

Cisco UCS Petabyte-Scale Solution for Splunk Enterprise

Powered by Cisco UCS S-Series Storage Servers

Last Updated: October 28, 2016



About Cisco Validated Designs

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

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

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

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

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, 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)

© 2016 Cisco Systems, Inc. All rights reserved.

Table of Contents

About Cisco Validated Designs	2
Executive Summary	8
Solution Overview.....	9
Audience.....	9
Solution Summary.....	9
Technology Overview	10
Cisco UCS 6300 Series Fabric Interconnects.....	10
Cisco UCS C220 Rack Mount Servers	10
Cisco UCS S3260 Storage Servers.....	11
Cisco UCS VIC 1387.....	12
Cisco UCS Manager.....	12
Solution Design.....	14
Splunk for Big Data Analytics	14
Key Features of Splunk Enterprise.....	14
Deployment Hardware and Software	15
Architecture.....	15
Rack and PDU Configuration	15
Port Configuration on Fabric Interconnects.....	16
Server Configuration and Cabling for Cisco UCS S3260 Storage Server.....	17
Configuration and Cabling for Cisco UCS C220 M4 Rack Servers	17
Rack Appearance	18
Software Distributions and Versions	19
Fabric Configuration.....	20
Performing Initial Setup of Cisco UCS 6332 Fabric Interconnects.....	20
Configure Fabric Interconnect A.....	20
Configure Fabric Interconnect B.....	21
Logging Into Cisco UCS Manager.....	22
Upgrading UCSM Software to Version 3.1(2b).....	23
Adding a Block of IP Addresses for KVM Access.....	23
Enabling Uplink Ports	24
Configuring VLANs	25
Enabling Server Ports.....	27
Creating an Organization.....	28
Creating a Storage Profile for Cisco UCS S3260 Boot Drives	29

Creating a Storage Profile for Cisco UCS C220 Boot Drives	33
Creating Chassis Profiles for Cisco S3260 Storage Servers	35
Creating Disk Zoning Policy	35
Creating Chassis Firmware Package Policy.....	39
Creating a Chassis Profile Template	40
Creating Chassis Profiles from Template	43
Associating Chassis Profiles to Individual Chassis.....	45
Creating Pools for Service Profile Templates	47
Creating MAC Address Pools.....	47
Creating Server Pools	50
Creating Policies for Service Profile Templates.....	53
Creating Host Firmware Package Policy	53
Creating QoS Policies	54
Platinum Policy	54
Setting Jumbo Frames.....	55
Creating Local Disk Configuration Policy.....	56
Creating Server BIOS Policy.....	57
Creating Boot Policies.....	62
Creating Power Control Policy.....	65
Creating a Service Profile Template for Cisco S3260 Storage Servers	67
Configuring the Storage Provisioning for the Template.....	68
Configuring Network Settings for the Template.....	70
Configuring SAN Connectivity for the Template	75
Configuring vNIC/vHBA Placement Policy for the Template.....	76
Configuring Server Boot Order for the Template.....	76
Configuring Server Assignment for the Template.....	77
Configuring Operational Policies for the Template	78
Creating a Service Profile Template for Cisco C220 M4 Servers	79
Creating Service Profiles from Template.....	85
Identifying the Servers	87
Installing Red Hat Enterprise Linux 7.2 on all servers.....	89
Post OS Install Configuration.....	112
Creating Red Hat Enterprise Linux (RHEL) 7.2 Local Repo.....	112
Creating the Red Hat Repository Database	113
Configuring /etc/hosts	115
Setting Up Password-less Login	117

Setting Up ClusterShell	117
Installing httpd	120
Disabling the Linux Firewall	121
Disabling SELinux	121
Set Up all Nodes to Use the RHEL Repository	122
Upgrading the Cisco Network Driver for VIC1387	123
Installing xfsprogs	124
NTP Configuration	125
Enabling Syslog	127
Setting Ulimit	127
Set TCP Retries	129
Configure VM Swapping	129
Disable IPv6 Defaults	129
Disable Transparent Huge Pages	130
Installing the LSI StorCLI Utility on All Indexers	131
Configuring the Virtual Drive on the Indexers	132
Configuring the XFS File System	134
Cluster Verification	136
Installing Splunk Enterprise 6.5	141
Splunk Architecture and Terminology	141
Splunk Services and Processes	142
Planning the Installation	143
Installing Splunk	144
Setting Up Login for Splunk User	146
Starting the Splunk Enterprise Cluster	148
Logging in for the First Time	149
Creating User Accounts	150
Initializing Splunk on Boot	151
Default Ports	151
Configuring the Splunk Enterprise Cluster	152
Configuring Splunk Enterprise Licenses	152
Setting Up License Master	152
Configure the Indexers, Search Heads, and Admin Nodes as License Slaves	154
Configure all the License Slaves at Once Using CLI (Clush)	154
(Optional) Configure License Slaves Individually Using the Web Interface	155
Verifying License-Slave Relationships	156

Configuring the Master Node / Cluster Master	158
Configure Indexing Peers.....	161
Configuring Indexer Clusters.....	161
Configure All Indexing Peers Using CLI (Clush).....	162
Configure Indexing Peers Individually Using the Web Interface (Optional).....	163
Setting Dedicated Replication Address.....	168
Verify Cluster Configuration	169
Configure Receiving on the Peer Nodes	170
Configure Master to Forward All its Data to the Indexer Layer.....	172
Configure Search Head Clustering	173
Add Search Heads to Master Node	174
Configure the Deployer.....	176
Configure Search Head Cluster Members.....	177
Elect a Search Head Captain.....	178
Configure Search Heads to Forward their Data to the Indexer Layer.....	179
Configure Search Head Load-Balancing.....	181
Configuring the Distributed Management Console	184
Configure Search Heads in Distributed Management Console.....	187
Configuring Archive of Data from Cold to Frozen.....	192
Configuring the Deployment Server.....	193
Installing a Universal Forwarder on a Test Server.....	194
Register Universal Forwarder with the Deployment Server.....	194
Configure an App within the Deployment Server.....	194
Installation Verification	198
Verifying from DMC	198
Verifying Master and Peer Replication	200
Verifying Data Replication	203
Verifying Transfer of Cold to Frozen Buckets	207
Post-Test Cleanup.....	208
Removing Test Data.....	208
Removing Test Indexes.....	209
Removing the Universal Forwarder.....	210
Remove Deployment Server App.....	210
Hardening the Splunk Installation.....	210
Turn Off Web Servers	211
Best Practices for Onboarding Data.....	212

The Universal Forwarder	212
Advantages of the Splunk Universal Forwarder.....	212
What About Systems Where the Splunk Universal Forwarder is Not Supported?.....	213
Additional Terminology	215
Recommended Apps and Add-ons for Data Collection.....	215
Conclusion	217
Bill of Materials	218
About the Authors.....	226
Acknowledgements	226

Executive Summary

Traditional tools for managing and monitoring IT infrastructures are out of step with the constant change happening in today's data centers. When problems arise, finding the root cause or gaining visibility across the infrastructure to pro-actively identify and prevent outages is nearly impossible. Virtualization and cloud infrastructures introduce additional complexity, resulting in an environment that is more challenging to control and manage.

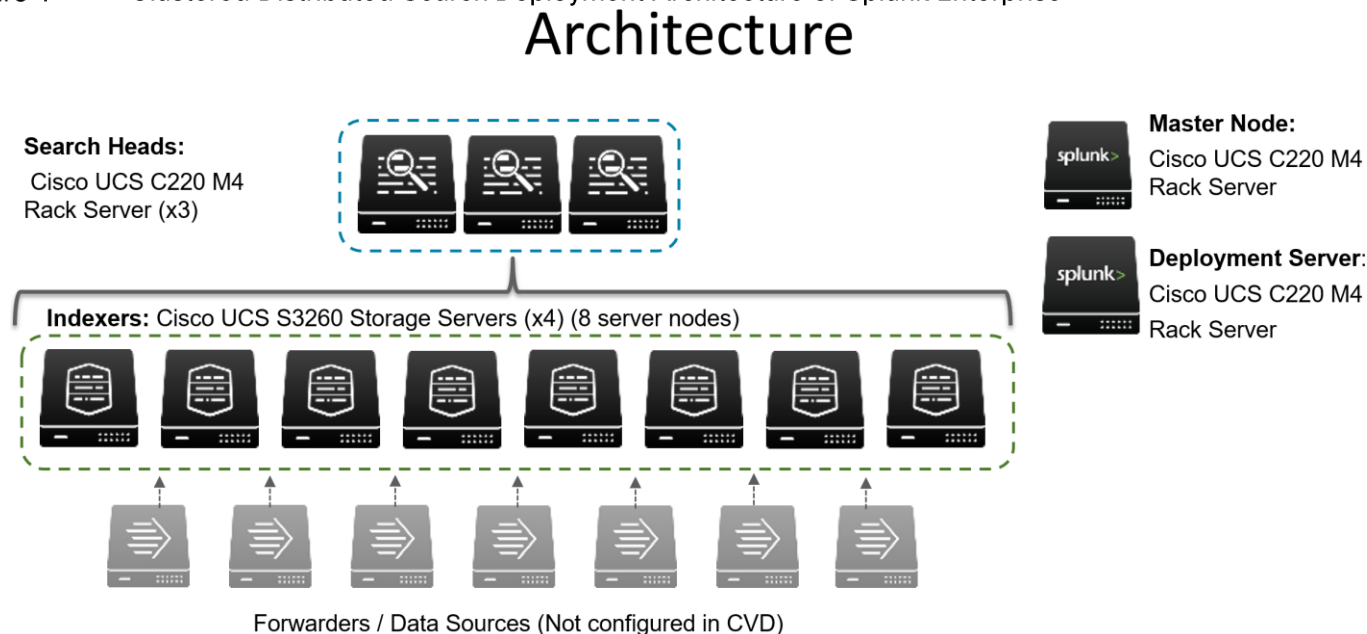
Splunk software reliably collects and indexes machine data, from a single source to tens of thousands of sources, all in real time. Organizations typically start with Splunk to solve a specific problem, and then expand from there to address a broad range of use cases, such as application troubleshooting, 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. Cisco UCS's ability to expand to thousands of servers allows the Splunk deployments to scale horizontally while continuously delivering exceptional performance.

The Cisco Validated Design (CVD) for Splunk Enterprise describes the architecture and deployment procedures for Splunk Enterprise on a Distributed High Capacity and Performance reference architecture based on Cisco UCS Integrated Infrastructure for Big Data (see Distributed Splunk Reference Architecture Solution Brief). The configuration consists of four (4) Cisco UCS S3260 Storage Servers as indexers, three (3) Cisco C220 M4 rack servers as search heads and two (2) Cisco C220 M4 rack servers to perform administrative functions.

Solution Overview

This CVD describes architecture and deployment procedures for Splunk Enterprise using four (4) Cisco UCS S3260 Storage Servers as indexers, three (3) Cisco UCS C220 M4 rack servers as search heads, and two (2) Cisco UCS C220 M4 rack servers to perform administrative functions. This architecture is based on the Cisco UCS Integrated Infrastructure for Big Data with Splunk. Figure 1 shows the architecture for the Splunk Enterprise deployment.

Figure 1 Clustered Distributed Search Deployment Architecture of Splunk Enterprise



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 on Cisco UCS Integrated Infrastructure for Big Data.

Solution Summary

This CVD offers a dependable deployment model for Splunk Enterprise which can be implemented rapidly and customized to meet Splunk requirements. The configuration detailed in the document can be extended to larger clusters. In this CVD, four Splunk Indexers provide capacity to index up to 2.4 TB of data per day. This configuration can scale to index hundreds of terabytes to petabytes of data every 24 hours, delivering real-time search results and meeting Splunk application demands with seamless data integration and analytics to multiple users across the globe.

Technology Overview

The Cisco UCS solution for Splunk Enterprise is based on Cisco UCS Integrated Infrastructure for Big Data and Analytics, a highly scalable architecture designed to meet a variety of scale-out application demands with seamless data integration and management integration capabilities built using the following components:

Cisco UCS 6300 Series Fabric Interconnects

Cisco UCS 6300 Series Fabric Interconnects provide high-bandwidth, low-latency connectivity for servers, with Cisco UCS Manager providing integrated, unified management for all connected devices. The Cisco UCS 6300 Series Fabric Interconnects are a core part of Cisco UCS, providing low-latency, lossless 40 Gigabit Ethernet, Fiber Channel over Ethernet (FCoE), and Fibre Channel functions with management capabilities for systems deployed in redundant pairs. Figure 2 shows the Cisco UCS 6332 UP-Port Fabric Interconnect.

Figure 2 Cisco UCS 6332 UP 32-Port Fabric Interconnect



Note: This Cisco Validated Design is built using third generation Fabric Interconnects (Cisco UCS 6332, as shown above in Figure 2), but the second generation Fabric Interconnects (Cisco UCS 6248 or Cisco UCS 6296) can be used as well. Use of second generation Fabric Interconnects will require 40GigE-to-10GigE converters.

Cisco UCS C220 Rack Mount Servers

Cisco UCS C220 M4 High-Density Rack servers (Small Form Factor Disk Drive Model) are enterprise-class systems that support a wide range of computing, I/O, and storage-capacity demands in compact designs. They are based on the Intel Xeon® E5-2600 v4 product family with a 12-Gbps SAS storage controller. The servers use 1.5-TB of main memory (128 or 256 GB is typical for big data applications) and a range of disk drive and SSD options. Cisco UCS virtual interface cards 1387 (VICs) designed for the M4 generation of Cisco UCS C-Series Rack Servers are optimized for high-bandwidth and low-latency cluster connectivity, with support for up to 256 virtual devices that are configured on demand through Cisco UCS Manager. Figure 3 shows the Cisco UCS C220 M4 Rack Server.

Figure 3 Cisco UCS C220 M4 Rack Server



Cisco UCS S3260 Storage Servers

The Cisco UCS S3260 Storage Server (0) is a high-density modular storage server designed to deliver efficient, industry-leading storage for data-intensive workloads. The Cisco UCS S3260 Storage Server is a modular chassis with dual server nodes (two servers per chassis) and up to 60 large-form-factor (LFF) drives in a 4RU form factor. The server uses dual Intel® Xeon® Processor E5-2600 v4 Series CPUs and supports up to 512 GB of main memory and a range of hard-disk-drive (HDD) and hybrid (solid-state-drive and hard-disk-drive) options. It comes with a 12Gbps-SAS RAID card with 4 GB cache and host bus adapter (HBA) controller, and up to two internal solid-state-disk (SSD) drives for booting.

Figure 4 Cisco UCS S3260 Storage Server



The Cisco UCS S3260 Storage Server chassis has 56 top-load LFF HDDs with a maximum capacity of 10 TB per HDD and can be mixed with up to 28 SSDs with a maximum capacity of 3.2 TB per SSD.

The modular Cisco UCS S3260 Storage Server chassis offers flexibility with more computing, storage, and PCIe expansion on the second slot in the chassis. This second slot can be used for:

- An additional server node
- Four additional LFF HDDs with up to 10 TB capacity per HDD
- New PCIe expansion tray with up to two x8 half-height, half-width PCIe slots that can use any industry-standard PCIe card including Fibre Channel and Ethernet cards.

The Cisco UCS S3260 Storage Server Chassis includes a Cisco UCS Virtual Interface Card (VIC) 1300 platform chip onboard the system I/O controller, offering high-performance bandwidth with dual-port 40 Gigabit Ethernet and FCoE interfaces per system I/O controller. Cisco UCS VIC 1387

Cisco UCS Virtual Interface Cards (VICs) are unique to Cisco. The Cisco UCS VIC 1387 incorporates next-generation converged network adapter (CNA) technology from Cisco, and offers dual 40-Gbps ports designed for use with Cisco UCS Rack-Mount Servers. Optimized for virtualized networking, this card delivers high performance and bandwidth utilization, and supports up to 256 virtual devices.

The Cisco UCS VIC 1387 (0) offers dual-port, Enhanced Quad, Small Form-Factor Pluggable (QSFP) 40 Gigabit Ethernet and Fiber Channel over Ethernet (FCoE), in a modular-LAN-on-motherboard (mLOM) form factor. The mLOM slot can be used to install a Cisco VIC without consuming a PCIe slot providing greater I/O expandability.

Figure 5 Cisco UCS VIC 1387

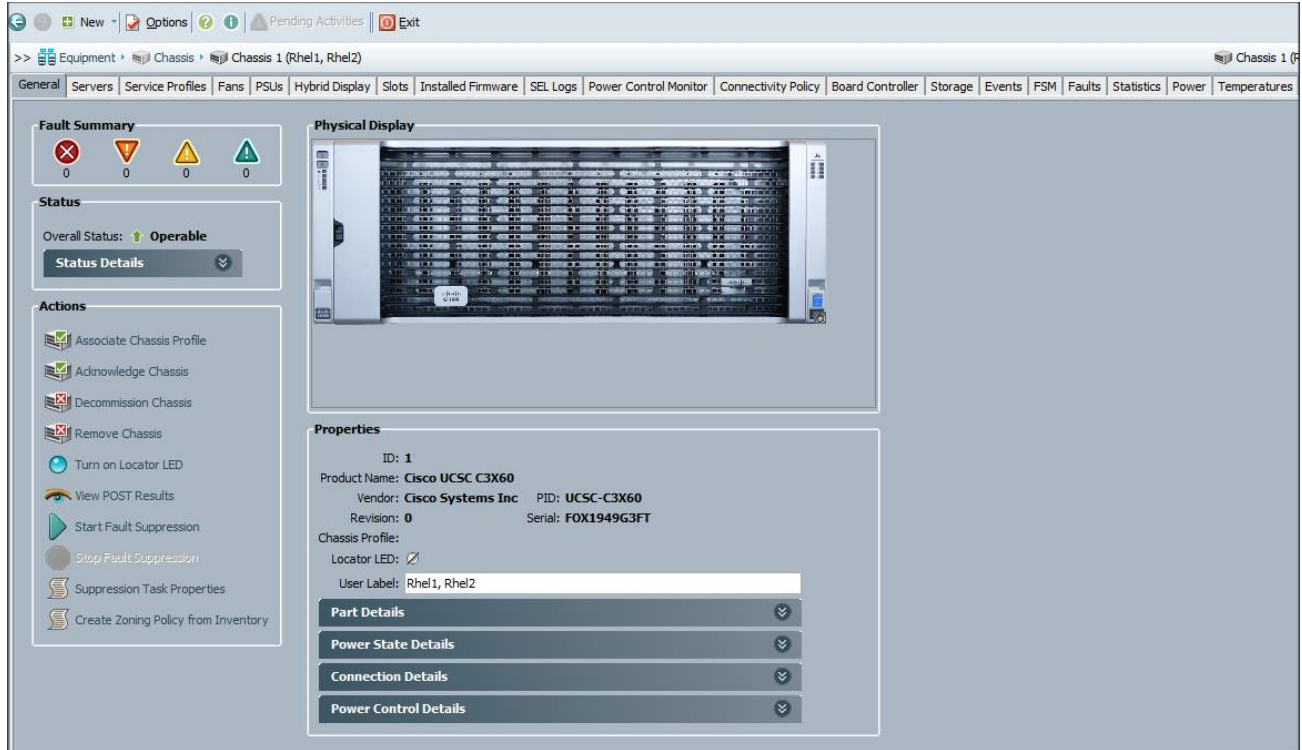


Cisco UCS Manager

Cisco UCS Manager resides within the Cisco UCS 6300 Series Fabric Interconnect. It makes the system self-aware and self-integrating, managing all of the system components as a single logical entity. Cisco UCS Manager can be accessed through an intuitive graphical user interface (GUI), as shown in 0, a command-line interface (CLI), or an XML application-programming interface (API). Cisco UCS Manager uses service profiles to define the personality, configuration, and connectivity of all resources within Cisco UCS, radically simplifying provisioning of resources so the process takes minutes instead of days. This simplification allows IT departments to shift their focus from constant maintenance to strategic business initiatives.

The new Cisco UCS Manager has smart capabilities such as predictive drive failure and rebuild. With the integration of Cisco UCS S3260 Storage Servers, Cisco UCS Manager can be configured to have hot spare drives in case of any drive failure. In such a case, Cisco UCS Manager will automatically detect the failed drives and replace it with one of the available hot spare drives, rebuild it, and make it available to use within the chassis.

Figure 6 Cisco UCS Manager



Solution Design

Splunk for Big Data Analytics

All your 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 is 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, across multi-geography, multi-datacenter, and hybrid cloud infrastructures.

Key Features of Splunk Enterprise

Splunk Enterprise provides an 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 horizontal scalability
- 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

Deployment Hardware and Software

Architecture

Reference Architecture for the Splunk Enterprise deployment is shown in Table 1

Table 1 Cisco UCS Reference Architecture for Splunk Enterprise (High Capacity and Performance)

Indexer	8 Cisco UCSC C3000 M4 Server Blades (in 4 S3260 chassis), each with: <ul style="list-style-type: none"> • 2 Intel Xeon® processor E5-2680 v4 CPUs (28 cores) • 256 GB of memory • Cisco 12-Gbps SAS modular RAID controller with 4-GB flash-backed write cache (FBWC) • Cisco UCS VIC 1387 (with two 40 Gigabit Ethernet QSFP ports) • Twenty 10-TB 7,200 rpm LFF SAS drives in a RAID10 configuration • Eight 1.6 TB solid state drives (SSDs)
Search head	3 Cisco UCS C220 M4 Rack Servers, each with: <ul style="list-style-type: none"> • 2 Intel Xeon® processor E5-2680 v4 CPUs (28 cores) • 256 GB of memory • Cisco 12-Gbps SAS modular RAID controller with 2-GB flash-backed write cache (FBWC) • Cisco UCS VIC 1387 • Two 600-GB 10K SFF SAS drives
Administration and master nodes	2 Cisco UCS C220 M4 Rack Servers, each with: <ul style="list-style-type: none"> • 2 Intel Xeon® processor E5-2620 v4 CPUs • 128 GB of memory • Cisco 12-Gbps SAS modular RAID controller with 2-GB flash-backed write cache (FBWC) • Cisco UCS VIC 1387 • Two 600-GB 10K SFF SAS drives
Networking	2 Cisco UCS 6332UP 32-Port Fabric Interconnects
Recommended indexing capacity	Up to 2 TB per day
Total storage capacity	888 TB
Rack space	23 RU

Rack and PDU Configuration

The rack consists of two vertical power distribution units (PDU), two Cisco UCS 6332UP Fabric Interconnects, four Cisco UCS S3260 Storage Servers, and five Cisco UCS C220 M4 servers. All the devices are connected to each of the vertical PDUs for redundancy, thereby ensuring availability during power source failure.

Table 2 describes the rack configuration used in this CVD.

Table 2 Rack Configurations

Position	Devices

42	Cisco UCS Fabric Interconnect 6332
41	Cisco UCS Fabric Interconnect 6332
40	Unused
39	Unused
38	Unused
37	Cisco UCS C220 M4
36	Cisco UCS C220 M4
35	Cisco UCS C220 M4
34	Cisco UCS C220 M4
33	Cisco UCS C220 M4
32	Unused
31	Unused
30	Unused
29	Unused
28	Unused
27	Unused
26	Unused
25	Unused
24	Unused
23	Unused
22	Unused
21	Unused
20	Unused
19	Unused
18	Unused
17	Unused
16	Cisco UCS S3260 Storage Server
15	
14	
13	
12	Cisco UCS S3260 Storage Server
11	
10	
9	
8	Cisco UCS S3260 Storage Server
7	
6	
5	
4	Cisco UCS S3260 Storage Server
3	
2	
1	

Port Configuration on Fabric Interconnects

Table 3 shows the network connectivity configurations used for developing this CVD.

Table 3 Port Types and Port Numbers

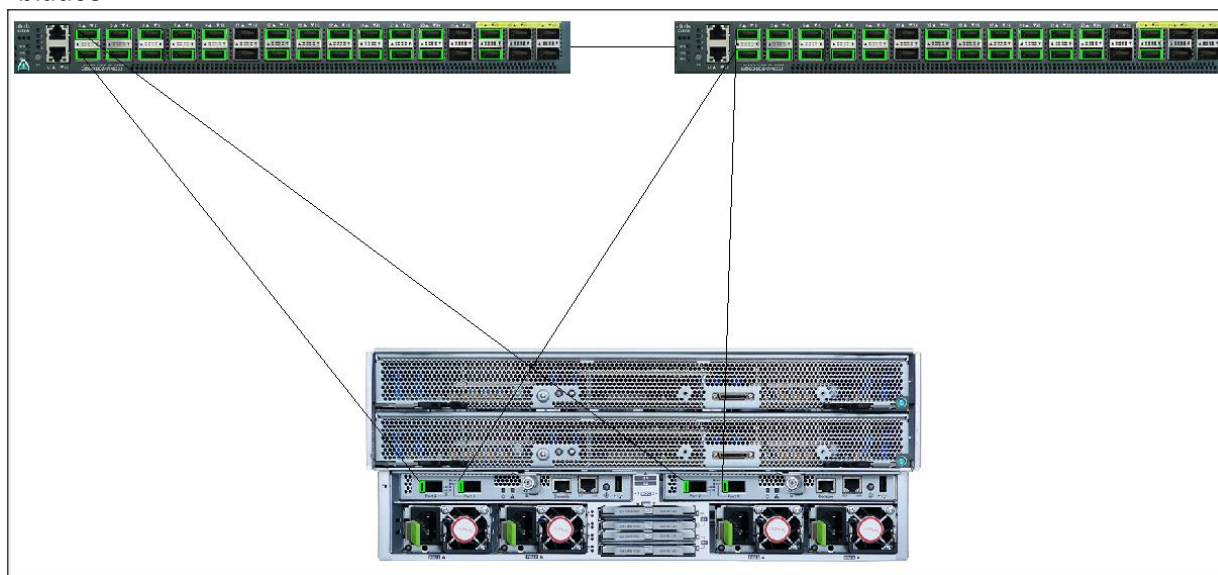
Port Type	Description	Port Number
Network	Uplink port	1
Server	Cisco UCS C220 M4 Servers	4 to 8
Server	Cisco UCS S3260 Servers	9 to 16

Server Configuration and Cabling for Cisco UCS S3260 Storage Server

The Cisco UCS S3260 Storage Server Chassis is equipped to fit up to two nodes and four 480 GB SATA SSD. Each server blade is equipped with Intel Xeon® E5-2680 v4 processors, 256 GB of memory, and a Cisco UCS C3X60 R4GB RAID controller.

Figure 13 illustrates the port connectivity between the Fabric Interconnect, and Cisco UCS S3260 Storage Server Chassis.

Figure 7 Fabric Topology for Cisco UCS S3260 Storage Server with UCSC C3000 M4 SRB server blades



For more information on physical connectivity illustrations and cluster setup, see:

http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c-series_integration/ucsm3-1/b_C-Series-Integration_UCSM3-1/b_C-Series-Integration_UCSM3-1_chapter_010.html

A similar connection will be used to connect five Cisco C220M4 rack mount servers as shown in the following section.

This solution makes use of 40 HDDs for cold storage and 16 SSDs for hot and warm storage. Each server in the Cisco S3260 will be assigned 20 HDDs and 8 SSDs. The HDDs should be installed in slots 1-40 and the SSDs should be installed in the last 16 slots: 41-56.

Configuration and Cabling for Cisco UCS C220 M4 Rack Servers

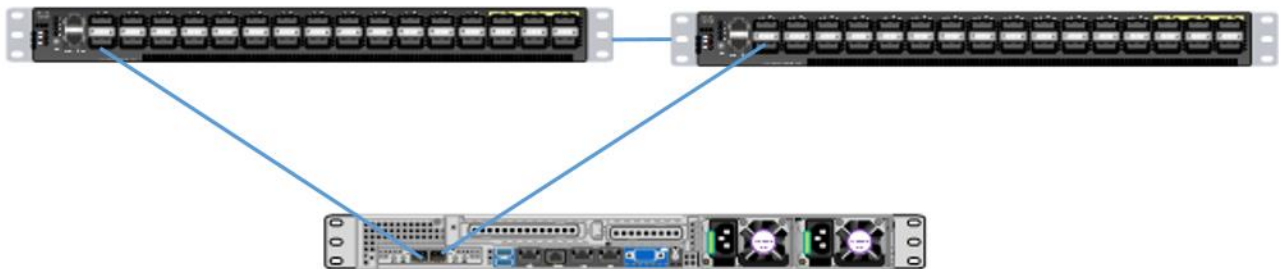
This solution makes use of five C220 M4 rack servers in two different configurations.

- The servers that function as the search heads are equipped with Intel Xeon® E5-2680 v4 processors, 256 GB of memory, Cisco UCS Virtual Interface Card 1387, Cisco 12-Gbps SAS Modular RAID controller with 2-GB FBWC, and two 600 GB 10K SFF SAS drives.
- The servers that function as the admin nodes are equipped with Intel Xeon® E5-2620 v4 processors, 128 GB of memory, Cisco UCS Virtual Interface Card 1387, Cisco 12-Gbps SAS Modular RAID controller with 2-GB FBWC, and two 600 GB 10K SFF SAS drives.

All five servers of this category are directly connected to the ports on the Cisco UCS 6332 Fabric Interconnects as shown below. These ports are configured as server ports in the UCS Manager.

Figure 14 illustrates the port connectivity between the Fabric Interconnects and Cisco UCS C220 M4 servers. Five Cisco UCS C220 M4 servers are used in the rack configuration.

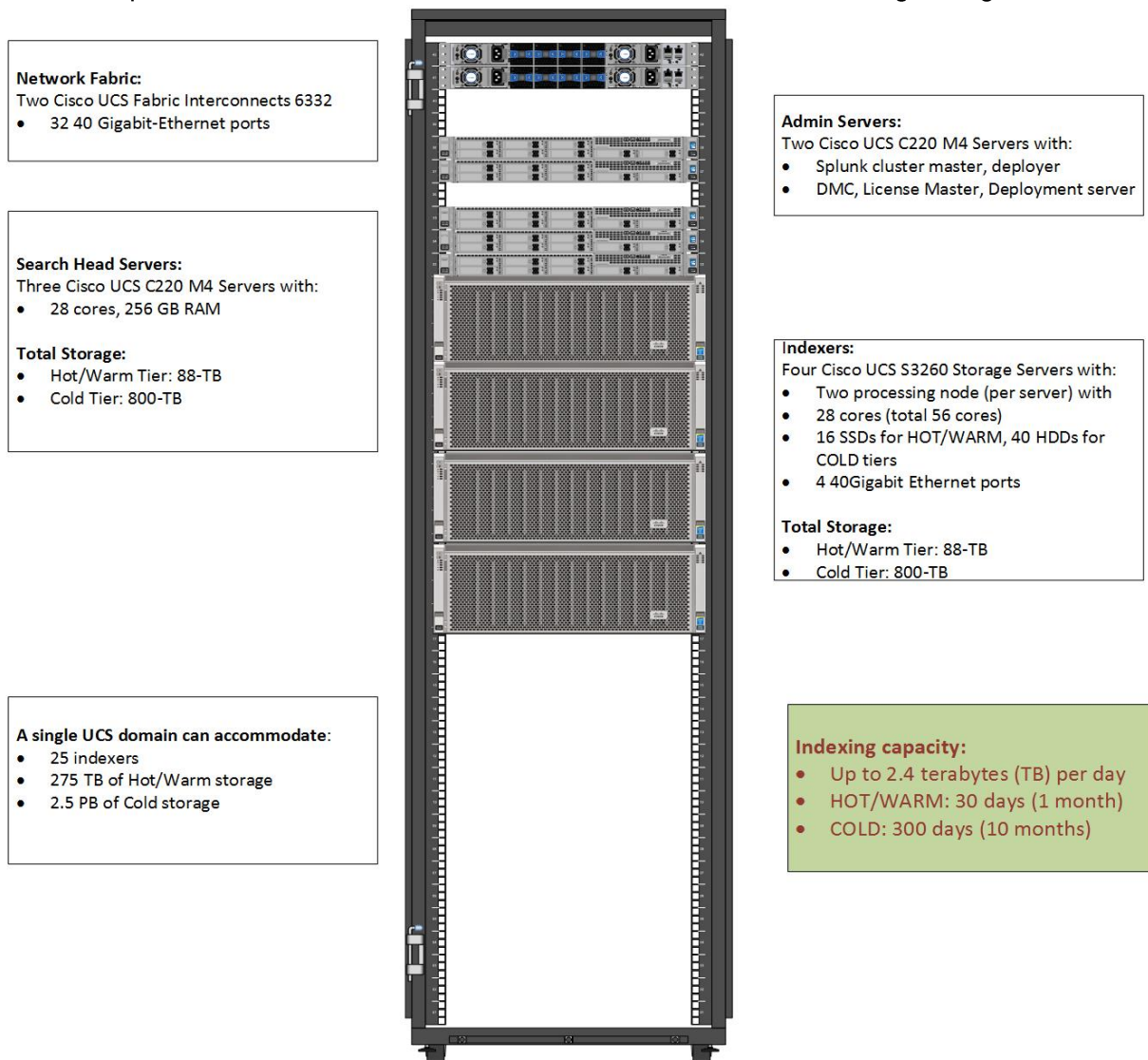
Figure 8 Cabling for Fabric Interconnects and Cisco C220 server



Rack Appearance

Figure 9 shows the single rack configuration containing five Cisco C220 M4 servers and four Cisco UCS S3260 Storage Servers. Each server is connected to each Fabric Interconnect by means of a dedicated (that is, directly) 10 Gigabit Ethernet link. Individual server connectivity diagrams can be seen above.

Figure 9 Splunk Distributed Search with Indexer and Search Head Clustering Configuration



Note: 2.4 TB/day is computed based on the indexer’s capability of indexing 300 Gigabytes per day for core IT operational analytics use cases.

Software Distributions and Versions

The software versions tested and validated in this document are shown in Table 4

Table 4 Software Versions

Layer	Component	Version or Release
Compute	Cisco UCS C220 M4	C220M4.2.0.13d
	Cisco UCS S3260	S3260M4.2.0.13c
Network	Cisco UCS 6332UP	UCS 3.1(2b) A
	Cisco UCS VIC1387 Firmware	4.1 (2d)
	Cisco UCS VIC1387 Driver	2.3.0.20
Storage	LSI SAS 3108	24.12.1-0049
Software	Red Hat Enterprise Linux Server	7.2 (x86_64)

	Cisco UCS Manager	3.1 (2b)
	Splunk Enterprise	6.5

To learn more about Splunk Enterprise, visit: <http://www.splunk.com>.



Note: The latest drivers can be downloaded from the link below:

<https://software.cisco.com/download/release.html?mdfid=283862063&flowid=25886&softwareid=283853158&release=1.5.7d&relind=AVAILABLE&rellifecycle=&reltype=latest>.

The latest supported RAID controller driver is already included with the RHEL 7.2 operating system. Broadwell CPUs (E5-2600 v4 processors) are supported from Cisco UCS firmware 3.1(2b) onward.

Fabric Configuration

To configure a fully redundant, highly available Cisco UCS 6332 Fabric Interconnect configuration, complete the following steps:

1. Initial setup of Fabric Interconnect A and B.
2. Connect to UCS Manager with the virtual IP address using a web browser.
3. Launch UCS Manager.
4. Enable server and uplink ports.
5. Start discovery process.
6. Create pools and policies for the Service Profile templates.
7. Create Service Profile templates and Service Profiles for admin nodes, search heads, and indexers.
8. Associate Service Profiles to servers.

Performing Initial Setup of Cisco UCS 6332 Fabric Interconnects

This section describes the steps to perform initial setup of the Cisco UCS 6332 Fabric Interconnects A and B.

Configure Fabric Interconnect A

1. Connect to the console port on the first Cisco UCS 6332 Fabric Interconnect.
2. At the prompt to enter the configuration method, enter `console` to continue.
3. If asked to either perform a new setup or restore from backup, enter `setup` to continue.
4. Enter `y` to continue to set up a new Fabric Interconnect.
5. Enter `y` to enforce strong passwords.

6. Enter the password for the admin user.
7. Enter the same password again to confirm the password for the admin user.
8. When asked if this Fabric Interconnect is part of a cluster, answer `y` to continue.
9. Enter `A` for the switch fabric.
10. Enter the cluster name for the system name.
11. Enter the Mgmt0 IPv4 address.
12. Enter the Mgmt0 IPv4 netmask.
13. Enter the IPv4 address of the default gateway.
14. Enter the cluster IPv4 address.
15. To configure DNS, answer `y`.
16. Enter the DNS IPv4 address.
17. Answer `y` to set up the default domain name.
18. Enter the default domain name.
19. When asked to `Join centralized management environment (UCS Central)?`, select `No`.



Note: UCS Central extends the policy-based functions and concepts of Cisco UCS Manager across multiple Cisco UCS domains in one or more physical locations. If you are using multiple UCS domains, select `Yes` for this question.

20. Review the settings that were printed to the console, and if they are correct, answer `yes` to save the configuration.
21. Wait for the login prompt to make sure the configuration has been saved.

Configure Fabric Interconnect B

1. Connect to the console port on the second Cisco UCS 6332 Fabric Interconnect.
2. When prompted to enter the configuration method, enter `console` to continue.
3. The installer detects the presence of the partner Fabric Interconnect and adds this Fabric Interconnect to the cluster. Enter `y` to continue the installation.

4. Enter the admin password that was configured for the first Fabric Interconnect.
5. Enter the Mgmt0 IPv4 address.
6. Answer `yes` to save the configuration.
7. Wait for the login prompt to confirm that the configuration has been saved.

For more information on configuring Cisco UCS 6300 Series Fabric Interconnect, go to:

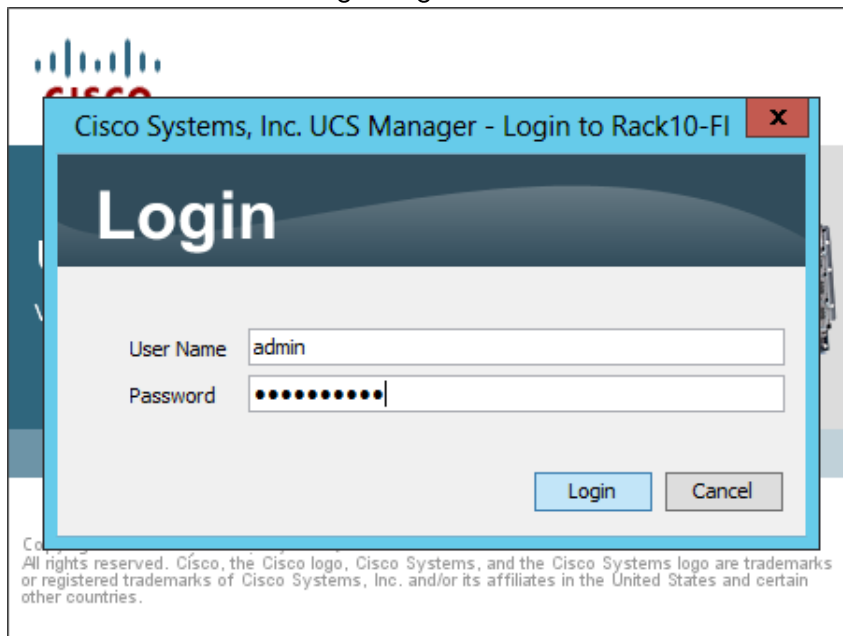
<http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-6300-series-fabric-interconnects/index.html>

Logging Into Cisco UCS Manager

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

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

Figure 10 Cisco UCS Manager Login Screen



Upgrading UCSM Software to Version 3.1(2b)

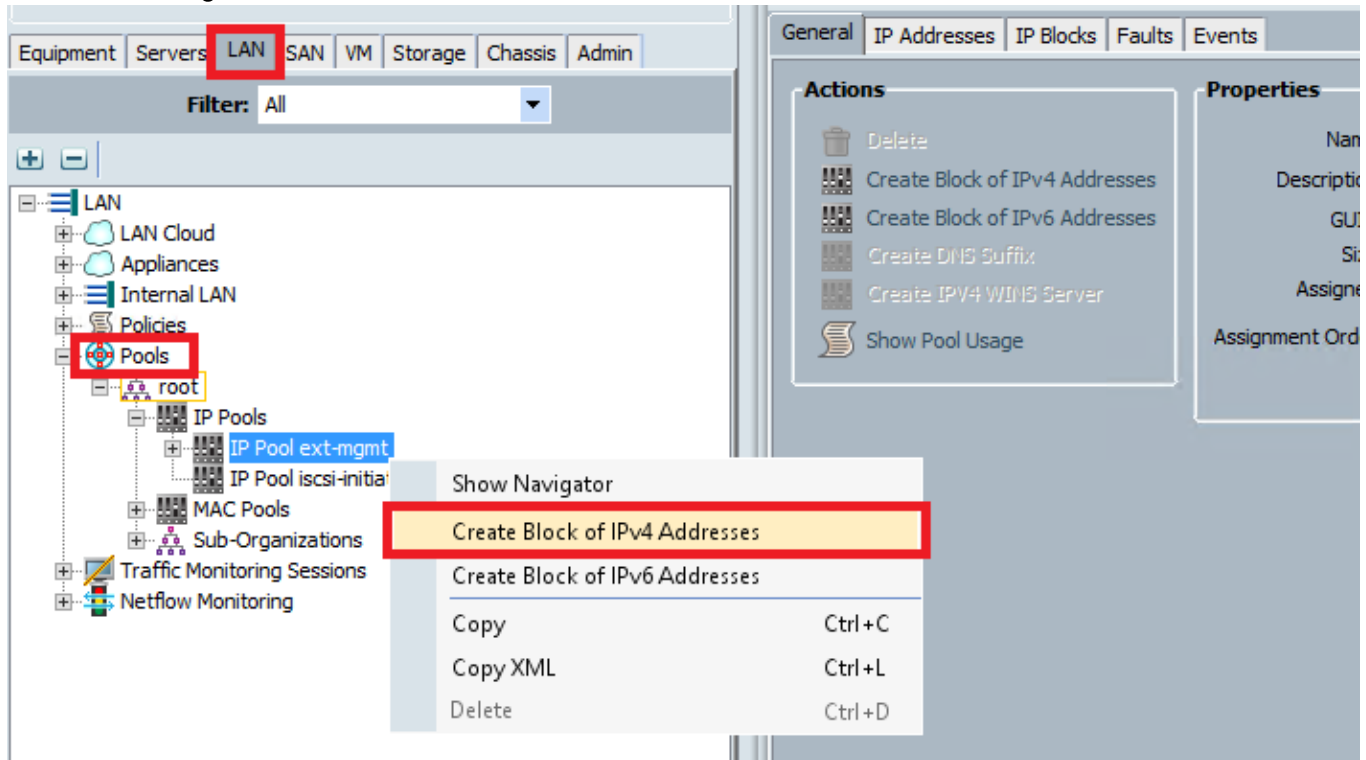
This document assumes the use of UCS 3.1(2b). Refer to [Cisco UCS 3.1 Release](#). Upgrade the Cisco UCS Manager software and UCS 6332 Fabric Interconnect software to version 3.1(2b). Also, make sure the UCS C-Series version 3.1(2b) software bundles are installed on the Fabric Interconnects.

Adding a Block of IP Addresses for KVM Access

To create a block of KVM IP addresses for server access in the Cisco UCS environment, complete the following steps:

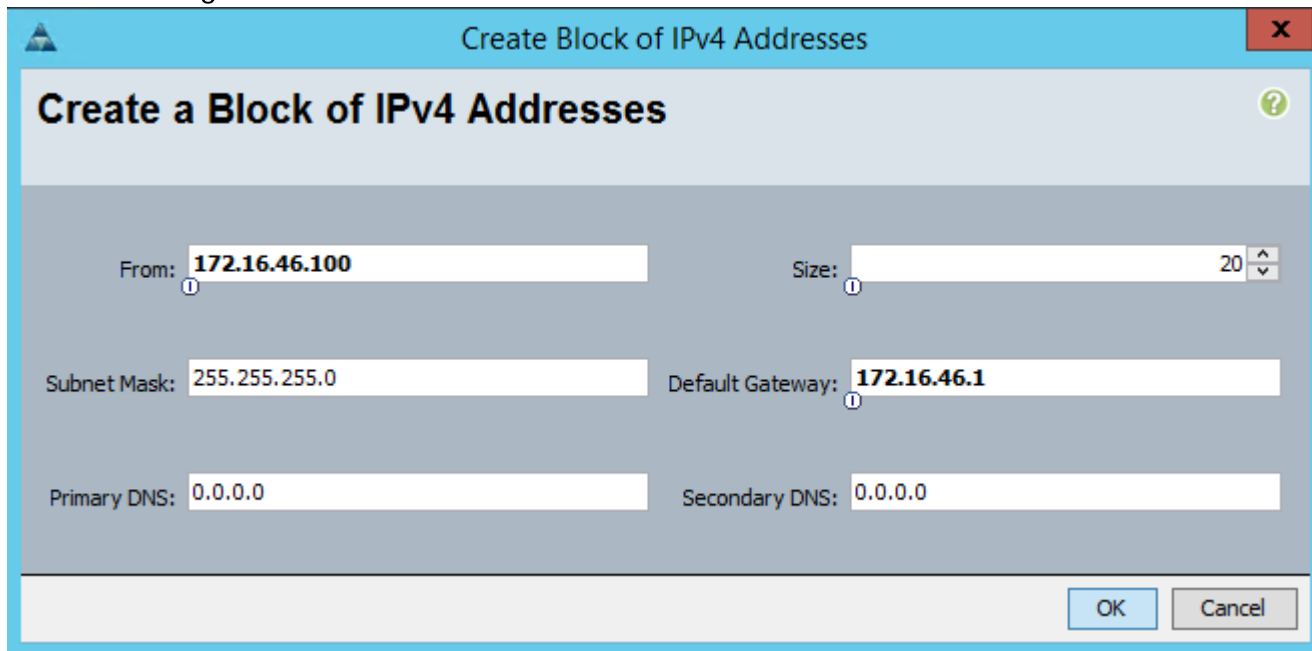
1. Select the `LAN` tab in the left pane in the Cisco UCS Manager GUI.
2. Select `Pools` → `IP Pools` → `IP Pool ext-mgmt`.
3. Right-click `IP Pool ext-mgmt`.
4. Select `Create Block of IPv4 Addresses` as shown in Figure 11

Figure 11 Adding Block of IPv4 Addresses for KVM Access: Part 1



5. Enter the starting IP address of the block and number of IPs needed (at least 13), as well as the subnet and gateway information as shown in Figure 12
6. Click `OK` to create the IP block.
7. Click `OK` in the message box.

Figure 12 Adding Block of IPv4 Addresses for KVM Access: Part 2



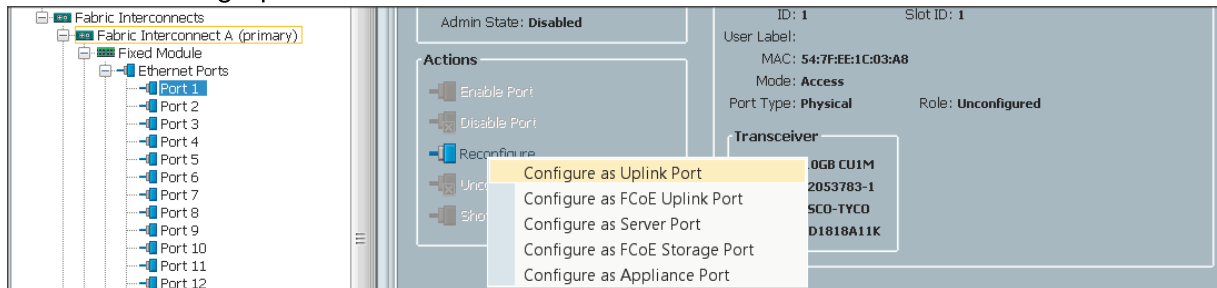
Enabling Uplink Ports

To enable uplink ports, complete the following steps:

1. Select the `Equipment` tab at the top of the left window.
2. Select `Equipment` → `Fabric Interconnects` → `Fabric Interconnect A (primary)` → `Fixed Module`.
3. Expand the `Ethernet Ports` section.
4. Select `Port 1` (which is connected to the uplink switch), then select `Reconfigure` >

`Configure as Uplink Port`, as shown in Figure 13

Figure 13 Enabling Uplink Ports



5. Select `Show Interface` and select `40GB` for Uplink Connection.
6. A pop-up window appears, asking to confirm your selection. Click `Yes`, then click `OK` to continue.

7. Select `Equipment` → `Fabric Interconnects` → `Fabric Interconnect B (subordinate)` → `Fixed Module`.
8. Expand the `Ethernet Ports` section.
9. Select `Port 1` (which is connected to the uplink switch), right-click, then select `Reconfigure > Configure as Uplink Port`.
10. Select `Show Interface` and select `40GB` for `Uplink Connection`.
11. A pop-up window appears, asking to confirm your selection. Click `Yes`, then click `OK` to continue.

Configuring VLANs

VLANs are configured as shown in Table 5

Table 5 VLAN Configurations

VLAN	Fabric	NIC Port	Function	Failover
default(VLAN1)	A	eth0	Management, User connectivity, Data Ingestion	Fabric Failover to B
vlan11_DATA1	B	eth1	Data Replication	Fabric Failover to A

Both VLANs must be trunked to the upstream distribution switch connecting the Fabric Interconnects. For this deployment, default VLAN1 is configured for management access (Installing and configuring OS, clustershell commands, setup NTP, user connectivity, etc) and for Splunk data ingestion from the forwarders. VLAN `vlan11_DATA1` will be used for the replication traffic between the indexers. This enables Splunk to take advantage of the UCS dual 40Gigabit Ethernet (40GigE) links to isolate the inter-server traffic (that is, index replication) from the ingress traffic (data ingestion from forwarders) on separate 40GigE links.

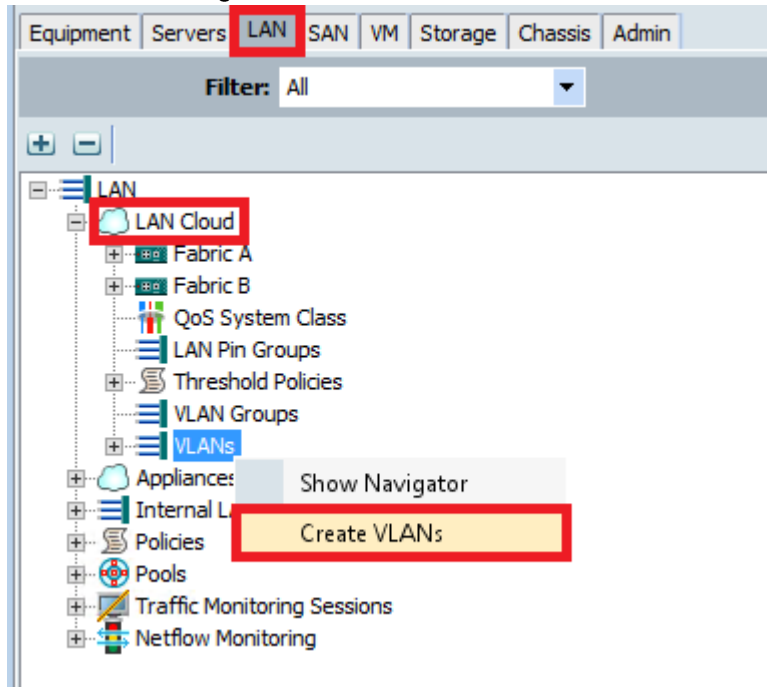


Note: We are using default VLAN1 for management traffic.

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

1. Select the `LAN` tab in the left pane in the UCS Manager GUI.
2. Select `LAN` → `LAN Cloud` → `VLANs`.
3. Right-click the `VLANs` under the root organization.
4. Select `Create VLANs` to create the VLAN, as shown in Figure 14 .

Figure 14 Creating VLAN



5. Enter `vlan11_DATA1` for the VLAN Name.
6. Click the Common/Global radio button for `vlan11_DATA1`.
7. Enter 11 in the VLAN IDs field.
8. Click OK and then, click Finish.
9. Click OK in the success message box.

Figure 15 Creating VLANs for Splunk Data Traffic

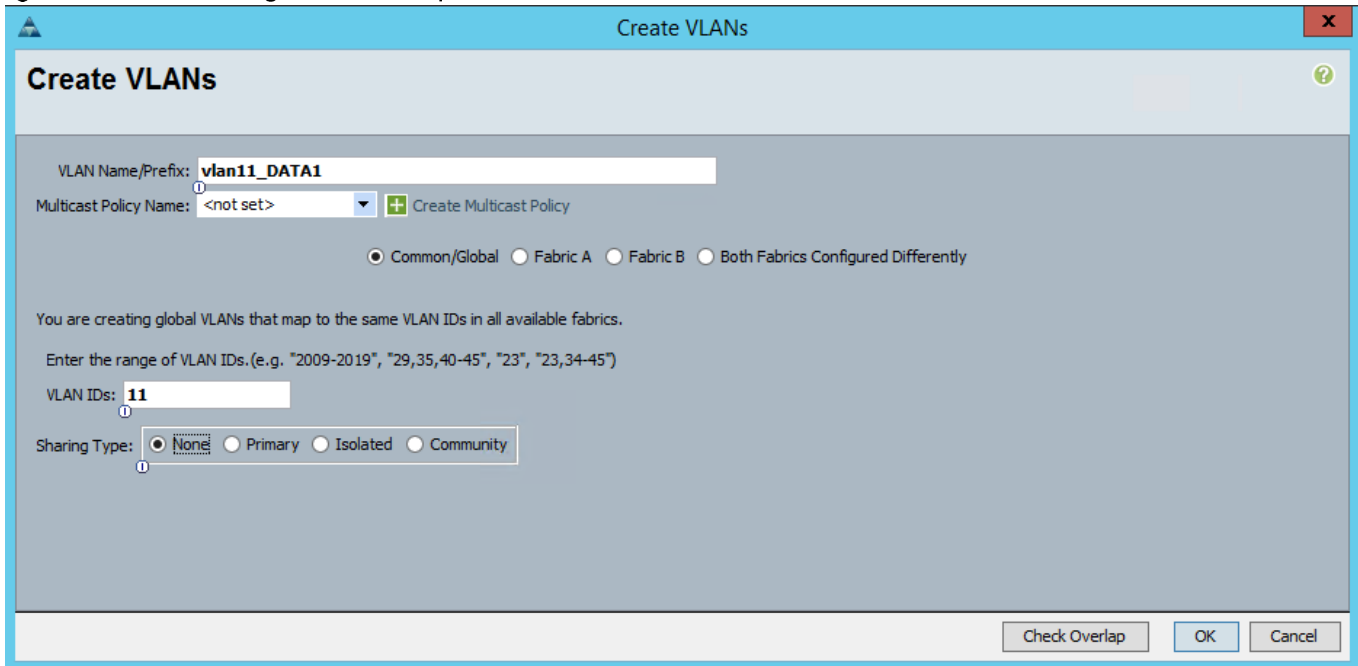
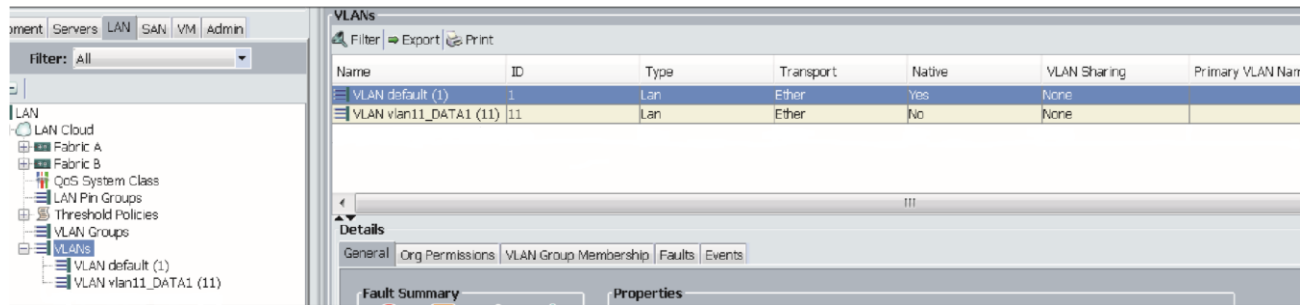


Figure 16 shows the created VLANs.

Figure 16 List of VLANs



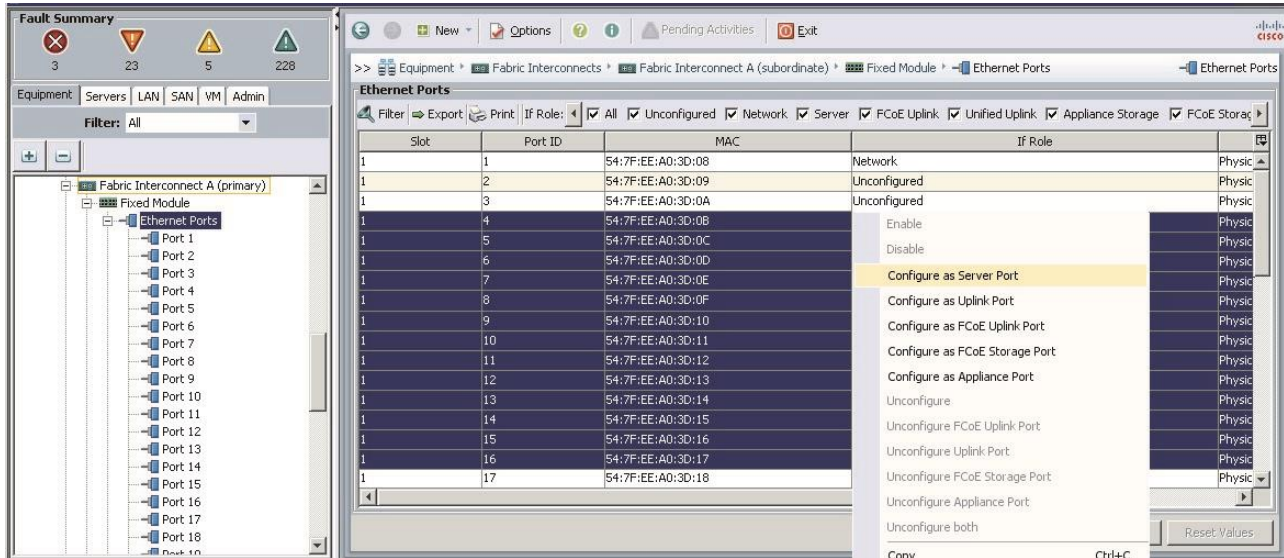
Enabling Server Ports

These steps provide details for enabling server ports:

1. Select the `Equipment` tab on the top left of the window.
2. Select `Equipment` → `Fabric Interconnects` → `Fabric Interconnect A (primary)` → `Fixed Module`.
3. Expand the `Ethernet Ports` section.
4. Select all the ports that are connected to the servers, right-click them, and select `Configure as a Server Port`, as shown in Figure 17
5. A pop-up window appears to confirm your selection. Click `Yes`, then `OK` to continue.
6. Select `Equipment` → `Fabric Interconnects` → `Fabric Interconnect B (subordinate)` → `Fixed Module`.

7. Expand the `Ethernet Ports` section.
8. Select all the ports that are connected to the servers, right-click them, and select `Reconfigure > Configure as a Server Port`.
9. A pop-up window appears, asking to confirm your selection. Click `Yes`, then `OK` to continue.

Figure 17 Enabling Server Ports

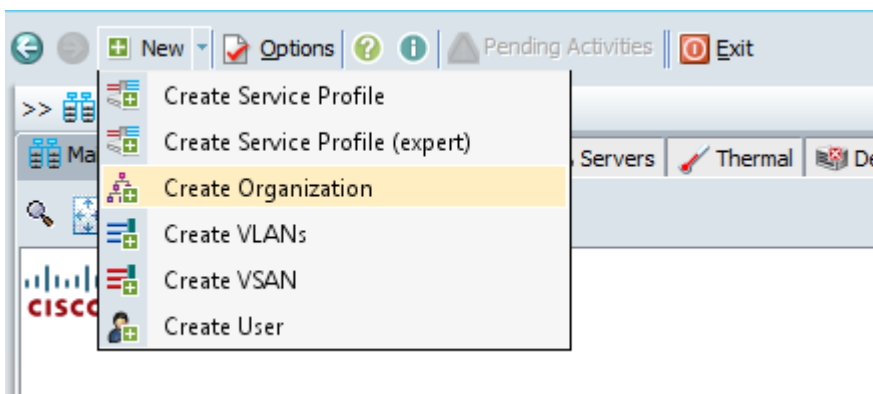


Creating an Organization

Organizations are used as a means to arrange and restrict access to various groups within the IT organization, thereby enabling multi-tenancy of the compute resources. This document does not assume the use of organizations; however, the necessary steps are provided for future reference. If you create an organization, you can choose it instead of `root` in the remaining instructions of this document (for example, step 1 of the next section, `Creating a Storage Profile for Cisco UCS S3260 Boot Drives`).

To configure an organization within the Cisco UCS Manager GUI, complete the following steps:

1. Click `New` on the top left corner in the right pane in the UCS Manager GUI.
2. Select `Create Organization` from the options.



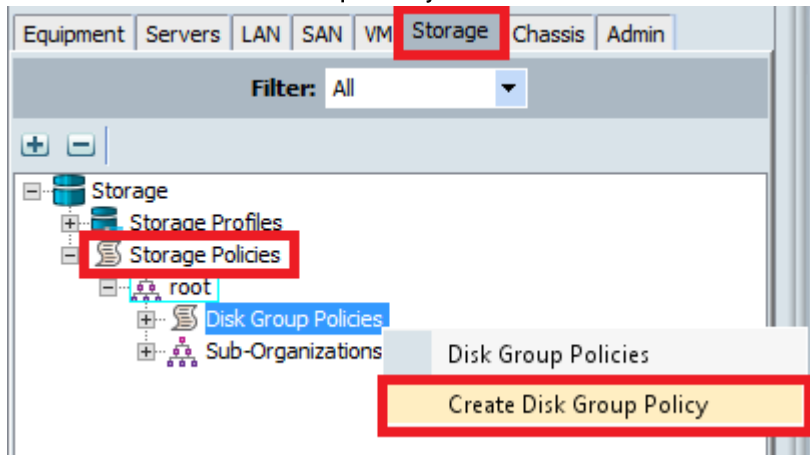
3. Enter a name for the organization.
4. (Optional) Enter a description for the organization.
5. Click **OK**.
6. Click **OK** in the success message box.

Creating a Storage Profile for Cisco UCS S3260 Boot Drives

To create a storage profile for the boot drives of the Cisco UCS S3260 Storage Servers, complete the following steps:

1. Go to the **Storage** tab and expand **Storage**→**Storage Policies**→**root**.
2. Right click on **Disk Group Policies** and click **Create Disk Group Policies** as shown in Figure 18

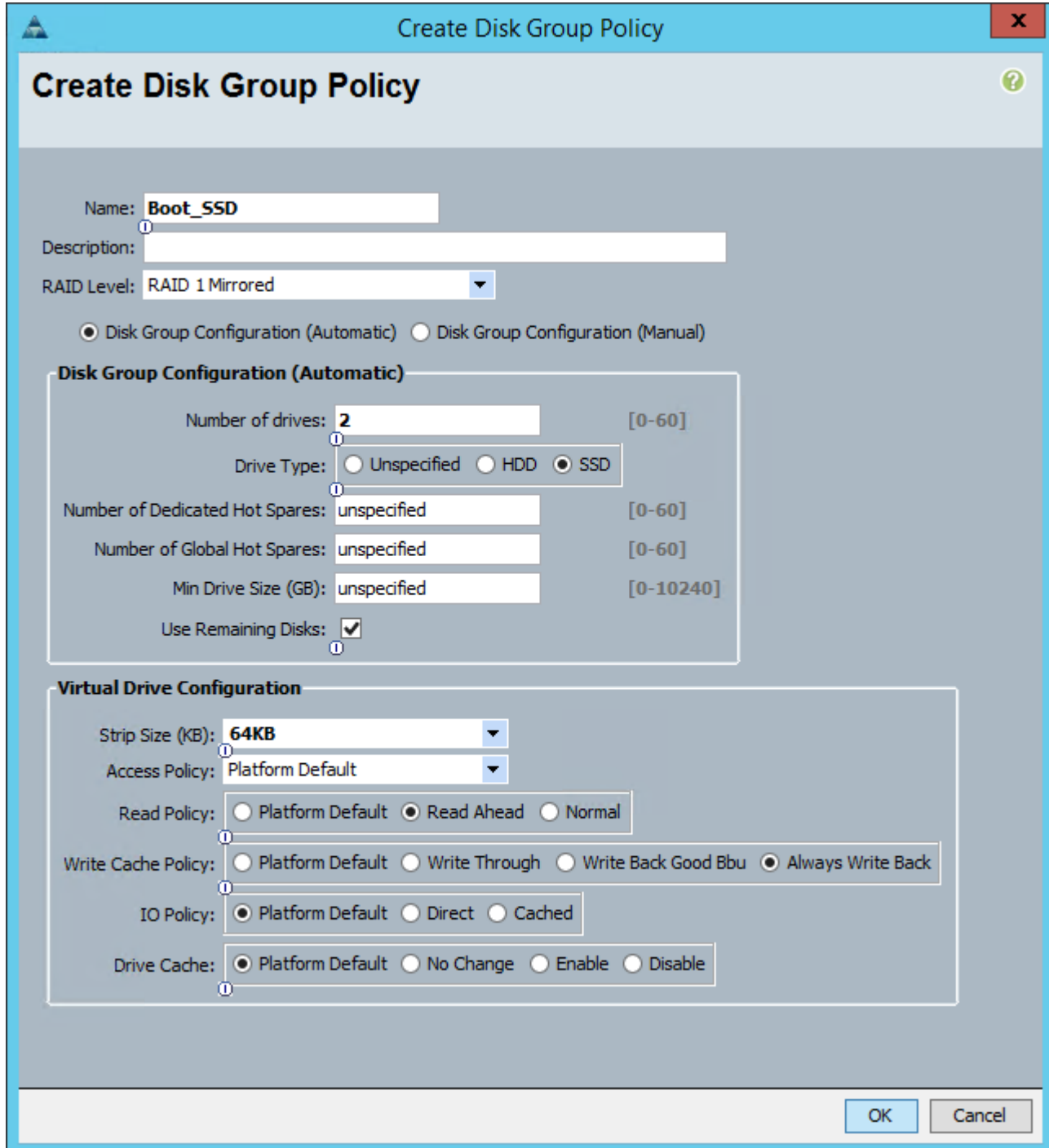
Figure 18 Create Disk Group Policy



3. In the **Create Disk Policy** window, configure the following parameters and click **OK**, as shown in Figure 19
 - a. Name = `Boot_SSD`
 - b. RAID Level = `RAID 1 Mirrored`
 - c. Disk Group Configuration = `Automatic`
 - d. Number of Drives = `2`
 - e. Drive Type = `SSD`
 - f. Use Remaining Disks = `checked`
 - g. Strip Size = `64 KB`
 - h. Access Policy = `Platform Default`
 - i. Read Policy = `Read Ahead`

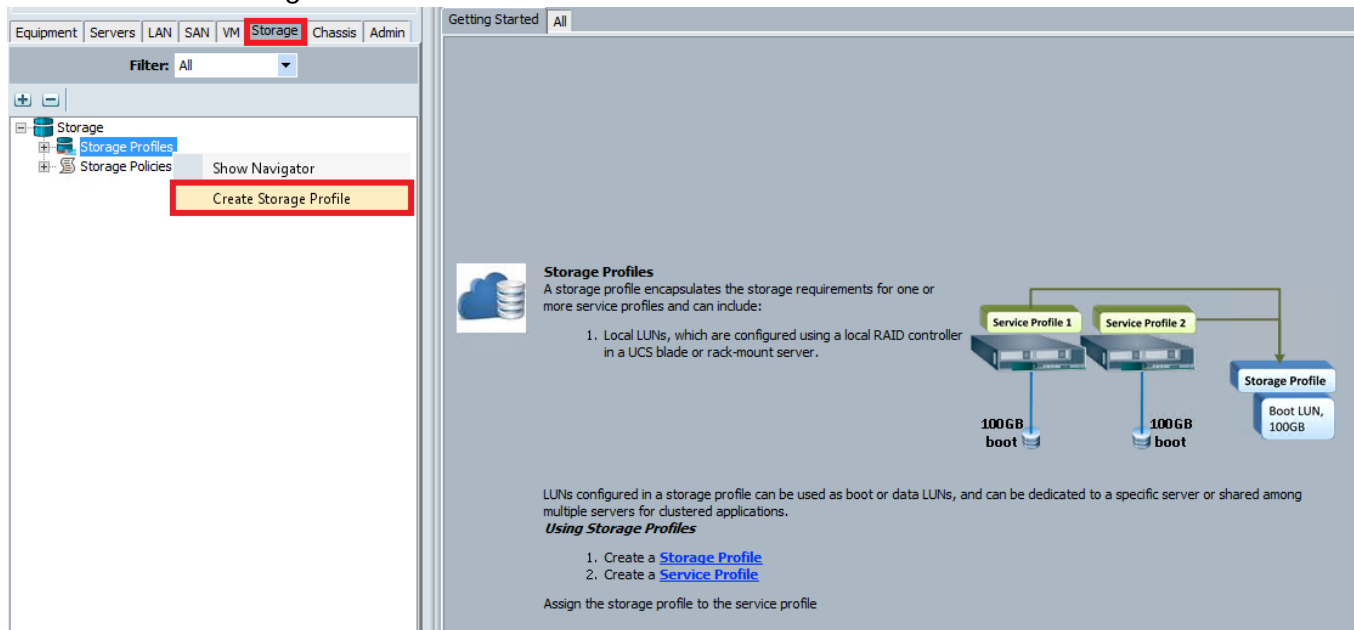
- j. Write Cache Policy = Write Back Good Bbu
- k. IO Policy = Platform Default
- l. Drive Cache = Platform Default

Figure 19 Create Disk Group Policy - SSD



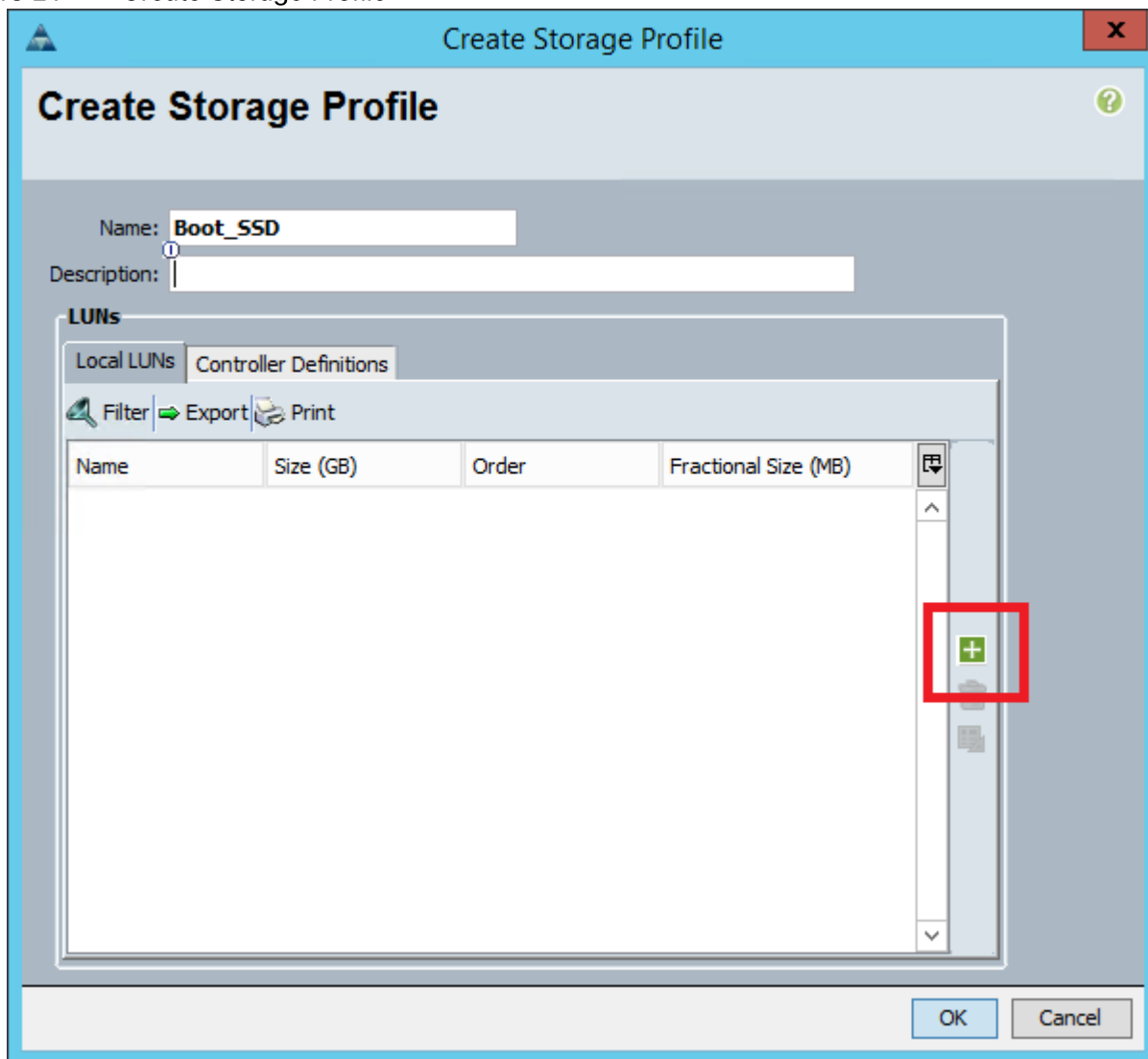
4. In the Storage tab, right click on Storage Profile, and click Create Storage Profile as shown in Figure 20

Figure 20 Create Storage Profile



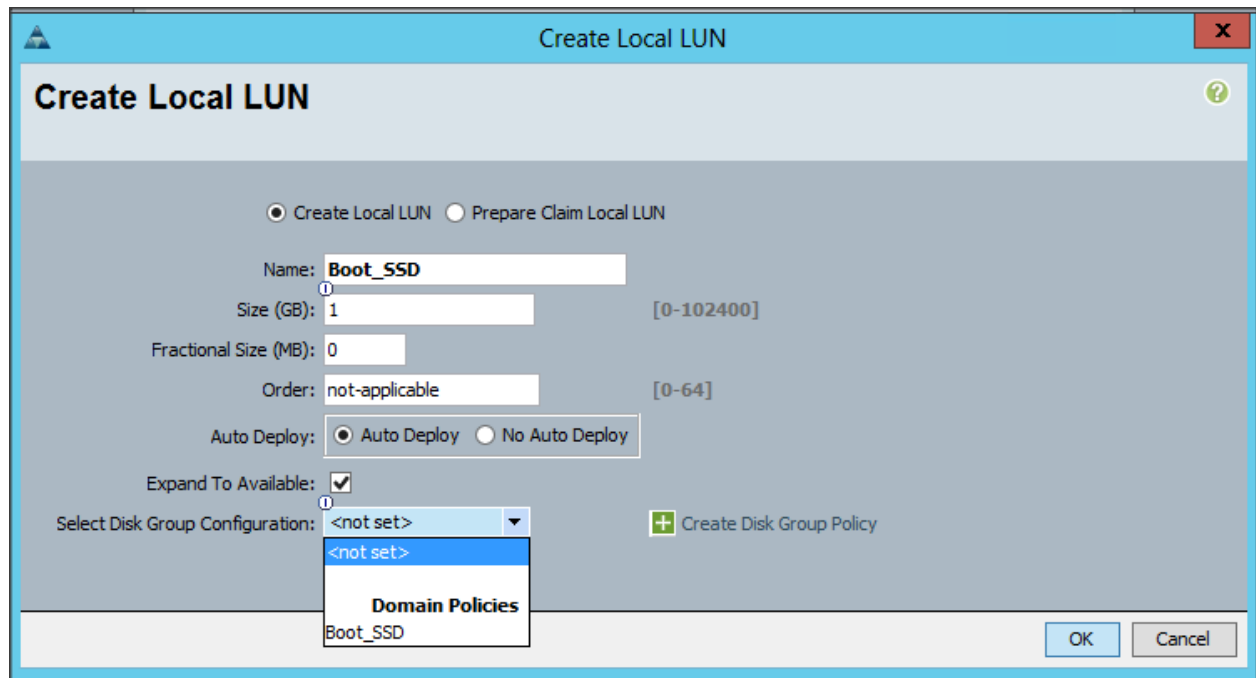
5. Enter `Boot_SSD` in the Name field. Under Local LUNs, click "+" to add local LUN, as shown in Figure 21

Figure 21 Create Storage Profile



6. In the `Create Local LUN` window, enter the name `Boot_SSD`, as shown in Figure 22
7. Check the `Expand to Available` checkbox to use all available space.
8. Under the `Select Disk Group Configuration` drop down list, choose `Boot_SSD`, which was created earlier.
9. Click `OK` and `OK` again to complete the configuration.

Figure 22 Create Local LUN



Creating a Storage Profile for Cisco UCS C220 Boot Drives

To create a storage profile for the Cisco UCS C220 rack mount servers, complete the following steps:

1. Go to the `Storage` tab and expand `Storage` → `Storage Policies` → `root`.
2. Right click on `Disk Group Policies` and click `Create Disk Group Policies`.
3. In the `Create Disk Policy` window, configure the following parameters and click `OK`, as shown in Figure 19
 - m. `Name` = `Boot_HDD`
 - n. `RAID Level` = `RAID 1 Mirrored`
 - o. `Disk Group Configuration` = `Automatic`
 - p. `Number of Drives` = `2`
 - q. `Drive Type` = `HDD`
 - r. `Use Remaining Disks` = `checked`
 - s. `Strip Size` = `64 KB`
 - t. `Access Policy` = `Platform Default`
 - u. `Read Policy` = `Read Ahead`
 - v. `Write Cache Policy` = `Write Back Good Bbu`
 - w. `IO Policy` = `Platform Default`
 - x. `Drive Cache` = `Platform Default`

Figure 23 Create Disk Group Policy – HDD

Create Disk Group Policy

Name: **Boot_HDD**

Description:

RAID Level: RAID 1 Mirrored

Disk Group Configuration (Automatic) Disk Group Configuration (Manual)

Disk Group Configuration (Automatic)

Number of drives: **2** [0-60]

Drive Type: Unspecified HDD SSD

Number of Dedicated Hot Spares: unspecified [0-60]

Number of Global Hot Spares: unspecified [0-60]

Min Drive Size (GB): unspecified [0-10240]

Use Remaining Disks:

Virtual Drive Configuration

Strip Size (KB): **64KB**

Access Policy: Platform Default

Read Policy: Platform Default Read Ahead Normal

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

IO Policy: Platform Default Direct Cached

Drive Cache: Platform Default No Change Enable Disable

OK Cancel

4. In the Storage tab, right click on Storage Profile, and click Create Storage Profile.
5. Enter "Boot_HDD" in the name field.
6. Under Local LUNs, click "+" to add local LUN.
7. In the Create Local LUN window, enter the name Boot_HDD.

8. Check the `Expand to Available` checkbox to use all available space.
9. Under the `Select Disk Group Configuration` drop down list, choose `Boot_HDD`, which was created earlier.
10. Click `OK` and `OK` again to complete the configuration.

Creating Chassis Profiles for Cisco S3260 Storage Servers

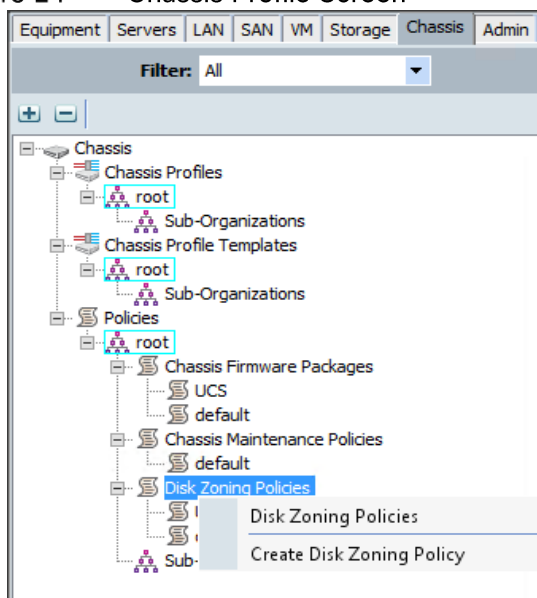
A chassis profile is required for discovering the Cisco S3260 server nodes, assigning the number of drives to particular server nodes, and upgrading the chassis firmware. To create a chassis profile, complete the following steps:

- Create a disk zoning policy to discover and allocate the hard disk drives per server node.
- Assign chassis firmware policy
- Create a chassis profile template.
- Create and associate a chassis profile for each S3260 chassis.

Creating Disk Zoning Policy

1. Click the `Chassis` tab on the top of the left navigation pane in UCS Manager.
2. Expand `Policies` → `root` → `Disk Zoning Policies`.
3. Right-click on `Disk Zoning Policies` and click `Create Disk Zoning Policy`, as shown in Figure 24

Figure 24 Chassis Profile Screen



4. In the `Create Disk Zoning Policy` window, enter `hybrid` for the `Name`, as shown in Figure 25

Figure 25 Disk Zoning Policy Screen

Create Disk Zoning Policy

Name:

Description:

Preserve Config:

Disk Zoning Information

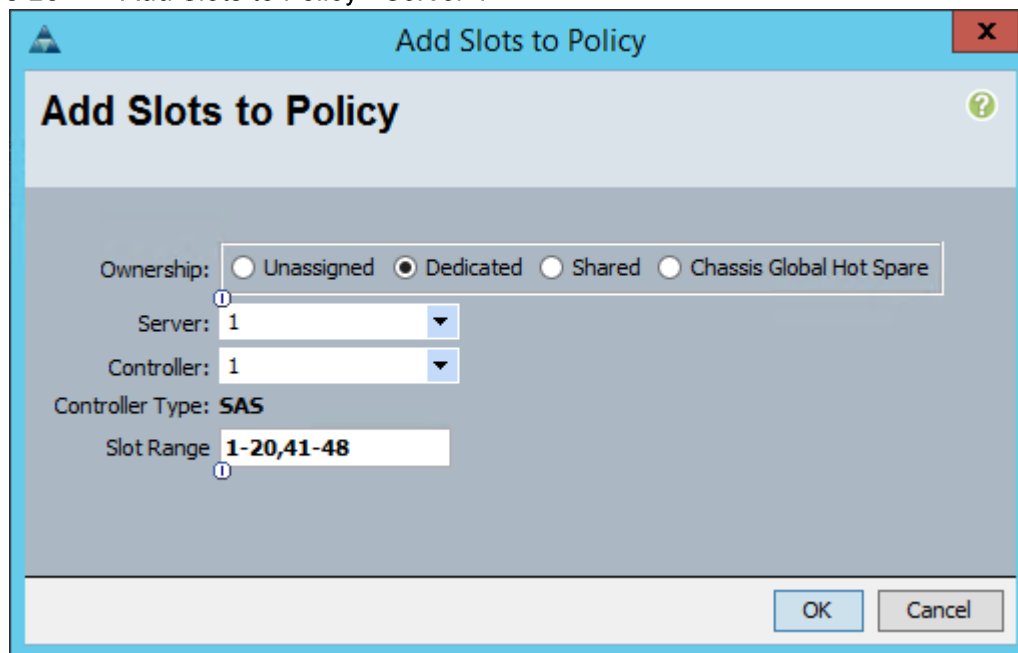
Name	Slot Number	Ownership	Assigned to Ser...	Assigned to Contr...	Controller Type
------	-------------	-----------	--------------------	----------------------	-----------------

+ Add Delete Modify

OK Cancel

5. Click **Add** to add the disk zoning information.
6. In the **Add Slots to Policy** window, select the **Dedicated** radio button.
7. From the **Server** drop down list choose 1.
8. From the **Controller** drop down list, choose 1.
9. For the **Slot Range** field, enter 1-20, 41-48 and click **OK**.

Figure 26 Add Slots to Policy – Server 1



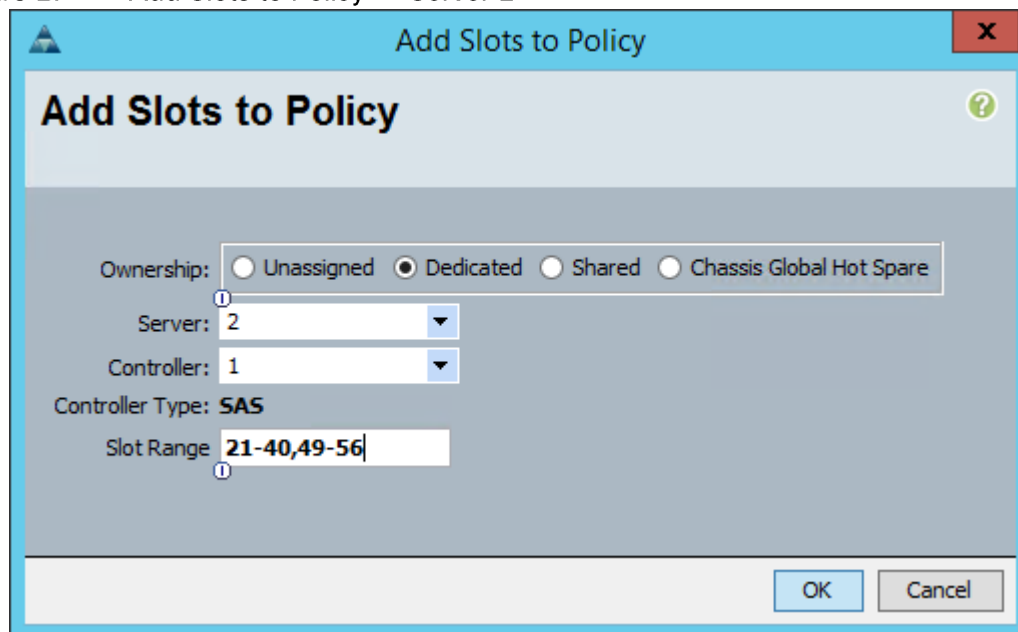
The screenshot shows a dialog box titled "Add Slots to Policy" with a blue header and a red close button. The main area is light gray and contains the following fields:

- Ownership:** A group box containing four radio buttons: Unassigned, Dedicated, Shared, and Chassis Global Hot Spare.
- Server:** A dropdown menu with "1" selected.
- Controller:** A dropdown menu with "1" selected.
- Controller Type:** A text field containing "SAS".
- Slot Range:** A text field containing "1-20,41-48".

At the bottom right, there are two buttons: "OK" and "Cancel".

10. Click **Add** again to assign the disks for the second server.
11. In the **Add Slots to Policy** window, select the **Dedicated** radio button.
12. From the **Server** drop down list, choose 2.
13. From the **Controller** drop down list, choose 1.
14. Enter 21-40, 49-56 for the slot range.

Figure 27 Add Slots to Policy -- Server 2



The screenshot shows a dialog box titled "Add Slots to Policy" with a blue header and a red close button. The main area is light gray and contains the following fields:

- Ownership:** A group box containing four radio buttons: Unassigned, Dedicated, Shared, and Chassis Global Hot Spare.
- Server:** A dropdown menu with "2" selected.
- Controller:** A dropdown menu with "1" selected.
- Controller Type:** A text field containing "SAS".
- Slot Range:** A text field containing "21-40,49-56".

At the bottom right, there are two buttons: "OK" and "Cancel".

15. Click **OK** to add the slots to the disk zoning policy.

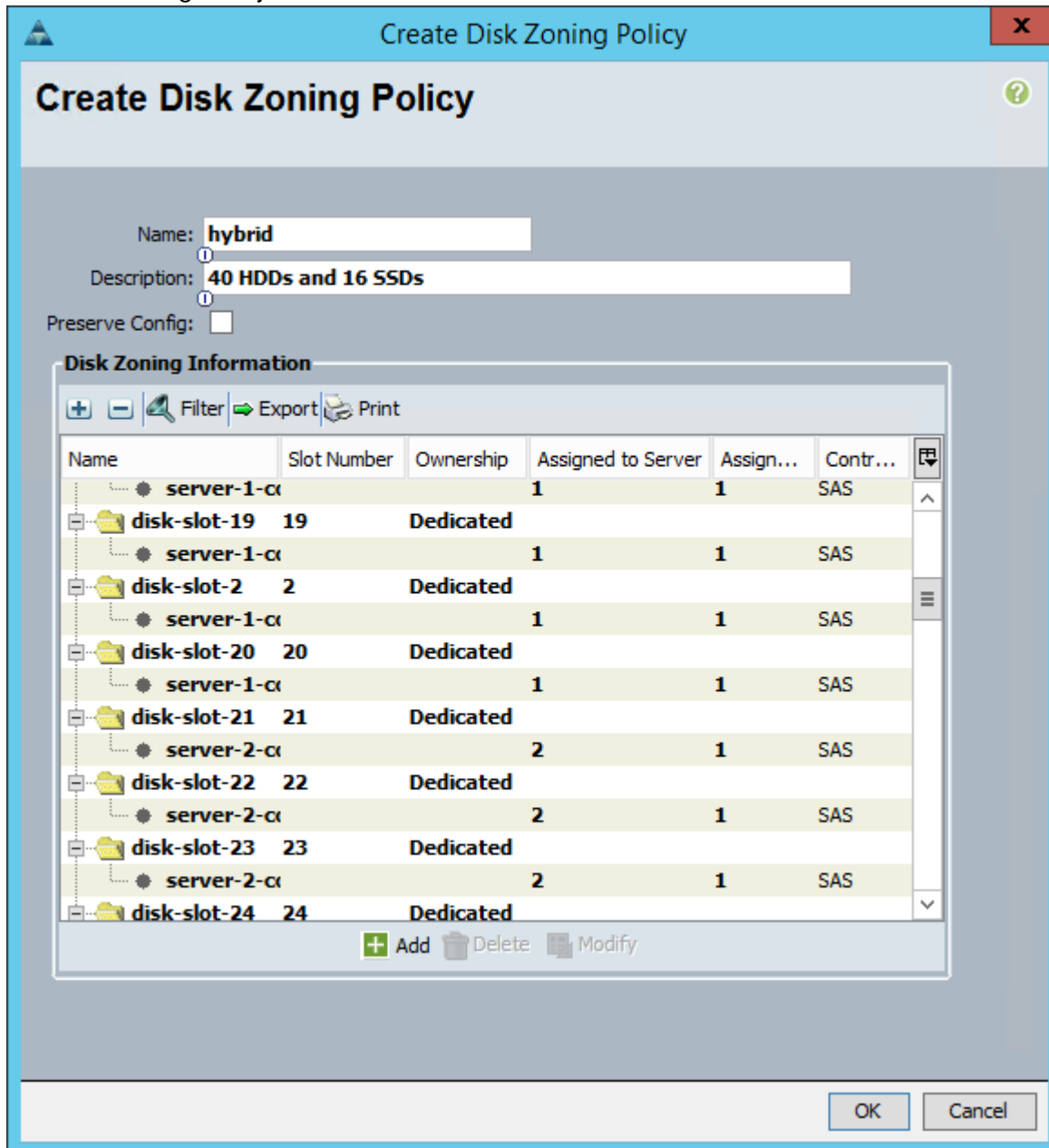


Note: In this configuration, two servers are used, with each server assigned to 28 drives. In a single server configuration, there would be room for 60 drives.

16. Scroll down the Disk Zoning Information section to confirm that all 56 drives have been added and allocated to the correct server. See figure below.

17. Click **OK** to finish creating the disk zoning policy.

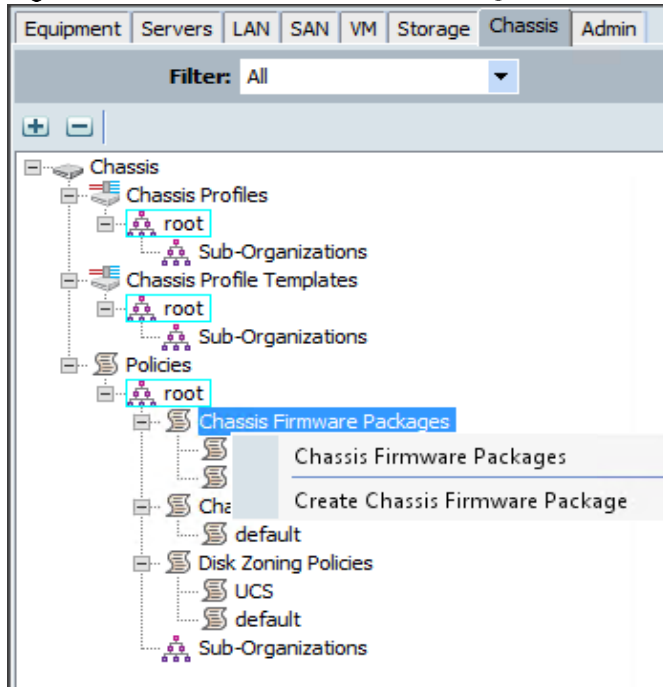
Figure 28 Zoning Policy



Creating Chassis Firmware Package Policy

1. In the **Chassis** tab, expand **Chassis** → **Policies** → **root**.
2. Right click on **Chassis Firmware Packages** and click **Create Chassis Firmware Packages** as shown in the figure below.

Figure 28 Chassis Firmware Packages



3. In the **Create Chassis Firmware Package** window, enter **UCS** as the Name.
4. From the **Chassis Packages** drop down list, choose the appropriate package (must be 3.1(2b) or above) and click **OK**.

Figure 29 Create Chassis Firmware Screen

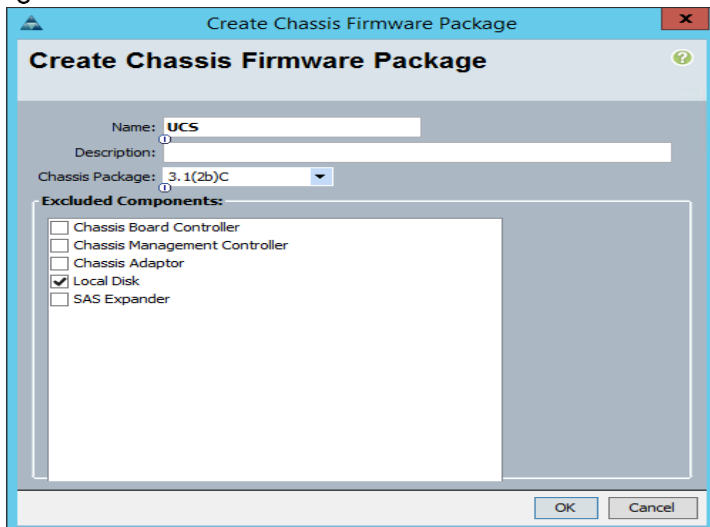


Figure 30 Chassis Firmware Package

Create Chassis Firmware Package

Name:

Description:

Chassis Package:

Excluded Components:

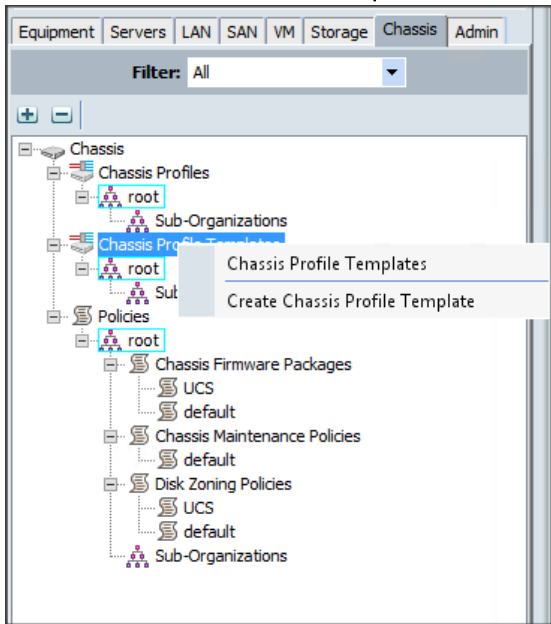
- Chassis Board Controller
- Chassis Management Controller
- Chassis Adaptor
- Local Disk
- SAS Expander

OK Cancel

Creating a Chassis Profile Template

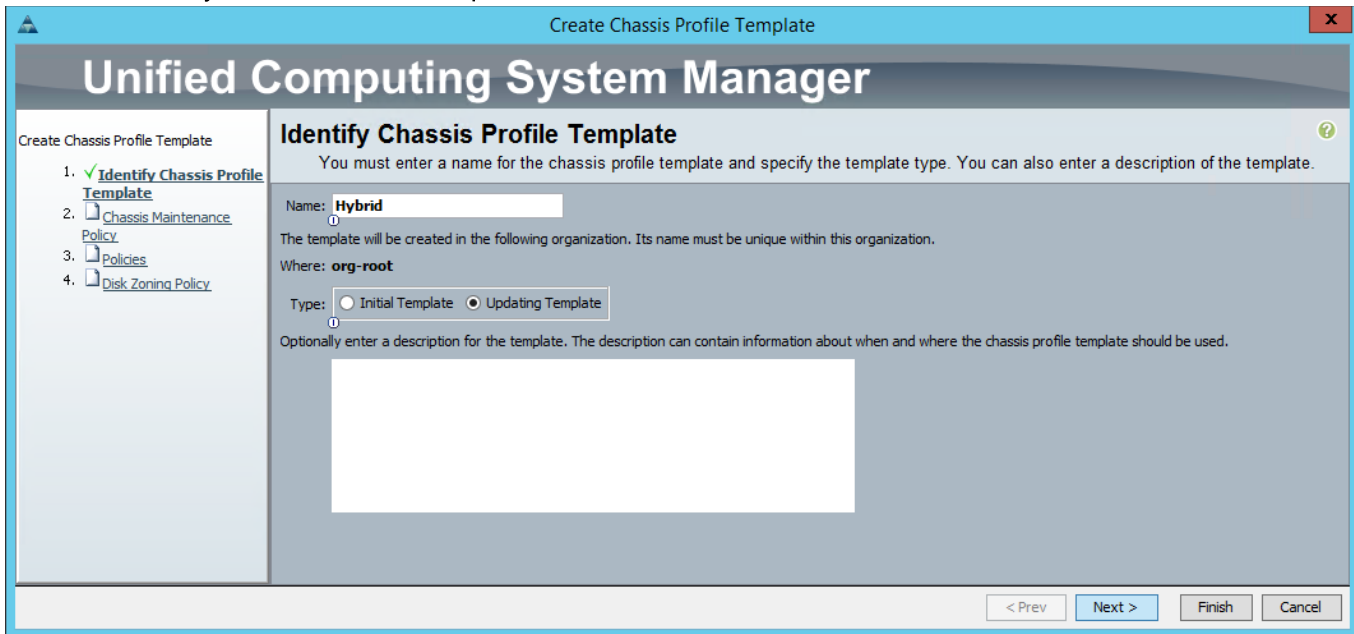
1. Under `Chassis Profile Templates`, right-click and click `Create Chassis Profile Template` as shown in Figure 31

Figure 31 Chassis Profile Template



2. Enter Hybrid for the Name. Select Updating Template for Type.

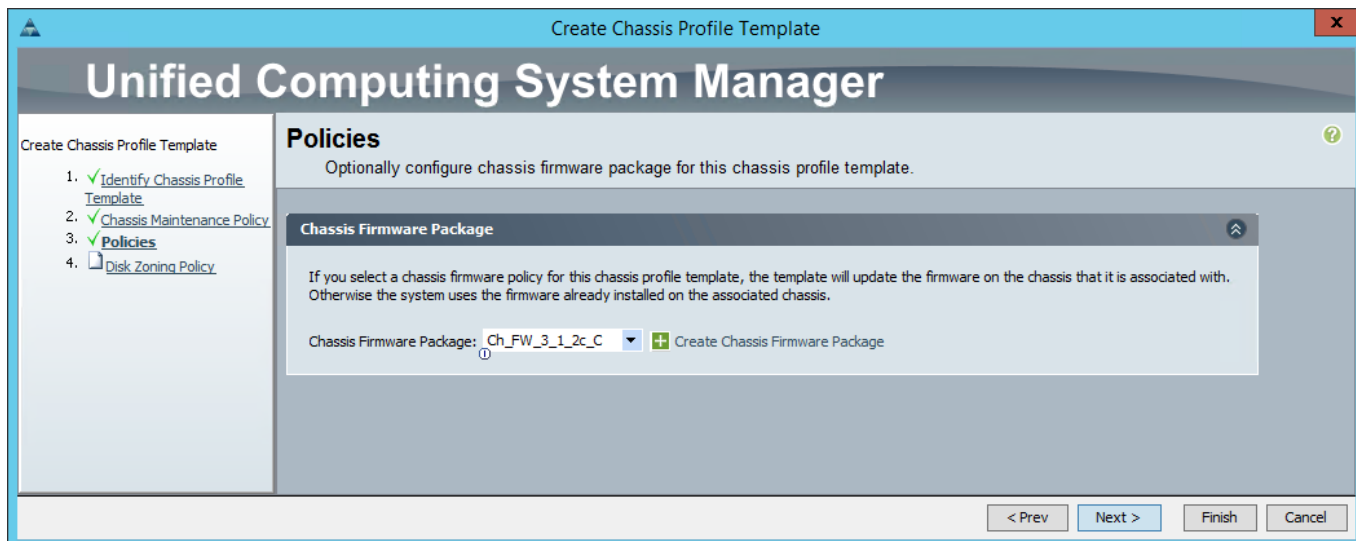
Figure 32 Identify Chassis Profile Template



3. Click Next and Next again to go to the Policies section.

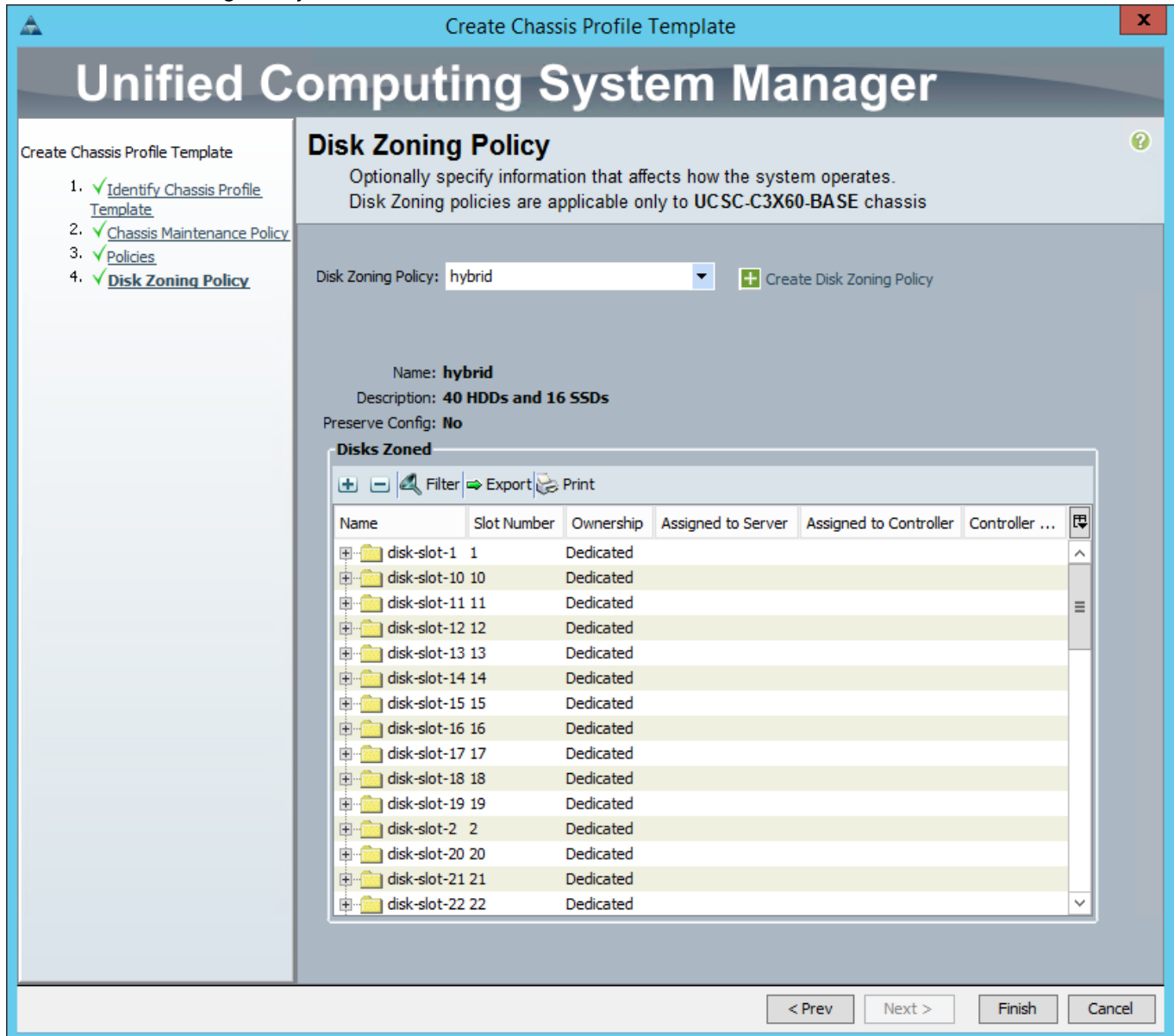
4. Expand the Chassis Firmware Package section. From the Chassis Firmware Package drop down list, choose UCS, and click Next. See Figure 33

Figure 33 UCS Policies



5. From the Disk Zoning Policy drop down list, choose hybrid.

Figure 34 Disk Zoning Policy



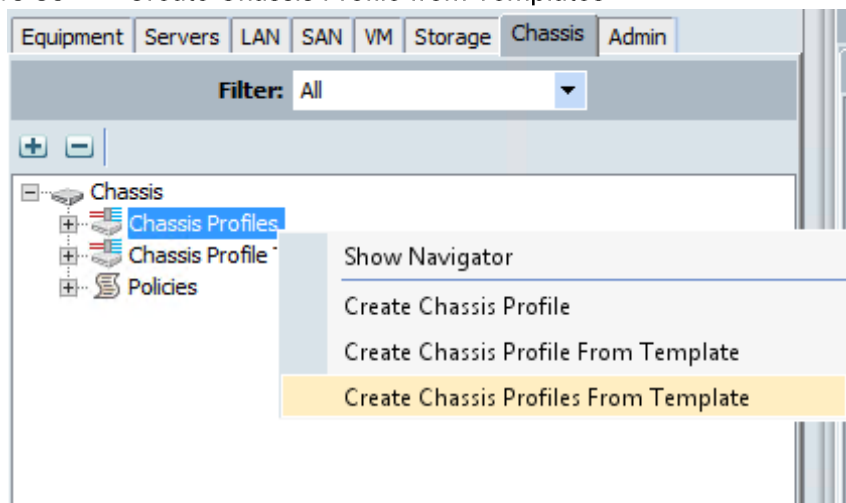
6. Click **Finish**. Click **OK** at the success message.

Creating Chassis Profiles from Template

To create four chassis profiles from the chassis profile template, complete the following steps:

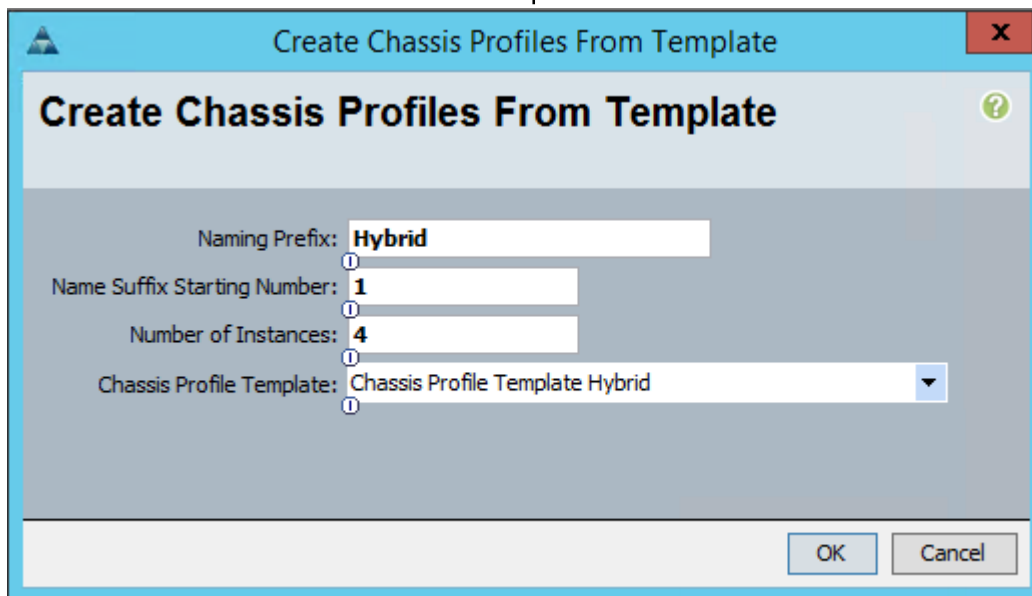
1. **Right-click** on **Chassis Profiles** and click **Create Chassis Profiles from Templates**. See Figure 35

Figure 35 Create Chassis Profile from Templates



2. Enter **Hybrid** for the Naming Prefix.
3. Enter **4** for the Number of Instances.
4. In the Chassis Profile Template drop down menu, choose **Chassis Profile Template Hybrid**, as shown in Figure 36
5. Click **OK**.

Figure 36 Create Chassis Profile from Template



6. Click **OK** on the success dialog box.

Associating Chassis Profiles to Individual Chassis

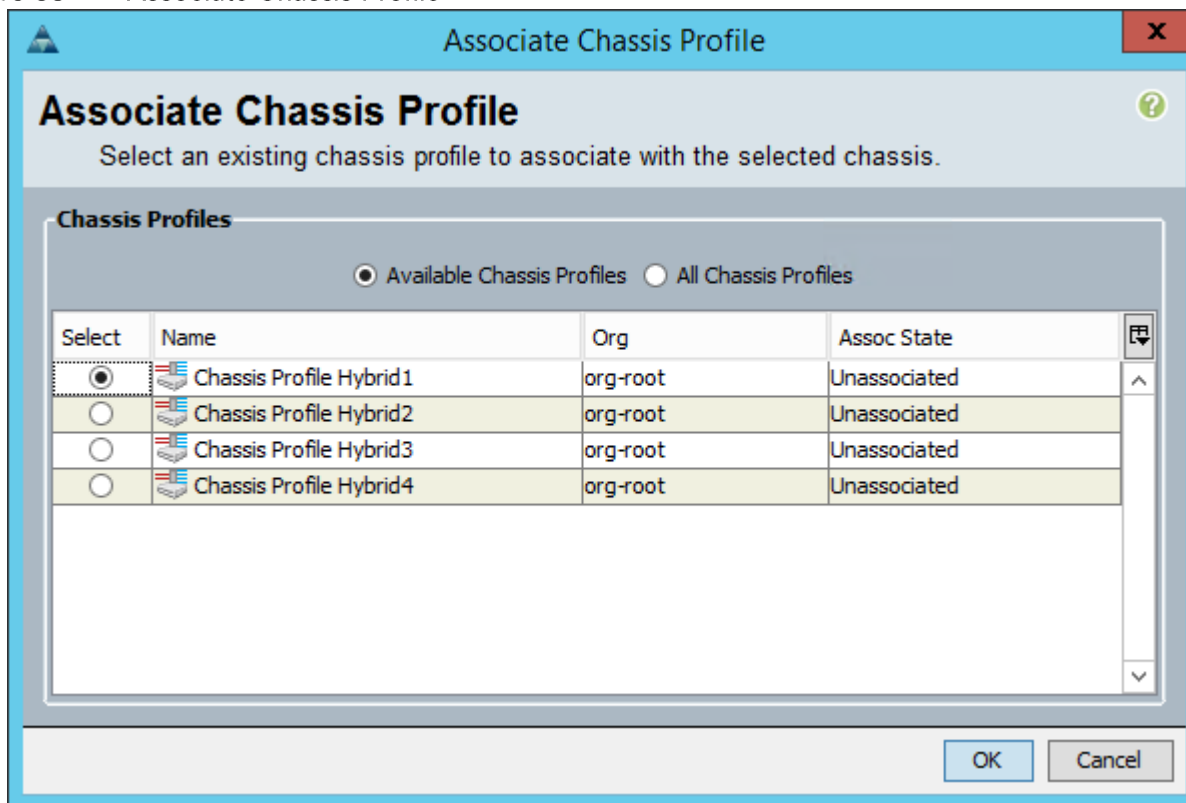
1. In the Cisco UCS Manager UI, select the **Equipment** tab. Under **Equipment**, expand **Chassis**.
2. Select the first chassis and click **Associate Chassis Profile** in the **Actions** section, as shown in Figure 37

Figure 37 Associate Chassis Profile



3. Select Chassis Profile Hybrid 1 and click OK.

Figure 38 Associate Chassis Profile



4. Repeat steps 2 and 3 for the remaining three chassis.
5. Once the chassis profiles are associated, only 28 disks will be assigned to each server node. To verify that, go to `Equipment > Chassis > Chassis 1 > Servers > Server 2`. In the right side of the window, click the `Inventory` tab, then the `Storage and Disks` tabs. Expand `Storage controller SAS 1`. See **Error! Reference source not found**. Note that Server 2's controller is responsible for the drives in slots 21-40 (HDD) and 49-56 (SSD).

Figure 39 Storage Controller SAS 1

The screenshot displays a storage management interface. On the left, a tree view shows the equipment hierarchy, with 'Server 2 (rhel19)' highlighted. On the right, the 'Storage' tab is active, showing a table of disks. The table has the following columns: Name, Size (MB), Serial, Operability, Drive State, Presence, Technology, and Bootable. The rows represent individual disks, with Disk 49 highlighted in red.

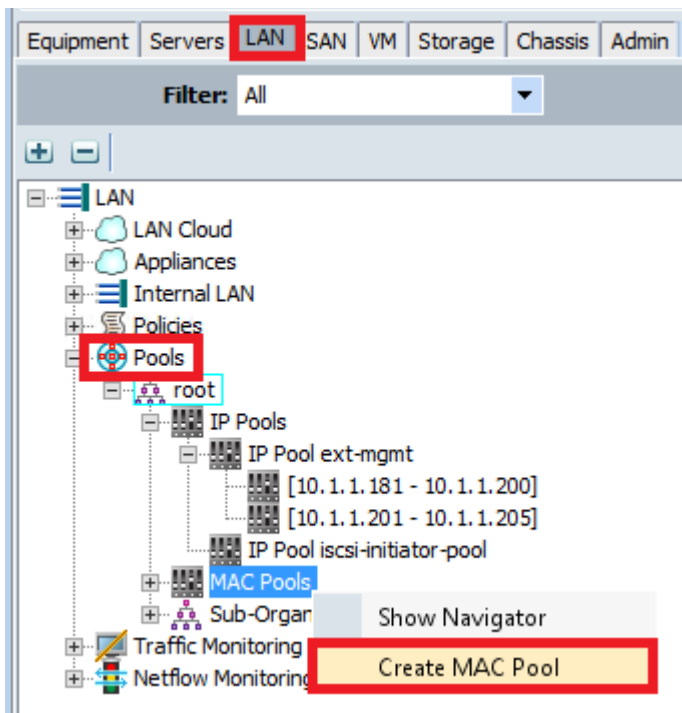
Name	Size (MB)	Serial	Operability	Drive State	Presence	Technology	Bootable
Storage Controller PCH 1							
Storage Controller SAS 1							
Disk 21	7630328	VKGNHM7X	Operable	Jbod	Equipped	HDD	False
Disk 22	7630328	VKGZ4JMX	Operable	Jbod	Equipped	HDD	False
Disk 23	7630328	VKGB8KX	Operable	Jbod	Equipped	HDD	False
Disk 24	7630328	VKHBTORX	Operable	Jbod	Equipped	HDD	False
Disk 25	7630328	VKGBSENX	Operable	Jbod	Equipped	HDD	False
Disk 26	7630328	VKHBNVX	Operable	Jbod	Equipped	HDD	False
Disk 27	7630328	VKH2WVNX	Operable	Jbod	Equipped	HDD	False
Disk 28	7630328	VKH124X	Operable	Jbod	Equipped	HDD	False
Disk 29	7630328	VKGVV7V	Operable	Jbod	Equipped	HDD	False
Disk 30	7630328	VKGVWB5V	Operable	Jbod	Equipped	HDD	False
Disk 31	7630328	VKGYP27X	Operable	Jbod	Equipped	HDD	False
Disk 32	7630328	VKGVY35X	Operable	Jbod	Equipped	HDD	False
Disk 33	7630328	VKH4E0B3X	Operable	Jbod	Equipped	HDD	False
Disk 34	7630328	VKH5H52X	Operable	Jbod	Equipped	HDD	False
Disk 35	7630328	VKGVJWAX	Operable	Jbod	Equipped	HDD	False
Disk 36	7630328	VKGYM29X	Operable	Jbod	Equipped	HDD	False
Disk 37	7630328	VKH2GNAX	Operable	Jbod	Equipped	HDD	False
Disk 38	7630328	VKH24MX	Operable	Jbod	Equipped	HDD	False
Disk 39	7630328	VKH5XHLX	Operable	Jbod	Equipped	HDD	False
Disk 40	7630328	VKGPSVNX	Operable	Jbod	Equipped	HDD	False
Disk 49	380516	25J0A00CTZV7	Operable	Jbod	Equipped	SSD	False
Disk 50	380516	3SL0A039TZV7	Operable	Jbod	Equipped	SSD	False
Disk 51	380516	25J0A004TZV7	Operable	Jbod	Equipped	SSD	False
Disk 52	380516	25J0A057TZV7	Operable	Jbod	Equipped	SSD	False
Disk 53	380516	15M0A00MTZV7	Operable	Jbod	Equipped	SSD	False
Disk 54	380516	25J0A003TZV7	Operable	Jbod	Equipped	SSD	False
Disk 55	380516	25J0A04ATZV7	Operable	Jbod	Equipped	SSD	False

Creating Pools for Service Profile Templates

Creating MAC Address Pools

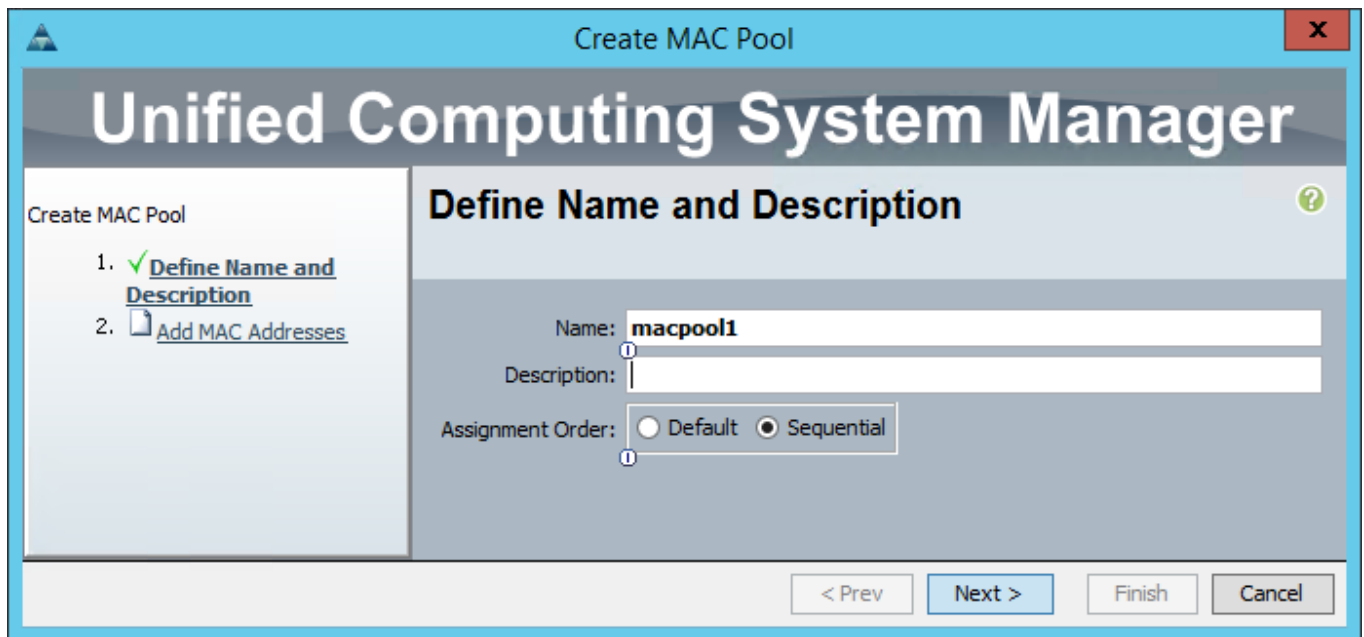
To create MAC address pools, complete the following steps:

1. Select the LAN tab on the left of the window.
2. Select Pools → root.
3. Right-click MAC Pools under the root organization.
4. Select Create MAC Pool to create the MAC address pool.



5. Enter `macpool1` for the MAC Pool name.
6. (Optional) Enter a description of the MAC pool.
7. Select the `Assignment Order` to be `Sequential`.

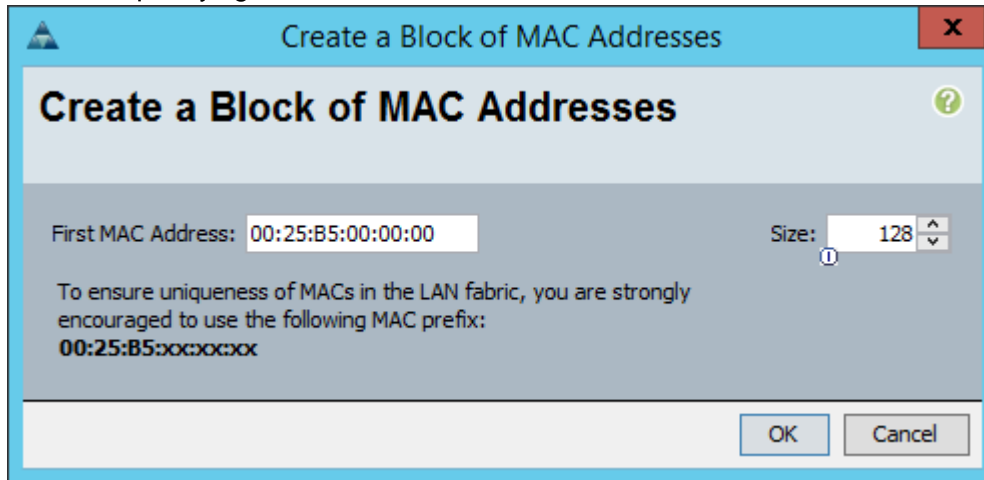
Figure 40 Create MAC Pool Window



8. Click `Next`.
9. Click `Add`.

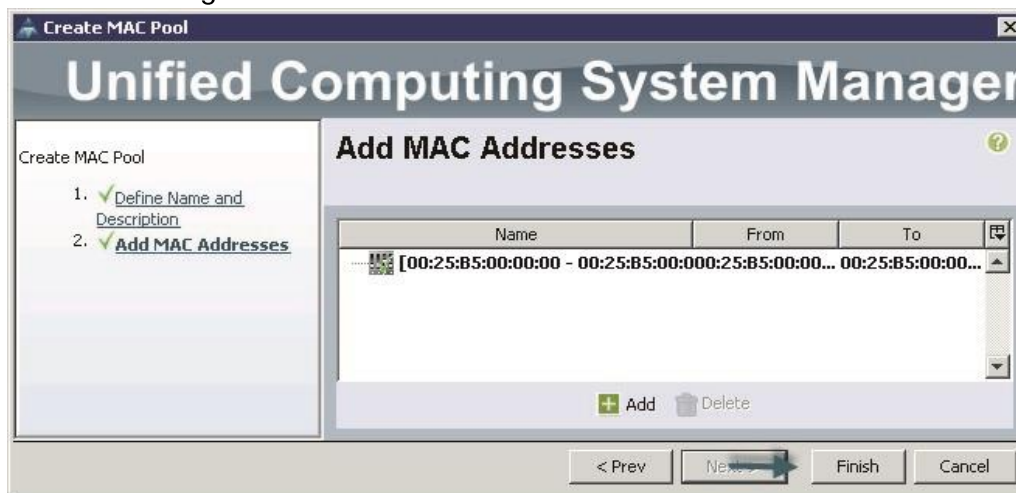
10. Specify a starting MAC address.
11. Specify a size of the MAC address pool, which is sufficient to support the available server resources, as shown in **Error! Reference source not found.**
12. Click **OK**.

Figure 41 Specifying First MAC Address and Size



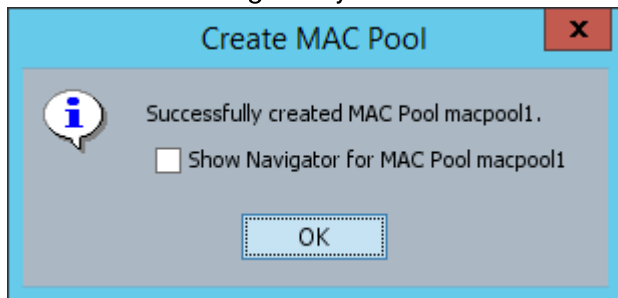
13. Click **Finish** as shown in Figure 42

Figure 42 Adding MAC Addresses



14. When the message box displays, click **OK**, as shown in **Error! Reference source not found.**

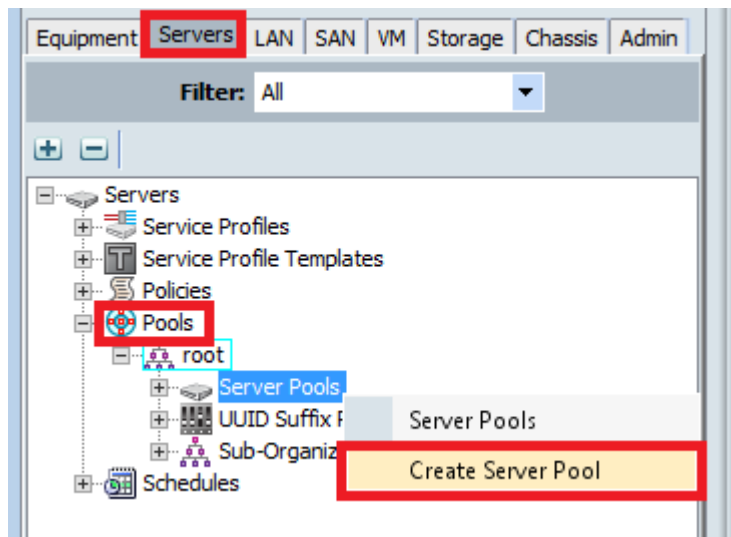
Figure 43 Confirming Newly Added MAC Pool



Creating Server Pools

A server pool contains a set of servers. These servers typically share the same characteristics. Those characteristics can be their location in the chassis, or an attribute such as server type, amount of memory, local storage, type of CPU, or local drive configuration. You can manually assign a server to a server pool, or use server pool policies and server pool policy qualifications to automate the assignment. Follow these steps to configure the server pool within the Cisco UCS Manager GUI:

1. Select the `Servers` tab in the left pane in the UCS Manager GUI.
2. Select `Pools` → `root`.
3. Right-click `Server Pools`.
4. Select `Create Server Pool`.



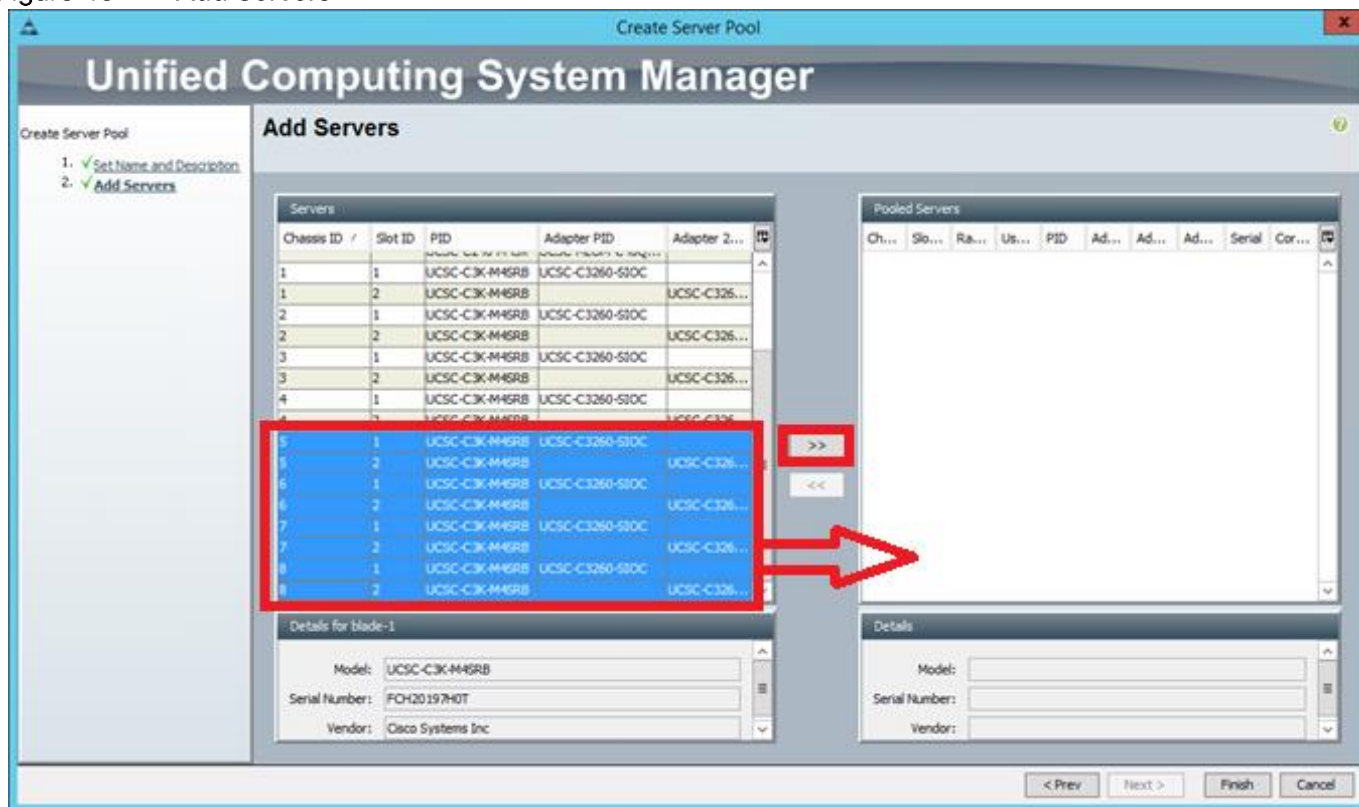
5. Enter `s3260` as the server pool name, as shown in **Error! Reference source not found.**
6. (Optional) Enter a description for the server pool.
7. Click `Next` to add the servers.

Figure 44 Setting Name and Description of Server Pool

The screenshot shows a window titled "Create Server Pool" from the "Unified Computing System Manager". The main heading is "Set Name and Description". On the left, a sidebar shows the progress: "1. ✓ Set Name and Description" and "2. Add Servers". The main area has two text input fields: "Name: S3260" and "Description: 8 servers in 4 Cisco UCS S3260 chassis used as indexers". At the bottom, there are four buttons: "< Prev", "Next >", "Finish", and "Cancel".

8. Select all eight of the Cisco UCS S3260 server nodes to be added to the server pool and then click >> to add them to the pool, as shown in Figure 45

Figure 45 Add Servers



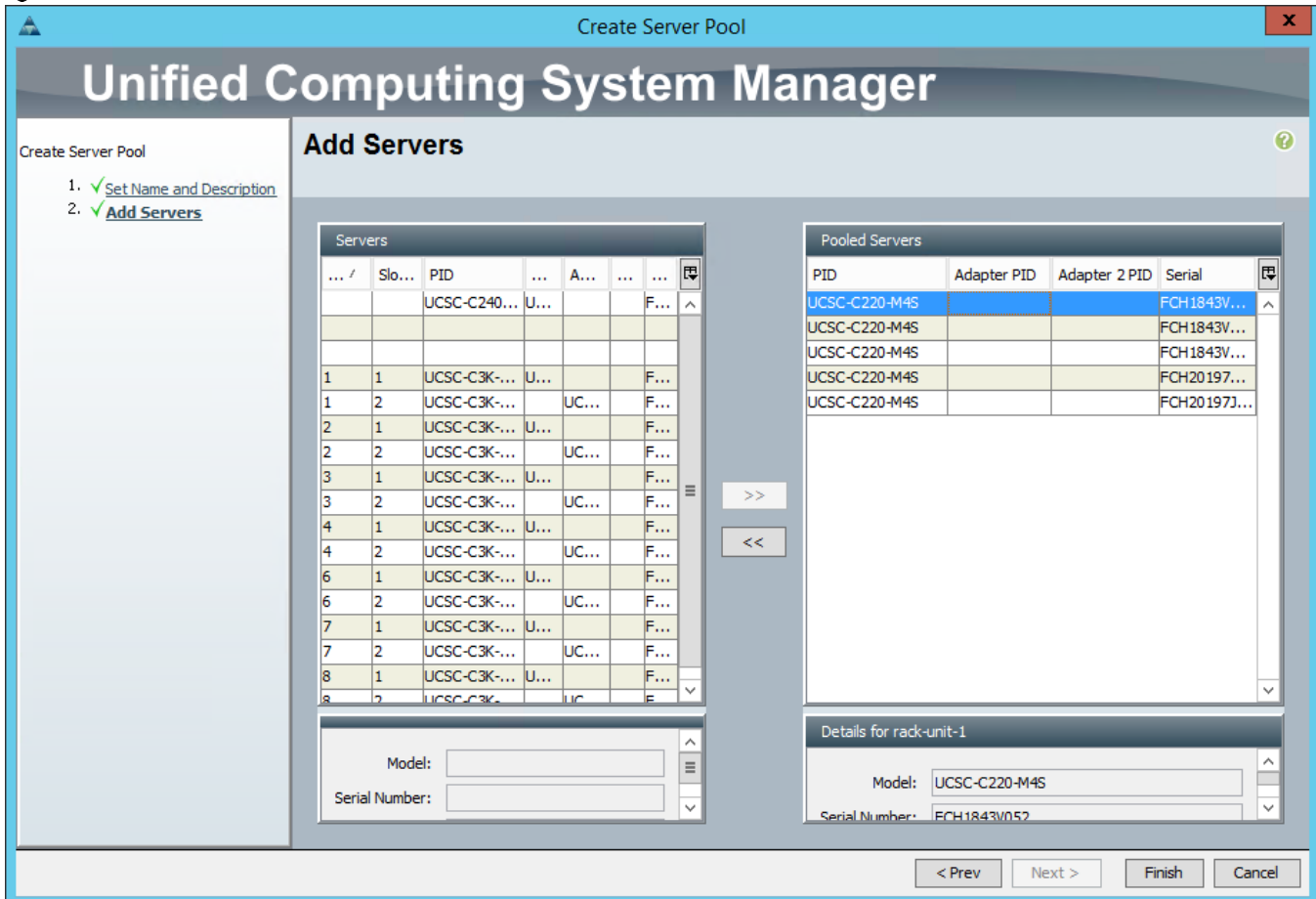
9. Click **Finish**.

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

11. Repeat Steps 1 through 7 to create another server pool named **c220**.

12. Select the five Cisco UCS 220 M4 Rack Servers to be added to the **c220** server pool and then click **>>** to add them to the pool.

Figure 46 Add Servers



13. Click **Finish**.

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

Creating Policies for Service Profile Templates

Creating Host Firmware Package Policy

Firmware management policies allow the administrator to select the corresponding packages for a given server configuration. These include adapters, BIOS, board controllers, FC adapters, HBA options, and storage controller properties as applicable.

To create a firmware management policy for a given server configuration using the Cisco UCS Manager GUI, complete the following steps:

1. Select the **Servers** tab in the left pane in the UCS Manager GUI.
2. Select **Policies** → **root**.
3. Right-click **Host Firmware Packages**.
4. Select **Create Host Firmware Package**.

5. Enter `ucs_FW_3_1_2b_C` for the host firmware package name, as shown in Figure 47. Name the Host Firmware Package appropriately to include the actual firmware version and package.
6. Click the `Simple` radio button to configure the host firmware package.
7. Select the appropriate `Rack Package` that has been installed.
8. Click `OK` to complete creating the management firmware package.
9. Click `OK`.

Figure 47 Creating Host Firmware Package

Create Host Firmware Package

Name:

Description:

How would you like to configure the Host Firmware Package? Simple Advanced

Blade Package:

Rack Package:

Excluded Components:

- Adapter
- BIOS
- CIMC
- Board Controller
- Flex Flash Controller
- GPUs
- FC Adapters
- HBA Option ROM
- Host NIC
- Host NIC Option ROM
- Local Disk
- PSU
- SAS Expander Regular Firmware
- SAS Expander
- Storage Controller
- Storage Controller Onboard Device
- Storage Controller Onboard Device Cpld
- Storage Device Bridge

OK Cancel

Creating QoS Policies

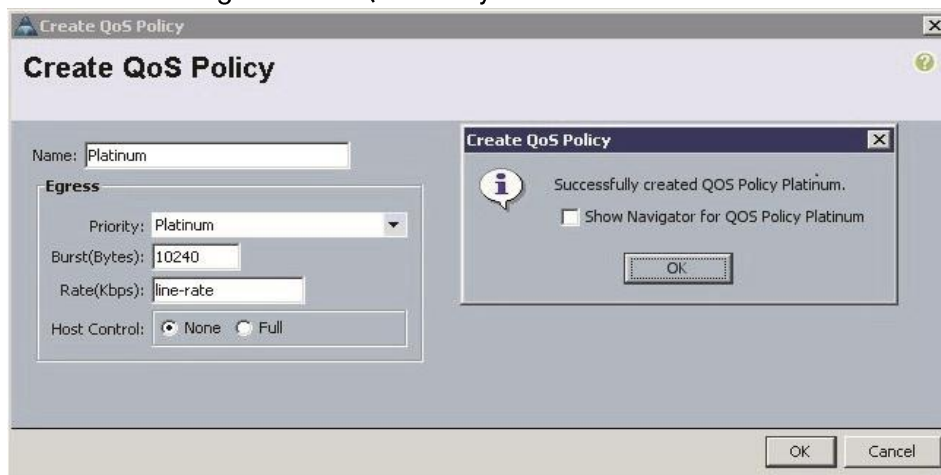
To create the QoS policy for a given server configuration using the Cisco UCS Manager GUI, complete the following steps:

Platinum Policy

1. Select the `LAN` tab in the left pane in the UCS Manager GUI.

2. Select `Policies > root`.
3. Right-click `QoS Policies`.
4. Select `Create QoS Policy`.
5. Enter `Platinum` as the name of the policy, as shown in Figure 48
6. Select `Platinum` from the `Priority` drop down menu.
7. Keep the `Burst (Bytes)` field as default (10240).
8. Keep the `Rate (Kbps)` field as default (line-rate).
9. Keep `Host Control` radio button as default (none).
10. Once the pop-up window appears, click `OK` to complete the creation of the policy.

Figure 48 Creating Platinum QoS Policy



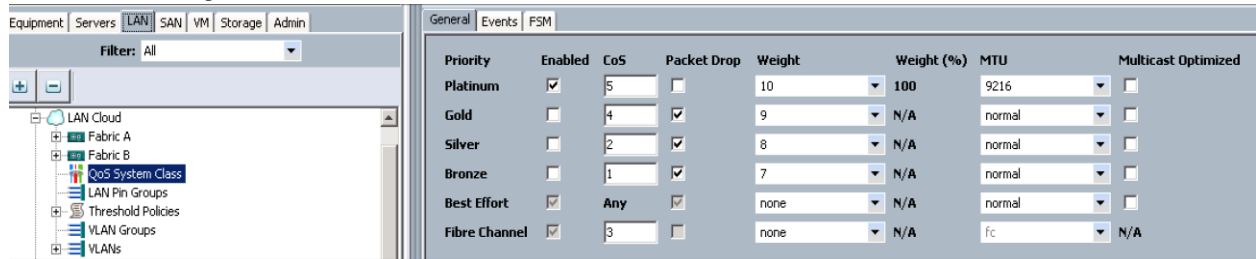
Setting Jumbo Frames

To set up jumbo frames and enable QoS, complete the following steps:

1. Select the `LAN` tab in the left pane in the UCS Manager GUI.
2. Select `LAN Cloud → QoS System Class`, as shown in Figure 49
3. In the right pane, select the `General` tab.
4. In the `Platinum` row, enter `9216` for `MTU`.
5. Check the `Enabled` check box next to `Platinum`.
6. In the `Best Effort` row, select `none` for `weight`.
7. In the `Fiber Channel` row, select `none` for `weight`.

8. Click `Save Changes`.
9. Click `OK`.

Figure 49 Setting Jumbo Frames



Creating Local Disk Configuration Policy

To create local disk configuration in the Cisco UCS Manager GUI, complete the following steps:

1. Select the `Servers` tab on the left pane in the UCS Manager GUI.
2. Go to `Policies` → `root`.
3. Right-click `Local Disk Config Policies`.
4. Select `Create Local Disk Configuration Policy`.
5. Enter `localdisk` for the local disk configuration policy name, as shown in Figure 50
6. Change the `Mode` to `Any Configuration`. Check the `Protect Configuration` box.
7. Keep the `FlexFlash State` field as default (`Disable`).
8. Keep the `FlexFlash RAID Reporting State` field as default (`Disable`).
9. Click `OK` to complete the creation of the local disk configuration policy.
10. Click `OK`.

Figure 50 Configuring Local Disk Policy

Create Local Disk Configuration Policy

Name:

Description:

Mode:

Protect Configuration:

If **Protect Configuration** is set, the local disk configuration is preserved if the service profile is disassociated with the server. In that case, a configuration error will be raised when a new service profile is associated with that server if the local disk configuration in that profile is different.

FlexFlash

FlexFlash State: Disable Enable

If **FlexFlash State** is disabled, SD cards will become unavailable immediately. Please ensure SD cards are not in use before disabling the FlexFlash State.

FlexFlash RAID Reporting State: Disable Enable

Creating Server BIOS Policy

The BIOS policy feature in Cisco UCS automates the BIOS configuration process. The traditional method of setting the BIOS is done manually and is often error-prone. By creating a BIOS policy and assigning the policy to a server or group of servers, you can enable transparency within the BIOS settings configuration.



Note: BIOS settings can have a significant performance impact, depending on the workload and the applications. The BIOS settings listed in this section is for configurations optimized for best performance which can be adjusted based on the application, performance, and energy efficiency requirements.

To create a server BIOS policy using the Cisco UCS Manager GUI, complete the following steps:

1. Select the `Servers` tab in the left pane in the UCS Manager GUI.
2. Select `Policies` → `root`.
3. Right-click `BIOS Policies`.

4. Select Create BIOS Policy.
5. Enter biospolicy for the BIOS policy name.
6. Change the BIOS settings as shown in the following figures:

Figure 51 Creating Server BIOS Policy – Main Screen

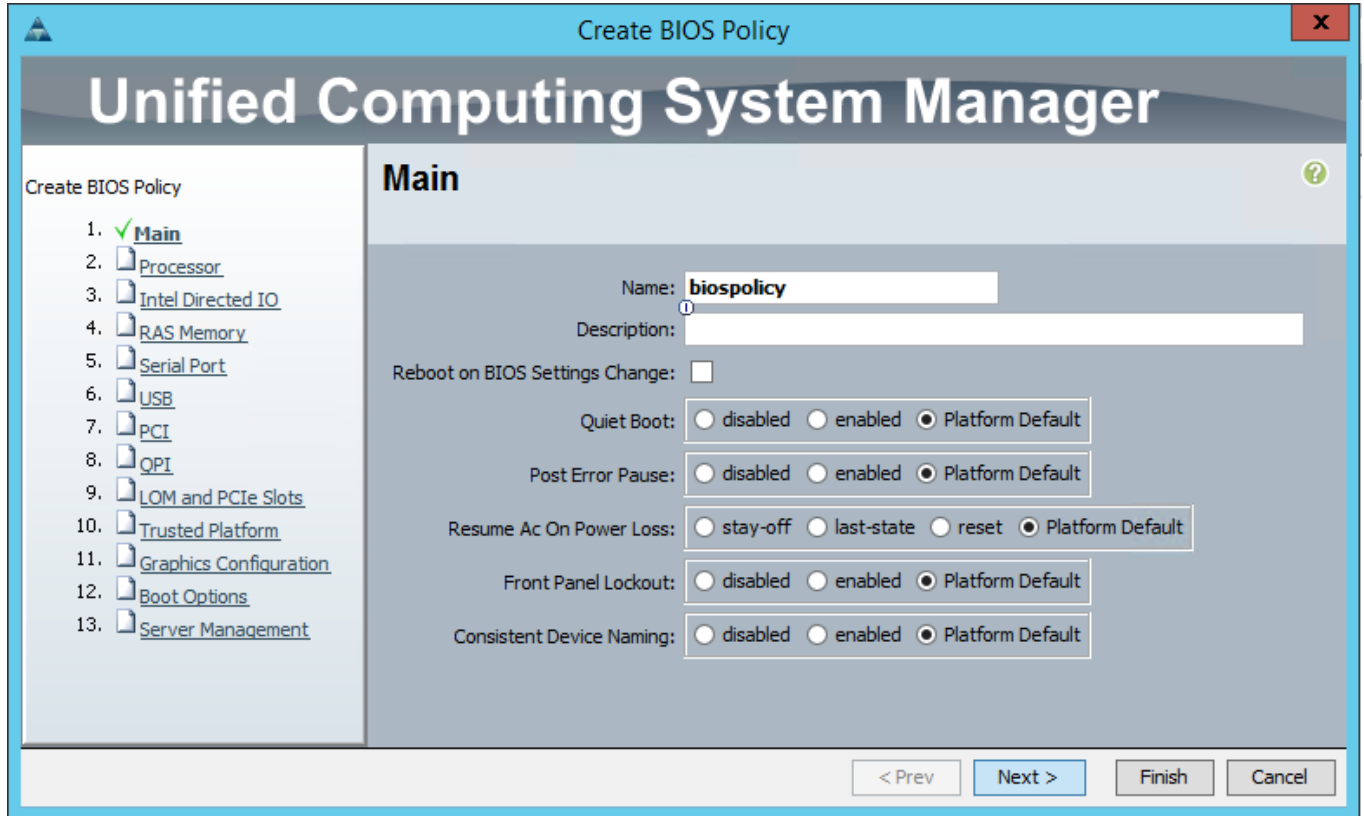
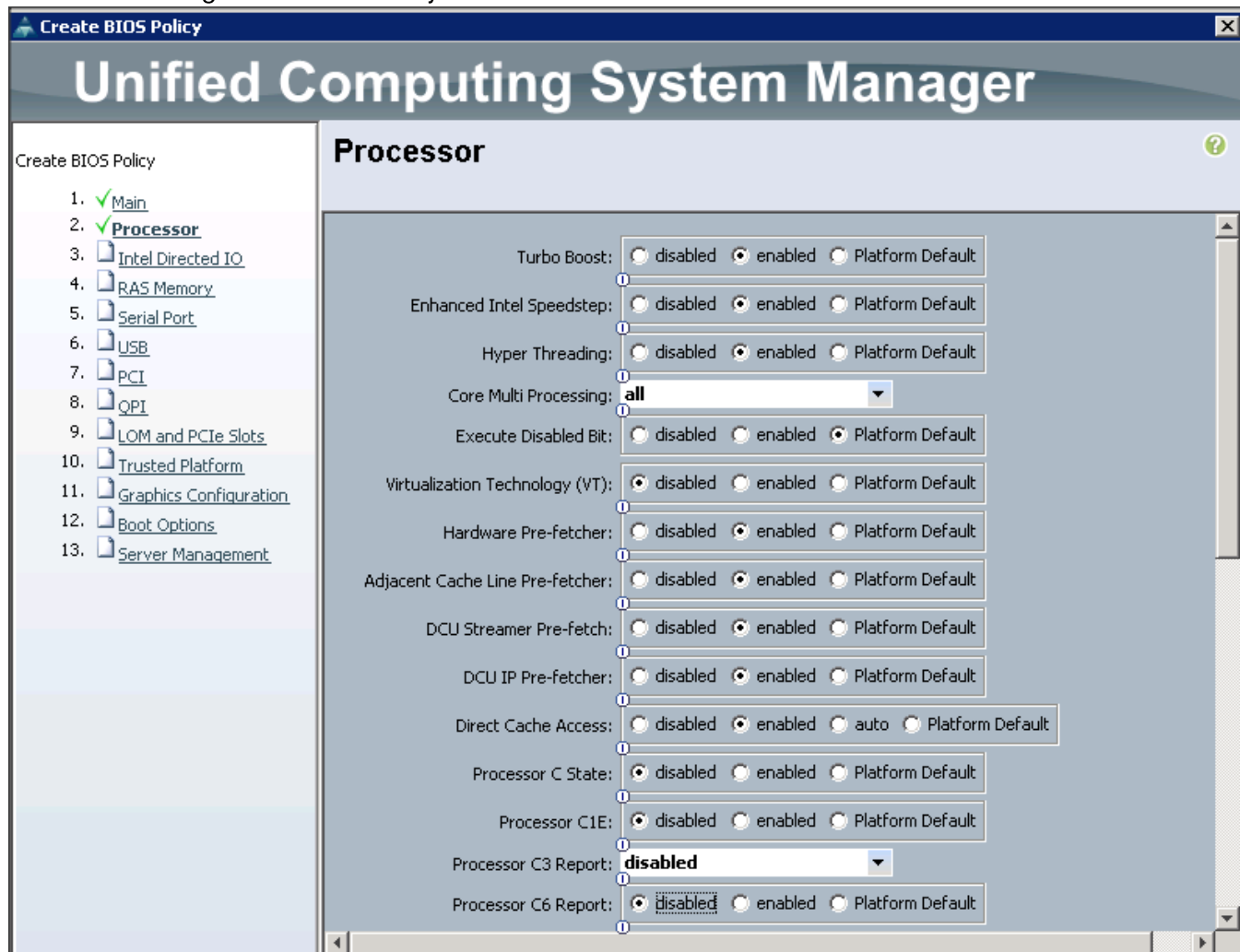


Figure 52 Creating Server BIOS Policy for Processor



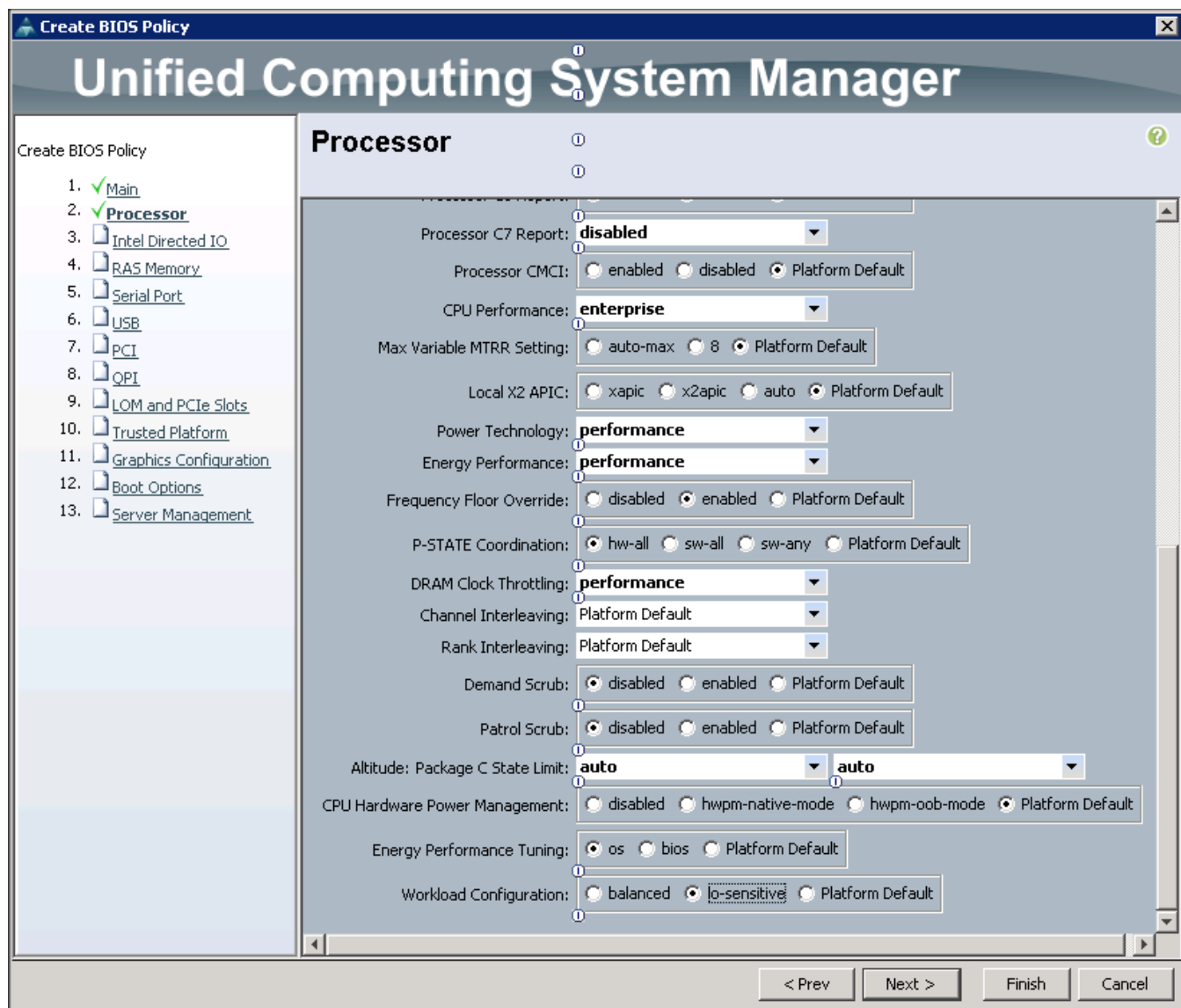
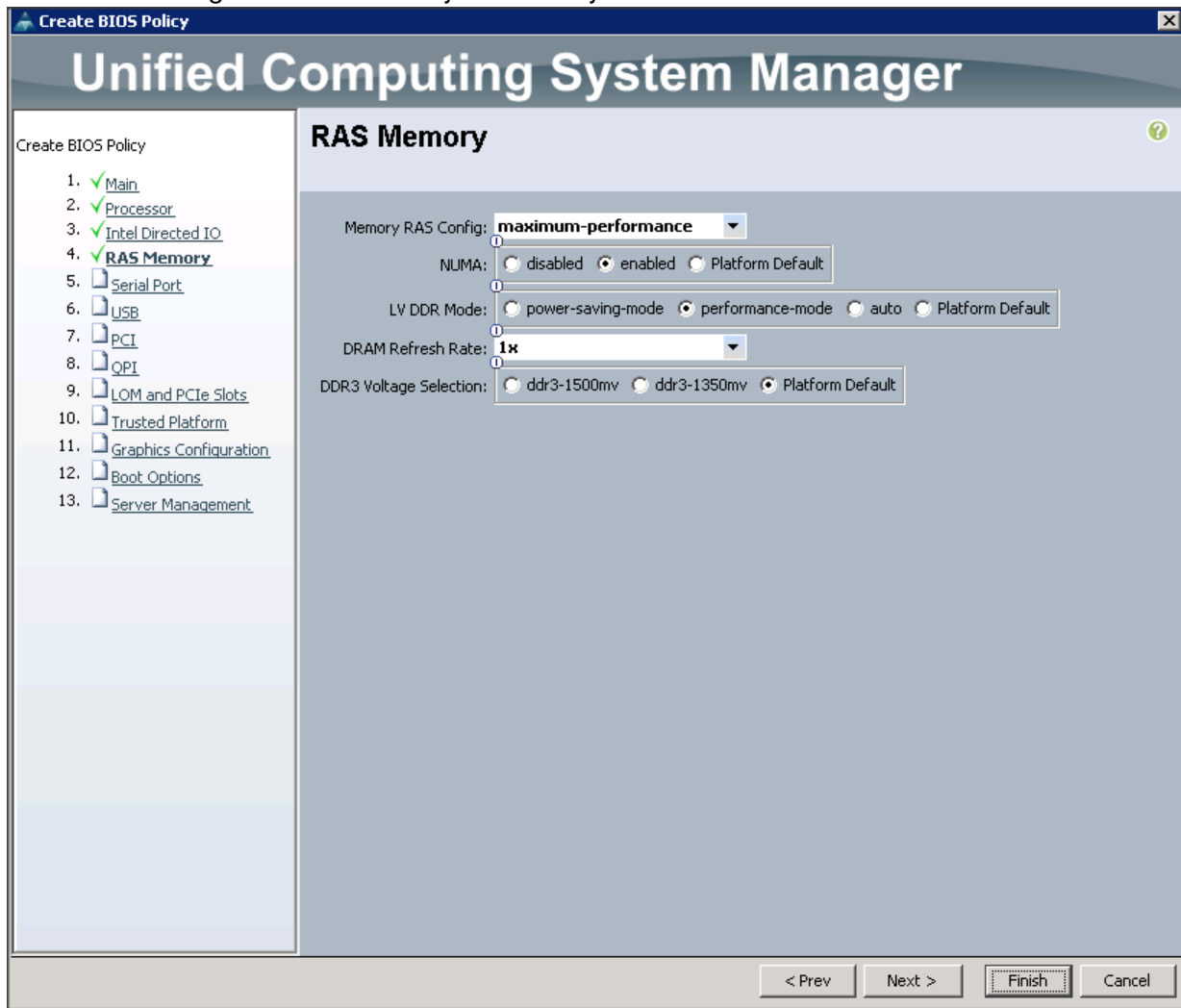


Figure 53 Creating Server BIOS Policy for Intel Directed IO



Figure 54 Creating Server BIOS Policy for Memory



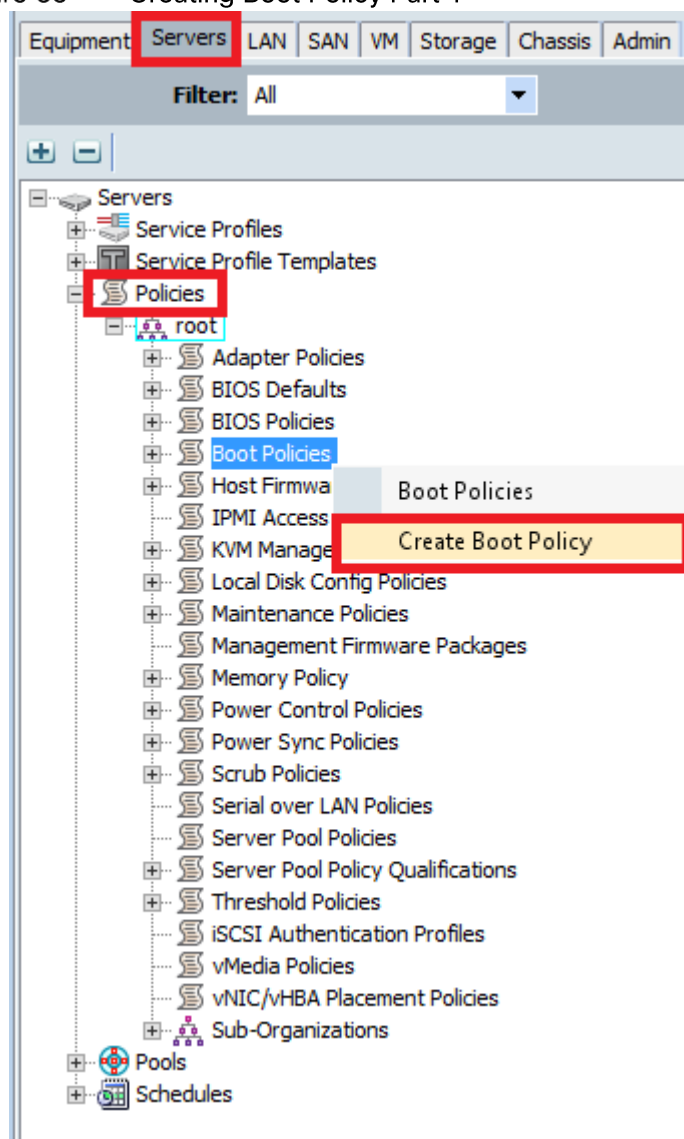
7. No changes need to be made in the remaining sections (5-13). Click **Finish** to complete creating the BIOS policy.
8. Click **OK**.

Creating Boot Policies

To create boot policies within the Cisco UCS Manager GUI, complete the following steps:

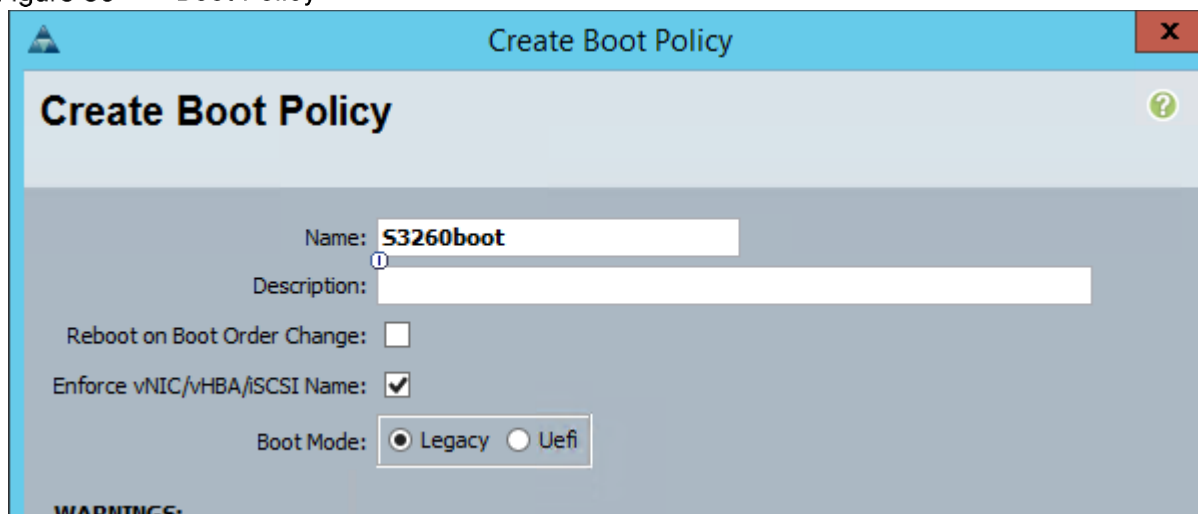
1. Select the **Servers** tab in the left pane in the UCS Manager GUI, as shown in Figure 55
2. Select **Policies** → **root**.
3. Right-click the **Boot Policies**.
4. Select **Create Boot Policy**.

Figure 55 Creating Boot Policy Part 1



5. Enter `S3260boot` for the boot policy name. Figure 56
6. (Optional) Enter a description for the boot policy.
7. Keep the `Reboot on Boot Order Change` check box unchecked.
8. Keep the `Enforce vNIC/vHBA/iSCSI Name` check box checked.
9. Keep `Boot Mode` as the default (Legacy).

Figure 56 Boot Policy



Create Boot Policy

Name:

Description:

Reboot on Boot Order Change:

Enforce vNIC/vHBA/iSCSI Name:

Boot Mode: Legacy Uefi

WARNINGS:

10. Expand `Local Devices` and select `Add Local LUN`.
11. In the `Add Local LUN Image Path` window, select `Primary` and enter the `Boot_SSD` for the LUN Name.



Note: The LUN name must match with the LUN name created earlier during the storage profile creation step.

Figure 57 Add Local LUN Image Path

Add Local LUN Image Path

Type: Primary Secondary Any

LUN Name:

OK Cancel

12. Expand **Local Devices** → **Add CD/DVD** and select **Add CD/DVD**.

13. Expand **vNICs** and select **Add LAN Boot** and enter **eth0**.

Figure 58 Add Boot Policy

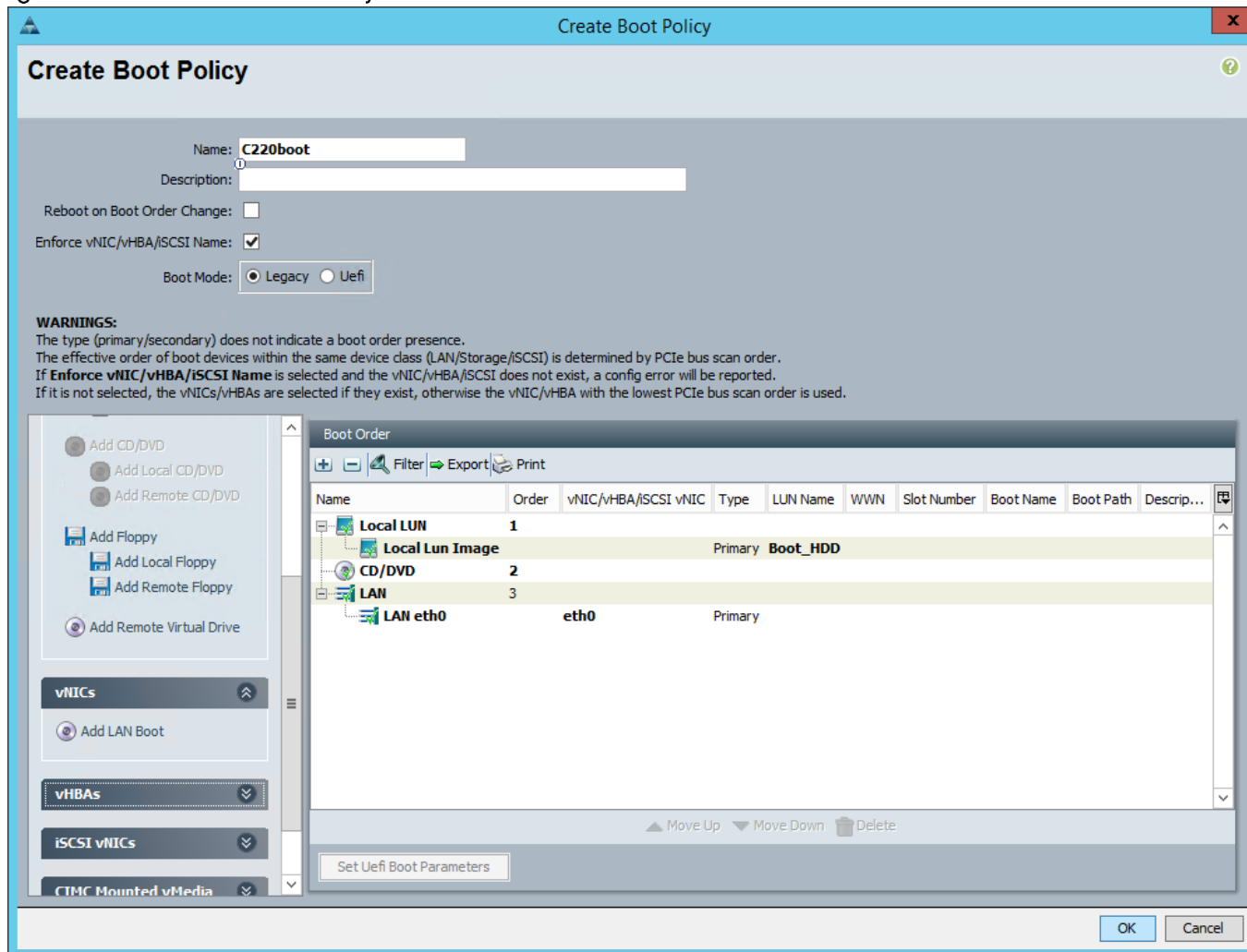
Name	Order	vNIC/vHBA/iSCSI vNIC	Type	LUN Name	WWN	Slot Number
Local LUN	1					
Local Lun Image			Primary	Boot_SSD		
CD/DVD	2					
LAN	3					
LAN eth0		eth0	Primary			

14. Click **OK** to add the boot policy.

15. Click **OK**.

16. Repeat Steps 1-15 to create a boot policy called **C220boot** with a LUN Name of **Boot_HDD**.

Figure 59 Create Boot Policy

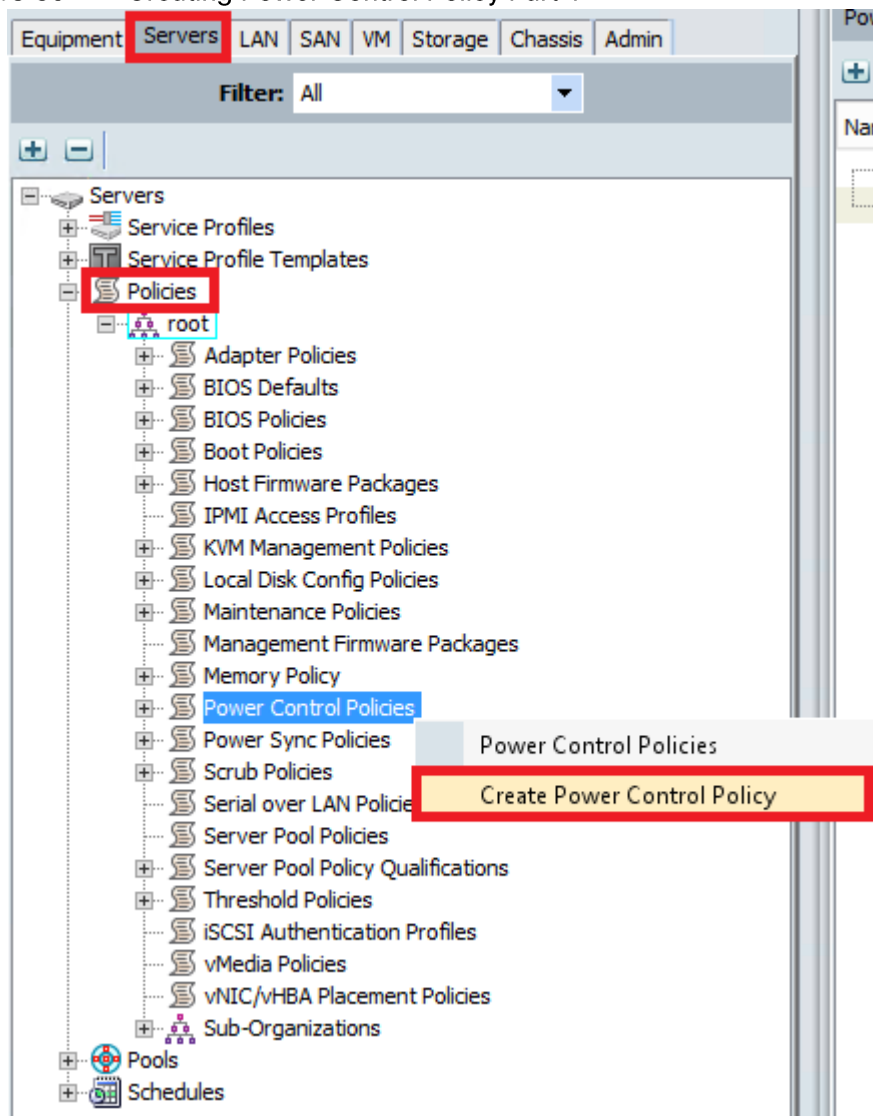


Creating Power Control Policy

To create the Power Control policies within the Cisco UCS Manager GUI, complete the following steps:

1. Select the `Servers` tab in the left pane in the UCS Manager GUI.
2. Select `Policies` → `root`.
3. Right-click `Power Control Policies`.
4. Select `Create Power Control Policy`, as shown in Figure 60

Figure 60 Creating Power Control Policy Part 1



5. Enter `power` for the power control policy name, as shown in Figure 61
6. (Optional) Enter a description for the power control policy.
7. Select `Performance` for the `Fan Speed Policy`.
8. Select `No cap` for the `Power Capping` selection.
9. Click `OK` to create the power control policy.
10. Click `OK`.

Figure 61 Creating Power Control Policy Part 2

Create Power Control Policy

Name: **power**

Description:

Fan Speed Policy: **Performance**

Power Capping

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

No Cap cap

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

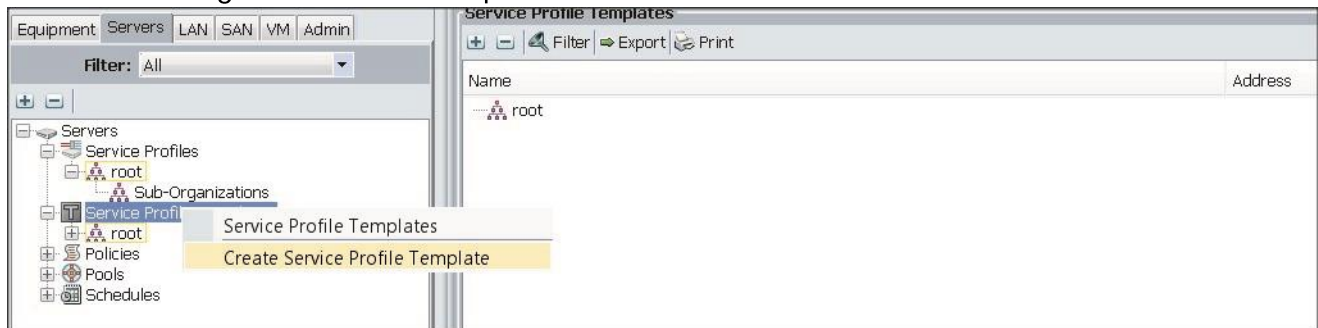
OK Cancel

Creating a Service Profile Template for Cisco S3260 Storage Servers

To create a service profile template, complete the following steps:

1. Select the `Servers` tab in the left pane in the UCS Manager GUI.
2. Right-click `Service Profile Templates`.
3. Select `Create Service Profile Template`. (Figure 62) The `Create Service Profile Template` window appears.

Figure 62 Creating Service Profile Template



4. Enter `S3260` for the service profile template name, as shown in Figure 63 .

5. Click the `Updating Template` radio button.
6. In the `UUID` section, select `Hardware Default` as the `UUID` Assignment.
7. Click `Next` to continue to the next section.

Figure 63 Identify Service Profile Template

Create Service Profile Template

Identify Service Profile Template

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

Name:

The template will be created in the following organization. Its name must be unique within this organization.
Where: **org-root**

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

Type: Initial Template Updating Template

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

UUID

UUID Assignment:

The UUID assigned by the manufacturer will be used.
Note: This UUID will not be migrated if the service profile is moved to a new server.

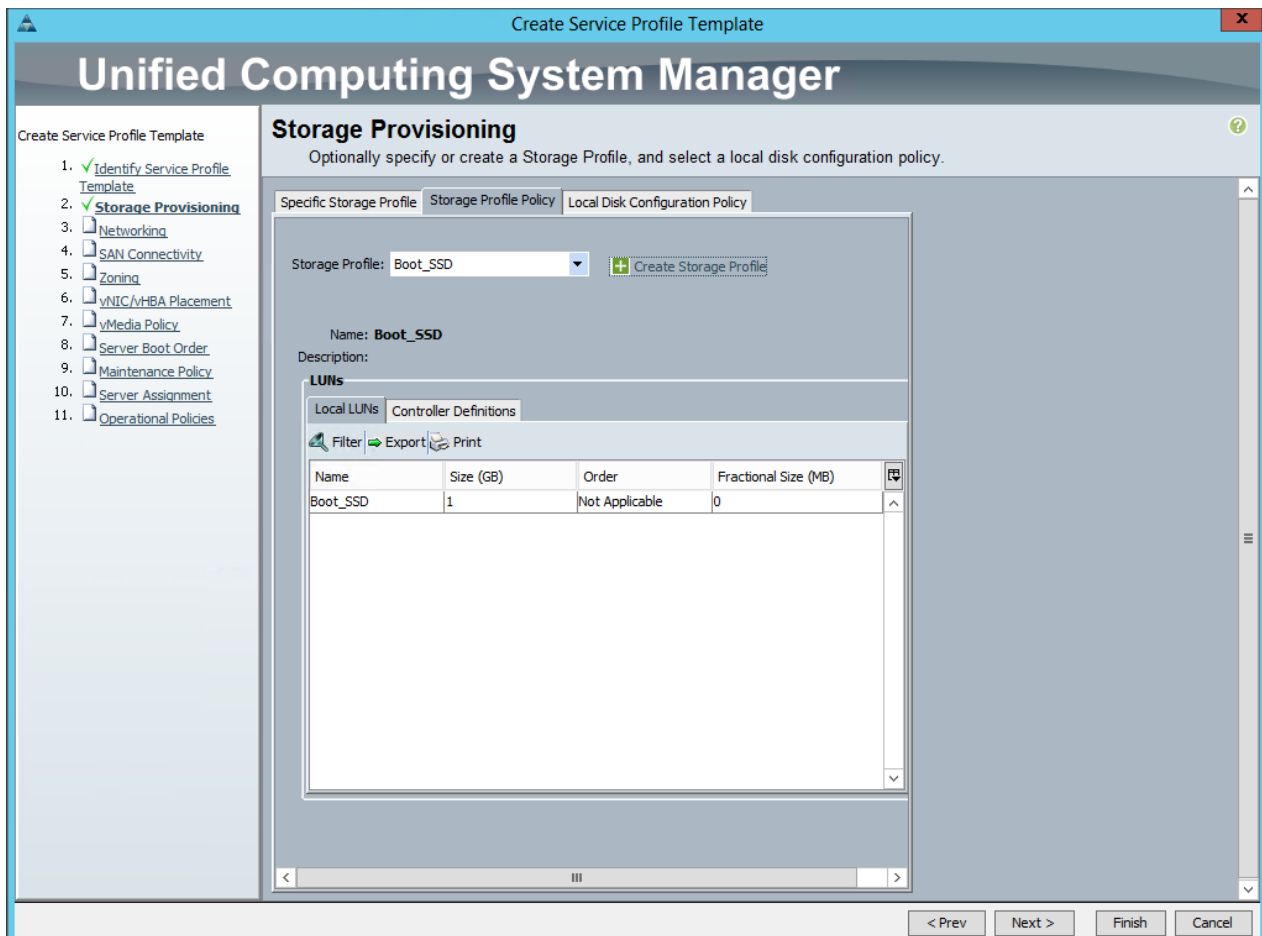
Optionally enter a description for the profile. The description can contain information about when and where the service profile should be used.

< Prev Next > Finish Cancel

Configuring the Storage Provisioning for the Template

To configure storage policies for the template, complete the following steps:

1. Click on the `Storage Profile Policy` tab.
2. Select `Boot_SSD` from the `Storage Profile` drop down list.



3. Click on the **Local Disk Configuration Policy** tab.
4. Select **localdisk** for the Local Storage. (Figure 64)
5. Click **Next** to continue to the next section.

Figure 64 Configuring Storage Settings

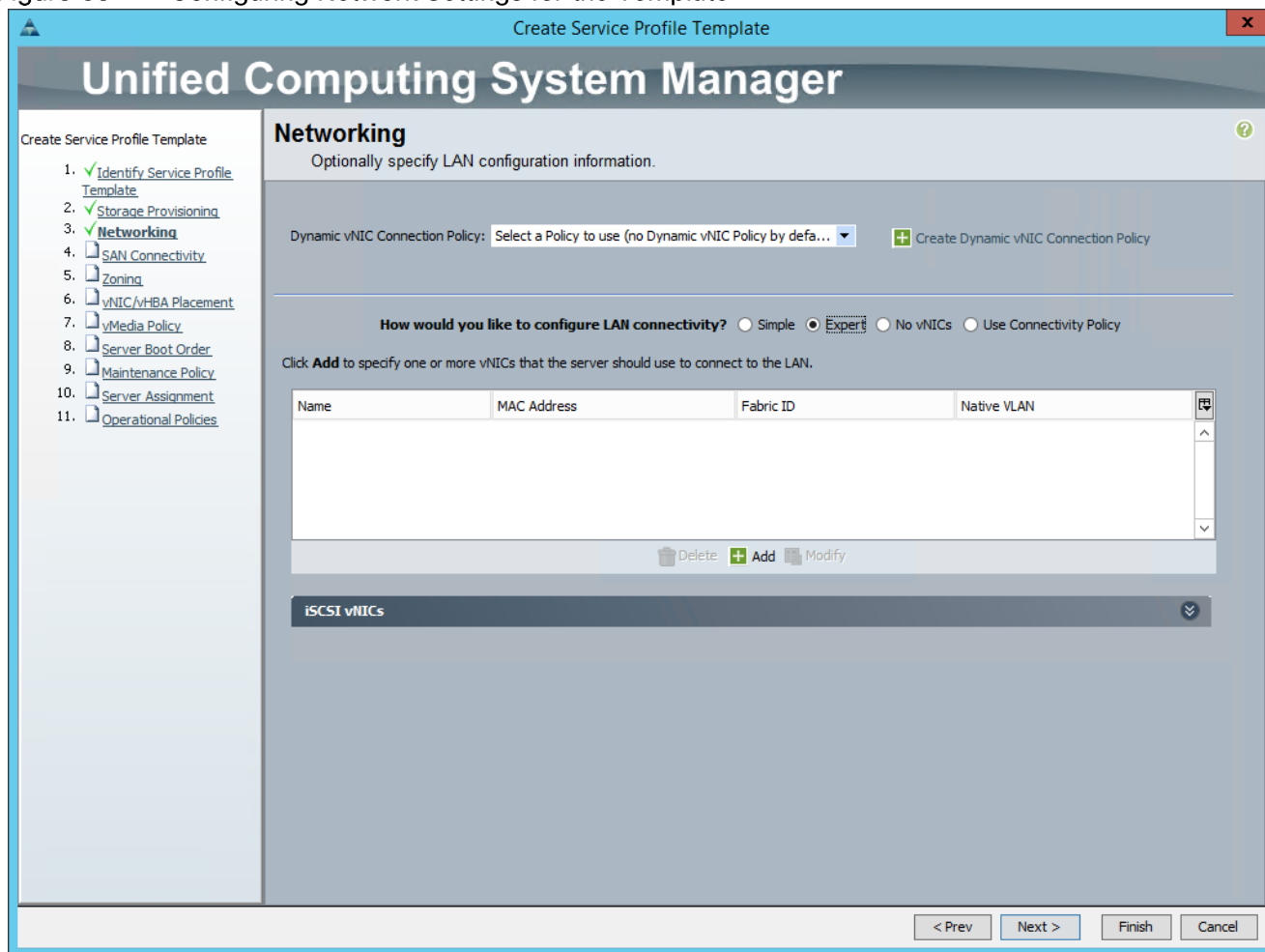


Configuring Network Settings for the Template

To configure the network settings for the template, complete the following steps:

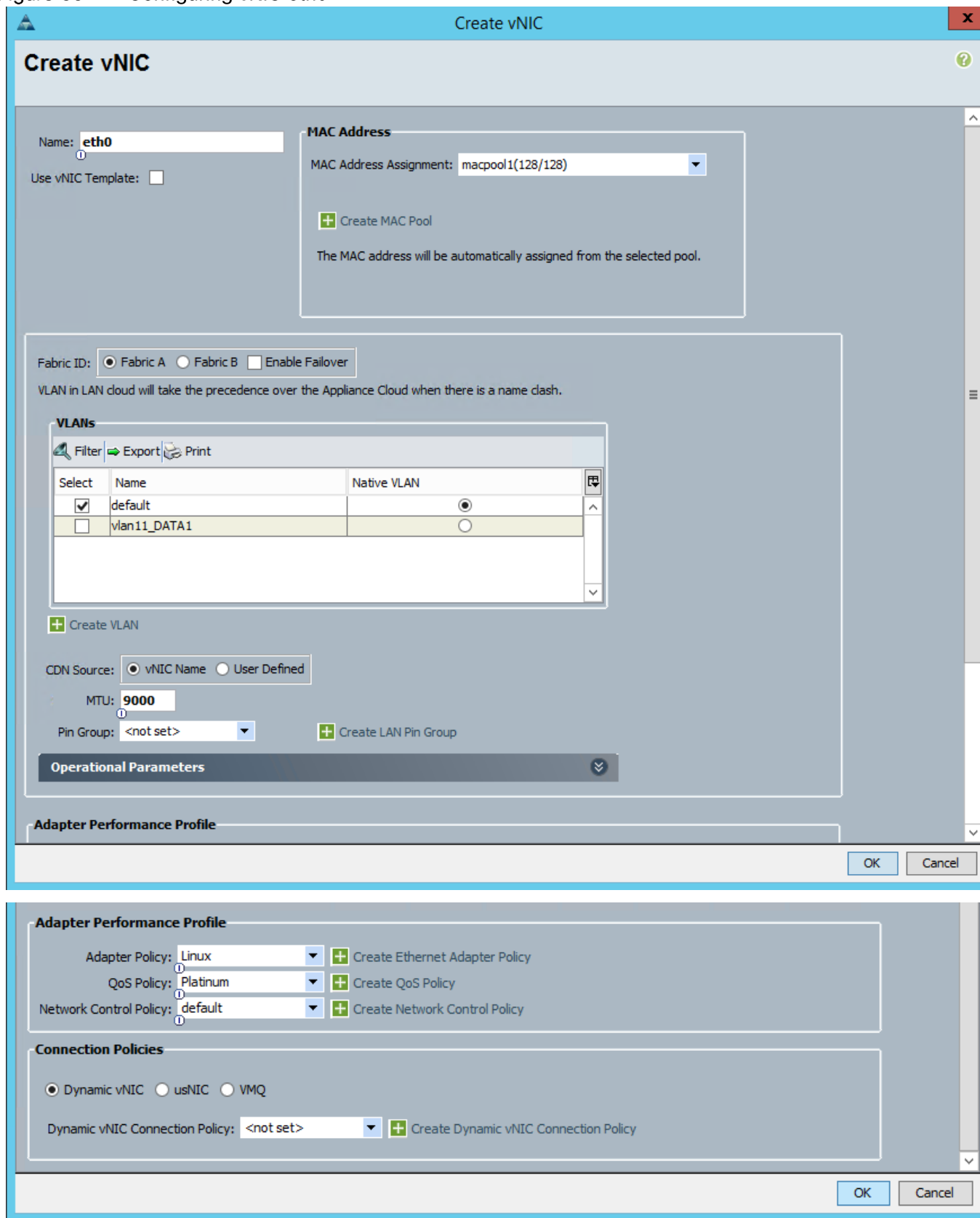
1. Keep the `Dynamic vNIC Connection Policy` field set to default. (Figure 65)
2. Click the `Expert` radio button for the option, `How would you like to configure LAN connectivity?`

Figure 65 Configuring Network Settings for the Template



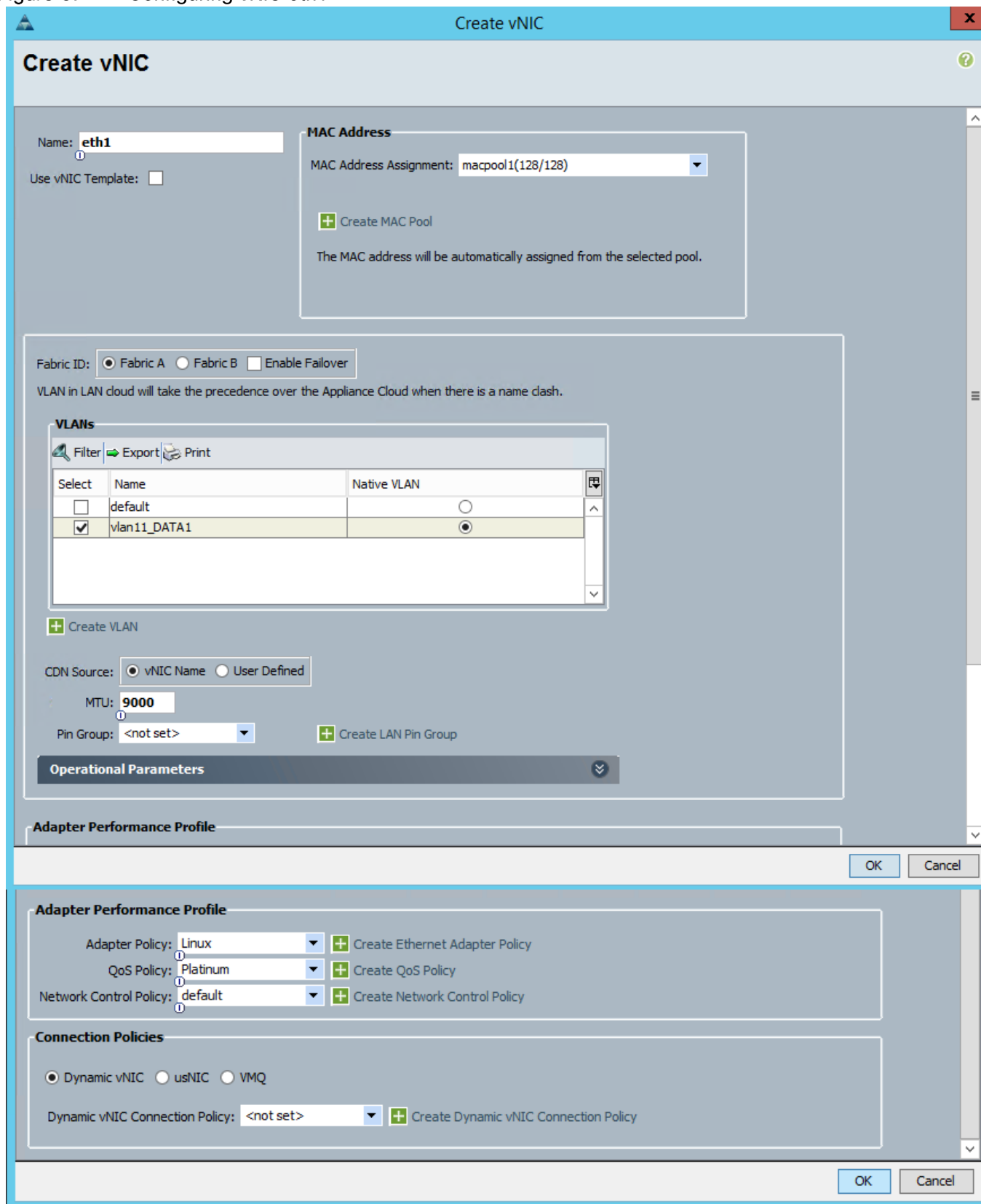
3. Click **Add** to add a vNIC to the template.
4. The **Create vNIC** window displays. Name the vNIC `eth0`. Figure 66
5. Select `macpool1` in the **Mac Address Assignment pool**.
6. For **Fabric ID**, click the `Fabric A` radio button and check the `Enable failover` check box.
7. Check the `default` check box for VLANs and click the `Native VLAN` radio button.
8. Set **MTU size** to `9000`.
9. In the **Adapter Performance Profile** section, set **Adapter Policy** to `Linux`. Set **QoS Policy** to `Platinum`. Keep **Network Control Policy** as `Default`.
10. In the **Connection Policies** section, keep the **Connection Policies** set at `Dynamic vNIC`. Keep the **Dynamic vNIC Connection Policy** as `<not set>`.
11. Click **OK**.

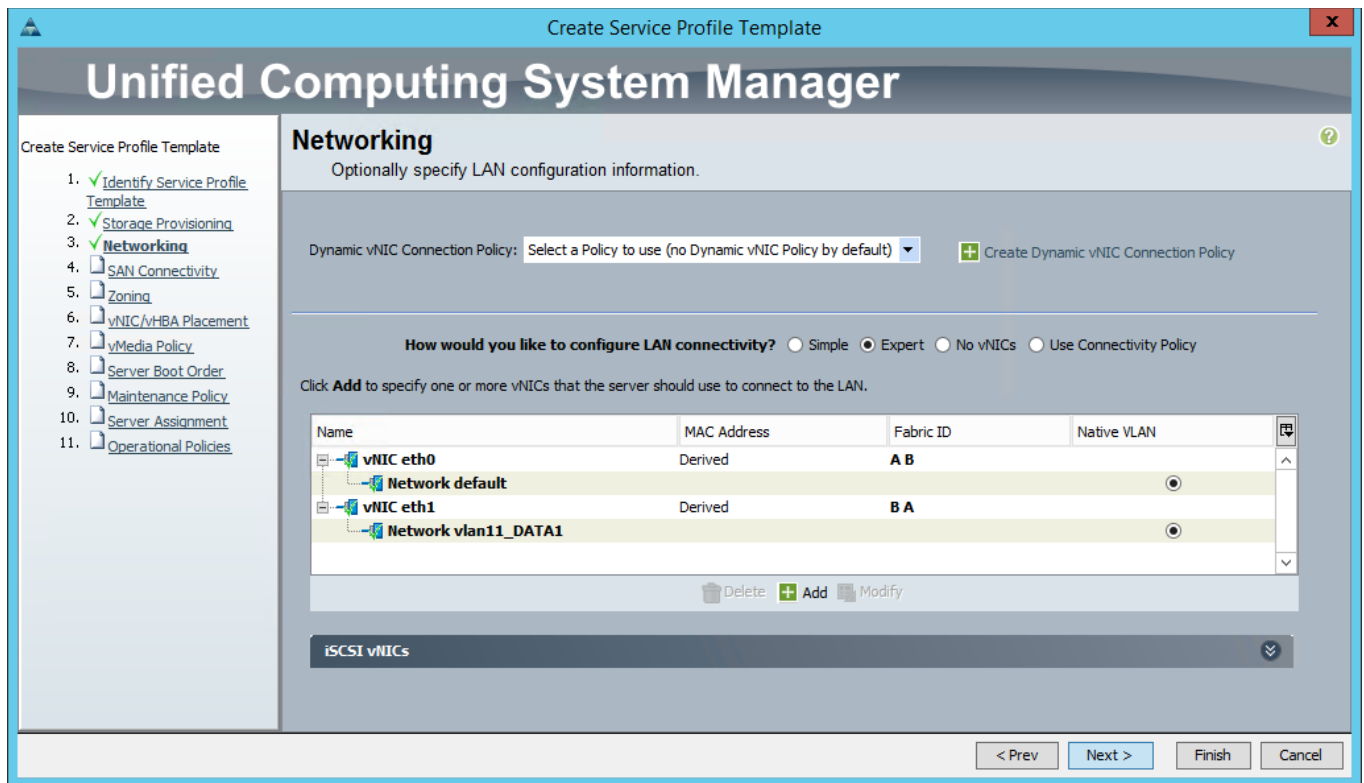
Figure 66 Configuring vNIC eth0



12. Click `Add` to add the second vNIC to the template.
13. The `Create vNIC` window appears. Name the vNIC `eth1`. **Figure 67**
14. Select `macpool1` in the `Mac Address Assignment pool`.
15. Click the `Fabric B` radio button and check the `Enable failover` check box for the Fabric ID.
16. Check the `vlan11_DATA1` check box for VLANs, and click the `Native VLAN` radio button
17. Set `MTU size` as `9000`.
18. In the `Adapter Performance Profile` section, set `Adapter Policy` as `Linux`. Set `QoS Policy` as `Platinum`. Set `Network Control Policy` as `Default`.
19. In the `Connection Policies` section, keep the `Connection Policies` as `Dynamic vNIC`. Keep the `Dynamic vNIC Connection Policy` as `<not set>`.
20. Click `OK`.

Figure 67 Configuring vNIC eth1



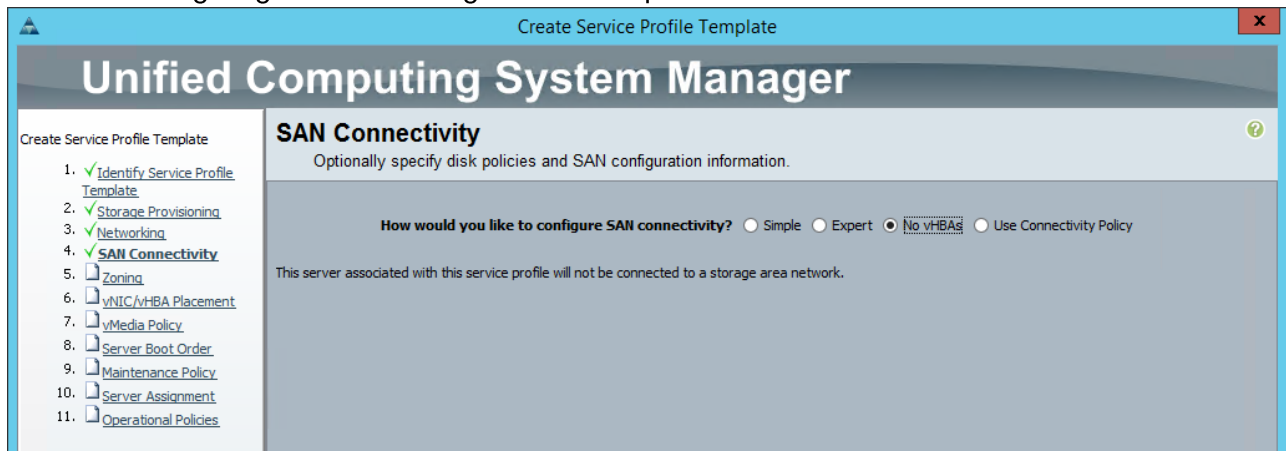


Configuring SAN Connectivity for the Template

To configure SAN connectivity, complete the following steps:

1. For **How would you like to configure SAN connectivity?**, select **no vHBAs**. Figure 68
2. Click **Next** to go to the next section.
3. Zoning information is not specified. Click **Next** to go to the next section.

Figure 68 Configuring Network Settings for the Template

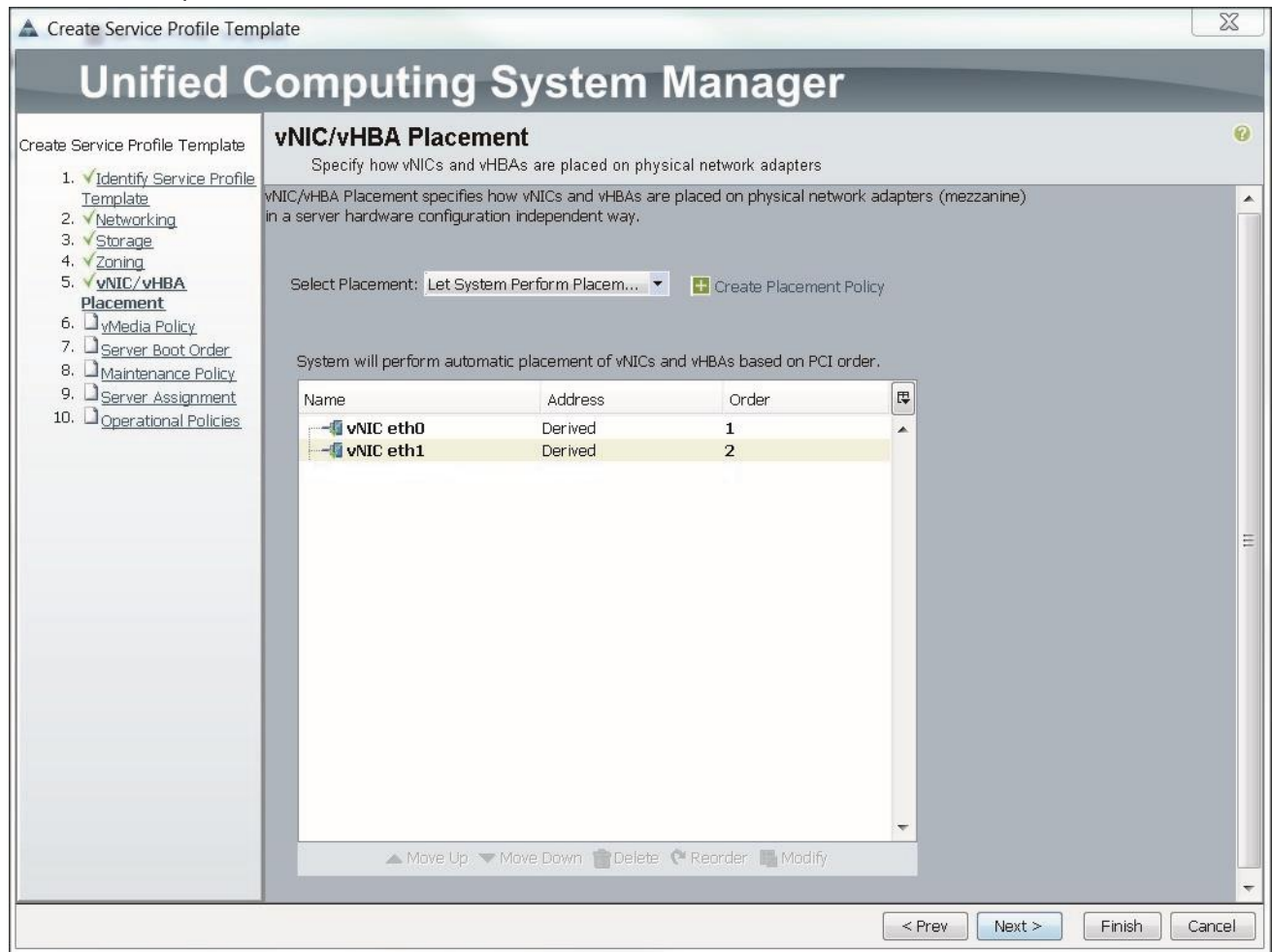


Configuring vNIC/vHBA Placement Policy for the Template

To configure the vNIC/vHBA placement policy, complete the following steps:

1. Keep the default option for the `Select Placement` field. Figure 69
2. Select `eth0` and `eth1` and assign the vNICs in the following order: `eth0`, `eth1`.
3. Review to make sure that both vNICs were assigned in the appropriate order.
4. Click `Next` to continue to the vMedia Policy section. Click `Next` again to continue to the Server Boot Order section.

Figure 69 vNIC/vHBA Placement



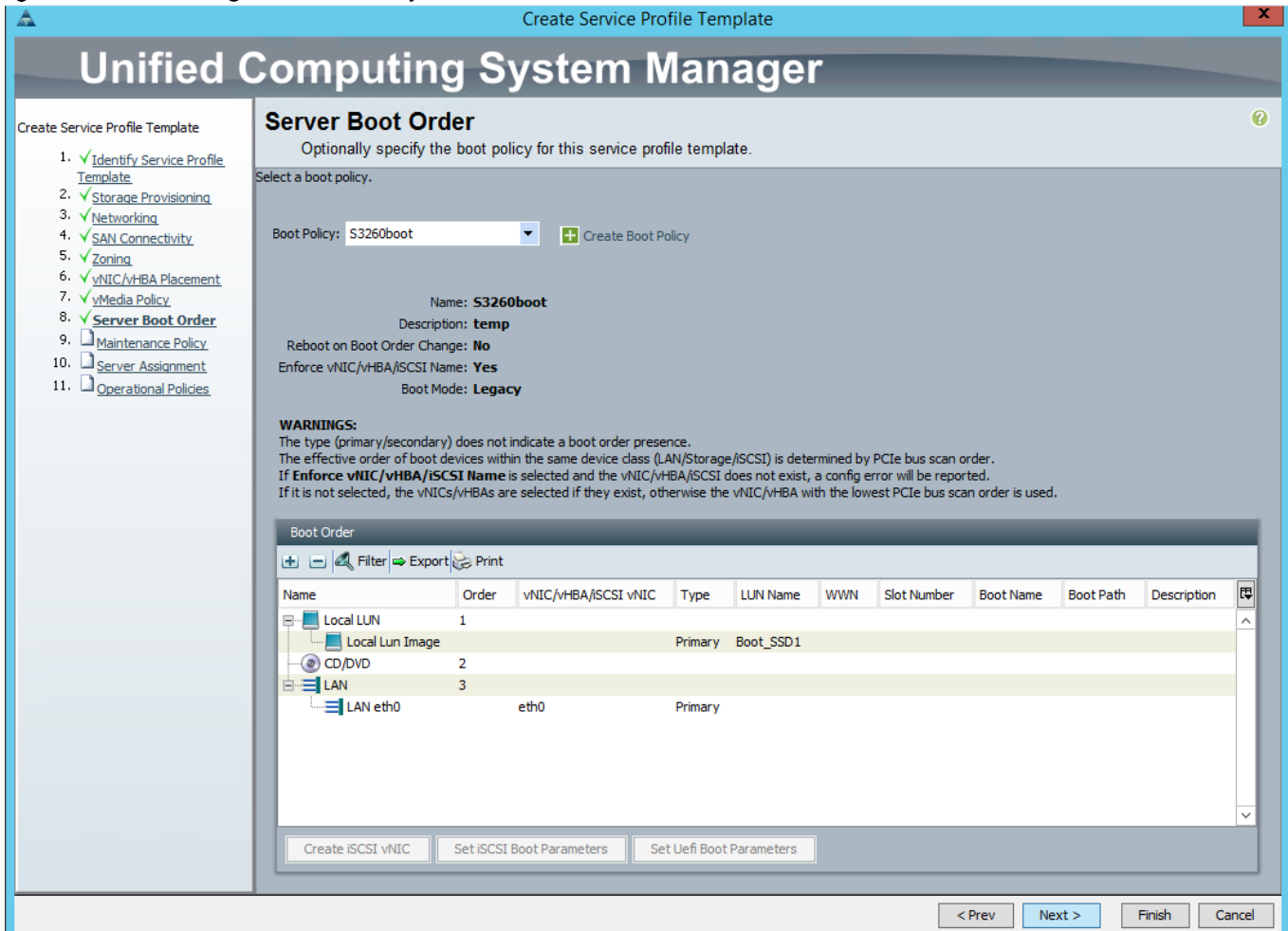
Configuring Server Boot Order for the Template

To set the boot order for the servers, complete the following steps:

1. Select `S3260boot` in the `Boot Policy` name field. Figure 70
2. Review to make sure that all of the boot devices were created and identified.

3. Verify that the boot devices are in the correct boot sequence.
4. Click **OK**.
5. Click **Next** to continue to the **Maintenance Policy** section. No maintenance policy will be used, so click **Next** to continue to the **Server Assignment** section.

Figure 70 Configure Boot Policy



Configuring Server Assignment for the Template

To assign the servers to the pool in the **Server Assignment** window, complete the following steps:

1. Select **S3260** for the **Pool Assignment** field, as shown in Figure 71
2. Select the power state to be **Up**.
3. Keep the **Server Pool Qualification** field at default (not set).
4. Check the **Restrict Migration** check box.
5. Expand the **Firmware Management** section.

- For the `Host Firmware Package`, select `ucs_FW_3_1_2b_C` from the drop-down list.

Figure 71 Server Pool and Host Firmware Policy Selection

Create Service Profile Template

Unified Computing System Manager

Create Service Profile Template

- ✓ Identify Service Profile Template
- ✓ Storage Provisioning
- ✓ Networking
- ✓ SAN Connectivity
- ✓ Zoning
- ✓ vNIC/vHBA Placement
- ✓ vMedia Policy
- ✓ Server Boot Order
- ✓ Maintenance Policy
- ✓ **Server Assignment**
- Operational Policies

Server Assignment

Optionally specify a server pool for this service profile template.

You can select a server pool you want to associate with this service profile template.

Pool Assignment: S3260 + Create Server Pool

Select the power state to be applied when this profile is associated with the server.

Up Down

The service profile template will be associated with one of the servers in the selected pool. If desired, you can specify an additional server pool policy qualification that the selected server must meet. To do so, select the qualification from the list.

Server Pool Qualification: <not set>

Restrict Migration:

Firmware Management (BIOS, Disk Controller, Adapter)

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

Host Firmware Package: ucs_FW_3_1_2b_C + Create Host Firmware Package

< Prev Next > Finish Cancel

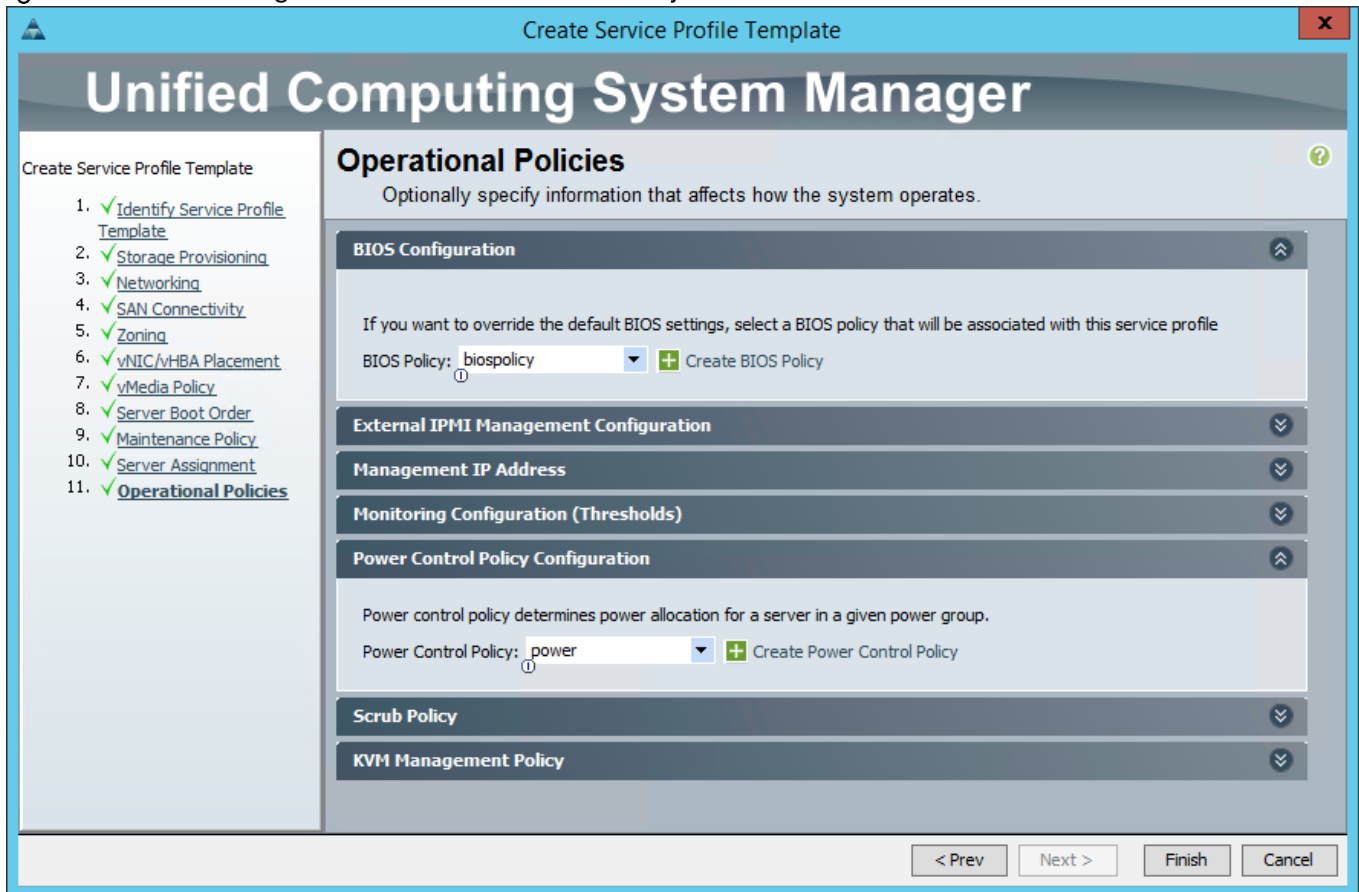
Configuring Operational Policies for the Template

In the Operational Policies Window, complete the following steps:

- In the `BIOS Configuration` section, select `biospolicy` in the `BIOS Policy` field, as shown in Figure 72
- Expand the `Power Control Policy Configuration` section. Select `power` in the `Power Control Policy` field.

3. Click **Finish** to create the service profile template.
4. Click **OK** in the pop-up window to proceed.

Figure 72 Selecting BIOS and Power Control Policy

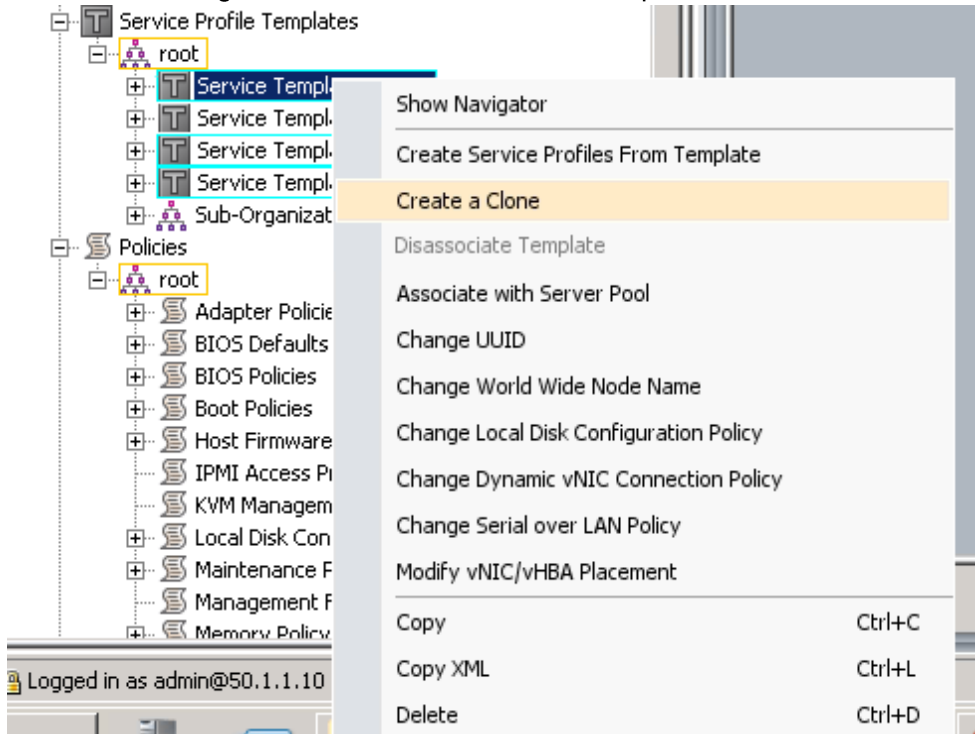


Creating a Service Profile Template for Cisco C220 M4 Servers

The Cisco UCS C220 servers need a separate service profile template. Copy the service profile template that was just created and then modify the storage profile and boot policy.

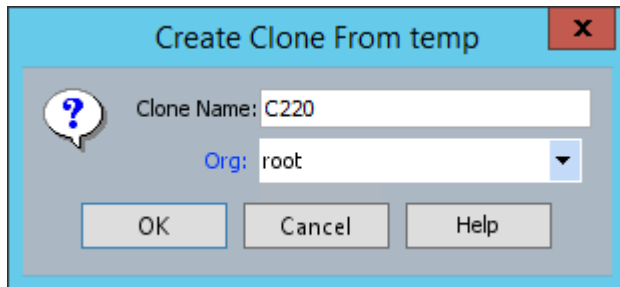
1. Select the **Servers** tab in the navigation pane.
2. Go to **Servers** → **Service Profile Templates** → **root**.
3. Right-click **Service Template S3260**.
4. Select **Create a Clone**, as shown in Figure 73 .

Figure 73 Creating a Clone of a Service Profile Template



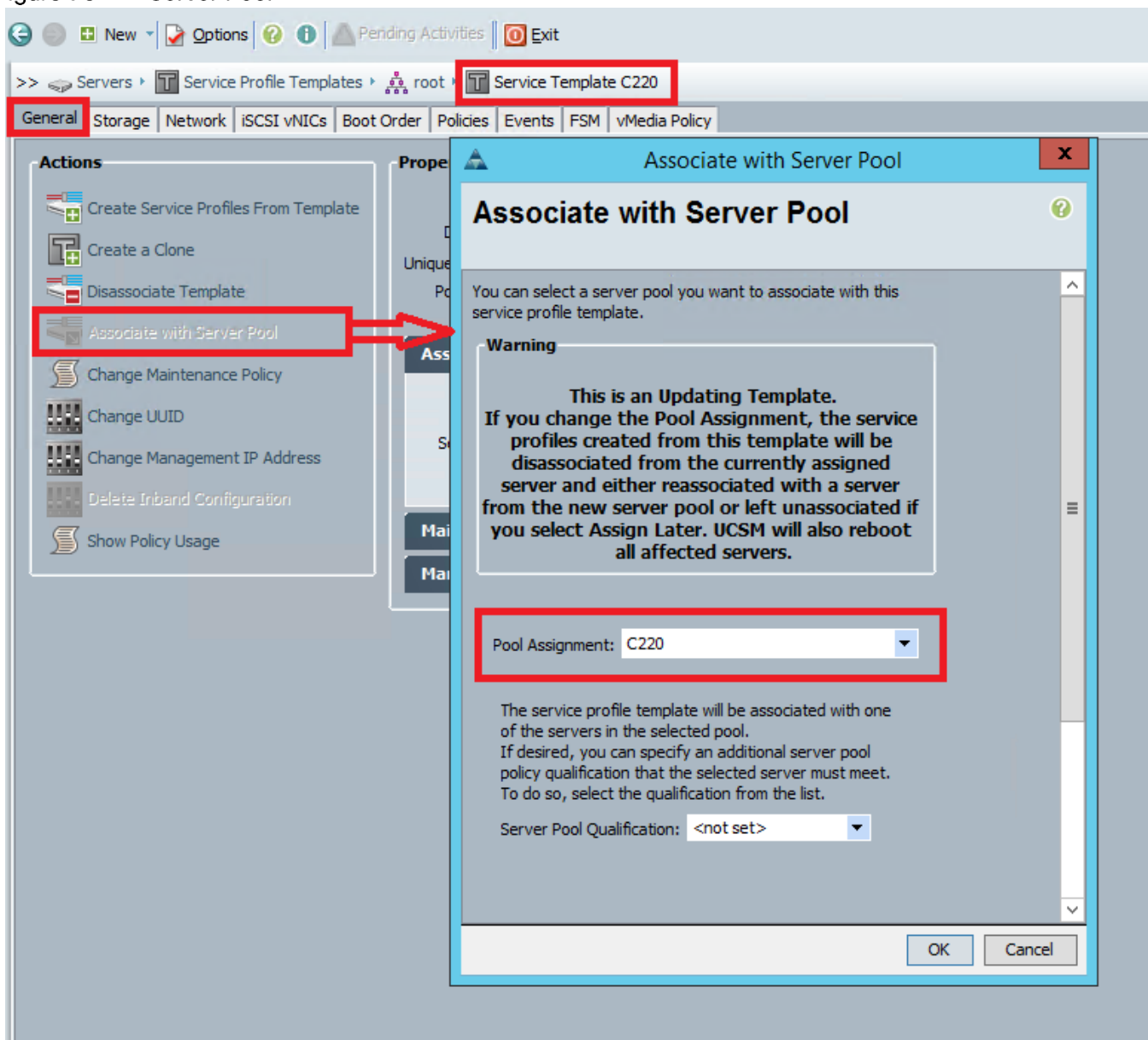
5. Enter C220 for the Clone Name and select root for the organization, as shown in Figure 74

Figure 74 Create Clone From ucs



6. Select the new service template in the navigation pane.
7. In the right side window, select the General tab and then click Associate with Server Pool.
8. The Associate with Server Pool window appears, as shown in Figure 75. In the Pool Assignment drop down list, choose C220.
9. Leave the Server Pool Qualification as <not set> and click OK.

Figure 75 Server Pool



10. Click **OK** at the success message.
11. Select the **Storage** and **Storage Profiles** tabs.
12. Click on **Modify Storage Profile**, as shown in Figure 76.

Figure 76 Storage Profile Tab for C220 Template

The screenshot shows the 'Storage Profile Policy' tab for the 'Boot_SSD' profile. The 'Storage Profile Instance' is 'org-root/profile-Boot_SSD'. Below this, the 'Local LUNs' table is empty except for the 'Boot_SSD' entry. The 'Details' section provides further information about the LUN configuration.

Name	RAID Level	Size (MB)	Config State	Deploy Name	LUN ID	Drive State
Boot_SSD	RAID 1 Mirrored	0	Not Applied			

LUN Details

Profile LUN Name: Boot_SSD	Order: Not Applicable
RAID Level: RAID 1 Mirrored	Size (MB) 0
Configured Size (GB) 1	Admin State: Online
Config State: Not Applied	Bootable: Disabled

Deployed LUN Details

LUN New Name:	Referenced LUN Name:
Deploy Name:	LUN ID:
Drive State:	

13. Click on the Storage Profile Policy tab.

14. Choose Boot_HDD for the Storage Profile, as shown in Figure 77

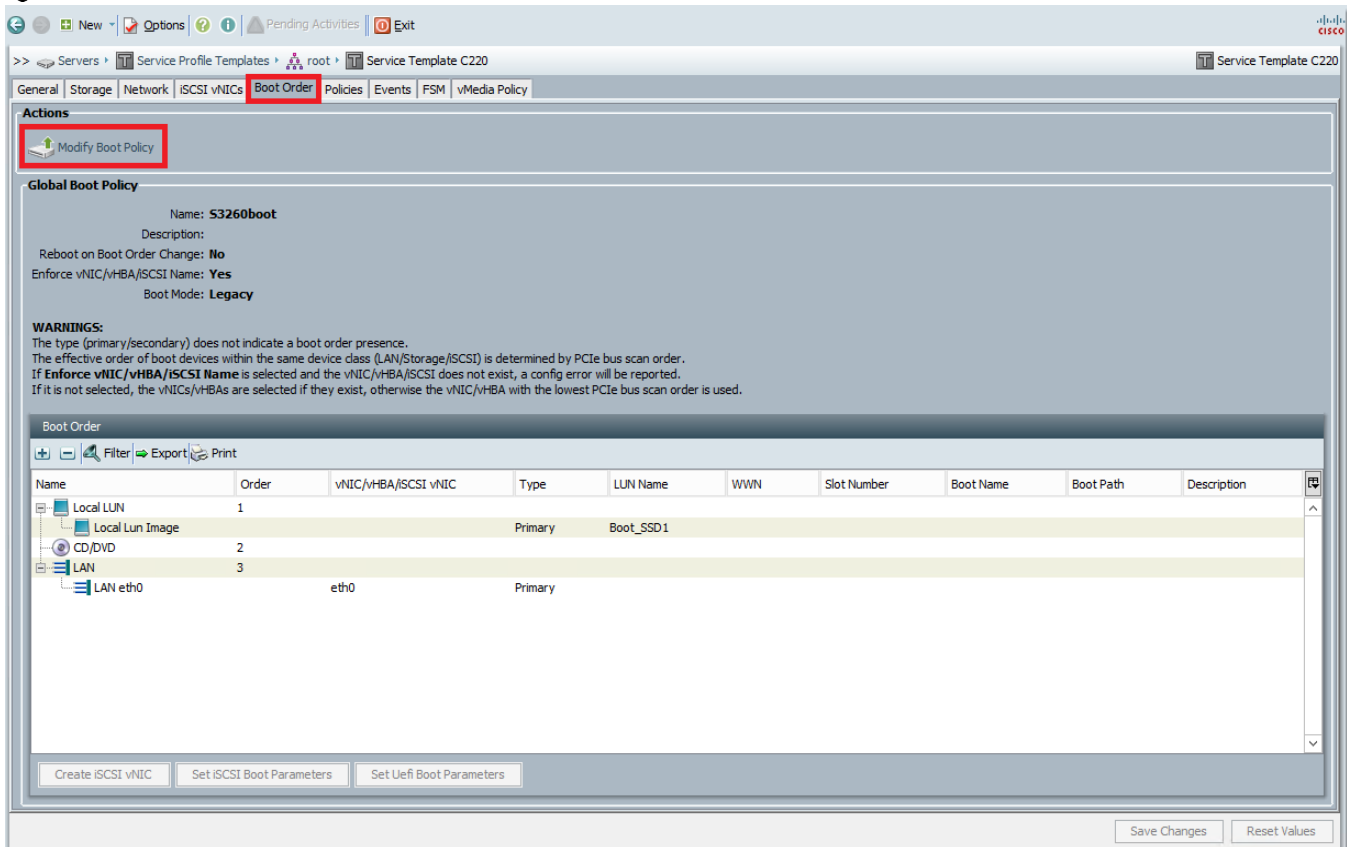
Figure 77 Storage Profile Policy



15. Click **OK** to finish modifying the storage profile. Click **OK** on the success dialog box.

16. Click on the **Boot Order** tab, as shown in Figure 78

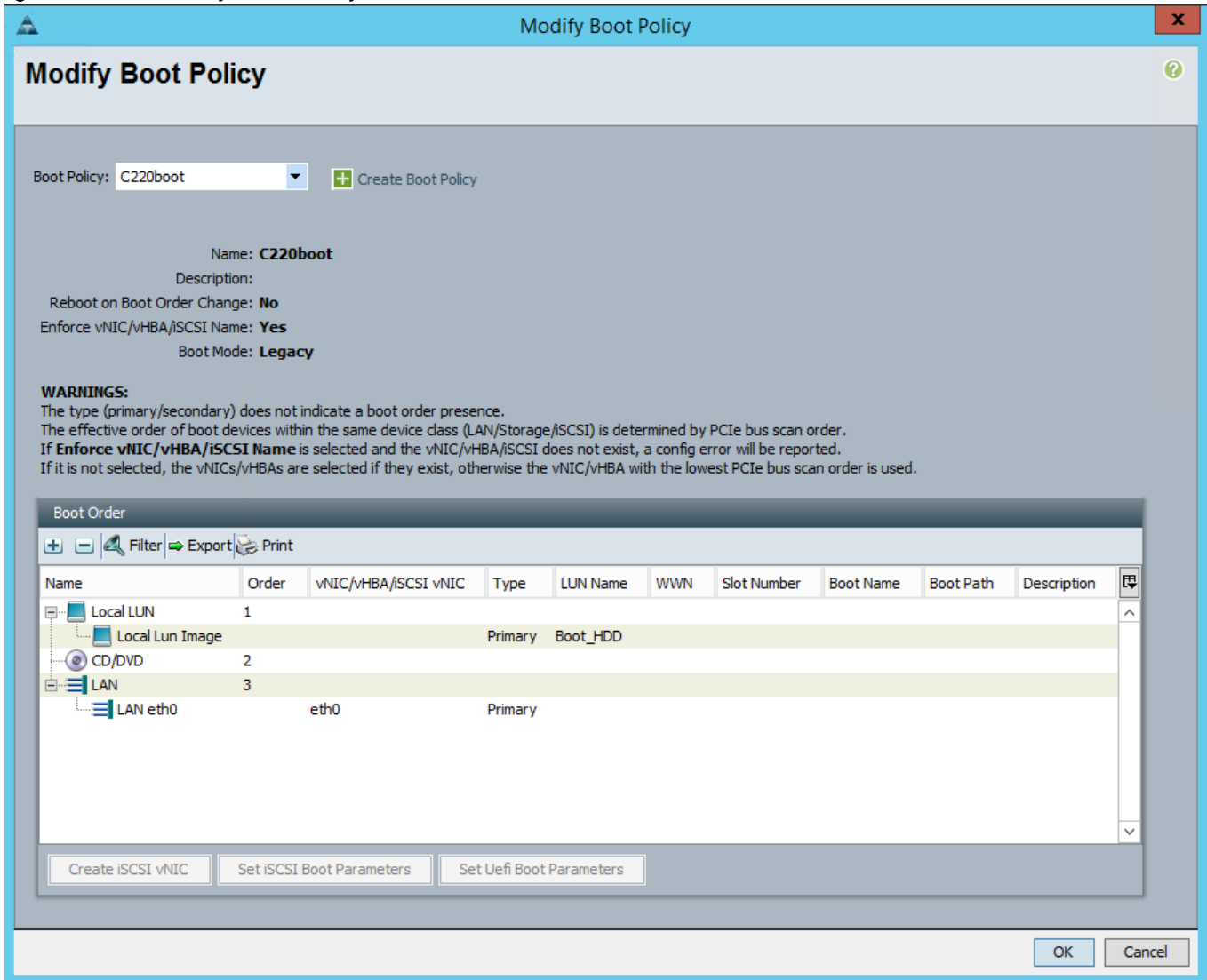
Figure 78 Boot Order



17. The current boot policy is S3260boot. Click on **Modify Boot Policy** to change it.

18. From the **Boot Policy** drop down list, choose C220boot and click **OK**.

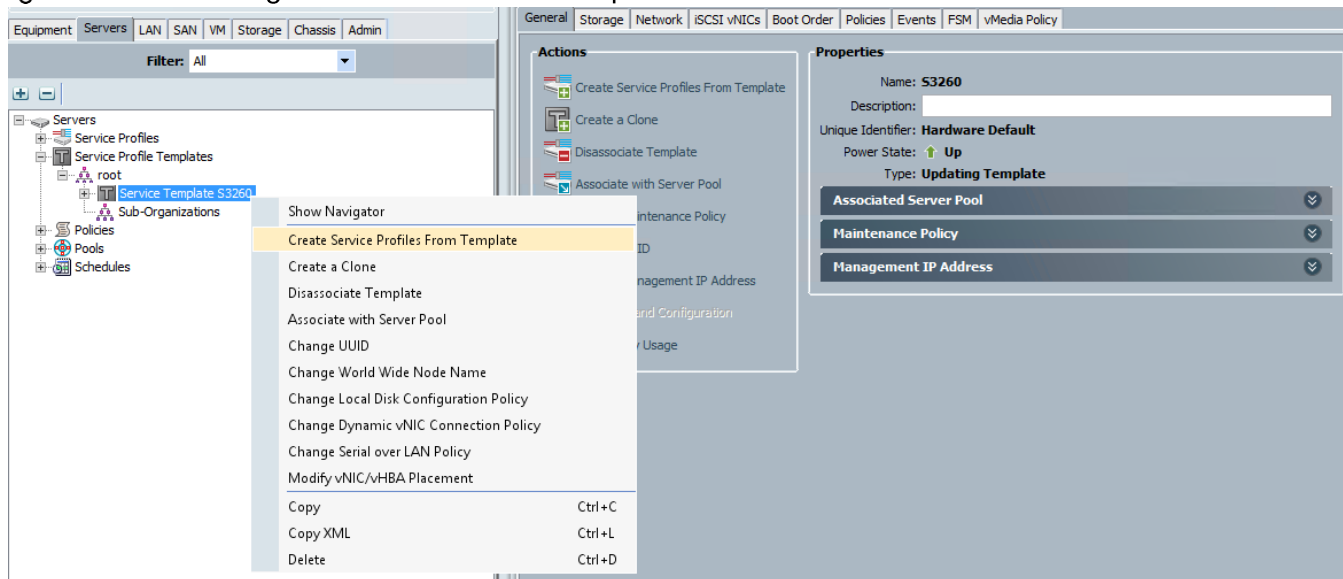
Figure 79 Modify Boot Policy



Creating Service Profiles from Template

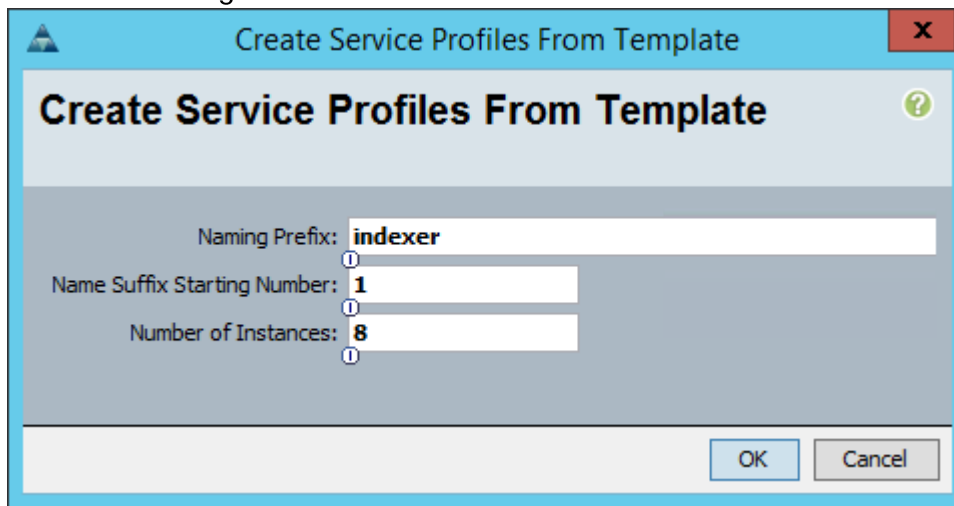
1. Select the `Servers` tab in the left pane of the UCS Manager GUI.
2. Go to `Service Profile Templates` → `root`, as shown in Figure 80
3. Right-click `Service Profile Templates S3260`.
4. Select `Create Service Profiles From Template`.

Figure 80 Creating Service Profiles from Template



5. In the Create Service Profile from Template window enter the following:
 - a. In the field Naming Prefix, enter `indexer`, as shown in Figure 81
 - b. In the field Enter Name Suffix Starting Number, enter 1.
 - c. In the field Number of Instances, enter 8.

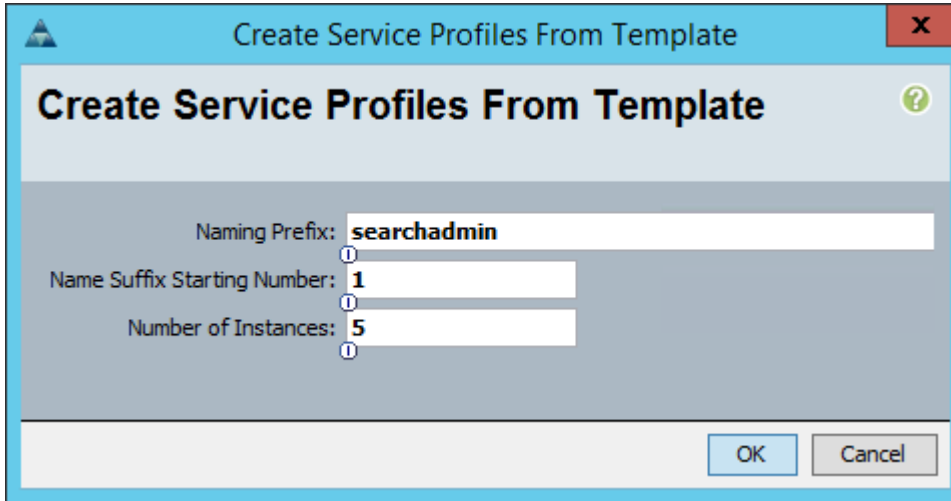
Figure 81 Selecting Name and Total number of Service Profiles for S3260 servers



Note: Association of the Service Profiles will take place automatically.

6. Repeat the above steps to create five service profiles from the C220 Service Profile Template. Use the parameters in Figure 82

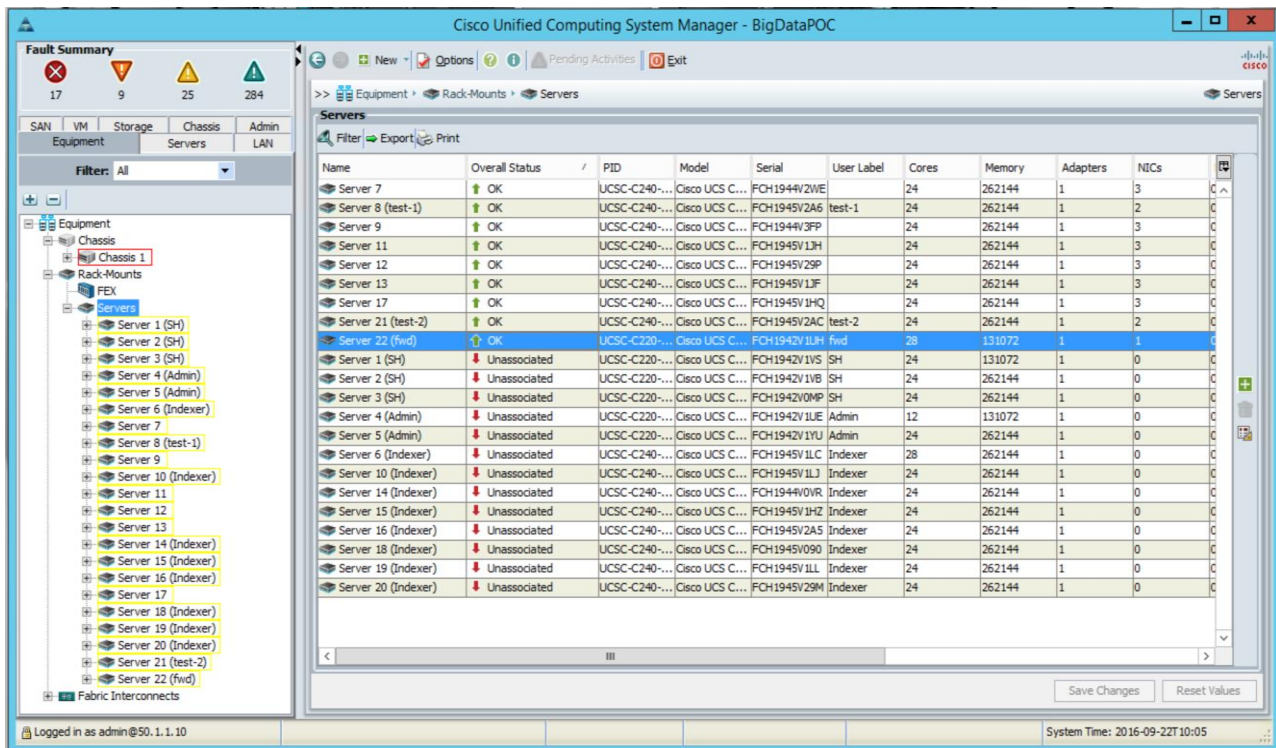
Figure 82 Selecting Name and Total number of Service Profiles for C220 servers



Identifying the Servers

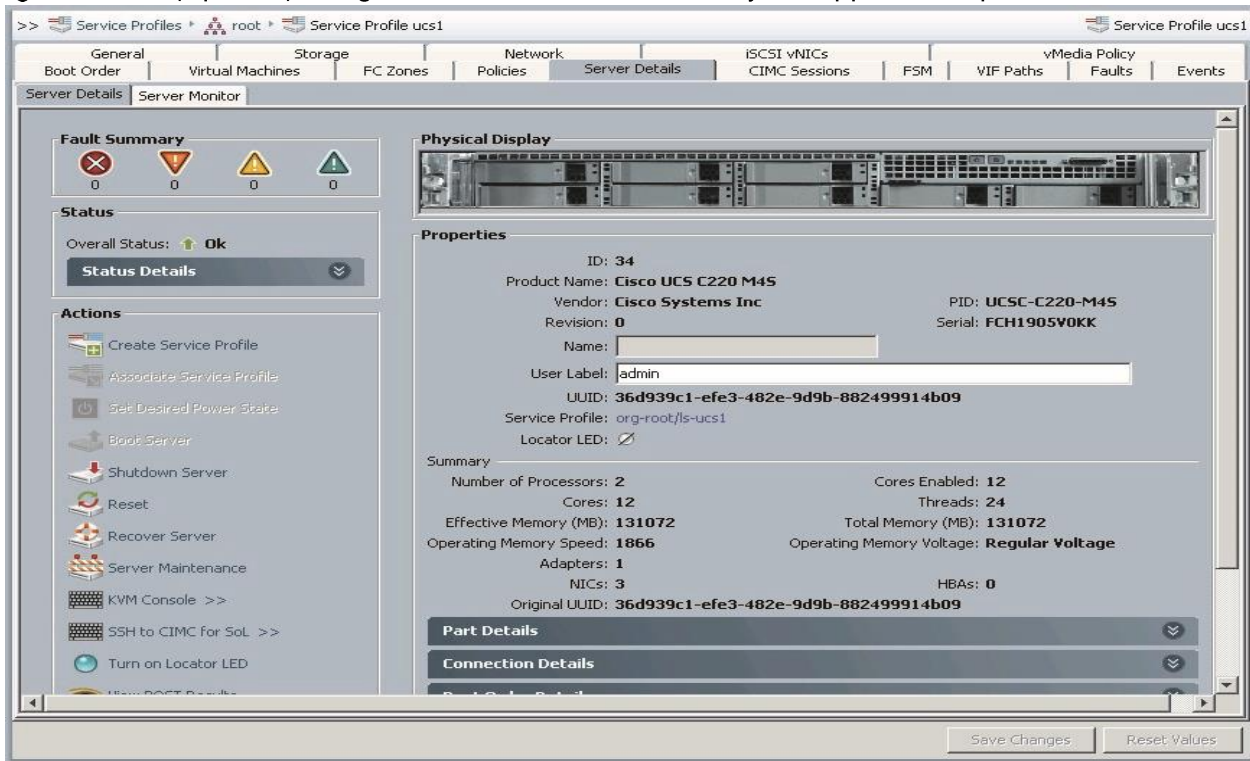
1. In the Equipment Tab, select Rack-Mounts for the Filter, and click on Servers. In the right pane all thirteen servers are displayed along with their details. Figure 83

Figure 83 Cisco UCS Rack Servers Associated with Created Service Profiles



2. (Optional) Double click on an individual server instance and enter an appropriate text string (name or role) in the User Label as shown below in Figure 84. This could be helpful in identifying the server's application-specific roles.

Figure 84 (Optional) Using the User Label Field to Identify the Application Specific Roles



Installing Red Hat Enterprise Linux 7.2 on all servers

The following section provides detailed procedures for installing Red Hat Enterprise Linux 7.2 using Software RAID (OS based Mirroring) on Cisco UCS C220 M4 Rack Servers and Cisco S3260 Storage Servers. There are multiple ways to install the Red Hat Linux operating system. The installation procedure described in this deployment guide uses KVM console and virtual media from Cisco UCS Manager.

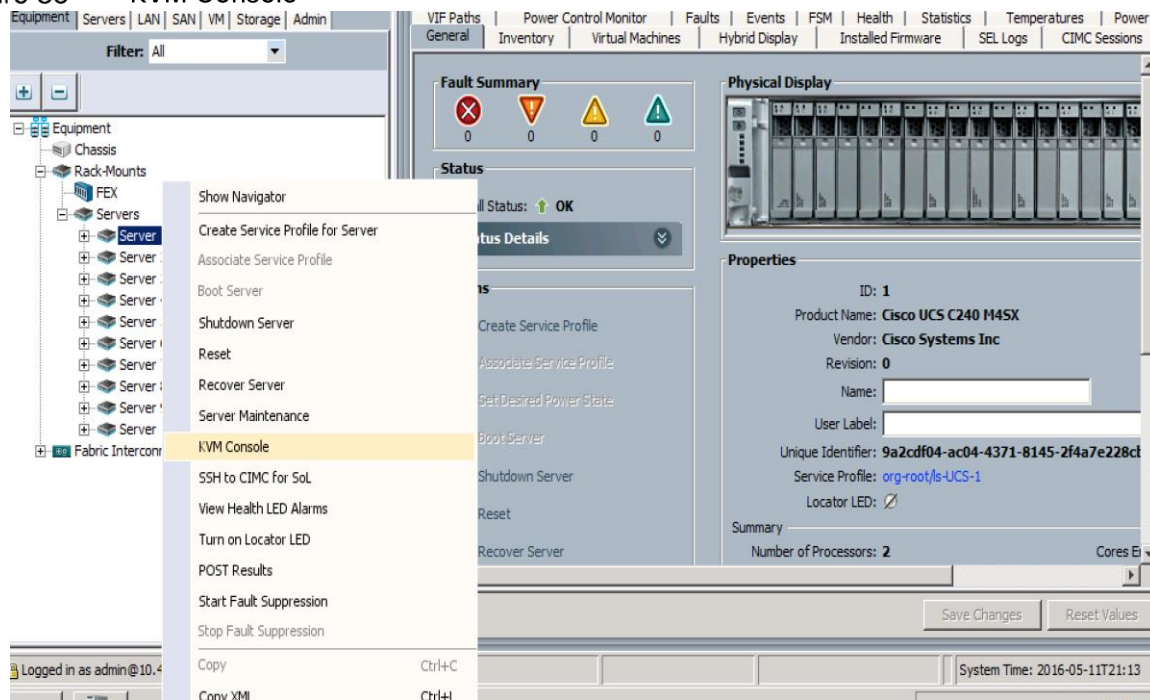


Note: This installation requires Red Hat Enterprise Linux (RHEL) 7.2 DVD/ISO.

To install the Red Hat Linux 7.2 operating system, complete the following steps:

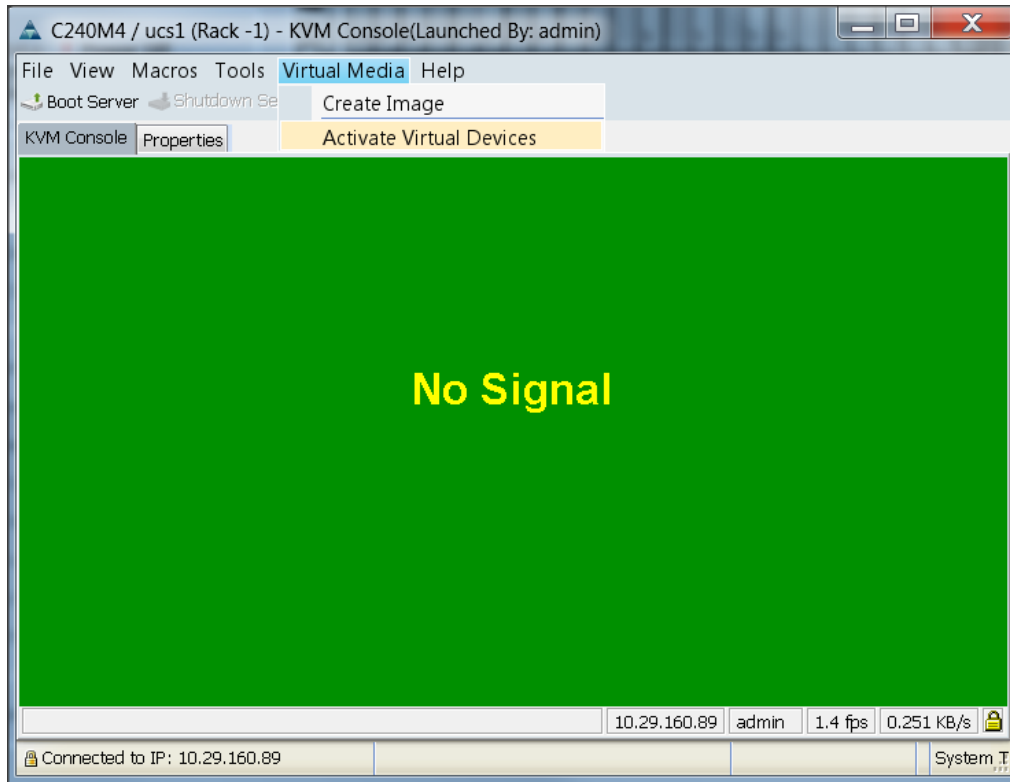
1. Log in to the Cisco UCS 6332 Fabric Interconnect and launch the Cisco UCS Manager application.
2. Select the `Equipment` tab as shown in Figure 85 .
3. To install Red Hat on a Cisco C220 server, expand `Rack-Mounts` and then `Servers`. To install Red Hat on a server node of the Cisco S3260 Storage Server, expand `Chassis`, select a chassis, expand `Servers`, and then select a server.
4. Right click on the server and select `KVM Console`.
5. In the KVM window, select the `Virtual Media` tab.

Figure 85 KVM Console



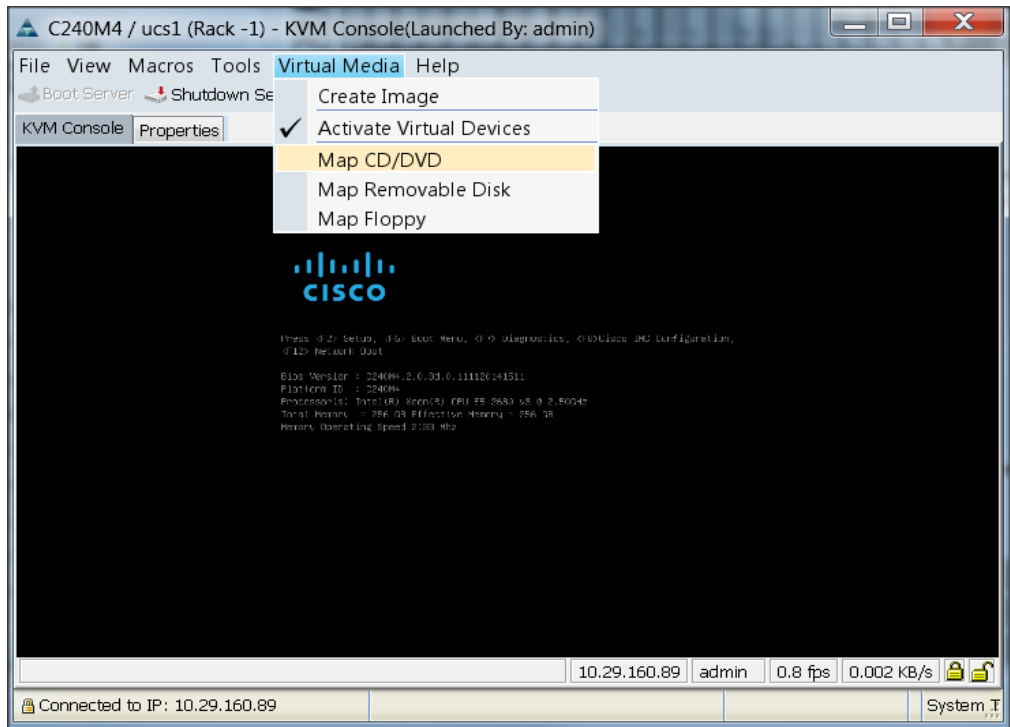
6. Click the `Activate Virtual Devices` found in the `Virtual Media` tab. (Figure 86 below.)

Figure 86 Virtual Media Tab



7. In the KVM window (Figure 87), select the Virtual Media tab and click Map CD/DVD.

Figure 87 KVM Console Window

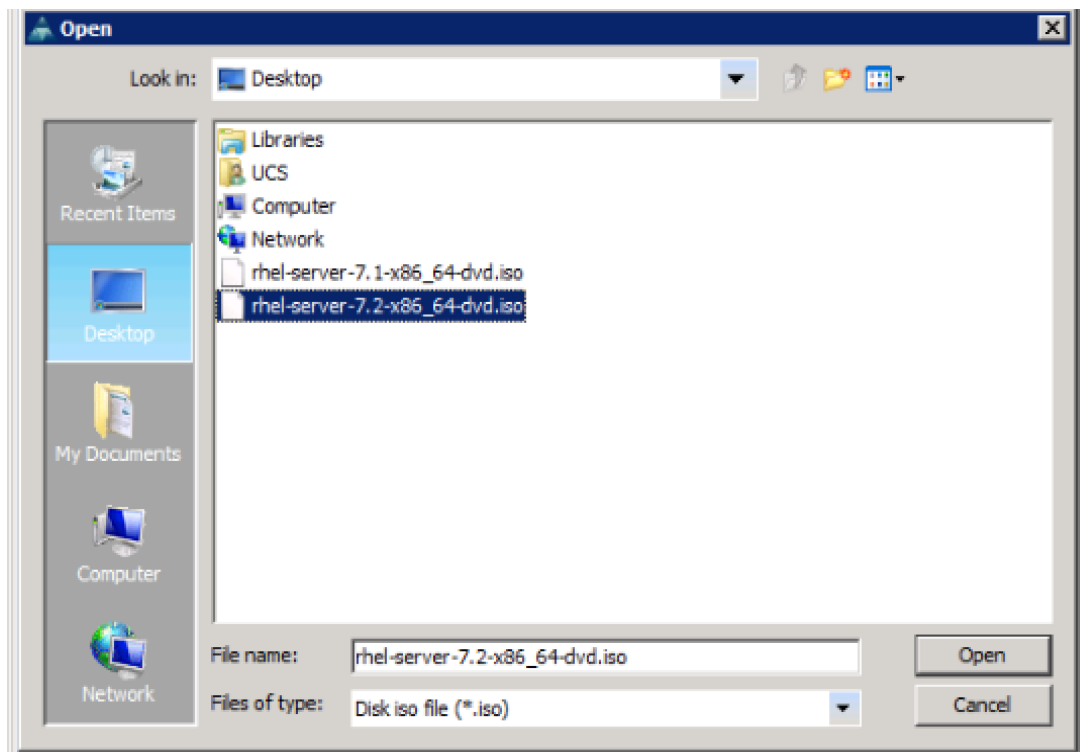


8. Browse to the Red Hat Enterprise Linux Server 7.2 installer ISO image file.

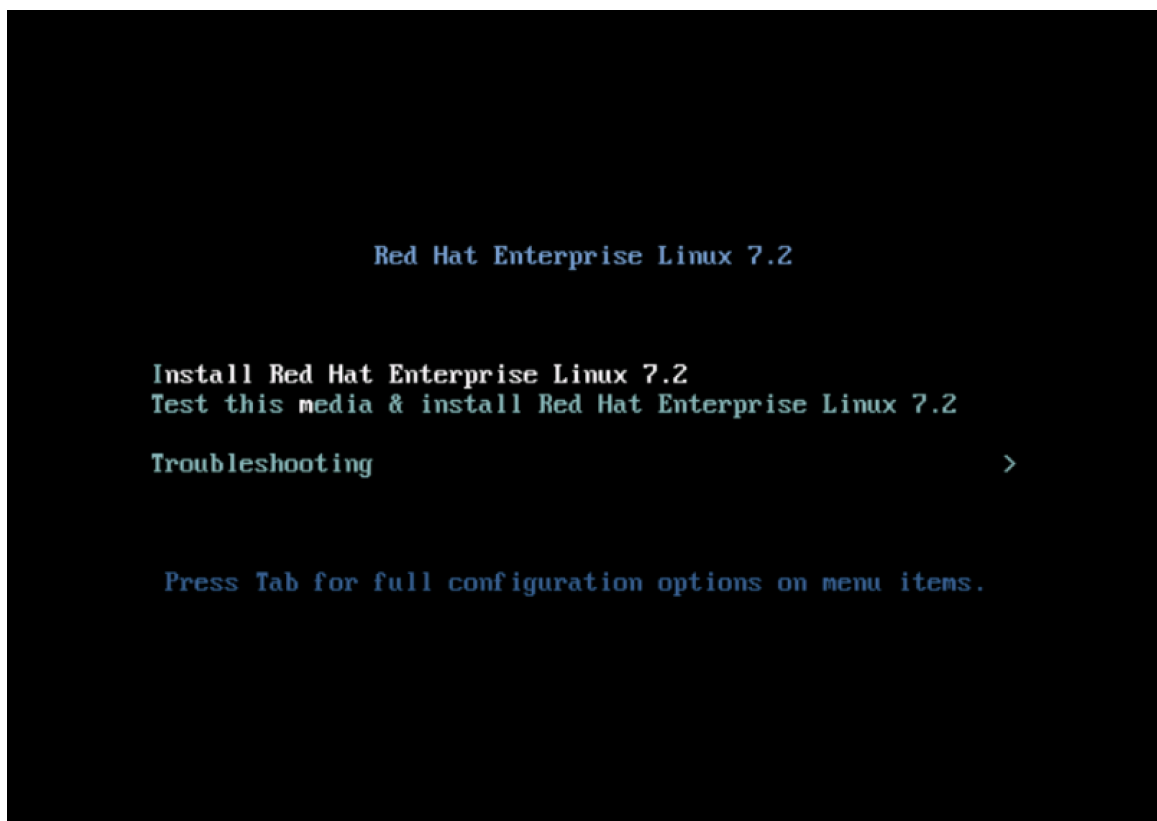


Note: The Red Hat Enterprise Linux 7.2 DVD is assumed to be on the client machine.

9. Click `Open` to add the image to the list of virtual media.



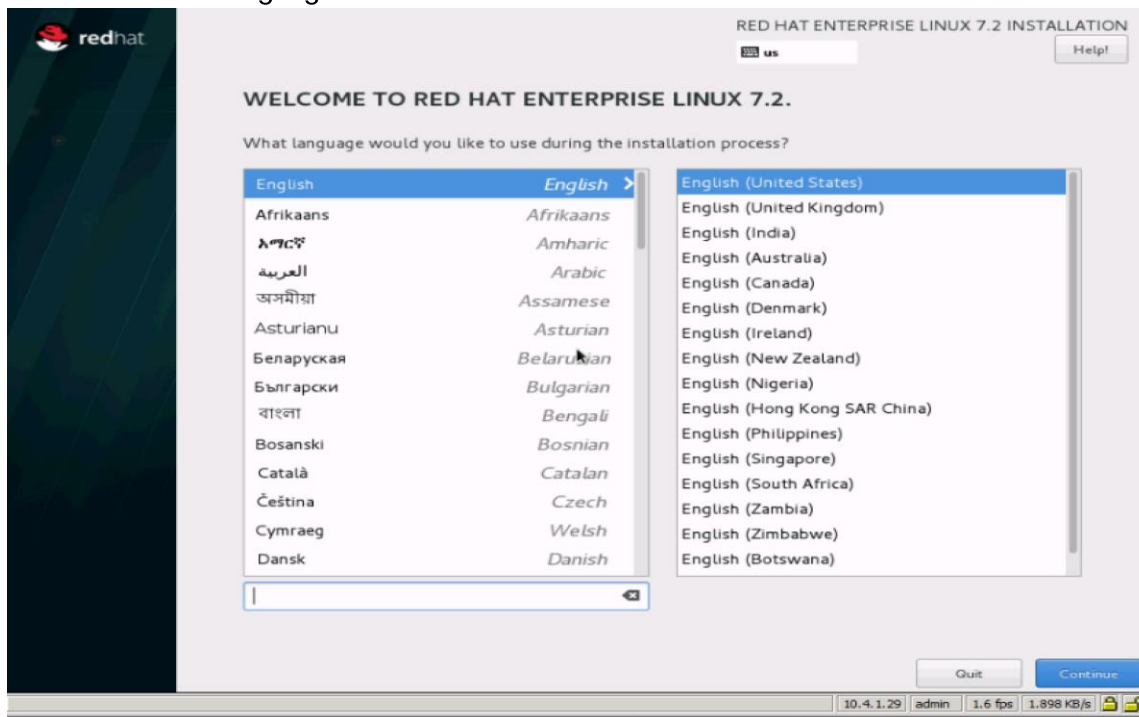
10. In the KVM window, select the `Macros` → `Static Macros` → `Ctrl-Alt-Del` button in the upper left corner.
11. Click `OK`.
12. Click `OK` to reboot the system.
13. On reboot, the machine detects the presence of the Red Hat Enterprise Linux Server 7.2 install media.
14. Select `Install or Upgrade an Existing System`.



15. Skip the Media test and start the installation.

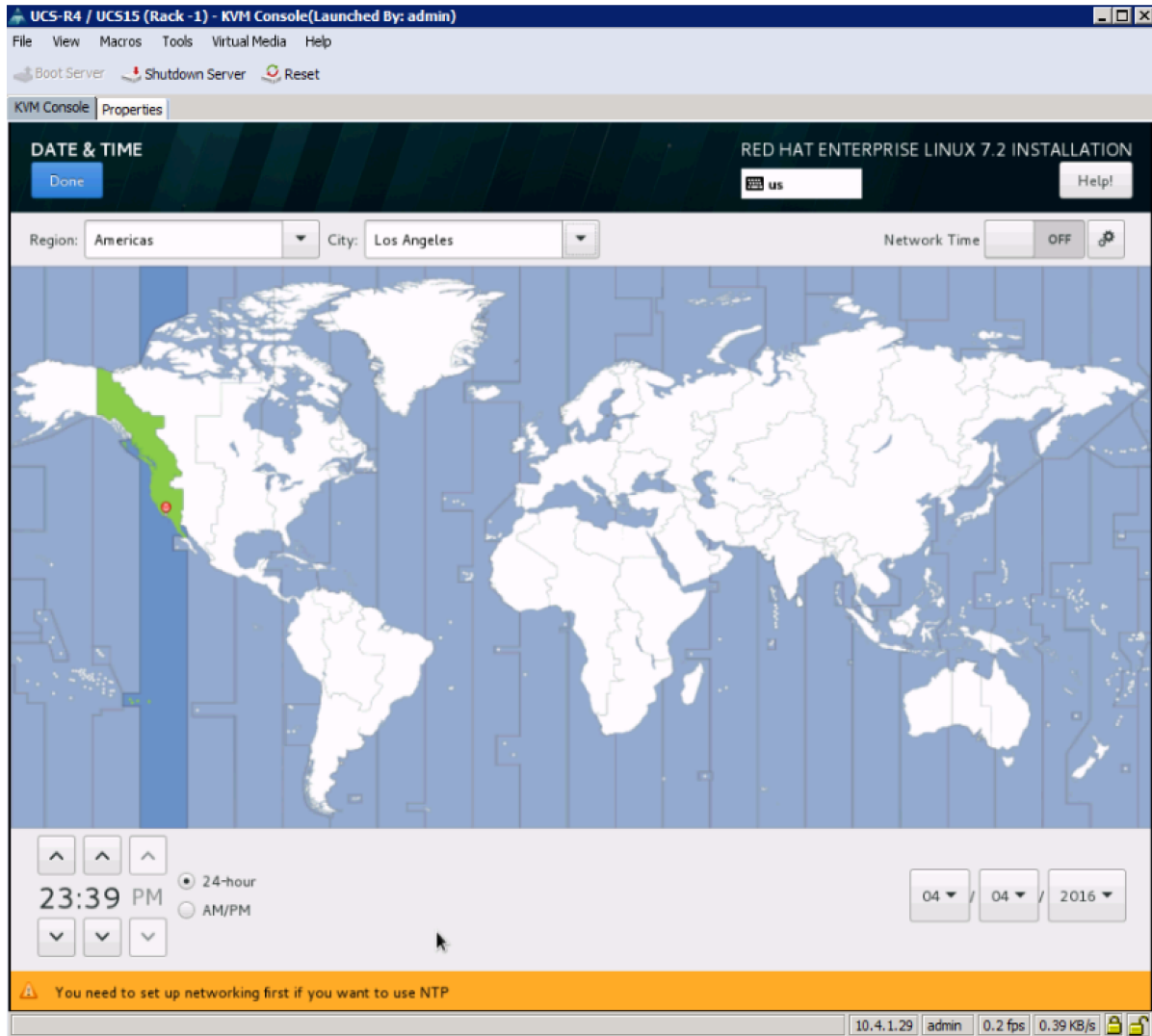
16. Select language of installation (Figure 88), and click Continue.

Figure 88 Select Language Window



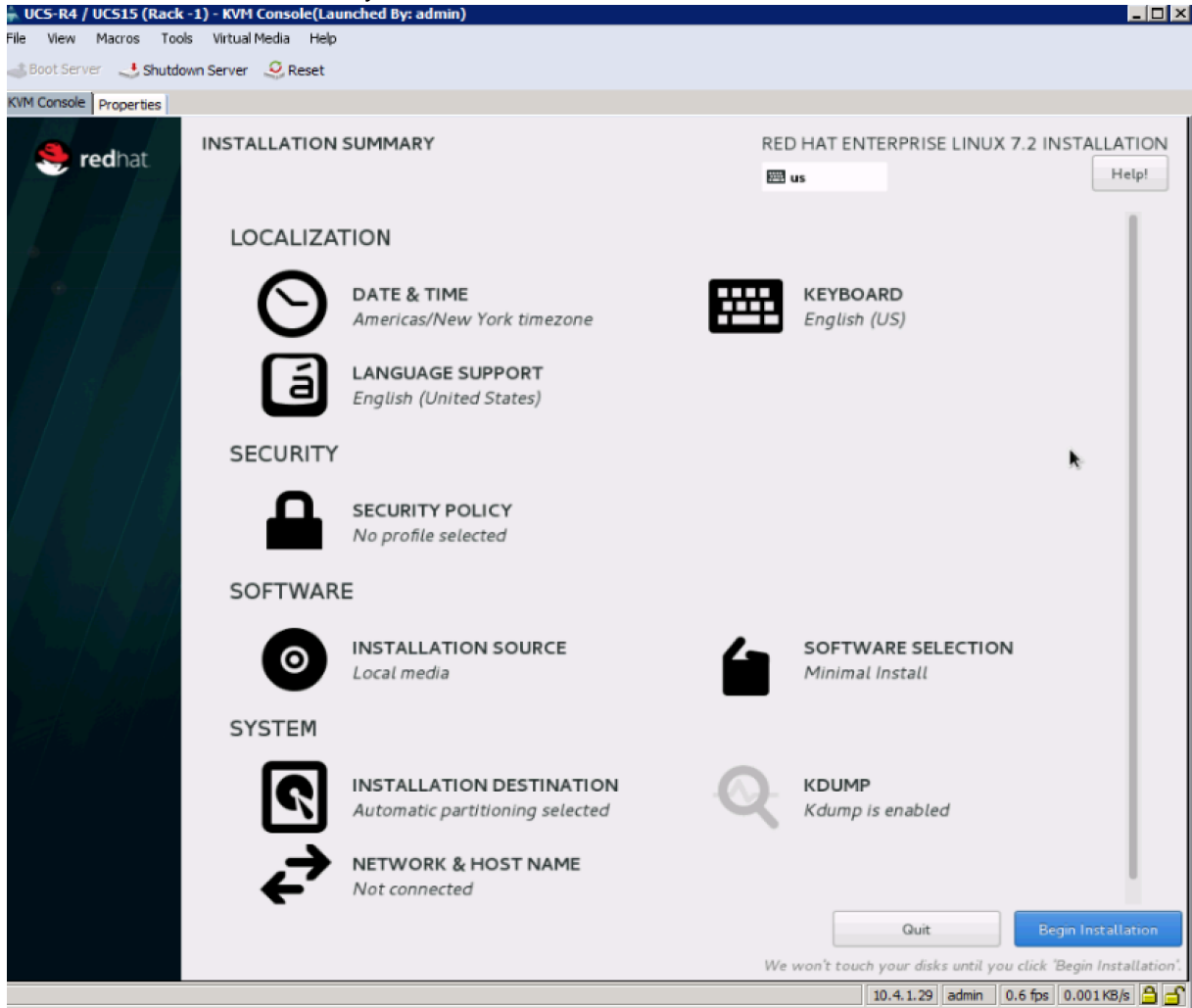
17. Select Date and Time.

Figure 89 Date and Time Window



18. Select the location on the map, set the time and click Done.

Figure 90 Installation Summary Window



19. Click on Installation Destination, shown above in Figure 90 .

Figure 91 Installation Summary Window



A Caution symbol appears next to Installation Destination as shown in Figure 91 above.

20. This opens the Installation Destination window displaying the boot disks. This is shown in Figure 92 below.

21. Make the appropriate disk selection, and choose I will configure partitioning. Click Done in the upper left.

Figure 92 Installation Destination Window for Cisco C220 server

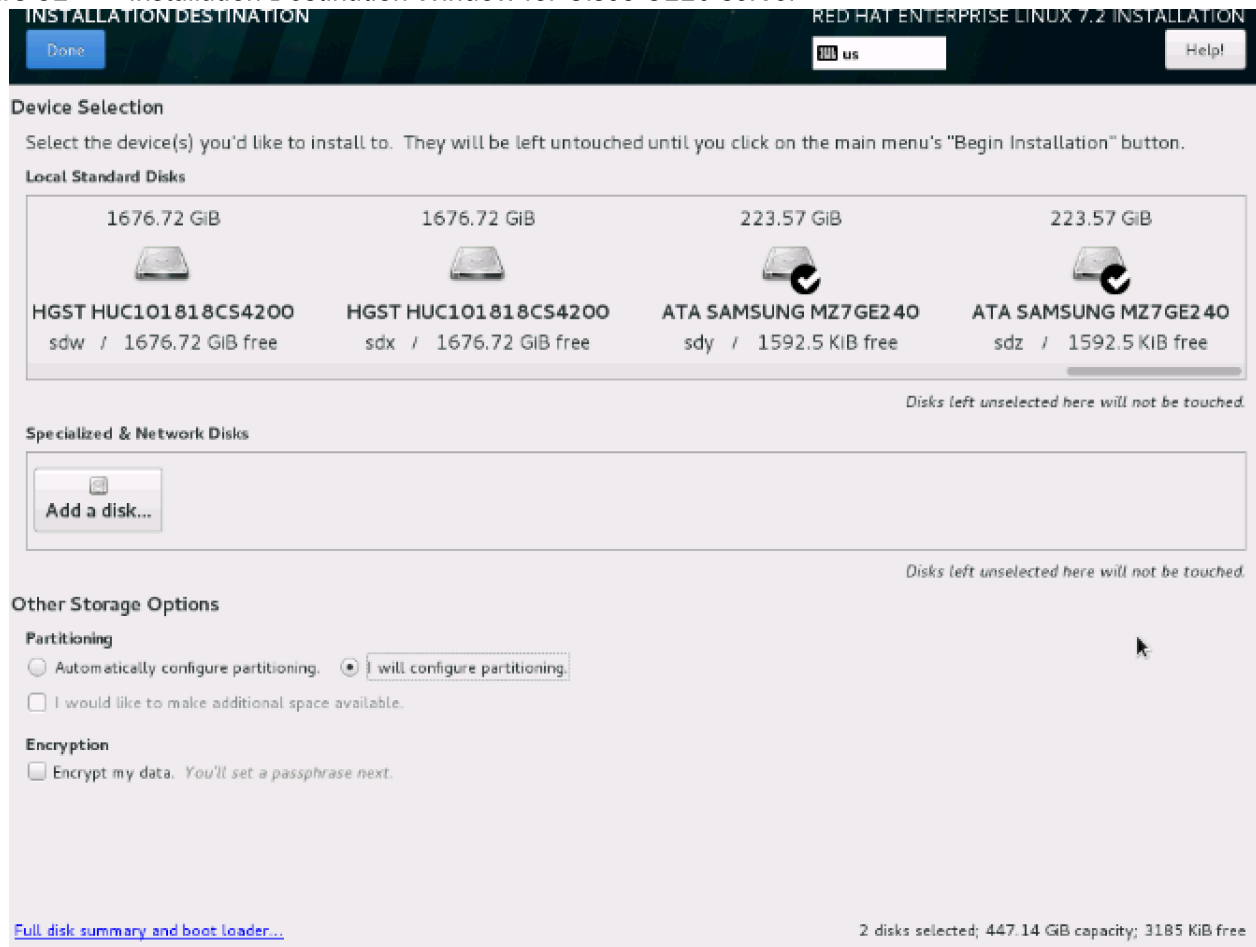
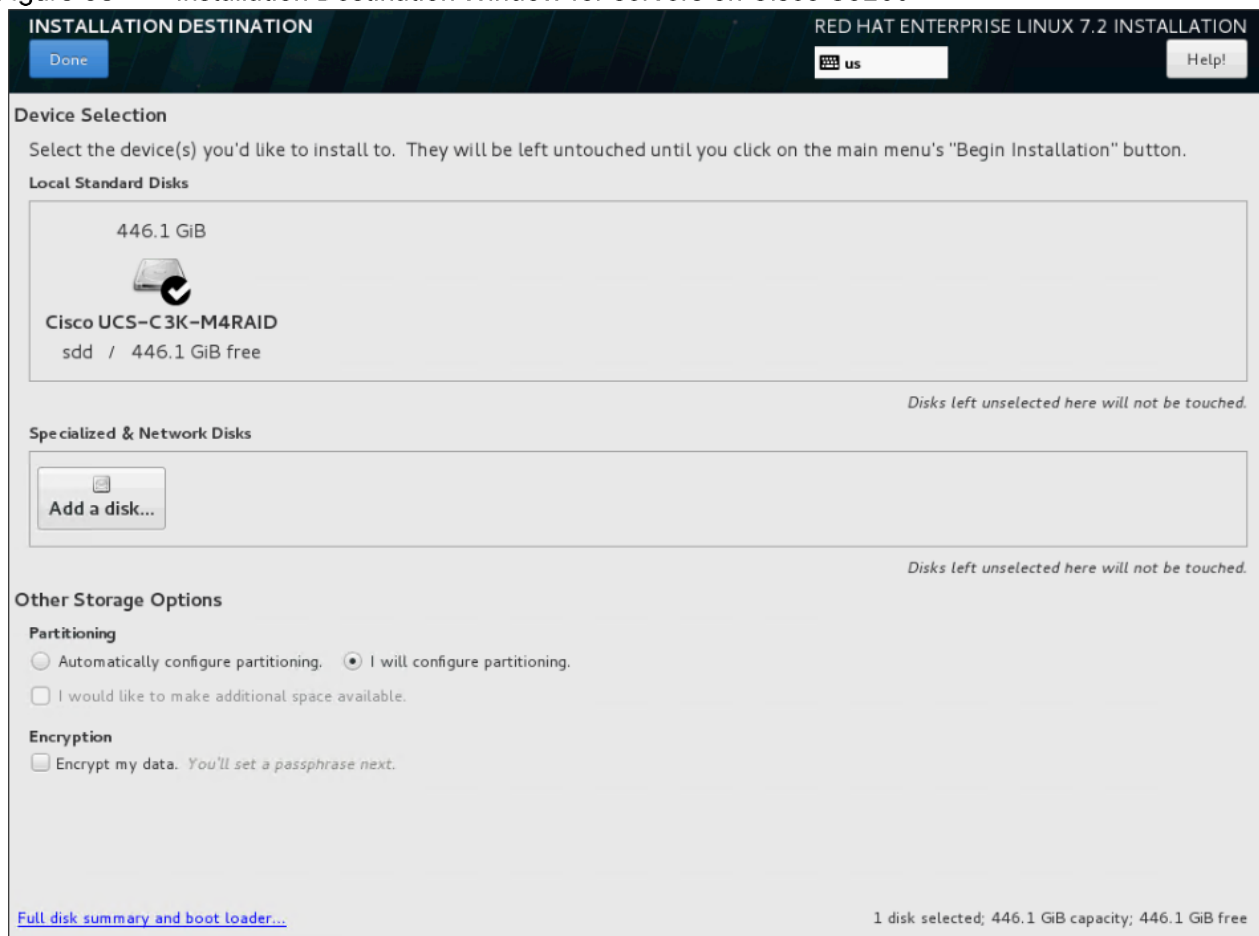


Figure 93 Installation Destination Window for servers on Cisco S3260

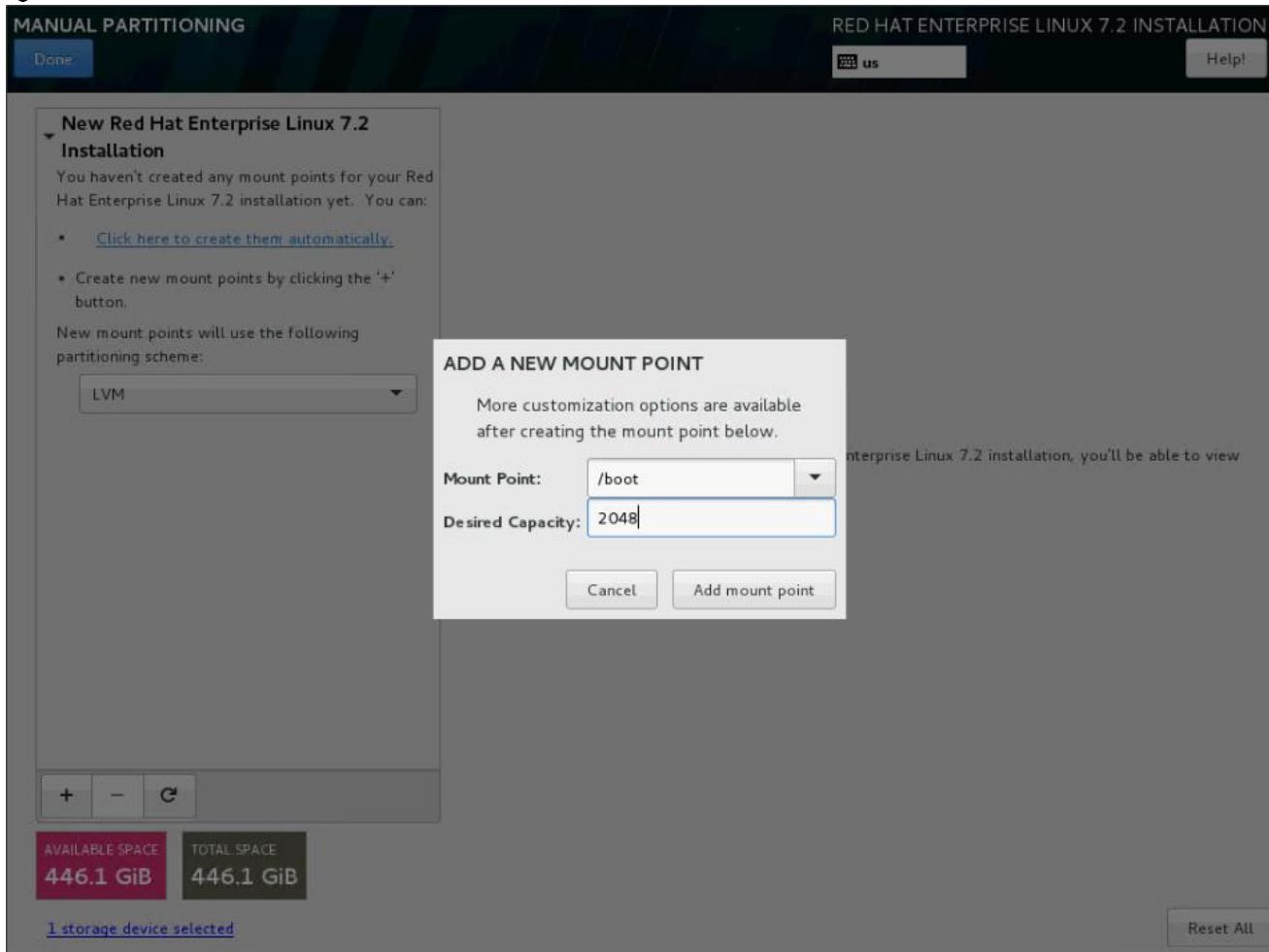


22. This opens the new window for creating the partitions.

23. Click on the + sign to add a new partition as shown below, boot partition of size 2048 MB.

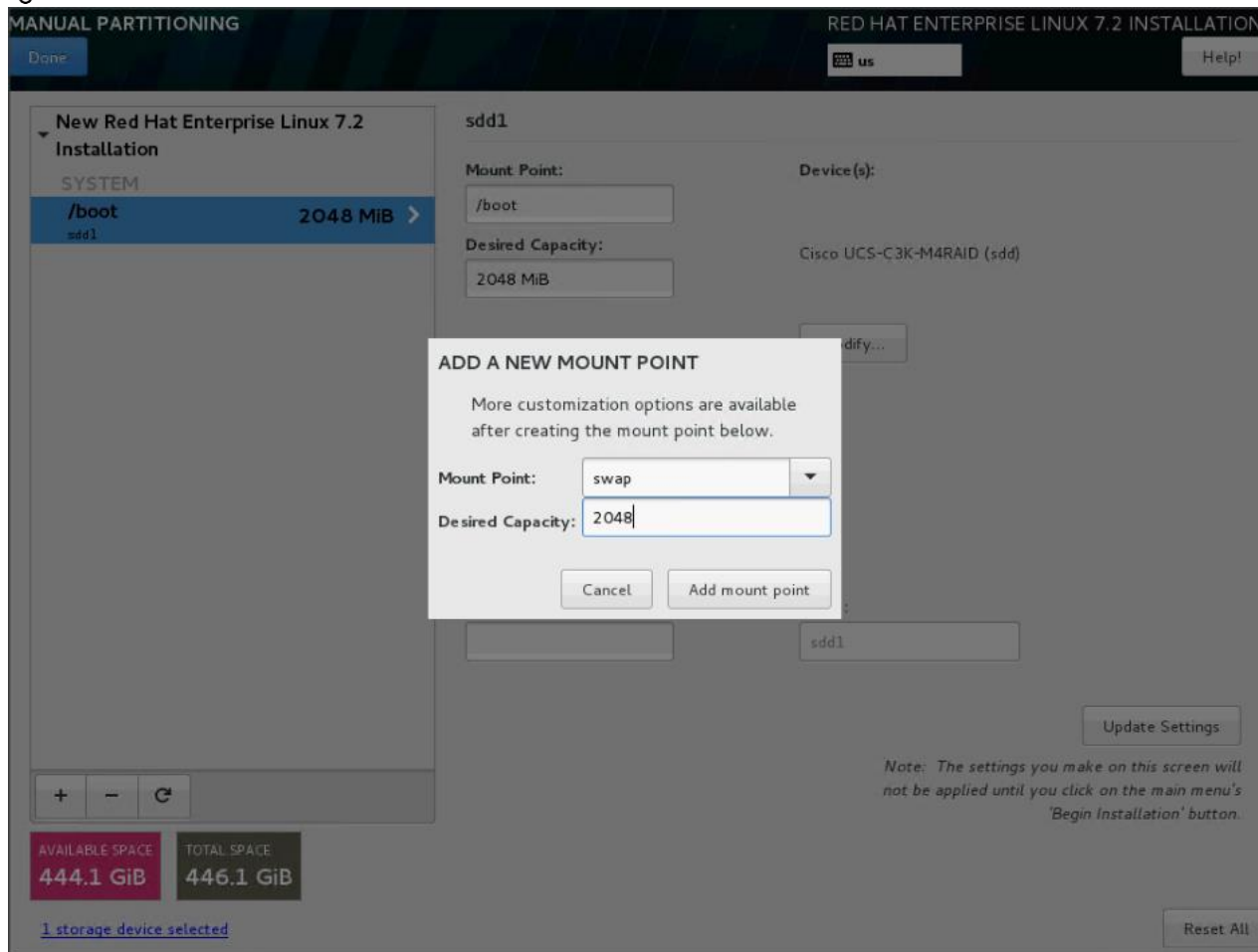
24. Click Add mount point to add the partition.

Figure 94 Add a New Mount Point



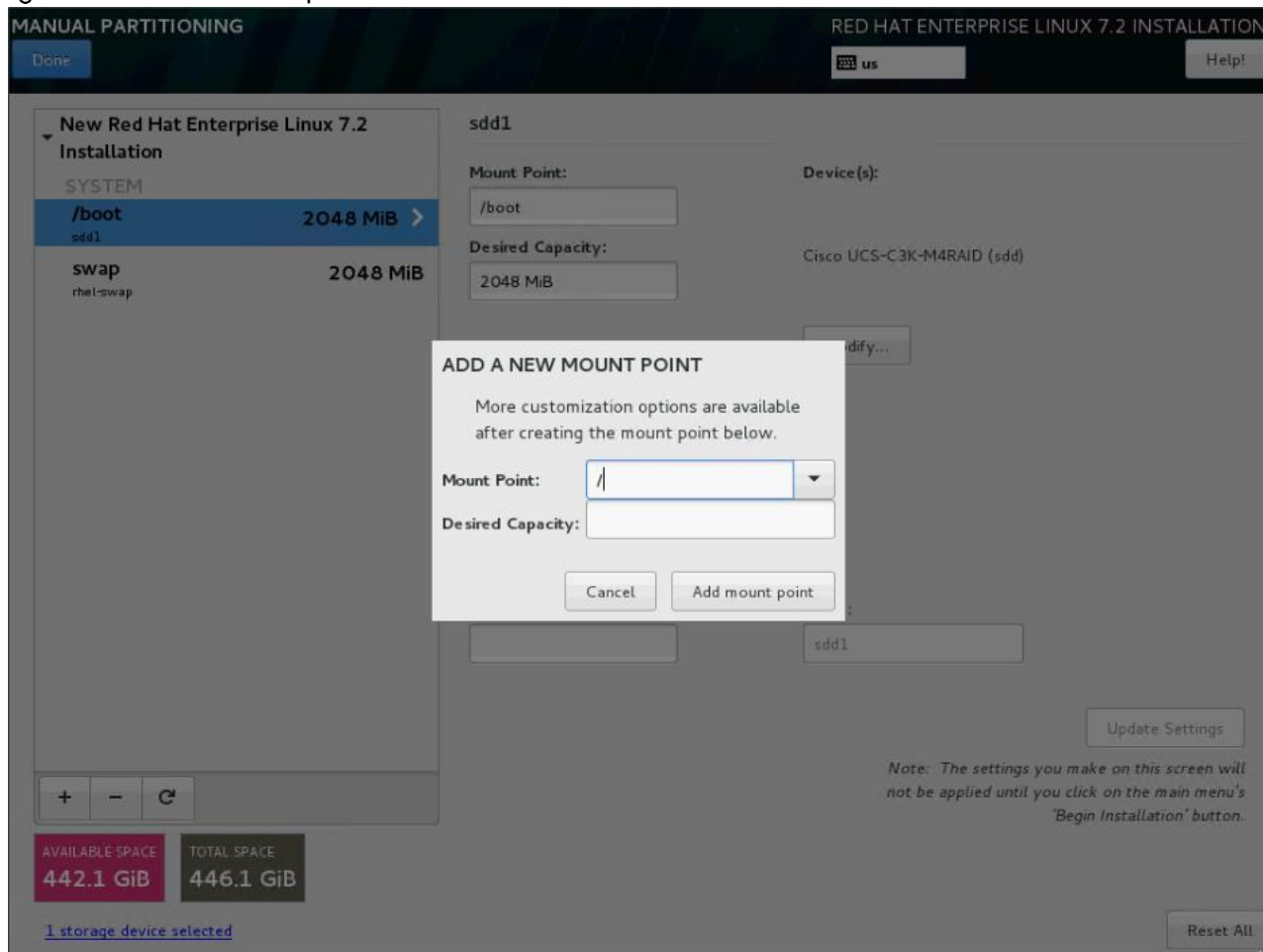
25. Click on the + sign to create the swap partition of size 2048 MB as shown below.

Figure 95 Add a New Mount Point



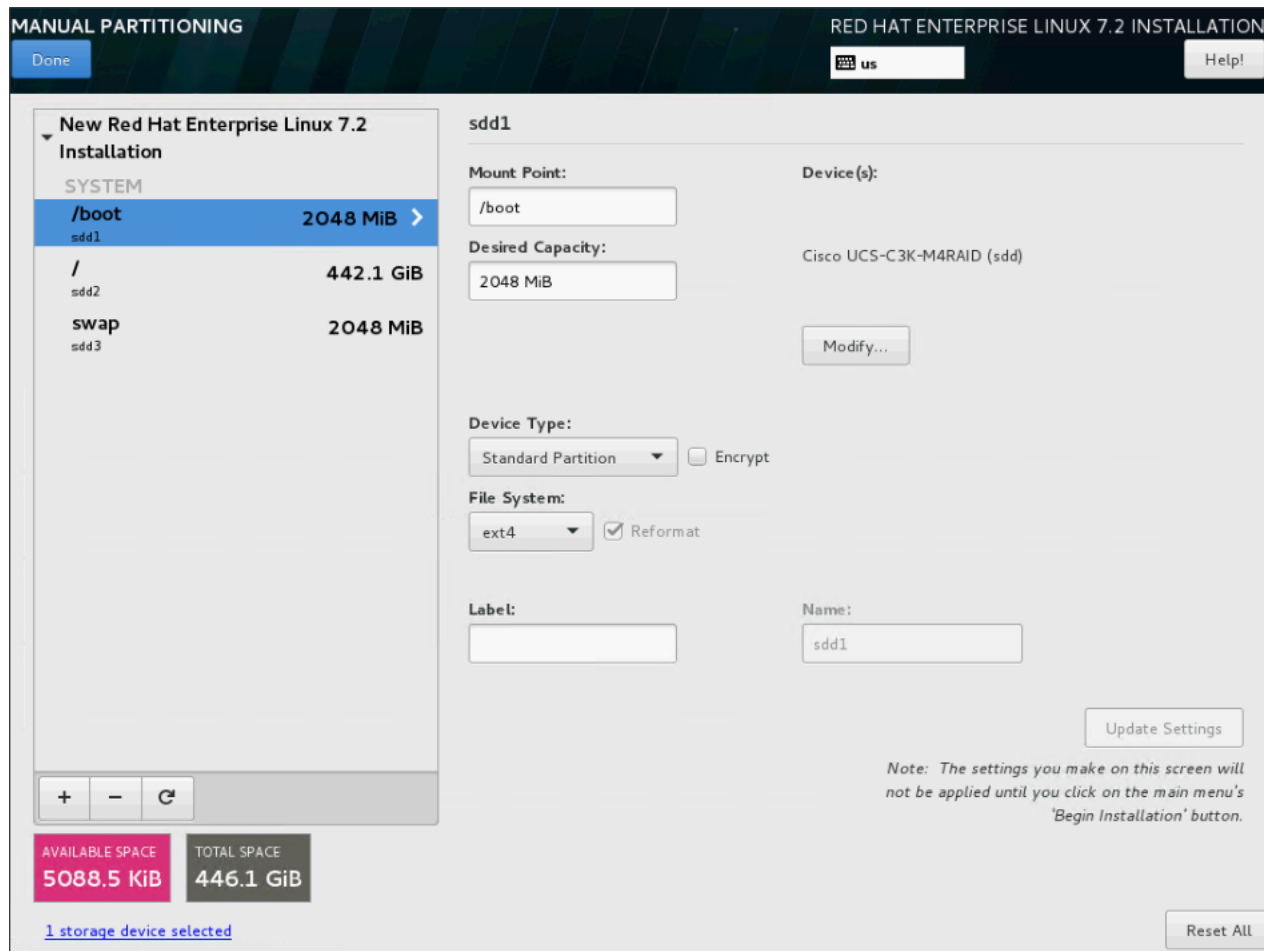
26. Repeat this process to add three more partitions: /home, /var/tmp, and /tmp with Desired Capacity of 5120.
27. Click + to add the / partition. The size can be left empty so it uses the remaining capacity and click Add mount point.

Figure 96 Add a swap



28. Select each partition except for swap. Ensure the Device Type is Standard Partition and change the file system to ext4.
29. Select the swap partition and ensure the Device Type is Standard Partition.

Figure 97 Standard Partition

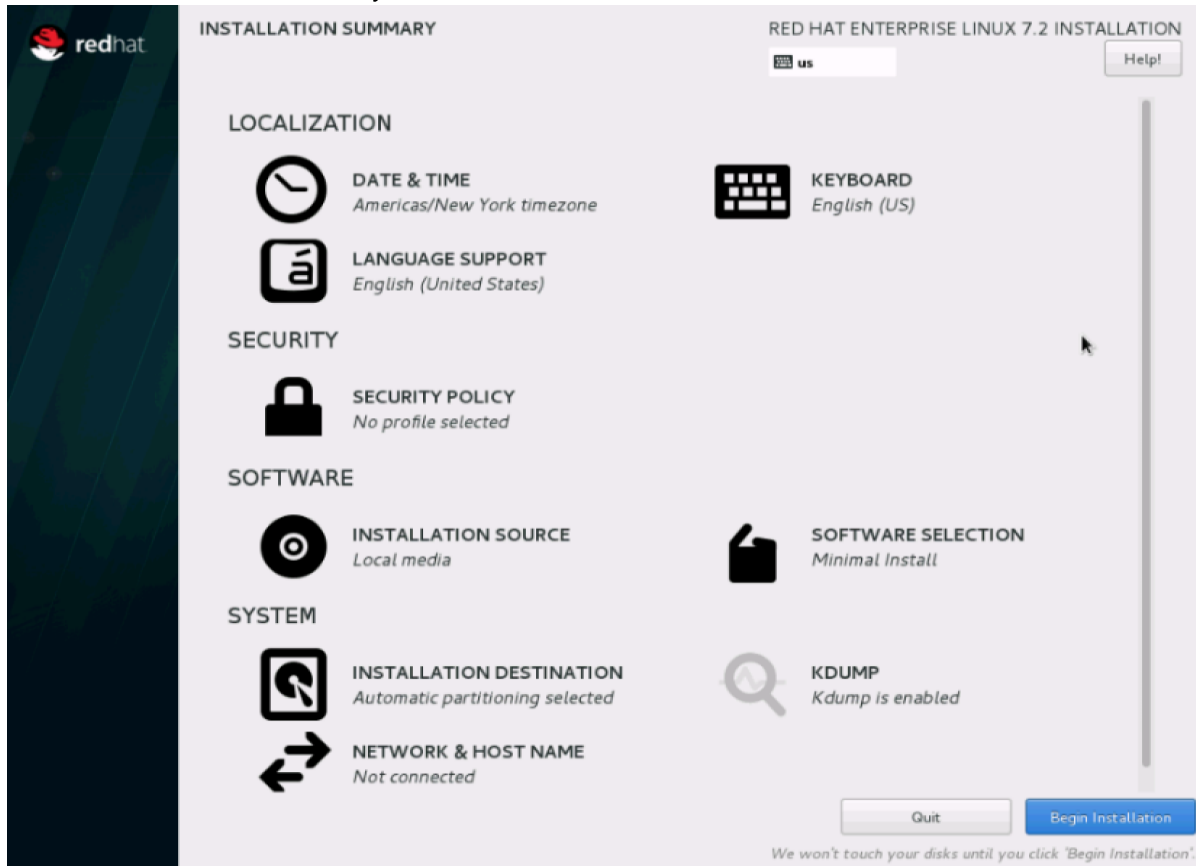


30. Click `Done` to go back to the main screen and continue the installation. A Summary of Changes screen may appear for review. Click `Accept Changes`.

The Installation screen opens (Figure 98).

31. Click on `Software Selection`.

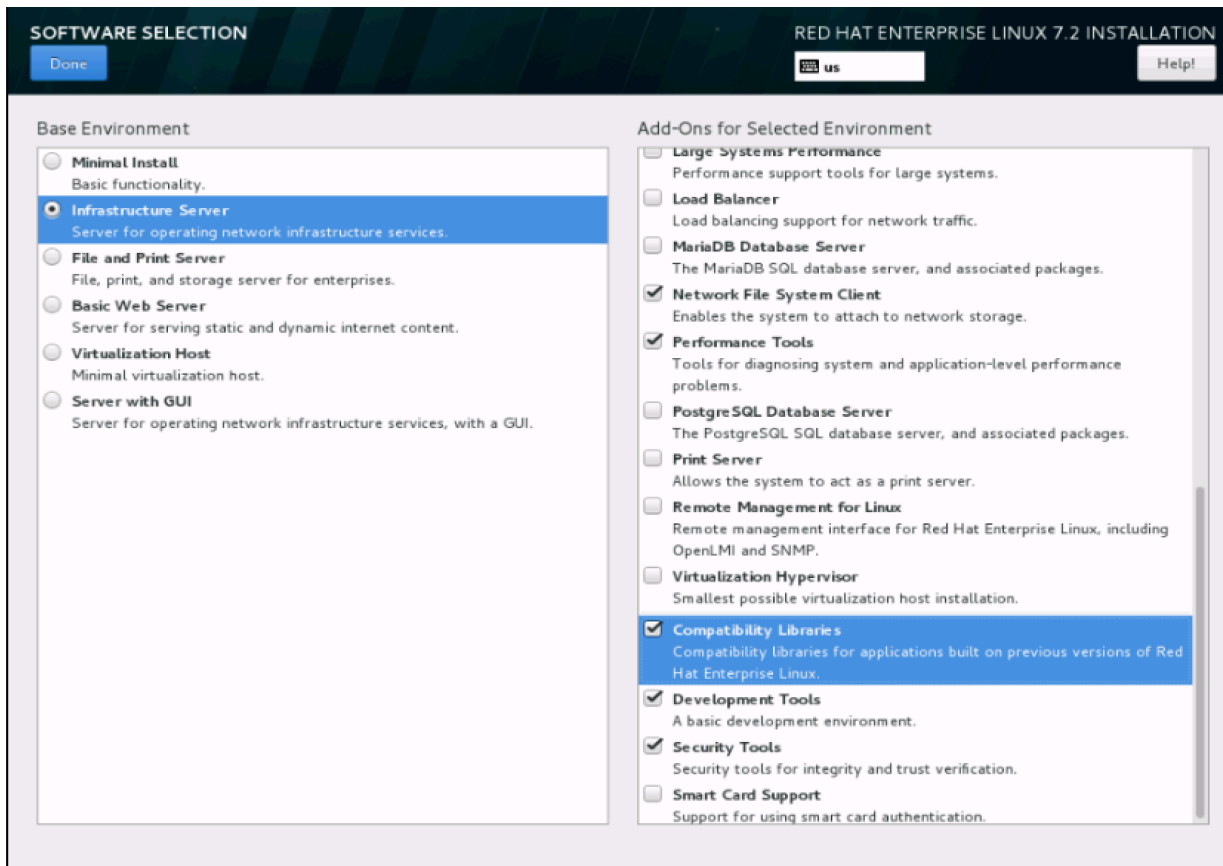
Figure 98 Installation Summary Window



The Software Selection screen opens (Figure 99).

32. Select Infrastructure Server and select the add-ons as noted below. Click Done.

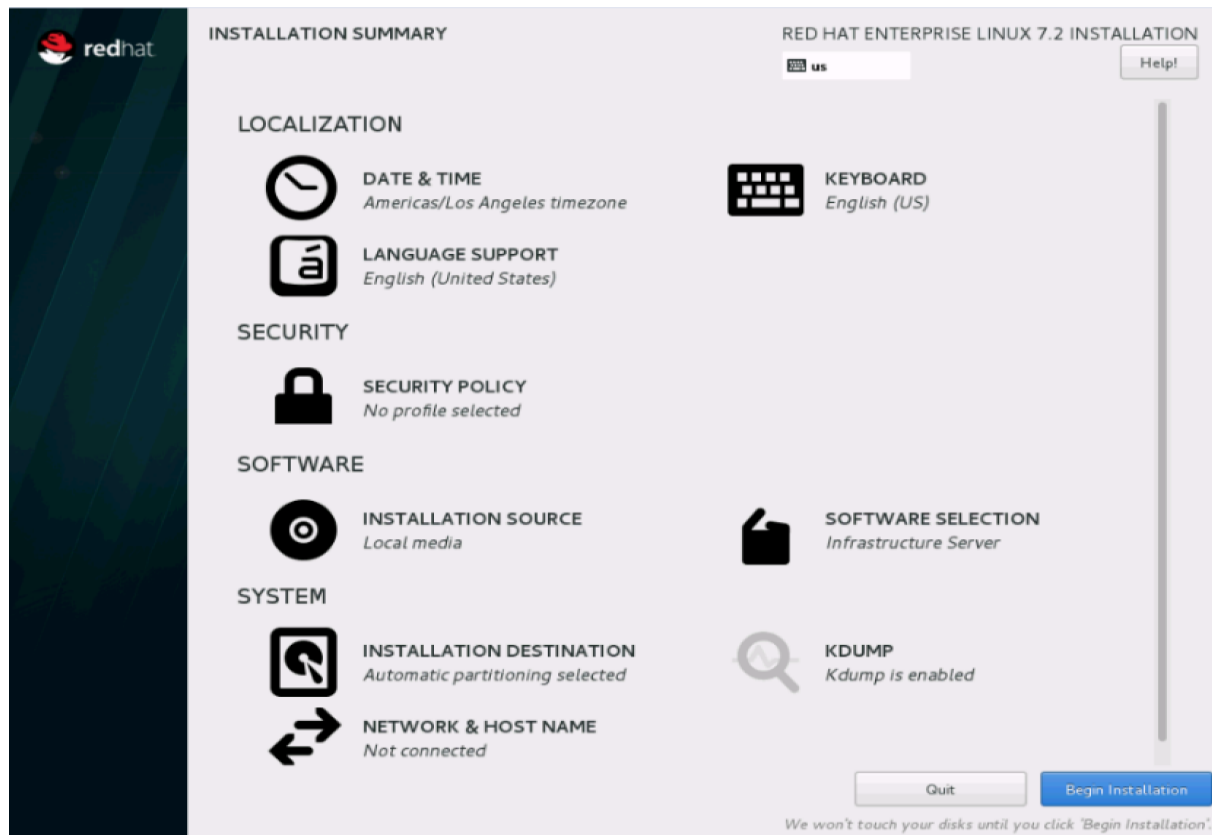
Figure 99 Software Selection



The Installation Summary window returns (Figure 98).

33. Click on `Network & Hostname`.

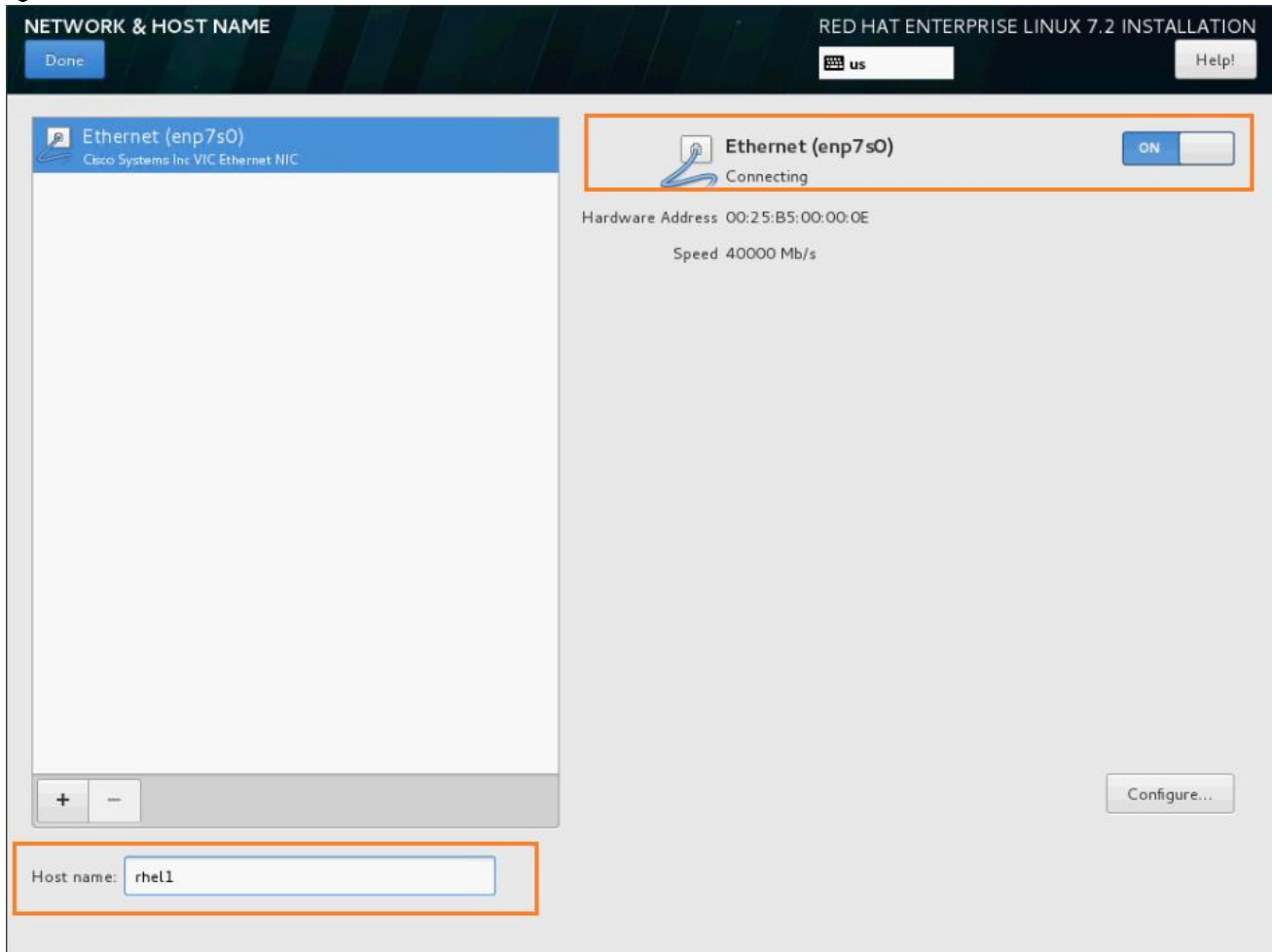
Figure 100 Network and Host Name Window



Configure Hostname and Networking for the Host (Figure 101).

34. Type in the hostname (for example, admin1).

Figure 101 Network and Host Name



35. Click on `Configure` to open the Network Connectivity window (Figure 102).

36. Click on `IPV4Settings`.

Figure 102 Network Connectivity Window

The screenshot shows the 'Editing enp7s0' window. At the top, the connection name is 'enp7s0'. Below this are tabs for 'General', 'Ethernet', '802.1x Security', 'DCB', 'IPv4 Settings', and 'IPv6 Settings'. The 'IPv4 Settings' tab is selected. Under 'Method', a dropdown menu is set to 'Manual'. Below this is an 'Addresses' section with a table:

Address	Netmask	Gateway	
			Add
			Delete

Below the table are fields for 'DNS servers:', 'Search domains:', and 'DHCP client ID:'. There is a checkbox labeled 'Require IPv4 addressing for this connection to complete' which is currently unchecked. A 'Routes...' button is located at the bottom right of the main settings area. At the very bottom of the window are 'Cancel' and 'Save' buttons.

37. Change the Method to Manual and click Add.

Figure 103 shows the Add Details pop up window.

38. Enter the IP Address, Netmask and Gateway details. Click Add after each addition.

Figure 103 Add IP Address, Netmask and Gateway Details

Editing enp7s0

Connection name:

General Ethernet 802.1x Security DCB **IPv4 Settings** IPv6 Settings

Method:

Addresses

Address	Netmask	Gateway	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>
			<input type="button" value="Delete"/>

DNS servers:

Search domains:

DHCP client ID:

Require IPv4 addressing for this connection to complete

Editing enp7s0

Connection name:

General Ethernet 802.1x Security DCB **IPv4 Settings** IPv6 Settings

Method:

Addresses

Address	Netmask	Gateway	
<input type="text" value="172.16.46.11"/>	<input type="text" value="24"/>	<input type="text" value="172.16.46.1"/>	<input type="button" value="Add"/>
			<input type="button" value="Delete"/>

DNS servers:

Search domains:

DHCP client ID:

Require IPv4 addressing for this connection to complete

39. Click **Save**.

40. Repeat IPv4 configuration process to configure the other connection.

41. Update the hostname and turn Ethernet ON. Click Done to return to the main menu.

The Installation Summary window opens (Figure 104).

42. Click `Begin Installation` in the main menu.

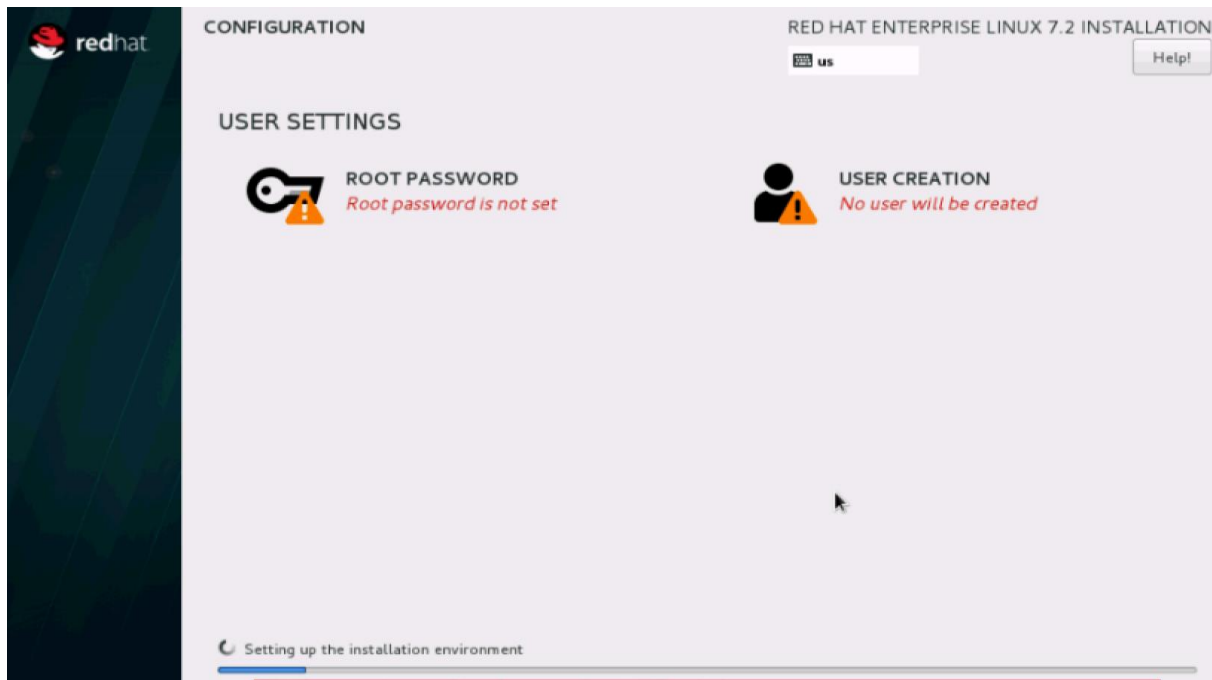
Figure 104 Installation Summary Window



A new window opens (Figure 105).

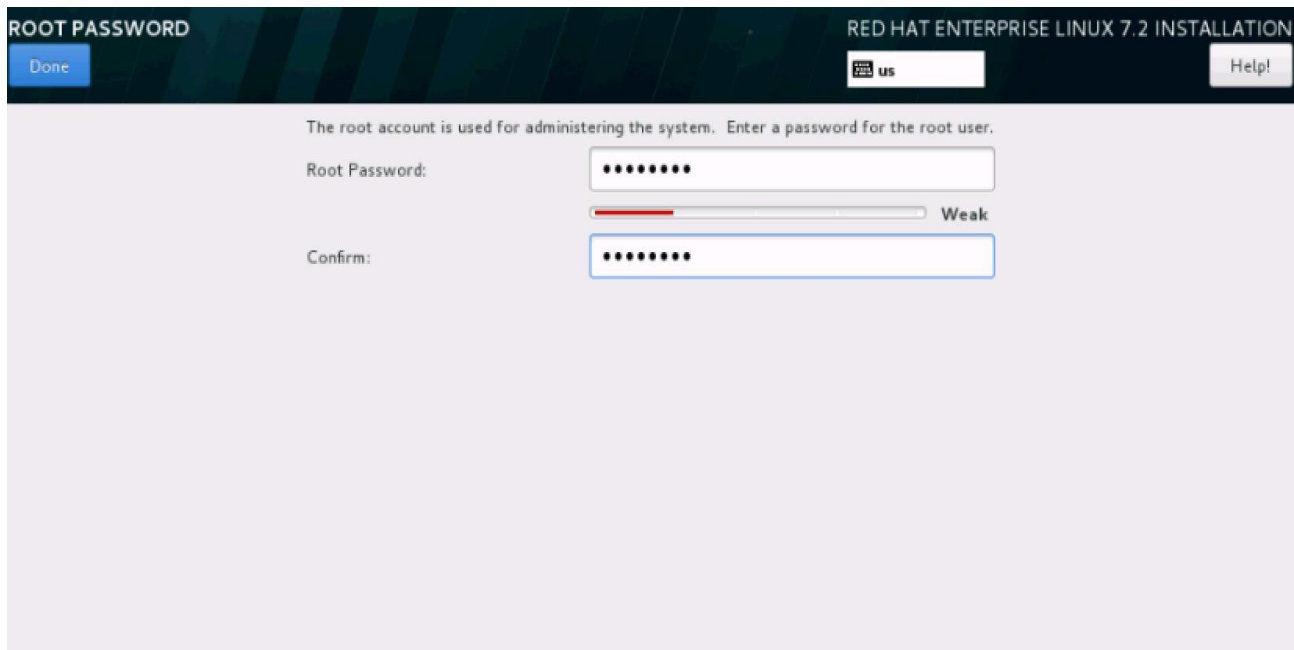
43. Select `Root Password` in the User Settings.

Figure 105 Select Root Password



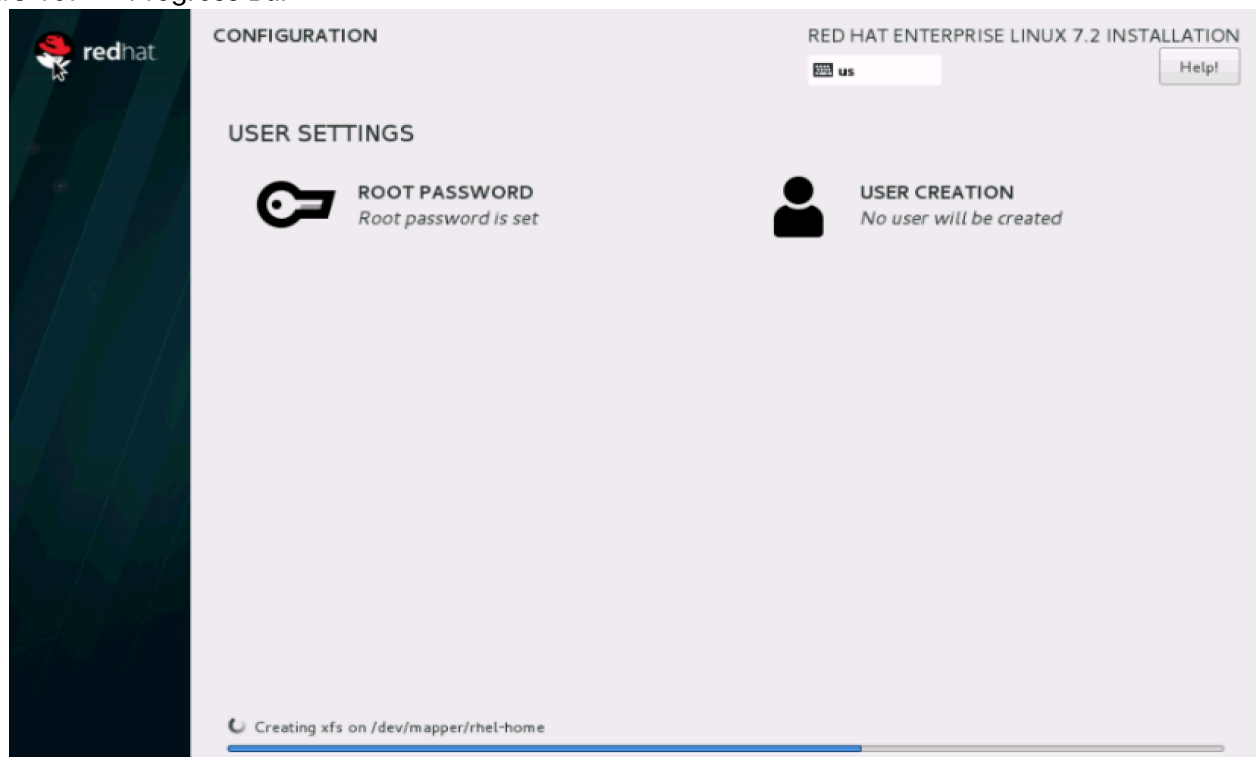
44. On the next screen (Figure 106), enter the Root Password and click done.

Figure 106 Enter the Root Password



A progress window will open (Figure 107).

Figure 107 Progress Bar



45. Once the installation is complete, reboot the system.

46. Repeat steps 1 to 43 to install Red Hat Enterprise Linux 7.2 on the other servers.



Note: The OS installation and configuration of the nodes that is mentioned above can be automated through PXE boot or third party tools.

The table below shows how the hosts are assigned with their host names. Within the UCS domain, the eth0 network (that is 50.1.1.X subnet) over Fabric A is used as the primary network for all Splunk related data traffic except the replication traffic. The Splunk index replication-related data traffic will be configured to use eth1 over Fabric B.

For example, the host names associated with the various interfaces of an indexer that is idx1 are as follows:

- eth0:
 - Used to ingest the data streaming in from forwarders, and for traffic between the search head and indexers.
 - Used for management traffic such as SSH, Web UI, NTP sync.
 - Hostname is idx1.
 - Configured with Platinum QOS policy.
- eth1:

- Used by the indexers to replicate indexes across each other.
- Hostname is idx-rep.
- Configured with Platinum QOS policy.

Table 6 Hostnames and IP Addresses

Hostname	eth0 Management Network, Data Ingestion	eth1 Data Replication Hostname: <hostname>-rep
admin1	50.1.1.101	192.168.11.101
admin2	50.1.1.102	192.168.11.102
sh1	50.1.1.111	192.168.11.111
sh2	50.1.1.112	192.168.11.112
sh3	50.1.1.113	192.168.11.113
idx1	50.1.1.121	192.168.11.121
idx2	50.1.1.122	192.168.11.122
idx3	50.1.1.123	192.168.11.123
idx4	50.1.1.124	192.168.11.124
idx5	50.1.1.125	192.168.11.125
idx6	50.1.1.126	192.168.11.126
idx7	50.1.1.127	192.168.11.127
idx8	50.1.1.128	192.168.11.128

Post OS Install Configuration

Choose one of the admin nodes of the cluster for management such as installation, cluster parallel shell, creating a local Red Hat repo, and others. In this document, we use admin1 for this purpose.

Creating Red Hat Enterprise Linux (RHEL) 7.2 Local Repo

To create a repository using RHEL DVD or ISO on the admin node (in this deployment, admin1 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 on to `admin1`. Create a directory that would contain the repository.

```
mkdir -p /var/www/html/rhelrepo
```

2. Copy the contents of the Red Hat DVD to `/var/www/html/rhelrepo`. Alternatively, if you have access to a Red Hat ISO Image, copy the ISO file to `admin1` with the command shown below:

```
scp rhel-server-7.2-x86_64-dvd.iso admin1:/root/
```



Note: Make sure the Red Hat ISO file is located in your present working directory. Use the IP address of the admin node instead of the hostname `admin1` if hostnames have not been configured on this computer.

3. Create the mount directory for the RedHat ISO.

```
mkdir -p /mnt/rheliso
```

4. Mount the Red Hat ISO image.

```
mount -t iso9660 -o loop /root/rhel-server-7.2-x86_64-dvd.iso /mnt/rheliso/
```

5. Copy the contents of the ISO to the `/var/www/html/rhelrepo` directory.

```
cp -r /mnt/rheliso/* /var/www/html/rhelrepo
```

```
[root@admin1 ~]# mkdir -p /var/www/html/rhelrepo
[root@admin1 ~]# mkdir -p /mnt/rheliso
[root@admin1 ~]# mount -t iso9660 -o loop /root/rhel-server-7.2-x86_64-dvd.iso /mnt/
rheliso
mount: /dev/loop0 is write-protected, mounting read-only
[root@admin1 ~]# cp -r /mnt/rheliso/* /var/www/html/rhelrepo
```

6. On `admin1`, create a `.repo` file to enable the use of the `yum` command.

```
vi /var/www/html/rhelrepo/rheliso.repo
[rhel7.2]
```



```
name=Red Hat Enterprise Linux 7.2
baseurl=http://50.1.1.101/rhelrepo
gpgcheck=0
enabled=1
```

7. Now copy the `rheliso.repo` file from `/var/www/html/rhelrepo` to `/etc/yum.repos.d` on `admin1`.

```
cp /var/www/html/rhelrepo/rheliso.repo /etc/yum.repos.d/
```



Note: Based on this repository file, yum requires httpd to be running on `admin1` for other nodes to access the repository.

Creating the Red Hat Repository Database

1. Install the `createrepo` package on the admin node (`admin1`). Use it to generate the repository database(s) for the local copy of the RHEL DVD contents.

```
yum -y install createrepo
```

```
[root@admin1 ~]# vi /etc/yum.repos.d/rheliso.repo
[root@admin1 ~]# yum -y install createrepo
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
rhel7.2 | 4.1 kB 00:00:00
(1/2): rhel7.2/group_gz | 136 kB 00:00:00
(2/2): rhel7.2/primary_db | 3.6 MB 00:00:00
Resolving Dependencies
--> Running transaction check
--> Package createrepo.noarch 0:0.9.9-23.el7 will be installed
--> Processing Dependency: deltarpm for package: createrepo-0.9.9-23.el7.noarch
--> Processing Dependency: python-deltarpm for package: createrepo-0.9.9-23.el7.noarch
--> Running transaction check
--> Package deltarpm.x86_64 0:3.6-3.el7 will be installed
--> Package python-deltarpm.x86_64 0:3.6-3.el7 will be installed
--> Finished Dependency Resolution
```

```

Dependencies Resolved
=====
Package                Arch                Version              Repository            Size
-----
Installing:
createrepo              noarch              0.9.9-23.e17        rhel7.2               92 k
Installing for dependencies:
deltarpm                x86_64              3.6-3.e17           rhel7.2               82 k
python-deltarpm         x86_64              3.6-3.e17           rhel7.2               31 k
Transaction Summary
=====
Install 1 Package (+2 Dependent packages)

Total download size: 205 k
Installed size: 553 k
Downloading packages:
-----
Total                                                                68 MB/s | 205 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : deltarpm-3.6-3.e17.x86_64                                1/3
  Installing : python-deltarpm-3.6-3.e17.x86_64                        2/3
  Installing : createrepo-0.9.9-23.e17.noarch                          3/3
rhel7.2/productid | 1.6 kB  00:00:00
  Verifying  : python-deltarpm-3.6-3.e17.x86_64                        1/3
  Verifying  : deltarpm-3.6-3.e17.x86_64                               2/3
  Verifying  : createrepo-0.9.9-23.e17.noarch                          3/3

Installed:
createrepo.noarch 0:0.9.9-23.e17

Dependency Installed:
deltarpm.x86_64 0:3.6-3.e17          python-deltarpm.x86_64 0:3.6-3.e17

Complete!
[root@admin1 ~]# █

```

2. Run the `createrepo` command on the RHEL repository to create the repository database on the admin node:

```
cd /var/www/html/rhelrepo
```

```
createrepo .
```

```

[root@admin1 ~]# cd /var/www/html/rhelrepo
[root@admin1 rhelrepo]# createrepo .
Spawning worker 0 with 196 pkgs
Spawning worker 1 with 196 pkgs
Spawning worker 2 with 196 pkgs
Spawning worker 3 with 196 pkgs
Spawning worker 4 with 196 pkgs
Spawning worker 5 with 196 pkgs
Spawning worker 6 with 196 pkgs
Spawning worker 7 with 195 pkgs
Spawning worker 8 with 195 pkgs
Spawning worker 9 with 195 pkgs
Spawning worker 10 with 195 pkgs
Spawning worker 11 with 195 pkgs
Spawning worker 12 with 195 pkgs
Spawning worker 13 with 195 pkgs
Spawning worker 14 with 195 pkgs
Spawning worker 15 with 195 pkgs
Spawning worker 16 with 195 pkgs
Spawning worker 17 with 195 pkgs
Spawning worker 18 with 195 pkgs
Spawning worker 19 with 195 pkgs
Spawning worker 20 with 195 pkgs
Spawning worker 21 with 195 pkgs
Spawning worker 22 with 195 pkgs
Spawning worker 23 with 195 pkgs
Workers Finished
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete
[root@admin1 rhelrepo]# █

```

Configuring /etc/hosts

To configure /etc/hosts on the admin node, complete the following steps:

1. Login to the admin node (admin1).

```
ssh 50.1.1.101
```

2. Populate the host file with IP addresses and corresponding hostnames. We will later copy this over to the other nodes.

On the Admin Node (admin1)

```
vi /etc/hosts
```

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
```

```
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6
50.1.1.101   admin1
192.168.11.101  admin1-rep
50.1.1.102   admin2
192.168.11.102  admin2-rep
50.1.1.121   idx1
192.168.11.103  idx1-rep
50.1.1.122   idx2
192.168.11.104  idx2-rep
50.1.1.123   idx3
192.168.11.105  idx3-rep
50.1.1.124   idx4
192.168.11.106  idx4-rep
50.1.1.125   idx5
192.168.11.107  idx5-rep
50.1.1.126   idx6
192.168.11.108  idx6-rep
50.1.1.127   idx7
192.168.11.109  idx7-rep
50.1.1.128   idx8
192.168.11.110  idx8-rep
50.1.1.111   sh1
192.168.11.111  sh1-rep
50.1.1.112   sh2
192.168.11.112  sh2-rep
50.1.1.113   sh3
192.168.11.113  sh3-rep
```

Setting Up Password-less Login

To manage all of the cluster's nodes from the admin node, set up password-less login. It assists in automating common tasks with ClusterShell, a cluster-wide parallel shell command utility, and shell scripts without having to use passwords.

Once Red Hat Linux is installed across all the nodes in the cluster, to enable password-less login across all the nodes, complete the following steps:

1. Login to the admin node (admin1). For example: `ssh 50.1.1.101`
2. Run the `ssh-keygen` command to create both public and private keys on the admin node.

```
[root@admin1 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
42:4e:45:fb:0f:1d:a5:7b:4c:0a:59:13:12:3b:bc:93 root@admin1
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      . o o . + . . |
|      . o = +      |
|      o . * o .    |
|      + . * *      |
|      o S E + o    |
|      .      + .   |
|      .            |
+-----+
[root@admin1 ~]#
```

3. Then run the following script from the admin node to copy the public key `id_rsa.pub` to all the nodes of the cluster. `ssh-copy-id` appends the keys to the remote-host's `.ssh/authorized_keys`.

```
for host in admin1 admin2 idx1 idx2 idx3 idx4 idx5 idx6 idx7 idx8 sh1 sh2 sh3;
do echo -n "$host -> "; ssh-copy-id -i ~/.ssh/id_rsa.pub $host; done
```

4. Enter **yes** at the prompt `Are you sure you want to continue connecting (yes/no)?` Enter the password of the remote host.



Note: The admin node's `/etc/hosts` should be copied over to all other servers by using the ClusterShell command after it is installed. This will be done in the section "Set Up All Nodes to Use the RHEL Repository".

Setting Up ClusterShell

ClusterShell (or `clush`) is a cluster-wide shell to run commands on several hosts in parallel. It is available from the EPEL (Extra Packages for Enterprise Linux) repository.

To download ClusterShell and install it on admin1, complete the following steps

1. From a server connected to the internet, download ClusterShell.

```
wget
```

```
ftp://ftp.pbone.net/mirror/download.fedora.redhat.com/pub/fedora/epel/7/x86\_64/c/clustershell-1.7.2-1.el7.noarch.rpm
```

```
[root@BDPOC-JB ~]# wget ftp://ftp.pbone.net/mirror/download.fedora.redhat.com/pub/fedora/epel/7/x86_64/c/clustershell-1.7.2-1.el7.noarch.rpm
--2016-10-26 11:42:17--  ftp://ftp.pbone.net/mirror/download.fedora.redhat.com/pub/fedora/epel/7/x86_64/c/clustershell-1.7.2-1.el7.noarch.rpm
      => "clustershell-1.7.2-1.el7.noarch.rpm"
Resolving ftp.pbone.net... 85.14.85.4
Connecting to ftp.pbone.net|85.14.85.4|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.      ==> PWD ... done.
==> TYPE I ... done.   ==> CWD (1) /mirror/download.fedora.redhat.com/pub/fedora/epel/7/x86_64/c ... done.
==> SIZE clustershell-1.7.2-1.el7.noarch.rpm ... 371704
==> PASV ... done.     ==> RETR clustershell-1.7.2-1.el7.noarch.rpm ... done.
Length: 371704 (363K) (unauthoritative)

100%[=====>] 371,704      326K/s   in 1.1s

2016-10-26 11:42:21 (326 KB/s) - "clustershell-1.7.2-1.el7.noarch.rpm" saved [371704]

[root@BDPOC-JB ~]# █
```

2. Copy ClusterShell to admin1 by executing the following command:

```
scp clustershell-1.7.2-1.el7.noarch.rpm admin1:/root/
```

3. Log in to admin1. Then install ClusterShell by executing the following command:

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

Installing ClusterShell (Output truncated)

```

[root@admin1 ~]# yum -y install clustershell-1.7.2-1.el7.noarch.rpm
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Examining clustershell-1.7.2-1.el7.noarch.rpm: clustershell-1.7.2-1.el7.noarch
Marking clustershell-1.7.2-1.el7.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package clustershell.noarch 0:1.7.2-1.el7 will be installed
--> Processing Dependency: PyYAML for package: clustershell-1.7.2-1.el7.noarch
--> Running transaction check
---> Package PyYAML.x86_64 0:3.10-11.el7 will be installed
--> Processing Dependency: libyaml-0.so.2()(64bit) for package: PyYAML-3.10-11.el7.x86_64
--> Running transaction check
---> Package libyaml.x86_64 0:0.1.4-11.el7_0 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

-----
Package                Arch             Version          Repository
-----
Installing:
clustershell           noarch           1.7.2-1.el7     /clustershell-1.7.2-1.el7.noarch
Installing for dependencies:
PyYAML                 x86_64           3.10-11.el7     rhel7.2
libyaml                x86_64           0.1.4-11.el7_0  rhel7.2
-----
Transaction Summary
-----
Install 1 Package (+2 Dependent packages)

Total size: 1.9 M
Total download size: 208 k
Installed size: 2.4 M
Downloading packages:
-----
Total
Running transaction check                                     81 MB/s | 208 kB

```

4. Edit the `/etc/clustershell/groups.d/local.cfg` file to create special groups that target cluster wide commands to a specific set of nodes in the cluster. The "all" group is the set of hosts taken when running `clush` with the `'-a'` option. The other three groups are grouped by their role in the Splunk deployment. Copy and paste the content below and save the groups configuration file.

```

vi /etc/clustershell/groups

all: admin[1-2],sh[1-3],idx[1-8]

admins: admin[1-2]

indexers: idx[1-8]

searchheads: sh[1-3]

```

```
# ClusterShell groups config local.cfg
#
# Replace /etc/clusterhell/groups
#
# Note: file auto-loaded unless /etc/clusterhell/groups is present
#
# See also groups.d/cluster.yaml.example for an example of multiple
# sources single flat file setup using YAML syntax.
#
# Feel free to edit to fit your needs.
all: admin[1-2],sh[1-3],idx[1-8]
admins: admin[1-2]
indexers: idx[1-8]
searchheads: sh[1-3]
```

For more information and documentation on ClusterShell, visit <https://github.com/cea-hpc/clusterhell/wiki/UserAndProgrammingGuide>.

Installing httpd

Setting up the RHEL repository on the admin node requires httpd. This section describes the process of setting one up.

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

```
yum -y install httpd
```

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

```
vi /etc/httpd/conf/httpd.conf
```

```
ServerName 50.1.1.101:80
```

```
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
#ServerName www.example.com:80
ServerName 50.1.1.101:80
```

3. Start `httpd`.

```
service httpd start
```



```
chkconfig httpd on
```

Disabling the Linux Firewall

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

```
clush -a -b "service firewalld stop"
```

```
clush -a -b "systemctl disable firewalld"
```

```
[root@admin1 ~]# clush -a -b service iptables stop
-----
admin[1-2],idx[1-8],sh[1-3],storage1 (14)
-----
iptables: Setting chains to policy ACCEPT: filter [ OK ]
iptables: Flushing firewall rules: [ OK ]
iptables: Unloading modules: [ OK ]
[root@admin1 ~]# clush -a -b chkconfig iptables off
```



Note: The user could re-configure the IP tables' settings in order to match the requirements of his/her particular deployment and turn the service back on. Consult Splunk documentation to determine the appropriate IP tables' settings.

Disabling SELinux

Security-Enhanced Linux (SELinux) must be disabled during the install procedure and cluster setup. SELinux can be enabled after installation and while the cluster is running. To disable SELINUX on all nodes, complete the following steps:



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

1. To disable SELinux, edit `/etc/selinux/config` and change the `SELINUX` line to `SELINUX=disabled`. This can be accomplished for all nodes with the following command:

```
clush -a -b "sed -i 's/SELINUX=enforcing/SELINUX=disabled/g'
/etc/selinux/config"
```

```
[root@admin1 ~]# clush -a -b "sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config"
[root@admin1 ~]# clush -a -b cat /etc/selinux/config
-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
# 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
```

2. Use the following command to enter permissive mode:

```
clush -a -b "setenforce 0"
```



Note: The command above may fail if SELinux is already disabled.

3. Reboot all servers to finish disabling SELinux. Confirm the SELinux status using the following command:

```
clush -a -b sestatus
```

```
[root@admin1 ~]# clush -a -b sestatus
-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
SELinux status:                               disabled
```

Set Up all Nodes to Use the RHEL Repository



Note: Based on this repo file yum requires httpd to be running on admin1 for other nodes to access the repository.

1. Copy the rheliso.repo to all the nodes of the cluster.

```
clush -a -b -c /etc/yum.repos.d/rheliso.repo
```

2. To make use of repository files on admin1 without httpd, edit the baseurl of the repository file `/etc/yum.repos.d/rheliso.repo` to point to the repository location in the file system.



Note: This step is needed to install software on the admin node (admin1) using the repository (such as httpd, createrepo, etc)

```
vi /etc/yum.repos.d/rheliso.repo
[rhel7.2]

name=Red Hat Enterprise Linux 7.2

baseurl=file:///var/www/html/rhelrepo

gpgcheck=0

enabled=1
```

3. Copy the /etc/hosts file to all nodes.

```
clush -a -B -x admin1 -c /etc/hosts
```

4. Purge the yum caches.

```
clush -a -B yum clean all
```

```
clush -a -B yum repolist
```

```
[root@admin1 ~]# clush -a -b -c /etc/yum.repos.d/rheliso.repo
[root@admin1 ~]# clush -a -B -x admin1 -c /etc/hosts
[root@admin1 ~]# clush -a -B yum clean all
-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Cleaning repos: rhel7.2
Cleaning up everything
[root@admin1 ~]# clush -a -B yum repolist
-----
admin2,idx[1-8],sh[1-3] (12)
-----
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
file:///var/www/html/rhelrepo/repodata/repomd.xml: [Errno 14] curl#37 - "Couldn't open file /var/www/html/rhelrepo/repodata/repomd.xml"
Trying other mirror.
file:///var/www/html/rhelrepo/repodata/repomd.xml: [Errno 14] curl#37 - "Couldn't open file /var/www/html/rhelrepo/repodata/repomd.xml"
Trying other mirror.
repo id                repo name                status
rhel7.2                Red Hat Enterprise Linux 7.2          0
repolist: 0
-----
admin1
-----
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
repo id                repo name                status
rhel7.2                Red Hat Enterprise Linux 7.2          4,687
repolist: 4,687
```

Upgrading the Cisco Network Driver for VIC1387

The latest Cisco Network driver is required for performance and updates. To download the latest drivers go to the link below:

[https://software.cisco.com/download/release.html?mdfid=283862063&release=2.0\(13\)&relind=AVAILABLE&flowid=25886&softwareid=283853158&rellifecycle=&reltype=latest](https://software.cisco.com/download/release.html?mdfid=283862063&release=2.0(13)&relind=AVAILABLE&flowid=25886&softwareid=283853158&rellifecycle=&reltype=latest)

1. In the ISO image, the required driver `kmod-enic-2.3.0.30-rhel7u2.el7.x86_64.rpm` can be located at `\Network\Cisco\VIC\RHEL\RHEL7.2`.

2. From a node connected to the Internet, download, extract and transfer `kmod-enic-2.3.0.30-rhel7u2.el7.x86_64.rpm` to `admin1` (admin node).
3. Install the rpm on all nodes of the cluster using the following clush commands. For this example the rpm is assumed to be in present working directory of `admin1`.

```
clush -a -b -c kmod-enic-2.3.0.30-rhel7u2.el7.x86_64.rpm
```

```
clush -a -b "rpm -ivh kmod-enic-2.3.0.30-rhel7u2.el7.x86_64.rpm"
```

4. Ensure that the above installed version of `kmod-enic` driver is being used on all nodes by running the command "`modinfo enic`" on all nodes.

```
clush -a -B "modinfo enic | head -5"
```

```
[root@admin1 ~]# clush -a -B "modinfo enic | head -5"
-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
filename:      /lib/modules/3.10.0-327.el7.x86_64/weak-updates/enic/enic.ko
version:      2.3.0.30
license:      GPL v2
author:       Scott Feldman <scofeldm@cisco.com>
description:  Cisco VIC Ethernet NIC Driver
```

5. It is recommended to download and install the `kmod-megaraid` driver for higher performance. The RPM can be found in the same package at `\Storage\LSI\Cisco_Storage_12G_SAS_RAID_controller\RHEL\RHEL7.2`

Installing xfsprogs

The `xfsprogs` package contains administration and debugging tools for the XFS file system. To install `xfsprogs` on all nodes, complete the following steps:

1. From the admin node `admin1`, run the command below to install `xfsprogs` on all the nodes for `xfs` filesystem.

```
clush -a -B yum -y install xfsprogs
```

```

Loaded plugins: product-id, search-disabled-repos, security, subscription-
                : manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to
register.
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package xfsprogs.x86_64 0:3.1.1-19.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch          Version        Repository      Size
=====
Installing:
xfsprogs                x86_64        3.1.1-19.el6   RHEL6.8         725 k

Transaction Summary
=====
Install      1 Package(s)

Total download size: 725 k
Installed size: 3.2 M
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : xfsprogs-3.1.1-19.el6.x86_64                1/1
  Verifying  : xfsprogs-3.1.1-19.el6.x86_64                1/1

Installed:
  xfsprogs.x86_64 0:3.1.1-19.el6

Complete!

```

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 located in the admin node (admin1). Configuring NTP is critical for any clustered application. If server clocks in the cluster drift out of sync, serious problems will occur.

Installing an internal NTP server keeps the cluster synchronized even when an outside NTP server is inaccessible.

1. Install NTP on all nodes, if needed.

```
clush -a -b "yum -y install ntp"
```

2. Configure `/etc/ntp.conf` on the admin node with the following content:

```

vi /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
```

3. Create `/tmp/ntp.conf` on the admin node and copy it to all nodes.

```
vi /tmp/ntp.conf

server 50.1.1.101

driftfile /var/lib/ntp/drift

restrict 127.0.0.1

restrict -6 ::1

includefile /etc/ntp/crypto/pw

keys /etc/ntp/keys
```

4. Copy `/tmp/ntp.conf` from the admin node to `/etc/ntp.conf` of all the other nodes by executing the following command in the admin node (admin1).

```
clush -a -B -x admin1 -c /tmp/ntp.conf --dest=/etc/ntp.conf
```

Verifying contents of `ntp.conf`

```
[root@admin1 ~]# clush -a -B cat /etc/ntp.conf
-----
admin2, idx[1-8], sh[1-3] (12)
-----
server 50.1.1.101
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys

-----
admin1
-----
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
```

5. Start the NTP service on the admin node (admin1).

```
service ntpd start
```

6. Run the following to synchronize the time and restart NTP daemon on all nodes.

```
clush -a -B "yum install -y ntpdate"
```

```
clush -a -b -x admin1 "service ntpd stop"

clush -a -b -x admin1 "ntpddate 50.1.1.101"

clush -a -b "service ntpd start"
```

7. Ensure restart of NTP daemon across reboots.

```
clush -a -b "chkconfig ntpd on"
```

Enabling Syslog

To preserve logs regarding killed processes or failed job, enable Syslog on each node. Versions such as syslog-ng and rsyslog are used, making it more difficult to be sure that a syslog daemon is present.

To confirm that the service is properly configured, run the following commands (the output below is truncated):

```
clush -B -a rsyslogd -v
clush -B -a service rsyslog status
```

```
[root@admin1 ~]# clush -B -a rsyslogd -v
-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
rsyslogd 7.4.7, compiled with:
    FEATURE_REGEX:                Yes
    FEATURE_LARGEFILE:             No
    GSSAPI Kerberos 5 support:     Yes
    FEATURE_DEBUG (debug build, slow code): No
    32bit Atomic operations supported: Yes
    64bit Atomic operations supported: Yes
    Runtime Instrumentation (slow code): No
    uuid support:                  Yes

See http://www.rsyslog.com for more information.
[root@admin1 ~]# clush -B -a service rsyslog status
-----
admin1
-----
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 Thu 2016-10-27 07:40:09 PDT; 1h 14min ago
 Main PID: 881 (rsyslogd)
   CGroup: /system.slice/rsyslog.service
           └─881 /usr/sbin/rsyslogd -n

Oct 27 07:40:09 admin1 systemd[1]: Starting System Logging Service...
Oct 27 07:40:09 admin1 systemd[1]: Started System Logging Service.
-----
```

Setting Ulimit

In Linux, the 'nofile' property 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 users root and splunk.



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

1. Set the "nofile" properties of root and splunk users to 64000 by editing the `/etc/security/limits.conf` on the admin node. Add the following lines to this file.

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

2. Copy the `/etc/security/limits.conf` file from admin node (admin1) to all the nodes using the following command. The second command verifies the contents of the file on all nodes.

```
clush -a -B -c /etc/security/limits.conf
clush -a -B grep 64000 /etc/security/limits.conf
```

```
[root@admin1 ~]# clush -a -B -c /etc/security/limits.conf
[root@admin1 ~]# clush -a -B grep 64000 /etc/security/limits.conf
-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
root soft nofile 64000
root hard nofile 64000
splunk soft nofile 64000
splunk hard nofile 64000
```

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

```
clush -B -a ulimit -n
```

```
[root@admin1 ~]# clush -B -a ulimit -n
-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
64000
```



Note: 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.

Set TCP Retries

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

1. Edit the file `/etc/sysctl.conf` on `admin1` and add the following line:

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

2. Copy the `/etc/sysctl.conf` file from the admin node (`admin1`) to all the other nodes using the following command:

```
clush -a -b -c /etc/sysctl.conf
```

3. Load the settings from default `sysctl` file `/etc/sysctl.conf` by running

```
clush -B -a sysctl -p
```

```
[root@admin1 ~]# clush -a -b -c /etc/sysctl.conf
[root@admin1 ~]# clush -B -a sysctl -p
-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
net.ipv4.tcp_retries2 = 5
```

Configure VM Swapping

`vm.swappiness`, with a value from 0 to 100, controls the degree to 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.

```
clush -a -B "echo vm.swappiness=1 >> /etc/sysctl.conf"
```

Disable IPv6 Defaults

IPv4 addresses are used, so IPv6 should be disabled. To do so, complete the following steps.

1. Disable IPv6 with the following commands:

```
clush -a -b "echo net.ipv6.conf.all.disable_ipv6 = 1 >> /etc/sysctl.conf"
```

```
clush -a -b "echo net.ipv6.conf.default.disable_ipv6 = 1 >> /etc/sysctl.conf"
```

```
clush -a -b "echo net.ipv6.conf.lo.disable_ipv6 = 1 >> /etc/sysctl.conf"
```

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

```
clush -a -B "sysctl -p"
```

```
[root@admin1 ~]# clush -a -B "echo vm.swappiness=1 >> /etc/sysctl.conf"
[root@admin1 ~]# clush -a -b "echo net.ipv6.conf.all.disable_ipv6 = 1 >> /etc/sysctl.conf"
[root@admin1 ~]# clush -a -b "echo net.ipv6.conf.default.disable_ipv6 = 1 >> /etc/sysctl.conf"
[root@admin1 ~]# clush -a -b "echo net.ipv6.conf.lo.disable_ipv6 = 1 >> /etc/sysctl.conf"
[root@admin1 ~]# clush -a -B "sysctl -p"

-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
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
```

Disable Transparent Huge Pages

Disabling Transparent Huge Pages (THP) reduces elevated CPU usage caused by THP.

1. From the admin node, run the following commands:

```
clush -a -b "echo never > /sys/kernel/mm/transparent_hugepage/enabled"
clush -a -b "echo never > /sys/kernel/mm/transparent_hugepage/defrag"
```

2. Run the commands above every time the Linux system starts up. Add these commands to `/etc/rc.local`, so they are executed automatically upon every reboot.

3. From the admin node, run the following commands:

```
rm -f /root/thp_disable
echo "echo never > /sys/kernel/mm/transparent_hugepage/enabled" > /root/thp_disable
echo "echo never > /sys/kernel/mm/transparent_hugepage/defrag " >> /root/thp_disable
```

4. Copy the file over to all the nodes.

```
clush -a -b -c /root/thp_disable
```

5. Append the content of file `thp_disable` to `/etc/rc.local`.

```
clush -a -b "cat /root/thp_disable >> /etc/rc.local"
```

```
[root@admin1 ~]# clush -a -b "echo never > /sys/kernel/mm/transparent_hugepage/enabled"
[root@admin1 ~]# clush -a -b "echo never > /sys/kernel/mm/transparent_hugepage/defrag"
[root@admin1 ~]# rm -f /root/thp_disable
[root@admin1 ~]# echo "echo never > /sys/kernel/mm/transparent_hugepage/enabled" > /root/thp_disable
[root@admin1 ~]# echo "echo never > /sys/kernel/mm/transparent_hugepage/defrag " >> /root/thp_disable
[root@admin1 ~]# clush -a -b -c /root/thp_disable
[root@admin1 ~]# clush -a -b "cat /root/thp_disable >> /etc/rc.local"
```

Installing the LSI StorCLI Utility on All Indexers

This section describes the steps to configure non-OS disk drives using the StorCLI command. HDDs used for cold storage will be configured as RAID10. SSDs used for hot and warm storage will be configured as RAID5.

1. To download StorCLI, go to <http://www.avagotech.com/support/download-search> and search for StorCLI. Go to Latest MegaRAID StorCLI and check that the version is 1.19.04.
2. Extract the zip file and copy storcli-1.19.04-1.noarch.rpm from the Linux directory.
3. Download StorCLI and its dependencies and transfer to the admin node.

```
scp storcli-1.19.04-1.noarch.rpm admin1:/tmp/
```

4. Copy the StorCLI rpm to all the indexers using the following command:

```
clush --group=indexers -c /tmp/storcli-1.19.04-1.noarch.rpm
```

5. Run the command below to install StorCLI all the indexers.

```
clush --group=indexers rpm -ivh /tmp/storcli-1.19.04-1.noarch.rpm
```

```
[root@admin1 ~]# clush --group=indexers -c /tmp/storcli-1.19.04-1.noarch.rpm
[root@admin1 ~]# clush --group=indexers rpm -ivh /tmp/storcli-1.19.04-1.noarch.rpm
idx1: Preparing... #####
idx1: Updating / installing...
idx2: Preparing... #####
idx3: Preparing... #####
idx2: Updating / installing...
idx3: Updating / installing...
idx8: Preparing... #####
idx8: Updating / installing...
idx5: Preparing... #####
idx5: Updating / installing...
idx4: Preparing... #####
idx4: Updating / installing...
idx7: Preparing... #####
idx7: Updating / installing...
idx6: Preparing... #####
idx6: Updating / installing...
idx8: storcli-1.19.04-1 #####
idx3: storcli-1.19.04-1 #####
idx2: storcli-1.19.04-1 #####
idx1: storcli-1.19.04-1 #####
idx5: storcli-1.19.04-1 #####
idx4: storcli-1.19.04-1 #####
idx7: storcli-1.19.04-1 #####
idx6: storcli-1.19.04-1 #####
```

Configuring the Virtual Drive on the Indexers

1. Create a script named `raid-server1.sh` on the admin node. It will be run on the first server of each of the four chassis.

```
vi /root/raid-server1.sh
```

2. Paste the following contents into the file and save it.

```
/opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid5 drives=$1:41-48 WT nora direct Strip=128
```

```
/opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid10 drives=$1:1-20
pdperarray=10 WB ra direct Strip=128
```

```
/opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid5 drives=$1:41-48 WT nora direct Strip=128
/opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid10 drives=$1:1-20 pdperarray=10 WB ra direct Strip=128
```

The first command creates a RAID 5 volume from slots 41-48, with write through, no read ahead, direct I/O, and a strip size of 128 kb. The second command creates a RAID 10 volume from slots 1-20 with 10 physical disks per array, write back, read ahead, direct I/O, and a strip size of 128 kb.

3. Create a script named `raid-server2.sh` on the admin node. It will be run on the second server of each of the four chassis.

```
vi /root/raid-server2.sh
```

6. Paste the following contents into the file and save it.

```
/opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid5 drives=$1:49-56 WT nora direct Strip=128
```

```
/opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid10 drives=$1:21-40
pdperarray=10 WB ra direct Strip=128
```

```
/opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid5 drives=$1:49-56 WT nora direct Strip=128
/opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid10 drives=$1:21-40 pdperarray=10 WB ra direct Strip=128
```



Note: Do not execute these scripts on the admin or search head nodes. This script is meant only for the indexers.



Note: These scripts needs to be executed on each of the indexer nodes manually. This is because the script takes the EnclosureID as input, which would generally be different on different indexer servers.

4. Change the mode to include execution privileges.

```
chmod +x /root/raid-server1.sh
```

```
chmod +x /root/raid-server2.sh
```

5. Copy the scripts over to all the indexers.

```
clush --group=indexers -B -c /root/raid-server1.sh
```

```
clush --group=indexers -B -c /root/raid-server2.sh
```

6. The script above requires enclosure ID as a parameter. Run the following command to get EnclosureID on each indexer by launching an SSH session onto that indexer.

```
/opt/MegaRAID/storcli/storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'
```

7. Run the appropriate script on each indexer to create a RAID10 volume and a RAID 5 volume, replacing <Enclosure ID> in the command below with the enclosure ID obtained from Step 5. Run the raidserver1.sh script only on the first server of each chassis (a total of four times). Run the raidserver2.sh script only on the second server of each chassis (a total of four times).

```
./raidserver1.sh <EnclosureID>
```

or

```
./raidserver2.sh <EnclosureID>
```



Note: The command above will not override any existing configuration. To clear and reconfigure existing configurations, refer to Embedded MegaRAID Software Users Guide available at www.lsi.com

```

[root@idx2 ~]# rpm -ivh storcli-1.03.11-1.noarch.rpm
Preparing...                               [100%]
Updating / installing...
 1:storcli-1.03.11-1                         [100%]
[root@idx2 ~]# chmod +x /root/raid-server1.sh
[root@idx2 ~]# /opt/MegaRAID/storcli/storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'
102
[root@idx2 ~]# ./raid-server1.sh 102

```



Note: The above figure shows the procedure for creating virtual drive on one indexer. This process needs to be performed on all eight indexers individually.

8. Verify the virtual drives created by using the following command.

```
lsblk
```

Configuring the XFS File System

The following script will format and mount the virtual drives that were created in the previous section. It looks at all available volumes on the indexers, but will skip OS/boot related volumes. The RAID 5 (SSD for hot/warm storage) volume will be mounted based on its UUID as /data/disk1. The RAID 10 (HDD for cold storage) volume will be mounted based on its UUID as /data/disk2.

To create partition tables and file systems on the local disks supplied to each of the nodes, run a script as the root user on each indexer node.

1. On the admin node, create a file containing the following script.



Note: The script assumes there are no partitions already existing on the data volumes. If there are partitions, then they have to be deleted first before running this script. This process is documented in the Note at the end of the section.

```
vi /root/driveconf-idx.sh

#!/bin/bash
[[ "-x" == "${1}" ]] && set -x && set -v && shift 1
count=1
for devX in /sys/class/scsi_host/host?/scan
do
echo '- - -' > ${devX}
done
for devD in $(lsblk | grep disk | cut -c1-3)
do
echo /dev/${devD}
devX=/dev/${devD}
if [[ -b ${devX} && `sbin/parted -s ${devX} print quit|bin/grep -c boot` -
ne 0 ]]
then
echo "${devX} bootable - skipping."
continue
else
echo ${devX}
echo "Setting up Drive => ${devX}"
/sbin/mkfs.xfs -f ${devX}
(( $? )) && continue
#Identify UUID
UUID=`blkid ${devX} | cut -d " " -f2 | cut -d "=" -f2 | sed 's/"//g'`

echo "UUID of ${devX} = ${UUID}, mounting ${devX} as UUID on
/data/disk${count}"
/bin/mkdir -p /data/disk${count}
(( $? )) && continue
/bin/mount -t xfs -o allocsize=128m,inode64,noatime,nobarrier,nodiratime -U
${UUID} /data/disk${count}
(( $? )) && continue
```

```

echo "UUID=${UUID} /data/disk${count} xfs
allocsize=128m,inode64,noatime,nobarrier,nodiratime 0 0" >> /etc/fstab
((count++))
fi
done

```

2. Run the following commands to change the permissions of `driveconf-idx.sh` and then copy it to all the indexers.

```

chmod 755 /root/driveconf-idx.sh

clush --group=indexers -B -c /root/driveconf-idx.sh

```

3. Run the following command from the admin node to run the script across all data nodes.

```

clush --group=indexers -B /root/driveconf-idx.sh

```

4. Run the following from the admin node to list the partitions and mount points to ensure that the drive `/data/disk1` is mounted properly.

```

clush --group=indexers df -h | grep disk1

clush --group=indexers df -h | grep disk2

clush --group=indexers mount | grep disk1

clush --group=indexers mount | grep disk2

clush --group=indexers grep disk1 /etc/fstab

clush --group=indexers grep disk2 /etc/fstab

```

```
[root@admin1 ~]# clush --group=indexers mount | grep disk1
idx2: /dev/sda on /data/disk1 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx4: /dev/sdc on /data/disk1 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx7: /dev/sda on /data/disk1 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx5: /dev/sda on /data/disk1 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx6: /dev/sda on /data/disk1 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx3: /dev/sdc on /data/disk1 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx1: /dev/sdc on /data/disk1 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx8: /dev/sda on /data/disk1 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
[root@admin1 ~]# clush --group=indexers mount | grep disk2
idx2: /dev/sdb on /data/disk2 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx4: /dev/sdd on /data/disk2 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx8: /dev/sdd on /data/disk2 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx7: /dev/sdd on /data/disk2 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx5: /dev/sdd on /data/disk2 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx6: /dev/sdd on /data/disk2 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx1: /dev/sdd on /data/disk2 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
idx3: /dev/sdd on /data/disk2 type xfs (rw, noatime, nodiratime, attr2, nobarrier, inode64, allocsize=131072k, noquota)
[root@admin1 ~]# clush --group=indexers grep disk1 /etc/fstab
idx1: UUID=763cfca9-7587-4f63-b56d-003760639800 /data/disk1 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx3: UUID=5161cf7b-6a38-44f1-8f26-ffbc9471e798 /data/disk1 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx7: UUID=56529a3f-c844-4443-9f06-cdad82a51dfd /data/disk1 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx4: UUID=0c9f0bed-9e40-46af-8a2b-05be20127191 /data/disk1 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx2: UUID=2fb6f9ae-f014-4b05-9bbe-f4d959d47f53 /data/disk1 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx5: UUID=f0adec05-7a66-4426-971a-aedd6180dc3c /data/disk1 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx6: UUID=e317740a-8f8d-44a5-8630-947298bfbed3 /data/disk1 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx8: UUID=d64be4ab-abc6-4789-84c1-626df9253ab6 /data/disk1 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
[root@admin1 ~]# clush --group=indexers grep disk2 /etc/fstab
idx3: UUID=3bcf7d8fb-46e8-40b5-ad0b-ca30378f00c5 /data/disk2 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx1: UUID=439f4248-417a-49f4-9818-d1d782c041b6 /data/disk2 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx2: UUID=fa0a4cbc-86c4-4bad-b078-ae85ca9960e7 /data/disk2 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx6: UUID=999fa721-148c-48f5-ae61-2914087359f9 /data/disk2 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx4: UUID=7bf7d746-03c8-43b2-bc50-b4119775f2e6 /data/disk2 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx5: UUID=acf692d8-cbec-4a1b-b51c-81b1cbd4b546 /data/disk2 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx7: UUID=c286404c-c04e-4fa9-957a-05f0d465e796 /data/disk2 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
idx8: UUID=b887d153-b774-4f32-80c5-d347df54d313 /data/disk2 xfs allocsize=128m, inode64, noatime, nobarrier, nodiratime 0 0
```



Note: If there is a need to delete any partitions; it can be done using the following:

Run command 'mount' to identify which drive is mounted to which device: /dev/sd<?>.

Unmount the drive for the partition to be deleted (disk1 is shown below as an example) and run fdisk to delete it as shown below

```
mount
```

```
umount /data/disk1
```

```
(echo d; echo w;) | sudo fdisk /dev/sd<?>
```



Be careful not to delete the OS partition as this will wipe out the installed OS.

Cluster Verification

The section describes the steps to create the script `cluster_verification.sh` that helps to verify CPU, memory, NIC, storage adapter settings across the entire cluster. This script also checks additional prerequisites such as NTP status, SELinux status, ulimit settings, IP address and hostname resolution, Linux version, and firewall settings.

1. Create the `cluster_verification.sh` script as follows on the admin node (admin1):

```
vi cluster_verification.sh
```

```
#!/bin/bash
```

```
shopt -s expand_aliases
```



```

# Setting Color codes

green='\e[0;32m'

red='\e[0;31m'

NC='\e[0m' # No Color

echo -e "${green} === Cisco UCS Integrated Infrastructure for Big Data \
Cluster

Verification === ${NC}"

echo ""

echo ""

echo -e "${green} ===== System Information ===== ${NC}"

echo ""

echo ""

echo -e "${green}System ${NC}"

clush -a -B " `which dmidecode` |grep -A2 '^System Information'"

echo ""

echo ""

echo -e "${green}BIOS ${NC}"

clush -a -B " `which dmidecode` | grep -A3 '^BIOS I'"

echo ""

echo ""

echo -e "${green}Memory ${NC}"

clush -a -B "cat /proc/meminfo | grep -i ^memt | uniq"

echo ""

echo ""

echo -e "${green}Number of Dimms ${NC}"

clush -a -B "echo -n 'DIMM slots: '; `which dmidecode` |grep -c \
'^[[[:space:]]*Locator:'"

clush -a -B "echo -n 'DIMM count is: '; `which dmidecode` | grep "Size"| grep
-c "MB""

```

```

clush -a -B "`which dmidecode` | awk '/Memory Device$/ ,/^$/ {print}' | grep -
e \

'^Mem' -e Size: -e Speed: -e Part | sort -u | grep -v -e 'NO DIMM' -e 'No
Module Installed' -e Unknown"

echo ""

echo ""

# probe for cpu info #

echo -e "${green}CPU ${NC}"

clush -a -B "grep '^model name' /proc/cpuinfo | sort -u"

echo ""

clush -a -B "`which lscpu` | grep -v -e op-mode -e ^Vendor -e family -e \
Model: -e Stepping: -e Bogomips -e Virtual -e ^Byte -e '^NUMA node(s)'"

echo ""

echo ""

# probe for nic info #

echo -e "${green}NIC ${NC}"

clush -a -B "`which ifconfig` | egrep ' (^e|^p)' | awk '{print \$1}' | xargs -l
\

`which ethtool` | grep -e ^Settings -e Speed"

echo ""

clush -a -B "`which lspci` | grep -i ether"

echo ""

echo ""

# probe for disk info #

echo -e "${green}Storage ${NC}"

clush -a -B "echo 'Storage Controller: '; `which lspci` | grep -i -e \
raid -e storage -e lsi"

echo ""

clush -a -B "dmesg | grep -i raid | grep -i scsi"

echo ""

```

```

clush -a -B "lsblk -id | awk '{print \$1,\$4}'|sort | nl"

echo ""

echo ""

echo -e "${green} ===== Software ===== ${NC}"

echo ""

echo ""

echo -e "${green}Linux Release ${NC}"

clush -a -B "cat /etc/*release | uniq"

echo ""

echo ""

echo -e "${green}Linux Version ${NC}"

clush -a -B "uname -srvm | fmt"

echo ""

echo ""

echo -e "${green}Date ${NC}"

clush -a -B date

echo ""

echo ""

echo -e "${green}NTP Status ${NC}"

clush -a -B "ntpstat 2>&1 | head -1"

echo ""

echo ""

echo -e "${green}SELINUX ${NC}"

clush -a -B "echo -n 'SElinux status: '; grep ^SELINUX= /etc/selinux/config
2>&1"

echo ""

echo ""

echo -e "${green}IPTables ${NC}"

clush -a -B "`which chkconfig` --list iptables 2>&1"

```

```

echo ""

clush -a -B " `which service` iptables status 2>&1 | head -10"

echo ""

echo ""

echo -e "${green}Transparent Huge Pages ${NC}"

clush -a -B " cat /sys/kernel/mm/*transparent_hugepage/enabled"

echo ""

echo ""

echo -e "${green}CPU Speed${NC}"

clush -a -B "echo -n 'CPUSpeed Service: '; `which service` cpuspeed status
2>&1"

clush -a -B "echo -n 'CPUSpeed Service: '; `which chkconfig` --list cpuspeed
2>&1"

echo ""

echo ""

echo -e "${green}Hostname Lookup${NC}"

clush -a -B " ip addr show"

echo ""

echo ""

echo -e "${green}Open File Limit${NC}"

clush -a -B 'echo -n "Open file limit(should be >32K): "; ulimit -n'

```

2. Change permissions to executable

```
chmod 755 cluster_verification.sh
```

3. Run the Cluster Verification tool from the admin node. This can be run before starting Splunk Enterprise software to identify any discrepancies in post-OS configuration among the servers.

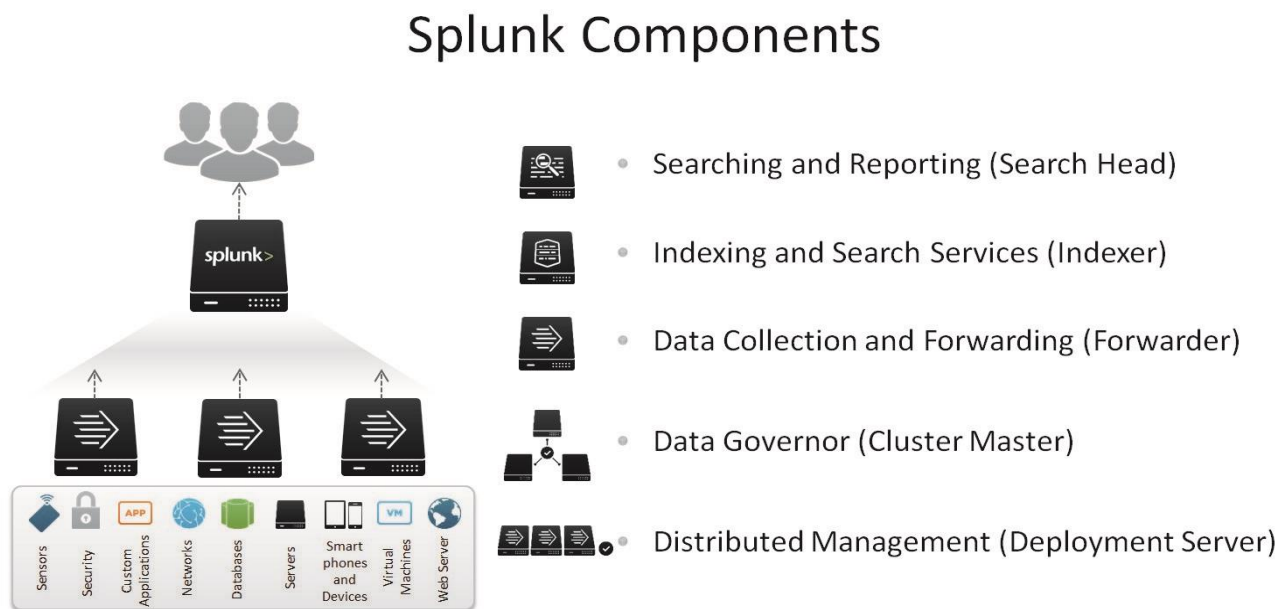
```
./cluster_verification.sh
```

Installing Splunk Enterprise 6.5

Splunk Architecture and Terminology

Splunk comes packaged as an 'all-in-one' distribution. The single file can be configured to function as one or all of the following components (Splunk's Universal Forwarder is a separate package). In a distributed deployment, installations follow a 3-tier approach, as shown in Figure 108

Figure 108 Splunk Components



- Search Head:** A Splunk Enterprise instance that handles search management functions in a distributed search environment, 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 searching (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 and placing the results into an index. It also 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:** A small-footprint version of a forwarder, a Splunk Enterprise instance that forwards data to another Splunk server or a third-party system without parsing.
- Heavy Forwarder:** A fully functional Splunk instance that is configured to send data to the indexing tier. The heavy forwarder performs Splunk's parsing phase before forwarding the data.

- **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.
- **Deployer (not pictured):** A Splunk Enterprise instance that distributes apps and certain other configuration updates to search head cluster members
- **License Master (not pictured):** A license master controls one or more license slaves. From the license master, you can define stacks and pools, add licensing capacity, and manage license slaves.
- **Distributed Management Console (not pictured):** The distributed management console lets you view detailed performance information about your Splunk Enterprise deployment. The topics in this chapter describe the available dashboards and alerts.

In this distributed configuration, indexers and search heads are configured in a clustered mode. Splunk Enterprise supports clustering for both search heads and indexers.

- A 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 more information, please refer to [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.

Planning the Installation

In this CVD, three (3) clustered search heads, eight (8) clustered indexers, a deployment server, a deployer, a distributed management console, a master node, and a license master are configured.

Installation order will be as follows:

- Splunk Installation
- Configure License Master
- Configure Master Node
- Configure Indexing Cluster
- Configure Deployer
- Configure Search Head Cluster
- Configure Distribution Management Console
- Configure Deployment Server
- Install universal forwarder
- Verify Installation
- Post Install Clean up

It is highly recommended that assigned hostnames match their corresponding function, for example a search head may be 'splksrch1.domain.com' or an indexer may be idx1.domain.com. Throughout this document, instructions are provided and examples include the use of hostnames. Your deployment may or may not use the same hostnames. Use Table 7 to plan and track assigned roles and hostnames/IP addresses:

Table 7 Assigned Roles and IP Addresses

CVD Hostname	Function / Model	Hostname	IP
sh1	Search Head 1 C220 M4		
sh2	Search Head 2 C220 M4		
sh3	Search Head 3 C220 M4		
idx1	Indexer 1 S3260		
idx2	Indexer 2 S3260		

CVD Hostname	Function / Model	Hostname	IP
idx3	Indexer 3 S3260		
idx4	Indexer 4 S3260		
idx5	Indexer 4 S3260		
idx6	Indexer 4 S3260		
idx7	Indexer 4 S3260		
idx8	Indexer 4 S3260		
admin1	Admin Box 1 (Master Node, License Master, Distributed Management Console, Deployer) C220 M4		
admin2	Admin Box 2 Deployment Server C220 M4		



Note: The IP addresses and hostnames used in this CVD can be found in Table 7.

Installing Splunk

The Splunk Enterprise software is a single software package that can be configured to function in a specific role. Installation of Splunk across all nodes will be the same, with no specific parameters required; configuration changes will be required for each respective component. As such, a simple installation across every server will be the base to build this architecture.

1. From a host connected to internet, download Splunk Enterprise software from the splunk.com website. Copy it over to the server admin1.

```
[root@admin1 ~]# ls splunk*
splunk-6.5.0-59c8927def0f-linux-2.6-x86_64.rpm
```

2. Copy Splunk software over to all the nodes (2 admins, 3 search heads, and 8 indexers).

```
clush -a -c ./splunk-6.5.0-59c8927def0f-linux-2.6-x86_64.rpm --dest=/tmp
```

3. Modify the permissions on the Splunk Enterprise RPM file to include execution privileges.

```
clush -a chmod +x /tmp/splunk-6.5.0-59c8927def0f-linux-2.6-x86_64.rpm
```


4. Create a directory tree “/data/disk1” on the search heads and admin nodes.

```
clush --group=admins,searchheads mkdir -p /data/disk1
```



Note: The indexers already have a similar directory that is /data/disk1 which serves as the mount point for the RAID5 volume we created in the earlier sections. This step will make the directory structure uniform across all nodes where Splunk Enterprise is installed.

5. Install Splunk Enterprise in the directory /data/disk1 of the indexers, search heads and admin nodes.

```
clush -a -B rpm -ivh --prefix=/data/disk1 /tmp/splunk-6.5.0-59c8927def0f-  
linux-2.6-x86_64.rpm
```

```
[root@admin1 ~]# clush -a -B rpm -ivh --prefix=/data/disk1 /tmp/splunk-6.5.0-59c8927def0f-linux-2.6-x86_64.rpm
-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
warning: /tmp/splunk-6.5.0-59c8927def0f-linux-2.6-x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 653fb112: NOKEY
Preparing...
Updating / installing...
splunk-6.5.0-59c8927def0f
complete
```

This step installs Splunk Enterprise and creates a user named splunk.



Note: When Splunk Enterprise is installed by means of the RPM package as mentioned above, the installation tool automatically creates a user named splunk and group named splunk.

6. Setup the environment variable:

```
clush -a "echo SPLUNK_HOME=/data/disk1/splunk >> /etc/environment"
```

```
[root@admin1 ~]# clush -a "echo SPLUNK_HOME=/data/disk1/splunk >> /etc/environment"
```

7. Log off and log back in to the server admin1.
8. Use the ClusterShell utility command to verify if the environment variable has been setup correctly.

```
clush -a -B echo $SPLUNK_HOME
```

```
[root@admin1 ~]# clush -a -B echo $SPLUNK_HOME
-----
admin[1-2],idx[1-8],sh[1-3] (13)
-----
/data/disk1/splunk
```

9. Verify the ownership of the SPLUNK_HOME directory and its contents. All of these files should belong to splunk user and splunk group.

```
clush -a -B ls -l $$SPLUNK_HOME
clush -a -B ls -l $$SPLUNK_HOME/bin/splunk
```

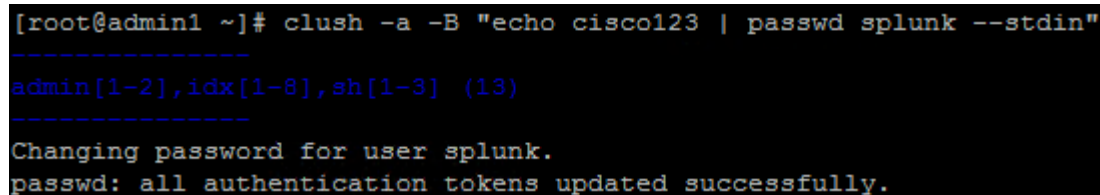
Setting Up Login for Splunk User

As mentioned above, the ‘splunk’ user is created without a password. This section describes the procedure to assign a password and configure the password-less login for that user account.

This facilitates the usage of ClusterShell commands.

1. From admin1, assign the password for the user `splunk` on all the Splunk indexers, search heads, and admin servers.

```
clush -a -B "echo cisco123 | passwd splunk --stdin"
```



```
[root@admin1 ~]# clush -a -B "echo cisco123 | passwd splunk --stdin"
admin[1-2],idx[1-8],sh[1-3] (13)
Changing password for user splunk.
passwd: all authentication tokens updated successfully.
```



Note: In this example, we are using a command line method with clear-text password for the sake of simplification. It is recommended to set up a strong password and set the password manually on each server individually to match the target datacenter’s security practices.

2. Log onto the admin node as user `splunk` using the password selected in the above step (`cisco123` is used in this CVD).
3. Run the `ssh-keygen` command to create both public and private keys on the admin node for the user `splunk`.

```

login as: splunk
splunk@50.1.1.101's password:
Last login: Thu Oct 27 16:16:32 2016
[splunk@admin1 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/data/disk1/splunk/.ssh/id_rsa):
Created directory '/data/disk1/splunk/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /data/disk1/splunk/.ssh/id_rsa.
Your public key has been saved in /data/disk1/splunk/.ssh/id_rsa.pub.
The key fingerprint is:
b1:d5:fa:47:43:a1:fa:a6:a6:47:d2:b6:f8:03:40:ab splunk@admin1
The key's randomart image is:
+--[ RSA 2048 ]-----+
|           .           |
|      . . . . .       |
|     . . . o .        |
|    o + o .           |
|   . S.o o           |
|  E  o +o . .         |
|       * .+ .         |
|      . =o .           |
|     . =o .           |
+-----+
[splunk@admin1 ~]$ █

```

- Run the following script from the admin node to copy the public key `id_rsa.pub` to all the Splunk servers, that is, indexers, search heads and admins of the cluster. `ssh-copy-id` appends the keys to the remote-host's `.ssh/authorized_key`.

```

for host in admin1 admin2 idx1 idx2 idx3 idx4 idx5 idx6 idx7 idx8 sh1 sh2 sh3;
do echo -n "$host -> "; ssh-copy-id -i ~/.ssh/id_rsa.pub $host; done

```

- Enter `yes` for, "Are you sure you want to continue connecting (yes/no)?" Enter the password of the remote host.
- Verify the password-less login by entering the following command. The output should display the hostnames of all splunk servers.

```

clush -a hostname

```

```
[splunk@admin1 ~]$ clush -a hostname
idx1: idx1
admin1: admin1
idx4: idx4
admin2: admin2
idx3: idx3
idx6: idx6
idx5: idx5
idx7: idx7
sh2: sh2
idx8: idx8
sh3: sh3
idx2: idx2
sh1: sh1
```

Starting the Splunk Enterprise Cluster

1. Log onto the admin node as user `splunk`.
2. Start the Splunk Enterprise services.

```
clush -a $SPLUNK_HOME/bin/splunk start --accept-license
```

```
[splunk@admin1 ~]$ clush -a $SPLUNK_HOME/bin/splunk start --accept-license
```

3. Verify the status of the Splunk Enterprise services.

```
clush -a $SPLUNK_HOME/bin/splunk status
```

```
[splunk@admin1 ~]$ clush -a $SPLUNK_HOME/bin/splunk status
admin1: splunkd is running (PID: 24234) .
admin1: splunk helpers are running (PIDs: 24239 24247 24320 24410) .
admin2: splunkd is running (PID: 21224) .
admin2: splunk helpers are running (PIDs: 21229 21237 21310 21400) .
idx1: splunkd is running (PID: 21871) .
idx1: splunk helpers are running (PIDs: 21877 21889 21952 22054) .
idx3: splunkd is running (PID: 28500) .
idx3: splunk helpers are running (PIDs: 28505 28513 28576 28676) .
idx4: splunkd is running (PID: 28449) .
idx4: splunk helpers are running (PIDs: 28454 28462 28525 28584) .
sh2: splunkd is running (PID: 21154) .
sh2: splunk helpers are running (PIDs: 21159 21167 21240 21321) .
sh1: splunkd is running (PID: 21154) .
sh1: splunk helpers are running (PIDs: 21159 21167 21240 21321) .
idx5: splunkd is running (PID: 22839) .
idx5: splunk helpers are running (PIDs: 22844 22852 22916 23016) .
idx8: splunkd is running (PID: 22833) .
idx8: splunk helpers are running (PIDs: 22838 22846 22909 23009) .
idx6: splunkd is running (PID: 22848) .
idx6: splunk helpers are running (PIDs: 22853 22861 22927 23027) .
idx7: splunkd is running (PID: 22805) .
idx7: splunk helpers are running (PIDs: 22810 22818 22881 22981) .
idx2: splunkd is running (PID: 5420) .
idx2: splunk helpers are running (PIDs: 5426 5435 5498 5599) .
sh3: splunkd is running (PID: 21145) .
sh3: splunk helpers are running (PIDs: 21150 21158 21231 21321) .
```

Logging in for the First Time

When logging in for the first time, the default password is 'changeme'. The GUI then prompts for the user to change the admin password. This can be completed by logging on to the GUI via every instance:

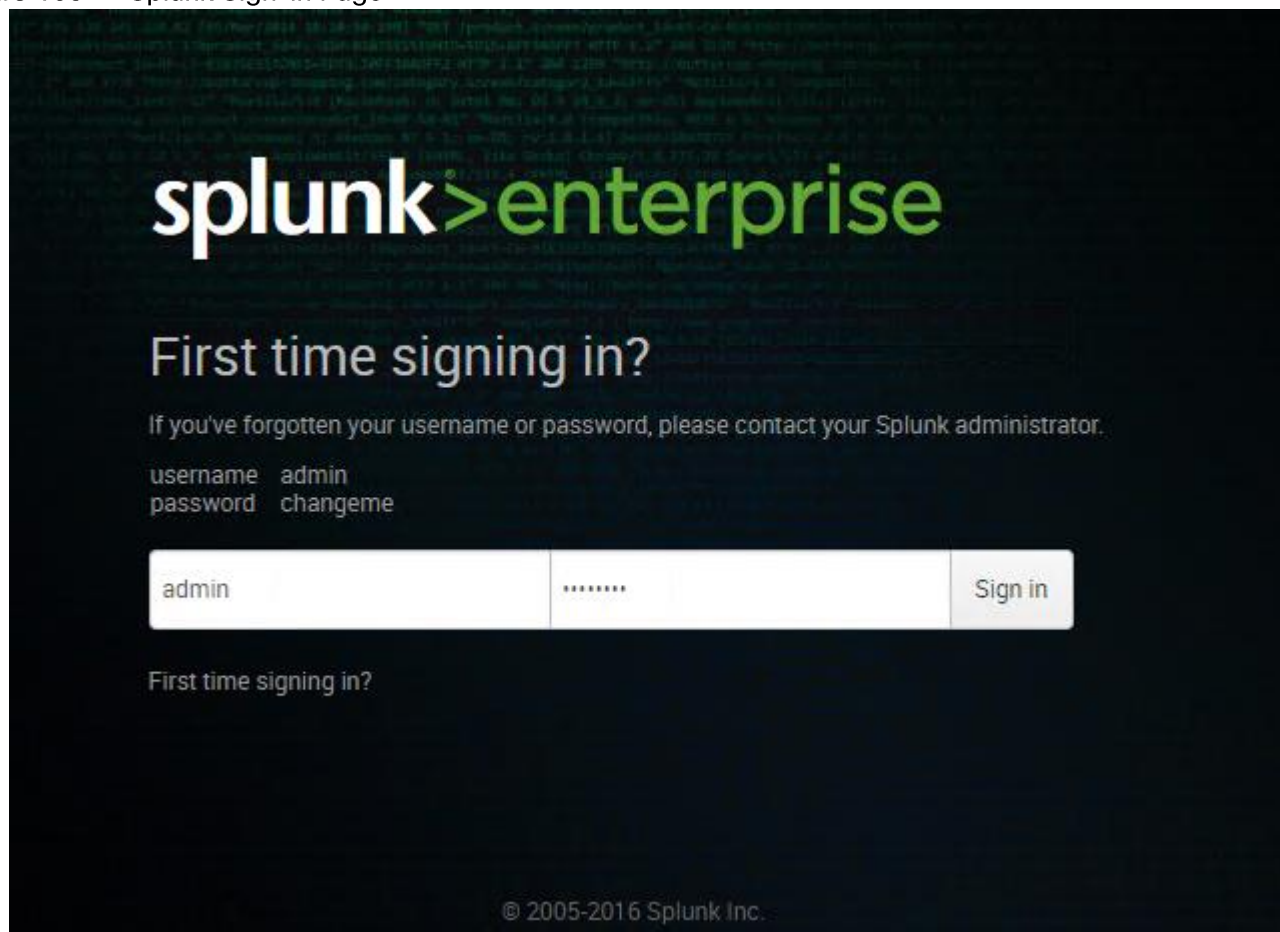
Log into the admin1 instance through the default port of '8000'. For example, use

`http://admin1:8000.`



Note: If you have not added these servers to DNS, you will need to use the IP address, for example, <http://50.1.1.101:8000/>

Figure 109 Splunk Sign-in Page



In this CVD, the password for the Splunk administrator is set to `cisco123` (the same as the OS `splunk` user). You will need to perform this action once on every node via the GUI.

Creating User Accounts

Splunk RPM packages automatically create the user `splunk` with the home directory of the original installation (for example: `/data/disk1/splunk`). If an alternative user is created, repeat the instructions under the previous section, “Setting Up Login for Splunk User”.



Note: The `splunk` user is installed without a password. A password should be assigned to the user `splunk` across all the nodes.

Throughout this CVD, the user `splunk` is used to run all Splunk processes. If there is a requirement to run Splunk as a different user, perform the following:

1. Export `/data/disk1/splunk` as `$SPLUNK_HOME`, add it to the `PATH`
2. The home Directory for new users should be Splunk installation directory (`/data/disk1/splunk/`)

3. Stop all Splunk processes.

```
$SPLUNK_HOME/bin/splunk stop
```

4. As the root user, invoke the `chown` command to change the ownership of the splunk directory and everything under it to the desired user (replacing `<user>`) and usergroup (replacing `<usergroup>`).

```
chown -R <user>:<usergroup> $SPLUNK_HOME/*
```

5. Change or `sudo` to new user.

6. As the desired non-root user, start all splunk processes:

```
$SPLUNK_HOME/bin/splunk start
```

7. When the CVD refers to the user `splunk`, substitute the alternate user.

Initializing Splunk on Boot

To initialize Splunk on boot, complete the following steps:

1. Log onto `admin1` as root user.
2. From the command line, launch the following command:

```
clush -a $SPLUNK_HOME/bin/splunk enable boot-start -user splunk
```

This will initialize Splunk to run as user `splunk` if any server is rebooted. If the desired Splunk user account is not `splunk`, change the `-user` reference accordingly. The output below has been truncated and should show the same message for all servers.

```
[root@admin1 ~]# clush -a $SPLUNK_HOME/bin/splunk enable boot-start -user splunk
admin1: Init script installed at /etc/init.d/splunk.
admin1: Init script is configured to run at boot.
```

Default Ports

The following are the default ports that are used by Splunk software on every node. For more information, please refer to [Splunk Documentation](#).

Table 8 Default Ports used by Splunk

Function	Default Port
Management Port	8089
Web Interface	8000

Configuring the Splunk Enterprise Cluster

Configuring Splunk Enterprise Licenses

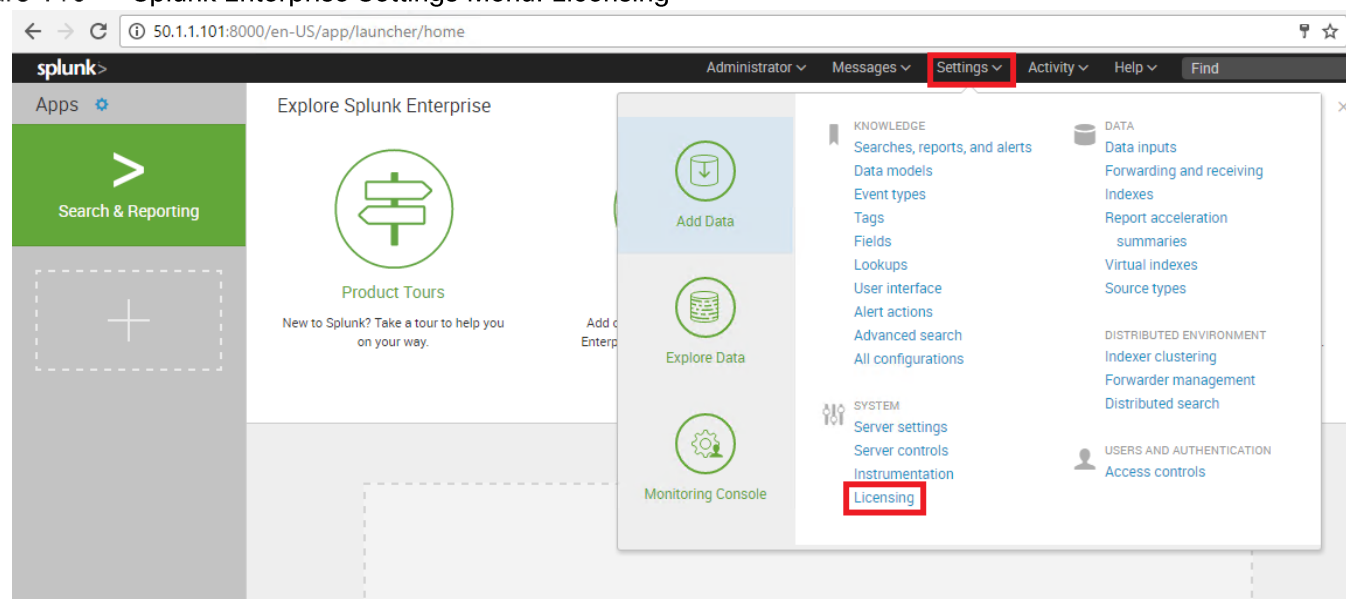
The servers in the Splunk Enterprise infrastructure that perform indexing must be licensed. Any Splunk instance can be configured to perform the role of license master. In this CVD, the admin node (admin1) is configured to be the license master and all the other Splunk instances are configured as license slaves.

Setting Up License Master

Configure the server admin1 as the central license master by following the procedures detailed below.

1. Log onto the server admin1 through the web GUI as user admin.
2. Navigate to the licensing screen by clicking on Settings → Licensing, as shown in Figure 110

Figure 110 Splunk Enterprise Settings Menu: Licensing



3. Click on Change License Group.
4. Click on the Enterprise License radio button, as shown in Figure 111

Figure 111 Change License Group

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 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. Click on `Save`.
6. In the `Add new license` dialog, click on `Choose File` to select your license file.
7. Click `Install` to install the license. See Figure 112

Figure 112 Add New License

splunk> Apps Administrator Messages Settings Activity Help Find

Add new license
[Licensing](#) » Add new 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`):

Choose File splunk.license

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

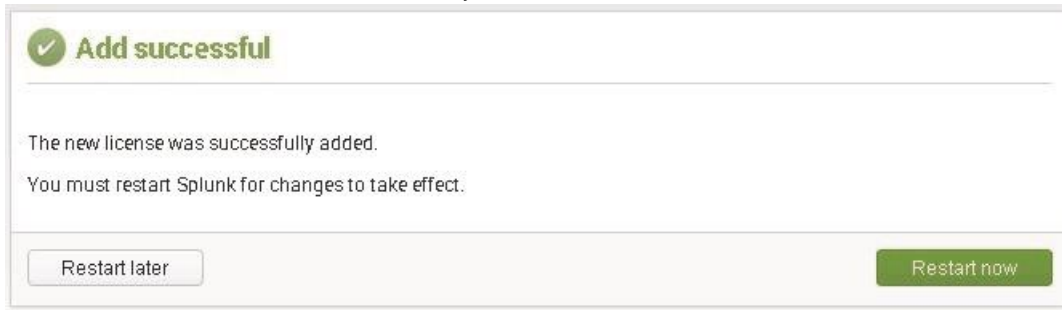
Cancel Install

About Support File a Bug Documentation Privacy Policy © 2005-2015 Splunk Inc. All rights reserved.

8. Click on `Restart now`.

- Click **OK** to restart Splunk and complete the license installation.

Figure 113 Added License Successfully



- Log back in to 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 [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, that is, admin1. This can be performed by following one of the two methods described below.

The first and preferred method is to use the ClusterShell command (`clush`) to configure all the Splunk instances to be license slaves to the license master in admin1. The second (optional) method is to configure each node as a license slave individually by accessing their respective Web UI.

Configure all the License Slaves at Once Using CLI (Clush)

- From the admin node (admin1) as user `splunk` execute the command:

```
clush -a -x admin1 -B $SPLUNK_HOME/bin/splunk edit licenser-localslave -
master_uri https://admin1:8089 -auth admin:cisco123
```

```
[splunk@admin1 ~]$ clush -a -x admin1 -B $SPLUNK_HOME/bin/splunk edit licenser-localslave -master_uri https://admin1:8089 -auth admin:cisco123
-----
admin2,idx[1-8],sh[1-3] (12)
-----
The licenser-localslave object has been edited.
You need to restart the Splunk Server (splunkd) for your changes to take effect.
```

- Restart all nodes except for admin1, by issuing the following command (output below is truncated):

```
clush -a -x admin1 $SPLUNK_HOME/bin/splunk restart
```

```
[splunk@admin1 ~]$ clush -a -x admin1 $SPLUNK_HOME/bin/splunk restart
admin2: Stopping splunkd...
admin2: Shutting down. Please wait, as this may take a few minutes.
idx3: Stopping splunkd...
idx3: Shutting down. Please wait, as this may take a few minutes.
```

- During restart, you will receive confirmation that the instances are running as license-slaves.

```

idx3: Bypassing local license checks since this instance is configured with a remote license master.
idx3:
idx6:
idx6:
idx6: Bypassing local license checks since this instance is configured with a remote license master.
idx6:
idx4:
idx4:
idx4: Bypassing local license checks since this instance is configured with a remote license master.
idx4:

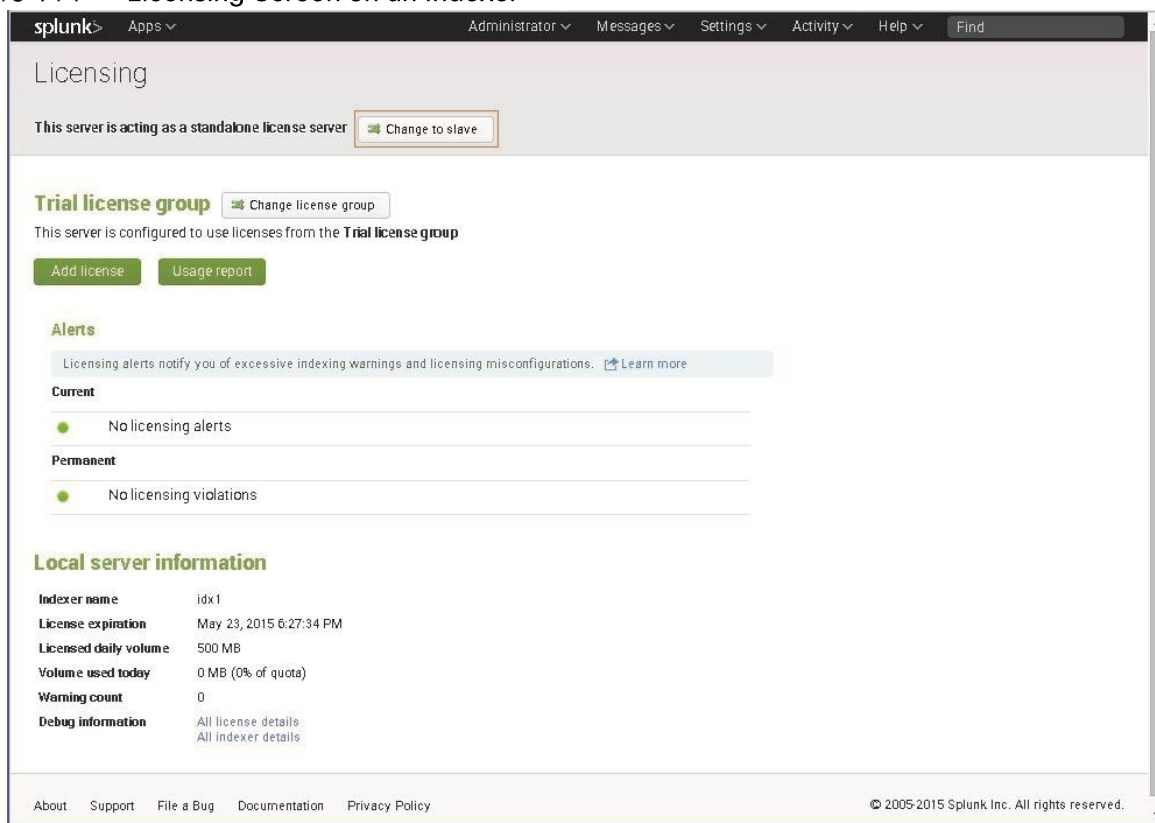
```

4. Proceed to the section ‘Verifying License–Slave Relationships’.

(Optional) Configure License Slaves Individually Using the Web Interface

1. Log on to an indexer server, such as idx1, as user `admin`. (for example, `https://idx1:8000`)
2. Navigate to the licensing screen by clicking on `Settings` → `Licensing`.
3. Click on the button `Change` to `slave`, as shown in Figure 114 .

Figure 114 Licensing Screen on an Indexer



4. In the `Change master association` dialog, click on the `Designate a different Splunk instance as the master license server` radio button, as shown in Figure 115
5. Enter the Master license server URI in the format `https://<IP-or-hostname>:8089`. (for example, `https://admin1:8089`)



Note: The port 8089 is the management port chosen while the server admin1 was provisioned as the master node.

6. Click `Save`.

Figure 115 Configure the Indexer to Choose Admin1 as its License Master

Change master association

This server, `idx1`, is currently acting as a master license server.

Designate this Splunk instance, `idx1`, as the master license server

Choosing this option will:

- Point the local indexer at the local master license server
- Disconnect the local indexer from any remote license server

Designate a different Splunk instance as the master license server

Choosing this option will:

- Deactivate the local master license server
- Point the local indexer at license server specified below
- Discontinue license services to remote indexers currently pointing to this server

Master license server URI

For example: `https://splunk_license_server:8089`
Use https and specify the management port.

Cancel
Save

7. Restart Splunk by clicking on `Restart now`.

Figure 116 Successfully Changed Master Association

✓

Change successful

The master server was successfully changed.

Restart later
Restart now

Repeat the steps above to configure all eight indexers, all three search heads, and the second admin node (admin2), to become license slaves to the license master on the server admin1.

Verifying License-Slave Relationships

To confirm the license configurations, complete the following steps:

1. Go to the master node's Splunk GUI and navigate to `Settings` → `Licensing`.

- Under the section "Local server information", click `All indexer Details` to view the license slaves, as shown in Figure 117

Figure 117 Verifying Indexer Details

Local server information

Indexer name	admin1
Volume used today	0 MB
Warning count	0
Debug information	All license details All indexer details

There should be thirteen license slaves listed: eight indexers, three search heads, and two admin nodes.

Figure 118 Indexer Details

Indexers connected to: admin1 (13)

- idx2**
 - active_pool_names**
 - auto_generated_pool_enterprise
 - added_usage_parsing_warnings** None
 - label** idx2
 - pool_names**
 - auto_generated_pool_download-trial
 - auto_generated_pool_enterprise
 - auto_generated_pool_forwarder
 - auto_generated_pool_free
 - stack_names**
 - download-trial
 - enterprise
 - forwarder
 - free
 - warning_count** 0
- idx8**
 - active_pool_names**
 - auto_generated_pool_enterprise
 - added_usage_parsing_warnings** None
 - label** idx8
 - pool_names**
 - auto_generated_pool_download-trial
 - auto_generated_pool_enterprise
 - auto_generated_pool_forwarder
 - auto_generated_pool_free
 - stack_names**
 - download-trial
 - enterprise
 - forwarder
 - free
 - warning_count** 0
- idx1**
 - active_pool_names**
 - auto_generated_pool_enterprise



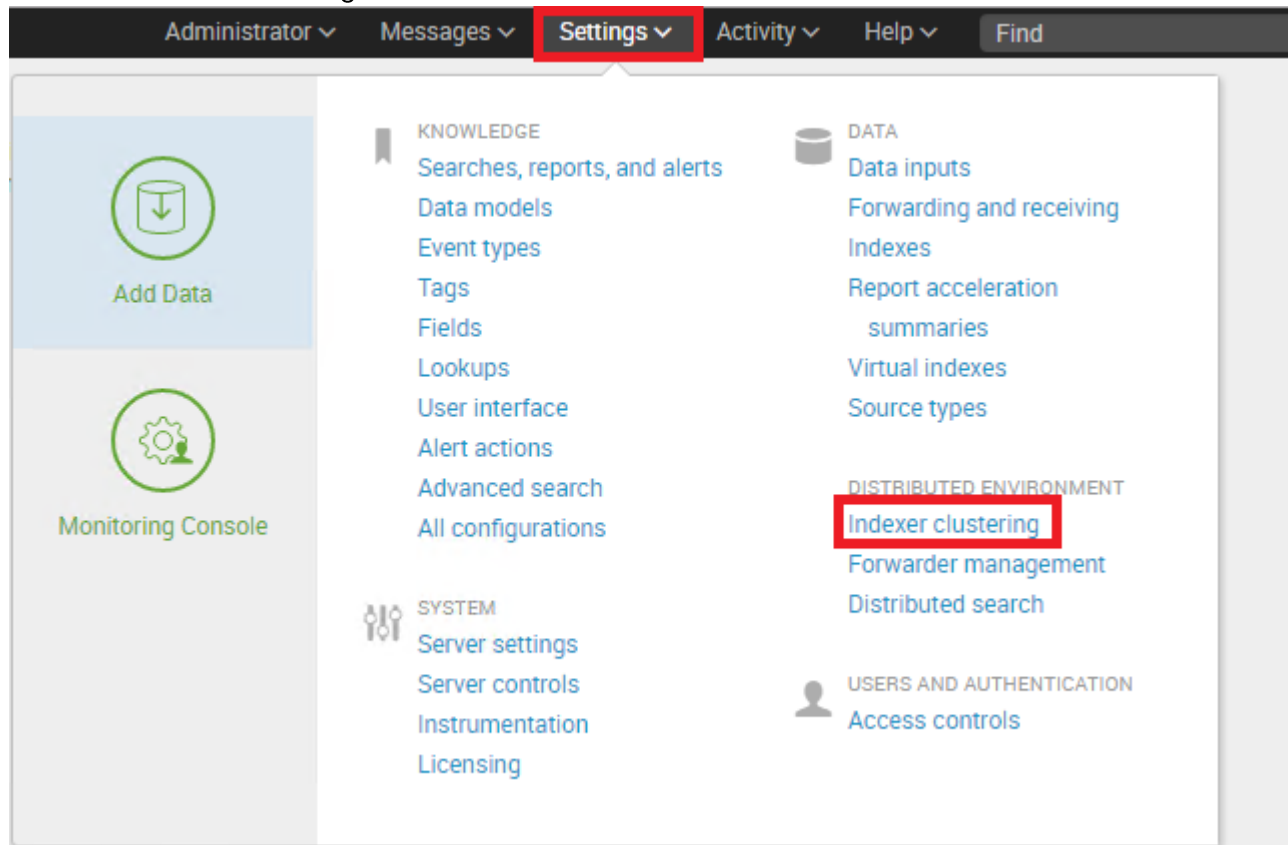
Note: The License Master counts all the license slaves as Splunk indexer instances in spite of the actual roles the instances have been configured to perform.

Configuring the Master Node / Cluster Master

To start, configure admin1 as the indexer Cluster Master.

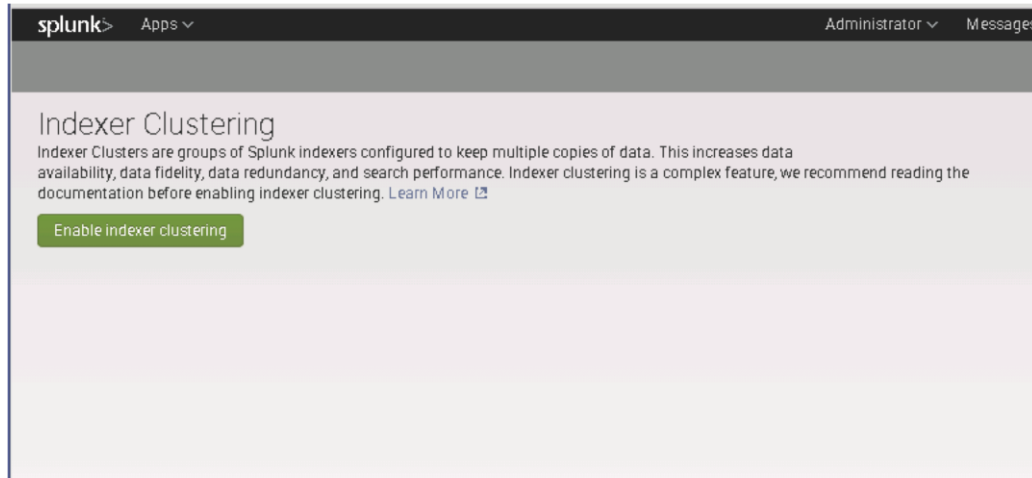
1. Using your browser, go to the master node (admin1) using the format `http://hostname-or-IP:8000/` (for example, `https://admin1:8000/`)
2. Click on the `Settings` → `Indexer Clustering`, as shown in Figure 119

Figure 119 Indexer Clustering



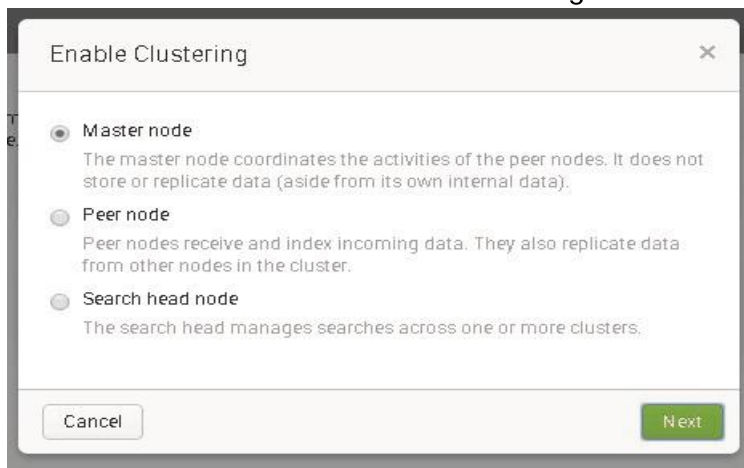
3. Click `Enable Indexer Clustering`, as shown in Figure 120

Figure 120 Enable Indexer Clustering



4. Click the `Master Node` radio button and then click `Next`. See Figure 121

Figure 121 Select the Node to Enable Clustering



5. Set the `Replication Factor` field to 2, and the `Search Factor` field to 2.
6. Set up a `Security Key`; in this installation, `splunk+cisco` is used as the security key.
7. Click `Enable Master Node`.

Figure 122 Master Node Configuration

Master Node Configuration [X]

Replication Factor
 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
 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
 Name your cluster using this field. This label is also used to identify this cluster in the Distributed Management Console.



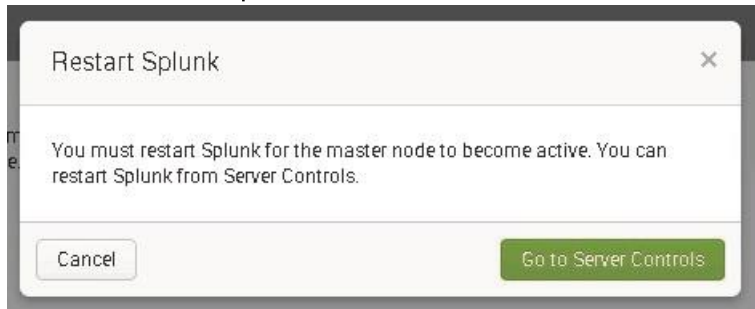
Note: Replication and search factors vary by deployment. The replication factor indicates the number of copies to be maintained on the indexers. The search factor indicates how many of those copies will return search results. In the configuration above, one indexer could be down and searches will still return all results. If the configuration needs to be more resilient, the replication factor may be increased, but this will also increase disk consumption. Consult the documentation for more information. <http://docs.splunk.com/Documentation/Splunk/6.4.1/Indexer/Thereplicationfactor>



Note: Make sure to apply a security key.

8. Click on `Go to Server Controls` to proceed with restarting Splunk as indicated. See Figure 123

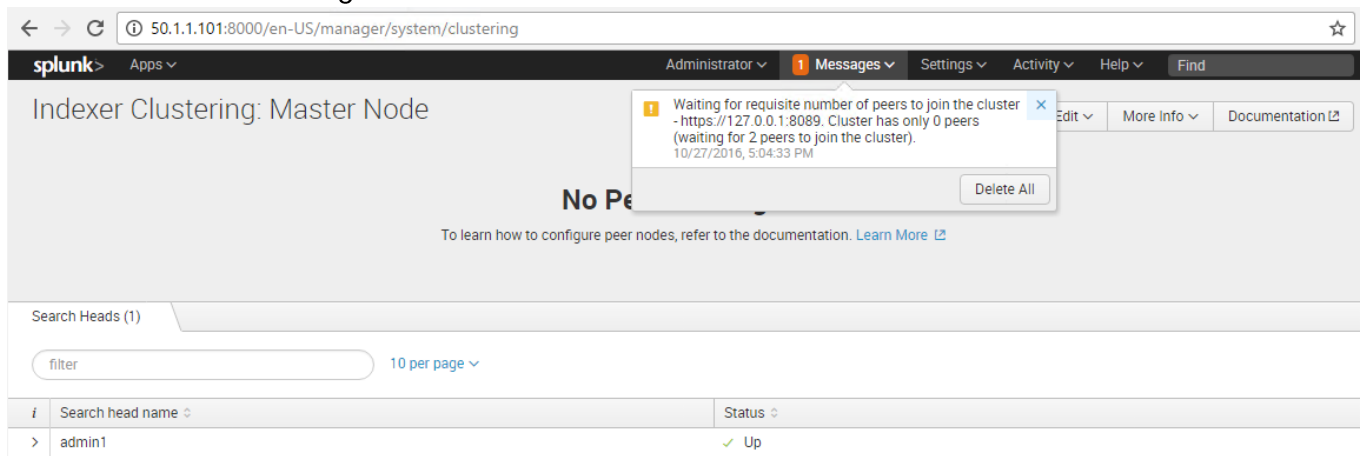
Figure 123 Restart Splunk to Make the Master Node Active



9. The `Restart Successful` message appears. Click `OK` to go back to the login screen.
10. Log back in as the admin user.
11. Return to `Settings` → `Indexer Clustering`.

A message appears indicating that the necessary number of peers must join the cluster. For a replication factor of 2, Splunk Enterprise needs a minimum of 2 peers. Figure 124

Figure 124 Indexer Clustering Master Node



Configure Indexing Peers

Configure all the Splunk instances to be the indexing peers to the master node, `admin1`. This can be performed by following one of the two methods described below.

The first and preferred method is to use ClusterShell command (`clush`) to configure all the Cisco S3260 Storage Servers to be indexing peers to the cluster master in `admin1`. The second (optional) method is to configure each Cisco S3260 Storage Server as an indexing peer individually by accessing their respective Web UI.

Configuring Indexer Clusters

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 single 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 the replication factor. By maintaining multiple, identical copies of Splunk Enterprise data, clusters prevent data loss while promoting data availability for searching.

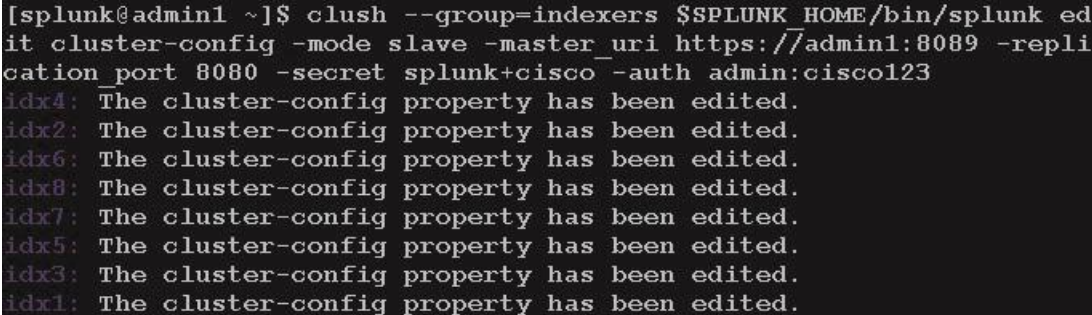
Indexer clusters feature automatic failover from one indexer to the next. This means that, if one or more indexers fail, incoming data continues to get indexed and indexed data continues to be searchable.

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

Configure All Indexing Peers Using CLI (Clush)

1. From the admin1 box, as the splunk user, issue the command:

```
clush --group=indexers $SPLUNK_HOME/bin/splunk edit cluster-config -mode slave
-master_uri https://admin1:8089 -replication_port 8080 -secret splunk+cisco -
auth admin:cisco123
```



```
[splunk@admin1 ~]$ clush --group=indexers $SPLUNK_HOME/bin/splunk ed
it cluster-config -mode slave -master_uri https://admin1:8089 -repli
cation_port 8080 -secret splunk+cisco -auth admin:cisco123
idx4: The cluster-config property has been edited.
idx2: The cluster-config property has been edited.
idx6: The cluster-config property has been edited.
idx8: The cluster-config property has been edited.
idx7: The cluster-config property has been edited.
idx5: The cluster-config property has been edited.
idx3: The cluster-config property has been edited.
idx1: The cluster-config property has been edited.
```

2. After editing the cluster configuration, the affected boxes must be restarted.

```
clush --group=indexers $SPLUNK_HOME/bin/splunk restart
```

3. After all the splunk process in peer nodes are restarted, check the master node's (admin1) web UI. The master node will report the number of available peers, as shown in Figure 125

Figure 125 Available Peers in the Master Node

Indexer Clustering: Master Node

✓ All Data is Searchable
✓ Search Factor is Met
✓ Replication Factor is Met

8 searchable **0** not searchable Peers
 2 searchable **0** not searchable Indexes

Peers (8) | Indexes (2) | Search Heads (1)

filter 10 per page

i	Peer Name	Fully Searchable	Status	Buckets
>	idx8	✓ Yes	Up	10
>	idx1	✓ Yes	Up	10
>	idx7	✓ Yes	Up	8
>	idx5	✓ Yes	Up	8
>	idx2	✓ Yes	Up	14
>	idx6	✓ Yes	Up	12
>	idx3	✓ Yes	Up	8
>	idx4	✓ Yes	Up	10

[About](#)
[Support](#)
[File a Bug](#)
[Documentation](#)
[Privacy Policy](#)
© 2005-2016 Splunk Inc. All rights reserved.

4. Proceed to the section "Setting Dedicated Replication Address".



Note: Once the indexers are added to the cluster, it is not advised to use the command ``$SPLUNK_HOME/bin/splunk restart`` on individual indexers. For further information, see: <http://docs.splunk.com/Documentation/Splunk/latest/Indexer/Restartthecluster>

Configure Indexing Peers Individually Using the Web Interface (Optional)



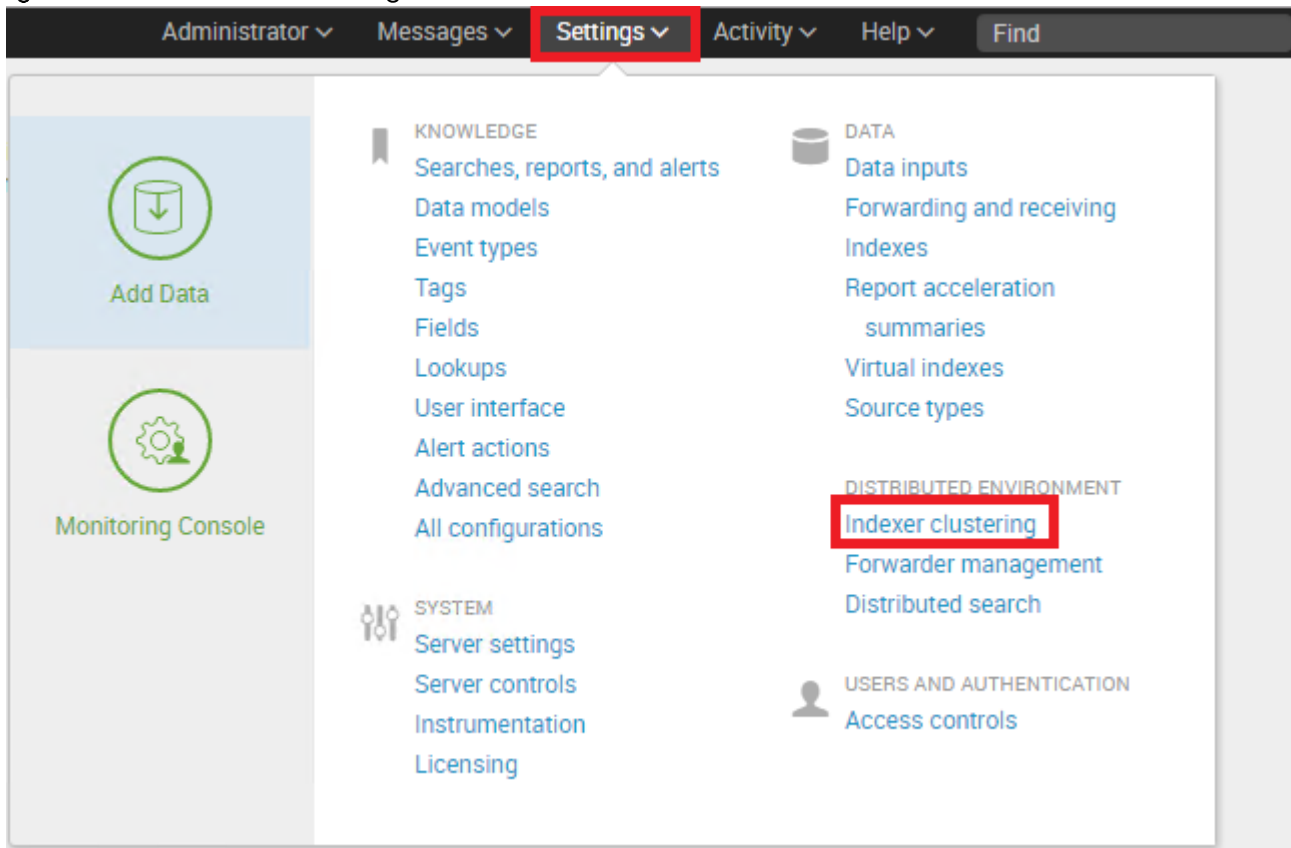
Note: This is an optional method that may be followed to configure each indexer manually through the Splunk Web-UI. The preferred method is to perform the configuration via CLI as shown in the previous section. See the procedure in "Configure All Indexing Peers Using CLI (clush)".

To enable an indexer as a peer node, complete the following steps:

1. Go to an Indexer node's Splunk Web-UI (for example, `http://idx1:8000/`)
2. Login as `admin` user with password `cisco123`.
3. Click `Settings` in the upper right corner of Splunk Web.

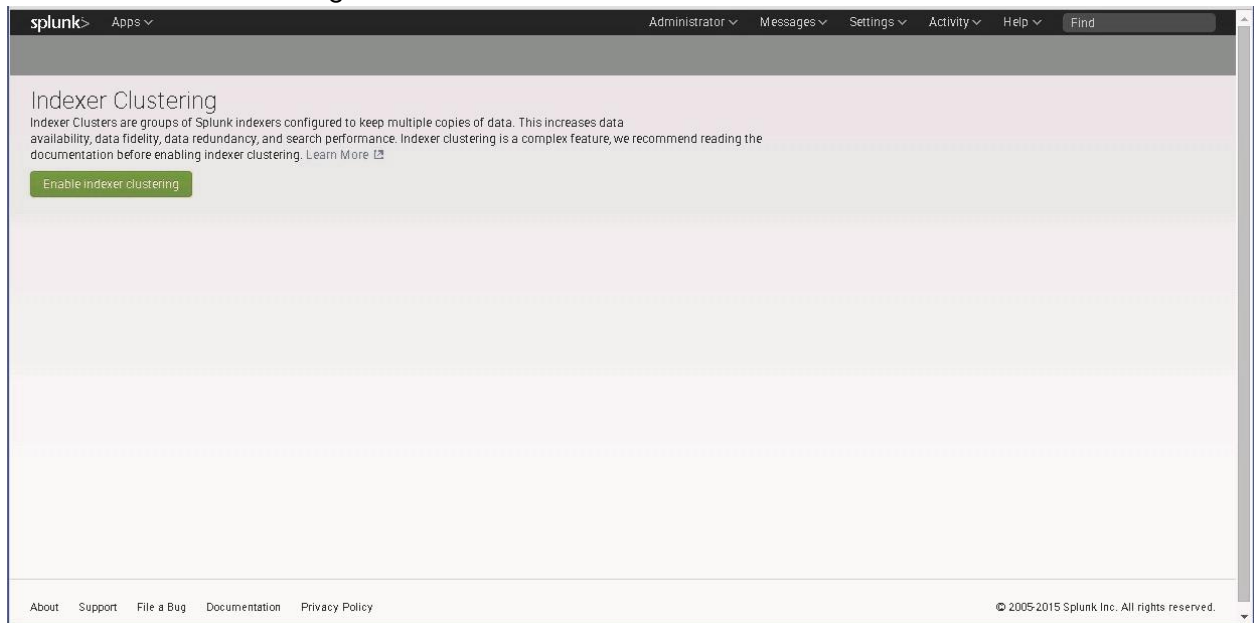
4. In the Distributed environment group, click Indexer Clustering.

Figure 126 Indexer Clustering



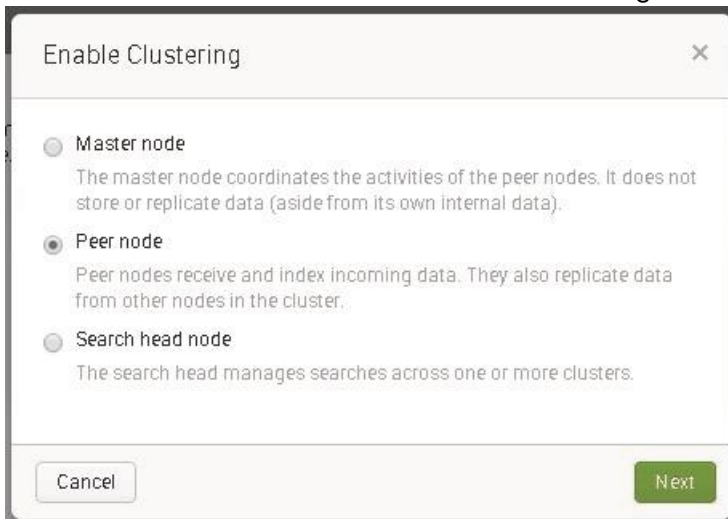
5. Select Enable indexer clustering.

Figure 127 Enable Clustering



6. Select `Peer node` and click `Next`. See Figure 128

Figure 128 Select Desired Node to Enable Clustering



7. Complete entries for the following fields:
 - a. **Master IP address or Hostname:** Enter the master's IP address or hostname. For example, `https://admin1`
 - b. **Master port:** Enter the master's management port. For example, 8089.
 - c. **Peer replication port:** This is the port on which the peer receives replicated data streamed from the other peers. You can specify any available, unused port for this purpose. This port must be different from the management or receiving ports.
 - d. **Security key:** This is the key that authenticates communication between the master and the peers and search heads. The key must be the same across all cluster instances. If the master has a security key, you must enter it here.
8. Click `Enable peer node`. See Figure 129

Figure 129 Enable Peer Node

Peer node configuration

Master IP address or Hostname:
E.g. https://10.152.31.202

Master port:
E.g. 8089

Peer replication port:
The port peer nodes use to stream data to each other (Eg: 8080).

Security key:
This key authenticates communication between the master and the peers and search heads.

Back Enable peer node

9. A message appears, informing that "You must restart Splunk for the peer node to become active."

Figure 130 Restart Splunk for the Peer Node to get Active

Restart Splunk

You must restart Splunk for the peer node to become active.
Optional next steps after restart:

1. Configure the indexes for the peers.
The index file determines the peers set of indexes and the size and attributes of its buckets. This file must be identical across all peer nodes. Peer index files are edited and distributed from the Master Node. [Learn More](#)
2. Use forwarders to get data to this peer.
There are two reasons for using forwarders to send data to your cluster.
1. To ensure that all incoming data gets indexed. 2. To handle potential node failure. [Learn More](#)

You can restart Splunk from Server Controls.

Cancel Go to Server Controls

10. Click **Go to Server Controls**. This will take you to the **Settings** page where you can initiate the restart.



Note: The figures below show the Splunk restart process on indexer `idx1` (that is, 50.1.1.121).

Figure 131 Restart Splunk in Server Control Setting Page

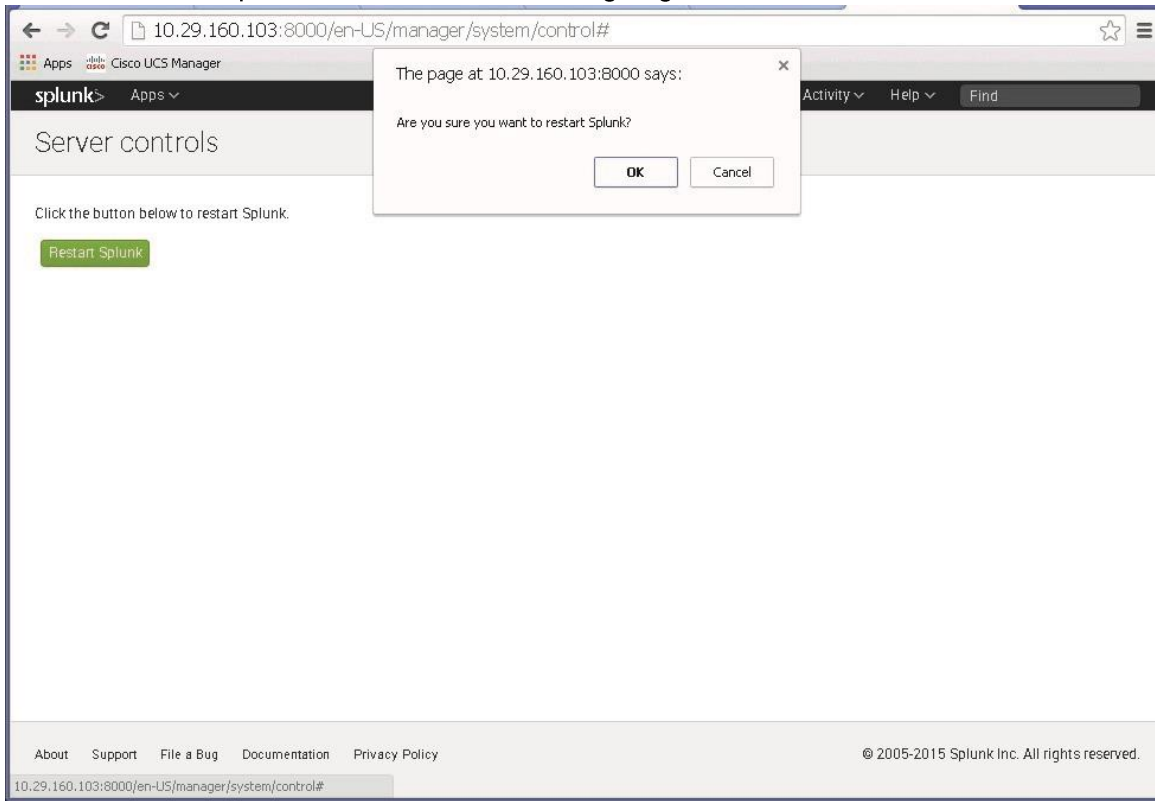
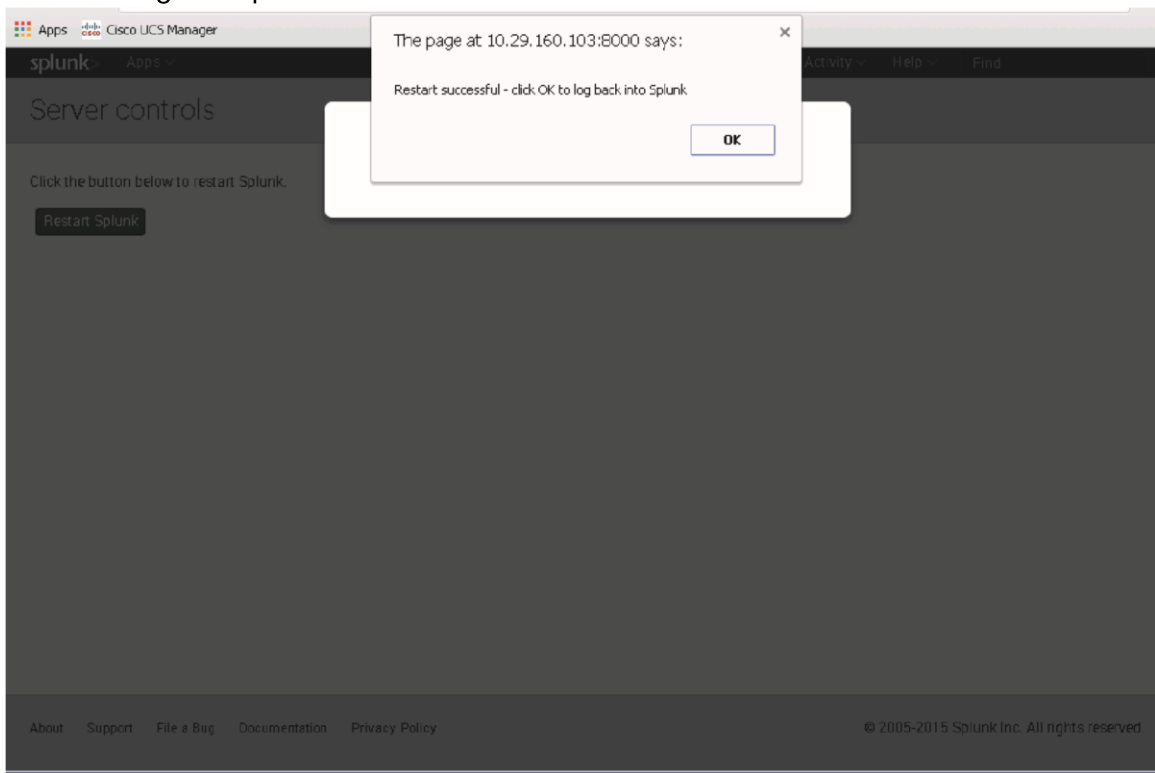
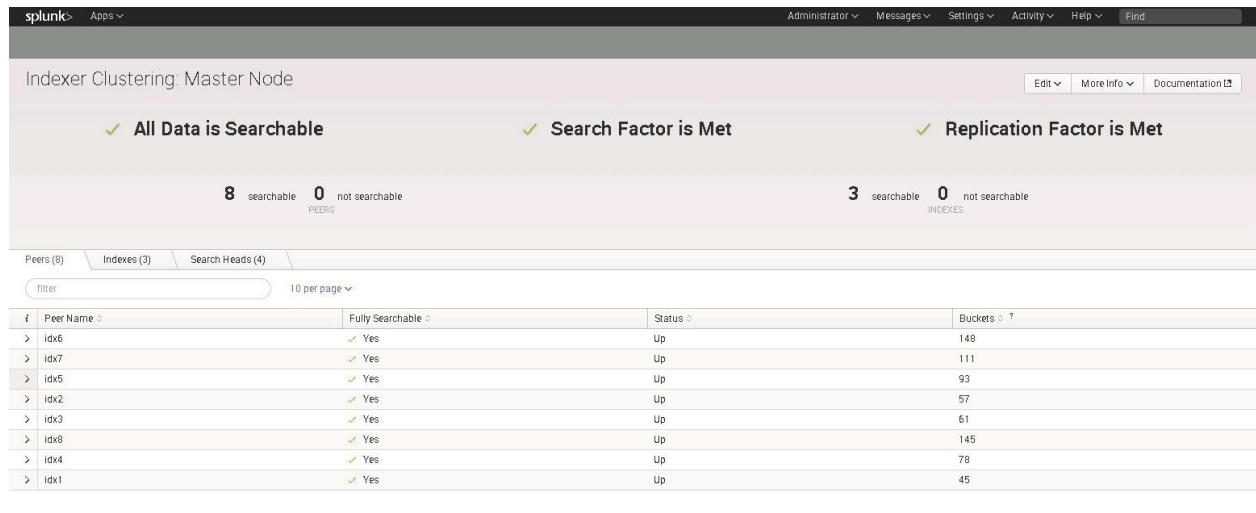


Figure 132 Log into Splunk



11. After the peer node restarts, check the master node's Indexer Clustering page. The Master node must report number of available peers.

Figure 133 Master Node with Available Peers



- Repeat this process for all the cluster's peer nodes (indexers). When complete, the screen should report 8 indexers as reflected in figure above.



Note: Once the indexers are added to the cluster, it is not advised to use the command `$SPLUNK_HOME/bin/splunk restart` on individual indexers. For further information, see: <http://docs.splunk.com/Documentation/Splunk/latest/Indexer/Restartthecluster>.

Setting Dedicated Replication Address

Splunk Enterprise provides a way to make use of a dedicated network interface for index replication of data traffic that happens between the indexers in Splunk Enterprise Indexer cluster. In this CVD, the eth2 interface with an IP address in the range 192.168.12.0/24 is utilized for this purpose. This feature is configured in the `server.conf` file on each Splunk Enterprise indexer instance by setting the `register_replication_address` property. This property can be configured with an IP address or a resolvable hostname.

- SSH to idx1.
- As the splunk user, edit the file `$SPLUNK_HOME/etc/system/local/server.conf`.
- Under the section `[clustering]`, include the line:

```
register_replication_address = idx1-rep
```

```
[clustering]
master_uri = https://admin1:8089
mode = slave
pass4SymmKey = $1$7E/toE/2WNC0ygWKeQ==
register_replication_address = idx1-rep
```




Note: It is important to make sure that the host name, such as `idx1-rep`, or IP address used when setting the `register_replication_address` field is local to the server on which the `server.conf` file resides. The value entered must reflect the replication address of the local server (for example, `idx1-rep`).

4. Save the file.
5. Repeat this across all indexers (`idx1-8`) with the corresponding replication address.
6. SSH to the master node, `admin1`.
7. As user `splunk`, issue the command (If you are prompted for a Splunk username, enter the credentials for the admin user):

```
$SPLUNK_HOME/bin/splunk rolling-restart cluster-peers
```

Verify Cluster Configuration

1. Navigate to the master node's web GUI (for example, <https://admin1:8000>).
2. Select `Settings` → `Index Clustering`.
3. All eight (8) indexers should appear as searchable.

Figure 134 Searchable Indexer Nodes

Indexer Clustering: Master Node

✓ All Data is Searchable ✓ Search Factor is Met

8 searchable 0 not searchable
PEERS

Peers (8) Indexes (2) Search Heads (1)

filter 10 per page

i	Peer Name	Fully Searchable	Status
>	idx2	✓ Yes	Up
>	idx8	✓ Yes	Up
>	idx1	✓ Yes	Up
>	idx4	✓ Yes	Up
>	idx6	✓ Yes	Up
>	idx5	✓ Yes	Up
>	idx7	✓ Yes	Up
>	idx3	✓ Yes	Up

Configure Receiving on the Peer Nodes

In order for the indexers (also known as 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, that is, `admin1`, as follows.

1. On the command line of the master node (`admin1`), navigate to

```
$SPLUNK_HOME/etc/master-apps/_cluster/local
```

2. Create and edit the file `inputs.conf` with the following content:

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

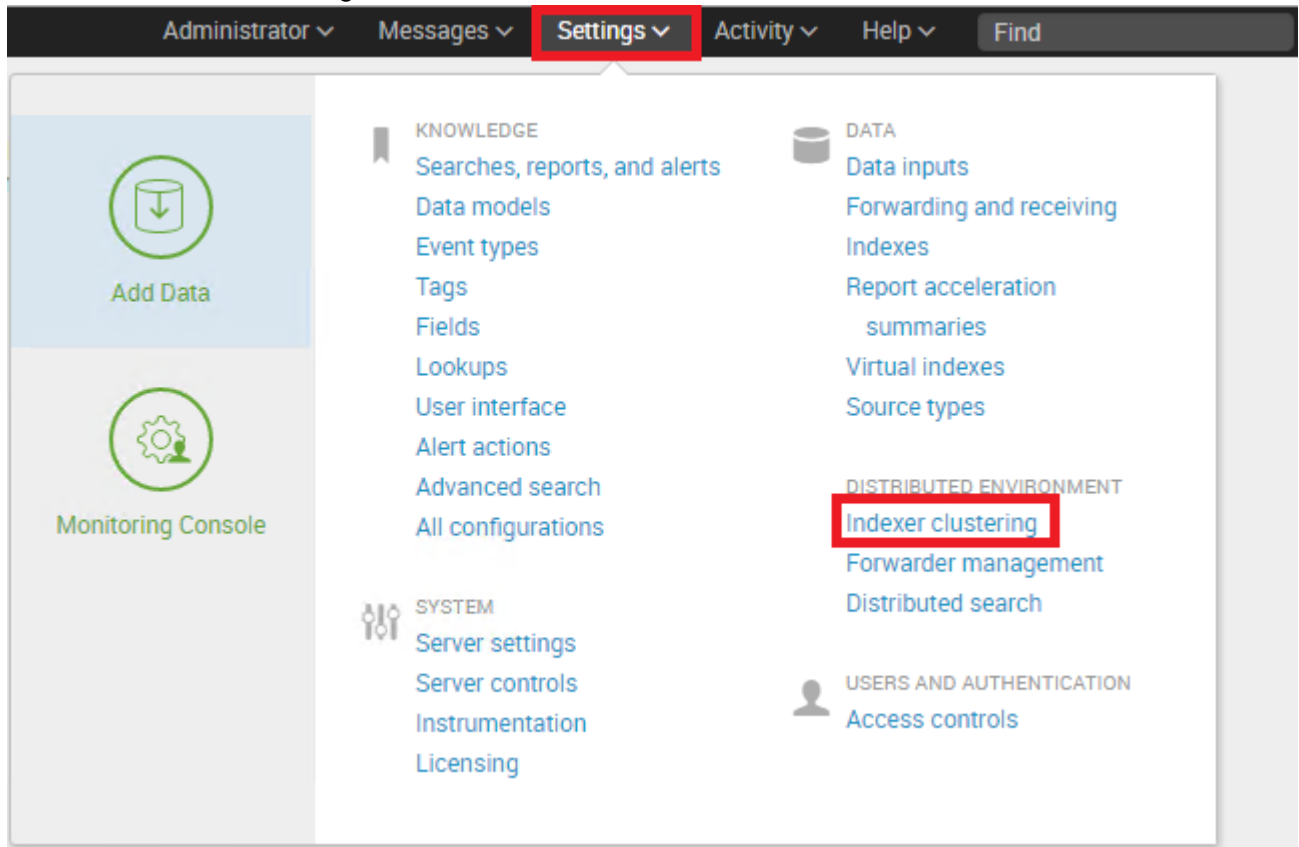
```
[splunk@admin1 ~]$ cd $SPLUNK_HOME/etc/master-apps/_cluster/local
[splunk@admin1 local]$ pwd
/data/disk1/splunk/etc/master-apps/_cluster/local
[splunk@admin1 local]$ vi inputs.conf
[splunk@admin1 local]$ cat inputs.conf
[splunktcp://:9997]
connection_host = ip
```



Note: If this configuration uses DNS, edit 'connection_host = dns'.

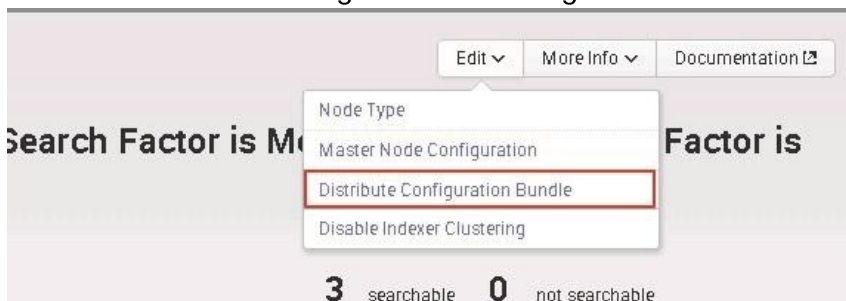
3. Navigate to the `admin1` web interface via browser.
4. Navigate to `Settings` → `Distributed Environment` → `Indexer Clustering`.

Figure 135 Indexer Clustering



5. Select `Edit` → `Distribute Configuration Bundle`.

Figure 136 Indexer Clustering: Distribute Configuration Bundle



6. Select `Distribute Configuration Bundle`.

Figure 137 Distribute Configuration Bundle



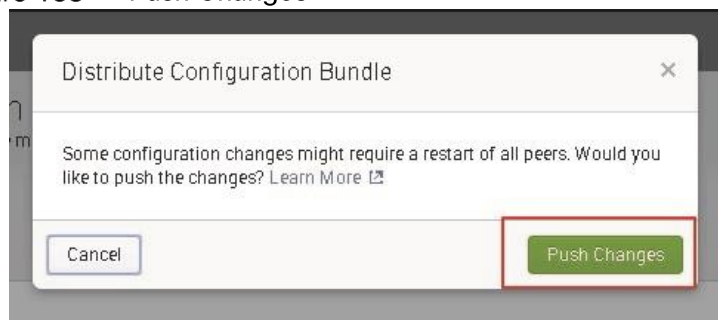
Last Push: ✓ Successful

Time 3/24/2015, 3:31:25 PM

Bundle ID ? 0DE75C59B77A4F1B3296FB0E75B1D750

7. Acknowledge the warning and push changes.

Figure 138 Push Changes



8. Once Push is complete, the GUI should reflect a successful push.

Last Push: ✓ Successful

Time 4/6/2015, 2:38:25 PM

Bundle ID ? 7AD6211B37846DC07D746847C08597CE

Configure Master to Forward All its 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:

- 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.

- 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`, perform the following:

1. Create an `outputs.conf` file in the master node at `$SPLUNK_HOME/etc/system/local` directory
2. Configure the `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
```

```
[splunk@admin1 ~]$ cd $SPLUNK_HOME/etc/system/local
[splunk@admin1 local]$ vi outputs.conf
[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 autoLB = true
```



Note: It may be advantageous for scalability to use the "indexer discovery" feature instead of statically assigning indexers to forward to.

3. Restart Splunk (`$SPLUNK_HOME/bin/splunk restart`)

Configure Search Head Clustering

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.



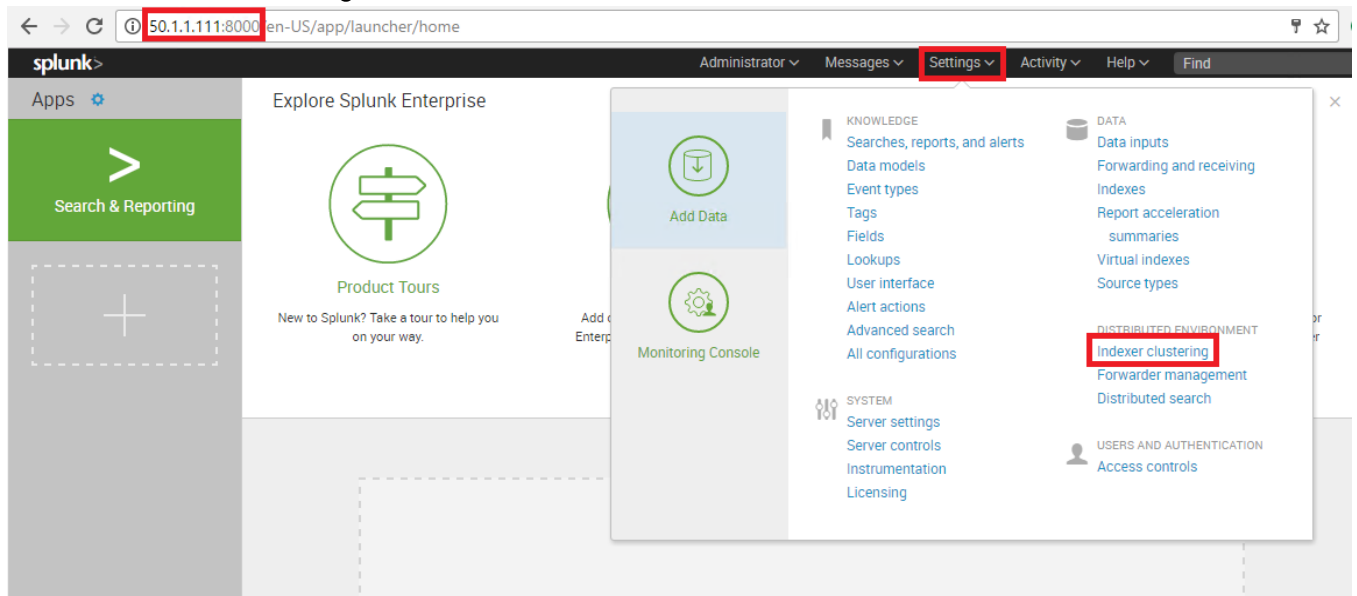
Note: In order to take full advantage of the search head cluster (also described in Splunk Architecture & Terminology), you must utilize a virtual or physical load balancer according to the enterprise's standards. Due to variability, the operator is suggested to use their own discretion in installing and configuring this. Further notes for configuration are provided under the section "Configuring Search Head Load-Balancing".

Add Search Heads to Master Node

A Splunk Enterprise instance can be configured as a search head via the Indexer clustering feature.

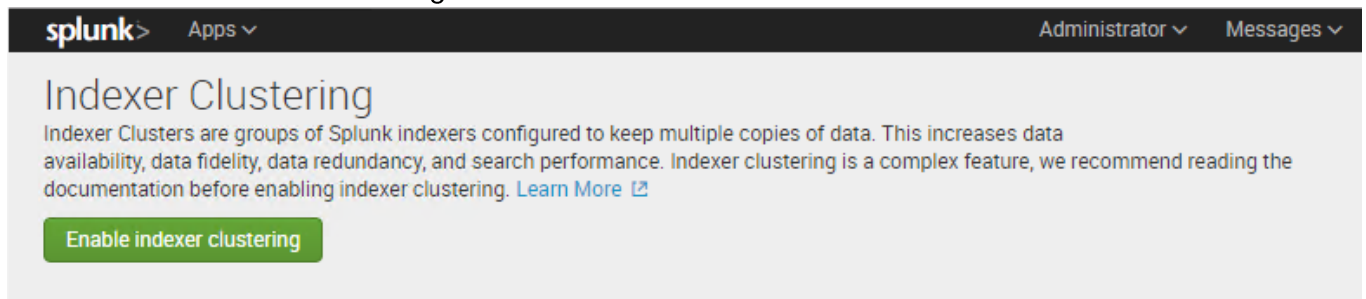
1. Log onto one of the search heads as user admin.
2. Navigate to Settings → Indexer clustering, as shown in Figure 139

Figure 139 Indexer Clustering



3. Click Enable Indexer Clustering.

Figure 140 Enable Indexer Clustering



4. In the Enable Clustering dialog box, click on Search head node.

5. Click `Next`.

Figure 141 Select Node to Enable Clustering

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.

Cancel Next

6. Enter the hostname of the master node including the the Master port number (default: 8089) in the format `https://<hostname_or_IP>`. (For example, <https://admin1:8089>)
7. Enter the same security key that was used while configuring the master node for example, `splunk+cisco`.
8. Click `Enable search head node`.

Figure 142 Search Head Node Configuration

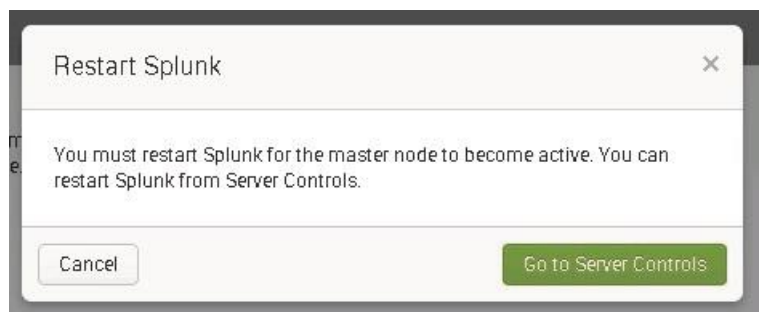
Search head node configuration

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.

Back Enable search head node

9. Click `Go to server controls` to view the `Server controls` screen and to restart Splunk.



10. Repeat the above steps to configure all three servers with hostnames sh1, sh2 and sh3 to be search heads.
11. Verify the search head cluster members in the master node, by navigating to **Setting** > **Indexer clustering**.
12. Click the **Search Heads** tab, as shown in Figure 143

Figure 143 Verify Search Head Cluster Members in the Master Node

Search head name	Status
admin1	Up
sh3	Up
sh2	Up
sh1	Up

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, admin1 is selected to serve this function as well.



Note: 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.

1. Open an SSH session to admin1.
2. Navigate to:

```
$$SPLUNK_HOME/etc/system/local/
```

3. As the user `splunk`, edit `server.conf` to include the following, replacing `<your_secret_key>` with a password such as `splunk+cisco`:

```
[shclustering]
pass4SymmKey = <your_secret_key>
```

```
[license]
active_group = Enterprise

[clustering]
access_logging_for_heartbeats = 1
max_peer_build_load = 5
mode = master
pass4SymmKey = $1$u4JF7Kk+sEEH57FZwg==
replication_factor = 2
service_interval = 1

[shclustering]
pass4SymmKey = splunk+cisco
-- INSERT --
```

4. Restart the admin1 instance.

```
$$SPLUNK_HOME/bin/splunk restart
```

Configure Search Head Cluster Members

1. Log in to a search head as the user `splunk`.
2. Enter the following commands to make this search head join the search head cluster. Change the `mgmt._uri` per respective search head.

```
$$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 splunk+cisco
```

3. Restart Splunk Search Head after the command is issued

```
$SPLUNK_HOME/bin/splunk restart
```

```
[splunk@sh1 ~]$ $SPLUNK_HOME/bin/splunk init shcluster-config -auth admin:cisco123 -mgmt_uri
https://sh1:8089 -replication_port 18081 -replication_factor 2 -conf_deploy_fetch_url https
://admin1:8089 -secret splunk+Cisco
Search head clustering has been initialized on this node.
You need to restart the Splunk Server (splunkd) for your changes to take effect.
[splunk@sh1 ~]$ $SPLUNK_HOME/bin/splunk restart
Stopping splunkd...
Shutting down. Please wait, as this may take a few minutes.
.. [ OK ]
Stopping splunk helpers... [ OK ]
Done.

Splunk> Be an IT superhero. Go home early.

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.
  Checking critical directories... Done
  Checking indexes...
    Validated: _audit_blocksignature _internal_introspection _thefishbucket hi
story main summary
    Done

Bypassing local license checks since this instance is configured with a remote license master.

  Checking filesystem compatibility... Done
  Checking conf files for problems...
  Done
  Checking replication_port port [18081]: open
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
Done [ OK ]

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

The Splunk web interface is at http://sh1:8000
```

4. Repeat the above steps for all search heads.

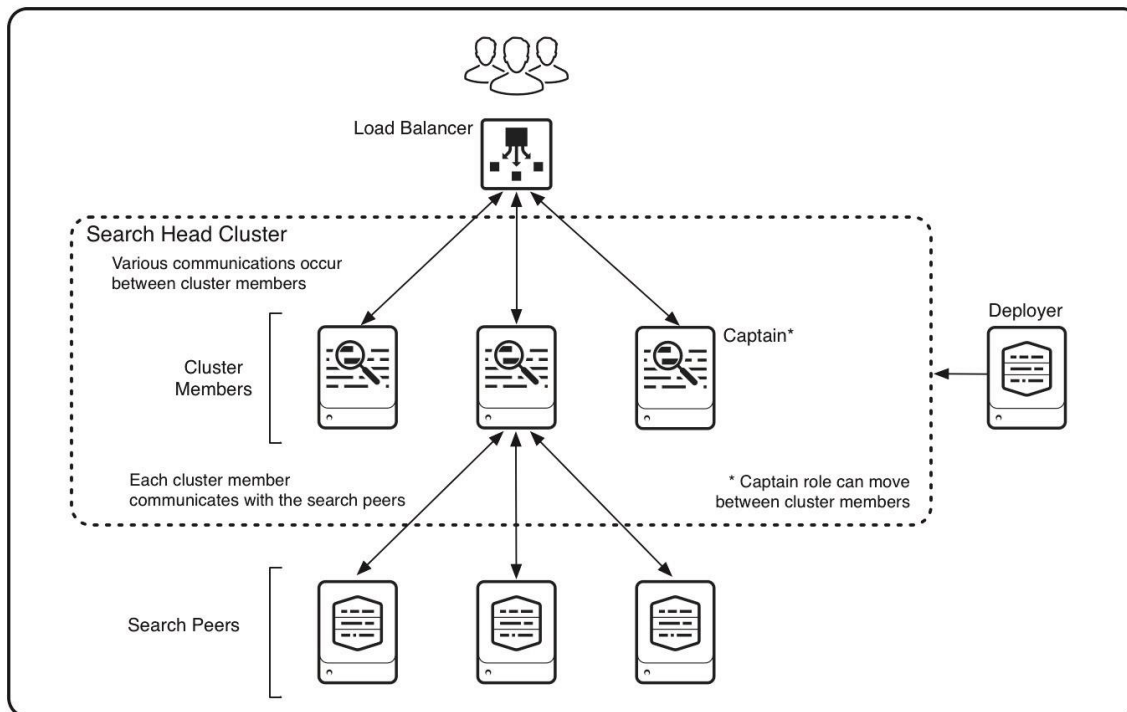
Elect a Search Head Captain

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.

The following illustration shows a small search head cluster, consisting of three members:

Figure 144 Search Head Cluster with its 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.

1. Log into any search head as user `splunk`.
2. Start the search head captain election bootstrap process by using the following command as the `splunk` user. It may take a minute to complete.

```
$SPLUNK_HOME/bin/splunk bootstrap shcluster-captain -servers_list
"https://sh1:8089,https://sh2:8089,https://sh3:8089" -auth admin:cisco123
```

```
[splunk@sh1 ~]$ $SPLUNK_HOME/bin/splunk bootstrap shcluster-captain -servers_list "https://sh1:8089,
https://sh2:8089,https://sh3:8089" -auth admin:cisco123
Successfully bootstrapped this node as the captain with the given servers.
```



Note: The search head captain election process can be started from any of the search head cluster members.

Configure Search Heads to Forward their 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:

- 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.
- By forwarding the results of summary index searches to the indexer level, all search heads have access to them. Otherwise, they are 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, so that the search head does not both retain the data locally as well as forward it to the search peers.

1. Using the CLI, as the `splunk` user on `admin1`, navigate to `$SPLUNK_HOME/etc/shcluster/apps`.
2. Create the directory `outputs` and `outputs/local`.

```
mkdir -p outputs/local
```

3. Navigate to the newly created `local` directory.

```
cd outputs/local
```

4. Within the `$SPLUNK_HOME/etc/shcluster/apps/outputs/local/` directory, create the file `outputs.conf` with the following content in Step 5.

```
vi outputs.conf
```

5. Copy and paste the following contents.

```
# 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
```



Note: It may be advantageous for scalability to use the "indexer discovery" feature instead of statically assigning indexers to forward to.

6. Execute the `apply shcluster-bundle` command:

```
$SPLUNK_HOME/bin/splunk apply shcluster-bundle -target https://sh1:8089 -auth admin:cisco123
```

```
[splunk@admin1 local]$ $SPLUNK_HOME/bin/splunk apply shcluster-bundle -target https://sh1:8089 -auth admin:cisco123
Warning: Depending on the configuration changes being pushed, this command might initiate a rolling restart of the cluster members. Please refer to the documentation for the details. Do you wish to continue? [y/n]: y
Bundle has been pushed successfully to all the cluster members.
[splunk@admin1 local]$
```

```
# 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
```

7. Acknowledge the warning. A message pop-up will notify that the bundle has been pushed successfully.

Configure Search Head Load-Balancing

As described above in the introductory note about search head clustering, it is useful to utilize a load balancer to take advantage of the search head cluster.

1. Designate a common URL for use throughout the enterprise (For example, <https://splunk.domain.com>)
2. The common URL should balance traffic between all three search heads and their respective ports. For example, <https://sh1:8000>, <https://sh2:8000>, <https://sh3:8000>.



Note: Explicit instructions for configuring the designated load balancer will differ by vendor, but the functionality and load balancing direction is the same.

Verify Search Head Clustering:

3. SSH to any search head.
4. As the `splunk` user, issue the command `$SPLUNK_HOME/bin/splunk show shcluster-status -auth <username>:<password>`.

```
$SPLUNK_HOME/bin/splunk show shcluster-status -auth admin:cisco123
```

```
[splunk@sh1 ~]$ $SPLUNK_HOME/bin/splunk show shcluster-status -auth admin:cisco123

Captain:
    dynamic_captain : 1
    elected_captain  : Thu Oct 27 19:46:07 2016
                   id : 199E5633-875C-473D-8A74-4FC935347F74
    initialized_flag : 1
                   label : sh1
                   mgmt_uri : https://sh1:8089
    min_peers_joined_flag : 1
    rolling_restart_flag : 0
    service_ready_flag : 1

Members:
  sh3
    label : sh3
    last_conf_replication : Thu Oct 27 19:48:37 2016
    mgmt_uri : https://sh3:8089
    mgmt_uri_alias : https://50.1.1.113:8089
    status : Up

  sh1
    label : sh1
    mgmt_uri : https://sh1:8089
    mgmt_uri_alias : https://50.1.1.111:8089
    status : Up

  sh2
    label : sh2
    last_conf_replication : Thu Oct 27 19:48:38 2016
    mgmt_uri : https://sh2:8089
    mgmt_uri_alias : https://50.1.1.112:8089
    status : Up
```

5. Alternatively, run `$SPLUNK_HOME/bin/splunk list shcluster-members -auth <username>:<password>` to view the various members.

```
$SPLUNK_HOME/bin/splunk list shcluster-members -auth admin:cisco123
```

```
[splunk@sh1 ~]$ $SPLUNK_HOME/bin/splunk list shcluster-members -auth admin
cisco123
    37386839-F178-49CB-9BEB-BAA7277150E4
        adhoc_searchhead:0
        advertise_restart_required:0
        artifact_count:0
        delayed_artifacts_to_discard:
        fixup_set:
        host_port_pair:192.168.11.113:8089
        kv_store_host_port:192.168.11.113:8191
        label:sh3
        last_heartbeat:1429319835
        peer_scheme_host_port:https://192.168.11.113:8089
        pending_job_count:0
        replication_count:0
        replication_port:18081
        replication_use_ssl:0
        site:default
        status:Up

    4C26874E-A557-4D94-AD8C-5E6B6788AE04
        adhoc_searchhead:0
        advertise_restart_required:0
        artifact_count:0
        delayed_artifacts_to_discard:
        fixup_set:
        host_port_pair:192.168.11.112:8089
        kv_store_host_port:192.168.11.112:8191
        label:sh2
        last_heartbeat:1429319834
        peer_scheme_host_port:https://192.168.11.112:8089
        pending_job_count:0
        replication_count:0
        replication_port:18081
        replication_use_ssl:0
        site:default
        status:Up

    AD4CFC8-B976-4EBB-A0AC-9CF1100F0547
        adhoc_searchhead:0
        advertise_restart_required:0
        artifact_count:0
        delayed_artifacts_to_discard:
        fixup_set:
        host_port_pair:192.168.11.111:8089
        kv_store_host_port:192.168.11.111:8191
        label:sh1
        last_heartbeat:1429319835
        peer_scheme_host_port:https://192.168.11.111:8089
        pending_job_count:0
        replication_count:0
        replication_port:18081
        replication_use_ssl:0
        site:default
        status:Up
```

- Navigate to the directory `$SPLUNK_HOME/etc/apps/outputs/default/` on any search head. List the directory. `outputs.conf` will be listed, verifying that it has been pushed by the Deployer.

```
cd $SPLUNK_HOME/etc/apps/outputs/default
```

```
ls -l
```

```
[splunk@sh2 default]$ pwd
/data/disk1/splunk/etc/apps/outputs/default
[splunk@sh2 default]$ ls -l
total 8
-rw-----. 1 splunk splunk  77 Apr 10 10:13 app.conf
-rw-rw-r--. 1 splunk splunk 296 Apr 10 10:13 outputs.conf
```



Note: This app will not appear under 'apps' within the GUI, but will appear under [Apps](#) → [Manage Apps](#)

Set Cluster Labels

Setting labels on indexer clusters and search head clusters allows the Monitoring Console (which will be configured in the next section) to identify instances associated with them. This identification allows the Monitoring Console to populate the indexer clustering and search head clustering dashboards.

To configure the indexer cluster with the label `idxcluster` and the search head cluster with the label `shc1`, execute these commands from `admin1` as the `splunk` user (you may be prompted to enter admin credentials):

```
splunk edit cluster-config -cluster_label idxcluster
splunk edit shcluster-config -shcluster_label shc1
```

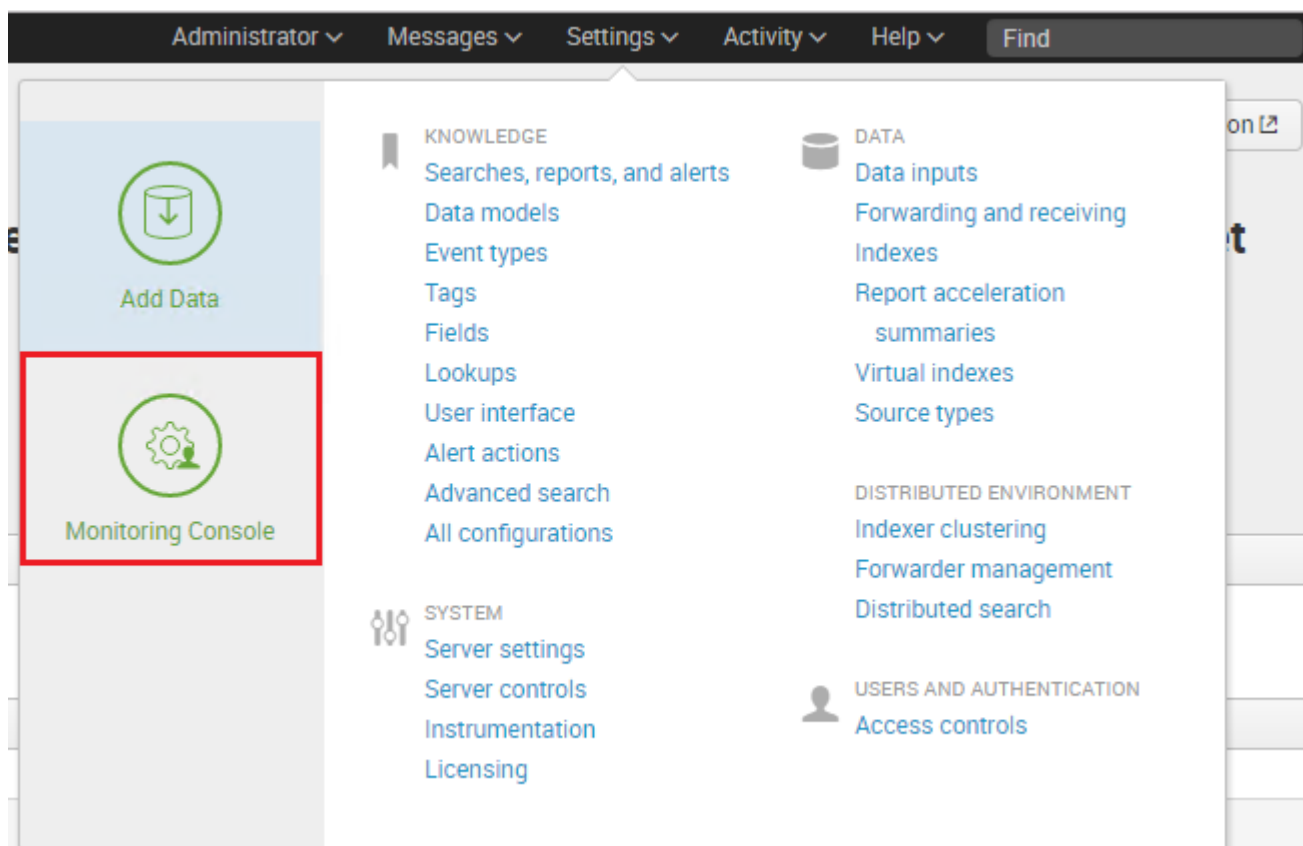
```
[splunk@admin1 ~]$ splunk edit cluster-config -cluster_label idxcluster
Your session is invalid. Please login.
Splunk username: admin
Password:
The cluster-config property has been edited.
[splunk@admin1 ~]$ splunk edit shcluster-config -shcluster_label shc1
The shcluster-config property has been edited.
```

Configuring the Monitoring Console

The distributed management console is a special purpose pre-packaged app that comes with Splunk Enterprise providing detailed performance information about the Splunk Enterprise deployment.

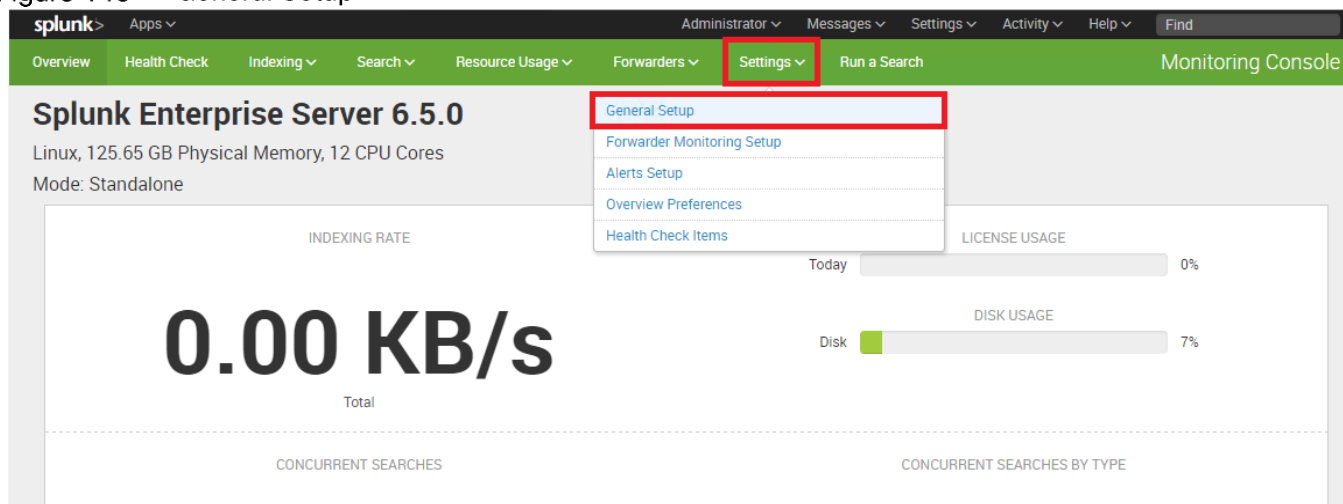
This section describes the procedure to configure the Distributed Management Console for this deployment. It is installed on the admin node, that is, `admin1`. Please refer to [[Splunk Documentation](#)] for learning about other installation options.

1. Navigate to the Splunk Web Interface on `admin1` (<https://admin1:8000/>).
2. Click [Settings](#) → [Monitoring Console](#), as shown below. This was formerly called the Distributed Management Console.



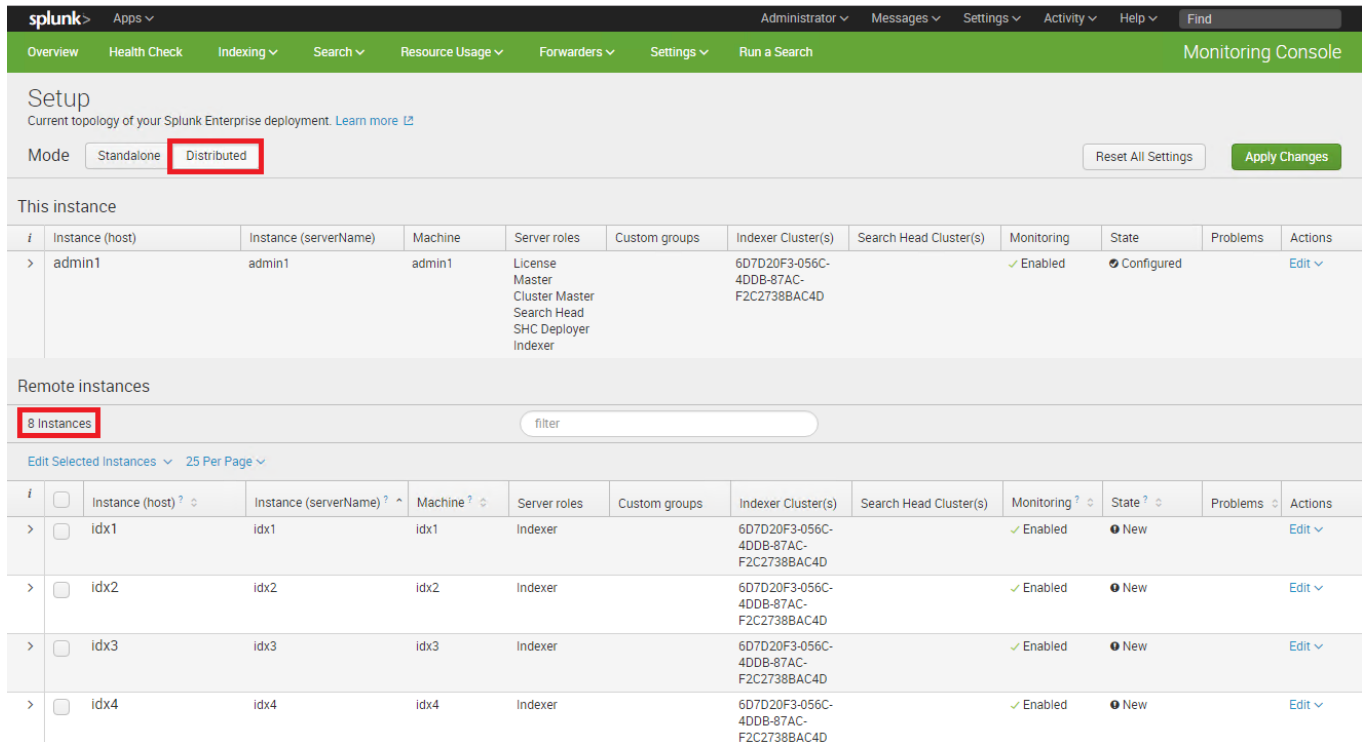
3. In the Monitoring Console app, click **Settings** → **General Setup**, as shown in Figure 145

Figure 145 General Setup



4. Click the **Distributed mode**. Click **Continue** through the warning (make sure this is on admin1). This should show all eight indexers as remote instances. See Figure 146

Figure 146 Indexers as Remote Instances



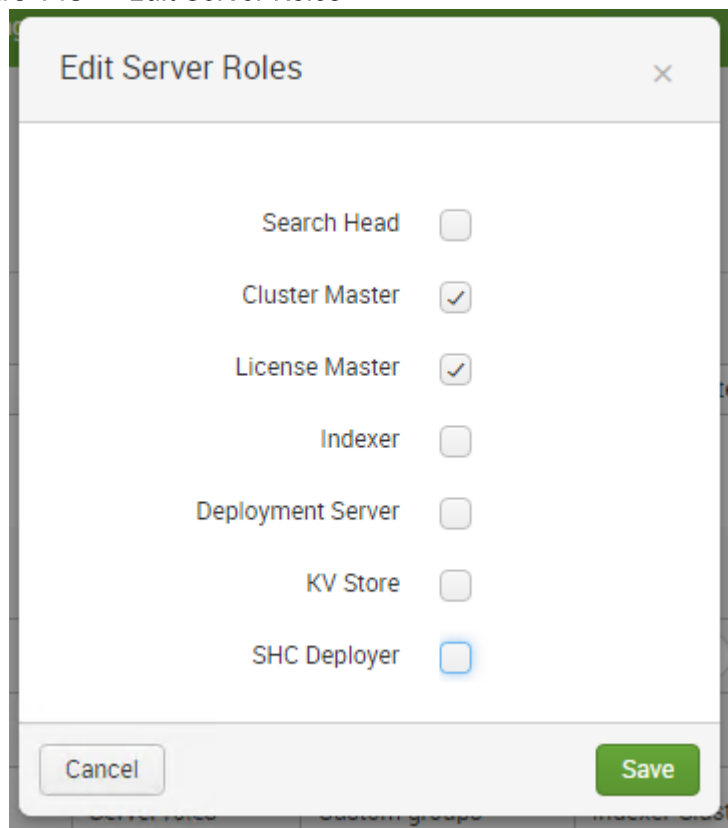
5. Select **Edit** on the admin1 box. The server must change roles to function properly. See Figure 147 for the current configuration that needs to be changed.

Figure 147 Change Server Roles



6. Select only **Cluster Master** and **License Master**, as shown below.

Figure 148 Edit Server Roles



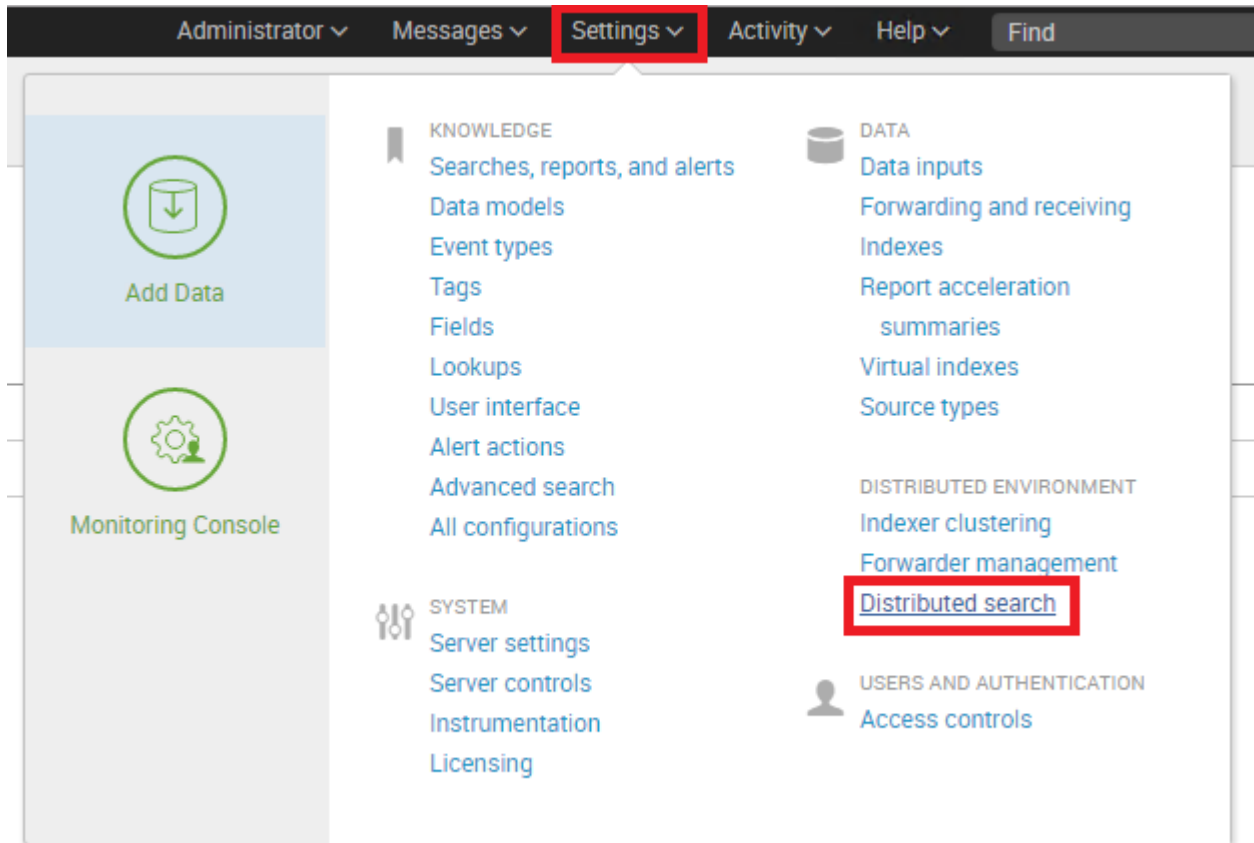
7. Click `Save`.
8. Click the `Apply Changes` button at the top right.
9. Confirm that changes have been saved successfully.

Configure Search Heads in the Monitoring Console

In the previous section the Monitoring Console was configured to manage the indexers and the master node. This section provides the procedure to configure the Monitoring Console to monitor the search heads.

1. Navigate to the Master Node (`admin1`) via the GUI.
2. Open `Settings` → `Distributed Environment` → `Distributed search`.

Figure 149 Select Distributed Search



3. Select Search Peers, as shown in Figure 150

Figure 150 Search Peers



4. Select New.

Figure 151 Selecting New Peers



Figure 152 Add Search Peers

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 *

Search peer is either `servername:management_port` or `IP:management_port`. For example `'myhost: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

Cancel Save

5. Add a search Peer.

[Peer] – Enter the hostname or IP of one of your search heads

[Remote username] – use admin

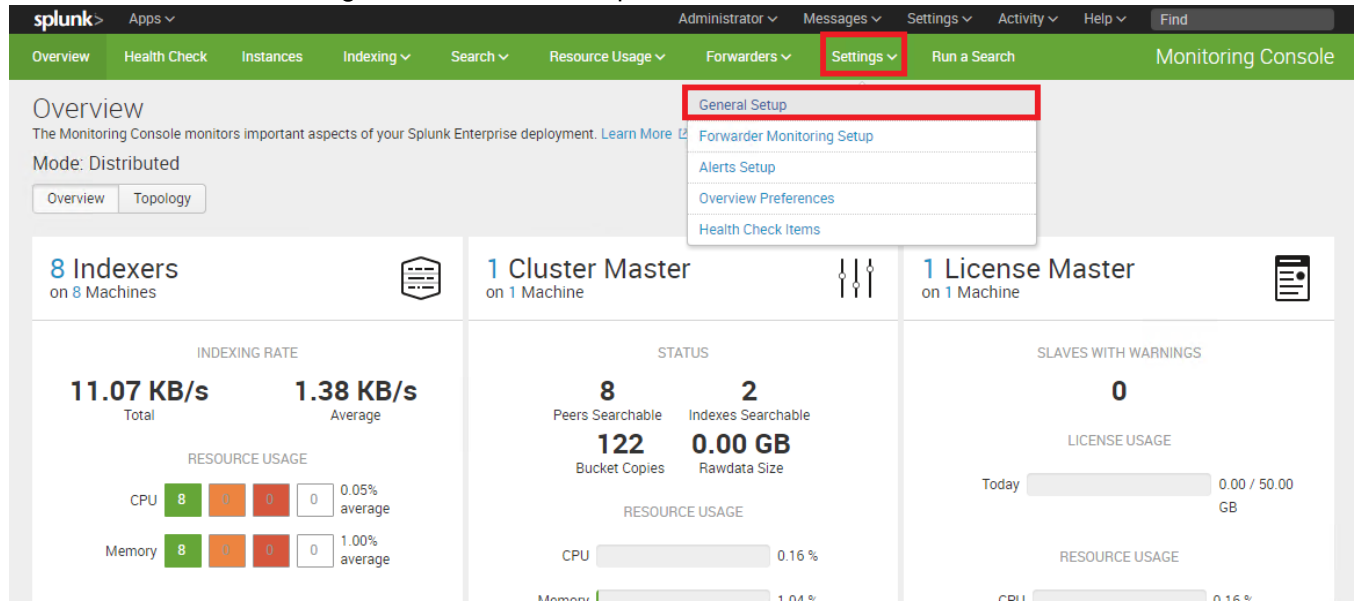
[Remote password] – the password to the Splunk admin account on the search head

6. Repeat this process for the other two search heads. After this is completed, you should see eight indexers and three search heads in the Search peers screen.

7. On the Master Node (admin1), navigate to Settings > Monitoring Console.

8. In the Monitoring Console app, click Settings > General Setup, as shown in Figure 153

Figure 153 Distributed Management Console: Setup



9. The three newly added search heads should be listed under `Remote` instances with a State of `New`.

10. Ensure that the server roles are `Search Head` and `KV Store`, as shown in Figure 154

Figure 154 Verify Server Roles

Remote instances					
11 Instances				filter	
Edit Selected Instances ▾ 25 Per Page ▾					
<i>i</i>	<input type="checkbox"/>	Instance (host) ? ↕	Instance (serverName) ? ^	Machine ? ↕	Server roles
>	<input type="checkbox"/>	idx1	idx1	idx1	Indexer
>	<input type="checkbox"/>	idx2	idx2	idx2	Indexer
>	<input type="checkbox"/>	idx3	idx3	idx3	Indexer
>	<input type="checkbox"/>	idx4	idx4	idx4	Indexer
>	<input type="checkbox"/>	idx5	idx5	idx5	Indexer
>	<input type="checkbox"/>	idx6	idx6	idx6	Indexer
>	<input type="checkbox"/>	idx7	idx7	idx7	Indexer
>	<input type="checkbox"/>	idx8	idx8	idx8	Indexer
>	<input type="checkbox"/>	sh1	sh1	sh1	Search Head KV Store
>	<input type="checkbox"/>	sh2	sh2	sh2	Search Head KV Store
>	<input type="checkbox"/>	sh3	sh3	sh3	Search Head KV Store

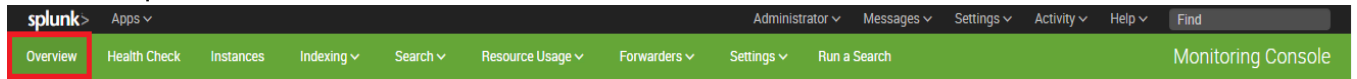
11. Edit server roles and custom groups, if needed. Confirm changes and roles.
12. Ensure that the Master Node (admin1) does not have the role of Search Head.
13. Click `Apply Changes`, as shown in Figure 155

Figure 155 Click on Apply Changes after Verifying

Setup											
Current topology of your Splunk Enterprise deployment. Learn more											
Mode <input type="radio"/> Standalone <input type="radio"/> Distributed								Reset All Settings		Apply Changes	
This instance											
<i>i</i>	Instance (host)	Instance (serverName)	Machine	Server roles	Custom groups	Indexer Cluster(s)	Search Head Cluster(s)	Monitoring	State	Problems	Actions
>	admin1	admin1	admin1	Cluster Master License Master		idxcluster		✓ Enabled	⚙ Configured		Edit ▾

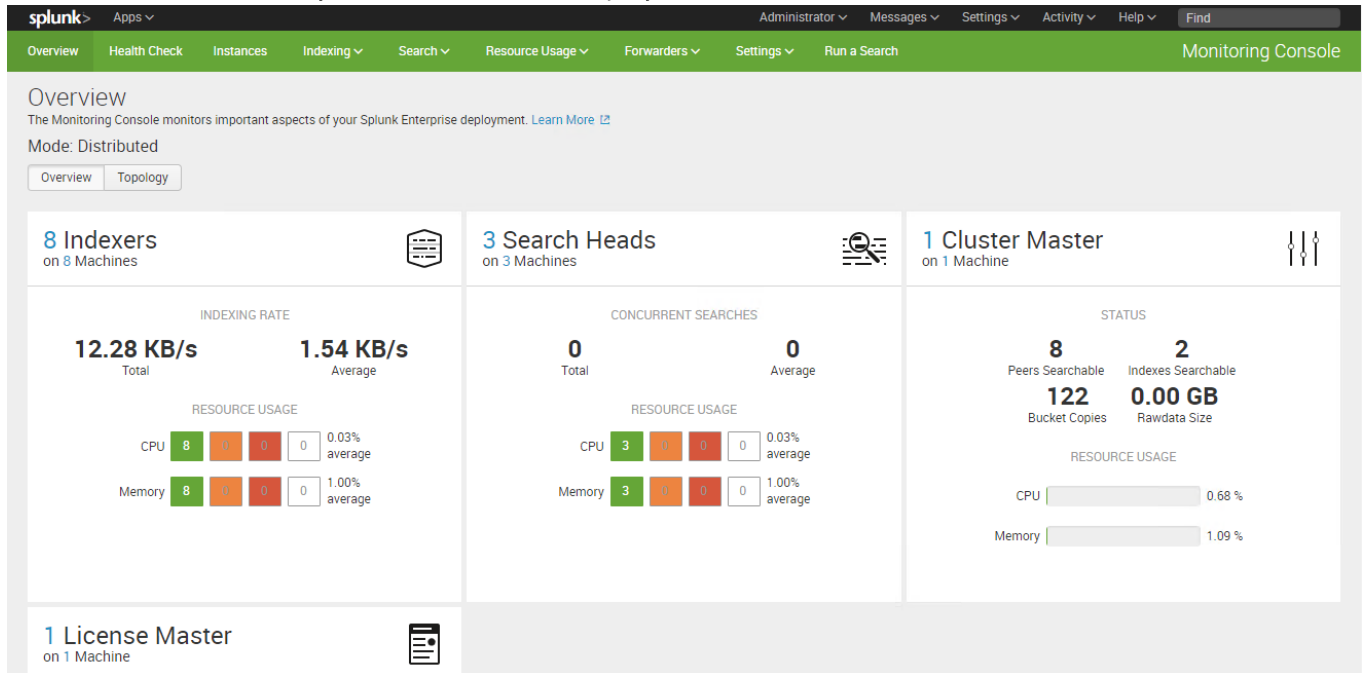
14. Click `Overview`.

Figure 156 Splunk Overview



15. DMC should now display “Search Heads” within the overview. Figure 157

Figure 157 Overview: Verify the Search Heads Displayed



Configuring Multiple Volumes for Hot, Warm, and Cold Data

As the indexer indexes the incoming data, it creates two types of files:

- The raw data in compressed form (rawdata files)
- Indexes that point to the raw data, plus some metadata (index 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**.

- An "index" contains compressed raw data and associated index files.
- An index resides across many age-designated index directories.
- An index directory is called a bucket.

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. In the section [Verifying Master and Peer Replication](#), an index will be created for the purposes of testing. Different configurations will apply to indexes as the Splunk installation matures.

For testing purposes only, an index will be pushed from the master node (`admin1`) in the verification stage of this CVD by applying the following stanza:

```
[archival]

coldToFrozenDir = /path/to/frozen
```

More information regarding archival can be found in the [documentation](#)

Configuring the Deployment Server

In this section, the server `admin2` is configured to function as the Deployment server, and procedure to push a sample "Splunk App" from the Deployment Server to a Universal Forwarder on a test server (not part of this CVD).

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/`.

In the following section, a Universal Forwarder will be installed on a machine separate from the servers that make up the Splunk Enterprise platform of this CVD. The only requirement for this is it must be reachable via the same network to which the Indexers are connected to.

Once the machine is connected to the network with connectivity to the UCS platform, follow the steps below.



Note: In this documentation, it is assumed that the machine with Universal Forwarder is reachable via 192.168.11.0/24 network (in other words via NIC eth1 of the Cisco UCS servers). This would require the respective VLANs configured appropriately to provide appropriate connectivity between the Cisco UCS infrastructure on which Splunk platform is built and the server with a universal forwarder.



Note: The Deployment Server is installed by default when Splunk Enterprise is deployed. In this CVD, the admin2 box will function as the designated Deployment Server.

Installing a Universal Forwarder on a Test Server

1. Download Splunk Universal Forwarder: <http://www.splunk.com/download/universalforwarder>
2. Install the package as detailed in the [documentation](#) for the appropriate operating system of the Universal Forwarder host.

Register Universal Forwarder with the Deployment Server

1. Via the command line, access the system hosting the Universal Forwarder.
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 = admin2:8089
```

`clientName` = the name or identifier of the host system

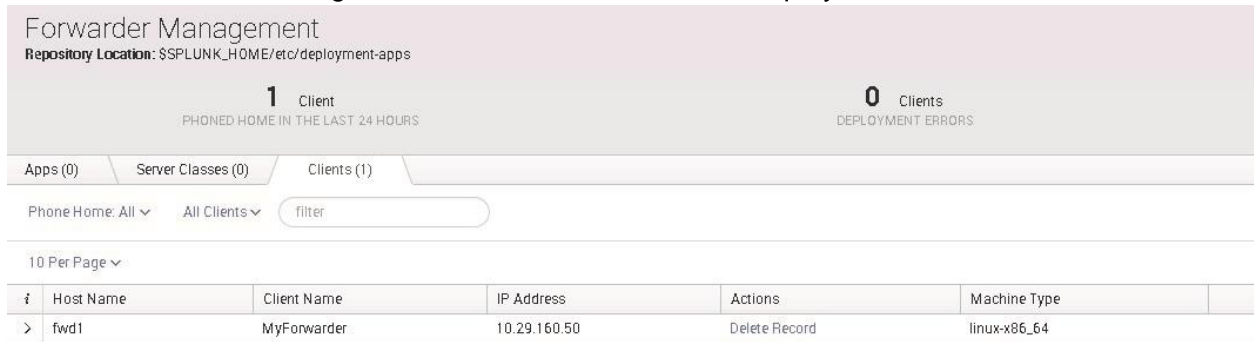
`targetUri` = the hostname/IP and port of the Deployment Server (for example, admin2:8089)

4. As the user `splunk`, restart the Universal Forwarder (`$SPLUNK_HOME/bin/splunk restart`)

Configure an App within the Deployment Server

1. In a browser, navigate to the Splunk instance's Web Interface of server admin2. (that is, `https://admin2:8000/`)
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), as shown in Figure 158

Figure 158 Forwarder Management: Communication with the Deployment Server



Forwarder Management
Repository Location: \$SPLUNK_HOME/etc/deployment-apps

1 Client
PHONED HOME IN THE LAST 24 HOURS

0 Clients
DEPLOYMENT ERRORS

Apps (0) Server Classes (0) Clients (1)

Phone Home: All All Clients filter

10 Per Page

i	Host Name	Client Name	IP Address	Actions	Machine Type
>	fwd1	MyForwarder	10.29.160.50	Delete Record	linux-x86_64

4. Using the command line, navigate to the Deployment Server, admin2.
5. Navigate to `$SPLUNK_HOME/etc/deployment-apps/`
6. Create the directory `appTest`.
7. Within `appTest` create the directory `local`.

```
[root@admin2 ~]# cd $SPLUNK_HOME/etc/deployment-apps
[root@admin2 deployment-apps]# mkdir appTest
[root@admin2 deployment-apps]# cd appTest
[root@admin2 appTest]# mkdir local
[root@admin2 appTest]# cd local
[root@admin2 local]# vi app.conf
```

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
```

```
[root@admin2 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
[root@admin2 local]#
```

9. As the `splunk` user, execute the command to reload the deployment server:

```
$SPLUNK_HOME/bin/splunk reload deploy-server
```

```
[splunk@admin2 local]$ $SPLUNK_HOME/bin/splunk reload deploy-server
Your session is invalid. Please login.
Splunk username: admin
Password:
Login successful, running command...
Reloading serverclass(es).
```



Note: The login step could be bypassed by appending `auth admin:cisco` to the command line.

10. Navigate to the Web GUI on admin2 (<http://admin2:8000>) and navigate to **Settings > Forwarder Management**. Click the **Apps** tab, as shown in Figure 159

Figure 159 Forwarder Management

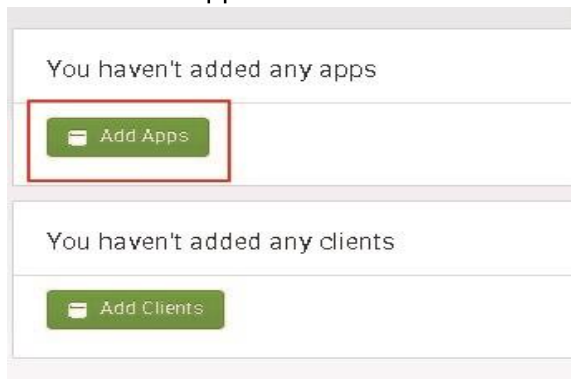
Name	Actions	After Installation	Clients
appTest	Edit	Enable App	0 deployed

11. Zero apps have been deployed. Click **Server Class**.
12. Click **Create One** and give it the name `TestForwarder`.

Figure 160 Create Server Class

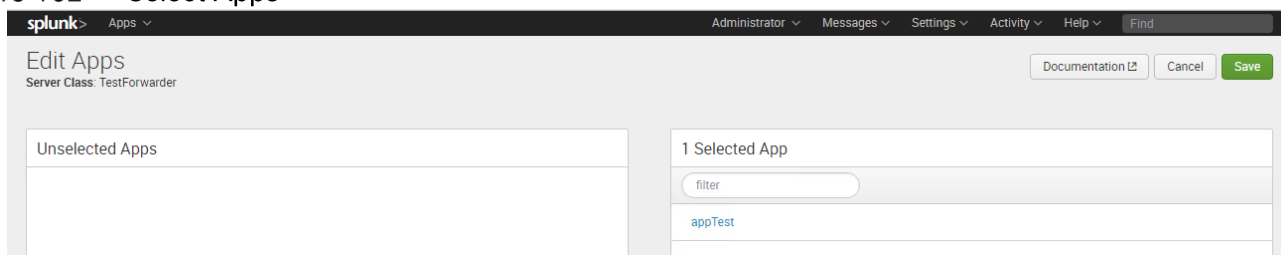
13. Figure 161 presents options for adding apps and clients. Click **Add Apps**.

Figure 161 Add Apps to New Server Class



14. Click `appTest` in the `Unselected Apps` section to move it to `Selected Apps` section. Figure 162

Figure 162 Select Apps

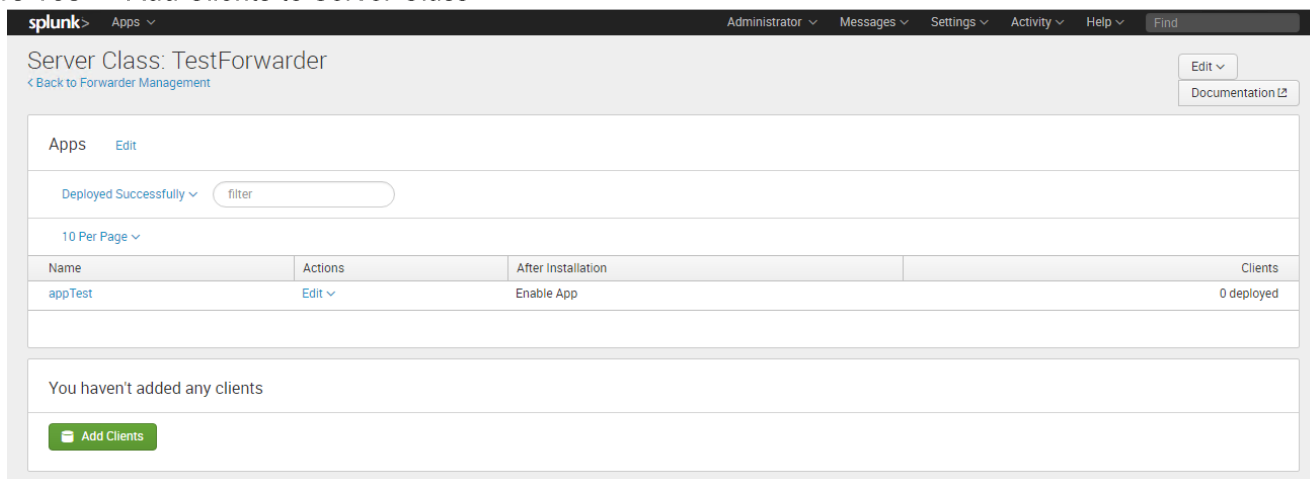


15. Click the `Save` button in the upper right.

16. The next screen will show the apps listed under `Apps`.

17. Click the `Add Clients` button, as shown in Figure 163

Figure 163 Add Clients to Server Class



18. Within the `Edit Clients` screen, add the hostname of the forwarder to the whitelist. In this instance, the forwarder used is named `fwd1`. See Figure 164

Figure 164 Edit Clients

Include (whitelist)

Exclude (blacklist)

Filter by Machine Type (machineTypesFilter)

optional

Can be client name, host name, IP address, or DNS name.
Examples: 192.2.3.*, fwd1-*

Can be client name, host name, IP address, or DNS name.
Examples: rom1e, rar1ty

Learn more [↗](#)

Learn more [↗](#)

Cancel Preview Save

All Matched Unmatched filter

10 Per Page

Matched	Host Name	DNS Name	Client Name	IP Address	Machine Type	Phone Home
	fwd1	192.168.11.50	MyForwarder	192.168.11.50	linux-x86_64	6 minutes ago

19. Click Save.

20. Go back to the Forwarder Management screen. (Select Settings → Distributed Environment → Forwarder Management).

Figure 165 Forwarder Management

Forwarder Management [Documentation](#)

Repository Location: \$SPLUNK_HOME/etc/deployment-apps

1 Client
PHONED HOME IN THE LAST 24 HOURS

0 Clients
DEPLOYMENT ERRORS

0 Total downloads
IN THE LAST 1 HOUR

Apps (1) Server Classes (1) Clients (1)

Phone Home: All All Clients filter

10 Per Page

#	Host Name	Client Name	IP Address	Actions	Machine Type	Deployed Apps	Phone Home
>	fwd1	MyForwarder	192.168.11.50	Delete Record	linux-x86_64	1 deployed	6 minutes ago

21. On the forwarder box, navigate to `$SPLUNK_HOME/etc/apps`. List the directory to view the newly deployed app.

```
[root@fwd1 apps]# pwd
/opt/splunkforwarder/etc/apps
[root@fwd1 apps]# ls -la
total 4
drwxr-xr-x 7 splunk splunk 117 Apr 10 12:10 .
drwxr-xr-x 13 splunk splunk 4096 Mar 27 09:43 ..
drwxr-xr-x 4 splunk splunk 30 Mar 27 01:19 introspection_generator_addon
drwxr-xr-x 5 splunk splunk 47 Mar 27 09:43 learned
drwx----- 4 root root 33 Apr 10 12:10 outputTest
drwxr-xr-x 5 splunk splunk 49 Mar 27 01:19 search
drwxr-xr-x 4 splunk splunk 35 Mar 27 01:19 SplunkUniversalForwarder
[root@fwd1 apps]#
```

Installation Verification

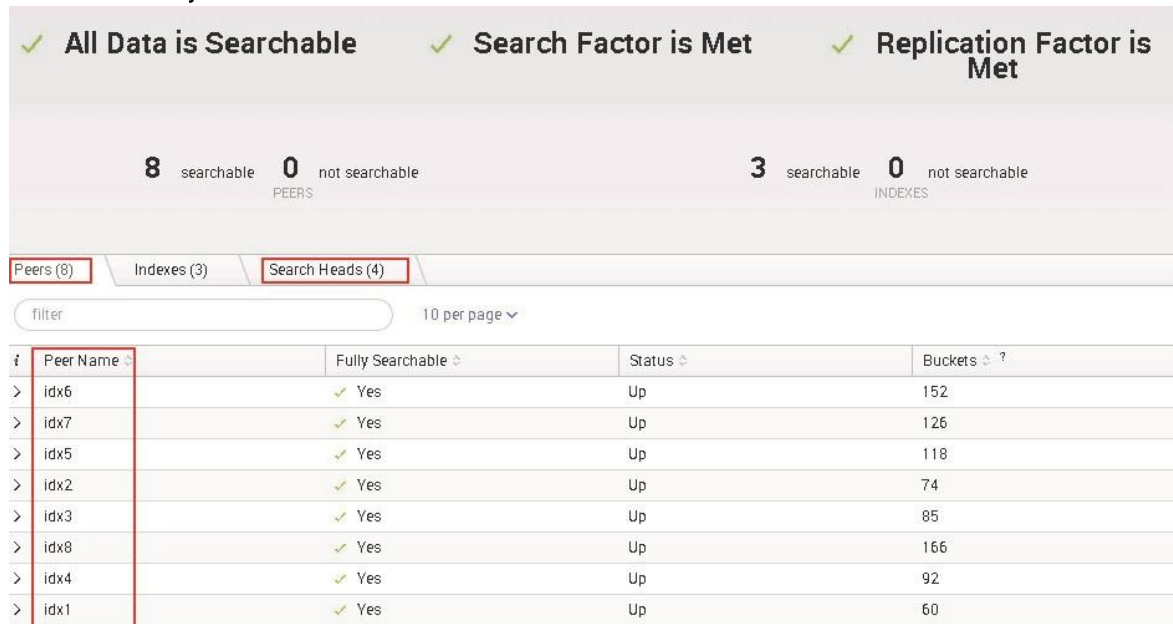
The purpose of this verification is to ensure connectivity between the indexers, search heads, license master, master node, and the distributed management console.

Verifying from DMC

1. Log into the Master Node (admin1).

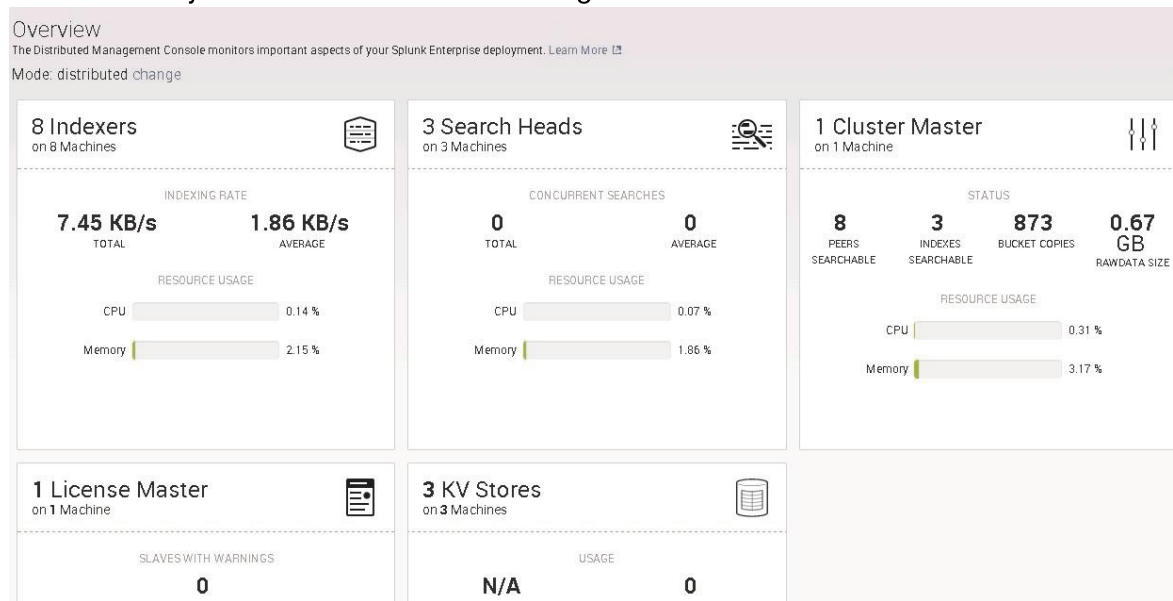
2. Navigate to Settings → Indexer Clustering.
3. Verify that all search heads and indexers are listed, as shown in Figure 166

Figure 166 Verify Search Heads and Indexers



4. Navigate to Settings → Distributed Management Console.
5. The overview page should display similar results (8 Indexers, 3 Search Heads, 1 Cluster Master, 1 License Master, 3 KV stores).

Figure 167 Verify the Nodes in the Overview Page



6. Navigate to Settings → General Setup.

7. Ensure that the Master Node (admin1) is not attributed with the role of Search Head. If so, edit the role to only reflect License Master and Cluster Master, as shown in Figure 168

Figure 168 Verify the Nodes

This instance								
#	Instance (host)	Instance (serverName)	Machine	Actions	Server roles	Custom groups	Status	State
>	admin1	admin1	admin1	Edit ▾	Cluster Master License Master		✓ Enabled	⚙ Configured

Verifying Master and Peer Replication

The purpose of this test is to ensure that indexes are distributed across each peer. By creating an index for testing, as well as a small retention time-frame, we will force the instance to delete data that is rolled to the frozen bucket.

1. SSH into the master node (admin1) as the splunk user.
2. Navigate to `$SPLUNK_HOME/etc/master-apps/_cluster/local/`
3. Create and edit the file `indexes.conf`.

```
[splunk@admin1 ~]$ cd $SPLUNK_HOME/etc/master-apps/_cluster/local
[splunk@admin1 local]$ vi indexes.conf
```

4. Add the following stanzas:

```
### TESTING PURPOSES ONLY ###

[archival]

repFactor = auto

homePath = $SPLUNK_DB/archival/db

coldPath = $SPLUNK_DB/archival/colddb

thawedPath = $SPLUNK_DB/archival/thaweddb

maxDataSize = 1024

frozenTimePeriodInSecs = 3600
```

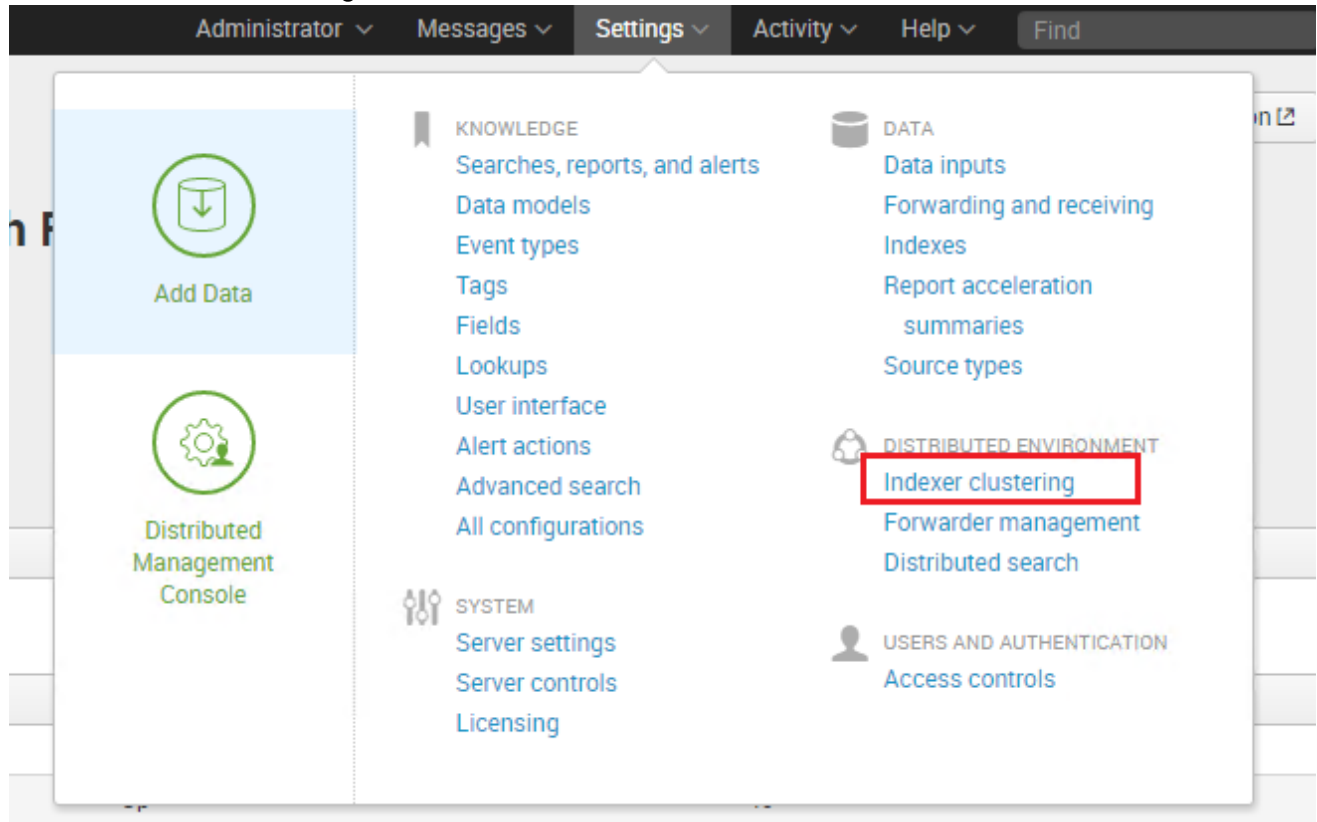
```
### TESTING PURPOSES ONLY ###
[archival]
repFactor = auto
homePath = $SPLUNK_DB/archival/db
coldPath = $SPLUNK_DB/archival/colddb
thawedPath = $SPLUNK_DB/archival/thaweddb
maxDataSize = 1024
frozenTimePeriodInSecs = 3600
```



Note: This test index configuration make use of the frozen data path that was created in the earlier section. See NFS Configurations for the Splunk Frozen Data Storage. Save the file

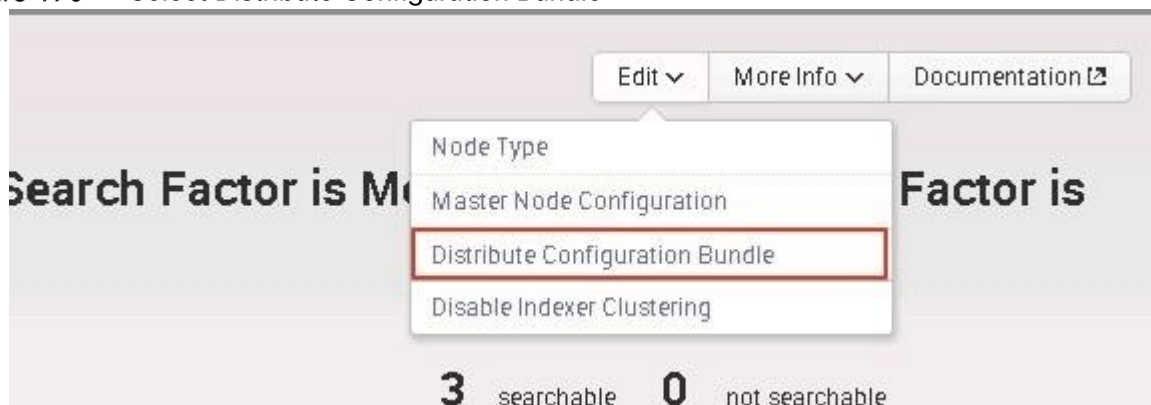
5. Log in to the Web Interface of the Master Node (<http://admin1:8000>).
6. Navigate to `Settings` → `Distributed Environment` → `Indexer Clustering`, as shown in Figure 169

Figure 169 Indexer Clustering



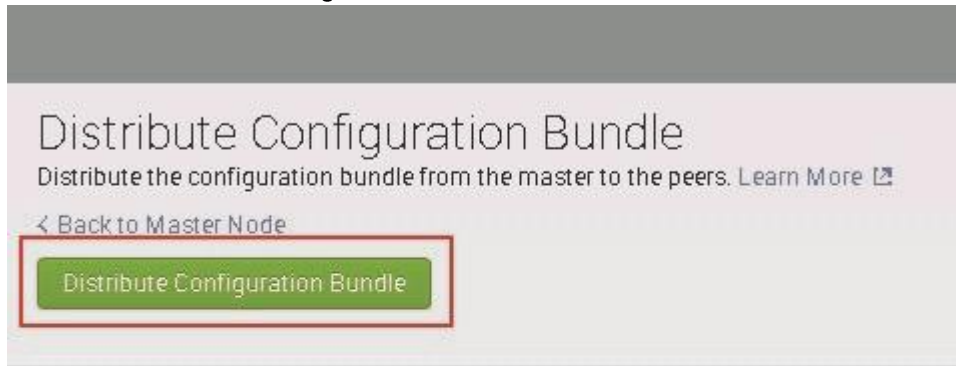
7. Click `Edit` → `Distribute Configuration Bundle`.

Figure 170 Select Distribute Configuration Bundle



8. Click `Distribute Configuration Bundle`.

Figure 171 Distribute Configuration Bundle



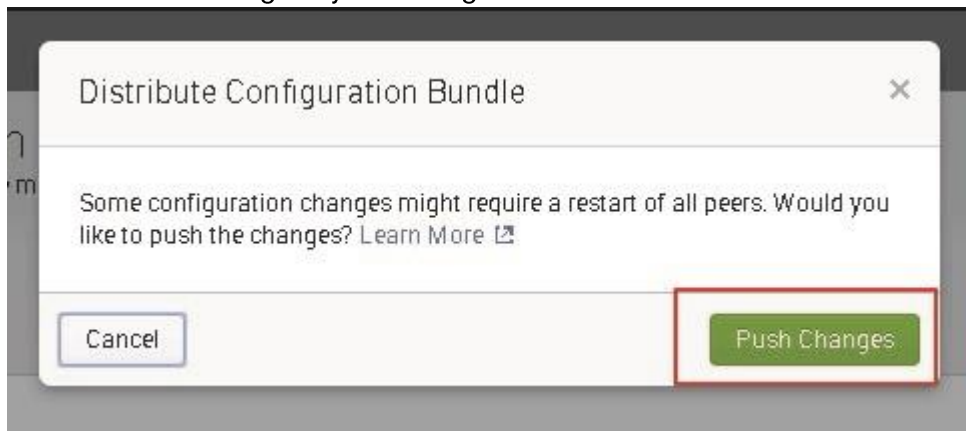
Last Push: ✓ Successful

Time 3/24/2015, 3:31:25 PM

Bundle ID ? 0DE75C59B77A4F1B3296FB0E75B1D750

9. A pop-up window will appear. Select `Push Changes`. Figure 172

Figure 172 Push Changes by Restarting the Peers



10. Verify that the push was successful.

Figure 173 Details of the Last Push



11. SSH to any indexer as the `splunk` user.

12. Navigate to the directory `$SPLUNK_HOME/etc/slave-apps/_cluster/local/`

13. Verify that the new `indexes.conf` file exists.

```
[splunk@idx1 ~]$ cd $SPLUNK_HOME/etc/slave-apps/_cluster/local
[splunk@idx1 local]$ ls -l
total 12
-rw-rw-r-- 1 splunk splunk 274 Apr 18 04:19 indexes.conf
-rw-rw-r-- 1 splunk splunk  41 Apr 18 04:19 inputs.conf
-r--r--r-- 1 splunk splunk 231 Apr 18 04:19 README
[splunk@idx1 local]$ cat ./indexes.conf
### TESTING PURPOSES ONLY ###
[archival]
repFactor = auto
homePath   = $SPLUNK_DB/archival/db
coldPath   = $SPLUNK_DB/archival/colddb
thawedPath = $SPLUNK_DB/archival/thaweddb
maxDataSize = 1024
frozenTimePeriodInSecs = 3600
coldToFrozenDir = /data/frzn_data/archival/frzn
```

Verifying Data Replication

Next, verify that data is distributed across indexer nodes and is replicated across 'live' nodes when an indexer is down. In order to verify that the indexers are replicating data, the indexers must have a sample set of data to work with.

Any random syslog or file in which each line is a new event is acceptable. 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 1million events. If a file is not available for testing, one may be found [here](#). Alternatively, you may use a random syslog generator.

Default throughput of universal forwarders is 256KBps. This may be increased by editing the `maxKBps` stanza in `$SPLUNK_HOME/etc/system/local/limits.conf`. Consult the [documentation](#) for more information. When testing, larger max KBps rates may be used (this configuration tested with 10240) but this may not be suitable for all environments depending upon network infrastructure.

Previously a universal forwarder was configured in the section "Configure the Universal Forwarder". It must be accessible from the same network which UCS is attached to.

Once the new system is available, follow the steps below:

1. From a command line interface on the universal forwarder system, enter the following command:

```
/opt/splunkforwarder/bin/splunk add oneshot -source ./your_test_file.log -
sourcetype syslog -index archival -auth admin:your_admin_password
```

```
[root@fwd1 data]# /opt/splunkforwarder/bin/splunk add oneshot -source ./test.log
-sourcetype syslog -index archival -auth admin:changeme
Warning: overriding $SPLUNK_HOME setting in environment ("/opt/splunk") with "/o
pt/splunkforwarder". If this is not correct, edit /opt/splunkforwarder/etc/splu
nk-launch.conf
Oneshot '/DATA/sdb/sbk/sbk/data/test.log' added
[root@fwd1 data]# █
```



Note: The `$SPLUNK_HOME` on universal forwarder may be set to `/opt/splunkforwarder` to avoid the warning seen in the screenshot above.

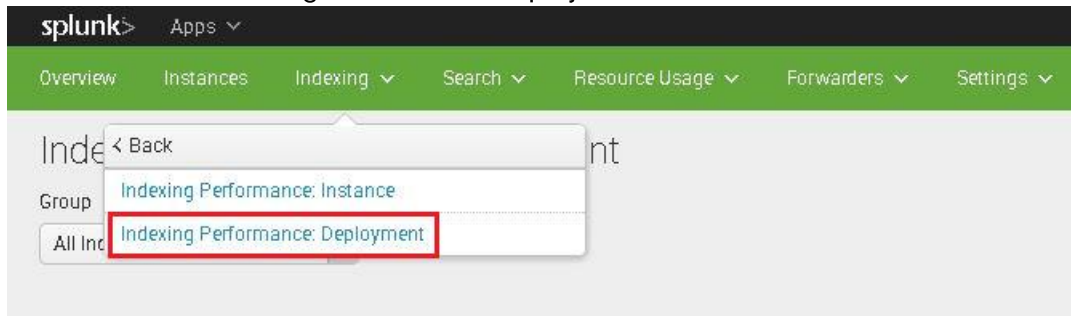
2. Screen will echo Oneshot `'your_test_file.log'` added.
3. Note the time that this step was executed.



Note: The time that this export was executed will be used to verify data transport to the archival node.

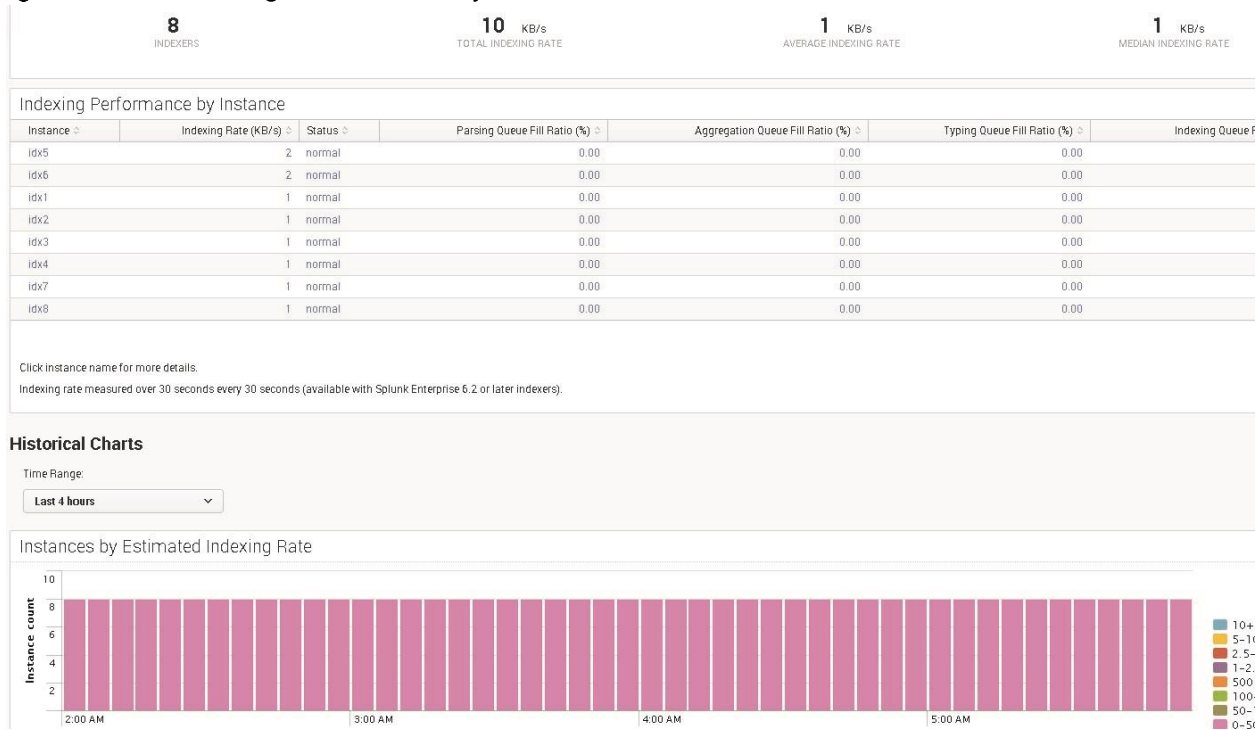
4. Issue the same command, but alter the stanza from “-index archival” to “-index main”. One dataset will be within two indexes.
5. Navigate to the DMC and your Master Node (admin1) in your browser by going to Settings → Distributed Management Console.
6. Select Indexing → Performance → Indexing Performance Deployment.

Figure 174 Select Indexing Performance Deployment



Note that the DMC reflects indexing rates and data passing through the system.

Figure 175 Indexing Performance by Instance




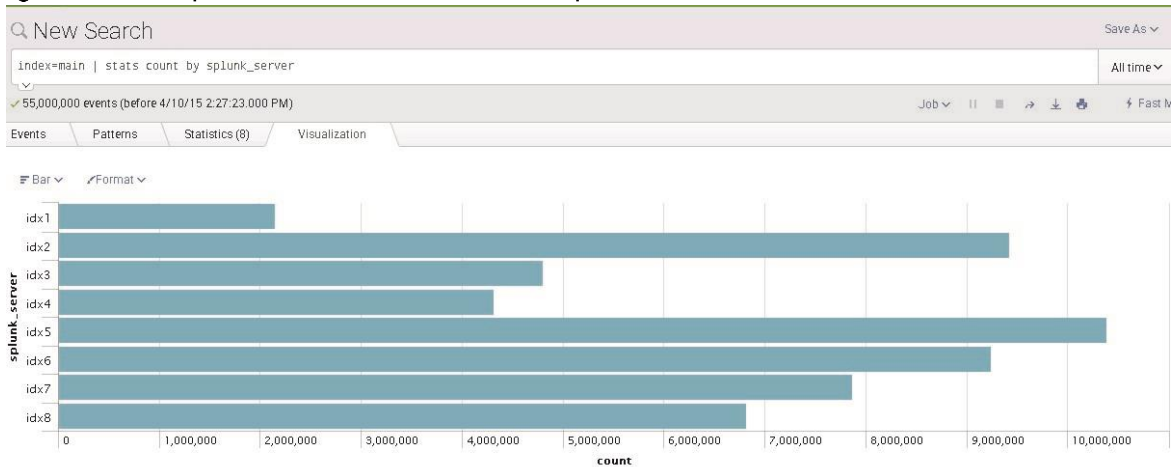
- Navigate to any of the search heads in your browser.
- Click on **Searching and Reporting**.
- In the Splunk search bar, enter the following search: `index="archival" | stats count by splunk_server`
- Note the indexer(s).
- In the Splunk search bar, enter the following search: `index="main" | stats count by splunk_server`
- Change the time range picker to **All time**.
- Change the search mode to **fast mode**.
- Click on **search** (magnifying glass) .
- Change view to **Visualization** and set the chart type to 'Column'. Note the distribution of data across each of the indexers, as shown in Figure 176

Figure 176 Splunk Server Versus Count Graph



16. Change the view to *Statistics*. Note 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. See Figure 177

Figure 177 Splunk Server Versus Count: Statistics

splunk_server	count	Total
idx1	629072	
idx3	59511	
idx4	311417	
	1000000	Total

17. Write down or take a screenshot of the totals per indexer for reference later.

18. Use shell access to navigate to one of the indexers that reported data.

19. Approximately 5 minutes after sending data to the indexers, proceed to step 24.

20. Issue the command `$SPLUNK_HOME/bin/splunk offline`.

21. Return to the browser and run the same search again, as shown in Figure 178

Figure 178 Splunk Server per Indexer Graph



Note: Jot down the distribution of events and the total. The event count from this search and the previous (step 18) should be the same. Bring your indexer back up (`$SPLUNK_HOME/bin/splunk start`).

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.

If the test does not present data across the indexers, the most common reasons for failure are listed below:

- The universal forwarder does not have the appropriate configurations listed in 'outputs.conf'
- There are network connectivity issues between the Universal Forwarder and the assigned receiving port on the indexers
- The dataset was large and has not finished replication to other indexers in the allotted amount of time

Verifying Transfer of Cold to Frozen Buckets

In the previous test, the time that a 'oneshot' to the index 'archival' was performed was noted (Step 8). In this configuration, this setting is for one hour. For more information on frozen data, see [Configuring Archival of Data From Cold to Frozen](#).



Note: This is NOT a recommended setting, but simply for quick testing of archival transfer. You cannot verify the archival of frozen data until one hour has passed from the time of indexing.

1. As the 'splunk' user, SSH to the indexer which reported data ("verifying data replication: step 11") 2. Navigate to `/data/disk2/archival/frzn`
2. Issue the command:

```
ls -la
```

This displays the ‘frozen’ bucket(s) that have been moved.

```
[splunk@idx1 frzn]$ ls -la
total 12
drwxrwxrwx 4 splunk splunk 4096 Apr  6 21:19 .
drwxrwxrwx 4 splunk splunk  28 Feb  2  2010 ..
drwx--x--x 3 splunk splunk  20 Apr  6 19:16 db_1428343271_1428298175_0_ED614B15-7220-4
14
drwx--x--x 3 splunk splunk  20 Apr  6 21:19 db_1428354209_1428322606_1_ED614B15-7220-4
14
```



Note: If the selected indexer did not receive data, frozen buckets will not be present.

Post-Test Cleanup

Removing Test Data

To remove the data indexed during the test, complete the following steps:

1. Stop the Splunk service on the admin node. As the splunk user on admin1, issue the command:

```
$SPLUNK_HOME/bin/splunk stop
```

2. Stop all indexers:

```
clush --group=indexers $SPLUNK_HOME/bin/splunk stop
```

3. SSH to each indexer (idx1-idx8). As the splunk user, issue the command

```
$SPLUNK_HOME/bin/splunk clean eventdata -index main
```



Note: Alternatively, the clush command could be used to delete the indexes from all the peers at once by applying the force parameter ‘-f’. that is, `clush --group=indexers $SPLUNK_HOME/bin/splunk clean eventdata -index main -f`. Use this command with **extreme caution** as this action can’t be undone.

4. Confirm the cleaning of events ***on every indexer!!***

5. Start the master node (admin1) as splunk user:

```
$SPLUNK_HOME/bin/splunk start
```

6. Start indexing peers as user ‘splunk’

```
clush --group=indexers $SPLUNK_HOME/bin/splunk start
```

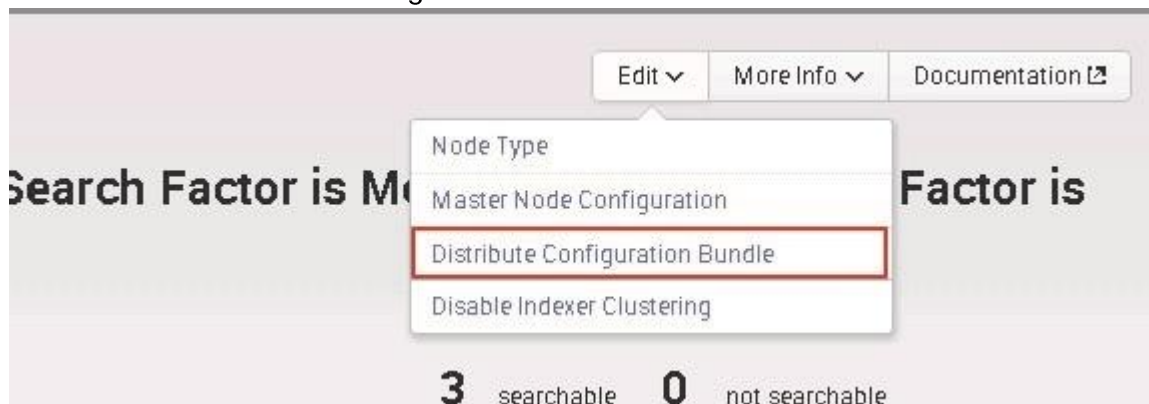

Removing Test Indexes

1. As user `splunk`, SSH into the master node (`admin1`)
2. Navigate to `$SPLUNK_HOME/etc/master-apps/_cluster/local/`
3. Remove the `indexes.conf` file. This file contains the 'archival' index and is not needed beyond testing.

```
[splunk@admin1 local]$ pwd
/data/disk1/splunk/etc/master-apps/_cluster/local
[splunk@admin1 local]$ ls -la
total 20
drwxr-xr-x 2 splunk splunk 4096 Apr 18 12:57 .
drwxr-xr-x 4 splunk splunk 4096 Apr 9 10:33 ..
-rw-rw-r-- 1 splunk splunk 9 Apr 18 12:57 indexes.conf
-rw-rw-r-- 1 splunk splunk 41 Apr 18 12:10 inputs.conf
-r--r--r-- 1 splunk splunk 231 Feb 18 15:01 README
[splunk@admin1 local]$ rm indexes.conf
[splunk@admin1 local]$ ls -la
total 16
drwxr-xr-x 2 splunk splunk 4096 Apr 18 12:58 .
drwxr-xr-x 4 splunk splunk 4096 Apr 9 10:33 ..
-rw-rw-r-- 1 splunk splunk 41 Apr 18 12:10 inputs.conf
-r--r--r-- 1 splunk splunk 231 Feb 18 15:01 README
[splunk@admin1 local]$
```

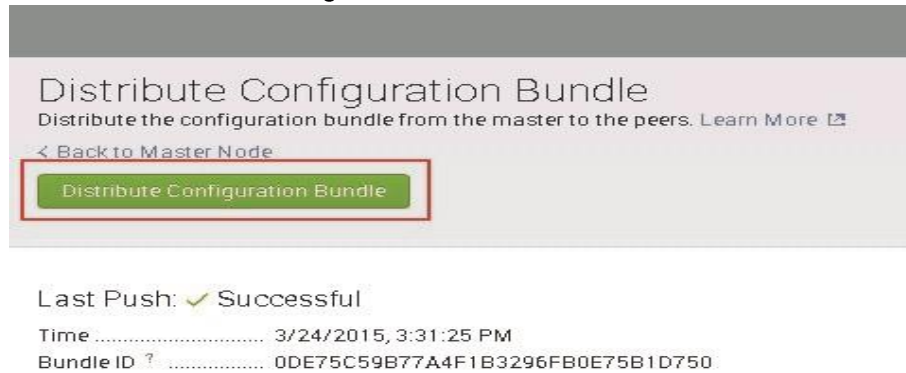
4. Using your browser, navigate to the master node web interface (`admin1`).
5. Select `Settings` → `Indexer Clustering` → `Edit` → `Distribute Configuration Bundle`.
Figure 179

Figure 179 Select Distribute Configuration Bundle



6. Click `Distribute Configuration Bundle`. (Figure 180)

Figure 180 Distribute Configuration Bundle from Master to Peer



7. Click `Push Changes`. This will remove the `indexes.conf` file which was created for testing purposes.

Removing the Universal Forwarder

1. Navigate to the host system of the Universal Forwarder
2. Stop the Universal Forwarder (`$SPLUNK_HOME/bin/splunk stop`)
3. Remove or uninstall the Universal Forwarder (actions will vary per OS)

Remove Deployment Server App

1. Via the CLI of the Deployment Server, navigate to `$SPLUNK_HOME/etc/deployment-apps`
2. Remove the directory 'outputTest'.

```
rm -r outputTest/
```

3. Reload the deployment Server.

```
$SPLUNK_HOME/bin/splunk reload deploy-server
```

4. Navigate to the URL of the Deployment Server.
5. Select `Settings > Forwarder Management`.
6. Under the tab `Server Classes`, click `Edit` → `Delete`.

The Splunk Enterprise installation is now in a 'clean' state, with no test data or forwarders.

Hardening the Splunk Installation

Taking the proper steps to secure Splunk Enterprise reduces its attack surface and mitigates the risk and impact of most vulnerabilities. We highly recommend you harden and tune the environment to your standards, so long as they do not overwrite configurations described within this document.

Turn Off Web Servers

Web connectivity should be limited to only those instances that require it. Web services should run on:

- Search Heads
- Distributed Management Console
- License Master

Web servers are not required to run on:

- Indexers

To disable web servers.

1. SSH into the instance of each indexer
2. As the `splunk` user, issue the command ``${SPLUNK_HOME}/bin/splunk disable webserver -auth <username>:<password>'`.`
3. SSH into the master node (admin1)
4. Restart the indexing tier with the command:

```
`${SPLUNK_HOME}/bin/splunk rolling-restart cluster-peers
```

```
[splunk@admin1 local]$ `${SPLUNK_HOME}/bin/splunk rolling-restart cluster-peers
```

For up-to-date information regarding hardening the Splunk environment, visit [‘Securing Splunk Enterprise](#)

Best Practices for Onboarding Data

The Universal Forwarder

There are several methods to consider for collecting data into Splunk, otherwise referred to as ‘onboarding’. Techniques include syslog forwarding, remote-polling techniques such as SNMP and WMI, writing of application logs to shared storage; batch uploads, and dedicated agents. Each of these approaches comes with limitations, and in many cases, with additional costs for licensing or for computing overhead.

Optimal collection of data combines several factors: it has minimal overhead, supports myriad data sources including data that is not in log files, is securely transmitted, works with low bandwidth, sends in real time, and has scalable, and robust delivery support. There should be centralized management of what is being collected.

To meet these goals, Splunk recommends the use of the Splunk Universal Forwarder (UF) on every server where this is possible. The Universal Forwarder is a centrally managed, lightweight tool for collecting and forwarding data to your indexers, and it is available for installation on nearly all standard operating systems: Linux, Windows, OSX AIX, HP-UX, Solaris, FreeBSD, even Raspberry Pi.

Advantages of the Splunk Universal Forwarder

The use of the Universal Forwarder allows a platform-agnostic approach to managing data collection from your environment.

Here is how the Splunk UF meets each of the goals:

Minimal Overhead: The Splunk Universal Forwarder is a lightweight software package; its sole purpose is to collect and forward data. Unlike heavyweight agents, it does not analyze data locally for lowest local overhead.

Data sources: Like most other options, the UF can collect data from local syslog files (*NIX) and Event Logs (Windows). The Splunk UF can also read from virtually *any* local file source so long as it is in ASCII format. The UF can also collect data that does not exist on disk at all:

- For all data being forwarded, the UF provides metadata for all data sources, including: hostname and time zone (per OS), source (typically the full file path), sourcetype, and routing information of destination indexes in Splunk.
- Each Universal Forwarder can call shell, Python, or PowerShell scripts to monitor OS-level and application-level usage; one example is to monitor and report on open network ports.
- Splunk Stream Forwarder can be configured to listen on network interfaces and collect protocol level data directly off the network stream. This is particularly useful when application logs lack the details necessary for your monitoring or analytics needs.
- The Splunk UF can listen on UDP or TCP ports directly, allowing applications to send application logs directly and avoid disk I/O concerns. The UF routes this directly to the indexers, removing the need to have compression / routing logic at the application layer.

Secure, low bandwidth: After collecting the raw data, Splunk uses data compression and optional SSL compression when sending the data to the Splunk indexers. SSL overhead is minimized by keeping TCP sessions open for set periods of time.

Real time: With the UF, Splunk can monitor and analyze data in near real time. As events are generated (for example, appended to a log file), they are immediately forwarded to indexers, where they are typically available for analysis within a second or so of initial generation.

Data Delivery: The UF is designed with high availability and guaranteed delivery in mind. Delivery is over TCP rather than UDP, ensuring that the UF “knows” if the data was received or not. Every UF can be configured with one or more indexers as targets, automatically spreading the load of collected data across the indexers. When one or more indexers are off-line, the UF will automatically find an indexer that is available. If all indexers are unavailable, The UF keeps track of last data sent – when an indexer becomes available, data transmission continues from where it left off.

Management: Splunk offers central management of the configuration of Universal Forwarders. Each UF connects to the Deployment server on a scheduled basis to check for new configurations. The Deployment server offers granular control over classes of systems that will collect any given data source. A Splunk administrator can change collection configurations and roll this out within minutes.

What About Systems Where the Splunk Universal Forwarder is Not Supported?

Networking and Storage gear, virtual appliances, and other “non-OS” devices are a vital part of any company’s environment, and should be monitored for performance and reliability. When the Splunk Universal Forwarder cannot be installed locally, here are a few recommended options to consider.

IPFIX/NetFlow: Most networking equipment (physical or virtual) supports either IPFIX, NetFlow, sflow or jflow for pushing out performance or security data. Systems sending on these protocols are called “exporters”. IPFIX and *Flow are binary protocols and cannot be sent directly to Splunk. Recommended approaches (select one):

- Have the exporters send their data to a system running the Splunk Universal Forwarder with the NetFlow or IPFIX TAs. These TAs translate the protocol from binary to ASCII.
- Have the exporters send their data to a 3rd party NetFlow/IPFIX parser, such as the NetFlow Integrator by NetFlow Logic. These systems accept binary data in, convert the data to syslog, and send out over the network. Install a Splunk UF on the same system, listening for network data streamed out of the middleware.

SNMP (polling): SNMP provides a valuable method for remotely collecting information from devices without a “normal” OS, such as network switches and routers, and on hardware management ports of physical server hardware. Recommended approaches (select one):

- Set up a Splunk heavy forwarder with the SNMP modular input app. The app will poll SNMP data and store it directly in Splunk. Details are in the app’s documentation. (Simply install on the Splunk search head for smaller deployments.)
- On any system where a Splunk UF could be installed, use an SNMP polling agent to collect data as necessary, and output the results to a log file. The UF can then collect the output files in the same manner as any other log file. The SNMP polling agent might be a commercial tool for this purpose, or something as simple as the ‘snmpwalk’ command running from a script.

SNMP (traps): SNMP traps are sent on alert conditions, typically by network devices. Recommended approaches (select one):

- Set SNMP devices to send their traps to a system running Splunk Universal Forwarder and the Splunk for Stream app. Configure Stream to listen for the SNMP protocol, forwarding whichever SNMP data is required.
- Set SNMP devices to send their traps to a system capable of running an SNMP daemon and a Splunk UF. Configure the SNMP daemon to log traps to a file, configure the UF to read the logs.

Syslog Forwarding: Many devices, virtual appliances, and bare-metal hypervisors offer the ability to send critical information via Syslog. (Linux and UNIX family OSes do to – but those systems support UF installation.) Recommended approach:

- Configure a system that runs a supported syslog server to listen for syslog data. (“syslog-ng” and “rsyslog” are excellent free options for Linux systems.)
- Configure the log servers to store logs in host-specific folders.
- When possible, configure syslog senders to use TCP rather than UDP. This ensures that critical data will not be dropped.
- Install the Splunk Universal Forwarder or heavy forwarder on the system, and configure it to monitor the log files. Tell Splunk to extract the hostnames from the file paths. (Heavy forwarder is necessary for certain syslog streams, such as ESXi data.) Install additional TAs as recommended by documentation, depending on syslog data sources.
- Optionally, create two syslog collection systems – and put them behind a load balancer. Have the syslog sources send to the load balancer via TCP. This ensures that if a single syslog server is down, the data will still continue coming to Splunk in real time.

Proprietary APIs: There are a large number of computing infrastructure components that only provide the full set of information when polled through API calls. These include network, storage, power system controllers, and other devices. A few specific examples include VMware vCenter servers, NetApp OnTap filers, Checkpoint firewalls. Because these systems provide a piece of the overall infrastructure picture for performance and security, bringing this data into Splunk is important for many Splunk customers. There are many approaches available. Here is a recommended methodology for getting this data in:



Note: Do not use the “find more apps” function within the Splunk UI.

- Check on splunkbase.splunk.com for an app that is designed to handle the technology. For example, search for “cisco”.
- If an app exists – read the documentation for that app.
- If an app does not exist on Splunkbase, simply perform an internet search for “Splunk” and the technology.
- If all else fails, contact Splunk support to ask for suggestions.

Additional Terminology

When onboarding data, Splunk provides a number of apps and add-ons via splunkbase.splunk.com. It is imperative that the Splunk administrator is familiar with the following terms:

Apps: An application that runs on Splunk Enterprise and typically addresses several use cases. An app typically contains both components of a Technology add-on and a Search add-on. An app contains one or more views. An app can include various Splunk Enterprise knowledge objects such as reports, lookups, scripted inputs, and modular inputs. An app sometimes depends on one or more add-ons for specific functionality. Examples of apps are the Splunk Enterprise Search app, the Splunk on Splunk app, and the Splunk Enterprise Security app.

Technology Add-on (TA): A technology add-on is a Splunk app that extracts knowledge from IT data so that it can be processed by Splunk, as well as other apps that leverage the Common Information Model (CIM). The technology add-on may pull data into Splunk or simply map data that is already coming in. Technology add-ons may conflict with or duplicate other Splunk apps that are already pulling in the same sort of data if they disagree on the source type. The difference between a technology add-on and another Splunk app is compliance with the Common Information Model. Technology add-ons typically reside on the universal forwarder or on the indexing tier.

Search Add-on (SA): A search add-on is a Splunk app that contains pre-built dashboards, searches, look-ups, forms, and various search components. The difference between a search add-on and a technology add-on is that SAs are primarily focused on visualizing data. Search add-ons exist on the search head(s).

Common Information Model (CIM): The Common Information Model Add-on is based on the idea that you can break down most log files into two components:

- fields
- event category tags

With these two components a knowledge manager can set up their log files in a way that makes them easy to process by Splunk and which normalizes noncompliant log files and forces them to follow a similar schema. The Common Information Model details the standard fields and event category tags that Splunk uses when it processes most IT data.

The Common Information Model is an overlay function and does not normalize or overwrite the raw data, it categorizes various fields into corresponding categories.

Recommended Apps and Add-ons for Data Collection

Here are the most commonly deployed add-ons, and what they collect.

You will find these add-ons at <https://splunkbase.splunk.com>. For each add-on, you will also see a complete description as well as the documentation on how to install them. Add-ons are installed to the forwarders, indexers, or both.

Splunk Technical Add-on for Cisco UCS: Splunk's first (and only) supported integration for server environments provides real-time operational visibility across multiple Cisco UCS domains and enables our joint customers to identify & resolve problems faster, proactively monitor systems & infrastructure, track key performance indicators & understand trends & patterns of activity & behavior.

The app grabs UCS faults, events, performance statistics such as temperature, power and network throughput from one or more Cisco UCS Managers to:

- Deliver real time and historical visibility centrally across your entire UCS deployment
- Provide analytics such as available capacity, trending of faults over time, tracking of power, and cooling costs.
- Correlate UCS performance, fault and events data with user, application, and hypervisor data to analyze, prevent, and fix problems across broad infrastructure or application environments.

Splunk Add-on for Unix and Linux: This add-on includes predefined inputs to collect data from *NIX systems, and maps to normalize the data to the Common Information Model. It provides easy collection from standard system log directories (such as /var/log), and excludes collection of common binary files. Examples are provided for monitoring the contents of specific files, such as /etc/hosts. Scripted inputs are included to monitor a variety of OS performance and network data points.

Splunk App for Stream: This provides a scalable and easy-to-configure solution to capture real-time streaming wire data from anywhere in your datacenter through protocol-level inspection. Stream data is useful for IT Ops, DevOps, and Security use cases. The Stream forwarder can run directly on endpoint servers – no need for SPAN/TAP ports; this is particularly useful in public cloud environments where SPAN/TAP are not an option. Capture only the relevant wire data for analytics, through filters and aggregation rules. Manage wire data volumes with fine-grained precision by selecting or deselecting protocols and associated attributes within the App interface.

Splunk DB Connect 2: Enrich your data results by accessing the data stored in your database servers. Splunk can access your structured data on-demand, for providing supplemental information, or on a monitoring basis where Splunk indexes the new data in selected tables. Use the Outputs function to export Splunk results into your legacy database.

Splunk Support for Active Directory / Idapsearch: Enrich your data results by reading data stored in your LDAP directory servers, including Active Directory. Use cases include mapping host names to additional information, mapping user names to HR information, or accessing asset management information stored in LDAP.

Splunk add-on for Microsoft Windows: This add-on includes predefined inputs to collect data from Windows systems, and maps to normalize the data to the Common Information Model. Supported data includes performance data, event logs, commonly used log files, and Windows Registry content. Scripted inputs are included to monitor open Network ports and installed applications.

Splunk App for Windows Infrastructure: The Splunk App for Windows Infrastructure provides examples of pre-built data inputs, searches, reports, and dashboards for Windows server and desktop management. You can monitor, manage, and troubleshoot Windows operating systems, including Active Directory elements, all from one place. The App also contains dashboards needed to monitor your Active Directory environment and allows for correlation opportunities from the Active Directory data back to the Operating System.

For a complete list of all Splunk supported apps go to:

<https://splunkbase.splunk.com/apps/#/order/latest/author/splunk>.

Conclusion

Splunk Enterprise delivers operational visibility and digital intelligence by monitoring all machine generated data and making it accessible, usable, and valuable across the organization. Cisco UCS Integrated Infrastructure for Big Data with its compute, storage, connectivity, and unified management features, streamlines the deployment and offers dependable, scalable integrated infrastructure that delivers predictable performance and high-availability for your Splunk Enterprise platform with a lower TCO.

The configuration detailed in the document can be extended to clusters of various sizes depending on application demands.

- Up to 80 servers (4 racks) can be supported with second generation Fabric Interconnects (6200 series).
- Up to 30 servers (2 racks) can be supported with third generation Fabric Interconnects (6300 series) with high speed 40 Gigabit Ethernet.
- Scaling beyond the above configurations can be implemented by interconnecting multiple UCS domains using Nexus 6000/7000 Series switches, scalable to thousands of servers and to hundreds of petabytes storage, and managed from a single pane using [UCS Central](#).

Bill of Materials

Splunk High Capacity and Performance Reference Configuration with Third-Generation Fabric Interconnects

Table 9 provides the bill of materials (BOM) for a 13 node Splunk Enterprise cluster with third-generation Fabric Interconnects. Table 9

Table 9 Bill of Materials for High Capacity and Performance Configuration with 3rd-Gen FIs

Part Number	Description	Quantity
Cisco UCS C220 M4 Rack Server configuration for Splunk search heads		
UCSC-C220-M4S	UCS C220 M4 SFF w/o CPU, mem, HD, PCIe, PSU, rail kit	3
CON-OSP-C220M4S	SNTC-24X7X4OS UCS C220 M4 SFF w/o CPU, mem, HD	3
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 M4 and C240 M4 rack servers	3
UCSC-PSU1-770W	770W AC Hot-Plug Power Supply for 1U C-Series Rack Server	6
UCSC-SCCBL220	Supercap cable 950mm	3
N20-BBLKD	UCS 2.5 inch HDD blanking panel	18
UCSC-HS-C220M4	Heat sink for UCS C220 M4 rack servers	6
UCSC-MRAID12G	Cisco 12G SAS Modular Raid Controller	3
UCSC-MRAID12G-2GB	Cisco 12Gbps SAS 2GB FBWC Cache module (Raid 0/1/5/6)	3
C1UCS-OPT-OUT	Cisco ONE Data Center Compute Opt Out Option	3
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	6
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	24
UCS-HD600G10K12G	600GB 12G SAS 10K RPM SFF HDD	6
UCS-M4-V4-LBL	Cisco M4 - v4 CPU asset tab ID label (Auto-Expand)	3
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	3

Part Number	Description	Quantity
CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	6
Cisco UCS C220 M4 Rack Server configuration for Splunk admin nodes		
UCSC-C220-M4S	UCS C220 M4 SFF w/o CPU, mem, HD, PCIe, PSU, rail kit	3
CON-OSP-C220M4S	SNTC-24X7X40S UCS C220 M4 SFF w/o CPU, mem, HD	3
UCSC-HS-C220M4	Heat sink for UCS C220 M4 rack servers	6
UCSC-MRAID12G	Cisco 12G SAS Modular Raid Controller	3
UCSC-MRAID12G-2GB	Cisco 12Gbps SAS 2GB FBWC Cache module (Raid 0/1/5/6)	3
UCSC-SCCBL220	Supercap cable 950mm	3
N20-BBLKD	UCS 2.5 inch HDD blanking panel	18
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 M4 and C240 M4 rack servers	3
UCSC-PSU1-770W	770W AC Hot-Plug Power Supply for 1U C-Series Rack Server	6
C1UCS-OPT-OUT	Cisco ONE Data Center Compute Opt Out Option	3
UCS-CPU-E52620E	2.10 GHz E5-2620 v4/85W 8C/20MB Cache/DDR4 2133MHz	6
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	12
UCS-HD600G10K12G	600GB 12G SAS 10K RPM SFF HDD	6
UCS-M4-V4-LBL	Cisco M4 - v4 CPU asset tab ID label (Auto-Expand)	3
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	3
CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	6
Rack		
RACK-UCS2	Cisco R42610 standard rack, w/side panels	1
CON-SNT-R42610	SNTC-8X5XNBD Cisco R42610 expansion rack, no side pan	1
RP208-30-1P-U-2	Cisco RP208-30-U-2 Single Phase PDU 20x C13, 4x C19	2
CON-SNT-RPDUX	SNTC-8X5XNBD Cisco RP208-30-U-X Single Phase PDU	2

Part Number	Description	Quantity
Cisco UCS S3260 Storage Server configuration for Splunk indexers		
UCSC-C3260	Cisco UCS C3260 Base Chassis w/4x PSU, SSD, Railkit	4
CON-OSP-C3260BSE	SNTC 24X7X4OS, Cisco UCS C3260 Base Chassis w/4x PSU	4
UCSC-C3160-BEZEL	Cisco UCS C3160 System Bezel	4
UCSC-C3X60-RAIL	UCS C3X60 Rack Rails Kit	4
UCSC-PSU1-1050W	UCS C3X60 1050W Power Supply Unit	16
UCSC-C3K-M4SRB	UCS C3000 M4 Server Node for Intel E5-2600 v4	4
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	32
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	4
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	8
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	4
UCS-C3X60-G2SD12	UCSC C3X60 120GB Boot SSD (Gen 2)	16
UCSC-C3X60-10TB	UCSC C3X60 10TB 4Kn for Top-Load	48
UCS-C3K-3XTSSD16	Cisco UCS C3000 Top Load 3X 1.6TB SSD	64
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	8
UCSC-C3K-M4SRB	UCS C3000 M4 Server Node for Intel E5-2600 v4	4
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	8
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	32
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	4
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	8
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	4
UCS-C3K-28HD10	UCS C3X60 2 row of 10TB NL-SAS drives (28 Total) 280TB	4
UCSC-C3X60-10TB	UCSC C3X60 10TB 4Kn for Top-Load	112
CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	16

Part Number	Description	Quantity
RHEL-2S2V-3A=	Red Hat Enterprise Linux (1-2 CPU,1-2 VN); 3-Yr Support Req	13
CON-ISV1-EL2S2V3A	ISV 24X7 RHEL Server 2Socket-OR-2Virtual; ANNUAL List Price	13
UCS-RHEL-TERMS	Acceptance of Terms, Standalone RHEL License for UCS Servers	13
3rd Generation Fabric Interconnects		
UCS-FI-6332	UCS 6332 IRU Fabric Interconnect/No PSU/32 QSFP+ports/8p Lic	2
N10-MGT014	UCS Manager v3.1	2
UCS-ACC-6332	UCS 6332 Chassis Accessory Kit	2
UCS-FAN-6332	UCS 6332 Fan Module	8
UCS-PSU-6332-AC	UCS 6332 Power Supply/100-240VAC	4
CAB-C13-C14-2M	Power Cord Jumper, C13-C14 Connectors, 2 Meter Length	4
QSFP-H40G-AOC5M	40GBASE Active Optical Cable, 5m	30
UCS-LIC-6300-40GC=	3rd Gen FI Per port License to connect C-direct only	14

Splunk High Capacity and Performance Reference Configuration with Second-Generation Fabric Interconnects

Table 10 provides the bill of materials (BOM) for a 13 node Splunk Enterprise cluster with third-generation Fabric Interconnects. Table 9

Table 10 Bill of Materials for High Capacity and Performance Configuration with 3rd-Gen FIs

Part Number	Description	Quantity
Cisco UCS C220 M4 Rack Server configuration for Splunk search heads		
UCSC-C220-M4S	UCS C220 M4 SFF w/o CPU, mem, HD, PCIe, PSU, rail kit	3
CON-OSP-C220M4S	SN7C-24X7X4OS UCS C220 M4 SFF w/o CPU, mem, HD	3
UCSC-MLOM-CSC-02	Cisco UCS VIC1227 VIC MLOM - Dual Port 10Gb SFP+	3
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 M4 and C240 M4 rack servers	3
UCSC-PSU1-770W	770W AC Hot-Plug Power Supply for 1U C-Series Rack Server	6

Part Number	Description	Quantity
CAB-N5K6A-NA	Power Cord, 200/240V 6A North America	6
UCSC-SCCBL220	Supercap cable 950mm	3
N20-BBLKD	UCS 2.5 inch HDD blanking panel	18
UCSC-HS-C220M4	Heat sink for UCS C220 M4 rack servers	6
UCSC-MRAID12G	Cisco 12G SAS Modular Raid Controller	3
UCSC-MRAID12G-2GB	Cisco 12Gbps SAS 2GB FBWC Cache module (Raid 0/1/5/6)	3
C1UCS-OPT-OUT	Cisco ONE Data Center Compute Opt Out Option	3
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	6
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	24
UCS-HD600G10K12G	600GB 12G SAS 10K RPM SFF HDD	6
UCS-M4-V4-LBL	Cisco M4 - v4 CPU asset tab ID label (Auto-Expand)	3
Cisco UCS C220 M4 Rack Server configuration for Splunk admin nodes		
UCSC-C220-M4S	UCS C220 M4 SFF w/o CPU, mem, HD, PCIe, PSU, rail kit	2
CON-OSP-C220M4S	SNTC-24X7X40S UCS C220 M4 SFF w/o CPU, mem, HD	2
UCSC-HS-C220M4	Heat sink for UCS C220 M4 rack servers	4
UCSC-MRAID12G	Cisco 12G SAS Modular Raid Controller	2
UCSC-MRAID12G-2GB	Cisco 12Gbps SAS 2GB FBWC Cache module (Raid 0/1/5/6)	2
CAB-N5K6A-NA	Power Cord, 200/240V 6A North America	4
UCSC-SCCBL220	Supercap cable 950mm	2
N20-BBLKD	UCS 2.5 inch HDD blanking panel	12
UCSC-MLOM-CSC-02	Cisco UCS VIC1227 VIC MLOM - Dual Port 10Gb SFP+	2
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 M4 and C240 M4 rack servers	2
UCSC-PSU1-770W	770W AC Hot-Plug Power Supply for 1U C-Series Rack Server	4

Part Number	Description	Quantity
C1UCS-OPT-OUT	Cisco ONE Data Center Compute Opt Out Option	2
UCS-CPU-E52620E	2.10 GHz E5-2620 v4/85W 8C/20MB Cache/DDR4 2133MHz	4
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	8
UCS-HD600G10K12G	600GB 12G SAS 10K RPM SFF HDD	4
UCS-M4-V4-LBL	Cisco M4 - v4 CPU asset tab ID label (Auto-Expand)	2
Rack		
RACK-UCS2	Cisco R42610 standard rack, w/side panels	1
CON-SNT-R42610	SNTC-8X5XNBD Cisco R42610 expansion rack, no side pan	1
RP208-30-1P-U-2	Cisco RP208-30-U-2 Single Phase PDU 20x C13, 4x C19	2
CON-SNT-RPDUX	SNTC-8X5XNBD Cisco RP208-30-U-X Single Phase PDU	2
CVR-QSFP-SFP10G=	QSFP to SFP10G adapter	16
Cisco UCS S3260 Storage Server configuration for Splunk indexers		
UCSC-C3260	Cisco UCS C3260 Base Chassis w/4x PSU, SSD, Railkit	4
CON-OSP-C3260BSE	SNTC 24X7X4OS, Cisco UCS C3260 Base Chassis w/4x PSU	4
CAB-N5K6A-NA	Power Cord, 200/240V 6A North America	16
UCSC-C3160-BEZEL	Cisco UCS C3160 System Bezel	4
UCSC-C3X60-RAIL	UCS C3X60 Rack Rails Kit	4
UCSC-PSU1-1050W	UCS C3X60 1050W Power Supply Unit	16
UCSC-C3K-M4SRB	UCS C3000 M4 Server Node for Intel E5-2600 v4	4
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	32
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	4
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	8
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	4

Part Number	Description	Quantity
UCS-C3X60-G2SD12	UCSC C3X60 120GB Boot SSD (Gen 2)	16
UCSC-C3X60-10TB	UCSC C3X60 10TB 4Kn for Top-Load	48
UCS-C3K-3XTSSD16	Cisco UCS C3000 Top Load 3X 1.6TB SSD	64
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	8
UCSC-C3K-M4SRB	UCS C3000 M4 Server Node for Intel E5-2600 v4	4
UCS-CPU-E52680E	2.40 GHz E5-2680 v4/120W 14C/35MB Cache/DDR4 2400MHz	8
UCS-MR-1X322RV-A	32GB DDR4-2400-MHz RDIMM/PC4-19200/dual rank/x4/1.2v	32
UCS-C3K-M4RAID	Cisco UCS C3000 RAID Controller M4 Server w 4G RAID Cache	4
UCSC-HS-C3X60	Cisco UCS C3X60 Server Node CPU Heatsink	8
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	4
UCS-C3K-28HD10	UCS C3X60 2 row of 10TB NL-SAS drives (28 Total) 280TB	4
UCSC-C3X60-10TB	UCSC C3X60 10TB 4Kn for Top-Load	112
RHEL-2S2V-3A=	Red Hat Enterprise Linux (1-2 CPU,1-2 VN); 3-Yr Support Req	13
CON-ISV1-EL2S2V3A	ISV 24X7 RHEL Server 2Socket-OR-2Virtual; ANNUAL List Price	13
UCS-RHEL-TERMS	Acceptance of Terms, Standalone RHEL License for UCS Servers	13
3rd Generation Fabric Interconnects		
UCS-FI-6296UP-UPG	UCS 6296UP 2RU Fabric Int/No PSU/48 UP/ 18p LIC	2
UCS-ACC-6296UP	UCS 6296UP Chassis Accessory Kit	2
N10-MGT014	UCS Manager v3.1	2
UCS-BLKE-6200	UCS 6200 Series Expansion Module Blank	6
UCS-FAN-6296UP	UCS 6296UP Fan Module	8
UCS-PSU-6296UP-AC	UCS 6296UP Power Supply/100-240VAC	4
CAB-N5K6A-NA	Power Cord, 200/240V 6A North America	4
SFP-H10GB-CU5M	10GBASE-CU SFP+ Cable 5 Meter	30

About the Authors

Karthik Karupasamy, Technical Marketing Engineer, Data Center Solutions Group (Cisco Systems)

Karthik Karupasamy is a Technical Marketing Engineer in Data Center Solutions Group at Cisco Systems. His main focus areas are architecture, solutions and emerging trends in big data related technologies and infrastructure in the Data Center.

Wissam Ali-Ahmad, Senior Sales Engineer, Global Strategic Alliances Group, Splunk

Wissam Ali-Ahmad is a Senior Sales Engineer in the Global Strategic Alliances group of Splunk. He focuses on technology alignment and innovation between Splunk and partners, in particular partners in IT Operations and Infrastructure.

Acknowledgements

The authors acknowledge contributions of Edmund Tran (Cisco), Ted Wu (Cisco), Brian Wooden (Splunk), and Barbara Dixon (Cisco) in developing this document.

