

Cisco Data Intelligence Platform with Hortonworks Data Platform 3.1 and Cloudera Data Science Workbench 1.5

Deployment Guide for the Cisco Data Intelligence Platform with Hortonworks Data Platform 3.1.0

Last Updated: October 18, 2019



About the Cisco Validated Design Program

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

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

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

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

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

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

© 2019 Cisco Systems, Inc. All rights reserved.

Table of Contents

Executive Summary	8
Solution Overview	9
Cisco Data Intelligence Platform	9
Audience.....	11
Purpose of this Document.....	11
What's New in this Release?	11
What's Next?	11
Reference Architecture	12
Data Lake Reference Architecture.....	12
AI Computing Farm Reference Architecture	13
Scaling the Solution.....	16
Scaled Architecture with 3:1 Oversubscription with Cisco Fabric Interconnects and Cisco ACI	17
Scaled Architecture with 2:1 Oversubscription with Cisco ACI	18
Technology Overview.....	20
Cisco UCS Integrated Infrastructure for Big Data and Analytics	20
Cisco Unified Computing System.....	20
Cisco UCS 6300 Series Fabric Interconnects	20
Cisco UCS C-Series Rack-Mount Servers.....	20
Cisco UCS Virtual Interface Cards	25
Cisco UCS Manager	25
NVIDIA GPU.....	26
NVIDIA CUDA.....	26
Cloudera Enterprise Data Hub and Hortonworks Data Platform.....	26
Cloudera (CDH 6.2.0).....	26
Cloudera Data Science Workbench	27
Hortonworks Data Platform.....	29
Apache Ambari.....	29
HDP for Data Access.....	30
Submarine	31
Docker Containerization.....	32
YARN Support For Docker	32
NVIDIA Docker	33
GPU Pooling and Isolation	33
Red Hat Ansible Automation.....	34
Solution Design.....	35

Requirements	35
Rack and PDU Configuration.....	35
Port Configuration on Fabric Interconnect	35
Cabling for Cisco UCS C240 M5.....	35
Software Distributions and Versions	36
Hortonworks Data Platform (HDP 3.1.0)	36
Red Hat Enterprise Linux (RHEL)	36
Software Versions	36
Fabric Configuration	37
Perform Initial Setup of Cisco UCS 6332 Fabric Interconnects.....	38
Configure Fabric Interconnect A	38
Configure Fabric Interconnect B	38
Log Into Cisco UCS Manager.....	39
Upgrade Cisco UCS Manager Software to Version 4.0(4b)	39
Add a Block of IP Addresses for KVM Access.....	39
Enable Uplink Ports	40
Configure VLANs	42
Enable Server Ports	44
Create Pools for Service Profile Templates.....	45
Create an Organization.....	45
Create MAC Address Pools.....	46
Create a Server Pool.....	48
Create Policies for Service Profile Templates.....	49
Create Host Firmware Package Policy.....	49
Create QoS Policies	50
Create the Local Disk Configuration Policy	53
Create the Server BIOS Policy	53
Create the Boot Policy	56
Create the Power Control Policy	58
Create the Service Profile Template	62
Configure the Storage Provisioning for the Template	64
Configure Network Settings for the Template	64
Configure the vMedia Policy for the Template	68
Configure the Server Boot Order for the Template	69
Configure the Server Assignment for the Template	70
Configure the Operational Policies for the Template.....	71
Install Red Hat Enterprise Linux 7.6	73

Post OS Install Configuration	91
Configure /etc/hosts	91
Set Up the Passwordless Login	92
Create the Red Hat Enterprise Linux (RHEL) 7.6 Local Repository	93
Create the Red Hat Repository Database	94
Set Up Ansible	95
Install httpd	97
Set Up All Nodes to Use the RHEL Repository.....	97
Upgrade the Cisco Network Driver for VIC1387	98
Install xfsprogs	98
Set Up JAVA	98
Configure NTP.....	100
Enable Syslog	102
Set ulimit.....	102
Disable SELinux.....	102
Set TCP Retries.....	103
Disable the Linux Firewall	103
Disable Swapping	104
Disable Transparent Huge Pages	104
Disable IPv6 Defaults.....	105
Configure Data Drives on Name Node and Other Management Nodes	105
Configure Data Drives on Data Nodes.....	108
Configure the Filesystem for NameNodes and Datanodes	109
Cluster Verification	110
Install HDP 3.1.0	112
Prerequisites for HDP Installation	113
Hortonworks Repository.....	113
Downgrade Snappy on All Nodes.....	115
HDP Installation	115
Install and Setup Ambari Server on rhel1.....	115
Setup Ambari Server On Admin Node(Rhel1).....	120
Launch the Ambari Server	122
Create the Cluster.....	123
Select Version.....	124
Select Hosts	125
Hostname Pattern Expressions	126
Confirm Hosts	127

Choose Services	127
Assign Masters	128
Assign Slaves and Clients.....	129
Customize Services.....	130
HDFS.....	134
MapReduce2	136
YARN	136
Configure the HDFS NameNode High Availability.....	137
HBase.....	138
Zookeeper.....	138
Storm	138
Ambari Metrics.....	138
Accumulo	139
Atlas.....	139
Kafka.....	140
Knox.....	140
SmartSense	140
Spark.....	141
Review	141
Deploy.....	142
Summary of the Installation Process	143
High Availability for HDFS NameNode and YARN ResourceManager	144
Configure the HDFS NameNode High Availability.....	144
Configure the YARN ResourceManger HA	152
Bill of Materials	157
Appendix – A	162
Configure Data Drives on Name Node and Other Management Nodes	162
Configure Data Drives on Data Nodes.....	163
Cloudera Data Science Workbench (CDSW)	164
Install the Prerequisites for CUDA.....	165
Install GCC.....	166
Install Kernel Headers and Installation Packages.....	167
Install DKMS.....	168
Install NVIDIA GPU Drivers	169
Install CUDA	170
Download and Setup NVIDIA CUDA Deep Neural Network Library (cuDNN).....	172
Installation Prerequisites for CDSW.....	173

Set Up a Wildcard DNS Subdomain	173
Supported JDK Version.....	174
IP Tables and Security on CDSW Nodes	174
Configure Block Devices	175
Download and Install CDSW with HDP 3.1.0	175
Create HDFS User Directories	176
Install Cloudera Data Science Workbench on the Master Host	177
Install Cloudera Data Science Workbench on Worker Hosts	177
Create the Administrator Account	178
Non-Kerberized Clusters.....	179
Use GPUs for Cloudera Data Science Workbench Workloads	180
Enable GPU with CDSW.....	180
Create a Custom CUDA-Capable Engine Image	182
Configure CDSW to Run Docker Containers	185
Set Up Docker Registry.....	185
Create a Custom CUDA-capable Engine Image	186
Allocate GPUs for Sessions and Jobs.....	188
About the Authors.....	191
Acknowledgements	191



CISCO™

CLOUDERA



Hortonworks



NVIDIA®

Executive Summary

Data scientists are constantly searching for newer techniques and methodologies that can unlock the value of big data and distill this data further to identify additional insights which could transform productivity and provide business differentiation.

One such area is Artificial Intelligence/Machine Learning (AI/ML), which has seen tremendous development with bringing in new frameworks and new forms of compute (CPU, GPU and FPGA) to work on data to provide key insights. While data lakes have historically been data intensive workloads, these advancements in technologies have led to a new growing demand of compute intensive workloads to operate on the same data.

While data scientists want to be able to use the latest and greatest advancements in AI/ML software and hardware technologies on their datasets, the IT team is also constantly looking at enabling these data scientists to be able to provide such a platform to a data lake. This has led to architecturally siloed implementations. When data, which is ingested, worked, and processed in a data lake, needs to be further operated by AI/ML frameworks, it often leaves the platform and has to be on-boarded to a different platform to be processed. This would be fine if this demand is seen only on a small percentage of workloads. However, AI/ML workloads working closely on the data in a data lake are seeing an increase in adoption. For instance, data lakes in customer environment are seeing deluge of data from new use cases such as IoT, autonomous driving, smart cities, genomics and financials, who are all seeing more and more demand of AI/ML processing of this data.

IT is demanding newer solutions to enable data scientists to operate on both a data lake and an AI/ML platform (or a compute farm) without worrying about the underlying infrastructure. IT also needs this to seamlessly grow to cloud scale while reducing the TCO of this infrastructure and without affecting utilization. Thus, driving a need to plan a data lake along with an AI/ML platform in a systemic fashion.

Seeing this increasing demand by IT, and also envisioning this as a natural extension of a data lake, we announced [Cisco Data Intelligence Platform](#). Cisco Data Intelligence Platform is discussed in detail [here](#).

This CVD implements Cisco Data Intelligence Platform on Cisco Unified Computing System (Cisco UCS) using Hortonworks Data Platform 3.1.

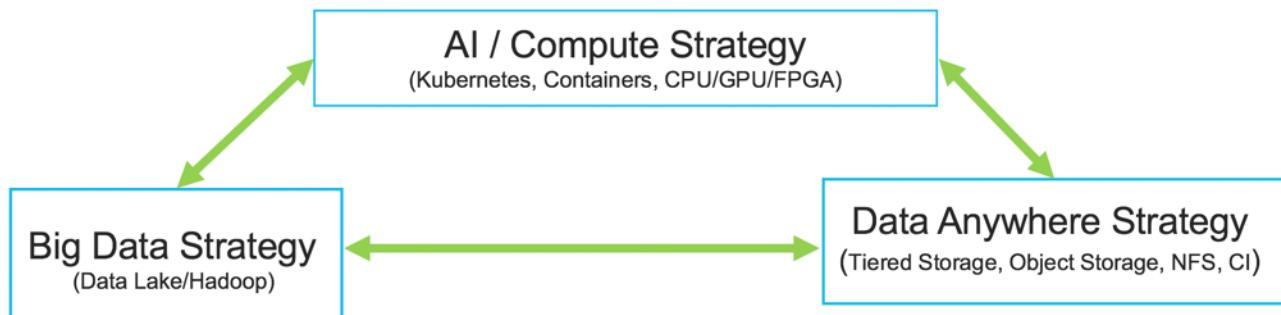
Solution Overview

Cisco Data Intelligence Platform

Cisco Data Intelligence Platform (CDIP) is a cloud scale architecture which brings together big data, AI/compute farm, and storage tiers to work together as a single entity while also being able to scale independently to address the IT issues in the modern data center. This architecture allows for:

- Extremely fast data ingest, and data engineering done at the data lake
- AI compute farm allowing for different types of AI frameworks and compute types (GPU, CPU, FPGA) to work on this data for further analytics
- A storage tier, allowing to gradually retire data which has been worked on to a storage dense system with a lower \$/TB providing a better TCO
- Seamlessly scale the architecture to thousands of nodes with a single pane of glass management using Cisco Application Centric Infrastructure (ACI)

Cisco Data Intelligence Platform caters to the evolving architecture bringing together a fully scalable infrastructure with centralized management and fully supported software stack (in partnership with industry leaders in the space) to each of these three independently scalable components of the architecture including data lake, AI/ML and Object stores.



Cisco has developed numerous industry leading Cisco Validated Designs (CVDs) in the area of Big Data (CVDs with Cloudera, Hortonworks, and MapR), compute farm with Kubernetes (CVD with RedHat OpenShift) and Object store (Scality, SwiftStack, Cloudian, and others).

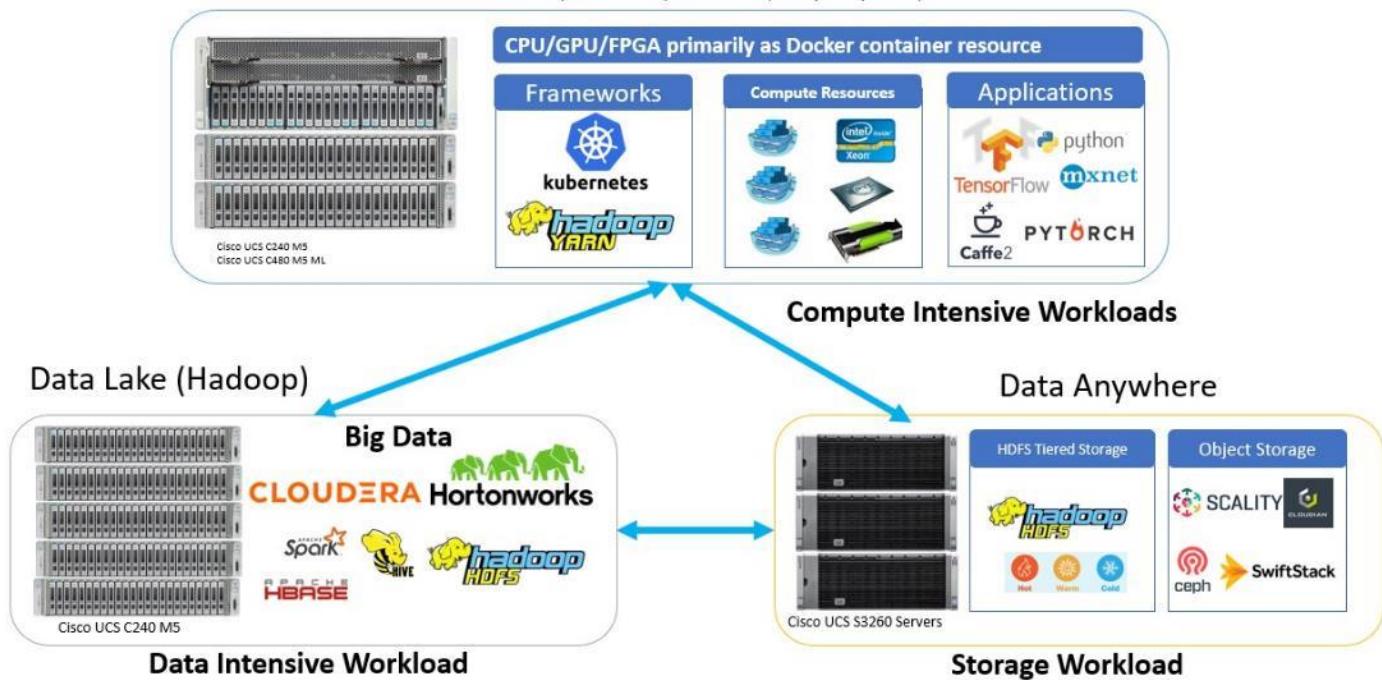
This Cisco Data Intelligence Platform can be deployed in two variants:

- CDIP with Cloudera with Data Science Workbench (powered by Kubernetes) and Tiered Storage with Hadoop
- CDIP with Hortonworks with Apache Hadoop 3.1 and Data Science Workbench (powered by Kubernetes) and Tiered Storage with Hadoop

Figure 1 Cisco Data Intelligence Platform with Hadoop, Kubernetes, and Object Store

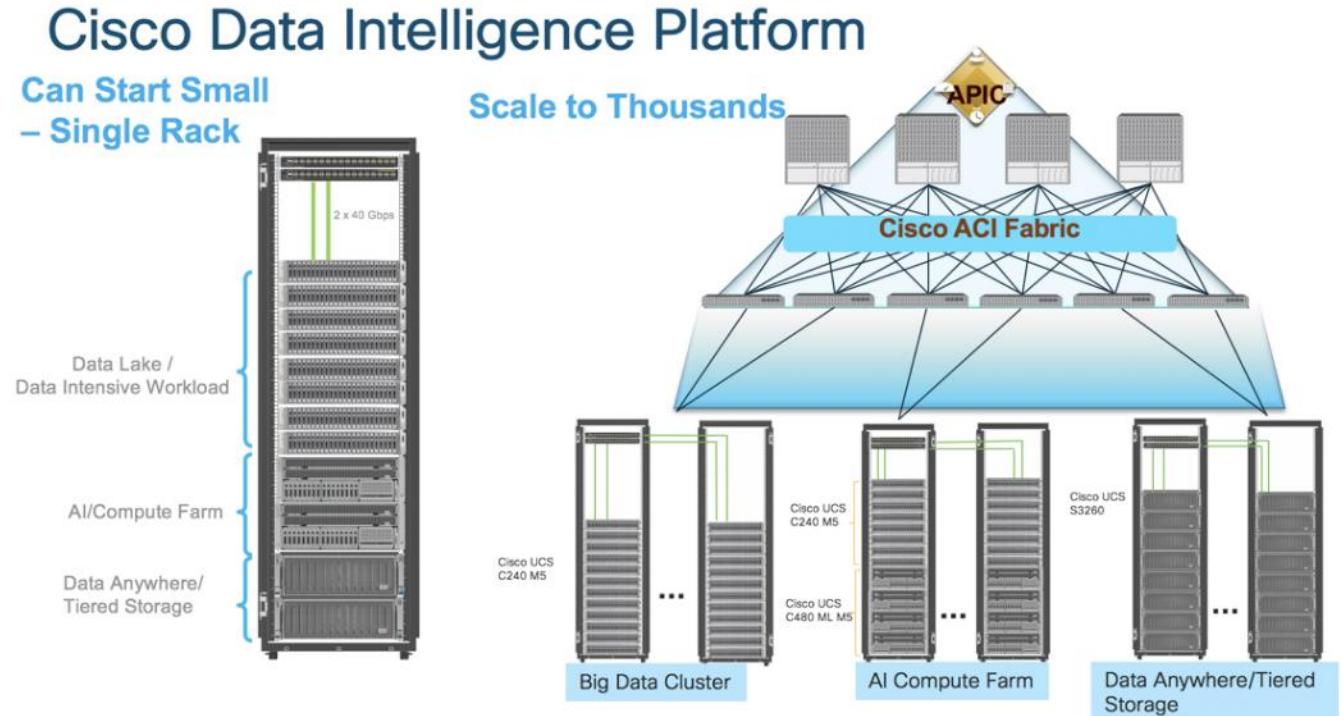
AI / Compute Farm

(Kubernetes, Containers, CPU/GPU/FPGA)



This architecture can start from a single rack and scale to thousands of nodes with a single pane of glass management with Cisco Application Centric Infrastructure (ACI).

Figure 2 Solution Architecture



Audience

The intended audience of this document includes, but not limited to, sales engineers, field consultants, professional services, IT managers, partner engineering and customers who want to deploy Cisco Data Intelligence Platform using Hortonworks Data Platform (HDP 3.1.0) on Cisco UCS. You are assumed to have intermediate level of knowledge for Apache Hadoop and Cisco UCS based scale-out infrastructure.

Purpose of this Document

This document describes the architecture and step by step guidelines of deployment procedures for Cisco Data Intelligence Platform using Hortonworks Data Platform (HDP) 3.1.0 on Cisco UCS C240 M5.

This document walks through the process of deploying the three independently scalable components of the architecture including data lake, AI/ML and Object stores:

- Data Lake with Hortonworks Data Platform 3.1 or Cloudera Enterprise Data Hub 6.2
- Kubernetes / AI farm with Cloudera Data Science Workbench
- Hadoop Tiered storage with S3260
- Distributed AI/ML with Apache Submarine

What's New in this Release?

This CVD describes the deployment procedure for the following:

- Data Lake with Cloudera Enterprise Data Hub 6.2.0 or Hortonworks Data Platform 3.1.0
- Kubernetes / AI farm with Cloudera Data Science Workbench (CDSW) 1.5
 - Enable CUDA for the GPUs
 - Enable GPU as a resource to the Docker Containers through CDSW
 - Enable GPU isolation and scheduling (with Docker Containers) through YARN 2.0
 - Downloading a TensorFlow image from NVIDIA Cloud (NGC)
 - Adding trusted registries for Docker for YARN 2.0
 - Execute a sample TensorFlow job accessing data from Hadoop and running on a Docker container with GPU as a resource scheduled by YARN 2.0
- Distributed Deep Learning with Submarine

What's Next?

This CVD showcases Cisco UCS Manager (UCSM). This solution can also be deployed using Cisco Intersight. This along with other additional Cisco UCS features will be added to the appendix section in the following months. Some of these include,

- Cisco Intersight
- Cloudera Data Science Workbench
- Tiered Storage with HDFS on Cisco UCS S3260

- Cisco Boot optimized M.2 Raid controller for hardware RAID
- 4th Generation Fabric Interconnect
- Hadoop data offload to S3 compliant storage

Reference Architecture

Table 1, Table 2, and Table 3 summarize the reference architecture configuration details for the data lake, AI/ML components of the data lake, and tiered storage.

Data Lake Reference Architecture

Table 1 lists the data lake reference architecture configuration details for Cisco UCS Integrated Infrastructure for Big Data and Analytics.

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 2 nd Gen Intel Xeon Processor Scalable Family 6230 (2 x 20 cores, 2.1 GHz)	2 x Intel Xeon Processor Scalable Family 6132 (2 x 14 cores, 2.6 GHz)	2 x 2 nd Gen Intel Xeon Processor Scalable Family 5220 (2 x 18 cores, 2.2 GHz)
Memory	12 x 32 GB 2933 MHz (384 GB)	6 x 32 GB 2666 MHz (192GB)	12 x 32 GB 2666 MHz (384 GB)
Boot	M.2 with 2 x 240-GB SSDs	M.2 with 2 x 240-GB SSDs	2 x 240G SATA BOOT SSD
Storage	26 x 2.4 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	28 x 6 TB 7.2K rpm LFF SAS HDDs
VIC	40 Gigabit Ethernet (Cisco UCS VIC 1387) or 25 Gigabit Ethernet (Cisco UCS VIC 1455)	40 Gigabit Ethernet (Cisco UCS VIC 1387) or 25 Gigabit Ethernet (Cisco UCS VIC 1455)	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 or Cisco UCS 6454 Fabric Interconnect	Cisco UCS 6332 Fabric Interconnect or Cisco UCS 6454 Fabric Interconnect	Cisco UCS 6332 Fabric Interconnect
GPU (Optional)	2 x NVIDIA TESLA V100 with 32G memory each	2 x NVIDIA TESLA V100 with 32G memory each	

AI Computing Farm Reference Architecture

Table 2 lists the AI computing farm reference architecture configuration details for high-density CPU cores and GPU nodes.

Table 2 High-Density CPU Cores and GPU Nodes

	Select stack	Elite stack	Premier stack
Servers	8 x Cisco UCS C240 M5 Rack Servers 4 x Cisco UCS C480 M5 Rack Servers	8 x Cisco UCS C240 M5 Rack Servers 4 x Cisco UCS C480 ML M5 Rack Servers	8 x Cisco UCS C4200 Rack Servers Each with: 4 x Cisco UCS C125 M5 Rack Servers
CPU	2 x 2 nd Gen Intel Xeon Scalable 6230 processors (2 x 20 cores, at 2.1 GHz)	2 x Intel Xeon Scalable 6230 processors (2 x 16 cores, at 2.6 GHz)	2 x AMD EPYC 7401 processors (2 x 24 cores, at 2.0 or 2.8 GHz)
Memory	12 x 32GB 2933MHz DDR4 (384 GB)	12 x 32GB 2666MHz DDR4 (384 GB)	16 x 32GB 2666 MHz DDR4 (512 GB)
Boot	M.2 with 2 x 960-GB SSDs	M.2 with 2 x 960-GB SSDs	M.2 with 2 x 240-GB SATA SSDs
Storage	26 x 1.8-TB 10K rpm SFF SAS HDDs or 12 x 1.6-TB Enterprise Value SATA SSDs	24 x 1.8-TB 10K rpm SFF SAS HDDs or 12 x 1.6-TB Enterprise Value SATA SSDs	6 x 3.8-TB Enterprise Value SATA SSDs
VIC	40 Gigabit Ethernet (Cisco UCS VIC 1387) or 25 Gigabit Ethernet (Cisco UCS VIC 1455)	40 Gigabit Ethernet (Cisco UCS VIC 1387) or 25 Gigabit Ethernet (Cisco UCS VIC 1455)	25 Gigabit Ethernet (Cisco UCS VIC 1455)
Storage controller	Cisco 12-Gbps SAS modular RAID controller with 4-GB FBWC or Cisco 12-Gbps modular SAS HBA	Cisco 12-Gbps SAS modular RAID controller with 4-GB FBWC or Cisco 12-Gbps modular SAS HBA	Cisco 12-Gbps SAS 9460-8i RAID controller with 2-GB FBWC
Network connectivity	Cisco UCS 6332 Fabric Interconnect or Cisco UCS 6454 Fabric Interconnect	Cisco UCS 6332 Fabric Interconnect or Cisco UCS 6454 Fabric Interconnect	Cisco UCS 6454 Fabric Interconnect
GPU	For C240 M5: 2 x NVIDIA TESLA V100 with 32-GB memory each or 2 x NVIDIA T4 For C480 M5: 4 x NVIDIA TESLA V100 with 32-GB memory each or 4 x NVIDIA T4	For C240 M5: 2 x NVIDIA TESLA V100 with 32-GB memory each or 2 x NVIDIA T4 For C480 M5 ML: 8 x NVIDIA TESLA V100 with 32-GB memory each and with NVLink	



High density GPU servers have higher storage for OS M.2 drives for docker volumes on the OS drives.

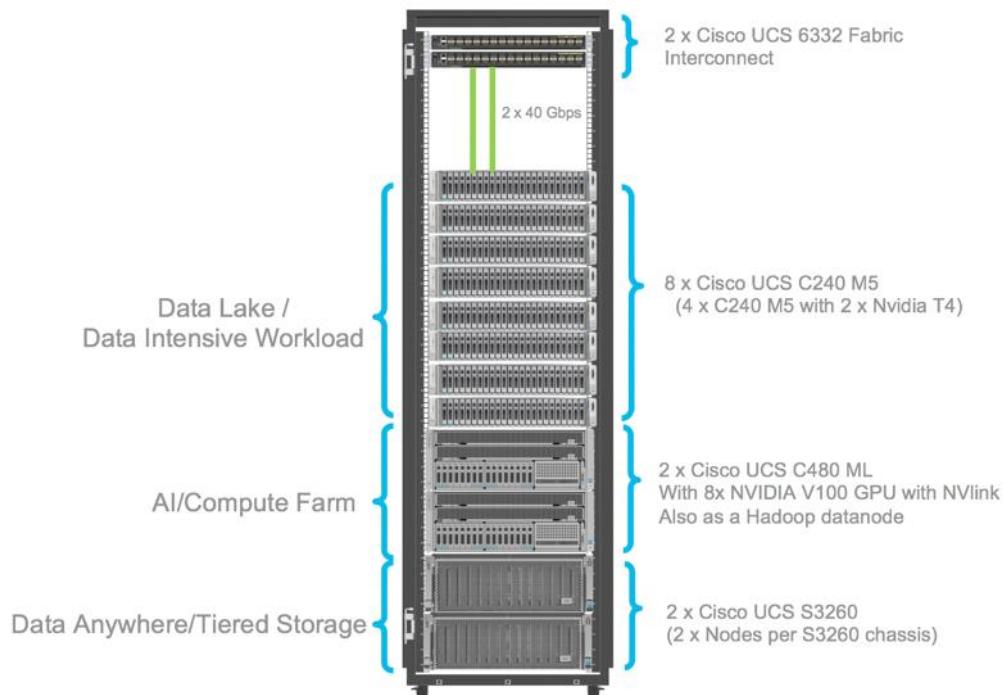
Table 3 lists the tiered storage reference architecture configuration details for Cisco UCS Integrated Infrastructure for Big Data and Analytics.

Table 3 Tiered Storage (Data Lake Reference Architecture)

	High capacity
Servers	8 x Cisco UCS S3260 Storage Servers
CPU	2 x Intel Xeon Scalable 5220 processors (2 x 18 cores, at 2.2 GHz)
Memory	12 x 32-GB 2666 MHz (192 GB)
Boot	2 x 240G SATA BOOT SSD
Storage	28 x 6-TB 7.2K rpm LFF SAS HDDs
VIC	40 Gigabit Ethernet (Cisco UCS VIC 1387)
Storage controller	Cisco UCS S3260 dual RAID controller
Network connectivity	Cisco UCS 6332 Fabric Interconnect

Figure 3 illustrates a 12-node starter cluster with all the 3 components in a single rack. The top 8 nodes has Cisco UCS C240 M5 servers as a data lake. Each link in the figure represents a 40 Gigabit Ethernet link from each of the 12 servers directly connected to a Fabric Interconnect. The second 2 x Cisco UCS C480 ML M5 Servers and the last 4 servers illustrate a data tiering on 2xS3260 servers. Every server is connected to both Fabric Interconnects.

Figure 3 Topology



As illustrated in Figure 4, a 30-node starter cluster. Rack #1 has sixteen Cisco UCS C240 M5 servers. Each link in the figure represents a 40 Gigabit Ethernet link from each of the sixteen servers directly connected to a Fabric Interconnect. Rack #2 has six Cisco UCS C240 M5 and four Cisco UCS S3260 servers. Every server is connected to both Fabric Interconnects.

Figure 4 Cisco Data Intelligence Platform - 30 Node Configuration with Cloudera CDH 6.2 and CDSW 1.5

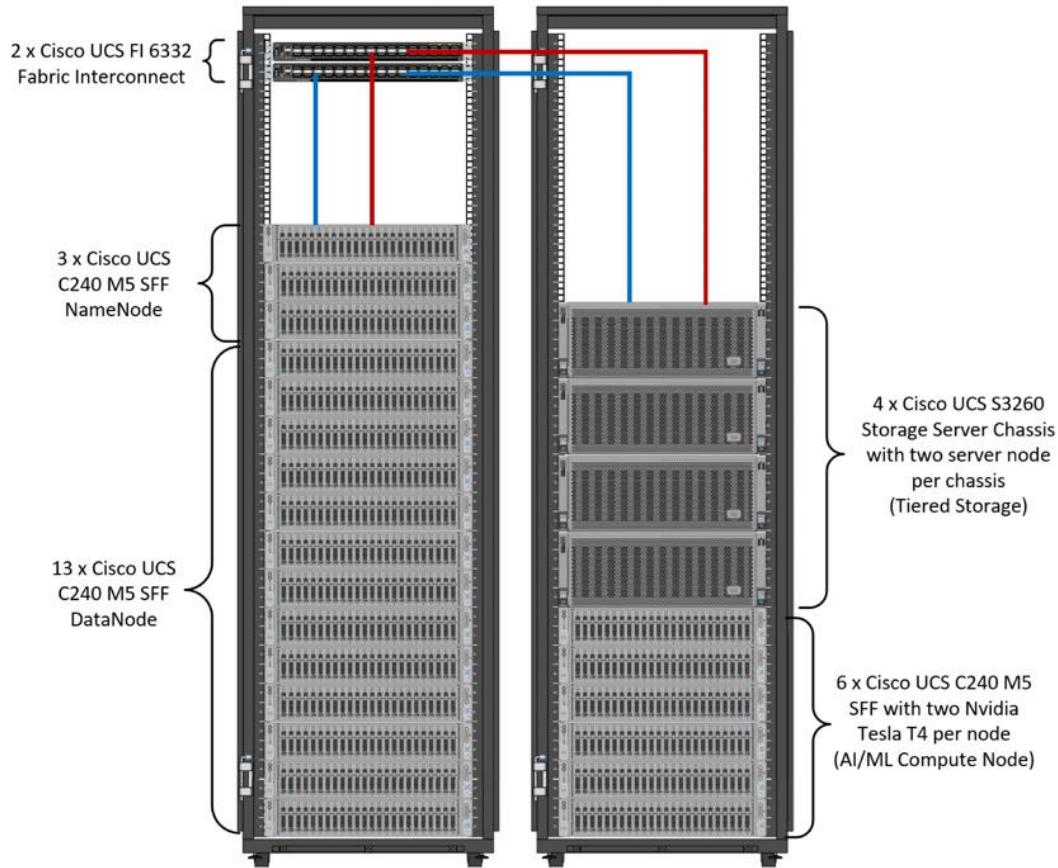


Figure 4 shows an alternate configuration for cases where more GPU capacity is needed. Four of the Cisco UCS C240 M5 servers from the previous configuration in Figure 3 are replaced with Cisco UCS C480 M5 ML M5 server which support up to eight V100 MXM GPUs.



Each Cisco UCS C480 ML M5 has 8 x NVIDIA SXM2 V100 32GB modules with NVLink interconnect. Each Cisco UCS C240 M5 supports up to two PCIe GPU adapters with NVIDIA Tesla V100. For more information about Cisco UCS C240 M5 Server installation and GPU card configuration rules, go to https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c/hw/C240M5/install/C240M5/C240M5_appendix_0101.html

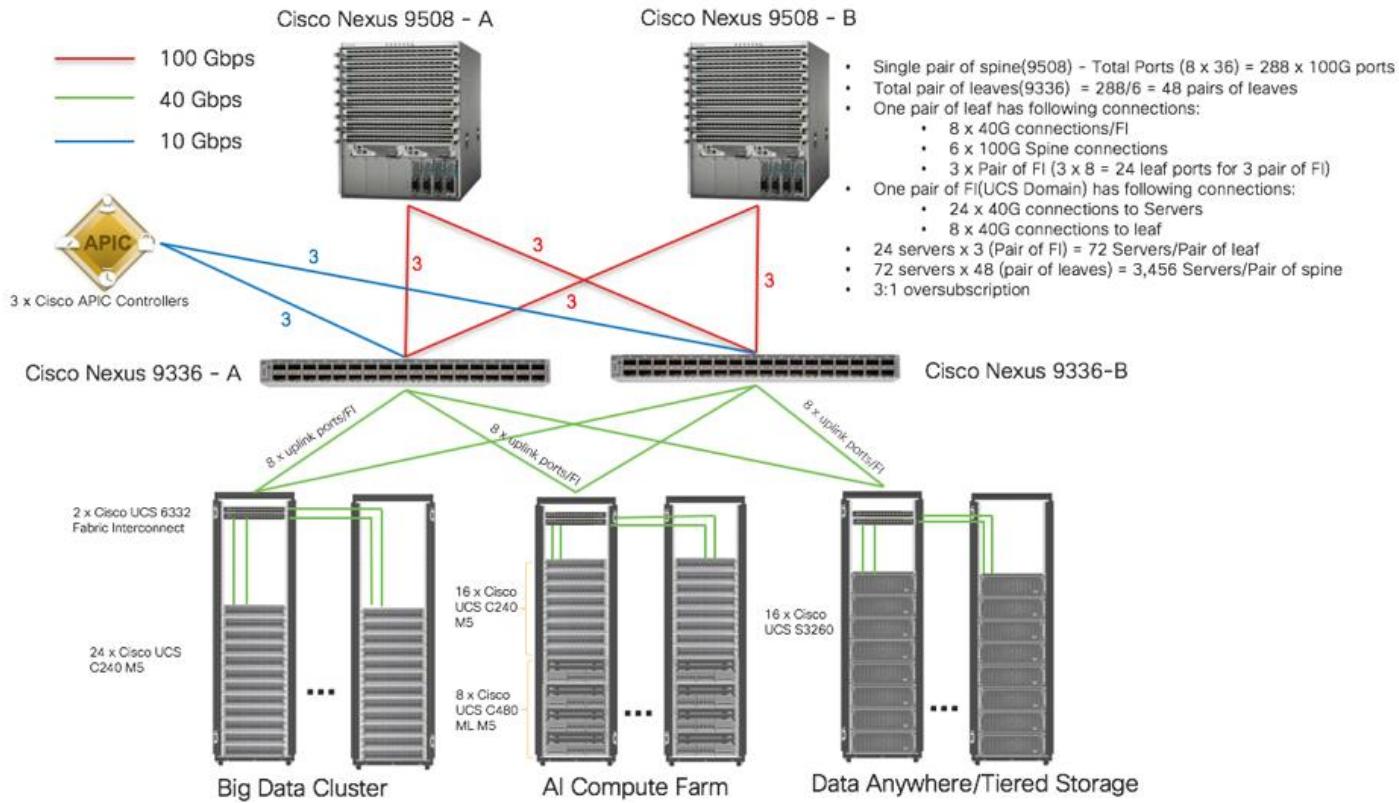


Power requirements per rack must be calculated since the exact values will change based on the power needs of the GPUs.

Scaling the Solution

Figure 5 illustrates 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 5 Scaling the Solution



In the reference architectures discussed in this document, each of the components is scaled separately, and for the purposes of this example, scaling is uniform. Two scale scenarios are discussed here:

- Scaled architecture with 3:1 oversubscription with Cisco fabric interconnects and Cisco ACI
- Scaled architecture with 2:1 oversubscription with Cisco ACI

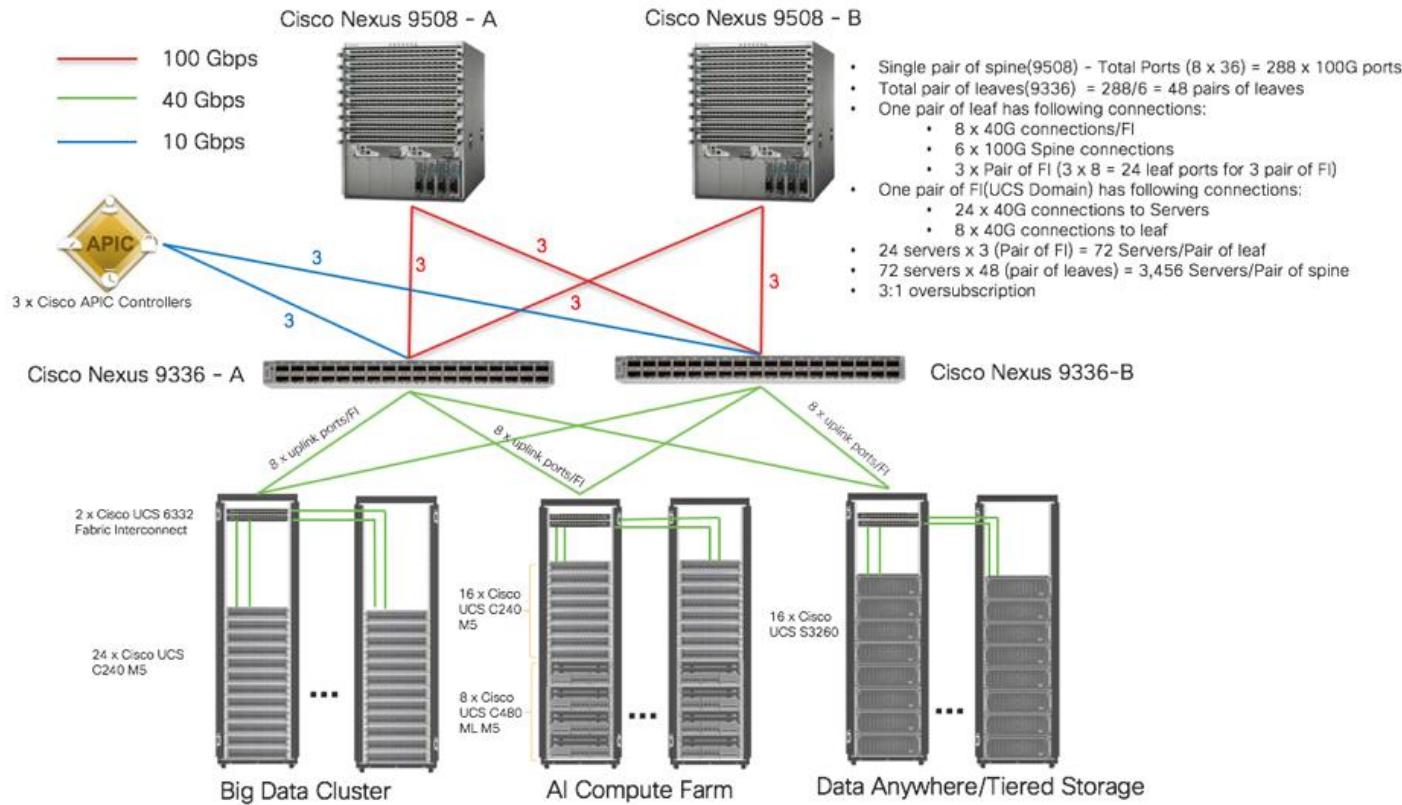
In the following scenarios, the goal is to populate up to a maximum of 200 leaf nodes in a Cisco ACI domain. Not all cases reach that number because they use the Cisco Nexus® 9508 Switch for this sizing and not the Cisco Nexus 9516 Switch.

Scaled Architecture with 3:1 Oversubscription with Cisco Fabric Interconnects and Cisco ACI

The architecture discussed in this document and shown in Figure 6 supports 3:1 network oversubscription from every node to every other node across a multidomain cluster (nodes in a single domain within a pair of Cisco fabric interconnects are locally switched and not oversubscribed).

From the viewpoint of the data lake, 24 Cisco UCS C240 M5 Rack Servers are connected to a pair of Cisco UCS 6332 Fabric Interconnects (with 32 x 40-Gbps throughput). From each fabric interconnect, 8 x 40-Gbps links connect to a pair of Cisco Nexus 9336 Switches. Two pairs of fabric interconnects can connect to a single pair of Cisco Nexus 9336 Switches (8 x 2 40-Gbps links). Each of these Cisco Nexus 9336 Switches connects to a pair of Cisco Nexus 9508 Cisco ACI switches with 6 x 100-Gbps uplinks (connecting to a Cisco NgK-X9736C-FX line card).

Figure 6 Scaled Architecture with 3:1 Oversubscription with Cisco Fabric Interconnects and Cisco ACI



Scaled Architecture with 2:1 Oversubscription with Cisco ACI

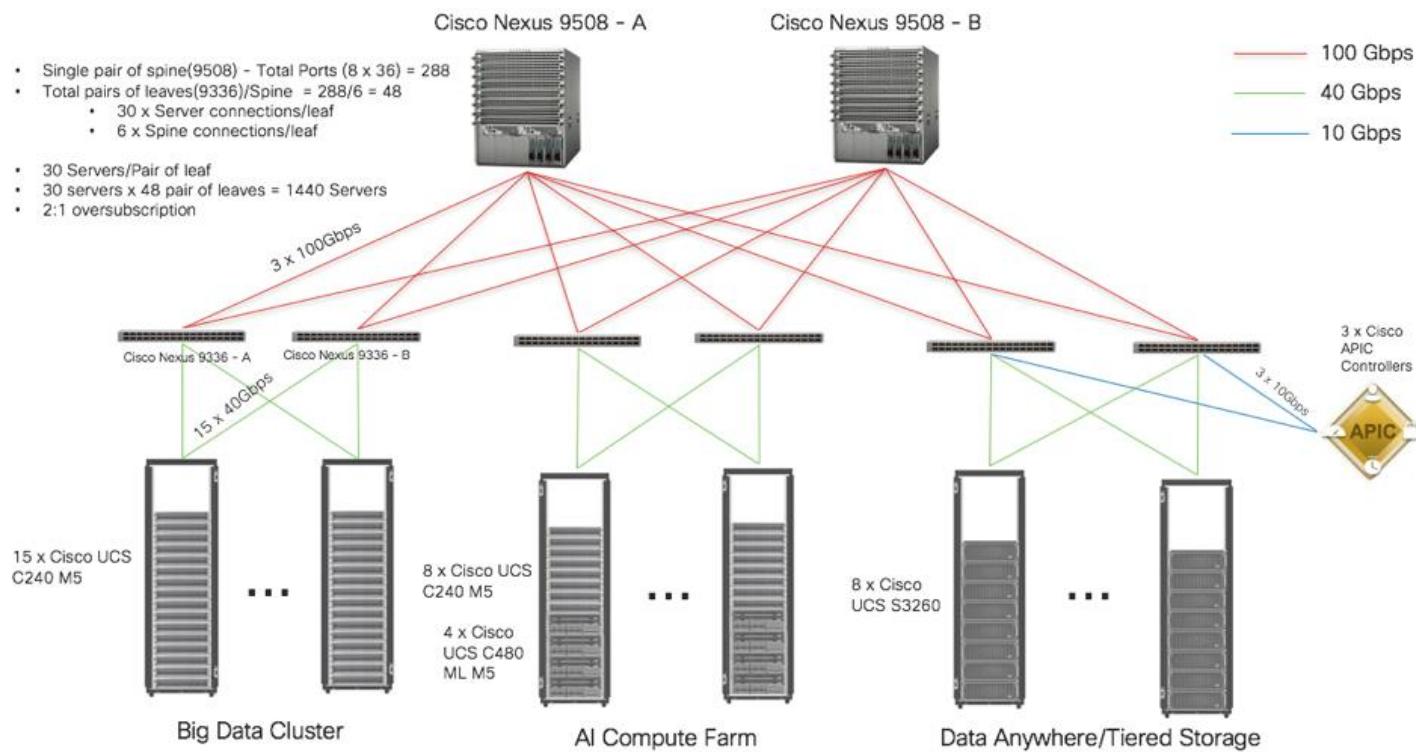
In the scenario discussed here and shown in Figure 7, the Cisco Nexus 9508 Switch with the Cisco NgK-X9736C-FX line card can support up to 36 x 100-Gbps ports, each and 8 such line cards.

For the 2:1 oversubscription, 30 Cisco UCS C240 M5 Rack Servers are connected to a pair of Cisco Nexus 9336 Switches, and each Cisco Nexus 9336 connects to a pair of Cisco Nexus 9508 Switches with three uplinks each. A pair of Cisco Nexus 9336 Switches can support 30 servers and connect to a spine with 6 x 100-Gbps links on each spine. This single pod (pair of Cisco Nexus 9336 Switches connecting to 30 Cisco UCS C240 M5 servers and 6 uplinks to each spine) can be repeated 48 times (288/6) for a given Cisco Nexus 9508 Switch and can support up to 1440 servers.

To reduce the oversubscription ratio (to get 1:1 network subscription from any node to any node), you can use just 15 servers under a pair of Cisco Nexus 9336 Switches and then move to Cisco Nexus 9516 Switches (the number of leaf nodes would double).

To scale beyond this number, multiple spines can be aggregated.

Figure 7 Scaled Architecture with 2:1 Oversubscription with Cisco ACI



Technology Overview

Cisco UCS Integrated Infrastructure for Big Data and Analytics

The Cisco UCS Integrated Infrastructure for Big Data and Analytics solution for Hortonworks Data Platform on [Cisco UCS Integrated Infrastructure for Big Data and Analytics](#), is 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 Unified Computing System

Cisco Unified Computing System (Cisco UCS) is a next-generation solution for blade and rack server computing. Cisco UCS integrates a low-latency, lossless 10 and 40 Gigabit Ethernet unified network fabric with enterprise-class, x86-architecture servers. Cisco UCS is an integrated, scalable, multi-chassis platform in which all resources participate in a unified management domain. Cisco UCS accelerates the delivery of new services simply, reliably, and securely through end-to-end provisioning and migration support for both virtualized and non-virtualized systems. Cisco UCS fuses access layer networking and servers. This high-performance, next-generation server system provides a data center with a high degree of workload agility and scalability.

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 (UCSM). 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 8 Cisco UCS 6332 UP 32 -Port Fabric Interconnect



Cisco UCS C-Series Rack-Mount Servers

Cisco UCS C-Series Rack-Mount Servers keep pace with Intel Xeon processor innovation by offering the latest processors with increased processor frequency and improved security and availability features. With the increased performance provided by the Intel Xeon Scalable Family Processors, Cisco UCS C-Series servers offer an improved price-to-performance ratio. They also extend Cisco UCS innovations to an industry-standard rack-mount form factor, including a standards-based unified network fabric, Cisco VN-Link virtualization support, and Cisco Extended Memory Technology.

It is designed to operate both in standalone environments and as part of Cisco UCS managed configuration, these servers enable organizations to deploy systems incrementally—using as many or as few servers as needed—on a schedule that best meets the organization's timing and budget. Cisco UCS C-Series servers offer investment protection through the capability

to deploy them either as standalone servers or as part of Cisco UCS. One compelling reason that many organizations prefer rack-mount servers is the wide range of I/O options available in the form of PCIe adapters. Cisco UCS C-Series servers support a broad range of I/O options, including interfaces supported by Cisco and adapters from third parties.

Cisco UCS C240 M5 Rack-Mount Server

The Cisco UCS C240 M5 Rack-Mount Server (Figure 9) 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 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

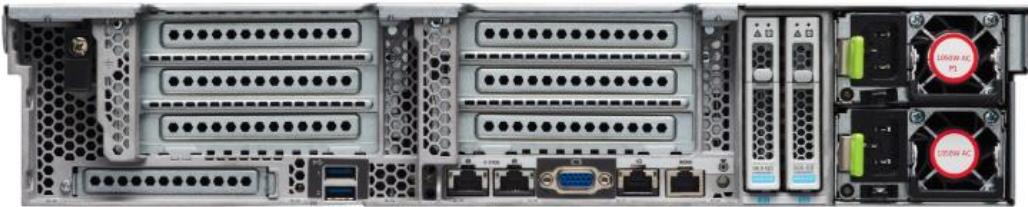
Non-Volatile Memory Express (NVMe) PCI Express (PCIe) Solid-State Disks (SSDs) compared to the previous generation of servers. These improvements deliver significant performance and efficiency gains that will improve your application performance. The Cisco UCS C240 M5 delivers outstanding levels of storage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

- Latest 2nd Gen 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 9 Cisco UCS C240 M5 Rack-Mount Server – Front View



Figure 10 Cisco UCS C240 M5 Rack-Mount Server – Rear View



Cisco UCS C480 M5 Rack-Mount Server

The Cisco UCS C480 M5 Rack-Mount Server is a storage and I/O-optimized enterprise-class rack-mount 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 11) delivers outstanding levels of expandability and performance for standalone or Cisco Unified Computing System managed environments in a 4RU form-factor. 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
- 2933-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 11 Cisco UCS C480 M5 Rack-Mount Server – Front View



Figure 12 Cisco UCS C480 M5 Rack-Mount Server – Rear View



For more information about Cisco UCS C480 M5 Rack Server, go to:

<https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/datasheet-c78-739291.html>

Cisco UCS C480 ML M5 Rack Server

The Cisco UCS C480 ML M5 Rack Server is a purpose-built server for Deep Learning. It is storage and I/O optimized to deliver an industry-leading performance for training Models. The Cisco UCS C480 ML M5 delivers outstanding levels of storage expandability and performance options for standalone or Cisco Unified Computing System managed environments in a 4RU form factor. Because of its modular design, you pay for only what you need. It offers these capabilities:

- 8 NVIDIA SXM2 V100 32G modules with NVLink interconnect
- Latest Intel Xeon Scalable processors with up to 28 cores per socket and support for two processor configurations
- 2666-MHz DDR4 memory and 24 DIMM slots for up to 3 terabytes (TB) of total memory
- 4 PCI Express (PCIe) 3.0 slots for 100G UCS VIC 1495
- Flexible storage options with support for up to 24 Small-Form-Factor (SFF) 2.5-inch, SAS/SATA Solid-State Disks (SSDs) and Hard-Disk Drives (HDDs)
- Up to 6 PCIe NVMe disk drives
- Cisco 12-Gbps SAS Modular RAID Controller in a dedicated slot
- M.2 boot options
- Dual embedded 10 Gigabit Ethernet LAN-On-Motherboard (LOM) ports

Figure 13 Cisco UCS C480 ML M5 Purpose Built Deep Learning Server – Front View



Figure 14 Cisco UCS C480 ML M5 Purpose Built Deep Learning Server – Rear View



For more information about Cisco UCS C480 ML M5 Server, go to:

<https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/c480m5-specsheet-ml-m5-server.pdf>

Table 4 lists the features and benefits of Cisco UCS C480 ML M5 Server.

Table 4 Feature and Benefits for Cisco UCS C480 ML M5 Server

Feature	Benefits
8 x NVIDIA SXM2 V100 32GB modules with NVLink interconnect	Fast Deep Learning model training
Modular storage support with up to 24 front accessible hot-swappable Hard Disk Drives (HDDs) and Solid-State Disks (SSDs)	Modularity to right-size storage options to match training requirements Flexibility to expand as storage needs increase
High-capacity memory support of up to 3 TB using 128-GB DIMMs	Large memory footprint to deliver performance and capacity for large model training
Up to 6 PCIe NVMe drives	Up to 6 Gen3 x4 lanes NVMe drives for extreme I/O performance for faster model training
Support for up to 4 PCIe Generation 3.0 slots	Support for up to four 10/25 or 40/100G Cisco VICs
Hot-swappable, redundant power supplies	Increased high availability

Feature	Benefits
Integrated dual 10-Gbps Ethernet	Increased network I/O performance and additional network options

Cisco UCS Virtual Interface Cards

Cisco UCS Virtual Interface Cards (VICs) 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 15 Cisco UCS VIC 1387



For more information about Cisco UCS Adapters, go to: <https://www.cisco.com/c/en/us/products/interfaces-modules/unified-computing-system-adapters/index.html>

Cisco UCS Manager

Cisco UCS Manager (UCSM) 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 GUI, a CLI, or an XML 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.

For more information about Cisco UCS Manager, go to: <https://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-manager/index.html>

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 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 is 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.

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 Enterprise Data Hub and Hortonworks Data Platform

This CVD can be implemented with Cloudera Enterprise Data Hub and also with Hortonworks Data Platform

Cloudera (CDH 6.2.0)

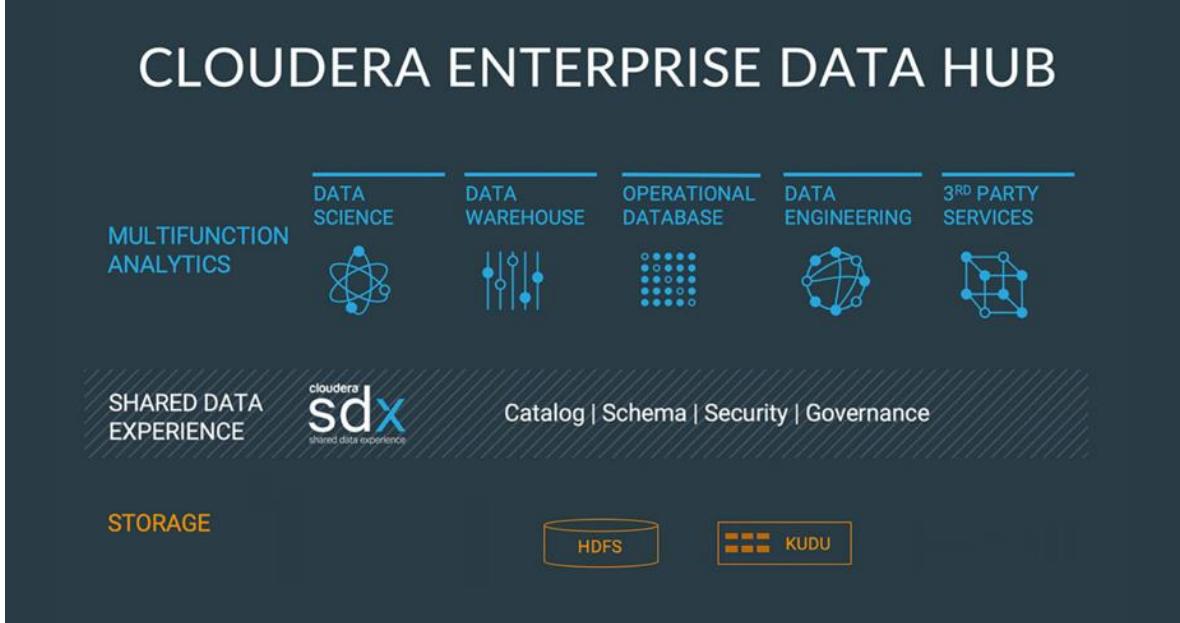
Built on the transformative Apache Hadoop open source software project, Cloudera Enterprise is a hardened distribution of Apache Hadoop and related projects designed for the demanding requirements of enterprise customers. Cloudera is the leading contributor to the Hadoop ecosystem, and has created a 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 16 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, and so on. Additionally, CDSW enables access to available GPU resources for deep learning workloads which means users become productive faster with no need for retraining and no time lost learning a new programming language.

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.

Additionally, 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.

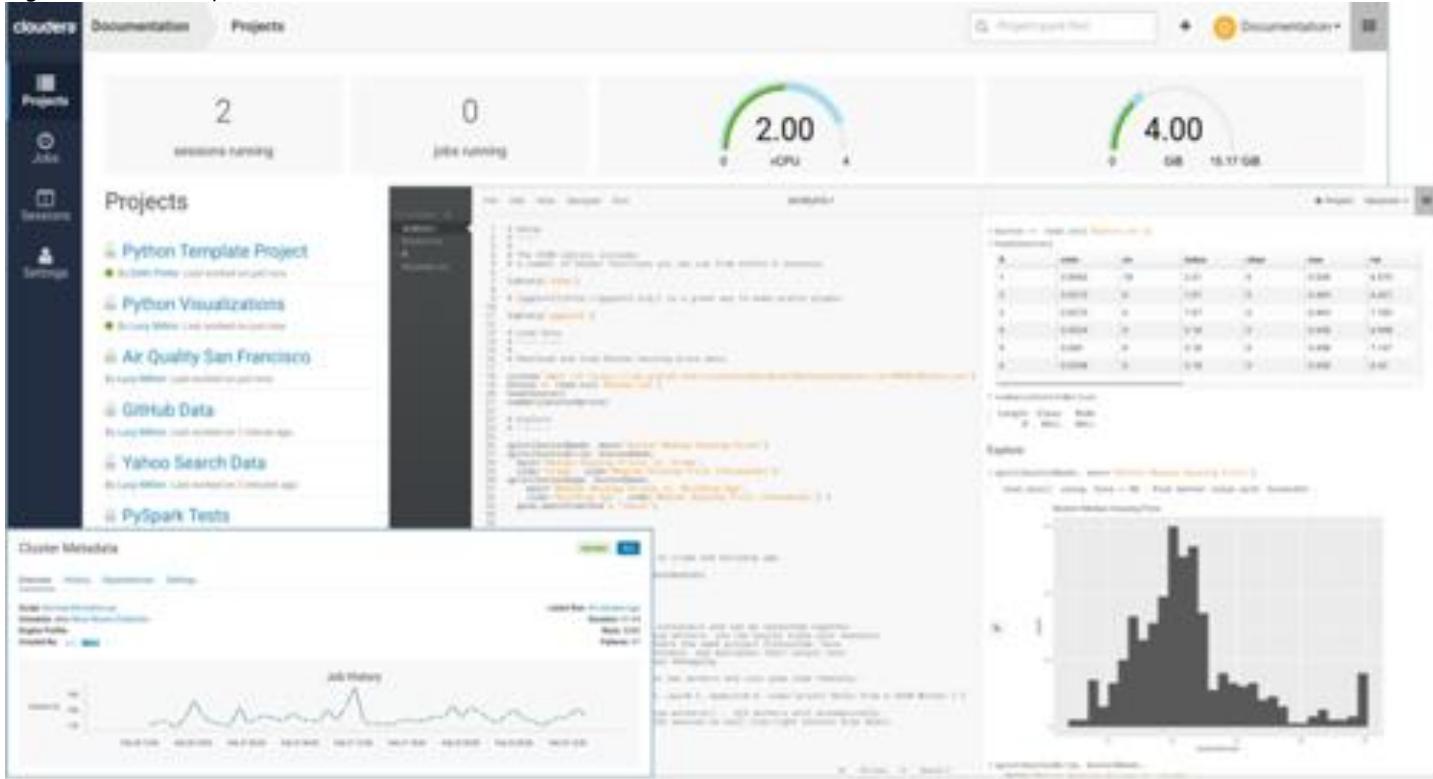
Cloudera Data Science Workbench allows you to automate analytics workloads with a built-in job and pipeline scheduling system that supports real-time monitoring, job history, and email alerts. Jobs are created and can be configured to run on a recurring schedule, as well as providing alerts for successful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

Data Science Workbench provides the following features:

- CPU and GPU as a resource: Data Science Workbench provides basic support for the use of existing general-purpose CPUs for each stage of the workflow and, optionally, accelerates the math-intensive steps with the selective application of special-purpose GPUs all through a Docker container, with Kubernetes scheduling these resources in the back end.
- Self-service portal: The Data Science Workbench web user interface console provides a self-service portal for data scientists to create an environment for their workloads (Figure 17). Currently, R, Python, and Scala are supported.
- Jupyter Notebook: Most data scientists use Jupyter Notebooks for AI/ML analysis and development. Data Science Workbench provides a Jupyter Notebook environment when data scientists create a portal, and these notebooks can be shared or worked in a collaborative manner.

Figure 17 Example of Cloudera Data Science Workbench WebUI



Hortonworks Data Platform

The Hortonworks Data Platform (HDP 3.1.0) delivers essential capabilities in a completely open, integrated and tested platform that is ready for enterprise usage. With Hadoop YARN at its core, HDP provides flexible enterprise data processing across a range of data processing engines, paired with comprehensive enterprise capabilities for governance, security and operations.

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

Hortonworks Data Platform (HDP) 3.0 delivers significant new features, including the ability to launch apps in a matter of minutes and address new use cases for high-performance deep learning and machine learning apps. In addition, this new version of HDP enables enterprises to gain value from their data faster, smarter, in a hybrid environment.

Apache Ambari

Apache Ambari is a completely open source management platform. It performs provisioning, managing, securing, and monitoring Apache Hadoop clusters. Apache Ambari is a part of Hortonworks Data Platform and it allows enterprises to plan and deploy HDP cluster. It also provides ongoing cluster maintenance and management.

Ambari provides an intuitive Web UI as well as an extensive REST API framework which is very useful for automating cluster operations.

The following are the core benefits that Hadoop operators get with Ambari:

- Simplified Installation, Configuration and Management. Easily and efficiently create, manage and monitor clusters at scale. Takes the guesswork out of configuration with [Smart Configs](#) and Cluster Recommendations. Enables repeatable, automated cluster creation with [Ambari Blueprints](#).

- **Centralized Security Setup.** Reduce the complexity to administer and configure cluster security across the entire platform. Helps automate the setup and configuration of advanced cluster security capabilities such as Kerberos and [Apache Ranger](#).
- **Full Visibility into Cluster Health.** Ensure your cluster is healthy and available with a holistic approach to monitoring. Configures predefined alerts — based on operational best practices — for cluster monitoring. Captures and visualizes critical operational metrics — using [Grafana](#) — for analysis and troubleshooting. Integrated with [Hortonworks SmartSense](#) for proactive issue prevention and resolution.
- **Highly Extensible and Customizable.** Fit Hadoop seamlessly into your enterprise environment. Highly extensible with [Ambari Stacks](#) for bringing custom services under management, and with [Ambari Views](#) for customizing the Ambari Web UI.

HDP for Data Access

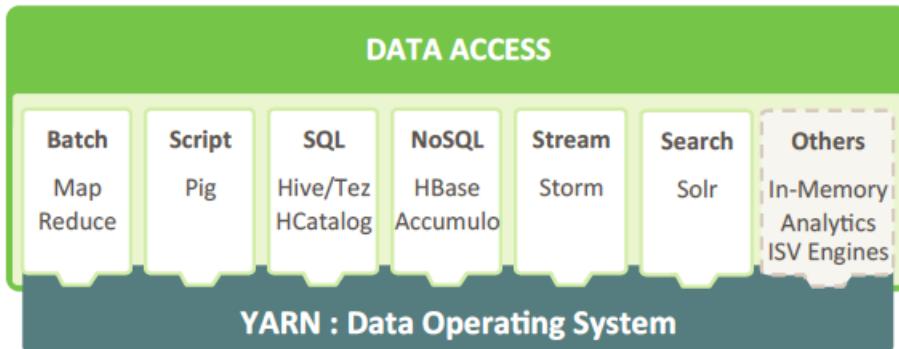
With YARN at its foundation, HDP provides a range of processing engines that allow users to interact with data in multiple and parallel ways, without the need to stand up individual clusters for each data set/application. Some applications require batch while others require interactive SQL or low-latency access with NoSQL. Other applications require search, streaming or in-memory analytics. Apache Solr, Storm and Spark fulfill those needs respectively.

To function as a true data platform, the YARN-based architecture of HDP enables the widest possible range of access methods to coexist within the same cluster avoiding unnecessary and costly data silos.

As shown in Figure 18, HDP Enterprise natively provides for the following data access types:

- **Batch** – Apache MapReduce has served as the default Hadoop processing engine for years. It is tested and relied upon by many existing applications.
- **Interactive SQL Query** - Apache Hive is the de facto standard for SQL interactions at petabyte scale within Hadoop. Hive delivers interactive and batch SQL querying across the broadest set of SQL semantics.
- **Search** - HDP integrates Apache Solr to provide high-speed indexing and sub-second search times across all your HDFS data.
- **Scripting** - Apache Pig is a scripting language for Hadoop that can run on MapReduce or Apache Tez, allowing you to aggregate, join and sort data.
- **Low-latency access via NoSQL** - Apache HBase provides extremely fast access to data as a columnar format, NoSQL database. Apache Accumulo also provides high-performance storage and retrieval, but with fine-grained access control to the data.
- **Streaming** - Apache Storm processes streams of data in real time and can analyze and take action on data as it flows into HDFS.

Figure 18 YARN



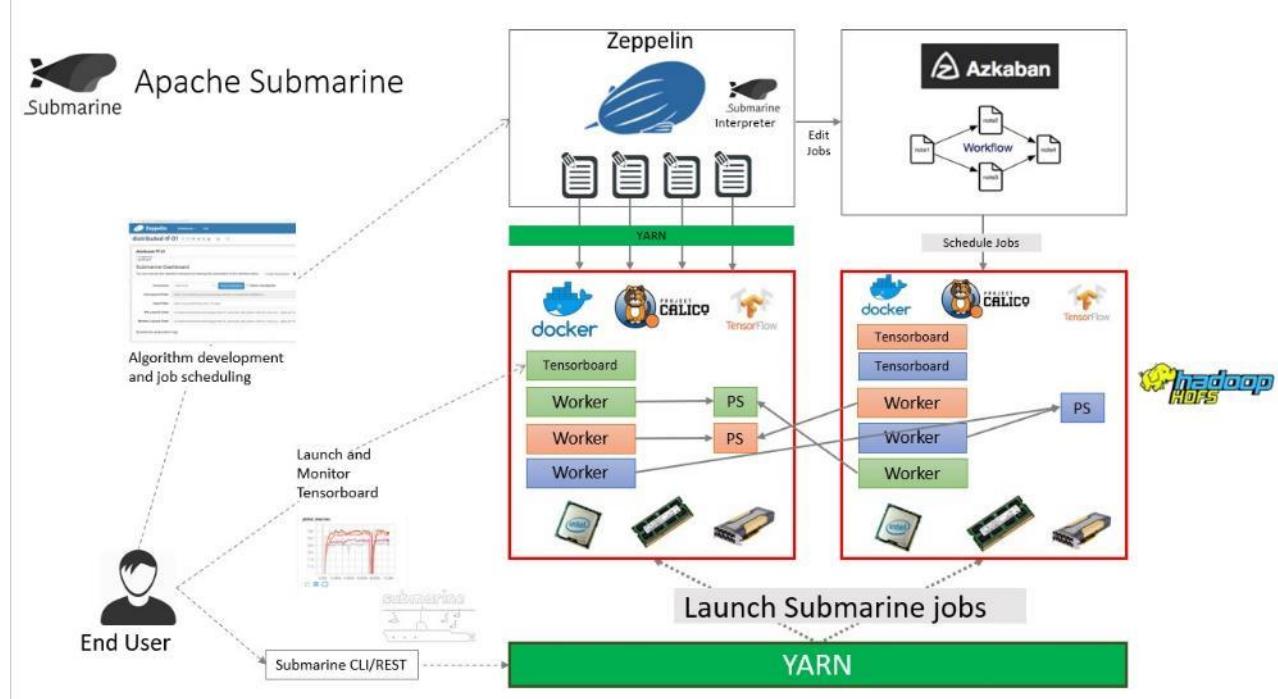
Submarine

Deep learning is useful for enterprises tasks in the field of speech recognition, image classification, AI chatbots, machine translation, just to name a few. In order to train deep learning/machine learning models, frameworks such as TensorFlow / MXNet / PyTorch / Caffe / XGBoost can be leveraged. And sometimes these frameworks are used together to solve different problems.

To make distributed deep learning/machine learning applications easily launched, managed and monitored, Hadoop community initiated the Submarine project along with other improvements such as first-class GPU support, Docker container support, container-DNS support, scheduling improvements, and so on.

These improvements make distributed deep learning/machine learning applications run on Apache Hadoop YARN as simple as running it locally, which can let machine-learning engineers focus on algorithms instead of worrying about underlying infrastructure. By upgrading to latest Hadoop, users can now run deep learning workloads with other ETL/streaming jobs running on the same cluster. This can achieve easy access to data on the same cluster and achieve better resource utilization.

Figure 19 Submarine Workflow



Docker Containerization

Hortonworks Data Platform (HDP 3.0) 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 that performs operating-system-level virtualization, also known as "containerization." 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:

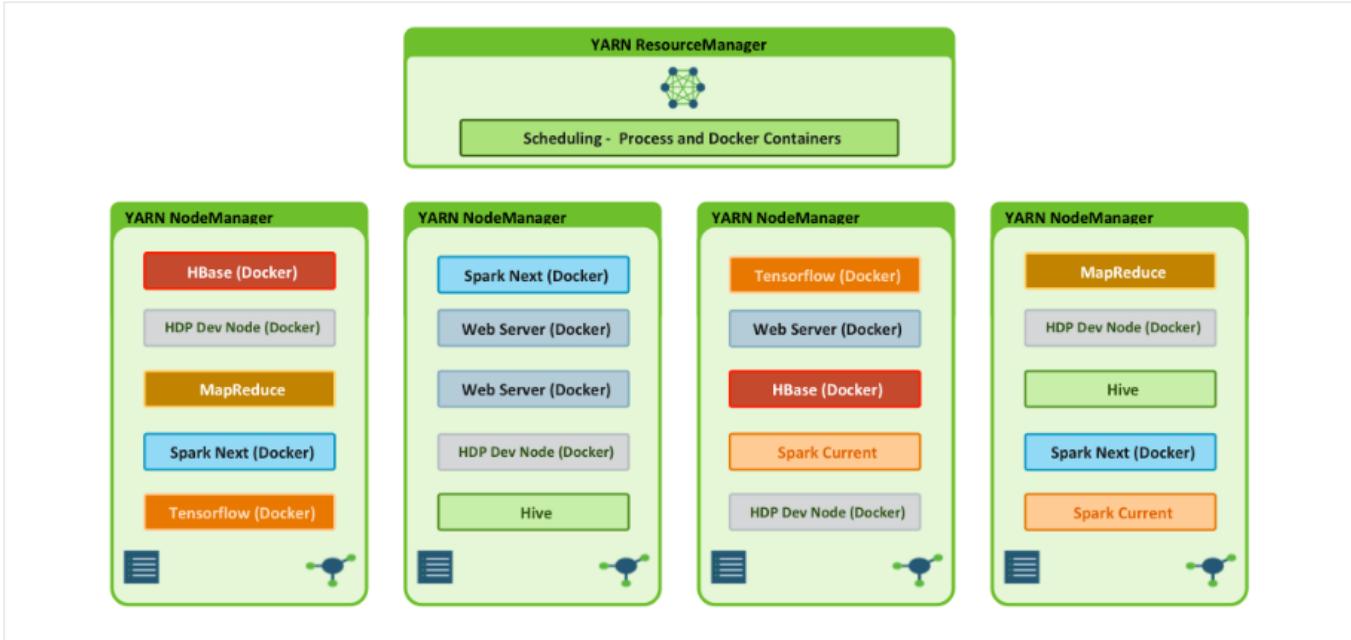
- **Ease of use:** Docker makes easier for anyone—developers, systems admins, architects and others—to take advantage of containers in order to quickly build and test portable applications. It allows anyone to package an application on their development system, which can then run *unmodified* on any 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.

YARN Support For Docker

Containerization provides YARN support for Docker containers, which makes it easier to bundle libraries and dependencies along with their application, allowing third-party applications to run on Apache Hadoop (for example, containerized applications), enabling:

- Faster time to deployment by enabling third-party apps.
- The ability to run multiple versions of an application, enabling users to rapidly create features by developing and testing new versions of services without disrupting old ones.
- Improved resource utilization and increased task throughput for containers, yielding faster time to market for services.
- Orchestration of stateless distributed applications.
- Packaging libraries for Spark application, eliminating the need for operations to deploy those libraries cluster wide.

Figure 20 Containerized Application on Apache Hadoop YARN 3.1



As shown in Figure 20, YARN Services Framework in addition with Docker containerization, it is now possible to run both existing Hadoop frameworks, such as Hive, Spark, etc., and new containerized workloads on the same underlying infrastructure. Apache Hadoop 3.1 further improved these capabilities to enable advanced use cases such as TensorFlow and HBase.

NVIDIA Docker

Docker containers are platform-agnostic, but also hardware-agnostic. This presents a problem when using specialized hardware such as NVIDIA GPUs which require kernel modules and user-level libraries to operate. As a result, Docker does not natively support NVIDIA GPUs within containers.

One of the early workarounds to this problem was to fully install the NVIDIA drivers inside the container and map in the character devices corresponding to the NVIDIA GPUs (for example, /dev/nvidiao) on launch. This solution is brittle because the version of the host driver must exactly match the version of the driver installed in the container. This requirement drastically reduced the portability of these early containers, undermining one of Docker's more important features.

To enable portability in Docker images that leverage NVIDIA GPUs, NVIDIA developed `nvidia-docker`, an open-source project hosted on GitHub that provides the two critical components needed for portable GPU-based containers:

- driver-agnostic CUDA images; and a Docker command line wrapper that mounts the user mode components of the driver and the GPUs (character devices) into the container at launch.
- `nvidia-docker` is essentially a wrapper around the `docker` command that transparently provisions a container with the necessary components to execute code on the GPU.



As of the publishing of this CVD, Hortonworks only supports `nvidia-docker` version 1.

GPU Pooling and Isolation

GPU pooling and isolation allows GPU to be a first-class resource type in Hadoop, making it easier for customers to run machine learning and deep learning workloads.

- Compute-intensive analytics require not only a large compute pool, but also a fast and expensive processing pool with GPUs in tandem
- Customers can share cluster-wide GPU resources without having to dedicate a GPU node to a single tenant or workload
- GPU isolation dedicates a GPU to an application so that no other application has access to that GPU

When it comes to resource scheduling, it is important to recognize GPU as a resource. YARN extends the resource model to more flexible mode which makes it easier to add new countable resource-types. When GPU is added as resource type, YARN can schedule applications on GPU machines. Furthermore, by specifying the number of requested GPU to containers, YARN can find machines with available GPUs to satisfy container requests.



When GPU scheduling is enabled, YARN can schedule non-GPU applications such as LLAP, Tez, and etc. to servers without GPU. Moreover, YARN can allocate GPU applications such as TensorFlow, Caffe, MXNet, and so on, to servers with GPU.

Red Hat Ansible Automation

Red Hat Ansible Automation is a powerful IT automation tool. It is capable of provisioning numerous types of resources and deploying applications. It can configure and manage devices and operating system components. Due to its simplicity, extensibility, and portability, this solution extensively utilizes Ansible for performing repetitive deployment steps across the nodes.



For more information about Ansible, go to: <https://www.redhat.com/en/technologies/management/ansible>.

Solution Design

Requirements

This CVD describes the architecture and deployment procedures for Hortonworks Data Platform (HDP) 3.1.0 on a 31 node cluster based on Cisco UCS Integrated Infrastructure for Big Data and Analytics. The solution goes into detail configuring HDP 3.1.0 on the Cisco UCS Integrated infrastructure for Big Data. In addition, it also details the configuration for Hortonworks Dataflow for various use cases.

The cluster configuration consists of the following:

- 2 Cisco UCS 6332UP Fabric Interconnects
- 22 Cisco UCS C240 M5 Rack-Mount servers
- 8 Cisco UCS C3260 M5 Storage Server
- 12 NVIDIA T4 GPUs
- 2 Cisco R42610 standard racks
- 4 Vertical Power distribution units (PDUs) (Country Specific) per rack

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 6 Cisco UCS C240 M5 Servers and 4 Cisco UCS S3260 Modular Storage Server Chassis connected to each of the vertical PDUs for redundancy; thereby, ensuring availability during power source failure, similar to the first rack.

Port Configuration on Fabric Interconnect

Table 5 lists the port configuration on Cisco UCS FI 6332 Fabric Interconnect.

Table 5 Port Configuration on Fabric Interconnect

Port Type	Port Number
Server	1-26
Network	29-32



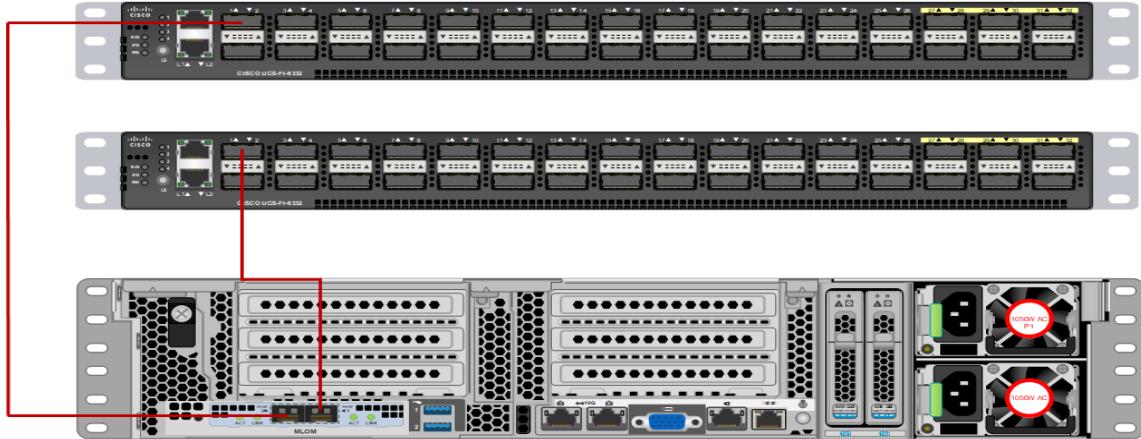
Please contact your Cisco representative for country specific information.

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 1387 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.

Figure 21 illustrates the port connectivity between the Fabric Interconnect, and Cisco UCS C240 M5 server. Sixteen Cisco UCS C240 M5 servers are used in Master rack configurations.

Figure 21 Cisco UCS C240 M5 and 6300 Series Fabric Interconnect Port Connectivity



For information about physical connectivity and single-wire management, go to:

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 about physical connectivity illustrations and cluster setup, go to:

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.

Hortonworks Data Platform (HDP 3.1.0)

The Hortonworks Data Platform supported is HDP 3.1.0. For more information, go to: <http://www.hortonworks.com>.

Red Hat Enterprise Linux (RHEL)

The operating system supported is Red Hat Enterprise Linux 7.6. For more information, go to: <http://www.redhat.com>.

Software Versions

The software versions tested and validated in this document are shown in Table 6.

Table 6 Software Versions

Layer	Component	Version or Release
Compute	Cisco UCS C240 M5	C240M5.4.0.2a
	Cisco UCS C480 ML M5	
Network	Cisco UCS 6332	UCS 4.0(4b)
	Cisco UCS VIC1387 Firmware	4.3(2a)

Layer	Component	Version or Release
	Cisco UCS VIC1387 Driver	3.1.137.5
Storage	SAS Expander	65.02.15.00
	Cisco 12G Modular Raid controller	50.6.0-1952
Software	Red Hat Enterprise Linux Server	7.6
	Cisco UCS Manager	4.0(4b)
	HDP	3.1.0
	Docker	1.13.1
	Ansible	2.4.6.0
	Nvidia-docker	1.0.1
GPU	CUDA	10.1
	NVIDIA GPU Driver	418.67



The latest drivers can be downloaded from this link:

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



The latest supported RAID controller driver is already included with the RHEL 7.6 operating system.



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 the details to configure a fully redundant, highly available Cisco UCS 6332 fabric configuration. The following is the high-level workflow to setup Cisco UCS:

- 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 and uplink ports
- Start discovery process
- Create pools and polices for service profile template

- Create Service Profile template
- Create service profile for each server from service profile template
- Associate Service Profiles to servers5

Perform 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

To configure Fabric Interconnect A, follow these steps:

1. Connect to the console port on the first Cisco UCS 6332 Fabric Interconnect.

At the prompt to enter the configuration method, enter **console** to continue.
If asked to either perform a new setup or restore from backup, enter **setup** to continue.
Enter **y** to continue to set up a new Fabric Interconnect.
Enter **y** to enforce strong passwords.

2. Enter the password for the admin user.
3. Enter the same password again to confirm the password for the admin user.

When asked if this fabric interconnect is part of a cluster, answer **y** to continue.
Enter **A** for the switch fabric.

4. Enter the cluster name for the system name.
5. Enter the Mgmt IPv4 address.
6. Enter the Mgmt IPv4 netmask.
7. Enter the IPv4 address of the default gateway.
8. Enter the cluster IPv4 address.

To configure DNS, answer **y**.

9. Enter the DNS IPv4 address.

Answer **y** to set up the default domain name.

10. Enter the default domain name.

Review the settings that were printed to the console, and if they are correct, answer **yes** to save the configuration.

11. Wait for the login prompt to make sure the configuration has been saved.

Configure Fabric Interconnect B

To configure Fabric Interconnect B, follow these steps:

1. Connect to the console port on the second Cisco UCS 6332 Fabric Interconnect.

When prompted to enter the configuration method, enter console to continue. The installer detects the presence of the partner Fabric Interconnect and adds this fabric interconnect to the cluster. Enter y to continue the installation.

2. Enter the admin password that was configured for the first Fabric Interconnect.
3. Enter the Mgmt IP address.
4. Answer yes to save the configuration.
5. Wait for the login prompt to confirm that the configuration has been saved.

For more information about configuring Cisco UCS 6332 Series Fabric Interconnect, go 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

Log Into Cisco UCS Manager

To log into Cisco UCS Manager, follow these 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.

Upgrade Cisco UCS Manager Software to Version 4.0(4b)

This document assumes the use of UCS 4.0(4b). Refer to the [Cisco UCS 4.0 Release](#) (upgrade Cisco UCS Manager software and Cisco UCS 6332 Fabric Interconnect software to version 4.0(4b)). Also, make sure the Cisco UCS C-Series version 4.0(4b) software bundles are installed on the Fabric Interconnects.



Upgrading Cisco UCS firmware is beyond the scope of this document. However for complete Cisco UCS Install and Upgrade Guides, go to: <https://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-manager/products-installation-guides-list.html>

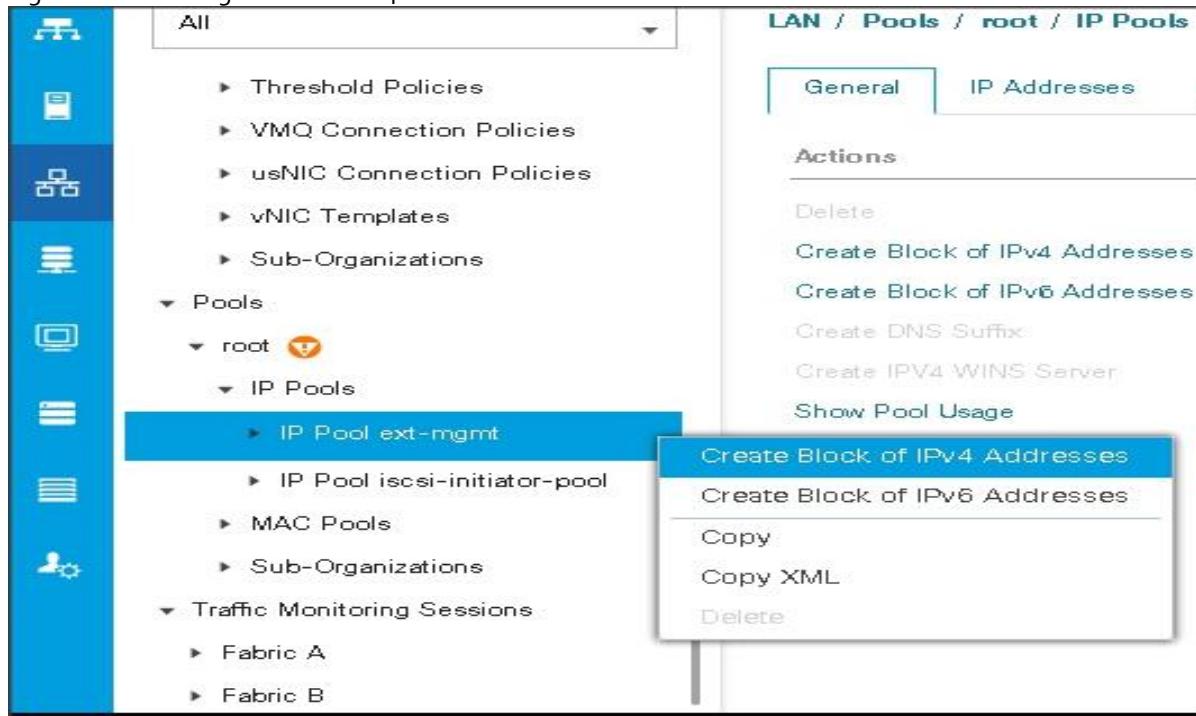
Add a Block of IP Addresses for KVM Access

To create a block of KVM IP addresses for server access in the Cisco UCS environment, follow these 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 22 Adding a Block of IPv4 Addresses for KVM Access Part 1



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

Figure 23 Adding Block of IPv4 Addresses for KVM Access Part 2

The dialog box has the title 'Create Block of IPv4 Addresses'. It contains the following fields:

- From :
- Size :
- Subnet Mask :
- Default Gateway :
- Primary DNS :
- Secondary DNS :

6. Click OK to create the IP block.
7. Click OK in the message box.

Enable Uplink Ports

To enable uplinks ports, follow these 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 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 24 Enabling Uplink Ports Part1

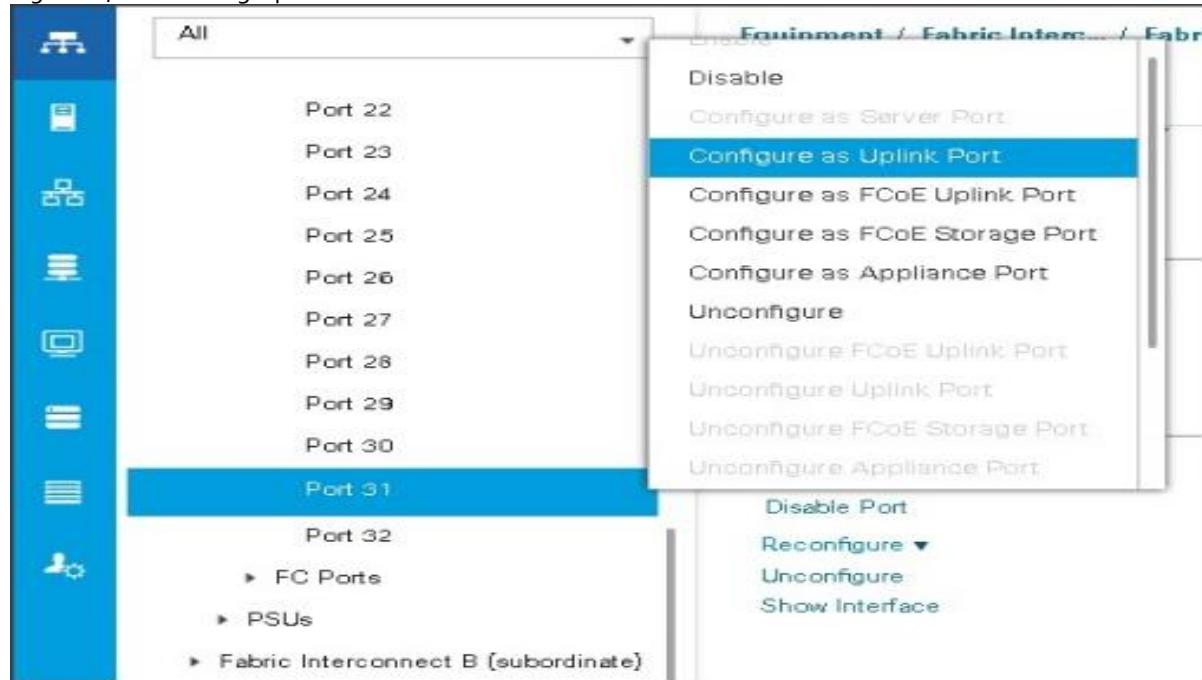


Figure 25 Enabling Uplink Ports Part2

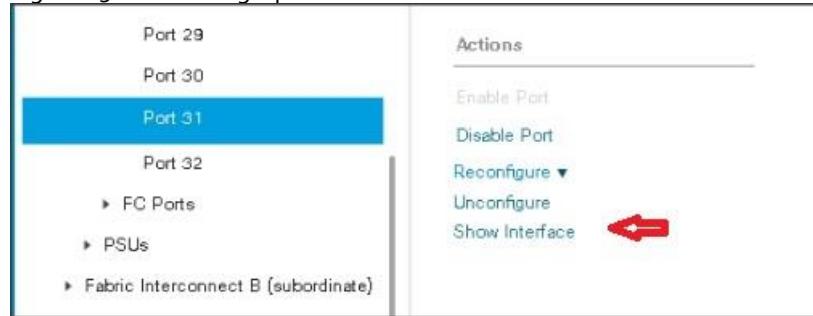
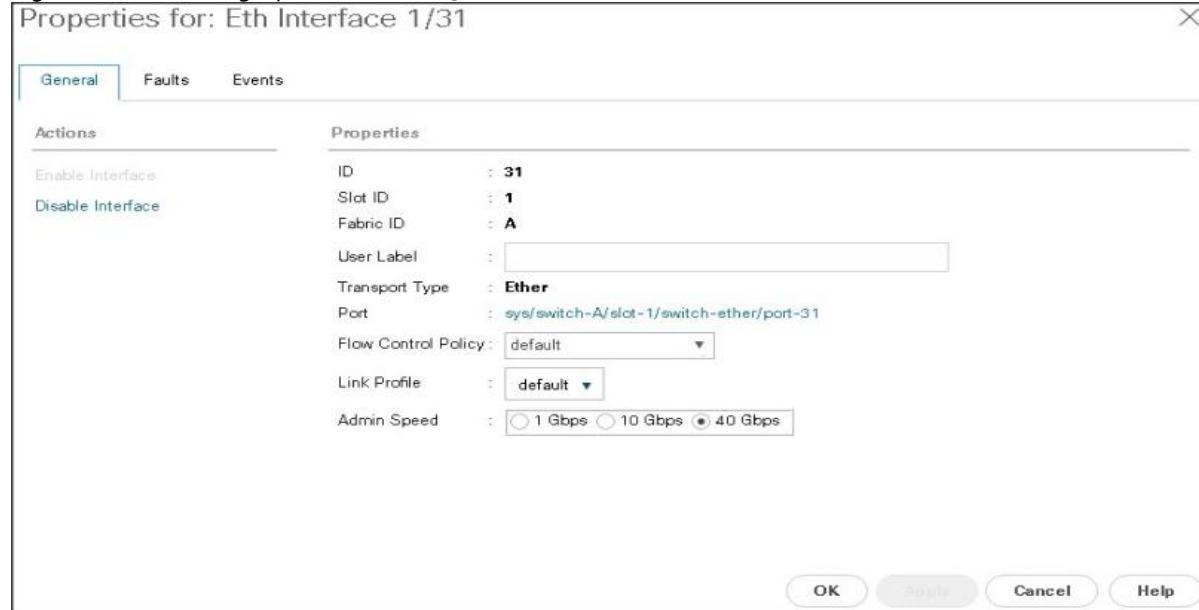


Figure 26 Enabling Uplink Ports Part 3



Configure VLANs

VLANs are configured as shown in Table 7.

Table 7 VLAN Configurations

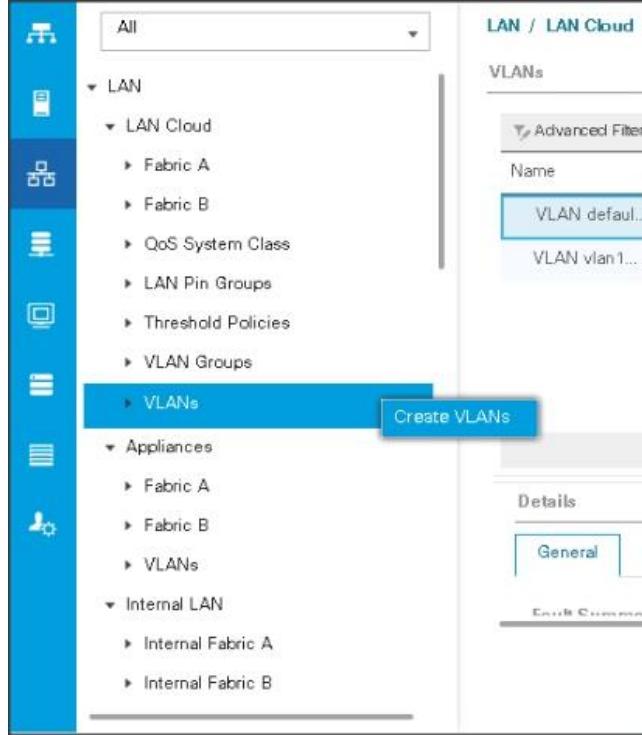
VLAN	NIC Port	Function
VLAN13	eth0	Data

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, follow these 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 27 Creating a VLAN



5. Enter **vlan13** for the VLAN Name.
6. Keep multicast policy as <not set>.
7. Select Common/Global for **vlan16**.
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 28 Creating VLAN for Data

Create VLANs

VLAN Name/Prefix :

Multicast Policy Name : [Create Multicast Policy](#)

Common/Global Fabric A Fabric B Both Fabrics Configured Differently

You are creating global VLANs that map to the same VLAN IDs in all available fabrics.
Enter the range of VLAN IDs.(e.g. "2009-2019", "29,35,40-45", "23", "23,34-45")

VLAN IDs :

Sharing Type : None Primary Isolated Community

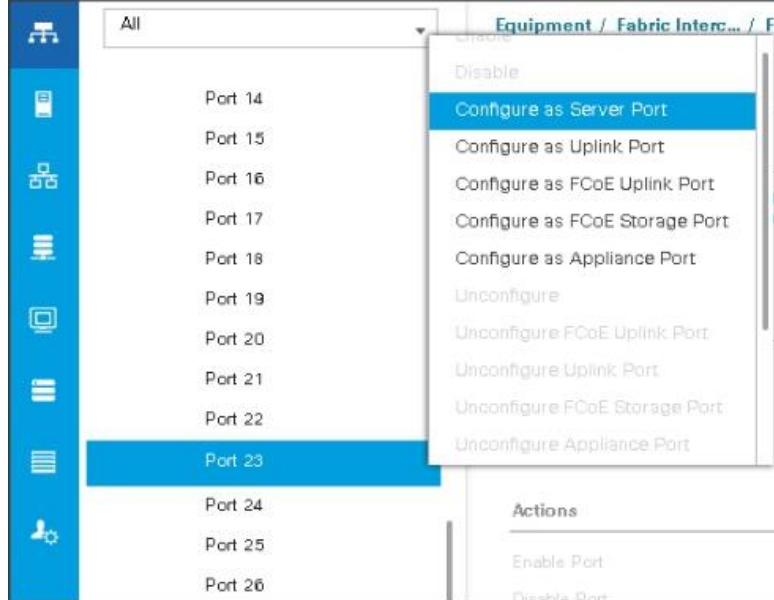
11. Click OK and then click Finish.

Enable Server Ports

To enable server ports, follow these 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 29 Enabling Server Ports



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

Figure 30 Ports Status after the Server Discover

The screenshot shows the Cisco UCS Manager interface. The left sidebar navigation tree is expanded to show 'Fabric Interconnect A (primary)', 'Fixed Module', and 'Ethernet Ports'. Under 'Ethernet Ports', 'Port 1' through 'Port 9' are listed. The right pane displays a table titled 'Equipment / Fabric Interconnect... / Fixed Module / Ethernet Ports' with columns for Slot, Aggr. Port ID, Port ID, MAC, If Role, If Type, Overall Status, and Admin State. All ports are listed as 'Up' and 'Enabled'. The top navigation bar shows icons for Home, Logout, Refresh, Help, and Settings.

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

Create Pools for Service Profile Templates

Create an Organization

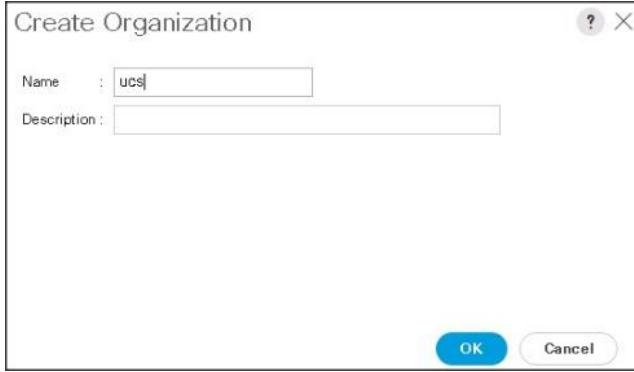
Organizations are used as a means to arrange and restrict access to various groups within the IT organization, thereby enabling multi-tenancy of the compute resources. This document does not assume the use of Organizations; however, the necessary steps are provided for future reference.

To configure an organization within the Cisco UCS Manager GUI, follow these 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. The left sidebar shows a list of servers: 'uce-11', 'uce-12', 'uce-13', 'uce-14', 'uce-15', 'uce-16', and 'uce-17'. The right pane displays 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 'OK' status and associated with 'sys/rack-unit-1', 'sys/rack-unit-10', and 'sys/rack-unit-11' respectively. A red arrow points to the 'Create Organization' option in the context menu that appears when clicking the Quick Action icon.

Name	User Label	Overall Status	Assoc State	Server
Service Profile ucs-1		OK	Associated	sys/rack-unit-1
Service Profile ucs-10		OK	Associated	sys/rack-unit-10
Service Profile ucs-11		OK	Associated	sys/rack-unit-11



Create MAC Address Pools

To create MAC address pools, follow these steps:

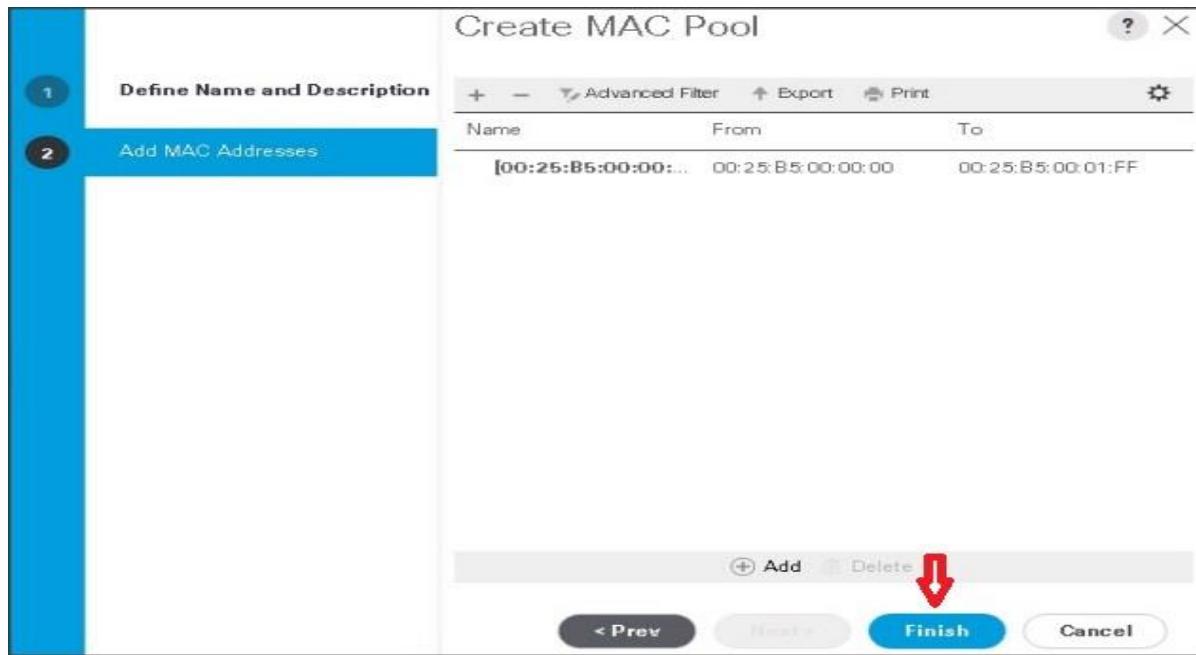
1. Select the LAN tab on the left of the window.
2. Select Pools > root > MAC Pools
3. Right-click MAC Pools under the root organization.
4. Select Create MAC Pool to create the MAC address pool. Enter ucs for the name of the MAC pool.
5. (Optional) Enter a description of the MAC pool.
6. Select Assignment 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 vSphere Web Client interface. On the left, a navigation sidebar lists various categories like All, vNIC Templates, Sub-Organizations, Pools, IP Pools, MAC Pools, Sub-Organizations, Traffic Monitoring Sessions, Fabric A, and Fabric B. The 'MAC Pools' item under 'Pools' is highlighted with a blue box. To the right, the main pane displays the 'LAN / Pools / root / MAC Pools' section. It shows two existing MAC pools: 'MAC Pool default' and 'MAC Pool ucs'. A 'Create MAC Pool' button is visible. Below this, a 'Create MAC Pool' dialog box is open. The first tab, 'Define Name and Description', is selected. It contains fields for 'Name' (set to 'ucs'), 'Description' (empty), and 'Assignment Order' (radio button selected for 'Sequential'). The second tab, 'Add MAC Addresses', is partially visible. At the bottom of the dialog are buttons for 'Previous', 'Next >', 'Finish', and 'Cancel'.

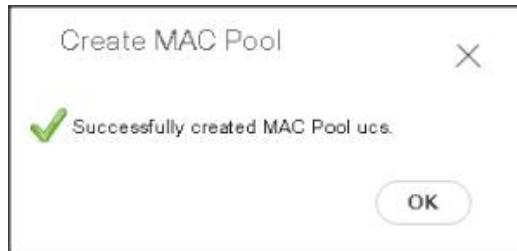
Figure 31 Specifying first MAC Address and Size

This dialog box is titled 'Create a Block of MAC Addresses'. It has fields for 'First MAC Address' (00:25:B5:00:00:00) and 'Size' (512). A note at the bottom states: 'To ensure uniqueness of MACs in the LAN fabric, you are strongly encouraged to use the following MAC prefix: 00:25:B5:xxxx:xxxx'. At the bottom are 'OK' and 'Cancel' buttons.

12. Click Finish.



13. When the message box displays, click OK.

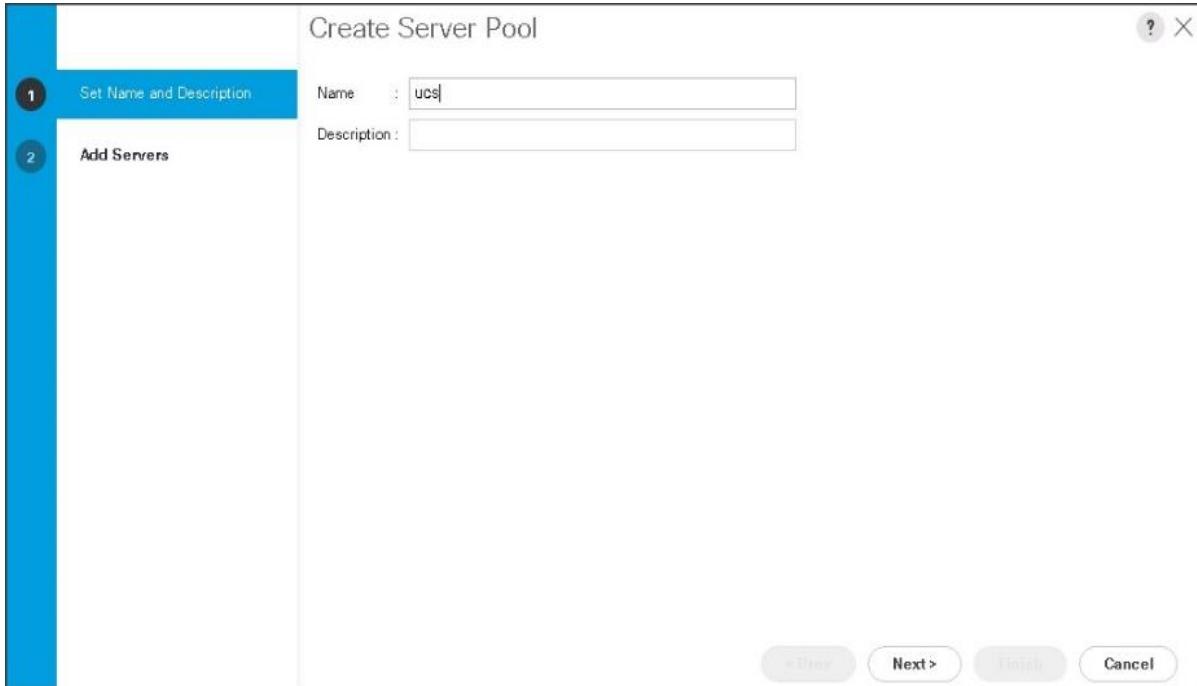


Create a Server Pool

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, follow these steps:

1. Select the Servers tab in the left pane in the Cisco 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.

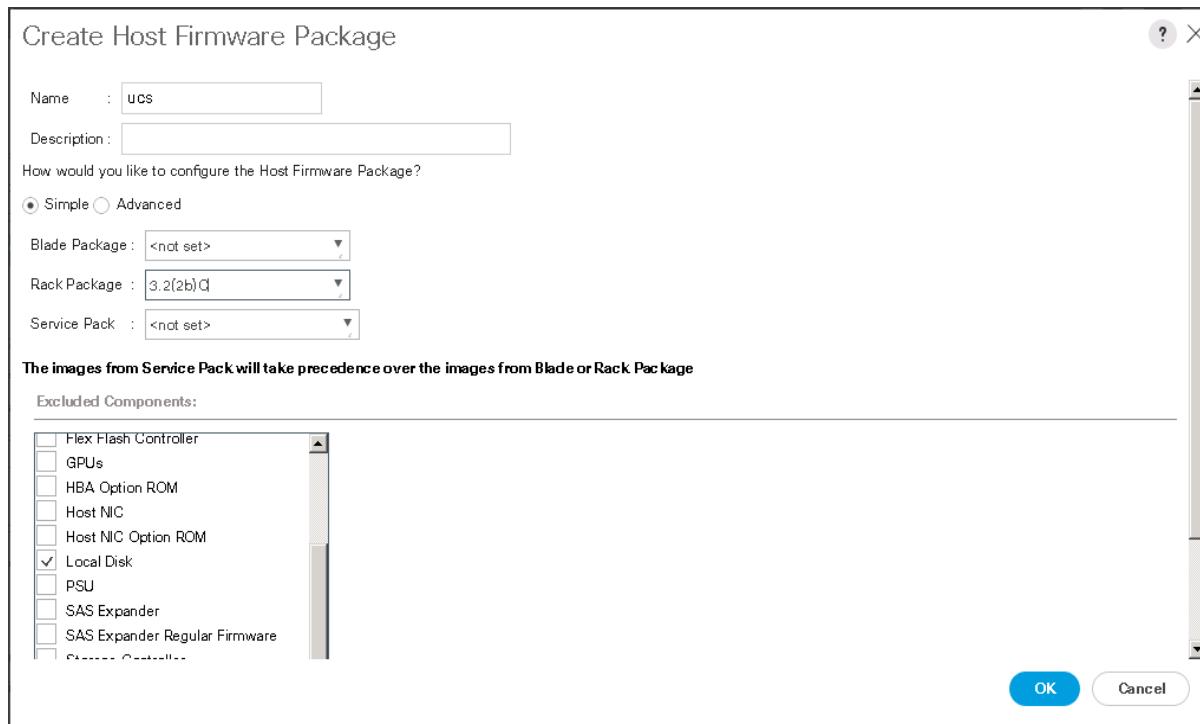
Create Policies for Service Profile Templates

Create 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, follow these steps:

1. Select the Servers tab in the left pane in the Cisco 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.



Create QoS Policies

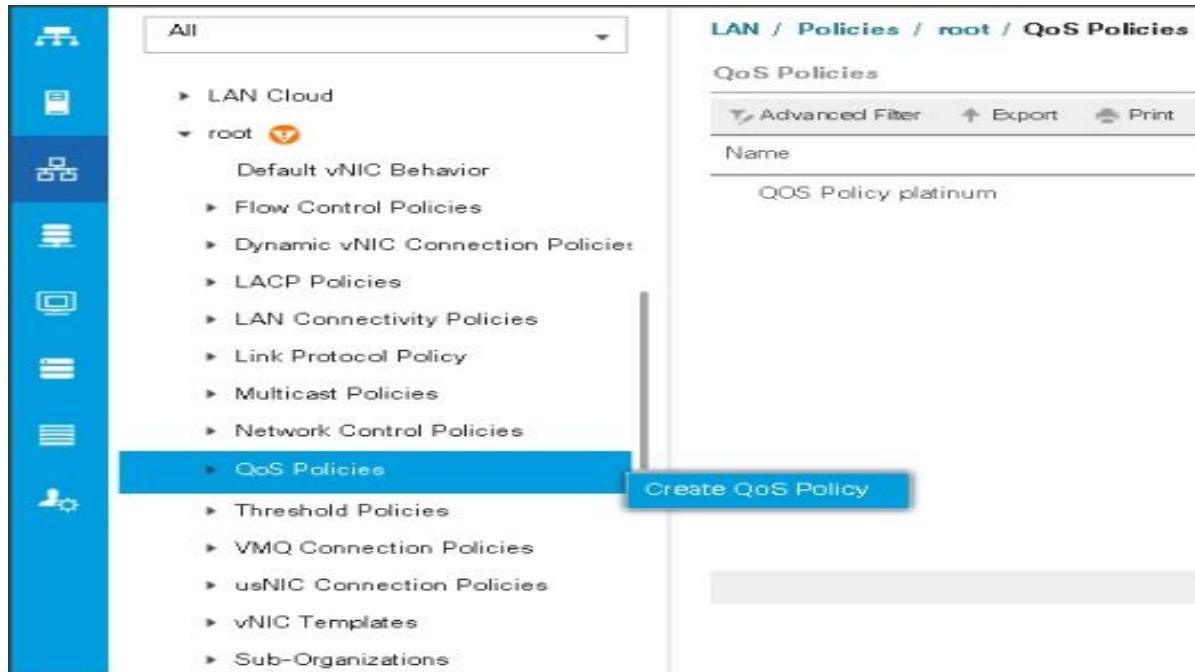
To create the QoS policy for a given server configuration using the Cisco UCS Manager GUI, follow these steps:

Platinum Policy

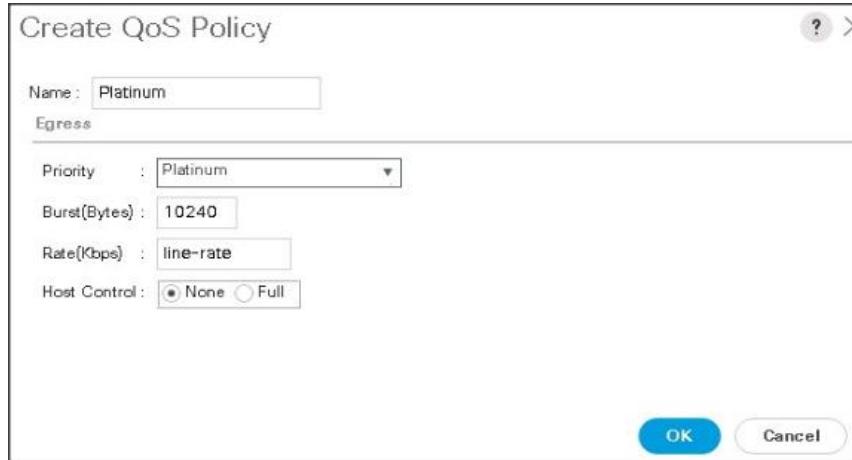
To create the Platinum policy, follow these steps:

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.



Set Jumbo Frames

To set Jumbo frames and enable QoS, follow these steps:

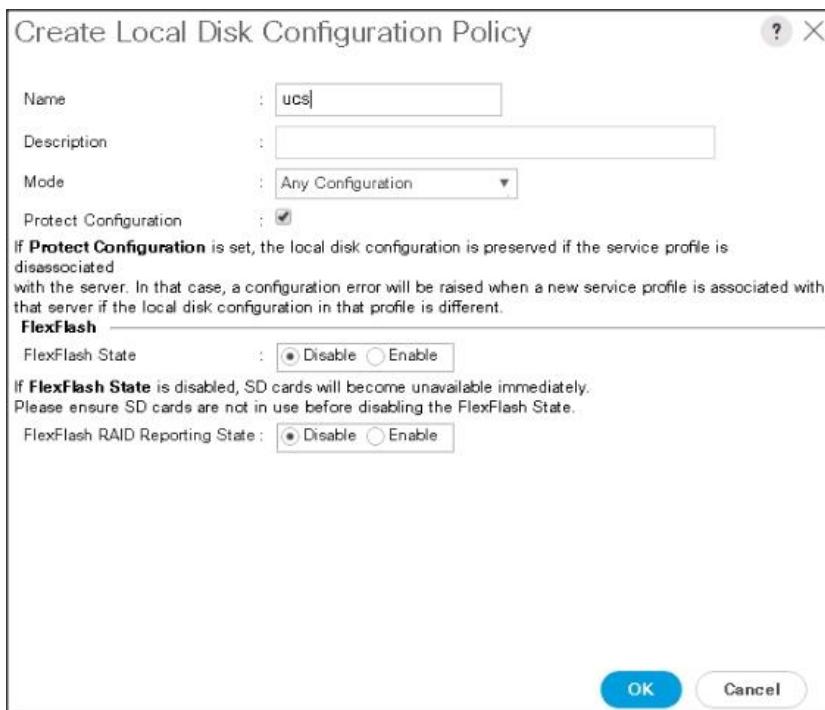
1. Select the LAN tab in the left pane in the Cisco UCS Manager GUI.
2. Select LAN Cloud > QoS System Class.
3. In the right pane, select the General tab
4. In the Platinum row, enter 9216 for MTU.
5. Check the Enabled Check box next to Platinum.
6. In the Best Effort row, select none for weight.
7. In the Fiber Channel row, select none for weight.
8. Click Save Changes.
9. Click OK.

QoS System Class		Priority	Enabled CoS	Packet Drop	Weight	Weight (%)	MTU
LAN Pin Groups	Platinum	<input checked="" type="checkbox"/>	5	<input type="checkbox"/>	10	N/A	9216
Threshold Policies	Gold	<input type="checkbox"/>	4	<input checked="" type="checkbox"/>	9	N/A	normal
VLAN Groups	Silver	<input type="checkbox"/>	2	<input checked="" type="checkbox"/>	8	N/A	normal
VLANs	Bronze	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	7	N/A	normal
Appliances	Best Effort	<input checked="" type="checkbox"/>	Any	<input checked="" type="checkbox"/>	none	50	normal
Fabric A	Fibre	<input checked="" type="checkbox"/>	3	<input type="checkbox"/>	none	50	normal
Fabric B							
VLANs							
Internal LAN							
Internal Fabric A							
Internal Fabric B							
Threshold Policies							

Create the Local Disk Configuration Policy

To create local disk configuration in the Cisco UCS Manager GUI, follow these 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.



Create the 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.



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, follow these 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.

Processor 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

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

Advanced Filter Export 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

Advanced Filter Export 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

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

UCS Manager

Servers / Policies / root / BIOS Policies / ucs

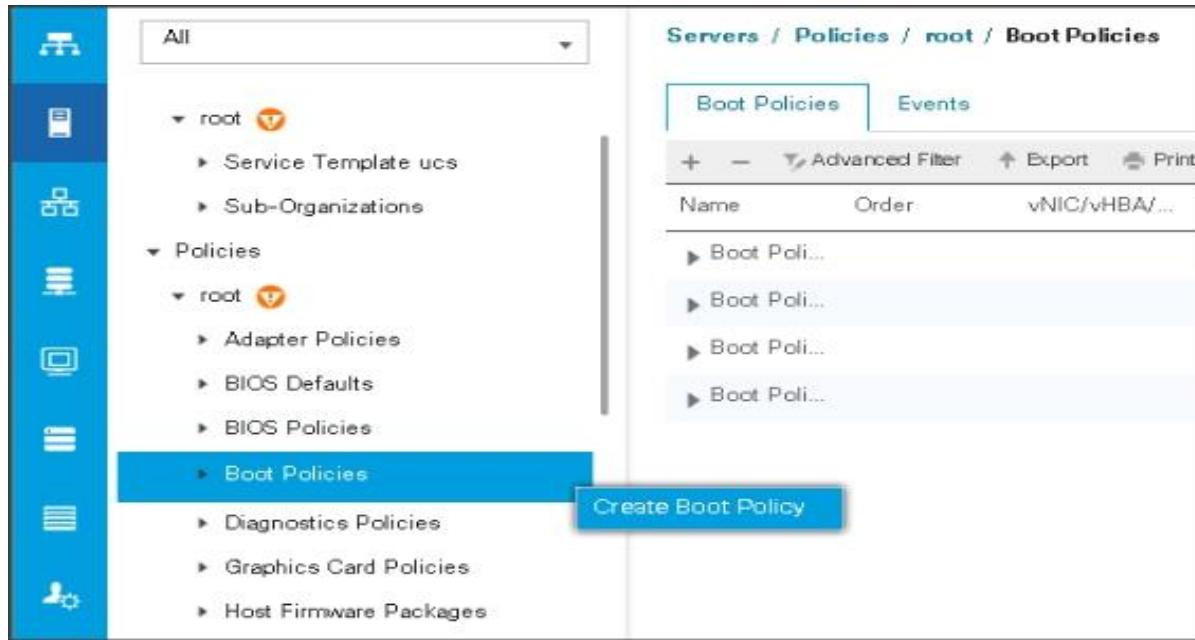
Main Advanced Boot Options Server Management Events

BIOS Setting	Value
DDR3 Voltage Selection	Platform Default
DRAM Refresh Rate	1x
LV DDR Mode	Platform Default
Mirroring Mode	Platform Default
NUMA optimized	Platform Default
Memory RAS configuration	Maximum Performance

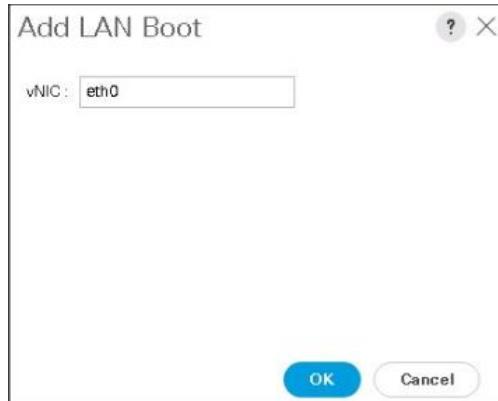
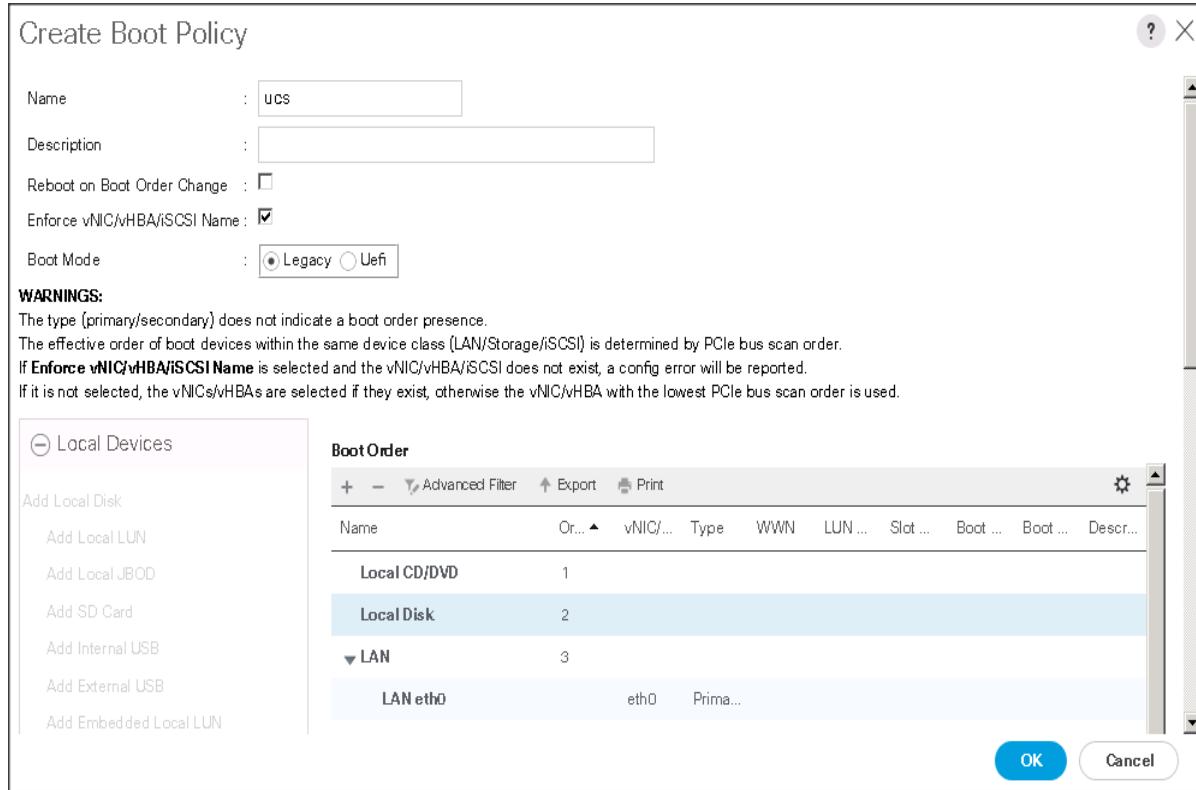
Create the Boot Policy

To create boot policies within the Cisco UCS Manager GUI, follow these steps:

1. Select the Servers tab in the left pane in the Cisco 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.



Create the Power Control Policy

To create Power Control policies within the Cisco UCS Manager GUI, follow these 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.

The screenshot shows the Cisco UCS Manager interface. On the left is a navigation sidebar with various icons. The main area has a title bar 'Servers / Policies / root / Power Control Policies'. Below this is a toolbar with 'Power Control Policies' selected, along with 'Events', 'Advanced Filter', 'Export', and 'Print' buttons. A search bar labeled 'Name' contains 'ucs'. The main content area lists several policy types: Boot Policies, Diagnostics Policies, Graphics Card Policies, Host Firmware Packages, IPMI Access Profiles, KVM Management Policies, Local Disk Config Policies, Maintenance Policies, Management Firmware Packages, Memory Policy, Power Control Policies, Power Sync Policies, Scrub Policies, and Serial over LAN Policies. A blue callout box points to the 'Create Power Control Policy' button at the bottom of this list.

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.

Create Power Control Policy

Name :	ucs
Description :	
Fan Speed Policy:	Performance

Power Capping

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

No Cap cap

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

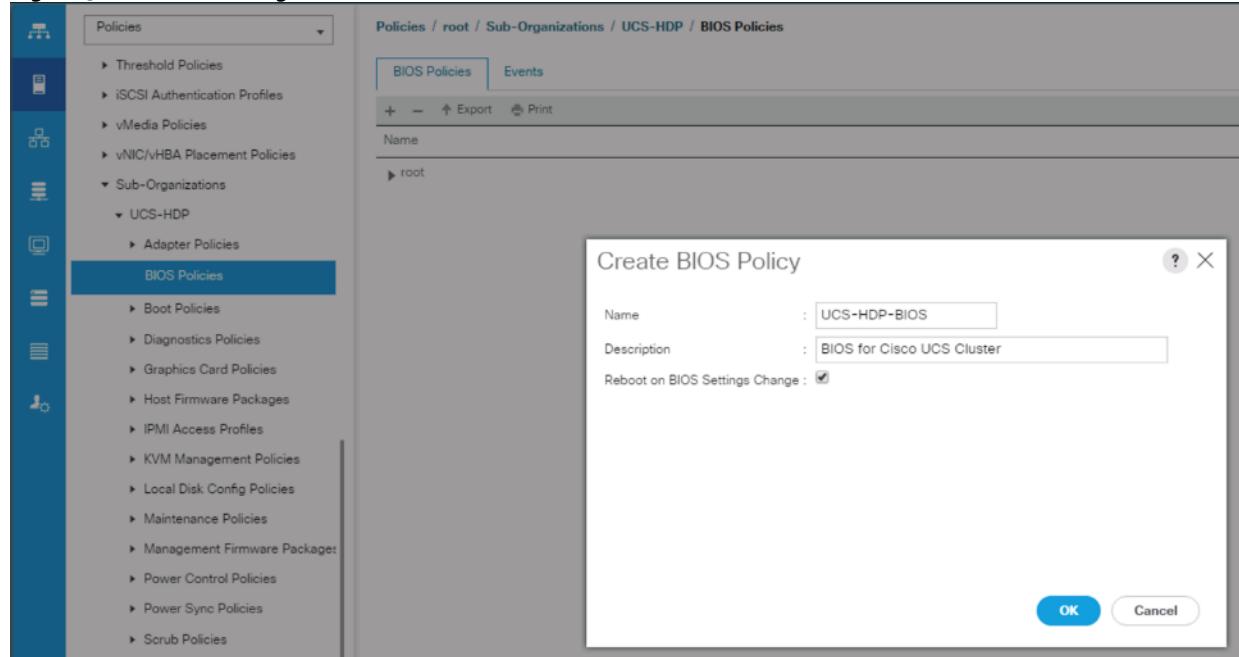
OK **Cancel**

Create Server BIOS Policy

To create a server BIOS policy for the Cisco UCS environment, follow these steps:

1. In Cisco UCS Manager, click the Servers tab in the navigation pane.
2. Select Policies > root > Sub-Organization > UCS-HDP > BIOS Policies.
3. Right-click BIOS Policies.
4. Select Create BIOS Policy.
5. Enter C240M5-BIOS as the BIOS policy name.

Figure 32 BIOS Configuration



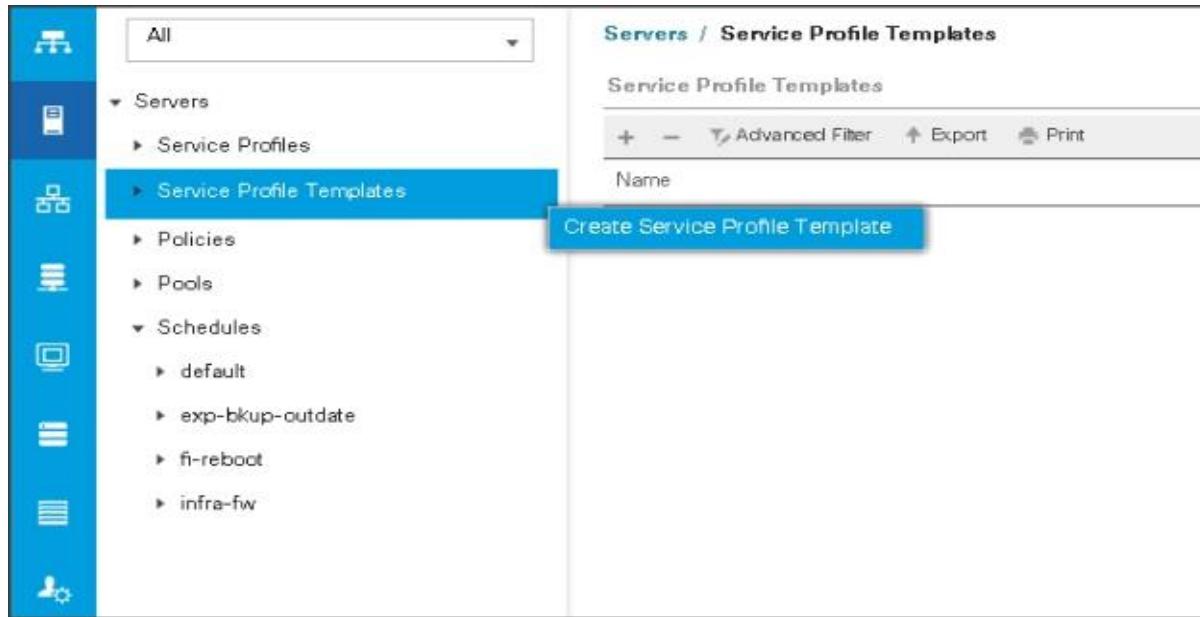
Policies / root / Sub-Organizations / TPC-BDA / BIOS Policies / BDA-BIOS									
Main	Advanced	Boot Options	Server Management	Events					
Processor	Intel Directed IO	RAS Memory	Serial Port	USB	PCI	QPI	LOM and PCIe Slots	Trusted Platform	Graphics Configuration
T Advanced Filter Export Print									
BIOS Setting									Value
Altitude	Platform Default								
CPU Hardware Power Management	Platform Default								
Boot Performance Mode	Platform Default								
CPU Performance	Enterprise								
Core Multi Processing	All								
DCPMM Firmware Downgrade	Platform Default								
DRAM Clock Throttling	Performance								
Direct Cache Access	Enabled								
Energy Performance Tuning	Platform Default								
Enhanced Intel SpeedStep Tech	Enabled								
Execute Disable Bit	Platform Default								
Frequency Floor Override	Platform Default								
Intel HyperThreading Tech	Enabled								
Energy Efficient Turbo	Platform Default								
Intel Turbo Boost Tech	Enabled								
Intel Virtualization Technology	Disabled								
Intel Speed Select	Platform Default								
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								
Max Variable MTRR Setting	Platform Default								
P STATE Coordination	HW ALL								
Package C State Limit	Platform Default								
Autonomous Core C-state	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								

		Value																										
BIOS Setting																												
Energy Performance		Performance																										
ProcessorEppProfile		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																										
Core Performance Boost		Platform Default																										
Downcore control		Platform Default																										
Global C-state Control		Platform Default																										
L1 Stream HW Prefetcher		Platform Default																										
L2 Stream HW Prefetcher		Platform Default																										
Determinism Slider		Platform Default																										
IOMMU		Platform Default																										
Bank Group Swap		Platform Default																										
Bank Group Swap		Platform Default																										
Chipselect Interleaving		Platform Default																										
Configurable TDP Control		Platform Default																										
AMD Memory Interleaving		Platform Default																										
AMD Memory Interleaving Size		Platform Default																										
SME		Platform Default																										
SMT Mode		Platform Default																										
SVM Mode		Platform Default																										
Demand Scrub		Enabled																										
Patrol Scrub		Enabled																										
Workload Configuration		Platform Default																										
Policies / root / Sub-Organizations / TPC-BDA / BIOS Policies / BDA-BIOS																												
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>Main</th> <th>Advanced</th> <th>Boot Options</th> <th>Server Management</th> <th>Events</th> <th colspan="8"></th> </tr> <tr> <th>Processor</th> <th>Intel Directed IO</th> <th>RAS Memory</th> <th>Serial Port</th> <th>USB</th> <th>PCI</th> <th>QPI</th> <th>LOM and PCIe Slots</th> <th>Trusted Platform</th> <th colspan="4">Graphics Configuration</th> </tr> </table>			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			
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																			
<table border="1"> <thead> <tr> <th colspan="2"></th> <th style="text-align: right;">Value</th> </tr> </thead> <tbody> <tr> <td colspan="2">BIOS Setting</td> <td></td> </tr> <tr> <td>DDR3 Voltage Selection</td> <td></td> <td>Platform Default</td> </tr> <tr> <td>DRAM Refresh Rate</td> <td></td> <td>Platform Default</td> </tr> <tr> <td>LV DDR Mode</td> <td></td> <td>Platform Default</td> </tr> <tr> <td>Mirroring Mode</td> <td></td> <td>Platform Default</td> </tr> <tr> <td>NUMA optimized</td> <td></td> <td>Platform Default</td> </tr> <tr> <td>Memory RAS configuration</td> <td></td> <td>Maximum Performance</td> </tr> </tbody> </table>					Value	BIOS Setting			DDR3 Voltage Selection		Platform Default	DRAM Refresh Rate		Platform Default	LV DDR Mode		Platform Default	Mirroring Mode		Platform Default	NUMA optimized		Platform Default	Memory RAS configuration		Maximum Performance		
		Value																										
BIOS Setting																												
DDR3 Voltage Selection		Platform Default																										
DRAM Refresh Rate		Platform Default																										
LV DDR Mode		Platform Default																										
Mirroring Mode		Platform Default																										
NUMA optimized		Platform Default																										
Memory RAS configuration		Maximum Performance																										

Create the Service Profile Template

To create the Service Profile Template, follow these 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, follow these 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.

Create Service Profile Template

Identify Service Profile Template

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

Storage Provisioning
Name : `ucs`

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

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

SAN Connectivity
Type : Initial Template Updating Template

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

Zoning
UUID

vNIC/vHBA Placement
UUID Assignment: `Hardware Default`

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

vMedia Policy

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

Maintenance Policy

Server Assignment

Buttons: < Prev, Next >, Finish, Cancel

Configure the Storage Provisioning for the Template

To configure storage policies, follow these 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		

Create Local Disk Configuration Policy

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**

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

Configure Network Settings for the Template

To configure the network settings for the templates, follow these steps:

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



4. The Create vNIC window displays. Name the vNIC as eth0.
5. Select `ucs` in the Mac Address Assignment pool.
6. 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

Name :

MAC Address

MAC Address Assignment:

[Create MAC Pool](#)
The MAC address will be automatically assigned from the selected pool.

Use vNIC Template :

Fabric ID : Fabric A Fabric B Enable Failover

VLAN in LAN cloud will take the precedence over the Appliance Cloud when there is a name clash.

	VLANs	VLAN Groups
<input type="checkbox"/>	default	<input type="radio"/>
<input checked="" type="checkbox"/>	vlan13_data	<input type="radio"/>
<input type="checkbox"/>	vlan14	<input type="radio"/>

OK **Cancel**

Create vNIC

CDN Source : vNIC Name User Defined

MTU :

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 : Create LAN Pin Group

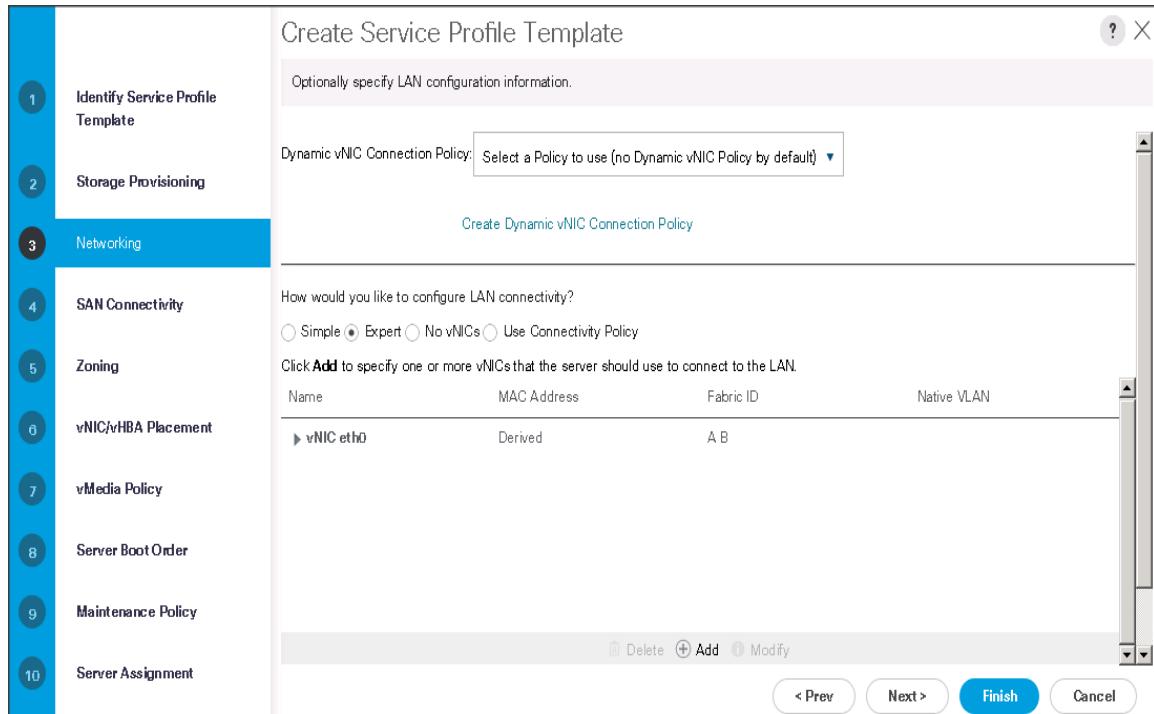
+ Operational Parameters

Adapter Performance Profile

Adapter Policy :	<input type="text" value="Linux"/>	Create Ethernet Adapter Policy
QoS Policy :	<input type="text" value="platinum"/>	Create QoS Policy
Network Control Policy :	<input type="text" value="default"/>	Create Network Control Policy

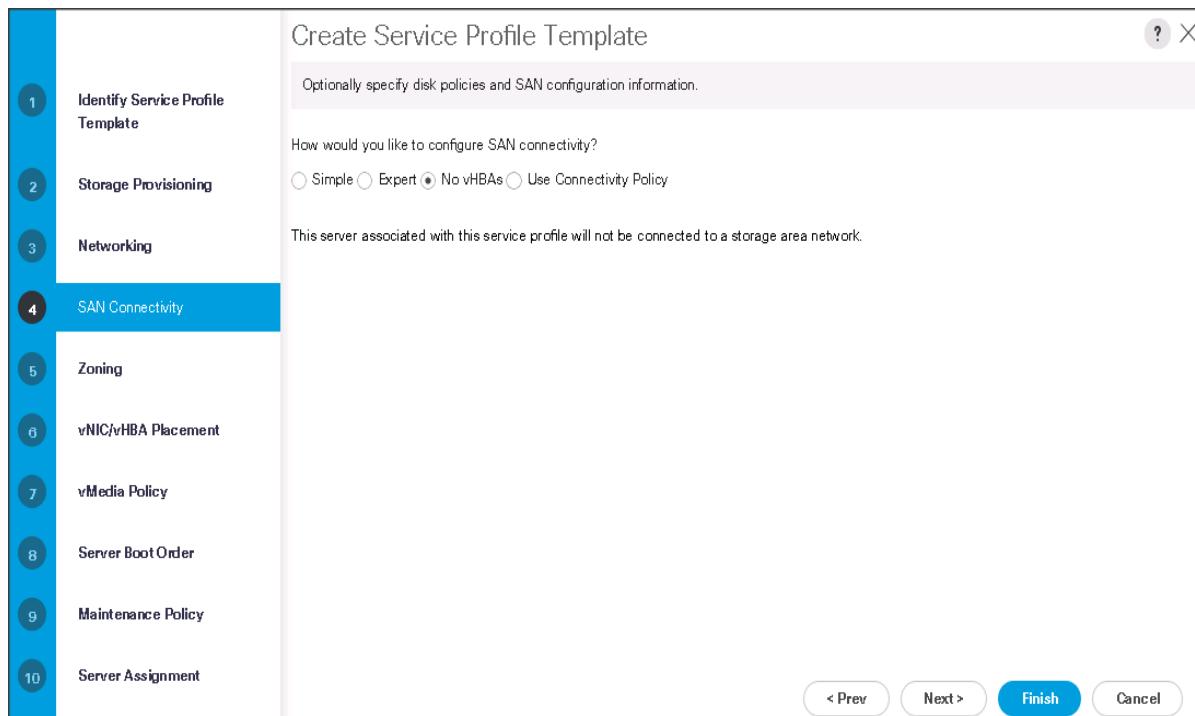
Connection Policies

OK **Cancel**

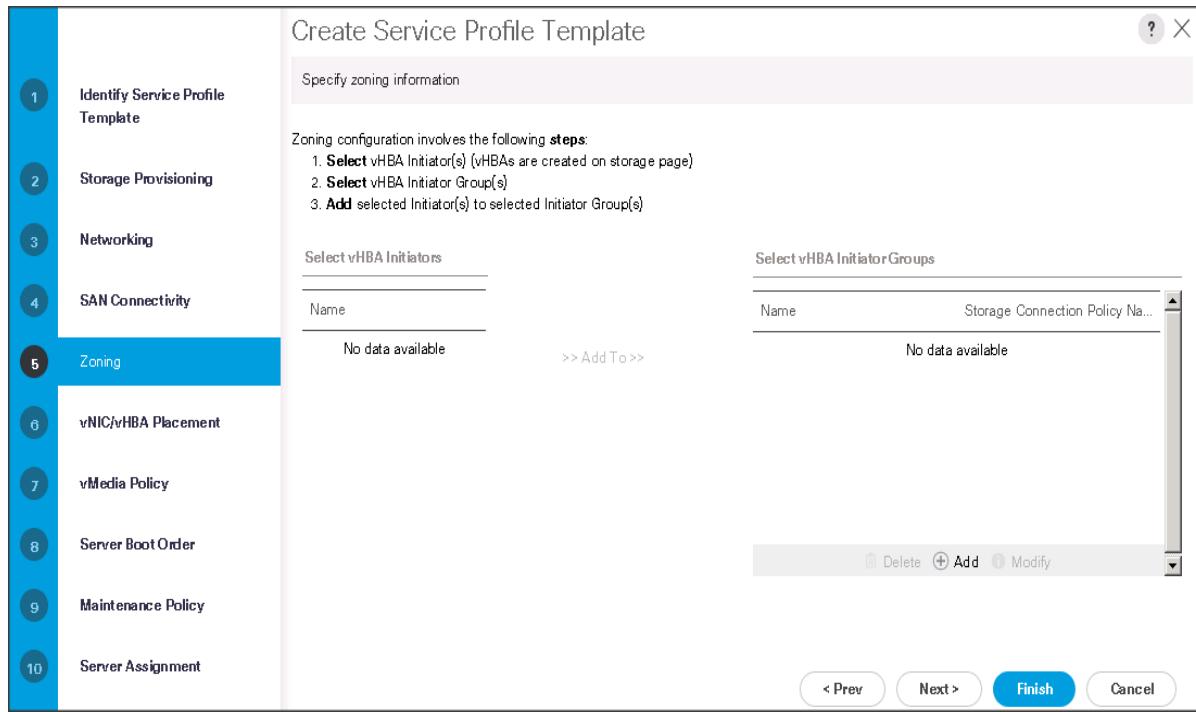


Optionally, Network Bonding can be setup on the vNICs for each host for redundancy as well as for increased throughput.

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.

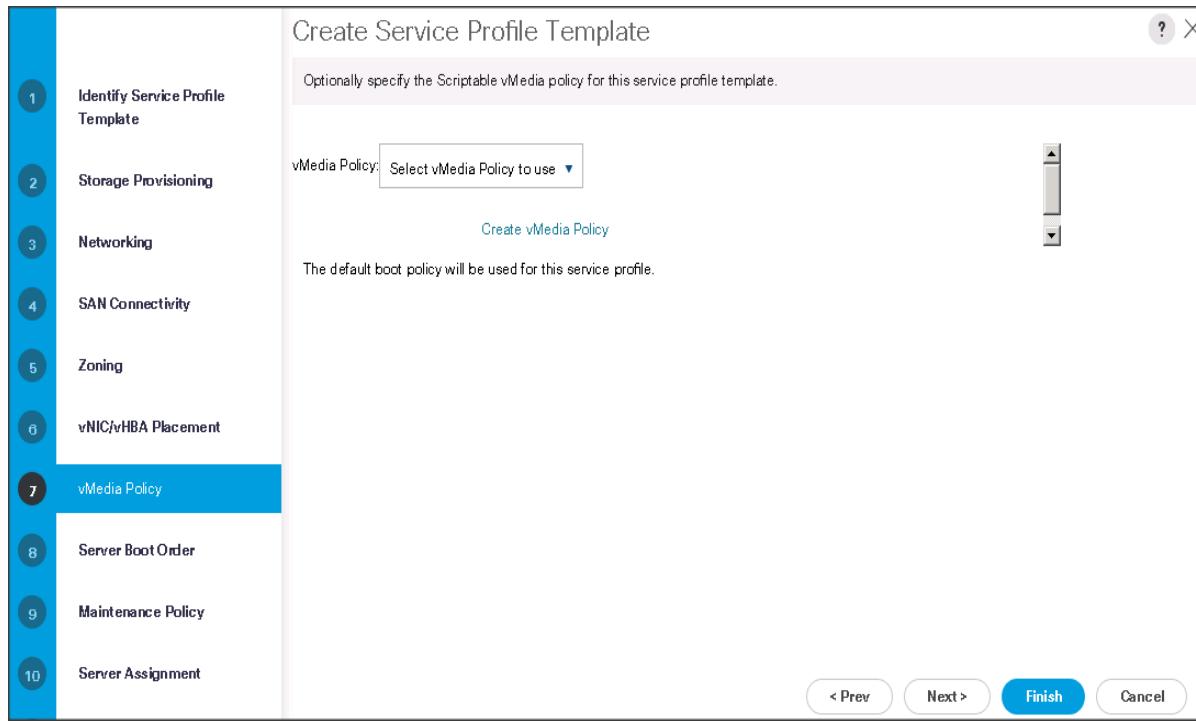


17. Click Next to configure vMedia Policy.

Configure the vMedia Policy for the Template

To configure the vMedia policy for the template, follow these steps:

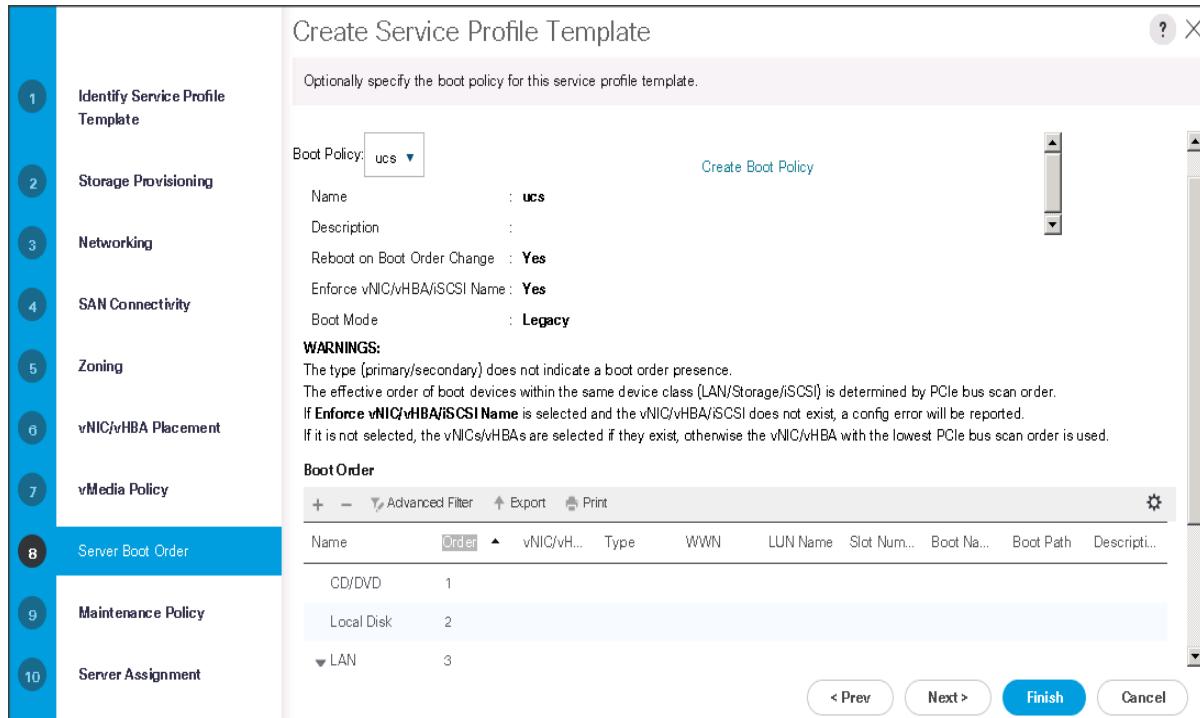
1. Click **Next** once the vMedia Policy window appears to go to the next section.



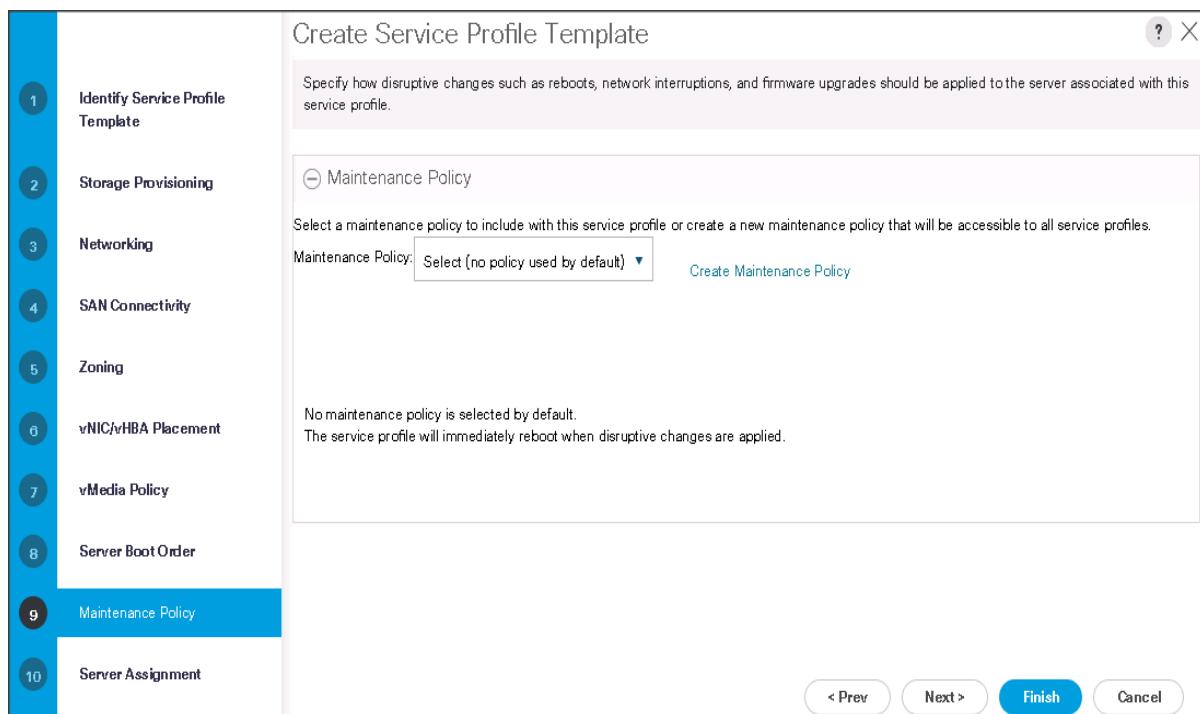
Configure the Server Boot Order for the Template

To set the boot order for the servers, follow these 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.



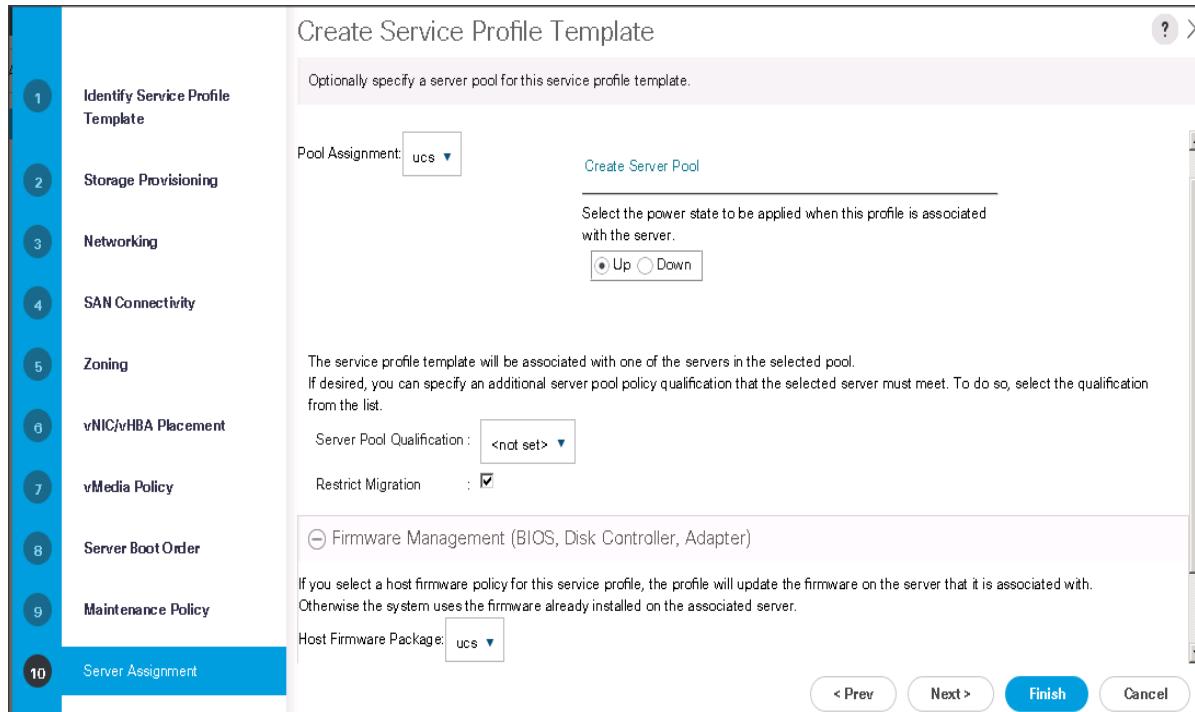
6. In the Maintenance Policy window, apply the maintenance policy.
7. Keep the Maintenance policy at no policy used by default. Click Next to continue to the next section.



Configure the Server Assignment for the Template

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

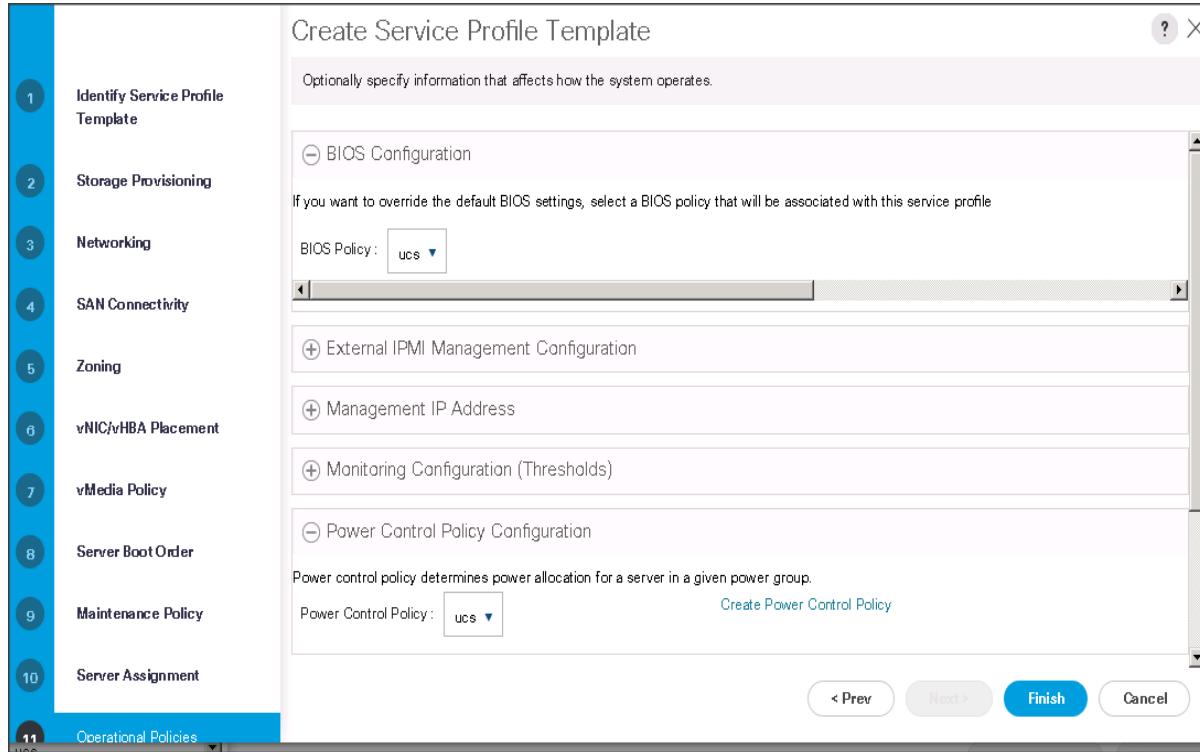
1. Select ucs for the Pool Assignment field.
2. Select the power state to be Up.
3. Keep the Server Pool Qualification field set to <not set>.
4. Check the Restrict Migration check box.
5. Select ucs in Host Firmware Package.



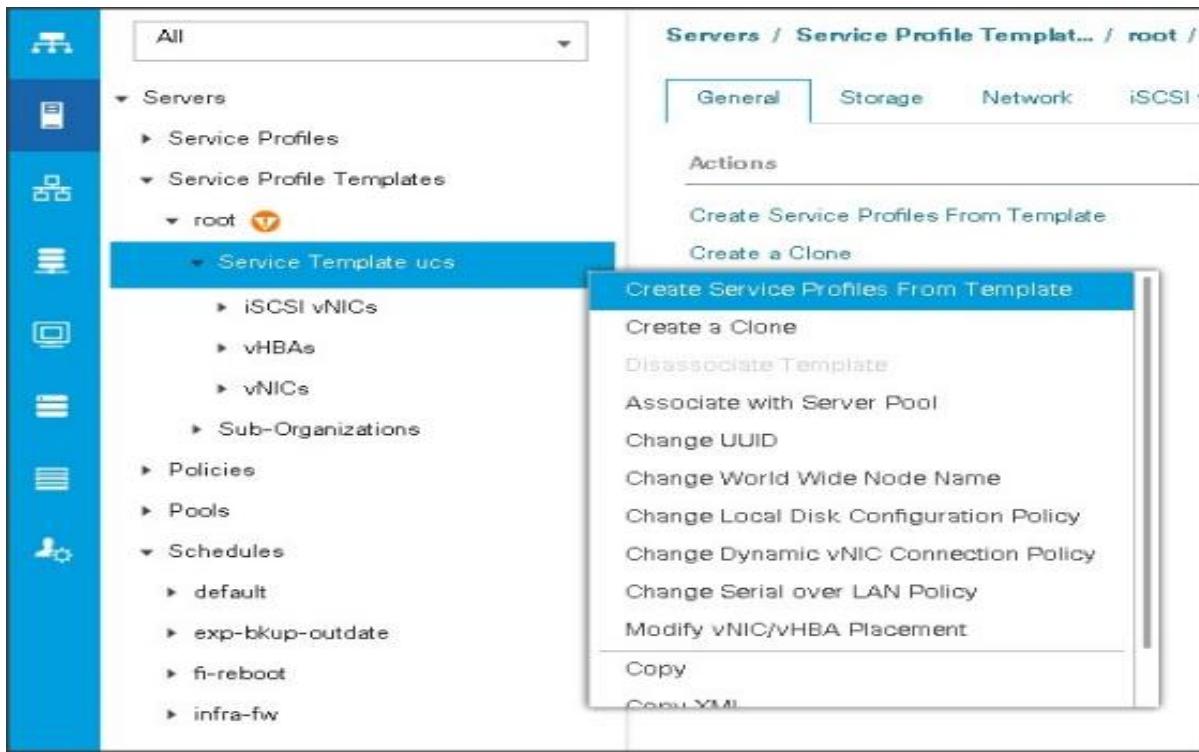
Configure the Operational Policies for the Template

To configure the operational policies for the template, in the Operational Policies Window, follow these 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.

Create Service Profiles From Template ? X

Naming Prefix :	ucs
Name Suffix Starting Number:	1
Number of Instances :	28

OK Cancel



Association of the Service Profiles will take place automatically.

Install Red Hat Enterprise Linux 7.6

This section provides detailed procedures to install Red Hat Enterprise Linux 7.6 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 the KVM console and virtual media from Cisco UCS Manager.



This installation requires RHEL 7.6 DVD/ISO.

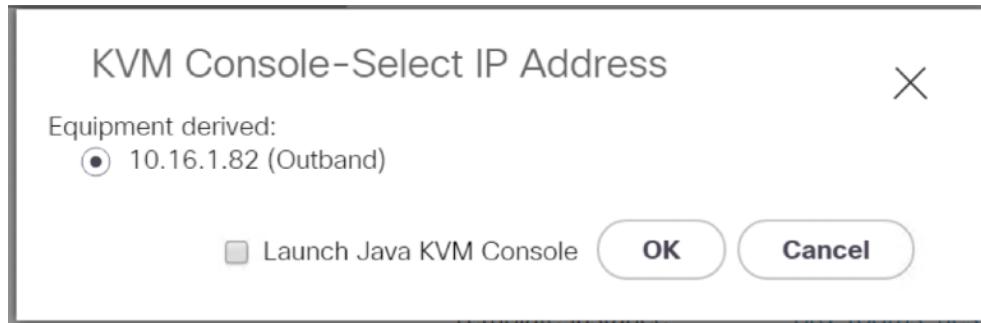
To install the Red Hat Linux 7.6 operating system, follow these steps:

1. Log into 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 >>.

The screenshot shows the Cisco UCS Management Interface. On the left, the navigation pane is open, showing 'Servers' and 'Service Profiles'. Under 'Service Profiles', 'root' is expanded, and 'ucs1' is selected. The main pane displays the path 'Servers / Service Profiles / root / Service Profile ucs1'. Below this, there are tabs for General, Storage, Network, and iSCSI vNICs. The General tab is selected. In the 'Actions' section, several options are listed: Set Desired Power State, Boot Server, Shutdown Server, Reset, KVM Console >>, SSH to CIMC for SoL >>, Rename Service Profile, and Create a Clone. The 'KVM Console >>' option is highlighted with a blue background.

5. Click OK on the KVM Console – Select IP address pop-up window.

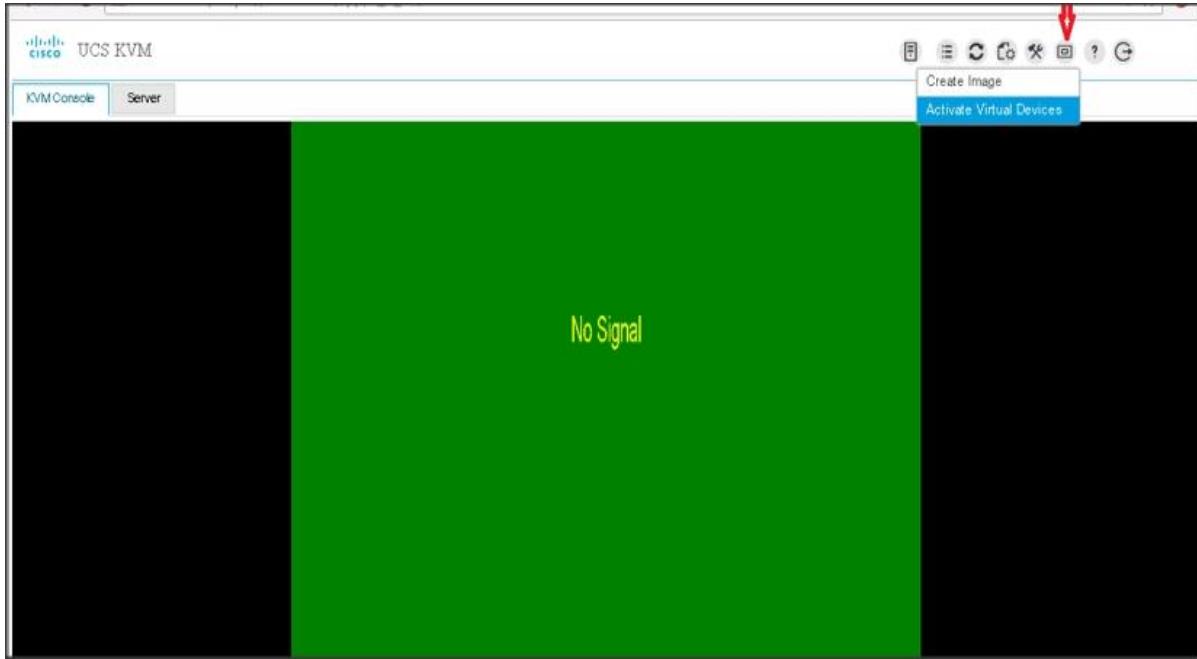


6. Click the link to launch the KVM console.

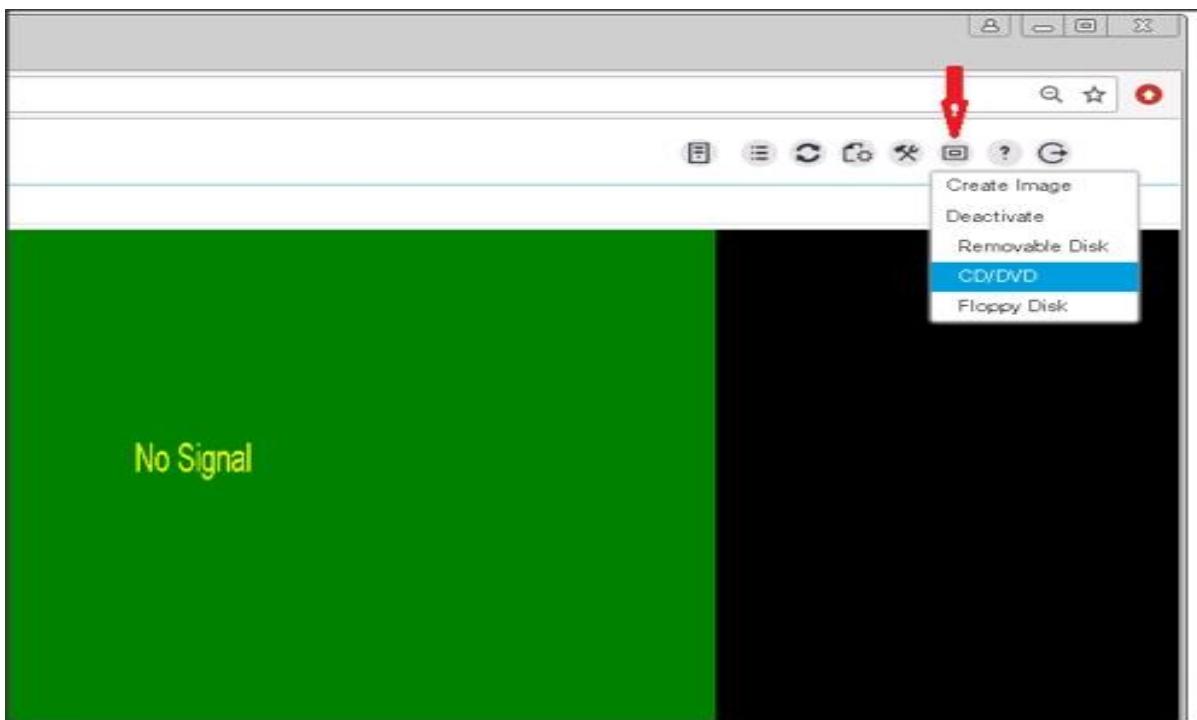
?redirect_url=https://10.16.1.10/app/4_0_2_80a/kvm.html%3F%26kvmIpAddr%3D10.16.1.82

KVM server certificate has been accepted. Click this link to continue loading the KVM client application:
https://10.16.1.10/app/4_0_2_80a/kvm.html?&kvmIpAddr=10.16.1.82

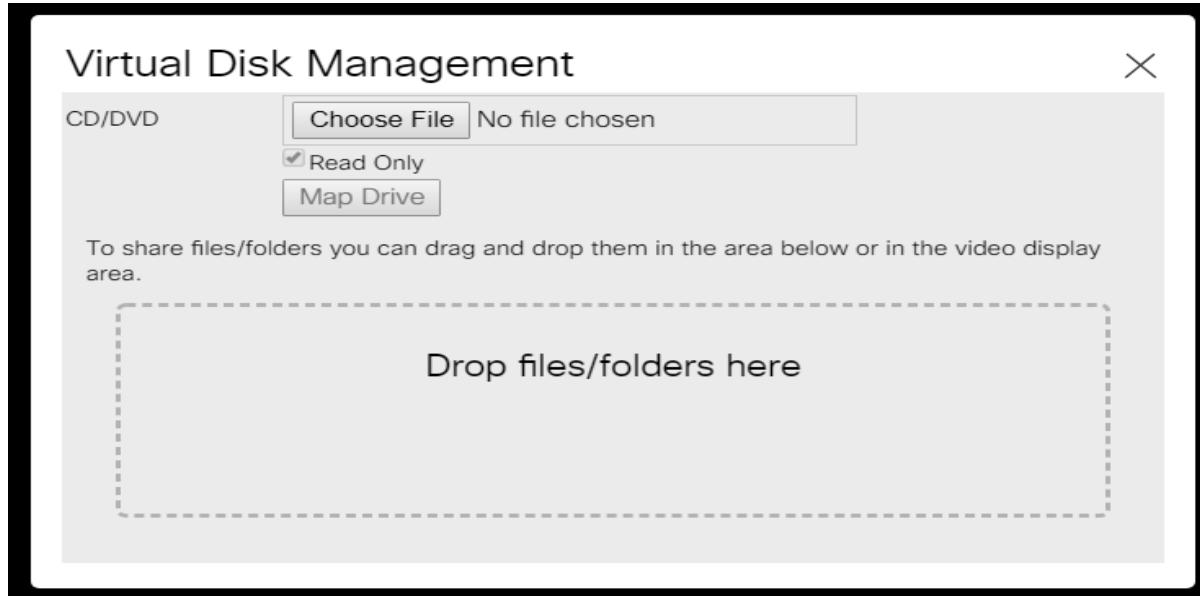
7. Point the cursor over the top right corner, select the Virtual Media tab.



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



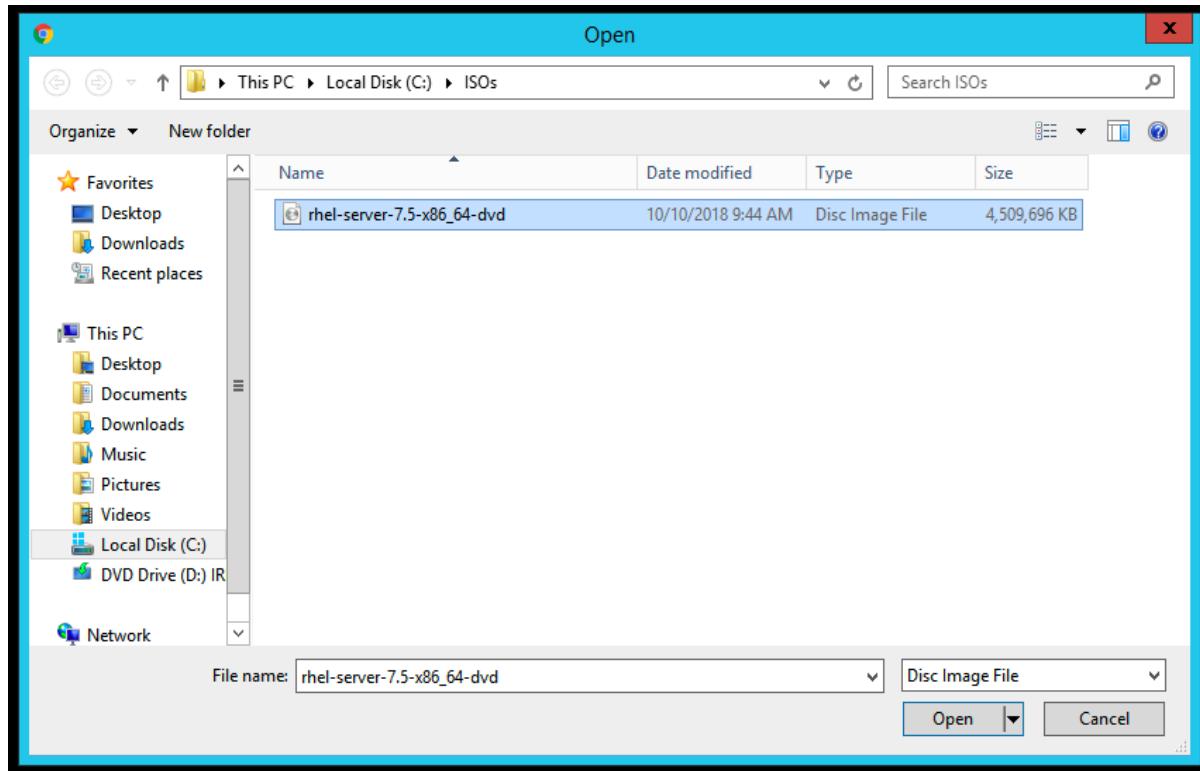
10. Select Choose File in the Virtual Disk Management windows.



11. Browse to the Red Hat Enterprise Linux Server 7.6 installer ISO image File.

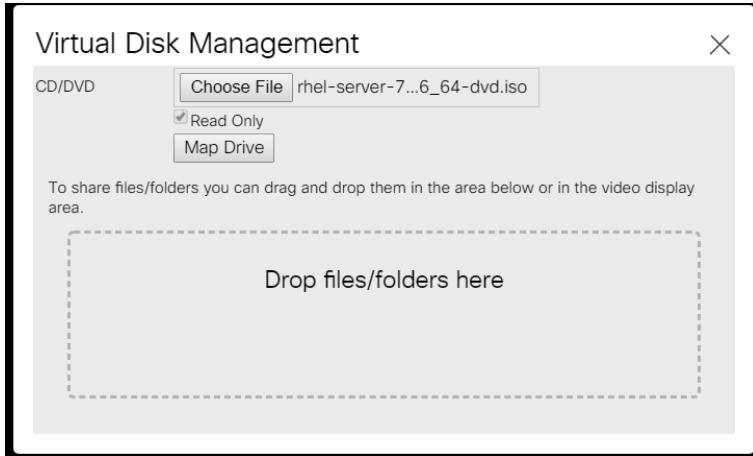


The Red Hat Enterprise Linux 7.6 DVD is assumed to be on the client machine.

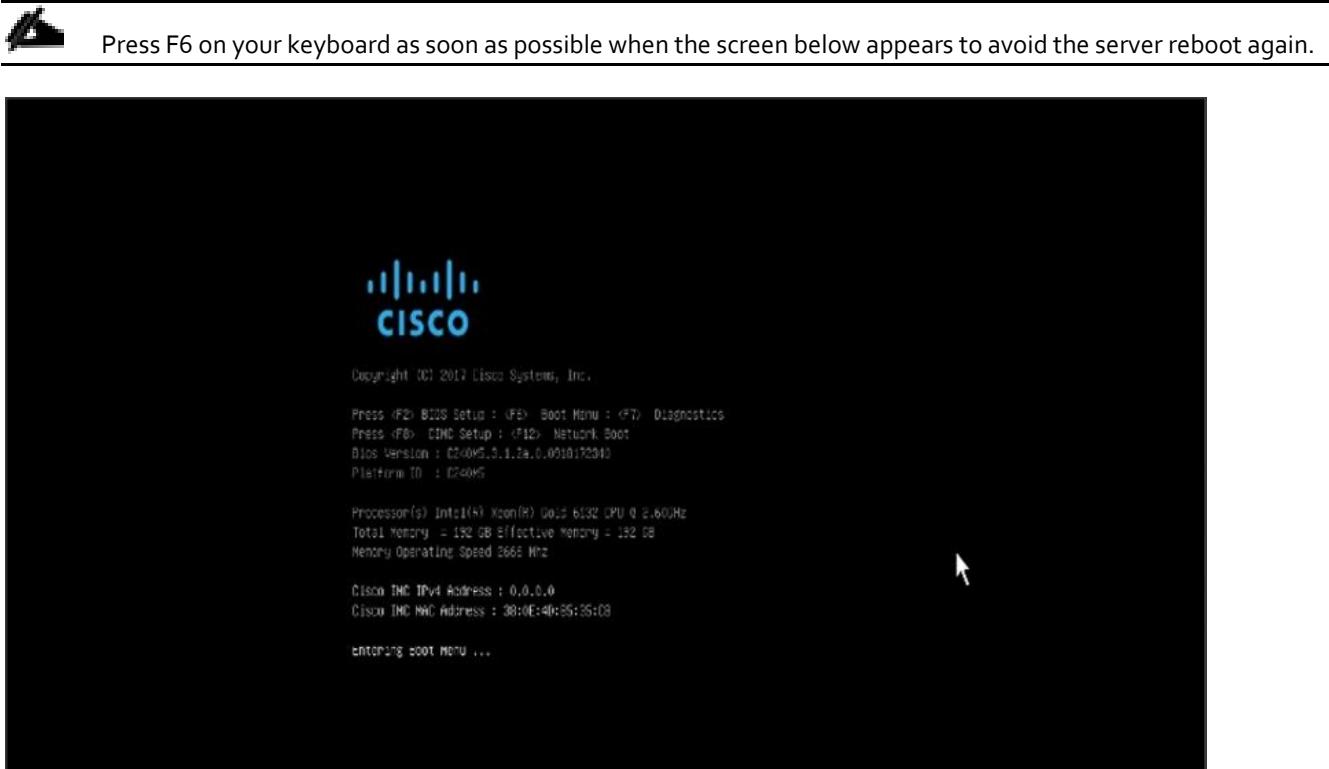


12. Click Open to add the image to the list of virtual media.

13. Click Map Drive after selecting the .iso file.



14. In the KVM window, select the KVM tab to monitor during boot.
15. In the KVM window, select the Macros > Static Macros > Ctrl-Alt-Del button in the upper left corner.
16. Click OK.
17. Click OK to reboot the system.
18. Press F6 key on the keyboard to select install media.



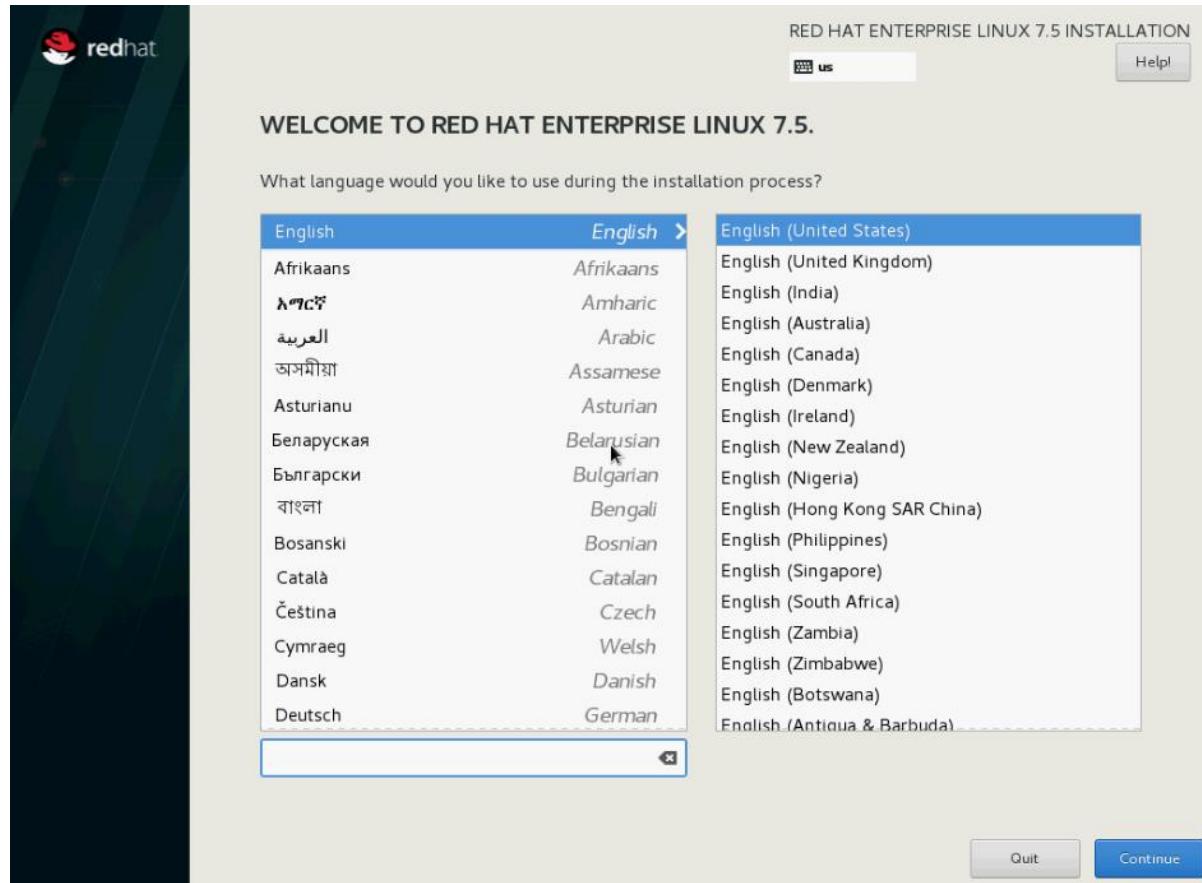
19. On reboot, the machine detects the presence of the Red Hat Enterprise Linux Server 7.6 install media.



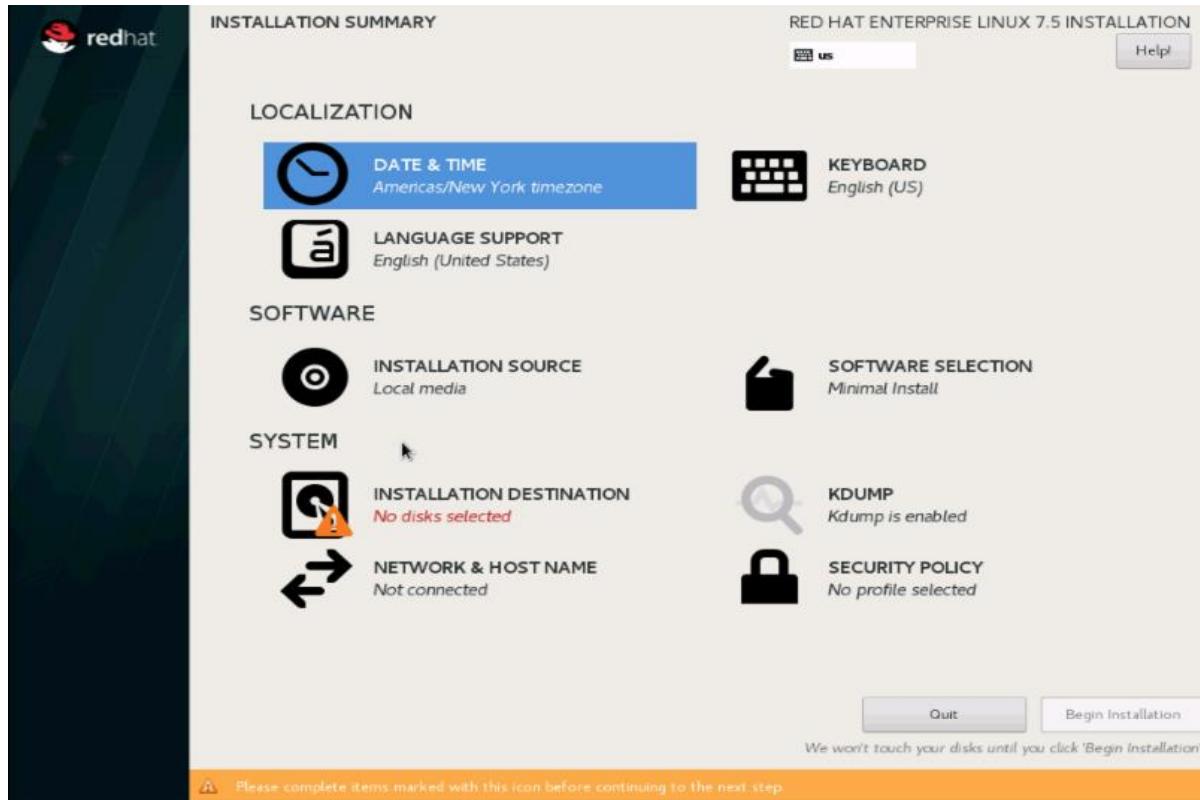
20. Select the Install Red Hat Enterprise Linux 7.6.



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



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



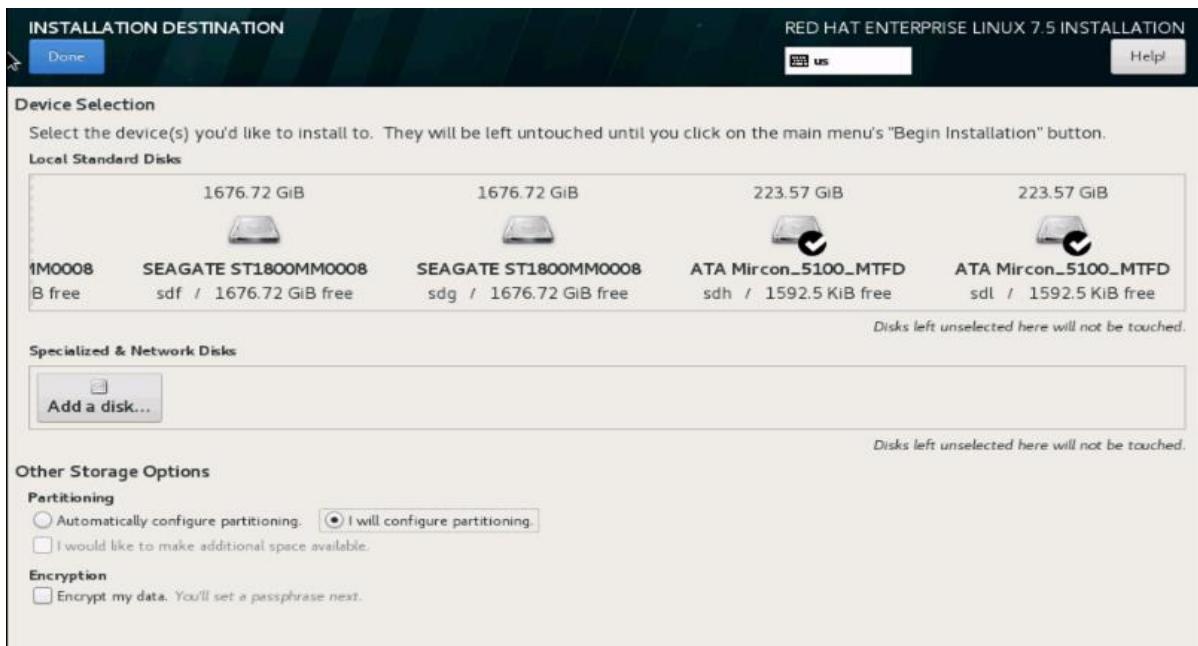
23. Select the location on the map, set the time, and click Done.



24. Click INSTALLATION DESTINATION.

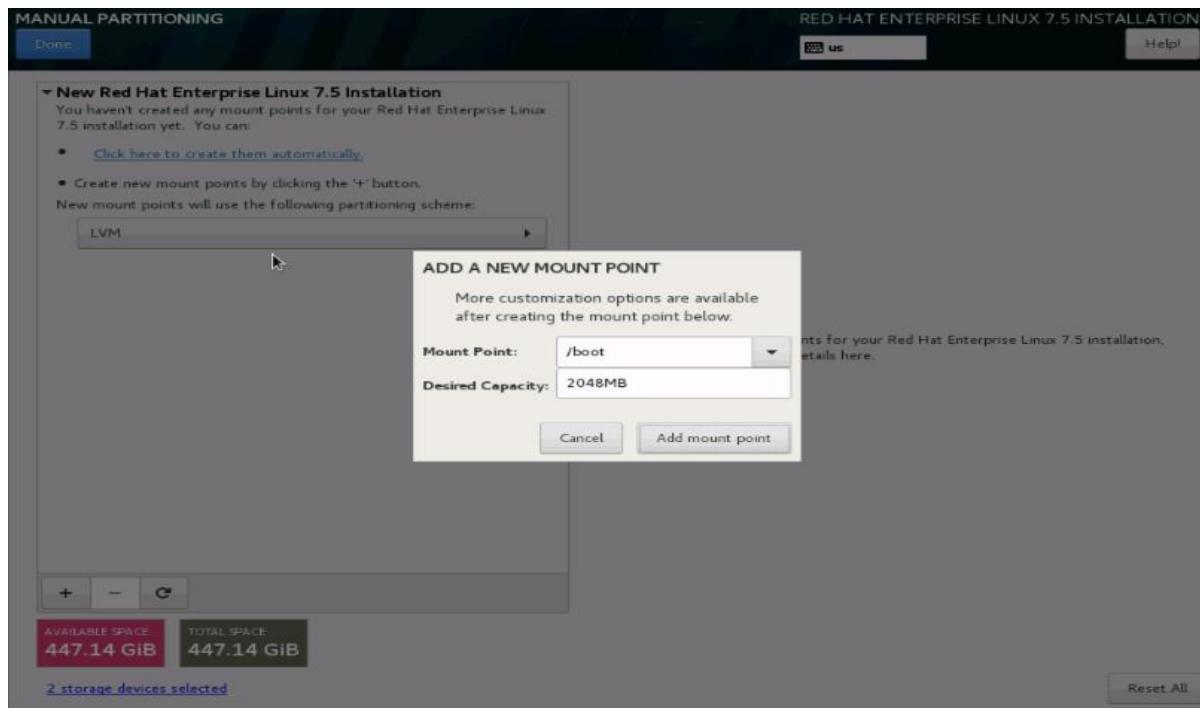


25. This opens a new window with the boot disks. Make the selection and choose I will configure partitioning. Click Done.

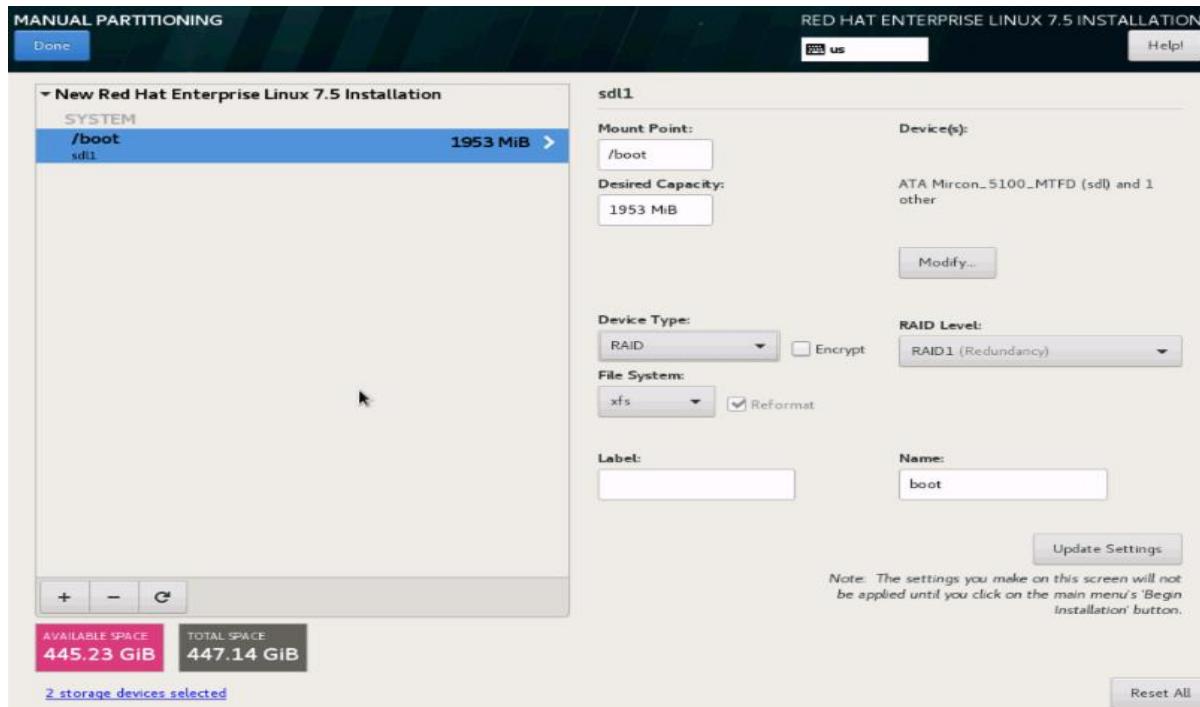


26. This opens the new window for creating the partitions. Click the + sign to add a new partition as shown below, boot partition of size 2048 MB.

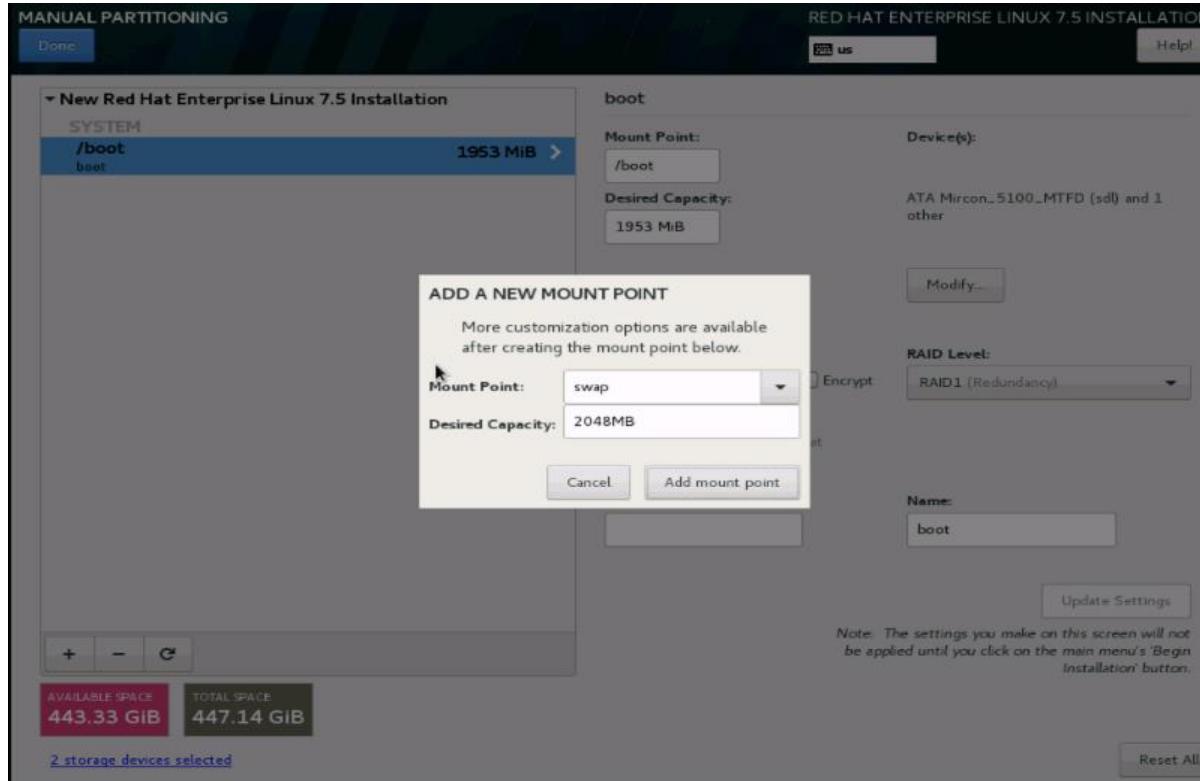
27. Click Add Mount Point to add the partition.



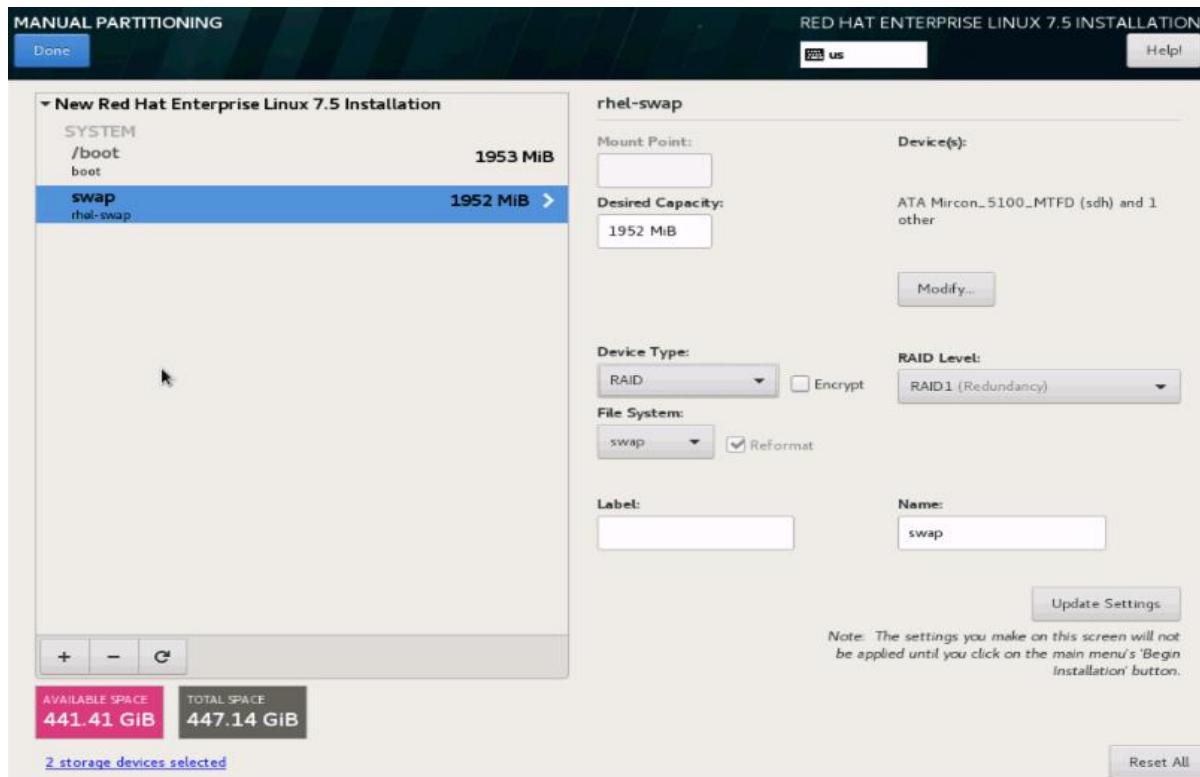
28. Change the Device type to RAID and make sure the RAID Level is RAID1 (Redundancy) and click Update Settings to save the changes.



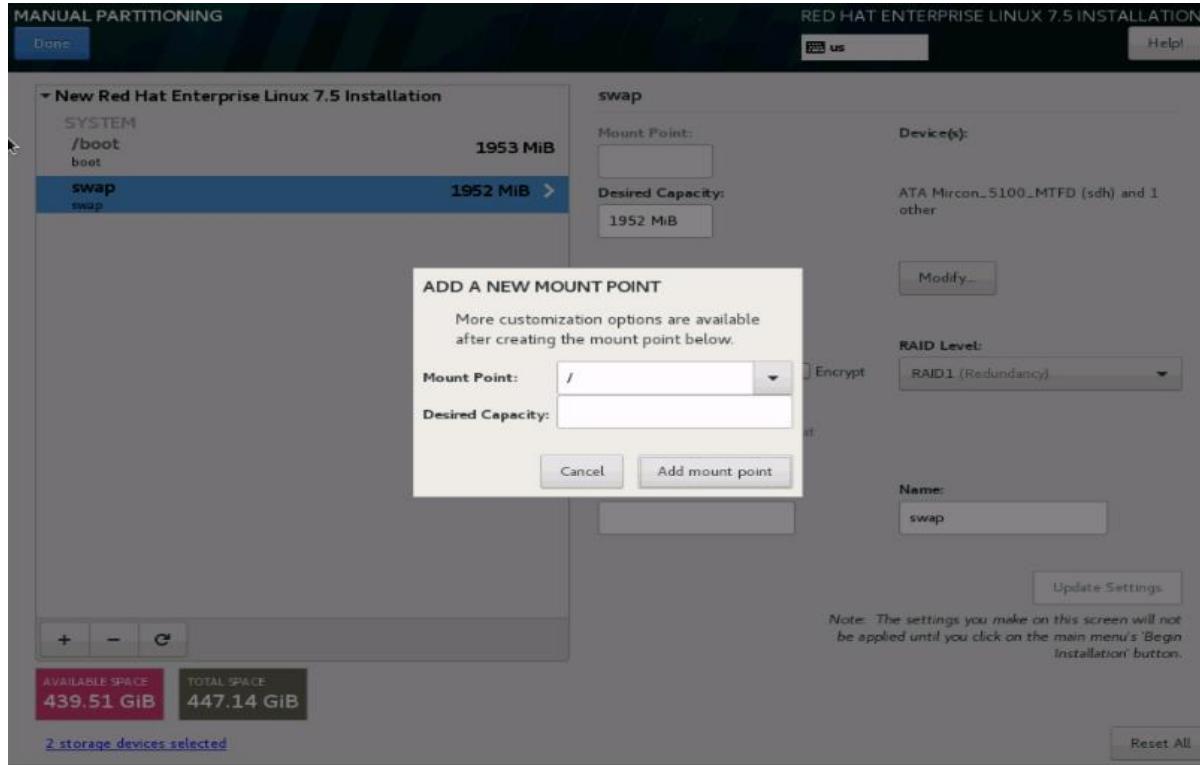
29. Click the + sign to create the swap partition of size 2048 MB as shown below.



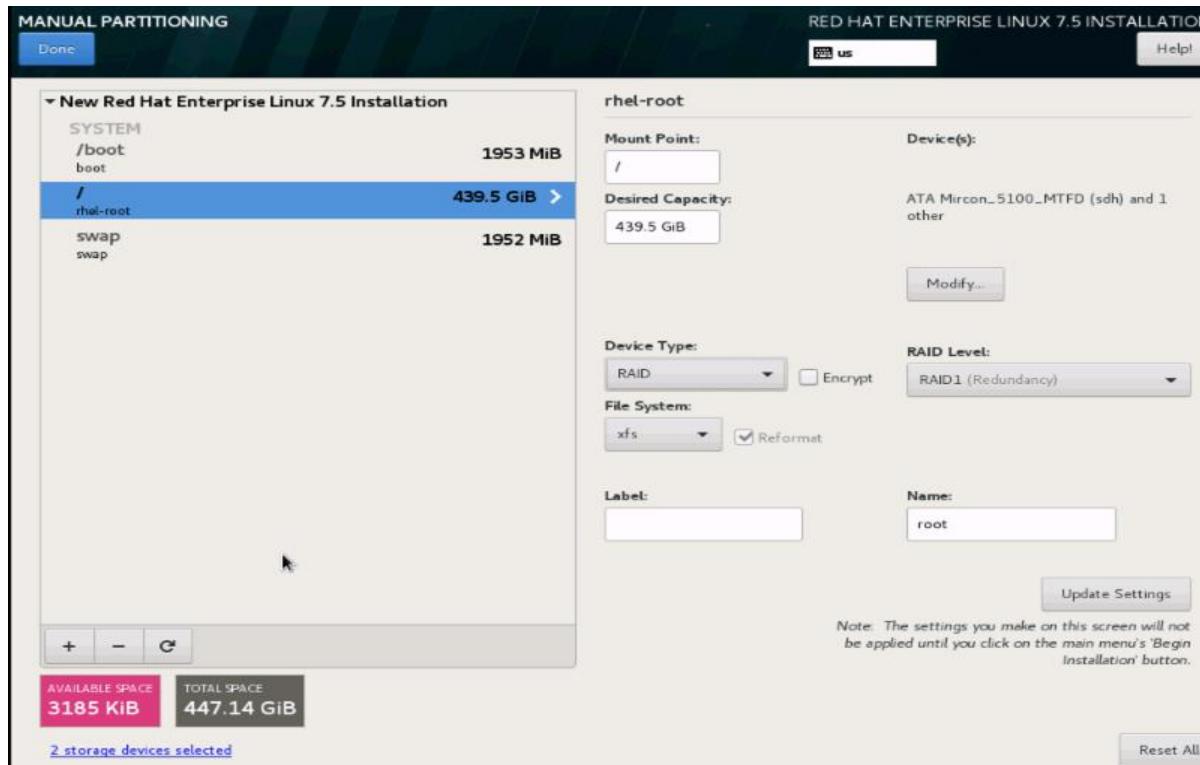
30. Change the Device type to RAID and RAID level to RAID1 (Redundancy) and click Update Settings.



31. Click + to add the / partition. The size can be left empty, so it uses the remaining capacity and click Add Mount Point.

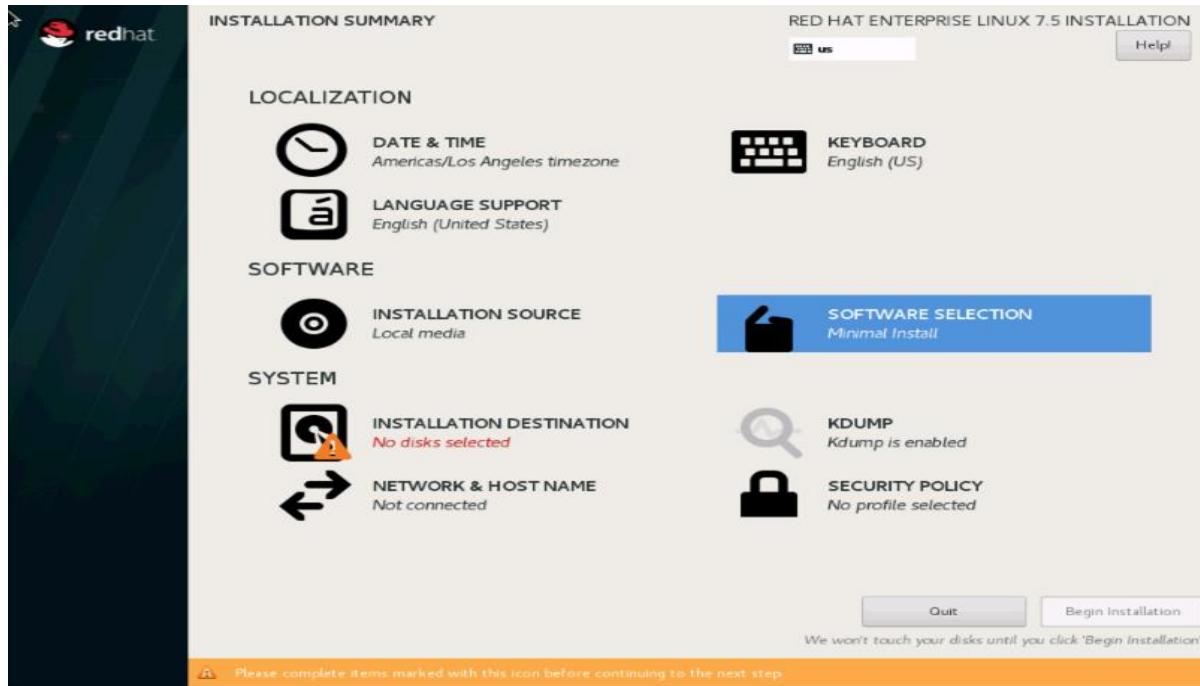


32. Change the Device type to RAID and RAID level to RAID1 (Redundancy). Click Update Settings.

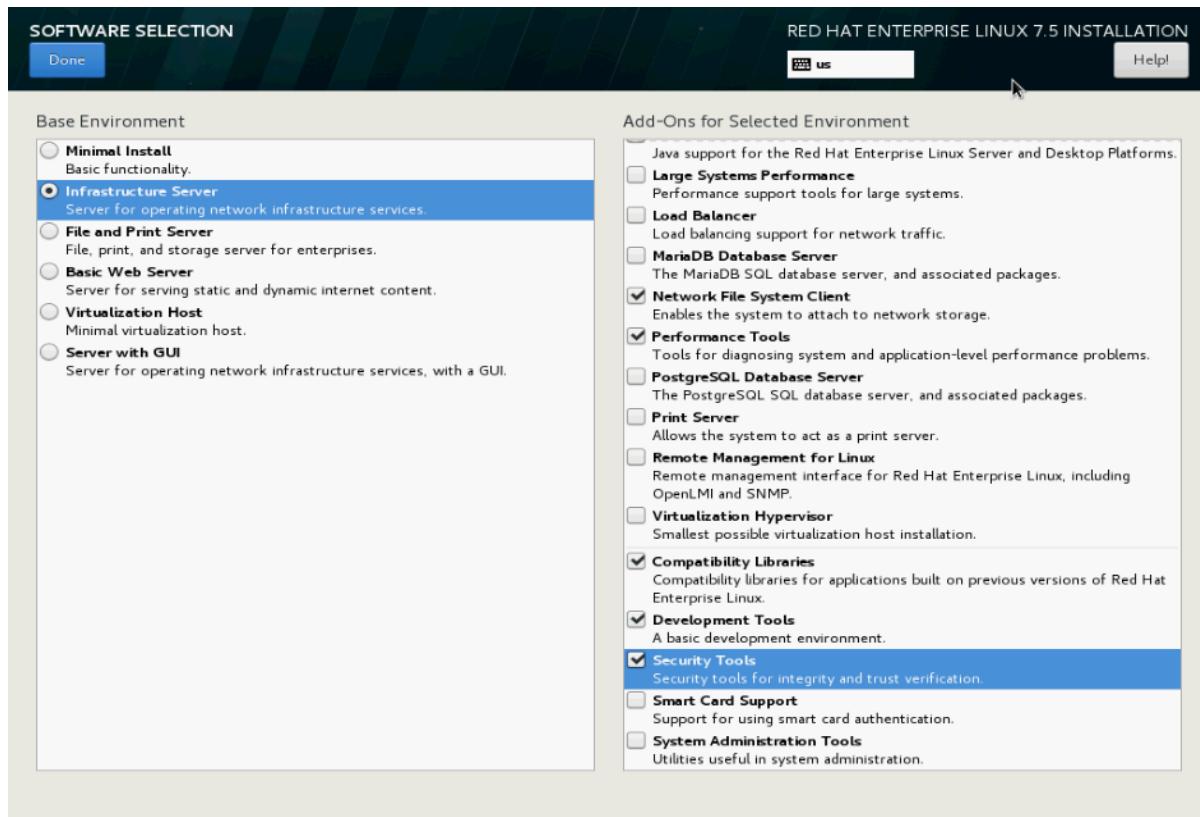


33. Click Done to return to the main screen and continue the Installation.

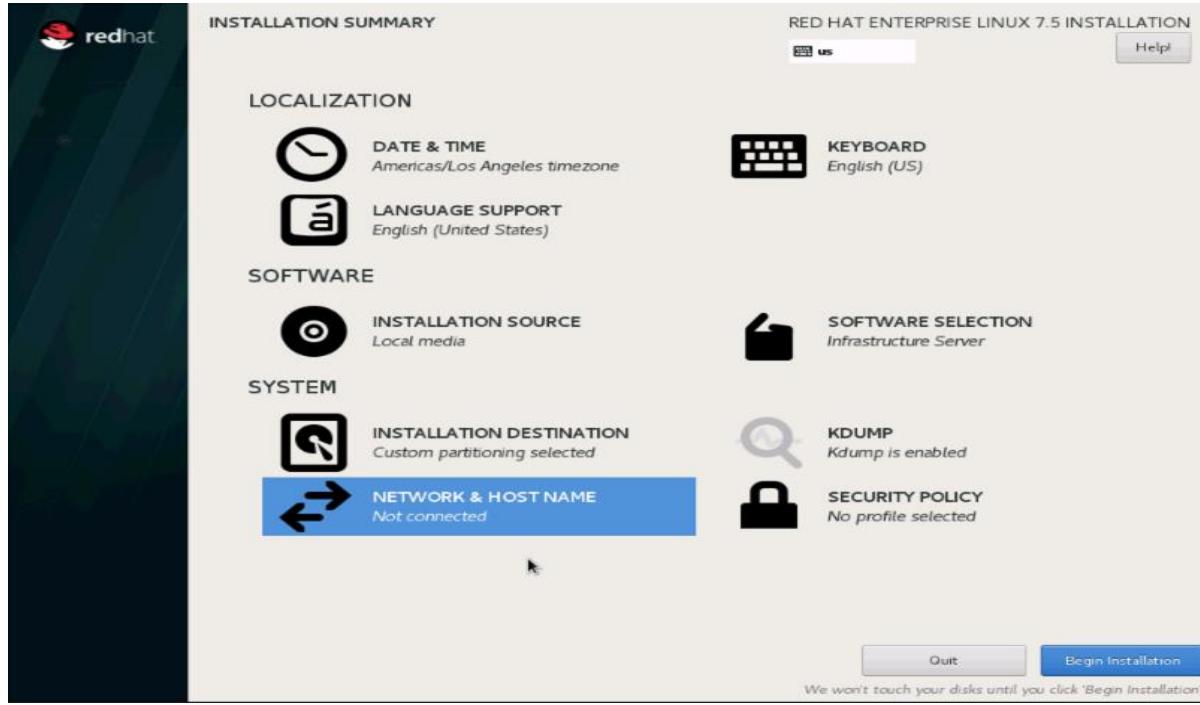
34. Click SOFTWARE SELECTION.



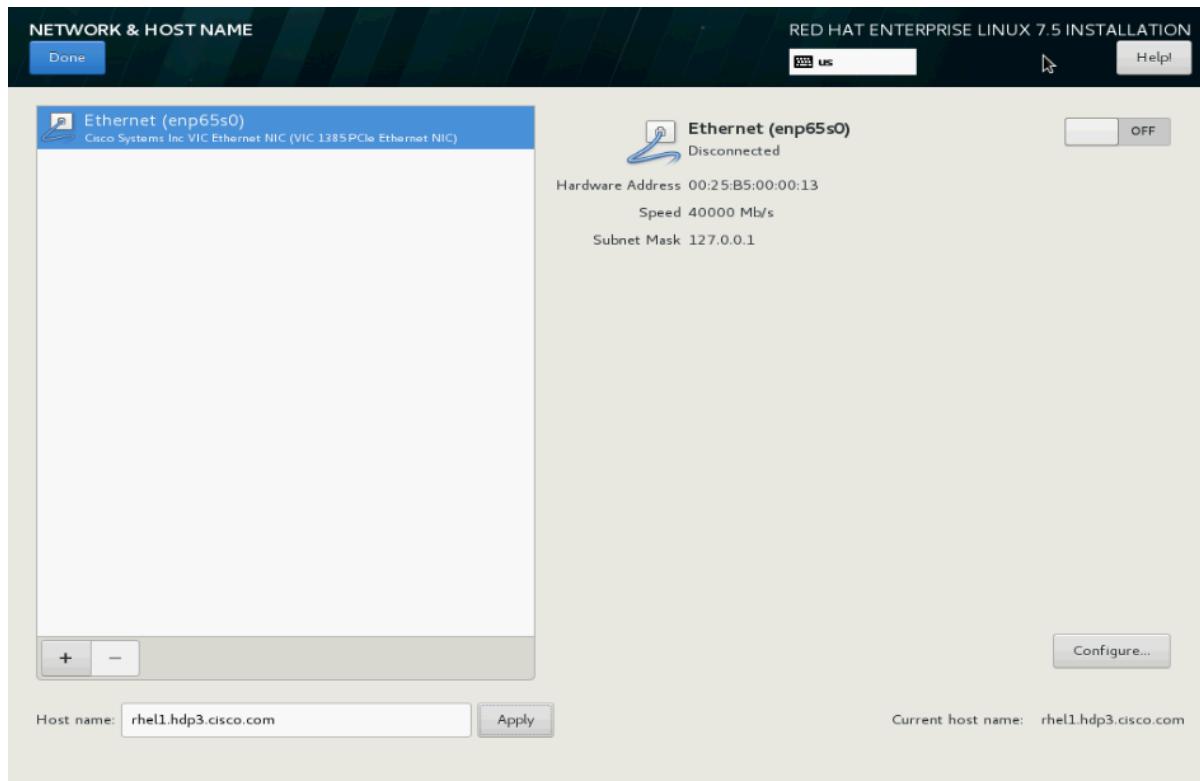
35. Select Infrastructure Server and select the Add-Ons as noted below. Click Done.



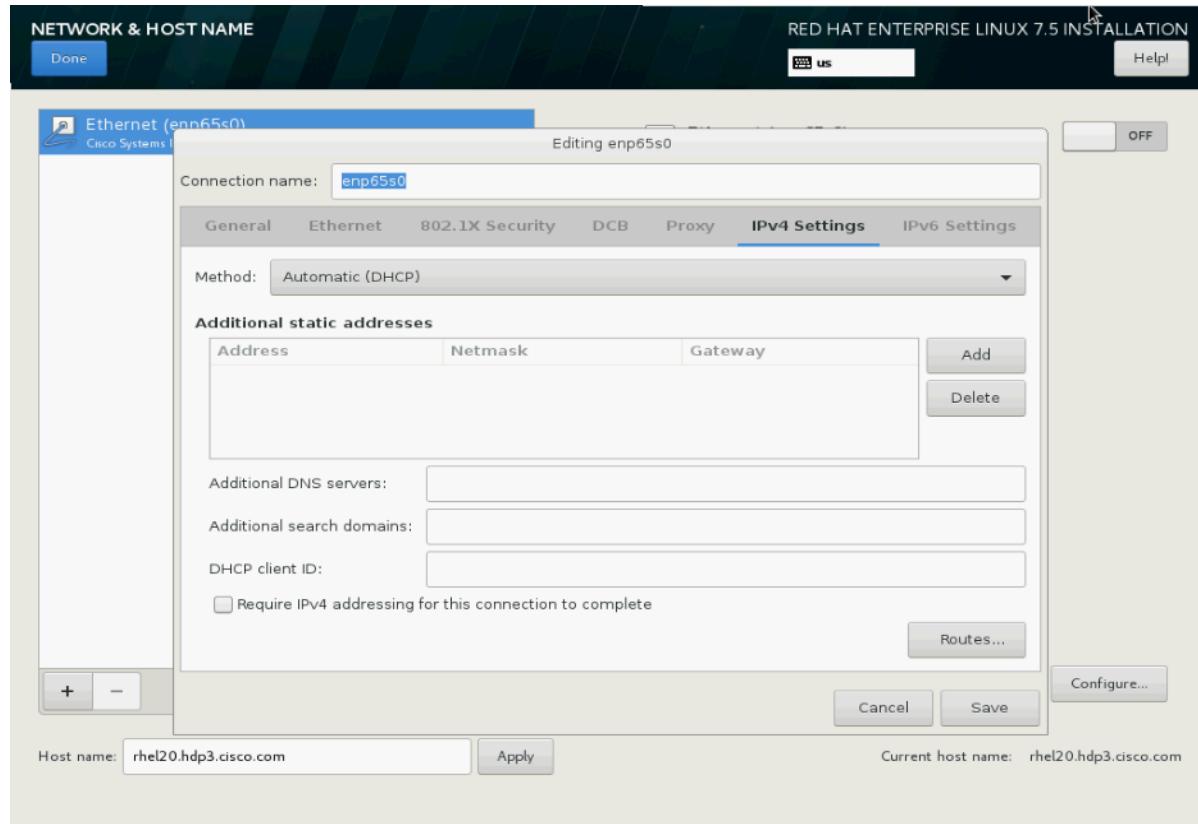
36. Click **NETWORK & HOSTNAME** and configure Hostname and Networking for the Host.



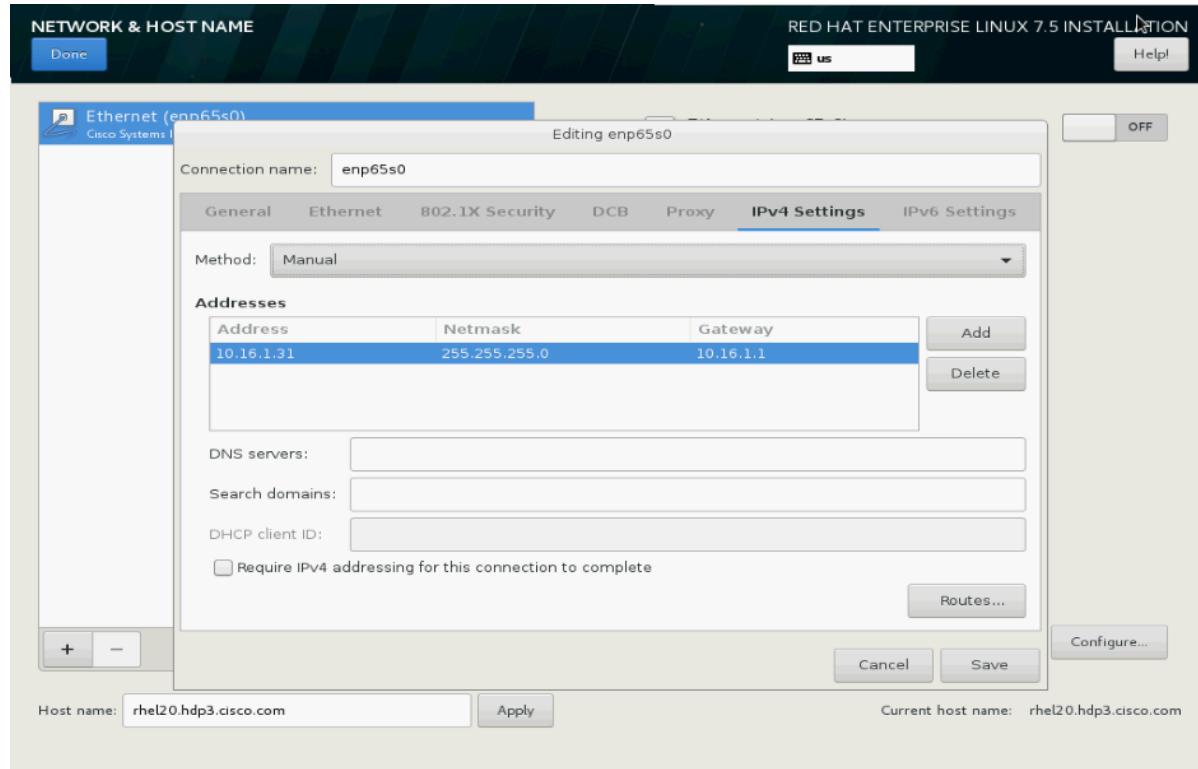
37. Type in the hostname as shown below.



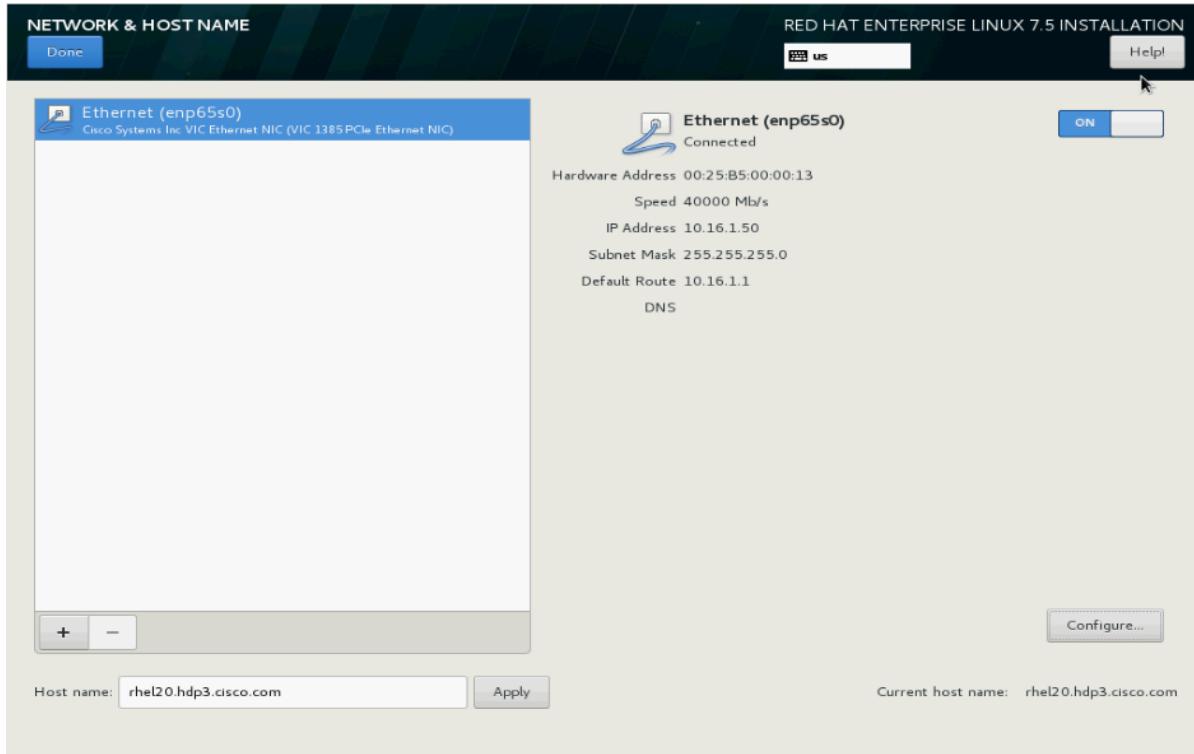
38. Click Configure to open the Network Connectivity window. Click IPV4Settings.



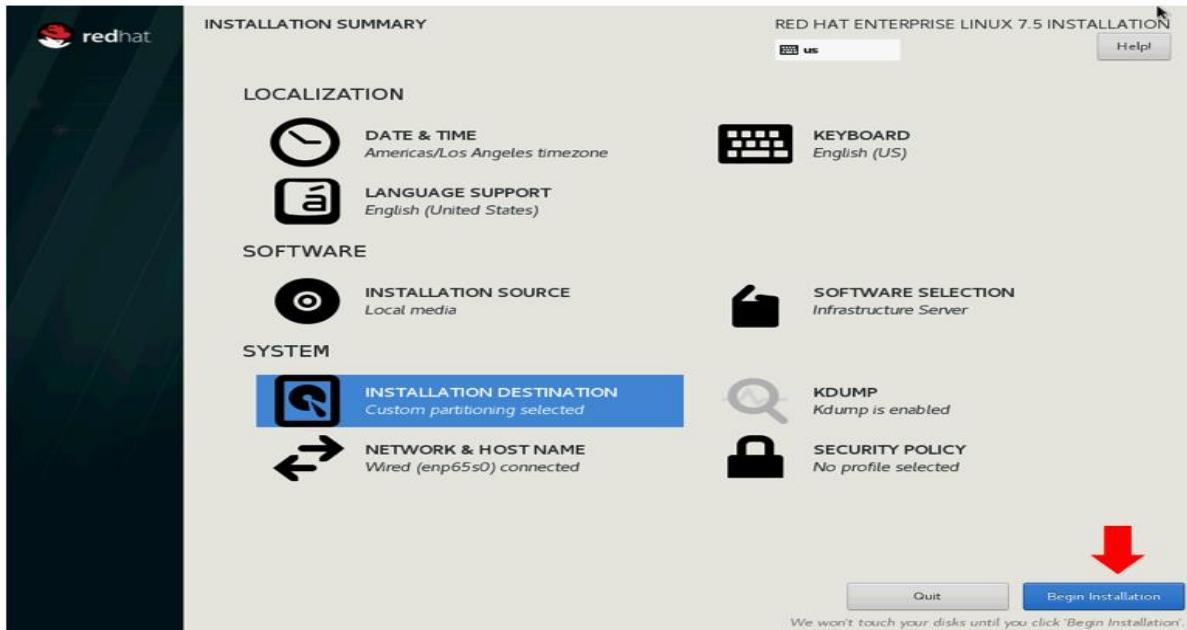
39. Change the Method to Manual and click Add to enter the IP Address, Netmask, and Gateway details.



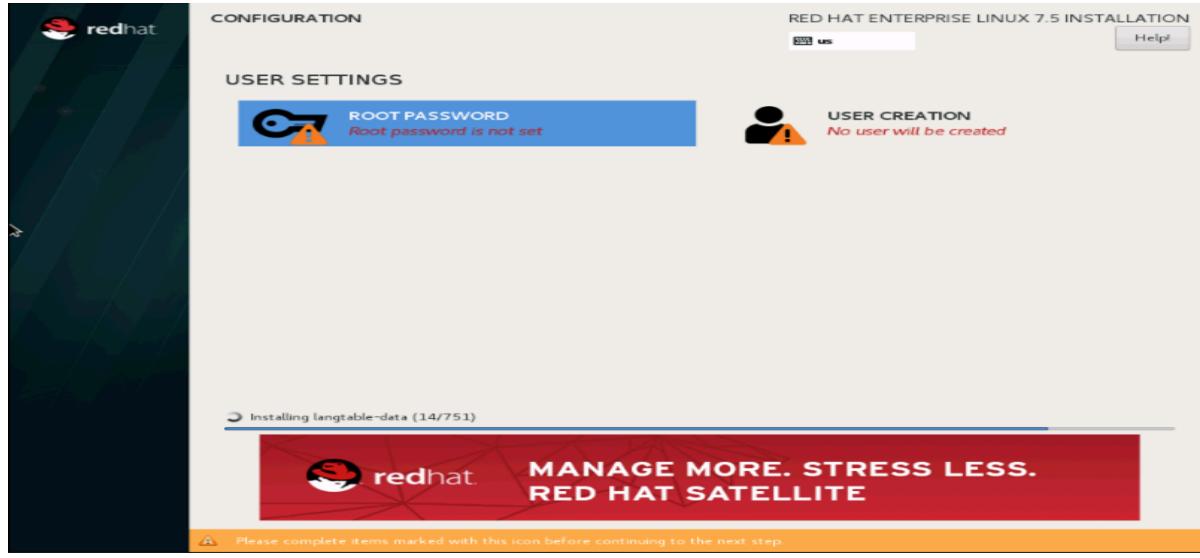
40. Click Save and update the hostname and turn Ethernet ON. Click Done to return to the main menu.



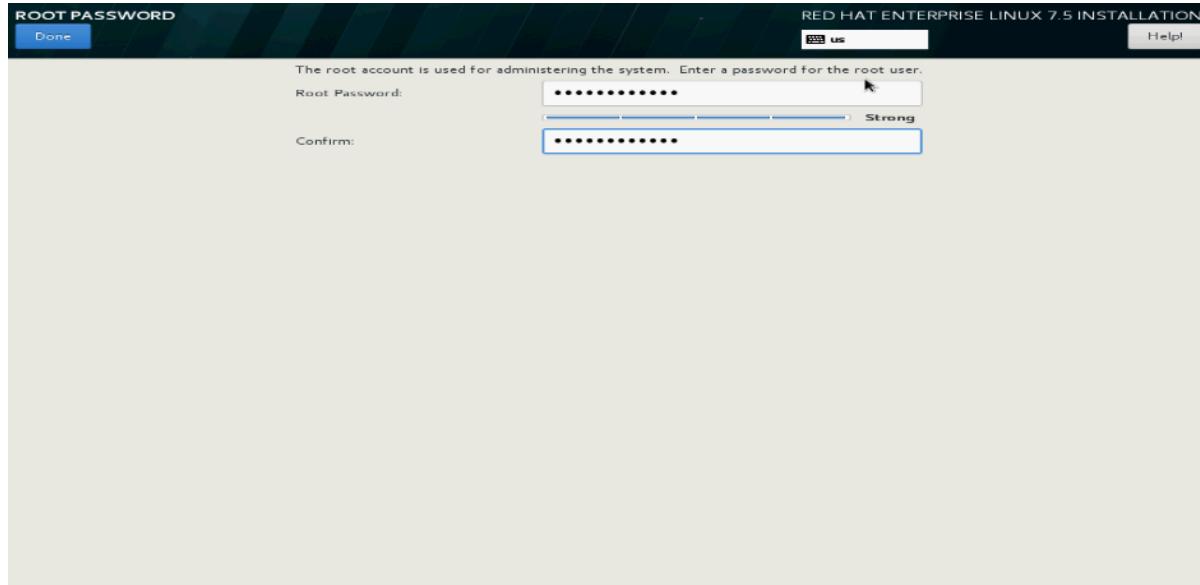
41. Click Begin Installation on the Main menu.



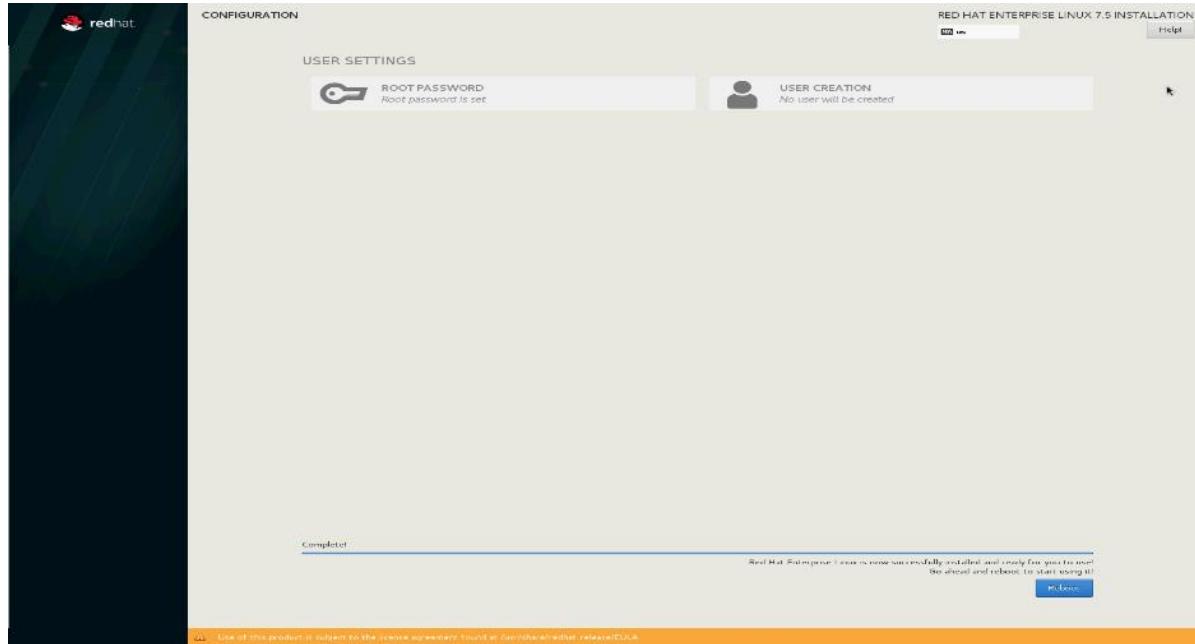
42. Select Root Password in the User Settings.



43. Enter the Root Password and click Done.



44. When the installation is complete reboot the system.



45. Repeat the steps to install Red Hat Enterprise Linux 7.6 on the remaining servers.



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 8.

Table 8 Hostnames and IP Addresses

Hostname	eth0
rhel1	10.16.1.31
rhel2	10.16.1.32
rhel3	10.16.1.33
rhel4	10.16.1.34
rhel1	10.16.1.35
rhel6	10.16.1.36
rhel7	10.16.1.37
rhel8	10.16.1.38
rhel9	10.16.1.39
rhel10	10.16.1.40
rhel11	10.16.1.41

Hostname	eth0
rhel12	10.16.1.42
rhel13	10.16.1.43
rhel14	10.16.1.44
rhel15	10.16.1.45
rhel16	10.16.1.46
...	...
rhel24	10.16.1.54



Multi-homing configuration is not recommended in this design, so please assign only one network interface on each host.



For simplicity outbound NATing is configured for internet access when desired such as accessing public repos and/or accessing Red Hat Content Delivery Network. However, configuring outbound NAT is beyond the scope of this document.

Post OS Install Configuration

Choose one of the nodes of the cluster or a separate node as the Admin Node for management such as HDP installation, Ansible, creating a local Red Hat repo and so on. In this document, rhel1 has been used for this purpose.

Configure /etc/hosts

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



For the purpose of simplicity, /etc/hosts file is configured with hostnames in all the nodes. However, in large scale production grade deployment, DNS server setup is highly recommended. Furthermore, /etc/hosts file is not copied into containers running on the platform.

Below are the sample A records for DNS configuration within Linux environment.

```
$ORIGIN hdp3.cisco.com.
rhel1    A 10.16.1.31
rhel2    A 10.16.1.32
rhel3    A 10.16.1.33
...
...
rhel28   A 10.16.1.59
```

To create the host file on the admin node, follow these steps:

1. Log into the Admin Node (rhel1).

```
#ssh 10.16.1.31
```

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

3. On Admin Node (rhel1):

```
# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
10.16.1.31 rhel1 rhel1.hdp3.cisco.com
10.16.1.32 rhel2 rhel2.hdp3.cisco.com
10.16.1.33 rhel3 rhel3.hdp3.cisco.com
10.16.1.34 rhel4 rhel4.hdp3.cisco.com
10.16.1.35 rhel5 rhel5.hdp3.cisco.com
10.16.1.36 rhel6 rhel6.hdp3.cisco.com
10.16.1.37 rhel7 rhel7.hdp3.cisco.com
10.16.1.38 rhel8 rhel8.hdp3.cisco.com
10.16.1.39 rhel9 rhel9.hdp3.cisco.com
10.16.1.40 rhel10 rhel10.hdp3.cisco.com
10.16.1.41 rhel11 rhel11.hdp3.cisco.com
10.16.1.42 rhel12 rhel12.hdp3.cisco.com
10.16.1.43 rhel13 rhel13.hdp3.cisco.com
10.16.1.44 rhel14 rhel14.hdp3.cisco.com
10.16.1.45 rhel15 rhel15.hdp3.cisco.com
10.16.1.46 rhel16 rhel16.hdp3.cisco.com
10.16.1.47 rhel17 rhel17.hdp3.cisco.com
```

Set Up the Passwordless Login

To manage all of the cluster nodes from the admin node password-less login needs to be setup. It assists in automating common tasks with Ansible, and shell-scripts without having to use passwords.

To enable a passwordless login across all the nodes when Red Hat Linux is installed across all the nodes in the cluster, follow these steps:

1. Log into the Admin Node (rhel1).

```
#ssh 10.13.1.31
```

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

```
# ssh-keygen -N '' -f ~/.ssh/id_rsa
```

Figure 33 ssh-keygen

```
[root@rhel1 ansible]# ssh-keygen -N '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
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:j+IdDaaxUBH2ciy/c4M0YcDPhgOoRWrsb8NGnaaj28s root@rhel1
The key's randomart image is:
+---[RSA 2048]---+
| ... .=.
| ..o ...=
| .= .++*
| + ..++*.
| . . .+oS
| + o. * O
| O + * =
| * o. o + .
| o.E. . .
+---[SHA256]---+
```

3. 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 i in {1..17}; do echo "copying rhel$i.hdp3.cisco.com"; ssh-copy-id -i
~/.ssh/id_rsa.pub root@rhel$i.hdp3.cisco.com; done;
```

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

Create the Red Hat Enterprise Linux (RHEL) 7.6 Local Repository

To create the 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 into rhel1. Create a directory that would contain the repository.

```
# mkdir -p /var/www/html/rhelrepo
Copy the contents of the Red Hat DVD to /var/www/html/rhelrepo
```

2. Alternatively, if you have access to a Red Hat ISO Image, Copy the ISO file to rhel1.

```
Log back into rhel1 and create the mount directory.
# scp rhel-server-7.6-x86_64-dvd.iso rhel1:/root/

# mkdir -p /mnt/rheliso
# mount -t iso9660 -o loop /root/rhel-server-7.6-x86_64-dvd.iso /mnt/rheliso/
Copy the contents of the ISO to the /var/www/html/rhelrepo directory:
# cp -r /mnt/rheliso/* /var/www/html/rhelrepo
```

3. On rhel1 create a .repo file to enable the use of the yum command:

```
# vi /var/www/html/rhelrepo/rheliso.repo
[rhel7.6]
name=Red Hat Enterprise Linux 7.6
baseurl=http://10.16.1.31/rhelrepo
gpgcheck=0
enabled=1
```

4. 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/
```



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

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.



This step is needed to install software on Admin Node (rhel1) using the repo (such as httpd, create-repo, and so on.)

```
# vi /etc/yum.repos.d/rheliso.repo
[rhel7.6]
name=Red Hat Enterprise Linux 7.6
baseurl=file:///var/www/html/rhelrepo
gpgcheck=0
enabled=1
```

Create the Red Hat Repository Database

To create the Red Hat repository database, follow these steps:

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
```

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

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

Figure 34 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
```

Set Up Ansible

To set up Ansible, follow these steps:

1. Download Ansible rpm from the following link:

https://releases.ansible.com/ansible/rpm/release/epel-7-x86_64/ansible-2.4.6.0-1.el7.ans.noarch.rpm

```
# wget https://releases.ansible.com/ansible/rpm/release/epel-7-x86_64/ansible-2.4.6.0-1.el7.ans.noarch.rpm
```



For more information about to download and install Ansible engine, please follow the link
<https://access.redhat.com/articles/3174981>

2. Run the following command to install ansible:

```
# yum localinstall -y ansible-2.4.6.0-1.el7.ans.noarch.rpm
```

3. Verify Ansible installation by running the following commands:

```
# ansible --version
# ansible localhost -m ping
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit
localhost does not match 'all'

localhost | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
```

4. Prepare the host inventory file for Ansible as shown below. Various host groups have been created based on any specific installation requirements of certain hosts:

```
[root@rhel1 ~]# cat /etc/ansible/hosts
[admin]
rhel1.hdp3.cisco.com

[namenodes]
rhel1.hdp3.cisco.com
rhel2.hdp3.cisco.com
rhel3.hdp3.cisco.com

[datanodes]
rhel4.hdp3.cisco.com
rhel5.hdp3.cisco.com
rhel6.hdp3.cisco.com
rhel7.hdp3.cisco.com
rhel8.hdp3.cisco.com
rhel9.hdp3.cisco.com
rhel10.hdp3.cisco.com
rhel11.hdp3.cisco.com
rhel12.hdp3.cisco.com
rhel13.hdp3.cisco.com
rhel14.hdp3.cisco.com
rhel15.hdp3.cisco.com
rhel16.hdp3.cisco.com
```

```
rhel124.hdp3.cisco.com
rhel125.hdp3.cisco.com
rhel126.hdp3.cisco.com
rhel127.hdp3.cisco.com
rhel128.hdp3.cisco.com
rhel129.hdp3.cisco.com
rhel130.hdp3.cisco.com
rhel131.hdp3.cisco.com
```

```
[gpunodes]
Rhel18.hdp3.cisco.com
Rhel19.hdp3.cisco.com
Rhel20.hdp3.cisco.com
rhel21.hdp3.cisco.com
rhel22.hdp3.cisco.com
rhel23.hdp3.cisco.com
```

```
[nodes]
rhel1.hdp3.cisco.com
rhel2.hdp3.cisco.com
rhel3.hdp3.cisco.com
rhel4.hdp3.cisco.com
rhel5.hdp3.cisco.com
rhel6.hdp3.cisco.com
rhel7.hdp3.cisco.com
rhel8.hdp3.cisco.com
rhel9.hdp3.cisco.com
rhel10.hdp3.cisco.com
rhel11.hdp3.cisco.com
rhel12.hdp3.cisco.com
rhel13.hdp3.cisco.com
rhel14.hdp3.cisco.com
rhel15.hdp3.cisco.com
rhel16.hdp3.cisco.com
rhel17.hdp3.cisco.com
rhel18.hdp3.cisco.com
rhel19.hdp3.cisco.com
rhel20.hdp3.cisco.com
rhel21.hdp3.cisco.com
rhel22.hdp3.cisco.com
rhel23.hdp3.cisco.com
rhel24.hdp3.cisco.com
rhel25.hdp3.cisco.com
rhel26.hdp3.cisco.com
rhel27.hdp3.cisco.com
rhel28.hdp3.cisco.com
rhel29.hdp3.cisco.com
rhel30.hdp3.cisco.com
rhel31.hdp3.cisco.com
```

5. Verify host group by running the following commands. Figure 35 shows the outcome of the ping command:

```
# ansible gpunodes -m ping
```

Figure 35 Ansible – Ping Hosts

```
[root@rhel1 ~]# ansible nodeswithgpu -m ping
rhel17.hdp3.cisco.com | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
rhel15.hdp3.cisco.com | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
rhel16.hdp3.cisco.com | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
rhel14.hdp3.cisco.com | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
```

Install httpd

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

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



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

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

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

```
# vi /etc/httpd/conf/httpd.conf
ServerName 10.16.1.31:80
```

3. Start httpd:

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

Set Up All Nodes to Use the RHEL Repository

To set up all nodes to use the RHEL repository, follow these steps:



Based on this repository 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:

```
# ansible nodes -m copy -a "src=/var/www/html/rhelrepo/rheliso.repo dest=/etc/yum.repos.d/."
```

2. Copy the /etc/hosts file to all nodes:

```
# ansible nodes -m copy -a "src=/etc/hosts dest=/etc/hosts"
```

3. Purge the yum caches:

```
# ansible nodes -a "yum clean all"
# ansible nodes -a "yum repolist"
```



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/
```

Upgrade 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/283862063/type/283853158/release/4.0.2>

1. In the ISO image, the required driver kmod-enic-.....rpm can be located at Net-work\Cisco\VIC\RHEL\RHEL7.6.
2. From a node connected to the Internet, download, extract and transfer kmod-enic-.rpm to rhel1 (admin node).
3. Copy the rpm on all nodes of the cluster using the following Ansible commands. For this example, the rpm is assumed to be in present working directory of rhel1:

```
[root@rhel1 ~]# ansible all -m copy -a "src=/root/kmod-enic-3.2.210.22-738.18.rhel7u6.x86_64.rpm dest=/root/."
```

4. Use the yum module to install the enic driver rpm file on all the nodes through Ansible:

```
[root@rhel1 ~]# ansible all -m yum -a "name=/root/kmod-enic-3.2.210.22-738.18.rhel7u6.x86_64.rpm state=present"
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 ~]# ansible all -m command -a "modinfo enic"
```

5. 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\RHEL7.6

Install xfsprogs

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

```
# ansible all -m yum -a "name=xfsprogs state=present"
```

Set Up JAVA

To setup JAVA, follow these steps:



HDP 3.1.0 requires JAVA 8.

1. Download jdk-7u76-linux-x64.rpm and src the rpm to admin node (rhel1) from the link (<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>)

2. Copy JDK rpm to all nodes:

```
# ansible nodes -m copy -a "src=/root/jdk-8u201-linux-x64.rpm dest=/root/."
```

3. Extract and Install JDK on all nodes:

```
# ansible all -m command -a "rpm -ivh jdk-8u201-linux-x64.rpm"

[root@rhel1 ~]# ansible nodes -m command -a "rpm -ivh jdk-8u181-linux-x64.rpm"
[WARNING]: Consider using yum, dnf or zypper module rather than running rpm

rhel3.hdp3.cisco.com | SUCCESS | rc=0 >>
Preparing... #####################################################
Updating / installing...
jdk1.8-2000:1.8.0_181-fcs #####################################################
Unpacking JAR files...
    tools.jar...
    plugin.jar...
    javaws.jar...
    deploy.jar...
    rt.jar...
    jsse.jar...
    charsets.jar...
    localedata.jar...warning: jdk-8u181-linux-x64.rpm: Header V3 RSA/SHA256 Signature,
rhel2.hdp3.cisco.com | SUCCESS | rc=0 >>
Preparing... #####################################################
Updating / installing...
jdk1.8-2000:1.8.0_181-fcs #####################################################
Unpacking JAR files...
    tools.jar...
```

4. Create the following files java-set-alternatives.sh and java-home.sh on admin node (rhel1):

```
vi java-set-alternatives.sh
#!/bin/bash
for item in java javac javaws jar jps javah javap jcontrol jconsole jdb; do
  rm -f /var/lib/alternatives/$item
  alternatives --install /usr/bin/$item $item /usr/java/jdk1.8.0_181-amd64/bin/$item 9
  alternatives --set $item /usr/java/jdk1.8.0_181-amd64/bin/$item
done

vi java-home.sh
export JAVA_HOME=/usr/java/jdk1.8.0_181-amd64
```

5. Make the two java scripts created above executable:

```
chmod 755 ./java-set-alternatives.sh ./java-home.sh
```

6. Copying java-set-alternatives.sh to all nodes.

```
ansible nodes -m copy -a "src=/root/java-set-alternatives.sh dest=/root/."
ansible nodes -m file -a "dest=/root/java-set-alternatives.sh mode=755"
ansible nodes -m copy -a "src=/root/java-home.sh dest=/root/."
ansible nodes -m file -a "dest=/root/java-home.sh mode=755"
```

7. Setup Java Alternatives

```
[root@rhel1 ~]# ansible all -m shell -a "/root/java-set-alternatives.sh"
```

8. Make sure correct java is setup on all nodes (should point to newly installed java path):

```
# ansible all -m shell -a "alternatives --display java | head -2"
```

9. Setup JAVA_HOME on all nodes:

```
# ansible all -m shell -a "source /root/java-home.sh"
```

10. Display JAVA_HOME on all nodes:

```
# ansible all -m command -a "echo $JAVA_HOME"
```

11. Display current java -version:

```
# ansible all -m command -a "java -version"
```

```
[root@rhel1 ~]# ansible all -m command -a "java -version"
rhel3.hdp3.cisco.com | SUCCESS | rc=0 >>
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

Configure NTP

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.

```
# ansible all -m yum -a "name=ntp state=present"
```



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 10.16.1.31
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):

```
# ansible nodes -m copy -a "src=/root/ntp.conf dest=/etc/ntp.conf"
```

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

```
# ansible all -m service -a "name=ntpd state=stopped"
# ansible all -m command -a "ntpdate rhel1.hdp3.cisco.com"
# ansible all -m service -a "name=ntpd state=started"
```

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

```
# ansible all -a "systemctl enable ntpd"
```

6. Verify NTP is up and running in all nodes by running the following commands:

```
# ansible all -a "systemctl status ntpd"
```

```
[root@rhel1 ~]# ansible all -m command -a "systemctl status ntpd"
rhel1.hdp3.cisco.com | SUCCESS | rc=0 >>
● ntpd.service - Network Time Service
  Loaded: loaded (/usr/lib/systemd/system/ntp.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2018-10-23 10:50:25 PDT; 1 months 2 days ago
    Main PID: 1401 (ntpd)
      Tasks: 1
     Memory: 4.0K
       CGroup: /system.slice/ntp.service
           └─1401 /usr/sbin/ntp -u ntp:ntp -g
```



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

7. Install the Chrony service:

```
# ansible all -m yum -a "name=chrony state=present"
```

8. Activate the Chrony service at boot:

```
# ansible all -a "systemctl enable chronyd"
```

9. Start the Chrony service:

```
# ansible all -m service -a "name=chronyd state=started"systemctl start chronyd
The Chrony configuration is in the /etc/chrony.conf file, configured similar to
/etc/ntp.conf.
```

Enable 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:

```
# ansible all -m command -a "rsyslogd -v"
# ansible all -m command -a "service rsyslog status"
```

Set 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.

To set ulimit, follow these steps:

- 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:

```
# ansible nodes -m copy -a "src=/etc/security/limits.conf dest=/etc/security/limits.conf"
```

- 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
```



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.

Disable 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`.

To disable SELinux, follow these steps:

1. The following command will disable SELinux on all nodes:

```
# ansible nodes -m shell -a "sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config"
# ansible nodes -m shell -a "setenforce 0"
```

The command (above) may fail if SELinux is already disabled. This require reboot to take effect.

2. q the machine, if needed for SELinux to be disabled in case it does not take effect. It can be checked using the following command:

```
# ansible nodes -a "sestatus"

[root@rhel1 ~]# ansible nodes -a "sestatus"
rhel5.hdp3.cisco.com | SUCCESS | rc=0 >>
SELinux status:                 disabled

rhel6.hdp3.cisco.com | SUCCESS | rc=0 >>
SELinux status:                 disabled

rhel2.hdp3.cisco.com | SUCCESS | rc=0 >>
SELinux status:                 disabled
```

Set TCP Retries

Adjusting the `tcp_retries` parameter for the system network enables faster detection of failed nodes. Given the advanced networking features of UCS, this is a safe and recommended change (failures observed at the operating system layer are most likely serious rather than transitory).

To set TCP retries, follow these steps:



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
Copy the /etc/sysctl.conf file from admin node (rhel1) to all the nodes using the following
command:
# ansible nodes -m copy -a "src=/etc/sysctl.conf dest=/etc/sysctl.conf"
```

2. Load the settings from default sysctl file `/etc/sysctl.conf` by running the following command:

```
# ansible nodes -m command -a "sysctl -p"
```

Disable 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.

```
# ansible all -m command -a "firewall-cmd --zone=public --add-port=80/tcp --permanent"
# ansible all -m command -a "firewall-cmd --reload"
```

```
# ansible all -m command -a "systemctl disable firewalld"
```

Disable Swapping

To disable swapping, follow these steps:

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:

```
# ansible all -m shell -a "echo 'vm.swappiness=1' >> /etc/sysctl.conf"
```

2. Load the settings from default sysctl file `/etc/sysctl.conf` and verify the content of `sysctl.conf`:

```
# ansible all -m shell -a "cat /etc/sysctl.conf"
# ansible all -m shell -a "cat /etc/sysctl.conf"
```

```
[root@rhel1 ~]# ansible all -m shell -a "cat /etc/sysctl.conf"
rhel3.hdp3.cisco.com | SUCCESS | rc=0 >>
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.tcp_retries2=5
vm.swappiness=1
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Disable Transparent Huge Pages

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

To disable Transparent Huge Pages, follow these steps:

1. The following commands must be run for every reboot, so copy this command to `/etc/rc.local` so they are executed automatically for every reboot:

```
# ansible all -m shell -a "echo never > /sys/kernel/mm/transparent_hugepage/enabled"
# ansible all -m shell -a "echo never > /sys/kernel/mm/transparent_hugepage/defrag"
```

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:

```
# ansible nodes -m copy -a "src=/root/thp_disable dest=/root/thp_disable"
```

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

```
# ansible nodes -m shell -a "cat /root/thp_disable >> /etc/rc.local"
```

Disable IPv6 Defaults

To disable IPv6 defaults, follow these steps:

1. Disable IPv6 as the addresses used are IPv4:

```
# ansible all -m shell -a "echo 'net.ipv6.conf.all.disable_ipv6 = 1' >> /etc/sysctl.conf"
# ansible all -m shell -a "echo 'net.ipv6.conf.default.disable_ipv6 = 1' >> /etc/sysctl.conf"
# ansible all -m shell -a "echo 'net.ipv6.conf.lo.disable_ipv6 = 1' >> /etc/sysctl.conf"
```

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

```
# ansible all -m shell -a "sysctl -p"
```

Configure Data Drives on Name Node and Other Management Nodes

This section describes the steps to configure non-OS disk drives as RAID1 using the Storcli command. All drives are 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.

To configure data drives on Name node and other nodes, follow these steps:

1. From the website download storcli **Error! Hyperlink reference not valid.**<https://www.broadcom.com/support/download-search/?pg=&pf=&pn=&po=&pa=&dk=storcli>.
2. Extract the zip file and copy storcli-1.14.12-1.noarch.rpm from the linux directory.
3. Download storcli and its dependencies and transfer to Admin node:

```
#scp storcli-1.14.12-1.noarch.rpm rhel1:/root/
```

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

```
# ansible all -m copy -a "src=/root/storcli-1.14.12-1.noarch.rpm dest=/root/."
```

5. Run this command to install storcli on all the nodes:

```
# ansible all -m shell -a "rpm -ivh storcli-1.14.12-1.noarch.rpm"
```

6. Run this command to copy storcli64 to root directory:

```
# ansible all -m shell -a "cp /opt/MegaRAID/storcli/storcli64 /root/."
```

7. Run this command to check the state of the disks:

```
# ansible all -m shell -a "./storcli64 /c0 show"
```



If the drive state shows up as JBOD, creating RAID in the subsequent steps will fail with the error “*The specified physical disk does not have the appropriate attributes to complete the requested command.*”

8. If the drive state shows up as JBOD, it can be converted into Unconfigured Good using Cisco UCSM or storcli64 command. Following steps should be performed if the state is JBOD.

9. Get the enclosure id as follows:

```
ansible all -m shell -a "./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'"

[root@rhel1 ~]# ansible all -m shell -a "./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'"
rhel1.hdp3.cisco.com | SUCCESS | rc=0 >>
  8 Enclosure Device ID: 66
  8 Enclosure position: 0

rhel2.hdp3.cisco.com | SUCCESS | rc=0 >>
  8 Enclosure Device ID: 66
  8 Enclosure position: 0
```



It is observed that some earlier versions of storcli64 complains about above mentioned command as if it is deprecated. In this case, please ./storcli64 /co show all| awk '{print \$1}'| sed -n ':[0-9]:[0-9]/p'|awk '{print substr(\$1,1,2)}'|sort -u command to determine enclosure id.

10. Convert to unconfigured good:

```
ansible datanodes -m command -a "./storcli64 /c0 /e66 /sall set good force"

[root@rhel1 ~]# ansible datanodes -m command -a "./storcli64 /c0 /e66 /sall set good force"
rhel1.hdp3.cisco.com | SUCCESS | rc=0 >>
Controller = 0
Status = Success
Description = Set Drive Good Succeeded.

rhel2.hdp3.cisco.com | SUCCESS | rc=0 >>
Controller = 0
Status = Success
Description = Set Drive Good Succeeded.
```

11. Verify status by running the following command:

```
# ansible datanodes -m command -a "./storcli64 /c0 /e66 /sall show"
[root@rhel1 ~]#
[root@rhel1 ~]# ansible datanodes -m command -a "./storcli64 /c0 /e66 /sall show"
rhel1.hdp3.cisco.com | SUCCESS | rc=0 >>
Controller = 0
Status = Success
Description = Show Drive Information Succeeded.

Drive Information :
=====



| EID:Sltn | DID | State | DG | Size  | Intf | Med | SED | PI | SeSz | Model | Sp           |   |
|----------|-----|-------|----|-------|------|-----|-----|----|------|-------|--------------|---|
| 66:1     | 44  | UGood | -  | 1.635 | TB   | SAS | HDD | N  | N    | 4 KB  | ST1800MM0129 | U |
| 66:2     | 45  | UGood | -  | 1.635 | TB   | SAS | HDD | N  | N    | 4 KB  | ST1800MM0129 | U |
| 66:3     | 42  | UGood | -  | 1.635 | TB   | SAS | HDD | N  | N    | 4 KB  | ST1800MM0129 | U |
| 66:4     | 43  | UGood | -  | 1.635 | TB   | SAS | HDD | N  | N    | 4 KB  | ST1800MM0129 | U |
| 66:5     | 41  | UGood | -  | 1.635 | TB   | SAS | HDD | N  | N    | 4 KB  | ST1800MM0129 | U |
| 66:6     | 39  | UGood | -  | 1.635 | TB   | SAS | HDD | N  | N    | 4 KB  | ST1800MM0129 | U |
| 66:7     | 38  | UGood | -  | 1.635 | TB   | SAS | HDD | N  | N    | 4 KB  | ST1800MM0129 | U |
| 66:8     | 40  | UGood | -  | 1.635 | TB   | SAS | HDD | N  | N    | 4 KB  | ST1800MM0129 | U |



EID-Enclosure Device ID|Sltn-Slot No.|DID-Device ID|DG-DriveGroup
DHS-Dedicated Hot Spare|UGood-Unconfigured Good|GHS-Global Hotspare
UBad-Unconfigured Bad|Onln-Online|Offln-Offline|Intf-Interface
Med-Media Type|SED-Self Encryptive Drive|PI-Protection Info
SeSz-Sector Size|Sp-Spin|U-Up|D-Down|T-Transition|F-Foreign
UGUnsp-Unsupported|UGShld-UnConfigured shielded|HSPShld-Hotspare shielded
CFShld-Configured shielded
```

12. Run this script as root user on rhel1 to rhel3 to create the virtual drives for the management nodes:

```
#vi /root/raid1.sh
./storcli64 -cfgldadd
$1[$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] wb ra nocachedbadbbu strpsz1024 -a0
```



The script (above) requires enclosure ID as a parameter.

13. Run the following command to get enclosure id:

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

14. Run MegaCli script:

```
#!/usr/bin/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
```



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

15. Run the following command. State should change to Online:

```
ansible namenodes -m command -a "./storcli64 /c0 /e66 /sall show"
[root@rhell1 ~]# ansible namenodes -m command -a "./storcli64 /c0 /e66 /sall show"
rhel2.hdp3.cisco.com | SUCCESS | rc=0 >>
Controller = 0
Status = Success
Description = Show Drive Information Succeeded.

Drive Information :
=====
-----
```

EID	Sltn	DID	State	DG	Size	Intf	Med	SED	PI	SeSz	Model	Sp	
66:1	43	Onln	0	1.635	TB	SAS	HDD	N	N	4	KB	ST1800MM0129	U
66:2	42	Onln	0	1.635	TB	SAS	HDD	N	N	4	KB	ST1800MM0129	U
66:3	45	Onln	0	1.635	TB	SAS	HDD	N	N	4	KB	ST1800MM0129	U
66:4	44	Onln	0	1.635	TB	SAS	HDD	N	N	4	KB	ST1800MM0129	U
66:5	41	Onln	0	1.635	TB	SAS	HDD	N	N	4	KB	ST1800MM0129	U
66:6	38	Onln	0	1.635	TB	SAS	HDD	N	N	4	KB	ST1800MM0129	U
66:7	40	Onln	0	1.635	TB	SAS	HDD	N	N	4	KB	ST1800MM0129	U
66:8	39	Onln	0	1.635	TB	SAS	HDD	N	N	4	KB	ST1800MM0129	U

16. State can also be verified in UCSM as shown below in Equipment>Rack-Mounts>Servers>Server # under Inventory/Storage/Disk tab:

Name	Size (MB)	Serial	Operability	Drive State	Presence
Disk 1	1715655	WBN05WHC0000E8236...	Operable	Online	Equipped
Disk 2	1715655	WBN06QTS0000E826M...	Operable	Online	Equipped
Disk 3	1715655	WBN047DN0000E8280...	Operable	Online	Equipped
Disk 4	1715655	WBN05WGS0000E809D...	Operable	Online	Equipped

Configure Data Drives on Data Nodes

To configure non-OS disk drives as individual RAID0 volumes using StorCli command, follow these steps. These volumes will be used for HDFS Data.

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

```
[root@rhel1 ~]# ansible datanodes -m command -a "./storcli64 -cfgeachdskraid0 WB RA direct NoCachedBadBBU strpsz1024 -a0"

rhel7.hdp3.cisco.com | SUCCESS | rc=0 >>
Adapter 0: Created VD 0
Configured physical device at Encl-66:Slot-7.
Adapter 0: Created VD 1
Configured physical device at Encl-66:Slot-6.
Adapter 0: Created VD 2
Configured physical device at Encl-66:Slot-8.
Adapter 0: Created VD 3
Configured physical device at Encl-66:Slot-5.
Adapter 0: Created VD 4
Configured physical device at Encl-66:Slot-3.
Adapter 0: Created VD 5
Configured physical device at Encl-66:Slot-4.
Adapter 0: Created VD 6
Configured physical device at Encl-66:Slot-1.
Adapter 0: Created VD 7
Configured physical device at Encl-66:Slot-2.
..... Omitted Output
24 physical devices are Configured on adapter 0.

Exit Code: 0x00
```



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

Configure the Filesystem for NameNodes and Datanodes

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

- From the Admin node, 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:



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 section [Delete Partitions](#).

```
#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##*/}
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
Run the following command to copy driveconf.sh to all the nodes:
```

```
# chmod 755 /root/driveconf.sh
# ansible datanodes -m copy -a "src=/root/driveconf.sh dest=/root/."
# ansible nodes -m file -a "dest=/root/driveconf.sh mode=755"
```

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

```
# ansible datanodes -m shell -a "/root/driveconf.sh"
```

3. Run the following from the admin node to list the partitions and mount points:

```
# ansible datanodes -m shell -a "df -h"
# ansible datanodes -m shell -a "mount"
# ansible datanodes -m shell -a "cat /etc/fstab"
```

Delete Partitions

To delete a partition, follow these steps:

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.



Do not to delete the OS partition since 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.

To verify a cluster, follow these steps:

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 Analytics \
Cluster Verification === ${NC}"
echo ""
echo ""
echo -e "${green} ===== System Information ===== ${NC}"
echo ""
echo ""
echo -e "${green}System ${NC}"
clush -a -B " `which dmidecode` |grep -A2 '^System Information'""
echo ""
echo ""
```

```

echo -e "${green}BIOS ${NC}"
clush -a -B " `which dmidecode` | grep -A3 '^BIOS I'"
echo ""
echo ""
echo -e "${green}Memory ${NC}"
clush -a -B "cat /proc/meminfo | grep -i ^memt | uniq"
echo ""
echo ""
echo -e "${green}Number of Dimms ${NC}"
clush -a -B "echo -n 'DIMM slots: ' ; `which dmidecode` |grep -c \ '^\[[[:space:]]]*Locator:'"
clush -a -B "echo -n 'DIMM count is: ' ; `which dmidecode` | grep \ "Size" | grep -c "MB"
clush -a -B " `which dmidecode` | awk '/Memory Device$/,{print}' | \ grep -e '^Mem' -e
Size: -e Speed: -e Part | sort -u | grep -v -e 'NO \ DIMM' -e 'No Module Installed' -e
Unknown"
echo ""
echo ""
# probe for cpu info #
echo -e "${green}CPU ${NC}"
clush -a -B "grep '^model name' /proc/cpuinfo | sort -u"
echo ""
clush -a -B "`which lscpu` | grep -v -e op-mode -e ^Vendor -e family -e Model: -e Stepping:
-e BogoMIPS -e Virtual -e ^Byte -e '^NUMA node(s)'"
echo ""
echo ""
# probe for nic info #
echo -e "${green}NIC ${NC}"
clush -a -B "`which ifconfig` | egrep '(^e|^p)' | awk '{print \$1}' | \ xargs -l `which
ethtool` | grep -e ^Settings -e Speed"
echo ""
clush -a -B "`which lspci` | grep -i ether"
echo ""
echo ""
# probe for disk info #
echo -e "${green}Storage ${NC}"
clush -a -B "echo 'Storage Controller: ' ; `which lspci` | grep -i -e \ raid -e storage -e
lsi"
echo ""
clush -a -B "dmesg | grep -i raid | grep -i scsi"
echo ""
clush -a -B "lsblk -id | awk '{print \$1,\$4}'|sort | nl"
echo ""
echo ""

echo -e "${green} ===== Software ===== ${NC}"
echo ""
echo ""
echo -e "${green}Linux Release ${NC}"
clush -a -B "cat /etc/*release | uniq"
echo ""
echo ""
echo -e "${green}Linux Version ${NC}"
clush -a -B "uname -srvm | fmt"
echo ""
echo ""
echo -e "${green}Date ${NC}"
clush -a -B date
echo ""
echo ""
echo -e "${green}NTP Status ${NC}"
clush -a -B "ntpstat 2>&1 | head -1"
echo ""
echo ""
echo -e "${green}SELINUX ${NC}"

```

```

clush -a -B "echo -n 'SELinux status: ' ; grep ^SELINUX= \ /etc/selinux/config 2>&1"
echo ""
echo ""
clush -a -B "echo -n 'CPUUspeed Service: ' ; `which service` cpuspeed \ status 2>&1"
clush -a -B "echo -n 'CPUUspeed 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 ${JAVA_HOME:-Not \ Defined!}'
echo ""
echo ""
echo -e "${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:

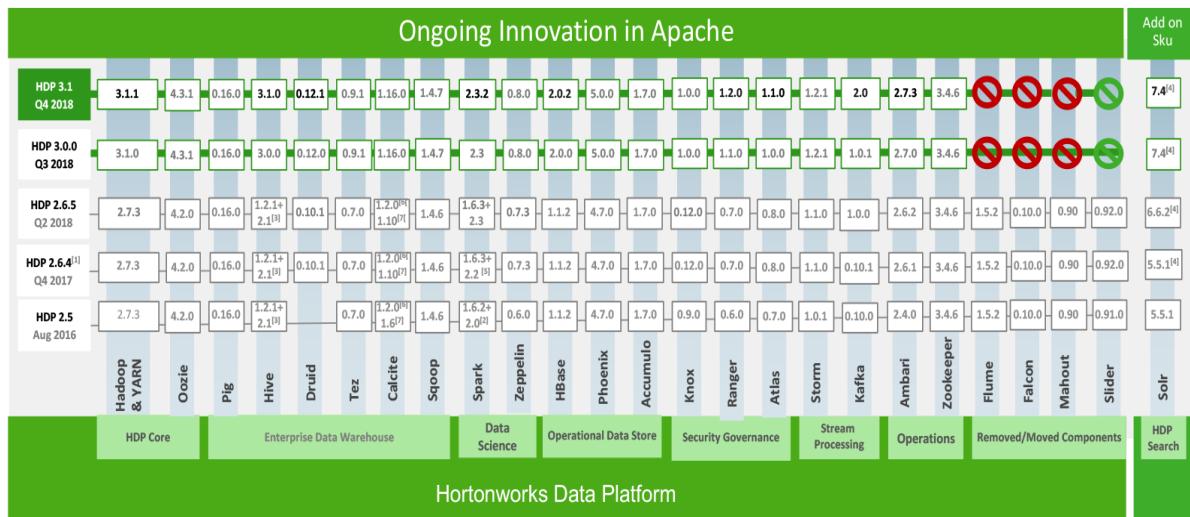
```
# chmod 755 cluster_verification.sh
```

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
```

Install HDP 3.1.0

HDP is an enterprise grade, hardened Hadoop distribution. HDP combines Apache Hadoop and its related projects into a single tested and certified package. HDP 3.1.0 components are depicted in below. This section details how to install HDP 3.1.0 on the cluster.



[1] HDP 2.6 – Shows current Apache branches being used. Final component version subject to change based on Apache release process.

[2] Spark 1.6.3+ Spark 2.1 – HDP 2.6 supports both Spark 1.6.3 and Spark 2.1 as GA.

[3] Hive 2.1 is GA within HDP 2.6.

[4] Apache Solr is available as an add-on product HDP Search.

[5] Spark 2.2 is GA

Prerequisites for HDP Installation

This section details the prerequisites for the HDP installation, such as setting up HDP repositories.

Hortonworks Repository

- From a host connected to the Internet, create a Hortonworks folder and download the Hortonworks repositories as shown below, then transfer it to the admin node:



If the admin node is connected to the internet via outbound NAT, repositories can be downloaded directly into the admin node.

```
# mkdir -p /tmp/Hortonworks/
# cd /tmp/Hortonworks
```

- Download Hortonworks HDP repo:

```
# wget http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/HDP-3.1.0.0-
centos7-rpm.tar.gz
--2018-10-13 11:02:02--  http://public-repo-
1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/HDP-3.1.0.0-centos7-rpm.tar.gz
Resolving public-repo-1.hortonworks.com (public-repo-1.hortonworks.com) ... 13.35.121.86,
13.35.121.14, 13.35.121.127, ...
Connecting to public-repo-1.hortonworks.com (public-repo-
1.hortonworks.com)|13.35.121.86|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8964079720 (8.3G) [application/x-tar]
Saving to: 'HDP-3.1.0.0-centos7-rpm.tar.gz'

100% [=====>] 8,964,079,720 50.3MB/s   in 2m 42s

2018-10-13 11:04:44 (52.9 MB/s) - 'HDP-3.1.0.0-centos7-rpm.tar.gz' saved
[8964079720/8964079720]
```

- Download Hortonworks HDP-Utils repo:

```
# wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7/HDP-UTILS-
1.1.0.22-centos7.tar.gz
--2018-10-13 11:05:30--  http://public-repo-1.hortonworks.com/HDP-UTILS-
1.1.0.22/repos/centos7/HDP-UTILS-1.1.0.22-centos7.tar.gz
Resolving public-repo-1.hortonworks.com (public-repo-1.hortonworks.com) ... 13.35.121.86,
13.35.121.127, 13.35.121.14, ...
Connecting to public-repo-1.hortonworks.com (public-repo-
1.hortonworks.com)|13.35.121.86|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 90606616 (86M) [application/x-tar]
Saving to: 'HDP-UTILS-1.1.0.22-centos7.tar.gz'

100% [=====>] 90,606,616 45.0MB/s   in 1.9s

2018-10-13 11:05:33 (45.0 MB/s) - 'HDP-UTILS-1.1.0.22-centos7.tar.gz' saved
[90606616/90606616]
```

- Download HDP-GPL repo:

```
# wget http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-centos7-gpl.tar.gz
--2018-10-13 12:10:45--  http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-centos7-gpl.tar.gz
Resolving public-repo-1.hortonworks.com (public-repo-1.hortonworks.com) ... 13.35.121.120,
13.35.121.127, 13.35.121.86, ...
Connecting to public-repo-1.hortonworks.com (public-repo-1.hortonworks.com) |13.35.121.120|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 162054 (158K) [application/x-tar]
Saving to: 'HDP-GPL-3.1.0.0-centos7-gpl.tar.gz'

100%[=====] 162,054 --.-K/s in 0.1s

2018-10-13 12:10:45 (1.27 MB/s) - 'HDP-GPL-3.1.0.0-centos7-gpl.tar.gz' saved [162054/162054]
```

5. Download the Ambari repo:

```
# wget http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.7.1.0/ambari-2.7.1.0-centos7.tar.gz
```

6. Copy the repository directory to the admin node (rhel1):

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

7. Extract the tar ball:

```
[root@rhel1 Hortonworks]# tar -zxvf HDP-3.1.0.0-centos7-rpm.tar.gz
[root@rhel1 Hortonworks]# tar -zxvf HDP-UTILS-1.1.0.22-centos7.tar.gz
[root@rhel1 Hortonworks]# tar -zxvf HDP-GPL-3.1.0.0-centos7-gpl.tar.gz
[root@rhel1 Hortonworks]# tar -zxvf ambari-2.7.1.0-centos7.tar.gz
```

8. Create HDP repo with the following contents:

```
[root@rhel1]# cat /etc/yum.repos.d/hdp.repo
[HDP-3.1.0.0]
name= Hortonworks Data Platform Version - HDP-3.1.0.0
baseurl= http://rhel1.hdp3.cisco.com/Hortonworks/HDP/centos7/3.1.0.0-187
gpgcheck=0
enabled=1
priority=1

[HDP-GPL-3.1.0.0]
name=Hortonworks GPL Version - HDP-GPL-3.1.0.0
baseurl= http://rhel1.hdp3.cisco.com/Hortonworks/HDP-GPL/centos7/3.1.0.0-187
gpgcheck=0
enabled=1
priority=1

[HDP-UTILS-1.1.0.22]
name=Hortonworks Data Platform Utils Version - HDP-UTILS-1.1.0.22
baseurl= http://rhel1.hdp3.cisco.com/Hortonworks/HDP-UTILS/centos7/1.1.0.22
gpgcheck=0
enabled=1
priority=1
```



To verify the files, go to: <http://rhel1.hdp3.cisco.com/Hortonworks>.

9. Create the Ambari repo:

```
vi /etc/yum.repos.d/ambari.repo

[Updates-ambari-2.7.1.0]
name=ambari-2.7.1.0 - Updates
baseurl=http://rhel1.hdp3.cisco.com/Hortonworks/ambari/centos7/2.7.1.0-169
gpgcheck=0
enabled=1
priority=1
From the admin node copy the repo files to /etc/yum.repos.d/ of all the nodes of the
cluster:
# ansible nodes -m copy -a "src=/etc/yum.repos.d/hdp.repo dest=/etc/yum.repos.d/."
# ansible nodes -m copy -a "src=/etc/yum.repos.d/ambari.repo dest=/etc/yum.repos.d/."
```

Downgrade Snappy on All Nodes

Downgrade snappy on all data nodes by running this command from admin node:

```
# ansible all -m command -a "yum -y downgrade snappy"
```

HDP Installation

To install HDP, complete the following the steps:

Install and Setup Ambari Server on rhel1

1. Run the following command in rhel1 to install ambary-server:

```
#yum -y install ambari-server
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to
register.
Resolving Dependencies
--> Running transaction check
--> Package ambari-server.x86_64 0:2.7.1.0-169 will be installed
--> Processing Dependency: postgresql-server >= 8.1 for package: ambari-server-2.7.1.0-
169.x86_64
--> Running transaction check
--> Package postgresql-server.x86_64 0:9.2.23-3.el7_4 will be installed
--> Processing Dependency: postgresql-libs(x86-64) = 9.2.23-3.el7_4 for package: postgresql-
server-9.2.23-3.el7_4.x86_64
--> Processing Dependency: postgresql(x86-64) = 9.2.23-3.el7_4 for package: postgresql-
server-9.2.23-3.el7_4.x86_64
--> Processing Dependency: libpq.so.5()(64bit) for package: postgresql-server-9.2.23-
3.el7_4.x86_64
--> Running transaction check
--> Package postgresql.x86_64 0:9.2.23-3.el7_4 will be installed
--> Package postgresql-libs.x86_64 0:9.2.23-3.el7_4 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

Package Repository	Arch	Size	Version

```

Installing:
  ambari-server                               x86_64           2.7.1.0-169
Updates-ambari-2.7.1.0                         353 M
Installing for dependencies:
  postgresql                                 x86_64           9.2.23-3.el7_4
rhe17.6                                         3.0 M
  postgresql-libs                            x86_64           9.2.23-3.el7_4
rhe17.6                                         234 k
  postgresql-server                          x86_64           9.2.23-3.el7_4
rhe17.6                                         3.8 M

Transaction Summary
=====
=====
Install 1 Package (+3 Dependent packages)

Total download size: 360 M
Installed size: 452 M
Downloading packages:
(1/4): postgresql-libs-9.2.23-3.el7_4.x86_64.rpm
| 234 kB  00:00:00
(2/4): postgresql-9.2.23-3.el7_4.x86_64.rpm
| 3.0 MB  00:00:00
(3/4): postgresql-server-9.2.23-3.el7_4.x86_64.rpm
| 3.8 MB  00:00:00
(4/4): ambari-server-2.7.1.0-169.x86_64.rpm
| 353 MB  00:00:03
-----
-----
Total
109 MB/s | 360 MB 00:00:03
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : postgresql-libs-9.2.23-3.el7_4.x86_64
1/4
  Installing : postgresql-9.2.23-3.el7_4.x86_64
2/4
  Installing : postgresql-server-9.2.23-3.el7_4.x86_64
3/4
  Installing : ambari-server-2.7.1.0-169.x86_64
4/4
  Verifying   : postgresql-9.2.23-3.el7_4.x86_64
1/4
  Verifying   : postgresql-libs-9.2.23-3.el7_4.x86_64
2/4
  Verifying   : postgresql-server-9.2.23-3.el7_4.x86_64
3/4
  Verifying   : ambari-server-2.7.1.0-169.x86_64
4/4

Installed:
  ambari-server.x86_64 0:2.7.1.0-169

Dependency Installed:
  postgresql.x86_64 0:9.2.23-3.el7_4                  postgresql-libs.x86_64 0:9.2.23-3.el7_4
  postgresql-server.x86_64 0:9.2.23-3.el7_4

Complete!
Install PostgreSQL in rhe12
The PostgreSQL database is used by Ambari, Hive, and Oozie services.

```

The rhel2 hosts the Hive and Oozie services and Ambari Server is installed on rhel1. To install, follow these steps:

2. Log into rhel2.
3. Install Red Hat Package Manager (RPM) according to the requirements of your operating system:

```
yum install https://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-9.6-3.noarch.rpm
```

4. Install PostgreSQL version 9.5 or later:

```
yum install postgresql96-server postgresql96-contrib postgresql96
```

5. Initialize the database as shown in the below figure by running the following command:

```
/usr/pgsql-9.6/bin/postgresql96-setup initdb
```

```
[root@rhel2 ~]#
[root@rhel2 ~]#
[root@rhel2 ~]# /usr/pgsql-9.6/bin/postgresql96-setup initdb
Initializing database ... OK

[root@rhel2 ~]#
```

6. Start PostgreSQL:

```
# systemctl enable postgresql-9.6.service
# systemctl start postgresql-9.6.service
```

7. Open /var/lib/pgsql/9.6/data/postgresql.conf and update to the following:

```
listen_addresses = '*'
```

8. Update these files on rhel2 in the location chosen to install the databases for Hive, Oozie and Ambari, using the host ip addresses:

```
[root@rhel2 ~]# cat /var/lib/pgsql/9.6/data/pg_hba.conf|tail -n 20
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
#local all all peer
# IPv4 local connections:
#host all all 127.0.0.1/32 ident
# IPv6 local connections:
host all all ::1/128 ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local replication postgres peer
#host replication postgres 127.0.0.1/32 ident
#host replication postgres ::1/128 ident

local all postgres peer
local all all md5
host all postgres,hive,oozie 10.16.1.32/24 md5
host all ambari 10.16.1.31/24 md5

[root@rhel2 ~]#
```



Before adding new entries, comment the old entries as mentioned above.

9. Restart PostgreSQL:

```
# systemctl stop postgresql-9.6.service
# systemctl start postgresql-9.6.service
```

10. Run the following:

```
sudo -u postgres psql
[root@rhel2 ~]#
[root@rhel2 ~]# sudo -u postgres psql
could not change directory to "/root": Permission denied
psql (9.6.10)
Type "help" for help.

postgres=# \q
[root@rhel2 ~]#
```



For more information about setting up PostgreSQL, go to: https://docs.hortonworks.com/HDPDocuments/Ambari-2.7.1.0/bk_ambari-installation/content/install-postgres.html

Create Database for Ambari

To create the database for Ambari, follow these steps:

1. Run the following commands mentioned below in bold to create and prepare database for Ambari:

```
[root@rhel2 ~]# sudo -u postgres psql
could not change directory to "/root": Permission denied
psql (9.6.10)
Type "help" for help.

postgres=# \dt
No relations found.
postgres=#
postgres=#
postgres=# create database ambari;
CREATE DATABASE
postgres=# create user ambari with password 'bigdata';
CREATE ROLE
postgres=# grant all privileges on database ambari to ambari;
GRANT
postgres=# \connect ambari;
You are now connected to database "ambari" as user "postgres".
ambari=# create schema ambari authorization ambari;
CREATE SCHEMA
ambari=# alter schema ambari owner to ambari;
ALTER SCHEMA
ambari=# alter role ambari set search_path to 'ambari', 'public';
ALTER ROLE
ambari=# \q
```

2. Restart PostgreSQL:

```
[[root@rhel2 ~]# systemctl restart postgresql-9.6.service
```

3. Verify the ambari user by logging into psql:

```
# psql -U ambari -d ambari

[root@rhel2 ~]#
[root@rhel2 ~]# psql -U ambari -d ambari
psql (9.6.10)
Type "help" for help.

ambari=> 
```

- Load the Ambari Server database schema:



Pre-load the Ambari database schema into your PostgreSQL database using the schema script.

- Find the Ambari-DDL-Postgres-CREATE.sql file in the /var/lib/ambari-server/resources/ directory of the Ambari Server host after you have installed Ambari Server.

```
Copy /var/lib/ambari-server/resources/ from rhel1 to rhel2:/tmp/.
[root@rhel1 ~]# scp -r /var/lib/ambari-server/resources/* rhel2:/tmp/
```

- Run the following command to launch the Ambari-DDL-Postgres-CREATE.sql script:

```
[root@rhel2 tmp]# cd /tmp

[root@rhel2 tmp]# psql -U ambari -d ambari
Password for user ambari:
psql (9.6.10)
Type "help" for help.

ambari=> \i Ambari-DDL-Postgres-CREATE.sql
CREATE TABLE
CREATE TABLE
CREATE TABLE
..... OUTPUT OMITTED ----
```

- Check the table is created by running \dt command:

```
[root@rhel2 ~]# psql -U ambari -d ambari
psql (9.6.10)
Type "help" for help.

ambari=>
ambari=>
ambari=> \dt
              List of relations
 Schema |           Name            | Type  | Owner
-----+---------------------+-----+-----
 ambari | adminpermission        | table | ambari
 ambari | adminprincipal         | table | ambari
 ambari | adminprincipaltype    | table | ambari
 ambari | adminprivilege         | table | ambari
 ambari | adminresource          | table | ambari
 ambari | adminresourcetype     | table | ambari
 ambari | alert_current          | table | ambari
 ambari | alert_definition       | table | ambari
 ambari | alert_group             | table | ambari
 ambari | alert_group_target     | table | ambari
 ambari | alert_grouping          | table | ambari
 ambari | alert_history           | table | ambari
```

- Restart PostgreSQL:

```
[root@rhel2 ~]# systemctl restart postgresql-9.6.service
```

Create Database for Hive

Run the following command as shown in bold to create and prepare database for Hive:

```
[root@rhel2 ~]# sudo -u postgres psql
could not change directory to "/root": Permission denied
psql (9.6.10)
Type "help" for help.

postgres=# create database hive;
CREATE DATABASE
postgres=# create user hive with password 'bigdata';
CREATE ROLE
postgres=# grant all privileges on database hive to hive;
GRANT
postgres=# \q
[root@rhel2 ~]#
```

Create Database for Oozie

Run the following command to create and prepare database for Oozie:

```
[root@rhel2 ~]# sudo -u postgres psql
could not change directory to "/root": Permission denied
psql (9.6.10)
Type "help" for help.

postgres=# create database oozie;
CREATE DATABASE
postgres=# create user oozie with password 'bigdata';
CREATE ROLE
postgres=# grant all privileges on database oozie to oozie;
GRANT
postgres=# \q
[root@rhel2 ~]#
```

Setup Ambari Server On Admin Node(Rhel1)

To setup the Ambari server, follow these steps:

1. Install the PostgreSQL JDBC driver:

```
[root@rhel1 2.7.1.0-169]# yum -y install postgresql-jdbc*
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
Resolving Dependencies
--> Running transaction check
----> Package postgresql-jdbc.noarch 0:9.2.1002-5.el7 will be installed
----> Processing Dependency: jpackage-utils for package: postgresql-jdbc-9.2.1002-5.el7.noarch
----> Processing Dependency: java for package: postgresql-jdbc-9.2.1002-5.el7.noarch
--> Running transaction check
----> Package java-1.8.0-openjdk.x86_64 1:1.8.0.161-2.b14.el7 will be installed
```

2. Configure Ambari server to use the JDBC driver for connectivity to Ambari database in PostgreSQL:

```
[root@rhel1 2.7.1.0-169]# ambari-server setup --jdbc-db=postgres --jdbc-driver=/usr/share/java/postgresql-jdbc.jar
```

```

Using python /usr/bin/python
Setup ambari-server
Copying /usr/share/java/postgresql-jdbc.jar to /var/lib/ambari-server/resources/postgresql-
jdbc.jar
If you are updating existing jdbc driver jar for postgres with postgresql-jdbc.jar. Please
remove the old driver jar, from all hosts. Restarting services that need the driver, will
automatically copy the new jar to the hosts.
JDBC driver was successfully initialized.
Ambari Server 'setup' completed successfully.

```

3. Setup Ambari Server by running the following command:

```

[root@rhel1 ~]# ambari-server setup -j $JAVA_HOME
Using python /usr/bin/python
Setup ambari-server
Checking SELinux...
SELinux status is 'disabled'
Customize user account for ambari-server daemon [y/n] (n)? n
Adjusting ambari-server permissions and ownership...
Checking firewall status...
Checking JDK...
WARNING: JAVA_HOME /usr/java/jdk1.8.0_181-amd64 must be valid on ALL hosts
WARNING: JCE Policy files are required for configuring Kerberos security. If you plan to use
Kerberos, please make sure JCE Unlimited Strength Jurisdiction Policy Files are valid on all
hosts.
Check JDK version for Ambari Server...
JDK version found: 8
Minimum JDK version is 8 for Ambari. Skipping to setup different JDK for Ambari Server.
Checking GPL software agreement...
GPL License for LZO: https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html
Enable Ambari Server to download and install GPL Licensed LZO packages [y/n] (n)? y
Completing setup...
Configuring database...
Enter advanced database configuration [y/n] (n)? y
Configuring database...
=====
Choose one of the following options:
[1] - PostgreSQL (Embedded)
[2] - Oracle
[3] - MySQL / MariaDB
[4] - PostgreSQL
[5] - Microsoft SQL Server (Tech Preview)
[6] - SQL Anywhere
[7] - BDB
=====
Enter choice (4):
Hostname (rhel2.hdp3.cisco.com):
Port (5432):
Database name (ambari):
Postgres schema (ambari):
Username (ambari):
Enter Database Password (bigdata):
Configuring ambari database...
Configuring remote database connection properties...
WARNING: Before starting Ambari Server, you must run the following DDL against the database
to create the schema: /var/lib/ambari-server/resources/Ambari-DDL-Postgres-CREATE.sql
Proceed with configuring remote database connection properties [y/n] (y)? y
Extracting system views...
.....
Adjusting ambari-server permissions and ownership...
Ambari Server 'setup' completed successfully.

```

4. Start the Ambari Server:

```
[root@rhel1 ~]# ambari-server start
```

```
[root@rhel1 ~]# ambari-server start
Using python /usr/bin/python
Starting ambari-server
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Ambari database consistency check started...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start.....
Server started listening on 8080

DB configs consistency check: no errors and warnings were found.
Ambari Server 'start' completed successfully.
[root@rhel1 ~]# ^C
[root@rhel1 ~]#
```

5. To check status of Ambari Server, run the following command:

```
# ambari-server status
```

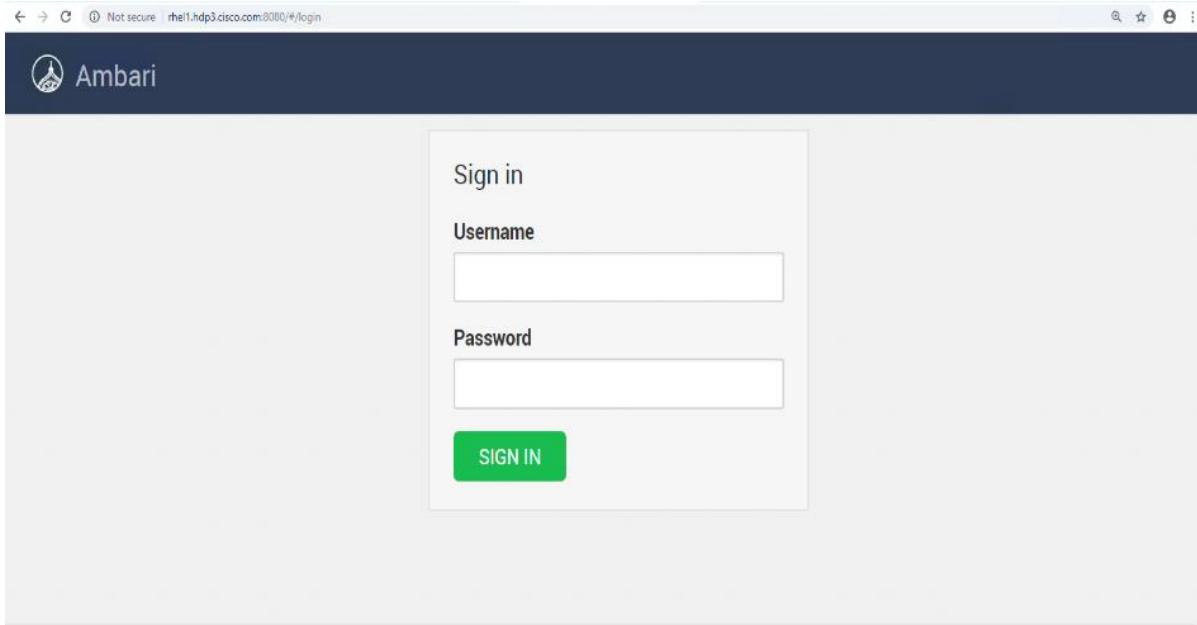
```
[root@rhel1 ~]# ambari-server status
Using python /usr/bin/python
Ambari-server status
Ambari Server running
Found Ambari Server PID: 65658 at: /var/run/ambari-server/ambari-server.pid
[root@rhel1 ~]#
```

Launch the Ambari Server

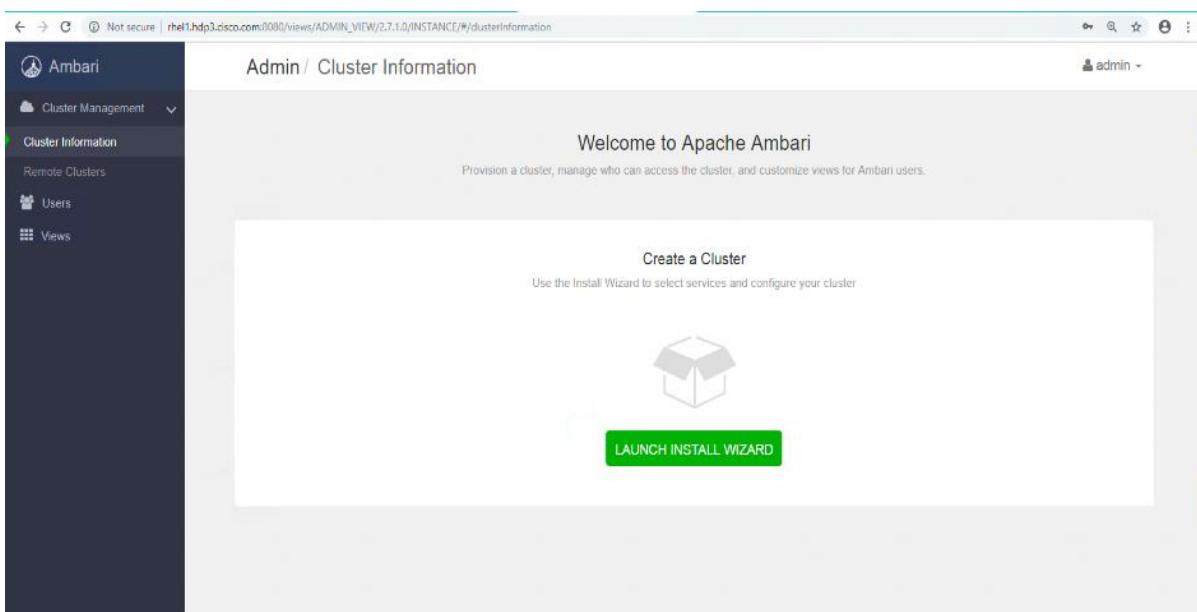
When the Ambari service starts, access the Ambari Install Wizard through the browser. To launch the Ambari server, follow these steps:

1. Point the browser to <http://<ip address for rhel1>:8080> or <http://rhel1.hdp3.cisco.com:8080>

The Ambari Login screen opens.



2. Log into the Ambari Server using the default username/password: **admin/admin**. This can be changed at a later period of time.
3. When logged in the "Welcome to Apache Ambari" window opens.



Create the Cluster

To create the cluster, follow these steps:

1. To create a cluster, click "LAUNCH INSTALL WIZARD."
2. From the Get started page type "Cisco_HDP" for the name for the cluster.
3. Click Next.

Get Started

This wizard will walk you through the cluster installation process. First, start by naming your cluster.

Name your cluster [Learn more](#)

Cisco_HDP

Select Version

To select the version, follow these steps:

1. In the Select Version section, choose the HDP 3.1.0. version.

Select Version

Select the software version and method of delivery for your cluster.

HDP-3.0

Service	Version
Accumulo	1.7.0
Infra Solr	0.1.0
Ambari Metrics	0.1.0
Atlas	1.0.0
Druid	0.12.1
HBase	2.0.0

Repositories

Using a Public Repository requires Internet connectivity. Using a Local Repository requires you have configured the software in a repository available in your network.

Use Public Repository Use Local Repository

Provide Base URLs for the Operating Systems you are configuring.

2. Under Repositories, select "Use Local Repository."
3. Update the Redhat 7 HDP-3.0 URL to <http://rhel1.hdp3.cisco.com/Hortonworks/HDP/centos7/3.1.0.0-187/>

4. Update the Redhat 7 HDP-3.0-GPL URL to <http://rhel1.hdp3.cisco.com/Hortonworks/HDP-GPL/centos7/3.1.0.0-187/>
5. Update the Redhat 7 HDP-UTILS-1.1.0.22 to <http://rhel1.hdp3.cisco.com/Hortonworks/HDP-UTILS/centos7/1.1.0.22/>

	HDP-3.0	http://rhel1.hdp3.cisco.com/Hortonworks/HDP/centos7/3.0.1.0-187/	
redhat7	HDP-3.0-GPL	http://rhel1.hdp3.cisco.com/Hortonworks/HDP-GPL/centos7/3.1.0.0-187/	
	HDP-UTILS-1.1.0.22	http://rhel1.hdp3.cisco.com/Hortonworks/HDP-UTILS/centos7/1.1.0.22/	



Make sure there are no trailing spaces after the URLs.

Select Hosts

To build up the cluster, you need to provide the general information about how you want to set up the cluster. This requires providing the Fully Qualified Domain Name (FQDN) of each of the hosts. You also need to provide access to the private key file that was created in Set Up Password-less SSH; this is used to locate all the hosts in the system and to access and interact with them securely.

To select hosts, follow these steps:

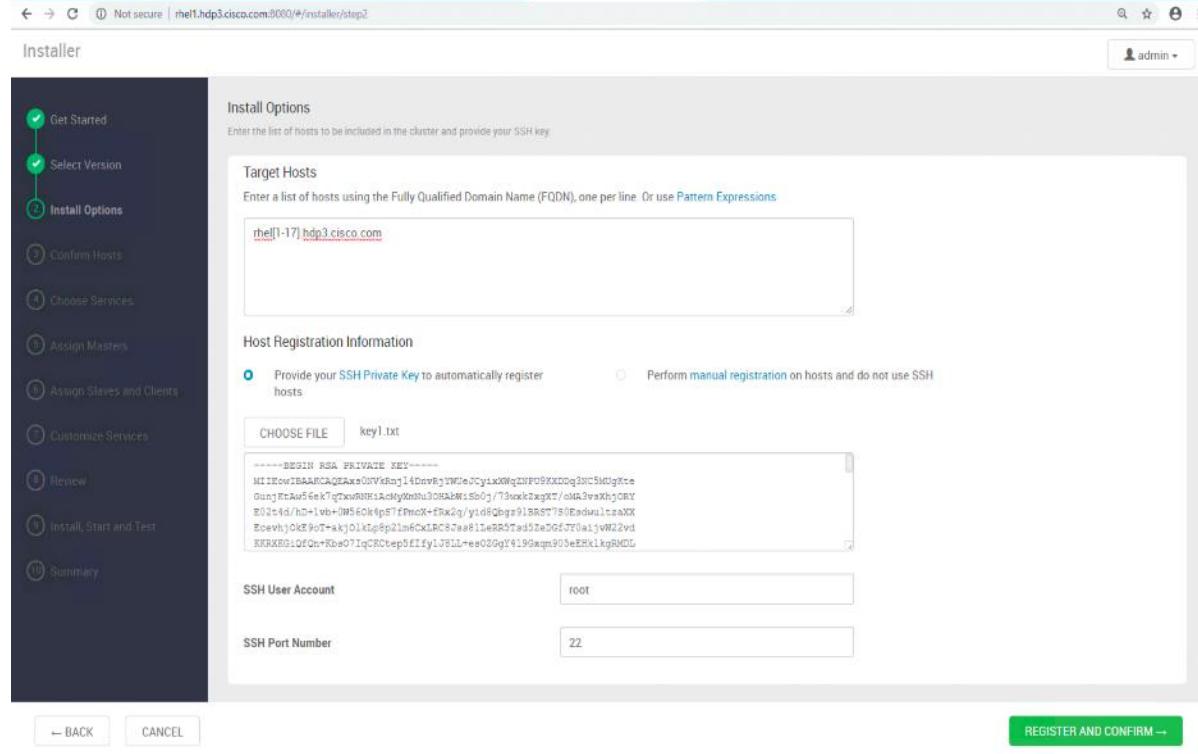
1. Use the **Target Hosts** text box to enter the list of host names, one per line. Ranges inside brackets can also be used to indicate larger sets of hosts.
2. Select the option **Provide your SSH Private Key** in the Ambari cluster install wizard.
3. Copy the contents of the file `/root/.ssh/id_rsa` on `rhel1` and paste it in the text area provided by the Ambari cluster install wizard.



Make sure there is no extra white space after the text----END RSA PRIVATE KEY----

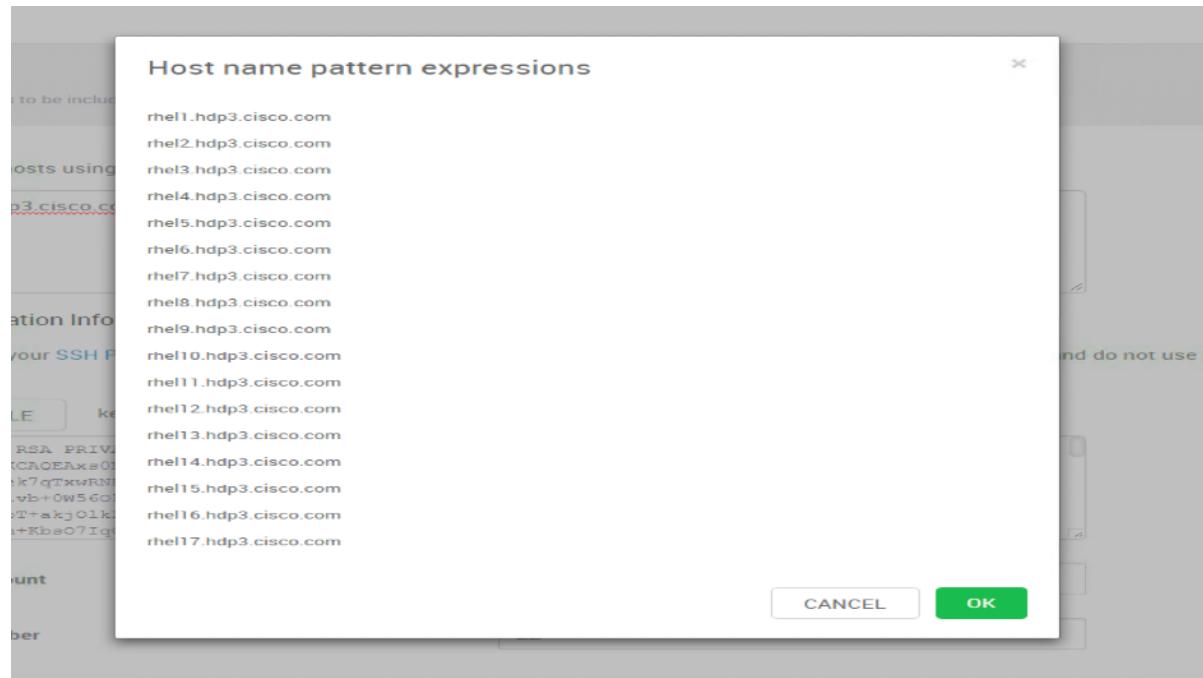
```
[root@rhel1 ~]# cat /root/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQEAYDoIRbk4mBzrizec0/g0M2iYT2h4vxkIx/A/uvQVPthFreUdgT
zehw/Qtck7meeqhgqsHmb1CriF0m6SxvPEXW2cGoAx75hZwTuDIR3Q1vk6oYUmDW
BKq5TMfUMKfd7tKnkGkg5N+YHsPCoNILLz/wqco1hzzotCmrxeRnPgs1JY74/Db
AOBewMuNajAoVppPD6cLGf6/NKORpEDUncuwe5pcRV5tko+gzBeBF5oeCS6Ya6I7
ns0Hp1JXV0Mv23SNUw13csqbqLdrr3atG6YRieVrmmr/P1rKMp192tzQ1mhZMBqG
w1RJTILjygwoqgp5g7NQBGeM7sX4V60mzv4vmzwIBIWKAQEAq4+UEI+o2PjKVCuX
2h+xEWmuXCJ3KoNEyBpr2n7KxcKyAs/8oLN6B1pYROUB3x2YZVc6hBwuLI+JDMK
hrGNMALqwDjthu1oyX/9HD1mlDyTo9k8LvPY2q8zqvHnJ+3Jisi92Dspc01xRRxQ
wnpofjAm1CDx5Wxp4MZYX9HynCcKmheFefobLys6g1oxd84eHW1y6boxU1dh7hsQ
pck+xpdbfshYFBvcKTuCHUAezF4+uBT5F0PMid7PwzrvbxKA65ABuezv9gg2/I1
FekIkRvbosniFBUi2Z0s1uN/gsa2gmsQ9gTarJ1V8zMy6K31LETcockl2LZHRX2
5sEx6wKBgQD9CiKc0HFiu1rqWW5cLTDJU8wzTiNK4M91qb2Lohffuzf1uiA13Ref
yil9MjE3A5Mnn9pcRxMmXPF4t9iuLh3+3tCsr1TzFm14WT+Fipa9sh+3JZ2HKgm
pcquAEdeoRK4oP3/yYQg95gie2Sc9sBoz6zvohdyNUvnkiMb9vw13wKBgQDKiyTi
Yu4210wsyKF27YjomjRUKF4CKtnyJy1Sm3wFPnRzJd4BuamQ0DaTxr2tw4si+4
t88M8xs6FHGHymSsqRtL0tYzM1nmwutjCLNZQfqSeg1NovekxxcLoiUze18FL3zOH
Aebj0/GLQ3SF/PGWMokCwNtaJov/x1dBdi sqEQKBgEERPbm8UVF3Nz9ZYVqtMYo
o9Ktsu3Ex52x0ad1VpHt5Tssmo1kv06TEE+8cw41fzx5j+vXwxh+bjozBj30/Dwc
GGGbrQbrKscs5HLL3z5+QqtWePB4hiqnukvnVvHP1QMJA6S53YxCdz7KH1ypnqq
bkWQFKhW2QEIIUivDKuR1AoGASzr/EKIATufb5GdbjOn4V3Y6Gb7kY3DvNs1BhSm
rk7ADAdTnzX5Nz3L08gAf9Tws+ppfx+zTfNIinOMFmNY1Y9EpyJs0S/1adLEoroWu
sc8J8bu/5RNWk8z+z9s5zwUrd5txT2cY1J8t1KQGtWyUPxoVoe/ccfENA5LP872s
xnsCgYAFRE4SbB416p9miR1+gNCi ihm9N+FmHMmcP/y80QL/MoAYoHB1Tn8cwVu
l+sju4bWGuzvnGMWXwpEU5zVBra+yShh309IwjP/1kpCNWz7Cx+/uI6FY+s1ZxTr
t5P/AvhovUKMhRFjXFQoY5yqNUkasvIu6s8q1unl8N2ihEgw1g==
-----END RSA PRIVATE KEY-----
```

4. Click the **Register and Confirm** to continue.

Figure 36 Install Options

Hostname Pattern Expressions

1. Click OK on the Host Name Pattern Expressions popup.



Confirm Hosts

Confirm Hosts helps ensure that Ambari has located the correct hosts for the cluster and checks those hosts to make sure they have the correct directories, packages, and processes to continue the install.

To confirm host, follow these steps:

1. If any host was selected in error, remove it by selecting the appropriate checkboxes and clicking the grey **Remove Selected** button.
2. To remove a single host, click the small white **Remove** button in the Action column.
3. When the list of hosts is confirmed, click **Next**.

Host	Progress	Status	Action
rhe1.hdp3.cisco.com	Green Bar	Success	Remove
rhe2.hdp3.cisco.com	Green Bar	Success	Remove
rhe3.hdp3.cisco.com	Green Bar	Success	Remove
rhe4.hdp3.cisco.com	Green Bar	Success	Remove
rhe5.hdp3.cisco.com	Green Bar	Success	Remove
rhe6.hdp3.cisco.com	Green Bar	Success	Remove
rhe7.hdp3.cisco.com	Green Bar	Success	Remove
rhe8.hdp3.cisco.com	Green Bar	Success	Remove
rhe9.hdp3.cisco.com	Green Bar	Success	Remove
rhe10.hdp3.cisco.com	Green Bar	Success	Remove
rhe11.hdp3.cisco.com	Green Bar	Success	Remove

16 Other Registered Hosts

Please wait while the hosts are being checked for potential problems.

Choose Services

HDP is made up of a number of components. Go to Hortonworks [Understand the Basics](#) for more information.

To choose services, follow these steps:

1. Select **all** to preselect all items.
2. When you have made your selections, click **Next**.

The screenshot shows the Ambari Installation Wizard at Step 4: Choose Services. The sidebar on the left lists the following steps:

- Get Started (Completed)
- Select Version (Completed)
- Install Options (Completed)
- Confirm Hosts (Completed)
- Choose Services** (Current Step)
- Assign Masters
- Assign Slaves and Clients
- Customize Services
- Review
- Install, Start and Test
- Summary

The main content area has two sections:

- Choose File System**: A table showing the selected service "HDFS" with version 3.1.1 and description "Apache Hadoop Distributed File System".
- Choose Services**: A table listing various services with their versions and descriptions. Services listed include YARN + MapReduce2 (version 3.1.1), Tez (version 0.9.1), Hive (version 3.1.0), HBase (version 2.0.0), Pig (version 0.16.0), Sqoop (version 1.4.7), and Oozie (version 4.3.1).

Assign Masters

The Ambari installation wizard attempts to assign the master nodes for various services that have been selected to appropriate hosts in the cluster, as listed in Table 9. The right column shows the current service assignments by host, with the hostname and its number of CPU cores and amount of RAM indicated.

1. Reconfigure the service assignments to match Table 9.

Table 9 Reconfigure the Service Assignments

Service Name	Host
NameNode	rhel1, rhel3 (HA)
SNameNode	rhel2
History Server	rhel2
App Timeline Server	rhel2
Resource Manager	rhel2, rhel3 (HA)
Hive Metastore	rhel2
WebHCat Server	rhel2
HiveServer2	rhel2
HBase Master	rhel2
Oozie Server	rhel1
Zookeeper	rhel1, rhel2, rhel3
Spark History Server	rhel2
SmartSense HST Server	rhel1

Service Name	Host
Grafana	rhel1
Atlas Metadata Server	rhel2
Metrics Collector	rhel1

2. Click **Next**.

Assign Slaves and Clients

The Ambari install wizard attempts to assign the slave components (DataNodes, NFSGateway, NodeManager, RegionServers, Supervisor, and Client) to appropriate hosts in the cluster.

To assign slaves and clients, follow these steps:

1. Reconfigure the service assignment to match the values shown in Table 10.
2. Assign DataNode, NodeManager, RegionServer, and Supervisor on nodes rhel3- rhel28.
3. Assign Client to all nodes.
4. Click Next.

Table 10 Services and Hostnames

Client Service Name	Host
DataNode	rhel4-rhel16; rhel24-rhel31
NFSGateway	rhel1
NodeManager	rhel4-rhel16; rhel24-rhel31
RegionServer	rhel4-rhel13; rhel24-rhel31
Supervisor	rhel4-rhel13; rhel24-rhel31
Client	All nodes, rhel1-rhel31
Submarine Nodes	rhel13-rhel16
CDSW Nodes	rhel18-rhel23

Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on. Hosts that are assigned master components are shown with *.

Client will install HDFS Client, YARN Client, MapReduce2 Client, Tez Client, Hive Client, HBase Client, Pig Client, Sqoop Client, Oozie Client, ZooKeeper Client, Accumulo Client, Infra Solr Client, Atlas Metadata Client and Spark2 Client.

Host	all none	all none	all none	all none	all none	all none	all none	all none	all none	all none	all none						
rhel1.hdp3.cisco.com*	<input type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input type="checkbox"/>	NodeManager	<input type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input type="checkbox"/>	Supervisor	<input type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel2.hdp3.cisco.com*	<input type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input type="checkbox"/>	NodeManager	<input type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input type="checkbox"/>	Supervisor	<input type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel3.hdp3.cisco.com*	<input type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input type="checkbox"/>	NodeManager	<input type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input type="checkbox"/>	Supervisor	<input type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel4.hdp3.cisco.com	<input checked="" type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input checked="" type="checkbox"/>	Supervisor	<input checked="" type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel5.hdp3.cisco.com	<input checked="" type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input checked="" type="checkbox"/>	Supervisor	<input checked="" type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel6.hdp3.cisco.com	<input checked="" type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input checked="" type="checkbox"/>	Supervisor	<input checked="" type="checkbox"/>	Accumulo TServer	<input checked="" type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel7.hdp3.cisco.com	<input checked="" type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input checked="" type="checkbox"/>	Supervisor	<input checked="" type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel8.hdp3.cisco.com	<input checked="" type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input checked="" type="checkbox"/>	Supervisor	<input checked="" type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel9.hdp3.cisco.com	<input checked="" type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input checked="" type="checkbox"/>	Supervisor	<input checked="" type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel10.hdp3.cisco.com	<input checked="" type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input checked="" type="checkbox"/>	Supervisor	<input checked="" type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel11.hdp3.cisco.com	<input checked="" type="checkbox"/>	DataNode	<input type="checkbox"/>	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input checked="" type="checkbox"/>	Supervisor	<input checked="" type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>
rhel12.hdp3.cisco.com	<input checked="" type="checkbox"/>	DataNode	<input checked="" type="checkbox"/>	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input checked="" type="checkbox"/>	Supervisor	<input checked="" type="checkbox"/>	Accumulo TServer	<input type="checkbox"/>	Ranger Tagsync	<input type="checkbox"/>

BACK **CANCEL** **NEXT →**

Customize Services

This section shows the tabs that manage configuration settings for Hadoop components. The wizard attempts to set reasonable defaults for each of the options here, but this can be modified to meet specific requirements. The following sections provide configuration guidance that should be refined to meet specific use case requirements.

The following changes need to be made:

- Memory and service level settings for each component and service level tuning.
- Customize the log locations of all the components to make sure growing logs do not cause the SSDs to run out of space.

Credentials

Specify the credentials as per your organizational policies for services in CREDENTIALS tab as shown below:

Databases

In the DATABASES tab, to configure the database for DRUID, HIVE, OOZIE, and RANGER, follow these steps:

1. Configure DRUID as shown below:

2. Change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

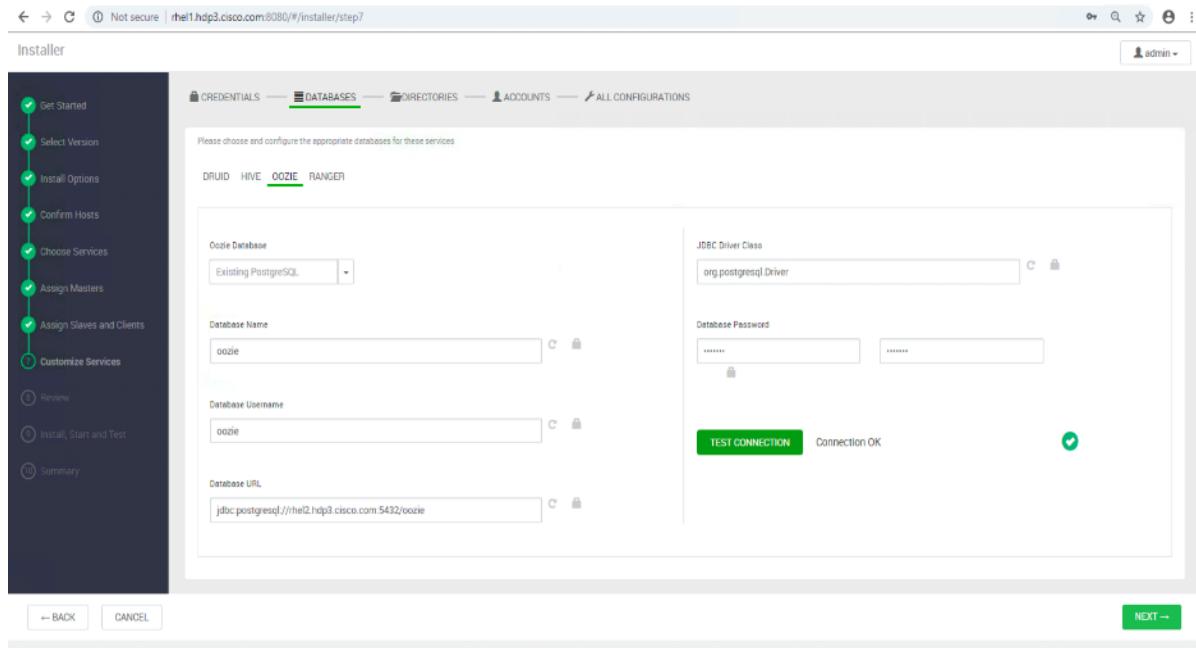
druid_log_dir	/data/disk1/log/druid
Druid PID dir	/var/run/druid

3. Configure HIVE:

- a. Select Existing PostgreSQL database in the Hive Database drop-down list.
- b. Enter Database Name as hive
- c. Enter Database Username as hive
- d. Enter Database URL as jdbc:postgresql://rhel2.hdp3.cisco.com/hive
- e. Enter Database password (use the password created during hive database setup in earlier steps; for example, big-data)
- f. Click TEST CONNECTION to verify the connectivity
- g. In Advanced tab, Change the default log location by filtering the Log Dir property and modifying the /var prefix to /data/disk1.
- h. Change the WebHCat log directory by filtering the Log Dir property and modifying the /var prefix to /data/disk1.

4. Configure OOZIE:

- a. Select Existing PostgreSQL database in the Hive Database drop-down list.
- b. Enter Database Name as oozie.
- c. Enter Database Username as oozie.
- d. Enter Database URL as jdbc:postgresql://rhel2.hdp3.cisco.com/oozie.
- e. Enter Database password (use the password created during hive database setup in earlier steps; for example, big-data).
- f. Click TEST CONNECTION to verify the connectivity.
- g. In Advanced tab, Change the default log location by filtering the Log Dir property and modifying the /var prefix to /data/disk1.



Advanced oozie-env

Oozie Log Dir

5. Configure RANGER:

- a. Select POSTGRES from DB FLAVOR drop-down list.
- b. Provide a Ranger DB name. For example, ranger.
- c. Provide Ranger DB Username such as rangeradmin.
- d. Enter JDBC connect string for a Ranger Database as jdbc:postgresql://rhel2.hdp3.cisco.com:5432/ranger.
- e. Ranger DB Host as rhel2.hdp3.cisco.com.
- f. Enter Ranger DB password. (Ranger database is not previously created. provide password string that would be configured in the DB. For example, bigdata).
- g. Select "Yes" for Setup Database and Database User.
- h. Enter "postgres" in Database Administrator (DBA) username.
- i. Enter Database Administrator (DBA) password.
- j. Enter jdbc:postgresql://rhel2.hdp3.cisco.com:5432/postgres in JDBC connect string for root user.
- k. Update Ranger Admin Log Dir from /var to /data/disk1.

HDFS

- In Ambari, select the HDFS Service tab and use the "Search" box to filter for the properties mentioned in Table 11 and update their values.
- Update the following HDFS configurations in Ambari.

Table 11 HDFS Configurations in Ambari

Property Name	Value
NameNode Java Heap Size	4096
Hadoop maximum Java heap size	4096
DataNode maximum Java heap size	4096
Datanode failed disk tolerance	5

- Change the default log location by filtering the Log Dir property and modifying the /var prefix to /data/disk1.

Not secure | rhe1.hdp3.cisco.com:8080/#/installer/step7

Installer

CREDENTIALS DATABASES DIRECTORIES ACCOUNTS ALL CONFIGURATIONS

HDFS YARN MAPREDUCE2 TEZ HIVE HBASE PIG SQOOP OOZIE ZOOKEEPER STORM ACCUMULO INFRA SOLR AMBARI METRICS ATLAS KAFKA KN > 16

Default (17) Filter...

SETTINGS ADVANCED

NameNode

NameNode directories: /data/disk1/hadoop/hdfs/namenode

NameNode Java heap size: 4096 MB

NameNode Server threads: 1400

Minimum replicated blocks %: 100%

DataNode

DataNode directories: /data/disk1/hadoop/hdfs/data, /data/disk2/hadoop/hdfs/data, /data/disk3/hadoop/hdfs/data, /data/disk4/hadoop/hdfs/data, /data/disk5/hadoop/hdfs/data, /data/disk6/hadoop/hdfs/data, /data/disk7/hadoop/hdfs/data, /data/disk8/hadoop/hdfs/data

DataNode failed disk tolerance: 3

DataNode maximum Java heap size: 4096 MB

NameNode

NameNode directories: /data/disk1/hadoop/hdfs/namenode

NameNode Java heap size: 4096 MB

NameNode Server threads: 1400

DataNode

DataNode directories: /data/disk1/hadoop/hdfs/data, /data/disk2/hadoop/hdfs/data, /data/disk3/hadoop/hdfs/data, /data/disk4/hadoop/hdfs/data, /data/disk5/hadoop/hdfs/data, /data/disk6/hadoop/hdfs/data, /data/disk7/hadoop/hdfs/data, /data/disk8/hadoop/hdfs/data

DataNode failed disk tolerance: 3

DataNode maximum Java heap size: 4096 MB

General

WebHDFS enabled:

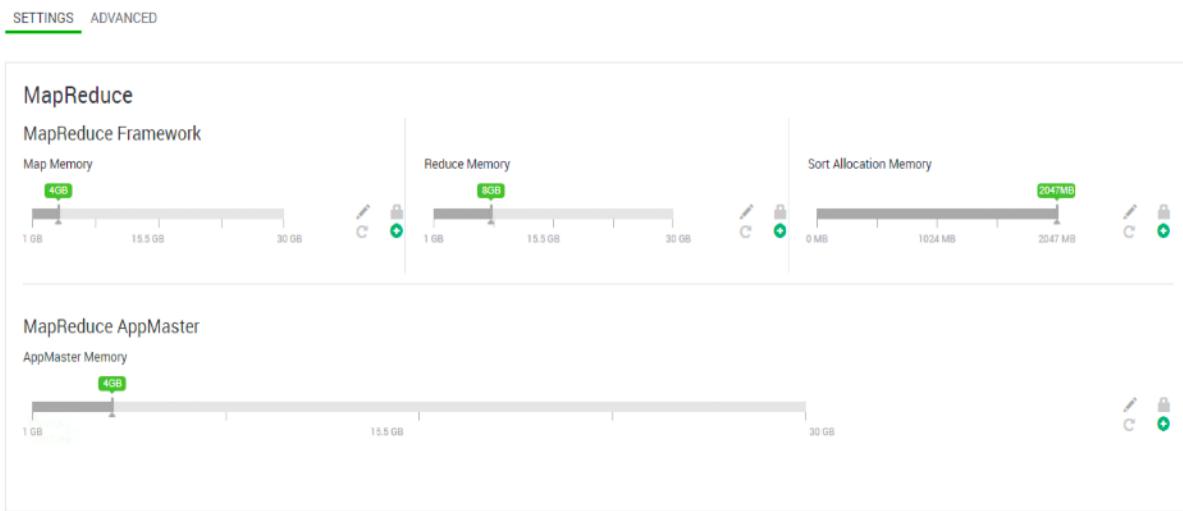
Hadoop maximum Java heap size: 4096 MB

Advanced hadoop-env

Hadoop PID Dir Prefix	/var/run/hadoop
Hadoop Root Logger	INFO,RFA
Hadoop Log Dir Prefix	/data/disk1/log/hadoop

MapReduce2

1. In Ambari, choose the MapReduce Service tab and update the values as shown below.
2. Under the MapReduce2 tab, change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.



Advanced mapred-env

Mapreduce Log Dir Prefix	/data/disk1/log/hadoop-mapreduce
Mapreduce PID Dir Prefix	/var/run/hadoop-mapreduce

YARN

1. In Ambari, select the YARN Service, and update the following as shown in Table 12.

Table 12 YARN Configuration

Property Name	Value
ResourceManager Java heap size	4096
NodeManager Java heap size	2048
YARN Java heap size	4096

The screenshot shows the Ambari web UI with three tabs: Resource Manager, Node Manager, and Application Timeline Server. Under each tab, there is a 'General' section with a 'Java heap size' input field. The values are: Resource Manager (4096 MB), Node Manager (2048 MB), and YARN (4096 MB).

Component	Java heap size
Resource Manager	4096 MB
Node Manager	2048 MB
YARN	4096 MB

2. Under YARN tab, change the default log location by filtering the Log Dir property and modifying the /var prefix to /data/disk1.

The screenshot shows the 'YARN Log Dir Prefix' configuration set to '/data/disk1/log/hadoop-yarn'. The 'YARN PID Dir Prefix' is shown below it as '/var/run/hadoop-yarn'.

Property	Value
YARN Log Dir Prefix	/data/disk1/log/hadoop-yarn
YARN PID Dir Prefix	/var/run/hadoop-yarn



YARN requires other configurations such as config group, node labeling, enabling docker runtime, CPU/GPU scheduling and isolation, and so on, which can be found in section [High Availability for HDFS NameNode and YARN ResourceManager](#).



High Availability for NameNode and YARN Resource Manager can be configured using Ambari or also on non-Ambari clusters. This deployment guide explains the configuration of high availability using Ambari – Use the Ambari wizard interface to configure HA of the components.

The Ambari web UI provides a wizard-driven user experience that allows to configure high availability of the components in many Hortonworks Data Platform (HDP) stack services. The high availability of the components are achieved by setting up primary and secondary components. In the event that the primary component fails or becomes unresponsive, services failover to secondary component. After configuring the high availability for a service, Ambari enables you to manage and disable (roll back) the high availability of components within that service.

Configure the HDFS NameNode High Availability

The HDFS NameNode high availability feature enables you to run redundant NameNodes in the same cluster in an Active/Passive configuration with a hot standby. This eliminates the NameNode as a potential single point of failure (SPOF) in an HDFS cluster. With the release of Hadoop 3.0, you can configure more than one backup NameNode.

Prior to the release of Hadoop 2.0, the NameNode represented a single point of failure (SPOF) in an HDFS cluster. Each cluster had a single NameNode, and if that machine or process became unavailable, the cluster as a whole would be unavailable until the NameNode was either restarted or brought up on a separate machine. This situation impacted the total availability of the HDFS cluster in two major ways:

- In the case of an unplanned event such as a machine crash, the cluster would be unavailable until an operator restarted the NameNode.
- Planned maintenance events such as software or hardware upgrades on the NameNode machine would result in periods of cluster downtime.

HDFS NameNode High Availability avoids this by facilitating either a fast failover to one or more backup NameNodes during machine crash, or a graceful administrator-initiated failover during planned maintenance.



Secondary NameNode is not required in high availability configuration because the Standby node also performs the tasks of the Secondary NameNode.

HBase

Under the HBase tab, change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

HBase Log Dir Prefix

/data/disk1/log/hbase

Zookeeper

Under the Zookeeper tab, change the default log location by filtering the Log Dir property and modifying the /var prefix to /data/disk1.

Advanced zookeeper-env

ZooKeeper Log Dir

/data/disk1/log/zookeeper

ZooKeeper PID Dir

/var/run/zookeeper

Storm

Under the Storm tab, change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

Advanced storm-env

Storm Log directory

/data/disk1/log/storm

Storm PID directory

/var/run/storm

Ambari Metrics

1. Choose the Ambari Metrics Service and expand the General tab and make the changes shown below.
2. Enter the Grafana Admin password as per organizational policy.
3. Change the default log location for Metrics Collector, Metrics Monitor and Metrics Grafana by finding the Log Dir property and modifying the /var prefix to /data/disk1.
4. Change the default data dir location for Metrics Grafana by finding the data Dir property and modifying the /var prefix to /data/disk1.

General

Metrics Service operation mode	embedded
Metrics Collector log dir	/data/disk1/log/ambari-metrics-collector
Metrics Collector pid dir	/var/run/ambari-metrics-collector
Metrics Monitor log dir	/data/disk1/log/ambari-metrics-monitor
Metrics Monitor pid dir	/var/run/ambari-metrics-monitor
Grafana Admin Username	admin
Grafana Admin Password

Advanced ams-grafana-env

Metrics Grafana data dir	/data/disk1/lib/ambari-metrics-grafana
Metrics Grafana log dir	/data/disk1/log/ambari-metrics-grafana
Metrics Grafana pid dir	/var/run/ambari-metrics-grafana

Advanced ams-hbase-env

HBase Log Dir Prefix	/data/disk1/log/ambari-metrics-collector
----------------------	--

Accumulo

Select Accumulo Service and change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

Advanced accumulo-env

Accumulo Log Dir	/data/disk1/log/accumulo
------------------	--------------------------

Atlas

Under the Atlas tab, change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

Advanced atlas-env

Metadata Data directory	/var/lib/atlas/data
Metadata Log directory	/data/disk1/log/atlas
Metadata PID directory	/var/run/atlas

Kafka

Under the Kafka tab, change the default log location by finding the Log Dir property and modifying the /var prefix to /data/disk1.

Advanced kafka-env

Kafka Log directory	/data/disk1/log/kafka
---------------------	-----------------------

Knox

1. Select the Knox Service tab and expand the Knox gateway tab and make the changes shown below.
2. Enter the Knox Master Secret password as per organizational policy.
3. For Knox, change the gateway port to 8444 to ensure no conflicts with local HTTP server.

Knox Gateway

Knox Gateway host	rhel1.hdp3.cisco.com
Knox Master Secret

Advanced gateway-site

gateway.port	8444
--------------	------

SmartSense

The SmartSense account requires the Hortonworks support subscription. Subscribers can populate the properties as shown below:

SmartSense Account	Local Storage
Customer account name <input type="text" value="unspecified"/>	Bundle storage directory <input type="text" value="/var/lib/smartsense/hst-server/data"/>
SmartSense ID <input type="text" value="unspecified"/>	Server temporary data directory <input type="text" value="/var/lib/smartsense/hst-server/tmp"/>
Notification Email <input type="text" value="unspecified"/>	Agent temporary data directory <input type="text" value="/var/lib/smartsense/hst-agent/data/tmp"/>
Enable Flex Subscription <input checked="" type="checkbox"/> No	

Spark

Select the Spark tab, change the default log location by finding the Log Dir property and modifying the `/var` prefix to `/data/disk1`.

Advanced livy2-env

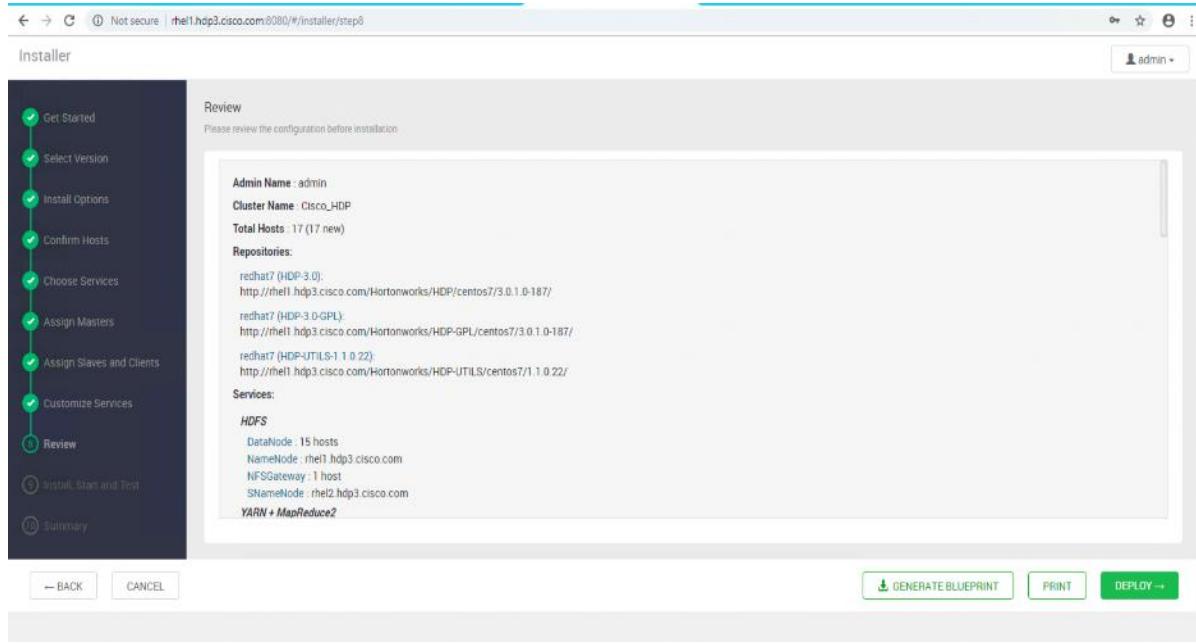
Livy2 Log directory	<input type="text" value="/data/disk1/log/livy2"/>
Livy2 PID directory	<input type="text" value="/var/run/livy2"/>

Advanced spark2-env

Spark Log directory	<input type="text" value="/data/disk1/log/spark2"/>
Spark PID directory	<input type="text" value="/var/run/spark2"/>

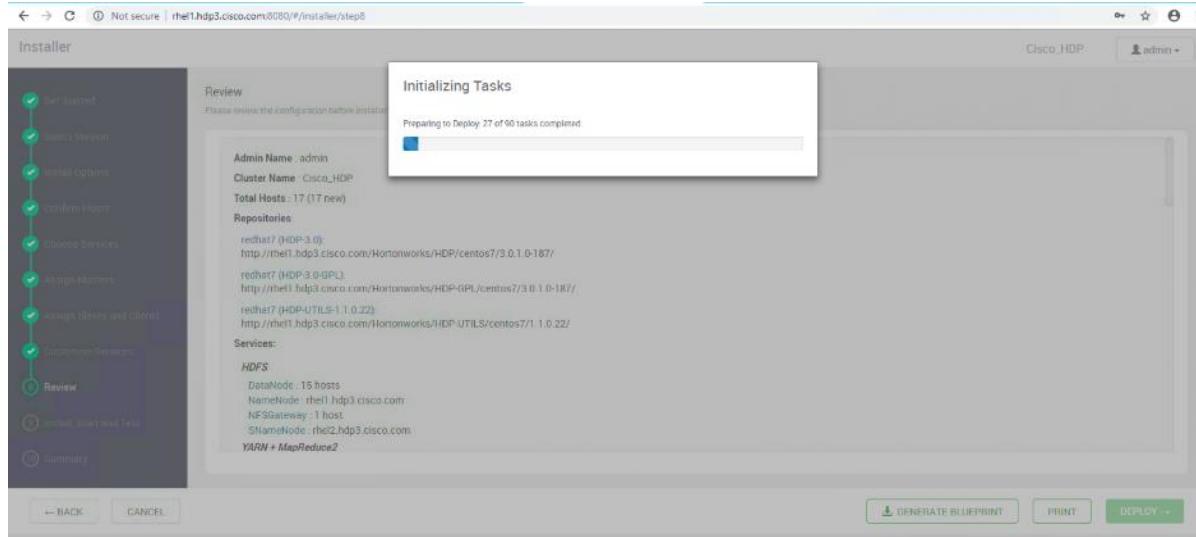
Review

The assignments that have been made are displayed. Check to make sure all is correct before clicking the Deploy button. If any changes are necessary, use the left navigation bar to return to the appropriate screen.



Deploy

- When the review is complete, click **DEPLOY**.



- Follow the onscreen installation process. Watch for warnings and failures by clicking the link as shown below:

The screenshot shows the 'Install, Start and Test' step of the HDP Installer. On the left, a vertical progress bar indicates completed steps (Get Started, Select Version, Install Options, Confirm Hosts, Choose Services, Assign Masters, Assign Slaves and Clients, Customize Services, Review) and the current step (Install, Start and Test). The main area displays a table of hosts and their installation status:

Host	Status	Message
rhel1.hdp3.cisco.com	19%	Installing Ranger Admin
rhel2.hdp3.cisco.com	33%	Install complete (Waiting to start)
rhel3.hdp3.cisco.com	33%	Install complete (Waiting to start)
rhel4.hdp3.cisco.com	33%	Install complete (Waiting to start)
rhel5.hdp3.cisco.com	33%	Install complete (Waiting to start)
rhel6.hdp3.cisco.com	33%	Install complete (Waiting to start)
rhel7.hdp3.cisco.com	33%	Install complete (Waiting to start)
rhel8.hdp3.cisco.com	33%	Install complete (Waiting to start)
rhel9.hdp3.cisco.com	33%	Install complete (Waiting to start)
rhel10.hdp3.cisco.com	33%	Install complete (Waiting to start)

A progress bar at the top right shows 32% overall completion.

Summary of the Installation Process

- On the Summary page click "COMPLETE."

The screenshot shows the Ambari Metrics dashboard. The left sidebar lists services: HDFS, YARN, MapReduce2, Tez, Hive, HBase, Pig, Sqoop, Oozie, Zookeeper, Storm, Accumulo, Infra Sols, Ambari Metrics, Atlas, Kafka, Knox, Ranger, SmartSense, Spark2, Zeppelin Note..., and Druid. The main area displays a grid of metrics:

- NameNode Heap: 6%
- HDFS Disk Usage: 0%
- NameNode CPU WIO: 0.0%
- DataNodes Live: 15/15
- NameNode RPC: 0.15 ms
- Memory Usage: 93.1 GB
- Network Usage: 19.5 KB
- CPU Usage: 100% (50%)
- Cluster Load: 40 (20)
- NameNode Uptime: 14h 45m 7s
- HBase Master Heap: 13%
- HBase Ave Load: 0.31
- Region In Transition: 0
- HBase Master Uptime: 14h 26m 43s
- ResourceManager Heap: 17%
- NodeManagers Live: 15/15

High Availability for HDFS NameNode and YARN ResourceManager

High availability for NameNode and YARN Resource Manager can be configured using Ambari or also on non-Ambari clusters. This deployment guide covers the configuration of high availability using Ambari – Use the Ambari wizard interface to configure HA of the components.

The Ambari web UI provides a wizard-driven user experience that allows to configure high availability of the components in many Hortonworks Data Platform (HDP) stack services. The high availability of the components are achieved by setting up primary and secondary components. In the event that the primary component fails or becomes unresponsive, services failover to secondary component. After configuring the high availability for a service, Ambari enables you to manage and disable (roll back) the high availability of components within that service.

Configure the HDFS NameNode High Availability

The HDFS NameNode high availability feature enables you to run redundant NameNodes in the same cluster in an Active/Passive configuration with a hot standby. This eliminates the NameNode as a potential single point of failure (SPOF) in an HDFS cluster. With the release of Hadoop 3.0, you can configure more than one backup NameNode.

Prior to the release of Hadoop 2.0, the NameNode represented a single point of failure (SPOF) in an HDFS cluster. Each cluster had a single NameNode, and if that machine or process became unavailable, the cluster as a whole would be unavailable until the NameNode was either restarted or brought up on a separate machine. This situation impacted the total availability of the HDFS cluster in two major ways:

- In the case of an unplanned event such as a machine crash, the cluster would be unavailable until an operator restarted the NameNode.
- Planned maintenance events such as software or hardware upgrades on the NameNode machine would result in periods of cluster downtime.

HDFS NameNode HA avoids this by facilitating either a fast failover to one or more backup NameNodes during machine crash, or a graceful administrator-initiated failover during planned maintenance.



Secondary NameNode is not required in high availability configuration because the Standby node also performs the tasks of the Secondary NameNode.

Active NameNode honors all the client requests and the Standby NameNode acts as a backup. The Standby NameNode keeps its state synchronized with Active NameNode through a group of JournalNodes(JNs). When the Active node performs any namespace modification, the Active node durably logs a modification record to a majority of these JNs. The Standby node reads the edits from the JNs and continuously watches the JNs for changes to the edit log.

Prerequisites for NameNode High Availability

The following are the prerequisites for NameNode high availability:

- NameNode Machine: Hardware for Active and Standby node should be exactly identical.
- JournalNodes Machine: JournalNode daemon is relatively lightweight, therefore it can be co-located on machines with other Hadoop daemons; it is typically located on the management nodes.
- There MUST be at least three JournalNodes, because the edit log modifications must be written to a majority of JNs. This allows the system tolerate failure of a single machine. You may also run more than three JournalNodes, but in order to increase the number of failures that the system can tolerate, you must run an odd number of JNs (3, 5, 7, and so on).

- ZooKeeper Machines: For automatic failover capability, an existing Zookeeper cluster must exist. The Zookeeper service can also co-exist with other Hadoop daemons.
- In HA Cluster, the Standby NameNode also performs the checkpoints of the namespace state, therefore do not deploy a Secondary NameNode, CheckpointNode, or BackupNode in an high availability cluster.

Deploy the NameNode High Availability Cluster

To deploy the NameNode high availability on a Ambari managed cluster, follow these steps:



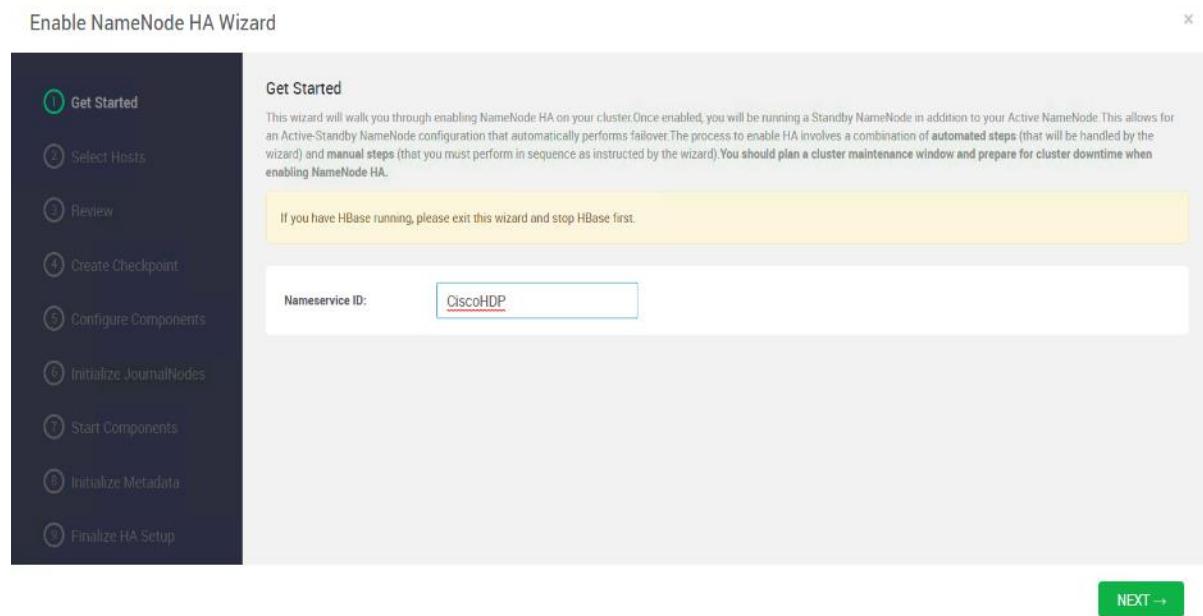
High availability cannot accept HDFS cluster names that include underscore (_).

1. Log into Ambari. Click HDFS > CONFIGS. Click the ACTIONS drop-down list and click Enable NameNode HA to launch the wizard.

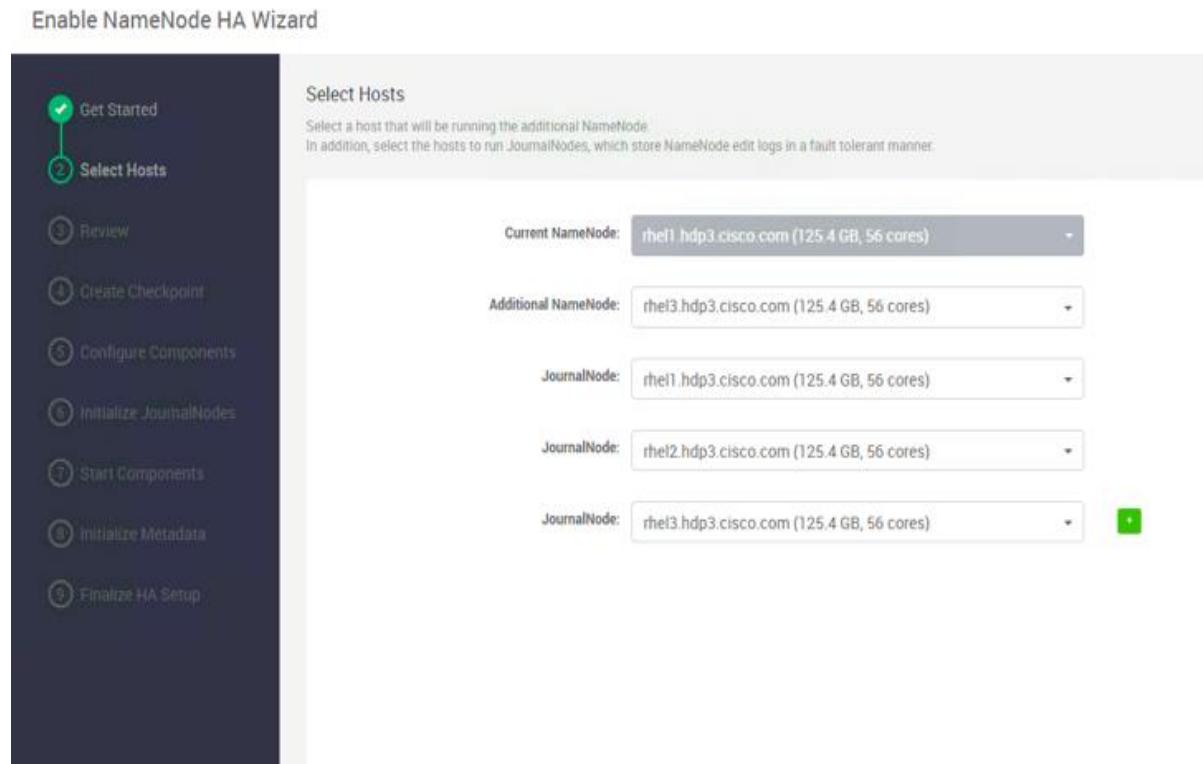
Figure 37 Enable NameNode HA

The screenshot shows the Ambari interface for managing HDFS configurations. The top navigation bar includes links for Home, Services, HDFS, and Configs, with the Configs tab currently selected. Below the navigation is a toolbar with icons for Settings, Advanced, and a search bar. A message box indicates 'No properties to display.' On the right side, there is a 'Config Group' dropdown set to 'Default (17)'. A prominent 'ACTIONS' dropdown menu is open, listing various cluster management options. The 'Enable NameNode HA' option is highlighted with a blue background and white text, indicating it is the selected action.

2. Step 1 launches the Enable NameNode HA wizard. On the Get Started page, specify the Nameservice ID as shown below. Click Next.

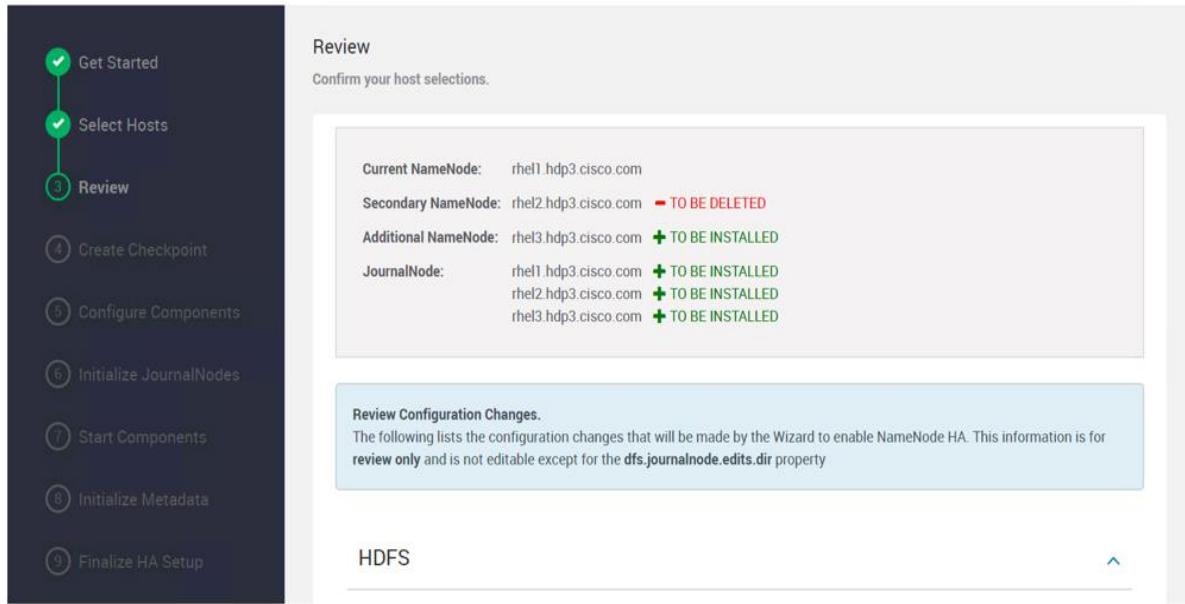
Figure 38 Enable NameNode HA Wizard – Get Started

3. On the Select Hosts page, select the Additional NameNode and JournalNode. Click Next.

Figure 39 Enable NameNode HA Wizard – Select Hosts

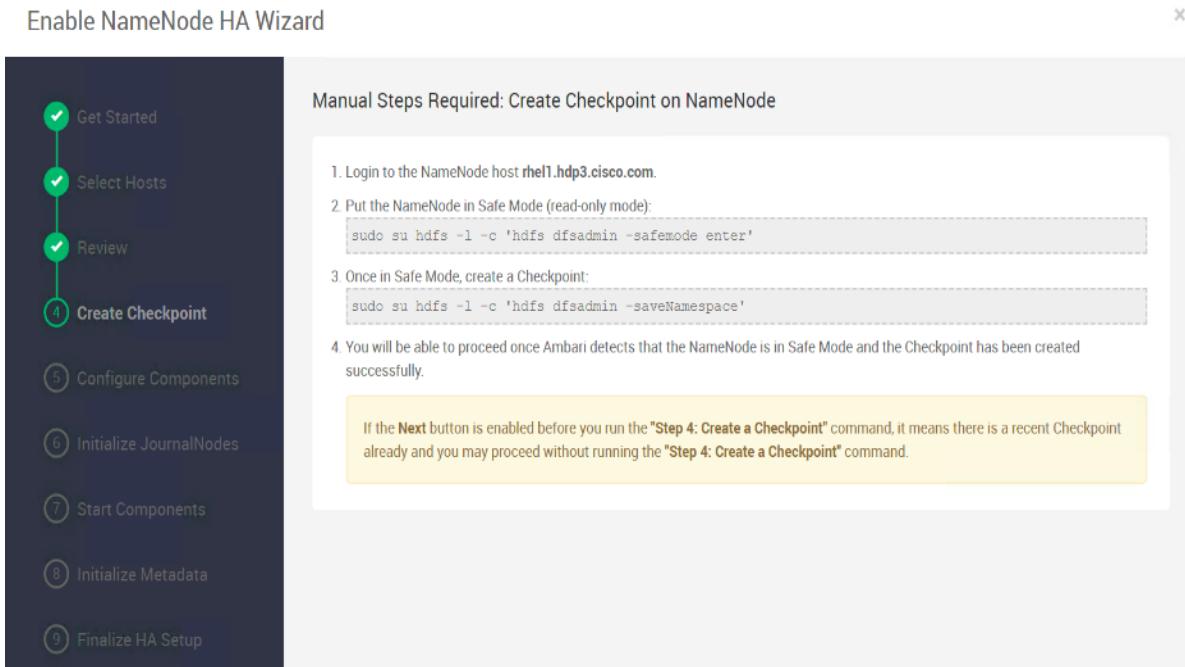
4. On the Review page, confirm the selection. To change any values, click Back, or to continue click Next.

Figure 40 Enable NameNode HA Wizard – Review



5. Create a checkpoint on the NameNode on the linux server (rhel1.hdp3.cisco.com) as shown below:

Figure 41 Enable NameNode HA Wizard – Create Checkpoint



6. SSH to current NameNode, rhel1.hdp3.cisco.com and run the following commands:

```
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs dfsadmin -safemode enter'
Safe mode is ON
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs dfsadmin -saveNamespace'
```

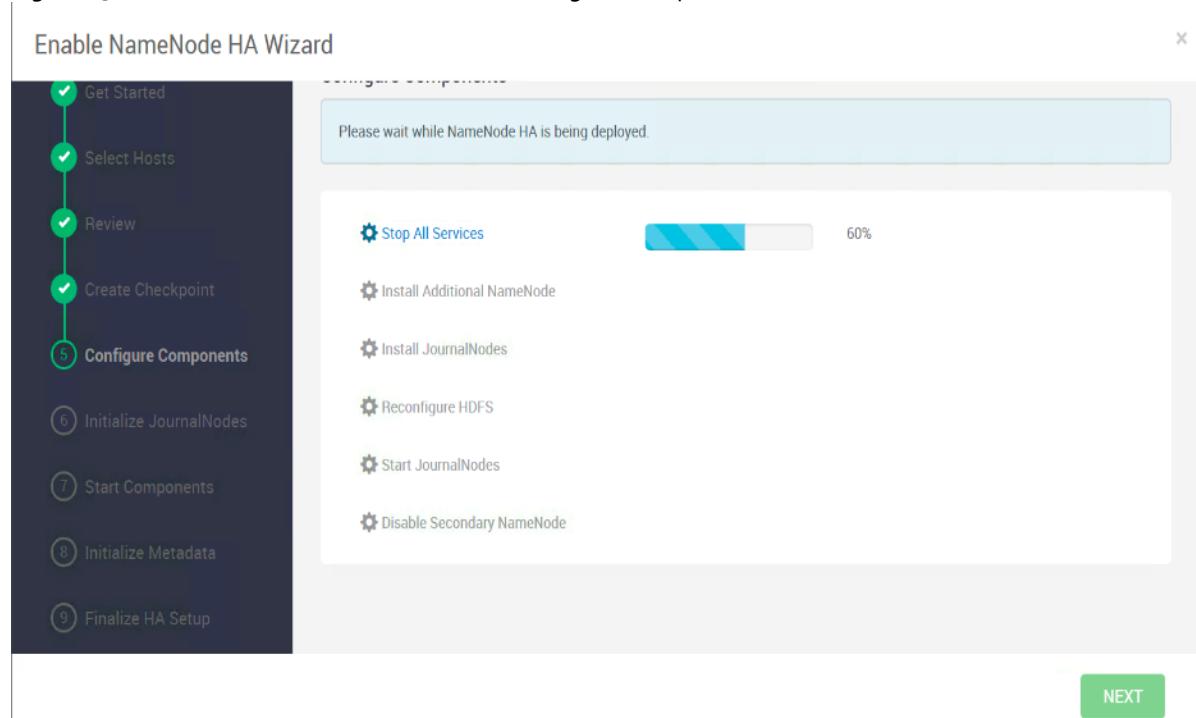
```
Save namespace successful
[root@rhel1 ~]#
```

Figure 42 Current NameNode – Safe Mode and Create Checkpoint Command

```
[root@rhel1 ~]#
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs dfsadmin -safemode enter'
Safe mode is ON
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs dfsadmin -saveNamespace'
Save namespace successful
[root@rhel1 ~]#
```

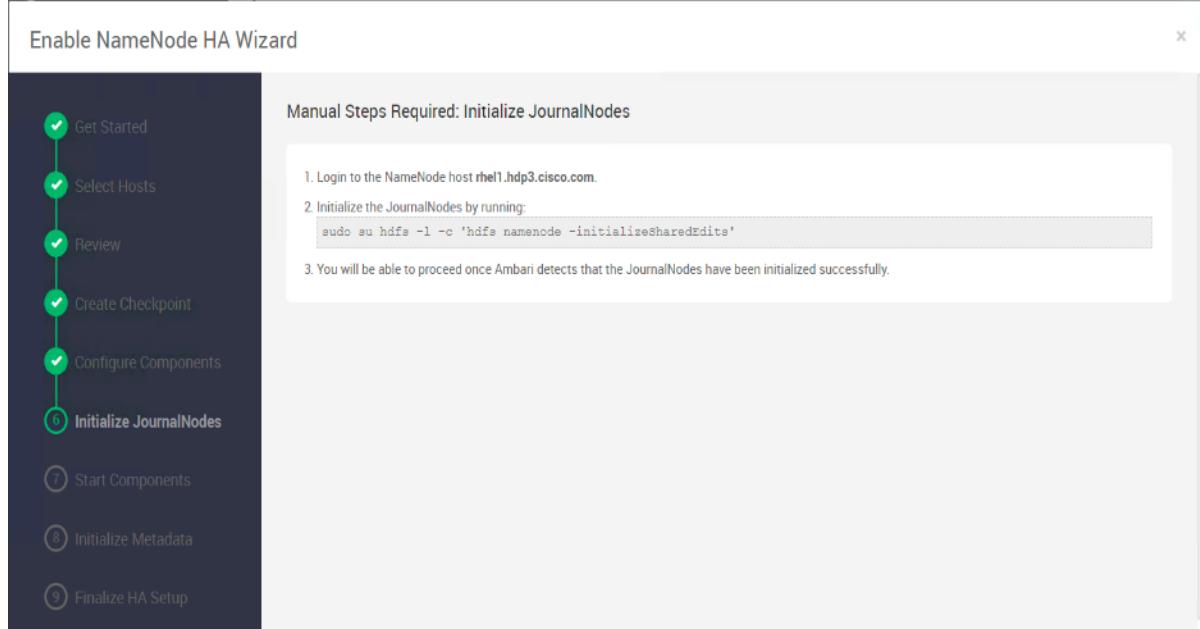
7. Return to the Ambari web UI, verify that the Checkpoint was created. Click Next.
8. See the progress bar on the Configure Components page. When the configuration steps are completed, click Next.

Figure 43 Enable NameNode HA Wizard – Configure Components



9. Initialize the JournalNodes as shown below:

Figure 44 Enable NameNode HA Wizard – Initialize JournalNodes



10. SSH to the current NameNode, for example rhel1.hdp3.cisco.com.

11. Run the following command:

```
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs namenode -initializeSharedEdits'
```

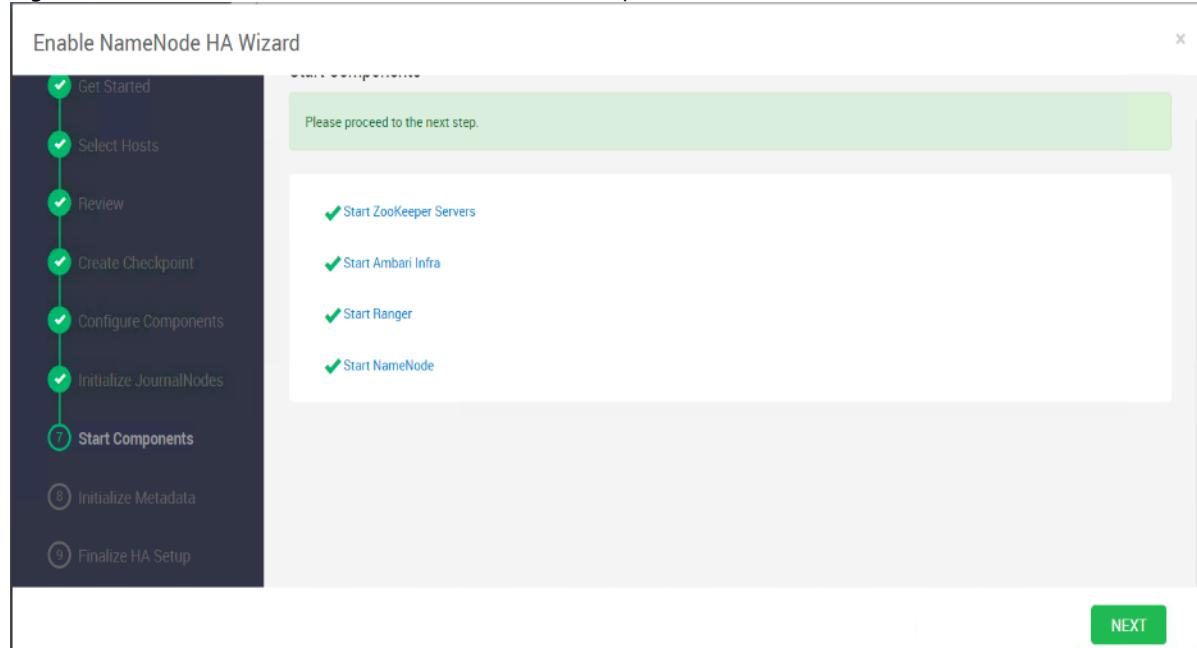
Figure 45 Initialize JournalNodes

```
[root@rhel1 ~]#
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs namenode -initializeSharedEdits'
19/01/15 12:43:27 INFO namenode.NameNode: STARTUP_MSG:
*****STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = rhel1/10.16.1.31
STARTUP_MSG: args = [-initializeSharedEdits]
STARTUP_MSG: version = 3.1.1.3.0.1.0-187
STARTUP_MSG: classpath = /usr/hdp/3.0.1.0-187/hadoop/conf:/usr/hdp/3.0.1.0-187/hadoop/lib/nimbus-jose-jwt-4.41.1.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-hdfs-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jersey-server-1.19.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-plugin-classloader-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jersey-servlet-1.19.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-yarn-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jetty-util-9.3.19.v20170502.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/accessors-smart-1.2.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/protobuf-java-2.5.0.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/asm-5.0.4.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/
```

12. Return to the Ambari UI, when Ambari detects success, click Next.

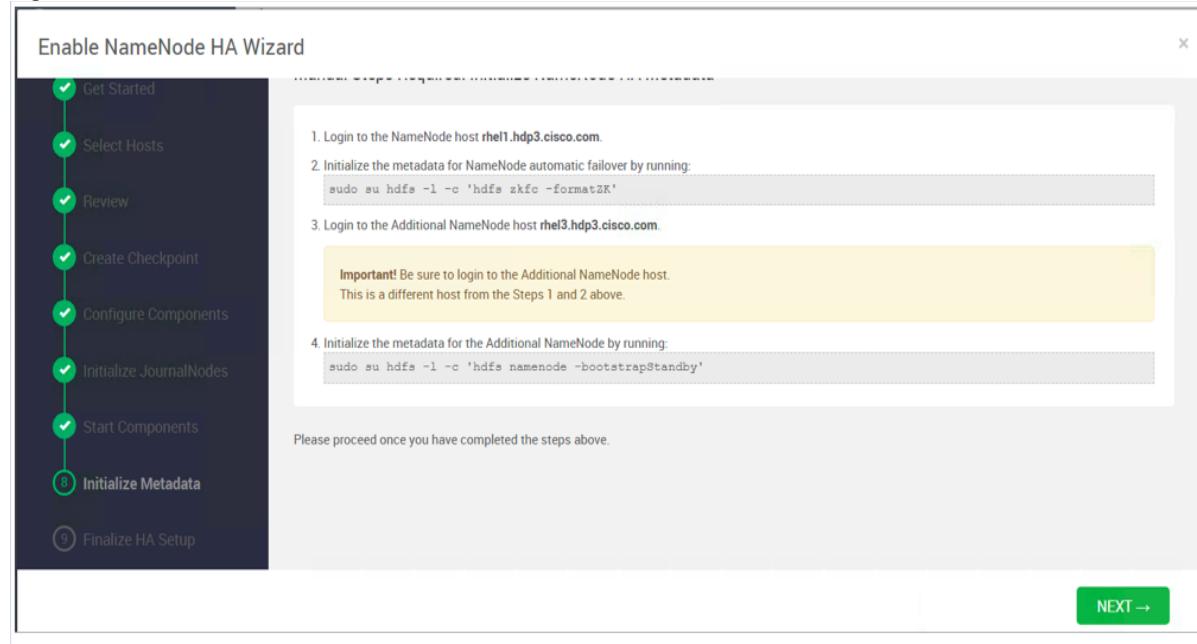
13. On the Start Components page, when completed, click Next.

Figure 46 Enable NameNode HA Wizard – Start Components



14. On the Initialize Metadata page, add the information as shown below:

Figure 47 Enable NameNode HA Wizard – Initialize Metadata



15. SSH to rhel1.hdp3.cisco.com and run the following command:

```
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs zkfc -formatZK'
```

Figure 48 Initialize the Metadata for NameNode

```
[root@rhel1 ~]#
[root@rhel1 ~]# sudo su hdfs -l -c 'hdfs zkfc -formatZK'
19/01/15 12:49:24 INFO tools.DFSZKFailoverController: STARTUP_MSG:
*****
STARTUP_MSG: Starting DFSZKFailoverController
STARTUP_MSG: host = rhel1/10.16.1.31
STARTUP_MSG: args = [-formatZK]
STARTUP_MSG: version = 3.1.1.3.0.1.0-187
STARTUP_MSG: classpath = /usr/hdp/3.0.1.0-187/hadoop/conf:/usr/hdp/3.0.1.0-187/hadoop/lib/nimbus-jose-jwt-4.41.1.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-hdfs-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jersey-server-1.19.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-plugin-classloader-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jersey-servlet-1.19.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/ranger-yarn-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/jetty-util-9.3.19.v20170502.jar:/usr/hdp/3.0.1.0-187/hadoop/lib/accessors-smart-1.2.jar:/usr/hdp/3.0.1.0-187/hadoop/l
```

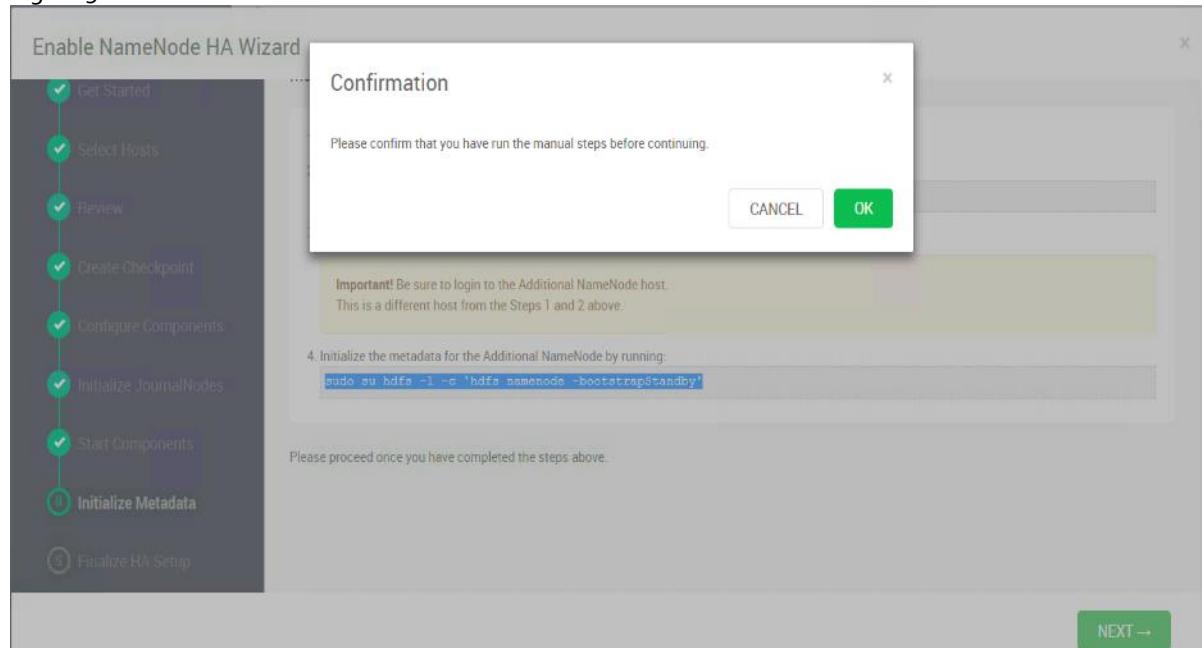
16. SSH to an additional NameNode, for example, rhel3.hdp3.cisco.com and run the following command:

```
[root@rhel3 ~]# sudo su hdfs -l -c 'hdfs namenode -bootstrapStandby'
```

Figure 49 Initialize the Metadata for Additional NameNode

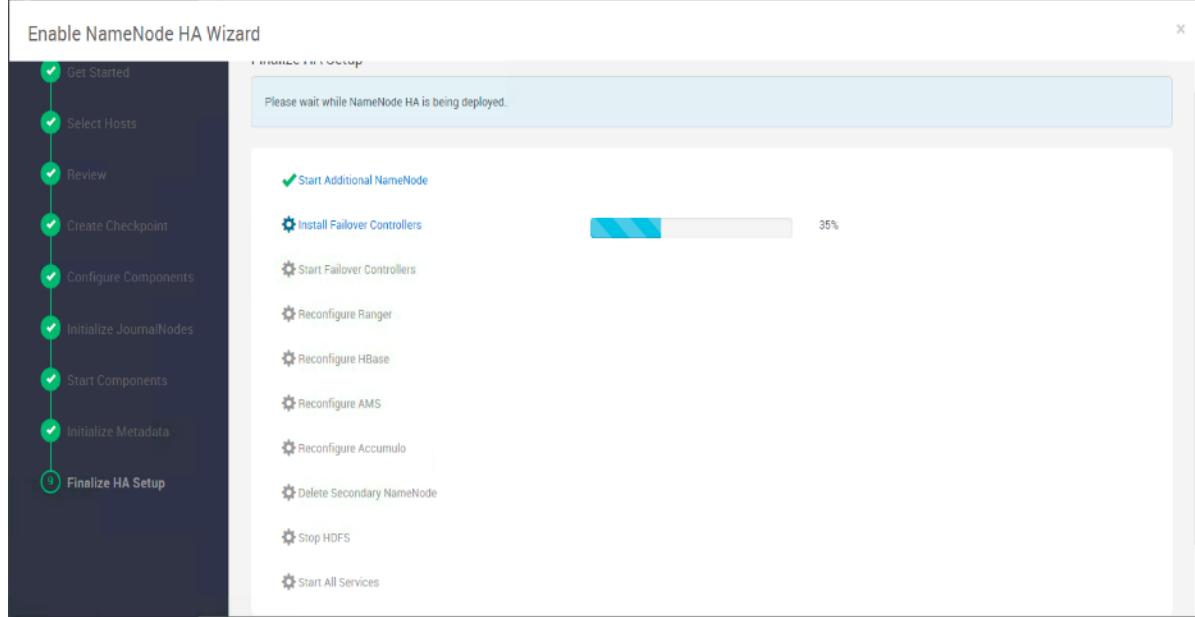
```
[root@rhel3 ~]#
[root@rhel3 ~]# sudo su hdfs -l -c 'hdfs namenode -bootstrapStandby'
19/01/17 14:34:59 INFO namenode.NameNode: STARTUP_MSG:
*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = rhel3/10.16.1.33
STARTUP_MSG: args = [-bootstrapStandby]
STARTUP_MSG: version = 3.1.1.3.0.1.0-187
STARTUP_MSG: classpath = /usr/hdp/3.0.1.0-187/hadoop/conf:/usr/hdp/3.0.1.0-187,hadoop/lib/ranger-hdfs-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187,hadoop/lib/ranger-plugin-classloader-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187,hadoop/lib/ranger-yarn-plugin-shim-1.1.0.3.0.1.0-187.jar:/usr/hdp/3.0.1.0-187,hadoop/lib/accessors-smart-1.2.jar:/usr/hdp/3.0.1.0-187,hadoop/lib/asm-5.0.4.jar:/usr/hdp/3.0.1.0-187,hadoop/lib/re2j-1.1.jar:/usr/hdp/3.0.1.0-187,hadoop/lib/jul-to-slf4j-1.7.25.jar:/usr/hdp/3.0.1.0-187,hadoop/lib/comm
```

17. Return to the Ambari web UI, click NEXT. Click OK on the confirmation pop-up window. Make sure the initialization of metadata was performed in NameNode and an additional NameNode as mentioned in step 15 and 16.

Figure 50 Enable NameNode HA Wizard – Confirmation for Initialize the Metadata

18. On the Finalize HA Setup page, you can see the progress bar as the high availability completes.

Figure 51 Enable NameNode HA Wizard – Finalize High Availability Setup



19. Click COMPLETE when done.
20. Click HDFS > SUMMARY tab, verify the Active and Standby NameNode. The Quick Links pane also shows that rhel1.hdp3.cisco.com is running the Active NameNode and rhel3.hdp3.cisco.com is running in Standby NameNode.

Figure 52 Ambari – HDFS – Summary Information

Summary		Quick Links	
Components	Started ACTIVE NAMENODE	Started ZKFAILOVERCONTROLLER	Started STANDBY NAMENODE
	2m 42s NAMENODE UPTIME	10.8% 435.0 MB / 4.0 GB NAMENODE HEAP	Started ZKFAILOVERCONTROLLER
	15/15 Started DATANODES	3/3 Live JOURNALNODES	1/1 Started NFSGATEWAYS
	DATANODES STATUS		
	15 Live	0 Dead	0 Decommissioning

Configure the YARN ResourceManager HA

This section provides instructions on setting up the ResourceManager (RM) HA feature in a HDFS cluster. The Active and Standby ResourceManagers embed the ZooKeeper based ActiveStandbyElector to determine which RM should be active.

Prerequisites for ResourceManager HA

The following are the prerequisites for ResourceManager HA:

- The servers where Active and Standby RMs are run should have identical hardware.
- For automated failover configurations, there must be an existing Zookeeper cluster available. The ZooKeeper service nodes can be co-located with the other Hadoop daemons.



At least three ZooKeeper servers must be running.

Deploy the ResourceManager HA

ResourceManager HA can be configured manually or through Ambari. These instructions are based on configuring ResourceManager HA using Ambari.

To setup ResourceManager HA, follow these steps:

1. From the Ambari web UI, click Services > YARN. Click the ACTIONS drop-down list and select Enable ResourceManager HA.

Figure 53 Services/YARN - Enable ResourceManager HA

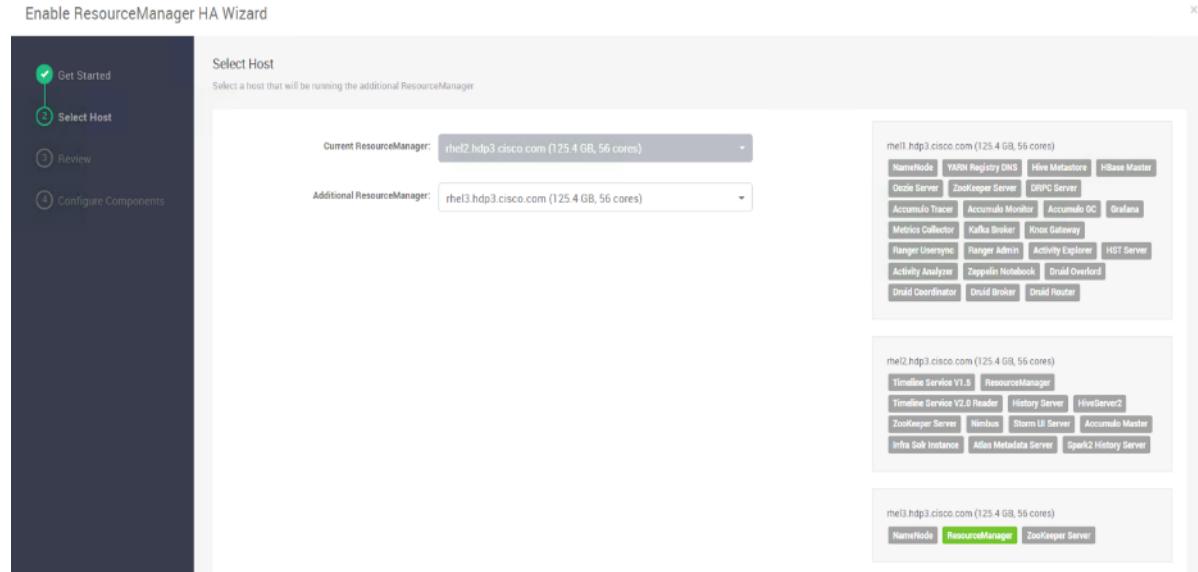
The screenshot shows the Ambari web interface for the YARN service. The top navigation bar includes links for Home, Services, YARN, Summary, and Admin. The main content area displays a summary of components: Timeline Service V1.5 (Started), ResourceManager (Started), Timeline Service V2.0 Reader (Started), and YARN Registry DNS (Stopped). Below this, it shows 16/16 NodeManagers started and 18 YARN Clients installed. The ResourceManager Uptime is listed as 6m 16s. On the right, the 'ACTIONS' dropdown menu is open, showing various options like Start, Stop, Refresh YARN Capacity Scheduler, and Enable ResourceManager HA. The 'Enable ResourceManager HA' option is highlighted with a blue background.

2. This launches the ResourceManager HA wizard as shown below. Click NEXT.

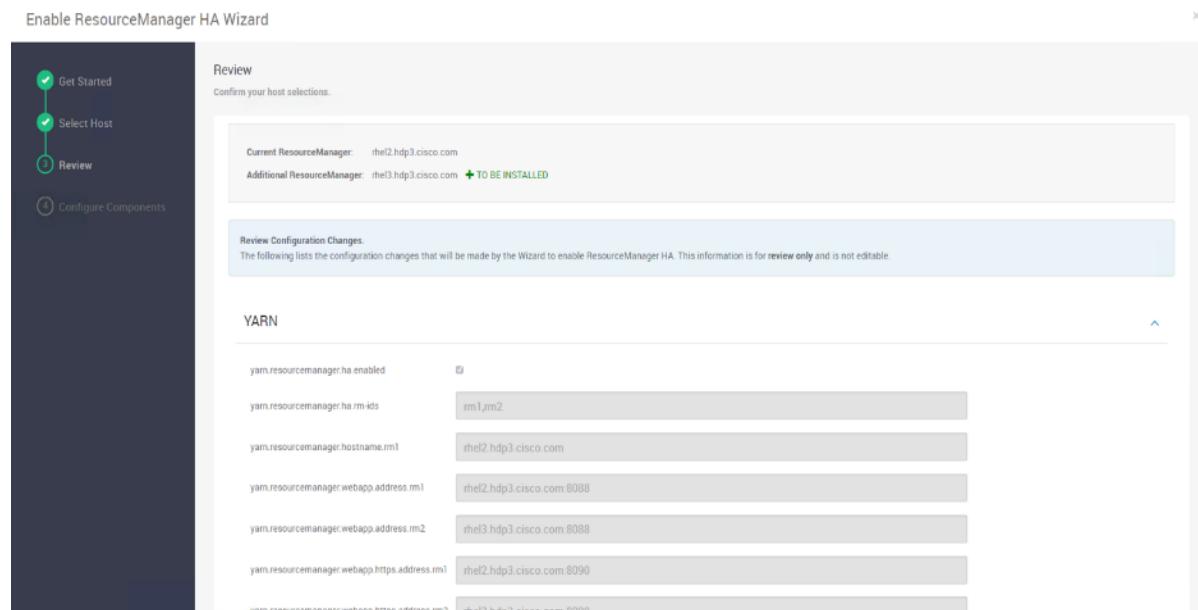
Figure 54 Enable ResourceManager HA Wizard – Get Started

The screenshot shows the 'Enable ResourceManager HA Wizard' window. On the left, a vertical navigation bar lists four steps: 1. Get Started (current), 2. Select Host, 3. Review, and 4. Configure Components. The main content area is titled 'Get Started' and contains the following text: 'This wizard will walk you through enabling ResourceManager HA on your cluster. Once enabled, you will be running a Standby ResourceManager in addition to your Active ResourceManager. This allows for an Active-Standby ResourceManager configuration that automatically performs failover.' At the bottom right is a green 'NEXT >' button.

3. From the Select Host page, select the host for Additional Resource Manager. Click NEXT.

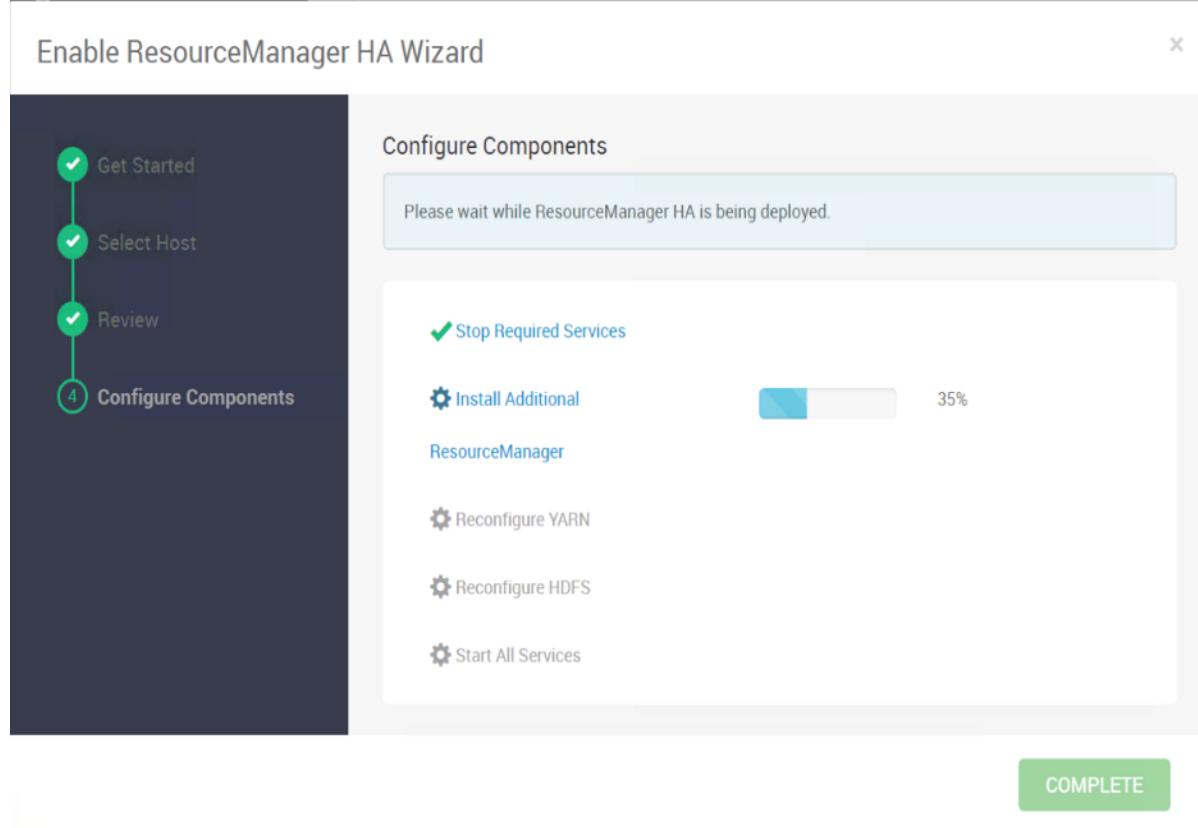
Figure 55 Enable ResourceManager HA Wizard – Select Host

4. Proceed to the Review page.

Figure 56 Enable ResourceManager HA Wizard - Review

5. The Configure Components page shows the progress bar as the Additional ResourceManager is being deployed.

Figure 57 Enable ResourceManager HA Wizard – Configure Components



6. Click COMPLETE when done.



It was observed that in certain circumstances, services might fail to restart. Click COMPLETE and restart the services in Ambari dashboard.

7. Verify the ResourceManager HA setup by clicking Services > YARN > SUMMARY tab. The Quick Links pane identifies Active and Standby ResourceManager.

Figure 58 Services/YARN/Summary – Verify ResourceManger HA

The screenshot shows the Hadoop YARN Summary page. At the top, there are navigation links: Home / Services / YARN / Summary. On the right, there are icons for system status, notifications (with 1 alert), and user admin. The cluster name is Cisco_HDP.

The main area has tabs: SUMMARY (selected), HEATMAPS, CONFIGS, METRICS, and ACTIONS.

Summary section:

- Components**:
 - Timeline Service v1.5: Started
 - Active ResourceManager: Started
 - Standby ResourceManager: Started
 - Timeline Service v2.0 Reader: Started (with a red exclamation mark)
 - YARN Registry DNS: Started
- Nodes**:
 - 16/16 Started NodeManagers
 - 18 Installed YARN Clients
- Uptime**: 13m
- NodeManager Status**:

Status	Count
Active	16
Lost	0
Unhealthy	0
Rebooted	0

Quick Links sidebar:

- rhel2.hdp3.cisco.com (Active)
 - ResourceManager UI
 - ResourceManager logs
 - ResourceManager JMX
 - Thread Stacks
- rhel3.hdp3.cisco.com (Standby)
 - ResourceManager UI
 - ResourceManager logs
 - ResourceManager JMX
 - Thread Stacks

Bill of Materials

This section provides the BOM for the 16 Nodes Hadoop Base Rack, 6 Nodes Data Science (AI/ML) Rack and 8 Nodes Hadoop tiered storage. See Table 13 for BOM for the Hadoop Base rack, Table 14 for BOM for Data Science(AI/ML) Expansion Rack, Table 15 for BOM for Hadoop Tiered Storage. Table 16, Table 17, and Table 18 for software components. Table 19 lists Cloudera SKUs available from Cisco.



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 13 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	16
CON-OSP-C240M5A2	SNTC 24X7X4OS UCS C240 M5 A2	16
UCS-CPU-I6230	2.1 GHz 6230/125W 20C/27.5MB Cache/DDR4 2933MHz	32
UCS-MR-X32G2RT-H	32GB DDR4-2933-MHz RDIMM/2Rx4/1.2V	192
UCSC-PCI-1-C240M5	Riser 1 including 3 PCIe slots (x8, x16, x8); slot 3 required CPU2	16
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	16
UCSC-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	32
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	32
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220 & C240 M4 & M5 rack servers	16
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	16
UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs & below	32
UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	416
UCSC-PCIF-240M5	C240 M5 PCIe Riser Blanking Panel	16
CBL-SC-MR12GM5P	Super Cap cable for UCSC-RAID-M5HD	16
UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	16
UCSC-RAID-M5HD	Cisco 12G Modular RAID controller with 4GB cache	16
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

Part Number	Description	Qty
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	32
UCS-HD24TB10K4KN	2.4 TB 12G SAS 10K RPM SFF HDD (4K)	312
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	15

Table 14 Bill of Materials for Hadoop Nodes Expansion Rack

Part Number	Description	Qty
UCS-SP-C240M5-A2	SP C240 M5SX w/2x6132,6x32GB mem,VIC1387	6
CON-OSP-C240M5A2	SNTC 24X7X4OS UCS C240 M5 A2	6
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 15 Bill of Materials for Hadoop Nodes Tiered Storage Rack

Part Number	Description	Qty
UCSS-S3260	Cisco UCS S3260 Storage Server Base Chassis	4
CON-OSP-UCSS3260	SNTC 24X7X4OS, Cisco UCS S3260 Storage Server Base Chassis	4
UCSC-PSU1-1050W	Cisco UCS 1050W AC Power Supply for Rack Server	16
CAB-N5K6A-NA	Power Cord, 200/240V 6A North America	16
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	4
UCSC-C3X60-RAIL	UCS C3X60 Rack Rails Kit	4
UCSS-S3260-BBEZEL	Cisco UCS S3260 Bezel	4
N20-BBLKD-7MM	UCS 7MM SSD Blank Filler	8
N20-BKVM	KVM local IO cable for UCS servers console port	8
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	4
UCSC-C3260-SIOC	Cisco UCS C3260 System IO Controller with VIC 1300 incl.	4
UCS-S3260-M5SRB	UCS S3260 M5 Server Node for Intel Scalable CPUs	4
UCS-S3260-DRAID	UCS S3260 Dual Raid based on LSI 3316	4
UCS-S3260-M5HS	UCS S3260 M5 Server Node HeatSink	8
UCS-S3260-M5SRB	UCS S3260 M5 Server Node for Intel Scalable CPUs	4
UCS-S3260-DRAID	UCS S3260 Dual Raid based on LSI 3316	4
UCS-S3260-M5HS	UCS S3260 M5 Server Node HeatSink	8
UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	48
UCS-MR-X32G2RS-H	32GB DDR4-2666-MHz RDIMM/PC4-21300/dual rank/x4/1.2v	48
UCS-C3K-HD4TB	UCS C3000 4TB NL-SAS 7200 RPM 12Gb HDD w Carrier- Top Load	224
UCS-S3260-G3SD48	UCS S3260 480G Boot SSD (Micron 6G SATA)	16
UCS-S3260-56HD4	Cisco UCS C3X60 Four row of drives containing 56 x 4TB	4
UCS-CPU-I5220	Intel 5220 2.2GHz/125W 18C/24.75MB DCP DDR4 2666 MHz	8
UCS-CPU-I5220	Intel 5220 2.2GHz/125W 18C/24.75MB DCP DDR4 2666 MHz	8
RACK2-UCS2	Cisco R42612 standard rack, w/side panels	1

Part Number	Description	Qty
CON-SNT-RCK2UCS2	SNTC 8X5XNBD, Cisco R42612 standard rack, w side panels	1

Table 16 Bill of Materials for CDSW Nodes Rack

Part Number	Description	Qty
UCS-SP-C240M5-A2	SP C240 M5SX w/2x6132,6x32GB mem, VIC1387	6
CON-OSP-C240M5A2	SNTC 24X7X4OS UCS C240 M5 A2	6
UCS-CPU-I6230	2.1 GHz 6230/125W 20C/27.5MB Cache/DDR4 2933MHz	12
UCS-MR-X32G2RT-H	32GB DDR4-2933-MHz RDIMM/2Rx4/1.2V	72
UCSC-PCI-1-C240M5	Riser 1 including 3 PCIe slots (x8, x16, x8); slot 3 required CPU2	6
UCSC-MLOM-C40Q-03	Cisco VIC 1387 Dual Port 40Gb QSFP CNA MLOM	6
UCSC-PSU1-1600W	Cisco UCS 1600W AC Power Supply for Rack Server	12
CAB-9K12A-NA	Power Cord, 125VAC 13A NEMA 5-15 Plug, North America	12
UCSC-RAILB-M4	Ball Bearing Rail Kit for C220, C240 M4 and M5 rack servers	6
CIMC-LATEST	IMC SW (Recommended) latest release for C-Series Servers.	6
UCSC-HS-C240M5	Heat sink for UCS C240 M5 rack servers 150W CPUs & below	12
UCSC-BBLKD-S2	UCS C-Series M5 SFF drive blanking panel	156
CBL-SC-MR12GM5P	Super Cap cable for UCSC-RAID-M5HD	6
UCSC-SCAP-M5	Super Cap for UCSC-RAID-M5, UCSC-MRAID1GB-KIT	6
UCSC-RAID-M5HD	Cisco 12G Modular RAID controller with 4GB cache	6
UCSC-PCI-2A-C240M5	Riser 2A including 3 PCIe slots (x8, x16, x16) supports GPU	6
UCS-SP-SD-1P6TB-4X	UCS SP 1.6TB 2.5inch Ent. Perf 12G SAS SSD (10Xendurance)4Pk	6
UCS-SP-SD-1P6TB	1.6TB 2.5inch Ent. Performance 12G SAS SSD (10X endurance)	24
UCSC-GPU-T4-16=	NVIDIA Tesla T4 16GB	12
CON-SNT-CGPUT416	SNTC-24X7X4OS NVIDIA T4 PCIE 75W 16GB	12

Table 17 Red Hat Enterprise Linux License

Part Number	Description	Qty
RHEL-2S2V-3A	Red Hat Enterprise Linux	30
CON-ISV1-EL2S2V3A	3 year Support for Red Hat Enterprise Linux	30

Table 18 Cloudera Data Science Workbench 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 19 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
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

Appendix – A

Configure Data Drives on Name Node and Other Management Nodes

This section describes the steps needed to configure non-OS disk drives as RAID1 using the StorCli command. All drives are 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.

To configure data drives on Name node and other nodes, If the drive state shows up as JBOD, creating RAID in the subsequent steps will fail with the error “*The specified physical disk does not have the appropriate attributes to complete the requested command.*”

To configure data drives on Name Node and others, follow these steps:

1. If the drive state shows up as JBOD, it can be converted into Unconfigured Good using Cisco UCSM or storcli64 command. Following steps should be performed if the state is JBOD.
2. Get the enclosure id as follows:

```
ansible all -m shell -a "./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'"

[root@rhel01 ~]# ansible all -m shell -a "./storcli64 pdlist -a0 | grep Enc | grep -v 252 | awk '{print $4}' | sort | uniq -c | awk '{print $2}'"
rhel06.hdp3.cisco.local | CHANGED | rc=0 >>
    24 Enclosure Device ID: 66
    24 Enclosure position: 0

rhel04.hdp3.cisco.local | CHANGED | rc=0 >>
    24 Enclosure Device ID: 66
    24 Enclosure position: 0

rhel08.hdp3.cisco.local | CHANGED | rc=0 >>
    24 Enclosure Device ID: 66
    24 Enclosure position: 0
```



It is observed that some earlier versions of storcli64 complains about above mentioned command as if it is deprecated. In this case, please use “./storcli64 /c0 show all| awk '{print \$1}'| sed -n '/[0-9]:[0-9]/p'|awk '{print substr(\$1,1,2)}'|sort -u” command to determine enclosure id.



In case of S3260 use -ao and -a1 or c0 and c1 as there are two controller per node.

3. Convert to unconfigured good:

```
ansible datanodes -m command -a "./storcli64 /c0 /e66 /sall set good force"
```

4. Verify status by running the following command:

```
# ansible datanodes -m command -a "./storcli64 /c0 /e66 /sall show"
```

5. Run this script as root user on rhel01 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] wb ra nocachedbadbbu strpsz1024 -a0
```



The script (above) requires enclosure ID as a parameter.

- Run the following command to get enclosure id:

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

- Run MegaCli script:

```
#!/usr/bin/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
```



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

- Run the following command. State should change to Online:

```
ansible namenodes -m command -a "./storcli64 /c0 /e66 /sall show"
```

- State can also be verified in UCSM as shown below in Equipment>Rack-Mounts>Servers>Server # under Inventory/Storage/Disk tab:

	Size (MB)	Serial	Operability	Drive Status	Processor	Technology	Bios/BIOS
Disk 1	17'5655	S320217142000055484L1	Operable	Online	Up-to-date	HDD	False
Disk 2	17'5655	S320171420000055484L5	Operable	Online	Up-to-date	HDD	False
Disk 3	17'5655	S320171420000055125L8	Operable	Online	Up-to-date	HDD	False
Disk 4	17'5655	S320171420000055131L2	Operable	Online	Up-to-date	HDD	False
Disk 5	17'5655	S320171420000055147L9	Operable	Online	Up-to-date	HDD	False
Disk 6	17'5655	S320171420000055159LJ	Operable	Online	Up-to-date	HDD	False
Disk 7	17'5655	S320171420000055157L1	Operable	Online	Up-to-date	HDD	False

Configure Data Drives on Data Nodes

To configure non-OS disk drives as individual RAID0 volumes using StorCli command, follow these steps. These volumes will be used for HDFS Data.

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

```
[root@rhel01 ~]# ansible datanodes -m command -a "./storcli64 -cfgeachdiskraido WB RA direct
NoCachedBadBBU strpsz1024 -a0"

rhel7.hdp3.cisco.local | SUCCESS | rc=0 >>
Adapter 0: Created VD 0
```

```

Configured physical device at Encl-66:Slot-7.
Adapter 0: Created VD 1
Configured physical device at Encl-66:Slot-6.
Adapter 0: Created VD 2
Configured physical device at Encl-66:Slot-8.
Adapter 0: Created VD 3
Configured physical device at Encl-66:Slot-5.
Adapter 0: Created VD 4
Configured physical device at Encl-66:Slot-3.
Adapter 0: Created VD 5
Configured physical device at Encl-66:Slot-4.
Adapter 0: Created VD 6
Configured physical device at Encl-66:Slot-1.
Adapter 0: Created VD 7
Configured physical device at Encl-66:Slot-2.
..... Omitted Output
24 physical devices are Configured on adapter 0.

Exit Code: 0x00

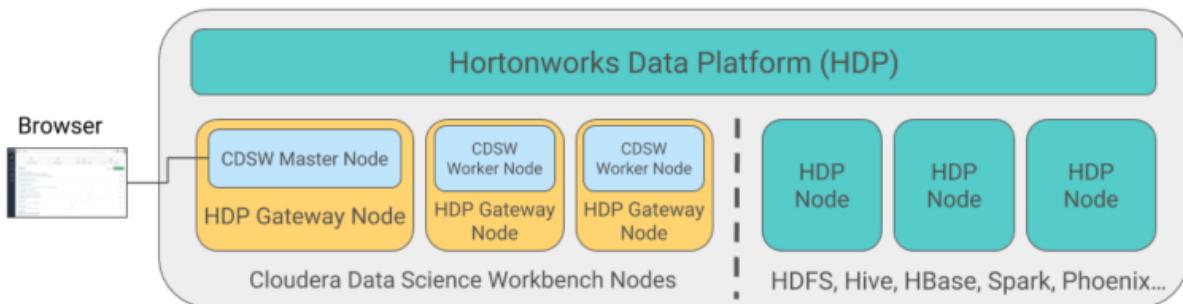
```



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

Cloudera Data Science Workbench (CDSW)

Cloudera Data Science Workbench runs on one or more dedicated gateway / edge hosts on HDP clusters. A gateway host is one that does not have any cluster services running on them. They only run the clients for cluster services (such as the HDFS Client, YARN Client, Spark2 Client, and so on). These clients ensure that Cloudera Data Science Workbench has all the libraries and configuration files necessary to securely access the HDP cluster and their respective services.



Cloudera Data Science Workbench does not support running any other services on these gateway hosts. Each gateway host must be dedicated solely to Cloudera Data Science Workbench. This is because user workloads require dedicated CPU and memory, which might conflict with other services running on these hosts.

From the gateway hosts assigned to Cloudera Data Science Workbench, one will serve as the master host, which also runs the CDSW web application, while others will serve as worker hosts. You should note that worker hosts are not required for a fully-functional Cloudera Data Science Workbench deployment. For proof-of-concept deployments you can deploy a 1-host cluster with just a Master host. The Master host can run user workloads just as a worker can.

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

- CUDA pre-requisites
- CUDA

- CDSW pre-requisites
- CDSW



This CVD incorporates 6x Cisco UCS C240 M5 with 2x T4 GPUs; the following steps in this section detail how to enable GPU as a Hadoop resource.



For supported platforms and requirements, go to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_hdp.html

Figure 59 Cisco UCS C240 M5 resource inventory as seen through Cisco UCS Manager

Equipment / Rack-Mounts / Servers / Server 23 (GPU-Group-6)

General	Inventory	Virtual Machines	Hybrid Display	Installed Firmware	SEL Logs	CIMC Sessions	VIF Paths	Power Control Monitor	Health	Diagnostics	Faults	Event >																																																				
Motherboard	CIMC	CPUs	Coprocessor Cards	GPUs	PCI Switch	Memory	Adapters	HBAs	NICs	iSCSI vNICs	Storage	Persistent Memory																																																				
<table border="1"> <thead> <tr> <th colspan="2">Advanced Filter</th> <th colspan="2">Export</th> <th colspan="2">Print</th> <th colspan="7"></th> </tr> <tr> <th>Name</th> <th>ID</th> <th>Model</th> <th>Serial</th> <th colspan="9">Mode</th> </tr> </thead> <tbody> <tr> <td>Graphics Card 1</td> <td>1</td> <td>nVidia T4 PG183-200</td> <td>1561819010997</td> <td colspan="9">Compute</td> </tr> <tr> <td>Graphics Card 2</td> <td>2</td> <td>nVidia T4 PG183-200</td> <td>1561819011347</td> <td colspan="9">Compute</td> </tr> </tbody> </table>													Advanced Filter		Export		Print									Name	ID	Model	Serial	Mode									Graphics Card 1	1	nVidia T4 PG183-200	1561819010997	Compute									Graphics Card 2	2	nVidia T4 PG183-200	1561819011347	Compute								
Advanced Filter		Export		Print																																																												
Name	ID	Model	Serial	Mode																																																												
Graphics Card 1	1	nVidia T4 PG183-200	1561819010997	Compute																																																												
Graphics Card 2	2	nVidia T4 PG183-200	1561819011347	Compute																																																												

Details	
ID : 1	PCI Slot : 2
Expander Slot ID : NA	PID : UCSC-GPU-T4-16GB-V1
Is Supported : Yes	Vendor : nVidia
Model : nVidia T4 PG183-200	Serial : 1561819010997
Running Version : 90.04.38.00.03 G183.0200.00.02	Activate Status : Ready
Mode : Compute	Temperature : 24

Install the Prerequisites for CUDA

To install the prerequisites for CUDA, follow these steps:



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



These commands are run as root or sudo.

1. List GPUs and CPUs installed:

```
[root@rhel1 ~]# ansible gpunodes -m shell -a "lspci | grep -i nvidia"
```

```
[root@rhel1 ~]# ansible gpunodes -m shell -a "lspci | grep -i nvidia"
rhel22.hdp3.cisco.com | SUCCESS | rc=0 >>
5e:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)
af:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)

rhel23.hdp3.cisco.com | SUCCESS | rc=0 >>
5e:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)
af:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)

rhel18.hdp3.cisco.com | SUCCESS | rc=0 >>
5e:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)
af:00.0 3D controller: NVIDIA Corporation Device 1eb8 (rev a1)
```

```
[root@rhel1 ~]# ansible gpunodes -m shell -a "lscpu"
```

```
[root@rhel1 ~]# ansible gpunodes -m shell -a "lscpu"
rhel23.hdp3.cisco.com | SUCCESS | rc=0 >>
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: 2600.000
CPU max MHz: 2600.0000
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
```

Install GCC

To install GCC, follow these steps:

1. Make sure gcc is installed in the system:

```
[root@rhel1 ~]# ansible gpunodes -m shell -a "gcc --version"
```

```
[root@rhel1 ~]# ansible gpunodes -m shell -a "gcc --version"
rhel22.hdp3.cisco.com | SUCCESS | rc=0 >>
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

rhel23.hdp3.cisco.com | SUCCESS | rc=0 >>
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

rhel18.hdp3.cisco.com | SUCCESS | rc=0 >>
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Install Kernel Headers and Installation Packages

The CUDA Driver requires that the kernel headers and development packages for the running version of the kernel are installed at the time of the driver installation, as well as whenever the driver is rebuilt. For example, if your system is running kernel version 3.17.4-301, the 3.17.4-301 kernel headers and development packages must also be installed.

```
ansible gpunodes -m shell -a "uname -r"
```

```
[root@rhel1 ~]# ansible gpunodes -m shell -a "uname -r"
rhel22.hdp3.cisco.com | SUCCESS | rc=0 >>
3.10.0-957.el7.x86_64

rhel23.hdp3.cisco.com | SUCCESS | rc=0 >>
3.10.0-957.el7.x86_64

rhel18.hdp3.cisco.com | SUCCESS | rc=0 >>
3.10.0-957.el7.x86_64
```

```
[root@rhel1 ~]# ansible gpunodes -m yum -a "name=kernel-devel-$(uname -r) state=present"
```

```
[root@rhel1 ~]# ansible gpunodes -m yum -a "name=kernel-devel-$(uname -r) state=present"
rhel23.hdp3.cisco.com | SUCCESS => {
    "changed": false,
    "failed": false,
    "msg": "",
    "rc": 0,
    "results": [
        "kernel-devel-3.10.0-957.el7.x86_64 providing kernel-devel-3.10.0-957.el7.x86_64 is already installed"
    ]
}
rhel18.hdp3.cisco.com | SUCCESS => {
    "changed": false,
    "failed": false,
    "msg": "",
    "rc": 0,
    "results": [
        "kernel-devel-3.10.0-957.el7.x86_64 providing kernel-devel-3.10.0-957.el7.x86_64 is already installed"
    ]
}
rhel22.hdp3.cisco.com | SUCCESS => {
    "changed": false,
    "failed": false,
    "msg": "",
    "rc": 0,
    "results": [
        "kernel-devel-3.10.0-957.el7.x86_64 providing kernel-devel-3.10.0-957.el7.x86_64 is already installed"
    ]
}
```

```
[root@rhel1 ~]# ansible gpunodes -m yum -a "name=kernel-headers-$(uname -r) state=present"
```

```
[root@rhel1 ~]# ansible gpunodes -m yum -a "name=kernel-headers-$(uname -r) state=present"
rhel23.hdp3.cisco.com | SUCCESS => {
    "changed": false,
    "failed": false,
    "msg": "",
    "rc": 0,
    "results": [
        "kernel-headers-3.10.0-957.el7.x86_64 providing kernel-headers-3.10.0-957.el7.x86_64 is already installed"
    ]
}
rhel18.hdp3.cisco.com | SUCCESS => {
    "changed": false,
    "failed": false,
    "msg": "",
    "rc": 0,
    "results": [
        "kernel-headers-3.10.0-957.el7.x86_64 providing kernel-headers-3.10.0-957.el7.x86_64 is already installed"
    ]
}
rhel22.hdp3.cisco.com | SUCCESS => {
    "changed": false,
    "failed": false,
    "msg": "",
    "rc": 0,
    "results": [
        "kernel-headers-3.10.0-957.el7.x86_64 providing kernel-headers-3.10.0-957.el7.x86_64 is already installed"
    ]
}
```

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](#).

To install DKMS, follow these steps:

<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-2.7.1-1.el7.noarch.rpm

```
# wget http://rpmfind.net/linux/epel/7/x86_64/Packages/d/dkms-2.7.1-1.el7.noarch.rpm
```

1. Copy dkms rpm to all the GPU servers:

```
[root@rhel1 ~]# ansible gpnodes -m copy -a "src=/root/dkms-2.7.1-1.el7.noarch.rpm dest=/root/."
```

2. Install dkms with yum install:

```
[root@rhel1 ~]# ansible gpnodes -m command -a "yum install -y /root/dkms-2.7.1-1.el7.noarch.rpm"
```

Install NVIDIA GPU Drivers

To install the NVIDIA GPU drivers, follow these steps:

1. Download this NVIDIA GPU driver from <https://www.nvidia.com/Download/index.aspx?lang=en-us>
2. For the NVIDIA driver download, select the product type, Series, Product, OS, and CUDA toolkit.



For this deployment, select 10.1 for CUDA Toolkit.

DOWNLOAD DRIVERS

NVIDIA > [Download Drivers](#)



NVIDIA Driver Downloads

Option 1: Manually find drivers for my NVIDIA products.

[Help](#)

Product Type:

Product Series:

Product:

Operating System:

CUDA Toolkit:

Language:

[SEARCH](#)

RTX. IT'S ON.

3. Click SEARCH. The selected driver is shown below:

DOWNLOAD DRIVERS

NVIDIA > DRIVERS > TESLA DRIVER FOR LINUX X64



TESLA DRIVER FOR LINUX X64

Version: 418.87
Release Date: 2019.8.14
Operating System: Linux 64-bit
CUDA Toolkit: 10.1
Language: English (US)
File Size: 103.47 MB

[DOWNLOAD](#)

- Click DOWNLOAD. The download link can be captured by right-clicking AGREE & DOWNLOAD as shown below.

DOWNLOAD DRIVERS

NVIDIA Home > Download Drivers > Download Confirmation



Download

By clicking the "Agree & Download" button below, you are confirming that you have read and agree to be bound by the [License For Customer Use of NVIDIA Software](#) for use of the driver. The driver will begin downloading immediately after clicking on the "Agree & Download" button below. NVIDIA recommends users update to the latest driver version. Please review [NVIDIA Product Security](#) for more information.

[AGREE & DOWNLOAD](#)

[DECLINE](#)

```
[root@rhel1 ~]# wget http://us.download.nvidia.com/tesla/418.67/nvidia-diag-driver-local-repo-rhel7-418.67-1.0-1.x86\_64.rpm
```

- Copy the .rpm file in all the GPU nodes as shown below.

```
[root@rhel1 ~]# ansible gpunodes -m copy -a "src=/root/nvidia-diag-driver-local-repo-rhel7-418.67-1.0-1.x86_64.rpm dest=/root/."
```

- Install the driver by running the following command:

```
[root@rhel1 ~]# ansible gpunodes -m command -a "rpm -ivh /root/nvidia-diag-driver-local-repo-rhel7-418.67-1.0-1.x86_64.rpm"
rhel16.hdp3.cisco.com | SUCCESS | rc=0 >>
Preparing... #####
Updating / installing...
nvidia-diag-driver-local-repo-rhel7-39#####warning:
/root/nvidia-diag-driver-local-repo-rhel7-418.67-1.0-1.x86_64.rpm: Header V3 RSA/SHA512
Signature, key ID 7fa2af80: NOKEY
```

Install CUDA

To install CUDA, follow these steps:

- Download CUDA 10.1.



TensorFlow needs CUDA; make sure that the version of CUDA is supported by TensorFlow before installing CUDA. Earlier versions of CUDA are here: <https://developer.nvidia.com/cuda-toolkit-archive>.

The latest version of CUDA is available here:

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

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows	Linux	Mac OS X			
Architecture	x86_64	ppc64le				
Distribution	Fedora	OpenSUSE	RHEL	CentOS	SLES	Ubuntu
Version	8	7	6			
Installer Type	runfile (local)	rpm (local)	rpm (network)	cluster (local)		

Download Installer for Linux RHEL 7 x86_64

The base installer is available for download below.

Base Installer

Installation Instructions:

```
$ wget http://developer.download.nvidia.com/compute/cuda/10.1/Prod/local_installers/cuda-repo-rhel7-10-1-local-10.1.243-418.87.00-1.0-1.x86_64.rpm
$ sudo rpm -i cuda-repo-rhel7-10-1-local-10.1.243-418.87.00-1.0-1.x86_64.rpm
$ sudo yum clean all
$ sudo yum -y install nvidia-driver-latest-dkms cuda
```

The CUDA Toolkit contains Open-Source Software. The source code can be found [here](#).
The checksums for the installer and patches can be found in [Installer Checksums](#).
For further information, see the [Installation Guide for Linux](#) and the [CUDA Quick Start Guide](#).

```
[root@rhel1 ~]# wget https://developer.nvidia.com/compute/cuda/10.1/Prod/local_installers/cuda-repo-rhel7-10-1-local-10.1.168-1.x86_64
```

2. Copy .rpm file to all GPU nodes as follows:

```
[root@rhel1 ~]# ansible gpunodes -m copy -a "src=/root/cuda-repo-rhel7-10-1-local-10.1.168-1.x86_64.rpm dest=/root/."
```

3. Install CUDA in all GPU nodes by running the following set of commands:

```
[root@rhel1 ~]# ansible gpunodes -m shell -a "rpm -i cuda-repo-rhel7-10-1-local-10.1.168-1.x86_64.rpm"
[root@rhel1 ~]# ansible gpunodes -m shell -a "yum clean all"
[root@rhel1 ~]# ansible gpunodes -m shell -a "yum -y install cuda"
```

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

Download cuDNN 10.1

To download and set up NVIDIA cuDNN, follow these steps:

1. Download cuDNN from <https://developer.nvidia.com/cudnn> for the same CUDA version.
2. Copy cuDNN into all the GPU servers.

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



This step may require joining the NVIDIA developer community.

3. Run the following Ansible commands in all GPU nodes to setup cuDNN:

```
[root@rhel1 ~]# ansible gpunodes -m copy -a "src=/root/cudnn-10.1-linux-x64-v7.1.tgz dest=/root/."
[root@rhel1 ~]# ansible gpunodes -m shell -a "tar -xzvf cudnn-10.1-linux-x64-v7.1.tgz"
[root@rhel1 ~]# ansible gpunodes -m shell -a "cp /root/cuda/include/cudnn.h /usr/local/cuda-10.1/include"
[root@rhel1 ~]# ansible gpunodes -m shell -a "cp /root/cuda/lib64/libcudnn* /usr/local/cuda-10.1/lib64"
[root@rhel1 ~]# ansible gpunodes -m shell -a "chmod a+r /usr/local/cuda-10.1/include/cudnn.h /usr/local/cuda-10.1/lib64/libcudnn*"
```

Post Installation Steps

1. Add CUDA in PATH and LD_LIBRARY_PATH variable in all the GPU nodes.
2. The PATH and LD_LIBRARY_PATH variable need to include /usr/local/<cuda>/bin:

```
export PATH=/usr/local/cuda-10.1/bin${PATH:+:$PATH}
export LD_LIBRARY_PATH=/usr/local/cuda-10.1/lib64${LD_LIBRARY_PATH:+:$LD_LIBRARY_PATH}
```

3. Reboot the GPU nodes:

```
# ansible gpunodes -a "/sbin/reboot"
```



The SSH and Ansible connection will disconnect with this command.



For more information, go to:

Verify Drivers

To verify the drivers have been installed, run the following commands in all GPU nodes as shown below:

```
[root@rhel1 ~]# ansible gpunodes -a "nvidia-smi"
Or on specific node
[root@rhel1 ~]# ansible gpunodes -a "nvidia-smi"
```

```
[root@rhell ~]# ansible gpunodes -a "nvidia-smi"
rhel22.hdp3.cisco.com | SUCCESS | rc=0 >>
Tue Aug 27 11:16:08 2019
+-----+
| NVIDIA-SMI 418.67      Driver Version: 418.67      CUDA Version: 10.1 |
+-----+
| GPU  Name      Persistence-M| Bus-Id      Disp.A  | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
+-----+
| 0  Tesla T4           off | 00000000:5E:00.0 Off |          0 |
| N/A   25C    P8    9W / 70W |     0MiB / 15079MiB |     0%       Default |
+-----+
| 1  Tesla T4           off | 00000000:AF:00.0 Off |          0 |
| N/A   24C    P8    9W / 70W |     0MiB / 15079MiB |     0%       Default |
+-----+
+-----+
| Processes:                               GPU Memory |
| GPU  PID  Type  Process name             Usage      |
+-----+
| No running processes found
+-----+
```

```
[root@rhell ~]# ansible gpunodes -m shell -a "/usr/local/cuda-10.1/bin/nvcc --version"
```

```
[root@rhell ~]# ansible gpunodes -m shell -a "/usr/local/cuda-10.1/bin/nvcc --version"
rhel23.hdp3.cisco.com | SUCCESS | rc=0 >>
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Wed_Apr_24_19:10:27_PDT_2019
Cuda compilation tools, release 10.1, V10.1.168

rhel22.hdp3.cisco.com | SUCCESS | rc=0 >>
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Wed_Apr_24_19:10:27_PDT_2019
Cuda compilation tools, release 10.1, V10.1.168

rhel18.hdp3.cisco.com | SUCCESS | rc=0 >>
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Wed_Apr_24_19:10:27_PDT_2019
Cuda compilation tools, release 10.1, V10.1.168
```

```
[root@rhell ~]# ansible gpunodes -m shell -a "export NVIDIA_DRIVER_VERSION=418.67"
```

Installation Prerequisites for CDSW

To install the prerequisites for CDSW, complete the steps in the following sections 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/1-6-x/topics/cdsw_install.html

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.

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. For more information, refer to: https://www.cloudera.com/documentation/data-science-workbench/1-6-x/topics/cdsw_install.html#wildcard_dns



For the Solution validation we had installed DNS server on windows 2012 R2 and setup DNS wild card configuration on the same.

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

1. Update /etc/dnsmasq.conf to enable Wildcard entry:

```
address=/cdsw/10.15.1.53
```

```
#service dnsmasq restart
```

2. 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, go to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_install.html#pre_install.

Supported JDK Version

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

Spark 2 which is needed by CDSW to run Spark jobs (on Hadoop nodes) requires JDK 1.8. On CSD-based deployments, the version of Java installed on Cloudera Data Science Workbench gateway hosts and also updating the JAVA home directory in the "cdsw.conf" file located in /etc/cdsw/config directory..

To update the JDK version, go to "[Changing the JDK version on an Existing HDP Cluster](#)"



For more details, go to: 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 1.5.x Requirements and Supported Platforms, go to:
https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_requirements_supported_versions.html



Please refer to section [Disable SELinux](#) if SELinux is enabled.

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 HDP 3.1.0.



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 HDP 3.1.0

To download and install CDSW, complete the following steps:

1. Log into the Ambari Server
2. Go to the Hosts page and select Actions > + Add New Hosts.
3. On the Install Options page, enter the fully-qualified domain names for your new hosts.

The wizard also needs the private key file you created when you set up password-less SSH. Using the host names and key file information, the wizard can locate, access, and interact securely with all the hosts in the cluster. Alternatively, you can [manually install and start the Ambari agents](#) on all the new hosts.

4. Click Register and Confirm. For more detailed instructions, refer to "https://docs.hortonworks.com/HDPDocuments/Ambari-2.6.2.2/bk_ambari-installation/content/install_options.html"
5. The Confirm Hosts page prompts you to confirm that Ambari has located the correct hosts for your cluster and to check those hosts to make sure they have the correct directories, packages, and processes required to continue the install. When you are satisfied with the list of hosts, click Next.

For detailed instructions, go to https://docs.hortonworks.com/HDPDocuments/Ambari-2.6.2.2/bk_ambari-installation/content/confirm_hosts.html.

6. On the Assign Slaves and Clients page, select the Clients that should be installed on the new hosts. To install clients on all hosts, select the Client checkbox for every host. You can use the all option for each available client to expedite this.



Make sure no other services are running on these hosts. To make this easier, select the none option for all other services.

7. On the Configurations page, select the "[configuration groups](#)" for the new hosts.
8. The Review page displays the host assignments you have made. Check to make sure everything is correct. If you need to make changes, use the left navigation bar to return to the appropriate screen.
9. Click Deploy.
10. The Install, Start and Test page displays progress as the clients are installed and deployed on each host. When the process is complete, click Next.
11. The Summary page provides you a list of the accomplished tasks. Click Complete and you will be directed back to the Hosts page.

Create HDFS User Directories

To run workloads that leverage HDP cluster services, make sure that HDFS directories (/user/<username>) are created for each user so that they can seamlessly connect to HDP from Cloudera Data Science Workbench.

Follow these steps for each user directory that must be created:

1. SSH to a host in the cluster that includes the HDFS client.
2. Switch to the `hdfs` system account user:

```
su - hdfs
```

3. Create an HDFS directory for the user. For example, you would create the following directory for the default user `admin`:

```
hdfs dfs -mkdir /user/admin
```

4. Assign ownership of the new directory to the user. For example, for the new `/user/admin` directory, make the `admin` user the owner of the directory:

```
hdfs dfs -chown admin:hadoop /user/admin
```

Install Cloudera Data Science Workbench on the Master Host

To install Cloudera Data Science Workbench on the master host, follow these steps:



The airgapped clusters and non-airgapped clusters use different files for installation. We used non-airgapped Installations

1. Download Cloudera Data Science Workbench:

For non-airgapped installations, download this file and save it to `/etc/yum.repos.d/`: [cloudera-cdsw.repo](#)

2. The Cloudera Public GPG repository key verifies that you are downloading genuine packages. Add the repository key:

```
sudo rpm --import https://archive.cloudera.com/cdsw1/1.5.0/redhat7/yum/RPM-GPG-KEY-cloudera
```

3. Install the latest RPM with the following command:

```
sudo yum install cloudera-data-science-workbench
```

4. Initialize and start Cloudera Data Science Workbench:

```
cdsw start
```

The application will take a few minutes to bootstrap. You can watch the status of the application installation and startup with `watch cdsw status`.

Install Cloudera Data Science Workbench on Worker Hosts

To install Cloudera Data Science Workbench on worker hosts, follow these steps:



Worker hosts are not required for a fully-functional Cloudera Data Science Workbench deployment. For proof-of-concept deployments, you can deploy a 1-host cluster with just a Master host. The Master host can run user workloads just as a worker host can.

1. Download Cloudera Data Science Workbench:

For non-airgapped installations, download this file and save it to `/etc/yum.repos.d/`: [cloudera-cdsw.repo](#)

2. The Cloudera Public GPG repository key verifies that you are downloading genuine packages. Add the repository key:

```
sudo rpm --import https://archive.cloudera.com/cdsw1/1.5.0/redhat7/yum/RPM-GPG-KEY-cloudera
```

3. Install the latest RPM with the following command:

```
sudo yum install cloudera-data-science-workbench
```

4. Copy cdsw.conf file from the master host:

```
scp root@<cdsw-master-hostname.your_domain.com>:/etc/cdsw/config/cdsw.conf  
/etc/cdsw/config/cdsw.conf
```

After initialization, the cdsw.conf file includes a generated bootstrap token that allows worker hosts to securely join the cluster. You can get this token by copying the configuration file from master and ensuring it has 600 permissions.

If your hosts have heterogeneous block device configurations, modify the Docker block device settings in the worker host configuration file after you copy it. Worker hosts do not need application block devices, which store the project files and database state, and this configuration option is ignored.

5. Create /var/lib/cdsw on the worker host. This directory must exist on all worker hosts. Without it, the next step that registers the worker host with the master will fail.
6. Unlike the master host, the /var/lib/cdsw directory on worker hosts does not need to be mounted to an Application Block Device. It is only used to store client configuration for HDP services on workers.
7. Initialize and start Cloudera Data Science Workbench:

```
cdsw join
```

This causes the worker hosts to register themselves with the Cloudera Data Science Workbench master host and increase the available pool of resources for workloads.

8. Return to the master host and verify the host is registered with this command:

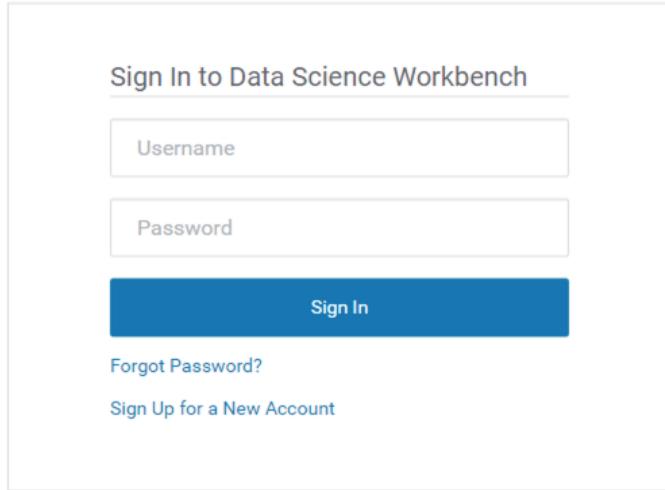
```
cdsw status
```

Create the Administrator Account

After your installation is complete, set up the initial administrator account.

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



In this CVD, Kerberos is not enabled.



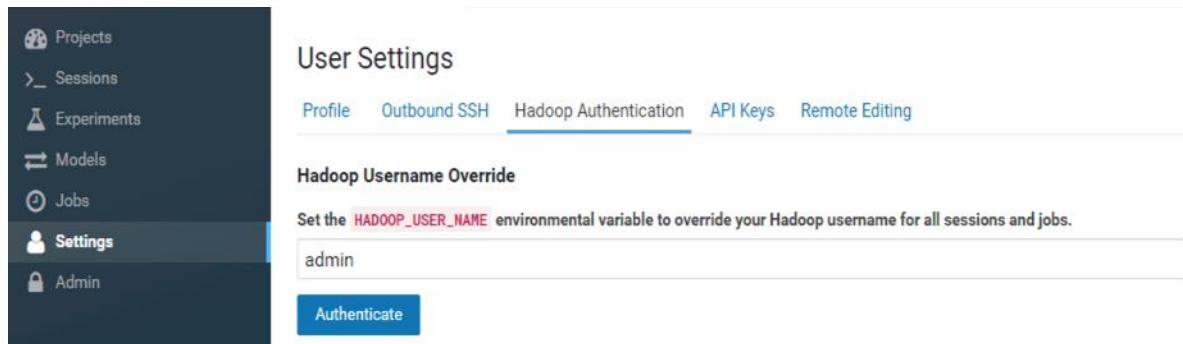
To enable Kerberos, refer to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_kerberos.html.

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.

```
# sudo -u hdfs hdfs dfs -mkdir /user/admin
# sudo -u hdfs hdfs dfs -chown admin:admin /user/admin
```

```
[root@rhel01 ~]# hdfs dfs -ls /user
Found 8 items
drwxr-xr-x  - admin  admin          0 2019-08-20 15:17 /user/admin
drwxrwxrwx  - mapred hadoop        0 2019-08-08 16:37 /user/history
drwxrwxr-t  - hive   hive          0 2019-08-08 16:37 /user/hive
drwxrwxr-x  - hue    hue           0 2019-08-08 16:38 /user/hue
drwxrwxr-x  - impala impala        0 2019-08-08 16:36 /user/impala
drwxrwxr-x  - oozie  oozie         0 2019-08-08 16:37 /user/oozie
drwxr-x--x  - spark  spark         0 2019-08-08 16:37 /user/spark
drwxr-xr-x  - hdfs  supergroup     0 2019-08-08 16:37 /user/yarn
```

4. Go to the Settings on Cloudera Data Science WorkBench, Click on Hadoop Authentication > Hadoop Username Override. Enter admin. Click on Authenticate.



5. Restart CDSW from command-line using “cdsw stop” followed by “cdsw start”.

Use 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.



For more information, go to: https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsw_gpu.html.

Enable 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. Create your own custom CUDA-capable engine image using the instructions described in this topic.
- Cloudera Data Science Workbench does not install or configure the NVIDIA drivers on the Cloudera Data Science Workbench gateway hosts. These depend on your GPU hardware and will have to be installed by your system administrator. The steps provided in this topic are generic guidelines that will help you evaluate your setup.
- The instructions described in this topic require Internet access. If you have an airgapped deployment, you will be required to manually download and load the resources onto your hosts.
- For a list of known issues associated with this feature, refer Known Issues - [GPU Support](#).

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 removed and since by default CDSW version 1.5 does not come with nvidia-docker 1.0 plugins which is a thin wrapper around the Docker CLI and a Docker plugin, shall be installed separately.

- Set the following parameter in /etc/cdsd/config/cdsd.conf on all Cloudera Data Science Workbench hosts. You must make sure that cdsd.conf is consistent across all hosts, irrespective of whether they have GPU hardware installed on them.

NVIDIA_GPU_ENABLE	Set this property to <code>true</code> to enable GPU support for Cloudera Data Science Workbench workloads. When this property is enabled on a host that is equipped with GPU hardware, the GPU(s) will be available for use by Cloudera Data Science Workbench.
--------------------------	--

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.



CDSW 1.5 does not ship with nvidia-docker version 1.0 and shall be installed independently.

- Run a small container to create the Docker volume structure

```
#docker run --rm nvidia/cuda nvidia-smi
```

```
[root@rhel23 ~]# docker run --rm nvidia/cuda nvidia-smi
Tue Aug 27 21:01:52 2019
+-----+
| NVIDIA-SMI 418.67      Driver Version: 418.67      CUDA Version: 10.1 |
+-----+
| GPU  Name     Persistence-M| Bus-Id      Disp.A  | Volatile Uncorr. ECC |
| Fan  Temp   Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
|-----+-----+-----+-----+-----+-----+-----+-----+
|  0  Tesla T4           off | 00000000:5E:00.0 Off |          0 |                |
| N/A   25C    P8    9W / 70W |     0MiB / 15079MiB |     0%   Default |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  1  Tesla T4           off | 00000000:AF:00.0 Off |          0 |                |
| N/A   25C    P8    9W / 70W |     0MiB / 15079MiB |     0%   Default |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| Processes:                               GPU Memory |
| GPU     PID  Type  Process name        Usage      |
|-----+-----+-----+-----|
| No running processes found               |
+-----+
```

Use the following Docker command to verify that Cloudera Data Science Workbench can access the GPU:

```
#vi nvidia-gpu-access.sh
mkdir /var/lib/nvidia-docker/
mkdir /var/lib/nvidia-docker/volumes/
mkdir /var/lib/nvidia-docker/volumes/nvidia_driver/
mkdir /var/lib/nvidia-docker/volumes/nvidia_driver/418.67
mkdir /var/lib/nvidia-docker/volumes/nvidia_driver/418.67/bin
mkdir /var/lib/nvidia-docker/volumes/nvidia_driver/418.67/lib64
cp /usr/bin/nvidia* /var/lib/nvidia-docker/volumes/nvidia_driver/418.67/bin
cp /usr/lib64/libcuda* /var/lib/nvidia-docker/volumes/nvidia_driver/418.67/lib64
cp /usr/lib64/libnvidia* /var/lib/nvidia-docker/volumes/nvidia_driver/418.67/lib64
```



Please replace the version of NVidia driver to one that matches your environment. Run "nvidia-smi" to capture the running version of NVidia driver.

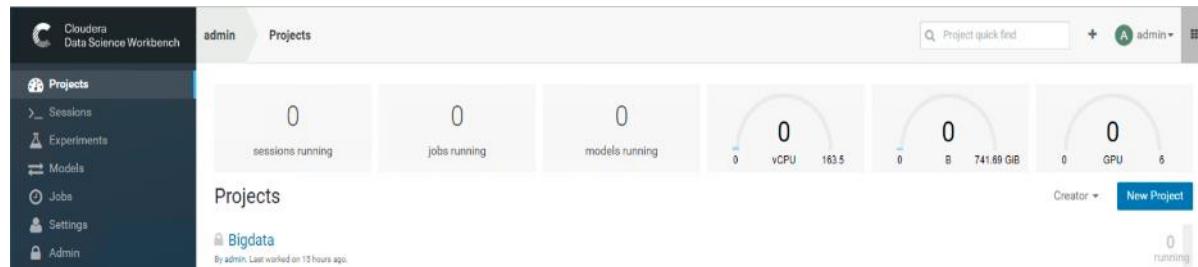
3. In CDSW, local mount point (/usr/local/nvidia) for the docker containers will be established manually while creating the directory and copying the files from the CUDA library locations:

```
# docker run --net host \
--device=/dev/nvidiactl \
--device=/dev/nvidia-uvm \
--device=/dev/nvidia0 \
-v /var/lib/nvidia-docker/volumes/nvidia_driver/418.67:/usr/local/nvidia/ \
-it nvidia/cuda \
/usr/local/nvidia/bin/nvidia-smi
```

```
Tue Aug 27 21:08:12 2019
+-----+
| NVIDIA-SMI 418.67      Driver Version: 418.67      CUDA Version: 10.1 |
+-----+
| 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 T4      off   00000000:5E:00.0 Off  0MiB / 15079MiB | 0%      Default | |
| N/A   25C   P8    9W / 70W |                   0MiB / 15079MiB |          0      |
| 1  Tesla T4      off   00000000:AF:00.0 Off  0MiB / 15079MiB | 0%      Default |
| N/A   25C   P8    9W / 70W |                   0MiB / 15079MiB |          0      |
+-----+
Processes:
+-----+
| GPU  PID  Type  Process name               GPU Memory Usage |
|-----+
| No running processes found                         |
+-----+
```

Test Cloudera Data Science Workbench to Detect GPUs

Once Cloudera Data Science Workbench has successfully restarted, if NVIDIA drivers have been installed on the Cloudera Data Science Workbench hosts, Cloudera Data Science Workbench will now be able to detect the GPUs available on its hosts.



Create a Custom CUDA-Capable Engine Image

The base engine image (`docker.repository.cloudera.com/cdsweb/engine:8`) 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, go to https://www.cloudera.com/documentation/data-science-workbench/latest/topics/cdsweb_gpu.html.

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@rhel23 ~]# docker run -d -p 5000:5000 --restart=always --name registry -v /mnt/registry:/var/lib/registry registry:2
[Unabled to find image 'registry:2' locally
[2: Pulling from library/registry
c87736221ed0: Pull complete
1cc8e0bb44df: Pull complete
54d33bcb37f5: Pull complete
e8afc091c171: Pull complete
b4541f6d3db6: Pull complete
Digest: sha256:8004747f1e8cd820a148fb7499d71a76d45ff66bac6a29129bfdbfdc0154d146
Status: Downloaded newer image for registry:2
966c9599941764019519bc1c42747b111cbfa994e1fcc34080bff0d92be17922
```

```
[root@rhel23 ~]# docker pull nvidia/cuda:10.1-base
10.1-base: Pulling from nvidia/cuda
[7413c47ba209: Already exists
0fe7e7cbb2e8: Already exists
1d425c982345: Already exists
344da5c95cec: Already exists
43bcc41986db: Already exists
76661327d908: Already exists
abdc887b90e5: Already exists
Digest: sha256:c94fd0a8f25122ee74553e78469bd342124dde1abe3a5bfffcccc84969f4ce2ac
Status: Downloaded newer image for nvidia/cuda:10.1-base
```

```
[root@rhel23 ~]# docker tag nvidia/cuda:10.1-base cdsw1.hdp3.cisco.com:5000/nvidia-cuda-cisco-demo
[root@rhel23 ~]# docker push cdsw1.hdp3.cisco.com:5000/nvidia-cuda-cisco-demo
[The push refers to a repository [cdsw1.hdp3.cisco.com:5000/nvidia-cuda-cisco-demo]
[Get https://cdsw1.hdp3.cisco.com:5000/v1/_ping: http: server gave HTTP response to HTTPS client
```

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 1.14.0 uses CUDA 10.0 and requires cuDNN 7.4.

1. To create the cuda.Dockerfile, run the following command:

```
FROM docker.repository.cloudera.com/cdsw/engine:8

RUN NVIDIA_GPGKEY_SUM=d1be581509378368edeec8c1eb2958702feedf3bc3d17011adb24efacce4ab5 && \
    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 10.1.168
LABEL com.nvidia.cuda.version="${CUDA_VERSION}"

ENV CUDA_PKG_VERSION 10-1=${CUDA_VERSION}-1
RUN apt-get update && apt-get install -y --no-install-recommends \
    cuda-cudart-${CUDA_PKG_VERSION} && \
    ln -s cuda-10.1 /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 7.6.2.24
LABEL com.nvidia.cudnn.version="${CUDNN_VERSION}"

RUN apt-get update && apt-get install -y --no-install-recommends \
    libcudnn7=${CUDNN_VERSION}-1+cuda10.1 && \
    apt-mark hold libcudnn7 && \
    rm -rf /var/lib/apt/lists/*

```

- Build a [custom engine image](#) out of cuda.Dockerfile using the following sample command:

```
docker build --network host -t <company-registry>/cdsw-cuda:8 . -f cuda.Dockerfile
```

```

[root@rhel23 ~]# docker build --network host -t cdsw1.hdp3.cisco.com/cdsw-cuda:8 . -f cuda.Dockerfile
Sending build context to Docker daemon 4.136 GB
Step 1/14 : FROM docker.repository.cloudera.com/cdsw/engine:8
--> a583d1499187
Step 2/14 : RUN NVIDIA_GPGKEY_SUM=d1be58b1509378368edeec8c1eb2958702feedf3bc3d17011adb24efacce4ab5 && 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
--> Running in 03c42d238cc4
Executing: /tmp/tmp.IxWlbcM0c/gpg.1.sh --fetch-keys
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub
gpg: key 7FA2AF80: public key <cudatools@nvidia.com> imported
gpg: Total number processed: 1
gpg:           imported: 1 (RSA: 1)
cudasign.pub: OK
--> f3aa407017e1
Removing intermediate container 03c42d238cc4
Step 3/14 : ENV CUDA_VERSION 10.1.168
--> Running in 80bb7ffd820d
--> 0d192d9e0329
Removing intermediate container 80bb7ffd820d
Step 4/14 : LABEL com.nvidia.cuda.version "${CUDA_VERSION}"
--> Running in fd8c2afffcbc
--> 8843bd799c42
Removing intermediate container fd8c2afffcbc
Step 5/14 : ENV CUDA_PKG_VERSION 10-1=${CUDA_VERSION}-1
--> Running in 929172d0e780
--> c723b19a7a9d
Removing intermediate container 929172d0e780
Step 6/14 : RUN apt-get update && apt-get install -y --no-install-recommends cuda-cudart-${CUDA_PKG_VERSION} && ln -s cuda-10.1 /usr/local/cuda && rm -rf /var/lib/apt/lists/*
--> Running in c7f4ad476b6b

```

3. Push this new engine image to a public Docker registry so that it can be made available for Cloudera Data Science Workbench workloads. For example:

```
docker push <company-registry>/cdsw-cuda:8
```

```
[root@rhel23 ~]# docker push cdsw1.hdp3.cisco.com:5000/cisco-gpu-demo
The push refers to a repository [cdsw1.hdp3.cisco.com:5000/cisco-gpu-demo]
Get https://cdsw1.hdp3.cisco.com:5000/v1/_ping: http: server gave HTTP response to HTTPS client
```



- 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>

Image Management

Images can be preloaded on all NodeManager hosts or they can be implicitly pulled at runtime if they are available in a public Docker registry, such as Docker hub. If the image does not exist on the NodeManager and cannot be pulled, the container will fail.



- For AI framework specific images, it is recommended using Docker images from NG Cloud, which is described in subsequent sections of this document.



- It is also recommended to have a private Docker image repository. This CVD details the setup of a private registry for simplicity but doesn't go into details on a registry setup with high availability and is provided only as an example.



- On a Multi-GPU, server, the output of this command will show all the GPUs for some reason. This behavior needs to be further validated with the vendors.

Configure CDSW to Run Docker Containers

Set Up Docker Registry

Private trusted registry is required to provision YARN container. This topic provides basic information about deploying and configuring a registry.



- This is a sample registry to showcase the use-case and not recommended for Production grade setup.

Designate a Server for Docker and Start the Registry

To designate a server for Docker and start the registry, follow these steps:

1. Designate a server in the cluster for the Docker registry. Minimal resources are required, but sufficient disk space is needed to store the images and metadata. Docker must be installed and running.
2. Optional: By default, data will only be persisted within the container. If you would like to persist the data on the host, you can customize the bind mounts using the -v option.

3. Create /var/lib/registry folder:

```
# mkdir /var/lib/registry
```

4. Configure Docker to allow pulling from this insecure registry. Modify /etc/docker/daemon.json on all nodes in the cluster to include the following configuration options:

```
# cat /etc/docker/daemon.json
{
    "live-restore" : true,
    "debug" : true,
    "insecure-registries" : ["linuxjh.hdp3.cisco.com:5000"]
}
```

5. Restart Docker on all nodes.

6. Provision the registry container by running the following command:

```
docker run -d -p 5000:5000 --restart=always --name registry -v
/mnt/registry:/var/lib/registry registry:2
```

7. Verify the registry container is provisioned by running docker ps command:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			
037d176d2576	registry:2	"/entrypoint.sh /etc..."	14 minutes ago	Up 4
minutes	0.0.0.0:5000->5000/tcp	registry		

Create a Custom CUDA-capable Engine Image

```
[root@rhel17 ~]# docker run -d -p 5000:5000 --restart=always --name registry registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry
c87736221ed0: Pull complete
1cc8e0bb44df: Pull complete
54d33bcb37f5: Pull complete
e8afc091c171: Pull complete
b4541f6d3db6: Pull complete
Digest: sha256:8004747f1e8cd820a148fb7499d71a76d45ff66bac6a29129bfdbfdc0154d146
Status: Downloaded newer image for registry:2
1e294ea97d773d296c2dbf516a744c76b24048e82c2d665b248c304d58158607
[root@rhel17 ~]# docker build --network host -t localhost/cdh-cdip-demo:1 . -f
cuda.Dockerfile
Sending build context to Docker daemon 7.017 GB
Step 1/14 : FROM docker.repository.cloudera.com/cdsw/engine:8
--> a583d1499187
Step 2/14 : 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
--> Running in 1a0dde1e2df2
Executing: /tmp/tmp.YMCjemm0nM/gpg.1.sh --fetch-keys
```

```

http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub
gpg: key 7FA2AF80: public key "cudatools <cudatools@nvidia.com>" imported
gpg: Total number processed: 1
gpg:           imported: 1  (RSA: 1)
cudasign.pub: OK
Selecting previously unselected package cuda-license-10-1.
(Reading database ... 106755 files and directories currently installed.)
Preparing to unpack .../cuda-license-10-1_10.1.243-1_amd64.deb ...
Unpacking cuda-license-10-1 (10.1.243-1) ...
Selecting previously unselected package cuda-cudart-10-1.
Preparing to unpack .../cuda-cudart-10-1_10.1.168-1_amd64.deb ...
Unpacking cuda-cudart-10-1 (10.1.168-1) ...
Setting up cuda-license-10-1 (10.1.243-1) ...
*** LICENSE AGREEMENT ***
By using this software you agree to fully comply with the terms and
conditions of the EULA (End User License Agreement). The EULA is located
at /usr/local/cuda-10.1/doc/EULA.txt. The EULA can also be found at
http://docs.nvidia.com/cuda/eula/index.html. If you do not agree to the
terms and conditions of the EULA, do not use the software.

Get:26 http://archive.ubuntu.com/ubuntu xenial-updates/restricted amd64 Packages [13.1 kB]
Get:27 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [983 kB]
Get:28 http://archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 Packages [19.1 kB]
Get:29 http://archive.ubuntu.com/ubuntu xenial-backports/main amd64 Packages [7,942 B]
Get:30 http://archive.ubuntu.com/ubuntu xenial-backports/universe amd64 Packages [8,807 B]
Fetched 16.3 MB in 7s (2,304 kB/s)
Reading package lists...
W: http://archive.cloudera.com/kudu/ubuntu/xenial/amd64/kudu/dists/xenial-kudu5/InRelease:
Signature by key F36A89E33CC1BD0F71079007327574EE02A818DD uses weak digest algorithm (SHA1)
Reading package lists...
Building dependency tree...
Reading state information...
The following NEW packages will be installed:
  libcudnn7
0 upgraded, 1 newly installed, 0 to remove and 53 not upgraded.
Need to get 181 MB of archives.
After this operation, 435 MB of additional disk space will be used.
Get:1 http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1604/x86_64
libcudnn7 7.6.2.24-1+cuda10.1 [181 MB]
Fetched 181 MB in 3s (48.4 MB/s)
Selecting previously unselected package libcudnn7.
(Reading database ... 106775 files and directories currently installed.)
Preparing to unpack .../libcudnn7_7.6.2.24-1+cuda10.1_amd64.deb ...
Unpacking libcudnn7 (7.6.2.24-1+cuda10.1) ...
Processing triggers for libc-bin (2.23-0ubuntu11) ...
Setting up libcudnn7 (7.6.2.24-1+cuda10.1) ...
Processing triggers for libc-bin (2.23-0ubuntu11) ...
libcudnn7 set on hold.
  --> elf6c9eb21ae
Removing intermediate container 0359fa10d1b7
Successfully built elf6c9eb21ae
[root@rhel17 ~]#

```

Push Image to the Local Registry

```

[root@rhel17 ~]# docker push localhost:5000/cdh-cdip-demo
The push refers to a repository [localhost:5000/cdh-cdip-demo]
cb2ec428a23f: Preparing
2b90248bbef5: Preparing
38aad97b5250: Preparing
bd674e93042a: Preparing
219140170d8f: Preparing
f1baf9cfdd99: Pushed

```

```

4ac7dde4d412: Pushed
678edc9f70e4: Pushed
b6bd66b29fce: Pushed
40b3a57bc14b: Pushed
9e5cf89282a3: Pushed
0b3c4fae714a: Pushed
7e9c16b43b28: Pushed
c1d79d93e9cc: Pushed
b2eb07d9f14e: Pushed
0f6344a53716: Pushed
529336ddfdde: Pushed
706ac5c1c7d2: Pushed
4a9eea23fa45: Pushed
92d3f22d44f3: Pushed
10e46f329a25: Pushed
24ab7de5faec: Pushed
1ea5a27b0484: Pushed
latest: digest: sha256:7e02caa111131b19c0c1c28dd177964c04b08bf59d432635b283c3cfa82905f2 size:
23951

```

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.

To allocate GPUs for sessions and jobs, follow these steps:

6. Sign into Cloudera Data Science Workbench as a site administrator.
7. Click Admin.
8. Go to the Engines tab.
9. From the Maximum GPUs per Session/Job drop-down list, select the maximum number of GPUs that can be used by an engine.
10. 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. Enter description and Tag created for repository. Click Add.

The screenshot shows the Cloudera Data Science Workbench Site Administration interface. On the left is a dark sidebar with icons for Projects, Sessions, Experiments, Models, Jobs, Settings, and Admin. The Admin icon is selected. The main area has tabs for Admin and Engines, with Engines selected. Below that is a sub-tab for Site Administration.

Engines Profiles

Description	vCPU (burstable)	Memory (GiB)	Actions
1 vCPU / 2 GiB Memory	1	2	Edit Delete
1 vCPU / 1.75 GiB Memory	1	1.75	Edit Delete
2 vCPU / 4 GiB Memory	2	4	Edit Delete
4 vCPU / 8 GiB Memory	4	8	Edit Delete
1 vCPU (burstable), 1.75 GiB memory	1	1.75	Edit Delete Add

Maximum GPUs per Session/Job

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.

Engine Images

Description	Repository.Tag	Default	Actions
Base Image v7	docker.repository.cloudera.com/cdswh/engine:7	<input checked="" type="radio"/>	Edit Deprecate
CDIP CDSW CUDA	rhel70.hdp3.cisco.local:5000/cdip-cuda	<input type="radio"/>	Edit Deprecate
			Add

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.

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 existing project or create a new project and choose Settings.
3. Go to the Engines tab.
4. Under Engine Image, add the CUDA-capable engine image.

Project Settings

Engine Image

Select the Docker image that Cloudera Data Science Workbench should use to run sessions and jobs in this project. If you'd like to use a different image, contact your site administrator.

CDIP CDSW CUDA, rhel20.hdp3.cisco.local:5000/cdip-cuda

Environmental Variables

Set project environmental variables that can be accessed from your scripts.

Name	Value	Actions
		Add

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.

5. Go to the Sessions tab and click an existing session or create a new session and allocate resources.
6. Click Terminal Access.



About the Authors

Yogesh Ramesh, Big Data Solutions Architect, Computing Systems Product Group, Cisco Systems, Inc.

Yogesh Ramesh is a Big Data Solutions Architect in the Computing Systems Product Group. He is part of the solution engineering team focusing on big data infrastructure, solutions, and performance.

Muhammad Afzal, Engineering Architect, Computing Systems Product Group, Cisco Systems, Inc.

Muhammad Afzal is an Engineering Architect and Technical Marketing Engineer in Cisco UCS Product Management and Datacenter Solutions Engineering. He is currently responsible for designing, developing, and producing validated architectures for Big Data and analytics while working collaboratively with product partners. Previously, Afzal had been a lead architect for various cloud and data center solutions in Solution Development Unit at Cisco. Prior to this, Afzal has been a Solutions Architect in Cisco's Advanced Services group, where he worked closely with Cisco's large enterprise and service provider customers delivering data center and cloud solutions. Afzal holds an MBA in Finance and a BS in Computer Engineering.

Acknowledgements

For their support and contribution to the design, validation, and creation of this Cisco Validated Design, the authors would like to thank:

- Karthik Kulkarni, Architect, Computing Systems Product Group, Cisco Systems, Inc.
- Silesh Bijjahalli, Product Management, Computing Systems Product Group, Cisco Systems, Inc.
- Ali Bajwa, Hortonworks
- Wangda Tan, Hortonworks
- Harsh Shah, Hortonworks
- Sicong Ji, NVIDIA