Cloudera Manager 7.11.3

Installation

Date published: 2020-11-30
Date modified: 2024-07-19

https://docs.cloudera.com/

Legal Notice

Contents

Cloudera Private Cloud Base Installation Guide..7

Version and Download Information...7

Cloudera Manager Version Information..7
Cloudera Manager Download Information..8
Cloudera Runtime Version Information...10
Cloudera Runtime Download Information..10
Cloudera Manager support for CDH, Cloudera Runtime and CDP Private Cloud Data Services.....................10
Cloudera Private Cloud Base Trial Download Information...12

Cloudera Private Cloud Base Requirements and Supported Versions............ 12

Hardware Requirements...13
Cloudera Manager..13
Cloudera Runtime...17
Operating System Requirements...27
Database Requirements..32
RDBMS High Availability Support..33
Java Requirements...35
Supported JDKs...35
Support Notes..37
Networking and Security Requirements..37
Data at Rest Encryption Requirements..43
Third-party filesystems..44

Trial Installation... 44

Installing a Trial Cluster...45
Before You Begin a Trial Installation...45
Download the Trial version of Cloudera Private Cloud Base...................................46
Run the Cloudera Manager Server Installer...46
Install Cloudera Runtime..51
Set Up a Cluster Using the Wizard..64
Stopping the Embedded PostgreSQL Database..67
Starting the Embedded PostgreSQL Database...67
Changing Embedded PostgreSQL Database Passwords..68
Migrating from the Cloudera Manager Embedded PostgreSQL Database Server to an External PostgreSQL Database.....................69
Prerequisites..70
Identify Roles that Use the Embedded Database Server.......................................70
Migrate Databases from the Embedded Database Server to the External PostgreSQL Database Server.....................72

Installing and Configuring CDP with FIPS..75

Overview...75
Prerequisites for using FIPS..77
About CDP with FIPS..77

Step 1: Prepare hosts...79
Step 2: Install and configure the SafeLogic modules and packages for RHEL 7 OS...85
Step 3: Install Cloudera Manager server...86
Step 4: Validate the CCJ and CCS installation...86
Step 5: Install and configure databases...88
Configure Cloudera Manager for FIPS...89
Install and configure additional required components...89

Production Installation...91
Before You Install...91
Installing Python 3...92
Storage Space Planning for Cloudera Manager...113
Configure Network Names...122
Setting SELinux Mode...123
Disabling the Firewall...124
Enable an NTP Service...124
Impala Requirements...125
Runtime Cluster Hosts and Role Assignments...127
Allocating Hosts for Key Trustee Server and Key Trustee KMS...134
Configuring Local Package and Parcel Repositories...134
Configuring /tmp directory for cluster hosts...141
Production Installation: Installing Cloudera Manager, Cloudera Runtime, and Managed Services...142
Step 1: Configure a Repository for Cloudera Manager...143
Step 2: Install Java Development Kit...145
Installing OpenJDK on Cloudera Manager...146
Installing OpenJDK for CDP Runtime...147
Installing Oracle JDK for CDP Runtime...150
Step 3. Install and Configure Databases...150
Required Databases...151
Install and Configure PostgreSQL for CDP...152
Install and Configure MySQL for Cloudera Software...160
Install and Configure MariaDB for Cloudera Software...168
Configure Oracle Database for Cloudera Software...175
Enabling TLS 1.2 on Database Server...179
Enabling Kerberos (MIT and AD) authentication for MariaDB Database Server...186
Setting up the database for Reports Manager...190
Configuring a database for Ranger or Ranger KMS...191
Configuring the Database for Streaming Components...201
Step 4: Deploy Cloudera Manager Server and Cloudera Manager Agents...204
Step 5: Set up and configure the Cloudera Manager database...206
Configuring TLS 1.2 for Cloudera Manager...207
Configuring TLS 1.2 for Reports Manager...212
Syntax for scm_prepare_database.sh...213
Configuring Oracle RAC for the Cloudera Manager database...217
PostgresSQL High Availability (HA)...218
Step 6: Start the Cloudera Manager Server and Agents...218
Adding the Reports Manager role...219
Step 7: Using the installation wizard...220
Step 8: Set Up a Cluster Using the Wizard...229
Select Services...229
Assign Roles...230
Setup Database...230
Enter Required Parameters...238
Review Changes...241
Configure Kerberos...242

Command Details...242
Summary... 242
Tuning JVM Garbage Collection... 242
(optional) Enable high availability for Cloudera Manager... 245
(Recommended) Enable Auto-TLS... 245
(Recommended) Enable Kerberos...245
Additional Steps for Apache Ranger... 245
Enable Plugins... 245
Add Solr WebUI Users... 246
Update the Time-to-live configuration for Ranger Audits...246
Installing Apache Knox...246
Apache Knox Install Role Parameters... 248
Setting Up Data at Rest Encryption for HDFS... 249
Installing Ranger KMS backed by a Database and HA... 250
Installing Ranger KMS backed with a Key Trustee Server and HA... 256
Installing a Java Keystore KMS... 270
Installing Cloudera Navigator Key Trustee Server...275
Installing Key Trustee Server using Cloudera Manager...275
Securing Key Trustee Server Host...276
Leveraging Built-in Processor Instruction Sets...277
Initializing Key Trustee Server... 278
Installing Cloudera Navigator Encrypt...278
Setting Up an Internal Repository...278
Installing Navigator Encrypt (RHEL-Compatible)... 278
Installing Navigator Encrypt (SLES-12)...280
Installing Navigator Encrypt (SLES-15)...280
Installing Navigator Encrypt (Ubuntu)...282
Installing for Ranger KMS...283
Setting Up TLS for Navigator Encrypt Clients... 283
Entropy Requirements... 284
Uninstalling and Reinstalling Navigator Encrypt... 285
Installing Cloudera Navigator Key HSM...286
Additional steps for FIPS + JDK 11...286
Installing Ranger RMS... 287

Custom Installation Solutions...296
Privileged commands for Cloudera Manager installation...296
Prerequisites and exceptions for the example configuration...296
Example configuration to add to the sudoers file...297
Creating Virtual Images of Cluster Hosts...298
Creating a Pre-Deployed Cloudera Manager Host... 298
Instantiating a Cloudera Manager Image... 299
Creating a Pre-Deployed Worker Host... 300
Instantiating a worker host... 301
Manually Install Cloudera Software Packages... 301
Install Cloudera Manager Packages... 301
Manually Install Cloudera Manager Agent Packages...302

Installation Reference...303
Ports...303
Ports Used by Cloudera Manager... 303
Ports Used by Cloudera Navigator Key Trustee Server... 308
Ports Used by Cloudera Runtime Components... 308
Ports Used by DistCp...315

Ports Used by Third-Party Components.... 316
Service Dependencies in Cloudera Manager.... 317
Cloudera Manager sudo command options.... 319
Introduction to Parcels.... 320

After You Install.... 320
Deploying Clients.... 321
Initializing Solr and creating HDFS home directory.... 321
Testing the Installation.... 321
Checking Host Heartbeats.... 321
Running a MapReduce Job.... 322
Testing with Hue.... 322
Deploying Atlas service.... 323
Secure Your Cluster.... 325
Installing the GPL Extras Parcel.... 325
Configuring HDFS properties to optimize log collection.... 326
Migrating a YARN Queue Manager database to an external database.... 327

Troubleshooting Installation Problems.... 328

Uninstalling Cloudera Manager and Managed Software.... 331
Record User Data Paths.... 332
Stop all Services.... 332
Deactivate and Remove Parcels.... 332
Delete the Cluster.... 333
Uninstall the Cloudera Manager Server.... 333
Uninstall Cloudera Manager Agent and Managed Software.... 333
Remove Cloudera Manager, User Data, and Databases.... 334
Uninstalling a Runtime Component From a Single Host.... 335

Cloudera Private Cloud Base Installation Guide

Use this Installation Guide to learn how to install Cloudera software, including Cloudera Manager, Cloudera Runtime, and other managed services, in a production or trial environment.

Related Information

Version and Download Information

Cloudera Private Cloud Base Requirements and Supported Versions

Trial Installation

Production Installation

Custom Installation Solutions

Installation Reference

After You Install

Troubleshooting Installation Problems

Uninstalling Cloudera Manager and Managed Software

Cloudera Manager Download Information

Cloudera Manager Version Information

Version and Download Information

The following topics describe the available versions and download locations for Cloudera Manager and Cloudera Runtime.

Related Information

Cloudera Private Cloud Base Installation Guide

Cloudera Manager Version Information

You must choose the correct Cloudera Manager for your deployment This page provides a reference of Cloudera Manager versions, their release dates, and important compatibility information.

Cloudera Manager 7.11.3 CHF7 is the current release of Cloudera Manager required for CDP Private Cloud Base version 7.1.9 SP1.

Cloudera Manager 7.11.3 CHF4 is the current release of Cloudera Manager required for CDP Private Cloud Base version 7.1.7 SP3.

Important:

You must install Python 3 on all hosts before installing or upgrading to Cloudera Manager 7.11.3. For more information, see the Installing Python 3.

Release date: July 19, 2024

Previous releases:

- Cloudera Manager 7.11.3 CHF4 Release Date: April 05, 2024
- Cloudera Manager 7.11.3 Release Date: August 18, 2023
- Cloudera Manager 7.10.1 Release Date: June 13, 2023
- Cloudera Manager 7.9.5 Release Date: January 25, 2023
- Cloudera Manager 7.8.1 Release Date: November 18, 2022
- Cloudera Manager 7.7.3 Release Date: October 28, 2022
- Cloudera Manager 7.7.1 Release Date: August 30, 2022

- Cloudera Manager 7.6.5 Release Date: May 25, 2022
- Cloudera Manager 7.6.1 (SP1) Release Date: March 30, 2022
- Cloudera Manager 7.5.5 Release Date: April 13, 2022
- Cloudera Manager 7.5.4-20668437 Release Date: January 13, 2022
- Cloudera Manager 7.5.4 Release Date: November 8, 2021
- Cloudera Manager 7.5.1 Release Date: October 4, 2021
- Cloudera Manager 7.4.4 Release Date: August 5, 2021
- Cloudera Manager 7.3.1 Release Date: March 3, 2021
- Cloudera Manager 7.2.4 Release Date: November 30 2020
- Cloudera Manager 7.1.4 Release Date: October 13, 2020
- Cloudera Manager 7.1.3 Release Date: August 10, 2020
- Cloudera Manager 7.1.2 Release Date: July 13, 2020
- Cloudera Manager 7.1.1 Release Date: May 22, 2020
- Cloudera Manager 7.0.3 Release Date: November 22, 2019

Related Information

Cloudera Private Cloud Base Installation Guide

Cloudera Manager Download Information

Important: Access to Cloudera Manager binaries for production purposes requires authentication. To access the binaries at the locations below, you must first have an active subscription agreement and obtain a license key file along with the required authentication credentials (username and password).

The license key file and authentication credentials are provided in an email sent to customer accounts from Cloudera when a new license is issued. If you have an existing license with a Cloudera Private Cloud Base Edition entitlement, you might not have received an email. In this instance you can identify the authentication credentials from the license key file. If you do not have access to the license key, contact your account representative to receive a copy.

To identify your authentication credentials using your license key file, complete the following steps:

- From cloudera.com, log into the cloudera.com account associated with the Cloudera Private Cloud Base license and subscription agreement.
- On the Cloudera Private Cloud BaseCDP Private Cloud Base Download page, click Download Now and scroll down to the Credential Generator.
- In the Generate Credentials text box, copy and paste the text of the “PGP Signed Message” within your license key file and click Get Credentials. The credentials generator returns your username and password.

Important: Make a note of the authentication credentials. You might need them during installation to complete tasks such as configuring a remote parcel repository, or installing Cloudera Manager packages using a package manager such as YUM, APT, or other tools that you might be using in your environment.

When you obtain your authentication credentials, use them to form the URL where you can access the Cloudera Manager repository in the Cloudera Archive.

Important:

Before performing an upgrade of Cloudera Manager or the CDP Runtime, creating a backup of all the metadata databases is important. This includes the Cloudera Manager database, and the various Runtime component databases such as Hive Metadata Server, Ranger Admin, Ranger KMS, Schema Registry, etc. The backups are necessary if there is a reason to rollback to the prior version.

For information on repositories for the Cloudera Manager 7.11.3 cumulative hotfixes, see Cumulative hotfixes.

The repositories for Cloudera Manager 7.11.3 are listed in the following tables:

Table 1: Cloudera Manager 7.11.3

Repository Type	Repository Location

RHEL 9 Compatible	Repository:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/redhat9/yum`
	Repository File:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/redhat9/yum/cloudera-manager.repo`

RHEL 8 Compatible	Repository:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/redhat8/yum`
	Repository File:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/redhat8/yum/cloudera-manager.repo`

RHEL 7 Compatible	Repository:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/redhat7/yum`
	Repository File:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/redhat7/yum/cloudera-manager.repo`

SLES 15	Repository:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/sles15/yum`
	Repository File:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/sles15/yum/cloudera-manager.repo`

SLES 12	Repository:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/sles12/yum`
	Repository File:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/sles12/yum/cloudera-manager.repo`

Repository Type	Repository Location

Ubuntu 20	Repository:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/ubuntu2004/apt`
	Repository file:
	`https://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/7.11.3/ubuntu2004/apt/cloudera-manager.list`

Related Information

Cloudera Private Cloud Base Installation Guide

Cloudera Runtime Version Information

Version numbers for current and previous releases of Cloudera Runtime 7.x.

Cloudera Runtime 7.1.9 SP1 is based on Apache Hadoop 3. For more information, see *Cloudera Runtime Component Versions*.

Release date: July 19, 2024

Previous releases:

- Cloudera Runtime 7.1.7 SP3 Release Date: May 07, 2024
- Cloudera Runtime 7.1.9 Release Date: September 08, 2023
- Cloudera Runtime 7.1.7 SP2 Release Date: January 31, 2023
- Cloudera Runtime 7.1.8 Release Date: August 30, 2022
- Cloudera Runtime 7.1.7 SP1 Release Date: March 30, 2022
- Cloudera Runtime 7.1.7 Release Date: August 5, 2021
- Cloudera Runtime 7.1.6 Release Date: March 3, 2021
- Cloudera Runtime 7.1.5 Release Date: November 30, 2020
- Cloudera Runtime 7.1.4 Release Date: October 13, 2020
- Cloudera Runtime 7.1.3 Release Date: August 10, 2020
- Cloudera Runtime 7.1.2 Release Date: July 13, 2020
- Cloudera Runtime 7.1.1 Release Date: May 22, 2020
- Cloudera Runtime 7.0.3 Release Date: November 22, 2019

Cloudera Runtime Download Information

Important: Access to Cloudera Runtime parcels for production purposes requires authentication. To access the parcels at the locations below, you must first have an active subscription agreement and obtain a license key file along with the required authentication credentials (username and password).

For information on Cloudera Runtime parcel, see the Cloudera Runtime Download Information documentation.

For information on Cloudera Runtime CHF, see the Cloudera Runtime CHF documentation.

Cloudera Manager support for CDH, Cloudera Runtime and CDP Private Cloud Data Services

Describes which versions of CDH, Cloudera Runtime and CDP Private Cloud Data Services are supported by Cloudera Manager.

Note: Not all combinations of Cloudera Manager, Cloudera Runtime, and CDP Private Cloud Data Services are supported. Ensure that the version of Cloudera Manager you are using supports the version of Cloudera Runtime and CDP Private Cloud Data Services you have selected. For more information, see the Cloudera Support Matrix.

The versions of Cloudera Runtime, CDP Private Cloud Data Services, and CDH clusters that can be managed by Cloudera Manager are limited to the following:

For CDP Private Cloud Base

Table 2: Cloudera Manager support for CDP Private Cloud Base

Cloudera Manager Version	Supported CDH/Runtime versions	Supported Cloudera Private Cloud Data Services versions
Cloudera Manager 7.11.3 Latest cumulative hotfix	• Cloudera Runtime 7.1.9 SP1 • Cloudera Runtime 7.1.7 SP3 • Cloudera Runtime 7.1.9 • Cloudera Runtime 7.1.7 SP2 • Cloudera Runtime 7.1.8 • Cloudera Runtime 7.1.7 SP1 • Cloudera Runtime 7.1.7 • Cloudera Runtime 7.1.6 • Cloudera Runtime 7.1.5 • Cloudera Runtime 7.1.4 • Cloudera Runtime 7.1.3 • Cloudera Runtime 7.1.2 • Cloudera Runtime 7.1.1 • Cloudera Runtime 7.0.3 • CDH 6.3 • CDH 6.2 • CDH 6.1 • CDH 6.0	None
Cloudera Manager 7.11.3 **Note:** You must install Python 3.8 (or 3.9 for RHEL 9.1) on all hosts before installing or upgrading to Cloudera Manager 7.11.3. For more information, see the Installing Python 3.	• Cloudera Runtime 7.1.9 • Cloudera Runtime 7.1.7 SP2 • Cloudera Runtime 7.1.8 • Cloudera Runtime 7.1.7 SP1 • Cloudera Runtime 7.1.7 • Cloudera Runtime 7.1.6 • Cloudera Runtime 7.1.5 • Cloudera Runtime 7.1.4 • Cloudera Runtime 7.1.3 • Cloudera Runtime 7.1.2 • Cloudera Runtime 7.1.1 • Cloudera Runtime 7.0.3 • CDH 6.3 • CDH 6.2 • CDH 6.1 • CDH 6.0	None

For CDP Private Cloud Data Services

Table 3: Cloudera Manager support for CDP Private Cloud Data Services

Cloudera Manager Version	Supported CDH/Runtime versions	Supported Cloudera Private Cloud Data Services versions
7.11.3 cumulative hotfix 11	Supported with Cloudera Runtime 7.1.7 SP3, 7.1.9 CHF7 or higher, and 7.1.9 SP1 CHF4 or higher only when CDP Private Cloud Data Services is deployed.	1.5.4 SP1 Only supported with Cloudera Runtime 7.1.7 SP3, 7.1.9 CHF7 or higher, and 7.1.9 SP1 CHF4 or higher.
7.11.3 cumulative hotfix 9.1 (version: 7.11.3.24)	Supported with Cloudera Runtime 7.1.7 SP3, 7.1.9 CHF6 or higher, and 7.1.9 SP1 only when CDP Private Cloud Data Services is deployed.	1.5.4 CHF3 Only supported with Cloudera Runtime 7.1.7 SP3, 7.1.9 CHF6 or higher, and 7.1.9 SP1.
7.11.3 cumulative hotfix 7 Data Services (version: 7.11.3.14)	Supported with Cloudera Runtime 7.1.7 SP3, 7.1.9 CHF5 or higher, and 7.1.9 SP1 only when CDP Private Cloud Data Services is deployed.	1.5.4 CHF1 Only supported with Cloudera Runtime 7.1.7 SP3, 7.1.9 CHF5 or higher, and 7.1.9 SP1.
7.11.3 cumulative hotfix 6	• Supported with Cloudera Runtime 7.1.7 SP3, 7.1.8 CHF22 or higher, and 7.1.9 CHF6 only when CDP Private Cloud Data Services is deployed.	1.5.4 Only supported with Cloudera Runtime 7.1.7 SP3, 7.1.8 CHF22 or higher, and 7.1.9 CHF6.
7.11.3 cumulative hotfix 4	• Supported with Cloudera Runtime 7.1.7 SP2, 7.1.8 CHF19 or higher, and 7.1.9 CHF3 only when CDP Private Cloud Data Services is deployed.	1.5.3 Only supported with Cloudera Runtime 7.1.7 SP2, 7.1.8 CHF19 or higher, and 7.1.9 CHF3.
7.11.3 cumulative hotfix 1	• Supported with Cloudera Runtime 7.1.7 SP2, 7.1.8 CHF11 or higher, 7.1.9, and 7.1.9 CHF1 only when CDP Private Cloud Data Services is deployed.	1.5.2 Only supported with Cloudera Runtime 7.1.7 SP2, 7.1.8 CHF11 or higher, 7.1.9, and 7.1.9 CHF1.

Cloudera Private Cloud Base Trial Download Information

You can try the Cloudera Private Cloud Base Edition of Cloudera Data Platform for 60 days without obtaining a license key file.

To download Cloudera Private Cloud Base without obtaining a license key file, visit the Cloudera Private Cloud Base Trial Download page, click Try Now, and follow the download instructions. When you install Cloudera Private Cloud Base without a license key, you are performing a trial installation that includes an embedded PostgreSQL database and is not suitable for a production environment. For more information on trial installations, see the trial installation documentation.

A 60-day trial of Cloudera Private Cloud Base Edition can be enabled permanently with the appropriate license. To obtain a Cloudera Private Cloud Base Edition license, fill in the Contact Us form or call 866-843-7207

Related Information

Trial Installation

Cloudera Private Cloud Base Requirements and Supported Versions

Refer to the following topics for information about hardware, operating system, and database requirements, as well as product compatibility matrices.

Related Information

Cloudera Private Cloud Base Installation Guide

Hardware Requirements

This topic specifies the hardware requirements for Cloudera Private Cloud Base.

As you create the architecture of your cluster, you will need to allocate Cloudera Manager and Runtime roles among the hosts in the cluster to maximize your use of resources. Cloudera provides some guidelines about how to assign roles to cluster hosts. See Recommended Cluster Hosts and Role Distribution. When multiple roles are assigned to hosts, add together the total resource requirements (memory, CPUs, disk) for each role on a host to determine the required hardware.

Attention: All recommendations for the number of cores refer to logical cores, not physical cores.

For more information about sizing for a particular component, see the following minimum requirements:

Cloudera Manager

Hardware requirements for Cloudera Manager Server and related components.

Cloudera Manager Server

Table 4: Cloudera Manager Server Storage Requirements

Component	Storage	Notes
Partition hosting /usr	1 GB	
Partition hosting /var	100 GB to 5 TB	Scales according to number of nodes managed. See table below.
Partition hosting /opt	100 GB minimum	Usage grows as the number of parcels downloaded increases. Budget 12 GB for each additional CDH parcel, and 1 GB for each additional non-CDH parcel.
Cloudera Manager Database Server	• < 500 hosts: 5 GB • > 500 hosts: 10 GB	Minimum memory and processor requirements should allow support for the following number of parallel database connections: • < 500 hosts: 100 database connections • > 500 hosts: 250 database connections
Reports Manager Database Server	Minimum 1 GB	Reports Manager growth depends on number of HDFS users and monitored directories.

Table 5: Host Based Cloudera Manager Server Requirements

Number of Cluster Hosts	Database Host Configuration and HMON+SMON host sharing	Cloudera Manager Server Heap Size	Logical Processors	Cloudera Manager Server /var Directory	SMON and HMON / var Directory
Very small (#10)	Shared	8 GB	4	50 GB	50 GB
Small (#20)	Shared	10 GB	6	100 GB	100 GB
Medium (#200)	Dedicated	16 GB	8	1 TB	1 TB
Large (#500)	Dedicated	32 GB	12	2.5 TB	2.5 TB
Extra Large (>500)	Dedicated	48 GB	16	> 2.5 TB	> 2.5 TB

Important: For medium and larger clusters, Host Monitor (HMON) and Service Monitor (SMON) should run on a host that is separate from Cloudera Manager. For medium and larger clusters, the SQL database should not be shared between Cloudera Manager and CDH component services. Host Monitor and Service Monitor do not use SQL database. They use an on-disk LevelDB database in the /var partition.

Note: To increase the Cloudera Manager Server heap size you must update the parameter under

/etc/default/cloudera-scm-server export CMF_JAVA_OPTS="-Xmx4G. Later restart the Cloudera Manager server for the changes to take effect.

Service Monitor Requirements

The requirements for the Service Monitor are based on the number of monitored entities. To see the number of monitored entities, perform the following steps:

1. Open the Cloudera Manager Admin Console and click Clusters Cloudera Management Service .
2. Find the Cloudera Management Service Monitored Entities chart. If the chart does not exist, add it from the Chart Library.

For more information about Cloudera Manager entities, see *Cloudera Manager Entity Types*.

Note: Java Heap Size values (see the tables below) are rough estimates and some tuning might be necessary. From Cloudera Manager, Cloudera recommends using G1 garbage collector (G1GC) for Service Monitor. G1GC eliminates long JVM pauses, but uses a bit more CPU and RAM. It is the default for new installations. See Tuning JVM Garbage Collection.

Important: Service Monitor is not supported when installed on the BTRFS filesystem.

Table 6: Clusters with HDFS, YARN, or Impala

Use the recommendations in this table for clusters where the only services with worker roles are HDFS, YARN, or Impala.

Number of Monitored Entities	Number of Hosts	Required Java Heap Size	Recommended Non-Java Heap Size
0-2,000	0-100	1 GB	6 GB
2,000-4,000	100-200	1.5 GB	6 GB
4,000-8,000	200-400	1.5 GB	12 GB
8,000-16,000	400-800	2.5 GB	12 GB
16,000-20,000	800-1,000	3.5 GB	12 GB

Table 7: Clusters with HBase, Solr, Kafka, or Kudu

Use these recommendations when services such as HBase, Solr, Kafka, or Kudu are deployed in the cluster. These services typically have larger quantities of monitored entities.

Number of Monitored Entities	Number of Hosts	Required Java Heap Size	Recommended Non-Java Heap Size
0-30,000	0-100	2 GB	12 GB
30,000-60,000	100-200	3 GB	12 GB
60,000-120,000	200-400	3.5 GB	12 GB
120,000-240,000	400-800	8 GB	20 GB

Related Information

Host Monitor and Service Monitor Memory Configuration

Host Monitor

The requirements for the Host Monitor are based on the number of monitored entities.

To see the number of monitored entities, perform the following steps:

1. Open the Cloudera Manager Admin Console and click Clusters Cloudera Management Service .
2. Find the Cloudera Management Service Monitored Entities chart. If the chart does not exist, add it from the Chart Library.

For more information about Cloudera Manager entities, see *Cloudera Manager Entity Types*.

Important: Host Monitor is not supported when installed on the BTRFS filesystem.

Number of Hosts	Number of Monitored Entities	Heap Size	Non-Java Heap Size
0-200	<6k	1 GB	2 GB
200-800	6k-24k	2 GB	6 GB
800-1000	24k-30k	3 GB	6 GB

Ensure that you have at least 25 GB of disk space available for the Host Monitor, Service Monitor, Reports Manager, and Events Server databases.

Related Information

Cloudera Manager Entity Types

Host Monitor and Service Monitor Memory Configuration

Reports Manager

The Reports Manager fetches the fsimage from the NameNode at regular intervals. It reads the fsimage and creates a Lucene index for it. To improve the indexing performance, Cloudera recommends provisioning a host as powerful as possible and dedicating an SSD disk to the Reports Manager.

Table 8: Reports Manager

Component	Java Heap	CPU	Disk
Reports Manager	`The recommended value is (4 * FsImage size + 2 GB). For Example: For a 20 GB FsImage size; this means 4 * 20 GB + 2 GB = 82 GB.` **Important:** Calculating directory size with snapshots included substantially increases the required Java heap size. If this feature is enabled using the Enable snapshot processing configuration parameter, check the JVM Heap Memory Usage chart for Reports Manager and increase the heap size until it no longer shows consistently high usage. Up to four times the above recommended value may be necessary.	• Minimum: 8 cores • Recommended: 16 cores (32 cores, with hyperthreading enabled.)	1 dedicated disk that is at least 20 times the size of the fsimage. Cloudera strongly recommends using SSD disks.

Agent Hosts

An unpacked parcel requires approximately three times the space of the packed parcel that is stored on the Cloudera Manager Server.

Component	Storage	Notes
Partition hosting /opt	100 GB minimum	Usage grows as new parcels are downloaded to cluster hosts.
/var/log	2 GB per role	Each role running on the host will need at least 2 GB of disk space.

Event Server

The following table lists the minimum requirements for the Event Server:

CPU	RAM	Storage
1 core	256 MB	• 5 GB for the Event Database • 20 GB for the Event Server Index Directory. The location of this directory is set by the Event Server Index Directory Event Server configuration property.

Alert Publisher

The following table lists the minimum requirements for the Alert Publisher:

CPU	RAM	Storage
1 core	1 GB	Minimum of 1 disk for log files

Cloudera Runtime

Hardware requirements for Cloudera Runtime components.

Atlas

You must verify the supported hardware components for Atlas.

Memory	CPU	Disk	Additional Dependencies
Small: 4 GB Large: 32 GB	Minimum: 4 Medium: 8 Large: 16	No special requirement because HBase is used for storage.	Solr Shards: 4 (property: atlas_solr_shards) The shards for Atlas collections within Solr is determined by this number.

HDFS heap sizing

You can provision an HDFS cluster for optimal performance based on the desired storage capacity.

Component	Memory	CPU	Disk
JournalNode	1 GB (default) Set this value using the Java Heap Size of JournalNode in Bytes HDFS configuration property.	1 core minimum	1 dedicated disk
NameNode	• Minimum: 1 GB (for proof-of-concept deployments) • Add an additional 1 GB for each additional 1,000,000 blocks Snapshots and encryption can increase the required heap memory. See *Sizing NameNode Heap Memory*. Set this value using the Java Heap Size of NameNode in Bytes HDFS configuration property.	Minimum of 4 dedicated cores; more may be required for larger clusters	• Minimum of 2 dedicated disks for metadata • 1 dedicated disk for log files (This disk may be shared with the operating system.) • Maximum disks: 4
DataNode	Minimum: 4 GB Maximum: 8 GB Increase the memory for higher replica counts or a higher number of blocks per DataNode. When increasing the memory, Cloudera recommends an additional 1 GB of memory for every 1 million replicas above 4 million on the DataNodes. For example, 5 million replicas require 5 GB of memory. Set this value using the Java Heap Size of DataNode in Bytes HDFS configuration property.	Minimum: 4 cores. Add more cores for highly active clusters.	Minimum: 4 Maximum: 24 The maximum acceptable size will vary depending upon how large average block size is. The DN's scalability limits are mostly a function of the number of replicas per DN, not the overall number of bytes stored. That said, having ultra-dense DNs will affect recovery times in the event of machine or rack failure. Cloudera does not support exceeding 100 TB per data node. You could use 12 x 8 TB spindles or 24 x 4TB spindles. Cloudera does not support drives larger than 8 TB.

Warning: Running Runtime on storage platforms other than direct-attached physical disks can provide suboptimal performance. Cloudera Enterprise and the majority of the Hadoop platform are optimized to provide high performance by distributing work across a cluster that can utilize data locality and fast local I/O.

HBase

Know the hardware resource requirements for HBase server in CDP Private Cloud Base.

Component	Java Heap	CPU	Disk
Master	• 100-10,000 regions: 4 GB • 10,000 or more regions with 200 or more Region Servers: 8 GB • 10,000 or more regions with 300 or more Region Servers: 12 GB Set this value using the Java Heap Size of HBase Master in Bytes HBase configuration property.	Minimum 4 dedicated cores. You can add more cores for larger clusters, when using replication, or for bulk loads.	1 disk for local logs, which can be shared with the operating system and/or other Hadoop logs
Region Server	• Minimum: 8 GB • Medium-scale production: 16 GB • Heap larger than 16 GB requires special Garbage Collection tuning. See *Configuring the HBase BlockCache*. Set this value using the Java Heap Size of HBase RegionServer in Bytes HBase configuration property.	Minimum: 4 dedicated cores	• 4 or more spindles for each HDFS DataNode • 1 disk for local logs (this disk can be shared with the operating system and/or other Hadoop logs
Thrift Server	1 GB - 4 GB Set this value using the Java Heap Size of HBase Thrift Server in Bytes HBase configuration property.	Minimum 2 dedicated cores.	1 disk for local logs, which can be shared with the operating system and other Hadoop logs.

Note: Consider adding more HBase Thrift Servers for production environments and deployments with a large number of Thrift client to scale horizontally.

Related Information

Configuring HBase BlockCache

Hive

<table>
<tr><th>Component</th><th colspan="2">Java Heap</th><th>CPU</th><th>Disk</th></tr>
<tr><td rowspan="7">HiveServer 2</td><td>Single Connection</td><td>4 GB</td><td rowspan="7">Minimum 4 dedicated cores</td><td rowspan="7">Minimum 1 disk
This disk is required for the following:
• HiveServer2 log files
• stdout and stderr output files
• Configuration files
• Operation logs stored in the operation_lo gs_dir directory, which is configurable
• Any temporary files that might be created by local map tasks under the /tmp directory</td></tr>
<tr><td>2-10 connections</td><td>4-6 GB</td></tr>
<tr><td>11-20 connections</td><td>6-12 GB</td></tr>
<tr><td>21-40 connections</td><td>12-16 GB</td></tr>
<tr><td>41 to 80 connections</td><td>16-24 GB</td></tr>
<tr><td colspan="2">Cloudera recommends splitting HiveServer2 into multiple instances and load balancing them once you start allocating more than 16 GB to HiveServer2. The objective is to adjust the size to reduce the impact of Java garbage collection on active processing by the service.</td></tr>
<tr><td colspan="2">Set this value using the Java Heap Size of HiveServer2 in Bytes Hive configuration property.</td></tr>
<tr><td rowspan="6">Hive Metastore</td><td>Single Connection</td><td>4 GB</td><td rowspan="6">Minimum 4 dedicated cores</td><td rowspan="6">Minimum 1 disk
This disk is required so that the Hive metastore can store the following artifacts:
• Logs
• Configuration files
• Backend database that is used to store metadata if the database server is also hosted on the same node</td></tr>
<tr><td>2-10 connections</td><td>4-10 GB</td></tr>
<tr><td>11-20 connections</td><td>10-12 GB</td></tr>
<tr><td>21-40 connections</td><td>12-16 GB</td></tr>
<tr><td>41 to 80 connections</td><td>16-24 GB</td></tr>
<tr><td colspan="2">Set this value using the Java Heap Size of Hive Metastore Server in Bytes Hive configuration property.</td></tr>
<tr><td>Beeline CLI</td><td colspan="2">Minimum: 2 GB</td><td>N/A</td><td>N/A</td></tr>
</table>

Hue

Review the memory, CPU, and storage requirements needed for running a Hue server in CDP Private Cloud Base.

Component	Memory	CPU	Disk
Hue Server	• Minimum: 8 GB • Maximum 16 GB • If the cluster uses the Hue load balancer, add additional memory	Minimum: 1 Core to run Django When Hue is configured for high availability, add additional cores	Minimum: 32 GB for the database (16 GB) and /tmp (temporary) directory (16 GB). Database grows proportionally according to the cluster size and workloads. Add additional space when configuring Hue in High Availability.

The term *cluster size* refers to the number of nodes in the cluster. *Workload* in Hue means the number of queries run and the number of concurrent unique users using the application in a given period of time.

A minimum of 16 GB is needed for the database. The Hive MetaStore and Hue Query Processor services largely use the database. The database grows in size quickly because of the query history that it retains. To optimize performance, you must regluarly cleanup old documents and queries.

Note: Hue is limited by cgroup settings. In Cloudera Manager, all memory soft/hard limits are set to -1.

Related Information

Adding a Load Balancer for Hue

Impala

Sizing requirements for Impala can vary significantly depending on the size and types of workloads using Impala.

<table>
<tr><th>Component</th><th>Native Memory</th><th>JVM Heap</th><th>CPU</th><th>Disk</th></tr>
<tr><td>Impala Daemon</td><td>Set this value using the Impala Daemon Memory Limit configuration property.
• Minimum: 32 GB
• Recommended: 128 GB</td><td>Set this value using the Java Heap Size of Impala Daemon in Bytes configuration property for the Coordinator Impala Daemons.
• Minimum: 4 GB
• Recommended: 8 GB</td><td>• Minimum: 4
• Recommended: 16 or more
CPU instruction set: AVX2</td><td>• Minimum: 1 disk
• Recommended: 8 or more</td></tr>
<tr><td>Catalog Server</td><td colspan="2">Set this value using the Java Heap Size of Catalog Server in Bytes configuration property.
• Minimum: 4 GB
• Recommended: 8 GB</td><td>• Minimum: 4
• Recommended: 16 or more
CPU instruction set: AVX2</td><td>• Minimum and Recommended: 1 disk</td></tr>
</table>

For the networking topology for multi-rack cluster, Leaf-Spine is recommended for the optimal performance.

Kafka

Kafka requires a fairly small amount of resources, especially with some configuration tuning. By default, Kafka, can run on as little as 1 core and 1GB memory with storage scaled based on requirements for data retention.

CPU is rarely a bottleneck because Kafka is I/O heavy, but a moderately-sized CPU with enough threads is still important to handle concurrent connections and background tasks.

Kafka brokers tend to have a similar hardware profile to HDFS data nodes. How you build them depends on what is important for your Kafka use cases.

Use the following guidelines:

To affect performance of these features:	Adjust these parameters:
Message Retention	Disk size
Client Throughput (Producer & Consumer)	Network capacity
Producer throughput	Disk I/O
Consumer throughput	Memory

A common choice for a Kafka node is as follows:

Component	Memory/Java Heap	CPU	Disk
Broker	• RAM: 64 GB • Recommended Java heap: 4 GB Set this value using the Java Heap Size of Broker Kafka configuration property. See Other Kafka Broker Properties table.	12- 24 cores	• 1 HDD For operating system • 1 HDD for Zookeeper dataLogDir • 10- HDDs, using Raid 10, for Kafka data

Component	Memory/Java Heap	CPU	Disk
KRaft Controller	• RAM: 4 GB • Recommended Java heap: 4 GB Set these values using the Additional KRaft Controller Java Options Kafka property.	1 dedicated core	A single 64 GB SSD
Cruise Control	1 GB	1 core **Note:** A moderately-sized CPU with enough threads is important to handle metric fetching from Kafka and background tasks.	Because Cruise Control stores its data in Kafka the storage requirements will depend on the retention settings of the related Kafka topics.
Kafka Connect	0.5 - 4 GB heap size depending on the Connectors in use.	4 cores **Note:** Depends on the Connectors in use.	
MirrorMaker	1 GB heap Set this value using the Java Heap Size of MirrorMaker Kafka configuration property.	1 core per 3-4 streams	No disk space needed on MirrorMaker instance. Destination brokers should have sufficient disk space to store the topics being copied over.
Schema Registry	1 GB heap	2 cores	1 MB Serialization JAR files may be uploaded and may be of any size. The disk usage depends on the JAR files uploaded. The files may be stored locally on the same host where SchemaRegistry is running or in HDFS if available.
Streams Messaging Manager **Note:** The hardware requirements for SMM depends on the number of Kafka partitions.	8 GB heap	8 cores	5 GB
Streams Replication Manager	• 1 GB heap for SRM driver • 1 GB heap for SRM Service	The performance of the SRM driver is mostly impacted by network throughput and latency.	No resources required

Networking requirements: Gigabit Ethernet or 10 Gigabit Ethernet. Avoid clusters that span multiple data centers.

Kafka and Zookeeper: It is common to run ZooKeeper on 3 broker nodes that are dedicated for Kafka. However, for optimal performance Cloudera recommends the usage of dedicated Zookeeper hosts. This is especially true for larger, production environments.

Key Trustee Server

Component	Memory	CPU	Disk
Key Trustee Server **Note:** KTS requires a additional dedicated resources.	8 GB	1 GHz 64-bit quad core	20 GB, using moderate to high-performance drives

Related Information

Encrypting Data at Rest

Ranger KMS

Component	Memory	CPU	Disk
Ranger KMS **Note:** Cloudera recommends using machines with CPUs that support the AES-NI instruction set and have a similar performance to the CPUs available to the NameNodes, so as not to introduce a bottleneck to HDFS client operations.	8 GB	1 GHz 64-bit quad core	20 GB, using moderate to high-performance drives

Kudu

Understand the resource requirements for Kudu before making the resource configuration changes in Cloudera Manager.

Component	Memory	CPU	Disk
Tablet Server	• Minimum: 4 GB • Recommended: 10 GB Additional hardware may be required, depending on the workloads running in the cluster.	Kudu currently requires a CPU that supports the SSSE3 and SSE4.2 instruction sets. If you are to run Kudu inside a VM, enable SSE4.2 pass-through to pass through SSE4.2 support into the VM.	1 disk for write-ahead log (WAL). Using an SSD drive may improve performance.
Master	• Minimum: 256 MB • Recommended: 1 GB		1 disk

Related Information

Apache Kudu configuration

Oozie

Understand the resource requirements for Oozie before making the resource configuration changes in Cloudera Manager.

Component	Java Heap	CPU	Disk
Oozie	• Minimum: 1 GB (this is the default set by Cloudera Manager). This is sufficient for less than 10 simultaneous workflows, without forking. • If you notice excessive garbage collection, or out-of-memory errors, increase the heap size to 4 GB for medium-size production clusters or to 8 GB for large-size production clusters. • Set this value using the Java Heap Size of Oozie Server in Bytes Oozie configuration property.	No resources required	No resources required

Additional tuning:

For workloads with many coordinators that run with complex workflows (a max concurrency reached! warning appears in the log and the Oozie admin -queuedump command shows a large queue):

- Increase the value of the oozie.service.CallableQueueService.callable.concurrency property to 50.
- Increase the value of the oozie.service.CallableQueueService.threads property to 200.

Do not use a Derby database as a backend database for Oozie.

Ozone hardware recommendations

This guide helps you choose hardware for Ozone based on your data storage needs. Following these recommendations will ensure that you get optimum performance from your Ozone cluster.

Table 9: Recommendations

The values are minimums and can go higher depending on your requirements. However, the above recommendations will apply to the vast majority of deployments.

<table>
<tr><th>Node Type</th><th>Chasis</th><th>CPU</th><th>Node RAM</th><th>RAM for each service</th><th>OS Disk</th><th>Meta Disk (NVMe)</th><th>Data Disk</th><th>Network</th><th>Disk Controllers</th><th>GPU</th></tr>
<tr><td>Master node (OM and SCM and Recon)</td><td>1U</td><td>2 x 20c</td><td rowspan="2">256 GB</td><td>64 GB</td><td rowspan="4">2 x 480 GB SSD</td><td>2 x 4 TB</td><td>-</td><td rowspan="3">2x 25Gbps</td><td>-</td><td rowspan="2">-</td></tr>
<tr><td>Datanode (Ozone, no compute)</td><td rowspan="2">2U</td><td>2 x 12c</td><td rowspan="2">31 GB**</td><td>2 x 1.5 TB</td><td rowspan="2">24 x 16TB</td><td rowspan="2">2x 12 Gbps (low)</td></tr>
<tr><td>Datanode (Ozone, mixed compute)</td><td rowspan="2">2 x 24c</td><td rowspan="2">512 GB</td><td>2 x 3 TB</td><td rowspan="2">Optional</td></tr>
<tr><td>Compute node (No Storage)</td><td>1U</td><td>-</td><td>1 x 4 TB</td><td>-</td><td>1x 25Gbps</td><td>-</td></tr>
</table>

**Avoid using heap sizes of 32GB to 47GB because the JVM cannot use Compressed oops for heap sizes > 31GB. This reduces the effective memory available to the process. If you want to configure heaps > 31GB, then use a heap size of at least 48GB or higher.

Notes

The above configuration will support up to 10B keys because of the 4 TB NVMe on the master nodes.

The absolute minimum recommended configuration is 3 master nodes and 9 datanodes. This will support Erasure Coding with the RS(6,3) configuration with full High Availability. Additional datanodes can be added in increments of 1 to increase storage.

Network

The network between the datanodes and the compute nodes cannot be oversubscribed by more than 2:1. Networking is sized to support the full (real-world) bandwidth of the drives across the network. More drives require faster networks, both at the server level and the switch level.

NVMe

NVMe should be configured in RAID1 pairs to provide business continuity for Ozone metadata in case of hardware failure.

The master nodes and datanodes use NVMe to store Ozone metadata. The compute nodes use NVMe for shuffle (Spark, MapReduce, and Tez) and caching (LLAP). The mixed compute datanodes use NVMe for both Ozone metadata and shuffle (Spark, MapReduce, and Tez) plus caching (LLAP).

Cloudera recommends mounting Ozone partitions across the NVMe drive pair as RAID1 (800GB) with the remaining space used for shuffle or cache as independent JBOD partitions. RAID can be configured either in hardware or in software.

Example sizing calculator

Suggested Value of Parameter	Logical Capacity 2PB	Logical Capacity 8PB	Logical Capacity 16PB	Additional information
Number of Data Nodes if using Erasure Coding rs(6,3)	9	31	64	These are calculated based on actual file storage required (See Row 1)
Logical data size proposed (TB, EC 6,3)	2304	8192	16384	-
Raw disk capacity (TB)	3456	12288	24576	-
Number of Data Nodes if using triple replication	16	64	128	These are calculated based on actual file storage required (See Row 1)
Logical data size - conservative using 3x (TB)	2048	8192	16384	-
Raw disk capacity (TB)	6144	24576	49152	-

Related Information

Ozone Architecture

Phoenix

Know the hardware resource requirements for Phoenix Query Server (PQS) in CDP Private Cloud Base.

Component	Java Heap	CPU	Disk
Phoenix Query Server	1 GB - 4 GB Set this value using the Phoenix Query Server Max Heapsize configuration property. Increase this property value if you run any of these queries, aggregates, joins, or subqueries and if the query processing requires more memory.	Minimum 2 dedicated cores.	1 disk for local logs, which can be shared with the operating system and other Hadoop logs.

Ranger

Memory	CPU	Disk	Additional Dependencies
Ranger Admin: 1 GB minimum, then adjust heap as required (8 GB-16 GB)	1 core minimum	No special requirement.	
Ranger Usersync: 1 GB minimum	1 core minimum	No special requirement.	
Ranger Tagsync: 1 GB minimum	1 core minimum	No special requirement.	

Solr

Component	Java Heap	CPU	Disk
Solr	• Small workloads, or evaluations: 16 GB • Smaller production environments: 32 GB • Larger production environments: 96 GB is sufficient for most clusters. Set this value using the Java Heap Size of Solr Server in Bytes Solr configuration property.	• Minimum: 4 • Recommended: 16 for production workloads	No requirement. Solr uses HDFS for storage.

Note the following considerations for determining the optimal amount of heap memory:

- Size of searchable material: The more searchable material you have, the more memory you need. All things being equal, 10 TB of searchable data requires more memory than 1 TB of searchable data.
- Content indexed in the searchable material: Indexing all fields in a collection of logs, email messages, or Wikipedia entries requires more memory than indexing only the Date Created field.
- The level of performance required: If the system must be stable and respond quickly, more memory may help. If slow responses are acceptable, you may be able to use less memory.

Related Information

Deployment Planning for Cloudera Search

Spark

Component	Java Heap	CPU	Disk
Spark History Server	Minimum: 512 MB Set this value using the Java Heap Size of History Server in Bytes Spark configuration property.	1 **Important:** Cloudera recommends that you adjust the number of CPUs and memory for the Spark History Server based on your specific cluster usage patterns.	Minimum 1 disk for log files.

Livy

Understand the resource requirements for Livy before making the resource configuration changes in Cloudera Manager.

Component	Java Heap	CPU	Disk
Livy	Minimum: 512 MB Set this value using the Maximum size for the Java process heap memory Livy configuration property.	Minimum 1 core	Minimum 1 disk

Zeppelin

Understand the resource requirements for Zeppelin before making the resource configuration changes in Cloudera Manager.

Following are the resource requirements for Zeppelin to run together with the other components. In general, Zeppelin mainly contains one `Zeppelin Server` and multiple `Zeppelin Interpreters`. The interpreters may range from 2 to 6. By default, the four supported interpreters in CDP are JDBC, Livy, Markdown, and Angular.

Component	Java Heap	CPU	Memory
Zeppelin	Minimum: 1 GB Set the heap size using the Zeppelin Server Advanced Configuration Snippet (Safety Valve) for the `zeppelin-env.sh` property in Cloudera Manager Zeppelin configuration page by setting the ZEPPELIN_MEM and ZEPPELIN_INTP_MEM environment variables. For example, export ZEPPELIN_ MEM = -Xms1024m -Xmx 2048m export ZEPPELIN_INTP_MEM= -Xms1024m -Xmx 2048m	The Zeppelin Server and Zeppelin Interpreter processes are not restricted in the container. So, the CPU requirement for the Zeppelin process is not easy to define in CDP. But the physical nodes that Zeppelin runs on should at least have 8 physical cores because Zeppelin may spin up hundreds of threads when busy.	**For Zeppelin Server:** Minimum memory: 1 GB Recommended minimum memory: 2 GB Maximum Memory: 16 GB Recommended maximum memory: 8 GB **For Zeppelin Interpreter:** Minimum memory: 1 GB Recommended minimum memory: 2 GB Maximum Memory: 16 GB Recommended maximum memory: 4 GB **Note:** By default, the running Zeppelin Server and all Interpreters use at least 1 + 4 = 5 GB memory in total. Cloudera recommends 64 GB for the physical node that Zeppelin runs on.

Note: You can change the resource configuration using Cloudera Manager based on your needs.

YARN

Component	Java Heap	CPU	Other Recommendations
Job History Server	• Minimum: 1 GB • Increase memory by 1.6 GB for each 100,000 tasks kept in memory. For example: 5 jobs @ 100, 000 mappers + 20,000 reducers = 600,000 total tasks requiring 9.6 GB of heap. See the Other Recommendations column for additional tuning suggestions. Set this value using the Java Heap Size of JobHistory Server in Bytes YARN configuration property.	Minimum: 1 core	• • Set the mapreduce.jobhis tory.loadedtasks.cache.size property to a total loaded task count. Using the example in the Java Heap column to the left, of 650,000 total tasks, you can set it to 700,000 to allow for some safety margin. This should also prevent the JobHistoryServer from hanging during garbage collection, since the job count limit does not have a task limit.
NodeManager	Minimum: 1 GB. Configure additional heap memory for the following conditions: • Large number of containers • Large shxmluffle sizes in Spark or MapReduce Set this value using the Java Heap Size of NodeManager in Bytes YARN configuration property.	• Minimum: 8-16 cores • Recommended: 32-64 cores	Disks: • Minimum: 8 disks • Recommended: 12 or more disks Networking: • Minimum: Dual 1Gbps or faster • Recommended: Single/Dual 10 Gbps or faster

Component	Java Heap	CPU	Other Recommendations
ResourceManager	Minimum: 1 GB Configure additional heap memory for the following conditions: • More jobs • Larger cluster size • Number of retained finished applications (configured with the yarn.resourcemanager .max-completed-applications property. • Scheduler configuration Set this value using the Java Heap Size of ResourceManager in Bytes YARN configuration property.	Minimum: 1 core	
Other Settings	• Set the ApplicationMaster Memory YARN configuration property to 512 MB • Set the Container Memory Minimum YARN configuration property to 1 GB.	N/A	N/A

Related Information

Tuning Apache Hadoop YARN

ZooKeeper

Understand the resource requirements for ZooKeeper before making the resource configuration changes in Cloudera Manager.

Component	Java Heap	CPU	Disk
ZooKeeper Server	• Minimum: 1 GB • Increase heap size when watching 10,000 - 100,000 ephemeral znodes and are using 1,000 or more clients. Set this value using the Java Heap Size of ZooKeeper Server in Bytes ZooKeeper configuration property.	Minimum: 4 cores	You can use HDDs or SSDs as per your requirement.

Related Information

Add a ZooKeeper service

Operating System Requirements

This topic describes the operating system requirements for Cloudera Private Cloud Base. Azul OpenJDK, OpenJDK 8, OpenJDK 11, and OpenJDK 17 are TCK certified for CDP.

Cloudera Private Cloud Base Supported Operating Systems

Please see the Cloudera Support Matrix for detailed information about supported operating systems.

Operating System support for the Cloudera Private Cloud Base Trial Installer

SLES 15 SP4 is supported when using the Trial Installer (cloudera-manager-installer.bin) to install Cloudera Manager.

Important: Extra step required when using Cloudera Manager Trial installer on SLES 15 SP4.

When using cloudera-manager-installer.bin to install a trial version of Cloudera Manager, the installation will fail.

Before running cloudera-manager-installer.bin, run the following command:

```
SUSEConnect --list-extensions
SUSEConnect -p sle-module-legacy/15.4/x86_64
zypper install libncurses5
```

Important information about Runtime and Cloudera Manager Supported Operating Systems

Runtime provides parcels for select versions of RHEL-compatible operating systems.

Important:

In order to be covered by Cloudera Support:

- All Runtime hosts in a logical cluster must run on the same major OS release.
- Cloudera supports a temporarily mixed OS configuration during an OS upgrade project.
- Cloudera Manager must run on the same OS release as one of the clusters it manages.

Cloudera recommends running the same minor release on all cluster nodes. However, the risk caused by running different minor OS releases is considered lower than the risk of running different major OS releases.

Points to note:

- Cloudera does not support Runtime cluster deployments in Docker containers.
- Cloudera Enterprise is supported on platforms with Security-Enhanced Linux (SELinux) enabled and in enforcing mode. Cloudera is not responsible for policy support or policy enforcement. If you experience issues with SELinux, contact your OS provider.
- An operating system that has Control Group v2 enabled by default (all supported OS versions) must be manually set to use Control Group v1. For more information, see TSB 2024-757: Deployments using cgroups v2 on RHEL9 will fail to initialize.

Important:

- NavEncrypt and KTS are not supported in 7.1.8 and 7.1.9 when using SLES 15 SP4
- Cloudera Manager 7.11.3 supports only SLES 15 SP4 but not SLES 12. So it is not possible to have temporarily mixed OS configurations during the upgrade.

CDP Private Cloud Base supported operating systems

Operating System	Version
IBM PowerPC on RHEL	The following components are not supported: • Impala • Kudu • Ozone • Navigator Encrypt **Note:** Ranger KMS is the recommended Key Management Server for PowerPC deployments.

Operating System and IBM PowerPC support matrix

This matrix explains the operating system supported on IBM PowerPC. There are two core configurations with CDP Private Cloud Base and different PowerPC version deployments:

1. IBM PowerPC only and CDP Private Cloud Base
2. IBM PowerPC CPU, IBM Spectrum Scale Storage, and CDP Private Cloud Base. This is a subset of what is supported generally on IBM PowerPC.

IBM PowerPC Support	Documentation
PowerPC 8 and 9 generally without Spectrum Scale Storage	https://www.ibm.com/docs/en/linux-on-systems?topic=lpo-supported-linux-distributions-virtualization-options-power8-power9-linux-power-systems
PowerPC 10 generally without Spectrum Scale Storage	https://www.ibm.com/docs/en/linux-on-systems?topic=lpo-supported-linux-distributions-virtualization-options-power10-linux-power-servers
IBM Spectrum Scale Storage with CDP Private Cloud Base on x86 and PowerPC combinations	https://www.ibm.com/docs/en/spectrum-scale-bda?topic=requirements-support-matrix

Software Dependencies

- Python - Python dependencies for the different CDP components is mentioned below:
  Cloudera Manager

  (For 7.1.9 SP1) You must install Python 3.8 (or 3.9 for RHEL 9.1) on all hosts before upgrading to Cloudera Manager 7.11.3.

  (For 7.1.9 SP1) You must install Python 3.10 for SLES 15 SP4 and SLES 15 SP5 on all hosts before upgrading to Cloudera Manager 7.11.3 CHF7. For more information, see Installing Python 3.

  Important:

  Cloudera Manager now requires Python 3.10 on all versions of SLES 15, including SLES 15 SP4. It is not possible to support two different versions of Python for the same major version of operating system. If the cluster was previously running Python 3.8, then you must upgrade to Python 3.10.

  Due to a change in support from Python 3.8 to Python 3.10 for SLES 15 SP4 and SLES 15 SP5, only a regular upgrade of Cloudera Manager to 7.11.3 CHF7 and CDP Runtime cluster to 7.1.9 SP1 is possible and must occur sequentially without starting the cluster between the Cloudera Manager and CDP Runtime cluster upgrades.

  Ubuntu 18 Operating System is not supported from Cloudera Manager 7.11.3 to Cloudera Manager 7.11.3 CHF7 versions. You must upgrade the Operating System from Ubuntu 18 to Ubuntu 20 before you upgrade to Cloudera Manager 7.11.3 CHF7. For performing major OS upgrade, see Upgrading the Operating System to a new Major Version.

  Hue

  (For 7.1.9) The minimum required version of Python 3.8 is 3.8.12. The minimum version of Python 3.9 (RHEL 9) is 3.9.14.

  (For 7.1.9 SP1) The recommended minimum version of Python 3.10 on SLES 15 SP4, SLES 15 SP5, and Ubuntu 22 is 3.10.14.

  Spark

  Important: Spark 2 will be deprecated in Cloudera Runtime 7.1.9. Therefore, 7.1.9 is the last runtime release where Spark 2 is supported. For more information, see Deprecation Notices in Cloudera Runtime.

  Spark 2.4 supports Python 2.7 and 3.4-3.7.

  Spark 3.0 supports Python 2.7 and 3.4 and higher, although support for Python 2 and 3.4 to 3.5 is deprecated.

  Spark 3.1 supports Python 3.6 and higher.

  If the right level of Python is not picked up by default, set the PYSPARK_PYTHON and PYSPARK_DRIVER_PYTHON environment variables to point to the correct Python executable before running the `pyspark` command.
- Perl - Cloudera Manager requires perl.
- python-psycopg2 - Cloudera Manager 7 has a dependency on the package python-psycopg2. PostgreSQL-backed Hue in Runtime 7 requires a higher version of psycopg2 than is required by the Cloudera Manager dependency. For more information, see *Installing the psycopg2 Python Package*.
- iproute package - Cloudera Private Cloud Base has a dependency on the iproute package. Any host that runs the Cloudera Manager Agent requires the package. The required version varies depending on the operating system:

Table 10: iproute package

Operating System	iproute version
RHEL	iproute

Operating System	iproute version
Ubuntu	iproute2
SLES	iproute2

- **rpcbind package** - CDP Private Cloud Base has a dependency on the `rpcinfo` command which is usually found in the rpcbind package. Any host that runs the Cloudera Manager Agent requires this package. The required version varies depending on the operating system.

Filesystem Requirements

Supported Filesystems

The Hadoop Distributed File System (HDFS) is designed to run on top of an underlying filesystem in an operating system. Cloudera recommends that you use either of the following filesystems tested on the supported operating systems:

- ext3: This is the most tested underlying filesystem for HDFS.
- ext4: This scalable extension of ext3 is supported in more recent Linux releases.

> **Important:** Cloudera does not support in-place upgrades from ext3 to ext4. Cloudera recommends that you format disks as ext4 before using them as data directories.

- XFS: This is the default filesystem in RHEL 7.
- S3: Amazon Simple Storage Service

Kudu Filesystem Requirements - Kudu is supported on ext4 and XFS. Kudu requires a kernel version and filesystem that supports hole punching. Hole punching is the use of the fallocate(2) system call with the FALLOC_FL_PUNCH_HOLE option set.

File Access Time

Linux filesystems keep metadata that record when each file was accessed. This means that even reads result in a write to the disk. To speed up file reads, Cloudera recommends that you disable this option, called atime, using the noatime mount option in /etc/fstab:

```
/dev/sdb1 /data1 ext4 defaults,noatime 0
```

Apply the change without rebooting:

```
mount -o remount /data1
```

Filesystem Mount Options

The filesystem mount options have a sync option that allows you to write synchronously.

Using the sync filesystem mount option reduces performance for services that write data to disks, such as HDFS, YARN, Kafka and Kudu. In CDP, most writes are already replicated. Therefore, synchronous writes to disk are unnecessary, expensive, and do not measurably improve stability.

NFS and NAS options are not supported for use as DataNode Data Directory mounts, even when using Hierarchical Storage features.

Cloudera supports mounting /tmp with the noexec option. Mounting /tmp as a filesystem with the noexec option is sometimes done as an enhanced security measure to prevent the execution of files stored there.

Filesystem Requirements

You can control resource allocation for Cloudera Manager and CDP Runtime services (nproc, nofile, etc) from /etc/security/limits.conf, and through `init` scripts on traditional SysV Init systems. However, on systems using systemd the limits either needs to be set in the service's unit file, or in /etc/systemd/system.conf, or in files present under /etc/systemd/system.conf.d/*. This is due to a known limitation with systemd as it does not use PAM login sessions (pam_limits.so) for daemon services, thereby ignoring the limits defined in /etc/security/limits.conf. Both Cloudera Manager Agent and Supervisord (responsible for starting CDP Runtime services) are daemonised during system initialisation.

You can perform either of the following steps to modify the resource limit:

1. For system-wide change, uncomment the process properties from /etc/systemd/system.conf, or create an override .conf under /etc/systemd/system.conf.d/. This requires a **nix* system reboot for the changes to take effect. For more information, see Limits.conf.
2. To apply custom limits on CDP Runtime services, add the required process properties to the `[Service]` section in /usr/lib/systemd/system/cloudera-scm-supervisord.service.

   For instance, to customise the number of child processes a process can fork. You can set the property as follows:

   ```
   LimitNPROC=<value>
   ```

   Then reload the configuration by running the following command for the limits to be applied in the subsequent service restarts:

   ```
   # systemctl daemon-reload
   ```

   Here are the list of available process properties.

nscd for Kudu

Although not a strict requirement, it's highly recommended that you use nscd to cache both DNS name resolution and static name resolution for Kudu.

Configuring system level operating system

Cloudera recommends you to set up the following configurations:

- Disabling Transparent Hugepages (THP)
- vm.swappiness Linux Kernel Parameter

For setting these configurations, see Disabling Transparent Hugepages (THP) and Setting the vm.swappiness Linux Kernel Parameter.

Related Information

Installing the psycopg2 Python package for PostgreSQL-backed Hue

Database Requirements

This topic describes the database requirements for Cloudera Private Cloud Base.

See Cloudera Support Matrix for detailed information about supported databases based on the CDP and Cloudera Manager versions.

Important: When you restart processes, the configuration for each of the services is redeployed using information saved in the Cloudera Manager database. If this information is not available, your cluster cannot start or function correctly. You must schedule and maintain regular backups of the Cloudera Manager database to recover the cluster in the event of the loss of this database. For more information, see *Backing Up Databases*.

Cloudera Manager and Runtime come packaged with an embedded PostgreSQL database for use in non-production environments. The embedded PostgreSQL database is not supported in production environments. For production environments, you must configure your cluster to use dedicated external databases. You must ensure latency between Cloudera Manager server and the database is < 10 ms. You can verify the latency with a simple SQL command from your Cloudera Manager server host to the database. Start your database's command line client tool and connect to the Cloudera Manager database. Run the following SQL command:

```
SELECT 1;
```

Important: Migrating from one external database server to a different type of database server after installing Cloudera Manager and Runtime services is a complex process that requires modification of the schema and matching the data in the database tables to the new schema. Cloudera expects you to perform the migration using any off-the-shelf tool. If you require any assistance, Cloudera recommends that you must engage with Cloudera Professional Services team.

After installing a database, upgrade to the latest patch and apply appropriate updates. Available updates may be specific to the operating system on which it is installed.

Notes:

- Cloudera recommends installing the databases on different hosts than the services, located in the same data center. Separating databases from services can help isolate the potential impact from failure or resource contention in one or the other. It can also simplify management in organizations that have dedicated database administrators.
- Hue Query Processor in CDP 7.1.8 requires a non-SSL enabled PostgreSQL database.
- CDP does not support Percona for MySQL as a backend database for Hive Metastore (HMS).
- Use the appropriate UTF8 encoding for Metastore, Oozie, Hive, and Hue.

  Oozie also supports UTF8MB4 character encoding out of box without any configuration change when the Oozie custom database is created with the encoding of UTF8MB4.

  MySQL and MariaDB must use the MySQL utf8 encoding, not utf8mb4.
- Ranger only supports the InnoDB engine for MySQL and MariaDB databases.
- YARN Queue Manager requires a Postgres database server with a dedicated database. This is a temporary requirement.
- For MySQL 5.7, you must install the MySQL-shared-compat or MySQL-shared package. This is required for the Cloudera Manager Agent installation.
- MySQL GTID-based replication is not supported.
- Both the Community and Enterprise versions of MySQL are supported, as well as MySQL configured by the AWS RDS service.
- Before upgrading from CDH 5 to CDH 6, check the value of the COMPATIBLE initialization parameter in the Oracle Database using the following SQL query:

  ```
  SELECT name, value FROM v$parameter WHERE name = 'compatible'
  ```

  The default value is 12.2.0. If the parameter has a different value, you can set it to the default as shown in the Oracle Database Upgrade Guide.

  Note: Before resetting the COMPATIBLE initialization parameter to its default value, make sure you consider the effects of this change can have on your system.
- If you are using PostgreSQL High Availability (HA), then add the following property to the /etc/cloudera-scm-server/db.properties file in the Cloudera Manager server host:

  ```
  com.cloudera.cmf.orm.hibernate.c3p0.preferredTestQuery="select pg_curren
  t_wal_lsn()"
  ```

Related Information

Required Databases

RDBMS High Availability Support

Various Cloudera components rely on backing RDBMS services as critical infrastructure. You may require Cloudera components to support deployment in environments where RDBMS services are made highly-available. High availability (HA) solutions for RDBMS are implementation-specific, and can create constraints or behavioral changes in Cloudera components.

This section clarifies the support state and identifies known issues and limitations for HA deployments.

Upgrading Cloudera Manager and the Cloudera Manager database

When upgrading Cloudera Manager, there may be a minimum version requirement for the database server.

Ensure that the Cloudera Manager database server is upgraded to at least this minimum requirement prior to starting the new version of Cloudera Manager for the first time.

1. Stop Cloudera Manager Server service.
2. Upgrade the Cloudera Manager RPMs.
3. Upgrade the Cloudera Manager database server version.
4. Start the Cloudera Manager.

Upgrading CDP and the CDP Services databases

When upgrading CDP to a new version, the new version of CDP may have a minimum version requirement for the database server which holds services metadata.

If the database server needs to be upgraded, follow this proces:

1. Stop CDP services which depend on the database server, or alternatively, stop the entire cluster.
2. Upgrade the database server version.
3. Upgrade CDP using Cloudera Manager.
4. Start the stopped services, or the entire cluster.

   When a database server upgrade is required, it is not possible to perform a rolling upgrade of the cluster.

High Availability vs. Load Balancing

Understanding the difference between HA and load balancing is important for Cloudera components, which are designed to assume services are provided by a single RDBMS instance. Load balancing distributes operations across multiple RDBMS services in parallel, while HA focuses on service continuity. Load balanced deployments are often used as part of HA strategies to overcome demands of monitoring and failover management in an HA environment. While less easier to implement, load-balanced deployments require applications tailored to the behavior and limitations of the particular technology.

Support Statement: Cloudera components are not designed for and do not support load balanced deployments of any kind. Any HA strategy involving multiple active RDBMS services must ensure all connections are routed to a single RDBMS service at any given time, regardless of vendor or HA implementation/technology.

General High Availability Support

Cloudera supports various RDBMS options, each of which have multiple possible strategies to implement HA. Cloudera cannot reasonably test and certify on each strategy for each RDBMS. Cloudera expects HA solutions for RDBMS to be transparent to Cloudera software, and therefore are not supported and debugged by Cloudera. It is the responsibility of the customer to provision, configure, and manage the RDBMS HA deployment, so that Cloudera software behaves as it would when interfacing with a single, non-HA service. Cloudera will support and help customers troubleshoot issues when a cluster has HA enabled. While diagnosing database-related problems in Cloudera components, customers may be required to temporarily disable or bypass HA mechanisms for troubleshooting purposes. If an HA-related issue is found, it is the responsibility of the customer to engage with the database vendor so that a solution to that issue can be found.

Support Statement: Cloudera Support may require customers to temporarily bypass HA layers and connect directly to supported RDBMS back-ends to troubleshoot issues. Issues observed only when connected through HA layers are the responsibility of the customer DBA staff to resolve.

RDBMS Storage Sizing

The amount of RDBMS storage space used by CDP Private Cloud Base varies depending on the services that are installed and the operations performed. Approximately, the amount of RDBMS storage needed is between 10 MB and 100 MB per host in the CDP cluster.

You can better estimate the RDBMS storage space by deploying a test cluster with the approximate proportion of service roles that the full cluster can bear. Later, execute a sample set of operations, (including Data Recovery backup) for about 24 hours and observe the storage usage on the RDBMS. Next, extrapolate the usage to the full cluster size.

Sharing an RDBMS with other applications

The ability to share an RDBMS storage between CDP Private Cloud Base and other applications depends on many factors. Cloudera recommends that you do not share the RDBMS used by CDP Private Cloud Base with any other application.

For non-production clusters where cluster size is small, not expected to grow, and occasional glitches are tolerable, it is acceptable to share a database with other applications.

MySQL

For a production cluster, CDP Runtime services must not share a database server with other applications. For small clusters, this database can be shared by the CDP Runtime services. For large clusters (hosts > 500), each CDP Runtime service must have its own database server.

PostgreSQL

If you have a dedicated database team managing high-performance hardware, with the CDP Private Cloud Base databases stored on their own spindles (or raid array), then it can be possible to have the DB server shared with other applications. When the cluster size is very large (hosts > 1000), there might be performance issues between shared applications. Cloudera recommends that you do not share the CDP Private Cloud Base database server with other application usages.

If you do not have a dedicated database team that can analyze and tune RDBMS performance, it is recommended to follow the advice for MySQL as detailed above.

Oracle

For single-server Oracle installations, see the above description related to PostgreSQL.

If you are using a clustered system like Oracle RAC, with multiple servers, it is possible to use a shared DB service, since it is no longer a single server. The end user's DB team must monitor DB latency, scale the hardware, or tune DB parameters to ensure optimal performance.

Latency target

For end users attempting to tune a shared RDBMS, ensure that elapsed times must not exceed 40 milliseconds for the 99th percentile of SELECT statements on indexed single-row queries.

Java Requirements

CDP Private Cloud Base comprises of Cloudera Manager and Runtime services. Understand the specific JDK requirements for your deployments.

Related Information

Step 2: Install Java Development Kit

Supported JDKs

Please see the Cloudera Support Matrix for detailed information about supported JDKs.

For Cloudera Runtime services, **JDK-17** is provided as an additional platform support for Cloudera Runtime 7.1.9.

Supported JDK versions

Warning:

- Upgrading to Oracle JDK 1.8.351 or higher, OpenJDK 11.0.17 or higher, or OpenJDK 1.8.392, and Spark3 in CDE uses OpenJDK 11.0.17 which causes a Kerberos issue when deprecated 3DES and RC4 permitted encryption types are used.

  Workaround: Remove the deprecated 3DES and RC4 encryption types in the krb5.conf and kdc.conf files.
- JDK 8u271, JDK 8u281, and JDK 8u291 may cause socket leak issues due to JDK-8245417 and JDK-8256818. Pay attention to the build version of your JDK because some later builds are fixed as described in JDK-8256818.

  Workaround: Consider using a more recent version of the JDK like 8u282, or builds of the JDK where the issue is fixed.
- Oozie Workflow Graph Display in Hue does not work properly with JDK versions lower than 8u40.
- The default value of the YARN config **Add add-opens flags to MR containers** is **false** regardless of the JDK, Cloudera Manager or CDP version. To run distcp or any MapReduce application, this config must be turned on whenever JDK17 is used.
- If you are using JDK 17 on your cluster, you must run the JVM option `--add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.management/com.sun.jmx.mbeanserver=ALL-UNNAMED --add-exports=java.management/com.sun.jmx.mbeanserver=ALL-UNNAMED --add-exports=java.base/sun.net.dns=ALL-UNNAMED --add-exports=java.base/sun.net.util=ALL-UNNAMED` to ensure the jobs run successfully.

Table 11: Azul Open JDK versions that are tested and recommended

Azul Open JDK Version	Notes
17.0.7	
11.50.19	
8.56.0.21	Minimum required version

Table 12: Oracle JDK versions that are tested and recommended

Oracle JDK Version	Notes
17.0.6	
11.0.10+8	
1.8u181	Minimum required version

Table 13: OpenJDK versions that are tested and recommended

OpenJDK Version	Notes
17.0.7	17.0.2 is the minimum required OpenJDK 17 version for FIPS
11.0.4+11	11.0.3 is the minimum required OpenJDK 11 version for FIPS
1.8u231	For FIPS minimum required / latest version tested
1.8u232	Minimum required / Latest version tested

Note: Note the following about OpenJDK support:

- Updates above the minimum that are not listed are supported but not tested.
- Cloudera tests only the OpenJDK builds that are provided by each operating system, and only the versions listed in the support matrix.

Note: Cloudera Manager supports TLS 1.2 for Java 8 and Java 11. For Java 17 and higher versions, Cloudera Manager supports TLS 1.2 and TLS 1.3. For TLS 1.0 and TLS 1.1, Cloudera Manager supports Java 8, though Cloudera recommends not to use TLS 1.0 and TLS 1.1.

Important:

For JDK 8u241 and higher versions running on Kerberized clusters, you must disable referrals by setting sun.security.krb5.disableReferrals=true.

For example, with OpenJDK 1.8.0u242:

1. Open /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.el7_7.x86_64/jre/lib/security/java.security with a text editor.
2. Add sun.security.krb5.disableReferrals=true (it can be at the bottom of the file).
3. Add this property on each node that has the impacted JDK version.
4. Restart the applications using the JDK so the change takes effect.

For more information, see the KB article.

Support Notes

Note: Cloudera strongly recommends installing Oracle JDK at /usr/java/<*JDK-VERSION*> and OpenJDK at /usr/lib/jvm, which allows Cloudera Manager to auto-detect and use the correct JDK version. If you install the JDK anywhere else, there are additional steps required to configure Cloudera Manager with your chosen location. See Configuring a custom Java Home Location.

Note: A Java optimization called compressed oops (ordinary object pointers) enables a 64-bit JVM to address heap sizes up to about 32 GB using 4-byte pointers. For larger heap sizes, 8-byte pointers are required. This means that a heap size slightly less than 32 GB can hold more objects than a heap size slightly more than 32 GB.

If you do not need more than 32 GB heap, set your heap size to 31GB or less to avoid this issue. If you need 32 GB or more, set your heap size to 48 GB or higher to account for the larger pointers. In general, for heap sizes above 32 GB, multiply the amount of heap you need by 1.5.

Only 64 bit JDKs are supported.

Unless specifically excluded, Cloudera supports later updates to a major JDK release from the release that support was introduced. Cloudera excludes or removes support for select Java updates when security is jeopardized.

Running Runtime nodes within the same cluster on different JDK releases is not supported. All cluster hosts must use the same JDK update level.

Networking and Security Requirements

This topic describes the networking and security requirements for CDP Private Cloud Base.

Cloudera Runtime and Cloudera Manager Supported Transport Layer Security Versions

Note:

- Cloudera Manager supports TLS 1.2 for Java 8 and Java 11. For Java 17 and higher versions, Cloudera Manager supports TLS 1.2 and TLS 1.3. For TLS 1.0 and TLS 1.1, Cloudera Manager supports Java 8, though Cloudera recommends not to use TLS 1.0 and TLS 1.1.
- Domain name requirement: Environments, where domain names have non-LDH characters such as letters, digits, or hyphens, must be avoided for Hadoop deployments as they are unsupported.

The following components are supported by the indicated versions of Transport Layer Security (TLS):

Table 14: Components Supported by TLS

Component	Role	Name	Port	Version
Cloudera Manager	Cloudera Manager Server		7182	TLS 1.2
Cloudera Manager	Cloudera Manager Server		7183	TLS 1.2
Flume			9099	TLS 1.2
Flume		Avro Source/Sink		TLS 1.2
Flume		Flume HTTP Source/Sink		TLS 1.2
HBase	Master	HBase Master Web UI Port	60010	TLS 1.2
HDFS	NameNode	Secure NameNode Web UI Port	50470	TLS 1.2
HDFS	Secondary NameNode	Secure Secondary NameNode Web UI Port	50495	TLS 1.2
HDFS	HttpFS	REST Port	14000	TLS 1.1, TLS 1.2
Hive	HiveServer2	HiveServer2 Port	10000	TLS 1.2
Hue	Hue Server	Hue HTTP Port	8888	TLS 1.2
Impala	Impala Daemon	Impala Daemon Beeswax Port	21000	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala Daemon	Impala Daemon HiveServer2 Port	21050	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala Daemon	Impala Daemon Backend Port	22000	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala StateStore	StateStore Service Port	24000	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala Daemon	Impala Daemon HTTP Server Port	25000	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.

Component	Role	Name	Port	Version
Impala	Impala StateStore	StateStore HTTP Server Port	25010	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala Catalog Server	Catalog Server HTTP Server Port	25020	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Impala	Impala Catalog Server	Catalog Server Service Port	26000	TLS 1.0, TLS 1.1, TLS 1.2 We recommend that clients use the highest supported version, TLS 1.2.
Oozie	Oozie Server	Oozie HTTPS Port	11443	TLS 1.1, TLS 1.2
Ranger	Ranger Admin	Admin HTTPS Port	6182	TLS 1.2
Ranger KMS	Ranger KMS Server	Ranger KMS HTTPS Port	9494	TLS 1.2
Solr	Solr Server	Solr HTTP Port	8983	TLS 1.1, TLS 1.2
Solr	Solr Server	Solr HTTPS Port	8985	TLS 1.1, TLS 1.2
Spark	History Server		18080	TLS 1.2
YARN	ResourceManager	ResourceManager Web Application HTTP Port	8090	TLS 1.2
YARN	JobHistory Server	MRv1 JobHistory Web Application HTTP Port	19890	TLS 1.2

Cloudera Runtime and Cloudera Manager Networking and Security Requirements

The hosts in a Cloudera Manager deployment must satisfy the following networking and security requirements:

- Cluster Host Requirements:

  The hosts you intend to use must satisfy the following requirements:

  - You must be able to log in to the Cloudera Manager Server host using the root user account or an account that has passwordless sudo privileges.
  - The Cloudera Manager Server host must have uniform SSH access on the same port to all hosts. For more information, see *Runtime and Cloudera Manager Networking and Security Requirements*.
  - All hosts must have access to standard package repositories for the operating system and either archive.cloudera.com or a local repository with the required installation files.
  - SELinux must be disabled or set to permissive mode before running the installer.
- Networking Protocols Support

  CDP requires IPv4. IPv6 is not supported and must be disabled.

  Important: Refer to your OS documentation or contact your OS vendor for instructions on disabling IPv6.

  See also *Configure Network Names*.
- Multihoming Support

  Multihoming Cloudera Runtime or Cloudera Manager is not supported outside specifically certified Cloudera Data Platform partner appliances such as Oracle Big Data Appliance, Teradata Appliance for Hadoop, and Cray® Urika-XA system. Cloudera Data Platform finds that current Hadoop architectures combined with modern network infrastructures and security practices remove the need for multihoming. Multihoming, however, is beneficial internally in appliance form factors to take advantage of high-bandwidth InfiniBand interconnects.

Although some subareas of the product may work with unsupported custom multihoming configurations, there are known issues with multihoming. In addition, unknown issues may arise because multihoming is not covered by our test matrix outside the Cloudera Data Platform-certified partner appliances.

- Entropy

  Data at rest encryption requires sufficient entropy to ensure randomness.

  See entropy requirements in *Data at Rest Encryption Requirements*.
- Cluster hosts must have a working network name resolution system and correctly formatted /etc/hosts file. All cluster hosts must have properly configured forward and reverse host resolution through DNS. The /etc/hosts files must:

  - Contain consistent information about hostnames and IP addresses across all hosts
  - Not contain uppercase hostnames
  - Not contain duplicate IP addresses

  Cluster hosts must not use aliases, either in /etc/hosts or in configuring DNS. A properly formatted /etc/hosts file should be similar to the following example:

```
127.0.0.1 localhost.localdomain localhost
192.168.1.1 cluster-01.example.com cluster-01
192.168.1.2 cluster-02.example.com cluster-02
192.168.1.3 cluster-03.example.com cluster-03
```
- In most cases, the Cloudera Manager Server must have SSH access to the cluster hosts when you run the installation or upgrade wizard. You must log in using a root account or an account that has password-less sudo permission. For authentication during the installation and upgrade procedures, you must either enter the password or upload a public and private key pair for the root or sudo user account. If you want to use a public and private key pair, the public key must be installed on the cluster hosts before you use Cloudera Manager.

  For a list of sudo commands that are supported by Cloudera Manager, see Cloudera Manager sudo command options.

  Cloudera Manager uses SSH only during the initial install or upgrade. Once the cluster is set up, you can disable root SSH access or change the root password. Cloudera Manager does not save SSH credentials, and all credential information is discarded when the installation is complete.
- The Cloudera Manager Agent runs as root so that it can make sure that the required directories are created and that processes and files are owned by the appropriate user (for example, the hdfs and mapred users).
- Security-Enhanced Linux (SELinux) must not block Cloudera Manager or Runtime operations.

  Note: Cloudera Enterprise is supported on platforms with Security-Enhanced Linux (SELinux) enabled and in enforcing mode. Cloudera is not responsible for SELinux policy development, support, or enforcement. If you experience issues running Cloudera software with SELinux enabled, contact your OS provider for assistance.

  If you are using SELinux in enforcing mode, Cloudera Support can request that you disable SELinux or change the mode to permissive to rule out SELinux as a factor when investigating reported issues.
- Firewalls (such as iptables and firewalld) must be disabled or configured to allow access to ports used by Cloudera Manager, Runtime, and related services.
- For RHEL and CentOS, the /etc/sysconfig/network file on each host must contain the correct hostname.
- Cloudera Manager and Runtime use several user accounts and groups to complete their tasks. The set of user accounts and groups varies according to the components you choose to install. Do not delete these accounts or groups and do not modify their permissions and rights. Ensure that no existing systems prevent these accounts and groups from functioning. For example, if you have scripts that delete user accounts not in an allowlist, add these

accounts to the list of permitted accounts. Cloudera Manager, Runtime, and managed services create and use the following accounts and groups:

Table 15: Users and Groups

Component (Version)	Unix User ID	Groups	Functionality
Apache Atlas	atlas	atlas, hadoop	Apache Atlas by default has atlas as user and group. It is configurable
Apache Flink	flink	flink	The Flink Dashboard runs as this user.
Apache HBase	hbase	hbase	The Master and the RegionServer processes run as this user.
Apache HBase Indexer	hbase	hbase	The indexer servers are run as this user.
Apache HDFS	hdfs	hdfs, hadoop	The NameNode and DataNodes run as this user, and the HDFS root directory as well as the directories used for edit logs should be owned by it.
Apache Hive Hive on Tez	hive	hive	The HiveServer2 process and the Hive Metastore processes run as this user.A user must be defined for Hive access to its Metastore DB (for example, MySQL or Postgres) but it can be any identifier and does not correspond to a Unix uid. This is javax.jdo.option.ConnectionUserName in hive-site.xml.
Apache Impala	impala	impala, hive	Impala services run as this user.
Apache Kafka	kafka	kafka	Kafka brokers, mirrorMaker, and Connect workers run as this user.
Apache Knox	knox	knox	Apache Knox Gateway Server runs as this user
Apache Kudu	kudu	kudu	Kudu services run as this user.
Apache Livy	livy	livy	The Livy Server process runs as this user
Apache NiFi	nifi	nifi	Runs as the nifi user
Apache NiFi Registry	nifiregistry	nifiregistry	Runs as the nifiregistry user
Apache Oozie	oozie	oozie	The Oozie service runs as this user.
Apache Ozone	hdfs	hdfs, hadoop	Ozone Manager, Storage Container Manager (SCM), Recon and Ozone Datanodes run as this user.
Apache Parquet	~	~	No special users.
Apache Phoenix	phoenix	phoenix	The Phoenix Query Server runs as this user
Apache Ranger	ranger	ranger, hadoop	Ranger Admin, Usersync and Tagsync services by default have ranger as user and ranger, hadoop as groups. It is configurable.

Component (Version)	Unix User ID	Groups	Functionality
Apache Ranger KMS	kms	kms	Ranger KMS runs with kms user and group. It is configurable.
Apache Ranger Raz	rangerraz	ranger	Ranger Raz runs with rangerraz user and is part of the ranger group.
Apache Ranger RMS	rangerrms	ranger	Ranger RMS runs with rangerrms user and is part of the ranger group.
Apache Solr	solr	solr	The Solr processes run as this user.
Apache Spark	spark	spark	The Spark History Server process runs as this user.
Apache Sqoop	sqoop	sqoop	This user is only for the Sqoop1 Metastore, a configuration option that is not recommended.
Apache YARN	yarn	yarn, hadoop	Without Kerberos, all YARN services and applications run as this user. The LinuxContainerExecutor binary is owned by this user for Kerberos.
Apache Zeppelin	zeppelin	zeppelin	The Zeppelin Server process runs as this user
Apache ZooKeeper	zookeeper	zookeeper	The ZooKeeper processes run as this user. It is not configurable.
Cloudera Manager (all versions)	cloudera-scm	cloudera-scm	Clusters managed by Cloudera Manager run Cloudera Manager Server, monitoring roles, and other Cloudera Server processes as cloudera-scm. Requires keytab file named cmf.keytab because name is hard-coded in Cloudera Manager.
Cruise Control	cruisecontrol	hadoop	The Cruise Control process runs as this user.
HttpFS	httpfs	httpfs	The HttpFS service runs as this user. See “HttpFS authentication” for instructions on how to generate the merged httpfs-http.keytab file.
Hue	hue	hue	Hue services run as this user.
Hue Load Balancer	apache	apache	The Hue Load balancer has a dependency on the apache2 package that uses the apache user name. Cloudera Manager does not run processes using this user ID.
Key Trustee Server	keytrustee	keytrustee	The Key Trustee Server service runs as this user.
Schema Registry	schemaregistry	hadoop	The Schema Registry process runs as this user.
Streams Messaging Manager	streamsmsgmgr	streamsmsgmgr	The Streams Messaging Manager processes runs as this user.

Component (Version)	Unix User ID	Groups	Functionality
Streams Replication Manager	streamsrepmgr	streamsrepmgr	The Streams Replication Manager processes runs as this user.

Data at Rest Encryption Requirements

This topic describes the data at rest encryption requirements for Cloudera Private Cloud Base.

Encryption comprises several components, each with its own requirements.

Data at rest encryption protection can be applied at a number of levels within Hadoop:

- OS filesystem-level
- Network-level
- HDFS-level (protects both data at rest and in transit)

This section contains the various hardware and software requirements for all encryption products used for Data at Rest Encryption.

For more information on supported operating systems, see Cloudera Support Matrix.

For more information on the components, concepts, and architecture for encrypting data at rest, see Encrypting Data at Rest.

Entropy Requirements

Cryptographic operations require entropy to ensure randomness.

You can check the available entropy on a Linux system by running the following command:

```
cat /proc/sys/kernel/random/entropy_avail
```

The output displays the entropy currently available. Check the entropy several times to determine the state of the entropy pool on the system. If the entropy is consistently low (500 or less), you must increase it by installing rng-tools and starting the rngd service.

For RHEL 7, run the following commands:

```
sudo yum install rng-tools
cp /usr/lib/systemd/system/rngd.service /etc/systemd/system/
sed -i -e 's/ExecStart=\/sbin\/rngd -f/ExecStart=\/sbin\/rngd -f -r \/dev\/u
random/' /etc/systemd/system/rngd.service
systemctl daemon-reload
systemctl start rngd
systemctl enable rngd
```

Make sure that the hosts running Key Trustee Server and Ranger KMS have sufficient entropy to perform cryptographic operations.

Cloudera Manager Requirements

Installing and managing Key Trustee Server using Cloudera Manager requires Cloudera Manager 5.4.0 and higher.

umask Requirements

Key Trustee Server installation requires the default umask of 0022.

Network Requirements

For new Key Trustee Server installations (5.4.0 and higher) and migrated upgrades (see Migrate Apache Web Server to CherryPy for more information), Key Trustee Server requires the following TCP ports to be opened for inbound traffic:

- 11371

  Clients connect to this port over HTTPS.
- 11381 (PostgreSQL)

  The passive Key Trustee Server connects to this port for database replication.

For upgrades that are not migrated to the CherryPy web server, the pre-upgrade port settings are preserved:

- 80

  Clients connect to this port over HTTP to obtain the Key Trustee Server public key.
- 443 (HTTPS)

  Clients connect to this port over HTTPS.
- 5432 (PostgreSQL)

  The passive Key Trustee Server connects to this port for database replication.

TLS Certificate Requirements

To ensure secure network traffic, Cloudera recommends obtaining Transport Layer Security (TLS) certificates specific to the hostname of your Key Trustee Server. To obtain the certificate, generate a Certificate Signing Request (CSR) for the fully qualified domain name (FQDN) of the Key Trustee Server host. The CSR must be signed by a trusted Certificate Authority (CA). After the certificate has been verified and signed by the CA, the Key Trustee Server TLS configuration requires:

- The CA-signed certificate
- The private key used to generate the original CSR
- The intermediate certificate/chain file (provided by the CA)

Cloudera recommends not using self-signed certificates. If you use self-signed certificates, you must use the --skip-ssl-check parameter when registering Navigator Encrypt with the Key Trustee Server. This skips TLS hostname validation, which safeguards against certain network-level attacks. For more information regarding insecure mode, see *Registration Options*.

Third-party filesystems

This topic describes the third-party filesystems supported by Cloudera Private Cloud Base.

Cloudera Private Cloud Base supports the following third-party filesystems:

Please see the CDP Private Cloud Base Release Guide for Dell EMC PowerScale support.

Please see the CDP Private Cloud Base Release Guide for IBM Spectrum Scale support.

Trial Installation

These topics provide instructions for installing the trial version of Cloudera Private Cloud Base in a non-production environment for demonstration and proof-of-concept use cases.

In these procedures, Cloudera Manager automates the installation of the JDK, Cloudera Manager Server, an embedded PostgreSQL database, Cloudera Manager Agent, Cloudera Runtime, and other managed services on cluster hosts. Cloudera Manager also configures databases for the Cloudera Manager Server and Hive Metastore, Ranger, and for Cloudera Management Service roles.

This installation method is recommended for trial deployments, but is not supported for production deployments because it is not designed to scale. To use this method, server and cluster hosts must satisfy the following requirements:

- All hosts must have a supported operating system installed.
- You must be able to log in to the Cloudera Manager Server host using the root user account or an account that has passwordless sudo privileges.
- The Cloudera Manager Server host must have uniform SSH access on the same port to all hosts. For more information, see Runtime and Cloudera Manager Networking and Security Requirements.
- All hosts must have access to standard package repositories for the operating system and either archive.cloudera .com or a local repository with the required installation files.
- SELinux must be disabled or set to permissive mode before running the installer.

Related Information

Cloudera Private Cloud Base Trial Download Information

Cloudera Private Cloud Base Installation Guide

Installing a Trial Cluster

In this procedure, Cloudera Manager automates the installation of the Oracle JDK, Cloudera Manager Server, embedded PostgreSQL database, Cloudera Manager Agent, Runtime, and managed service software on cluster hosts. Cloudera Manager also configures databases for the Cloudera Manager Server and Hive Metastore and optionally for Cloudera Management Service roles.

Important: This procedure is intended for trial and proof-of-concept deployments only. It is not supported for production deployments because it is not designed to scale.

Cluster Host Requirements:

The hosts you intend to use must satisfy the following requirements:

- You must be able to log in to the Cloudera Manager Server host using the root user account or an account that has passwordless sudo privileges.
- The Cloudera Manager Server host must have uniform SSH access on the same port to all hosts. For more information, see *Runtime and Cloudera Manager Networking and Security Requirements*.
- All hosts must have access to standard package repositories for the operating system and either archive.cloudera .com or a local repository with the required installation files.
- SELinux must be disabled or set to permissive mode before running the installer.

Refer to the following topics for the steps required to install a trial cluster.

Before You Begin a Trial Installation

Before you begin a trial installation, you must disable SELinux if you want the Cloudera Manager installer to run. You can also optionally configure an HTTP proxy.

(Optional) Configure an HTTP Proxy

The Cloudera Manager installer accesses archive.cloudera.com by using yum on RHEL systems. zypper on SLES systems, or apt-get on Ubuntu systems. If your hosts access the Internet through an HTTP proxy, you can configure yum system-wide, to access archive.cloudera.com through a proxy.

To do so, modify the system configuration on every cluster host as follows:

OS	File	Property
RHEL-compatible	/etc/yum.conf	proxy=http://*SERVER*:*PORT*/
SLES	/root/.curlrc	--proxy=http://*SERVER*:*PORT*/

OS	File	Property
Ubuntu	/etc/apt/apt.conf	Acquire::http::Proxy "http ://*SERVER*:*PORT*";

Disable SELinux

Note: Cloudera Private Cloud Base is supported on platforms with Security-Enhanced Linux (SELinux) enabled and in enforcing mode. Cloudera is not responsible for SELinux policy development, support, or enforcement. If you experience issues running Cloudera software with SELinux enabled, contact your OS provider for assistance.

If you are using SELinux in enforcing mode, Cloudera Support can request that you disable SELinux or change the mode to permissive to rule out SELinux as a factor when investigating reported issues.

Although Cloudera supports running Cloudera software with SELinux enabled, the Cloudera Manager installer will not proceed if SELinux is enabled. Disable SELinux or set it to permissive mode before running the installer.

After you have installed and deployed Cloudera Manager and Runtime, you can re-enable SELinux by changing SELINUX=permissive back to SELINUX=enforcing in /etc/selinux/config (or /etc/sysconfig/selinux), and then running the following command to immediately switch to enforcing mode:

```
setenforce 1
```

If you are having trouble getting Cloudera Software working with SELinux, contact your OS vendor for support. Cloudera is not responsible for developing or supporting SELinux policies.

Download the Trial version of Cloudera Private Cloud Base

You can download the trial version of Cloudera Private Cloud Base from the Cloudera Download site.

About this task

You can use the trial software for 60 days without obtaining a license key file. The trial installation includes an embedded PostgreSQL database and is not suitable for a production environment.

Procedure

1. Go to the trial download page for Cloudera Private Cloud Base.
2. Click Try Now.
3. Follow the download-instructions.

What to do next

Run the Cloudera Manager Server Installer.

Run the Cloudera Manager Server Installer

Run the Cloudera Manager installer to the cluster host to which you are installing the Cloudera Manager Server. By default, the automated installer binary (cloudera-manager-installer.bin) installs the highest version of Cloudera Manager.

Before you begin

- Download the trial software.

Procedure

1. Run the Cloudera Manager installer:
   a) Change cloudera-manager-installer.bin to have execute permissions:

   ```
   chmod u+x cloudera-manager-installer.bin
   ```

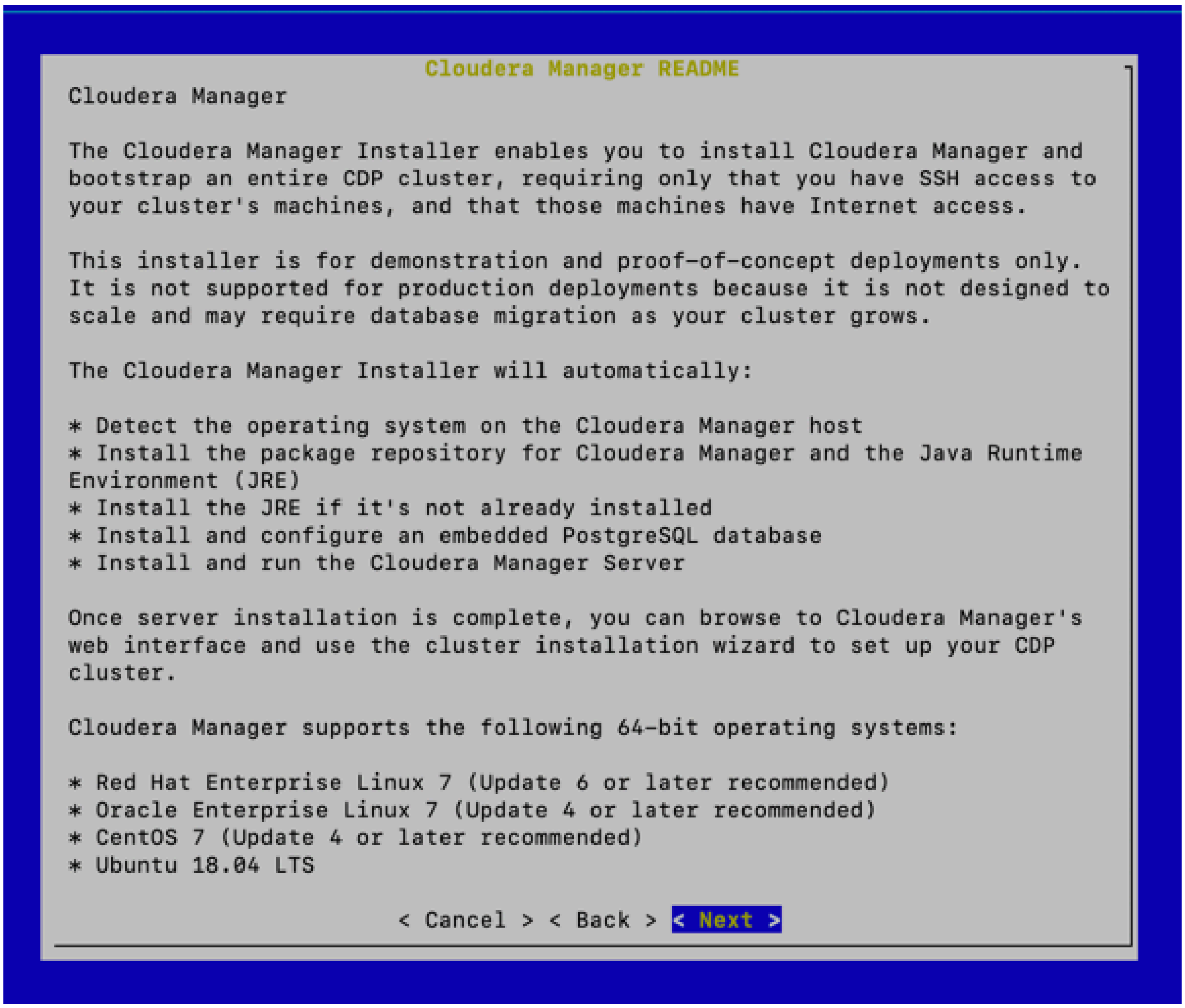
   b) Run the Cloudera Manager Server installer:

   ```
   sudo ./cloudera-manager-installer.bin
   ```

   c) For clusters without Internet access: Install Cloudera Manager packages from a local repository:

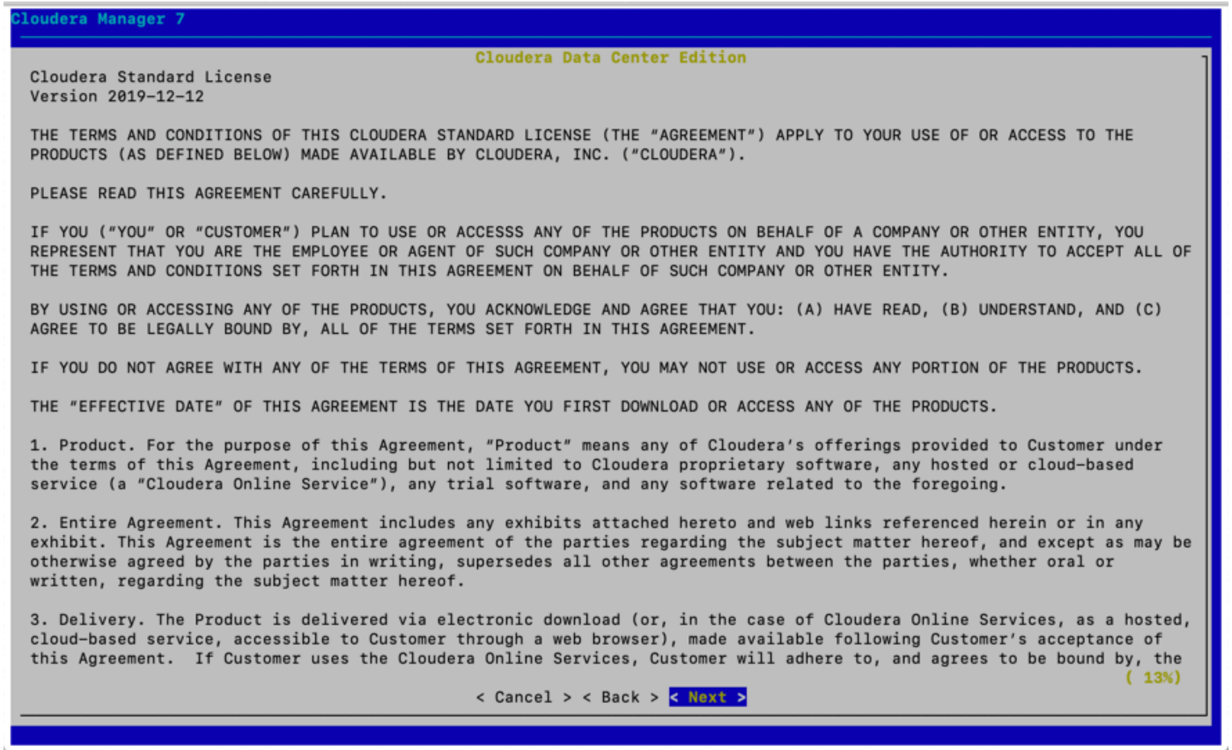
   ```
   sudo ./cloudera-manager-installer.bin --skip_repo_package=1
   ```

   The **Cloudera Manager Read Me** page appears.



2. Click Next.

The **Cloudera Standard License** page appears.



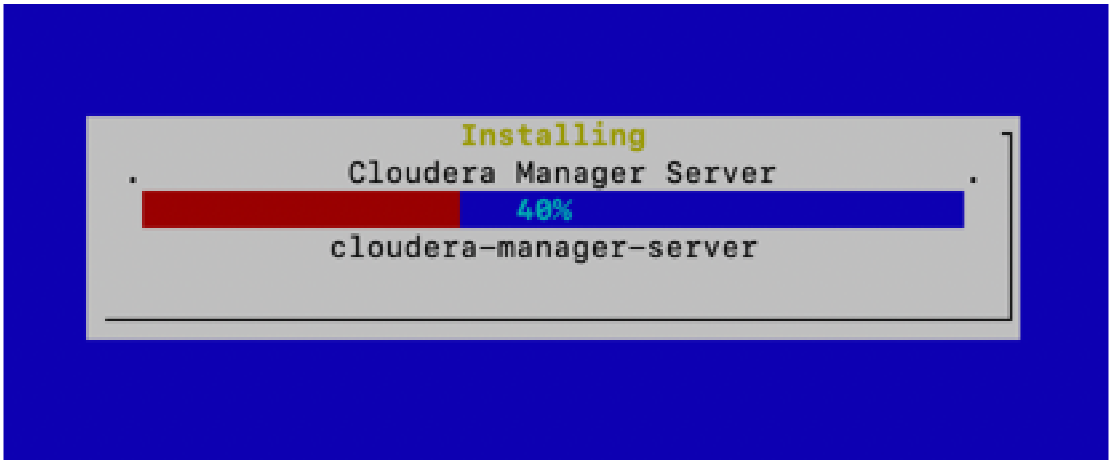
3. Click Next to accept the license agreement.

   The the installer starts and does the following:

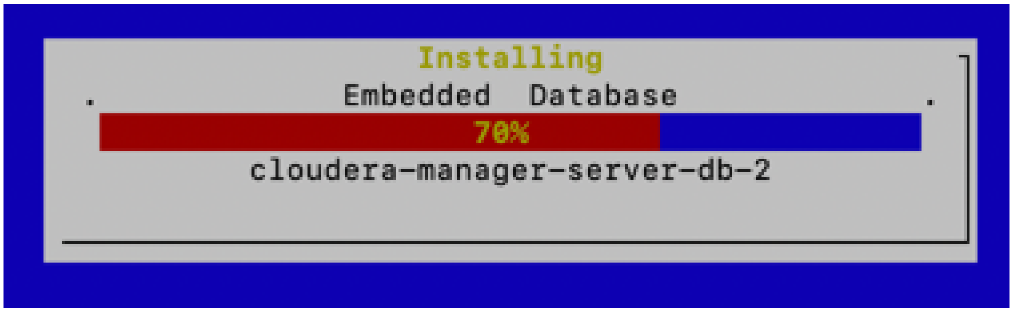
   a. Installs Oracle JDK.

   b. Installs the Cloudera Manager Server.

   c. Installs the embedded PostgreSQL packages and starts the database and Cloudera Manager Server.

Note: If the installation is interrupted, run the following command on the Cloudera Manager Server host before you retry the installation:

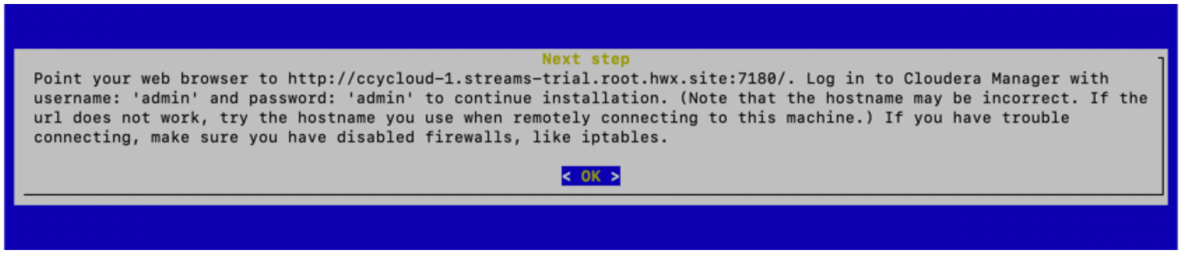
```
sudo /usr/share/cmf/uninstall-cloudera-manager.sh
```

The log files for the installer are stored in /var/log/cloudera-manager-installer/.

4. Exit the installer:

a) When the installation completes, the complete URL for the Cloudera Manager Admin Console displays, including the default port number: 7180.

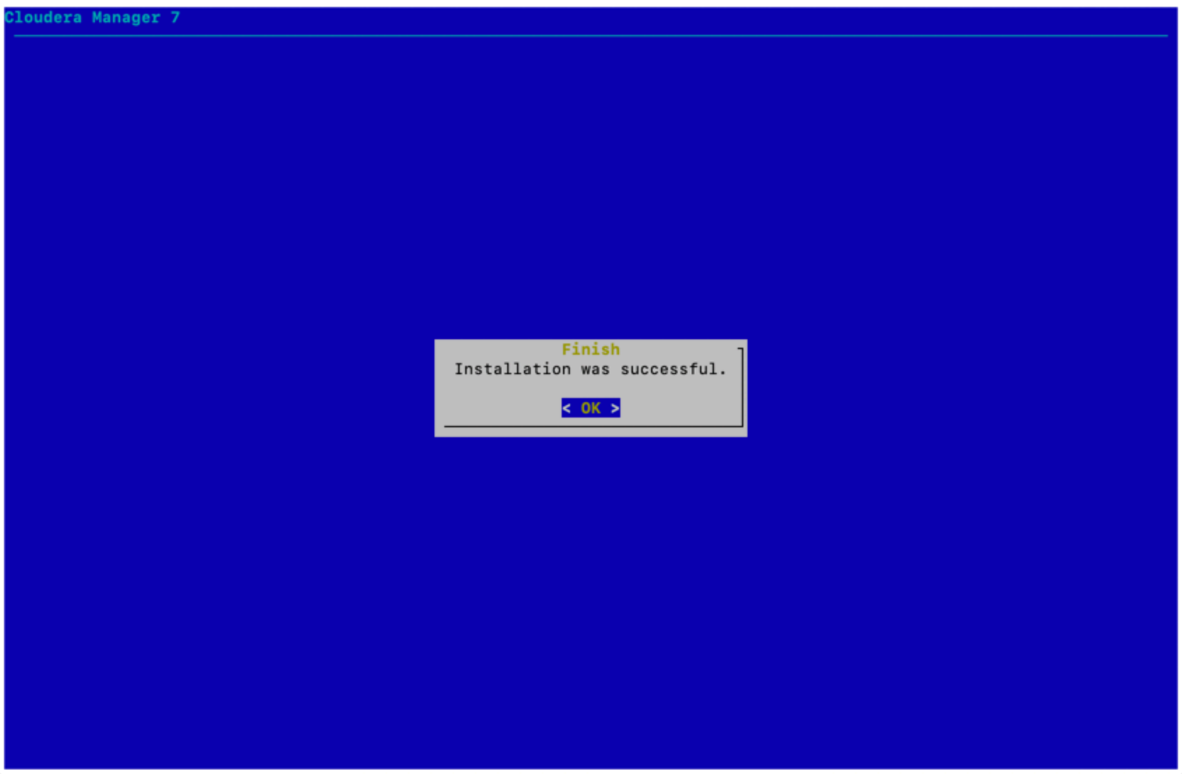
Important: Make a note of this URL or take a screen capture as you will need it for the next task.



b) Click OK.

The success message appears.

c) Click OK to exit the installer.



d) Wait a few minutes for the Cloudera Manager Server to start. To observe the startup process, run sudo tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log on the Cloudera Manager Server host. When you see the following log entry, the Cloudera Manager Admin Console is ready:

```
INFO WebServerImpl:com.cloudera.server.cmf.WebServerImpl: Started Jetty
server.
```

What to do next

Install Cloudera Runtime

Install Cloudera Runtime

After you have installed Cloudera Manager, log in to Cloudera Manager to access the **Add Cluster - Installation** wizard. Here you will add hosts to form a cluster and install Cloudera Runtime and Cloudera Manager Agent software.

Before you begin

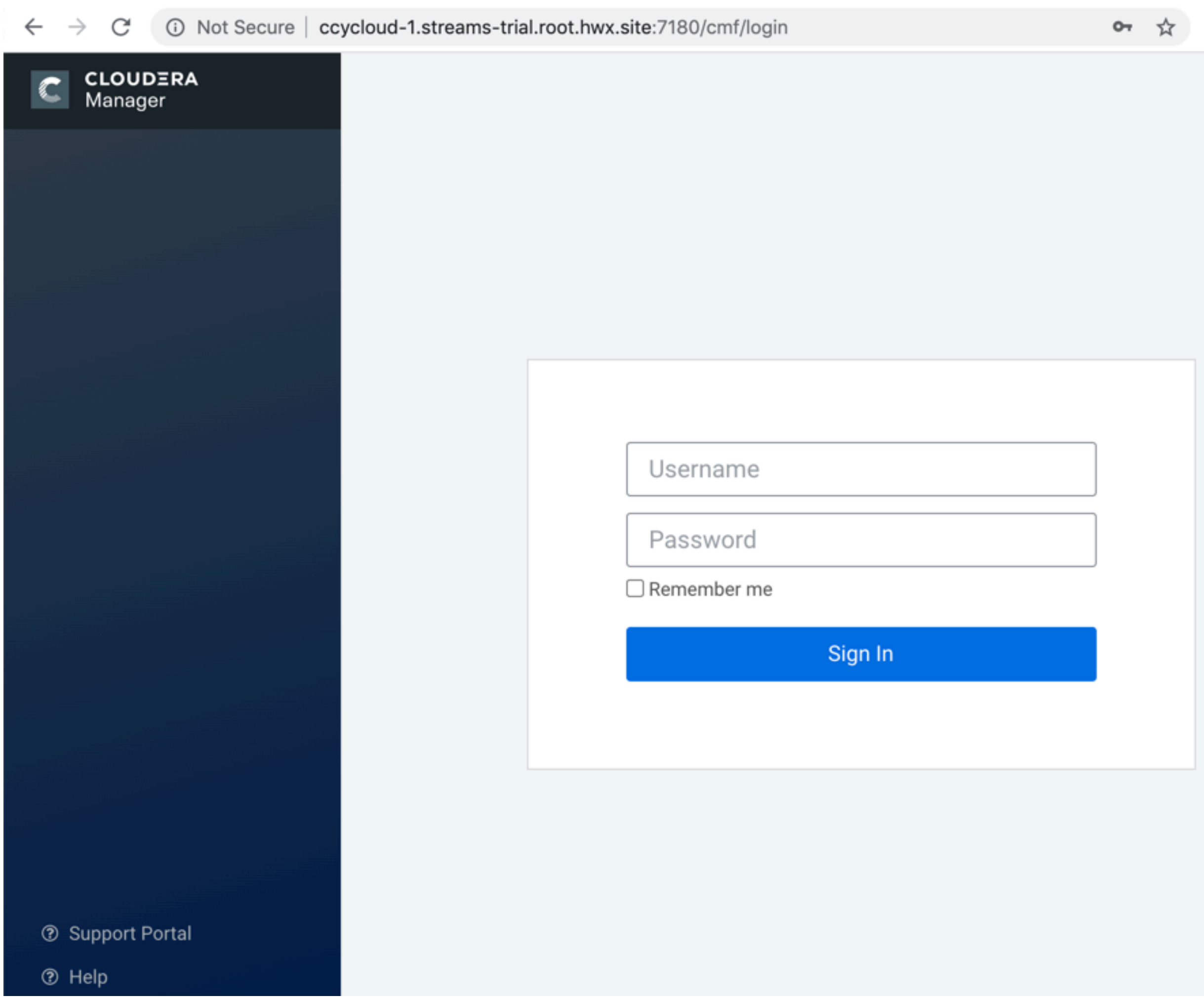
- You have installed Cloudera Manager.

Procedure

1. In a web browser, enter the URL that the Cloudera Manager Installer displayed in the previous task: http://<*SERVER_HOST*>:7180, where <*SERVER_HOST*> is the FQDN or IP address of the host where the Cloudera Manager Server is running.

   For example: http://ccycloud-1.streams-trial.root.hwx.site:7180

   The **Cloudera Manager Sign In** page appears.

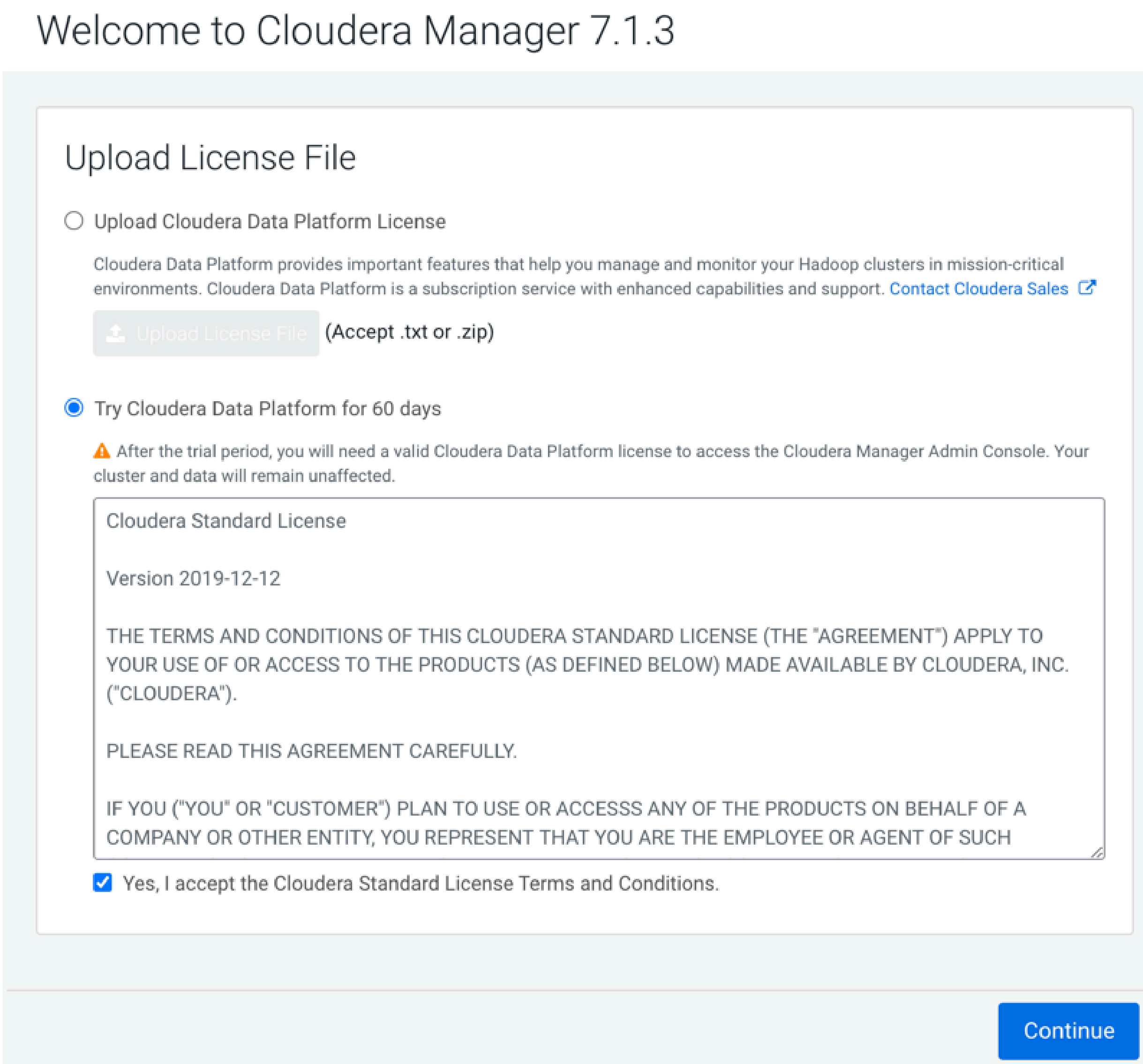
2. Sign in with the default credentials:
   - Username: admin
   - Password: admin

   Click Sign In.

The **Welcome to Cloudera Manager** page appears.

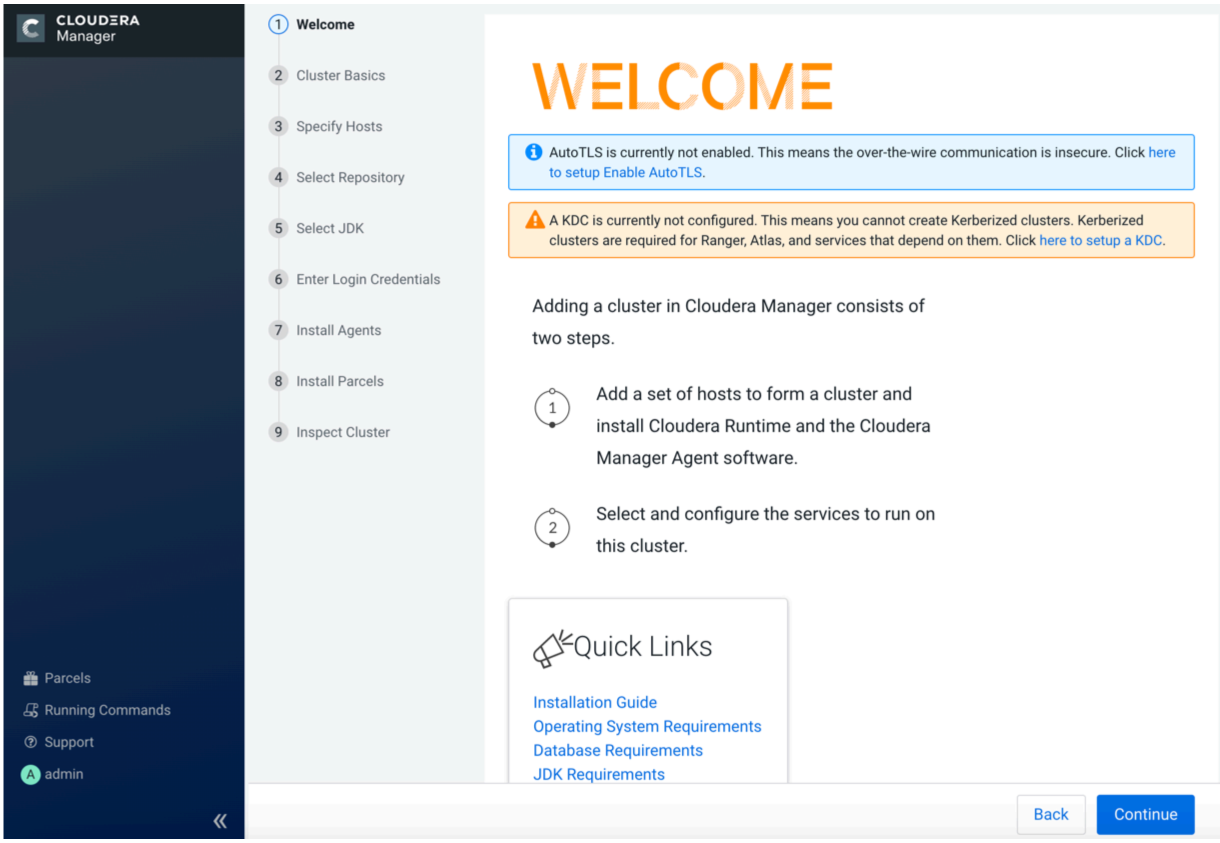
3. Select:

- Try Cloudera Data Platform for 60 days
- Yes, I accept the Cloudera Standard License Terms and Conditions



4. Click Continue.

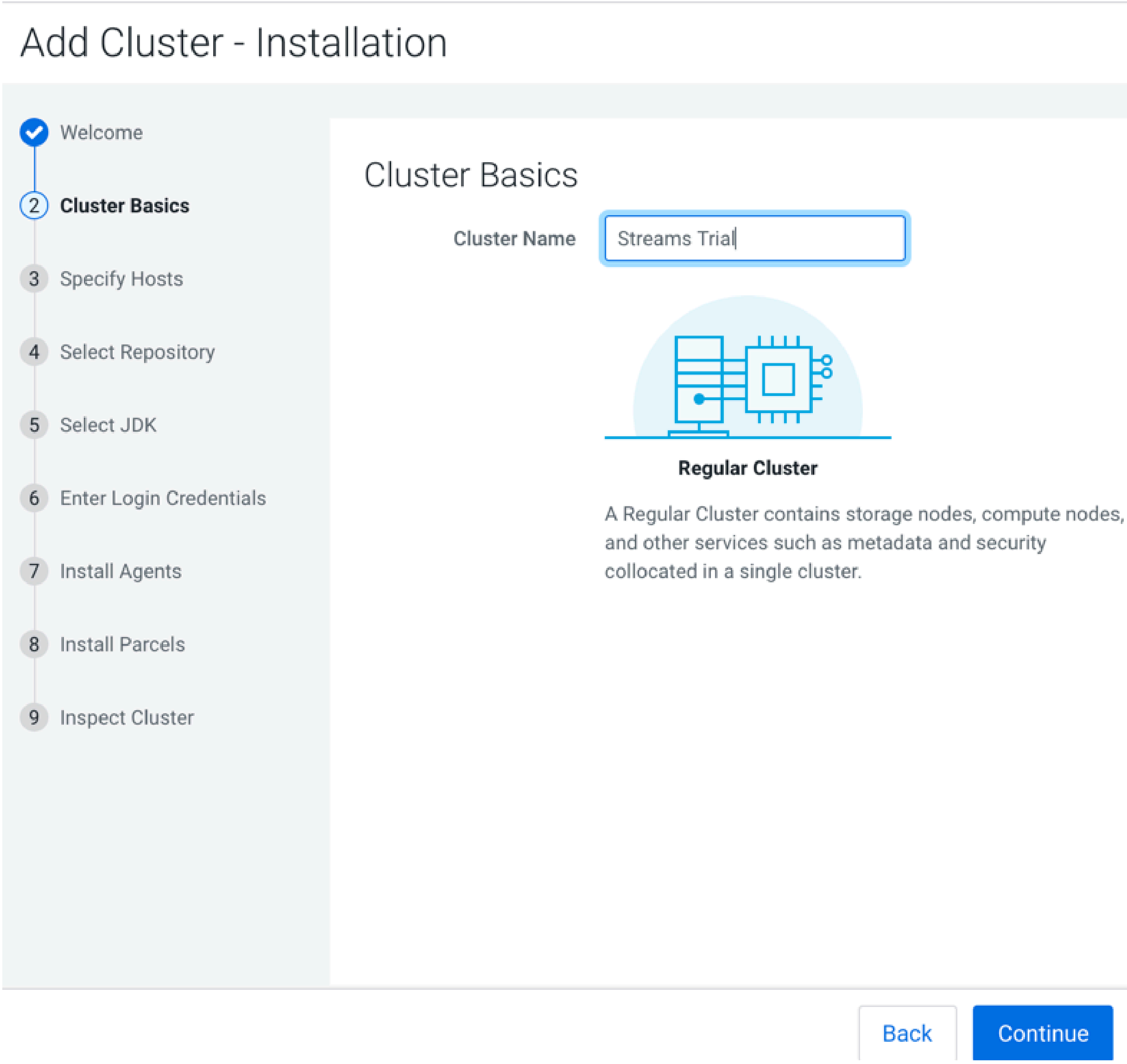
The **Add Cluster - Installation** page, **Welcome** section appears. The steps on the left let you know where you are in the workflow.



5. Click Continue.

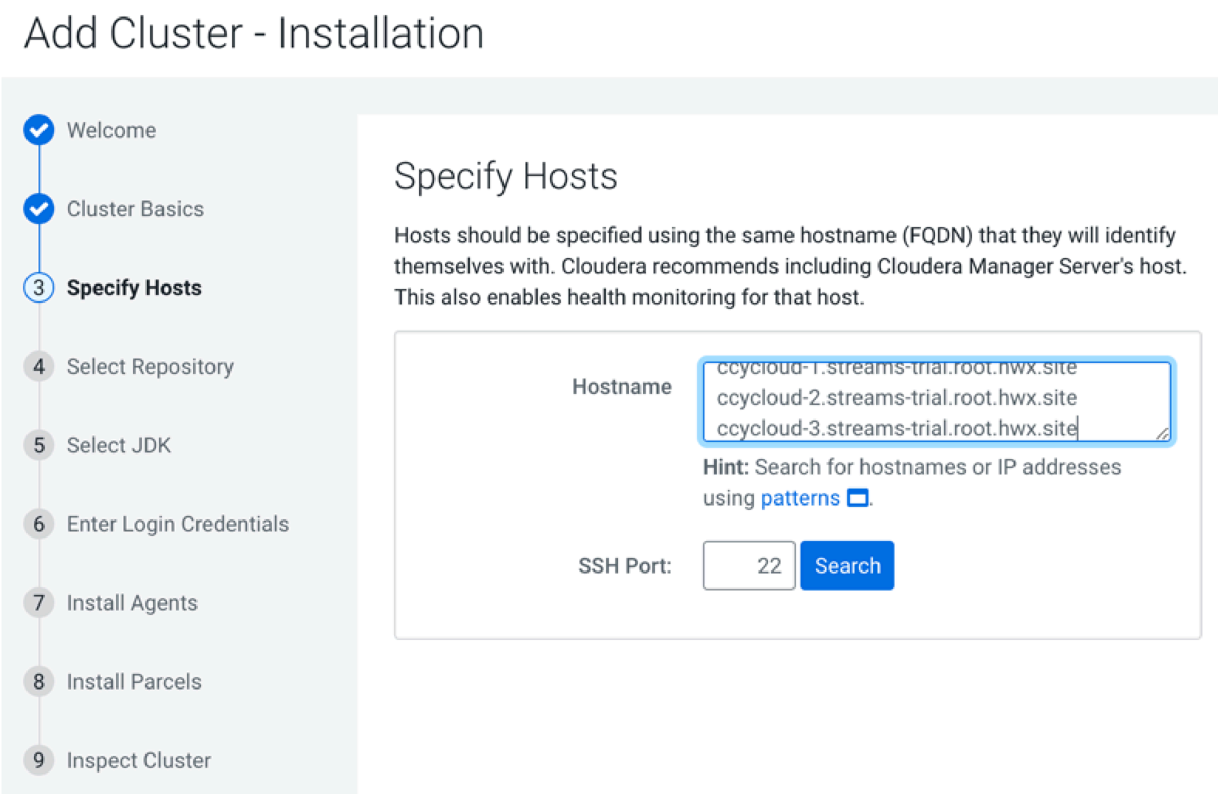
The Cluster Basics section appears.

6. Enter a name for the cluster and click Continue.



The **Specify Hosts** section appears.

7. Enter the cluster host names or IP addresses in the Hostnames field.

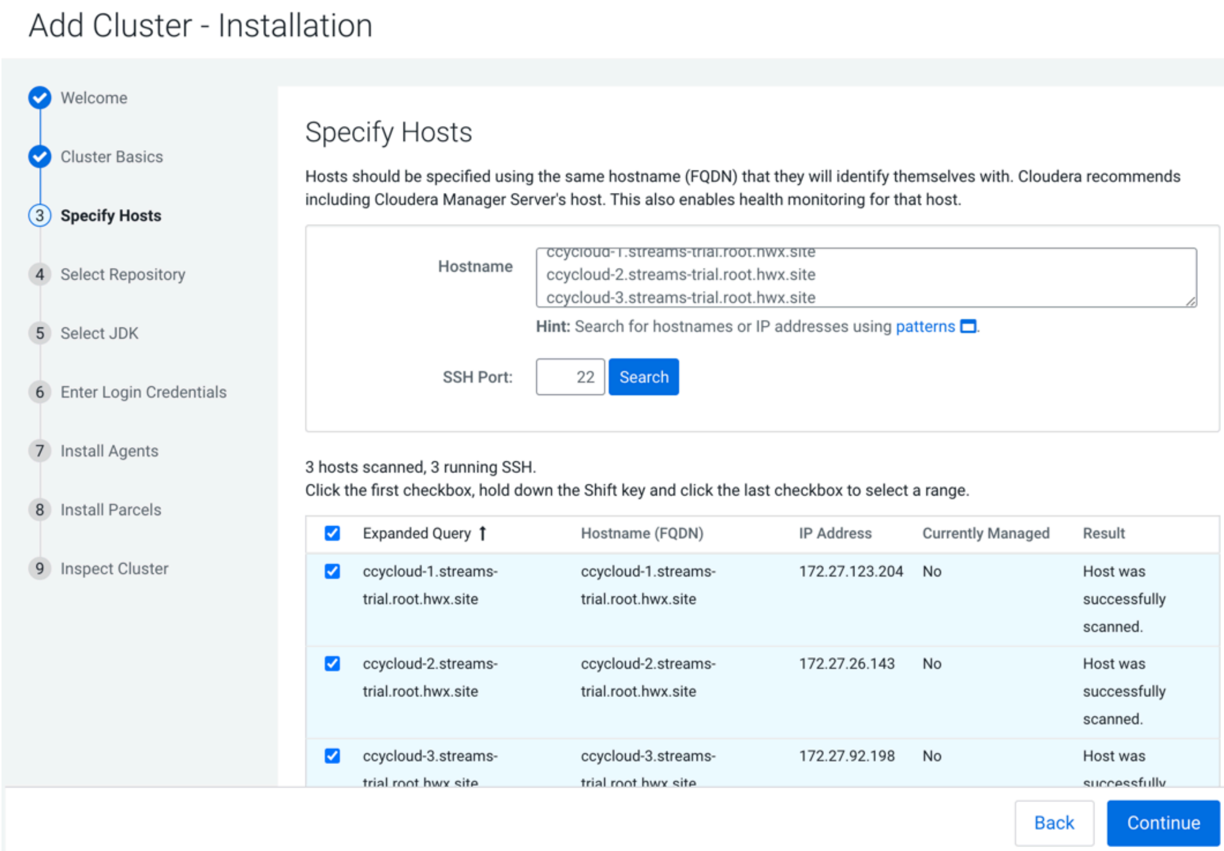


You can specify host name and IP address ranges as follows:

Expansion Range	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].example.com	host1.example.com, host2.example.com, host3.example.com
host[07-10].example.com	host07.example.com, host08.example.com, host09.example.com, host10.example.com

8. Click Search.

Cloudera Manager discovers the hosts.

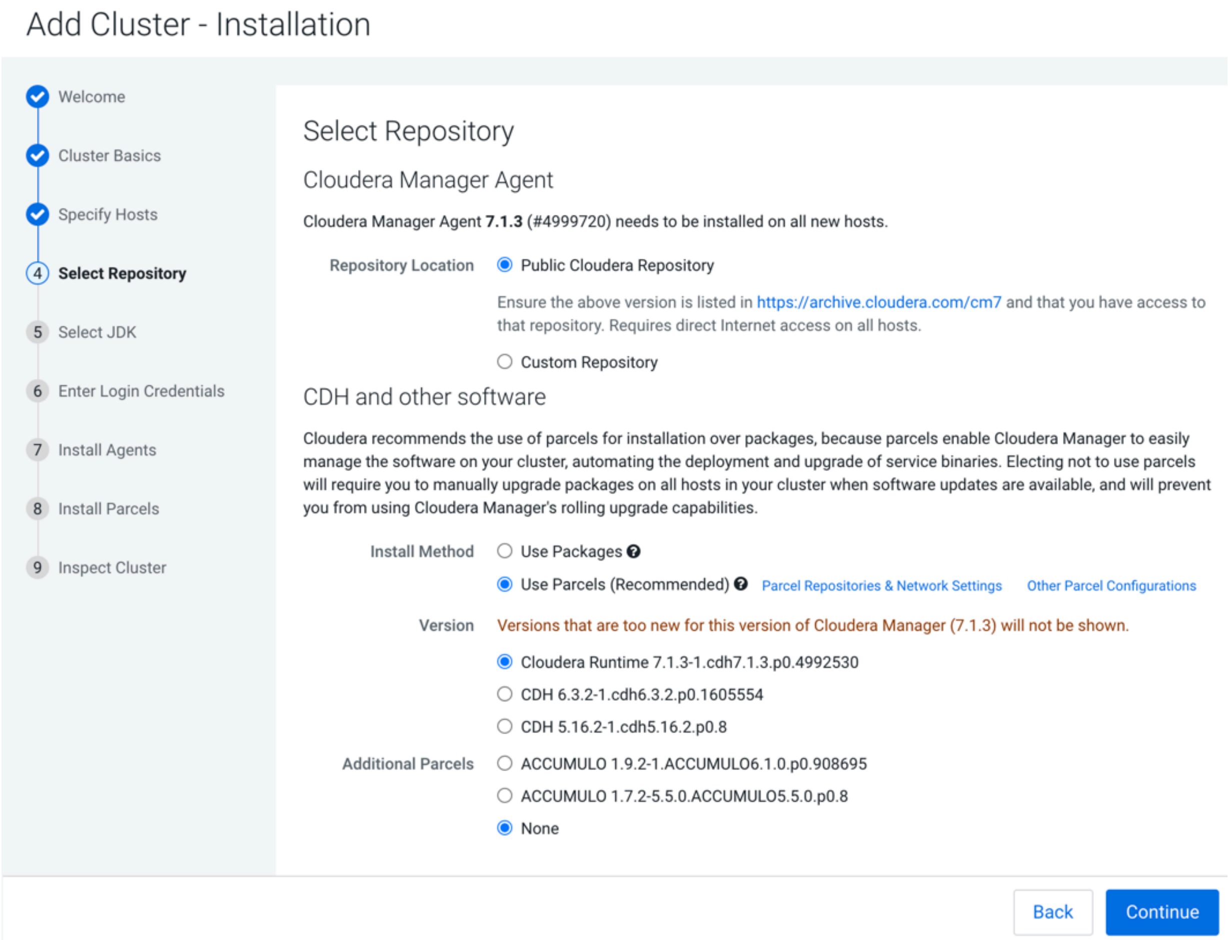


9. Verify host entries, deselect any that you do not want to install services on, and click Continue.

The **Select Repository** section appears.

10. Select the following options:

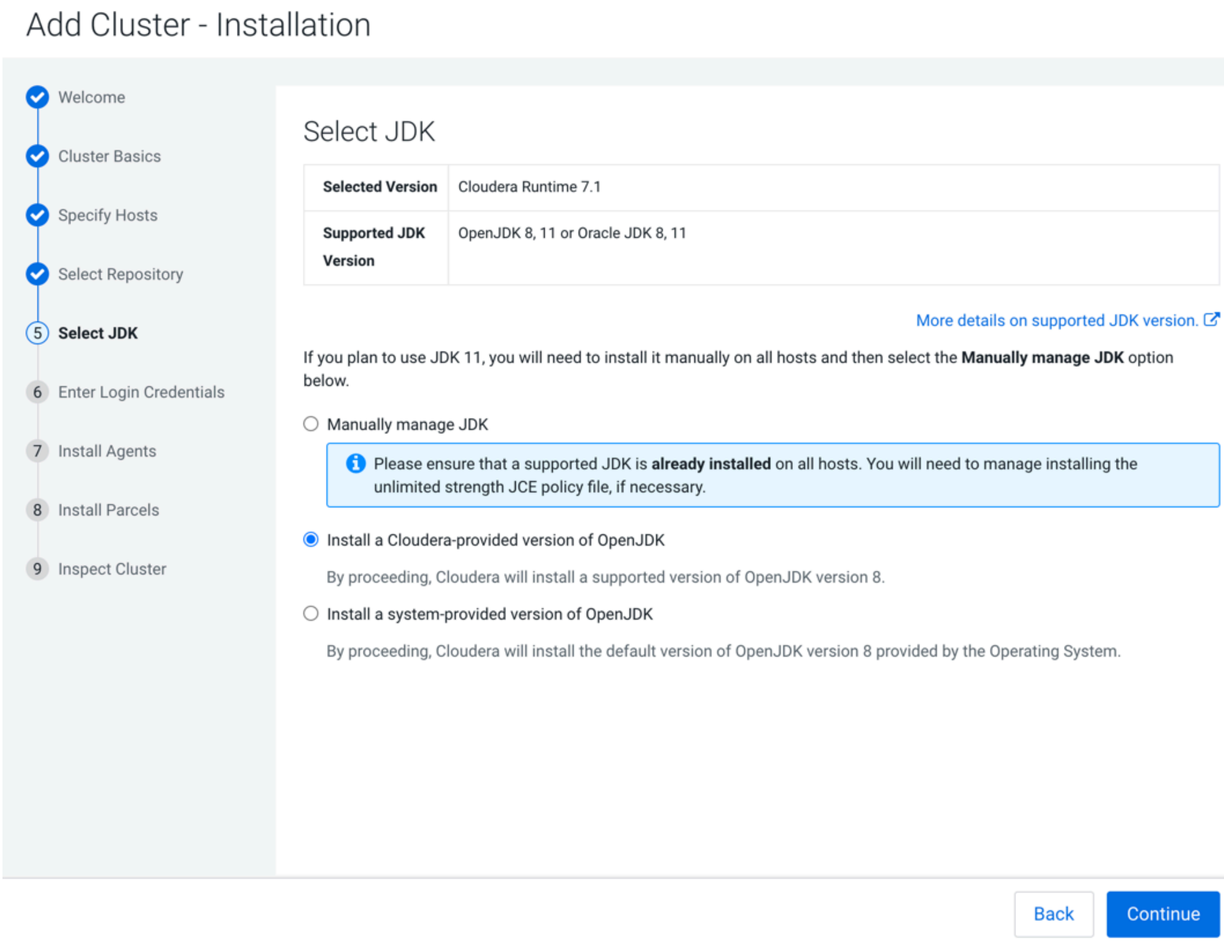
- Public Cloudera Repository
- Use Parcels
- The version of Cloudera Runtime that you want to install.
- In the Additional Parcels section, None.



11. Click Continue.

The **Select JDK** section appears.

12. Select Install a Cloudera-provided version of OpenJDK.

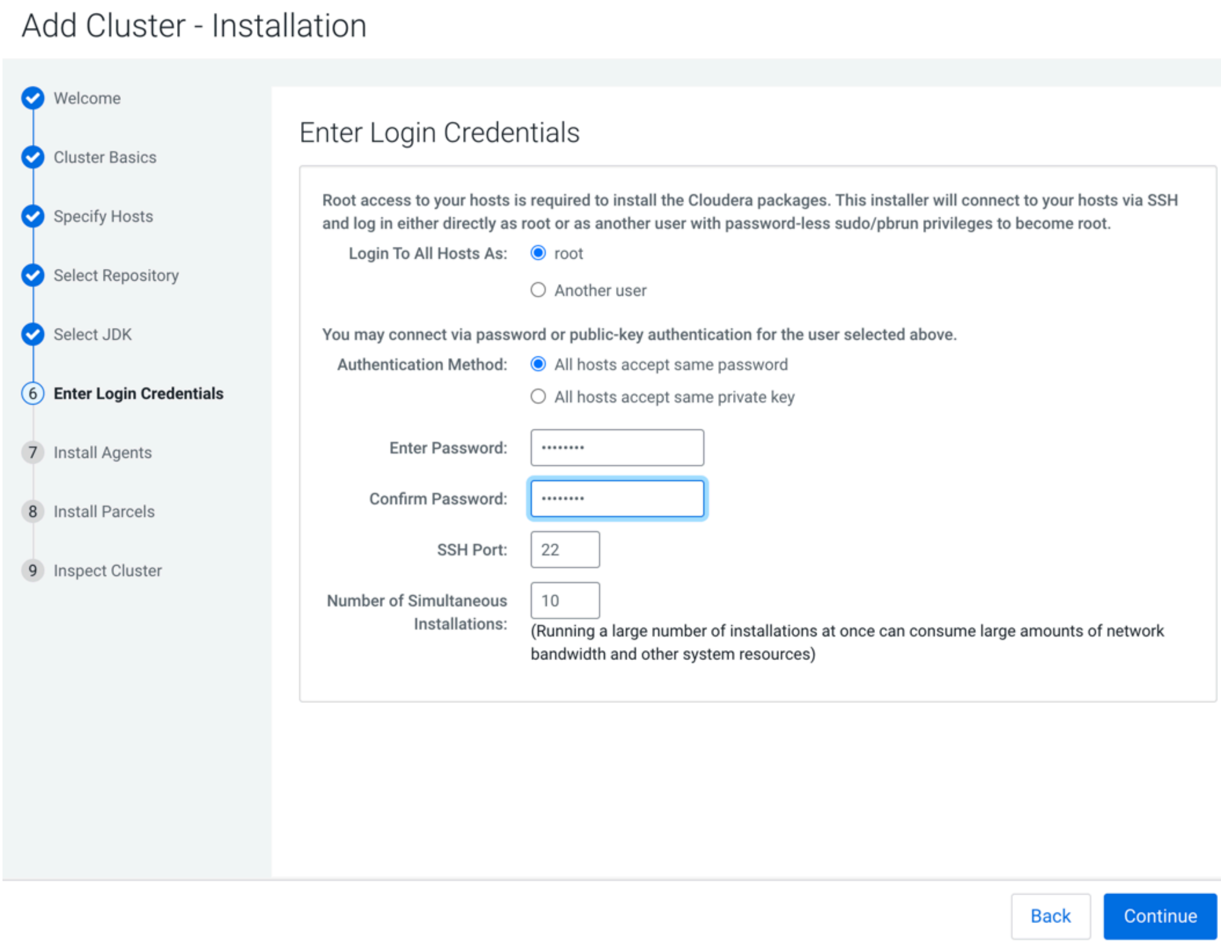


13. Click **Continue**.

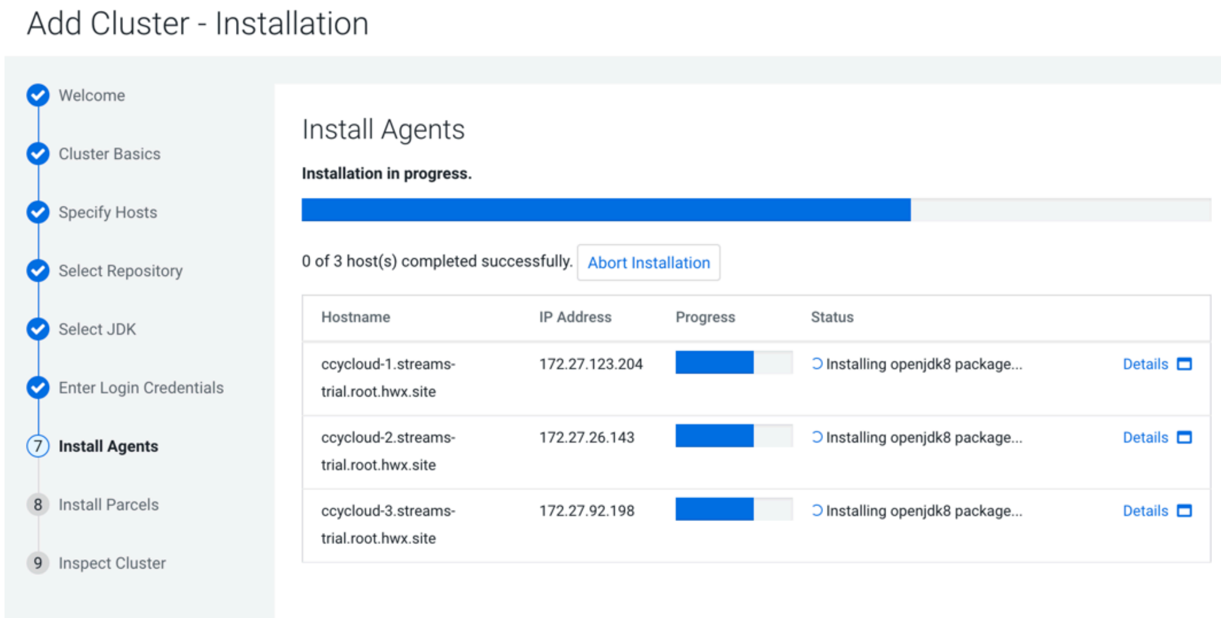
The **Enter Login Credentials** section appears.

14. Do the following:

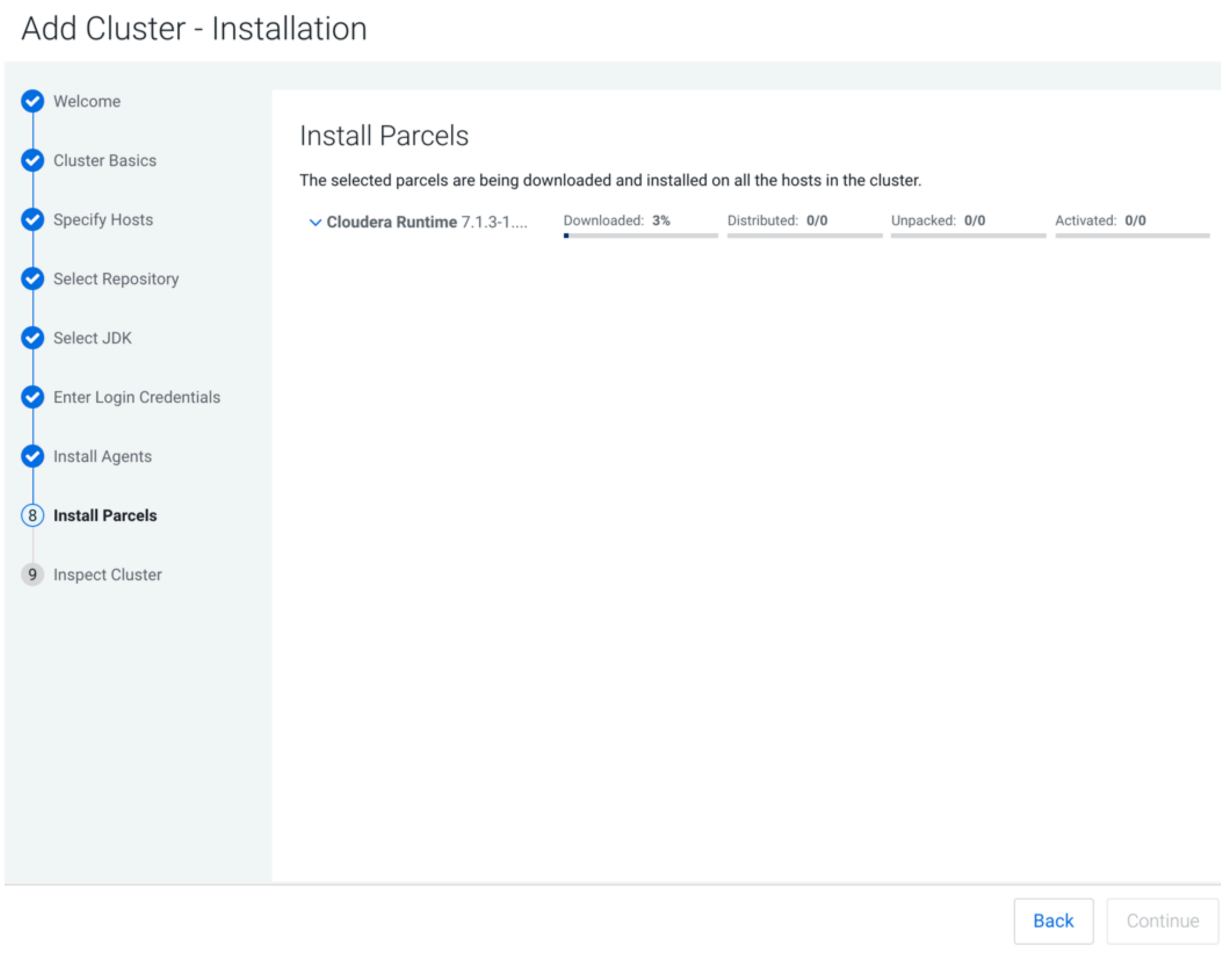
- Select root.
- Select All hosts accept same password.
- Enter the password for the account that allows root access to your hosts.
- Click Continue.



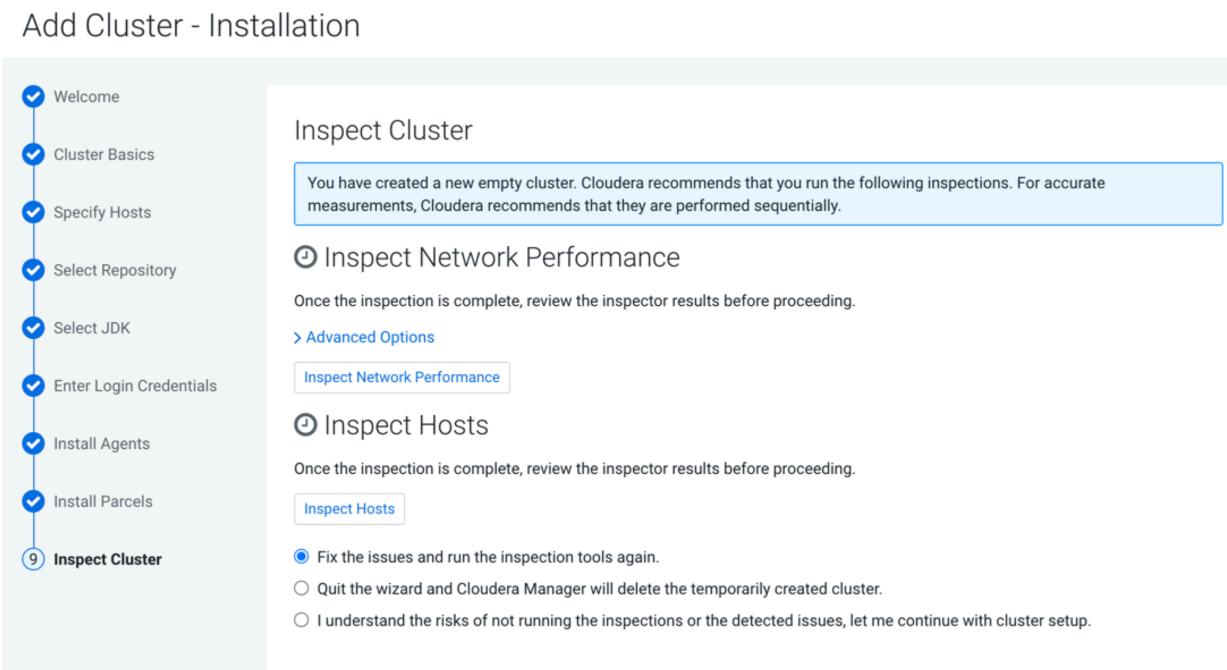
The **Install Agents** section appears showing the progress of the installation.



After the agents are installed, the **Install Parcels** section appears showing the progress of the parcel installation.



After the parcels are installed the **Inspect Cluster** section appears.


Add Cluster - Installation
Welcome
Cluster Basics
Specify Hosts
Select Repository
Select JDK
Enter Login Credentials
Install Agents
Install Parcels
9 Inspect Cluster
Inspect Cluster
You have created a new empty cluster. Cloudera recommends that you run the following inspections. For accurate measurements, Cloudera recommends that they are performed sequentially.
Inspect Network Performance
Once the inspection is complete, review the inspector results before proceeding.
Advanced Options
Inspect Network Performance
Inspect Hosts
Once the inspection is complete, review the inspector results before proceeding.
Inspect Hosts
Fix the issues and run the inspection tools again.
Quit the wizard and Cloudera Manager will delete the temporarily created cluster.
I understand the risks of not running the inspections or the detected issues, let me continue with cluster setup.

15. Do the following:

a) Select Inspect Network Performance.

You can click Advanced Options to customize some ping parameters.

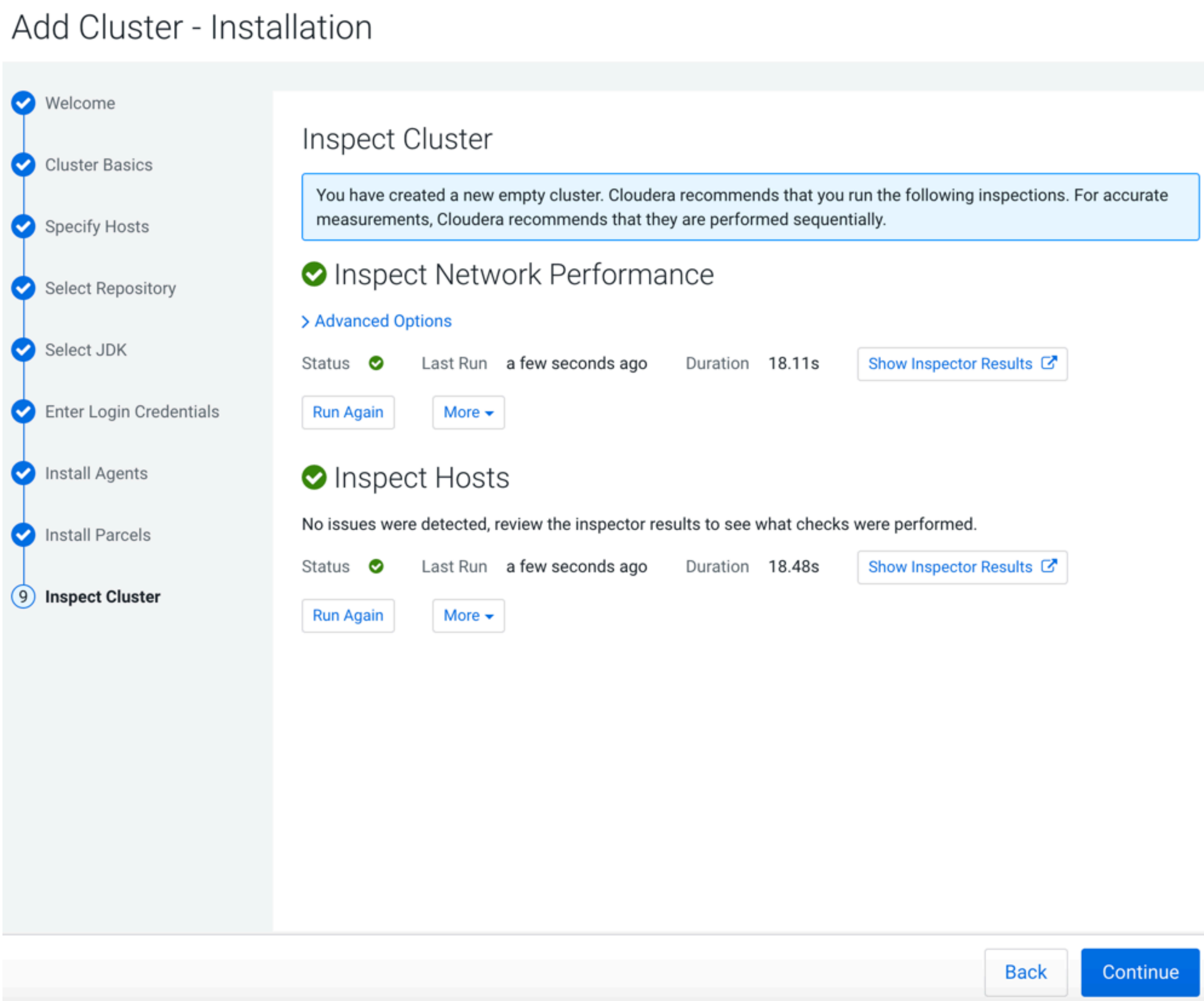
b) After the network inspector completes, click Show Inspector Results to view the results in a new tab.

Address any reported issues, and click Run Again.

c) Click Inspect Hosts.

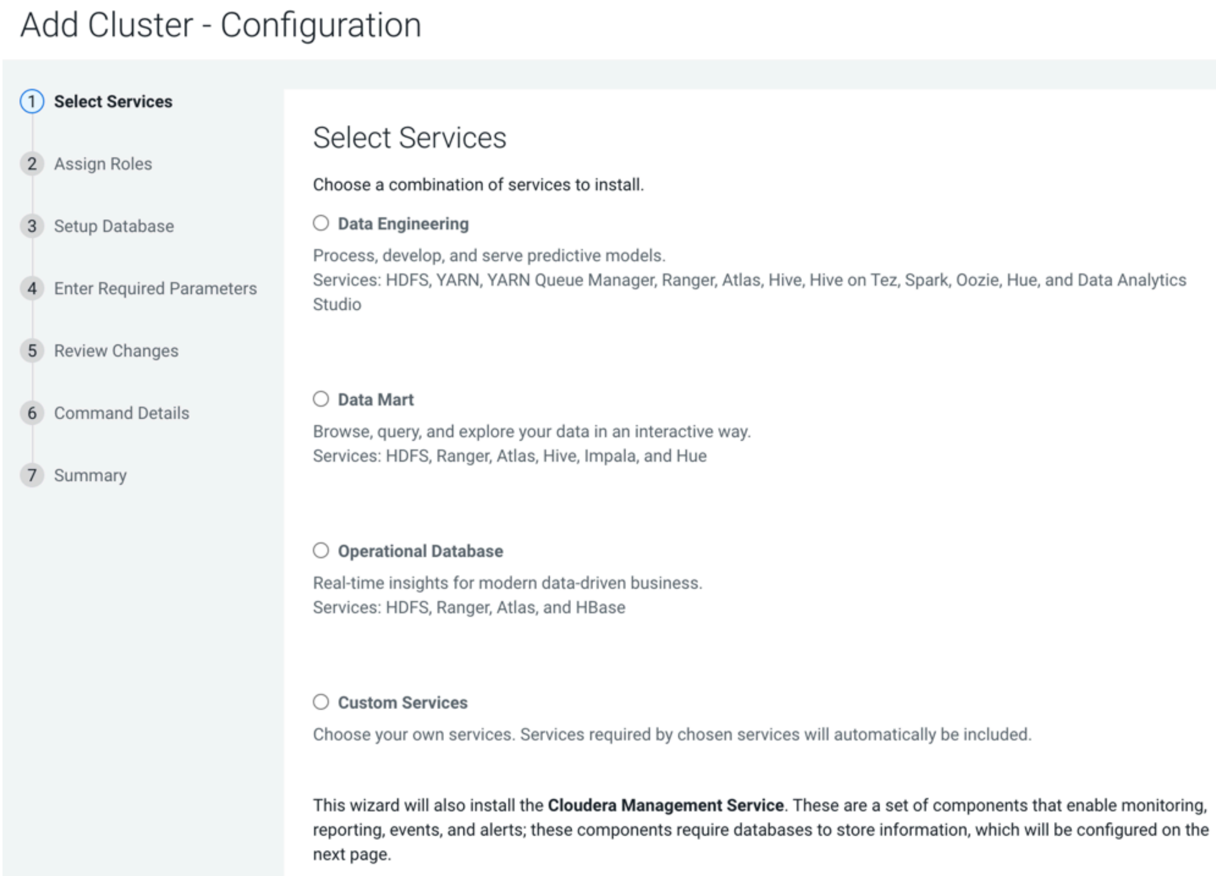
d) After the host inspector completes, click Show Inspector Results to view the results in a new tab.

Address any reported issues, and click Run Again.



16. Click Continue.

The **Add Cluster - Configuration** page appears.



Results

This completes the **Add Cluster - Installation** wizard.

What to do next

Set up a cluster.

Set Up a Cluster Using the Wizard

After completing the Cluster Installation wizard, the Cluster Setup wizard automatically starts. The following sections guide you through each page of the wizard.

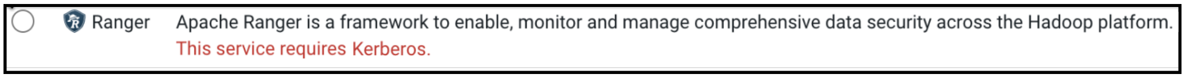
Select Services

The Select Services page allows you to select the services you want to install and configure. Make sure that you have the appropriate license key for the services you want to use.

Important:

In case you deploy a scratch cluster that has not yet been kerberized, you cannot choose any service that requires Kerberos as a pre-requisite, including Ranger, RangerKMS, Atlas, and Knox.

Each of these services will carry such note near it (see example for Ranger):



Choosing either of these services to be installed without having a Kerberos-based cluster **will lead to a non-functioning cluster**. It is advised to install these services separately only after enabling Kerberos on the cluster. For more information, see Set up a KDC.

After selecting the services you want to add, click Continue. The Assign Roles page displays.

Note that if you did not add Kerberos to the cluster at the beginning of the Wizard (i.e. setting Cloudera Manager as Key Distribution Center), you will need to choose only Customer Services option and pick the non-kerberos services you want to deploy, and only post enabling Kerberos, you will repeat the Wizard and complete the Kerberos based services.

You can choose from:

Regular (Base) Clusters

Data Engineering

Process develop, and serve predictive models.

Services included: HDFS, YARN, YARN Queue Manager, Ranger, Atlas, Hive, Hive on Tez, Spark, Oozie, and Hue

Data Mart

Browse, query, and explore your data in an interactive way.

Services included: HDFS, Ranger, Atlas, Hive, and Hue

Operational Database

Real-time insights for modern data-driven business.

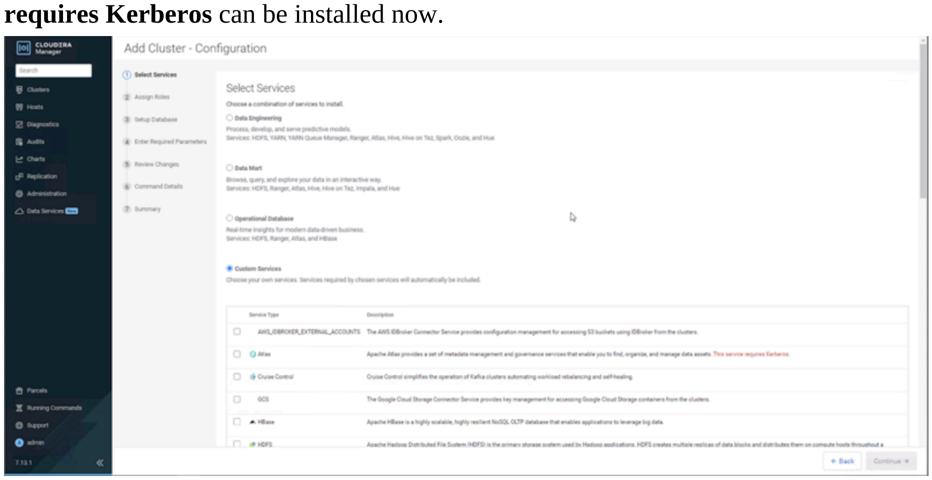
Services included: HDFS, Ranger, Atlas, and HBase

Custom Services

Choose your own services. Services required by chosen services will automatically be included.

Note:

In most cases, you will choose Custom Services and will pick the services that were purchased by the customer. Remember that if this is a scratch cluster installation and KDC/TLS were not installed, any service marked with a red note stating **This service requires Kerberos** can be installed now.



After picking the services you would like to deploy, for each service you will be asked to choose the nodes on which the service will be deployed. Assign the service as instructed by Cloudera according to the Cloudera pre-defined role distribution design table.

Note:

Though the Wizard installs the services chosen, some functions that are required for these services are not installed during the Wizard's initial installation due to dependencies and will need to be added later manually, such as service GateWays for HBASE, HDFS, Solr, and Yarn. To add them manually later, once the cluster is up, simply go to the service instances page, choose to add a role instance, pick the Gateway instance, and assign it according to the Cloudera pre-defined role distribution design table to the correct nodes.

After selecting the services you want to add, click Continue. The Assign Roles page displays.

Assign Roles

The Assign Roles page suggests role assignments for the hosts in your cluster. You can click on the hostname for a role to select a different host. You can also click the View By Host button to see all the roles assigned to a host.

To review the recommended role assignments, see *Recommended Cluster Hosts and Role Distribution*.

After assigning all of the roles for your services, click Continue. The Setup Database page displays.

Setup Database

When using the Cloudera Manager installer with the embedded database, the Setup Database page is pre-populated with the database names and passwords. Click Test Connection to validate the settings. If the connection is successful, a green checkmark and the word Successful appears next to each service. If there are any problems, the error is reported next to the service that failed to connect. Some databases will be created in a future step. For these, the words Skipped. Cloudera Manager will create this database in a later step. appear next to the green checkmark.

After verifying that each connection is successful, click Continue. The Review Changes page displays.

Enter Required Parameters

The **Enter Required Parameters** page lists required parameters for the Cloudera Manager API client, and Ranger.

If you do not have an existing user for the Cloudera Manager API client, use the default username and password "admin" for both the The Existing Cloudera Manager API Client Username and The Existing Cloudera Manager API Client Password.

The Ranger Admin user, Usersync user, Tagsync User, and KMS Keyadmin User are created during cluster deployment. In this page you must give a password for each of these users.

Note: Passwords for the Ranger Admin, Usersync, Tagsync, and KMS Keyadmin users must be a minimum of 8 characters long, with at least one alphabetic and one numeric character. The following characters are not valid: " ' \ ` ´.

The Ranger database host, name, user, and user password were configured when you created the required Ranger database. If you ran the gen_embedded_ranger_db.sh script to create the Ranger database, the output of the script contained the host and database user password. Enter those here. The default database name is "ranger" and the default database user is "rangeradmin."

Review Changes

The Review Changes page lists default and suggested settings for several configuration parameters, including data directories.

Warning: Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which results in missing blocks.

Review and make any necessary changes, and then click Continue. The Command Details page displays.

Command Details

The Command Details page lists the details of the First Run command. You can expand the running commands to view the details of any step, including log files and command output. You can filter the view by selecting Show All Steps, Show Only Failed Steps, or Show Running Steps.

After the First Run command completes, click Continue to go to the Summary page.

Summary

The Summary page reports the success or failure of the setup wizard. Click Finish to complete the wizard. The installation is complete.

Cloudera recommends that you change the default password as soon as possible by clicking the logged-in username at the top right of the home screen and clicking Change Password.

Stopping the Embedded PostgreSQL Database

To stop the embedded PostgreSQL database, stop the services and servers in the order listed below.

Procedure

1. Log into the Cloudera Manager user interface and stop the services that have a dependency on the Hive metastore (Hue, Impala, and Hive) in the following order:
   - Stop the Hue and Impala services.
   - Stop the Hive service.
2. On the Cloudera Manager **Home** page, click the 3 vertical dots next to Cloudera Management Service and select Stop to stop the Cloudera Management Service.
3. Stop the Cloudera Manager Server.

   RHEL 7:

   ```
   sudo systemctl stop cloudera-scm-server.service
   ```
4. Stop the Cloudera Manager Server database.

   RHEL 7:

   ```
   sudo systemctl stop cloudera-scm-server-db.service
   ```

Starting the Embedded PostgreSQL Database

To start the embedded PostgreSQL database, start the servers and services in the order listed below.

Procedure

1. Start the Cloudera Manager Server database.

   RHEL 7:

   ```
   sudo systemctl start cloudera-scm-server-db.service
   ```

2. Start the Cloudera Manager Server.

   RHEL 7:

```
sudo systemctl start cloudera-scm-server.service
```

3. Log into Cloudera Manager and start the Cloudera Manager Service. On the Cloudera Manager **Home** page, click the 3 vertical dots next to Cloudera Management Service and select Start.
4. In the Cloudera Manager user interface, start the services that have a dependency on the Hive metastore (Hue, Impala, and Hive) in the following order:
   - Start the Hive service.
   - Start the Hue and Impala services.

Changing Embedded PostgreSQL Database Passwords

The embedded PostgreSQL database has generated user accounts and passwords. You can change a password associated PostgreSQL database account.

About this task

You can see the generated accounts and passwords during the installation process and you should record them at that time.

To find information about the PostgreSQL database account that the Cloudera Manager Server uses, read the /etc/cloudera-scm-server/db.properties file:

```
# cat /etc/cloudera-scm-server/db.properties

Auto-generated by scm_prepare_database.sh
#
Sat Oct 1 12:19:15 PDT 201
#
com.cloudera.cmf.db.type=postgresql
com.cloudera.cmf.db.host=localhost:7432
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=TXqEESuhj5
```

To change a password associated with a PostgreSQL database account:

Procedure

1. Obtain the root password from the /var/lib/cloudera-scm-server-db/data/generated_password.txt file:

```
# cat /var/lib/cloudera-scm-server-db/data/generated_password.txt

MnPwGeWaip

The password above was generated by /usr/share/cmf/bin/initialize_embedded
_db.sh (part of the cloudera-scm-server-db package)
and is the password for the user 'cloudera-scm' for the database in the
current directory.

Generated at Fri Jun 29 16:25:43 PDT 2012.
```

2. On the host on which the Cloudera Manager Server is running, log into PostgreSQL as the root user:

```
psql -U cloudera-scm -p 7432 -h localhost -d postgres
Password for user cloudera-scm: MnPwGeWaip
psql (8.4.18)
```

```
Type "help" for help.

postgres=#
```

3. Determine the database and owner names:

```
postgres=# \l
                                        List of databases
   Name    |    Owner     | Encoding | Collation  |   Ctype    |
 Access privileges
-----------+--------------+----------+------------+------------+-------
---------------------------
 amon      | amon         | UTF8     | en_US.UTF8 | en_US.UTF8 |
 hive      | hive         | UTF8     | en_US.UTF8 | en_US.UTF8 |
 nav       | nav          | UTF8     | en_US.UTF8 | en_US.UTF8 |
 navms     | navms        | UTF8     | en_US.UTF8 | en_US.UTF8 |
 postgres  | cloudera-scm | UTF8     | en_US.UTF8 | en_US.UTF8 |
 rman      | rman         | UTF8     | en_US.UTF8 | en_US.UTF8 |
 scm       | scm          | UTF8     | en_US.UTF8 | en_US.UTF8 |
 template0 | cloudera-scm | UTF8     | en_US.UTF8 | en_US.UTF8 | =c/"clou
dera-scm"
                                                                : "cloudera
-scm"=CTc/"cloudera-scm"
 template1 | cloudera-scm | UTF8     | en_US.UTF8 | en_US.UTF8 | =c/"cl
oudera-scm"
                                                                : "cloude
ra-scm"=CTc/"cloudera-scm"
(9 rows)
```

4. Set the password for an owner using the \password command. For example, to set the password for the amon owner, do the following:

```
postgres=# \password amon
Enter new password:
Enter it again:
```

5. Configure the role with the new password:
   a) In the Cloudera Manager Admin Console, select ClustersCloudera Management Service.
   b) Click the Configuration tab.
   c) In the Scope section, select the role where you are configuring the database.
   d) Select CategoryDatabase category.
   e) Set the *ROLE NAME* Database Password property.
   f) Enter a Reason for change, and then click Save Changes to commit the changes.

Migrating from the Cloudera Manager Embedded PostgreSQL Database Server to an External PostgreSQL Database

If you have already used the embedded PostgreSQL database and you are unable to redeploy a fresh cluster, you must migrate the embedded PostgreSQL database sever to an external PostgreSQL database.

Cloudera Manager provides an embedded PostgreSQL database server for trial and proof of concept deployments when creating a cluster. To remind users that this embedded database is not suitable for production, Cloudera Manager displays the banner text: "You are running Cloudera Manager in non-production mode, which uses an embedded PostgreSQL database. Switch to using a supported external database before moving into production."

If, however, you have already used the embedded database, and you are unable to redeploy a fresh cluster, then you must migrate to an external PostgreSQL database.

Note: This procedure does not describe how to migrate to a database server other than PostgreSQL. Moving databases from one database server to a different type of database server is a complex process that requires modification of the schema and matching the data in the database tables to the new schema. It is strongly recommended that you engage with Cloudera Professional Services if you wish to perform a migration to an external database server other than PostgreSQL.

Prerequisites

Before migrating the Cloudera Manager embedded PostgreSQL database to an external PostgreSQL database, ensure that your setup meets the following conditions:

- The external PostgreSQL database server is running.
- The database server is configured to accept remote connections.
- The database server is configured to accept user logins using md5.
- No one has manually created any databases in the external database server for roles that will be migrated.

  Note: To view a list of databases in the external database server (requires default superuser permission):

  ```
  sudo -u postgres psql -l
  ```
- All health issues with your cluster have been resolved.

For details about configuring the database server, see *Configuring and Starting the PostgreSQL Server*.

Important: Only perform the steps in *Configuring and Starting the PostgreSQL Server*. Do not proceed with the creation of databases as described in the subsequent section.

For large clusters, Cloudera recommends running your database server on a dedicated host. Engage Cloudera Professional Services or a certified database administrator to correctly tune your external database server.

Identify Roles that Use the Embedded Database Server

Before you can migrate to another database server, you must first identify the databases using the embedded database server.

About this task

When the Cloudera Manager Embedded Database server is initialized, it creates the Cloudera Manager database and databases for roles in the Management Services. The Installation Wizard (which runs automatically the first time you log in to Cloudera Manager) or Add Service action for a cluster creates additional databases for roles when run. It is in this context that you identify which roles are used in the embedded database server.

To identify which roles are using the Cloudera Manager embedded database server:

Procedure

1. Obtain and save the cloudera-scm superuser password from the embedded database server. You will need this password in subsequent steps:

   ```
   head -1  /var/lib/cloudera-scm-server-db/data/generated_password.txt
   ```
2. Make a list of all services that are using the embedded database server. Then, after determining which services are not using the embedded database server, remove those services from the list. The scm database must remain in your list. Use the following table as a guide:

   Table 16: Cloudera Manager Embedded Database Server Databases

Service	Role	Default Database Name	Default Username
Cloudera Manager Server		scm	scm

Service	Role	Default Database Name	Default Username
Cloudera Management Service	Activity Monitor	amon	amon
Hive	Hive Metastore Server	hive	hive
Hue	Hue Server	hue	7uu7uu7uhue
Oozie	Oozie Server	oozie_oozie_server	oozie_oozie_server
Cloudera Management Service	Reports Manager	rman	rman
Ranger		ranger	rangeradmin

3. Verify which roles are using the embedded database. Roles using the embedded database server always use port 7432 (the default port for the embedded database) on the Cloudera Manager Server host.

   a. Verify which roles are using the embedded database. Roles using the embedded database server always use port 7432 (the default port for the embedded database) on the Cloudera Manager Server host.

      For Cloudera Management Services:

      1. Select Cloudera Management Service > Configuration, and type "7432" in the Search field.
      2. Confirm that the hostname for the services being used is the same hostname used by the Cloudera Manager Server.

      Note:

      If any of the following fields contain the value "7432", then the service is using the embedded database:

      - Activity Monitor
      - Reports Manager

      For the Oozie Service:

      1. Select Oozie service > Configuration, and type "7432" in the Search field.
      2. Confirm that the hostname is the Cloudera Manager Server.

      For Hive and Hue Services:

      1. Select the specific service > Configuration, and type "database host" in the Search field.
      2. Confirm that the hostname is the Cloudera Manager Server.
      3. In the Search field, type “database port” and confirm that the port is 7432.
      4. Repeat these steps for each of the services (Hive and Hue).

4. Verify the database names in the embedded database server match the database names on your list (Step 2). Databases that exist on the database server and not used by their roles do not need to be migrated. This step is to confirm that your list is correct.

   Note: Do not add the postgres, template0, or template1 databases to your list. These are used only by the PostgreSQL server.

```
psql -h localhost -p 7432 -U cloudera-scm -l
```

```
Password for user cloudera-scm: <password>

                                                List of databases
      Name          |        Owner         | Encoding |  Collate   |   Ctype
    |         Access
--------------------+--------------------+----------+-----------+-------
-----+----------------
amon                | amon                 | UTF8     | en_US.UTF8 | en_US.U
TF8 |
```

```
hive                  | hive                  | UTF8     | en_US.UTF8 | en_US.UT
F8 |
hue                   | hue                   | UTF8     | en_US.UTF8 | en_US
.UTF8 |
navms                 | navms                 | UTF8     | en_US.UTF8 | en_US.
UTF8 |
oozie_oozie_server | oozie_oozie_server | UTF8     | en_US.UTF8 | en_US.U
TF8 |
postgres              | cloudera-scm          | UTF8     | en_US.UTF8 | en_US.UT
F8 |
rman                  | rman                  | UTF8     | en_US.UTF8 | en_US
.UTF8 |
scm                   | scm                   | UTF8     | en_US.UTF8 | en_US.
UTF8 |
template0             | cloudera-scm          | UTF8     | en_US.UTF8 | en_US.U
TF8 | =c/"cloudera-scm"
template1             | cloudera-scm          | UTF8     | en_US.UTF8 | en_US.
UTF8 | =c/"cloudera-scm"
(12 rows)
```

Results

You should now have a list of all roles and database names that use the embedded database server, and are ready to proceed with the migration of databases from the embedded database server to the external PostgreSQL database server.

Migrate Databases from the Embedded Database Server to the External PostgreSQL Database Server

After you identify the roles that use the embedded database, you are ready to migrate from the embedded database server to an external PostgreSQL database server.

About this task

While performing this procedure, ensure that the Cloudera Manager Agents remain running on all hosts. Unless otherwise specified, when prompted for a password use the cloudera-scm password.

Note: After completing this migration, you cannot delete the cloudera-scm postgres superuser unless you remove the access privileges for the migrated databases. Minimally, you should change the cloudera-scm p ostgres superuser password.

Procedure

1. In Cloudera Manager, stop the cluster services identified as using the embedded database server. Be sure to stop the Cloudera Management Service as well. Also be sure to stop any services with dependencies on these services. The remaining Runtime services will continue to run without downtime.

   Note: If you do not stop the services from within Cloudera Manager before stopping Cloudera Manager Server from the command line, they will continue to run and maintain a network connection to the embedded database server. If this occurs, then the embedded database server will ignore any command line stop commands (Step 2) and require that you manually stop the process, which in turn causes the services to crash instead of stopping cleanly.
2. Navigate to Hosts > All Hosts, and make note of the number of roles assigned to hosts. Also take note whether or not they are in a commissioned state. You will need this information later to validate that your scm database was migrated correctly.
3. Stop the Cloudera Manager Server. To stop the server:

```
sudo service cloudera-scm-server stop
```

4. Obtain and save the embedded database superuser password (you will need this password in subsequent steps) from the generated_password.txt file:

```
head -1  /var/lib/cloudera-scm-server-db/data/generated_password.txt
```

5. Export the PostgreSQL user roles from the embedded database server to ensure the correct users, permissions, and passwords are preserved for database access. Passwords are exported as an md5sum and are not visible in plain text. To export the database user roles (you will need the cloudera-scm user password):

```
pg_dumpall -h localhost -p 7432 -U cloudera-scm -v --roles-only -f "/var/
tmp/cloudera_user_roles.sql"
```

6. Edit the /var/tmp/cloudera_user_roles.sql file to remove any CREATE ROLE and ALTER ROLE commands for databases not in your list. Leave the entries for the cloudera-scm user untouched, because this user role is used during the database import.

   Important: If the external PostgresSQL database is an Amazon's Relational Database Service (RDS), then remove all entries for ALTER ROLE or CREATE ROLE commands from the /var/tmp/cloudera_user_roles.sql file for the Cloudera Manager database's user such as cloudera-scm, and then add the following command for the same user:

```
CREATE ROLE cloudera-scm WITH NOSUPERUSER INHERIT NOCREATEROLE NOCRE
ATEDB LOGIN NOREPLICATION NOBYPASSRLS PASSWORD '<stripped>';
```

7. Export the data from each of the databases on your list you created in *Identify Roles that Use the Embedded Database Server*:

```
pg_dump -F c -h localhost -p 7432 -U cloudera-scm [database_name]  > /var/
tmp/[database_name]_db_backup-$(date +"%m-%d-%Y").dump
```

   The following is a sample data export command for the scm database:

```
pg_dump -F c -h localhost -p 7432 -U cloudera-scm scm > /var/tmp/scm_db_
backup-$(date +%m-%d-%Y).dump
```

```
Password:
```

8. Stop and disable the embedded database server:

```
service cloudera-scm-server-db stop
chkconfig cloudera-scm-server-db off
```

   Confirm that the embedded database server is stopped:

```
netstat -at | grep 7432
```

9. Back up the Cloudera Manager Server database configuration file:

```
cp /etc/cloudera-scm-server/db.properties /etc/cloudera-scm-server/db.pr
operties.embedded
```

10. Copy the file /var/tmp/cloudera_user_roles.sql and the database dump files from the embedded database server host to /var/tmp on the external database server host:

```
cd /var/tmp
scp cloudera_user_roles.sql *.dump <user>@<postgres-server>:/var/tmp
```

11. Import the PostgreSQL user roles into the external database server.

The external PostgreSQL database server superuser password is required to import the user roles. If the superuser role has been changed, you will be prompted for the username and password.

Note: Only run the command that applies to your context; do not run both commands.

- To import users when using the default PostgreSQL superuser role:

```
sudo -u postgres psql -f /var/tmp/cloudera_user_roles.sql
```

- To import users when the superuser role has been changed:

```
psql -h <database-hostname> -p <database-port> -U <superuser> -f /var/tm
p/cloudera_user_roles.sql
```

For example:

```
psql -h pg-server.example.com -p 5432 -U postgres -f /var/tmp/cloudera_u
ser_roles.sql
```

```
Password for user postgres
```

12. Import the Cloudera Manager database on the external server. First copy the database dump files from the Cloudera Manager Server host to your external PostgreSQL database server, and then import the database data:

Note: To successfully run the pg_restore command, there must be an existing database on the database server to complete the connection; the existing database will not be modified. If the -d <existing-databas e> option is not included, then the pg_restore command will fail.

```
pg_restore -C -h <database-hostname> -p <database-port> -d <existing-dat
abase> -U cloudera-scm -v <data-file>
```

Repeat this import for each database.

The following example is for the scm database:

```
pg_restore -C -h pg-server.example.com -p 5432 -d postgres -U cloudera-scm
 -v /var/tmp/scm_server_db_backup-20180312.dump
```

```
pg_restore: connecting to database for restore
Password:
```

13. Update the Cloudera Manager Server database configuration file to use the external database server. Edit the /etc/ cloudera-scm-server/db.properties file as follows:

a) Update the com.cloudera.cmf.db.host value with the hostname and port number of the external database server.
b) Change the com.cloudera.cmf.db.setupType value from "EMBEDDED" to "EXTERNAL".

14. Start the Cloudera Manager Server and confirm it is working:

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

Note that if you start the Cloudera Manager GUI at this point, it may take up to five minutes after executing the start command before it becomes available.

In Cloudera Manager Server, navigate to Hosts > All Hosts and confirm the number of roles assigned to hosts (this number should match what you found in Step 2); also confirm that they are in a commissioned state that matches what you observed in Step 2.

15. Update the role configurations to use the external database hostname and port number. Only perform this task for services where the database has been migrated.

For Cloudera Management Services:

a. Select Cloudera Management Service > Configuration, and type "7432" in the Search field.
b. Change any database hostname properties from the embedded database to the external database hostname and port number.
c. Click Save Changes.

For the Oozie Service:

a. Select Oozie service > Configuration, and type "7432" in the Search field.
b. Change any database hostname properties from the embedded database to the external database hostname and port number.
c. Click Save Changes.

For Hive and Hue Services:

a. Select the specific service > Configuration, and type "database host" in the Search field.
b. Change the hostname from the embedded database name to the external database hostname.
c. Click Save Changes.

16. Start the Cloudera Management Service and confirm that all management services are up and no health tests are failing.

17. Start all Services via the Cloudera Manager web UI. This should start all services that were stopped for the database migration. Confirm that all services are up and no health tests are failing.

18. On the embedded database server host, remove the embedded PostgreSQL database server:

a) Make a backup of the /var/lib/cloudera-scm-server-db/data directory:

```
tar czvf /var/tmp/embedded_db_data_backup-$(date +”%m-%d-%Y”).tgz /var/l
ib/cloudera-scm-server-db/data
```

b) Remove the embedded database package:

For RHEL/SLES:

```
rpm --erase cloudera-manager-server-db-2
```

For Ubuntu:

```
apt-get remove cloudera-manager-server-db-2
```

c) Delete the /var/lib/cloudera-scm-server-db/data directory.

Installing and Configuring CDP with FIPS

This guide provides instructions for installing and configuring FIPS encryption on CDP.

Overview

This topic provides important background information about CDP running in FIPS-compliant mode via the use of FIPS 140-2 validated cryptographic modules.

The Federal Information Processing Standards (FIPS) publications are publicly available standards and guidelines jointly developed by the US government and industry, and issued by the National Institute of Standards and Technology (NIST) for use in information systems by Federal government agencies and government contractors.

Within the FIPS Publications, the FIPS Publication 140-2 (FIPS 140-2) is the standard for encryption modules. FIPS 140-2 specifies the security requirements that must be satisfied by a cryptographic module utilized within a security system protecting sensitive information. To ensure conformance with the FIPS 140-2 standard, cryptographic modules must first be validated as conforming to the FIPS 140-2 standard using the Cryptographic Module Validation Program (CMVP). Through the CMVP, validation of a cryptographic module for FIPS 140-2 conformance is conducted by independent Cryptographic and Security Testing (CST) laboratories that have been accredited by the National Voluntary Laboratory Accreditation Program (NVLAP). Following validation, cryptographic modules are issued validation certificates, and these modules can then be deployed by Federal agencies as part of information systems that require the protection of sensitive information. For additional details on the FIPS 140-2 Publication and standard, see Security Requirements for Cryptographic Modules.

While the FIPS 140-2 standard has been broadly adopted, in the United States the FIPS 140-2 standard is leveraged in several government security compliance and accreditation mandates and frameworks, including FISMA, FedRAMP, and DISA Security Technical Implementation Guides (STIG). Within many of these compliance and accreditation mandates, it is specified that FIPS validated cryptography must be used at a minimum when encryption is employed within an information system. These mandates also specify that FIPS-validated cryptography should be used in a compliant manner consistent with the standard and mandates. As a result, system operators are typically required to:

1. Enable FIPS mode within an operating system (OS) used in the information system (e.g., RHEL and CentOS), to ensure that the OS is configured to be compliant with the FIPS 140-2 standard, and to enforce the use of FIPS-approved keystores, algorithms, and strengths with FIPS 140-2 validated cryptographic modules.
2. With respect to the application running on top of the OS, ensure that FIPS 140-2 validated cryptographic modules are used with the application. In addition, configure the application to be compliant with the FIPS 140-2 standard as well as the government security compliance and applicable accreditation mandate or framework. This typically includes employing the use of FIPS-approved keystores and algorithms with FIPS 140-2 validated cryptographic modules, as well as employing strong authentication, authorization, audit, data governance, in-transient data encryption, and at-rest data encryption features.

Beginning with CDP Private Cloud Base version 7.1.5, CDP Private Cloud Base can be configured to operate in a FIPS-compliant mode by leveraging FIPS 140-2 validated cryptographic modules. Cloudera has accomplished this by supporting the deployment of the CDP Private Cloud Base platform on a FIPS-mode enabled Operating System (e.g., RHEL or CentOS) that is integrated with FIPS 140-2 validated cryptography modules.

SafeLogic provides Cloudera with supported FIPS 140-2 validated modules that have been approved through the NIST CMVP. These modules replace JCE (Java Cryptographic Extension) providers (such as Bouncy Castle), OpenSSL, NSS, and Libgcrypt cryptographic libraries. The SafeLogic CryptoComply modules are separately licensed as a bundle and can be acquired through a request to your Cloudera Account Team. The SafeLogic CryptoComply modules are as follows:

- CryptoComply for Server (CCS) - OpenSSL RPMs
- CryptoComply for Java (CCJ) - Java Cryptography Extension JAR
- CryptoComply for Libgcrypt RPMs

Cloudera has integrated these FIPS-validated libraries with the CDP Private Cloud Base platform through installation and runtime configuration of the CDP Private Cloud Base platform. You must install the SafeLogic CryptoComply modules on the applicable operating system, configured in FIPS mode, and then configure CDP Private Cloud Base to use these modules in a FIPS-compliant manner.

Beginning with version 7.1.5, CDP Private Cloud Base is capable of running in a FIPS-compliant mode. Cloudera currently supports a subset of platform components and features running on a FIPS-compliant operating system. Cloudera does not guarantee that the platform components themselves are FIPS 140-2 compliant. However, future releases of CDP Private Cloud Base will replace non-compliant platform components and features with fully compliant FIPS 140-2 implementations.

Given that the use of CDP Private Cloud Base security features within regulated government environments is commonplace, these features should be configurable to be compliant with the FIPS 140-2 standard, as well as the applicable government security compliance and accreditation mandate or framework. This typically includes the use of FIPS-approved keystores and algorithms with FIPS 140-2 validated cryptographic modules, strong authentication, authorization, audit, data governance, in-transient data encryption, and at-rest data encryption features. Cloudera recommends the use of the following components as described in the installation documentation:

- Encryption in-motion using TLS (Auto-TLS support).
- Encryption at-rest with HDFS Transparent Data Encryption (TDE), Ranger KMS, and Key Trustee Server as the backend keystore.
- Strong authentication with Kerberos and Apache Knox.
- Authorization, audit, and data governance with Apache Ranger and Apache Atlas.

CDP Private Cloud Base with FIPS-compliant mode can only be configured during new cluster installations. Converting an existing CDP Base cluster from non-FIPS mode to FIPS-compliant mode is not supported. However, FIPS-enabled CDP Base clusters can be upgraded between versions, provided both the source and target versions support FIPS. Before upgrading, verify FIPS support for the target CDP and Cloudera Manager versions using the Cloudera support matrix.

Cloudera is not responsible for providing instructions for enabling FIPS mode on a RHEL or CentOS-based operating system, or instructions on configuring the required external databases in a FIPS 140-2 compliant manner. Please consult the vendor documentation for your database for details.

The supported platform components and features and all limitations with this release are documented in the Prerequisites for using FIPS on page 77 section.

In summary, the ability to run CDP Private Cloud Base in a FIPS 140-2 compliant mode allows CDP Private Cloud Base FIPS platform customers to improve conformance with their compliance and accreditation standards within their information systems..

Prerequisites for using FIPS

This page provides comprehensive information regarding prerequisites that you must be aware of while using FIPS for CDP.

About CDP with FIPS

Creating a new, fresh cluster is the only way to enable or disable FIPS.

Known Issues

Important: If you are using RHEL 8.8 in FIPS mode, then before you initiate the Cloudera Manager cluster installation or upgrade process, ensure to add the symbolic links to the OpenSSL libraries on all the cluster hosts. For information about workaround, see Known Issues in Cloudera Manager 7.11.3.

See the Cloudera Manager release notes.

Unsupported Features

- Upgrades are not currently supported to or from CDP with FIPS.

  Note: Upgrades that do not change FIPS status, i.e., CDP FIPS to CDP FIPS, are supported.

- Replication is not currently supported.
- MRIT localfs is not supported in FIPS environments where SHA2 compatibility is required.
- If you are upgrading to Cloudera Runtime 7.1.8 from 7.1.7 SP1 + FIPS, the package manager enabled version does not support FIPS and many Runtime components may not start. To overcome this scenario, you must revert back to the original SafeLogic openssl-devel-y package.

System Requirements

- **Operating system:**

  RHEL/Centos 7.9 or RHEL 8.8. For more information, see Operating system requirements.

- **Java:**

  Support for FIPS + OpenJDK 8 or Oracle JDK 8 (**started from Cloudera Runtime 7.1.9 and Cloudera Manager 7.11.3 release onwards**).

  Support for FIPS + OpenJDK 11 or Oracle JDK 11 (**started from Cloudera Runtime 7.1.9 CHF3 and Cloudera Manager 7.11.3 CHF3 release onwards**).

  Support for FIPS + OpenJDK 17 (**started from Cloudera Runtime 7.1.9 SP1 and Cloudera Manager 7.11.3 CHF7 release onwards**). For more information, see Java requirements.
- OpenJDK 8 versions: For FIPS the only required and tested versions are 1.8u231 and 1.8u232.
- Install and configure a database. See Step 3. Install and Configure Databases on page 150

  Important:

  The minimum supported database for CDP Private Cloud Base 7.1.9 in FIPS mode starts from PostgresSQL 10, 11, 12, 13, and 14 versions.

  The minimum supported database for CDP Private Cloud Base 7.1.9 SP1 in FIPS mode starts from PostgresSQL 10, 13, 14, 15, 16 (11 and 12 version support was dropped, and 15, 16 were introduced) versions.

  (The embedded version of PostgreSQL is not supported).

  Caution: Cloudera requires disabling the fapolicyd daemon present in RHEL 8 (and later) systems before beginning installation of Cloudera Manager application. Be informed that fapolicyd is a user space daemon that determines access rights to files based on attributes of the process and file. It can be used to either blacklist or whitelist processes or file access. Proceed with caution with enforcing the use of this daemon. Improper configuration may render the system non-functional.

Supported CDP Versions

- Cloudera Manager versions 7.2.4, 7.3.1, 7.4.4, 7.6.1, 7.7.1, 7.7.3, and 7.11.3
- CDP Private Cloud Base versions 7.1.5, 7.1.6, 7.1.7, 7.1.7 SP1, 7.1.7 SP2, 7.1.8, 7.1.9, and 7.1.9 SP1.

Supported CDP Components

The following components are supported in FIPS mode:

- Atlas
- Avro
- Cloudera Manager
- Cruise Control
- Hadoop
- Hadoop Credential Provider
- HDFS
- HBase
- Hive
- Hive-on-Tez
- Hive Meta Store
- Hive Warehouse Connector
- Hue
- Iceberg
- Impala
- Kafka
- Kerberos
- Key Trustee Server
- Knox

- Kudu
- Livy
- MapReduce
- OMID
- Oozie
- Parquet
- Phoenix
- Queue Manager
- Ranger
- Ranger KMS
- Schema Registry
- Streams Messaging Manager
- Streams Replication Manager
- Solr
- Spark
- Sqoop
- Tez
- TLS
- YARN
- Zeppelin

  Note: FIPS support for JDK11 in Zeppelin is available starting from Cloudera Manager 7.11.3 Cumulative hotfix 4

- ZooKeeper

  Note:

  Navigator Encrypt is not FIPS-compliant.

Important:

Please contact your Cloudera Sales or Support team to obtain the CDP PVC Base SafeLogic FIPS modules.

Step 1: Prepare hosts

Procedure

1. Cryptographic operations require entropy to ensure randomness.

   Check the available entropy:

   cat /proc/sys/kernel/random/entropy_avail must be always above 2k

   - In order to keep the entropy high, install the below tools and keep them running:
     - rng-tools - For information about checking available entropy and using rng-tools, see "Entropy Requirements" in Data at Rest Encryption Requirements.
     - haveged - For information about using the haveged entropy daemon, see the haveged documentation.

   Note: Virtual Machines might have insufficient sources of entropy, which can cause extreme slowness of many Cloudera Manager and CDP operations even with rngd and haveged enabled. You might need to take the following additional hypervisor-level measures to provide enough entropy for strong cryptographic functions required for FIPS compliance:

   - Ensure that the Hardware Random Number Generator (Hwrng) is enabled by setting disableHwrng to false
   - Set different entropy sources if necessary

2. Configure the operating system for FIPS.

   See, for RHEL 7:https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/chap-federal_standards_and_regulations.

   See, for RHEL 8: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening#switching-the-system-to-fips-mode_using-the-system-wide-cryptographic-policies
3. On all hosts, run one of the following commands to verify that FIPS mode is enabled:

```
cat /proc/sys/crypto/fips_enabled
sysctl crypto.fips_enabled
```

4. Configure a repository to install Cloudera Manager and other required packages.
   a) On the Cloudera Manager server host, download the repository file for your operating system and version:

   ```
   https://[username]:[password]@archive.cloudera.com/p/cm7/7.4.4/redhat7/y
   um/cloudera-manager.repo
   ```

   b) Open the /etc/yum.repos.d/cloudera-manager.repo file in a text editor and replace the changeme placeholder values with your user name and password.

   ```
   [cloudera-manager]
   name=Cloudera Manager 7.4.4
   baseurl=https://archive.cloudera.com/p/cm7/7.4.4/redhat7/yum/
   gpgkey=https://archive.cloudera.com/p/cm7/7.4.4/redhat7/yum/
   RPM-GPG-KEY-cloudera
   username=changeme
   password=changeme
   gpgcheck=1
   enabled=1
   autorefresh=0
   type=rpm-md
   ```

   c) If your hosts do not have access to https://archive.cloudera.com, you will need to set up a local repository. See Configuring a Local Package Repository on page 136.
5. Manually install OpenJDK 8 / Oracle JDK 8 or OpenJDK 11 / Oracle JDK 11 / OpenJDK 17 (**From Cloudera Runtime 7.1.9 SP1 release onwards**) on all hosts.

   - Installing OpenJDK for CDP Runtime on page 147
   - Installing Oracle JDK for CDP Runtime on page 150

Attention: The next step differs for JDK 8 and JDK 11:

6. Download and Install CryptoComply for Java (CC for Java) SafeLogic - Java JCE Provider on all hosts:

   For JDK 8:
   a) Obtain the SafeLogic CC Java module JAR file.
   b) Copy the ccj-3.0.2.1.jar file to $JAVA_HOME/jre/lib/ext.
   c) Obtain the SafeLogic BCTLS Java module JAR file.
   d) Copy the bctls-safelogic.jar file to $JAVA_HOME/jre/lib/ext.
   e) Change the file permissions on both the ccj-3.0.2.1.jar and bctls-safelogic.jar files to root and 0644:

   ```
   chown root: ${java_home}/jre/lib/ext/ccj-3.0.2.1.jar
   chmod 0644 ${java_home}/jre/lib/ext/ccj-3.0.2.1.jar
   chown root: ${java_home}/jre/lib/ext/bctls-safelogic.jar
   ```

```
chmod 0644 ${java_home}/jre/lib/ext/bctls-safelogic.jar
```

For JDK 11:

a) Obtain the SafeLogic CC Java module JAR file.
b) Copy the ccj-3.0.2.1.jar file to /path/of/your/choice/.
c) Obtain the SafeLogic BCTLS Java module JAR file.
d) Copy the bctls-safelogic.jar file to /path/of/your/choice/.
e) Change the file permissions on both the ccj-3.0.2.1.jar and bctls-safelogic.jar files to root and 0644:

```
chown root: /path/of/your/choice/ccj-3.0.2.1.jar
chmod 0644 /path/of/your/choice/ccj-3.0.2.1.jar
chown root: /path/of/your/choice/bctls-safelogic.jar
chmod 0644 /path/of/your/choice/bctls-safelogic.jar
```

Note: The path /path/of/your/choice/ must be the same for both ccj-3.0.2.1.jar and bctls-safelogic.jar.

Attention: /path/of/your/choice/ is /usr/share/java.

For JDK 17: Follow the same instructions as provided for JDK 11.

7. Add the CCJ configuration to the $JAVA_HOME/conf/security/java.policy file within the closed bracket as shown below:

```
//CCJ Java Permissions
permission java.lang.RuntimePermission "getProtectionDomain";
permission java.lang.RuntimePermission "accessDeclaredMembers";
permission java.util.PropertyPermission "java.runtime.name", "read";
permission java.security.SecurityPermission "putProviderProperty.CCJ";
//CCJ Key Export and Translation
permission com.safelogic.cryptocomply.crypto.CryptoServicesPermission "exp
ortKeys";
//CCJ SSL
permission com.safelogic.cryptocomply.crypto.CryptoServicesPermission "t
lsAlgorithmsEnabled";
//CCJ Setting of Default SecureRandom
permission com.safelogic.cryptocomply.crypto.CryptoServicesPermission "d
efaultRandomConfig";
//CCJ Setting CryptoServicesRegistrar Properties
permission com.safelogic.cryptocomply.crypto.CryptoServicesPermission "glo
balConfig";
//CCJ Enable JKS
permission com.safelogic.cryptocomply.jca.enable_jks "true";
}
```

8. Edit the $JAVA_HOME/conf/security/java.security file as follows:

Note:

Using yum install java-1.8.0-openjdk-devel on RHEL 8 will install a higher-than-recommended version of OpenJDK, such as 8u402, which should work with one adjustment to the java.security file. If any "fips.provider" parameters exist in java.security, they must be replaced with the following:

```
#
# Security providers used when FIPS mode support is active
#
#fips.provider.1=sun.security.pkcs11.SunPKCS11 ${java.home}/lib/secur
ity/nss.fips.cfg
#fips.provider.2=sun.security.provider.Sun
#fips.provider.3=sun.security.ec.SunEC
#fips.provider.4=com.sun.net.ssl.internal.ssl.Provider SunPKCS11-NSS-
FIPS
fips.provider.1=com.safelogic.cryptocomply.jcajce.provider.CryptoCo
mplyFipsProvider
fips.provider.2=org.bouncycastle.jsse.provider.BouncyCastleJsseProv
ider fips:CCJ
fips.provider.3=sun.fips.provider.Sun
fips.provider.4=sun.security.rsa.SunRsaSign
fips.provider.5=sun.security.ec.SunEC
#fips.provider.6=com.sun.net.ssl.internal.ssl.Provider
fips.provider.6=com.sun.crypto.provider.SunJCE
fips.provider.7=sun.security.jgss.SunProvider
fips.provider.8=com.sun.security.sasl.Provider
fips.provider.9=org.jcp.xml.dsig.internal.dom.XMLDSigRI
#fips.provider.11=sun.security.smartcardio.SunPCSC
```

a) Add the following lines:

```
#
# List of providers and their preference orders (see above):
#
security.provider.1=com.safelogic.cryptocomply.jcajce.provider.CryptoCo
mplyFipsProvider
security.provider.2=org.bouncycastle.jsse.provider.BouncyCastleJsseProv
ider fips:CCJ
security.provider.3=sun.security.provider.Sun
security.provider.4=sun.security.rsa.SunRsaSign
security.provider.5=sun.security.ec.SunEC
#security.provider.6=com.sun.net.ssl.internal.ssl.Provider
security.provider.6=com.sun.crypto.provider.SunJCE
security.provider.7=sun.security.jgss.SunProvider
security.provider.8=com.sun.security.sasl.Provider
security.provider.9=org.jcp.xml.dsig.internal.dom.XMLDSigRI
#security.provider.11=sun.security.smartcardio.SunPCSC
```

b) Comment out the ssl.KeyManagerFactory.algorithm=SunX509 line and add a new line with the text ssl.KeyManagerFactory.algorithm=X.509.

```
#
# Determines the default key and trust manager factory algorithms for
# the javax.net.ssl package.
#
#ssl.KeyManagerFactory.algorithm=SunX509
ssl.KeyManagerFactory.algorithm=X.509
ssl.TrustManagerFactory.algorithm=PKIX
```

ONLY for JDK 11 or above:

9. Make the following changes to the Cloudera Manager configuration:

a) Open /etc/default/cloudera-scm-server.

b) uncomment the below configs related to FIPS:

```
#
# Enable FIPS mode
#
# To enable FIPS mode set the -Dcom.cloudera.cmf.fipsMode to true
#
# export CMF_JAVA_OPTS="${CMF_JAVA_OPTS} -Dcom.cloudera.cmf.fipsMode=
true"
#
# If JDK version is 11 or higher:
# Uncomment and provide values below to include CCJ with FIPS mode
# export CMF_JAVA_OPTS="${CMF_JAVA_OPTS} -Dcom.cloudera.cloudera.cmf.fi
psMode.jdk11plus.ccj.jar.path=/path/to/ccj.jar -Dcom.cloudera.cloudera.c
mf.fipsMode.jdk11plus.ccj.moduleName=ccj_module_name"
#
# If JDK version is 11 or higher:
# Uncomment and provide values below to include BCTLS with FIPS mode
# export CMF_JAVA_OPTS="${CMF_JAVA_OPTS} -Dcom.cloudera.cloudera.cmf.
fipsMode.jdk11plus.bctls.jar.path=/path/to/bctls.jar -Dcom.cloudera.clou
dera.cmf.fipsMode.jdk11plus.bctls.moduleName=bctls_module_name"
#
```

to

```
#
# Enable FIPS mode
#
# To enable FIPS mode set the -Dcom.cloudera.cmf.fipsMode to true
#
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS} -Dcom.cloudera.cmf.fipsMode=true"
#
# If JDK version is 11 or higher:
# Uncomment and provide values below to include CCJ with FIPS mode
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS} -Dcom.cloudera.cloudera.cmf.fi
psMode.jdk11plus.ccj.jar.path=/path/of/your/choice/ccj-3.0.2.1.jar -Dcom
.cloudera.cloudera.cmf.fipsMode.jdk11plus.ccj.moduleName=ccj_module_name
"
#
# If JDK version is 11 or higher:
# Uncomment and provide values below to include BCTLS with FIPS mode
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS} -Dcom.cloudera.cloudera.cmf.fi
psMode.jdk11plus.bctls.jar.path=/path/of/your/choice/bctls-safelogic.jar
 -Dcom.cloudera.cloudera.cmf.fipsMode.jdk11plus.bctls.moduleName=bctls_m
odule_name"
```

```
#
```

Note: The moduleName of ccj and bctls modules depends on the how the jars were published. If the jar has a 'Automatic Module Name' label inside the META-INF/MANIFEST.MF file of the jar, then use that as the module name. Otherwise, the name of the jar (separated by dots, explained below) becomes the default module name.

For example:

If the ccj jar file name is ccj-test-3.0.2.1.jar, then the module name becomes cj.test. The version numbers are ignored.

You can also find out what the module name would be, using the following command:

```
$ sudo ${JAVA_HOME}/bin/jar --file=/path/of/your/choice/ccj-test-3.0.2.1
.jar --describe-module
No module descriptor found. Derived automatic module.
```

```
ccj.test@3.0.2.1 automatic <---- module Name is ccj.test
requires java.base mandated
contains com.safelogic.cryptocomply
```

10. For JDK 11 and 17, execute the following after updating the two paths /path/of/your/choice/.... to match your ccj and bctls jar paths. This will create an default environment variable for Java commands to register the Safelogic Java modules.

    echo "export JDK_JAVA_OPTIONS='--module-path=/path/of/your/choice/ccj-3.0.2.1.jar:/path/of/your/choice/bctls.jar --add-exports java.base/sun.security.provider=com.safelogic.cryptocomply.fips.core --add-modules com.safelogic.cryptocomply.fips.core,bctls'" >/etc/profile.d/cloudera-ccj.sh

11. Add the following safety-valve settings for Spark cluster mode:

    For SPARK_ON_YARN>GATEWAY role:

    spark-conf/spark-env.sh_client_config_safety_valve

```
export SPARK_SUBMIT_OPTS="$SPARK_SUBMIT_OPTS --add-exports=java.
base/sun.security.provider=bctls --add-exports=java.base/sun.sec
urity.provider=com.safelogic.cryptocomply.fips.core --add-module
s=com.safelogic.cryptocomply.fips.core,bctls --module-path=<BCTL
S_JARS_DIR>"
```

    For SPARK3_ON_YARN > GATEWAY role:

    spark3-conf/spark-env.sh_client_config_safety_valve

```
export SPARK_SUBMIT_OPTS="$SPARK_SUBMIT_OPTS --add-exports=java.
base/sun.security.provider=bctls --add-exports=java.base/sun.sec
urity.provider=com.safelogic.cryptocomply.fips.core --add-module
s=com.safelogic.cryptocomply.fips.core,bctls --module-path=<BCTL
S_JARS_DIR>"
```

    Where <BCTLS_JARS_DIR> is the directory containing the SafeLogic bctls and fips core jar files.

For Spark to work correctly on FIPS for JDK 11 or above:

12. Add the following safety-valve settings:

    For SPARK_ON_YARN>GATEWAY role:

    spark-conf/spark-defaults.conf_client_config_safety_valve

```
spark.yarn.am.extraJavaOptions=--add-exports=java.base/sun.secur
ity.provider=bctls --add-exports=java.base/sun.security.provider
=com.safelogic.cryptocomply.fips.core --add-modules=com.safelogi
c.cryptocomply.fips.core,bctls --module-path=<BCTLS_JARS_DIR> -D
com.safelogic.cryptocomply.fips.approved_only=true -Djava.net.pr
```

```
eferIPv4Stack=true -Djdk.tls.ephemeralDHKeySize=2048 -Djdk.tls.t
rustNameService=true -Dorg.bouncycastle.jsse.client.assumeOrigin
alHostName=true
spark.driver.extraJavaOptions=--add-exports=java.base/sun.securi
ty.provider=bctls --add-exports=java.base/sun.security.provider=
com.safelogic.cryptocomply.fips.core --add-modules=com.safelogic
.cryptocomply.fips.core,bctls --module-path=<BCTLS_JARS_DIR> -Dc
om.safelogic.cryptocomply.fips.approved_only=true -Djava.net.pre
ferIPv4Stack=true -Djdk.tls.ephemeralDHKeySize=2048 -Djdk.tls.tr
ustNameService=true -Dorg.bouncycastle.jsse.client.assumeOrigina
lHostName=true
spark.executor.extraJavaOptions=--add-exports=java.base/sun.secur
ity.provider=bctls --add-exports=java.base/sun.security.provider
=com.safelogic.cryptocomply.fips.core --add-modules=com.safelogi
c.cryptocomply.fips.core,bctls --module-path=<BCTLS_JARS_DIR> -D
com.safelogic.cryptocomply.fips.approved_only=true -Djava.net.pr
eferIPv4Stack=true -Djdk.tls.ephemeralDHKeySize=2048 -Djdk.tls.t
rustNameService=true -Dorg.bouncycastle.jsse.client.assumeOrigin
alHostName=true
```

For SPARK3_ON_YARN > GATEWAY role:

spark3-conf/spark-defaults.conf_client_config_safety_valve

```
spark.yarn.am.extraJavaOptions=--add-exports=java.base/sun.secur
ity.provider=bctls --add-exports=java.base/sun.security.provider
=com.safelogic.cryptocomply.fips.core --add-modules=com.safelogi
c.cryptocomply.fips.core,bctls --module-path=<BCTLS_JARS_DIR> -D
com.safelogic.cryptocomply.fips.approved_only=true -Djava.net.pr
eferIPv4Stack=true -Djdk.tls.ephemeralDHKeySize=2048 -Djdk.tls.t
rustNameService=true -Dorg.bouncycastle.jsse.client.assumeOrigin
alHostName=true
spark.driver.defaultJavaOptions=--add-exports=java.base/sun.secu
rity.provider=bctls --add-exports=java.base/sun.security.provide
r=com.safelogic.cryptocomply.fips.core --add-modules=com.safelog
ic.cryptocomply.fips.core,bctls --module-path=<BCTLS_JARS_DIR> -
Dcom.safelogic.cryptocomply.fips.approved_only=true -Djava.net.p
referIPv4Stack=true -Djdk.tls.ephemeralDHKeySize=2048 -Djdk.tls.
trustNameService=true -Dorg.bouncycastle.jsse.client.assumeOrigi
nalHostName=true
spark.executor.defaultJavaOptions=--add-exports=java.base/sun.s
ecurity.provider=bctls --add-exports=java.base/sun.security.prov
ider=com.safelogic.cryptocomply.fips.core --add-modules=com.safe
logic.cryptocomply.fips.core,bctls --module-path=<BCTLS_JARS_DIR
> -Dcom.safelogic.cryptocomply.fips.approved_only=true -Djava.ne
t.preferIPv4Stack=true -Djdk.tls.ephemeralDHKeySize=2048 -Djdk.t
ls.trustNameService=true -Dorg.bouncycastle.jsse.client.assumeOr
iginalHostName=true
```

Where <BCTLS_JARS_DIR> is the directory containing the SafeLogic bctls and fips core jar files.

Step 2: Install and configure the SafeLogic modules and packages for RHEL 7 OS

For RHEL 7, install and configure SafeLogic packages.

About this task

Important:

RHEL 7: install and configure SafeLogic modules.

as described below

RHEL 8:

Skip this step. DO NOT install SafeLogic modules and packages.

OS-provided OpenSSL

Skip this step. DO NOT install SafeLogic modules and packages.

Please contact your Cloudera Sales or Support team to obtain the CDP PVC Base SafeLogic FIPS modules.

Installing the SafeLogic modules and packages may overwrite existing packages on your hosts. Cloudera recommends that you test any non-Cloudera software running on the host for correct functionality.

Procedure

1. Obtain the CryptoComply for Libgcrypt (CC for Libgcrypt) and CryptoComply for Server (CC for Server) SafeLogic modules and packages.
2. Copy the CryptoComply for Server (CCS) - OpenSSL RPMs to all hosts.
   a) Unzip the download.
   b) Run the following command to install the packages.

```
yum localinstall -y openssl-1.0.2y-1.el7.centos.x86_64.rpm \
openssl-devel-1.0.2y-1.el7.centos.x86_64.rpm \
openssl-libs-1.0.2y-1.el7.centos.x86_64.rpm \
openssl-perl-1.0.2y-1.el7.centos.x86_64.rpm \
openssl-static-1.0.2y-1.el7.centos.x86_64.rpm
```

3. Copy the CryptoComply for Libgcrypt RPMs to all hosts.
   a) Unzip the file.
   b) Run the following command to install the packages.

```
yum localinstall -y libgcrypt-1.5.3-12.el7.centos.1.x86_64.rpm
```

Step 3: Install Cloudera Manager server

Procedure

1. Log in to the Cloudera Manager server host.
2. Run the following command to install Cloudera Manager server.

```
sudo yum install cloudera-manager-daemons cloudera-manager-agent cloudera-
manager-server
```

3. Add the following line at the end of the /etc/default/cloudera-scm-server file:

```
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS} -Dcom.cloudera.cmf.fipsMode=true -
Dcom.safelogic.cryptocomply.fips.approved_only=true"
```

Step 4: Validate the CCJ and CCS installation

Run the following commands on each host to validate the CCJ and CCS installation.

Procedure

1. Run the following command:

```
sysctl crypto.fips_enabled
```

This should return:

```
crypto.fips_enabled = 1
```

2. Run the following command:

```
echo greeting | openssl md5
```

This command should fail, indicating that FIPS is enabled.

Attention: The next step differs for JDK 8 and JDK 11:

3. Run the following command:

JDK 8

Run the following command to verify the list of security providers in JDK 8

```
read -r -d '' list_providers <<EOF
p = java.security.Security.getProviders();
for (i = 0; i < p.length; i++) { java.lang.System.out.println(
p[i]); }
EOF
${JAVA_HOME}/bin/jrunscript -e "$list_providers"
```

This command should return the version numbers of the SafeLogic packages, for example:

```
CCJ version 1.01
SUN version 1.8
SunRsaSign version 1.8
SunEC version 1.8
SunJSSE version 1.8
SunJCE version 1.8
SunJGSS version 1.8
SunSASL version 1.8
XMLDSig version 1.8
SunPCSC version 1.8
```

JDK 11 (Recommended)

Run the following command on the Cloudera Manager Server to verify the list of security providers in JDK 11, which grabs the chosen path for the ccj jar file (assume that the bctls file is in the same directory) and show providers with those modules added:

```
cat > ListSecurityProviders.java <<-EOF
import java.security.Provider;
import java.security.Security;

public class ListSecurityProviders {
 public static void main(String[] args) {
 Provider[] providers = Security.getProviders();
 for (Provider provider : providers) {
 System.out.println("Provider: " + provider.getName());
 System.out.println("Version: " + provider.getVersionStr());
 System.out.println("Info: " + provider.getInfo());
 System.out.println();
```

```
  }
  }
}
EOF
```

```
java -p /directory/chosen/for/ccj-bctls/jars/ ListSecurityProvid
ers.java
```

The output will include the following providers if they are configured and referenced properly: for example:

```
Provider: CCJ
Version: <version>
Info: CryptoComply® for Java version <version>
Provider: BCJSSE
Version: <version>
Info: Bouncy Castle JSSE Provider Version <version>
```

For **JDK 17:** Follow the same instruction as provided for JDK 11.

4. Run the following command:

```
read -r -d '' do_maxAESKeyLength <<EOF
java.lang.System.out.println(javax.crypto.Cipher.getMaxAllowedKeyLength("
AES/CBC/PKCS5Padding"));
EOF
answer=`${JAVA_HOME}/bin/jrunscript -Dcom.safelogic.cryptocomply.fips.a
pproved_only=true -e "$do_maxAESKeyLength"`
echo $answer
```

This command should return:

```
2147483647
```

Step 5: Install and configure databases

About this task

Note:

Running the embedded PostgreSQL database on a FIPS mode enabled OS, configured in a FIPS compliant manner, is not supported.

Procedure

1. Configure the database in a FIPS-compliant manner. Consult the vendor documentation for your database for details.
2. Enable the database for TLS/SSL clients, to ensure that all JDBC connections into these databases are FIPS compliant. Consult the vendor documentation for your database for details.
3. Configure JDBC Driver in a FIPS compliant manner with TLS/SSL and BCFKS provided by CCJ JCE provider. Consult the following Cloudera Knowledge Base article for more information: Configuring SSL/TLS from the various CDH Services to their respective PostgreSQL Databases.
4. Complete the setup of your databases for use with Cloudera Manager and Cloudera Runtime components. See Install and Configure PostgreSQL for CDP on page 152.

   Note: Starting 7.1.9 SP1, if you are using PostgreSQL as a backend database for Hue on a FIPS cluster, you must install a version of the psycopg2 package from the source to be at least 2.9.5 on all Hue hosts because the psycopg2-binary package uses its version of the libssl library file which does not support FIPS. See Installing the psycopg2 Python package for PostgreSQL database on a FIPS cluster (RHEL 8).

Configure Cloudera Manager for FIPS

Perform the following steps to install and configure Cloudera Manager for FIPS.

Procedure

1. Log in to the Cloudera Manager server host and run the following command to start the Cloudera Manager server:

```
sudo systemctl start cloudera-scm-server
```

2. Wait a few minutes for Cloudera Manager server to start.
3. Run the following command to start the Cloudera Manager agents, the `supervisord` process, and all managed service processes:

```
sudo systemctl start cloudera-scm-agent
```

4. Use a web browser to navigate to http://<server_host>:7180, where <server_host> is the FQDN or IP address of the host where the Cloudera Manager server is running.
5. Log into Cloudera Manager Admin Console. The default credentials are:
   - User name: admin
   - Password: admin
6. Set up a cluster. See Step 6: Start the Cloudera Manager Server and Agents on page 218 and Step 7: Using the installation wizard on page 220 for more information.

Install and configure additional required components

Use the following steps to install additional required components for FIPS.

About this task

Note:

FIPS publications (including the 140-2 publication) do not mandate the use of Apache Knox, Kerberos, Apache Ranger, Apache Atlas, HDFS TDE, Ranger KMS, or Key Trustee Server. However, it is typical that these features are employed with Cloudera deployments in government regulated environments to achieve many types of government authorizations with various government standards, such as FedRAMP or FISMA. FIPS and NIST publications serve as the basis for many of these government standards.

Procedure

1. Perform the Additional Steps for Apache Ranger on page 245.
2. Add Ranger to the Shadow group.

```
usermod -a -G shadow ranger
```

3. Install and Configure TLS either automatically or manually.

   If you are using Auto-TLS, see:

   - Use case 1: Use CM to generate an internal CA and corresponding certificates
   - Use case 2: Use an existing Certificate Authority

   If you are manually configuring TLS, see:

   - Manually Configuring TLS Encryption for Cloudera Manager

   Note: Additional manual TLS configuration steps may be required for stack components.

   Generate certificates in BCFKS format

   The standard keytool utility distributed with the JDK can generate BCFKS formatted keystores using the CCJ security provider. When the CCJ security provider is statically installed into the JDK as previously described, there is no need to pass the keytool utility the -providerpath path/to/ccj-3.0.2.1.jar or -providerclass com.safelogic.cryptocomply.jcajce.provider.ProvBCFKS arguments. It is only necessary to pass BCFKS as the storetype for the keytool operation being invoked.

   For example, keytool -importkeystore can be used to import a PKCS12 keystore into a BCFKS keystore:

```
keytool \
    -importkeystore -v \
    -srckeystore <pkcs12_keystore_file> \
    -srcstoretype PKCS12 \
    -srcstorepass <pkcs12_pass> \
    -destkeystore <bcfks_keystore_file> \
    -deststoretype BCFKS \
    -deststorepass <bcfks_keystore_pass> \
    -destkeypass <bcfks_key_pass>
```

   Systems administrators and other platform implementers should consult their organization’s information systems security managers for the correct procedures for generating keypairs and requesting signing of x509 certificates. The Cloudera Data Platform requires the private key and signed certificate in both PEM encoded and BCFKS keystore format. The steps to accomplish this task might look similar to the following:

   a) openssl genpkey
   b) openssl req
   c) Have the CA sign the CSR.
   d) Import the private key and signed certificate into a PKCS12 keystore:

```
openssl pkcs12
```

   e) Import the PKCS12 keystore into a BCFKS keystore:

```
keytool -importkeystore
```

4. Enable Kerberos authentication using the Cloudera Manager Kerberos wizard.
5. Set the kdc_timeout value in the krb5.conf file to a high enough setting to avoid client timeout errors while running queries.
   a) Open the /etc/krb5.conf file with a text editor.
   b) Under [libdefaults], set the kdc_timeout value to a minimum of 5000 (5 seconds).
6. Install Apache Knox. See Installing Apache Knox on page 246.
7. Install Ranger KMS with Key Trustee Server. See Installing Ranger KMS backed with a Key Trustee Server and HA
8. Configure HDFS Transparent Data Encryption with Ranger KMS with Key Trustee Server.

Production Installation

These topics provide procedures for installing Cloudera Private Cloud Base in a production environment.

Related Information

Cloudera Private Cloud Base Installation Guide

Before You Install

Before you begin a production installation of Cloudera Manager, Cloudera Runtime, and other managed services, review the Cloudera Private Cloud Base Requirements and Supported Versions on page 12, in addition to the Cloudera Data Platform Release Notes.

For planning, best practices, and recommendations, review the CDP Private Cloud Base Reference Architecture.

Important:

- In a typical installation process, socket's somax connection must NOT be set to a very low value (default is 128, which is very low for Hadoop systems). Cloudera recommends that you set socket's somax connection value to atleast 16000 OOTB via Cloudera Manager or host inspector.
- Security-Enhanced Linux (SELinux) allows you to set access control through policies. However, if you are unable to deploy the Runtime cluster using your policies, you can set SELinux in permissive mode on each host of your cluster before you deploy the Runtime parcels.

Caution: Cloudera requires disabling the fapolicyd daemon present in RHEL 8 (and later) systems before beginning installation of Cloudera Manager application. Be informed that fapolicyd is a user space daemon that determines access rights to files based on attributes of the process and file. It can be used to either blacklist or whitelist processes or file access. Proceed with caution with enforcing the use of this daemon. Improper configuration may render the system non-functional.

Note: The importance of security in a production environment cannot be understated. TLS and Kerberos form the baseline for secure operations of your CDP Runtime environment. Cloudera supports security services such as Ranger and Atlas only when they are run on clusters where Kerberos is enabled to authenticate users.

YARN Queue Manager in Cloudera Data Platform (CDP) Private Cloud Base 7.1.9 CHF 2 does not require you to install a PostGres database, therefore users should not see the Setup Database screen and should be able to skip the Setup Database screen. When conducting a fresh install of 7.1.9 CHF 2, and if you are unable to bypass the Setup Database screen, perform the following steps:

1. Ensure that you have both CDP and Cloudera Manager upgraded to 7.1.9 CHF 2.
2. When you reach the Setup Database screen in the Cloudera Manager installation wizard for Queue Manager, enter any dummy values for the following fields:

   a. Database name: configstore
   b. Database Username: dbuser
   c. Database Password: dbpassword
   d. Database Hostname: localhost

   YARN Queue Manager does not connect to PostGres with the above details and falls back to the embedded database.
3. Run the following script command in a browser console to enable the Continue button:

   ```
   document.querySelector('.btn.next').removeAttribute('disabled');
   ```
4. Click Continue and proceed with the YARN Queue Manager installation.

5. After installation is complete, SSH into the host that has Queue Manager installed, and run this command: sed -i ' s/migrationCompleted=true/migrationCompleted=false/' /var/lib/hadoop-yarn/migration.properties

   Note: Enable Queue Manager in the YARN configurations, and restart YARN.

6. Restart YARN Queue Manager.

The following topics describe additional considerations you should be aware of before beginning an installation:

Installing Python 3

Certain CDP services such as Cloudera Manager and Hue use Python 3.x. Installing Python 3.x is mandatory to deploy CDP Private Cloud Base. Review the supported operating systems (OS) and use the corresponding instructions to install the recommended Python version.

Cloudera Manager 7.11.3 is supported only on the following operating systems. The following table lists the supported operating systems, the Python version that is distributed with those operating systems, and the Python 3 versions required for deploying CDP:

Operating system	Python version distributed with the OS	Python version required for deploying CDP	Installation required?
RHEL 7.9	Python 2	Python 3.8	Yes
RHEL 7.9 FIPS	Python 2	Python 3.8	Yes
RHEL 8.6	Python 3.6	Python 3.8	Yes
RHEL 8.8	Python 3.6	Python 3.8	Yes
RHEL 8.8 FIPS	Python 3.6	Python 3.8	Yes
For 7.1.9 SP1 RHEL 8.10	Python 3.6	Python 3.8	Yes
For 7.1.9 SP1 RHEL 9.4	Python 3.9	Python 3.9	Yes
For 7.1.9 SP1 RHEL 9.2	Python 3.9	Python 3.9	Yes
RHEL 9.1	Python 3.9	Python 3.9	Yes
For 7.1.9 GA upto 7.1.9 CHF6 SLES 15 SP4	Python 3.6	Python 3.8	Yes
For 7.1.9 SP1 SLES 15 SP4	Python 3.6	Python 3.10	Yes
For 7.1.9 SP1 SLES 15 SP5	Python 3.11	Python 3.10	Yes
CentOS 7.9	Python 2	Python 3.8	Yes
CentOS 7.9 FIPS	Python 2	Python 3.8	Yes
CentOS 8	None	Python 3.8	Yes
Oracle 8.8 UEK	Python 3.6	Python 3.8	Yes
For 7.1.9 SP1 Oracle 8.10 (supported with Red Hat Compatible Kernel (RHCK) only)	Python 3.6	Python 3.8	Yes
Ubuntu 20.04	Python 3.8	Python 3.8	No

Operating system	Python version distributed with the OS	Python version required for deploying CDP	Installation required?
For 7.1.9 SP1 Ubuntu 22.04	Python 3.10	Python 3.10	No

Important: The minimum required version of Python 3.8 is 3.8.12. The minimum version of Python 3.9 is 3.9.14.

Cloudera Manager 7.11.3 does not work with Python 2.7. While using Cloudera Manager 7.11.3 with Cloudera Runtime 7.1.8 or 7.1.9 version, you may remove all Python 2 versions from the operating system, only when the operating system allows you to remove the Python 2 version.

If you are running Cloudera Runtime 7.1.7 SP2 or below versions with Cloudera Manager 7.11.3, then Python 2.7 is still required for the Cloudera Runtime components. In this scenario, you must install both Python 2.7 (for Cloudera Runtime components) and Python 3 (for Cloudera Manager 7.11.3).

Difference between standard and custom Python packages

The term *standard Python package* in Cloudera documentation refers to the Python package that is distributed with the OS, or an unmodified Python binary that is compiled from the source. If you have made modifications or customizations to the Python source code and then compiled the binaries into a package, then that is referred to as the *custom Python package*.

Deployment options for Python 3

You can install Python in a standard or custom location on your filesystem. In Cloudera documentation, the term *standard location* refers to one of the following default locations where Python is installed, depending on the package:

- /usr/bin
- /usr/local/python38/bin
- /usr/local/bin
- /opt/rh/rh-python38/root/usr/bin

A *custom location* is any location on the filesystem other than these standard locations.

Typically, all standard packages that are distributed with an OS are installed in the standard locations. However, you have an option to install custom Python 3 packages in standard or custom locations, or standard Python binaries in standard or custom locations.

If the supported Python 3 version (standard package) is distributed with the OS, you can use the OS's package manager to install Python, which gets installed in the standard location. If you want to install a standard Python binary at a standard or custom location, then you must manually download, build, and install the corresponding version of the Python binaries from the source code.

Important: Since Python 3.8 standard package is not available with RHEL 7, you must manually download, install, and build Python 3.8 from source on RHEL 7.

Important: When you install Python in a custom location, then you must create symbolic links to enable Cloudera Manager agents to access Python binaries from the custom location. If you are using Hue, then you must also prepend the custom path to the custom bin directory in the Hue Service Environment Advanced Configuration Snippet (Safety Valve) field in Cloudera Manager.

Installing standard Python 3.8 binary on CentOS 7, CentOS 8 at a standard or custom location

Cloudera Manager agents and Hue require Python 3.8. You must install Python 3.8.12 or higher on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow the instructions in this topic to manually download, install, and build *standard Python 3.8 binaries* from source on CentOS 7 and CentOS 8 at a standard or custom location.

About this task

You must manually download, install, and build Python 3.8 from source on CentOS 7 and CentOS 8 because:

- Python 2.7.5 is preinstalled on CentOS 7, but CDP requires Python 3.8
- Python is not distributed with CentOS 8, unlike other Linux distributions

You can install Python 3.8 in a standard location or in a custom location depending on your organization's policies. To install Python 3.8 in a custom location, replace *[***CUSTOM-INSTALL-PATH***]* in the instructions with the custom path.

Before you begin

Install the following necessary developer tools and packages on your system:

- gcc
- make
- openssl-devel
- bzip2-devel
- libffi-devel
- zlib-devel

```
yum install gcc openssl-devel bzip2-devel libffi-devel zlib-devel -y
```

Note: You cannot install the “openssl-devel” package on a FIPS cluster.

Installing Python 3.8 from CentOS Software Collections (SCL)

1. You can install the Python 3.8 from CentOS Software Collections (SCL) by running the following commands:

   ```
   yum install centos-release-scl ( part of the extra repository )
   yum install rh-python38 rh-python38-python-pip rh-python38-python-pip-wh
   eel
   ```
2. Verify Python 3.8 is installed successfully.
3. If you are unable to install Python 3.8 from CentOS Software Collections (SCL), then you can perform the following steps to build the Python 3.8 from source distribution.

Procedure

1. SSH into the host system as a root user.
2. Change directory to /opt:

   ```
   cd /opt
   ```
3. Download Python 3.8 as follows:

   ```
   curl -O https://www.python.org/ftp/python/3.8.17/Python-3.8.17.tgz
   ```
4. Decompress the package as follows:

   ```
   tar -zxvf Python-3.8.17.tgz
   ```
5. Change directory to where you decompressed the Python 3.8 package:

   ```
   cd /opt/Python-3.8.17
   ```

6. Install Python 3.8 as follows:

Standard location:

```
./configure --enable-shared
```

Attention: By default, Python is installed in any one of the following locations:

- /usr/bin
- /usr/local/python38/bin
- /usr/local/bin
- /opt/rh/rh-python38/root/usr/bin

Custom location:

```
./configure --enable-shared --prefix=[***CUSTOM-INSTALL-PATH***]
```

If you are installing Python 3.8 in any non-standard location, then you must specify the path using the --prefix option. You must also create a symbolic link pointing to /usr/bin or to /usr/local/bin by running the following command:

```
ln -s [***SOURCE***] [***DESTINATION***]
```

For example:

```
ln -s [***CUSTOM-INSTALL-PATH***]/python3.8 /usr/bin/python3.8
```

The --enabled-shared option is used to build a shared library instead of a static library.

7. Build Python 3.8 as follows:

a) Run the make command to compile the files:

```
make
```

b) Run the following command to put the compiled files in the default location, or in the custom location that you specified using the --prefix option:

```
make altinstall
```

Using altinstall prevents you from overwriting the system Python, if any.

c) Copy the shared compiled library files (libpython3.8.so) to the /lib64/ directory:

```
cp --no-clobber ./libpython3.8.so* /lib64/
```

The --no-clobber option is used to prevent overwriting files.

d) Change the permissions of the libpython3.8.so files as follows:

```
chmod 755 /lib64/libpython3.8.so*
```

If you see an error such as “error while loading shared libraries: libpython3.8.so.1.0: cannot open shared object file: No such file or directory”, then run the following command:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib/
```

8. Change the permission of the Python 3.8 installation directory to 755 so that Hue and its related services can leverage the binary and the site packages, as follows:

```
chmod -R 755 /usr/local/lib/python3.8
```

What to do next

Enable Cloudera Manager to use Python 3.8 installed in a custom location

If you have installed Python 3.8 at a custom location, then you must either create a symbolic link using the `alternatives` command or by adding an environment variable to the Cloudera Manager Agent in the systemd service definition files with the full path to the Python 3.8 custom-built binary.

To use the `alternatives` command, get help by running the command `alternatives --help`. You should create symbolic links from either /usr/bin/python3.8 or /usr/local/bin/python3.8 to *[***CUSTOM-INSTALL-PATH***]*/bin/python3.8 where you installed the Python 3.8 package.

Important: Using the alternatives command to set up symbolic links does not support setting up of the PYTHONHOME or PYTHONPATH variables if that is necessary for your custom version of Python 3.8 installation.

To add an environment variable to the Cloudera Manager Agent in the systemd service definition files perform the following steps:

1. SSH into the host system on which you installed Python 3.8 as a root user.
2. Open the /usr/lib/systemd/system/cloudera-scm-supervisord.service file for editing.
3. Add the following line in the cloudera-scm-supervisord.service file under the [Service] section:

```
Environment="PYTHONBIN=[***CUSTOM-INSTALL-PATH***]/bin/python3
.8"
```

4. Open the /usr/lib/systemd/system/cloudera-scm-agent.service file for editing.
5. Add the following line in the cloudera-scm-agent.service file under the [Service] section:

```
Environment="PYTHONBIN=[***CUSTOM-INSTALL-PATH***]/bin/python3
.8"
```

Important: If your custom version of Python 3.8 installation requires setting up the PYTHONHOME or PYTHONPATH variables, then add them with additional Environment lines.

Enable Hue to use Python 3.8 installed in a custom location

If you have installed Python 3.8 in a custom location and you did not configure alternatives, then you must prepend the custom path to the custom bin directory in Cloudera Manager Clusters Hue Configuration Hue Service Environment Advanced Configuration Snippet (Safety Valve) separated by colon (:) as follows:

Key: PATH

Value: *[***CUSTOM-INSTALL-PATH***]*:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

Install the psycopg2-binary package

Install the “psycopg2-binary” package after installing Python 3.8 for using PostgreSQL as a backend database for Hue.

Related Information

Introduction to the alternatives command in Linux

Installing the psycopg2 Python package for PostgreSQL-backed Hue

Installing Python 3.8 standard package on RHEL 8

Cloudera Manager agents and Hue require Python 3.8. You must install Python 3.8.12 or higher on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow the instructions in this topic to install the standard Python 3.8 packages in a standard location on RHEL 8.

About this task

Python 3 is distributed in versions 3.6, 3.8, and 3.9, provided by the python36, python38, and python39 modules on RHEL 8. CDP requires that you install Python version 3.8 on your cluster hosts.

Before you begin

Attention: If you are using RHEL 8, Cloudera recommends that you uninstall Python 2.7, if it is present on your hosts. If you are using RHEL 7, do not uninstall Python 2.7.

Procedure

1. SSH into the host system as a root user.
2. Install Python 3.8 by running the following command:

   ```
   yum install python38
   ```

   Attention: By default, Python is installed in any one of the following locations:

   - /usr/bin
   - /usr/local/python38/bin
   - /usr/local/bin
   - /opt/rh/rh-python38/root/usr/bin
3. Verify the Python version as follows:

   ```
   python3.8 --version
   ```

What to do next

Install the psycopg2-binary package

Install the “psycopg2-binary” package after installing Python 3.8 for using PostgreSQL as a backend database for Hue.

Related Information

Introduction to the alternatives command in Linux

Installing the psycopg2 Python package for PostgreSQL-backed Hue

Installing Python 3.8 standard package on RHEL 7

Cloudera Manager agents and Hue require Python 3.8. You must install Python 3.8.12 or higher on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow the instructions in this topic to install the standard Python 3.8 packages in a standard location on RHEL 7.

About this task

Python 3 is distributed in versions 3.6, 3.8, and 3.9, provided by the python36, python38, and python39 modules on RHEL 7. CDP requires that you install Python version 3.8 on your cluster hosts.

Before you begin

Note:

If you are using RHEL 8, Cloudera recommends that you uninstall Python 2.7, if it is present on your hosts. If you are using RHEL 7, do not uninstall Python 2.7.

Procedure

1. SSH into the host system as a root user.

2. Run the following command to enable the additional repositories:

```
subscription-manager repos --enable rhel-server-rhscl-7-rpms
```

Note:

The benefit of using Red Hat Software Collections is that you can have multiple versions of Python installed at the same time along with the base Python 2.7 that shipped with RHEL 7. You can easily switch between versions with scl enable.

3. Install Python 3.8 by running the following command:

```
sudo yum install rh-python38
```

Note: By default, Python is installed in the following location:

- /opt/rh/rh-python38/root/usr/bin/python3.8

4. Install the psycopg2-binary package by running the following command:

```
scl enable rh-python38 -- pip install --upgrade pip
scl enable rh-python38 -- pip install psycopg2-binary
```

5. Verify the Python version as follows:

```
/opt/rh/rh-python38/root/usr/bin/python3.8 --version
```

Installing standard Python 3.8 binary on RHEL 7, RHEL 8 at a standard or custom location

Cloudera Manager agents and Hue require Python 3.8. You must install Python 3.8.12 or higher on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow the instructions in this topic to manually download, install, and build *standard Python 3.8 binaries* from source on RHEL 7 or RHEL 8 at a standard or custom location.

About this task

You can install Python 3.8 in a standard location or in a custom location depending on your organization's policies. To install Python 3.8 at a custom location, replace *[***CUSTOM-INSTALL-PATH***]* in the instructions with the custom path.

Before you begin

Attention: If you are using RHEL 8, Cloudera recommends that you uninstall Python 2.7, if it is present on your hosts. If you are using RHEL 7, do not uninstall Python 2.7.

Install the following necessary developer tools and packages on your system:

- gcc
- make
- openssl-devel
- bzip2-devel
- libffi-devel
- zlib-devel

```
yum install gcc openssl-devel bzip2-devel libffi-devel zlib-devel -y
```

Note: You cannot install the “openssl-devel” package on a FIPS cluster.

Procedure

1. SSH into the host system as a root user.
2. Change directory to /opt:

   ```
   cd /opt
   ```

3. Download Python 3.8 as follows:

   ```
   curl -O https://www.python.org/ftp/python/3.8.17/Python-3.8.17.tgz
   ```

4. Decompress the package as follows:

   ```
   tar -zxvf Python-3.8.17.tgz
   ```

5. Change directory to where you decompressed the Python 3.8 package:

   ```
   cd /opt/Python-3.8.17
   ```

6. Install Python 3.8 as follows:

   Standard location:

   ```
   ./configure --enable-shared
   ```

   Attention: By default, Python is installed in any one of the following locations:

   - /usr/bin
   - /usr/local/python38/bin
   - /usr/local/bin
   - /opt/rh/rh-python38/root/usr/bin

   Custom location:

   ```
   ./configure --enable-shared --prefix=[***CUSTOM-INSTALL-PATH***]
   ```

   If you are installing Python 3.8 in any non-standard location, then you must specify the path using the --prefix option. You must also create a symbolic link pointing to /usr/bin or to /usr/local/bin by running the following command:

   ```
   ln -s [***SOURCE***] [***DESTINATION***]
   ```

   For example:

   ```
   ln -s [***CUSTOM-INSTALL-PATH***]/bin/python3.8 /usr/bin/python3.8
   ```

   The --enabled-shared option is used to build a shared library instead of a static library.

7. Build Python 3.8 as follows:
   a) Run the make command to compile the files:

   ```
   make
   ```

   b) Run the following command to put the compiled files in the default location, or in the custom location that you specified using the --prefix option:

   ```
   make altinstall
   ```

   Using altinstall prevents you from overwriting the system Python, if any.
   c) Copy the shared compiled library files (libpython3.8.so) to the /lib64/ directory:

   ```
   cp --no-clobber ./libpython3.8.so* /lib64/
   ```

   The --no-clobber option is used to prevent overwriting files.
   d) Change the permissions of the libpython3.8.so files as follows:

   ```
   chmod 755 /lib64/libpython3.8.so*
   ```

   If you see an error such as “error while loading shared libraries: libpython3.8.so.1.0: cannot open shared object file: No such file or directory”, then run the following command:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib/
   ```

8. Change the permission of the Python 3.8 installation directory to 755 so that Hue and its related services can leverage the binary and the site packages, as follows:

   ```
   chmod -R 755 /usr/local/lib/python3.8
   ```

What to do next

Enable Cloudera Manager to use Python 3.8 installed in a custom location

If you have installed Python 3.8 at a custom location, then you must either create a symbolic link using the `alternatives` command or by adding an environment variable to the Cloudera Manager Agent in the systemd service definition files with the full path to the Python 3.8 custom-built binary.

To use the `alternatives` command, get help by running the command `alternatives --help`. You should create symbolic links from either /usr/bin/python3.8 or /usr/local/bin/python3.8 to *[***CUSTOM-INSTALL-PATH***]*/bin/python3.8 where you installed the Python 3.8 package.

Important:

Using the `alternatives` command to set up symbolic links affects the host environment globally. All applications on the host will be affected. If this is different from what you intend, you might need to customize the PYTHONHOME or PYTH ONPATH variables for the CDP environment. For more information, see the following section.

Customizing Python environment for CDP Services

This section is intended for expert users with unique requirements to customize their Python environment for CDP services or non-Cloudera applications running on cluster hosts. These steps are not needed for normal configurations. The users must decide if they are using the PYTHONHO ME or PYTHONPATH variables.

Caution: Cloudera does not provide information or guidance on the usage or the effects of setting the PYTHONHOME or PYTHONPATH variables.

To add an environment variable to the Cloudera Manager Agent in the systemd service definition files perform the following steps:

1. SSH into the host system on which you installed Python 3.8 as a root user.
2. Open the /usr/lib/systemd/system/cloudera-scm-supervisord.service file for editing.
3. Add the following line in the cloudera-scm-supervisord.service file under the [Service] section:

```
Environment=”PYTHONBIN=[***CUSTOM-INSTALL-PATH***]/bin/python3
.8”
```

4. Open the /usr/lib/systemd/system/cloudera-scm-agent.service file for editing.
5. Add the following line in the cloudera-scm-agent.service file under the [Service] section:

```
Environment=”PYTHONBIN=[***CUSTOM-INSTALL-PATH***]/bin/python3
.8”
```

Important: If your custom version of Python 3.8 installation requires setting up the PYTHONHOME or PYTHONPATH variables, then add them with additional Environment lines.

Enable Hue to use Python 3.8 installed in a custom location

If you have installed Python 3.8 in a custom location and you did not configure alternatives, then you must prepend the custom path to the custom bin directory in Cloudera Manager Clusters Hue Configuration Hue Service Environment Advanced Configuration Snippet (Safety Valve) separated by colon (:) as follows:

Key: PATH

Value: *[***CUSTOM-INSTALL-PATH***]*:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

Install the psycopg2-binary package

Install the “psycopg2-binary” package after installing Python 3.8 for using PostgreSQL as a backend database for Hue.

Related Information

Introduction to the alternatives command in Linux

Installing the psycopg2 Python package for PostgreSQL-backed Hue

Installing Python 3.9 standard package on RHEL 9

Cloudera Manager agents and Hue require Python 3.9 on RHEL 9. You must install Python 3.9.14 or higher on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow instructions in this topic to install the standard Python 3.9 packages in a standard location on RHEL 9.1, RHEL 9.2, or RHEL 9.4.

About this task

Python 3.9 is the default Python implementation provided by RHEL 9 and is usually installed by default. Perform this task to install or re-install it manually.

Procedure

1. SSH into the host system as a root user.

2. Install Python 3.9 by running one of the following commands:

```
dnf install python3
```

or

```
yum install python3
```

Attention: By default, Python is installed in any one of the following locations:

- /usr/bin
- /usr/local/python39/bin
- /usr/local/bin
- /opt/rh/rh-python39/root/usr/bin

3. Verify the Python version as follows:

```
python3 -–version
```

or

```
python3.9 -–version
```

What to do next

Install the psycopg2-binary package

Install the “psycopg2-binary” package after installing Python 3.8 for using PostgreSQL as a backend database for Hue.

Related Information

Introduction to the alternatives command in Linux

Installing the psycopg2 Python package for PostgreSQL-backed Hue

Installing standard Python 3.9 binary on RHEL 9 at a standard or custom location

Cloudera Manager agents and Hue require Python 3.9 on RHEL 9. You must install Python 3.9.14 or higher on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow the instructions in this topic to manually download, install, and build *standard Python 3.9 binaries* from source on RHEL 9 to install Python at a standard or custom location.

About this task

You must manually download, install, and build Python 3.9 from source to install Python at a custom location. You must also replace *[***CUSTOM-INSTALL-PATH***]* in the instructions with the custom path.

Before you begin

Install the following necessary developer tools and packages on your system:

- gcc
- make
- openssl-devel
- bzip2-devel
- libffi-devel
- zlib-devel

```
yum install gcc openssl-devel bzip2-devel libffi-devel zlib-devel -y
```

Note: You cannot install the “openssl-devel” package on a FIPS cluster.

Procedure

1. SSH into the host system as a root user.
2. Change directory to /opt:

```
cd /opt
```

3. Download Python 3.9 as follows:

```
curl -O https://www.python.org/ftp/python/3.9.14/Python-3.9.14.tgz
```

4. Decompress the package as follows:

```
tar -zxvf Python-3.9.14.tgz
```

5. Change directory to where you decompressed the Python 3.9 package:

```
cd /opt/Python-3.9.14
```

6. Install Python 3.9 as follows:

   Standard location:

```
./configure --enable-shared
```

Attention: By default, Python is installed in any one of the following locations:

- /usr/bin
- /usr/local/python39/bin
- /usr/local/bin
- /opt/rh/rh-python39/root/usr/bin

Custom location:

```
./configure --enable-shared --prefix=[***CUSTOM-INSTALL-PATH***]
```

If you are installing Python 3.9 in any non-standard location, then you must specify the path using the --prefix option. You must also create a symbolic link pointing to /usr/bin or to /usr/local/bin by running the following command:

```
ln -s [***SOURCE***] [***DESTINATION***]
```

For example:

```
ln -s [***CUSTOM-INSTALL-PATH***]/python3.9 /usr/bin/python3.9
```

The --enabled-shared option is used to build a shared library instead of a static library.

7. Build Python 3.9 as follows:

a) Run the make command to compile the files:

```
make
```

b) Run the following command to put the compiled files in the default location, or in the custom location that you specified using the --prefix option:

```
make altinstall
```

Using altinstall prevents you from overwriting the system Python, if any.

c) Copy the shared compiled library files (libpython3.9.so) to the /lib64/ directory:

```
cp --no-clobber ./libpython3.9.so* /lib64/
```

The --no-clobber option is used to prevent overwriting files.

d) Change the permissions of the libpython3.9.so files as follows:

```
chmod 755 /lib64/libpython3.9.so*
```

If you see an error such as “error while loading shared libraries: libpython3.9.so.1.0: cannot open shared object file: No such file or directory”, then run the following command:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib/
```

8. Change the permission of the Python 3.9 installation directory to 755 so that Hue and its related services can leverage the binary and the site packages, as follows:

```
chmod -R 755 /usr/local/lib/python3.9
```

What to do next

Enable Cloudera Manager to use Python 3.9 installed in a custom location

If you have installed Python 3.9 at a custom location, then you must either create a symbolic link using the `alternatives` command or by adding an environment variable to the Cloudera Manager Agent in the systemd service definition files with the full path to the Python 3.9 custom-built binary.

To use the `alternatives` command, get help by running the command `alternatives --help`. You should create symbolic links from either /usr/bin/python3.9 or /usr/local/bin/python3.9 to *[***CUSTOM-INSTALL-PATH***]*/bin/python3.9 where you installed the Python 3.9 package.

Important: Using the alternatives command to set up symbolic links does not support setting up of the PYTHONHOME or PYTHONPATH variables if that is necessary for your custom version of Python 3.9 installation.

To add an environment variable to the Cloudera Manager Agent in the systemd service definition files perform the following steps:

1. SSH into the host system on which you installed Python 3.9 as a root user.
2. Open the /usr/lib/systemd/system/cloudera-scm-supervisord.service file for editing.
3. Add the following line in the cloudera-scm-supervisord.service file under the [Service] section:

```
Environment=”PYTHONBIN=[***CUSTOM-INSTALL-PATH***]/bin/python3.9”
```

4. Open the /usr/lib/systemd/system/cloudera-scm-agent.service file for editing.

5. Add the following line in the cloudera-scm-agent.service file under the [Service] section:

```
Environment=”PYTHONBIN=[***CUSTOM-INSTALL-PATH***]/bin/python3
.9”
```

Important: If your custom version of Python 3.9 installation requires setting up the PYTHONHOME or PYTHONPATH variables, then add them with additional Environment lines.

Enable Hue to use Python 3.9 installed in a custom location

If you have installed Python 3.9 in a custom location and you did not configure alternatives, then you must prepend the custom path to the custom bin directory in Cloudera Manager Clusters Hue Configuration Hue Service Environment Advanced Configuration Snippet (Safety Valve) separated by colon (:) as follows:

Key: PATH

Value: *[***CUSTOM-INSTALL-PATH***]*:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

Install the psycopg2-binary package

Install the “psycopg2-binary” package after installing Python 3.9 for using PostgreSQL as a backend database for Hue.

Related Information

Introduction to the alternatives command in Linux

Installing the psycopg2 Python package for PostgreSQL-backed Hue

Installing Python 3.10 standard package on SLES 15 SP4

For CDP Runtime 7.1.9 SP1, Cloudera Manager agents and Hue require Python 3.10 on SLES 15 SP4. You must install Python 3.10 on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow the instructions in this topic to install the standard Python 3.10 packages in a standard location on SLES 15 SP4.

About this task

Note: SLES 15 SP4 is distributed with Python 3.6. Because CDP Runtime 7.1.9 SP1 requires Python 3.10, you must install it manually.

Procedure

1. SSH into the host system as a root user.
2. Ensure the SLE-Module-Python3-15-SP4-Updates repository is present in the system before installing Python 3.10.
3. Install Python 3.10 by running the following command:

```
sudo zypper install python310 python310-devel
```

4. Verify the Python version as follows:

```
python3.10 -–version
```

What to do next

Install the psycopg2-binary package

Install the “psycopg2-binary” package after installing Python 3.10 for using PostgreSQL as a backend database for Hue.

Installing standard Python 3.10 binary on SLES 15 SP5 at a standard or custom location

For CDP Runtime 7.1.9 SP1, Cloudera Manager agents and Hue require Python 3.10 on SLES 15 SP5. You must install Python 3.10 on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow the instructions in this topic to manually download, build, and install *standard Python 3.10 binary* from source on SLES 15 SP5 to install Python at a standard or custom location.

About this task

Important: Since the standard Python 3.10 package is unavailable, you must manually download, install, and build Python 3.10 from source to install Python at a standard or custom location.

To install Python 3.10 at a custom location, you must also replace *[***CUSTOM-INSTALL-PATH***]* in the instructions with the custom path.

Before you begin

Install the following necessary developer tools and packages on your system:

- gcc
- make
- openssl-devel
- bzip2-devel
- libffi-devel
- zlib-devel
- libbz2-devel
- gzip

```
zypper install gcc openssl-devel bzip2-devel libffi-devel zlib-devel make li
bbz2-devel gzip
```

Procedure

1. SSH into the host system as a root user.
2. Change directory to /opt:

   ```
   cd /opt
   ```
3. Download Python 3.10 as follows:

   ```
   curl -O https://www.python.org/ftp/python/3.10.14/Python-3.10.14.tgz
   ```
4. Decompress the package as follows:

   ```
   tar -zxvf Python-3.10.14.tgz
   ```
5. Change directory to where you decompressed the Python 3.10 package:

   ```
   cd /opt/Python-3.10.14
   ```

6. Install Python 3.10 as follows:

   Standard location:

   ```
   ./configure --enable-shared
   ```

   Attention: By default, Python is installed in any one of the following locations:

   - /usr/bin
   - /usr/local/python310/bin
   - /usr/local/bin

   Custom location:

   ```
   ./configure --enable-shared --prefix=[***CUSTOM-INSTALL-PATH***]
   ```

   If you are installing Python 3.10 in any non-standard location, then you must specify the path using the --prefix option. You must also create a symbolic link pointing to /usr/bin or to /usr/local/bin by running the following command:

   ```
   ln -s [***SOURCE***] [***DESTINATION***]
   ```

   For example:

   ```
   ln -s [***CUSTOM-INSTALL-PATH***]/python3.10 /usr/bin/python3.10
   ```

   The --enabled-shared option is used to build a shared library instead of a static library.
7. Build Python 3.10 as follows:
   a) Run the make command to compile the files:

      ```
      make
      ```

   b) Run the following command to put the compiled files in the default location, or in the custom location that you specified using the --prefix option:

      ```
      make altinstall
      ```

      Using altinstall prevents you from overwriting the system Python, if any.
   c) Copy the shared compiled library files (libpython3.10.so) to the /lib64/ directory:

      ```
      cp --no-clobber ./libpython3.10.so* /lib64/
      ```

      The --no-clobber option is used to prevent overwriting files.
   d) Change the permissions of the libpython3.10.so files as follows:

      ```
      chmod 755 /lib64/libpython3.10.so*
      ```

   If you see an error such as “error while loading shared libraries: libpython3.10.so.1.0: cannot open shared object file: No such file or directory”, then run the following command:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib/
   ```
8. Change the permission of the Python 3.10 installation directory to 755 so that Hue and its related services can leverage the binary and the site packages, as follows:

   ```
   chmod -R 755 /usr/local/lib/python3.10
   ```

What to do next

Enable Cloudera Manager to use Python 3.10 installed in a custom location

If you have installed Python 3.10 at a custom location, then you must either create a symbolic link using the `alternatives` command or by adding an environment variable to the Cloudera Manager Agent in the systemd service definition files with the full path to the Python 3.10 custom-built binary.

To use the `alternatives` command, get help by running the command `alternatives --help`. You should create symbolic links from either /usr/bin/python3.10 or /usr/local/bin/python3.10 to *[***CUSTOM-INSTALL-PATH***]*/bin/python3.10 where you installed the Python 3.10 package.

> **Important:** Using the alternatives command to set up symbolic links does not support setting up of the PYTHONHOME or PYTHONPATH variables if that is necessary for your custom version of Python 3.10 installation.

To add an environment variable to the Cloudera Manager Agent in the systemd service definition files perform the following steps:

1. SSH into the host system on which you installed Python 3.10 as a root user.
2. Open the /usr/lib/systemd/system/cloudera-scm-supervisord.service file for editing.
3. Add the following line in the cloudera-scm-supervisord.service file under the [Service] section:

   ```
   Environment=”PYTHONBIN=[***CUSTOM-INSTALL-PATH***]/bin/python3
   .10”
   ```
4. Open the /usr/lib/systemd/system/cloudera-scm-agent.service file for editing.
5. Add the following line in the cloudera-scm-agent.service file under the [Service] section:

   ```
   Environment=”PYTHONBIN=[***CUSTOM-INSTALL-PATH***]/bin/python3
   .10”
   ```

   > **Important:** If your custom version of Python 3.10 installation requires setting up the PYTHONHOME or PYTHONPATH variables, then add them with additional Environment lines.

Enable Hue to use Python 3.10 installed in a custom location

If you have installed Python 3.10 in a custom location and you did not configure alternatives, then you must prepend the custom path to the custom bin directory in Cloudera Manager Clusters Hue Configuration Hue Service Environment Advanced Configuration Snippet (Safety Valve) separated by colon (:) as follows:

Key: PATH

Value: *[***CUSTOM-INSTALL-PATH***]*:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

Install the psycopg2-binary package

Install the “psycopg2-binary” package after installing Python 3.10 for using PostgreSQL as a backend database for Hue.

Installing Python 3.8 standard package on SLES 12 SP5, SLES 15 SP4

Cloudera Manager agents and Hue require Python 3.8 on SLES 12 SP5 or SLES 15 SP4. You must install Python 3.8.12 or higher on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow instructions in this topic to install the standard Python 3.8 packages in a standard location on SLES 12 SP5 or SLES 15 SP4.

About this task

Important: SLES 12 SP5 is distributed with Python 3.6. Because CDP requires Python 3.8, you must install it manually.

Important: SLES 15 SP4 is distributed with Python 3.10. Because CDP requires Python 3.8, you must install it manually.

Procedure

1. SSH into the host system as a root user.
2. Install Python 3.8 as follows:

   You can check the current availability of standard Python 3.8 package in the official SUSE repositories by running the following comand:

   ```
   zypper search python3.8
   ```

   If standard Python 3.8 package is available, then you can proceed to install it by running the following command:

   ```
   zypper install python3.8
   ```

   If standard Python 3.8 package is not available, then you must manually download, install, and build Python 3.8 from source to install Python at a standard or custom location. For more information, see Installing standard Python 3.8 binary on SLES 12 SP5 or SLES 15 SP4 at a standard or custom location.
3. Verify the Python version as follows:

   ```
   python3.8 -–version
   ```

What to do next

Install the psycopg2-binary package

Install the “psycopg2-binary” package after installing Python 3.8 for using PostgreSQL as a backend database for Hue.

Related Information

Introduction to the alternatives command in Linux

Installing the psycopg2 Python package for PostgreSQL-backed Hue

Installing standard Python 3.8 binary on SLES 12 SP5 or SLES 15 SP4 at a standard or custom location

Cloudera Manager agents and Hue require Python 3.8 on SLES 12 SP5 or SLES 15 SP4. You must install Python 3.8.12 or higher on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow the instructions in this topic to manually download, build, and install *standard Python 3.8 binary* from source on SLES 12 SP5 or SLES 15 SP4 to install Python at a standard or custom location.

About this task

You must manually download, install, and build Python 3.8 from source to install Python at a standard or custom location. To install Python 3.8 at a custom location, you must also replace *[***CUSTOM-INSTALL-PATH***]* in the instructions with the custom path.

Before you begin

Install the following necessary developer tools and packages on your system:

- gcc
- make

- openssl-devel
- bzip2-devel
- libffi-devel
- zlib-devel
- libbz2-devel
- gzip

```
zypper install gcc openssl-devel bzip2-devel libffi-devel zlib-devel make li
bbz2-devel gzip
```

Procedure

1. SSH into the host system as a root user.
2. Change directory to /opt:

```
cd /opt
```

3. Download Python 3.8 as follows:

```
curl -O https://www.python.org/ftp/python/3.8.17/Python-3.8.17.tgz
```

4. Decompress the package as follows:

```
tar -zxvf Python-3.8.17.tgz
```

5. Change directory to where you decompressed the Python 3.8 package:

```
cd /opt/Python-3.8.17
```

6. Install Python 3.8 as follows:

Standard location:

```
./configure --enable-shared
```

Attention: By default, Python is installed in any one of the following locations:

- /usr/bin
- /usr/local/python38/bin
- /usr/local/bin

Custom location:

```
./configure --enable-shared --prefix=[***CUSTOM-INSTALL-PATH***]
```

If you are installing Python 3.8 in any non-standard location, then you must specify the path using the --prefix option. You must also create a symbolic link pointing to /usr/bin or to /usr/local/bin by running the following command:

```
ln -s [***SOURCE***] [***DESTINATION***]
```

For example:

```
ln -s [***CUSTOM-INSTALL-PATH***]/python3.8 /usr/bin/python3.8
```

The --enabled-shared option is used to build a shared library instead of a static library.

7. Build Python 3.8 as follows:

a) Run the make command to compile the files:

```
make
```

b) Run the following command to put the compiled files in the default location, or in the custom location that you specified using the --prefix option:

```
make altinstall
```

Using altinstall prevents you from overwriting the system Python, if any.

c) Copy the shared compiled library files (libpython3.8.so) to the /lib64/ directory:

```
cp --no-clobber ./libpython3.8.so* /lib64/
```

The --no-clobber option is used to prevent overwriting files.

d) Change the permissions of the libpython3.8.so files as follows:

```
chmod 755 /lib64/libpython3.8.so*
```

If you see an error such as "error while loading shared libraries: libpython3.8.so.1.0: cannot open shared object file: No such file or directory", then run the following command:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib/
```

8. Change the permission of the Python 3.8 installation directory to 755 so that Hue and its related services can leverage the binary and the site packages, as follows:

```
chmod -R 755 /usr/local/lib/python3.8
```

What to do next

Enable Cloudera Manager to use Python 3.8 installed in a custom location

If you have installed Python 3.8 at a custom location, then you must either create a symbolic link using the `alternatives` command or by adding an environment variable to the Cloudera Manager Agent in the systemd service definition files with the full path to the Python 3.8 custom-built binary.

To use the `alternatives` command, get help by running the command `alternatives --help`. You should create symbolic links from either /usr/bin/python3.8 or /usr/local/bin/python3.8 to *[***CUSTOM-INSTALL-PATH***]*/bin/python3.8 where you installed the Python 3.8 package.

Important: Using the alternatives command to set up symbolic links does not support setting up of the PYTHONHOME or PYTHONPATH variables if that is necessary for your custom version of Python 3.8 installation.

To add an environment variable to the Cloudera Manager Agent in the systemd service definition files perform the following steps:

1. SSH into the host system on which you installed Python 3.8 as a root user.
2. Open the /usr/lib/systemd/system/cloudera-scm-supervisord.service file for editing.
3. Add the following line in the cloudera-scm-supervisord.service file under the [Service] section:

```
Environment="PYTHONBIN=[***CUSTOM-INSTALL-PATH***]/bin/python3.8"
```

4. Open the /usr/lib/systemd/system/cloudera-scm-agent.service file for editing.

5. Add the following line in the cloudera-scm-agent.service file under the [Service] section:

```
Environment=”PYTHONBIN=[***CUSTOM-INSTALL-PATH***]/bin/python3
.8”
```

Important: If your custom version of Python 3.8 installation requires setting up the PYTHONHOME or PYTHONPATH variables, then add them with additional Environment lines.

Enable Hue to use Python 3.8 installed in a custom location

If you have installed Python 3.8 in a custom location and you did not configure alternatives, then you must prepend the custom path to the custom bin directory in Cloudera Manager Clusters Hue Configuration Hue Service Environment Advanced Configuration Snippet (Safety Valve) separated by colon (:) as follows:

Key: PATH

Value: *[***CUSTOM-INSTALL-PATH***]*:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

Install the psycopg2-binary package

Install the “psycopg2-binary” package after installing Python 3.8 for using PostgreSQL as a backend database for Hue.

Related Information

Introduction to the alternatives command in Linux

Installing the psycopg2 Python package for PostgreSQL-backed Hue

Installing Python 3.8 on Oracle UEK 8.8

Cloudera Manager agents and Hue require Python 3.8. You must install Python 3.8.12 or higher on all cluster hosts before installing Cloudera Manager and adding services to your cluster. Follow instructions in this topic to install the standard Python 3.8 binaries at a standard location on Oracle UEK 8.8.

Before you begin

Install the Extra Packages for Enterprise Linux (EPEL) as follows:

```
dnf install epel-release
```

Procedure

1. SSH into the host system as a root user.
2. Install Python 3.8 by running the follwing command:

```
dnf install python38 python38-devel
```

Attention: By default, Python is installed in any one of the following locations:

- /usr/bin
- /usr/bin/python3.8-config
- /usr/local/bin

You can check the location at which Python 3.8 is installed by running the following commands:

```
whereis python3.8
```

Sample output:

```
python3: /usr/bin/python3.8 /usr/bin/python3.8-config /usr/bin/python3.8-
x86_64-config /usr/bin/python3 /usr/lib/python3.6 /usr/lib/python3.8 /usr/
```

```
lib64/python3.6 /usr/lib64/python3.8 /usr/include/python3.6m /usr/include/
python3.8 /usr/share/man/man1/python3.1.gz
```

```
echo $PATH
```

Sample output:

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

3. Verify the Python version as follows:

```
python3.8 --version
```

Sample output:

```
Python 3.8.16
```

What to do next

Install the psycopg2-binary package

Install the “psycopg2-binary” package after installing Python 3.8 for using PostgreSQL as a backend database for Hue.

Related Information

Introduction to the alternatives command in Linux

Installing the psycopg2 Python package for PostgreSQL-backed Hue

Storage Space Planning for Cloudera Manager

This topic helps you plan for the storage needs and data storage locations used by the Cloudera Manager Server and the Cloudera Management Service to store metrics and data.

Minimum Required Role: Full Administrator. This feature is not available when using Cloudera Manager to manage Data Hub clusters.

Cloudera Manager tracks metrics of services, jobs, and applications in many background processes. All of these metrics require storage. Depending on the size of your organization, this storage can be local or remote, disk-based or in a database, managed by you or by another team in another location.

Most system administrators are aware of common locations like /var/log/ and the need for these locations to have adequate space. Failing to plan for the storage needs of all components of the Cloudera Manager Server and the Cloudera Management Service can negatively impact your cluster in the following ways:

- The cluster might not be able to retain historical operational data to meet internal requirements.
- The cluster might miss critical audit information that was not gathered or retained for the required length of time.
- Administrators might be unable to research past events or health status.
- Administrators might not have historical MR1, YARN, or Impala usage data when they need to reference or report on them later.
- There might be gaps in metrics collection and charts.
- The cluster might experience data loss due to filling storage locations to 100% of capacity. The effects of such an event can impact many other components.

The main theme here is that you must architect your data storage needs well in advance. You must inform your operations staff about your critical data storage locations for each host so that they can provision your infrastructure adequately and back it up appropriately. Make sure to document the discovered requirements in your internal build documentation and run books.

This topic describes both local disk storage and RDBMS storage. This distinction is made both for storage planning and also to inform migration of roles from one host to another, preparing backups, and other lifecycle management events.

The following tables provide details about each individual Cloudera Management service to enable Cloudera Manager administrators to make appropriate storage and lifecycle planning decisions.

Table 17: Cloudera Manager Server

Configuration Topic	Cloudera Manager Server Configuration
Default Storage Location	RDBMS: Any Supported RDBMS. Disk: Cloudera Manager Server Local Data Storage Directory (command_storage_path) on the host where the Cloudera Manager Server is configured to run. This local path is used by Cloudera Manager for storing data, including command result files. Critical configurations are not stored in this location. Default setting: /var/lib/cloudera-scm-server/
Storage Configuration Defaults, Minimum, or Maximum	There are no direct storage defaults relevant to this entity.
Where to Control Data Retention or Size	The size of the Cloudera Manager Server database varies depending on the number of managed hosts and the number of discrete commands that have been run in the cluster. To configure the size of the retained command results in the Cloudera Manager Administration Console, select AdministrationSettings and edit the following property: **Command Eviction Age** Length of time after which inactive commands are evicted from the database. Default is two years.
Sizing, Planning & Best Practices	The Cloudera Manager Server database is the most vital configuration store in a Cloudera Manager deployment. This database holds the configuration for clusters, services, roles, and other necessary information that defines a deployment of Cloudera Manager and its managed hosts. Make sure that you perform regular, verified, remotely-stored backups of the Cloudera Manager Server database.

Table 18: Cloudera Management Service - Service Monitor Configuration

Configuration Topic	Service Monitor Configuration
Default Storage Location	/var/lib/cloudera-service-monitor/ on the host where the Service Monitor role is configured to run.
Storage Configuration Defaults / Minimum / Maximum	• 10 GiB Services Time Series Storage • 1 GiB Impala Query Storage • 1 GiB YARN Application Storage Total: ~12 GiB Minimum (No Maximum)

Configuration Topic	Service Monitor Configuration
Where to Control Data Retention or Size	Service Monitor data growth is controlled by configuring the maximum amount of storage space it can use. To configure data retention in Cloudera Manager Administration Console: **1.** Go the Cloudera Management Service. **2.** Click the Configuration tab. **3.** Select Scope Service Monitor or Cloudera Management Service (Service-Wide) . **4.** Select Category Main . **5.** Locate the *PROPERTYNAME* property or search for it by typing its name in the Search box. **Time-Series Storage** The approximate amount of disk space dedicated to storing time series and health data. When the store has reached its maximum size, it deletes older data to make room for newer data. The disk usage is approximate because the store only begins deleting data when it reaches the limit. Note that Cloudera Manager stores time-series data at a number of different data granularities, and these granularities have different effective retention periods. The Service Monitor stores metric data not only as raw data points but also as ten-minute, hourly, six-hourly, daily, and weekly summary data points. Raw data consumes the bulk of the allocated storage space and weekly summaries consume the least. Raw data is retained for the shortest amount of time while weekly summary points are unlikely to ever be deleted. Select Cloudera Management ServiceCharts Library tab in Cloudera Manager for information about how space is consumed within the Service Monitor. These pre-built charts also show information about the amount of data retained and time window covered by each data granularity. **Impala Storage** The approximate amount of disk space dedicated to storing Impala query data. When the store reaches its maximum size, it deletes older data to make room for newer queries. The disk usage is approximate because the store only begins deleting data when it reaches the limit. **YARN Storage** The approximate amount of disk space dedicated to storing YARN application data. When the store reaches its maximum size, it deletes older data to make room for newer applications. The disk usage is approximate because Cloudera Manager only begins deleting data when it reaches the limit. **6.** Enter a Reason for change, and then click Save Changes to commit the changes.
Sizing, Planning, and Best Practices	The Service Monitor gathers metrics about configured roles and services in your cluster and also runs active health tests. These health tests run regardless of idle and use periods, because they are always relevant. The Service Monitor gathers metrics and health test results regardless of the level of activity in the cluster. This data continues to grow, even in an idle cluster.

Table 19: Cloudera Management Service - Host Monitor

Configuration Topic	Host Monitor Configuration
Default Storage Location	/var/lib/cloudera-host-monitor/ on the host where the Host Monitor role is configured to run.
Storage Configuration Defaults / Minimum/ Maximum	Default (and minimum): 10 GiB Host Time Series Storage

Configuration Topic	Host Monitor Configuration
Where to Control Data Retention or Size	Host Monitor data growth is controlled by configuring the maximum amount of storage space it can use. See *Data Storage for Monitoring Data*. To configure these data retention configuration properties in the Cloudera Manager Administration Console: **1.** Go the Cloudera Management Service. **2.** Click the Configuration tab. **3.** Select Scope Host Monitor or Cloudera Management Service (Service-Wide). **4.** Select Category Main . **5.** Locate each property or search for it by typing its name in the Search box. **Time-Series Storage** The approximate amount of disk space dedicated to storing time series and health data. When the store reaches its maximum size, it deletes older data to make room for newer data. The disk usage is approximate because the store only begins deleting data when it reaches the limit. Note that Cloudera Manager stores time-series data at a number of different data granularities, and these granularities have different effective retention periods. Host Monitor stores metric data not only as raw data points but also as summaries of ten minute, one hour, six hour, one day, and one week increments. Raw data consumes the bulk of the allocated storage space and weekly summaries consume the least. Raw data is retained for the shortest amount of time, while weekly summary points are unlikely to ever be deleted. See the Cloudera Management Service Charts Library tab in Cloudera Manager for information on how space is consumed within the Host Monitor. These pre-built charts also show information about the amount of data retained and the time window covered by each data granularity. **6.** Enter a Reason for change, and then click Save Changes to commit the changes.
Sizing, Planning and Best Practices	The Host Monitor gathers metrics about host-level items of interest (for example: disk space usage, RAM, CPU usage, swapping, etc) and also informs host health tests. The Host Monitor gathers metrics and health test results regardless of the level of activity in the cluster. This data continues to grow fairly linearly, even in an idle cluster.

Table 20: Cloudera Management Service - Event Server

Configuration Topic	Event Server Configuration
Default Storage Location	/var/lib/cloudera-scm-eventserver/ on the host where the Event Server role is configured to run.
Storage Configuration Defaults	5,000,000 events retained
Where to Control Data Retention or Minimum /Maximum	The amount of storage space the Event Server uses is influenced by configuring how many discrete events it can retain. To configure data retention in Cloudera Manager Administration Console, **1.** Go the Cloudera Management Service. **2.** Click the Configuration tab. **3.** Select Scope Event Server or Cloudera Management Service (Service-Wide). **4.** Select CategoryMain. **5.** Edit the following property: **Maximum Number of Events in the Event Server Store** The maximum size of the Event Server store, in events. When this size is exceeded, events are deleted starting with the oldest first until the size of the store is below this threshold **6.** Enter a Reason for change, and then click Save Changes to commit the changes.

Configuration Topic	Event Server Configuration
Sizing, Planning, and Best Practices	The Event Server is a managed Lucene index that collects relevant events that happen within your cluster, such as results of health tests, log events that are created when a log entry matches a set of rules for identifying messages of interest and makes them available for searching, filtering and additional action. You can view and filter events on the Diagnostics Events tab of the Cloudera Manager Administration Console. You can also poll this data using the Cloudera Manager API. **Note:** The Cloudera Management Service role Alert Publisher sources all the content for its work by regularly polling the Event Server for entries that are marked to be sent out using SNMP or SMTP(S). The Alert Publisher is not discussed because it has no noteworthy storage requirements of its own.

Table 21: Cloudera Management Service - Reports Manager

Configuration Topic	Reports Manager Configuration
Default Storage Location	RDBMS: Any Supported RDBMS. Disk: /var/lib/cloudera-scm-headlamp/ on the host where the Reports Manager role is configured to run.
Storage Configuration Defaults	RDBMS: There are no configurable parameters to directly control the size of this data set. Disk: There are no configurable parameters to directly control the size of this data set. The storage utilization depends not only on the size of the HDFS fsimage, but also on the HDFS file path complexity. Longer file paths contribute to more space utilization.
Where to Control Data Retention or Minimum / Maximum	The Reports Manager uses space in two main locations: on the Reports Manager host and on its supporting database. Cloudera recommends that the database be on a separate host from the Reports Manager host for process isolation and performance.
Sizing, Planning, and Best Practices	Reports Manager downloads the fsimage from the NameNode (every 60 minutes by default) and stores it locally to perform operations against, including indexing the HDFS filesystem structure. More files and directories results in a larger fsimage, which consumes more disk space. Reports Manager has no control over the size of the fsimage. If your total HDFS usage trends upward notably or you add excessively long paths in HDFS, it might be necessary to revisit and adjust the amount of local storage allocated to the Reports Manager. Periodically monitor, review, and adjust the local storage allocation.

Table 22: Cloudera Navigator - Navigator Audit Server

Configuration Topic	Navigator Audit Server Configuration
Default Storage Location	Any Supported RDBMS.
Storage Configuration Defaults	Default: 90 Days retention

Configuration Topic	Navigator Audit Server Configuration
Where to Control Data Retention or Min/Max	Navigator Audit Server storage usage is controlled by configuring how many days of data it can retain. Any older data is purged. To configure data retention in the Cloudera Manager Administration Console: **1.** Go the Cloudera Management Service. **2.** Click the Configuration tab. **3.** Select Scope Navigator Audit Server or Cloudera Management Service (Service-Wide). **4.** Select CategoryMain. **5.** Locate the Navigator Audit Server Data Expiration Period property or search for it by typing its name in the Search box. **Navigator Audit Server Data Expiration Period** In Navigator Audit Server, purge audit data of various auditable services when the data reaches this age in days. By default, Navigator Audit Server keeps data about audits for 90 days. **6.** Click Save Changes to commit the changes.
Sizing, Planning, and Best Practices	The size of the Navigator Audit Server database directly depends on the number of audit events the cluster's audited services generate. Normally the volume of HDFS audits exceeds the volume of other audits (all other components like MRv1, Hive and Impala read from HDFS, which generates additional audit events). The average size of a discrete HDFS audit event is ~1 KB. For a busy cluster of 50 hosts with ~100K audit events generated per hour, the Navigator Audit Server database would consume ~2.5 GB per day. To retain 90 days of audits at that level, plan for a database size of around 250 GB. If other configured cluster services generate roughly the same amount of data as the HDFS audits, plan for the Navigator Audit Server database to require around 500 GB of storage for 90 days of data. Notes: • Individual Hive and Impala queries themselves can be very large. Since the query itself is part of an audit event, such audit events consume space in proportion to the length of the query. • The amount of space required increases as activity on the cluster increases. In some cases, Navigator Audit Server databases can exceed 1 TB for 90 days of audit events. Benchmark your cluster periodically and adjust accordingly. To map Cloudera Navigator versions to Cloudera Manager versions, see *Product Compatibility Matrix for Cloudera Navigator*.

Table 23: Cloudera Navigator - Navigator Metadata Server

Configuration Topic	Navigator Metadata Server Configuration
Default Storage Location	RDBMS: Any Supported RDBMS. Disk: /var/lib/cloudera-scm-navigator/ on the host where the Navigator Metadata Server role is configured to run.
Storage Configuration Defaults	RDBMS: There are no exposed defaults or configurations to directly cull or purge the size of this data set. Disk: There are no configuration defaults to influence the size of this location. You can change the location itself with the Navigator Metadata Server Storage Dir property. The size of the data in this location depends on the amount of metadata in the system (HDFS fsimage size, Hive Metastore size) and activity on the system (the number of MapReduce Jobs run, Hive queries executed, etc).

Configuration Topic	Navigator Metadata Server Configuration
Where to Control Data Retention or Min/Max	RDBMS: The Navigator Metadata Server database should be carefully tuned to support large volumes of metadata. Disk: The Navigator Metadata Server index (an embedded Solr instance) can consume lots of disk space at the location specified for the Navigator Metadata Server Storage Dir property. Ongoing maintenance tasks include purging metadata from the system.
Sizing, Planning, and Best Practices	Memory: See *Navigator Metadata Server Tuning*. RDBMS: The database is used to store policies and authorization data. The dataset is small, but this database is also used during a Solr schema upgrade, where Solr documents are extracted and inserted again in Solr. This has same space requirements as above use case, but the space is only used temporarily during product upgrades. Use the product compatibility matrix to map Cloudera Navigator and Cloudera Manager versions. Disk: This filesystem location contains all the metadata that is extracted from managed clusters. The data is stored in Solr, so this is the location where Solr stores its index and documents. Depending on the size of the cluster, this data can occupy tens of gigabytes. A guideline is to look at the size of HDFS fsimage and allocate two to three times that size as the initial size. The data here is incremental and continues to grow as activity is performed on the cluster. The rate of growth can be on order of tens of megabytes per day.

General Performance Notes

When possible:

- For entities that use an RDBMS, install the database on a separate host from the service, and consolidate roles that use databases on as few servers as possible.
- Provide a dedicated spindle to the RDBMS or datastore data directory to avoid disk contention with other read/write activity.

Cluster Lifecycle Management with Cloudera Manager

Cloudera Manager clusters that use parcels to provide Cloudera Runtime and other components require adequate disk space in the following locations:

Table 24: Parcel Lifecycle Management

Parcel Lifecycle Path (default)	Notes
Local Parcel Repository Path (/opt/cloudera/parcel-repo)	This path exists only on the host where Cloudera Manager Server (cloudera-scm-server) runs. The Cloudera Manager Server stages all new parcels in this location as it fetches them from any external repositories. Cloudera Manager Agents are then instructed to fetch the parcels from this location when the administrator distributes the parcel using the Cloudera Manager Administration Console or the Cloudera Manager API. Sizing and Planning The default location is /opt/cloudera/parcel-repo but you can configure another local filesystem location on the host where Cloudera Manager Server runs. Provide sufficient space to hold all the parcels you download from all configured Remote Parcel Repository URLs. Cloudera Manager deployments that manage multiple clusters store all applicable parcels for all clusters. Parcels are provided for each operating system, so be aware that heterogeneous clusters (distinct operating systems represented in the cluster) require more space than clusters with homogeneous operating systems. For example, a cluster with both RHEL6.x and 7.x hosts must hold -el6 and -el7 parcels in the Local Parcel Repository Path, which requires twice the amount of space. Lifecycle Management and Best Practices Delete any parcels that are no longer in use from the Cloudera Manager Administration Console, (never delete them manually from the command line) to recover disk space in the Local Parcel Repository Path and simultaneously across all managed cluster hosts which hold the parcel. Backup Considerations Perform regular backups of this path, and consider it a non-optional accessory to backing up Cloudera Manager Server. If you migrate Cloudera Manager Server to a new host or restore it from a backup (for example, after a hardware failure), recover the full content of this path to the new host, in the /opt/cloudera/parcel-repo directory before starting any cloudera-scm-agent or cloudera-scm-server processes.
Parcel Cache (/opt/cloudera/parcel-cache)	Managed Hosts running a Cloudera Manager Agent stage distributed parcels into this path (as .parcel files, unextracted). Do not manually manipulate this directory or its files. Sizing and Planning Provide sufficient space per-host to hold all the parcels you distribute to each host. You can configure Cloudera Manager to remove these cached .parcel files after they are extracted and placed in /opt/cloudera/parcels/. It is not mandatory to keep these temporary files but keeping them avoids the need to transfer the .parcel file from the Cloudera Manager Server repository should you need to extract the parcel again for any reason. To configure this behavior in the Cloudera Manager Administration Console, select AdministrationSettingsParcelsRetain Downloaded Parcel Files

Parcel Lifecycle Path (default)	Notes
Host Parcel Directory (/opt/cloudera/parcels)	Managed cluster hosts running a Cloudera Manager Agent extract parcels from the /opt/cloudera/parcel-cache directory into this path upon parcel activation. Many critical system symlinks point to files in this path and you should never manually manipulate its contents. Sizing and Planning Provide sufficient space on each host to hold all the parcels you distribute to each host. Be aware that the typical Runtime or CDH parcel size is approximately 2 GB per parcel, and some third party parcels can exceed 3 GB. If you maintain various versions of parcels staged before and after upgrading, be aware of the disk space implications. You can configure Cloudera Manager to automatically remove older parcels when they are no longer in use. As an administrator you can always manually delete parcel versions not in use, but configuring these settings can handle the deletion automatically, in case you forget. To configure this behavior in the Cloudera Manager Administration Console, select AdministrationSettingsParcels and configure the following property: **Automatically Remove Old Parcels** This parameter controls whether parcels for old versions of an activated product should be removed from a cluster when they are no longer in use. The default value is Disabled. **Number of Old Parcel Versions to Retain** If you enable Automatically Remove Old Parcels, this setting specifies the number of old parcels to keep. Any old parcels beyond this value are removed. If this property is set to zero, no old parcels are retained. The default value is 3.

Table 25: Management Service Lifecycle - Space Reclamation Tasks

Task	Description
Activity Monitor (One-time)	The Activity Monitor only works against a MapReduce (MR1) service, not YARN. So if your deployment has fully migrated to YARN and no longer uses a MapReduce (MR1) service, your Activity Monitor database is no longer growing. If you have waited longer than the default Activity Monitor retention period (14 days) to address this point, then the Activity Monitor has already purged it all for you and your database is mostly empty. If your deployment meets these conditions, consider cleaning up by dropping the Activity Monitor database (again, only when you are satisfied that you no longer need the data or have confirmed that it is no longer in use) and the Activity Monitor role.
Service Monitor and Host Monitor (One-time)	For those who used Cloudera Manager version 4.x and have now upgraded to version 5.x: The Service Monitor and Host Monitor were migrated from their previously-configured RDBMS into a dedicated time series store used solely by each of these roles respectively. After this happens, there is still legacy database connection information in the configuration for these roles. This was used to allow for the initial migration but is no longer being used for any active work. After the above migration has taken place, the RDBMS databases previously used by the Service Monitor and Host Monitor are no longer used. Space occupied by these databases is now recoverable. If appropriate in your environment (and you are satisfied that you have long-term backups or do not need the data on disk any longer), you can drop those databases.
Ongoing Space Reclamation	Cloudera Management Services are automatically rolling up, purging or otherwise consolidating aged data for you in the background. Configure retention and purging limits per-role to control how and when this occurs. These configurations are discussed per-entity above. Adjust the default configurations to meet your space limitations or retention needs.

Log File Storage Space

All cluster hosts write out separate log files for each role instance assigned to the host. Cluster administrators can monitor and manage the disk space used by these roles and configure log rotation to prevent log files from consuming too much disk space.

Configure Network Names

You must configure each host in the cluster to ensure that all members can communicate with each other.

About this task

Important: Cloudera Runtime requires IPv4. IPv6 is not supported.

Tip: When bonding, use the bond0 IP address as it represents all aggregated links.

Procedure

1. Set the hostname to a unique name (not localhost).

```
sudo hostnamectl set-hostname foo-1.example.com
```

2. Edit /etc/hosts with the IP address and fully qualified domain name (FQDN) of each host in the cluster. You can add the unqualified name as well.

```
1.1.1.1  foo-1.example.com  foo-1
2.2.2.2  foo-2.example.com  foo-2
3.3.3.3  foo-3.example.com  foo-3
4.4.4.4  foo-4.example.com  foo-4
```

Important:

- The canonical name of each host in /etc/hosts must be the FQDN (for example myhost-1.example .com), not the unqualified hostname (for example myhost-1). The canonical name is the first entry after the IP address.
- Do not use aliases, either in /etc/hosts or in configuring DNS.
- Unqualified hostnames (short names) must be unique in a Cloudera Manager instance. For example, you cannot have both *HOST01.EXAMPLE.COM* and *HOST01.STANDBY.EXAMPLE.COM* managed by the same Cloudera Manager Server.

3. Edit /etc/sysconfig/network with the FQDN of this host only:

```
HOSTNAME=foo-1.example.com
```

4. Verify that each host consistently identifies to the network:

a) Run uname -a and check that the hostname matches the output of the hostname command.

b) Run /sbin/ifconfig and note the value of inet addr in the eth0 (or bond0) entry, for example:

```
eth0      Link encap:Ethernet  HWaddr 00:0C:29:A4:E8:97
          inet addr:172.29.82.176  Bcast:172.29.87.255  Mask:255.255.2
48.0
...
```

c) Run host -v -t A $(hostname) and verify that the output matches the hostname command. The IP address should be the same as reported by ifconfig for eth0 (or bond0):

```
Trying "foo-1.example.com"
...
;; ANSWER SECTION:
```

```
foo-1.example.com. 60 IN A 172.29.82.176
```

Important: If the host command is not installed on your system, then install it by running the following command:

- RHEL:

```
yum install bind-utils
```

- Ubuntu:

```
apt install bind9-host
```

- SLES:

```
zypper in bind-utils
```

Setting SELinux Mode

Security-Enhanced Linux (SELinux) allows you to set access control through policies. If you are having trouble deploying Runtime or CDH with your policies, set SELinux in permissive mode on each host before you deploy Runtime or CDH on your cluster.

About this task

Note: CDP Private Cloud Base, with the exception of Cloudera Navigator Encrypt, is supported on platforms with Security-Enhanced Linux (SELinux) enabled and in enforcing mode. Cloudera is not responsible for SELinux policy development, support, or enforcement. If you experience issues running Cloudera software with SELinux enabled, contact your OS provider for assistance.

If you are using SELinux in enforcing mode, Cloudera Support can request that you disable SELinux or change the mode to permissive to rule out SELinux as a factor when investigating reported issues.

Procedure

1. Check the SELinux state:

   ```
   getenforce
   ```

2. If the output is either Permissive or Disabled, you can skip this task and continue to Disabling the Firewall to disable the firewall on each host in your cluster. If the output is enforcing, continue to the next step.
3. Open the /etc/selinux/config file (in some systems, the /etc/sysconfig/selinux file).
4. Change the line SELINUX=enforcing to SELINUX=permissive.
5. Save and close the file.
6. Restart your system or run the following command to disable SELinux immediately:

   ```
   setenforce 0
   ```

   After you have installed and deployed Runtime or CDH, you can re-enable SELinux by changing SELINUX=permissive back to SELINUX=enforcing in /etc/selinux/config (or /etc/sysconfig/selinux), and then running the following command to immediately switch to enforcing mode:

   ```
   setenforce 1
   ```

   If you are having trouble getting Cloudera Software working with SELinux, contact your OS vendor for support. Cloudera is not responsible for developing or supporting SELinux policies.

Disabling the Firewall

To disable the firewall on each host in your cluster, perform the following steps on each host.

Procedure

1. If the iptables command is not installed on your system, then install it by running the following command:
   - RHEL:
     ```
     sudo yum install iptables
     ```
   - SLES:
     ```
     sudo zypper install iptables
     ```
   - Ubuntu:
     ```
     sudo apt-get install iptables
     ```
2. For iptables, save the existing rule set:
   ```
   sudo iptables-save > ~/firewall.rules
   ```
3. Disable the firewall:
   - RHEL 7 compatible:
     ```
     sudo systemctl disable firewalld
     sudo systemctl stop firewalld
     ```
   - SLES:
     ```
     sudo chkconfig SuSEfirewall2_setup off
     sudo chkconfig SuSEfirewall2_init off
     sudo rcSuSEfirewall2 stop
     ```
   - Ubuntu:
     ```
     sudo service ufw stop
     ```

Enable an NTP Service

Runtime requires that you configure a Network Time Protocol (NTP) service on each machine in your cluster. Most operating systems include the ntpd service for time synchronization.

About this task

Some operating systems use chronyd by default instead of ntpd. If chronyd is running (on any OS), Cloudera Manager uses it to determine whether the host clock is synchronized. Otherwise, Cloudera Manager uses ntpd.

Note: If you are using ntpd to synchronize your host clocks, but chronyd is also running, Cloudera Manager relies on chronyd to verify time synchronization, even if it is not synchronizing properly. This can result in Cloudera Manager reporting clock offset errors, even though the time is correct.

To fix this, either configure and use chronyd or disable it and remove it from the hosts.

Important:

You should verify whether ntpd or chronyd is already installed and running. If ntpd or chronyd is already installed and running, then you can skip enabling NTP service.

Please refer your OS vendor's documentation to install and configure either ntpd or chronyd.

Impala Requirements

To perform as expected, Impala depends on the availability of the software, hardware, and configurations described in the following sections.

Product Compatibility Matrix

The ultimate source of truth about compatibility between various versions of Cloudera Runtime, Cloudera Manager, and various Runtime components is the Product Compatibility Matrix.

Supported Operating Systems

The relevant supported operating systems and versions for Impala are the same as for the corresponding Cloudera Runtime platforms. For details, see the *Operating System Requirements* topic.

Hive Metastore and Related Configuration

Impala can interoperate with data stored in Hive, and uses the same infrastructure as Hive for tracking metadata about schema objects such as tables and columns. The following components are prerequisites for Impala:

To install the metastore:

1. Install a MySQL or PostgreSQL database. Start the database if it is not started after installation.
2. Download the MySQL Connector or the PostgreSQL connector and place it in the /usr/share/java/ directory.
3. Use the appropriate command line tool for your database to create the metastore database.
4. Use the appropriate command line tool for your database to grant privileges for the metastore database to the hive user.
5. Modify hive-site.xml to include information matching your particular database: its URL, username, and password. You will copy the hive-site.xml file to the Impala Configuration Directory later in the Impala installation process.

Java Dependencies

Although Impala is primarily written in C++, it does use Java to communicate with various Hadoop components:

- The officially supported JVMs for Impala are the OpenJDK JVM and Oracle JVM. Other JVMs might cause issues, typically resulting in a failure at `impalad` startup. In particular, the JamVM used by default on certain levels of Ubuntu systems can cause `impalad` to fail to start.
- Internally, the `impalad` daemon relies on the JAVA_HOME environment variable to locate the system Java libraries. Make sure the `impalad` service is not run from an environment with an incorrect setting for this variable.
- All Java dependencies are packaged in the impala-dependencies.jar file, which is located at /usr/lib/impala/lib/. These map to everything that is built under fe/target/dependency.

Networking Configuration Requirements

As part of ensuring best performance, Impala attempts to complete tasks on local data, as opposed to using network connections to work with remote data. To support this goal, Impala matches the hostname provided to each Impala daemon with the IP address of each DataNode by resolving the hostname flag to an IP address. For Impala to work with local data, use a single IP interface for the DataNode and the Impala daemon on each machine. Ensure that the Impala daemon's hostname flag resolves to the IP address of the DataNode. For single-homed machines, this is usually automatic, but for multi-homed machines, ensure that the Impala daemon's hostname resolves to the correct interface. Impala tries to detect the correct hostname at start-up, and prints the derived hostname at the start of the log in a message of the form:

```
Using hostname: impala-daemon-1.example.com
```

In the majority of cases, this automatic detection works correctly. If you need to explicitly set the hostname, do so by setting the --hostname flag.

Note: RedHat Users with server configurations that utilise networks of 10Gb/s or more might need to adjust the vm.min_free_kbytes parameter to a significantly increased value to ensure ample space for storing packet backlogs before the operating system processes them. This helps prevent packet drops and avoids connection timeouts.

These timeouts might manifest as EndDataStream errors or generic Impala Connection timeouts.

Additional details regarding modifications to the vm.min_free_kbytes parameter and the reasons for its necessity can be found here.

Glibc Version Requirement

Impala daemons should be deployed on nodes using the same Glibc version since different Glibc version supports different Unicode standard version. Not using the same Glibc version might result in inconsistent UTF-8 behavior when UTF8_MODE is set to true.

Hardware Requirements

The memory allocation should be consistent across Impala executor nodes. A single Impala executor with a lower memory limit than the rest can easily become a bottleneck and lead to suboptimal performance.

This guideline does not apply to coordinator-only nodes.

Hardware Requirements for Optimal Join Performance

During join operations, portions of data from each joined table are loaded into memory. Data sets can be very large, so ensure your hardware has sufficient memory to accommodate the joins you anticipate completing.

While requirements vary according to data set size, the following is generally recommended:

- CPU

  Impala version 2.2 and higher uses the SSSE3 instruction set, which is included in newer processors.

  Note: This required level of processor is the same as in Impala version 1.x. The Impala 2.0 and 2.1 releases had a stricter requirement for the SSE4.1 instruction set, which has now been relaxed.
- Memory

  128 GB or more recommended, ideally 256 GB or more. If the intermediate results during query processing on a particular node exceed the amount of memory available to Impala on that node, the query writes temporary work data to disk, which can lead to long query times. Note that because the work is parallelized, and intermediate results for aggregate queries are typically smaller than the original data, Impala can query and join tables that are much larger than the memory available on an individual node.
- JVM Heap Size for Catalog Server

  4 GB or more recommended, ideally 8 GB or more, to accommodate the maximum numbers of tables, partitions, and data files you are planning to use with Impala.
- Storage

  DataNodes with 12 or more disks each. I/O speeds are often the limiting factor for disk performance with Impala. Ensure that you have sufficient disk space to store the data Impala will be querying.

OS Memory Map Area Requirement for Impala

Default settings for max_map_count can be insufficient for Impala installations that run with many concurrent queries. We recommend increasing max_map_count to avoid potential failures due to exhausting memory mapping limits under heavy load.

max_map_count is the OS virtual memory parameter and defines the maximum number of memory map areas that a process can use.

To increase the max_map_countparameter in your OS and to make the above setting durable, refer to your OS documentation.

- For example, if you are using RHEL 9, add the following line to /etc/sysctl.conf:

```
/etc/sysctl.conf:vm.max_map_count=8000000
```

- Reload the config as root: sysctl -p.
- Check the new value: cat /proc/sys/vm/max_map_count.
- Restart.

User Account Requirements

For user account requirements, see the topic User Account Requirements in the Impala documentation.

Runtime Cluster Hosts and Role Assignments

Cluster hosts can be broadly described as master hosts, utility hosts, gateway hosts, or worker hosts.

- Master hosts run Hadoop master processes such as the HDFS NameNode and YARN Resource Manager.
- Utility hosts run other cluster processes that are not master processes such as Cloudera Manager and one or more Hive Metastores.
- Gateway hosts are client access points for launching jobs in the cluster. The number of gateway hosts required varies depending on the type and size of the workloads.
- Worker hosts primarily run DataNodes and other distributed processes such as Impala.

Important: Cloudera recommends that you always enable high availability when Runtime is used in a production environment.

The following tables describe the recommended role allocations for different cluster sizes. Note that these configurations take into account services dependencies that might not be obvious. For example, running Atlas or Ranger requires also running HBase, Kafka, Solr, and ZooKeeper. For details see Service Dependences in Cloudera Manager.

Attention: When High Availability (HA) is enabled and the total number of nodes is under 10, you must carefully plan the composition of the worker nodes. That is the utility nodes and master nodes. If you decide that your development cluster is to be HA enabled, you must add the HA configuration for at least 3-10 hosts for seamless performance.

3 - 10 Worker Hosts without High Availability

<table>
<tr><th>Master Hosts</th><th>Utility Hosts</th><th>Gateway Hosts</th><th>Worker Hosts</th></tr>
<tr>
<td>Master Host 1:
• NameNode
• YARN ResourceManager
• YARN Queue Manager
• JobHistory Server
• ZooKeeper
• Kudu master
• Spark History Server
• HBase master
• Schema Registry
• Ozone Manager (A maximum of 3 Ozone Managers are supported)
• Storage Container Manager</td>
<td colspan="2">One host for all Utility and Gateway roles:
• Secondary NameNode
• Cloudera Manager
• Cloudera Manager Management Service
• Cruise Control
• Hive Metastore
• HiveServer2
• Impala Catalog Server
• Impala StateStore
• Hue
• Oozie
• Gateway configuration
• HBase backup master
• Ranger Admin, Tagsync, Usersync servers
• Ranger KMS
• Ranger RMS
• Atlas server
• Solr server (CDP-INFRA-SOLR instance to support Atlas)
• Streams Messaging Manager
• Streams Replication Manager Service
• ZooKeeper
• Knox: One KnoxGateway service on utility or gateway hosts.
• S3 Gateway</td>
<td>3 - 10 Worker Hosts:
• DataNode
• NodeManager
• Impalad
• Kudu tablet server
• Kafka Broker
• Kafka Connect
• HBase RegionServer
• Solr server (For Cloudera Search)
• Streams Replication Manager Driver
• ZooKeeper (Recommend 3 servers total)
• Ozone DataNode</td>
</tr>
</table>

3 - 20 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • JournalNode • FailoverController • YARN ResourceManager • YARN Queue Manager • ZooKeeper • JobHistory Server • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 2: • NameNode • JournalNode • FailoverController • YARN ResourceManager • YARN Queue Manager • ZooKeeper • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 3: • Kudu master (Kudu requires an odd number of masters for HA.) • Spark History Server • JournalNode (requires dedicated disk) • ZooKeeper • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager	Utility Host 1: • Cloudera Manager • Cloudera Manager Management Service • Cruise Control • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (CDP-INFRA-SOLR instance to support Atlas) • Streams Messaging Manager • Streams Replication Manager Service • Knox: One KnoxGateway service for HA. Instead of utility, you can also select gateway hosts. Utility Host 2: • Hive Metastore • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (CDP-INFRA-SOLR instance to support Atlas) • Knox: One KnoxGateway service for HA. Instead of utility, you can also select gateway hosts.	One or more Gateway Hosts: • Hue • HiveServer2 • Gateway configuration • S3 Gateway	3 - 20 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server • Kafka Broker (Recommend 3 brokers minimum) • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search, recommend 3 servers minimum) • Streams Replication Manager Driver • Ozone DataNode

20 - 80 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • JournalNode • FailoverController • YARN ResourceManager • YARN Queue Manager • ZooKeeper • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 2: • NameNode • JournalNode • FailoverController • YARN ResourceManager • YARN Queue Manager • ZooKeeper • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 3: • ZooKeeper • JournalNode • JobHistory Server • Spark History Server • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager	Utility Host 1: • Cloudera Manager • Cruise Control • Hive Metastore • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (CDP-INFRA-SOLR instance to support Atlas) • Streams Messaging Manager • Streams Replication Manager Service Utility Host 2: • Cloudera Manager Management Service • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (CDP-INFRA-SOLR instance to support Atlas)	One or more Gateway Hosts: • Hue • HiveServer2 • Gateway configuration • S3 Gateway Two or more Gateway Hosts: • Knox: Two KnoxGateway services, one on each of the first two Gateway Hosts for HA. • S3 Gateway	20 - 80 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server • Kafka Broker (Recommend 3 brokers minimum) • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search, recommend 3 servers minimum) • Streams Replication Manager Driver • Ozone DataNode

80 - 200 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • JournalNode • FailoverController • YARN ResourceManager • YARN Queue Manager • ZooKeeper • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 2: • NameNode • JournalNode • FailoverController • YARN ResourceManager • YARN Queue Manager • ZooKeeper • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 3: • ZooKeeper • JournalNode • JobHistory Server • Spark History Server • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager	Utility Host 1: • Cloudera Manager • Cruise Control • Streams Messaging Manager • Streams Replication Manager Service Utility Host 2: • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie Utility Host 3: • Host Monitor Utility Host 4: • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server Utility Host 5: • Hive Metastore • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server Utility Host 6: • Reports Manager Utility Host 7: • Service Monitor	One or more Gateway Hosts: • Hue • HiveServer2 • Gateway configuration • S3 Gateway Two or more Gateway Hosts: • Knox: Two KnoxGateway services, one on each of the first two Gateway Hosts for HA. • S3 Gateway	80 - 200 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server (Recommend 100 tablet servers maximum) • Kafka Broker (Recommend 3 brokers minimum) • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search, recommend 3 servers minimum) • Streams Replication Manager Driver • Ozone DataNode

200 - 500 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • JournalNode • FailoverController • ZooKeeper • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 2: • NameNode • JournalNode • FailoverController • ZooKeeper • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 3: • YARN ResourceManager • YARN Queue Manager • ZooKeeper • JournalNode • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 4: • YARN ResourceManager • YARN Queue Manager • ZooKeeper • JournalNode • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 5: • JobHistory Server • Spark History Server • ZooKeeper • JournalNode • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Cloudera recommends no more than three masters for Kudu and HBase.	Utility Host 1: • Cloudera Manager • Cruise Control • Streams Messaging Manager • Streams Replication Manager Service Utility Host 2: • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie Utility Host 3: • Host Monitor Utility Host 4: • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (CDP-INFRA-SOLR instance to support Atlas) Utility Host 5: • Hive Metastore • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (CDP-INFRA-SOLR instance to support Atlas) Utility Host6: • Reports Manager Utility Host 7: • Service Monitor	One or more Gateway Hosts: • Hue • HiveServer2 • Gateway configuration • S3 Gateway Two or more Gateway Hosts: • Knox: Two KnoxGateway services, one on each of the first two Gateway Hosts for HA. • S3 Gateway	200 - 500 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server (Recommend 100 tablet servers maximum) • Kafka Broker (Recommend 3 brokers minimum) • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search, recommend 3 servers minimum) • Streams Replication Manager Driver • Ozone DataNode

500 -1000 Worker Hosts with High Availability

Master Hosts	Utility Hosts	Gateway Hosts	Worker Hosts
Master Host 1: • NameNode • JournalNode • FailoverController • ZooKeeper • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 2: • NameNode • JournalNode • FailoverController • ZooKeeper • Kudu master • HBase master • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 3: • YARN ResourceManager • YARN Queue Manager • ZooKeeper • JournalNode • Kudu master • HBase master • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 4: • YARN ResourceManager • YARN Queue Manager • ZooKeeper • JournalNode • Schema Registry • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Master Host 5: • JobHistory Server • Spark History Server • ZooKeeper • JournalNode • Ozone Manager (A maximum of 3 Ozone Managers are supported) • Storage Container Manager Cloudera recommends no more than three masters for Kudu and HBase.	Utility Host 1: • Cloudera Manager • Cruise Control • Streams Messaging Manager • Streams Replication Manager Service Utility Host 2: • Hive Metastore • Impala Catalog Server • Impala StateStore • Oozie Utility Host 3: • Host Monitor Utility Host 4: • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (CDP-INFRA-SOLR instance to support Atlas) Utility Host 5: • Hive Metastore • Ranger Admin, Tagsync, Usersync servers • Ranger KMS • Ranger RMS • Atlas server • Solr server (CDP-INFRA-SOLR instance to support Atlas) Utility Host 6: • Reports Manager Utility Host 7: • Service Monitor	One or more Gateway Hosts: • Hue • HiveServer2 • Gateway configuration • S3 Gateway Two or more Gateway Hosts: • Knox: Two KnoxGateway services, one on each of the first two Gateway Hosts for HA. • S3 Gateway	500 - 1000 Worker Hosts: • DataNode • NodeManager • Impalad • Kudu tablet server (Recommend 100 tablet servers maximum) • Kafka Broker (Recommend 3 brokers minimum) • Kafka Connect • HBase RegionServer • Solr server (For Cloudera Search, recommend 3 servers minimum) • Streams Replication Manager Driver • Ozone DataNode

Related Information

Service Dependencies in Cloudera Manager
Configuring HMS for high availability

Allocating Hosts for Key Trustee Server and Key Trustee KMS

If you are enabling data-at-rest encryption for a Cloudera Runtime cluster, Cloudera recommends that you isolate the Key Trustee Server from other enterprise data hub (EDH) services by deploying the Key Trustee Server on dedicated hosts in a separate cluster managed by Cloudera Manager.

Cloudera also recommends deploying Key Trustee KMS on dedicated hosts in the same cluster as the EDH services that require access to Key Trustee Server. This architecture helps users avoid having to restart the Key Trustee Server when restarting a cluster.

For production environments in general, or if you have enabled high availability for HDFS and are using data-at-rest encryption, Cloudera recommends that you enable high availability for Key Trustee Server and Key Trustee KMS.

Configuring Local Package and Parcel Repositories

Cloudera hosts two types of software repositories that you can use to install products such as Cloudera Manager or Cloudera Runtime—parcel repositories and package repositories. These repositories are effective solutions in most cases, but custom installation solutions are sometimes required.

For example, using the Cloudera-hosted software repositories requires client access over the Internet. Typical installations use the latest available software. In some scenarios, these behaviors might not be desirable, such as:

- You need to install older product versions. For example, in a Runtime cluster, all hosts must run the same Runtime version. After completing an initial installation, you may want to add hosts. This could be to increase the size of your cluster to handle larger tasks or to replace older hardware.
- The hosts on which you want to install Cloudera products are not connected to the Internet, so they cannot reach the Cloudera repository (for a parcel installation, only the Cloudera Manager Server needs Internet access, but for a package installation, all cluster hosts require access to the Cloudera repository). Most organizations partition parts of their network from outside access. Isolating network segments improves security, but can add complexity to the installation process.

In both of these cases, using an internal repository allows you to meet the needs of your organization, whether that means installing specific versions of Cloudera software or installing Cloudera software on hosts without Internet access.

Understanding Package Management

Before you configure a custom package management solution in your environment, understand the concepts of package management tools and package repositories.

Package Management Tools

Packages (rpm or deb files) help ensure that installations complete successfully by satisfying package dependencies. When you install a particular package, all other required packages are installed at the same time. For example, hado op-0.20-hive depends on hadoop-0.20.

Package management tools, such as yum (RHEL), zypper (SLES), and apt-get (Ubuntu) are tools that can find and install required packages. For example, on a RHEL compatible system, you might run the command yum install hado op-0.20-hive. The yum utility informs you that the Hive package requires hadoop-0.20 and offers to install it for you. zypper and apt-get provide similar functionality.

Package Repositories

Package management tools rely on package repositories to install software and resolve any dependency requirements. For information on creating an internal repository, see *Configuring a Local Package Repository*.

Repository Configuration Files

Information about package repositories is stored in configuration files, the location of which varies according to the package management tool.

- RHEL compatible (yum): /etc/yum.repos.d
- SLES (zypper): /etc/zypp/zypper.conf

- Ubuntu (apt-get): /etc/apt/apt.conf (Additional repositories are specified using .list files in the /etc/apt/sources.list.d/ directory.)

For example, on a typical CentOS system, you might find:

```
ls -l /etc/yum.repos.d/
total 36
-rw-r--r--. 1 root root 1664 Dec  9  2015 CentOS-Base.repo
-rw-r--r--. 1 root root 1309 Dec  9  2015 CentOS-CR.repo
-rw-r--r--. 1 root root  649 Dec  9  2015 CentOS-Debuginfo.repo
-rw-r--r--. 1 root root  290 Dec  9  2015 CentOS-fasttrack.repo
-rw-r--r--. 1 root root  630 Dec  9  2015 CentOS-Media.repo
-rw-r--r--. 1 root root 1331 Dec  9  2015 CentOS-Sources.repo
-rw-r--r--. 1 root root 1952 Dec  9  2015 CentOS-Vault.repo
-rw-r--r--. 1 root root  951 Jun 24  2017 epel.repo
-rw-r--r--. 1 root root 1050 Jun 24  2017 epel-testing.repo
```

The .repo files contain pointers to one or more repositories. In the following excerpt from CentOS-Base.repo, there are two repositories defined: one named Base and one named Updates. The mirrorlist parameter points to a website that has a list of places where this repository can be downloaded.

```
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearc
h&repo=os&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&
repo=updates&infra=$infra
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

Listing Repositories

You can list the enabled repositories by running one of the following commands:

- RHEL compatible: yum repolist
- SLES: zypper repos
- Ubuntu: apt-get does not include a command to display sources, but you can determine sources by reviewing the contents of /etc/apt/sources.list and any files contained in /etc/apt/sources.list.d/.

The following shows an example of the output of yum repolist on a CentOS 7 sytstem:

```
repo id                 repo name                                           st
atus
base/7/x86_64           CentOS-7 - Base
9,591
epel/x86_64             Extra Packages for Enterprise Linux 7 - x86_64
12,382
extras/7/x86_64         CentOS-7 - Extras
  392
updates/7/x86_64        CentOS-7 - Updates                                   1
,962
repolist: 24,327
```

Configuring a Local Package Repository

You can create a package repository for Cloudera Manager either by hosting an internal web repository or by manually copying the repository files to the Cloudera Manager Server host for distribution to Cloudera Manager Agent hosts.

Creating a Permanent Internal Repository

The following sections describe how to create a permanent internal repository using Apache HTTP Server.

Setting Up a Web Server

To host an internal repository, you must install or use an existing Web server on an internal host that is reachable by the Cloudera Manager host, and then download the repository files to the Web server host.

About this task

The examples in this section use Apache HTTP Server as the Web server. If you already have a Web server in your organization, you can skip to *Downloading and Publishing the Package Repository*.

Procedure

1. Install Apache HTTP Server:

   RHEL / CentOS

   ```
   sudo yum install httpd
   ```

   SLES

   ```
   sudo zypper install httpd
   ```

   Ubuntu

   ```
   sudo apt-get install httpd
   ```
2. Start Apache HTTP Server:

   RHEL 7

   ```
   sudo systemctl start httpd
   ```

   SLES 12, Ubuntu 18

   ```
   sudo systemctl start apache2
   ```

Downloading and Publishing the Package Repository

Download the package repository for the product you want to install.

About this task

Note:

Assuming your system does not have Internet access due to security concerns, you will need to download the package to an offline storage (such as USB disk or external disk device), hand the storage to the customer, ask him to whitelist and validate it with his Information Security team and once approved, the customer should move the content of the external storage given to the system from which it could be applied.

The Cloudera sources are divided into several parts according to the type of solution your system will have.While all deployments require the CDP Private Cloud base packages (Cloudera Manager packages and Clouder Runtime parcel), other solutions might require additional deliverables such as Nifi, Data Services and more. The location of all these sources is in Cloudera online archive repository (archive.cloudera.com), and the access to it requires a license (provided by your account team). According to the license you get and based on what was purchased, access will be granted to relevant locations on the archive to download the needed deliverables.

Procedure

1. **Cloudera Manager 7**

   To download the files for a Cloudera Manager release, download the repository tarball for your operating system, located under archive.cloudera.com/p/cm7.

   You can reach that location directly by typing on your browser https://[username]:[password]@archive.cloudera.com/p/cm7 and then you should pick the relevant release you need, and in it, choose the tarball matching your deployed operating system. Once the tarball is approved to be inserted into the customer network (security scan etc.), you will need to move either into your local deployment server (where you will build a local repository for the deployment) or into the customer organizational repository (such as JFrog/Nexus etc.) from which all deliverables will be distributed to the cluster. The following example assumes a local repository is built on one of your machines that already has the httpd service installed on it.

   In that case, you should create a folder for the local repository and unpack the tarball into it, ensuring it carries read permissions (755) for all as described below:

   ```
   sudo mkdir -p /var/www/html/cloudera-repos/cm7
   ```

   ```
   tar xvf <CM package name for the relevant OS>.tar.gz -C /var/www/
   html/cloudera-repos/cm7
   ```

   ```
   sudo chmod -R 755 /var/www/html/cloudera-repos/cm7
   ```

2. Make sure that your httpd service is enabled and up and running and then visit the Repository URL http ://<*WEB_SERVER*>/cloudera-repos/ in your browser and verify the files you downloaded are present.

Important: If you do not see the list of downloaded files in your web browser, then you might have been configured not to display indexes. Verify your web browser settings.

Creating a Temporary Internal Repository

You can quickly create a temporary remote repository to deploy packages on a one-time basis. Cloudera recommends using the same host that runs Cloudera Manager, or a gateway host.

About this task

This example uses Python SimpleHTTPServer as the Web server to host the /var/www/html directory, but you can use a different directory.

Procedure

1. Download the repository you need following the instructions in *Downloading and Publishing the Package Repository*.
2. Determine a port that your system is not listening on. This example uses port 8900.
3. Start a Python SimpleHTTPServer in the /var/www/html directory:

   ```
   cd /var/www/html
   python -m SimpleHTTPServer 8900
   ```

   ```
   Serving HTTP on 0.0.0.0 port 8900 ...
   ```

4. Visit the Repository URL http://<*WEB_SERVER*>:8900/cloudera-repos/ in your browser and verify the files you downloaded are present.

Configuring Hosts to Use the Internal Repository

After you establish the repository, modify the client configuration to use it.

Note: In case the local repository is configured to use TLS/SSL to secure the communication, make sure to add the repository's root and any intermediate CA certificates to the operating system's truststore.

OS	Procedure

RHEL compatible	Create /etc/yum.repos.d/cloudera-repo.repo files on cluster hosts with the following content, where *<WEB_SERVER>* is the hostname of the Web server:
	`[cloudera-repo]`
	`name=cloudera-repo`
	`baseurl=http://<WEB_SERVER>/cloudera-repos/cm7`
	`enabled=1`
	`gpgcheck=0`

SLES	Use the zypper utility to update client system repository information by issuing the following command: `zypper addrepo http://<WEB_SERVER>/cm <ALIAS>`

Ubuntu	Create /etc/apt/sources.list.d/cloudera-repo.list files on all cluster hosts with the following content, where *<WEB_SERVER>* is the hostname of the Web server:
	`deb http://<WEB_SERVER>/cm <CODENAME> <COMPONENTS>`
	You can find the *<CODENAME>* and *<COMPONENTS>* variables in the ./conf/distributions file in the repository.
	After creating the .list file, run the following command:
	`sudo apt-get update`

Configuring a Local Parcel Repository

You can create a parcel repository for Cloudera Manager either by hosting an internal Web repository or by manually copying the repository files to the Cloudera Manager Server host for distribution to Cloudera Manager Agent hosts.

Related Information

Overview of Parcels

Using an Internally Hosted Remote Parcel Repository

The following sections describe how to use an internal Web server to host a parcel repository.

Related Information

Overview of Parcels

Setting Up a Web Server

To host an internal repository, you must install or use an existing Web server on an internal host that is reachable by the Cloudera Manager host, and then download the repository files to the Web server host.

About this task

The examples on this page use Apache HTTP Server as the Web server. If you already have a Web server in your organization, you can skip to *Downloading and Publishing the Parcel Repository*.

Procedure

1. Install Apache HTTP Server:

   RHEL / CentOS

   ```
   sudo yum install httpd
   ```

   SLES

   ```
   sudo zypper install httpd
   ```

   Ubuntu

   ```
   sudo apt-get install httpd
   ```

2. Edit the Apache HTTP Server configuration file (/etc/httpd/conf/httpd.conf by default) to add or edit the following line in the <IfModule mime_module> section:

   ```
       AddType application/x-gzip .gz .tgz .parcel
   ```

   If the <IfModule mime_module> section does not exist, you can add it in its entirety as follows:

   Note: This example configuration was modified from the default configuration provided after installing Apache HTTP Server on RHEL 7.

   ```
   <IfModule mime_module>
       #
       # TypesConfig points to the file containing the list of mappings from
       # filename extension to MIME-type.
       #
       TypesConfig /etc/mime.types
       #
       # AddType allows you to add to or override the MIME configuration
       # file specified in TypesConfig for specific file types.
       #
       #AddType application/x-gzip .tgz
       #
       # AddEncoding allows you to have certain browsers uncompress
       # information on the fly. Note: Not all browsers support this.
       #
       #AddEncoding x-compress .Z
       #AddEncoding x-gzip .gz .tgz
       #
       # If the AddEncoding directives above are commented-out, then you
       # probably should define those extensions to indicate media types:
       #
       AddType application/x-compress .Z
       AddType application/x-gzip .gz .tgz .parcel

       #
       # AddHandler allows you to map certain file extensions to "handlers":
       # actions unrelated to filetype. These can be either built into the se
   rver
       # or added with the Action directive (see below)
       #
       # To use CGI scripts outside of ScriptAliased directories:
       # (You will also need to add "ExecCGI" to the "Options" directive.)
       #
       #AddHandler cgi-script .cgi

       # For type maps (negotiated resources):
   ```

```
    #AddHandler type-map var

    #
    # Filters allow you to process content before it is sent to the client
.
    #
    # To parse .shtml files for server-side includes (SSI):
    # (You will also need to add "Includes" to the "Options" directive.)
    #
    AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
</IfModule>
```

Warning: Skipping this step could result in an error message Hash verification failed when trying to download the parcel from a local repository, especially in Cloudera Manager 6 and higher.

3. Start Apache HTTP Server:

   RHEL 7

   ```
   sudo systemctl start httpd
   ```

   SLES 12, Ubuntu 18

   ```
   sudo systemctl start apache2
   ```

Downloading and Publishing the Parcel Repository

Download the parcels that you want to install and publish the parcel directory.

Procedure

1. To download the files for the latest Clouder Runtime 7 release, create the following new folder on the Web server host which holds the repository you have created for the Cloudera Manager deliverables:

   ```
   sudo mkdir -p /var/www/html/cloudera-repos/cdh7
   ```

2. Connect to the cloudera online archive located under archive.cloudera.com/p/cdh7. You can reach that location directly by typing on your browser https://[username]:[password]@archive.cloudera.com/p/cdh7 and then you should choose the relevant release you need, and in it, choose parcels and download the files related to your operating system (.parcel, .parcel.sha1, .parcel.sha256). In case needed, pass the files to the customer for approval, and once they are approved, ask the customer to place them in the target folder created above (or into the customer organizational repository in case such is used).

   Once the files were moved into the folder, ensure that they have the same permissions as the cm7 files by running the following command:

   ```
   sudo chmod -R 755 /var/www/html/cloudera-repos/cdh7
   ```

3. Visit the Repository URL http://<*WEB_SERVER*>/cloudera-repos/ in your browser and verify the files you downloaded are present. If you do not see anything, your Web server may have been configured to not show indexes.

Related Information

Overview of Parcels

Configuring Cloudera Manager to Use an Internal Remote Parcel Repository

In Cloudera Manager's parcel settings, add a path to the internal parcel repository.

About this task

Note: In case the local repository is configured to use TLS/SSL to secure the communication, make sure to add the repository's root and any intermediate CA certificates to Cloudera Manager's truststore.

Procedure

1. Use one of the following methods to open the parcel settings page:
   - Navigation bar:
     a. Click the parcel icon in the left navigation bar or click Hosts and click the Parcels tab.
     b. Click the Configuration button.
   - Menu:
     a. Select AdministrationSettings.
     b. Select CategoryParcels.
2. Enter the path to the parcel. For example: http://<*WEB_SERVER*>/cloudera-parcels/cdh7/7.0.3.1/

Using a Local Parcel Repository

To use a local parcel repository, complete the following steps:

Procedure

1. Open the Cloudera Manager Admin Console and click Parcels in the left-side navigation menu.
2. Select Configuration and verify that you have a Local Parcel Repository path set. By default, the directory is /opt/cloudera/parcel-repo.
3. Remove any Remote Parcel Repository URLs that you are not using, including ones that point to Cloudera archives.
4. Add the parcel you want to use to the local parcel repository directory that you specified. For instructions on downloading parcels, see Downloading and Publishing the Parcel Repository above.
5. In the command line, navigate to the local parcel repository directory.
6. Create a SHA1 hash for the parcel you added and save it to a file named *PARCEL_NAME*.parcel.sha.

   For example, the following command generates a SHA1 hash for the parcel :

   ```
   sha1sum  | awk '{ print $1 }' > .sha
   ```
7. Change the ownership of the parcel and hash files to cloudera-scm:

   ```
   sudo chown -R cloudera-scm:cloudera-scm /opt/cloudera/parcel-repo/*
   ```
8. In the Cloudera Manager Admin Console, click Parcels page in the left-side navigation menu.
9. Click Check for New Parcels and verify that the new parcel appears.
10. Download, distribute, and activate the parcel.

Configuring /tmp directory for cluster hosts

You must ensure that the /tmp directory is writable so that Cloudera Manager can use the directory for installing hosts and for generating certificates and credential scripts.

About this task

By default, the /tmp directory is writable. If you have changed the default permissions for the /tmp directory, then you must reset the permissions so that the /tmp directory is writable (having the drwxrwxrwt permission). Cloudera Manager uses the /tmp directory when you install hosts using the Cloudera Manager server and for generating certificates and credential scripts. Cloudera Manager’s single file installer also uses the /tmp directory.

Procedure

1. SSH into the host system as a root user.
2. Run the following command to set write access permission to the /tmp directory:

```
chmod 1777 /tmp
```

3. Verify the permission of the /tmp directory by running the list command as follows:

```
ls -la
```

   The permissions of the /tmp directory should show drwxrwxrwt.

Results

Your /tmp directory is now writable on your cluster hosts.

What to do next

Repeat this task on every host in your cluster.

Production Installation: Installing Cloudera Manager, Cloudera Runtime, and Managed Services

This procedure is recommended for installing Cloudera Manager and Cloudera Runtime for production environments. For a non-production trial install see *Installing the Cloudera Private Cloud Base Trial*.

Before you begin the installation, make sure you have reviewed the requirements and other considerations described in *Before You Install*.

Important: The Cloudera Manager agent uses Python version 3, however it is not necessary to set Python 3 as the default Python. If you want to use Cloudera Runtime 7.1.7 SP2 or below versions, you must set Python 2 as the system default Python on all cluster hosts. Use the following steps to proceed further:

1. ssh to each host.
2. Run alternatives --set python /usr/bin/python2
3. Restart the agent: systemctl restart cloudera-scm-agent
4. Run the host validation.
5. Inspect Network Performance

If you want to install Cloudera Manager with high availability, there are several additional steps, including installing the Cloudera Manager software on an additional host. See Configuring Cloudera Manager for High Availability.

Important: Do not start the additional instance of Cloudera Manager until all of the installation steps including setting up a cluster are completed.

The general steps in the installation procedure are as follows:

- Step 1: Configure a Repository for Cloudera Manager on page 143
- Step 2: Install Java Development Kit on page 145
- Step 3. Install and Configure Databases on page 150
- Step 4: Deploy Cloudera Manager Server and Cloudera Manager Agents on page 204
- Step 5: Set up and configure the Cloudera Manager database on page 206
- Step 6: Start the Cloudera Manager Server and Agents on page 218
- Step 8: Set Up a Cluster Using the Wizard on page 229
- (optional) Enable high availability for Cloudera Manager on page 245

Important: When you install the Cloudera Manager Agent package, it creates a Linux user account called cloudera-scm on each host. This user account is added to the Linux group called wheel. The cloudera-scm user account does not need to be part of the wheel group, and you can remove it manually. Note that during an upgrade, the cloudera-scm user account is added back to the wheel group.

Related Information

Use case 1: Use Cloudera Manager to generate internal CA and corresponding certificates

Use case 2: Enabling Auto-TLS with an intermediate CA signed by an existing Root CA

Use case 3: Enabling Auto-TLS with Existing Certificates

Step 1: Configure a Repository for Cloudera Manager

Cloudera Manager is installed using package management tools such as yum for RHEL compatible systems. These tools depend on access to repositories to install software. Cloudera maintains Internet-accessible repositories for Runtime and Cloudera Manager installation files.

You can also create your own internal repository for hosts that do not have Internet access. For more information on creating an internal repository for Cloudera Manager, see Configuring Hosts to Use the Internal Repository on page 137.

Note: The username and password fields are pre-populated. The usernames and passwords are backfilled in .repo and .list files when users download them from archive.cloudera.com. When downloading these files, the logged in user credentials are substituted in the .list (Debian/Ubuntu) and .repo (RedHat/Centos) files, so the users does not have to enter the credentials manually in the file.

To use the Cloudera repository:

For RHEL compatible

1. Download the cloudera-manager.repo file for your OS version to the /etc/yum.repos.d/ directory on the Cloudera Manager Server host.

   You can download the repository file at the following location:

   - RHEL 9

     ```
     https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/cm7/[**CLOUDERA
      MANAGER VERSION**]/redhat9/yum/cloudera-manager.repo
     ```
   - RHEL 8

     ```
     https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/cm7/[**CLOUDERA
      MANAGER VERSION**]/redhat8/yum/cloudera-manager.repo
     ```
   - RHEL 7

     ```
     https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/cm7/[**CLOUDERA
      MANAGER VERSION**]/redhat7/yum/cloudera-manager.repo
     ```

   For example:

   ```
   sudo wget https://[username]:[password]@archive.cloudera.com/p/cm7/[**Cl
   oudera Manager version**]/redhat9/yum/cloudera-manager.repo
   ```
2. Edit the *CLOUDERA-MANAGER.REPO* file and replace *USERNAME:PASSWORD* with your Cloudera authentication credentials. For example:

   ```
   [cloudera-manager]
   name=Cloudera Manager 7.11.3
   baseurl=https://myUsername:myPassword@archive.cloudera.com/p/cm7/7.11.3/
   redhat9/yum/
   ```

```
gpgkey=https://myUsername:myPassword@archive.cloudera.com/p/cm7/7.11.3/r
edhat9/yum/RPM-GPG-KEY-cloudera
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md
[postgresql10]
name=Postgresql 10
baseurl=https://archive.cloudera.com/postgresql10/redhat9/
gpgkey=https://archive.cloudera.com/postgresql10/redhat9/RPM-GPG-KEY-PGD
G-10
enabled=1
gpgcheck=1
module_hotfixes=true
```

3. Import the repository signing GPG key:

- RHEL 9 compatible:

```
sudo rpm --import https://[USERNAME]:[PASSWORD]@archive.cloudera.com/
p/cm7/[**CLOUDERA MANAGER VERSION**]/redhat9/yum/RPM-GPG-KEY-cloudera
```

- RHEL 8 compatible:

```
sudo rpm --import https://[USERNAME]:[PASSWORD]@archive.cloudera.com/
p/cm7/[**CLOUDERA MANAGER VERSION**]/redhat8/yum/RPM-GPG-KEY-cloudera
```

- RHEL 7 compatible:

```
sudo rpm --import https://[USERNAME]:[PASSWORD]@archive.cloudera.com/
p/cm7/[**CLOUDERA MANAGER VERSION**]/redhat7/yum/RPM-GPG-KEY-cloudera
```

4. Continue to *Step 2: Install Java Development Kit*.

For SLES

1. Update your system package index by running:

```
sudo zypper refresh
```

2. Add the repo using zypper addrepo.

You can find the URL on the Cloudera Manager Download Page.

For example:

```
sudo zypper addrepo -f https://[USERNAME]:
[PASSWORD]@archive.cloudera.com/p/cm7/7.11.3/sles15/yum/cloudera-
manager.repo
```

3. Edit the */ETC/ZYPP/REPOS.D/CLOUDERA-MANAGER.REPO* file and replace *USERNAME:PASSWORD* with your Cloudera authentication credentials. For example:

```
[cloudera-manager]bn
name=Cloudera Manager 7.11.3
baseurl=https://myUsername:myPassword@archive.cloudera.com/p/cm7/7.11.3/
sles15/yum/
gpgkey=https://myUsername:myPassword@archive.cloudera.com/p/cm7/7.11.3/s
les15/yum/RPM-GPG-KEY-cloudera
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md

[postgresql10]
```

```
name=Postgresql 10
baseurl=https://archive.cloudera.com/postgresql10/sles15/
gpgkey=https://archive.cloudera.com/postgresql10/sles15/RPM-GPG-KEY-P
GDG-10
enabled=1
gpgcheck=1
module_hotfixes=true
```

4. Import the repository signing GPG key (substitute the correct URL):

```
sudo rpm --import https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/
cm7/[**CLOUDERA MANAGER VERSION**]/sles15/yum/RPM-GPG-KEY-cloudera
```

5. Continue to *Step 2: Install Java Development Kit*.

For Ubuntu

1. Download the cloudera-manager.list file for your OS version to the /etc/apt/sources.list.d/ directory on the Cloudera Manager Server host.

   You can find the URL on the Cloudera Manager Download Page.
2. Edit the *CLOUDERA-MANAGER.LIST* file and replace *USERNAME:PASSWORD* with your Cloudera authentication credentials. For example:

```
# Cloudera Manager 7.11.3
# Changeme: change username and password below to match your license
deb [arch=amd64] https://myUsername:myPassword@archive.cloudera.com/p/
cm7/7.11.3/ubuntu2004/apt bionic-cm7.11.3 contrib
```

3. Import the repository signing GPG key (substitute the correct URL):

```
wget https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/
cm7/[**CLOUDERA MANAGER VERSION**]/ubuntu2004/apt/archive.key
sudo apt-key add archive.key
```

4. Import the PostgreSQL repository signing key:

```
wget https://archive.cloudera.com/postgresql10/deb/ACCC4CF8.asc
sudo apt-key add ACCC4CF8.asc
```

5. Update your system package index by running:

```
sudo apt-get update
```

6. Continue to *Step 2: Install Java Development Kit*.

Step 2: Install Java Development Kit

Cloudera Private Cloud Base requires a JDK installed on all hosts., you can either install OpenJDK or a Oracle JDK directly from Oracle.

There are several options for installing a JDK on your Cloudera Private Cloud Base hosts:

- Install OpenJDK 8[*] on the Cloudera Manager server host and then allow Cloudera Manager to install OpenJDK 8[*] on its managed hosts. This is the automatic option.
- Manually install a supported JDK on all cluster hosts before installing Cloudera software.

  Please see the Cloudera Support Matrix for detailed information about supported JDKs.

[*] Azul OpenJDK, OpenJDK 8, OpenJDK 11, and OpenJDK 17 are TCK certified for CDP.

Important:

Cloudera strongly recommends installing Oracle JDK at /usr/java/<jdk-version> and OpenJDK at /usr/lib/jvm, which allows Cloudera Manager to auto-detect and use the correct JDK version.

When installing the relevant JDK version using yum/zypper/apt-get (depending on the OS) ensure that the OS will handle the installation location and ensure the environment is updated accordingly so Cloudera services will auto-detect the correct version.

Also, it is possible (though not recommended) to deploy multiple JDK versions on the same server, as long as it is done using the above package deployment process, and using the OS tools (such as "alternatives --config java" application in RHEL) to set the relevant JDK release to use.

Requirements:

- The installed JDK must be a supported version as documented in .
- The same version of the JDK must be installed on each cluster host.

Important: On SLES platforms, do not install or try to use the IBM Java version bundled with the SLES distribution.

Related Information

Java Requirements

Java Requirements

Installing OpenJDK on Cloudera Manager

Prior to installing the Cloudera Manager packages, you must install a JDK for Cloudera Manager on the Cloudera Manager Server host. This section describes how to install OpenJDK on the Cloudera Manager Server host using your package manager. Also, you have the option of installing the Oracle JDK. See Installing Oracle JDK for CDP Runtime documentation for instructions.

Important:

Note that each time a system that uses OpenJDK is being patched for security on the OS level, the OpenJDK minor release will be updated as well (as it comes with the OS and not as an external kit that was manually fetched like Oracle JDK or Azul JDK).

In some cases, the customer security team might ask to update the chain certificate with the updated JDK cacert that came with the updated patched OpenJDK (whatever comes with the OS).

When that happens, you will need to regenerate the certificate chains and the CSR based. For more information, please contact Cloudera support and the customer project team.

Important: Cloudera Manager Server with JDK 8 does not support G1GC.

Important: If your are using OpenJDK versions 1.8 u421 or 11.0.24 and have enabled Kerberos, you may experience authentication errors when running cluster services. To work around this problem:

1. Log in to the Cloudera Manager Admin Console.
2. Go to Administration Settings .
3. Select the Advanced category.
4. Locate the JVM Arguments for Java-based services parameter and enter the following:

   ```
   -Dsun.security.krb5.disableReferrals=true
   ```

5. Restart any stale services.

* Azul OpenJDK, OpenJDK 8, OpenJDK 11, and OpenJDK 17 are TCK certified for CDP.

- RHEL Compatible

```
sudo yum install java-1.8.0-openjdk-devel
```

- SLES

```
sudo zypper install java-1_8_0-openjdk-devel
```

- Ubuntu

```
sudo apt-get install openjdk-8-jdk
```

You can use Cloudera Manager to install Open JDK 8[*] on the remaining cluster hosts in an upcoming step. Continue to *Step 3. Installing Cloudera Manager Server*.

Installing OpenJDK for CDP Runtime

This section is optional. The CDP Runtime requires a JDK to be installed on all cluster hosts prior to Cloudera Manager and Runtime installation. Cloudera Manager can automatically install an OpenJDK that has been pre-packaged by Cloudera. This OpenJDK provided by Cloudera may not be the latest version.

About this task

If you do not want to use the OpenJDK provided by Cloudera, you must install a JDK of your choice on all hosts in the cluster. These instructions describes how to install the OpenJDK package provided by your Operating System Vendor.

Manually install a supported JDK on all cluster hosts before installing Cloudera software.

Note:

Using this method has the advantage of getting JDK updates easily from your Operating System Vendor. The JDK can be updated by running the Operating System's package update tool. If you wish to install a different JDK, then see the instructions provided with that JDK.

Please see the Cloudera Support Matrix for detailed information about supported JDKs.

Note that the path for the default truststore for OpenJDK 8[*] is jre/lib/security/cacerts.

- The package names used when installing the OpenJDK 8[*], OpenJDK 11[*], and OpenJDK 17[*] are different and are noted in the steps below.
- The path for the default truststore has changed from (OpenJDK 8[*]) jre/lib/security/cacerts to (OpenJDK 11[*]) lib/security/cacerts
- The path to default truststore for OpenJDK 17[*] is lib/security/cacerts
- See the following blog post for general information about migrating to Java 11: All You Need to Know For Migrating To Java 11.
- See the following blog post for general information about migrating to Java 17: Migrate to Java 17.

Important: When you install Cloudera Private Cloud Base , Cloudera Manager includes an option to install Oracle JDK. De-select this option before continuing with the installation.

You must install a supported version of OpenJDK. If your deployment uses a version of OpenJDK lower than 1.8.0_181, see *TLS Protocol Error with OpenJDK*.

Note: If you intend to enable Auto-TLS, note the following:

You can specify a PEM file containing trusted CA certificates to be imported into the Auto-TLS truststore. If you want to use the certificates in the cacerts truststore that comes with OpenJDK, you must convert the truststore to PEM format first. However, OpenJDK ships with some intermediate certificates that cannot be imported into the Auto-TLS truststore. You must remove these certificates from the PEM file before importing the PEM file into the Auto-TLS truststore. This is not required when upgrading to OpenJDK from a cluster where Auto-TLS has already been enabled.

Procedure

1. Log in to each host and run the command for the version of the JDK you want to install:

RHEL

OpenJDK 8 *

```
sudo yum install java-1.8.0-openjdk-devel
```

OpenJDK 11*

```
sudo yum install java-11-openjdk-devel
```

OpenJDK 17*

```
sudo yum install java-17-openjdk-devel
```

Ubuntu

OpenJDK 8*

```
sudo apt-get install openjdk-8-jdk
```

OpenJDK 11*

```
sudo apt install openjdk-11-jdk
```

OpenJDK 17*

```
sudo apt install openjdk-17-jdk
```

SLES

OpenJDK 8*

```
sudo zypper install java-1_8_0-openjdk-devel
```

OpenJDK 11*

```
sudo zypper install java-11-openjdk-devel
```

OpenJDK 17*

```
sudo zypper --non-interactive install java-17-openjdk-devel
```

Note: If JDK11 is being installed, then use 'AES256-cts' Kerberos Encryption Type while setting up Kerberos.

2. Tune the JDK (OpenJDK 11* or OpenJDK 17*).

OpenJDK 11* or OpenJDK 17* uses new defaults for garbage collection and other Java options specified when launching Java processes. Due to these changes you may need to tune the garbage collection by adjusting the Java options used to run cluster services, which are configured separately for each service using the service's

* Azul OpenJDK, OpenJDK 8, OpenJDK 11, and OpenJDK 17 are TCK certified for CDP.

configuration parameters. To locate the correct parameter, log in to the Cloudera Manager Admin Console, go to the cluster and service you want to configure and search for "Java Configuration Options".

When using OpenJDK 11* or OpenJDK 17*, Cloudera Manager and most Cloudera Runtime services use G1GC as the default method of garbage collection. Java 8 used "ConcurrentMarkSweep" (CMS) for garbage collection. When using G1GC, the pauses for garbage collection are shorter, so components will usually be more responsive, but they are more sensitive to JVMs with overcommitted memory usage. See Tuning JVM Garbage Collection on page 242.

Important: For OpenJDK 17*, Cloudera Manager and most services run with default GC without any custom tuning for any service.

Installing Oracle JDK for CDP Runtime

This section is optional. The CDP Runtime requires a JDK to be installed on all cluster hosts prior to Cloudera Manager and Runtime installation. Cloudera Manager can automatically install an OpenJDK that has been pre-packaged by Cloudera. This OpenJDK provided by Cloudera may not be the latest version. If you do not want to use the OpenJDK provided by Cloudera, you can follow these instructions to install the Oracle JDK.

About this task

The Oracle JDK must be installed on all cluster hosts. The Oracle JDK installer is available both as an RPM-based installer for RPM-based systems, and as a .tar.gz file. These instructions are for the RPM-based installation.

Important: Cloudera Manager Server with JDK 8 does not support G1GC.

Procedure

1. Download the latest RPM version of the Oracle JDK for Linux x64 from the support site of Oracle (assuming you have a license for that). It can be either Oracle 1.8 / Oracle 11 versions.

   Note:

   You can install only one version of JDK of the same feature release. If you try to install the newer version of the same feature release while the older version exists, the installer uninstalls the older version and installs the new version. For example, you can't install jdk-11 and jdk-11.0.1 simultaneously. If you attempt to install jdk-11.0.1 after jdk-11 is installed, the installer uninstalls jdk-11 and installs jdk-11.0.1.

   Also, every update release will be installed in a separate directory, such as /usr/lib/jvm/jdk-<VERSION>-oracle-x64 directory, where <VERSION> is a full version string such as 11.0.3 and 11.0.25.

   Additionally /usr/java/jdk-11 symbolic link that is pointing to the installation directory is created for backward compatibility.
2. The downloaded RPM will be named as jdk-11_linux-x64_bin.rpm.
3. Install the rpm by running the following command:

   ```
   sudo rpm -ivh jdk-11_linux-x64_bin.rpm
   ```
4. Repeat this procedure on all cluster hosts.

Results

After you have finished, continue to *Step 3: Install and configure Databases*.

Step 3. Install and Configure Databases

Cloudera Manager uses various databases and datastores to store information about the Cloudera Manager configuration, as well as information such as the health of the system, or task progress.

Although you can deploy different types of databases in a single environment, doing so can create unexpected complications. Cloudera recommends choosing one supported database provider for all of the Cloudera databases.

Cloudera recommends installing the databases on different hosts than the services, located in the same data center. Separating databases from services can help isolate the potential impact from failure or resource contention in one or the other. It can also simplify management in organizations that have dedicated database administrators.

For information about supported databases, see Database Requirements

Required Databases

The following components all require databases: Cloudera Manager Server, Oozie Server, Sqoop Server, Reports Manager, Hive Metastore Server, Hue Server, and Ranger.

The type of data contained in the databases and their relative sizes are as follows:

- Cloudera Manager Server - Contains all the information about services you have configured and their role assignments, all configuration history, commands, users, and running processes. This relatively small database (< 100 MB) is the most important to back up.

  Important: When you restart processes, the configuration for each of the services is redeployed using information saved in the Cloudera Manager database. If this information is not available, your cluster cannot start or function correctly. You must schedule and maintain regular backups of the Cloudera Manager database to recover the cluster in the event of the loss of this database.
- Oozie Server - Contains Oozie workflow, coordinator, and bundle data. Can grow very large. (Only available when installing CDH 5 or CDH 6 clusters.)
- Sqoop Server - Contains entities such as the connector, driver, links and jobs. Relatively small. (Only available when installing CDH 5 or CDH 6 clusters.)
- Reports Manager - Tracks disk utilization and processing activities over time. Medium-sized.
- Hive Metastore Server - Contains Hive metadata. Relatively small.
- Hue Server - Contains user account information, job submissions, and Hive queries. Relatively small.
- YARN Queue Manager - If you install CDP 7.1.9 CHF 2 or later, no database changes are required. YARN Queue Manager will continue to use your current embedded database. If iyou install CDP 7.1.9 or CDP 7.1.9 CHF 1, you must use a PostgreSQL database which stores information about queues created by YARN Queue Manager. If you choose to update from an earlier CDP 7.1.9 version to CHF 2, CDP will continue to use your PostgreSQL database and will not migrate back to the embedded database.
- Sentry Server - Contains authorization metadata. Relatively small.
- Cloudera Navigator Audit Server - Contains auditing information. In large clusters, this database can grow large. (Only available when installing CDH 5 or CDH 6 clusters.)
- Cloudera Navigator Metadata Server - Contains authorization, policies, and audit report metadata. Relatively small.(Only available when installing CDH 5 or CDH 6 clusters.)
- Ranger Admin - Contains administrative information such as Ranger users, groups, and access policies. Medium-sized.
- Ranger KMS database - Stores the encrypted keys.
- Streaming Components:
  - Schema Registry - Contains the schemas and their metadata, all the versions and branches. You can use either MySQL, Postgres, or Oracle.

    Important: For the Schema Registry database, you must set collation to be case sensitive.
  - Streams Messaging Manager Server - Contains Kafka metadata, stores metrics, and alert definitions. Relatively small.

The Host Monitor and Service Monitor services use local disk-based datastores.

The JDBC connector for your database must be installed on the hosts where you assign the Activity Monitor and Reports Manager roles.

For instructions on installing and configuring databases for Cloudera Manager, Runtime, and other managed services, see the instructions for the type of database you want to use.

Related Information

Database Requirements

Install and Configure PostgreSQL for CDP

To use a PostgreSQL database, follow these procedures. For information on compatible versions of the PostgreSQL database, see Database Requirements on page 32.

Note: The following instructions are for a dedicated PostgreSQL database for use in production environments, and are unrelated to the embedded PostgreSQL database provided by Cloudera for trial installations.

Installing Postgres JDBC Driver

You must install the required Postgres JDBC driver.

Download, extract, and copy the JDBC driver, renamed, to /usr/share/java/. If the target directory does not yet exist, create it.

Installing the Postgres JDBC Driver

1. Install the PostgreSQL JDBC driver by running the following command:

```
wget https://jdbc.postgresql.org/download/postgresql-42.<version>.jar
```

Important: Make sure to download the specific JAR file for your current JAVA version based on the chart provided in PostgreSQL JDBC Driver website.

2. Alternatively, if you would like to use the PostgreSQL JDBC driver version shipped with the OS repositories, run the following commands with a driver version that is compatible with the PostgreSQL Server version:
RHEL

```
sudo yum install postgresql-jdbc<compatible_version>
```

Ubuntu

```
sudo apt-get install libpostgresql-jdbc-java<compatible_version>
```

SLES

```
sudo zypper install postgresql-jdbc<compatible_version>
```

3. Rename the Postgres JDBC driver .jar file to postgresql-connector-java.jarand copy it to the /usr/share/java directory. The following copy command can be used if the Postgres JDBC driver .jar file is installed from the OS repositories:

```
cp /usr/share/java/postgresql-jdbc.jar /usr/share/java/postgresql-connec
tor-java.jar
```

4. Confirm that the .jar file is in the Java share directory:

```
ls /usr/share/java/
```

5. Change the access mode of the .jar file to 644:

```
chmod 644 /usr/share/java/postgresql-connector-java.jar
```

Installing PostgreSQL Server

Install the PostgreSQL packages on the PostgreSQL server.

Note:

- If you already have a PostgreSQL database set up, you can skip to the section *Configuring and Starting the PostgreSQL Server* to verify that your PostgreSQL configurations meet the requirements for Cloudera Manager.
- Make sure that the data directory, which by default is /var/lib/postgresql/data/, is on a partition that has sufficient free space.
- Cloudera Manager supports the use of a custom schema name for the Cloudera Manager Server database, but not the Runtime component databases (such as Hive and Hue). For more information, see *Scehmas* in the PostgreSQL documentation.

Install the PostgreSQL packages as follows:

RHEL

Important: To install any specific version of PostgreSQL packages, see this page.

```
sudo yum install postgresql-server
```

SLES

```
sudo zypper addrepo -t YUM http://packages.2ndquadrant.com/postgresql-z-suse
/zypper/sles-11sp3-s390x pg
sudo zypper refresh
sudo zypper in postgresql postgresql-server postgresql-contrib
```

Note: This command installs PostgreSQL 11. If you want to install a different version, you can use zypper s earch postgresql to search for an available supported version.

Ubuntu

```
sudo apt-get install postgresql
```

Installing the psycopg2 Python package for PostgreSQL database

If you are using PostgreSQL as a backend database for Hue on CDP Private Cloud Base 7, then you must install a version of the psycopg2 package to be at least 2.9.5 on all Hue hosts. The psycopg2 package is automatically installed as a dependency of Cloudera Manager Agent, but the version installed is often lower than 2.9.3.

Before you begin, you must disable the postgresql10 section from the cloudera-manager.repo file as follows:

1. SSH in to the Cloudera Manager host as an Administrator.
2. Change to the directory where you had downloaded the cloudera-manager.repo file. On RHEL, the file is present under the /etc/yum.repos.d directory.
3. Open the file for editing and update the value of the enabled property to 0 as follows:

   ```
   [postgresql10]
   name=Postgresql 10
   baseurl=https://archive.cloudera.com/postgresql10/redhat8/
   gpgkey=https://archive.cloudera.com/postgresql10/redhat8/RPM-GPG-KEY-PG
   DG-10
   enabled=0
   gpgcheck=1
   module_hotfixes=true
   ```
4. Save the file and exit.

Note: The steps to disable the postgresql10 section are applicable to all supported operating systems (CentOS, RHEL, SLES, and Ubuntu).

For CentOS RHEL OEL 7 8

The following steps apply to CentOS 7, RHEL 7, and OEL 7, CentOS 8, RHEL 8, OEL 8:

1. SSH into the Hue server host as a root user.
2. Install the psycopg2-binary package as follows:

```
pip3.8 install psycopg2-binary
```

3. Repeat these steps on all the Hue server hosts.

If you get the "Error: pg_config executable not found" error while installing the psycopg2-binary package, then run the following commands to install the postgresql, postgresql-devel, python-devel packages:

```
yum install postgresql postgresql-devel python-devel
```

For RHEL 9

The following steps apply to RHEL 9, as the minimum version of Python is 3.9:

1. SSH into the Hue server host as a root user.
2. Install pip for Python as follows:

```
yum install python3-pip -y
```

3. Add the /usr/local/bin path to the PATH environment variable:

```
export PATH=$PATH:/usr/local/bin
echo $PATH
```

4. Install the psycopg2-binary package as follows:

```
pip3 install psycopg2-binary
```

5. Repeat these steps on all the Hue server hosts.

If you get the "Error: pg_config executable not found" error while installing the psycopg2-binary package, then run the following commands to install the postgresql, postgresql-devel, python-devel packages:

```
yum install postgresql postgresql-devel python-devel
```

For SLES

1. SSH into the Hue host as a root user.
2. Install the psycopg2 package dependencies for SLES by running the following commands:

```
zypper install xmlsec1
zypper install xmlsec1-devel
zypper install xmlsec1-openssl-devel
```

3. Install the postgresql-devel package corresponding to your database version by running the following command:

```
zypper -n postgresql[***DB-VERSION***]-devel
```

4. Add the location of the installed postgresql-devel package to the PATH environment variable by running the following command:

```
export PATH=$PATH:/usr/local/bin
```

5. Install the psycopg2 package by running the following command:

```
pip3.8 install psycopg2==2.9.3 --ignore-installed
```

For Ubuntu

1. SSH into the Hue host as a root user.
2. Install the psycopg2 package dependencies for Ubuntu by running the following commands:

```
apt-get install -y xmlsec1
apt-get install libxmlsec1-openssl
apt-get install libpq-dev python3-pip -y
```

3. Install the python3-dev and libpq-dev packages by running the following command:

```
apt install python3-dev libpq-dev
```

4. Add the location of the installed postgresql-devel package to the PATH environment variable by running the following command:

```
export PATH=$PATH:/usr/local/bin
```

5. Install the psycopg2 package by running the following command:

```
pip3.8 install psycopg2==2.9.3 --ignore-installed
```

Installing the psycopg2 Python package for PostgreSQL database on a FIPS cluster (RHEL 8)

If you use PostgreSQL as a backend database for Hue on a FIPS cluster, you must install a version of the psycopg2 package from the source to be at least 2.9.5 on all Hue hosts because the psycopg2-binary package uses its version of the libssl library file which does not support FIPS.

About this task

Note: This topic provides instructions on downloading and installing the psycopg2 package from the source. This is required for using the PostgreSQL database with Hue on FIPS-enabled CDP 7.1.9 SP1 cluster and higher on RHEL 8.

Before you begin

1. Uninstall the preinstalled psycopg2 or psycopg2-binary packages.
2. Download and install the PostgreSQL database using the following commands:

```
dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL
-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm
dnf install -y postgresql[***DATABASE-VERSION***]-server  // if you omit
the db_version it defaults to 12
yum install -y postgresql[***DATABASE-VERSION***]-devel
```

Replace *[***DATABASE-VERSION***]* with the actual database version you are want to insta;;. For example, 16.

Procedure

1. SSH into the Hue host as a root user.
2. Download the psycopg2 package as follows:

```
wget https://files.pythonhosted.org/packages/d1/1e/b450599a27b1809bccbd4
e369f397cb18dc56b875778d961f9ae180b54b7/psycopg2-2.9.3.tar.gz
```

3. Extract the tarball as follows:

```
tar -xzvf psycopg2-2.9.3.tar.gz
```

4. Locate the pg_config executable file as follows:

```
find / -name pg_config
```

5. Add the directory containing the pg_config file to the $PATH variable:

```
export PATH="/usr/pgsql-[***DATABASE-VERSION***]/bin:$PATH"
```

6. Build and install the psycopg2 package as follows:

```
/usr/bin/python3.8 setup.py build_ext --pg-config=/usr/pgsql-[***DATABASE-VERSION***]/bin/pg_config install
```

Configuring and Starting the PostgreSQL Server

By default, PostgreSQL only accepts connections on the loopback interface. Configure PostgreSQL to accept the connections based on hostname, IP address (including CIDR address), or MAC address. A fully qualified domain name (FQDN) is not a requirement. If you do not make these changes, the services cannot connect to and use the database on which they depend.

Before you begin

If you are making changes to an existing database, make sure to stop any services that use the database before continuing.

Procedure

1. Initialize the PostgreSQL database cluster:

OS	Command
RHEL, SLES, and Ubuntu	`/usr/bin/postgresql-setup initdb`

2. Make sure that LC_ALL is set to en_US.UTF-8 and initialize the database as follows:

   - **RHEL**

   ```
   echo 'LC_ALL="en_US.UTF-8"' >> /etc/locale.conf
   ```

   - **SLES**

   ```
   sudo su -l postgres -c "initdb --pgdata=/var/lib/pgsql/data --encoding=UTF-8"
   ```

   - **Ubuntu**

   ```
   echo ‘LC_ALL=“en_US.UTF-8”’ >> /etc/default/locale.conf
   ```

3. Enable SCRAM-SHA-256 authentication. Edit pg_hba.conf, which is usually found in /var/lib/pgsql/data or /etc/postgresql/<*VERSION*>/main. Add the following line:

```
host all all 127.0.0.1/32 scram-sha-256
```

If the default pg_hba.conf file contains the following line:

```
host all all 127.0.0.1/32 ident
```

then the host line specifying scram-sha-256 authentication shown above must be inserted before this ident line. Failure to do so may cause an authentication error when running the scm_prepare_database.sh script. You can modify the contents of the scram-sha-256 line shown above to support different configurations. For example, if you want to access PostgreSQL from a different host, replace 127.0.0.1 with your IP address and update postgresql.conf, which is typically found in the same place as pg_hba.conf, to include:

```
listen_addresses = '*'
```

Attention: 27.0.0.1 is a loopback address. Making 127.0.0.1 the only IP for authentication means that the only host that is allowed to authenticate to the database is the one on which it is running. When you have multiple services that use the database, if they are located on other hosts, they could have problems connecting to the database and could fail. If the goal is to allow open authentication from all hosts, that should be 0.0.0.0/32, not 127.0.0.1. Otherwise, you should add lines for explicit IP addresses of hosts that need to authenticate.

4. Configure settings to ensure your system performs as expected. Update these settings in the /var/lib/pgsql/data/postgresql.conf or /var/lib/postgresql/data/postgresql.conf file. Settings vary based on cluster size and resources as follows:

- Small to mid-sized clusters - Consider the following settings as starting points. If resources are limited, consider reducing the buffer sizes and checkpoint segments further. Ongoing tuning may be required based on each host's resource utilization. For example, if the Cloudera Manager Server is running on the same host as other roles, the following values may be acceptable:
  - password_encryption = scram-sha-256
  - max_connection - In general, allow each database on a host 100 maximum connections and then add 50 extra connections. You may have to increase the system resources available to PostgreSQL, as described at Connection Settings.
  - shared_buffers - 256MB
  - wal_buffers - 8MB
  - checkpoint_segments - 16

    Note: The checkpoint_segments setting is removed in PostgreSQL 9.5 and higher, replaced by min_wal_size and max_wal_size. The PostgreSQL 9.5 release notes provides the following formula for determining the new settings:

    ```
    max_wal_size = (3 * checkpoint_segments) * 16MB
    ```
  - checkpoint_completion_target - 0.9

- Large clusters - Can contain up to 1000 hosts. Consider the following settings as starting points.
  - password_encryption = scram-sha-256
  - max_connection - For large clusters, each database is typically hosted on a different host. In general, allow each database on a host 100 maximum connections and then add 50 extra connections. You may have to increase the system resources available to PostgreSQL, as described at Connection Settings.
  - shared_buffers - 1024 MB. This requires that the operating system can allocate sufficient shared memory. See PostgreSQL information on Managing Kernel Resources for more information on setting kernel resources.
  - wal_buffers - 16 MB. This value is derived from the shared_buffers value. Setting wal_buffers to be approximately 3% of shared_buffers up to a maximum of approximately 16 MB is sufficient in most cases.
  - checkpoint_segments - 128. The PostgreSQL Tuning Guide recommends values between 32 and 256 for write-intensive systems, such as this one.

    Note: The checkpoint_segments setting is removed in PostgreSQL 9.5 and higher, replaced by min_wal_size and max_wal_size. The PostgreSQL 9.5 Release Notes provides the following formula for determining the new settings:

    ```
    max_wal_size = (3 * checkpoint_segments) * 16MB
    ```
  - checkpoint_completion_target - 0.9.

  Note: On PostgreSQL12 and above, Cloudera recommends to disable jit. You can set jit=off.

5. Configure the PostgreSQL server to start at boot.

OS	Command
RHEL, SLES, and Ubuntu	`sudo systemctl enable postgresql`

6. Restart the PostgreSQL database by running the following command:

OS	Command
RHEL, SLES, and Ubuntu	`sudo systemctl restart postgresql`

Important:

On RHEL 8.4 or higher, while restarting PostgreSQL if you are unable to proceed further with the process then perform the following steps:

a. Create a directory by running the following command:

```
mkdir -p /etc/systemd/system/postgresql.service.d
```

b. Create a configuration file by running the following command:

```
cat >
/etc/systemd/system/postgresql.service.d/stop.conf <<
EOF
[Service]
ExecStopPost=+/usr/bin/systemctl daemon-reload
EOF
```

c. Reload the systemd daemon by running the following command:

```
systemctl daemon-reload
```

Creating Databases for Cloudera Software

You must create databases and service accounts for components that require databases.

About this task

The following components require databases:

- Cloudera Manager Server
- Cloudera Management Service roles:
    - Reports Manager
- Hue
- Each Hive metastore
- Oozie
- Schema Registry
- Streams Messaging Manager

The databases must be configured to support the PostgreSQL UTF8 character set encoding.

Record the values you enter for database names, usernames, and passwords. The Cloudera Manager installation wizard requires this information to correctly connect to these databases.

Note: The instructions for Cloudera Manager Server, Cloudera Management Service roles,Reports Manager, Hue, Hive metastores, and Oozie are documented in this topic.

Additional configuration for Ranger is documented in the following two topics. Refer to those topics for detailed instructions on the Ranger database.

To create databases for Cloudera Manager Server, Cloudera Management Service roles, Reports Manager, Hue, Hive metastores, and Oozie, complete the following steps:

Procedure

1. Connect to PostgreSQL:

```
sudo -u postgres psql
```

2. Create databases for each service you are using from the below table:

```
CREATE ROLE <USER> LOGIN PASSWORD '<PASSWORD>';
```

```
CREATE DATABASE <DATABASE> OWNER <USER> ENCODING 'UTF8';
```

You can use any value you want for *<DATABASE>*, *<USER>*, and *<PASSWORD>*. The following examples are the default names provided in the Cloudera Manager configuration settings, but you are not required to use them:

Table 26: Databases for Cloudera Software

Service	Database	User
Cloudera Manager Server	scm	scm
Reports Manager	rman	rman
Ranger RHEL/CentOS/Ubuntu	ranger	rangeradmin
Ranger KMS RHEL/CentOS	rangerkms	rangerkms
Hue	hue	hue
Hive Metastore Server	hive	hive
Oozie	oozie	oozie

Service	Database	User
Schema Registry	schemaregistry	schemaregistry
Streams Messaging Manager	smm	smm

Record the databases, usernames, and passwords chosen because you will need them later.

3. For PostgreSQL 8.4 and higher, set standard_conforming_strings=off for the Hive Metastore and Oozie databases:

```
ALTER DATABASE <DATABASE> SET standard_conforming_strings=off;
```

What to do next

- If you plan to use Apache Ranger, see the following topic for instructions on creating and configuring the Ranger database and to install the JDBC driver for the database. See Configuring a PostgreSQL Database for Ranger or Ranger KMS on page 194.
- If you plan to use Schema Registry or Streams Messaging Manager, see the following topic for instructions on configuring the database: Configuring the Database for Streaming Components on page 201
- After you install and configure PostgreSQL databases for Cloudera software, continue to Set up and Configure the Cloudera Manager Database to configure a database for Cloudera Manager.

Install and Configure MySQL for Cloudera Software

You can install a MySQL database for use with Cloudera Manager and other components that require a database.

To use a MySQL database, follow these procedures. For information on compatible versions of the MySQL database, see Database Requirements on page 32.

Before you begin

Ensure that the MySQL DB is configured with the InnoDB engine by running the following command from the MySQL shell:

```
mysql> show table status;
```

Installing the MySQL Server

Note:

- If you already have a MySQL database set up, you can skip to the section Configuring and Starting the MySQL Server on page 161 to verify that your MySQL configurations meet the requirements for Cloudera Manager.
- For MySQL 5.6 and 5.7, you must install the MySQL-shared-compat or MySQL-shared package. This is required for the Cloudera Manager Agent package installation.
- It is important that the datadir directory, which, by default, is /var/lib/mysql, is on a partition that has sufficient free space.
- Cloudera Manager installation fails if GTID-based replication is enabled in MySQL.

1. Install the MySQL database:

OS	Command

RHEL	MySQL is no longer included with RHEL. You must download the repository from the MySQL site and install it directly. You can use the following commands to install MySQL. For more information, visit the MySQL website.
	`wget <URL TO MYSQL RPM>`
	`sudo rpm -ivh <FILENAME>.rpm`
	`sudo yum update`
	`sudo yum install mysql-server`
	`sudo systemctl start mysqld`

SLES	`sudo zypper install mysql libmysqlclient_r17`
Ubuntu	`sudo apt-get install mysql-server`

Configuring and Starting the MySQL Server

Note: If you are making changes to an existing database, make sure to stop any services that use the database before continuing.

1. Stop the MySQL server if it is running.

OS	Command
RHEL 7 Compatible, SLES, and Ubuntu	`sudo systemctl stop mysqld`

2. Move old InnoDB log files /var/lib/mysql/ib_logfile0 and /var/lib/mysql/ib_logfile1 out of /var/lib/mysql/ to a backup location.

3. Determine the location of the option file, my.cnf (/etc/my.cnf by default).

4. Update my.cnf so that it conforms to the following requirements:

- To prevent deadlocks, set the isolation level to READ-COMMITTED.
- Configure the InnoDB engine.

  Important: Cloudera Manager does not start if its tables are configured with the MyISAM engine. (Typically, tables revert to MyISAM if the InnoDB engine is misconfigured.)

- The default settings in the MySQL installations in most distributions use conservative buffer sizes and memory usage. Cloudera Management Service roles need high write throughput because they might insert many records in the database. Cloudera recommends that you set the innodb_flush_method property to O_DIRECT.
- Set the max_connections property according to the size of your cluster:
  - Fewer than 50 hosts - You can store more than one database (for example, both the Cloudera Manager Server and Reports Manager) on the same host. If you do this, you should:
    - Put each database on its own physical disk for best performance. You can do this by manually setting up symbolic links or running multiple database instances (each instance uses a different data directory path).
    - Allow 100 maximum connections for each database and then add 50 extra connections. For example, for two databases, set the maximum connections to 250. If you store four databases on one host (the databases for Cloudera Manager Server, Hue, Reports Manager, and Hive metastore), set the maximum connections to 450.
  - More than 50 hosts - Do not store more than one database on the same host. Use a separate host for each database/host pair. The hosts do not need to be reserved exclusively for databases, but each database should be on a separate host.
- If the cluster has more than 1000 hosts, set the max_allowed_packet property to 16M. Without this setting, the cluster may fail to start due to the following exception: com.mysql.jdbc.PacketTooBigException.
- Binary logging is not a requirement for Cloudera Manager installations. Binary logging provides benefits such as MySQL replication or point-in-time incremental recovery after database restore. Examples of this configuration follow. For more information, see The Binary Log.

Here is an option file with Cloudera recommended settings:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
transaction-isolation = READ-COMMITTED
# Disabling symbolic-links is recommended to prevent assorted security r
isks;
# to do so, uncomment this line:
symbolic-links = 0

key_buffer_size = 32M
max_allowed_packet = 16M
thread_stack = 256K
thread_cache_size = 64

# The following 3 parameters only apply to MySQL version 5.7 and lower:
query_cache_limit = 8M
query_cache_size = 64M
query_cache_type = 1

max_connections = 550
#expire_logs_days = 10
#max_binlog_size = 100M

#log_bin should be on a disk with enough free space.
#Replace '/var/lib/mysql/mysql_binary_log' with an appropriate path for
 your
#system and chown the specified folder to the mysql user.
```

```
log_bin=/var/lib/mysql/mysql_binary_log
#In later versions of MySQL, if you enable the binary log and do not set
#a server_id, MySQL will not start. The server_id must be unique within
#the replicating group.
server_id=1

binlog_format = mixed

read_buffer_size = 2M
read_rnd_buffer_size = 16M
sort_buffer_size = 8M
join_buffer_size = 8M

# InnoDB settings
innodb_file_per_table = 1
innodb_flush_log_at_trx_commit  = 2
innodb_log_buffer_size = 64M
innodb_buffer_pool_size = 4G
innodb_thread_concurrency = 8
innodb_flush_method = O_DIRECT
innodb_log_file_size = 512M

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

5. If you are using MySQL version 8, remove the following three parameters from the options file:

```
query_cache_limit = 8M
query_cache_size = 64M
query_cache_type = 1
```

6. If AppArmor is running on the host where MySQL is installed, you might need to configure AppArmor to allow MySQL to write to the binary.
7. Ensure the MySQL server starts at boot:

OS	Command
RHEL 7 Compatible, SLES, and Ubuntu	`sudo systemctl enable mysqld`

8. Start the MySQL server:

OS	Command
RHEL 7 Compatible, SLES, and Ubuntu	`sudo systemctl start mysqld`

9. Run /usr/bin/mysql_secure_installation to set the MySQL root password and other security-related settings. In a new installation, the root password is blank. Press the Enter key when you're prompted for the root password. For the rest of the prompts, enter the responses listed below in bold:

```
sudo /usr/bin/mysql_secure_installation
```

```
[...]
Enter current password for root (enter for none):
OK, successfully used password, moving on...
[...]
Set root password? [Y/n] Y
New password:
Re-enter new password:
Remove anonymous users? [Y/n] Y
```

```
[...]
Disallow root login remotely? [Y/n] N
[...]
Remove test database and access to it [Y/n] Y
[...]
Reload privilege tables now? [Y/n] Y
All done!
```

Installing the MySQL JDBC Driver

Install the JDBC driver on the Cloudera Manager Server host, as well as any other hosts running services that require database access.

Note: If you already have the JDBC driver installed on the hosts that need it, you can skip this section. However, MySQL 5.7 requires a 5.1 driver version 5.1.x.. You can also use version 5.1.x.x to connect to MySQL 8.x.

Important: If you are using TLS v1.2, you must use version 5.1.48.

Cloudera recommends that you consolidate all roles that require databases on a limited number of hosts, and install the driver on those hosts. Locating all such roles on the same hosts is recommended but not required. Make sure to install the JDBC driver on each host running roles that access the database.

Note: Cloudera recommends using only version 8.0 of the JDBC driver.

<table>
<tr><th>OS</th><th>Command</th></tr>
<tr><td>RHEL</td><td>
Important: Using the yum install command to install the MySQL driver package before installing a JDK installs OpenJDK, and then uses the Linux alternatives command to set the system JDK to be OpenJDK. If you intend to use an Oracle JDK, make sure that it is installed before installing the MySQL driver using yum install. If you want to use OpenJDK, you can install the driver using yum.
Alternatively, use the following procedure to manually install the driver.

Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html (in .tar.gz format). As of the time of writing, you can download version 8.0.22 using wget as follows:
<code>wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.22.tar.gz</code>
Extract the JDBC driver JAR file from the downloaded file. For example:
<code>tar zxvf mysql-connector-java-8.0.22.tar.gz</code>
Copy the JDBC driver, renamed, to /usr/share/java/. If the target directory does not yet exist, create it. For example:
<code>sudo mkdir -p /usr/share/java/</code>
<code>cd mysql-connector-java-8.0.22</code>
<code>sudo cp mysql-connector-java-8.0.22-bin.jar /usr/share/java/mysql-connector-java.jar</code>

</td></tr>
<tr><td>SLES</td><td><code>sudo zypper install mysql-connector-java</code></td></tr>
<tr><td>Ubuntu</td><td><code>sudo apt-get install libmysql-java</code></td></tr>
</table>

Installing the MySQL client

To use MySQL as a backend database for Hue, you must install the MySQL client and other required dependencies on all the Hue hosts based on your operating system.

For Cent OS

1. SSH into the Hue host as a root user.
2. Download the MySQL yum repository as follows:

```
curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el7-5.noa
rch.rpm
```

3. Install the package as follows:

```
rpm -ivh mysql80-community-release-el7-5.noarch.rpm
```

4. Install the required dependencies as follows:

```
yum install mysql-devel
yum install -y xmlsec1  xmlsec1-openssl
```

   For MySQL version 8.0.27, add the mysql-community-client-8.0.25 client package as follows:

```
yum install mysql-community-client-8.0.25
```

5. Add the path where you installed the MySQL client and packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

6. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

For RHEL

1. SSH into the Hue host as a root user.
2. Download the MySQL yum repository as follows:

   (RHEL 7)

```
curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el7-5.noa
rch.rpm
```

   (RHEL 8)

```
curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el8-8.noa
rch.rpm
```

   (RHEL 9)

```
curl -sSLO https://dev.mysql.com/get/mysql80-community-release-el9-4.noa
rch.rpm
```

3. Install the package as follows:

   (RHEL 7)

   ```
   rpm -ivh mysql80-community-release-el7-5.noarch.rpm
   ```

   (RHEL 8)

   ```
   rpm -ivh mysql80-community-release-el8-8.noarch.rpm
   ```

   (RHEL 9)

   ```
   rpm -ivh mysql80-community-release-el9-4.noarch.rpm
   ```
4. Install the required dependencies as follows:

   ```
   yum install mysql-devel
   yum install -y xmlsec1  xmlsec1-openssl
   ```
5. Add the path where you installed the MySQL client and packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```
6. Install the MySQL client as follows:

   (RHEL 8)

   ```
   pip3.8 install mysqlclient
   ```

   (RHEL 9)

   ```
   pip3.9 install mysqlclient
   ```

For SLES

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

   ```
   zypper install libmysqlclient-devel
   zypper install xmlsec1
   zypper install xmlsec1-devel
   zypper install xmlsec1-openssl-devel
   ```
3. Add the path where you installed the packages to the PATH environment variable as follows:

   ```
   export PATH=/usr/local/bin:$PATH
   ```
4. Install the MySQL client as follows:

   ```
   pip3.8 install mysqlclient
   ```

For Ubuntu

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

   ```
   apt-get install libmysqlclient-dev
   apt-get install -y xmlsec1
   apt-get install libxmlsec1-openssl
   ```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

Creating Databases for Cloudera Software

Services that require databases

Create databases and service accounts for components that require databases:

- Cloudera Manager Server
- Cloudera Management Service roles:
  - Reports Manager
- Hue
- Each Hive metastore
- Oozie
- Schema Registry
- Streams Messaging Manager

Steps

1. Log in as the root user, or another user with privileges to create database and grant privileges:

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

```
Enter password:
```

2. Create databases for each service deployed in the cluster using the following commands. You can use any value you want for the *<DATABASE>*, *<USER>*, and *<PASSWORD>* parameters. The Databases for Cloudera Software table, below lists the default names provided in the Cloudera Manager configuration settings, but you are not required to use them.

   Configure all databases to use the utf8 character set.

   Include the character set for each database when you run the CREATE DATABASE statements described below.

```
CREATE DATABASE <DATABASE> DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8
_general_ci;
```

```
Query OK, 1 row affected (0.00 sec)
```

   Create USER by following the steps in this topic: CREATE USER Statement.

```
GRANT ALL ON <DATABASE>.* TO '<USER>'@'%' IDENTIFIED BY '<password>';
```

```
Query OK, 0 rows affected (0.00 sec)
```

Table 27: Databases for Cloudera Software

Service	Database	User
Cloudera Manager Server	scm	scm

Service	Database	User
Reports Manager	rman	rman
Ranger RHEL/CentOS/Ubuntu	ranger	rangeradmin
Ranger KMS RHEL/CentOS	rangerkms	rangerkms
Hue	hue	hue
Hive Metastore Server	hive	hive
Oozie	oozie	oozie
Schema Registry	schemaregistry	schemaregistry
Streams Messaging Manager	smm	smm

3. Confirm that you have created all of the databases:

   ```
   SHOW DATABASES;
   ```

   You can also confirm the privilege grants for a given user by running:

   ```
   SHOW GRANTS FOR '<USER>'@'%';
   ```

4. Record the values you enter for database names, usernames, and passwords. The Cloudera Manager installation wizard requires this information to correctly connect to these databases.

Next Steps

- If you plan to use Apache Ranger, see the following topic for instructions on creating and configuring the Ranger database. See Configuring a Ranger or Ranger KMS Database: MySQL/MariaDB on page 191.
- If you plan to use Schema Registry or Streams Messaging Manager, see the following topic for instructions on configuring the database: Configuring the Database for Streaming Components on page 201.
- After you install and configure MySQL databases for Cloudera software, continue to Set up and Configure the Cloudera Manager Database to configure a database for Cloudera Manager.

Install and Configure MariaDB for Cloudera Software

You can install a MariaDB database for use with Cloudera Manager and other components that require a database.

To use a MariaDB database, follow these procedures. For information on compatible versions of the MariaDB database, see Database Requirements on page 32.

Installing MariaDB Server

Note:

- If you already have a MariaDB database set up, you can skip to the section Configuring and Starting the MariaDB Server on page 169 to verify that your MariaDB configurations meet the requirements for Cloudera Manager.
- It is important that the datadir directory (/var/lib/mysql by default), is on a partition that has sufficient free space. For more information, see Hardware Requirements on page 13.

1. Install MariaDB server:

OS	Command
RHEL compatible	`sudo yum install mariadb-server`
SLES	`sudo zypper install mariadb-server`
Ubuntu	`sudo apt-get install mariadb-server`

If these commands do not work, you might need to add a repository or use a different yum install command, particularly on RHEL 6 compatible operating systems. For more assistance, see the following topics on the MariaDB website:

- RHEL compatible: Installing MariaDB with yum
- SLES: MariaDB Package Repository Setup and Usage
- Ubuntu: Installing MariaDB .deb Files

Configuring and Starting the MariaDB Server

Note: If you are making changes to an existing database, make sure to stop any services that use the database before continuing.

1. Stop the MariaDB server if it is running:

OS	Command
RHEL 7 Compatible, SLES, and Ubuntu	`sudo systemctl stop mariadb`

2. If they exist, move old InnoDB log files /var/lib/mysql/ib_logfile0 and /var/lib/mysql/ib_logfile1 out of /var/lib/mysql/ to a backup location.
3. Determine the location of the option file, my.cnf (/etc/my.cnf by default).
4. Update my.cnf so that it conforms to the following requirements:

- To prevent deadlocks, set the isolation level to READ-COMMITTED.
- The default settings in the MariaDB installations in most distributions use conservative buffer sizes and memory usage. Cloudera Management Service roles need high write throughput because they might insert many records in the database. Cloudera recommends that you set the innodb_flush_method property to O_DIRECT.
- Set the max_connections property according to the size of your cluster:
  - Fewer than 50 hosts - You can store more than one database (for example, both the Cloudera Manager Server and Reports Manager) on the same host. If you do this, you should:
    - Put each database on its own physical disk for best performance. You can do this by manually setting up symbolic links or running multiple database instances (each instance uses a different data directory path).
    - Allow 100 maximum connections for each database and then add 50 extra connections. For example, for two databases, set the maximum connections to 250. If you store four databases on one host (the

databases for Cloudera Manager Server, Hue, Reports Manager, and Hive metastore), set the maximum connections to 450.

    - More than 50 hosts - Do not store more than one database on the same host. Use a separate host for each database/host pair. The hosts do not need to be reserved exclusively for databases, but each database should be on a separate host.
- If the cluster has more than 1000 hosts, set the max_allowed_packet property to 16M. Without this setting, the cluster may fail to start due to the following exception: com.mysql.jdbc.PacketTooBigException.
- Although binary logging is not a requirement for Cloudera Manager installations, it provides benefits such as MariaDB replication or point-in-time incremental recovery after a database restore. The provided example configuration enables the binary log. For more information, see The Binary Log.

Here is an option file with Cloudera recommended settings:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
transaction-isolation = READ-COMMITTED
# Disabling symbolic-links is recommended to prevent assorted security r
isks;
# to do so, uncomment this line:
symbolic-links = 0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

key_buffer = 16M
key_buffer_size = 32M
max_allowed_packet = 32M
thread_stack = 256K
thread_cache_size = 64
query_cache_limit = 8M
query_cache_size = 64M
query_cache_type = 1

max_connections = 550
#expire_logs_days = 10
#max_binlog_size = 100M
#log_bin should be on a disk with enough free space.
#Replace '/var/lib/mysql/mysql_binary_log' with an appropriate path for yo
ur
#system and chown the specified folder to the mysql user.
log_bin=/var/lib/mysql/mysql_binary_log

#In later versions of MariaDB, if you enable the binary log and do not set
#a server_id, MariaDB will not start. The server_id must be unique within
#the replicating group.
server_id=1

binlog_format = mixed

read_buffer_size = 2M
read_rnd_buffer_size = 16M
sort_buffer_size = 8M
join_buffer_size = 8M
# InnoDB settings
innodb_file_per_table = 1
innodb_flush_log_at_trx_commit  = 2
innodb_log_buffer_size = 64M
innodb_buffer_pool_size = 4G
innodb_thread_concurrency = 8
innodb_flush_method = O_DIRECT
```

```
innodb_log_file_size = 512M
[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid
#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
```

5. If AppArmor is running on the host where MariaDB is installed, you might need to configure AppArmor to allow MariaDB to write to the binary.
6. Ensure the MariaDB server starts at boot:

OS	Command
RHEL 7 Compatible, SLES, and Ubuntu	`sudo systemctl enable mariadb`

7. Start the MariaDB server:

OS	Command
RHEL 7 Compatible, SLES, and Ubuntu	`sudo systemctl start mariadb`

8. Run /usr/bin/mysql_secure_installation to set the MariaDB root password and other security-related settings. In a new installation, the root password is blank. Press the Enter key when you're prompted for the root password. For the rest of the prompts, enter the responses listed below in bold:

```
sudo /usr/bin/mysql_secure_installation
```

```
[...]
Enter current password for root (enter for none):
OK, successfully used password, moving on...
[...]
Set root password? [Y/n] Y
New password:
Re-enter new password:
[...]
Remove anonymous users? [Y/n] Y
[...]
Disallow root login remotely? [Y/n] N
[...]
Remove test database and access to it [Y/n] Y
[...]
Reload privilege tables now? [Y/n] Y
[...]
All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Installing the MySQL JDBC Driver for MariaDB

The MariaDB JDBC driver is not supported. Follow the steps in this section to install and use the MySQL JDBC driver instead.

Install the JDBC driver on the Cloudera Manager Server host, as well as any other hosts running services that require database access.

Cloudera recommends that you consolidate all roles that require databases on a limited number of hosts, and install the driver on those hosts. Locating all such roles on the same hosts is recommended but not required. Make sure to install the JDBC driver on each host running roles that access the database.

Note: Cloudera recommends using only version 8.0 of the JDBC driver.

<table>
<tr><th>OS</th><th>Command</th></tr>
<tr><td>RHEL</td><td>

Important: Using the yum install command to install the MySQL driver package before installing a JDK installs OpenJDK, and then uses the Linux alternatives command to set the system JDK to be OpenJDK. If you intend to use an Oracle JDK, make sure that it is installed before installing the MySQL driver using yum install. If you want to use OpenJDK, you can install the driver using yum.

Alternatively, use the following procedure to manually install the driver.

1. Download the MySQL JDBC driver from http://www.mysql.com/downloads/connector/j/5.1.html (in .tar.gz format). As of the time of writing, you can download version 8.0.22 using wget as follows:

```
wget https://dev.mysql.com/get/Downloads/Connector-J/mys
ql-connector-java-8.0.22.tar.gz
```

2. Extract the JDBC driver JAR file from the downloaded file. For example:

```
tar zxvf mysql-connector-java-8.0.22.tar.gz
```

3. Copy the JDBC driver, renamed, to /usr/share/java/. If the target directory does not yet exist, create it. For example:

```
sudo mkdir -p /usr/share/java/
cd mysql-connector-java-8.0.22
sudo cp mysql-connector-java-8.0.22-bin.jar /usr/share/
java/mysql-connector-java.jar
```

</td></tr>
<tr><td>SLES</td><td>

```
sudo zypper install mysql-connector-java
```

</td></tr>
<tr><td>Ubuntu</td><td>

```
sudo apt-get install libmysql-java
```

</td></tr>
</table>

Installing the MySQL client

To use MariaDB as a backend database for Hue, you must install the MySQL client and other required dependencies on all the Hue hosts based on your operating system.

For Cent OS

1. SSH into the Hue host as a root user.
2. Install the required dependencies as follows:

```
yum install -y xmlsec1  xmlsec1-openssl
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

For RHEL

1. SSH into the Hue host as a root user.
2. Install the required dependencies as follows:

```
yum install mysql-devel
yum install -y xmlsec1  xmlsec1-openssl
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

(RHEL 8)

```
pip3.8 install mysqlclient
```

(RHEL 9)

```
pip3.9 install mysqlclient
```

For SLES

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

```
zypper install libmysqlclient-devel
zypper install xmlsec1
zypper install xmlsec1-devel
zypper install xmlsec1-openssl-devel
```

Attention: While installing the mysql-devel and libmysqlclient-devel packages on SLES 15, use the “--replacefiles” zypper switch or manually enter yes on the interactive pop-up that you see when the files are being overwritten. Else, you may see an error such as: File /usr/bin/mariadb_config from install of MariaDB-devel-<version>.x86_64 conflicts with file from install of libmariadb-devel-3.1.21-150000.3.33.3.x86_64 (SLES Module Server Applications Updates).

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

For Ubuntu

1. SSH into the Hue host as a root user.
2. Install the required packages and dependencies as follows:

```
apt-get install libmysqlclient-dev
apt-get install -y xmlsec1
apt-get install libxmlsec1-openssl
```

3. Add the path where you installed the packages to the PATH environment variable as follows:

```
export PATH=/usr/local/bin:$PATH
```

4. Install the MySQL client as follows:

```
pip3.8 install mysqlclient
```

Creating Databases for Cloudera Software

Services that require databases

Create databases and service accounts for components that require databases:

- Cloudera Manager Server
- Cloudera Management Service roles:
  - Reports Manager
- Hue
- Each Hive metastore
- Oozie
- Schema Registry
- Streams Messaging Manager

Steps

1. Log in as the root user, or another user with privileges to create database and grant privileges:

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

```
Enter password:
```

2. Create databases for each service deployed in the cluster using the following commands. You can use any value you want for the *<DATABASE>*, *<USER>*, and *<PASSWORD>* parameters. The Databases for Cloudera Software table, below lists the default names provided in the Cloudera Manager configuration settings, but you are not required to use them.

   Configure all databases to use the utf8 character set.

   Include the character set for each database when you run the CREATE DATABASE statements described below.

```
CREATE DATABASE <DATABASE> DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8
_general_ci;
```

```
Query OK, 1 row affected (0.00 sec)
```

   Create USER by following the steps in this topic: CREATE USER Statement.

```
GRANT ALL ON <DATABASE>.* TO '<USER>'@'%' IDENTIFIED BY '<password>';
```

```
Query OK, 0 rows affected (0.00 sec)
```

Table 28: Databases for Cloudera Software

Service	Database	User
Cloudera Manager Server	scm	scm
Reports Manager	rman	rman
Ranger RHEL/CentOS/Ubuntu	ranger	rangeradmin
Ranger KMS RHEL/CentOS	rangerkms	rangerkms
Hue	hue	hue
Hive Metastore Server	hive	hive
Oozie	oozie	oozie
Schema Registry	schemaregistry	schemaregistry
Streams Messaging Manager	smm	smm

3. Confirm that you have created all of the databases:

```
SHOW DATABASES;
```

You can also confirm the privilege grants for a given user by running:

```
SHOW GRANTS FOR '<USER>'@'%';
```

4. Record the values you enter for database names, usernames, and passwords. The Cloudera Manager installation wizard requires this information to correctly connect to these databases.

Next Steps

- If you plan to use Apache Ranger, see the following topic for instructions on creating and configuring the Ranger database. See Configuring a Ranger or Ranger KMS Database: MySQL/MariaDB on page 191.
- If you plan to use Schema Registry or Streams Messaging Manager, see the following topic for instructions on configuring the database: Configuring the Database for Streaming Components on page 201
- After you install and configure MariaDB databases for Cloudera software, continue to Set up and Configure the Cloudera Manager Database to configure a database for Cloudera Manager.

Configure Oracle Database for Cloudera Software

You can configure an external Oracle database for use with Cloudera Manager and other components that require a database.

To use an Oracle database, follow these procedures. For information on compatible versions of the Oracle database, see Database Requirements on page 32.

Related Information

Configuring Oracle RAC for the Cloudera Manager database

Collecting Oracle Database Information

To configure Cloudera Manager to work with an Oracle database, get the following information from your Oracle DBA:

- Hostname - The DNS name or the IP address and the port of the host where the Oracle database is installed.
- Port - The port number on which Oracle server listens.
- SID - The name of the schema that will store Cloudera Manager information.
- Username - A username for each schema that is storing information. You could have four unique usernames for the four schema.
- Password - A password corresponding to each username.
- Connection string of the DB (usually in format of USERNAME:PASSWORD@//HOSTNAME:PORT/SID)

Ensure that your customer defines the table it will create (see below) using UTF-8.

Installing the Oracle JDBC Connector

You must install the JDBC connector on the Cloudera Manager Server host and any other hosts that use a database. The JDBC connector MUST be supplied by the Oracle DBA to ensure it matches the Oracle DB server release.

Cloudera recommends that you assign all roles that require a database on the same host and install the connector on that host. Locating all such roles on the same host is recommended but not required. If you install a role, such as Reports Manager, on one host and other roles on a separate host, you would install the JDBC connector on each host running roles that access the database.

Alternatively, you can perform the following steps to distribute the Oracle JDBC connector file to all the nodes on the same directory:

1. Obtain the Oracle JDBC Driver from the Oracle DBA. For example, Oracle RAC 19c is provided with both ojdbc8.jar and ojdbc10.jar, and you should receive the later release as it is more updated (ojdbc10.jar).

2. Copy the Oracle JDBC JAR file as root user to /usr/share/java/oracle-connector-java.jar on the relevant nodes. The Cloudera Manager databases and the Hive Mestastore database use this shared filename and location. For example:

```
sudo mkdir -p /usr/share/java
sudo cp /tmp/ojdbc10.jar /usr/share/java/oracle-connector-java.jar
sudo chmod 644 /usr/share/java/oracle-connector-java.jar
```

Creating Databases for Cloudera Software

Provide the Oracle DBA with the information for creating the Create schema and user accounts for components that require databases (depending on the type of services that you install on the Cloudera Base Cluster).

To ease the tracking, provide a prefix name for each schema you define based on the environment you build such as cdp_**tst**_db_xxx for test environment, cdp_**prd**_db_xxx for production etc. For more information about a prefix name, see the following table:

Table 29: Databases for Cloudera Private Cloud Base Cluster

Service	Database	Username
Cloudera Manager Server	cdp_xxx_db_scm	cdp_xxx_scm
Reports Manager	cdp_xxx_db_rman	cdp_xxx_rman
Hive Metastore	cdp_xxx_db_hive	cdp_xxx_hive
Ranger	cdp_xxx_db_ranger	cdp_xxx_rangeradmin
Ranger KMS	cdp_xxx_db_rangerkms	cdp_xxx_rangerkms
Hue	cdp_xxx_db_hue	cdp_xxx_hue
Schema Registry (only if used)	cdp_xxx_db_schemaregistry	cdp_xxx_schemaregistry
Streams Messaging Manager (only if used)	cdp_xxx_db_smm	cdp_xxx_smm

Important: For information about creating the relevant schema DBs, see Creating the relevant schema DBs.

Note that Yarn Queue Manager schema in Cloudera works only with either internal DB or external PostgreSQL, so you will need to configure it separately.

You can create the Oracle database, schema and users on the host where the Cloudera Manager Server will run, or on any other hosts in the cluster. For performance reasons, you should install each database on the host on which the service runs, as determined by the roles you assign during installation or upgrade. In larger deployments or in cases where database administrators are managing the databases the services use, you can separate databases from services, but use caution.

The databases must be configured to support UTF-8 character set encoding.

Record the values you enter for database names, usernames, and passwords. The Cloudera Manager installation wizard requires this information to correctly connect to these databases.

Note: If you are deploying Oracle RAC (for high availability), enter the database name using the following format:

```
(DESCRIPTION=(LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=5)(
TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS=(PROTOCOL=TCP)(H
OST=[***HOSTNAME***])(PORT=[***PORT***]))(CONNECT_DATA=(SERVICE_NAM
E=[***SERVICE_NAME***])))
```

Provide the Oracle DBA with a list of commands to run in order to create the above schema,users and passwords on the Oracle server. Make sure to ask the DBA to provide you back the password created for each user/schema.

Note that you can ask the DBA to set the quota to unlimited to each user as follows:

```
ALTER USER <USER> quota unlimited on <TABLESPACE>;
```

Creating the relevant schema DBs

The format of the user/schema table is built using “xxx” where xxx should be replaced by either dev/tst/prd/dr etc. (for different environments separation on the Oracle DB)

Also, the “password” that is marked bold should be updated with a unique password given by the Oracle DBA per the security regulation of the Oracle DB server. That password will be used for connecting to the schema and will be configured on the Cloudera Manager UI later.

Important:

Unless stated otherwise, each DB is given GRANTs based on the need. The only exception is RMAN which needs GRANT ALL.

Important:

In the following commands, you should replace the **XXX** with the string for the relevant environment, and make sure the password string is replaced with the unique password string generated by Oracle DBA during the creation of the user.

You can create the relevant schema DBs by running the following set of commands:
Cloudera Manager Server DB

```
CREATE DATABASE cdp_xxx_db_scm CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_scm'@'%' IDENTIFIED BY
 'cdp_xxx_scm_password';
GRANT SELECT_CATALOG_ROLE, CONNECT, RESOURCE ON cdp_xxx_db_scm.*
 TO cdp_xxx_scm'@'%';
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,CR
EATE SEQUENCE,CREATE TRIGGER,UNLIMITED TABLESPACE ON cdp_xxx_db_
scm.* TO cdp_xxx_scm'@'%';
ALTER USER cdp_xxx_scm DEFAULT TABLESPACE cdp_xxx_db_scm;
ALTER USER cdp_xxx_scm quota unlimited on cdp_xxx_db_scm;
```

Reports Manager DB

Important: Reports Manager (rman) needs GRANT ALL PRIVILEGES on its DB.

```
CREATE DATABASE cdp_xxx_db_rman CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_rman'@'%' IDENTIFIED BY
 'cdp_xxx_rman_password';
GRANT ALL PRIVILEGES ON cdp_xxx_db_rman.* TO 'cdp_xxx_rman'@'%';
```

Hive Metastore DB

```
CREATE DATABASE cdp_xxx_db_hive CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_hive'@'%' IDENTIFIED BY
 'cdp_xxx_hive_password';
GRANT SELECT_CATALOG_ROLE, CONNECT, RESOURCE ON cdp_xxx_db_hive.*
 TO cdp_xxx_hive'@'%';
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,C
REATE SEQUENCE,CREATE TRIGGER,UNLIMITED TABLESPACE ON cdp_xxx_db_
hive.* TO cdp_xxx_hive'@'%';
ALTER USER cdp_xxx_hive DEFAULT TABLESPACE cdp_xxx_db_hive;
```

```
ALTER USER cdp_xxx_hive quota unlimited on cdp_xxx_db_hive;
```

Ranger DB

```
CREATE DATABASE cdp_xxx_db_ranger CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_rangeradmin'@'%' IDENTIFIED BY
 'cdp_xxx_rangeradmin_password';
GRANT SELECT_CATALOG_ROLE, CONNECT, RESOURCE TO cdp_xxx_rangera
dmin'@'%';
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,CR
EATE SEQUENCE,CREATE PUBLIC SYNONYM,CREATE ANY SYNONYM,CREATE TR
IGGER,UNLIMITED TABLESPACE TO cdp_xxx_rangeradmin'@'%';
ALTER USER cdp_xxx_rangeradmin DEFAULT TABLESPACE cdp_xxx_db_rang
eradmin;
ALTER USER cdp_xxx_rangeradmin quota unlimited on cdp_xxx_db_rang
eradmin;
```

Important:

CREATE_PUBLIC_SYNONYM and CREATE_ANY_SYNONYM are optional and not mandatory.

Ranger KMS DB

```
CREATE DATABASE cdp_xxx_db_rangerkms CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_rangerkms'@'%' IDENTIFIED BY
 'cdp_xxx_rangerkms_password';
GRANT SELECT_CATALOG_ROLE, CONNECT, RESOURCE TO cdp_xxx_rangerk
ms'@'%';
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,
CREATE SEQUENCE,CREATE PUBLIC SYNONYM,CREATE ANY SYNONYM,CREATE
TRIGGER,UNLIMITED TABLESPACE TO cdp_xxx_rangerkms'@'%';
ALTER USER cdp_xxx_rangerkms DEFAULT TABLESPACE cdp_xxx_db_rang
erkms;
ALTER USER cdp_xxx_rangerkms quota unlimited on cdp_xxx_db_rang
erkms;
```

Important:

CREATE_PUBLIC_SYNONYM and CREATE_ANY_SYNONYM are optional and not mandatory.

Hue DB

```
CREATE DATABASE cdp_xxx_db_hue CHARACTER SET AL32UTF8;
CREATE USER 'cdp_xxx_hue'@'%' IDENTIFIED BY
 'cdp_xxx_hue_password';
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,CR
EATE SEQUENCE,CREATE TRIGGER,UNLIMITED TABLESPACE TO cdp_xxx_hue
'@'%';
ALTER USER cdp_xxx_hue DEFAULT TABLESPACE cdp_xxx_db_hue;
ALTER USER cdp_xxx_hue quota unlimited on cdp_xxx_db_hue;
```

STREAMS MESSAGING MANAGER (SMM) / SCHEMA REGISTRY

```
CREATE TABLESPACE cdp_xxx_db_schemaregistry ONLINE;
CREATE TABLESPACE cdp_xxx_db_smm ONLINE;
CREATE USER 'cdp_xxx_schemaregistry'@'%' IDENTIFIED BY
 'cdp_xxx_schemaregistry_password';
CREATE USER 'cdp_xxx_smm'@'%' IDENTIFIED BY
 'cdp_xxx_smm_password';
```

```
ALTER USER cdp_xxx_schemaregistry DEFAULT TABLESPACE cdp_xxx_db_
schemaregistry;
ALTER USER cdp_xxx_smm DEFAULT TABLESPACE cdp_xxx_db_smm;

GRANT CONNECT, CREATE PROCEDURE,CREATE TABLE,CREATE SEQUENCE,C
REATE INDEX,ALTER PROCEDURE,ALTER TABLE,ALTER SEQUENCE,ALTER IND
EX,DROP PROCEDURE,DROP TABLE,DROP SEQUENCE,DROP INDEX,UNLIMITED
TABLESPACE TO cdp_xxx_schemaregistry'@'%';

GRANT CONNECT, CREATE PROCEDURE,CREATE TABLE,CREATE SEQUENCE,CREA
TE INDEX,ALTER PROCEDURE,ALTER TABLE,ALTER SEQUENCE,ALTER INDEX,
DROP PROCEDURE,DROP TABLE,DROP SEQUENCE,DROP INDEX,UNLIMITED TAB
LESPACE TO cdp_xxx_smm'@'%';
```

Important: The new users for smm/schema registry need to have permissions to create or drop tables and other database structures because when the Schema Registry or SMM service starts, it automatically creates the database structure. If the database structure already exists, then the service alters it as necessary.

```
FLUSH PRIVILEGES;
```

For further information about Oracle privileges, see Authorization: Privileges, Roles, Profiles, and Resource Limitations.

Next Steps

- If you plan to use **Apache Ranger**, see the following topic for instructions on creating and configuring the Ranger database and to install the JDBC driver for the database. See Configuring a Ranger or Ranger KMS Database: Oracle on page 193.
- If you plan to use **Schema Registry** or **Streams Messaging Manager**, see the following topic for instructions on configuring the database: Configuring the Database for Streaming Components on page 201
- After you install and configure Oracle databases for Cloudera software, continue to Set up and Configure the Cloudera Manager Database to configure a database for Cloudera Manager.
- If you plan to use **Hue** in the cluster, see Configuring the Hue Server to Store Data in the Oracle database.

Enabling TLS 1.2 on Database Server

Perform the following steps for managing the configuration of the Database Server and setting up TLS 1.2 connections to ensure the secure and proper functioning of the database environment.

Enabling TLS 1.2 for MySQL Database Server

TLS 1.2 encrypts the connection between the MySQL server and the Cloudera Manager server. You must enable TLS 1.2 for the MySQL database before setting up Cloudera Manager and add the MySQL root Certificate Authorities (CA) to the Cloudera Manager truststore.

Procedure

1. SSH into the MySQL database host.
2. Start the MySQL server:

```
service mysqld start
```

3. Establish an encrypted connection with the client:

```
mysql -p --ssl-mode=required
```

4. Verify whether TLS 1.2 is enabled on MySQL by running the following command:

```
mysql> show global variables like '%ssl%';
```

If TLS 1.2 is enabled, you see the value of have_ssl equal to YES, as follows. Otherwise, you see the value of have_ssl equal to DISABLED:

```
+---------------+----------+
| Variable_name | Value    |
+---------------+----------+
| have_openssl  | YES      |
| have_ssl      | YES      |
| ...           | ...      |
```

If TLS 1.2 is enabled, then you can skip the following steps and go to Importing the MySQL root certificate.

5. Create a certificate authority by running the following commands:

```
mkdir /etc/my.cnf.d/ssl/
cd /etc/my.cnf.d/ssl/
openssl genrsa 2048 > ca-key.pem
```

6. Create a certificate for the server using the CA certificate generated earlier by running the following command:

```
openssl req -new -x509 -nodes -days 365000 -key ca-key.pem -out ca-cert.
pem
openssl req -newkey rsa:2048 -days 365 -nodes -keyout server-key.pem -out
 server-req.pem
openssl rsa -in server-key.pem -out server-key.pem
```

7. Create a certificate for the clients using the same CA certificate by running the following command:

```
openssl x509 -req -in server-req.pem -days 365 -CA ca-cert.pem -CAkey ca-
key.pem -set_serial 01 -out server-cert.pem
```

8. Add the following lines in the /etc/my.cnf.d/server.cnf file under the [mysqld] section:

```
ssl-ca=/etc/my.cnf.d/ssl/ca-cert.pem
ssl-cert=/etc/my.cnf.d/ssl/server-cert.pem
ssl-key=/etc/my.cnf.d/ssl/server-key.pem
bind-address=*
```

You can view the content of the server.cnf file by running the following command:

```
vim /etc/my.cnf.d/server.cnf
```

9. Restart the MySQL server:

```
service mysqld restart
```

10. Check the TLS 1.2 status by running the following commands:

```
mysql -p --ssl-mode=required
> SHOW VARIABLES LIKE '%ssl%';
> status
```

Sample output:

```
> SHOW VARIABLES LIKE '%ssl%';
+-------------------------------------+-----------------+
| Variable_name                       | Value           |
+-------------------------------------+-----------------+
```

```
| admin_ssl_ca                        |                  |
| admin_ssl_capath                    |                  |
| admin_ssl_cert                      |                  |
| admin_ssl_cipher                    |                  |
| admin_ssl_crl                       |                  |
| admin_ssl_crlpath                   |                  |
| admin_ssl_key                       |                  |
| have_openssl                        | YES              |
| have_ssl                            | YES              |
| mysqlx_ssl_ca                       |                  |
| mysqlx_ssl_capath                   |                  |
| mysqlx_ssl_cert                     |                  |
| mysqlx_ssl_cipher                   |                  |
| mysqlx_ssl_crl                      |                  |
| mysqlx_ssl_crlpath                  |                  |
| mysqlx_ssl_key                      |                  |
| performance_schema_show_processlist | OFF              |
| ssl_ca                              | ca.pem           |
| ssl_capath                          |                  |
| ssl_cert                            | server-cert.pem  |
| ssl_cipher                          |                  |
| ssl_crl                             |                  |
| ssl_crlpath                         |                  |
| ssl_fips_mode                       | OFF              |
| ssl_key                             | server-key.pem   |
+-------------------------------------+------------------+

> status
SSL:   Cipher in use is ECDHE-RSA-AES128-GCM-SHA256
```

Enabling TCPS for Oracle Database Server

You must enable TCPS for the Oracle database before setting up Cloudera Manager. Enabling TCPS establishes a secure channel between the client (Cloudera Manager) and the server (Oracle Database Server).

Procedure

1. SSH into the Oracle database server host.
2. Change to the "oracle" user as follows:

   ```
   sudo -su oracle
   ```
3. Append the location of ORACLE_HOME to the PATH environment variable by running the following commands:

   ```
   export ORACLE_HOME=/opt/oracle/product/19c/dbhome_1
   export PATH=${PATH}:${ORACLE_HOME}/bin
   ```
4. Create an auto-login wallet by running the following command:

   ```
   orapki wallet create -wallet /opt/oracle/product/19c/dbhome_1/wallet -au
   to_login
   ```

   An auto-login wallet uses SSL's single sign-on functionality. The users do not need to specify password each time they open the wallet.
5. Add a self-signed certificate to this wallet by running the following command:

   ```
   orapki wallet add -wallet /opt/oracle/product/19c/dbhome_1/wallet -dn "C
   N=server" -keysize 4096 -self_signed -validity 365
   ```

6. Export the certificate from the Oracle wallet by running the following command:

```
orapki wallet export -wallet /opt/oracle/product/19c/dbhome_1/wallet -dn
 "CN=server" -cert server_ca.cert
```

This exports a certificate with the subject's distinguished name (-dn) (CN=server) from a wallet to the file that is specified by -cert (server_ca.cert).

7. Add the following lines to the /opt/oracle/product/19c/dbhome_1/network/admin/listener.ora configuration file:

```
SSL_CLIENT_AUTHENTICATION = FALSE
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /opt/oracle/product/19c/dbhome_1/wallet)
    )
  )
Register a new address in LISTENER:
(ADDRESS = (PROTOCOL = TCPS)(HOST = [***HOST***])(PORT = 2484))
```

8. Add the following lines to the /opt/oracle/product/19c/dbhome_1/network/admin/sqlnet.ora profile configuration file:

```
SSL_CLIENT_AUTHENTICATION = FALSE
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /opt/oracle/product/19c/dbhome_1/wallet)
    )
  )
```

9. Add the following lines to the /opt/oracle/product/19c/dbhome_1/network/admin/tnsnames.ora configuration file:

```
ORCLPDB1_SSL =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCPS)(HOST = [***HOST***])(PORT = 2484))
      (CONNECT_DATA =
        (SERVER = DEDICATED)
        (SERVICE_NAME = ORCLPDB1)
      )
      (SECURITY =
        (MY_WALLET_DIRECTORY = /opt/oracle/product/19c/dbhome_1/wallet)
      )
    )
```

10. Restart the listener by running the following commands:

```
lsnrctl stop
lsnrctl start
```

11. Check the TCPS status by running the following command

```
sqlplus cm/cmverystr0ngP4ss@ORCLPDB1_SSL
SELECT sys_context('USERENV', 'NETWORK_PROTOCOL') as network_protocol F
ROM dual;
```

Sample output:

```
NETWORK_PROTOCOL
```

```
----------------------------------------------------------------------
---------
tcps
```

Enabling TLS 1.2 for MariaDB Database Server

TLS 1.2 encrypts the connection between the MariaDB server and the Cloudera Manager server. You must enable TLS 1.2 for the MariaDB database before setting up Cloudera Manager and add the MariaDB root Certificate Authorities (CA) to the Cloudera Manager truststore.

Procedure

1. SSH into the MariaDB database host.
2. Start the MariaDB server:

```
service mysqld start
```

3. Establish an encrypted connection with the client:

```
mysql -p --ssl=true
```

4. Verify whether TLS 1.2 is enabled on MariaDB by running the following command:

```
mysql> show global variables like '%ssl%';
```

If TLS 1.2 is enabled, you see the value of have_ssl equal to YES, as follows. Otherwise, you see the value of have_ssl equal to DISABLED:

```
+---------------+----------+
| Variable_name | Value    |
+---------------+----------+
| have_openssl  | YES      |
| have_ssl      | YES      |
| ...           | ...      |
```

If TLS 1.2 is enabled, then you can skip the following steps and go to Importing the MariaDB root certificate.

5. Create a certificate authority by running the following commands:

```
mkdir /etc/my.cnf.d/ssl/
cd /etc/my.cnf.d/ssl/
openssl genrsa 2048 > ca-key.pem
```

6. Create a certificate for the server using the CA certificate generated earlier by running the following command:

```
openssl req -new -x509 -nodes -days 365000 -key ca-key.pem -out ca-cert.
pem
openssl req -newkey rsa:2048 -days 365 -nodes -keyout server-key.pem -out
 server-req.pem
openssl rsa -in server-key.pem -out server-key.pem
```

7. Create a certificate for the clients using the same CA certificate by running the following command:

```
openssl x509 -req -in server-req.pem -days 365 -CA ca-cert.pem -CAkey ca-
key.pem -set_serial 01 -out server-cert.pem
```

8. Add the following lines in the /etc/my.cnf.d/server.cnf file under the [mysqld] section:

```
ssl-ca=/etc/my.cnf.d/ssl/ca-cert.pem
ssl-cert=/etc/my.cnf.d/ssl/server-cert.pem
ssl-key=/etc/my.cnf.d/ssl/server-key.pem
```

```
bind-address=*
```

You can view the content of the server.cnf file by running the following command:

```
vim /etc/my.cnf.d/server.cnf
```

9. Restart the MariaDB server:

```
service mysqld restart
```

10. Check the TLS 1.2 status by running the following commands:

```
mysql -p --ssl=true
> SHOW VARIABLES LIKE '%ssl%';
> status
```

Sample output:

```
> SHOW VARIABLES LIKE '%ssl%';
+---------------------+-----------------------------------+
| Variable_name       | Value                             |
+---------------------+-----------------------------------+
| have_openssl        | YES                               |
| have_ssl            | YES                               |
| ssl_ca              | /etc/my.cnf.d/ssl/ca-cert.pem     |
| ssl_capath          |                                   |
| ssl_cert            | /etc/my.cnf.d/ssl/server-cert.pem |
| ssl_cipher          |                                   |
| ssl_crl             |                                   |
| ssl_crlpath         |                                   |
| ssl_key             | /etc/my.cnf.d/ssl/server-key.pem  |
| version_ssl_library | OpenSSL 1.0.2k-fips  26 Jan 2017  |
+---------------------+-----------------------------------+

> status
SSL:   Cipher in use is DHE-RSA-AES256-GCM-SHA384
```

Enabling TLS 1.2 for PostgreSQL Database Server

TLS 1.2 encrypts the connection between the PostgreSQL server and the Cloudera Manager server. You must enable TLS 1.2 for the PostgreSQL database before setting up Cloudera Manager.

Before you begin

Important:

Depending on your PostgreSQL installation, the PostgreSQL version number and the exact paths might vary in your environment. Ensure to adjust the commands according to the PostgreSQL version number. For more information about supported PostgreSQL versions, see Cloudera Support Matrix

Procedure

1. SSH into the PostgreSQL database host.

2. Start the PostgreSQL server by running the following command:

```
systemctl start postgresql-14
```

3. Verify whether TLS 1.2 is enabled on PostgreSQL by running the following command:

```
SHOW ssl;
```

If TLS 1.2 is enabled, you see the value of ssl equal to on, as follows:

```
ssl
-----
 on
(1 row)
```

If TLS 1.2 is enabled, then you can skip the following steps and go to Importing the PostgreSQL root certificate.

4. Create a certificate authority by running the following commands:

```
cd /var/lib/pgsql/14/data
openssl genrsa -des3 -out server.key 1024
openssl rsa -in server.key -out server.key
chmod 400 server.key
```

5. Create a certificate for the server using the CA certificate generated earlier by running the following command:

```
openssl req -new -key server.key -days 3650 -out server.crt -x509 -subj '/
CN=hostname'
```

6. Change the ownership and permissions of the files by running the following commands:

```
chown postgres server.crt server.key
chmod 400 server.key server.crt
```

7. Go to /var/lib/pgsql/14/data and open the postgresql.conffile to update the following database configurations:

```
ssl = on
ssl_cert_file = '/var/lib/pgsql/14/data/server.crt'
ssl_key_file = '/var/lib/pgsql/14/data/server.key'
```

8. Restart the PostgreSQL server by running the following command:

```
systemctl restart postgresql-14.service
```

9. Check the TLS 1.2 status by running the following commands:

```
SELECT name, setting
FROM pg_settings
WHERE name LIKE '%ssl%';
```

Sample output:

```

+----------------------------------------+----------------------------+
| name                                   | setting                    |
+----------------------------------------+----------------------------+
| ssl                                    | on                         |
| ssl_ca_file                            | server.crt                 |
| ssl_cert_file                          | server.crt                 |
| ssl_ciphers                            | HIGH:MEDIUM:+3DES:!aNULL   |
| ssl_crl_dir                            |                            |
| ssl_crl_file                           |                            |
| ssl_dh_params_file                     |                            |
| ssl_ecdh_curve                         | prime256v1                 |
| ssl_key_file                           | server.key                 |
```

```
| ssl_library                            | OpenSSL                     |
| ssl_max_protocol_version               |                             |
| ssl_min_protocol_version               | TLSv1.2                     |
| ssl_passphrase_command                 |                             |
| ssl_passphrase_command_supports_reload | off                         |
| ssl_prefer_server_ciphers              | on                          |

+----------------------------------------+----------------------------+
(15 rows)
```

Important: To change the default setting for ssl_ciphers from HIGH:MEDIUM:+3DES:!aNULL to HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!kRSA:!PSK:AESGCM. Add the following line to the postgresql.conf file:

```
ssl_ciphers= 'HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!kRSA:!PSK:A
ESGCM'
```

Restart the PostgreSQL server to check the new ssl_ciphers is reflected in the sample output.

Enabling Kerberos (MIT and AD) authentication for MariaDB Database Server

Perform the following steps for enabling Kerberos authentication on MariaDB Database Server and to connect Cloudera Manager Server to Kerberos enabled MariaDB. These steps are applicable for TLS 1.2 and non-TLS 1.2 clusters.

Procedure

1. SSH into the MariaDB database host.
2. Run the following command to install the auth_gssapi.so plugin:

   ```
   sudo yum install MariaDB-gssapi-server
   ```

   Ensure the plugin is present in the following directory /usr/lib64/mysql/plugin.

3. On the MariaDB server, create Service Principal and Keytab by running the following commands:

   a. **For MIT Kerberos**

      Run the following command to create Service Principal:

      ```
      kadmin -p root/admin -q "addprinc -randkey mariadb/${HOST}"
      ```

      Note: For example, kadmin -p root/admin -q "addprinc -randkey mariadb/mariaa-1.mariaa.root.hwx.site".

      For AD Kerberos

      Run the following command on AD server to create Service Principal:

      ```
      dsadd user CN=mariadb,CN=Users,DC=qe-infra-ad,DC=cloudera,DC=com
       -pwd Test123 -samid mariadb -upn mariadb@QE-INFRA-AD.CLOUDERA.C
      OM
      ```

   b. **For MIT Kerberos**

      Run the following command to create Keytab:

      ```
      kadmin -p root/admin -q "ktadd -k /path/to/mariadb.keytab mariad
      b/${HOST}"
      ```

      For AD Kerberos

      Run the following command on AD server to create Keytab:

      ```
      ktpass.exe /princ mariadb@QE-INFRA-AD.CLOUDERA.COM /mapuser mari
      adb@QE-INFRA-AD.CLOUDERA.COM /pass Test123 /out mariadb.keytab /
      crypto all /ptype KRB5_NT_PRINCIPAL /mapop set
      ```

      Copy the Keytab file to the database host.

   c. **For AD Kerberos only**

      Run the following command to map service principal name to the principal:

      ```
      setspn -s mariadb/QE-INFRA-AD.CLOUDERA.COM mariadb
      ```

   d. Add the Service Principal and Keytab to the /etc/my.cnf configuration file:

      ```
      gssapi_keytab_path=/path/to/mariadb.keytab
      gssapi_principal_name=service_principal_name/host.domain.com@REALM
      ```

4. You must install the plugin on the Mariadb (database) side. You can do this either by following Step 4. a. or Step 4. b.:

   a. Run the following query on Mariadb database:

   ```
   INSTALL SONAME 'auth_gssapi';
   ```

   b. Add the following plugin to the /etc/my.cnf configuration file by running the following command:

   ```
   plugin_load_add = auth_gssapi
   ```

   Restart the MariaDB server by running the following command:

   ```
   sudo systemctl restart mariadb
   ```

   After performing Step 4. a. or Step 4. b., the plugin is visible in the plugin table. To verify the plugin, run the following query:

   ```
   SHOW PLUGINS;
   ```

5. Create user principals and do kinit on the Cloudera Manager Server by running the following commands:

   For MIT Kerberos

   a. Create user principal by running the following command:

   ```
   kadmin -p root/admin -q "addprinc cm_kerb_user"
   ```

   b. Do kinit by running the following command:

   ```
   kinit cm_kerb_user
   ```

   For AD Kerberos

   a. Run the following command to create user in Active Directory:

   ```
   dsadd user CN=cm_kerb_user,CN=Users,DC=qe-infra-ad,DC=cloude
   ra,DC=com -pwd Test123 -samid cm_kerb_user -upn cm_kerb_user
   @QE-INFRA-AD.CLOUDERA.COM
   ```

   b. Run the following command to create Keytab for the user in Active Directory:

   ```
   ktpass.exe /princ cm_kerb_user@QE-INFRA-AD.CLOUDERA.COM /map
   user cm_kerb_user@QE-INFRA-AD.CLOUDERA.COM /pass Test123 /ou
   t cm_kerb_user.keytab /crypto all /ptype KRB5_NT_PRINCIPAL /
   mapop set
   ```

   c. Run the following command to map service principal name to the principal in Active Directory:

   ```
   setspn -s cm_kerb_user/QE-INFRA-AD.CLOUDERA.COM cm_kerb_user
   ```

   d. Copy the Keytab file to the Cloudera Manager Server host.

   e. Do kinit by running the following command in the Cloudera Manager Server host:

   ```
   sudo /usr/bin/kinit -kt /cdep/keytabs/cm_kerb_user.keytab -l
    1d -r 8d cm_kerb_user
   ```

6. On MariaDB Database, you must create the user for using the Kerberos authentication by running the following commands:

```
CREATE USER 'cm_kerb_user'@'%' IDENTIFIED WITH gssapi;
```

```
GRANT ALL PRIVILEGES ON cm.* TO 'cm_kerb_user'@'%' IDENTIFIED VIA gssapi;
```

7. Run the following command to log in the user using Kerberos and verify whether the Kerberos configuration and user are working correctly:

```
 mysql --plugin-dir=/usr/lib64/mysql/plugin --user=cm_kerb_user --host=h
ostname
```

8. Connect Cloudera Manager Server to Kerberos enabled MariaDB by performing the following steps:

    a. By default mysql-connector-java.jar is available. You must have MariaDB Connector/J JDBC driver for building Java applications on top of MariaDB such as mariadb-java-client-3.1.4.jar.

       Add mariadb-java-client-3.1.4.jar file to /usr/share/java location.

    b. Add the following line to the /etc/default/cloudera-scm-server file.

```
export CMF_JDBC_DRIVER_JAR="${CMF_JDBC_DRIVER_JAR}:/usr/share/java/maria
db-java-client-3.3.3.jar"
```

    c. Create a /etc/jaas.conf file with the following content:

```
Krb5ConnectorContext {
            com.sun.security.auth.module.Krb5LoginModule required
            useKeyTab=true
            keyTab="/tmp/cm_kerb_user.keytab" (/location of keytab)
            principal="cm_kerb_user@ROOT.COMOPS.SITE"
            doNotPrompt=true;
        };
```

    d. **For TLS cluster**

       Update the JDBC URL in /etc/cloudera-scm-server/db.properties as follows:

       Important: Do the following modifications to the original JDBC URL:

       - Modify from user=cm to user=cm_kerb_user
       - Remove useSSL=true and add sslMode=trust

       The updated URL should look like this:

```
com.cloudera.cmf.orm.hibernate.connection.url=jdbc:mysql://local
host:3306/cm?user=cm_kerb_user&sslMode=trust&trustCertificateKey
StoreUrl=file:///cdep/mariadbssl/db_keystore.jks&trustCertificat
```

```
eKeyStoreType=jks&trustCertificateKeyStorePassword=verystrongpas
sword&enabledTLSProtocols=TLSv1.2
```

For non-TLS cluster

Update the JDBC URL in /etc/cloudera-scm-server/db.properties as follows:

Important: Do the following modifications to the original JDBC URL:

- Modify from user=cm to user=cm_kerb_user

The updated URL should look like this:

```
com.cloudera.cmf.orm.hibernate.connection.url=jdbc:mariadb://<<h
ost>>:3306/cm?user=cm_kerb_user
```

e. Add below properties to /etc/default/cloudera-scm-server:

```
export CMF_JAVA_OPTS="${CMF_JAVA_OPTS}
-Djava.security.krb5.kdc=krbmariadb-1.krbmariadb.root.hwx.site
-Djava.security.krb5.realm=ROOT.HWX.SITE -Djava.security.auth.login.co
nfig=/etc/jaas.conf"
```

f. Restart the Cloudera Manager Server by running the following command:

```
sudo systemctl restart cloudera-scm-server
```

Setting up the database for Reports Manager

Reports Manager uses a relational database for storing preferences such as the type of directories that are selected to be included in reports and data to be displayed on the reports themselves. You must set up this database manually before adding the Reports Manager role.

About this task

Cloudera recommends that you set up the database on the same host where you are planning to add the Reports Manager role.

Before adding the first cluster, ensure you set up the database for Reports Manager on a new Cloudera Manager deployment.

Perform the following steps for setting up the database for Reports Manager.

The exact parameters and SQL are dependent on the database environment.

MySQL or MariaDB

Set up the MySQL or MariaDB database by running the following command:

```
CREATE USER rman IDENTIFIED BY '***password***';
CREATE DATABASE rman;
GRANT ALL ON rman.* TO rman;
```

PostgreSQL

Set up the PostgreSQL database by running the following command:

```
CREATE DATABASE rman;
CREATE USER rman WITH PASSWORD '***password***';
GRANT ALL ON DATABASE rman TO rman;
```

Oracle

For creating a regular (not containerized) Oracle DB Schema for Reports Manager, see Creating the relevant schema DBs.

Oracle (container example)

Set up the Oracle database by running the following commands:

1. Log into the container database using SQL*Plus

```
sqlplus system/ as sysdba
```

2. Create a pluggable database:

```
CREATE PLUGGABLE DATABASE rmanpdb
   ADMIN USER rmanpdbadmin IDENTIFIED BY ***admin password***
   FILE_NAME_CONVERT = ('/opt/oracle/oradata/ORCLCDB/pdbseed',
                        '/opt/oracle/oradata/ORCLCDB/rmanpdb');
```

3. Create a user in the pluggable database:

```
ALTER SESSION SET CONTAINER = rmanpdb;
ALTER DATABASE OPEN;
CREATE USER rman IDENTIFIED BY ***user password***;
GRANT ALL PRIVILEGES TO rman;
```

Warning:

The new user password might be saved in the shell history when using different types of databases. This can happen if you paste or type the above commands directly into a database shell. When setting passwords in your database shells, ensure to disable the history. To achieve this, you can use batch mode in MySQL and MariaDB. You can enable this mode by using the -B command line option.

If you are using PostgreSQL, turn off Readline support by using the -n option. If you are using Oracle SQL* Plus, ensure to disable history by running the command SET HISTORY OFF and avoid using rlwrap or similar tools.

Configuring a database for Ranger or Ranger KMS

Additional steps to configure databases for Ranger or Ranger KMS.

After you have installed a database, use these steps to configure the database for Ranger or Ranger KMS. Ranger and Ranger KMS should use separate databases.

Configuring a Ranger or Ranger KMS Database: MySQL/MariaDB

Prior to upgrading your cluster to Cloudera Private Cloud Base you must configure the MySQL or MariaDB database instance for Ranger by creating a Ranger database and user. Before you begin the transition, review the support policies of database and admin policy support for transactions.

Before you begin

A supported version of MySQL or MariaDB must be running and available to be used by Ranger. See Database Requirements.

Important:

- Ranger and Ranger KMS should use separate databases.
- Ranger only supports the InnoDB engine for MySQL and MariaDB databases.

When using MySQL or MariaDB, the storage engine used for the Ranger admin policy store tables must support transactions. InnoDB supports transactions. A storage engine that does not support transactions is not suitable as a policy store.

Procedure

1. Log in to the host where you want to set up the MySQL database for Ranger.
2. Make sure you have the MYSQL connector for MYSQL version 5.7 or higher in the /usr/share/java/ directory with name mysql-connector-java.jar..

   Important: If you are using TLS v1.2, you must use version 5.1.48
3. Edit the following file: /etc/my.cnf and add the following line:

   ```
   log_bin_trust_function_creators = 1
   ```

   Warning: If you do not add this configuration, the upgrade will fail and reverting your deployment to a stable state will be difficult.
4. Restart the database:

   ```
   systemctl restart mysqld
   ```

   or:

   ```
   systemctl restart mariadb
   ```
5. Log in to mysql:

   ```
   mysql -u root
   ```
6. Run the following commands to create the Ranger database and user.

   Substitute the following in the command:

   - (optional) Replace rangeradmin with a username of your choice. Note this username, you will need to enter it later when running the Upgrade Cluster command.
   - (optional) Replace cloudera with a password of your choice. Note this password, you will need to enter it later when running the Upgrade Cluster command.
   - *<RANGER ADMIN ROLE HOSTNAME>* – the name of the host where the Ranger Admin role will run. Note this host, you will need to enter it later when running the Upgrade Cluster command.

   ```
   CREATE DATABASE ranger;
   CREATE USER 'rangeradmin'@'%' IDENTIFIED BY 'cloudera';
   CREATE USER 'rangeradmin'@'localhost' IDENTIFIED BY 'cloudera';
   CREATE USER 'rangeradmin'@'<RANGER ADMIN ROLE HOSTNAME>' IDENTIFIED BY
   'cloudera';
   GRANT ALL PRIVILEGES ON ranger.* TO 'rangeradmin'@'%';
   GRANT ALL PRIVILEGES ON ranger.* TO 'rangeradmin'@'localhost';
   GRANT ALL PRIVILEGES ON ranger.* TO 'rangeradmin'@'<RANGER ADMIN ROLE
    HOSTNAME>';
   FLUSH PRIVILEGES;
   ```
7. Use the exit; command to exit MySQL.
8. Test connecting to the database using the following command:

   ```
   mysql -u RANGERADMIN -pCLOUDERA
   ```
9. After testing the connection, use the exit; command to exit MySQL.
10. Continue with the cluster installation or upgrade to complete the transition.

Configuring a Ranger or Ranger KMS Database: Oracle

Prior to upgrading your cluster to CDP Private Cloud Base you must configure the Oracle database instance for Ranger by creating a Ranger database and user. Before you begin the transition, review the support policies of database and admin policy support for transactions.

Before you begin

A supported version of Oracle must be running and available to be used by Ranger.

Procedure

1. Log in to the host where the Oracle database is running and launch Oracle sqlplus:

```
sqlplus sys/root as sysdba
```

2. Provide the Ranger database and user instructions to the DBA to execute. Consult the username/schema name carefully in case of using multiple environments (test/dev etc.) on the same Oracle DB server. For information about creating Ranger database and user, see Creating the relevant schema DBs.

What to do next

Continue installing or upgrading your cluster.

Configuring a Ranger or Ranger KMS Database: Oracle using /ServiceName format

Prior to upgrading your cluster to Cloudera Private Cloud Base you must configure the Oracle database instance for Ranger by creating a Ranger database and user. Before you begin the transition, review the support policies of database and admin policy support for transactions.

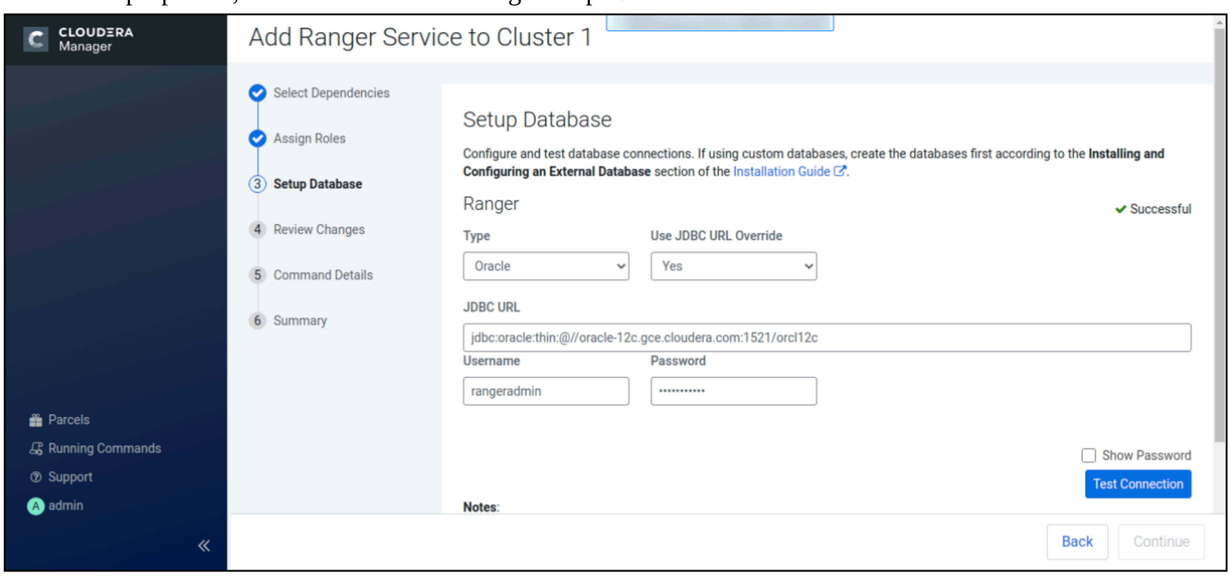
Before you begin

A supported version of Oracle must be running and available to be used by Ranger. See Database Requirements.

Important: Ranger and Ranger KMS should use separate databases.

Procedure

1. While installing Ranger service from Cloudera Manager using the installation wizard, in Setup Database, set the connection properties, as shown in the following example:



2. In Database Type, select Oracle.

3. In Use JDBC URL Override, select Yes.
4. In JDBC URL, type:

```
jdbc:oracle:thin:@//host:port/ServiceName
```

   This sets the JDBC URL to have the ServiceName connection url format.
5. In Username, type the user name required for connecting to the Oracle database Service Name you defined in the JDBC URL.
6. In Password, type the user name required for connecting to the Oracle database Service Name you defined in the JDBC URL.

   Note: Use similar steps to configure Ranger KMS service with Oracle database type using /ServiceName format. For Ranger KMS, use rangerkms rather than rangeradmin.

7. Click Test Connection.
8. After connection succeeds, click Continue.

What to do next

Continue installing or upgrading your cluster.

Configuring a PostgreSQL Database for Ranger or Ranger KMS

Complete the following steps to configure a PostgreSQL database instance for Ranger or Ranger KMS.

Configuring a PostgreSQL Database for Ranger or Ranger KMS on RHEL/Centos

Before you begin

Important: Ranger and Ranger KMS should use separate databases.

Note: For supported RHEL/Centos versions, see Cloudera Support Matrix.

Procedure

1. Run the following command to install PostgreSQL server:

```
sudo yum install postgresql-server
```

2. Initialize the Postgres database and start PostgreSQL:

```
sudo postgresql-setup initdb
sudo systemctl start postgresql
```

3. Optional: Configure PostgreSQL to start on boot:

```
sudo systemctl enable postgresql
```

4. Update the postgresql.conf file, which is usually found in /var/lib/pgsql/data or /var/lib/postgresql/data:

   - Uncomment and change #listen_addresses = 'localhost' to listen_addresses = '*'
   - Uncomment the #port = line and specify the port number (the default is 5432)
   - Optional: Uncomment and change #standard_conforming_strings= to standard_conforming_strings = off

5. Update the pg_hba.conf file, which is usually found in /var/lib/pgsql/data or /etc/postgresql/<version>/main:

    - Add the following line to allow connection to the Ranger database from any host:

    ```
    host    ranger          rangeradmin     0.0.0.0/0                 md5
    ```

    Note: For Ranger KMS, use rangerkms rather than rangeradmin.

6. Restart PostgreSQL:

```
sudo systemctl restart postgresql
```

7. The PostgreSQL database administrator should be used to create the Ranger databases. The following series of commands could be used to create the rangeradmin user and grant it adequate privileges. Be sure to replace 'password' with a strong password.

```
echo "CREATE DATABASE ranger;" | sudo -u postgres psql -U postgres
echo "CREATE USER rangeradmin WITH PASSWORD 'password';" | sudo -u postgr
es psql -U postgres
echo "GRANT ALL PRIVILEGES ON DATABASE ranger TO rangeradmin;" | sudo -u
 postgres psql -U postgres
```

Note: For Ranger KMS, use rangerkms rather than rangeradmin.

8. Install the PostgreSQL JDBC driver. If you would like to use the PostgreSQL JDBC driver version shipped with the OS repositories, run the following command:

```
yum install postgresql-jdbc*
```

You can also download the JDBC driver from the official PostgreSQL JDBC Driver website – https://jdbc.postgresql.org/download/.

9. Rename the Postgres JDBC driver .jar file to postgresql-connector-java.jarand copy it to the /usr/share/java directory. The following copy command can be used if the Postgres JDBC driver .jar file is installed from the OS repositories:

```
cp /usr/share/java/postgresql-jdbc.jar /usr/share/java/postgresql-connec
tor-java.jar
```

10. Confirm that the .jar file is in the Java share directory:

```
ls /usr/share/java/postgresql-connector-java.jar
```

11. Change the access mode of the .jar file to 644:

```
chmod 644 /usr/share/java/postgresql-connector-java.jar
```

What to do next

Ensure that the Ranger Solr and Ranger HDFS plugins are enabled. See Additional Steps for Apache Ranger on page 245 for details.

Configuring a PostgreSQL Database for Ranger on Ubuntu

Procedure

1. Run the following command to install PostgreSQL server:

```
apt-get install postgresql-server
```

2. Initialize the Postgres database and start PostgreSQL:

```
sudo postgresql-setup initdb
sudo systemctl start postgresql
```

3. Optional: Configure PostgreSQL to start on boot:

```
sudo systemctl enable postgresql
```

4. Edit the /var/lib/pgsql/data/postgresql.conf file:

   - Uncomment and change #listen_addresses = 'localhost' to listen_addresses = '*' *
   - Uncomment the #port = line and specify the port number (the default is 5432)
   - Optional: Uncomment and change #standard_conforming_strings= to standard_conforming_strings = off

5. Update the /var/lib/pgsql/data/pg_hba.conf file to allow connection to the Ranger database from any host:

   - Add the following line:

```
host    ranger          rangeradmin     0.0.0.0/0               md5
```

6. Restart PostgreSQL:

```
sudo systemctl restart postgresql
```

7. The PostgreSQL database administrator should be used to create the Ranger databases. The following series of commands could be used to create the rangeradmin user and grant it adequate privileges. Be sure to replace 'password' with a strong password.

```
echo "CREATE DATABASE ranger;" | sudo -u postgres psql -U postgres
echo "CREATE USER rangeradmin WITH PASSWORD 'password';" | sudo -u postgr
es psql -U postgres
echo "GRANT ALL PRIVILEGES ON DATABASE ranger TO rangeradmin;" | sudo -u
 postgres psql -U postgres
```

8. Install the PostgreSQL connector:

```
apt-get install postgresql-jdbc
```

9. Copy the connector .jar file to the Java share directory:

```
cp /usr/share/java/postgresql-jdbc.jar /usr/share/java/postgresql-connec
tor-java.jar
```

10. Confirm that the .jar file is in the Java share directory:

```
ls /usr/share/java/postgresql-connector-java.jar
```

11. Change the access mode of the .jar file to 644:

```
chmod 644 /usr/share/java/postgresql-connector-java.jar
```

What to do next

Ensure that the Ranger Solr and Ranger HDFS plugins are enabled. See the following topic, *Additional Steps for Ranger*, for details.

Configure Ranger with SSL/TLS enabled PostgreSQL Database

Steps to configure Ranger service with SSL/TLS enabled PostgreSQL database.

Before you begin

Make sure that:

- The database and database user for Ranger service are created in the required postgreSQL.
- A database server certificate is issued by a trusted certificate authority.
- The server host name matches the host name in the database server certificate.

  Note: From CDPDC-7.1.5 onwards, Ranger service requires postgres jdbc driver version >= 42.2.5. The Ranger code also constructs the JDBC connection string to have sslmode=verify-full, if Ranger Database SSL configurations are set in case of postgresql database type.
- Copy the database server certificate to /var/lib/ranger/ path , or use any custom path.

About this task

While installing Ranger service from Cloudera Manager using the installation wizard, stop at Setup Database to set the connection properties. The following steps apply to both FIPS-enabled and non FIPS-enabled clusters.

Procedure

1. In Setup Databases Type , select PostgreSQL.
2. In Use JDBC URL Override, select Yes.
3. In JDBC URL, type the following connection URL format:

```
jdbc:postgresql://<DB-HOST>:<DB-PORT>/<RANGER-DB>?sslmode=verify-
full&sslrootcert=
<path-to-database-server-certificate>
```

4. Click Test Connection.
5. Click Continue.
6. In Review Config, update the following configurations:

   ranger.db.ssl.enabled

   true

   ranger.db.ssl.verifyServerCertificate

   true

   ranger.db.ssl.auth.type

   1-way

   ranger.db.ssl.certificateFile

   <path-to-db-server-certificate>

Enable HA for a Ranger Postgres database

Ranger supports high availability enabled databases when deployed with Postgres.

About this task

To support high availability (HA) of a Postgres Ranger database, integrate HAProxy (a third-party load balancer product) into your environment. HAProxy will handle the load balancing. Replication Manager (repmgr) handles failover and recovery of the database.

Note: Currently, recovery of a previous master (as secondary) is not supported.

Before you begin

Ranger jdbc URL must point to a load balancer URL.

Example JDBC URL to connect to load balancer:

SSL:

```
jdbc:postgresql://<lb_hostname>:<lb_port>/ranger1?sslmode=verify-ca&sslrootc
ert=/cdep/pgssl/server.crt
```

Non-SSL:

```
jdbc:postgresql://<lb_hostname>:<lb_port>/ranger1
```

where:

Lb_hostname = Load balancer hostname

For example: mvnssl-sync-1.mvnssl-sync.root.hwx.site

Lb_port = Load balancer port

For example: 6432

The following steps describe how to setup HAProxy as a load balancer and `repmgr` as replication manager for postgres database:

Procedure

1. Install the HAProxy load balancer.

   Note: Install the load balancer only on the first node.

   a) On CentOS 7, use yum repository to install HAProxy.

   ```
   yum install haproxy
   ```

   The latest version of HAProxy available installs in yum repository.

   b) Run the following command to ensure that HAProxy runs every time the server reboots.

   ```
   chkconfig haproxy on
   ```

   c) Start the service, using the following command.

   ```
   systemctl start haproxy && systemctl status haproxy
   ```

2. Install the replication manager.

   Note: Perform this step on all the nodes of databases.

   You can use `repmgr` package for performing replication of databases. `repmgr` is available along with postgres repository so we can directly run the install command to install this package.

   ```
   sudo yum install repmgr_12
   ```

   Note: this command installs postgres 12. Use a repmgr version that supports the postgres version installed.

3. Configure the HA proxy:

a) Edit the HAProxy Configuration file.

```
vi /etc/haproxy/haproxy.cfg
```

b) Add the primary database server hostname and port number.

Example configuration file for haproxy.cfg:

```
frontend                        postgres-front
    mode tcp
    option http-server-close
    timeout client          3h
    timeout http-keep-alive 10s
    bind                        *:6432
    option httpchk
    default_backend             postgres-back
    option tcplog

backend postgres-back
    mode tcp
    option http-server-close
    retries                 5
    timeout connect         3s
    timeout server          3h
    timeout http-keep-alive 10s
    option httpchk
    option tcp-check
    option tcplog
    tcp-check connect
    balance                     source
server postgresprimary <hostname_of_primary_db>:5432 check
```

4. Configure the replication manager:

Postgres service must be up & running.

a) Update the following settings in postgresql.conf on the primary server.

```
max_wal_senders = 10

max_replication_slots = 10

wal_level = 'hot_standby' or 'replica' or 'logical'

hot_standby = on
archive_mode = on

archive_command = '/bin/true'

shared_preload_libraries = 'repmgr'
```

Note: In case of synchronous replication, synchronous_standby_names config param needs to be set in postgresql.conf

b) Create a dedicated user for repmgr.

```
create user repmgr;
create database repmgr with owner repmgr;
```

c) Ensure the repmgr user has appropriate permissions in pg_hba.conf and can connect in replication mode.

```
   local   replication   repmgr                          trust
    host    replication   repmgr      127.0.0.1/32            trust
```

```
    host    replication   repmgr      192.168.1.0/24          trust

    local   repmgr        repmgr                              trust
    host    repmgr        repmgr      127.0.0.1/32            trust
    host    repmgr        repmgr      192.168.1.0/24          trust
```

d) Create a repmgr.conf on the master server with the following entries:

```
cluster='failovertest'

node_id=1

node_name=node1

conninfo='host=node1 user=repmgr dbname=repmgr connect_timeout=2'

data_directory='/var/lib/pgsql/12/data/'
failover=automatic
promote_command='/usr/pgsql-12/bin/repmgr standby promote -f /var/lib/p
gsql/repmgr.conf --log-to-file'
follow_command='/usr/pgsql-12/bin/repmgr standby follow -f /var/lib/pgs
ql/repmgr.conf --log-to-file --upstream-node-id=%n'
```

Note: Users can also provide a script to automate the updating of HAProxy configurations in case of a failover. Provide this script in the above promote_command config.

e) Register the primary server with repmgr:

```
-bash-4.2$ /usr/pgsql-12/bin/repmgr -f /var/lib/pgsql/repmgr.conf primar
y register
```

f) Create the repmgr.conf file on standby server with following settings:

Note: In the commands for the host IP info, specify the IP of the standby server.

Note: Users can also provide a script to automate the updating of HAProxy configurations in case of a failover. Provide this script in the promote_command config.

```
-bash-4.2$ cat repmgr.conf

node_id=2

node_name=node2

conninfo='host=172.16.140.137 user=repmgr dbname=repmgr connect_timeout=
2'

data_directory='/var/lib/pgsql/12/data'
failover=automatic
promote_command='/usr/pgsql-12/bin/repmgr standby promote -f /var/lib/p
gsql/repmgr.conf --log-to-file'
follow_command='/usr/pgsql-12/bin/repmgr standby follow -f /var/lib/pgs
ql/repmgr.conf --log-to-file --upstream-node-id=%n'
```

g) Start cloning the database, using the following command:

```
-bash-4.2$ /usr/pgsql-12/bin/repmgr -h 172.16.140.135 -U repmgr -d repmg
r -f /var/lib/pgsql/repmgr.conf standby clone
```

h) Register the standby server with repmgr :

```
-bash-4.2$ /usr/pgsql-12/bin/repmgr -f /var/lib/pgsql/repmgr.conf standb
y register
```

i) Enable the automatic failover, by starting the repmgrd daemon process on master and slave:

```
-bash-4.2$ /usr/pgsql-12/bin/repmgrd -f /var/lib/pgsql/repmgr.conf
```

Related Information

HA Proxy Basics: Load Balance your servers

Prerequisites for setting up a basic replication cluster with repmgr

Configuring the Database for Streaming Components

Additional steps to configure the databases for Schema Registry and Streams Messaging Manager (SMM).

Configure PostgreSQL for Streaming Components

If you are installing Schema Registry or Streams Messaging Manager (SMM), you must configure the database to store metadata.

About this task

After you install PostgreSQL, configure the database to store:

- Schema Registry data such as the schemas and their metadata, all the versions and branches.
- SMM data such as Kafka metadata, stores metrics, and alert definitions.

Important: For the Schema Registry database, you must set collation to be case sensitive.

.

Procedure

1. Log in to Postgres:

```
sudo su postgres
psql
```

2. For the Schema Registry metadata store, create a database called registry with the password registry:

```
create database registry;
CREATE USER registry WITH PASSWORD 'registry';
GRANT ALL PRIVILEGES ON DATABASE "registry" to registry;
```

3. For the SMM metadata store, create a database called streamsmsgmgr with the password streamsmsgmgr:

```
create database streamsmsgmgr;
CREATE USER streamsmsgmgr WITH PASSWORD 'streamsmsgmgr';
GRANT ALL PRIVILEGES ON DATABASE "streamsmsgmgr" to streamsmsgmgr;
```

If you cannot grant all privileges, grant the following privileges that SMM and Schema Registry require at a minimum:

- CREATE/ALTER/DROP TABLE
- CREATE/ALTER/DROP INDEX
- CREATE/ALTER/DROP SEQUENCE

- CREATE/ALTER/DROP PROCEDURE

For example:

```
grant create session to streamsmsgmgr;
grant create table to streamsmsgmgr;
grant create sequence to streamsmsgmgr;
```

Configuring MySQL for Streaming Components

If you intend to use MySQL to store the metadata for Streams Messaging Manager or Schema Registry, you must configure the MySQL database.

About this task

Configure the database to store:

- In Schema Registry, the schemas and their metadata, all the versions and branches.
- In SMM, the Kafka metadata, stores metrics, and alert definitions.

Important: For the Schema Registry database, you must set collation to be case sensitive.

.

Procedure

1. Log in to the host.
   a) Run the following command for Schema Registry:

   ```
   ssh [MY_SCHEMA_REGISTRY_HOST]
   ```

   b) Run the following command for Streams Messaging Manager:

   ```
   ssh [MY_STREAMS_MESSAGING_MANAGER_HOST]
   ```

2. Launch the MySQL monitor:

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

3. Create the database for the Schema Registry and the SMM metastore:

   ```
   create database registry;
   create database streamsmsgmgr;
   ```

4. Create Schema Registry and SMM user accounts, replacing the final IDENTIFIED BY string with your password:

   ```
   CREATE USER 'registry'@'%' IDENTIFIED BY 'R12$%34qw';
   CREATE USER 'streamsmsgmgr'@'%' IDENTIFIED BY 'R12$%34qw';
   ```

5. Assign privileges to the user account:

   ```
   GRANT ALL PRIVILEGES ON registry.* TO 'registry'@'%' WITH GRANT OPTION ;
   ```

```
GRANT ALL PRIVILEGES ON streamsmsgmgr.* TO 'streamsmsgmgr'@'%' WITH GRANT
OPTION ;
```

If you cannot grant all privileges, grant the following privileges that SMM and Schema Registry require at a minimum:

- CREATE/ALTER/DROP TABLE
- CREATE/ALTER/DROP/REFERENCES TABLE
- CREATE/ALTER/DROP INDEX
- CREATE/ALTER/DROP SEQUENCE
- CREATE/ALTER/DROP PROCEDURE

For example:

```
grant create session to streamsmsgmgr;
grant create table to streamsmsgmgr;
grant create sequence to streamsmsgmgr;
```

6. Commit the operation:

```
commit;
```

Configuring Oracle for Streaming Components

If you intend to use Oracle to store the metadata for Streams Messaging Manager or Schema Registry, you must configure the Oracle database.

About this task

After you install Oracle, configure the database to store:

- Schema Registry data such as the schemas and their metadata, all the versions and branches.
- SMM data such as Kafka metadata, stores metrics, and alert definitions.

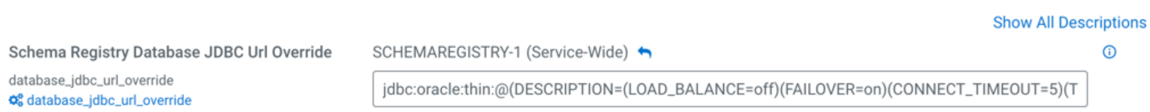
For setting the relevant tablespaces for the SMM and Schema Registry, see Creating the relevant schema DBs.

Results

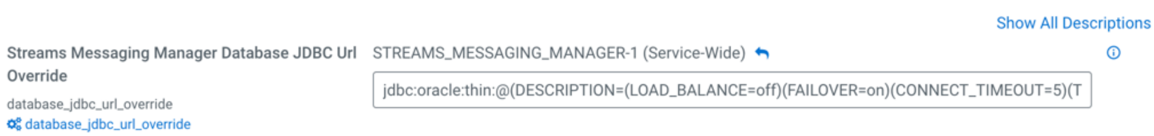
The new user needs to have permissions to create or drop tables and other database structures because when the Schema Registry or SMM service starts, it automatically creates the database structure. If the database structure already exists, then the service alters it as necessary.

Note: When using Oracle RAC, you need to enter the JDBC connection URL into the database_jdbc_url_override field in Cloudera Manager for both Schema Registry and SMM, as shown in the following images:

For Schema Registry,



For SMM,



For more information on URL for Oracle RAC, see https://docs.oracle.com/cd/E57185_01/EPMIS/apbs01s01.html.

Step 4: Deploy Cloudera Manager Server and Cloudera Manager Agents

Cloudera Manager packages are installed on the Cloudera Manager Server host (for the server packages) and on all the cluster nodes (the agent packages for the rest of the cluster).

Before you begin

Important: Cloudera deployment requires a local user with full sudo permissions without which you cannot install Cloudera Manager packages.

Procedure

1. On the Cloudera Manager Server host, type the following commands to install the Cloudera Manager packages:

OS	Command
RHEL	`sudo yum install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server`
SLES	`sudo zypper install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server`
Ubuntu	`sudo apt-get install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server`

2. On all the cluster nodes (apart from the server which you already installed in Step 1), run the following command to install the Cloudera Manager agent packages:

OS	Command
RHEL	`sudo yum install cloudera-manager-daemons cloudera-manager-agent`
SLES	`sudo zypper install cloudera-manager-daemons cloudera-manager-agent`
Ubuntu	`sudo apt-get install cloudera-manager-daemons cloudera-manager-agent`

Tip:

You can execute this command using either ansible-playbook by running yml script from an ansible server that has access with sudo permissions to all the cluster nodes.

Alternatively, you can install (using sudo) the EPEL clustershell package on a node that has access and permission to run sudo on all the cluster nodes (preferably the server node).

The EPEL clustershell package is available on RHEL/SLES/Ubuntu, run the following command to install the EPEL clustershell package:
RHEL

```
sudo yum install clustershell python3-clustershell
```

SLES

```
sudo zypper install clustershell python3-clustershell
```

Ubuntu

```
sudo apt-get install clustershell python3-clustershe
ll
```

Then, for example, if the customer grants you a local user called **CDP** with local sudo permission to root, then create your own customization file called /etc/clustershell/groups and change its ownership to **CDP** user (for easy updates in future). Then, add to it a line to represent all the cluster nodes. See the following example for a cluster with 10 nodes:

```
###
all: server[1-10].example.com
###
```

Note that in the same manner, you can set any alias for a subset of the cluster nodes (Nifi/Ozone etc.) and run a command only on that subset of nodes.

Now, you can run the following command from that node:

```
# clush -Bg all “sudo yum install cloudera-manager-daemons cloudera-
manager-agent -y”
```

This command will run on all the servers in “all” in parallel and will install the packages, and once done, the output of all the executions is shown on the CLI.

Note that the command includes the “-y” option as the yum command should be executed here without user interaction.

3. If you are installing on Ubuntu, and are planning to add the Kudu service to the cluster and are planning to enable Apache Ranger, run the following command on all cluster hosts:

```
sudo apt-get install gettext-base
```

Note: If you know in advance which hosts will be running the Kudu service roles, you only need to run this command on those hosts.

Step 5: Set up and configure the Cloudera Manager database

Cloudera Manager Server includes the scm_prepare_database.sh script that can create and configure a database.

The scm_prepare_database.sh script can perform the following activities:

- Create the Cloudera Manager Server database configuration file.
- (PostgreSQL) Create and configure a database for Cloudera Manager Server to use.
- (PostgreSQL) Create and configure a user account for Cloudera Manager Server.

The scm_prepare_database.sh script checks the connection between the Cloudera Manager Server and the database. Upon successful connection, the script writes the /etc/cloudera-scm-server/db.properties file. When you start Cloudera Manager for the first time, the scm_prepare_database.sh script creates and populates the necessary tables.

Although the script can create a database, the following procedures assume that you have already created the database as described in *Install and Configure Databases*. For more information about tuning the Cloudera Manager database for best performance, see the corresponding Knowledge article: hibernate.c3p0 Configs for Cloudera Manager.

The following sections describe the syntax for the script and demonstrate how to use it:

Configuring TLS 1.2 for Cloudera Manager

Cloudera recommends that you secure the network connection between the Cloudera Manager server and the backend database using TLS (Transport Layer Security) 1.2 encryption. Failure to do this will result in exposing vulnerabilities to various advanced cryptographic threats. Use the following topics to learn how to enable TLS 1.2 on Database Server and Cloudera Manager Server in the database environment.

Before you begin

The first step in enabling TLS 1.2 for Cloudera Manager is to configure the Database Server to accept TLS 1.2 connections. By configuring the Database Server to accept TLS 1.2 connections, it is ready to establish secure connections with the Cloudera Manager. To enable TLS 1.2 on Database Server, perform the steps from Enabling TLS 1.2 on Database Server.

Important: Cloudera Manager does not manage the backend database, so you must perform this step before you configure TLS 1.2 in the Cloudera Manager database connection.

Enabling TLS 1.2 on Cloudera Manager Server

To enable TLS 1.2 on Cloudera Manager, perform the following procedures on Cloudera Manager Server host.

Setting up a certificate in Cloudera Manager

To set up the certificate in Cloudera Manager, perform the following steps on the Cloudera Manager server host.

About this task

Copy the content of the exported certificate and add it to the Keystore on the Cloudera Manager server host. This step ensures that the certificate is available for secure communication.

Import the root certificate on the Cloudera Manager server host. This ensures that the Cloudera Manager server can trust the certificate authority that issued the certificate.

The following steps explain to import the root certificate of the different databases on the Cloudera Manager Server host.

Importing the MySQL root certificate

To import the MySQL database root certificate on the Cloudera Manager Server host, perform the following steps:

1. View the contents of the ssl-client.xml file by running the following commands:

```
export SSL_CLIENT=/etc/hadoop/conf/ssl-client.xml
echo $SSL_CLIENT
```

2. Obtain the truststore's location and password by running the following commands:

```
export TRUSTSTORE_LOCATION=$(xmllint --xpath "//configuration/property[n
ame='ssl.client.truststore.location']/value/text()" $SSL_CLIENT)
```

```
export TRUSTSTORE_PASSWORD=$(xmllint --xpath "//configuration/property[n
ame='ssl.client.truststore.password']/value/text()" $SSL_CLIENT)
```

3. Verify the contents of the truststore by running the following command:

```
/usr/java/default/bin/keytool -list -rfc -keystore $TRUSTSTORE_LOCATION -
storetype JKS -storepass $TRUSTSTORE_PASSWORD
```

4. Import the MySQL root certificate by running the following command:

```
/usr/java/default/bin/keytool -importcert -alias mysql -file /var/lib/my
sql/ca.pem -keystore $TRUSTSTORE_LOCATION -storetype jks -noprompt -stor
epass $TRUSTSTORE_PASSWORD
```

5. Verify the contents of the truststore again by running the following command:

```
/usr/java/default/bin/keytool -list -rfc -keystore $TRUSTSTORE_LOCATION -
storetype JKS -storepass $TRUSTSTORE_PASSWORD
```

Importing the MariaDB root certificate

To import the Maria DB database root certificate on the Cloudera Manager Server host, perform the following steps:

1. View the contents of the ssl-client.xml file by running the following commands:

```
export SSL_CLIENT=/etc/hadoop/conf/ssl-client.xml
echo $SSL_CLIENT
```

2. Obtain the truststore's location and password by running the following commands:

```
export TRUSTSTORE_LOCATION=$(xmllint --xpath "//configuration/property[n
ame='ssl.client.truststore.location']/value/text()" $SSL_CLIENT)
```

```
export TRUSTSTORE_PASSWORD=$(xmllint --xpath "//configuration/property[n
ame='ssl.client.truststore.password']/value/text()" $SSL_CLIENT)
```

3. Verify the contents of the truststore by running the following command:

```
/usr/java/default/bin/keytool -list -rfc -keystore $TRUSTSTORE_LOCATION -
storetype JKS -storepass $TRUSTSTORE_PASSWORD
```

4. Import the MariaDB root certificate by running the following command:

```
/usr/java/default/bin/keytool -importcert -alias mariadb -file /etc/my.c
nf.d/ssl/ca-cert.pem -keystore $TRUSTSTORE_LOCATION -storetype jks -nopr
ompt -storepass $TRUSTSTORE_PASSWORD
```

5. Verify the contents of the truststore again by running the following command:

```
/usr/java/default/bin/keytool -list -rfc -keystore $TRUSTSTORE_LOCATION -
storetype JKS -storepass $TRUSTSTORE_PASSWORD
```

Importing the Oracle root certificate

To import the Oracle database root certificate on the Cloudera Manager Server host, perform the following steps:

1. Copy the content of the certificate that you exported earlier and add it to the keystore on the base cluster instances.
2. Paste the copied content to the ca-cert.pem file.

   Important: You must create a separate truststore path for Oracle database certificates instead of using /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks.

3. Fetch the keystore password from the /etc/hadoop/conf/ssl-client.xml file by running the following command:

```
/usr/java/default/bin/keytool -importcert -alias oracle -file ca-cert.pe
m -keystore /var/lib/example/truststore.jks -storetype jks -noprompt -st
orepass [***PASSWORD***]
```

Importing the PostgreSQL root certificate

If the Database host and Cloudera Manager Server host are located on the same machine, then perform the following steps to import the PostgreSQL database root certificate:

1. Go to the path where root cerificates are stored. By default it is /var/lib/pgsql/14/data/.

```
cd /var/lib/pgsql/14/data/
```

2. Copy the PostgreSQL root certificate by running the following command:

```
cp server.crt root.crt
```

3. Create a new directory in the following path by running the following command:

```
mkdir -p /var/lib/cloudera-scm-server/.postgresql
```

4. Copy the root certificate to the new directory on the Cloudera Manager server host by running the following command:

```
cd /var/lib/cloudera-scm-server/.postgresql
cp /var/lib/pgsql/14/data/root.crt .
```

5. Change the ownership of the root certificate by running the following command:

```
chown cloudera-scm root.crt
```

   Then, include this root certificate path in the JDBC URL as follows:

```
jdbc:postgresql://<DB HOSTNAME>:<DB-PORT>/<DB NAME>?ssl=true&sslmode=ver
ify-ca&sslrootcert=<PATH_TO_ROOT_CERTIFICATE>
```

If the Database host and Cloudera Manager Server host are not located on the same machine, then perform the following steps to import the PostgreSQL database root certificate:

1. Perform the following steps from the Database host:

   a. Go to the path where root cerificates are stored. By default it is /var/lib/pgsql/14/data/.

   ```
   cd /var/lib/pgsql/14/data/
   ```

   b. Copy the PostgreSQL root certificate by running the following command:

   ```
   cp server.crt root.crt
   ```

2. Perform the following steps from the Cloudera Manager Server host:

   a. Create a new directory in the following path by running the following command:

   ```
   mkdir -p /var/lib/cloudera-scm-server/.postgresql
   ```

   b. Go to the new directory on the Cloudera Manager server host by running the following command:

   ```
   cd /var/lib/cloudera-scm-server/.postgresql
   ```

   c. Secure copy (scp) the root certificate from the Database host to the Cloudera Manager Server's PostgreSQL configuration directory by running the following command:

   ```
   scp root@<DB_HOST>:/var/lib/pgsql/14/data/root.crt .
   ```

   d. Change the ownership of the root certificate by running the following command:

   ```
   chown cloudera-scm root.crt
   ```

   Then, include this root certificate path in the JDBC URL as follows:

   ```
   jdbc:postgresql://<DB HOSTNAME>:<DB-PORT>/<DB NAME>?ssl=true&sslmode=ver
   ify-ca&sslrootcert=<PATH_TO_ROOT_CERTIFICATE>
   ```

Creating Cloudera Manager Server database configuration file

The scm_prepare_database.sh script is used to create the configuration file for the Cloudera Manager Server database. This file contains the necessary settings for the connection between Cloudera Manager and the database.

About this task

The scm_prepare_database.sh script checks the connection between the Cloudera Manager server and the database, and upon successful connection, it generates the /etc/cloudera-scm-server/db.properties file. This file includes the required configurations for the database connection.

Procedure

1. Create the Cloudera Manager Server database configuration file. See Syntax for scm_prepare_database.sh.
2. Upon successful connection, the scm_prepare_database.sh script writes the content of /etc/cloudera-scm-server/db.properties file as shown below:

   ```
   # Auto-generated by scm_prepare_database.sh on Tue Jun 13 07:22:02 UTC 2
   023
   #
   # For information describing how to configure the Cloudera Manager Server
   # to connect to databases, see the "Cloudera Manager Installation Guide."
   #
   com.cloudera.cmf.db.type=mysql
   com.cloudera.cmf.db.host=localhost
   com.cloudera.cmf.db.name=cm
   com.cloudera.cmf.db.user=cm
   com.cloudera.cmf.db.setupType=EXTERNAL
   com.cloudera.cmf.db.password=cm
   com.cloudera.cmf.orm.hibernate.connection.url=jdbc:mysql://<DB-HOST>:<DB-
   PORT>/<DB-Name>?sslMode=VERIFY_CA&trustCertificateKeyStoreUrl=file://<PATH
   ```

```
_TO_TRUSTSTORE_FILE>&trustCertificateKeyStoreType=jks&trustCertificateKe
yStorePassword=<TRUSTSTORE_PASSWORD>&enabledTLSProtocols=TLSv1.2
```

Important:

The format of the com.cloudera.cmf.orm.hibernate.connection.url property in the /etc/cloudera-scm-server/db.properties file can vary depending on the database type. For information about the specific JDBC URL format for each database type, see JDBC URL format.

What to do next

Important:

After you complete the TLS 1.2 setup for the database and Cloudera Manager Server, you must restart the Cloudera Manager Server. This process ensures that the necessary changes are in place and the Cloudera Manager server can establish secure connections with the configured database.

Restart the Cloudera Manager Server by running the following command:

```
sudo systemctl restart cloudera-scm-server
```

JDBC URL format

This topic provides the JDBC (Java Database Connectivity) URL format for each database type. The JDBC URL specifies how to connect to a database using the JDBC API. The JDBC URL is specific to the database system you are connecting to and it varies depending on the database vendor and the configuration of your database instance.

Table 30: JDBC URL format for specific database

Database type	JDBC URL
MySQL	jdbc:mysql://<DB-HOST>:<DB-PORT>/<DB-Name>?sslMode=VERIFY_CA&trustCertificateKeyStoreUrl=file://<PATH_TO_TRUSTSTORE_FILE>&trustCertificateKeyStoreType=jk
Oracle TCPS	jdbc:oracle:thin:@tcps://<DB-HOST>:<DB-PORT>:<SERVICE_NAM javax.net.ssl.trustStore=<PATH_TO_TRUSTSTORE_FILE>&javax.net
PostgreSQL	If you are using a truststore for the root certificates, then the format of th JDBC URL must be: jdbc:postgresql://<DB-HOST>:<DB-PORT>/<DB NAME>?sslMode=VERIFY_CA&trustCertificateKeyStoreUrl=file://<PATH_TO_TRUSTSTORE_FILE>&trustCertificateKeyStoreType=jk If you are not using a truststore for the root certificates, then the format c be: jdbc:postgresql://<DB-HOST>:<DB-PORT>/<DB-NAME>?ssl=true ca&sslrootcert=<PATH_TO_ROOT_CERTIFICATE>
MariaDB	jdbc:mysql://<DB-HOST>:<DB-PORT>/<CM-DB>?sslMode=VERIFY_CA&trustCertificateKeyStoreUrl=file://<PATH_TO_TRUSTSTORE_FILE>&trustCertificateKeyStoreType=jk

p

Enabling TLS 1.2 for Cloudera Manager High Availability

If you have configured Cloudera Manager for high availability with active and passive instances of the Cloudera Manager server, perform the following steps to enable TLS 1.2 for Cloudera Manager Server host.

Procedure

1. Set up a certificate in Cloudera Manager. Refer to the following page: Setting up a certificate in Cloudera Manager.

2. Create the configuration file for the Cloudera Manager Server database. Refer to the following page: Creating Cloudera Manager Server database configuration file.

   Important: Do not restart the Cloudera Manager Server if you have not completed step 1 and 2.

3. Restart the Cloudera Manager Server by running the following command:

```
sudo systemctl restart cloudera-scm-server
```

Configuring TLS 1.2 for Reports Manager

Cloudera recommends that you secure the network connection between the Reports Manager and the database using TLS (Transport Layer Security) 1.2 encryption. Use the following topics to learn how to enable TLS 1.2 on Reports Manager in the database environment.

Before you begin

Important:

Ensure you complete the following tasks before you start performing the steps to configure TLS 1.2 on the Reports Manager for communicating with the database:

1. The first step in enabling TLS 1.2 for Reports Manager is to set up the database. See Setting up the database for Reports Manager.
2. Add the Reports Manager role. See Adding the Reports Manager role.
3. Add a first CDP Private Cloud Base cluster and install the required services. See Adding a Cluster Using New Hosts.

About this task

Perform the following steps to configure TLS 1.2 on the Reports Manager for communicating with the database:

Procedure

1. On the Cloudera Manager UI, navigate to Clusters Cloudera Management Service .
2. Select the Configuration tab and search for reportsmanager_db_safety_valve.
3. Based on your database type you must override headlamp.db.properties file with JDBC URL properties. Enter the appropriate values in the following format to override the connection to use TLS 1.2.

   • MySQL

```
com.cloudera.headlamp.orm.hibernate.connection.url=jdbc:mysql://<DB-HOST
>:<DB-PORT>/<DB_NAME>?useSSL=true&trustCertificateKeyStoreUrl=<PATH_TO_T
RUSTSTORE_FILE>&trustCertificateKeyStoreType=<TRUSTSTORE_TYPE>&trustCert
ificateKeyStorePassword=<TRUSTSTORE_PASSWORD>&enabledTLSProtocols=TLSv1.
2
com.cloudera.headlamp.db.type=mysql
com.cloudera.headlamp.db.host=<DB-HOST>:<DB-PORT>
com.cloudera.headlamp.db.name=<DB_NAME>
```

   • MariaDB

```
com.cloudera.headlamp.orm.hibernate.connection.url=jdbc:mysql://<DB-HOST
>:<DB-PORT>/<DB_NAME>?useSSL=true&trustCertificateKeyStoreUrl=<PATH_TO_T
RUSTSTORE_FILE>&trustCertificateKeyStoreType=<TRUSTSTORE_TYPE>&trustCert
ificateKeyStorePassword=<TRUSTSTORE_PASSWORD>&enabledTLSProtocols=TLSv1.
2
com.cloudera.headlamp.db.type=mariadb
com.cloudera.headlamp.db.host=<DB-HOST>:<DB-PORT>
```

```
com.cloudera.headlamp.db.name=<DB_NAME>
```

- PostgreSQL

```
com.cloudera.headlamp.orm.hibernate.connection.url=jdbc:postgresql://<DB
-HOST>:<DB-PORT>/<DB_NAME>?useSSL=true&trustCertificateKeyStoreUrl=<PATH
_TO_TRUSTSTORE_FILE>&trustCertificateKeyStoreType=<TRUSTSTORE_TYPE>&trus
tCertificateKeyStorePassword=<TRUSTSTORE_PASSWORD>
com.cloudera.headlamp.db.type=postgresql
com.cloudera.headlamp.db.host=<DB-HOST>:<DB-PORT>
com.cloudera.headlamp.db.name=<DB_NAME>
```

- Oracle TCPS

```
com.cloudera.headlamp.orm.hibernate.connection.url=jdbc:oracle:thin:@tcp
s://<DB-HOST>:<DB-PORT>:<DB_NAME>?javax.net.ssl.trustStore=<PATH_TO_TRUS
TSTORE_FILE>&javax.net.ssl.trustStorePassword=<TRUSTSTORE_PASSWORD>&orac
le.net.ssl_server_dn_match=false
com.cloudera.headlamp.db.type=oracle
com.cloudera.headlamp.db.host=<DB-HOST>:<DB-PORT>
com.cloudera.headlamp.db.name=<DB_NAME>
```

Syntax for scm_prepare_database.sh

Review the syntax of the scm_prepare_database.sh script before you run it to configure the Cloudera Manager database.

The syntax for the scm_prepare_database.sh script is as follows:

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh <optional parameter
> <DATABASETYPE> <DATABASENAME> <DATABASEUSER> <PASSWORD>
```

Important: To create a new database, you must specify the -u and -p parameters for a user with privileges to create databases. For more information about optional parameter, see Table 32: Options on page 214. If you have already created the database as instructed in *Step 3: Install and Configure Databases*, do not specify these options.

Important: You can also run scm_prepare_database.sh without options to see the syntax.

The following tables describe the parameters and options for the scm_prepare_database.sh script:

Table 31: Parameters

Parameter (Required in bold)	Description
<DATABASETYPE>	One of the supported database types: • PostgreSQL: postgresql • MariaDB: mysql • MySQL: mysql • Oracle: oracle
<DATABASENAME>	The name of the Cloudera Manager Server database to use. For PostgreSQL databases, the script can create the specified database if you specify the -u and -p options with the credentials of a user that has privileges to create databases and grant privileges. The default database name provided in the Cloudera Manager configuration settings is scm, but you can also use any other database name such as cm_db or cmdb1.
<DATABASEUSER>	The username for the Cloudera Manager Server database to create or use. The default username provided in the Cloudera Manager configuration settings is scm_user, but you can also use any other database user such as cm_user or cm_db_user.

Parameter (Required in bold)	Description
<PASSWORD>	The password for the *<DATABASEUSER>* to create or use. If you do not want the password visible on the screen or stored in the command history, do not specify the password, and you are prompted to enter it as follows: `Enter SCM password:`

Table 32: Options

Option	Description
-s|--ssl	Optional parameter to enable SSL/TLS encryption. This parameter will enable SSL. Without this parameter in the command, you cannot enable the SSL encryption.
--jdbc-url	Optional parameter which allows users to provide a custom JDBC URL for connecting to the database. The provided values in the JDBC URL will override the default values for the database hostname, database type, and database name.
-?|--help	Display help.
--config-path	The path to the Cloudera Manager Server configuration files. The default is /etc/cloudera-scm-server.
-f|--force	If specified, the script does not stop if an error occurs.
-h|--host	The IP address or hostname of the host where the database is installed. The default is to use localhost.
-p|--password	The admin password for the database application. Use with the -u option. The default is no password. Do not put a space between -p and the password (for example, -phunter2). If you do not want the password visible on the screen or stored in the command history, use the -p option without specifying a password, and you are prompted to enter it as follows: `Enter database password:` If you have already created the database, do not use this option.
-P|--port	The port number to use to connect to the database. The default port is: • PostgreSQL: 5432 • MariaDB: 3306 • MySQL: 3306 • Oracle: 1521 This option is used for a remote connection only.
--scm-host	The hostname where the Cloudera Manager Server is installed. If the Cloudera Manager Server and the database are installed on the same host, do not use this option or the -h option.
--scm-password-script	A script to run whose stdout provides the password for user SCM (for the database).
-u|--user	The admin username for the database application. Use with the -p option. Do not put a space between -u and the username (for example, -uroot). If this option is supplied, the script creates a user and database for the Cloudera Manager Server. If you have already created the database, do not use this option.

The following examples demonstrate the syntax and output of the scm_prepare_database.sh script for different scenarios:

Example 1: Running the script to configure MySQL with TLS 1.2 enabled

This example demonstrates how to run the script on the MySQL or MariaDB host (db01.example.com):

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh -u<USER 1 TO CREATE THE
 DB> -p<PASSWORD> -h<DB HOSTNAME> --jdbc-url "jdbc:mysql://<DB HOSTNAME>:<DB
-PORT>/<DB-Name>?sslMode=VERIFY_CA&trustCertificateKeyStoreUrl=file://<PATH_
TO_TRUSTSTORE_FILE>&trustCertificateKeyStoreType=jks&trustCertificateKeyStor
```

```
ePassword=<TRUSTSTORE_PASSWORD>&enabledTLSProtocols=TLSv1.2" mysql <DB NAME>
 <USER 2 TO USE FROM CM> <PASSWORD> --ssl
```

```
JAVA_HOME=/usr/java/jdk1.8.0_232-cloudera
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing:  /usr/java/jdk1.8.0_232-cloudera/bin/java -cp /usr/share/java/m
ysql-connector-java.jar:/usr/share/java/oracle-connector-java.jar:/usr/share
/java/postgresql-connector-java.jar:/opt/cloudera/cm/schema/../lib/* com.clo
udera.enterprise.dbutil.DbCommandExecutor /etc/cloudera-scm-server/db.proper
ties com.cloudera.cmf.db.
[main] DbCommandExecutor INFO  A JDBC URL override was specified. Using this
 as the URL to connect to the database and overriding all other values.
[main] DbCommandExecutor INFO  Successfully connected to database.
All done, your SCM database is configured correctly!
```

Example 2: Running the script to configure PostgreSQL with TLS 1.2 enabled

This example demonstrates how to run the script on the PostgreSQL host (db01.example.com):

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh -u<USER 1 TO CREATE THE
 DB> -p<PASSWORD> -h<DB HOSTNAME> --jdbc-url "jdbc:postgresql://<DB HOSTNAME
>:<DB-PORT>/<DB NAME>?ssl=true&sslmode=verify-ca&sslrootcert=<PATH_TO_ROOT_C
ERTIFICATE>” postgresql <DB NAME> <USER 2 TO USE FROM CM> <PASSWORD> --ssl
```

```
JAVA_HOME=/usr/java/jdk1.8.0_232-cloudera
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing: /usr/java/jdk1.8.0_232-cloudera/bin/java -cp /usr/share/java/my
sql-connector-java.jar:/usr/share/java/oracle-connector-java.jar:/usr/share/
java/postgresql-connector-java.jar:/opt/cloudera/cm/schema/../lib/* com.clou
dera.enterprise.dbutil.DbCommandExecutor /etc/cloudera-scm-server/db.propert
ies com.cloudera.cmf.db.
[main] DbCommandExecutor INFO A JDBC URL override was specified. Using this
 as the URL to connect to the database and overriding all other values.
[main] DbCommandExecutor INFO Successfully connected to database.
All done, your SCM database is configured correctly!
```

Example 3: Running the script to configure Oracle with TLS 1.2 enabled

This example demonstrates how to run the script on the Oracle host (db01.example.com):

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh -u<USER 1 TO CREATE THE
 DB> -p<PASSWORD> -h<DB HOSTNAME> --jdbc-url "jdbc:oracle:thin:@tcps://<DB H
OSTNAME>:<DB-PORT>:<SERVICE_NAME>?javax.net.ssl.trustStore=<PATH_TO_TRUSTSTO
RE_FILE>&javax.net.ssl.trustStorePassword=<TRUSTSTORE_PASSWORD>&oracle.net.s
sl_server_dn_match=false" oracle <SERVICE_ NAME> <USER 2 TO USE FROM CM> <PA
SSWORD> --ssl
```

```
JAVA_HOME=/usr/java/jdk1.8.0_232-cloudera
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing:  /usr/java/jdk1.8.0_232-cloudera/bin/java -cp /usr/share/java/m
ysql-connector-java.jar:/usr/share/java/oracle-connector-java.jar:/usr/share
/java/postgresql-connector-java.jar:/opt/cloudera/cm/schema/../lib/* com.clo
udera.enterprise.dbutil.DbCommandExecutor /etc/cloudera-scm-server/db.proper
ties com.cloudera.cmf.db.
2023-07-14 07:37:58,575 [main] INFO  com.cloudera.enterprise.dbutil.DbComm
andExecutor  - A JDBC URL override was specified. Using this as the URL to c
onnect to the database and overriding all other values.
```

```
2023-07-14 07:37:59,416 [main] INFO  com.cloudera.enterprise.dbutil.DbComman
dExecutor  - Successfully connected to database.
All done, your SCM database is configured correctly!
```

Example 4: Running the script when MySQL or MariaDB is co-located with the Cloudera Manager Server

This example assumes that you have already created the Cloudera Management Server database and database user, naming both scm:

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh mysql scm scm
```

```
Enter SCM password:
JAVA_HOME=/usr/java/jdk1.8.0_141-cloudera
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing:  /usr/java/jdk1.8.0_141-cloudera/bin/java -cp /usr/share/java/m
ysql-connector-java.jar:/usr/share/java/oracle-connector-java.jar:/usr/share
/java/postgresql-connector-java.jar:/opt/cloudera/cm/schema/../lib/* com.clo
udera.enterprise.dbutil.DbCommandExecutor /etc/cloudera-scm-server/db.proper
ties com.cloudera.cmf.db.
[                          main] DbCommandExecutor              INFO  Succ
essfully connected to database.
All done, your SCM database is configured correctly!
```

Example 5: Running the script when MySQL or MariaDB is installed on another host

This example demonstrates how to run the script on the Cloudera Manager Server host (cm01.example.com) and connect to a remote MySQL or MariaDB host (db01.example.com):

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh mysql -h db01.example.c
om --scm-host cm01.example.com scm scm
```

```
Enter database password:
JAVA_HOME=/usr/java/jdk1.8.0_141-cloudera
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing:  /usr/java/jdk1.8.0_141-cloudera/bin/java -cp /usr/share/java/
mysql-connector-java.jar:/usr/share/java/oracle-connector-java.jar:/usr/shar
e/java/postgresql-connector-java.jar:/opt/cloudera/cm/schema/../lib/* com.cl
oudera.enterprise.dbutil.DbCommandExecutor /etc/cloudera-scm-server/db.prope
rties com.cloudera.cmf.db.
[                          main] DbCommandExecutor              INFO  Suc
cessfully connected to database.
All done, your SCM database is configured correctly!
```

Example 6: Running the script to configure Oracle

```
sudo /opt/cloudera/cm/schema/scm_prepare_database.sh -h cm-oracle.example.co
m oracle orcl sample_user sample_pass
```

```
JAVA_HOME=/usr/java/jdk1.8.0_141-cloudera
Verifying that we can write to /etc/cloudera-scm-server
Creating SCM configuration file in /etc/cloudera-scm-server
Executing:  /usr/java/jdk1.8.0_141-cloudera/bin/java -cp /usr/share/java/m
ysql-connector-java.jar:/usr/share/java/oracle-connector-java.jar:/usr/share
/java/postgresql-connector-java.jar:/opt/cloudera/cm/schema/../lib/*cloudera
.enterprise.dbutil.DbCommandExecutor /etc/cloudera-scm-server/db.properties
com.cloudera.cmf.db.
[ main] DbCommandExecutor INFO Successfully connected to database.
```

```
All done, your SCM database is configured correctly!
```

Note: If you have deployed Oracle RAC, there are additional steps, see the link below.

Related Information

Configuring Oracle RAC for the Cloudera Manager database

Configuring Oracle RAC for the Cloudera Manager database

After running the scm_prepare_database.sh script, there is additional configuration required.

Before you begin

You must run the scm_prepare_database.sh before continuing with the following steps.

Procedure

1. Log in to the Cloudera Manager server host using ssh.
2. Edit the /etc/cloudera-scm-server/db.properties file and add the following properties:

   Important:

   Under /etc/cloudera-scm-server/db.properties, you must set the following parameters such as Oracle username, Oracle password, Oracle hostname, Oracle host port number, etc. This is applicable for the scm service only.

   ```
   com.cloudera.cmf.orm.hibernate.connection.driver_class=oracle.jdbc.drive
   r.OracleDriver
   com.cloudera.cmf.orm.hibernate.connection.username=[***ORACLE USERNAME***]
   com.cloudera.cmf.orm.hibernate.connection.password=[***ORACLE PASSWORD***]
   com.cloudera.cmf.orm.hibernate.connection.url=jdbc:oracle:thin:@(DESCRI
   PTION=
   (LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=5)(TRANSPORT_CONNECT_TIME
   OUT=3)(RETRY_COUNT=3)(ADDRESS=(PROTOCOL=TCP)(HOST=[***ORACLE HOSTNAME***])
   (PORT=[***ORACLE HOST PORT NUMBER***]))(CONNECT_DATA=(SERVICE_NAME=[*** O
   racle DB Schema Name/SID ***])))
   ```
3. If you have configured Cloudera Manager for high availability with active and passive instances of the Cloudera Manager server, repeat the above steps for the other Cloudera Manager server instance.
4. (Optional) If you do not want to expose the password in the db.properties file, you can create a script that outputs the password of the database upon its execution. You can then add this script to the /etc/cloudera-scm-server/db.properties file.

   A sample script:

   ```
   cat > /[***PATH***]/retrieve_db_password.sh
   echo "[***MY_DATABASE_PASSWORD***]"
   ```

   Add the script to the db.properties file:

   ```
   com.cloudera.cmf.db.password_script=/[***PATH***]/retrieve_db_password.sh
   ```

What to do next

Continue with the CDP Private Cloud Base Production Installation steps.

Related Information

Syntax for scm_prepare_database.sh

Configure Oracle Database for Cloudera Software

PostgresSQL High Availability (HA)

Postgres HA support involves enabling Postgres HA and configuring Postgres HA behind a load balancer for the Cloudera Manager database.

Enabling Postgres HA

You must first enable the Postgres HA setup with load balancer configured as HA Proxy.

Cloudera Manager server application connects to the primary database node through the HA proxy. The HA proxy has exactly one backend server in configuration which is the cluster's primary database.

Configuring PostGres HA for the Cloudera Manager database

After running the scm_prepare_database.sh script, there is additional configuration required.

Before you begin

You must run the scm_prepare_database.sh before continuing with the following steps.

Procedure

1. Log in to the Cloudera Manager server host using ssh.
2. Edit the /etc/cloudera-scm-server/db.properties file and add the following properties:

```
com.cloudera.cmf.db.type=postgresql
com.cloudera.cmf.db.host=[***postgres host***]
com.cloudera.cmf.db.name=[***postgres name***]
com.cloudera.cmf.db.user=[***postgres user name***]
com.cloudera.cmf.db.setupType=EXTERNAL
com.cloudera.cmf.db.password=[***postgres password***]
com.cloudera.cmf.orm.hibernate.connection.url=jdbc
//<DB-HOST[***postgres HA host ***]>:<DB-PORT[***postgres HA port ***]>/<
DB-NAME>?sslMode=VERIFY_CA&trustCertificateKeyStoreUrl=file://<PATH_TO_T
RUSTSTORE_FILE>&trustCertificateKeyStoreType=jks&trustCertificateKeyStor
ePassword=<TRUSTSTORE_PASSWORD>
```

3. If you have configured Cloudera Manager for high availability with active and passive instances of the Cloudera Manager server, repeat the above steps for the other Cloudera Manager server instance.
4. (Optional) If you do not want to expose the password in the db.properties file, you can create a script that outputs the password of the database upon its execution. You can then add this script to the /etc/cloudera-scm-server/db.properties file.

   A sample script:

```
cat > /[***PATH***]/retrieve_db_password.sh
echo "[***MY_DATABASE_PASSWORD***]"
```

   Add the script to the db.properties file:

```
com.cloudera.cmf.db.password_script=/[***PATH***]/retrieve_db_password.sh
```

What to do next

Continue with the CDP Private Cloud Base Production Installation steps.

Step 6: Start the Cloudera Manager Server and Agents

After you set up the Cloudera Manager database, start Cloudera Manager Server and Agents, and log in to the Cloudera Manager Admin Console. Then proceed through the installation wizard.

Procedure

1. Ensure that the Cloudera Manager and Cloudera Agents are set to load automatically post-installation, to avoid cases where the server is rebooted and the service won't go up. Then start the services by performing the following steps:
   a) Run the following commands on the Cloudera Manager Server:

   ```
   sudo systemctl enable cloudera-scm-server
   ```

   ```
   sudo systemctl start cloudera-scm-server
   ```

   b) Run the following commands on each of the Cloudera Manager Agent (either one by one or using ansible-playbook/clush on all the nodes remotely):

   ```
   sudo systemctl enable cloudera-scm-agent
   ```

   ```
   sudo systemctl start cloudera-scm-agent
   ```

2. Wait several minutes for the Cloudera Manager Server to start. To observe the startup process, run the following on the Cloudera Manager Server host:

   ```
   sudo tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log
   ```

   When you see this log entry, the Cloudera Manager Admin Console is ready and you can stop the tail command:

   ```
   INFO WebServerImpl:com.cloudera.server.cmf.WebServerImpl: Started Jetty
   server.
   ```

   If the Cloudera Manager Server does not start, see Troubleshooting Installation Problems.
3. In a web browser, go to http://<*SERVER_HOST*>:7180, where <*SERVER_HOST*> is the FQDN or IP address of the host where the Cloudera Manager Server is running.
4. Log into Cloudera Manager Admin Console. The default credentials are:

   Username: admin

   Password: admin

   Note: Cloudera Manager does not support changing the admin username for the installed account. You can change the password using Cloudera Manager after you run the installation wizard. Although you cannot change the admin username, you can add a new user, assign administrative privileges to the new user, and then delete the default admin account.

Results

After logging in, the installation wizard launches. The following sections guide you through each step of the installation wizard.

Adding the Reports Manager role

After creating the database for the Reports Manager, you must add Reports Manager role on the Cloudera Manager.

About this task

Minimum Required Role: Limited Cluster Administrator (also provided by Full Administrator and Cluster Administrator)

Perform the following steps for adding the Report Manager role:

Procedure

1. Log in to the Cloudera Manager Admin Console.
2. On the Cloudera Manager UI, navigate to Cloudera Management Service Instances Add Role Instances .
3. Select a host for Reports Manager. Ensure that you select the host where the database was created while Setting up the database for Reports Manager.
4. Provide the details for the database that you have created while Setting up the database for Reports Manager.
   a) When you select Oracle as a database type, Cloudera Manager prompts for a username, password, and database SID.

      For a database created according to the example in Setting up the database for Reports Manager, the database SID is the name of the Pluggable database, and the user details are those of the user created by CREATE U SER (rman and ***user password*** in the example).
   b) For other database types, you need to specify a database name in addition to a username and password. These details are as defined in the SQL while Setting up the database for Reports Manager.
5. Start the Cloudera Management Service when the Reports Manager role is ready. See Starting the Cloudera Management Service.

Step 7: Using the installation wizard

Proceed through the installation wizard to accept licenses, install and configure Cloudera Runtime, and more.

Upload License File

On the Upload License File page, you can select either the trial version of CDP Private Cloud Base or upload a license file:

1. Select one of the following options:
   - Upload Cloudera Data Platform License
   - Try Cloudera Data Platform for 60 days. The CDP Private Cloud Base trial does not require a license file, but the trial expires after 60 days.

   If you have a license file for CDP Private Cloud Base, then perform the following steps to upload the license file:

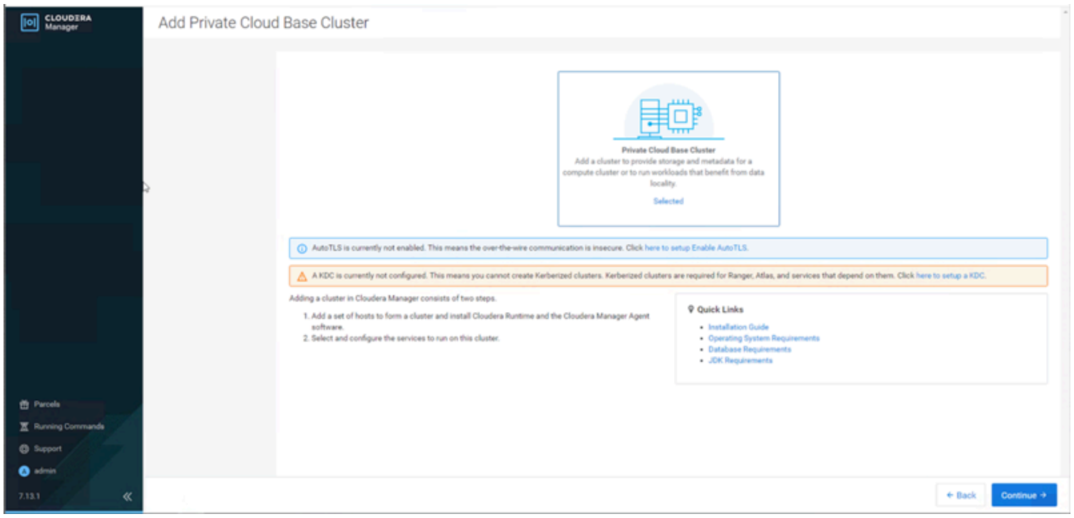
   a. Select Upload Cloudera Data Platform License.
   b. Click Upload License File.
   c. Browse to the location of the license file, select the file, and click Open.
   d. Click Upload.
   e. Click Continue.

   If you select the Try Cloudera Data Platform for 60 days option which is CDP Private Cloud Base Edition Trial, you can upload a license file at a later time. Read the license agreement and select Yes, I accept the Cloudera Standard License Terms and Conditions if you accept the terms and conditions of the license agreement. Then click Continue.
2. Click Continue to proceed with the installation.

   The Welcome page displays.

Add Private Cloud Base Cluster

The Add Private Cloud Base Cluster wizard provides a brief overview of the installation and configuration procedure, as well as some links to relevant documentation.



Setup Auto-TLS

The Setup Auto-TLS page provides instructions for initializing the certificate manager for auto-TLS as part of setting up a new cluster.

This setup assumes that Cloudera Manager is the Certificate Authority (CA) to create and manage all the certificates for the cluster (either self-signed certificates or when Cloudera Manager acts as part of the customer intermediate Certificate Authority (CA) chain).

In case you do not approve adding Cloudera Manager to your intermediate CA chain, it means you are responsible to generate and maintain all the cluster certificates (that is the Use Case 3). In such a case, setting up AutoTLS should not be used.

Setup a KDC

A Key Distribution Center(KDC) is required in order to create kerberized clusters. Kerberized clusters are required for Ranger, Atlas, and other services that depend on them.

The Setup KDC for this Cloudera Manager wizard walks you through the steps to configure Cloudera Manager for Kerberos Authentication. On completing the steps, the wizard displays a message indicating that KDC is setup and you can now create kerberized clusters.

Note: Setting KDC (Key Distribution Center) as part of Cloudera Manager means allowing Cloudera Manager to create and manage the Kerberos Principal creations for the cluster internally, without involving your external Kerberos server.

In case you want to manage the principals and want to create, remove, or update them on your Kerberos server, then Cloudera Manager can not be set as KDC, and the KDC wizard should not be used (hence the installation will continue without adding Kerberos at this point).

Please note that in such a case, you should not select any service which requires Kerberos (such as Ranger, RangerKMS, Atlas, Knox), and only once AutoTLS or Kerberos is set (post the Cloudera Manager Wizard deployment, separately), you can add these services to the cluster. If you deploy these services without Kerberos, the Ranger policies might not be created correctly and you need to remove them and regenerate them again.

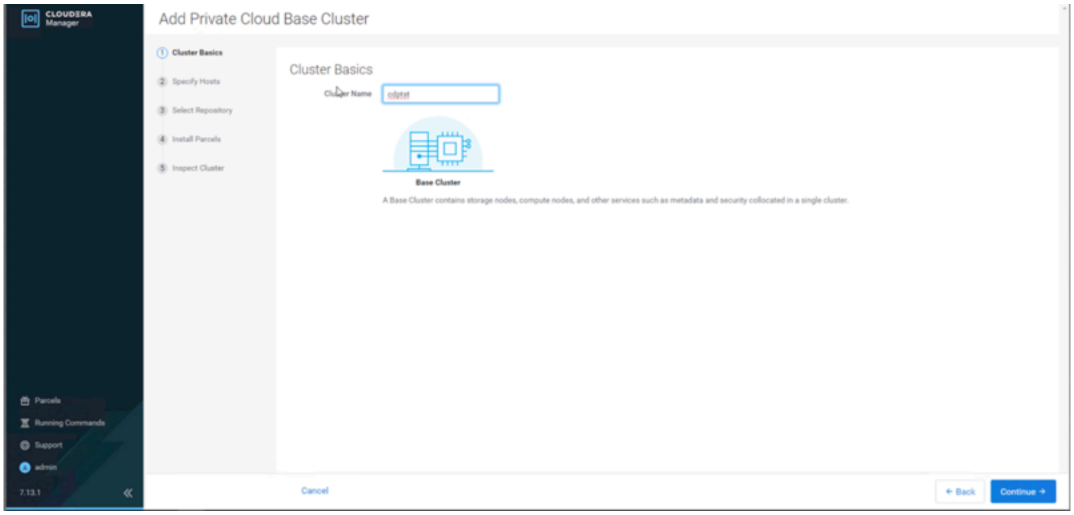
Press Continue to start the Wizard setup.

Cluster Basics

The Cluster Basics page allows you to specify the Cluster Name

For new installations, a Regular Cluster (also called a base cluster) is the only option. You can add a compute cluster after you finish installing the base cluster.

For more information on regular and compute clusters, and data contexts, see Virtual Private Clusters and Cloudera SDX.



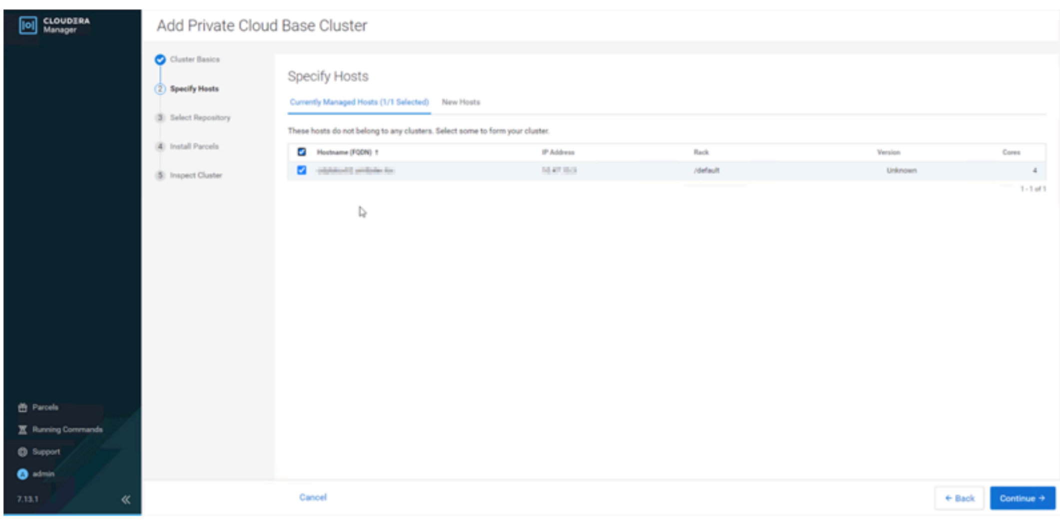
Enter a cluster name and click Continue.

Specify Hosts

Choose which hosts will run Runtime and other managed services.

Note: If you have enabled Auto-TLS, you must include the Cloudera Manager server host when you specify hosts.

1. To enable Cloudera Manager to automatically discover hosts on which to install Runtime and managed services, enter the cluster hostnames or IP addresses in the Hostnames field on the New Hosts tab.



You can specify hostname and IP address ranges as follows:

Expansion Range	Matching Hosts
10.1.1.[1-4]	10.1.1.1, 10.1.1.2, 10.1.1.3, 10.1.1.4
host[1-3].example.com	host1.example.com, host2.example.com, host3.example.com

Expansion Range	Matching Hosts
host[07-10].example.com	host07.example.com, host08.example.com, host09.example.com, host10.example.com

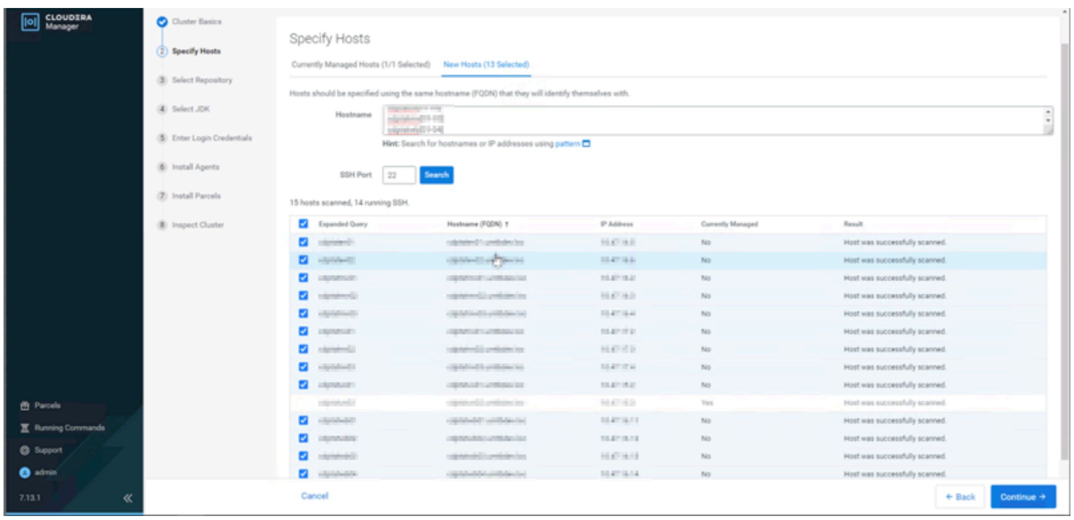
Important: Unqualified hostnames (short names) must be unique in a Cloudera Manager instance. For example, you cannot have both *HOST01.EXAMPLE.COM* and *HOST01.STANDBY.EXAMPLE.COM* managed by the same Cloudera Manager Server.

You can specify multiple addresses and address ranges by separating them with commas, semicolons, tabs, or blank spaces, or by placing them on separate lines. Use this technique to make more specific searches instead of searching overly wide ranges. Only scans that reach hosts running SSH will be selected for inclusion in your cluster by default. You can enter an address range that spans over unused addresses and then clear the nonexistent hosts later in the procedure, but wider ranges require more time to scan.

2. Click Search. If there are a large number of hosts on your cluster, wait a few moments to allow them to be discovered and shown in the wizard. If the search is taking too long, you can stop the scan by clicking Abort Scan. You can modify the search pattern and repeat the search as many times as you need until you see all of the expected hosts.

   Note: Cloudera Manager scans hosts by checking for network connectivity. If there are some hosts where you want to install services that are not shown in the list, make sure you have network connectivity between the Cloudera Manager Server host and those hosts, and that firewalls and SE Linux are not blocking access.

3. Verify that the number of hosts shown matches the number of hosts where you want to install services. Clear host entries that do not exist or where you do not want to install services.



4. Click Continue.

   The Select Repository screen displays.

Select Repository

Important: You cannot install software using both parcels and packages in the same cluster.

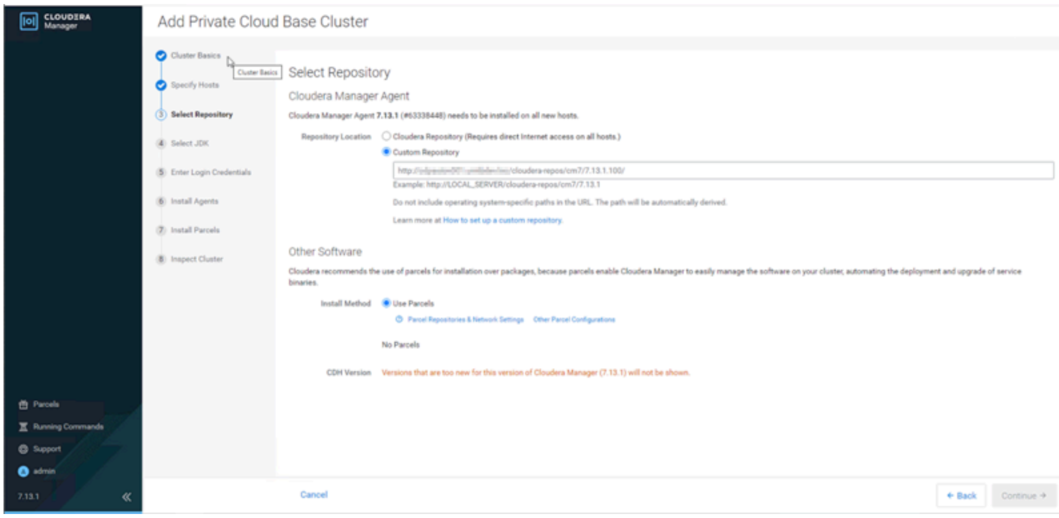
The Select Repository page allows you to specify repositories for Cloudera Manager Agent and Other software.

In the Cloudera Manager Agent section:

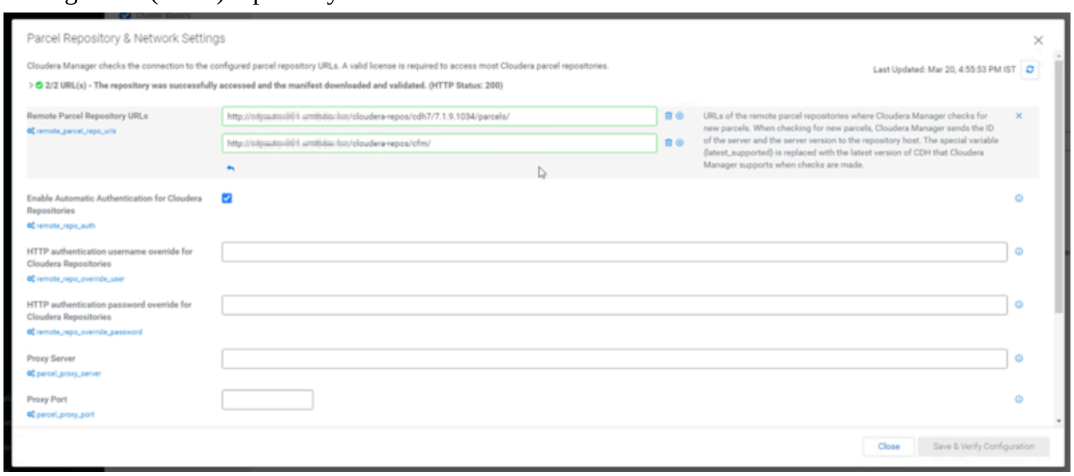
1. Select either Cloudera Repository (in case all cluster nodes have direct access to the Internet) or Custom Repository for the Cloudera Manager Agent software.
2. If you select Custom Repository, ensure the path points to the local repository you have previously prepared as seen in the following example. For instructions on setting up a custom repository, see *Configuring a Local Package Repository*.

In the Other software section,

1. Under Install Method Use Parcels , choose Parcel Repositories & Network Settings to validate/setup the local repository setting.

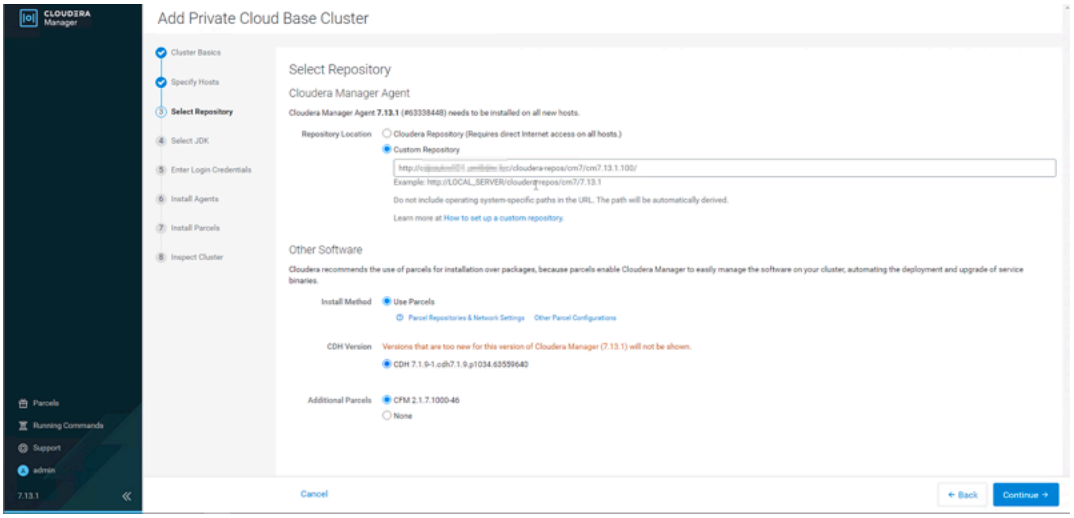


2. On the page that will open, ensure to validate the local repository location for the CDH parcels you will install and for any other parcel you want to bring to the cluster (the following example also shows Cloudera Flow Management (=Nifi) repository.



3. Once you have updated the URL, click Save & Verify Configuration and you should see that the URLs were successfully validated.
4. Any default URL that points to the Internet (https://archive.cloudera.com/XXX) should be removed.
5. Once done, click Close.

6. Now on the screen you should see the chosen CDH release (assuming you had one in your local repository) and any additional parcels you set to deploy (in the following example - CFM parcels)



7. Click Continue.

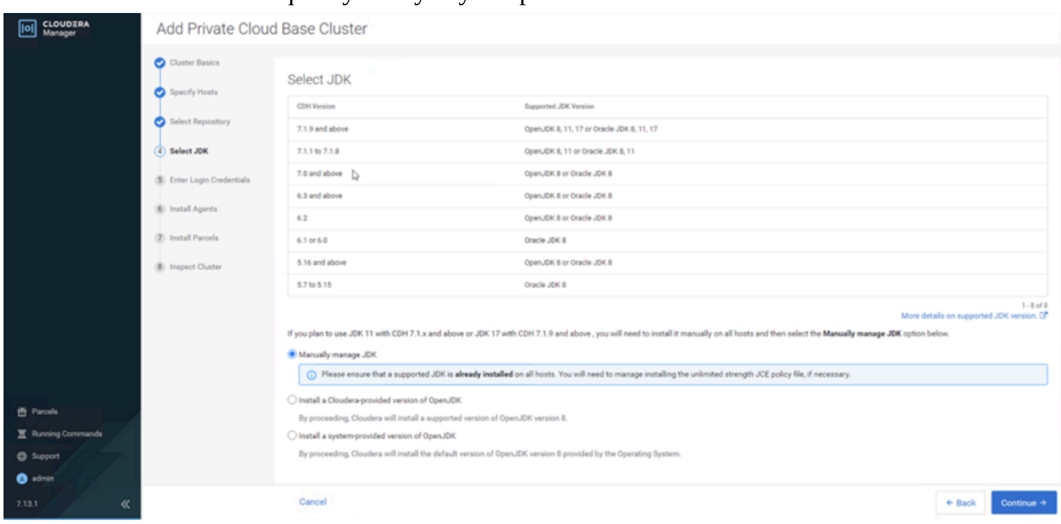
Select JDK

Note: CDP Data Center is no longer bundled with Oracle JDK software. Cloudera provides a supported version of OpenJDK.

In most cases for customer deployments, the customer is providing an OS deployment that comes pre-installed with the Java version that was asked for the cluster.

As such, you should select to Manually manage JDK.

The other cases are used only if you want to use Jdk 1.8 that comes preinstalled with Cloudera, though it is not recommended as it will require you anyway to update the version to the latest minor release.



After selecting the applicable boxes, click Continue.

Enter Login Credentials

1. Enter the root name or username for the root account that has password-less sudo privileges. (In the /etc/sudoers file, the entry for this should like this:

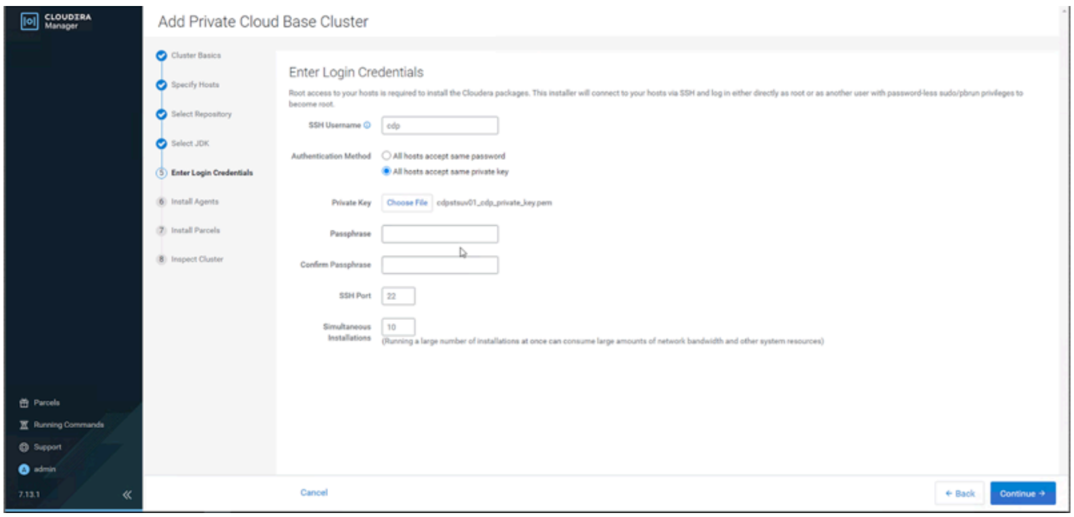
```
%<USERNAME> ALL=(ALL) NOPASSWD: ALL
```

2. Select an authentication method:

    - If you select the All hosts accept same password option for password authentication, enter and confirm the password.

      In most cases, customers will not provide the root credentials, rather they will provide a local user that is capable of performing sudo to user root without a password on any station. That user should be allowed by the customer to “hop” using ssh from the Cloudera Manager (CM) node to all cluster nodes without a password to allow the cluster installation and parcel distribution.
    - If you select the All hosts accept same private key option for public-key authentication, provide the matching OpenSSH private key file provided by the customer for the given local power user. In case the OpenSSH private key is encrypted (which is not the default case) please provide the passphrase to open the private key.

    You can modify the default SSH port if necessary.
3. Specify the maximum number of host installations to run at once. The default and recommended value is 10. You can adjust this based on your network capacity.



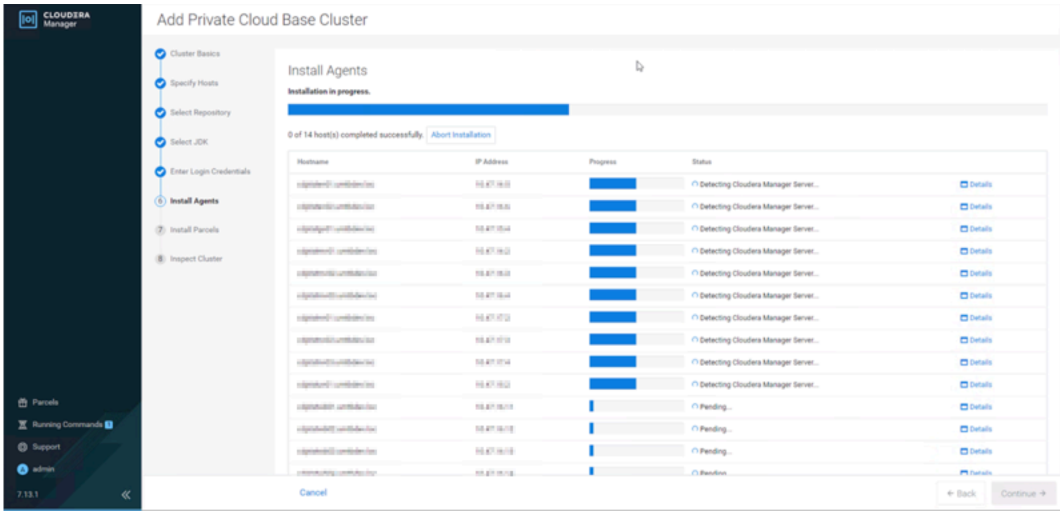
4. Click Continue.

The Install Agents page displays.

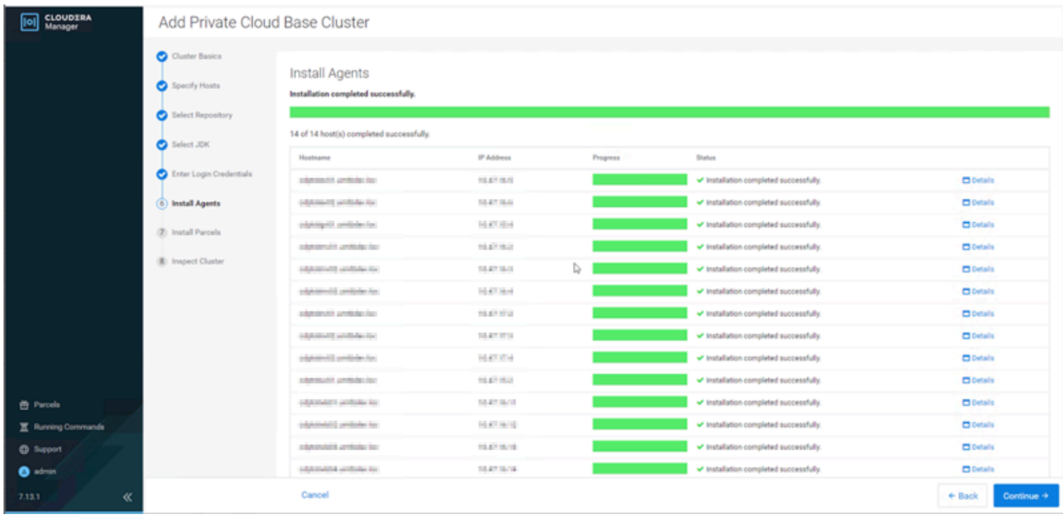
Install Agents

The Install Agents page displays the progress of the installation. You can click on the Details link to view the installation log of any host. Optionally, you can click the Abort Installation button to cancel the installation and then view the installation logs to troubleshoot the problem.

If the installation fails on any hosts, you can click the Retry Failed Hosts to retry all failed hosts.



After installing the Cloudera Manager Agent on all hosts, the following page will be displayed:

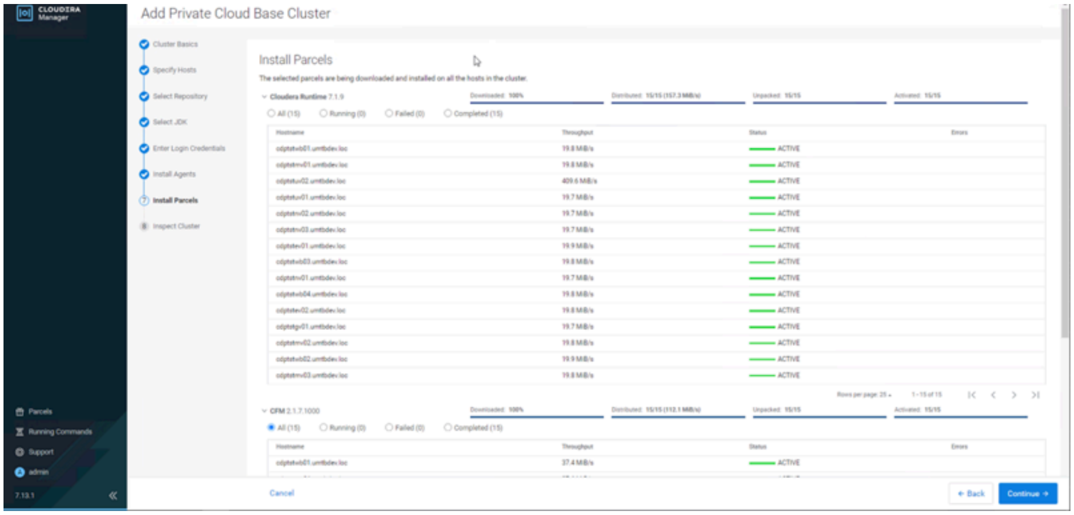


Click Continue.

The Install Parcels page displays.

Install Parcels

The Install Parcels page reports the installation progress of the parcels you selected earlier. After the parcels are downloaded, progress bars appear representing each cluster host. You can click on an individual progress bar for details about that host.



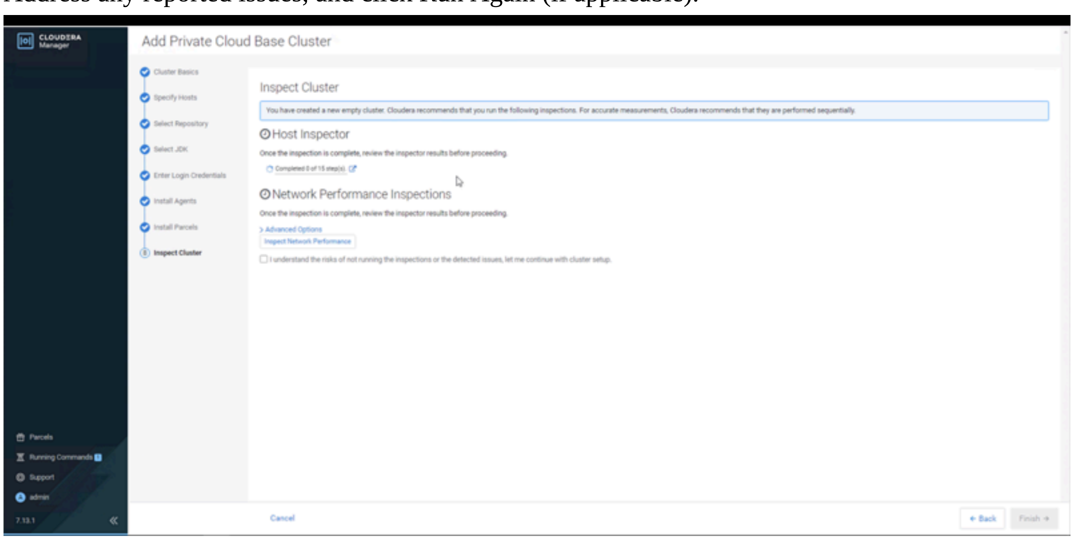
After the installation is complete, click Continue.

The Inspect Cluster page displays.

Inspect Cluster

The Inspect Cluster page provides a tool for inspecting network performance as well as the Host Inspector to search for common configuration problems. Cloudera recommends that you run the inspectors sequentially:

1. Run the Inspect Network Performance tool. You can click Advanced Options to customize some ping parameters.
2. After the network inspector completes, click Show Inspector Results to view the results in a new tab.
3. Address any reported issues, and click Run Again (if applicable).
4. Click Inspect Hosts to run the Host Inspector utility.
5. After the host inspector completes, click Show Inspector Results to view the results in a new tab.
6. Address any reported issues, and click Run Again (if applicable).



Important:

If you encounter any issues and need any guidance during this process, contact Cloudera support for further assistance.

If the reported issues cannot be resolved in a timely manner, and you want to abandon the cluster creation wizard to address them, then click Cancel.

After addressing any identified problems, select I understand the risks of not running the inspections or the detected issues, let me continue with cluster setup, and then click Finish.

This completes the Add Private Cloud Base Cluster wizard operation and launches the Add Cluster - Configuration wizard.

Continue to *Step 8: Set Up a Cluster Using the Wizard*.

Step 8: Set Up a Cluster Using the Wizard

After you complete the Add Cluster - Installation wizard, the Add Cluster - Configuration wizard automatically starts. The following sections guide you through each page of the wizard.

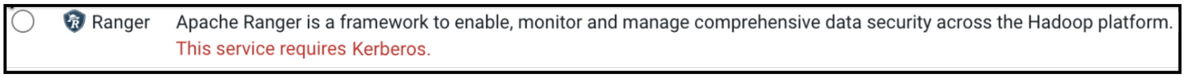
Select Services

The Select Services page allows you to select the services you want to install and configure.

Important:

In case you deploy a scratch cluster that has not yet been kerberized, you cannot choose any service that requires Kerberos as a pre-requisite, including Ranger, RangerKMS, Atlas, and Knox.

Each of these services will carry such note near it (see example for Ranger):



Choosing either of these services to be installed without having a Kerberos-based cluster **will lead to a non-functioning cluster**. It is advised to install these services separately only after enabling Kerberos on the cluster. For more information, see Set up a KDC.

After selecting the services you want to add, click Continue. The Assign Roles page displays.

Note that if you did not add Kerberos to the cluster at the beginning of the Wizard (i.e. setting Cloudera Manager as Key Distribution Center), you will need to choose only Customer Services option and pick the non-kerberos services you want to deploy, and only post enabling Kerberos, you will repeat the Wizard and complete the Kerberos based services.

Select one of the following options:

Regular (Base) Clusters

Data Engineering

Process develop, and serve predictive models.

Services included: HDFS, YARN, YARN Queue Manager, Ranger, Atlas, Hive, Hive on Tez, Spark, Oozie, and Hue

Data Mart

Browse, query, and explore your data in an interactive way.

Services included: HDFS, Ranger, Atlas, Hive, and Hue

Operational Database

Real-time insights for modern data-driven business.

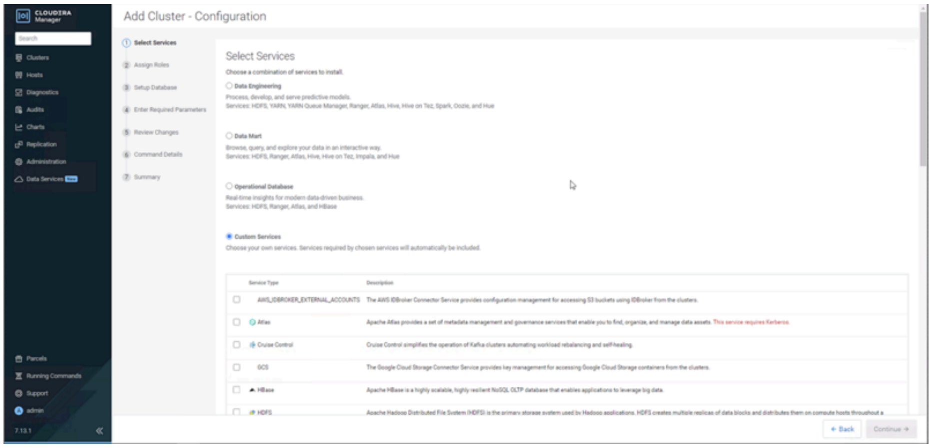
Services included: HDFS, Ranger, Atlas, and HBase

Custom Services

Choose your own services. Services required by chosen services will automatically be included.

Note:

In most cases, you will choose Custom Services and will pick the services that were purchased by the customer. Remember that if this is a scratch cluster installation and KDC/TLS were not installed, any service marked with a red note stating **This service requires Kerberos** can be installed now.



After picking the services you would like to deploy, for each service you will be asked to choose the nodes on which the service will be deployed. Assign the service as instructed by Cloudera according to the Cloudera pre-defined role distribution design table.

Note:

Though the Wizard installs the services chosen, some functions that are required for these services are not installed during the Wizard's initial installation due to dependencies and will need to be added later manually, such as service GateWays for HBASE, HDFS, Solr, and Yarn. To add them manually later, once the cluster is up, simply go to the service instances page, choose to add a role instance, pick the Gateway instance, and assign it according to the Cloudera pre-defined role distribution design table to the correct nodes.

Assign Roles

The Assign Roles page suggests role assignments for the hosts in your cluster.

You can click on the hostname for a role to select a different host. You can also click the View By Host button to see all the roles assigned to a host.

After assigning all of the roles for your services, click Continue. The Setup Database page displays.

Setup Database

On the Setup Database page, you can enter the database hosts, names, usernames, and passwords you created in *Step 3: Install and Configure Databases*.

Important:

Currently the Cloudera Manager wizard does not support connection testing to an external Oracle as Metastore DB for the Hue/Yarn Queue Manager services.

In case the Cloudera Manager wizard is being used to deploy a fresh cluster, do not install Hue or Yarn Queue Manager, and once the Cloudera Manager wizard finishes, follow the guidelines of each service on how to manually set it to work with Oracle.

For services that support it, you can add finer-grained customizations using a JDBC URL override.

Select the database type and enter the database name, username, and password for each service.

For MariaDB, select MySQL.

For services that support it, to specify a JDBC URL override, select Yes in the Use JDBC URL Override dropdown menu. You must also specify the database type, username, and password.

If you are using a TLS 1.2-enabled MySQL, PostreSQL, or MariaDB databases, or TCPS-enabled Oracle database for your Runtime service, then see the service-specific instructions under Database setup details for cluster services for TLS 1.2/TCPS-enabled databases on page 231.

Tip: MySQL 8 uses caching_sha2_password as the default authentication method. Follow the instructions in Database setup details for Hive Metastore for TLS 1.2/TCPS-enabled databases for proper configuration.

Click Test Connection to validate the settings. If the connection is successful, a green checkmark and the word Successful appears next to each service. If there are any problems, the error is reported next to the service that failed to connect.

After verifying that each connection is successful, click Continue. The Review Changes page displays.

Database setup details for cluster services for TLS 1.2/TCPS-enabled databases

Cloudera Manager supports TLS connection to backend databases for Hadoop services such as Hue, Ranger, Oozie etc. You must modify the configuration of the services in CM to set a TLS 1.2 connection with their respective databases.

Database setup details for Hue for TLS 1.2/TCPS-enabled databases

Hue automatically uses TLS 1.2 when you create a CDP cluster with the Auto-TLS option. You can also enable TLS in Hue configurations after you add the Hue service to your cluster using Cloudera Manager.

About this task

If TLS 1.2 is enabled on the database servers, and the databases are restricted or enforced to use TLS 1.2, then Hue automatically uses the TLS1.2-compatible ciphers to communicate with the database securely. You do not have to configure any setting in Hue's Advanced Configuration Snippet or any other configurations. This is applicable when using MySQL, MariaDB, or PostgreSQL databases as a backend database for Hue.

To restrict the MySQL and MariaDB databases to use TLS 1.2, set the value of the require_secure_transport to true in the my.cnf file.

To enable TLS 1.2 on the Hue instance, go to Cloudera Manager Clusters Hue service Configurations and select the Enable TLS/SSL for Hue option.

If TCPS is enabled on the Oracle database, then you can specify the connection string containing the “TCPS” protocol in the Database SID field.

Before you begin

- You must have enabled TLS 1.2 or TCPS on the Hue database.
- You must have created database users.
- You must have installed the MySQL client (for MySQL or MariaDB databases).
- You must have installed the psycopg2 Python package (for PostgreSQL database).

Procedure

1. Select the appropriate database type from the Type drop-down menu.
2. Enter the Fully Qualified Domain Name (FQDN) of the host on which you have installed the database in the Database Hostname field.

   If the database is not running on its default port, then specify the port number in the following format: *[***HUE-DB-HOST***]:[***DB-PORT***]*.

   Where,

   *[***HUE-DB-HOST***]* is the FQDN of the database host

   *[***DB-PORT***]* is the database port

3. Specify the database name in the Database Name field.

   For Oracle databases, specify the SID in the Database SID field.

   If you are using Oracle Service Name instead of SID, then specify the database name in the following format:

   ```
   [***HUE-DB-HOST***]:1521/[***SERVICE-NAME***]
   ```

   Where,

   *[***HUE-DB-HOST***]* is the FQDN of the database host

   *[***SERVICE-NAME***]* is the Oracle service name

   You can also specify the following connection string in the Database Name field:

   ```
   (DESCRIPTION=(LOAD_BALANCE=off)(FAILOVER=on)(CONNECT_TIMEOUT=5)(TRANSPOR
   T_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(ADDRESS=(PROTOCOL=TCPS)(HOST=[***HUE-
   DB-HOST***])(PORT=[***HUE-DB-PORT***]))(CONNECT_DATA=(SERVICE_NAM
   E=[***SERVICE-NAME***])(SECURITY =
    (MY_WALLET_DIRECTORY = /[***PATH-TO-WALLET-FILE***])))
   ```

   Where,

   *[***HUE-DB-HOST***]* is the FQDN of the database host

   *[***HUE-DB-PORT***]* is the port for the Hue database

   *[***SERVICE-NAME***]* is the Oracle service name

   *[***PATH-TO-WALLET-FILE***]* is the location at which you have copied the wallet file (cwallet.sso) on the Hue host
4. Enter the database username and password you set up for Hue database in the Username and Password fields.
5. Click Test Connection.

   If the connection test fails, review your configuration, fix any errors, and rerun the connection test.
6. Click Continue to continue with cluster installation.

Database setup details for Ranger KMS for TLS 1.2/TCPS-enabled databases

Updating the Ranger KMS Database JDBC Url Override and additional configuration to connect to the secure databases.

Before you begin

- Ensure that TLS 1.2 has already been enabled on the Ranger database.
- Ensure Use JDBC Override URL has been set toYes in the Setup Database page.

Procedure

1. Select / Enter the following configuration values depending on the database type.

   - MySQL

Label	Configuration	Value
Ranger KMS Database Type	ranger_kms_database_type	MySQL
Ranger KMS Database User	ranger_kms_database_user	<username>
Ranger KMS Database User Password	ranger_kms_database_password	<password>

Label	Configuration	Value
Ranger KMS Database JDBC Url Override	ranger_kms_database_jdbc_url	jdbc:mysql://*[***DB-HOST***]*:*[***DB-PORT***]*/*[***RANGER-KMS-DB-NAME***]*?sslMode=VERIFY_CA&trustCertificateKeyStoreU

- Oracle

Label	Configuration	Value
Ranger KMS Database Type	ranger_kms_database_type	Oracle
Ranger KMS Database User	ranger_kms_database_user	<username>
Ranger KMS Database User Password	ranger_kms_database_password	<password>
Ranger KMS Database JDBC Url Override	ranger_kms_database_jdbc_url	jdbc:oracle:thin:@tcps://*[***DB-HOST***]*:*[***DB-PORT***]*:*[***SERVICE_NAME***]*?javax.net.ssl.trustStore=*[***PATH_TO_TRUSTST*

- PostgreSQL

Label	Configuration	Value
Ranger KMS Database Type	ranger_kms_database_type	PostgreSQL
Ranger KMS Database User	ranger_kms_database_user	<username>
Ranger KMS Database User Password	ranger_kms_database_password	<password>
Ranger KMS Database JDBC Url Override	ranger_kms_database_jdbc_url	jdbc:postgresql://*[***DB-HOST***]*:*[***DB-PORT***]*/*[***RANGER-KMS-DB***]*?sslmode=verify-full&sslrootcert=*[***PATH-TO-DATABASE-SERVER-CERTIFICATE***]*&enabledTLSProtocols=TLSv

2. Click Test Connection.
3. Once the test connection succeeds, click Continue.

Database setup details for Ranger for TLS 1.2/TCPS-enabled databases

Updating the Ranger Database JDBC Url Override and additional configuration to connect to the secure databases.

Before you begin

- Ensure that TLS 1.2 has already been enabled on the Ranger database.
- Ensure Use JDBC Override URL has been set toYes in the Setup Database page.

Procedure

1. Select / Enter the following configuration values depending on the database type.

   - MySQL

Label	Configuration	Value
Ranger Database Type	ranger_database_type	MySQL
Ranger Database User	ranger_database_user	<username>
Ranger Database User Password	ranger_database_password	<password>
Ranger Database JDBC Url Override	ranger_database_jdbc_url	jdbc:mysql://[***DB-HOST***]:[***DB-PORT***]/[***RANGER-DB-NAME***]?sslMode=VERIFY_CA&trustCertificateKeyStore

   - Oracle

Label	Configuration	Value
Ranger Database Type	ranger_database_type	Oracle
Ranger Database User	ranger_database_user	<username>
Ranger Database User Password	ranger_database_password	<password>
Ranger Database JDBC Url Override	ranger_database_jdbc_url	dbc:oracle:thin:@tcps://[***DB-HOST***]:[***DB-PORT***]:[***SERVICE_NAME***]?javax.net.ssl.trustStore=[***PATH_TO_TRUSTST

   - PostgreSQL

Label	Configuration	Value
Ranger Database Type	ranger_database_type	PostgreSQL
Ranger Database User	ranger_database_user	<username>
Ranger Database User Password	ranger_database_password	<password>
Ranger Database JDBC Url Override	ranger_database_jdbc_url	jdbc:postgresql://[***DB-HOST***]:[***DB-PORT***]/[***RANGER-DB***]?sslmode=verify-full&sslrootcert=[***PATH-TO-DATABASE-SERVER-CERTIFICATE***]&enabledTLSProtocols=TLSv

2. Click Test Connection.
3. Once the test connection succeeds, click Continue.

Database setup details for Oozie for TLS 1.2/TCPS-enabled databases

You can configure Oozie's database connection properties after you add Oozie as a service.

Procedure

1. Add Oozie as a service.
2. In the configuration wizard, configure the database type, host, port, and database name.

3. Before starting Oozie, fine-tune your Oozie's database connection, including adding your database client trustStore or certificate location to the connection string.

   For more details, see *Fine-tuning Oozie's database connection*.

Related Information

Fine-tuning Oozie's database connection

Database setup details for Streams Messaging Manager for TLS 1.2/TCPS-enabled databases

Learn how you can configure Streams Messaging Manager (SMM) to securely connect to its database using TLS 1.2 when installing a new cluster.

About this task

When installing a new cluster, Streams Messaging Manager's database connection is set up during the Setup Database step of the Add Cluster - Configuration wizard. Complete the following when you reach this step in the wizard to configure TLS 1.2.

Before you begin

- Ensure that TLS 1.2 has already been enabled on the SMM database.
- Ensure that a truststore file containing the database certificate is available on the SMM hosts. Additionally, ensure that you know the location of the file and that the user SMM runs as has access to the file. The default user for SMM is streamsmsgmgr.

Procedure

1. Select the appropriate database type from the Type drop-down list.
2. Select yes from the Use JDBC URL Override drop-down list.
3. Enter the database username and password you set up for SMM in the Username and Password fields.

   The username and password you must enter are set up in a previous step of the installation. For more information, see *Configuring the Database for Streaming Components*.
4. Enter an appropriate JDBC URL in the JDBC URL field.

   The JDBC URL must contain all necessary properties needed for SMM to establish a secure connection with its database. Use the following templates to construct the JDBC URL.
   MySQL

   ```
   jdbc:mysql://[***DB HOST***]:[***DB PORT***]/[***DB NAME***]?
   useSSL=true&trustCertificateKeyStoreUrl=file://[***TRUSTSTORE
    PATH***]&trustCertificateKeyStoreType=jks&trustCertificateKeySto
   rePassword=[***TRUSTSTORE PASSWORD***]&enabledTLSProtocols=TLS
   v1.2
   ```

   PostgreSQL

   ```
   jdbc:postgresql://[***DB HOST***]:[***DB PORT***]/[***DB
    NAME***]?useSSL=true&trustCertificateKeyStoreUrl=fil
   e://[***TRUSTSTORE PATH***]&trustCertificateKeyStoreType=jks&tr
   ustCertificateKeyStorePassword=[***TRUSTSTORE PASSWORD***]&ena
   bledTLSProtocols=TLSv1.2
   ```

   Oracle

   ```
   jdbc:oracle:thin:@tcps://[***DB HOST***]:[***DB PORT***]/[***DB
    NAME***]?javax.net.ssl.trustStore=[***TRUSTSTORE PATH***]&jav
   ```

```
ax.net.ssl.trustStorePassword=[***TRUSTSTORE PASSWORD***]&oracle.
net.ssl_server_dn_match=false
```

- Replace *[***DB HOST***]*, *[***DB PORT***]*, and *[***DB NAME***]* with the host, port, and name of the database.
- Replace *[***TRUSTSTORE PATH***]* with the full path to a truststore that contains the database certificate. The truststore must be available on the host that SMM is deployed on. Additionally, the user that the SMM service runs as, default is streamsmsgmgr, must have access to the file.
- Replace *[***TRUSTSTORE PASSWORD***]* with the password used to access the truststore you specify in *[***TRUSTSTORE PATH***]*.

5. Click Test Connection.

   If the connection test fails, review your configuration, fix any errors, and rerun the connection test.
6. Click Continue to continue with cluster installation.

Results

The SMM service establishes a secure connection with its database.

Related Information

Configuring the Database for Streaming Components

Database setup details for Schema Registry for TLS 1.2/TCPS-enabled databases

Learn how you can configure Schema Registry to securely connect to its database using TLS 1.2 when installing a new cluster.

About this task

When installing a new cluster, Schema Registry's database connection is set up during the Setup Database step of the Add Cluster - Configuration wizard. Complete the following when you reach this step in the wizard to configure TLS 1.2.

Before you begin

- Ensure that TLS 1.2 is enabled on the Schema Registry database.
- Ensure that a truststore file containing the database certificate is available on the Schema Registry hosts. Additionally, ensure that you know the location of the file and that the user Schema Registry runs as has access to the file. The default user for Schema Registry is schemaregistry.

Procedure

1. Select the appropriate database type from the Type drop-down list.
2. Select yes from the Use JDBC URL Override drop-down list.
3. Enter the database username and password you set up for Schema Registry in the Username and Password fields.

   The username and password you must enter are set up in a previous step of the installation. For more information, see *Configuring the Database for Streaming Components*
4. Enter an appropriate JDBC URL in the JDBC URL field.

   The JDBC URL must contain all necessary properties needed for Schema Registry to establish a secure connection with its database. Use the following templates to construct the JDBC URL.

   MySQL

```
jdbc:mysql://[***DB HOST***]:[***DB PORT***]/[***DB NAME***]?
useSSL=true&trustCertificateKeyStoreUrl=file://[***TRUSTSTORE
 PATH***]&trustCertificateKeyStoreType=jks&trustCertificateKeySto
```

```
rePassword=[***TRUSTSTORE PASSWORD***]&enabledTLSProtocols=TLS
v1.2
```

PostgreSQL

```
jdbc:postgresql://[***DB HOST***]:[***DB PORT***]/[***DB
 NAME***]?useSSL=true&trustCertificateKeyStoreUrl=fil
e://[***TRUSTSTORE PATH***]&trustCertificateKeyStoreType=jks&tr
ustCertificateKeyStorePassword=[***TRUSTSTORE PASSWORD***]&ena
bledTLSProtocols=TLSv1.2
```

Oracle

```
jdbc:oracle:thin:@tcps://[***DB HOST***]:[***DB PORT***]/[***DB
 NAME***]?javax.net.ssl.trustStore=[***TRUSTSTORE PATH***]&jav
ax.net.ssl.trustStorePassword=[***TRUSTSTORE PASSWORD***]&oracle.
net.ssl_server_dn_match=false
```

- Replace *[***DB HOST***]*, *[***DB PORT***]*, and *[***DB NAME***]* with the host, port, and name of the database.
- Replace *[***TRUSTSTORE PATH***]* with the full path to a truststore that contains the database certificate. The truststore must be available on the host that Schema Registry is deployed on. Additionally, the user that the Schema Registry service runs as, default is schemaregistry, must have access to the file.
- Replace *[***TRUSTSTORE PASSWORD***]* with the password used to access the truststore you specify in *[***TRUSTSTORE PATH***]*.

5. Click Test Connection.

If the connection test fails, review your configuration, fix any errors, and rerun the connection test.

6. Click Continue to continue with cluster installation.

Results

The Schema Registry service establishes a secure connection with its database.

Related Information

Configuring the Database for Streaming Components

Database setup details for Hive Metastore for TLS 1.2/TCPS-enabled databases

Learn how you can configure the Hive (Hive Metastore) service to securely connect to its database using TLS 1.2 when installing a new cluster.

About this task

When installing a new cluster, Hive Metastore's database connection is set up during the Setup Database step of the Add Cluster - Configuration wizard. Complete the following when you reach this step in the wizard to configure TLS 1.2.

Before you begin

- Ensure that TLS 1.2 is enabled on the Hive Metastore database.
- Ensure that the Database SSL trusted certificates are exported and added to the Java truststore file.

Procedure

1. In the **Setup Database** page for Hive, click the Type drop-down list and select the appropriate database type.
2. Click the Use JDBC URL Override drop-down list and select Yes.

3. In the JDBC URL field, specify the appropriate JDBC URL connection string.

   The JDBC URL must contain all necessary properties required for Hive Metastore to establish a secure connection with its database. Use the following templates to construct the JDBC URL:

   MySQL

   ```
   jdbc:mysql://[***DB-HOST***]:[***DB-PORT***]/[***DB-NAME***]?ssl
   Mode=VERIFY_CA&trustCertificateKeyStoreUrl=file://[***TRUSTSTORE
   -PATH***]&trustCertificateKeyStoreType=jks&trustCertificateKeySt
   orePassword=[***TRUSTSTORE-PASSWORD***]&enabledTLSProtocols=TLSv
   1.2
   ```

   PostgreSQL

   ```
   jdbc:postgresql://[***DB-HOST***]:[***DB-PORT***]/[***DB-NAME***
   ]?sslMode=VERIFY_CA&trustCertificateKeyStoreUrl=file://[***TRUST
   STORE-PATH***]&trustCertificateKeyStoreType=jks&trustCertificate
   KeyStorePassword=[***TRUSTSTORE-PASSWORD***]&enabledTLSProtocols
   =TLSv1.2
   ```

   MariaDB

   ```
   jdbc:mysql://[***DB-HOST***]:[***DB-PORT***]/[***DB-NAME***]?ssl
   Mode=VERIFY_CA&trustCertificateKeyStoreUrl=file://[***TRUSTSTORE
   -PATH***]&trustCertificateKeyStoreType=jks&trustCertificateKeySt
   orePassword=[***TRUSTSTORE-PASSWORD***]&enabledTLSProtocols=TLSv
   1.2
   ```

   Oracle

   ```
   jdbc:oracle:thin:@tcps://[***DB-HOST***]:[***DB-PORT***]/[***DB-
   NAME***]?javax.net.ssl.trustStore=[***TRUSTSTORE-PATH***]&javax.
   net.ssl.trustStorePassword=[***TRUSTSTORE-PASSWORD***]&oracle.ne
   t.ssl_server_dn_match=false
   ```

   Where,

   - [***DB-HOST***], [***DB-PORT***], and [***DB-NAME***] represent the Host, Port, and Database name used for the Hive Metastore service.
   - [***TRUSTSTORE-PATH***] represents the path to the Java truststore file.
   - [***TRUSTSTORE-PASSWORD***] represents the password used to access the Java truststore file.
4. Click Test Connection to validate the settings.

   If the connection fails, review your configuration, fix any errors, and test the connection again.
5. Click Continue to proceed with the installation.

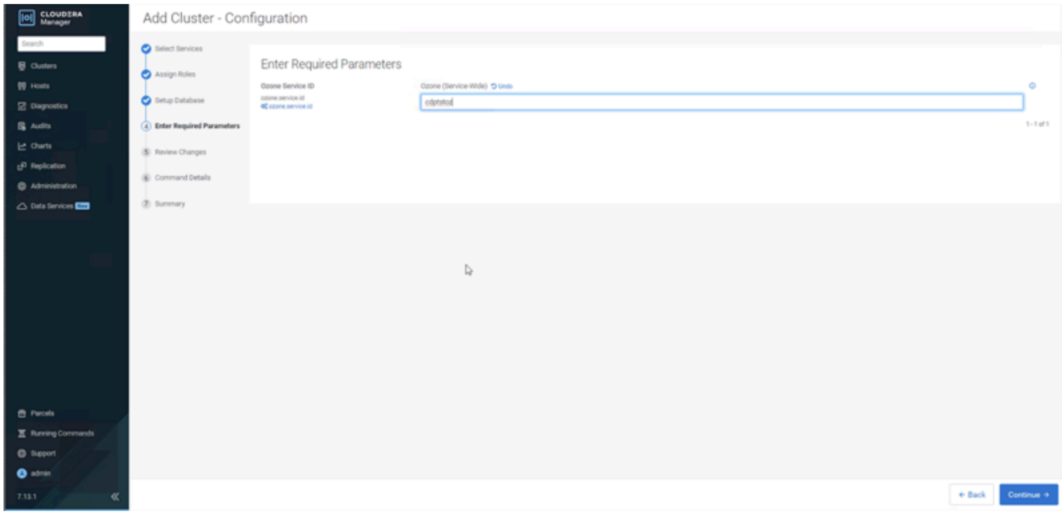
Results

The Hive Metastore service establishes a secure connection with its database.

Enter Required Parameters

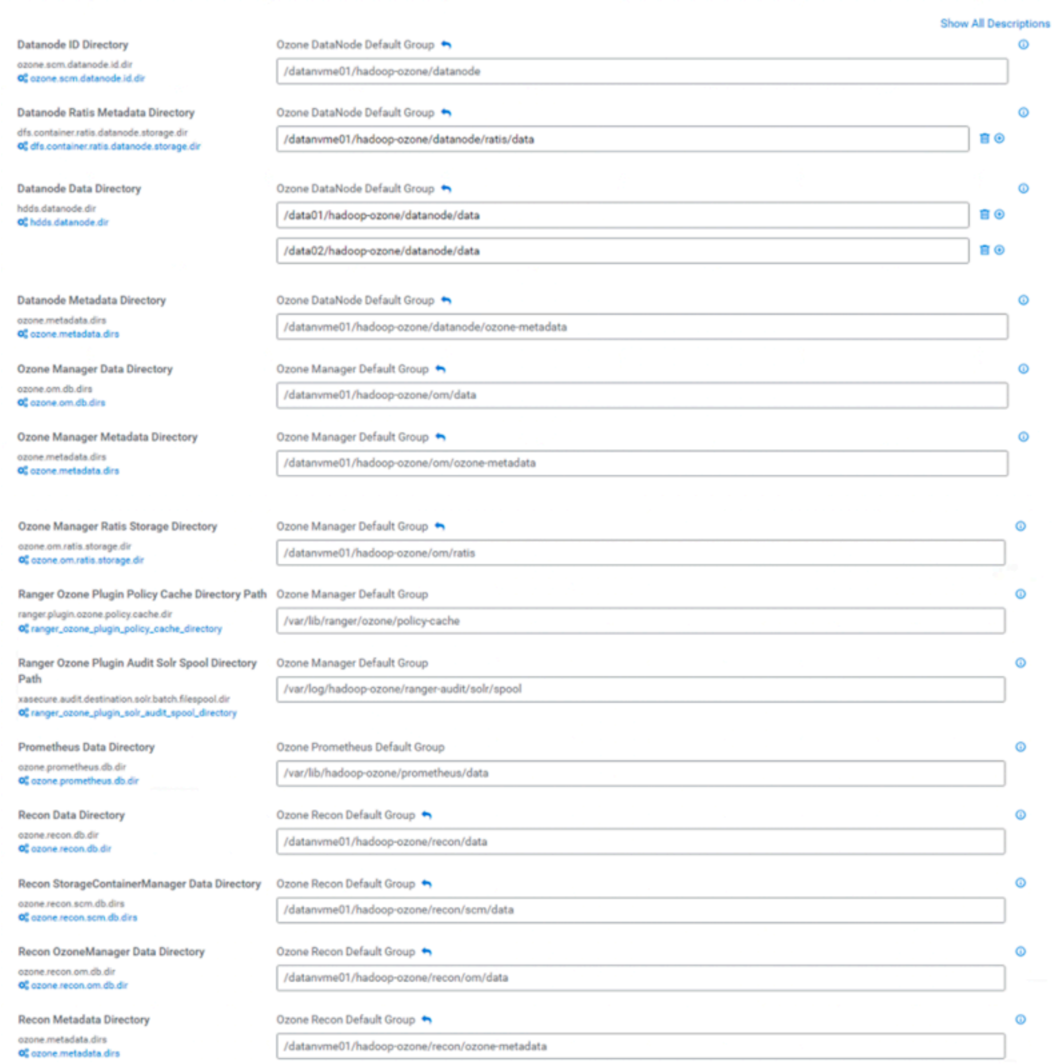
The **Enter Required Parameters** page lists required parameters for all the Cloudera services that were chosen for installation.

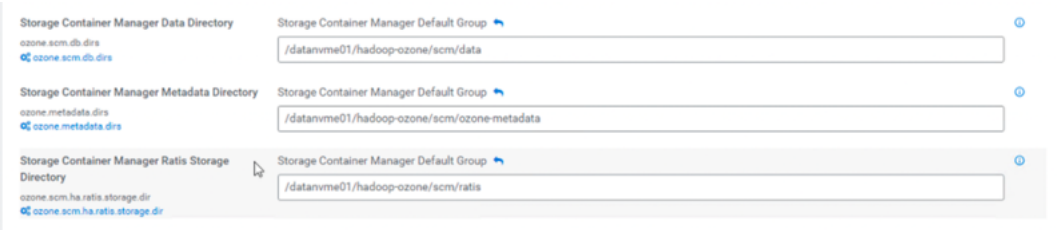
The first page (for example, assuming you have added Ozone) will ask you to enter the Ozone Service ID (unique string identifies the Ozone Service across the platform). See the following example:

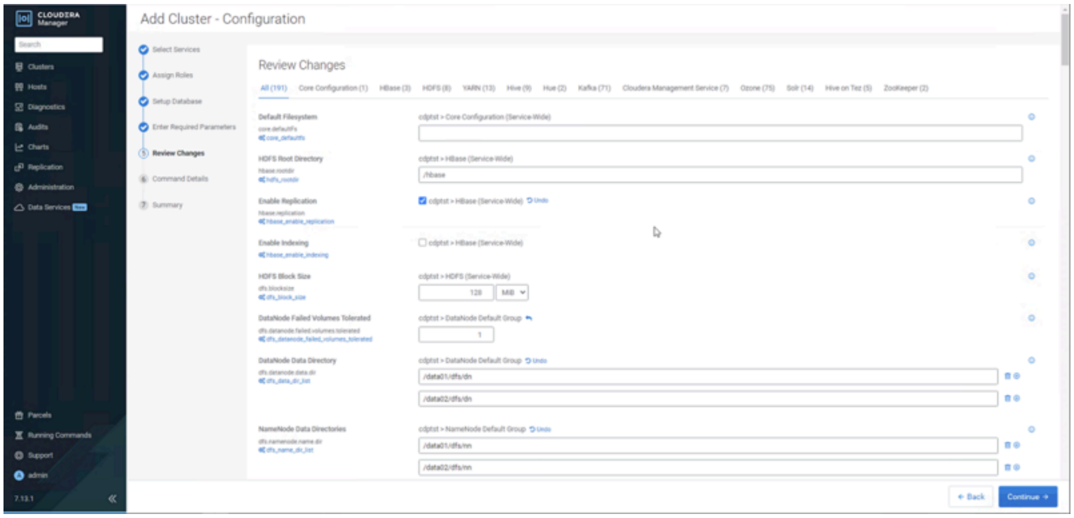


After you enter the Ozone Service ID. Click Continue.

On the next screen, you will need to update values according to the project settings on each of the services you have chosen. You can do it on the “All” tab or choose to update per service. As Example for change - replacing the default location of the Data Nodes / NameNode / SCM folders from the OS location (/var/lib/XXX) to physical disks in case of deployment the service over BareMetal nodes. For example, see the following figures:







Note:

There is a need for some changes that you must add on the configuration before continuing with the Wizard.

In case Ozone is chosen as a service, and the version deployed is lower than 7.3.1.0 there is a change that needs to be implemented to avoid collision between Ozone and HDFS services running on the same node.

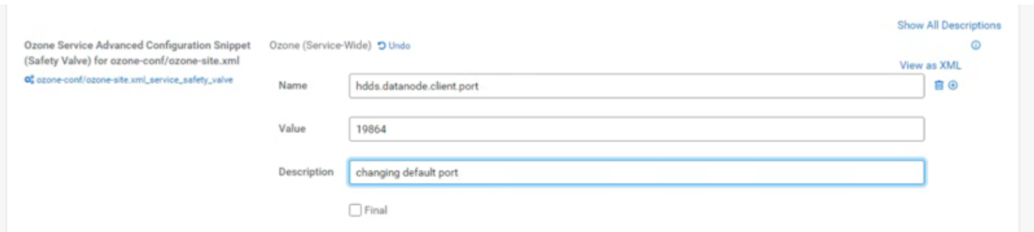
In case Hue is chosen (all CDH versions), you must add a safety value. (see below)

For Zookeeper, you must set an additional flag.

For Ozone:

1. Please open another parallel session of the Cloudera Manager and go to Ozone Configuration.
2. Search for ozone-conf/ozone-site.xml_service_safety_value and add a new attribute by clicking on the small + sign to the right:
   - Name: hdds.datanode.client.port
   - Value: 19864
   - Description: Changing default port to avoid collision with HDFS service

   For example, see the following figure:



For Hue (if chosen for installation):

1. Go to Hue Configuration and click on the Database category to the left.
2. Validate all DB values (user/password/DB Schema name/DB name).

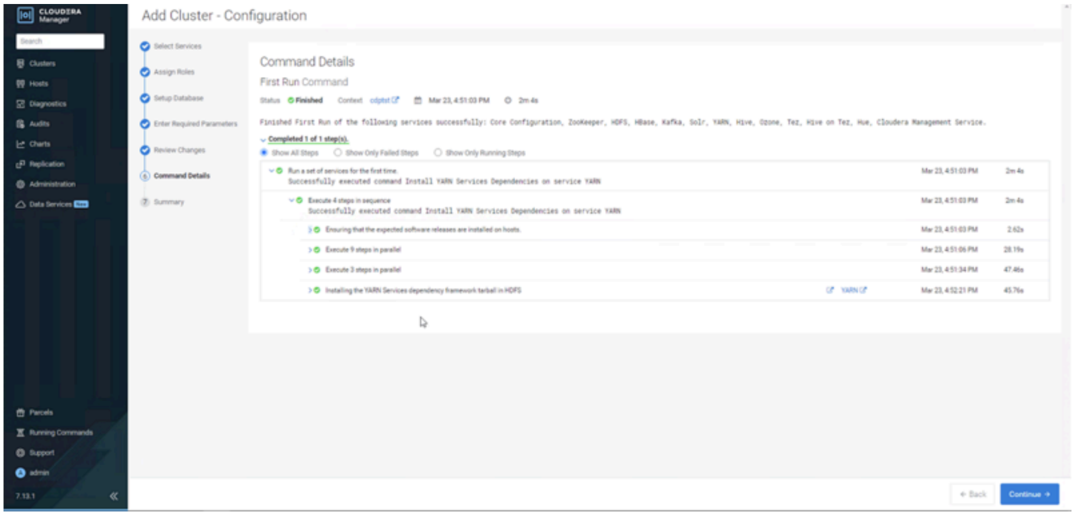
3. Search for hue_service_safety_value and add the following and save it:

```
[desktop]
[[database]]
port=0
[beewax]
```

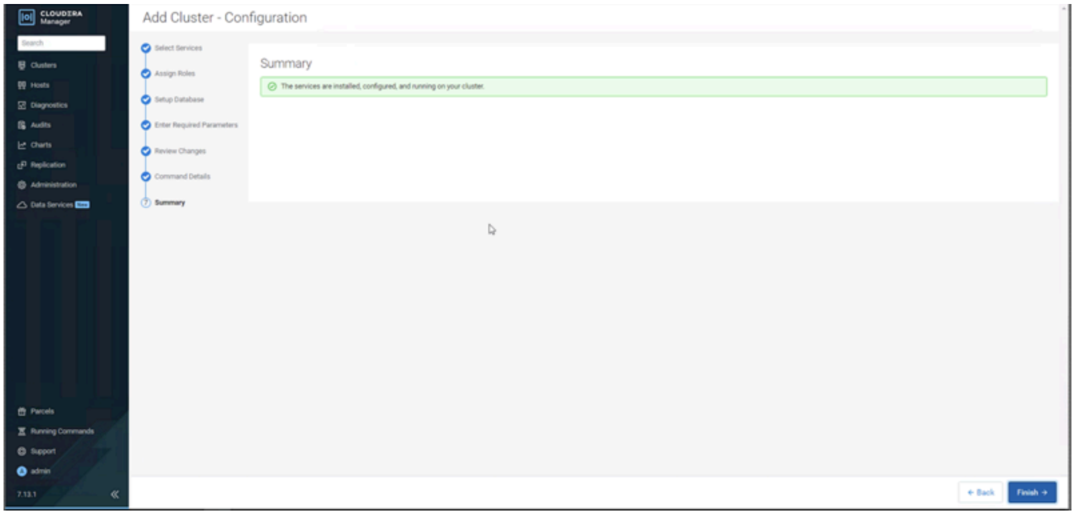
For Zookeeper:

1. Go to the Zookeeper configuration and check the box near zookeeper_datadir_autocreate.
2. Save the configuration.

Once all configurations are validated, return to the Wizard running screen and click Continue. The Wizard will start deploying the changes.



When the deployment is done, the following message is displayed. Click Finish to close the Wizard.



The main cluster page will appear, most likely with some configuration issues. If one of the services wants to redeploy the configuration, do it, and the cluster will restart itself.

Review Changes

The Review Changes page lists default and suggested settings for several configuration parameters, including data directories.

Warning: Do not place DataNode data directories on NAS devices. When resizing an NAS, block replicas can be deleted, which results in missing blocks.

Review and make any necessary changes, and then click Continue. The Command Details page displays.

Configure Kerberos

Kerberos is a network authentication protocol that provides security for your cluster. The Configure Kerberos page allows you to enable Kerberos for your cluster.

After selecting Enable kerberos for this cluster, install kerberos client libraries, according to your OS type, on all hosts before proceeding.

- RHEL/ CentOS

  ```
  $ yum install krb5-workstation krb5-libs
  ```

  If Redhat IPA is used as the KDC,

  ```
  yum install freeipa-client
  ```

- SUSE

  ```
  zypper install krb5-client
  ```

  If Redhat IPA is used as the KDC,

  ```
  zypper install freeipa-client
  ```

- Ubuntu

  ```
  apt-get install krb5-user
  ```

  If Redhat IPA is used as the KDC,

  ```
  apt-get install freeipa-client
  ```

Configure Datanode ports

Configure the privileged ports required by the datanodes in a secure HDFS service by selecting values for DataNode Transceiver Port and DataNode HTTP Web UI port.

Command Details

The Command Details page lists the details of the First Run command.

You can expand the running commands to view the details of any step, including log files and command output. You can filter the view by selecting Show All Steps, Show Only Failed Steps, or Show Only Running Steps.

If cluster deployment fails, ensure to click Resume in the wizard after you fix any issues. If you do not click Resume, the cluster may not be in a functional state.

After the First Run command completes, click Continue to go to the Summary page.

Summary

The Summary page reports the success or failure of the setup wizard.

Click Finish to complete the wizard. The installation is complete.

Cloudera recommends that you change the default password as soon as possible by clicking the logged-in username at the top right of the home screen and clicking Change Password.

Tuning JVM Garbage Collection

When using OpenJDK 11[*], Cloudera Manager and most Cloudera Runtime services use G1GC as the default method of garbage collection. (Java 8 used "ConcurrentMarkSweep" (CMS) for garbage collection.) When using G1GC, the pauses for garbage collection are shorter, so components will usually be more responsive, but they are more sensitive to overcommitted memory usage. You should monitor memory usage to determine whether memory is overcommitted.

Important: Cloudera Manager Server with JDK 8 does not support G1GC.

Cloudera Manager alerts you when memory is overcommitted on cluster hosts. To view these alerts and adjust the allocations:

1. Log in to the Cloudera Manager Admin Console
2. Go to HomeConfigurationConfiguration Issues.
3. Look for entries labeled Memory Overcommit Validation Threshold and note the hostname of the affected host.
4. Go to HostsAll Hosts and click on the affected host.
5. Click the Resources tab.
6. Scroll down to the Memory section.

   A list of role instances and their memory allocations are displayed. The Description column displays the configuration property name where the memory allocation can be set.
7. To adjust the memory allocation, search for the configuration property and adjust the value to reduce the overcommitment of memory. You may need to move some roles to other hosts if there is not sufficient memory for the roles running on the host.
8. After making any changes, Cloudera Manager will indicate that the service has a stale configuration and prompt you to restart the service.

You may also need to adjust the Java options used to start Java processes. You can add Java startup options using Cloudera Manager configuration properties that are available for all service roles. Cloudera has provided default arguments for some of the services where they are needed. You can add to these, or completely override all of the provided Java options. For more information on configuring G1GC. see The OpenJDK documentation.

If default options are provided, the role configuration specifies a single value, {{JAVA_GC_ARGS}}. This value is a placeholder for the default Java Garbage Collection options provided with Cloudera Manager and Cloudera Runtime.

To modify Java options:

1. Log in to the Cloudera Manager Admin Console.
2. Go to the service where you want to modify the options. (For the Cloudera Manager Service Monitor, select the Cloudera Management Service.)
3. Select the Configuration tab.
4. Enter "Java" in the search box.
5. Locate the Java Configuration Options property named for the role you want to modify. For example, in the HDFS service, you will see parameters like Java Configuration Options for DataNode and Java Configuration Options for JournalNode.
6. To add to the Java options, enter additional options before or after the {{JAVA_GC_ARGS}} placeholder, separated by spaces. For example:

   ```
   {{JAVA_GC_ARGS}} -XX:MaxPermSize=512M
   ```
7. To replace the default Java options, delete the {{JAVA_GC_ARGS}} placeholder and replace it with one or more Java options, separated by spaces.
8. The service will now have a stale configuration and must be restarted. See Restarting a service.

[*] Azul OpenJDK, OpenJDK 8, and OpenJDK 11 are TCK certified for CDP.

Table 33: Default Java Options

Service and Role	Default Java 8 Options	Default Java 11 Options

• Cloudera Manager Service Monitor	`-XX:+UseConcMarkSweepGC`
	`-XX:+UseParNewGC`
	To enable G1GC:
	`-XX:+UseG1GC`
	`-XX:-UseConcMarkSweepGC`
	`-XX:-UseParNewGC`	

• HDFS DataNode
• HDFS NameNode
• HDFS Secondary NameNode
	`-XX:+UseParNewGC`
	`-XX:+UseConcMarkSweepGC`
	`-XX:CMSInitiatingOccupancyFracti`
	`-XX:+CMSParallelRemarkEnabled`
		`-XX:+UseConcMarkSweepGC`
		`-XX:CMSInitiatingOccupancyFraction=70`
		`-XX:+CMSParallelRemarkEnabled`

• Hive Metastore Server
• HiveServer 2
• WebHCat Server
	`-XX:+UseParNewGC`
	`-XX:+UseConcMarkSweepGC`
	`-XX:CMSInitiatingOccupancyFraction=70`
	`-XX:+CMSParallelRemarkEnabled`
		None, G1GC is enabled by default.

• HBase REST Server
• HBase Thrift Server
• HBase Master
• HBase RegionServer
	`-XX:+UseParNewGC`
	`-XX:+UseConcMarkSweepGC`
	`-XX:CMSInitiatingOccupancyFraction=70`
	`-XX:+CMSParallelRemarkEnabled`
		None, G1GC is enabled by default.

• HBase Region Server	`-XX:+UseParNewGC`
	`-XX:+UseConcMarkSweepGC`
	`-XX:CMSInitiatingOccupancyFraction=70`
	`-XX:+CMSParallelRemarkEnabled`
	`-verbose:gc -XX:+PrintGCDetails`
	`-XX:+PrintGCDateStamps`
		`-verbose:gc -Xlog:gc`

• MapReduce JobTracker
• MapReduce TaskTracker
	`-XX:+UseParNewGC`
	`-XX:+UseConcMarkSweepGC`
	`-XX:CMSInitiatingOccupancyFraction=70`
	`-XX:+CMSParallelRemarkEnabled`
		None, G1GC is enabled by default.

• Solr Server	`-XX:+UseParNewGC`
	`-XX:+UseConcMarkSweepGC`
	`-XX:CMSInitiatingOccupancyFraction=70`
	`-XX:+CMSParallelRemarkEnabled`
		None, G1GC is enabled by default.

• YARN JobHistory Server	`-XX:+UseParNewGC`	`-Dlibrary.leveldbjni.path={{`

(optional) Enable high availability for Cloudera Manager

After you have set up a cluster, you can enable high availability for Cloudera Manager. See Configuring Cloudera Manager for High Availability.

(Recommended) Enable Auto-TLS

Auto-TLS greatly simplifies the process of enabling and managing TLS encryption on your cluster.

For information on using Auto-TLS to simplify the process of configuring TLS encryption for Cloudera Manager, see *Configuring TLS Encryption for Cloudera Manager Using Auto-TLS*.

Related Information

Configuring TLS Encryption for Cloudera Manager Using Auto-TLS

(Recommended) Enable Kerberos

Kerberos is an authentication protocol that relies on cryptographic mechanisms to handle interactions between a requesting client and server, greatly reducing the risk of impersonation.

For information on enabling Kerberos, see *Enabling Kerberos Authentication for CDP*.

Note: Authorization through Apache Ranger is just one element of a secure production cluster: Cloudera supports Ranger only when it runs on a cluster where Kerberos is enabled to authenticate users.

Related Information

Enabling Kerberos Authentication for CDP.

Additional Steps for Apache Ranger

After installing Cloudera Manager and adding a cluster, there are additional steps required to complete the installation of Apache Ranger.

Related Information

Configure a resource-based policy: Solr

Enabling Solr clients to authenticate with a secure Solr

Update Ranger audit configration parameters

Enable Plugins

About this task

The Ranger plugins for HDFS and Solr may not be enabled by default. Ranger plugins enable Cloudera Manager stack components – such as HDFS and Solr – to connect to Ranger and access its authorization and audit services. Verify that the HDFS and Solr plugins are enabled after you install and start the Ranger service.

Procedure

1. To enable the HDFS plugin:
    a) Login to Cloudera Manager.
    b) Go to the HDFS Service status page.
    c) Click the Configuration tab.
    d) Search for the Enable Ranger Authorization configuration property.
    e) If the Enable Ranger Authorization property is not selected, select it and save the changes.
    f) Go to the Ranger Service status page and click ActionsSetup Ranger Plugin Service.
    g) Restart the HDFS service.
2. To enable the Ranger Solr plugin:
    a) Login to Cloudera Manager.
    b) Go to the Solr Service status page.
    c) Click the Configuration tab.
    d) Search for the Enable Ranger Authorization configuration property.
    e) If the Enable Ranger Authorization property is not selected, select it and save the changes.

    Note: Do not select the Ranger Service dependency parameter. This is used for enabling a Solr service instance that is not used by the Ranger service.

    f) Restart the Solr service.

Add Solr WebUI Users

Procedure

Add the username of any users to the Ranger Solr policy who should have access to the Solr Web UI in the Ranger Policy for Solr. The user should have full access privileges.

Update the Time-to-live configuration for Ranger Audits

How to change the default time settings that control how long Ranger keeps audit data collected by solr.

Procedure

1. From Cloudera Manager choose Ranger Configuration .
2. In Search, type ranger.audit.solr.config, then press Return.
3. In ranger.audit.solr.config.ttl, set the the number of days to keep audit data.
4. In ranger.audit.solr.config.delete.trigger set the number and units (days, minutes, hours, or seconds) to keep data for expired documents
5. Refresh the configuration, using one of the following two options:
    a) Click Refresh Configuration, as prompted or, if Refresh Configuration does not appear,
    b) In Actions, click Update Solr config-set for Ranger, then confirm.

Installing Apache Knox

This document provides instructions on how to install Apache Knox using the Cloudera Private Cloud Base installation process.

About this task

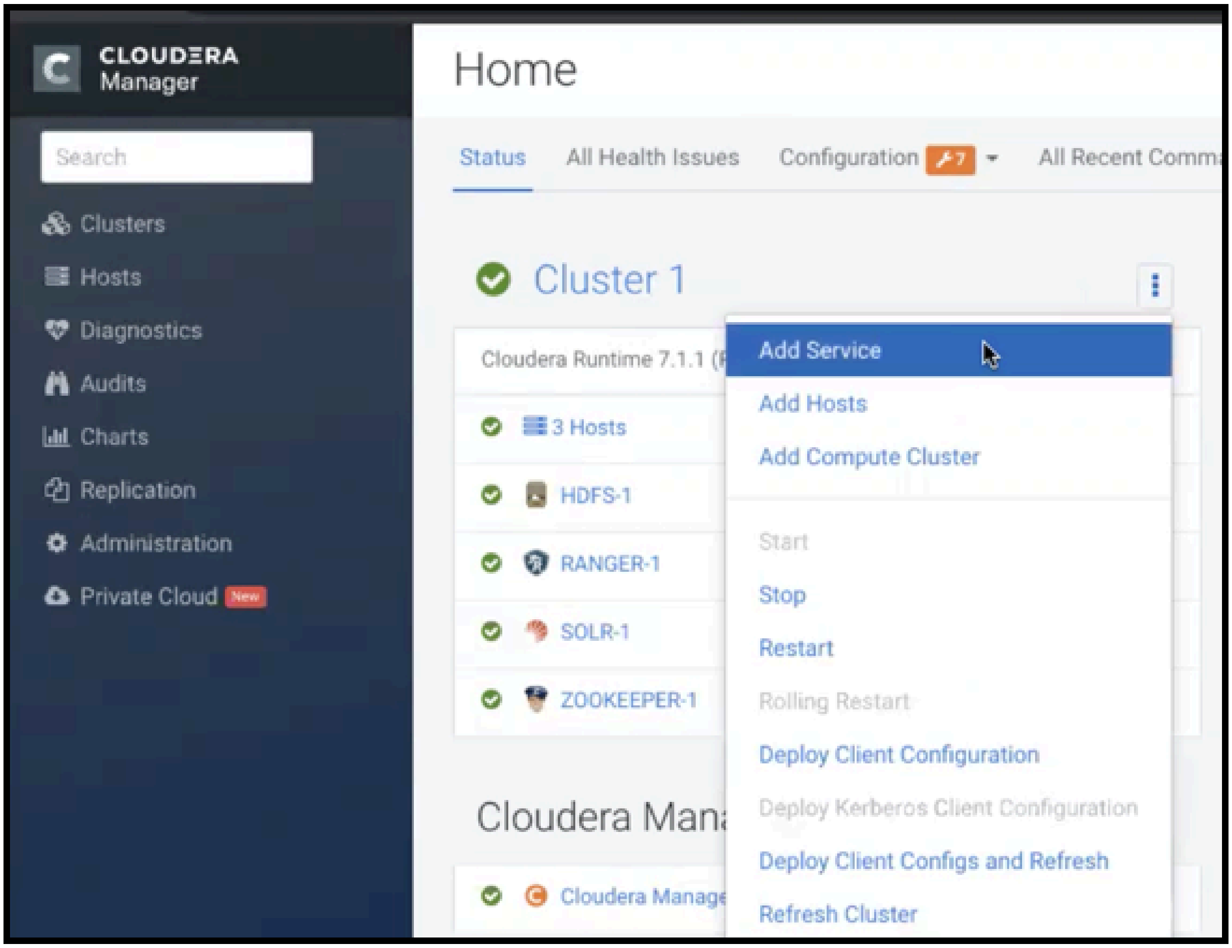
Apache Knox is an application gateway for interacting with the REST APIs and UIs. The Knox Gateway provides a single access point for all REST and HTTP interactions in your Cloudera Data Platform cluster.

Before you begin

When installing Knox, you must have Kerberos enabled on your cluster.

Procedure

1. From your Cloudera Manager homepage, go to Status tab $Cluster Name ... Add Service



2. From the list of services, select Knox and click Continue.
3. On the **Select Dependencies** page, choose the dependencies you want Knox to set up:

HDFS, Ranger, Solr, Zookeeper	For users that require Apache Ranger for authorization. HDFS with Ranger. HDFS depends on Zookeeper, and Ranger depends on Solr.
HDFS, Zookeeper	HDFS depends on Zookeeper.
No optional dependencies	For users that do not wish to have Knox integrate with HDFS or Ranger.

4. On the **Assign Roles** page, select role assignments for your dependencies and click Continue:

Knox service roles	Description	Required?
Knox Gateway	If Knox is installed, at least one instance of this role should be installed. This role represents the Knox Gateway which provides a single access point for all REST and HTTP interactions with Apache Hadoop clusters.	Required

Knox service roles	Description	Required?
KnoxIDBroker*	It is strongly recommended that this role is installed on its own dedicated host. As its name suggests this role will allow you to take advantage of Knox's Identity Broker capabilities, an identity federation solution that exchanges cluster authentication for temporary cloud credentials.*	Optional*
Gateway	This role comes with the CSD framework. The gateway structure is used to describe the client configuration of the service on each host where the gateway role is installed.	Optional

* Note: KnoxIDBroker appears in the Assign Roles page, but it is not currently supported in CDP Private Cloud.

5. On the **Review Changes** page, most of the default values are acceptable, but you must Enable Kerberos Authentication and supply the Knox Master Secret. There are additional parameters you can specify or change, listed in "Knox Install Role Parameters".
   a) Click Enable Kerberos Authentication

      Kerberos is required where Knox is enabled.
   b) Supply the Knox Master Secret, e.g. knoxsecret.
   c) Click Continue.
6. The **Command Details** page shows the status of your operation. After completion, your system admin can view logs for your installation under stdout.

Related Information

Apache Knox Install Role Parameters

Apache Knox Install Role Parameters

Reference information on all the parameters available for Knox service roles.

Service-level parameters

Table 34: Required service-level parameters

Name	In Wizard	Type	Default Value
kerberos.auth.enabled*	Yes	Boolean	false
ranger_knox_plugin_hdfs_audit_directory	No	Text	${ranger_base_audit_url}/knox
autorestart_on_stop	No	Boolean	false
knox_pam_realm_service	No	Text	login
save_alias_command_input_password	No	Text	-

Knox Gateway role parameters

Table 35: Required parameters for Knox Gateway role

Name	In Wizard	Type	Default Value
gateway_master_secret	Yes	Password	-
gateway_conf_dir	Yes	Path	/var/lib/knox/gateway/conf
gateway_data_dir	Yes	Path	/var/lib/knox/gateway/data
gateway_port	No	Port	8443
gateway_path	No	Text	gateway

Name	In Wizard	Type	Default Value
gateway_heap_size	No	Memory	1 GB (min = 256 MB; soft min = 512 MB)
gateway_ranger_knox_plugin_conf_path	No	Path	/var/lib/knox/ranger-knox-plugin
gateway_ranger_knox_plugin_policy_cache_directory	No	Path	/var/lib/ranger/knox/gateway/policy-cache
gateway_ranger_knox_plugin_hdfs_audit_spool_directory	No	Path	/var/log/knox/gateway/audit/hdfs/spool
gateway_ranger_knox_plugin_solr_audit_spool_directory	No	Path	/var/log/knox/gateway/audit/solr/spool

Table 36: Optional parameters for Knox Gateway role

Name	Type	Default Value
gateway_default_topology_name	Text	cdp-proxy
gateway_auto_discovery_enabled	Boolean	true
gateway_cluster_configuration_monitor_interval	Time	60 seconds (minimum = 30 seconds)
gateway_auto_discovery_advanced_configuration_monitor_interval	Time	10 seconds (minimum = 5 seconds)
gateway_cloudera_manager_descriptors_monitor_interval	Time	10 seconds (minimum = 5 seconds)
gateway_auto_discovery_cdp_proxy_enabled_*	Boolean	true
gateway_auto_discovery_cdp_proxy_api_enabled_*	Boolean	true
gateway_descriptor_cdp_proxy	Text Array	Contains the required properties of cdp-proxy topology
gateway_descriptor_cdp_proxy_api	Text Array	Contains the required properties of cdp-proxy-api topology
gateway_sso_authentication_provider	Text Array	Contains the required properties of the authentication provider used by the UIs using the Knox SSO capabilities (Admin UI and Home Page). Defaults to PAM authentication.
gateway_api_authentication_provider	Text Array	Contains the required properties of the authentication provider used by pre-defined topologies such as admin, metadata or cdp-proxy-api. Defaults to PAM authentication.

Related Information

Installing Apache Knox

Setting Up Data at Rest Encryption for HDFS

This section describes how to enable end-to-end data encryption to-and-from HDFS. For optimal performance, High Availability (HA) is also provided.

Important: Before setting up HDFS Data at Rest encryption, Cloudera highly recommends reading the Encrypting Data at Rest content, which provides more information about HDFS encryption, the supported architecture, planning, encryption requirements, and more.

If you require a third-party HSM for key storage, Cloudera also recommends reading the Integrating Components for Encrypting Data at Rest content.

Links are provided in the Related Information section below.

Depending on your encryption key root trustee requirements, you can enable HDFS encryption with one of the following options:

- Ranger Key Management Service backed by Key Trustee Server, which sources the encryption zone keys from a backing Ranger Key Trustee Server and includes HA.
- Ranger Key Management Service backed by Database, which sources the encryption zone keys from a backing Database and includes HA.
- A file-based password protected Java Keystore, which adds the Java KeyStore KMS service to the cluster. The Java KeyStore KMS service uses a password-protected Java KeyStore for cryptographic key management. This option does not include HA.

Warning:

Cloudera strongly recommends NOT using this option for production environments. The file-based Java KeyStore root of trust is insufficient to provide the security, scalability, and manageability required by most production systems. More specifically, the Java KeyStore KMS does not provide:

- Scalability. You are limited to having only one Key Management System (KMS), which can result in bottlenecks.
- High Availability (HA).
- Recoverability. If you lose the node where the Java KeyStore is stored, you can lose access to all the encrypted data

Related Information

Encrypting Data at Rest

Data at Rest Encryption Reference Architecture

Data at Rest Encryption Requirements

Resource Planning for Data at Rest Encryption

Data Encryption Components and Solutions

Working with an HSM for KTS and Key HSM

Working with an HSM for Ranger KMS

Installing Ranger KMS backed by a Database and HA

The tasks and steps for installing the Ranger Key Management System (KMS) with High Availability (HA) service that uses a database as the backing key store.

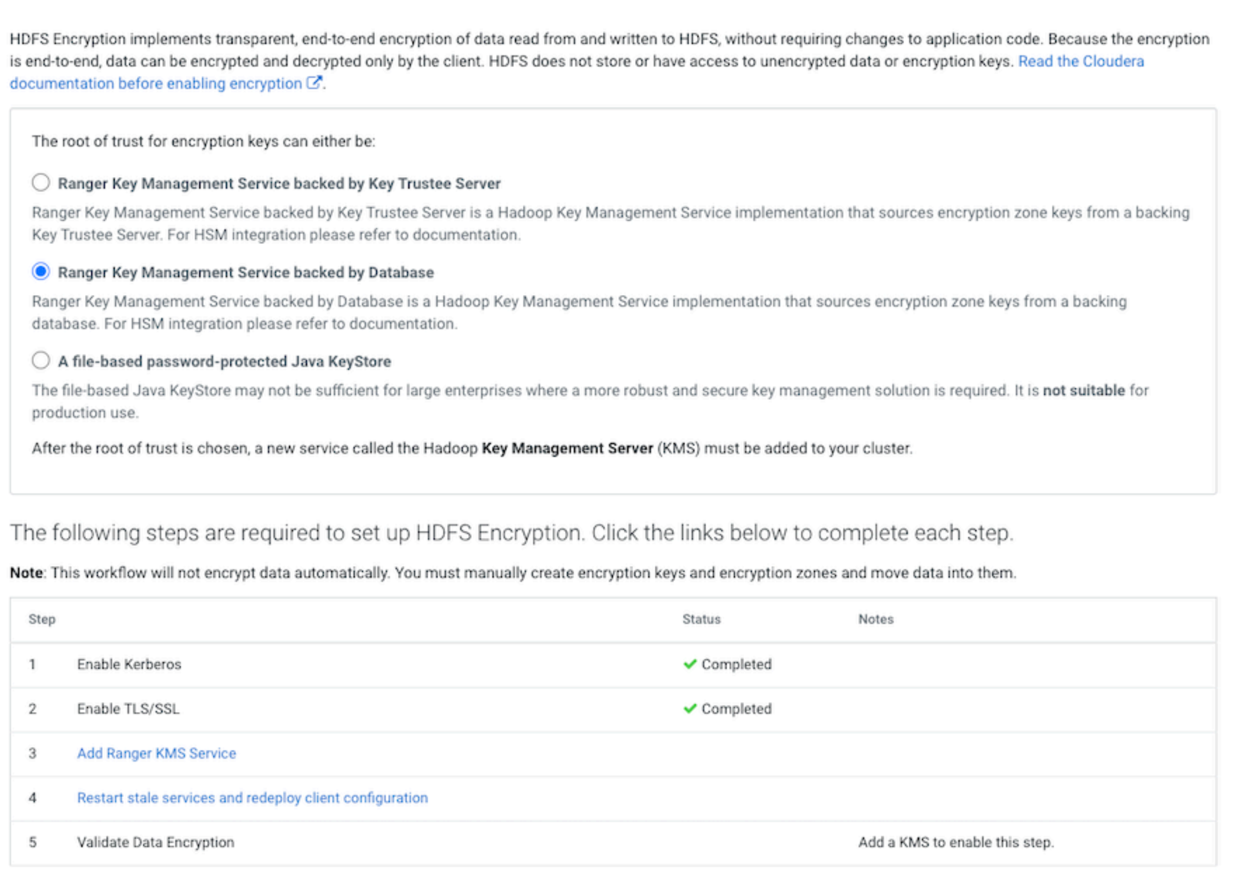
About this task

This task uses the Set up HDFS Data At Rest Encryption wizard to install a Ranger KMS with HA service that uses a database as the backing key store.

The following image shows the Set up HDFS Data At Rest Encryption page. When you select your encryption keys root of trust option, a list of tasks that you must do to enable encryption to-and-from HDFS is displayed.

You complete each task independently from the other tasks. Where, the task’s Status column indicates whether the step has been completed and the Notes column provides additional context for the task. If your Cloudera Manager user account does not have sufficient privileges to complete a task, the Notes column indicates the privileges that are required.

When selected, each task contains links to wizards or documentation that help you complete the task. If a task is unavailable, due to insufficient privileges or an incomplete prerequisite step, no links are present and the Notes column displays the reason.



Note: It is assumed that you have already created a database on a server that does not contain the Cloudera Ranger service.

For more information on how to create a database for Ranger KMS, see the Related Information links below.

The Wizard steps are as follows and must be completed in the order listed:

1. Enable Kerberos

   Note: The instructions assume that you have enabled Kerberos. If this is not the case, click the link associated with the uncompleted task and follow the Wizard's instructions.

2. Enable TLS/SSL

   Note: The instructions assume that you have enabled TLS. If this is not the case, click the link associated with the uncompleted task and follow the Wizard's instructions.

3. Add a Ranger KMS Service
4. Restart the stale services and redeploy the client configuration
5. Validate the Data Encryption

The following lists the post installation tasks for Installing the Ranger KMS backed by a Database and HA:

- Update the Ranger KMS backed by a Database service's URL
- Create a Ranger Audit Directory

Before you begin

Verify the following:

- The cluster in which Cloudera Manager and the Cloudera Ranger service is installed, is up and running.
- A Ranger KMS database has been created as the underlying keyStore mechanism. This database must be separate from the Ranger database.

- Communication through secure connections is enabled with the Transport Layer Security (TLS) protocol and your network authentication is enabled with the Kerberos protocol.
- You have securely recorded the following backing key store database access credentials, as you will be required to supply them during the installation steps:
  - The Database name.
  - The Database hostname.
  - The user name and password that has full administrative privileges to the backing key store database.

Procedure

1. In a supported web browser on the cluster in which the Ranger service is installed, log in to Cloudera Manager as a user with full administrative privileges.
2. From the Cloudera Manager navigation side-bar, select Administration Security .
3. On the Security Status page, click Set up HDFS Data At Rest Encryption.
4. In the Set up HDFS Data At Rest Encryption page, select the Ranger Key Management Service backed by Database option.

   A list of tasks are displayed at the bottom of the page. To successfully set up HDFS Data at Rest encryption, these tasks must be completed.

   Important: Kerberos and TLS must be enabled. If the steps associated with these tasks do not display Completed in the Status column, before continuing, click the link associated with the uncompleted task and follow the Wizard's instructions.
5. To set up HDFS Encryption, follow the instructions as described below for each of the Set up HDFS Data At Rest Encryption Wizard's steps.

Related Information

Configuring a database for Ranger or Ranger KMS

TLS/SSL and Its Use of Certificates

Enabling Kerberos Authentication for CDP

Installing the Ranger KMS Service

The Set up HDFS Data At Rest Encryption wizard's installation step that installs the Ranger Key Management System (KMS) with High Availability (HA) service on your cluster.

About this task

Describes the steps that install the Ranger Key Management System (KMS) service on your cluster and associates it with your backing key store database.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Add Ranger KMS Service.

   The Add Ranger KMS Service to Cluster Wizard opens.
2. In the Assign Roles page, verify that the hostname is the required server on which to install the Ranger KMS service by clicking inside the listed server field. By default, this field is populated by the Wizard.

   The Hosts Selected page opens.
3. In the Hosts Selected page, scroll down and from the Hostname column, locate the hostname that was selected by the Wizard. Notice in the Added Roles column the Ranger KMS Server (RK) role icon. This role is added during the installation.

4. Do one of the following:

   - If the pre-selected host is correct, confirm the Wizard's choice by clicking OK.
   - If the pre-selected host is incorrect, deselect the check box of the Wizard's choice, select the hostname check box of the required server, and then click OK.

     Note: If you require more than one KMS service, select the hostname check box for each server on which to install a Ranger KMS service.

5. Back in the Assign Roles page, click Continue.

   The Setup Database page opens.

6. In the Setup Database page, do the following:

   a. In the Database Hostname field, enter the hostname of the backing key store database.
   b. In the Database Name field, enter the name of the backing key store database.
   c. In the Username field, enter the user name that has full administrative privileges to the backing key store database.
   d. In the Password field, enter the password of the user name that has full administrative privileges to the backing key store database.
   e. (Optional) Verify that the credentials you entered are correct by clicking Test Connection.
   f. Click Continue.

   The Review Changes page opens.

7. In the required Ranger KMS Master Key Password field, enter the password that will be used to encrypt the master key.

   Tip: You can confirm the password's value by clicking the ranger_kms_master_key_password link.

8. Review the rest of the settings before clicking Continue.
9. In the Command Details page, monitor the installation of the Ranger KMS server. When the Status displays Finished the Ranger KMS is installed and tested.

   Tip: If the Ranger KMS start task fails during the First Run Command process, click the Context Ranger KMS link, which opens the Ranger KMS service page. From the Actions list, select Start.

10. Click Continue.

    The Summary page opens.

11. Click Finish, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.
12. Verify that the Ranger KMS service appears in the Cloudera Manager Clusters components list and that the service has been started.

    If the Ranger KMS service was not started by the installation wizard, do the following:

    a. Go to Cloudera Manager's Home page by clicking the Cloudera Manager icon.
    b. In the Cloudera Manager Clusters components list, locate and click Ranger KMS.
    c. From the Actions menu, click Start.

What to do next

Adding Ranger KMS to a cluster triggers additional property updates for other services. Cloudera Manager may flag these with stale configuration warnings. Restart the stale services and redeploy the client configuration.

Restarting the Stale Services and Redeploying the Client Configuration

The Set up HDFS Data At Rest Encryption wizard's step for restarting stale services and redeploying the client configuration.

About this task

Describes the steps that restart stale services after installing the Data-at-Rest HDFS Ranger KMS service option on your cluster.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Restart stale services and redeploy client configuration..

   The Stale Configurations page opens.
2. Click Restart Stale Services.

   The restart Stale Services page opens.
3. Verify that the Re-deploy client configuration check box is selected and click Restart Now.
4. In the Command Details page, monitor the restart process. When the Status displays Finished, click Continue, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.

What to do next

Validate that the Data-at-Rest HDFS Ranger KMS service option can successfully encrypt your data to-and-from HDFS.

Validating Data Encryption to-and-from HDFS

The Set up HDFS Data At Rest Encryption wizard's step for validating the data encryption to-and-from HDFS.

About this task

Describes the steps which verify that the Data-at-Rest HDFS Ranger KMS service option can successfully encrypt your data to-and-from HDFS.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Validate Data Encryption.

   The Validate Data Encryption page opens, which displays a list of commands and instructions for creating an encryption zone and adding data.
2. In a terminal, log in to one of the hosts in your cluster and run each of the following commands:

   a) Create a key and directory by entering the following:

   ```
   kinit KEY_ADMIN_USER
   hadoop key create mykey1
   hdfs dfs -mkdir /tmp/zonel
   ```

   Where, *KEY_ADMIN_USER* is the key administrator whose role can perform the following actions:

   - Configure HDFS encryption, administer Key Trustee Server, and manage encryption keys
   - Start, stop, and restart Ranger KMS
   - Configure Ranger KMS Policies
   - View configuration and monitoring information in Cloudera Manager
   - View service and monitoring information
   - View events and logs
   - View Replication jobs and snapshot policies
   - View YARN applications and Impala queries

   b) Create a zone and link to the key, by entering the following:

   ```
   kinit hdfs hdfs
   hdfs crypto -createZone -keyName mykey1 -path /tmp/zone1
   ```

c) Create a file, put it in your zone, and verify that the file can be decrypted, by entering the following:

```
kinit KEY ADMIN_USER
echo "Hello World" > /tmp/helloWorld.txt
hdfs dfs -put /tmp/helloWorld.txt /tmp/zone1
hdfs dfs -cat /tmp/zonel/helloWorld.txt
rm /tmp/helloWorld.txt
```

d) Verify that the stored file is encrypted, by entering the following:

```
kinit hdfs
hdfs dfs -cat /.reserved/raw/tmp/zonel/helloWorld.txt
hdfs dfs -rm -R / tmp/zone1
hdfs crypto -listZones
```

3. When completed, click Close, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.

Post-Tasks for the Data-at-Rest HDFS Ranger KMS Service

The post-tasks that you must perform after you have set up the Data-at-Rest HDFS Ranger KMS service option.

About this task

Describes the post-task steps.

Depending on which Data-at-Rest HDFS Ranger KMS service option was set up, two or more of the following post-tasks must be completed:

- Update the Data-at-Rest HDFS Ranger KMS service's URL
- Create a Ranger Audit Directory
- (Ranger KMS with Key Trustee Server service only) Update the Authentication Properties and KMS Hadoop cache settings

Procedure

1. Update the Data-at-Rest HDFS Ranger KMS service's URL by doing the following:

   a. In the Cloudera Manager Clusters components list, locate and click the Ranger service.

   b. Log in to the Ranger Web UI as the Ranger KMS user, whose default user name credential is keyadmin and default password is admin123.

   c. In the cm_kms service, click the Edit icon and update the KMS URL field value as follows:

      1. In the KMS URL field, enter the URL value using the following syntax:

      kms://http@*KMS_HOST1*;*KMS_HOST2*:*KMS_PORT*/kms

      Where,

      - *KMS_HOST* is the host where either the Ranger KMS with Key Trustee Server or the Ranger KMS backed by a database is installed.
      - *KMS_PORT* is the port number. By default, this is 9292. For example,

        kms://http@*KMS_HOST1*;*KMS_HOST2*:9292/kms

        Important: If SSL is enabled, use https and port number 9494. For example:

        kms://https@*KMS_HOST1*;*KMS_HOST2*:9494/kms

      2. To confirm your URL setting, click Test Connection.

      3. Click Save.

2. Create a Ranger Audit Directory by doing the following:
   a. Depending on which Data-at-Rest HDFS Ranger KMS service you set up, in the Cloudera Manager Clusters components list, locate and click either the Ranger KMS with Key Trustee Server service or the Ranger KMS service.
   b. From the Actions menu, click Create Ranger Plugin Audit Directory.
   c. When the Create Ranger Plugin Audit Directory message appears, confirm its creation by clicking Create Ranger Plugin Audit Directory.
   d. Monitor the creation process. When the Status displays Finished, click Close.

Installing Ranger KMS backed with a Key Trustee Server and HA

The tasks and steps for installing the Ranger Key Management System (KMS) with High Availability (HA) that uses Key Trustee Server (KTS) as the backing key store.

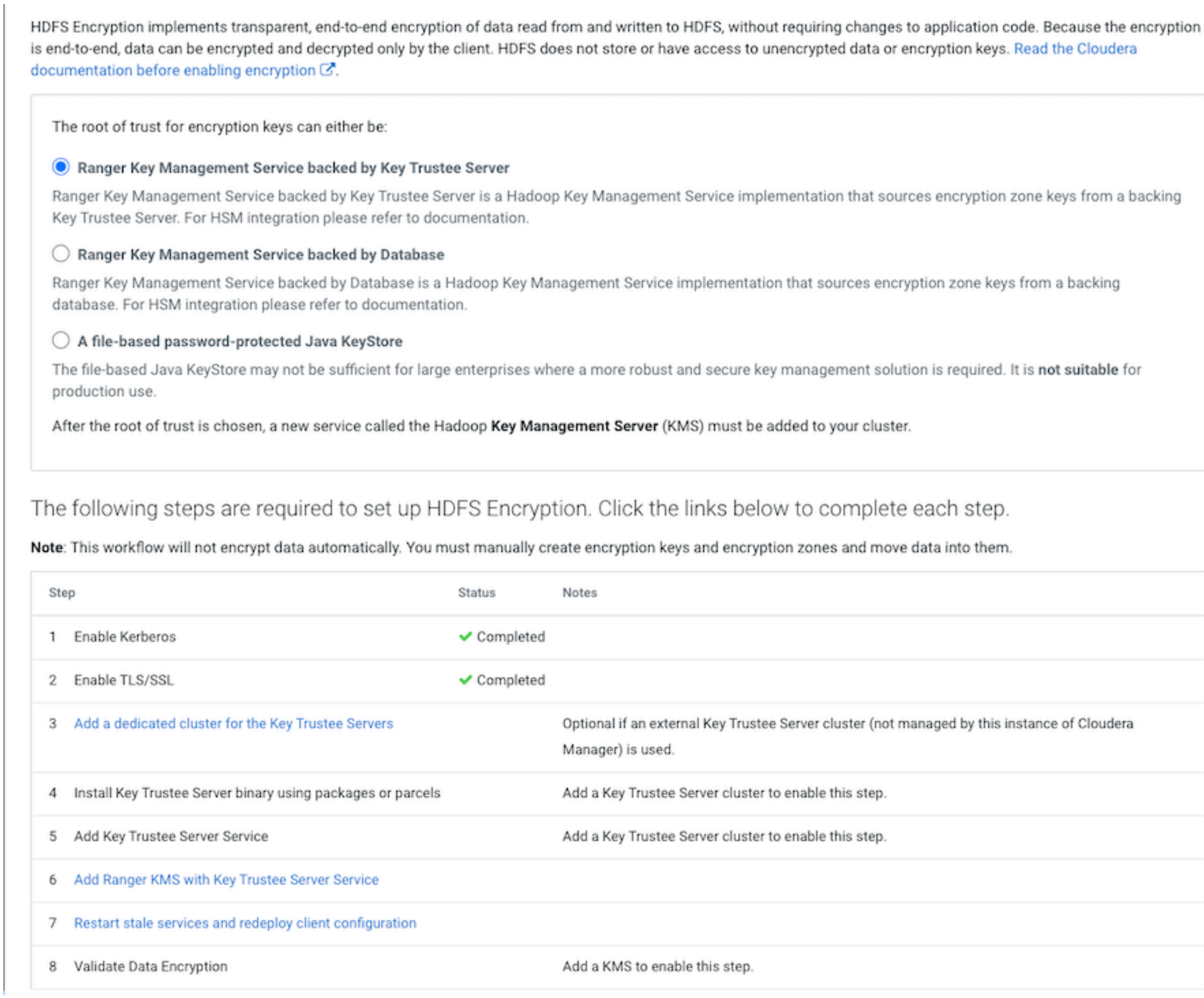
About this task

This task uses the Set up HDFS Data At Rest Encryption wizard to install a Ranger KMS with HA that uses KTS as the backing key store.

The following image shows the Set up HDFS Data At Rest Encryption page. When you select your encryption keys root of trust option, a list of tasks that you must do to enable encryption to-and-from HDFS is displayed.

You complete each task independently from the other tasks. Where, the task's Status column indicates whether the step has been completed and the Notes column provides additional context for the task. If your Cloudera Manager user account does not have sufficient privileges to complete a task, the Notes column indicates the privileges that are required.

When selected, each task contains links to wizards or documentation that help you complete the task. If a task is unavailable, due to insufficient privileges or an incomplete prerequisite step, no links are present and the Notes column displays the reason.



The Wizard steps are as follows and must be completed in the order listed:

1. Enable Kerberos

   Note: The instructions assume that you have enabled Kerberos. If this is not the case, click the link associated with the uncompleted task and follow the Wizard's instructions.

2. Enable TLS/SSL

   Note: The instructions assume that you have enabled TLS. If this is not the case, click the link associated with the uncompleted task and follow the Wizard's instructions.

3. Add a dedicated cluster for the Key Trustee Servers
4. Install the Key Trustee Server binary using packages or parcels

   Note: The instructions assume that you have installed the Key Trustee Server service binary. If this is not the case, click the link associated with the uncompleted task and follow the Wizard's instructions.

5. Add the Key Trustee Service
6. Add the Ranger KMS with Key Trustee Server Service

   Important: The Ranger KMS with Key Trustee Server must not be installed on the host that contains a Key Trustee Server.

7. Restart the stale services and redeploy the client configuration
8. Validate the Data Encryption

The following lists the post installation tasks for Installing the Ranger KMS backed with a Key Trustee Server and HA:

- Update the KMS with Key Trustee Server service's URL
- Create a Ranger Audit Directory
- Update the Ranger KMS with Key Trustee Server configuration settings

Before you begin

Verify the following:

- The cluster in which Cloudera Manager and the Cloudera Ranger service is installed, is up and running.
- The Cloudera Manager host has access to your internal repository hosting the Key Trustee Server (KTS) software.
- Communication through secure connections is enabled with the Transport Layer Security (TLS) protocol and your network authentication is enabled with the Kerberos protocol.

Procedure

1. In a supported web browser on the cluster in which the Ranger service is installed, log in to Cloudera Manager as a user with full administrative privileges.
2. From the Cloudera Manager navigation side-bar, select Administration Security .
3. On the Security Status page, click Set up HDFS Data At Rest Encryption.
4. In the Set up HDFS Data At Rest Encryption page, select the Ranger Key Management Service backed by Key Trustee Server option.

   A list of tasks are displayed at the bottom of the page. To successfully set up HDFS Data at Rest encryption, these tasks must be completed.

   Important: Kerberos and TLS must be enabled. If the steps associated with these tasks do not display Completed in the Status column, before continuing, click the link associated with the uncompleted task and follow the Wizard's instructions.
5. To set up HDFS Encryption, follow the instructions as described below for each of the Set up HDFS Data At Rest Encryption Wizard's steps.

Related Information

TLS/SSL and Its Use of Certificates

Enabling Kerberos Authentication for CDP

Adding an External Dedicated Cluster for the Key Trustee Server Service

The Set up HDFS Data At Rest Encryption wizard's installation step adds a dedicated cluster for the Key Trustee Server (KTS) service.

About this task

Describes the steps that add a dedicated cluster for the Key Trustee Server service, which sets up the Cloudera Manager agent and Key Trustee Server parcel and creates a new cluster specifically for the Key Trustee Server hosts. Isolating the Key Trustee Server host from other services adds another layer of security.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Add a dedicated cluster for the Key Trustee Servers.

   The Add a dedicated cluster for the Key Trustee Servers Wizard opens.
2. In the Getting Started page, verify that the Enable High Availability check box is selected and then click Continue.

   The Specify Host page opens.
3. In the Hostname field of the Specify Host page, enter the fully qualified domain name (FQDN) of the host on which the Key Trustee Server is to be installed.

4. Click Search.
5. Depending on your Key Trustee Server requirements, select one or multiple host check boxes and then click Continue.

   The Select Repository page opens.
6. In the Select Repository page, select the required repository option and in the text field, enter the full path to its location.
7. Click Continue.

   The Select JDK page opens.
8. In the Select JDK page, select the required JDK option and then click Continue.

   The Enter Login Credentials page opens.
9. In the Enter Login Credentials page, select a secure user option and an authentication method. In the Password field, enter the secure user's password and then click Continue.

   The Install Agents page and opens.
10. Monitor the installation of the Agents and Parcels and when completed successfully, click Continue.

    The Summary page opens.
11. In the Summary page, click Finish, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.

What to do next

Follow the steps to add the Key Trustee Server Service.

> **Note:** The Installing Ranger KMS backed with a Key Trustee Server and HA instructions assume that you have installed the Key Trustee Server service binary. If this is not the case, before adding the Key Trustee Server service, click the link associated with the uncompleted task and follow the Wizard's instructions.

Installing the Key Trustee Server to the Dedicated Cluster

The Set up HDFS Data At Rest Encryption wizard's installation step adds the Key Trustee Server service to dedicated cluster created in the previous step.

About this task

Describes the steps that add the Key Trustee Server service, which enables you to select each Active and Passive Key Trustee Servers for HA, synchronizes the server's Private Keys, and starts them.

The Active Key Trustee Server host is the primary server and the Passive Key Trustee Server host is the backup server that takes over when the primary server disconnects or fails. The primary and backup combination provides a highly-available and continuous operation.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Add Key Trustee Server Service.

   The Add Key Trustee Server Service to Key Trustee Server Cluster Wizard opens.

   > **Important:** For increased security and performance, Cloudera recommends that each selected Key Trustee Server host is not used by any other services.
2. In the Getting Started page, verify that you understand that the Key Trustee Server service is not added to a cluster with existing services by selecting the I understand the risks. Let me proceed. check box and then click Continue.

   The Assign Roles page opens.
3. In the Assign Roles page, verify that the hostname is the required server for the Active Key Trustee Server role by clicking inside the Active Key Trustee Server field. By default,this field is populated with the Active Key Trustee Server name.

   The Hosts Selected page opens.

4. In the Hosts Selected page, scroll down and from the Hostname column, locate the Active hostname that was selected by the Wizard. Notice in the Added Roles column the Key Trustee Server Active Key Trustee Server (AK...) role icon. This role is added during the installation.
5. Do one of the following:
   - If the pre-selected host is correct, confirm the Wizard's choice by clicking OK.
   - If the pre-selected host is incorrect, deselect the check box of the Wizard's choice, select the hostname check box of the required Active server, and then click OK.

   Note: You can also specify hostnames for an external Active and Passive Key Trustee Server service as long as the Key Trustee Server services are not shared across multiple CDP clusters.
6. Back in the Assign Roles page, click inside the Passive Key Trustee Server field.

   The Hosts Selected page opens.
7. In the Hosts Selected page, scroll down and from the Hostname column, locate and select the required Passive hostname check box. Notice in the Added Roles column the Key Trustee Server Passive Key Trustee Server (PK...) role icon. This role is added during the installation.
8. Click OK, which takes you back to the Assign Roles page.
9. (Optional) If you require multiple Key Trustee Server services, select the Active and Passive hostname check box for each server where a Key Trustee Server service is to be installed.
10. Click Continue.

    The Setup Entropy page opens, which displays a list of commands that determine if the available entropy on the Key Trustee Server service is low and provides instructions and commands for installing an entropy generator that increases the entropy for cryptographic operations.
11. In a terminal, determine the amount of available entropy on your target machines, by entering the following:

```
ssh root@ACTIVE_FQDN
cat /proc/sys/kernel/random/entropy_avail
```

    Where, *ACTIVE_FQDN* is the fully qualified domain name of the Active host.

    If the result is below 500, consider installing an entropy generator, such as the rng-tools utilities. Before proceeding, consult the security policies, procedures, and practices in your organization.
12. If you require the rng-tools utilities, do the following:

    a) To install the rng-tools utility, in a terminal, enter one of the following:
    - For Centos/RHEL 6, 7+ systems, enter:

```
yum install rng-tools
```

    - For Debian systems, enter:

```
apt-qet install rng-tools
```

    - For SLES systems, enter:

```
zypper install rng-tools
```

    b) Enable the rng-tools utility, by entering one of the following:
    - For Centos/RHEL 6, Debian and SLES systems, enter:

```
echo 'EXTRAOPTIONS="-r /dev/urandom"' >> /etc/sysconfig/rngd
service rngd start
chkconfig rngd on
cat /proc/sys/kernel/random/entropy_avail
```

- For For Centos/RHEL 7+ systems, enter:

```
cat /proc/sys/kernel/random/entropy_avail
cp /usr/lib/systemd/system/rngd.service/etc/systemd/system/
sed -i -e 's/ExecStart=\/sbin\/rngd -f/ExecStart=\/sbin\/rngd -f -r /
dev\/urandom/' /etc/systemd/system/rngd.service
systemctl daemon-reload
systemctl start rngd
systemctl status rngd
# if the status command returns the service is loaded and enabled,
 skip the following step
systemctl enable rngd
```

13. When completed, click Continue.

The Synchronize Active and Passive Key Trustee Server Private Keys page opens, which displays a list of instructions and commands for initializing and generating a private key that will be used by both the Active and Passive Key Trustee Server service.

Note: For production environments, Cloudera recommends transferring the private key using offline media, such as a removable USB drive. For environments where maximum security is not required, such as testing or development environments, you can copy the private key over the network using the rsync command.

14. In a terminal, initialize the Active Key Trustee Server and generate the private key, by entering the following commands:

```
ssh root@ACTIVE_FQDN
ktadmin init
```

Where, *ACTIVE_FQDN* is the fully qualified domain name of the Active host.

15. Copy the Active Key Trustee Server private key to the Passive Key Trustee Server by doing one of the following:

- For production environments, Cloudera recommends transferring the private key using offline media, such as a removable USB drive.
- For environments where maximum security is not required, such as testing or development, you can copy the private key over the network using the rsync command:

```
rsync -zav --exclude .ssl /var/lib/keytrustee/.keytrustee PASSIVE_FQDN:/
var/lib/keytrustee/
```

16. In a terminal, initialize the Passive Key Trustee Server with the private key generated previously, by entering the following:

```
ssh root@PASSIVE_FQDN
ktadmin init
```

17. Verify that the Active and Passive ktadmin commands output the same initialized directory.

18. When completed, select the I have synchronized the private keys check box and click Continue.

The Setup TLS for Key Trustee Server page opens, which displays a list of instructions for generating CA-signed certificates.

19. Follow the instructions and perform the required steps. When completed, click Continue.

The Review Changes page opens.

20. Review the settings and make any required changes before clicking Continue.

Table 37: TLS/SSL Settings

Property	Default value	Description
Database Storage Directory db_root	/var/lib/keytrustee/db	The directory on the local filesystem where the Key Trustee Server database is stored.
Active Key Trustee Server TLS/SSL Server Private Key File (PEM Format) sll.privatekey.location	/var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keytrustee-pk.pem	The path to the Active Key Trustee Server TLS certificate's private key, which must be in the PEM format. • To use the Cloudera auto-generated private key, do nothing. • To use your company's Certificate Authority (CA) signed certificate, enter the path to its location.
Active Key Trustee Server TLS/SSL Server Certificate File (PEM Format) ssl.cert.location	/var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keytrustee.pem	The path to the Active Key Trustee Server TLS certificate, which must be in the PEM format. • To use the Cloudera auto-generated private key, do nothing. • To use your company's CA-signed certificate, enter the path to its location.
Active Key Trustee Server TLS/SSL Server CA Certificate (PEM Format) ssl.cacert.location	none	The path to the file that contains the CA certificate, if applicable, its intermediate certificates, and the SSL/TLS Certificate, which are used to sign the Active Key Trustee Server certificate and enable the receiver to verify that the sender and all CA's are trustworthy. The file must be in the PEM format. Enter path to the location of the CA certificate chain.
Active Key Trustee Server TLS/SSL Private Key Password ssl.privatekey.password	none	The password for the Active Key Trustee Server private key's file. If the file is not password-protected, leave this field blank.
Passive Key Trustee Server TLS/SSL Server Private Key File (PEM Format) ssl.privatekey.location	/var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keytrustee-pk.pem	The path to the Passive Key Trustee Server TLS certificate's private key, which must be in the PEM format. • To use the Cloudera auto-generated private key, do nothing. • To use your company's CA-signed certificate, enter the path to its location.
Passive Key Trustee Server TLS/SSL Server Certificate File (PEM Format) ssl.cert.location	/var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keytrustee.pem	The path to the Passive Key Trustee Server TLS certificate, which must be in the PEM format. • To use the Cloudera auto-generated private key, do nothing. • To use your company's CA-signed certificate, enter the path to its location.
Passive Key Trustee Server TLS/SSL Server CA Certificate (PEM Format) ssl.cacert.location	none	The path to the file that contains the CA certificate, if applicable, its intermediate certificates, and the SSL/TLS Certificate, which are used to sign the Passive Key Trustee Server certificate and enable the receiver to verify that the sender and all CA's are trustworthy. The file must be in the PEM format. Enter path to the location of the CA certificate chain.

Property	Default value	Description
Passive Key Trustee Server TLS/SSL Private Key Password ssl.privatekey.password	none	The password for the Passive Key Trustee Server private key's file. If the file is not password-protected, leave this field blank.

21. In the Command Details page, monitor the installation of the Key Trustee Server service. When the Status displays Finished the Key Trustee Server service is installed and tested on the dedicated cluster created in the previous step.
22. Click Continue.

    The Summary page opens.
23. Click Finish, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.

What to do next

Follow the steps to add the Ranger KMS with Key Trustee Server service.

Installing the Ranger KMS with Key Trustee Server Service

The Set up HDFS Data At Rest Encryption wizard's installation step adds the Ranger KMS with Key Trustee Server service.

About this task

Describes the steps that add the Ranger KMS with KTS service, which enables the HDFS encryption to use the Key Trustee Server for cryptographic key management.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Add Ranger KMS with Key Trustee Server Service.

   The Add Ranger KMS with Key Trustee Server Service Wizard opens.
2. In the Getting Started page, verify that the hostnames are the required Key Trustee Server service's Active and Passive hosts that you previously set up. By default, the Key Trustee Server fields are populated by the Wizard.

   If you have an existing Key Trustee Server pair outside of the Cloudera Manager's control, in the External Key Trustee Server field, enter the fully-qualified domain names (FQDNs) of the Active and Passive hosts.
3. Click Continue.

   The Assign Roles page opens.
4. In the Assign Roles page, verify that the hostname is the required server for the Ranger KMS with KTS service role by clicking inside the Ranger KMS Server with KTS field. By default, this field is populated by the Wizard.

   The Hosts Selected page opens.
5. In the Hosts Selected page, scroll down and from the Hostname column, locate the hostname that was selected by the Wizard. Notice in the Added Roles column, the Ranger KMS with Key Trustee Server Ranger KMS Server with KTS (RK...) role icon. This role is added during the installation.
6. Do one of the following:

   - If the pre-selected host is correct, confirm the Wizard's choice by clicking OK.
   - If the pre-selected host is incorrect, deselect the check box of the Wizard's choice, select the hostname check box of the required server, and then click OK.

   Note: For production environments, Cloudera recommends installing a Ranger KMS with KTS service on at least two dedicated hosts for high availability. These hosts should not run other CDP cluster services. If you proceed with installing Ranger KMS with KTS on only one host, you can enable high availability later.

7. Back in the Assign Roles page, click Continue.

   The Setup Entropy page opens, which displays a list of commands that determine if the available entropy on the Key Trustee Server service is low and provides instructions and commands for installing an entropy generator that increases the entropy for cryptographic operations.
8. In a terminal, determine the amount of available entropy on your target machines, by entering the following:

   ```
   ssh root@ACTIVE_FQDN
   cat /proc/sys/kernel/random/entropy_avail
   ```

   Where, *ACTIVE_FQDN* is the fully qualified domain name of the Active host.

   If the result is below 500, consider installing an entropy generator, such as the rng-tools utilities. Before proceeding, consult the security policies, procedures, and practices in your organization.
9. If you require the rng-tools utilities, do the following:

   a) To install the rng-tools utility, in a terminal, enter one of the following:

   - For Centos/RHEL 6, 7+ systems, enter:

     ```
     yum install rng-tools
     ```
   - For Debian systems, enter:

     ```
     apt-qet install rng-tools
     ```
   - For SLES systems, enter:

     ```
     zypper install rng-tools
     ```

   b) Enable the rng-tools utility, by entering one of the following:

   - For Centos/RHEL 6, Debian and SLES systems, enter:

     ```
     echo 'EXTRAOPTIONS="-r /dev/urandom"' >> /etc/sysconfig/rngd
     service rngd start
     chkconfig rngd on
     cat /proc/sys/kernel/random/entropy_avail
     ```
   - For For Centos/RHEL 7+ systems, enter:

     ```
     cat /proc/sys/kernel/random/entropy_avail
     cp /usr/lib/systemd/system/rngd.service/etc/systemd/system/
     sed -i -e 's/ExecStart=\/sbin\/rngd -f/ExecStart=\/sbin\/rngd -f -r /
     dev\/urandom/' /etc/systemd/system/rngd.service
     systemctl daemon-reload
     systemctl start rngd
     systemctl status rngd
     # if the status command returns the service is loaded and enabled,
      skip the following step
     systemctl enable rngd
     ```
10. When completed, click Continue.

    The Setup Authorization Secret page opens, which displays a list of instructions and commands for naming an organization and retrieving the secret authentication code that is required to register with the Key Trustee Server.
11. In the Org Name field, enter a name for the organization and then click Generate Instruction.

    A list of commands are generated and displayed.
12. Open a terminal and run the displayed commands.
13. From the terminal output, copy the auth_secret value into the displayed text field and click Continue.

    The Setup TLS for Ranger KMS with Key Trustee Server page opens, which provides high-level instructions for where TLS communication must be enabled.

14. Read and take note of the provided information and then click Continue.

    The Review Changes page opens.
15. Review the settings and make any required changes before clicking Continue.
16. In the Command Details page, monitor the installation of the Ranger KMS with KTS service. When the Status displays Finished the Ranger KMS with KTS service is installed and tested.
17. Click Continue.

    The Synchronized Private Keys and HDFS Dependency page opens, which provides instructions for copying the private key from one Key Management Server Proxy role to all other roles.

    Note: For production environments, Cloudera recommends transferring the private key using offline media, such as a removable USB drive. For environments where maximum security is not required, such as testing or development environments, you can copy the private key over the network using the rsync command.
18. Follow the instructions and perform the required steps. When completed, select the I have synchronized the private keys check box and click Continue.

    The Summary page opens.
19. Click Finish, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.
20. (Optional) Verify that the Ranger KMS with Key Trustee Server service appears in the Cloudera Manager Clusters components list and that the service has been started.

    If the Ranger KMS with Key Trustee Server service was not started by the installation wizard, do the following:

    a. Go to Cloudera Manager's Home page by clicking the Cloudera Manager icon.
    b. In the Cloudera Manager Clusters components list, locate and click Ranger KMS with Key Trustee Server.
    c. From the Actions menu, click Start.

What to do next

Adding Ranger KMS to a cluster triggers additional property updates for other services. Cloudera Manager may flag these with stale configuration warnings. Restart the stale services and redeploy the client configuration.

Restarting the Stale Services and Redeploying the Client Configuration

The Set up HDFS Data At Rest Encryption wizard's step for restarting stale services and redeploying the client configuration.

About this task

Describes the steps that restart stale services after installing the Data-at-Rest HDFS Ranger KMS service option on your cluster.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Restart stale services and redeploy client configuration..

   The Stale Configurations page opens.
2. Click Restart Stale Services.

   The restart Stale Services page opens.
3. Verify that the Re-deploy client configuration check box is selected and click Restart Now.
4. In the Command Details page, monitor the restart process. When the Status displays Finished, click Continue, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.

What to do next

Validate that the Data-at-Rest HDFS Ranger KMS service option can successfully encrypt your data to-and-from HDFS.

Validating Data Encryption to-and-from HDFS

The Set up HDFS Data At Rest Encryption wizard's step for validating the data encryption to-and-from HDFS.

About this task

Describes the steps which verify that the Data-at-Rest HDFS Ranger KMS service option can successfully encrypt your data to-and-from HDFS.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Validate Data Encryption.

   The Validate Data Encryption page opens, which displays a list of commands and instructions for creating an encryption zone and adding data.
2. In a terminal, log in to one of the hosts in your cluster and run each of the following commands:
   a) Create a key and directory by entering the following:

      ```
      kinit KEY_ADMIN_USER
      hadoop key create mykey1
      hdfs dfs -mkdir /tmp/zonel
      ```

      Where, *KEY_ADMIN_USER* is the key administrator whose role can perform the following actions:

      - Configure HDFS encryption, administer Key Trustee Server, and manage encryption keys
      - Start, stop, and restart Ranger KMS
      - Configure Ranger KMS Policies
      - View configuration and monitoring information in Cloudera Manager
      - View service and monitoring information
      - View events and logs
      - View Replication jobs and snapshot policies
      - View YARN applications and Impala queries
   b) Create a zone and link to the key, by entering the following:

      ```
      kinit hdfs hdfs
      hdfs crypto -createZone -keyName mykey1 -path /tmp/zone1
      ```
   c) Create a file, put it in your zone, and verify that the file can be decrypted, by entering the following:

      ```
      kinit KEY ADMIN_USER
      echo "Hello World" > /tmp/helloWorld.txt
      hdfs dfs -put /tmp/helloWorld.txt /tmp/zone1
      hdfs dfs -cat /tmp/zonel/helloWorld.txt
      rm /tmp/helloWorld.txt
      ```
   d) Verify that the stored file is encrypted, by entering the following:

      ```
      kinit hdfs
      hdfs dfs -cat /.reserved/raw/tmp/zonel/helloWorld.txt
      hdfs dfs -rm -R / tmp/zone1
      hdfs crypto -listZones
      ```
3. When completed, click Close, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.

Post-Tasks for the Data-at-Rest HDFS Ranger KMS Service

The post-tasks that you must perform after you have set up the Data-at-Rest HDFS Ranger KMS service option.

About this task

Describes the post-task steps.

Depending on which Data-at-Rest HDFS Ranger KMS service option was set up, two or more of the following post-tasks must be completed:

- Update the Data-at-Rest HDFS Ranger KMS service's URL
- Create a Ranger Audit Directory
- (Ranger KMS with Key Trustee Server service only) Update the Authentication Properties and KMS Hadoop cache settings

Procedure

1. Update the Data-at-Rest HDFS Ranger KMS service's URL by doing the following:
   a. In the Cloudera Manager Clusters components list, locate and click the Ranger service.
   b. Log in to the Ranger Web UI as the Ranger KMS user, whose default user name credential is keyadmin and default password is admin123.
   c. In the cm_kms service, click the Edit icon and update the KMS URL field value as follows:
      1. In the KMS URL field, enter the URL value using the following syntax:

         kms://http@*KMS_HOST1*;*KMS_HOST2*:*KMS_PORT*/kms

         Where,

         - *KMS_HOST* is the host where either the Ranger KMS with Key Trustee Server or the Ranger KMS backed by a database is installed.
         - *KMS_PORT* is the port number. By default, this is 9292. For example,

           kms://http@*KMS_HOST1*;*KMS_HOST2*:9292/kms

           Important: If SSL is enabled, use https and port number 9494. For example:

           kms://https@*KMS_HOST1*;*KMS_HOST2*:9494/kms
      2. To confirm your URL setting, click Test Connection.
      3. Click Save.
2. Create a Ranger Audit Directory by doing the following:
   a. Depending on which Data-at-Rest HDFS Ranger KMS service you set up, in the Cloudera Manager Clusters components list, locate and click either the Ranger KMS with Key Trustee Server service or the Ranger KMS service.
   b. From the Actions menu, click Create Ranger Plugin Audit Directory.
   c. When the Create Ranger Plugin Audit Directory message appears, confirm its creation by clicking Create Ranger Plugin Audit Directory.
   d. Monitor the creation process. When the Status displays Finished, click Close.

Updating the Ranger KMS with KTS Service Configuration Properties

Describes the authentication and other property settings that you update after you have set up the Ranger KMS backed with a Key Trustee Server and HA service.

About this task

Describes the steps that update the Ranger KMS with Key Trustee Server service's authentication and KMS Hadoop cache settings.

Procedure

1. In the Cloudera Manager Clusters components list, locate and click the Ranger KMS with Key Trustee Server service.

2. From the Ranger KMS with Key Trustee Server services page, select the Configuration tab.
3. In the Search field, enter Ranger KMS Server with KTS Advanced Configuration Snippet (Safety Valve) for conf/kms-site.xml.
4. Create a new property value by clicking the Add (+) icon and then depending on your requirements, do one or more of the following:

- To override the ZooKeeper connection string's default value:

  a. In the Name field, enter hadoop.kms.authentication.zk-dt-secret-manager.zkConnectionString
  b. In the Value field, enter *ZOOKEEPER_HOSTNAME*:2181

     Note: In a cluster with multiple ZooKeeper hosts, enter each hostname separated by a comma. For example:

     ZOOKEEPER_HOSTNAME1:2181,*ZOOKEEPER_HOSTNAME2*:2181,....
  c. Click Save Changes.
- To override the ZooKeeper path's default value:

  a. Create a new property value by clicking the Add (+) icon.
  b. In the Name field, enter hadoop.kms.authentication.zk-dt-secret-manager.znodeWorkingPath
  c. In the Value field, enter the *ZNODE_WORKING PATH*. To avoid collision do not use /zkdtsm.

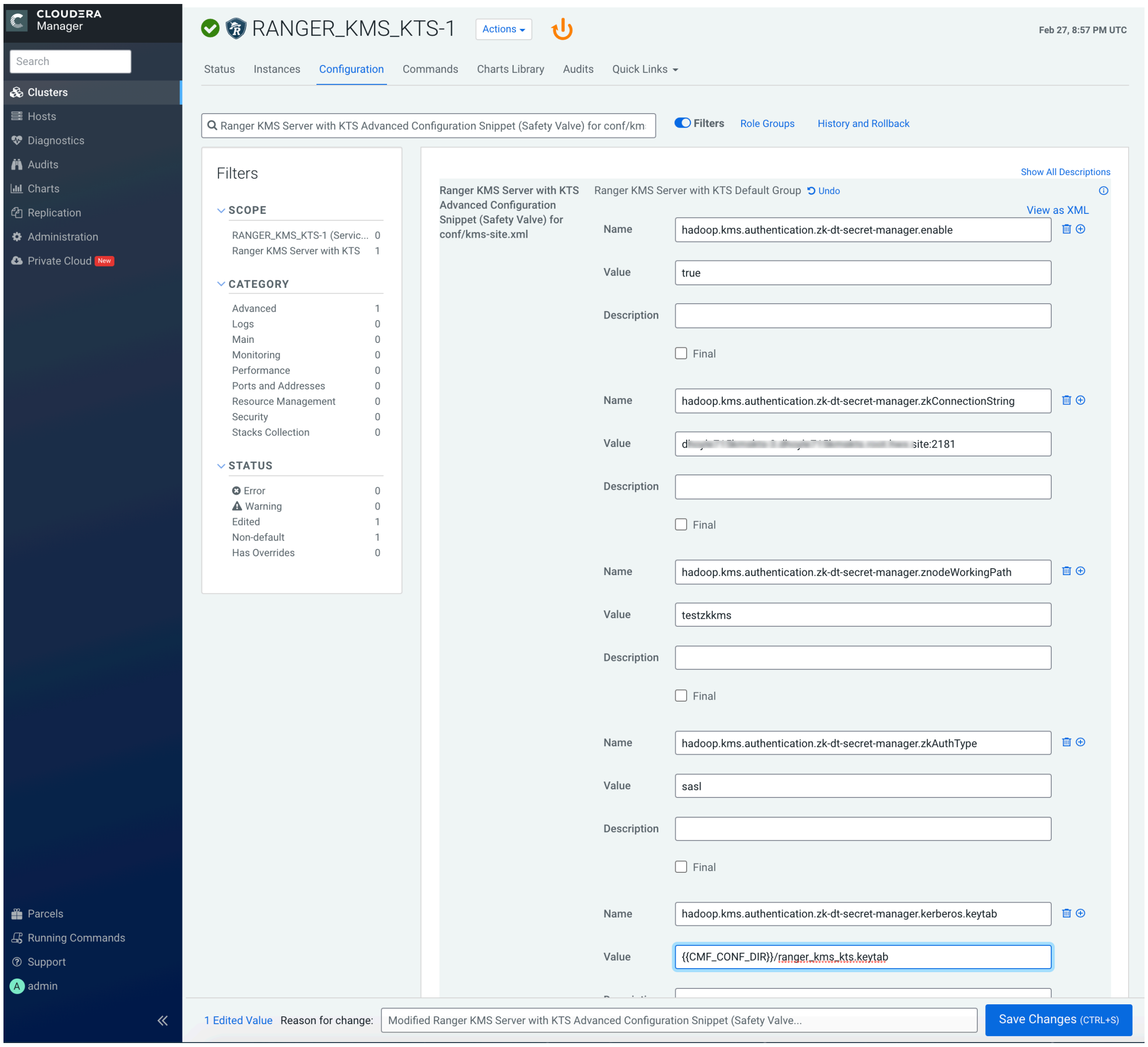
     Example: hadoop.kms.authentication.zk-dt-secret-manager.znodeWorkingPath = testzkkms

     Note: The znode working path must not contain a leading slash.
  d. Click Save Changes.
- To override the ZooKeeper authentication type:

  a. Create a new property value by clicking the Add (+) icon.
  b. In the Name field, enter hadoop.kms.authentication.zk-dt-secret-manager.zkAuthType
  c. In the Value field, enter sasl.

     By default the value is *NONE*, which gives all the permissions to the default node created in ZooKeeper. Cloudera recommends using the Simple Authentication and Security Layer (SASL) protocol.
  d. If you set SASL as the hadoop ZooKeeper authentication value then you must also set the Kerberos authentication by creating a new property value. In the Name field, enter Kerberos

hadoop.kms.authentication.zk-dt-secret-manager.kerberos.keytab and in the Value field, enter {{CMF_CONF_DIR}}/ranger_kms_kts.keytab.

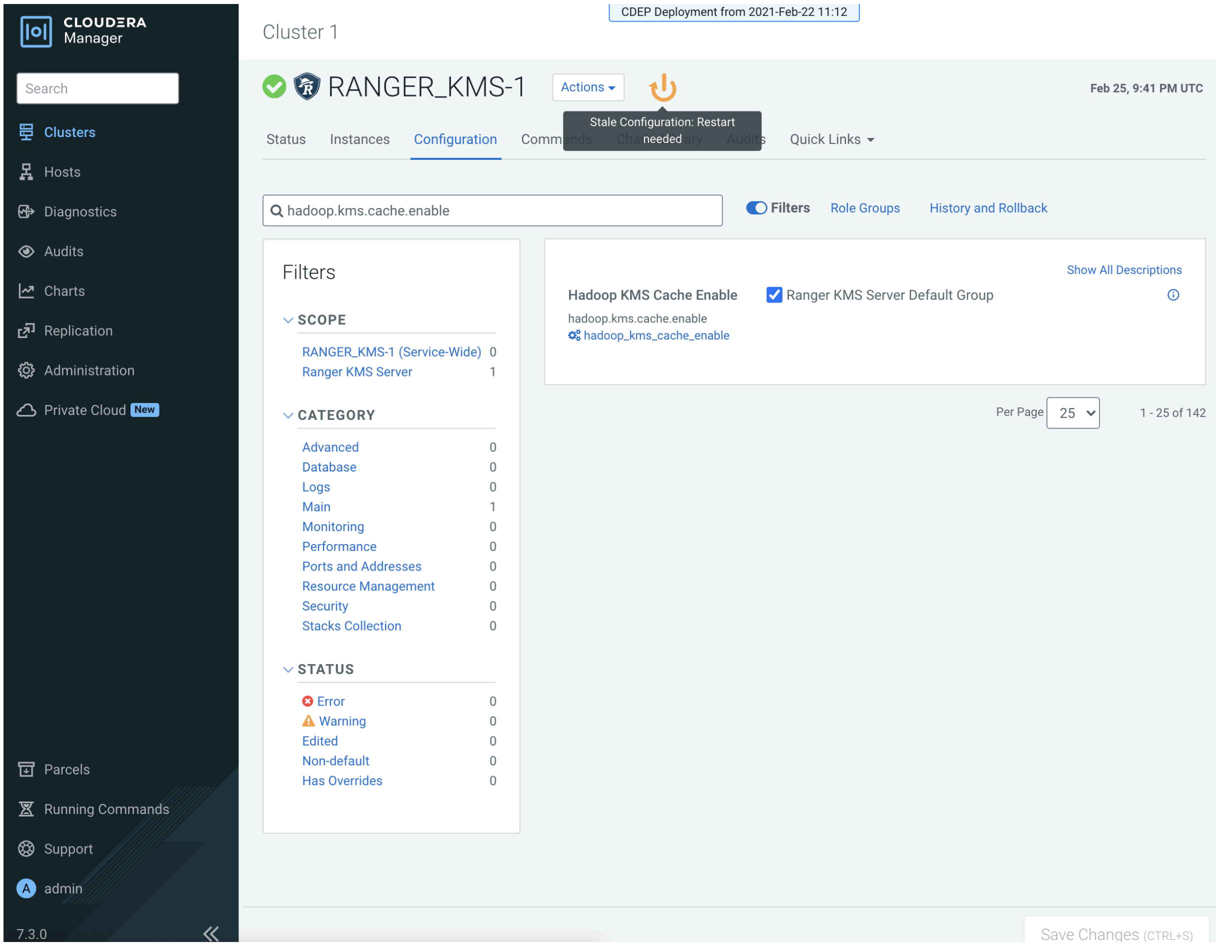


e. Click Save Changes.

5. In the Search field, enter Hadoop KMS Authentication Signer Secret Provider Zookeeper Auth Type and in the property select the sasl option.

6. Click Save Changes.

7. In the Search field, enter Hadoop KMS Cache Enable and verify that the Ranger KMS Server Default Group check box is selected.



8. Click the Stale Configuration Restart icon and then on the Stale Configurations page, click Restart Stale Services.
9. On the Restart Stale Services page, verify that the Re-deploy client configuration check box is selected and then click Restart Now.
10. In the Command Details page, monitor the restart process. When the Status displays Finished, click Finish.

Installing a Java Keystore KMS

The tasks and steps for installing a Java KeyStore Ranger Key Management System (KMS) service to the cluster for your HDFS data-at-rest encryption.

About this task

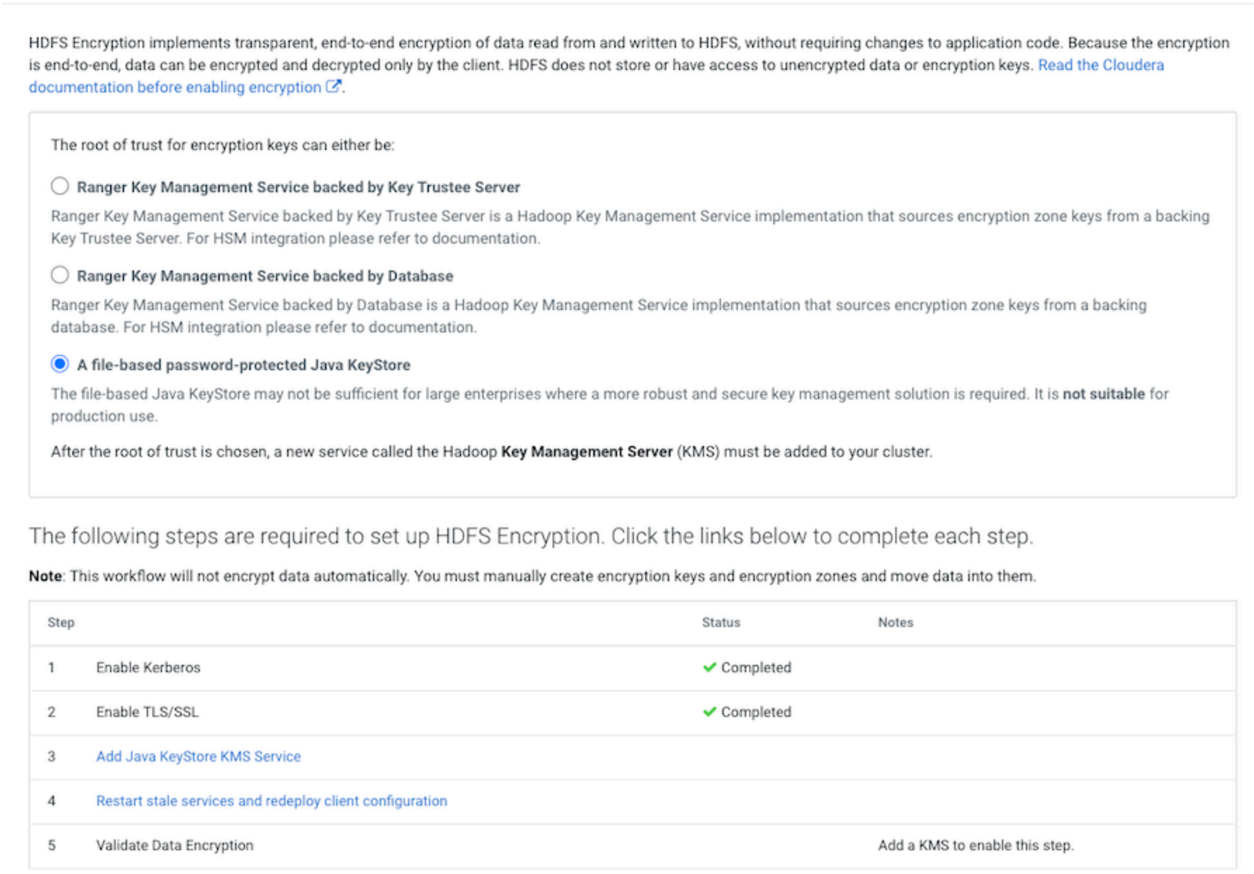
Describes how to install a file-based password-protected Java KeyStore KMS service to your cluster. The Java KeyStore KMS service uses a password-protected Java KeyStore for cryptographic key management. This option does not include HA.

Warning: The file-based Java KeyStore root of trust is insufficient to provide the security, scalability, and manageability required by most production systems. Therefore, Cloudera strongly recommends using the Cloudera Ranger KMS backed with a Key Trustee Server and HA option as the root of trust for production environments.

The following image shows the Set up HDFS Data At Rest Encryption page. When you select your encryption keys root of trust option, a list of tasks that you must do to enable encryption to-and-from HDFS is displayed.

You complete each task independently from the other tasks. Where, the task's Status column indicates whether the step has been completed and the Notes column provides additional context for the task. If your Cloudera Manager user account does not have sufficient privileges to complete a task, the Notes column indicates the privileges that are required.

When selected, each task contains links to wizards or documentation that help you complete the task. If a task is unavailable, due to insufficient privileges or an incomplete prerequisite step, no links are present and the Notes column displays the reason.



The Wizard steps are as follows and must be completed in the order listed:

1. Enable Kerberos

   Note: The instructions assume that you have enabled Kerberos. If this is not the case, click the link associated with the uncompleted task and follow the Wizard's instructions.

2. Enable TLS/SSL

   Note: The instructions assume that you have enabled TLS. If this is not the case, click the link associated with the uncompleted task and follow the Wizard's instructions.

3. Add the Java KeyStore KMS Service
4. Restart the stale services and redeploy the client configuration
5. Validate the Data Encryption

Before you begin

Verify the following:

- The cluster in which Cloudera Manager and the Cloudera Ranger service is installed, is up and running.
- Communication through secure connections is enabled with the Transport Layer Security (TLS) protocol and your network authentication is enabled with the Kerberos protocol.

Procedure

1. In a supported web browser on the cluster in which the Ranger service is installed, log in to Cloudera Manager as a user with full administrative privileges.
2. From the Cloudera Manager navigation side-bar, select Administration Security .
3. On the Security Status page, click Set up HDFS Data At Rest Encryption.
4. In the Set up HDFS Data At Rest Encryption page, select the A file-based password-protected Java KeyStore option.

   A list of tasks are displayed at the bottom of the page. To successfully set up HDFS Data at Rest encryption, these tasks must be completed.

   Note: Kerberos and TLS must be enabled. If the steps associated with these tasks do not display Completed in the Status column, before continuing, click the link associated with the uncompleted task and follow the Wizard's instructions.
5. To set up HDFS Encryption, follow the instructions as described below for each of the Set up HDFS Data At Rest Encryption Wizard's steps.

Related Information

TLS/SSL and Its Use of Certificates

Enabling Kerberos Authentication for CDP

Adding the Java KeyStore KMS Service

The Set up HDFS Data At Rest Encryption wizard's installation step that installs the Java KeyStore KMS service on your cluster.

About this task

Describes the steps that add the Java KeyStore KMS service to the cluster.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Add a Java KeyStore KMS Service.

   The Add Java KeyStore KMS Service to Cluster Wizard opens.
2. In the Assign Roles page, verify that the hostname is the required server on which to add the Java KeyStore KMS service by clicking inside the Key Management Server field. By default, this field is populated by the Wizard.

   The Hosts Selected page opens.
3. In the Hosts Selected page, scroll down and from the Hostname column, locate the hostname that was selected by the Wizard. Notice in the Added Roles column the Java KeyStore KMS Key Management Server (KMS) role icon. This role is added during the installation.
4. Do one of the following:
   - If the pre-selected host is correct, confirm the Wizard's choice by clicking OK.
   - If the pre-selected host is incorrect, deselect the check box of the Wizard's choice, select the hostname check box of the required server, and then click OK.
5. Back in the Assign Roles page, click Continue.

   The Setup Access Control List (ACL) page opens, which enables you to create a key admin user and group that can perform special functions. By default, non-admin users cannot access encrypted data.

6. Depending on your requirements do one of the following:
   - To use your own kms-acls.xml file, select the Use Your Own kms-acls.xml File option and in the kms-acls.xml text box, enter the contents of your kms-acls.xml file and then click Continue.
   - To have the wizard create a kms-acls.xml file and appropriate ACLs do the following:
     a. Select the Use Recommendation.
     b. In the Key Admin User field, enter the name of the user with administrative privileges.
     c. In the Key Admin Group field, enter the name of the Group in which the user's can perform special functions.

        Note: For multiple users and groups, enter the user and group names separated by a comma.

     d. Click Generate ACLs, which populates the text field with a list of appropriate ACLs.
     e. Click Continue.

   The Setup TLS for Java KeyStore KMS page opens, which provides high-level instructions for configuring TLS communication between your cluster and the Java KeyStore KMS.
7. Click Continue.

   The Review Changes page opens.
8. Review the settings and make any required changes before clicking Continue.

   Tip: For information about a setting, click the Show Description icon.
9. In the Command Details page, monitor the installation of the Java KeyStore KMS service. When the Status displays Finished the Java KeyStore KMS is installed and tested.
10. Click Continue.

    The Setup HDFS Dependency page opens.
11. Click Continue, which enables the HDFS service and opens the Summary page.
12. Click Finish, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.
13. (Optional) Verify that the Java KeyStore KMS service appears in the Cloudera Manager Clusters components list.

    If the Java KeyStore KMS service was not started by the installation wizard, do the following:
    a. Go to Cloudera Manager's Home page by clicking the Cloudera Manager icon.
    b. In the Cloudera Manager Clusters components list, locate and click Java KeyStore KMS.
    c. From the Actions menu, click Start.

What to do next

Restart the stale services and redeploy the client configuration.

Restarting the Stale Services and Redeploying the Client Configuration

The Set up HDFS Data At Rest Encryption wizard's step for restarting stale services and redeploying the client configuration.

About this task

Describes the steps that restart stale services after installing the Data-at-Rest HDFS Ranger KMS service option on your cluster.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Restart stale services and redeploy client configuration..

   The Stale Configurations page opens.

2. Click Restart Stale Services.

   The restart Stale Services page opens.
3. Verify that the Re-deploy client configuration check box is selected and click Restart Now.
4. In the Command Details page, monitor the restart process. When the Status displays Finished, click Continue, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.

What to do next

Validate that the Data-at-Rest HDFS Ranger KMS service option can successfully encrypt your data to-and-from HDFS.

Validating Data Encryption to-and-from HDFS

The Set up HDFS Data At Rest Encryption wizard's step for validating the data encryption to-and-from HDFS.

About this task

Describes the steps which verify that the Data-at-Rest HDFS Ranger KMS service option can successfully encrypt your data to-and-from HDFS.

Procedure

1. From the Step column in the Set up HDFS Data at Rest Encryption for Cluster page, click Validate Data Encryption.

   The Validate Data Encryption page opens, which displays a list of commands and instructions for creating an encryption zone and adding data.
2. In a terminal, log in to one of the hosts in your cluster and run each of the following commands:

   a) Create a key and directory by entering the following:

   ```
   kinit KEY_ADMIN_USER
   hadoop key create mykey1
   hdfs dfs -mkdir /tmp/zonel
   ```

   Where, *KEY_ADMIN_USER* is the key administrator whose role can perform the following actions:

   - Configure HDFS encryption, administer Key Trustee Server, and manage encryption keys
   - Start, stop, and restart Ranger KMS
   - Configure Ranger KMS Policies
   - View configuration and monitoring information in Cloudera Manager
   - View service and monitoring information
   - View events and logs
   - View Replication jobs and snapshot policies
   - View YARN applications and Impala queries

   b) Create a zone and link to the key, by entering the following:

   ```
   kinit hdfs hdfs
   hdfs crypto -createZone -keyName mykey1 -path /tmp/zone1
   ```

   c) Create a file, put it in your zone, and verify that the file can be decrypted, by entering the following:

   ```
   kinit KEY ADMIN_USER
   echo "Hello World" > /tmp/helloWorld.txt
   hdfs dfs -put /tmp/helloWorld.txt /tmp/zone1
   hdfs dfs -cat /tmp/zonel/helloWorld.txt
   rm /tmp/helloWorld.txt
   ```

   d) Verify that the stored file is encrypted, by entering the following:

   ```
   kinit hdfs
   ```

```
hdfs dfs -cat /.reserved/raw/tmp/zonel/helloWorld.txt
hdfs dfs -rm -R / tmp/zone1
hdfs crypto -listZones
```

3. When completed, click Close, which returns you to the Set up HDFS Data at Rest Encryption for Cluster page.

Installing Cloudera Navigator Key Trustee Server

You can install Navigator Key Trustee Server using Cloudera Manager with parcels. Command line package installs are not supported.

Important: Before installing Cloudera Navigator Key Trustee Server, see *Encrypting Data at Rest* for important considerations.

When the Key Trustee Server role is created it is tightly bound to the identity of the host on which it is installed. Moving the role to a different host, changing the host name, or changing the IP of the host is not supported.

Note: If you are using or planning to use Key Trustee Server in conjunction with a Runtime cluster, Cloudera strongly recommends using Cloudera Manager to install and manage Key Trustee Server to take advantage of Cloudera Manager's robust deployment, management, and monitoring capabilities.

See *Encrypting Data at Rest* for more information about encryption and Key Trustee Server requirements.

Installing Key Trustee Server using Cloudera Manager

If you are installing Navigator Key Trustee Server (KTS) for use with HDFS Transparent Encryption, the Set up HDFS Data At Rest Encryption wizard installs and configures KTS.

Before you begin

Note: In RHEL 8 and RHEL 9, the default is Python 3 which KTS does not support. So, you need to install Python 2.7 in the specified location and Python 2.7 must be linked to /usr/bin/python.

Important: The prerequisite is for RHEL9 only.

Navigator KTS works on Python 2.7. RHEL9 does not support Python 2.7 and the corresponding Python 2.7 packages including virtualenv. You need to install Python 2.7 in the specified location which will be used by KTS. Do not change the paths mentioned in the following steps as the paths are hardcoded and any change might result in failure. This is required only for the KTS cluster and not for CDP cluster where other services are installed.

1. Install Crontabs manually as they are not present in RHEL 9 by default:

```
dnf install -y crontabs
```

2. Install the zlib package before installing Python 2.7:

```
dnf install zlib-devel
```

Note: In addition, you must install all necessary Linux packages before installing Python 2.7. If you face any missing Python package error during KTS installation, like ImportError: No module named zlib, then after installing the necessary Linux packages, reinstall Python 2.7, so that the necessary Python packages are included in the Python installation.

3. Execute the following command to install Python 2.7:

```
mkdir /usr/local/custom_py_2.7.16
wget http://www.python.org/ftp/python/2.7.16/Python-2.7.16.tar.xz
xz -df Python-2.7.16.tar.xz
tar -xvf Python-2.7.16.tar
```

```
cd Python-2.7.16
./configure --prefix=/usr/local/custom_py_2.7.16 --with-threads
make
make install
cd ..
```

Procedure

1. (Recommended) Create a new cluster in Cloudera Manager containing only the host that Key Trustee Server will be installed on. Cloudera recommends that each cluster use its own KTS instance. Although sharing a single KTS across clusters is technically possible, it is neither approved nor supported for security reasons—specifically, the increased security risks associated with single point of failure for encryption keys used by multiple clusters. For a better understanding of additional security reasons for this recommendation, see *Data at Rest Encryption Reference Architecture*.

   Important: The Add Cluster wizard prompts you to install Runtime and other cluster services. To exit the wizard without installing Runtime, select a version of Runtime to install and continue. When the installation begins, click the Cloudera Manager logo in the upper left corner and confirm you want to exit the wizard. This allows you to create the dedicated cluster with the Key Trustee Server hosts without installing Runtime or other services that are not required for Key Trustee Server.
2. Download the latest Key Trustee Server parcel from the Cloudera.com/downloads page. From CDP Private Cloud Base 7.1.6, the KEYTRUSTEE_SERVER parcel is available in the same location in which the Cloudera runtime parcel is placed. If you have configured the parcel repository for CDP Private Cloud Base upgrade, the KEYTRUSTEE_SERVER parcel is displayed automatically.
3. Follow the steps in Using a Local Parcel Repository to register the local parcel with Cloudera Manager.
4. On the Key Trustee Server cluster home page, click the More Options (ellipsis) icon, then click Add Service.
5. Select Key Trustee Server, then click Continue.
6. Use the Add Key Trustee Server Service wizard to install Key Trustee Server.
7. Key Trustee Server appears in the cluster components list.

What to do next

After installing Key Trustee Server using Cloudera Manager, continue to *Securing Key Trustee Server Host*.

Securing Key Trustee Server Host

Cloudera strongly recommends securing the Key Trustee Server host to protect against unauthorized access to Key Trustee Server. Red Hat provides security guides for RHEL 7.

Cloudera also recommends configuring the Key Trustee Server host to allow network communication only over certain ports.

You can use the following examples to create iptables rules for an EDH cluster. Add any other ports required by your environment, subject to your organization security policies. Note that in this example port 5432 is the database port for the Key Trustee database on legacy machines (prior to release 5.5). Port 11371 is the current port on which Key Trustee communicates, and port 11381 is the database port. Exercise caution if blocking other ports, as this can cause a disruption in service.

```
# Flush iptables
iptables -F
iptables -X

# Allow unlimited traffic on loopback (localhost) connection
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
# Allow established, related connections
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# Open all Cloudera Manager ports to allow Key Trustee Server to work prop
erly

iptables -A INPUT -p tcp -m tcp --dport 5432 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 11371 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 11381 -j ACCEPT

# Drop all other connections
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP

# Save iptables rules so that they're loaded if the system is restarted
sed 's/IPTABLES_SAVE_ON_STOP="no"/IPTABLES_SAVE_ON_STOP="yes"/' -i /etc/sy
sconfig/iptables-config
sed 's/IPTABLES_SAVE_ON_RESTART="no"/IPTABLES_SAVE_ON_RESTART="yes"/' -i /et
c/sysconfig/iptables-config
```

Leveraging Built-in Processor Instruction Sets

Learn about the built-in processor instruction sets including AES-NI and Intel RDRAND.

AES-NI

The Advanced Encryption Standard New Instructions (AES-NI) instruction set is designed to improve the speed of encryption and decryption using AES. Some newer processors come with AES-NI, which can be enabled on a per-server basis. If you are uncertain whether AES-NI is available on a device, run the following command to verify:

```
grep -o aes /proc/cpuinfo
```

To determine whether the AES-NI kernel module is loaded, run the following command:

```
sudo lsmod | grep aesni
```

If the CPU supports AES-NI but the kernel module is not loaded, see your operating system documentation for instructions on installing the aesni-intel module.

Intel RDRAND

The Intel RDRAND instruction set, along with its underlying Digital Random Number Generator (DRNG), is useful for generating keys for cryptographic protocols without using haveged.

To determine whether the CPU supports RDRAND, run the following command:

```
grep -o rdrand /proc/cpuinfo
```

To enable RDRAND, install rng-tools version 4 or higher:

1. Download the source code:

   ```
   sudo wget http://downloads.sourceforge.net/project/gkernel/rng-tools/4/r
   ng-tools-4.tar.gz
   ```

2. Extract the source code:

   ```
   tar xvfz rng-tools-4.tar.gz
   ```

3. Enter the rng-tools-4 directory:

   ```
   cd rng-tools-4
   ```

4. Run ./configure.

5. Run make.
6. Run make install.

Start rngd with the following command:

```
sudo rngd --no-tpm=1 -o /dev/random
```

Initializing Key Trustee Server

After installing Key Trustee Server, you must initialize it before it is operational. See the Key Trustee Server documentation and High Availability documentation for details.

Related Information

Initializing Standalone Key Trustee Server

Cloudera Navigator Key Trustee Server Overview

Setting Up Key Trustee Server High Availability

Installing Cloudera Navigator Encrypt

Learn about installing Navigator Encrypt, setting up TLS certificates on a Navigator Encrypt client, entropy requirements , and uninstalling and reinstalling NavEncrypt.

Before you begin

See Data at Rest Encryption Requirements for more information about encryption and Navigator Encrypt requirements.

About this task

Important: Before installing Cloudera Navigator Encrypt, see Encrypting Data at Rest and the Product Compatibility Matrix for Cloudera Navigator Encryption for important considerations.

Setting Up an Internal Repository

You must create an internal repository to install or upgrade Navigator Encrypt. For instructions on creating internal repositories (including Cloudera Manager, CDH, and Cloudera Navigator encryption components), see Configuring a Local Package Repository.

Installing Navigator Encrypt (RHEL-Compatible)

Learn how to install RHEL compatible Navigator Encrypt. The steps below show an example of how to install NavEncrypt on a cluster running Red Hat Linux.

About this task

Note: For details about supported Linux Operating Systems, refer to the Product Compatibility Matrix for Cloudera Navigator Encryption.

Procedure

1. Install the EPEL Repository.

   Dependent packages are available through the Extra Packages for Enterprise Linux (EPEL) repository. To install the EPEL repository, install the epel-release package. The EPEL repository for each release of RHEL is different ,so confirm the host is set up correctly.

2. Install the NavEncrypt Repository.
   a) mkdir -p /root/navencrypt-repo
   b) Fetch the NavEncrypt repository from the Cloudera download site, for example: wget https://archive.cloudera.com/p/navencrypt7/7.1.9.1000/rhel8/navigator-encrypt-7.1.9.1000-el8.tar.gz
   c) tar -zxvf navigator-encrypt-7.1.9.1000-el8.tar.gz --directory /root/navencrypt-repo
3. Create and edit file /etc/yum.repos.d/navencrypt-repo:

   [navencrypt-repo] name=Cloudera NavEncrypt enabled=1 autorefresh=0 gpgcheck=1 baseurl=file:///root/navencrypt-repo gpgkey=file:///root/navencrypt-repo/nepub.asc
4. Install Kernel Libraries.

   For Navigator Encrypt to run as a kernel module, you must download and install the kernel development headers. Each kernel module is compiled specifically for the underlying kernel version. Running as a kernel module allows Navigator Encrypt to provide high performance and complete transparency to user-space applications.

   To determine your current kernel version, run uname -r.

   To install the development headers for your current kernel version, run:

```
sudo yum install kernel-headers-$(uname -r) kernel-devel-$(uname -r)
```

5. (RHEL or CentOS Only) Manually Install dkms.

   With some versions of RHEL7 and CentOS 7, because of a broken dependency, you must manually install the dkms package. To do this ,you must locate a repo that has a version of dkms that is compatible with the version of RHEL the host is running.

```
sudo yum install https://download-ib01.fedoraproject.org/pub/epel/7/aarc
h64/Packages/d/dkms-2.7.1-1.el7.noarch.rpm
```

6. yum install libkeytrustee
7. yum install navencrypt-kernel-module

   Note: navencrypt-kernel-module depends on package dkms. Depending on the versions of the packages yum might update the linux kernel. Check for this and if it has occurred, reboot the host.
8. Install Navigator Encrypt.

   Install the Navigator Encrypt client using the yum package manager:

```
sudo yum install navencrypt
```

   If you attempt to install navencrypt-kernel-module with incorrect or missing kernel headers, you see a message like the following:

```
Building navencryptfs 3.8.0 DKMS kernel module...

#################### BUILDING ERROR ####################


Creating symlink /var/lib/dkms/navencryptfs/3.8.0/source ->
                 /usr/src/navencryptfs-3.8.0
DKMS: add completed.
Error! echo
Your kernel headers for kernel 3.10.0-229.4.2.el7.x86_64 cannot be found a
t
/lib/modules/3.10.0-229.4.2.el7.x86_64/build or /lib/modules/3.10.0-229.4.
2.el7.x86_64/source.

#################### BUILDING ERROR ####################
```

```
Failed installation of navencryptfs 3.8.0 DKMS kernel module !
```

To recover, see Navigator Encrypt Kernel Module Setup.

9. Confirm NavEncrypt is installed.

   yum list installed | egrep "naven|keytrust"

Installing Navigator Encrypt (SLES-12)

Learn how to install SLES 12 compatible Navigator Encrypt . The steps below show an example of installing SLES 12 compatible NavEncrypt, assuming the user is root.

Procedure

1. Install the NavEncrypt Repository.
   a) mkdir -p /root/navencrypt-repo
   b) Fetch the NavEncrypt repository from the Cloudera download site, for example: wget https://archive.cloudera.com/p/navencrypt7/7.1.9.1000/sles15/navigator-encrypt-7.1.9.1000_sles15.4-0.tar.gz
   c) tar -zxvf navigator-encrypt-7.1.9.1000_sles15.4-0.tar.gz --directory /root/navencrypt-repo
2. Create and edit file /etc/zypp/repos.d/navencrypt.repo:

   [navencrypt-repo] name=Cloudera NavEncrypt enabled=1 autorefresh=0 gpgcheck=1 baseurl=file:///root/navencrypt-repo gpgkey=file:///root/navencrypt-repo/nepub.asc
3. Confirm zypper can access repo.

   ```
   zypper search -r navencrypt-repo
   ```
4. zypper install libkeytrustee
5. Install the Kernel Module Package and Navigator Encrypt Client.

   Install the kernel module package (KMP) and Navigator Encrypt client with zypper:

   ```
   sudo zypper install cloudera-navencryptfs-kmp-default
   sudo zypper install navencrypt
   ```
6. Confirm NavEncrypt is installed

   zypper search -i | egrep "naven|keytrust"
7. systemctl daemon-reload

Installing Navigator Encrypt (SLES-15)

Learn how to install SLES 15 compatible Navigator Encrypt. The following steps show an example of installing SLES 15 compatible Navigator Encrypt, assuming the user is root.

Procedure

1. Install the Navigator Encrypt repository.
   a) Execute the following command:

   ```
   mkdir -p /root/navencrypt-repo
   ```

   b) Fetch the Navigator Encrypt repository from the Cloudera download site.

   For example,

   ```
   wget https://archive.cloudera.com/p/navencrypt7/7.1.9.1000/sles15/naviga
   tor-encrypt-7.1.9.1000-49-sles15.tar.gz
   ```

   c) Execute the following command:

   ```
   tar -zxvf navigator-encrypt-7.1.9.1000-49-sles15.tar.gz --directory /roo
   t/navencrypt-repo
   ```

2. Create and edit the /etc/zypp/repos.d/navencrypt.repo file:

   ```
   [navencrypt-repo] name=Cloudera NavEncrypt enabled=1 autorefresh=0 gpgch
   eck=1 baseurl=file:///root/navencrypt-repo gpgkey=file:///root/navencrypt-
   repo/nepub.asc
   ```

3. Confirm that Zypper can access the repository.

   ```
   zypper se -v  navencrypt libkeytrustee
   ```

   ```
           S  | Name                              | Type       | Version
                    | Arch   | Repository
           ---+-----------------------------------+-----------+----------
   ------------------+--------+-------------------
              | cloudera-navencryptfs             | srcpackage | 7.1.9-0.0
                    | noarch | Cloudera NavEncrypt
               name: cloudera-navencryptfs
           i+ | cloudera-navencryptfs-kmp-default | package    | 7.1.9_k5.
   14.21_150400.22-0.0 | x86_64 | Cloudera NavEncrypt
               name: cloudera-navencryptfs-kmp-default
            v  | cloudera-navencryptfs-kmp-default | package    | 7.1.9_k5.1
   4.21_150500.53-0.0 | x86_64 | Cloudera NavEncrypt
               name: cloudera-navencryptfs-kmp-default
           i+ | libkeytrustee                     | package    | 7.1.9.1000_s
   les15.4-49       | x86_64 | Cloudera NavEncrypt
               name: libkeytrustee
           v  | libkeytrustee                     | package    | 7.1.9.1000
   _sles15.5-49       | x86_64 | Cloudera NavEncrypt
               name: libkeytrustee
           i+ | navencrypt                        | package    | 7.1.9.1000_s
   les15.4-49       | x86_64 | Cloudera NavEncrypt
               name: navencrypt
           v  | navencrypt                        | package    | 7.1.9.100
   0_sles15.5-49       | x86_64 | Cloudera NavEncrypt
               name: navencrypt
   ```

   You can see that the repository contains packages for SLES15-SP4 and SLES15-SP5.

4. If you are running SLES15-SP4, install:

   - zypper in libkeytrustee-7.1.9.1000_sles15.4-49
   - zypper in cloudera-navencryptfs-kmp-default-7.1.9_k5.14.21_150400.22-0.0
   - zypper in navencrypt-7.1.9.1000_sles15.4-49

5. Confirm that Navigator Encrypt SLES15.4 is installed:

```
zypper se -v -i navencrypt libkeytrustee
```

```
S  | Name                               | Type    | Version
      | Arch   | Repository
---+------------------------------------+---------+-----------------------
------+--------+--------------------
i+ | cloudera-navencryptfs-kmp-default | package | 7.1.9_k5.14.21_15040
0.22-0.0 | x86_64 | Cloudera NavEncrypt
    name: cloudera-navencryptfs-kmp-default
i+ | libkeytrustee                      | package | 7.1.9.1000_sles15.4-49
      | x86_64 | Cloudera NavEncrypt
    name: libkeytrustee
i+ | navencrypt                         | package | 7.1.9.1000_sles15.4-49
      | x86_64 | Cloudera NavEncrypt
    name: navencrypt
```

6. If you are running SLES15-SP5, install:

   - zypper in libkeytrustee-7.1.9.1000_sles15.5-49
   - zypper in cloudera-navencryptfs-kmp-default-7.1.9_k5.14.21_150500.53-0.0
   - zypper in navencrypt-7.1.9.1000_sles15.5-49

7. Confirm that Navigator Encrypt SLES15.5 is installed:

```
zypper se -v -i navencrypt libkeytrustee
```

```
S  | Name                               | Type    | Version
      | Arch   | Repository
---+------------------------------------+---------+-----------------------
------+--------+--------------------
i+ | cloudera-navencryptfs-kmp-default | package | 7.1.9_k5.14.21_15050
0.53-0.0 | x86_64 | Cloudera NavEncrypt
    name: cloudera-navencryptfs-kmp-default
i+ | libkeytrustee                      | package | 7.1.9.1000_sles15.5-49
      | x86_64 | Cloudera NavEncrypt
    name: libkeytrustee
i+ | navencrypt                         | package | 7.1.9.1000_sles15.5-49
      | x86_64 | Cloudera NavEncrypt
    name: navencrypt
```

8. Reload the systemd files:

```
systemctl daemon-reload
```

Installing Navigator Encrypt (Ubuntu)

Learn how to install Ubuntu compatible Navigator Encrypt . The steps below show an example of installing Ubuntu compatible NavEncrypt, assuming the user is root.

Procedure

1. Install the NavEncrypt Repository.

   a) mkdir -p /root/navencrypt-repo
   b) Fetch the NavEncrypt repository from the Cloudera download site, for example: wget https://archive.cloudera.com/p/navencrypt7/7.1.9.1000/ubuntu/navigator-encrypt-7.1.9.1000-61-ubuntu20.tar.gz
   c) tar -zxvf navigator-encrypt-7.1.9.1000-61-ubuntu20.tar.gz --directory /root/navencrypt-repo
   d) apt-key add /root/navencrypt-repo/nepub.asc

2. Install Kernel Headers.

   Determine your kernel version by running uname -r, and install the appropriate headers:

   ```
   sudo apt-get install linux-headers-$(uname -r)
   ```

3. apt-get install libkeytrustee4
4. apt-get install navencrypt-kernel-module
5. Install the Navigator Encrypt Client.

   Install Navigator Encrypt:

   ```
   sudo apt-get install navencrypt
   ```

6. Confirm if NavEncrypt is installed.

   ```
   apt-cache search . | egrep "naven|keytrust"
   ```

Installing for Ranger KMS

If you are using Ranger KMS as your Key Management server you need to do the following:

Procedure

1. Generate a valid Kerberos ticket.

   There is a utility called navencrypt-gen-keytab that works with Cloudera Manager to create a valid Kerberos ticket.
2. Start the navencrypt-krb5 service after the keytab file is generated.

Setting Up TLS for Navigator Encrypt Clients

Transport Layer Security (TLS) certificates are used to secure communication with Navigator Encrypt. Cloudera strongly recommends using certificates signed by a trusted Certificate Authority (CA).

About this task

If the TLS certificate is signed by an unrecognized CA, such as an internal CA, then you must add the root certificate to the host certificate truststore of each Navigator Encrypt client. Be aware that Navigator Encrypt uses the operating system's truststore, which is distinct from the JDK truststore used by Cloudera Manager.

To set up TLS certificates on a Navigator Encrypt client:

Procedure

1. If not already installed, install the CA-certificates:

   ```
   yum install ca-certificates
   ```

2. Enable the dynamic CA configuration feature:

   ```
   update-ca-trust enable
   ```

3. Copy the root certificate into the host certificate truststore:

   ```
   cp /path/to/root.pem /etc/pki/ca-trust/source/anchors/
   ```

4. Update the host certificate truststore:

   ```
   update-ca-trust
   ```

Example

Example:

```
[root@navencrypt-1 ~]# systemctl stop navencrypt-mount
Stopping navencrypt directories
 * Umounting /dev/nvtest/test1 ...                            [  OK  ]
 * Umounting /dev/nvtest/test2 ...                            [  OK  ]
 * Unloading module ...                                       [  OK  ]

[root@navencrypt-1 ~]# update-ca-trust enable
[root@navencrypt-1 ~]# cp dd-1.lab.usa.company.com.pem /etc/pki/ca-trust/s
ource/anchors/
[root@navencrypt-1 ~]# update-ca-trust

[root@navencrypt-1 ~]# systemctl stop navencrypt-mount
Starting navencrypt directories
 * Mounting '/dev/nvtest/test1'                               [  OK  ]
 * Mounting '/dev/nvtest/test2'
```

Entropy Requirements

Many cryptographic operations, such as those used with TLS or HDFS encryption, require a sufficient level of system entropy to ensure randomness; likewise, Navigator Encrypt needs a source of random numbers to ensure good performance.

About this task

Hence, you need to make sure that the hosts running Navigator Encrypt (as well as Key Trustee Server, Key Trustee KMS) and have sufficient entropy to perform cryptographic operations.

You can check the available entropy on a Linux system by running the following command:

```
cat /proc/sys/kernel/random/entropy_avail
```

The output displays the entropy currently available. Check the entropy several times to determine the state of the entropy pool on the system. On hosts running a Linux kernel version less than 5.10.119, if the entropy is consistently low (500 or less), you must increase it by installing rng-tools version 4 or higher, and starting the rngd service. On hosts running a Linux kernel version of 5.10.119 or higher the entropy version will be stable at 256, unless there are special entropy requirements in place, no further action is required.

Install rng_tools Using Package Manager

Learn how to install rng_tools using Package Manager.

About this task

If version 4 or higher of the rng-tools package is available from the local package manager (yum), then install it directly from the package manager. If the appropriate version of rng-tools is unavailable, see Building rng-tools From Source on page 284.

For RHEL 7, run the following commands:

```
sudo yum install rng-tools
cp /usr/lib/systemd/system/rngd.service /etc/systemd/system/
systemctl daemon-reload
systemctl start rngd
systemctl enable rngd
```

Building rng-tools From Source

If you are unable to install rng-tools using package manager, you can build from source.

About this task

Note: If your package manager only offers an older version (3.x or earlier), then you must build from source.

To install and start rngd and build from source:

1. Download the source code:

   ```
   sudo wget http://downloads.sourceforge.net/project/gkernel/rng-tools/4/r
   ng-tools-4.tar.gz
   ```
2. Extract the source code:

   ```
   tar xvfz rng-tools-4.tar.gz
   ```
3. Enter the rng-tools-4 directory:

   ```
   cd rng-tools-4
   ```
4. Run ./configure
5. Run make
6. Run make install

After you have installed rng-tools, start the rngd daemon by running the following command as root:

```
sudo rngd --no-tpm=1 -o /dev/random
```

For improved performance, Cloudera recommends configuring Navigator Encrypt to read directly from /dev/random instead of /dev/urandom.

To configure Navigator Encrypt to use /dev/random as an entropy source, add --use-random to the navencrypt-prepare command when you are setting up Navigator Encrypt.

Uninstalling and Reinstalling Navigator Encrypt

Learn how to uninstall and reinstall Navigator Encrypt.

About this task

Uninstalling Navigator Encrypt

For RHEL-compatible OSes:

```
sudo yum remove navencrypt
sudo yum remove navencrypt-kernel-module
```

These commands remove the software itself. On RHEL-compatible OSes, the /etc/navencrypt directory is not removed as part of the uninstallation. Remove it manually if required.

Reinstalling Navigator Encrypt

After uninstalling Navigator Encrypt, repeat the preceding installation instructions for your distribution.

When Navigator Encrypt is uninstalled, the configuration files and directories located in /etc/navencrypt are not removed. Consequently, you do not need to use the navencrypt register command during reinstallation. If you no longer require the previous installation configuration information in the directory /etc/navencrypt, you can remove its contents.

Installing Cloudera Navigator Key HSM

Cloudera Navigator Key HSM is a universal hardware security module (HSM) driver that translates between the target HSM platform and Cloudera Navigator Key Trustee Server. Navigator Key HSM allows you to use a Key Trustee Server to securely store and retrieve encryption keys and other secure objects, without being limited solely to a hardware-based platform.

Before you begin

Important: Before installing Cloudera Navigator Key HSM, see Encrypting Data at Rest for important considerations.

You must install Key HSM on the same host as Key Trustee Server. See Data at Rest Encryption Requirements for more information about encryption and Key HSM requirements.

Procedure

1. Set up the Key HSM repository.

   Download the Key HSM tarball and create a local Key HSM repository with the files from the tarball.

   You must create an internal repository to install Cloudera Navigator Key HSM. For instructions on creating internal repositories, see Configuring a Local Package Repository.
2. Install the Key HSM repository.

   Add the local Key HSM repository you created earlier. See Configuring a Local Package Repository for more information.

   Run the following command to import the GPG key:

   ```
   $ sudo rpm --import http://repo.example.com/path/to/RPM-GPG-KEY-cloudera
   ```
3. Install the CDH repository.

   Key Trustee Server and Key HSM depend on the bigtop-utils package, which is included in the CDH repository. For instructions on adding the CDH repository, see Configuring a Local Package Repository.
4. Install Navigator Key HSM.

   Run the following command to install the Navigator Key HSM package:

   ```
   sudo yum install keytrustee-keyhsm
   ```

   Cloudera Navigator Key HSM is installed to the /usr/share/keytrustee-server-keyhsm directory by default.

Additional steps for FIPS + JDK 11

FIPS specific jars need to be included in the Key HSM classpath before starting Key HSM.

Procedure

1. Locate the FIPS specific jars.

   For example,

   ```
   -rw-r--r-- 1 root root 4093016 Jan  7 20:00 com-safelogic-cryptocomply-f
   ips-core.jar
   -rw-r--r-- 1 root root  759443 Jan  7 20:00 bctls.jar
   ```
2. Navigate to the Key HSM base directory:

   ```
   cd /usr/share/keytrustee-server-keyhsm/
   ```

3. Open the start script:

```
vim start.sh
```

4. Append the path for java -classpath variable to add the jars.

Your path may vary as per your environment.

Existing path :

```
java -classpath "*:/usr/safenet/lunaclient/jsp/lib/*:/opt/nfast/java/cla
sses/*:/opt/cloudhsm/java/*:/usr/share/keytrustee-server-keyhsm/conf/"
```

After appending :

```
java -classpath "*:/usr/safenet/lunaclient/jsp/lib/*:/opt/nfast/java/cla
sses/*:/opt/cloudhsm/java/*:/usr/share/keytrustee-server-keyhsm/conf/:/c
dep/extra_jars/*" -Djava.library.path=/usr/safenet/lunaclient/jsp/lib/:/
opt/cloudhsm/lib/
```

5. Start the Key HSM service.

Installing Ranger RMS

Ranger Resource Mapping Server (RMS) enables automatic translation of access policies from Hive to HDFS.

About this task

Note: Ranger RMS is an optional service and cannot be installed through the Add Service express wizard during the initial installation of services. To install Ranger RMS, you must first have a CDP Private Cloud Base cluster with Ranger, HDFS, and Hive services already installed.

Legacy CDH users used Hive policies in Apache Sentry that automatically linked Hive permissions with HDFS ACLs. This was especially convenient for external table data used by Spark or Hive.

Previously, Ranger only supported managing Hive and HDFS policies separately. Ranger RMS (Resource Mapping Server) allows you to authorize access to HDFS directories and files using policies defined for Hive tables. RMS is the service that enables Hive-HDFS ACL Sync.

Important: Cloudera does not support the MySQL server with Global Transaction Identifier (GTID) support enabled as a Ranger RMS database.

Before you begin

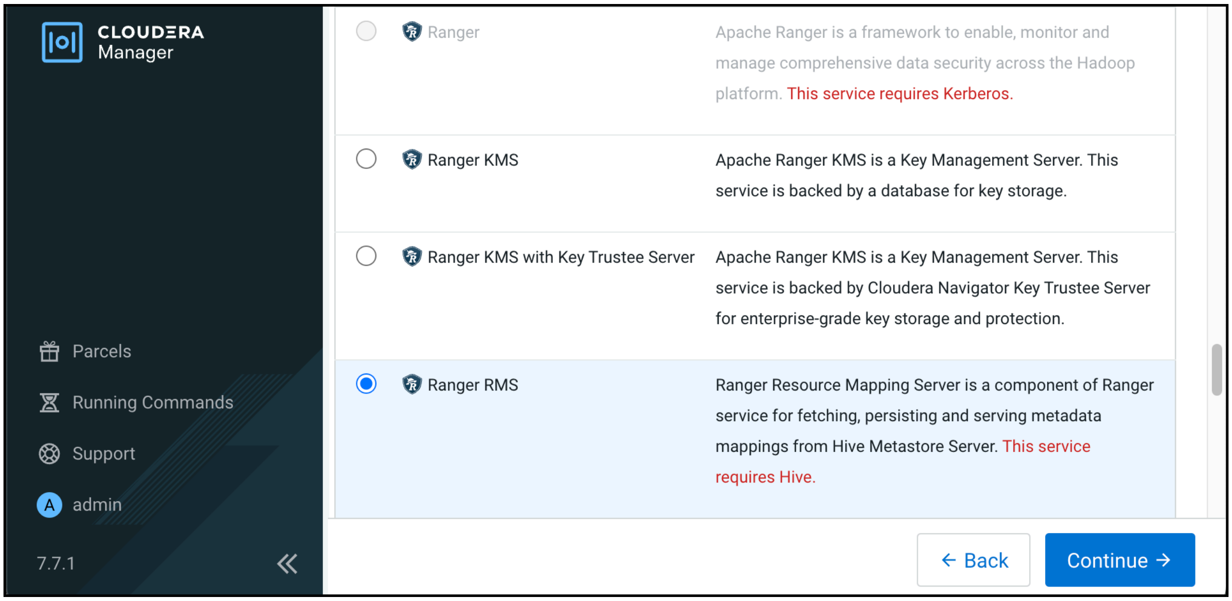
You must have installed:

- A CDP Private Cloud Base 7.1.4 or higher version cluster with Apache Ranger, Hive, and HDFS.
- Ranger RMS on the host where Hive_Gateway is available.

Procedure

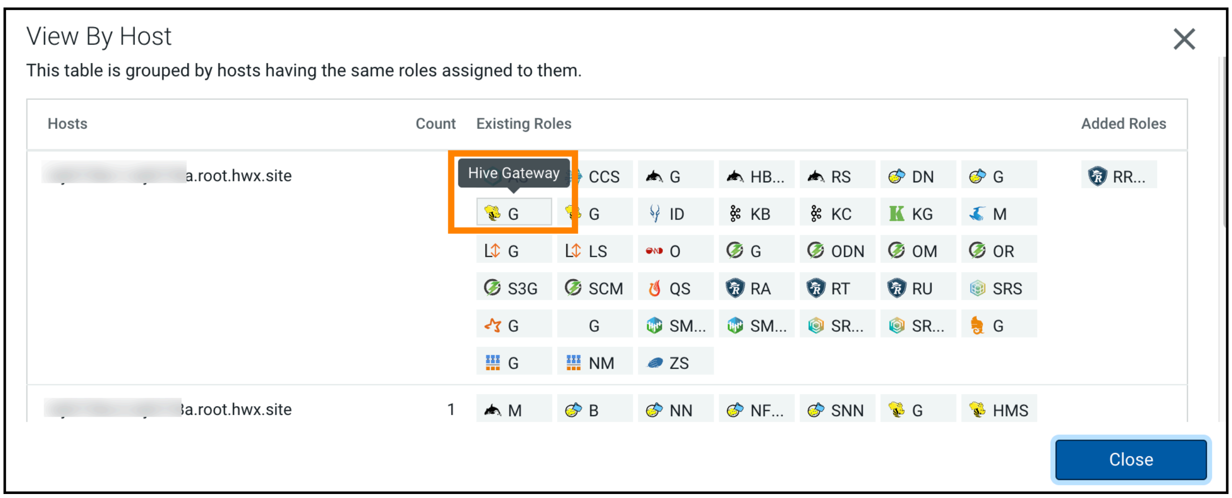
1. On the cluster home page, click , then click Add Service.

2. Select Ranger RMS, then click Continue.

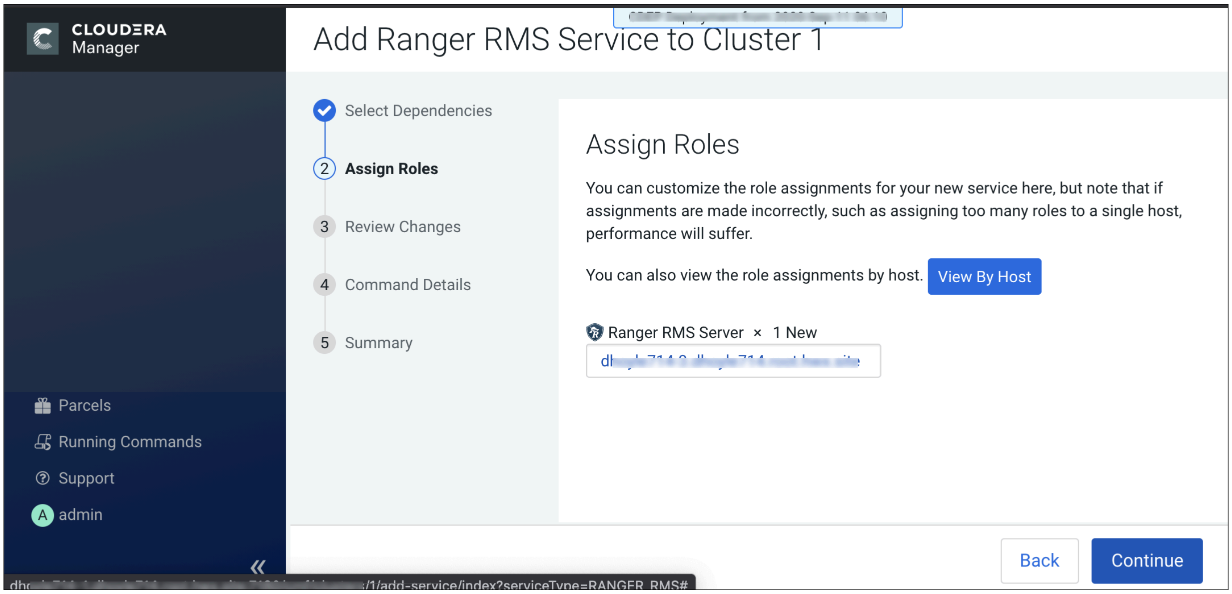


3. On Assign Roles, click View by Host.
4. On View by Host, verfiy that the host on which you install Ranger RMS has the required Hive Gateway role assigned, then click Close.

Figure 1: Verifying Hive Gateway role on a host



5. On Assign Roles, click Continue.



6. On **Review Changes**,

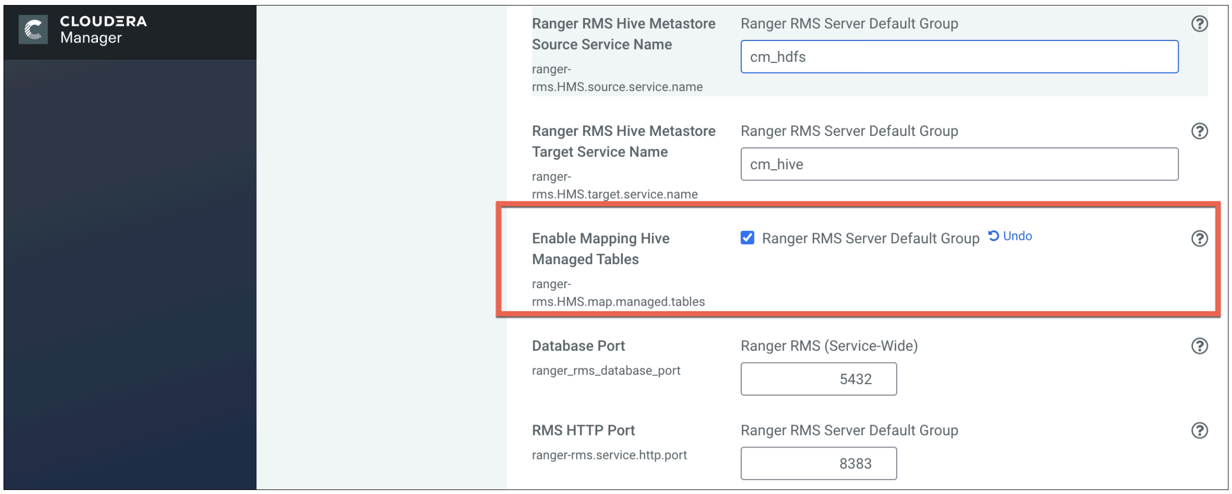
Note:

Ranger RMS uses the same database settings that are used in Ranger.

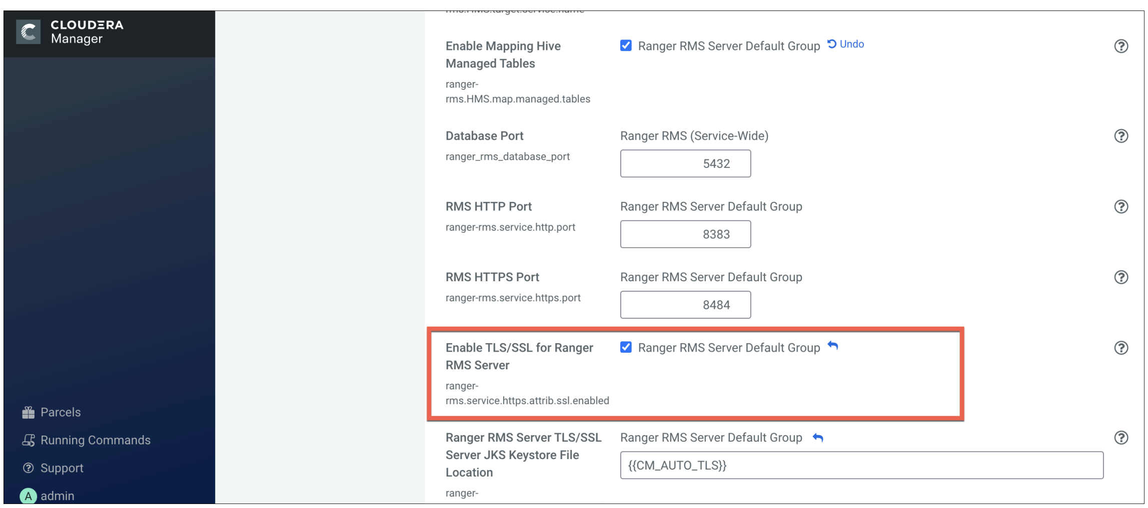
Ranger RMS should be installed in the same database instance as Ranger.

If Ranger service is using SSL-enabled database, make sure to have the database certificate / keystore / truststore file used by Ranger to connect to the SSL-enabled database present on the node where Ranger RMS is installed.

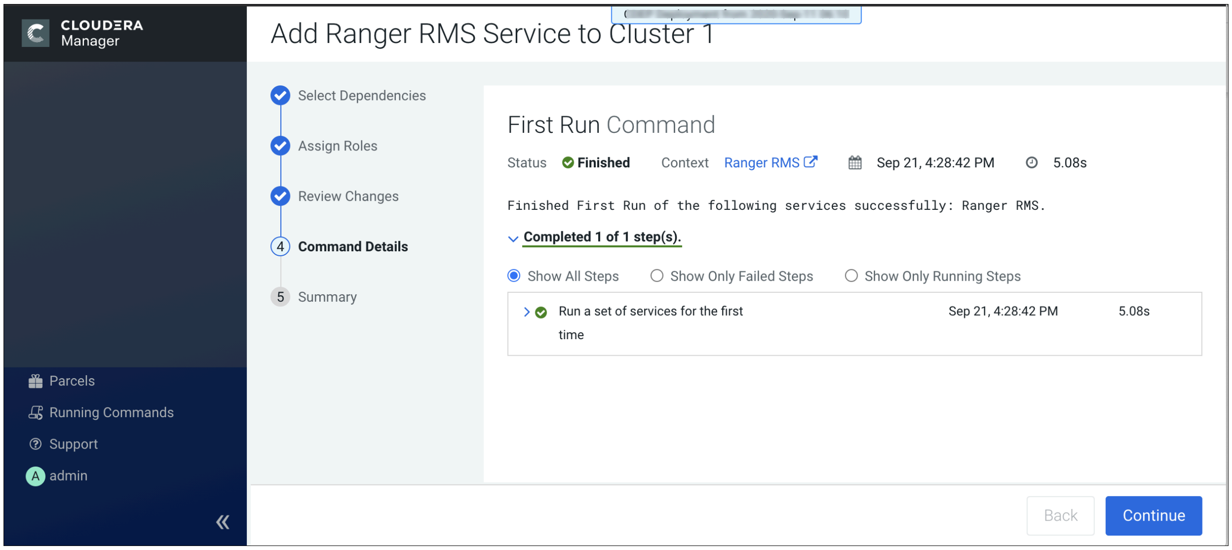
To track managed tables, select the Enable Mapping Hive Managed Tables option.



Note: If you are adding Ranger RMS in a cluster with SSL enabled, the Enable TLS/SSL for Ranger RMS Server option should be selected by default.

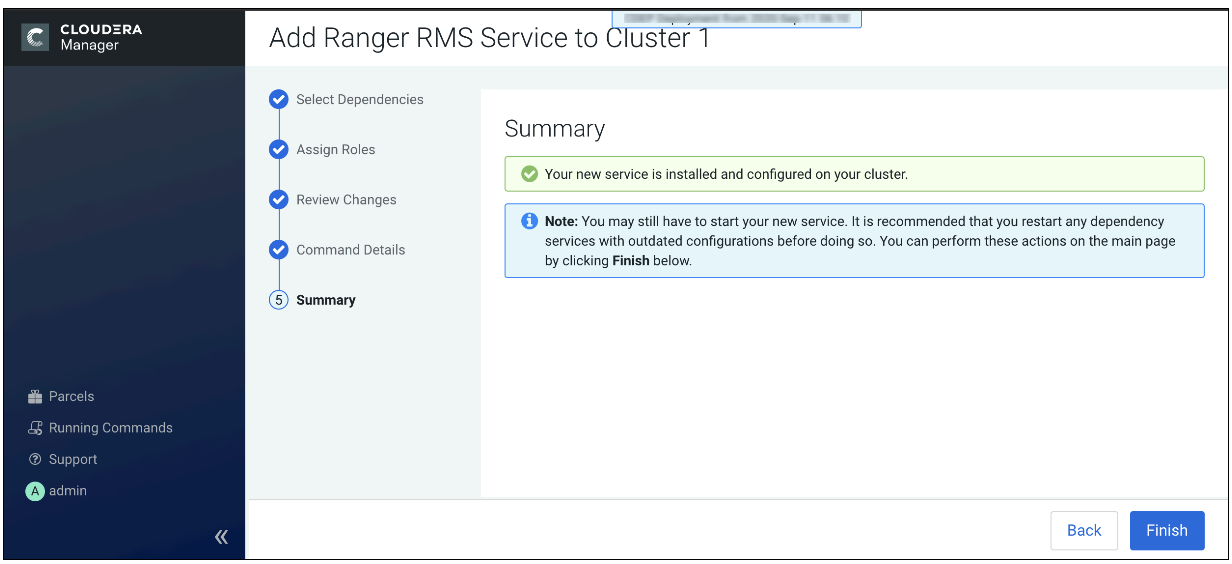


7. On the **Command Details** page, select run options, then click Continue.



8. On the **Summary** page, click Finish.

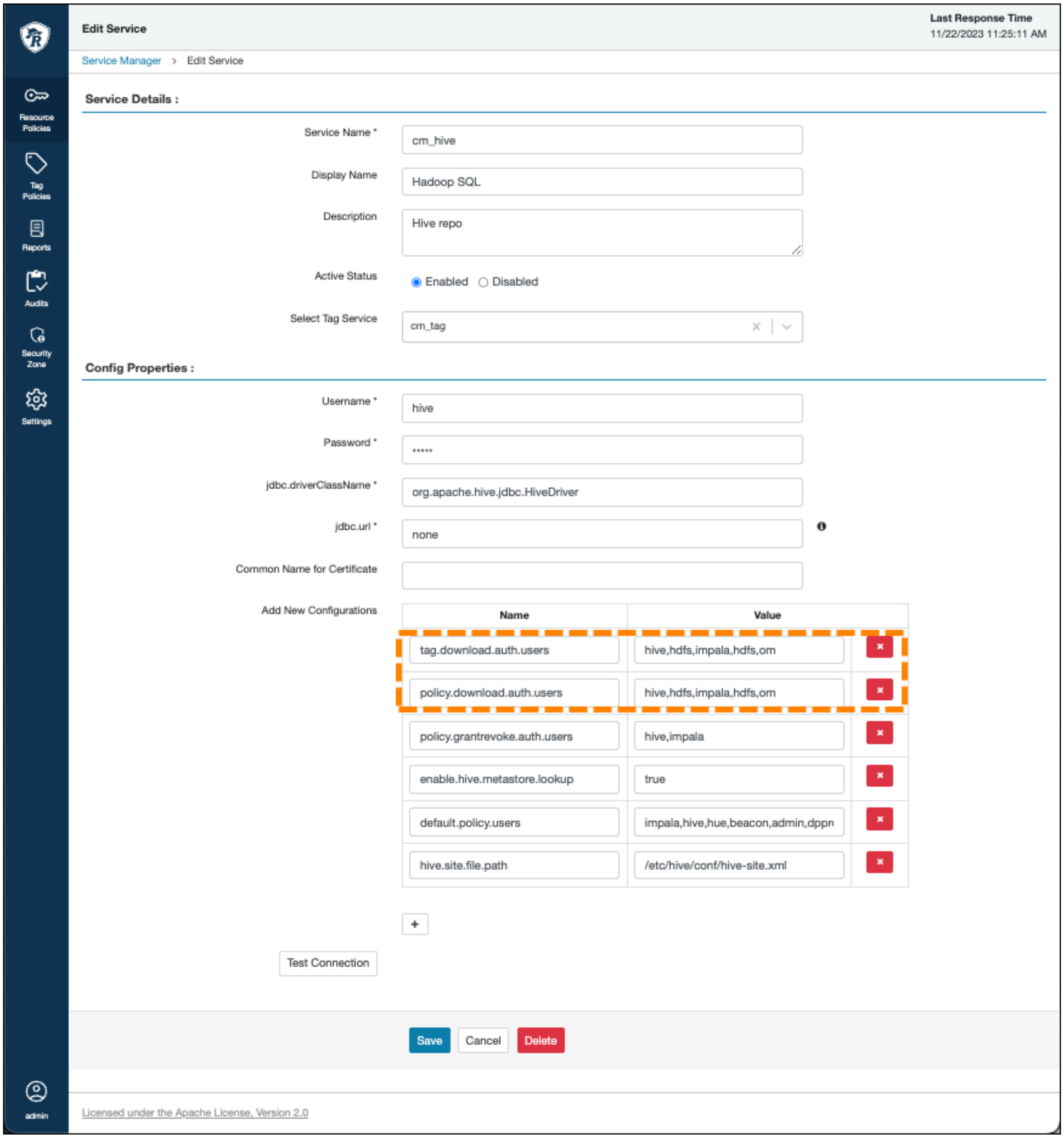
Important: Do not start the Ranger RMS service before completing the following additional configuration steps. Ranger RMS restarts when you restart stale services after completing configuration changes.



9. In Cloudera Manager Hive Service Configuration verify that the Hive Metastore Access Control and Ranger RMS Proxy User Hosts property, hadoop.proxyuser.rangerrms.hosts is set to *.

Note: rangerrms user is given superuser privilege only for the HiveMetaStore service, so rangerrms can access metadata information without an explicit Ranger policy allowing it necessary permissions. However, Hive operations such as drop database must be authorized in the hive-server2 by Ranger policies. You must create an appropriate Ranger policy which grants the user executing this command the required permission to do so.

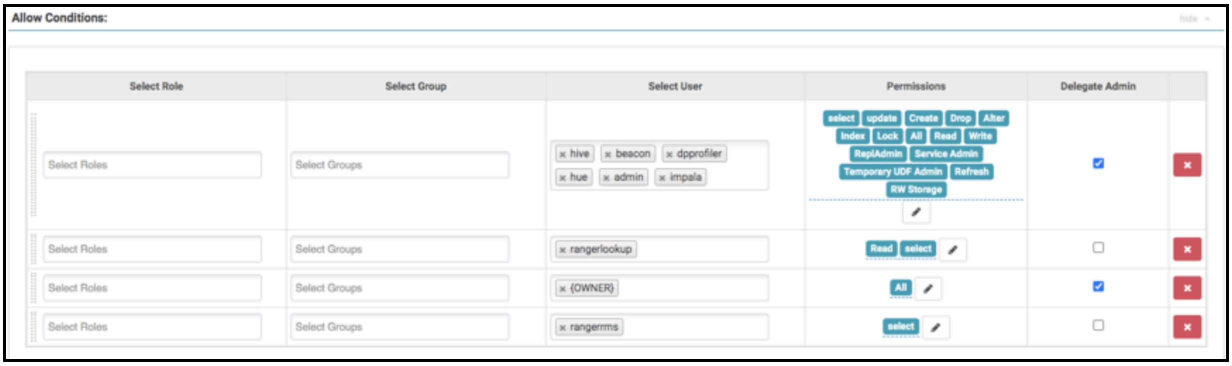
10. Log in to the Ranger Admin web UI. On the **Service Manager** page, click Edit for the Hadoop SQL service, then verify that hdfs has been added to the tag.download.auth.users and policy.download.auth.users configurations.



11. Configure Ranger policies with rangerrms user access before starting RMS and running the first sync from the Hive Metastore (HMS).

For example, you must give the rangerrms ID select access to Hive tables. This is configured under the policy "all - database, table".

Figure 2: Granting RMS user Select access to Hive tables

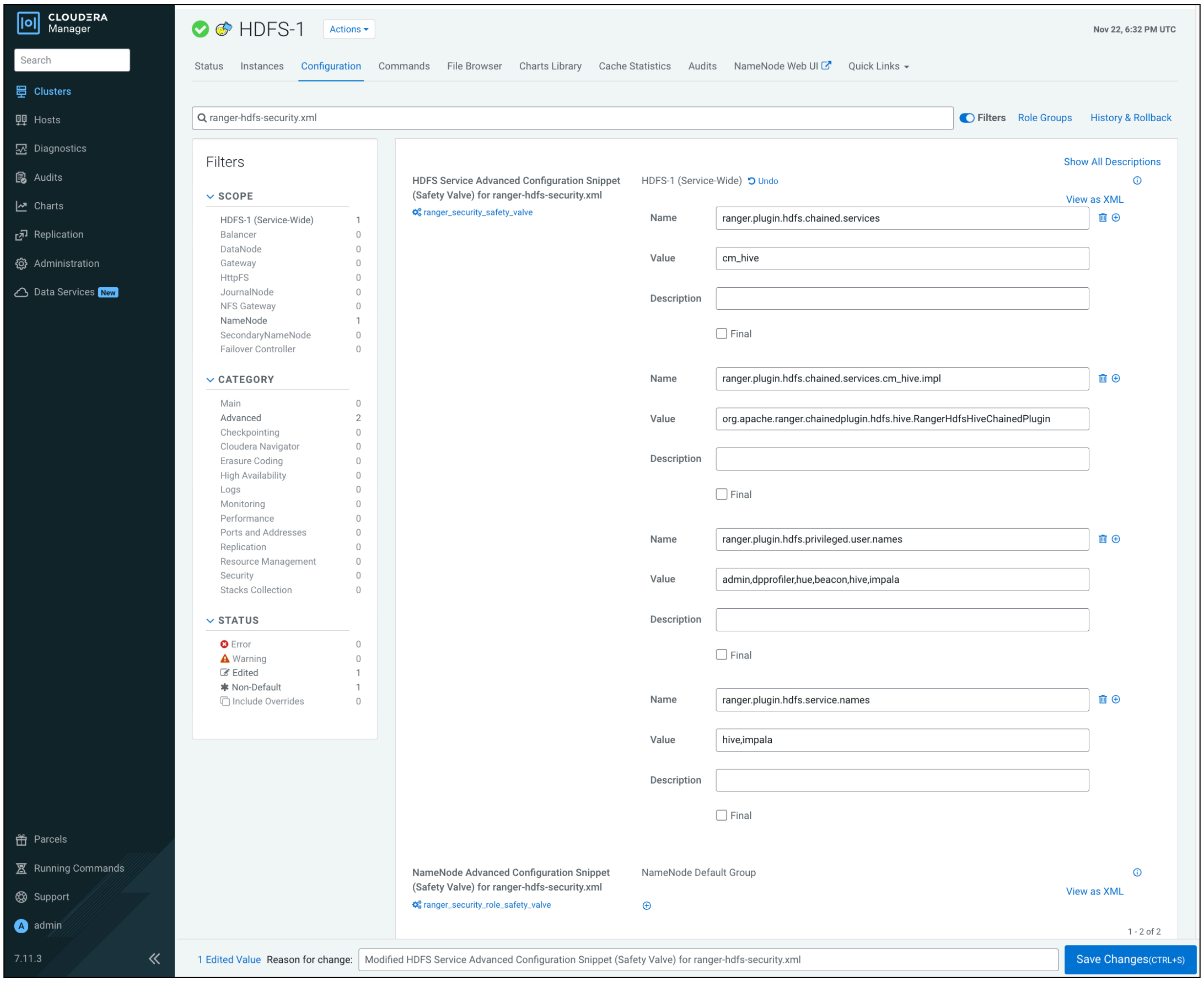


12. In Cloudera Manager, select HDFS Configuration , then search for Advanced Configuration Snippet (Safety Valve) for ranger-hdfs-security.xml. Use the Add (+) icons to add the following properties, then click Save Changes.

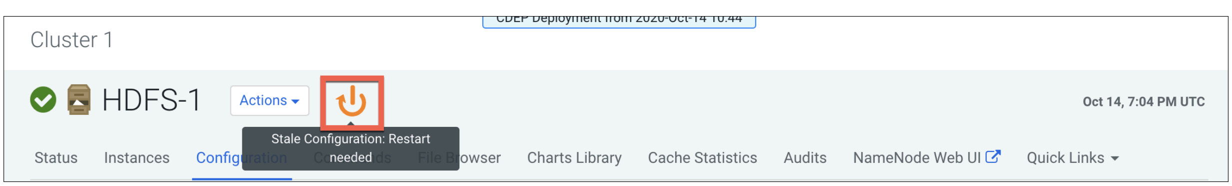
Name	Value
ranger.plugin.hdfs.chained.services	cm_hive
ranger.plugin.hdfs.chained.services.cm_hive.impl	org.apache.ranger.chainedplugin.hdfs.hive.RangerHdfsHiveChainedPlugin
ranger.plugin.hdfs.privileged.user.names	admin,dpprofiler,hue,beacon,hive,impala

Name	Value
ranger.plugin.hdfs.service.names	hive,impala

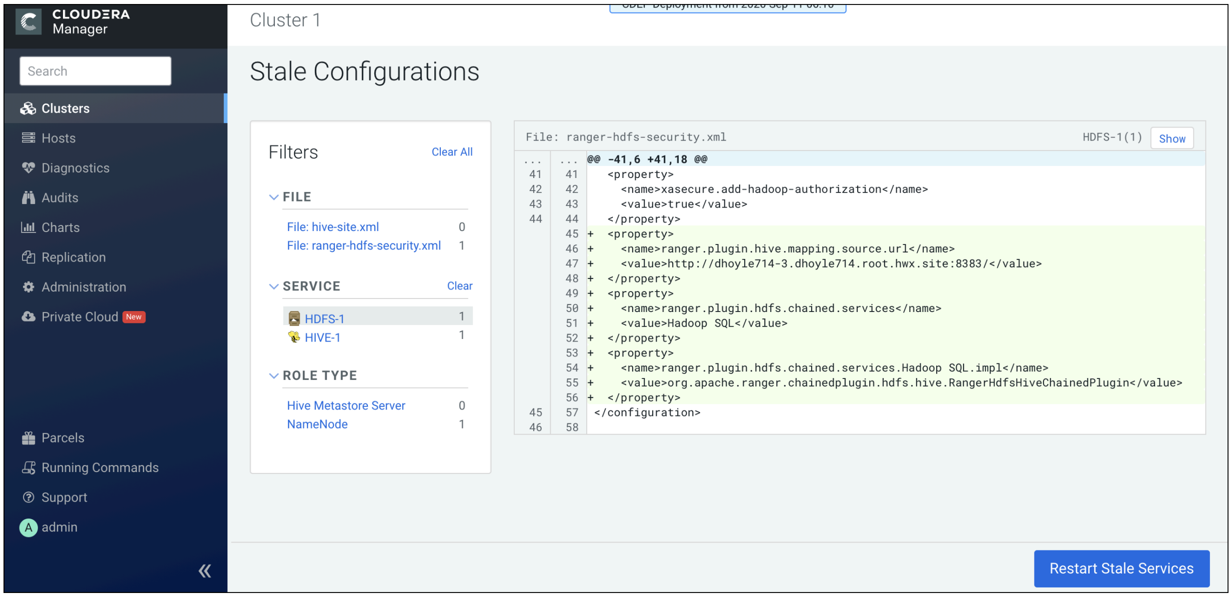
Note: The comma-separated lists that you define for hdfs privileged user names and service names are users that, based on default Hive policies, have all access permissions for all Hive resources. Therefore, for these users, checking Hive policies when they access storage locations which map to Hive resources is unnecessary, and may cause access violations if masking/row-filtering policies are configured for public group.



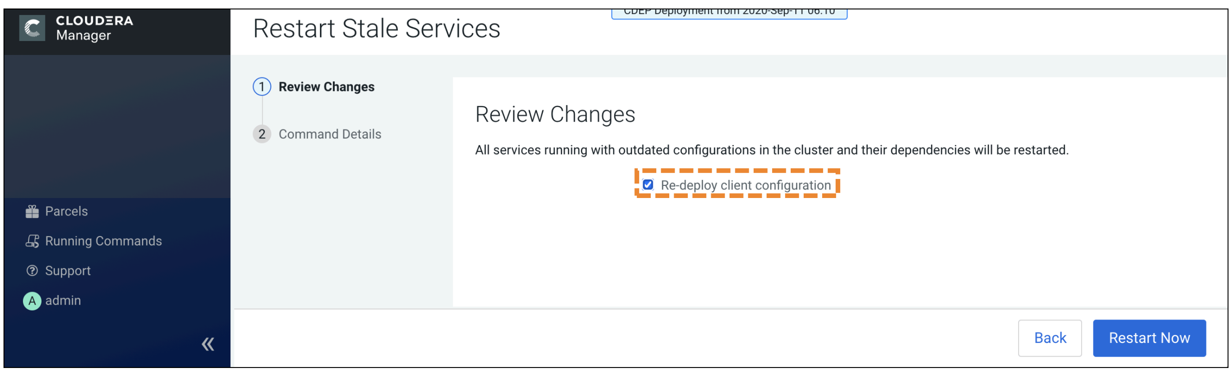
13. Click HDFS Restart.



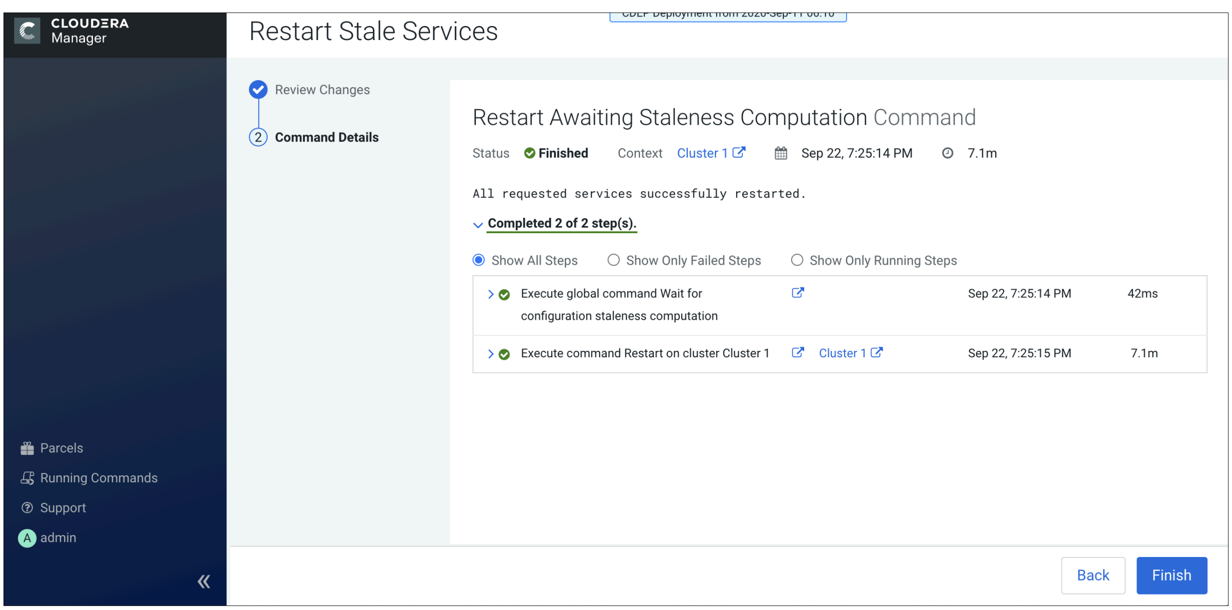
14. On the **Stale Configurations** page, click Restart Stale Services.



15. On the **Restart Stale Services** page, select the Re-deploy client configuration option, then click Restart Now.



16. Click Finish after the services restart.



Related Information

Configuring and Using Hive-HDFS ACL Sync

Custom Installation Solutions

Some installations may require custom solutions such as creating virtual images of cluster hosts, configuring a custom Java home location, or creating a Runtime cluster using a template.

Related Information

Cloudera Private Cloud Base Installation Guide

Privileged commands for Cloudera Manager installation

When installing Cloudera Manager using a non-root user, the sudo command is required to run specific commands with elevated privilege.

To restrict the sudo privilege specific commands, add the following list of commands to the sudoers file. You can find the sudoers file in /etc/sudoers by running the visudo command.

The sudoers file on the host running the Cloudera Manager Server must be modified as indicated in the Example configuration to add to the sudoers file. To install the Cloudera Manager Agent on CDP cluster hosts as a non root user, you must modify the sudoers file on each host. The user who is given sudo privilege must be the non-root user specified in the Cloudera Manager Add-Host Wizard when adding new hosts to a cluster.

Prerequisites and exceptions for the example configuration

Review the prerequisites and exceptions before adding the example configuration to the sudoers file.

Procedure

1. As a root user, you must create the repository file on the Cloudera Manager Server. For instance, as a root user you can create the /etc/yum.repos.d/cloudera-manager.repo file with the following content:

```
[cloudera-manager]
name=Cloudera Manager 7.x.0
baseurl=https://archive.cloudera.com/p/cm6/7.x.0/redhat7/yum/
gpgkey=https://archive.cloudera.com/p/cm6/7.x.0/redhat7/yum/RPM-GPG-KEY-
cloudera
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md
```

2. To enable Auto-TLS, use the Cloudera Manager user interface:
   a) Administration > Security > Enable Auto-TLS wizard
   b) For information on how to generate an internal CA and corresponding certificates, see Use case 1: Use CM to generate an internal CA and corresponding certificates
3. Database Configuration:
   a) To install the postgreSQL as the Cloudera Manager Database, the root user must run the following commands for the user1:

```
i) user1@cmsudo-1 ~]$ echo 'LC_ALL="en_US.UTF-8"' >> /etc/locale.conf
-bash: /etc/locale.conf file

ii) sudo su -l postgres -c "postgresql-setup initdb"
```

   b) To enable MD5 authentication, user1 must have root permission:

   Edit pg_hba.conf , which is usually found in /var/lib/pgsql/data or /etc/postgresql/<version>/main. For more information, see step 2, Enable MD5 authentication.

4. To install Ranger on any host and configure PostgreSQL database for Ranger, you need a new sudoers command list. For more information, see Configuring a PostgreSQL Database for Ranger or Ranger KMS
5. For the KDC setup, you must manage the krb5.conf file through Cloudera Manager user interface. For more information, see Enable Kerberos using the wizard
6. Cloudera Manager Upgrade:

   To set up the Cloudera Manager repository, user1 must have write access to the repository file. For more information, see Upgrading the Cloudera Manager Server
7. If you are setting up a KTS cluster, do the following as root user:
   a) You must obtain root access to install a file under "/etc/systemd/system/" as a prerequisite for installing rng-tools package. After installing the "rng-tools" package, user1 will require root user to run the following commands:

```
i) cp /usr/lib/systemd/system/rngd.service /etc/systemd/system/

ii) sed -i -e 's/ExecStart=\ /sbin/rngd -f /ExecStart=\ /sbin\ /rngd -f
 -r \ / dev \ / urandom/'/etc/systemd/system/rngd.service
```

8. Run the rsync command to copy the Key Trustee Server keys.

```
rsync -zav --exclude .ssl /var/lib/keytrustee/ .keytrustee cmsudo-6.vpc.
cloudera.com:/var/lib/keytrustee/
```

Example configuration to add to the sudoers file

The following can be used to provide root privileges to a non-root user to install Cloudera Manager server.

In the below case, user1 is a non-root user and is used to install and run Cloudera Manager server and agent on the Cloudera Manager server host.

Important: The following sudoers file is a suggestion. You can also use the other ways of choosing the user account with administrative access. The important part is the list of commands that must be allowed.

sudoers format for RHEL7 host where Cloudera Manager server is installed:

```
user1 ALL=(ALL) NOPASSWD:SETENV: /usr/bin/hostnamectl set-hostname *, /sbin/
iptables-save, /usr/bin/rpm --import *,
/usr/bin/yum install *cloudera*, /usr/bin/yum install ntp, /bin/wget, /bin/
systemctl status cloudera*,
/bin/systemctl disable firewalld, systemctl stop firewalld, /bin/systemctl
status ntp, /bin/systemctl enable ntp,
/usr/bin/vim /etc/ntp.conf, /usr/sbin/hwclock --systohc, /bin/tail, /usr/bin
/yum install java-1.8.0-openjdk-devel,
/usr/bin/yum install postgresql-server, /usr/bin/yum install python-pip, /us
r/bin/pip install psycopg2==2.7.5 --ignore-installed,
 /bin/systemctl enable postgresql, /bin/systemctl restart postgresql, /bin/
systemctl enable cloudera-scm-server,
/bin/yum -y install openjdk8.x86_64,  /bin/yum install krb5-workstation k
rb5-libs, /usr/bin/yum clean all,
/usr/bin/install rng-tools, /usr/bin/yum upgrade *cloudera*, /bin/systemctl
 restart cloudera*,
/bin/systemctl daemon-reload, /bin/systemctl start rngd, /bin/systemctl s
top rngd, /bin/systemctl status rngd,
/bin/systemctl/ enable rngd, /usr/bin/ktadmin init
```

sudoers format for RHEL7 Cloudera Manager agent installs through the Add Host Wizard of Cloudera Manager:

```
user1 ALL=(ALL) NOPASSWD:SETENV: /usr/bin/hostnamectl set-hostname *, /sbin/
iptables-save,
```

```
/usr/bin/rpm --import *, /usr/bin/yum --disablerepo=* --enablerepo=cloudera*
 *, /usr/bin/yum install ntp, /bin/wget,
/bin/systemctl status cloudera-scm-agent, /bin/systemctl status ntp, /bin/
systemctl enable ntp, /usr/bin/vim /etc/ntp.conf,
/usr/sbin/hwclock --systohc, /bin/id, /usr/bin/install -m 644 --backup=nu
mbered *, /usr/bin/rm -Rf /var/cache/yum/*,
/bin/cp /etc/cloudera-scm-agent/*, /usr/bin/sed -e * -i /etc/cloudera-scm-a
gent/*,
/usr/bin/tail -n * /var/log/cloudera-scm-agent/*, /usr/bin/mkdir -m 0755 -p
 /var/lib/cloudera-scm-agent/agent-cert,
/usr/bin/tar xf * -C /var/lib/cloudera-scm-agent/agent-cert, /bin/tail, /u
sr/bin/yum install java-1.8.0-openjdk-devel,
/usr/bin/yum install python-pip, /usr/bin/pip install psycopg2==2.7.5 --ig
nore-installed,
/bin/yum -y install openjdk8.x86_64, /bin/yum -y install *cloudera*, /bin/y
um install krb5-workstation krb5-libs,
/usr/bin/yum clean all, /usr/bin/yum upgrade *cloudera*, /bin/systemctl res
tart cloudera*, /usr/bin/ktadmin init
```

Note: The above sudoers configuration can be safely merged into a single configuration line for user1.

Note: In this example configuration, PostgreSQL Server has been used. However, the configuration might vary depending on the Database installed in your environment.

Creating Virtual Images of Cluster Hosts

You can create virtual machine images, such as PXE-boot images, Amazon AMIs, and Azure VM images of cluster hosts with pre-deployed Cloudera software that you can use to quickly spin up virtual machines.

You can create virtual machine images, such as PXE-boot images, Amazon AMIs, and Azure VM images of cluster hosts with pre-deployed Cloudera software that you can use to quickly spin up virtual machines. These images use parcels to install Runtime software. This topic describes the procedures to create images of the Cloudera Manager host and worker host and how to instantiate hosts from those images.

Creating a Pre-Deployed Cloudera Manager Host

Complete the steps below to create a Cloudera Manager virtual machine image.

Procedure

1. Instantiate a virtual machine image (an AMI, if you are using Amazon Web Services) based on a supported operating system and start the virtual machine. See the documentation for your virtualization environment for details.
2. Install Cloudera Manager and configure a database. You can configure either a local or remote database.
3. Wait for the Cloudera Manager Admin console to become active.
4. Log in to the Cloudera Manager Admin console.
5. Download any parcels for Runtime or other services managed by Cloudera Manager. Do not distribute or activate the parcels.

6. Log in to the Cloudera Manager server host:
   a) Run the following command to stop the Cloudera Manager service: service cloudera-scm-server stop
   b) Run the following command to disable autostarting of the cloudera-scm-server service:
      - RHEL 7.x /CentOS 7.x.x:

        ```
        systemctl disable cloudera-scm-server.service
        ```
      - Ubuntu:

        ```
        update-rc.d -f cloudera-scm-server remove
        ```
7. Create an image of the Cloudera Manager host.
8. If you installed the Cloudera Manager database on a remote host, also create an image of the database host.

   Note: Ensure that there are no clients using the remote database while creating the image.

Instantiating a Cloudera Manager Image

Complete the following steps to create a new Cloudera Manager instance from a virtual machine image.

Procedure

1. Instantiate the Cloudera Manager image.
2. If the Cloudera Manager database will be hosted on a remote host, also instantiate the database host image.
3. Ensure that the cloudera-scm-server service is not running by running the following command on the Cloudera Manager host:

   ```
   service cloudera-scm-server status
   ```

   If it is running, stop it using the following command:

   ```
   service cloudera-scm-server stop
   ```
4. On the Cloudera Manager host, create a file named uuid in the /etc/cloudera-scm-server directory. Add a globally unique identifier to this file using the following command:

   ```
   cat /proc/sys/kernel/random/uuid > /etc/cloudera-scm-server/uuid
   ```

   The existence of this file informs Cloudera Manager to reinitialize its own unique identifier when it starts.
5. Run the following command to start the Cloudera Manager service:

   ```
   service cloudera-scm-server start
   ```
6. Run the following command to enable automatic restart for the cloudera-scm-server:
   - SLES:

     ```
     chkconfig cloudera-scm-server on
     ```
   - RHEL 7.x /CentOS 7.x.x:

     ```
     systemctl enable cloudera-scm-server.service
     ```
   - Ubuntu:

     ```
     update-rc.d -f cloudera-scm-server defaults
     ```

Creating a Pre-Deployed Worker Host

Complete the steps below to create a pre-deployed worker host.

Procedure

1. Instantiate a virtual machine image (an AMI, if you are using Amazon Web Services) based on a supported operating system and start the virtual machine. See the documentation for your virtualization environment for details.
2. Download the parcels required for the worker host from the public parcel repository, or from a repository that you have created and save them to a temporary directory. See *Cloudera Manager 7 Download Information*.
3. From the same location where you downloaded the parcels, download the *PARCEL_NAME*.parcel.sha1 file for each parcel.
4. Calculate and compare the sha1 of the downloaded parcel to ensure that the parcel was downloaded correctly. For example:

```
sha1sum KAFKA-2.0.2-1.2.0.2.p0.5-el6.parcel | awk '{print $1}' > KAFKA-2
.0.2-1.2.0.2.p0.5-el6.parcel.sha

diff KAFKA-2.0.2-1.2.0.2.p0.5-el6.parcel.sha1 KAFKA-2.0.2-1.2.0.2.p0.5-el6
.parcel.sha
```

5. Unpack the parcel:
   a) Create the following directories:
      - /opt/cloudera/parcels
      - /opt/cloudera/parcel-cache
   b) Set the ownership for the two directories you just created so that they are owned by the username that the Cloudera Manager agent runs as.
   c) Set the permissions for each directory using the following command:

   ```
   chmod 755 DIRECTORY
   ```

   Note that the contents of these directories will be publicly available and can be safely marked as world-readable.
   d) Running as the same user that runs the Cloudera Manager agent, extract the contents of the parcel from the temporary directory using the following command:

   ```
   tar -zxvf PARCELFILE -C /opt/cloudera/parcels/
   ```

   e) Add a symbolic link from the product name of each parcel to the /opt/cloudera/parcels directory.

   For example, to link /opt/cloudera/parcels/CDH-6.0.0-1.cdh6.0.0.p0.309038 to /opt/cloudera/parcels/CDH, use the following command:

   ```
   ln -s /opt/cloudera/parcels/CDH-6.0.0-1.cdh6.0.0.p0.309038 /opt/cloudera
   /parcels/CDH
   ```

   f) Mark the parcels to not be deleted by the Cloudera Manager agent on start up by adding a .dont_delete marker file (this file has no contents) to each subdirectory in the /opt/cloudera/parcels directory. For example:

   ```
   touch /opt/cloudera/parcels/CDH/.dont_delete
   ```

6. Verify the file exists:

```
ls -l /opt/cloudera/parcels/PARCELNAME
```

You should see output similar to the following:

```
ls -al /opt/cloudera/parcels/CDH
total 100
drwxr-xr-x  9 root root  4096 Sep 14 14:53 .
drwxr-xr-x  9 root root  4096 Sep 14 06:34 ..
drwxr-xr-x  2 root root  4096 Sep 12 06:39 bin
-rw-r--r--  1 root root     0 Sep 14 14:53 .dont_delete
drwxr-xr-x 26 root root  4096 Sep 12 05:10 etc
drwxr-xr-x  4 root root  4096 Sep 12 05:04 include
drwxr-xr-x  2 root root 69632 Sep 12 06:44 jars
drwxr-xr-x 37 root root  4096 Sep 12 06:39 lib
drwxr-xr-x  2 root root  4096 Sep 12 06:39 meta
drwxr-xr-x  5 root root  4096 Sep 12 06:39 share
```

7. Install the Cloudera Manager agent. If you have not already done so, *Step 1: Configure a Repository for Cloudera Manager*.
8. Create an image of the worker host. See the documentation for your virtualization environment for details.

Instantiating a worker host

You must instantiate a worker host from the pre-deployed image that you have created.

Procedure

1. Instantiate the virtual machine image for the Cloudera worker host.
2. Edit the /etc/cloudera-scm-agent/config.ini file and set the server_host and server_port properties to reference the Cloudera Manager server host.
3. Configure TLS/SSL.
4. Start the Cloudera Manager agent service:

```
service cloudera-scm-agent start
```

Manually Install Cloudera Software Packages

This topic shows how to manually install Cloudera Manager packages. Package installations of Cloudera Runtime are not supported in Cloudera Private Cloud Base .

Before manual installation, you must configure a repository. See Step 1: Configure a Repository for Cloudera Manager on page 143.

Install Cloudera Manager Packages

Cloudera Manager is installed on the Cloudera Manager Server host using packages.

Procedure

1. On the Cloudera Manager Server host, type the following commands to install the Cloudera Manager packages:

OS	Command
RHEL	`sudo yum install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server`
SLES	`sudo zypper install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server`
Ubuntu	`sudo apt-get install cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server`

2. If you are using an Oracle database for Cloudera Manager Server, edit the /etc/default/cloudera-scm-server file on the Cloudera Manager server host. Locate the line that begins with export CMF_JAVA_OPTS and change the -Xmx2G option to -Xmx4G.

Manually Install Cloudera Manager Agent Packages

The Cloudera Manager Agent is responsible for starting and stopping processes, unpacking configurations, triggering installations, and monitoring all hosts in a cluster. You can install the Cloudera Manager agent manually on all hosts, or Cloudera Manager can install the Agents in a later step. To use Cloudera Manager to install the agents, skip this section.

About this task

To install the Cloudera Manager Agent packages manually, do the following on every cluster host (including those that will run one or more of the Cloudera Management Service roles: Service Monitor, Activity Monitor, Event Server, Alert Publisher, or Reports Manager):

Note: You must make sure that you set the Java home path correctly while installing the agents manually.

Add the Java home path in Cloudera Manager under Hosts > All Hosts > Configuration > Java Home Directory. Later, start the service.

Procedure

1. Use one of the following commands to install the Cloudera Manager Agent packages:

OS	Command
RHEL, if you have a yum repo configured:	`$ sudo yum install cloudera-manager-agent cloudera-manager-daemons`
RHEL, if you're manually transferring RPMs:	`$ sudo yum --nogpgcheck localinstall cloudera-manager-agent-package.*.x86_64.rpm cloudera-manager-daemons.*.x86_64.rpm`
SLES	`$ sudo zypper install cloudera-manager-agent cloudera-manager-daemons`
Ubuntu	`$ sudo apt-get install cloudera-manager-agent cloudera-manager-daemons`

2. On every cluster host, configure the Cloudera Manager Agent to point to the Cloudera Manager Server by setting the following properties in the /etc/cloudera-scm-agent/config.ini configuration file:

Property	Description
server_host	Name of the host where Cloudera Manager Server is running.
server_port	Port on the host where Cloudera Manager Server is running.

3. Start the Agents by running the following command on all hosts:

```
sudo systemctl start cloudera-scm-agent
```

If the agent starts without errors, no response displays.

When the Agent starts, it contacts the Cloudera Manager Server. If communication fails between a Cloudera Manager Agent and Cloudera Manager Server, see *Troubleshooting Installation Problems*. When the Agent hosts reboot, cloudera-scm-agent starts automatically.

Installation Reference

Reference information related to Cloudera Private Cloud Base installation.

Related Information

Cloudera Private Cloud Base Installation Guide

Ports

Cloudera Manager, Cloudera Runtime components, managed services, and third-party components use the ports listed in the tables that follow.

Before you deploy Cloudera Manager, Cloudera Runtime, managed services, and third-party components, make sure these ports are open on each system. If you are using a firewall, such as iptables or firewalld, and cannot open all the listed ports, you must disable the firewall completely to ensure full functionality.

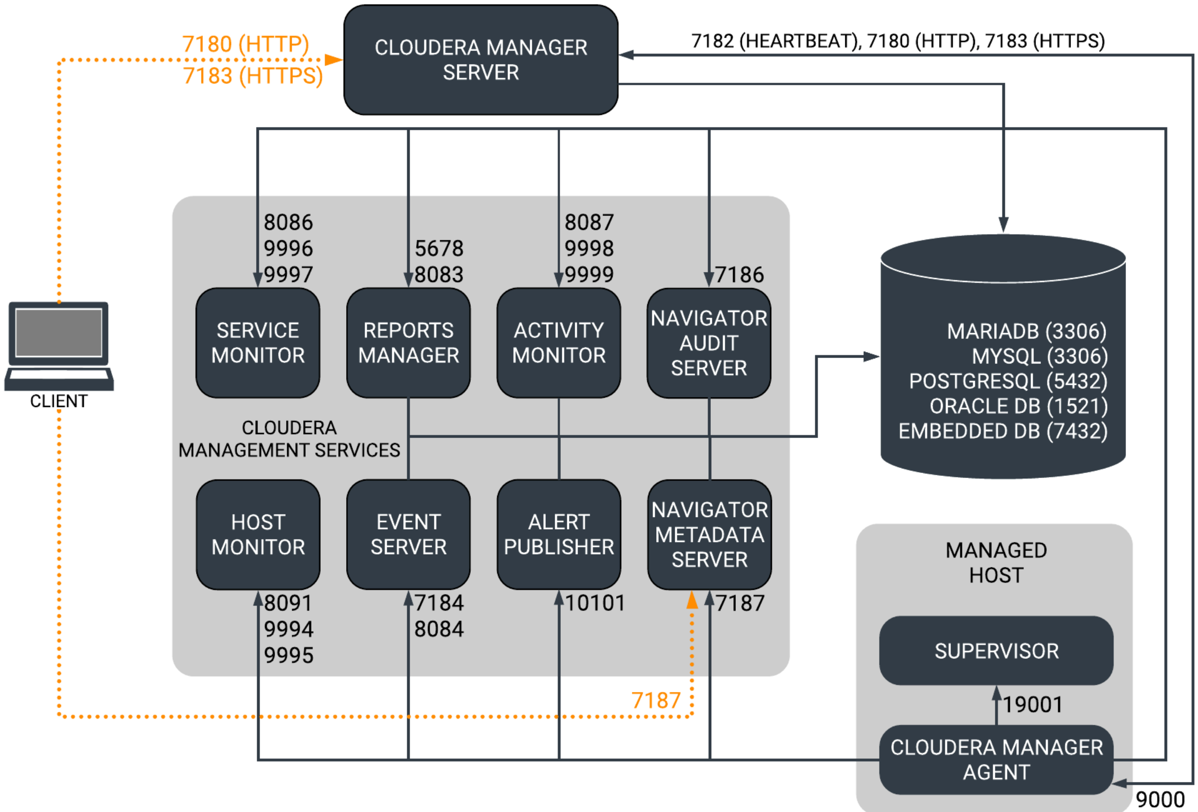
In the tables in the subsections that follow, the Access Requirement column for each port is usually either "Internal" or "External." In this context, "Internal" means that the port is used only for communication among the components (for example the JournalNode ports in an HA configuration); "External" means that the port can be used for either internal or external communication (for example, ports used by NodeManager and the JobHistory Server Web UIs).

Unless otherwise specified, the ports access requirement is unidirectional, meaning that inbound connections to the specified ports must be allowed. In most modern stateful firewalls, it is not necessary to create a separate rule for return traffic on a permitted session.

Ports Used by Cloudera Manager

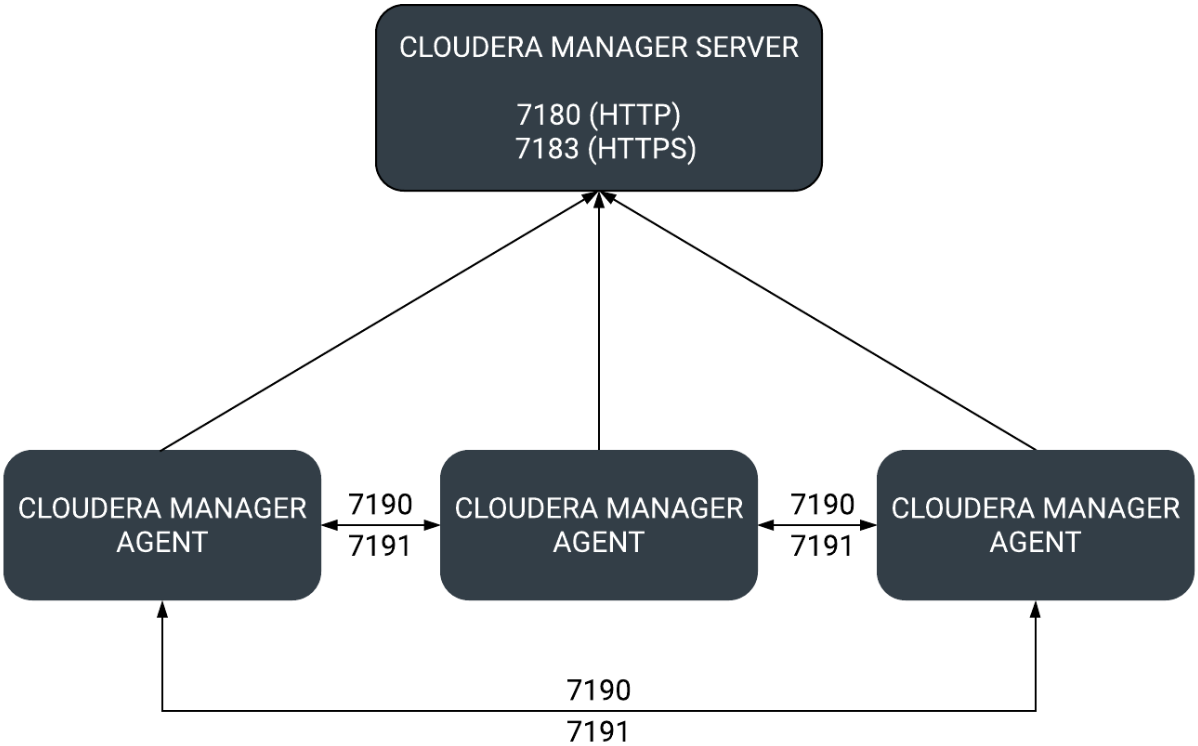
The diagrams and tables below provide an overview of some of the ports used by Cloudera Manager and Cloudera Management Service roles.

Figure 3: Ports Used by Cloudera Manager



When peer-to-peer distribution is enabled for parcels, the Cloudera Manager Agent can obtain the parcel from the Cloudera Manager Server or from other agents, as follows:

Figure 4: Ports Used in Peer-to-Peer Parcel Distribution



For further details, see the following tables. All ports listed are TCP.

In the following tables, Internal means that the port is used only for communication among the components; External means that the port can be used for either internal or external communication.

Table 38: External Ports

Component	Service	Port	Configuration	Description
Cloudera Manager Server	HTTP (Web UI)	7180	AdministrationSettingsCategoryPorts and AddressesHTTP Port for Admin Console	HTTP port used by the web console.
	HTTPS (Web UI)	7183	AdministrationSettingsCategoryPorts and AddressesHTTPS Port for Admin Console	Port used by the web console if HTTPS is enabled. If enabled, port 7180 remains open, but redirects all requests to HTTPS on port 7183.
Cloudera Manager Agent	HTTP (Debug)	9000	/etc/cloudera-scm-agent/config.ini	
Backup and Disaster Recovery	HTTP (Web UI)	7180	AdministrationSettingsCategoryPorts and AddressesHTTP Port for Admin Console	Used for communication to peer (source) Cloudera Manager.
	HTTPS (Web UI)	7183	AdministrationSettingsCategoryPorts and AddressesHTTPS Port for Admin Console	Used for communication to peer (source) Cloudera Manager when HTTPS is enabled.

Component	Service	Port	Configuration	Description
	HDFS NameNode	8020	HDFS serviceConfigurationCategoryPorts and AddressesNameNode Port	HDFS and Hive/ Impala replication: communication from destination HDFS and MapReduce hosts to source HDFS NameNode(s). Hive/ Impala Replication: communication from source Hive hosts to destination HDFS NameNode(s).
	HDFS DataNode	9866	HDFS serviceConfigurationCategoryPorts and AddressesDataNode Transceiver Port	HDFS and Hive/ Impala replication: communication from destination HDFS and MapReduce hosts to source HDFS DataNode(s). Hive/ Impala Replication: communication from source Hive hosts to destination HDFS DataNode(s).
Telemetry Publisher	HTTP	10110	ClustersCloudera Management ServiceCategoryPorts and AddressesTelemetry Publisher Server Port	The port where the Telemetry Publisher Server listens for requests
Telemetry Publisher	HTTP (Debug)	10111	ClustersCloudera Management ServiceCategoryPorts and AddressesTelemetry Publisher Web UI Port	The port where Telemetry Publisher starts a debug web server. Set to -1 to disable debug server.

Table 39: Internal Ports

Component	Service	Port	Configuration	Description
Cloudera Manager Server	Avro (RPC)	7182	AdministrationSettingsCategoryPorts and AddressesAgent Port to connect to Server	Used for Agent to Server heartbeats
	Embedded PostgreSQL database	7432		The optional embedded PostgreSQL database used for storing configuration information for Cloudera Manager Server.
	Peer-to-peer parcel distribution	7190, 7191	HostsAll HostsConfigurationP2P Parcel Distribution Port	Used to distribute parcels to cluster hosts during installation and upgrade operations.
Cloudera Manager Agent	HTTP (Debug)	The value set for the listening_port parameter in the /etc/cloudera-scm-agent/config.ini file, plus 1.	Not directly configurable. For example, the default external port is 9000. Therefore the default internal port is 9001.	
	supervisord	19001	For example, the default internal port is 19001. You can configure the custom port under /etc/cloudera-scm-agent/config.ini by updating supervisord_port.	This port is used to start the supervisord process.

Component	Service	Port	Configuration	Description
Event Server	Custom protocol	7184	Cloudera Management ServiceConfigurationCategoryPorts and AddressesEvent Publish Port	Port on which the Event Server listens for the publication of events.
	Custom protocol	7185	Cloudera Management ServiceConfigurationCategoryPorts and AddressesEvent Query Port	Port on which the Event Server listens for queries for events.
	HTTP (Debug)	8084	Cloudera Management ServiceConfigurationCategoryPorts and AddressesEvent Server Web UI Port	Port for the Event Server's Debug page. Set to -1 to disable debug server.
Alert Publisher	Custom protocol	10101	Cloudera Management ServiceConfigurationCategoryPorts and AddressesAlerts: Listen Port	Port where the Alert Publisher listens for internal API requests.
Service Monitor	HTTP (Debug)	8086	Cloudera Management ServiceConfigurationCategoryPorts and AddressesService Monitor Web UI Port	Port for Service Monitor's Debug page. Set to -1 to disable the debug server.
	HTTPS (Debug)		Cloudera Management ServiceConfigurationCategoryPorts and AddressesService Monitor Web UI HTTPS Port	Port for Service Monitor's HTTPS Debug page.
	Custom protocol	9997	Cloudera Management ServiceConfigurationCategoryPorts and AddressesService Monitor Listen Port	Port where Service Monitor is listening for agent messages.
	Internal query API (Avro)	9996	Cloudera Management ServiceConfigurationCategoryPorts and AddressesService Monitor Nozzle Port	Port where Service Monitor's query API is exposed.
Activity Monitor	HTTP (Debug)	8087	Cloudera Management ServiceConfigurationCategoryPorts and AddressesActivity Monitor Web UI Port	Port for Activity Monitor's Debug page. Set to -1 to disable the debug server.
	HTTPS (Debug)		Cloudera Management ServiceConfigurationCategoryPorts and AddressesActivity Monitor Web UI HTTPS Port	Port for Activity Monitor's HTTPS Debug page.
	Custom protocol	9999	Cloudera Management ServiceConfigurationCategoryPorts and AddressesActivity Monitor Listen Port	Port where Activity Monitor is listening for agent messages.
	Internal query API (Avro)	9998	Cloudera Management ServiceConfigurationCategoryPorts and AddressesActivity Monitor Nozzle Port	Port where Activity Monitor's query API is exposed.
Host Monitor	HTTP (Debug)	8091	Cloudera Management ServiceConfigurationCategoryPorts and AddressesHost Monitor Web UI Port	Port for Host Monitor's Debug page. Set to -1 to disable the debug server.
	HTTPS (Debug)	9091	Cloudera Management ServiceConfigurationCategoryPorts and AddressesHost Monitor Web UI HTTPS Port	Port for Host Monitor's HTTPS Debug page.
	Custom protocol	9995	Cloudera Management ServiceConfigurationCategoryPorts and AddressesHost Monitor Listen Port	Port where Host Monitor is listening for agent messages.

Component	Service	Port	Configuration	Description
	Internal query API (Avro)	9994	Cloudera Management ServiceConfigurationCategoryPorts and AddressesHost Monitor Nozzle Port	Port where Host Monitor's query API is exposed.
Reports Manager	Queries (Thrift)	5678	Cloudera Management ServiceConfigurationCategoryPorts and AddressesReports Manager Server Port	The port where Reports Manager listens for requests.
	HTTP (Debug)	8083	Cloudera Management ServiceConfigurationCategoryPorts and AddressesReports Manager Web UI Port	The port where Reports Manager starts a debug web server. Set to -1 to disable debug server.

Ports Used by Cloudera Navigator Key Trustee Server

The Cloudera Navigator Key Trustee Server uses certain ports to store and retrieve encryption information and information required for high availability.

All ports listed are TCP.

In the following table, the Access Requirement column for each port is usually either "Internal" or "External." In this context, "Internal" means that the port is used only for communication among the components; "External" means that the port can be used for either internal or external communication.

Component	Service	Port	Access Requirement	Configuration	Comment
Cloudera Navigator Key Trustee Server	HTTPS (key management)	11371	External	Key Trustee Server serviceConfigurationCategoryPorts and AddressesKey Trustee Server Port	Navigator Key Trustee Server clients (including Key Trustee KMS and Navigator Encrypt) access this port to store and retrieve encryption keys.
	PostgreSQL database	11381	External	Key Trustee Server serviceConfigurationCategoryPorts and AddressesKey Trustee Server Database Port	The Navigator Key Trustee Server database listens on this port. The Passive Key Trustee Server connects to this port on the Active Key Trustee Server for replication in Cloudera Navigator Key Trustee Server High Availability.

Ports Used by Cloudera Runtime Components

Cloudera Runtime components use a number of ports for associated services.

All ports listed are TCP.

In the following tables, Internal means that the port is used only for communication among the components; External means that the port can be used for either internal or external communication.

Table 40: External Ports

Component	Service	Port	Configuration	Comment
Apache Atlas	Non-SSL	31000	atlas.server.http.port	
	SSL	31443	atlas.server.https.port	This port is used only when Atlas is in SSL mode.
Apache Hadoop HDFS	DataNode	9866	dfs.datanode.address	DataNode server address and port for data transfer.
		9864	dfs.datanode.http.address	DataNode HTTP server port.
		9865	dfs.datanode.https.address	DataNode HTTPS server port.
		9867	dfs.datanode.ipc.address	DataNode IPC server port.

Component	Service	Port	Configuration	Comment
	NameNode	8020	fs.default.name or fs.defaultFS	fs.default.name is deprecated (but still works)
		8022	dfs.namenode.servicerpc-address	Optional port used by HDFS daemons to avoid sharing the RPC port used by clients (8020). Cloudera recommends using port 8022.
		9870	dfs.http.address or dfs.namenode.http-address	dfs.http.address is deprecated (but still works)
		9871	dfs.https.address or dfs.namenode.https-address	dfs.https.address is deprecated (but still works)
	NFS gateway	2049		nfs port (nfs3.server.port)
		4242		mountd port (nfs3.mountd.port)
		111		portmapper or rpcbind port.
		50079	nfs.http.port	The NFS gateway daemon uses this port to serve metrics. The port is configurable on versions 5.10 and higher.
		50579	nfs.https.port	The NFS gateway daemon uses this port to serve metrics. The port is configurable on versions 5.10 and higher.
	HttpFS	14000		HttpFS server port
		14001		HttpFS admin port
Apache Hadoop YARN (MRv2)	ResourceManager	8032	yarn.resourcemanager.address	
		8033	yarn.resourcemanager.admin.address	
		8088	yarn.resourcemanager.webapp.address	
		8090	yarn.resourcemanager.webapp.https.address	
	NodeManager	8042	yarn.nodemanager.webapp.address	
		8044	yarn.nodemanager.webapp.https.address	
	JobHistory Server	19888	mapreduce.jobhistory.webapp.address	
		19890	mapreduce.jobhistory.webapp.https.address	
	ApplicationMaster			The ApplicationMaster serves an HTTP service using an ephemeral port that cannot be restricted. This port is never accessed directly from outside the cluster by clients. All requests to the ApplicationMaster web server is routed using the YARN ResourceManager (proxy service). Locking down access to ephemeral port ranges within the cluster's network might restrict your access to the ApplicationMaster UI and its logs, along with the ability to look at running applications.
Apache HBase	Master	16000	hbase.master.port	IPC
		16010	hbase.master.info.port	HTTP
	RegionServer	16020	hbase.regionserver.port	IPC
		16030	hbase.regionserver.info.port	HTTP

Component	Service	Port	Configuration	Comment
	REST	20550	hbase.rest.port	The default REST port in HBase is 8080. Because this is a commonly used port, Cloudera Manager sets the default to 20550 instead.
	REST UI	8085	hbase.rest.info.port	
	HBase Thrift Server	9090	hbase.regionserver.thrift.port	
	HBase Thrift Serve Web UIr	9095		
	Lily HBase Indexer	11060		
Apache Hive	Metastore	9083		
	HiveServer2	10000	hive.server2.thrift.port	The Beeline command interpreter requires that you specify this port on the command line. If you use Oracle database, you must manually reserve this port.
	HiveServer2 Web User Interface (UI)	10002	hive.server2.webui.port in hive-site.xml	
Hue	Server	8888		
	Load Balancer	8889		
Apache Impala	Impala Daemon	21000		Used to transmit commands and receive results by impala-shell and version 1.2 of the Cloudera ODBC driver.
		21050		Used to transmit commands and receive results by applications, such as Business Intelligence tools, using JDBC, the Beeswax query editor in Hue, and version 2.0 or higher of the Cloudera ODBC driver.
		25000		Impala web interface for administrators to monitor and troubleshoot.
		28000		Used to transmit commands and receive results by client applications over HTTP through the HiveServer2 protocol.
	StateStore Daemon	25010		StateStore web interface for administrators to monitor and troubleshoot.
	Catalog Daemon	25020		Catalog service web interface for administrators to monitor and troubleshoot.
Apache Kafka	Kafka Broker	9092	port	The primary communication port used by producers and consumers; also used for inter-broker communication.
		9093	ssl_port	A secured communication port used by producers and consumers; also used for inter-broker communication.
	Kafka Connect	28083	rest.port	Kafka Connect Rest Port.
		28085	secure.rest.port	Kafka Connect Secure Rest Port
		28084	metrics.jetty.server.port	Jetty Metrics Port
		28087	metrics.jetty.server.secureport	Secure Jetty Metrics Port

Component	Service	Port	Configuration	Comment
Apache Knox	Knox Gateway	8443	gateway.port	The HTTPS port for the Gateway
	Knox Gateway (HTTPS)	8444	idbroker_gateway_port	
Apache Kudu	Master	7051		Kudu Master RPC port.
		8051		Kudu Master HTTP server port.
	TabletServer	7050		Kudu TabletServer RPC port.
		8050		Kudu TabletServer HTTP server port.
Apache Oozie	Oozie Server	11000	OOZIE_HTTP_PORT in oozie-env.sh	HTTP
		11443		HTTPS
Apache Ozone	Ozone Manager	9862	ozone.om.rpc-port	RPC endpoint for clients and applications.
		9874	ozone.om.http-port	HTTP port for the Ozone Manager web UI.
		9875	ozone.om.https-port	HTTPS port for the Ozone Manager web UI.
	Storage Container Manager	9876	ozone.scm.http-port	HTTP port for the SCM UI.
		9877	ozone.scm.https-port	HTTPS port for the SCM web UI.
	DataNode	9882	hdds.datanode.http-address	HTTP port for the DataNode web UI.
		9883	hdds.datanode.https-address	HTTPS port for the DataNode web UI.
		9858	dfs.container.ratis.ipc	RAFT server endpoint that is used by clients and other DataNodes to replicate RAFT transactions and write data.
		9859	dfs.container.ipc	Endpoint that is used by clients and other DataNodes to read block data.
	S3 Gateway	9878	ozone.s3g.http-port	HTTP port for the S3 API REST endpoint and web UI.
		9879	ozone.s3g.https-port	HTTPS port for the S3 API REST endpoint and web UI.
	Recon Service	9891	ozone.recon.rpc-port	Port used by DataNodes to communicate with the Recon Server.
		9888	ozone.recon.http-port	HTTP port for the Recon service web UI and REST API.
		9889	ozone.recon.https-port	HTTPS port for the Recon service web UI and REST API.
Apache Ranger	Non-SSL	6080	ranger.service.http.port	
	SSL	6182	ranger.service.https.port	This port is used only when Ranger is in SSL mode.
	Admin Unix Auth Service Port	5151	ranger.unixauth.service.port	
	Usersync HTTP Port	8280	ranger.usersync.http.port	HTTP port for Ranger Usersync
	Usersync HTTPS Port	8283	ranger.usersync.https.port	HTTPS port for Ranger Usersync
	Tagsync HTTP Port	8180	ranger.tagsync.http.port	HTTP port for Ranger Tagsync
	Tagsync HTTPS Port Port	8183	ranger.tagsync.https.port	HTTPS port for Ranger Tagsync
Ranger KMS	Ranger KMS nodes	9292	ranger.service.http.port	HTTP port for Ranger KMS.

Component	Service	Port	Configuration	Comment
	Ranger KMS nodes	9494	ranger.service.https.port	HTTPS port for Ranger KMS. Only used when SSL is enabled for Ranger KMS.
Ranger RMS	Ranger RMS nodes	8383	ranger.service.http.port	HTTP port for Ranger RMS.
	Ranger RMS nodes	8484	ranger.service.https.port	HTTPS port for Ranger RMS. Only used when SSL is enabled for Ranger RMS.
Apache Solr	Solr Server	8983		HTTP port for all Solr-specific actions, update/query.
	Solr Server	8985		HTTPS port for all Solr-specific actions, update/query.
Apache Spark	Shuffle service	7337	spark.shuffle.service.port	Port on which the Spark external shuffle service runs.
	History Server	18088	spark.history.ui.port	HTTP port for the Spark History Server WebUI.
	History Server with TLS	18488	spark.ssl.historyServer.port	HTTPS port for Spark History Server WebUI. Only used when SSL is enabled for Spark History Server.
Apache Sqoop	Metastore	16000	sqoop.metastore.server.port	
Apache Zeppelin	Zeppelin Server	8885	zeppelin.server.port	
	Zeppelin Server (SSL)	8886	zeppelin.server.ssl.port	
Apache ZooKeeper	Server (with Cloudera Runtime or Cloudera Manager)	2181	clientPort	Client port.
	Server (with Cloudera Runtime or Cloudera Manager) (SSL)	2182	secureClientPort	Secure client port.
Cruise Control	Cruise Control Server	8899	webserver.http.port	This is the main port that enables access to the Cruise Control Server
Livy	Livy Server Web UI	8998	livy.server.port	
	Livy Thrift Server	10090	livy.server.thrift.port	
Omid	TSO Server	54758		
Schema Registry	Schema Registry Server	7788	schema.registry.port	REST endpoint for Schema Registry.
		7789	schema.registry.adminPort	Page for monitoring the Schema Registry service to determine for example the health state and CPU usage.
		7790	schema.registry.ssl.port	When SSL is enabled, REST endpoint for Schema Registry.
		7791	schema.registry.ssl.adminPort	When SSL is enabled, the page for monitoring the Schema Registry service to determine for example the health state and CPU usage.
Streams Messaging Manager	Streams Messaging Manager Rest Admin Server	8585	streams.messaging.manager.port	Streams Messaging Manager Port
		8587	streams.messaging.manager.ssl.port	Streams Messaging Manager Port (SSL)
		8586	streams.messaging.manager.adminPort	Streams Messaging Manager Admin Port

Component	Service	Port	Configuration	Comment
		8588	streams.messaging.manager.ssl.adminPort	Streams Messaging Manager Admin Port (SSL)
	Streams Messaging Manager UI Server	9991	streams.messaging.manager.ui.port	The port on which server accepts connections. This port is used for both secured and unsecured connections.
Streams Replication Manager	SRM Service	6670	streams.replication.manager.service.port	SRM Service port.
		6671	streams.replication.manager.service.ssl.port	SRM Service port when SSL is enabled.

Table 41: Internal Ports

Component	Service	Port	Configuration	Comment
Apache Hadoop HDFS	Secondary NameNode	9868	dfs.secondary.http.address or dfs.namenode .secondary.http-address	dfs.secondary.http.addressis deprecated (but still works)
		9869	dfs.secondary.https.address	
	JournalNode	8485	dfs.namenode.shared.edits.dir	
		8480	dfs.journalnode.http-address	
		8481	dfs.journalnode.https-address	
	Failover Controller	8019		Used for NameNode HA
Apache Hadoop YARN (MRv2)	ResourceManager	8030	`yarn.resourcemanager .scheduler.address`	
		8031	yarn.resourcemanager.resource-tracker.ad dress	
	NodeManager	8040	yarn.nodemanager.localizer.address	
		8041	yarn.nodemanager.address	
	JobHistory Server	10020	mapreduce.jobhistory.address	
		10033	mapreduce.jobhistory.admin.address	
	Shuffle HTTP	13562	mapreduce.shuffle.port	
	Queue Manager	8082	queuemanager_webapp_port	
	Config Store/Service	8080	Set this configuration in the config.yml file for the service.	Reconfiguring this in a production environment is not recommended.
	Queue Manager Config-Service	8081	adminConnectorsPort	Set this configuration in the config.yml file for the service.
Apache Hadoop KMS	Key Management Server	16001	kms_admin_port	Applies to both Java KeyStore KMS and Key Trustee KMS.
Apache HBase	HQuorumPeer	2181	hbase.zookeeper.property.clientPort	HBase-managed ZooKeeper mode
		2888	hbase.zookeeper.peerport	HBase-managed ZooKeeper mode
		3888	hbase.zookeeper.leaderport	HBase-managed ZooKeeper mode
Apache Impala	Impala Daemon	23000		Internal use only. Impala daemons listen on this port for updates from the statestore daemon.

Component	Service	Port	Configuration	Comment
		27000		Internal use only. Impala daemons use this port for KRPC based communication with each other.
	StateStore Daemon	24000		Internal use only. The statestore daemon listens on this port for registration/unregistration requests.
	Catalog Daemon	23020		Internal use only. The catalog daemon listens on this port for updates from the statestore daemon.
		26000		Internal use only. The catalog service uses this port to communicate with the Impala daemons.
Apache Kafka	Kafka Broker	9092	port	The primary communication port used by producers and consumers; also used for inter-broker communication.
		9093	ssl_port	A secured communication port used by producers and consumers; also used for inter-broker communication.
		9393	jmx_port	Internal use only. Used for administration via JMX.
		9394	kafka.http.metrics.port	Internal use only. This is the port via which the HTTP metric reporter listens. It is used to retrieve metrics through HTTP instead of JMX.
	Kafka MirrorMaker	24042	jmx_port	Internal use only. Used to administer the producer and consumer of the MirrorMaker.
Apache Ozone	Ozone Manager	9872	ozone.om.ratis-port	RPC endpoint for Ozone Manager HA instances to form a RAFT consensus ring.
	Storage Container Manager	9861	ozone.scm.datanode.port	Port used by the DataNodes to communicate with the Storage Container Manager (SCM).
		9863	ozone.scm.block.client.port	Port used by the Ozone Manager to communicate with the SCM for block related operations.
		9860	ozone.scm.client.port	Port used by the Ozone Manager and other clients to communicate with the SCM for container operations.
		9894	ozone.scm.ratis.port	Port used by the SCM to communicate with other SCMs using Ratis.
		9895	ozone.scm.grpc.port	Port used by the SCM to communicate with other SCMs about the database checkpoint downloads.
Apache Phoenix	Phoenix Query Server Port	8765	phoenix.queryserver.http.port	
Apache Solr	Solr Server	8993		Infra-Solr HTTP port
	Solr Server	8995		Infra-Solr HTTPS port
Apache ZooKeeper	Server (with Cloudera Runtime only)	2888	X in server.N =host:X:Y	Peer
	Server (with Cloudera Runtime only)	3888	X in server.N =host:X:Y	Peer

Component	Service	Port	Configuration	Comment
	Server (with Cloudera Runtime and Cloudera Manager)	3181	X in server.N =host:X:Y	Peer
	Server (with Cloudera Runtime and Cloudera Manager)	4181	X in server.N =host:X:Y	Peer
	ZooKeeper JMX port	9010		ZooKeeper will also use another randomly selected port for RMI. To allow Cloudera Manager to monitor ZooKeeper, you must do one of the following: • Open up all ports when the connection originates from the Cloudera Manager Server • Do the following: **1.** Open a non-ephemeral port (such as 9011) in the firewall. **2.** Install Oracle Java 7u4 JDK or higher. **3.** Add the port configuration to the advanced configuration snippet, for example: -Dcom.sun.management.jmxremote.rmi.port=9011 **4.** Restart ZooKeeper.

Ports Used by DistCp

DistCp uses various ports for HDFS and HttpFS services.

All ports listed are TCP.

In the following table, the Access Requirement column for each port is usually either "Internal" or "External." In this context, "Internal" means that the port is used only for communication among the components; "External" means that the port can be used for either internal or external communication.

Component	Service	Qualifier	Port	Access Requirement	Configuration	Comment
Hadoop HDFS	NameNode		8020	External	`fs.default.name` or `fs.defaultFS`	`fs.default.name` is deprecated (but still works)
	DataNode	Secure	1004	External	`dfs.datanode.address`	
	DataNode		50010	External	`dfs.datanode.address`	
WebHDFS	NameNode		50070	External	`dfs.http.address` or `dfs.namenode.http-address`	`dfs.http.address` is deprecated (but still works)
	DataNode	Secure	1006	External	`dfs.datanode.http.address`	
HttpFS	web		14000			

Ports Used by Third-Party Components

Third-party components such as PostgreSQL and LDAP use a number of ports for associated services.

In the following table, the Access Requirement column for each port is usually either "Internal" or "External." In this context, "Internal" means that the port is used only for communication among the components; "External" means that the port can be used for either internal or external communication.

Component	Service	Qualifier	Port	Protocol	Access Requirement	Configuration	Comment
Ganglia	ganglia-gmond		8649	UDP/TCP	Internal		

Component	Service	Qualifier	Port	Protocol	Access Requirement	Configuration	Comment
	ganglia-web		80	TCP	External	Via Apache `httpd`	
Kerberos	KRB5 KDC Server	Secure	88	UDP/TCP	External	`kdc_ports` and `kdc_tcp_ports` in either the `[kdcdefaults]` or `[realms]` sections of `kdc.conf`	By default only UDP
	KRB5 Admin Server	Secure	749	TCP	External	`kadmind_port` in the `[realms]` section of `kdc.conf`	
	kpasswd		464	UDP/TCP	External		
SSH	ssh		22	TCP	External		
PostgreSQL			5432	TCP	Internal		
MariaDB			3306	TCP	Internal		
MySQL			3306	TCP	Internal		
LDAP	LDAP Server		389	TCP	External		
	LDAP Server over TLS/SSL	TLS/SSL	636	TCP	External		
	Global Catalog		3268	TCP	External		
	Global Catalog over TLS/SSL	TLS/SSL	3269	TCP	External		

Service Dependencies in Cloudera Manager

The following tables list service dependencies that exist between various services in a Cloudera Manager deployment.

When configuring CDP Runtime for production environments, be sure that Kerberos is enabled for user authentication. Cloudera supports security services such as Ranger and Atlas when they run on clusters where Kerberos is enabled to authenticate users.

Service dependencies for Spark 2 on YARN and Cloudera Data Science Workbench are listed separately.

Table 42: Service Dependencies

Service	Dependencies	Optional Dependencies
ADLS Connector		
Atlas	• HDFS • HBase • Kafka (Kafka broker role only) • Solr	Ranger
Cruise Control	• Kafka • Zookeeper	
Data Context Connector		
HBase	• HDFS • ZooKeeper	• Atlas • Ranger
HDFS		• ADLS Connector or S3 Connector • KMS, Thales KMS, Key Trustee, or Luna KMS • Ranger • ZooKeeper
Hive	HDFS	• Atlas • HBase • Kudu • Ranger • Spark on YARN • YARN • ZooKeeper
Hive-on-Tez	• HDFS • Hive • Tez	• Atlas • HBase • Ranger • YARN • ZooKeeper
Hue	• HDFS • Hive	• Atlas • HBase • Hive-on-Tez • Impala • Oozie • Solr • ZooKeeper
Impala	• HDFS • Hive	• Atlas • HBase • Kudu • Ranger • YARN • ZooKeeper
Kafka	ZooKeeper	• HDFS • Ranger

Service	Dependencies	Optional Dependencies
Key-Value Store Indexer	• HBase • Solr	Ranger
Kudu		Ranger
Livy	• Spark-on-YARN • YARN	Hive
Oozie	YARN	• Hive • Spark on YARN • ZooKeeper
Ozone		• HDFS • Ranger
Ranger	• HDFS • Solr	
S3 Connector		
Schema Registry		• HDFS • Ranger
Solr	• HDFS • ZooKeeper	Ranger
Spark on YARN	YARN	• Atlas • HBase
Streams Messaging Manager	Kafka	• Ranger • Schema Registry • Zookeeper
Streams Replication Manager		Kafka
Tez	YARN	
YARN	• HDFS • ZooKeeper	Ranger
Zeppelin	• HDFS • Spark-on-YARN • YARN	• Livy
ZooKeeper		

Related Information

Runtime Cluster Hosts and Role Assignments

Cloudera Manager sudo command options

To install, configure, start and stop the Cloudera Manager (CM), manage files, and so on, you can use the CM sudo commands.

Following is the list of sudo commands run by Cloudera Manager.

Note: In the list, RH6 = RHEL 6 / CentOS 6 / Oracle 6, RH7+ = RHEL 7 / CentOS 7 Oracel 7, and later, and SLES 11 and later,Ubuntu = All Ubuntu versions, and SLES = All SLES versions. For those command supported in all the Operating System (OS) versions, an OS flavor is not specified.

- sudo yum (RH6, RH7+) - Install or remove software.
- sudo apt-get (Ubuntu) - Install or remove software.

- sudo apt-key (Ubuntu) - Update Repository key.
- sudo sed - Edit one or more text files (stream editor).
- sudo systemctl (RH7+, Ubuntu) - Start, stop, or configure software.
- sudo service (RH6) - Start or stop software.
- sudo /sbin/chkconfig sudo chkconfig (RH6) - Configure software.
- sudo /usr/sbin/update-rc.d (Ubuntu) - Configure software.
- sudo id - Used for user identification.
- sudo rm - Remove files.
- sudo mv - Move or rename files.
- sudo chown - Modify file ownership.
- sudo install - Install software.
- sudo service (RH6) - Start, stop, or restart the Cloudera Manager Server and Cloudera Manager Agents on the cluster hosts.
- sudo systemctl (RH7+, Ubuntu) - Start, stop, or restart the Cloudera Manager Server and Cloudera Manager Agents on the cluster hosts.
- sudo cp - Used for file copy.
- sudo /opt/cloudera/cm-agent/bin/cm - Used for certificate management and troubleshooting.
- sudo mkdir - Used for directory creation.
- sudo /opt/cloudera/parcels/keycloak/cloudera_keycloak.sh - Configure and startup Keycloak.
- sudo keytrustee - Used for Keystrustee backup.
- sudo ln - Manage file links.
- sudo chmod - Manage file permissions.
- sudo wget - Used to host local repositories for CM and CDH.
- sudo -u postgres psql postgres - Connect to PSQL as postgres user.
- sudo -E tar - Archive CM agent data directories prior to updates or changes.
- sudo zypper clean --all (SLES) - Clean up the repository cache for SLES package manager (zypper).
- sudo ktadmin enable-synchronous-replication - Enable synchronous replication on the active Key Trustee Server.
- sudo ktadmin enable-synchronous-replication - Enable synchronous replication on the active Key Trustee Server.
- sudo rpm (RH6, RH7+) - Install or remove the CM RPM packages.

Introduction to Parcels

Parcels are a packaging format that facilitate upgrading software from within Cloudera Manager.

You can download, distribute, and activate a new software version all from within Cloudera Manager. Cloudera Manager downloads a parcel to a local directory. Once the parcel is downloaded to the Cloudera Manager Server host, an Internet connection is no longer needed to deploy the parcel. For detailed information about parcels, see Overview of Parcels.

If your Cloudera Manager Server does not have Internet access, you can obtain the required parcel files and put them into a parcel repository. For more information, see Configuring a Local Parcel Repository on page 138.

After You Install

The following topics describe post-installation actions, such as deploying client configuration and some simple tests to validate the installation and confirm that everything is working as expected.

Related Information

Cloudera Private Cloud Base Installation Guide

Deploying Clients

Client configuration files are generated automatically by Cloudera Manager based on the services you install.

Cloudera Manager deploys these configurations automatically at the end of the installation workflow. You can also download the client configuration files to deploy them manually.

If you modify the configuration of your cluster, you might need to redeploy the client configuration files. If a service's status is "Client configuration redeployment required," you need to redeploy those files.

Initializing Solr and creating HDFS home directory

After installing the Solr service and setting up Kerberos authentication, you need to initialize the service and create the HDFS home directory. Without this, the Solr service fails to start.

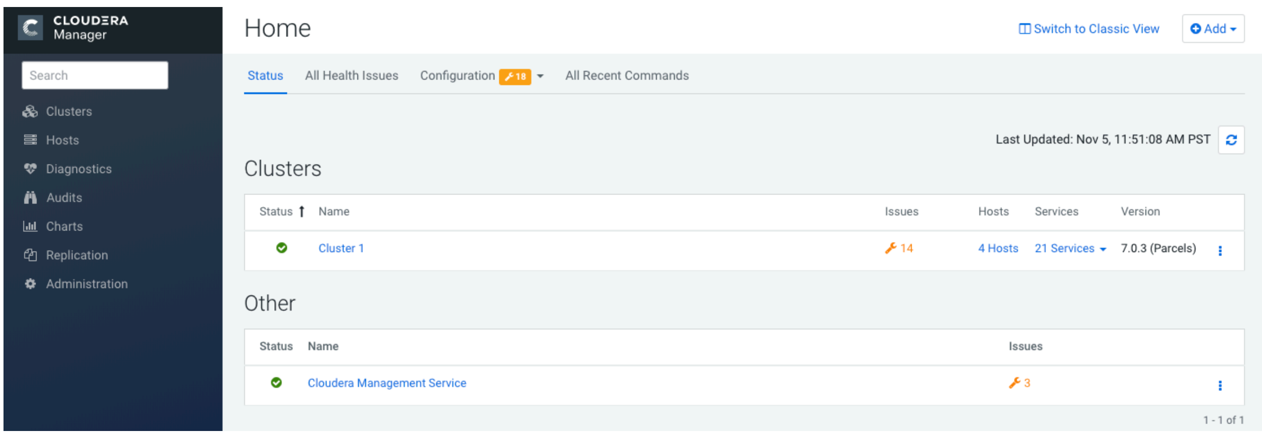
Procedure

1. In Cloudera Manager select the cluster where you want to initialize the Solr service.
2. Select the Solr service you want to initialize.
3. To initialize the Solr service select Actions Initialize Solr .
4. To create the HDFS Home directory, select Actions Create HDFS Home Dir .
5. Repeat these steps on all Solr services (Infra or workload) you need to initialize in the cluster.

Testing the Installation

Begin testing the installation from the **Home** page, where you can start by checking the health of the services.

To begin testing, start the Cloudera Manager Admin Console. Once you've logged in, the **Home** page should look something like this:



On the left side of the screen is a list of services currently running with their status information. All the services should be running with Good Health . You can click each service to view more detailed information about each service. You can also test your installation by either checking each Host's heartbeats, running a MapReduce job, or interacting with the cluster with an existing Hue application.

Checking Host Heartbeats

One way to check whether all the Agents are running is to look at the time since their last heartbeat. You can do this by clicking the Hosts tab where you can see a list of all the hosts along with the value of their Last Heartbeat.

By default, every Agent must heartbeat successfully every 15 seconds. A recent value for the Last Heartbeat means that the Server and Agents are communicating successfully.

Running a MapReduce Job

Run a PiEstimator job to manually verify that the Cloudera Private Cloud Base installation was successful.

About this task

Note: If you have a secure cluster, use the kinit command line tool to authenticate to Kerberos.

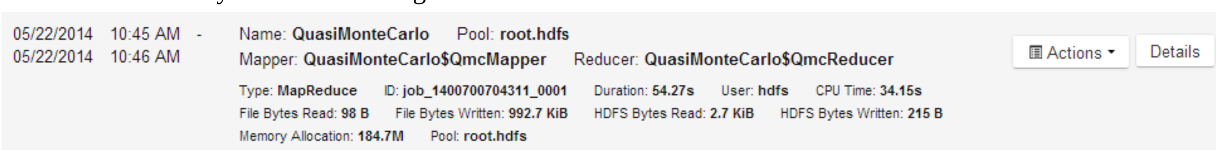
Procedure

1. Log into a host in the cluster.
2. Run the Hadoop PiEstimator example using the following command:

```
yarn jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-
examples.jar pi 10 100
```

3. In Cloudera Manager, navigate to Cluster *CLUSTERNAME* yarn Applications .
4. Check the results of the job.

   You will see an entry like the following:



Testing with Hue

You can test the cluster by running Hue.

About this task

Hue is a graphical user interface that allows you to interact with your clusters by running applications that let you browse HDFS and cloud object storage such as S3 and ABFS, manage a Hive metastore, and run Hive, Impala, and Search queries, and Oozie workflows.

Procedure

1. From Cloudera Manager, go to Clusters Hue service .
2. Click Web UI link and select the Hue web URL, which opens Hue in a new window.

   By default, Authentication Backend is set to AllowFirstUserDjangoBackend. This makes the first user who logs into Hue the Superuser and allows you to set the username and password, and create other users.

   You can change the Authentication Backend as per your requirements from Hue configurations in Cloudera Manager.
3. You can run a query or browse the database that you have set up for Hue.

   For more information, see the Hue documentation.

Deploying Atlas service

Post installation, you must plan to employ either the LDAP or Active Directory (AD) authentication mechanism to deploy the Atlas service in your production environment.

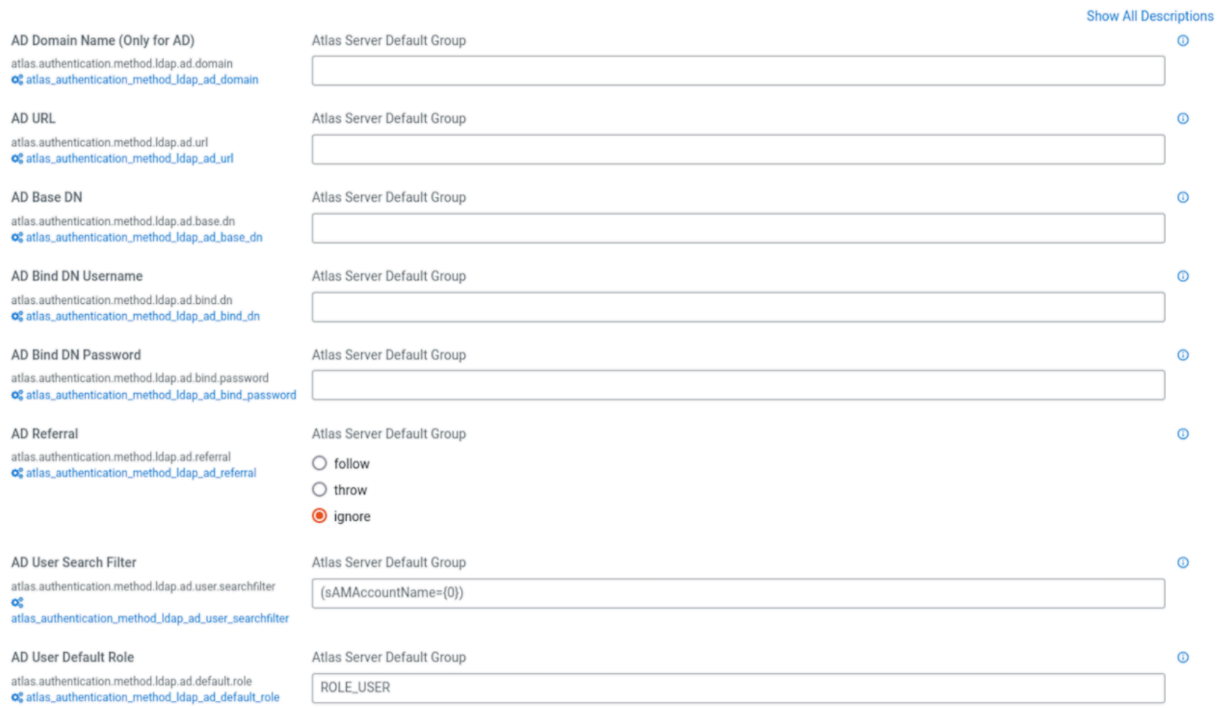
About this task

You can add LDAP or AD authentication configurations post-installation of Atlas.

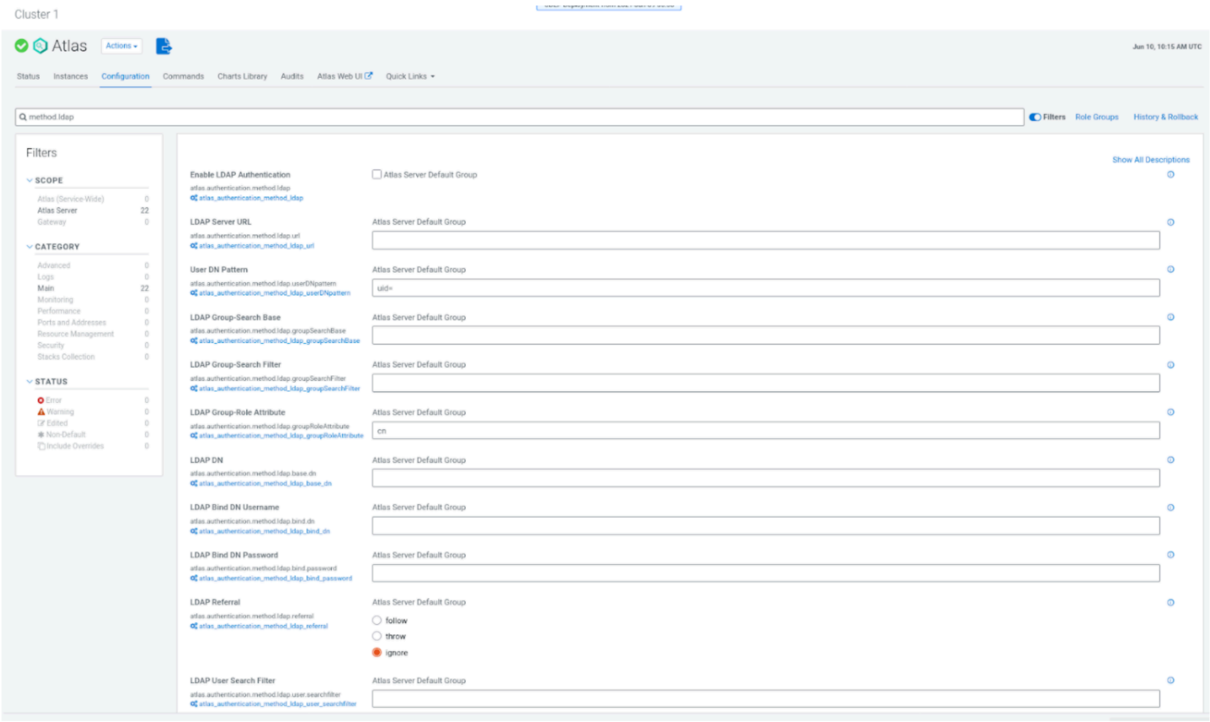
Important: When you install Atlas using the Add Service method in the Cloudera Manager instance, you must make sure to “uncheck” Enable File Authentication option.

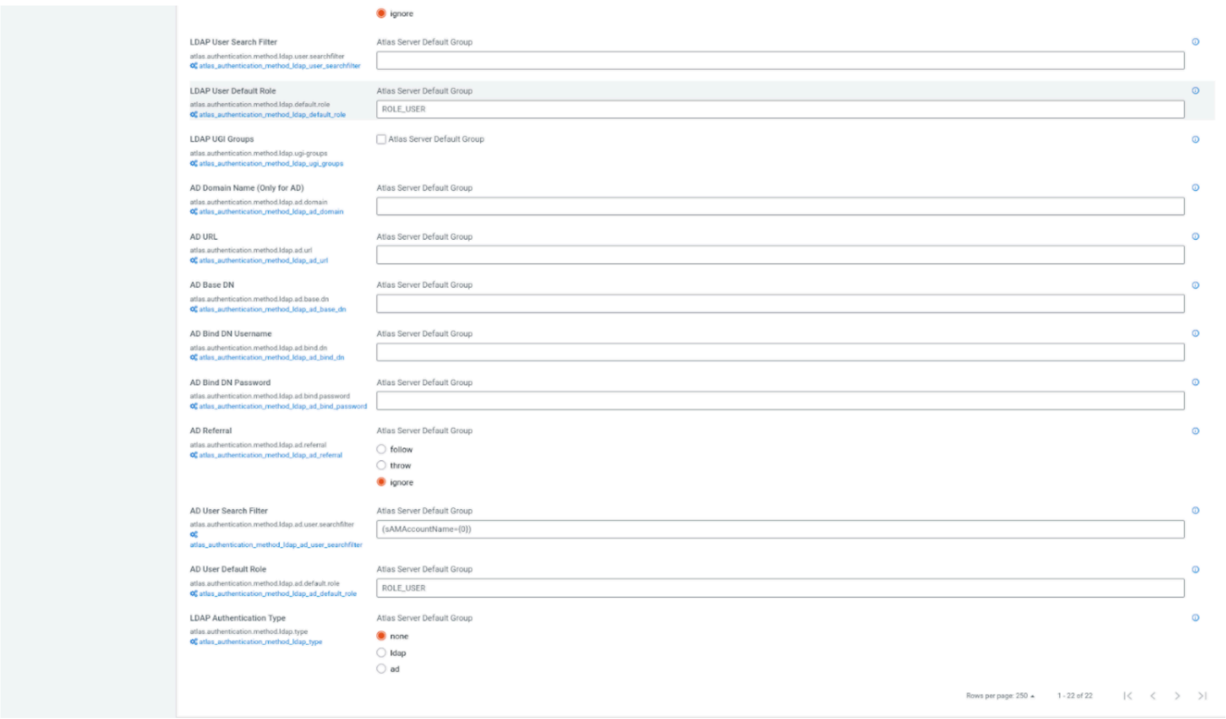
Procedure

1. In your Cloudera Manager instance > Select Clusters > Configuration tab > On the search bar, use the key: atla s.authentication.method.
   The list of LDAP and AD configurations are displayed.
2. In the search, you are allowed to select the type of Atlas installation for LDAP Authentication type:
   - none
   - ldap
   - ad
3. Selecting “ad”, prompts you to use appropriate active directory values to complete the Atlas authentication type.

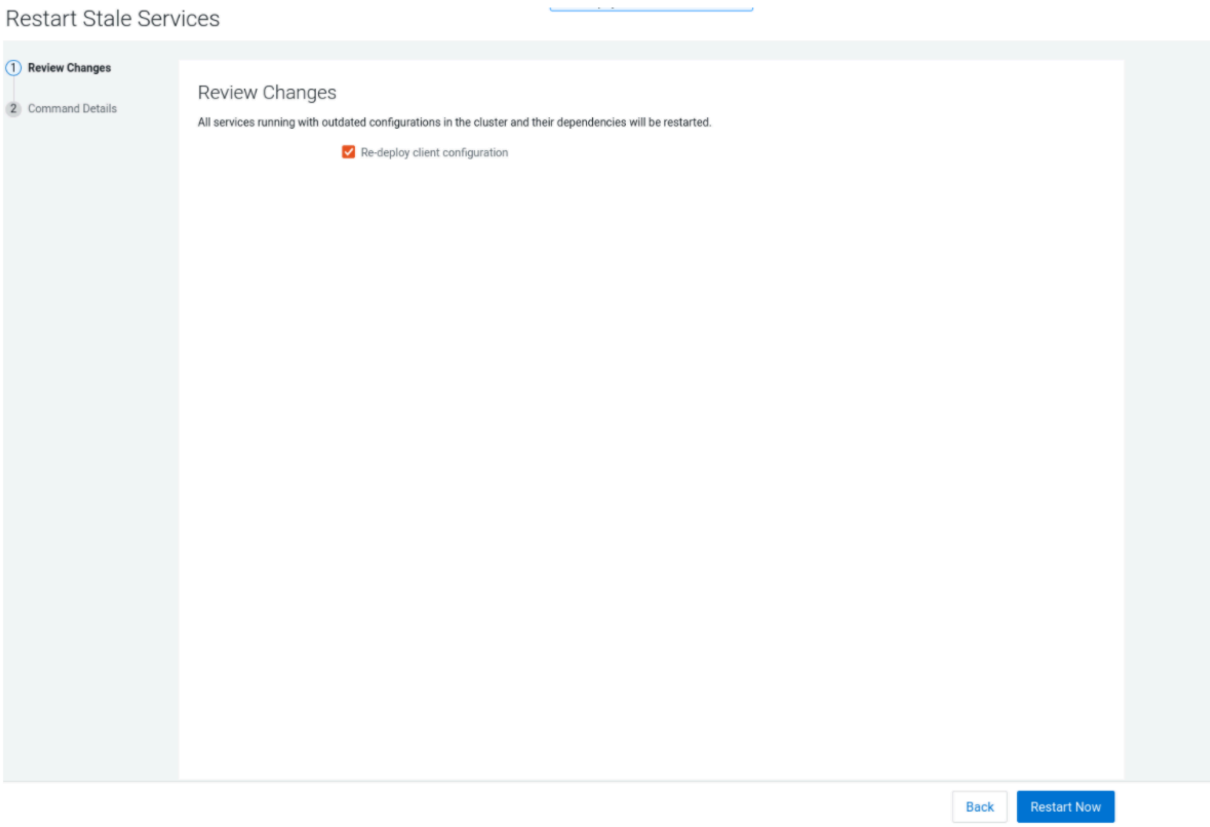


4. Selecting “ldap”, prompts you to use appropriate LDAP values to complete the Atlas authentication type.





5. You must restart the stale services.



Secure Your Cluster

After completing your Cloudera Enterprise installation and making sure that everything is working properly, secure your cluster by enabling authentication, authorization, auditing, and encryption.

For comprehensive instructions on securing your cluster, see the Security documentation.

Related Information

Security Overview

Installing the GPL Extras Parcel

GPL Extras contains functionality for compressing data using the LZO compression algorithm. To install the GPL Extras parcel:

About this task

Important: LZO compression is deprecated in 7.x and will be removed in a future release.

Procedure

1. Add the appropriate repository to the Cloudera Manager list of parcel repositories. Specify the repository in Cloudera Manager as follows:

```
https://USERNAME:PASSWORD@archive.cloudera.com/p/gplextras7/7.1.9.0/parce
ls/
```

   You can also download the parcel into a local parcel repository.
2. Download, distribute, and activate the parcel.
3. The LZO parcels require that the underlying operating system has the native LZO packages installed. If they are not installed on all cluster hosts, you can install them as follows:

   RHEL compatible:

```
sudo yum install lzo
```

   Debian or Ubuntu:

```
sudo apt-get install liblzo2-2
```

   SLES:

```
sudo zypper install liblzo2-2
```

4. To configure LZO compression, see Configuring Services to Use LZO Compression.

Configuring HDFS properties to optimize log collection

CDP uses “out_webhdfs” Fluentd output plugin to write records into HDFS, in the form of log files, which are then used by different Data Services to generate diagnostic bundles. Over time, these log files can grow in size. To optimize the size of logs that are captured and stored on HDFS, you must update certain HDFS configurations in the hdfs-site.xml file using Cloudera Manager.

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters HDFS service Configuration .
3. Select the Enable WebHDFS (dfs_webhdfs_enabled) option.
4. Add the following lines in the HDFS Service Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml field by clicking View as XML to enable append operations:

```
<property>
  <name>dfs.support.append</name>
  <value>true</value>
</property>

<property>
  <name>dfs.support.broken.append</name>
  <value>true</value>
</property>
```

5. Click Save Changes.
6. Restart the HDFS service.
7. Restart your CDP cluster.

Related Information

Fluentd documentation

Migrating a YARN Queue Manager database to an external database

If you prefer to continue using the embedded database and do not wish to use an external database, install Cloudera Data Platform (CDP) 7.1.9 CHF 2 and disregard these instructions. See Known issues in 7.1.9 CHF 2 linked below for more information. If you want to use an external database, use PostgreSQL database, install CDP 7.1.9 or CDP 7.1.9 CHF 1, and continue with these post-installation steps to migrate your database. This migration is needed because Queue Manager uses a different database as its backing store in CDP 7.1.9 and CDP 7.1.9 CHF 2 than what was used in previous versions of CDP 7.1.8 and below. Once you have configured the database as detailed in the steps below, the data migration process is automatic.

About this task

You must supply the details of a PostgreSQL database to migrate your old data to CDP 7.1.9 or CPD 7.1.9 CHF 1 as the only external database that Queue Manager supports in these versions is PostgreSQL. When updating to versions after 7.1.9 CHF 2, YARN QM will continue to use PostgreSQL if it was initially configured.

Important: For users who only use the Queue Manager API directly, you need to use the UI to setup the Database migration as this migration is triggered through the UI. You need to use the Cloudera Manager and Queue Manager's UIs to setup the database configuration, and to load the UI to allow the data migration to complete properly. You must also refrain from direct API calls during an upgrade to 7.1.9 and refrain from direct API calls until the data migration is completed. Ensure that your data migration has succeeded before you make any API calls

Before you begin

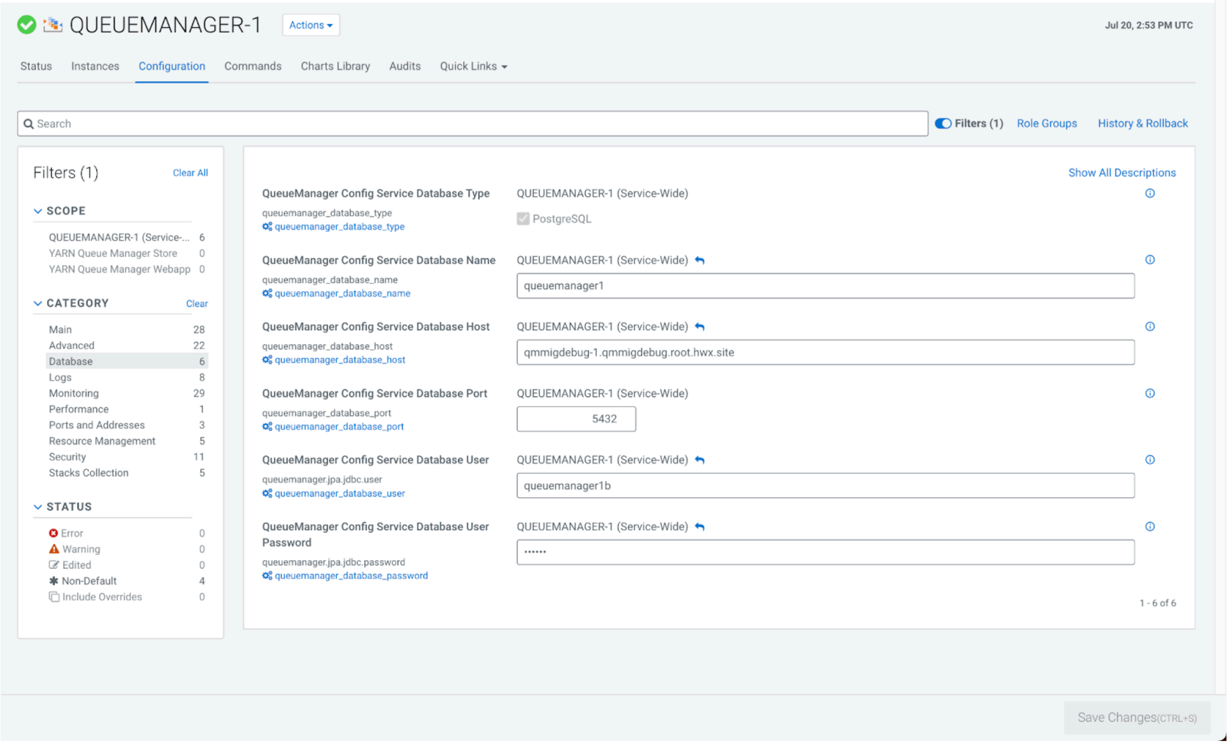
You must first create a database for Queue Manager and then configure the database to assign roles and permissions for the user (dbuser). The roles users need to assign are the following: SUPERUSER CREATEDB CREATEROLE INHERIT LOGIN. You can use the sample query or a similar query below:

```
CREATE ROLE qmadmin PASSWORD 'password' SUPERUSER CREATEDB CREATEROLE INHERI
T LOGIN;
CREATE DATABASE "configstore";
ALTER DATABASE "configstore" OWNER TO qmadmin;
```

Procedure

1. In your Cloudera Manager instance, navigate to Clusters Yarn Queue Manager .
2. Click the Configuration tab.
3. In the left navigation menu under Category, click Database.

4. Enter the following fields that are required for the PostgreSQL database that is to be used. You will have these values from the query that was run in the prerequisite:
    - QueueManager Config Service Database Name
    - QueueManager Config Service Database Host
    - QueueManager Config Service Database Port
    - QueueManager Config Service Database User
    - QueueManager Config Service Database User Password



5. Click Save Changes.
6. In Queue Manager, navigate to Actions and click Restart. The data migration to the PostgreSQLdata database may take a few minutes after Queue Manager is restarted.

Related Information

Known Issues in 7.1.9 CHF 2

Troubleshooting Installation Problems

This topic describes common installation issues and suggested solutions.

TLS Protocol Error with OpenJDK

If you are using an older version of OpenJDK 1.8 and have enabled SSL/TLS for the Cloudera Manager Admin Console, you may encounter a TLS protocol error when connecting to the Admin Console, stating that there are no ciphers in common. This is because older versions of OpenJDK may not implement certain TLS ciphers, causing an inability to log into the Cloudera Manager Admin Console when TLS is enabled.

Workaround:

You can workaround this issue by doing one of the following:

- Upgrade OpenJDK to a supported version of OpenJDK that is higher than version 1.8.0_181.

- If it is not possible to upgrade OpenJDK, enable less secure TLS ciphers in Cloudera Manager. You can do this by opening the /etc/default/cloudera-scm-server in a text editor and adding the following line:

```
export CMF_OVERRIDE_TLS_CIPHERS=<CIPHER_LIST>
```

Where <*CIPHER_LIST*> is a list of TLS cipher suites separated by colons. For example:

```
export CMF_OVERRIDE_TLS_CIPHERS="TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
:TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256:TLS_ECDHE_ECDSA_WITH_AES_256_GCM_
SHA384:TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384:TLS_DHE_RSA_WITH_AES_128_GC
M_SHA256:TLS_DHE_RSA_WITH_AES_256_GCM_SHA384:TLS_ECDHE_ECDSA_WITH_AES_12
8_CBC_SHA256:TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256:TLS_ECDHE_ECDSA_WITH_
AES_128_CBC_SHA:TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384:TLS_ECDHE_RSA_WITH
_AES_128_CBC_SHA:TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384:TLS_ECDHE_ECDSA
_WITH_AES_256_CBC_SHA:TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA:TLS_DHE_RSA_WIT
H_AES_128_CBC_SHA256:TLS_DHE_RSA_WITH_AES_128_CBC_SHA:TLS_DHE_RSA_WITH_A
ES_256_CBC_SHA256:TLS_DHE_RSA_WITH_AES_256_CBC_SHA:TLS_ECDHE_ECDSA_WITH_
3DES_EDE_CBC_SHA:TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA:TLS_EDH_RSA_WITH_3D
ES_EDE_CBC_SHA:TLS_RSA_WITH_AES_128_GCM_SHA256:TLS_RSA_WITH_AES_256_GCM_
SHA384:TLS_RSA_WITH_AES_128_CBC_SHA256:TLS_RSA_WITH_AES_256_CBC_SHA256:T
LS_RSA_WITH_AES_128_CBC_SHA:TLS_RSA_WITH_AES_256_CBC_SHA:TLS_RSA_WITH_3D
ES_EDE_CBC_SHA"
```

Cloudera Bug: OPSAPS-49578

Failed to start server reported by cloudera-manager-installer.bin

"Failed to start server" reported by cloudera-manager-installer.bin. /var/log/cloudera-scm-server/cloudera-scm-server .logcontains a message beginning Caused by: java.lang.ClassNotFoundException: com.mysql.jdbc.Driver...

Possible reason:

You might have SELinux enabled.

Possible solution:

Disable SELinux by running sudo setenforce 0 on the Cloudera Manager Server host. To disable it permanently, edit / etc/selinux/config.

Installation interrupted and installer does not restart

Possible reason:

You need to do some manual cleanup.

Possible solution:

See *Uninstalling Cloudera Manager and Managed Software*.

Cloudera Manager Server fails to start with MySQL

Cloudera Manager Server fails to start and the Server is configured to use a MySQL database to store information about service configuration.

Possible reason:

Tables might be configured with the ISAM engine. The Server does not start if its tables are configured with the MyISAM engine, and an error such as the following appears in the log file:

```
Tables ... have unsupported engine type ... .  InnoDB is required.
```

Possible solution:

Make sure that the InnoDB engine is configured, not the MyISAM engine. To check what engine your tables are using, run the following command from the MySQL shell: mysql> show table status;

For more information, see Install and Configure MySQL for Cloudera Software on page 160.

Agents fail to connect to Server

Agents fail to connect to Server. You get an Error 113 ('No route to host') in /var/log/cloudera-scm-agent/cloudera-scm-agent.log.

Possible reason:

You might have SELinux or iptables enabled.

Possible solution:

Check /var/log/cloudera-scm-server/cloudera-scm-server.log on the Server host and /var/log/cloudera-scm-agent/cloudera-scm-agent.log on the Agent hosts. Disable SELinux and iptables.

Cluster hosts do not appear

Some cluster hosts do not appear when you click Find Hosts in install or update wizard.

Possible reason:

You might have network connectivity problems.

Possible solution:

- Make sure all cluster hosts have SSH port 22 open.
- Check other common causes of loss of connectivity such as firewalls and interference from SELinux.

"Access denied" in install or update wizard

"Access denied" in install or update wizard during database configuration for Reports Manager.

Possible reason:

Hostname mapping or permissions are not set up correctly.

Possible solution:

- For hostname configuration, see *Configure Network Names*.
- For permissions, make sure the values you enter into the wizard match those you used when you configured the databases. The value you enter into the wizard as the database hostname must match the value you entered for the hostname (if any) when you configured the database.

  For example, if you had entered the following when you created the database

  ```
  grant all on activity_monitor.* TO 'amon_user'@'myhost1.myco.com' IDENTIFIED BY 'amon_password';
  ```

  the value you enter here for the database hostname must be myhost1.myco.com. If you did not specify a host, or used a wildcard to allow access from any host, you can enter either the fully qualified domain name (FQDN), or localhost. For example, if you entered

  ```
  grant all on activity_monitor.* TO 'amon_user'@'%' IDENTIFIED BY 'amon_password';
  ```

  the value you enter for the database hostname can be either the FQDN or localhost.

Databases fail to start.

Reports Manager or Service Monitor databases fail to start.

Possible reason:

MySQL binlog format problem.

Possible solution:

Set binlog_format=mixed in /etc/my.cnf. For more information, see this MySQL bug report. See also Step 3. Install and Configure Databases on page 150.

Cloudera services fail to start

Possible reason:

Java might not be installed or might be installed at a custom location.

Possible solution:

See *Configuring a Custom Java Home Location* for more information on resolving this issue.

Create Hive Metastore Database Tables command fails

The Create Hive Metastore Database Tables command fails due to a problem with an escape string.

Possible reason:

PostgreSQL versions 9 and higher require special configuration for Hive because of a backward-incompatible change in the default value of the standard_conforming_strings property. Versions up to PostgreSQL 9.0 defaulted to off, but starting with version 9.0 the default is on.

Possible solution:

As the administrator user, use the following command to turn standard_conforming_strings off:

```
ALTER DATABASE <hive_db_name> SET standard_conforming_strings = off;
```

Oracle invalid identifier

If you are using an Oracle database and the Cloudera Navigator AnalyticsAuditActivity tab displays "No data available" and there is an Oracle error about "invalid identifier" with the query containing the reference to dbms_crypto in the log.

Possible reason:

You have not granted execute permission to sys.dbms_crypto.

Possible solution:

Run GRANT EXECUTE ON sys.dbms_crypto TO *NAV*;, where *NAV* is the user of the Navigator Audit Server database.

Related Information

Cloudera Private Cloud Base Installation Guide

Uninstalling Cloudera Manager and Managed Software

Complete the following tasks to uninstall the Cloudera Manager Server, Agents, managed software, and databases.

Related Information

Cloudera Private Cloud Base Installation Guide

Record User Data Paths

Record the location of the user data paths by checking the configuration in each service.

The user data paths listed in the topic *Remove User Data*, /var/lib/flume-ng /var/lib/hadoop* /var/lib/hue /var/lib/navigator /var/lib/oozie /var/lib/solr /var/lib/sqoop* /var/lib/zookeeper *DATA_DRIVE_PATH*/dfs *DATA_DRIVE_PATH*/mapred *DATA_DRIVE_PATH*/yarn, are the default settings. However, at some point they might have been reconfigured in Cloudera Manager. If you want to remove all user data from the cluster and have changed the paths, either when you installed Runtime and managed services or at some later time, note the location of the paths by checking the configuration in each service.

Stop all Services

Stop all services for each cluster managed by Cloudera Manager.

Procedure

1. On the HomeStatus tab, click three dots to the right of the cluster name and select Stop.
2. Click Stop in the confirmation screen. The Command Details window shows the progress of stopping services. When All services successfully stopped appears, the task is complete and you can close the Command Details window.
3. On the HomeStatus tab, click the three dots to the right of the Cloudera Management Service entry and select Stop. The Command Details window shows the progress of stopping services.

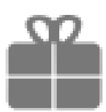
Results

When All services successfully stopped appears, the task is complete and you can close the Command Details window.

Deactivate and Remove Parcels

If you installed using packages, skip this step and go to *Uninstall the Cloudera Manager Server*; you will remove packages in *Uninstall Cloudera Manager Agent and Managed Software*. If you installed using parcels remove them as follows:

Procedure

1. Click the parcel indicator  in the left-hand navigation bar.
2. In the Location selector on the left, select All Clusters.
3. For each activated parcel, select ActionsDeactivate. When this action has completed, the parcel button changes to Activate.
4. For each activated parcel, select ActionsRemove from Hosts. When this action has completed, the parcel button changes to Distribute.
5. For each activated parcel, select ActionsDelete. This removes the parcel from the local parcel repository.

What to do next

There might be multiple parcels that have been downloaded and distributed, but that are not active. If this is the case, you should also remove those parcels from any hosts onto which they have been distributed, and delete the parcels from the local repository.

Delete the Cluster

On the Home page, Click the drop-down list next to the cluster you want to delete and select Delete.

Uninstall the Cloudera Manager Server

The commands for uninstalling the Cloudera Manager Server depend on the method you used to install it. Refer to steps below that correspond to the method you used to install the Cloudera Manager Server.

Procedure

1. If you used the cloudera-manager-installer.bin file (the trial installer): Run the following command on the Cloudera Manager Server host:

```
sudo /opt/cloudera/installer/uninstall-cloudera-manager.sh
```

2. If you did not use the cloudera-manager-installer.bin file: If you installed the Cloudera Manager Server using a different installation method such as Puppet, run the following commands on the Cloudera Manager Server host:

a) Stop the Cloudera Manager Server and its database:

```
sudo service cloudera-scm-server stop
sudo service cloudera-scm-server-db stop
```

b) Uninstall the Cloudera Manager Server and its database. This process described also removes the embedded PostgreSQL database software, if you installed that option. If you did not use the embedded PostgreSQL database, omit the cloudera-manager-server-db steps.

RHEL

```
sudo yum remove cloudera-manager-server
sudo yum remove cloudera-manager-server-db-2
```

>SLES

```
sudo zypper -n rm --force-resolution cloudera-manager-server
sudo zypper -n rm --force-resolution cloudera-manager-server-db-2
```

Ubuntu

```
sudo apt-get remove cloudera-manager-server
sudo apt-get remove cloudera-manager-server-db-2
```

Uninstall Cloudera Manager Agent and Managed Software

To uninstall Cloudera Manager Agent and managed software, stop the Cloudera Manager Agent on all hosts, remove the parcel installation, and run the clean command.

About this task

Do the following on all Agent hosts:

Procedure

1. Stop the Cloudera Manager Agent.

```
sudo systemctl stop supervisord
```

2. To uninstall managed software, run the following commands:

   RHEL: $ sudo yum remove 'cloudera-manager-*'

   RHEL

   ```
   sudo yum remove 'cloudera-manager-*'
   ```

   SLES

   ```
   sudo zypper remove 'cloudera-manager-*'
   ```

   Ubuntu

   ```
   sudo apt-get purge 'cloudera-manager-*'
   ```

3. Run the clean command:

   RHEL

   ```
   sudo yum clean all
   ```

   SLES

   ```
   sudo zypper clean
   ```

   Ubuntu

   ```
   sudo apt-get clean
   ```

Remove Cloudera Manager, User Data, and Databases

Permanently remove Cloudera Manager data, the Cloudera Manager lock file, and user data. Then stop and remove the databases.

Procedure

1. On all Agent hosts, stop any running Cloudera Manager and managed processes:

   ```
   for u in cloudera-scm flume hadoop hdfs hbase hive httpfs hue impala llama
    mapred oozie solr spark sqoop sqoop2 yarn zookeeper; do sudo kill $(ps -u
    $u -o pid=); done
   ```

   Note: This step should not be necessary if you stopped all the services and the Cloudera Manager Agent correctly.

2. If you are uninstalling on RHEL, run the following commands on all Agent hosts to permanently remove Cloudera Manager data. If you want to be able to access any of this data in the future, you must back it up before removing it. If you used an embedded PostgreSQL database, that data is stored in /var/lib/cloudera-scm-server-db.

   ```
   sudo umount cm_processes
   sudo rm -Rf /usr/share/cmf /var/lib/cloudera* /var/cache/yum/cloudera* /
   var/log/cloudera* /var/run/cloudera*
   ```

3. On all Agent hosts, run this command to remove the Cloudera Manager lock file:

   ```
   sudo rm /tmp/.scm_prepare_node.lock
   ```

4. This step permanently removes all user data. To preserve the data, copy it to another cluster using the distcp command before starting the uninstall process.
   a) On all Agent hosts, run the following commands:

   ```
   sudo rm -Rf /var/lib/flume-ng /var/lib/hadoop* /var/lib/hue /var/
   lib/navigator /var/lib/oozie /var/lib/solr /var/lib/sqoop* /var/lib/
   zookeeper
   ```

   b) Run the following command on each data drive on all Agent hosts (adjust the paths for the data drives on each host):

   ```
   sudo rm -Rf DATA_DRIVE_PATH/dfs DATA_DRIVE_PATH/mapred DATA_DRIVE_PATH/
   yarn
   ```

5. Stop and remove the databases. If you chose to store Cloudera Manager or user data in an external database, see the database vendor documentation for details on how to remove the databases.

Uninstalling a Runtime Component From a Single Host

The following procedure removes Runtime software components from a single host that is managed by Cloudera Manager.

Procedure

1. In the Cloudera Manager Administration Console, select HostsAll Hosts.
   A list of hosts in the cluster displays.
2. Select the host where you want to uninstall Runtime software.
3. Click the Actions for Selected button and select Remove From Cluster.
   Cloudera Manager removes the roles and host from the cluster.
4. Optionally, manually delete the krb5.conf file used by Cloudera Manager.