Cisco UCS Integrated Infrastructure for Big Data with Splunk Enterprise
With Cluster Mode for High Availability and Optional Data Archival

Last Updated: June 8, 2015
About the Authors

Raghunath Nambiar, Distinguished Engineer, Data Center Business Group (Cisco Systems)
Raghunath Nambiar is a Distinguished Engineer at Cisco's Data Center Business Group. His current responsibilities include emerging technologies and big data strategy.

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.

Dennis Bourg, Senior Sales Engineer, Global Strategic Alliances Group, Splunk
Dennis Bourg 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.
About Cisco Validated Design (CVD) Program

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.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB’s public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: http://www.cisco.com/go/trademarks. Third-party trademarks mentioned 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. (1110R).

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2015 Cisco Systems, Inc. All rights reserved.
Acknowledgment

The authors acknowledge contributions of Karthik Kulkarni, Manan Trivedi, Friea Berg (Splunk), Michael Donnelly (Splunk), Simeon Yep (Splunk) and Sindhu Sudhir in developing this document.
Cisco UCS Integrated Infrastructure for Big Data with 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.

Introduction

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. Into the bargain, virtualization and cloud infrastructures introduce additional complexity and create 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 a 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, across application troubleshooting, IT infrastructure monitoring, security, business analytics, Internet of Things, and many others that are entirely innovated by our customers. As the operational analytics become increasingly critical to day-to-day decision-making and Splunk software deployments expand to terabytes of data, a high-performance, highly scalable infrastructure is critical to ensuring rapid and predictable delivery of insights. UCS’s ability to expand to thousands of servers supports the massive scalability that Splunk deployments support while delivering exceptional performance.

The Cisco Validated Design (CVD) for Splunk describes the architecture and deployment procedures for Splunk Enterprise on Distributed High Capacity reference architecture based on Cisco UCS Integrated Infrastructure for Big Data. See Distributed Splunk Reference Architecture Solution Brief. The configuration consists of eight (8) Cisco UCS C240 M4 rack servers as indexers, three (3) C220 M4 rack servers as search heads and two (2) C220 M4 rack servers to perform administrative functions along with one (1) archival node (Cisco UCS C3160 rack servers) for frozen data.
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, eight Splunk Indexers provide capacity to index up to 2 TB of data per day (without replication). This configuration can scale to index hundreds of terabytes of data every 24 hours, delivering real-time search results and analytics to multiple users across the globe.

Cisco UCS Integrated Infrastructure for Big Data with Splunk Enterprise

The Cisco UCS solution for Splunk Enterprise is based on Cisco UCS Integrated Infrastructure for Big Data, 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 6200 Series Fabric Interconnects

Cisco UCS 6200 Series Fabric Interconnects provide high-bandwidth, low-latency connectivity for servers, with integrated, unified management provided for all connected devices by Cisco UCS Manager. Deployed in redundant pairs, Cisco fabric interconnects offer the full active-active redundancy, performance, and exceptional scalability needed to support the large number of nodes that are typical in clusters serving big data applications. Cisco UCS Manager enables rapid and consistent server configuration using service profiles, automating ongoing system maintenance activities such as firmware updates across the entire cluster as a single operation. Cisco UCS Manager also offers advanced monitoring with options to raise alarms and send notifications about the health of the entire cluster.

Figure 1  Cisco UCS 6296UP 96-Port Fabric Interconnect

Cisco UCS C-Series Rack Mount Servers

Cisco UCS C-Series Rack Mount C220 M4 High-Density Rack servers (Small Form Factor Disk Drive Model) and Cisco UCS C240 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. Cisco UCS C-Series Rack-Mount Servers are based on Intel Xeon E5-2600 v3 product family and 12-Gbps SAS throughput, delivering significant performance and efficiency gains over the previous generation of servers. The servers use dual Intel Xeon processor E5-2600 v3 series CPUs and
support up to 768 GB of main memory (128 or 256 GB is typical for big data applications) and a range of disk drive and SSD options. 24 Small Form Factor (SFF) disk drives are supported in performance-optimized option and 12 Large Form Factor (LFF) disk drives are supported in capacity-optimized option, along with 4 Gigabit Ethernet LAN-on-motherboard (LOM) ports. Cisco UCS virtual interface cards 1227 (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.

Cisco UCS C3160 Rack Server

Cisco UCS C3160 Rack Server is an advanced, modular rack server with extremely high storage density. Based on the Intel Xeon processor E5-2600 v2 series, it offers up to 360 TB of local storage in a compact 4-rack-unit (4RU) form factor. Because all its hard-disk drives are individually hot-swappable, and with its built-in enterprise-class Redundant Array of Independent Disks (RAID) redundancy, the Cisco UCS C3160 helps you achieve the highest levels of data availability. The Cisco UCS C3160 is ideal for Snapshots, active archiving, compliance, media storage, and distributed file systems for scenarios in which high storage capacity is important. Cisco UCS virtual interface cards 1227 (VICs) designed for the M4 generation of Cisco UCS C-Series Rack Servers and C3160 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.
Cisco UCS Virtual Interface Cards (VICs)

Cisco UCS Virtual Interface Cards (VICs), unique to Cisco, Cisco UCS Virtual Interface Cards incorporate next-generation converged network adapter (CNA) technology from Cisco, and offer dual 10-Gb/s ports designed for use with Cisco UCS C-Series Rack-Mount Servers. Optimized for virtualized networking, these cards deliver high performance and bandwidth utilization and support up to 256 virtual devices. The Cisco UCS Virtual Interface Card (VIC) 1227 is a dual-port, Enhanced Small Form-Factor Pluggable (SFP+), 10 Gigabit Ethernet and Fiber Channel over Ethernet (FCoE)-capable, PCI Express (PCIe) modular LAN on motherboard (mLOM) adapter. It is designed exclusively for the M4 generation of Cisco UCS C-Series Rack Servers and the C3160 dense storage servers.
Cisco UCS Manager

Cisco UCS Manager resides within the Cisco UCS 6200 Series Fabric Interconnects. 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), 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 that the process takes minutes instead of days. This simplification allows IT departments to shift their focus from constant maintenance to strategic business initiatives.
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’s also one of the most valuable, containing a definitive record of user transactions, customer behavior, sensor activity, machine behavior, security threats, fraudulent activity and more.

Splunk Enterprise provides a holistic way to organize and extract real-time insights from massive amounts of machine data from virtually any source. This includes data from websites, business applications, social media platforms, app servers, hypervisors, sensors, traditional databases and open source data stores. Splunk Enterprise scales to collect and index tens of terabytes of data per day, cross multi-geography, multi-datacenter and hybrid cloud infrastructures.

Key features of Splunk Enterprise

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

- Universal collection and indexing of machine data, from virtually any source
- Powerful search processing language (SPL) to search and analyze real-time and historical data
Solution Overview

This CVD describes architecture and deployment procedures for Splunk Enterprise using eight (8) Cisco UCS C240 M4 rack servers as indexers, three (3) Cisco UCS C220 M4 rack servers as search heads and 2 Cisco UCS C220 M4 rack servers to perform administrative functions along with 1 archival server for frozen data (Cisco UCS C3160 rack server). This architecture is based on the Cisco UCS Integrated Infrastructure for Big Data with Splunk. The reference architecture named as Distributed Deployment with High Capacity consist of 16 indexers for storage, of which 8 are considered for this CVD as well as an additional archival server for attached storage.

The solution goes into detail configuring distributed search on Splunk Enterprise platform along with the Archival node (Cisco UCS C3160):

Figure 6  Clustered Distributed Search deployment Architecture of Splunk Enterprise
**Table 1  Cisco UCS Reference Architecture for Splunk Enterprise (with Archival Nodes)**

<table>
<thead>
<tr>
<th>Role</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexer</td>
<td>8 Cisco UCS C240 M4 Rack Servers, each with:</td>
</tr>
<tr>
<td></td>
<td>2 Intel Xeon processor E5-2680 v3 CPUs (24 cores)</td>
</tr>
<tr>
<td></td>
<td>256 GB of memory</td>
</tr>
<tr>
<td></td>
<td>Cisco 12-Gbps SAS modular RAID controller with</td>
</tr>
<tr>
<td></td>
<td>2-GB flash-backed write cache</td>
</tr>
<tr>
<td></td>
<td>Cisco UCS VIC 1227</td>
</tr>
<tr>
<td></td>
<td>24 1.2-TB 10K SFF SAS drives in a RAID10 configuration</td>
</tr>
<tr>
<td></td>
<td>2 120-GB SSD for the operating system</td>
</tr>
<tr>
<td>Search head</td>
<td>3 Cisco UCS C220 M4 Rack Servers, each with:</td>
</tr>
<tr>
<td></td>
<td>2 Intel Xeon processor E5-2680 v3 CPUs (24 cores)</td>
</tr>
<tr>
<td></td>
<td>256 GB of memory</td>
</tr>
<tr>
<td></td>
<td>Cisco 12-Gbps SAS modular RAID controller with</td>
</tr>
<tr>
<td></td>
<td>2-GB flash-backed write cache</td>
</tr>
<tr>
<td></td>
<td>Cisco UCS VIC 1227</td>
</tr>
<tr>
<td></td>
<td>2 600-GB 10K SFF SAS drives</td>
</tr>
<tr>
<td>Administration and master nodes</td>
<td>3 Cisco UCS C220 M4 Rack Servers, each with:</td>
</tr>
<tr>
<td></td>
<td>2 Intel Xeon processor E5-2680 v3 CPUs (24 cores)</td>
</tr>
<tr>
<td></td>
<td>128 GB of memory</td>
</tr>
<tr>
<td></td>
<td>Cisco 12-Gbps SAS modular RAID controller with</td>
</tr>
<tr>
<td></td>
<td>2-GB flash-backed write cache</td>
</tr>
<tr>
<td></td>
<td>Cisco UCS VIC 1227</td>
</tr>
<tr>
<td></td>
<td>2 600-GB 10K SFF SAS drives</td>
</tr>
</tbody>
</table>
The rack consists of two vertical power distribution units (PDU) two Cisco UCS 6296UP Fabric Interconnects, eight Cisco UCS C240 M4 servers, five Cisco UCS C220 M4 servers and one Cisco UCS C3160 server. All the devices are connected to each of the vertical PDUs for redundancy; thereby ensuring availability during power source failure.

Note
Please contact your Cisco representative for country specific information.

Table 2 describes the rack configurations used in this CVD.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Rack Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cisco 42URack</strong></td>
<td><strong>Master Rack</strong></td>
</tr>
<tr>
<td>42</td>
<td>Cisco UCS FI 6296UP</td>
</tr>
<tr>
<td>41</td>
<td></td>
</tr>
<tr>
<td>Cisco 42URack</td>
<td>Master Rack</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>40</td>
<td>Cisco UCS FI 6296UP</td>
</tr>
<tr>
<td>39</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Cisco UCS C220 M4</td>
</tr>
<tr>
<td>37</td>
<td>Cisco UCS C220 M4</td>
</tr>
<tr>
<td>36</td>
<td>Unused</td>
</tr>
<tr>
<td>35</td>
<td>Cisco UCS C220 M4</td>
</tr>
<tr>
<td>34</td>
<td>Cisco UCS C220 M4</td>
</tr>
<tr>
<td>33</td>
<td>Cisco UCS C220 M4</td>
</tr>
<tr>
<td>32</td>
<td>Cisco UCS C240 M4</td>
</tr>
<tr>
<td>31</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Cisco UCS C240 M4</td>
</tr>
<tr>
<td>29</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Cisco UCS C240 M4</td>
</tr>
<tr>
<td>27</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Cisco UCS C240 M4</td>
</tr>
<tr>
<td>25</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Cisco UCS C240 M4</td>
</tr>
<tr>
<td>23</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Cisco UCS C240 M4</td>
</tr>
<tr>
<td>21</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Cisco UCS C240 M4</td>
</tr>
<tr>
<td>19</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Cisco UCS C240 M4</td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Unused</td>
</tr>
<tr>
<td>15</td>
<td>Unused</td>
</tr>
<tr>
<td>14</td>
<td>Unused</td>
</tr>
<tr>
<td>13</td>
<td>Unused</td>
</tr>
<tr>
<td>12</td>
<td>Unused</td>
</tr>
<tr>
<td>11</td>
<td>Unused</td>
</tr>
<tr>
<td>10</td>
<td>Unused</td>
</tr>
<tr>
<td>9</td>
<td>Unused</td>
</tr>
<tr>
<td>8</td>
<td>Unused</td>
</tr>
<tr>
<td>7</td>
<td>Unused</td>
</tr>
<tr>
<td>6</td>
<td>Unused</td>
</tr>
<tr>
<td>5</td>
<td>Unused</td>
</tr>
</tbody>
</table>
Port Configuration on Fabric Interconnects

The following table shows the network connectivity configurations used for developing this CVD.

<table>
<thead>
<tr>
<th>Port Type</th>
<th>Port Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>1</td>
</tr>
<tr>
<td>Cisco UCS C220 M4 Servers</td>
<td>4 to 8</td>
</tr>
<tr>
<td>Cisco UCS C240 M4 Servers</td>
<td>9 to 16</td>
</tr>
<tr>
<td>Cisco C3160 Server</td>
<td>17</td>
</tr>
</tbody>
</table>

Configuration and Cabling for C240 M4 Rack Servers

The C240 M4 rack server is equipped with Intel Xeon E5-2680 v3 processors, 256 GB of memory, Cisco UCS Virtual Interface Card 1227, Cisco 12-Gbps SAS Modular Raid Controller with 2-GB FBWC, 24 1.2-TB 10K SFF SAS drives, 2 120-GB SATA SSD for Boot.

All the eight servers of this category are all directly connected to the ports on the Cisco UCS FI6296 Fabric Interconnects as shown below. These ports are configured as server ports in the UCS Manager.

Figure 7 illustrates the port connectivity between the Fabric Interconnect and Cisco UCS C240 M4 server. Eight Cisco UCS C240 M4 servers are used as indexers in this rack configuration.
This solution makes use of five C220 M4 rack servers that are configured with two different classes of CPUs.

- The servers that function as the search heads are equipped with Intel Xeon E5-2680 v3 processors, 256 GB of memory, Cisco UCS Virtual Interface Card 1227, Cisco 12-Gbps SAS Modular Raid Controller with 2-GB FBWC, and 2 600 GB 10K SFF SAS drives.

- The servers that function as the admin nodes are equipped with Intel Xeon E5-2620 v3 processors, 128 GB of memory, Cisco UCS Virtual Interface Card 1227, Cisco 12-Gbps SAS Modular Raid Controller with 2-GB FBWC, and 2 600 GB 10K SFF SAS drives.

All the five servers of this category are all directly connected to the ports on the Cisco UCS FI6296 Fabric Interconnects as shown below. These ports are configured as server ports in the UCS Manager. **Figure 8** server illustrates the port connectivity between the Fabric Interconnect and Cisco UCS C220 M4 servers. Five Cisco UCS C220 M4 servers are used in rack configuration.
The C3160 rack server is equipped with 2 Intel Xeon E5-2695v2 processors, 256 GB of memory, 2 Cisco UCS Virtual Interface Card 1227, Cisco 12-Gbps SAS Modular Raid Controller with 4-GB FBWC, 60 4-TB 7.2K LFF SAS drives, 2 120-GB SATA SSD for Boot.

The servers of this category are all directly connected to the ports on the Cisco UCS FI6296 Fabric Interconnects as shown below. These ports are configured as appliance ports in the UCS Manager. Figure 9 illustrates the port connectivity between the Fabric Interconnect and Cisco UCS C3160 server as an Appliance port. One Cisco UCS C3160 server is used in master rack configurations.
Figure 9  Fabric Topology for Cisco UCS C3160 Rack Server

For more information on physical connectivity and single-wire management, see:

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

Rack Appearance

Figure 10 shows the single rack configuration containing five C220 M4 servers, eight Cisco UCS C240 M4 servers along with 1 Cisco UCS C3160 as an archival server. Each server is connected to each Fabric Interconnect by means of dedicated (i.e. directly) 10 Gigabit Ethernet link. The Cisco UCS C3160 connected to each Fabric Interconnect as an Appliance port by means of a dedicated 10 Gigabit Ethernet link. Individual server connectivity diagrams can be seen above.
Splunk Cluster Scalability

The Cisco UCS reference architectures for Splunk Enterprise support the massive scalability that Splunk deployments demand. The configuration described in this document can be extended to support up to 80 servers with a pair of 96-port fabric interconnects. Multiple Cisco UCS domains - up to thousands of servers - can be supported using Cisco Nexus 9000 or 7000 Series Switches.

The following tables are meant to provide high level sizing guideline for scaling the Splunk cluster, with the following assumptions:

- Indexer clustering is enabled
- Replication Factor is set to 2
• Search Factor is set to 2

Note
The Splunk indexer on a C240 M4 rack server is considered to be able to index about 250GB per day.

Table 4  High Level Splunk Enterprise Indexer Cluster and Archival Scalability

<table>
<thead>
<tr>
<th>Sample Retention</th>
<th>Server Type</th>
<th>Daily Indexing Volume (TB/day)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 months</td>
<td>Indexer (C240 M4 rack server)</td>
<td>1 2 3 4 8</td>
<td>← Indexing capacity</td>
</tr>
<tr>
<td>3 months</td>
<td>{Hot/Warm/Cold Data}</td>
<td>5 10 15 20 25</td>
<td>← Number of indexers (C240 M4 rack server)</td>
</tr>
<tr>
<td>6 months</td>
<td>14 28 42 56 112</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 months</td>
<td>Archival (C3160 rack server)</td>
<td>1 2 3 4 8</td>
<td>← Number of Archival nodes (C3160 rack server)</td>
</tr>
<tr>
<td>12 months</td>
<td>{Frozen Data}</td>
<td>2 4 6 8 16</td>
<td></td>
</tr>
</tbody>
</table>

Note
While scaling the cluster, it is recommended to maintain the ratio between the search heads and indexers at 1:8. This ratio is just a guideline. Depending on the actual deployment scenario, this ratio will be different.

Software Versions

The software versions tested and validated in this document are shown in table 5.

Table 5  Software Versions

<table>
<thead>
<tr>
<th>Layer</th>
<th>Component</th>
<th>Version or Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute</td>
<td>Cisco UCS C240 M4</td>
<td>C240M4.2.0.3d</td>
</tr>
<tr>
<td></td>
<td>Cisco UCS C220 M4</td>
<td>C220M4.2.0.3d</td>
</tr>
<tr>
<td></td>
<td>Cisco UCS C3160</td>
<td>C3160M3.2.0.2.*</td>
</tr>
<tr>
<td>Network</td>
<td>Cisco UCS 6296UP</td>
<td>UCS 2.2(3d)A</td>
</tr>
<tr>
<td></td>
<td>Cisco UCS VIC1227 Firmware</td>
<td>4.0(1d)</td>
</tr>
<tr>
<td></td>
<td>Cisco UCS VIC1227 Driver</td>
<td>2.1.1.66</td>
</tr>
<tr>
<td>Storage</td>
<td>LSI SAS 3108</td>
<td>24.5.0-0020</td>
</tr>
</tbody>
</table>
Fabric Configuration

Table 5  Software Versions

<table>
<thead>
<tr>
<th>Layer</th>
<th>Component</th>
<th>Version or Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>Red Hat Enterprise Linux Server</td>
<td>6.5 (x86_64)</td>
</tr>
<tr>
<td></td>
<td>Cisco UCS Manager</td>
<td>2.2(3d)</td>
</tr>
<tr>
<td></td>
<td>Splunk Enterprise</td>
<td>6.2.2</td>
</tr>
</tbody>
</table>

To learn more about Splunk Enterprise, see: 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 6.5 operating system.
• C240/C220 M4 Rack Servers are supported from UCS firmware 2.2(3d) onwards.

Fabric Configuration

This section provides details for configuring a fully redundant, highly available Cisco UCS 6296 fabric configuration.

1. Initial setup of the Fabric Interconnect A and B.
2. Connect to UCS Manager using virtual IP address of using the web browser.
3. Launch UCS Manager.
4. Enable server, uplink and appliance ports.
5. Start discovery process.
6. Create pools and polices for Service profile template.
7. Create Service Profile template and 64 Service profiles.
8. Associate Service Profiles to servers.

Performing Initial Setup of Cisco UCS 6296 Fabric Interconnects

This section describes the steps to perform initial setup of the Cisco UCS 6296 Fabric Interconnects A and B.

Configure Fabric Interconnect A

1. Connect to the console port on the first Cisco UCS 6296 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.
Fabric Configuration

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

Configure Fabric Interconnect B

1. Connect to the console port on the second Cisco UCS 6296 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 6200 Series Fabric Interconnect, see:

Logging Into Cisco UCS Manager

Follow these steps to login to Cisco UCS Manager.
1. Open a web browser and navigate to the Cisco UCS 6296 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.
Upgrading Cisco UCS Manager Software to Version 2.2(3d)

This document assumes the use of UCS 2.2(3d). Refer to Upgrading between Cisco UCS 2.0 Releases to upgrade the Cisco UCS Manager software and UCS 6296 Fabric Interconnect software to version 2.2(3d). Also, make sure the UCS C-Series version 2.2(3d) software bundles is installed on the Fabric Interconnects.

Adding Block of IP Addresses for KVM Access

These steps provide details for creating a block of KVM IP addresses for server access in the Cisco UCS environment.

1. Select the LAN tab at the top of the left window.
3. Right-click IP Pool ext-mgmt
4. Select Create Block of IPv4 Addresses.

5. Enter the starting IP address of the block and number of IPs needed, as well as the subnet and gateway information.
6. Click **OK** to create the IP block.
7. Click **OK** in the message box.

---

**Enabling Uplink Port**

These steps provide details for enabling uplinks 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 Unconfigured Ethernet Ports section.
4. Select port 1, that is connected to the uplink switch, right-click, then select **Reconfigure > Configure as Uplink Port**.
5. Select **Show Interface** and select 10GB for Uplink Connection.
6. A pop-up window appears 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 Unconfigured Ethernet Ports section.
9. Select port 1, that is connected to the uplink switch, right-click, then select Reconfigure > Configure as Uplink Port.

10. Select Show Interface and select 10GB for Uplink Connection.

11. A pop-up window appears to confirm your selection. Click Yes, then click OK to continue.

### Figure 14  Enabling Uplink Ports

#### Configuring VLANs

VLANs are configured as in shown in Table 6.

### Table 6  VLAN Configurations

<table>
<thead>
<tr>
<th>VLAN</th>
<th>Fabric</th>
<th>NIC Port</th>
<th>Function</th>
<th>Failover</th>
</tr>
</thead>
<tbody>
<tr>
<td>default(VLAN1)</td>
<td>A</td>
<td>eth0</td>
<td>Management, User connectivity</td>
<td>Fabric Failover to B</td>
</tr>
<tr>
<td>vlan11_DATA1</td>
<td>B</td>
<td>eth1</td>
<td>Hadoop</td>
<td>Fabric Failover to A</td>
</tr>
<tr>
<td>vlan12_DATA2</td>
<td>A</td>
<td>eth2</td>
<td>Hadoop with multiple NICs support</td>
<td>Fabric Failover to B</td>
</tr>
</tbody>
</table>

All of the VLANs created need to 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 vlan11_DATA1 is configured for Splunk data ingestion from the forwarders. The VLAN vlan12_DATA2 will be used for the replication traffic between the indexers. This enables Splunk to take advantage the UCS dual 10Gigabit Ethernet (10GigE) links to isolate the inter server traffic (i.e. index replication) from the ingress traffic (data ingestion from forwarders) on separate 10GigE links.

**Note**

We are using default VLAN1 for management traffic.

Follow these steps to configure the VLANs in the Cisco UCS Manager GUI:

1. Select the LAN tab in the left pane in the UCS Manager GUI.
2. Select LAN > VLANs.
3. Right-click the VLANs under the root organization.
4. Select Create VLANs to create the VLAN.
5. Enter vlan11_DATA1 for the VLAN Name.
6. Click the **Common/Global** radio button for the vlan11_DATA1.
7. Enter 11 on VLAN IDs of the Create VLAN IDs.
8. Click **OK** and then, click **Finish**.
9. Click **OK** in the success message box.
10. Select the LAN tab in the left pane again.
11. Select LAN > VLANs.
12. Right-click the VLANs under the root organization.
13. Select Create VLANs to create the VLAN.
14. Enter vlan12_DATA2 for the VLAN Name.
15. Click the Common/Global radio button for the vlan12_DATA2.
16. Enter 12 on VLAN IDs of the Create VLAN IDs.
17. Click OK and then, click Finish.
18. The below screenshot shows the created VLANs.

**Figure 17** Creating VLAN for Splunk Index Replication Data Traffic

**Figure 18** 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 Unconfigured Ethernet Ports section.
4. Select all the ports that are connected to the Servers right-click them, and select **Reconfigure** > **Configure as a Server Port**.
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 Unconfigured 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 to confirm your selection. Click **Yes** then **OK** to continue.

**Figure 19**  
Enabling Server Ports

10. Repeat this procedure to enable the server ports on Fabric Interconnect B.

Create VLAN for Appliance Port

These steps provide details for creating VLAN for Appliance port configuration.

1. Select the **LAN** tab in the left pane in the UCS Manager GUI.
2. Select **LAN** > **Appliances** > **VLANs**.
3. Right-click **VLANs** under the root organization.
4. Select **Create VLANs** to create the VLAN.
5. Enter vlan11_Appliance for the VLAN Name.
6. Click the **Common/Global** radio button.
7. Enter 11 for VLAN ID.

![Figure 20 Creating VLAN for Appliance Ports - Part 1](image1)

**Figure 20**  
Creating VLAN for Appliance Ports - Part 1

![Figure 21 Creating VLAN for Appliance Ports - Part 2](image2)

**Figure 21**  
Creating VLAN for Appliance Ports - Part 2

---

**Configuring Appliance Port**

These steps provide details for configuring Appliance 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 **Unconfigured Ethernet Ports** section.
4. Select port 2, right-click the port, and select **Reconfigure > Configure as an Appliance Port**.
5. A confirmation message box appears. Click **Yes**, then **OK** to continue.


7. Keep the Pin Group as **<not set>**.

8. Keep the Network Control Policy as Default.


10. Click the **10Gbps** radio button for the Admin Speed.

11. Click the **Trunk** radio button for the Port Mode.

12. Select **Default VLAN**, and click the **Native VLAN** radio button.
13. Select the **Equipment** tab on the top left of the window.

14. Select **Equipment > Fabric Interconnects > Fabric Interconnect B (Subordinate) > Fixed Module**.

15. Expand the Unconfigured Ethernet Ports section.

16. Select port 2, right-click the port, and select **Reconfigure > Configure as an Appliance Port**.
17. A confirmation message box appears. Click Yes, then OK to continue.
18. Select Platinum for the Priority.
19. Keep the Pin Group as <not set>.
20. Keep the Network Control Policy as Default.
22. Click the 10Gbps radio button for the Admin Speed.
23. Click the Trunk radio button for the Port Mode.
24. Select vlan11_Appliance, and click the Native VLAN radio button.
The following figure shows the list of all ports used in the fabric interconnects with different roles assigned.
Creating Pools for Service Profile Templates

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.

Follow these steps to configure an organization within the Cisco UCS Manager GUI:

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 MAC Address Pools

Follow these steps to create MAC address pools:

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. Enter ucs for the name of the MAC pool.
5. Enter the MAC Pool name, which is ucs.
6. (Optional) Enter a description of the MAC pool.
7. Select Assignment Order Sequential.
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.
12. Click OK.

Figure 27  Creating MAC Pool Window

![Creating MAC Pool Window](image1)

Figure 28  Specifying First MAC Address and Size

![Specifying First MAC Address and Size](image2)

13. Click Finish.
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 the Server Pools.
4. Select Create Server Pool.
5. Enter the Server Pool name, which is ucs.
6. (Optional) enter a description for the organization.
7. Click Next to add the servers.
8. Select all the Cisco UCS C240 M4 and all the five Cisco UCS C200 M4 servers to be added to the server pool and then Click >> to add them to the pool.

9. Click Finish.

10. 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, ROM and storage controller properties as applicable.

Follow these steps to create a firmware management policy for a given server configuration using the Cisco UCS Manager GUI:

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 the Host Firmware package name, which is ucs_FW_2.2_3e_C.
Creating Policies for Service Profile Templates

**Note**  Name the Host Firmware Package appropriately to include the actual firmware version and package info.

6. Click the **Simple** radio button to configure the Host Firmware package.
7. Select the appropriate Rack package that you have.
8. Click **OK** to complete creating the management firmware package.
9. Click **OK**.

**Figure 33 Creating Host Firmware Package**

---

Creating QoS Policies

Follow these steps to create the QoS policy for a given server configuration using the Cisco UCS Manager GUI:
Best Effort Policy

1. Select the LAN tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
4. Select Create QoS Policy.

5. Enter BestEffort as the name of the policy.
6. Select BestEffort from the 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.
Creating Policies for Service Profile Templates

Platinum Policy

1. Select the LAN tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
4. Select Create QoS Policy.
5. Enter Platinum as the name of the policy.
6. Select Platinum from the 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.
Creating Policies for Service Profile Templates

Setting Jumbo Frames

Follow these steps for setting up the Jumbo frames and enabling QoS:

1. Select the LAN tab in the left pane in the UCS Manager GUI.
2. Select LAN Cloud > QoS System Class.
3. In the right pane, select the General tab.
4. In the Platinum row, enter 9000 for MTU.
5. Check the Enabled Check box next to Platinum.
6. In the Best Effort row, select best-effort for weight.
7. In the Fiber Channel row, select none for weight.
8. Click Save Changes.
9. Click OK.
Creating Local Disk Configuration Policy

Follow these steps to create local disk configuration in the Cisco UCS Manager GUI:

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 the local disk configuration policy name, which is **ucs**.
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**.
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.

Follow these steps to create a server BIOS policy using the Cisco UCS Manager GUI:

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 the BIOS policy name, which is ucs.
6. Change the BIOS settings as per the following figures:

**Figure 39  Creating Server BIOS Policy**
Creating Policies for Service Profile Templates

**Figure 40** Creating Server BIOS Policy for Processor

![Unified Computing System Manager](image-url)

- Turbo Boost: disabled enabled Platform Default
- Enhanced Intel Speedstep: disabled enabled Platform Default
- Hyper Threading: disabled enabled Platform Default
- Core Multi Processing: all
- Execute Disabled Bit: disabled enabled Platform Default
- Virtualization Technology (VT): disabled enabled Platform Default
- Hardware Pre-fetcher: disabled enabled Platform Default
- Adjacent Cache Line Pre-fetcher: disabled enabled Platform Default
- DCU Streamer Pre-fetcher: disabled enabled Platform Default
- DCU JP Pre-fetcher: disabled enabled Platform Default
- Direct Cache Access: disabled enabled Platform Default
- Processor C State: disabled enabled Platform Default
- Processor C1E: disabled enabled Platform Default
- Processor C3 Report: disabled acpi-c2 acpi-c3 Platform Default
- Processor C6 Report: disabled enabled Platform Default
- Processor C7 Report: disabled enabled Platform Default
- CPU Performance: enterprise High-throughput HPC Platform Default
- Max Variable MTRR Settings: auto-max 8 Platform Default
- Local X2 APIC: xapic xsxapic auto Platform Default
- Power Technology: performance
- Energy Performance: performance
- Frequency Floor Overrides: disabled enabled Platform Default
- P-STATE Coordination: none small sw-any Platform Default
- DRAM Clock Throttling: performance
- Channel Interleaving: Platform Default
- Rank Interleaving: Platform Default
- Demand Scrub: disabled enabled Platform Default
- Patrol Scrub: disabled enabled Platform Default
7. Click **Finish** to complete creating the BIOS policy.

8. Click **OK**.
Creating Boot Policy

Follow these steps to create boot policies within the Cisco UCS Manager GUI:

1. Select the Servers tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click the Boot Policies.
4. Select Create Boot Policy.
5. Enter the boot policy name, which is ucs.
6. (Optional) enter a description for the boot policy.
7. Keep the Reboot on Boot Order Change check box unchecked.
8. Keep Enforce vNIC/vHBA/iSCSI Name check box checked.
10. Expand Local Devices > Add CD/DVD and select Add Local CD/DVD.
11. Expand Local Devices and select Add Local Disk.
12. Expand vNICs and select Add LAN Boot and enter eth0.
13. Click OK to add the Boot Policy.
14. Click OK.
Creating Power Control Policy

Follow these steps to create the Power Control policies within the Cisco UCS Manager GUI:

1. Select the Servers tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click the Power Control Policies.
4. Select Create Power Control Policy.
5. Enter the Power Control policy name, which is ucs.
6. (Optional) enter a description for the boot policy.
7. Select **No cap** for Power Capping selection.
8. Click **OK** to the Power Control Policy.
9. Click **OK**.
Creating Maintenance Policy

To create a maintenance policy, follow these steps:

1. Select the Servers tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click the Maintenance Policies.
4. Select Create Maintenance Policy.
5. Enter the Maintenance Policy name, which is user_ack.
6. Select User Ack as the Reboot Policy by clicking on it.
7. Click OK to create the Maintenance Policy.
8. Click OK.
Creating Service Profile Template

To create a service profile template, follow these 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.

To create a service profile template, follow these steps:

4. The Create Service Profile Template window appears. These steps below provide a detailed configuration procedure to identify the service profile template:
   a. Enter the service profile template name, which is ucs.
   b. Click the Updating Template radio button.
   c. In the UUID section, select Hardware Default as the UUID pool.
   d. Click Next to continue to the next section.
**Configuring Network Settings for the Template**

1. Keep the Dynamic vNIC Connection Policy field at the default.
2. Click the **Expert** radio button for the option, **how would you like to configure LAN connectivity?**
3. Click **Add** to add a vNIC to the template.
4. The Create vNIC window displays. Name the vNIC as eth0.
5. Select UCS in the Mac Address Assignment pool.
6. Click the Fabric A radio button and Check the Enable failover check box for the Fabric ID.
7. Check the default check box for VLANs and click the Native VLAN radio button.
8. Select MTU size as 1500
9. Select adapter policy as Linux
10. Select QoS Policy as BestEffort.
11. Keep the Network Control Policy as Default.
12. Keep the Connection Policies as Dynamic vNIC.
13. Keep the Dynamic vNIC Connection Policy as <not set>.
14. Click OK.
15. Click **Add** to add a vNIC to the template.

16. The Create vNIC window appears. Name the vNIC eth1.

17. Select ucs in the Mac Address Assignment pool.
18. Click the **Fabric B** radio button and Check the **Enable failover** check box for the Fabric ID.
19. Check the **vlan11_DATA1** check box for VLANs, and click the **Native VLAN** radio button.
20. Select MTU size as 9000.
21. Select adapter policy as Linux.
22. Select QoS Policy as Platinum.
23. Keep the Network Control Policy as Default.
24. Keep the Connection Policies as Dynamic vNIC.
25. Keep the Dynamic vNIC Connection Policy as <not set>.
26. Click **OK**.
27. Click **Add** to add a vNIC to the template.

28. The Create vNIC window appears. Name the vNIC eth2.

29. Select ucs in the Mac Address Assignment pool.

30. Click the **Fabric A** radio button, and then Check the **Enable failover** check box for the Fabric ID.
31. Check the **vlan12_DATA2** check box for VLANs, and then click the **Native VLAN** radio button.
32. Select MTU size as 9000.
33. Select adapter policy as Linux.
34. Select QoS Policy as Platinum.
35. Keep the Network Control Policy as Default.
36. Keep the Connection Policies as Dynamic vNIC.
37. Keep the Dynamic vNIC Connection Policy as <not set>.
38. Click **OK**.
Configuring Storage Policy for the Template

Follow these steps to configure storage policies:

1. Select ucs for the local disk configuration policy.
2. Click the **No vHBAs** radio button for the option, **How would you like to configure SAN connectivity?**

3. Click **Next** to continue to the next section.

---

4. Click **Next** once the zoning window appears to go to the next section.
Configuring vNIC/vHBA Placement for the Template

Follow these steps to configure vNIC/vHBA placement policy:

1. Select the Default Placement Policy option for the Select Placement field.
2. Select eth0, eth1 and eth2 assign the vNICs in the following order:
   a. eth0
   b. eth1
   c. eth2
3. Review to make sure that all of the vNICs were assigned in the appropriate order.
4. Click Next to continue to the next section.
Configuring vMedia Policy for the Template

1. Click **Next** once the vMedia Policy window appears to go to the next section.
Configuring Server Boot Order for the Template

Follow these steps to set the boot order for servers:

1. Select ucs in the Boot Policy name field.
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 next section.
In the Maintenance Policy window, follow these steps to apply the maintenance policy:

1. Keep the Maintenance policy at no policy used by default.
2. Click **Next** to continue to the next section.

**Configuring Maintenance Policy for the Template**

In the Maintenance Policy window, follow these steps to apply the maintenance policy:

1. Select the user_ack at the Maintenance Policy field.
2. Click **Next** to continue to the next section.
Configuring Server Assignment for the Template

In the Server Assignment window, follow these steps to assign the servers to the pool:

3. Select ucs for the Pool Assignment field.
4. Keep the Server Pool Qualification field at default.
5. Select the Host Firmware Package as ucs_FW_2.2_3e_C from the drop-down list.
Configuring Operational Policies for the Template

In the Operational Policies Window, follow these steps:

7. Select ucs in the Power Control Policy field.
8. Click Finish to create the Service Profile template.
9. Click OK in the pop-up window to proceed.
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**.
3. Right-click Service Profile Templates ucs.
4. Select Create Service Profiles From Template.
5. The Create Service Profile from Template window appears.

6. In the Create Service Profile from Template window enter the following:
   a. In the field Naming Prefix, enter ucs.
   b. In the field Enter Name Suffix Starting Number, enter 1
   c. In the field Number of Instances, enter 13.

   **Note** Association of the Service Profiles will take place automatically.
Identifying the Servers

In the Equipment Tab, select filter Rack-Mounts, and click on Servers. In the right pane all the thirteen servers are displayed along with the details.

Figure 65  UCS Rack Servers Associated with Created Service Profiles

7. (Optional) Double click on an individual server instance and enter an appropriate text string (name or role) in the User Label as shown below. This could be helpful in identifying the server’s application specific roles.
Installing Red Hat Enterprise Linux 6.5 on C220 M4 Systems

The search heads and the admin nodes are C220 M4 servers. These servers should be populated with a minimum of 2 identical hard disk drives. The procedures documented in this section are applicable for all servers performing the admin and the search head functions.

Note
This requires RHEL 6.5 DVD/ISO for the installation.

Creating Virtual Drive Using Cisco 12G SAS RAID Controller Utility

To create a virtual drive using Cisco 12G SAS RAID Controller Utility, follow these steps:

1. Log in to the Cisco UCS 6296 Fabric Interconnect and launch the Cisco UCS Manager application.
2. Select the Equipment tab.
3. In the navigation pane expand Rack-Mounts and then Servers.
4. Right-click on the server and select **KVM Console**.

**Figure 67**  
**KVM Console**

5. In the KVM window, select **Macros** > **Static Macros** > **Ctrl-Alt-Del**.
6. After the initial server configurations and POST completes. Press Ctrl-R when the following screen appears.

7. In the Cisco 12G SAS Modular Raid Controller BIOS Configuration Utility, use the arrow keys to highlight the Cisco 12G SAS Modular Raid controller line item.
8. Press **F2** to open up the sub-menu. Select **Create Virtual Drive**.

9. In the Create New VD dialog box, highlight the RAID Level field, press **ENTER** key to select RAID-1.

10. Select the drives.

11. Press **ENTER** on the Advanced button.

12. In the Create Virtual Drive-Advance dialog box.

13. Select 128KB as the Strip size.

14. Select Read Ahead as the Read policy.

15. Select Write Back with BBU as the Write Policy.

16. Select Direct as the I/O Policy.
17. Check the Initialize check box.

18. Select OK and press ENTER key.

19. In the Create Virtual Drive-Advanced window, press OK to continue.

20. In the Create New VD dialog box, review the configuration and press OK to create the Virtual Drive.

22. Use the tab key to navigate to the Boot device field. Press enter to choose the VD0 as the boot device.

23. Use the tab key to select APPLY to save the changes.

24. Press ESC and select OK to exit out of this utility.
25. Use the Macro menu to send Ctrl-Alt-Del macro to reboot the server.

Installing the Operating System

1. In the KVM console, select Virtual Media > Activate Virtual Devices.
2. Then select Virtual Media > Map CD/DVD.

![Map CD/DVD for Selecting RHEL ISO Image](image)

3. In the Virtual Media – Map CD/DVD dialog box, click on Browse button to choose the RHEL 6.5 Operating System ISO image file.
4. Click Map Device.

![Browse for the ISO](image)

5. Select Macros > Static Macros > Ctrl-Alt-Del option to reboot the server.
6. On reboot, the machine detects the presence of the Red Hat Enterprise Linux Server 6.5 install media.

7. Select the **Install or Upgrade an Existing System**.

8. Skip the Media test and start the installation.

9. Click **Next**.

10. Select language of installation and Click **Next**.
11. Select Basic Storage Devices and Click Next.
Figure 80  RHEL Installation: Select Storage Device

What type of devices will your installation involve?

- **Basic Storage Devices**
  Installs or upgrades to typical types of storage devices. If you're not sure which option is right for you, this is probably it.

- **Specialized Storage Devices**
  Installs or upgrades to enterprise devices such as Storage Area Networks (SANs). This option will allow you to add FC/eS / iSCSI / FCIP disks and to filter out devices the installer should ignore.
12. Provide Hostname and configure Networking for the Host.
Figure 82  **RHEL Installation: Enter Host Name**

![RHEL Installation: Enter Host Name](image)

Figure 83  **RHEL Installation: Configure Network Settings**

![RHEL Installation: Configure Network Settings](image)
Installing Red Hat Enterprise Linux 6.5 on C220 M4 Systems

Figure 84 RHEL Installation: Configure IPv4 Settings for eth 1
Figure 85 RHEL Installation: Configure IPv4 Settings for eth 2

- Connection name: eth2
- Connect automatically
- Available to all users
- Addresses:
  - Address: 192.168.12.101
  - Netmask: 255.255.255.0
- DNS servers:
- Search domains:
- DHCP client ID:
- Require IPv4 addressing for this connection to complete
Figure 86 RHEL Installation: Select Region

Please select the nearest city in your time zone:

Selected city: Los Angeles, America (Pacific Time)

System clock uses UTC
13. In the type of Installation, select **Create Custom Layout**.
14. Create three partitions as follows, they will be assigned to /boot, swap and / respectively.

**Note**  This partition layout customization is optional but highly recommended. As the Splunk software will be installed on the root partition under /data/disk1, it is recommended to allocate and ensure that sufficient storage is available to / partition.
15. Set the mount point as /boot, and specify the Size to be 2048 MB, and select Fixed size, Click OK.

16. Create another standard partition of fixed size 512 MB for Swap by selecting File System Type as swap.
17. Create the third standard partition with Mount Point set to / and Select the Fill to maximum allowable size in the Additional Size Options.

18. Click Next to continue.
19. Continue with RHEL Installation as shown below.
Installing Red Hat Enterprise Linux 6.5 using Software RAID on C240 M4 Systems

20. Once the installation is complete reboot the system.
Repeat the steps 1 to 20 to install Red Hat Enterprise Linux 6.5 on the four other C220 M4 servers serving as search heads and admin nodes. While doing so, assign the host names as follows.
admin2 for the other C220 M4 admin server, and sh1, sh2, sh3 for the three C220 M4 search head servers. Assign the respective IP addresses to these servers by referring to Table 7.

Installing Red Hat Enterprise Linux 6.5 using Software RAID on C240 M4 Systems

The following section provides detailed procedures for installing Red Hat Enterprise Linux 6.5 using Software RAID (OS based Mirroring) on Cisco UCS C240 M4 servers.
There are multiple methods to install Red Hat Linux operating system. The installation procedure described in this deployment guide uses KVM console and virtual media from Cisco UCS Manager.
Installing Red Hat Enterprise Linux 6.5 using Software RAID on C240 M4 Systems

Note

This requires RHEL 6.5 DVD/ISO for the installation

1. Log in to the Cisco UCS 6296 Fabric Interconnect and launch the Cisco UCS Manager application.
2. Select the Equipment tab.
3. In the navigation pane expand Rack-Mounts and then Servers.
4. Right click on the server and select KVM Console.

Figure 95 Opening the KVM Console

5. In the KVM window, select the Virtual Media tab.
6. Click the Activate Virtual Devices found in Virtual Media tab.
7. In the KVM window, select the **Virtual Media** tab and Click the Map CD/DVD.

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

**Figure 98** **Browse for Red Hat Enterprise Linux ISO Image**

10. In the KVM window, select the KVM tab to monitor during boot.

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

12. Click **OK**.

13. Click **OK** to reboot the system.

14. On reboot, the machine detects the presence of the Red Hat Enterprise Linux Server 6.5 install media.

15. Select the Install or Upgrade an Existing System.
16. Skip the Media test and start the installation.

17. Click **Next**.

18. Select language of installation and Click **Next**.
### Figure 100  
**RHEL Installation: Select Language**

<table>
<thead>
<tr>
<th>Language</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>العربية</td>
</tr>
<tr>
<td>Assamese</td>
<td>আসামসিক</td>
</tr>
<tr>
<td>Bengali</td>
<td>বাংলা</td>
</tr>
<tr>
<td>Bengali (India)</td>
<td>বাংলা (ভারত)</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>Български</td>
</tr>
<tr>
<td>Catalan</td>
<td>Català</td>
</tr>
<tr>
<td>Chinese (Simplified)</td>
<td>中文 (简体)</td>
</tr>
<tr>
<td>Chinese (Traditional)</td>
<td>中文 (正體)</td>
</tr>
<tr>
<td>Croatian</td>
<td>Hrvatski</td>
</tr>
<tr>
<td>Czech</td>
<td>Čeština</td>
</tr>
<tr>
<td>Danish</td>
<td>Dansk</td>
</tr>
<tr>
<td>Dutch</td>
<td>Nederlands</td>
</tr>
<tr>
<td>Estonian</td>
<td>eesti keel</td>
</tr>
<tr>
<td>Finnish</td>
<td>suomi</td>
</tr>
<tr>
<td>French</td>
<td>Français</td>
</tr>
<tr>
<td>German</td>
<td>Deutsch</td>
</tr>
<tr>
<td>Greek</td>
<td>Ελληνικά</td>
</tr>
<tr>
<td>Gujarati</td>
<td>ગુજરાતી</td>
</tr>
<tr>
<td>Hebrew</td>
<td>עברית</td>
</tr>
<tr>
<td>Hindi</td>
<td>हिंदी</td>
</tr>
<tr>
<td>Hungarian</td>
<td>Magyar</td>
</tr>
<tr>
<td>Icelandic</td>
<td>íslenska</td>
</tr>
<tr>
<td>Ilokano</td>
<td>Ilokano</td>
</tr>
<tr>
<td>Indonesian</td>
<td>Bahasa Indonesia</td>
</tr>
</tbody>
</table>

![RHEL Installation: Select Language](image)
19. Select Basic Storage Devices and Click Next.
Figure 102  RHEL Installation: Select Storage Device

What type of devices will your installation involve?

- **Basic Storage Devices**
  - Installs or upgrades to typical types of storage devices. If you're not sure which option is right for you, this is probably it.

- **Specialized Storage Devices**
  - Installs or upgrades to enterprise devices such as Storage Area Networks (SANs). This option will allow you to add PCIe / iSCSI / iSCSI disks and to filter out devices the installer should ignore.
20. Provide Hostname and configure Networking for the Host
Figure 104  RHEL Installation: Enter Host Name

Figure 105  RHEL Installation: Configure Network Settings
Installing Red Hat Enterprise Linux 6.5 using Software RAID on C240 M4 Systems

**Figure 106**  RHEL Installation: Configure IPv4 Settings for eth 1

**Figure 107**  RHEL Installation: Configure IPv4 Settings for eth 2
Figure 108  RHEL Installation: Select Region

Please select the nearest city in your time zone:

Selected city: Los Angeles, America (Pacific Time)

- America/Los Angeles

* System clock uses UTC

Back  Next
21. Choose Create custom layout for Installation type
Following steps can be used to create two software RAID 1 partitions for boot and / (root) partitions.

22. Choose free volume and click on Create and choose RAID Partition
23. Choose “Software RAID” for File system Type and set size for Boot volume.
24. Similarly do for the other free volume.
Figure 113  RHEL Installation: RAID Configuration - Create Storage
25. Now similarly create RAID partitions for root (/) partition on both the devices and use rest of the available space
Figure 115  
**RHEL Installation: RAID Configuration- Create Storage**

![RHEL Installation: RAID Configuration- Create Storage](image_url)
Figure 116  RHEL Installation: RAID Configuration - Add Partition
Figure 117  RHEL Installation: RAID Configuration—Create Storage
26. The above steps created 2 boot and 2 root (/) partitions. Following steps will RAID1 Devices
27. Choose one of the boot partitions and click on **Create > RAID Device**
28. Choose this as /boot (boot device) and in RAID members, choose all the boot partitions created above in order to create a software RAID 1 for boot
29. Similarly repeat for / partitions created above choosing both members with mount point as “/”.
### RHEL Installation: RAID Configuration - Create Storage

<table>
<thead>
<tr>
<th>Device</th>
<th>Size (MB)</th>
<th>Mount Point/RAID Volume</th>
<th>Type</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID Devices</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>md0</td>
<td>2045</td>
<td>/boot</td>
<td>ext4</td>
<td></td>
</tr>
<tr>
<td>Hard Drives</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sdd1</td>
<td>2048</td>
<td></td>
<td>soft</td>
<td></td>
</tr>
<tr>
<td>sdd</td>
<td>112424</td>
<td></td>
<td>soft</td>
<td></td>
</tr>
<tr>
<td>sde</td>
<td>112424</td>
<td></td>
<td>soft</td>
<td></td>
</tr>
</tbody>
</table>

**Create Storage**

- Create Partition
  - Standard Partition
- Create Software RAID
  - RAID Device
    - RAID Partition
      - Create a RAID formatted partition
    - RADEV
      - LVM Volume Group
        - Create a logical volume on selected volume group
      - LVM Physical Volume
        - Create an LVM formatted partition

**Options**

- Cancel
- Create
Figure 123  RHEL Installation: RAID Configuration - Make RAID Device
30. Click on Next
Figure 125 RHEL Installation: RAID Configuration - Partitioning Warnings

![Partitioning Warnings](image)

*Note*  Swap partition can be created using the similar steps, however, since these systems are high in memory, this step is skipped (click Yes)

31. Click **Next**, and **Format**
32. Select **Default Settings** and click **Next**.
Figure 127  RHEL Installation: RAID Configuration- Boot Loader Operating System

33. Continue with RHEL Installation as shown below
34. Once the installation is complete reboot the system.

Repeat the steps 1 to 34 to install Red Hat Enterprise Linux 6.5 on other seven C240 M4 servers i.e. idx2 - idx8. The hostnames and their corresponding IP addresses are shown in Table 7.

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

<table>
<thead>
<tr>
<th>Hostname</th>
<th>eth0 Management Network (Hostname: &lt;hostname&gt;.mgmt)</th>
<th>eth1 Main interface used for most Splunk data traffic (Hostname: &lt;hostname&gt;)</th>
<th>eth2 Replication network (Hostname: &lt;hostname&gt;.rep)</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin1</td>
<td>10.29.160.101</td>
<td>192.168.11.101</td>
<td>192.168.12.101</td>
</tr>
<tr>
<td>admin2</td>
<td>10.29.160.102</td>
<td>192.168.11.102</td>
<td>192.168.12.102</td>
</tr>
<tr>
<td>idx1</td>
<td>10.29.160.103</td>
<td>192.168.11.103</td>
<td>192.168.12.103</td>
</tr>
<tr>
<td>Idx2</td>
<td>10.29.160.104</td>
<td>192.168.11.104</td>
<td>192.168.12.104</td>
</tr>
<tr>
<td>idx3</td>
<td>10.29.160.105</td>
<td>192.168.11.105</td>
<td>192.168.12.105</td>
</tr>
</tbody>
</table>
Cisco UCS Integrated Infrastructure for Big Data with Splunk Enterprise

Configuring CIMC Access Using the CIMC Configuration Utility on C3160

Introduction

The following sections provide an overview of how to setup CIMC network connectivity for UCS C3160.

- Cisco C-Series Rack Servers provides a physical local video and two usb connections for a usb keyboard, video monitor, and usb mouse connection through the front/back panel of the Rack Server using the Cisco provided dongle connector.
- All Rack Servers can have up to 4 active KVM over IP sessions in addition to the local connection at front or rear panels. All active sessions have full control of the console.
- KVM over IP supports text and graphics modes of the graphics controller and needs no manual setting to view data.

Cable Connectivity

Cable connections for C3160 servers:

| idx8   | 10.29.150.110 | 192.168.11.110 | 192.168.12.110 |
| sh1    | 10.29.150.111 | 192.168.11.111 | 192.168.12.111 |
| sh2    | 10.29.150.112 | 192.168.11.112 | 192.168.12.112 |
| sh3    | 10.29.150.113 | 192.168.11.113 | 192.168.12.113 |
| storage1 | 10.29.150.114 | 192.168.11.114 | 192.168.12.114 |

Above table shows how the hosts are assigned with their host names. Within the UCS domain, the eth1 network (i.e. 192.168.11.X subnet) over FI B is used as the primary network for all the Splunk related data traffic except the replication traffic. The Splunk index replication related data traffic will be configured to make use of eth2 over FI A.

For example, the host names associated with the various interfaces of an indexer i.e. idx1 are as follows:

**Indexer idx1:**

- **Hostname:** idx1
- **eth1:** used to ingest the data streaming in from forwarders, and for traffic between the search head and indexers. Hostname is idx1. eth1 is configured with Platinum QOS policy.
- **eth0:** used for management traffic such as SSH, Web UI, NTP sync. It is assigned with the hostname idx1.mgmt. eth0 is configured with Best Effort QOS policy.
- **eth2:** used by the indexers to replicate indexes across each other. It is assigned with a hostname idx.rep. eth2 is configured with Platinum QOS policy.
Connection for C3160 Server:
- Connect video monitor, usb keyboard and usb mouse to Cisco C3160 Rack Servers through the back panel using the Cisco provided dongle connector.
- Connect the Network Port of the C3160 Server to Management port of the management switch.

**Power up the KVM**

Complete these steps in order to power up the server:
1. Plug the power cord into the chassis.

**Note**
- CIMC initializes system standby (Power Off mode).
- CIMC is active and can be controlled through GUI or CLI, if you know the IP address.
2. Depress Front Panel Power:
   - The Fans begins to spin up.
   - Then POST sequence begins.
   - At this point you can either boot up or begin the installation process.
F8 to configure/view CIMC IP

While in BIOS you can press F8 for the CIMC IP address configuration and password reset.

1. Set NIC mode to Dedicated.
2. Set NIC redundancy to None
3. Choose IPv4 for Static configuration.
4. Enter the CIMC IP, Subnet and Gateway information.
5. After entering IP information, press **F5** in order to display the configured IP.

```
Cisco IMC Configuration Utility Version 2.0  Cisco Systems, Inc.
NIC Properties
  NIC mode:  [ ]  NIC redundancy
    Dedicated:  [ ]  None:  [X]
    Shared LOM:  [ ]  Active-standby:  [ ]
    Cisco Card:  [ ]  Active-active:  [ ]
    SIOC Slot:  1
IP (Basic)
  IPV4:  [X]  IPV6:  [ ]
  DHCP enabled:  [ ]
  CMC IP:  10.25.160.230
  Prefix/Subnet:  255.255.255.0
  Gateway:  10.25.160.1
  Pref DNS Server:  0.0.0.0
VLAN (Advanced)
  VLAN enabled:  [ ]
  VLAN ID:  1
  Priority:  0
```

6. Press **F1** and enter Additional Settings (optional).

```
Cisco IMC Configuration Utility Version 2.0  Cisco Systems, Inc.
Common Properties
  Hostname:  C3160-FC11954J73U
  Dynamic DNS:  [X]
  DNS Domain:
FactoryDefaults
  Factory Default:  [ ]
Default User (Basic)
  Default password:
  Reenter password:
Port Profiles
  Reset:  [ ]
  Name:
```

7. Press **F10** in order to save the configuration.
8. Press <ESC> to exit.

Access CIMC

1. Then point a Web browser to the configured CIMC IP address http://10.29.160.230
   - Default username: admin
   - Default password: password
2. Once logged in successfully. The server can be controlled using CIMC.
3. Click launch **KVM** console.

4. Restart the server by using KVM Console, **Macros > Static Macros > Ctrl-Alt-Del**.

```
Press <F2> Setup, <F6> Boot Menu, <F7> Diagnostics, <F8>Cisco IMC Configuration, <F12> Network Boot

Bios Version : C3160M3.2.0.2a.0.030920140606
Platform ID : C3160M

Cisco IMC IPv4 Address : 10.28.160.230
Cisco IMC MAC Address : FC:5B:39:0A:04:E4

Processor(s) Intel(R) Xeon(R) CPU ES-2695 V2 @ 2.40GHz
Total Memory = 256 GB Effective Memory = 256 GB
Memory Operating Speed 1666 MHz
```
Installing Redhat Enterprise Linux 6.5 software Raid (OS based Mirroring) on C3160 System using CIMC

The following section provides detailed procedures for installing Red Hat Linux 6.5.

Access CIMC

1. Then point a Web browser to the configured CIMC IP address http://10.29.160.230
   - Default username: admin
   - Default password: password

2. Once logged in successfully. The server can be controlled using CIMC.
Installing Redhat Enterprise Linux 6.5 software Raid (OS based Mirroring) on C3160 System using CIMC

3. Click launch KVM console.

Restart the server by using KVM Console, Macros > Static Macros > Ctrl-Alt-Del

1. In the KVM window, select the Virtual Media tab.

2. Click the Activate Virtual Devices found under Virtual Media tab.
3. In the KVM window, select the **Virtual Media** tab and Click the Map CD/DVD.

**Note**  The Red Hat Enterprise Linux 6.5 DVD is assumed to be on the client machine.

5. Click **Open** to add the image to the list of virtual media
6. In the KVM window, select the KVM tab to monitor during boot.

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

8. Click OK.

9. Click OK to reboot the system.

10. On reboot, the machine detects the presence of the Red Hat Enterprise Linux Server 6.5 install media.

11. Select the option Install or Upgrade an Existing System.
12. Select **Skip** to skip the Media test and start the installation.

13. Click **Next**.

14. Select language of installation and Click **Next**.
15. Select Basic Storage Devices and Click **Next**.
Installing Redhat Enterprise Linux 6.5 software Raid (OS based Mirroring) on C3160 System using CIMC

Figure 142 RHEL Installation: Select Device Type

What type of devices will your installation involve?

- **Basic Storage Devices**
  - Installs or upgrades to typical types of storage devices. If you're not sure which option is right for you, this is probably it.

- **Specialized Storage Devices**
  - Installs or upgrades to enterprise devices such as Storage Area Networks (SANs). This option will allow you to add FC/iSCSI/iDRAC disks and to filter out devices the installer should ignore.
Figure 144  RHEL Installation: Host Name

Figure 145  RHEL Installation: RAID Configuration for eth0
Installing Redhat Enterprise Linux 6.5 software Raid (OS based Mirroring) on C3160 System using CIMC

Figure 146  RHEL Installation: RAID Configuration for eth1

Figure 147  RHEL Installation: Select Region
17. Choose Create custom layout for Installation type.
Following steps can be used to create two software RAID 1 partitions for boot and / (root) partitions.

18. Choose free volume and click on Create and choose RAID partition.
19. Choose **Software RAID** for File system Type and set size for Boot volume.
20. Similarly do for the other free volume.
Figure 152 RHEL Installation: Create Storage

[Image of storage setup interface]

- **Device**: /dev/sde
- **Size (MB)**: 114473
- **Mount Point**:
- **Type**: Free
- **Format**: Free

Create Storage Options:
- Create Partition
  - Standard Partition
- Create Software RAID
  - RAID Partition
  - RAID Device
- Create LVM
  - LVM Volume Group
  - LVM Logical Volume
  - LVM Physical Volume

Options: Create, Edit, Delete, Reset, Back, Next
21. Now similarly create RAID partitions for root (/) partition on both the devices and use rest of the available space.
Figure 154  RHEL Installation: Create Storage

Drive /dev/sdX (14473 MB) (Model: ATA INTEL SSDSC2BB12)

<table>
<thead>
<tr>
<th>Device</th>
<th>Size (MB)</th>
<th>Mount Point/RAID/Volume</th>
<th>Type</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>2048</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sde1</td>
<td>112424</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Create Storage
- Create Partition
- RAID Partition
- RAID Device
- LVM Volume Group
- LVM Logical Volume
- LVM Physical Volume

Create/Cancel/Information
Figure 155  RHEL Installation: Add Partition

![RHEL Installation: Add Partition](image)

- Device: sdd
  - Size: 114473 MB
  - Type: ATA INTEL SSDSC2BB12
  - Format: Free

- Device: sde
  - Size: 2048 MB
  - Type: ATA INTEL SSDSC2BB12
  - Format: Free

- Device: Free
  - Size: 112424 MB
  - Type: ATA INTEL SSDSC2BB12
  - Format: Free

Mount Point: <Not Applicable>

File System Type: software RAID

Allowable Drives:
- sdd 114473 MB ATA INTEL SSDSC2BB12
- sde 114473 MB ATA INTEL SSDSC2BB12

Size (MB): 200

Additional Size Options:
- Fill to maximum allowable size

Encrypt

Create  Edit  Delete  Reset  Back  Next
Installing Redhat Enterprise Linux 6.5 software Raid (OS based Mirroring) on C3160 System using CIMC

**Figure 156**  
**RHEL Installation: Create Storage**

<table>
<thead>
<tr>
<th>Device</th>
<th>Size (MB)</th>
<th>Mount Point/RAID/Volume</th>
<th>Type</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdd1</td>
<td>2048</td>
<td>sdd1</td>
<td>RAID</td>
<td></td>
</tr>
<tr>
<td>sdd2</td>
<td>112424</td>
<td>sdd2</td>
<td>RAID</td>
<td></td>
</tr>
<tr>
<td>sde1</td>
<td>2048</td>
<td>sde1</td>
<td>RAID</td>
<td></td>
</tr>
<tr>
<td>Free</td>
<td>112424</td>
<td>Free</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 157**  
**RHEL Installation: Add Partition**

<table>
<thead>
<tr>
<th>Device</th>
<th>Size (MB)</th>
<th>Mount Point/RAID/Volume</th>
<th>Type</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdd1</td>
<td>2048</td>
<td>sdd1</td>
<td>RAID</td>
<td></td>
</tr>
<tr>
<td>sdd2</td>
<td>112424</td>
<td>sdd2</td>
<td>RAID</td>
<td></td>
</tr>
<tr>
<td>sde1</td>
<td>2048</td>
<td>sde1</td>
<td>RAID</td>
<td></td>
</tr>
<tr>
<td>Free</td>
<td>112424</td>
<td>Free</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
22. The above steps created 2 boot and 2 root (/) partitions. Following steps will RAID1 Devices

Figure 158  RHEL Installation: Select a Device

<table>
<thead>
<tr>
<th>Device</th>
<th>Size (MB)</th>
<th>Type</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdd1</td>
<td>2048</td>
<td>software RAID</td>
<td>✓</td>
</tr>
<tr>
<td>sdd2</td>
<td>112424</td>
<td>software RAID</td>
<td>✓</td>
</tr>
<tr>
<td>sde1</td>
<td>2048</td>
<td>software RAID</td>
<td>✓</td>
</tr>
<tr>
<td>sde2</td>
<td>112424</td>
<td>software RAID</td>
<td>✓</td>
</tr>
</tbody>
</table>

23. Choose one of the boot partitions and click on Create > RAID Device.
24. Choose this as /boot (boot device) and in RAID members, choose all the boot partitions created above in order to create a software RAID 1 for boot.
25. Similarly repeat for / partitions created above choosing both members with mount point as "/".
Cisco UCS Integrated Infrastructure for Big Data with Splunk Enterprise

### Installing Red Hat Enterprise Linux 6.5 software RAID (OS based Mirroring) on C3160 System using CIMC

**Figure 161** RHEL Installation: Create Storage

<table>
<thead>
<tr>
<th>Device</th>
<th>Size (MB)</th>
<th>Mount Point/ RAID/Volume</th>
<th>Type</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>md0 (dev/md0)</td>
<td>2045</td>
<td>/boot</td>
<td>ext4</td>
<td>✓</td>
</tr>
<tr>
<td>Hard Drives</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sdd</td>
<td>2048</td>
<td>md0</td>
<td>soft</td>
<td></td>
</tr>
<tr>
<td>sdd2</td>
<td>112424</td>
<td></td>
<td>soft</td>
<td></td>
</tr>
</tbody>
</table>

![Create Storage](image_url)

- **Create Partition**
  - Standard Partition
    - General purpose partition creation
- **Create Software RAID**
  - RAID Partition
    - RAID device
      - Requires at least 2 free RAID formatted partitions
- **Create LVM**
  - LVM Volume Group
    - Requires at least 3 free LVM formatted partitions
  - LVM Logical Volume
    - Create a logical volume on selected volume group
  - LVM Physical Volume
    - Create an LVM formatted partition

[Back] [Next]
Installing Redhat Enterprise Linux 6.5 software Raid (OS based Mirroring) on C3160 System using CIMC

**Figure 162  RHEL Installation: Select RAID Members**

<table>
<thead>
<tr>
<th>Device</th>
<th>Size</th>
<th>Mount Point/RAID/Volume</th>
<th>Type</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>raid0</td>
<td>2048</td>
<td>/boot</td>
<td>ext4</td>
<td></td>
</tr>
<tr>
<td>sdd1</td>
<td>112424</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sde1</td>
<td>2048</td>
<td>mdo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sde2</td>
<td>112424</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Make RAID Device:
- Mount Point: /
- File System Type: ext4
- RAID Device: md0
- RAID Level: RAID1
- RAID Members: sdd2
- Number of spares: 0
- Encrypt: Off
26. Click on **Next**.
Figure 164  RHEL Installation: Partitioning Warnings

Note  Swap partition can be created using the similar steps, however, since these systems are high in memory, this step is skipped (click Yes).

27. Click Next, and Format.
28. Select default settings and click **Next**.
29. Continue with **RHEL** Installation as shown below.
30. Once the installation is complete reboot the system.

Repeat the steps 1 to 30 to install Red Hat Linux 6.5 on other storage servers, if any.

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

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.

Configuring /etc/hosts

Setup /etc/hosts on the Admin node and other nodes as follows:

1. Login to the Admin Node (admin1)

```
ssh 10.29.160.101
```

2. Populate the host file with IP addresses and corresponding hostnames on the Admin node (admin1) and other nodes as follows
On Admin Node (admin1)

vi /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.11.101 admin1
192.168.12.101 admin1.rep
10.29.160.101 admin1.mgmt
192.168.11.102 admin2
192.168.12.102 admin2.rep
10.29.160.102 admin2.mgmt
192.168.11.103 idx1
192.168.12.103 idx1.rep
10.29.160.103 idx1.mgmt
192.168.11.104 idx2
192.168.12.104 idx2.rep
10.29.160.104 idx2.mgmt
192.168.11.105 idx3
192.168.12.105 idx3.rep
10.29.160.105 idx3.mgmt
192.168.11.106 idx4
192.168.12.106 idx4.rep
10.29.160.106 idx4.mgmt
192.168.11.107 idx5
192.168.12.107 idx5.rep
10.29.160.107 idx5.mgmt
192.168.11.108 idx6
192.168.12.108 idx6.rep
10.29.160.108 idx6.mgmt
192.168.11.109 idx7
192.168.12.109 idx7.rep
10.29.160.109 idx7.mgmt
192.168.11.110 idx8
192.168.12.110 idx8.rep
10.29.160.110 idx8.mgmt
192.168.11.111 sh1
192.168.12.111 sh1.rep
10.29.160.111 sh1.mgmt
192.168.11.112 sh2
192.168.12.112 sh2.rep
10.29.160.112 sh2.mgmt
192.168.11.113 sh3
192.168.12.113 sh3.rep
10.29.160.113 sh3.mgmt
192.168.11.114 storage1
10.29.160.114 storage1.mgmt

Setting Up Password-less Login

To manage all of the clusters nodes from the admin node we need to setup password-less login. It assists in automating common tasks with Cluster-Shell - 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, follow the steps below in order to enable password-less login across all the nodes.

1. Login to the Admin Node (admin1)
   
   ssh 10.29.160.101

2. Run the ssh-keygen command to create both public and private keys on the admin node.
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_key`.

```bash
for host in admin1 admin2 idx1 idx2 idx3 idx4 idx5 idx6 idx7 idx8 sh1 sh2 sh3 storage1
  do
    echo -n "$host -> "; ssh-copy-id -i ~/.ssh/id_rsa.pub $host
  done
```

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 shall be copied over to all the thirteen other servers by using cluster shell command after it gets installed. See the next Setup ClusterShell.

**Setup ClusterShell**

ClusterShell (or clush) is cluster wide shell to run commands on several hosts in parallel.

From the system connected to the Internet download Cluster shell (clush) and install it on admin1. Cluster shell is available from EPEL (Extra Packages for Enterprise Linux) repository.

```bash
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/clustershell-1.6-1.el6.noarch.rpm
scp clustershell-1.6-1.el6.noarch.rpm admin1:/root/
```

Login to admin1 and install cluster shell

```bash
yum -y install clustershell-1.6-1.el6.noarch.rpm
```

**Edit /etc/clustershell/groups** file to include hostnames for all the nodes of the cluster. Create four special groups besides the group that takes all the hosts of the cluster. These groups help target the cluster wide commands to a specific set of nodes grouped by their role in the Splunk deployment.

```bash
vi /etc/clustershell/groups
```

**Copy and paste the below content and save the groups file.**

```bash
admins: admin[1-2]
indexers: idx[1-8]
searchheads: sh[1-3]
```
storage: storage1
all-splunk: admin[1-2],sh[1-3],idx[1-8]
all: admin[1-2],sh[1-3],idx[1-8],storage1

[root@admin1 ~]# cat /etc/clustershell/groups
admins: admin[1-2]
indexors: idx[1-6]
sorchoheads: sh[1-3]
storage: storage1
all-splunk: admin[1-2],sh[1-3],idx[1-8]
all: admin[1-2],sh[1-3],idx[1-8],storage1

---

Note

- The groups may also be configured with the “mgmt” hostnames or even IP addresses directly.
- For more information and documentation on ClusterShell, visit https://github.com/cea-hpc/clustershell/wiki/UserAndProgrammingGuide
- When the IP address or different hostname i.e. idx1.mgmt or 192.168.12.103 is used to configure the /etc/clustershell/groups file, ClusterShell will not work until a manual SSH session is initiated to the machine by using that IP or hostname (as it requires to be in known_hosts file), for instance, as in the case below for idx1.mgmt and 192.168.12.103.

[root@admin1 ~]# ssh idx1.mgmt
The authenticity of host ‘idx1.mgmt (10.29.160.103)’ can’t be established.
Are you sure you want to continue connecting (yes/no)? yes

[root@admin1 ~]# ssh 192.168.12.103
The authenticity of host ‘192.168.12.103 (192.168.12.103)’ can’t be established.
Are you sure you want to continue connecting (yes/no)? yes

4. From the admin node i.e. admin1, copy over the /etc/hosts file to all the other servers.
   clush -a -B -x admin1 -c /etc/hosts

[root@admin1 ~]# clush -a -B -x admin1 -c /etc/hosts

---

Creating Red Hat Enterprise Linux (RHEL) 6.5 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
3. Alternatively, if you have access to a Red Hat ISO Image, Copy the ISO file to admin1.
   scp rhel-server-6.5-x86_64-dvd.iso admin1:/root/
Here we assume you have the Red Hat ISO file located in your present working directory.
   mkdir -p /mnt/rheliso
   mount -t iso9660 -o loop /root/rhel-server-6.5-x86_64-dvd.iso /mnt/rheliso/
4. Next, copy the contents of the ISO to the /var/www/html/rhelrepo directory
   cp -r /mnt/rheliso/* /var/www/html/rhelrepo
5. Now on admin1 create a .repo file to enable the use of the yum command.

```
vi /var/www/html/rhelrepo/rheliso.repo
[rhel6.5]
name=Red Hat Enterprise Linux 6.5
baseurl=http://10.29.160.101/rhelrepo
gpgcheck=0
enabled=1
```

6. Now copy 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 repo file yum requires httpd to be running on admin1 for other nodes to access the repository.

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

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

```
[root@admin1 ~]# clush -a -b -c /etc/yum.repos.d/rheliso.repo
```

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

**Note** This step is needed to install software on Admin Node (admin1) using the repo (such as httpd, createrepo, etc)

```
vi /etc/yum.repos.d/rheliso.repo
[rhel6.5]
name=Red Hat Enterprise Linux 6.5
baseurl=file:///var/www/html/rhelrepo
gpgcheck=0
enabled=1
```

9. Creating the Red Hat Repository Database.

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

```
yum -y install createrepo
```
10. Run `createrepo` on the RHEL repository to create the repo database on admin node

   cd /var/www/html/rhelrepo
   createrepo .

```
[root@admin ~]# cd /var/www/html/rhelrepo/
[root@admin rhelrepo]# createrepo .
Spawning worker 0 with 3763 pkgs
Workers Finished Gathering worker results
Saving Primary metadatas
Saving filio lists metadatas
Saving othet metadatas
Generating sqlite DBs
Sqlite DBs complete
[root@admin rhelrepo]#
```

11. Finally, purge the yum caches after httpd is installed (steps in section “Install Httpd”)

### Installing httpd

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

1. Install `httpd` on the admin node to host repositories.

The Red Hat repository is hosted using HTTP on the admin node, this machine is accessible by all the hosts in the cluster.

   `yum -y install httpd`

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

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

   `ServerName 10.29.160.101:80`
3. Start httpd.

   service httpd start
   chkconfig httpd on

4. Purge the yum caches after httpd is installed (step followed from section Setup Red Hat Repo).

   clush -a -B yum clean all
   clush -a -B yum repolist

---

**Note**

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

---

### Upgrading Cisco Network driver for VIC1227

The latest Cisco Network driver is required for performance and updates. 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&reltype=latest

In the ISO image, the required driver kmod-enic-2.1.1.66-rhel6u5.el6.x86_64.rpm can be located at \Linux\Network\Cisco\12x5\RHEL\RHEL6.5

From a node connected to the Internet, download, extract and transfer kmod-enic-2.1.1.66-rhel6u5.el6.x86_64.rpm to admin1 (admin node).

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-2.1.1.66-rhel6u5.el6.x86_64.rpm
   clush -a -b rpm -ivh kmod-enic-2.1.1.66-rhel6u5.el6.x86_64.rpm

---

```bash
[root@admin1 ~]# clush -a -b -c kmod-enic-2.1.1.66-rhel6u5.el6.x86_64.rpm
    clush -a -b rpm -ivh kmod-enic-2.1.1.66-rhel6u5.el6.x86_64.rpm
```

---
5. 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

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

[root@admin ~]# crush -a -B "modinfo enic | head -5"
```

```
admin[1-2], idx[1-8], sh[1-3], storage1 (14)
```

```
filename: /lib/modules/2.6.32-431.el6.x86_64/extra/enic/enic.ko
version: 2.1.1.66
license: GPL v2
author: Scott Feldman <scofeldm@cisco.com>
description: Cisco VIC Ethernet NIC Driver
```

### Installing xfsprogs

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

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

[root@admin ~]# crush -a -B yum -y install xfsprogs

```

```
Loaded plugins: product-id, 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 0:3.1.1-14.el6 will be installed
---> Finished Dependency Resolution

Dependencies Resolved

Package  Arch Version Repository Size
--------- ------- ------ ------- ----
Installing:
  xfsprogs x86_64 3.1.1-14.el6 rhe16.5 724 k

Transaction Summary

Install 1 Package(s)

Total download size: 724 k
Installed size: 3.2 M

```

```
installation: xfsprogs-3.1.1-14.el6.x86_64
Verifying : xfsprogs-3.1.1-14.el6.x86_64

Installed:
  xfsprogs.x86_64 0:3.1.1-14.el6

Complete!
```
NTP Configuration

1. 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 applications.

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

   ```
   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 10.29.160.101
   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` file 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
   ```

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 “service ntpd stop”
   ```
clush -a -b -x admin1 "ntpdate 10.29.160.101"
clush -a -b "service ntpd start"

7. Ensure restart of NTP daemon across reboots
clush -a -b "chkconfig ntpd on"

Enabling Syslog

Syslog must be enabled on each node to preserve logs regarding killed processes or failed jobs. Modern versions such as syslog-ng and rsyslog are possible, making it more difficult to be sure that a syslog daemon is present. One of the following commands should suffice to confirm that the service is properly configured:

```
clush -B -a rsyslogd -v
[root@admin1 ~]# clush -B -a rsyslogd -v
rsyslogd 5.8.10, compiled with:
   FEATURE_REGEXP:         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
See http://www.rsyslog.com for more information.
[root@admin1 ~]#
```

clush -B -a service rsyslog status

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 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. Now, Copy the `/etc/security/limits.conf` file from admin node (admin1) to all the nodes using the following command.

```bash
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],storage1 (14)
----------
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.

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

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.

## Disabling SELinux

SELinux must be disabled during the install procedure and cluster setup. SELinux can be enabled after installation and while the cluster is running.

SELinux can be disabled by editing `/etc/selinux/config` and changing the `SELINUX` line to `SELINUX=disabled`. The following command will disable SELINUX on all node.

```bash
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 ~]# [root@admin1 ~]# clush -a -b cat /etc/selinux/config
----------
admin[1-2],idx[1-8],sh[1-3],storage1 (14)
```

# 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

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

Note: The above command may fail if SELinux is already disabled.
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, set the number of TCP retries to 5 can help detect unreachable nodes with less latency.

1. Edit the file /etc/sysctl.conf and on admin node admin1 and add the following lines:
   
   net.ipv4.tcp_retries2=5
   
   Copy the /etc/sysctl.conf file from admin node (admin1) to all the nodes using the following command.
   
   clush -a -b -c /etc/sysctl.conf
   
2. Load the settings from default sysctl file /etc/sysctl.conf by running
   
   clush -B -a sysctl -p

Disabling the Linux Firewall

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

   clush -a -b "service iptables stop"
   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.

Configure VM Swapping

vm.swappiness 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=0 >> /etc/sysctl.conf"
   
   Load the settings from default sysctl file /etc/sysctl.conf
   
   clush -a -B "sysctl -p"
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/redhat_transparent_hugepage/enabled"
   clush -a -b "echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag"
   ```

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

2. On Admin node, run the following commands
   
   ```
   rm -f /root/thp_disable
   echo "echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled" > /root/thp_disable
   echo "echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag" >> /root/thp_disable
   ```

3. Copy the file over to all the nodes.
   
   ```
   clush -a -b -c /root/thp_disable
   ```

4. Append the content of file thp_disable to `/etc/rc.local`
   
   ```
   clush -a -b "cat /root/thp_disable >> /etc/rc.local"
   ```

Install OpenSSL

Install OpenSSL and OpenSSL-devel version 1.0.1e-30 and above for RHEL6.5 on all nodes. This addresses some important security vulnerabilities in OpenSSL. If openssl is already installed (generally the case), use the following command to perform the upgrade.

```
clush -a -b -c /root/openssl-*
clush -a -b yum install -y install krb5-devel zlib-devel
clush -a -b rpm -Uvh openssl-1.0.1e-*.rpm openssl-devel-1.0.1e-*.rpm
```

RPMs available at

http://mirror.centos.org/centos/6/updates/x86_64/Packages/openssl-1.0.1e-30.el6_6.5.x86_64.rpm

http://mirror.centos.org/centos/6/updates/x86_64/Packages/openssl-devel-1.0.1e-30.el6_6.5.x86_64.rpm

**Note**

Krb5-devel and zlib-devel are dependencies. If not installed, the OpenSSL upgrade process will fail.

Disable IPv6 Defaults

Disable IPv6 as the addresses used are IPv4.

```
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"
```

Load the settings from default sysctl file `/etc/sysctl.conf`
Installing LSI StorCLI utility on all Indexers and Archival Nodes

This section describes steps to configure non-OS disk drives as RAID1 using StorCli command as described below. All the drives are going to be part of a single RAID1 volume. This volume can be used for Staging any client data to be loaded to HDFS. This volume won’t be used for HDFS data.

From the website download storcli

Extract the zip file and copy storcli-1.14.12-1.noarch.rpm from the linux directory.

1. Download storcli and its dependencies and transfer to Admin node.
   
   scp storcli-1.14.12-1.noarch.rpm admin1:/tmp/

2. Copy storcli rpm to all the Indexers and Storage node(s) using the following commands:
   
   clush -a -b -X searchheads,admins -c /tmp/storcli-1.14.12-1.noarch.rpm

3. Run the below command to install storcli all the Indexers and Storage node(s).
   
   clush -a -b -X searchheads,admins rpm -ivh /tmp/storcli-1.14.12-1.noarch.rpm

Configuring the Virtual Drive on the Indexers

1. Create a script by name raid10.sh on the admin node and copy it over to all indexers.
   
   vi /root/raid10.sh

2. Paste the following contents into the file and save it.
   
   /opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid10 drives=$1:1-24 pdperarray=12 WB ra direct Strip=128

   **Note**
   Do not execute this script on the admin or search head nodes. This script is meant for only the Indexers.

   **Note**
   This script needs to be executed manually 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.

3. Change the mode to include execution privileges.
   
   chmod +x /root/raid10.sh

4. Copy the script over to all the indexers.
   
   clush --group=indexers -c /root/raid10.sh

   ```
   [root@admin ~]# cat raid10.sh
   /opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid10 drives=$1:1-24 pdperarray=12 WB ra direct Strip=128
   [root@admin ~]# clush --group=indexers -c /root/raid10.sh
   ```
Configuring the Virtual Drive on the Indexers

5. The above script 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}'
```

6. Run the script to create a single RAID10 volume as follows:

```
./raid10.sh <EnclosureID> # ← obtained by running the command above
```

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 wwwlsi.com

Creating the Virtual Drive on one Indexer Node

```
[root@admin ~]# ssh idx1
Last login: Thu Apr 9 04:04:38 2015 from admin1
[root@idx1 ~]#
[root@idx1 ~]# cat raid10.sh
/opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid10 drives=81:1-24 pdperarray=12 wdm direct strip=128

[root@idx1 ~]# /opt/MegaRAID/storcli/storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'

0
[root@idx1 ~]# ./raid10.sh 0
Controller = 0
Status = Success
Description = Add VD Succeeded
```

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.

Configuring Data Drives on Indexers

The following script will format and mount the RAID10 volume (virtual drive) that was created in the previous section. It looks at all available volumes on the indexers, but would skip OS/boot related volumes. The RAID10 volume will be mounted based on its UUID as /data/disk1. This script assumes a Strip size of 128K and one RAID10 volume with 12 physical drives in each span.

Configuring the XFS File System

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

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

```bash
vi /root/driveconf-idx.sh
#!/bin/bash
[[ "-x" == "$1" ]] && set -x && set -v && shift 1
count=1
for X in /sys/class/scsi_host/host?/scan
do
echo ' - - ' > ${X}
done
for D in $(lsblk | grep disk | cut -c1-3)
do
echo /dev/$D
X=/dev/$D
if [[ -b ${X} && `/sbin/parted -s ${X} print quit|/bin/grep -c boot` -ne 0 ]];
then
echo "$X bootable - skipping." continue
else
Y=${X##*/}1
echo $Y
echo "Setting up Drive => ${X}"
/sbin/parted -s ${X} mklabel gpt quit -s
/sbin/parted -a optimal ${X} mkpart 1 0% 100% quit
/sbin/mkfs.xfs -f -q -l size=393216b,lazy-count=1,sw=128k,su=128k,sw=12 -L ${Y} $X
(( $? )) && continue
#Identify UUID
UUID=`blkid $X1 | cut -d " " -f3 | cut -d "=" -f2 | sed 's/"//g'`
echo "UUID of $X1 = $UUID, mounting $X1 as UUID on /data/disk$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 command to copy `driveconf-idx.sh` to all the indexers.

```bash
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.

```bash
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.

```bash
clush --group=indexers -B df -h
clush --group=indexers -B mount
clush --group=indexers -B cat /etc/fstab
```

### Note

In-case there is a need to delete any partitions; it can be done so using the following: Run command `mount` to identify which drive is mounted to which device `/dev/sd<>` umount the drive for which partition is to be deleted and run fdisk to delete as shown below. Care to be taken **not to delete the OS partition** as this will wipe out the installed OS.
mount
umount /data/disk1 # <- disk1 shown as example
(export d; echo w;) | sudo fdisk /dev/sd<?>

---

Configuring Data Drives on Archival Nodes using CIMC

This section describes steps to configure non-OS disk drives as 4 RAID6 volumes each with 15 drives using StorCLI command as described below. These volumes shall be shared as NFS exports and will be used for archiving frozen data.

1. In the node admin1, create a shell script with the StorCLI commands to create four RAID6 volumes with 15 drives in each volume.

   ```
   vi /root/raid6.sh
   /opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid6 drives=$1:1-15 WB ra direct Strip=1024
   /opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid6 drives=$1:16-30 WB ra direct Strip=1024
   /opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid6 drives=$1:31-45 WB ra direct Strip=1024
   /opt/MegaRAID/storcli/storcli64 /c0 add vd type=raid6 drives=$1:46-60 WB ra direct Strip=1024
   ```

2. Copy over this script to all the storage nodes.

   ```
   clush --group=storage -B -c /root/raid6.sh
   ```

3. Log onto each of the storage servers as user root.

   ```
   Note
   This document covers the procedure for creating the RAID volumes on only one storage server. If there are multiple storage nodes required in the solution, this process needs to be repeated on all the storage nodes individually.
   ```

4. Uses the following command to add execute permissions to the shell script.

   ```
   chmod +x ./raid6.sh
   ```

   Execute the following command to get the enclosure ID of the controller.

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

5. Execute shell script with the EnclosureID obtained in the previous step.

   ```
   ./raid6.sh <EnclosureID>
   ```

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

   ```
   lsblk
   ```

   ```
   [root@storage1 ~]# /opt/MegaRAID/storcli/storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'
   ```
Cisco UCS Integrated Infrastructure for Big Data with Splunk Enterprise

Configuring Data Drives on Archival Nodes using CIMC

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

Configuring the XFS File System

The following script will format and mount the available volumes on each Archival node. OS boot partitions are skipped. All drives shall be mounted based on their UUID as /data/disk1, /data/disk2, and so on.

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

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

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.

2. Create a file by name “driveconf-arch.sh” and copy-paste the following contents.

vi driveconf-arch.sh

Note
Copy and paste the below content into the script file.

#!/bin/bash
[[ "-x" == "$1" ]] && set -x && set -v && shift 1
count=1
for X in /sys/class/scsi_host/host?/scan
do
echo '- - -' > ${X}
done
for D in /sys/class/block/$(lsblk | grep disk | cut -c1-3)
do
echo /dev/$D
X=/dev/$D
if [[ -b $X && `/sbin/parted -s $X print quit` && `/bin/grep -c boot -ne 0` ]]
then
echo "$X bootable - skipping."
continue
else
Y=${X##*/}
echo "$Y"
echo "Setting up Drive => $X"
/sbin/parted -s $X mklabel gpt quit -s
/sbin/parted -a optimal $X mkpart 1 0% 100% quit
/sbin/mkfs.xfs -f -q -l size=393216b,lazy-count=1, su=1024k, sw=13
-L "$Y" $X
(( $? )) && continue
#Identify UUID
UUID=`blkid $X | cut -d " " -f3 | cut -d "=" -f2 | sed 's/"//g'`
echo "UUID of $X = $UUID, mounting $X as UUID on /data/disk${count}"
/bin/mkdir -p /data/disk${count}
/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

3. Copy the driveconf-arch.sh script file to all the Storage nodes.
   clush --group=storage -B -c /root/driveconf-arch.sh

4. Execute the script from the admin node targeting all the storage nodes.
   clush --group=storage -B /root/driveconf-arch.sh

5. Run the following from the admin node to list the partitions and mount points
   clush --group=storage -B df -h
   clush --group=storage -B mount
   clush --group=storage -B cat /etc/fstab
```bash
[root@admin1 ~]# clush --group=storage -B df -h

---

storage1

Filesystem  Size  Used  Avail  Use%  Mounted on
/dev/md1   10G  2.0G  10G  2%    /
/tmpfs  127G  0  127G  0%  /dev/shm
/dev/md0   2.0G  64M  1.9G  4%  /boot
/dev/sdc1  48T  34M  48T  1%  /data/disk1
/dev/sdd1  48T  34M  48T  1%  /data/disk2
/dev/sdc3  48T  34M  48T  1%  /data/disk3
/dev/sdf1  48T  34M  48T  1%  /data/disk4

[root@admin1 ~]# clush --group=storage -B mount

---

storage1

/dev/md1 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5, mode=620)
tmpfs on /dev/shm type tmpfs (rw, rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/md0 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
/dev/sdc1 on /data/disk1 type xfs (rw, noatime, nodiratime, allocsize=128m, no barrier)
/dev/sdd1 on /data/disk2 type xfs (rw, noatime, nodiratime, allocsize=128m, no barrier)
/dev/sdc3 on /data/disk3 type xfs (rw, noatime, nodiratime, allocsize=128m, no barrier)
/dev/sdf1 on /data/disk4 type xfs (rw, noatime, nodiratime, allocsize=128m, no barrier)

[root@admin1 ~]#

[root@admin1 ~]# clush --group=storage -B cat /etc/fstab

---

# /etc/fstab
# Created by anaconda on Wed Apr  8 16:29:20 2015
#
# Accessible filesystems, by reference, are maintained under /dev/disk
# See man pages fstab(5), find(8), mount(8) and/or blkid(8) for more info
#
# UUID=daa22d79-5a23-4743-b595-a06350069791  /ext4  defaults  1  1
# UUID=27d45238-001e-41f4-8e94-0229ec403db8  /boot  ext4  defaults  1  2
#tmpfs  /dev/shm  tmpfs  defaults  0  0
devpts  /dev/pts  devpts  gid=5, mode=620  0  0
/sysfs  /sys  sysfs  defaults  0  0
proc  /proc  proc  defaults  0  0

UUID=ee0659e0-0edd-44a2-aecb-56d024e41bd8  /data/disk1 xfs  allocsize=128m, no atime, no barrier, nodiratime 0 0
UUID=76145c8f-c5cd-4019-ba09-9a07c74720c3  /data/disk2 xfs  allocsize=128m, no atime, no barrier, nodiratime 0 0
UUID=9ba0140d-3b05-4b00-9d91-1af00c0d029a5  /data/disk3 xfs  allocsize=128m, no atime, no barrier, nodiratime 0 0
UUID=1c8c7d9-6e56-4325-b7c9-8fadd1e1dc0  /data/disk4 xfs  allocsize=128m, no atime, no barrier, nodiratime 0 0
```
Cluster Verification

Note

In-case there is a need to delete any of the partitions; it can be done so using the following: Run
command ‘mount’ to identify which drive is mounted to which device /dev/sd<?> umount the
drive for which partition is to be deleted and run fdisk to delete as shown below. Care to be taken
not to delete the OS partition as this will wipe out the installed OS.

mount
umount /data/disk1 # <- disk1 shown as example
(echo d; echo w;) | sudo fdisk /dev/sd<?>

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.
Create script cluster_verification.sh 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 ""

Cisco UCS Integrated Infrastructure for Big Data with Splunk Enterprise

182


Cluster Verification

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 ${NC}"
clush -a -B "echo 'Storage Controller: '; `which lspci` | grep -i "raid -e storage -e lsi"
 echo ""
clush -a -B "dmesg | grep -i raid | grep -i scsi"
 echo ""
clush -a -B "lsblk -id | awk '{print \$1\,\$4}'|sort | nl"
 echo ""
 echo ""
 echo -e "$\{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 $\{NC\}"
clush -a -B "`which chkconfig` --list iptables 2>&1"
 echo ""
 echo ""
 clush -a -B "`which service` iptables status 2>&1 | head -10"
 echo ""
 echo ""
 echo -e "$\{green\}Transparent Huge Pages $\{NC\}"
clush -a -B " cat /sys/kernel/mm/*transparent_hugepage/enabled"
 echo ""
 echo ""
 echo -e "$\{green\}CPU Speed$\{NC\}"
clush -a -B "echo -n 'CPU speed Service: '; `which service` cpuspeed \ status 2>&1"
clush -a -B "echo -n 'CPU speed Service: '; `which chkconfig` --list \ cpuspeed 2>&1"
 echo ""
 echo ""
 echo -e "$\{green\}Hostname Lookup$\{NC\}"
clush -a -B " ip addr show"
 echo ""
 echo ""
Installing Splunk Enterprise 6.2.2

Splunk Architecture & Terminology

Splunk comes packaged as an ‘all-in-one’ distribution. The single file can be configured to function as one of or all of the following components:

In a distributed deployment, installations follow a 3-tier approach.

**Figure 168 Splunk Components**

- **Search Head**: In a distributed search environment, a Splunk Enterprise instance that handles search management functions, directing search requests to a set of search peers and then merging the results back to the user. A Splunk Enterprise instance can function as both a search head and a search peer. If it does only 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.

• **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.

• **Deploeyer (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, 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.

• 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.

• An Archival node is configured to host the frozen data generated by the indexers. (See the sections NFS Configurations for the Splunk Frozen data storage, and Configuring Archival of data from cold to frozen)

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, deployer, master node, and license master are configured.

Installation order will be in the following:

• Splunk Installation
• Configure License Master
• Configure Master Node
• Configure Indexing Cluster
• Configure Deployer
• Configure Search Head Cluster
• Configure Distribution Management Console
• Configure Archival of frozen data
• 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 the following table to plan and track assigned roles and hostnames/IP addresses:

<table>
<thead>
<tr>
<th>CVD Hostname</th>
<th>Function / Model</th>
<th>Hostname</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>sh1</td>
<td>Search Head 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C220 M4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh2</td>
<td>Search Head 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C220 M4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sh3</td>
<td>Search Head 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C220 M4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>idx1</td>
<td>Indexer 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C240 M4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>idx2</td>
<td>Indexer 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C240 M4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>idx3</td>
<td>Indexer 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C240 M4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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-6.2.2-255606-linux-2.6-x86_64.rpm
```

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

   ```
   clush -a -X storage -c ./splunk-6.2.2-255606-linux-2.6-x86_64.rpm --dest=/tmp
   ```

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

   ```
   clush -a -X storage chmod +x /tmp/splunk-6.2.2-255606-linux-2.6-x86_64.rpm
   ```

### Table 8 AssIGNED ROLES and IP Addresses

<table>
<thead>
<tr>
<th></th>
<th>Assigned Roles and IP Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>idx4</td>
<td>Indexer 4</td>
</tr>
<tr>
<td></td>
<td>C240 M4</td>
</tr>
<tr>
<td>idx5</td>
<td>Indexer 4</td>
</tr>
<tr>
<td></td>
<td>C240 M4</td>
</tr>
<tr>
<td>idx6</td>
<td>Indexer 4</td>
</tr>
<tr>
<td></td>
<td>C240 M4</td>
</tr>
<tr>
<td>idx7</td>
<td>Indexer 4</td>
</tr>
<tr>
<td></td>
<td>C240 M4</td>
</tr>
<tr>
<td>idx8</td>
<td>Indexer 4</td>
</tr>
<tr>
<td></td>
<td>C240 M4</td>
</tr>
<tr>
<td>admin1</td>
<td>Admin Box 1</td>
</tr>
<tr>
<td></td>
<td>(Master Node, License Master,</td>
</tr>
<tr>
<td></td>
<td>Deployment Configuration Manager,</td>
</tr>
<tr>
<td></td>
<td>Deployer)</td>
</tr>
<tr>
<td></td>
<td>C220 M4</td>
</tr>
<tr>
<td>admin2</td>
<td>Admin Box 2</td>
</tr>
<tr>
<td></td>
<td>Deployment Server</td>
</tr>
<tr>
<td></td>
<td>C220 M4</td>
</tr>
<tr>
<td>storage1</td>
<td>C3160</td>
</tr>
</tbody>
</table>

**Note** The IP addresses and hostnames used in this CVD can be found in Table 7.
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 i.e. /data/disk1 which serves as the mount point for the RAID10 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 -X storage -B rpm -ivh --prefix=/data/disk1
   /tmp/splunk-6.2.2-255606-linux-2.6-x86_64.rpm

   This step installs the Splunk enterprise and creates a splunk user.

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

6. Setup the Environment Variable:

   clush --group=all-splunk "echo SPLUNK_HOME=/data/disk1/splunk >> /etc/environment"

7. Logoff and log back into the server admin1.

8. Use the cluster-shell utility command to verify if the environment variable has been setup correctly.

   clush --group=all-splunk -B echo $SPLUNK_HOME

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 --group=all-splunk -B ls -l $SPLUNK_HOME
   clush --group=all-splunk -B ls -l $SPLUNK_HOME/bin/splunk
Installing Splunk Enterprise 6.2.2

Setting Up Login for Splunk User

As mentioned above, the user ‘splunk’ 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 Cluster-Shell commands.

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

   ```bash
   clush --group=all-splunk -B "echo cisco123 | passwd splunk --stdin"
   ```

2. Log onto the admin node as user splunk using the password selected in the above step.

3. Run the `ssh-keygen` command to create both public and private keys on the admin node for the user ‘splunk’.

   ```bash
   ssh-keygen
   ```

   In this example, we are using a command line method with clear-text password for the sake of simplification. It is recommended to setup a strong password and set the password manually on each server individually to match the target datacenter’s security practices.
Installing Splunk Enterprise 6.2.2

4. Then run the following script from the admin node to copy the public key id_rsa.pub to all the Splunk servers i.e. indexers, search heads and admins of the cluster. `ssh-copy-id` appends the keys to the remote-host’s `.ssh/authorized_keys`.

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

5. Verify the password-less login, by entering the following command. The output should display the hostnames of all splunk servers.

```bash
clush --group=all-splunk hostname
```

```
[splunk@admin1 ~]$ clush --group=all-splunk hostname
admin1: admin1
idx2: idx2
idx4: idx4
admin2: admin2
sh3: sh3
idx3: idx3
idx6: idx6
sh2: sh2
idx5: idx5
idx7: idx7
sh1: sh1
idx1: idx1
idx8: idx8
```
Starting Splunk Enterprise Cluster

1. Log onto the admin node as user splunk.

2. Start the Splunk Enterprise

   clush --group=all-splunk $SPLUNK_HOME/bin/splunk start --accept-license

   [splunk@admin1 ~]$ clush --group=all-splunk $SPLUNK_HOME/bin/splunk start --accept-license

3. Verify the status of the Splunk Enterprise services.

   clush --group=all-splunk $SPLUNK_HOME/bin/splunk status

   [splunk@admin1 ~]$ clush --group=all-splunk $SPLUNK_HOME/bin/splunk status

   admin1: splunkd is running (PID: 13005).
   admin1: splunkd helpers are running (PIDs: 13006 13015 13067 13107).
   idx4: splunkd is running (PID: 2738).
   idx4: splunkd helpers are running (PIDs: 2739 2750 2800 2847).
   sh3: splunkd is running (PID: 4358).
   sh3: splunkd helpers are running (PIDs: 4359 4374 4464 4498).
   idx2: splunkd is running (PID: 14682).
   idx2: splunkd helpers are running (PIDs: 14683 14694 14744 14791).
   admin2: splunkd is running (PID: 7362).
   admin2: splunkd helpers are running (PIDs: 7363 7370 7421 7458).
   idx3: splunkd is running (PID: 10697).
   idx3: splunkd helpers are running (PIDs: 10698 10709 10759 10805).
   sh1: splunkd is running (PID: 30426).
   sh1: splunkd helpers are running (PIDs: 30427 30443 30540 30575).
   sh2: splunkd is running (PID: 25615).
   sh2: splunkd helpers are running (PIDs: 25616 25632 25729 25764).
   idx7: splunkd is running (PID: 11041).
   idx7: splunkd helpers are running (PIDs: 11042 11053 11103 11150).
   idx5: splunkd is running (PID: 16593).
   idx5: splunkd helpers are running (PIDs: 16594 16605 16655 16702).
   idx8: splunkd is running (PID: 11233).
   idx8: splunkd helpers are running (PIDs: 11234 11245 11295 11343).
   idx1: splunkd is running (PID: 10453).
   idx1: splunkd helpers are running (PIDs: 10454 10465 10515 10562).
   idx6: splunkd is running (PID: 13262).
   idx6: splunkd helpers are running (PIDs: 13263 13274 13324 13371).

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. The url will point to the default port of ‘8000’. For example http://admin1:8000

Note: If you have not added these servers to DNS, you will need to use the IP address, e.g. http://10.29.160.101:8000/
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 creates the user ‘splunk’ with the home directory of the original installation (for example: /data/disk1/splunk). If an alternative user is created, repeat instructions under “Setting up login for ‘splunk’” in the previous section.

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 utilized to run all Splunk processes. If there is a requirement to run Splunk as a different user, perform the following with:
  – Export /data/disk1/splunk as $SPLUNK_HOME, add it to the PATH
  – Home Dir. For new users should be Splunk installation directory (/data/disk1/splunk/)
  – Stop all splunk processes ($SPLUNK_HOME/bin/splunk stop)
  – Chown -R user:usergroup $SPLUNK_HOME/*
  – Change or sudo to new user
  – Start all splunk processes ($SPLUNK_HOME/bin/splunk start)
• When the CVD refers to the user ‘splunk’, substitute the alternate user.

Initializing Splunk on Boot

Log onto the admin server ‘admin1’ as root user.

From the command line, launch the following command:

```
clush --group=all-splunk $SPLUNK_HOME/bin/splunk enable boot-start -user splunk
```

This will initialize splunk running as user ‘splunk’ if any server is rebooted. If the splunk user account is not ‘splunk’, change the -user reference accordingly.
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>
<thead>
<tr>
<th>Function</th>
<th>Default Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management Port</td>
<td>8089</td>
</tr>
<tr>
<td>Web Interface</td>
<td>8000</td>
</tr>
</tbody>
</table>

NFS Configurations for the Splunk Frozen Data Storage

Create the User Splunk in the Storage Servers

1. From the node admin1, execute the following command to check the user splunk’s user and group identification info.
   ```
sudo -u splunk id
   
[root@admin1 ~]# sudo -u splunk id
uid=500(splunk) gid=500(splunk) groups=500(splunk)
   ```

2. Take a note of the uid and gid fields output from the command output

   Note In this case, the splunk user has been created with uid=500 and gid=500

3. Create a group named splunk and user named splunk with matching ids on all the storage nodes.
   ```
   clush --group=storage -B groupadd --gid=500 splunk
   clush --group=storage -B useradd --gid=500 --uid=500 splunk
   
[root@admin1 ~]# clush --group=storage -B groupadd --gid=500 splunk
[root@admin1 ~]# clush --group=storage -B useradd --gid=500 --uid=500 splunk
   ```

   Note The user splunk gets created without a password. If necessary use the passwd command to assign a password to this user.
NFS Server Setup on Archival Nodes

This section describes the procedure to configure NFS server on the storage servers. As described in section Configuring Data Drives on Archival Nodes using CIMC, each archival node consists of four volumes. They are mounted locally on the archival node i.e. hostname storage1, as /data/disk1, /data/disk2, /data/disk3 and /data/disk4. In each of these volumes a directory tree is created as /splunk/frzn/<indexer1-hostname>, <indexer2-hostname>, …]. The indexer specific directory under /data/disk[1-4]/splunk/frzn/ is then assigned to the respective indexer, resulting in one or more indexers would be getting assigned to a volume on the archival nodes depending on the number of indexers and number of storage nodes available in the particular Splunk deployment.

Use the following tables as a guideline to map a volume on a given storage node to one or more indexers. For more information about how Splunk stores and manages the Frozen data, see Configuring Archival of Data From Cold to Frozen.

Scenario A: 4 Indexers, 6 Indexers, 8 Indexers (or) 16 Indexers with only One Storage Node

In this scenario, since there is only a single storage node, a sub-directory is created in each of the volumes and assigned to an indexer. The location of the indexer specific sub-directory is determined by the ratio between the numbers of storage nodes and indexers.

The following table shows the sample scenarios. There is no hard and fast rule about how to perform this mapping; it can be changed as per the requirements of the particular deployment.

<table>
<thead>
<tr>
<th>Volume Name</th>
<th>1 Storage Node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4 indexers</td>
</tr>
<tr>
<td>1 /data/disk1</td>
<td>idx1</td>
</tr>
<tr>
<td>2 /data/disk2</td>
<td>idx2</td>
</tr>
<tr>
<td>3 /data/disk3</td>
<td>idx3</td>
</tr>
<tr>
<td>4 /data/disk4</td>
<td>idx4</td>
</tr>
</tbody>
</table>

Scenario B: 8 Indexers (or) 16 Indexers with Two Storage Nodes

The following table shows the mapping of eight or sixteen indexers across two storage nodes.

<table>
<thead>
<tr>
<th>Volume Name</th>
<th>8 Indexers</th>
<th>16 Indexers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Storage node #1</td>
<td>Storage node #2</td>
</tr>
<tr>
<td>1 /data/disk1</td>
<td>idx1</td>
<td>idx5</td>
</tr>
<tr>
<td>2 /data/disk2</td>
<td>idx2</td>
<td>idx6</td>
</tr>
</tbody>
</table>
The script provided in this section creates two sub-directories per disk volume thereby mapping two indexers to a disk volume.

1. Install the NFS tools on the storage servers.
   ```bash
   clush --group=storage -B yum install -y nfs-*
   ```

2. Create a file by name nfs_server_setup.sh.
   ```bash
   vi /root/nfs_server_setup.sh
   ```

3. Copy paste the following contents and save the file.
   ```bash
   #!/bin/bash
   ## Setup NFS Server for Splunk Frozen storage and setup exports
   ## Stop the NFS Service
   service nfs stop
   service rpcbind stop
   indexer=1
   ## Create the NFS Export directories and prepare the /etc/exports file
   for diskName in disk1 disk2 disk3 disk4
   do
     echo /data/$diskName
     mkdir -p /data/$diskName/splunk/frzn  ## Exported for Frozen Data from Splunk cluster
     ## Create the root directory for each indexer by their hostnames.
     ## Each volume is dedicated for two indexers.
     mkdir -p /data/$diskName/splunk/frzn/idx$indexer
     ((indexer++))
     chown -R splunk:splunk /data/$diskName/*
     ## Add the directory paths to the NFS Exports file.
     echo "/data/$diskName/splunk/frzn 192.168.11.0/24(rw,sync)" >> /etc/exports
   done
   ## Start the NFS Service in the proper order
   service rpcbind start
   service nfs start
   exit 0
   ```

   **Note** This script creates the necessary directories to accommodate eight indexers in the four RAID6 volumes and configures them to be NFS exports. In a deployment scenario where the number of indexers and storage nodes are different from what is described in this CVD, the script will need to be modified accordingly.
4. Change the mode to make it into an executable script.
   ```
   chmod +x /root/nfs_server_setup.sh
   ```

5. Copy over the script `nfs_server_setup.sh` to all the storage nodes.
   ```
   clush --group=storage -B -c /root/nfs_server_setup.sh
   ```

6. Execute the `nfs_server_setup.sh` script on all the storage nodes.
   ```
   clush --group=storage -B /root/nfs_server_setup.sh
   ```

7. Check the availability of the mount points from the server `admin1`.
   ```
   showmount -e storagel
   ```

---

**Note**

- During the shutdown of NFS daemons may show errors if they weren’t previously running. It is normal.
- This document presumes that there is only one storage/archival node. The script can be easily modified to accommodate another storage node and to distribute frozen data generated by the indexers between the storage nodes.

---

**Note**

Even though there are eight partitions have been exported from the storage node, this solution makes use of only the exports named as `/data/disk[1-4]/splunk/frzn`. Refer to Splunk Documentation for further information how Splunk indexer stores indexes and the aging policy.
NFS Client Configurations on the Indexers

This section describes the procedure to configure NFS clients on the indexer servers. One mount point on the server will be shared between two indexers. But, the indexers shall be configured to make use of their own respective root directory mounted in their own file system as /data/frzn_data. It is a recommended best practice to perform “hard” NFS mount without attribute caching.

1. Install the NFS tools on all the indexers.

   ```
   clush --group=indexers -B yum install -y nfs-*
   ```

2. Create a file by name nfs_client_setup.sh.

   ```
   vi /root/nfs_client_setup.sh
   ```

3. Copy paste the following contents and save the file.

   ```bash
   #!/bin/bash
   
   # NFS Clientside configurations for the Indexers
   # Create a temporary script directory for holding auto generated scripts.
   CISCO_SCRIPT_DIR=/root/cisco/scripts
   mkdir -p $CISCO_SCRIPT_DIR
   rm -rf $CISCO_SCRIPT_DIR/*
   clush --group=indexers -B mkdir -p $CISCO_SCRIPT_DIR
   clush --group=indexers -B rm -rf $CISCO_SCRIPT_DIR/*
   
   ## Create the mount points on the indexers
   clush --group=indexers -B mkdir -p /mnt/frzn
   
   ## Create the RAID6 volume to indexer-set map.
   FRZN_LINK_NAME="frzn_data"
   for K in "${!diskMap[@]}"
   do
     echo "$K --- ${diskMap[$K]}"
     optParam="nolock,tcp,rw,hard,intr,timeo=600,retrans=2,ssize=131072,wsize=131072"
     fstabEntry="storage1:/data/$K/splunk/frzn /mnt/frzn $efs $optParam"
     mountParam="-t nfs storagel:/data/$K/splunk/frzn -o $optParam"
     echo "mount $mountParam" >> $CISCO_SCRIPT_DIR/mount_script_$K.sh
     echo "echo $fstabEntry >> /etc/fstab" >> $CISCO_SCRIPT_DIR/mount_script_$K.sh
     chmod +x $CISCO_SCRIPT_DIR/mount_script_$K.sh
     clush -w ${diskMap[$K]} -B -c $CISCO_SCRIPT_DIR/mount_script_$K.sh
     clush -w ${diskMap[$K]} $CISCO_SCRIPT_DIR/mount_script_$K.sh
   done
   
   echo "echo removing old mapping /data/$FRZN_LINK_NAME" >
   $CISCO_SCRIPT_DIR/frzn_link_setup.sh
   echo "rm -f /data/$FRZN_LINK_NAME" >> $CISCO_SCRIPT_DIR/frzn_link_setup.sh
   echo "ln -s /mnt/frzn/$hostname /data/$FRZN_LINK_NAME" >> $CISCO_SCRIPT_DIR/frzn_link_setup.sh
   chmod +x $CISCO_SCRIPT_DIR/frzn_link_setup.sh
   clush --group=indexers -B -c $CISCO_SCRIPT_DIR/frzn_link_setup.sh
   ```

4. Change the mode to make it into an executable script.
NFS Client Configurations on the Indexers

5. Execute the script

```
chmod +x /root/nfs_client_setup.sh
./nfs_client_setup.sh
```

```
[root@admin1 ~]# ./nfs_client_setup.sh
disk4  ---  idx7, idx8
disk1  ---  idx1, idx2
disk3  ---  idx5, idx6
disk2  ---  idx3, idx4
---------
idx[1-3] (3)
---------
removing old mapping /data/frzn_data
```

Note: If the mount point on the indexers i.e. /mnt/frzn was previously used, it needs to be un-mounted prior to executing the above script. In such a case, use the command `clush --group=indexers -B umount /mnt/frzn` command. Use this command with caution.

6. Verify the NFS setup in all the indexers.

```
clush --group=indexers -B "mount -l | grep splunk"
clush --group=indexers -B ls -l /data
```
NFS Client Configurations on the Indexers

[root@admin ~]# clush --group=indexers -B "mount -l | grep splunk"

IDX1
storagel:/data/disk1/splunk/active on /mnt/frzn type nfs (rw,nolock,tcp,hard,intr,t imeo=600,retrans=2,rs=131072,ws=131072,vers=4,addr=192.168.11.101,client.addr= 192.168.11.101)

IDX2
storagel:/data/disk1/splunk/active on /mnt/frzn type nfs (rw,nolock,tcp,hard,intr,t imeo=600,retrans=2,rs=131072,ws=131072,vers=4,addr=192.168.11.101,client.addr= 192.168.11.101)

IDX3
storagel:/data/disk2/splunk/active on /mnt/frzn type nfs (rw,nolock,tcp,hard,intr,t imeo=600,retrans=2,rs=131072,ws=131072,vers=4,addr=192.168.11.104,client.addr= 192.168.11.104)

IDX4
storagel:/data/disk2/splunk/active on /mnt/frzn type nfs (rw,nolock,tcp,hard,intr,t imeo=600,retrans=2,rs=131072,ws=131072,vers=4,addr=192.168.11.104,client.addr= 192.168.11.104)

IDX5
storagel:/data/disk3/splunk/active on /mnt/frzn type nfs (rw,nolock,tcp,hard,intr,t imeo=600,retrans=2,rs=131072,ws=131072,vers=4,addr=192.168.11.105,client.addr= 192.168.11.105)

IDX6
storagel:/data/disk3/splunk/active on /mnt/frzn type nfs (rw,nolock,tcp,hard,intr,t imeo=600,retrans=2,rs=131072,ws=131072,vers=4,addr=192.168.11.105,client.addr= 192.168.11.105)

IDX7
storagel:/data/disk4/splunk/active on /mnt/frzn type nfs (rw,nolock,tcp,hard,intr,t imeo=600,retrans=2,rs=131072,ws=131072,vers=4,addr=192.168.11.106,client.addr= 192.168.11.106)

IDX8
storagel:/data/disk4/splunk/active on /mnt/frzn type nfs (rw,nolock,tcp,hard,intr,t imeo=600,retrans=2,rs=131072,ws=131072,vers=4,addr=192.168.11.106,client.addr= 192.168.11.106)
Configuring Splunk Enterprise Licenses

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

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

1. Log onto the server admin1 as user admin.
2. Navigate to the licensing screen by clicking on **Settings ➔ Licensing**.

3. Click on **Change License Group**

4. In the Click on **Enterprise License** radio button

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
8. Click on **Restart now**.
9. Click **OK** to restart Splunk to complete the license installation.

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 i.e. admin1. This can be performed by following one of the two methods described below.

The first and preferred method is to use Cluster-Shell 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 the respective Web UI.

**Configure all the License Slaves at Once Using CLI (clush)**

1. From the admin node (master1) as user ‘splunk’ execute the command:

   ```bash
   clush --group=all-splunk -x admin1 -B $SPLUNK_HOME/bin/splunk edit licenser-localslave
   -master_uri https://admin1:8089
   ```
2. Next issue the command:

```
clush --group=all-splunk -x admin1 $SPLUNK_HOME/bin/splunk restart
```

This will restart all nodes except for admin1 and storage1 nodes.

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

```
Checking kvstore port [8191]: open
Checking kvstore port [8191]: open
Checking kvstore port [8191]: open
Checking configuration... Done.
Checking critical directories... Done
Checking indexes...
Validated: _audit _blocks _signature _internal _introspection _thefishbucket _history 

Done

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

Done
```

4. Proceed to ‘Verifying License-Slave Relationships’

(Optional) Configure License Slaves Individually Using the Web Interface

1. Log onto an indexer server i.e. idx1 as user admin. (e.g. https://idx1:8000)
2. Navigate to the licensing screen by clicking on Settings ➔ Licensing.
3. Click on the button Change to slave.
4. In the Change master association dialog, click on Designate a different Splunk instance as the master license server radio button.

5. Enter the Master license server URI in the format https://<IP-or-hostname>:8089. (e.g. 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.
7. Restart Splunk by clicking on **Restart now**.

**Verifying License-Slave Relationships**

Confirm the license configurations.

1. Go to the master node’s Splunk GUI, and navigate to **Settings > Licensing**.
2. At the bottom of this screen, Click **All indexer Details** to view the license slaves.
### Local server information

<table>
<thead>
<tr>
<th>Indexer name</th>
<th>admin1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume used today</td>
<td>0 MB</td>
</tr>
<tr>
<td>Warning count</td>
<td>0</td>
</tr>
<tr>
<td>Debug information</td>
<td><strong>All license details</strong></td>
</tr>
</tbody>
</table>

There should be thirteen license slaves listed that is, eight indexers, three search heads plus the master node itself.
### Configuring the Splunk Enterprise Cluster

#### Figure 178  Indexer Details

<table>
<thead>
<tr>
<th>Indexers connected to: admin1 (13)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. idx2</strong></td>
</tr>
<tr>
<td>active_pool_names</td>
</tr>
<tr>
<td>added_usage_parsing_warnings</td>
</tr>
<tr>
<td>label</td>
</tr>
<tr>
<td>pool_names</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>stack_names</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>warning_count</td>
</tr>
<tr>
<td><strong>2. idx8</strong></td>
</tr>
<tr>
<td>active_pool_names</td>
</tr>
<tr>
<td>added_usage_parsing_warnings</td>
</tr>
<tr>
<td>label</td>
</tr>
<tr>
<td>pool_names</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>stack_names</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>warning_count</td>
</tr>
<tr>
<td><strong>3. idx1</strong></td>
</tr>
<tr>
<td>active_pool_names</td>
</tr>
</tbody>
</table>

---

**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 Index Cluster

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

- A 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 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].

Configuring the Master Node (aka: Cluster Master)

To start with configure the admin node admin1 as the Indexer Cluster master.

1. Using your browser, go to the master node (admin1)
   http://hostname-or-IP:8000/ (for example, https://admin1:8000/)

2. Click on the Settings > Indexer Clustering.

*Figure 179 Indexer Clustering*
3. Click **Enable Indexer Clustering**.

*Figure 180  Enable Indexer Clustering*

4. Click the **Master Node** radio button, and Click **Next**.

*Figure 181  Select the Node to Enable Clustering*

5. Set the fields Replication Factor to be 2 and Search Factor to be 2. Setup a ‘Security Key’, in this installation ‘splunk+cisco’ was used as the security key. Click **Enable Master Node**.
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 will also increase disk consumption. Consult the documentation for more information.

http://docs.splunk.com/Documentation/Splunk/6.2.2/Indexer/Thereplicationfactor

Be certain to apply a Security key.

6. Click on **Go to Server Controls** to proceed with restarting Splunk as indicated.
7. Restart Successful message appears, Click **OK** to go back to the Login screen.
8. Log back in as the “admin” user.
9. Return to **Settings > Indexer Clustering**.
10. Message appears indicating that necessary number of peers must join the cluster. For a replication factor of 2, Splunk Enterprise needs a minimum of 2 peers.

    **Configure Indexing Peers**

    Configure all the 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 Cluster-Shell command (clush) to configure all the C240 M4 servers to be indexing peers to the cluster master in admin1. The second (optional) method is to configure each C240 M4 server as an indexing peer individually by accessing the respective Web UI.

Configure All Indexing Peers Using CLI (clush)

1. From the admin1 box, as the ‘splunk’ users, issue the command:

   ```bash
   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
   
   id0: The cluster-config property has been edited.
   id2: The cluster-config property has been edited.
   id3: The cluster-config property has been edited.
   id4: The cluster-config property has been edited.
   id5: The cluster-config property has been edited.
   id6: The cluster-config property has been edited.
   id7: The cluster-config property has been edited.
   id8: The cluster-config property has been edited.
   
   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
   
   id0: The cluster-config property has been edited.
   id2: The cluster-config property has been edited.
   id3: The cluster-config property has been edited.
   id4: The cluster-config property has been edited.
   id5: The cluster-config property has been edited.
   id6: The cluster-config property has been edited.
   id7: The cluster-config property has been edited.
   id8: The cluster-config property has been edited.
   
   clush --group=indexers $SPLUNK_HOME/bin/splunk restart
   
   clush --group=indexers $SPLUNK_HOME/bin/splunk restart
   
   clush --group=indexers $SPLUNK_HOME/bin/splunk restart
   
   clush --group=indexers $SPLUNK_HOME/bin/splunk restart
   
   clush --group=indexers $SPLUNK_HOME/bin/splunk restart
   
   clush --group=indexers $SPLUNK_HOME/bin/splunk restart
   ```

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

   ```bash
   clush --group=indexers $SPLUNK_HOME/bin/splunk restart
   
   clush --group=indexers $SPLUNK_HOME/bin/splunk restart
   
   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 must report number of available peers.

   **Figure 185 Available Peers in the Master Node**

4. Proceed to Setting Dedicated Replication Address.
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](http://docs.splunk.com/Documentation/Splunk/latest/Indexer/Restartthecluster)

(Optional) Configure Indexing Peers Individually Using the Web Interface

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 “Configure All Indexing Peers Using CLI (clush)” procedure on page 212.

To enable an indexer as a peer node:
1. Go to an Indexer node’s Splunk Web-UI – `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 **Clustering**.

5. Select **Enable clustering**.
6. Select Peer node and click **Next**.

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**.

![Enable Peer Node](image)

9. The message appears, "You must restart Splunk for the peer node to become active."
Figure 190  Restart Splunk for the Peer Node to get Active

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

Note   The below figures show the Splunk restart process on indexer idx1 (i.e. 10.29.160.103).
Figure 191  Restart Splunk in Server Control Setting Page

The page at 10.29.160.103:8000 says:

Are you sure you want to restart Splunk?

OK  Cancel
11. After the peer node restarts, check the Master node’s web UI. The Master node must report number of available peers.

12. 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 data traffic that happens between the indexers in Splunk Enterprise Indexer cluster. In this CVD, the eth2 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.

1. SSH to idx1
2. As the splunk user, edit the file `$SPLUNK_HOME/etc/system/local/server.conf`
3. Under the section `[clustering]`, include the line;
   ```
   register_replication_address=idx1.rep
   ```

Note It is important to make sure that the host name i.e. idx1.rep or IP address used when setting the `register_replication_address` field is local to the server on which the server.conf resides. The value entered must reflect the replication address of the local server i.e. idx1.rep.

4. Save the file
5. Repeat this across all indexers (idx1-8)
6. SSH to the master node, admin1
7. As user `splunk` issue the command:
   `$SPLUNK_HOME/bin/splunk rolling-restart cluster-peers`

Verify Cluster Configuration

1. Navigate to the master node’s web GUI (e.g. https://admin1:8000)
2. Select Settings -> Index Clustering
3. All eight (8) indexers should appear as searchable.
Configure Receiving on the Peer Nodes

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

```
[splunk@admin1 local]$ pwd
/data/disk1/splunk/etc/master-apps/_cluster/local
[splunk@admin1 local]$ vi inputs.conf
```

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

Include the content:

```
[splunktcp://9979]
connection_host = ip
```
3. Navigate to the admin1 web interface via browser.
4. Navigate to Settings > Distributed Environment > Indexer Clustering.

Figure 195  Indexer Clustering

5. Select Edit > Distribute Configuration Bundle.

Figure 196  Indexer Clustering: Distribute Configuration Bundle

7. Acknowledge the warning, and push changes.

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

**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 ‘outputs.conf’ file in the master node at $SPLUNK_HOME/etc/system/local directory

```
[splunk@admin1 root]$ cd $SPLUNK_HOME/etc/system/local/
[splunk@admin1 local]$ vi outputs.conf
```

2. Create an outputs.conf file with the following content:

```
#Turn off indexing on the master
[indexAndForward]
index = false

tcpout:
defaultGroup = search_peers
forwardedindex.filter.disable = true
indexAndForward = false
autoLB = true
```

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), it is required to utilize a virtual or physical load balancer according to the enterprises standards. Due to variability, the operator is suggested to use their own discretion in installing and configuring this. Further notes for configuration are provided under “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.

3. Click Enable Indexer Clustering.
4. In the **Enable Clustering** dialog box, click on **Search head node**.

5. Click **Next**.

6. Enter the hostname of the master node in the format https://<hostname_or_IP>. (For example, https://admin1)

7. Enter the Master port number (default: 8089).

8. Enter the same security key that was used while configuring the master node e.g. splunk+cisco.
9. Click **Enable search head node**.

![Search Head Node Configuration](image)

10. Click **Go to server controls** to view the Server controls screen and to restart Splunk.

![Restart Splunk](image)

Repeat the above steps to configure all three servers with hostnames sh1, sh2 and sh3 to be search heads. Verify the search head cluster members in the master node, by navigating to Setting > Indexer clustering.

11. Click the **Search Heads** tab.
Configure the Deployer

A Splunk Enterprise instance that distributes apps and certain other configuration updates to search head cluster members is referred to as a ‘Deployer’. Any Splunk Enterprise instance can be configured to act as the Deployer. In this solution the 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:
Configuring the Splunk Enterprise Cluster

```plaintext
[license]
active_group = Enterprise

[clustering]
access_logging_for_heartbeats = 1
max_peer_build_load = 5
mode = master
pass4SymKey = $L8+$9zMEWeYoYoxHQ==
replication_factor = 2

[shclustering]
pass4SymKey = splunk+cisco
```

[pass4SymKey] = your_secret_key (e.g. splunk+cisco)

4. Restart the admin1 instance ($SPLUNK_HOME/bin/splunk restart)

**Configure Search Head Cluster Members**

1. As the user ‘splunk’, Enter the following commands to make this search head join the search head cluster. Change the mgmt._uri per respective Search Head

   ```bash
   ```

2. Restart Splunk Search Head after the command is issued

   ```bash
   $SPLUNK_HOME/bin/splunk restart
   ```
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.

Following illustration shows a small search head cluster, consisting of three 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.

```
```

**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're only available to the search head that generates them.
The recommended approach is to forward the data directly to the indexers, without indexing separately on the search head. You do this by configuring the search head as a forwarder by creating an outputs.conf file on the search head that configures the search head for load-balanced forwarding across the set of search peers (indexers). The indexing on the search head, 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’.
3. Navigate to the newly created ‘local’ directory.
4. Within the $SPLUNK_HOME/etc/shcluster/apps(outputs/local/) directory, create the file outputs.conf with the following content.

   ```bash
   cd $SPLUNK_HOME/etc/shcluster/apps
   mkdir -p outputs/local
   vi outputs.conf
   ```

5. Copy and paste the following contents.

   ```bash
   # Turn off indexing on the master
   [indexAndForward]
   index = false
   [tcpout]
   defaultGroup = search_peers
   forwardedindex.filter.disable = true
   indexAndForward = false
   [tcpout:search_peers]
   autoLB = true
   ```

6. Execute the ‘apply shcluster-bundle’ command:

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

   Acknowledge the warning and then 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)
To verify Search Head Clustering, follow these steps:

1. SSH to any Search Head.

2. 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
```

3. Alternatively, you can 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
```

---

**Note**  Explicit instructions for configuring the designated load balancer will differ by vendor, but the functionality and load balancing direction is the same.
4. 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.
Configuring the Splunk Enterprise Cluster

```
  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’.
```

Configuring the Distributed Management 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 i.e. 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 > Distributed Management Console.

```
Figure 205  Select Distributed Management Console
```

3. In the Distributed Management Console app. Click the ‘change’ hyperlink to change the mode.
4. Click **Distributed**. This should show all the eight indexers as remote instances.
5. Select ‘edit’ on the admin1 box, the server must change roles to function properly.

6. Select only ‘License Master’ and ‘Cluster Master’.
7. Click Save
8. Click Apply Changes button at the top right
9. Confirm that changes have been saved successfully.

**Configure Search Heads in Distributed Management Console**

In the previous section the Distributed Management Console (DMC) was configured to manage the indexers and the master node. This section provides the procedure to configure DMC to monitor the search heads.

1. Navigate to the Master Node (admin1) via the GUI
2. Open Settings -> Distributed Environment -> Distributed search.
3. Select **Search Peers**.

**Figure 211  Search Peers**

Distributed search

**Distributed search**
Perform a search across multiple Splunk indexers.

**Distributed search setup**

- **Search peers**

4. Select **New**.
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

1. Repeat this process for the two other additional search heads
2. On the Master Node (admin1) navigate to “Settings > Distributed Management Console”
3. Select “setup” within the Distributed Management Console.
4. The three newly added search heads should be listed under ‘remote instances’ as ‘new’.
5. Select ‘edit’ within the table next to the instance name, and ensure that the server roles are ‘Search Head’ and ‘KV Store’.

6. Confirm changes and roles.
7. Ensure that the Master Node (admin1) does not have the role of ‘search head’.
8. Click Apply Changes.

9. Click Overview.

10. DMC should now display “Search Heads” within the overview.
Configuring Archival of Data From Cold to Frozen

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 of 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 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

2. Install the package as detailed in the documentation for the appropriate operating system of the Universal Forwarder host.

Configure an App within the Deployment Server

1. In a browser, navigate to the Splunk instance’s Web Interface of server admin2 (i.e. 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).

4. Using the command line, navigate to the Deployment Server, admin2.
5. Navigate to $SPLUNK_HOME/etc/deployment-apps/
6. Create the directory ‘outputTest’.
7. Within ‘outputTest’ create the directory ‘local’.

8. Create the file ‘outputs.conf’ and include the following contents:

```conf
[tcpout]
defaultGroup = search_peers
[tcpout:search_peers]
autoLB = true
forceTimeBasedAutoLB = true
```
9. As the splunk users, execute the command

   $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:cisco123" to the command line.

10. Navigate to the Web GUI on admin2 (http://admin2:8000), and navigate to **Settings > Forwarder Management**. Click **Apps**.

![Forwarder Management](image)

11. Zero apps have been deployed. Click **Server Class**
12. Click **create one** and give it the name **TestForwarder**.
13. The following screen will present options for adding apps and clients, Click Add Apps.

14. Click OutputTest in the ‘Unselected Apps’ to move it to ‘Selected Apps’.

15. The next screen will show the apps listed under Apps.

16. Click Add Clients.
Configuring the Splunk Enterprise Cluster

**Figure 223**  
*Add Clients to Server Class*

17. Within the ‘Edit Clients’ screen, add the hostname of the forwarder to the whitelist. In this instance, the forwarder used is named ‘fwd1’.

**Figure 224**  
*Edit Clients*

18. Click **Save**.

19. View the Forwarder Manager.

**Figure 225**  
*Forwarder Management*
20. 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
-rwxr-xr-x 7 splunk splunk 117 Apr 10 12:10 .
-rwxr-xr-x 13 splunk splunk 4096 Mar 27 09:43 ..
-rwxr-xr-x 4 splunk splunk 30 Mar 27 01:12 introspection_generator_addon
-rwxr-xr-x 5 splunk splunk 47 Mar 27 09:43 learned
-rwxr-xr-x 4 root root 35 Apr 10 12:10 outputTest
-rwxr-xr-x 5 splunk splunk 49 Mar 27 01:13 search
-rwxr-xr-x 4 splunk splunk 35 Mar 27 01:13 SplunkUniversalForwarder
[root@fwd1 apps]# ```

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)

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.
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).

6. Navigate to 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.
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 push data into the archival directory.

1. SSH into the mast node (admin1) as the ‘splunk’ user
2. Navigate to $SPLUNK_HOME/etc/master-apps/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
```

Add the following stanzas;

```plaintext
### 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
```

### 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
```

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.

4. Save the file.
7. Click **Edit > Distribute Configuration Bundle.**

8. Click **Distribute Configuration Bundle.**
9. A pop-up window will appear, select **Push Changes**.

10. Verify that the push was successful.

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.
Configuring the Splunk Enterprise Cluster

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 1m 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 maxKBps 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 forward 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:

   ```bash
   [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/cold/db
   thawedPath = $SPLUNK_DB/archival/thawed/db
   maxDataSize = 1024
   frozenTimePeriodInSecs = 3600
   coldToFrozenDir = /data/frzn_data/archival/frzn
   ```

   ```bash
   [root@fw1 data]# /opt/splunkforwarder/bin/splunk add oneshot -source ./your_test_file.log -sourcetype syslog -index archival -auth admin:your_admin_password
   Warning: overriding SPLUNK_HOME setting in environment ("/opt/splunk") with "/opt/splunkforwarder". If this is not correct, edit /opt/splunkforwarder/etc/splunk-launch.conf
   Oneshot '*/data/sdb/sbk/sbk/data/test.log' added
   [root@fw1 data]#
   ```

   ```bash
   /opt/splunkforwarder/bin/splunk add oneshot -source ./your_test_file.log -sourcetype syslog -index archival -auth admin:your_admin_password
   ```

   **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 your Master Node (admin1) in your browser. “Settings -> Distributed Management Console”.

6. Select “Indexing Performance: Deployment”.

**Figure 234** Select Indexing Performance Deployment

7. Note that the DMC reflects indexing rates and data passing through the system.

**Figure 235** Indexing Performance by Instance

8. Navigate to any of the search heads in your browser.

9. Click on ‘Searching and Reporting’.
10. In the Splunk search bar, enter the following search:
   `index="archival" | stats count by splunk_server`
11. Note the indexer(s).
12. In the Splunk search bar, enter the following search:
   `index="main" | stats count by splunk_server`
13. Change the time range picker to ‘All time’.
14. Change the search mode to ‘fast mode’.
15. Click on ‘search’ (magnifying glass).
16. Change view to ‘Visualization’ and set the chart type to ‘Column’. Note the distribution of data across each of the indexers.

Figure 236 Splunk Server Versus Count Graph

17. 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.

Figure 237 Splunk Server Versus Count: Statistics

18. Write down or take a screenshot of the totals per indexer for reference later.
19. Use shell access to navigate to one of the indexers which reported data.
20. Approximately 5 minutes after sending data to the indexers, proceed to step 24.
21. Issue the command ‘$SPLUNK_HOME/bin/splunk offline’.
22. Return to the browser and run the same search again.
23. Note down the distribution of events and the total. The event count from this search and the previous
(step 18) should be the same.
24. Bring your indexer back up ($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 Frozen Buckets to Archival Storage

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/frzn_data/archival/frzn
3. Issue the command:
   ```
   ls -la
   ```
   This displays the ‘frozen’ bucket(s) that have been moved.
Post-Test Cleanup

Removing Test Data

1. To remove the data indexed during the test, perform the following:
2. Stop the Splunk service on the admin node. As the splunk user on admin1, issue the command:
   
   ```bash
   $SPLUNK_HOME/bin/splunk stop
   ```
3. Stop all indexers:
   
   ```bash
   clush --group=indexers $SPLUNK_HOME/bin/splunk stop
   ```
4. SSH to each indexer (idx1-idx8). As the splunk user, issue the command
   
   ```bash
   $SPLUNK_HOME/bin/splunk clean eventdata -index main
   ```

   **Note** Alternatively, the clush command could be used delete the indexes all the peers at once by applying the force parameter ‘-f’. i.e. 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.

5. Confirm the cleaning of events *on every indexer*!!
6. Start the master node (admin1) as splunk user:
   
   ```bash
   $SPLUNK_HOME/bin/splunk start
   ```
7. Start indexing peers as user ’splunk’
   
   ```bash
   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.
4. Using your browser, navigate to the master node web interface (admin1).

5. Select Settings > Indexer Clustering > Edit > Distribute Configuration Bundle.

6. Click Distribute Configuration Bundle.
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’.
   ```bash
   rm -r outputTest/
   ```
3. Reload the deployment Server.
   ```bash
   $SPLUNK_HOME/bin/splunk reload deploy-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 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`
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 & robust delivery support. There should be centralized management of what’s 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 meta-data for all data sources, including: hostname & 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- 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 logfile), 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 from where it left off.

Centralized 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 & 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:

• Check on splunkbase.splunk.com for an app that is designed to handle the technology. For example, search for “cisco”.

Note: Do not use the “find more apps” function within the Splunk UI

• 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](https://splunkbase.splunk.com); for each, you will also see a complete description as well as the documentation on how to install them. For some, these add-ons are installed to the forwarders, in some cases they are installed to the indexers, or to 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 & 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.

A complete list of all Splunk supported apps can be found here:
https://splunkbase.splunk.com/apps/#/order/latest/author/splunk

---

**Conclusion**

Splunk Enterprise delivers operational visibility and digital intelligence by monitoring and 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, streamline the deployment, offer 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 what application demands. Up to 80 servers (4 racks) can be supported with no additional switching in a single UCS domain without any network over-subscription. Scaling beyond 4 racks (80 servers) 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 Material

This section provides the BOM for 13 node Splunk Enterprise cluster plus 1 Cisco UCS C3160 as an archival node.

Table 12: Bill of Materials

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCSC-C220-M4S</td>
<td>UCS C220 M4 SFF w/o CPU mem HD PCIe PSU rail kit</td>
<td>3</td>
</tr>
<tr>
<td>CON-3SNTP-C220M4S</td>
<td>3YR SMARTNET 24X7X4 UCS C220 M4 SFF w/o CPU mem HD</td>
<td>3</td>
</tr>
<tr>
<td>UCS-CPU-E52680D</td>
<td>2.50 GHz E5-2680 v3/120W 12C/30MB Cache/DDR4 2133MHz</td>
<td>6</td>
</tr>
<tr>
<td>UCS-MR-1X162RU-A</td>
<td>16GB DDR4-2133-MHz RDIMM/PC4-17000/dual rank/x4/1.2v</td>
<td>48</td>
</tr>
<tr>
<td>A03-D600GA2</td>
<td>600GB 6Gb SAS 10K RPM SFF HDD/hot plug/drive sled mounted</td>
<td>6</td>
</tr>
<tr>
<td>UCSC-MLOM-CSC-02</td>
<td>Cisco UCS VIC1227 VIC MLOM - Dual Port 10Gb SFP+</td>
<td>3</td>
</tr>
<tr>
<td>UCSC-RAILB-M4</td>
<td>Ball Bearing Rail Kit for C220 M4 and C240 M4 rack servers</td>
<td>3</td>
</tr>
<tr>
<td>UCSC-PSU1-770W</td>
<td>770W AC Hot-Plug Power Supply for 1U C-Series Rack Server</td>
<td>6</td>
</tr>
<tr>
<td>CAB-9K12A-NA</td>
<td>Power Cord 125VAC 13A NEMA 5-15 Plug North America</td>
<td>6</td>
</tr>
<tr>
<td>N20-BBLKD</td>
<td>UCS 2.5 inch HDD blanking panel</td>
<td>18</td>
</tr>
<tr>
<td>UCSC-SCCBL220</td>
<td>Supercap cable 950mm</td>
<td>3</td>
</tr>
<tr>
<td>UCSC-HS-C220M4</td>
<td>Heat sink for UCS C220 M4 rack servers</td>
<td>6</td>
</tr>
<tr>
<td>UCSC-MRAID12G</td>
<td>Cisco 12G SAS Modular Raid Controller</td>
<td>3</td>
</tr>
<tr>
<td>UCSC-MRAID12G-2G</td>
<td>Cisco 12Gbps SAS 2GB FBWC Cache module (Raid 0/1/5/6)</td>
<td>3</td>
</tr>
<tr>
<td>RHEL-2S-1G-3A</td>
<td>RHEL/2 Socket/1 Guest/3Yr Svcs Required</td>
<td>3</td>
</tr>
<tr>
<td>CON-ISV1-RH2S1G3A</td>
<td>ISV 24X7 Rhel/2 Socket/1 Guest List Price is ANNUAL</td>
<td>3</td>
</tr>
</tbody>
</table>

Cisco UCS C240 M4 Rack Server configuration for Splunk indexers

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCSC-C240-M4SX</td>
<td>UCS C240 M4 SFF 24 HD w/o CPU mem HD PCIe PS railkt w/expndr</td>
<td>8</td>
</tr>
<tr>
<td>CON-3SNTP-C240M4SX</td>
<td>3YR SMARTNET 24X7X4 UCS C240 M4 SFF 24 HD w/o CPUmem</td>
<td>8</td>
</tr>
<tr>
<td>UCS-CPU-E52680D</td>
<td>2.50 GHz E5-2680 v3/120W 12C/30MB Cache/DDR4 2133MHz</td>
<td>16</td>
</tr>
<tr>
<td>UCS-MR-1X162RU-A</td>
<td>16GB DDR4-2133-MHz RDIMM/PC4-17000/dual rank/x4/1.2v</td>
<td>128</td>
</tr>
<tr>
<td>UCS-HD12T10KS2-E</td>
<td>1.2 TB 6G SAS 10K rpm SFF HDD</td>
<td>192</td>
</tr>
<tr>
<td>UCS-SD120G0KSB-EV</td>
<td>120 GB 2.5 inch Enterprise Value 6G SATA SSD (BOOT)</td>
<td>16</td>
</tr>
<tr>
<td>Part Number</td>
<td>Description</td>
<td>Quantity</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>UCSC-PCI-1C-240M4</td>
<td>Right PCIe Riser Bd (Riser 1) 2onbd SATA bootdrv+ 2PCI slts</td>
<td>8</td>
</tr>
<tr>
<td>UCSC-MLOM-CSC-02</td>
<td>Cisco UCS VIC1227 VIC MLOM - Dual Port 10Gb SFP+</td>
<td>8</td>
</tr>
<tr>
<td>UCSC-PSU2V2-1200W</td>
<td>1200W V2 AC Power Supply for 2U C-Series Servers</td>
<td>16</td>
</tr>
<tr>
<td>CAB-9K12A-NA</td>
<td>Power Cord 125VAC 13A NEMA 5-15 Plug North America</td>
<td>16</td>
</tr>
<tr>
<td>UCSC-RAILB-M4</td>
<td>Ball Bearing Rail Kit for C220 M4 and C240 M4 rack servers</td>
<td>8</td>
</tr>
<tr>
<td>UCSC-SCCBL240</td>
<td>Supercap cable 250mm</td>
<td>8</td>
</tr>
<tr>
<td>UCSC-HS-C240M4</td>
<td>Heat sink for UCS C240 M4 rack servers</td>
<td>16</td>
</tr>
<tr>
<td>UCSC-MRAID12G</td>
<td>Cisco 12G SAS Modular Raid Controller</td>
<td>8</td>
</tr>
<tr>
<td>UCSC-MRAID12G-2G</td>
<td>Cisco 12Gbps SAS 2GB FBWC Cache module (Raid 0/1/5/6)</td>
<td>8</td>
</tr>
<tr>
<td>RHEL-2S-1G-3A</td>
<td>Rhel/2 Socket/1 Guest/3Yr Svcs Required</td>
<td>8</td>
</tr>
<tr>
<td>CON-ISV1-RH2S1G3A</td>
<td>ISV 24X7 Rhel/2 Socket/1 Guest List Price is ANNUAL</td>
<td>8</td>
</tr>
</tbody>
</table>

### Cisco UCS C220 M4 Rack Server configuration for Splunk admin nodes

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCSC-C220-M4S</td>
<td>UCS C220 M4 SFF w/o CPU mem HD PCIe PSU rail kit</td>
<td>2</td>
</tr>
<tr>
<td>CON-3SNTP-C220M4S</td>
<td>3YR SMARTNET 24X7X4 UCS C220 M4 SFF w/o CPU mem HD</td>
<td>2</td>
</tr>
<tr>
<td>UCS-CPU-E52620D</td>
<td>2.40 GHz E5-2620 v3/85W 6C/15MB Cache/DDR4 1866MHz</td>
<td>4</td>
</tr>
<tr>
<td>UCS-MR-1X162RU-A</td>
<td>16GB DDR4-2133-MHz RDIMM/PC4-17000/dual rank/x4/1.2v</td>
<td>16</td>
</tr>
<tr>
<td>A03-D600GA2</td>
<td>600GB 6Gb SAS 10K RPM SFF HDD/hot plug/drive sled mounted</td>
<td>4</td>
</tr>
<tr>
<td>UCSC-MLOM-CSC-02</td>
<td>Cisco UCS VIC1227 VIC MLOM - Dual Port 10Gb SFP+</td>
<td>2</td>
</tr>
<tr>
<td>UCSC-RAILB-M4</td>
<td>Ball Bearing Rail Kit for C220 M4 and C240 M4 rack servers</td>
<td>2</td>
</tr>
<tr>
<td>UCSC-PSU1-770W</td>
<td>770W AC Hot-Plug Power Supply for 1U C-Series Rack Server</td>
<td>4</td>
</tr>
<tr>
<td>CAB-9K12A-NA</td>
<td>Power Cord 125VAC 13A NEMA 5-15 Plug North America</td>
<td>4</td>
</tr>
<tr>
<td>UCSC-SCCBL220</td>
<td>Supercap cable 950mm</td>
<td>2</td>
</tr>
<tr>
<td>UCSC-HS-C220M4</td>
<td>Heat sink for UCS C220 M4 rack servers</td>
<td>4</td>
</tr>
<tr>
<td>N20-BBLKD</td>
<td>UCS 2.5 inch HDD blanking panel</td>
<td>12</td>
</tr>
<tr>
<td>UCSC-MRAID12G</td>
<td>Cisco 12G SAS Modular Raid Controller</td>
<td>2</td>
</tr>
<tr>
<td>UCSC-MRAID12G-2G</td>
<td>Cisco 12Gbps SAS 2GB FBWC Cache module (Raid 0/1/5/6)</td>
<td>2</td>
</tr>
<tr>
<td>RHEL-2S-1G-3A</td>
<td>Rhel/2 Socket/1 Guest/3Yr Svcs Required</td>
<td>2</td>
</tr>
<tr>
<td>CON-ISV1-RH2S1G3A</td>
<td>ISV 24X7 Rhel/2 Socket/1 Guest List Price is ANNUAL</td>
<td>8</td>
</tr>
</tbody>
</table>

**Cisco UCS Fabric Interconnects Fl6296**
## Bill of Material

<table>
<thead>
<tr>
<th>Description</th>
<th>Model/Configuration</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCS-FI-6296UP-UPG</td>
<td>UCS 6296UP 2RU Fabric Int/No PSU/48 UP/18p LIC</td>
<td>2</td>
</tr>
<tr>
<td>CON-SNTP-FI6296UP</td>
<td>SMARTNET 24X7X4 UCS 6296UP 2RU Fabrc Int/2 PSU/4 Fans</td>
<td>2</td>
</tr>
<tr>
<td>SFP-H10GB-CU3M</td>
<td>10GBASE-CU SFP+ Cable 3 Meter</td>
<td>28</td>
</tr>
<tr>
<td>UCS-ACC-6296UP</td>
<td>UCS 6296UP Chassis Accessory Kit</td>
<td>2</td>
</tr>
<tr>
<td>UCS-PSU-6296UP-AC</td>
<td>UCS 6296UP Power Supply/100-240VAC</td>
<td>4</td>
</tr>
<tr>
<td>N10-MGT012</td>
<td>UCS Manager v2.2</td>
<td>2</td>
</tr>
<tr>
<td>UCS-BLKE-6200</td>
<td>UCS 6200 Series Expansion Module Blank</td>
<td>6</td>
</tr>
<tr>
<td>UCS-FAN-6296UP</td>
<td>UCS 6296UP Fan Module</td>
<td>8</td>
</tr>
<tr>
<td>CAB-N5K6A-NA</td>
<td>Power Cord 200/240V 6A North America</td>
<td>4</td>
</tr>
<tr>
<td><strong>Cisco UCS C3160 Rack Server - Splunk archival node</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UCSC-C3160</td>
<td>Cisco UCS C3160 Base Chassis w/ 4x PSU 2x120GB SSD RailKit</td>
<td>1</td>
</tr>
<tr>
<td>CON-SNTP-C3160BSE</td>
<td>SMARTNET 24X7X4 UCS C3160 Dense Storage Server</td>
<td>1</td>
</tr>
<tr>
<td>CAB-N5K6A-NA</td>
<td>Power Cord 200/240V 6A North America</td>
<td>4</td>
</tr>
<tr>
<td>UCSC-C3X60-12SSD</td>
<td>Cisco UCS C3X60 2x120GB SATA Enterprise Value SSD</td>
<td>2</td>
</tr>
<tr>
<td>UCSC-C3X60-RAIL</td>
<td>UCS C3X60 Rack Rails Kit</td>
<td>1</td>
</tr>
<tr>
<td>UCSC-PSU1-1050W</td>
<td>UCS C3X60 1050W Power Supply Unit</td>
<td>4</td>
</tr>
<tr>
<td>UCSC-C3160-BEZEL</td>
<td>Cisco UCS C3160 System Bezel</td>
<td>1</td>
</tr>
<tr>
<td>UCSC-C3X60-SBLKP</td>
<td>UCS C3x60 SIOC blanking plate</td>
<td>1</td>
</tr>
<tr>
<td>UCSC-C3X60-SVRN4</td>
<td>Cisco C3X60 Server Node E5-2695 v2 CPU 256GB 4GB RAID cache</td>
<td>1</td>
</tr>
<tr>
<td>UCS-CPU-E52695B</td>
<td>2.40 GHz E5-2695 v2/115W 12C/30MB Cache/DDR3 1866MHz</td>
<td>2</td>
</tr>
<tr>
<td>UCS-MR-1X162RZ-A</td>
<td>16GB DDR3-1866-MHz RDIMM/PC3-14900/dual rank/x4/1.5v</td>
<td>16</td>
</tr>
<tr>
<td>UCSC-C3X60-R4GB</td>
<td>UCS C3X60 12G SAS RAID Controller with 4GB cache</td>
<td>1</td>
</tr>
<tr>
<td>UCSC-HS-C3X60</td>
<td>Cisco UCS C3X60 Server Node CPU Heatsink</td>
<td>2</td>
</tr>
<tr>
<td>UCSC-C3160-SIOC</td>
<td>Cisco UCS C3160 System IO Controller with mLOM mez adapter</td>
<td>1</td>
</tr>
<tr>
<td>UCSC-MLOM-CSC-02</td>
<td>Cisco UCS VIC1227 VIC MLQM - Dual Port 10Gb SFP+</td>
<td>1</td>
</tr>
<tr>
<td>UCSC-C3X60-56HD4</td>
<td>Cisco UCS C3X60 Four rows56x 4TB (Total:224TB) Drives</td>
<td>56</td>
</tr>
<tr>
<td>UCSC-C3X60-HD4TB</td>
<td>UCS C3X60 4TB NL-SAS 7.2K HDD including C3X60 HDD carrier</td>
<td>1</td>
</tr>
<tr>
<td>UCSC-C3X60-EX16T</td>
<td>Cisco UCS C3X60 Disk Exp Tray w/ 4x 4TB Total:16TB Drives</td>
<td>1</td>
</tr>
<tr>
<td>UCS-HD4T7KS3-E</td>
<td>4TB SAS 7.2K RPM 3.5 inch HDD/hot plug/drive sled mounted</td>
<td>4</td>
</tr>
<tr>
<td>RHEL-2S-1G-3A</td>
<td>Rhel/2 Socket/1 Guest/3Yr Svcs Required</td>
<td>1</td>
</tr>
</tbody>
</table>
If 6 TB drives are required in the Cisco UCS C 3160 rack server configuration, use the following Product IDs in place of the 4TB hard disk drives and related accessories.

**Table 13 6 TB drives (replacement option) for the Cisco UCS C3160**

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCSC-C3X60-56HD6</td>
<td>Cisco UCS C3X60 Four row of drives containing 56 x 6TB (Total)</td>
<td>1</td>
</tr>
<tr>
<td>UCSC-C3X60-HD6TB</td>
<td>UCS C3X60 6TB 12Gbps NL-SAS 7200RPM HDD w carrier- Top-load</td>
<td>56</td>
</tr>
<tr>
<td>UCSC-C3X60-EX24T</td>
<td>Cisco UCS C3160 Expander with 4x 6TB 7200RPM NL-SAS Drives</td>
<td>1</td>
</tr>
<tr>
<td>UCSC-C3X60-6TBRR</td>
<td>UCS C3X60 6TB 12Gbps NL-SAS 7200RPM HDD w carrier- Rear-load</td>
<td>4</td>
</tr>
</tbody>
</table>

**Table 14 Splunk Enterprise License**

<table>
<thead>
<tr>
<th>Splunk Enterprise 6.2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Splunk Enterprise</td>
</tr>
<tr>
<td>Service Support</td>
</tr>
</tbody>
</table>