

Cisco UCS Integrated Infrastructure for Big Data and Analytics with Cloudera for Data Science at Scale

Building a 28-Node Cluster

Last Updated: June 7, 2018



About the Cisco Validated Design Program

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

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

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

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

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

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

© 2018 Cisco Systems, Inc. All rights reserved.

Table of Contents

Executive Summary.....	7
Solution Overview	8
Introduction.....	8
Audience	8
Purpose of this Document.....	8
Solution Summary	9
Scaling the Solution	12
Cloudera Data Science Workbench	13
NVIDIA GPU	15
Technology Overview.....	16
Cisco UCS Integrated Infrastructure for Big Data and Analytics	16
Cisco UCS 6300 Series Fabric Interconnects.....	16
Cisco UCS C-Series Rack-Mount Servers.....	16
Cisco UCS Virtual Interface Cards (VICs)	18
Cisco UCS Manager	19
NVIDIA CUDA	19
Cloudera (CDH 5.13.0).....	19
Cloudera Data Science Workbench.....	21
Docker Containers.....	21
Kubernetes	22
Solution Design	23
Requirements	23
Rack and PDU Configuration.....	23
Port Configuration on Fabric Interconnects	25
Server Configuration and Cabling for Cisco UCS C240 M5.....	25
Software Distributions and Versions	25
Cloudera (CDH 5.13.0)	25
Red Hat Enterprise Linux (RHEL)	25
Software Versions	25
Fabric Configuration	26
Performing Initial Setup of Cisco UCS 6332 Fabric Interconnects	27
Configure Fabric Interconnect A	27
Configure Fabric Interconnect B	28
Logging Into Cisco UCS Manager	28
Upgrading Cisco UCS Manager Software to Version 3.2(2b).....	28
Adding a Block of IP Addresses for KVM Access.....	28
Enabling Uplink Ports	30

Configuring VLANs	32
Enabling Server Ports.....	34
Creating Pools for Service Profile Templates.....	35
Creating an Organization	35
Creating MAC Address Pools.....	36
Creating a Server Pool	38
Creating Policies for Service Profile Templates.....	40
Creating Host Firmware Package Policy	40
Creating QoS Policies.....	41
Creating the Local Disk Configuration Policy	43
Creating Server BIOS Policy	44
Creating the Boot Policy.....	47
Creating Power Control Policy.....	49
Creating a Service Profile Template	50
Configuring the Storage Provisioning for the Template	51
Configuring Network Settings for the Template	52
Configuring the vMedia Policy for the Template	56
Configuring Server Boot Order for the Template	56
Configuring Server Assignment for the Template	58
Configuring Operational Policies for the Template	59
Installing Red Hat Enterprise Linux 7.4	61
Post OS Install Configuration	82
Setting Up Password-less Login	82
Configuring /etc/hosts	83
Creating a Red Hat Enterprise Linux (RHEL) 7.4 Local Repo	83
Creating the Red Hat Repository Database.....	84
Setting up ClusterShell.....	85
Installing httpd.....	87
Set Up All Nodes to use the RHEL Repository	87
Configuring DNS	87
Upgrading the Cisco Network Driver for VIC1387	88
Installing xfsprogs	89
NTP Configuration	89
Enabling Syslog	90
Setting ulimit	91
Disabling SELinux.....	91
Set TCP Retries	92
Disabling the Linux Firewall.....	92

Disable Swapping.....	92
Disable Transparent Huge Pages	92
Disable IPv6 Defaults.....	93
Configuring Data Drives on Name Node And Other Management Nodes.....	93
Configuring Data Drives on Data Nodes.....	95
Configuring the Filesystem for NameNodes and Datanodes.....	95
Cluster Verification	97
Installing Cloudera	99
Prerequisites for CDH Installation	99
Cloudera Manager Repository	99
Setting Up the Local Parcels for CDH 5.13.0.....	100
Downloading Parcels.....	100
Setting Up the MariaDB Database for Cloudera Manager	104
Cloudera Manager Installation	108
Setting Up the Cloudera Manager Server Database	108
Installing Cloudera Manager	109
Starting The Cloudera Manager Server.....	110
Installing Cloudera Enterprise Data Hub (CDH5)	110
Setting up the Database	124
Starting the Cluster Services.....	128
Scaling the Cluster	129
Enabling High Availability	129
HDFS High Availability	129
YARN High Availability.....	136
Setting up YARN HA	136
Configuring Yarn (MR2 Included) and HDFS Services.....	138
Apache Kafka Installation and Configuration.....	139
Configuring Spark	142
Tuning Resource Allocation for Spark	143
Submitting a Job	143
Shuffle Performance Improvement	144
Improving Serialization Performance.....	144
Spark SQL Tuning	145
Compression for Hive	146
Changing the Log Directory for All Applications.....	146
Cloudera Data Science Workbench (CDSW)	148
Installing Prerequisites for CUDA	148
GCC Installation	149

Install Kernel Headers and Installation Packages.....	149
Install dkms.....	149
Install NVIDIA GPU Drivers.....	150
Installing CUDA	150
Installation Prerequisites for CDSW.....	153
Set Up a Wildcard DNS Subdomain	153
Supported JDK Version	154
IP Tables and Security on CDSW Nodes.....	154
Disable SELinux.....	155
Configure Block Devices.....	155
Download and Install CDSW with Cloudera Manager.....	155
Installing Apache Spark 2 on YARN	156
Add the Cloudera Data Science Workbench Service.....	160
Create the Administrator Account	164
Non-Kerberized Clusters	165
Using GPUs for Cloudera Data Science Workbench Workloads	166
Enable GPU Support in Cloudera Data Science Workbench.....	168
Create a Custom CUDA-Capable Engine Image	169
Allocate GPUs for Sessions and Jobs.....	172
Bill of Materials.....	174
About the Authors.....	178
Acknowledgements	178

Executive Summary

The technology that implements big data systems has matured to the point where it is in wide use across all industries addressing a wide variety of complex business problems; yet even as the technology has matured, the rate of data growth has increased. In addition, machine learning is gaining in prominence. Machine learning is a set of techniques for sophisticated pattern matching that came out of research into Artificial Intelligence. Machine learning needs all the data from the big data systems plus very high-performance computing.

Businesses are now faced with a new set of challenges, namely, making this data available to the diverse set of people who need it, publishing their results so the organization can make use of it, enabling the automated production of those results, while also managing the data for compliance and governance, and doing all of this in an efficient way that scales as the data continues to grow.

New, better solutions to old problems, and new applications, with new revenue streams, are now within grasp, but require approaches to hardware and software where agility is a primary design driver. IT organizations with traditional infrastructure and software solutions struggle to respond to changing business conditions and to manage this infrastructure. These environments require administrators to spend excessive time configuring new server, storage, and network resources in order to keep up with the scale demanded by growing computing and storage needs.

In addition, the introduction of machine learning into the mix adds new requirements. Many machine learning tasks, especially deep learning tasks, require the use of GPUs, a specialized, very high-performance processor that is massively parallel in nature. GPUs are installed on the servers and it is critically important that these high-performance processors also scale with the data growth.

Cisco UCS Integrated Infrastructure for Big Data and Analytics is an optimal choice where world class performance and reliability are base requirements. It is the strong foundation upon which solutions are built. The architecture has the designed-in ability to scale from a small starting solution to thousands of servers and hundreds of petabytes of storage, all managed from a single pane.

Cloudera Enterprise combines distributed data processing with machine learning and analytics into a single scalable platform that enables businesses to tackle their most complex problems. Cloudera is a leading provider of Apache Hadoop distributions and its integrated ecosystem of projects. These tools are used by companies around the world to tackle problems as diverse as predictive maintenance on fleets of hundreds of thousands of vehicles to real-time processing of petabyte-scale data for market surveillance and compliance on stock exchanges.

Together, Cisco and Cloudera combine to create a dependable deployment system to address today's most challenging problems.

Solution Overview

Introduction

Both big data and machine learning technology have progressed to the point where they are being implemented in production systems running 24x7. There exists a very clear need for a proven, dependable, high-performance platform for the ingestion, processing, storage and analysis of the data, as well as the seamless dissemination of the output, results and insights of the analysis.

This solution implements the Cisco UCS Integrated Infrastructure for Big Data and Analytics, a world-class platform specifically designed for demanding workloads that is both easy to scale and easy to manage, even as the requirements grow to thousands of servers and petabytes of storage; and the Cloudera Data Science Workbench, an integrated set of tools designed to enable flexible, fast access to the entire data store.

Many companies, recognizing the immense potential of big data and machine learning technology, are gearing up to leverage these new capabilities, building out departments and increasing hiring. However, these efforts face a new set of challenges:

- making the data available to the diverse set of people who need it,
- enabling access to high-performance computing resources, GPUs, that also scale with the data growth
- allowing people to work with the data using the environments they are familiar with,
- publishing their results so the organization can make use of it,
- enabling the automated production of those results,
- managing the data for compliance and governance,
- scaling the system as the data grows
- managing and administering the system in an efficient, cost-effective way

This solution is based on the Cisco UCS Integrated Infrastructure for Big Data and Analytics and includes computing, storage, connectivity, and unified management capabilities to help companies manage the immense amount of data being collected. It is built on the Cisco Unified Computing System (Cisco UCS) infrastructure, using Cisco UCS 6332 Series Fabric Interconnects, and Cisco UCS C-Series Rack Servers. This architecture is specifically designed for performance and linear scalability for big data and machine learning workloads.

Audience

The intended audience of this document includes sales engineers, field consultants, professional services, IT managers, partner engineering and customers who want to deploy the Cloudera Distribution with Apache Hadoop (CDH 5.13.0) and Cloudera Data Science Workbench (CDSW 1.3.0) on the Cisco UCS Integrated Infrastructure for Big Data and Analytics (Cisco UCS M5 Rack mount servers).

Purpose of this Document

This document describes the architecture and deployment procedures for Cloudera 5.13.0 with Cloudera Data Science Workbench 1.3.0 on a 28-node Cisco UCS C240 M5 cluster based on Cisco UCS Integrated Infrastructure for Big Data and Analytics.

Solution Summary

This CVD describes in detail the process of installing Cloudera 5.13.0 with Cloudera Data Science Workbench (CDSW 1.3.0) and the configuration details of the cluster. The current version of Cisco UCS Integrated Infrastructure for Big Data and Analytics offers the following configurations depending on the compute and storage requirements as shown in Table 1.

Table 1 Cisco UCS Integrated Infrastructure for Big Data and Analytics Configuration Options

	Performance (UCS-SP-C240M5-A2)	Capacity (UCS-SPC240M5L-S1)	High Capacity (UCS-SP-S3260-BV)
Servers	16 x Cisco UCS C240 M5 Rack Servers with SFF drives	16 x Cisco UCS C240 M5 Rack Servers with LFF drives	8 x Cisco UCS S3260 Storage Servers
CPU	2 x Intel Xeon Processor Scalable Family 6132 (2 x 14 cores, 2.6 GHz)	2 x Intel Xeon Processor Scalable Family 4110 (2 x 8 cores, 2.1 GHz)	2 x Intel Xeon Processor Scalable Family 6132 (2 x 14 cores, 2.6 GHz)
Memory	6 x 32 GB 2666 MHz (192 GB)	6 x 32 GB 2666 MHz (192 GB)	6 x 32 GB 2666 MHz (192 GB)
Boot	M.2 with 2 x 240-GB SSDs	M.2 with 2 x 240-GB SSDs	M.2 with 2 x 240-GB SSDs
Storage	26 x 1.8 TB 10K rpm SFF SAS HDDs or 12 x 1.6 TB Enterprise Value SATA SSDs	12 x 8 TB 7.2K rpm LFF SAS HDDs + 2 SFF rear hot-swappable 1.6 TB Enterprise Value SATA SSDs	24 x 6 TB 7.2K rpm LFF SAS HDDs
VIC	40 Gigabit Ethernet (Cisco UCS VIC 1387)	40 Gigabit Ethernet (Cisco UCS VIC 1387)	40 Gigabit Ethernet (Cisco UCS VIC 1387)
Storage Controller	Cisco 12-Gbps SAS Modular RAID Controller with 4-GB flash-based write cache (FBWC) or Cisco 12-Gbps Modular SAS Host Bus Adapter (HBA)	Cisco 12-Gbps SAS Modular RAID Controller with 2-GB flash-based write cache (FBWC) or Cisco 12-Gbps Modular SAS Host Bus Adapter (HBA)	Cisco 12-Gbps SAS Modular RAID Controller with 4-GB flash-based write cache (FBWC)
Network Connectivity	Cisco UCS 6332 Fabric Interconnect	Cisco UCS 6332 Fabric Interconnect	Cisco UCS 6332 Fabric Interconnect

Table 2 lists the configuration details for Cloudera Data Science Workbench. These servers provide the high-performance GPU compute capacity.

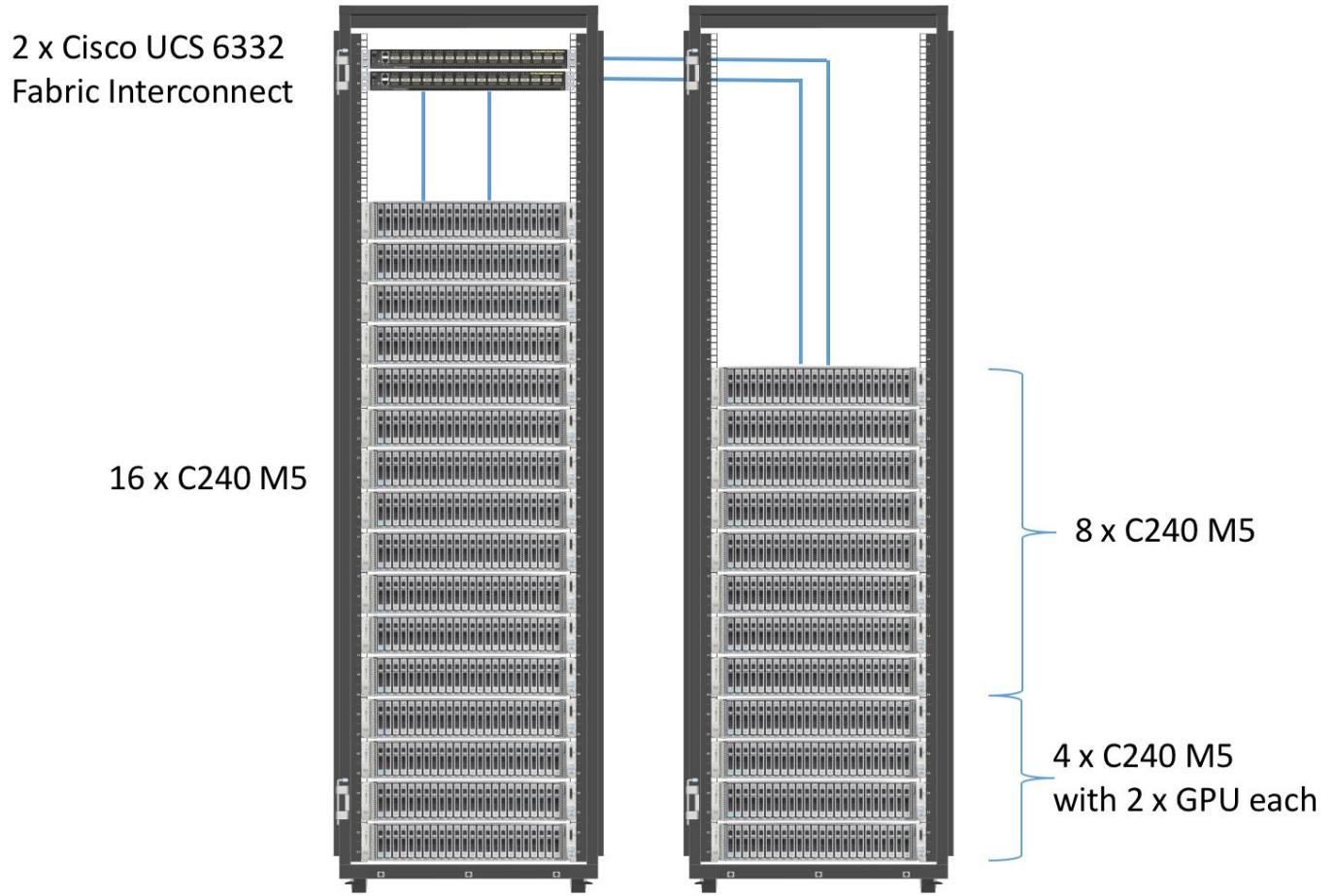
Table 2 Cisco UCS Integrated Infrastructure for Big Data and Analytics for CDSW

	Starter	High Performance
Servers	4 x Cisco UCS C240 M5 Rack Servers	4 x Cisco UCS C480 M5 Rack Servers
CPU	2 x Intel Xeon Processor Scalable Family 6132 (2 x 14 cores, 2.6 GHz)	2 x Intel Xeon Processor Scalable Family 6132 (2 x 14 cores, 2.6 GHz)
Memory	12 x 32 GB DDR4 (384 GB)	24 x 32 GB DDR4 (768 GB)
Boot	M.2 with 2 x 240-GB SSDs	M.2 with 2 x 240-GB SSDs
Storage	4 x 1.6 TB Enterprise Value SATA SSDs	8 x 1.6 TB Enterprise Value SATA SSDs
VIC	40 Gigabit Ethernet (Cisco UCS VIC 1387)	40 Gigabit Ethernet (Cisco UCS VIC 1387)

	Starter	High Performance
Storage Controller	Cisco 12-Gbps SAS Modular RAID Controller with 4-GB flash-based write cache (FBWC) or Cisco 12-Gbps Modular SAS Host Bus Adapter (HBA)	Cisco 12-Gbps SAS Modular RAID Controller with 4-GB flash-based write cache (FBWC) or Cisco 12-Gbps Modular SAS Host Bus Adapter (HBA)
Network Connectivity	Cisco UCS 6332 Fabric Interconnect	Cisco UCS 6332 Fabric Interconnect
GPU	2 x NVIDIA TESLA V100	6 x NVIDIA TESLA V100

Figure 1 depicts a 28-node starter cluster. Rack #1 has 16 Cisco UCS C240 M5 servers. Each link in the figure represents a 40 Gigabit Ethernet link from each of the 16 servers directly connected to a Fabric Interconnect. Rack #2 has 12 Cisco UCS C240 M5 servers. Every server is connected to both Fabric Interconnects.

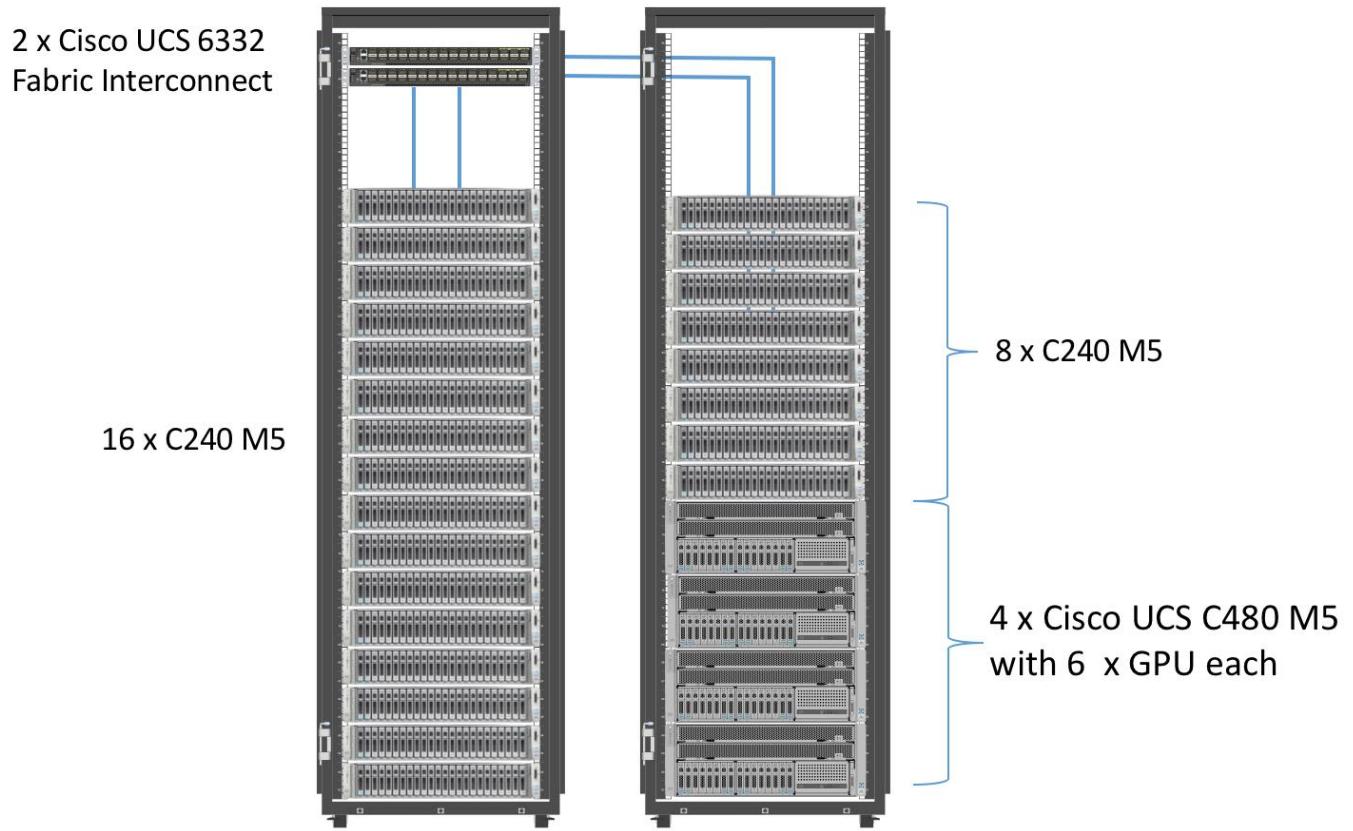
Figure 1 28 Node Starter Cluster Configuration for CDSW



Note: Power requirements per rack must be calculated as the exact values will change based on the power needs of the GPUs.

Figure 2 shows an alternate configuration for cases where more GPU capacity is needed. Four of the Cisco UCS C240 M5 servers from the previous figure are replaced with Cisco UCS C480 M5 servers. These servers support up to six GPUs each.

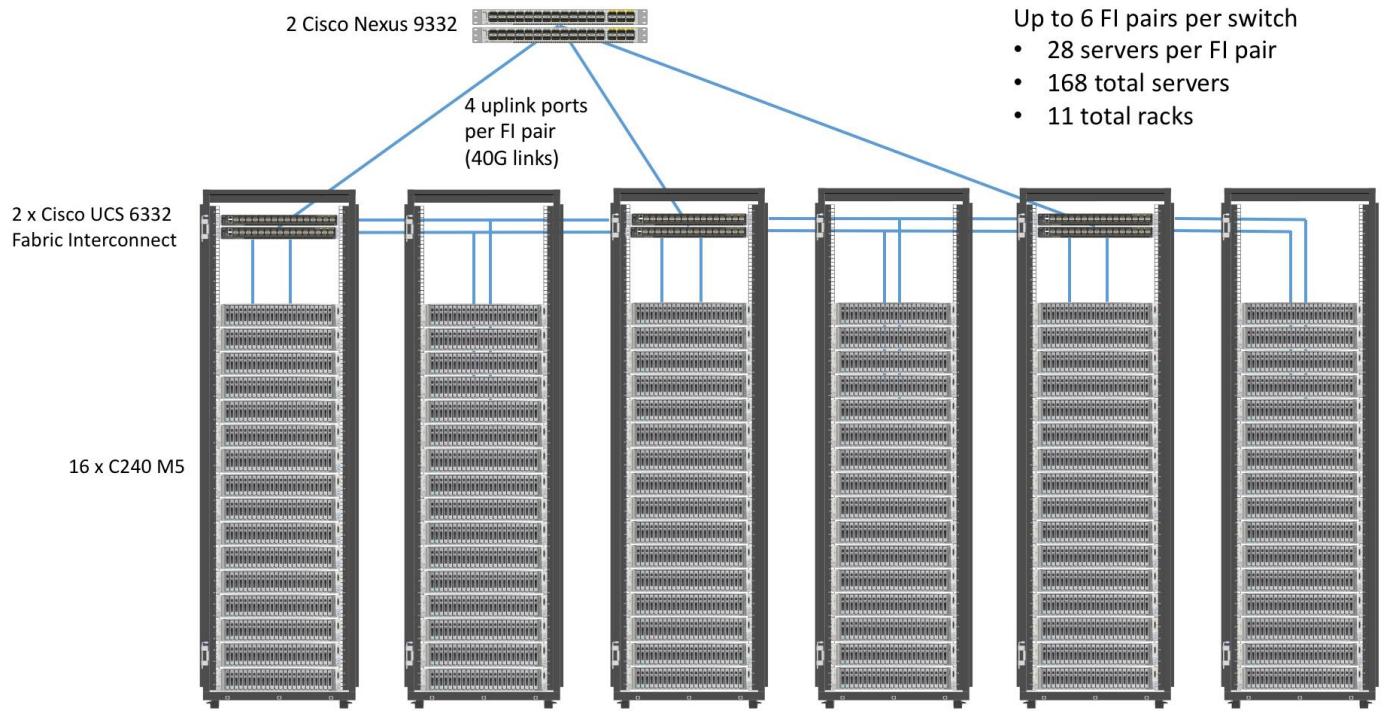
Figure 2 28 Node High Performance Cluster Configuration with additional GPU capacity



Note: Power requirements per rack must be calculated as the exact values will change based on the power needs of the GPUs.

Scaling the Solution

Figure 3 shows how to scale the solution. Each pair of Cisco UCS 6332 Fabric Interconnects has 28 Cisco UCS C240 M5 servers connected to it. This allows for four uplinks from each Fabric Interconnect to the Cisco Nexus 9332 switch. Six pairs of 6332 FI's can connect to a single switch with four uplink ports each. With 28 servers per FI, a total of 168 servers can be supported. Additionally, the can scale to thousands of nodes with the Nexus 9500 series family of switches.

Figure 3 Scaling the Solution

Cloudera Data Science Workbench

Cloudera Data Science Workbench (CDSW) is a web application that allows data scientists to use a variety of open source languages and libraries to directly and securely access the data in the Hadoop cluster. Direct access to the big data cluster means no more working with small subsets of the data on desktop systems; no sampling is required as the entire data set is available for use directly by the user. Further, users are not restricted to a single environment. Many popular open source libraries and languages are supported, including R, Python and Scala, which means users become productive faster with no need for retraining and no time lost learning a new programming language.

The screenshot shows the Cloudera Data Science Workbench interface. On the left sidebar, there are icons for Projects, Jobs, Sessions, and Settings. The main dashboard displays two sessions running and zero jobs running. A central area shows a progress bar for vCPU usage at 2.00 out of 4. Below this, a section titled 'Projects' lists several items: 'Python Template Project' (last worked on just now), 'Python Visualizations' (last worked on just now), 'Air Quality San Francisco' (last worked on just now), 'GitHub Data' (last worked on 1 minute ago), 'Yahoo Search Data' (last worked on 2 minutes ago), and 'PySpark Tests'. To the right, a terminal window runs an R script named 'analysis.r' which reads a CSV file ('Boston.csv') and performs some basic data summary and plotting. A histogram titled 'Boston Median Housing Price' is shown, with the x-axis labeled 'BostonMedv' and the y-axis labeled 'count'. The histogram shows a distribution centered around 15-20.

CDSW is addressing the key challenge that every team or user may require a different language, library or framework in order to be productive while the organization requires reproducibility and collaboration. By making the entire set of data in the cluster available to the user, CSDW eliminates the problem that what works on small samples or extracts of the data on a user's desktop computer may not scale across a large cluster. Cloudera Data Science Workbench gives data scientists the flexibility and simplicity they need to be productive and innovative at scale.

In addition, CDSW enables seamless access to high-performance processors in the form of GPUs. CSDW makes use of lightweight container architecture to rapidly and securely provide the environment and resources to the users.

Cloudera Data Science Workbench is directly aimed at helping data scientists build and test new analyses and analytics projects as quickly as possible in secure manner even in large scale environments. This flexibility improves the efficiency of the exploration process, a key requirement to meet in order to move rapidly from idea to answer. Most analytics problems, especially those with transformative power, are not standard analyses and require advanced models and iterative methods. Experimentation and innovation are the heart and soul of data science, but security is needed for compliance and governance.

Data has become one of the most strategic assets in the organization. Leveraging the data to drive the business forward is the primary motivation for building an enterprise data hub to support advanced analytics. Typically, when forced to make a choice between the security of the data and the flexibility to access it, security wins locking away the data from the people who most need it. CDSW address this issue by providing full authentication and access controls against data in the cluster, including complete Kerberos integration. It offers data science teams per-project isolation and reproducibility with no effort.

Cloudera Data Science Workbench allows you to automate analytics workloads with a built-in job and pipeline scheduling system that supports real-time monitoring, job history, and email alerts. Jobs are created and can be configured to run on a recurring schedule, as well as providing alerts for successful and failed runs. Multiple jobs can be scheduled together to create an automated pipeline; e.g., the first job performs data acquisition, the next data cleansing, then analytics, and so on.

Collaboration and sharing of results are implemented via project sharing (either globally or to specific users, and project forking. To share results, CSDW enables publishing output for viewing via a browser, and even makes the console log itself

available for viewing both during and after the run. Cloudera Data Science Workbench is a web application. It has no desktop footprint making it very easy to administer and maintain.

NVIDIA GPU

Graphics Processing Units, or GPUs, are specialized processors designed to render images, animation and video for computer displays. They perform this task by running many operations simultaneously. While the number and kinds of operations they can do are limited, they make up for it by being able to run many thousands in parallel. As the graphics capabilities of GPUs increased, it soon became apparent that the massive parallelism of GPUs could be put to other uses beside rendering graphics.

NVIDIA® GPU used in this document, NVIDIA Tesla® V100 is advanced data center GPU built to accelerate AI, HPC, and graphics. It's powered by NVIDIA Volta architecture, comes in 16 and 32 GB configurations.

NVIDIA GPUs bring two key advantages to the table. First, they make possible solutions that were simply not computationally possible before. Second, by providing the same processing power as scores of traditional CPUs they reduce the requirements for rack space, power, networking and cooling in the data center.

Technology Overview

Cisco UCS Integrated Infrastructure for Big Data and Analytics

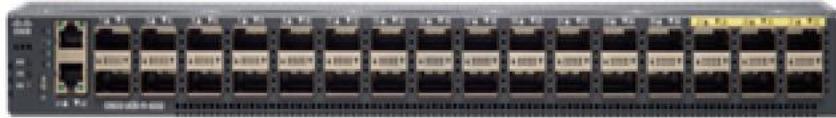
The Cisco UCS Integrated Infrastructure for Big Data and Analytics solution for Cloudera is based on [Cisco UCS Integrated Infrastructure for Big Data and Analytics](#), a highly scalable architecture designed to meet a variety of scale-out application demands with seamless data integration and management integration capabilities built using the components described in this section.

Cisco UCS 6300 Series Fabric Interconnects

Cisco UCS 6300 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.

The Cisco UCS 6300 series Fabric interconnects are a core part of Cisco UCS, providing low-latency, lossless 10 and 40 Gigabit Ethernet, Fiber Channel over Ethernet (FCoE), and Fiber Channel functions with management capabilities for the entire system. All servers attached to Fabric interconnects become part of a single, highly available management domain.

Figure 4 Cisco UCS 6332 UP 32 -Port Fabric Interconnect



Cisco UCS C-Series Rack-Mount Servers

The Cisco UCS C240 M5 Rack-Mount Server (Figure 5) is a 2-socket, 2-Rack-Unit (2RU) rack server offering industry-leading performance and expandability. It supports a wide range of storage and I/O-intensive infrastructure workloads, from big data and analytics to collaboration. Cisco UCS C-Series Rack Servers can be deployed as standalone servers or as part of a Cisco Unified Computing System (Cisco UCS) managed environment to take advantage of Cisco's standards-based unified computing innovations that help reduce customers' Total Cost of Ownership (TCO) and increase their business agility.

In response to ever-increasing computing and data-intensive real-time workloads, the enterprise-class Cisco UCS C240 M5 server extends the capabilities of the Cisco UCS portfolio in a 2RU form factor. It incorporates the Intel® Xeon® Scalable processors, supporting up to 20 percent more cores per socket, twice the memory capacity, and five times more

Non-Volatile Memory Express (NVMe) PCI Express (PCIe) Solid-State Disks (SSDs) compared to the previous generation of servers. These improvements deliver significant performance and efficiency gains that will improve your application performance. The Cisco UCS C240 M5 delivers outstanding levels of storage expandability with exceptional performance, along with the following:

- Latest Intel Xeon Scalable CPUs with up to 28 cores per socket
- Up to 24 DDR4 DIMMs for improved performance

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

Figure 5 Cisco UCS C240 M5 Rack-Mount Server



The Cisco UCS C480 M5 Rack-Mount Server is a storage- and I/O-optimized enterprise-class rack server that delivers industry-leading performance for in-memory databases, big data analytics, virtualization, Virtual Desktop Infrastructure (VDI), and bare-metal applications. The Cisco UCS C480 M5 (Figure 6) delivers outstanding levels of expandability and performance for standalone or Cisco Unified Computing System™ (Cisco UCS) managed environments in a 4RU form-factor. And because of its modular design, you pay for only what you need. It offers these capabilities:

- Latest Intel® Xeon® Scalable processors with up to 28 cores per socket and support for two-or four-processor configurations
- 2666-MHz DDR4 memory and 48 DIMM slots for up to 6 TeraBytes (TB) of total memory
- 12 PCI Express (PCIe) 3.0 slots
 - Six x8 full-height, full length slots
 - Six x16 full-height, full length slots
- Flexible storage options with support up to 32 Small-Form-Factor (SFF) 2.5-inch, SAS, SATA, and PCIe NVMe disk drives
- Cisco® 12-Gbps SAS Modular RAID Controller in a dedicated slot
- Internal Secure Digital (SD) and M.2 boot options
- Dual embedded 10 Gigabit Ethernet LAN-On-Motherboard (LOM) ports

Figure 6 Cisco UCS C480 M5 Rack-Mount Server



Cisco UCS Virtual Interface Cards (VICs)

Cisco UCS Virtual Interface Cards (VIC) are unique to Cisco. Cisco UCS Virtual Interface Cards incorporate next-generation converged network adapter (CNA) technology from Cisco, and offer dual 10- and 40-Gbps ports designed for use with Cisco UCS 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 1387 offers dual-port Enhanced Quad Small Form-Factor Pluggable (QSFP+) 40 Gigabit Ethernet and Fiber Channel over Ethernet (FCoE) in a modular-LAN-on-motherboard (mLOM) form factor. The mLOM slot can be used to install a Cisco VIC without consuming a PCIe slot providing greater I/O expandability.

Figure 7 Cisco UCS VIC 1387



Cisco UCS Manager

Cisco UCS Manager resides within the Cisco UCS 6300 Series Fabric Interconnect. It makes the system self-aware and self-integrating, managing all of the system components as a single logical entity. Cisco UCS Manager can be accessed through an intuitive graphical user interface (GUI), 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.

NVIDIA CUDA

GPUs are very good at running the same operation on different data simultaneously. This is often referred to as single instruction, multiple data, or SIMD. This is exactly what's needed to render graphics but many other computing problems can benefit from this approach. As a result, NVIDIA created CUDA. CUDA is a parallel computing platform and programming model that makes it possible to use a GPU for many general purpose computing tasks via commonly used programming languages like C and C++.

In addition to the general-purpose computing capabilities that CUDA enables there is also a special CUDA library for deep learning called the CUDA Deep Neural Network library, or cuDNN. cuDNN makes it easier to implement deep machine learning architectures that take full advantage of the GPU's capabilities.

Cloudera (CDH 5.13.0)

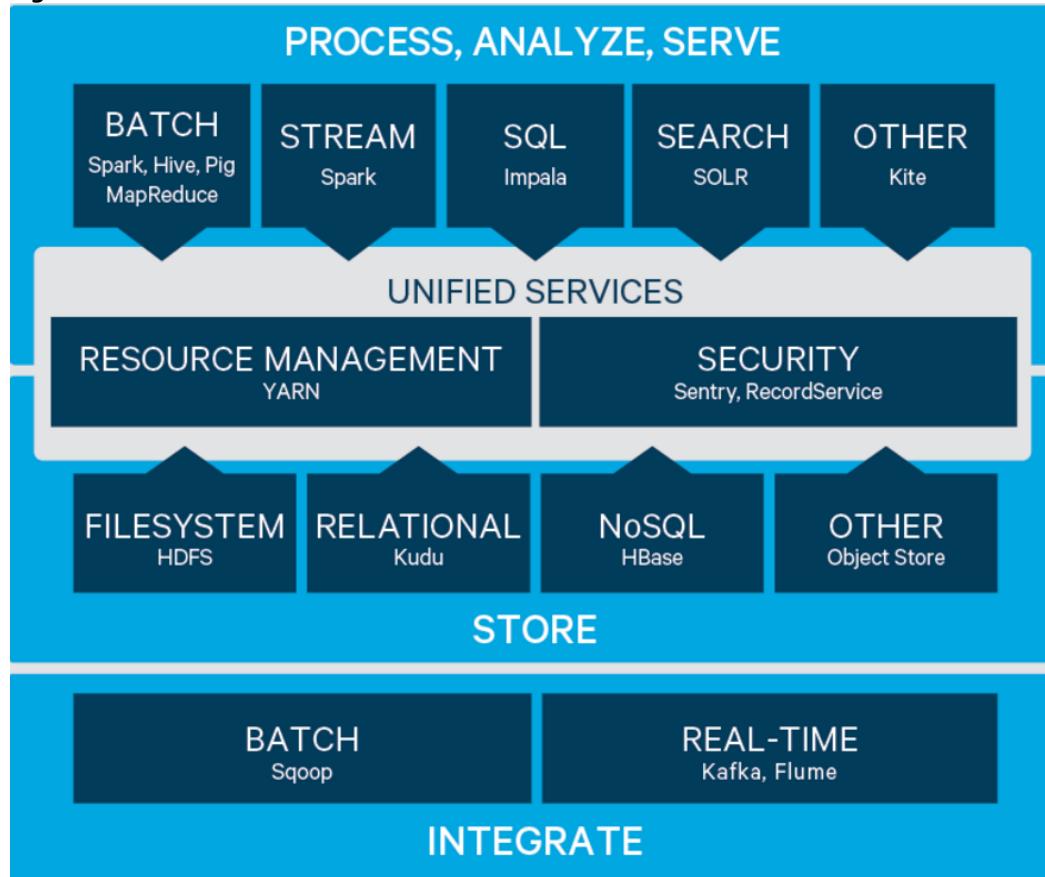
Built on the transformative Apache Hadoop open source software project, Cloudera Enterprise is a hardened distribution of Apache Hadoop and related projects designed for the demanding requirements of enterprise customers. Cloudera is the leading contributor to the Hadoop ecosystem, and has created a rich suite of complementary open source projects that are included in Cloudera Enterprise.

All the integration and the entire solution is thoroughly tested and fully documented. By taking the guesswork out of building out a Hadoop deployment, CDH gives a streamlined path to success in solving real business problems.

Cloudera Enterprise with Apache Hadoop is:

- **Unified** – one integrated system, bringing diverse users and application workloads to one pool of data on common infrastructure; no data movement required
- **Secure** – perimeter security, authentication, granular authorization, and data protection
- **Governed** – enterprise-grade data auditing, data lineage, and data discovery
- **Managed** – native high-availability, fault-tolerance and self-healing storage, automated backup and disaster recovery, and advanced system and data management
- **Open** – Apache-licensed open source to ensure both data and applications remain copy righted, and an open platform to connect with all of the existing investments in technology and skills.

Figure 8 Cloudera Data Hub



Cloudera provides a scalable, flexible, integrated platform that makes it easy to manage rapidly increasing volumes and varieties of data in any enterprise. Industry-leading Cloudera products and solutions enable to deploy and manage Apache Hadoop and related projects, manipulate and analyze data, and keep that data secure and protected.

Cloudera provides the following products and tools:

- **CDH**—The Cloudera distribution of Apache Hadoop and other related open-source projects, including Spark. CDH also provides security and integration with numerous hardware and software solutions.
- **Apache Spark**—An integrated part of CDH and supported with Cloudera Enterprise, Spark is an open standard for flexible in-memory data processing for batch, real time and advanced analytics. Via the one platform Cloudera is committed to adopting Spark as the default data execution engine for analytic workloads.

- [Cloudera Manager](#)—A sophisticated application used to deploy, manage, monitor, and diagnose issues with CDH deployments. Cloudera Manager provides the Admin Console, a web-based user interface that makes administration of any enterprise data simple and straightforward. It also includes the Cloudera Manager API, which can be used to obtain cluster health information and metrics, as well as configure Cloudera Manager.
- [Cloudera Navigator](#)—An end-to-end data management tool for the CDH platform. Cloudera Navigator enables administrators, data managers, and analysts to explore the large amounts of data in Hadoop. The robust auditing, data management, lineage management, and life cycle management in Cloudera Navigator allow enterprises to adhere to stringent compliance and regulatory requirements.

Cloudera Data Science Workbench

Cloudera Data Science Workbench (CDSW) is a web application that allows data scientists to use a variety of open source languages and libraries to directly and securely access the data in the Hadoop cluster. Direct access to the big data cluster means no more working with small subsets of the data on desktop systems; no sampling is required as the entire data set is available for use directly by the user. Further, users are not restricted to a single environment. Many popular open source libraries and languages are supported, including R, Python and Scala, as well as all of the ML/DL frameworks such as TensorFlow, Theano, Pytorch, etc. In addition, CDSW enables access to available GPU resources for deep learning workloads.

Docker Containers

Cloudera Data Science Workbench makes use of container technology. Containers are conceptually similar to virtual machines, but instead of virtualizing the hardware, a container virtualizes the operating system. With a VM there is an entire operating system sitting on top of the hypervisor. Containers dispense with this time-consuming and resource hungry requirement by sharing the host system's kernel. As a result, a container is far smaller, and its lightweight nature means they can be instantiated quickly. In fact, they can be instantiated so quickly that new application architectures are possible.

Docker is an open-source project based on Linux containers. It uses Linux kernel features like namespaces and control groups to create containers. These features are not new, but Docker has taken these concepts and improved them in the following ways:

- Ease of use: Docker makes easier for anyone—developers, systems admins, architects and others—to take advantage of containers in order to quickly build and test portable applications. It allows anyone to package an application on their development system, which can then run *unmodified* on any cloud or bare metal server. The basic idea is to create a “build once, run anywhere” system.
- Speed: Docker containers are very fast with a small footprint. Ultimately, containers are just sandboxed environments running on the kernel so they take up few resources. You can create and run a Docker container in seconds. Compare this to a VM which takes much longer because it has to boot up a full virtual operating system every time.
- Modularity: Docker makes it easy to take an application and breaks its functionality into separate individual containers. These containers can then be spun up and run as needed. This is particularly useful for cases where an application needs to hold and lock a particular resource, like a GPU, and then release it once it’s done using it. Modularity also enables each component, i.e., container to be updated independently.
- Scalability: modularity enables scalability. With different parts of the system running in different containers it becomes possible, and with Docker, it becomes easy to connect these containers together to create an application, which can then be scaled out as needed.

Kubernetes

Applications built using container technology provide a great deal of flexibility in terms of their architecture, deployment and scaling. Since containers provide VM-like separation of concerns but with far less overhead they allow system developers to package different services of the same application into separate containers. These containers can then be deployed in a very flexible manner including across clusters of physical and virtual machines. This builds the ability to scale directly into the application architecture. This ability in turn requires a tool to aid in deploying, managing and scaling container-based applications.

Kubernetes is an open source project specifically designed for deploying and managing multi-container applications at scale. Kubernetes automates and simplifies the following tasks:

- Deploying multi-container applications. With the application split into separate containers for different services, Kubernetes manages the deployment of the containers both at initial startup and in real-time as needed by the application.
- Scaling containers. Applications need to spin up and down containers to suit demand, to balance incoming load, and make better use of physical resources. Kubernetes provides the mechanisms for doing these things in a completely automated way.
- Updating applications. One advantage of container-based application development is the ability to independently change, improve and fix individual containers. Kubernetes has mechanisms for allowing graceful updates to new versions of container images, including rollbacks if something does not go as planned.

Kubernetes manages application status and any replication and load balancing needs. It also handles hardware resource allocation including GPUs. Kubernetes also has facilities for maximizing the use of hardware resources including memory, storage I/O, and network bandwidth. Applications can have soft and hard limits set on their resource usage. For example, many small applications that use minimal resources can be run together on the same hardware while resource hungry applications can be placed on different hardware and scale out as needed.

Solution Design

Requirements

This CVD describes architecture and deployment procedures for Cloudera (CDH 5.13.0) and Cloudera Data Science Workbench (CDSW 1.3.0) on a 28-node cluster based on Cisco UCS Integrated Infrastructure for Big Data and Analytics. The solution goes into detail configuring CDH 5.13.0 on the infrastructure, as well as the complete installation and configuration of CDSW 1.3.0 and all of its dependencies.

The cluster configuration consists of the following:

- Two Cisco UCS 6332UP Fabric Interconnects
- 28 UCS C240 M5 Rack-Mount servers
- 8 NVIDIA GPU
- Two Cisco R42610 standard racks
- Four Vertical Power distribution units (PDUs) (Country Specific)

Rack and PDU Configuration

Each rack consists of two vertical PDUs. The first rack consists of two Cisco UCS 6332UP Fabric Interconnects, 16 Cisco UCS C240 M5 Rack Servers connected to each of the vertical PDUs for redundancy; thereby, ensuring availability during power source failure. The second rack consists of 12 Cisco UCS C240 M5 Servers connected to each of the vertical PDUs for redundancy; thereby, ensuring availability during power source failure, similar to the first rack.



Note: Please contact your Cisco representative for country specific information.

Table 3 Describes the rack configurations.

Table 3 Rack Configurations

Cisco	First Rack	Cisco	Second Rack
42URack		42URack	
42	Cisco UCS FI 6332UP	42	Unused
41	Cisco UCS FI 6332UP	41	Unused
40	Unused	40	Unused
39	Unused	39	Unused
38	Unused	38	Unused
37	Unused	37	Unused
36	Unused	36	Unused
35		35	Unused

Cisco	First Rack	Cisco	Second Rack
34	Unused	34	Unused
33		33	
32	Cisco UCS C240 M5	32	Unused
31		31	
30	Cisco UCS C240 M5	30	Unused
29		29	
28	Cisco UCS C240 M5	28	Unused
27		27	
26	Cisco UCS C240 M5	26	Unused
25		25	
24	Cisco UCS C240 M5	24	Cisco UCS C240 M5
23		23	
22	Cisco UCS C240 M5	22	Cisco UCS C240 M5
21		21	
20	Cisco UCS C240 M5	20	Cisco UCS C240 M5
19		19	
18	Cisco UCS C240 M5	18	Cisco UCS C240 M5
17		17	
16	Cisco UCS C240 M5	16	Cisco UCS C240 M5
15		15	
14	Cisco UCS C240 M5	14	Cisco UCS C240 M5
13		13	
12	Cisco UCS C240 M5	12	Cisco UCS C240 M5
11		11	
10	Cisco UCS C240 M5	10	Cisco UCS C240 M5
9		9	
8	Cisco UCS C240 M5	8	Cisco UCS C240 M5

Cisco	First Rack	Cisco	Second Rack
7		7	with 2x NVIDIA GPUs
6	Cisco UCS C240 M5	6	Cisco UCS C240 M5
5		5	with 2x NVIDIA GPUs
4	Cisco UCS C240 M5	4	Cisco UCS C240 M5
3		3	with 2x NVIDIA GPUs
2	Cisco UCS C240 M5	2	Cisco UCS C240 M5
1		1	with 2x NVIDIA GPUs

Port Configuration on Fabric Interconnects

Port Type	Port Number
Network	29-32
Server	1-28

Server Configuration and Cabling for Cisco UCS C240 M5

The Cisco UCS C240 M5 rack server is equipped with 2 x Intel Xeon Processor Scalable Family 6132 (2 x 14 cores, 2.6 GHz), 192 GB of memory, Cisco UCS Virtual Interface Card 1337, Cisco 12-Gbps SAS Modular Raid Controller with 4-GB FBWC, 26 x 1.8 TB 10K rpm SFF SAS HDDs or 12 x 1.6 TB Enterprise Value SATA SSDs, M.2 with 2 x 240-GB SSDs for Boot.

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

https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c-series_integration/ucsm3-2/b_C-Series-Integration_UCSM3-2/b_C-Series-Integration_UCSM3-2_chapter_010.html?bookSearch=true

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

https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c-series_integration/ucsm3-2/b_C-Series-Integration_UCSM3-2/b_C-Series-Integration_UCSM3-2_chapter_010.html?bookSearch=true

Software Distributions and Versions

The software distributions required versions are listed below.

Cloudera (CDH 5.13.0)

The Cloudera Distribution for Apache Hadoop version used is 5.13.0. For more information visit www.cloudera.com.

Red Hat Enterprise Linux (RHEL)

The operating system supported is Red Hat Enterprise Linux 7.4. For more information visit <http://www.redhat.com>.

Software Versions

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

Table 4 Software Versions

Layer	Component	Version or Release
Compute	Cisco UCS C240-M5	C240M5.3.1.2a.0.09
Network	Cisco UCS 6332	UCS 3.2(2b) A
	Cisco UCS VIC1387 Firmware	4.2.2(a)
	Cisco UCS VIC1387 Driver	2.3.0.44
Storage	SAS Expander	65.02.13.00
	Cisco 12G Modular Raid controller	50.1.0-07.26
	LSI MegaRAID SAS Driver	07.703.06.00
Software	Red Hat Enterprise Linux Server	7.4 (x86_64)
	Cisco UCS Manager	3.2(2b)
	CDH	5.13.0
	CDSW	1.3.0
GPU	CUDA	8.1
	GPU Driver	390



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

<https://software.cisco.com/download/home/283862063/type/283853158/release/3.1%25283%2529.>



Note: The Latest Supported RAID controller Driver is already included with the RHEL 7.4 operating system.



Note: Cisco UCS C240 M5 Rack Servers with Intel Scalable Processor Family CPUs are supported from Cisco UCS firmware 3.2 onwards.

Fabric Configuration

This section provides details for configuring a fully redundant, highly available Cisco UCS 6332 fabric configuration.

- Initial setup of the Fabric Interconnect A and B

- Connect to Cisco UCS Manager using virtual IP address or using the web browser
- Launch Cisco UCS Manager.
- Enable server, uplink and appliance ports.
- Start discovery process.
- Create pools and polices for service profile template
- Create Service Profile template and 28 Service profiles
- Associate Service Profiles to servers

Performing Initial Setup of Cisco UCS 6332 Fabric Interconnects

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

Configure Fabric Interconnect A

1. Connect to the console port on the first Cisco UCS 6332 Fabric Interconnect.
2. At the prompt to enter the configuration method, enter `console` to continue.
3. If asked to either perform a new setup or restore from backup, enter `setup` to continue.
4. Enter `y` to continue to set up a new Fabric Interconnect.
5. Enter `y` to enforce strong passwords.
6. Enter the password for the admin user.
7. Enter the same password again to confirm the password for the admin user.
8. When asked if this fabric interconnect is part of a cluster, answer `y` to continue.
9. Enter `A` for the switch fabric.
10. Enter the cluster name for the system name.
11. Enter the Mgmto IPv4 address.
12. Enter the Mgmto 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 6332 Fabric Interconnect.
2. When prompted to enter the configuration method, enter `console` to continue.
3. The installer detects the presence of the partner Fabric Interconnect and adds this fabric interconnect to the cluster. Enter `y` to continue the installation.
4. Enter the admin password that was configured for the first Fabric Interconnect.
5. Enter the Mgmt IP 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 6332 Series Fabric Interconnect, refer to:

https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/ucs-manager/GUI-User-Guides/Getting-Started/3-2/b_UCSM_Getting_Started_Guide_3_2/b_UCSM_Getting_Started_Guide_3_2_chapter_0100.html

Logging Into Cisco UCS Manager

To log into Cisco UCS Manager, complete the following steps:

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

Upgrading Cisco UCS Manager Software to Version 3.2(2b)

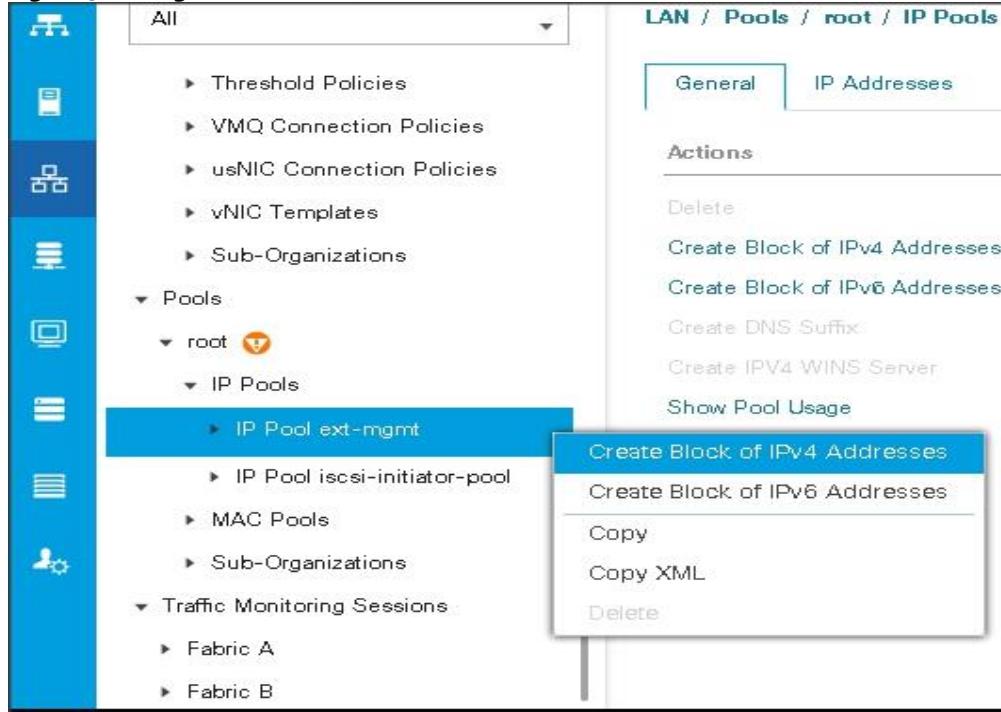
This document assumes the use of UCS 3.2(2b). Refer to Cisco UCS 3.2 Release (upgrade the Cisco UCS Manager software and UCS 6332 Fabric Interconnect software to version 3.2(2b)). Also, make sure the UCS C-Series version 3.2(2b) software bundles are installed on the Fabric Interconnects.

Adding a Block of IP Addresses for KVM Access

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

1. Select the `LAN` tab at the top of the left window.
2. Select `Pools > root > IpPools > Ip Pool ext-mgmt`.
3. Right-click `IP Pool ext-mgmt`.

4. Select Create Block of IPv4 Addresses.

Figure 9 Adding a Block of IPv4 Addresses for KVM Access Part 1

- Enter the starting IP address of the block and number of IPs needed, as well as the subnet and gateway information.

Figure 10 Adding Block of IPv4 Addresses for KVM Access Part 2

The screenshot shows the 'Create Block of IPv4 Addresses' dialog box. It contains fields for 'From' (10.13.1.11), 'Size' (28), 'Subnet Mask' (255.255.255.0), 'Default Gateway' (10.13.1.1), 'Primary DNS' (0.0.0.0), and 'Secondary DNS' (0.0.0.0).

- Click OK to create the IP block.
- Click OK in the message box.

Enabling Uplink Ports

To enable uplinks ports, complete the following steps:

- Select the Equipment tab on the top left of the window.
- Select Equipment > Fabric Interconnects > Fabric Interconnect A (primary) > Fixed Module.
- Expand the Unconfigured Ethernet Ports section.
- Select port 29-32 that is connected to the uplink switch, right-click, then select Reconfigure > Configure as Uplink Port.

5. Select Show Interface and select 40GB for Uplink Connection.
6. A pop-up window appears to confirm your selection. Click Yes then OK to continue.
7. Select Equipment > Fabric Interconnects > Fabric Interconnect B (subordinate) > Fixed Module.
8. Expand the Unconfigured Ethernet Ports section.
9. Select port number 29–32, which is connected to the uplink switch, right-click, then select Reconfigure > Configure as Uplink Port.
10. Select Show Interface and select 40GB for Uplink Connection.
11. A pop-up window appears to confirm your selection. Click Yes then OK to continue.

Figure 11 Enabling Uplink Ports Part1

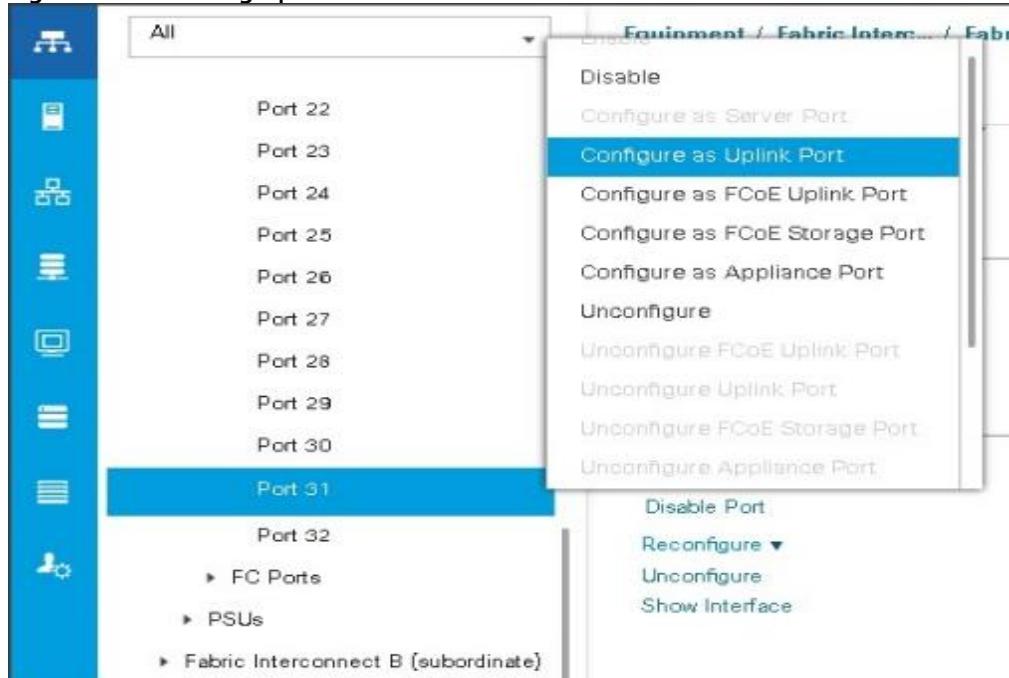
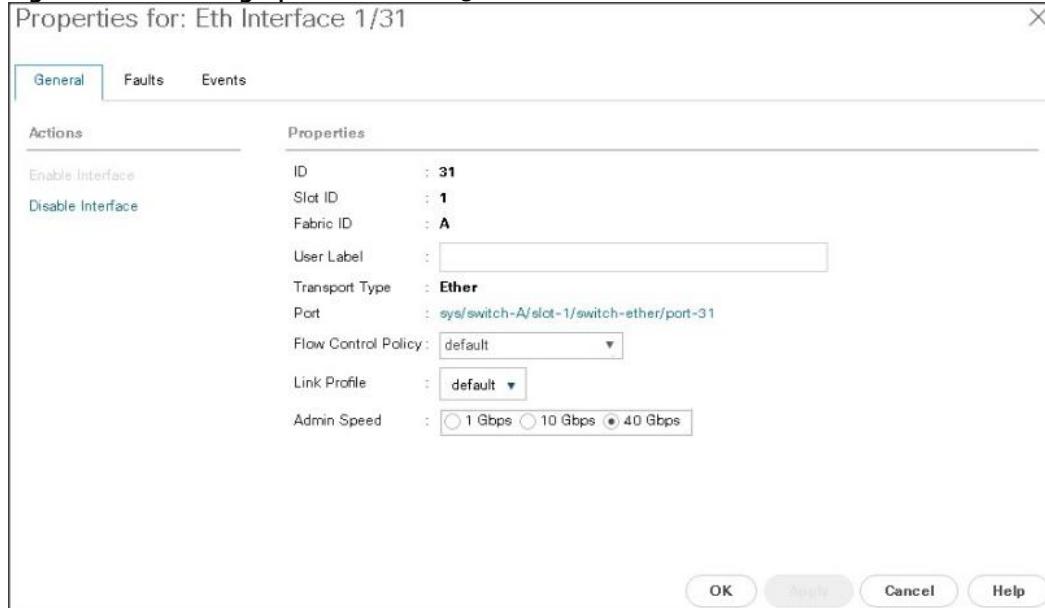


Figure 17 Enabling Uplink Ports Part 2**Figure 18** Enabling Uplink Ports Part 3

Configuring VLANs

VLANs are configured as shown in Table 5.

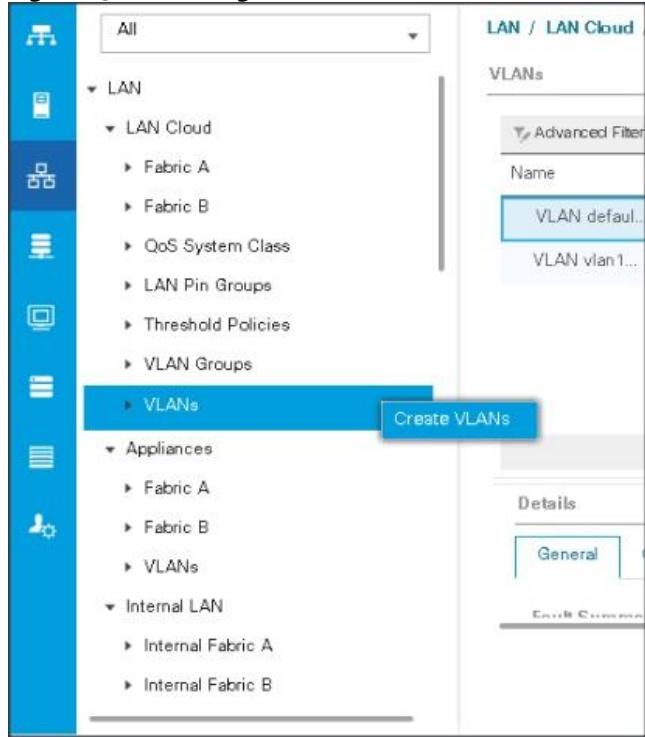
Table 5 VLAN Configurations

VLAN	NIC Port	Function
VLAN13	eth0	Data

The NIC will carry the data traffic from VLAN13. A single vNIC is used in this configuration and the Fabric Failover feature in Fabric Interconnects will take care of any physical port down issues. It will be a seamless transition from an application perspective.

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

1. Select the LAN tab in the left pane in the UCSM GUI.
2. Select LAN > LAN Cloud > VLANs.
3. Right-click the VLANs under the root organization.
4. Select Create VLANs to create the VLAN.

Figure 19 Creating a VLAN

5. Enter **vlan13** for the **VLAN Name**.
6. Keep **multicast policy** as **<not set>**.
7. Select **Common/Global** for **vlan13**.
8. Enter **13** in the **VLAN IDs** field for the **Create VLAN IDs**.
9. Click **OK** and then, click **Finish**.
10. Click **OK** in the success message box.

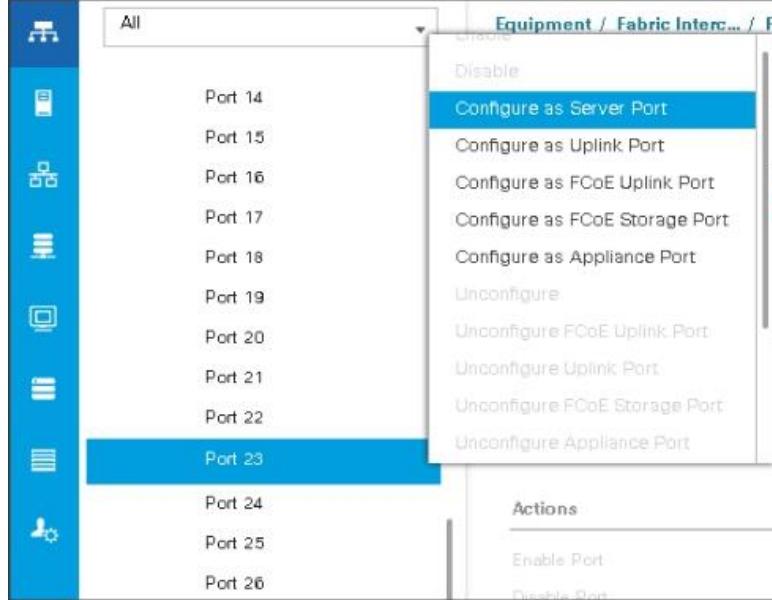
Figure 20 Creating VLAN for Data

11. Click OK and then, click Finish.

Enabling Server Ports

To enable server ports, complete the following steps:

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 21 Enabling Server Ports

After the Server Discovery, Port 29-32 will be a Network Port and 1-28 will be Server Ports.

Figure 22 Ports status after the Server Discover

Slot	Aggr. Port ID	Port ID	MAC	If Role	If Type	Overall Status	Admin State
1	0	1	70:7D:B9:F3:60...	Server	Physical	Up	Enabled
1	0	2	70:7D:B9:F3:60...	Server	Physical	Up	Enabled
1	0	3	70:7D:B9:F3:60...	Server	Physical	Up	Enabled
1	0	4	70:7D:B9:F3:60...	Server	Physical	Up	Enabled
1	0	5	70:7D:B9:F3:60...	Server	Physical	Up	Enabled
1	0	6	70:7D:B9:F3:60...	Server	Physical	Up	Enabled
1	0	7	70:7D:B9:F3:60...	Server	Physical	Up	Enabled
1	0	8	70:7D:B9:F3:60...	Server	Physical	Up	Enabled
1	0	9	70:7D:B9:F3:60...	Server	Physical	Up	Enabled
1	0	10	70:7D:B9:F3:60...	Server	Physical	Up	Enabled

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.

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

1. Click Quick Action icon on the top right corner in the right pane in the Cisco 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.

The screenshot shows the Cisco UCS Manager interface. At the top, there's a navigation bar with icons for Home, Logout, Refresh, and Help. Below it is a toolbar with various buttons. On the left, there's a sidebar with icons for Servers, Storage, Network, and Applications. The main area shows a table of 'Service Profiles' with columns for Name, User Label, Overall Status, Assoc State, and Server. Three service profiles are listed: 'Service Profile ucs-1', 'Service Profile ucs-10', and 'Service Profile ucs-11', all in an 'OK' status and associated with 'sys/rack-unit-1', 'sys/rack-unit-10', and 'sys/rack-unit-11' respectively. A context menu is open on the right side of the screen, with a red arrow pointing to the 'Create Organization' option. The menu also includes 'Create Service Profile', 'Create Service Profile (expert)', 'Create VLANs', 'Create VSAN', and 'Create User'. Below the main interface, a 'Create Organization' dialog box is open, showing fields for 'Name' (containing 'ucs') and 'Description' (empty), with 'OK' and 'Cancel' buttons at the bottom.

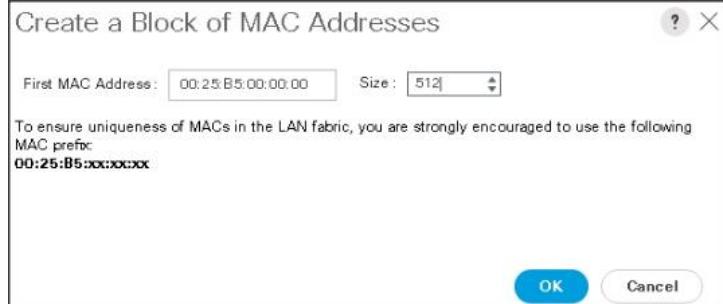
Creating MAC Address Pools

To create MAC address pools, complete the following steps:

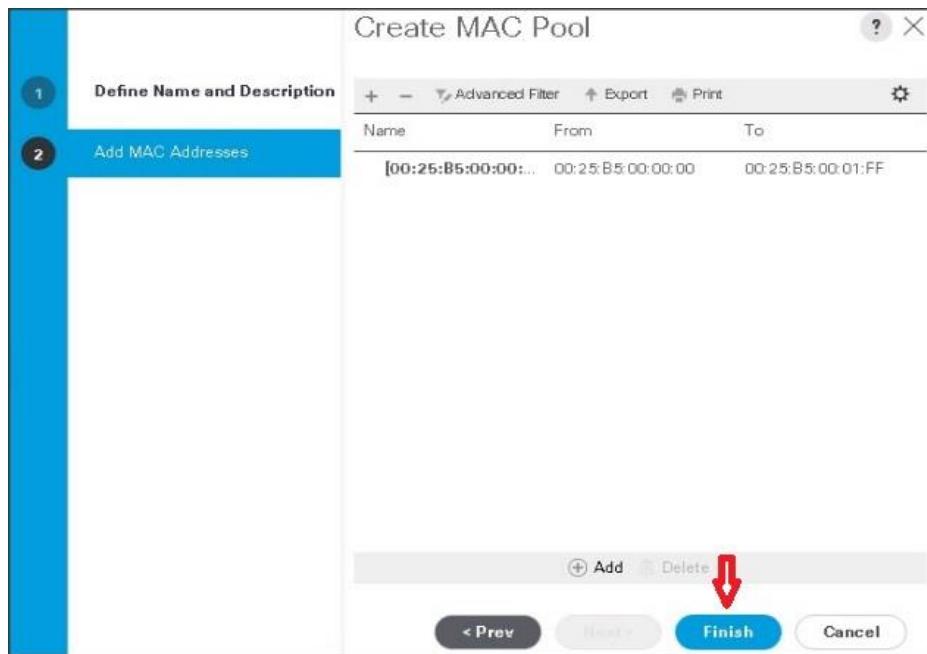
1. Select the LAN tab on the left of the window.
2. Select Pools > root > MAC Pools
3. Right-click MAC Pools under the root organization.
4. Select Create MAC Pool to create the MAC address pool. Enter ucs for the name of the MAC pool.
5. (Optional) Enter a description of the MAC pool.
6. Select Assignment Order Sequential.
7. Click Next.
8. Click Add.
9. Specify a starting MAC address.

10. Specify a size of the MAC address pool, which is sufficient to support the available server resources.
11. Click OK.

The screenshot shows the Cisco UCS Manager interface. On the left is a navigation sidebar with various icons and a tree view of organizational structures. The main pane is titled "LAN / Pools / root / MAC Pools". It displays a list of existing MAC pools: "MAC Pool default" and "MAC Pool ucs". A blue box highlights the "Create MAC Pool" button. Below this, a modal window titled "Create MAC Pool" is open. The first step, "Define Name and Description", is selected. It contains fields for "Name" (set to "ucs"), "Description" (empty), and "Assignment Order" (radio buttons for "Default" and "Sequential", with "Sequential" selected). The second step, "Add MAC Addresses", is shown below. At the bottom of the modal are buttons for "Previous", "Next >", "Finish", and "Cancel".

Figure 23 Specifying first MAC Address and Size

12. Click Finish.



13. When the message box displays, click OK.

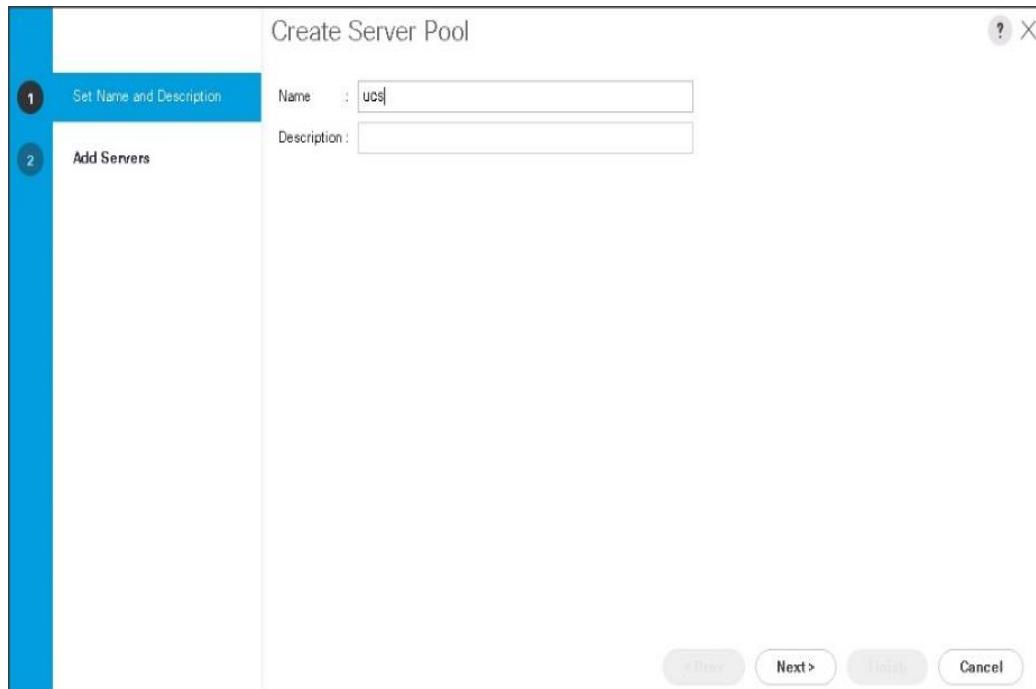


Creating a Server Pool

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.

To configure the server pool within the Cisco UCS Manager GUI, complete the following steps:

1. Select the Servers tab in the left pane in the UCS Manager GUI.
2. Select Pools > root.
3. Right-click the Server Pools.
4. Select Create Server Pool.
5. Enter your required name (ucs) for the Server Pool in the name text box.
6. (Optional) enter a description for the organization.
7. Click Next > to add the servers.



8. Select all the Cisco UCS C240M5 servers to be added to the server pool that was previously created (ucs), then Click >> to add them to the pool.



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

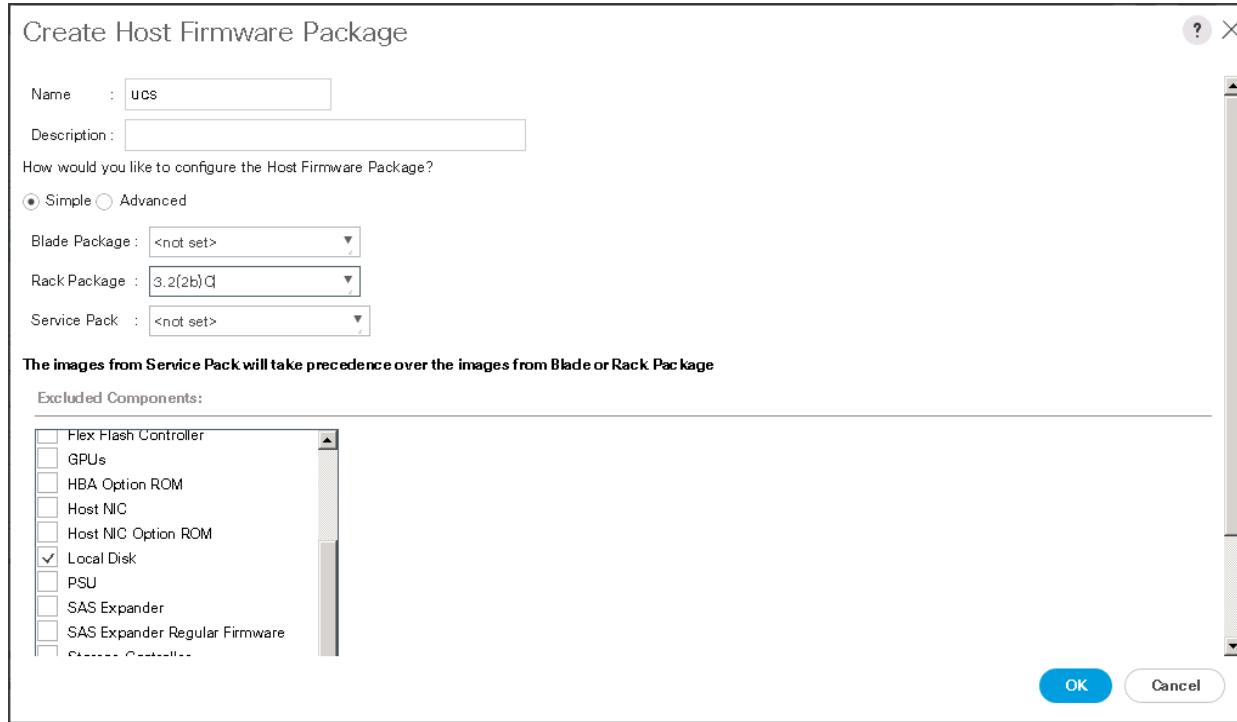
Creating Policies for Service Profile Templates

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, and storage controller properties as applicable.

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

1. Select the Servers tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click Host Firmware Packages.
4. Select Create Host Firmware Package.
5. Enter the required Host Firmware package name (ucs).
6. Select Simple radio button to configure the Host Firmware package.
7. Select the appropriate Rack package that has been installed.
8. Click OK to complete creating the management firmware package
9. Click OK.

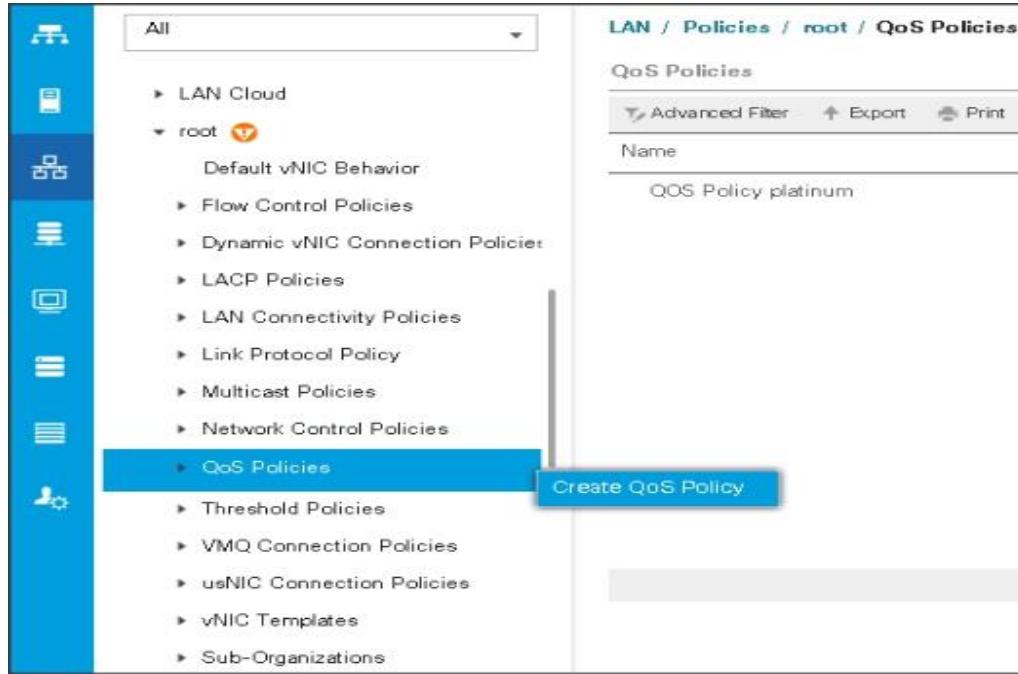


Creating QoS Policies

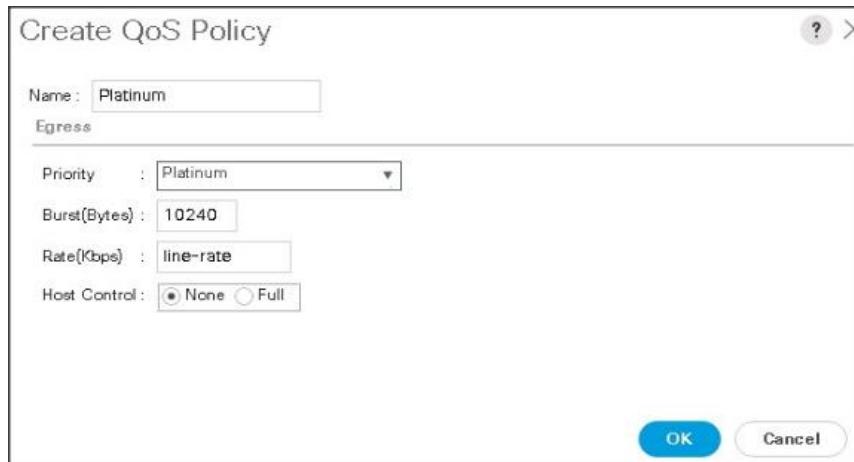
To create the QoS policy for a given server configuration using the Cisco UCS Manager GUI, complete the following steps:

Platinum Policy

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



5. Enter Platinum as the name of the policy.
6. Select Platinum from the drop-down list.
7. Keep the Burst (Bytes) field set to default (10240).
8. Keep the Rate (Kbps) field set to default (line-rate).
9. Keep Host Control radio button set to default (none).
10. When the pop-up window appears, click OK to complete the creation of the Policy.



Setting Jumbo Frames

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

1. Select the LAN tab in the left pane in the Cisco 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 9216 for MTU.
5. Check the Enabled Check box next to Platinum.
6. In the Best Effort row, select none for weight.
7. In the Fiber Channel row, select none for weight.
8. Click Save Changes.
9. Click OK.

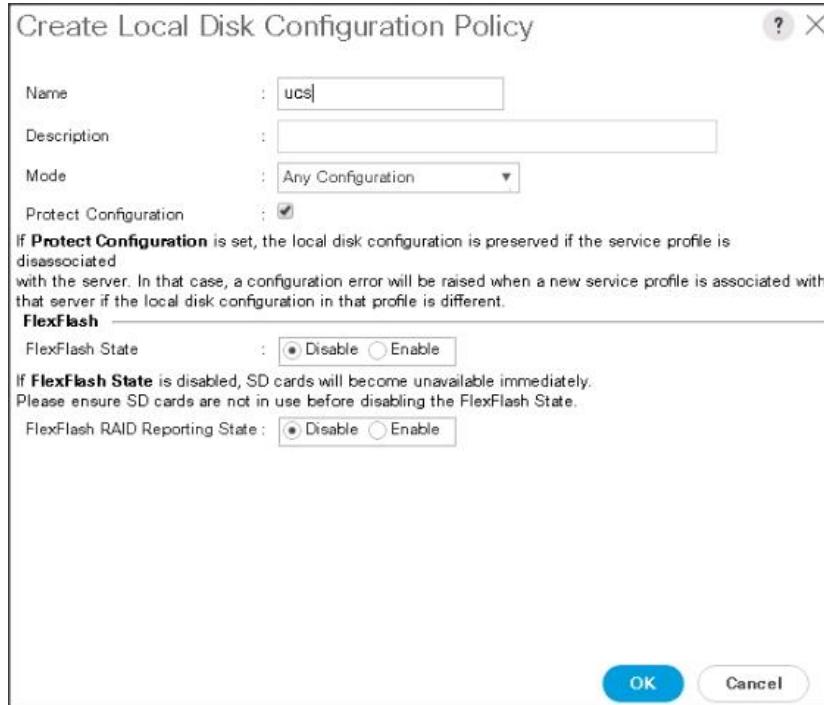
	Priority	Enabled CoS	Packet Drop	Weight	Weight (%)	MTU
Platinum	<input checked="" type="checkbox"/>	5	<input type="checkbox"/>	10	N/A	9216
Gold	<input type="checkbox"/>	4	<input checked="" type="checkbox"/>	9	N/A	normal
Silver	<input type="checkbox"/>	2	<input checked="" type="checkbox"/>	8	N/A	normal
Bronze	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	7	N/A	normal
Best Effort	<input checked="" type="checkbox"/>	Any	<input checked="" type="checkbox"/>	none	50	normal
Fibre	<input checked="" type="checkbox"/>	3	<input type="checkbox"/>	none	50	normal

Creating the Local Disk Configuration Policy

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

1. Select the Servers tab on the left pane in the Cisco UCS Manager GUI.
2. Go to Policies > root.
3. Right-click Local Disk Config Policies.
4. Select Create Local Disk Configuration Policy.
5. Enter ucs as the local disk configuration policy name.
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 manually, and is often error-prone. By creating a BIOS policy and assigning the policy to a server or group of servers, can enable transparency within the BIOS settings configuration.



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

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

1. Select the Servers tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click BIOS Policies.
4. Select Create BIOS Policy.
5. Enter your preferred BIOS policy name (ucs).
6. Change the BIOS settings as shown in the following figures.
7. Only changes that need to be made are in the Processor and RAS Memory settings.

UCS Manager

3 5 5 37

All

- ▶ BIOS Defaults
- ▼ BIOS Policies
 - SRIOV
 - ucs**
 - UCS1
 - usNIC
- ▶ Boot Policies
- ▶ Diagnostics Policies
- ▶ Graphics Card Policies
- ▼ Host Firmware Packages
 - default
 - ucs
 - ucs-3.22b
- ▶ IPMI Access Profiles
- ▶ KVM Management Policies
- ▶ Local Disk Config Policies
- ▶ Maintenance Policies

Servers / Policies / root / BIOS Policies / ucs

Main Advanced Boot Options Server Management Events

Processor Intel Directed IO RAS Memory Serial Port USB PCI QPI LOM and PCIe Slots Trusted Platform Graphics Configuration

Advanced Filter Export Print

BIOS Setting	Value
Altitude	Platform Default
CPU Hardware Power Management	Platform Default
Boot Performance Mode	Platform Default
CPU Performance	Enterprise
Core Multi Processing	All
DRAM Clock Throttling	Performance
Direct Cache Access	Enabled
Energy Performance Tuning	Platform Default
Enhanced Intel SpeedStep Tech	Disabled
Execute Disable Bit	Platform Default

All

- ▶ BIOS Defaults
- ▼ BIOS Policies
 - SRIOV
 - ucs**
 - UCS1
 - usNIC
- ▶ Boot Policies
- ▶ Diagnostics Policies
- ▶ Graphics Card Policies
- ▼ Host Firmware Packages
 - default
 - ucs
 - ucs-3.22b
- ▶ IPMI Access Profiles
- ▶ KVM Management Policies
- ▶ Local Disk Config Policies
- ▶ Maintenance Policies

Servers / Policies / root / BIOS Policies / ucs

Main Advanced Boot Options Server Management Events

Processor Intel Directed IO RAS Memory Serial Port USB PCI QPI LOM and PCIe Slots Trusted Platform Graphics Configuration

Advanced Filter Export Print

BIOS Setting	Value
Frequency Floor Override	Platform Default
Intel HyperThreading Tech	Enabled
Intel Turbo Boost Tech	Enabled
Intel Virtualization Technology	Disabled
Channel Interleaving	Auto
IMC Inteleave	Platform Default
Memory Interleaving	Platform Default
Rank Interleaving	Platform Default
Sub NUMA Clustering	Platform Default
Local X2 Apic	Platform Default

Servers / Policies / root / BIOS Policies / ucs

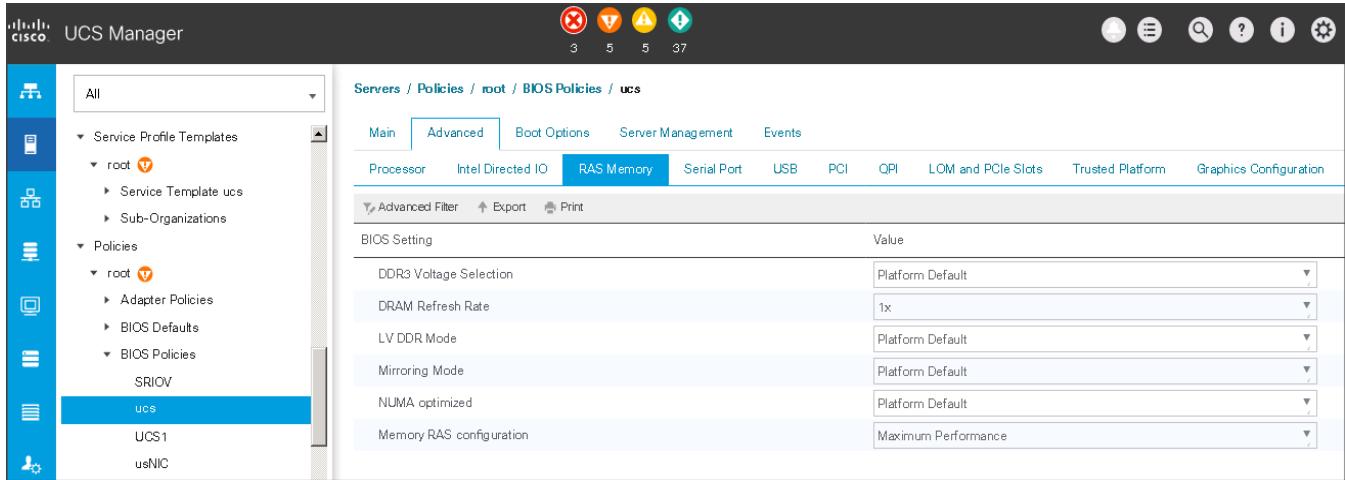
Main		Advanced	Boot Options	Server Management	Events							
		Processor	Intel Directed IO	RAS Memory	Serial Port	USB	PCI	QPI	LOM and PCIe Slots	Trusted Platform	Graphics Configuration	
<input type="checkbox"/> Advanced Filter		<input type="button" value="Export"/>	<input type="button" value="Print"/>									
BIOS Setting		Value										
Max Variable MTRR Setting		Platform Default										
P STATE Coordination		HW ALL										
Package C State Limit		Platform Default										
Processor C State		Disabled										
Processor C1E		Disabled										
Processor C3 Report		Disabled										
Processor C6 Report		Disabled										
Processor C7 Report		Disabled										
Processor CMCI		Platform Default										
Power Technology		Performance										

Servers / Policies / root / BIOS Policies / ucs

Main		Advanced	Boot Options	Server Management	Events							
		Processor	Intel Directed IO	RAS Memory	Serial Port	USB	PCI	QPI	LOM and PCIe Slots	Trusted Platform	Graphics Configuration	
<input type="checkbox"/> Advanced Filter		<input type="button" value="Export"/>	<input type="button" value="Print"/>									
BIOS Setting		Value										
Energy Performance		Performance										
Adjacent Cache Line Prefetcher		Enabled										
DCU IP Prefetcher		Enabled										
DCU Streamer Prefetch		Enabled										
Hardware Prefetcher		Enabled										
UPI Prefetch		Enabled										
LLC Prefetch		Enabled										
XPT Prefetch		Enabled										
Demand Scrub		Enabled										
Patrol Scrub		Enabled										

Servers / Policies / root / BIOS Policies / ucs

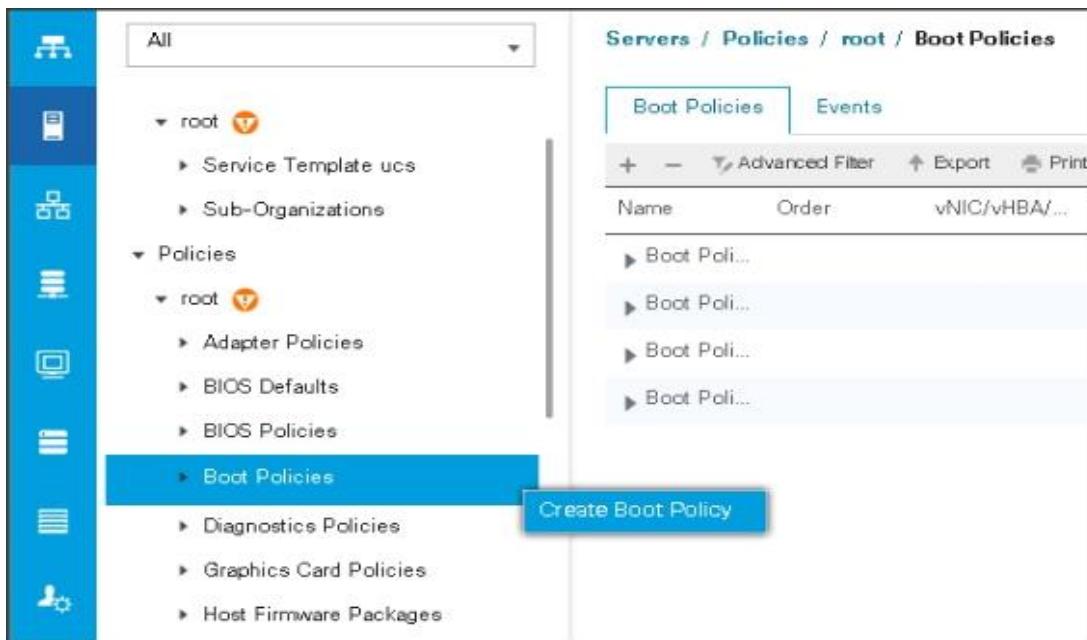
Main		Advanced	Boot Options	Server Management	Events							
		Processor	Intel Directed IO	RAS Memory	Serial Port	USB	PCI	QPI	LOM and PCIe Slots	Trusted Platform	Graphics Configuration	
<input type="checkbox"/> Advanced Filter		<input type="button" value="Export"/>	<input type="button" value="Print"/>									
BIOS Setting		Value										
Adjacent Cache Line Prefetcher		Enabled										
DCU IP Prefetcher		Enabled										
DCU Streamer Prefetch		Enabled										
Hardware Prefetcher		Enabled										
UPI Prefetch		Enabled										
LLC Prefetch		Enabled										
XPT Prefetch		Enabled										
Demand Scrub		Enabled										
Patrol Scrub		Enabled										
Workload Configuration		Platform Default										



Creating the Boot Policy

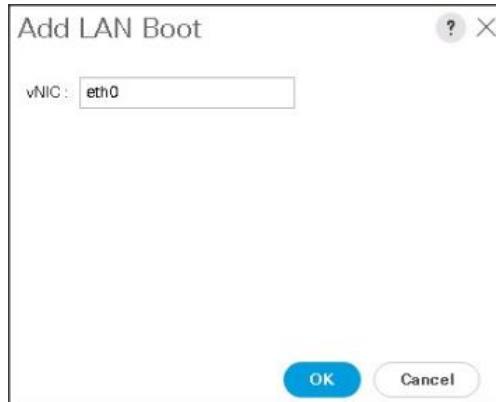
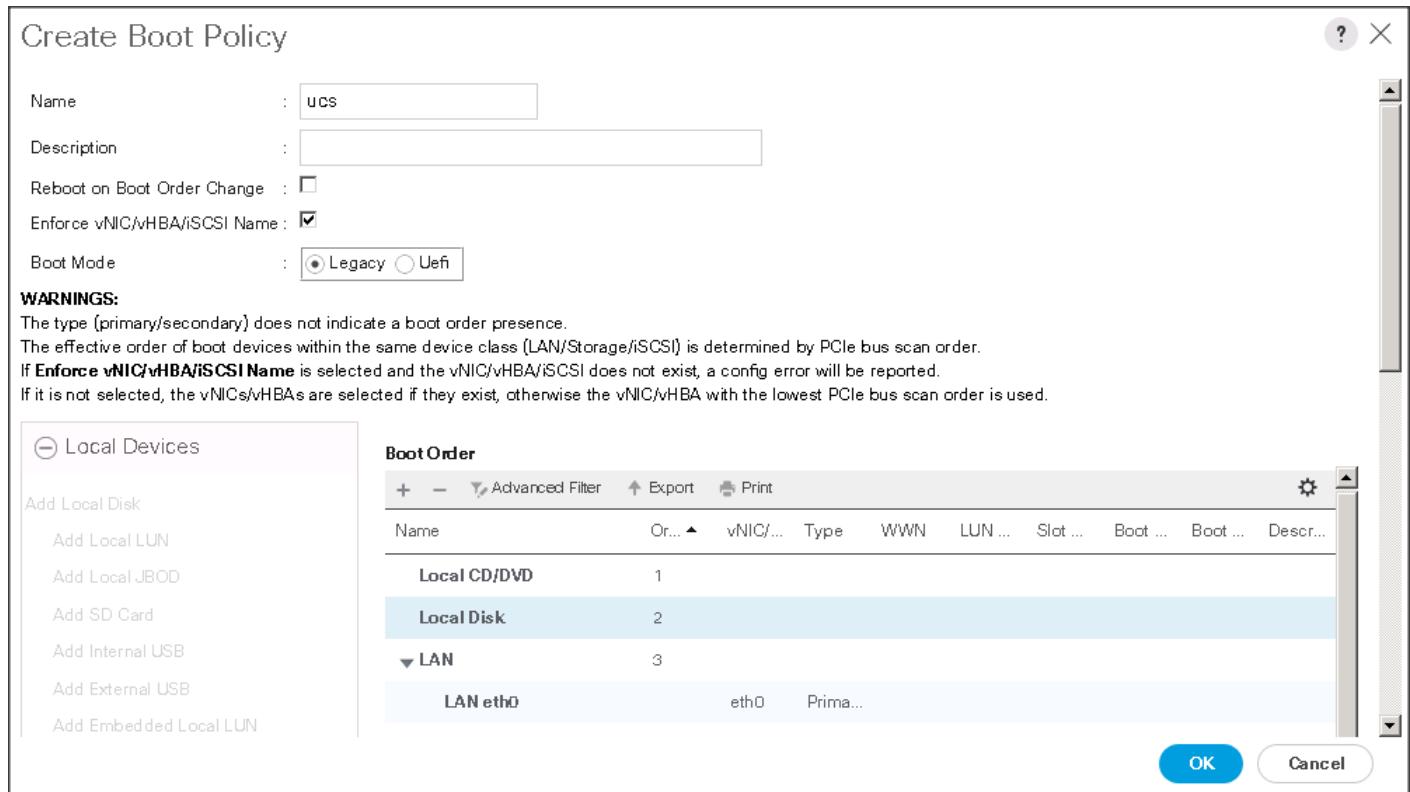
To create boot policies within the Cisco UCS Manager GUI, complete the following steps:

1. Select the Servers tab in the left pane in the UCS Manager GUI.
2. Select Policies > root.
3. Right-click the Boot Policies.
4. Select Create Boot Policy.



5. Enter ucs as the boot policy name.
6. (Optional) enter a description for the boot policy.
7. Keep the Reboot on Boot Order Change check box unchecked.

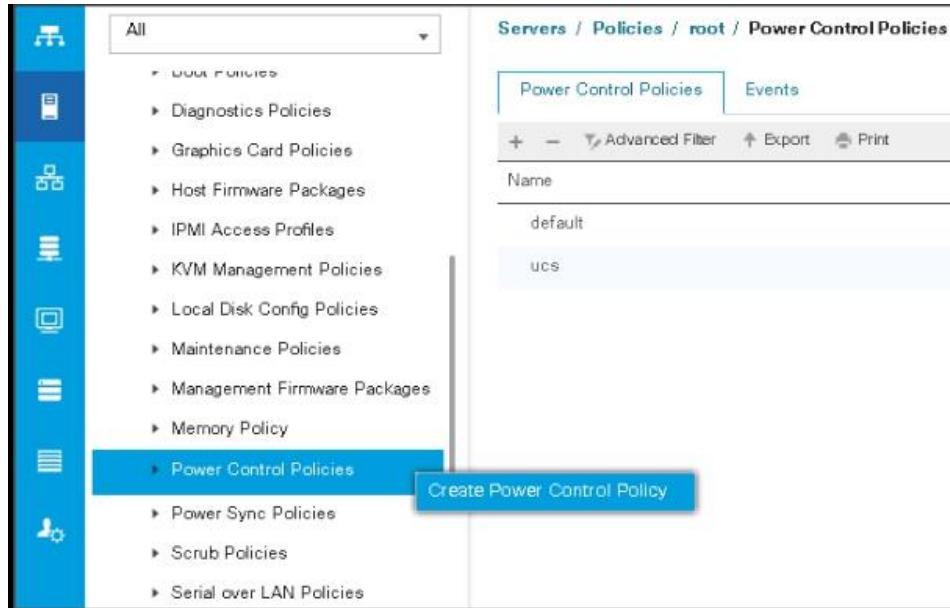
8. Keep Enforce vNIC/vHBA/iSCSI Name check box checked.
9. Keep Boot Mode Default (Legacy).
10. Expand Local Devices > Add CD/DVD and select Add Local CD/DVD.
11. Expand Local Devices and select Add Local Disk.
12. Expand vNICs and select Add LAN Boot and enter eth0.
13. Click OK to add the Boot Policy.
14. Click OK.



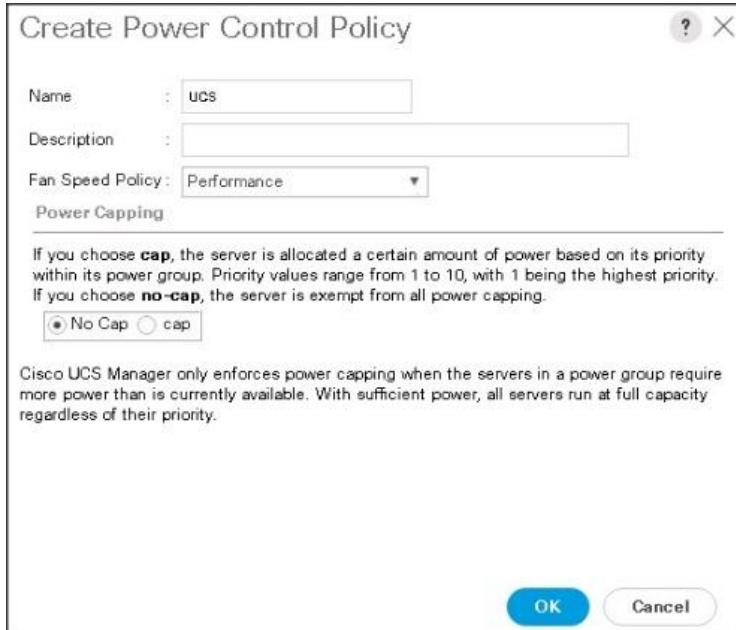
Creating Power Control Policy

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

1. Select the Servers tab in the left pane in the Cisco UCS Manager GUI.
2. Select Policies > root.
3. Right-click the Power Control Policies.
4. Select Create Power Control Policy.



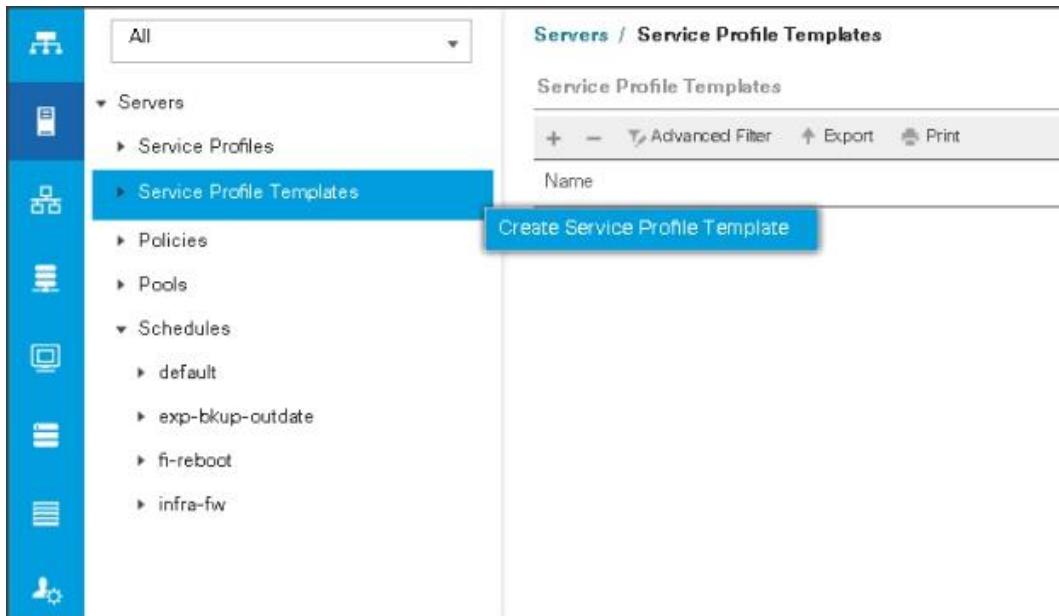
5. Enter ucs as the Power Control policy name.
6. (Optional) enter a description for the boot policy.
7. Select Performance for Fan Speed Policy.
8. Select No cap for Power Capping selection.
9. Click OK to create the Power Control Policy.
10. Click OK.



Creating a Service Profile Template

To create a Service Profile Template, complete the following steps:

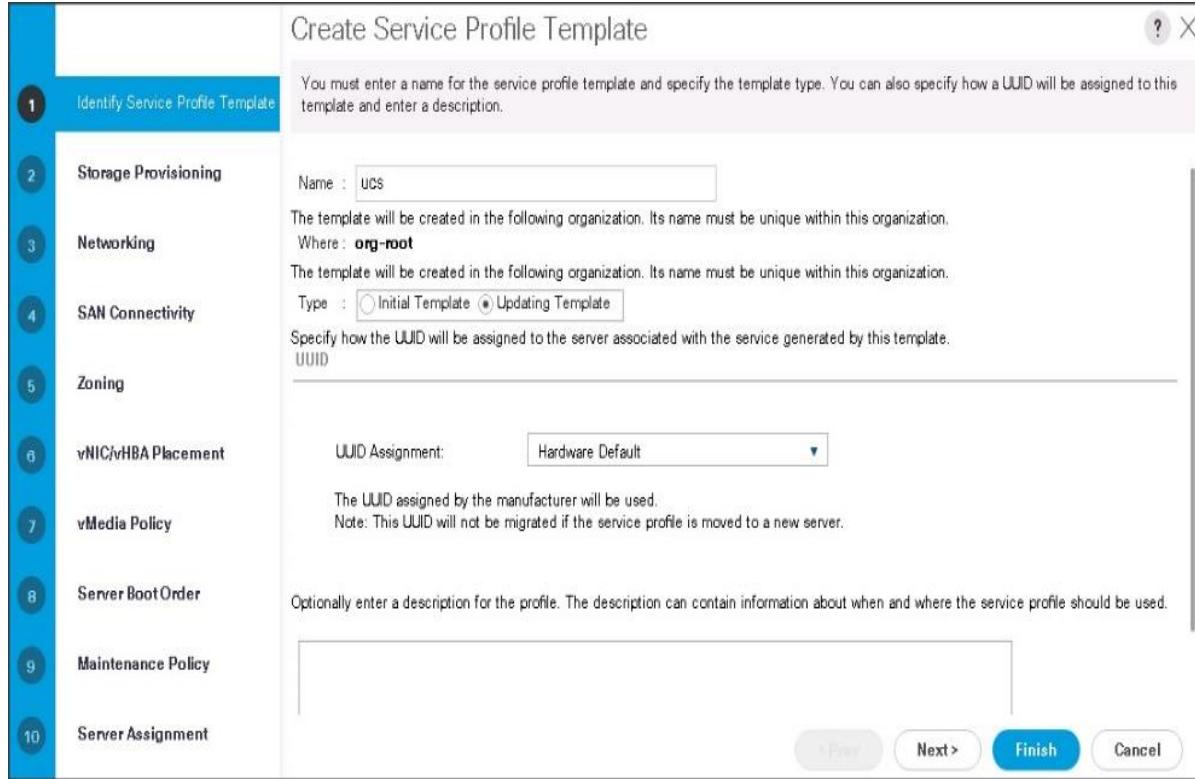
1. Select the Servers tab in the left pane in the Cisco UCS Manager GUI.
2. Right-click Service Profile Templates.
3. Select Create Service Profile Template.



The Create Service Profile Template window appears.

To identify the service profile template, complete the following steps:

1. Name the service profile template as `ucs`. Select the `Updating Template` radio button.
2. In the `UUID` section, select `Hardware Default` as the `UUID pool`.
3. Click `Next` to continue to the next section.



Configuring the Storage Provisioning for the Template

To configure storage policies, complete the following steps:

1. Go to the Local Disk Configuration Policy tab, and select `ucs` for the Local Storage.
2. Click `Next` to continue to the next section.

Create Service Profile Template

Optional specify or create a Storage Profile, and select a local disk configuration policy.

Specific Storage Profile Storage Profile Policy Local Disk Configuration Policy

Local Storage: ucs ▾

Mode : Any Configuration
Protect Configuration : No
If **Protect Configuration** is set, the local disk configuration is preserved if the service profile is disassociated with the server. In that case, a configuration error will be raised when a new service profile is associated with that server if the local disk configuration in that profile is different.

FlexFlash

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

FlexFlash RAID Reporting State : Disable

- Click Next once the Networking window appears to go to the next section.

Configuring Network Settings for the Template

- Keep the Dynamic vNIC Connection Policy field at the default.
- Select Expert radio button for the option how would you like to configure LAN connectivity?
- Click Add to add a vNIC to the template.

Create Service Profile Template

Optional specify LAN configuration information.

Dynamic vNIC Connection Policy: Select a Policy to use (no Dynamic vNIC Policy by default) ▾

Create Dynamic vNIC Connection Policy

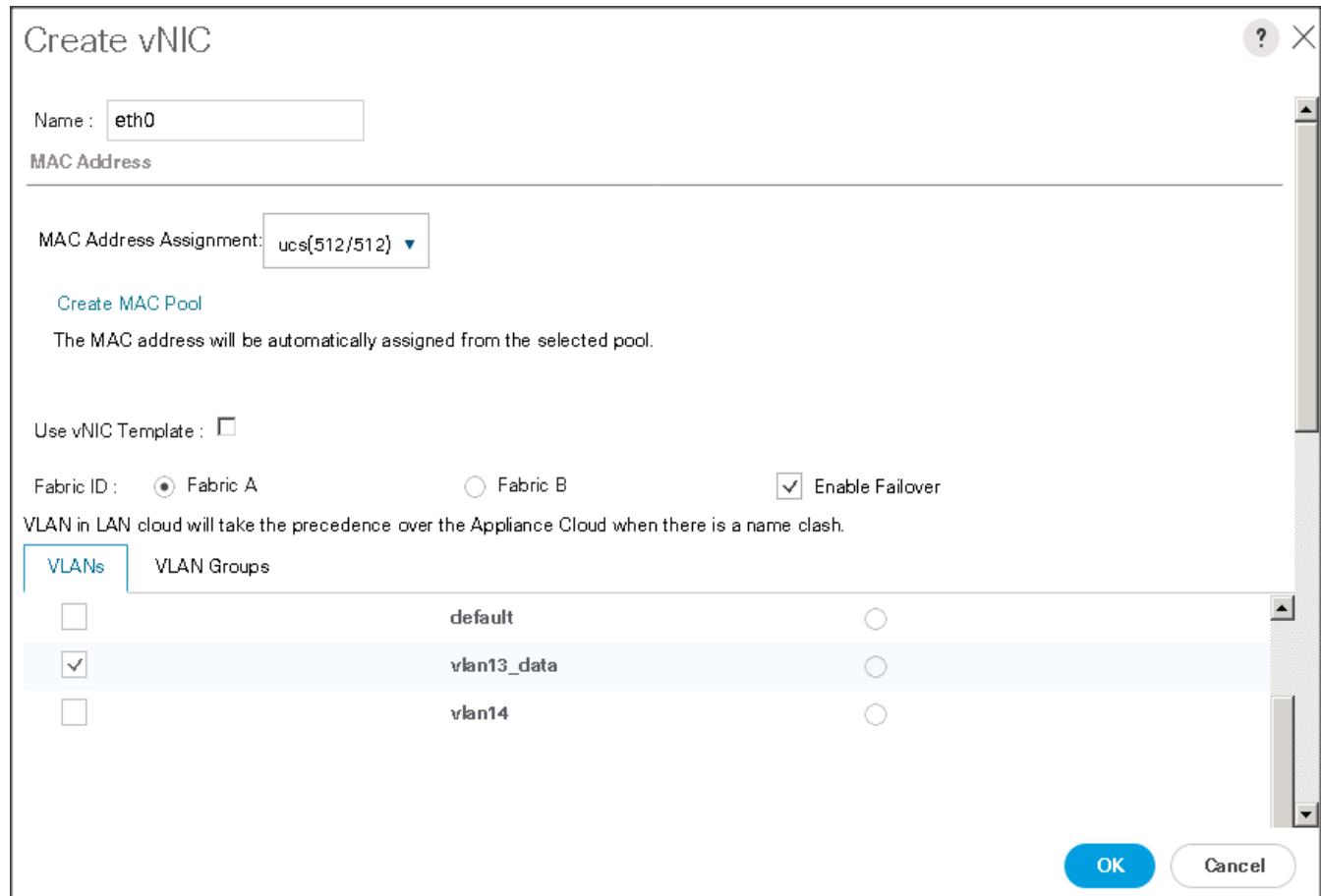
How would you like to configure LAN connectivity?
 Expert Simple No vNICs Use Connectivity Policy

Click **Add** to specify one or more vNICs that the server should use to connect to the LAN.

Name	MAC Address	Fabric ID	Native VLAN
No data available			

< Prev Next > **Finish** Cancel

4. The Create vNIC window displays. Name the vNIC as eth0.
5. Select ucs in the Mac Address Assignment pool.
6. Select the Fabric A radio button and check the Enable failover check box for the Fabric ID.
7. Check the VLAN13 check box for VLANs and select the Native VLAN radio button.
8. Select MTU size as 9000.
9. Select adapter policy as Linux.
10. Select QoS Policy as Platinum.
11. Keep the Network Control Policy as Default.
12. Click OK.



Create vNIC

CDN Source : vNIC Name User Defined

MTU : 9000

Warning
Make sure that the MTU has the same value in the QoS System Class corresponding to the Egress priority of the selected QoS Policy.

Pin Group : <not set> Create LAN Pin Group

Operational Parameters

Adapter Performance Profile

Adapter Policy :	Linux	Create Ethernet Adapter Policy
QoS Policy :	platinum	Create QoS Policy
Network Control Policy :	default	Create Network Control Policy

Connection Policies

OK Cancel

Create Service Profile Template

Identify Service Profile Template

Storage Provisioning

Networking

SAN Connectivity

Zoning

vNIC/vHBA Placement

vMedia Policy

Server Boot Order

Maintenance Policy

Server Assignment

Optional specify LAN configuration information.

Dynamic vNIC Connection Policy: Select a Policy to use (no Dynamic vNIC Policy by default)

Create Dynamic vNIC Connection Policy

How would you like to configure LAN connectivity?

Simple Expert No vNICs Use Connectivity Policy

Click Add to specify one or more vNICs that the server should use to connect to the LAN.

Name	MAC Address	Fabric ID	Native VLAN
vNIC eth0	Derived	A B	

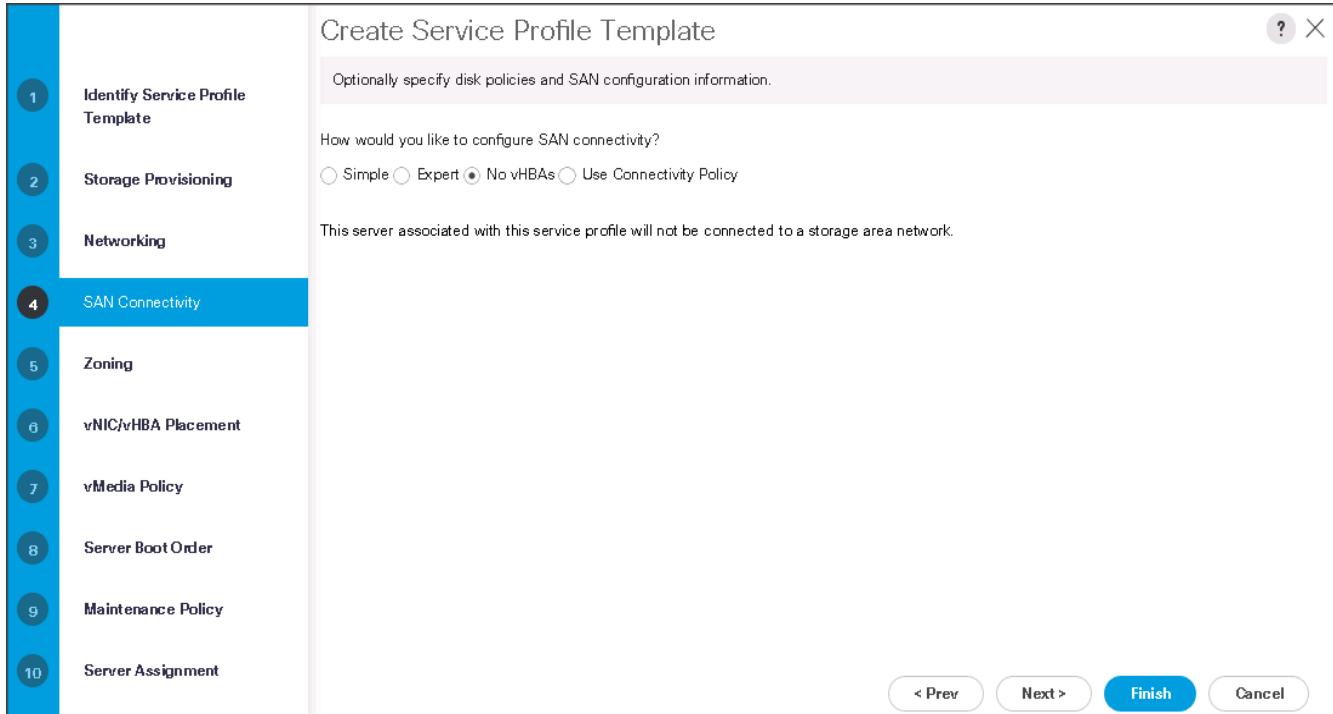
Delete Add Modify

< Prev Next > Finish Cancel

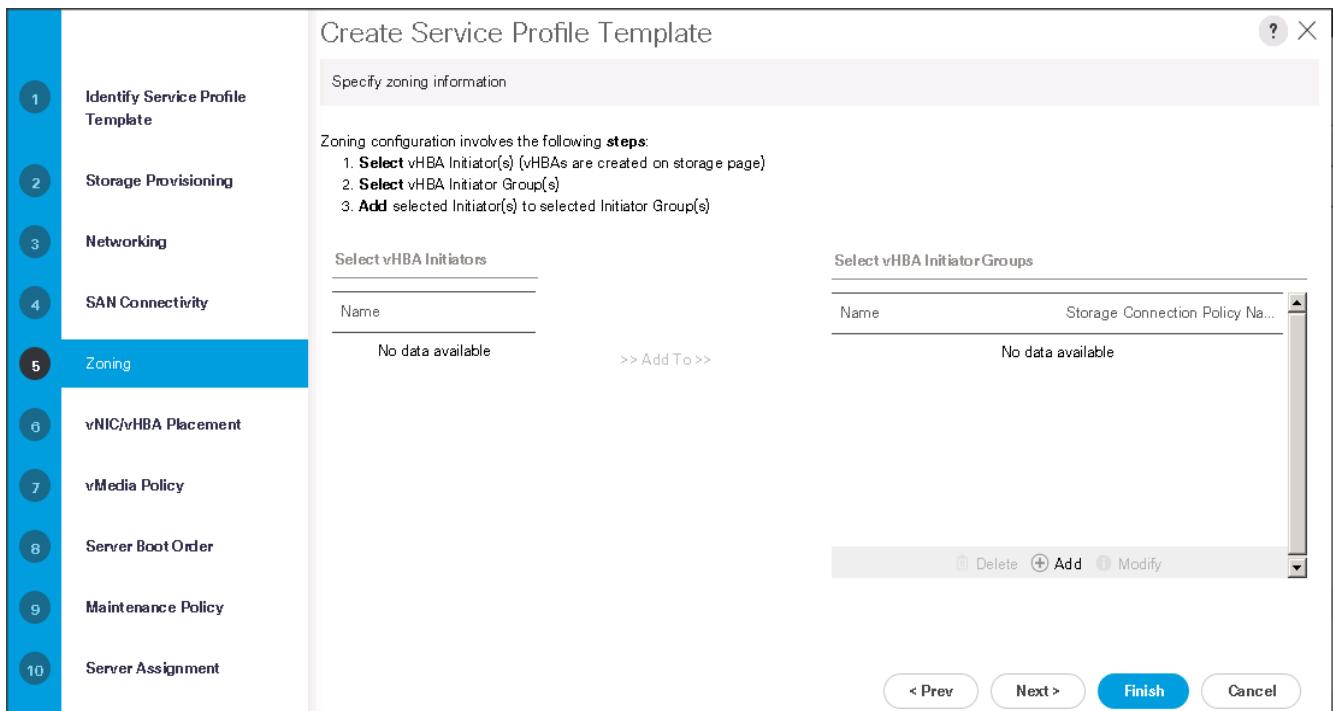


Note: Optionally Network Bonding can be setup on the vNICs for each host for redundancy as well as for increased throughput.

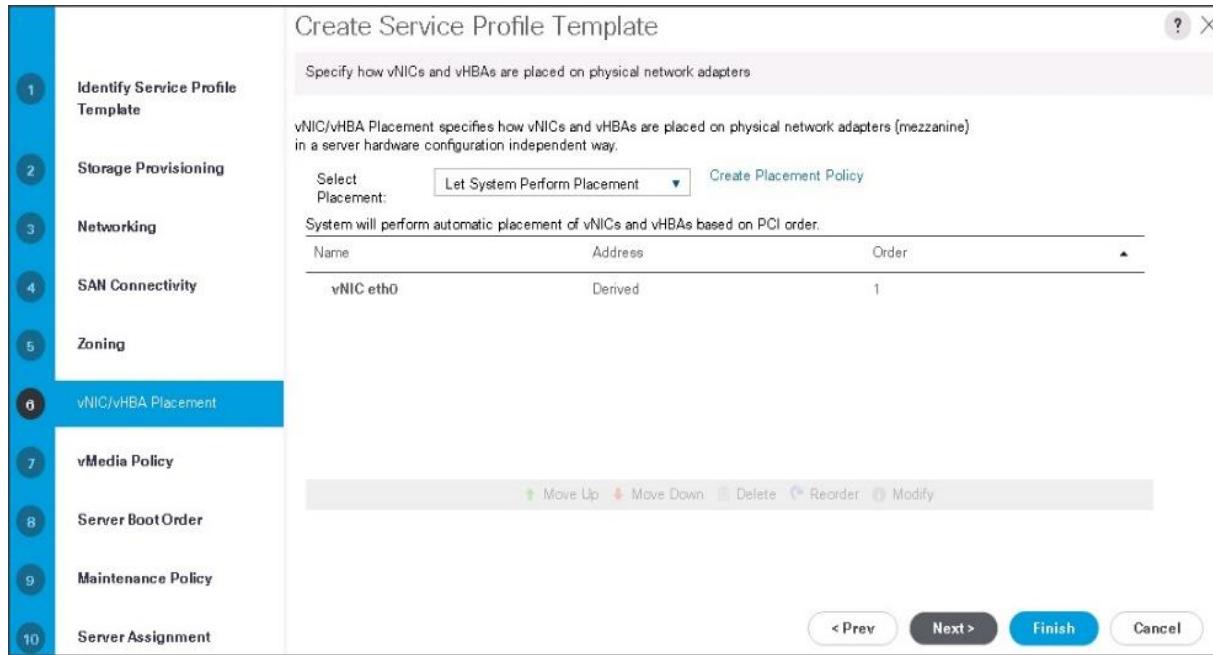
13. Click Next to continue with SAN Connectivity.
14. Select no vHBAs for How would you like to configure SAN Connectivity?



15. Click Next to continue with Zoning .



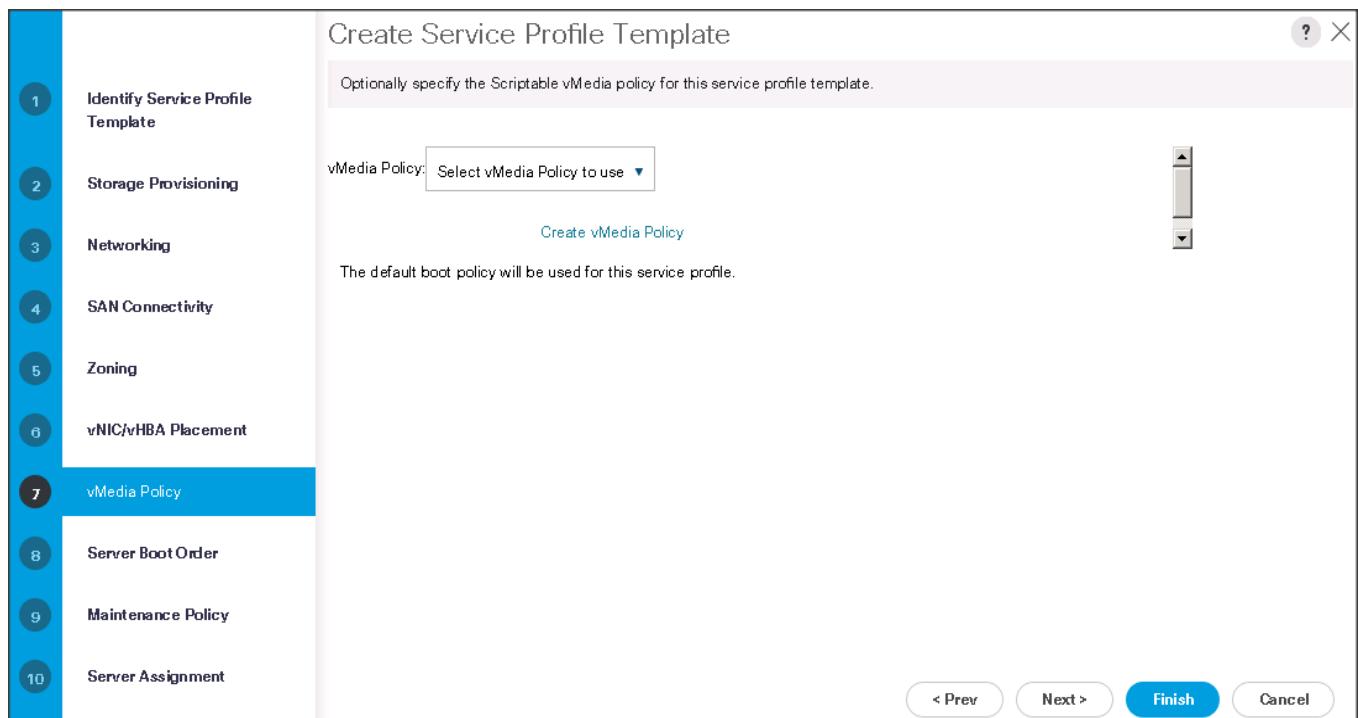
16. Click Next to continue with vNIC/vHBA placement.



17. Click Next to configure vMedia Policy.

Configuring the vMedia Policy for the Template

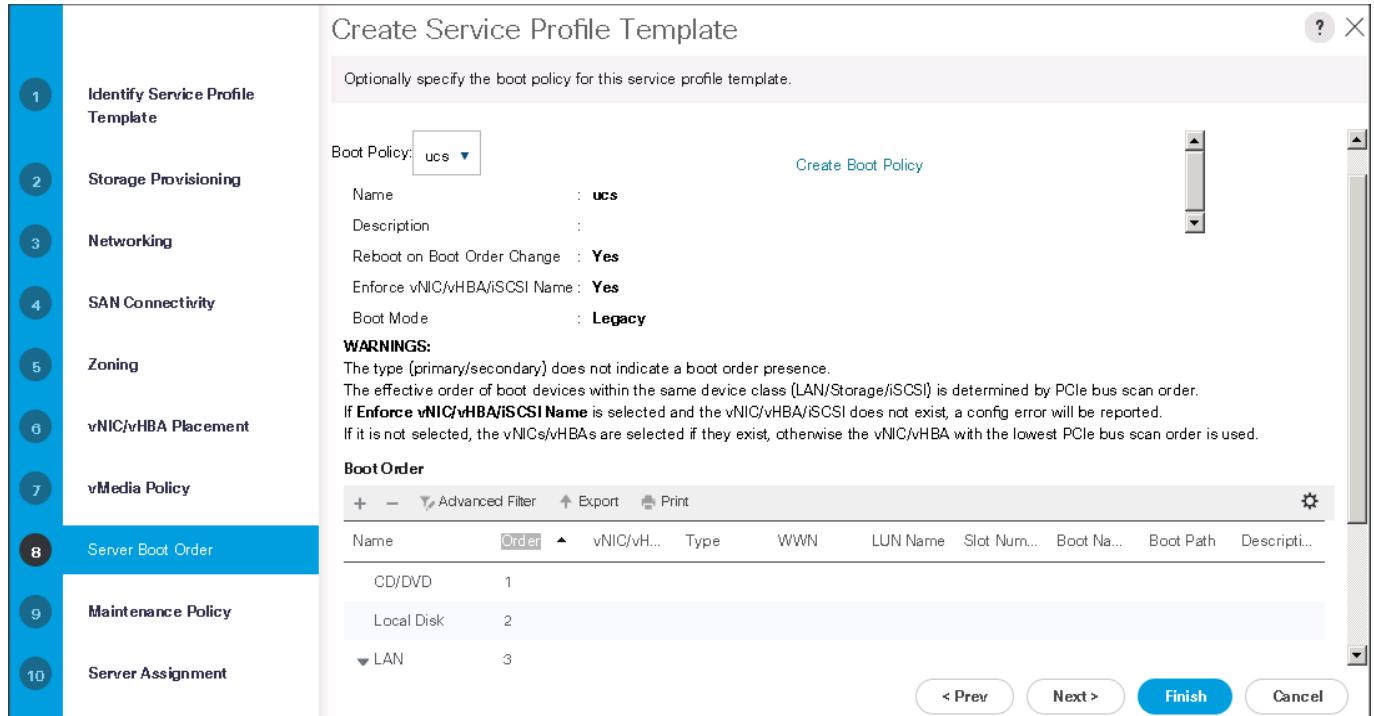
1. Click Next once the vMedia Policy window appears to go to the next section.



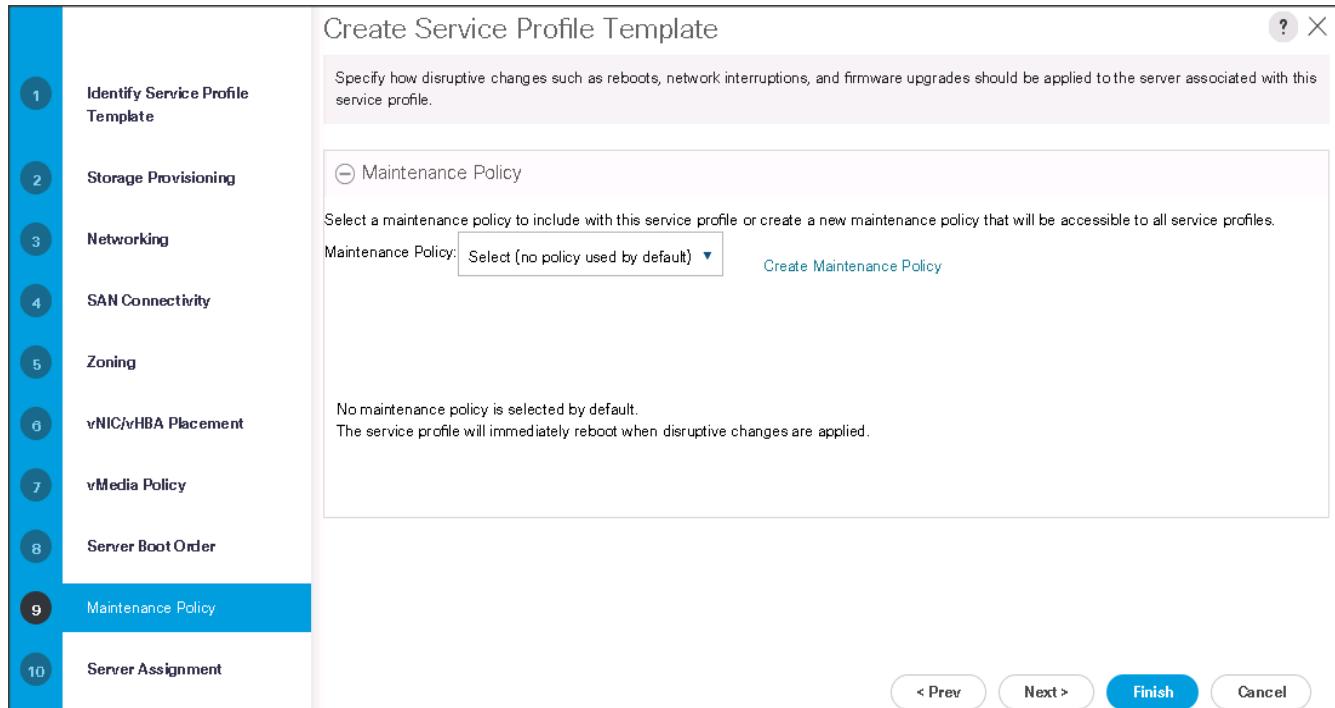
Configuring Server Boot Order for the Template

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

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.



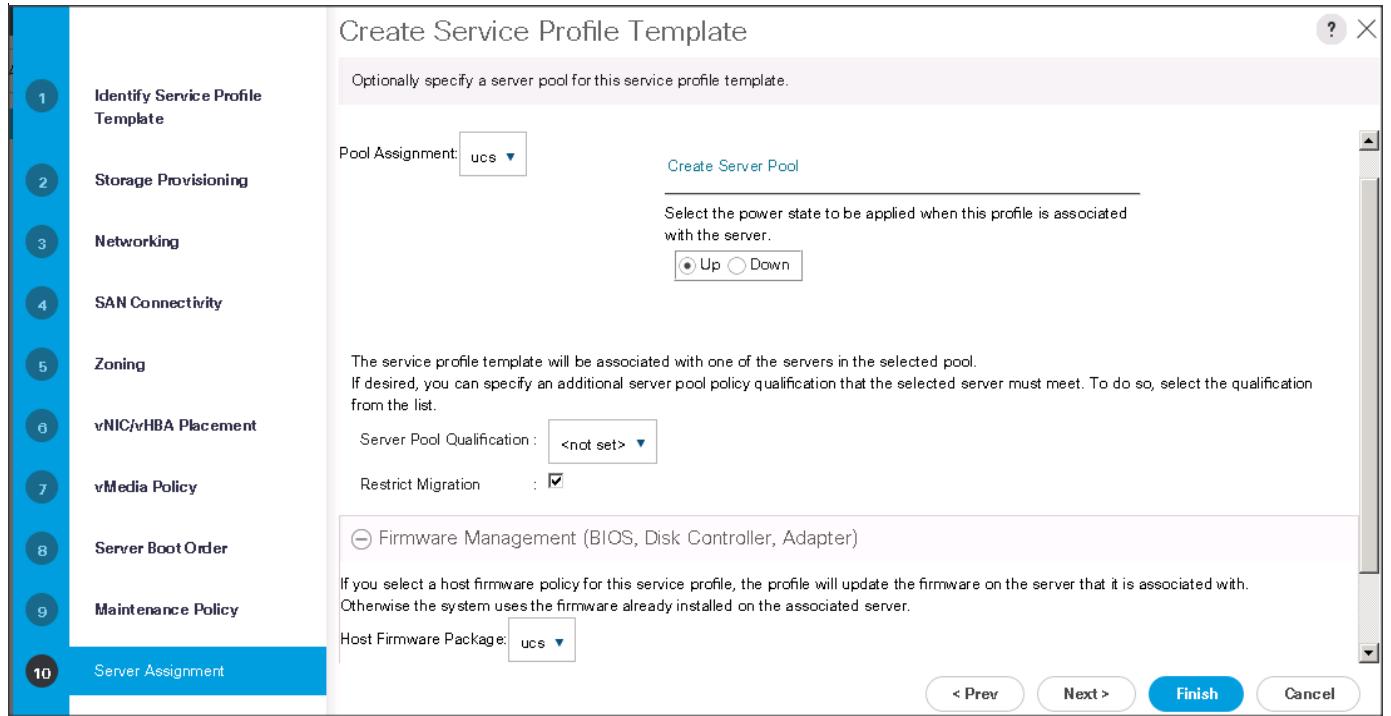
6. In the Maintenance Policy window, apply the maintenance policy.
7. Keep the Maintenance policy at no policy used by default. Click Next to continue to the next section.



Configuring Server Assignment for the Template

In the Server Assignment window, to assign the servers to the pool, complete the following steps:

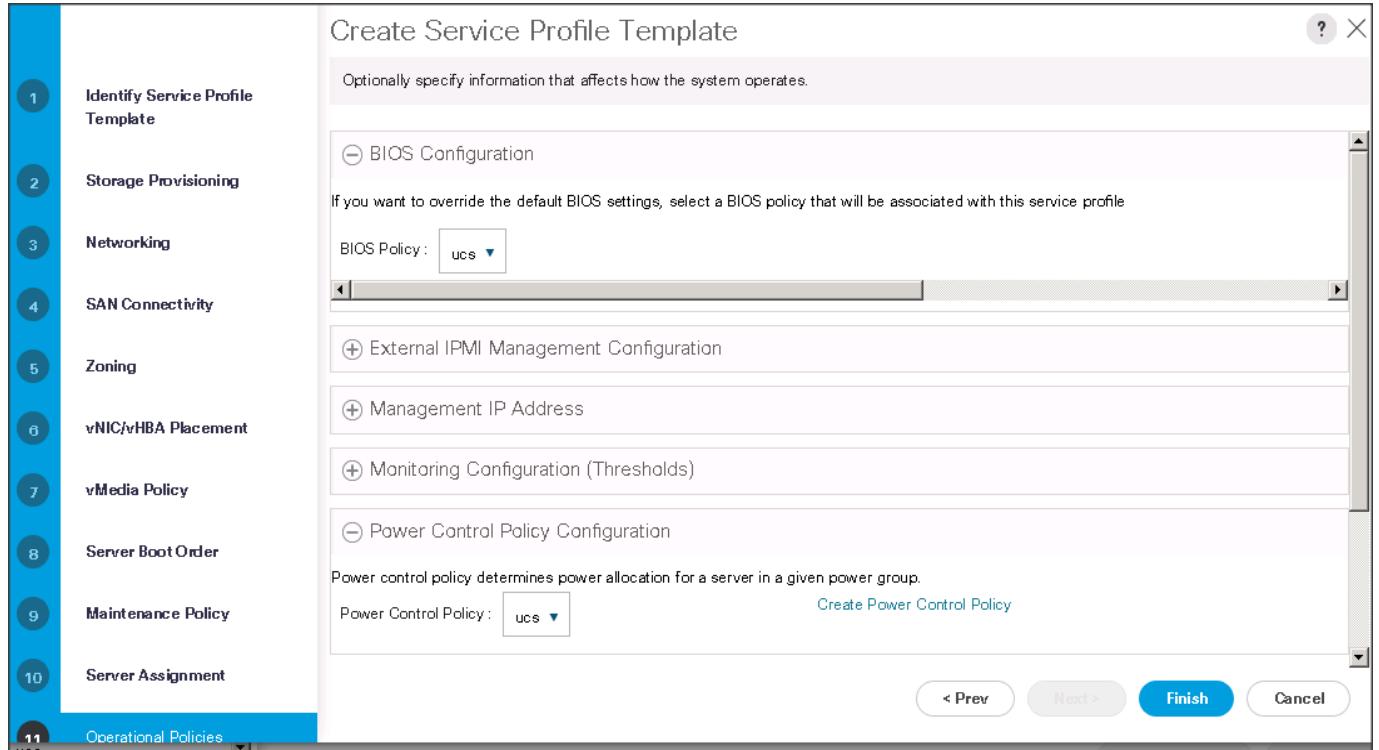
1. Select `ucs` for the Pool Assignment field.
2. Select the power state to be `Up`.
3. Keep the Server Pool Qualification field set to `<not set>`.
4. Check the Restrict Migration check box.
5. Select `ucs` in Host Firmware Package.



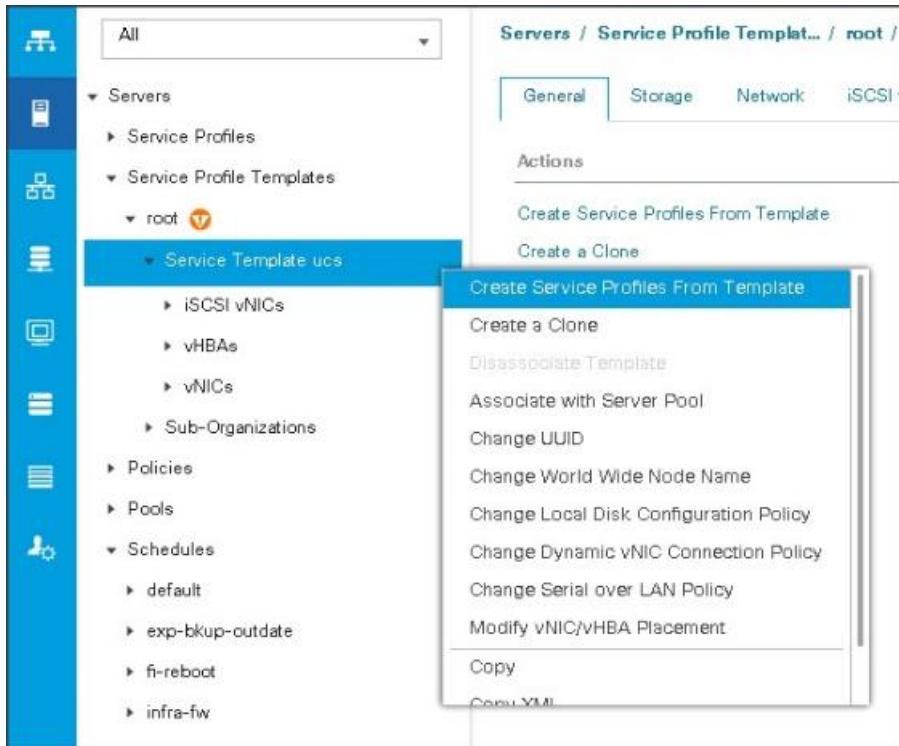
Configuring Operational Policies for the Template

In the Operational Policies Window, complete the following steps:

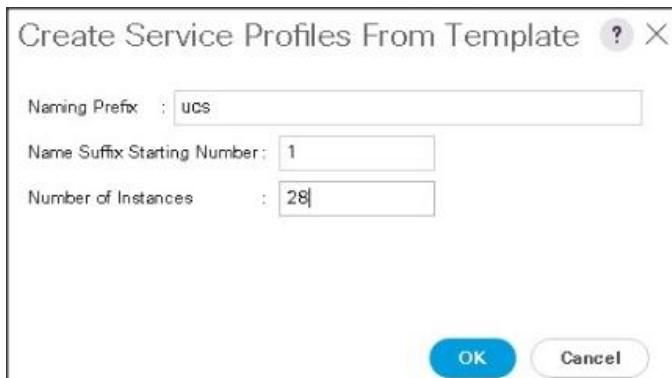
1. Select `ucs` in the BIOS Policy field.
2. Select `ucs` in the Power Control Policy field.



3. Click **Finish** to create the Service Profile template.
4. Click **OK** in the pop-up window to proceed.
5. Select the **Servers** tab in the left pane of the Cisco UCS Manager GUI.
6. Go to **Service Profile Templates > root**.
7. Right-click **Service Profile Templates ucs**.
8. Select **Create Service Profiles From Template**.



The Create Service Profiles from Template window appears.



Association of the Service Profiles will take place automatically.

Installing Red Hat Enterprise Linux 7.4

The following section provides detailed procedures for installing Red Hat Enterprise Linux 7.4 using Software RAID (OS based Mirroring) on Cisco UCS C240 M5 servers. There are multiple ways to install the Red Hat Linux operating system. The installation procedure described in this deployment guide uses KVM console and virtual media from Cisco UCS Manager.

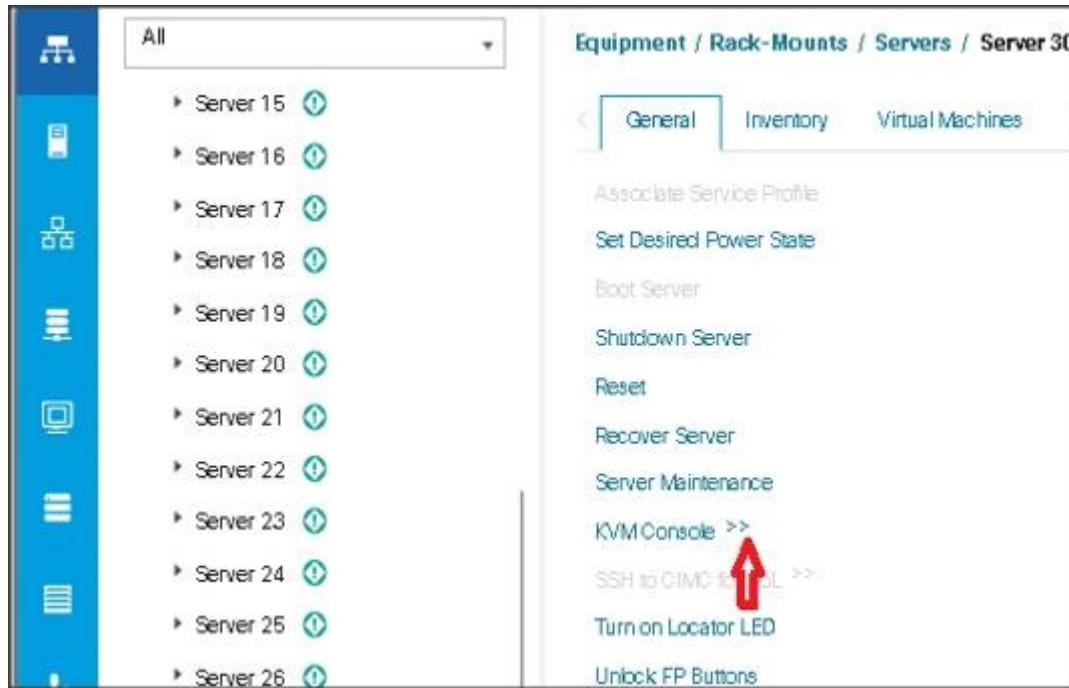


Note: This requires RHEL 7.4 DVD/ISO for the installation

To install the Red Hat Linux 7.4 operating system, complete the following steps:

1. Log in to the Cisco UCS 6332 Fabric Interconnect and launch the Cisco UCS Manager application.
2. Select the Equipment tab.

3. In the navigation pane expand Rack-Mounts and then Servers.
4. In the right pane, click the KVM Console >>.



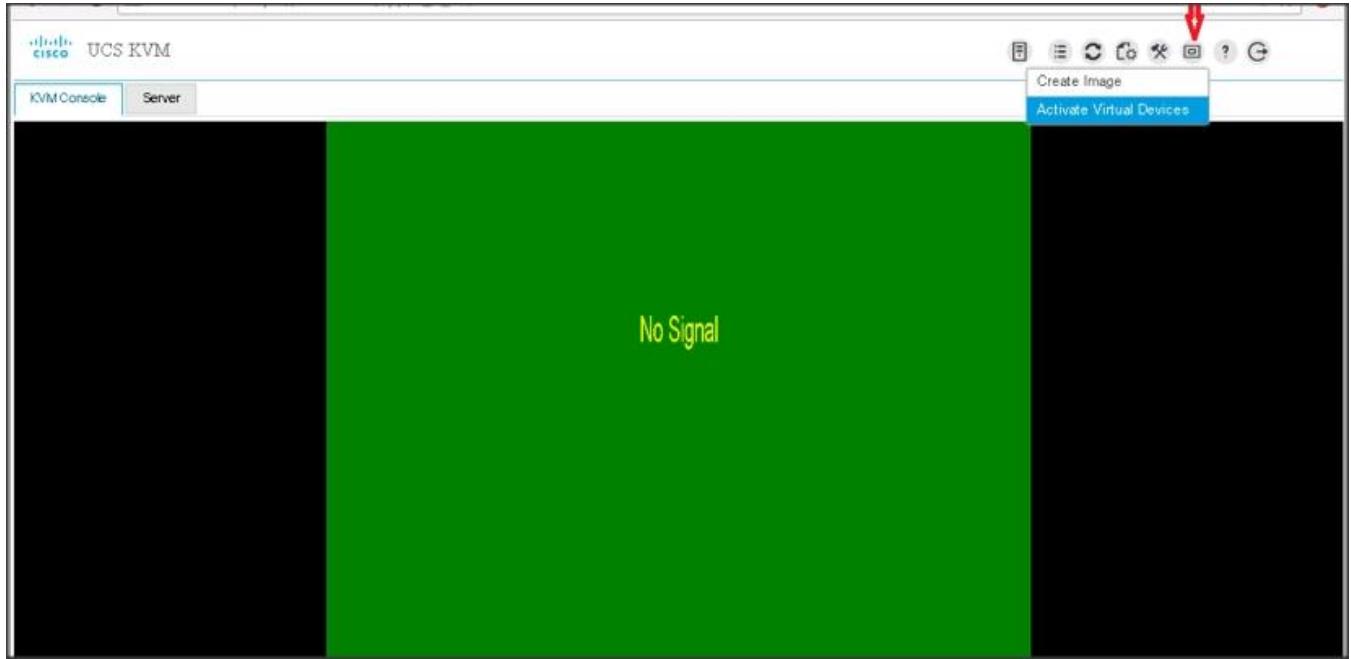
5. Click O.K on KVM Console – Select IP address pop-up window.



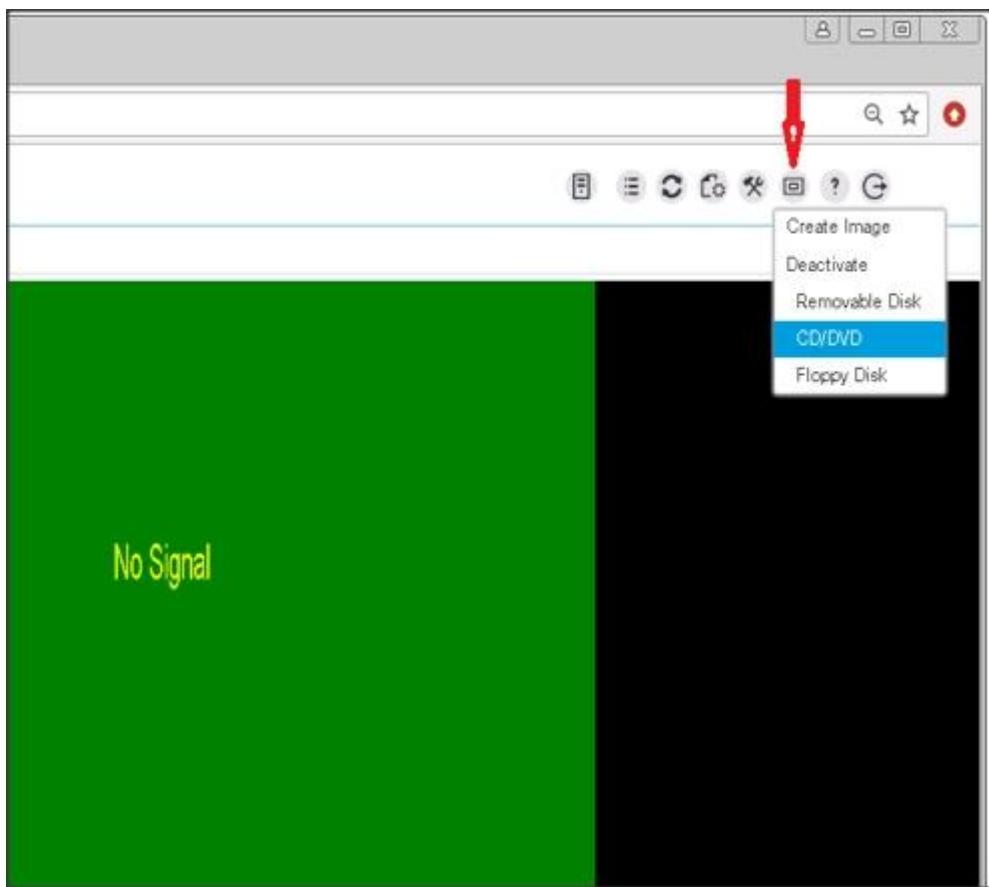
6. Click the link to launch the KVM console.



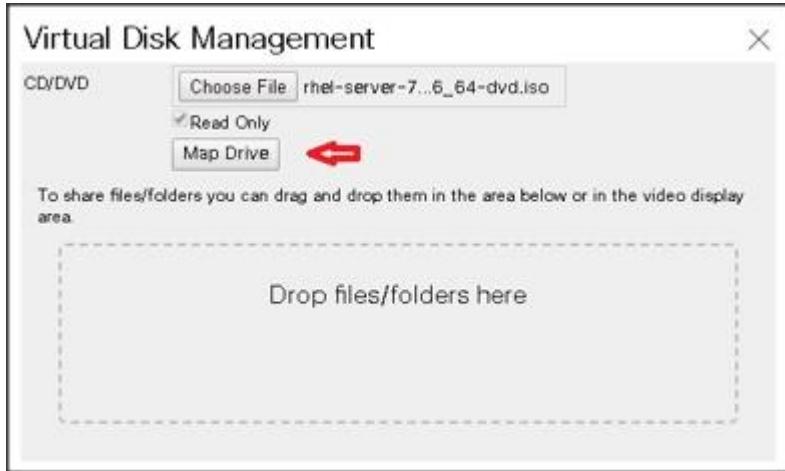
7. Point the cursor over the top right corner, select the Virtual Media tab.



8. Click the Activate Virtual Devices found in Virtual Media tab.
9. Click the Virtual Media tab again to select CD/DVD.



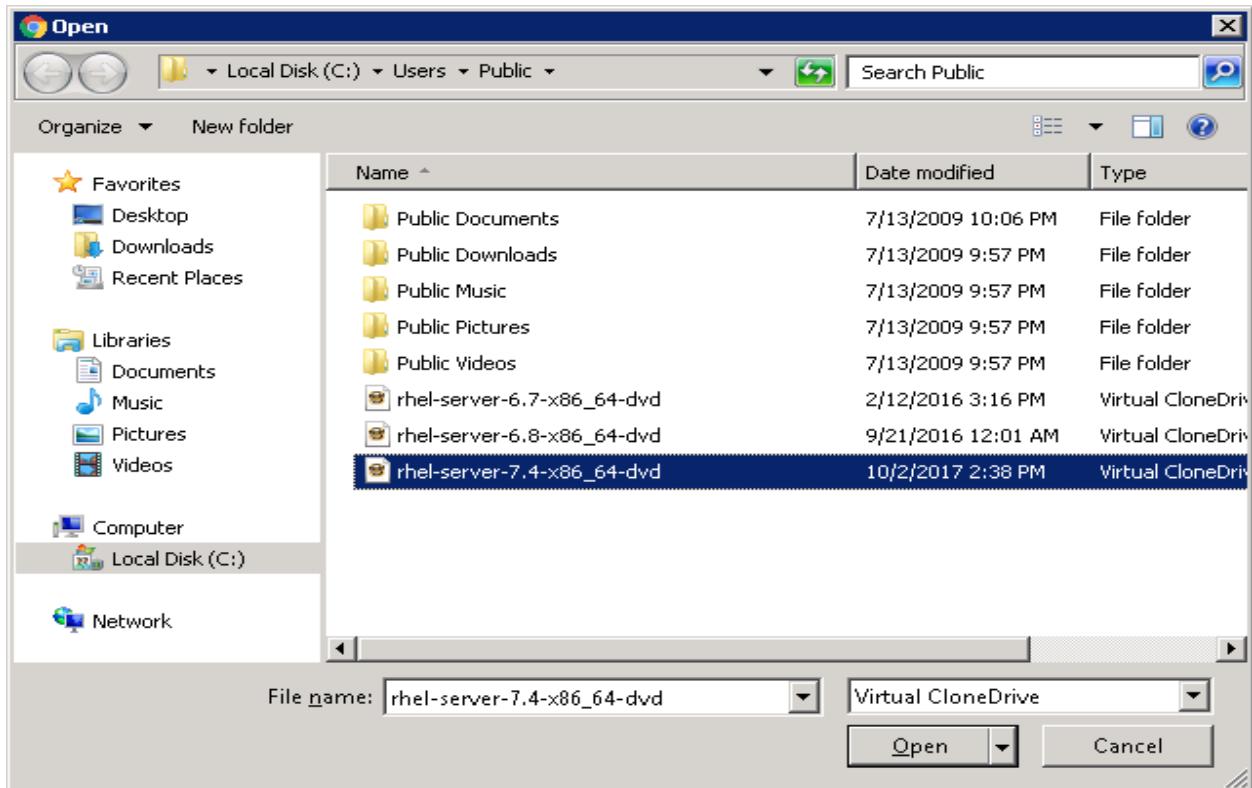
10. Select Map Drive in the Virtual Disk Management windows.



11. Browse to the Red Hat Enterprise Linux Server 7.4 installer ISO image file.



Note: The Red Hat Enterprise Linux 7.4 DVD is assumed to be on the client machine.



12. Click Open to add the image to the list of virtual media.
13. In the KVM window, select the KVM tab to monitor during boot.
14. In the KVM window, select the Macros > Static Macros > Ctrl-Alt-Del button in the upper left corner.
15. Click OK.

16. Click OK to reboot the system.
17. Press F6 key on the keyboard to select install media.



Note: Press F6 on your keyboard as soon as possible when the screen above appears to avoid the server reboot again.

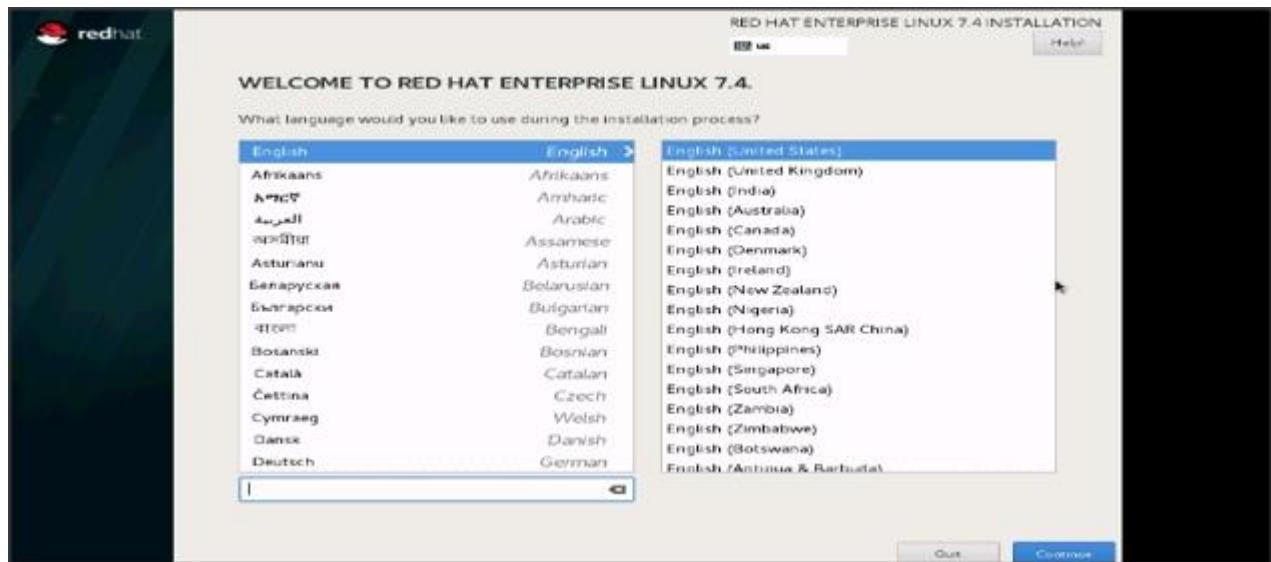
18. On reboot, the machine detects the presence of the Red Hat Enterprise Linux Server 7.4 install media.



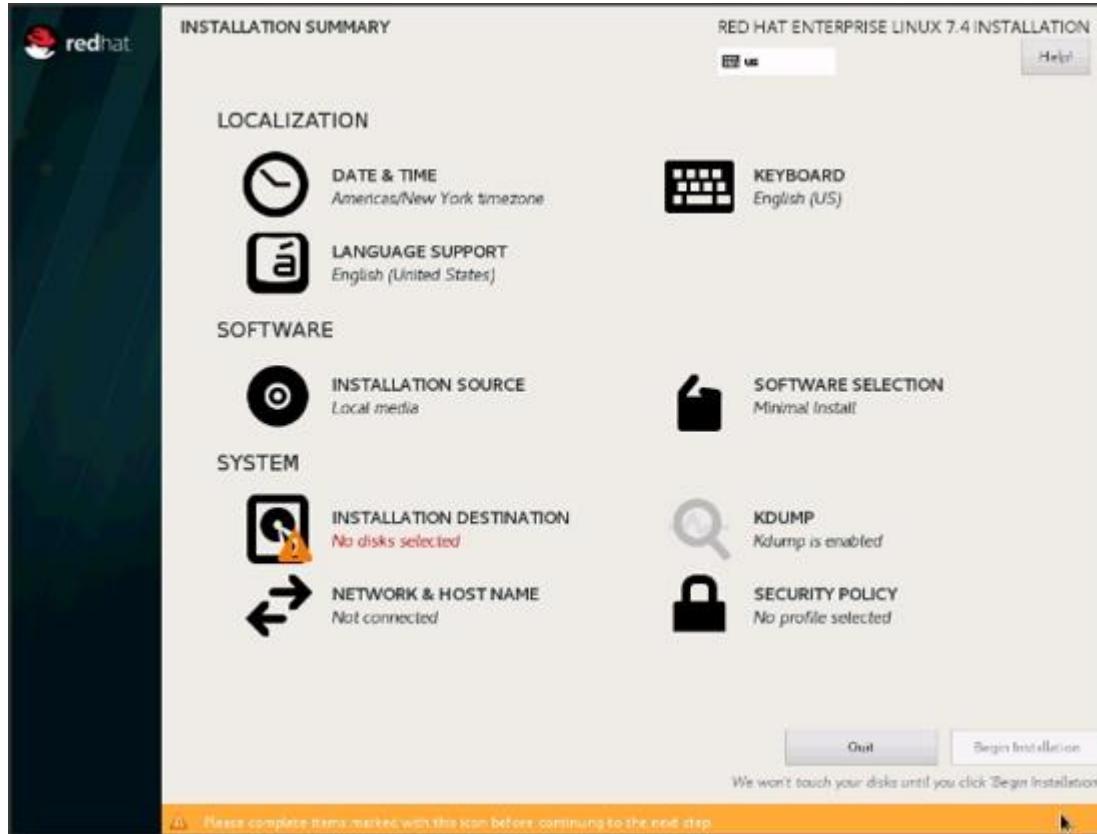
19. Select the Install Red Hat Enterprise Linux 7.4.



20. Skip the Media test and start the installation. Select language of installation and click Continue.



21. Select Date and time, which pops up another window as shown below:



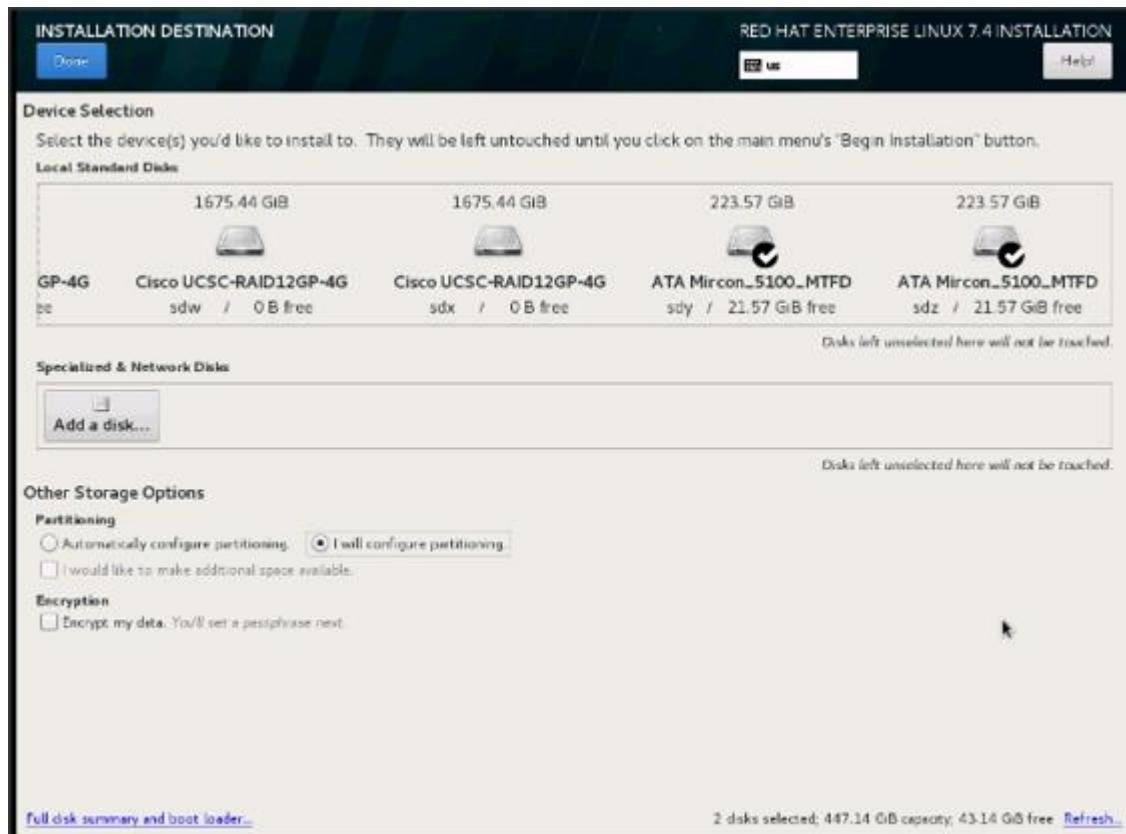
22. Select the location on the map, set the time, and click Done.



23. Click Installation Destination.

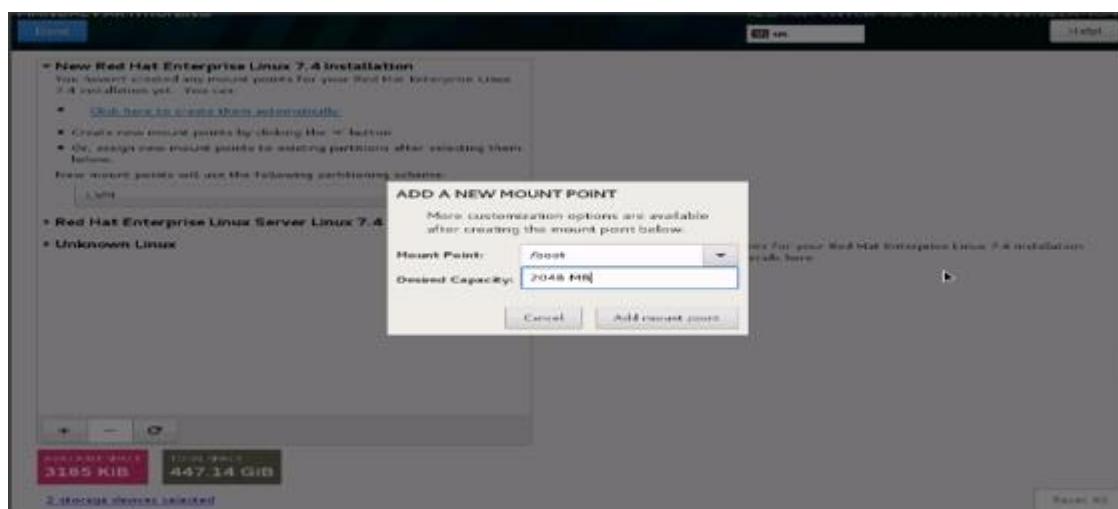


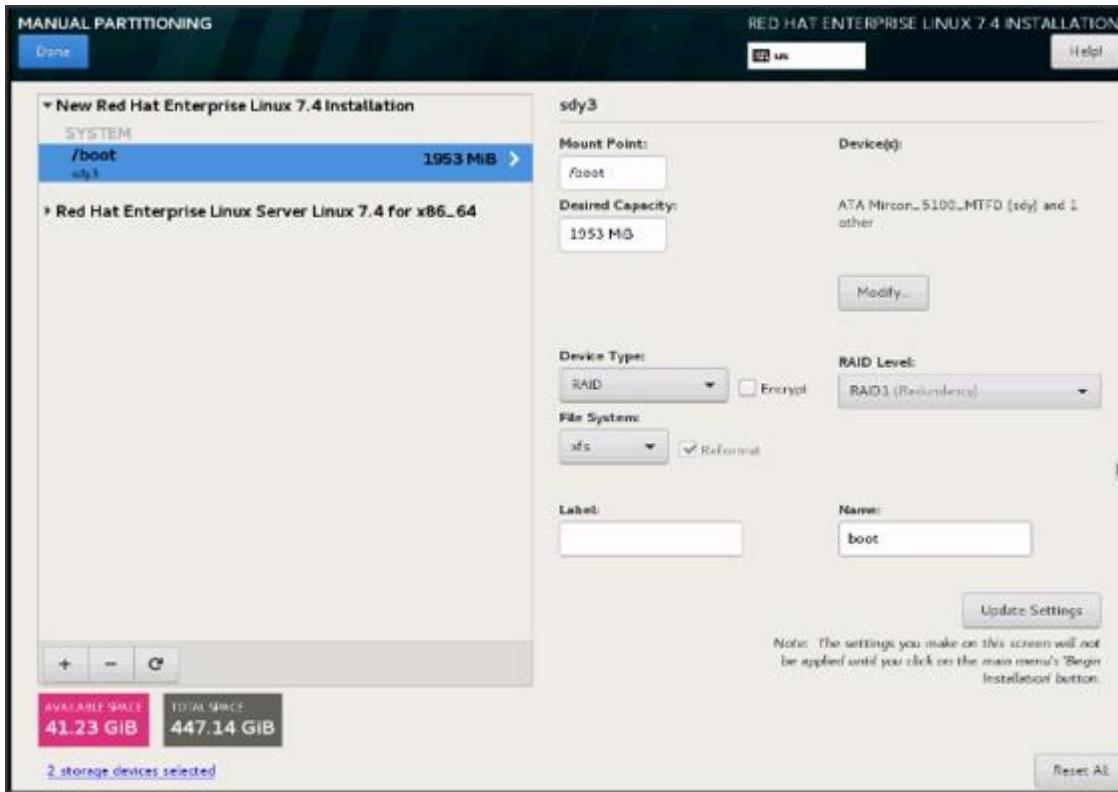
24. This opens a new window with the boot disks. Make the selection, and choose I will configure partitioning. Click Done.



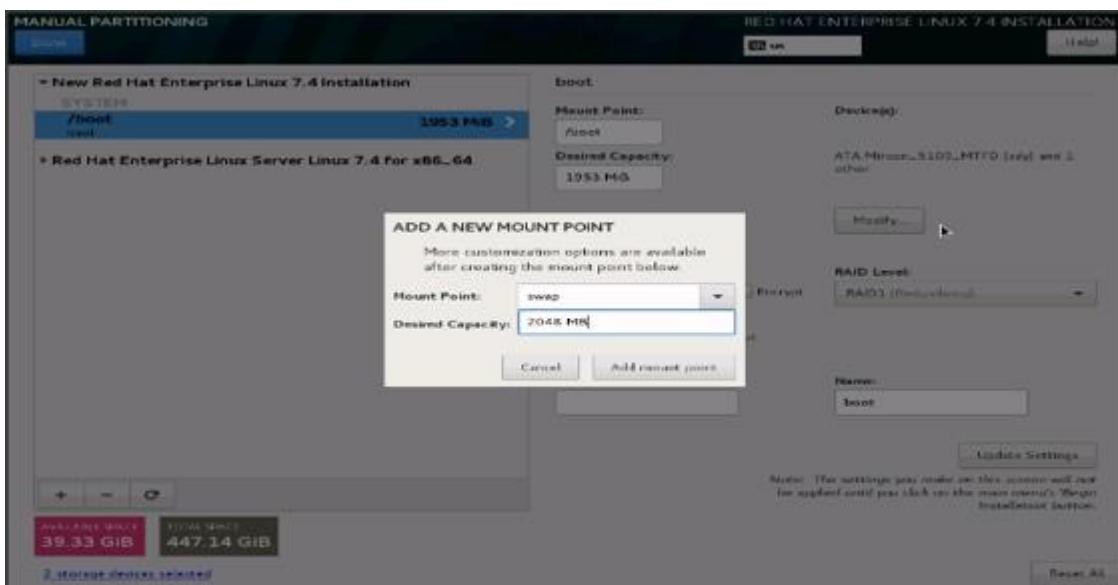
25. This opens the new window for creating the partitions. Click on the + sign to add a new partition as shown below, boot partition of size 2048 MB.

26. Click Add MountPoint to add the partition.

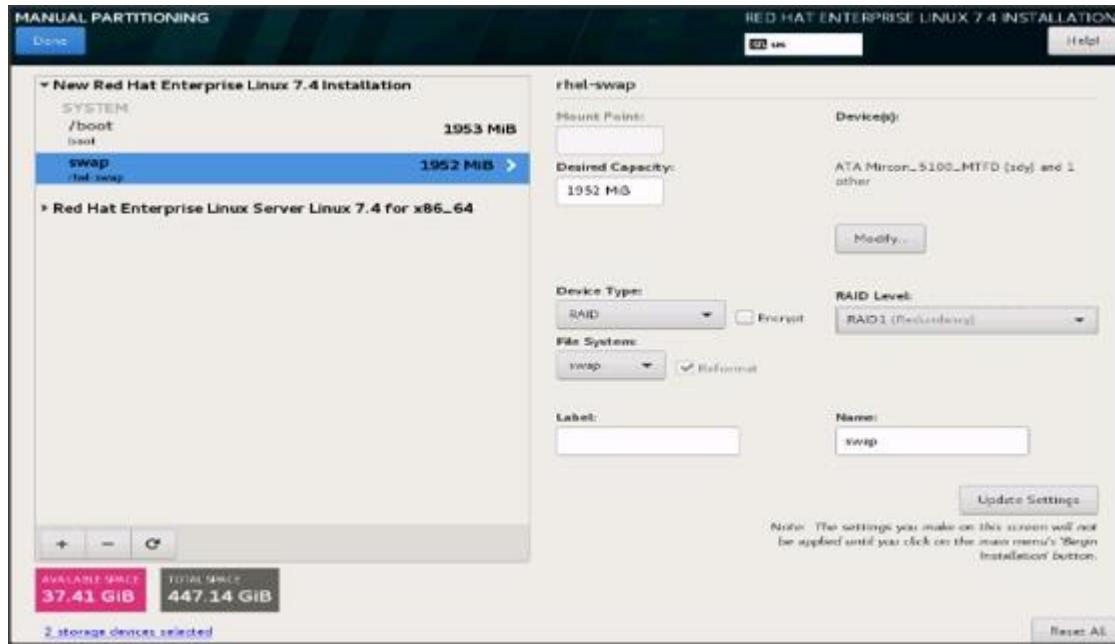




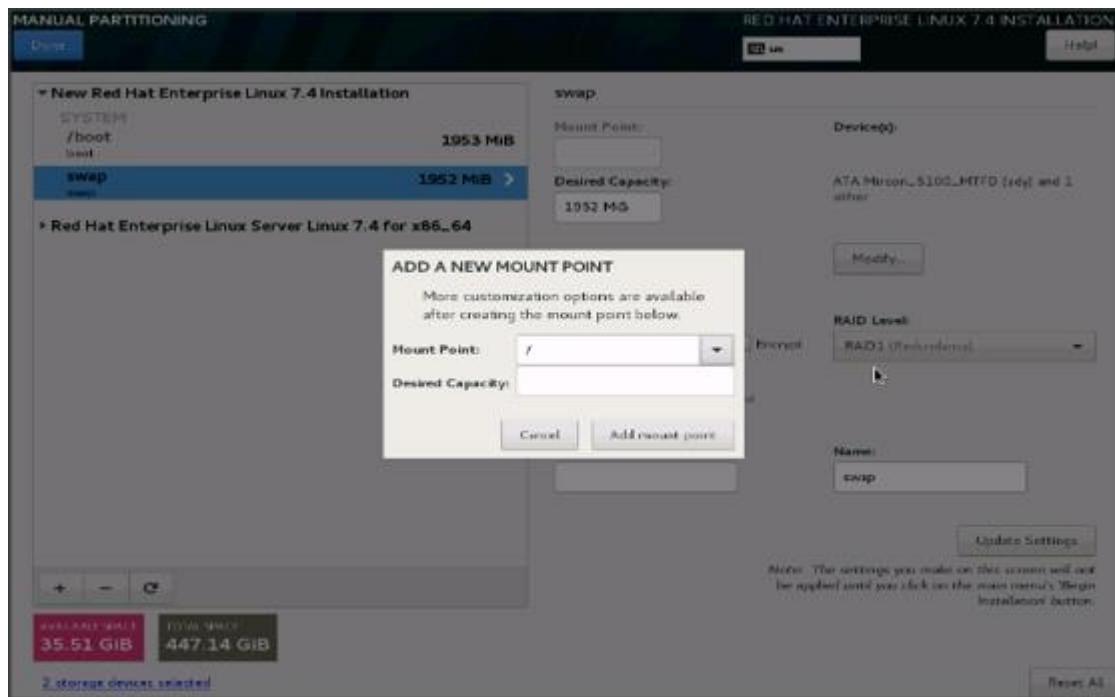
27. Change the Device type to RAID and make sure the RAID Level is RAID1 (Redundancy) and click on Update Settings to save the changes.
28. Click the + sign to create the swap partition of size 2048 MB as shown below.



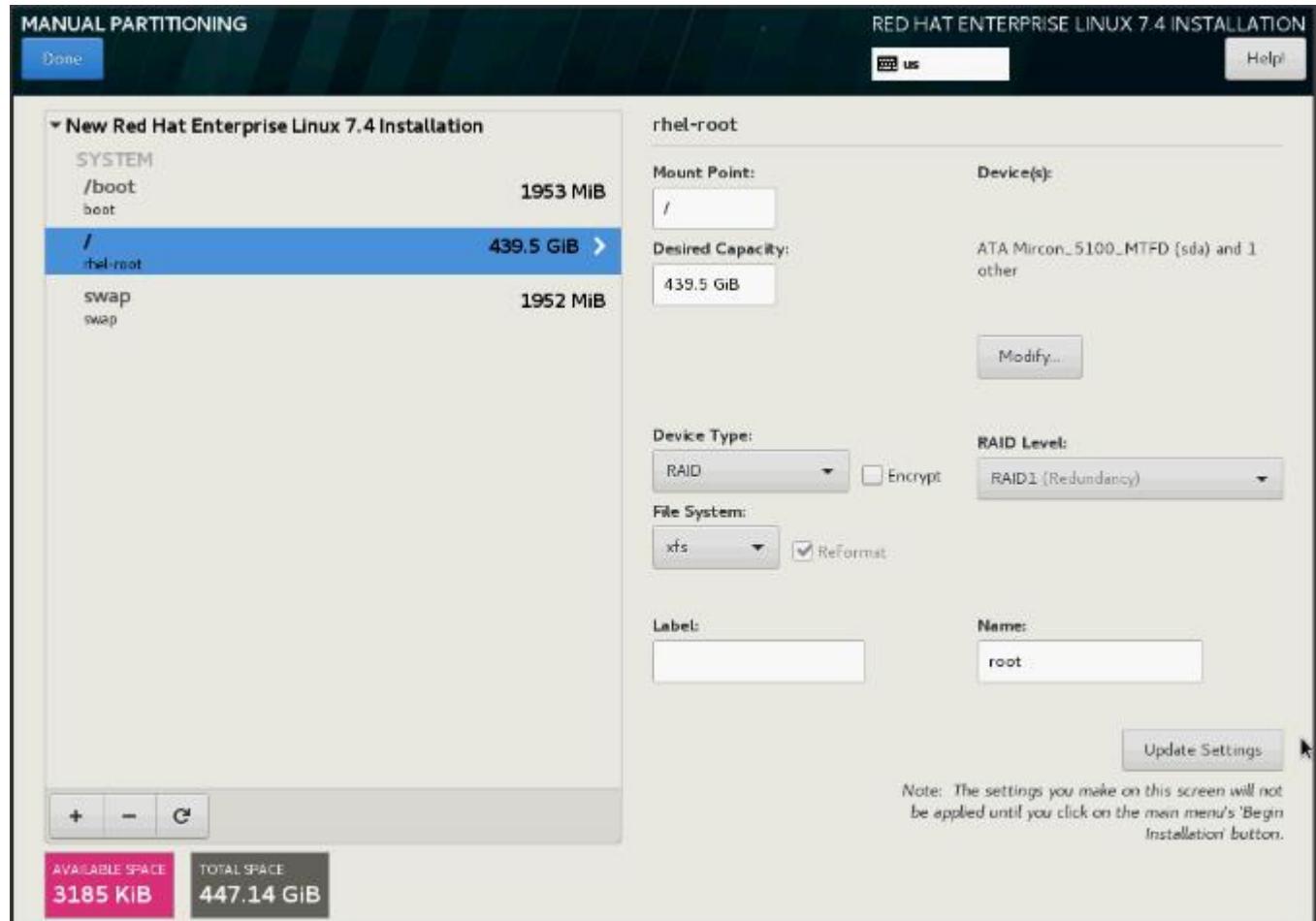
29. Change the Device type to RAID and RAID level to RAID1 (Redundancy) and click Update Settings.



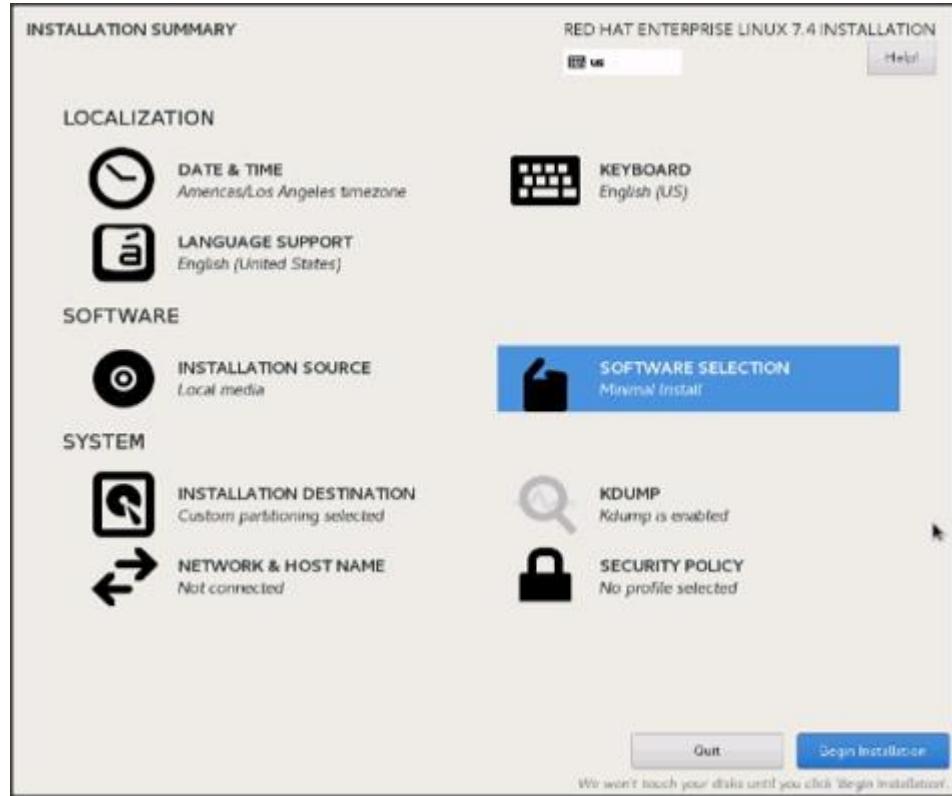
30. Click + to add the / partition. The size can be left empty so it uses the remaining capacity and click Add Mountpoint.



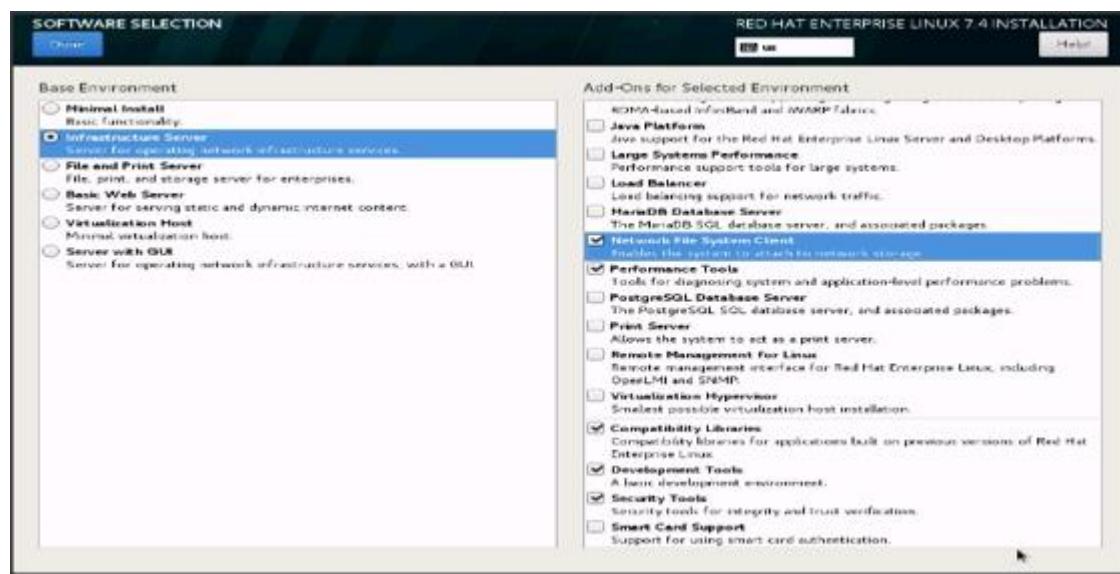
31. Change the Device type to RAID and RAID level to RAID1 (Redundancy). Click Update Settings.



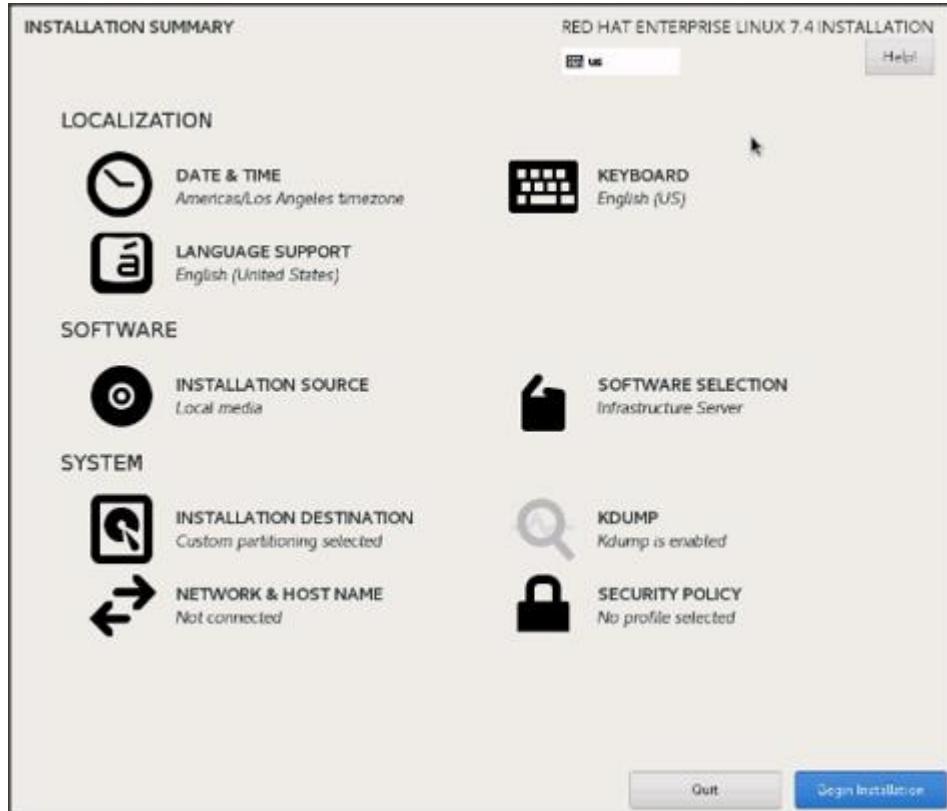
32. Click Done to go back to the main screen and continue the Installation.
33. Click Software Selection.



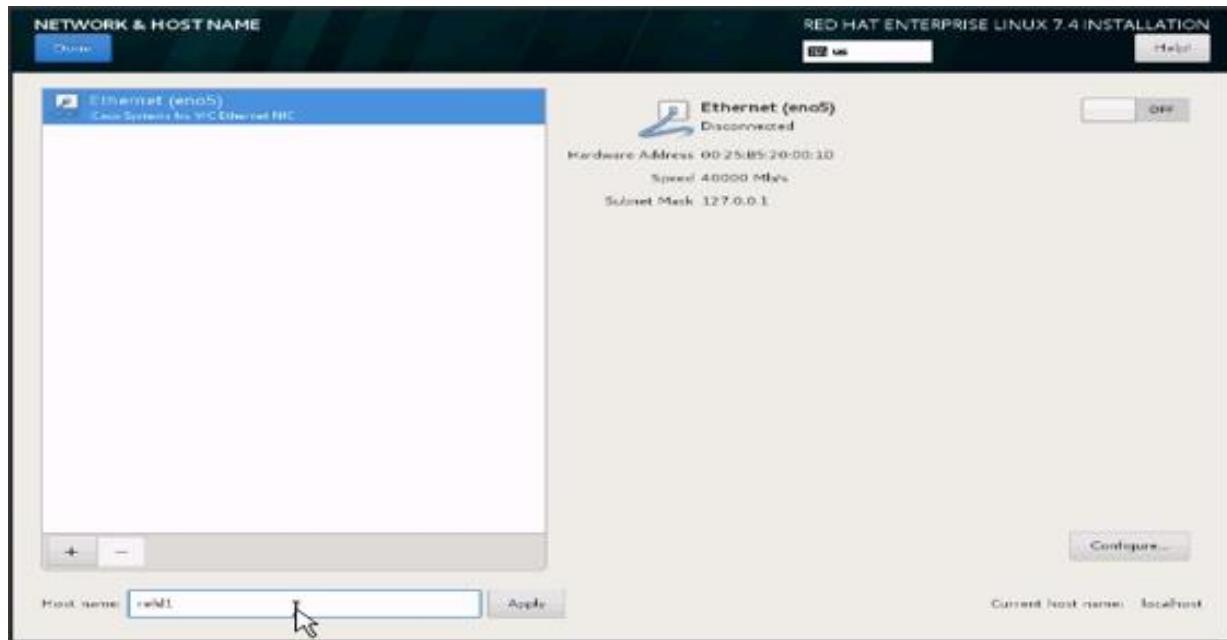
34. Select Infrastructure Server and select the Add-Ons as noted below. Click Done.



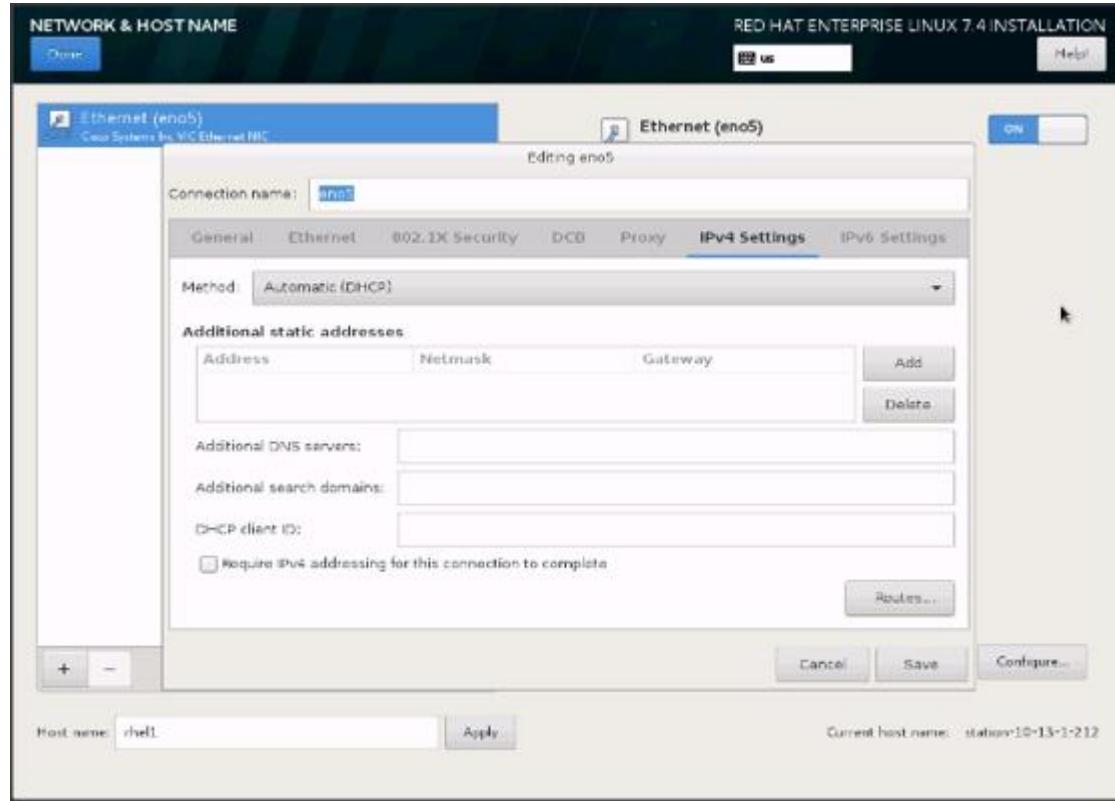
35. Click Network and Hostname and configure Hostname and Networking for the Host.



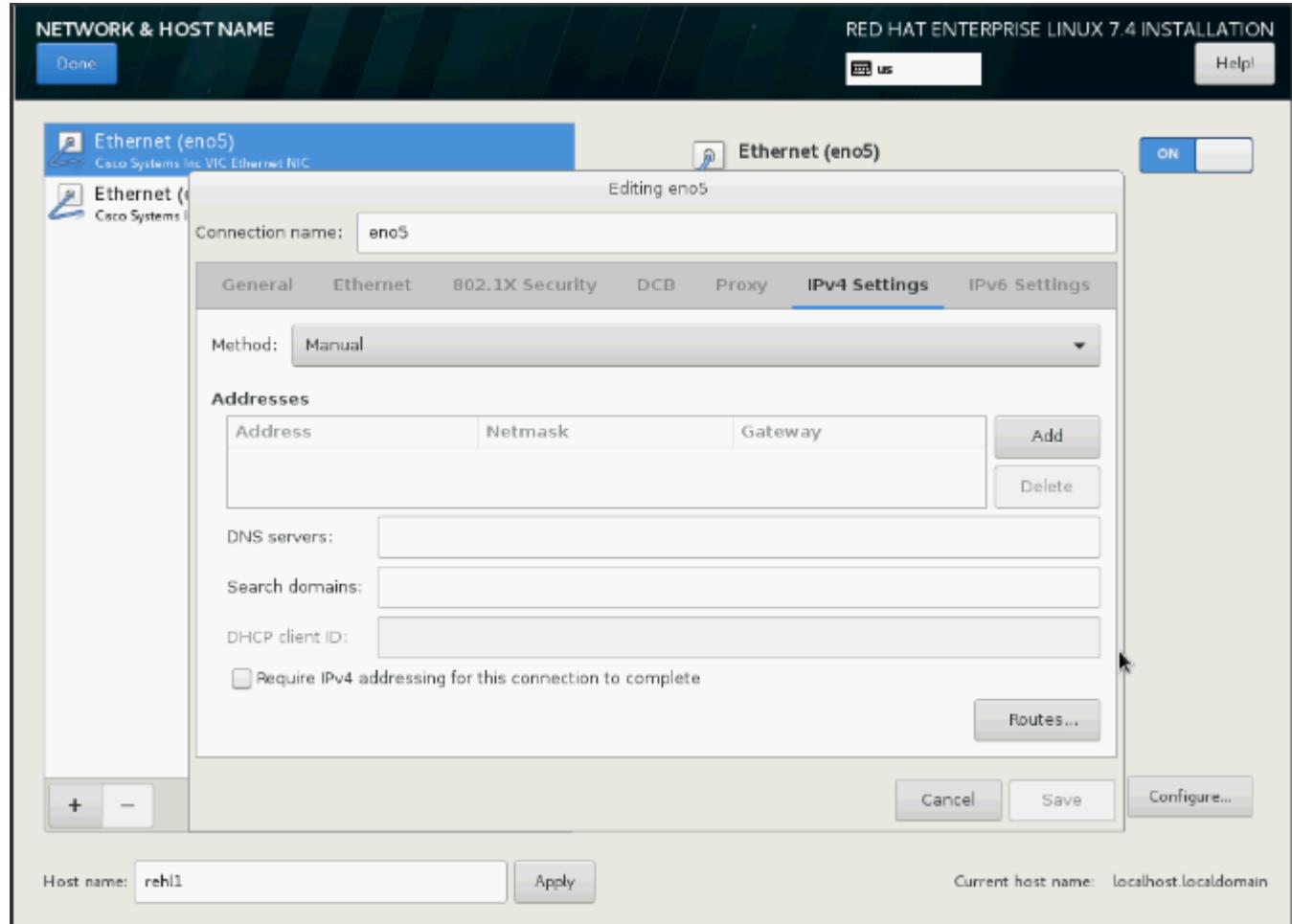
36. Type in the hostname as shown below.

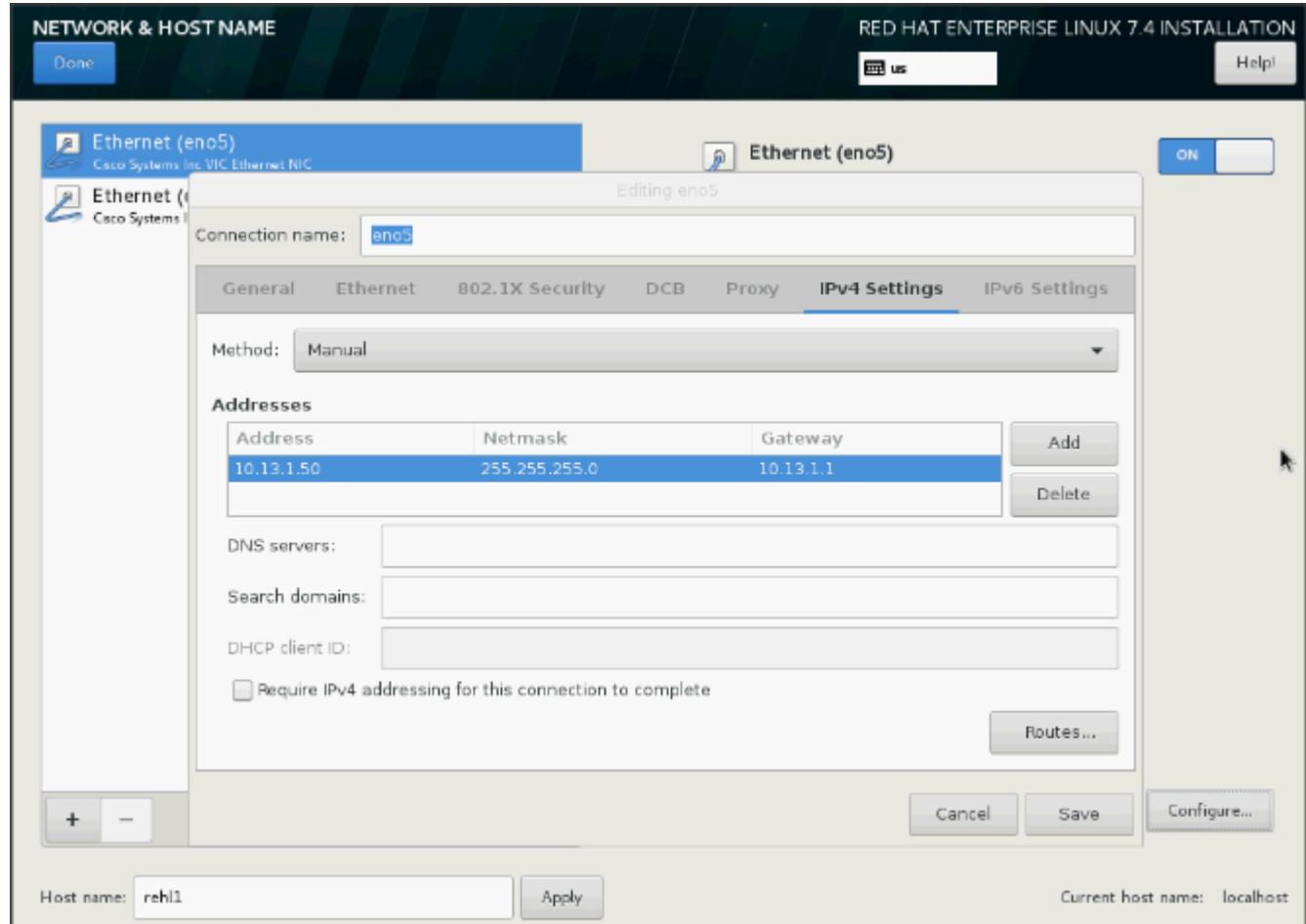


37. Click Configure to open the Network Connectivity window. Click IPV4Settings.

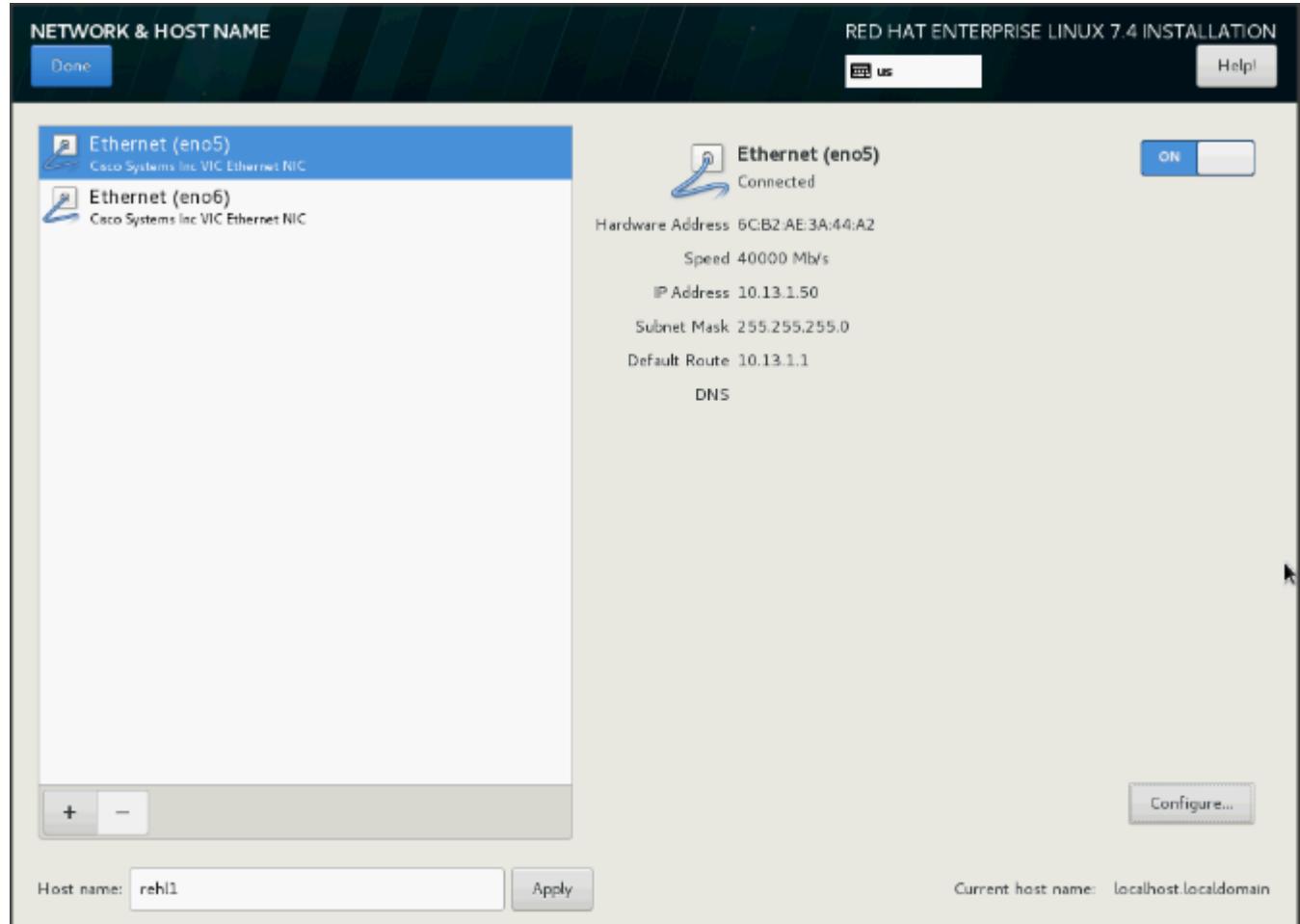


38. Change the Method to Manual and click Add to enter the IP Address, Netmask, and Gateway details.





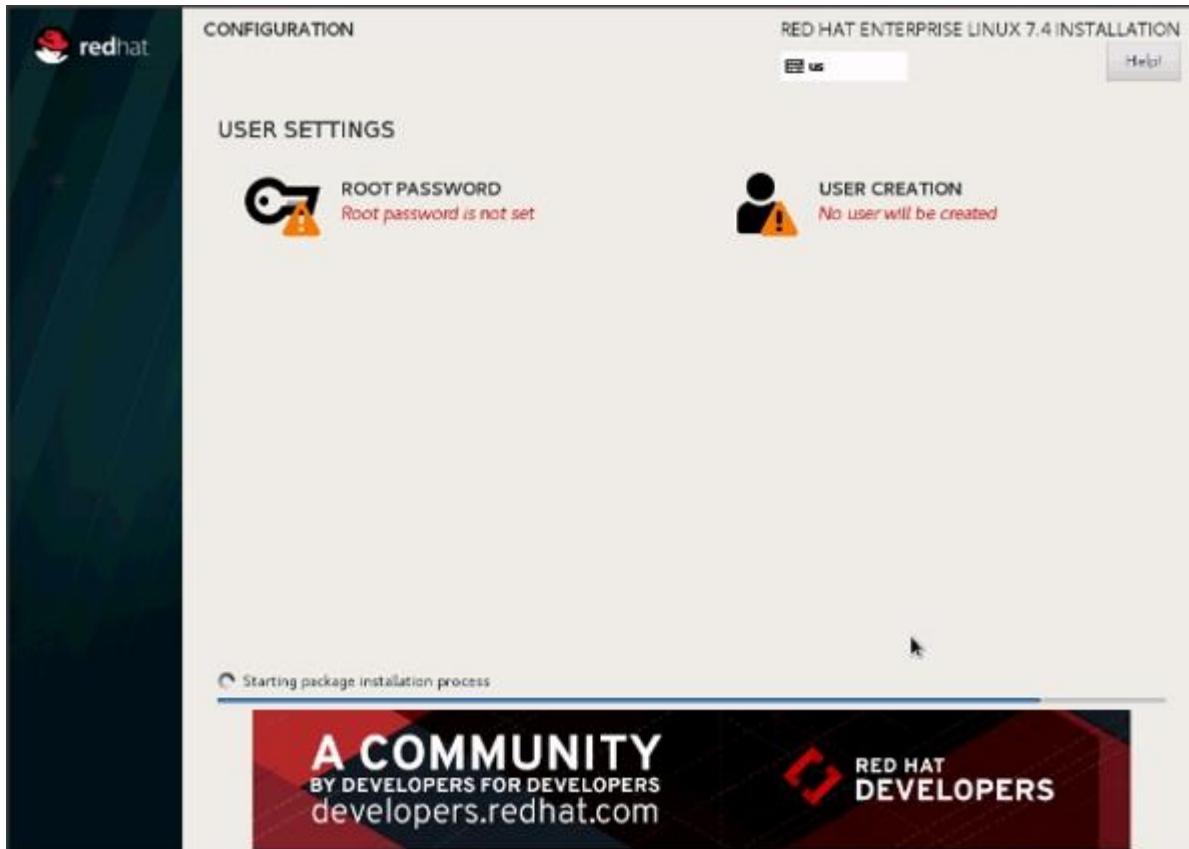
39. Click Save and update the hostname and turn Ethernet ON. Click Done to return to the main menu.



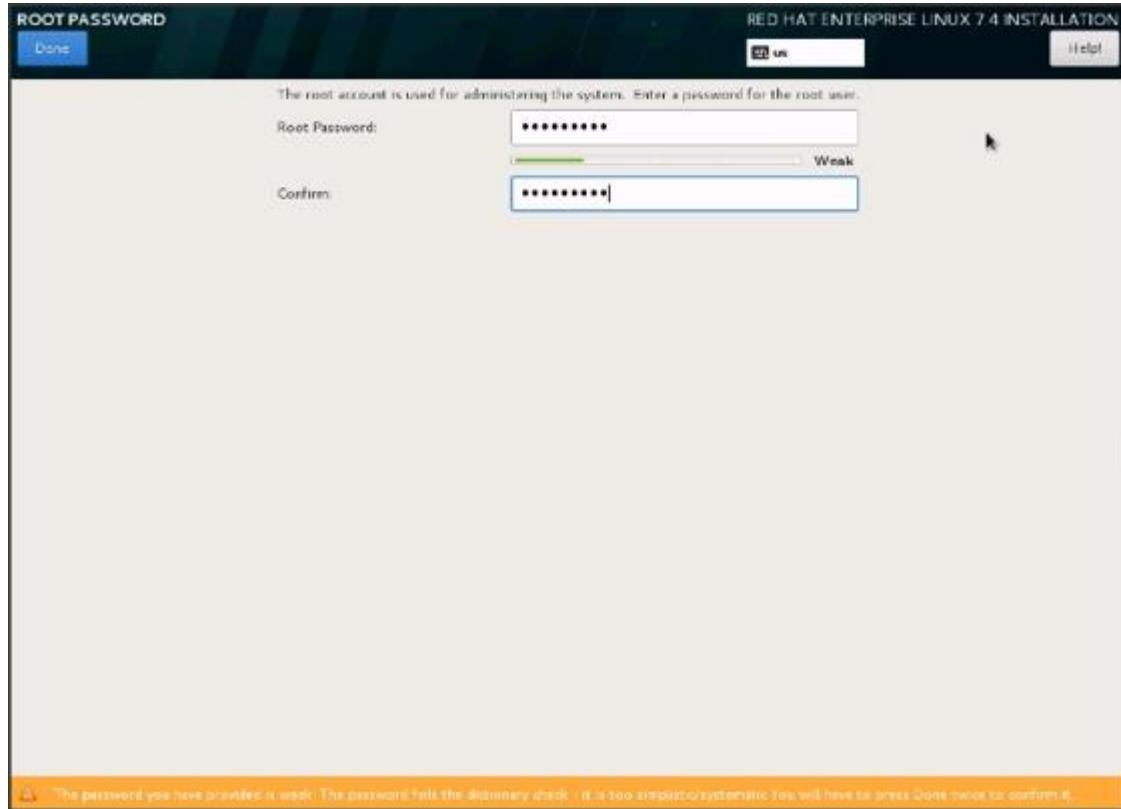
40. Click Begin Installation in the main menu.



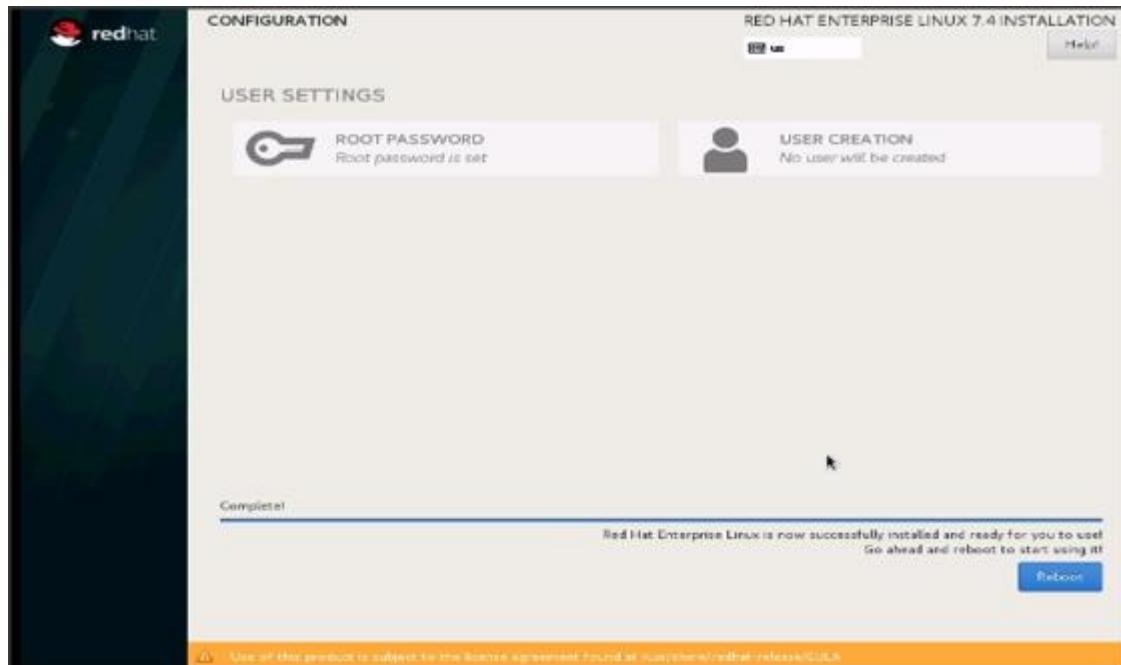
41. Select Root Password in the User Settings.



42. Enter the Root Password and click Done.



43. When the installation is complete reboot the system.



44. Repeat steps 1 to 43 to install Red Hat Enterprise Linux 7.4 on the remaining servers.



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

The hostnames and their corresponding IP addresses are shown in Table 6.

Table 6 Hostnames and IP Addresses

Hostname	etho
rhel1	10.13.1.50
rhel2	10.13.1.51
rhel3	10.13.1.52
rhel4	10.13.1.53
rhel1	10.13.1.54
rhel6	10.13.1.55
rhel7	10.13.1.56
rhel8	10.13.1.57
rhel9	10.13.1.58
rhel10	10.13.1.59
rhel11	10.13.1.60
rhel12	10.13.1.61
rhel13	10.13.1.62
rhel14	10.13.1.63
rhel15	10.13.1.64
rhel16	10.13.1.65
...	...
rhel24	10.13.1.73
cdsw1.cisco.com	10.13.1.250
Cdsw2.cisco.com	10.13.1.251
Cdsw3.cisco.com	10.13.1.252
Cdsw4.cisco.com	10.13.1.253



Note: Cloudera does not recommend multi-homing configurations, so please assign only one network to each node.

Post OS Install Configuration

Choose one of the nodes of the cluster or a separate node as the Admin Node for management such as CDH installation, cluster parallel shell, creating a local Red Hat repo and others. In this document, we use rhel1 for this purpose.

Setting Up Password-less Login

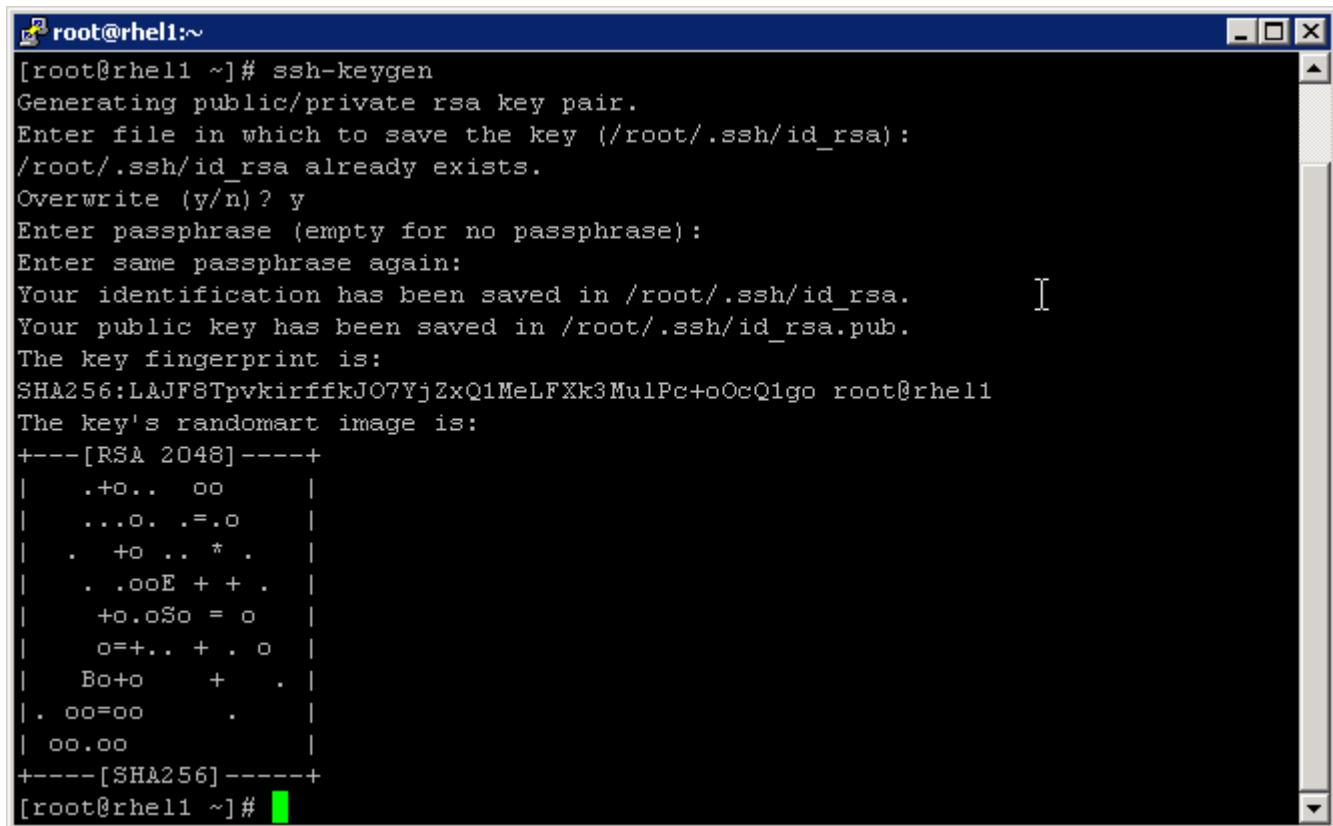
To manage all of the clusters nodes from the admin node password-less login needs to be setup. It assists in automating common tasks with clustershell (clush, a cluster wide parallel shell), and shell-scripts without having to use passwords.

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

1. Login to the Admin Node (rhel1).

```
#ssh 10.13.1.50
```

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



```
root@rhel1:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:LAJF8TpVkirffkJ07YjZxQ1MeLFXk3Mu1Pc+oOcQ1go root@rhel1
The key's randomart image is:
+---[RSA 2048]---+
|   .+o.. oo      |
|   ...o. .=.o    |
|   . +o .. * .   |
|   . .ooE + + .  |
|   +o.oSo = o    |
|   o=+.. + . o   |
|   Bo+o   +     . |
| . oo=oo       . |
| oo.oo          |
+---[SHA256]---+
[root@rhel1 ~]#
```

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

```
#for IP in {50..73}; do echo -n "$IP -> "; ssh-copy-id -i ~/.ssh/id_rsa.pub 10.13.1.$IP; done
```

```
#for IP in {250..253}; do echo -n "$IP -> "; ssh-copy-id -i
~/.ssh/id_rsa.pub 10.13.1.$IP; done
```

4. Enter yes for Are you sure you want to continue connecting (yes/no) ?
5. Enter the password of the remote host.

Configuring /etc/hosts

Setup /etc/hosts on the Admin node; this is a pre-configuration to setup DNS as shown in the next section.

To create the host file on the admin node, complete the following steps:

1. Populate the host file with IP addresses and corresponding hostnames on the Admin node (rhel1) and other nodes as follows:
2. On Admin Node (rhel1):

```
#vi /etc/hosts

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

10.13.1.50 rhel1
10.13.1.51 rhel2
10.13.1.52 rhel3
10.13.1.53 rhel4
10.13.1.54 rhel5
10.13.1.55 rhel6
10.13.1.56 rhel7
...
10.13.1.73 rhel24
10.13.1.250 cdsrw1.cisco.com
10.13.1.251 cdsrw2.cisco.com
10.13.1.252 cdsrw3.cisco.com
10.13.1.253 cdsrw4.cisco.com
```

Creating a Red Hat Enterprise Linux (RHEL) 7.4 Local Repo

To create a repository using RHEL DVD or ISO on the admin node (in this deployment rhel1 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 rhel1. Create a directory that would contain the repository.
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 rhel1.
4. Log back into rhel1 and create the mount directory.

```
#mkdir -p /var/www/html/rhelrepo
#scp rhel-server-7.4-x86_64-dvd.iso rhel1:/root/
#mkdir -p /mnt/rheliso
#mount -t iso9660 -o loop /root/rhel-server-7.4-x86_64-dvd.iso /mnt/rheliso/
```

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

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

6. On `rhel1` create a `.repo` file to enable the use of the `yum` command.

```
#vi /var/www/html/rhelrepo/rheliso.repo
[rhel7.4]
name=Red Hat Enterprise Linux 7.4
baseurl=http://10.13.1.50/rhelrepo
gpgcheck=0
enabled=1
```

7. Copy `rheliso.repo` file from `/var/www/html/rhelrepo` to `/etc/yum.repos.d` on `rhel1`.

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



Note: Based on this repo file yum requires httpd to be running on `rhel1` for other nodes to access the repository.

8. To make use of repository files on `rhel1` 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 (`rhel1`) using the `repo` (such as `httpd`, `createrepo`, etc.)

```
#vi /etc/yum.repos.d/rheliso.repo
[rhel7.4]
name=Red Hat Enterprise Linux 7.4
baseurl=file:///var/www/html/rhelrepo
gpgcheck=0
enabled=1
```

Creating the Red Hat Repository Database

1. Install the `createrepo` package on admin node (`rhel1`). Use it to regenerate the repository database(s) for the local copy of the RHEL DVD contents.

```
#yum -y install createrepo
```

```

root@rhel1:~#
[root@rhel1 ~]# vi /etc/yum.repos.d/rheliso.repo
[root@rhel1 ~]# yum -y install createrepo
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
rhel7.4
(1/2): rhel7.4/group_gz | 4.1 kB   00:00
(2/2): rhel7.4/primary_db | 137 kB   00:00
(2/2): rhel7.4/primary_db | 4.0 MB   00:00
Resolving Dependencies
--> Running transaction check
--> Package createrepo.noarch 0:0.9.9-28.el7 will be installed
--> Processing Dependency: deltarpm for package: createrepo-0.9.9-28.el7.noarch
--> Processing Dependency: python-deltarpm for package: createrepo-0.9.9-28.el7.noarch
--> Running transaction check
--> Package deltarpm.x86_64 0:3.6-3.el7 will be installed
--> Package python-deltarpm.x86_64 0:3.6-3.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version       Repository      Size
=====
Installing:
createrepo        noarch    0.9.9-28.el7   rhel7.4        94 k
Installing for dependencies:
deltarpm          x86_64   3.6-3.el7     rhel7.4        82 k
python-deltarpm   x86_64   3.6-3.el7     rhel7.4        31 k

Transaction Summary
=====

```

2. Run createrepo on the RHEL repository to create the repo database on admin node

```

#cd /var/www/html/rhelrepo
#createrepo .

[root@rhel1 rhelrepo]# createrepo .
Spawning worker 0 with 3763 pkgs
Workers Finished
Gathering worker results

Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete

```

Setting up ClusterShell

ClusterShell (or `clush`) is the cluster-wide shell that runs commands on several hosts in parallel. To set up the ClusterShell, complete the following steps:

1. From the system connected to the Internet download Cluster shell (`clush`) and install it on `rhel1`. Cluster shell is available from EPEL (Extra Packages for Enterprise Linux) repository.

```
#wget
http://rpm.pbone.net/index.php3/stat/4/idpl/31529309/dir/redhat_el_7/com/cluster-shell-1.7-1.el7.noarch.rpm.html

#scp cluster-shell-1.7-1.el7.noarch.rpm rhel1:/root/
Login to rhel1 and install cluster shell.
#yum -y install cluster-shell-1.7.1.el7.noarch.rpm
```

```
[root@rhel1 ~]# yum install cluster-shell-1.8-1.el7.noarch.rpm
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
Examining cluster-shell-1.8-1.el7.noarch.rpm: cluster-shell-1.8-1.el7.noarch
Marking cluster-shell-1.8-1.el7.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package cluster-shell.noarch 0:1.8-1.el7 will be installed
--> Processing Dependency: vim-common for package: cluster-shell-1.8-1.el7.noarch
--> Processing Dependency: vim-filesystem for package: cluster-shell-1.8-1.el7.noarch
--> Running transaction check
--> Package vim-common.x86_64 2:7.4.160-2.el7 will be installed
--> Package vim-filesystem.x86_64 2:7.4.160-2.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository          Size
=====
Installing:
cluster-shell     noarch   1.8-1.el7          /cluster-shell-1.8-1.el7.noarch    293 k
Installing for dependencies:
vim-common        x86_64   2:7.4.160-2.el7    rhel7.4             5.9 M
vim-filesystem   x86_64   2:7.4.160-2.el7    rhel7.4             9.8 k

Transaction Summary
=====
Install 1 Package (+2 Dependent packages)

Total size: 6.2 M
Total download size: 5.9 M
Installed size: 21 M
Is this ok [y/d/N]: y
Downloading packages:
=====
Total                                         279 MB/s | 5.9 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 2:vim-filesystem-7.4.160-2.el7.x86_64          1/3
  Installing : 2:vim-common-7.4.160-2.el7.x86_64             2/3
```

2. Edit /etc/cluster-shell/groups.d/local.cfg file to include hostnames for all the nodes of the cluster. This set of hosts is taken when running clush with the '-a' option.
3. For 28 node cluster as in our CVD, set groups file as follows:

```
#vi /etc/cluster-shell/groups.d/local.cfg
```

```
Complete!
[root@rhel1 ~]# vi /etc/cluster-shell/groups.d/local.cfg
[root@rhel1 ~]#
```

```
all: rhel[1-24],cdsw1.cisco.com, cdsw2.cisco.com, cdsw3.cisco.com,
cdsw4.cisco.com
```



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



Note: cluster-shell will not work if not ssh to the machine earlier (as it requires to be in known_hosts file), for instance, as in the case below for rhel<host>.

Installing httpd

Setting up RHEL repo on the admin node requires httpd. To set up RHEL repository on the admin node, complete the following steps:

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

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

```
#yum -y install httpd
```

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

```
#vi /etc/httpd/conf/httpd.conf
ServerName 10.13.1.50:80
```

3. Start httpd

```
#service httpd start
#chkconfig httpd on
```

Set Up All Nodes to use the RHEL Repository



Note: Based on this repo file yum requires httpd to be running on rhel1 for other nodes to access the repository.

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

```
#clush -w rhel[1-24],cdsw[1-4].cisco.com -b -c
/var/www/html/rhelrepo/rheliso.repo --dest=/etc/yum.repos.d/
```

2. Also copy the /etc/hosts file to all nodes.

```
#clush -a -b -c /etc/hosts --dest=/etc/hosts
```

3. Purge the yum caches after this

```
#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/
```

Configuring DNS

This section details setting up DNS using dnsmasq as an example based on the /etc/hosts configuration setup in the earlier section.

To create the host file across all the nodes in the cluster, complete the following steps:

1. Disable Network manager on all nodes.

```
#clush -a -b service NetworkManager stop  
#clush -a -b chkconfig NetworkManager off
```

2. Update /etc/resolv.conf file to point to Admin Node.

```
#vi /etc/resolv.conf  
  
nameserver 10.13.1.50
```



Note: Cloudera CDSW requires Wildcard entry dns configurations as detailed in section Cloudera Data Science Work-bench.



Note: Alternatively #systemctl start NetworkManager.service can be used to start the service. #systemctl stop NetworkManager.service can be used to stop the service. Use #systemctl disable NetworkManager.service to stop a service from being automatically started at boot time.

3. Install and Start dnsmasq on Admin node

```
#service dnsmasq start  
  
#chkconfig dnsmasq on
```

4. Deploy /etc/resolv.conf from the admin node (rhel1) to all the nodes via the following clush command:

```
#clush -a -B -c /etc/resolv.conf
```



Note: A clush copy without –dest copies to the same directory location as the source-file directory.

5. Make sure DNS is working fine by running the following command on Admin node and any data-node

```
[root@rhel2 ~]# nslookup rhel1  
Server: 10.13.1.50  
Address: 10.13.1.50#53  
Name: rhel1  
Address: 10.13.1.50 ↵
```



Note: yum install -y bind-utils will need to be run for nslookup to utility to run.

Upgrading the Cisco Network Driver for VIC1387

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/home/286318800/type/283853158/release/3.1%25283%2529>

In the ISO image, the required driver kmod-enic-2.3.0.44-rhel7u4.el7.x86_64.rpm can be located at \\Linux\\Network\\Cisco\\VIC\\RHEL\\RHEL7.4.

1. From a node connected to the Internet, download, extract and transfer kmod-enic-2.3.0.44-rhel7u4.el7.x86_64.rpm to rhel1(admin node).

2. Install the rpm on all nodes of the cluster using the following clash commands. For this example, the rpm is assumed to be in present working directory of rhel1.
3. [root@rhel1 ~]# clash -a -b -c kmod-enic-2.3.0.44-rhel7u4.el7.x86_64.rpm
4. [root@rhel1 ~]# clash -a -b "rpm -ivh kmod-enic-2.3.0.44-rhel7u4.el7.x86_64.rpm "
5. Make sure that the above installed version of kmod-enic driver is being used on all nodes by running the command "modinfo enic" on all nodes

```
[root@rhel1 ~]# clash -a -B "modinfo enic | head -5"
```
6. Also it is recommended to download the kmod-megaraid driver for higher performance , the RPM can be found in the same package at \Linux\Storage\LSI\Cisco_Storage_12G_SAS_RAID_controller\RHEL\RHEL7.4

Installing xfsprogs

From the admin node rhel1 run the command shown below to Install xfsprogs on all the nodes for xfs filesystem.

```
#clush -a -B yum -y install xfsprogs
[root@rhel1 ~]# clash -a -B yum -y install xfsprogs
```

NTP Configuration

The Network Time Protocol (NTP) is used to synchronize the time of all the nodes within the cluster. The Network Time Protocol daemon (ntpd) sets and maintains the system time of day in synchronism with the timeserver located in the admin node (rhel1). Configuring NTP is critical for any Hadoop Cluster. If server clocks in the cluster drift out of sync, serious problems will occur with HBase and other services.

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



Note: Installing an internal NTP server keeps your cluster synchronized even when an outside NTP server is inaccessible.

1. Configure /etc/ntp.conf on the admin node only 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
```

2. Create /root/ntp.conf on the admin node and copy it to all nodes:

```
#vi /root/ntp.conf
server 10.13.1.50
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
restrict -6 ::1
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

3. Copy ntp.conf file from the admin node to /etc of all the nodes by executing the following commands in the admin node (rhel1)

```
#for SERVER in {50..73}; do scp /root/ntp.conf  
10.13.1.$SERVER:/etc/ntp.conf; done  
#for SERVER in {250..253}; do scp /root/ntp.conf  
10.13.1.$SERVER:/etc/ntp.conf; done
```

ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00
ntp.conf	100%	136	0.1KB/s	00:00

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

```
#clush -a -b "service ntpd stop"  
#clush -a -b "ntpdate rhel1"  
#clush -a -b "service ntpd start"
```

5. Make sure to restart of NTP daemon across reboots:

```
#clush -a -b "systemctl enable ntpd"
```



Alternatively, the new Chrony service can be installed, that is quicker to synchronize clocks in mobile and virtual systems.

6. Install the Chrony service:

```
# yum install -y chrony
```

7. Activate the Chrony service at boot:

8. # systemctl enable chronyd

9. Start the Chrony service:

```
# systemctl start chronyd
```

The Chrony configuration is in the /etc/chrony.conf file, configured similar to /etc/ntp.conf.

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  
#clush -B -a service rsyslog status
```

Setting ulimit

On each node, `ulimit -n` specifies the number of inodes that can be opened simultaneously. With the default value of 1024, the system appears to be out of disk space and shows no inodes available. This value should be set to 64000 on every node.

Higher values are unlikely to result in an appreciable performance gain.

- For setting the ulimit on Redhat, edit `/etc/security/limits.conf` on admin node `rhel1` and add the following lines:

```
root soft nofile 64000
root hard nofile 64000
[root@rhel1 ~]# cat /etc/security/limits.conf | grep 64000
root soft nofile 64000
root hard nofile 64000
```

- Copy the `/etc/security/limits.conf` file from admin node (`rhel1`) to all the nodes using the following command.

```
#clush -a -b -c /etc/security/limits.conf --dest=/etc/security/
[root@rhel1 ~]# clush -a -b -c /etc/security/limits.conf --dest=/etc/security/
```

- Make sure that the `/etc/pam.d/su` file contains the following settings:

```
#%PAM-1.0

auth      sufficient      pam_rootok.so

# Uncomment the following line to implicitly trust users in the "wheel" group.

#auth      sufficient      pam_wheel.so trust use_uid

# Uncomment the following line to require a user to be in the "wheel" group.

#auth      required       pam_wheel.so use_uid

auth      include        system-auth

account   sufficient     pam_succeed_if.so uid = 0 use_uid quiet

account   include        system-auth

password  include        system-auth

session   include        system-auth

session   optional       pam_xauth.so
```



Note: The ulimit values are applied on a new shell, running the command on a node on 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 nodes.

```
#clush -a -b "sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config"
[root@rhel1 ~]# clush -a -b "sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config "
#clush -a -b "setenforce 0"
```



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

Reboot the machine, if needed for SELinux to be disabled in case it does not take effect. It can be checked using
`#clush -a -b sestatus`

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 `rhel1` and add the following lines:

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

2. Copy the `/etc/sysctl.conf` file from admin node (`rhel1`) to all the nodes using the following command:

```
#clush -a -b -c /etc/sysctl.conf --dest=/etc/
```

3. 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 Hadoop deployment. Since the UCS Big Data deployment will be in its own isolated network there is no need for that additional firewall.

```
#clush -a -b "firewall-cmd --zone=public --add-port=80/tcp --permanent"
#clush -a -b "firewall-cmd --reload"
#clush -a -b "systemctl disable firewalld"
```

Disable Swapping

1. In order to reduce Swapping, run the following on all nodes. Variable `vm.swappiness` defines how often swap should be used, 60 is default.

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

2. 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.

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

1. The above commands must be run for every reboot, so copy this command to /etc/rc.local so they are executed automatically for every reboot.
2. On the Admin node, run the following commands:

```
#rm -f /root/thp_disable

#echo "echo never > /sys/kernel/mm/transparent_hugepage/enabled" >>
/root/thp_disable

#echo "echo never > /sys/kernel/mm/transparent_hugepage/defrag" >>
/root/thp_disable
```

3. Copy file to each node:

```
#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"
```

Disable IPv6 Defaults

1. 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"
```

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

```
#clush -a -b "sysctl -p"
```

Configuring Data Drives on Name Node And Other Management 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 will not be used for HDFS data.

1. From the website download storcli <https://www.broadcom.com/support/download-search/?pg=&pf=&pn=&po=&pa=&dk=storcli>
2. Extract the zip file and copy storcli-007.0504.0000.0000-1.noarch.rpm from the linux directory.
3. Download storcli and its dependencies and transfer to Admin node.

```
#scp storcli-007.0504.0000.0000-1.noarch.rpm rhel1:/root/
```

4. Copy storcli rpm to all the nodes using the following commands:

```
#clush -a -b -c /root/storcli-007.0504.0000.0000-1.noarch.rpm --dest=/root/
```

5. Run the below command to install storcli on all the nodes

```
#clush -a -b "rpm -ivh storcli-007.0504.0000.0000-1.noarch.rpm"
```

6. Run the below command to copy storcli64 to root directory.

```
#cd /opt/MegaRAID/storcli/
#cp storcli64 /root/
```

```
[root@rhel1 ~]# cd /opt/MegaRAID/storcli/
[root@rhel1 storcli]# ls
install.log  libstorelibir-2.so  libstorelibir-2.so.14.07-0  storcli64
[root@rhel1 storcli]# cp storcli64 /root/
```

7. Copy storcli64 to all the nodes using the following commands:

```
#clush -a -b -c /root/storcli64 --dest=/root/
```

8. Run the following script as root user on rhel1 to rhel3 to create the virtual drives for the management nodes.

```
#vi /root/raid1.sh
./storcli64 -cfgldadd
r1[$1:1,$1:2,$1:3,$1:4,$1:5,$1:6,$1:7,$1:8,$1:9,$1:10,$1:11,$1:12,$1:13,$1:14,
$1:15,$1:16,$1:17,$1:18,$1:19,$1:20,$1:21,$1:22,$1:23,$1:24,$1:25,$1:26] wb ra
nocachedbadbbu strpsz1024 -a0
```

The above script requires enclosure ID as a parameter.

9. Run the following command to get enclosure id.

```
./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort |
uniq -c | awk '{print $2}'
#chmod 755 raid1.sh
```

10. Run MegaCli script as follows.

```
./raid1.sh <EnclosureID> obtained by running the command above
WB: Write back
RA: Read Ahead
NoCachedBadBBU: Do not write cache when the BBU is bad.
Strpsz1024: Strip Size of 1024K
```



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

Cloudera recommends the following disk configuration for the master nodes:

- At least 10 physical disks in following configuration
- 2 x RAID1 OS (Root disk)
- 4 x RAID 10 (DB filesystems)
- 2 x RAID 1 HDFS NameNode metadata
- 1 x JBOD - ZooKeeper
- 1 x JBOD - Quorum JournalNode

Configuring Data Drives on Data Nodes

To configure non-OS disk drives as individual RAID0 volumes using StorCli command, complete the following steps. These volumes are going to be used for HDFS Data.

1. Issue the following command from the admin node to create the virtual drives with individual RAID 0 configurations on all the data nodes.

```
#clush -w rhel[4-24] -B ./storcli64 -cfgeachdskraid0 WB RA direct
NoCachedBadBBU strpsz1024 -a0
WB: Write back
RA: Read Ahead
NoCachedBadBBU: Do not write cache when the BBU is bad.
Strpsz1024: Strip Size of 1024K
```



Note: Create Raid 1 for the Cloudera Data Science workbench nodes as shown as management nodes.



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

Configuring the Filesystem for NameNodes and Datanodes

The following script will format and mount the available volumes on each node whether it is Namenode or Data node. OS boot partition is going to be skipped. All drives are mounted based on their UUID as /data/disk1, /data/disk2, etc. To configure the filesystem for NameNodes and DataNodes, complete the following steps:

1. On the Admin node, create a file containing the following script.
2. 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, delete them before running the script. This process is documented in the "Note" section at the end of the section.

```
#vi /root/driveconf.sh
#!/bin/bash
[[ "-x" == "${1}" ]] && set -x && set -v && shift 1
count=1
for X in /sys/class/scsi_host/host?/scan
do
echo '---' > ${X}
done
for X in /dev/sd?
do
list+=$(echo $X " ")
done
for X in /dev/sd??
do
list+=$(echo $X " ")
done
for X in $list
```

```

do
echo "====="
echo $X
echo "====="
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 "Formatting and Mounting Drive => ${X}"
166
/sbin/mkfs.xfs -f ${X}
(( $? )) && continue
#Identify UUID
UUID=`blkid ${X} | cut -d " " -f2 | cut -d "=" -f2 | sed 's//g'`
/bin/mkdir -p /data/disk${count}
(( $? )) && continue
echo "UUID of ${X} = ${UUID}, mounting ${X} using UUID on
/data/disk${count}"
/bin/mount -t xfs -o inode64,noatime,nobarrier -U ${UUID}
/data/disk${count}
(( $? )) && continue
echo "UUID=${UUID} /data/disk${count} xfs inode64,noatime,nobarrier 0
0" >> /etc/fstab
((count++))
fi
done

```



Note: Do not run this script on the Cloudera Data Science Workbench nodes

- Run the following command to copy driveconf.sh to all the nodes:

```
#chmod 755 /root/driveconf.sh
#clush -a -B -c /root/driveconf.sh
```

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

```
#clush -a -B /root/driveconf.sh
```

- Run the following from the admin node to list the partitions and mount points

```
#clush -a -B df -h
#clush -a -B mount
#clush -a -B cat /etc/fstab
```



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

- Run the mount 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.



Note: Care should be taken **not to delete the OS partition** as this will wipe out the OS.

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

Cluster Verification

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

1. Create the script `cluster_verification.sh` as shown, on the Admin node (rhel1).

```
#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 and An-
alytics \ Cluster Verification === ${NC}"
echo ""
echo ""
echo -e "${green} === System Information === ${NC}"
echo ""
echo ""
echo -e "${green}System ${NC}"
clush -a -B " `which dmidecode` | grep -A2 '^System Information'"
echo ""
echo ""
echo -e "${green}BIOS ${NC}"
clush -a -B " `which dmidecode` | grep -A3 '^BIOS I'"
echo ""
echo ""
echo -e "${green}Memory ${NC}"
clush -a -B "cat /proc/meminfo | grep -i ^memt | uniq"
echo ""
echo ""
echo -e "${green}Number of Dimms ${NC}"
clush -a -B "echo -n 'DIMM slots: ' ; `which dmidecode` | grep -c \
'^[:space:]*Locator:'"
clush -a -B "echo -n 'DIMM count is: ' ; `which dmidecode` | grep \ "Size"|
grep -c "MB"""
clush -a -B " `which dmidecode` | awk '/Memory Device$/,{print}' |\
grep -e '^Mem' -e Size: -e Speed: -e Part | sort -u | grep -v -e 'NO \ DIMM' -e 'No
Module Installed' -e Unknown"
echo ""
echo ""
# probe for cpu info #
echo -e "${green}CPU ${NC}"
clush -a -B "grep '^model name' /proc/cpuinfo | sort -u"
echo ""
clush -a -B " `which lscpu` | grep -v -e op-mode -e ^Vendor -e family -e\ Mod-
el: -e Stepping: -e BogoMIPS -e Virtual -e ^Byte -e '^NUMA node(s)'"
echo ""
echo ""
```

```

# probe for nic info #
echo -e "${green}NIC ${NC}"
clush -a -B `which ifconfig` | egrep '^(e|p)' | awk '{print \$1}' | \ xargs
-l `which ethtool` | grep -e ^Settings -e Speed"
echo ""
clush -a -B `which lspci` | grep -i ether"
echo ""
echo ""
# probe for disk info #
echo -e "${green}Storage ${NC}"
clush -a -B "echo 'Storage Controller: '; `which lspci` | grep -i -e \ raid -e
storage -e lsi"
echo ""
clush -a -B "dmesg | grep -i raid | grep -i scsi"
echo ""
clush -a -B "lsblk -id | awk '{print \$1,\$4}'|sort | nl"
echo ""
echo ""

echo -e "${green} ===== Software ===== ${NC}"
echo ""
echo ""
echo -e "${green}Linux Release ${NC}"
clush -a -B "cat /etc/*release | uniq"
echo ""
echo ""
echo -e "${green}Linux Version ${NC}"
clush -a -B "uname -srvm | fmt"
echo ""
echo ""
echo -e "${green}Date ${NC}"
clush -a -B date
echo ""
echo ""
echo -e "${green}NTP Status ${NC}"
clush -a -B "ntpstat 2>&1 | head -1"
echo ""
echo ""
echo -e "${green}SELINUX ${NC}"
clush -a -B "echo -n 'SELinux status: ' ; grep ^SELINUX= \ /etc/selinux/config
2>&1"
echo ""
echo ""
clush -a -B "echo -n 'CPUspeed Service: ' ; `which service` cpuspeed \ status
2>&1"
clush -a -B "echo -n 'CPUspeed Service: ' ; `which chkconfig` --list \ cpuspeed
2>&1"
echo ""
echo ""
echo -e "${green}Java Version${NC}"
clush -a -B 'java -version 2>&1; echo JAVA_HOME is ${JAVA_HOME}:Not \ De-
fined!}'
echo ""
echo ""
echo -e "${green}Hostname LoOKup${NC}"
clush -a -B " ip addr show"
echo ""
echo ""

```

```
echo -e "${green}Open File Limit${NC}"
clush -a -B 'echo -n "Open file limit(should be >32K): "; ulimit -n'
```

2. Change permissions to executable.

```
chmod 755 cluster_verification.sh
```

3. Run the Cluster Verification tool from the admin node. This can be run before starting Hadoop to identify any discrepancies in Post OS Configuration between the servers or during troubleshooting of any cluster / Hadoop issues.

```
./cluster_verification.sh
```

Installing Cloudera

Cloudera's Distribution including Apache Hadoop (CDH) is an enterprise grade, hardened Hadoop distribution. CDH offers Apache Hadoop and several related projects into a single tested and certified product. It offers the latest innovations from the open source community with the testing and quality expected from enterprise quality software.

Prerequisites for CDH Installation

This section details the prerequisites for CDH installation such as setting up CDH Repo.

Cloudera Manager Repository

1. From a host connected to the Internet, download the Cloudera's repositories as shown below and transfer it to the admin node.

```
#mkdir -p /tmp/clouderarepo/
```

2. Download Cloudera Manager Repository.

```
#cd /tmp/clouderarepo/
#wget http://archive.cloudera.com/cm5/redhat/7/x86_64/cm/cloudera-manager.repo
#reposync --config=./cloudera-manager.repo --repoid=cloudera-manager
```

This downloads the Cloudera Manager RPMs needed for the Cloudera repository.

3. Run the following command to move the RPMs

4. Copy the repository directory to the admin node (rhel1)

```
#scp -r /tmp/clouderarepo/ rhel1:/var/www/html/
```

5. On admin node (rhel1) run create repo command.

```
#cd /var/www/html/clouderarepo/
#createrepo --baseurl http://10.13.1.50/clouderarepo/cloudera-manager/
/var/www/html/clouderarepo/cloudera-manager
```

```

Spawning worker 33 with 0 pkgs
Spawning worker 40 with 0 pkgs
Spawning worker 41 with 0 pkgs
Spawning worker 42 with 0 pkgs
Spawning worker 43 with 0 pkgs
Spawning worker 44 with 0 pkgs
Spawning worker 45 with 0 pkgs
Spawning worker 46 with 0 pkgs
Spawning worker 47 with 0 pkgs
Spawning worker 48 with 0 pkgs
Spawning worker 49 with 0 pkgs
Spawning worker 50 with 0 pkgs
Spawning worker 51 with 0 pkgs
Spawning worker 52 with 0 pkgs
Spawning worker 53 with 0 pkgs
Spawning worker 54 with 0 pkgs
Spawning worker 55 with 0 pkgs
Workers Finished
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete

```



Note: Visit <http://10.13.1.50/clouderarepo/> to verify the files.

6. Create the Cloudera Manager repo file with following contents:

```
#vi /var/www/html/clouderarepo/cloudera-manager/cloudera-manager.repo
[cloudera-manager]
name=Cloudera Manager
baseurl=http://10.13.1.50/clouderarepo/cloudera-manager/
gpgcheck=0
enabled=1
```

7. Copy the file cloudera-manager.repo into /etc/yum.repos.d/ on the admin node to enable it to find the packages that are locally hosted.

```
#cp /var/www/html/clouderarepo/cloudera-manager/cloudera-manager.repo
/etc/yum.repos.d/
```

8. From the admin node copy the repo files to /etc/yum.repos.d/ of all the nodes of the cluster.

```
#clush -a -B -c /etc/yum.repos.d/cloudera-manager.repo
```

Setting Up the Local Parcels for CDH 5.13.0

From a host connected the internet, download the appropriate CDH 5.13.0 parcels that are meant for RHEL7.4 from the URL: <http://archive.cloudera.com/cdh5/parcels/> and place them in the directory "/var/www/html/CDH5.13.0parcels" of the Admin node.

The following are the relevant files for RHEL7.4:

- CDH-5.13.0-1.cdh5.13.0.po.29-el7.parcel
- CDH-5.13.0-1.cdh5.13.0.po.29-el7.parcel.sha1 and
- manifest.json

Downloading Parcels

From a host connected to the Internet, download the Cloudera's parcels as shown below and transfer it to the admin node.

```
#mkdir -p /tmp/clouderarepo/CDH5.13.0parcels
```

1. Download parcels:

```
#cd /tmp/clouderarepo/CDH5.13.0parcels
#wget http://archive.cloudera.com/cdh5/parcels/5.13.0/CDH-5.13.0-
1.cdh5.13.0.p0.29-el7.parcel
#wget http://archive.cloudera.com/cdh5/parcels/5.13.0/CDH-5.13.0-
1.cdh5.13.0.p0.29-el7.parcel.sha1
#wget http://archive.cloudera.com/cdh5/parcels/5.13.0/manifest.json
```

2. Now edit the /tmp/clouderarepo/CDH5.13.0parcels/manifest.json file and remove the scripts that are not meant for RHEL7.4. Below is that script which can be copy and pasted.



Note: Please make sure the script starts and end with initial additional braces.

```
{
  "lastUpdated": 15075981980000,
  "parcels": [
    {
      "parcelName": "CDH-5.13.0-1.cdh5.13.0.p0.29-el7.parcel",
      "components": [
        {
          "pkg_version": "0.7.0+cdh5.13.0+0",
          "pkg_release": "1.cdh5.13.0.p0.34",
          "name": "bigtop-tomcat",
          "version": "6.0.53-cdh5.13.0"
        },
        {
          "pkg_version": "0.11.0+cdh5.13.0+101",
          "pkg_release": "1.cdh5.13.0.p0.34",
          "name": "crunch",
          "version": "0.11.0-cdh5.13.0"
        },
        {
          "pkg_version": "1.6.0+cdh5.13.0+169",
          "pkg_release": "1.cdh5.13.0.p0.34",
          "name": "flume-ng",
          "version": "1.6.0-cdh5.13.0"
        },
        {
          "pkg_version": "2.6.0+cdh5.13.0+2639",
          "pkg_release": "1.cdh5.13.0.p0.34",
          "name": "hadoop-0.20-mapreduce",
          "version": "2.6.0-cdh5.13.0"
        },
        {
          "pkg_version": "2.6.0+cdh5.13.0+2639",
          "pkg_release": "1.cdh5.13.0.p0.34",
          "name": "hadoop",
          "version": "2.6.0-cdh5.13.0"
        },
        {
          "pkg_version": "2.6.0+cdh5.13.0+2639",
          "pkg_release": "1.cdh5.13.0.p0.34",
          "name": "hadoop-hdfs",
          "version": "2.6.0-cdh5.13.0"
        }
      ]
    }
  ]
}
```

```

    "pkg_version": "2.6.0+cdh5.13.0+2639",
    "pkg_release": "1.cdh5.13.0.p0.34",
    "name": "hadoop-httpfs",
    "version": "2.6.0-cdh5.13.0"
},
{
    "pkg_version": "2.6.0+cdh5.13.0+2639",
    "pkg_release": "1.cdh5.13.0.p0.34",
    "name": "hadoop-kms",
    "version": "2.6.0-cdh5.13.0"
},
{
    "pkg_version": "2.6.0+cdh5.13.0+2639",
    "pkg_release": "1.cdh5.13.0.p0.34",
    "name": "hadoop-mapreduce",
    "version": "2.6.0-cdh5.13.0"
},
{
    "pkg_version": "2.6.0+cdh5.13.0+2639",
    "pkg_release": "1.cdh5.13.0.p0.34",
    "name": "hadoop-yarn",
    "version": "2.6.0-cdh5.13.0"
},
{
    "pkg_version": "1.2.0+cdh5.13.0+411",
    "pkg_release": "1.cdh5.13.0.p0.34",
    "name": "hbase",
    "version": "1.2.0-cdh5.13.0"
},
{
    "pkg_version": "1.5+cdh5.13.0+71",
    "pkg_release": "1.cdh5.13.0.p0.34",
    "name": "hbase-solr",
    "version": "1.5-cdh5.13.0"
},
{
    "pkg_version": "1.1.0+cdh5.13.0+1269",
    "pkg_release": "1.cdh5.13.0.p0.34",
    "name": "hive",
    "version": "1.1.0-cdh5.13.0"
},
{
    "pkg_version": "1.1.0+cdh5.13.0+1269",
    "pkg_release": "1.cdh5.13.0.p0.34",
    "name": "hive-hcatalog",
    "version": "1.1.0-cdh5.13.0"
},
{
    "pkg_version": "3.9.0+cdh5.13.0+7079",
    "pkg_release": "1.cdh5.13.0.p0.34",
    "name": "hue",
    "version": "3.9.0-cdh5.13.0"
},
{
    "pkg_version": "2.10.0+cdh5.13.0+0",
    "pkg_release": "1.cdh5.13.0.p0.34",
    "name": "impala",
    "version": "2.10.0-cdh5.13.0"
}

```

```

},
{
  "pkg_version": "1.0.0+cdh5.13.0+145",
  "pkg_release": "1.cdh5.13.0.p0.34",
  "name": "kite",
  "version": "1.0.0-cdh5.13.0"
},
{
  "pkg_version": "1.5.0+cdh5.13.0+0",
  "pkg_release": "1.cdh5.13.0.p0.34",
  "name": "kudu",
  "version": "1.5.0-cdh5.13.0"
},
{
  "pkg_version": "1.0.0+cdh5.13.0+0",
  "pkg_release": "1.cdh5.13.0.p0.34",
  "name": "llama",
  "version": "1.0.0-cdh5.13.0"
},
{
  "pkg_version": "0.9+cdh5.13.0+34",
  "pkg_release": "1.cdh5.13.0.p0.34",
  "name": "mahout",
  "version": "0.9-cdh5.13.0"
},
{
  "pkg_version": "4.1.0+cdh5.13.0+458",
  "pkg_release": "1.cdh5.13.0.p0.34",
  "name": "oozie",
  "version": "4.1.0-cdh5.13.0"
},
{
  "pkg_version": "1.5.0+cdh5.13.0+191",
  "pkg_release": "1.cdh5.13.0.p0.34",
  "name": "parquet",
  "version": "1.5.0-cdh5.13.0"
},
{
  "pkg_version": "0.12.0+cdh5.13.0+110",
  "pkg_release": "1.cdh5.13.0.p0.34",
  "name": "pig",
  "version": "0.12.0-cdh5.13.0"
},
{
  "pkg_version": "1.5.1+cdh5.13.0+410",
  "pkg_release": "1.cdh5.13.0.p0.34",
  "name": "sentry",
  "version": "1.5.1-cdh5.13.0"
},
{
  "pkg_version": "4.10.3+cdh5.13.0+519",
  "pkg_release": "1.cdh5.13.0.p0.34",
  "name": "solr",
  "version": "4.10.3-cdh5.13.0"
},
{
  "pkg_version": "1.6.0+cdh5.13.0+530",
  "pkg_release": "1.cdh5.13.0.p0.34",

```

```

        "name": "spark",
        "version": "1.6.0-cdh5.13.0"
    },
    {
        "pkg_version": "1.99.5+cdh5.13.0+46",
        "pkg_release": "1.cdh5.13.0.p0.34",
        "name": "sqoop2",
        "version": "1.99.5-cdh5.13.0"
    },
    {
        "pkg_version": "1.4.6+cdh5.13.0+116",
        "pkg_release": "1.cdh5.13.0.p0.34",
        "name": "sqoop",
        "version": "1.4.6-cdh5.13.0"
    },
    {
        "pkg_version": "0.9.0+cdh5.13.0+23",
        "pkg_release": "1.cdh5.13.0.p0.34",
        "name": "whirr",
        "version": "0.9.0-cdh5.13.0"
    },
    {
        "pkg_version": "3.4.5+cdh5.13.0+118",
        "pkg_release": "1.cdh5.13.0.p0.34",
        "name": "zookeeper",
        "version": "3.4.5-cdh5.13.0"
    }
],
"conflicts": "SPARK2 (<< 2.0.0.cloudera2)",
"replaces": "IMPALA, SOLR, SPARK, KUDU",
"hash": "bef6f3f074e0a88cd79d6d37abc6698471e3d279"
}
]
}
}

```

3. Copy /tmp/clouderarepo/CDH5.13.oparcels to the admin node (rhel1)

```
#scp -r /tmp/clouderarepo/CDH5.13.0parcels/ rhel1:/var/www/html/
```

4. Verify that these files are accessible by visiting the URL <http://10.13.1.50/CDH5.13.oparcels/> in admin node.

Setting Up the MariaDB Database for Cloudera Manager

- Install the MariaDB Server
- Configure and Start the MariaDB Server
- Install the MariaDB/MySQL JDBC Driver
- Create Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server

Installing the MariaDB Server

To use a MariaDB database, complete the following steps:

- In the admin node where Cloudera Manager will be installed, use the following command to install the mariadb/mysql server.

```
#yum -y install mariadb-server
```

- To configure and start the MySQL Server, stop the MariaDB server if it is running.

```
#service mariadb stop
```

- Move the old InnoDB log if exists.

- Move files /var/lib/mysql/ib_logfile0 and /var/lib/mysql/ib_logfile1 out of /var/lib/mysql/ to a backup location.

```
#mv /var/lib/mysql/ib_logfile0 /root/ib_logfile0.bkp
#mv /var/lib/mysql/ib_logfile1 /root/ib_logfile1.bkp
```

- Determine the location of the option file, my.cnf and edit/add following lines:

```
#vi /etc/my.cnf

[mysqld]
transaction-isolation = READ-COMMITTED

# InnoDB settings

innodb_flush_method = O_DIRECT

max_connections = 550
```

```
[root@rhell ~]# vi /etc/my.cnf
[root@rhell ~]# cat /etc/my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
transaction-isolation = READ-COMMITTED
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

# InnoDB settings
innodb_flush_method = O_DIRECT

max_connections = 550
```



Note: The max_connections need to be increased based on number of nodes and applications. Please follow the recommendations as mentioned in the Cloudera document
http://www.cloudera.com/documentation/enterprise/latest/topics/install_cm_mariadb.html#install_cm_mariadb_config

- Make sure MySQL Server starts at boot:

```
#systemctl enable mariadb.service
```

7. Start the MySQL Server:

```
#service mariadb start
```

8. Set the MySQL root password on admin node (rhel1)

```
#cd /usr/bin/  
#mysql_secure_installation
```

```
Enter current password for root (enter for none):  
OK, successfully used password, moving on...  
  
Setting the root password ensures that nobody can log into the MariaDB  
root user without the proper authorisation.  
  
Set root password? [Y/n] y  
New password:  
Re-enter new password:  
Password updated successfully!  
Reloading privilege tables..  
... Success!  
  
By default, a MariaDB installation has an anonymous user, allowing anyone  
to log into MariaDB without having to have a user account created for  
them. This is intended only for testing, and to make the installation  
go a bit smoother. You should remove them before moving into a  
production environment.  
  
Remove anonymous users? [Y/n] y  
... Success!  
  
Normally, root should only be allowed to connect from 'localhost'. This  
ensures that someone cannot guess at the root password from the network.  
  
Disallow root login remotely? [Y/n] n  
... skipping.  
  
By default, MariaDB comes with a database named 'test' that anyone can  
access. This is also intended only for testing, and should be removed  
before moving into a production environment.  
RHEL7 - ... / 1... 🔍 X \ Index of /cdh... | IDesign: Excel... > +  
Remove test database and access to it? [Y/n] y  
- Dropping test database...  
... Success!  
- Removing privileges on test database...  
... Success! is what we will use to communicate  
is for Q&A. We hope you enjoy the presentation.  
Reloading the privilege tables will ensure that all changes made so far  
will take effect immediately.  
  
Reload privilege tables now? [Y/n] y  
... Success!  
  
Cleaning up...  
Options  
All done! If you've completed all of the above steps, your MariaDB  
installation should now be secure.
```

Installing the MySQL JDBC Driver

Install the JDBC driver on the Cloudera Manager Server host, as well as hosts which run the Activity Monitor, Reports Manager, Hive Metastore Server, Sentry Server, Cloudera Navigator Audit Server, and Cloudera Navigator Metadata Server roles.

1. From a host connected to the Internet, download the MySQL JDBC Driver and transfer it to the admin node. Download the MySQL JDBC driver from the URL <http://www.mysql.com/downloads/connector/j/5.1.html>
2. Copy mysql-connector-java-5.1.37.tar.gz to admin node(rhel1)

```
#scp mysql-connector-java-5.1.37.tar.gz rhel1:/root/
```

3. Log in to the admin node and extract the file:

```
#tar xzvf mysql-connector-java-5.1.37.tar.gz
```

4. Create the /usr/share/java/ directory on the admin node (rhel1)

```
#mkdir -p /usr/share/java/
```

5. Go to the mysql-connector-java-5.1.37 directory on the admin node (rhel1) and copy mysql-connector-java-5.1.37-bin.jar to /usr/share/java/

```
#cd mysql-connector-java-5.1.37
#cp mysql-connector-java-5.1.37-bin.jar /usr/share/java/mysql-connector-
java.jar
```

Creating Databases for Activity Monitor, Reports Manager, Hive Metastore Server, Navigator Audit Server, and Navigator Metadata Server

1. In the admin node Log into MySQL as the root user:

```
#mysql -u root -p
```

2. Enter the password that was previously supplied.

Enter password:

3. Create databases for the Activity Monitor, Reports Manager and Hive Metastore Server using the command below:

```
mysql> create database amon DEFAULT CHARACTER SET utf8;
mysql> create database rman DEFAULT CHARACTER SET utf8;
mysql> create database metastore DEFAULT CHARACTER SET utf8;
mysql> create database nav DEFAULT CHARACTER SET utf8;
mysql> create database navms DEFAULT CHARACTER SET utf8;
mysql> create database sentry DEFAULT CHARACTER SET utf8;
mysql> create database oozie DEFAULT CHARACTER SET utf8;

mysql> grant all on rman.* TO 'root'@'%' IDENTIFIED BY 'password';
mysql> grant all on metastore.* TO 'root'@'%' IDENTIFIED BY 'password';
mysql> grant all on amon.* TO 'root'@'%' IDENTIFIED BY 'password';
mysql> grant all on nav.* TO 'root'@'%' IDENTIFIED BY 'password';
mysql> grant all on navms.* TO 'root'@'%' IDENTIFIED BY 'password';
mysql> grant all on sentry.* TO 'root'@'%' IDENTIFIED BY 'password';
mysql> grant all privileges on oozie.* to root@'%' IDENTIFIED BY 'pass-
word';
mysql> grant all on rman.* TO 'rman'@'%' IDENTIFIED BY 'password';
mysql> grant all on metastore.* TO 'hive'@'%' IDENTIFIED BY 'password';
```

```

mysql> grant all on amon.*TO 'amon'@'%' IDENTIFIED BY 'password';
mysql> grant all on nav.*TO 'nav'@'%' IDENTIFIED BY 'password';
mysql> grant all on navms.*TO 'navms'@'%' IDENTIFIED BY 'password';
mysql> grant all on sentry.*TO 'root'@'%' IDENTIFIED BY 'password';
mysql> grant all privileges on oozie.* to oozie@'%' IDENTIFIED BY 'password';
[root@rhel1 mysql-connector-java-5.1.37]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 5.5.41-MariaDB MariaDB Server

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database amon DEFAULT CHARACTER SET utf8;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> create database rman DEFAULT CHARACTER SET utf8;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> create database metastore DEFAULT CHARACTER SET utf8;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> create database nav DEFAULT CHARACTER SET utf8;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> create database navms DEFAULT CHARACTER SET utf8;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> grant all on rman.*TO 'root'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec) Index of /cdh...  Design: Excel...

MariaDB [(none)]> grant all on metastore.*TO 'root'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on amon.*TO 'root'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.01 sec) station.

MariaDB [(none)]> grant all on nav.*TO 'root'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on navms.*TO 'root'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> quit
Bye

```

Cloudera Manager Installation

The following section describes installation of Cloudera Manager first and then using Cloudera Manager to install CDH 5.13.0.

Setting Up the Cloudera Manager Server Database

The Cloudera Manager Server Database stores information about service and host configurations.

Installing Cloudera Manager

Cloudera Manager, an end to end management application, is used to install and configure CDH. During CDH Installation, Cloudera Manager's Wizard will help to install Hadoop services on all nodes using the following procedure:

- Discovery of the cluster nodes
- Configure the Cloudera parcel or package repositories
- Install Hadoop, Cloudera Manager Agent (CMA) and Impala on all the cluster nodes.
- Install the Oracle JDK if it is not already installed across all the cluster nodes.
- Assign various services to nodes.
- Start the Hadoop services.

To install Cloudera Manager, complete the following steps:

1. Update the repo files to point to local repository.

```
#rm -f /var/www/html/clouderarepo/*.repo
#cp /etc/yum.repos.d/c*.repo /var/www/html/clouderarepo/
```

2. Install the Oracle Java Development Kit on the Cloudera Manager Server host.

3. #yum install oracle-j2sdk1.7

4. Install the Cloudera Manager Server packages either on the host where the database is installed, or on a host that has access to the database.

```
#yum install cloudera-manager-daemons cloudera-manager-server
```

Preparing a Cloudera Manager Server External Database

1. Run the scm_prepare_database.sh script on the host where the Cloudera Manager Server package is installed (rhel1) admin node.

```
#cd /usr/share/cmfschema
./scm_prepare_database.sh mysql amon root <password>
./scm_prepare_database.sh mysql rman root <password>
./scm_prepare_database.sh mysql metastore root <password>
./scm_prepare_database.sh mysql nav root <password>
./scm_prepare_database.sh mysql navms root <password>
./scm_prepare_database.sh mysql sentry root <password>
./scm_prepare_database.sh mysql oozie root <password>
```

2. Verify the database connectivity using the following command.

```
[root@rhel1 ~]# mysql -u root -p
mysql> connect amon
mysql> connect rman
mysql> connect metastore
mysql> connect nav
mysql> connect navms
mysql> connect sentry
```

```
mysql> connect oozie
[root@rhel1 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.1.71 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> connect amon
Connection id:    15
Current database: amon

mysql> connect rman
Connection id:    16
Current database: rman

mysql> connect metastore
Connection id:    17
Current database: metastore
```

The MySQL External database setup is complete.

Starting The Cloudera Manager Server

1. Start the Cloudera Manager Server:

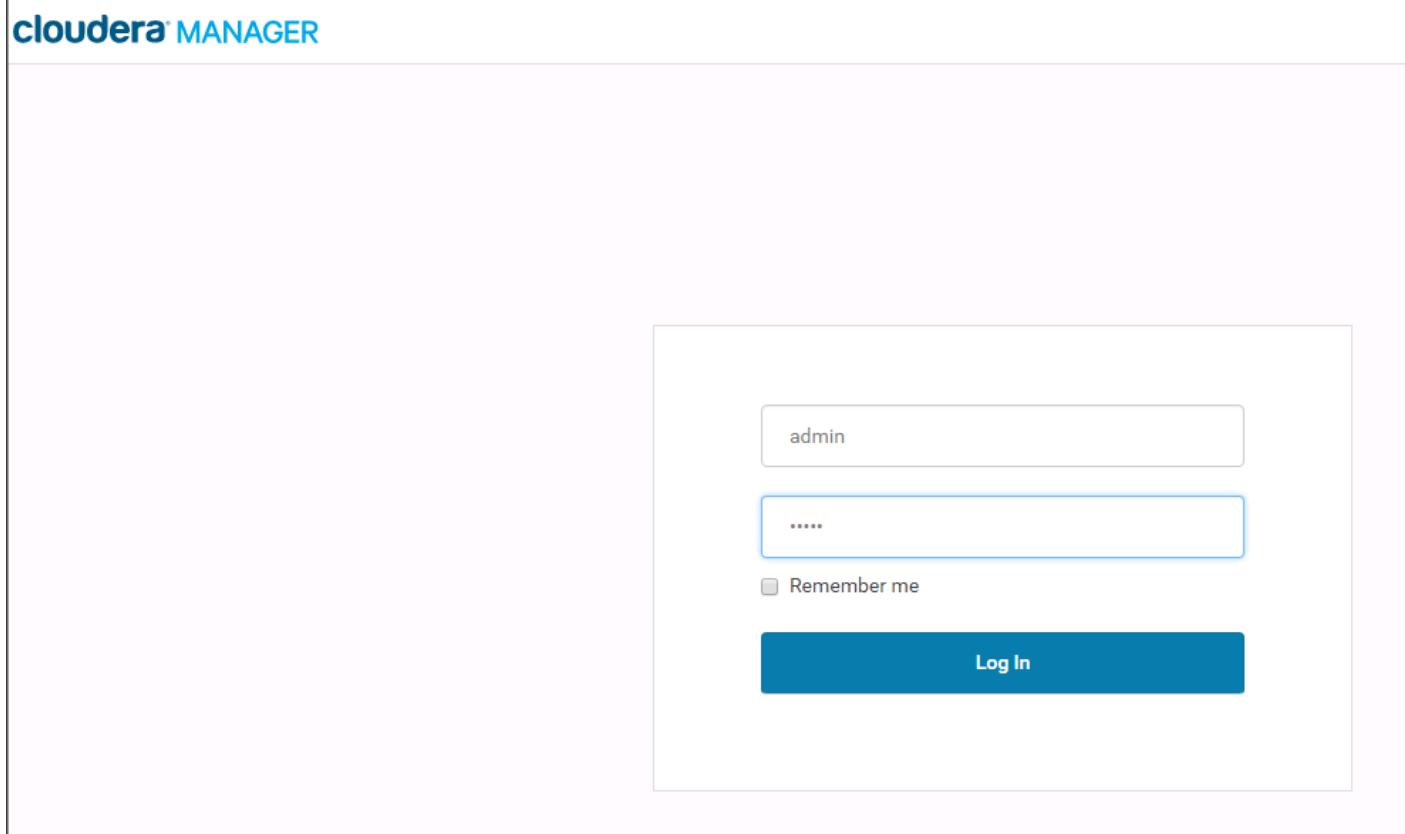
```
#service cloudera-scm-server start
```

2. Access the Cloudera Manager using the URL, <http://10.13.1.50:7180> to verify that the server is up.
3. Once the installation of Cloudera Manager is complete, install CDH5 using the Cloudera Manager Web interface.

Installing Cloudera Enterprise Data Hub (CDH5)

To install the Cloudera Enterprise Data Hub, complete the following steps:

1. Login to the Cloudera Manager. Enter "admin" for both the Username and Password fields.



Welcome to Cloudera Manager

End User License Terms and Conditions

Cloudera Standard License

Version 2016-05-26

END USER LICENSE TERMS AND CONDITIONS

THESE TERMS AND CONDITIONS (THESE "TERMS") APPLY TO YOUR USE OF THE PRODUCTS (AS DEFINED BELOW) PROVIDED BY CLOUDERA, INC. ("CLOUDERA").

PLEASE READ THESE TERMS CAREFULLY.

IF YOU ("YOU" OR "CUSTOMER") PLAN TO USE ANY OF THE PRODUCTS ON BEHALF OF A COMPANY OR OTHER ENTITY, YOU REPRESENT THAT YOU ARE THE EMPLOYEE OR AGENT OF SUCH COMPANY (OR OTHER ENTITY) AND YOU HAVE THE AUTHORITY TO ACCEPT ALL OF THE TERMS AND CONDITIONS SET FORTH IN AN ACCEPTED REQUEST (AS DEFINED BELOW) AND THESE TERMS (COLLECTIVELY, THE "AGREEMENT") ON BEHALF OF SUCH COMPANY (OR OTHER ENTITY).

BY USING ANY OF THE PRODUCTS, YOU ACKNOWLEDGE AND AGREE THAT:

- (A) YOU HAVE READ ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT;
- (B) YOU UNDERSTAND ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT;
- (C) YOU AGREE TO BE LEGALLY BOUND BY ALL OF THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT

IF YOU DO NOT AGREE WITH ANY OF THE TERMS OR CONDITIONS OF THESE TERMS, YOU MAY NOT USE ANY PORTION OF THE PRODUCTS.

Yes, I accept the End User License Terms and Conditions.

If your download and use of Cloudera Manager are on behalf of a company that has an existing agreement with Cloudera for the use of the software, your action does not modify that existing agreement.

1 2

[Back](#) [Continue](#)

2. If you do not have a Cloudera license, select Cloudera Enterprise Data Hub Trial Edition. If you do have a Cloudera license, Click “Upload License” and select your license.
3. Based on requirement, choose appropriate Cloudera Editions for the Installation.

Figure 24 Installing Cloudera Enterprise

Welcome to Cloudera Manager

Which edition do you want to deploy?

Upgrading to **Cloudera Enterprise** provides important features that help you manage and monitor your Hadoop clusters in mission-critical environments.

	Cloudera Express	Cloudera Enterprise Cloudera Enterprise Trial	Cloudera Enterprise
License	Free	60 Days After the trial period, the product will continue to function as Cloudera Express . Your cluster and your data will remain unaffected.	Annual Subscription Upload License Key Select License File Upload Cloudera Enterprise is available in three editions: <ul style="list-style-type: none">• Basic Edition• Flex Edition• Cloudera Enterprise
Node Limit	Unlimited	Unlimited	Unlimited
CDH	✓	✓	✓
Core Cloudera Manager Features	✓	✓	✓
Advanced Cloudera Manager Features		✓	✓
Cloudera Navigator		✓	✓

Back 1 2 Continue

- Click Continue on the confirmation page.

Thank you for choosing Cloudera Manager and CDH.

This installer will install **Cloudera Enterprise Trial 5.13.0** and enable you to later choose packages for the services below (there may be some license implications).

- Apache Hadoop (Common, HDFS, MapReduce, YARN)
- Apache HBase
- Apache ZooKeeper
- Apache Oozie
- Apache Hive
- Hue (Apache licensed)
- Apache Flume
- Cloudera Impala (Apache licensed)
- Apache Sentry
- Apache Sqoop
- Cloudera Search (Apache licensed)
- Apache Spark

You are using Cloudera Manager to install and configure your system. You can learn more about Cloudera Manager by clicking on the **Support** menu above.

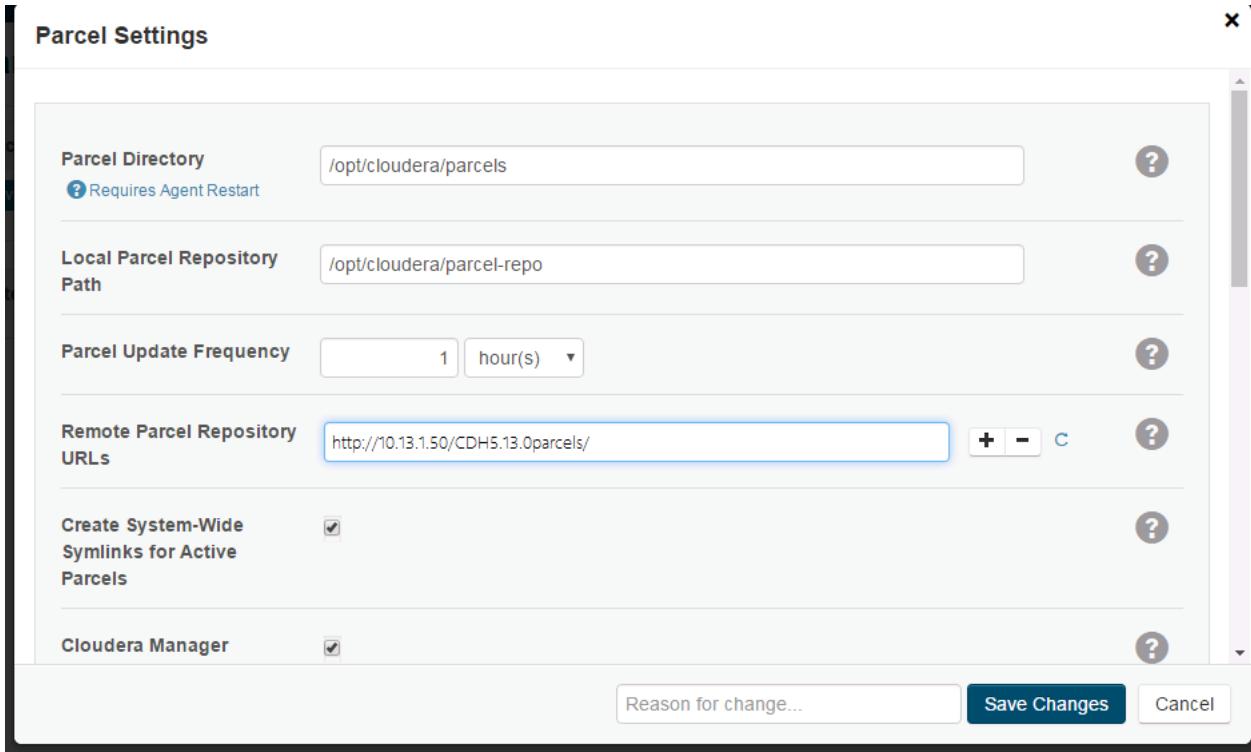
Before you proceed, be sure to checkout the [CDH and Cloudera Manager Requirements and Supported Versions](#) ↗

- [Supported Operating Systems](#) ↗
- [Supported Databases](#) ↗
- [Supported JDK Versions](#) ↗

Continue

Edit the Cloudera Enterprise Parcel Settings to Use the CDH 5.13.0 Parcels

1. Open another tab in the same browser window and visit the URL: <http://10.13.1.50:7180/cmf/parcel/status> for modifying the parcel settings.
2. Click Configuration on this page.
3. Click to remove the entire remote repository URLs, and add the URL to the location where we kept the CDH 5.13.0 parcels i.e. <http://10.13.1.50/CDH5.13.oparcels/>.



4. Click Save Changes to finish the configuration.
5. Navigate back to the Cloudera installation home page i.e. <http://10.13.1.50:7180>.
6. Click Continue on the confirmation page.

Thank you for choosing Cloudera Manager and CDH.

This installer will install **Cloudera Enterprise Trial 5.13.0** and enable you to later choose packages for the services below (there may be some license implications).

- Apache Hadoop (Common, HDFS, MapReduce, YARN)
- Apache HBase
- Apache ZooKeeper
- Apache Oozie
- Apache Hive
- Hue (Apache licensed)
- Apache Flume
- Cloudera Impala (Apache licensed)
- Apache Sentry
- Apache Sqoop
- Cloudera Search (Apache licensed)
- Apache Spark

You are using Cloudera Manager to install and configure your system. You can learn more about Cloudera Manager by clicking on the **Support** menu above.

Before you proceed, be sure to checkout the [CDH and Cloudera Manager Requirements and Supported Versions](#)

- [Supported Operating Systems](#)
- [Supported Databases](#)
- [Supported JDK Versions](#)

Continue

- Specify the hosts that are part of the cluster using their IP addresses or hostname. The figure below shows use of a pattern to specify the IP addresses range.

10.13.1.[50-73] or rhel[1-24]

- After the IP addresses or hostnames are entered, click Search.

Figure 25 Searching for Cluster Nodes

Specify hosts for your CDH cluster installation.

Hosts should be specified using the same hostname (FQDN) that they will identify themselves with. Cloudera recommends including Cloudera Manager Server's host. This also enables health monitoring for that host.

Hint: Search for hostnames and IP addresses using [patterns](#).

rhel[1-24]

SSH Port: 22 **Search**

- Cloudera Manager will "discover" the nodes in the cluster. Verify that all desired nodes have been found and selected for installation.

Specify hosts for your CDH cluster installation.

Hosts should be specified using the same hostname (FQDN) that they will identify themselves with.

Cloudera recommends including Cloudera Manager Server's host. This also enables health monitoring for that host.

Hint: Search for hostnames and IP addresses using [patterns](#).

7 hosts scanned, 7 running SSH.

[New Search](#)

<input checked="" type="checkbox"/> Expanded Query	Hostname (FQDN)	IP Address	Currently Managed	Result
<input checked="" type="checkbox"/> rhel1	rhel1	10.13.1.50	No	✓ Host ready: 0 ms response time.
<input checked="" type="checkbox"/> rhel2	rhel2	10.13.1.51	No	✓ Host ready: 0 ms response time.
<input checked="" type="checkbox"/> rhel3	rhel3	10.13.1.52	No	✓ Host ready: 0 ms response time.
<input checked="" type="checkbox"/> rhel4	rhel4	10.13.1.53	No	✓ Host ready: 1 ms response time.
<input checked="" type="checkbox"/> rhel5	rhel5	10.13.1.54	No	✓ Host ready: 0 ms response time.
<input checked="" type="checkbox"/> rhel6	rhel6	10.13.1.55	No	✓ Host ready: 1 ms response time.
<input checked="" type="checkbox"/> rhel7	rhel7	10.13.1.56	No	✓ Host ready: 0 ms response time.

10. Click Continue.
11. For the method of installation, select the Use Parcels (Recommended) radio button.
12. For the CDH version, select the CDH5.13.0-1.cdh5.13.0.p0.29 radio button.
13. For the specific release of Cloudera Manager, select the Custom Repository radio button.
14. Enter the URL for the repository within the admin node. <http://10.13.1.50/clouderarepo/cloudera-manager> and click Continue.

Cluster Installation

Select Repository

Cloudera recommends the use of parcels for installation over packages, because parcels enable Cloudera Manager to easily manage the software on your cluster, automating the deployment and upgrade of service binaries. Electing not to use parcels will require you to manually upgrade packages on all hosts in your cluster when software updates are available, and will prevent you from using Cloudera Manager's rolling upgrade capabilities.

Choose Method

- Use Packages (?)
- Use Parcels (Recommended) (?)

[More Options](#) [Proxy Settings](#)

Select the version of CDH

- CDH-5.13.0-1.cdh5.13.0.p0.29

Versions of CDH that are too new for this version of Cloudera Manager (5.13.0) will not be shown.

Select the specific release of the Cloudera Manager Agent you want to install on your hosts.

- Matched release for this Cloudera Manager Server
- Custom Repository

`http://10.13.1.50/clouderarepo/cloudera-manager/`

Example for SLES, Redhat or other RPM based distributions:

`https://archive.cloudera.com/cm5/redhat/6/x86_64/cm/5/`

Example for Ubuntu or other Debian based distributions:

`deb https://archive.cloudera.com/cm5/ubuntu/lucid/amd64/cm/ lucid-cm5 contrib`

Enter a custom URL for the location of the GPG signing key (applies to all custom repositories and without Internet access).

`Custom GPG Key URL`

1 2 3 4 5 6 7

[Back](#)

[Continue](#)

Cluster Installation

JDK Installation Options

Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX

ORACLE AMERICA, INC. ("ORACLE"), FOR AND ON BEHALF OF ITSELF AND ITS SUBSIDIARIES AND AFFILIATES UNDER COMMON CONTROL, IS WILLING TO LICENSE THE SOFTWARE TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY CODE LICENSE AGREEMENT AND SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT"). PLEASE READ THE AGREEMENT CAREFULLY. BY SELECTING THE "ACCEPT LICENSE AGREEMENT" (OR THE EQUIVALENT) BUTTON AND/OR BY USING THE SOFTWARE YOU ACKNOWLEDGE THAT YOU HAVE READ THE TERMS AND AGREE TO THEM. IF YOU ARE AGREEING TO THESE TERMS ON BEHALF OF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE LEGAL AUTHORITY TO BIND THE LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO BE BOUND BY THE TERMS, THEN SELECT THE "DECLINE LICENSE AGREEMENT" (OR THE EQUIVALENT) BUTTON AND YOU MUST NOT USE THE SOFTWARE ON THIS SITE OR ANY OTHER MEDIA ON WHICH THE SOFTWARE IS CONTAINED.

1. DEFINITIONS. "Software" means the software identified above in binary form that you selected for download, install or use (in the version You selected for download, install or use) from Oracle or its authorized licensees, any other machine readable materials (including, but not limited to, libraries, source files, header files, and data files), any updates or error corrections provided by Oracle, and any user manuals, programming guides and other documentation provided to you by Oracle under this Agreement. "General Purpose Desktop Computers and Servers" means computers, including desktop and laptop computers, or servers, used for general computing functions under end user control (such as but not specifically limited to email, general purpose Internet browsing, and office suite productivity tools). The use of Software in systems and solutions that provide dedicated functionality (other than as mentioned above) or designed for use in embedded or function-specific software applications, for example but not limited to: Software embedded in or bundled with industrial control systems, wireless mobile telephones, wireless handheld devices, kiosks, TV/STB, Blu-ray Disc devices, telematics and network control switching equipment, printers and storage management systems, and other related systems are excluded from this definition and not licensed under this Agreement. "Programs" means (a) Java technology applets and applications intended to run on the Java Platform, Standard Edition platform on Java-enabled General Purpose Desktop Computers and Servers; and (b) JavaFX technology applications intended to run on the JavaFX Runtime on JavaFX-enabled General Purpose Desktop Computers and Servers. "Commercial Features" means those features identified in

Install Oracle Java SE Development Kit (JDK)

Check this box to accept the Oracle Binary Code License Agreement and install the JDK. Leave it unchecked to use a currently installed JDK.

Install Java Unlimited Strength Encryption Policy Files

Check this checkbox if local laws permit you to deploy unlimited strength encryption and you are running a secure cluster.

Cluster Installation

Enable Single User Mode

Only supported for CDH 5.2 and above.

By default, service processes run as distinct users on the system. For example, HDFS DataNodes run as user "hdfs" and HBase RegionServers run as user "hbase." Enabling "single user mode" configures Cloudera Manager to run service processes as a single user, by default "cloudera-scm", thereby prioritizing isolation between managed services and the rest of the system over isolation between the managed services.

The **major benefit** of this option is that the Agent does not run as root. However, this mode complicates installation, which is described fully in the [documentation](#). Most notably, directories which in the regular mode are created automatically by the Agent, must be created manually on every host with appropriate permissions, and sudo (or equivalent) access must be set up for the configured user.

Switching back and forth between single user mode and regular mode is not supported.

Single User Mode 



15. Provide SSH login credentials for the cluster and click Continue.

Figure 26 Login Credentials to Start CDH Installation

Cluster Installation

Provide SSH login credentials.

Root access to your hosts is required to install the Cloudera packages. This installer will connect to your hosts via SSH and log in either directly as root or as another user with password-less sudo/pbrun privileges to become root.

Login To All Hosts As: root

Another user

You may connect via password or public-key authentication for the user selected above.

Authentication Method: All hosts accept same password

All hosts accept same private key

Enter Password:

Confirm Password:

SSH Port: 22

Number of Simultaneous Installations: 10 (Running a large number of installations at once can consume large amounts of network bandwidth and other system resources)

1 2 3 4 5 6 7

Back

Continue

16. Installation using parcels begins.

Cluster Installation

Installation completed successfully.

4 of 4 host(s) completed successfully.

Hostname	IP Address	Progress	Status	
rhel1	10.13.1.54	<div style="width: 100%; background-color: #2e7131;"></div>	✓ Installation completed successfully.	Details
rhel2	10.13.1.55	<div style="width: 100%; background-color: #2e7131;"></div>	✓ Installation completed successfully.	Details
rhel3	10.13.1.56	<div style="width: 100%; background-color: #2e7131;"></div>	✓ Installation completed successfully.	Details
rhel4	10.13.1.57	<div style="width: 100%; background-color: #2e7131;"></div>	✓ Installation completed successfully.	Details

Cluster Installation

Installing Selected Parcels

The selected parcels are being downloaded and installed on all the hosts in the cluster.



17. Once the installation is completed successfully, click Continue to select the required services.
18. Wait for Cloudera Manager to inspect the hosts on which it has just performed the installation.
19. Review and verify the summary. Click Continue.

Figure 27 Inspecting Hosts for Correctness

Cluster Installation

Inspect hosts for correctness [Run Again](#)

Validations

✓	Inspector ran on all 24 hosts.
✓	Individual hosts resolved their own hostnames correctly.
✓	No errors were found while looking for conflicting init scripts.
✓	No errors were found while checking /etc/hosts.
✓	All hosts resolved localhost to 127.0.0.1.
✓	All hosts checked resolved each other's hostnames correctly and in a timely manner.
✓	Host clocks are approximately in sync (within ten minutes).
✓	Host time zones are consistent across the cluster.
✓	No users or groups are missing.
✓	No conflicts detected between packages and parcels.
✓	No kernel versions that are known to be bad are running.
✓	No problems were found with /proc/sys/vm/swappiness on any of the hosts.
✓	No performance concerns with Transparent Huge Pages settings.
✓	CDH 5 Hue Python version dependency is satisfied.
✓	0 hosts are running CDH 4 and 4 hosts are running CDH 5.
✓	All checked hosts in each cluster are running the same version of components.
✓	All managed hosts have consistent versions of Java.

20. Select services that need to be started on the cluster.

Figure 28 Selecting CDH Version and Services**Cluster Setup**

Choose the CDH 5 services that you want to install on your cluster.

Choose a combination of services to install.

Core Hadoop

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, and Hue

Core with HBase

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and HBase

Core with Impala

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Impala

Core with Search

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Solr

Core with Spark

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, and Spark

All Services

HDFS, YARN (MapReduce 2 Included), ZooKeeper, Oozie, Hive, Hue, HBase, Impala, Solr, Spark, and Key-Value Store Indexer

Note: Please ensure that you have the appropriate license key for **Cloudera Impala**, **Cloudera Search**, **HBase**, and **Spark** or contact Cloudera for assistance.

Custom Services

Choose your own services. Services required by chosen services will automatically be included. Flume can be added after your initial cluster has been set up.

This wizard will also install the **Cloudera Management Service**. These are a set of components that enable monitoring, reporting, events, and alerts; these components require databases to store information, which will be configured on the next page.

Include Cloudera Navigator

21. This is a critical steps in the installation. Inspect and customize the role assignments of all the nodes based on your requirements and click Continue.
22. Reconfigure the service assignment to match Table 7below.

Table 7 Service Assignments

Service Name	Host
NameNode	rhel1, rhel2 (HA)
HistoryServer	rhel1
JournalNodes	rhel1,rhel2,rhel3
ResourceManager	rhel2, rhel3 (HA)
Hue Server	rhel2
HiveMetastore Server	rhel1
HiveServer2	rhel2
HBase Master	rhel2
Oozie Server	rhel1

Service Name	Host
ZooKeeper	rhel1, rhel2, rhel3
DataNode	rhel4 to rhel24
NodeManager	rhel4 to rhel24
RegionServer	rhel4 to rhel24
Sqoop Server	rhel1
Impala Catalog Server Daemon	rhel1
Impala State Store	rhel2
Impala Daemon	rhel4 to rhel24
Solr Server	rhel4 (can be installed on all hosts if needed, if there is a search use case)
Spark History Server	rhel1
Spark Executors	rhel4 to rhel24

Cluster Setup

Customize Role Assignments

You can customize the role assignments for your new cluster here, but if assignments are made incorrectly, such as assigning too many roles to a single host, this can impact the performance of your services. Cloudera does not recommend altering assignments unless you have specific requirements, such as having pre-selected a specific host for a specific role.

You can also view the role assignments by host: [View By Host](#)

HBase

M Master x 1 New	HRES HBase REST Server	HTS HBase Thrift Server	RS RegionServer x 3 New
rhel2 ▾	Select hosts	Select hosts	Same As DataNode ▾

HDFS

NN NameNode x 1 New	SNN SecondaryNameNode x 1 New	B Balancer x 1 New	HFS HttpFS
rhel1	rhel2 ▾	rhel1	Select hosts
NFSG NFS Gateway	DN DataNode x 23 New		
Select hosts	rhel[2-24] ▾		

Hive

G Gateway x 4 New	HMS Hive Metastore Server x 1 New	WHCS WebHCat Server x 1 New	HS2 HiveServer2 x 1 New
rhel[1-24]	rhel1	rhel1 ▾	rhel2 ▾

The screenshot displays the Cloudera Manager interface with various service configurations:

- Hue**: HS Hue Server (1 New) assigned to rhel1, LB Load Balancer (1 New) assigned to rhel2.
- Impala**: ISS Impala StateStore (1 New) assigned to rhel1, ICS Impala Catalog Server (1 New) assigned to rhel2, ID Impala Daemon (3 New) assigned to Same As DataNode.
- Key-Value Store Indexer**: LHB Lily HBase Indexer (1 New) assigned to rhel2.
- Cloudera Management Service**: SM Service Monitor (1 New) assigned to rhel1, AM Activity Monitor (1 New) assigned to rhel1, HM Host Monitor (1 New) assigned to rhel1, RM Reports Manager (1 New) assigned to rhel1.
- Oozie**: OS Oozie Server (1 New) assigned to rhel1.
- Solr**: SS Solr Server (1 New) assigned to rhel1.
- Spark**: HS History Server (1 New) assigned to rhel1, G Gateway (24 New) assigned to rhel [1-24].
- YARN (MR2 Included)**: RM ResourceManager (1 New) assigned to rhel2, JHS JobHistory Server (1 New) assigned to rhel1, NM NodeManager (3 New) assigned to Same As DataNode.
- ZooKeeper**: S Server (3 New) assigned to rhel[1-3].

Setting up the Database

The role assignment recommendation above is for clusters of up to 64 servers. For clusters larger than 64 nodes, use the HA recommendation defined in Table 7 above.

1. In the Database Host Name sections use port 3306 for TCP/IP because connection to the remote server always uses TCP/IP.

2. Enter the Database Name, username and password that were used during the database creation stage earlier in this document.
3. Click Test Connection to verify the connection and click Continue.

Figure 29 Database Setup

Cluster Setup

Database Setup

Configure and test database connections. Create the databases first according to the [Installing and Configuring an External Database](#) section of the [Installation Guide](#).

Hive ✓ Successful

Database Host Name: *	MySQL	Database Name : *	Username: *	Password:
rhel1	MySQL	metastore	root	*****

Activity Monitor ✓ Successful

Currently assigned to run on **rhel1**.

Database Host Name: *	MySQL	Database Name : *	Username: *	Password:
rhel1	MySQL	amon	root	*****

Reports Manager ✓ Successful

Currently assigned to run on **rhel1**.

Database Host Name: *	MySQL	Database Name : *	Username: *	Password:
rhel1	MySQL	rman	root	*****

Oozie Server ✓ Successful

Currently assigned to run on **rhel1**.

Database Host Name: *	MySQL	Database Name : *	Username: *	Password:
rhel1	MySQL	oozie	root	*****

Hue ✓ Successful

1 2 3 4 5 6

Back **Continue**

4. Review and customize the configuration changes based on your requirements.

Figure 30 Review the Configuration Changes Part1

Cluster Setup

Review Changes

HDFS Root Directory hbase.rootdir	Cluster 1 > HBase (Service-Wide) <input type="text" value="/hbase"/>	?
Enable Indexing	<input checked="" type="checkbox"/> Cluster 1 > HBase (Service-Wide) ↳	?
Enable Replication	<input checked="" type="checkbox"/> Cluster 1 > HBase (Service-Wide) ↳	?
HDFS Block Size dfs.block.size, dfs.blocksize	Cluster 1 > HDFS (Service-Wide) <input type="text" value="128"/> MiB ▼	?
DataNode Failed Volumes Tolerated dfs.datanode.failed.volumes.tolerated	Cluster 1 > DataNode Default Group ↳ <input type="text" value="12"/>	?
DataNode Data Directory dfs.datanode.data.dir, dfs.datanode.data.dir	Cluster 1 > DataNode Default Group ↳ <input type="text" value="/data/disk1/dfs/dn"/> <input type="text" value="/data/disk10/dfs/dn"/> <input type="text" value="/data/disk11/dfs/dn"/> <input type="text" value="/data/disk12/dfs/dn"/> <input type="text" value="/data/disk13/dfs/dn"/>	⊖ ⊕ ⊖ ⊕ ⊖ ⊕ ⊖ ⊕ ⊖ ⊕
Back 1 2 3 4 5 6		Continue

5. Click Continue to start running the cluster services.

Cluster Setup

First Run Command

Status **Running** Mar 14, 2:25:06 PM

Completed 1 of 10 step(s).

Show All Steps Show Only Failed Steps Show Only Running Steps

> <input checked="" type="radio"/> Ensuring that the expected software releases are installed on hosts. Successfully completed 1 steps.	Mar 14, 2:25:06 PM	37ms
> <input type="radio"/> Deploying Client Configuration	Cluster 1	Mar 14, 2:25:07 PM
> <input type="radio"/> Start Cloudera Management Service, ZooKeeper		
> <input type="radio"/> Start HDFS		
> <input type="radio"/> Start HBase, Solr		
> <input type="radio"/> Start Key-Value Store Indexer, YARN (MR2 Included)		
> <input type="radio"/> Start Spark		
> <input type="radio"/> Start Hive		
> <input type="radio"/> Start Impala, Oozie		
> <input type="radio"/> Start Hue		

Cluster Setup

First Run Command

Status **Finished** Mar 14, 2:25:06 PM 4.6m

Finished First Run of the following services successfully: ZooKeeper, HDFS, HBase, Solr, YARN (MR2 Included), Key-Value Store Indexer, Spark, Hive, Impala, Oozie, Hue, Cloudera Management Service.

Completed 10 of 10 step(s).

Show All Steps Show Only Failed Steps Show Only Running Steps

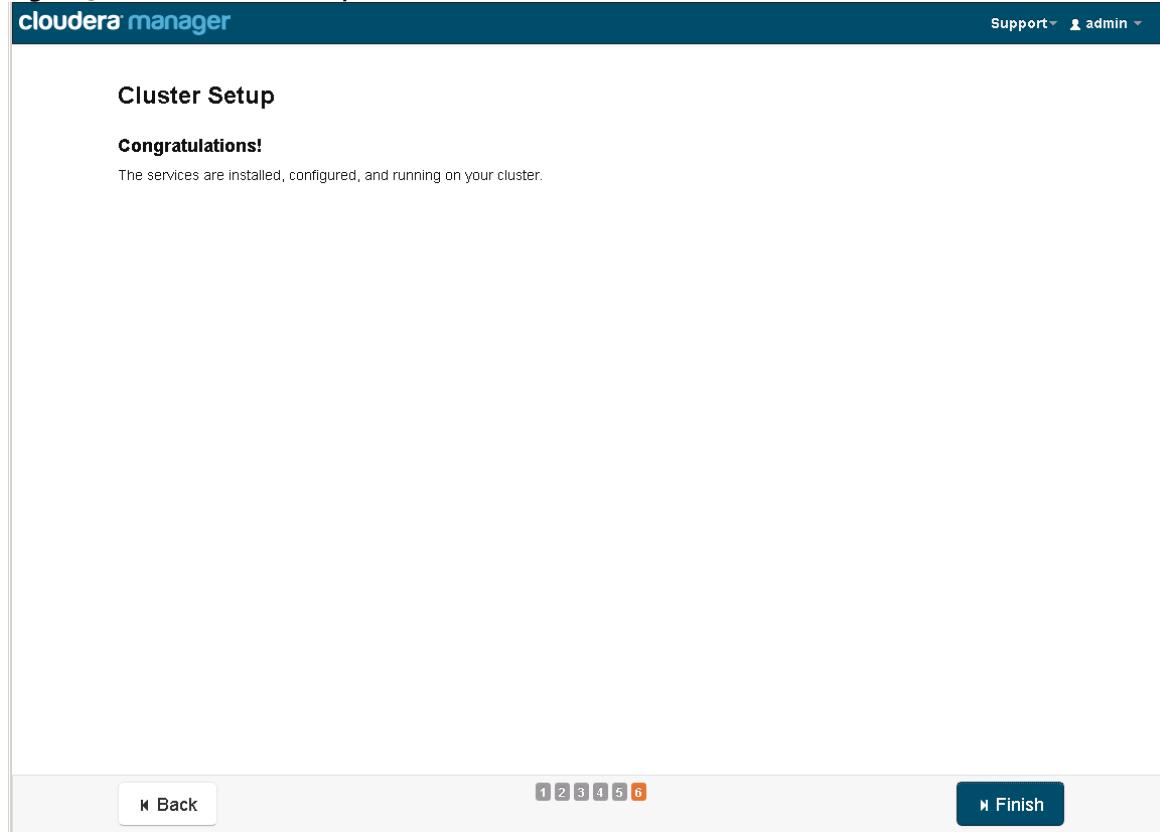
> <input checked="" type="radio"/> Ensuring that the expected software releases are installed on hosts. Successfully completed 1 steps.	Mar 14, 2:25:06 PM	37ms
> <input checked="" type="radio"/> Deploying Client Configuration Successfully deployed all client configurations.	Mar 14, 2:25:07 PM	16.23s
> <input checked="" type="radio"/> Start Cloudera Management Service, ZooKeeper Successfully completed 2 steps.	Mar 14, 2:25:23 PM	24.68s
> <input checked="" type="radio"/> Start HDFS Successfully completed 1 steps.	Mar 14, 2:25:48 PM	40.21s
> <input checked="" type="radio"/> Start HBase, Solr Successfully completed 2 steps.	Mar 14, 2:26:28 PM	32.3s
> <input checked="" type="radio"/> Start Key-Value Store Indexer, YARN (MR2 Included) Successfully completed 2 steps.	Mar 14, 2:27:00 PM	26.88s
> <input checked="" type="radio"/> Start Spark Successfully completed 1 steps.	Mar 14, 2:27:27 PM	31.27s

1 2 3 4 5 6

Starting the Cluster Services

1. Hadoop services are installed, configured and now running on all the nodes of the cluster. Click Finish to complete the installation.

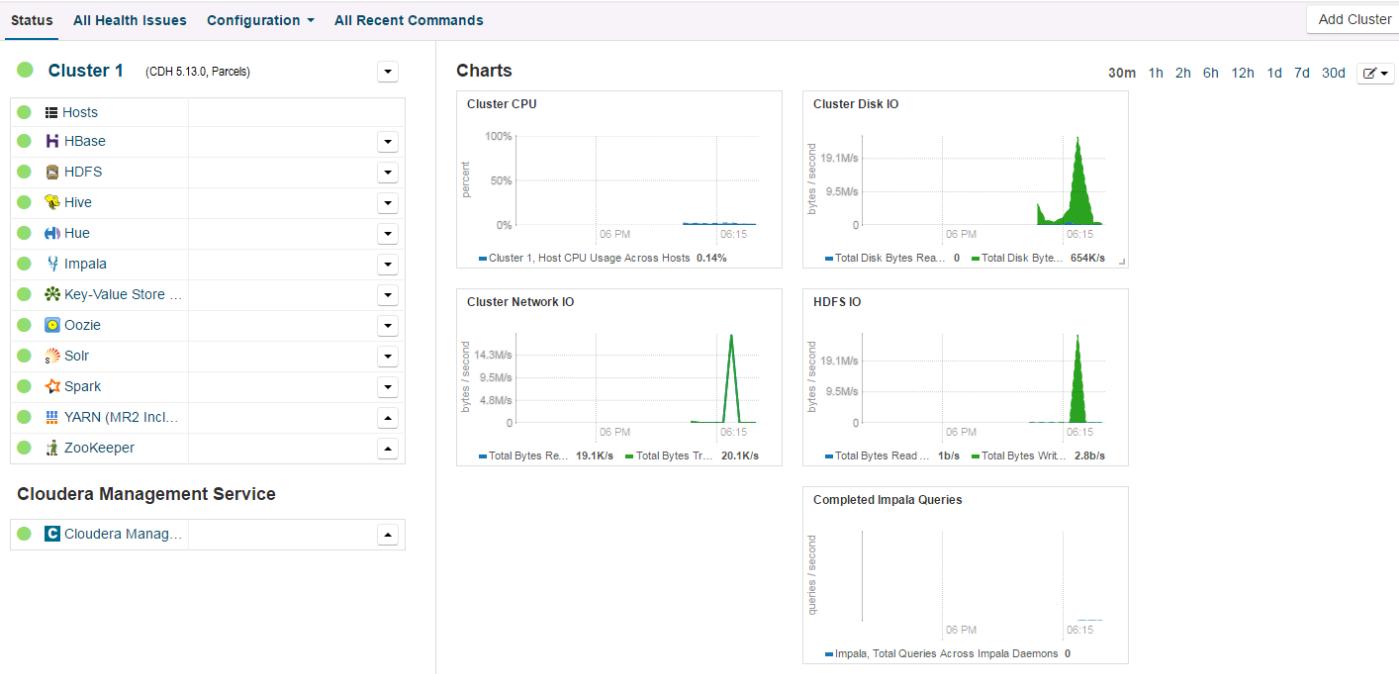
Figure 31 Installation Completion



Cloudera Manager now displays the status of all Hadoop services running on the cluster.

Figure 32 Service Status of the Cluster**Home**

30 minutes preceding April 29, 2016, 6:20 PM EDT



Scaling the Cluster

The role assignment recommendation above is for cluster with at least 64 servers and in High Availability (HA). For smaller cluster running without HA the recommendation is to dedicate one server for NameNode and a second server for secondary name node and YARN Resource Manager. For larger clusters larger than 28 nodes the recommendation is to dedicate one server each for name node, YARN Resource Manager and one more for running both NameNode (HA) and Resource Manager (HA) as in the table (no Secondary NameNode when in HA).

For production clusters it is recommended to set up NameNode and Resource manager in HA mode.

This implies that there will be at least 3 master nodes, running the NameNode, YARN Resource manager, the failover counter-part being designated to run on another node and a third node that would have similar capacity as the other two nodes.

All the three nodes will also need to run zookeeper and quorum journal node services. It is also recommended to have a minimum of 5 DataNodes in a cluster. Please refer to the next section for details on how to enable HA.

Enabling High Availability



Note: Setting up HA is done after the Cloudera Installation is completed.

HDFS High Availability

The HDFS HA feature provides the option of running two NameNodes in the same cluster, in an Active/Passive configuration. These are referred to as the Active NameNode and the Standby NameNode. Unlike the Secondary NameNode, the Standby NameNode is a hot standby, allowing a fast failover to a new NameNode in the case that a machine crashes, or a graceful administrator-initiated failover for the purpose of planned maintenance. There cannot be more than two NameNodes.

For more information go to:

http://www.cloudera.com/documentation/enterprise/latest/topics/cdh_hag_hdfs_ha_intro.html#topic_2_1

Setting Up HDFS HA

The Enable High Availability workflow leads through adding a second (standby) NameNode and configuring JournalNodes. During the workflow, Cloudera Manager creates a federated namespace.

1. Log in to the admin node (rhel1) and create the Edit directory for the JournalNode

```
#clush -w rhel[1-3] mkdir -p /data/disk1/namenode-edits
#clush -w rhel[1-3] chmod 777 /data/disk1/namenode-edits
```

```
[root@rhel1 ~]# clush -w rhel[1-3] mkdir -p /data/disk1/namenode-edits
[root@rhel1 ~]# clush -w rhel[1-3] chmod 77 /data/disk1/namenode-edits
```

2. Log in to the Cloudera manager and go to the HDFS service.
3. In the top right corner Select Actions> Enable High Availability. A screen showing the hosts that are eligible to run a standby NameNode and the JournalNodes displays.

Enable High Availability for HDFS

Getting Started

This wizard leads you through adding a standby NameNode, restarting this HDFS service and any dependent services, and then re-deploying client configurations.

Nameservice Name

Enabling High Availability creates a new nameservice. Accept the default name **nameservice1** or provide another name in **Nameservice Name**.

4. Specify a name for the nameservice or accept the default name nameservice1 and click Continue.
5. In the NameNode Hosts field, click Select a host. The host selection dialog displays.
6. Check the checkbox next to the hosts (rhel2) where the standby NameNode is to be set up and click OK.

Back

1 2 3 4 5

Continue



Note: The standby NameNode cannot be on the same host as the active NameNode, and the host that is chosen should have the same hardware configuration (RAM, disk space, number of cores, and so on) as the active NameNode.

7. In the **JournalNode Hosts** field, click **Select hosts**. The host selection dialog displays.
8. Check the checkboxes next to an odd number of hosts (a minimum of three) to act as JournalNodes and click **OK**. Here we are using the same nodes as Zookeeper nodes.

Enable High Availability for HDFS

Assign Roles

NameNode Hosts

JournalNode Hosts

We recommend that JournalNodes be hosted on machines of similar hardware specifications as the NameNodes. The hosts of NameNodes and the ResourceManager are generally good options. You must have a minimum of three and an odd number of JournalNodes.

Back
1
2
3
4
5
Continue



Note: JournalNodes should be hosted on hosts with similar hardware specification as the NameNodes. It is recommended that each JournalNode is put on the same hosts as the active and standby NameNodes, and the third JournalNode on ResourceManager node.

9. Click Continue.
10. In the JournalNode Edits Directory property, enter a directory location created earlier in step 1 for the JournalNode edits directory into the fields for each JournalNode host.

Enable High Availability for HDFS

Review Changes

Set the following configuration values for your new role(s). Required values are marked with *.

Parameter	Group	Value	Description
Service HDFS			
NameNode Data Directories* dfs.namenode.name.dir	rhel1 rhel2	/data/disk1/dfs/nn Inherited from: NameNode Default Group	Determines where on the local file system the NameNode should store the name table (fsimage). For redundancy, enter a comma-delimited list of directories to replicate the name table in all of the directories. Typical values are /data/N/dfs/nn where N=1..3.
JournalNode Edits Directory* dfs.journalnode.edits.dir	rhel1 rhel2 rhel3	/data/disk1/namenode-edits /data/disk1/namenode-edits /data/disk1/namenode-edits	Directory on the local file system where NameNode edits are written. Reset to empty default value Reset to empty default value Reset to empty default value

Back **Continue**

1 2 3 4 5



Note: The directories specified should be empty, and must have the appropriate permissions.

Extra Options: Decide whether Cloudera Manager should clear existing data in ZooKeeper, Standby NameNode, and JournalNodes. If the directories are not empty (for example, re-enabling a previous HA configuration), Cloudera Manager will not automatically delete the contents—select to delete the contents by keeping the default checkbox selection. The recommended default is to clear the directories.



Note: If you choose to not configure any of the extra options, the data should be in sync across the edits directories of the JournalNodes and should have the same version data as the NameNodes.

11. Click Continue.

Cloudera Manager executes a set of commands that will stop the dependent services, delete, create, and configure roles and directories as appropriate, create a nameservice and failover controller, and restart the dependent services and deploy the new client configuration.

Enable High Availability for HDFS

✓ Enable High Availability Command

Status: **Finished** Context: [HDFS](#) Start Time: May 1, 9:41:17 PM Duration: 10.9m

Successfully enabled High Availability and Automatic Failover

Details Completed 20 of 20 step(s).

All Failed Only Running Only

Step	Context	Start Time	Duration	Actions
✓ Check that name directories for the new Standby NameNode either do not exist or are writable and empty. Can optionally clear directories. Process host-validate-writable-empty-dirs (id=91) on host rhel2 (id=4) exited with 0 and expected 0	rhel2	May 1, 9:41:17 PM	15.14s	
» ✓ Check that edits directories for the nameservice either do not exist or are writable and empty. Can optionally clear directories.		May 1, 9:41:32 PM	15.26s	

[Back](#)

1 2 3 4 5

[Continue](#)



Note: Formatting the name directory is expected to fail, if the directories are not empty.

Enable High Availability for HDFS

Congratulations!

Successfully enabled High Availability.

The following manual steps must be performed after completing this wizard:

- Configure the HDFS Web Interface Role of Hue service(s) **Hue** to be an HTTPS role instead of a NameNode. [Documentation](#)
- For each of the Hive service(s) **Hive**, stop the Hive service, back up the Hive Metastore Database to a persistent store, run the service command "Update Hive Metastore NameNodes", then restart the Hive services.

[Back](#)

1 2 3 4 5

[Finish](#)

12. In the next screen additional steps are suggested by the Cloudera Manager to update the Hue and Hive metastore. Click Finish for the screen shown above.



Note: The following subsections cover configuring Hue and Hive for HA as needed.

Federation and High Availability

Add Nameservice				
Name	Highly Available	Automatic Failover	NameNode	SecondaryNameNode
nameservice1	Yes	Yes	NameNode_rhel1 (Active) NameNode_rhel2 (Standby)	
				Actions ▾

13. In the Cloudera Manager, Click Home > HDFS > Instances to see Namenode in High Availability.

Configuring Hive Metastore to Use HDFS HA

To configure the Hive Megastore to use HDFS high availability, complete the following steps:

1. Go the Hive service.
2. Select Actions > Stop.
3. Click Stop to confirm the command.
4. Back up the Hive metastore database (if any existing data is present).
5. Select Actions> Update Hive Metastore NameNodes and confirm the command.
6. Select Actions> Start.
7. Restart the Hue and Impala services if stopped prior to updating the Metastore.

Configuring Hue to Work with HDFS HA

To configure Hue to work with HDFS HA, complete the following steps:

1. Go to the HDFS service.
2. Click the Instances tab.
3. Click Add Role Instances.
4. Select the text box below the HttpFS field. The Select Hosts dialog displays.
5. Select the host on which to run the role and click OK.
6. Click Continue.
7. Check the checkbox next to the HttpFS role and select Actions for Selected> Start.

Actions for Selected (1)					
	Role Type	State	Host	Commission State	Role Group
<input type="checkbox"/>	Balancer	N/A	rhel1	Commissioned	Balancer Default Group
<input type="checkbox"/>	DataNode	Started	rhel7	Commissioned	DataNode Default Group
<input type="checkbox"/>	DataNode	Started	rhel5	Commissioned	DataNode Default Group
<input type="checkbox"/>	DataNode	Started	rhel4	Commissioned	DataNode Default Group
<input type="checkbox"/>	DataNode	Started	rhel1	Commissioned	DataNode Group 1
<input type="checkbox"/>	Failover Controller	Started	rhel1	Commissioned	Failover Controller Default Group
<input type="checkbox"/>	Failover Controller	Started	rhel2	Commissioned	Failover Controller Default Group
<input checked="" type="checkbox"/>	HttpFS	Stopped	rhel3	Commissioned	HttpFS Default Group
<input type="checkbox"/>	JournalNode	Started	rhel1	Commissioned	JournalNode Default Group

8. After the command has completed, go to the Hue service.
9. Click the **Configuration** tab.
10. Locate the **HDFS Web Interface Role** property or search for it by typing its name in the Search box.
11. Select the HttpFS role that was just created instead of the NameNode role, and save your changes.

The screenshot shows the Hue Configuration interface. The top navigation bar includes Status, Instances, Configuration (selected), Commands, Charts Library, Audits, Hue Web UI, Quick Links, and Actions. The main area is titled "Configuration" with tabs for "Switch to the classic layout", "Role Groups", and "History and Rollback". On the left, there's a "Filters" sidebar with sections for STATUS (Error: 0, Warning: 1, Edited: 1, Non-default: 11, Has Overrides: 1) and SCOPE (Hue (Service-Wide): 112, Hue Server: 59, Kerberos Ticket Renewer: 33, Load Balancer: 39). The main configuration pane shows three properties under "HDFS Web Interface": "Role" (set to "HttpFS (rhel3)", "webhdfs_url"), "Oozie Service" (set to "Oozie"), and "HBase Service" (set to "HBase"). At the bottom, there are fields for "Edited Value" (1 Edited Value) and "Reason for change...", and a prominent "Save Changes" button.

12. Restart the Hue service.



Note: Refer to the Cloudera website: http://www.cloudera.com/documentation/enterprise/5-3-x/topics/cdh_hag_hdfs_ha_cdh_components_config.html#concept_rj1_hsq_bp for further details on setting up HA for other components like Impala, Oozie, etc.

YARN High Availability

The YARN Resource Manager (RM) is responsible for tracking the resources in a cluster and scheduling applications (for example, MapReduce jobs). Before CDH 5, the RM was a single point of failure in a YARN cluster. The RM high availability (HA) feature adds redundancy in the form of an Active/Standby RM pair to remove this single point of failure. Furthermore, upon failover from the Standby RM to the Active, the applications can resume from their last check-pointed state; for example, completed map tasks in a MapReduce job are not re-run on a subsequent attempt. This allows events such the following to be handled without any significant performance effect on running applications.

- Unplanned events such as machine crashes.
- Planned maintenance events such as software or hardware upgrades on the machine running the ResourceManager

For more information, go to:

http://www.cloudera.com/documentation/enterprise/latest/topics/cdh_hag_rm_ha_config.html#xd_583c10bfd326ba--43d5fd93-1410993f8c2--7f77

Setting up YARN HA

To set up YARN HA, complete the following steps:

1. Log in to the Cloudera manager and go to the YARN service.
2. Select Actions> Enable High Availability.

A screen showing the hosts that are eligible to run a standby ResourceManager displays.

The host where the current ResourceManager is running is not available as a choice.

3. Select the host (rhel3) where the standby ResourceManager is to be installed, and click Continue.

Enable High Availability for YARN (MR2 Included)

Getting Started

This wizard leads you through adding a standby ResourceManager, restarting this YARN (MR2 Included) service and any dependent services, and then re-deploying client configurations.



Cloudera Manager proceeds to execute a set of commands that stop the YARN service, add a standby ResourceManager, initialize the ResourceManager high availability state in ZooKeeper, restart YARN, and redeploy the relevant client configurations.

Enable High Availability for YARN (MR2 Included)

✓ Enable ResourceManager HA Command

Status: **Finished** Context: [YARN \(MR2 Included\)](#) Start Time: May 1, 10:06:15 PM Duration: 4.9m

Successfully enabled ResourceManager HA.

Details		Completed 4 of 4 step(s).			<input checked="" type="radio"/> All	<input type="radio"/> Failed Only	<input type="radio"/> Running Only
Step	Context	Start Time	Duration	Actions			
➤ ✓ Stop	Cluster 1	May 1, 10:06:15 PM	90.89s				
All services successfully stopped.							
➤ ✓ Add Standby ResourceManager	ResourceManager (rhel3)	May 1, 10:07:46 PM	2ms				
Successfully added new ResourceManager to YARN (MR2 Included) on rhel3.							
➤ ✓ Start	Cluster 1	May 1, 10:07:47 PM	3m				
All services successfully started.							
➤ ✓ Deploy Client Configuration	Cluster 1	May 1, 10:10:50 PM	16.03s				
Successfully deployed all							

Back
1 2 3
Finish

- Click **Finish** once the installation is completed successfully.

Configuring Yarn (MR2 Included) and HDFS Services

The parameters in Table 8 are used for Cisco UCS Integrated Infrastructure for Big Data and Analytics Performance Optimized cluster configuration described in this document. These parameters are to be changed based on the cluster configuration, number of nodes and specific workload.

Table 8 YARN

Service	Value
mapreduce.map.memory.mb	3GiB
mapreduce.reduce.memory.mb	3GiB
mapreduce.map.java.opts.max.heap	2560 MiB
yarn.nodemanager.resource.memorymb	180 GiB
yarn.nodemanager.resource.cpu-vcores	32
yarn.scheduler.minimum-allocation-mb	4 GiB
yarn.scheduler.maximum-allocation-mb	180 GiB
yarn.scheduler.maximum-allocation-vcores	48
mapreduce.task.io.sort.mb	256 MiB

HDFS

Service	Value
dfs.datanode.failed.volumes.tolerated	6
dfs.datanode.du.reserved	50 GiB
dfs.datanode.data.dir.perm	755
Java Heap Size of Namenode in Bytes	2628 MiB
dfs.namenode.handler.count	54
dfs.namenode.service.handler.count	54
Java Heap Size of Secondary namenode in Bytes	2628 MiB

Apache Kafka Installation and Configuration

Cloudera Manager 5.4 or higher includes the Kafka service. To install, download Kafka using Cloudera Manager, distribute Kafka to the cluster, activate the new parcel, and add the service to the cluster.

1. Download the Kafka Parcels as shown below.
2. On a server that is accessible to the internet.
3. Create a directory for the Kafka parcels.

```
#mkdir /tmp/Kafka2.0.1Parcels
#cd /tmp/Kafka2.0.1Parcels
#wget http://archive.cloudera.com/kafka/parcels/2.0.1/KAFKA-2.0.1-1.2.0.1.p0.5-
e17.parcel
#wget http://archive.cloudera.com/kafka/parcels/2.0.1/KAFKA-2.0.1-1.2.0.1.p0.5-
e17.parcel.shal

#wget http://archive.cloudera.com/kafka/parcels/2.0.1/manifest.json
```

```
[root@linuxJbR4 disk1]# cd Kafka2.0.1Parcels/
[root@linuxJbR4 Kafka2.0.1Parcels]# wget http://archive.cloudera.com/kafka/parcels/2.0.1/KAFKA-2.0.1-1.2.0.1.p0.5-el7.parcel
--2016-05-01 18:43:21--  http://archive.cloudera.com/kafka/parcels/2.0.1/KAFKA-2.0.1-1.2.0.1.p0.5-el7.parcel
Resolving archive.cloudera.com... 23.235.47.167
Connecting to archive.cloudera.com[23.235.47.167]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 31969501 (30M)
Saving to: "KAFKA-2.0.1-1.2.0.1.p0.5-el7.parcel"

100%[=====] 31,969,501 11.1M/s  in 2.7s
2016-05-01 18:43:24 (11.1 MB/s) - "KAFKA-2.0.1-1.2.0.1.p0.5-el7.parcel" saved [31969501/31969501]

[root@linuxJbR4 Kafka2.0.1Parcels]# wget http://archive.cloudera.com/kafka/parcels/2.0.1/KAFKA-2.0.1-1.2.0.1.p0.5-el7.parcel.sha1
--2016-05-01 18:43:30--  http://archive.cloudera.com/kafka/parcels/2.0.1/KAFKA-2.0.1-1.2.0.1.p0.5-el7.parcel.sha1
Resolving archive.cloudera.com... 23.235.47.167
Connecting to archive.cloudera.com[23.235.47.167]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 41 [application/x-sha1]
Saving to: "KAFKA-2.0.1-1.2.0.1.p0.5-el7.parcel.sha1"

100%[=====] 41 --.-K/s  in 0s
2016-05-01 18:43:31 (10.3 MB/s) - "KAFKA-2.0.1-1.2.0.1.p0.5-el7.parcel.sha1" saved [41/41]

[root@linuxJbR4 Kafka2.0.1Parcels]# wget http://archive.cloudera.com/kafka/parcels/2.0.1/manifest.json
--2016-05-01 18:43:37--  http://archive.cloudera.com/kafka/parcels/2.0.1/manifest.json
Resolving archive.cloudera.com... 23.235.47.167
Connecting to archive.cloudera.com[23.235.47.167]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4180 (4.1K) [application/json]
Saving to: "manifest.json"

100%[=====] 4,180 --.-K/s  in 0s
2016-05-01 18:43:37 (534 MB/s) - "manifest.json" saved [4180/4180]
```

4. Change the contents of manifest.json to match the following:

```
{
  "lastUpdated": 14598141460000,
  "parcels": [
    {
      "parcelName": "KAFKA-2.0.1-1.2.0.1.p0.5-el7.parcel",
      "components": [
        {
          "pkg_version": "0.9.0+kafka2.0.1",
          "pkg_release": "1.2.0.1.p0.5",
          "name": "kafka",
          "version": "0.9.0-kafka2.0.1"
        }
      ],
      "depends": "CDH (>= 5.2), CDH (<< 6.0)",
      "replaces": "CLABS_KAFKA",
      "hash": "180d8322f2026f2b3609741216d2c25dd2dfb294"
    }
  ]
}
```

```

    }
]
}

```

5. Copy the Kafka parcels over to rhel1 under /var/www/html.

```
#scp -r Kafka2.0.1Parcels rhel1:/var/www/html/
```

6. From the browser go to the parcels at <http://10.13.1.50:7180/cmf/parcel/status>.

7. Click Configure. Add in a new parcel by clicking on the + button and giving the path to the new Kafka parcels that were downloaded in the previous step.

Parcel Settings

Parcel Directory /opt/cloudera/parcels ?

! Requires Agent Restart

Local Parcel Repository Path /opt/cloudera/parcel-repo ?

Parcel Update Frequency 1 hour(s) ?

Remote Parcel Repository URLs

- http://10.13.1.50/CDH5.7Parcel/ + - c ?
- http://10.13.1.50/Kafka2.0.1Parcels/ + - ?

Create System-Wide Symlinks for Active Parcels ?

Cloudera Manager Manages ?

Reason for change... Save Changes Cancel

8. Click Save Changes.
9. Click **Download** and then **Distribute** to download and distribute the parcels.
10. Click **Activate** to add the service to the cluster.

Packages

The screenshot shows the Cloudera Manager interface for managing parcels. On the left, there's a sidebar with 'Location' set to 'Cluster 1' and 'Available Remotely'. Below it are 'Filters' for 'PARCEL NAME' (All, CDH 5, KAFKA) and 'STATUS' (All, Distributed). The main area, titled 'Cluster 1', lists two parcels:

Parcel Name	Version	Status	Actions
CDH 5	5.7.0-1.cdh5.7.0.p0.45	Distributed, Activated	<button>Deactivate</button>
KAFKA	2.0.1-1.2.0.1.p0.5	Distributed	<button>Activate</button> ▾

Parcel Distribution Status



Configuring Spark

The two main resources that Spark (and YARN) are dependent on are CPU and memory. Disk and network I/O, of course, play a part in Spark performance as well, but neither Spark nor YARN currently can actively manage them. Every Spark executor in any application has the same fixed number of cores and same fixed heap size. The number of cores can be specified with the `executor-cores` flag when invoking `spark-submit`, `spark-shell`, and `pyspark` from the command line, or by setting the `spark.executor.cores` property in the `spark-defaults.conf` file or in the `SparkConf` object.

And the heap size can be controlled with the `executor-memory` flag or the `spark.executor.memory` property. The `cores` property controls the number of concurrent tasks an executor can run, `executor-cores = 5` mean that each executor can run a maximum of five tasks at the same time. The `memory` property impacts the amount of data Spark can cache, as well as the maximum sizes of the shuffle data structures used for grouping, aggregations, and joins.

The `num-executors` command-line flag or `spark.executor.instances` configuration property control the number of executors requested. Dynamic Allocation can be enabled from CDH5.4 instead setting the `spark.dynamicAllocation.enabled` to true. Dynamic allocation enables a Spark application to request executors when there is a backlog of pending tasks and free up executors when idle.

Asking for five executor cores will result in a request to YARN for five virtual cores. The memory requested from YARN is a little more complex for a couple reasons:

- `executor-memory/spark.executor.memory` controls the executor heap size, but JVMs can also use some memory off heap, for example for VM overhead, interned Strings and direct byte buffers. The value of the `spark.yarn.executor.memoryOverhead` property is added to the executor memory to determine the full memory request to YARN for each executor. It defaults to max (384, 0.10 * `spark.executor.memory`) .
- YARN may round the requested memory up a little. YARN's `yarn.scheduler.minimum-allocation-mb` and `yarn.scheduler.increment-allocation-mb` properties control the minimum and increment request values respectively.
- The application master is a non-executor container with the special capability of requesting containers from YARN, takes up resources of its own that must be budgeted in. In *yarn-client* mode, it defaults to a 1024MB and one vcore. In *yarn-cluster* mode, the application master runs the driver, so it's often useful to add its resources with the `-driver-memory` and `-driver-cores` properties.
- Running executors with too much memory often results in excessive garbage collection delays. 64GB is a rough guess at a good upper limit for a single executor.
- A good estimate is that at most five tasks per executor can achieve full write throughput, so it's good to keep the number of cores per executor around that number.
- Running tiny executors (with a single core and just enough memory needed to run a single task, for example) throws away the benefits that come from running multiple tasks in a single JVM. For example, broadcast variables need to be replicated once on each executor, so many small executors will result in many more copies of the data.

Tuning Resource Allocation for Spark

Below is an example of configuring a Spark application to use as much of the cluster as possible, we are using an example cluster with 16 nodes running NodeManagers, each equipped with 56 cores and 256GB of memory.
`yarn.nodemanager.resource.memory-mb` and `yarn.nodemanager.resource.cpu-vcores` should be set to $180 * 1024 = 184320$ (megabytes) and 48 respectively.

```
spark.executor.instances/num-executors = 63
spark.executor.cores/--executor-cores = 5
spark.executor.memory/--executor-memory = 41G
```

This configuration results in four executors on all nodes except for the one with the AM, which will have three executors.

`executor-memory` is derived as $(180/4 \text{ executors per node}) = 45; 45 * 0.10 = 4.5; 45 - 4.5 \sim 40$.

For taking care of long running processes use 2G for the spark driver

```
spark.driver.memory = 2G
```

Submitting a Job

```
--driver-memory 2G --executor-memory 40G --num-executors 63 --executor-cores 5 --properties-file /opt/cloudera/parcels/CDH/etc/spark/conf.dist/spark-defaults.conf
```

In *yarn-cluster* mode, the local directories used by the Spark executors and the Spark driver will be the local directories configured for YARN (Hadoop YARN config `yarn.nodemanager.local-dirs`). If the user specifies `spark.local.dir`, it will be ignored.

In *yarn-client* mode, the Spark executors will use the local directories configured for YARN while the Spark driver will use those defined in `spark.local.dir`. The Spark driver does not run on the YARN cluster in *yarn-client* mode, only the Spark executors do.

`spark.local.dir /tmp` (Directory to use for "scratch" space in Spark, including map output files and RDDs that get stored on disk. This should be on a fast, local disk in your system).

Every Spark stage has a number of tasks, each of which processes data sequentially. In tuning Spark jobs, this parallelism number is the most important parameter in determining performance. The number of tasks in a stage is the same as the number of partitions in the last RDD in the stage. The number of partitions in an RDD is the same as the number of partitions in the RDD on which it depends, with a couple exceptions: the coalesce transformation allows creating an RDD with fewer partitions than its parent RDD, the union transformation creates an RDD with the sum of its parents' number of partitions, and Cartesian creates an RDD with their product.

RDDs produced by a file have their partitions determined by the underlying MapReduce InputFormat that's used. Typically there will be a partition for each HDFS block being read. Partitions for RDDs produced by parallelize come from the parameter given by the user, or `spark.default.parallelism` if none is given.

The primary concern is that the number of tasks will be too small. If there are fewer tasks than slots available to run them in, the stage won't be taking advantage of all the CPU available.

If the stage in question is reading from Hadoop, your options are:

- Use the repartition transformation, which will trigger a shuffle.
- Configure your InputFormat to create more splits.
- Write the input data out to HDFS with a smaller block size.

If the stage is getting its input from another stage, the transformation that triggered the stage boundary will accept a `numPartitions` argument.

The most straightforward way to tune the number of partitions is experimentation: Look at the number of partitions in the parent RDD and then keep multiplying that by 1.5 until performance stops improving.

In contrast with MapReduce for Spark when in doubt, it is almost always better to be on the side of a larger number of tasks (and thus partitions).

Shuffle Performance Improvement

`spark.shuffle.compress true` (compress map output files)

`spark.broadcast.compress true` (compress broadcast variables before sending them)

`spark.io.compression.codec org.apache.spark.io.SnappyCompressionCodec` (codec used to compress internal data such as RDD partitions, broadcast variables and shuffle outputs)

`spark.shuffle.spill.compress true` (Whether to compress data spilled during shuffles.)

`spark.shuffle.io.numConnectionsPerPeer 4` (Connections between hosts are reused in order to reduce connection buildup for large clusters. For clusters with many hard disks and few hosts, this may result in insufficient concurrency to saturate all disks, and so users may consider increasing this value.)

`spark.shuffle.file.buffer 64K` (Size of the in-memory buffer for each shuffle file output stream. These buffers reduce the number of disk seeks and system calls made in creating intermediate shuffle file)

Improving Serialization Performance

Serialization plays an important role in the performance of any distributed application. Often, this will be the first thing that should be tuned to optimize a Spark application.

```

spark.serializer org.apache.spark.serializer.KryoSerializer (when speed is necessary)
spark.kryo.referenceTracking false
spark.kryoserializer.buffer 2000 (If the objects are large, may need to increase the size further to fit the size
of the object being deserialized).

```

SparkSQL is ideally suited for mixed procedure jobs where SQL code is combined with Scala, Java, or Python programs. In general the SparkSQL command line interface is used for single user operations and ad hoc queries.

For multi-user SparkSQL environments, it is recommended to use a Thrift server connected via JDBC.

Spark SQL Tuning

Below are some guidelines for Spark SQL tuning:

1. To compile each query to Java bytecode on the fly, turn on `sqlcodegen`. This can improve performance for large queries, but can slow down very short queries.

```

spark.sql_codegen true
spark.sql_unsafe.enabled true

```

2. Configuration of in-memory caching can be done using the `setConf` method on `SQLContext` or by running `SET key=value` commands using SQL.
3. `spark.sql.inMemoryColumnarStorage.compressed true` (will automatically select a compression codec for each column based on statistics of the data)
4. `spark.sql.inMemoryColumnarStorage.batchSize 5000` (Controls the size of batches for columnar caching. Larger batch sizes can improve memory utilization and compression, but risk OOMs when caching data)
5. The columnar nature of the ORC format helps avoid reading unnecessary columns, but it is still possible to read unnecessary rows. ORC avoids this type of overhead by using predicate push-down with three levels of built-in indexes within each file: file level, stripe level, and row level. This combination of indexed data and columnar storage reduces disk I/O significantly, especially for larger datasets where I/O bandwidth becomes the main bottleneck for performance.
6. By default, ORC predicate push-down is disabled in Spark SQL. To obtain performance benefits from predicate push-down, enable it explicitly, as follows:

7. `spark.sql.orc.filterPushdown=true`
8. In SparkSQL to automatically determine the number of reducers for joins and groupbys, use the parameter,
9. `spark.sql.shuffle.partitions 200`, (default value is 200)

This property can be put into `hive-site.xml` to override the default value.

10. Set log to WARN in `log4j.properties` to reduce log level.



Note: Running Thrift server and connecting to spark-sql through beeline is the recommended option for multi-session testing.

Compression for Hive

Set the following Hive parameters to compress the Hive output files using Snappy compression:

```
hive.exec.compress.output=true
hive.exec.orc.default.compress=SNAPPY
```

Changing the Log Directory for All Applications

To change the default log from the /var prefix to /data/disk1, complete the following steps:

1. Log into the cloudera home page and click My Clusters.
2. From the configuration drop-down list select "All Log Directories."

Reason for change...		Save Changes	27 Edited Values
DataNode Log Directory	DataNode Default Group	/data/disk1/log/hadoop-hdfs	Directory where DataNode will place its log files.
Failover Controller Log Directory	Failover Controller Default Group	/data/disk1/log/hadoop-hdfs	Directory where Failover Controller will place its log files.
HttpFS Log Directory	HttpFS Default Group	/data/disk1/hadoop-https	Directory where HttpFS will place its log files.
JournalNode Log Directory	JournalNode Default Group	/data/disk1/hadoop-hdfs	Directory where JournalNode will place its log files.
NFS Gateway Log Directory	NFS Gateway Default Group	/data/disk1/hadoop-hdfs	Directory where NFS Gateway will place its log files.
NameNode Log Directory	NameNode Default Group	/data/disk1/hadoop-hdfs	Directory where NameNode will place its log files.
SecondaryNameNode Log Directory	SecondaryNameNode Default Group	/data/disk1/hadoop-hdfs	Directory where SecondaryNameNode will place its log files.
Hive Metastore Server Log Directory	Hive Metastore Server Default Group	/data/disk1/log/hive	Directory where Hive Metastore Server will place its log files.
HiveServer2 Log Directory	HiveServer2 Default Group	/data/disk1/log/hive	Directory where HiveServer2 will place its log files.

WebHCat Server Log Directory	WebHCat Server Default Group C /data/disk1/log/hcatalog	Directory where WebHCat Server will place its log files.
Hue Server Log Directory	Hue Server Default Group C /data/disk1/log/hue	Directory where Hue Server will place its log files.
Kerberos Ticket Renewer Log Directory	Kerberos Ticket Renewer Default Group C /data/disk1/log/hue	Directory where Kerberos Ticket Renewer will place its log files.
Catalog Server Log Directory log_dir	Impala Catalog Server Default Group C /data/disk1/log/catalogd	Directory where Catalog Server will place its log files.
Impala Daemon Log Directory log_dir	Impala Daemon Default Group C /data/disk1/log/impalad	Directory where Impala Daemon will place its log files.
	Impala Daemon Group 1 C /data/disk1/log/impalad	
Llama Log Directory llama_log_dir	Impala Llama ApplicationMaster Default Group C /data/disk1/log/impala-llama	Directory where Llama will place its log files.
StateStore Log Directory log_dir	Impala StateStore Default Group C /data/disk1/log/statestore	Directory where StateStore will place its log files.
JobTracker Log Directory hadoop.log.dir	JobTracker Default Group C /data/disk1/log/hadoop-0.20-mapreduce	Directory where JobTracker will place its log files.
TaskTracker Log Directory hadoop.log.dir	TaskTracker Default Group C /data/disk1/log/hadoop-0.20-mapreduce	Directory where TaskTracker will place its log files.
	TaskTracker Group 1 C /data/disk1/log/hadoop-0.20-mapreduce	
Oozie Server Log Directory	Oozie Server Default Group C /data/disk1/log/oozie	Directory where Oozie Server will place its log files.
Solr Server Log Directory	Solr Server Default Group C /data/disk1/log/solr	Directory where Solr Server will place its log files.
History Server Log Directory log_dir	History Server Default Group C /data/disk1/log/spark	The log directory for log files of the role History Server.
Sqoop 2 Server Log Directory	Sqoop 2 Server Default Group C /data/disk1/log/sqoop	Directory where Sqoop 2 Server will place its log files.
JobHistory Server Log Directory hadoop.log.dir	JobHistory Server Default Group C /data/disk1/hadoop-mapreduce	Directory where JobHistory Server will place its log files.
NodeManager Log Directory hadoop.log.dir	NodeManager Default Group C /data/disk1/log/hadoop-yarn	Directory where NodeManager will place its log files.
ResourceManager Log Directory hadoop.log.dir	ResourceManager Default Group C /data/disk1/log/yarn	Directory where ResourceManager will place its log files.
ZooKeeper Log Directory	Server Default Group C /data/disk1/log/zookeeper	Directory where ZooKeeper will place its log files.
Display <input type="button" value="25"/> Entries		First Previous 1 2 Next Last

3. Click Save changes.

Cloudera Data Science Workbench (CDSW)

CDSW has pre-requisites, one of which is CUDA. CUDA itself, also has prerequisites. The order of installation is:

- CUDA pre-requisites
- CUDA
- CDSW pre-requisites
- CDSW

Installing Prerequisites for CUDA



Note: Details of CUDA installation can be found here: <http://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html>.



Note: These commands are run as root or sudo.

List GPUs and CPUs installed:

```
lspci | grep -i nvidia  
  
# lspci | grep -i nvidia  
5e:00.0 3D controller: NVIDIA Corporation GP100GL [Tesla P100 PCIe 16GB] (rev a1)  
af:00.0 3D controller: NVIDIA Corporation GP100GL [Tesla P100 PCIe 16GB] (rev a1)  
#lscpu
```

```
[root@UCS ~]# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                56
On-line CPU(s) list:   0-55
Thread(s) per core:    2
Core(s) per socket:    14
Socket(s):              2
NUMA node(s):           2
Vendor ID:              GenuineIntel
CPU family:             6
Model:                 85
Model name:             Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz
Stepping:               4
CPU MHz:                1031.062
CPU max MHz:            3700.0000
CPU min MHz:            1000.0000
BogoMIPS:               5200.00
Virtualization:         VT-x
L1d cache:              32K
L1i cache:              32K
L2 cache:                1024K
L3 cache:                19712K
NUMA node0 CPU(s):      0-13,28-41
NUMA node1 CPU(s):      14-27,42-55
```

GCC Installation

Make sure gcc is installed in the system:

```
# gcc --version
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-16)
Copyright (C) 2015 Free Software Foundation, Inc.

This is free software; see the source for copying conditions.
```

Install Kernel Headers and Installation Packages

```
# yum install kernel-devel-$ (uname -r) kernel-headers-$ (uname -r)
# yum install kernel-devel-$ (uname -r) kernel-headers-$ (uname -r)
Package kernel-devel-3.10.0-693.el7.x86_64 already installed and latest version
Package kernel-headers-3.10.0-693.el7.x86_64 already installed and latest version
Nothing to do
```

Install dkms

The NVIDIA driver RPM packages depend on other external packages, such as DKMS. Those packages are only available on third-party repositories, such as [EPEL](#).

<http://rpmfind.net/linux/rpm2html/search.php?query=dkms> for RHEL 7.x

RHEL 7.x http://rpmfind.net/linux/epel/7/x86_64/Packages/d/dkms-2.4.0-1.20170926git959bd74.el7.noarch.rpm

```
#wget http://rpmfind.net/linux/epel/7/x86_64/Packages/d/dkms-2.4.0-1.20170926git959bd74.el7.noarch.rpm
```

Copy dkms rpm to all the GPU servers

Install dkms with yum install

```
yum install dkms-2.4.0-1.20170926git959bd74.el7.noarch.rpm
```

Install NVIDIA GPU Drivers

Download this NVIDIA GPU driver from <http://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html#installdriver>.

```
# wget http://us.download.nvidia.com/tesla/390.12/nvidia-diag-driver-local-repo-rhel7-390.12-1.0-1.x86_64.rpm
```

```
# rpm -ivh nvidia-diag-driver-local-repo-rhel7-390.12-1.0-1.x86_64.rpm
```

```
warning: nvidia-diag-driver-local-repo-rhel7-390.12-1.0-1.x86_64.rpm: Header V3 RSA/SHA512 Signature, key ID 7fa2af80: NOKEY
```

```
Preparing... ##### [100%]
```

```
Updating / installing...
```

```
1:nvidia-diag-driver-local-repo-rhel7-390.12-1.0-1.x86_64 ##### [100%]
```

Installing CUDA

Download CUDA 8 (Tensorflow 1.4 needed CUDA 8, later version of tensorflow uses CUDA 9).



Note: Tensorflow needs CUDA version 8 or 9, make sure which version of CUDA is supported by tensorflow before installing CUDA. Earlier versions of CUDA are at <https://developer.nvidia.com/cuda-toolkit-archive>.

Latest version of CUDA is available at the following link:

https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&target_distro=RHEL&target_version=7&target_type=rpmlocal

[This CVD is showing sample steps to install CUDA 8 to use Tensorflow 1.4. Please use the appropriate CUDA as needed.](#)

```
#wget https://developer.nvidia.com/compute/cuda/8.0/Prod2/local_installers/cuda-repo-rhel7-8-0-local-ga2-8.0.61-1.x86_64-rpm
```

```
#rpm -i cuda-repo-rhel7-8-0-local-ga2-8.0.61-1.x86_64.rpm
```

```
warning: cuda-repo-rhel7-8-0-local-ga2-8.0.61-1.x86_64.rpm: Header V3 RSA/SHA512 Signature, key ID 7fa2af80: NOKEY
```

```
# yum clean all
```

```
Cleaning repos: cuda-8-0-local-ga2 nvidia-diag-driver-local-390.12 rhel7.4
```

```
Cleaning up everything
```

Install CUDA on the GPU nodes as follows:

```
[root@cdsw1 ~]# yum install cuda -nogpgcheck
# yum install cuda --nogpgcheck

Total size: 1.5 G
Total download size: 1.5 G
Installed size: 2.5 G
Is this ok [y/d/N]: y
```

```
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-
manager to register.
cuda-8-0-local-ga2
...

```

```
Installed:
  cuda.x86_64 0:8.0.61-1

Dependency Installed:
  cuda-8-0.x86_64 0:8.0.61-1
  cuda-cUBLAS-dev-8-0.x86_64 0:8.0.61-1
  cuda-cUFFT-dev-8-0.x86_64 0:8.0.61-1
  cuda-cUSolver-dev-8-0.x86_64 0:8.0.61-1
  cuda-documentation-8-0.x86_64 0:8.0.61-1
  cuda-misc-headers-8-0.x86_64 0:8.0.61-1
  cuda-nvGraph-dev-8-0.x86_64 0:8.0.61-1
  cuda-runtime-8-0.x86_64 0:8.0.61-1
                                         cuda-command-line-tools-8-0.x86_64 0:8.0.61-1
                                         cuda-cuDart-8-0.x86_64 0:8.0.61-1
                                         cuda-curand-8-0.x86_64 0:8.0.61-1
                                         cuda-cusparse-8-0.x86_64 0:8.0.61-1
                                         cuda-driver-dev-8-0.x86_64 0:8.0.61-1
                                         cuda-npp-8-0.x86_64 0:8.0.61-1
                                         cuda-nvML-dev-8-0.x86_64 0:8.0.61-1
                                         cuda-samples-8-0.x86_64 0:8.0.61-1
```

Download and Setup NVIDIA CUDA Deep Neural Network library (cuDNN)

Download cuDNN 6.5

Download cuDNN from <https://developer.nvidia.com/cudnn> for the same CUDA version.

Copy it into all the GPU servers.

<https://developer.nvidia.com/rdp/cudnn-archive>

```
tar -xzvf cudnn-6.5-linux-x64-v2.tgz
```

```
[root@cdsw1 cudnn-6.5-linux-x64-v2]# pwd
/root/cudnn-6.5-linux-x64-v2
[root@cdsw1 cudnn-6.5-linux-x64-v2]# ls
cudnn.h  CUDNN_License.pdf  INSTALL.txt  libcudnn.so  libcudnn.so.6.5  libcudnn.so.6.5.48  libcudnn_static.a
[root@cdsw1 cudnn-6.5-linux-x64-v2]#
```

```
cp cudnn.h /usr/local/cuda-8.0/include
```

```
cp libcudnn* /usr/local/cuda-8.0/lib64
```

```
chmod a+r /usr/local/cuda-8.0/include/cudnn.h /usr/local/cuda-8.0/lib64/libcudnn*
```

```
[root@UCS ~]# cp cuda/include/cudnn.h /usr/local/cuda-9.1/include
```

```
[root@UCS ~]# cp cuda/lib64/libcudnn* /usr/local/cuda-9.1/lib64
```

```
[root@UCS ~]# chmod a+r /usr/local/cuda-9.1/include/cudnn.h /usr/local/cuda-9.1/lib64/libcudnn*
```

Post Installation Steps

Add CUDA in PATH and LD_LIBRARY_PATH variable

The PATH and LD_LIBRARY_PATH variable needs to include /usr/local/<cuda 8 or 9 path>/bin

```
$ export PATH=/usr/local/<cuda location>/bin${PATH:+:$PATH}
$ export LD_LIBRARY_PATH=/usr/local//<cuda location>/lib\
${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```



Note: For more information, refer to: <https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#post-installation-actions>.

Verify Drivers

Restart the servers to verify the drivers have been installed:

```
#nvidia-smi
```

```
[root@cdsw1 ~]# nvidia-smi
Mon Mar 19 17:32:16 2018
+-----+
| NVIDIA-SMI 390.12          Driver Version: 390.12      |
|-----+
| GPU  Name     Persistence-MI Bus-Id     Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
|-----+
|  0  Tesla P100-PCIE... Off  | 00000000:5E:00.0 Off |                0 |
| N/A   31C    P0    26W / 250W |    10MiB / 16280MiB |     0%       Default |
+-----+
|  1  Tesla P100-PCIE... Off  | 00000000:AF:00.0 Off |                0 |
| N/A   30C    P0    24W / 250W |    10MiB / 16280MiB |     0%       Default |
+-----+
+-----+
| Processes:                               GPU Memory |
| GPU  PID  Type  Process name        Usage        |
|-----+
| No running processes found            |
+-----+
```

```
[root@cdsw1 bin]# pwd
```

```
/usr/local/cuda-8.0/bin
```

```
[root@cdsw1 bin]# ./nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2017 NVIDIA Corporation
Built on Fri Nov 3 21:07:56 CDT 2017
Cuda compilation tools, release 8.0, v8.0
```

Based on the Driver version, add the following in ~/.bash_profile of root

```
export NVIDIA_DRIVER_VERSION=390.12
```

Installation Prerequisites for CDSW

Complete the following steps on all CDSW nodes:



For more information, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_requirements_supported_versions.html#cdsw_requirements_supported_versions and https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_install.html#pre_install.

Set Up a Wildcard DNS Subdomain

Cloudera Data Science Workbench uses subdomains to provide isolation for user-generated HTML and JavaScript, and routing requests between services. To set up subdomains for Cloudera Data Science Workbench, configure your DNS server with an A record for a wildcard DNS name such as `*.cdsw.<your_domain>.com` for the master host, and a second A record for the root entry of `cdsw.<your_domain>.com`.

```
[root@cdsw ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6
10.13.1.50   rhel1
10.13.1.51   rhel2
10.13.1.52   rhel3
10.13.1.53   rhel4
10.13.1.54   rhel5
10.13.1.55   rhel6
10.13.1.56   rhel7
10.13.1.57   rhel8
...
10.13.1.73   rhel24

10.13.1.250  cdsw1.cisco.com
10.13.1.251  cdsw2.cisco.com
10.13.1.252  cdsw3.cisco.com
10.13.1.253  cdsw4.cisco.com
```

You can also use a wildcard CNAME record if it is supported by your DNS provider.



This Wildcard DNS subdomain need to be used by the jump host/edge node or the bastion server as well.

With dnsmasq, to add a wildcard DNS subdomain, do the following for just the master host.

Update `/etc/dnsmasq.conf` to enable Wildcard entry

```
address=/cdsw/10.13.1.250
```

```
#service dnsmasq restart
```

Test the working of wildcard DNS with dig or nslookup

```
#nslookup *.cdsw.<your domain>.com
```

```
#dig cdsw.<your_domain>.com
```

```
#dig *.cdsw.<your domain>.com
```



For more information, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_install.html#pre_install.

Supported JDK Version

The entire CDH cluster, including Cloudera Data Science Workbench gateway nodes, must use Oracle JDK. OpenJDK is not supported by CDH, Cloudera Manager, or Cloudera Data Science Workbench.

Spark 2.2 which is needed by CDSW to run Spark jobs (on Hadoop nodes) requires JDK 1.8. On CSD-based deployments, Cloudera Manager automatically detects the path and version of Java installed on Cloudera Data Science Workbench gateway hosts.

To upgrade your entire CDH cluster to JDK 1.8, see [Upgrading to Oracle JDK 1.8](#).



For more details https://www.cloudera.com/documentation/enterprise/release-notes/topics/rn_consolidated_pcm.html#pcm_jdk

IP Tables and Security on CDSW Nodes

Disable all pre-existing iptables rules. While Kubernetes makes extensive use of iptables, it is difficult to predict how pre-existing iptables rules will interact with the rules inserted by Kubernetes. Therefore, Cloudera recommends you use the following commands to disable all pre-existing rules before you proceed with the installation.

```
yum -y install iptables-service  
yum install initscripts  
systemctl stop firewalld  
systemctl mask firewalld  
systemctl disable firewalld  
systemctl enable iptables  
systemctl start iptables  
iptables -P INPUT ACCEPT  
iptables -P FORWARD ACCEPT  
iptables -P OUTPUT ACCEPT  
iptables -t nat -F  
iptables -t mangle -F  
iptables -F  
iptables -X  
service iptables save
```



For more information about Cloudera Data Science Workbench 1.3.x Requirements and Supported Platforms, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_requirements_supported_versions.html

Disable SELinux

```
sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
setenforce 0
sestatus
```

Configure Block Devices

Docker Block Device

The Cloudera Data Science Workbench installer will format and mount Docker on each gateway host. Make sure there is no important data stored on these devices. Do not mount these block devices prior to installation.

Application Block Device or Mount Point

The master host on Cloudera Data Science Workbench requires at least 500 GB for database and project storage. This recommended capacity is contingent on the expected number of users and projects on the cluster. While large data files should be stored on HDFS, it is not uncommon to find gigabytes of data or libraries in individual projects. Running out of storage will cause the application to fail. Make sure you continue to carefully monitor disk space usage and I/O using Cloudera Manager.



Note: To enable data resilience, enable this drive as RAID1 of SSDs (using commands as shown in configuring namenode).

Cloudera Data Science Workbench will store all application data at `/var/lib/cdsw`. In a CSD-based deployment, this location is not configurable. Cloudera Data Science Workbench will assume the system administrator has formatted and mounted one or more block devices to `/var/lib/cdsw`.

Regardless of the application data storage configuration you choose, `/var/lib/cdsw` must be stored on a separate block device (the RAID1 of SSDs created for this).

Download and Install CDSW with Cloudera Manager

To download and install CDSW, complete the following steps:

1. Download the Cloudera Data Science Workbench Custom Service Descriptor (CSD)
[CLOUDERA_DATA_SCIENCE_WORKBENCH-1.3.0.jar](#)



Note: For more information, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_install_parcel.html#csd.

2. Log on to the Cloudera Manager Server host, and place the CSD [CLOUDERA_DATA_SCIENCE_WORKBENCH-1.3.0.jar](#) file under `/opt/cloudera/csd`, which is the default location for CSD files.
3. Once cloudera Manager is installed under `/opt/cloudera/csd` set the file ownership to `cloudera-scm:cloudera-scm` with permission `644`
4. `chown cloudera-scm:cloudera-scm CLOUDERA_DATA_SCIENCE_WORKBENCH-1.3.0.jar`
5. `chmod 644 CLOUDERA_DATA_SCIENCE_WORKBENCH-1.3.0.jar`
6. Restart cloudera-management service

7. service cloudera-scm-server restart
8. Log into the Cloudera Manager Admin Console and restart the Cloudera Management Service.
 - a. Select Clusters > Cloudera Management Service.
 - b. Select Actions > Restart.
 - c. Download Cloudera CDSW parcel from <http://archive.cloudera.com/cdsw/1/parcels/1.3.0/> into the server hosting Cloudera parcels copy all these files to CDSW folder and move it to /var/www/html/
 - d. Log into the Cloudera Manager Admin Console.
 - e. Click Hosts > Parcels in the main navigation bar.

The screenshot shows the Cloudera Manager interface. At the top, there are tabs for 'Clusters', 'Hosts' (which is currently selected and highlighted in pink), 'Diagnostics', and 'Audits'. Below the tabs, the word 'Home' is displayed. Underneath 'Home', there are three links: 'Status', 'All Health Issues', and 'Configuration' (with a gear icon and the number '5'). A cluster summary for 'Cluster 1 (CDH 5.13.0, Parcels)' is shown, featuring a green checkmark icon and a blue link. To the right of the cluster summary, a vertical sidebar has a dropdown menu for 'Hosts'. The menu items are 'All Hosts', 'Roles', 'Host Templates', 'Disks Overview', and 'Parcels', with 'Parcels' being the active item and highlighted in blue.

- f. Download the Cloudera Data Science Workbench parcel, distribute the parcel to the hosts in your cluster, and then activate the parcel.



For more information, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_install_parcel.html

Installing Apache Spark 2 on YARN

To download and install Apache Spark 2 on YARN, complete the following steps:



To install CDS Powered by Apache Spark, refer to:
https://www.cloudera.com/documentation/spark2/latest/topics/spark2_installing.html.

1. Download these files from this link
https://www.cloudera.com/documentation/spark2/latest/topics/spark2_packaging.html#versions and place them under /opt/cloudera/csd:

`manifest.json`

`SPARK2-2.2.0.cloudera2-1.cdh5.12.0.p0.232957-e17.parcel.shal`

`SPARK2-2.2.0.cloudera2-1.cdh5.12.0.p0.232957-e17.parcel`

SPARK2_ON_YARN-2.2.0.cloudera2.jar

SPARK2_ON_YARN-2.2.0.cloudera2.jar

2. Set the file ownership to cloudera-scm:cloudera-scm with permission 644

```
chown cloudera-scm:cloudera-scm SPARK2_ON_YARN-2.2.0.cloudera2.jar
```

```
chmod 644 SPARK2_ON_YARN-2.2.0.cloudera2.jar
```

3. Copy all files to spark2 folder and move them to /var/www/html/:

```
Service clouderar-scm-server restart
```

4. Click the Parcel tab > host add configuration link then download > add service and download and distribute.

5. Restart the Cloudera Management service.

Select Clusters > Cloudera Management Service.

Select Actions > Restart.

6. Add new hosts for CDSW and add the repository for CDH, CDSW and Spark2:

- a. Hosts > Now add new host to the cluster and add cdsw

- b. Hosts > Now add new host to the cluster and add Spark2

Parcel Settings

Parcel Directory	/opt/cloudera/parcels	The directory that parcels will be installed into on the host(s). The download 'parcel-cache' directory will be placed next to this directory on the filesystem. If the 'parcel_dir' variable is set in the Agent's config.ini file, it will override this value.
Local Parcel Repository Path		/opt/cloudera/parcel-repo
Parcel Update Frequency	1 hour(s)	Path to the local package parcel repository from which binaries are served to the Agents.
Remote Parcel Repository URLs	<ul style="list-style-type: none"> http://10.13.1.50/CDHParcels/ http://10.13.1.50/CDSW/ http://10.13.1.50/spark2/ 	How often to check local and remote parcel repositories for new parcels and if any old parcels should be cleaned up.
Create System-Wide Symlinks for Active Parcels	<input checked="" type="checkbox"/>	URLs of the remote parcel repositories where Cloudera Manager checks for new parcels. When checking for new parcels, Cloudera Manager sends the ID of the server and the server version to the repository host. The special variable {latest_supported} is replaced with the latest version of CDH that Cloudera Manager supports when checks are made.
		Whether system-wide symlinks should be created for the active parcels (for example, /usr/bin/hadoop).
Reason for change...		Save Changes

The repository is being distributed and activated.

cloudera MANAGER Clusters ▾ Hosts ▾ Diagnostics ▾ Audits Charts ▾ Backup ▾ Administration ▾ Search Support ▾ admin ▾

Parcels

Location Cluster 1

Parcel Name	Version	Status	Actions
CDH 5	5.13.0-1.cdh5.13.0.p0.29	Distributed, Activated	
CDSW	1.3.0.p1.244221	Distributed, Activated	
SPARK2	2.2.0.cloudera2-1.cdh5.12.0.p0.232957	Available Remotely	

Filters

PARCEL NAME

CDH 5 1

cloudera MANAGER Clusters ▾ Hosts ▾ Diagnostics ▾ Audits Charts ▾ Backup ▾ Administration ▾ Search Support ▾ admin ▾

Parcels

Location Cluster 1

Parcel Name	Version	Status	Actions
CDH 5	5.13.0-1.cdh5.13.0.p0.29	Distributed, Activated	
CDSW	1.3.0.p1.244221	Distributed, Activated	
SPARK2	2.2.0.cloudera2-1.cdh5.12.0.p0.232957	Distributing 0% Details...	

Filters

PARCEL NAME

CDH 5 1

cloudera MANAGER Clusters ▾ Hosts ▾ Diagnostics ▾ Audits Charts ▾ Backup ▾ Administration ▾ Search Support ▾ admin ▾

Parcels

Location Cluster 1

Parcel Name	Version	Status	Actions
CDH 5	5.13.0-1.cdh5.13.0.p0.29	Distributed, Activated	
CDSW	1.3.0.p1.244221	Distributed, Activated	
SPARK2	2.2.0.cloudera2-1.cdh5.12.0.p0.232957	Activating 0% Details...	

Filters

PARCEL NAME

CDH 5 1

- Add the Spark 2 service to the cluster and deploy Spark 2 Gateway roles to the CDSW hosts, which are going to deploy Spark Job through CDSW.



Note: CDSW need Spark 2 to deploy Spark Jobs.

cloudera® MANAGER Clusters ▾ Hosts ▾ Di

Home

Status All Health Issues 0 1 Configuration 5 ▾ All Recent C

Cluster 1 (CDH 5.13.0, Parcels)

- Hosts**
- HBase**
- HDFS**

Add Service

Start

Sentry Sentry service stores authorization policy metadata and provides clients concurrent and secure access to this metadata.

Solr Solr is a distributed service for indexing and searching data stored in HDFS.

Spark Apache Spark is an open source cluster computing system. This service runs Spark as an application on YARN.

Spark (Standalone) Apache Spark is an open source cluster computing system. This is the standalone version of the service which does not use YARN for resource management. Cloudera recommends using Spark on YARN instead of this standalone version.

Spark 2 Apache Spark is an open source cluster computing system. This service runs Spark 2 as an application on YARN. **Before adding this service, ensure that you have installed the Spark2 binaries, which are not included in CDH.**

Sqoop 1 Client Configuration and connector management for Sqoop 1.

Sqoop 2 Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases. The version supported by Cloudera is Sqoop 2.

Back **Continue**

Add Spark 2 Service to Cluster 1

Select the set of dependencies for your new Spark 2

HBase	HDFS	Hive	Spark	YARN (MR2 Included)	ZooKeeper
<input type="radio"/>	HDFS			YARN (MR2 Included)	ZooKeeper
<input checked="" type="radio"/> HBase	HDFS	Hive	Spark	YARN (MR2 Included)	ZooKeeper

cloudera MANAGER Clusters ▾ Hosts ▾ Diagnostics ▾ Audits Charts ▾ Backup ▾ Administration ▾

cds Actions ▾

Status Configuration Processes Resources Components Commands Charts Library Audits Quick Links ▾

Filters Clear All

CATEGORY Clear

- Advanced 3
- Monitoring 24
- Parcels 1
- Resource Management 2

STATUS

- Error 0
- Warning 0

Search

P2P Parcel Distribution Port 7191

Java Home Directory /usr/java/jdk1.8.0_91

Cloudera Manager Agent
Monitoring Advanced
Configuration Snippet (Safety Valve)

Add Spark 2 Service to Cluster 1

First Run Command

Status **Finished** Mar 15, 11:34:30 AM 48.19s

Finished First Run of the following services successfully: Spark 2-2.

Completed 4 of 4 step(s).

Show All Steps Show Only Failed Steps Show Only Running Steps

> Ensuring that the expected software releases are installed on hosts. Successfully completed 1 steps.	Mar 15, 11:34:30 AM	44ms
> Deploying Client Configuration Successfully deployed all client configurations.	Mar 15, 11:34:30 AM	17.61s
> SeqCmdWork of 2 steps in sequence Successfully executed command Create Spark History Log Dir on service Spark 2-2	Mar 15, 11:34:48 AM	7.96s
> Start Spark 2 Successfully started service.	Mar 15, 11:34:56 AM	22.26s

Add the Cloudera Data Science Workbench Service

To install CDSW, complete the following steps:



For more information, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_install_parcel.html.

1. Log into the Cloudera Manager Admin Console.
2. On the **Home > Status** tab, right-click the cluster name and select **Add a Service** to launch the wizard. A list of services will be displayed.

Home

Status All Health Issues Configuration 🔧 5 ▾ All Recent C

Cluster 1 (CDH 5.13.0, Parcels)

- Hosts
- HBase
- HDFS

Add Service

Start

Stop

- Select the Cloudera Data Science Workbench service and click Continue.

Add Service to Cluster 1

Select the type of service you want to add.

Service Type	Description
<input type="radio"/> Accumulo	The Apache Accumulo sorted, distributed key/value store is a robust, scalable, high performance data storage and retrieval system. This service only works with releases based on Apache Accumulo 1.6 or later.
<input checked="" type="radio"/> Cloudera Data Science Workbench	Cloudera Data Science Workbench enables fast, easy, and secure self-service data science for the enterprise.

Add Cloudera Data Science Workbench Service to Cluster 1

Select the set of dependencies for your new Cloudera Data Science Workbench

HBase	HDFS	Hive	Solr	Spark 2	Spark	YARN (MR2 Included)	ZooKeeper
<input type="radio"/> HBase	HDFS	Hive		Spark 2-2	Spark	YARN (MR2 Included)	ZooKeeper
<input checked="" type="radio"/> HBase	HDFS	Hive	Solr	Spark 2-2	Spark	YARN (MR2 Included)	ZooKeeper

- Assign the Master and Worker roles to the gateway hosts. You must assign the Cloudera Data Science Workbench Master role to one gateway host, and optionally, assign the Worker role to one or more gateway hosts.



Note: Do not assign Masters and Workers to the same host; even on single node deployments, the Master can perform the functions of a Worker as needed.

Add Cloudera Data Science Workbench Service to Cluster 1

Customize Role Assignments for Cloudera Data Science Workbench

You can customize the role assignments for your new service here, but note that if assignments are made incorrectly, such as assigning too many roles to a single host, performance will suffer.

You can also view the role assignments by host.

[View By Host](#)



5. Configure the following parameters and click Continue



Note: For more information about the parameters, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_install_parcel.html.

Add Cloudera Data Science Workbench Service to Cluster 1

Review Changes

Cloudera Data Science Workbench Domain DOMAIN	Cloudera Data Science Workbench (Service-Wide) ↵ <input type="text" value="cdsw.cisco.com"/>	Wildcard DNS domain configured to point to the master node. If the wildcard DNS entries are configured as cdsw.your_domain.com and *.cdsw.your_domain.com, then this property should be set to cdsw.your_domain.com. For more information on setting up a wildcard DNS subdomain, see the Cloudera Data Science Workbench Installation Guide .
 		✖
Master Node IPv4 Address MASTER_IP	Cloudera Data Science Workbench (Service-Wide) ↵ <input type="text" value="10.13.1.250"/>	?
 		✖
Install Required Packages cdsw.install.required.packages	<input checked="" type="checkbox"/> Cloudera Data Science Workbench (Service-Wide)	?
 		✖
Docker Block Device DOCKER_BLOCK_DEVICES	Docker Daemon Default Group ↵ <input type="text" value="/dev/sdb"/>	Cloudera Data Science Workbench installer will format and mount Docker on each gateway host. Do not mount these block devices prior to installation. Every Cloudera Data Science Workbench gateway host must have one or more block devices with at least 500 GB dedicated to storage of Docker images. SSDs are strongly recommended. Use the full path, for instance "/dev/xvde".
 		✖

[Back](#)
[Continue](#)

1
2
3
4
5

Add Cloudera Data Science Workbench Service to Cluster 1

First Run Command

Status Finished Mar 15, 11:58:01 AM 40.57s

Finished First Run of the following services successfully: Cloudera Data Science Workbench.

▼ Completed 4 of 4 step(s).

Show All Steps Show Only Failed Steps Show Only Running Steps

➤ ✓ Ensuring that the expected software releases are installed on hosts. Successfully completed 1 steps.	Mar 15, 11:58:01 AM	70ms
➤ ✓ Deploying Client Configuration 🔗 Cluster 1 🔗 Successfully deployed all client configurations.	Mar 15, 11:58:01 AM	2.25s
➤ ✓ Execute command Prepare Node on service Cloudera Data Science Workbench Command Prepare Node finished successfully on service cds.	Mar 15, 11:58:03 AM	16.24s
➤ ✓ Start Cloudera Data Science Workbench 🔗 Cloudera Data Science Workbench 🔗 Successfully started service.	Mar 15, 11:58:20 AM	21.49s

Back

1 2 3 4 5

Continue

6. The wizard will now begin a First Run of the Cloudera Data Science Workbench service. This includes deploying client configuration for HDFS, YARN and Spark 2, installing the package dependencies on all hosts, and formatting the Docker block device. The wizard will also assign the Application role to the host running Master, and the Docker Daemon role to all the Cloudera Data Science Workbench gateway hosts.
7. Once the First Run command has completed successfully, click Finish to go back to the Cloudera Manager home page.



Note: CDSW will take 10-15 minutes to come online for the first time; they can follow the Docker/Master/Application process logs to track progress.

The screenshot shows the Cloudera Manager Home page. At the top, there are navigation links: Clusters, Hosts, Diagnostics, Audits, Charts, and Backup. Below the navigation is a breadcrumb trail: Status > All Health Issues > Configuration > All Recent Commands. On the left, a sidebar lists various cluster components under Cluster 1 (CDH 5.13.0, Parcels), each with a dropdown arrow: Hosts, Cloudera D..., HBase, HDFS, Hive, Hue (with a warning icon), Impala, Key-Value ..., Oozie, Solr, Spark, Spark 2-2, YARN (MR2...), and ZooKeeper. To the right, there are two charts. The first chart, titled 'Cluster CPU', shows a horizontal line at 0.28% usage from 01:15 to 01:30. The second chart, titled 'Cluster Network IO', shows a line graph of bytes/second from 01:15 to 01:30, with two series: Total Bytes Re... at 26.9K/s and Total Bytes Tr... at 28.9K/s.

Create the Administrator Account

After your installation is complete, set up the initial administrator account. Go to the Cloudera Data Science Workbench web application at http://cdsw.<your_domain>.com.

You must access Cloudera Data Science Workbench from the Cloudera Data Science Workbench Domain configured when setting up the service, and not the hostname of the master node. Visiting the hostname of the master node will result in a 404 error.

The first account that you create becomes the site administrator. You may now use this account to create a new project and start using the workbench to run data science workloads.

Non-Kerberized Clusters



Note: In this CVD, we did not enable Kerberos.



Note: To enable Kerberos, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_kerberos.html.

1. To disable Kerberos, delete the file /etc/krb5.conf on the CDSW nodes.
2. For a non-kerberized cluster, by default, your Hadoop username will be set to your Cloudera Data Science Workbench login username. Override this default and set an alternative HADOOP_USER_NAME ("admin" in this case which was the admin user created).
3. Create User admin in hdfs.

```
hdfs dfs -mkdir /user/admin
hdfs dfs -chwon admin:admin /user/admin
```

```
[root@rhel1 ~]# hdfs dfs -ls /user
Found 7 items
drwxr-xr-x  - admin  admin          0 2018-03-19 15:47 /user/admin
drwxrwxrwx  - mapred hadoop        0 2018-03-14 14:27 /user/history
drwxrwxr-t  - hive   hive          0 2018-03-14 14:28 /user/hive
drwxrwxr-x  - hue    hue           0 2018-03-14 14:28 /user/hue
drwxrwxr-x  - impala impala        0 2018-03-14 14:28 /user/impala
drwxrwxr-x  - oozie  oozie          0 2018-03-14 14:28 /user/oozie
drwxr-x--x  - spark  spark          0 2018-03-15 11:34 /user/spark
```

4. Go to the Hadoop Username Override setting at Account settings > Hadoop Authentication.

User Settings

[Profile](#) [SSH Keys](#) [Hadoop Authentication](#) [API Keys](#)

Hadoop Username Override

Set the `HADOOP_USER_NAME` environmental variable to override your Hadoop username for all sessions and jobs.

admin

[Authenticate](#)

5. Restart CDSW from Cloudera Manager.

Using GPUs for Cloudera Data Science Workbench Workloads

A GPU is a specialized processor that can be used to accelerate highly parallelized computationally-intensive workloads. Because of their computational power, GPUs have been found to be particularly well-suited to [deep learning workloads](#). Ideally, CPUs and GPUs should be used in tandem for data engineering and data science workloads. A typical machine learning workflow involves data preparation, model training, model scoring, and model fitting. You can use existing general-purpose CPUs for each stage of the workflow, and optionally accelerate the math-intensive steps with the selective application of special-purpose GPUs. For example, GPUs allow you to accelerate model fitting using frameworks such as [Tensorflow](#), [PyTorch](#), [Keras](#), [MXNet](#), and [Microsoft Cognitive Toolkit \(CNTK\)](#).

By enabling GPU support, data scientists can share GPU resources available on Cloudera Data Science Workbench nodes. Users can request a specific number of GPU instances, up to the total number available on a node, which are then allocated to the running session or job for the duration of the run. Projects can use isolated versions of libraries, and even different CUDA and cuDNN versions via Cloudera Data Science Workbench's extensible engine feature.



Note: For more information, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_gpu.html.

Enabling GPU with CDSW

- Cloudera Data Science Workbench only supports CUDA-enabled NVIDIA GPU cards.
- Cloudera Data Science Workbench does not support heterogeneous GPU hardware in a single deployment.
- Cloudera Data Science Workbench does not include an engine image that supports NVIDIA libraries.

This section provides instructions about creating your own custom CUDA-capable engine image.

To enable Docker containers to use the GPUs, the previously installed NVIDIA driver libraries must be consolidated in a single directory named after the <driver_version>, and mounted into the containers. This is done using the [nvidia-docker](#) package, which is a thin wrapper around the Docker CLI and a Docker plugin.

The following sample steps demonstrate how to use nvidia-docker to set up the directory structure for the drivers so that they can be easily consumed by the Docker containers that will leverage the GPU. Perform these steps on all nodes with GPU hardware installed.

1. Download and install nvidia-docker.

```
wget https://github.com/NVIDIA/nvidia-docker/releases/download/v1.0.1/nvidia-docker-1.0.1-1.x86_64.rpm
yum install nvidia-docker-1.0.1-1.x86_64.rpm
```



Note: This example uses nvidia-docker 1.0.

2. Start the necessary services and plugins:

```
systemctl start nvidia-docker
systemctl enable nvidia-docker
```

3. Run a small container to create the Docker volume structure

```
nvidia-docker run --rm nvidia/cuda nvidia-smi
```

```
[root@cdsw1 ~]# nvidia-docker run --rm nvidia/cuda nvidia-smi
Using default tag: latest
latest: Pulling from nvidia/cuda
22dc81ace0ea: Pull complete
1a8b3c87dba3: Pull complete
91390a1c435a: Pull complete
07844b14977e: Pull complete
b78396653dae: Pull complete
95e837069dfa: Pull complete
fef4aadda783: Pull complete
343234bd5cf3: Pull complete
64e8786fc8c1: Pull complete
d6a4723d353c: Pull complete
Digest: sha256:3524adf9b563c27d9a0f6d0584355c1f4f4b38e90b66289b8f8de026a9162eee
Status: Downloaded newer image for nvidia/cuda:latest
Tue Mar 20 02:00:20 2018
+-----+
| NVIDIA-SMI 390.12          Driver Version: 390.12 |
+-----+
| GPU  Name      Persistence-MI Bus-Id     Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
|=====+=====+=====+=====+=====+=====+=====+=====
|  0  Tesla P100-PCIE... Off  | 00000000:5E:00.0 Off |          0 |
| N/A   30C    P0    26W / 250W |    10MiB / 16280MiB |     0%   Default |
+-----+
|  1  Tesla P100-PCIE... Off  | 00000000:AF:00.0 Off |          0 |
| N/A   29C    P0    24W / 250W |    10MiB / 16280MiB |     0%   Default |
+-----+
+-----+
| Processes:                               GPU Memory |
| GPU     PID  Type  Process name        Usage      |
| =====+=====+=====+=====
| No running processes found
+-----+
[root@cdsw1 ~]#
```

4. Verify that the `/var/lib/nvidia-docker/volumes/nvidia_driver/$NVIDIA_DRIVER_VERSION/` directory was created.
5. Use the following Docker command to verify that Cloudera Data Science Workbench can access the GPU:

```
docker run --net host \
    --device=/dev/nvidiactl \
    --device=/dev/nvidia-uvm \
    --device=/dev/nvidia0 \
    -v /var/lib/nvidia-
    docker/volumes/nvidia_driver/$NVIDIA_DRIVER_VERSION/:/usr/local/nvidia/ \
    -it nvidia/cuda \
    /usr/local/nvidia/bin/nvidia-smi
```

```
[root@cdsw1 ~]# docker run --net host \
>   --device=/dev/nvidiactl \
>   --device=/dev/nvidia-uvm \
>   --device=/dev/nvidia0 \
>   -v /var/lib/nvidia-docker/volumes/nvidia_driver/$NVIDIA_DRIVER_VERSION:/usr/local/nvidia/ \
>   -it nvidia/cuda \
>   /usr/local/nvidia/bin/nvidia-smi
Tue Mar 20 02:01:29 2018
+-----+
| NVIDIA-SMI 390.12          Driver Version: 390.12 |
|-----+
| GPU  Name      Persistence-MI Bus-Id     Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
|-----+
|  0  Tesla P100-PCIE... Off  00000000:5E:00.0 Off |          0 |
| N/A   30C    P0    26W / 250W |    10MiB / 16280MiB |     0%       Default |
+-----+
+-----+
| Processes:                               GPU Memory |
| GPU     PID  Type  Process name        Usage        |
|-----+
| No running processes found               |
+-----+
```



Note: On a multi-GPU machine the output of this command will show exactly one GPU. This is because we have run this sample Docker container with only one device (/dev/nvidia0).

6. Restart CDSW.

Enable GPU Support in Cloudera Data Science Workbench

To enable Cloudera Data Science Workbench to identify the GPUs installed, complete the following steps:

- In Cloudera Manager – Cloudera Data Science Workbench Configuration, set the following properties:
 - Enable GPU Support
 - Complete path to the NVIDIA driver libraries "/var/lib/nvidia-docker/volumes/nvidia_driver/\$NVIDIA_DRIVER_VERSION/"

Enable GPU Support

NVIDIA_GPU_ENABLE

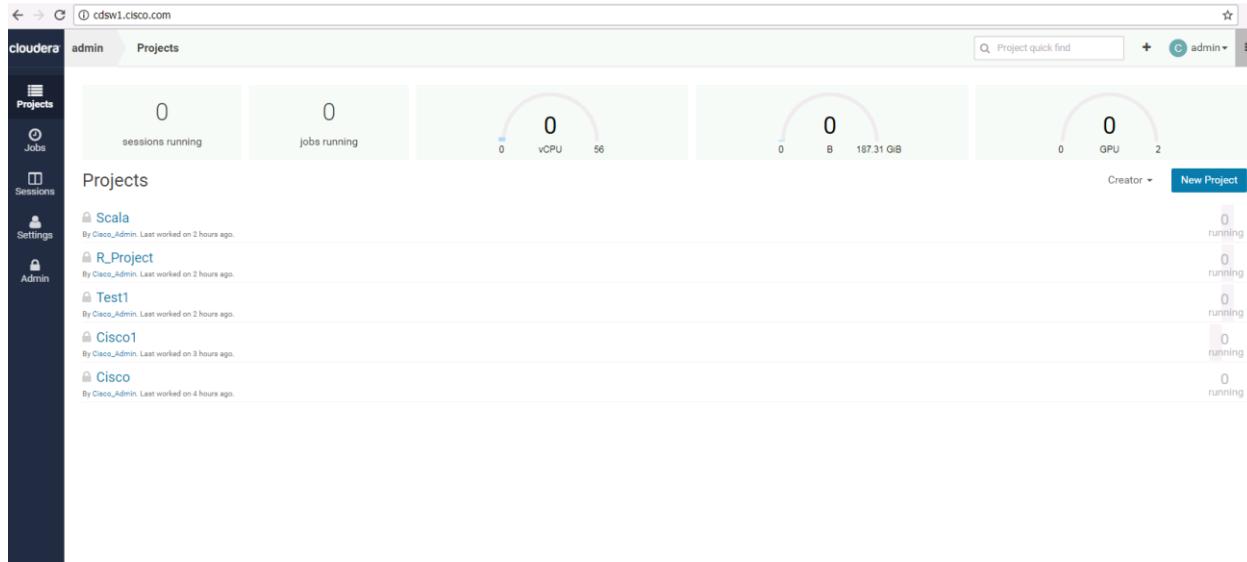
Cloudera Data Science Workbench (Service-Wide)

NVIDIA_LIBRARY_PATH

Cloudera Data Science Workbench (Service-Wide)

/var/lib/nvidia-docker/volumes/nvidia_driver/390.12/

2. Restart CDSW from Cloudera Manager for GPU to display in Cloudera Data Science Workbench:



Create a Custom CUDA-Capable Engine Image

The base engine image (`docker.repository.cloudera.com/cdsw/engine:4`) that ships with Cloudera Data Science Workbench will need to be extended with CUDA libraries to make it possible to use GPUs in jobs and sessions.



For more information, refer to https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_gpu.html.

For this we first create a local registry

Run a Local Registry

To start the registry container, run the following command:

```
$ docker run -d -p 5000:5000 --restart=always --name registry registry:2
```

```
[root@cdsw1 CUDA]# docker run -d -p 5000:5000 --restart=always --name registry registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry
81033e7c1d6a: Pull complete
b235084c2315: Pull complete
c692f3a6894b: Pull complete
ba2177f3a70e: Pull complete
a8d793620947: Pull complete
Digest: sha256:672d519d7fd7bbc7a448d17956ebeefe225d5eb27509d8dc5ce67ecb4a0bce54
Status: Downloaded newer image for registry:2
78a51a347abee4016d3bd92be7f2281f98dee8a1ccd57f68a3c320476d15b46f
[root@cdsw1 CUDA]#
```



For more information about creating a local docker registry, refer to:
<https://docs.docker.com/registry/deploying/#copy-an-image-from-docker-hub-to-your-registry>

The following sample Dockerfile illustrates an engine on top of which machine learning frameworks such as Tensorflow and PyTorch can be used. This Dockerfile uses a deep learning library from NVIDIA called [NVIDIA CUDA Deep Neural Network \(cuDNN\)](#). Make sure you check with the machine learning framework that you intend to use in order to know which version of cuDNN is needed. As an example, Tensorflow 1.4 uses CUDA 8.0 and requires cuDNN 6.0.

To create the cuda.Dockerfile, run the following command:

```
FROM docker.repository.cloudera.com/cdsw/engine:4

RUN
NVIDIA_GPGKEY_SUM=d1be581509378368edeec8c1eb2958702feedf3bc3d17011adbf24efacce4ab5
&& \
    NVIDIA_GPGKEY_FPR=ae09fe4bbd223a84b2ccfce3f60f4b3d7fa2af80 && \
    apt-key adv --fetch-keys
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub && \
    apt-key adv --export --no-emit-version -a $NVIDIA_GPGKEY_FPR | tail -n +5 >
cudasign.pub && \
    echo "$NVIDIA_GPGKEY_SUM cudasign.pub" | sha256sum -c --strict - && rm
cudasign.pub && \
    echo "deb
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64 /" >
/etc/apt/sources.list.d/cuda.list

ENV CUDA_VERSION 8.0.61
LABEL com.nvidia.cuda.version="${CUDA_VERSION}"

ENV CUDA_PKG_VERSION 8-0=${CUDA_VERSION}-1
RUN apt-get update && apt-get install -y --no-install-recommends \
    cuda-nvrtc-$CUDA_PKG_VERSION \
    cuda-nvgraph-$CUDA_PKG_VERSION \
    cuda-cusolver-$CUDA_PKG_VERSION \
    cuda-cublas-8-0=8.0.61.2-1 \
    cuda-cufft-$CUDA_PKG_VERSION \
    cuda-curand-$CUDA_PKG_VERSION \
    cuda-cusparse-$CUDA_PKG_VERSION \
    cuda-npp-$CUDA_PKG_VERSION \
    cuda-cudart-$CUDA_PKG_VERSION && \
    ln -s cuda-8.0 /usr/local/cuda && \
    rm -rf /var/lib/apt/lists/*

RUN echo "/usr/local/cuda/lib64" >> /etc/ld.so.conf.d/cuda.conf && \
    ldconfig

RUN echo "/usr/local/nvidia/lib" >> /etc/ld.so.conf.d/nvidia.conf && \
    echo "/usr/local/nvidia/lib64" >> /etc/ld.so.conf.d/nvidia.conf

ENV PATH /usr/local/nvidia/bin:/usr/local/cuda/bin:${PATH}
ENV LD_LIBRARY_PATH /usr/local/nvidia/lib:/usr/local/nvidia/lib64

RUN echo "deb http://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1604/x86_64 /" > /etc/apt/sources.list.d/nvidia-ml.list

ENV CUDNN_VERSION 6.0.21
LABEL com.nvidia.cudnn.version="${CUDNN_VERSION}"

RUN apt-get update && apt-get install -y --no-install-recommends \
    libcudnn6=$CUDNN_VERSION-1+cuda8.0 && \
    rm -rf /var/lib/apt/lists/*
```

Build the [custom engine image](#) out of cuda.Dockerfile using the following command:

```
#docker build --network host -t localhost/cisco-gpu-demo:2 . -f cuda.Dockerfile
```

```
[root@cdsw1 CUDA]# docker build --network host -t localhost/cisco-gpu-demo:2 . -f cuda.Dockerfile
Sending build context to Docker daemon 2.625 GB
Step 1/14 : FROM docker.repository.cloudera.com/cdsw/engine:4
--> 62a5f1fa8baa
Step 2/14 : RUN NVIDIA_GPGKEY_SUM=d1be581509378368edeec8c1eb2958702feedf3bc3d17011adbf24efacce4ab5 &&      NVIDIA_GPK
developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub &&      apt-key adv --export --no-emit
SUM cudasign.pub" | sha256sum -c --strict - && rm cudasign.pub &&      echo "deb http://developer.download.nvidia.com
--> Running in 64be3bc174f3
Executing: /tmp/tmp.dGB5d7KKJ0/gpg.1.sh --fetch-keys
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub
gpg: key 7FA2AF80: public key "cuda tools <cuda-tools@nvidia.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)
cudasign.pub: OK
--> 5385d841fc98
Removing intermediate container 64be3bc174f3
Step 3/14 : ENV CUDA_VERSION 8.0.61
--> Running in 9e8bfc8856a6
--> 2c7cf0cbba27
Removing intermediate container 9e8bfc8856a6
Step 4/14 : LABEL com.nvidia.cuda.version "${CUDA_VERSION}"
--> Running in 9a55e251f33f
--> 68809416d0d7
Removing intermediate container 9a55e251f33f
Step 5/14 : ENV CUDA_PKG_VERSION 8-0=${CUDA_VERSION}-1
--> Running in 33e22995b2b7
--> aa4fe053d6a7
```

```
# docker images
```

```
[root@cdsw1 CUDA]# docker images
REPOSITORY          TAG        IMAGE ID       CREATED        SIZE
localhost/cisco-gpu-demo   2          3716cf330bb8    13 seconds ago  7.01 GB
nvidia/cuda          latest     3fd923127acb    11 days ago   2.23 GB
docker.repository.cloudera.com/cdsw/1.3.0/spark-port-forwarder 9bb84f6    773e3faae30e    8 weeks ago   33.7 MB
docker.repository.cloudera.com/cdsw/1.3.0/kinit           9bb84f6    d1e855f63c70    8 weeks ago   5.53 MB
docker.repository.cloudera.com/cdsw/1.3.0/reconciler      9bb84f6    e776910dbdf3    8 weeks ago   4.16 MB
docker.repository.cloudera.com/cdsw/1.3.0/engine-deps      9bb84f6    d4a46cc6d98c8    8 weeks ago   30.2 MB
docker.repository.cloudera.com/cdsw/1.3.0/web            9bb84f6    4276fe9b0c0      8 weeks ago   94.1 MB
docker.repository.cloudera.com/cdsw/1.3.0/cron           9bb84f6    20e3f1aed0c4    8 weeks ago   2.12 GB
docker.repository.cloudera.com/cdsw/1.3.0/chronic         9bb84f6    43de51cb4ef3    8 weeks ago   6.23 MB
docker.repository.cloudera.com/cdsw/1.3.0/livelog         9bb84f6    431954f734508   8 weeks ago   287 MB
registry              2          d1fd7d86a825    2 months ago   33.3 MB
nvidia/cuda          8.0        a60a3c83aeed    2 months ago   0 B
docker.repository.cloudera.com/cdsw/engine               4          62a5f1fa8baa    3 months ago   6.19 GB
gcr.io/google_containers/kube-apiserver-amd64          v1.6.11    b183f1acd036    5 months ago   151 MB
gcr.io/google_containers/kube-controller-manager-amd64  v1.6.11    45a72a7fe4da    5 months ago   133 MB
gcr.io/google_containers/kube-scheduler-amd64          v1.6.11    1f58e513ebbc    5 months ago   76.8 MB
gcr.io/google_containers/kube-proxy-amd64             v1.6.11    b6ca9542717d    5 months ago   109 MB
gcr.io/google_containers/k8s-dns-sidecar-amd64          1.14.5     fed89e8b4248   5 months ago   41.8 MB
gcr.io/google_containers/k8s-dns-kube-dns-amd64          1.14.5     512cd7425a73    5 months ago   49.4 MB
gcr.io/google_containers/k8s-dns-dnsmasq-nanny-amd64    1.14.5     459944ce8cc4   5 months ago   41.4 MB
docker.repository.cloudera.com/cdsw/1.3.0/third-party/postgres 9.3       817c286a6a50    6 months ago   262 MB
docker.repository.cloudera.com/cdsw/1.3.0/third-party/weave-npc 1.9.4     e97ebd34f5e3    12 months ago  58.2 MB
docker.repository.cloudera.com/cdsw/1.3.0/third-party/weave-kube 1.9.4     77be5bc6f826    12 months ago  163 MB
docker.repository.cloudera.com/cdsw/1.3.0/third-party/weaveexec 1.9.4     2ddf2a46788c    12 months ago  91 MB
docker.repository.cloudera.com/cdsw/1.3.0/third-party/traefik   v1.1.2-alpine  f97a0e82f469   12 months ago  21.9 MB
gcr.io/google_containers/etcd-amd64                     3.0.17    243830dae7dd   13 months ago  169 MB
docker.repository.cloudera.com/cdsw/1.3.0/third-party/node-problem-detector v0.1      0e9556b82579   21 months ago  40.1 MB
gcr.io/google_containers/pause-amd64                    3.0       99e59f495ffa   22 months ago  747 kB
```

Tag the image as `localhost:5000/<image name>`. This creates an additional tag for the existing image. When the first part of the tag is a hostname and port, Docker interprets this as the location of a registry, when pushing.

```
[root@cdsw1 CUDA]# docker tag localhost/cisco-gpu-demo:2 localhost:5000/cisco-gpu-demo
```

Push the image to the local registry running at `localhost:5000`:

```
$ docker push localhost:5000/cisco-gpu-demo
```

```
182d2a55830d: Pushed
5a4c2c9a24fc: Pushed
cb11ba605400: Pushed
latest: digest: sha256:69629b3ed9f0fb852f76a7e840a451bb8032a702c206e17ed1952d52c306ff23 size: 19358
[root@cdsw1 CUDA]#
```

Allocate GPUs for Sessions and Jobs

Once Cloudera Data Science Workbench has been enabled to use GPUs, a site administrator must whitelist the CUDA-capable engine image created in the previous step. Site administrators can also set a limit on the maximum number of GPUs that can be allocated per session or job.

1. Sign in to Cloudera Data Science Workbench as a site administrator.
2. Click **Admin**.
3. Go to the **Engines** tab.
4. From the **Maximum GPUs per Session/Job drop-down** list, select the maximum number of GPUs that can be used by an engine.

The screenshot shows the 'Engines' tab selected in the navigation bar. Under 'Engines Profiles', there is a table with three rows:

Description	vCPU (burstable)	Memory (GiB)	Actions
1 vCPU / 2 GiB Memory	1	2	Edit Delete
8 vCPU / 8 GiB Memory	8	8	Edit Delete
1 vCPU (burstable), 1.75 GiB memory	1	1.75	Add

Below this is a dropdown labeled 'Maximum GPUs per Session/Job' set to '2'. A note explains: 'vCPU is expressed in fractional virtual cores and allows bursting. Memory is expressed in fractional GiB and is enforced by memory killer. GPU indicates the number of GPUs that need to be used by the engine. Configurations larger than the maximum allocatable CPU, memory and GPU per node will be unschedulable.'

5. Under **Engine Images**, add the custom CUDA-capable engine image created in the previous step. This whitelists the image and allows project administrators to use the engine in their jobs and sessions.

Engine Images

Description	Repository:Tag	Default	Actions
Base Image v4	docker.repository.cloudera.com/cdsw/engine:4	<input checked="" type="radio"/>	Edit Deprecate
GPU Cisco	localhost:5000/cisco-gpu-demo	<input type="radio"/>	Edit Deprecate

Whitelist Docker images for project owners to use in their jobs and sessions. These must be public images in registries that are accessible from the Cloudera Data Science Workbench hosts.

6. Click **Update**.

Project administrators can now whitelist the CUDA engine image to make it available for sessions and jobs within a particular project by completing the following steps:

1. Go to Projects Overview
2. Click on any **projects** and choose **Settings**
3. Go to the **Engines** tab in the above page
4. Under **Engine Image**, add the CUDA-capable engine image.

The screenshot shows the 'Project Settings' page for a project named 'admin'. The left sidebar has icons for Overview, Jobs, Sessions, Files, Team, and Settings, with 'Settings' currently selected. The top navigation bar shows 'Not secure | cdsw1.cisco.com/admin/gpu1/settings/engine'. The main content area has tabs for Options, Engine (which is selected), Tunnels, and Delete Project. The 'Engine Image' section contains a dropdown menu with the value 'GPU Image, localhost:5000/cdsw-cuda'. Below this is the 'Environmental Variables' section, which includes a table for adding variables, a note about visibility, and a 'Save Environment' button. The 'Security' section at the bottom explains environmental variable values.

Name	Value	Actions
<input type="text"/>	<input type="text"/>	<button>Add</button>

Press tab or enter to add another.

Save Environment

Security

Environmental variable **values** are only visible to collaborators with **write** or higher access. They are a great way to securely store confidential information such as your AWS or database credentials. Names are available to all users with access to the project.

Bill of Materials

This section provides the BOM for the 24 Nodes Hadoop Base Rack and 8 Nodes Hadoop Expansion Rack. See Table 9 for BOM for the Hadoop Base rack, Table 10 for BOM for Hadoop Expansion Rack, 11 for BOM for Hadoop GPU Rack, Table 12 and Table 13 and Table 15 for software components. Table 14 lists Cloudera SKUs available from Cisco.



Note: If UCS-SP-CPA4-P2 is added to the BOM all the required components for 16 servers only are automatically added. If not customers can pick each of the individual components that are specified after this and build the BOM manually.

Table 9 Bill of Materials for C240M5SX Hadoop Nodes Base Rack

Part Number	Description	Qty
UCS-SP-C240M5-A2	SP C240 M5SX w/2x6132,6x32GB mem,VIC1387	24
CON-OSP-C240M5A2	SNTC 24X7X4OS UCS C240 M5 A2	24
UCS-CPU-6132	2.6 GHz 6132/140W 14C/19.25MB Cache/DDR4 2666MHz	48
UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	144
UCSC-PCI-1-C240M5	Riser 1 including 3 PCIe slots (x8, x16, x8); slot 3 required CPU2	24
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	24
UCSC-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	48
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	48
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 & C240 M4 & M5 rack servers	24
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	24
UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs & below	48
UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	624
UCSC-PCIF-240M5	C240 M5 PCIe Riser Blanking Panel	24
CBL-SC-MR12GM5P	Super Cap cable for UCSC-RAID-M5HD	24
UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	24
UCSC-RAID-M5HD	Cisco 12G Modular RAID controller with 4GB cache	24
UCS-SP-FI6332-2X	UCS SP Select 2 x 6332 FI	1
UCS-SP-FI6332	(Not sold standalone) UCS 6332 1RU FI/12 QSFP+	2
CON-OSP-SPFI6332	ONSITE 24X7X4 (Not sold standalone) UCS 6332 1RU FI/No PSU/3	2
UCS-PSU-6332-AC	UCS 6332 Power Supply/100-240VAC	4
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	4
QSFP-H40G-CU3M	40GBASE-CR4 Passive Copper Cable, 3m	16
QSFP-40G-SR-BD	QSFP40G BiDi Short-reach Transceiver	8
N10-MGT015	UCS Manager v3.2(1)	2
UCS-ACC-6332	UCS 6332 Chassis Accessory Kit	2
UCS-FAN-6332	UCS 6332 Fan Module	8
QSFP-H40G-CU3M=	40GBASE-CR4 Passive Copper Cable, 3m	48

Part Number	Description	Qty
UCS-SP-H1P8TB-4X	UCS SP 1.8 TB 12G SAS 10K RPM SFF HDD (4K) 4Pk	96
UCS-SP-H1P8TB	1.8 TB 12G SAS 10K RPM SFF HDD (4K)	384
UCS-SP-HD-1P8T-2	1.8TB 12G SAS 10K RPM SFF HDD (4K) 2 Pack	16
UCS-SP-HD-1P8T	SP 1.8TB 12G SAS 10K RPM SFF HDD (4K)	32

Table 10 Bill of Materials for Hadoop Nodes Expansion Rack

Part Number	Description	Qty
UCS-SP-C240M5-A2	SP C240 M5SX w/2x6132,6x32GB mem,VIC1387	8
CON-OSP-C240M5A2	SNTC 24X7X4OS UCS C240 M5 A2	8
UCS-CPU-6132	2.6 GHz 6132/140W 14C/19.25MB Cache/DDR4 2666MHz	16
UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	48
UCSC-PCI-1-C240M5	Riser 1 including 3 PCIe slots (x8, x16, x8); slot 3 required CPU2	8
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	8
UCSC-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	16
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	16
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 & C240 M4 & M5 rack servers	8
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	8
UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs and below	16
UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	208
UCSC-PCIF-240M5	C240 M5 PCIe Riser Blanking Panel	8
CBL-SC-MR12GM5P	Super Cap cable for UCSC-RAID-M5HD	8
UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	8
UCSC-RAID-M5HD	Cisco 12G Modular RAID controller with 4GB cache	8
UCS-SP-H1P8TB-4X	UCS SP 1.8 TB 12G SAS 10K RPM SFF HDD (4K) 4Pk	48
UCS-SP-H1P8TB	1.8 TB 12G SAS 10K RPM SFF HDD (4K)	192
UCS-SP-HD-1P8T-2	1.8TB 12G SAS 10K RPM SFF HDD (4K) 2 Pack	8
UCS-SP-HD-1P8T	SP 1.8TB 12G SAS 10K RPM SFF HDD (4K)	16

Table 11 Bill of Materials for CDSW Nodes Rack

Part Number	Description	Qty
UCS-SP-C240M5-A2	SP C240 M5SX w/2x6132,6x32GB mem,VIC1387	4
CON-OSP-C240M5A2	SNTC 24X7X4OS UCS C240 M5 A2	4
UCS-CPU-6132	2.6 GHz 6132/140W 14C/19.25MB Cache/DDR4 2666MHz	8
UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	24
UCSC-PCI-1-C240M5	Riser 1 including 3 PCIe slots (x8, x16, x8); slot 3 required CPU2	4
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	4
UCSC-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	8
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	8
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220, C240 M4 and M5 rack serv-	4

	ers	
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	4
UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs & below	8
UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	104
CBL-SC-MR12GM5P	Super Cap cable for UCSC-RAID-M5HD	4
UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	4
UCSC-RAID-M5HD	Cisco 12G Modular RAID controller with 4GB cache	4
UCSC-PCI-2A-240M5	Riser 2A including 3 PCIe slots (x8, x16, x16) supports GPU	4
UCS-SP-SD-1P6TB-4X	UCS SP 1.6TB 2.5 inch Ent. Perf 12G SAS SSD(10Xendurance)4Pk	4
UCS-SP-SD-1P6TB	1.6TB 2.5 inch Ent. Performance 12G SAS SSD(10X endurance)	16
UCSC-GPU-P100-16G=	NVIDIA P100 16GB	8
CON-OSP-UCSCM16G	SNTC-24X7X4OS NVIDIA P100 16GB	8
UCS-P100CBL-240M5=	C240M5 NVIDIA P100 / V100 Cable	8

Table 12 Red Hat Enterprise Linux License

Red Hat Enterprise Linux		
RHEL-2S2V-3A	Red Hat Enterprise Linux	28
CON-ISV1-EL2S2V3A	3 year Support for Red Hat Enterprise Linux	28

Table 13 Cloudera Data Science Work Bench Software

UCS-BD-CDSWB=	UCS-BD-CDSWB-1Y	Cloudera Data Science Work Bench, 10-user pack - 1 Year
UCS-BD-CDSWB=	UCS-BD-CDSWB-2Y	Cloudera Data Science Work Bench, 10-user pack - 2 Year
UCS-BD-CDSWB=	UCS-BD-CDSWB-3Y	Cloudera Data Science Work Bench, 10-user pack - 3 Year

Table 14 Cloudera SKU's Available at Cisco

Cisco TOP SKU	Cisco PID with Duration	Product Name
UCS-BD-CEBN-BZ=	UCS-BD-CEBN-BZ-3Y	Cloudera Enterprise Basic Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEBN-BZI=	UCS-BD-CEBN-BZI-3Y	Cloudera Enterprise Basic Edition + Indemnification, Node License, Bronze Support - 3 Year
UCS-BD-CEBN-GD=	UCS-BD-CEBN-GD-3Y	Cloudera Enterprise Basic Edition, Node License, Gold Support - 3 Year
UCS-BD-CEBN-GDI=	UCS-BD-CEBN-GDI-3Y	Cloudera Enterprise Basic Edition + Indemnification, Node License, Gold Support - 3 Year
UCS-BD-CEDEN-BZ=	UCS-BD-CEDEN-BZ-3Y	Cloudera Enterprise Data Engineering Edition, Node License, Bronze Support - 3 Year

Cisco TOP SKU	Cisco PID with Duration	Product Name
UCS-BD-CEDEN-GD=	UCS-BD-CEDEN-GD-3Y	Cloudera Enterprise Data Engineering Edition, Node License, Gold Support - 3 Year
UCS-BD-CEODN-BZ=	UCS-BD-CEODN-BZ-3Y	Cloudera Enterprise Operational Database Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEODN-GD=	UCS-BD-CEODN-GD-2Y	Cloudera Enterprise Operational Database Edition, Node License, Gold Support - 2 Year
UCS-BD-CEODN-GD=	UCS-BD-CEODN-GD-3Y	Cloudera Enterprise Operational Database Edition, Node License, Gold Support - 3 Year
UCS-BD-CEADN-BZ=	UCS-BD-CEADN-BZ-3Y	Cloudera Enterprise Analytical Database Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEADN-GD=	UCS-BD-CEADN-GD-3Y	Cloudera Enterprise Analytical Database Edition, Node License, Gold Support - 3 Year
UCS-BD-CEDHN-BZ=	UCS-BD-CEDHN-BZ-3Y	Cloudera Enterprise Data Hub Edition, Node License, Bronze Support - 3 Year
UCS-BD-CEDHN-GD=	UCS-BD-CEDHN-GD-3Y	Cloudera Enterprise Data Hub Edition, Node License, Gold Support - 3 Year

About the Authors

Karthik Kulkarni, Architect, Computing Systems Product Group, Cisco Systems, Inc.

Karthik Kulkarni is an Architect with the Computing Systems Product Group. His focus includes Big Data and analytics system, next gen data center architecture and performance.

Manan Trivedi, Big Data Solutions Architect, Cisco Systems, Inc.

Manan Trivedi is a Big Data Solutions Architect at Computing Systems Product Group. He is part of the solution engineering team focusing on big data infrastructure, solutions, and performance.

Acknowledgements

The authors would like to thank the following for their support and contribution to the design, creating, and validation of this Cisco Validated Design:

- Rajesh Shroff, Big Data Solutions Architect, Data Center Solutions Group, Cisco Systems, Inc.
- Shane Handy, Big Data Solutions Engineer, Data Center Solutions Group, Cisco Systems, Inc.