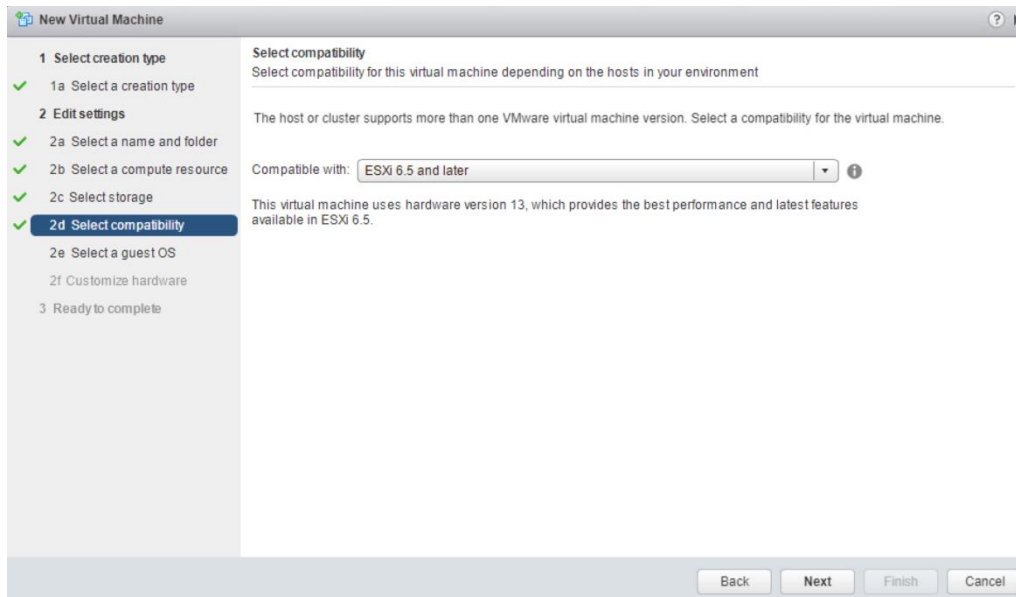


5. Select SPKDS as storage and click NEXT.

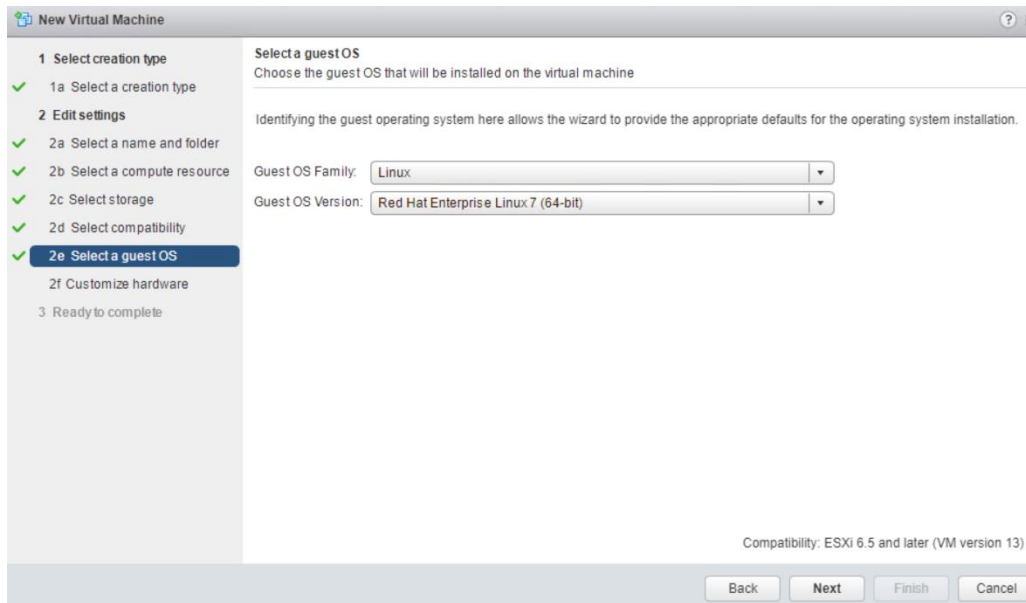


6. Select highest compatibility from list – here it is ESXi 6.5 and later, and click NEXT.

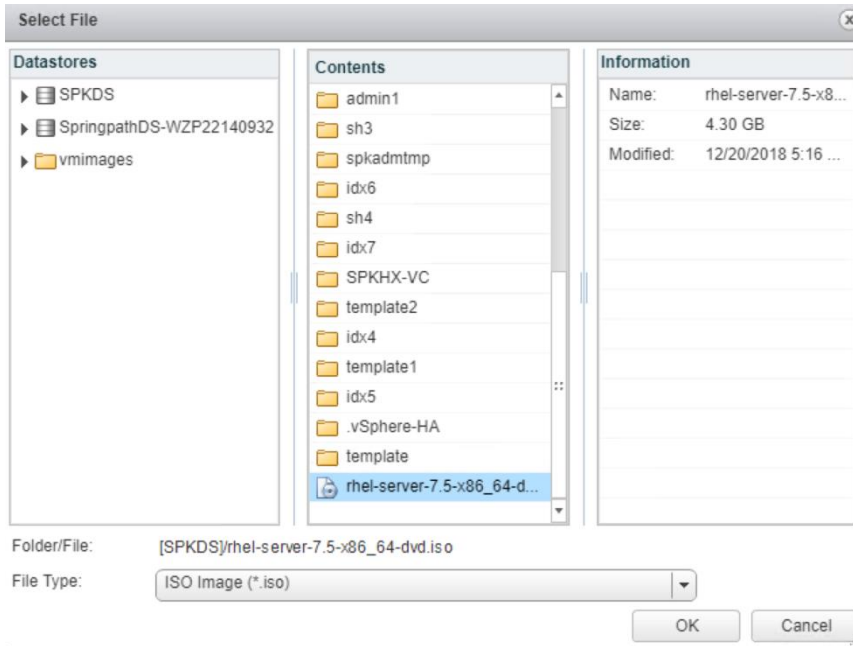




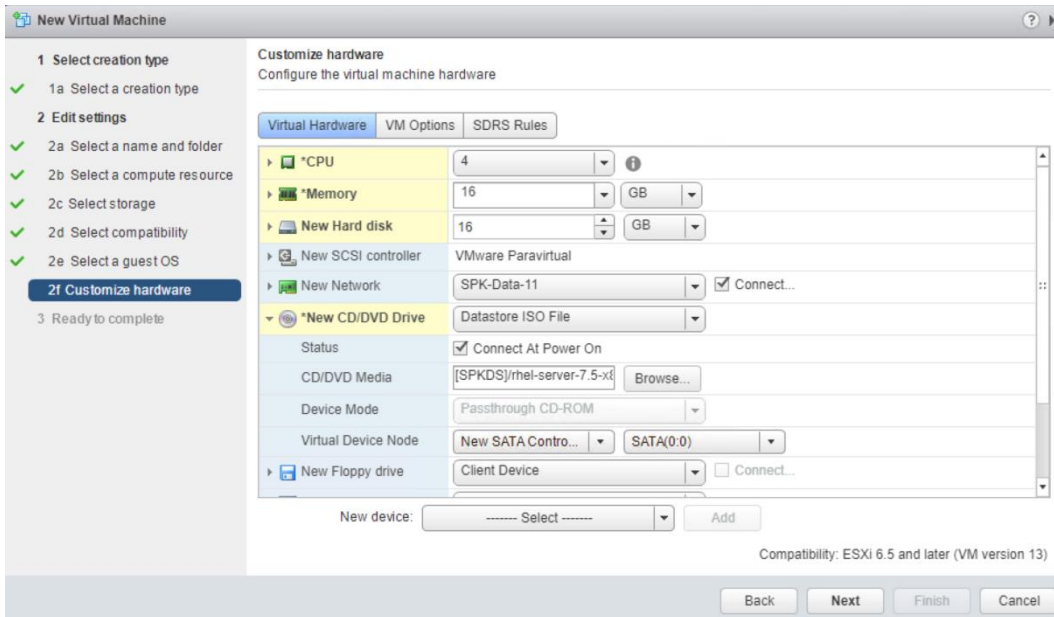
7. Select Red Hat Enterprise Linux 7 (64-Bit) as the Guest OS version and click NEXT.



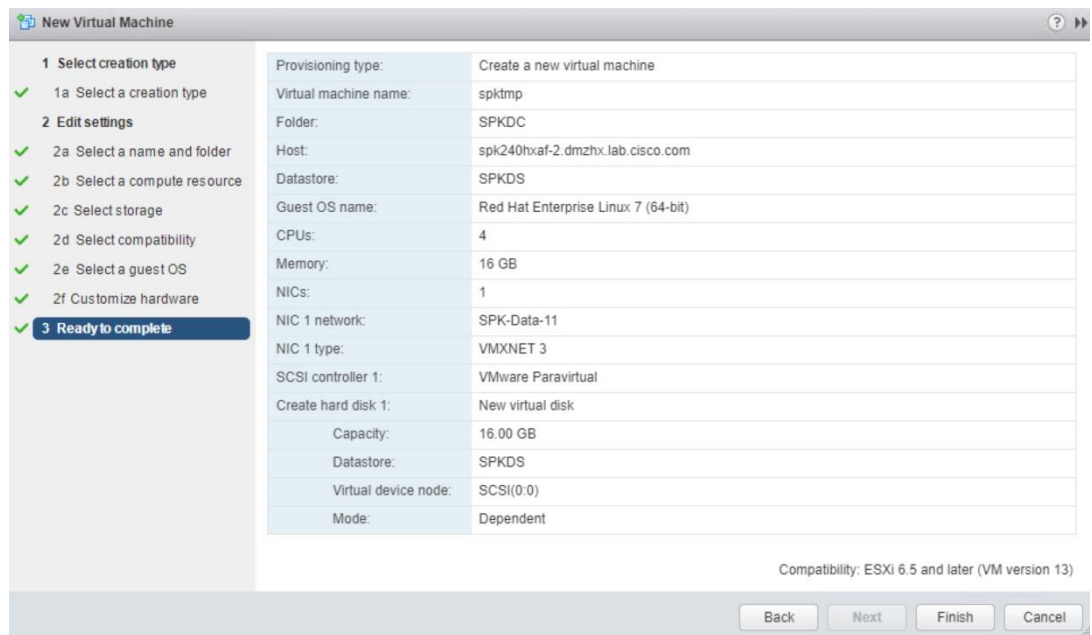
8. Configure the Virtual Machine with 4 vCPUs (either one or two sockets).
9. Configure 16GB of Memory and select the Check-Box for "Reserve all guest memory (All locked)."
10. Configure one hard drive with 16GB capacity, Thin Provision and connected to SCSI Controller 0.
11. Configure one network with Adapter type VMXNET3 connected to the Splunk Data network.
12. Use the RedHat Enterprise Linux 7.5 ISO image for the CD/DVD Media and select "Connect At Power On."



13. Click NEXT.



14. Check the configuration details and click FINISH.

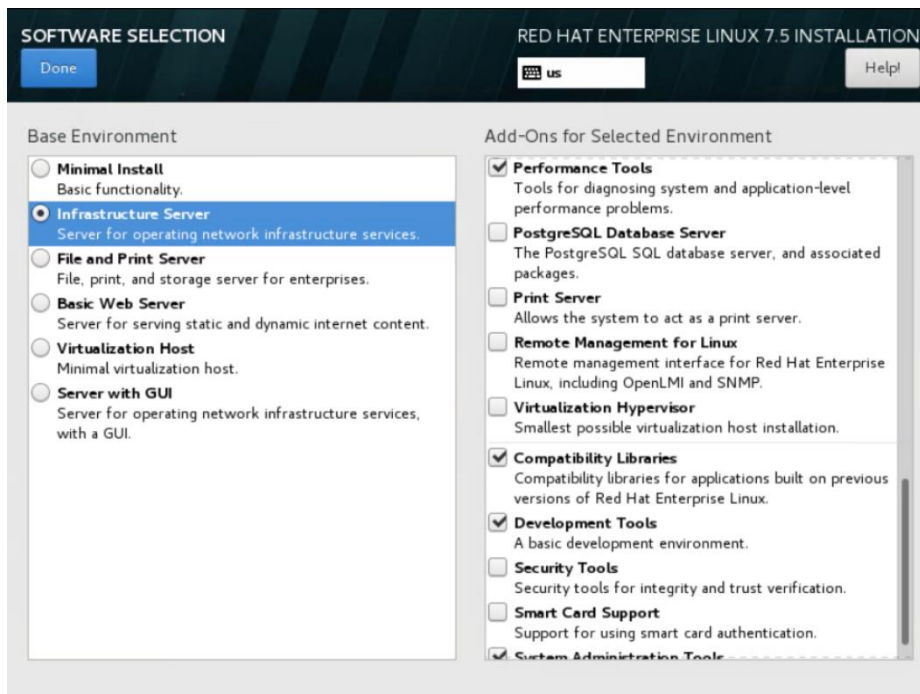


15. Power on the Virtual Machine and open the Console.

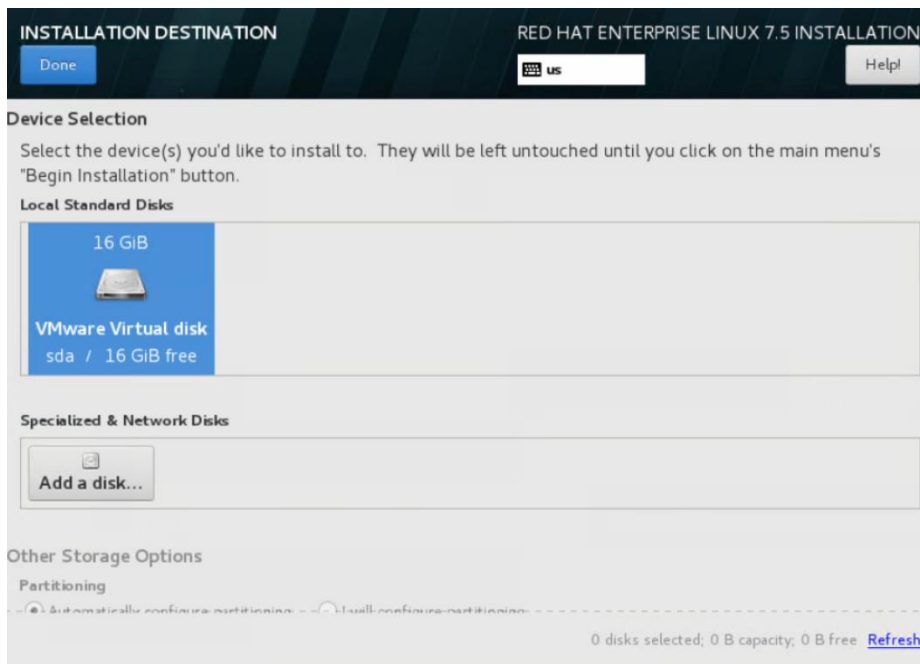
16. Install RedHat Enterprise Linux.



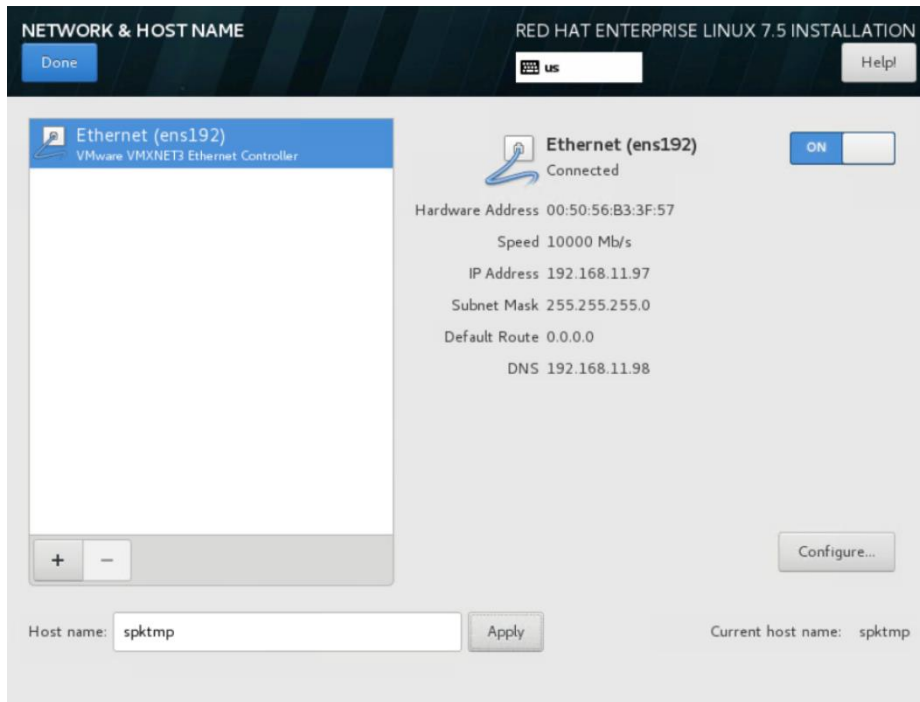
17. Select Infrastructure Server plus Hardware Monitoring Utilities, Performance Tools, Compatibility Libraries, Development Tools and System Administration Tools as Software to install.



18. Select the 16 GB disks to install the OS.

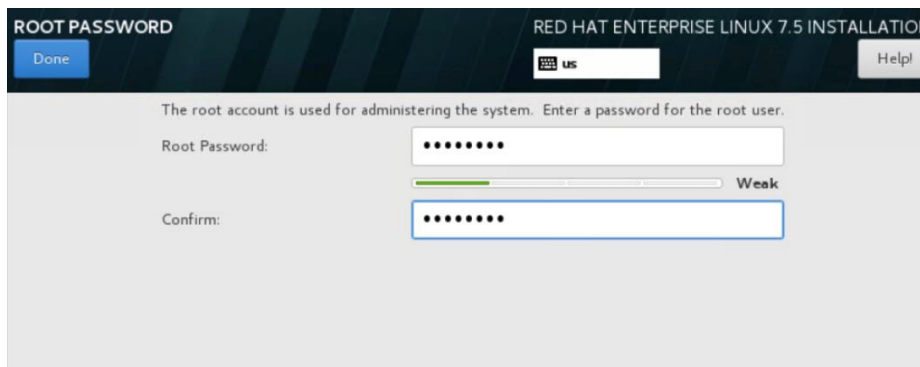


19. Configure a temporary IP address for the network adapter.



20. Start the installation process and wait until completion.

21. Enter the password for user root.



22. Reboot the system.

23. Check that the system boots up and that you can log into the operating system.

## Configure Splunk Base Virtual Machine

### Create and Configure a Linux Virtual Machine as Workstation -admin

Prior to configuring the Splunk base virtual machine, a Linux virtual machine –admin was created as the place to configure and manage the Splunk cluster. The features enabled on this admin virtual machine include, but are not limited to the following:

- Installing RedHat Enterprise Linux 7.5 with the option Server with GUI
- Installing ClusterShell

- Setting up password-less remote login from admin to all Splunk virtual machines
- Installing and enabling httpd service
- Installing and enabling ntpd service
- Creating a local OS repository from Redhat Enterprise 7.5 ISO image
- Installing a web browser, e.g. Mozilla Firefox, to access Splunk management UI
- Installing Splunk Enterprise software that creates a user named splunk. Set the appropriate password for the user splunk.

Table 21 lists the virtual machine – admin’s configuration.

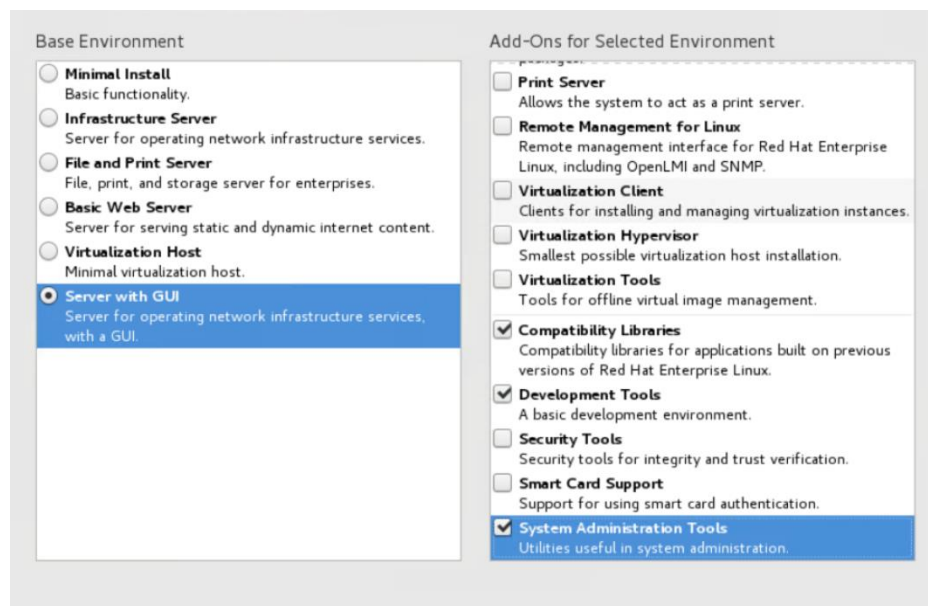
Table 21 Linux Work Station Configuration - admin

Setting	Value
Virtual Machine Name	admin
Virtual Machine Location	2 <sup>nd</sup> HX converged node
Compatibility	ESXi 6.5 and later
Operating System	RedHat Enterprise Linux 7.x
vCPUs	4
Memory	64 GB
Hard Disk Size for OS	128 GB
Provisioning Type	Thin
SCSI Controller #	0
SCSI Controller Type	VMWare Paravirtual
Management Network Type	VMXNET3 (HX Storage Controller Management VLAN)
Data Network Type	VMXNET3 (Splunk Data VLAN)

To create the admin virtual machine, follow these steps:

1. Open the vSphere web client, right-click the HyperFlex cluster and select “New Virtual Machine.”
2. Select “Create a new virtual machine”. Click NEXT.
3. Enter admin as virtual machine name and click NEXT.
4. Select the second one of the HXAF240c nodes of the HyperFlex cluster and click NEXT.
5. Select SPKDS as storage and click NEXT.
6. Select highest compatibility from list – here it is ESXi 6.5 and later, and click NEXT.
7. Select Red Hat Enterprise Linux 7 (64-Bit) as the Guest OS version and click NEXT.
8. Configure the Virtual Machine with 4 vCPUs (either one or two sockets).

9. Configure 64GB of Memory.
10. Configure one hard drive with 128GB capacity, Thin Provision and connected to SCSI Controller 0.
11. Configure one network with Adapter type VMXNET3 connected to the HX Storage Controller Management network.
12. Configure additional network with Adapter type VMXNET3 connected to the Splunk Data network.
13. Use the RedHat Enterprise Linux 7.5 ISO image for the CD/DVD Media and select "Connect At Power On."
14. Click NEXT.
15. Check the configuration details and click FINISH.
16. Power on the Virtual Machine and open the Console.
17. Install RedHat Enterprise Linux.
18. Select Server with GUI plus Hardware Monitoring Utilities, Performance Tools, Compatibility Libraries, Development Tools and System Administration Tools as Software to install.



19. Select the 128 GB disks to install the OS.
20. Start the installation process and wait until completion.
21. Enter the password for user root.
22. Reboot the system.
23. Check that the system boots up and logon to the operating system as root user from the console.
24. Use nmtui to configure the network and hostname of the system.

Hostname	Management Network (VLAN 3021)	Data Network (VLAN 11)
admin	192.168.66.98/24	192.168.11.98/24

```

Profile name eth0
Device ens192 (00:50:56:B0:8F:3F)

= ETHERNET <Show>
= IPv4 CONFIGURATION <Manual> <Hide>
  Addresses 192.168.66.98/24 <Remove>
             <Add...>
  Gateway   <Add...>
  DNS servers 192.168.66.104 <Remove>
             <Add...>
  Search domains <Add...>

  Routing (No custom routes) <Edit...>
  [ ] Never use this network for default route
  [ ] Ignore automatically obtained routes
  [ ] Ignore automatically obtained DNS parameters
  [ ] Require IPv4 addressing for this connection

= IPv6 CONFIGURATION <Ignore> <Show>

[X] Automatically connect
[X] Available to all users

<Cancel> <OK>
    
```

```

Profile name eth1
Device ens224 (00:50:56:B0:F1:B4)

= ETHERNET <Show>
= IPv4 CONFIGURATION <Manual> <Hide>
  Addresses 192.168.11.98/24 <Remove>
             <Add...>
  Gateway   <Add...>
  DNS servers 192.168.11.104 <Remove>
             <Add...>
  Search domains <Add...>

  Routing (No custom routes) <Edit...>
  [ ] Never use this network for default route
  [ ] Ignore automatically obtained routes
  [ ] Ignore automatically obtained DNS parameters
  [ ] Require IPv4 addressing for this connection

= IPv6 CONFIGURATION <Ignore> <Show>

[X] Automatically connect
[X] Available to all users

<Cancel> <OK>
    
```

```

Set Hostname

Hostname admin

<Cancel> <OK>
    
```





---

Although the steps to create a new Linux work station is described here for this solution, use of any existing Linux server that has access to the Splunk data network as work machine is supported.

---

### Configuring /etc/hosts

To setup /etc/hosts on the admin node and then copy the file to the spktmp virtual machine template, follow these steps:

1. Remote Log into the admin virtual machine: ssh [root@192.168.66.98](mailto:root@192.168.66.98)
2. Edit the host file /etc/hosts with the following IP addresses and corresponding hostnames:

On admin VM:

```
[root@admin ~]# vi /etc/hosts
[root@admin ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain
::1        localhost localhost.localdomain
192.168.11.52 admin1
192.168.11.53 admin2
192.168.11.54 admin3
192.168.11.55 admin4
192.168.11.56 admin5
192.168.11.57 admin6
192.168.11.58 sh1
192.168.11.59 sh2
192.168.11.60 sh3
192.168.11.61 sh4
192.168.11.62 idx1
192.168.11.63 idx2
192.168.11.64 idx3
192.168.11.65 idx4
192.168.11.66 idx5
192.168.11.67 idx6
192.168.11.68 idx7
192.168.11.69 idx8
192.168.11.70 idx9
192.168.11.71 idx10
192.168.11.72 idx11
192.168.11.73 idx12
192.168.11.74 idx13
192.168.11.75 idx14
192.168.11.76 idx15
192.168.11.77 idx16
192.168.11.78 idx17
192.168.11.79 idx18
192.168.11.80 idx19
192.168.11.81 idx20
192.168.11.98 admin
192.168.11.97 spktmp
```

3. On admin, copy the file /etc/hosts to the remote host – spktmp.

```
[root@admin .ssh]# scp /etc/hosts root@spktmp://etc/.
The authenticity of host 'spktmp (192.168.11.97)' can't be established.
ECDSA key fingerprint is SHA256:Ch/Ba7ov+jfV0QkmxM+Bf7qNSkTn7DroXc9q47c5NAI.
ECDSA key fingerprint is MD5:ed:43:d9:37:09:d2:be:8d:16:60:d2:e2:34:90:72:b6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'spktmp,192.168.11.97' (ECDSA) to the list of known hosts.
hosts
100% 797 3.2MB/s 00:00
[root@admin .ssh]# █
```

4. Login as the root user to the Splunk base virtual machine – spktmp: ssh [root@192.168.11.97](mailto:root@192.168.11.97)
5. On spktmp, verify that /etc/hosts got copied and has the identical contents.

On spktmp virtual machine:

```
[root@spktmp ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain
::1 localhost localhost.localdomain
192.168.11.52 admin1
192.168.11.53 admin2
192.168.11.54 admin3
192.168.11.55 admin4
192.168.11.56 admin5
192.168.11.57 admin6
192.168.11.58 sh1
192.168.11.59 sh2
192.168.11.60 sh3
192.168.11.61 sh4
192.168.11.62 idx1
192.168.11.63 idx2
192.168.11.64 idx3
192.168.11.65 idx4
192.168.11.66 idx5
192.168.11.67 idx6
192.168.11.68 idx7
192.168.11.69 idx8
192.168.11.70 idx9
192.168.11.71 idx10
192.168.11.72 idx11
192.168.11.73 idx12
192.168.11.74 idx13
192.168.11.75 idx14
192.168.11.76 idx15
192.168.11.77 idx16
192.168.11.78 idx17
192.168.11.79 idx18
192.168.11.80 idx19
192.168.11.81 idx20
192.168.11.98 admin
192.168.11.97 spktmp
```

### Setup Passwordless Login

To manage all of the Splunk nodes from the admin virtual machine you need to setup a passwordless login. It assists in automating common tasks with ClusterShell; a cluster-wide parallel shell command utility, and shell-scripts without having to use passwords.

To enable passwordless login from admin to spktmp, follow these steps:

1. Remote log into the admin virtual machine: ssh root@192.168.66.98
2. Run the ssh-keygen command to create both public and private keys on the admin node.

```
[root@admin ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:b3r00TK9JSnPLqS2zGz8Kdhh7Y0eN8yqDCDy5cWJ+x4 root@localhost.localdomain
The key's randomart image is:
+---[RSA 2048]----+
|
|   o .
|  o . o + S.
|.o + o o.+ .
| . + E= X+oo..
|   +.=0+%=o
|   .==XB0B=o
+-----[SHA256]-----+
```

3. Run the `ssh-copy-id` command from the admin node to copy the public key `id_rsa.pub` to the `spktmp` virtual machine template. It appends the keys to the remote-host's `.ssh/authorized_key`.
4. On admin, login as the root user to the Splunk base virtual machine – `spktmp` using the hostname and IP address. Verify that the password-less login succeeds.

```
[root@admin ~]# ssh-copy-id -i .ssh/id_rsa.pub -o StrictHostKeyChecking=no spktmp
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@spktmp's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh -o 'StrictHostKeyChecking=no' 'spktmp'"
and check to make sure that only the key(s) you wanted were added.

[root@admin ~]# ssh spktmp
[root@spktmp ~]# exit
logout
Connection to spktmp closed.
[root@admin ~]# ssh 192.168.11.97
Last login: Fri Feb 15 15:02:37 2019 from 192.168.11.98
[root@spktmp ~]# exit
logout
Connection to 192.168.11.97 closed.
[root@admin ~]#
```

## Setup ClusterShell

ClusterShell (or clush) is a cluster-wide shell to run commands on several hosts in parallel. To setup a clustershell, follow these steps:

1. Download the file of `python2-clustershell-1.8.1-1.el7.noarch.rpm` from the internet: <https://pkgs.org/download/python2-clustershell>
2. Download the file of `clustershell-1.8.1-1.el7.noarch.rpm` from the internet: [https://centos.pkgs.org/7/epel-x86\\_64/clustershell-1.8.1-1.el7.noarch.rpm.html](https://centos.pkgs.org/7/epel-x86_64/clustershell-1.8.1-1.el7.noarch.rpm.html)
3. Install `python2` for clustershell on admin server:

```
[root@admin ~]# yum -y install python2-clustershell-1.8.1-1.el7.noarch.rpm
```

```
[root@admin ~]# yum -y install python2-clustershell-1.8.1-1.el7.noarch.rpm
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
Examining python2-clustershell-1.8.1-1.el7.noarch.rpm: python2-clustershell-1.8.1-1.el7.noarch
Marking python2-clustershell-1.8.1-1.el7.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package python2-clustershell.noarch 0:1.8.1-1.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

-----
Package Arch Version Repository Size
-----
Installing:
python2-clustershell noarch 1.8.1-1.el7 /python2-clustershell-1.8.1-1.el7.noarch 1.4 M
-----
Transaction Summary
-----
Install 1 Package

Total size: 1.4 M
Installed size: 1.4 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : python2-clustershell-1.8.1-1.el7.noarch 1/1
  Verifying : python2-clustershell-1.8.1-1.el7.noarch 1/1

Installed:
  python2-clustershell.noarch 0:1.8.1-1.el7

Complete!
```

4. Install clustershell on admin server:

```
[root@admin ~]# yum -y install clustershell-1.8.1-1.el7.noarch.rpm

[root@admin ~]# yum -y install clustershell-1.8.1-1.el7.noarch.rpm
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
Examining clustershell-1.8.1-1.el7.noarch.rpm: clustershell-1.8.1-1.el7.noarch
Marking clustershell-1.8.1-1.el7.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package clustershell.noarch 0:1.8.1-1.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

-----
Package Arch Version Repository Size
-----
Installing:
clustershell noarch 1.8.1-1.el7 /clustershell-1.8.1-1.el7.noarch 295 k
-----
Transaction Summary
-----
Install 1 Package

Total size: 295 k
Installed size: 295 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : clustershell-1.8.1-1.el7.noarch 1/1
  Verifying : clustershell-1.8.1-1.el7.noarch 1/1

Installed:
  clustershell.noarch 0:1.8.1-1.el7

Complete!
```

5. Edit /etc/clustershell/groups file to pre-define hostnames for all the nodes of the Splunk cluster. Create three special groups [admin/searchhead/indexer] besides the group that takes all the hosts of the cluster. These groups help target the cluster wide commands to a specific set of nodes grouped by their role in the Splunk deployment.

On admin virtual machine:

```
[root@admin ~]# vi /etc/clustershell/groups
[root@admin ~]# cat /etc/clustershell/groups
spkadm: admin[1-6]
spkidx: idx[1-10]
spksh: sh[1-4]
all: admin[1-6],sh[1-4],idx[1-10]
[root@admin ~]#

[root@admin ~]# vi /etc/clustershell/groups
[root@admin ~]# cat /etc/clustershell/groups
spkadm: admin[1-6]
spkidx: idx[1-10]
spksh: sh[1-4]
all: admin[1-6],sh[1-4],idx[1-10]
[root@admin ~]#
[root@admin ~]#
```



For more information and documentation about ClusterShell, see <https://github.com/cea-hpc/clustershell/wiki/UserAndProgrammingGuide>.

### Create Red Hat Enterprise Linux (RHEL) 7.5 Local Repository

To create a repository using RHEL DVD or ISO on the work station (in this deployment admin is used for this purpose), create a directory with all the required RPMs, run the createrepo command and then publish the resulting repository.

1. Log into admin, create a directory that would contain the repository:

```
[root@admin ~]# mkdir -p /var/www/html/rhelrepo
```

2. Copy the contents of the Red Hat DVD to /var/www/html/rhelrepo:



It is assumed you have the Red Hat ISO file located in your present working directory.

```
[root@admin ~]# mkdir -p /mnt/rheliso
[root@admin ~]# mount -t iso9660 -o loop /root/rhel-server-7.5-x86_64-dvd.iso
/mnt/rheliso/
```

3. Copy the contents of the ISO to the /var/www/html/rhelrepo directory:

```
[root@admin ~]# cp -r /mnt/rheliso/* /var/www/html/rhelrepo
```

```
[root@admin ~]# mkdir -p /var/www/html/rhelrepo
[root@admin ~]# mkdir -p /mnt/rheliso
[root@admin ~]# mount -t iso9660 -o loop /root/rhel-server-7.5-x86_64-dvd.iso /mnt/rheliso/
mount: /dev/loop0 is write-protected, mounting read-only
[root@admin ~]# cp -r /mnt/rheliso/* /var/www/html/rhelrepo
```

4. On admin, create a .repo file to enable the use of the yum command:

On admin virtual machine:

```
[root@admin ~]# vi /var/www/html/rhelrepo/rheliso.repo
[root@admin ~]# cat var/www/html/rhelrepo/rheliso.repo
[rhel7.5]
name=Red Hat Enterprise Linux 7.5
baseurl=http://192.168.11.98/rhelrepo
gpgcheck=0
enabled=1
```

```
[splunk@admin ~]$ vi /var/www/html/rhelrepo/rheliso.repo
[splunk@admin ~]$ cat /var/www/html/rhelrepo/rheliso.repo
[rhel7.5]
name=Red Hat Enterprise Linux 7.5
baseurl=http://192.168.11.98/rhelrepo
gpgcheck=0
enabled=1
[splunk@admin ~]$
```

5. Copy the rheliso.repo file from /var/www/html/rhelrepo to /etc/yum.repos.d on admin:

```
[root@admin ~]# cp /var/www/html/rhelrepo/rheliso.repo /etc/yum.repos.d/
```

6. Copy the rheliso.repo to spktmp:

```
[root@admin ~]# scp /var/www/html/rhelrepo/rheliso.repo root@spktmp:/etc/yum.repos.d/.
```

```
[root@admin ~]# cp /var/www/html/rhelrepo/rheliso.repo /etc/yum.repos.d/
[root@admin ~]# ls /etc/yum.repos.d/
redhat.repo  rheliso.repo
[root@admin ~]# scp /var/www/html/rhelrepo/rheliso.repo root@spktmp:/etc/yum.repos.d/.
rheliso.repo                                100% 101 103.7KB/s 00:00
```



Based on this repo file, yum requires httpd to be running on admin for other nodes to access the repository.

- To make use of repository files on admin without httpd, edit the baseurl of repo file /etc/yum.repos.d/rheliso.repo to point repository location in the file system.



This step is needed to install software on admin using the repo (such as httpd, createrepo, and so on).

```
[root@admin ~]# vi /etc/yum.repos.d/rheliso.repo
[root@admin ~]# cat /etc/yum.repos.d/rheliso.repo
[rhel7.5]
name=Red Hat Enterprise Linux 7.5
baseurl=file:///var/www/html/rhelrepo
gpgcheck=0
enabled=1
```

```
[root@admin ~]# vi /etc/yum.repos.d/rheliso.repo
[root@admin ~]# cat /etc/yum.repos.d/rheliso.repo
[rhel7.5]
name=Red Hat Enterprise Linux 7.5
baseurl=file:///var/www/html/rhelrepo
gpgcheck=0
enabled=1
[root@admin ~]#
```

- Creating the Red Hat Repository Database. Install the createrepo package on admin. Use it to regenerate the repository database(s) for the local copy of the RHEL DVD contents.

```
[root@admin ~]# yum -y install createrepo
```

```
[root@admin ~]# yum -y install createrepo
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
rhel7.5
(1/2): rhel7.5/group_gz | 4.3 kB 00:00:00
(2/2): rhel7.5/primary_db | 145 kB 00:00:00
Package createrepo-0.9.9-28.el7.noarch already installed and latest version
Nothing to do
```

- Run createrepo on the RHEL repository to create the repo database on admin node:

```
[root@admin ~]# cd /var/www/html/rhelrepo
[root@admin rhelrepo]# createrepo
```

```
[root@admin ~]# cd /var/www/html/rhelrepo
[root@admin rhelrepo]# createrepo .
Spawning worker 0 with 1300 pkgs
Spawning worker 1 with 1300 pkgs
Spawning worker 2 with 1299 pkgs
Spawning worker 3 with 1299 pkgs
Workers Finished
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete
```

- Purge the yum caches after httpd is installed (see the following section Install httpd).

## Install httpd

Setting up RHEL repository on the admin node requires httpd. To install httpd, follow these steps:

1. Install httpd on the admin node to host repositories. The Red Hat repository is hosted using HTTP on the admin node, this machine is accessible by all the hosts in the cluster.

```
[root@admin]# yum -y install httpd
```

```
[root@admin rhelrepo]# yum -y install httpd
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.6-80.el7 will be installed
--> Processing Dependency: httpd-tools = 2.4.6-80.el7 for package: httpd-2.4.6-80.el7.x86_64
--> Running transaction check
--> Package httpd-tools.x86_64 0:2.4.6-80.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================================================================
 Package                               Arch                               Version                               Repository                             Size
=====================================================================================================================================
Installing:
 httpd                                  x86_64                              2.4.6-80.el7                           rhel7.5                                 1.2 M
Installing for dependencies:
 httpd-tools                             x86_64                              2.4.6-80.el7                           rhel7.5                                 89 k
=====================================================================================================================================

Transaction Summary
-----
Install 1 Package (+1 Dependent package)
Total download size: 1.3 M
Installed size: 3.9 M
Downloading packages:
-----
Total                                                                           238 MB/s | 1.3 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : httpd-tools-2.4.6-80.el7.x86_64                                     1/2
  Installing : httpd-2.4.6-80.el7.x86_64                                         2/2
rhel7.5/productid | 1.6 kB  00:00:00
  Verifying  : httpd-2.4.6-80.el7.x86_64                                         1/2
  Verifying  : httpd-tools-2.4.6-80.el7.x86_64                                   2/2

Installed:
  httpd.x86_64 0:2.4.6-80.el7

Dependency Installed:
  httpd-tools.x86_64 0:2.4.6-80.el7

Complete!
```

2. Add ServerName and make the necessary changes to the server configuration file.

```
[root@admin rhelrepo]# vi /etc/httpd/conf/httpd.conf
[root@admin rhelrepo]# cat /etc/httpd/conf/httpd.conf | grep ServerName
# ServerName gives the name and port that the server uses to identify itself.
ServerName 192.168.11.98:80
```

```
[root@admin rhelrepo]# vi /etc/httpd/conf/httpd.conf
[root@admin rhelrepo]# cat /etc/httpd/conf/httpd.conf | grep ServerName
# ServerName gives the name and port that the server uses to identify itself.
ServerName 192.168.11.98:80
```

3. Start httpd:

```
[root@admin rhelrepo]# /bin/systemctl start httpd
[root@admin rhelrepo]# /bin/systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
```

```
[root@admin rhelrepo]# /bin/systemctl start httpd
[root@admin rhelrepo]# /bin/systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@admin rhelrepo]# /bin/systemctl status httpd
? httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2019-01-29 13:55:58 PST; 13s ago
     Docs: man:httpd(8)
           man:apachectl(8)
   Main PID: 30614 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic: 0 B/sec"
   CGroup: /system.slice/httpd.service
           +-30614 /usr/sbin/httpd -DFOREGROUND
           +-30615 /usr/sbin/httpd -DFOREGROUND
           +-30616 /usr/sbin/httpd -DFOREGROUND
           +-30617 /usr/sbin/httpd -DFOREGROUND
           +-30618 /usr/sbin/httpd -DFOREGROUND
           +-30619 /usr/sbin/httpd -DFOREGROUND

Jan 29 13:55:58 admin systemd[1]: Starting The Apache HTTP Server...
Jan 29 13:55:58 admin systemd[1]: Started The Apache HTTP Server.
```

- Purge the yum caches after httpd is installed and check the repository list on admin (step followed from section Setup Red Hat Repository):

```
[root@admin rhelrepo]# yum clean all
[root@admin rhelrepo]# rm -rf /var/cache/yum
[root@admin rhelrepo]# yum repolist
```

```
[root@admin rhelrepo]# yum clean all
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
Cleaning repos: rhel7.5
Cleaning up everything
Maybe you want: rm -rf /var/cache/yum, to also free up space taken by orphaned data from disabled or removed repos
[root@admin rhelrepo]# rm -rf /var/cache/yum
[root@admin rhelrepo]# yum repolist
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
rhel7.5 | 2.9 kB 00:00:00
rhel7.5/primary_db | 4.2 MB 00:00:00
repo id          repo name          status
rhel7.5          Red Hat Enterprise Linux 7.5 5,198
```

- Log into spktmp, purge the yum caches after httpd is installed and check the repository list on spktmp:

```
[root@spktmp]# yum clean all
[root@spktmp]# rm -rf /var/cache/yum
[root@spktmp]# yum repolist
```

```
[root@admin rhelrepo]# ssh root@spktmp
Last login: Tue Jan 29 13:58:03 2019 from admin
[root@spktmp ~]# yum clean all
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
Cleaning repos: rhel7.5
Cleaning up everything
Maybe you want: rm -rf /var/cache/yum, to also free up space taken by orphaned data from disabled or removed repos
[root@spktmp ~]# rm -rf /var/cache/yum
[root@spktmp ~]# yum repolist
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
rhel7.5 | 2.9 kB 00:00:00
rhel7.5/primary_db | 4.2 MB 00:00:00
repo id          repo name          status
rhel7.5          Red Hat Enterprise Linux 7.5 5,198
repolist: 5,198
```



While the suggested configuration is to disable SELinux as shown below, if for any reason SELinux needs to be enabled on the cluster, then run `chcon -R -t httpd_sys_content_t /var/www/html/` to make sure that the httpd is able to read the Yum repofiles.

## NTP Configuration

The Network Time Protocol (NTP) is used to synchronize the time of all the nodes within the cluster. The Network Time Protocol daemon (ntpd) sets and maintains the system time of day in synchronism with the timeserver. Configuring NTP is critical for any clustered applications.

Using a private timeserver keeps your cluster synchronized even when an outside NTP server is inaccessible although outside NTP server is also supported.

To install a private NTP server on the admin server, follow these steps:

- On admin virtual machine, configure `/etc/ntp.conf` on with the following contents:

```
[root@admin ~]# vi /etc/ntp.conf
[root@admin ~]# cat /etc/ntp.conf
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
```



```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

```
[root@admin ~]# vi /etc/ntp.conf
[root@admin ~]# cat /etc/ntp.conf
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
server 127.127.1.0
fudge 127.127.1.0 stratum 10
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

- On the spktmp virtual machine, configure /etc/ntp.conf with the following contents by adding admin as NTP server:

```
[root@spktmp ~]# vi /etc/ntp.conf
[root@spktmp ~]# cat /etc/ntp.conf
server 192.168.11.98
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

```
[root@admin ~]# ssh spktmp
Last login: Tue Apr 2 20:17:38 2019 from admin
[root@spktmp ~]# vi /etc/ntp.conf
[root@spktmp ~]# cat /etc/ntp.conf
server 192.168.11.98
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
[root@spktmp ~]#
```

- Install the NTP service on the admin virtual machine:

```
[root@admin ~]# yum -y install ntp
[root@admin ~]# yum -y install ntp
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
Resolving Dependencies
--> Running transaction check
--> Package ntp.x86_64 0:4.2.6p5-28.el7 will be installed
--> Processing Dependency: libopts.so.25()(64bit) for package: ntp-4.2.6p5-28.el7.x86_64
--> Running transaction check
--> Package autogen-libopts.x86_64 0:5.18-5.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================================================================
Package Arch Version Repository Size
=====================================================================================================================================
Installing:
ntp x86_64 4.2.6p5-28.el7 rhel7.5 549 k
Installing for dependencies:
autogen-libopts x86_64 5.18-5.el7 rhel7.5 66 k
Transaction Summary
-----
Install 1 Package (+1 Dependent package)

Total download size: 615 k
Installed size: 1.5 M
Downloading packages:
-----
Total 140 MB/s | 615 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : autogen-libopts-5.18-5.el7.x86_64 1/2
Installing : ntp-4.2.6p5-28.el7.x86_64 2/2
warning: /etc/ntp.conf created as /etc/ntp.conf.rpmnew
Verifying : autogen-libopts-5.18-5.el7.x86_64 1/2
Verifying : ntp-4.2.6p5-28.el7.x86_64 2/2

Installed:
ntp.x86_64 0:4.2.6p5-28.el7

Dependency Installed:
autogen-libopts.x86_64 0:5.18-5.el7

Complete!
```

- Start the NTP service on the admin virtual machine:

```
[root@admin ~]# /bin/systemctl start ntpd.service
```

```
[root@admin ~]# /bin/systemctl enable ntpd.service
[root@admin ~]# /bin/systemctl start ntpd.service
[root@admin ~]# /bin/systemctl enable ntpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to /usr/lib/systemd/system/ntpd.service.
[root@admin ~]# /bin/systemctl status ntpd.service
? ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2019-01-29 14:34:27 PST; 15s ago
 Main PID: 31629 (ntpd)
   CGroup: /system.slice/ntpd.service
           +-31629 /usr/sbin/ntpd -u ntp:ntp -g

Jan 29 14:34:27 admin ntpd[31629]: Listen normally on 2 lo 127.0.0.1 UDP 123
Jan 29 14:34:27 admin ntpd[31629]: Listen normally on 3 ens192 192.168.66.98 UDP 123
Jan 29 14:34:27 admin ntpd[31629]: Listen normally on 4 ens224 192.168.11.98 UDP 123
Jan 29 14:34:27 admin ntpd[31629]: Listen normally on 5 virbr0 192.168.122.1 UDP 123
Jan 29 14:34:27 admin ntpd[31629]: Listening on routing socket on fd #22 for interface updates
Jan 29 14:34:27 admin ntpd[31629]: 0.0.0.0 c016 06 restart
Jan 29 14:34:27 admin ntpd[31629]: 0.0.0.0 c012 02 freq_set kernel 0.000 PPM
Jan 29 14:34:27 admin ntpd[31629]: 0.0.0.0 c011 01 freq_not_set
Jan 29 14:34:27 admin systemd[1]: Started Network Time Service.
Jan 29 14:34:28 admin ntpd[31629]: 0.0.0.0 c514 04 freq_mode
```

5. Install the NTP service on the spktmp virtual machine:

```
[root@spktmp ~]# yum -y install ntp
[root@spktmp ~]# yum -y install ntp
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
Resolving Dependencies
--> Running transaction check
--> Package ntp.x86_64 0:4.2.6p5-28.el7 will be installed
--> Processing Dependency: ntpdate = 4.2.6p5-28.el7 for package: ntp-4.2.6p5-28.el7.x86_64
--> Processing Dependency: libopts.so.25()(64bit) for package: ntp-4.2.6p5-28.el7.x86_64
--> Running transaction check
--> Package autogen-libopts.x86_64 0:5.18-5.el7 will be installed
--> Package ntpdate.x86_64 0:4.2.6p5-28.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

-----
Package Arch Version Repository Size
-----
Installing:
ntp x86_64 4.2.6p5-28.el7 rhel7.5 549 k
Installing for dependencies:
autogen-libopts x86_64 5.18-5.el7 rhel7.5 86 k
ntpdate x86_64 4.2.6p5-28.el7 rhel7.5 86 k
-----
Transaction Summary
-----
Install 1 Package (+2 Dependent packages)

Total download size: 701 k
Installed size: 1.6 M
Downloading packages:
(1/3): autogen-libopts-5.18-5.el7.x86_64.rpm | 86 kB 00:00:00
(2/3): ntpdate-4.2.6p5-28.el7.x86_64.rpm | 86 kB 00:00:00
(3/3): ntp-4.2.6p5-28.el7.x86_64.rpm | 549 kB 00:00:00
-----
Total 14 MB/s | 701 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : autogen-libopts-5.18-5.el7.x86_64 1/3
Installing : ntpdate-4.2.6p5-28.el7.x86_64 2/3
Installing : ntp-4.2.6p5-28.el7.x86_64 3/3
warning: /etc/ntp.conf created as /etc/ntp.conf.rpmnew
Verifying : ntpdate-4.2.6p5-28.el7.x86_64 1/3
Verifying : ntp-4.2.6p5-28.el7.x86_64 2/3
Verifying : autogen-libopts-5.18-5.el7.x86_64 3/3

Installed:
ntp.x86_64 0:4.2.6p5-28.el7

Dependency Installed:
autogen-libopts.x86_64 0:5.18-5.el7 ntpdate.x86_64 0:4.2.6p5-28.el7

Complete!
```

6. Start the NTP service on the spktmp virtual machine:

```
[root@ spktmp ~]# /bin/systemctl start ntpd.service
[root@ spktmp ~]# /bin/systemctl enable ntpd.service
```

```
[root@spktmp ~]# /bin/systemctl start ntpd.service
[root@spktmp ~]# /bin/systemctl enable ntpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to /usr/lib/systemd/system/ntpd.service.
[root@spktmp ~]# /bin/systemctl status ntpd.service
? ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2019-01-29 14:36:39 PST; 16s ago
   Main PID: 1669 (ntpd)
   CGroup: /system.slice/ntpd.service
           +-1669 /usr/sbin/ntpd -u ntp:ntp -g

Jan 29 14:36:39 spktmp ntpd[1669]: ntp_io: estimated max descriptors: 1024, initial socket boundary: 16
Jan 29 14:36:39 spktmp ntpd[1669]: Listen and drop on 0 v4wildcard 0.0.0.0 UDP 123
Jan 29 14:36:39 spktmp ntpd[1669]: Listen and drop on 1 v6wildcard :: UDP 123
Jan 29 14:36:39 spktmp ntpd[1669]: Listen normally on 2 lo 127.0.0.1 UDP 123
Jan 29 14:36:39 spktmp ntpd[1669]: Listen normally on 3 ens192 192.168.11.97 UDP 123
Jan 29 14:36:39 spktmp ntpd[1669]: Listen normally on 4 ens192 fe80::47f4:c80e:f24:5e89 UDP 123
Jan 29 14:36:39 spktmp ntpd[1669]: Listening on routing socket on fd #21 for interface updates
Jan 29 14:36:39 spktmp ntpd[1669]: 0.0.0.0 c016 06 restart
Jan 29 14:36:39 spktmp ntpd[1669]: 0.0.0.0 c012 02 freq_set kernel 0.000 PPM
Jan 29 14:36:39 spktmp ntpd[1669]: 0.0.0.0 c011 01 freq_not_set
```

7. Run the following commands to synchronize the time and restart NTP daemon on spktmp virtual machine:

```
[root@spktmp ~]# /bin/systemctl stop ntpd.service
[root@spktmp ~]# ntpdate 192.168.11.98
[root@spktmp ~]# /bin/systemctl start ntpd.service
```

```
[root@spktmp ~]# /bin/systemctl stop ntpd.service
[root@spktmp ~]# ntpdate 192.168.11.98
29 Jan 14:37:56 ntpdate[1699]: step time server 192.168.11.98 offset 14.826852 sec
[root@spktmp ~]# /bin/systemctl start ntpd.service
[root@spktmp ~]# /bin/systemctl status ntpd.service
? ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2019-01-29 14:38:09 PST; 7s ago
   Process: 1706 ExecStart=/usr/sbin/ntpd -u ntp:ntp $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 1707 (ntpd)
   CGroup: /system.slice/ntpd.service
           +-1707 /usr/sbin/ntpd -u ntp:ntp -g

Jan 29 14:38:09 spktmp ntpd[1707]: ntp_io: estimated max descriptors: 1024, initial socket boundary: 16
Jan 29 14:38:09 spktmp ntpd[1707]: Listen and drop on 0 v4wildcard 0.0.0.0 UDP 123
Jan 29 14:38:09 spktmp ntpd[1707]: Listen and drop on 1 v6wildcard :: UDP 123
Jan 29 14:38:09 spktmp ntpd[1707]: Listen normally on 2 lo 127.0.0.1 UDP 123
Jan 29 14:38:09 spktmp ntpd[1707]: Listen normally on 3 ens192 192.168.11.97 UDP 123
Jan 29 14:38:09 spktmp ntpd[1707]: Listen normally on 4 ens192 fe80::47f4:c80e:f24:5e89 UDP 123
Jan 29 14:38:09 spktmp ntpd[1707]: Listening on routing socket on fd #21 for interface updates
Jan 29 14:38:09 spktmp ntpd[1707]: 0.0.0.0 c016 06 restart
Jan 29 14:38:09 spktmp ntpd[1707]: 0.0.0.0 c012 02 freq_set kernel 0.000 PPM
Jan 29 14:38:09 spktmp ntpd[1707]: 0.0.0.0 c011 01 freq_not_set
```

## Disable the Linux Firewall

The default Linux firewall settings are too restrictive for any application deployment. Since the Cisco UCS Big Data deployment will be in its own isolated network, the firewall service is disabled.

To disable the Linux firewall, follow these steps:

1. Stop and disable the Firewall service on the admin virtual machine:

```
[root@admin ~]# systemctl stop firewalld
[root@admin ~]# systemctl disable firewalld
```

```
[root@admin ~]# systemctl stop firewalld
[root@admin ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@admin ~]# systemctl status firewalld
? firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)

Jan 25 10:47:25 admin systemd[1]: Starting firewalld - dynamic firewall daemon...
Jan 25 12:47:27 admin systemd[1]: Started firewalld - dynamic firewall daemon.
Jan 25 17:35:13 admin systemd[1]: Stopping firewalld - dynamic firewall daemon...
Jan 25 17:35:14 admin systemd[1]: Stopped firewalld - dynamic firewall daemon.
```

2. Stop and disable the Firewall service on the spktmp virtual machine:

```
[root@spktmp ~]# systemctl stop firewalld
[root@spktmp ~]# systemctl disable firewalld
[root@spktmp ~]# systemctl stop firewalld
[root@spktmp ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@spktmp ~]# systemctl status firewalld
? firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)

Jan 25 11:47:25 spktmp systemd[1]: Starting firewalld - dynamic firewall daemon...
Jan 25 13:47:27 spktmp systemd[1]: Started firewalld - dynamic firewall daemon.
Jan 25 17:45:13 spktmp systemd[1]: Stopping firewalld - dynamic firewall daemon...
Jan 25 17:45:14 spktmp systemd[1]: Stopped firewalld - dynamic firewall daemon.
```



You can reconfigure the IP tables' settings in order to match the requirements of your particular deployment and turn the service back on. Consult the Splunk documentation to determine the appropriate IP tables' settings.

## Disable SELinux

SELinux must be disabled during the installation procedure and cluster setup. SELinux can be enabled after installation and while the cluster is running. SELinux can be disabled by editing `/etc/selinux/config` and changing the SELINUX line to SELINUX=disabled.

To disable SELinux, follow these steps:

1. Disable SELinux on the admin virtual machine:

```
[root@admin ~]# sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
[root@admin ~]# setenforce 0
[root@admin ~]# sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
[root@admin ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

[root@admin ~]# setenforce 0
```

2. Disable SELinux on the spktmp virtual machine:

```
[root@spktmp ~]# sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
[root@spktmp ~]# setenforce 0
[root@spktmp ~]# sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
[root@spktmp ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

[root@spktmp ~]# setenforce 0
```

## Enable Syslog

Syslog must be enabled on each node to preserve logs pertaining to killed processes or failed jobs. Modern versions such as syslog-ng and rsyslog are possible, making it more difficult to be sure that a syslog daemon is present.

To confirm that the service is properly configured, run the following command:

1. On admin virtual machine:

```
[root@admin ~]# rsyslogd -v
[root@admin ~]# service rsyslog status

[root@admin ~]$ rsyslogd -v
rsyslogd 8.24.0, compiled with:
PLATFORM:                               x86_64-redhat-linux-gnu
PLATFORM (lsb_release -d):
FEATURE_REGEXP:                          Yes
GSSAPI Kerberos 5 support:               Yes
FEATURE_DEBUG (debug build, slow code):  No
32bit Atomic operations supported:       Yes
64bit Atomic operations supported:       Yes
memory allocator:                         system default
Runtime Instrumentation (slow code):     No
uuid support:                             Yes
Number of Bits in RainerScript integers: 64

See http://www.rsyslog.com for more information.
[root@admin ~]$ service rsyslog status
Redirecting to /bin/systemctl status rsyslog.service
? rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-01-25 12:47:28 PST; 4h 59min ago
     Docs: man:rsyslogd(8)
           http://www.rsyslog.com/doc/
   Main PID: 1294 (rsyslogd)
   CGroup: /system.slice/rsyslog.service
           └─1294 /usr/sbin/rsyslogd -n

Jan 25 12:47:28 admin systemd[1]: Starting System Logging Service...
Jan 25 12:47:28 admin rsyslogd[1294]: [origin software="rsyslogd" swVersion="8.24.0" x-pid="1294" x-info="http://www.rsyslog.com"] start
Jan 25 12:47:28 admin systemd[1]: Started System Logging Service.
```

2. On spktmp virtual machine:

```
[root@spktmp ~]# rsyslogd -v
[root@spktmp ~]# systemctl status rsyslog
```

```

[root@spktmp ~]# rsyslogd -v
rsyslogd 8.24.0, compiled with:
  PLATFORM:                               x86_64-redhat-linux-gnu
  PLATFORM (lsb_release -d):
  FEATURE_REGEX:                           Yes
  GSSAPI Kerberos 5 support:                Yes
  FEATURE_DEBUG (debug build, slow code):  No
  32bit Atomic operations supported:        Yes
  64bit Atomic operations supported:        Yes
  memory allocator:                         system default
  Runtime Instrumentation (slow code):      No
  uuid support:                             Yes
  Number of Bits in RainerScript integers:  64

See http://www.rsyslog.com for more information.
[root@spktmp ~]# systemctl status rsyslog
● rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-02-15 16:54:58 PST; 1h 20min ago
     Docs: man:rsyslogd(8)
           http://www.rsyslog.com/doc/
   Main PID: 1022 (rsyslogd)
   CGroup: /system.slice/rsyslog.service
           └─1022 /usr/sbin/rsyslogd -n

Feb 15 16:54:58 spktmp systemd[1]: Starting System Logging Service...
Feb 15 16:54:58 spktmp rsyslogd[1022]: [origin software="rsyslogd" swVersion="8.24.0" x-pid="1022" x-info="http://www.rsyslog.com"] start
Feb 15 16:54:58 spktmp systemd[1]: Started System Logging Service.
[root@spktmp ~]#

```

## Set ulimit

In Linux, the 'nofile' property located in /etc/security/limits.conf, defines the number of i-nodes that can be opened simultaneously; with the default value of 1024, the system may appear to be out of disk space and would show no i-nodes are available. This value should be set to 64000 on every node for user root and user splunk.



When the Splunk Enterprise software is installed, a service user account with the name "splunk" gets created automatically. Since all Splunk related operations are performed as the user "splunk", its ulimits need to be increased. Higher values are unlikely to result in an appreciable performance gain.

To set the ulimit, follow these steps:

1. Set the 'nofile' properties of root and splunk users to 64,000 by editing the /etc/security/limits.conf on admin virtual machine. Add the following lines to this file:

```

root soft nofile 64000
root hard nofile 64000
splunk soft nofile 64000
splunk hard nofile 64000

```

2. Verify the ulimit settings by running the following command. The command should report 64000:

```

[root@admin ~]# vi /etc/security/limits.conf
[root@admin ~]# ulimit -n

```



The ulimit values are applied only to a new shell; running the command on a node from an earlier instance of a shell will show old values.

```
[root@admin ~]# vi /etc/security/limits.conf
[root@admin ~]# grep 64000 /etc/security/limits.conf
root soft nofile 64000
root hard nofile 64000
splunk soft nofile 64000
splunk hard nofile 64000
[root@admin ~]# ulimit -n
1024
[root@admin ~]# exit
logout
Connection to 192.168.66.98 closed.

[2019-01-25 15:52:38] ~
[Administrator.HXJump1Win1] ? ssh root@192.168.66.98
Last login: Fri Jan 25 17:28:30 2019 from 192.168.66.103
[root@admin ~]# ulimit -n
64000
```

3. Set the 'nofile' properties of the root and splunk users to 64000 by editing the /etc/security/limits.conf on the spktmp virtual machine. Add the following lines to this file:

```
root soft nofile 64000
root hard nofile 64000
splunk soft nofile 64000
splunk hard nofile 64000
```

4. Verify the ulimit settings by running the following command. The command should report 64000.

```
[root@spktmp ~]# vi /etc/security/limits.conf
[root@spktmp ~]# ulimit -n
```

```
[root@spktmp ~]# vi /etc/security/limits.conf
[root@spktmp ~]# cat /etc/security/limits.conf | grep 64000
root soft nofile 64000
root hard nofile 64000
splunk soft nofile 64000
splunk hard nofile 64000
[root@spktmp ~]#
[root@spktmp ~]#
[root@spktmp ~]#
[root@spktmp ~]# exit
logout
Connection to spktmp closed.
[root@admin ~]# ssh spktmp
Last login: Tue Apr 2 23:21:02 2019 from admin
[root@spktmp ~]# ulimit -n
64000
[root@spktmp ~]# █
```

### Set TCP Retries

Adjusting the `tcp_retries` parameter for the system network enables a faster detection of failed nodes. Given the advanced networking features of Cisco UCS, this is a safe and recommended change (failures observed at the operating system layer are most likely serious rather than transitory). On the Splunk node, setting the number of TCP retries to 5 can help detect unreachable nodes with less latency.

To set the `tcp_retries` parameter, follow these steps:

1. Edit the file /etc/sysctl.conf and on both spktmp and admin virtual machines by adding the following lines:

```
net.ipv4.tcp_retries2=5
```

### Configure Virtual Machine Swapping

The `vm.swappiness` value from 0 to 100 controls the degree in which the system swaps. A high value prioritizes system performance, aggressively swapping processes out of physical memory when they are not active. A low value avoids

swapping processes out of physical memory for as long as possible. In order to reduce swapping, run the following on all nodes. The default value is 60.

1. Edit the file `/etc/sysctl.conf` and on both `spktmp` and `admin` virtual machines adding the following lines:

```
vm.swappiness=1
```

### Disable IPv6 Defaults

You need to disable IPv6 since the addresses used are IPv4. To disable the IPv6, follow these steps:

1. Edit the file `/etc/sysctl.conf` on both `spktmp` and `admin` virtual machines by adding the following lines:

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

2. Load the settings from default `sysctl` file `/etc/sysctl.conf`:

```
[root@spktmp ~]# sysctl -p
[root@spktmp ~]# vi /etc/sysctl.conf
[root@spktmp ~]# cat /etc/sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.tcp_retries2=5
vm.swappiness=1
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1

[root@spktmp ~]#
[root@spktmp ~]# sysctl -p
net.ipv4.tcp_retries2 = 5
vm.swappiness = 1
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
[root@spktmp ~]#
```

### Disable Transparent Huge Pages

Disabling the Transparent Huge Pages (THP) reduces elevated CPU usage caused by the THP. To disable the THP, follow these steps:

1. From the `spktmp` and `admin` virtual machines, run the following commands:

```
echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
```



The above commands need to be run every time the Linux system starts up. Therefore, you need to add the following commands to `/etc/rc.local`, so that they are executed automatically upon every reboot.

---

2. On the `admin` and `spktmp` virtual machines, edit the file `/root/thp_disable` by adding the following lines:

```
echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
```

3. Append the contents of file `thp_disable` to `/etc/rc.local`:



```
cat /root/thp_disable >> /etc/rc.local
```

```
[root@spktmp ~]# cd /sys/kernel/mm/
[root@spktmp mm]# ls
hugepages ksm transparent_hugepage
[root@spktmp mm]# echo never > /sys/kernel/mm/transparent_hugepage/enabled
[root@spktmp mm]# echo never > /sys/kernel/mm/transparent_hugepage/defrag
[root@spktmp mm]# cat /etc/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
[root@spktmp mm]# cat /root/thp_disable
cat: /root/thp_disable: No such file or directory
[root@spktmp mm]#
[root@spktmp mm]#
[root@spktmp mm]# cat /root/thp_disable
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo never > /sys/kernel/mm/transparent_hugepage/defrag
[root@spktmp mm]# cat /root/thp_disable >> /etc/rc.local
[root@spktmp mm]# cat /etc/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo never > /sys/kernel/mm/transparent_hugepage/defrag
[root@spktmp mm]#
```

- Shutdown spktmp, change the CD/DVD setting from Datastore ISO File to Client Device. It will be used as a base template for cloning other Splunk virtual machine templates. This completes the steps to create and configure the Splunk base template – spktmp.

## Create Splunk Admin Virtual Machine Template

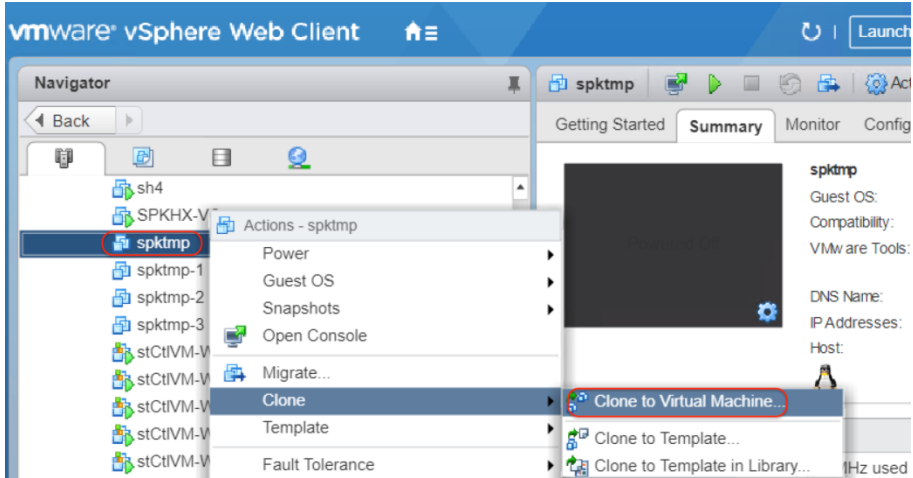
Table 22 lists the virtual machine – spktmp-1's configuration.

Table 22 Splunk Admin Virtual Machine Configuration – spktmp-1

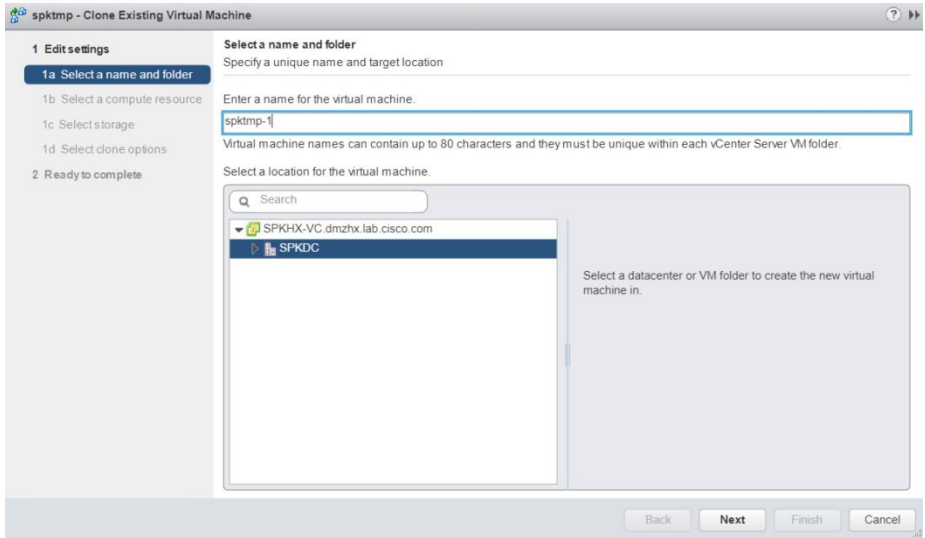
Setting	Value
Virtual Machine Name	spktmp-1
Virtual Machine Location	2nd HX converged node
Compatibility	ESXi 6.5 and later
Operating System	RedHat Enterprise Linux 7.x
vCPUs	4
Memory	16 GB (fully reserved)
Hard Disk Size for OS	16 GB
Provisioning Type	Thin
Number of Data Hard Disks	1
Hard Disk Size for Data	48 GB

Setting	Value
Provisioning Type	Thin
SCSI Controller #	0
SCSI Controller Type	VMWare Paravirtual
Data Network Type	VMXNET3 (Splunk Data VLAN)

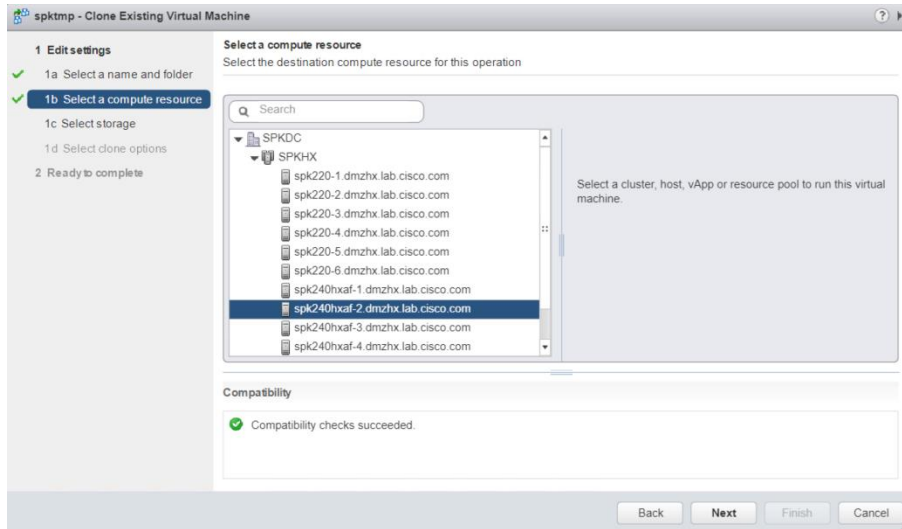
1. Open the vSphere web client, right-click the virtual machine spktmp, select Clone > Clone to Virtual Machine:



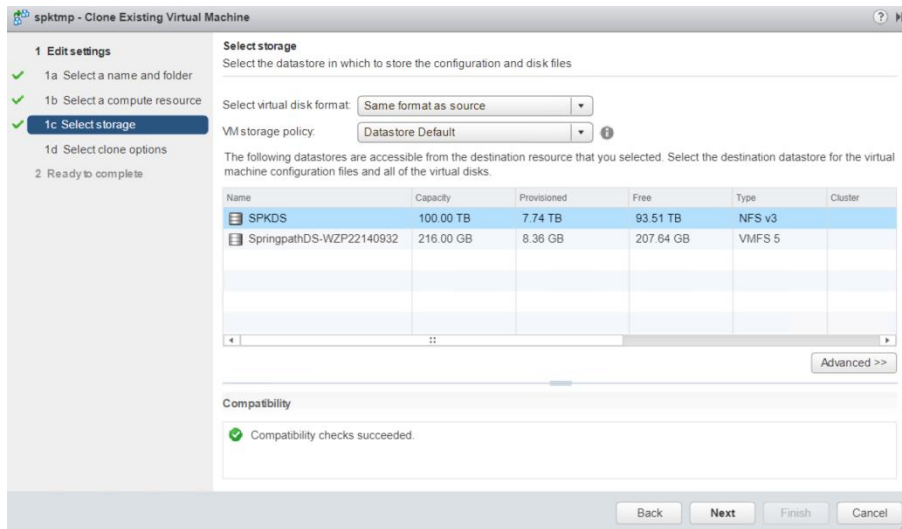
2. Enter spktmp-1 as virtual machine name and click NEXT.



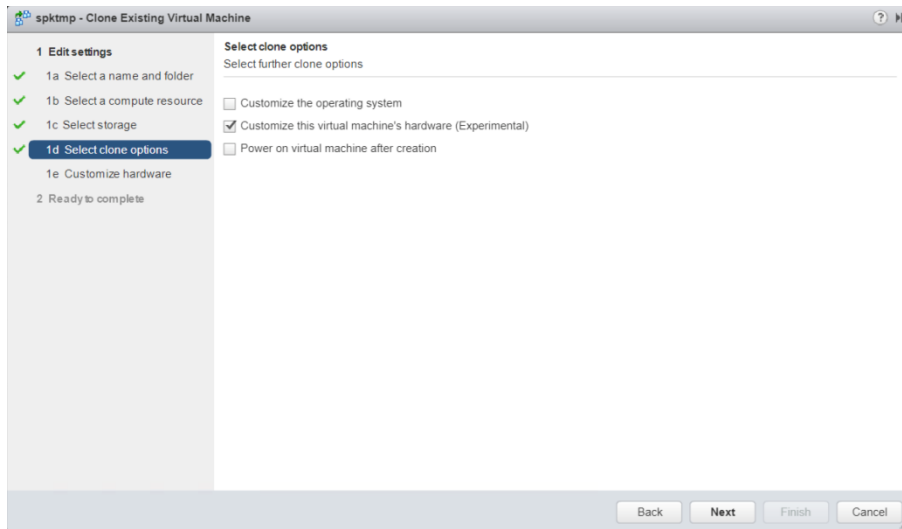
3. Select the second one of the HXAF240c nodes of the HyperFlex cluster and click NEXT.



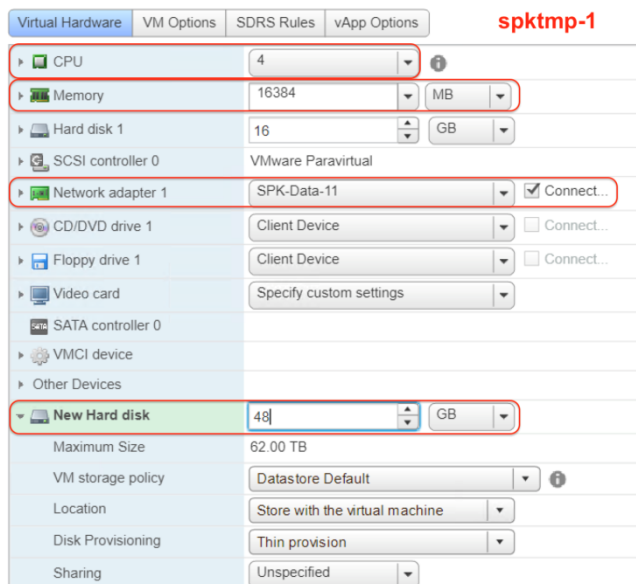
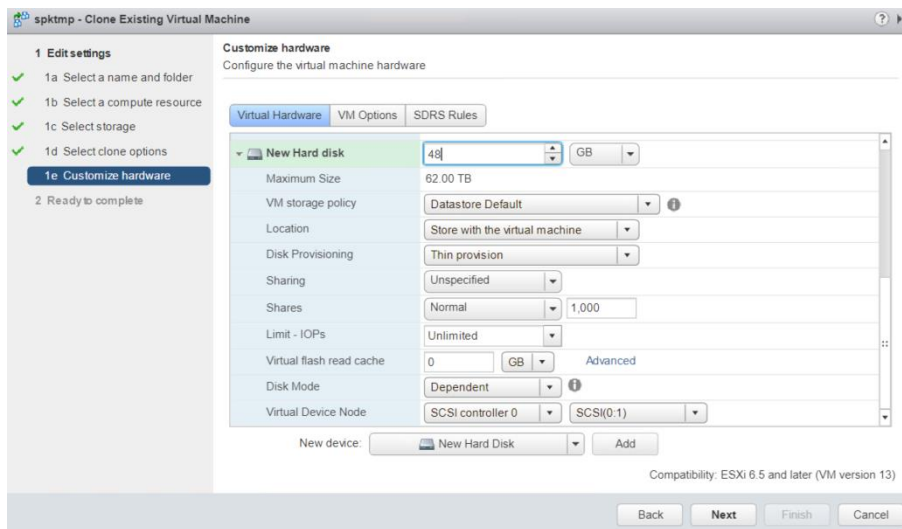
4. Select SPKDS as storage and click NEXT.



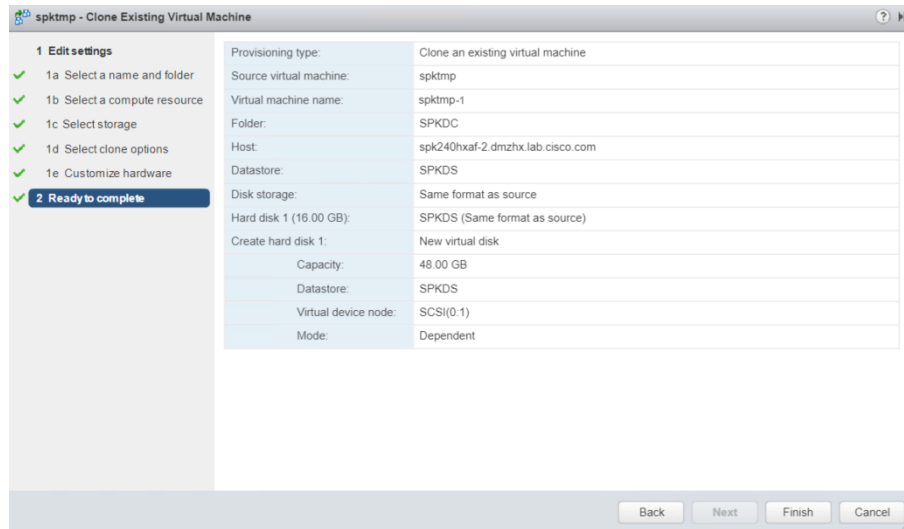
5. Select Customize the virtual machine's hardware (Experimental) as clone options and click NEXT.



6. Add a New Hard Disk with 48 GB capacity, Thin Provision and connected to SCSI Controller 0. Click NEXT.



7. Check the configuration details and click FINISH.



8. Power on the Virtual Machine and open the Console.
9. Log into the operating system as root user from the console.
10. Use nmtui to configure the network and hostname of the system.

Hostname	Network Interface	Data Network (VLAN 11)
spktmp-1	eth0	192.168.11.97/24

11. Edit the file /etc/hosts on admin, replace the entry spktmp with spktmp-1.
12. From admin, log into spktmp-1. Run lsblk to get the name of the newly added hard disk (e.g. sdb).

```
[root@admin ~]# ssh spktmp-1
Last login: Thu Apr 4 13:49:11 2019 from admin
[root@spktmp-1 ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0         2:0    1   4K  0 disk
sda         8:0    0  16G  0 disk
├─sda1      8:1    0   1G  0 part /boot
├─sda2      8:2    0  15G  0 part
├─rhel-root 253:0   0 13.4G  0 lvm  /
├─rhel-swap 253:1   0  1.6G  0 lvm  [SWAP]
└─sdb       8:16   0  48G  0 disk
sr0        11:0    1 1024M  0 rom
[root@spktmp-1 ~]#
```

13. Create and run the script `driveconf-idx.sh` to create XFS file system on the new hard disk and mount it to the drive /data/disk1.

```
[root@spktmp-1 ~]# cat driveconf-idx.sh
/sbin/mkfs.xfs -f /dev/sdb
```

```
#Identify UUID
```

```
UUID=`blkid /dev/sdb | cut -d " " -f2 | cut -d "=" -f2 | sed 's"/"/g'`
```

```
#echo "UUID of sdb = ${UUID}, mounting sdb as UUID on /data/disk1"
/bin/mkdir -p /data/disk1
```

```

/bin/mount -t xfs -o allocsize=128m,inode64,noatime,nobarrier,nodiratime -U ${UUID}
/data/disk1
echo "UUID=${UUID} /data/disk1 xfs allocsize=128m,inode64,noatime,nobarrier,nodiratime 0
0" >> /etc/fstab

```

```

[root@spktmp-1 ~]#cat /root/driveconf-idx.sh
/sbin/mkfs.xfs -f /dev/sdb

#Identify UUID
UUID=`blkid /dev/sdb | cut -d " " -f2 | cut -d "=" -f2 | sed 's//g'`
#echo "UUID of sdb = ${UUID}, mounting sdb as UUID on /data/disk1"
/bin/mkdir -p /data/disk1

/bin/mount -t xfs -o allocsize=128m,inode64,noatime,nobarrier,nodiratime -U ${UUID} /data/disk1

echo "UUID=${UUID} /data/disk1 xfs allocsize=128m,inode64,noatime,nobarrier,nodiratime 0 0" >> /etc/fstab

[root@spktmp-1 ~]# ./driveconf-idx.sh
meta-data=/dev/sdb          isize=512    agcount=4, agsize=3145728 blks
           =                sectsz=512   attr=2,    projid32bit=1
           =                crc=1          finobt=0,  sparse=0
data      =                bsize=4096 blocks=12582912, imaxpct=25
           =                sunit=0        swidth=0  blks
naming    =version 2       bsize=4096 ascii-ci=0  ftype=1
log       =internal log   bsize=4096 blocks=6144,  version=2
           =                sectsz=512   sunit=0   blks, lazy-count=1
realtime  =none           extsz=4096  blocks=0,  rtextents=0

```

14. Reboot spktmp-1 and make sure /data/disk1 is auto mounted upon reboot.

```

[root@spktmp-1 ~]# reboot
Connection to spktmp-1 closed by remote host.
Connection to spktmp-1 closed.
[root@admin ~]#
[root@admin ~]#
[root@admin ~]# ssh spktmp-1
Last login: Thu Apr 4 13:54:56 2019 from admin
[root@spktmp-1 ~]# df -k
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/mapper/rhel-root 14034944 1914024 12120920 14% /
devtmpfs        32891932      0 32891932  0% /dev
tmpfs           32904148      0 32904148  0% /dev/shm
tmpfs           32904148     9072 32895076  1% /run
tmpfs           32904148      0 32904148  0% /sys/fs/cgroup
/dev/sdb        50307072    32944 50274128  1% /data/disk1
/dev/sdal      1038336     147948  890388 15% /boot
tmpfs          6580832      0 6580832  0% /run/user/0
[root@spktmp-1 ~]#

```

15. Download Splunk Enterprise version 7.2.3 software splunk-7.2.3-06d57c595b80-linux-2.6-x86\_64.rpm from Splunk website. Save it to /tmp/ directory on spktmp-1 virtual machine.
16. Modify the permissions on the Splunk Enterprise RPM file to include execution privileges:.

```

[root@spktmp-1 ~]# chmod +x /tmp/splunk-7.2.3-06d57c595b80-linux-2.6-x86_64.rpm
Install Splunk software on spktmp-1.
[root@spktmp-1 ~]# rpm -ivh --prefix=/data/disk1 /tmp/splunk-7.2.3-06d57c595b80-linux-2.6-x86_64.rpm

```

```

[root@spktmp-1 ~]# rpm -ivh --prefix=/data/disk1 /tmp/splunk-7.2.3-06d57c595b80-linux-2.6-x86_64.rpm
warning: /tmp/splunk-7.2.3-06d57c595b80-linux-2.6-x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID b3cd4420: NOKEY
Preparing...
Updating / installing...
 1:splunk-7.2.3-06d57c595b80
complete

```

17. Setup the environment variable \$SPLUNK\_HOME. Note that you will see the variable defined after logging off and logging back into the server.

```

[root@spktmp-1 ~]# echo SPLUNK_HOME=/data/disk1/splunk >> /etc/environment

```

```
[root@spktmp-1 ~]# echo SPLUNK_HOME=/data/disk1/splunk >> /etc/environment
[root@spktmp-1 ~]# echo $SPLUNK_HOME

[root@spktmp-1 ~]#
[root@spktmp-1 ~]# exit
logout
Connection to 192.168.11.97 closed.
[root@admin ~]# ssh 192.168.11.97
Last login: Tue Feb 19 01:27:22 2019 from admin
[root@spktmp-1 ~]# echo $SPLUNK_HOME
/data/disk1/splunk
```

18. When the Splunk Enterprise software is installed, a service user account by name “splunk” gets created automatically. The user “splunk” is created without a password. Now assign the password for the user ‘splunk’:

```
[root@spktmp-1 ~]# echo Cisco123 | passwd splunk --stdin
[root@spktmp-1 ~]# echo Cisco123 | passwd splunk --stdin
Changing password for user splunk.
passwd: all authentication tokens updated successfully.
[root@spktmp-1 ~]# exit
logout
Connection to spktmp-1 closed.
[root@admin ~]# ssh splunk@spktmp-1
splunk@spktmp-1's password:
Last login: Mon Feb 18 14:41:02 2019 from admin
[splunk@spktmp-1 ~]$
```



The password for the user “splunk” on the admin node shall be set to the same as on all the Splunk nodes.

19. Configure the password-less login for that user account “splunk.”
20. Log into the admin server with the user account “splunk.” Run the **ssh-keygen** command to create both public and private keys on the admin node for the user “splunk.”
21. Run the **ssh-copy-id** command from the admin node to copy the public key `id_rsa.pub` to `spktmp-1` virtual machine. It appends the keys to the remote-host’s `$SPLUNK_HOME/.ssh/authorized_key`.

```
[splunk@admin ~]$ ls .ssh/
id_rsa id_rsa.pub
[splunk@admin ~]$ ssh-copy-id -i .ssh/id_rsa.pub -o StrictHostKeyChecking=no 192.168.11.97
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
splunk@192.168.11.97's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh -o 'StrictHostKeyChecking=no' '192.168.11.97'"
and check to make sure that only the key(s) you wanted were added.
```

22. On admin, login as the splunk user to `spktmp-1`. Verify that the password-less login succeeds.
23. On `spktmp-1`, start the Splunk Enterprise for the first time:

```
[splunk@spktmp-1 ~]$ $SPLUNK_HOME/bin/splunk start --accept-license --no-prompt
```

```

[splunk@spktmp-1 ~]$ $SPLUNK_HOME/bin/splunk start --accept-license --no-prompt

This appears to be your first time running this version of Splunk.

IMPORTANT: Because an admin password was not provided, the admin user
will not be created. You will have to set up an admin username/password
later using user-seed.conf.
Copying '/data/disk1/splunk/etc/openldap/ldap.conf.default' to '/data/disk1/splunk/etc/openldap/ldap.conf'.
Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
writing RSA key

Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
writing RSA key

Moving '/data/disk1/splunk/share/splunk/search_mrsparkle/modules.new' to '/data/disk1/splunk/share/splunk/search_mrsparkle/modules'.

Splunk> The IT Search Engine.

Checking prerequisites...
  Checking http port [8000]: open
  Checking mgmt port [8089]: open
  Checking appserver port [127.0.0.1:8065]: open
  Checking kvstore port [8191]: open
  Checking configuration... Done.
  Creating: /data/disk1/splunk/var/lib/splunk
  Creating: /data/disk1/splunk/var/run/splunk
  Creating: /data/disk1/splunk/var/run/splunk/appserver/i18n
  Creating: /data/disk1/splunk/var/run/splunk/appserver/modules/static/css
  Creating: /data/disk1/splunk/var/run/splunk/upload
  Creating: /data/disk1/splunk/var/spool/splunk
  Creating: /data/disk1/splunk/var/spool/dirmoncache
  Creating: /data/disk1/splunk/var/lib/splunk/authDb
  Creating: /data/disk1/splunk/var/lib/splunk/hashDb
New certs have been generated in '/data/disk1/splunk/etc/auth'.
  Checking critical directories... Done
  Checking indexes...
    Validated: _audit _internal _introspection _telemetry _thefishbucket history main summary
  Done
  Checking filesystem compatibility... Done
  Checking conf files for problems...
  Done
  Checking default conf files for edits...
  Validating installed files against hashes from '/data/disk1/splunk/splunk-7.2.3-06d57c595b80-linux-2.6-x86_64-manifest'
  All installed files intact.
  Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
Generating a 2048 bit RSA private key
.....+++++
.....+++++
writing new private key to 'privKeySecure.pem'
-----
Signature ok
subject=CN=spktmp-1/O=SplunkUser
Getting CA Private Key
writing RSA key
Done

[ OK ]

waiting for web server at http://127.0.0.1:8000 to be available... Done

If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

The Splunk web interface is at http://spktmp-1:8000

```

24. When starting Splunk Enterprise for the first time and using the `--no-prompt` CLI argument, Splunk Enterprise starts without an administrator user and web login is prevented. Therefore, you must create the credentials immediately after Splunk process starts and then restart Splunk Enterprise. The administrator password is defined in the file `$SPLUNK_HOME/etc/system/local/user-seed.conf`:

```

[splunk@spktmp-1 ~]$ SPLUNK_HOME/bin/splunk stop
[splunk@spktmp-1 ~]$ vi $SPLUNK_HOME/etc/system/local/user-seed.conf

[user_info]
USERNAME = admin
PASSWORD = Cisco123

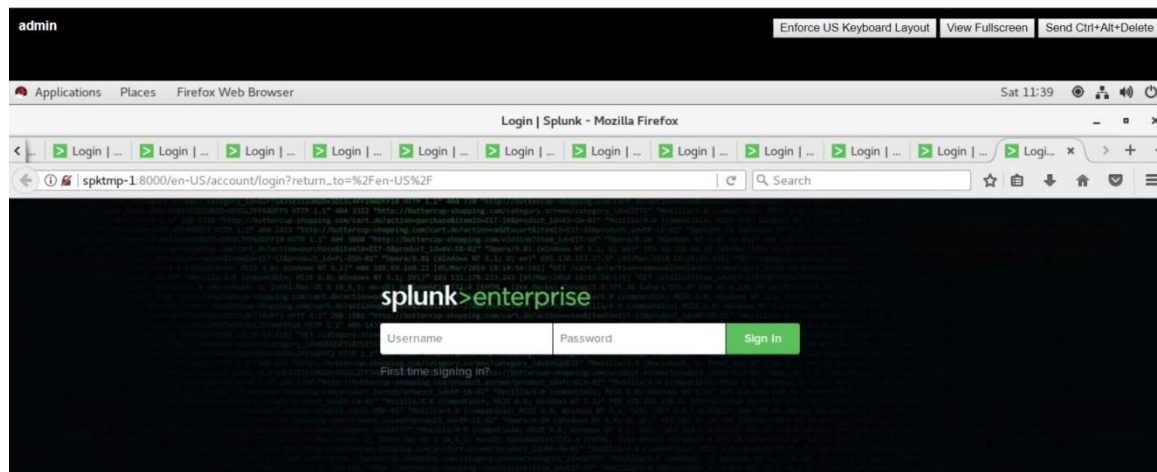
[splunk@spktmp-1 ~]$ SPLUNK_HOME/bin/splunk start

```



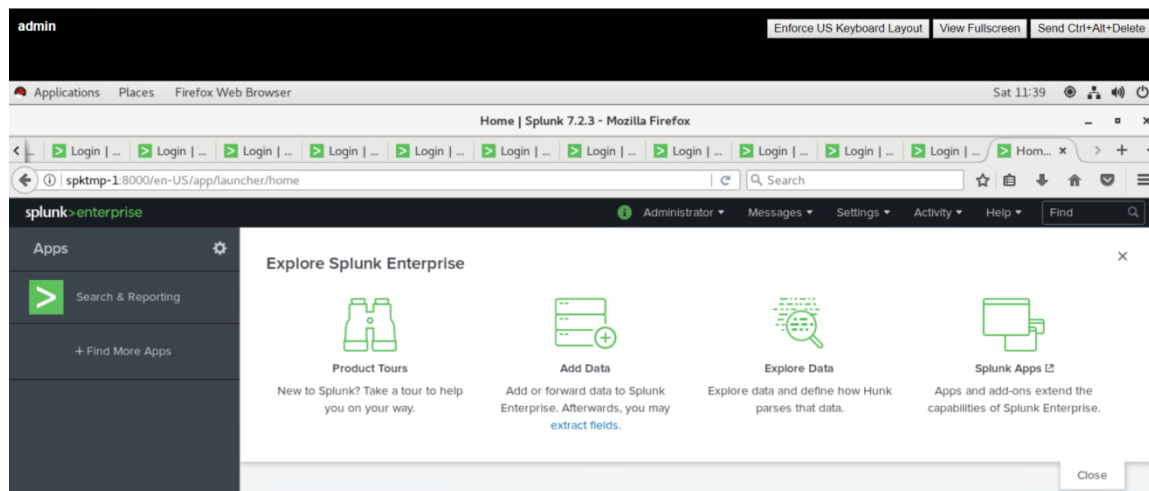
```
[splunk@spktmp-1 ~]$ $SPLUNK_HOME/bin/splunk stop
Stopping splunkd...
Shutting down. Please wait, as this may take a few minutes.
.. [ OK ]
Stopping splunk helpers...
.. [ OK ]
Done.
[splunk@spktmp-1 ~]$ $SPLUNK_HOME/bin/splunk status
splunkd is not running.
[splunk@spktmp-1 ~]$ ls -l $SPLUNK_HOME/etc/system/local/
total 16
-rw----- 1 splunk splunk 26 Feb 18 16:20 inputs.conf
-rw----- 1 splunk splunk 60 Feb 18 16:20 migration.conf
-r--r--r-- 1 splunk splunk 265 Dec 21 17:26 README
-rw----- 1 splunk splunk 578 Feb 18 16:20 server.conf
[splunk@spktmp-1 ~]$ vi $SPLUNK_HOME/etc/system/local/user-seed.conf
[splunk@spktmp-1 ~]$
[splunk@spktmp-1 ~]$
[splunk@spktmp-1 ~]$ $SPLUNK_HOME/bin/splunk start
```

25. From the admin console, start the web browser; type <http://spktmp-1:8000>, to open Splunk Enterprise login page:



Splunk software uses port number 8000 as the default web interface and port number 8089 as the default management port.

26. Enter the admin user and the password to sign into the management GUI. In this CVD the password for the Splunk Administrator is set to 'Cisco123' for all the Splunk nodes.



27. Initialize that Splunk process automatically runs upon the server boot. Log into the spktmp-1 virtual machine as root user. Run the command `$SPLUNK_HOME/bin/splunk enable boot-start -user splunk` from the command line.
28. (Optional) Add splunk start/stop/restart to the system control processes that can be executed by the user splunk as the sudo user, by adding the following contents into the file `/etc/sudoers`.

```
splunk ALL=(root) NOPASSWD: /usr/bin/systemctl restart Splunkd.service
splunk ALL=(root) NOPASSWD: /usr/bin/systemctl stop Splunkd.service
splunk ALL=(root) NOPASSWD: /usr/bin/systemctl start Splunkd.service
```

29. Reboot the server. When the server boots up, verify that the Splunk software starts upon boot and verify the sudo commands work.

```
[root@spktmp-1 ~]# $SPLUNK_HOME/bin/splunk enable boot-start -user splunk
Init script installed at /etc/systemd/system/.
Init script is configured to run at boot.
[root@spktmp-1 ~]# vi /etc/sudoers
[root@spktmp-1 ~]# grep splunk /etc/sudoers
splunk ALL=(root) NOPASSWD: /usr/bin/systemctl restart Splunkd.service
splunk ALL=(root) NOPASSWD: /usr/bin/systemctl stop Splunkd.service
splunk ALL=(root) NOPASSWD: /usr/bin/systemctl start Splunkd.service
[root@spktmp-1 ~]# reboot
Connection to 192.168.11.97 closed by remote host.
Connection to 192.168.11.97 closed.
[spunk@admin ~]# ssh 192.168.11.97
Last login: Tue Feb 19 01:37:00 2019 from admin
[spunk@spktmp-1 ~]# $SPLUNK_HOME/bin/splunk status
splunkd is running (PID: 1140).
splunk helpers are running (PIDs: 1616 1629 1710 1824 1851).
[spunk@spktmp-1 ~]# sudo systemctl restart Splunkd.service
[spunk@spktmp-1 ~]# $SPLUNK_HOME/bin/splunk status
splunkd is running (PID: 2075).
splunk helpers are running (PIDs: 2143 2156 2294 2313).
[spunk@spktmp-1 ~]#
```

30. Delete eth0 interface, shutdown spktmp-1 virtual machine. This completes the steps to prepare spktmp-1 as the Splunk administration virtual machine template.

## Create Splunk Search Head Virtual Machine Template

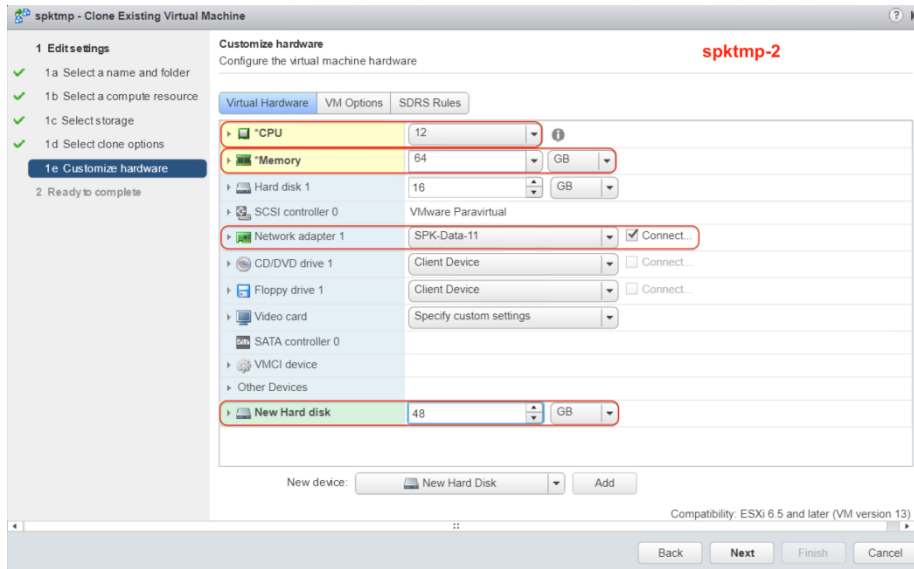
Table 23 lists the virtual machine – spktmp-2’s configuration.

Table 23 Splunk Search Head Virtual Machine Configuration – spktmp-2

Setting	Value
Virtual Machine Name	spktmp-2
Virtual Machine Location	3 <sup>rd</sup> HX converged node
Compatibility	ESXi 6.5 and later
Operating System	RedHat Enterprise Linux 7.x
vCPUs	12
Memory	64 GB (fully reserved)
Hard Disk Size for OS	16 GB
Provisioning Type	Thin
Number of Data Hard Disks	1

Setting	Value
Hard Disk Size for Data	48 GB
Provisioning Type	Thin
SCSI Controller #	0
SCSI Controller Type	VMWare Paravirtual
Data Network Type	VMXNET3 (Splunk Data VLAN)

Repeat the steps described in section Create Splunk Admin Virtual Machine Template and create the virtual machine template for Splunk search heads – spktmp-2 now. At Step 6, modify the hardware configuration differently as follows:



### Create Splunk Indexer Virtual Machine Template

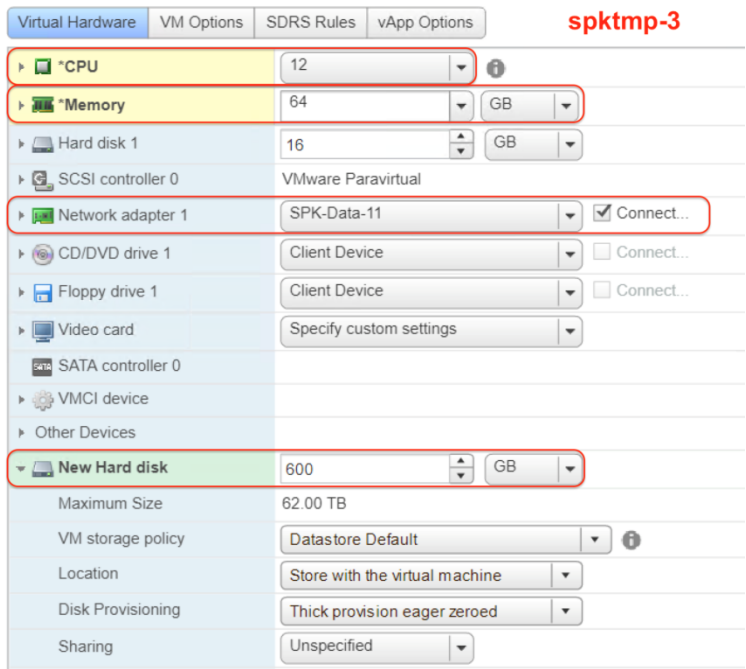
Table 24 lists the virtual machine – spktmp-3’s configuration.

Table 24 Splunk Search Head Virtual Machine Configuration – spktmp-3

Setting	Value
Virtual Machine Name	spktmp-3
Virtual Machine Location	4 <sup>th</sup> HX converged node
Compatibility	ESXi 6.5 and later
Operating System	RedHat Enterprise Linux 7.x
vCPUs	12
Memory	64 GB (fully reserved)
Hard Disk Size for OS	16 GB
Provisioning Type	Thin
Number of Data Hard Disks	1

Setting	Value
Hard Disk Size for Data	600 GB
Provisioning Type	Thick Provision Eager Zeroed
SCSI Controller #	0
SCSI Controller Type	VMWare Paravirtual
Data Network Type	VMXNET3 (Splunk Data VLAN)

Repeat the steps described in section Create Splunk Admin Virtual Machine Template and create the virtual machine template for Splunk Indexers – spktmp-3 now. At Step 6, modify the hardware configuration differently as follows:



## Create Splunk Virtual Machines

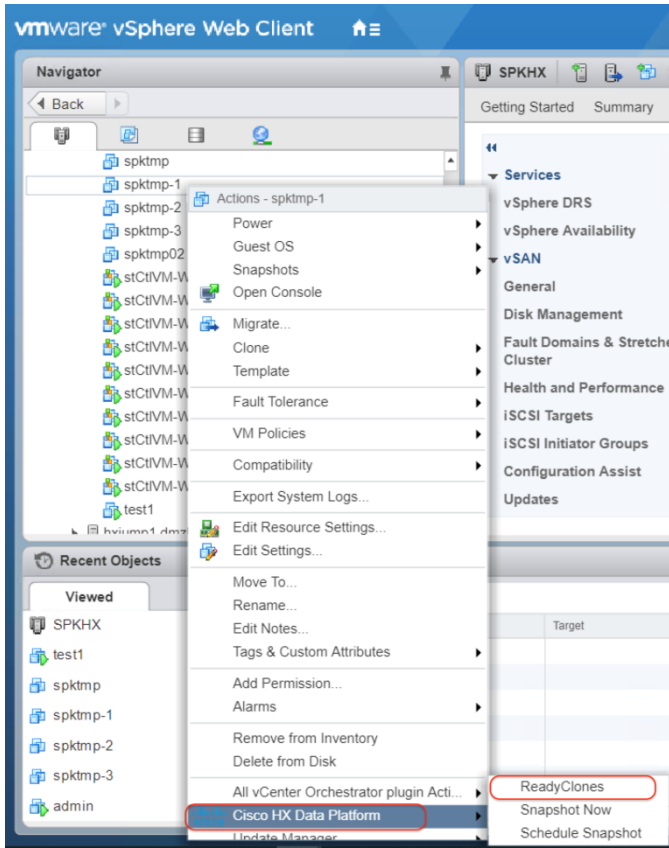
To create Splunk virtual machines from the templates, as listed in Table 25 , follow these steps:

Table 25 Splunk Virtual Machines

Name	IP Address	Origin Template	Role	Placement
admin1	192.168.11.52	spktmp-1	Cluster Master; Distributed Monitoring Console	Separate on HyperFlex computing-only nodes
admin2	192.168.11.53	spktmp-1	License Master	
admin3	192.168.11.54	spktmp-1	Search Head Cluster (SHC) Deployer	
admin4	192.168.11.55	spktmp-1	Deployment Server	
admin5	192.168.11.56	spktmp-1	Reserve	

Name	IP Address	Origin Template	Role	Placement
admin6	192.168.11.57	spktmp-1	Forwarder (Test Device)	
sh1	192.168.11.58	spktmp-2	Search Head	Separate on HyperFlex converged nodes
sh2	192.168.11.59	spktmp-2	Search Head	
sh3	192.168.11.60	spktmp-2	Search Head	
sh4	192.168.11.61	spktmp-2	Search Head	
idx1	192.168.11.62	spktmp-3	Indexer	
idx2	192.168.11.63	spktmp-3	Indexer	Separate on all HyperFlex nodes
idx3	192.168.11.64	spktmp-3	Indexer	
idx4	192.168.11.65	spktmp-3	Indexer	
idx5	192.168.11.66	spktmp-3	Indexer	
idx6	192.168.11.67	spktmp-3	Indexer	
idx7	192.168.11.68	spktmp-3	Indexer	
idx8	192.168.11.69	spktmp-3	Indexer	
idx9	192.168.11.70	spktmp-3	Indexer	
idx10	192.168.11.71	spktmp-3	Indexer	

1. Open the vSphere web client, right-click the virtual machine spktmp-1, from Cisco HX Data Platform plugin, choose ReadyClones to ready clone it to the Splunk administration virtual machines.



2. On ReadyClones page, enter the Number of clones, VM Name Prefix, starting clone number. Check the box Power on virtual machines after cloning, click OK.

**ReadyClones - spktmp-1** ✕

**Number of clones**

**Customization Specification**

Clone Names

**Resource Pool**

Clone Names

**VM Name Prefix**       **Starting clone number**

Use same name for 'Guest Name'      **Increment clone number by**

Preview

VM Name	Guest Name
admin2	admin2
admin3	admin3
admin4	admin4
admin5	admin5
admin6	admin6

Power on VMs after cloning

3. Right-click the virtual machine spktmp-2, from Cisco HX Data Platform plugin, choose ReadyClones to ready clone it to the Splunk search head virtual machines.
4. On ReadyClones page, enter the Number of clones, VM Name Prefix, starting clone number. Check the box Power on virtual machines after cloning, click OK.

ReadyClones - spktmp-2 ✕

Number of clones

Customization Specification None ▼

Clone Names Resource Pool

Resource Pool None ▼

Clone Names

VMName Prefix  Starting clone number

Use same name for 'Guest Name' Increment clone number by

Preview

VM Name	Guest Name
sh1	sh1
sh2	sh2
sh3	sh3
sh4	sh4

Power on VMs after cloning

5. Right-click the virtual machine spktmp-3, from Cisco HX Data Platform plugin, choose ReadyClones to ready clone it to the Splunk indexer virtual machines.
6. On ReadyClones page, enter the Number of clones, VM Name Prefix, starting clone number. Check the box Power on virtual machines after cloning, click OK.

ReadyClones - spktmp-3

Number of clones

Customization Specification

Clone Names Resource Pool

Resource Pool

Clone Names

VM Name Prefix  Starting clone number

Use same name for 'Guest Name' Increment clone number by

Preview

VM Name	Guest Name
idx1	idx1
idx2	idx2
idx3	idx3
idx4	idx4
idx5	idx5

Power on VMs after cloning

OK Cancel

- Wait until all cloned virtual machines powered on.
- Right-click admin1, choose Open Console. Login as root user.
- Run the following command to configure the IP address for admin1:

```
nmcli con add type ethernet con-name eth0 ifname ens192 ip4 [admin1-IP] autoconnect yes
```

- Run the following command to change admin1's hostname:

```
nmcli general hostname admin1
```

```
[root@spktmp-1 ~]# nmcli
ens192: disconnected
    "Umare VMXNET3 Ethernet Controller"
    ethernet (vmxnet3), 08:50:56:B0:50:81, hw, mtu 1500

lo: unmanaged
    "lo"
    loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

Use "nmcli device show" to get complete information about known devices and
"nmcli connection show" to get an overview on active connection profiles.

Consult nmcli(1) and nmcli-examples(5) manual pages for complete usage details.
[root@spktmp-1 ~]# nmcli connection add type ethernet con-name eth0 ifname ens192 ip4 192.168.11.52/24 autoconnect yes
Connection 'eth0' (9e848aeb-8fd8-4983-916e-7d00e2bbe08f) successfully added.
[root@spktmp-1 ~]# nmcli general hostname admin1
[root@spktmp-1 ~]#
```

- Log out and log in as root user. Verify that the hostname and IP address has been configured correctly.



```

Red Hat Enterprise Linux Server 7.5 (Maipo)
Kernel 3.10.0-862.el7.x86_64 on an x86_64

admin1 login: root
Password:
Last login: Mon Apr  8 22:23:23 on tty1
[root@admin1 ~]#
[root@admin1 ~]#
[root@admin1 ~]# nmcli
ens192: connected to eth0
"VMware UPDNET3 Ethernet Controller"
ethernet (vmmnet3), 08:50:56:80:50:81, hw, mtu 1500
inet4 192.168.11.52/24
route4 192.168.11.0/24
inet6 fe80::55a7:e223:cf20:2d97:64
route6 ff00::/8
route6 fe80::/64
route6 fe80::/64

```

12. Repeat step 8-11 for all cloned Splunk virtual machines. Verify that the hostnames and IP addresses for all the virtual machines are set correctly as listed in Table 25 .



Step 1 to 12 can be automated using PowerShell script (.ps1). The example PowerShell script about how to clone the Splunk virtual machines is provided in Appendix B: PowerShell Script Example – Clone Splunk Virtual Machines.

13. Edit the file `$SPLUNK_HOME/etc/system/local/inputs.conf` on each Splunk virtual machine, change the host to the new hostname.
14. Edit the file `$SPLUNK_HOME/etc/system/local/server.conf` on each Splunk virtual machine, change the `ServerName` to the new hostname.
15. Remove the old session instance file `$SPLUNK_HOME/etc/instance.cfg` on each Splunk virtual machine.
16. Restart Splunk service on all the Splunk virtual machines.

```

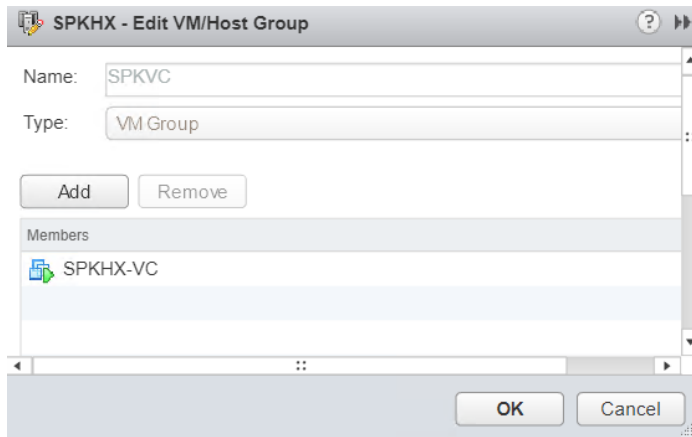
[splunk@admin ~]$ clush --group=all -B rm /data/disk1/splunk/etc/instance.cfg
[splunk@admin ~]$ clush --group=all -B sudo /usr/bin/systemctl restart Splunkd.service
[splunk@admin ~]$

```

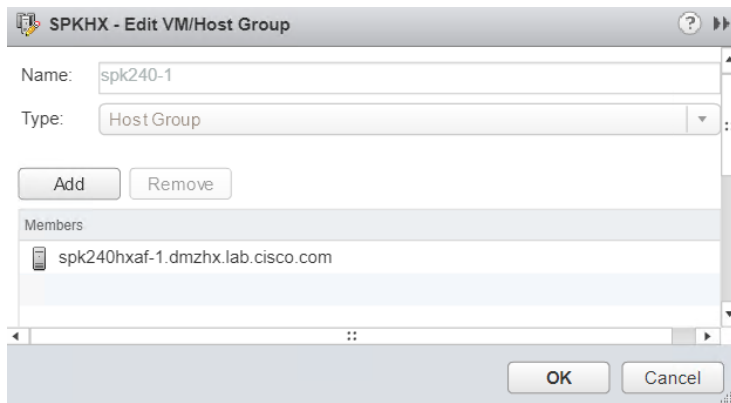
17. Create the Distributed Resource Scheduling (DRS) pin rules for the embedded vCenter virtual machine.

This process places the vCenter virtual machine on a known host, making troubleshooting and manual restart easier. You may need to search for the vCenter virtual machine on all hosts to perform any manual steps, such as bringing up the vCenter virtual machine after a full shutdown. See the VMware documentation for additional information.

18. Log into the vSphere web client.
19. Click cluster > Configure > VM/Host Groups.
20. Click Add, enter the name and select Type: VM group.
21. Click Add, select the vCenter virtual machine SPKHX-VC, click OK, and click OK again.

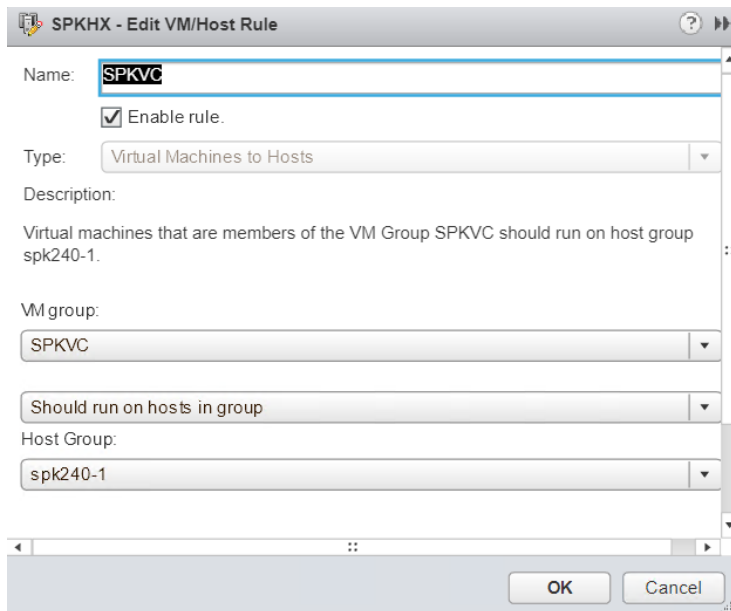


22. Click Add, enter the name and select Type: Host group, add the 1<sup>st</sup> HX converged host, click OK, and click OK again.



23. Click VM/Host Rules and select Type: Virtual Machines to Hosts.

24. Select the virtual machine group created earlier, select "Should run on hosts in group," select the host group created earlier, and enter the rule name. Click OK.



25. Create DRS pin rules for the Splunk virtual machines following the procedures described at step 13.

This step pins the Splunk virtual machines to specific hosts or ensure that anti-affinity rules are created so that indexers and search heads are distributed across the HX nodes to avoid overloaded on single resources in the cluster.

Table 26 Anti-affinity rules for Splunk Virtual Machines

Virtual Machine Group	Splunk Virtual Machine s	Host Group	HX ESXi Hosts	VM/Host Rules	Policy Type
spk-admin	admin	spk240-2	spk240hxaf-2	SPK-admin	Run virtual machines on Hosts
spkadm	admin1	compute-nodes	spk220-1	spkadm1	Run virtual machines on Hosts
	admin2		spk220-2	spkadm2	
admin3	spk220-3				
admin4	spk220-4				
admin5	spk220-5				
	admin6		spk220-6		Separate Virtual Machines
spksh	sh1	hx240-nodes	spk240hxaf-1	spksh1	Run virtual machines on Hosts
	sh2		spk240hxaf-2	spksh2	
sh3	spk240hxaf-3				
sh4	spk240hxaf-4				
spkidx	idx1	hxallnodes	spk240hxaf-1	spkidx	Separate Virtual Machines
	idx2		spk240hxaf-2		
	idx3		spk240hxaf-3		
	idx4		spk240hxaf-4		
	idx5		spk220-1		
	idx6		spk220-2		
	idx7		spk220-3		
	idx8		spk220-4		
	idx9		spk220-5		
	idx10		spk220-6		

Figure 32 VM/Host Rules Created for this Setup

**VM/Host Rules**

Add... Edit... Delete

Name	Type	Enabled	Conflicts	Defined By
SPKVC	Run VMs on Hosts	Yes	0	User
SPK-admin	Run VMs on Hosts	Yes	0	User
spkidx	Separate Virtual Machines	Yes	0	User
spkadm1	Run VMs on Hosts	Yes	0	User
spkadm2	Separate Virtual Machines	Yes	0	User
spksh1	Run VMs on Hosts	Yes	0	User
spksh2	Separate Virtual Machines	Yes	0	User

**VM/Host Rule Details**

Virtual Machines that are members of the VM Group should run on hosts that are members of the Host Group.

Add... Remove

spkadm Group Members	compute-nodes Group Members
admin3	spk220-5.dmzhx.lab.cisco.com
admin5	spk220-4.dmzhx.lab.cisco.com
admin4	spk220-3.dmzhx.lab.cisco.com
admin6	spk220-6.dmzhx.lab.cisco.com
admin2	spk220-2.dmzhx.lab.cisco.com
admin1	spk220-1.dmzhx.lab.cisco.com

## Install SwiftStack Object Storage System on Cisco UCS S3260 Servers

For detailed information about the installation steps, please refer to the [Cisco UCS S3260 Storage Server with SwiftStack Software Defined Object Storage CVD](#).

The following are the high-level installation steps:

1. Connect Cisco UCS S3260 M5 chassis/servers (SwiftStack PACO nodes) and Cisco C220 M5 servers (SwiftStack controller nodes) to FI 6332 switches, enable the server ports and wait until the hardware has been discovered successfully.
2. Create the necessary resource pools and the necessary global policies in Cisco UCS Manager, prepare for the creation the server profiles.
3. Create the chassis profile template then create the chassis profiles from the template.
4. Associate the chassis profiles to the Cisco UCS S3260 chassis.
5. Create the storage profiles for the servers.
6. Create the service profile templates then create the service profiles from the template.
7. Associate the service profiles to the Cisco UCS S3260 M5 servers and Cisco UCS C220 M5 SwiftStack controller servers.
8. Install the Operating System on the SwiftStack Nodes.

Red Hat Operating System 7.3 was installed. Minimal Installation is fine.

9. Make Post Operating System Installation on the Nodes.
  - Check and ping the default gateway for North bound connection. This is needed to register the servers.

- Register the server and attach subscriptions.
- Update DNS server and /etc/hosts as needed in your setup for name resolution.
- Update /etc/rhsm/rhsm.conf with proxy entries in case you are behind the proxy server.
- The partition /home can be removed, and the space can be taken into the root partition if needed.

10. Install the additional packages. Not all packages were needed but they were installed for operational activities.

11. Install SwiftStack Software on the Controller nodes.

a. Make Pre-install checks.

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

b. Install On-Premise Controller Software.

There are two modes of installation for SwiftStack controller. The controller can be hosted on SwiftStack site and can be installed through <http://platform.swiftstack.com>. You can also install Controller On-Premise (the current scope of this CVD is limited to On-Premise controller only). Please Contact SwiftStack to obtain the On-Premise Controller software and verify your hardware requirements.

For detailed information about the software, log into [portal.swiftstack.com](http://portal.swiftstack.com) and check for On-Premise Controller installation under the Admin section.

c. Install Software.

Obtain the installer software and run the script in a Linux shell as shown below:

```

root@swiftcontroller ~]# ./SwiftStack-Controller-5.2.0.2-installer.sh
Extracting SwiftStack Controller 5.2.0.2
.....
997580 blocks

....
....
....

Installing SwiftStack Controller...
Preparing... #####
Updating / installing...
swiftstack-controller-5.2.0.2-1.el7 #####
**** SwiftStack Controller install details will be logged to /tmp/install-2017-04-06-
14:01:57.log
    
```

Controller install succeeded!

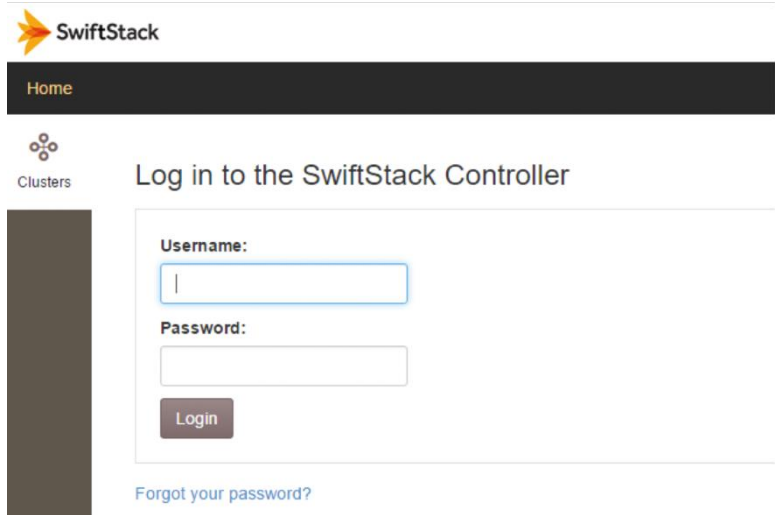
You must now complete setup by pointing a web browser at this server using HTTPS and the default port of 443. E.g.

<https://192.168.66.247/>

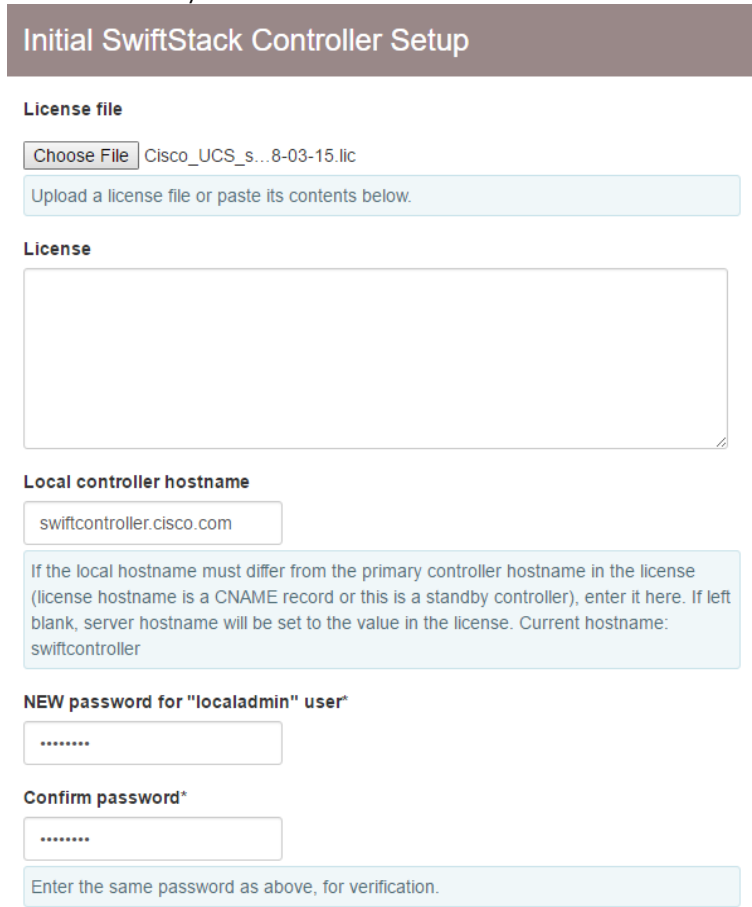
Log in with the username "localadmin" and the default password, "password"

d. Post-install Configuration.

- i. Log into the URL displayed after the installation with the user as localadmin and the default password.



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



It may take a few minutes for the setup to complete.

## Initial SwiftStack Controller Setup

### Current Status:

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

Continue

- iii. Click Continue and login with the new password; the system will prompt you to add the nodes and create the cluster. It may display as shown below to run the curl command on the storage nodes:

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

## 12. Install SwiftStack Software on Storage Nodes

Before running the installer software make sure that SSL certificates are installed. They could either be from commercial CA or Self-Signed. Self-Signed certificate will be generated on the controller node and needs to be copied to all the server nodes.

- a. On Controller Node:

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

- b. Copy this certificate to all Server nodes:

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

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

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

- d. Run the Curl command on the storage node:

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

After completing this command, the system will print the claim URL. This can also be printed by running 'sclaimurl' command on the server as shown below:

```
[root@swiftstack-server2 ~]# ssclaimurl
```

```
+-----+  
|  
| Your claim URL is:  
| https://swiftcontroller.cisco.com:443/claim/9723dbc2-61e1-11e7-8fcb-0025b500024f  
|  
+-----+
```

- e. Run the Curl Command on each storage node and get the ssclaimurl output. The nodes have to be claimed through http request to the controller.

13. Configure SwiftStack Controller for Nodes.

- a. Run the Claim URL in a browser to claim the nodes.

Claiming Node

UUID: 0f1cb554-09e7-11e7-a8aa-0025b50002ff  
(MAC address: 00:25:b5:00:02:ff)



- b. Enter the cluster name, select the Deployment Status, and click Create Cluster.

Create New Cluster:

Name\*

Deployment Status\*  
 ?

- c. Click Configure and configure the basic settings as shown and click Submit Changes.



**Basic Cluster Info**

Name\* Cluster1

Deployment Status\* Production

---

**Network Configuration**

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

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

How will you terminate SSL?  SSL will be terminated externally  SSL should be terminated on the Swift node

Cluster API Hostname\* swift-cluster.dmzlx.lab.cisco

SSL/TLS Protocols\*  TLSv1.0  TLSv1.1  TLSv1.2

SSL Cipher suite\* ECCDH-ECDH-AES256-GCM

SSL Certificate  No file chosen

SSL Private Key  No file chosen

---

**NTP Information**

NTP Server 1\* 192.168.0.104

NTP Server 2

NTP Server 3

NTP Server 4

---

**Advanced Options**

In the screenshot shown above, since a SwiftStack load balancer is used, you need to enter the name of the Load Balancer configured. For details about configuring SwiftStack Load Balancer, please refer to the SwiftStack documentation.



**Note:** It is recommended to use the same NTP server for SwiftStack nodes, HyperFlex nodes and Splunk nodes so the timing will be in sync across all the configurations. You might need to run the 'ntpdate <NTP server>' command manually on all the SwiftStack nodes to get time to be synced.

- d. Click Networks and provide the three network configurations and save them as rules. Every node ingested will inherit these rules by default.

**Manage Interface Configuration Rules**

See the [Network Rules documentation](#) for more information about this page.

**Outward Facing**

The outward-facing network primarily handles two types of traffic: incoming Object API (S3 / Swift) requests to your proxy servers and secure VPN traffic with the SwiftStack controller.

Interface rules that are associated with an [BRDNS Group](#) cannot be deleted, and can only be edited to be larger (have a smaller number of mask bits).

Subnet	Actions
192.168.11.0/24	Cannot delete rules associated with <a href="#">RRDNS Groups</a> .

**Cluster Facing**

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

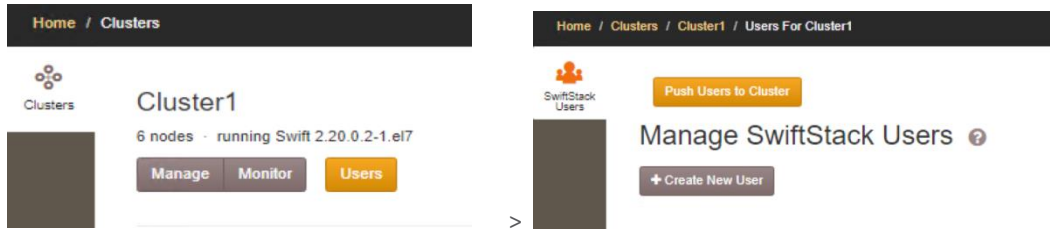
Subnet	Actions
192.168.13.0/24	<input type="button" value="X"/>

**Replication Facing**

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

Subnet	Actions
192.168.12.0/24	<input type="button" value="X"/>

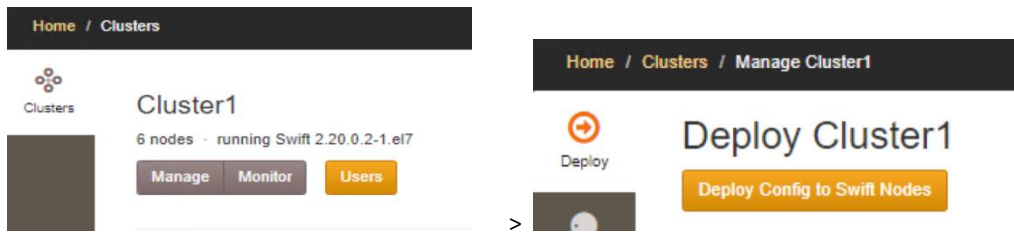
- e. Click Home > Clusters > User > Create New User to create SwiftStack users. These are the SwiftStack Cluster accounts and a minimum of one account is needed for the cluster. Create a backup user as needed.



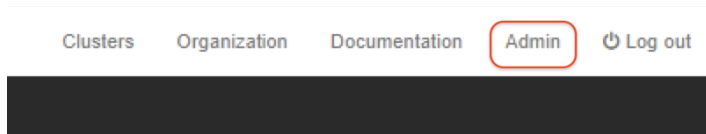
- f. Click Home > Clusters > Manage > Nodes on the left pane and ingest the nodes. Confirm the networks when prompted by the system for outward facing, cluster and replication networks.
- g. Click Setup of each node.
  - i. Select all the disks and format the drives.
  - ii. Select SSD disks (2 disks in single node and 1 disk in dual node configurations), click Add Policies and select 'Account and Container'.
  - iii. Select HDD disks (54 disks in Single Node and 27 disks in dual node configurations), Click Add Policies and select 'Standard-Replica'.

Drives	Device Path	Serial	Blink?	Size (GB = 10 <sup>9</sup> )	SSD	Mount	Policies
<input type="checkbox"/>	sda	35LBAB31T2V7	No	400.1 GB	Yes	/srv/node/d12	Account & Container
<input type="checkbox"/>	sdq	AA996LGH	No	12000.1 GB	No	/srv/node/d0	Standard-Replica
<input type="checkbox"/>	sdx	AA992T1H	No	12000.1 GB	No	/srv/node/d1	Standard-Replica
<input type="checkbox"/>	sdz	AA995GVH	No	12000.1 GB	No	/srv/node/d2	Standard-Replica
<input type="checkbox"/>	sdr	8HKMNL5H	No	12000.1 GB	No	/srv/node/d3	Standard-Replica
<input type="checkbox"/>	sdaa	AA99DR6H	No	12000.1 GB	No	/srv/node/d4	Standard-Replica
<input type="checkbox"/>	sdaa	AA99SH3H	No	12000.1 GB	No	/srv/node/d5	Standard-Replica

- iv. In the left menu, enable the node.
- v. This completes the setup of one node. Repeat the procedure for all the storage nodes.
- h. Go to Home > Clusters > Manage > Deploy and click Deploy Config to SwiftStack Nodes to deploy the configuration to the nodes.



- 14. Click Home > Admin tab and update the default values as needed in your setup.



- 15. Click Backups in Admin page. Enter the backup settings and click Validate Swift Credentials and submit the changes and then queue a backup job.

## Backup Settings

Backup period hours*	<input type="text" value="6"/>
	<small>Backup DB &amp; configuration every X hours (up to 24; defaults to 6)</small>
Backup local retention days*	<input type="text" value="4"/>
	<small>Number of days worth of local backups to keep (&gt;= 1)</small>
Metrics sync period*	<input type="text" value="360"/>
	<small>Sync metrics every X minutes (&gt;= 30)</small>
	<input type="checkbox"/> Save backups to Swift
	<input type="button" value="Save backups in a Swift cluster"/>
Auth URL for Swift cluster	<input type="text" value="http://swift-cluster.dmhix.lab."/>
	<small>The Swift v1 Auth URL to connect to</small>
Swift username	<input type="text" value="controller_config_backups"/>
	<small>The Swift v1 Auth User</small>
Swift password	<input type="password" value="*****"/>
	<small>The Swift v1 Auth Key/password</small>
Chunk size	<input type="text" value="512"/>
	<small>Store backups in chunks this size; actual stored objects will be smaller due to compression (MiB)</small>
Concurrency	<input type="text" value="2"/>
	<small>Use this many threads when compressing/uploading or decompressing/downloading</small>

## 16. Install and configure the Standby controller for SwiftStack nodes.

The Standby Controller is always in passive node. The primary and standby are not in Active/Active, but in Active/Passive mode. Hence the standby controller is configured and left as is. Refer to the [SwiftStack on S3260 CVD](#) on how to activate the Standby Controller in case of failure of Primary Controller.

- a. Follow step 11 to install the SwiftStack software on the standby controller by running the command `./SwiftStack-Controller-5.2.0.2-installer.sh`

The installation should complete with the message as:

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

`https://192.168.66.248/`

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

- b. After entering the above URL in the browser, make sure to check 'This is a Standby Controller' as shown below. Enter the hostname same as the primary controller.

**Local controller hostname**

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

**NEW password for "localadmin" user**

**Confirm password\***

Enter the same password as above, for verification.

Use insecure fake entropy

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

This is a standby controller

17. Go to the SwiftStack controller’s Monitoring and Metrics to check the cluster’s health.

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

a. Click Home > Clusters > Monitor > Server Stats to check the server statistics.



b. Click Home > Clusters > Monitor > Storage Stats to check the storage statistics.



18. The SwiftStack nodes' alerts can be managed through the controller and alerts can be acknowledged and then archived as desired. From the Home page, click the red flagged 'New Alerts' to view.

The Alerts page shows the following information:

- Alert 1:** Services on peer nodes unreachable! One or more peer services are not reachable from mtl4.dnchx.lab.cisco.com (25ad019-3431-11e9-b7c2-0025650004f). NOT OK: IP service location(s): 192.168.13.242, 6006, 6006, 6007, 6008, 6009, 6010, 6012, 6013, 6014, 6015, 6016, 6017, 6018, 6019, 6020, 6021, 6022, 6023, 6024, 6025, 6026, 6027, 6028, 6029, 6030, 6031, 6032 not reachable. Began: Sun 25 Apr 2019 07:02:44 AM UTC. Ended: Sun 28 Apr 2019 07:03:44 AM UTC.
- Alert 2:** Services on peer nodes unreachable! One or more peer services are not reachable from mtl2.dnchx.lab.cisco.com (9564d75c-3962-11e9-ace9-0025650004f). NOT OK: IP service location(s): 192.168.12.242, 6004 not reachable. Began: Tue 23 Apr 2019 06:08:57 PM UTC. Ended: Tue 28 Apr 2019 06:10:03 PM UTC.
- Alert 3:** Services on peer nodes unreachable! One or more peer services are not reachable from mtl5.dnchx.lab.cisco.com (932c365c-3963-11e9-bf46-0025650004f). NOT OK: IP service location(s): 192.168.12.246, 6004, 192.168.13.244, 50310. Began: Tue 23 Apr 2019 06:08:47 PM UTC. Ended: Tue 23 Apr 2019 06:09:49 PM UTC.

19. When an alert is received, you can review the alert guide in the SwiftStack Controller documentation <https://www.swiftstack.com/docs/>. Click Alerts/Events section to get more information about the alerts.

## Configuration and Validation

---

### Configure Splunk Enterprise Cluster

This section explains how to configure the roles of the Splunk virtual machines to provide the corresponding services in the Splunk Enterprise cluster. In this CVD, four (4) clustered Search Heads, ten (10) clustered indexers, a deployment server, a search head cluster deployer, a master node where the distributed monitoring console sits, and a license master are configured.

The installation order is as follows:

1. Configure License Master
2. Configure Master Node
3. Configure Indexing Cluster
4. Configure Deployer
5. Configure Search Head Cluster
6. Configure Distributed Monitoring Console
7. Configure Deployment Server
8. Install universal forwarder (Testing Deployment Server)
9. Configure SmartStore with S3 Storage

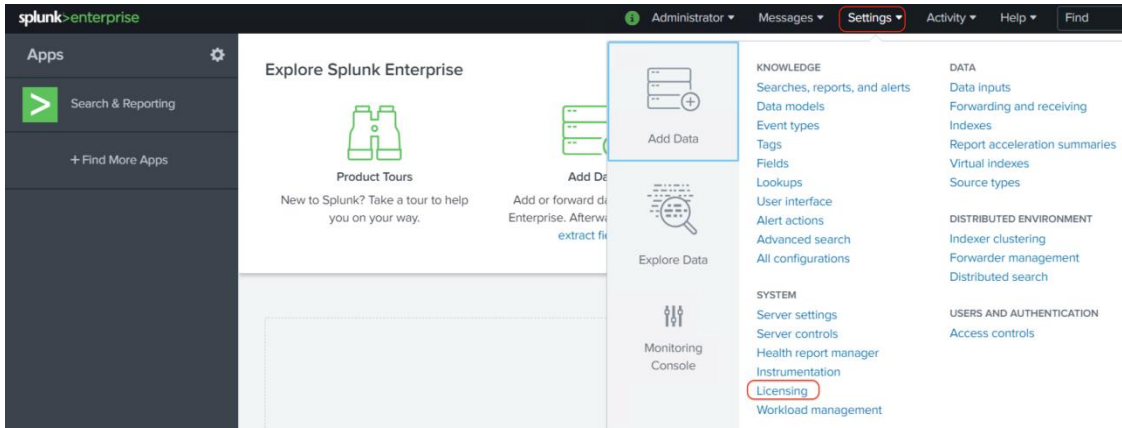
### Splunk Enterprise Licenses

The servers in the Splunk Enterprise infrastructure that performs indexing must be licensed. Any Splunk can be configured perform the role of license master. In this CVD, the admin node (admin2) is configured to be the license master and all the other Splunk instances are configured as license slaves.

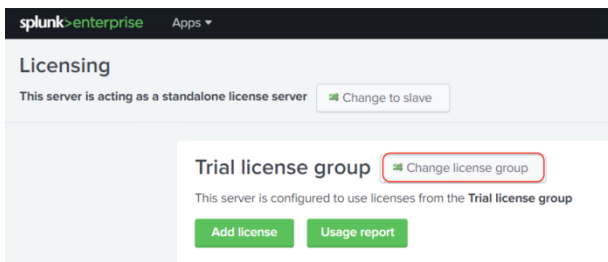
#### Setup License Master

To configure the server admin2 as the central license master, follow these steps:

1. Log into the server admin2 as user admin.
2. Navigate to the licensing screen by clicking on Settings > System > Licensing.



3. Click Change License Group.



4. Click Enterprise license radio button. Click Save.

### Change license group

The type of license group determines what sorts of licenses can be used in the pools on this license server. [Learn more](#)

- Enterprise license
 

This license adds support for multi-user and distributed deployments, alerting, role-based security, single sign-on, scheduled PDF delivery, and unlimited data volumes.

There are no valid Splunk Enterprise licenses installed. You will be prompted to install a license if you choose this option.
- Forwarder license
 

Use this group when configuring Splunk as a forwarder. [Learn more](#)
- Free license
 

Use this group when you are running Splunk Free. This license has no authentication or user and role management, and has a 500MB/day daily indexing volume. [Learn more](#)
- Enterprise Trial license
 

This is your included download trial. IMPORTANT: If you switch to another license, you cannot return to the Trial. You must install an Enterprise license or switch to Splunk Free.

Cancel Save

5. In the Add new license dialog, click Choose File to select your license file. Click Install to install the license.

**Add new license**

Learn more about your license options at the [licensing section](#) on splunk.com.

To install a license, upload a license file here (license files end with .license):

Splunk.License

Or, copy & paste the license XML directly...

6. Click Restart.

**✓ Add successful**

The new license was successfully added.  
You must restart Splunk for changes to take effect.

7. Click OK to restart Splunk to complete the license installation.
8. Log back into Splunk. If “are you sure you want to restart Splunk” is still visible, click Cancel.

For more information about Splunk Enterprise licensing, please refer to the Splunk Documentation.

**Configure the Indexers, Search Heads, and Admin Nodes as License Slaves**

Configure all the other Splunk instances to be the License slaves to the Splunk License master, for example, admin2. This can be performed by following one of the two methods: 1) The first and preferred method is to use ClusterShell command (clush) to configure all the Splunk instances to be license slaves to the license master. 2) The second method is to configure each node as a license slave individually from the respective Web UI. This document only shows how to configure the license slaves using CLI (clush).

To configure the indexers, search heads, and admin nodes, follow these steps:

1. Log into the admin node as user 'splunk' and execute the command:

```
[splunk@admin ~]$ clush --group=all -x admin2 -B /data/disk1/splunk/bin/splunk edit licenser-localslave -master_uri https://admin2:8089 -auth admin:Cisco123
[splunk@admin tmp]$ clush --group=all -x admin2 -B /data/disk1/splunk/bin/splunk edit licenser-localslave -master_uri https://admin2:8089 -auth admin:Cisco123
admin[1,3-6],idx[1-10],sh[1-4] (19)
The licenser-localslave object has been edited.
[splunk@admin tmp]$
```

2. Restart Splunk service on the slave nodes.

```
[splunk@admin ~]$ clush --group=all -x admin2 -B sudo /usr/bin/systemctl restart Splunkd.service
[splunk@admin tmp]$ clush --group=all -x admin2 -B sudo /usr/bin/systemctl restart Splunkd.service
[splunk@admin tmp]$
```

**Verify License-Slave Relationships**

To verify the license-slave relationships, follow these steps:

1. Go to the master (admin2) node’s Splunk GUI, and navigate to Settings > Licensing.



2. Click All Indexer Details to view the license slaves.

### Local server information

Indexer name	admin2
Volume used today	0 MB
Warning count	0
Debug Information	<a href="#">All license details</a> <a href="#">All Indexer details</a>



There should be twenty license slaves listed, for example, ten indexers; four search heads plus six admin nodes including License Master node itself.

### Connected indexers

Licensing > Connected indexers

#### Indexers connected to: admin2 (20)

##### 1. sh1

active_pool_names	<ul style="list-style-type: none"> <li>• auto_generated_pool_enterprise</li> </ul>
added_usage_parsing_warnings	None
label	sh1
pool_names	<ul style="list-style-type: none"> <li>• auto_generated_pool_enterprise</li> <li>• auto_generated_pool_forwarder</li> <li>• auto_generated_pool_free</li> </ul>
stack_names	<ul style="list-style-type: none"> <li>• enterprise</li> <li>• forwarder</li> <li>• free</li> </ul>
warning_count	0

##### 2. sh2

active_pool_names	<ul style="list-style-type: none"> <li>• auto_generated_pool_enterprise</li> </ul>
added_usage_parsing_warnings	None
label	sh2
pool_names	<ul style="list-style-type: none"> <li>• auto_generated_pool_enterprise</li> <li>• auto_generated_pool_forwarder</li> <li>• auto_generated_pool_free</li> </ul>



The License Master counts all the license slaves as Splunk Indexer instances in spite of the actual roles the instances have been configured to perform.

## Configure Index Cluster

An indexer cluster is a group of Splunk Enterprise instances, or nodes, that, working in concert, provide a redundant indexing and searching capability. The parts of an indexer cluster are:

- A master node to manage the cluster
- A number of peer nodes to index and maintain multiple copies of the data and to search the data.

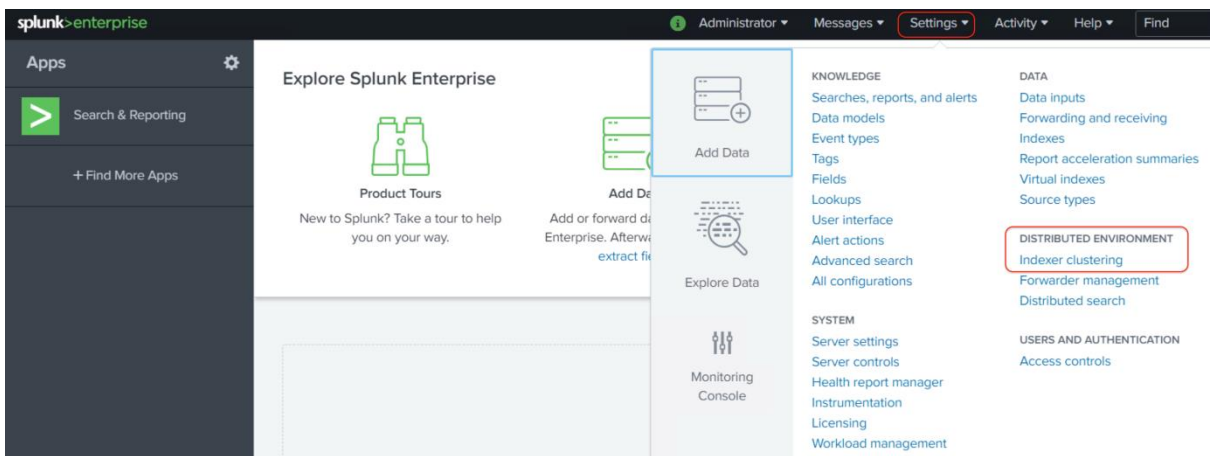
- One or more search heads to coordinate searches across the set of peer nodes

The Splunk Enterprise indexers of an indexer cluster are configured to replicate each other’s data, so that the system keeps multiple copies of all data. This process is known as index replication. The number of copies is controlled by a parameter known as replication factor. By maintaining multiple, identical copies of Splunk Enterprise data, clusters prevent data loss while promoting data availability for searching. Indexer clusters feature automatic failover from one indexer to the next. This means that, if one or more indexers fail, incoming data continues to get indexed and indexed data continues to be searchable. For more information please refer to [Splunk Documentation].

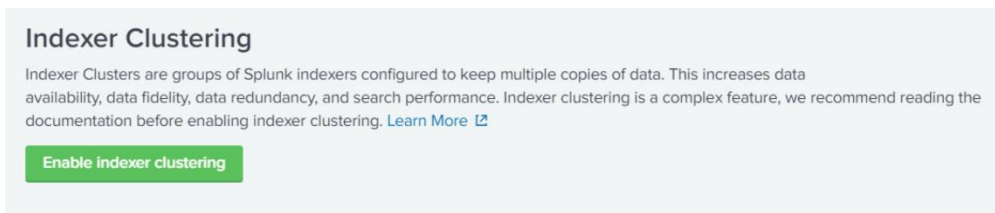
### Configure the Master Node (aka: Cluster Master)

To configure the admin node admin1 as the Indexer Cluster master, follow these steps:

1. Using the web browser, go to the master node (admin1) `http://hostname-or-IP:8000/` (such as `https://admin1:8000/`).
2. Click the Settings > Distributed Environment > Indexer Clustering.



3. Click Enable Indexer Clustering.



4. Select Master Node and click Next.

## Enable Clustering ×

- Master node  
The master node coordinates the activities of the peer nodes. It does not store or replicate data (aside from its own internal data).
- Peer node  
Peer nodes receive and index incoming data. They also replicate data from other nodes in the cluster.
- Search head node  
The search head manages searches across one or more clusters.

- Set the fields Replication Factor to be 2 and Search Factor to be 2. Enter your Security Key and Cluster Label for the cluster. Click Enable Master Node button.

## Master Node Configuration ×

Replication Factor	2	The number of copies of raw data that you want the cluster to maintain. A higher replication factor protects against loss of data if peer nodes fail.
Search Factor	2	The number of searchable copies of data the cluster maintains. A higher search factor speeds up the time to recover lost data at the cost of disk space. Must be less than or equal to Replication Factor.
Security Key	.....	This key authenticates communication between the master and the peers and search heads.
Cluster Label	HXSplunk	Name your cluster using this field. This label is also used to identify this cluster in the Monitoring Console.



Replication and Search factors vary by deployment. The replication factor indicates the number of copies to be maintained on the indexers. The search factor indicates how many of those copies will return search results. In the configuration shown above, one indexer could be down and searches still return all results. If the configuration needs to be more resilient, the replication factor may be increased, but will also increase disk consumption. Refer to the Splunk documentation for more information:

<http://docs.splunk.com/Documentation/Splunk/latest/Indexer/The replication factor> .

- Click Restart Now to restart the Splunk service as indicated.

## Restart Required



You must restart Splunk for the master node to become active.

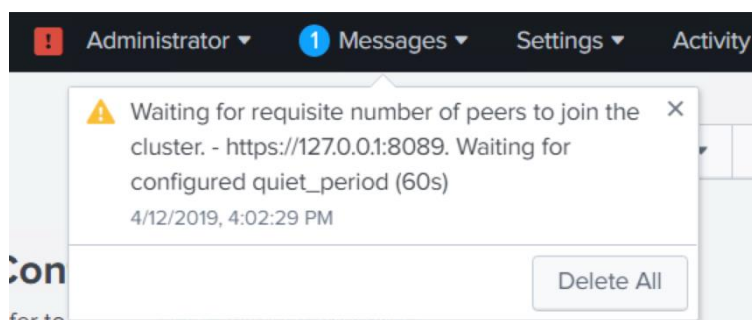
Restart Later

Restart Now

7. Wait until the Restart Successful message appears, click OK to go back to the Login screen.
8. Log in as the admin user.



A message appears indicating that necessary number of peers must join the cluster. For a replication factor of 2, Splunk Enterprise needs a minimum of 2 peers. Ignore the message. The master node is ready to accept peers now.



### Configure Indexing Peers

Add the Splunk indexer instances as the indexing peers to the Cluster Master, for example, admin1. This can be performed by following one of the two methods described below. This can be performed by following one of the two methods: 1) The first and preferred method is to use ClusterShell command (clush) to configure all the Splunk indexer virtual machines to be indexing peers to the cluster master. 2) The second method is to configure each Splunk indexer virtual machine as an indexing peer individually by accessing the respective Web UI. This CVD only shows how to configure the indexing peers using CLI (clush).

To configure indexing peers, follow these steps:

1. From the admin node, as the 'splunk' users, issue the command:

```
[splunk@admin ~]$ clush --group=spkidx -B $SPLUNK_HOME/bin/splunk edit cluster-config -
mode slave -master_uri https://admin1:8089 -replication_port 8080 -secret Cisco123 -auth
admin:Cisco123
```

2. After editing the cluster configuration, restart the effected virtual machines:

```
[splunk@admin ~]$ clush --group=spkidx -B sudo /usr/bin/systemctl restart Splunkd.service
```

3. After all the splunk process in peer nodes are restarted, check the Master node's (i.e. admin1) web UI. The Master node must report number of available peers.

<p>✓ All Data is Searchable</p> <p>10 searchable Peers    0 not searchable Peers</p>		<p>✓ Search Factor is Met</p>		<p>✓ Replication Factor is Met</p> <p>2 searchable Indexes    0 not searchable Indexes</p>	
<p>Peers (10)    Indexes (2)    Search Heads (1)</p> <p>filter <input type="text"/> 10 per page ▼</p>					
Peer Name	Fully Searchable	Status	Buckets		
> idx10	✓ Yes	Up	40		
> idx9	✓ Yes	Up	40		
> idx4	✓ Yes	Up	40		
> idx8	✓ Yes	Up	40		
> idx7	✓ Yes	Up	43		
> idx6	✓ Yes	Up	34		
> idx1	✓ Yes	Up	36		
> idx3	✓ Yes	Up	35		
> idx5	✓ Yes	Up	39		
> idx2	✓ Yes	Up	33		



When the indexers are added to the cluster, it is not advised to use the command '\$SPLUNK\_HOME/bin/splunk re-start' on individual indexers. Refer to the documentation for detailed information: <http://docs.splunk.com/Documentation/Splunk/latest/Indexer/Restartthecluster>.

- Verify that all ten (10) indexers should appear as searchable and the Search Factor and Replication Factor are met.

### Configure Receiving on the Peer Nodes

In order for the indexers (aka peer nodes) to receive data from the forwarders, the inputs.conf file of all the indexers needs to be configured with a stanza to enable the tcp port 9997. This is done by editing a special purpose app's inputs.conf file in the cluster master (admin1).

To configure receiving on the peer nodes, follow these steps:

- On the command line of the master node (admin1), navigate to \$SPLUNK\_HOME/etc/master-apps/\_cluster/local.
- Create and edit the file 'inputs.conf' with the following content:

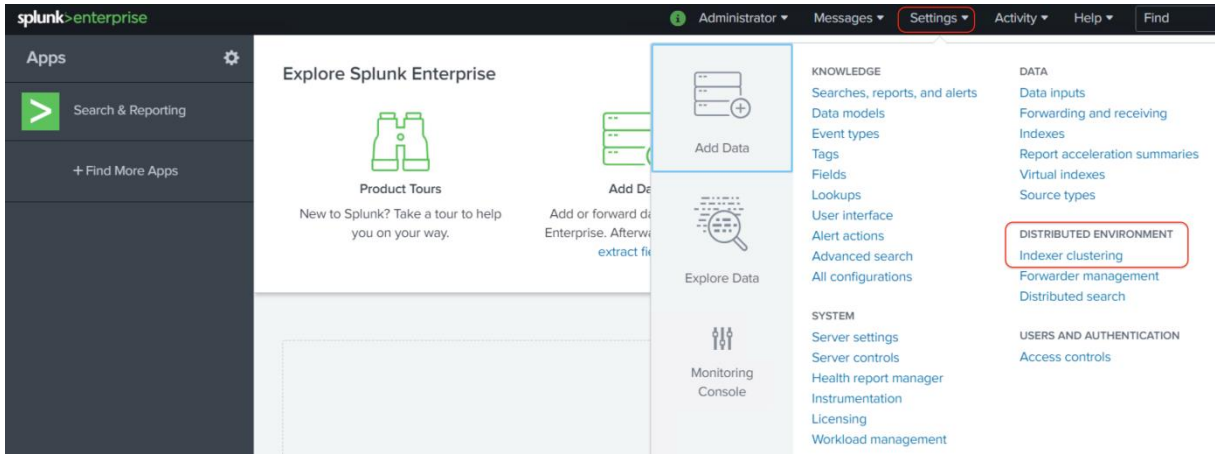
```
[splunktcp://:9997]
connection_host = ip
```

```
[splunk@admin1 local]$ cd $SPLUNK_HOME/etc/master-apps/_cluster/local
[splunk@admin1 local]$ vi inputs.conf
[splunk@admin1 local]$
[splunk@admin1 local]$ cat inputs.conf
[splunktcp://:9997]
connection_host = ip
```

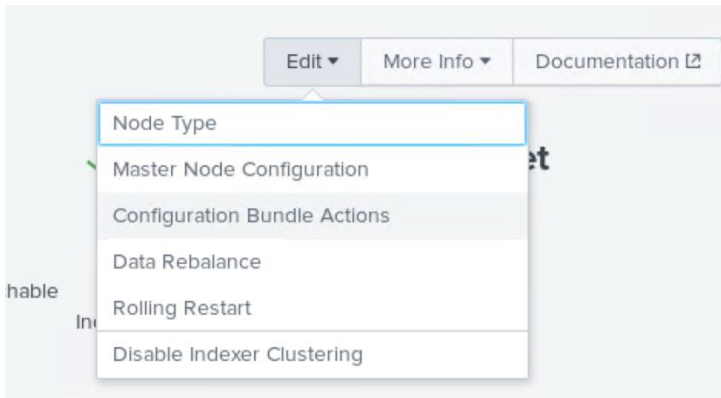


if this configuration uses DNS, edit 'connection\_host = dns'.

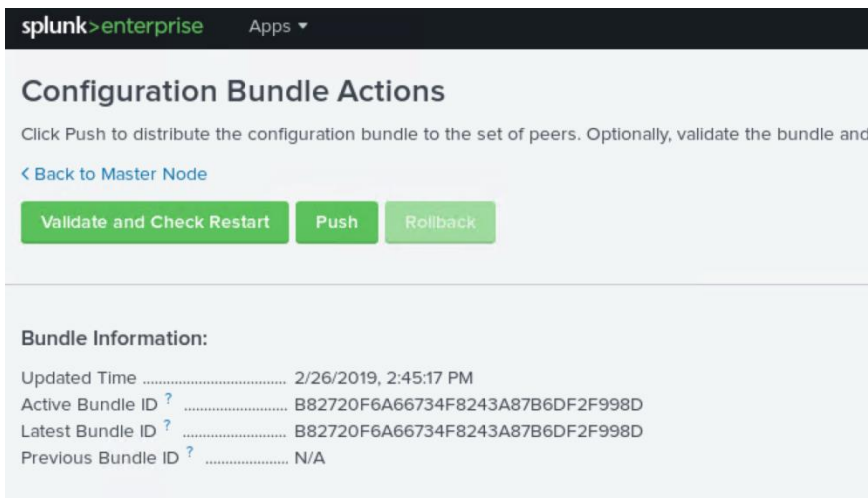
- Open the admin1 web management interface through browser. Navigate to 'Settings> Distributed Environment > Indexer Clustering'.



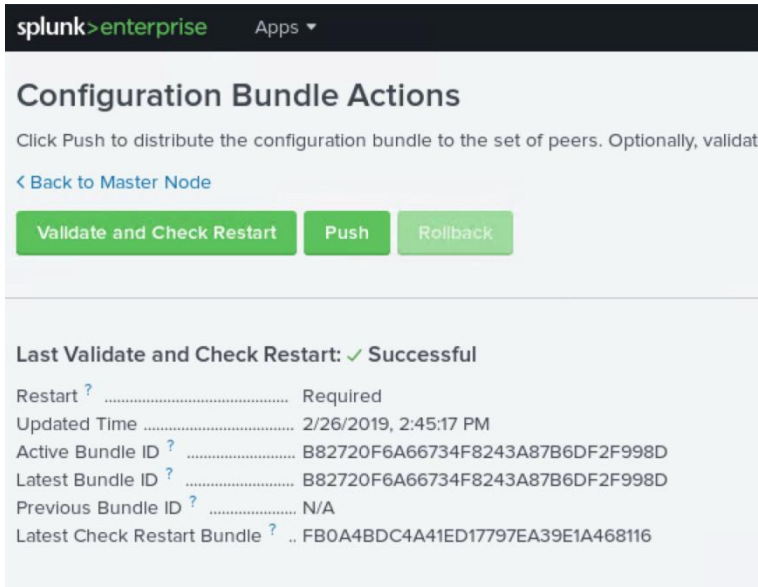
4. Select Edit > Configuration Bundle Actions.



5. Select Validate and Check Restart option.



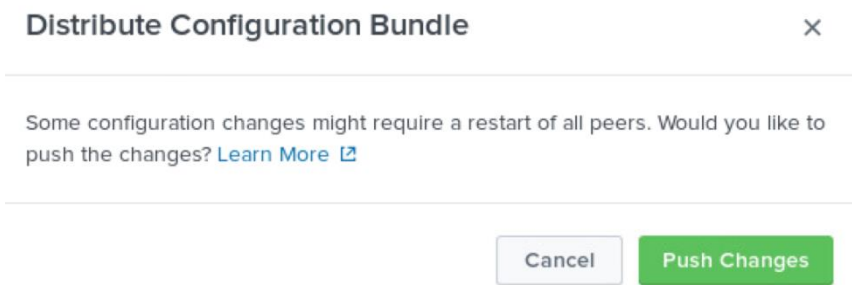
6. Wait until check Successful.



The screenshot shows the 'Configuration Bundle Actions' page in Splunk Enterprise. At the top, there is a navigation bar with 'splunk > enterprise' and 'Apps' with a dropdown arrow. Below the navigation bar, the page title 'Configuration Bundle Actions' is displayed. A brief instruction reads: 'Click Push to distribute the configuration bundle to the set of peers. Optionally, validate'. A link '< Back to Master Node' is provided. Three green buttons are visible: 'Validate and Check Restart', 'Push', and 'Rollback'. Below these buttons, a section titled 'Last Validate and Check Restart: ✓ Successful' contains a list of configuration details:

Restart ?	Required
Updated Time	2/26/2019, 2:45:17 PM
Active Bundle ID ?	B82720F6A66734F8243A87B6DF2F998D
Latest Bundle ID ?	B82720F6A66734F8243A87B6DF2F998D
Previous Bundle ID ?	N/A
Latest Check Restart Bundle ?	.. FB0A4BDC4A41ED17797EA39E1A468116

- 7. Select Push. Acknowledge the warning, and Push Changes.

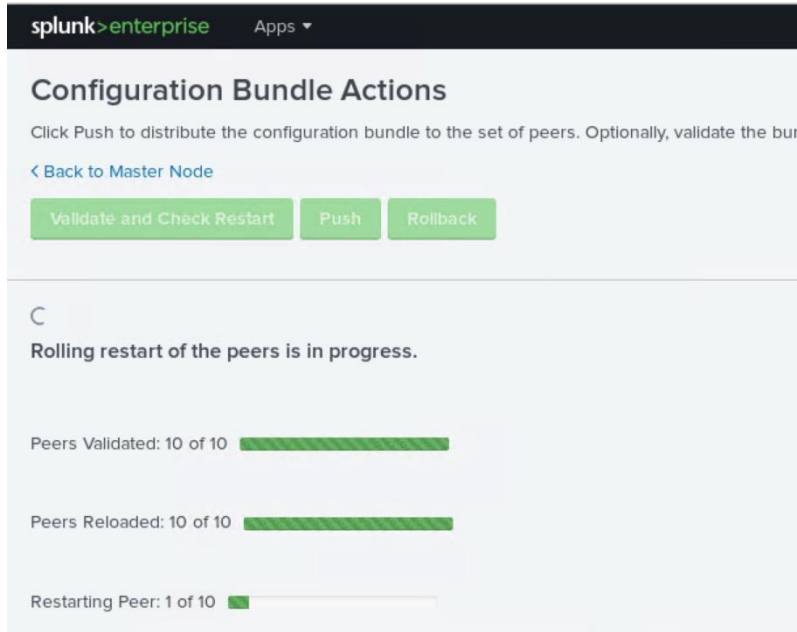


The screenshot shows a dialog box titled 'Distribute Configuration Bundle' with a close button (X) in the top right corner. The main text reads: 'Some configuration changes might require a restart of all peers. Would you like to push the changes? [Learn More](#)'. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Push Changes'.

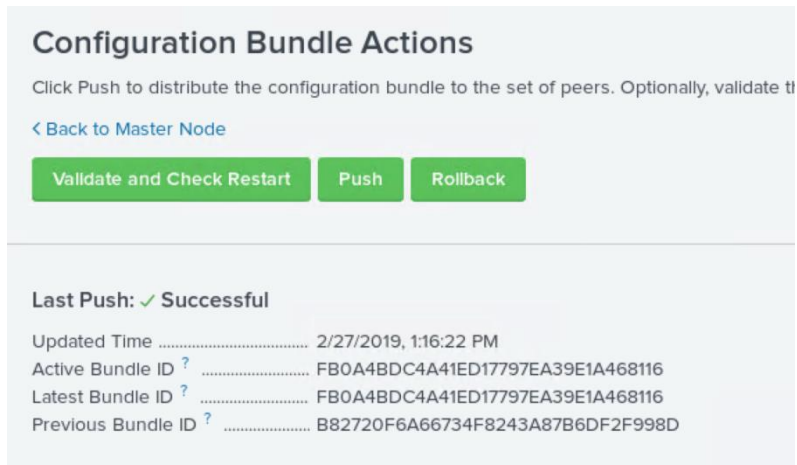
---

 The configuration changes are pushed to all the indexing peers. Rolling restart of all the peers are in progress.

---



8. When Push is complete, the GUI should reflect a successful push.



9. Log into one of the indexers through SSH.
10. Navigate to \$SPLUNK\_HOME/etc/slave-apps/\_cluster/local.
11. Verify that the file 'inputs.conf' has been pushed to the indexers.

```
[splunk@idx1 ~]$ cd /data/disk1/splunk/etc/slave-apps/_cluster/local
[splunk@idx1 local]$ ls -lts
total 16
4 -rw-rw-r-- 1 splunk splunk 2140 Apr 17 12:24 indexes.conf
4 -rw-rw-r-- 1 splunk splunk 1799 Apr 17 12:24 indexes.conf.bkp
4 -rw-rw-r-- 1 splunk splunk 41 Apr 17 12:24 inputs.conf
4 -r--r--r-- 1 splunk splunk 233 Apr 17 12:24 README
[splunk@idx1 local]$ cat inputs.conf
[splunktcp://:9997]
connection_host = ip
```



## Configure Master to Forward All Data to the Indexer Layer

It is a best practice to forward all master node internal data to the indexer (peer node) layer. This has several advantages: It enables diagnostics for the master node if it goes down. The data leading up to the failure is accumulated on the indexers, where a search head can later access it. The preferred approach is to forward the data directly to the indexers, without indexing separately on the master. You do this by configuring the master as a forwarder. These are the main steps: 1) Make sure that all necessary indexes exist on the indexers. This is normally the case, unless you have created custom indexes on the master node. Since `_audit` and `_internal` exist on indexers as well as the master, there is no need to create separate versions of those indexes to hold the corresponding master data. 2) Configure the master as a forwarder. Create an `outputs.conf` file on the master node that configures it for load-balanced forwarding across the set of peer nodes. The indexing function on the master must also be turned off, so that the master does not retain the data locally as well as forward it to the peers.

In the cluster master node `admin1`, follow these steps:

1. Create `'outputs.conf'` file in the master node at `$SPLUNK_HOME/etc/system/local` directory.
2. Create an `outputs.conf` file with the following content:

```
#Turn off indexing on the master
[indexAndForward]
index = false
[tcput]
defaultGroup = search_peers
forwardedindex.filter.disable = true
indexAndForward = false
[tcput:search_peers]
server=idx1:9997,idx2:9997,idx3:9997,idx4:9997,idx5:9997,idx6:9997,idx7:9997,idx8:9997
autoLB = true
```

3. Restart Splunk service on `admin1`.

```
[splunk@admin1 local]$ cd $SPLUNK_HOME/etc/system/local
[splunk@admin1 local]$ vi outputs.conf
[splunk@admin1 local]$
[splunk@admin1 local]$ cat outputs.conf
#Turn off indexing on the master
[indexAndForward]
index = false
[tcput]
defaultGroup = search_peers
forwardedindex.filter.disable = true
indexAndForward = false
[tcput:search_peers]
server=idx1:9997,idx2:9997,idx3:9997,idx4:9997,idx5:9997,idx6:9997,idx7:9997,idx8:9997,idx9:9997,idx10:9997
autoLB = true
[splunk@admin1 local]$ sudo /usr/bin/systemctl restart Splunkd.service
[splunk@admin1 local]$
```

## Configure Search Head Cluster

A search head cluster is a group of Splunk Enterprise search heads that serves as a central resource for searching. The members of a search head cluster are essentially interchangeable. You can run the same searches, view the same dashboards, and access the same search results from any member of the cluster.

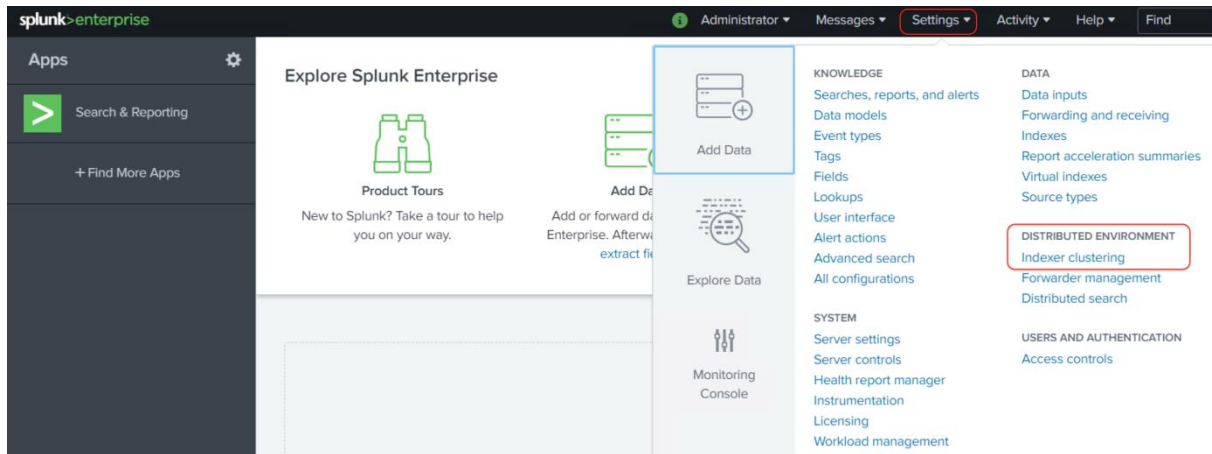


In order to take full advantage of the search head cluster, it is required to utilize a virtual or physical load balancer according to the enterprises standards. Due to variability, the operator is suggested to use their own discretion in installing and configuring this.

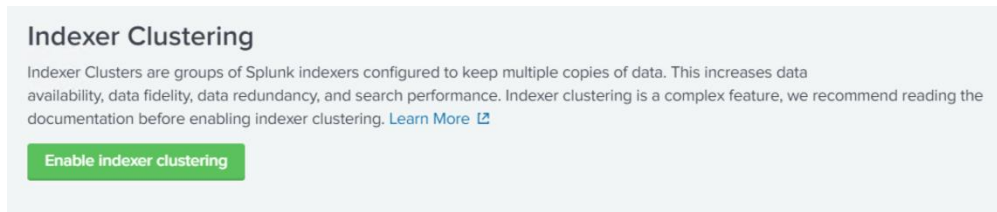
### Add Search Heads to Master Node

A Splunk Enterprise instance can be configured as a search head through the Indexer clustering feature. To add search heads to the master node, follow these steps:

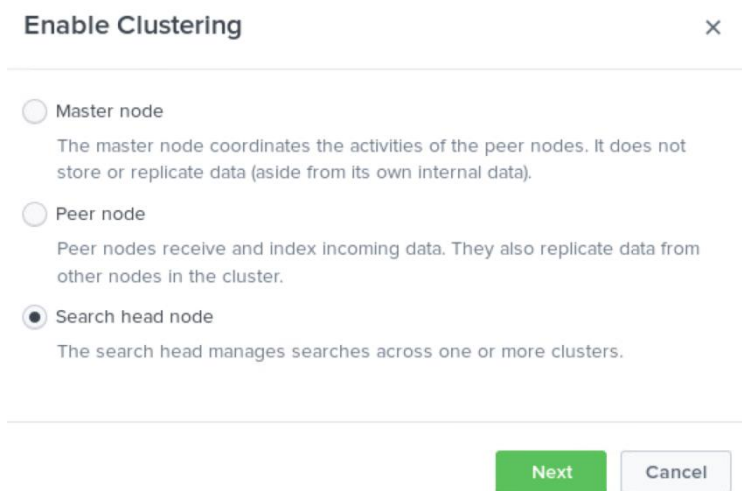
1. Log into one of the search heads as user admin.
2. Navigate to Settings> Distributed Environment > Indexer Clustering.



3. Click Enable Indexer Clustering.



4. In the Enable Clustering dialog box, click Search head node. Click Next.



5. Enter the Master URI in the format [https://<hostname\\_or\\_IP>:<master\\_port\\_number>](https://<hostname_or_IP>:<master_port_number>). (such as <https://admin1:8089>) The default Master port number is 8089.

6. Enter the same security key that was used while configuring the master node.

### Search head node configuration ×

---

Master URI   
E.g. https://10.152.31.202:8089 This can be found in the Master Node dashboard.

Security key   
This key authenticates communication between the master and search head.

---

7. Click Enable search head node.
8. Click Restart Now to restart Splunk service as indicated.

### Restart Required ×

---

You must restart Splunk for the search node to become active.

---

9. Wait until Restart Successful message appears, click OK to go back to the Login screen.
10. Repeat steps 1-9 to configure all four servers with hostnames sh1, sh2, sh3 and sh4 to be search heads.
11. Verify the search head cluster members in the master node, by navigating to Settings > Distributed Environment > Indexer clustering.
12. Click Search Heads tab.

The screenshot shows the Splunk Enterprise Admin Console interface for Indexer Clustering: Master Node. At the top, there are navigation tabs for Administrator, Messages, Settings, Activity, Help, and Find. Below the title, there are three status indicators, each with a green checkmark: "All Data is Searchable", "Search Factor is Met", and "Replication Factor is Met". Under "All Data is Searchable", it shows "10 searchable" and "0 not searchable" Peers. Under "Replication Factor is Met", it shows "2 searchable" and "0 not searchable" Indexes. Below these indicators, there are tabs for Peers (10), Indexes (2), and Search Heads (5). The Search Heads tab is active, showing a table with columns for Search head name and Status. The table lists five search heads: sh1, sh2, admin1, sh4, and sh3, all with a status of "Up".

### Configure Search Head Cluster members

To configure the search head cluster members, follow these steps:

1. Log into the search head virtual machine sh1 as the user 'splunk'.
2. Enter the following commands to make this search head join the search head cluster.

```
[splunk@sh1 ~]$ $SPLUNK_HOME/bin/splunk init shcluster-config -auth admin:Cisco123 -
mgmt_uri https://sh1:8089 -replication_port 18081 -replication_factor 2 -
conf_deploy_fetch_url https://admin1:8089 -secret Cisco123 -shcluster_label hxshcluster
[splunk@sh1 ~]$ $SPLUNK_HOME/bin/splunk init shcluster-config -auth admin:Cisco123 -mgmt_uri
https://sh1:8089 -replication_port 18081 -replication_factor 2 -conf_deploy_fetch_url https
://admin1:8089 -secret Cisco123
Search head clustering has been initialized on this node.
You need to restart the Splunk Server (splunkd) for your changes to take effect.
```



The `-shcluster_label` parameter is optional but is useful for identifying the cluster in the monitoring console. If you did not configure the label for the search cluster, you will get a warning message while setting up the Distributed Monitoring Console. If you configure the label on one member, you must configure it with the same value on all members, as well as on the deployer. See the [Splunk Documentation](#) about how to set the cluster labels if you did not set it at this step.

3. Restart Splunk service on the Search Head sh1.

```
[splunk@sh1 ~]$ sudo /usr/bin/systemctl restart Splunkd.service
[splunk@sh1 ~]$ sudo /usr/bin/systemctl restart Splunkd.service
[splunk@sh1 ~]$
```

4. Repeat steps 1-3 for all the search heads.

### Select a Search Head Captain

A search head cluster consists of a group of search heads that share configurations, job scheduling, and search artifacts. The search heads are known as the cluster members. One cluster member has the role of captain, which means that it coordinates job scheduling and replication activities among all the members. It also serves as a search head like any other member, running search jobs, serving results, and so on. Over time, the role of captain can shift among the cluster members.

A search head cluster uses a dynamic captain. This means that the member serving as captain can change over the life of the cluster. Any member has the ability to function as captain. When necessary, the cluster holds an election, which can

result in a new member taking over the role of captain. The procedure described in this section helps bootstrap the election process.

To elect a search head captain, follow these steps:

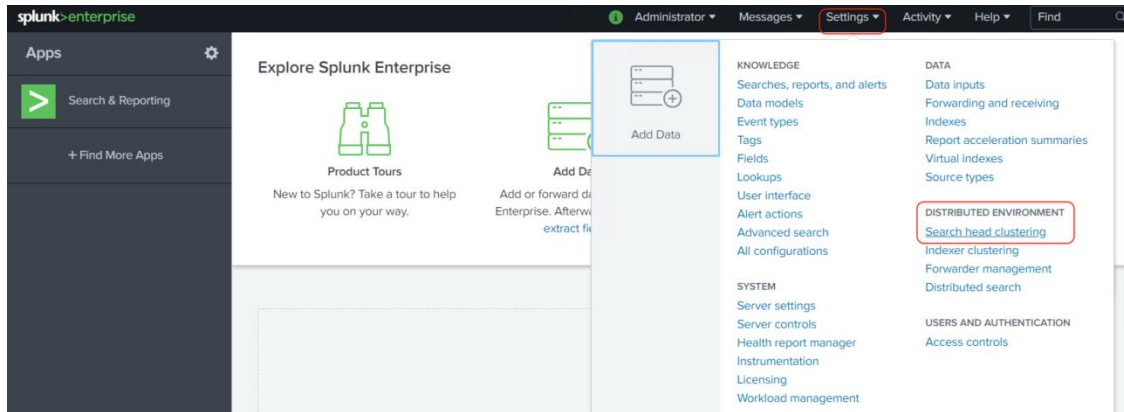
1. Log into any one of the search heads as the user 'splunk'.
2. Start the search head captain election bootstrap process by running the following command:

```
$SPLUNK_HOME/bin/splunk bootstrap shcluster-captain -servers_list
"https://sh1:8089,https://sh2:8089,https://sh3:8089, https://sh4:8089" -auth
admin:Cisco123
```

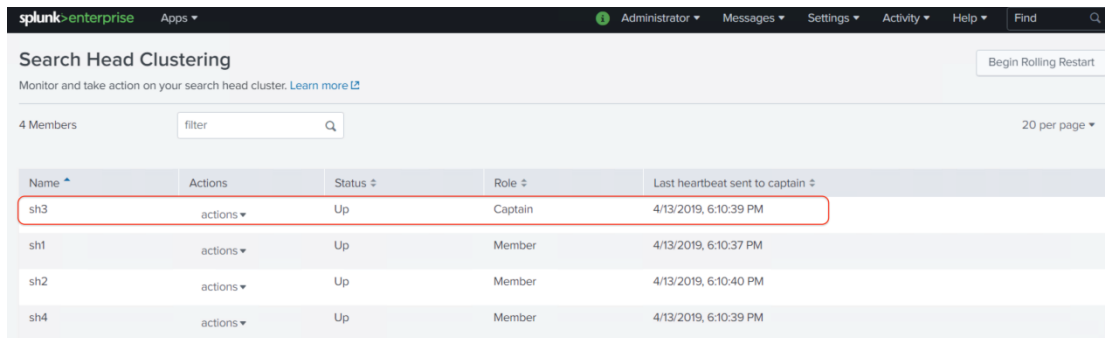


The search head captain election process can be started from any of the search head cluster members.

3. Log into one of the search heads GUI as user admin. Navigate to 'Settings> Distributed Environment > Search Head Clustering'.



4. Verify the Captain has been dynamically chosen.



### Configure the Deployer

A Splunk Enterprise instance that distributes apps and certain other configuration updates to search head cluster members is referred to as a 'Deployer'. Any Splunk Enterprise instance can be configured to act as the Deployer. In this solution the admin3 is selected to serve this function as well.



Do not locate deployer functionality on a search head cluster member. The deployer must be a separate instance from any cluster member, as it is used to manage the configurations for the cluster members.

To configure the Deployer, follow these steps:

1. Open an SSH session to admin3 as the user 'splunk'.
2. Navigate to \$SPLUNK\_HOME/etc/system/local/.
3. Edit server.conf to include the following stanza:

```
[shclustering]
pass4SymmKey = your_secret_key
shcluster_label = hxshcluster
```

4. Restart the Splunk instance on admin3.

```
[splunk@admin3 ~]$ cd $SPLUNK_HOME/etc/system/local/
[splunk@admin3 local]$ vi server.conf
[splunk@admin3 local]$
[splunk@admin3 local]$ cat server.conf | tail -4

[shclustering]
pass4SymmKey = $7$3jwu1NqkCQ5/0JcQT/jCnbdBBaAbGAiNwW2wKosUuqeF3Lndu3wxow==
shcluster_label = hxshcluster
[splunk@admin3 local]$
[splunk@admin3 local]$ sudo /usr/bin/systemctl restart Splunkd.service
[splunk@admin3 local]$
```

### Configure Search Heads to Forward Data to the Indexer Layer

It is a best practice to forward all search head internal data to the search peer (indexer) layer. This has several advantages: 1) It enables diagnostics for the search head if it goes down. The data leading up to the failure is accumulated on the indexers, where another search head can later access it. 2) By forwarding the results of summary index searches to the indexer level, all search heads have access to them. Otherwise, they're only available to the search head that generates them.

The recommended approach is to forward the data directly to the indexers, without indexing separately on the search head. You do this by configuring the search head as a forwarder by creating an outputs.conf file on the search head that configures the search head for load-balanced forwarding across the set of search peers (indexers). The indexing on the search head is disabled so that the search head does not both retain the data locally as well as forward it to the search peers.

To configure the search heads to forward data to the indexer layer, follow these steps:

1. SSH to admin3 node as the splunk user.
2. Navigate to \$SPLUNK\_HOME/etc/shcluster/apps.
3. Create the directory 'outputs' and 'outputs/local'.
4. Navigate to the newly created 'local' directory, create the file outputs.conf with the following content:

```
# Turn off indexing on the master
[indexAndForward]
index = false
[tcput]
defaultGroup = search_peers
forwardedindex.filter.disable = true
indexAndForward = false
[tcput:search_peers]
server=idx1:9997,idx2:9997,idx3:9997,idx4:9997,idx5:9997,idx6:9997,idx7:9997,idx8:9997
autoLB = true
```

```
[splunk@admin3 ~]$ cd $SPLUNK_HOME/etc/shcluster/apps/
[splunk@admin3 apps]$ mkdir -p outputs
[splunk@admin3 apps]$ mkdir -p outputs/local
[splunk@admin3 apps]$ cd outputs/local/
[splunk@admin3 local]$ vi outputs.conf
[splunk@admin3 local]$ cat outputs.conf
# Turn off indexing on the master
[indexAndForward]
index = false

[tcput]
defaultGroup = search_peers
forwardedindex.filter.disable = true
indexAndForward = false

[tcput:search_peers]
server=idx1:9997,idx2:9997,idx3:9997,idx4:9997,idx5:9997,idx6:9997,idx7:9997,idx8:9997,idx9:9997,idx10:9997
autoLB = true
[splunk@admin3 local]$
```

- Execute the 'apply shcluster-bundle' command:

```
$SPLUNK_HOME/bin/splunk apply shcluster-bundle -target https://sh1:8089 -auth
admin:Cisco123
```

- Acknowledge the warning, a prompt will notify that the bundle has been pushed successfully.

```
[splunk@admin3 local]$ $SPLUNK_HOME/bin/splunk apply shcluster-bundle -target https://sh1:8089
-auth admin:Cisco123
Warning: Depending on the configuration changes being pushed, this command might initiate a ro
lling restart of the cluster members. Please refer to the documentation for the details. Do yo
u wish to continue? [y/n]: y
Bundle has been pushed successfully to all the cluster members.
[splunk@admin3 local]$
```

- Log into one of the search heads through SSH.
- Navigate to \$SPLUNK\_HOME/etc/apps/outputs/default.
- Verify that the file 'outputs.conf' has been pushed to the search heads.

```
[splunk@sh1 ~]$ cd $SPLUNK_HOME/etc/apps/outputs/default
[splunk@sh1 default]$ ls -lts
total 8
4 -rw----- 1 splunk splunk 77 Apr 17 12:29 app.conf
4 -rw-rw-r-- 1 splunk splunk 311 Apr 17 12:29 outputs.conf
[splunk@sh1 default]$ cat outputs.conf
# Turn off indexing on the master
[indexAndForward]
index = false

[tcput]
defaultGroup = search_peers
forwardedindex.filter.disable = true
indexAndForward = false

[tcput:search_peers]
server=idx1:9997,idx2:9997,idx3:9997,idx4:9997,idx5:9997,idx6:9997,idx7:9997,idx8:9997,idx9:9997,idx10:9997
autoLB = true
[splunk@sh1 default]$
```

## Configure Search Head Load-Balancing

It is useful to utilize a load balancer to take advantage of the Search Head Cluster:

- Designate a common URL for use throughout the enterprise (e.g. <https://splunk.domain.com>);
- The common URL should balance traffic between all four search heads and their respective ports, e.g. <https://sh1:8000>, <https://sh2:8000>, <https://sh3:8000>, <https://sh4:8000>.



Explicit instructions for configuring the designated load balancer will differ by vendor, but the functionality and load balancing direction is the same. No load balancer is used for this CVD.

---

## Integrate the Search Head Cluster with the Indexer Cluster

To integrate a search head cluster with an indexer cluster, configure each member of the search head cluster as a search head on the indexer cluster. Once you do that, the search heads get their list of search peers from the master node of the indexer cluster.

To integrate the search head cluster with the indexer cluster, follow these steps:

1. SSH to any one of the Search Head virtual machines as the 'splunk' user.
2. Issue the command '\$SPLUNK\_HOME/bin/splunk edit cluster-config -mode searchhead -master\_uri https://admin1:8089 -auth <username>:<password>' to add the virtual machine to the indexer cluster as a search head.
3. Restart the Splunk instance on the search head.

```
[splunk@sh4 ~]$ $SPLUNK_HOME/bin/splunk edit cluster-config -mode searchhead -master_uri https://admin1:8089 -auth admin:Cisco123
The cluster-config property has been edited.
[splunk@sh4 ~]$
[splunk@sh4 ~]$ sudo /usr/bin/systemctl restart Splunkd.service
[splunk@sh4 ~]$
```

4. On the search head, verify that the server mode has been defined as 'searchhead'.

```
[splunk@sh4 local]$ cat $SPLUNK_HOME/etc/system/local/server.conf | grep -w clustering -A 4
[clustering]
master_uri = https://admin1:8089
mode = searchhead
pass4SymmKey = $/5uhXgVVjomQ2dDtSjX5C+RDBmLNBzw0zLzqY4QcM3GBIaG/4twknfg==
[splunk@sh4 local]$
```

5. Repeat steps 1-4 for all the search heads.

## Verify Search Head Clustering

To verify the search head clustering, follow these steps:

1. SSH to any one of the Search Head virtual machines as the 'splunk' user.
2. Issue the command '\$SPLUNK\_HOME/bin/splunk show shcluster-status -auth <username>:<password>' to check the status of the search head cluster.



```
[splunk@sh4 ~]$ $SPLUNK_HOME/bin/splunk show shcluster-status -auth admin:Cisco123

Captain:
    dynamic_captain : 1
    elected_captain : Fri Apr 12 08:37:39 2019
                   id : 8F55F140-7D3B-4ED3-91A8-5CC54EADA353
    initialized_flag : 1
                   label : sh3
                   mgmt_uri : https://sh3:8089
    min_peers_joined_flag : 1
    rolling_restart_flag : 0
    service_ready_flag : 1

Members:
sh1
    label : sh1
    last_conf_replication : Sat Apr 13 19:05:17 2019
    mgmt_uri : https://sh1:8089
    mgmt_uri_alias : https://192.168.11.58:8089
    status : Up

sh2
    label : sh2
    last_conf_replication : Sat Apr 13 19:05:17 2019
    mgmt_uri : https://sh2:8089
    mgmt_uri_alias : https://192.168.11.59:8089
    status : Up

sh4
    label : sh4
    last_conf_replication : Sat Apr 13 19:05:15 2019
    mgmt_uri : https://sh4:8089
    mgmt_uri_alias : https://192.168.11.61:8089
    status : Up

sh3
    label : sh3
    mgmt_uri : https://sh3:8089
    mgmt_uri_alias : https://192.168.11.60:8089
    status : Up

[splunk@sh4 ~]$
```

- Alternatively, you can run '\$SPLUNK\_HOME/bin/splunk list shcluster-members -auth <username>:<password>' to view the various members.

```

[splunk@sh4 ~]$ $SPLUNK_HOME/bin/splunk list shcluster-members -auth admin:Cisco123
07512DD2-9572-4F24-8234-2C6D08706B50
  adhoc_searchhead:0
  advertise_restart_required:0
  artifact_count:2
  delayed_artifacts_to_discard:
  fixup_set:
  host_port_pair:192.168.11.58:8089
  is_captain:0
  kv_store_host_port:sh1:8191
  label:sh1
  last_heartbeat:1555207613
  mgmt_uri:https://sh1:8089
  no_artifact_replications:0
  peer_scheme_host_port:https://192.168.11.58:8089
  pending_job_count:0
  preferred_captain:1
  replication_count:0
  replication_port:18081
  replication_use_ssl:0
  site:default
  status:Up

135CDC2D-72F0-4F76-B548-2BAF92123F8A
  adhoc_searchhead:0
  advertise_restart_required:0
  artifact_count:1
  delayed_artifacts_to_discard:
  fixup_set:
  host_port_pair:192.168.11.59:8089
  is_captain:0
  kv_store_host_port:sh2:8191
  label:sh2
  last_heartbeat:1555207610
  mgmt_uri:https://sh2:8089
  no_artifact_replications:0
  peer_scheme_host_port:https://192.168.11.59:8089
  pending_job_count:0
  preferred_captain:1
  replication_count:0
  replication_port:18081
  replication_use_ssl:0
  site:default
  status:Up

45290CD4-9E2D-48D6-BCC0-FD21D7C52CD1
  adhoc_searchhead:0
  advertise_restart_required:0
  artifact_count:1
  delayed_artifacts_to_discard:
  fixup_set:
  host_port_pair:192.168.11.61:8089
  is_captain:0
  kv_store_host_port:sh4:8191
  label:sh4
  last_heartbeat:1555207609
  mgmt_uri:https://sh4:8089
  no_artifact_replications:0
  peer_scheme_host_port:https://192.168.11.61:8089
  pending_job_count:0
  preferred_captain:1
  replication_count:0
  replication_port:18081
  replication_use_ssl:0
  site:default
  status:Up

D8F4EA34-3E89-49AC-8828-927BE58E1B4E
  adhoc_searchhead:0
  advertise_restart_required:0
  artifact_count:0
  delayed_artifacts_to_discard:
  fixup_set:
  host_port_pair:192.168.11.60:8089
  is_captain:1
  kv_store_host_port:sh3:8191
  label:sh3
  last_heartbeat:1555207614
  mgmt_uri:https://sh3:8089
  no_artifact_replications:0
  peer_scheme_host_port:https://192.168.11.60:8089
  pending_job_count:0
  preferred_captain:1
  replication_count:0
  replication_port:18081
  replication_use_ssl:0
  site:default
  status:Up
[splunk@sh4 ~]$ █

```

4. Navigate to the directory `$SPLUNK_HOME/etc/apps/outputs/default/` on any one of the search heads.
5. Verify that the `outputs.conf` file has been pushed by the Deployer to the search heads.

```
[splunk@sh4 ~]$ cd $SPLUNK_HOME/etc/apps/outputs/default/
[splunk@sh4 default]$ ls
app.conf  outputs.conf
[splunk@sh4 default]$ cat outputs.conf
# Turn off indexing on the master
[indexAndForward]
index = false

[tcput]
defaultGroup = search_peers
forwardedindex.filter.disable = true
indexAndForward = false

[tcput:search_peers]
server=idx1:9997,idx2:9997,idx3:9997,idx4:9997,idx5:9997,idx6:9997,idx7:9997,idx8:9997,idx9:9997,idx10:9997
autoLB = true
[splunk@sh4 default]$
```

## Configure Deployment Server

In this section, the server admin4 is configured to function as the Deployment server, and the procedure to push a sample “Splunk App” from the Deployment Server to a Universal Forwarder on a test server (admin6) is depicted.

Any Splunk instance can act as a Deployment Server that assists in maintaining and deploying apps. In particular, the Deployment Server acts as a central manager for Universal Forwarders deployed throughout the enterprise. Any configuration to be pushed to remote instances will be hosted under `$SPLUNK_HOME/etc/deployment-apps/`.



The Deployment Server is installed by default when Splunk Enterprise is deployed. In this CVD the admin4 instance will function as the designated Deployment Server.

## Install a Universal Forwarder on a Test Server (admin6)

To install a universal forwarder on a test server, follow these steps:

1. Download the Splunk Universal Forwarder software file `splunkforwarder-7.2.3-06d57c595b80-linux-2.6-x86_64.rpm` from <http://www.splunk.com/download/universalforwarder>. Copy the image to the test server admin6.
2. Issue the following command to install the package:

```
rpm -ivh /tmp/splunkforwarder-7.2.3-06d57c595b80-linux-2.6-x86_64.rpm
```

The Splunk Universal Forwarder RPM software will be installed in the default directory `/opt/splunkforwarder/`.

```
[root@admin6 /]# cd /opt/
[root@admin6 opt]# ls -lts
total 0
0 drwxr-xr-x  8 splunk splunk 231 Feb 28 14:31 splunkforwarder
0 drwxr-xr-x  2 root  root   6 Aug  4 2017 rh
```

Refer to the [Splunk Documentation](#) for information about how to install a Universal Forwarder on appropriate operating system of the Universal Forwarder host.

## Register the Universal Forwarder with the Deployment Server

To register the universal forwarder with the deployment server, follow these steps:

1. Through SSH, log into the system hosting the Universal Forwarder – admin6 as the user splunk.
2. Navigate to the `$SPLUNK_HOME/etc/system/local` directory.
3. Create and edit the file `'deploymentclient.conf'` with the following content:

```
[deployment-client]
clientName = MyForwarder
```

```
[target-broker:deploymentServer]
targetUri = admin4:8089

[splunk@admin6 ~]$ cd $SPLUNK_HOME/etc/system/local
[splunk@admin6 local]$ vi deploymentclient.conf
[splunk@admin6 local]$ cat deploymentclient.conf
[deployment-client]
clientName = MyForwarder

[target-broker:deploymentServer]
targetUri = admin4:8089
```

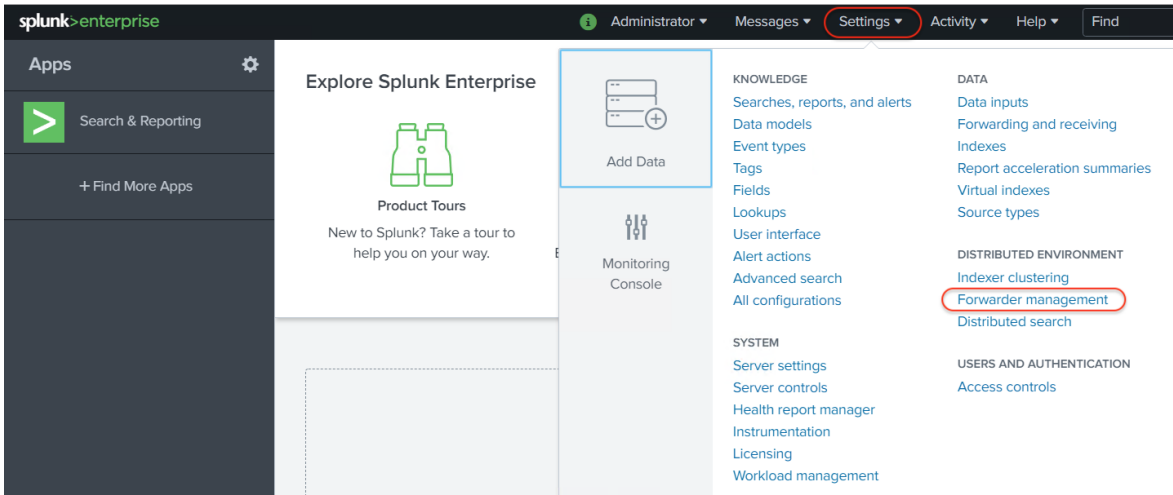
4. Restart Splunk service so as to start the Universal Forwarder.

```
[root@admin6 ~]# sudo /usr/bin/systemctl restart Splunkd.service
[root@admin6 ~]#
```

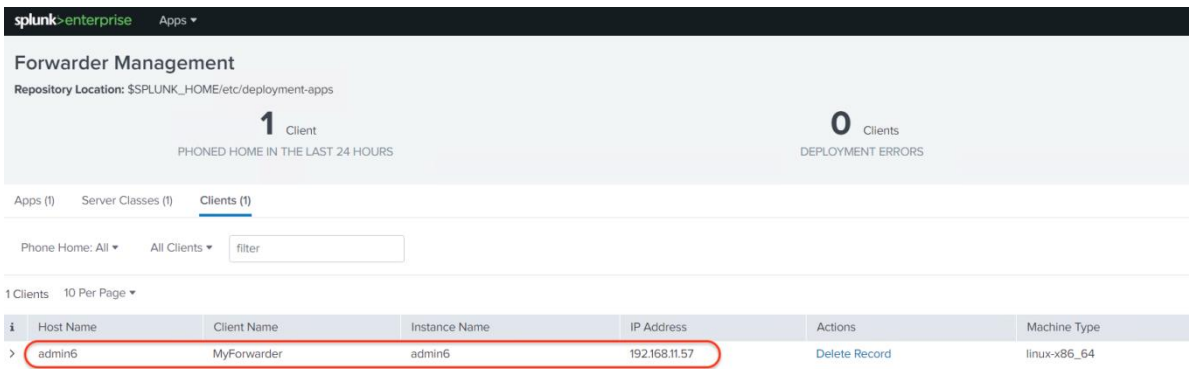
Configure an App within the Deployment Server (admin4)

To configure an app within the deployment server, follow these steps:

1. In a browser, navigate to the Splunk instance’s Web Interface of server admin4, for example, <https://admin4:8000>.
2. Select Settings > Distributed Environment > Forwarder management.



3. Notice the record of the Universal Forwarder communicating with the Deployment Server (this step may take up to five minutes due to polling cycle).



4. SSH to the Deployment Server - admin4 as the splunk user.
5. Navigate to \$SPLUNK\_HOME/etc/deployment-apps/.
6. Create the directory appTest.
7. Within appTest create the directory local.
8. Create the file app.conf and include the following contents:

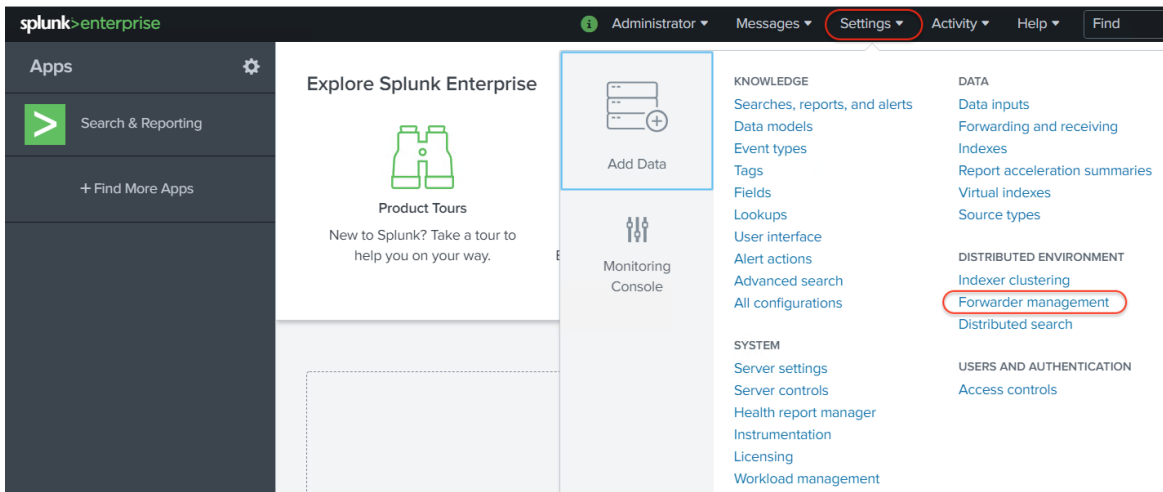
```
[tcpout]
defaultGroup = search_peers
[tcpout:search_peers]
autoLB = true
forceTimebasedAutoLB = true
server=idx1:9997,idx2:9997,idx3:9997,idx4:9997,idx5:9997,idx6:9997,idx7:9997,idx8:9997
```

9. Restart the deployment server by running the command:

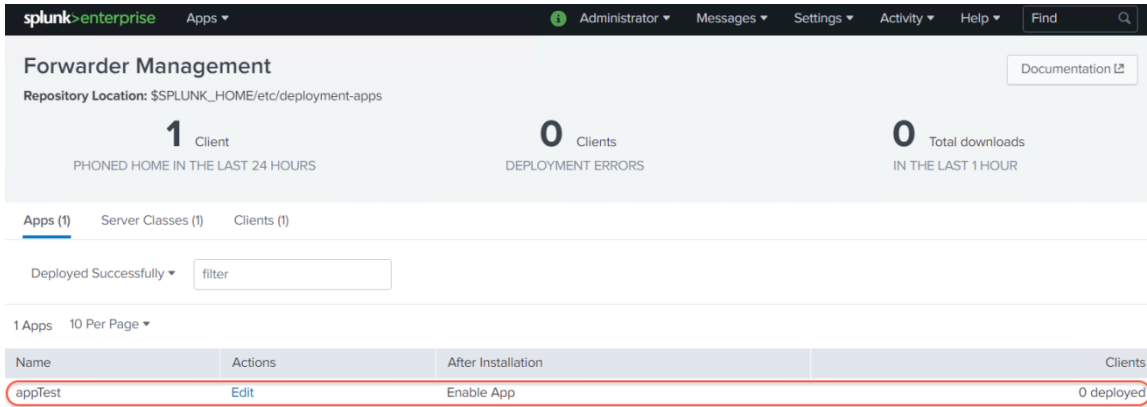
```
$SPLUNK_HOME/bin/splunk reload deploy-server -auth admin:Cisco123
```

```
[splunk@admin4 ~]$ cd $SPLUNK_HOME/etc/deployment-apps/
[splunk@admin4 deployment-apps]$ mkdir appTest
[splunk@admin4 deployment-apps]$ cd appTest/
[splunk@admin4 appTest]$ mkdir local
[splunk@admin4 appTest]$ cd local/
[splunk@admin4 local]$ vi app.conf
[splunk@admin4 local]$ cat app.conf
[tcpout]
defaultGroup = search_peers
[tcpout:search_peers]
autoLB = true
forceTimebasedAutoLB = true
server=idx1:9997,idx2:9997,idx3:9997,idx4:9997,idx5:9997,idx6:9997,idx7:9997,idx8:9997,idx9:9997,idx10:9997
[splunk@admin4 local]$ $SPLUNK_HOME/bin/splunk reload deploy-server -auth admin:Cisco123
Reloading serverclass(es).
[splunk@admin4 local]$
```

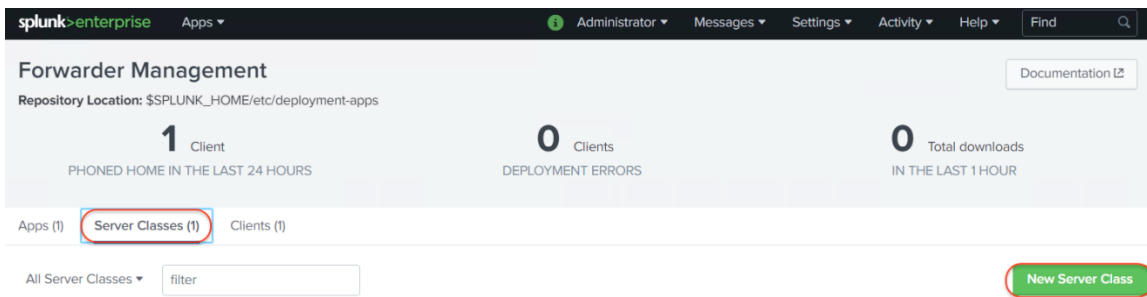
10. Go back to the Web GUI on admin4 (<http://admin4:8000>).
11. Navigate to Settings > Distributed Environment > Forwarder management.



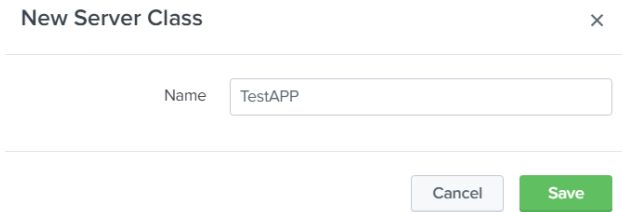
12. Click Apps. Zero apps have been deployed.



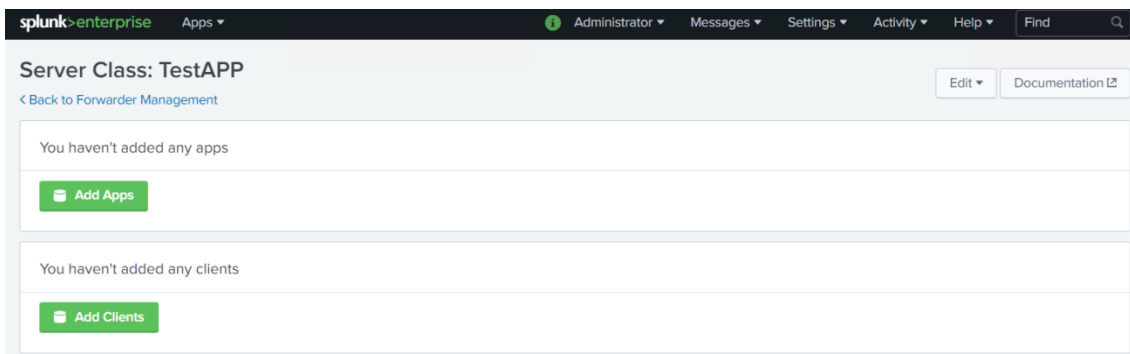
13. Click Server Class and select New Server Class.



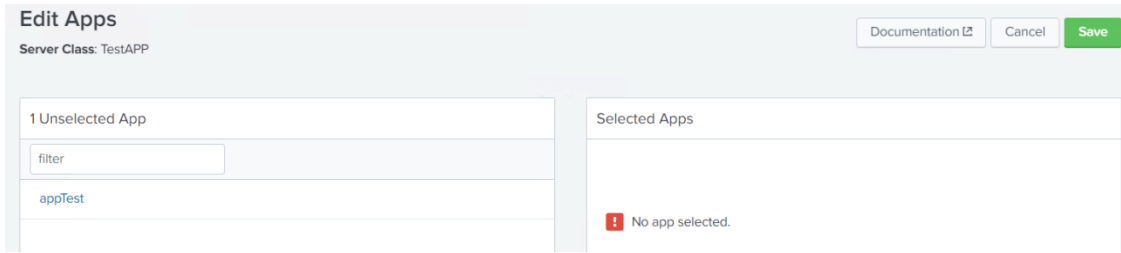
14. Give the name of 'TestAPP' to the new server class. Click Save.



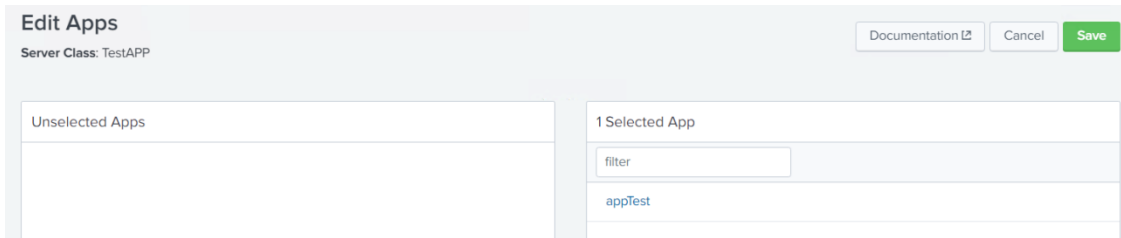
The following screenshot will present options for adding apps and clients.



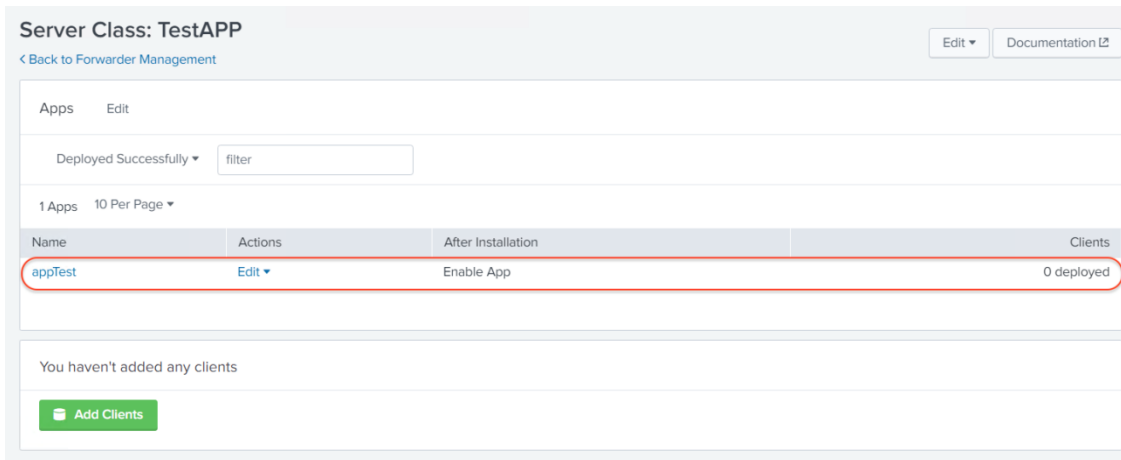
15. Click Add Apps. Check Unselected Apps.



16. Click appTest in the Unselected Apps to move it to Selected Apps.



17. Select Save and go back to the Sever Class page. The app appTest should show here as a selected application.



18. Click Add Clients.

19. Within the 'Edit Clients' screen, add the hostname of the forwarder to the whitelist. In this instance, the forwarder used is named 'admin6'.

**Edit Clients**  
Server Class: TestAPP

**Include (whitelist)**  
admin6

**Exclude (blacklist)**  
Optional

**Filter by Machine Type (machineTypesFilter)**  
Optional

Can be client name, host name, IP address, or DNS name. Examples: 185.2.3.\*, fwdr-\*

Can be client name, host name, IP address, or DNS name. Examples: ronnie, rarity

Cancel Preview **Save**

All Matched Unmatched filter

1 10 Per Page

Matched	Host Name	DNS Name	Client Name	Instance Name	IP Address	Machine Type	Phone Home
	admin6	admin6	MyForwarder	admin6	192.168.11.57	linux-x86_64	a few seconds ago

20. Click Save and return to the Server Class screen. Notice that the appTest is now deployed on one client.

**Server Class: TestAPP**  
Back to Forwarder Management Edit Documentation

1 App IN THE SERVER CLASS 1 Client IN THE SERVER CLASS 100% Clients DEPLOYED APPS SUCCESSFULLY

Apps Edit

Deployed Successfully filter

1 Apps 10 Per Page

Name	Actions	After Installation	Clients
appTest	Edit	Enable App	1 deployed

Clients Edit

Phone Home: All All Clients filter

1 Clients 10 Per Page

i	Host Name	Client Name	Instance Name	IP Address	Actions	Machine Type	Deployed Apps	Phone Home
>	admin6	MyForwarder	admin6	192.168.11.57	Delete Record	linux-x86_64	1 deployed	a few seconds ago

21. Click Back to Forwarder Management.

**Server Class: TestAPP**  
[Back to Forwarder Management](#)

22. View Server Classes in the Forwarder Manager, notice that the class TestAPP has been deployed.



**Forwarder Management** Documentation [🔗](#)

Repository Location: \$SPLUNK\_HOME/etc/deployment-apps

**1** Client

PHONED HOME IN THE LAST 24 HOURS

**0** Clients

DEPLOYMENT ERRORS

**1** Total download

IN THE LAST 1 HOUR

Apps (1) **Server Classes (2)** Clients (1)

All Server Classes ▾  New Server Class

2 Server Classes 10 Per Page ▾

Last Reload	Name	Actions	Apps	Clients
4 minutes ago	TestAPP	Edit ▾	1	1 deployed

23. View Apps in the Forwarder Manager, notice that the appTest has been deployed.

**Forwarder Management** Documentation [🔗](#)

Repository Location: \$SPLUNK\_HOME/etc/deployment-apps

**1** Client

PHONED HOME IN THE LAST 24 HOURS

**0** Clients

DEPLOYMENT ERRORS

**1** Total download

IN THE LAST 1 HOUR

Apps (1) **Server Classes (2)** Clients (1)

Deployed Successfully ▾

1 Apps 10 Per Page ▾

Name	Actions	After Installation	Clients
appTest	Edit ▾	Enable App	1 deployed

24. Log into the forwarder server (admin6) as user splunk.

25. Navigate to \$SPLUNK\_HOME/etc/apps. List the directory to view the newly deployed app.

```
[splunk@admin6 ~]$ cd $SPLUNK_HOME/etc/apps
[splunk@admin6 apps]$ ls -lts
total 0
0 drwxr----- 4 splunk splunk 35 Apr 15 11:38 appTest
0 drwxr-xr-x 5 splunk splunk 50 Apr 15 08:48 SplunkForwarder
0 drwxr-xr-x 6 splunk splunk 64 Feb 25 16:17 splunk_archiver
0 drwxr-xr-x 11 splunk splunk 135 Feb 18 16:21 splunk_monitoring_console
0 drwxr-xr-x 8 splunk splunk 92 Feb 18 16:20 splunk_instrumentation
0 drwxr-xr-x 6 splunk splunk 160 Feb 18 16:20 framework
0 drwxr-xr-x 5 splunk splunk 50 Feb 18 16:20 learned
0 drwxr-xr-x 4 splunk splunk 37 Feb 18 14:34 user-prefs
0 drwxr-xr-x 3 splunk splunk 21 Feb 18 14:34 splunk_httpinput
0 drwxr-xr-x 7 splunk splunk 130 Feb 18 14:34 splunk_gdi
0 drwxr-xr-x 9 splunk splunk 109 Feb 18 14:34 search
0 drwxr-xr-x 6 splunk splunk 66 Feb 18 14:34 sample_app
0 drwxr-xr-x 6 splunk splunk 68 Feb 18 14:34 launcher
0 drwxr-xr-x 3 splunk splunk 21 Feb 18 14:34 legacy
0 drwxr-xr-x 6 splunk splunk 68 Feb 18 14:34 gettingstarted
0 drwxr-xr-x 4 splunk splunk 32 Feb 18 14:34 introspection_generator_addon
0 drwxr-xr-x 7 splunk splunk 79 Feb 18 14:34 alert_webhook
0 drwxr-xr-x 4 splunk splunk 37 Feb 18 14:34 appsbrowser
0 drwxr-xr-x 7 splunk splunk 79 Feb 18 14:34 alert_logevent
0 drwxr-xr-x 4 splunk splunk 37 Feb 18 14:34 SplunkLightForwarder
[splunk@admin6 apps]$
```

Now the procedures to install and validate the deployment server is completed.

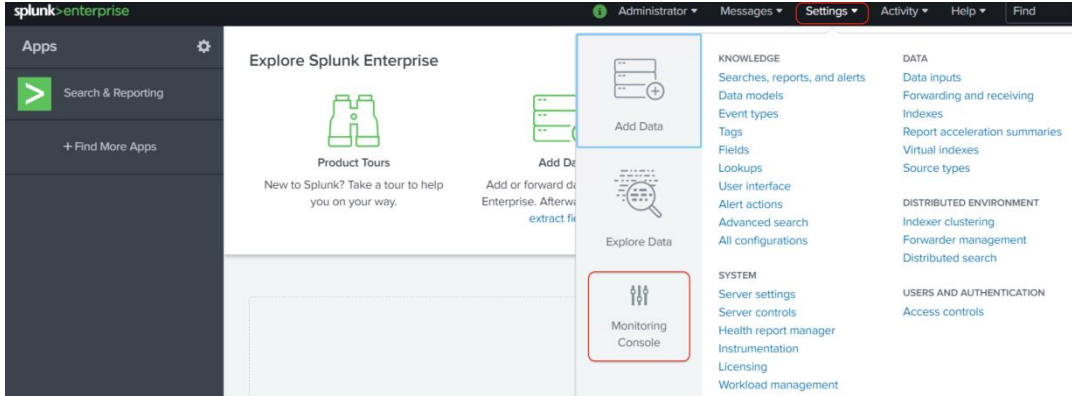
### Configure Distributed Monitoring Console

The Distributed Monitoring Console (DMC) is a special purpose pre-packaged app that comes with Splunk Enterprise providing detailed performance information about the Splunk Enterprise deployment.

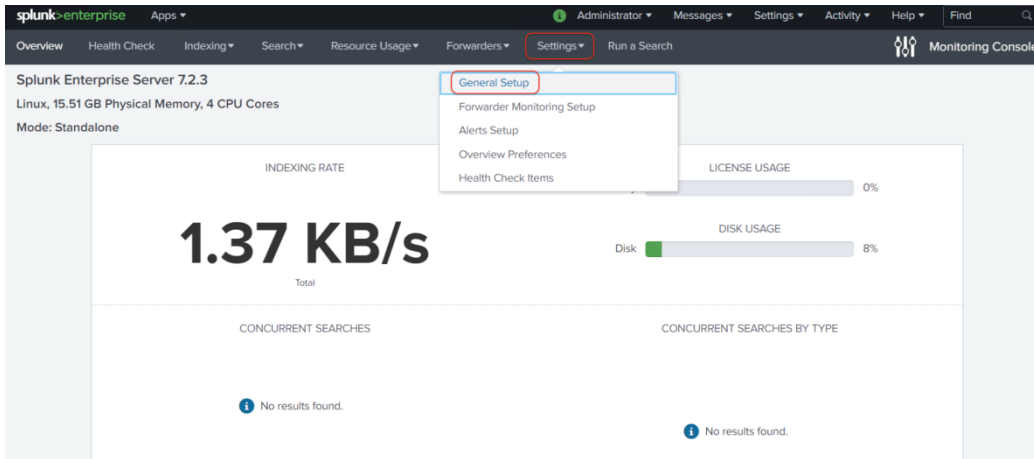
This section describes the procedure to configure the Distributed Monitoring Console for this deployment. In this solution DMC is installed on the master node i.e. admin1. Please refer to the [Splunk Documentation](#) for learning more about the installation options.

To configure the DMC, follow these steps:

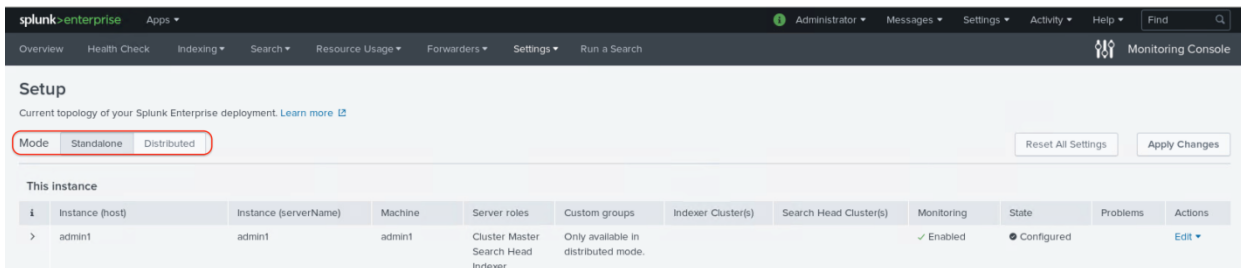
1. Navigate to the Splunk Web Interface on admin1 (such as <https://admin1:8000/>).
2. Click settings > Monitoring Console.



3. In the Monitoring Console app, click settings > General Setup.



4. At the General Setup page, click the Distributed button to change the monitoring console on the master node from the Standalone mode to the Distribute mode.



5. Acknowledge the warning message by clicking Continue.

### Switch to Distributed Mode

**⚠** Do not configure the DMC in distributed mode if this is a production search head. Doing so can change the behavior of all searches on this instance. This is dangerous and unsupported.

If you want to configure the DMC in distributed mode, you must locate the DMC on an instance that is not a production search head.  
[Learn more](#)

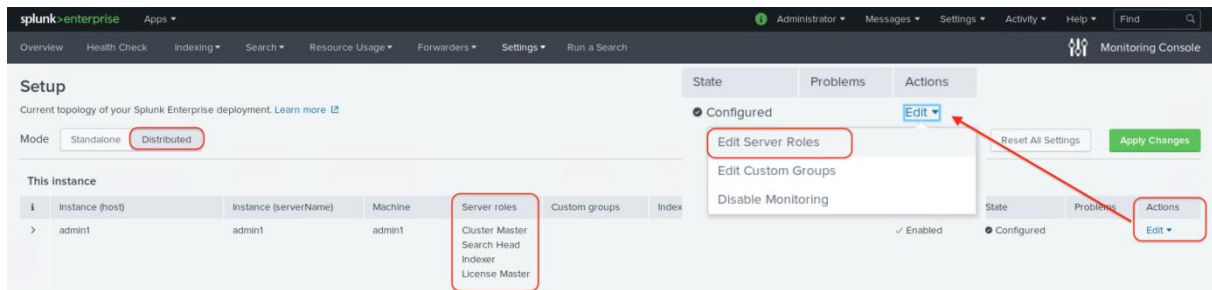
Are you sure you want to continue?



6. Enabling the Distributed mode should show all the 10 indexers as remote instances.

Remote instances												
10 Instances <input type="text" value="filter"/>												
Edit Selected Instances 25 Per Page												
i	<input type="checkbox"/>	Instance (host)	Instance (serverName)	Machine	Server roles	Custom groups	Indexer Cluster(s)	Search Head Cluster(s)	Monitoring	State	Problems	Actions
>	<input type="checkbox"/>	idx1	idx1	idx1	Indexer				✓ Enabled	● New		Edit
>	<input type="checkbox"/>	idx10	idx10	idx10	Indexer				✓ Enabled	● New		Edit
>	<input type="checkbox"/>	idx2	idx2	idx2	Indexer				✓ Enabled	● New		Edit
>	<input type="checkbox"/>	idx3	idx3	idx3	Indexer				✓ Enabled	● New		Edit
>	<input type="checkbox"/>	idx4	idx4	idx4	Indexer				✓ Enabled	● New		Edit
>	<input type="checkbox"/>	idx5	idx5	idx5	Indexer				✓ Enabled	● New		Edit
>	<input type="checkbox"/>	idx6	idx6	idx6	Indexer				✓ Enabled	● New		Edit
>	<input type="checkbox"/>	idx7	idx7	idx7	Indexer				✓ Enabled	● New		Edit
>	<input type="checkbox"/>	idx8	idx8	idx8	Indexer				✓ Enabled	● New		Edit
>	<input type="checkbox"/>	idx9	idx9	idx9	Indexer				✓ Enabled	● New		Edit

7. Select Edit > Edit Server Role on the admin1 instance, the server must change roles to function properly.



8. Select only 'Cluster Master' role for admin1 node. Click Save.

**Edit Server Roles** [X]

Search Head

**Cluster Master**

License Master

Indexer

Deployment Server

KV Store

SHC Deployer

Cancel Save

9. Confirm that the changes have been saved successfully.

**Success!**

Your server roles have updated successfully.

Done

10. Click Done to return to General Setup page. Click Apply Changes.

splunk-enterprise Apps

Administrator Messages Settings Activity Help Find

Overview Health Check Indexing Search Resource Usage Forwarders Settings Run a Search Monitoring Console

**Setup**

Current topology of your Splunk Enterprise deployment. [Learn more](#)

Mode Standalone Distributed Reset All Settings **Apply Changes**

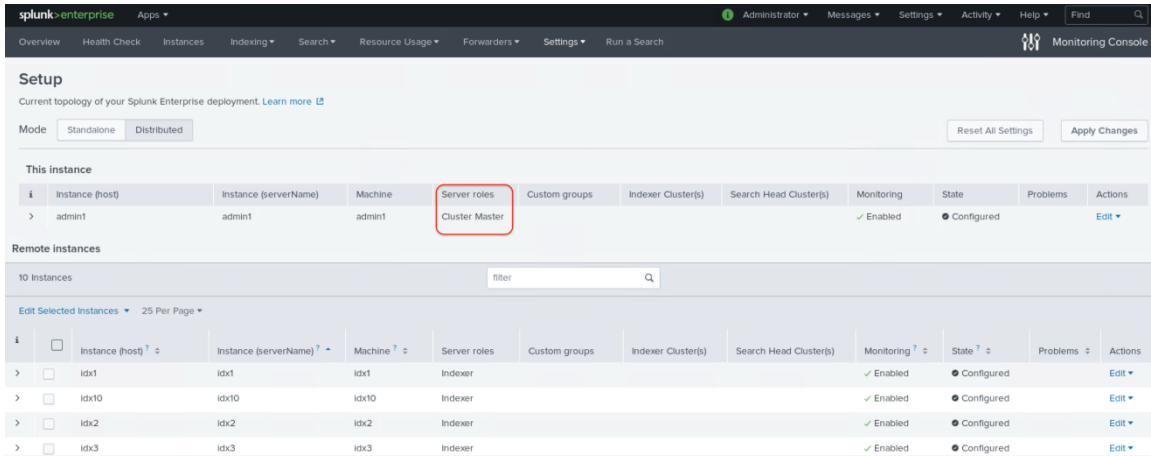
11. Confirm that changes have been applied successfully.

**Success!**

Your changes have been applied.  
**It may take a few minutes for your instances to be updated.**

Go to Overview Refresh

12. Click Refresh to return to the Setup page, verify that the role of admin1 has been changed. Ensure that the Master Node (admin1) does not have the role of 'search head.'

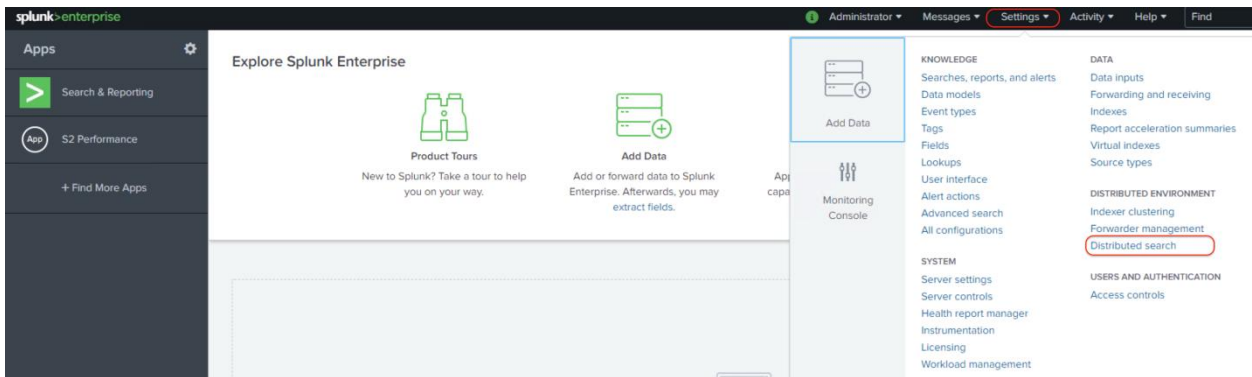


### Configure Search Heads in Distributed Monitoring Console

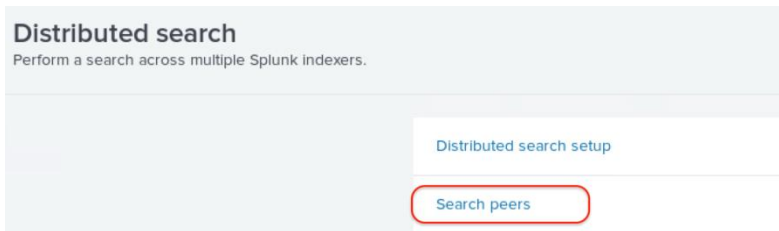
In the previous section, the Distributed Monitoring Console (DMC) was configured to manage the indexers and the master node. This section provides the procedure to configure DMC to monitor the search heads.

To configure the search heads in the DMC, follow these steps:

1. Navigate to the Master Node (admin1) through the GUI.
2. Open Settings > Distributed Environment > Distributed search.



3. Select 'Search Peers'.



4. All the ten indexers should display here as search peers. Select New Search Peer to add the search heads as search peers.

The screenshot shows the 'Search peers' page in Splunk Enterprise. It displays a table with 10 rows of search peers. Each row includes a Peer URI, Splunk instance name, State, Replication status, Cluster label, Health status, Health check failures, Status, and Actions.

Peer URI	Splunk instance name	State	Replication status	Cluster label	Health status	Health check failures	Status	Actions
192.168.11.62:8089	idx1	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.63:8089	idx2	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.64:8089	idx3	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.65:8089	idx4	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.66:8089	idx5	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.67:8089	idx6	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.68:8089	idx7	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.69:8089	idx8	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.70:8089	idx9	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.71:8089	idx10	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete

5. Add a new search peer with the following inputs:
  - a. Peer URI – Enter [the hostname or IP of one of your search heads]:[Management Port]
  - b. Remote username – use 'admin'
  - c. Remote password – the password to the Splunk admin account on the search head

The screenshot shows the 'Add new' form for adding a search peer. It includes a 'Peer URI' field with the value 'https://192.168.11.61:8089', a 'Distributed search authentication' section with 'Remote username' set to 'admin', and 'Remote password' and 'Confirm password' fields with masked characters. There are 'Cancel' and 'Save' buttons at the bottom right.

6. Click Save to add the search peer.
7. Repeat steps 1-6 for all the four search heads to add them as the distributed search peers on the master node.

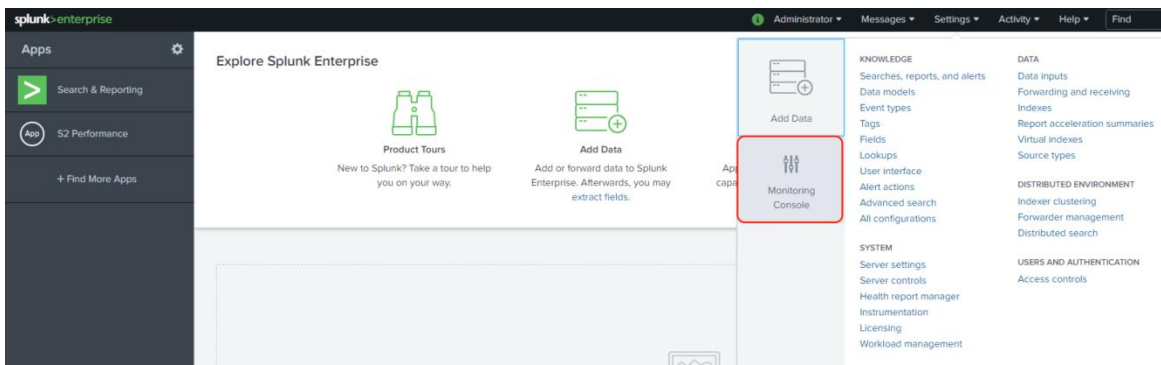
Search peers  
Distributed search > Search peers

Successfully saved "https://192.168.11.61:8089/"

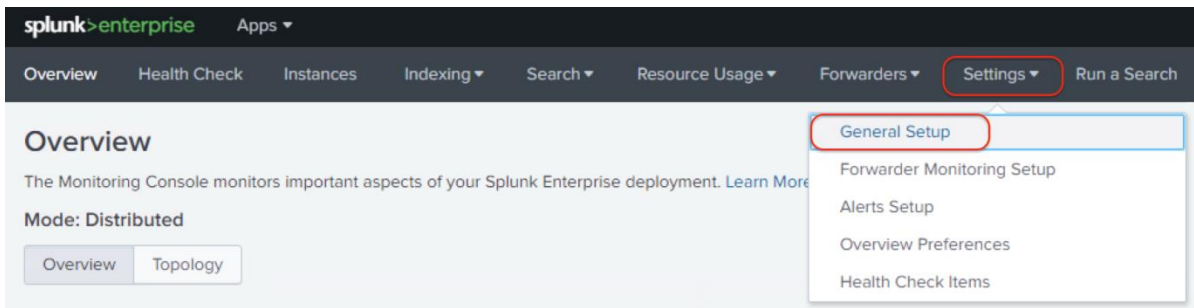
Showing 1-14 of 14 items

Peer URI	Splunk instance name	State	Replication status	Cluster label	Health status	Health check failures	Status	Actions
192.168.11.58:8089	sh1	Up	Initial	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.59:8089	sh2	Up	Initial	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.60:8089	sh3	Up	Initial	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.61:8089	sh4	Up	Initial	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.62:8089	idx1	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.63:8089	idx2	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.64:8089	idx3	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.65:8089	idx4	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.66:8089	idx5	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.67:8089	idx6	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.68:8089	idx7	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete
192.168.11.69:8089	idx8	Up	Successful	None	Healthy	None	Enabled   Disable	Quarantine   Delete

8. On the Master Node (admin1) navigate to Settings > Monitoring Console.



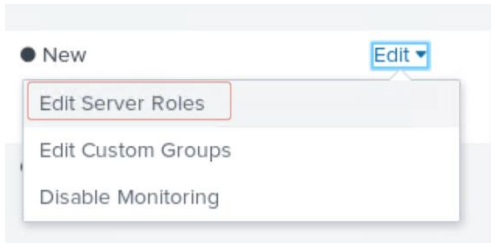
9. Select Settings > General Setup within the Distributed Monitoring Console.



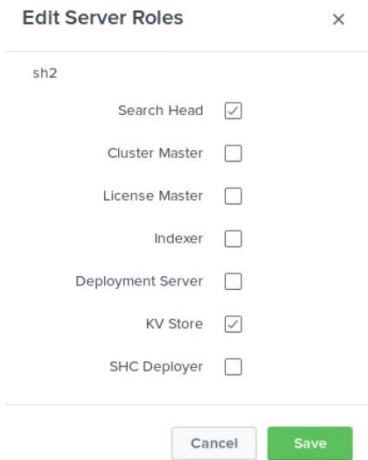
The four newly added search heads should be listed under 'remote instances' as 'new'.

>	<input type="checkbox"/>	sh1	sh1	sh1	Search Head KV Store	8F55F140-7D3B-4ED3-91A8-5CC54EADA...	✓ Enabled	New	Edit
>	<input type="checkbox"/>	sh2	sh2	sh2	Indexer Search Head KV Store	8F55F140-7D3B-4ED3-91A8-5CC54EADA...	✓ Enabled	New	Edit
>	<input type="checkbox"/>	sh3	sh3	sh3	Indexer Search Head KV Store	8F55F140-7D3B-4ED3-91A8-5CC54EADA...	✓ Enabled	New	Edit
>	<input type="checkbox"/>	sh4	sh4	sh4	Indexer Search Head KV Store	8F55F140-7D3B-4ED3-91A8-5CC54EADA...	✓ Enabled	New	Edit

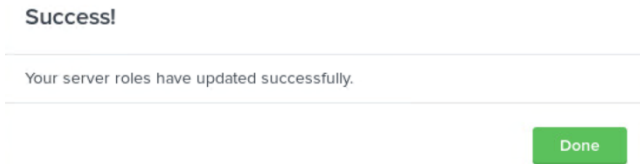
10. Select Edit > Edit Server Roles within the table for one search head instance.



11. Ensure that the server roles are Search Head and KV Store.



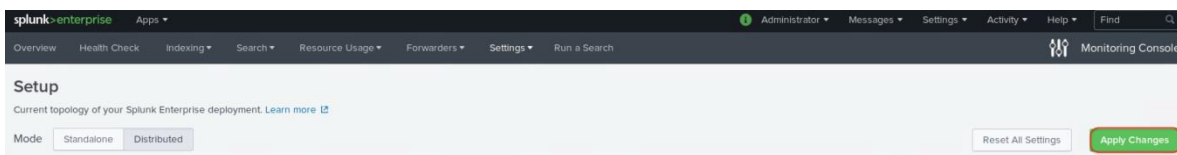
12. Click Save then click Done to complete the update.



13. Repeat Step 10-12 for each search head instance. Confirm the changes and roles.

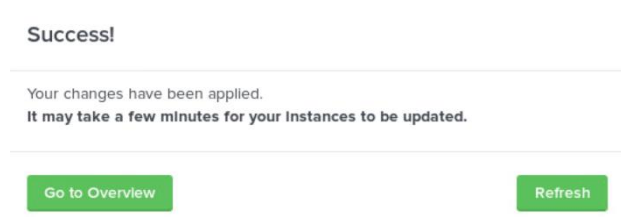


14. Click Apply Changes.

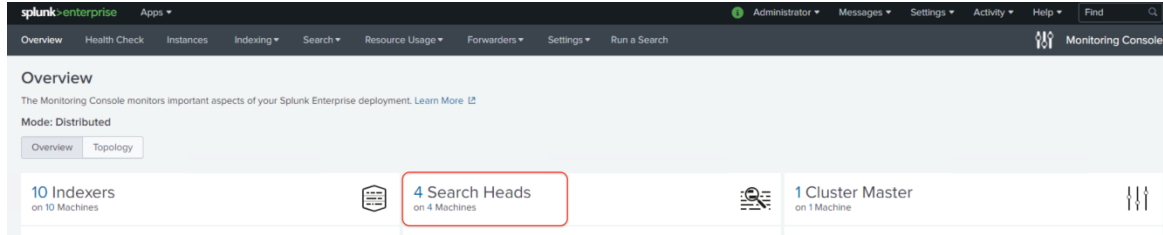




- Confirm that the changes have been applied successfully.



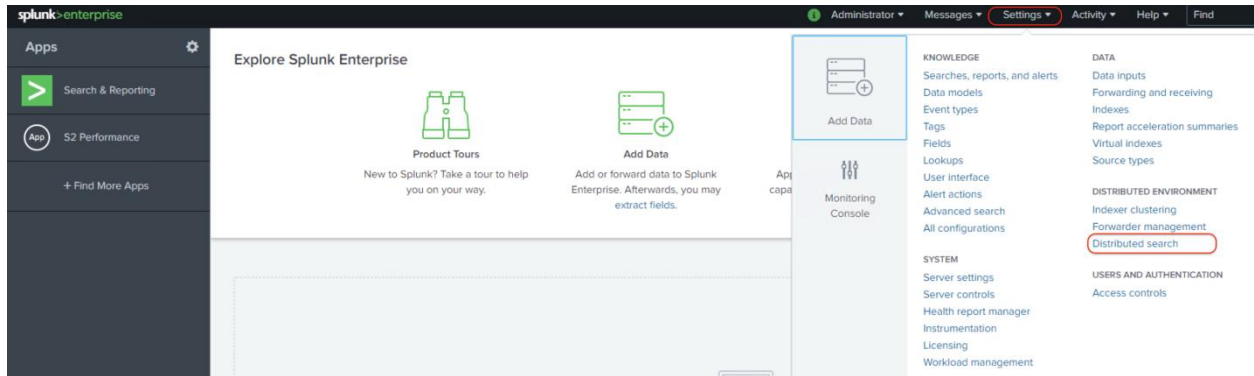
- Click Go to Overview, then go to the DMC Overview page. DMC should now display Search Heads within the overview.



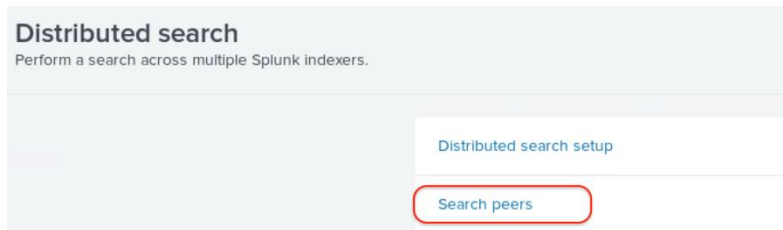
### Configure License Master in Distributed Monitoring Console

To configure DMC to monitor the license master, follow these steps:

- Navigate to the Master Node (admin1) through the GUI.
- Open 'Settings > Distributed Environment > Distributed search.



- Select Search Peers.



- Select New Search Peer to add the license master as search peers.



5. Add a new search peer with the following inputs:
  - a. Peer URI – Enter [the hostname or IP of your license master admin2]:[Management Port]
  - b. Remote username – use 'admin'
  - c. Remote password – the password to the Splunk admin account on the search head

**Add search peers**

Use this page to explicitly add distributed search peers. Enable distributed search through the Distributed search setup page in Splunk Settings.

Peer URI \*

Specify the search peer as `servername:mgmt_port` or `URI:mgmt_port`. You must prefix the URI with its scheme. For example: 'https://spl.example.com:8089'.

**Distributed search authentication**

To share a public key for distributed authentication, enter a username and password for an admin user on the remote search peer.

Remote username \*

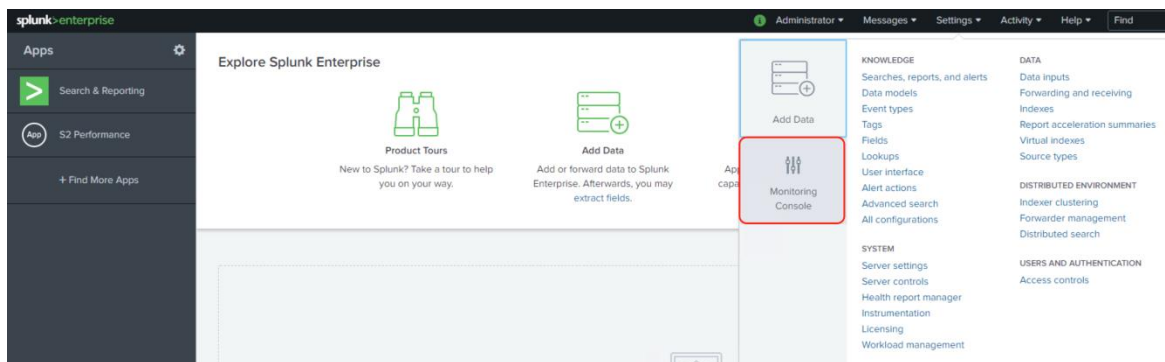
Remote password \*

Confirm password

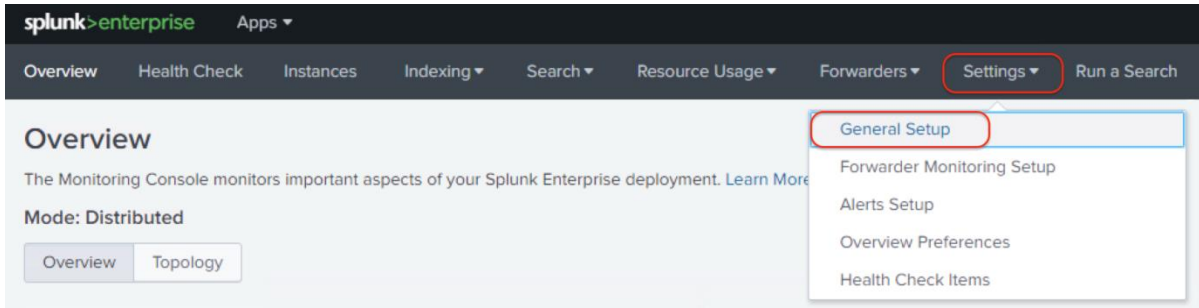
6. Click Save so that the license master node admin2 has been added to the distributed search peers on the master node.



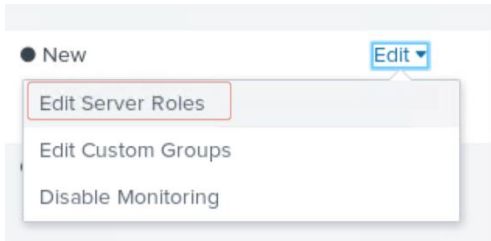
7. On the Master Node (admin1) navigate to Settings > Monitoring Console.



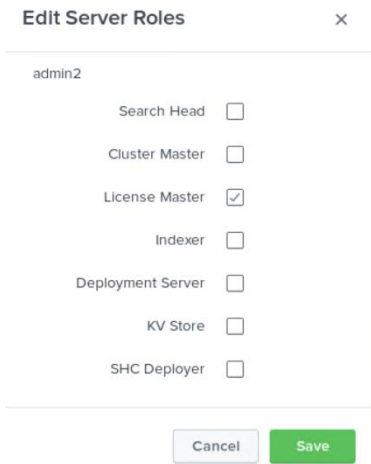
8. Select Settings > General Setup within the Distributed Monitoring Console.



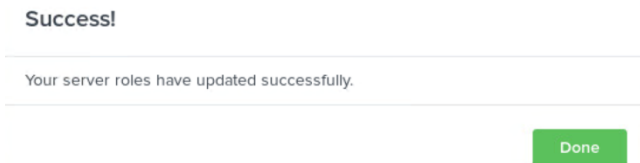
- 9. The newly added admin2 node should be listed under 'remote instances' as 'new'.
- 10. Select Edit > Edit Server Roles within the table for admin2 instance.



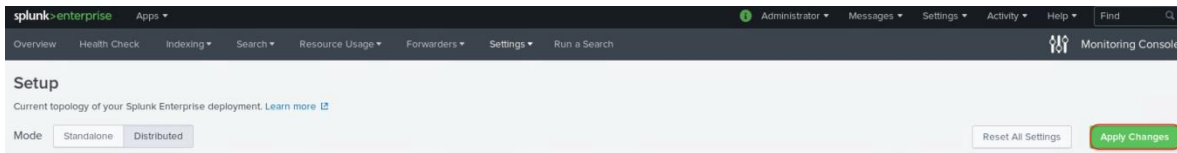
- 11. Edit the Server Roles to License Master.



- 12. Click Save then click Done to complete the update.



- 13. Click Apply Changes.



14. Confirm that the changes have been applied successfully.

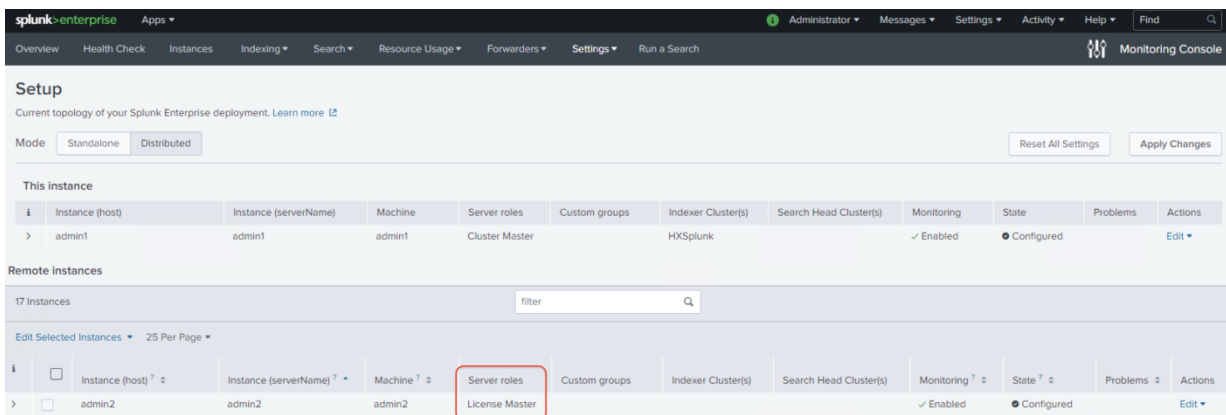
### Success!

Your changes have been applied.  
**It may take a few minutes for your Instances to be updated.**

[Go to Overview](#)

[Refresh](#)

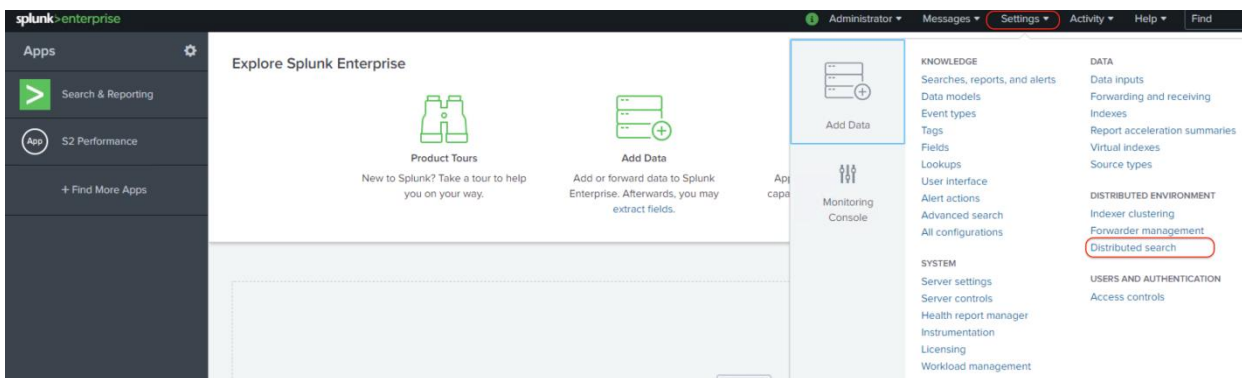
15. Click Refresh and ensure the server role of admin2 has been set to License Master.



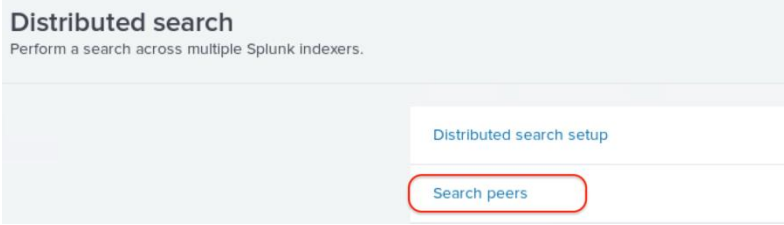
### Configure Search Head Cluster (SHC) Deployer in Distributed Monitoring Console

To configure DMC to monitor the search head cluster deployer, follow these steps:

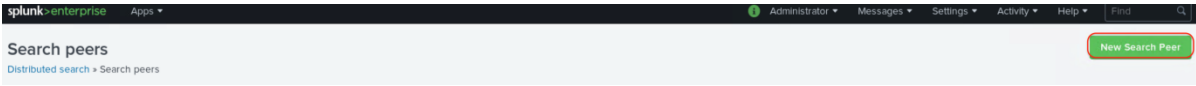
1. Navigate to the Master Node (admin1) through the GUI.
2. Open 'Settings → Distributed Environment > Distributed search.



3. Select Search Peers.

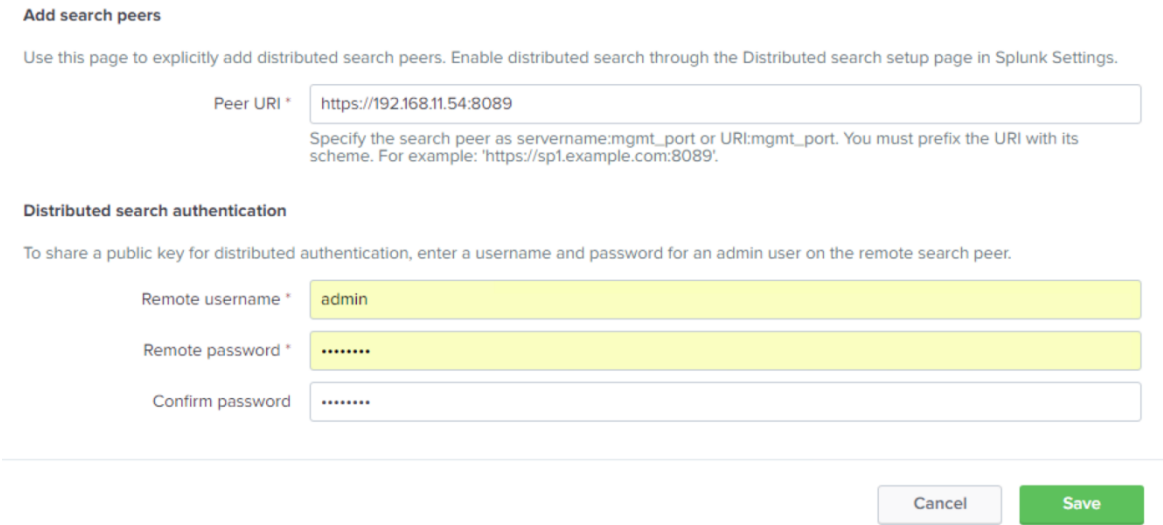


4. Select New Search Peer at the top right to add the SHC deployer as search peers.

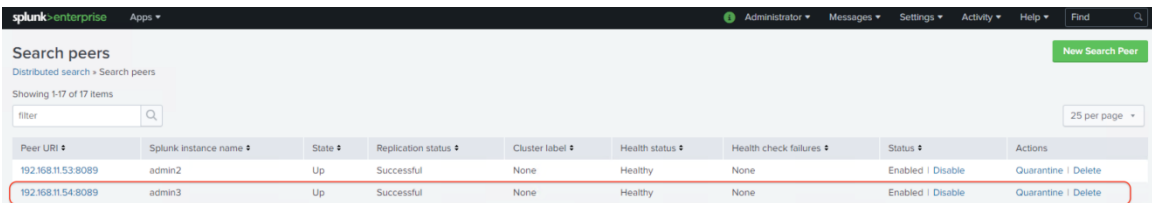


5. Add a new search peer with the following inputs:

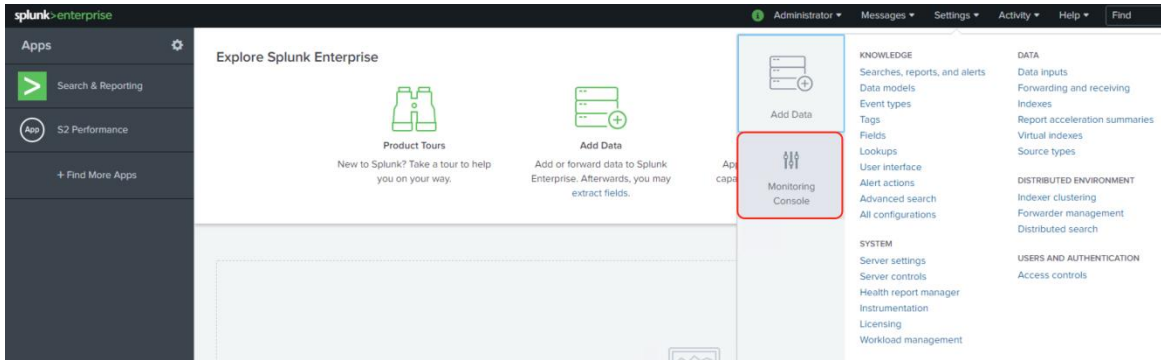
- Peer URI – Enter [the hostname or IP of your SHC deployer - admin3]:[Management Port]
- Remote username – use 'admin'
- Remote password – the password to the Splunk admin account on the search head



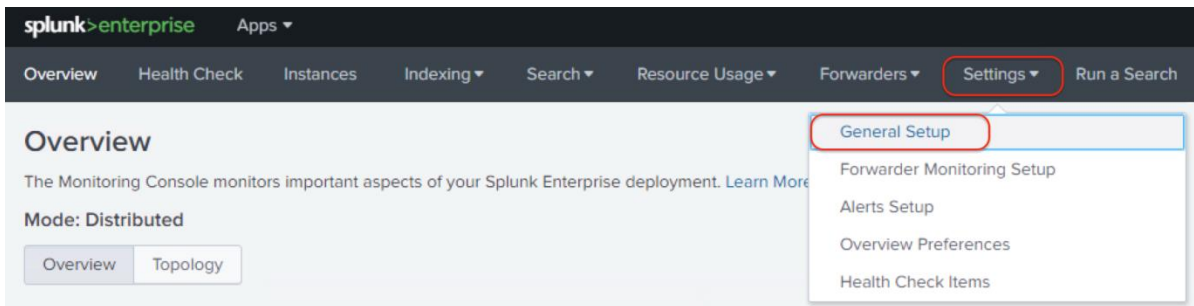
6. Click Save so that the SHC deployer - admin3 has been added to the distributed search peers on the master node.



7. On the Master Node (admin1) navigate to Settings > Monitoring Console.

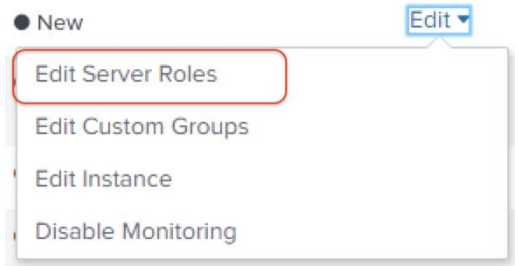


8. Select Settings > General Setup within the Distributed Monitoring Console.

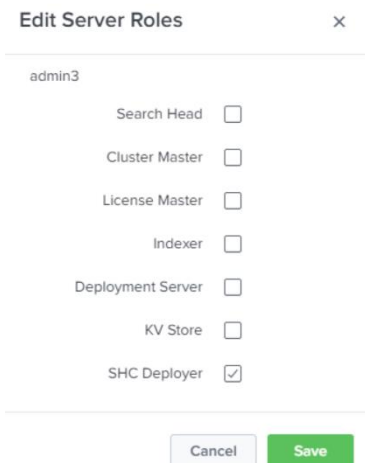


The newly added admin3 node should be listed under remote instances' as 'new.

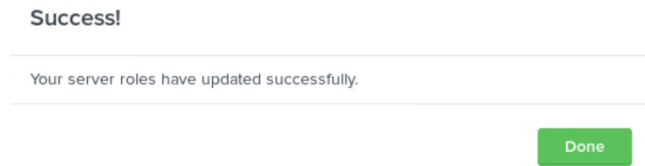
9. Select Edit > Edit Server Roles within the table for admin3 instance.



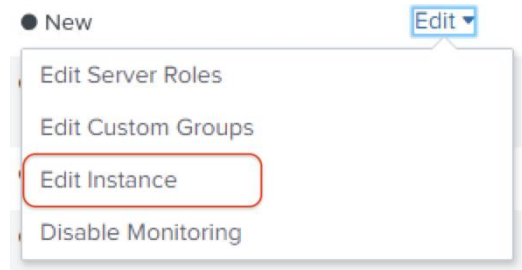
10. Change the server role to SHC Deployer.



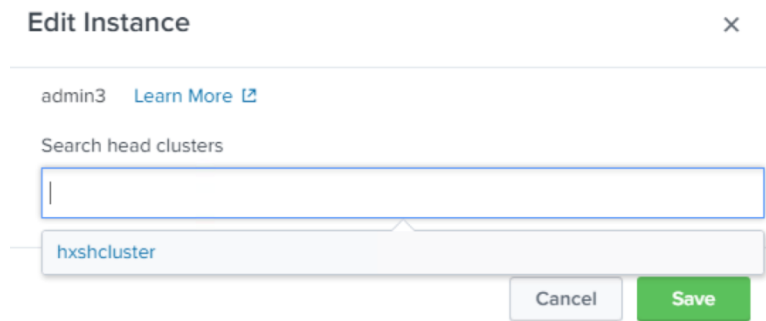
11. Click Save then click Done to complete the update.



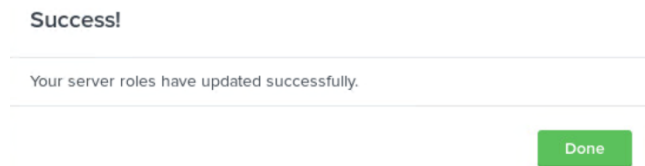
12. Select Edit > Edit Instance within the table for admin3 instance.



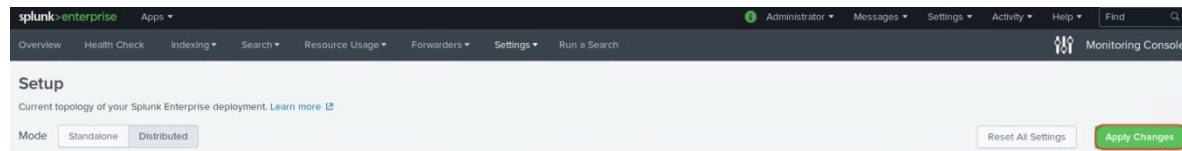
13. Configure the label of the search head clusters as hxshcluster.



14. Click Save then click Done to complete the update.



15. Click Apply Changes.



16. Confirm that the changes have been applied successfully.

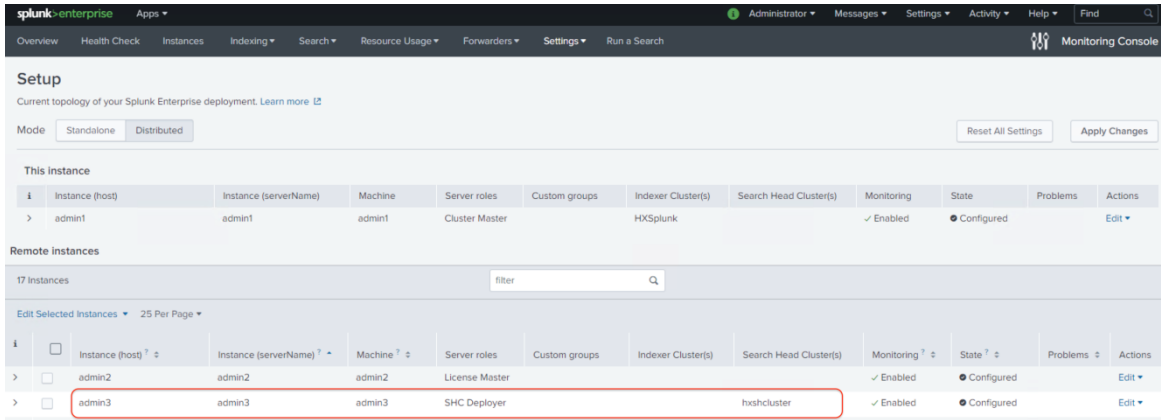
**Success!**

Your changes have been applied.  
**It may take a few minutes for your instances to be updated.**

[Go to Overview](#)

[Refresh](#)

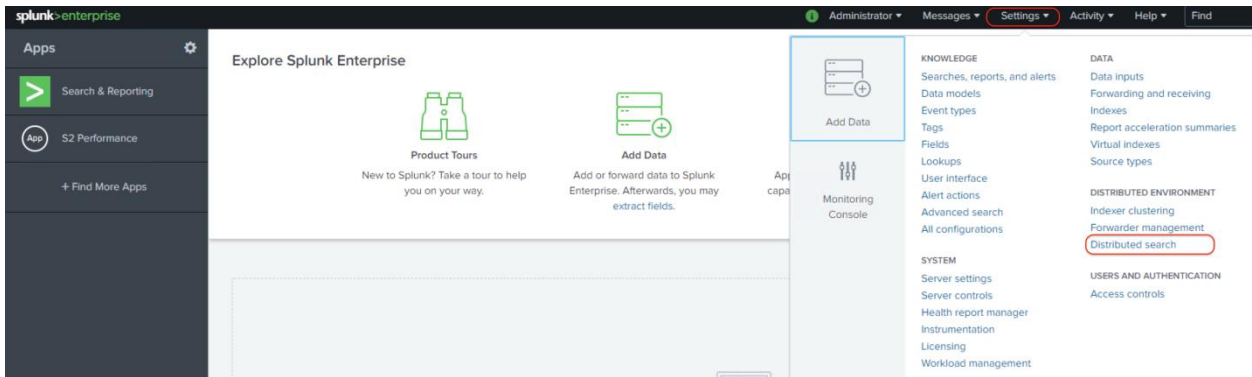
17. Click Refresh, ensure that the server role of admin3 has been set to SHC Deployer.



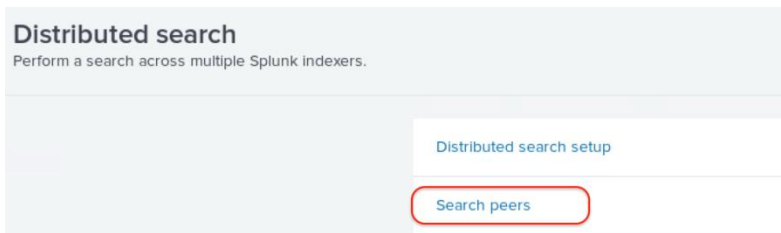
**Configure Deployment Server in Distributed Monitoring Console**

To configure DMC to monitor the deployment server, follow these steps:

1. Navigate to the Master Node (admin1) through the GUI.
2. Open Settings > Distributed Environment > Distributed search.

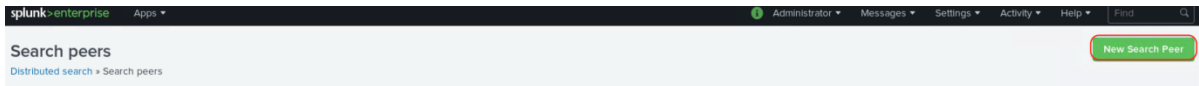


3. Select Search Peers.



4. Select New Search Peer to add the deployment server as search peers.





5. Add a new search peer with the following inputs:
  - a. Peer URI – Enter [the hostname or IP of your deployment server - admin<sub>4</sub>]:[Management Port]
  - b. Remote username – use 'admin'
  - c. Remote password – the password to the Splunk admin account on the search head

**Add search peers**

Use this page to explicitly add distributed search peers. Enable distributed search through the Distributed search setup page in Splunk Settings.

Peer URI \*

Specify the search peer as servername:mgmt\_port or URI:mgmt\_port. You must prefix the URI with its scheme. For example: 'https://sp1.example.com:8089'.

**Distributed search authentication**

To share a public key for distributed authentication, enter a username and password for an admin user on the remote search peer.

Remote username \*

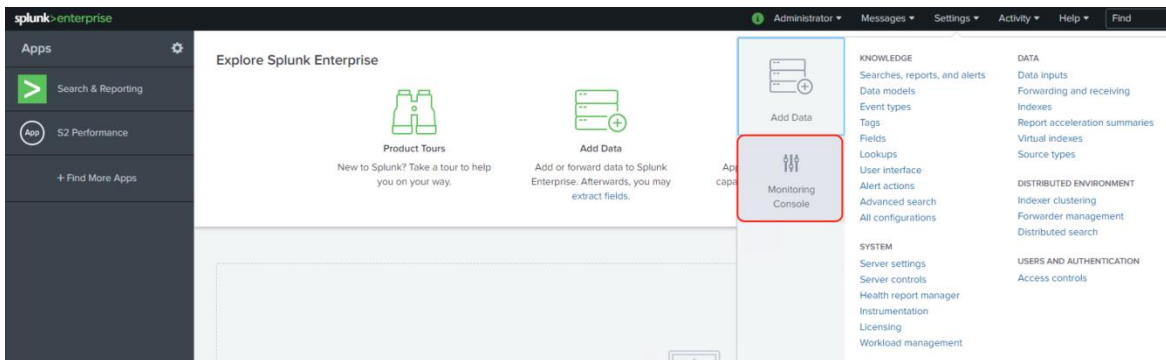
Remote password \*

Confirm password

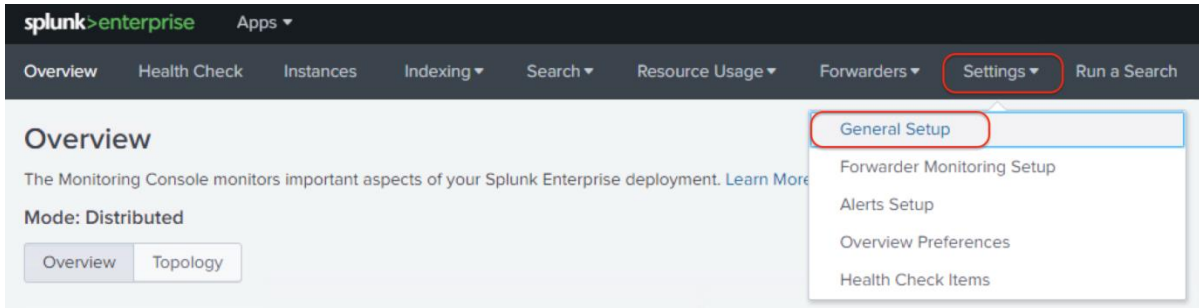
6. Click Save so that the deployment server – admin<sub>4</sub> has been added to the distributed search peers on the master node.



7. On the Master Node (admin<sub>1</sub>) navigate to Settings > Monitoring Console.

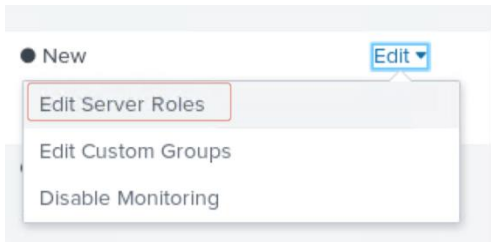


8. Select Settings > General Setup within the Distributed Monitoring Console.

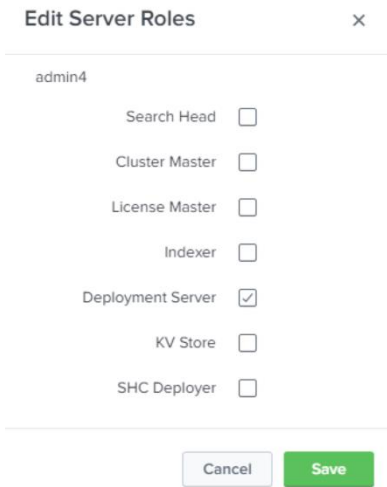


The newly added admin4 node should be listed under remote instances as new.

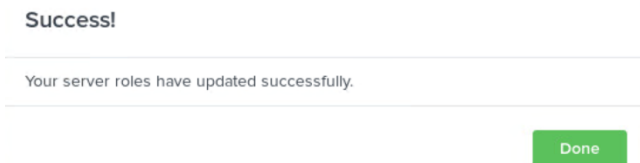
- 9. Select Edit > Edit Server Roles within the table for admin4 instance.



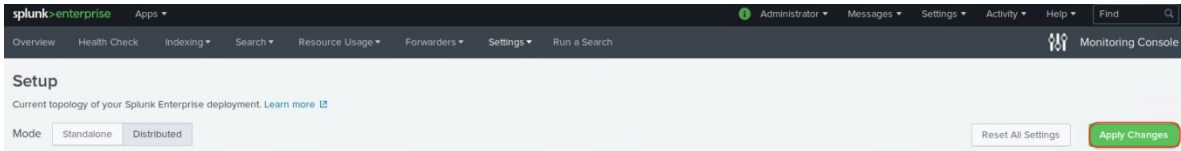
- 10. Change the server role to Deployment Server.



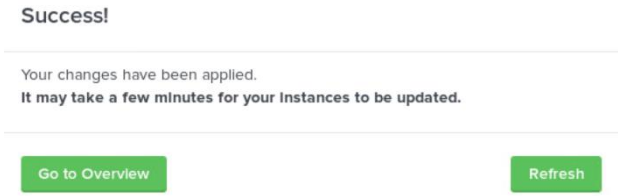
- 11. Click Save then click Done to complete the update.



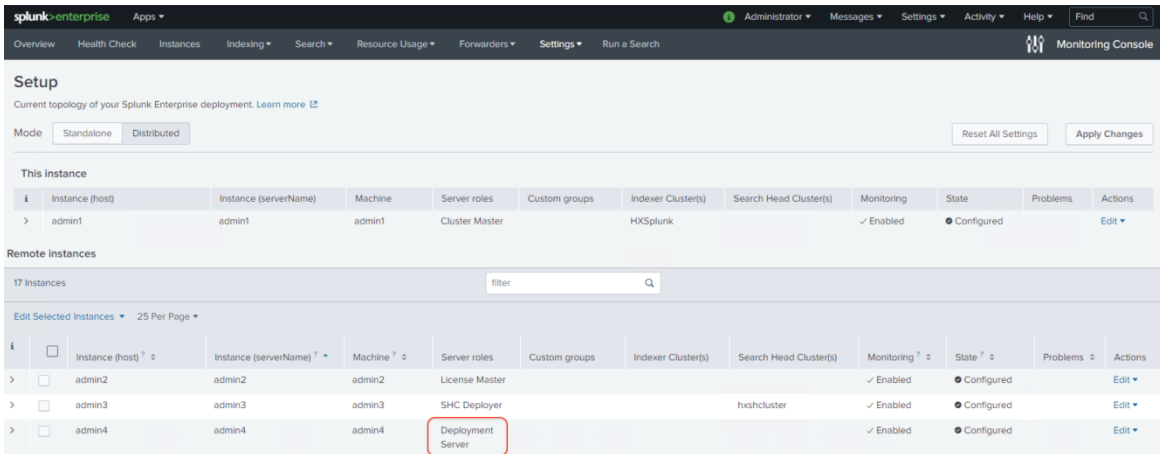
- 12. Click Apply Changes.



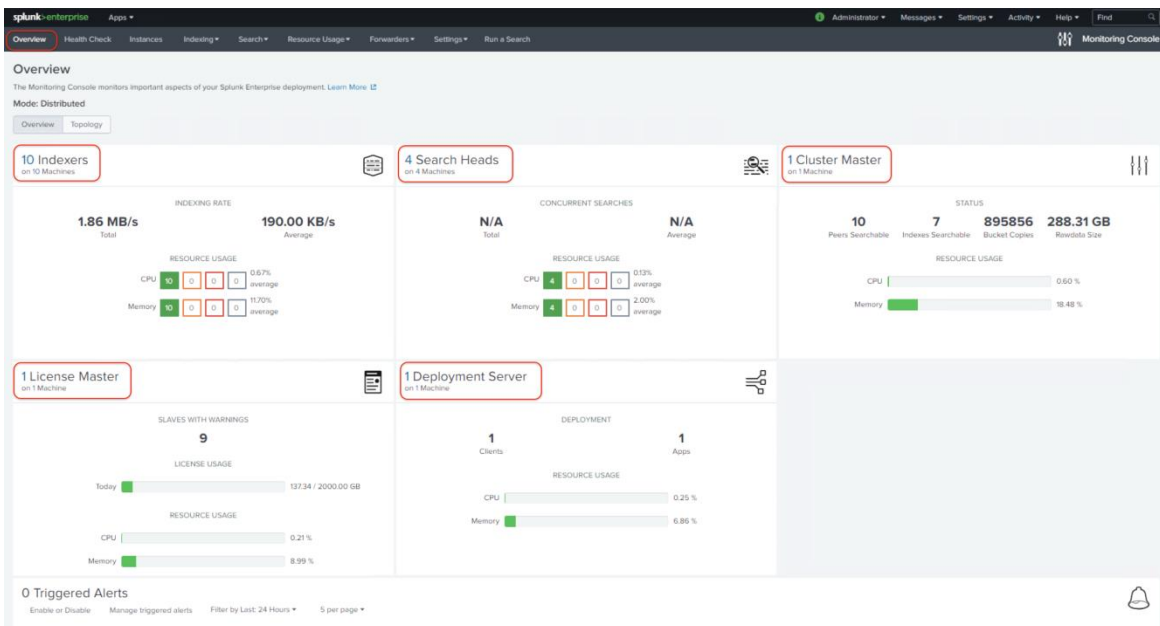
13. Confirm that changes have been applied successfully.



14. Click Refresh and ensure the server role of admin4 is set to Deployment Server.



15. Click Overview and verify that you can monitor the different type of Splunk components in DMC.



16. Click Topology in Overview, check the status of all the Splunk virtual machines. You can see each search head will distribute the searches across all the ten indexers.

The screenshot shows the Splunk Overview page in the 'Topology' view. The 'Indexers (10)' section lists 10 indexers (idx1 to idx10) with a status bar showing 10 green, 0 red, and 0 grey. The 'Search heads (4)' section lists 4 search heads (sh1 to sh4) with a status bar showing 4 green, 0 red, and 0 grey. The 'Other (4)' section lists 4 administrators (admin1 to admin4) with a status bar showing 4 green, 0 red, and 0 grey. A network diagram connects the indexers to the search heads, showing that each search head is connected to all 10 indexers.

## Configure SmartStore Indexes on the SwiftStack Object Storage

In the Splunk software, as the indexer indexes the incoming data, it creates a number of files. These files contain two types of data:

- The raw data in compressed form (rawdata files)
- Indexes that point to the raw data, plus some metadata (index files, also known as tsidx files)

Together, these files constitute the Splunk Enterprise index. The files reside in sets of directories organized by age. Some directories contain newly indexed data; others contain previously indexed data. The number of such directories can grow quite large, depending on how much data you're indexing.

In short, each of the index directories is known as a "bucket" which is a file system directory containing a portion of a Splunk Enterprise index. An "index" contains compressed raw data and associated index files. An index resides across many age-designated index directories.

A bucket moves through several stages as it ages: hot >warm >cold >frozen >thawed.

As buckets age, they "roll" from one stage to the next. As data is indexed, it goes into a hot bucket. Hot buckets are both searchable and actively being written to. An index can have several hot buckets open at a time. When certain conditions occur (for example, the hot bucket reaches a certain size or splunkd gets restarted), the hot bucket becomes a warm bucket ("rolls to warm"), and a new hot bucket is created in its place. Warm buckets are searchable, but are not actively written to. There are many warm buckets.

Once further conditions are met (for example, the index reaches some maximum number of warm buckets), the indexer begins to roll the warm buckets to cold, based on their age. It always selects the oldest warm bucket to roll to cold. Buckets continue to roll to cold as they age in this manner. After a set period of time, cold buckets roll to frozen, at which point they are either archived or deleted. By editing attributes in `indexes.conf`, the bucket aging policy can be specified, which determines when a bucket moves from one stage to the next.

If the frozen data has been archived, it can later be thawed. Thawed data is available for searches. If archival of specific sets of data is required, each additional index that is added will require the stanza: `coldToFrozenDir = <directory of frozen data>`. Each index that is added will require this stanza to be appended.

The collection of buckets in a particular stage is sometimes referred to as a database or "db": the "hot db", the "warm db", the "cold db", and so on. More information regarding how the indexer stores indexes can be found in the [documentation](#).

SmartStore is an indexer capability that provides a way to use remote object stores, such as Amazon S3, to store indexed data. SmartStore introduces a remote storage tier and a cache manager. These features allow data to reside either locally on indexers or on the remote storage tier. Data movement between the indexer and the remote storage tier is managed by the cache manager, which resides on the indexer. With SmartStore, you can reduce the indexer local storage footprint and choose I/O optimized compute resources. Most data resides on remote storage, while the indexer maintains a local cache that contains a minimal amount of data: hot buckets, copies of warm buckets participating in active or recent searches, and bucket metadata.

The buckets of SmartStore indexes ordinarily have just two active states: hot and warm. The cold state, which is used with non-SmartStore indexes to distinguish older data eligible for moving to cheap storage, is not necessary with SmartStore because warm buckets already reside on inexpensive remote storage. Buckets roll directly from warm to frozen.

With SmartStore indexes, as with non-SmartStore indexes, hot buckets are built in the indexer's local storage cache. However, with SmartStore indexes, when a bucket rolls from hot to warm, a copy of the bucket is uploaded to remote storage. The remote copy then becomes the master copy of the bucket. Eventually, the cache manager evicts the local bucket copy from the cache. When the indexer needs to search a warm bucket for which it doesn't have a local copy, the cache manager fetches a copy from remote storage and places it in the local cache. The remote storage has a copy of every warm bucket. More information regarding Splunk SmartStore feature can be found in the [documentation](#).

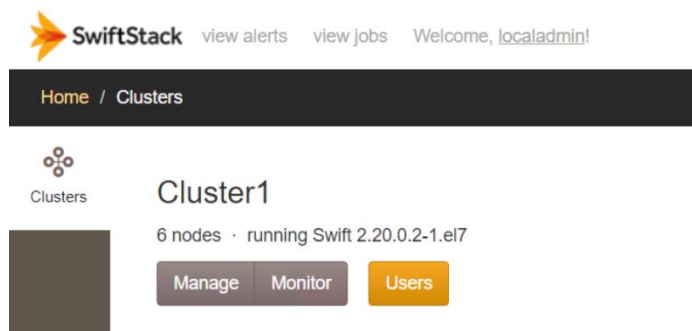
This CVD presents the Cisco HyperFlex storage as the Splunk indexer's local storage cache for hot and warm buckets, and the SwiftStack Object Storage System with Cisco UCS S3260 as the remote storage that maintains the copy of all the warm buckets.

There are two steps to do Splunk SmartStore configuration: 1) Create the remote object storage containers for SmartStore; 2) Configure the different SmartStore indexes with the location of the remote object storage.

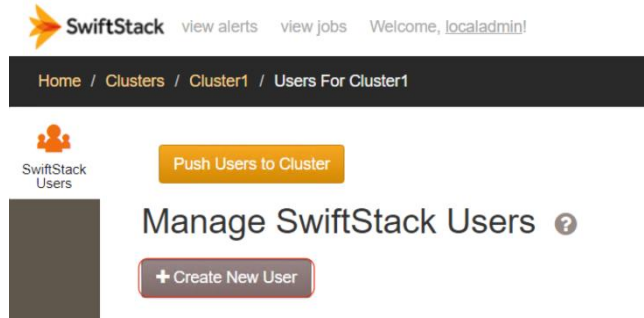
## Create SwiftStack Object Storage Containers for SmartStore

To create SwiftStack object storage containers for SmartStore, follow these steps:

1. Using a web browser, sign into the SwiftStack controller GUI.

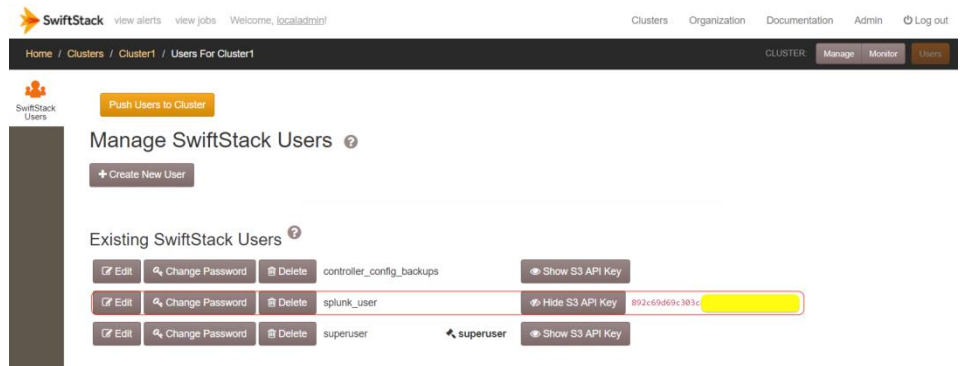


- From the Clusters window, choose the SwiftStack cluster you created during the installation. Click Users to create a new user for Splunk account.



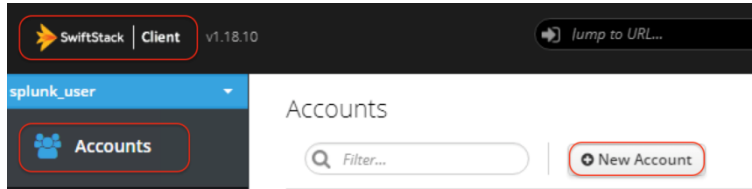
- Enter the user name, for example, we named the user splunk\_user, and the password. Click the box 'Enabled'. Select Submit.

- After creating the new user, click the button to display the user's S3 API key, and note this for later.



- Push this new user out to the SwiftStack cluster by clicking Push Users to Cluster.

6. You can also create the user account directly on the SwiftStack Client. Download the latest SwiftStack Client from SwiftStack website: <https://www.swiftstack.com/downloads> .
7. Install the latest SwiftStack Client.
8. Open the SwiftStack Client and create a user for the Splunk account.



9. Choose V1 as Auth Type. Enter the user name and password, and the Auth URL to the SwiftStack UCS-S3260 cluster. You can leave the other field empty.

New Account:

Auth Type  V1  V2  V3

Username

Password

Auth URL

Storage URL

Container

Optionally, specify a container to connect directly to when using this account.

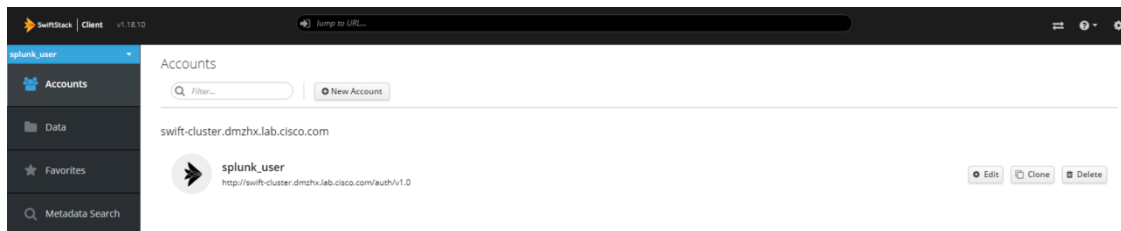
HTTP(S) Proxy

Specify a HTTP proxy URL for this account, if any.

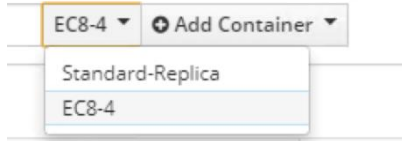
Allow insecure TLS connections

These account details have not yet been verified.

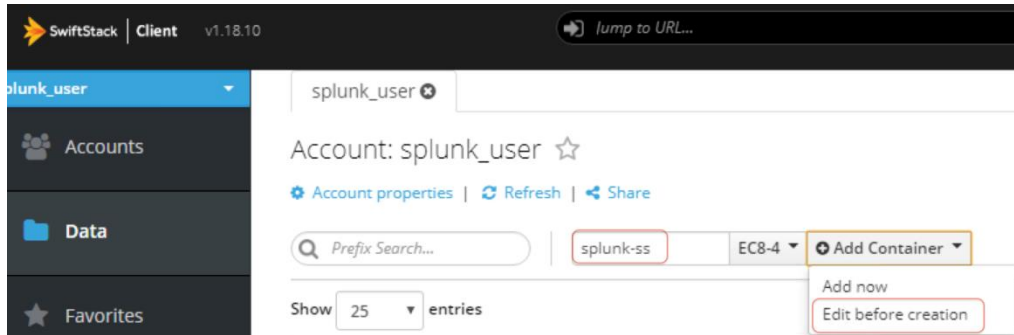
10. Click Save so the user will be created.



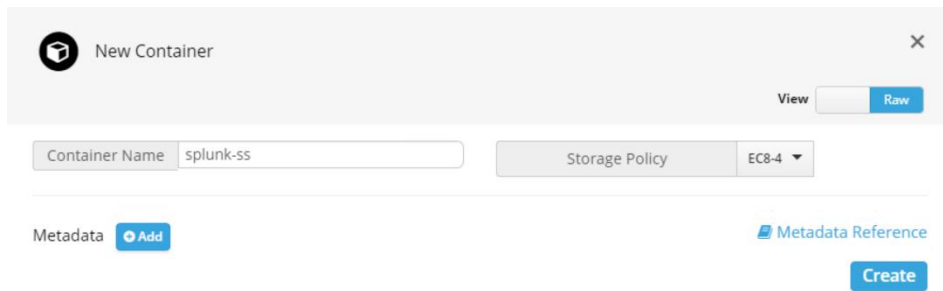
11. Click the Splunk user.
12. From the Data window, choose EC8-4 (Erasure Coding) as Replica policy.



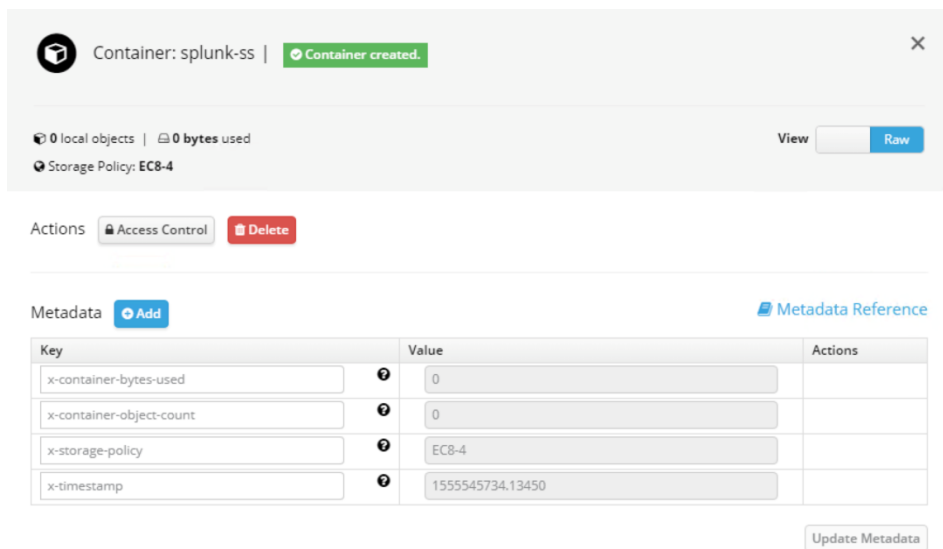
13. Enter splunk-ss as the name for the new container. Select Edit before creation.



14. In the New Container window, change the View model from Simple to Raw.



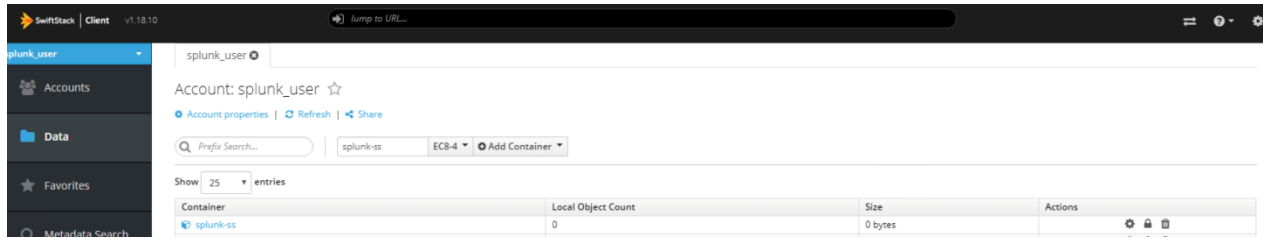
15. Click Create.



16. Leave everything set as default. Close the window by clicking the x.

17. Verify that the splunk-ss container has been created.





## Configure SmartStore Indexes

To configure the SmartStore indexes, follow these steps:

1. Log into the command line of the master node (admin1).
2. Navigate to `$SPLUNK_HOME/etc/master-apps/_cluster/local`.
3. Create and edit the file 'indexes.conf'. In the indexes.conf file you will need to configure the remote storage volume that is the remote location where SmartStore stores warm buckets, such as a location on an S3 bucket. You can configure the same SmartStore volume for all indexes, or you can specify different remote storage volumes for different SmartStore indexes. In this CVD a single remote volume is used for all SmartStore indexes. The indexer cluster can have both SmartStore and non-SmartStore indexes defined. In that way you need to configure SmartStore on a per-index basis. An example of indexes.conf file for configuring SmartStore indexes is provided in Appendix C: SmartStore indexes.conf File Example.

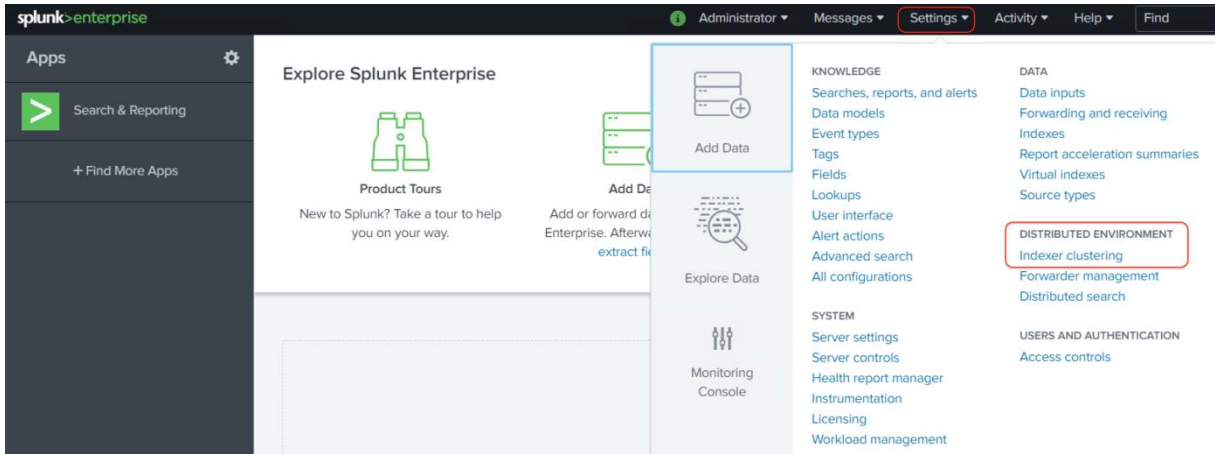
```
[splunk@admin1 local]$ cd $SPLUNK_HOME/etc/master-apps/_cluster/local
[splunk@admin1 local]$ vi indexes.conf
[splunk@admin1 local]$ vi indexes.conf
[splunk@admin1 local]$ cat indexes.conf | tail -25
[default]
remotePath = volume:s3/$_index_name
repFactor=auto

[volume:s3]
storageType = remote
path = s3://splunk-ss
remote.s3.access_key = splunk_user
remote.s3.secret_key = 892c69d69c303c87ddd40b31e28a737a
remote.s3.endpoint = http://swift-cluster.dmzhx.lab.cisco.com
remote.s3.auth_region = us-east-1

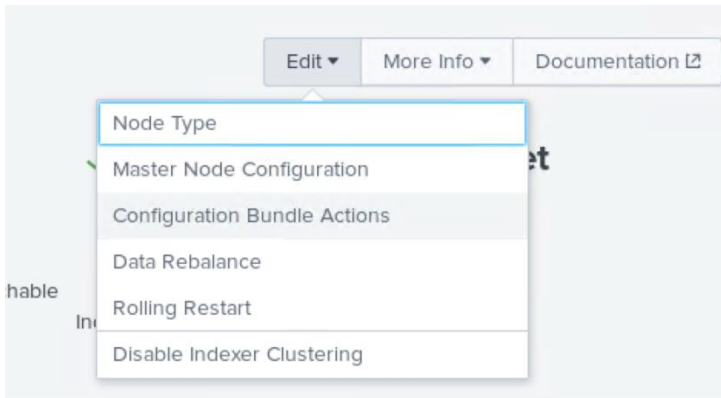
[cs_indexc]
homePath = $SPLUNK_DB/cs_indexc/db
coldPath=$SPLUNK_DB/cs_indexc/colddb
thawedPath=$SPLUNK_DB/cs_indexc/thaweddb
maxDataSize = auto
remotePath=volume:s3/$_index_name

[cs_indexl]
homePath = $SPLUNK_DB/cs_indexl/db
coldPath=$SPLUNK_DB/cs_indexl/colddb
thawedPath=$SPLUNK_DB/cs_indexl/thaweddb
maxDataSize = auto
remotePath=volume:s3/$_index_name
[splunk@admin1 local]$
```

4. Open the admin1 web management interface through browser. Navigate to Settings> Distributed Environment > Indexer Clustering.



5. Select Edit > Configuration Bundle Actions.



6. Select Validate and Check Restart option.

**Configuration Bundle Actions**

Click Push to distribute the configuration bundle to the set of peers. Optionally, validate the bundle and check if peer restart is required without distributing the bundle, or rollback to the previous bundle. [Learn More](#)

[Back to Master Node](#)

**Validate and Check Restart** **Push** **Rollback**

**Bundle Information:**

Updated Time \_\_\_\_\_ 4/17/2019, 12:24:06 PM  
 Active Bundle ID ? \_\_\_\_\_ 466078A97E1AD531540D99DEA348B0AB  
 Latest Bundle ID ? \_\_\_\_\_ 466078A97E1AD531540D99DEA348B0AB  
 Previous Bundle ID ? \_\_\_\_\_ D0D92B1A0A9523F9F94DB2ED98C999FC

i	Peer	Site	Status	Action Status
>	idx10	default	Up	None
>	idx9	default	Up	None
>	idx4	default	Up	None
>	idx8	default	Up	None
>	idx7	default	Up	None
>	idx6	default	Up	None
>	idx1	default	Up	None
>	idx3	default	Up	None
>	idx5	default	Up	None
>	idx2	default	Up	None

7. Wait until check Successful.

### Configuration Bundle Actions

Click Push to distribute the configuration bundle to the set of peers. Optionally, validate the bundle and check if peer restart is required.

[Back to Master Node](#)

[Validate and Check Restart](#) [Push](#) [Rollback](#)

---

**Last Validate and Check Restart: ✓ Successful**

Restart ? ..... Not Required  
Updated Time ..... 4/17/2019, 12:24:06 PM  
Active Bundle ID ? ..... 466078A97E1AD531540D99DEA348B0AB  
Latest Bundle ID ? ..... 466078A97E1AD531540D99DEA348B0AB  
Previous Bundle ID ? ..... D0D92B1A0A9523F9F94DB2ED98C999FC  
Latest Check Restart Bundle ? ... 8E489A3E85CDCDE8F8BE1CD18A2A5614

8. Select Push. Acknowledge the warning, and Push Changes.

### Distribute Configuration Bundle

Some configuration changes might require a restart of all peers. Would you like to push the changes? [Learn More](#)

[Cancel](#) [Push Changes](#)

The configuration changes are pushed to all the indexing peers. Rolling restart of all the peers are in progress.

splunk > enterprise Apps

### Configuration Bundle Actions

Click Push to distribute the configuration bundle to the set of peers. Optionally, validate the bundle and check if peer restart is required.

[Back to Master Node](#)

[Validate and Check Restart](#) [Push](#) [Rollback](#)

---

**C**

**Rolling restart of the peers is in progress.**

Peers Validated: 10 of 10

Peers Reloaded: 10 of 10

Restarting Peer: 1 of 10

When the Push is complete, the GUI should reflect a successful push.

## Configuration Bundle Actions

Click Push to distribute the configuration bundle to the set of peers. Optionally, validate the bundle.

[← Back to Master Node](#)

Validate and Check Restart

Push

Rollback

Last Push: ✓ Successful

Updated Time ..... 4/17/2019, 5:42:29 PM  
 Active Bundle ID ? ..... 8E489A3E85CDCDE8F8BE1CD18A2A5614  
 Latest Bundle ID ? ..... 8E489A3E85CDCDE8F8BE1CD18A2A5614  
 Previous Bundle ID ? ..... 466078A97E1AD531540D99DEA348B0AB

9. Log into one of the indexers through SSH.
10. Navigate to \$SPLUNK\_HOME/etc/slave-apps/\_cluster/local.
11. Verify that the file indexes.conf has been pushed to the indexers.

```
[splunk@idx1 ~]$ cd /data/disk1/splunk/etc/slave-apps/_cluster/local
[splunk@idx1 local]$ ls -lts
total 24
4 -rw-rw-r-- 1 splunk splunk 1780 Apr 17 17:42 indexes.conf
4 -rw-rw-r-- 1 splunk splunk 1910 Apr 17 17:42 indexes.conf.1
4 -rw-rw-r-- 1 splunk splunk 2137 Apr 17 17:42 indexes.conf.2
4 -rw-rw-r-- 1 splunk splunk 1799 Apr 17 17:42 indexes.conf.bkp
4 -rw-rw-r-- 1 splunk splunk 41 Apr 17 17:42 inputs.conf
4 -r--r--r-- 1 splunk splunk 233 Apr 17 17:42 README
[splunk@idx1 local]$ cat indexes.conf | tail -25
[default]
remotePath = volume:s3/$_index_name
repFactor=auto

[volume:s3]
storageType = remote
path = s3://splunk-ss
remote.s3.access_key = splunk user
remote.s3.secret_key = 892c69d69c303c87ddd40b31e28a737a
remote.s3.endpoint = http://swift-cluster.dmzhx.lab.cisco.com
remote.s3.auth_region = us-east-1

[cs_indexc]
homePath = $SPLUNK_DB/cs_indexc/db
coldPath=$SPLUNK_DB/cs_indexc/colddb
thawedPath=$SPLUNK_DB/cs_indexc/thaweddb
maxDataSize = auto
remotePath=volume:s3/$_index_name

[cs_index1]
homePath = $SPLUNK_DB/cs_index1/db
coldPath=$SPLUNK_DB/cs_index1/colddb
thawedPath=$SPLUNK_DB/cs_index1/thaweddb
maxDataSize = auto
remotePath=volume:s3/$_index_name
[splunk@idx1 local]$
```

## Validation Testing

### Data Ingestion

For the validation testing, first you need to ingest the data into the Splunk cluster so that you can monitor the cluster's indexing and searching efficiency. You can observe how the data was tiered and replicated in the cluster. You can also observe how Splunk's S3 architecture leverages both local storage on Cisco HyperFlex systems and SwiftStack's remote object storage together. There are a few ways to generate simulated data for testing and three of them are briefly described here:

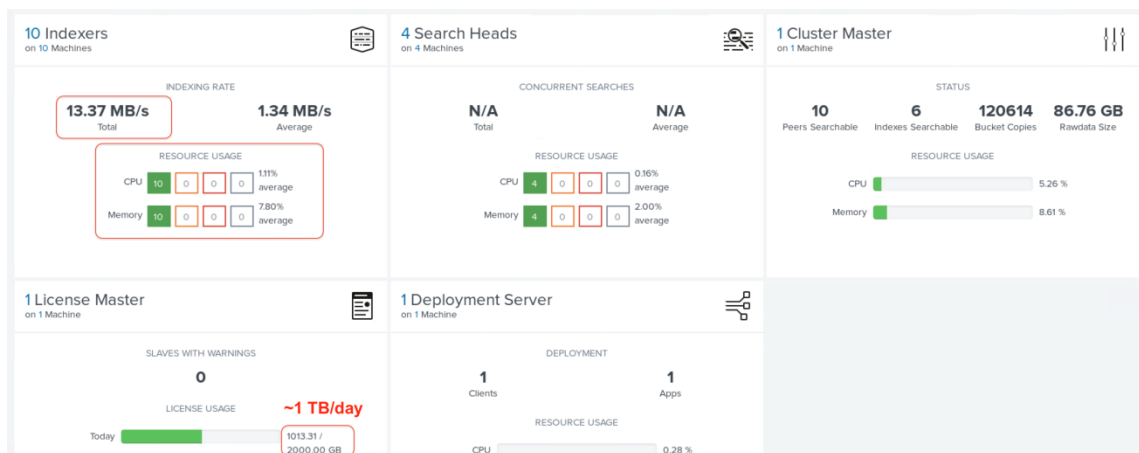
- Use the event generator provided by Splunk. This *eventgen* utility can be downloaded from Github: <https://github.com/splunk/eventgen>.
- Use the Splunk UI to add the data to the Splunk cluster. This can come from different data sources and for different purposes. If random syslog or file in which each line is a new event is used, it is suggested that syslog data be used for verification due to the known and expected format. The recommended file size is at minimum ~250MB or 1m events. Details of these options can be found in the Splunk documentation at <http://docs.splunk.com/Documentation/Splunk/latest/Indexer/forwardersdirecttopeers>.
- Use a custom methodology to generate the events.

How you simulate your data set really depends on the data patterns from your applications in your IT environment. Please consult with Splunk or your software vendors for the guidance.

In this solution a custom methodology is used that uses a command such as logger on multiple Linux servers to generate the logging events and ingest those events into Splunk. The logging data was generated from the load generators and sent directly to all the indexers without an universal forwarder. Appendix D: Custom Event Generation Script is an example of a logger script which can be run in multiple instances across a number of virtual machines.

To ingest the data into the Splunk cluster, follow these steps:

1. Start to generate the logging events at 1 TB/day (~13 MB/s) and send the data to the SmartStore index *cs\_indexc* that is distributed across all the ten indexers. Monitor the indexing performance from the Distributed Monitoring Console (DMC). Pay attention to the resource usage including CPU and Memory.



2. During the data ingestion, do some searches on the index *cs\_indexc*, monitor the searching performance and pay attention to the resource usage including CPU and Memory.

**host**  
23 Values, 100% of events

Top 10 Values	Count	%
192.168.11.11	314,150	6.984%
192.168.11.12	313,819	6.976%
192.168.11.8	260,773	5.797%
192.168.11.13	258,520	5.747%
192.168.11.5	238,895	5.311%
192.168.11.7	238,349	5.298%
192.168.11.3	237,366	5.277%
192.168.11.2	237,177	5.272%
192.168.11.24	236,166	5.25%
192.168.11.20	234,898	5.222%

- Keep sending the data at the rate of 1 TB/day for seven days. Then increase the load generation so the logging events are generated and sent to the SmartStore index cs\_indexc at 2 TB/day. Monitor the indexing performance from the Distributed Monitoring Console (DMC). Pay attention to the resource usage including CPU and Memory.

**10 Indexers** (on 10 Machines)

- INDEXING RATE: Total 23.82 MB/s, Average 2.38 MB/s
- RESOURCE USAGE: CPU 10 (0, 0, 0, 0) average 219%, Memory 10 (0, 0, 0, 0) average 1110%

**4 Search Heads** (on 4 Machines)

- CONCURRENT SEARCHES: Total N/A, Average N/A
- RESOURCE USAGE: CPU 4 (0, 0, 0, 0) average 0.11%, Memory 4 (0, 0, 0, 0) average 2.00%

**1 Cluster Master** (on 1 Machine)

- STATUS: 10 Peers Searchable, 6 Indexes Searchable
- 532877 Bucket Copies, 234.66 GB Rawdata Size
- RESOURCE USAGE: CPU 0.45%, Memory 11.56%

**1 License Master** (on 1 Machine)

- SLAVES WITH WARNINGS: 0
- LICENSE USAGE: Today 1990.92 / 2000.00 GB (2 TB/day)
- RESOURCE USAGE: CPU 0.25%, Memory 9.60%

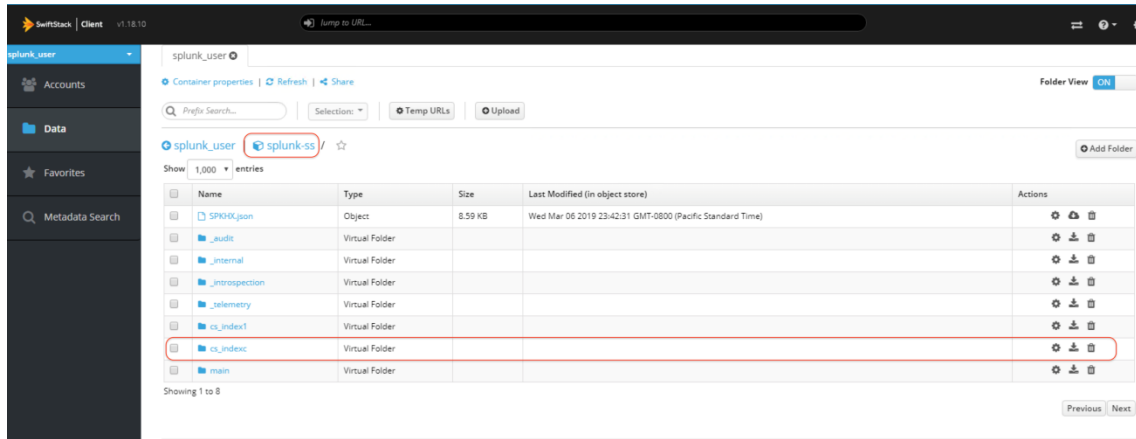
**1 Deployment Server** (on 1 Machine)

- DEPLOYMENT: 1 Clients, 1 Apps
- RESOURCE USAGE: CPU 0.18%, Memory 7.19%

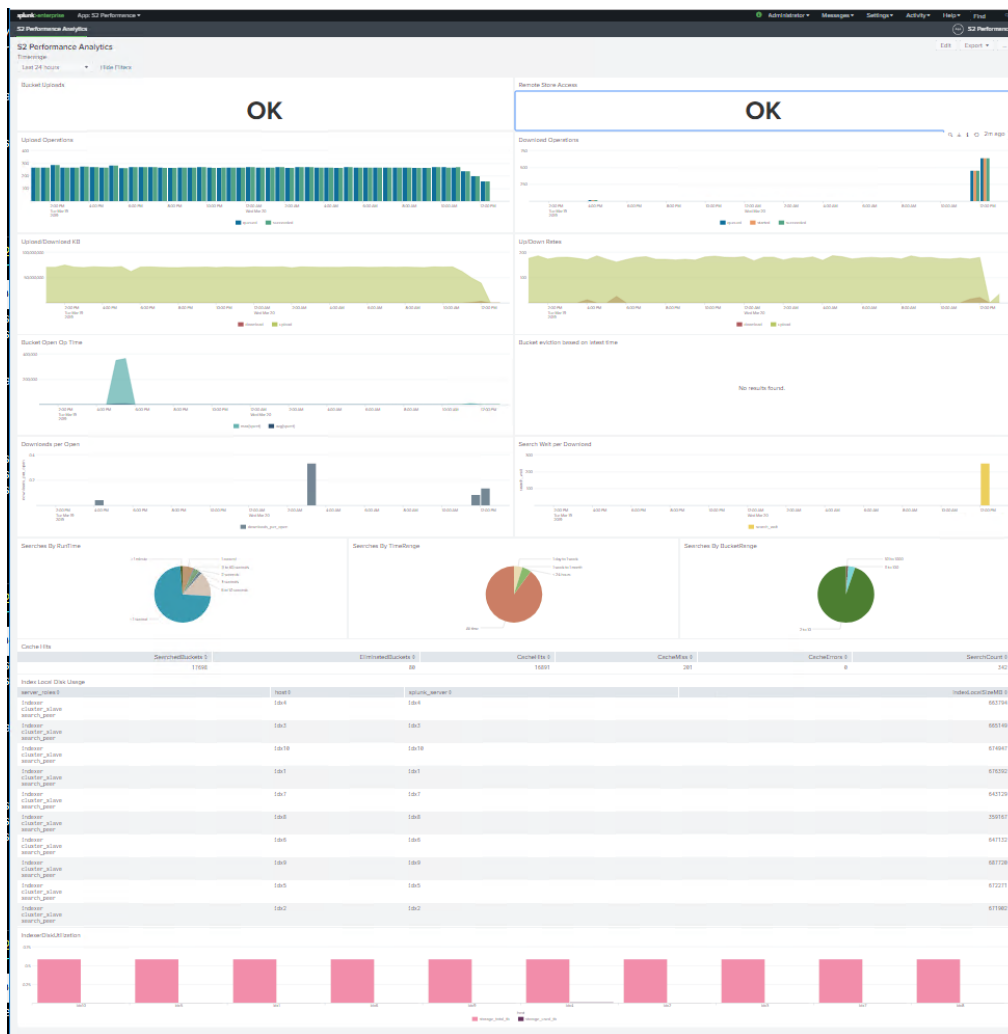
0 Triggered Alerts

- During the data ingestion, do some searches on the index cs\_indexc, monitor the searching performance and pay attention to the resource usage including CPU and Memory.





7. Install the S2 Performance Analytics application add-on provided by Splunk on the master node of the indexer cluster. Monitor the S2 performance statistics using this application. Pay attention to the Upload/Download operations and rates, the cache hits and the disk utilization on the indexers.



For details about how to observe indexing performance, refer to Splunk’s documentation here: <https://docs.splunk.com/Documentation/Splunk/latest/Indexer/Viewindexingperformance>.



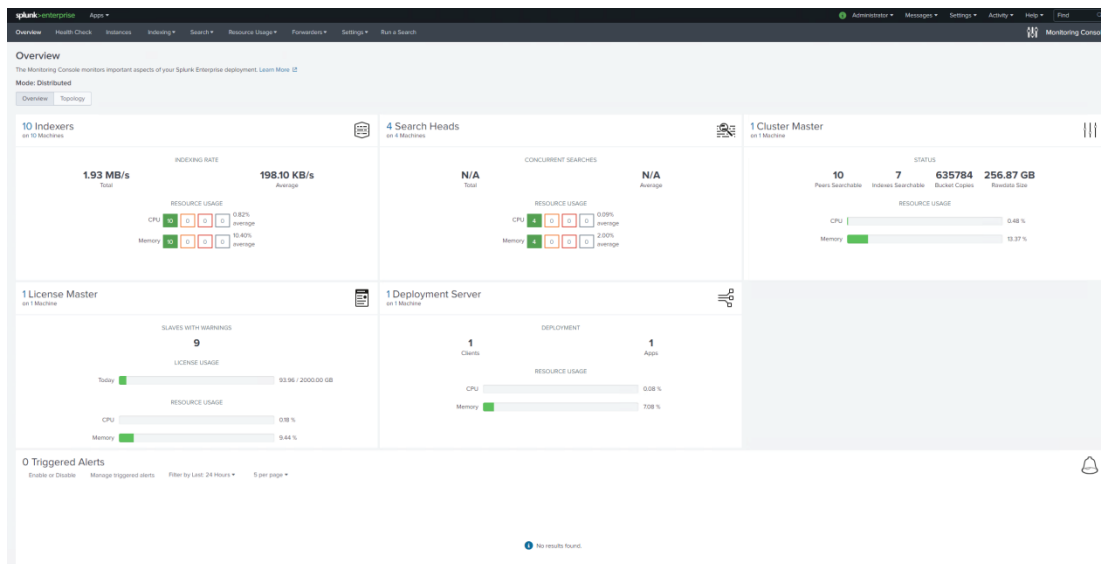
For details about how search types including dense, super dense, rare, and super rare affect Splunk performance, refer to Splunk's documentation here:  
<https://docs.splunk.com/Documentation/Splunk/latest/Capacity/HowsearchtypesaffectSplunkEnterpriseperformance> .

## Verify Data Replication

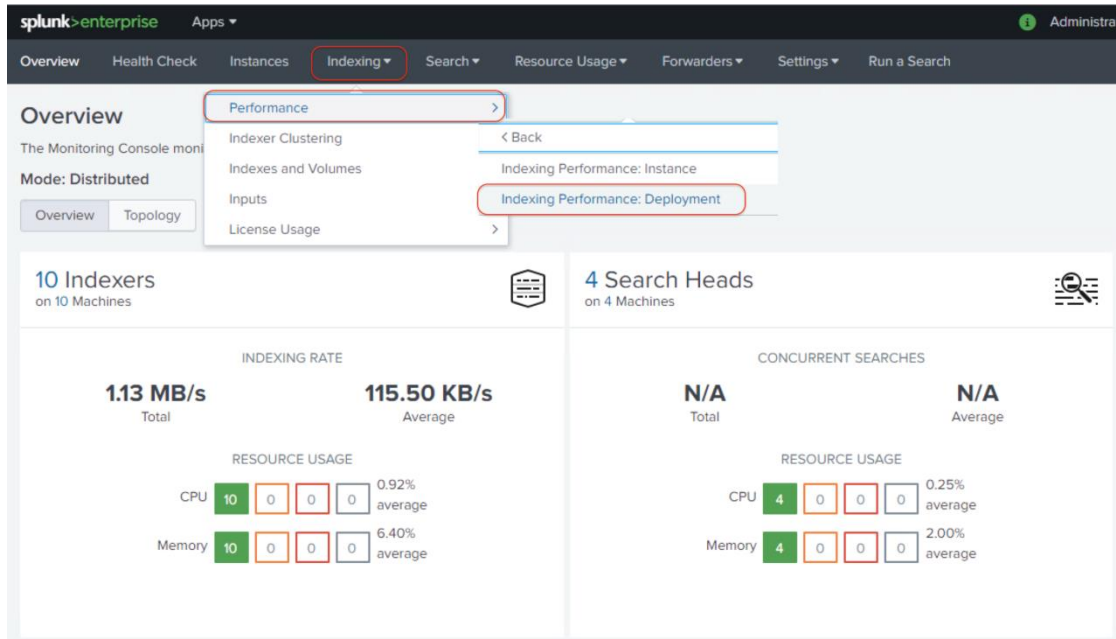
During the installation and configuration of the Splunk cluster it has been verified that the connectivity and relationships between the indexers, search heads, license master, master node, and the Distributed Monitoring Console. Now verify that data is distributed across indexer nodes and is replicated across the active nodes when an indexer is down.

To verify data replication, follow these steps:

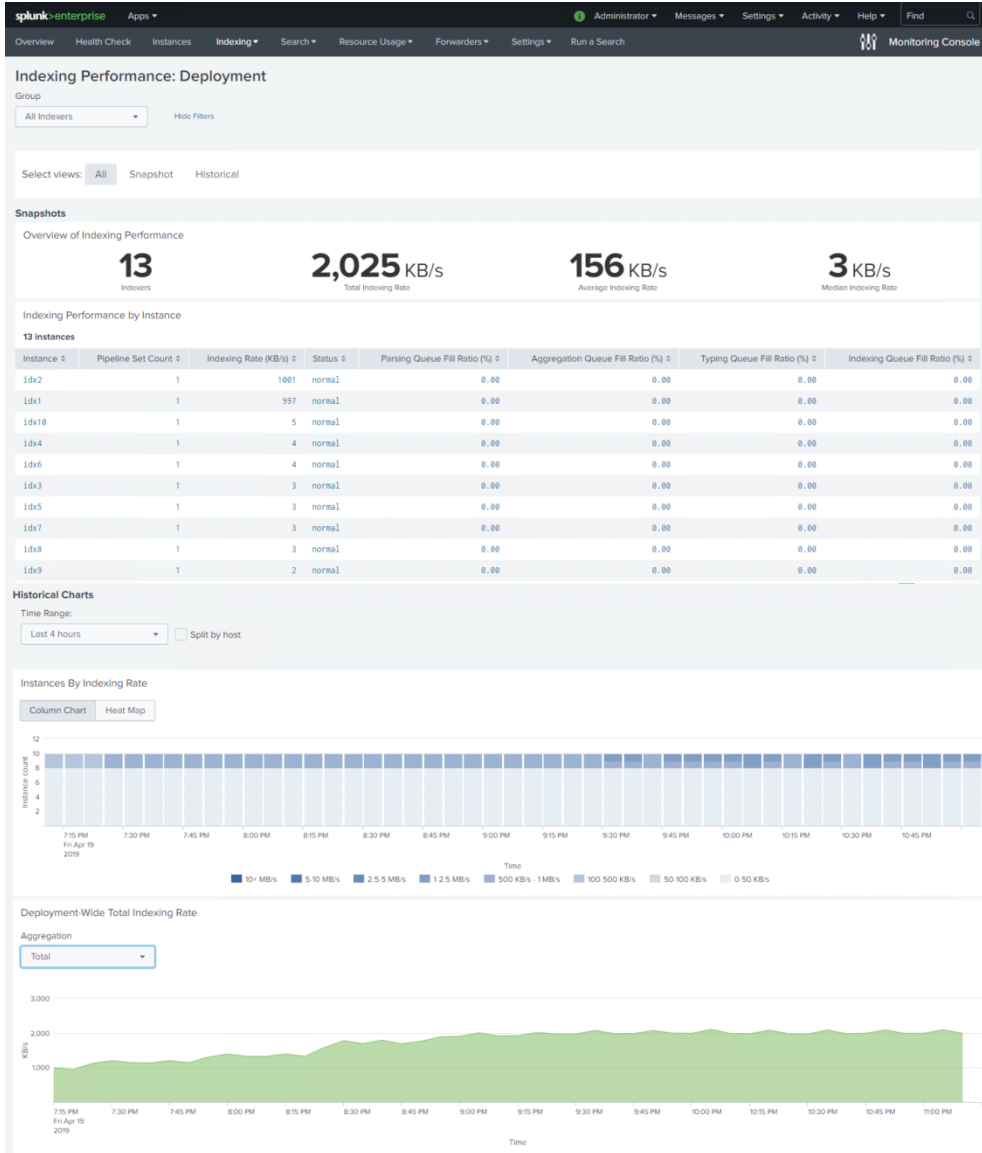
1. Generate the logging events and send to the SmartStore index `cs_index1` at 160 GB/day (two load generators sending the data to `idx1` and `idx2`). Reduce the ingesting rate here so you can prepare smaller amount of data in a new SmartStore index for the cache performance testing.



2. Keep sending the data for seven days.
3. While ingesting the data, log into the management GUI of the master node from the web browser.
4. Navigate to the distributed monitoring console by clicking Settings > Monitoring Console.
5. In DMC, select Indexing > Performance > Indexing Performance: Deployment.

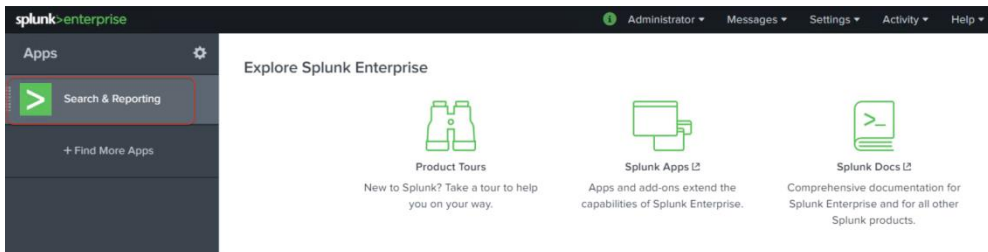


6. Note the table and charts that reflect the indexing and rates and data passing through the system.



7. Navigate to the management GUI of one search head in the web browser.

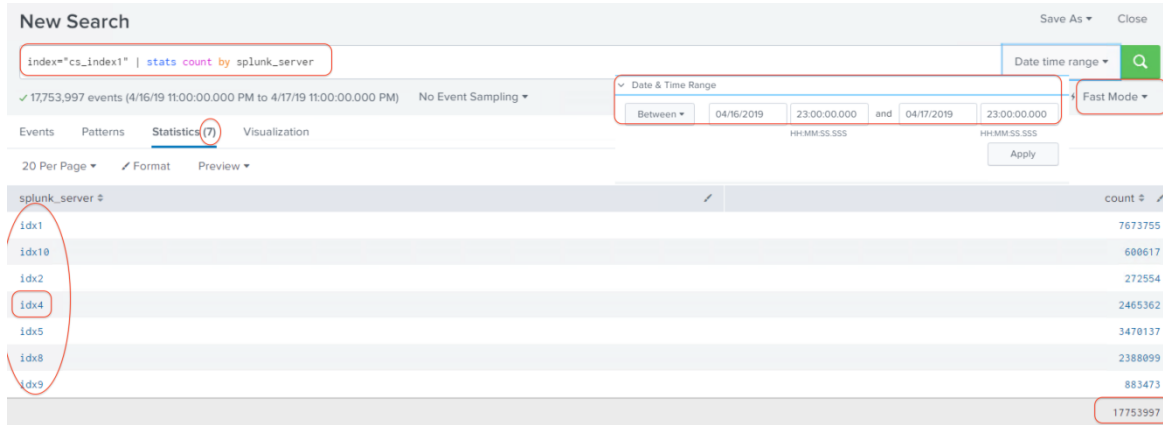
8. Click Search & Reporting.



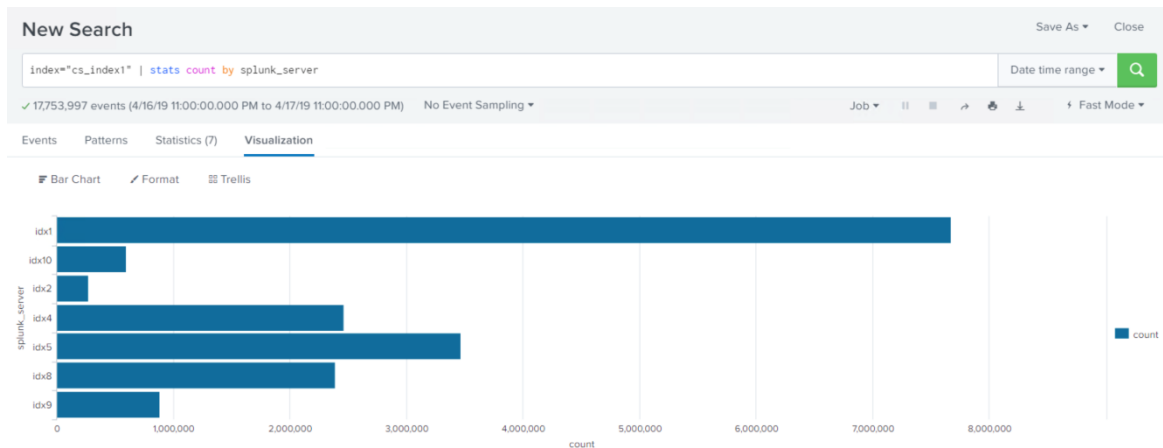
9. In the New Search bar, enter the following search: `index="cs_index1" | stats count by splunk_server`.

10. Change the time range to a historical one-day range.

- Change the search mode to fast mode.
- Click the Search icon (magnifying glass in green color) to start searching.
- In the view of Statistics, record the number of events per indexer, as well as the total number of events, visible in the panel as well as the event summary under the search bar.



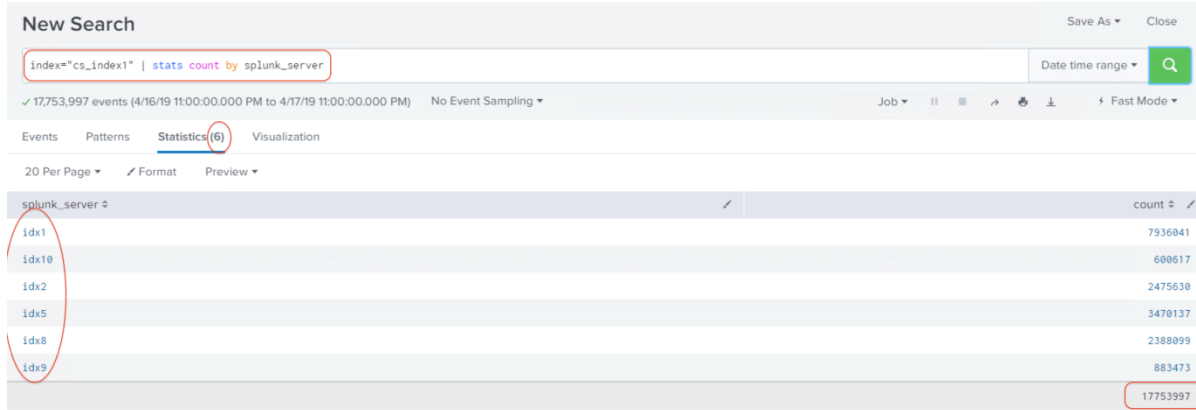
- Change view to Visualization and set the chart type to Column. Note the distribution of data across each of the indexers.



- Log into one of the indexers which reported data (such as idx4) through SSH.
- Issue the command `sudo /usr/bin/systemctl stop Splunkd.service` to stop Splunkd service on that indexer.
- Check the status of Splunkd service, ensure the process is inactive.

```
[splunk@idx4 ~]$ sudo /usr/bin/systemctl stop Splunkd.service
[splunk@idx4 ~]$ systemctl status Splunkd
● Splunkd.service - Systemd service file for Splunk, generated by 'splunk enable boot-start'
   Loaded: loaded (/etc/systemd/system/Splunkd.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since Fri 2019-04-19 23:57:33 PDT; 5s ago
     Process: 2291 ExecStartPost=/bin/bash -c chown -R splunk:splunk /sys/fs/cgroup/memory/system.slice/%n (code=exited, status=0/SUCCESS)
     Process: 2289 ExecStartPost=/bin/bash -c chown -R splunk:splunk /sys/fs/cgroup/cpu/system.slice/%n (code=exited, status=0/SUCCESS)
     Process: 2288 ExecStart=/data/disk1/splunk/bin/splunk _internal_launch_under_systemd (code=killed, signal=TERM)
   Main PID: 2288 (code=killed, signal=TERM)
[splunk@idx4 ~]$
```

- Return to the search head's web GUI and run the same search again.



19. Note the distribution of events and the total. The event count from this search and the previous (step 9) should be the same. However, idx4 is not on the distributed splunk\_server list any more.
20. While one indexer is down, this step has verified that the indexed data has been replicated and that the search results are consistent, even when an indexer is not functioning.
21. Bring your indexer back up running the command `sudo /usr/bin/systemctl start Splunkd.service`.

```
[splunk@idx4 ~]$ sudo /usr/bin/systemctl start Splunkd.service
[splunk@idx4 ~]$ systemctl status Splunkd
● Splunkd.service - Systemd service file for Splunk, generated by 'splunk enable boot-start'
   Loaded: loaded (/etc/systemd/system/Splunkd.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2019-04-19 23:59:04 PDT; 13s ago
     Process: 10361 ExecStartPost=/bin/bash -c chown -R splunk:splunk /sys/fs/cgroup/memory/system.slice/%n (code=exited, status=0/SUCCESS)
     Process: 10358 ExecStartPost=/bin/bash -c chown -R splunk:splunk /sys/fs/cgroup/cpu/system.slice/%n (code=exited, status=0/SUCCESS)
    Main PID: 10357 (splunkd)
      Memory: 753.4M (Limits: 100.0G)
      CGroup: /system.slice/Splunkd.service
              └─10357 splunkd --under-systemd --systemd-delegate=yes -p 8089 _internal_launch_under_systemd
                  └─10426 [splunkd pid=10357] splunkd --under-systemd --systemd-delegate=yes -p 8089 _internal_launch_under_systemd [process-runner]
                      └─10443 mongod --dbpath=/data/disk1/splunk/var/lib/splunk/kvstore/mongo --storageEngine=mmapv1 --port=8191 --timeStampFormat=iso8601-utc
[splunk@idx4 ~]$
```

### Verify Transfer of Warm Buckets to the Remote Storage

With Splunk Smartstore, indexing and searching of data occurs on the indexers, just as in a traditional deployment that stores all data locally. The main difference is that warm buckets for an S3 index are stored on a remote object store, rather than locally. A cache manager on the indexer fetches copies of warm buckets from the remote store and places them in the indexer's local cache when the buckets are needed for a search. The cache manager also evicts warm bucket copies from the cache once their likelihood of being searched again diminishes.

With S3 indexes, as with non-S3 indexes, hot buckets are also built in the indexer's local cache. However, with S3, when a bucket rolls from hot to warm, a copy of the bucket is uploaded to remote storage. Eventually, the cache manager evicts the local bucket copy from the cache depending on how recent the searches on that bucket were. When the indexer needs to search a warm bucket for which it doesn't have a local copy, the cache manager fetches a copy from remote storage and places it in the local cache.

The remote storage has a copy of every warm bucket. Each indexer's local cache contains several types of data:

- Hot buckets which are created in local storage. They continue to reside on the indexer until they roll to warm.
- Copies of warm buckets that are currently participating in searches.
- Copies of recently created or recently searched warm buckets. The indexer maintains a cache of warm buckets, to minimize the need to fetch the same buckets from remote storage repeatedly.
- Metadata for remote buckets. The indexer maintains a small amount of information about each bucket in remote storage.

Each indexer incorporates a cache manager that manages the SmartStore data in local storage. The cache manager attempts to maintain in local storage any data that is likely to participate in future searches. The cache manager performs these functions:

- It copies buckets from local to remote storage when the buckets roll from hot to warm.
- It fetches bucket files from remote storage into the local storage cache when a search needs the files.
- It evicts files from the local storage cache when they are no longer likely to be needed for future searches.
- While considering the sizing of the local cache configuration it depends on the following factors:
  - How long will the cache be retained period,
  - Replication Factor for the cluster,
  - The data compression ratio (default 50 percent), and
  - The data Ingesting rate.

In this section it is depicted how to verify the integrity of the warm buckets when they are evicted from the local cache to the remote S3 storage volumes. We can wait for the eviction happen when the local cache gets full with ingesting large amount of events; or configure the `maxDataSize` to a small number for a SmartStore index to force more frequent transfer; or use the following commands to manually initiate rolling the hot data to warm buckets and evicting warm buckets to the remote S3 storage:

- Manually force the hot data to roll to warm buckets:
  - `splunk _internal call /data/indexes/<index_name>/roll-hot-buckets -auth<admin>:<password>`
  - Evict all data from the cache (use the cache manager `_evict` call with a large amount of free space, such as 1000000000):
  - `splunk _internal call /services/admin/cacheman/_evict -post:mb 1000000000 -post:path $SPLUNK_DB -method POST`

To validate the testing, follow these steps:



One common problem that might impact SmartStore functions is connectivity with the remote storage. Connectivity problems can result from network or permissions issues.

---

1. Verify the SwiftStack S3 container is accessible as the remote store from outside of the Splunk servers.
2. Run the aws command on any Linux server (such as admin virtual machine) and ensure the bucket on the SwiftStack cluster returns the appropriate contents.

```
[root@admin ~]# pwd
/root
[root@admin ~]# mkdir .aws
[root@admin ~]# cd .aws
[root@admin .aws]# vi config
[root@admin .aws]# cat config
[profile default]
aws_access_key_id = splunk_user
aws_secret_access_key = 892c69d69c[REDACTED]
region = us-east-1

[root@admin .aws]# aws --endpoint-url http://swift-cluster.dmzhx.lab.cisco.com s3 ls s3://splunk-ss
PRE _audit/
PRE _internal/
PRE introspection/
PRE telemetry/
PRE cs_index1/
PRE cs_indexc/
PRE main/
2019-03-06 15:42:31      8593 SPKHX.json
[root@admin .aws]#
```

- Run the following command to test the remote storage access from the indexer cluster:

[splunk@idx1 ~]\$ splunk cmd splunkd rfs -- ls --starts-with volume:s3/cs\_index1/db/

The following screenshot partially displays the last output data across an S3 bucket.

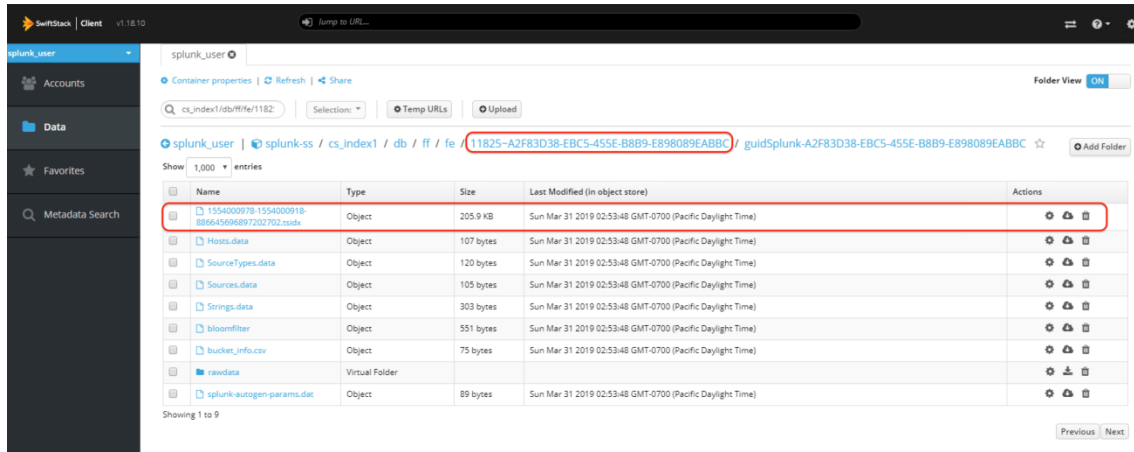
```
9,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/.rawSize
9,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/sizeManifest4.1
205892,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/1554033014-13046742936233146254.tsidx
107,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/Hosts_data
120,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/SourceTypes_data
105,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/Sources_data
303,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/Strings_data
553,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/bloomfilter
75,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/bucket_info.csv
629642,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/rawdata/journal.gz
1052,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/rawdata/slicemin.dat
12015,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/rawdata/slicesv2.dat
89,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/splunk-autogen-params.dat
1745,cs_index1/db/fff/12372-A2F83D38-EBC5-455E-B8B9-E898089EABBC/receipt.json
9,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/.rawSize
9,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/sizeManifest4.1
205896,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/1554000918-886645696897202702.tsidx
120,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/Hosts_data
105,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/SourceTypes_data
303,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/Strings_data
553,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/bloomfilter
75,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/bucket_info.csv
630802,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/rawdata/journal.gz
1071,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/rawdata/slicemin.dat
12016,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/rawdata/slicesv2.dat
89,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/splunk-autogen-params.dat
1741,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/receipt.json
[splunk@idx1 ~]$
```

- Choose the last bucket from the output and run the command:

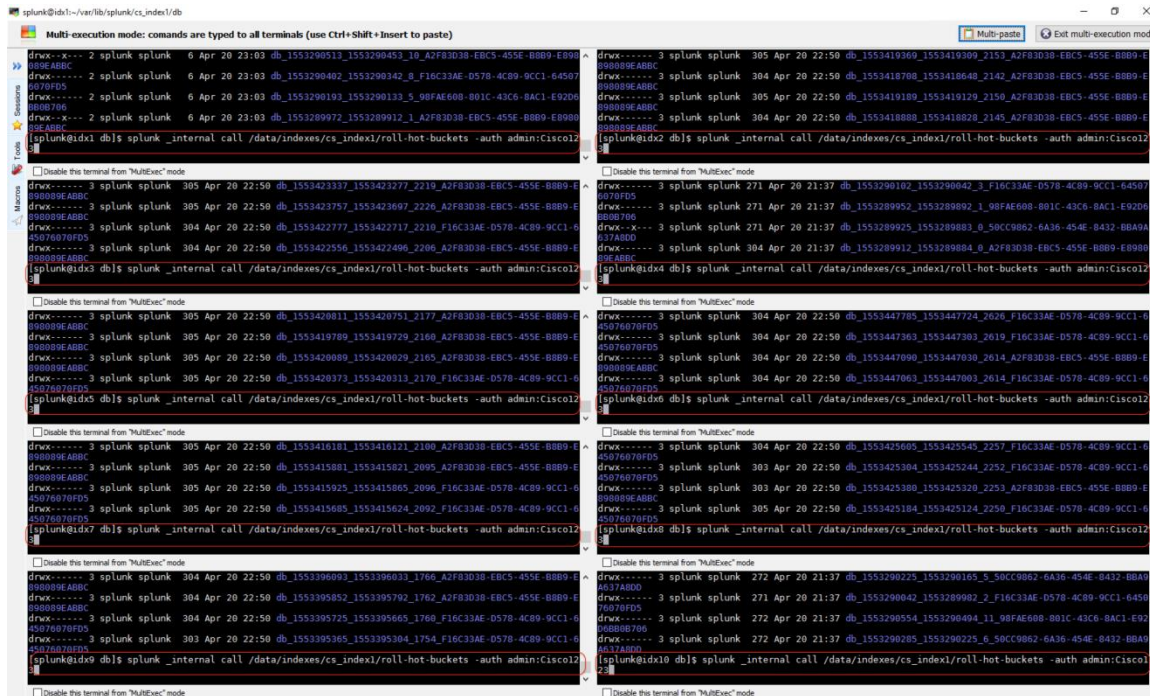
[splunk@idx1 ~]\$ splunk cmd splunkd rfs -- ls bucket:cs\_index1~11825~A2F83D38-EBC5-455E-B8B9-E898089EABBC

```
[splunk@idx1 ~]$ splunk cmd splunkd rfs -- ls bucket:cs_index1~11825~A2F83D38-EBC5-455E-B8B9-E898089EABBC
#For full paths run: splunkd rfs -- ls --starts-with volume:s3/cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/
size,name
9,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/.rawSize
9,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/sizeManifest4.1
205896,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/1554000918-886645696897202702.tsidx
120,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/Hosts_data
105,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/SourceTypes_data
303,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/Strings_data
553,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/bloomfilter
75,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/bucket_info.csv
630802,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/rawdata/journal.gz
1071,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/rawdata/slicemin.dat
12016,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/rawdata/slicesv2.dat
89,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/guidSplunk-A2F83D38-EBC5-455E-B8B9-E898089EABBC/splunk-autogen-params.dat
1741,cs_index1/db/fff/11825-A2F83D38-EBC5-455E-B8B9-E898089EABBC/receipt.json
[splunk@idx1 ~]$
```

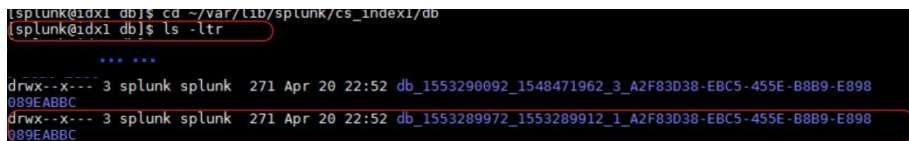
The output from the command should match what was seen in the splunk-ss bucket through the SwiftStack Client.



5. Stop ingesting the log events to the Splunk cluster.
6. Use the roll-hot-buckets command on all indexer nodes to manually force all hot data to roll to warm buckets.



7. From the command line interface on idx1 node, navigate to the directory \$SPLUNK\_HOME/var/lib/splunk/cs\_index1/db, then run ls -ltr command.



8. Go to the last bucket (If the last bucket gets evicted to SwiftStack, it is safe to assume that all the previous buckets also got flushed.), list the contents inside it. Note down the file size of the .tsdix file in this folder.
9. Run the md5sum command on this .tsdix file and write down MD5 checksum number for this file. This MD5 number is needed to verify the file integrity with comparison to the same file on the SwiftStack S3 volume.



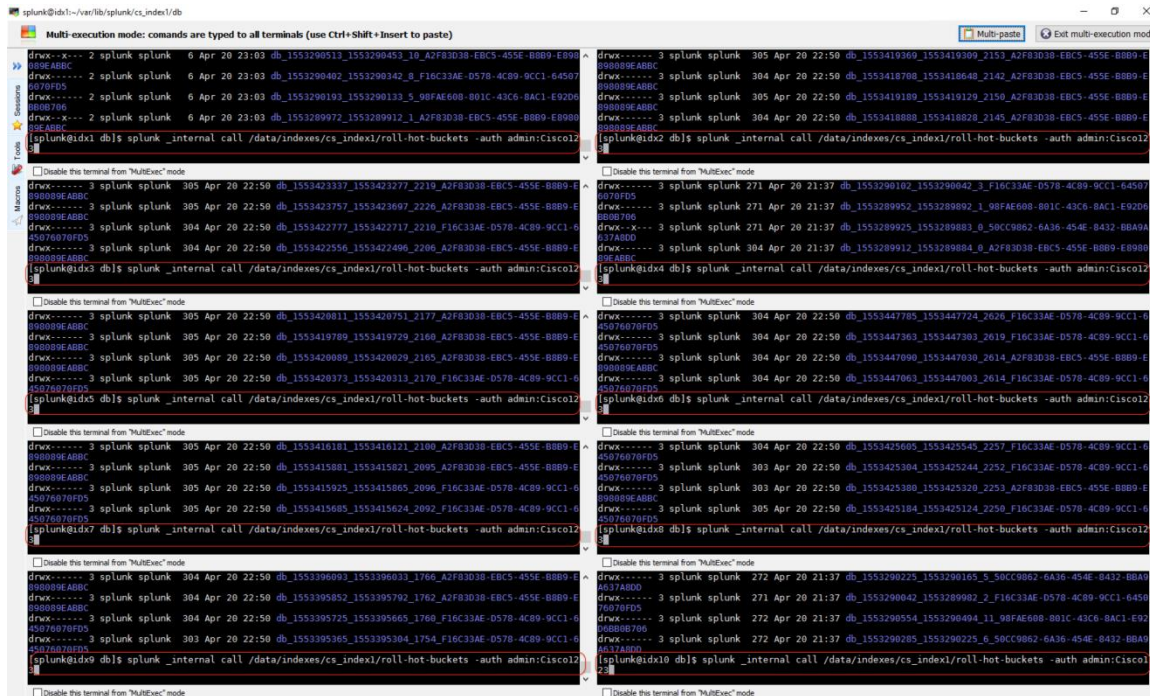


- Use consecutive searches to partially populate Splunk's cache as desired (e.g., first search for events in the past 1/2/3/4 days, then for a subsequent search the events for the past 4 days so the searching scenarios should be ~25%, ~50%, ~75% and ~100% cached).
- Observe and record the searching time using the Distributed Monitoring Console on the master node.

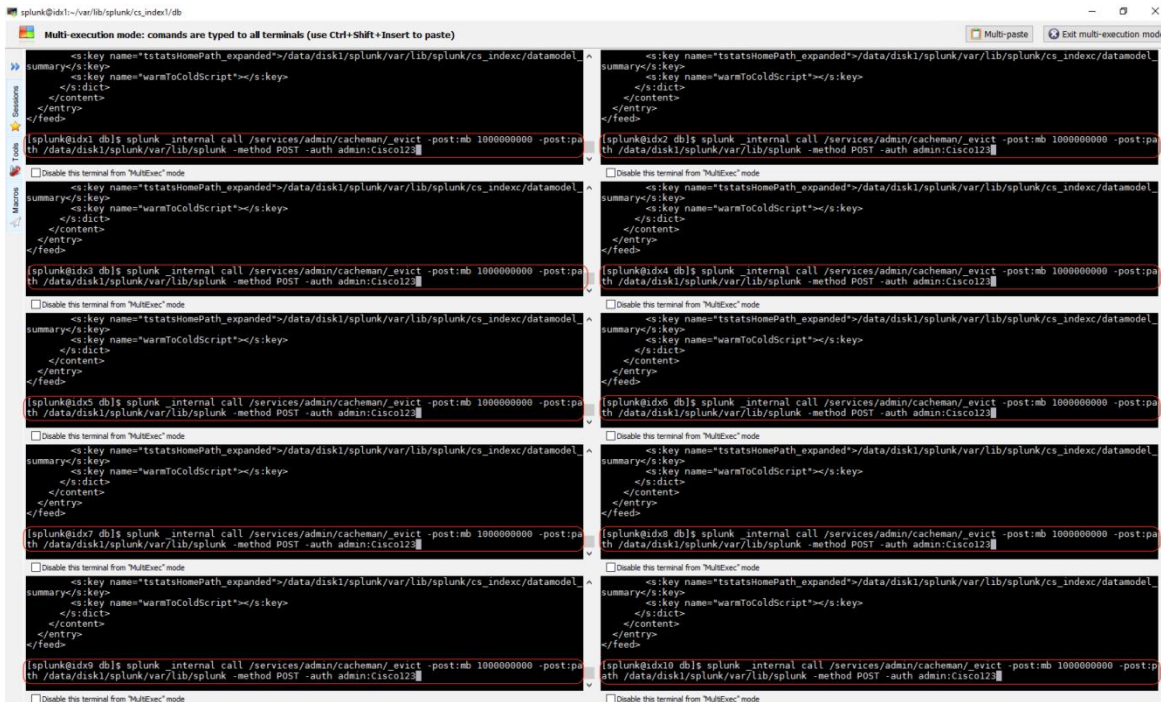
SmartStore Search: 0% Cached

Follow these steps:

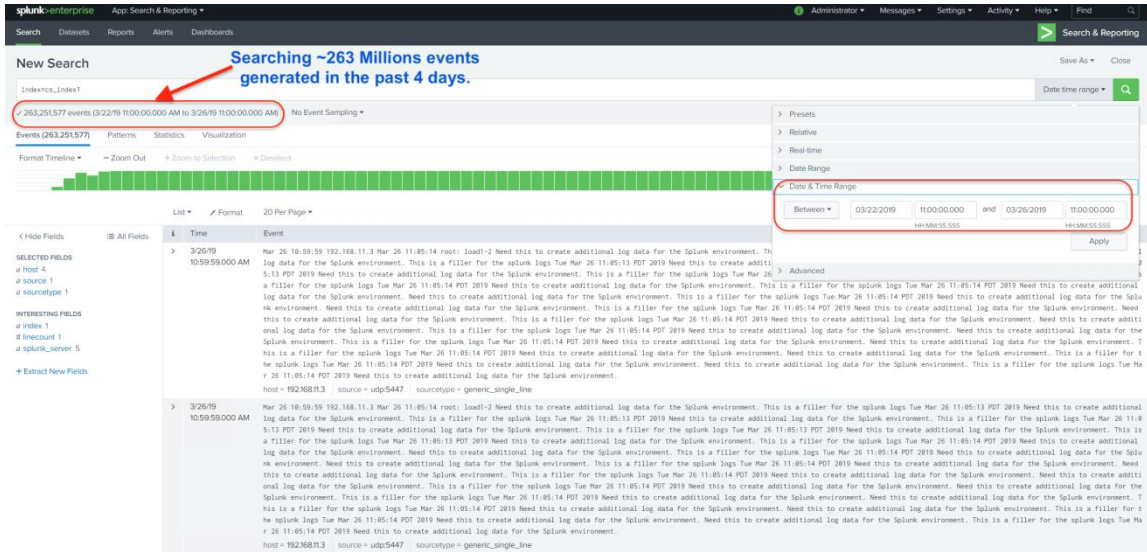
1. Use the 'roll-hot-buckets' command on all indexer nodes to manually force all hot data to roll to warm buckets.



2. Use the `splunk _internal call /services/admin/cacheman/_evict` command on all indexer nodes to evict all data from the cache. The execution of this command will ensure that any subsequent search will fetch data from SwiftStack S3 target and not read any data from cache.



3. Complete a search on the SmartStore index cs\_index1 with a past four-day time range.



4. Go to the DMC on the Splunk master node, click Search > Activity > Search Usage Statistics: instance. Record the Search Runtime for the above search.

Search #	Search	Search Runtime #	Search Start #	Earliest Time #	Latest Time #	Type #	User #
1	search index=cs_index1	7min 12.38s	04/10/2019 23:47:08 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
2	search index=cs_index1	8min 38.42s	04/10/2019 23:37:49 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
3	search index=cs_index1	10min 30.45s	04/10/2019 23:27:10 -0700	Sat Mar 23 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
4	search index=cs_index1	10min 16.46s	04/10/2019 23:12:11 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
5	search index=cs_index1	6min 54.34s	04/10/2019 23:03:49 -0700	Sun Mar 24 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
6	search index=cs_index1	11min 52.51s	04/10/2019 22:50:31 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
7	search index=cs_index1	3min 24.25s	04/10/2019 22:46:36 -0700	Mon Mar 25 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
8	search index=cs_index1	13min 30.52s	04/10/2019 22:22:45 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin

### SmartStore Search: 25% Cached

Follow these steps:

1. Use the roll-hot-buckets command on all indexer nodes to manually force all hot data to roll to warm buckets.
2. Use the splunk \_internal call /services/admin/cacheman/\_evict command on all indexer nodes to evict all data from the cache. The execution of this command will ensure that any subsequent search will fetch data from SwiftStack S3 target and not read any data from cache.
3. Complete a search on the SmartStore index cs\_index1 with a past 1-day time range.
4. Subsequently complete a search on the SmartStore index 'cs\_index1' with a past 4-day time range.
5. Go to the DMC on the Splunk master node, click Search > Activity > Search Usage Statistics: instance. Record the Search Runtime for the above search.

Search	Search Runtime	Search Start	Earliest Time	Latest Time	Type	User
1 search index=cs_index1	7min 12.38s	04/10/2019 23:47:08 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
2 search index=cs_index1	8min 38.42s	04/10/2019 23:37:49 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
3 search index=cs_index1	10min 30.45s	04/10/2019 23:27:10 -0700	Sat Mar 23 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
4 search index=cs_index1	10min 16.46s	04/10/2019 23:12:11 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
5 search index=cs_index1	6min 54.34s	04/10/2019 23:03:49 -0700	Sun Mar 24 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
6 search index=cs_index1	11min 52.51s	04/10/2019 22:50:31 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
7 search index=cs_index1	3min 24.25s	04/10/2019 22:46:36 -0700	Mon Mar 25 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
8 search index=cs_index1	13min 30.52s	04/10/2019 22:22:45 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin

### SmartStore Search: 50% Cached

Follow these steps:

1. Use the roll-hot-buckets command on all indexer nodes to manually force all hot data to roll to warm buckets.
2. Use the splunk \_internal call /services/admin/cacheman/\_evict command on all indexer nodes to evict all data from the cache. The execution of this command will ensure that any subsequent search will fetch data from SwiftStack S3 target and not read any data from cache.
3. Complete a search on the SmartStore index cs\_index1 with a past 2-day time range.
4. Subsequently complete a search on the SmartStore index 'cs\_index1' with a past 4-day time range.
5. Go to the DMC on the Splunk master node, click Search > Activity > Search Usage Statistics: instance. Record the Search Runtime for the above search.

Search	Search Runtime	Search Start	Earliest Time	Latest Time	Type	User
1 search index=cs_index1	7min 12.38s	04/10/2019 23:47:08 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
2 search index=cs_index1	8min 38.42s	04/10/2019 23:37:49 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
3 search index=cs_index1	10min 30.45s	04/10/2019 23:27:10 -0700	Sat Mar 23 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
4 search index=cs_index1	10min 16.46s	04/10/2019 23:12:11 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
5 search index=cs_index1	6min 54.34s	04/10/2019 23:03:49 -0700	Sun Mar 24 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
6 search index=cs_index1	11min 52.51s	04/10/2019 22:50:31 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
7 search index=cs_index1	3min 24.25s	04/10/2019 22:46:36 -0700	Mon Mar 25 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
8 search index=cs_index1	13min 30.52s	04/10/2019 22:22:45 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin

### SmartStore Search: 75% Cached

Follow these steps:

1. Use the roll-hot-buckets command on all indexer nodes to manually force all hot data to roll to warm buckets.

- Use the splunk \_internal call /services/admin/cacheman/\_evict command on all indexer nodes to evict all data from the cache. The execution of this command will ensure that any subsequent search will fetch data from SwiftStack S3 target and not read any data from cache.
- Complete a search on the SmartStore index cs\_index1 with a past 3-day time range.
- Subsequently complete a search on the SmartStore index cs\_index1 with a past 4-day time range.
- Go to the DMC on the Splunk master node, click Search > Activity > Search Usage Statistics: instance. Record the Search Runtime for the above search.

Search	Search Runtime	Search Start	Earliest Time	Latest Time	Type	User
1 search index*cs_index1	7min 12.38s	04/10/2019 23:47:08 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
2 search index*cs_index1	8min 38.42s	04/10/2019 23:37:49 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
3 search index*cs_index1	10min 30.45s	04/10/2019 23:27:10 -0700	Sat Mar 23 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
4 search index*cs_index1	10min 16.46s	04/10/2019 23:12:11 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
5 search index*cs_index1	6min 54.34s	04/10/2019 23:03:49 -0700	Sun Mar 24 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
6 search index*cs_index1	11min 52.51s	04/10/2019 22:50:31 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
7 search index*cs_index1	3min 24.25s	04/10/2019 22:46:36 -0700	Mon Mar 25 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
8 search index*cs_index1	13min 30.52s	04/10/2019 22:22:45 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin

### SmartStore Search: 100% Cached

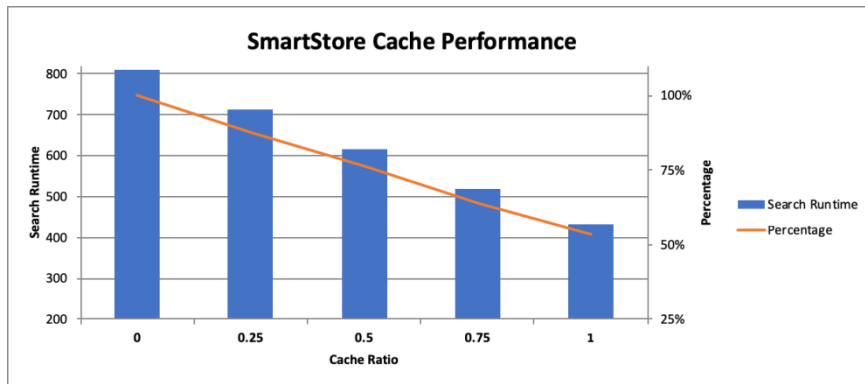
Follow these steps:

- Use the roll-hot-buckets command on all indexer nodes to manually force all hot data to roll to warm buckets.
- Use the splunk \_internal call /services/admin/cacheman/\_evict command on all indexer nodes to evict all data from the cache. The execution of this command will ensure that any subsequent search will fetch data from SwiftStack S3 target and not read any data from cache.
- Complete a search on the SmartStore index cs\_index1 with a past 4-day time range.
- Subsequently complete a search on the SmartStore index cs\_index1 with a past 4-day time range.
- Go to the DMC on the Splunk master node, click Search > Activity > Search Usage Statistics: instance. Record the Search Runtime for the above search.

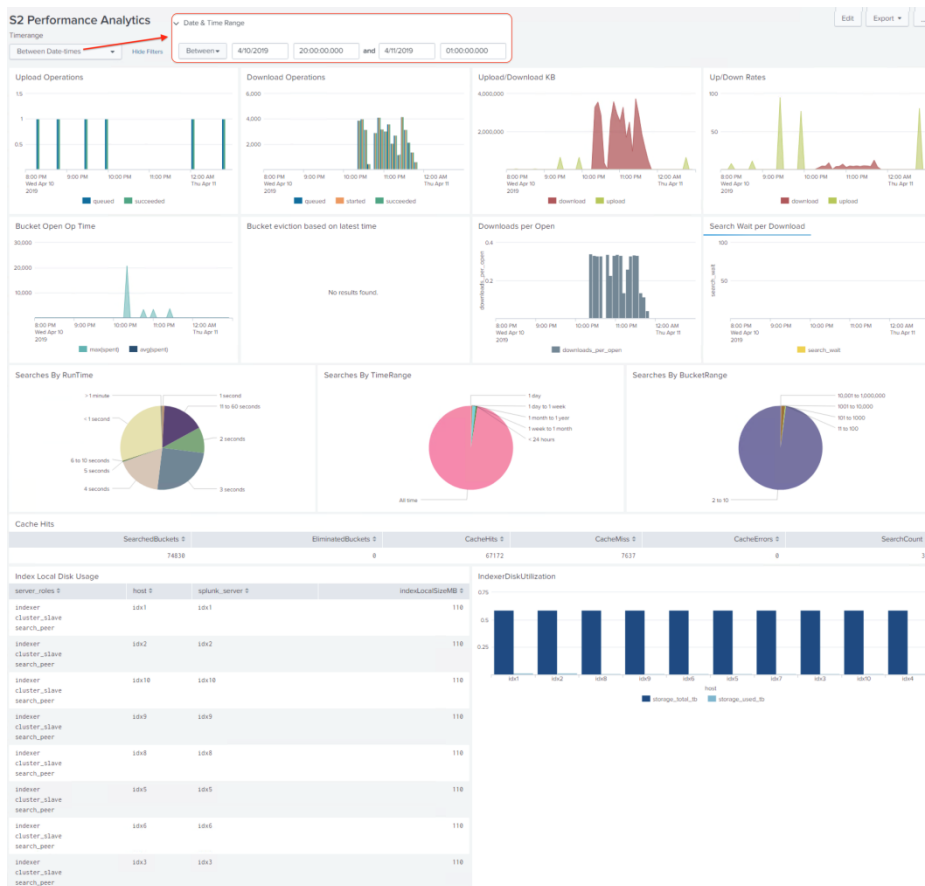
Search	Search Runtime	Search Start	Earliest Time	Latest Time	Type	User
1 search index*cs_index1	7min 12.38s	04/10/2019 23:47:08 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
2 search index*cs_index1	8min 38.42s	04/10/2019 23:37:49 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
3 search index*cs_index1	10min 30.45s	04/10/2019 23:27:10 -0700	Sat Mar 23 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
4 search index*cs_index1	10min 16.46s	04/10/2019 23:12:11 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
5 search index*cs_index1	6min 54.34s	04/10/2019 23:03:49 -0700	Sun Mar 24 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
6 search index*cs_index1	11min 52.51s	04/10/2019 22:50:31 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
7 search index*cs_index1	3min 24.25s	04/10/2019 22:46:36 -0700	Mon Mar 25 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin
8 search index*cs_index1	13min 30.52s	04/10/2019 22:22:45 -0700	Fri Mar 22 11:00:00 2019	Tue Mar 26 11:00:00 2019	ad hoc	admin

### Summary of the Cache Testing Results

The following chart shows the improvement on the search runtime with different cache ratios.



Monitor the S2 performance using the S2 Performance Analytics application on the master node. Pay attention to the Upload/Download operations, Upload/Download rates, and the cache hits.



## Bill of Materials

This section provides the BOM of the Cisco devices that are ordered to build the test bed to validate this solution.

Line Number	Part Number	Description	Qty	Note
1.0	UCSC-C220-M5SX	UCS C220 M5 SFF 10 HD w/o CPU, mem, HD, PCIe, PSU	6	HX Computing-only Nodes

1.0.1	CON-OSP-C220M5SX	SNTC 24X7X4OS UCS C220 M5 SFF 10 HD w/o CPU, mem, HD, PCIe,	6	
1.1	UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	72	
1.2	UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	6	
1.3	UCS-M2-240GB	240GB SATA M.2	6	
1.4	CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	6	
1.5	UCSC-PSU1-770W	Cisco UCS 770W AC Power Supply for Rack Server	12	
1.6	CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	12	
1.7	UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 & C240 M4 & M5 rack servers	6	
1.8	UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	60	
1.9	UCS-MSTOR-M2	Mini Storage carrier for M.2 SATA/NVME (holds up to 2)	6	
1.1	CBL-SC-MR12GM52	Super Cap cable for UCSC-RAID-M5 on C240 M5 Servers	6	
1.11	UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	6	
1.12	UCS-SID-INFR-OI	Other Infrastructure	6	
1.13	UCS-SID-WKL-SOS	Scale Out Storage (Scality, SwiftStack, COS, Cloudian only)	6	
1.14	UCS-SID-WKL-BD	Big Data and Analytics (Hadoop/IoT/ITOA)	6	
1.15	UCSC-RAID-M5	Cisco 12G Modular RAID controller with 2GB cache	6	
1.16	UCS-CPU-6132	2.6 GHz 6132/140W 14C/19.25MB Cache/DDR4 2666MHz	12	
1.17	UCS-MSD-32G	32GB Micro SD Card for UCS M5 servers	6	
2.0	HX-D-FI6332	HX SP Hyperflex System 6332 FI	2	UCS 6332 Fabric interconnects
2.0.1	CON-SNT-XDFI6332	SNTC-8X5XNBD HX SP Hyperflex System 6332 FI	2	
2.1	N10-MGT015-HX	UCS Manager v3.2(1) for HyperFlex	2	
2.2	CVR-QSFP-SFP10G	QSFP to SFP10G adapter	4	

2.3	UCS-PSU-6332-AC	UCS 6332 Power Supply/100-240VAC	4	
2.4	UCS-ACC-6332	UCS 6332/ 6454 Chassis Accessory Kit	2	
2.5	UCS-FAN-6332	UCS 6332/ 6454 Fan Module	8	
2.6	CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	4	
3.0	HXAF-SP-240M5SX-P2	SP HXAF240c Hyperflex System w/2x6148,12x32Gmem	4	HyperFlex Nodes
3.0.1	CON-SNT-HXAFSPP2	SNTC-8X5XNBD SP HXAF240c Hyperflex System w/2x6148	4	
3.1	HX-CPU-6148	2.4 GHz 6148/150W 20C/27.50MB Cache/DDR4 2666MHz	8	
3.2	HX-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	48	
3.3	HX-PCI-1-C240M5	Riser 1 incl 3 PCIe slots (x8, x16, x8)	4	
3.4	HX-PCI-2B-240M5	Riser 2B incl 3PCieslots(x8,x16,x8)+2NVMe(1cnctr) supports GPU	4	
3.5	HX-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	4	
3.6	HX-M2-240GB	240GB SATA M.2	4	
3.7	HX-MSD-32G	32GB Micro SD Card for UCS M5 servers	4	
3.8	HX-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	8	
3.9	HX-RAILB-M4	Ball Bearing Rail Kit for C220 M4 and C240 M4 rack servers	4	
3.1	UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs & below	8	
3.11	UCS-MSTOR-M2	Mini Storage carrier for M.2 SATA/NVME (holds up to 2)	4	
3.12	HXAF240C-BZL-M5SX	HXAF240C M5 Security Bezel	4	
3.13	UCSC-RSAS-240M5X	C240 Rear UCS-RAID-M5HD SAS cbl(1) kit incl fan, bkpln	4	
3.14	HX-SAS-M5HD	Cisco 12G Modular SAS HBA for up to 26 drives	4	
3.15	HX-STD-08	HX Standard w/1x400GB SAS, 1x240GB SATA, 11x960GB SATA	4	
3.16	HX-SD240G61X-EV	240GB 2.5 inch Enterprise Value 6G SATA SSD	4	



3.17	HX-NVMEHW-H1600	1.6TB 2.5in U.2 HGST SN200 NVMe High Perf. High Endurance	4	
3.18	HX-SD960G61X-EV	960GB 2.5 inch Enterprise Value 6G SATA SSD	92	
3.19	HX-VSP-6-5-FND-D	Factory Installed -vSphere SW 6.5 End-user to provide License	4	
3.2	HX-VSP-6-5-FND-DL	Factory Installed - VMware vSphere 6.5 SW Download	4	
3.21	CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	8	
4.0	UCSS-S3260	Cisco UCS S3260 Storage Server Base Chassis	3	S3260 Dual-server Chassis
4.0.1	CON-OSP-UCSS3260	SNTC 24X7X4OS, Cisco UCS S3260 Storage Server Base Chassis	3	
4.1	UCSC-PSU1-1050W	Cisco UCS 1050W AC Power Supply for Rack Server	12	
4.2	CAB-N5K6A-NA	Power Cord, 200/240V 6A North America	12	
4.3	CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	3	
4.4	UCSC-C3X60-RAIL	UCS C3X60 Rack Rails Kit	3	
4.5	UCSS-S3260-BBEZEL	Cisco UCS S3260 Bezel	3	
4.6	N20-BBLKD-7MM	UCS 7MM SSD Blank Filler	12	
4.7	N20-BKVM	KVM local IO cable for UCS servers console port	6	
4.8	UCS-S3260-M5SRB	UCS S3260 M5 Server Node for Intel Scalable CPUs	3	
4.9	UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	24	
4.1	UCS-S3260-DRAID	UCS S3260 Dual Raid based on LSI 3316	3	
4.11	UCS-S3260-NVG25	UCS S3260 M5/SIOC 500G NVMe (no Sled)	6	
4.12	UCS-S3260-M5HS	UCS S3260 M5 Server Node Heat Sink	6	
4.13	UCS-S3260-NVMSLD1	UCS S3260 M5 Svr Node NVMe Sled	3	
4.14	UCS-S3260-M5SRB	UCS S3260 M5 Server Node for Intel Scalable CPUs	3	
4.15	UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	24	
4.16	UCS-S3260-DRAID	UCS S3260 Dual Raid based on LSI 3316	3	

4.17	UCS-S3260-NVG25	UCS S3260 M5/SIOC 500G NVMe (no Sled)	6	
4.18	UCS-S3260-M5HS	UCS S3260 M5 Server Node Heat Sink	6	
4.19	UCS-S3260-NVMSLD1	UCS S3260 M5 Svr Node NVMe Sled	3	
4.2	UCS-S3260-56HD12	UCS S3260 4rows of drives 56x 12TB Total: 672TB	3	
4.21	UCS-S3260-HD12T	UCS S3260 12TB NL-SAS 7200 RPM 12Gb HDD w Carrier- Top Load	168	
4.22	UCS-C3X60-12G240	UCSC C3X60 400GB 12Gbps SSD (Gen 2)	6	
4.23	UCS-CPU-6132	2.6 GHz 6132/140W 14C/19.25MB Cache/DDR4 2666MHz	6	
4.24	UCS-CPU-6132	2.6 GHz 6132/140W 14C/19.25MB Cache/DDR4 2666MHz	6	
4.25	UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	3	
4.26	UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	3	
5.0	UCSC-C220-M5SX	UCS C220 M5 SFF 10 HD w/o CPU, mem, HD, PCIe, PSU	2	SwiftStack Controllers
5.0.1	CON-OSP-C220M5SX	SNTC 24X7X4OS UCS C220 M5 SFF 10 HD w/o CPU, mem, HD, PCIe,	2	
5.1	UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	24	
5.2	UCS-SD16TBKS4-EV	1.6TB 2.5 inch Enterprise Value 6G SATA SSD	4	
5.3	UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	2	
5.4	UCS-M2-240GB	240GB SATA M.2	4	
5.5	CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	2	
5.6	UCSC-PSU1-770W	Cisco UCS 770W AC Power Supply for Rack Server	4	
5.7	CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	4	
5.8	UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 & C240 M4 & M5 rack servers	2	
5.9	UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	12	

5.1	UCS-MSTOR-M2	Mini Storage carrier for M.2 SATA/NVME (holds up to 2)	2	
5.11	CBL-SC-MR12GM52	Super Cap cable for UCSC-RAID-M5 on C240 M5 Servers	2	
5.12	UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	2	
5.13	UCS-SID-INFR-OI	Other Infrastructure	2	
5.14	UCS-SID-WKL-SOS	Scale Out Storage (Scality, SwiftStack, COS, Cloudian only)	2	
5.15	UCS-SID-WKL-BD	Big Data and Analytics (Hadoop/IoT/ITOA)	2	
5.16	UCSC-RAID-M5	Cisco 12G Modular RAID controller with 2GB cache	2	
5.17	UCS-CPU-6132	2.6 GHz 6132/140W 14C/19.25MB Cache/DDR4 2666MHz	4	
6.0	SWIFTSTACK-LICENSE	SolutionsPlus: SwiftStack Subscription SW Licenses	1	SwiftStack License
6.0.1	SSTACK-MC	SwiftStack Multi-Cloud Subscription (Single-Region)	1	

## Summary

---

Machine data offers a trove of insights, leading to organizational success and efficiency — but mining that data can be complicated without the right data analytics platform. Splunk Enterprise software delivers best in class operational visibility and digital intelligence by monitoring all machine generated data and making it accessible, usable and valuable across the organization. Splunk deployments typically start small and expand rapidly to address additional use cases. Cisco HyperFlex systems provide optimized hyperconverged infrastructure for any workload at any scale. It simplifies and accelerates the deployment process, hosts the data with consistency and resiliency, and can be easily scaled out with the growth of the need. Along with SwiftStack scalable object storage systems, this infrastructure provides an excellent choice of hardware and storage for the high-performing virtual infrastructure required for Splunk deployment in VMware ESXi-based virtualized environment.

The configuration detailed in this document can be extended to clusters of various sizes depending on what application demands. Together Splunk Enterprise, Cisco HyperFlex platform, and SwiftStack object storage system on Cisco UCS S3260 help the customers more quickly build and maintain a next-generation data center infrastructure that can scale quickly to meet the need of data growth and provide smarter business outcomes. Combining these three vendors' technologies, expertise and experience in the field, we are able to provide an enterprise ready hardware and software choices for a scale-out data center analytics solution that is simple to install, scalable, and performant.

## For More Information

For additional information, see the following:

Cisco HyperFlex products, services, and solutions: <https://www.cisco.com/go/hyperflex>

Cisco Big Data products, services, and solutions: <https://www.cisco.com/go/bigdata>

Cisco Software Defined Storage products, services, and solutions: <https://www.cisco.com/go/sds>

CVD: [Cisco HyperFlex 3.0 for Virtual Server Infrastructure with VMware ESXi](#)

CVD: [Cisco UCS Integrated Infrastructure for Big Data and Analytics with Splunk Enterprise](#)

CVD: [Cisco UCS S3260 Storage Server with SwiftStack Software Defined Object Storage](#)

Splunk Tech Brief: [Deploying Splunk Enterprise Inside Virtual Environments](#)

## Appendix

### Appendix A: HyperFlex Cluster Capacity Calculations

A HyperFlex HX Data Platform cluster capacity is calculated as follows:

$$\frac{((\text{capacity disk size in GB} \times 10^9) / 1024^3) \times \text{number of capacity disks per node} \times \text{number of HyperFlex nodes} \times 0.92}{\text{replication factor}}$$

Divide the result by 1024 to get a value in TiB

The replication factor value is 3 if the HX cluster is set to RF=3, and the value is 2 if the HX cluster is set to RF=2.

The 0.92 multiplier accounts for an 8% reservation set aside on each disk by the HX Data Platform software for various internal filesystem functions.

Calculation example:

<capacity disk size in GB> = 1200 for 1.2 TB disks

<number of capacity disks per node> = 15 for an HX240c-M4SX model server

<number of HyperFlex nodes> = 8

replication factor = 3

Result:  $\frac{((1200 \times 10^9) / 1024^3) \times 15 \times 8 \times 0.92}{3} = 41127.2049$

$41127.2049 / 1024 = 40.16$  TiB

### Appendix B: PowerShell Script Example – Clone Splunk Virtual Machines

```
# splunkvm.ps1
# Description: ReadyClone Splunk VMs in vCenter
# Usage: Modify the variables to specify the vCenter server address, user, password;
# HX Controller VM address, root password; Guest VM root password.
#
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore -Confirm:$false | Out-Null
Add-PSSnapin 'VMware.VimAutomation.Core'

##### Change Test Parameters Here #####
$VC='192.168.66.51'           # vcenter info
$user='dmzhx\huich'         # vcenter username
$password= 'Cisco123'       # vcenter password
$guestuser='root'          # guest username
$guestpwd= 'Cisco123'       # guest password
$hxCtrl='192.168.66.32'     # HX controller info
$hxCtrlUser='root'          # HX controller username
$hxCtrlPwd= 'C!sco12345'    # HX controller password

##### Connect to vCenter #####
Connect-VIServer -Server $VC -user $user -password $password # Connect to vCenter

##### Clone template VMs #####
$expression1="C:\putty\plink $hxCtrl -l $hxCtrlUser -pw $hxCtrlPwd echo test"
```

```

write-host $expression1 -foreground green
invoke-expression $expression1

$clone1="admin"          #Clone Splunk admin VMs
$num1=6                 #Splunk admin VM numbers
$expression="C:\putty\plink $hxCtrl -l $hxuser -pw $hxpwd stcli vm clone --name spktmp-1
--clone $clone1 --startnumber 1 --number $num1 --poweron"
write-host $expression -foreground green
invoke-expression $expression
sleep 5

$clone2="sh"           #Clone Splunk Search Head (SH) VMs
$num2=4               #Splunk SH VM numbers
$expression="C:\putty\plink $hxCtrl -l $hxuser -pw $hxpwd stcli vm clone --name spktmp-2
--clone $clone2 --startnumber 1 --number $num2 --poweron"
write-host $expression -foreground green
invoke-expression $expression
sleep 5

$clone3="idx"         #Clone Splunk Indexer VMs
$num3=10             #Splunk Indexer VM numbers
$expression="C:\putty\plink $hxCtrl -l $hxuser -pw $hxpwd stcli vm clone --name spktmp-3
--clone $clone3 --startnumber 1 --number $num3 --poweron"
write-host $expression -foreground green
invoke-expression $expression

##### Wait 1 minute for VMs to power on #####
sleep 60

##### Assign IP addresses to VMs #####
##### Configure Admin VMs
$adm=(1..$num1)

$ip=52              #starting IP for Admin VMs
foreach($i in $adm ) {

$vmip1='192.168.11.'+$ip+'/24'
Write-Host "vmip1 $vmip1" -ForegroundColor Green

$newVM = "admin"+"$i"
Write-Host "newVM is $newVM" -ForegroundColor Green

Invoke-VMScript -VM $newVM -ScriptText "nmcli con add type ethernet con-name eth0 ifname
ens192 ip4 $vmip1 autoconnect yes" -guestuser $guestuser -guestpassword $guestpwd
sleep 15
Invoke-VMScript -VM $newVM -ScriptText "nmcli general hostname $newVM" -guestuser
$guestuser -guestpassword $guestpwd

$ip=$ip+1

}

##### Configure SH VMs
$sha=(1..$num2)

$ip=58              #starting IP for SH VMs

```

```

foreach($i in $sha ) {

$vmip1='192.168.11.'+$ip+'/24'
Write-Host "vmip1 $vmip1" -ForegroundColor Green

$newVM = "sh"+"$i"
Write-Host "newVM is $newVM" -ForegroundColor Green

Invoke-VMScript -VM $newVM -ScriptText "nmcli con add type ethernet con-name eth0 ifname
ens192 ip4 $vmip1 autoconnect yes" -guestuser $guestuser -guestpassword $guestpwd
sleep 15
Invoke-VMScript -VM $newVM -ScriptText "nmcli general hostname $newVM" -guestuser
$guestuser -guestpassword $guestpwd

$ip=$ip+1

}

##### Configure Indexer VMs
$idx1=(1..$num3)

$ip=62      #starting IP for Indexer VMs
foreach($i in $idx1 ) {

$vmip1='192.168.11.'+$ip+'/24'
Write-Host "vmip1 $vmip1" -ForegroundColor Green

$newVM = "idx"+"$i"
Write-Host "newVM is $newVM" -ForegroundColor Green

Invoke-VMScript -VM $newVM -ScriptText "nmcli con add type ethernet con-name eth0 ifname
ens192 ip4 $vmip1 autoconnect yes" -guestuser $guestuser -guestpassword $guestpwd
sleep 15
Invoke-VMScript -VM $newVM -ScriptText "nmcli general hostname $newVM" -guestuser
$guestuser -guestpassword $guestpwd

$ip=$ip+1
}

# sleep 30 seconds, then disconnect.
sleep 30

Disconnect-VIServer -Server $VC -Confirm:$false

```

## Appendix C: SmartStore indexes.conf File Example

```

# Version 7.2.3
# DO NOT EDIT THIS FILE!
# Please make all changes to files in $SPLUNK_HOME/etc/master-apps/_cluster/local.
# To make changes, copy the section/stanza you want to change from
#$SPLUNK_HOME/etc/system/default
# into ../local and edit there.
#
# This file configures Splunk's indexes and their properties for a cluster.
#

```

```
#####
# index definitions
#####

[main]
repFactor = auto

[history]
repFactor = auto

[summary]
repFactor = auto

[_internal]
repFactor = auto

[_audit]
repFactor = auto

[_thefishbucket]
repFactor = auto

[_telemetry]
homePath    = $SPLUNK_DB/_telemetry/db
coldPath    = $SPLUNK_DB/_telemetry/colddb
thawedPath  = $SPLUNK_DB/_telemetry/thaweddb
repFactor   = auto

# this index has been removed in the 4.1 series, but this stanza must be
# preserved to avoid displaying errors for users that have tweaked the index's
# size/etc parameters in local/indexes.conf.
#
[splunklogger]
repFactor = auto

[_introspection]
repFactor = auto

[default]
remotePath = volume:s3/$_index_name
repFactor=auto

[volume:s3]
storageType = remote
path = s3://splunk-ss
remote.s3.access_key = splunk_user
remote.s3.secret_key = 892c69d69c303c87ddd40b31e28a737a
remote.s3.endpoint = http://swift-cluster.dmzhx.lab.cisco.com
remote.s3.auth_region = us-east-1

[cs_indexc]
homePath = $SPLUNK_DB/cs_indexc/db
coldPath=$SPLUNK_DB/cs_indexc/colddb
thawedPath=$SPLUNK_DB/cs_indexc/thaweddb
maxDataSize = auto
```



```

remotePath=volume:s3/$_index_name

[cs_index1]
homePath = $SPLUNK_DB/cs_index1/db
coldPath=$SPLUNK_DB/cs_index1/colddb
thawedPath=$SPLUNK_DB/cs_index1/thaweddb
maxDataSize = auto
remotePath=volume:s3/$_index_name

```

## Appendix D: Custom Event Generation Script

```

#
# This script file generates the logging events and sends them to the Splunk indexers.
# It is added into the job scheduler utility cron table on the Linux servers so that it
runs
# continuously to reach the target daily indexing rate.
# [root@load1-2 ~]# crontab -l
# 0,30 * * * * /var/tmp/logger.generator
# 0,30 * * * * /var/tmp/logger.generator.1
# 0,30 * * * * /var/tmp/logger.generator.2
# 0 0 * * * /usr/bin/pkill logger.generator
# 0 0 * * * /usr/bin/pkill logger.generator.1
# 0 0 * * * /usr/bin/pkill logger.generator.2
#
#####
# /var/tmp/logger.generator
#####
#
#!/bin/bash
# set n to 1
count=1

while [ $count -le 900000]
do
logger -n 192.168.11.63 -p 5447 "$(hostname) Need this to create additional log data for
the Splunk environment. This is a filler for the splunk logs $(date) Need this to create
additional log data for the Splunk environment. This is a filler for the splunk logs
$(date) Need this to create additional log data for the Splunk environment. This is a
filler for the splunk logs $(date) Need this to create additional log data for the Splunk
environment. This is a filler for the splunk logs $(date) Need this to create additional
log data for the Splunk environment. This is a filler for the splunk logs $(date) Need
this to create additional log data for the Splunk environment. This is a filler for the
splunk logs $(date) Need this to create additional log data for the Splunk
environment. Need this to create additional log data for the Splunk environment. This is
a filler for the splunk logs $(date) Need this to create additional log data for the
Splunk environment. Need this to create additional log data for the Splunk environment.
This is a filler for the splunk logs $(date) Need this to create additional log data for
the Splunk environment. Need this to create additional log data for the Splunk
environment. This is a filler for the splunk logs $(date) Need this to create additional
log data for the Splunk environment. Need this to create additional log data for the
Splunk environment. This is a filler for the splunk logs $(date) Need this to create
additional log data for the Splunk environment. Need this to create additional log data
for the Splunk environment. This is a filler for the splunk logs $(date) Need this to

```

```
create additional log data for the Splunk environment. Need this to create additional log
data for the Splunk environment. This is a filler for the splunk logs $(date) Need this
to create additional log data for the Splunk environment. Need this to create additional
log data for the Splunk environment. This is a filler for the splunk logs $(date) Need
this to create additional log data for the Splunk environment."
count=$(( count+1 ))      # increments $n
done
```

## About the Authors

---

**Hui Chen, Technical Marketing Engineer, Cisco UCS Data Center Engineering Group, Cisco Systems, Inc.**

Hui is a network and storage veteran with over 15 years of experience on the unified computing, Fibre Channel-based storage area networking, the LAN/SAN convergence systems; and how to build end-to-end, from the application, server, networking to storage, solutions in the data center. Currently he focuses on Cisco's Software Defined Storage (SDS) and Hyperconverged Infrastructure (HCI) solutions. Hui is also a seasoned CCIE.

## Acknowledgements

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

- Karthik Karupasamy, Technical Marketing Engineer, Cisco Systems Inc.
- Anup Pal, Solution Engineer, SwiftStack Inc.
- Brian Wooden, Director Partner Integrations, Splunk Inc.