CDP Private Cloud Base 7

In-Place Upgrade CDH 6 to CDP Private Cloud Base

Date published: 2019-11-22
Date modified: 2024-12-10

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2026. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Upgrading CDH 6 to Cloudera Private Cloud Base 11

Assessing the Impact of an Upgrade 12

How much time should I plan for to complete my upgrade? 13

About using this online Upgrade Guide 13

Cloudera Base on premises Pre-upgrade transition steps 14

Set log level for KeyTrustee KMS to INFO 14
Transitioning from Sentry Policy Files to the Sentry Service 15
Transitioning the Sentry service to Apache Ranger 16
Configuring a Ranger or Ranger KMS Database: MySQL/MariaDB 16
Configuring a Ranger Database: PostgreSQL 18
Configuring a Ranger or Ranger KMS Database: Oracle 19
Import Key Trustee KMS ACLs to Ranger KMS policies 19
Transitioning Navigator content to Atlas 20
High-level transition process 21
Mapping Navigator metadata to Atlas 32
Transitioning Navigator audits 33
What's new in Atlas for Navigator Users? 34
Preparing the backend HMS database for upgrade 34
Migrating Hive 1-2 to Hive 3 35
Hive Configuration Changes Requiring Consent 35
Remove transactional=false from Table Properties 40
Check SERDE Definitions and Availability 40
Checking Apache HBase 40
Check co-processor classes 41
Clean the HBase Master procedure store 42
CDH cluster upgrade requirements for Replication Manager 43

Upgrading the JDK 44

Manually Installing Oracle JDK 1.8 44
OpenJDK 46
Manually Installing OpenJDK 46
Manually Migrating to OpenJDK 48
Using AES-256 Encryption 52
Configuring a Custom Java Home Location 53
Tuning JVM Garbage Collection 54

Upgrading the Operating System 57

Step 1: Getting Started with Operating System Upgrades 57
Prerequisites 57

Step 2: Backing Up Host Files Before Upgrading the Operating System....57
Backing Up....57
Backing up Cloudera Manager databases....58
Step 3: Before You Upgrade the Operating System....59
Decommission and Stop Running Roles....59
Stop Cloudera Manager Agent....60
Stop Cloudera Manager Server & Agent....60
Stop Databases....61
Remove Packages & Parcels....61
Upgrade the Operating System....62
Step 4: After You Upgrade the Operating System....62
Establish Access to the Software....62
Reinstall Daemon & Agent Packages....64
Reinstall Server, Daemon & Agent Packages....64
Start Databases....65
Start Server & Agent....65
Start Roles....66

Upgrading Cloudera Manager 6....66
Step 1: Getting Started Upgrading Cloudera Manager 6....67
Collect Information....68
Preparing to Upgrade Cloudera Manager....69
Step 2: Backing Up Cloudera Manager 6....71
Collect Information for Backing Up Cloudera Manager....72
Back Up Cloudera Manager Agent....73
Back Up the Cloudera Management Service....73
Back Up Cloudera Navigator Data....74
Stop Cloudera Manager Server & Cloudera Management Service....74
Back Up the Cloudera Manager Databases....75
Back Up Cloudera Manager Server....76
(Optional) Start Cloudera Manager Server & Cloudera Management Service....77
Step 3: Upgrading the Cloudera Manager Server....78
Establish Access to the Software....79
Install Java (JDK)....81
Upgrade the Cloudera Manager Server....83
Step 4: Upgrading the Cloudera Manager Agents....87
Upgrade the Cloudera Manager Agents (Cloudera Manager 7.0.3 and higher)....88
Step 5: After You Upgrade Cloudera Manager....92
Perform Post-Upgrade Steps....92
Upgrade Key Trustee Server to 7.1.x....96
Upgrade Navigator Encrypt to 7.1.x....97
Upgrading Cloudera Navigator Key HSM....101
Upgrading Key Trustee KMS....103
Troubleshooting a Cloudera Manager Upgrade....104
The Cloudera Manager Server fails to start after upgrade....104
Re-Running the Cloudera Manager Upgrade Wizard....104
Reverting a Failed Cloudera Manager Upgrade....104
Ensure Cloudera Manager Server and Agent are stopped....105
Restore the Cloudera Manager Database (if necessary)....105
Establish Access to the Software....106
Downgrade the Cloudera Manager Packages....107
Restore the Cloudera Manager Directory....108
Start Cloudera Manager Again....109

Validate TLS configurations.....109

Expediting the Hive upgrade.....111
Overview of the expedited Hive upgrade.....111
Preparing tables for migration.....112
Configuring HSMM to prevent migration.....113
Understanding the Hive upgrade.....114

Installing dependencies for Hue.....115
Installing the psycopg2 Python package for PostgreSQL database.....115
Installing MySQL client for MySQL databases.....117
Installing MySQL client for MariaDB databases.....119

Upgrading a CDH 6 Cluster.....121
Step 1: Getting Started Upgrading a Cluster.....122
Collect Information.....124
Preparing to Upgrade a Cluster.....124
Step 2: Review Notes and Warnings.....130
Step 3: Backing Up the Cluster.....134
Back Up Databases.....135
Back Up ZooKeeper.....136
Back up Solr.....136
Back Up HDFS.....137
Back Up Key Trustee Server and Clients.....138
Back Up HSM KMS.....138
Back Up Navigator Encrypt.....138
Back Up HBase.....138
Back Up YARN Queue Manager.....139
Back up Atlas.....139
Back Up Sqoop 2.....140
Back Up Hue.....141
Back Up Impala.....141
Step 4: Back Up Cloudera Manager.....141
Collect Information for Backing Up Cloudera Manager.....141
Back Up Cloudera Manager Agent.....142
Back Up the Cloudera Management Service.....142
Back Up Cloudera Navigator Data.....143
Stop Cloudera Manager Server & Cloudera Management Service.....144
Back Up the Cloudera Manager Databases.....144
Back Up Cloudera Manager Server.....145
(Optional) Start Cloudera Manager Server & Cloudera Management Service.....146
Step 5: Complete Pre-Upgrade steps for upgrades to Cloudera Base on premises.....147
Run Hue Document Cleanup.....149
Check Oracle Database Initialization.....152
Step 6: Access Parcels.....152
Step 7: Configure Streams Messaging Manager.....153
.....153
Step 8: Configure Schema Registry.....153
.....153
.....154
Step 9: Enter Maintenance Mode.....154

Step 10: Run the Upgrade Cluster Wizard....154
Fair Scheduler to Capacity Scheduler transition....162
Configure TLS/SSL for Ranger in a manually configured TLS/SSL environment....181
Step 11: Finalize the HDFS or Ozone Upgrade....184
Step 12: Complete Post-Upgrade steps for upgrades to Cloudera Base on premises....184
Step 13: Exit Maintenance Mode....186
Troubleshooting Upgrades....187
Cloudera Runtime upgrade failure....187
"Access denied" in install or update wizard....187
The Hive on Tez service and the RangerAdmin role of the Ranger service kept going down....187
Cluster hosts do not appear....188
Cannot start services after upgrade....188
HDFS DataNodes fail to start....188
Cloudera services fail to start....188
Host Inspector Fails....189

Manual upgrade to Cloudera Base on premises.... 189
Upgrade Ranger database and apply patches....190
Setup Ranger Admin Component....190
Start Ranger....190
Set up the Ranger Plugin service....190
Start Kudu....190
Start ZooKeeper....190
Upgrade HDFS Metadata....190
Start HDFS....190
Start YARN QueueManager....190
Import Sentry Polices to Ranger....191
Start HBASE....191
Start YARN QueueManager....191
Clean NodeManager Recovery Directory (YARN)....191
Reset ACLs on YARN Zookeeper nodes....191
Install YARN MapReduce Framework Jars....191
Start YARN....191
Deploy Client Configuration Files....192
Reinitialize Solr State for Upgrade....192
Bootstrap Solr Configuration....192
Start Solr....192
Bootstrap Solr Collections....192
Create HDFS Home directory....192
Create Ranger Plugin Audit Directory....192
Start infrastructure Solr....192
Start HBASE....193
Start KAFKA....193
Create Ranger Kafka Plugin Audit Directory....193
Create HBase tables for Atlas....193
Start Atlas....193
Create Ranger Atlas Plugin Audit Directory....193
Start Phoenix....193
Install MapReduce Framework Jars....193
Start YARN....194
Deploy Client Configuration Files....194
Upgrade the Hive Metastore Database....194
Start Hive....194
Create Hive Warehouse Directory....194
Create Hive Warehouse External Directory....194

Create Hive Sys database....194
Create Ranger Plugin Audit Directory....195
Start Impala....195
Create Ranger Plugin Audit Directory....195
Create Spark Driver Log Dir....195
Start Spark....195
Start Livy....195
Upgrade Oozie Database Schema....195
Updating column types in Oozie Database....195
Upgrade Oozie SharedLib....197
Upload Tez tar file to HDFS....197
Migrate Hive tables for Cloudera Base on premises upgrade....197
Create Ranger Plugin Audit Directory....197
Start Hive on Tez....197
Start Hue....197
Start the Remaining Cluster Services....198
Validate the Hive Metastore Database Schema....198
Test the Cluster and Finalize HDFS Metadata....198
Clear the Upgrade State Table....198

Rolling Back a Cloudera Private Cloud Base Upgrade from version 7.1.9 to CDH 6....199

Review Limitations....199
Stop the Cluster....200
(Parcels) Downgrade the Software....200
Stop Cloudera Manager....201
Restore Cloudera Manager Databases....201
Restore Cloudera Manager Server....202
Start Cloudera Manager....202
Optional Step....203
Roll Back ZooKeeper....203
Roll Back HDFS....203
Start the HBase Service....208
Restore CDH Databases....209
Start the Sentry Service....210
Roll Back Cloudera Search....210
Roll Back Atlas....210
Roll Back Hue....211
Roll Back Kafka....211
Deploy the Client Configuration....212
Restart the Cluster....212
Roll Back Cloudera Navigator Encryption Components....212
    Roll Back Key Trustee Server....212
    Start the Key Management Server....213
    Roll Back Key HSM....213
    Roll Back Key Ranger KMS Parcels....214
    Roll Back HSM KMS Parcels....214
    Roll Back Navigator Encrypt....215
(Optional) Cloudera Manager Rollback Steps....215
    Stop Cloudera Manager....216
    Restore the Cloudera Manager 6 Repository Files....216
    Restore Packages....216
    Restore Cloudera Manager Databases....217
    Restore Cloudera Manager Server....218
    Start the Cloudera Manager Server and Agents....219

Rolling Back a Cloudera Private Cloud Base Upgrade from versions 7.1.1 - 7.1.7 to CDH 6 219
Review Limitations 219
Stop the Cluster 220
(Parcels) Downgrade the Software 221
Stop Cloudera Manager 221
Restore Cloudera Manager Databases 221
Restore Cloudera Manager Server 222
Start Cloudera Manager 223
Roll Back ZooKeeper 223
Roll Back HDFS 223
Start the HBase Service 227
Restore CDH Databases 229
Start the Sentry Service 229
Roll Back Cloudera Search 229
Roll Back Hue 230
Roll Back Kafka 230
Deploy the Client Configuration 231
Restart the Cluster 231
Roll Back Cloudera Navigator Encryption Components 231
Roll Back Key Trustee Server 231
Start the Key Management Server 232
Roll Back Key HSM 232
Roll Back Key Ranger KMS Parcels 233
Roll Back HSM KMS Parcels 233
Roll Back Navigator Encrypt 234
(Optional) Cloudera Manager Rollback Steps 234
Stop Cloudera Manager 234
Restore the Cloudera Manager 6 Repository Files 235
Restore Packages 235
Restore Cloudera Manager Databases 236
Restore Cloudera Manager Server 237
Start the Cloudera Manager Server and Agents 238

Rolling Back a Cloudera Private Cloud Base Upgrade from version 7.1.8 to CDH 6 238
Review Limitations 239
Stop the Cluster 239
(Parcels) Downgrade the Software 240
Stop Cloudera Manager 240
Restore Cloudera Manager Databases 241
Restore Cloudera Manager Server 241
Start Cloudera Manager 242
Roll Back ZooKeeper 242
Roll Back HDFS 242
Start the HBase Service 247
Restore CDH Databases 248
Start the Sentry Service 249
Roll Back Cloudera Search 249
Roll Back Hue 249
Roll Back Kafka 250
Deploy the Client Configuration 250

Restart the Cluster ... 250
Roll Back Cloudera Navigator Encryption Components ... 250
Roll Back Key Trustee Server ... 251
Start the Key Management Server ... 251
Roll Back Key HSM ... 252
Roll Back Key Ranger KMS Parcels ... 253
Roll Back HSM KMS Parcels ... 253
Roll Back Navigator Encrypt ... 253
(Optional) Cloudera Manager Rollback Steps ... 254
Stop Cloudera Manager ... 254
Restore the Cloudera Manager 6 Repository Files ... 255
Restore Packages ... 255
Restore Cloudera Manager Databases ... 256
Restore Cloudera Manager Server ... 256
Start the Cloudera Manager Server and Agents ... 257

Configuring a Local Package Repository ... 258
Creating a Permanent Internal Repository ... 258
Setting Up a Web server ... 258
Downloading and publishing the package repository for Cloudera Manager ... 259
Creating a Temporary Internal Repository ... 260
Configuring Hosts to Use the Internal Repository ... 260

Configuring a Local Parcel Repository ... 261
Using an Internally Hosted Remote Parcel Repository ... 261
Setting Up a Web Server ... 261
Downloading and Publishing the Parcel Repository ... 263
Configuring Cloudera Manager to Use an Internal Remote Parcel Repository ... 263
Using a Local Parcel Repository ... 263

CDH 6 to Cloudera Base on premises post-upgrade transition steps ... 264
Update permissions for Replication Manager service ... 264
Migrating Spark workloads to Cloudera ... 265
Spark 2.3 to Spark 2.4 Refactoring ... 265
Apache Hive Expedited Migration Tasks ... 271
Preparing tables for migration ... 272
Creating a list of tables to migrate ... 273
Migrating tables to CDP ... 273
Apache Hive Changes in CDP ... 274
Preparing tables for migration ... 274
Hive Configuration Property Changes ... 276
LOCATION and MANAGEDLOCATION clauses ... 284
Handling table reference syntax ... 285
Key semantic changes and workarounds ... 285
Hive unsupported interfaces and features ... 290
Changes to CDH Hive Tables ... 292
Apache Hive Post-Upgrade Tasks ... 293
Customizing critical Hive configurations ... 293
Setting Hive Configuration Overrides ... 293
Hive Configuration Requirements and Recommendations ... 294
Fixing the canary test after upgrading ... 296
Configuring HiveServer for ETL using YARN queues ... 297
Removing Hive on Spark Configurations ... 297

Configuring authorization to tables....298
Making the Hive plugin for Ranger visible....298
Setting up access control lists....299
Configure encryption zone security....300
Configure edge nodes as gateways....300
Spark integration with Hive....301
Configure HiveServer HTTP mode....301
Configuring HMS for high availability....301
Installing Hive on Tez and adding a HiveServer role....302
Updating Hive and Impala JDBC/ODBC drivers....305
Apache Impala changes in CDP....305
Set ACLs for Impala....308
Impala Configuration Changes....311
Interoperability between Hive and Impala....313
Revert to CDH-like Tables....314
Authorization Provider for Impala....314
Data Governance Support by Atlas....315
Handling Data Files....316
Hue post-upgrade tasks....316
Updating group permissions for Hive query editor....317
Adding Security Browser to the blocked list of applications....317
Adding Query Processor service to a cluster....317
Importing Sentry privileges into Ranger policies....318
Apache Knox - create plugin audit directory....319
Apache Ranger TLS Post-Upgrade Tasks....320
Migrating ACLs from Key Trustee KMS to Ranger KMS....320
Key Trustee KMS operations not supported by Ranger KMS....324
ACLs supported by Ranger KMS and Ranger KMS Mapping....324
Apache Hadoop YARN default value changes....326
Upgrade Notes for Apache Kudu 1.15 / CDP 7.1....326
Apache HBase post-upgrade tasks....327
Configure SMM to monitor SRM replications....328
Configure SMM's service dependency on Schema Registry....328
Apache Sqoop Changes....329
Check Parquet writer implementation property....330
Configure a Sqoop Action globally and for all Hue workspaces....330
Cloudera Search changes....331

Applications Upgrade....332

Upgrading CDH 6 to Cloudera Private Cloud Base

High-level upgrade procedures for upgrades from CDH to Cloudera Private Cloud Base.

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.

Upgrading Cloudera Private Cloud Base consists of two major steps, upgrading Cloudera Manager and upgrading the cluster. You are not required to upgrade Cloudera Manager and the cluster at the same time, but the versions of Cloudera Manager and the cluster must be compatible. The major+minor version of Cloudera Manager must be equal to or higher than the major+minor version of CDH or Cloudera Runtime.

Workflow

An upgrade from CDH 6 to CDP Private Cloud Base has the following high-level workflow:

1. Prepare to upgrade:
   a. Review the Supported Upgrade Paths for your upgrade.
   b. Review the Requirements and Supported Versions for your upgrade
   c. Review the Migrating Navigator to Atlas for Cloudera Private Cloud with Cloudera Manager 6 for upgrading from Cloudera Manager 6 + CDH 6 to Cloudera Runtime 7.1.9 / Cloudera Manager 7.11.3 Latest cumulative hotfix.
   d. Review the Migrating Navigator to Atlas for Cloudera Private Cloud with Cloudera Manager 7 for upgrading from Cloudera Manager 7 + CDH 6 to Cloudera Runtime 7.1.9 / Cloudera Manager 7.11.3 Latest cumulative hotfix.
   e. Review the Release Notes for the version of Cloudera Private Cloud Base you are upgrading to.
   f. Gather information on your deployment. See Step 1: Getting Started Upgrading Cloudera Manager 6 on page 67 and Step 1: Getting Started Upgrading a Cluster on page 122.
   g. If you have Cloudera Flow Management installed, verify its compatibility with your target CDP version before starting the upgrade process. For. more information, see the Cloudera Flow Management Upgrade and migration paths documentation.
   h. Plan how and when to begin your upgrade.
2. If necessary, Upgrade the JDK.
3. If necessary, Upgrade the Operating System.
4. Perform any needed pre-upgrade transition steps for the components deployed in your clusters. See Cloudera Base on premises Pre-upgrade transition steps on page 14
5. Upgrade Cloudera Manager to version 7.4.4 or higher. After upgrading to Cloudera Manager 7.4.4 or higher, Cloudera Manager can manage upgrading your cluster to a higher version. See Upgrading Cloudera Manager 6 on page 66.
6. Use Cloudera Manager to Upgrade CDH to Cloudera Runtime 7, or from Cloudera Runtime to a higher version of Cloudera Runtime. See Upgrading a CDH 6 Cluster on page 121.
7. Perform any needed post-upgrade transition steps for the components deployed in your clusters. See CDH 6 to Cloudera Base on premises post-upgrade transition steps on page 264.

Note: You can retain older parcels for 30 days in case it's necessary to revert an upgrade, and then delete the older parcels.

Component Changes in CDP Private Cloud Base 7

YARN Fair Scheduler is being removed.

The YARN Fair Scheduler is being replaced with the YARN Capacity Scheduler. A transition tool will be provided to convert the Fair Scheduler configurations to Capacity Scheduler.

Hive-on-Spark and Hive-on-MapReduce have been removed. Similar functionality is available with Hive-on-Tez.

Pig, Flume, Sentry, and Navigator have been removed.

- Pig can be replaced with Hive or Spark.
- Flume has been replaced with Cloudera Flow Management (CFM). CFM is a no-code data ingestion and management solution powered by Apache NiFi. Contact your Cloudera account team for more information about moving from Flume to CFM.
- Sentry has been replaced with Ranger. A Sentry-to-Ranger policy transition tool is available for CDP Private Cloud Base 7.1 and transitions will be supported when Replication Manager is used to transition Hive tables from CDH to CDP.
- Navigator has been replaced with Atlas. Navigator lineage data is transferred to Atlas as part of the CDH to CDP Private Cloud Base upgrade process. Navigator audit data is not transferred to Atlas.

Assessing the Impact of an Upgrade

Understanding the impact of an upgrade.

Plan for a sufficient maintenance window to perform an upgrade. Depending on which components you are upgrading, the number of hosts in your cluster, and the type of hardware, you might need up to a full day to upgrade your cluster. Before you begin the upgrade, you need to gather some information; these steps are also detailed in the upgrade procedures.

Important: Cloudera recommends that you test upgrades on non-production clusters before upgrading your production clusters.

There are three types of upgrades: major, minor, and maintenance:

Major Upgrades

Major upgrades include the following:

- From Cloudera Manager 5.x or 6.x and CDH 5.x or 6.x to Cloudera Manager and Cloudera Runtime 7.1.1 or higher
- From Cloudera Manager and Cloudera Runtime 7.0.3 to Cloudera Manager and Cloudera Runtime 7.2 (Cloudera Base on premises)
- From Cloudera Manager 6.x to Cloudera Manager 7.1.1

A major upgrade typically has the following characteristics:

- Large changes to functionality and update of Hadoop to a more recent version
- Incompatible changes in data formats
- Significant changes and additions to the user interface in Cloudera Manager
- Database schema changes for Cloudera Manager that are automatically handled by the upgrade process
- Significant down time is required to upgrade the cluster.
- Client Configurations are redeployed.

Minor Upgrades

Minor upgrades upgrade your software to a higher minor version of a major release—for example from version 7.1.0 to version 7.2.0—and typically include the following:

- New functionality
- Bug fixes
- Potential database schema changes for Cloudera Manager that are handled automatically
- Client Configurations are redeployed.

Incompatible changes or changes to data formats are generally not introduced in minor upgrades.

Patch Upgrades

Patches fix critical bugs or address security issues. The version numbers for maintenance releases differ only in the fourth digit, for example, when upgrading from version 7.1.3 to 7.1.4.

How much time should I plan for to complete my upgrade?

An in-place upgrade can take a variable amount of time to complete. Learn about how to plan for and shorten the amount of time required for your upgrade.

The amount of time required for an in-place upgrade depends on many factors, including:

- The number of hosts in your clusters.
- The mix of services you have deployed in your clusters.
- The amount of data stored in your clusters.

Generally, an upgrade can be completed in 24-48 hours. Upgrades from HDP to Cloudera may take somewhat longer due to the Ambari to Cloudera Manager conversion process (AM2CM).

The following table provides some additional information to help you plan for your upgrade.

Table 1: Upgrade Time Planning

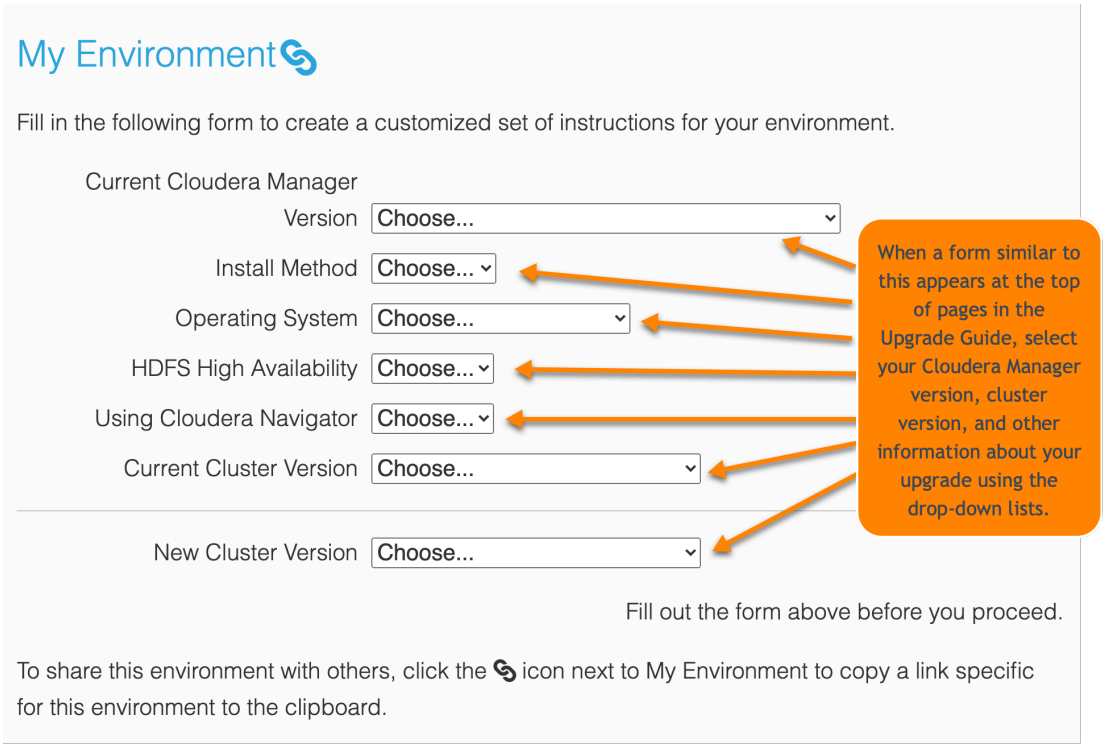
Component/Process	Notes
Cloudera Runtime Parcel	The Cloudera Runtime parcel must be distributed to all hosts before upgrading the hosts. Downloading the parcel directly from archive.cloudera.com over the internet may add additional time. You can download the parcels and serve them from a local web server to decrease this time. In addition, after downloading the parcels to a local repository, you can distribute them in advance of launching the upgrade wizard to save additional time.
Cloudera Manager	You must upgrade Cloudera Manager before upgrading your clusters. Cloudera Manager can continue to mange older versions of Cloudera Runtime and CDH until the upgrade.
Cluster cold start	The cluster will need to be restarted at least once during an in-place upgrade. The amount of time required for a restart depends on how many files and blocks are stored in the cluster and the number of hosts in the cluster.
Navigator to Atlas Migration	Depending on the amount of data, this can take a significant amount of time. See Transitioning Navigator content to Atlas Transitioning Navigator content to Atlas
Hive	The Hive strict managed migration process can take a significant amount of time. See for more information about mitigating that impact. See Understanding the Hive upgrade (CDH) Understanding the Hive upgrade (CDH)
HBase checks	While Running HBase checks does not take significant time, remediating any issues can take significant time. To save time during the upgrade, you can plan to do this before running the Upgrade Wizard.
Sentry to Ranger migration	This process runs quickly and usually takes less than 20 minutes.
Solr export/backup	This process depends on how much data has to be imported after the upgrade.

About using this online Upgrade Guide

How to fill in forms to customize the documentation for your upgrade.

This online version of the Cloudera Upgrade Guide allows you to create a customized version of the guide on many pages that only includes the steps required for your upgrade. Use the My Environment form at the top of pages in this guide to select the Cloudera Manager, CDH or Cloudera Runtime version for your upgrade as well as the operating system version, database type, and other information about your upgrade. After making these selections, the pages in the guide will only include the required steps for your upgrade. The information you enter is retained on each page in the guide.

Figure 1: My Environment Form Example



Note: The HDP upgrade procedures do not include a My Environment form at the top of the page.

Cloudera Base on premises Pre-upgrade transition steps

The following procedures must be completed before performing a cluster upgrade to Cloudera Base on premises (cluster version Cloudera Runtime 7.4.4 or higher). Only complete the procedures for services running in your source cluster.

Set log level for KeyTrustee KMS to INFO

Reduce the log output from org.apache.ranger.plugin.* by changing the log level setting for your Ranger KMS from DEBUG to INFO.

About this task

Upgrading a CDH cluster to CDP includes converting KeyTrustee KMS to Ranger KMS.

In some rare cases, the KeyTrustee KMS logging may be set to DEBUG level when investigating services issues. When KeyTrustee KMS is converted to Ranger KMS during a CDH to CDP upgrade, some configuration settings, such as the log_threshold setting, may be transferred over. While log_level set to DEBUG minimally impacts CDH clusters, clusters upgraded to CDP may experience a negative performance impact from Ranger KMS if the log_threshold setting remains at DEBUG.

Recommended Practice: leave the log_threshold setting configured to INFO or higher, unless actively debugging a service issue in Ranger KMS.

Setting the log_threshold to DEBUG on Ranger KMS can produce a huge number of log entries from org.apache.ranger.plugin.*. Due to the frequency of logs generated, the Ranger KMS can experience periods of slow response, negatively impacting file operations on HDFS.

Note: Additional information:

For CDH5 clusters, setting log level to DEBUG does not afftect Tomcat logging.

For CDH6 clusters, Jetty embedded inside KMS magnifies this issue.

Procedure

1. During pre-upgrade, review the logging level of your KeyTrustee KMS service.
2. Make sure that DEBUG/TRACE is not enabled for KeyTrustee KMS.
3. During post-upgrade, review the logging level of your Ranger KMS service.
4. Make sure that DEBUG/TRACE is not enabled for Ranger KMS.

What to do next

Complete additional pre-upgrade tasks.

Transitioning from Sentry Policy Files to the Sentry Service

If your cluster uses Sentry policy file authorization, you must transition the policy files to the database-backed Sentry service before you upgrade to CDH 6 or Cloudera Base on premises 7.1.

Complete the following steps to upgrade from Sentry policy files to the database-backed Sentry service:

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

1. Disable the existing Sentry policy file for any Hive, Impala, or Solr services on the cluster. To do this:
   a. Go to the Hive, Impala, or Solr service.
   b. Click the Configuration tab.
   c. Select Scope *SERVICE NAME* (Service-Wide).
   d. Select CategoryPolicy File Based Sentry.
   e. Clear Enable Sentry Authorization using Policy Files. Cloudera Manager throws a validation error if you attempt to configure the Sentry service while this property is checked.
   f. Repeat for any remaining Hive, Impala, or Solr services.
2. Add the new Sentry service to your cluster. For instructions, see Installing and Upgrading the Sentry Service.
3. To begin using the Sentry service, see Configuring the Sentry Service

4. (Optional) Use command line tools to transition existing policy file grants.
   - If you want to transition existing Sentry configurations for Solr, use the solrctl sentry --convert-policy-file command, described in solrctl Reference.
   - For Hive and Impala, use the command-line interface Beeline to issue grants to the Sentry service to match the contents of your old policy file(s). For more details on the Sentry service and examples on using Grant/Revoke statements to match your policy file, see Hive SQL syntax for use with Sentry.
5. Restart the affected services to apply the changes.

Transitioning the Sentry service to Apache Ranger

Before transitioning your cluster to Cloudera Base on premises, you must prepare the Apache Sentry authorization privileges so they can be converted to Apache Ranger permissions. Apache Ranger supports the components like HDFS, Hive, and YARN. Apache Ranger functions as a centralized security administrator and provides greater access controls and auditing capabilities.

Warning: Before upgrading CDH 5 clusters with Sentry to Cloudera Runtime 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.

Important: If HDFS ACL sync is enabled (hdfs_sentry_sync_enable=true) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see Installing Ranger RMS.

Perform the following steps after you have upgraded Cloudera Manager to version 7.1 or higher:

1. Verify that the HDFS service is in the Start state.

   Starting from Cloudera Manager 7.4.4, the Export Sentry Permissions command is executed as part of the upgrade flow that requires the HDFS service to be in the start state.

   If you are using Cloudera Manager 7.3.1, 7.2.4, or any Cloudera Manager 7.1.x version, go to the Sentry service and select Actions > Export Permissions to export the sentry permissions.
2. Make sure a MySQL, Oracle, or PostgreSQL database instance is running and available to be used by Ranger before you create a new cluster or upgrade your cluster from CDH to Cloudera Runtime. See the links below for procedures to set up these databases.

   Important: The Ranger database should not be shared with other services or applications.
3. After you have set up the database, you can continue upgrading the cluster.

After upgrading Cloudera Manager and the cluster, you must import Sentry privileges using Ranger so that Sentry privileges translate to Ranger service policies. For more information about completing this translation process, see Importing Sentry privileges into Ranger policies on page 318.

Warning: The automated translation process does not manage Solr permissions. You must translate Solr permissions manually. For more information, see Mapping Sentry permissions for Solr to Ranger policies.

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.

Configuring a Ranger or Ranger KMS Database: MySQL/MariaDB

Prior to upgrading your cluster to Cloudera Base on premises 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 Database: PostgreSQL

Prior to upgrading your cluster to Cloudera Base on premises you must configure the PostgreSQL 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 PostgreSQL must be running and available to be used by Ranger. See Install and Configure PostgreSQL for CDP.

Procedure

1. Log in to the host where you want to set up the PostgreSQL database for Ranger.
2. On the PostgreSQL host, install the applicable PostgreSQL connector:

   RHEL/CentOS/Oracle Linux

   ```
   yum install postgresql-jdbc*
   ```

   SLES

   ```
   zypper install -y postgresql-jdbc
   ```

3. Edit the pg_hba.conf file, located either in the /var/lib/pgsql/data or /etc/postgresql/<version>/main directory and add the following line:

   ```
   host all all 127.0.0.1/32 md5
   ```

   If this file contains the line host all all 127.0.0.1/32 ident, then
4. Edit the /var/lib/pgsql/data/postgresql.conf file and add the following line if it is not already there:

   ```
   listen_addresses='*'
   ```

5. Enable the PostgreSQL server to start automatically on boot-up:

   ```
   sudo systemctl enable postgresql
   ```

6. Restart the PostgreSQL server:

   ```
   sudo systemctl restart postgresql
   ```

7. Log in to PostgreSQL:

   ```
   sudo -u postgres psql postgres
   ```

8. Create the Ranger database and user. Run the following commands:

   ```
   create user rangeradmin with createdb login password 'rangeradmin';
   create database ranger with owner rangeradmin;
   ```

```
GRANT ALL PRIVILEGES ON SCHEMA public TO rangeradmin;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO rangeradmin;
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO rangeradmin;
```

9. Use the \q command to exit PostgresSQL.

What to do next

Continue installing or upgrading your cluster.

Configuring a Ranger or Ranger KMS Database: Oracle

Prior to upgrading your cluster to Cloudera Base on premises 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. On the Ranger host, install the appropriate JDBC .jar file.
   a) Download the Oracle JDBC (OJDBC) driver from https://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.
      - For Oracle Database 12c: select Oracle Database 12c Release 2 driver > ojdbc8.jar.
   b) Copy the .jar file to the Java share directory.

      ```
      cp ojdbc8-12.2.0.1.jar /usr/share/java/
      ```

      Make sure the .jar file has the appropriate permissions. For example:

      ```
      chmod 644 /usr/share/java/ojdbc8-12.2.0.1
      ```

2. Log in to the host where the Oracle database is running and launch Oracle sqlplus:

   ```
   sqlplus sys/root as sysdba
   ```

3. Create the Ranger database and user. Run the following commands:

   ```
   CREATE USER rangeradmin IDENTIFIED BY rangeradmin;
   GRANT SELECT_CATALOG_ROLE TO rangeradmin;
   GRANT CONNECT, RESOURCE TO rangeradmin;
   QUIT;
   GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,CREATE SEQ
   UENCE,CREATE PUBLIC SYNONYM,CREATE ANY SYNONYM,CREATE TRIGGER,UNLIMITED
   TABLESPACE TO rangeradmin;
   ALTER USER rangeradmin DEFAULT TABLESPACE <tablespace>;
   ALTER USER rangeradmin quota unlimited on <tablespace>;
   ```

What to do next

Continue installing or upgrading your cluster.

Import Key Trustee KMS ACLs to Ranger KMS policies

This topic is intended to provide pre-upgrade checks about auth_to_local rules when importing Key trsutee KMS ACLs to Ranger KMS policies.

About this task

The "cm_kms" plugin repository for the Ranger KMS service is not present in the Ranger Admin UI for the "keyadmin" user.The role assigned to the "rangerkms" user is a non-admin role. Non-admin roles are not allowed to create, read, update, delete the policies in cm_kms plugin repository. By default, we set auth_to_local rules to convert rangerkms principal to user name 'keyadmin' and assign KeyAdmin role in ranger.

Procedure

1. If custom rules are set in the CDH cluster, add the following two rules to HDFS auth_to_local configuration, prior to upgrade.
   a) RULE:[2:$1@$0](rangeradmin@SUPPORT.COM)s/(.*)@SUPPORT.COM/ranger/
   b) RULE:[2:$1@$0](rangerkms@SUPPORT.COM)s/(.*)@SUPPORT.COM/keyadmin/
2. If above rules are not added prior to the upgrade, log in to Ranger Admin UI as KeyAdmin user, and add KeyAdmin role to 'rangerkms' user. :
   a) Go to Ranger UI Security Users/Groups/Roles > "rangerkms" > Assign "Key Admin" role.

Transitioning Navigator content to Atlas

During the transition from CDH to Cloudera Private Cloud base, you can transition the metadata from Navigator to Apache Atlas, for a scalable and robust infrastructure that supports enhanced metadata searchability by collecting of metadata directly from your cluster.

Cloudera Runtime 7 includes Apache Atlas to collect technical metadata from cluster services. Atlas replaces Cloudera Navigator Data Management for these clusters. Cloudera has incorporated many features from Navigator into Apache Atlas to make sure that the rich metadata collected in Navigator can be represented in Atlas. Atlas provides scalable and robust infrastructure that supports metadata searches and lineage across enterprise production clusters.

Note: Governance through Apache Atlas is just one element of a secure production cluster: Cloudera supports Atlas when it runs on a cluster where Kerberos is enabled to authenticate users. When upgrading from to Cloudera Runtime 7.1.1 or later and running Apache Atlas, the new cluster must have Kerberos enabled.

You may choose not to transition Navigator content to Atlas at all: this document describes how to think about archiving your Navigator audits and metadata.

Whether you choose to transition Navigator contents to Atlas or not, this document describes how to use Atlas to accomplish the tasks you are accustomed to performing in Navigator.

What's transitioned?

Business metadata is transitioned into Atlas, including:

- Tags
- Custom properties (definitions and entity assignments)
- Managed metadata properties (definitions and entity assignments)
- Original and updated entity names and descriptions

Technical metadata from the following sources are transitioned into Atlas:

- Hive
- Impala
- Spark
- Referenced HDFS / S3

What's NOT transitioned?

- Audits. In Cloudera, Apache Ranger collects audit information for successful and failed access to objects under its control. This audit system is focused and powerful, but it's enough different from how Navigator collected audits that transition isn't appropriate. This document includes information on how to transition your auditing to Ranger and how to archive your existing Navigator audit information.
- Entity Metadata. The following metadata entities in Navigator are not transitioned to Atlas:
  - Unreferenced S3 and HDFS entities. Files in HDFS and S3 that are not included in lineage from Hive, Spark, or Impala entities are not transitioned.
  - Metadata for Sqoop, Pig, Map-Reduce v1 and v2, Oozie, and YARN.
- Policies. Navigator policies are not transitioned to Atlas.
- Configuration settings. Configuration properties you've set in Cloudera Manager that determine Navigator behavior are not transitioned to the new environment. If you have properties that may apply in Atlas, such as authentication credentials, you'll need to reset them in the new environment.

Will Navigator still run in Cloudera Manager?

After upgrading Cloudera Manager, Navigator continues to collect metadata and audit information from CDH cluster services. There are no changes to Navigator functionality; all Navigator data is retained in the Cloudera Manager upgrade.

After upgrading a CDH cluster, services that previously sent metadata and audit information to Navigator, such as Hive, Impala, Spark, and HBase, are configured to pass metadata to Atlas. Navigator audit collection for those services is disabled. You can still access audits and metadata through Navigator; however, Navigator will not collect new information from cluster services. When you determine that you have maximized the value of the Navigator audits and successfully converted Navigator metadata to Atlas content, you can proceed to disable Navigator servers. Afterward, you may proceed with removing the role from the cluster.

High-level transition process

Before transitioning from Navigator to Apache Atlas, review the transition paths. You must extract, transform, and import the content from Navigator to Apache Atlas. After the transition, services start producing metadata for Atlas and audits for Ranger.

There are two main paths that describe a Navigator-to-Atlas transition scenario:

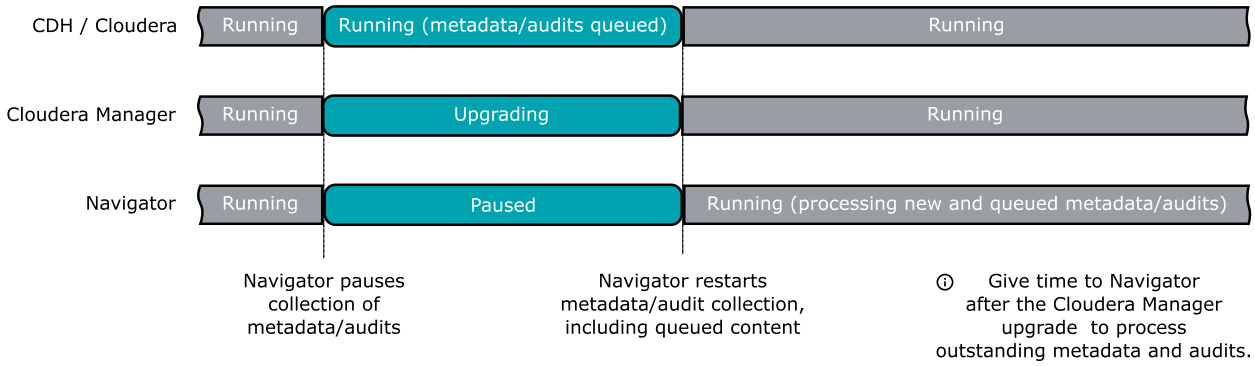
- Upgrading Cloudera Manager to CDP 7 and upgrading all of your CDH clusters to Cloudera Runtime.

  In this case, you can stop Cloudera Navigator after migrating its content to Atlas.
- Upgrading Cloudera Manager to CDP 7 but managing some or all of your existing CDH clusters as CDH 5.x or 6.x.

  In this case, CDP runs Cloudera Navigator to continue extracting metadata and audits from existing CDH clusters and runs Atlas and Ranger to support metadata and audit extraction from new or potential new Cloudera Runtime clusters.

In both scenarios, you'll complete the upgrade of Cloudera Manager first. While Cloudera Manager is upgrading, Navigator pauses collection of metadata and audits from cluster activities. After the upgrade is complete, Navigator processes the queued metadata and audits.

In the timeline diagrams that follow, the blue color indicates steps that because you trigger them manually, you can control their timing.



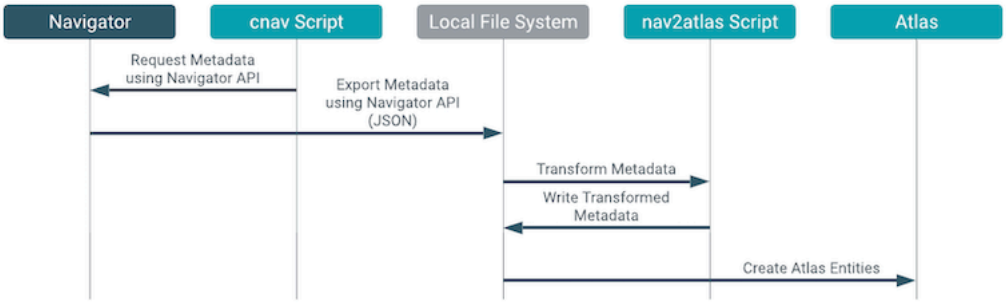
The transition of Navigator content to Atlas occurs in the upgrade from CDH to CDP. The transition involves three phases:

- Extracting metadata from Navigator.

  The Atlas installation includes a script (cnav.sh) that calls Navigator APIs to extract all technical and business metadata from Navigator. The process takes about 4 minutes per one million Navigator entities. The script compresses the result and writes it to the local file system on the host where the Atlas server is installed. Plan for about 100 MB for every one million Navigator entities; lower requirements for larger numbers of entities.
- Transforming the Navigator metadata into a form that Atlas can consume.

  The Atlas installation includes a script (nav2atlas.sh) that converts the extracted content and again compresses it and writes it to the local file system. This process takes about 1.5 minutes per million Navigator entities. The script compresses the results and writes it to the local file system on the host where the Atlas server is installed. Plan for about 100 to 150 MB for every one million Navigator entities; higher end of the range for larger numbers of entities.
- Importing the transformed metadata into Atlas.

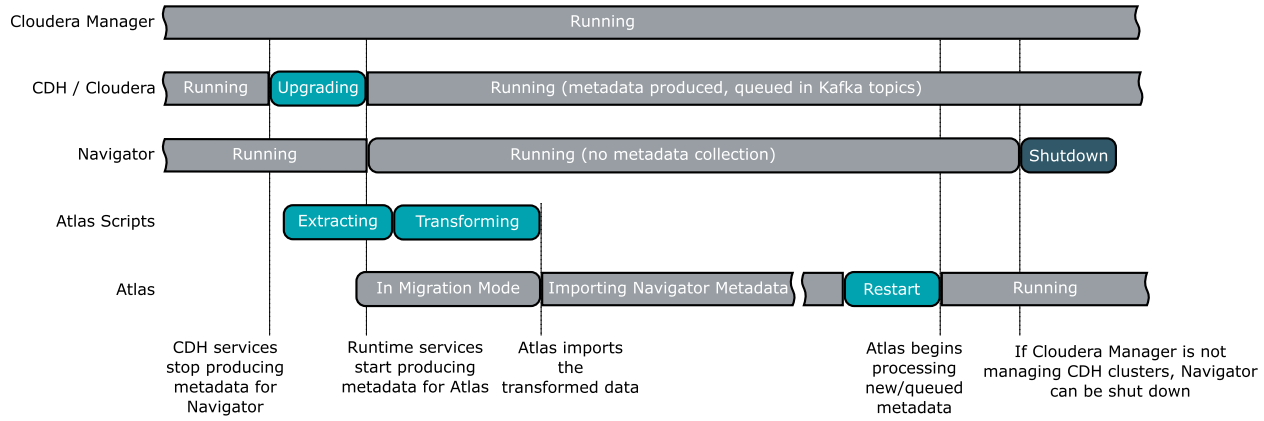
  After the CDP upgrade completes, Atlas starts up in "migration mode," where it waits to find the transformed data file and does not collect metadata from cluster services. When the transformation is complete, Atlas begins importing the content, creating equivalent Atlas entities for each Navigator entity. This process takes about 35 minutes for each one million Navigator entities, counting only the entities that are migrated into Atlas.



To make sure you don't miss metadata for cluster operations, give time after the CM upgrade and before the CDH upgrade for Navigator to process all the metadata produced by CDH service operations. See Navigator Extraction Timing for more information.

You can start extracting metadata from Navigator as soon as the Cloudera parcel is deployed on the cluster. After CDP is started, Navigator no longer collects metadata or audits from the services on that cluster; instead services produce metadata for Atlas and audits for Ranger.

Important: After the CDH upgrade, Atlas starts in migration mode and does not process metadata. When the transition completes, you must manually update the Atlas configuration in Cloudera Manager to have Atlas begin processing metadata.



The following topics describe the details of the events in these timelines:

Assumptions and prerequisites

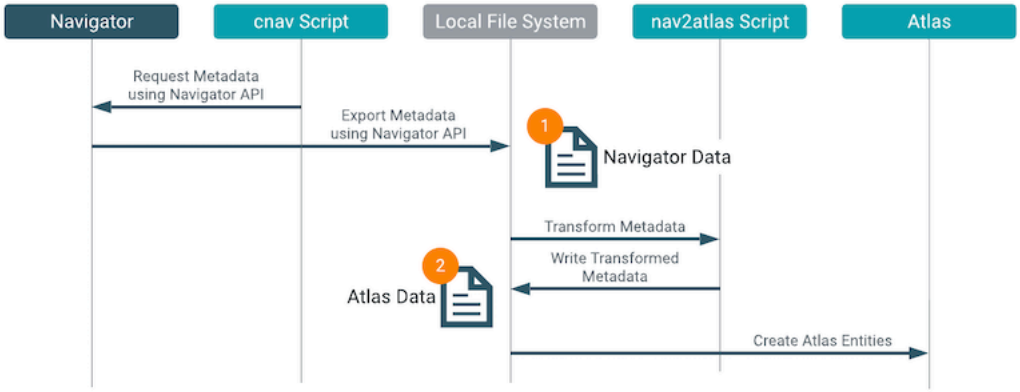
Before you transition your cluster to CDP Private Cloud Base or migrating content from Navigator to Apache Atlas, ensure that you have collected all the credentials and set expectations for the time required for completing the transition. The prerequisites in this section help you to prepare in advance to transition.

In addition to the prerequisites outlined for the Cloudera Manager and Cloudera upgrades, you'll need the following for the Navigator to Atlas transition:

- Deleted entities in Navigator. Check the Navigator Administration page to make sure that a successful purge has run recently. If it hasn't, consider running a purge before the transition. See Managing Metadata Storage with Purge.
- Role to host assignments. Before you begin upgrading to Cloudera, make a plan for where you will install the Atlas server. In addition, Atlas depends upon HBase, Kafka, and Solr services; your plan should include host assignments for installing the components of these services. See Runtime Cluster Hosts and Role Distribution.
- Resources for Atlas service. Atlas requires 16 GB of Java heap (Atlas Max Heapsize property) and 4 Solr shards (Initial Solr Shards for Atlas Collections property). Make sure the host you choose for Atlas has enough resources for all the services' requirements.

  Attention: You must note about the default values for Initial Solr Shards for Atlas Collections in your Cloudera Manager UI. Before you commence the Atlas initialization process, based on your performance requirements, you must decide the actual (correct) values for Initial Solr Shards for Atlas Collections in your Cloudera Manager instance. Cloudera recommends to set 4 Solr shards (Initial Solr Shards for Atlas Collections property). You must also note that, you must not update or modify these values once the Atlas initialization has commenced. Additionally, note that once the Atlas initialization process is completed, modifying the value of Initial Solr Shards for Atlas Collections or Initial Solr Replication Factor for Collections will not have any effect on the collections for Atlas in Solr.
- Resources for Solr service. During transition, Solr running to serve as Atlas' index requires 12 GB of Java heap (Java Heap Size of Solr Server in Bytes property). You can reset this back to make sure the host you choose for Atlas has enough resources for all the services' requirements.
- Navigator credentials. The transition requires the username and password for a Navigator user with administrator privileges.
- Local disk space needed for intermediate processing. The first two phases of the Navigator-to-Atlas transition produce intermediate files in /tmp in the local file system where Atlas is installed. See Estimating the time and resources needed for transition on page 24.

- Local disk space for transition staging files. The first two phases of the Navigator-to-Atlas transition produce staging files on the local disk where Atlas is installed. See Estimating the time and resources needed for transition on page 24.



- Time estimates for transition phases. Each phase of the transition runs independently from the upgrade. You can trigger them to run when convenient. See Estimating the time and resources needed for transition on page 24.

Estimating the time and resources needed for transition

While the cluster is starting up, you can plan for and start the transition process.

1. Inspect Navigator installation to determine the number of Navigator entities that will be transitioned. See How many Navigator entities are transitioned? on page 25
2. Estimate the time and disk space required for each phase of the transition.

   The following transition rates are approximate and depend on the resources available on the Atlas host and other unknown factors. Note that the number of entities actually imported may be considerably less that the number of entities extracted. The transition process discards HDFS entities that are not referenced by processes that are transitioned (Hive, Impala, Spark).

Transition Phase	Transition Rate	Disk Space	Output File Size	Trial Data Points
Extraction	4 minutes / 1 million entities	100 MB / 1 million entities, less as volumes increase	65 MB / 1 million entities	10 million entities takes about 30 minutes; 256 million takes about 18 hours.
Transformation	1.5 minutes / 1 million entities	100 to 150 MB / 1 million entities, higher end of range with larger volumes	150 MB / 1 million entities	10 million entities takes about 20 minutes; 256 million takes about 6 hours.
Import	35 minutes / 1 million migrated entities	N/A	N/A	10 million entities takes about 4 hours; 256 million takes about 6 days.

Migration from Navigator to Atlas can be run only in non-HA mode

Migration import works only with a single Atlas instance.

If Atlas has been set up in HA mode before migration, you must remove the additional instances of Atlas, so that Atlas service has only one instance.

Later, start Atlas in the migration mode and complete the migration. Perform the necessary checks to verify if the data has been imported correctly.

Restart Atlas in non-migration mode.

- If you have Atlas setup in HA mode, retain only one instance and remove the others.

- Ensure that the ZIP files generated as output of Nav2Atlas conversion are placed at the same location where the Atlas node is present.

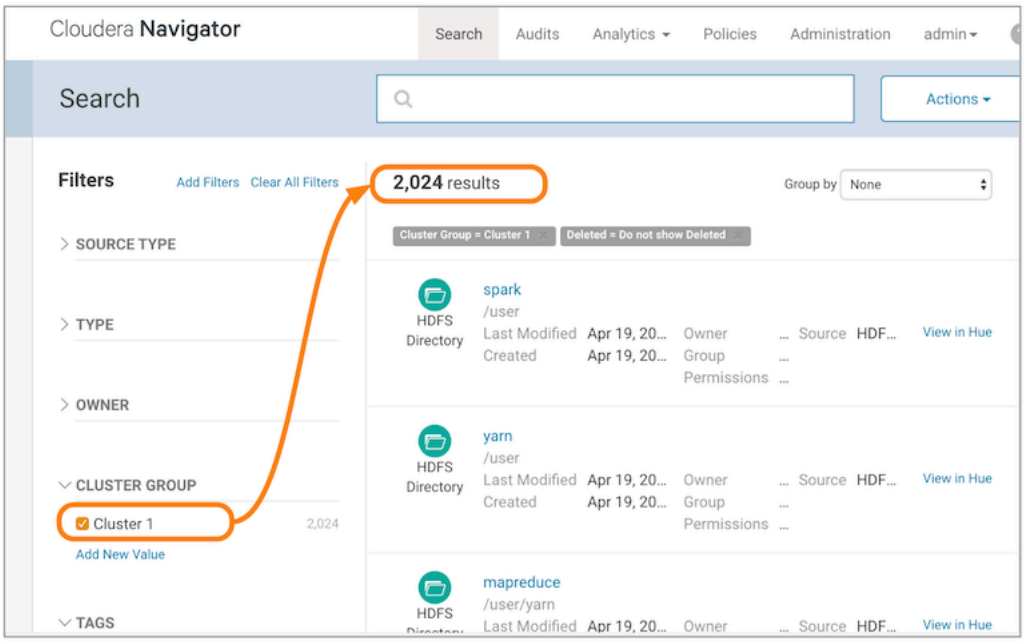
How many Navigator entities are transitioned?

When preparing to transition content from Navigator to Atlas, it helps in planning the transition to know how many Navigator entities will be extracted. Use Navigator's search facets to figure this out.

To determine the number of Navigator entities extracted for extraction and transformation phases of the transition:

1. Log into Navigator.
2. In the Cluster Group facet in the left panel, select the cluster you are migrating from.

   The main panel displays the count of entities in that cluster. Use this value for estimating the extraction and transformation phase durations.



Not all Navigator entities are imported into Atlas. To estimate the subset of entities included in the import phase:

1. Log into Navigator.
2. In the Cluster Group facet in the left panel, select the cluster you are migrating from.
3. In the Source Type facet in the left panel, select "Hive", "Impala", and "Spark".

   The main panel displays the count of entities in from these sources in this cluster.
4. Double the number from the search results to account for the physical files that correspond to the tables and jobs. The HDFS entities referenced by the Hive, Impala, and Spark entities are included in the transition.

The transition brings over all business metadata definitions and associations with transitioned entities. To determine the number of Navigator managed properties to transition:

1. Log into Navigator.
2. In the left Search panel, find the Tags facet.

   This facet lists all the tags defined in Navigator. Navigator tags are imported into Atlas as labels.
3. Go to AdministrationManaged Properties.

   The Navigator namespaces are imported as Atlas business metadata collections. Each managed property is imported as a business metadata attribute.

Installing Atlas in the Cloudera Manager upgrade wizard

You can use the wizard in Cloudera Manager to transition from CDH to , install Apache Atlas and its dependencies, and to generate customized commands to initiate the migration phases. After the wizard is complete, you can run the migration commands from the host where Atlas is installed.

This page describes the Cloudera Manager wizard steps that help you to setup Atlas and its service dependencies.

Attention: Before you perform the upgrade process, note the following pertaining to Atlas operations.

When a cluster is upgraded from CDH 5 or CDH 6 to CDP 7, by default the atlas.metadata.namespace property is set to cm.

If a different namespace property needs to be set, for example: cluster1, the same needs to be set while running the nav2atlas script as "-clusterName" parameter and also in the atlas-application properties in the upgrade wizard.

Post-upgrade, note that the different value "cluster1" is not automatically updated in Hooks for services like Hive, HBase, Impala, Spark, and Kafka. You must make sure that before you upgrade your cluster and once the services are installed, you must set the value "cluster1" for all the available services. And later complete the upgrade process.

As an example process, follow these steps if namespace other than default "cm" needs to be set:

1. Provide the namespace ("cluster1") in the transformation step of Nav2Atlas script.
2. Add atlas.metadata.namespace =cluster1 in atlas-application properties in the following window of the upgrade wizard for Atlas.
3. Open another tab of the Cloudera Manager while upgrade process is in progress and add atlas.metadata.namespace =cluster1 in Safety Valve of atlas-application.properties for all the Hook services (Hive, HiveServer2 , Spark , HBase, Impala, and Kafka).
4. Perform all the other steps in the upgrade wizard and complete the upgrade.
5. Remove Atlas from the migration mode.

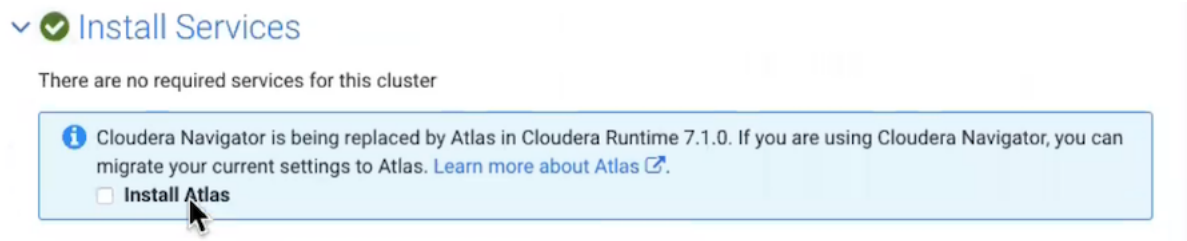
To return to the main wizard documentation, go to Upgrading a CDH 6 Cluster.

Follow instructions in the upgrade wizard "Install Services" section

See Step 10: Run the Upgrade Cluster Wizard.

- Enable Atlas install.

  If the CDH cluster being upgraded was running Navigator, the upgrade wizard shows a note recommending that you enable Atlas in the new cluster. Check the Install Atlas option.



- Install Atlas dependencies.

  The wizard steps through the installation for Atlas' dependencies, assuming these services haven't already been included in the installation:

  - ZooKeeper. Assign one or more hosts for the ZooKeeper role.
  - HDFS. Already included in the installation.
  - Kafka. Select the optional dependency of HDFS. Atlas requires configuring the Broker service only, not MirrorMaker, Connect, or Gateway.
  - HBase. Atlas requires configuring HBase Master and RegionServers only, not REST or Thrift Server. Assign a Master role on at least one host. Assign RegionServers to all hosts.
  - Solr. Assign a host for the Solr Server role. Set the Java Heap Size of Solr Server in Bytes property to 12 GB (to support the migration operation).

  For recommendations on where in the cluster to install the service roles, see Runtime Cluster Hosts and Role Distribution.

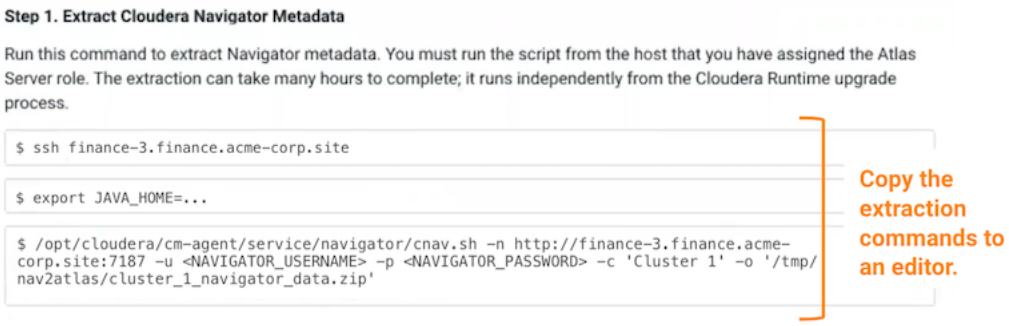
- Click Add Atlas Service. The wizard steps through choosing a host and setting migration details.
  - Set the host for the Atlas server roles and click Continue.

    Tip: Remember this host as you'll need to SSH to it later to trigger the content migration from Navigator.
  - The Atlas Migrate Navigator Data screen displays.

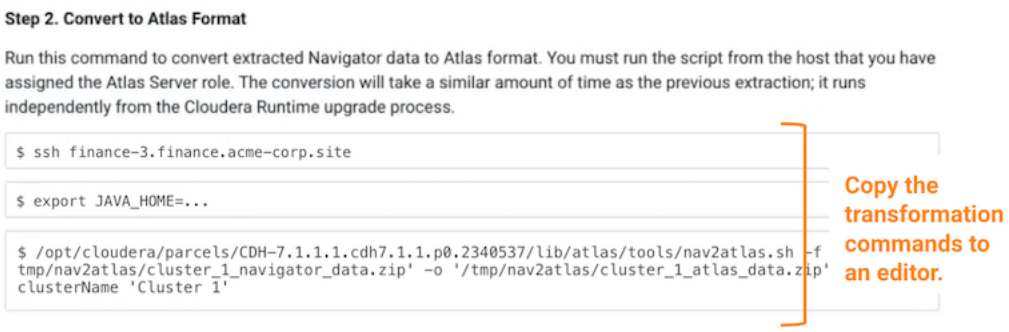
    This screen contains migration commands that are customized to your environment. When you fill in the output file paths, the command text changes to incorporate your settings.

    1. Set migration data-staging locations.

       The migration process creates two data files on the local file system on the host where Atlas is installed. Make sure there is enough disk space to hold these files; see Estimating the time and resources needed for transition on page 24.
    2. Copy the extraction command text to an editor.



    3. Copy the transformation command text to an editor.



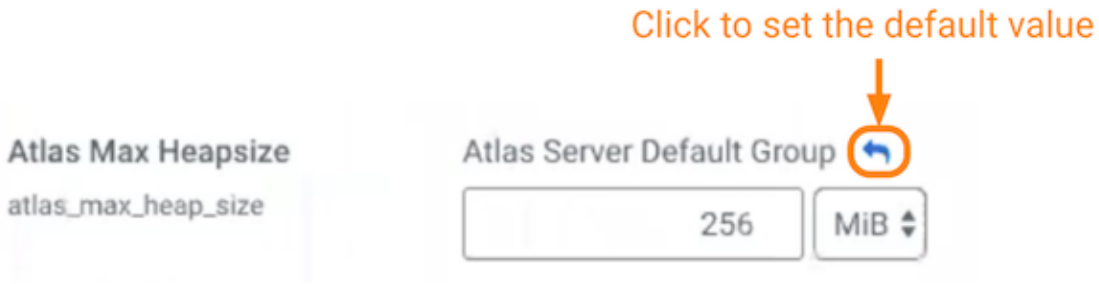
    Note: The transformed data - "cluster_1_atlas_data.zip", is imported to Atlas when started in Migration mode, however, there are edge cases where the transformed data is created as multiple part files (as cluster-atlas-data-1-of-3.zip, cluster-atlas-data-2-of-3.zip, cluster-atlas-data-3-of-3.zip) instead of "cluster_1_atlas_data.zip". As the part file name is different from the file name specified in "atlas.migration.data.filename=", Atlas will not import the file. To proceed further, use the following instructions.

    - Set Atlas safety value to: atlas.migration.data.filename=</path/to/cluster-atlas-data-1-of-3.zip>
    - Restart Atlas and launch Atlas UI to validate the import process and repeat the same steps to import all of the part files.

    4. Confirm the output file location. This is the location where Atlas will look for the content to import. Make sure it matches the location you plan to use for the output of the transformation command.
    5. Click Continue.
- The Atlas Enable Migration Mode screen displays. Review the Atlas Safety Valve content and click Continue.

  After the migration is complete, you will manually remove these settings to start Atlas in normal operation.

- The Atlas Review Changes screen displays. Review the configurations and make any necessary changes.You must provide a value for the following:
  - Admin Password – choose a password for the preconfigured admin user.
  - Atlas Max Heapsize – set the max heapsize to the default value by clicking the curved blue arrow. If you plan to migrate content from Cloudera Navigator to Atlas, consider setting the heapsize to 16 GB.



- Click Continue.

To complete the Navigator-to-Atlas migration outside of the CDP Runtime upgrade, see Transitioning Navigator data using customized scripts on page 29.

Continue with the upgrade wizard

The Cloudera Manager upgrade wizard continues with "Other Tasks" and "Inspector Checks" sections. The wizard steps for installing Atlas are complete at this point and you can continue to complete the Cloudera Runtime upgrade.

Related Information

Configure Atlas file-based authentication

Transitioning Navigator data using customized scripts

You can run the customized scripts generated by the Cloudera Manager wizard to configure the Apache Atlas installation and start the Navigator-to-Atlas data migration process when you step into the Cloudera upgrade wizard. You can also run the migration scripts independently from the Cloudera upgrade.

The transition has three phases: extraction, transformation, and import. If you haven't already, estimate the time and resource requirements for the migration steps as described in Assumptions and prerequisites.

Run the extraction

You can run the extraction in the background as soon as the Cloudera Runtime parcel is deployed. To customize and run the extraction command:

1. Go back to the editor where you saved the extraction commands, from the Copy the extraction command text substep from the step "Click Add Atlas Service."
2. Open a terminal window or command prompt where you have access to the cluster.
3. Using the provided command, SSH into the Atlas host.
4. Make sure the JAVA_HOME variable is set; if it isn't, run the export command pointing to the location of the JDK.
5. Customize the extraction command to include the Navigator admin user and password.
6. Run the extraction command.

   When the extraction is complete, you'll see a status message in the command output.

If Navigator is configured with TLS/SSL enabled, the cnav script needs the following credential information:

- Truststore path
- Truststore password
- Keystore path
- Keystore password

To make these parameters available, run the following commands before running the cnav script:

```
export KEYSTORE_PATH=<keystore-path>;
export KEYSTORE_PASSWORD=<keystore-password>;
export TRUSTSTORE_PATH=<truststore-path>;
export TRUSTSTORE_PASSWORD=<truststore-password>
```

For example, the command sequence might look similar to the following (line breaks are introduced for readability):

```
export KEYSTORE_PATH=/opt/cloudera/CMCA/trust-store/acme_truststore.jks;
export
KEYSTORE_PASSWORD=Q01FAeH53dn1HLY74D68Kk1yMAQVGtOI_cLznArccid48DDzS0VXY-DW
nzpz0Ug10BvikGMoovYaZT2EEEdBGgLPiDCRKHyzFExE3OITRGazjKPtZxAaXOUzgKMMmQQgJKw-
5JW9I6WgLGBhCpKfBa7vP3z6PFtm6XfYB-o3R6qmcdzZLwslDIQ18mowuFV1ouQIzA;
export TRUSTSTORE_PATH=/var/lib/cloudera-scm-agent/agent-cert/cm-auto-host_k
eystore.jks;
export TRUSTSTORE_PASSWORD=123420978alngdfdfjliaiu;
/opt/cloudera/cm-agent/service/navigator/cnav.sh -n https://acme-finance-
1.acme-finance:7187 -u admin -p adminpass -c "Cluster 2" -o /tmp/nav2atlas/n
av2altas_nav_export.zip
```

Note: The -c flag parameter in the nav2atlas script represents the Navigation Metadata Server (NMS) cluster name and NOT the Cloudera Manager cluster name. You must retrieve the value of -c from the Navigator UI to use it while running the nav2atlas script.

Important: While running Nav2Atlas data migration, if the file name given for the output is provided in the upper case, `cnav` converts it into lowercase and generates the zip. If you provide uppercase filename to Nav2Atlas script as -f parameter, it errors out with **FileNotFoundException** because of case sensitivity. Therefore, you must provide the filename for Nav2Atlas -f parameter with the same file output generated from the `cnav` command.

Run the transformation

You can run the transformation in the background as soon as the extraction completes. To customize and run the transformation command:

1. Go back to the editor where you saved the transformation commands, from the "Copy the transformation command text" substep from the step "Click Add Atlas Service."
2. If needed, open a terminal window or command prompt where you have access to the cluster.
3. If needed, SSH into the Atlas host.
4. If needed, make sure the JAVA_HOME variable is set; if it isn't, run the export command pointing to the location of the JDK.
5. Run the transformation command.

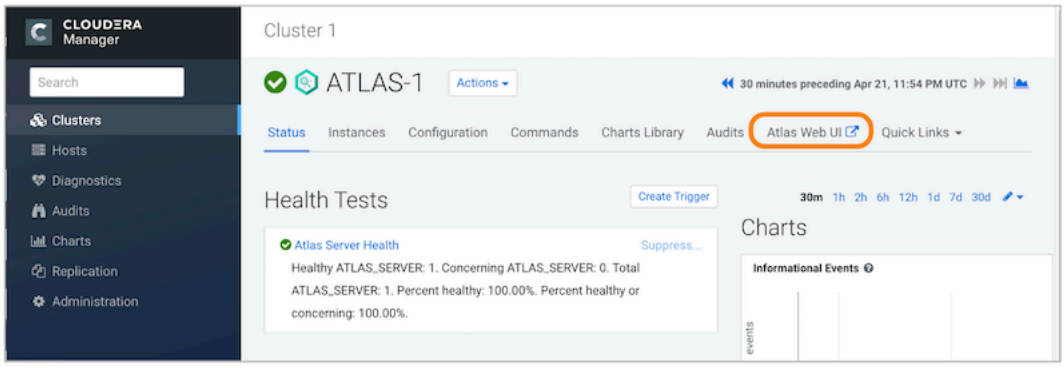
   When the transformation is complete, you'll see a status message in the command output.

Run the import

When Atlas starts up, it checks the output file location for a completed file from the transformation phase. When Atlas finds the completed file, it begins importing entities automatically. To see the progress of the import:

1. Open the Atlas UI.

   You can open Atlas from the Atlas service page in Cloudera Manager.



2. Review transition progress in the Statistics page.

   The normal Atlas UI does not appear in migration mode; instead you'll see the **Statistics** page, which shows a real-time report of the number of entities, classifications, and other metadata that have been created in Atlas.

Validate the transition

To give yourself confidence that the transition was successful, use the Statistics page in Atlas to compare to the metadata in Navigator. See the "How many Navigator entities are transitioned?" section in Assumptions and prerequisites for instructions on how to lookup counts in Navigator.

- Count of migrated entities. Does the total number of imported entities match what you expect from Navigator? Remember that not all Navigator entities are not migrated: HDFS entities are only migrated if they are referenced in Hive, Impala, or Spark operations that are included in the transition.
- Count of managed metadata that became business metadata in Atlas.
- Count of managed metadata assignments. Consider reproducing searches on commonly used business metadata to validate that you see the same results in each system.

Move Atlas out of migration mode

After installation, Atlas runs in migration mode:

- Atlas does not collect metadata from services running on the cluster. The metadata remains in Kafka topics and will be collected later.
- Atlas starts importing metadata when it finds a final transformation file in the location you specified in step "Confirm the output file location" from the step "Click Add Atlas Service."

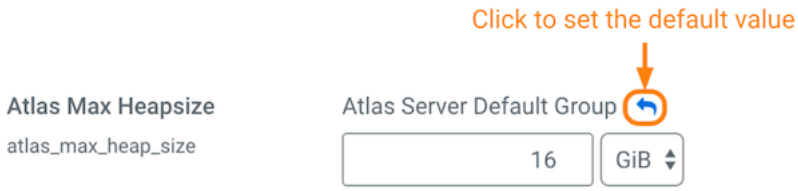
To move Atlas from migration mode into normal operation:

1. Open Cloudera Manager to the Atlas service.
2. Go to the Configuration tab.
3. Filter the list of properties by typing "Safety" in the filter field.
4. Remove the migration-specific entries from the Atlas Server Advanced Configuration Snippet (Safety Valve) for conf/atlas-application.properties.

   Remove the following properties:

```
atlas.migration.data.filename
atlas.migration.mode.batch.size
atlas.migration.mode.workers
```

5. Reset the Atlas Max Heapsize property back to the default value.



6. Click Save Changes.
7. Restart Atlas.

   Choose Action Restart .

Mapping Navigator metadata to Atlas

You must validate the Navigator to Apache Atlas transition by reviewing the list of metadata mapping and its types transitioned from the Navigator.

Use this topic as a checklist to help you validate the transition of metadata from Navigator to Atlas.

User-supplied metadata mapping

The following user-supplied metadata is transitioned to Atlas, including definitions and assignments to entities. Enumerations defined in Navigator are created as instances of the enumeration type in Atlas.

Navigator Construct	Atlas Construct
Tag	Label
User-defined Properties • Key • Value	User-defined properties • Key • Value
Managed Properties • Namespace • Namespace description • Property Name (original name) • Classes (entity types that property can be assigned to) • Display Name • Property description • Multivalued • Type (Text, Number, Boolean, Date, Enumeration)	Business Metadata Attributes • Business Metadata name • Business Metadata description • Attribute name • Applicable Types (entity types that property can be assigned to) • Display Name (not used in the UI) • Description (not used in the UI) • Enable Multivalues • Type (string, Boolean, byte, short, int, float, double, long, date, enumeration)

Policy mapping

Navigator policies provided the ability to apply metadata to Navigator entities. They do not have an equivalent function in Atlas and are not transitioned. You may be able to implement similar functionality through data profilers in the Cloudera Data Catalog. You can create a report of Navigator policies using the Navigator API, for example, use a GET call to the /policy API:

```
curl -X GET "<navigator host>:<port>/api/v14/policy" -u <username>:<pass
word>
```

Technical metadata mapping

The transition process maps all the needed technical metadata for entities from Navigator to be represented in Atlas. There are some instances where the transition has to convert data from one type to another or generate new content to conform to the Atlas data model. The details of the entity mapping are exhaustively described in the Atlas technical metadata transition reference.

Transitioning Navigator audits

Existing Cloudera Navigator audits are not transitioned to the Cloudera cluster. To transition reports running against Navigator data to Apache Ranger and other resources you must review the available options.

To manage Navigator audits in a Cloudera Runtime cluster, consider the following options:

Maintain legacy audits in Navigator

You can continue to run Navigator to access your existing audits (and/or metadata). If you choose to keep Navigator running, make sure that its users don't add content to the archived system rather than the new Atlas instance. Consider:

- Removing editing privileges from users. If Navigator is configured for LDAP or Active Directory authentication, you can modify users or groups to remove privileges for adding or editing metadata. For details, see Administering Navigator User Roles.
- Marking Navigator entities as stale. If you are managing more than one cluster in Navigator, you can mark entities from the upgraded cluster to indicate to users that the entities are no longer maintained in Navigator. One way to do this is to create a policy that assigns a tag to the entities from the upgraded cluster. For details, see Using Policies to Automate Metadata Tagging.

Archive your Navigator audits

When Cloudera Manager upgrades to 7.x, it maintains the database of Navigator audits. After the upgrade, you can access audits through Navigator as normal; new audits continue to be collected from CDH services.

When you upgrade a CDH cluster to Cloudera Runtime, the Navigator audits persist. However, services no longer produce audits for Navigator. You can continue to run Navigator to be able to access the audits; at some point—perhaps after your need for immediate access to the audits expires—you'll want to archive the audits. For details, see Maintaining Navigator Audit Server.

At that point, if Cloudera Manager is not managing another CDH cluster, you can shut down Navigator.

Transition audit reports and processes to Ranger

In Cloudera, Ranger performs auditing against the data access policies defined for each service. For example, if a Ranger policy allows only users from the Finance group to access a particular Hive database, Ranger audits will show when those users accessed the database successfully and when other users attempted to access the database and were denied. While the Ranger audits are a significant subset of the audits performed by Navigator, the format and content is different enough that Cloudera doesn't provide a transition path for Navigator audits into the same repository as Ranger audits.

When redirecting reports or processes to Ranger, you'll need to:

- Identify the audited information: does an equivalent exist in Ranger?
- Identify the method of accessing the audit data and map it to Ranger: Did the reporting application use the Navigator API? Did it access archived data or the Navigator audit database? Ranger provides an API to access audits; audit data is written to HDFS (under /ranger/audit/<component name>). 30 days of audit records are indexed in Solr. The audit events are stored in JSON format. For details, see Managing Auditing with Ranger.

What's new in Atlas for Navigator Users?

Reviewing the differences between Navigator and Apache Atlas helps you to know more about the features and functionalities available for metadata management in Apache Atlas.

Customers migrating from CDH with Navigator will be able to take advantage of these Atlas features:

Control data access in Ranger using Atlas metadata

Atlas classifications can control table, row, or column level access; Ranger can use Atlas classifications as the trigger to instruct services to mask data. See Configure Atlas Authorization using Ranger.

Reliability and scalability

Atlas uses Kafka to pass metadata and HBase and Solr to store it. These services leverage the distributed nature of the cluster to provide scalability and can be further annealed to be highly-available. See Apache Atlas architecture

Additional metadata sources

Atlas is flexible when it comes to adding metadata models for additional sources through REST APIs for metadata capture; it supports NiFi and Kafka metadata in addition to the sources integrated with Navigator. See Extending Atlas to manage metadata from additional sources.

Business Glossary

Atlas provides an interface to create and manage a glossary of business terms that can clarify and standardize how data is identified and used in an organization. See Atlas Glossaries overview

Data profiling

Cloudera Data Catalog includes automatic data tagging for a list of common types of data and allows you to tag additional types that can be identified using regular expressions.

When running Atlas with Cloudera Public Cloud , you'll also get the benefit of being able to see metadata across all the workloads in a single environment.

Preparing the backend HMS database for upgrade

Learn how you can prevent upgrade failures when you have materialized views or MSSQL indexed views created on top of Hive backend schemas, such as SYS or INFORMATION_SCHEMA tables.

About this task

If you have created any materialized views or MSSQL indexed views on Hive SYS or INFORMATION_SCHEMA tables, your upgrade process can fail when the upgrade SQL statements are trying to drop these tables.

You must drop the materialized views before performing an upgrade and then recreate the views after the upgrade process is complete.

Important: Cloudera recommends that you do not create any materialized views or indexed views on the SYS and INFORMATION_SCHEMA tables. However, if you have already created the views, then follow the steps provided in this topic to identify such views and drop them before upgrading the cluster.

Before you begin

You must back up the Hive metastore (HMS) backend database before dropping the materialized views.

Procedure

1. Start a Hive Beeline session and run the following query to identify materialized views that are created on top of the SYS and INFORMATION_SCHEMA tables:

```
SELECT DISTINCT d.DB_LOCATION_URI, d.NAME, t.TBL_NAME, t.TBL_TYPE, t.OWN
ER, t.VIEW_EXPANDED_TEXT
FROM sys.TBLS t
         INNER JOIN sys.DBS d ON t.DB_ID = d.DB_ID
         INNER JOIN sys.MV_CREATION_METADATA mv ON mv.TBL_NAME = t.TBL_N
AME
         INNER JOIN sys.MV_TABLES_USED tu ON mv.MV_CREATION_METADATA_ID =
 tu.MV_CREATION_METADATA_ID
WHERE tu.TBL_ID IN (SELECT distinct t.TBL_ID
                    FROM sys.MV_CREATION_METADATA mv
                             INNER JOIN sys.MV_TABLES_USED tu ON mv.MV_CR
EATION_METADATA_ID = tu.MV_CREATION_METADATA_ID
                             INNER JOIN sys.TBLS t ON tu.TBL_ID = t.TBL_ID
                             INNER JOIN sys.DBS d ON t.DB_ID = d.DB_ID
                    WHERE lower(d.NAME) IN ('sys', 'information_schema'))
  AND upper(t.TBL_TYPE) = 'MATERIALIZED_VIEW';
```

2. If the query returns any materialized views, drop each view using the DROP statement.
   DROP MATERIALIZED VIEW [db_name.]materialzed_view_name;
3. Upgrade the cluster and recreate the views after the upgrade process is complete.

Migrating Hive 1-2 to Hive 3

If you have a large Hive metastore implementation, preparing the metastore for the upgrade by finding missing tables, missing partitions, and problematic SERDE definitions can take a long time. Cloudera Community tools can save significant time. For more information, see Expediting the Hive Upgrade. Check with your Cloudera account team resources regarding professional services.

Warning: Custom JARs configured in the Hive auxiliary library (auxlib) path in CDH clusters can cause upgrade failures when migrating to CDP Private Cloud Base. To ensure a successful upgrade, remove any custom JARs from the Hive auxlib path before proceeding. You can recompile and use these JARs with the latest CDP Private Cloud Base libraries after the upgrade.

Hive Configuration Changes Requiring Consent

Before the CDH to CDP upgrade process starts, the pre-upgrade wizard asks you to consent to a number of critical configuration changes that occur after the upgrade. To prepare for this step, you can review the default before and after upgrade values of the properties.

Property Values Before and After Upgrade

Before starting the upgrade process, you are asked to consent to changes in the values of the following properties:

hive.conf.hidden.list

Before upgrade:

```
javax.jdo.option.ConnectionPassword,hive.server2.keystore.passwo
rd,hive.metastore.dbaccess.ssl.truststore.password,fs.s3.awsAcce
ssKeyId,fs.s3.awsSecretAccessKey,fs.s3n.awsAccessKeyId,fs.s3n.aw
sSecretAccessKey,fs.s3a.access.key,fs.s3a.secret.key,fs.s3a.prox
y.password,dfs.adls.oauth2.credential,fs.adl.oauth2.credential,f
s.azure.account.oauth2.client.secret
```

After upgrade:

```
javax.jdo.option.ConnectionPassword,hive.server2.keystore.passwo
rd,hive.druid.metadata.password,hive.driver.parallel.compilation
.global.limit
```

hive.conf.restricted.list

Before upgrade:

```
hive.security.authenticator.manager,hive.security.authorization.
manager,hive.users.in.admin.role,hive.server2.xsrf.filter.enable
d,hive.spark.client.connect.timeout,hive.spark.client.server.con
nect.timeout,hive.spark.client.channel.log.level,hive.spark.clie
nt.rpc.max.size,hive.spark.client.rpc.threads,hive.spark.client.
secret.bits,hive.spark.client.rpc.server.address,hive.spark.clie
nt.rpc.server.port,hive.spark.client.rpc.sasl.mechanisms,hadoop.
bin.path,yarn.bin.path,spark.home,bonecp.,hikaricp.,hive.driver.
parallel.compilation.global.limit,_hive.local.session.path,_hive
.hdfs.session.path,_hive.tmp_table_space,_hive.local.session.pat
h,_hive.hdfs.session.path,_hive.tmp_table_space
```

After upgrade:

```
hive.security.authenticator.manager,hive.security.authorization.
manager,hive.security.metastore.authorization.manager,hive.secur
ity.metastore.authenticator.manager,hive.users.in.admin.role,hiv
e.server2.xsrf.filter.enabled,hive.security.authorization.enable
d,hive.distcp.privileged.doAs,hive.server2.authentication.ldap.b
aseDN,hive.server2.authentication.ldap.url,hive.server2.authenti
cation.ldap.Domain,hive.server2.authentication.ldap.groupDNPatte
rn,hive.server2.authentication.ldap.groupFilter,hive.server2.aut
hentication.ldap.userDNPattern,hive.server2.authentication.ldap.
userFilter,hive.server2.authentication.ldap.groupMembershipKey,h
ive.server2.authentication.ldap.userMembershipKey,hive.server2.a
uthentication.ldap.groupClassKey,hive.server2.authentication.lda
p.customLDAPQuery,hive.privilege.synchronizer.interval,hive.spar
k.client.connect.timeout,hive.spark.client.server.connect.timeou
t,hive.spark.client.channel.log.level,hive.spark.client.rpc.max.
size,hive.spark.client.rpc.threads,hive.spark.client.secret.bits
,hive.spark.client.rpc.server.address,hive.spark.client.rpc.serv
er.port,hive.spark.client.rpc.sasl.mechanisms,bonecp.,hive.druid
.broker.address.default,hive.druid.coordinator.address.default,h
ikaricp.,hadoop.bin.path,yarn.bin.path,spark.home,hive.driver.pa
rallel.compilation.global.limit,_hive.local.session.path,_hive.h
dfs.session.path,_hive.tmp_table_space,_hive.local.session.path,
_hive.hdfs.session.path,_hive.tmp_table_space
```

hive.default.fileformat.managed

Before upgrade: None

After upgrade: ORC

hive.default.rcfile.serde

Before upgrade: org.apache.hadoop.hive.serde2.columnar.ColumnarSerDe

After upgrade: org.apache.hadoop.hive.serde2.columnar.LazyBinaryColumnarSerDe

hive.exec.dynamic.partition.mode

Before upgrade: strict

After upgrade: nonstrict

hive.exec.max.dynamic.partitions

Before upgrade: 1000

After upgrade: 5000

hive.exec.max.dynamic.partitions.pernode

Before upgrade: 100

After upgrade: 2000

hive.exec.post.hooks

Before upgrade:

```
com.cloudera.navigator.audit.hive.HiveExecHookContext,org.apache
.hadoop.hive.ql.hooks.LineageLogger
```

After upgrade: org.apache.hadoop.hive.ql.hooks.HiveProtoLoggingHook

hive.execution.engine

Before upgrade: mr

After upgrade: tez

hive.metastore.disallow.incompatible.col.type.changes

Before upgrade: FALSE

After upgrade: TRUE

hive.metastore.warehouse.dir

Before upgrade: /user/hive/warehouse

After upgrade: /warehouse/tablespace/managed/hive

hive.script.operator.env.blacklist

Before upgrade: hive.txn.valid.txns,hive.script.operator.env.blacklist

After upgrade:

```
hive.txn.valid.txns,hive.txn.tables.valid.writeids,hive.txn.vali
d.writeids,hive.script.operator.env.blacklist
```

hive.security.authorization.sqlstd.confwhitelist

Before upgrade:

```
hive\.auto\..*hive\.cbo\..*hive\.convert\..*hive\.exec\.dynamic\
.partition.*hive\.exec\..*\.dynamic\.partitions\..*hive\.exec\.c
ompress\..*hive\.exec\.infer\..*hive\.exec\.mode.local\..*hive\.
exec\.orc\..*hive\.exec\.parallel.*hive\.explain\..*hive\.fetch.
task\..*hive\.groupby\..*hive\.hbase\..*hive\.index\..*hive\.ind
ex\..*hive\.intermediate\..*hive\.join\..*hive\.limit\..*hive\.l
og\..*hive\.mapjoin\..*hive\.merge\..*hive\.optimize\..*hive\.or
c\..*hive\.outerjoin\..*hive\.parquet\..*hive\.ppd\..*hive\.prew
arm\..*hive\.server2\.proxy\.userhive\.skewjoin\..*hive\.smbjoin
\..*hive\.stats\..*hive\.strict\..*hive\.tez\..*hive\.vectorized
\..*mapred\.map\..*mapred\.reduce\..*mapred\.output\.compression
\.codecmapred\.job\.queuenamemapred\.output\.compression\.typema
pred\.min\.split\.sizemapreduce\.job\.reduce\.slowstart\.complet
edmapsmapreduce\.job\.queuenamemapreduce\.job\.tagsmapreduce\.in
put\.fileinputformat\.split\.minsizemapreduce\.map\..*mapreduce\
.reduce\..*mapreduce\.output\.fileoutputformat\.compress\.codecm
apreduce\.output\.fileoutputformat\.compress\.typeoozie\..*tez\.
am\..*tez\.task\..*tez\.runtime\..*tez\.queue\.namehive\.transpo
se\.aggr\.joinhive\.exec\.reducers\.bytes\.per\.reducerhive\.cli
```

```
ent\.stats\.countershive\.exec\.default\.partition\.namehive\.ex
ec\.drop\.ignorenonexistenthive\.counters\.group\.namehive\.defa
ult\.fileformat\.managedhive\.enforce\.bucketmapjoinhive\.enforc
e\.sortmergebucketmapjoinhive\.cache\.expr\.evaluationhive\.quer
y\.result\.fileformathive\.hashtable\.loadfactorhive\.hashtable\
.initialCapacityhive\.ignore\.mapjoin\.hinthive\.limit\.row\.max
\.sizehive\.mapred\.modehive\.map\.aggrhive\.compute\.query\.usi
ng\.statshive\.exec\.rowoffsethive\.variable\.substitutehive\.va
riable\.substitute\.depthhive\.autogen\.columnalias\.prefix\.inc
ludefuncnamehive\.autogen\.columnalias\.prefix\.labelhive\.exec\
.check\.crossproductshive\.cli\.tez\.session\.asynchive\.compath
ive\.exec\.concatenate\.check\.indexhive\.display\.partition\.co
ls\.separatelyhive\.error\.on\.empty\.partitionhive\.execution\.
enginehive\.exec\.copyfile\.maxsizehive\.exim\.uri\.scheme\.whit
elisthive\.file\.max\.footerhive\.insert\.into\.multilevel\.dirs
hive\.localize\.resource\.num\.wait\.attemptshive\.multi\.insert
\.move\.tasks\.share\.dependencieshive\.support\.quoted\.identif
iershive\.resultset\.use\.unique\.column\.nameshive\.analyze\.st
mt\.collect\.partlevel\.statshive\.exec\.schema\.evolutionhive\.
server2\.logging\.operation\.levelhive\.server2\.thrift\.results
et\.serialize\.in\.taskshive\.support\.special\.characters\.tabl
enamehive\.exec\.job\.debug\.capture\.stacktraceshive\.exec\.job
\.debug\.timeouthive\.llap\.io\.enabledhive\.llap\.io\.use\.file
id\.pathhive\.llap\.daemon\.service\.hostshive\.llap\.execution\
.modehive\.llap\.auto\.allow\.uberhive\.llap\.auto\.enforce\.tre
ehive\.llap\.auto\.enforce\.vectorizedhive\.llap\.auto\.enforce\
.statshive\.llap\.auto\.max\.input\.sizehive\.llap\.auto\.max\.o
utput\.sizehive\.llap\.skip\.compile\.udf\.checkhive\.llap\.clie
nt\.consistent\.splitshive\.llap\.enable\.grace\.join\.in\.llaph
ive\.llap\.allow\.permanent\.fnshive\.exec\.max\.created\.filesh
ive\.exec\.reducers\.maxhive\.reorder\.nway\.joinshive\.output\.
file\.extensionhive\.exec\.show\.job\.failure\.debug\.infohive\.
exec\.tasklog\.debug\.timeouthive\.query\.id
```

After upgrade:

```
hive\.auto\..*hive\.cbo\..*hive\.convert\..*hive\.druid\..*hive\
.exec\.dynamic\.partition.*hive\.exec\.max\.dynamic\.partitions.
*hive\.exec\.compress\..*hive\.exec\.infer\..*hive\.exec\.mode.l
ocal\..*hive\.exec\.orc\..*hive\.exec\.parallel.*hive\.exec\.que
ry\.redactor\..*hive\.explain\..*hive\.fetch.task\..*hive\.group
by\..*hive\.hbase\..*hive\.index\..*hive\.index\..*hive\.interme
diate\..*hive\.jdbc\..*hive\.join\..*hive\.limit\..*hive\.log\..
*hive\.mapjoin\..*hive\.merge\..*hive\.optimize\..*hive\.materia
lizedview\..*hive\.orc\..*hive\.outerjoin\..*hive\.parquet\..*hi
ve\.ppd\..*hive\.prewarm\..*hive\.query\.redaction\..*hive\.serv
er2\.thrift\.resultset\.default\.fetch\.sizehive\.server2\.proxy
\.userhive\.skewjoin\..*hive\.smbjoin\..*hive\.stats\..*hive\.st
rict\..*hive\.tez\..*hive\.vectorized\..*hive\.query\.reexecutio
n\..*reexec\.overlay\..*fs\.defaultFSssl\.client\.truststore\.lo
cationdistcp\.atomicdistcp\.ignore\.failuresdistcp\.preserve\.st
atusdistcp\.preserve\.rawxattrsdistcp\.sync\.foldersdistcp\.dele
te\.missing\.sourcedistcp\.keystore\.resourcedistcp\.liststatus\
.threadsdistcp\.max\.mapsdistcp\.copy\.strategydistcp\.skip\.crc
distcp\.copy\.overwritedistcp\.copy\.appenddistcp\.map\.bandwidt
h\.mbdistcp\.dynamic\..*distcp\.meta\.folderdistcp\.copy\.listin
g\.classdistcp\.filters\.classdistcp\.options\.skipcrccheckdistc
p\.options\.mdistcp\.options\.numListstatusThreadsdistcp\.option
s\.mapredSslConfdistcp\.options\.bandwidthdistcp\.options\.overw
ritedistcp\.options\.strategydistcp\.options\.idistcp\.options\.
p.*distcp\.options\.updatedistcp\.options\.deletemapred\.map\..*
mapred\.reduce\..*mapred\.output\.compression\.codecmapred\.job\
```

```
.queue\.namemapred\.output\.compression\.typemapred\.min\.split\
.sizemapreduce\.job\.reduce\.slowstart\.completedmapsmapreduce\.
job\.queuenamemapreduce\.job\.tagsmapreduce\.input\.fileinputfor
mat\.split\.minsizemapreduce\.map\..*mapreduce\.reduce\..*mapred
uce\.output\.fileoutputformat\.compress\.codecmapreduce\.output\
.fileoutputformat\.compress\.typeoozie\..*tez\.am\..*tez\.task\.
.*tez\.runtime\..*tez\.queue\.namehive\.transpose\.aggr\.joinhiv
e\.exec\.reducers\.bytes\.per\.reducerhive\.client\.stats\.count
ershive\.exec\.default\.partition\.namehive\.exec\.drop\.ignoren
onexistenthive\.counters\.group\.namehive\.default\.fileformat\.
managedhive\.enforce\.bucketmapjoinhive\.enforce\.sortmergebucke
tmapjoinhive\.cache\.expr\.evaluationhive\.query\.result\.filefo
rmathive\.hashtable\.loadfactorhive\.hashtable\.initialCapacityh
ive\.ignore\.mapjoin\.hinthive\.limit\.row\.max\.sizehive\.mapre
d\.modehive\.map\.aggrhive\.compute\.query\.using\.statshive\.ex
ec\.rowoffsethive\.variable\.substitutehive\.variable\.substitut
e\.depthhive\.autogen\.columnalias\.prefix\.includefuncnamehive\
.autogen\.columnalias\.prefix\.labelhive\.exec\.check\.crossprod
uctshive\.cli\.tez\.session\.asynchive\.compathive\.display\.par
tition\.cols\.separatelyhive\.error\.on\.empty\.partitionhive\.e
xecution\.enginehive\.exec\.copyfile\.maxsizehive\.exim\.uri\.sc
heme\.whitelisthive\.file\.max\.footerhive\.insert\.into\.multil
evel\.dirshive\.localize\.resource\.num\.wait\.attemptshive\.mul
ti\.insert\.move\.tasks\.share\.dependencieshive\.query\.results
\.cache\.enabledhive\.query\.results\.cache\.wait\.for\.pending\
.resultshive\.support\.quoted\.identifiershive\.resultset\.use\.
unique\.column\.nameshive\.analyze\.stmt\.collect\.partlevel\.st
atshive\.exec\.schema\.evolutionhive\.server2\.logging\.operatio
n\.levelhive\.server2\.thrift\.resultset\.serialize\.in\.taskshi
ve\.support\.special\.characters\.tablenamehive\.exec\.job\.debu
g\.capture\.stacktraceshive\.exec\.job\.debug\.timeouthive\.llap
\.io\.enabledhive\.llap\.io\.use\.fileid\.pathhive\.llap\.daemon
\.service\.hostshive\.llap\.execution\.modehive\.llap\.auto\.all
ow\.uberhive\.llap\.auto\.enforce\.treehive\.llap\.auto\.enforce
\.vectorizedhive\.llap\.auto\.enforce\.statshive\.llap\.auto\.ma
x\.input\.sizehive\.llap\.auto\.max\.output\.sizehive\.llap\.ski
p\.compile\.udf\.checkhive\.llap\.client\.consistent\.splitshive
\.llap\.enable\.grace\.join\.in\.llaphive\.llap\.allow\.permanen
t\.fnshive\.exec\.max\.created\.fileshive\.exec\.reducers\.maxhi
ve\.reorder\.nway\.joinshive\.output\.file\.extensionhive\.exec\
.show\.job\.failure\.debug\.infohive\.exec\.tasklog\.debug\.time
outhive\.query\.idhive\.query\.tag
```

hive.security.command.whitelist

Before upgrade: set,reset,dfs,add,list,delete,reload,compile

After upgrade: set,reset,dfs,add,list,delete,reload,compile,llap

hive.server2.enable.doAs

Before upgrade: TRUE (in case of unsecure cluster only)

After upgrade: FALSE (in all cases)

hive.server2.parallel.ops.in.session

Before upgrade: TRUE

After upgrade: FALSE

hive.support.special.characters.tablename

Before upgrade: FALSE

After upgrade: TRUE

Remove transactional=false from Table Properties

In CDH 5.x it is possible to create tables with having the property transactional=false set. While this is a no-op setting, if any of your Hive tables explicitly set this, the upgrade process fails.

About this task

You must remove 'transactional'='false' from any tables you want to upgrade from CDH 5.x to CDP.

Procedure

Alter the table as follows:

```
ALTER TABLE my_table UNSET TBLPROPERTIES ('transactional');
```

Check SERDE Definitions and Availability

Ensure correct Serde definitions and a reference to a SERDE exists to ensure a successful upgrade.

About this task

You perform this step if you do not modify the HSMM process for expediting the Hive upgrade.

Procedure

1. Check Serde definitions for correctness and check for SERDE availability.
2. Correct any problems found as follows:
   - Remove the table having the problematic SERDE.
   - Ensure the SERDE is available during the upgrade, so the table can be evaluated.

Checking Apache HBase

To upgrade to Cloudera Base on premises from CDH and have the HBase service installed, there are several pre-upgrades steps you are required to complete.

Note: Before upgrading the dependent services such as HBase, you must verify and ensure that the HDFS safemode is off.

Note: You must complete these steps when you run the Upgrade Wizard, after the Cloudera Runtime parcel has been distributed, but before finishing the Upgrade Wizard.

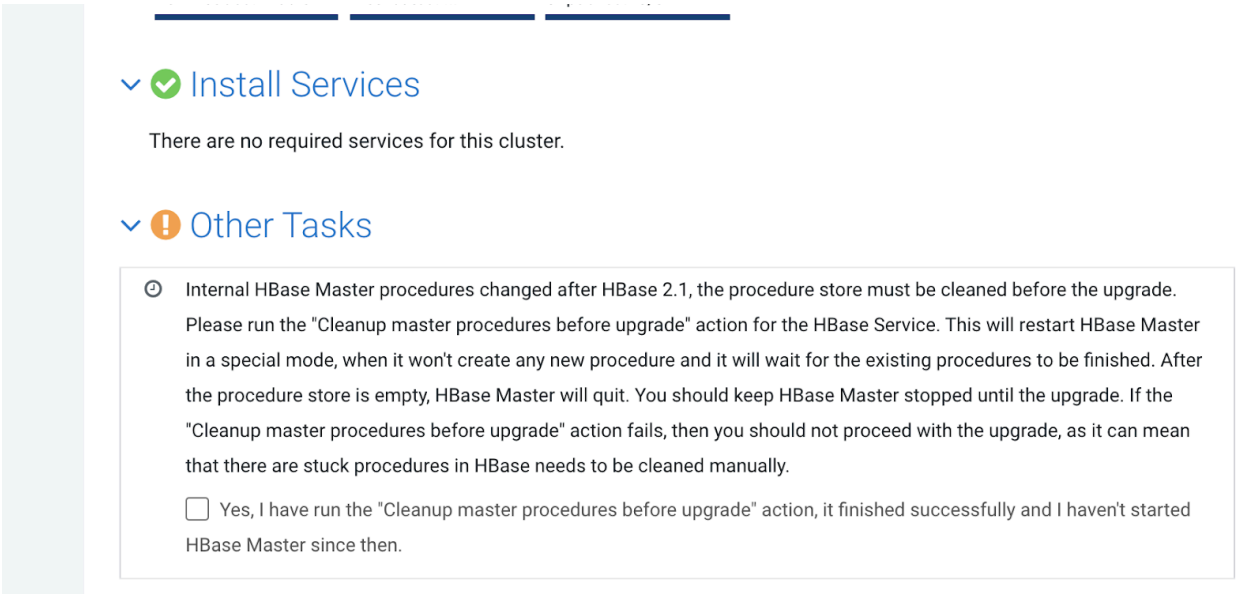
Important:

Ensure that you complete all the pre-upgrade steps if you have Apache HBase installed in your existing CDH cluster.

When you are attempting to upgrade from a CDH cluster to a Cloudera Base on premises cluster, checkboxes appear to ensure you have performed all the necessary HBase related pre-upgrade transition step:

- Check co-processor classes
- Clean the HBase Master procedure store

The upgrade continues only if you check the following statements: Yes, I have run the "Cleanup master procedure before upgrade" action, it finished sucessfully and I haven't started HBase Master since then.



Check co-processor classes

External co-processors are not automatically upgraded, you must upgrade them manually. Before upgrading, ensure that your co-processors are compatible with the upgrade.

About this task

Important:

Ensure that you complete all the pre-upgrade steps if you have Apache HBase installed in your existing CDH cluster.

There are two ways to handle co-processor upgrade:

- Upgrade your co-processor jars manually before continuing the upgrade.
- Temporarily unset the co-processors and continue the upgrade.

  Once they are manually upgraded, they can be reset.

Attempting to upgrade without upgrading the co-processor jars can result in unpredictable behaviour such as HBase role start failure, HBase role crashing, or even data corruption.

If your cluster is Kerberized, ensure that you run the kinit command as a hbase user before running the pre-upgrade commands.

Procedure

1. Download and distribute parcels for target version of Cloudera Base on premises.

   Important: Do not activate the parcel yet.

   If the downloaded parcel version is higher than the current Cloudera Manager version, the following error message displayed:

   Error for parcel CDH-7.X.parcel : Parcel version 7.X is not supported by this version of Cloudera Manager. Upgrade Cloudera Manager to at least 7.X before using this version of the parcel.

   You can safely ignore this error message.

2. Run the hbase pre-upgrade validate-cp commands to check if your co-processors are compatible with the upgrade.

   Use the CDP parcel to run the pre-upgrade commands. Cloudera recommends that you run them on an HMaster host.

   For example, you can check for co-processor compatibility on master:

   ```
   $ /opt/cloudera/parcels/CDH-7.1.1-1.cdh7.1.1.p0.3224867/bin/hbase pre-up
   grade validate-cp -jar /opt/cloudera/parcels/CDH-7.1.1-1.cdh7.1.1.p0.322
   4867/jars/ -config
   ```

   Note: Modify the jar file path /opt/cloudera/parcels/CDH-7.1.1-1.cdh7.1.1.p0.3224867/jars according to the co-processors' location.

   Or, you can validate every table level co-processors where the table name matches to the .* regular expression:

   ```
   $ /opt/cloudera/parcels/CDH-7.1.1-1.cdh7.1.1.p0.3224867/bin/hbase pre-up
   grade validate-cp -table '.*'
   ```

   Optinally, you can run the following command for a more detailed output:

   ```
   HBASE_ROOT_LOGGER=DEBUG,console hbase pre-upgrade validate-cp -table '.*'
   ```

   This way you can verify that all of the required tables were checked. The detailed output should contain lines like the following where test_table is a table on the server:

   ```
   21/05/10 11:07:58 DEBUG coprocessor.CoprocessorValidator: Validating table
    test_table
   ```

3. Check the output to determine if your co-processors are compatible with the upgrade.

   The output looks similar to the following:

   ```
   $ hbase pre-upgrade validate-cp -config
   ... some output ...
   $ echo $?
   0
   ```

   If echo $? prints 0, the check was successful and your co-processors are compatible. A non-zero value means unsuccessful, your co-processors are not compatible.

What to do next

Clean the HBase Master procedure store.

Clean the HBase Master procedure store

HBase Master procedures changed after HBase 2.1, therefore the procedure store must be cleaned before upgrading from CDH 6 to CDP.

About this task

From HBase 2.2 it is changed how HBase Master performs internal, housekeeping operations such as table creation or region removal. As a result if a procedure is started in HBase 2.1 (CDH 6 version), it cannot be continued after upgrading to HBase 2.2 (CDP version).

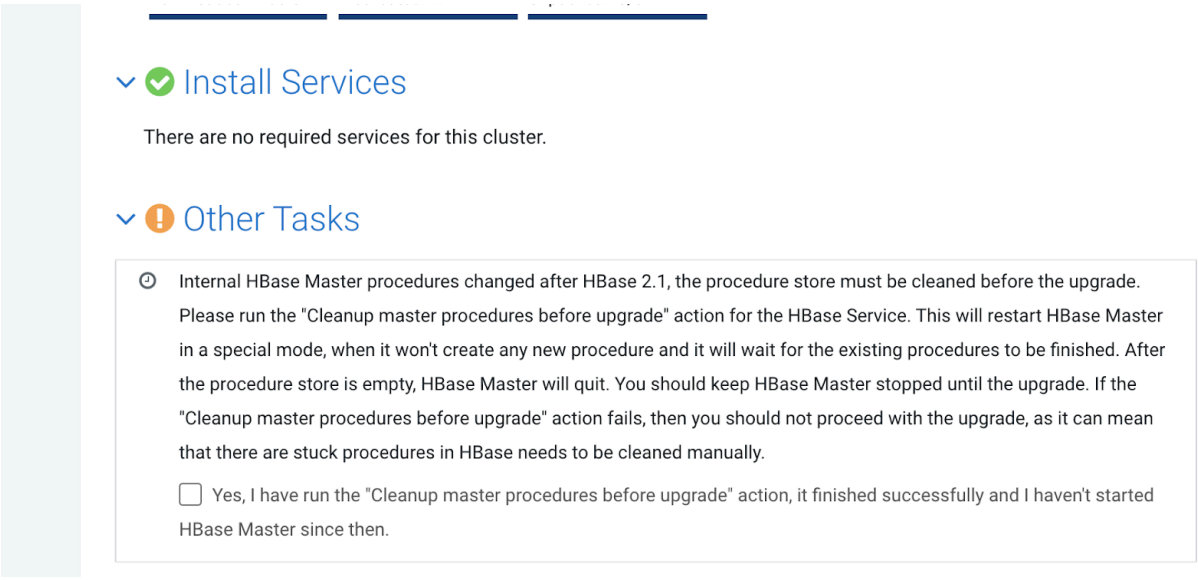
To prevent such cases you must clean the HMase Master procedure store before starting the upgrade and you can find this functionality from Cloudera Manager 7.7.1 and above.

Procedure

1. In Cloudera Manager, select the HBase service.
2. Click Actions Cleanup Master Procedures Before Upgrade
   HBase Master rules are stopped and restarted in a so-called upgrade mode. In this mode HBase Masters are waiting for all ongoing procedures to finish. Once an HBase Master procedure store is empty, its HBase Master quits automatically.
3. If the cleanup command fails, check if there are any stuck procedures that need to be cleaned manually before upgrade.

   If the command fails, the following error message is displayed: Failed to prepare HBase for the upgrade. There might be some HBase Master procedures which haven't been finished in time. Please make sure you clean these procedures before you would continue the upgrade.

   For more information, see *Using the HBCK2 tool to remediate HBase clusters*.
4. Start the upgrade.
5. Find the Other Tasks section of the Upgrade Wizard.
6. Check the Yes, I have run the "Cleanup master procedure before upgrade" action, it finished successfully and I haven't started HBase Master since then.



What to do next

Continue the upgrade using the Cloudera Manager upgrade wizard.

Related Information

Using the HBCK2 tool to remediate HBase clusters

CDH cluster upgrade requirements for Replication Manager

Reviewing the CDH cluster upgrade guidelines and requirements for Replication Manager help you to upgrade successfully. Before you start the upgrade, check the version numbers to ensure that the clusters are in sync.

- Ensure that the supported source and target clusters and the corresponding Cloudera Manager versions are in sync with respect to the cluster configurations prior to starting the upgrade.
- Upgrade your target cluster to Cloudera Base on premises first. Upgrading the target CDH cluster first ensures that your data on the source cluster is not corrupted or rendered invalid.
- Upgrade the source cluster to Cloudera Base on premises after the data is transitioned to the Cloudera Base on premises cluster (target). And, later verify that both source and target clusters are upgraded to Cloudera Base on premises clusters.

Upgrading the JDK

Cloudera Manager, Cloudera Runtime, and CDH require a supported version of the Java Development Kit (JDK) to be installed on all hosts. For details, see CDP Java Requirements.

Loading Filters ...

Warning:

- If you are upgrading from a lower major version of the JDK to JDK 1.8 or from JDK 1.6 to JDK 1.7, and you are using AES-256 bit encryption, you must install new encryption policy files. (In a Cloudera Manager deployment, you automatically install the policy files; for unmanaged deployments, install them manually.) See Using AES-256 Encryption on page 52. This step is not required when using JDK 1.8.0_162 or greater. JDK 1.8.0_162 enables unlimited strength encryption by default.

  You must also ensure that the Java Truststores are retained during the upgrade. (

  Cloudera recommends the following for keystores and truststores for Cloudera Manager clusters:

  - Create a separate keystore for each host. Each keystore should have a name that helps identify it as to the type of host—server or agent, for example. The keystore contains the private key and should be password protected.
  - Create a single truststore that can be used by the entire cluster. This truststore contains the root CA and intermediate CAs used to authenticate certificates presented during TLS/SSL handshake. The truststore does not need to be password protected. (See Understanding Keystores and Truststores to Truststore for more information about the truststore for TLS/SSL and Cloudera clusters.)

There are several procedures you can use to upgrade the JDK:

- Installing Java During an Upgrade

  If you are upgrading to Cloudera Manager 6.0.0 or higher, you can manually install JDK 1.8 on the Cloudera Manager server host, and then, as part of the Cloudera Manager upgrade process, you can specify that Cloudera Manager upgrade the JDK on the remaining hosts.

  Note: Cloudera Manager only installs Oracle JDK. You can upgrade to OpenJDK using these steps.

- Manually Installing Oracle JDK 1.8 on page 44

  You can manually install JDK 1.8 on all managed hosts. If you are upgrading to any version of Cloudera Manager 5.x, you must use this procedure. Continue with the steps in the next section.
- Manually Migrating to OpenJDK on page 48

Manually Installing Oracle JDK 1.8

Important: Manual upgrade of Oracle JDK 1.8 requires down time to stop and restart your cluster.

You can manually install Oracle JDK 1.8 on all managed hosts. If you are upgrading to any version of Cloudera Manager 5.x, you must use the following procedure:

1. Download the .tar.gz file for one of the 64-bit versions of Oracle JDK 1.8 from Java SE 8 Downloads. (This link is correct at the time of writing, but can change.)

2. Perform the following steps on all hosts that you are upgrading:
   a. Log in to the host as root using ssh.
   b. Copy the downloaded .tar.gz file to the host.
   c. Extract the JDK to the folder /usr/java/*JDK-VERSION*. For example:

      ```
      tar xvfz /PATH/TO/jdk-8u<UPDATE_VERSION>-linux-x64.tar.gz -C /usr/java/
      ```

3. If you have configured TLS for Cloudera Manager (see Encrypting Data in Transit), copy the jssecacerts file from the previous JDK installation to the new JDK installation. This step is not required when using JDK 1.8.0_162 or greater. JDK 1.8.0_162 enables unlimited strength encryption by default.

   For example:

   ```
   cp PREVIOUS_JAVA_HOME/jre/lib/security/jssecacerts NEW_JAVA_HOME/jre/lib/
   security
   ```

   (Substitute *PREVIOUS_JAVA_HOME* and *NEW_JAVA_HOME* with the paths to the JDK installations.)
4. Configure the location of the JDK on cluster hosts.
   a. Open the Cloudera Manager Admin Console.
   b. In the main navigation bar, click the Hosts tab. If you are configuring the JDK location on a specific host only, click the link for that host.
   c. Click the Configuration tab.
   d. Select Category Advanced .
   e. Set the Java Home Directory property to the custom location.
   f. Click Save Changes.
5. On the Cloudera Manager Server host only (not required for other hosts):
   a. Open the file /etc/default/cloudera-scm-server in a text editor.
   b. Edit the line that begins with export JAVA_HOME (if this line does not exist, add it) and change the path to the path of the new JDK (you can find the path under /usr/java).

      For example: (RHEL and SLES)

      ```
      export JAVA_HOME="/usr/java/jdk1.8.0_141-cloudera"
      ```

      For example: (Ubuntu)

      ```
      export JAVA_HOME="/usr/lib/jvm/java-8-oracle-cloudera"
      ```

   c. Save the file.
   d. Restart the Server.

      ```
      sudo systemctl restart cloudera-scm-server
      ```

6. Restart the .
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Restart .
7. Restart all clusters:
   a. On the Home Status tab, click the Options menu to the right of the cluster name and select Restart.
   b. Click Restart that appears in the next screen to confirm. If you have enabled high availability for HDFS, you can choose Rolling Restart instead to minimize cluster downtime. The Command Details window shows the progress of stopping services.

      When All services successfully started appears, the task is complete and you can close the Command Details window.

8. Delete the files from your previous Java installation. If you do not delete these files, Cloudera Manager and other components may continue to use the old version of the JDK.

OpenJDK

You must install a supported version of OpenJDK. If your deployment uses a version of OpenJDK lower than 1.8.0_232, see TLS Protocol Error with OpenJDK.

Important: For OpenJDK 8u241 and higher versions running on Kerberized clusters, you must disable referrals:

- Cloudera Manager 7.1.1 or higher:
  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.
- Cloudera Manager 7.0.3 or lower:
  1. Edit the Java Security file on all hosts by adding or changing the following configuration :
  2. ```
     sun.security.krb5.disableReferrals=true
     ```

     If the configuration already exists and is set to false, change it to true.
  3. Restart the cluster.

For more information, see the KB article.

Manually Installing OpenJDK

Before installing or upgrading Cloudera Manager and CDH/Cloudera Runtime, perform the steps in this section to install OpenJDK on all hosts in your cluster(s).

When you install Cloudera Enterprise, Cloudera Manager includes an option to install Oracle JDK. De-select this option.

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.
     ```

1. Log in to each host and run the following command:

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*

```
zypper install java-11-openjdk-devel
```

OpenJDK 17*

```
sudo zypper --non-interactive install java-17-openjdk-devel
```

* Azul OpenJDK, OpenJDK 8, OpenJDK 11, and OpenJDK 17 are TCK certified for Cloudera.

2. If you are using the SLES operating system, Cloudera Manager needs an additional configuration so that the JDK can be located:
   a. Log in to the Cloudera Manager server host.
   b. Open the following file in a text editor:

      ```
      /etc/default/cloudera-scm-server
      ```

   c. Add the following line:

      ```
      export JAVA_HOME=PATH TO THE JAVA INSTALLATION DIRECTORY
      ```

      For example:

      ```
      export JAVA_HOME=/usr/lib64/jvm/java-1.8.0-openjdk-1.8.0
      ```

   d. Save the file.
   e. Restart the Server.

      ```
      sudo systemctl restart cloudera-scm-server
      ```

Important: If you are upgrading to OpenJDK 11 or higher, set allow_weak_crypto to 'true' in /etc/krb5.conf ,to get Kerberos setup working with JDK11(or higher) and keep this set during the Cloudera Manager upgrade.

Important: If you are upgrading to OpenJDK 1.8.0_392 or higher, or Oracle JDK 1.8u351 or higher, set allo w_weak_crypto to 'true' in /etc/krb5.conf ,to get Kerberos setup working with OpenJDK 1.8.0_392 or higher, or Oracle JDK 1.8u351 or higher and keep this set during the Cloudera Manager upgrade.

Manually Migrating to OpenJDK

If you have Oracle JDK 1.7, Oracle JDK 1.8, or OpenJDK 8* installed on the hosts managed by Cloudera Manager, use the steps in this section to transition your deployment to use OpenJDK. The steps below require you to restart all clusters, which will cause downtime as the hosts restart. If your clusters have enabled high availability for HDFS, you can use a Rolling Restart to restart the clusters without downtime. Note that until the rolling restart completes, some of the hosts in your cluster will still be using the Oracle JDK. If you do not want a temporarily mixed environment, you can stop the cluster before performing the steps in this section to transition the JDK.

OpenJDK 11* is supported as of Cloudera Manager and CDH 6.3. Note the following:

- You must upgrade to Cloudera Manager 6.3 or higher, **before** upgrading to OpenJDK 11*.
- The package names used when installing the OpenJDK 11* 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
- See the following blog post for general information about migrating to Java 11: All You Need to Know For Migrating To Java 11.

OpenJDK 17* is supported as of Cloudera Manager 7.11.3 and CDP 7.1.9. Note the following:

- You must upgrade to Cloudera Manager 7.11.3 or higher, **before** upgrading to OpenJDK 17*.
- The package names used when installing the OpenJDK 17* are different and are noted in the steps below.
- The path to default truststore for OpenJDK 17* is lib/security/cacerts.
- See the following blog post for general information about migrating to Java 17: Migrate to Java 17.

1. Find out the package name of your currently installed JDK by running the following commands. The grep commands attempt to locate the installed JDK. If the JDK package is not returned, try looking for the string **jdk**.
   RHEL

   Oracle JDK 8

   ```
   yum list installed |grep oracle
   ```

   OpenJDK 8*, OpenJDK 11*, or OpenJDK 17*

   ```
   yum list installed |grep openjdk
   ```

   Ubuntu

   Oracle JDK 8

   ```
   apt list --installed | grep oracle
   ```

   OpenJDK 8*, OpenJDK 11*, or OpenJDK 17*

   ```
   apt list --installed | grep openjdk
   ```

   SLES

   Oracle JDK 8

   ```
   zypper search --installed-only |grep oracle
   ```

   OpenJDK 8*, OpenJDK 11*, or OpenJDK 17*

   ```
   zypper search --installed-only |grep openjdk
   ```

   The command will return values similar to the following example::

   ```
   oracle-j2sdk1.7.x86_64              1.7.0+update67-1           java-1.8.0-
   openjdk-devel
   ```

   The Oracle JDK package name in the above example is: oracle-j2sdk1.7.x86_64. The OpenJDK package is java -1.8.0-openjdk-devel.

2. Log in to each host managed by Cloudera Manager (including the Cloudera Manager server host) and run the following command to install OpenJDK:

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*

```
zypper install java-11-openjdk-devel
```

OpenJDK 17*

```
sudo zypper --non-interactive install java-17-openjdk-devel
```

3. (This step is required for Oracle JDK 8 or OpenJDK 8* only) On the Cloudera Manager Server host only (not required for other hosts):
   a. Open the file /etc/default/cloudera-scm-server in a text editor.
   b. Edit the line that begins with export JAVA_HOME (if this line does not exist, add it) and change the path to the path of the new JDK (the JDK is usually installed in /usr/lib/jvm)(or /usr/lib64/jvm on SLES 12), but the path may differ depending on how the JDK was installed).

      For example:

      RHEL 7, 8

      ```
      export JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk"
      ```

      Ubuntu

      ```
      export JAVA_HOME="/usr/lib/jvm/openjdk-8-jdk"
      ```

      SLES

      ```
      export JAVA_HOME="/usr/lib64/jvm/java-1.8.0-openjdk"
      ```

   c. Save the file.
   d. Restart the Server.

      ```
      sudo systemctl restart cloudera-scm-server
      ```

4. 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 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 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 54.

   Important: For OpenJDK 17*, Cloudera Manager and most services run with default GC without any custom tuning for any service.

5. Restart the .
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Restart .
6. Restart all clusters:
   a. On the Home Status tab, click to the right of the cluster name and select either Restart or Rolling Restart. Selecting Rolling Restart minimizes cluster downtime and is available only If you have enabled Auto TLS.
   b. Click Restart or Rolling Restart that appears in the next screen to confirm. The Command Details window shows the progress of stopping services.

      When All services successfully started appears, the task is complete and you can close the Command Details window.

7. Remove the JDK:
   a. Perform the following steps on all hosts managed by Cloudera Manager:
      1. Run the following command to remove the JDK, using the package names from Step 1: (If you do not delete these files, Cloudera Manager and other components may continue to use the old version of the JDK.)
         RHEL

         ```
         yum remove <JDK PACKAGE NAME>
         ```

         Ubuntu

         ```
         apt-get remove <JDK PACKAGE NAME>
         ```

         SLES

         ```
         zypper rm <JDK PACKAGE NAME>
         ```

      2. Confirm that the package has been removed:
         RHEL

         ```
         yum list installed |grep -i java
         ```

         Ubuntu

         ```
         apt list --installed | grep -i java
         ```

         SLES

         ```
         zypper search --installed-only |grep -i java
         ```

Using AES-256 Encryption

Note: This step is not required when using JDK 1.8.0_162 or greater. JDK 1.8.0_162 enables unlimited strength encryption by default.

If you are using CentOS/Red Hat Enterprise Linux 5.6 or higher, or Ubuntu, which use AES-256 encryption by default for tickets, you must install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy File on all cluster and Hadoop user machines. For JCE Policy File installation instructions, see the README.txt file included in the jce_policy-x.zip file. For more information, see Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy File

Alternately, you can configure Kerberos to not use AES-256 by removing aes256-cts:normal from the supported_en ctypes field of the kdc.conf or krb5.conf file. After changing the kdc.conf file, you must restart both the KDC and the kadmin server for those changes to take affect. You may also need to re-create or change the password of the relevant principals, including, potentially the Ticket Granting Ticket principal (krbtgt/REALM@REALM). If AES-256 is still used after completing steps, the aes256-cts:normal setting existed when the Kerberos database was created. To fix this, create a new Kerberos database and then restart both the KDC and the kadmin server.

To verify the type of encryption used in your cluster:

1. On the local KDC host, type this command to create a test principal:

   ```
   kadmin -q "addprinc test"
   ```

2. On a cluster host, type this command to start a Kerberos session as the test principal:

```
kinit test
```

3. On a cluster host, type this command to view the encryption type in use:

```
klist -e
```

If AES is being used, output like the following is displayed after you type the klist command; note that AES-256 is included in the output:

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: test@SCM
Valid starting     Expires            Service principal
05/19/11 13:25:04  05/20/11 13:25:04  krbtgt/SCM@SCM
    Etype (skey, tkt): AES-256 CTS mode with 96-bit SHA-1 HMAC, AES-256 CT
S mode with 96-bit SHA-1 HMAC
```

Configuring a Custom Java Home Location

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, you must follow these instructions to configure Cloudera Manager with your chosen location. The following procedure only changes the JDK location for Cloudera Management Services and cluster processes that are launched by the Cloudera Manager agents.

Important:

The procedure described on this page does not affect the JDK used by other non-Cloudera processes, including Hadoop processes such as the hdfs command.

Although not recommended, the Java Development Kit (JDK), which Cloudera services require, may be installed at a custom location if necessary. These steps assume you have already installed the JDK during product installation or as part of an upgrade.

To modify the Cloudera Manager configuration to ensure the JDK can be found:

1. Log into the Cloudera Manager server host.
2. Open the following file in a text editor:

```
/etc/default/cloudera-scm-server
```

3. Add the following line:

```
export JAVA_HOME=PATH TO THE JAVA INSTALLATION DIRECTORY
```

For example:

```
export JAVA_HOME=/usr/lib64/jvm/java-1.8.0-openjdk-1.8.0
```

4. Save the file.
5. Restart the Server.

```
sudo systemctl restart cloudera-scm-server
```

6. Open the Cloudera Manager Admin Console.
7. In the main navigation bar, click the Hosts tab. If you are configuring the JDK location on a specific host only, click the link for that host.
8. Click the Configuration tab.

9. Select CategoryAdvanced.
10. Set the Java Home Directory property to the custom location.
11. Click Save Changes.
12. Restart all services.

Tuning JVM Garbage Collection

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 overcommitted memory usage. You should monitor memory usage to determine whether memory is overcommitted.

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 roles 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
```

* Azul OpenJDK, OpenJDK 8, OpenJDK 11, and OpenJDK 17 are TCK certified for Cloudera.

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.

Important: No additional GC tuning has been applied to any service if they are running with OpenJDK 17*.

Important: Cloudera Manager Server with JDK 8 does not support G1GC.

Table 2: 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={{`

Upgrading the Operating System

This topic describes the additional steps needed to upgrade the operating system of a host managed by Cloudera Manager to a higher version, including major and minor releases.

Upgrading the operating system to a higher version but within the same major release is called a minor release upgrade. For example, upgrading from Redhat 6.8 to 6.9. This is a relatively simple procedure that involves properly shutting down all the components, performing the operating system upgrade, and then restarting everything in reverse order.

Upgrading the operating system to a different major release is called a major release upgrade. For example, upgrading from Redhat 6.8 to 7.4. This is a much more complex procedure to do it in-place, and some operating systems do not support these upgrades. Therefore, the procedures for upgrading specific operating systems are not covered in this topic.

This topic primarily describes all the additional steps such as backing up essential files and removing and reinstalling all the necessary Cloudera Enterprise packages and parcels.

Note: You must determine whether to upgrade the operating system or Cloudera Manager first:

For example, consider the following upgrade from Cloudera Manager 5.13.3 to Cloudera Manager 7.1.4 and an operating system upgrade from RHEL 7.6 to RHEL 7.8

- Cloudera Manager 5.13.3 supports only RHEL 7.6
- Cloudera Manager 7.1.4 supports RHEL 7.6 and RHEL 7.8

In this case you must upgrade Cloudera Manager first, otherwise Cloudera Manager version 5.13.3 would be deployed on an unsupported operating system (RHEL 7.6) and may fail.

Step 1: Getting Started with Operating System Upgrades

Prerequisites

- Ensure that the versions of Cloudera Manager and CDH or Cloudera Runtime support your new operating system.
  - See Operating System Requirements for Cloudera Base on premises.

  If you are using unsupported versions, see Upgrade Cloudera Manager or Upgrading a Cloudera Runtime Cluster.
- Ensure that the host has access to the Cloudera Manager server, daemon and agent packages that are supported for the new operating system, either by having access to https://archive.cloudera.com or a local package repository.
- Ensure that the Cloudera Manager server has access to the parcels that are using supported for the new Operating System, either by having access to https://archive.cloudera.com or a local parcel repository.
- If you have a patched package or parcel installed, make sure you have the same package or parcel for the new Operating System and it has been made available to Cloudera Manager.
- Understand that performing a major release upgrade for the operating system in-place may be quite tricky and risky.

Step 2: Backing Up Host Files Before Upgrading the Operating System

This topic describes how to backup important files on your host before upgrading the operating system.

Backing Up

1. Create a top-level backup directory.

```
export CM_BACKUP_DIR="`date +%F`-CM"
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

2. Back up the Agent directory and the runtime state.

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-agent.tar --exclude=*.sock /et
c/cloudera-scm-agent /etc/default/cloudera-scm-agent /var/run/cloudera-s
cm-agent /var/lib/cloudera-scm-agent
```

3. Back up the Server directories:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-scm-s
erver /etc/default/cloudera-scm-server
```

4. Back up the Cloudera Manager databases. See Backing up Cloudera Manager databases on page 58

Note: Backup is recommended but not always required for a minor release upgrade.

Backing up Cloudera Manager databases

Cloudera recommends that you schedule regular backups of the databases that Cloudera Manager uses to store configuration, monitoring, and reporting data and for managed services that require a database:

Backing Up PostgreSQL Databases

To back up a PostgreSQL database, use the same procedure whether the database is embedded or external:

1. Log in to the host where the Cloudera Manager Server is installed.
2. Get the name, user, and password properties for the Cloudera Manager database from /etc/cloudera-scm-server/db.properties:

```
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=NnYfWIjlbk
```

3. Run the following command as root using the parameters from the preceding step:

```
# pg_dump -h HOSTNAME -p 7432 -U scm > /tmp/scm_server_db_backup.$(date +
%Y%m%d)
```

4. Enter the password from the com.cloudera.cmf.db.password property in step 2.
5. To back up a database created for one of the roles on the local host as the *ROLEUSER* user:

```
# pg_dump -h HOSTNAME -p 7432 -U ROLEUSER > /tmp/ROLEDB
```

6. Enter the password specified when the database was created.

Backing Up MariaDB Databases

To back up the MariaDB database, run the mysqldump command on the MariaDB host, as follows:

```
mysqldump -hHOSTNAME -uUSERNAME -pPASSWORD DATABASE > /tmp/DATABASE-
BACKUP.sql
```

For example, to back up the Activity Monitor database amon created in Creating Databases for Cloudera Software, on the local host as the root user, with the password amon_password:

```
mysqldump -pamon_password amon > /tmp/amon-backup.sql
```

To back up the sample Activity Monitor database amon on remote host myhost.example.com as the root user, with the password amon_password:

```
mysqldump -hmyhost.example.com -uroot -pamon_password amon > /tmp/amon-backu
p.sql
```

Backing Up MySQL Databases

To back up the MySQL database, run the mysqldump command on the MySQL host, as follows:m

```
mysqldump -hHOSTNAME -uUSERNAME -pPASSWORD DATABASE > /tmp/DATABASE-
BACKUP.sql
```

For example, to back up the Activity Monitor database amon created in Creating Databases for Cloudera Software, on the local host as the root user, with the password amon_password:

```
mysqldump -pamon_password amon > /tmp/amon-backup.sql
```

To back up the sample Activity Monitor database amon on remote host myhost.example.com as the root user, with the password amon_password:

```
mysqldump -hmyhost.example.com -uroot -pamon_password amon > /tmp/amon-backu
p.sql
```

You can back up all database using the following command:

```
mysqldump --all-databases -pPASSWORD > /tmp/all1/all.sql
```

Backing Up Oracle Databases

For Oracle, work with your database administrator to ensure databases are properly backed up.

Step 3: Before You Upgrade the Operating System

This topic describes steps you must perform before upgrading the operating system on a host managed by Cloudera Manager.

Decommission and Stop Running Roles

1. Log in to the Cloudera Manager Admin Console.
2. From the All Hosts page, select the host that you wish to upgrade. Cloudera recommends that you upgrade only one host at a time.
3. Select Begin Maintenance (Suppress Alerts/Decommission) from the Actions menu.
4. Select Host Decommission from the Actions menu. Any roles that do not require decommission will be skipped.

5. If the operating system upgrade procedure takes less than 30 minutes per node, you do not need to decommission the DataNode.

   If the Cloudera Manager and CDH/ version are both 5.14 or greater, you can also choose the Take DataNode Offline feature.

   If in doubt, decommission the roles.

   - When a DataNode is decommissioned, the NameNode ensures that every block from the DataNode is still available across the cluster as specified by the replication factor. This procedure involves copying blocks off the DataNode in small batches. In cases where a DataNode has several thousand blocks, decommissioning takes several hours.
   - When a DataNode is turned off without being decommissioned:
     - The NameNode marks the DataNode as dead after a default of 10m 30s (controlled by the dfs.heartbeat.interval and dfs.heartbeat.recheck.interval configuration properties).
     - The NameNode schedules the missing replicas to be placed on other DataNodes.
     - When the DataNode comes back online and reports to the NameNode, the NameNode schedules blocks to be copied to it while other nodes are decommissioned or when new files are written to HDFS.
   - You can also speed up the decommissioning of a DataNode by increasing values for these properties:
     - dfs.max-repl-streams: The number of simultaneous streams used to copy data.
     - dfs.balance.bandwidthPerSec: The maximum amount of bandwidth that each DataNode can utilize for balancing, in bytes per second.
     - dfs.namenode.replication.work.multiplier.per.iteration: NameNode configuration requiring a restart, defaults to 2 but can be raised to 10 or higher.

       This determines the total amount of block transfers to begin in parallel at a DataNode for replication, when such a command list is being sent over a DataNode heartbeat by the NameNode. The actual number is obtained by multiplying this value by the total number of live nodes in the cluster. The result number is the number of blocks to transfer immediately, per DataNode heartbeat.
6. Once that is completed, select the same host again and choose Stop Roles on Hosts.

Warning: If you have not enabled high availability for HDFS, HBase, MapReduce, YARN, Oozie, or Sentry, stopping the running single master role will cause an outage for that service. Specifically, secondary roles on other hosts will stop abruptly. Cloudera recommends that you stop these services prior to the host upgrade.

Important: When upgrading hosts that are part of a ZooKeeper quorum, ensure that the majority of the quorum is still available.

Stop Cloudera Manager Agent

1. Hard Stop the Agent.
   RHEL 7, SLES 12, Ubuntu 18.04 and higher

   ```
   sudo systemctl stop cloudera-scm-supervisord.service
   ```

   RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04 or 14.04

   ```
   sudo service cloudera-scm-agent hard_stop
   ```

Important: This will ask you to confirm with hard_stop_confirmed because this will terminate any Hadoop services on the host (if any) unconditionally.

Stop Cloudera Manager Server & Agent

1. Hard Stop the Agent.
   RHEL 7, SLES 12, Ubuntu 18.04 and higher

   ```
   sudo systemctl stop cloudera-scm-supervisord.service
   ```

   RHEL 5 or 6, SLES 11, Debian 6 or 7, Ubuntu 12.04 or 14.04

   ```
   sudo service cloudera-scm-agent hard_stop
   ```

   Important: This will ask you to confirm with hard_stop_confirmed because this will terminate any Hadoop services on the host (if any) unconditionally.
2. Stop the .
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Stop .
3. Stop the Server.

   ```
   sudo systemctl stop cloudera-scm-server
   ```

Stop Databases

1. If you are using the embedded PostgreSQL database, stop the Embedded PostgreSQL database:

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

   If you are not using the embedded PostgreSQL database and you attempt to stop it, you might see a message indicating that the service cannot be found.

   In case of using the embedded PostgreSQL, if you see a message that the shutdown failed, then the embedded database is still running, probably because services are connected to the Hive metastore. If the database shutdown fails due to connected services, issue the following command:

   ```
   sudo service cloudera-scm-server-db next_stop_fast
   sudo service cloudera-scm-server-db stop
   ```
2. If there are other database servers running on this host, they must be stopped also.

Remove Packages & Parcels

Packages for the older operating system won't be able to start on the new operating system. Remove old packages from the host.

1. **RHEL / CentOS**

   ```
   sudo yum remove cloudera-manager-daemons cloudera-manager-
   agent cloudera-manager-server-db-2
   ```

   SLES

   ```
   sudo zypper remove cloudera-manager-daemons cloudera-manager-
   agent cloudera-manager-server-db-2
   ```

   Ubuntu

   ```
   sudo apt-get purge cloudera-manager-daemons cloudera-manager-
   agent cloudera-manager-server-db-2
   ```

2. **RHEL / CentOS**

```
sudo yum remove cloudera-manager-server cloudera-manager-daemons
 cloudera-manager-agent cloudera-manager-server-db-2
```

SLES

```
sudo zypper remove cloudera-manager-server cloudera-manager-
daemons cloudera-manager-agent cloudera-manager-server-db-2
```

Ubuntu

```
sudo apt-get purge cloudera-manager-server cloudera-manager-
daemons cloudera-manager-agent cloudera-manager-server-db-2
```

3. Remove old CDH parcels from the host. These are built for your old operating system.

   The Cloudera Manager agent will download and activate the proper parcel for the new operating system when it is started.

   Empty the contents of the following directories. These are the defaults for parcel storage - if you use other directories, please change accordingly.

```
sudo rm -rf /opt/cloudera/parcels/*
```

```
sudo rm -rf /opt/cloudera/parcel-cache/*
```

Upgrade the Operating System

Important: When there are no Hadoop services or Cloudera Manager roles running from this host, you may proceed to upgrade the operating system of this host, make sure to leave the data partitions (for example, dfs.data.dir) unchanged.

Use the operating system upgrade procedures provided by your operating system vendor (for example: RedHat or Ubuntu) to download their software and perform the operating system upgrade.

Step 4: After You Upgrade the Operating System

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

This topic describes how to upgrade the operating system on a Cloudera Manager managed host.

Establish Access to the Software

Cloudera Manager needs access to a package repository that contains the updated software packages. You can choose to access the Cloudera public repositories directly, or you can download those repositories and set up a local repository to access them from within your network. If your cluster hosts do not have connectivity to the Internet, you must set up a local repository.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Log in to each cluster host.

```
ssh CLUSTER_HOST
```

3. Remove any older files in the existing repository directory:
   RHEL / CentOS

   ```
   sudo rm /etc/yum.repos.d/cloudera*manager.repo*
   ```

   SLES

   ```
   sudo rm /etc/zypp/repos.d/cloudera*manager.repo*
   ```

   Ubuntu

   ```
   sudo rm /etc/apt/sources.list.d/cloudera*.list*
   ```
4. Fill in the form at the top of this page.
5. Create a repository file so that the package manager can locate and download the binaries. Do one of the following, depending on whether or not you are using a local package repository:
   - Using a local package repository. (Required when cluster hosts do not have access to the internet.)
     a. Configure a local package repository hosted on your network.
     b. In the Package Repository URL, replace the entire URL with the URL for your local package repository. A username and password are not required to access local repositories.
     c. Click Apply.
   - Using the Cloudera public repository
     a. Substitute your *USERNAME* and *PASSWORD* in the Package Repository URL where indicated in the URL.
     b. Click Apply

   Tip: If you have a mixed operating system environment, adjust the Operating System filter at the top of the page for each operating system. The guide will generate the repo file for you automatically here.
6. **RHEL / CentOS**

   Create a file named /etc/yum.repos.d/cloudera-manager.repo with the following content:

   ```
   [cloudera-manager]
   # Packages for Cloudera Manager
   name=Cloudera Manager
   baseurl=https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/5.15
   gpgkey=https://archive.cloudera.com/cm5/redhat/7/x86_64/cm/RPM-
   GPG-KEY-cloudera
   gpgcheck=1
   ```

   SLES

   Create a file named /etc/zypp/repos.d/cloudera-manager.repo with the following content:

   ```
   [cloudera-manager]
   # Packages for Cloudera Manager
   name=Cloudera Manager
   baseurl=https://archive.cloudera.com/cm5/sles/12/x86_64/cm/5.15
   gpgkey=https://archive.cloudera.com/cm5/sles/12/x86_64/cm/RPM-
   GPG-KEY-cloudera
   gpgcheck=1
   ```

   Ubuntu

   Create a file named /etc/apt/sources.list.d/cloudera_manager.list with the following content:

   ```
   # Packages for Cloudera Manager
   ```

```
deb https://archive.cloudera.com/cm5/debian/jessie/amd64/cm/
 jessie-cm5.15 contrib
deb-src https://archive.cloudera.com/cm5/debian/jessie/amd64/cm/
 jessie-cm5.15 contrib
```

Run the following command:

```
sudo apt-get update
```

The repository file, as created, refers to the most recent maintenance release of the specified minor release. If you would like to use a specific maintenance version, for example 5.15.1, replace 5.15 with 5.15.1 in the generated repository file shown above.

7. A Cloudera Manager upgrade can introduce new package dependencies. Your organization may have restrictions or require prior approval for installation of new packages. You can determine which packages may be installed or upgraded:
   RHEL / CentOS

   ```
   yum deplist cloudera-manager-agent
   ```

   SLES

   ```
   zypper info --requires cloudera-manager-agent
   ```

   Ubuntu

   ```
   apt-cache depends cloudera-manager-agent
   ```

Reinstall Daemon & Agent Packages

Re-install the removed packages.

1. Install the agent packages. Include the cloudera-manager-server-db-2 package in the command only if you are using the embedded PostgreSQL database.
   RHEL / CentOS

   ```
   sudo yum clean all
   sudo yum install cloudera-manager-daemons cloudera-manager-ag
   ent cloudera-manager-server-db-2
   ```

   SLES

   ```
   sudo zypper clean --all
   sudo zypper install cloudera-manager-daemons cloudera-manager-ag
   ent cloudera-manager-server-db-2
   ```

   Ubuntu

   ```
   sudo apt-get clean
   ```

   ```
   sudo apt-get update
   ```

   ```
   sudo apt-get install cloudera-manager-daemons cloudera-manager-a
   gent cloudera-manager-server-db-2
   ```

Verify that the configuration files (that were backed up) are intact. Correct if necessary.

Reinstall Server, Daemon & Agent Packages

Re-install the removed packages.

1. Install the packages. Include the cloudera-manager-server-db-2 package in the command only if you are using the embedded PostgreSQL database.
   RHEL / CentOS

   ```
   sudo yum clean all
   sudo yum install cloudera-manager-server cloudera-manager-daemons
    cloudera-manager-agent cloudera-manager-server-db-2
   ```

   SLES

   ```
   sudo zypper clean --all
   sudo zypper install cloudera-manager-server cloudera-manager-dae
   mons cloudera-manager-agent cloudera-manager-server-db-2
   ```

   Ubuntu

   ```
   sudo apt-get clean
   ```

   ```
   sudo apt-get update
   ```

   ```
   sudo apt-get install cloudera-manager-server cloudera-manager-da
   emons cloudera-manager-agent cloudera-manager-server-db-2
   ```

Verify that the configuration files (that were backed up) are intact. Correct if necessary.

If you customized the /etc/cloudera-scm-agent/config.ini file, your customized file is renamed with the extension .rpm save or .dpkg-old. Merge any customizations into the /etc/cloudera-scm-agent/config.ini file that is installed by the package manager.

Edit repository file to point to the repositories designed for your new operating system.

Start Databases

1. If you are using the embedded PostgreSQL database, start the database:

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

2. If there were database servers stopped, they must be restarted.

Start Server & Agent

The appropriate services typically will start automatically on reboot. Otherwise, start the Server & Agent as necessary.

1. Start the rpcbind service if it is not automatically started.

   ```
   sudo service rpcbind start
   ```

2. Start the Agent.

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

   If the agent starts without errors, no response displays.

3. Start the Server.

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

4. Verify that the Agent downloaded a proper parcel for your new operating system. You can use the following command to check in logs for downloaded parcels:

```
grep "Completed download" /var/log/cloudera-scm-agent/cloudera-scm-agent
.log
```

(Download might take some time. Look for the operating system in the names of the downloaded parcels.

Start Roles

1. From the All Hosts page, select the host that you have just upgraded.
2. Choose End Maintenance (Enable Alerts/Decommission) from the Actions menu and confirm.
3. Start any roles that were running on this host and were stopped.
4. Choose Host Recommission from the Actions menu and confirm.
5. Choose Start Roles on Hosts from the Actions menu and confirm.
6. Start any services that were stopped due to lack of high availability.

Upgrading Cloudera Manager 6

Steps to upgrade Cloudera Manager.

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

Important: To upgrade Cloudera Manager from any 5.x or 6.x version to a higher version of Cloudera Manager 5.x or 6.x, do not perform the following instructions on this page. Instead, use the instructions from the Cloudera Enterprise Upgrade Guide.

Important:

From Cloudera Manager 7.13.1 onwards Cloudera Manager by default does not support Cloudera Runtime 6 cluster because CDH 6 jars have security vulnerabilities.

If you want to run Cloudera Manager 7.13.1 or higher versions with CDH 6 cluster then you need to install an additional cloudera-manager-daemons-cdh6 rpm/debian package on every cluster host after installation/ upgrade to Cloudera Manager 7.13.1. This package is available along with the other Cloudera Manager packages on the public repository.

These topics describes how to upgrade Cloudera Manager from any 5.x or 6.x version to a higher version of Cloudera Manager 7.1 and higher, including major, minor, and maintenance releases. The upgrade procedures use operating system command-line package commands to upgrade Cloudera Manager, and then complete the upgrade using Cloudera Manager.

When you upgrade Cloudera Manager, you use RPM-based package commands to upgrade the software on the Cloudera Manager server host and then Cloudera Manager manages upgrading the Cloudera Manager Agents on the remaining managed hosts. Cloudera Manager can also automatically install some versions of the required JDK on the managed hosts.

Upgrades are not supported between all versions of Cloudera Manager, CDH, or Cloudera Runtime. See Supported Upgrade Paths.

Cloudera Navigator is also upgraded when you upgrade Cloudera Manager 5.x or 6.x. Cloudera Navigator has been replaced by Apache Atlas as of Cloudera Runtime 7.0.3. If you are using Cloudera Manager 7.0.3 or higher to manage CDH clusters, those clusters can continue using Cloudera Navigator.

The Cloudera Manager upgrade process does the following:

- Upgrades the database schema to reflect the current version.
- Upgrades the Cloudera Manager Server and all supporting services.
- Upgrades the Cloudera Manager agents on all hosts.

- Redeploys client configurations to ensure that client services have the most current configuration.
- Upgrades Cloudera Navigator (for upgrades to Cloudera Manager 7.1, you can transition Cloudera Navigator to Apache Atlas).

To upgrade Cloudera Manager, you perform the following tasks:

1. Back up the Cloudera Manager server databases, working directories, and several other entities. These backups can be used to restore your Cloudera Manager deployment if there are problems during the upgrade.
2. Upgrade the Cloudera Manager server software on the Cloudera Manager host using package commands from the command line (for example, yum on RHEL systems). Cloudera Manager automates much of this process and is recommend for upgrading and managing your CDH/Cloudera Runtime clusters.
3. Upgrade the Cloudera Manager agent software on all cluster hosts. The Cloudera Manager upgrade wizard can upgrade the agent software (and, optionally, the JDK), or you can install the agent and JDK software manually. The CDH or Cloudera Runtime software is not upgraded during this process.

Upgrading Cloudera Manager does not upgrade Cloudera Runtime clusters. See Upgrading a CDH 6 Cluster on page 121 for upgrade procedures.

Step 1: Getting Started Upgrading Cloudera Manager 6

Note: Ubuntu 18 Operating System is not supported from Cloudera Manager 7.11.3 to Cloudera Manager 7.11.3 CHF4 versions. You must upgrade the Operating System from Ubuntu 18 to Ubuntu 20 before you upgrade to Cloudera Manager 7.11.3 CHF4. For performing major OS upgrade, see Upgrading the Operating System to a new Major Version.

Note: Not all combinations of Cloudera Manager and Cloudera Runtime are supported. Ensure that the version of Cloudera Manager you are using supports the version of Cloudera Runtime you have selected. For details, see Cloudera Manager support for Cloudera Runtime, CDH and CDP Private Cloud Experiences .

Note: Cloudera Private Cloud Data Services version 1.3.4 requires Cloudera Manager 7.5.5 and Cloudera Runtime version 7.1.6 or 7.1.7 For more information, see Cloudera Private Cloud Data Services.

Note: If you are upgrading to Cloudera Manager 7.5.1 or higher in order to install CDP Private Cloud Experiences version 1.3.1, you must use Cloudera Runtime version 7.1.6 or 7.1.7. For more information, see CDP Private Cloud Exeriences.

Important: Upgrades to Cloudera Manager 7.0.3 are not supported.

Note: The embedded PostgreSQL version within Key Trustee Server is upgraded from 14.2 to 14.16 in Cloudera Runtime 7.1.9 SP1 CHF10. If you are using KTS, you need to upgrade your Cloudera Manager to 7.11.3 CHF17 first in order to upgrade to Cloudera Runtime 7.1.9 SP1 CHF10. Upgrading from CDH6 to a Cloudera Runtime 7.1.9 SP1 CHF is supported with Cloudera Manager 7.11.3 CHF11, and not supported with Cloudera Manager 7.11.3 CHF12 and later versions.

Before you upgrade Cloudera Manager, you need to gather some information and review the limitations and release notes. Fill in the My Environment form below to customize your Cloudera Manager upgrade procedures. See the Collect Information section below for assistance in locating the required information.

Note: If you are upgrading to Cloudera Manager 7.5.1 or higher in order to install CDP Private Cloud Experiences version 1.3.1, you must use Cloudera Runtime version 7.1.6.

Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported

Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher

Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.

Warning: Before upgrading CDH 5 clusters with Sentry to Cloudera Runtime 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.

Important: If HDFS ACL sync is enabled (hdfs_sentry_sync_enable=true) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see Installing Ranger RMS.

Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Warning: You cannot upgrade from a cluster that uses Oracle 12.

Warning: You cannot upgrade from a cluster that uses Oracle 19.

Collect Information

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Collect the following information about your environment and fill in the form above. This information will be remembered by your browser on all pages in this Upgrade Guide.
   a. The current version of the Operating System:

```
lsb_release -a
```

   Database parameters:

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

```
...
com.cloudera.cmf.db.type=mysql
com.cloudera.cmf.db.host=DATABASE_HOSTNAME:DATABASE_PORT
com.cloudera.cmf.db.name=SCM
com.cloudera.cmf.db.user=SCM
com.cloudera.cmf.db.password=SOME_PASSWORD
```

b. Log in to the Cloudera Manager Admin console and find the following:

1. The version of Cloudera Manager used in your cluster. Go to Support About .
2. The version of the JDK deployed in the cluster. Go to Support About .

Preparing to Upgrade Cloudera Manager

- Access to Cloudera Manager binaries for production purposes requires authentication. In order to download the software, you must first have an active subscription agreement and obtain a license key file along with the required authentication credentials (username and password). See Cloudera Manager Download Information.
- You must have SSH access to the Cloudera Manager server hosts and be able to log in using the root account or an account that has password-less sudo permission for all hosts.
- Review the following when upgrading to Cloudera Manager 7.1 or higher:

  Cloudera Private Cloud Base Requirements and Supported Versions
- You may be required to upgrade the operating system before upgrading. See Operating System Requirements to determine operating system support for the version of Cloudera Manager you are upgrading to. Depending on the support, you may need to upgrade the operating system.

  If you must or choose to upgrade to a supported operating system, you must determine whether to upgrade the operating system first or Cloudera Manager first. If the current version of Cloudera Manager and the version you are upgrading to both support a newer version of the operating system but the new version of Cloudera Manager does not support the older operating system, you must upgrade to the newer operating system before upgrading Cloudera Manager. If this is not true, then you must upgrade Cloudera Manager before upgrading the operating system.

  See Upgrading the Operating System.
- Install a supported version of the Java Development Kit (JDK) on all hosts. If you are upgrading to Cloudera Manager and CDP Private Cloud Base 7.1.1 and higher, you can choose to install OpenJDK 1.8 instead of the Oracle JDK.

  There two options for JDK installation:

  - Manually install the Oracle JDK or OpenJDK on all hosts.
  - Manually install the Oracle JDK 1.8 on the Cloudera Manager host, and then select the Install Oracle Java SE Development Kit checkbox when prompted while running the Cloudera Manager Upgrade wizard.

  See Upgrading the JDK on page 44
- Review the Release Notes.

  - Cloudera Base on premises

    - Cloudera Manager Release Notes
    - Cloudera Runtime Release Notes
- Review the Cloudera Security Bulletins.
- The embedded PostgreSQL database installed with the Trial Installer is not supported in production environments because a trial installation cannot easily be upgraded, backed up, or migrated into a production-ready configuration without manual steps requiring down time.

  Consider Migrating from the Cloudera Manager Embedded PostgreSQL Database Server to an External PostgreSQL Database before upgrading Cloudera Manager.

- If your cluster uses Oracle for any databases, before upgrading CDH 5, 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 effect this change can have on your system.

Using Python 3.8 with the Cloudera Manager Agents

If you require Python 3.8 to be used on your cluster hosts, you must install Python 3.8 before you upgrade Cloudera Manager. See Installing Python 3.8 for Cloudera Manager 7.7.3

Important: Cloudera Manager 7.7.3 should only be used when you need to use Python 3.8 for the Cloudera Manager agents. You must install Python 3.8 on all hosts before installing or upgrading to Cloudera Manager 7.7.3. Cloudera Manager 7.7.3 is only supported with RHEL 7.9 . 8.4, and 8.6 . See the CDP Private Cloud Base Installation guide for more information.

Using Python 3 with the Cloudera Manager Agents

You must install Python 3 on all hosts before upgrading to Cloudera Manager 7.11.3. See Installing Python 3.

Using Python 3 with the Cloudera Manager Agents

You must install Python 3 on all hosts before upgrading to Cloudera Manager 7.13.1. See Installing Python 3.

Migrating from Python 3.8 to Python 3.9 on RHEL 8.8 or RHEL 8.10

Cloudera recommends you to install Python 3.9 before upgrading Cloudera Manager to 7.13.1 version to ensure smooth transition with minimal downtime.

(Recommended) Installing Python 3.9 on RHEL 8 before upgrading Cloudera Manager to 7.13.1 and CDP Runtime to 7.3.1

Learn how to migrate Cloudera Manager and CDP Runtime from Python 3.8 to Python 3.9 on RHEL 8.8 or RHEl 8.10. This is the recommended workflow for customers upgrading Cloudera Manager to 7.13.1 version and CDP Runtime to 7.3.1 version since the minimum recommended version of Python is now 3.9.

1. Install Python 3.9 from standard repository. For information about installing Python 3.9 on RHEL 8, see Installing Python 3.9 standard package on RHEL 8.

   Important:

   - Install Python 3.9 before upgrading Cloudera Manager to 7.13.1 version to avoid performing a Cloudera Manager Agent restart later. If you install Python 3.9 after upgrading Cloudera Manager to 7.13.1 version, then you must perform Cloudera Manager Agent restart after installing Python 3.9.
   - Install Python 3.9 before upgrading CDP Runtime to 7.3.1 version as Hue requires Python 3.9 and does not start until you install Python 3.9. If you want to upgrade CDP Runtime to 7.3.1, then you must install Python 3.9 on all Hue servers before you upgrade CDP Runtime to 7.3.1 version.

2. Upgrade Cloudera Manager to 7.13.1 version. For information about upgrading Cloudera Manager, see Upgrading Cloudera Manager 7.
3. Upgrade CDP Runtime to 7.3.1 version. For information about upgrading Cloudera Runtime Cluster, see Upgrading a Cloudera Runtime Cluster.

4. Uninstall Python 3.8. To uninstall Python 3.8 run the following command:

```
yum remove python38
```

(Not Recommended) Installing Python 3.9 on RHEL 8 after upgrading Cloudera Manager to 7.13.1 and CDP Runtime to 7.3.1

Learn how to upgrade Cloudera Manager and CDP Runtime while delaying the upgrade of Python from 3.8 to 3.9 version on RHEL 8.8 or RHEL 8.10. This is not a recommended workflow due to Hue supports only Python 3.9 in CDP Runtime 7.3.1.

Important: Hue does not start until you install Python 3.9.

1. Upgrade Cloudera Manager to 7.13.1 version. For information about upgrading Cloudera Manager, see Upgrading Cloudera Manager 7.
2. Upgrade CDP Runtime to 7.3.1 version. For information about upgrading Cloudera Runtime Cluster, see Upgrading a Cloudera Runtime Cluster.

Warning:

Hue will fail to start. Continue the Cloudera Runtime Cluster upgrade process manually by following the manual upgrade instructions from Manual upgrade to CDP Private Cloud Base.

3. Install Python 3.9 from standard repository. For information about installing Python 3.9 on RHEL 8, see Installing Python 3.9 standard package on RHEL 8.

Important: If you install Python 3.9 after upgrading Cloudera Manager to 7.13.1 version and CDP Runtime to 7.3.1 version, then you must restart the Cloudera Manager Agent and all CDP Runtime services respectively after installing Python 3.9.

4. Restart Cloudera Manager Agents. For information about restarting Cloudera Manager Agents, see Starting, Stopping, and Restarting Cloudera Manager Agents.

   After restarting Cloudera Manager Agents, Cloudera Manager Agent will pick up the highest Python version available between Python 3.8 and Python 3.9.
5. Resart all CDP Runtime services so that they will pick up the Python 3.9 version. To restart all CDP Runtime services, restart the cluster. For information about restarting a cluster, see Restarting a Cluster.

Important: Hue starts now successfully.

6. Uninstall Python 3.8. To uninstall Python 3.8 run the following command:

```
yum remove python38
```

When Cloudera Manager is upgraded to 7.13.1 version and when CDP Runtime is either 7.1.8, 7.1.7 SP3 or 7.1.9 for RHEL8, all the CDP Runtime process will continue to use Python 3.8, whereas Cloudera Manager Agent will pick up the highest Python version available between Python 3.8 and Python 3.9.

Step 2: Backing Up Cloudera Manager 6

Cloudera recommends that you backup Cloudera Manager before upgrading.

This topic contains procedures to back up Cloudera Manager. Cloudera recommends that you perform these backup steps before upgrading. The backups will allow you to rollback your Cloudera Manager upgrade if needed.

Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported

Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher

Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.

Warning: Before upgrading CDH 5 clusters with Sentry to Cloudera Runtime 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.

Important: If HDFS ACL sync is enabled (hdfs_sentry_sync_enable=true) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see Installing Ranger RMS.

Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Collect Information for Backing Up Cloudera Manager

Information you should collect before backing up Cloudera Manager.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Collect database information by running the following command:

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

For example:

```
...
com.cloudera.cmf.db.type=...
com.cloudera.cmf.db.host=DATABASE_HOSTNAME:DATABASE_PORT
com.cloudera.cmf.db.name=SCM
com.cloudera.cmf.db.user=SCM
com.cloudera.cmf.db.password=SOME_PASSWORD
```

3. Collect information (host name, port number, database name, user name and password) for the following databases.

   - Reports Manager

   You can find the database information by using the Cloudera Manager Admin Console. Go to Clusters Cloudera Management Service Configuration and select the Database category. You may need to contact your database administrator to obtain the passwords.
4. Find the host where the Service Monitor, Host Monitor and Event Server roles are running. Go to Clusters Cloudera Management Service Instances and note which hosts are running these roles.

Back Up Cloudera Manager Agent

Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups, defined by the environment variable CM_BACKUP_DIR, which is used in all the backup commands. You may change these destination paths in the command as needed for your deployment.

The tar commands in the steps below may return the following message. It is safe to ignore this message:

```
tar: Removing leading `/' from member names
```

Backup up the following Cloudera Manager agent files on all hosts:

- Create a top level backup directory.

```
export CM_BACKUP_DIR="`date +%F`-CM"
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

- Back up the Cloudera Manager Agent directory and the runtime state.

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-agent.tar --exclude=*.sock /etc/cloudera-scm-agent /etc/default/cloudera-scm-agent /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent
```

- Back up the existing repository directory.
  RHEL / CentOS

  ```
  sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
  ```

  SLES

  ```
  sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
  ```

  Ubuntu

  ```
  sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources.list.d
  ```

Back Up the Cloudera Management Service

Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups. You may change these destination paths in the command as needed for your deployment.

1. Stop the Cloudera Management Service.
   a. Log in to the Cloudera Manager Admin Console.
   b. Go to Cloudera Management Service .
   c. Select Actions Stop .
2. On the host where the Service Monitor role is configured to run, backup the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-SERVICE-MONITOR /var/lib/cloudera-service-monitor-`date +%F`-CM
```

3. On the host where the Host Monitor role is configured to run, backup the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-HOST-MONITOR /var/lib/cloudera-host-monitor-
`date +%F`-CM
```

4. On the host where the Event Server role is configured to run, back up the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-SCM-EVENTSERVER /var/lib/cloudera-scm-event
server-`date +%F`-CM
```

5. Restart the Cloudera Management Service if you are not upgrading Cloudera Manager immediately.
   a. Log in to the Cloudera Manager Admin Console.
   b. Select Clusters Cloudera Management Service .
   c. Select Actions Start .

Back Up Cloudera Navigator Data

1. **Important:** Upgrading from Cloudera Manager 5.9 (Navigator 2.8) and earlier can take a significant amount of time, depending on the size of the Navigator Metadata storage directory. When the Cloudera Manager upgrade process completes and Cloudera Navigator services restart, the Solr indexing upgrade automatically begins. No other actions can be performed until Solr indexing completes (a progress message displays during this process). It can take as long as two days to upgrade a storage directory with 60 GB. To help mitigate this extended upgrade step, make sure to clear out all unnecessary metadata using purge, check the size of the storage directory, and consider rerunning purge with tighter conditions to further reduce the size of the storage directory.
2. Make sure a purge task has run recently to clear stale and deleted entities.
   - You can see when the last purge tasks were run in the Cloudera Navigator console (From the Cloudera Manager Admin console, go to ClustersCloudera Navigator. Select AdministrationPurge Settings.)
   - If a purge hasn't run recently, run it by editing the Purge schedule on the same page.
   - Set the purge process options to clear out as much of the backlog of data as you can tolerate for your upgraded system. See Managing Metadata Storage with Purge.
3. Stop the Navigator Metadata Server.
   a. Go to ClustersCloudera Management ServiceInstances.
   b. Select Navigator Metadata Server.
   c. Click Actions for SelectedStop.
4. Back up the Cloudera Navigator Solr storage directory.

```
sudo cp -rp /VAR/LIB/CLOUDERA-SCM-NAVIGATOR /var/lib/cloudera-scm-navig
ator-`date +%F`-CM
```

5. If you are using an Oracle database for audit, in SQL*Plus, ensure that the following additional privileges are set:

```
GRANT EXECUTE ON sys.dbms_crypto TO nav;
GRANT CREATE VIEW TO NAV;
```

   where *NAV* is the user of the Navigator Audit Server database.

Stop Cloudera Manager Server & Cloudera Management Service

1. Stop the Cloudera Management Service.
   a. Log in to the Cloudera Manager Admin Console.
   b. Select Clusters Cloudera Management Service .
   c. Select Actions Stop .

2. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

3. Stop the Cloudera Manager Server.

```
sudo systemctl stop cloudera-scm-server
```

Back Up the Cloudera Manager Databases

During the upgrade, Cloudera Manager modifies the schema of the Cloudera Manager database. In case of failures during the upgrade, it may be necessary to rollback to the previous version of Cloudera Manager while addressing the upgrade failures.

When performing a rollback to a previous version, the Cloudera Manager Database must be restored to the previous database schema. For this reason, you must do the Cloudera Manager database backup, so that it can be restored if a rollback is necessary.

Tip:

You can get the name, user, and password properties for the Cloudera Manager database from the /etc/cloudera-scm-server/db.properties file:

```
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=NnYfWIjlbk
```

1. Back up the Cloudera Manager server database as per the instructions provided in your respective database documentation on how to backup and restore the databases.

2. Back up All other Cloudera Manager databases - Use the database information that you collected in a previous step. You may need to contact your database administrator to obtain the passwords.

   These databases can include the following:

   - 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 database. If this information is not available, your cluster cannot start or function correctly. You must schedule and maintain regular backups of the 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.
   - Sentry Server - Contains authorization metadata. Relatively small.
   - Navigator Audit Server - Contains auditing information. In large clusters, this database can grow large.(Only available when installing CDH 5 or CDH 6 clusters.)
   - Navigator Metadata Server - Contains authorization, policies, and audit report metadata. Relatively small. (Only available when installing CDH 5 or CDH 6 clusters.)
   - DAS PostgreSQL server - Contains Hive and Tez event logs and DAG information. Can grow very large.
   - Ranger Admin - Contains administrative information such as Ranger users, groups, and access policies. Medium-sized.
   - 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.

   For more information about the number of databases that should be backed up, and restored if necessary, see Required Databases.

Back Up Cloudera Manager Server

Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups, defined by the environment variable CM_BACKUP_DIR, which is used in all the backup commands. You may change these destination paths in the command as needed for your deployment.

The tar commands in the steps below may return the following message. It is safe to ignore this message:

```
tar: Removing leading `/' from member names
```

1. Log in to the Cloudera Manager Server host.

   ```
   ssh MY_CLOUDERA_MANAGER_SERVER_HOST
   ```

2. Create a top-level backup directory.

```
export CM_BACKUP_DIR="`date +%F`-CM"
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

3. **a.** Back up the Cloudera Manager Server directories along with the CSP key file when Credential Storage Provider (CSP) is enabled:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-s
cm-server /etc/default/cloudera-scm-server ***/path/to/csp/keys***
```

Important: In the above command, you must replace the ***/path/to/csp/key*** with the actual path where the CSP key has been stored from the following options:

- /var/lib/cloudera-scm-server/csp-data
- /opt/cloudera/
- /etc/cloudera/
- /var/cloudera/

b. Back up the Cloudera Manager Server directories without CSP key file:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-s
cm-server /etc/default/cloudera-scm-server
```

4. Back up the existing repository directory.
RHEL / CentOS

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
```

SLES

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
```

Ubuntu

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources
.list.d
```

(Optional) Start Cloudera Manager Server & Cloudera Management Service

Start the Cloudera Manager server and Cloudera Manager Management service.

If you will be immediately upgrading Cloudera Manager, skip this step and continue with Step 3: Upgrading the Cloudera Manager Server on page 78.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Start the Server.

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

3. Start the .

a. Log in to the Admin Console.
b. Select Clusters .
c. Select Actions Start .

Step 3: Upgrading the Cloudera Manager Server

You can also use the procedures on this page to install Cloudera Manager patches. You must obtain the download URL for the patch before proceeding – you will use this information later in this procedure.

Important: Upgrades to Cloudera Manager 7.0.3 are not supported.

Note: The embedded PostgreSQL version within Key Trustee Server is upgraded from 14.2 to 14.16 in Cloudera Runtime 7.1.9 SP1 CHF10. If you are using KTS, you need to upgrade your Cloudera Manager to 7.11.3 CHF17 first in order to upgrade to Cloudera Runtime 7.1.9 SP1 CHF10. Upgrading from CDH6 to a Cloudera Runtime 7.1.9 SP1 CHF is supported with Cloudera Manager 7.11.3 CHF11, and not supported with Cloudera Manager 7.11.3 CHF12 and later versions.

Note: Upgrades from CDH 6.x are supported only for upgrades to Cloudera Manager 7.4.4 or higher and Cloudera Runtime 7.1.7 or higher. Upgrades from CDH 6.0 are not supported. For a full list of supported upgrades, see Upgade Paths.

Warning: Upgrading to Cloudera Manager version 7.7.1 and above introduces a new service called CORE_SETTINGS-2 which leads to stale configurations on different services in the cluster.

This topic provides procedures for backing up the Cloudera Manager Server.

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

After you complete the steps in Step 1: Getting Started Upgrading Cloudera Manager 6 on page 67 and Step 2: Backing Up Cloudera Manager 6 on page 71, continue with the following:

Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported

Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher

Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.

Warning: Before upgrading CDH 5 clusters with Sentry to Cloudera Runtime 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.

Important: If HDFS ACL sync is enabled (hdfs_sentry_sync_enable=true) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see Installing Ranger RMS.

Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Important: Please note the following:

- A valid Enterprise license file and a username and password are required to download and install the software. You can obtain the username and password from the Cloudera CDH Download page. See Your license file must be current and uploaded to .

  To upload a license:

  1. Download the license file and save it locally.
  2. In , go to the Home page.
  3. Select Administration License .
  4. Click Upload License.
  5. Browse to the license file you downloaded.
  6. Click Upload.
- If you are using Express, you cannot upgrade or CDH.
- Several steps in the procedures have changed and now require the username and password.
- Download URLs have changed.

Important: If you encounter problems, see the following:

- Troubleshooting a Cloudera Manager Upgrade on page 104
- Reverting a Failed Cloudera Manager Upgrade on page 104

Establish Access to the Software

needs access to a package repository that contains the updated software packages. You can choose to access the public repositories directly, or you can download those repositories and set up a local repository to access them from within your network. If your cluster hosts do not have connectivity to the Internet, you must set up a local repository.

If you have enabled high availability for , perform the following steps on the hosts for both the active and passive instances of the server.

1. Log in to the Server host.

   ```
   ssh MY_CLOUDERA_MANAGER_SERVER_HOST
   ```

2. Remove any older files in the existing repository directory:
   RHEL / CentOS

   ```
   sudo rm /etc/yum.repos.d/cloudera*manager.repo*
   ```

   SLES

   ```
   sudo rm /etc/zypp/repos.d/cloudera*manager.repo*
   ```

   Ubuntu

   ```
   sudo rm /etc/apt/sources.list.d/cloudera*.list*
   ```

3. Create a repository file so that the package manager can locate and download the binaries.

   Note: If you are upgrading to a patch, substitute the download URL for the patch when creating the repository file (cloudera-manager.repo).

   Do one of the following, depending on whether or not you are using a local package repository:

   - Use a local package repository (Required when cluster hosts do not have access to the internet.) See Configuring a Local Package repository.
   - Use the public repository

   RHEL / CentOS

   a. Create a file named /etc/yum.repos.d/cloudera-manager.repo with the following content:

   ```
   [cloudera-manager]
   name=
   baseurl=https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
    VERSION>/redhat<OS MAJOR VERSION>/yum/
   gpgkey =https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
    VERSION>/redhat<OS MAJOR VERSION>/yum/RPM-GPG-KEY-cloudera
   username=CHANGEME
   password=CHANGEME
   gpgcheck=1
   enabled=1
   autorefresh=0
   type=rpm-md
   ```

   Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the /etc/yum.repos.d/clouder a-manager.repo file.

   SLES

   a. Create a file named /etc/zypp/repos.d/cloudera-manager.repo with the following content:

   ```
   [cloudera-manager]
   name=
   baseurl=https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
    VERSION>/sles<OS MAJOR VERSION>/yum/
   gpgkey =https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
    VERSION>/sles<OS MAJOR VERSION>/yum/RPM-GPG-KEY-cloudera
   username=CHANGEME
   password=CHANGEME
   gpgcheck=1
   enabled=1
   autorefresh=0
   type=rpm-md
   ```

   b. Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the /etc/zypp/repos.d/cloude ra-manager.repo file.

   Ubuntu

   Debian is not a supported operating system for 6.x.

   a. Create a file named /etc/apt/sources.list.d/cloudera_manager.list with the following content:

   ```
   #  <CLOUDERA MANAGER VERSION>
   deb [arch=amd64]
    http://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/<CLOUDERA
    MANAGER VERSION>/ubuntu1804/apt -cm<CLOUDERA MANAGER VERSION>
    contrib
   ```

b. Run the following command:

```
sudo apt-get update
```

c. Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the /etc/apt/sources.list.d/cloudera_manager.list file.

Tip: If you have a mixed operating system environment, adjust the Operating System filter at the top of the page for each operating system. The guide will generate the repo file for you automatically here.

4. A upgrade can introduce new package dependencies. Your organization may have restrictions or require prior approval for the installation of new packages. You can determine which packages may be installed or upgraded:
RHEL / CentOS

```
yum deplist cloudera-manager-agent
```

SLES

```
zypper info --requires cloudera-manager-agent
```

Ubuntu

```
apt-cache depends cloudera-manager-agent
```

Install Java (JDK)

Oracle JDK 1.8 is required on all cluster hosts managed by Cloudera Manager 6.0.0 or higher. If it is supported for your version of Cloudera Manager, you can also install OpenJDK 8*, OpenJDK 11*, or OpenJDK 17*. See Manually Installing OpenJDK. If OpenJDK 8* is already installed on your hosts, skip the steps in this section.

If you are upgrading to Cloudera Manager 6.0.0 or higher, you can manually install JDK 8 on the Cloudera Manager server host, and then, as part of the Cloudera Manager upgrade process, you can specify that Cloudera Manager upgrade the JDK on the remaining hosts.

A supported JDK is required on all hosts. During a Cloudera Manager upgrade, you can install OpenJDK 8* on the Cloudera Manager server host, and then Cloudera Manager can install the new JDK on the managed hosts. You can also choose to install Oracle JDK 8, OpenJDK 8*, or OpenJDK 11* manually, on all hosts before beginning the upgrade.

Note: Cloudera Manager no longer installs Oracle JDKs.

A supported JDK is required on all hosts. During a Cloudera Manager upgrade, you can install OpenJDK 8* on the Cloudera Manager server host, and then Cloudera Manager can install the new JDK on the managed hosts. You can also choose to install Oracle JDK 8, OpenJDK 8*, OpenJDK 11*, or OpenJDK 17* manually, on all hosts before beginning the upgrade.

Note: Cloudera Manager no longer installs Oracle JDKs.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

* Azul OpenJDK, OpenJDK 8, OpenJDK 11, and OpenJDK 17 are TCK certified for CDP.

2. Stop the Server.

```
sudo systemctl stop cloudera-scm-server
```

3. Remove the JDK:

   a. Perform the following steps on all hosts managed by Cloudera Manager:

      1. Run the following command to remove the JDK, using the package names from Step 1: (If you do not delete these files, Cloudera Manager and other components may continue to use the old version of the JDK.)
         RHEL

         ```
         yum remove <JDK PACKAGE NAME>
         ```

         Ubuntu

         ```
         apt-get remove <JDK PACKAGE NAME>
         ```

         SLES

         ```
         zypper rm <JDK PACKAGE NAME>
         ```

      2. Confirm that the package has been removed:
         RHEL

         ```
         yum list installed |grep -i java
         ```

         Ubuntu

         ```
         apt list --installed | grep -i java
         ```

         SLES

         ```
         zypper search --installed-only |grep -i java
         ```

4. Install OpenJDK
   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*

   ```
   zypper install java-11-openjdk-devel
   ```

   OpenJDK 17*

   ```
   sudo zypper --non-interactive install java-17-openjdk-devel
   ```

5. Start the Server.

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

Upgrade the Cloudera Manager Server

1. Log in to the Cloudera Manager server host.
2. If your cluster is running the embedded PostgreSQL database, stop all services that are using the embedded database. These can include:
   - Hive service and all services such as Impala and Hue that use the Hive metastore
   - Oozie
   - Sentry
   - Sqoop

3. Stop the Cloudera Management Service.
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Stop .

   Important: Not stopping the Cloudera Management Service at this point might cause management roles to crash or the Cloudera Manager Server might fail to restart.

4. Ensure that you have disabled any scheduled replication or snapshot jobs and wait for any running commands from the Cloudera Manager Admin Console to complete before proceeding with the upgrade.

   Important: If there are replication jobs, snapshot jobs, or other commands running when you stop Cloudera Manager Server, Cloudera Manager Server might fail to start after the upgrade.

5. If you have any Hive Replication Schedules that replicate to a cloud destination, delete these replication clusters before continuing with the upgrade. You can re-create these Replication Schedules after the Cloudera Manager upgrade is complete.
6. If your cluster is running Ubuntu version 18, stop all clusters before upgrading Cloudera Manager. (For each cluster, go to *CLUSTER NAME*ActionsStop.)
7. Stop the Cloudera Manager server host, agent, and embedded database (if installed). If you have enabled high availability for Cloudera Manager, perform the following steps on the hosts for both the active and passive instances of Cloudera Manager:
   a. Log in to the Cloudera Manager Server host:

   ```
   ssh MY_CLOUDERA_MANAGER_SERVER_HOST
   ```

   b. Stop the Server.

   ```
   sudo systemctl stop cloudera-scm-server
   ```

   c. If you are using the embedded PostgreSQL database, stop the Embedded PostgreSQL database:

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

   If you are not using the embedded PostgreSQL database and you attempt to stop it, you might see a message indicating that the service cannot be found.

   In case of using the embedded PostgreSQL, if you see a message that the shutdown failed, then the embedded database is still running, probably because services are connected to the Hive metastore. If the database shutdown fails due to connected services, issue the following command:

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

   d. Stop the Agent.

   ```
   sudo systemctl stop cloudera-scm-agent
   ```

8. Upgrade the Cloudera Manager software. If you have enabled high availability for Cloudera Manager, perform the following steps on the hosts for both the active and passive instances of the Cloudera Manager server.
   a. Upgrade the packages. Include the cloudera-manager-server-db-2 package in the command only if you are using the embedded PostgreSQL database.
   RHEL / CentOS

   ```
   sudo yum clean all
   ```

```
sudo yum upgrade cloudera-manager-server cloudera-manager-daemons
 cloudera-manager-agent cloudera-manager-server-db-2
```

SLES

```
sudo zypper clean --all
sudo zypper up cloudera-manager-server cloudera-manager-daemons
cloudera-manager-agent cloudera-manager-server-db-2
```

Ubuntu

```
sudo apt-get clean
```

```
sudo apt-get update
```

```
sudo apt-get dist-upgrade
```

```
sudo apt-get install cloudera-manager-server cloudera-manager-da
emons cloudera-manager-agent cloudera-manager-server-db-2
```

You might be prompted about your configuration file version:

```
Configuration file '/etc/cloudera-scm-agent/config.ini'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ? Your options are:
Y or I : install the package maintainer's version
N or O : keep your currently-installed version
D : show the differences between the versions
Z : start a shell to examine the situation
The default action is to keep your current version.
```

You may receive a similar prompt for /etc/cloudera-scm-server/db.properties. Answer N to both prompts.

You may be prompted to accept the GPG key. Answer y.

```
Retrieving key from https://archive.cloudera.com/.../cm/RPM-GPG-KEY-clo
udera
Importing GPG key ...
 Userid     : "Yum Maintainer <webmaster@cloudera.com>"
 Fingerprint: ...
 From       : https://archive.cloudera.com/.../RPM-GPG-KEY-cloudera
```

Note: If you receive the following error message when running these commands: [Errno 14] HTTP Error 404 - Not Found, make sure the URL in the cloudera-manage.list cloudera-manager.repo file is correct and is reachable from the server host.

b. If you customized the /etc/cloudera-scm-agent/config.ini file, your customized file is renamed with the extension .rpmsave or .dpkg-old. Merge any customizations into the /etc/cloudera-scm-agent/config.ini file that is installed by the package manager.

c. Verify that you have the correct packages installed.

Ubuntu

```
dpkg-query -l 'cloudera-manager-*'
```

```
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aW
ait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                  Version               Description
+++-=====================-=====================-===============
=========================================
ii  cloudera-manager-agent 5.15.0-0.cm...~sq The  Agent
ii  cloudera-manager-daemo 5.15.0-0.cm...~sq Provides daemons for
 monitoring Hadoop and related tools.
ii  cloudera-manager-serve 5.15.0-0.cm...~sq The  Server
```

RHEL / CentOS / SLES

```
rpm -qa 'cloudera-manager-*'
```

```
cloudera-manager-server-5.15.0-...
cloudera-manager-agent-5.15.0-...
cloudera-manager-daemons-5.15.0-...
cloudera-manager-server-db-2-5.15.0-...
```

d. If you are using the embedded PostgreSQL database, start the database:

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

9. Start the Cloudera Manager agent and server. If you have enabled high availability for Cloudera Manager, start only the host for the active instance of Cloudera Manager:

a. Start the Agent.

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

If the agent starts without errors, no response displays.

b. The Cloudera Manager server now requires 4GB of heap. On the Cloudera Manager server host, edit the /etc/default/cloudera-scm-server file and change the line that begins with export CMF_JAVA_OPTS=. Change the -Xmx2G parameter to -Xmx4G.

c. Start the Server.

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

d. If you have problems starting the server or the agent, such as database permissions problems, you can use log files to troubleshoot the problem:

Server log:

```
tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log
```

Agent log:

```
tail -f /var/log/cloudera-scm-agent/cloudera-scm-agent.log
```

or

```
tail -f /var/log/messages
```

10. Verify that all cluster hosts appear in the Cloudera Manager Admin Console.
    a. Use a Web browser to open the Cloudera Manager Admin Console using the following URL:

    ```
    http://CLOUDERA_MANAGER_SERVER_HOSTNAME:7180/cmf/upgrade
    ```

    It can take several minutes for the Cloudera Manager Server to start, and the Cloudera Manager Admin Console is unavailable until the server startup is complete and the Upgrade Cloudera Manager page displays.
    b. Verify that all cluster hosts are visible. (Go to HostsAll Hosts.)
11. If you have enabled high availability for Cloudera Manager, start the host for the passive instance of Cloudera Manager:
    a. Start the Agent.

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

    If the agent starts without errors, no response displays.
    b. The Cloudera Manager server now requires 4GB of heap. On the Cloudera Manager server host, edit the /etc/default/cloudera-scm-server file and change the line that begins with export CMF_JAVA_OPTS=. Change the -Xmx2G parameter to -Xmx4G.
    c. Start the Server.

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

    d. If you have problems starting the server or the agent, such as database permissions problems, you can use log files to troubleshoot the problem:

    Server log:

    ```
    tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log
    ```

    Agent log:

    ```
    tail -f /var/log/cloudera-scm-agent/cloudera-scm-agent.log
    ```

    or

    ```
    tail -f /var/log/messages
    ```

To complete the Cloudera Manager upgrade, continue with Step 4: Upgrading the Cloudera Manager Agents on page 87.

Step 4: Upgrading the Cloudera Manager Agents

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

Important: Upgrades to Cloudera Manager 7.0.3 are not supported.

Note: The embedded PostgreSQL version within Key Trustee Server is upgraded from 14.2 to 14.16 in Cloudera Runtime 7.1.9 SP1 CHF10. If you are using KTS, you need to upgrade your Cloudera Manager to 7.11.3 CHF17 first in order to upgrade to Cloudera Runtime 7.1.9 SP1 CHF10. Upgrading from CDH6 to a Cloudera Runtime 7.1.9 SP1 CHF is supported with Cloudera Manager 7.11.3 CHF11, and not supported with Cloudera Manager 7.11.3 CHF12 and later versions.

Important: If you encounter problems, see the following:

- Troubleshooting a Cloudera Manager Upgrade on page 104

Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported

Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher

Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.

Warning: Before upgrading CDH 5 clusters with Sentry to Cloudera Runtime 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.

Important: If HDFS ACL sync is enabled (hdfs_sentry_sync_enable=true) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see Installing Ranger RMS.

Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Upgrade the Cloudera Manager Agents (Cloudera Manager 7.0.3 and higher)

1. Ensure that the ptrace_scope operating system control is set to 0:

   The Cloudera Manager Agent installation process uses a re-parenting mechanism to ensure that running Cloudera Services are not impacted by the Cloudera Manager Agent upgrade. This re-parenting mechanism utilizes the Linux kernel's ptrace capability. If the ptrace_scope system control is set to a non-zero value, then the installer will not be able to re-parent running Cloudera services. The Cloudera Manager Agent RPM will refuse to install if the ptrace_scope control has a non-zero value. For more information, see https://www.kernel.org/doc/Documentation/security/Yama.txt

   - Verify that the value of ptrace_scope is set to zero. Run the following command to check the value:

     ```
     cat /proc/sys/kernel/yama/ptrace_scope
     ```

   - If the value is not set to 0, set the value to 0 by running the following command on all cluster hosts:

     ```
     echo 0 >> /proc/sys/kernel/yama/ptrace_scope
     ```

     If you do not want to allow ptrace_scope to run, run the following command on all cluster hosts to force the Cloudera Manager Agent upgrade:

     ```
     touch /tmp/CLOUDERA_SKIP_PTRACE_CHECK_ON_UPGRADES
     ```

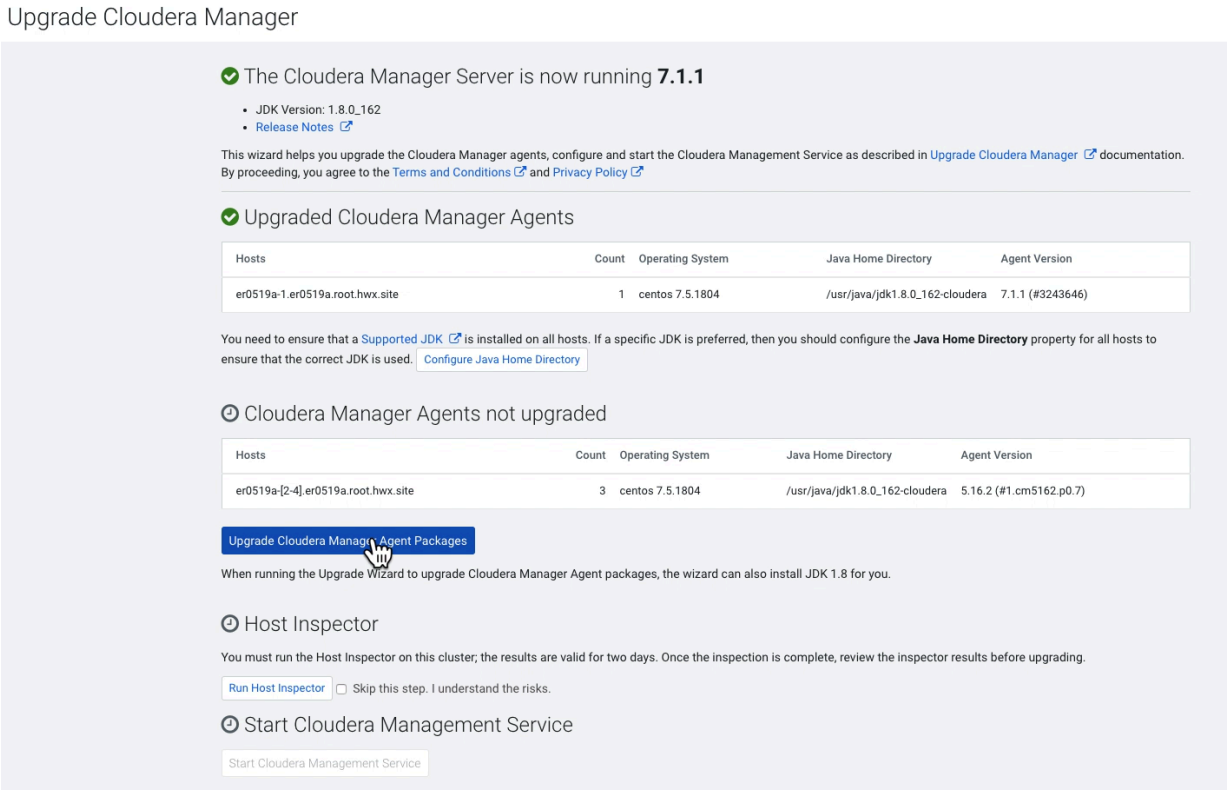
   - If necessary, you can set ptrace_scope to its original value after the agent upgrades are complete.

     Note: Ubuntu 18 and Ubuntu 20 set ptrace_scope to a non-zero value by default.

2. After upgrading and starting the Cloudera Manager server, open the Cloudera Manager Admin Console (if you have not already done so) using the following URL:

```
https://CLOUDERA_MANAGER_SERVER_HOSTNAME:7183/cmf/upgrade
```

The Upgrade Cloudera Manager screen displays:



3. Click Upgrade Cloudera Manager Agent packages

The Upgrade Cloudera Manager Agent Packages page displays the Select Repository step.

4. Select one of the following:
   - Select Public Cloudera Repository if the Cloudera Manager server host has access to the internet.
   - Select the Custom Repository If you are using a local package repository instead of the public repository at https://archive.cloudera.com, option and enter the Custom Repository URL.
5. Click Continue.
6. The Select JDK screen displays the available options for the JDK used in the cluster. Choose one of the following options to install a JDK:
   - Manually Manage JDK – Select this option if you have already installed a supported JDK. For information on installing a JDK, see Upgrading the JDK on page 44.
   - Install a Cloudera-provided version of OpenJDK – Cloudera Manager installs OpenJDK 8 on all your cluster hosts, except for the Cloudera Manager server host(s).
   - Install a system-provided version of OpenJDK – Cloudera Manager installs the default version of OpenJDK provided by the host operating system.
7. Click Continue.

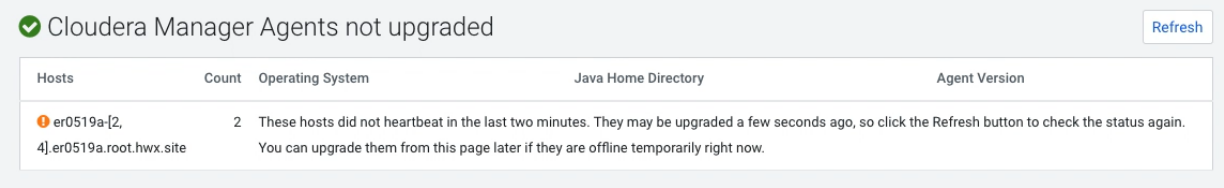
The Enter Login Credentials page displays.

8. Specify the credentials and initiate Agent installation:
   a. Select root for the root account, or select Another user and enter the username for an account that has password-less sudo permission.
   b. Select an authentication method:
      - If you choose the All hosts accept same password option, enter and confirm the password.
      - If you choose the All hosts accept same private key option, provide a passphrase and path to the required key files.
   c. Modify the default SSH port if necessary.
   d. Specify the maximum Number of Simultaneous Installations to run at once. The default and recommended value is 10. Adjust this parameter based on your network capacity.
9. Click Continue.

   The Cloudera Manager Agent packages and, if selected, the JDK are installed.

10. When the installations are complete, click Finish.

    The Upgrade Cloudera Manager page displays the status of the upgrade. If you see a message listing Cloudera Manager Agents not upgraded, wait a few minutes for the agents to heartbeat and the click the Refresh button.



    If some hosts do not report a heartbeat, you must upgrade the Cloudera Manager Agents manually. Do the following on these hosts:

    a. Ensure that the hosts have access to the package repositories. See Configure a Repository for Cloudera Manager.
    b. Run the following commands on all affected hosts to remove the cloudera-manager-agent, cloudera-manager-daemons, and cloudera-manager-server packages and install the agent and daemons. (Omit cloudera-manager-server on the Cloudera Manager server host):

Operating System	Command

RHEL	`sudo yum remove cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server`
	
	`sudo yum clean all`
	`sudo yum install cloudera-manager-agent cloudera-manager-daemons`

SLES	`sudo zypper remove cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server`
	
	`sudo zypper refresh -s`
	`sudo zypper install cloudera-manager-agent cloudera-manager-daemons`

Ubuntu or Debian	`sudo apt-get purge cloudera-manager-daemons cloudera-manager-agent cloudera-manager-server`
	
	`sudo apt-get update`
	`sudo apt-get install cloudera-manager-agent cloudera-manager-daemons`

    c. Copy the agent's config.ini file from the backup taken before upgrading to /etc/cloudera-scm-agent/config.ini. (See Back Up Cloudera Manager Agent on page 73)
    d. Restart the agents:

```
sudo systemctl stop cloudera-scm-supervisord.service
sudo systemctl start cloudera-scm-agent
```

11. After the Agents are all upgraded, Click Run Host Inspector to run the host inspector. Inspect the output and correct any warnings. If problems occur, you can make changes and then rerun the inspector.
12. When you are satisfied with the inspection results, click Start the Cloudera Management Service.
13. Confirm that you want to start the Cloudera Management Service by clicking Continue.

14. After the Cloudera Management Service has started, click Finish.

    You will see a message indicating that the Cloudera Management Service has started.

    The upgrade is now complete.
15. Click the Home Page link to return to the Home page. Review and fix any critical configuration issues. You may need to restart any clusters if they indicate stale configurations.

    To return to the Upgrade Cloudera Manager page, go to HostsAll HostsReview Upgrade Status.
16. If you stopped any clusters before upgrading Cloudera Manager, start them now. (For each cluster, go to *CLUSTER NAME*ActionsStart.)
17. If you set ptrace_scope to 0 and want to use the original or a different value, you can reset it by running the following command on all hosts:

    ```
    echo [NEW_VALUE] >> /proc/sys/kernel/yama/ptrace_scope
    ```
18. If you have the Embedded Container Service (ECS) deployed in any clusters, do the following
    a. Restart the ECS Cluster. Go to the ECS cluster, click the actions menu and select Restart.
    b. Unseal the Vault. Go to the ECS service and click Actions Unseal.

Step 5: After You Upgrade Cloudera Manager

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported

Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher

Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.

Warning: Before upgrading CDH 5 clusters with Sentry to Cloudera Runtime 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.

Important: If HDFS ACL sync is enabled (hdfs_sentry_sync_enable=true) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see Installing Ranger RMS.

Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Perform Post-Upgrade Steps

1. If you upgraded the JDK, do the following:
   a. If the Cloudera Manager Server host is also running a Cloudera Manager Agent, restart the Cloudera Manager Server:
   b. Restart the Server.

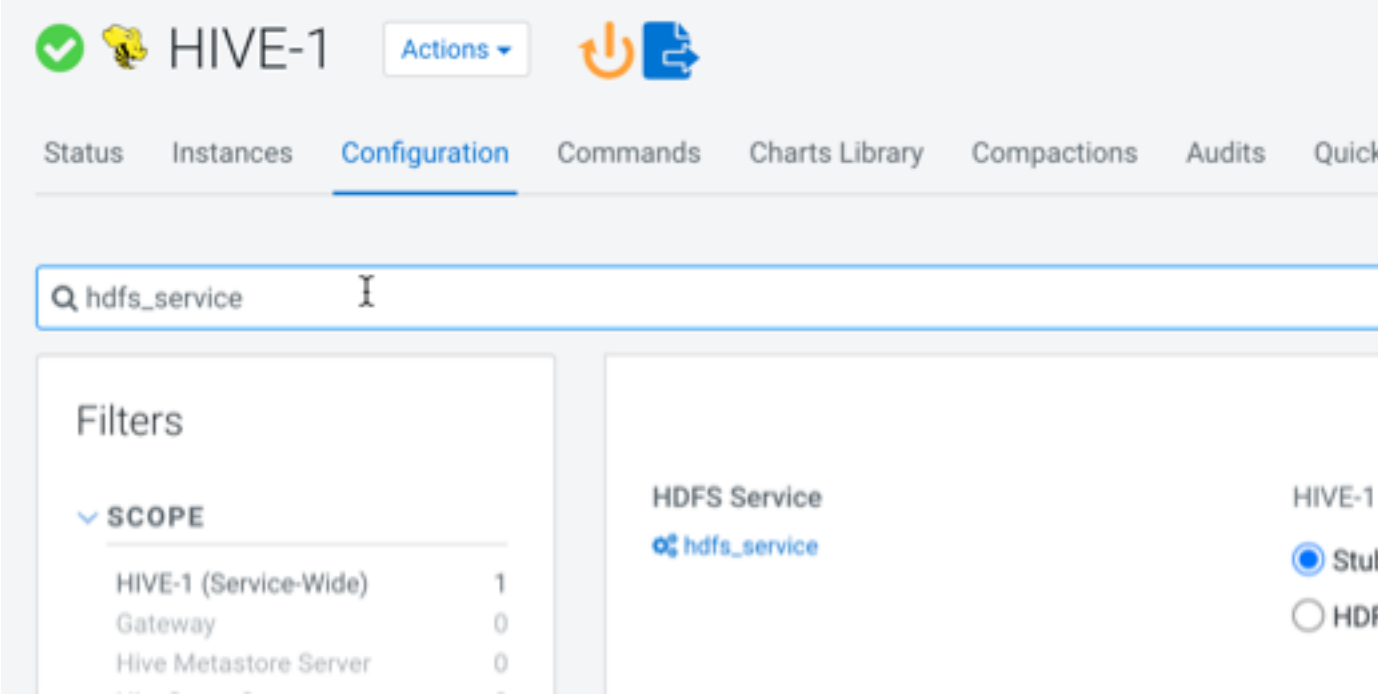
      ```
      sudo systemctl restart cloudera-scm-server
      ```

   c. Open the Cloudera Manager Admin Console and set the Java Home Directory property in the host configuration:
      1. Go to HomeAll HostsConfiguration.
      2. Set the value to the path to the new JDK.
      3. Click Save Changes.
   d. Restart all services:
      1. On the HomeStatus tab, click next to the cluster name, select Restart and confirm.
2. Start the Cloudera Management Service and adjust any configurations when prompted.
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Start .

3. Add and configure the Stub DFS service and reset the dependent services:
   a. Add the Stub DFS service to each cluster managed by Cloudera Manager that contains the services returned by the query you ran when upgrading the Cloudera Manager server. (Step 4, on the Upgrade the Cloudera Manager Server on page 83 page.)
      1. In the Cloudera Manager Admin Console, go to the status page the cluster.
      2. Click the Actions menu and select Add Service.
      3. Select the Stub DFS service.
      4. Click Continue.

         The Assign Roles page displays.
      5. Accept the role assignments, or change them if necessary.
      6. Click Continue.

         Cloudera Manager adds the Stub DFS service.
   b. In the Cloudera Manager Admin Console, for each of these services, go to the Configuration tab and set the configuration property hdfs_service to Stub DFS. For example:



4. If your deployment uses LDAP, you may see that its health test has a Disabled status, you can configure an LDAP Bind Distinguished Name and password to enable the health test.

   In the Cloudera Manager Admin Console, go to AdministrationSettingsExternal Authentication and set the following parameters:

   - LDAP Bind Distinguished Name for Monitoring
   - LDAP Bind Password for Monitoring

   If these parameters are left blank, Cloudera Manager attempts to use the bind credentials specified for authentication.
5. If you have deployed Apache Atlas in the cluster, restart the Apache Atlas service.
6. If you have deployed Kafka in the cluster, perform a rolling restart the Kafka service.

7. If Cloudera Manager reports stale configurations after the upgrade, you might need to restart the cluster services and redeploy the client configurations. If any managed cluster includes the Hive and YARN components, this is required. If you will also be upgrading CDH, this step is not required.

   Stale configurations can occur after a Cloudera Manager upgrade when a default configuration value has changed, which is often required to fix a serious problem. Configuration changes that result in Cloudera Manager reporting stale configurations are described the Cloudera Manager 7.7.1 release notes:
8. If you are using Streams Messaging Manager, you need to configure database related configuration properties.

   Note: If you are also upgrading your distribution to Runtime, you can choose to skip database configuration and complete it during the upgrade to Runtime.

   a. Select the Streams Messaging Manager service.
   b. Go to Configuration.
   c. Find and configure the following properties:
      - Streams Messaging Manager Database User Password
      - Streams Messaging Manager Database Type
      - Streams Messaging Manager Database Name
      - Streams Messaging Manager Database User
      - Streams Messaging Manager Database Host
      - Streams Messaging Manager Database Port
   d. Click Save Changes.
9. If you are using Schema Registry, you need to configure database related configuration properties.

   Note: If you are also upgrading your distribution to Runtime, you can choose to skip database configuration now and complete it during the upgrade to Runtime.

   a. Select the Schema Registry service.
   b. Go to Configuration.
   c. Find and configure the following properties:
      - Schema Registry Database User Password
      - Schema Registry Database Type
      - Schema Registry Database Name
      - Schema Registry Database User
      - Schema Registry Database Host
      - Schema Registry Database Port
   d. Click Save Changes.
10. On the HomeStatus tab, click next to the cluster name, select Restart and confirm.
11. On the HomeStatus tab, click next to the cluster name, select Deploy Client Configuration and confirm.
12. If you disabled any backup or snapshot jobs before the upgrade, now is a good time to re-enable them
13. If you deleted any Hive Replication schedules before the Cloudera Manager upgrade, re-create them now.
14. Check if you have set allow_weak_crypto to true in /etc/krb5.confduring JDK upgrade. If yes, proceed to the last step to complete the Cloudera Manager Upgrade, else proceed to the next step to set the kerberos encryption type.

15. If you have upgraded to Cloudera Manager 7.11.3, and are using JDK 11 or higher, and Cloudera Manager is managing the krb5.conf (when Cloudera Manager Administration Settings Manage krb5.conf through Cloudera Manager is enabled) then,
    a. Stop the cluster.
    b. Set the value of Cloudera Manager Administration Settings Kerberos Encryption Types to any 'AES'. Cloudera recommends setting it to *AES256-CTS*.
    c. If MIT is in use, make sure the KDC version is greater than 1.18.2 to support any aes256 encryption type and:
        - Edit the kdc.conf and add aes-256 to supported_enctypes.
        - Restart KDC.
    d. If, and only if, Active Directory (KDC) server is in use, then set Cloudera Manager Administration Settings Active Directory Set Encryption Types to 'True'.
    e. Complete the Deploy Kerberos Client Configuration wizard OR call the CM API deployClusterClientConfig.
    f. Regenerate the credentials. Go to Cloudera Manager Administration Security Kerberos Credentials tab, select all and click Regenerate Selected .

    Note: If the KDC is AD, then Regenerate Credentials can fail due to infrastructure issues. One common reason could be the replication delay between AD servers.

    Note: You can alternatively use the CM API to regenerate the credentials i.e. use deleteCredentials and then call generateCredentials API call. These API calls must be made once the cluster has been stopped.
    g. Start the cluster.
16. The Cloudera Manager upgrade is now complete. If Cloudera Manager is not working correctly, or the upgrade did not complete, see Troubleshooting a Cloudera Manager Upgrade on page 104.

Upgrade Key Trustee Server to 7.1.x

How to upgrade Key Trustee Server to CDP Private Cloud 7.1.x.

About this task

Running “Upgrading a Cluster” will not upgrade the Key Trustee Server. KTS is not part of the overall CDH parcel and must be upgraded separately. Upgrading the KTS can be done at any point after performing Upgrading Cloudera Manager.

If you are upgrading from CDH 5.13 through CDH 5.16, you must first upgrade the CDH Key Trustee Server to 5.15 before upgrading to the CDP 7.1.x Key Trustee Server.

If you are upgrading from CDH 6.1 through CDH 6.3, you must first upgrade the CDH Key Trustee Server to 6.1 before upgrading to the CDP 7.1.x Key Trustee Server.

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.

Important: If you are upgrading from CDH 6.x to CDP 7.1.9 (KTS with Postgres 9.x to Postgres 14.16) , you must:

1. First, upgrade from CDH 6.x to CDP 7.1.x.
2. Upgrade KTS cluster to 7.1.x.
3. Restart Active KTS instance followed by Passive KTS instance.
4. Upgrade CDP 7.1.x to 7.1.9.
5. Upgrade KTS cluster to 7.1.9.
6. Restart Active KTS instance followed by Passive KTS instance.

If you are using a package-based KTS install, see “Migrating Unmanaged Key Trustee Server to Cloudera Manager”.

Procedure

1. Back up the Key Trustee Server:
   a) Select the Key Trustee Server service configuration that you wish to back up.
   b) From the Actions menu, select Create Backup on Active Server (or Create Backup on Passive Server) . A successfully completed backup of the Key Trustee Server is indicated by the message Command Create Backup on Active Server finished successfully on service keytrustee_server.
2. Add your internal parcel repository to Cloudera Manager following the instructions in “Configuring a Local Parcel Repository” (see “Configuring Cloudera Manager to Use an Internal Remote Parcel Repository ”).
3. Download, distribute, and activate the latest Key Trustee Server parcel on the cluster containing the Key Trustee Server host, following the instructions in “Access Parcels”.

Important: The KEYTRUSTEE parcel in Cloudera Manager is not the Key Trustee Server parcel; it is the Key Trustee KMS parcel. The parcel name for Key Trustee Server is KEYTRUSTEE_SERVER

Note: Do not accept the prompt from CM to perform a rolling restart. Instead, restart the KTS services manually, beginning with active instance followed by the passive instance(s).

What to do next

If the Key Trustee Server active or passive database does not start properly after upgrade from 5.x, 6.x or 7.0 to 7.1, manually restart the Key Trustee Server service to correct the problem: Key Trustee Server service Actions Restart .

Related Information

Upgrading a Cluster
Upgrading Cloudera Manager
Upgrading Cloudera Navigator Key Trustee Server
Migrating Unmanaged Key Trustee Server to Cloudera Manager
Configuring a Local Parcel Repository
Access Parcels

Upgrade Navigator Encrypt to 7.1.x

How to upgrade Navigator Encrypt from CDH to CDP Private Cloud 7.1.x.

About this task

Running “Upgrading a Cluster” will not upgrade Navigator Encrypt, because Navigator Encrypt is not part of the overall CDH parcel and needs to be upgraded separately. Upgrading NavEncrypt can be done at any point after performing “Upgrading Cloudera Manager”.

You can upgrade from Navigator Encrypt 3.16-6.2 to 7.1.x. If you are upgrading from an older version of Navigator Encrypt, first upgrade to 3.16+ using “Upgrading Cloudera Navigator Encrypt”.

Upgrading Navigator Encrypt (RHEL/Centos/Oracle)

1. Back up the /etc/navencrypt directory before upgrading.

   If you have problems accessing encrypted data after upgrading the OS or kernel, restore /etc/navencrypt from your backup and try again.
2. Add your internal package repository to Cloudera Manager following the instructions in “Configuring a Local Package Repository”.

3. Install the Cloudera Repository:
   a. Add the internal repository you created by following the instructions in “Configuring a Local Package Repository” (see “Configuring Hosts to Use the Internal Repository”.)
   b. Import the GPG key by running the following command:

```
sudo rpm --import HTTPS://ARCHIVE.CLOUDERA.COM/P/NAVENCRYPT7/7.1.9.0/REDHAT9/YUM/RPM-GPG-KEY-CLOUDERA
```

4. Stop Navigator Encrypt:

```
sudo systemctl stop navencrypt-mount
```

5. Upgrade Navigator Encrypt client:

```
sudo yum update navencrypt
```

6. Start Navigator Encrypt:

```
sudo systemctl start navencrypt-mount
```

7. If using an RSA master key file, then you should change the master key to use OAEP padding:

```
# navencrypt key --change --rsa-oaep
...
>> Choose NEW MASTER key type:
 1) Passphrase (single)
 2) Passphrase (dual)
 3) RSA private key
Select: 3
Type MASTER RSA key file:
Type MASTER RSA key passphrase:
```

   To check the type of padding currently in use:

```
# navencrypt key --get-rsa-padding
Type your Master key
Type MASTER RSA key file:
Type MASTER RSA key passphrase:

Verifying Master Key against keytrustee (wait a moment)...
RSA_PKCS1_OAEP_PADDING
```

Upgrading Navigator Encrypt (SLES)

1. Back up the /etc/navencrypt directory before upgrading.

   If you have problems accessing encrypted data after upgrading the OS or kernel, restore /etc/navencrypt from your backup and try again.
2. Add your internal package repository to Cloudera Manager following the instructions in “Configuring a Local Package Repository”.
3. Install the Cloudera Repository:
   a. Add the internal repository you created by following the instructions in “Configuring a Local Package Repository” (see “Configuring Hosts to Use the Internal Repository”.)
   b. Import the GPG key by running the following command:

```
sudo rpm --import HTTPS://ARCHIVE.CLOUDERA.COM/P/NAVENCRYPT7/7.1.9.0/REDHAT9/YUM/RPM-GPG-KEY-CLOUDERA
```

4. Stop Navigator Encrypt:

```
sudo service navencrypt-mount stop
```

5. Upgrade the Kernel Module Package (KMP):

```
sudo zypper update cloudera-navencryptfs-kmp-KERNEL_FLAVOR
```

Replace *KERNEL_FLAVOR* with the kernel flavor for your system. Navigator Encrypt supports the default, xen, and ec2 kernel flavors.

6. Upgrade Navigator Encrypt client:

```
sudo zypper update navencrypt
```

7. Enable Unsupported Modules:

Edit /etc/modprobe.d/unsupported-modules and set allow_unsupported_modules to 1. For example:

```
#
# Every kernel module has a flag 'supported'. If this flag is not set load
ing
# this module will taint your kernel. You will not get much help with a
kernel
# problem if your kernel is marked as tainted. In this case you firstly h
ave
# to avoid loading of unsupported modules.
#
# Setting allow_unsupported_modules 1 enables loading of unsupported mo
dules
# by modprobe, setting allow_unsupported_modules 0 disables it. This can
# be overridden using the --allow-unsupported-modules command line switch.
allow_unsupported_modules 1
```

8. Start Navigator Encrypt:

```
sudo service navencrypt-mount start
```

9. If using an RSA master key file, then you should change the master key to use OAEP padding:

```
# navencrypt key --change --rsa-oaep
...
>> Choose NEW MASTER key type:
 1) Passphrase (single)
 2) Passphrase (dual)
 3) RSA private key
Select: 3
Type MASTER RSA key file:
Type MASTER RSA key passphrase:
```

To check the type of padding currently in use:

```
# navencrypt key --get-rsa-padding
Type your Master key
Type MASTER RSA key file:
Type MASTER RSA key passphrase:

Verifying Master Key against keytrustee (wait a moment)...
RSA_PKCS1_OAEP_PADDING
```

Upgrading Navigator Encrypt (Ubuntu)

1. Back up the /etc/navencrypt directory before upgrading.

   If you have problems accessing encrypted data after upgrading the OS or kernel, restore /etc/navencrypt from your backup and try again.
2. Add your internal package repository to Cloudera Manager following the instructions in “Configuring a Local Package Repository”.
3. Install the Cloudera Repository:
   a. Add the internal repository you created by following the instructions in “Configuring a Local Package Repository” (see “Configuring Hosts to Use the Internal Repository”.)
   b. Run:

   ```
   echo "deb http://repo.example.com/path/to/ubuntu/stable $DISTRIB_CODENAM
   E main" | sudo tee -a /etc/apt/sources.list
   ```

   c. Import the GPG key by running the following command:

   ```
   wget -O - https://archive.cloudera.com/p/navencrypt7/7.1.9.0/redhat9/yum
   /RPM-GPG-KEY-cloudera | apt-key add -
   ```

   d. Update the repository index:

   ```
   apt-get update
   ```
4. Stop Navigator Encrypt:

   ```
   sudo service navencrypt-mount stop
   ```
5. Upgrade Navigator Encrypt client:

   ```
   sudo apt-get install navencrypt
   ```
6. Start Navigator Encrypt:

   ```
   sudo service navencrypt-mount start
   ```
7. If using an RSA master key file, then you should change the master key to use OAEP padding:

   ```
   # navencrypt key --change --rsa-oaep
   ...
   >> Choose NEW MASTER key type:
    1) Passphrase (single)
    2) Passphrase (dual)
    3) RSA private key
   Select: 3
   Type MASTER RSA key file:
   Type MASTER RSA key passphrase:
   ```

   To check the type of padding currently in use:

   ```
   # navencrypt key --get-rsa-padding
   Type your Master key
   Type MASTER RSA key file:
   Type MASTER RSA key passphrase:

   Verifying Master Key against keytrustee (wait a moment)...
   RSA_PKCS1_OAEP_PADDING
   ```

Related Information

Upgrading Cloudera Navigator Encrypt

Configuring a Local Package Repository

Upgrading Cloudera Navigator Key HSM

Learn how to upgrade Navigator Key HSM.

Setting Up an Internal Repository

Although it is possible to upgrade Cloudera Navigator KeyHSM by using the KeyHSM RPM package directly, Cloudera recommends setting up a YUM package repository to perform the upgrade.

The steps given below assume that a repository containing the KeyHSM RPM package downloaded from the paywall has been created. For more information on creating such a repository, see https://wiki.centos.org/HowTos/CreateLocalRepos.

Upgrading Key HSM (Minor and Patch Version Upgrades)

If you are upgrading from Key HSM 1.x (shipped with CDH 5.x and earlier) to Key HSM 7.x, use the instructions in Upgrading Key HSM (Major Version Upgrades) on page 101; do not use the procedure documented in this section.

Important: If you have implemented Key Trustee Server high availability, upgrade Key HSM on each Key Trustee Server.

1. Install the KeyHSM Repository

   Add the internal repository that you created.
2. Stop the Key HSM Service

   Stop the Key HSM service before upgrading:

   ```
   sudo service keyhsm shutdown
   ```
3. Upgrade Navigator Key HSM

   Upgrade the Navigator Key HSM package using yum:

   ```
   sudo yum update keytrustee-keyhsm
   ```

   Cloudera Navigator Key HSM is installed to the /usr/share/keytrustee-server-keyhsm directory by default.
4. Start the Key HSM Service

   Start the Key HSM service:

   ```
   sudo service keyhsm start
   ```

Upgrading Key HSM (Major Version Upgrades)

Important: Only use this procedure if you are upgrading from Key HSM 1.x (shipped with CDH 5.x and earlier) to Key HSM 7.x. There is a unique configuration issue that impacts this upgrade scenario, and the steps here are different from those required for all minor Key HSM upgrades. This procedure is not applicable to minor version or patch release upgrades.

1. Install the KeyHSM Repository

   Add the internal repository that you created.
2. Stop the Key HSM Service

   Stop the Key HSM service before upgrading:

   ```
   sudo service keyhsm shutdown
   ```

3. Upgrade Navigator Key HSM

   Upgrade the Navigator Key HSM package using yum:

   ```
   sudo yum update keytrustee-keyhsm
   ```

   Cloudera Navigator Key HSM is installed to the /usr/share/keytrustee-server-keyhsm directory by default.
4. Rename Configuration Files that were created earlier

   For Key HSM major version upgrades, previously-created configuration files do not authenticate with the HSM and Key Trustee Server, so you must recreate these files by re-executing the setup and trust commands. First, navigate to the Key HSM installation directory and rename the applications.properties, keystore, and truststore files:

   ```
   cd /usr/share/keytrustee-server-keyhsm/
   mv application.properties application.properties.bak
   mv keystore keystore.bak
   mv truststore truststore.bak
   ```
5. Initialize Key HSM

   Run the service keyhsm setup command in conjunction with the name of the target HSM distribution:

   ```
   sudo service keyhsm setup [keysecure|thales|luna]
   ```

   For more details, see Initializing Navigator Key HSM.
6. Establish Trust Between Key HSM and the Key Trustee Server

   The Key HSM service must explicitly trust the Key Trustee Server certificate (presented during TLS handshake). To establish this trust, run the following command:

   ```
   sudo keyhsm trust /path/to/key_trustee_server/cert
   ```

   For more details, see Integrating Key HSM with Key Trustee Server.
7. Start the Key HSM Service

   Start the Key HSM service:

   ```
   sudo service keyhsm start
   ```
8. Establish Trust Between Key Trustee Server and Key HSM

   Establish trust between the Key Trustee Server and the Key HSM by specifying the path to the private key and certificate:

   ```
   sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 \
   --client-certfile /etc/pki/cloudera/certs/mycert.crt \
   --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
   ```

   For a password-protected Key Trustee Server private key, add the --passphrase argument to the command (enter the password when prompted):

   ```
   sudo ktadmin keyhsm --passphrase \
   --server https://keyhsm01.example.com:9090 \
   --client-certfile /etc/pki/cloudera/certs/mycert.crt \
   --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
   ```

   For additional details, see Integrating Key HSM with Key Trustee Server.

9. Remove Configuration Files From Previous Installation

   After completing the upgrade, remove the saved configuration files from the previous installation:

```
cd /usr/share/keytrustee-server-keyhsm/
rm application.properties.bak
rm keystore.bak
rm truststore.bak
```

Key Trustee Server SSL Certificate Regeneration

When Key HSM is upgraded to Cloudera version 7.1.4 or above, the SSL certificates of the Key Trustee Server (KTS) might need to be regenerated if the self-signed certificates that are created by the `ktadmin` command are being used.

Perform the following steps to regenerate the KTS SSL certificate:

1. Stop the KTS service from Cloudera Manager.
2. Navigate to the location /var/lib/keytrustee/.keytrustee/.ssl/ to take a backup of the certificate files ssl-cert-keytrustee-pk.pem and ssl-cert-keytrustee.pem:

```
cd /var/lib/keytrustee/.keytrustee/.ssl/
```

3. Backup the certificate files:

```
mv ssl-cert-keytrustee-pk.pem ssl-cert-keytrustee-pk_backup.pem
mv ssl-cert-keytrustee.pem ssl-cert-keytrustee_backup.pem
```

4. Regenerate the certificate file:

```
ktadmin init
```

5. Configure the Key HSM to trust the new certificate file:

```
keyhsm trust /var/lib/keytrustee/.keytrustee/.ssl/ssl-cert-keytrustee.pem
```

6. Restart the Key HSM service.
7. Start the KTS service from Cloudera Manager.
8. Run the following command to test and validate certificate regeneration:

```
curl -vk https://$(hostname -f):11371/test_hsm
```

Upgrading Key Trustee KMS

Upgrading from CDH to CDP automatically replaces your existing Key Trustee KMS service with Ranger KMS. Your Key Trustee KMS ACLs are converted to Ranger policies.

When you upgrade from CDH 5 or 6 to CDP 7.1.1+, the Cloudera Manager removes your existing Key Trustee KMS installation and installs Ranger KMS. Next, Key Trustee KMS ACLs are converted to Ranger policies. Some ACLs are not supported and will be ignored by Ranger KMS.

This process is entirely automatic. There are links below to help you learn about your new KMS service: how the ACL conversion process works, which ACLs are not supported, and how Ranger KMS policy evaluation works.

Related Information

Migrating ACLs from Key Trustee KMS to Ranger KMS

Installing Ranger KMS backed with a Key Trustee Server and HA

Setting Up Data at Rest Encryption for HDFS

Configuring a database for Ranger or Ranger KMS

Troubleshooting a Cloudera Manager Upgrade

The Cloudera Manager Server fails to start after upgrade.

The Cloudera Manager Server fails to start after upgrade.

Possible Reasons

There were active commands running before upgrade. This includes commands a user might have run and also commands Cloudera Manager automatically triggers, either in response to a state change, or something configured to run on a schedule, such as Backup and Disaster Recovery replication or snapshot jobs.

Possible Solutions

- Stop any running commands from the Cloudera Manager Admin Console or wait for them to complete. See Aborting a Pending Command.
- Ensure that you have disabled any scheduled replication or snapshot jobs from the Cloudera Manager Admin Console to complete before proceeding with the upgrade. See HDFS Replication.

Re-Running the Cloudera Manager Upgrade Wizard

Minimum Required Role: Full Administrator. This feature is not available when using Cloudera Manager to manage Data Hub clusters.

The first time you log in to the Cloudera Manager server after upgrading your Cloudera Manager software, the upgrade wizard runs. If you did not complete the wizard at that time, or if you had hosts that were unavailable at that time and still need to be upgraded, you can re-run the upgrade wizard:

1. Click the Hosts tab.
2. Click Re-run Upgrade Wizard or Review Upgrade Status. This takes you back through the installation wizard to upgrade Cloudera Manager Agents on your hosts as necessary.
3. Select the release of the Cloudera Manager Agent to install. Normally, this is the Matched Release for this Cloudera Manager Server. However, if you used a custom repository (instead of archive.cloudera.com) for the Cloudera Manager server, select Custom Repository and provide the required information. The custom repository allows you to use an alternative location, but that location must contain the matched Agent version.
4. Specify credentials and initiate Agent installation:
   a. Select root for the root account, or select Another user and enter the username for an account that has password-less sudo privileges.
   b. Select an authentication method:
      - If you choose password authentication, enter and confirm the password.
      - If you choose public-key authentication, provide a passphrase and path to the required key files.

      You can modify the default SSH port if necessary.
   c. 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.
   d. Click Continue.

When you click Continue, the Cloudera Manager Agent is upgraded on all the currently managed hosts. You cannot search for new hosts through this process. To add hosts to your cluster, click the Add New Hosts to Cluster button.

Reverting a Failed Cloudera Manager Upgrade

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

This topic describes how to reinstall the same version of Cloudera Manager you were using previously, so that the version of your Cloudera Manager Agents match the server. The steps below assume that the Cloudera Manager Server is already stopped (because it failed to start after the attempted upgrade).

Important:

The following instructions assume that a Cloudera Manager upgrade failed, and that the upgraded server never started, so that the remaining steps of the upgrade process were not performed. The steps below are not sufficient to revert from a running Cloudera Manager deployment.

To revert a running Cloudera Manager Server and Cloudera Manager Agents after a successful upgrade, see Step 19 of Rolling back a CDP Private Cloud Base Upgrade from version 7.1.8 to CDH 6 or Rolling back a CDP Private Cloud Base Upgrade from versions 7.1.1 - 7.1.7 to CDH 6 which has instructions for reverting the Cloudera Manager packages on all hosts.

Ensure Cloudera Manager Server and Agent are stopped.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Stop the Server.

```
sudo systemctl stop cloudera-scm-server
```

3. Stop the Agent.

```
sudo systemctl stop cloudera-scm-agent
```

Restore the Cloudera Manager Database (if necessary)

If your Cloudera Manager upgrade fails, you need to determine whether the upgrade process has successfully completed updating the schema of the Cloudera Manager database. If the schema update has begun, you must restore the Cloudera Manager database using a backup taken before you began the upgrade.

1. To determine whether the schema has been updated, examine the Cloudera Manager server logs, and look for a message similar to the following: Updated Schema Version to 60000. (The version number may be different for your environment.)

   Run the following command to find the log entry (f the log file is in a different location, substitute the correct path):

```
grep 'Updated Schema Version to ' /var/log/cloudera-scm-server/cloudera-scm-server.log
```

2. If required, restore the database.

   The procedure for restoring the database depends on the type of database used by Cloudera Manager.

3. If you are using the embedded PostgreSQL database, stop the Embedded PostgreSQL database:

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

   If you are not using the embedded PostgreSQL database and you attempt to stop it, you might see a message indicating that the service cannot be found.

   In case of using the embedded PostgreSQL, if you see a message that the shutdown failed, then the embedded database is still running, probably because services are connected to the Hive metastore. If the database shutdown fails due to connected services, issue the following command:

```
sudo service cloudera-scm-server-db next_stop_fast
```

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

Establish Access to the Software

needs access to a package repository that contains the updated software packages. You can choose to access the public repositories directly, or you can download those repositories and set up a local repository to access them from within your network. If your cluster hosts do not have connectivity to the Internet, you must set up a local repository.

If you have enabled high availability for , perform the following steps on the hosts for both the active and passive instances of the server.

1. Log in to the Server host.

   ```
   ssh MY_CLOUDERA_MANAGER_SERVER_HOST
   ```

2. Remove any older files in the existing repository directory:
   RHEL / CentOS

   ```
   sudo rm /etc/yum.repos.d/cloudera*manager.repo*
   ```

   SLES

   ```
   sudo rm /etc/zypp/repos.d/cloudera*manager.repo*
   ```

   Ubuntu

   ```
   sudo rm /etc/apt/sources.list.d/cloudera*.list*
   ```

3. Create a repository file so that the package manager can locate and download the binaries.

   Note: If you are upgrading to a patch, substitute the download URL for the patch when creating the repository file (cloudera-manager.repo).

   Do one of the following, depending on whether or not you are using a local package repository:

   - Use a local package repository (Required when cluster hosts do not have access to the internet.) See Configuring a Local Package repository.
   - Use the public repository

     RHEL / CentOS

     a. Create a file named /etc/yum.repos.d/cloudera-manager.repo with the following content:

     ```
     [cloudera-manager]
     name=
     baseurl=https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
      VERSION>/redhat<OS MAJOR VERSION>/yum/
     gpgkey =https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER
      VERSION>/redhat<OS MAJOR VERSION>/yum/RPM-GPG-KEY-cloudera
     username=CHANGEME
     password=CHANGEME
     gpgcheck=1
     enabled=1
     autorefresh=0
     ```

```
type=rpm-md
```

Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the /etc/yum.repos.d/cloudera-manager.repo file.

SLES

a. Create a file named /etc/zypp/repos.d/cloudera-manager.repo with the following content:

```
[cloudera-manager]
name=
baseurl=https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER VERSION>/sles<OS MAJOR VERSION>/yum/
gpgkey =https://archive.cloudera.com/p/cm7/<CLOUDERA MANAGER VERSION>/sles<OS MAJOR VERSION>/yum/RPM-GPG-KEY-cloudera
username=CHANGEME
password=CHANGEME
gpgcheck=1
enabled=1
autorefresh=0
type=rpm-md
```

b. Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the /etc/zypp/repos.d/cloudera-manager.repo file.

Ubuntu

Debian is not a supported operating system for 6.x.

a. Create a file named /etc/apt/sources.list.d/cloudera_manager.list with the following content:

```
#  <CLOUDERA MANAGER VERSION>
deb [arch=amd64]
 http://USERNAME:PASSWORD@archive.cloudera.com/p/cm7/<CLOUDERA MANAGER VERSION>/ubuntu1804/apt -cm<CLOUDERA MANAGER VERSION>
 contrib
```

b. Run the following command:

```
sudo apt-get update
```

c. Replace *CHANGEME* with your *USERNAME* and *PASSWORD* in the /etc/apt/sources.list.d/cloudera_manager.list file.

Tip: If you have a mixed operating system environment, adjust the Operating System filter at the top of the page for each operating system. The guide will generate the repo file for you automatically here.

4. A upgrade can introduce new package dependencies. Your organization may have restrictions or require prior approval for the installation of new packages. You can determine which packages may be installed or upgraded:
RHEL / CentOS

```
yum deplist cloudera-manager-agent
```

SLES

```
zypper info --requires cloudera-manager-agent
```

Ubuntu

```
apt-cache depends cloudera-manager-agent
```

Downgrade the Cloudera Manager Packages

Note: Make sure the repository file above matches the specific maintenance version before the upgrade.

1. Downgrade the packages. Note: Only add cloudera-manager-server-db-2 if you are using the embedded PostgreSQL database.
   RHEL / CentOS

   ```
   sudo yum clean all
   sudo yum repolist
   ```

   ```
   sudo yum downgrade "cloudera-manager-*"
   ```

   SLES

   ```
   sudo zypper clean --all
   ```

   ```
   sudo zypper dup -r BASEURL
   ```

   Ubuntu

   There is no action that downgrades to the version currently in the repository.
2. Verify that you have the correct packages installed.

   Ubuntu

   ```
   dpkg-query -l 'cloudera-manager-*'
   ```

   ```
   Desired=Unknown/Install/Remove/Purge/Hold
   | Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aW
   ait/Trig-pend
   |/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
   ||/ Name                  Version               Description
   +++-=====================-=====================-===============
   ==========================================
   ii  cloudera-manager-agent 5.15.0-0.cm...~sq The  Agent
   ii  cloudera-manager-daemo 5.15.0-0.cm...~sq Provides daemons for
    monitoring Hadoop and related tools.
   ii  cloudera-manager-serve 5.15.0-0.cm...~sq The  Server
   ```

   RHEL / CentOS / SLES

   ```
   rpm -qa 'cloudera-manager-*'
   ```

   ```
   cloudera-manager-server-5.15.0-...
   cloudera-manager-agent-5.15.0-...
   cloudera-manager-daemons-5.15.0-...
   cloudera-manager-server-db-2-5.15.0-...
   ```

Restore the Cloudera Manager Directory

1. Run the following commands to extract the backups:

   ```
   cd $CM_BACKUP_DIR
   tar -xf cloudera-scm-agent.tar
   tar -xf cloudera-scm-server.tar
   ```

2. Restore the Cloudera Manager server directory from a backup taken during the upgrade process:

```
sudo -E cp -rp $CM_BACKUP_DIR/etc/cloudera-scm-server/* /etc/cloudera-scm-server
```

```
sudo -E cp -rp $CM_BACKUP_DIR/etc/default/cloudera-scm-server /etc/default/cloudera-scm-server
```

3. If the Cloudera Manager server host has an agent installed, restore the Cloudera Manager agent directory from a backup taken during the upgrade process:

```
sudo -E cp -rp $CM_BACKUP_DIR/etc/cloudera-scm-agent/* /etc/cloudera-scm-agent
```

```
sudo -E cp -rp $CM_BACKUP_DIR/etc/default/cloudera-scm-agent /etc/default/cloudera-scm-agent
```

```
sudo -E cp -rp $CM_BACKUP_DIR/var/run/cloudera-scm-agent/* /var/run/cloudera-scm-agent
```

```
sudo -E cp -rp $CM_BACKUP_DIR/var/lib/cloudera-scm-agent/* /var/lib/cloudera-scm-agent
```

Start Cloudera Manager Again

1. If you are using the embedded PostgreSQL database, start the database:

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

2. Start the Agent.

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

If the agent starts without errors, no response displays.

3. Start the Server.

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

4. Start the .
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Start .

Note: Troubleshooting: If you have problems starting the server, such as database permissions problems, you can use the server's log to troubleshoot the problem.

```
vim /var/log/cloudera-scm-server/cloudera-scm-server.log
```

Validate TLS configurations

If you are upgrading to from CDH to Cloudera Manager 7.4.4 or earlier, you need to validate TLS configurations. By validating TLS, you can avoid upgrade failure by properly configuring Cloudera Manager properties if your clusters are TLS-enabled.

About this task

Cloudera Manager does not configure the following properties for Hive metastore (HMS):

- HDFS NameNode TLS/SSL Trust Store File (namenode_truststore_file)
- HDFS NameNode TLS/SSL Trust Store Password (namenode_truststore_password)
- Hive Metastore TLS/SSL Trust Store File (hive.metastore.dbaccess.ssl.truststore.path)
- Hive Metastore TLS/SSL Trust Store Password (hive.metastore.dbaccess.ssl.truststore.password)

These configurations are required. Ranger Plugin needs to validate the TLS connection to Ranger to download policies. The Hive Strict Managed Migration (HSMM) reports success, but actually fails. The HSMM log appears as follows:

```
HiveStrictManagedMigration: [main]: Found 0 databases
HiveStrictManagedMigration: [main]: Done processing databases
```

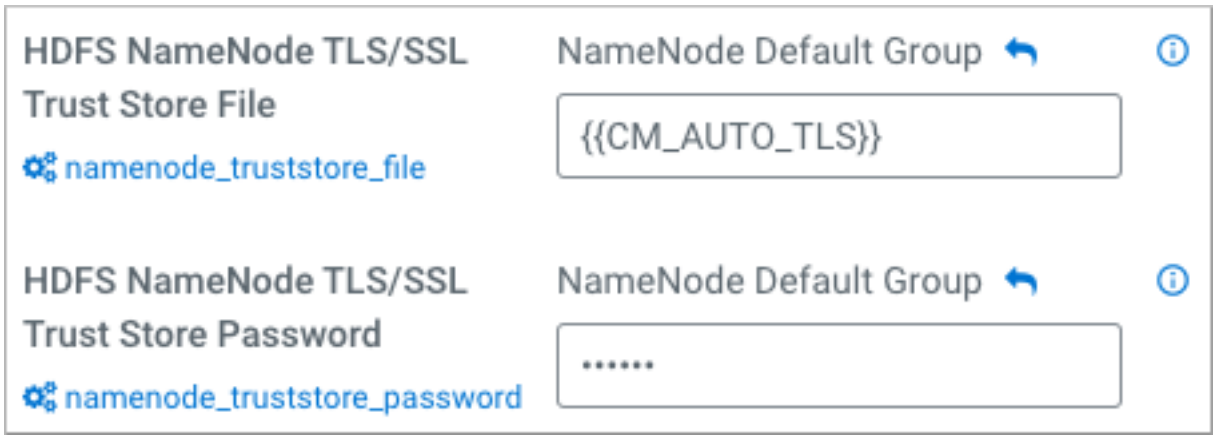
You must configure the HDFS and HMS truststore file and password properties after upgrading Cloudera Manager to 7.3.1 or later.

Before you begin

- You completed the upgrade to Cloudera Manager 7.4.4 or earlier.
- Your CDP cluster will be TLS-enabled.

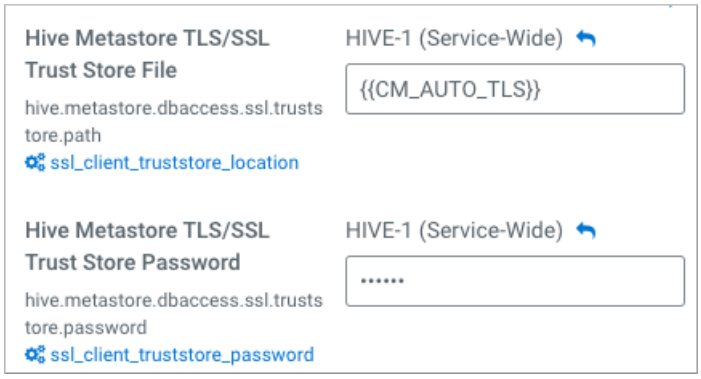
Procedure

1. In Cloudera Manager, to configure HDFS properties click Clusters HDFS-1 Configuration .
2. Search for namenode_truststore.
3. Set HDFS NameNode TLS/SSL Trust Store File to {{CM_AUTO_TLS}}.



4. Set HDFS TLS/SSL Trust Store Password.
5. In Cloudera Manager, to configure Hive Metastore properties click Clusters Hive-1 Configuration .
6. Search for hive.metastore.dbaccess.

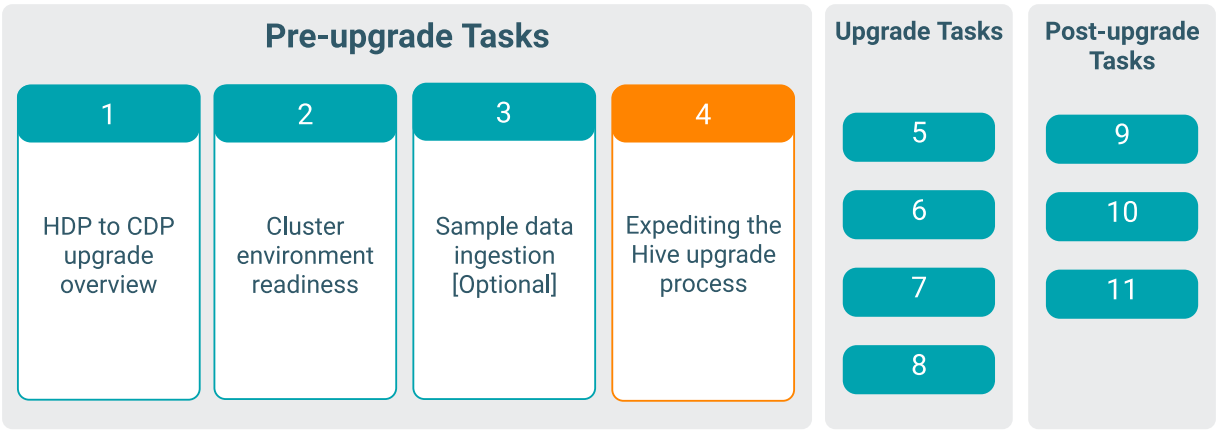
7. Set Hive Metastore TLS/SSL Trust Store File to {{CM_AUTO_TLS}}.



8. Set Hive Metastore TLS/SSL Trust Store Password.
9. Save changes.

Expediting the Hive upgrade

Preparing the Hive metastore for the upgrade can take a long time. Checking and correcting your Hive metastore partitions and SERDE definitions is critical for a successful upgrade. If you have many tables and partitions, it might be difficult to manually identify these problems. The free Hive Upgrade Check tool helps identify these problems.



The Hive Upgrade Check tool is Community software that scans your Hive metastore to identify potential upgrade problems. You can also use the tool to perform the following tasks:

- Convert legacy managed tables (non-acid) to external tables.
- Report potential problems, such as tables that do not have matching HDFS directories, to resolve before the upgrade.

The cluster upgrade to CDP runs the Hive Strict Managed Migration (HSMM) process that performs the same tasks. During the cluster upgrade, you can skip the HSMM process, migrating none of your tables and database definitions.

Overview of the expedited Hive upgrade

You perform tasks before and after the Hive migration to hasten the upgrade. The sequence of steps involved in expediting the Hive upgrade includes identifying problems in tables and databases before upgrading, configuring the Hive Strict Managed Migration (HSMM) to prevent migration, and completing the upgrade. After the upgrade to CDP, you migrate the tables and databases.

1. Prepare tables for migration, identifying potential migration problems using the Hive Upgrade Check tool.
2. Decide to expedite the upgrade by not migrating your databases and tables during the upgrade.
3. Upgrade Cloudera Manager, and then start to upgrade your cluster.
4. In the upgrade wizard, after adding the Hive-on-Tez service, temporarily leave the upgrade wizard.
5. Configure HSMM to prevent migration of your databases and tables.
6. Return to the upgrade wizard and continue upgrading your cluster.

   None of your databases or tables are migrated to CDP.

If you did not migrate your Hive data, do so after the upgrade to CDP as follows:

1. Prepare tables for migration, identifying and fixing potential migration problems using the Hive Upgrade Check tool.
2. Create a list of databases and tables to migrate.
3. Migrate tables and databases to CDP.

   You cannot use unmigrated tables in CDP.

Preparing tables for migration

You download the Hive Upgrade Check tool and use it to identify problems in unmigrated tables. These problems can cause upgrade failure. It saves time to fix the problems and avoid failure. The tool provides help for fixing those problems before migrating the tables to CDP.

About this task

You use the Hive Upgrade Check community tool to help you identify tables that have problems affecting migration. You resolve problems revealed by the Hive Upgrade Check tool to clean up the Hive Metastore before migration. If you do not want to use the Hive Upgrade Check tool, you need to perform the tasks described in the following subtopics to migrate Hive data to CDP:

- Check SERDE Definitions and Availability
- Handle Missing Table or Partition Locations
- Manage Table Location Mapping
- Make Tables SparkSQL Compatible

Procedure

1. Obtain the Hive Upgrade Check tool.

   Download the Hive SRE Upgrade Check tool from the Cloudera labs github location.
2. Follow instructions in the github readme to run the tool.
   The Hive Upgrade Check (v.2.3.5.6+) will create a yaml file (hsmm_<name>.yaml) identifying databases and tables that require attention.
3. Follow instructions in prompts from the Hive Upgrade Check tool to resolve problems with the tables.

   At a minimum, you must run the following processes described in the github readme:

   - process ID 1 Table / Partition Location Scan - Missing Directories
   - process id 3 Hive 3 Upgrade Checks - Managed Non-ACID to ACID Table Migrations

Check SERDE Definitions and Availability

Ensure correct Serde definitions and a reference to a SERDE exists to ensure a successful upgrade.

About this task

You perform this step if you do not modify the HSMM process for expediting the Hive upgrade.

Procedure

1. Check Serde definitions for correctness and check for SERDE availability.

2. Correct any problems found as follows:
   - Remove the table having the problematic SERDE.
   - Ensure the SERDE is available during the upgrade, so the table can be evaluated.

Handle Missing Table or Partition Locations

You need to identify missing table or partition locations, or both, to prevent upgrade failure. If the table and partition locations do not exist in the file system, you must either create a replacement partition directory (recommended) or drop the table and partition.

About this task

You perform this step if you did not modify the HSMM process to expedite the Hive upgrade.

Procedure

Ensure the table and partition locations exist on the file system. If these locations don't exist either create a replacement partition directory (recommended) or drop the table and partition.

Managed Table Location Mapping

A managed table location must map to one managed table only. If multiple managed tables point to the same location, upgrade problems occur.

Make Tables SparkSQL Compatible

Non-Acid, managed tables in ORC or in a Hive Native (but non-ORC) format that are owned by the POSIX user hive will not be SparkSQL-compatible after the upgrade unless you perform manual conversions.

About this task

If your table is a managed, non-ACID table, you can convert it to an external table using this procedure (recommended). After the upgrade, you can easily convert the external table to an ACID table, and then use the Hive Warehouse Connector to access the ACID table from Spark.

Take one of the following actions.

- Convert the tables to external Hive tables before the upgrade.

  ALTER TABLE ... SET TBLPROPERTIES('EXTERNAL'='TRUE','external.table.purge'='true')
- Change the POSIX ownership to an owner other than hive.

  You will need to convert managed, ACID v1 tables to external tables after the upgrade.

Configuring HSMM to prevent migration

You need to know how to configure the Hive Strict Managed Migration (HSMM) to prevent migrating your tables and databases as you run the upgrade process in Cloudera Manager. You briefly leave the upgrade process, do the configuration, and than proceed with the upgrade.

Before you begin

- You are in the middle of the CDH to CDP Private Cloud Base in-place upgrade and installed Hive-on-Tez.

About this task

In this task, you set the table migration control file URL property to an arbitrary value, deliberately causing HSMM to fail to migrate your tables and databases. You must manually migrate these tables later.

Procedure

1. In Cloudera Manager, go to Clusters Hive-on-Tez .
2. Stop the Hive-on-Tez service.

3. In Configuration , search for table migration control file URL.
4. Set the value of the Table migration control file URL property to the absolute path and file name of your YAML include list.
5. Save configuration changes.
6. Start the Hive-on-Tez service.
7. Return to the CDH to CDP Private Cloud Base in-place upgrade wizard to complete the cluster upgrade.

Related Information

Running the Hive Upgrade Check tool

Understanding the Hive upgrade

You need an understanding of the Hive Strict Managed Migration (HSMM) and the Hive Upgrade Check tool for a successful upgrade.

HSMM vs the Hive Upgrade Check tool

It is difficult to estimate how long the Hive Strict Managed Migration will take. The following factors are just a few that might affect how long it takes:

- Number of managed tables
- Core processing power
- Backend metastore database speed

The process runs across all Hive metastore databases and tables by default, identifying managed tables that need to undergo compaction or conversion to Hive 3 ACID V2 tables.

Consider expediting the upgrade process if one of the following conditions exist:

- You have few, or no, ACID tables but do have many legacy managed tables in your environment.
- Reducing downtime is critical, and justifies the extra effort to expedite the upgrade process.

Why upgrading takes so long

The underlying Hive upgrade process Hive Strict Managed Migration (HSMM) is an Apache Hive conversion utility that makes adjustments to Hive tables under the enhanced and strict Hive 3 environment to meet the needs of the most demanding workloads and governance requirements for Data Lake implementations. There are some changes to the standard behaviors in Hive table definitions and locations. The HSMM reviews every database and table to determine if changes are needed to meet these requirements.

With systems that have been around for a while, or have adopted some ingest patterns, there may be artifacts in the metastore that cannot be reconciled, including the following artifacts:

- Tables and partitions without reciprocating storage locations
- Tables using SERDEs that have been abandoned.
- ACIDv1 tables

  These tables must be fully compacted before the upgrade. If tables are not compacted, data loss is highly likely.

When these irreconcilable conditions occur, it requires manual intervention to fix problems before it can proceed.

The Hive upgrade process iterates through the databases and tables, attempting to materialize each of them using the Hive Metastore and public Thrift APIs. That creates a heavy load on the underlying metastore database and entire system.

Installing dependencies for Hue

You must install the psycopg2 Python package for PostgreSQL-backed Hue and MySQL clients for MariaDB and MySQL databases depending on your operating systems.

Note: This task is applicable only if you are upgrading to Cloudera Runtime 7.1.8 and higher. If you are upgrading to Cloudera Runtime 7.1.7 or lower, then you can skip this task.

If you are using Oracle as a backend database for Hue, then review Using Oracle database with Hue to ensure that Hue connects to your database.

Installing the psycopg2 Python package for PostgreSQL database

If you are using PostgreSQL as a backend database for Hue on Cloudera Base on premises 7.x.x, 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, OEL 7, CentOS 8, RHEL 8, and 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

The following steps apply to SLES 12 (upgrades to Cloudera Runtime 7.1.8) and SLES 15 SP4 (upgrades to Cloudera Runtime 7.1.9 or higher):

1. SSH into the Hue host as a root user.
2. Install the psycopg2 package dependencies 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

The following steps apply to Ubuntu 18 (upgrades to Cloudera Runtime 7.1.8) and Ubuntu 20 (upgrades to Cloudera Runtime 7.1.9 or higher):

1. SSH into the Hue host as a root user.
2. Install the psycopg2 package dependencies 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 MySQL client for MySQL databases

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.

Attention: The version 2.2.0 of the MySQL client requires that you set the values of the MYSQLCLIENT_CFLAGS and MYSQLCLIENT_LDFLAGS environment variables, as follows:

```
$ export MYSQLCLIENT_CFLAGS=`pkg-config mysqlclient --cflags`
$ export MYSQLCLIENT_LDFLAGS=`pkg-config mysqlclient --libs`
```

The version 2.2.0 of the MySQL client also requires you to install the Python 3 and MySQL development headers and libraries, as follows on Debian or Ubuntu operating systems:

```
sudo apt-get install python3-dev default-libmysqlclient-dev build-es
sential
```

or as follows on CentOS and RHEL operating systems:

```
sudo yum install python3-devel mysql-devel
```

Alternatively, you can install and use the version 2.1.1 of the MySQL client, as follows, which does not have this requirement:

```
pip3 install mysqlclient=2.1.1
```

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
   ```

Installing MySQL client for MariaDB databases

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
```

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
```

Upgrading a CDH 6 Cluster

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

This topic describes how to upgrade a CDH or Cloudera Runtime cluster in any of the following scenarios:

Attention: To upgrade to Cloudera Manager or CDH 5.x or 6.x, do not use the instructions on this page. See the Cloudera Enterprise Upgrade Guide.

When you upgrade a cluster, you use Cloudera Manager to upgrade the cluster software across an entire cluster using Cloudera Parcels. Package-based installations are not supported for Cloudera Runtime and Cloudera Base on premises upgrades. You must transition your CDH clusters to use Parcels before upgrading to Cloudera Base on premises. See Migrating from Packages to Parcels.

Cluster upgrades update the Hadoop software and other components. You can use Cloudera Manager to upgrade a cluster for major, minor, and maintenance upgrades. The procedures vary depending on the version of Cloudera Manager you are using, the version of the cluster you are upgrading, and the version of Cloudera Runtime you are upgrading to.

Ranger plugins use a local copy of policies (client-side caching), so even if Ranger Admin server is not available during the schema update part of the process, upgrades do not impact Ranger authorization. In addition, Ranger Admin is deployed in active/active mode, so restarts do not impact the API interfaces. Ranger Admin HA will work with or without a load balancer. During Ranger Admin service upgrade, policy creates/updates are not available. This does not impact Ranger authorization since Ranger plugins continue to use the local cache.

After completing preparatory steps, you use the Cloudera Manager upgrade wizard to complete the upgrade. Cloudera Manager will restart all services after the upgrade.

- Atlas
- HBase
- HDFS
- Hive-on-Tez
- Hive Metastore
- Ranger
- Hue
- Kafka
- Key Trustee Server
- Knox
- Kudu – see Orchestrating a rolling restart with no downtime.
- MapReduce
- Oozie
- Ranger KMS
- Schema Registry
- YARN
- ZooKeeper

Note: Rolling Upgrades are not supported when upgrading to Cloudera Base on premises.

Warning: If there is any failure during the upgrade process, Cloudera recommends you to contact Cloudera customer support.

Step 1: Getting Started Upgrading a Cluster

Tasks you should perform before starting the upgrade.

Note: Not all combinations of Current Cluster Version to New Cluster Version are supported. Before you upgrade your CDP cluster from one version to another, you must review table 2 to know the supported upgrades paths. For more information, see Supported in-place upgrade paths.

Note: Not all combinations of Cloudera Manager and Cloudera Runtime are supported. Ensure that the version of Cloudera Manager you are using supports the version of Cloudera Runtime you have selected. For details, see Cloudera Manager support for Cloudera Runtime, CDH and CDP Private Cloud Experiences .

Note: Cloudera Private Cloud Data Services version 1.3.4 requires Cloudera Manager 7.5.5 and Cloudera Runtime version 7.1.6 or 7.1.7 For more information, see Cloudera Private Cloud Data Services.

Important: Upgrades to Cloudera Runtime 7.1.7 SP1 (7.1.7.1000) are supported only from Cloudera Runtime 7.1.7.

Note: If you are upgrading to Cloudera Manager 7.5.1 or higher in order to install CDP Private Cloud Experiences version 1.3.1, you must use Cloudera Runtime version 7.1.6 or 7.1.7. For more information, see CDP Private Cloud Exeriences.

Important: Upgrades from CDH 6.1, 6.2, and 6.3 are only supported for upgrades to CDP Private Cloud Base 7.1.7 or higher. Upgrades from CDH 6.0 are not supported.

Important: To upgrade to Cloudera Manager or CDH 5.x or 6.x, do not use the instructions on this page. See the Cloudera Enterprise Upgrade Guide.

Warning: Upgrades to Cloudera Runtime 7.0.3 are not supported.

Note: Upgrades from CDH 6.x are supported only for upgrades to Cloudera Manager 7.4.4 or higher and Cloudera Runtime 7.1.7 or higher. Upgrades from CDH 6.0 are not supported.

Warning: Upgrades from CDH 5.12 and lower to Cloudera Base on premises are not supported. You must upgrade the cluster to CDH versions 5.13 - 5.16 before upgrading to Cloudera Base on premises.

The version of CDH or that you can upgrade to depends on the version of that is managing the cluster. You may need to upgrade before upgrading your clusters. Upgrades are not supported when using 7.0.3.

Before you upgrade a cluster,

- You need to gather information, review the limitations and release notes and run some checks on the cluster. See the Collect Information section below. Fill in the My Environment form below to customize your upgrade procedures.
- You must clean all the HBase Master procedure stores. For more details, see Clean the HBase Master procedure store.

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

Important: If you have any add-on services installed using a CSD (Custom Service Descriptor), you must use Cloudera Manager 7.1.1 or higher to install the CDH 6 version of the CSD before upgrading the cluster to Cloudera Runtime 7.1.1 or higher. During the upgrade, Cloudera Manager will prompt you to also install the Cloudera Runtime 7 version of the CSD. Cloudera Manager version 7.1.4 or higher will prompt you to install any required intermediate versions of the CSD.

To successfully complete the upgrade you must have the CDH 5, CDH 6, and Cloudera Runtime 7 versions of the CSD installed. After the upgrade, you can delete the CDH 5 and CDH 6 versions of the add-on service.

This affects the following Cloudera services: CDSW, Nifi, and Nifi Registry as well as any CSDs created by third parties. See Add-on Services.

Note: Isilon is not supported for Cloudera Base on premises version 7.1.5 and lower.

Note: If your cluster uses Compute clusters in a Virtual Private Cluster (VPC) architecture, you must remove the compute cluster before upgrading the Base cluster. You can recreate the Compute cluster after the upgrade

Note:

After upgrading from CDH to CDP, the NodeManager recovery feature is enabled by default. This means that the yarn.nodemanager.recovery.enabled property is set to true. Cloudera recommends that you keep the NodeManager recovery feature enabled. If you set this property to false in your CDP cluster and then upgrade to a later CDP version, the feature will remain disabled.

Important:

In Cloudera Runtime 7.1.6 and higher, the way Streams Messaging Manager (SMM) integrates with Streams Replication Manager (SRM) has changed. SMM can only connect to and monitor an SRM service that is running in the same cluster as SMM. Monitoring an SRM service that is running in a cluster that is external to SMM is no longer supported.

Connectivity between the two services is disabled by default after a successful upgrade. If you want to continue using SMM to monitor SRM, you must reconnect the two services following the upgrade.

Important:

In Cloudera Runtime 7.1.6 and higher, the way Streams Messaging Manager (SMM) integrates with Streams Replication Manager (SRM) has changed. SMM can only connect to and monitor an SRM service that is running in the same cluster as SMM. Monitoring an SRM service that is running in a cluster that is external to SMM is no longer supported.

Connectivity between the two services is disabled by default after a successful upgrade. If you want to continue using SMM to monitor SRM, you must reconnect the two services following the upgrade.

Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported

Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher

Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.

Warning: Before upgrading CDH 5 clusters with Sentry to Cloudera Runtime 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.

Important: If HDFS ACL sync is enabled (hdfs_sentry_sync_enable=true) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see Installing Ranger RMS.

Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

Collect Information

Collect the following information about your environment and fill in the form above. This information will be remembered by your browser on all pages in this Upgrade Guide.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Run the following command to find the current version of the Operating System:

```
lsb_release -a
```

3. Log in to the Cloudera Manager Admin console and find the following:
   a. The version of Cloudera Manager used in your cluster. Go to Support About .
   b. The version of the JDK deployed in the cluster. Go to Support About .
   c. Whether High Availability is enabled for HDFS.

      If you see a standby namenode instead of a secondary namenode listed under Cloudera Manager HDFS Instances , then High Availability is enabled.
   d. The Install Method and Current cluster version. The cluster version number and Install Method are displayed on the Cloudera Manager Home page, to the right of the cluster name.

Preparing to Upgrade a Cluster

1. You must have SSH access to the Cloudera Manager server hosts and be able to log in using the root account or an account that has password-less sudo permission to all the hosts.
2. Review the Requirements and Supported Versions for the new versions you are upgrading to. See: Cloudera Base on premises requirements and supported versions If your hosts require an operating system upgrade, you must perform the upgrade before upgrading the cluster. See Upgrading the Operating System on page 57.
3. Ensure that a supported version of Java is installed on all hosts in the cluster. See the links above. For installation instructions and recommendations, see Upgrading the JDK on page 44.
4. Review the following documents:

   Cloudera Runtime 7

   - Review the following when upgrading to Cloudera Runtime 7.1 or higher:

     Cloudera Base on premises requirements and supported versions

5. If your deployment has defined a Compute cluster and an associated Data Context, you will need to delete the Compute cluster and Data context before upgrading the base cluster and then recreate the Compute cluster and Data context after the upgrade.

   See Starting, Stopping, Refreshing, and Restarting a Cluster and Virtual Private Clusters and Cloudera SDX.
6. Review the upgrade procedure and reserve a maintenance window with enough time allotted to perform all steps. For production clusters, Cloudera recommends allocating up to a full day maintenance window to perform the upgrade, depending on the number of hosts, the amount of experience you have with Hadoop and Linux, and the particular hardware you are using.
7. If the cluster uses Impala, check your SQL against the newest reserved words listed in incompatible changes. If upgrading across multiple versions, or in case of any problems, check against the full list of Impala reserved words.
8. If the cluster uses Hive, validate the Hive Metastore Schema:
   a. In the Cloudera Manager Admin Console, Go to the Hive service.
   b. Select ActionsValidate Hive Metastore Schema.
   c. Fix any reported errors.
   d. Select ActionsValidate Hive Metastore Schema again to ensure that the schema is now valid.
9. Run the Security Inspector and fix any reported errors.

   Go to Administration Security Security Inspector .
10. Log in to any cluster node as the hdfs user, run the following commands, and correct any reported errors:

```
hdfs fsck / -includeSnapshots -showprogress
```

Note: The fsck command might take 10 minutes or more to complete, depending on the number of files in your cluster.

```
hdfs dfsadmin -report
```

   See HDFS Commands Guide in the Apache Hadoop documentation.
11. Log in to any DataNode as the hbase user, run the following command, and correct any reported errors:

```
hbase hbck
```

12. If the cluster uses Kudu, log in to any cluster host and run the ksck command as the kudu user (sudo -u kudu). If the cluster is Kerberized, first kinit as kudu then run the command:

```
kudu cluster ksck <master_addresses>
```

    For the full syntax of this command, see Checking Cluster Health with ksck.
13. If you are upgrading to CDP 7.1.x or higher, and Hue is deployed in the cluster, and Hue is using PostgreSQL as its database, you must manually install psycopg2. See Installing the psycopg2 Python package for PostgreSQL database on page 115.

Note:

If you are using Oracle database with Hue and are upgrading to CDP 7.x from CDH 5 or CDH 6, then deactivate the Oracle instant client parcel, download and install the Oracle instant client separately, and then connect it to Hue. See Configuring the Hue Server to Store Data in the Oracle database .

14. If your cluster uses Impala and Llama, this role has been deprecated as of CDH 5.9 and you must remove the role from the Impala service before starting the upgrade. If you do not remove this role, the upgrade wizard will halt the upgrade.

    To determine if Impala uses Llama:

    a. Go to the Impala service.
    b. Select the Instances tab.
    c. Examine the list of roles in the Role Type column. If Llama appears, the Impala service is using Llama.

    To remove the Llama role:

    a. Go to the Impala service and select Actions Disable YARN and Impala Integrated Resource Management .

       The Disable YARN and Impala Integrated Resource Management wizard displays.
    b. Click Continue.

       The Disable YARN and Impala Integrated Resource Management Command page displays the progress of the commands to disable the role.
    c. When the commands have completed, click Finish.
15. If your cluster uses the Ozone technical preview, you must stop and delete this service before upgrading the cluster.
16. If your cluster uses Streams Replication Manager and you configured the Log Format property, you must take note of your configuration. The value set for Log Format is cleared during the upgrade and must be manually reconfigured following the upgrade.
17. If your cluster uses Streams Replication Manager and you are affected by OPSAPS-62546, ensure that you apply the workaround detailed in the Known issue. If the workaround is not applied, you might run into issues after the upgrade. For more information, see the SRM Known Issues.
18. If your cluster uses Streams Replication Manager, export or migrate aggregated metrics.

    In Cloudera Runtime 7.1.9, major changes are made to the internal Kafka Streams application of SRM. As a result, SRM by default loses all aggregated metrics that were collected before the upgrade. This means that you will not be able to query metrics with the SRM Service REST API that describe the pre-upgrade state of replications. If you want to retain the metrics, you can either export them, for archival purposes, or migrate them to the new format used by SRM. If you do not need to retain metrics, you can skip this step and continue with the upgrade.

    Exporting metrics creates a backup of the metric data, however, exported metrics cannot be imported into the SRM Service for consumption. As a result, exporting metrics is only useful for data archival purposes.

    Migrating metrics can be done in two different ways depending on whether you are doing a rolling upgrade or a non-rolling upgrade.

    - In case of a non-rolling upgrade, migration happens following the upgrade. In this case, the new version of the internal Kafka Streams application running in the upgraded cluster starts to process historical metrics as soon as it is online. However, until the metrics are processed, the SRM Service cannot serve requests regarding latest metrics and returns empty or missing responses on its REST API. The duration of this downtime depends on the number SRM Service instances and the amount of metrics in the cluster.
    - In case of a rolling upgrade, a migration process called SRM Service Migrator is initiated during the upgrade. The Migrator processes existing metrics so that they become compatible with your upgraded cluster.

Depending on the size of your cluster and the amount of metrics you have, this process may take up to multiple hours to finish.

For Export metrics

Use the following endpoints of the SRM Service REST API to export metrics.

If upgrading from Cloudera Runtime 7.1.8:

- `/v2/topic-metrics/{source}/{target}/{upstreamTopic}/{metric}`
- `/v2/cluster-metrics/{source}/{target}/{metric}`

If upgrading from Cloudera Runtime 7.1.7 or lower:

- `/topic-metrics/{topic}/{metric}`
- `/cluster-metrics/{cluster}/{metric}`

For more information regarding the SRM Service REST API, see Streams Replication Manager Service REST API or Streams Replication Manager REST API Reference.

For Non-rolling upgrade migration

a. In Cloudera Manager, select the SRM service.

b. Go to Configuration.

c. Add the following to the SRM Service Environment Advanced Configuration Snippet (Safety Valve) property:

```
Key: SRM_SERVICE_SKIP_MIGRATION
Value: false
```

For Rolling upgrade migration

Note: All SRM Service role hosts experience increased resource utilization during a rolling upgrade if you enable migration. This is because the migration process uses the same system configurations (for example, heap size) as the SRM Service role.

a. Ensure that the target clusters of the SRM Service are available and healthy.

If a target cluster is unavailable, the upgrade will fail. As a result, if a target cluster is unavailable, or you expect a target cluster to become unavailable during the upgrade, remove it from SRM's configuration for the duration of the upgrade. Metrics in the target clusters that you remove are not migrated. Target clusters are specified in Streams Replication Manager Service Target Cluster.

b. In Cloudera Manager, select the SRM service and go to Configuration.

c. Add the following to the SRM Service Environment Advanced Configuration Snippet (Safety Valve) property:

```
Key: SRM_SERVICE_SKIP_MIGRATION
```

```
Value: false
```

d. Fine-tune the behavior of the migration process.

The SRM Service Metrics Migrator (the migration process) has a number of user configurable properties. Fine tuning the configuration can help in reducing the time it takes to migrate the metrics.

These properties do not have dedicated entries in Cloudera Manager, instead you must use SRM Service Advanced Configuration Snippet (Safety Valve) for srm-service.yaml to configure them. If you are unsure about configuration, skip configuration and continue with the next step.

Table 3: SRM Service Migrator properties and recommendations

Property	Default Value	Description	Cloudera recommendations
streams.replication.manager.migrator.monitor.timeout.ms	3,600,000	The time in milliseconds after the Streams Replication Manager (SRM) Service Metrics Migrator times out.	Set this timeout to a value that is higher than the expected migration time. Cloudera recommends a value that is at least three times the expected migration time. The migration time depends on the amount of metrics in your deployment. The higher the number of metrics, the longer the migration process, and the higher this property must be set. For example, a deployment with 10,000 partitions (100 topics with a 100 partitions each) and a 2 hour retention period produces, at minimum, 30,000 metrics per metric emission cycle. In a case like this, migration takes around 10 minutes to finish.
streams.replication.manager.migrator.monitor.backoff.ms	120,000	The frequency at which the progress of the Streams Replication Manager (SRM) Service Metrics Migrator is checked.	The recommended values for this property differ depending on the version you are upgrading from. **If upgrading from 7.1.8:** Set this property to a value that is identical with or similar to the interval set in SRM Service Streams Commit Interval The default value of this property is identical with the default value of SRM Service Streams Commit Interval. **If upgrading from 7.1.7 or lower:** Set this property to 30,000 (30 seconds).

Property	Default Value	Description	Cloudera recommendations
streams.replication.manager.migrator.monitor.stop.delay.ms	60,000	The amount of time in milliseconds that the Streams Replication Manager (SRM) Service Metrics Migrator processes metrics after the streams application is considered caught up	
streams.replication.manager.migrator.monitor.min.consecutive.successful.checks	3	The number of consecutive checks where the lag must be within the configured threshold to consider the Streams Replication Manager (SRM) Service Metrics Migrator successful. All target clusters of the SRM Service must be caught up for a check to be successful.	
streams.replication.manager.migrator.monitor.max.offset.lag	Calculated automatically if left empty.	The amount of offsets that the streams application in the Streams Replication Manager (SRM) Service Metrics Migrator is allowed to lag behind and still be considered up to date. When left empty, the migration logic will automatically calculate the maximum offset lag based on the Kafka Streams application configuration and the amount of metrics messages. Low offset lag values result in more up-to-date metrics processing following the upgrade, but also increase the time required for the upgrade to finish.	An appropriate value for this property is calculated automatically if the property is left empty. As a result, Cloudera recommends that you leave this property empty and use the automatically calculated value.

e. Click Save Change.

f. Restart the SRM service.

19. If your cluster uses Cruise Control and you have customized the goals in Cruise Control, ensure that you have created a copy from the values of the following goals before upgrading your cluster:

- Default goals
- Supported goals
- Hard goals
- Self-healing goals
- Anomaly detection goals

For more information, see the Cruise Control Known Issues.

20. The following services are no longer supported as of Cloudera Base on premises:

    - Accumulo
    - Data Analytics Studio (DAS)
    - Sqoop 2
    - MapReduce 1
    - Record Service

    You must stop and delete these services before upgrading a cluster. You can delete the associated databases to free-up resources.
21. Open the Cloudera Manager Admin console and collect the following information about your environment:

    a. The version of Cloudera Manager. Go to Support About .
    b. The version of the JDK deployed. Go to Support About .
    c. The version of CDH or Cloudera Runtime and whether the cluster was installed using parcels or packages. It is displayed next to the cluster name on the Home page.
    d. The services enabled in your cluster.

       Go to Clusters *CLUSTER NAME* .
22. Back up Cloudera Manager before beginning the upgrade. See Step 2: Backing Up Cloudera Manager 6 on page 71.

Step 2: Review Notes and Warnings

Notes and warnings to consider before upgrading to Cloudera Base on premises.

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

Note the following before upgrading your clusters:

Warning: If there is any failure during the upgrade process, you must contact Cloudera customer support.

Warning: If there is any failure during the Cloudera Runtime upgrade process, you must not delete or modify the records in the upgrade_state table. If you still want to delete or modify the upgrade_state table, you must contact the Cloudera development team for modification or deletion approval. If you proceed without approval from the Cloudera development team, it can cause additional issues while resolving the runtime upgrade failure.

Note: Cloudera recommends all upgrades to happen in a maintenance window by throttling and scaling down workloads during that time as a best practice.

Important:

Spark 3 is the default Spark version in Cloudera Runtime. Spark 2 (along with Livy 2) has been removed and no longer available in 7.3.1

Spark 3 contains a large number of changes from Spark 2. Refer to Upgrading Spark for more information on upgrading Spark clusters to 7.3.1, and Migrating Spark Applications for more information on migrating your existing Spark applications between versions 2 and 3.

Important:

- The embedded PostgreSQL database is NOT supported in production environments.
- The following services are no longer supported as of Enterprise 6.0.0:
  - Sqoop 2
  - MapReduce 1
  - Spark 1.6
  - Record Service
- Running Apache Accumulo on top of CDP Private Cloud Base 7.1.x cluster is not currently supported. If you try to upgrade to CDP Private Cloud Base 7.1.x, you will be asked to remove the Accumulo service from your cluster.
- Upgrading Apache HBase from CDH to Cloudera Runtime 7.1.1 gives you a warning in Cloudera Manager that the Dynamic Jars Directory feature property hbase.dynamic.jars.dir is deprecated. You can ignore this warning when using Apache HBase with HDFS storage on Cloudera Base on premises. The hbase.dynamic.jars.dir property is incompatible with Apache HBase on cloud deployments using cloud storage.
- The minor version of Cloudera Manager you use to perform the upgrade must be equal to or greater than the CDH or Cloudera Runtime minor version. Cloudera recommends that you upgrade to the latest maintenance version of Cloudera Manager before upgrading your cluster. See Supported Upgrade Paths. To upgrade Cloudera Manager, see Upgrading Cloudera Manager 6 on page 66.

  For example:

  - Supported:
    - Cloudera Manager 7.1 or higher and Cloudera Runtime 7.0
    - Cloudera Manager 7.1 and CDH 5.
    - Cloudera Manager 6.0.0 and CDH 5.14.0
    - Cloudera Manager 5.14.0 and CDH 5.13.0
    - Cloudera Manager 5.13.1 and CDH 5.13.3
  - Not Supported:
    - Cloudera Manager 5.14.0 and CDH 6.0.0
    - Cloudera Manager 5.12 and CDH 5.13
    - Cloudera Manager 6.0.0 and CDH 5.6

Note:

After upgrading from CDH to CDP Private Cloud Base, the NodeManager recovery feature is enabled by default. This means that the yarn.nodemanager.recovery.enabled property is set to true. Cloudera recommends that you keep the NodeManager recovery feature enabled. If you set this property to false in your CDP Private Cloud Base cluster and then upgrade to a later CDP Private Cloud Base version, the feature will remain disabled.

Note: Upgrades to Cloudera Runtime 7.0.3 are not supported.

Note:

When upgrading CDH using Rolling Restart (Minor Upgrade only):

- Automatic failover does not affect the rolling restart operation.
- After the upgrade has completed, do not remove the old parcels if there are MapReduce or Spark jobs currently running. These jobs still use the old parcels and must be restarted in order to use the newly upgraded parcel.
- Ensure that Oozie jobs are idempotent.
- Do not use Oozie Shell Actions to run Hadoop-related commands.
- Rolling upgrade of Spark Streaming jobs is not supported. Restart the streaming job once the upgrade is complete, so that the newly deployed version starts being used.
- Runtime libraries must be packaged as part of the Spark application.
- You must use the distributed cache to propagate the job configuration files from the client gateway hosts.
- Do not build "uber" or "fat" JAR files that contain third-party dependencies or CDH/Cloudera Runtime classes as these can conflict with the classes that Yarn, Oozie, and other services automatically add to the CLASSPATH.
- Build your Spark applications without bundling CDH/Cloudera Runtime JARs.

Cruise Control might fail during an upgrade

Warning: Cruise Control might fail during the restart process when upgrading to CDP Private Cloud Base 7.1.4. For more information, see the Cruise Control Release Notes.

Cruise Control does not function properly during the upgrade

Warning: Cruise Control does not work properly during the upgrade process of the Kafka service. This also applies to rolling upgrades.

The integration between Streams Messaging Manager and Streams Replication Manager is changed

Important:

In Cloudera Runtime 7.1.6 and higher, the way Streams Messaging Manager (SMM) integrates with Streams Replication Manager (SRM) has changed. SMM can only connect to and monitor an SRM service that is running in the same cluster as SMM. Monitoring an SRM service that is running in a cluster that is external to SMM is no longer supported.

Connectivity between the two services is disabled by default after a successful upgrade. If you want to continue using SMM to monitor SRM, you must reconnect the two services following the upgrade.

Replication Manager

Before you use Replication Manager with CDP Private Cloud Base 7.1.9 SP1 CHF11 and Cloudera Manager 7.13.1.400, you must be aware that certain Cloudera Replication Manager fixes are not included in this combination. While general usage is supported, specific features or use cases might be impacted if they rely on the missing fixes.

To ensure full feature compatibility and to access the latest fixes, Cloudera recommends upgrading to Cloudera Manager 7.13.1.500 or higher versions when using CDP Private Cloud Base 7.1.9 SP1 CHF11.

HDFS

Warning:

- Cloudera recommends you to not create any new encryption zone unless HDFS metadata is finalized.

Ozone

Important:

- Using Cloudera Manager 7.7.1, if you are upgrading from CDP Private Cloud Base 7.1.8 with Ozone parcels to CDP Private Cloud Base 7.1.9 using Cloudera Manager 7.11.3, then you must
  1. If you have Ozone parcel 1.0 on the CDP Private Cloud Base 7.1.8 cluster, you must deactivate and undistribute Ozone parcel 1.0. If you have Ozone parcel 2.x on the CDP Private Cloud Base 7.1.8 cluster, you must only deactivate Ozone parcel 2.x.
  2. Upgrade Cloudera Manager from 7.7.1 to 7.11.3. Do not restart the Cloudera Base on premises cluster.
  3. Upgrade from CDP Private Cloud Base 7.1.8 to CDP Private Cloud Base 7.1.9.
- When you upgrade from the lower version of CDP Private Cloud Base to CDP Private Cloud Base 7.1.9, the quota related information will be repaired during the cluster upgrade. The upgrade activity will take time based on number of keys present in the system. This is a one time activity to correct the quota and usages information for space and namespace usages.
- If you have the Ozone HttpFS role added to the Ozone service on your CDP Private Cloud Base 7.1.8 cluster, you must stop and delete the Ozone HttpFS role from the Ozone service before upgrading the Cloudera Runtime cluster to CDP Private Cloud Base 7.1.9. After you upgrade the Cloudera Runtime cluster to CDP Private Cloud Base 7.1.9, you can add the HttpFS role back to the Ozone service.
- Non-HA Ozone upgrades are not supported using Cloudera Manager.
- Multiple Ozone deployments are not supported in the Cloudera cluster. Ensure that you have only one Ozone deployement Service ID in the ozone.om.service.ids configuration of the Cloudera cluster. If you have multiple Ozone deployement Service IDs in the ozone.om.service.ids configuration, then remove all of them except one.

Oozie ShareLib update and authorization

Updating the Oozie ShareLib is considered as an admin operation. If you have configured authorization in Oozie, then only admin users are able to trigger a ShareLib update.

Important: During a runtime upgrade, Cloudera Manager triggers a ShareLib update automatically. Please ensure that the admin users are configured correctly or this operation fails, resulting in an upgrade failure.

Hive metastore

If you have created any materialized views or MSSQL indexed views on Hive backend schemas, such as SYS or INFORMATION_SCHEMA tables, your upgrade process can fail when the upgrade SQL statements are trying to drop these tables.

You must drop the materialized views before performing an upgrade and then recreate the views after the upgrade process is complete.

Important: Cloudera recommends that you do not create any materialized views or indexed views on the SYS and INFORMATION_SCHEMA tables. However, if you have already created the views, then perform the following steps to identify such views and drop them before upgrading the cluster.

1. Ensure that you have backed up the Hive metastore (HMS) backend database before dropping materialized views.
2. Start a Hive Beeline session and run the following query to identify materialized views that are created on top of the SYS and INFORMATION_SCHEMA tables:

```
SELECT DISTINCT d.DB_LOCATION_URI, d.NAME, t.TBL_NAME, t.TBL_TYPE, t.OWN
ER, t.VIEW_EXPANDED_TEXT
FROM sys.TBLS t
         INNER JOIN sys.DBS d ON t.DB_ID = d.DB_ID
         INNER JOIN sys.MV_CREATION_METADATA mv ON mv.TBL_NAME = t.TBL_N
AME
         INNER JOIN sys.MV_TABLES_USED tu ON mv.MV_CREATION_METADATA_ID =
 tu.MV_CREATION_METADATA_ID
WHERE tu.TBL_ID IN (SELECT distinct t.TBL_ID
```

```
                      FROM sys.MV_CREATION_METADATA mv
                               INNER JOIN sys.MV_TABLES_USED tu ON mv.MV_CR
EATION_METADATA_ID = tu.MV_CREATION_METADATA_ID
                               INNER JOIN sys.TBLS t ON tu.TBL_ID = t.TBL_ID
                               INNER JOIN sys.DBS d ON t.DB_ID = d.DB_ID
                      WHERE lower(d.NAME) IN ('sys', 'information_schema'))
  AND upper(t.TBL_TYPE) = 'MATERIALIZED_VIEW';
```

3. If the query returns any materialized views, drop each view using the DROP statement.

```
DROP MATERIALIZED VIEW [db_name.]materialzed_view_name;
```

4. Upgrade the cluster and recreate the views after the upgrade process is complete.

Known issue during CDP Private Cloud Base 7.1.7 SP2 to CDP Private Cloud Base 7.1.9 upgrade

OPSAPS-68279: When upgrading CDP Private Cloud Base 7.1.7 SP2 to CDP Private Cloud Base 719, sometimes the command step DeployClientConfig fails due to the following error:

Error Message:Client configuration generation requires the following additional parcels to be acti vated:[cdh]

This can be because of the failure of the activation of the Cloudera Base on premises 7.1.9 parcels. To verify:

1. Navigate to the parcels page.
2. See if the following error is displayed: Error when distributing to <hostname>: Sc file/o pt/cloudera/parcels/.flood/CDH-7.1.9-1.cdh7.1.9.p0.43968053-el7.parcel/CDH-7.1. 1.cdh7.1 .9.0.43968053-el7.parcel does not exist.
3. Using the error above, identify the host and ssh into the host by running the command `ssh <hostname>`.
4. Navigate to the agent directory by running the command `cd /var/log/cloudera-scm-agent`.
5. Find the following pattern in agent log file(s) Exception: Untar failed with return code: 2, wi th tar output: stdout: [b"], stderr: [b'\ngzip: stdin: invalid compressed data--format violated \ntar: Unexpected EOF in archive\ntar: Unexpected EOF in archive\ntar: Error is not reco verable: exiting now\n'].

1. If the above exception appears, you must restart the agent on that host by running the command `systemctl restart cloudera-scm-agent`.
2. After the agent restarts, click resume to continue with the upgrade.

Phoenix

CDPD-67700/CDPQE-30593: While upgrading CDH 6 to CDP Private Cloud Base 7.1.7 SP3 using a Phoenix parcel, Phoenix drops the Tephra support and causes a classpath issue

This issue occurs because an old Phoenix parcel is used.

To resolve this issue, deactivate the old Phoenix parcel.

Step 3: Backing Up the Cluster

Steps to back up your cluster before the upgrade.

This topic describes how to back up a cluster managed by Cloudera Manager prior to upgrading the cluster. These procedures do not back up the data stored in the cluster. Cloudera recommends that you maintain regular backups of your data using the Backup and Disaster Recovery features of Cloudera Manager.

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

The following components do not require backups:

- MapReduce
- YARN
- Spark
- Impala

Complete the following backup steps before upgrading your cluster:

Back Up Databases

Warning: Backing up databases requires that you stop some services, which might make them unavailable during backup.

Gather the following information:

- Type of database (PostgreSQL, Embedded PostgreSQL, MySQL, MariaDB, or Oracle)
- Hostnames of the databases
- Database names
- Port number used by the databases
- Credentials for the databases

Open the Cloudera Manager Admin Console to find the database information for any of the following services you have deployed in your cluster:

- Sqoop, Oozie, and Hue – Go to *CLUSTER NAME* Configuration Database Settings .

  Note: The Sqoop Metastore uses a HyperSQL (HSQLDB) database. See the HyperSQL documentation for backup procedures.

  Note: Sqoop 2 is not supported in CDP Private Cloud Base.

- Hive Metastore – Go to the Hive service, select Configuration, and select the Hive Metastore Database category.
- Sentry – Go to the Sentry service, select Configuration, and select the Sentry Server Database category.
- Ranger – Go to the Ranger service, select Configuration, and search on "database."
- Queue Manager – Go to the Queue Manager service, select the Configuration tab. In the List of Filters on the left side, click the Category drop-down and select Database.
- Schema Registry and Streams Messaging Manager – Select the service, go to Configuration, and select the Database category.

To back up the databases

Perform the following steps for each database you back up:

1. If not already stopped, stop the service.
   a. On the Home Status tab, click to the right of the service name and select Stop.
   b. Click Stop in the next screen to confirm. When you see a Finished status, the service has stopped.
2. Back up the database. Substitute the database name, hostname, port, user name, and backup directory path and run the following command:
   MySQL

```
mysqldump --databases
                    DATABASE_NAME
                    --host=DATABASE_HOSTNAME
                    --port=DATABASE_PORT -u
                    DATABASE_USERNAME -p >
                    BACKUP_DIRECTORY_PATH/DATABASE_NAME-backup-
`date
```

```
                        +%F`-CDH.sql
```

PostgreSQL/Embedded

```
pg_dump -h DATABASE_HOSTNAME -U DATABASE_USERNAME
 -W -p DATABASE_PORT DATABASE_NAME
 > BACKUP_DIRECTORY_PATH/DATABASE_NAME-backup-`date +%F`-CDH.sql
```

Oracle

Work with your database administrator to ensure databases are properly backed up.

For additional information about backing up databases, see these vendor-specific links:

- MariaDB 10.2: https://mariadb.com/kb/en/backup-and-restore-overview/
- MySQL 5.7: https://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html
- PostgreSQL10: https://www.postgresql.org/docs/10/static/backup.html
- Oracle 12c: https://docs.oracle.com/en/database/oracle/oracle-database/12.2/bradv/index.html

3. Start the service.
   a. On the HomeStatus tab, click to the right of the service name and select Start.
   b. Click Start in the next screen to confirm. When you see a Finished status, the service has started.

Back Up ZooKeeper

1. On all ZooKeeper hosts, back up the ZooKeeper data directory specified with the Data Directory property and ZooKeeper transaction log directory specified with the Transaction Log Directory property in the ZooKeeper configuration. The default location for both these directories is /var/lib/zookeeper.

   For example:

   ```
   cp -rp /var/lib/zookeeper/ /var/lib/zookeeper-backup-`date +%F`CM-CDH
   ```

2. To identify the ZooKeeper hosts, open the Cloudera Manager Admin console and go to the ZooKeeper service and click the Instances tab.

Record the permissions of the files and directories; you will need these to roll back ZooKeeper.

Back up Solr

You are recommended to take a backup of Solr data and indexes before an upgrade, so that data can be restored during a rollback.

When backing up the Infra Solr service, make sure that you backup all of the following collections:

- vertex_index (Atlas)
- edge_index (Atlas)
- fulltext_index (Atlas)
- ranger_audits (Ranger)

Store them in a location that is accessible to the solr service user during a rollback. Cloudera recommends you target your backups to a shared file system, even if the Solr service uses local file system.

Before you start: Solr gateway has to be installed on all edge/client nodes.

- Create a backup of Solr collections.

  For information on backing up Solr collections, see Backing up a collection from HDFS (default location for Workload Solr) or Backing up a collection from local file system (default location for Infra Solr), depending on the type of storage used.

Back Up HDFS

Follow this procedure to back up an HDFS deployment.

Note: To locate the hostnames required to backup HDFS (for JournalNodes, DataNodes, and NameNodes), open the Cloudera Manager Admin Console, go to the HDFS service, and click the Instances tab.

1. If high availability is enabled for HDFS, run the following command on all hosts running the JournalNode role:

```
cp -rp /dfs/jn /dfs/jn-CM-CDH
```

2. On all NameNode hosts, back up the NameNode runtime directory. Run the following commands:

```
mkdir -p /etc/hadoop/conf.rollback.namenode
```

```
cd /var/run/cloudera-scm-agent/process/ && cd `ls -t1 | grep -e "-NAMENODE
\$" | head -1`
```

```
cp -rp * /etc/hadoop/conf.rollback.namenode/
```

```
rm -f /etc/hadoop/conf.rollback.namenode/log4j.properties
```

```
cp -rp /etc/hadoop/conf.cloudera.HDFS_SERVICE_NAME/log4j.properties /etc/h
adoop/conf.rollback.namenode/
```

These commands create a temporary rollback directory. If a rollback is required later, the rollback procedure requires you to modify files in this directory.

3. Back up the runtime directory for all DataNodes. Run the following commands on all DataNodes:

```
mkdir -p /etc/hadoop/conf.rollback.datanode/
```

```
cd /var/run/cloudera-scm-agent/process/ && cd `ls -t1 | grep -e "-DATANODE
\$" | head -1`
```

```
cp -rp * /etc/hadoop/conf.rollback.datanode/
```

```
rm -f /etc/hadoop/conf.rollback.datanode/log4j.properties
```

```
cp -rp /etc/hadoop/conf.cloudera.HDFS_SERVICE_NAME/log4j.properties /etc/h
adoop/conf.rollback.datanode/
```

4. If high availability is not enabled for HDFS, backup the runtime directory of the Secondary NameNode. Run the following commands on all Secondary NameNode hosts:

```
mkdir -p /etc/hadoop/conf.rollback.secondarynamenode/
```

```
cd /var/run/cloudera-scm-agent/process/ && cd `ls -t1 | grep -e "-SECOND
ARYNAMENODE\$" | head -1`
```

```
cp -rp * /etc/hadoop/conf.rollback.secondarynamenode/
```

```
rm -f /etc/hadoop/conf.rollback.secondarynamenode/log4j.properties
```

```
cp -rp /etc/hadoop/conf.cloudera.HDFS_SERVICE_NAME/log4j.properties /etc/h
adoop/conf.rollback.secondarynamenode/
```

Note: For more information on backing up HDFS, see Checkpoint HDFS documentation.

Back Up Key Trustee Server and Clients

For the detailed procedure, see Backing Up Key Trustee Server and Clients.

Back Up HSM KMS

When running the HSM KMS in high availability mode, if either of the two nodes fails, a role instance can be assigned to another node and federated into the service by the single remaining active node. In other words, you can bring a node that is part of the cluster, but that is not running HSM KMS role instances, into the service by making it an HSM KMS role instance–more specifically, an HSM KMS proxy role instance and an HSM KMS metastore role instance. So each node acts as an online ("hot" backup) backup of the other. In many cases, this will be sufficient. However, if a manual ("cold" backup) backup of the files necessary to restore the service from scratch is desirable, you can create that as well.

To create a backup, copy the /var/lib/hsmkp and /var/lib/hsmkp-meta directories on one or more of the nodes running HSM KMS role instances.

To restore from a backup: bring up a completely new instance of the HSM KMS service, and copy the /var/lib/hsmkp and /var/lib/hsmkp-meta directories from the backup onto the file system of the restored nodes before starting HSM KMS for the first time.

Back Up Navigator Encrypt

It is recommended that you back up Navigator Encrypt configuration directory after installation, and again after any configuration updates.

1. To manually back up the Navigator Encrypt configuration directory (/etc/navencrypt):

```
$ zip -r --encrypt nav-encrypt-conf.zip /etc/navencrypt
```

   The --encrypt option prompts you to create a password used to encrypt the zip file. This password is also required to decrypt the file. Ensure that you protect the password by storing it in a secure location.
2. Move the backup file (nav-encrypt-conf.zip) to a secure location.

Warning: Failure to back up the configuration directory makes your backed-up encrypted data unrecoverable in the event of data loss.

Back Up HBase

Because the rollback procedure also rolls back HDFS, the data in HBase is also rolled back. In addition, HBase metadata stored in ZooKeeper is recovered as part of the ZooKeeper rollback procedure.

If your cluster is configured to use HBase replication, Cloudera recommends that you document all replication peers. If necessary (for example, because the HBase znode has been deleted), you can roll back HBase as part of the HDFS rollback without the ZooKeeper metadata. This metadata can be reconstructed in a fresh ZooKeeper installation, with the exception of the replication peers, which you must add back. For information on enabling HBase replication, listing peers, and adding a peer, see HBase Replication in the CDH 5 documentation.

Back Up YARN Queue Manager

Learn how to back up Yarn Queue Manager for CDP Private Cloud Base versions 7.1.8 and below. These steps are necessary if you want to upgrade to CDP Private Cloud Base 7.1.9 from version 7.1.8 and below as there is no ability to roll back changes if a CDP Private Cloud Base 7.1.9 upgrade is unsuccessful.

1. In Cloudera Manager, navigate to Clusters Hosts . Backup the configuration service database.
2. Locate the host that has the Yarn Queue Manager Store running.
3. Find the location of the config-service database file by navigating to Cluster QueueManager Service Configurations tab Scope , and click the Yarn Queue Manager Store.
4. Locate the Location for config-service DB field. If the field is empty, then use the default location: Database Location -> /var/lib/hadoop-yarn/
5. Open a SSH terminal and enter the following command: ssh [***your_username***]@[***queue_manager_host_ip_address***]
6. Navigate to the directory where the configuration database file is stored: cd {Database Location}
7. Find two database files with these names: -config-service.mv.db

   -config-service.trace.db

   Notice that config-service.trace.db is in the same location.
8. Secure copy the config-service.mv.db and config-service.trace.db files to the machines where the backups are to be stored. For example: scp -i ~/.ssh/{ssh_key} config-service.mv.db root@{hostName}:{Your_Backup_Folder}/config-service.mv.db
9. Use sha1sum to verify that the files in the current host and the location of where the backup is stored have the same hash.

Back up Atlas

When you plan to back up your Atlas data, it is a two-step process where you must first back up Solr and HBase data before proceeding further.

Back up Solr

Follow the instructions to back up your data in Solr. You must run these commands on single solr server.

1. `curl -ivk "https://host1.example.com:8993/solr/admin/collections?action=BACKUP&name=vertex_index_bkp&collection=vertex_index&location=/tmp/"`
2. `curl -ivk "https://host1.example.com:8993/solr/admin/collections?action=BACKUP&name=edge_index_bkp&collection=edge_index&location=/tmp/"`
3. `curl -ivk "https://host1.example.com:8993/solr/admin/collections?action=BACKUP&name=fulltext_index_bkp&collection=fulltext_index&location=/tmp/"`

Note:

1. /tmp/ this can be replaced with the backup directory path which is to be used to store Solr backup. This path needs to be accessible for the solr service user.
2. If the cluster is kerberized, then run kinit against Solr keytab first and add "--negotiate -u :" after the -ivk flag in the above curl commands. Example: `curl -ivk --negotiate -u : https://host1.example.com:8993`...
3. If you have multiple Solr services, ensure you create the Solr backup directories on all the services. Else the backup fails indicating that there should be a shared storage used for backup.
4. In some cases if Shards are present on different nodes, backup might fail with following message: `org.apache.solr.client.solrj.impl.HttpSolrClient $RemoteSolrException:Error from server at http://host1.example.com:8993/solr: Failed to backup core=vertex_index_shard because org.apache.solr.common.SolrException: Directory to contain snapshots doesn't exist: file:///tmp/vertex_index. Note that Backup/Restore of a SolrCloud collection requires a shared file system mounted at the same path on all nodes!"`
5. Solr recommends having a backup using HDFS repository in such a scenario. For more information, see Backup and Restore Solr Repositories.

Back up HBase

Follow the instructions to back up your data in HBase:

If the cluster is kerberized, then run kinit against HBase keytab

1. Create HBase table snapshot:
   a. `hbase shell`

      ```
      hbase> snapshot 'atlas_janus', 'atlas_janus_snapshot_<insert-date-here>'
      hbase> snapshot 'ATLAS_ENTITY_AUDIT_EVENTS',
      'atlas_entity_audit_events_snap_<insert-date-here>'
      ```

      # exit

      Note: You can use the Table Browser in Cloudera Manager to take a snapshot from the atlas_janus table.

2. Export Snapshot from server terminal:
   a. `hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot 'atlas_janus_snapshot_<insert-date-here>' -copy-to /tmp/hbasebackup/`
   b. `hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot 'atlas_entity_audit_events_snap_<insert-date-here>' -copy-to /tmp/hbasebackup/`

The contents of '/tmp/hbasebackup/' contain the table backup.

In case of below error:

```
ERROR snapshot.ExportSnapshot: Snapshot export failed
org.apache.hadoop.security.AccessControlException: Permission denied:
user=hbase, access=WRITE, inode="/user":hdfs:supergroup:drwxr-xr-x at
org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.check(FSPermissionChecker.
java:553)
```

To resolve the above error, provide the necessary permission to “hbase” user in “all - path” policy in the cm_hdfs service in Ranger. Also ensure, “hbase” user has permission in the “hbase-archive” policy as well.

Back Up Sqoop 2

If you are not using the default embedded Derby database for Sqoop 2, back up the database you have configured for Sqoop 2. Otherwise, back up the repository subdirectory of the Sqoop 2 metastore directory. This location is specified with the Sqoop 2 Server Metastore Directory property. The default location is: /var/lib/sqoop2. For this default location, Derby database files are located in /var/lib/sqoop2/repository.

Note: Sqoop 2 is not supported in CDP Private Cloud Base.

Back Up Hue

Back up the app registry file on all hosts running the Hue Server role if you have installed CDP Private Cloud Base using RPM packages.

The app registry file (app.reg) is present in the /usr/lib/hue directory if you have installed Hue using the RPM package. It is a JSON file which contains the details of all apps that are used within Hue. If you have installed Hue using the parcels, then the app.reg file may not be present on your system, and you do not need to back it up.

Run the following command to back up the app.reg file for installations using RPM packages:

```
cp -rp /usr/lib/hue/app.reg /usr/lib/hue_backup/app.reg-CM-CDH
```

Back Up Impala

Back up the Impala profile logs and logs of catalogd, statestore, and coordinators that can help investigate performance regressions. Perform the following tasks to back up the Impala cluster:

- Back up query profile logs of Impala. The logs are present in the /profiles directory under the coordinator log directory and the filename is similar to impala_profile_log_1.1-1715070416212.

  To investigate performance regressions that are introduced after an upgrade, you will require the query profiles from the previous cluster (older version). You can use tools like impala-profile-tool to decode the query profiles into text profiles and then search for the runs before the upgrade for a comparison.
- Back up logs of catalogd, statestore, and coordinators that can also be helpful in troubleshooting. By default, the Impala logs are stored at /var/log/catalogd/, /var/log/statestore/, and /var/log/impalad/.

  The log folders can be configured in Cloudera Manager by navigating to Clusters IMPALA and searching for the "Catalog Server Log Directory", "StateStore Log Directory", and "Impala Daemon Log Directory" properties.

Step 4: Back Up Cloudera Manager

After upgrading Cloudera Manager and before upgrading a cluster, you should backup Cloudera Manager again.

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

Collect Information for Backing Up Cloudera Manager

Information you should collect before backing up Cloudera Manager.

1. Log in to the Cloudera Manager Server host.

   ```
   ssh MY_CLOUDERA_MANAGER_SERVER_HOST
   ```
2. Collect database information by running the following command:

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

   For example:

   ```
   ...
   com.cloudera.cmf.db.type=...
   ```

```
com.cloudera.cmf.db.host=DATABASE_HOSTNAME:DATABASE_PORT
com.cloudera.cmf.db.name=SCM
com.cloudera.cmf.db.user=SCM
com.cloudera.cmf.db.password=SOME_PASSWORD
```

3. Collect information (host name, port number, database name, user name and password) for the following databases.
   - Reports Manager

   You can find the database information by using the Cloudera Manager Admin Console. Go to Clusters Cloudera Management Service Configuration and select the Database category. You may need to contact your database administrator to obtain the passwords.
4. Find the host where the Service Monitor, Host Monitor and Event Server roles are running. Go to Clusters Cloudera Management Service Instances and note which hosts are running these roles.

Back Up Cloudera Manager Agent

Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups, defined by the environment variable CM_BACKUP_DIR, which is used in all the backup commands. You may change these destination paths in the command as needed for your deployment.

The tar commands in the steps below may return the following message. It is safe to ignore this message:

```
tar: Removing leading `/' from member names
```

Backup up the following Cloudera Manager agent files on all hosts:

- Create a top level backup directory.

```
export CM_BACKUP_DIR="`date +%F`-CM"
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

- Back up the Cloudera Manager Agent directory and the runtime state.

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-agent.tar --exclude=*.sock /et
c/cloudera-scm-agent /etc/default/cloudera-scm-agent /var/run/cloudera-s
cm-agent /var/lib/cloudera-scm-agent
```

- Back up the existing repository directory.
  RHEL / CentOS

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
```

  SLES

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
```

  Ubuntu

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources
.list.d
```

Back Up the Cloudera Management Service

Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups. You may change these destination paths in the command as needed for your deployment.

1. Stop the Cloudera Management Service.
   a. Log in to the Cloudera Manager Admin Console.
   b. Go to Cloudera Management Service .
   c. Select Actions Stop .
2. On the host where the Service Monitor role is configured to run, backup the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-SERVICE-MONITOR /var/lib/cloudera-service-m
onitor-`date +%F`-CM
```

3. On the host where the Host Monitor role is configured to run, backup the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-HOST-MONITOR /var/lib/cloudera-host-monitor-
`date +%F`-CM
```

4. On the host where the Event Server role is configured to run, back up the following directory:

```
sudo cp -rp /VAR/LIB/CLOUDERA-SCM-EVENTSERVER /var/lib/cloudera-scm-event
server-`date +%F`-CM
```

5. Restart the Cloudera Management Service if you are not upgrading Cloudera Manager immediately.
   a. Log in to the Cloudera Manager Admin Console.
   b. Select Clusters Cloudera Management Service .
   c. Select Actions Start .

Back Up Cloudera Navigator Data

1. **Important:** Upgrading from Cloudera Manager 5.9 (Navigator 2.8) and earlier can take a significant amount of time, depending on the size of the Navigator Metadata storage directory. When the Cloudera Manager upgrade process completes and Cloudera Navigator services restart, the Solr indexing upgrade automatically begins. No other actions can be performed until Solr indexing completes (a progress message displays during this process). It can take as long as two days to upgrade a storage directory with 60 GB. To help mitigate this extended upgrade step, make sure to clear out all unnecessary metadata using purge, check the size of the storage directory, and consider rerunning purge with tighter conditions to further reduce the size of the storage directory.
2. Make sure a purge task has run recently to clear stale and deleted entities.
   - You can see when the last purge tasks were run in the Cloudera Navigator console (From the Cloudera Manager Admin console, go to ClustersCloudera Navigator. Select AdministrationPurge Settings.)
   - If a purge hasn't run recently, run it by editing the Purge schedule on the same page.
   - Set the purge process options to clear out as much of the backlog of data as you can tolerate for your upgraded system. See Managing Metadata Storage with Purge.
3. Stop the Navigator Metadata Server.
   a. Go to ClustersCloudera Management ServiceInstances.
   b. Select Navigator Metadata Server.
   c. Click Actions for SelectedStop.
4. Back up the Cloudera Navigator Solr storage directory.

```
sudo cp -rp /VAR/LIB/CLOUDERA-SCM-NAVIGATOR /var/lib/cloudera-scm-navig
ator-`date +%F`-CM
```

5. If you are using an Oracle database for audit, in SQL*Plus, ensure that the following additional privileges are set:

```
GRANT EXECUTE ON sys.dbms_crypto TO nav;
GRANT CREATE VIEW TO NAV;
```

where *NAV* is the user of the Navigator Audit Server database.

Stop Cloudera Manager Server & Cloudera Management Service

1. Stop the Cloudera Management Service.
   a. Log in to the Cloudera Manager Admin Console.
   b. Select Clusters Cloudera Management Service .
   c. Select Actions Stop .
2. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

3. Stop the Cloudera Manager Server.

```
sudo systemctl stop cloudera-scm-server
```

Back Up the Cloudera Manager Databases

During the upgrade, Cloudera Manager modifies the schema of the Cloudera Manager database. In case of failures during the upgrade, it may be necessary to rollback to the previous version of Cloudera Manager while addressing the upgrade failures.

When performing a rollback to a previous version, the Cloudera Manager Database must be restored to the previous database schema. For this reason, you must do the Cloudera Manager database backup, so that it can be restored if a rollback is necessary.

Tip:

You can get the name, user, and password properties for the Cloudera Manager database from the /etc/cloudera-scm-server/db.properties file:

```
com.cloudera.cmf.db.name=scm
com.cloudera.cmf.db.user=scm
com.cloudera.cmf.db.password=NnYfWIjlbk
```

1. Back up the Cloudera Manager server database as per the instructions provided in your respective database documentation on how to backup and restore the databases.

2. Back up All other Cloudera Manager databases - Use the database information that you collected in a previous step. You may need to contact your database administrator to obtain the passwords.

   These databases can include the following:

   - 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 database. If this information is not available, your cluster cannot start or function correctly. You must schedule and maintain regular backups of the 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.
   - Sentry Server - Contains authorization metadata. Relatively small.
   - Navigator Audit Server - Contains auditing information. In large clusters, this database can grow large.(Only available when installing CDH 5 or CDH 6 clusters.)
   - Navigator Metadata Server - Contains authorization, policies, and audit report metadata. Relatively small. (Only available when installing CDH 5 or CDH 6 clusters.)
   - DAS PostgreSQL server - Contains Hive and Tez event logs and DAG information. Can grow very large.
   - Ranger Admin - Contains administrative information such as Ranger users, groups, and access policies. Medium-sized.
   - 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.

   For more information about the number of databases that should be backed up, and restored if necessary, see Required Databases.

Back Up Cloudera Manager Server

Note: Commands are provided below to backup various files and directories used by Cloudera Manager Agents. If you have configured custom paths for any of these, substitute those paths in the commands. The commands also provide destination paths to store the backups, defined by the environment variable CM_BACKUP_DIR, which is used in all the backup commands. You may change these destination paths in the command as needed for your deployment.

The tar commands in the steps below may return the following message. It is safe to ignore this message:

```
tar: Removing leading `/' from member names
```

1. Log in to the Cloudera Manager Server host.

   ```
   ssh MY_CLOUDERA_MANAGER_SERVER_HOST
   ```

2. Create a top-level backup directory.

```
export CM_BACKUP_DIR="`date +%F`-CM"
echo $CM_BACKUP_DIR
mkdir -p $CM_BACKUP_DIR
```

3. **a.** Back up the Cloudera Manager Server directories along with the CSP key file when Credential Storage Provider (CSP) is enabled:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-s
cm-server /etc/default/cloudera-scm-server ***/path/to/csp/keys***
```

Important: In the above command, you must replace the ***/path/to/csp/key*** with the actual path where the CSP key has been stored from the following options:

- /var/lib/cloudera-scm-server/csp-data
- /opt/cloudera/
- /etc/cloudera/
- /var/cloudera/

b. Back up the Cloudera Manager Server directories without CSP key file:

```
sudo -E tar -cf $CM_BACKUP_DIR/cloudera-scm-server.tar /etc/cloudera-s
cm-server /etc/default/cloudera-scm-server
```

4. Back up the existing repository directory.
RHEL / CentOS

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
```

SLES

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
```

Ubuntu

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources
.list.d
```

(Optional) Start Cloudera Manager Server & Cloudera Management Service

Start the Cloudera Manager server and Cloudera Manager Management service.

If you will be immediately upgrading Cloudera Manager, skip this step and continue with Step 3: Upgrading the Cloudera Manager Server on page 78.

1. Log in to the Cloudera Manager Server host.

```
ssh MY_CLOUDERA_MANAGER_SERVER_HOST
```

2. Start the Server.

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

3. Start the .

a. Log in to the Admin Console.
b. Select Clusters .
c. Select Actions Start .

Step 5: Complete Pre-Upgrade steps for upgrades to Cloudera Base on premises

Steps to complete before upgrading CDH to CDP.

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

Ensure that you have completed the following steps when upgrading from CDH 5.x to Cloudera Base on premises 7.1.

- Flume – Flume is not supported in Cloudera Base on premises. You must remove the Flume service before upgrading to Cloudera Base on premises.
- HBase – See Checking Apache HBase on page 40.
- Hive – See Migrating Hive 1-2 to Hive 3 on page 35
- Kafka – In CDH 5.x, Kafka was delivered as a separate parcel and could be installed along with CDH 5.x using Cloudera Manager. In Runtime 7.0.3 and later, Kafka is part of the Cloudera Runtime distribution and is deployed as part of the Cloudera Runtime parcels. To successfully upgrade Kafka you need to set the protocol version to match what's being used currently among the brokers and clients.

  Important: Upgrading CDK to Cloudera Runtime 7.1.1 or higher is only supported from CDK 4.1.0. If you are running an earlier version of CDK, you must first upgrade to CDK 4.1.0 before upgrading to Cloudera Runtime 7.1.1.

  1. Explicitly set the Kafka protocol version to match what's being used currently among the brokers and clients. Update kafka.properties on all brokers as follows:
     a. Log in to the Cloudera Manager Admin Console
     b. Choose the Kafka service.
     c. Click Configuration.
     d. Use the Search field to find the Kafka Broker Advanced Configuration Snippet (Safety Valve) for kafka.properties configuration property.
     e. Add the following properties to the snippet:
        - inter.broker.protocol.version = *[***CURRENT KAFKA VERSION***]*
        - log.message.format.version = *[***CURRENT KAFKA VERSION***]*

        Replace *[***CURRENT KAFKA VERSION***]* with the version of Apache Kafka currently being used. See the Product Compatibility Matrix for CDK Powered By Apache Kafka to find out which upstream version is used by which version of CDK. Make sure you enter full Apache Kafka version numbers with three values, such as 0.10.0. Otherwise, you will see an error message similar to the following:

```
2018-06-14 14:25:47,818 FATAL kafka.Kafka$:
java.lang.IllegalArgumentException: Version `0.10` is not a valid ve
rsion
        at kafka.api.ApiVersion$$anonfun$apply$1.apply(ApiVersion.
scala:72)
        at kafka.api.ApiVersion$$anonfun$apply$1.apply(ApiVersion.
scala:72)
        at scala.collection.MapLike$class.getOrElse(MapLike.scala:
128)
```

  2. Save your changes. The information is automatically copied to each broker.

- Kafka – To successfully upgrade Kafka, you need to set the protocol version to match what's being used currently among the brokers and clients. Following a successful upgrade, you will need to reset the configuration change made.

  1. Explicitly set the Kafka protocol version to match what's being used currently among the brokers and clients. Update kafka.properties on all brokers as follows:

     a. Log in to the Cloudera Manager Admin Console
     b. Choose the Kafka service.
     c. Click Configuration.
     d. Use the Search field to find the Kafka Broker Advanced Configuration Snippet (Safety Valve) for kafka.properties configuration property.
     e. Add the following properties to the snippet:

        - inter.broker.protocol.version = *[***CURRENT KAFKA VERSION***]*
        - log.message.format.version = *[***CURRENT KAFKA VERSION***]*

     Replace *[***CURRENT KAFKA VERSION***]* with the version of Apache Kafka currently being used. See the CDH 6 Packaging Information to find out which upstream version is used by which version of CDH 6. Make sure you enter full Apache Kafka version numbers with three values, such as 0.10.0. Otherwise, you will see an error message similar to the following:

     ```
     2018-06-14 14:25:47,818 FATAL kafka.Kafka$:
     java.lang.IllegalArgumentException: Version `0.10` is not a valid ve
     rsion
             at kafka.api.ApiVersion$$anonfun$apply$1.apply(ApiVersion.
     scala:72)
             at kafka.api.ApiVersion$$anonfun$apply$1.apply(ApiVersion.
     scala:72)
             at scala.collection.MapLike$class.getOrElse(MapLike.scala:
     128)
     ```

  2. Save your changes. The information is automatically copied to each broker.
- Navigator –- See Transitioning Navigator content to Atlas on page 20
- Replication Schedules – See CDH cluster upgrade requirements for Replication Manager on page 43.
- Sentry The Sentry service has been replace with Apache Ranger in Cloudera Runtime 7.1. You must perform several steps before upgrading your cluster. See Transitioning the Sentry service to Apache Ranger on page 16.
- Virtual Private Clusters:

  If your deployment has defined a Compute cluster and an associated Data Context, you will need to delete the Compute cluster and Data context before upgrading the base cluster and then recreate the Compute cluster and Data context after the upgrade.
- YARN : Decommission and recommission the YARN NodeManagers but do not start the NodeManagers. A decommission is required so that the NodeManagers stop accepting new containers, kill any running containers, and then shutdown.

  1. Ensure that new applications, such as MapReduce or Spark applications, will not be submitted to the cluster until the upgrade is complete.
  2. In the Cloudera Manager Admin Console, navigate to the YARN service for the cluster you are upgrading.
  3. On the Instances tab, select all the NodeManager roles. This can be done by filtering for the roles under Role Type.
  4. Click Actions for Selected (number) Decommission .

     If the cluster runs CDH 5.9 or higher and is managed by Cloudera Manager 5.9 or higher, and you configured graceful decommission, the countdown for the timeout starts.

     A Graceful Decommission provides a timeout before starting the decommission process. The timeout creates a window of time to drain already running workloads from the system and allow them to run to completion.

Search for the Node Manager Graceful Decommission Timeout field on the Configuration tab for the YARN service, and set the property to a value greater than 0 to create a timeout.

5. Wait for the decommissioning to complete. The NodeManager State is Stopped and the Commission State is Decommissioned when decommissioning completes for each NodeManager.
6. With all the NodeManagers still selected, click Actions for Selected (number) Recommission .

Important: Do not start the NodeManagers.

- HDFS: Review the current JVM heap size for the DataNodes on your cluster and ensure that the heap size is configured at the rate of 1 GB for every million blocks. Use the Java Heap Size of DataNode in Bytes property to configure the value.

Note: If upgrading to 7.1.7 or greater, you need *not* increase the heap size.

In addition, you can track the JVM heap usage through Cloudera Manager charts, as specified in the following steps:

1. Open the Cloudera Manager Admin Console.
2. Go to the HDFS service.
3. Click the Charts Library tab.
4. Select DataNodes from the list on the left.
5. Click the Memory tab.
6. Look at the chart titled DataNode JVM Heap Used Distribution. The maximum heap usage usage is the value in the last bucket of that histogram.

Run Hue Document Cleanup

If your cluster uses Hue, perform the following steps (not required for maintenance releases). These steps clean up the database tables used by Hue and can help improve performance after an upgrade.

Important: The clean-up steps only deletes the unsaved documents and workflows. Saved data and information is not cleaned up. Cloudera recommends that you take a backup of your databases before starting the clean-up activity.

1. Back up your database before starting the cleanup activity.
2. Check the saved documents such as Queries and Workflows for a few users to prevent data loss.
3. Connect to the Hue database. See Hue Custom Databases in the Hue component guide for information about connecting to your Hue database.
4. Check the size of the desktop_document, desktop_document2, oozie_job, beeswax_session, beeswax_savedquery and beeswax_queryhistory tables to have a reference point. If any of these have more than 100k rows, run the cleanup.

```
select count(*) from desktop_document;
select count(*) from desktop_document2;
select count(*) from beeswax_session;
select count(*) from beeswax_savedquery;
select count(*) from beeswax_queryhistory;
select count(*) from oozie_job;
```

5. SSH in to an active Hue instance as a root user.

6. If you are upgrading from CDH 5.x or 6.x to CDP, then follow the below steps:
   a. Download the Hue cleanup scripts by using one of the commands:

      ```
      git clone https://github.com/cmconner156/hue_scripts.git /opt/cloudera/hue_scripts
      ```

      or

      ```
      wget -qO- -O /tmp/hue_scripts.zip https://github.com/cmconner156/hue_scripts/archive/master.zip && unzip -d /tmp /tmp/hue_scripts.zip
      mv /tmp/hue_scripts-master /opt/cloudera/hue_scripts
      ```

      Alternatively, you can contact Cloudera Support for assistance.
   b. Run the script as the root user:

      ```
      DESKTOP_DEBUG=True /opt/cloudera/hue_scripts/script_runner hue_desktop_document_cleanup --keep-days 30 --cm-managed
      ```

      (Optional) Specify DESKTOP_DEBUG=True if you want to log information for troubleshooting purposes. Alternatively, you can view the logs from the following location: /var/log/hue/hue_desktop_document_cleanup.log. The first run can typically take around 1 minute per 1000 entries in each table.
   c. Check the size of the desktop_document, desktop_document2, oozie_job, beeswax_session, beeswax_savedquery and beeswax_queryhistory tables and confirm they are now smaller.

      ```
      select count(*) from desktop_document;
      select count(*) from desktop_document2;
      select count(*) from beeswax_session;
      select count(*) from beeswax_savedquery;
      select count(*) from beeswax_queryhistory;
      select count(*) from oozie_job;
      ```
   d. If the hue_scripts script has run successfully, the table size should decrease, and you can now set up a cron job for scheduled cleanups.
   e. Copy the wrapper script for cron by running the following command:

      ```
      cp /opt/cloudera/hue_scripts/hue_history_cron.sh /etc/cron.daily
      ```
   f. Specify the cleanup interval in the --keep-days property in the hue_history_cron.sh file as shown in the following example:

      ```
      ${SCRIPT_DIR}/script_runner hue_desktop_document_cleanup --keep-days 120
      ```

      In this case, the data will be retained in the tables for 120 days.
   g. Change the permissions on the script so only the root user can run it.

      ```
      chmod 700 /etc/cron.daily/hue_history_cron.sh
      ```

7. If you are upgrading from a previous CDP release, the follow the below steps:
   a. Change to the Hue home directory:

      ```
      cd /opt/cloudera/parcels/CDH/lib/hue
      ```

   b. Run the following command as the root user:

      ```
      ./build/env/bin/hue desktop_document_cleanup --keep-days X--cm-managed
      ```

      The --keep-days property is used to specify the number of days for which Hue will retain the data in the backend database.

      (Optional) Specify DESKTOP_DEBUG=True if you want to log information for troubleshooting purposes. Alternatively, you can view the logs from the following location: /var/log/hue/hue_desktop_document_cleanup.log.

      For example:

      ```
      DESKTOP_DEBUG=True ./build/env/bin/hue desktop_document_cleanup --keep-d
      ays 90 --cm-managed 90
      ```

      In this case, Hue will retain data for the last 90 days.

      The first run can typically take around 1 minute per 1000 entries in each table, but can take much longer depending on the size of the tables.

   c. Check the size of the desktop_document, desktop_document2, oozie_job, beeswax_session, beeswax_savedquery and beeswax_queryhistory tables and confirm they are now smaller.

      ```
      select count(*) from desktop_document;
      select count(*) from desktop_document2;
      select count(*) from beeswax_session;
      select count(*) from beeswax_savedquery;
      select count(*) from beeswax_queryhistory;
      select count(*) from oozie_job;
      ```

   d. If any of the tables are still above 100k in size, run the command again while specifying less number of days this time. For example, 60 or 30.

      Note: The optimal number of documents that can be stored in a table is less than or equal to 30,000. Consider this number while specifying the cleanup interval.

Troubleshooting

If the script fails to find the HUE_SERVER process directory, then you may see the following error: KeyError: 'HUE_CONF_DIR'.

To resolve this issue, set the following environment variable before running the script:

```
export HUE_CONF_DIR=`ls -1dtr /var/run/cloudera-scm-agent/process/*HUE_SERVE
R | tail -1`
```

If the script cannot decrypt the Hue database password from the configuration files, then you may see the following error:

```
settings DEBUG DESKTOP_DB_TEST_USER SET: hue_test
Error: Password not present
...
 File "/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p4099.4889921/lib/hue/de
sktop/core/src/desktop/lib/conf.py", line 721, in coerce_password_from_scrip
t
 raise subprocess.CalledProcessError(p.returncode, script)
```

```
subprocess.CalledProcessError: Command '/var/run/cloudera-scm-agent/process
/5260-hue-HUE_SERVER/altscript.sh sec-5-password' returned non-zero exit sta
tus 1
```

To resolve this issue, specify the Hue database password through an environment variable:

```
export HUE_IGNORE_PASSWORD_SCRIPT_ERRORS=1
export HUE_DATABASE_PASSWORD=[***PASSWORD***]
```

Check Oracle Database Initialization

If your cluster uses Oracle for any databases, before upgrading from CDH 5 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.

Step 6: Access Parcels

Steps to access the Parcels required to install Cloudera Runtime.

Parcels contain the software used in your CDP Private Cloud Base clusters. If Cloudera Manager has access to the public Internet, Cloudera Manager automatically provides access to the latest version of the Cloudera Runtime 7.x.x Parcels directly from the Cloudera download site.

If Cloudera Manager does not have access to the internet, you must download the Parcels and set up a local Parcel repository. See Configuring a Local Parcel Repository on page 261. Enter the URL of your repository using the steps below.

If you want to upgrade to a different version of Cloudera Runtime 7.x.x, select the cluster version at the top of this page, and perform the following steps to add the Parcel URL:

Note: For a full list of Parcel URLs for older versions, see Cloudera Runtime Download Information.

To add a new Parcel URL:

1. Log in to the Cloudera Manager Admin Console.
2. Click Parcels from the left menu.
3. Click Parcel Repositories & Network Settings.
4. In the Remote Parcel Repository URLs section, click the "+" icon and add the URL for your Parcel repository.
5. Click Save & Verify Configuration. A message with the status of the verification appears above the Remote Parcel Repository URLs section. If the URL is not valid, check the URL and enter the correct URL.
6. After the URL is verified, click Close.
7. Locate the row in the table that contains the new Cloudera Runtime parcel and click the Download button. If the parcel does not appear on the Parcels page, ensure that the Parcel URL you entered is correct.
8. After the parcel is downloaded, click the Distribute button.

   Wait for the parcel to be distributed. Cloudera Manager displays the status of the Cloudera Runtime parcel distribution.
9. Click the Cloudera Manager logo to return to the home page.

Step 7: Configure Streams Messaging Manager

Additional steps

If your cluster uses Streams Messaging Manager, you need to update database related configuration properties and configure the streamsmsgmgr user's home directory.

1. Stop the Streams Messaging Manager Service:
   a. In Cloudera Manager, select the Streams Messaging Manager service.
   b. Click Actions Stop .
   c. Click Stop on the next screen to confirm.

      When you see a Finished status, the service has stopped.
   d. Click Close.
2. Configure database related properties:

   Note: You can skip this step if you have already configured database related properties during the Cloudera Manager upgrade.

   a. In Cloudera Manager, select the Streams Messaging Manager service.
   b. Go to Configuration.
   c. Find and configure the following properties:
      - Streams Messaging Manager Database User Password
      - Streams Messaging Manager Database Type
      - Streams Messaging Manager Database Name
      - Streams Messaging Manager Database User
      - Streams Messaging Manager Database Host
      - Streams Messaging Manager Database Port
   d. Click Save Changes.
3. Change the streamsmsgmgr user's home directory:
   a. Log in to the Streams Messaging Manager host.

      ```
      ssh [MY_STREAMS_MESSAGING_MANAGER_HOST]
      ```
   b. Change the streamsmsgmgr user's home directory to /var/lib/streams_messaging_manager.

      Rhel-compatible:

      ```
      usermod -d /var/lib/streams_messaging_manager -m streamsmsgmgr
      ```

Step 8: Configure Schema Registry

Steps to update database-related configurations for Schema Registry

If your cluster uses Schema Registry, you need to update database related configuration properties.

1. Configure database related properties:

   Note: You can skip this step if you have already configured database related properties during the Cloudera Manager upgrade.

   a. In Cloudera Manager, select the Schema Registry service.
   b. Go to Configuration.
   c. Find and configure the following properties:
      - Schema Registry Database User Password
      - Schema Registry Database Type
      - Schema Registry Database Name
      - Schema Registry Database User
      - Schema Registry Database Host
      - Schema Registry Database Port
   d. Click Save Changes.

If your cluster uses Schema Registry, you need to update database related configuration properties.

1. Configure database related properties:

   Note: You can skip this step if you have already configured database related properties during the Cloudera Manager upgrade.

   a. In Cloudera Manager, select the Schema Registry service.
   b. Go to Configuration.
   c. Find and configure the following properties:
      - Schema Registry Database User Password
      - Schema Registry Database Type
      - Schema Registry Database Name
      - Schema Registry Database User
      - Schema Registry Database Host
      - Schema Registry Database Port
   d. Click Save Changes.

Step 9: Enter Maintenance Mode

You can enable Maintenance Mode to avoid unnecessary alerts during the upgrade.

To avoid unnecessary alerts during the upgrade process, enter maintenance mode on your cluster before you start the upgrade. Entering maintenance mode stops email alerts and SNMP traps from being sent, but does not stop checks and configuration validations. Be sure to exit maintenance mode when you have finished the upgrade to re-enable Cloudera Manager alerts. More Information.

On the Home > Status tab, click the actions menu next to the cluster name and select Enter Maintenance Mode.

Step 10: Run the Upgrade Cluster Wizard

The Upgrade Wizard manages the upgrade of your Cloudera Runtime software. The Upgrade Wizard is not used for upgrades to Service Packs or Hotfixes.

Important: You have selected an upgrade to Cloudera Runtime Service Pack 1 (7.1.7.1000). The upgrade process for this does not use the Upgrade Wizard if and only if the current Cluster is it's own GA release. Skip the steps on this page and continue with the steps in following document: Upgrading to a Service Pack.

Note: Not all combinations of Cloudera Manager and Cloudera Runtime are supported. Ensure that the version of Cloudera Manager you are using supports the version of Cloudera Runtime you have selected. See Cloudera Manager support for Cloudera Runtime, CDH, and Cloudera Data Services on premises

Minimum Required Role: Cluster Administrator (also provided by Full Administrator)

1. Log in to the Cloudera Manager Admin Console.
2. Ensure that you have completed the steps to add the Parcel URL in Cloudera Manager. See Step 6: Access Parcels.
3. Ensure that all services in the cluster that are being upgraded are running and in good health.
4. Click the Actions menu and select Upgrade Cluster.

   The Getting Started screen of the Upgrade Wizard displays.
5. Click the Upgrade to Version: drop-down and select the version of Cloudera Runtime for your upgrade.

   The wizard now runs several checks to make sure that your cluster is ready for upgrade. You must resolve any reported issues before continuing.
6. The Install Services section displays any additional services that you need to install to upgrade your cluster.

   If you are upgrading a cluster that has the Hive service, you will be prompted to add the Tez, Zookeeper, Hive on Tez, and YARN QueueManager services.

   Warning: When you add Hive-on-Tez service, the Assign Roles page displays. You must ensure that the number of HiveServer2 roles present in the Hive service before the upgrade are included when the Assign Roles page displays. (You can verify this by opening the Cloudera Manager Admin Console Home page in a new browser tab, and going to the Instances tab in the Hive service.) If the number of HiveServer2 roles is not the same, the cluster upgrade will fail and the cluster will be unusable. If your upgrade fails, please contact Cloudera Support.

   You must also select the same hosts for the HiveServer2 roles that were used before the upgrade. If you choose other hosts you must regenerate the keytabs for those hosts. See Managing Kerberos credentials using Cloudera Manager.

7. The Sentry service is replaced by Apache Ranger in CDP Private Cloud Base. If the cluster has the Sentry service installed, you can migrate to Apache Ranger.

   The Apache Ranger service depends on the ZooKeeper and Solr services. The upgrade wizard display buttons for installing several dependent services that are required for Apache Ranger. If your cluster does not include these services, buttons will appear to install them.

   Note: The Solr service used by the Apache Ranger service is a separate, dedicated service. If you have other instances of the Solr service, ensure that these services have configurations that do not overlap. Cloudera Manager configures the following values by default for the Solr service dedicated to Apache Ranger:

   - ZooKeeper Znode: /solr-infra
   - HDFS Data Directory: /solr-infra
   - Solr Data Directory: /var/lib/solr-infra
   - Solr Server Log Directory: /var/log/solr-infra
   - Solr HTTP Port: 8993
   - Solr HTTPS Port: 8995
   - Deploy Directory: /etc/solr-infra
   - Ranger Policy Cache Directory: /var/lib/ranger/solr-infra/policy-cache
   - Ranger DFS Audit Path: ${ranger_base_audit_url}/solr-infra
   - Ranger Audit DFS Spool Dir: /var/log/solr-infra/audit/hdfs/spool
   - Ranger Audit Solr Spool Dir: /var/log/solr-infra/audit/solr/spool

   a. Follow the steps for Transitioning the Sentry service to Apache Ranger on page 16 before continuing.

   b. If the cluster does not already have the ZooKeeper service, click the Add ZooKeeper Service button.

   The Assign Roles page displays with the role assignment for the ZooKeeper service. You can keep the assigned host or assign the role to a different host.

   c. Click Continue.

   The Review Changes screen displays where you can change the default configurations.

   d. Click Continue.

   The upgrade wizard resumes.

   e. Click the Add Solr Service button to install the Solr service dedicated to Apache Ranger.

   The Assign Roles page displays with the role assignment for the Solr service. You can keep the assigned host or assign the role to a different host.

   f. Click Continue.

   The Review Changes screen displays where you can change the default configurations.

   g. Click Continue.

   The upgrade wizard resumes.

   h. Click the Add Ranger Service button

   The Assign Roles page displays with the role assignment for the Ranger service.

   i. Assign the following Ranger roles to cluster hosts:

   - Ranger Admin -- you must assign this role to the host you specified when you set up the Ranger database.
   - Ranger Usersync
   - Ranger Tagsync

   j. In Setup Database, update the Ranger database parameters:

   - Ranger Database Type - Choose either MySQL, PostgreSQL, or Oracle.
   - Ranger Database Host - enter the hostname where the Ranger database is running.
   - Ranger Database Name - enter the database name created for Ranger.
   - Ranger Database User - enter the user created to connect Ranger database.

- Ranger Database User Password - enter the password you created when you created the Ranger database and the user rangeradmin.

k. The Ranger Review Changes screen displays. Review the configurations and make any necessary changes. You must provide values for the following:

- Ranger Admin User Initial Password – choose a password.
- Ranger Usersync User Initial Password – choose a password.
- Ranger Tagsync User Initial Password – choose a password.
- Ranger KMS Keyadmin user initial Password – choose a password.
- Ranger Admin Max Heapsize – set the default value instead of minimum value by clicking the curved blue arrow.
- Ranger Tagsync Max Heapsize – set the default value instead of minimum value by clicking the curved blue arrow.
- Ranger Usersync Max Heapsize – set the default value instead of minimum value by clicking the curved blue arrow.
- If enabling Ranger TLS, see Configure TLS/SSL for Ranger in a manually configured TLS/SSL environment on page 181.

l. Update Auth-To-Local Rule in Hdfs.

IF the Additional Rules to Map Kerberos Principals to Short Names (hadoop.security.auth_to_local) configs have been updated, THEN you must:

1. Update the Additional Rules to Map Kerberos Principals to Short Names config to include the following rules for Ranger & Ranger KMS services principals before upgrade.

```
RULE:[2:$1@$0](rangeradmin@<REALM>)s/(.*)@<REALM>/ranger/
RULE:[2:$1@$0](rangertagsync@<REALM>)s/(.*)@<REALM>/rangertagsync/
RULE:[2:$1@$0](rangerusersync@<REALM>)s/(.*)@<REALM>/rangertagsync/
RULE:[2:$1@$0](rangerkms@<REALM>)s/(.*)@<REALM>/keyadmin/
```

2. Append these rules to the existing ones getting used.
3. Custom rules syntax may be applied to these rules. Make sure the principals are always mapped to the above-provided user names.

m. If you have a workload Solr service besides the Solr instance dedicated to the Ranger service (infra-solr) in your upgraded cluster, you must add its name to Knox as a custom service parameter for both Knox Simplified Topology Management - cdp-proxy and Knox Simplified Topology Management - cdp-proxy-api. This

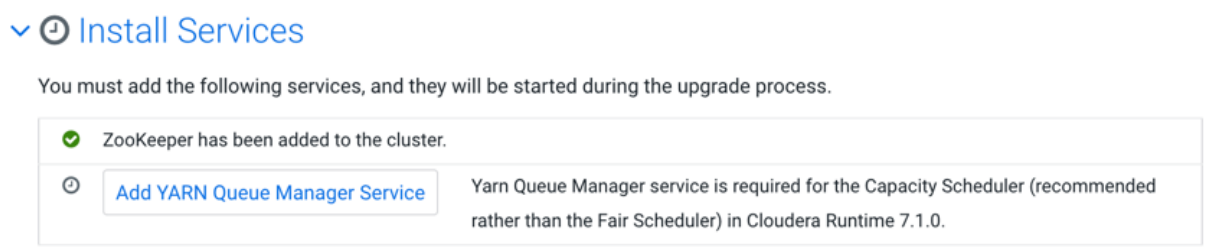
is to avoid Knox forwarding requests to the wrong Solr instance. If the only Solr instance in your cluster is the one dedicated to Ranger, skip this step.

Provide the workload Solr service name in the SOLR:discovery-service-display-name=*[***WORKLOAD-SOLR-SERVICE-NAME***]* format, replacing *[***WORKLOAD-SOLR-SERVICE-NAME***]* with the name of the workload Solr service.

For example, SOLR:discovery-service-display-name=*SOLR-1*:

For more information, see Add custom service parameter to descriptor in *Apache Knox Authentication*.

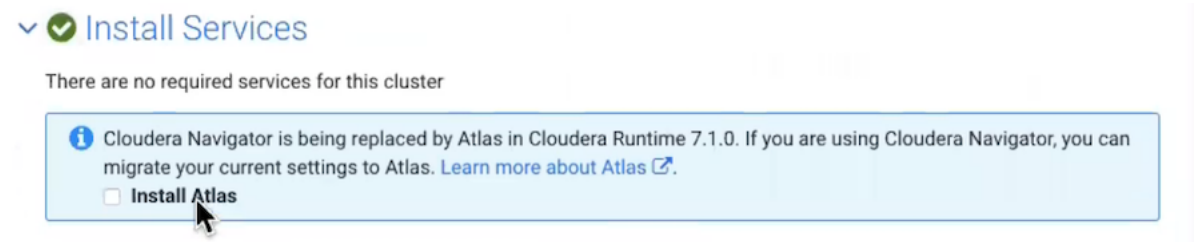
8. If your cluster does not have the YARN Queue Manager, installed, a button will appear to add the YARN Queue Manager service because it is required for the Capacity Scheduler, which is the supported scheduler.



The first step of Adding YARN Queue Manager Service is to copy scheduler settings. For more information about how to transition from Fair Scheduler to Capacity Scheduler, see Fair Scheduler to Capacity Scheduler transition on page 162.

9. Enable Atlas install.

If the cluster being upgraded was running Navigator, the upgrade wizard shows a note recommending that you enable Atlas in the new cluster. Check the Install Atlas option.



10. Install Atlas dependencies.

The wizard steps through the installation for Atlas' dependencies, assuming these services haven't already been included in the installation:

- ZooKeeper. Assign one or more hosts for the ZooKeeper role.
- HDFS. Already included in the installation.
- Kafka. Select the optional dependency of HDFS. Atlas requires configuring the Broker service only, not MirrorMaker, Connect, or Gateway.
- HBase. Atlas requires configuring HBase Master and RegionServers only, not REST or Thrift Server. Assign a Master role on at least one host. Assign RegionServers to all hosts.
- Solr. Assign a host for the Solr Server role. Set the Java Heap Size of Solr Server in Bytes property to 12 GB (to support the migration operation).

For recommendations on where in the cluster to install the service roles, see Runtime Cluster Hosts and Role Distribution.

11. Click Add Atlas Service. The wizard steps through choosing a host and setting migration details.
    - Set the host for the Atlas server roles and click Continue.

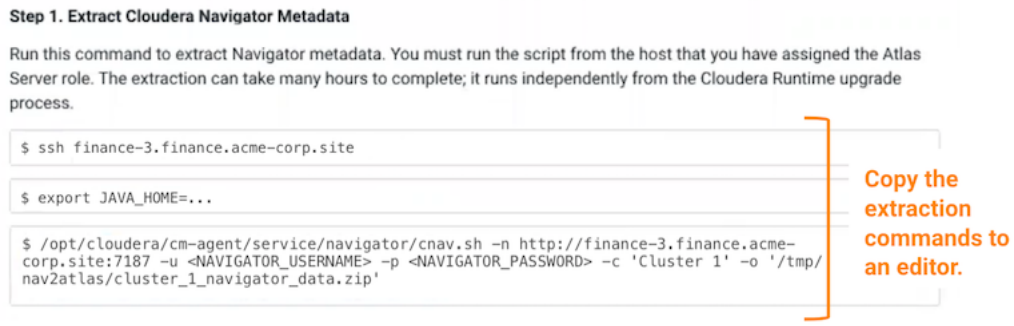
      Tip: Remember this host as you'll need to SSH to it later to trigger the content migration from Navigator.
    - The Atlas Migrate Navigator Data screen displays.

      This screen contains migration commands that are customized to your environment. When you fill in the output file paths, the command text changes to incorporate your settings.

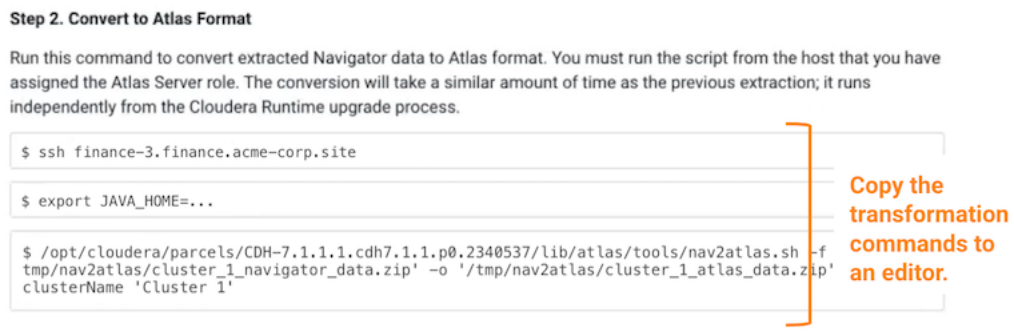
      a. Set migration data-staging locations.

      The migration process creates two data files on the local file system on the host where Atlas is installed. Make sure there is enough disk space to hold these files; see Estimating the time and resources needed for transition on page 24.

      b. Copy the extraction command text to an editor.

      c. Copy the transformation command text to an editor.

      Important: While running the nav2atlas.sh script, make sure that the CDH cluster name does not contain whitespaces or hyphen.

      For example:

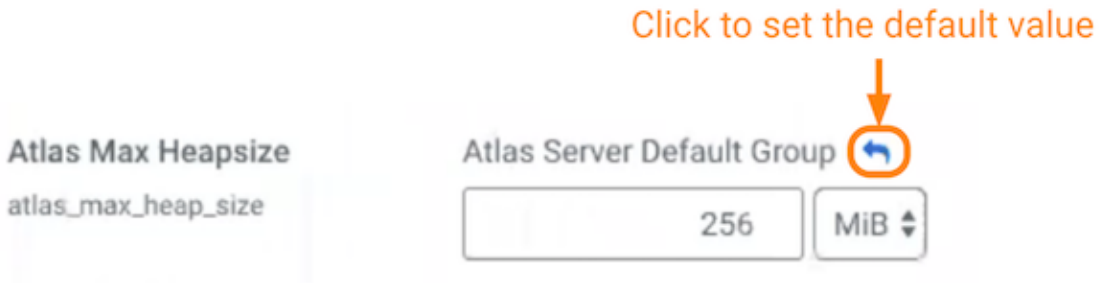
      _# /opt/cloudera/parcels/CDH/lib/atlas/tools/nav2atlas/nav2atlas.sh -f '/tmp/cluster_navigator_data.zip' -o '/root/nav2atlas/cluster_atlas_data.zip' -clusterName ‘Demo_Big_Data'

      d. Confirm the output file location. This is the location where Atlas will look for the content to import. Make sure it matches the location you plan to use for the output of the transformation command.

      e. Click Continue.
    - The Atlas Enable Migration Mode screen displays. Review the Atlas Safety Valve content and click Continue.

      After the migration is complete, you will manually remove these settings to start Atlas in normal operation.
    - The Atlas Review Changes screen displays. Review the configurations and make any necessary changes.You must provide a value for the following:
      - Admin Password – choose a password for the preconfigured admin user.

- Atlas Max Heapsize – set the max heapsize to the default value by clicking the curved blue arrow. If you plan to migrate content from Cloudera Navigator to Atlas, consider setting the heapsize to 16 GB.



- Click Continue.

To complete the Navigator-to-Atlas migration outside of the Cloudera Runtime upgrade, see Transitioning Navigator data using customized scripts on page 29.

12. The Other Tasks section lists other tasks or reminders to note before continuing. Select the option to confirm that you understand before continuing.

Note: You may need to perform some additional steps for clusters with Apache HBase installed and when transitioning from Fair Scheduler to Capacity Scheduler.

- HBase: See Checking Apache HBase on page 40.
- Transitioning from Fair Scheduler to Capacity Scheduler: See Fair Scheduler to Capacity Scheduler transition on page 162;

Important: Post upgrade, for large clusters, the restart of services may fail with the error: Command aborted because of exception: Command timed-out after 150 seconds. If you encounter this error, then perform the workaround steps mentioned in the KB article, and resume the upgrade procedure.

13. The Inspector Checks section displays sever inspectors you must run before continuing. If these inspectors report errors, you must resolve those before continuing.

- Click the Show Inspector Results button to see details of the inspection.
- Click the Run Again button to verify that you have resolved the issue.
- If you are confident that the errors are not critical, select Skip this step. I understand the risks..

The Inspector Checks section includes the following inspectors:

- Host Inspector
- Service Inspector

Note: If the Hive service is present in the cluster, the Inspector Checks include a Validate Hive Metastore schema step. This check may return a "green" result but the validation may actually contain failures tagged with the WARN label. You should look through the inspector results for Hive and correct any failures before continuing with the upgrade.

Run these inspectors and correct any reported errors before continuing.

14. The Database Backup section asks you to verify that you have completed the necessary backups. Select Yes, I have performed these steps.
15. Click Continue. (The Continue button remains greyed out until all upgrades steps are complete and all warnings have been acknowledged.)
16. Click Continue again to shut down the cluster and begin the upgrade.

The Upgrade Cluster Command screen opens and displays the progress of the upgrade.

17. When the Upgrade steps are complete, click Continue.

The Summary page opens and displays any additional steps you need to complete the upgrade.

18. Click Continue.

Fair Scheduler to Capacity Scheduler transition

You must transition from Fair Scheduler to Capacity Scheduler when upgrading your cluster to Cloudera. The transition process involves automatically converting certain Fair Scheduler configuration to Capacity Scheduler configuration prior to the upgrade and manual fine tuning after the upgrade.

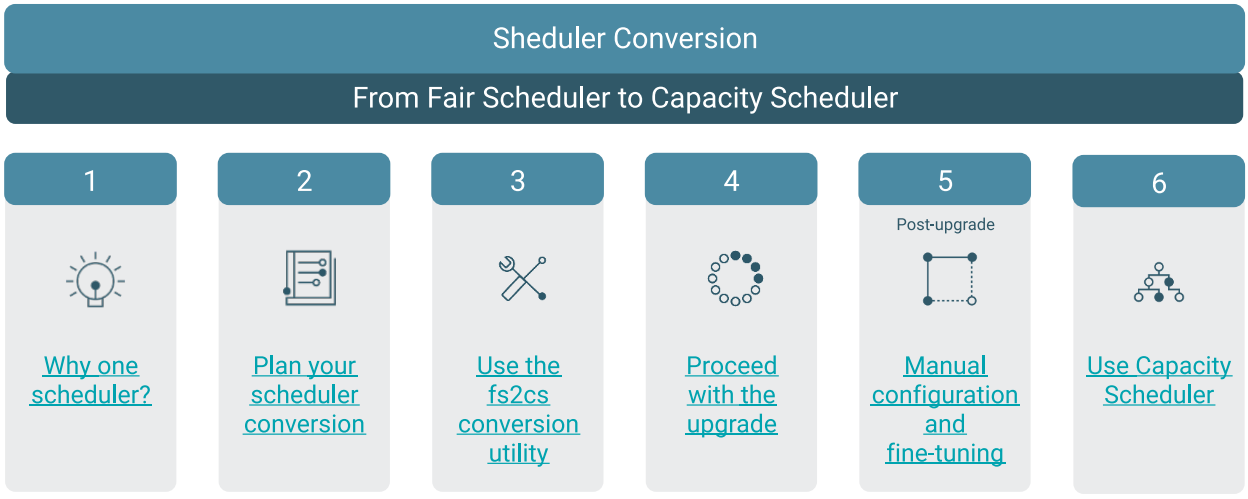
In Cloudera, Capacity Scheduler is the default and supported scheduler. You have to transition from Fair Scheduler to Capacity Scheduler when upgrading from CDH to Cloudera Base on premises.

The scheduler transition process includes migrating the YARN settings from Fair Scheduler to Capacity Scheduler:

1. Preparing for cluster upgrade: As using the Upgrade Cluster Wizard in Cloudera Manager you have to add the necessary services before you can start upgrading your cluster. When you add the YARN QueueManager Service you have to copy your scheduler settings. That is when you can use the fs2cs conversion utility to automatically convert Fair Scheduler into Capacity Scheduler as a part of the Upgrade Wizard in Cloudera Manager.
2. Upgrade the cluster to Cloudera Base on premises.
3. Post-upgrade: Manually configure and fine-tune the scheduler after the upgrade is completed.

Important: The features of Capacity Scheduler are not the same as the features of Fair Scheduler. Hence, the fs2cs conversion utility cannot convert every Fair Scheduler configuration into a Capacity Scheduler configuration. After the automatic conversion and once the upgrade is completed, you must manually tune the scheduler configurations to ensure that the resulting scheduling configuration fits your organization's internal goals and SLAs.

For more information, click on the step that interests you:



Plan your scheduler transition

Before starting the scheduler transition, you must learn about what Fair Scheduler configuration can be converted into a Capacity Scheduler configuration prior to the upgrade, what configuration requires manual configuration and fine-tuning.

The features of Capacity Scheduler are not exactly the same as the features of Fair Scheduler. Hence, the fs2cs conversion utility cannot convert every Fair Scheduler configuration into a Capacity Scheduler configuration. You must learn about what properties are auto-converted and what requires manual configuration. In addition, there are Fair Scheduler features that do not have an equivalent feature in Capacity Scheduler.

Scheduler transition limitations

There are some hard limitations on converting a Fair Scheduler configuration into a Capacity Scheduler configuration as these two schedulers are not equivalent. Learning about these major limitations can help you understand the challenges you might encounter after the scheduler transitions.

The features and configurations of Capacity Scheduler differ from the features and configurations of Fair Scheduler resulting in scheduler transition limitations. These limitations sometimes can be overcome either by manual configuration, fine-tuning or some trial-and-error, but in many cases there is no workaround.

Note: This is not a complete list. It only contains the scheduler transition limitations that most commonly cause issues.

Static and dynamic leaf queues cannot be created on the same level

If you have a parent queue defined in capacity-scheduler.xml file with at least a single leaf queue, it is not possible to dynamically create a new leaf under this particular parent.

Resolved: Weight mode and Dynamic Auto Child Creation is supported from Cloudera Runtime 7.1.6. In weight mode there can be static and dynamic leaf queues on the same level.

Placement rules and mapping rules are different

Depending on your target cluster version you will encounter different placement rules limitations. For more details, see *Placement rules transition*.

The capacity value of dynamic queues is fixed

In Fair Scheduler, fair shares are recalculated each time a new queue is created. In contrast, Capacity Scheduler assigns a predefined percentage value for dynamically created queues.

This predefined percentage can be changed, but it is fixed until the scheduler is reconfigured. Once this value reaches 100, the next dynamic queue will be created with the value 0. For example, if the value is set to 25.00, then the fifth queue under the same parent will have a capacity of 0.

The following is an example of how you can convert the Fair Scheduler queue weights to Capacity Scheduler queue capacity (percentage relative to its parents) :

Table 4: Weight conversion example

Queue Path	Weight	Capacity Scheduler equivalent (capacity) yarn.scheduler.capacity.<*QUEUE-PATH*>.capacity
root	1	100%
root.default	10	25%
root.users	30	75%
root.users.alice	1	33.333%
root.users.bob	1	33.333%
root.users.charlie	1	33.334%

In Cloudera Runtime 7.1.5 and lower versions the fs2cs conversion utility ensures that all percentages of direct children under one parent queue add up exactly to 100.000%, as it is demonstrated in the table. For example, all queues under root.users: root.users.alice + root.users.bob + root.users.charlie = 100.000%.

Weights are converted into percentage-based capacities the following way: On queue-level root, there are 2 queues: default and users. Because it is specified as 10 + 30 weights (40 altogether), 1 “unit of weight” is 2.5%. This is why root.default has 25% and root.users has 75% of the capacity. This calculation can be applied to all queue-levels.

Resolved: Weight mode and Dynamic Auto Child Creation is supported from Cloudera Runtime 7.1.6. and the fs2cs conversion utility converts into weight mode by default.

Placement Rules transition

Placement rules transition is part of the Fair Scheduler to Capacity Scheduler transition process. Learn about the limitations of this transition and how you can overcome them.

Placement rules (used in Fair Scheduler) and mapping rules (used in Capacity Scheduler) are very different, therefore auto conversion of placement rules into mapping rules are not possible. You manually have to configure placement rules and mapping rules on the upgrade from CDH to CDP is completed.

The following are the most substantial limitation:

- In Fair Scheduler you can use special placement rules like "default" or "specified" which are completely absent in Capacity Scheduler.
- In Fair Scheduler you can set a "create" flag for every rule. Mapping rules do not support this.
- In Fair Scheduler in case of nested rules the "create" flag is interpreted for both rules. This is not true in Capacity Scheduler.
- If a rule can return a valid queue in Fair Scheduler, it proceeds to the next rule. Capacity Scheduler, on the other hand, returns “root.default”.

For more information see *Fair Scheduler features and conversion details*.

In Cloudera Runtime 7.1.6 and later releases there is a new placement engine that supports a new JSON-based placement rule format. These new placement rules eliminated many previous placement rules limitations caused by the transitioning from Fair Scheduler to Capacity Scheduler. Note that in weight mode more limitations are resolved than in percentage mode.

The new placement engine can be thought of as a superset of Fair Scheduler and Capacity Scheduler placement evaluation logic. This means two things:

- Everything that could be described in Fair Scheduler's <queuePlacementPolicy> section can be converted into Capacity Scheduler with some minor exceptions.
- Full backward compatbility with the old queue-mapping rule format.

The following are the most substantial differences between the old placement and the new placement rules:

- The rules are described in JSON, however, this is transparent to the user in CDP. The generated JSON can be viewed in Cloudera Manager as part of the Capacity Scheduler Configuration Advanced Configuration Snippet (Safety Valve) setting.
- You can configure what should happen when a rule cannot place the application to the target queue. There are three options: proceed ti the next rule, reject the submission, place the application in the default queue.
- New policies (mapping actions) are available: specified, defaultQueue, and reject.
- The create flag was introduced: Non-existing queues are only created dynamically if it is enabled.

The following limitations remains when transitioning from the Fair Scheduler placement rules to the Capacity Scheduler placement rules:

- When using nested placement rules, it is not possible to define two separate create flag.
- Fair Scheduler performs a strict validation whether a rule in the chain is reachable or not. The placement engine in Capacity Scheduler does not perform such a validation.

For more information, see *Auto-converted Fair scheduler properties*, *Fair Scheduler features and conversion details*, and *Managing placement rules*.

The following table shows how the fs2cs conversion utility converts the olda placement rules into the new JSON-based placement rule format. This conversion happens automatically.

Important: You cannot manage the new JSON-based placement rules by directly editing them in the capa city-scheduler.xml configuration file. Instead, you use the YARN Queue Manager UI to manage placement rules. The table is only provided so that you can better understand how the automatic conversion happens.

Table 5: Automatic placement rule conversion

Fait Scheduler placement rule	JSON-based Capacity Scheduler placement rules
`<rule name=”specified” />`	`{ "type":"user", "matches":"*", "policy":"specified", "fallbackResult":"skip" }`
`<rule name=”user” />`	`{ "type":"user", "matches":"*", "policy":"user", "fallbackResult":"skip" }`
`<rule name=”default” />`	`{ "type":"user", "matches":"*", "policy":"defaultQueue", "fallbackResult":"skip" }`

`<rule name=”default” queue=”root.tmp” />`	`{ "type":"user", "matches":"*", "policy":"setDefaultQueue", “value”: “root.tmp”, "fallbackResult":"skip" }, { "type":"user", "matches":"*", "policy":"defaultQueue", "fallbackResult":"skip" }`
	or
	`{ "type":"user", "matches":"*", "policy":"custom", “customPlacement:”root.tmp”, "fallbackResult":"skip" }`

`<rule name=”primaryGroup” />`	`{ "type":"user", "matches":"*", "policy":"primaryGroup", "fallbackResult":"skip" }`
`<rule name=”secondaryGroupExistingQueue” />`	`{ "type":"user", "matches":"*", "policy":"secondaryGroup", "fallbackResult":"skip" }`

Auto-converted Fair Scheduler properties

The fs2cs conversion utility automatically converts certain Fair Scheduler properties into Capacity Scheduler properties. Reviewing the list of auto-converted properties enables you to verify the conversion and plan the manual fine-tuning that requires to be done after the upgrade is completed.

Table 6: Queue resource-quota related features

Property	Description
Pre-created hierarchical queues.	The same queue hierarchy is achieved after conversion.
<weight>	Weight: The steady fair share of a queue. The queue.capacity property will be set with the same ratio.
<maxAMShare>	Maximum AM share: Limits the fraction of the queue's fair share that can be used to run application masters
<maxRunningApps>	Maximum running apps: Limits the number of apps from the queue to run at once
<maxContainerAllocation>	Maximum container allocation: Maximum amount of resources a queue can allocate for a single container.
<schedulingPolicy>	Scheduling policy of a queue (for example, how submitted applications are ordered over time). It is converted with some limitations. For more information, see *Fair Scheduler features and the conversion details*.
<aclSubmitApps> <aclAdministerApps>	ACL settings: List of users and/or groups that can submit apps to the queue or can administer a queue.
maximum-am-resource-percent	Specifies the maximum percentage of resources in the cluster which can be used to run application masters for the queue.
acl_submit_applications	Specifies the ACL which controls who can submit applications to the given queue.
acl_administer_queue	Specifies the ACL which controls who can administer applications in the given queue.
ordering-policy	Specifies the queue ordering policies to FIFO or fair on the given queue.

Table 7: Global scheduling settings

Property	Description
yarn.scheduler.fair.allow-undeclared-pools	Allow undeclared pools. Sets whether new queues can be created at application submission time.
yarn.scheduler.fair.sizebasedweight	Size based weight. Whether to assign shares to individual apps based on their size, rather than providing an equal share to all apps regardless of size.
<queueMaxAppsDefault>	Queue max apps default: Sets the default running app limit for all queues.
<queueMaxAMShareDefault>	Default max AM share: Sets the default AM resource limit for queue.

Property	Description
yarn.scheduler.fair.locality.threshold.node	Locality threshold node: For applications that request containers on particular nodes, the number of scheduling opportunities since the last container assignment to wait before accepting a placement on another node.
yarn.scheduler.fair.locality.threshold.rack	Locality threshold rack: For applications that request containers on particular racks, the number of scheduling opportunities since the last container assignment to wait before accepting a placement on another rack.
yarn.scheduler.fair.max.assign	Maximum assignments: If assignmultiple is true and dynamic.max.assign is false, the maximum amount of containers that can be assigned in one heartbeat.
yarn.scheduler.fair.assignmultiple	Assign multiple: Whether to allow multiple container assignments in one heartbeat.
yarn.resourcemanager.scheduler.monitor.enable	Allows higher-priority applications to preempt lower-priority applications.
yarn.scheduler.capacity.maximum-am-resource-percent	Specifies the maximum percentage of resources in the cluster which can be used to run application masters.

Table 8: Global scheduling settings

Property	Description
yarn.scheduler.fair.allow-undeclared-pools	Allow undeclared pools. Sets whether new queues can be created at application submission time.
yarn.scheduler.fair.sizebasedweight	Size based weight. Whether to assign shares to individual apps based on their size, rather than providing an equal share to all apps regardless of size.
<queueMaxAppsDefault>	Queue max apps default: Sets the default running app limit for all queues.
<queueMaxAMShareDefault>	Default max AM share: Sets the default AM resource limit for queue.
yarn.scheduler.fair.locality.threshold.node	Locality threshold node: For applications that request containers on particular nodes, the number of scheduling opportunities since the last container assignment to wait before accepting a placement on another node.
yarn.scheduler.fair.locality.threshold.rack	Locality threshold rack: For applications that request containers on particular racks, the number of scheduling opportunities since the last container assignment to wait before accepting a placement on another rack.
yarn.scheduler.fair.max.assign	Maximum assignments: If assignmultiple is true and dynamic.max.assign is false, the maximum amount of containers that can be assigned in one heartbeat.
yarn.scheduler.fair.assignmultiple	Assign multiple: Whether to allow multiple container assignments in one heartbeat.
yarn.resourcemanager.scheduler.monitor.enable	Allows higher-priority applications to preempt lower-priority applications.
yarn.scheduler.capacity.maximum-am-resource-percent	Specifies the maximum percentage of resources in the cluster which can be used to run application masters.
<userMaxAppsDefault>	Default maximum running applications.

Property	Description
<user name="..."> <maxRunningApps>...</maxRunningApps></user>	Maximum running applications per user.
yarn.scheduler.fair.user-as-default-queue	Whether to use the username associated with the allocation as the default queue name. Weight mode: This behavior is simulated with a placement rule (in fact, even in Fair Scheduler, this is translated into a placement rule internally): `{ "type":"user", "matches":"*", “parentQueue”: “root”, "policy":"user", “create”: true, "fallbackResult":"skip" }` For information about percentage mode, see *Fair Scheduler features ad conversion details*.

Table 9: Preemption

Property	Description
yarn.scheduler.fair.preemption	Fair Scheduler preemption turned on. After the conversion capacity Scheduler preemption is turned on by default using the default values.
<allowPreemptionFrom>	Per-queue preemption disabled. After the conversion the same queue preemption disabled by default.
yarn.scheduler.fair.waitTimeBeforeKill	Wait time before killing a container
disable_preemption	Disables preemption of application containers submitted to a given queue.

Table 10: Placement rules

Fair Scheduler placement rules	Description	Conversion details
create="false" or "true"	Disable or enable creating a queue dynamically in YARN. This option cannot be specified on the following placement rule policies: • reject • setDefaultQueue • defaultQueue	Weight mode: This flag is fully supported, except for nested rules, where you can define a single “create” flag only. Therefore, “true/false” and “false/true” cannot be set. Relative mode: Partially supported. A managed parent queue must be chosen as a parent. The flag has no effect on regular parent queues.
<rule name="specified"/>	If a user has submitted the application by specifying a queue name (other than the “default” queue), then this rule will be successful. Hence the remaining set of rules won't be executed.	Supported in both weight and percentage mode.
<rule name="primaryGroup"/>	If the submitted user’s(userA) primary group name (groupA) exists, submit to groupA.	The matching policy is called primaryGroup.
<rule name="secondaryGroupExistingQueue" />	If the submitted user’s(userA) secondary group name (groupB) exists, submit to groupB.	The matching policy is called secondaryGroup.

Fair Scheduler placement rules	Description	Conversion details
<rule name="nestedUserQueue">	Depending on the nested rule, this places the job to the following queues: • root.[primaryGroup].[userName] • root.[secondaryGroup].[userName] • root.[queuePath].[userName]	Supported by Capacity Scheduler. The three possible policies are (depending on the outer rule): • primaryGroupUser • secondaryGroupUser • user with a parentQueue set explicitly.
<rule name="default" queue=”qName”/>	Places the application into the default queue called “root.default” or to a user-specific one denoted by the “queue” attribute.	The default rule has a matching policy called defaultQueue. If “root.default” is not the intended default queue, then two approaches are possible: • Use the setDefaultQueue policy to change “root.default”, then apply defaultQueue. • Use the custom policy with the policy string being set to the target queue.

Fair Scheduler features and conversion details

Certain Fair Scheduler properties cannot be auto-converted by the fs2cs conversion utility. Review the list of these properties and if they are supported in Capacity Scheduler and by Queue Manager UI to learn how you can configure them.

Table 11: Queue resource-quota related features

Property	Description	Conversion information
<minResources>	Minimum resources the queue is entitled to.	Partially supported in Capacity Scheduler. Ignored by the fs2cs conversion utility. Not supported by Queue Manager UI.
<maxResources>	Maximum amount of resources that will be allocated to a queue.	There is an equivalent feature in Capacity Scheduler. Ignored by the fs2cs conversion utility. For each queue, max-capacity will be set to 100%. Supported by Queue Manager UI.
<maxChildResources>	Maximum amount of resources that can be allocated to an ad hoc child queue.	There is an equivalent feature in Capacity Scheduler. Ignored by the fs2cs conversion utility. Its value can be two distinct percentages (vcore/memory) or an absolute resources, but the leaf-queue-template only accepts a single percentage. Supported by Queue Manager UI.
<schedulingPolicy>	Scheduling policy of a queue (for example, how submitted applications should be ordered over time). .	There is an equivalent feature in Capacity Scheduler. Manual fine tuning might be necessary. **Note:** if DRF is used anywhere in Fair Scheduler, then the converted configuration utilizes DRF everywhere and it is not possible to place a queue with “Fair” policy under one which has “DRF” enabled. Supported by Queue Manager UI.

Table 12: Queue resource-quota related features

Property	Description	Conversion information
<minResources>	Minimum resources the queue is entitled to.	Partially supported in Capacity Scheduler. Ignored by the fs2cs conversion utility. Not supported by Queue Manager UI.
<maxResources>	Maximum amount of resources that will be allocated to a queue.	There is an equivalent feature in Capacity Scheduler. Ignored by the fs2cs conversion utility. For each queue, max-capacity will be set to 100%. Supported by Queue Manager UI.
<maxChildResources>	Maximum amount of resources that can be allocated to an ad hoc child queue.	There is an equivalent feature in Capacity Scheduler. Ignored by the fs2cs conversion utility. Its value can be two distinct percentages (vcore/memory) or an absolute resources, but the leaf-queue-template only accepts a single percentage. Supported by Queue Manager UI.

Table 13: Global scheduling settings

Property	Description	Conversion information
<user name="..."> <maxRunningApps>...</maxRunningApps></user>	Maximum running apps per user	There is an equivalent feature in Capacity Scheduler. Fine-tuning of the following three properties are required: • Maximum apps per queue • User limit percent • User limit factor Supported by Queue Manager UI.
<userMaxAppsDefault>	Default maximum running apps	Not supported in Capacity Scheduler.
yarn.scheduler.fair.max.assign	Dynamic maximum assign	There is an equivalent feature in Capacity Scheduler. Fine-tuning of the following three properties are required: • yarn.scheduler.capacity.per-node-heartbeat.multiple-assignments-enable • yarn.scheduler.capacity.per-node-heartbeat.maximum-container-assignments • yarn.scheduler.capacity.per-node-heartbeat.maximum-offswitch-assignments Supported by Queue Manager UI.

Property	Description	Conversion information
yarn.scheduler.fair.user-as-default-queue	User as default queue	There is a very similar feature in Capacity Scheduler. Perform the following steps: **1.** Create a queue, such as root.users and enable the suto-create-child-queue setting for it. **2.** Use the following placement rule: "u%user:%user" The following restrictions apply: • It is not possible to have root as a parent for dynamically created queues. • root.users cannot have static leafs, that is, queues that are defined in the capacity-scheduler.xml file. For more information, see the *Placement Rules* table. Supported by Queue Manager UI.

Table 14: Global scheduling settings

Property	Description	Conversion information
yarn.scheduler.fair.max.assign	Dynamic maximum assign	There is an equivalent feature in Capacity Scheduler. Fine-tuning of the following three properties are required: • yarn.scheduler.capacity.per-node-heartbeat.multiple-assignments-enable • yarn.scheduler.capacity.per-node-heartbeat.maximum-container-assignments • yarn.scheduler.capacity.per-node-heartbeat.maximum-offswitch-assignments Not supported by Queue Manager UI.

Property	Description	Conversion information
yarn.scheduler.fair.user-as-default-queue	User as default queue	Relative mode:A placement rule needs to be created. 1. Create a queue, such as "root.users" and enable Dynamic Auto Child Creation for it (make it a Managed Parent Queue). 2. Create the following placement rule: `{ "type":"user", "matches":"*", “parentQueue”: “root.users”, "policy":"user", “create”: true, "fallbackResult":"skip" }` The following limitations apply: • It is not possible to have "root" as a parent for dynamically created queues. • "root.users" queue cannot have static leafs. Those are queues that always exist and are created manually. For information about weight mode see *Auto-converted Fair Scheduler properties*. Supported by Queue Manager UI.

Table 15: Preemption

Property	Description	Conversion information
yarn.scheduler.fair.preemption.cluster-utilization-threshold	The utilization threshold after which preemption kicks in.	There is an equivalent feature in Capacity Scheduler: yarn.resourcemanager.monitor.capacity.preemption.max_ignored_over_capacity. It specifies the resource usage threshold over its configured capacity that a queue must meet before it is eligible for preemption. Supported by Queue Manager UI.
minSharePreemptionTimeout	The number of seconds the queue is under its minimum share before it will try to preempt containers to take resources from other queue.s	Not supported in Capacity Scheduler.
fairSharePreemptionTimeout	The number of seconds the queue is under its fair share threshold before it will try to preempt containers to take resources from other queues.	Partially supported in Capacity Scheduler. This can be achieved by using the following configurations together: • yarn.resourcemanager.monitor.capacity.preemption.natural_termination_factor • yarn.resourcemanager.monitor.capacity.preemption.max_wait_before_kill Supported by Queue Manager UI.

Property	Description	Conversion information
fairSharePreemptionThreshold	The fair share preemption threshold for the queue.	Partially supported in Capacity Scheduler. This can be achieved by using the following configurations together: • yarn.resourcemanager.monitor.capacity.preemption.natural_termination_factor • yarn.resourcemanager.monitor.capacity.preemption.max_wait_before_kill Supported by Queue Manager UI.

Table 16: Placement rules

Fair Scheduler placement rules	Description	Conversion information
create="false" or "true"	Disable or enable creating a queue dynamically in YARN. This option can be specified on all rules.	Partially supported in Capacity Scheduler. Use the Capacity Scheduler Dynamic Queue Mappings policies: • u:%user:[managedParentQueueName].[queueName] • u:%user:[managedParentQueueName].%user • u:%user:[managedParentQueueName].%primary_group • u:%user:[managedParentQueueName].%secondary_group Supported by Queue Manager UI.
<rule name="specified"/>	If a user has submitted the application by specifying a queue name (other than the “default” queue), then this rule will be successful. Hence the remaining set of rules won't be executed.	Not supported in Capacity Scheduler.
<rule name="primaryGroupExistingQueue"/>	If submitted user’s(userA) primary group name (groupA) exists, submit to groupA.	There is an equivalent placement rule in Capacity Scheduler: <value>u:%user:%primary_group</value> Supported by Queue Manager UI.
<rule name="secondaryGroupExistingQueue" />	If submitted user’s(userA) secondary group name (groupA) exists, submit to groupA.	There is an equivalent placement rule in Capacity Scheduler: <value>u:%user:%secondary_group</value> Supported by Queue Manager UI.
<rule name="nestedUserQueue">	Depending on the nested rule, this places the job to the following queues: • root.[primaryGroup].[userName] • root.[secondaryGroup].[userName] • root.[queuePath].[userName]	Not supported in Capacity Scheduler.
<rule name="default" queue=”qName”/>	Fall back policy by which rule will fall back to queue named in the property ‘queue’ or the “default“ queue if no queue property is specified (if all matches fail).	There is an equivalent placement rule in Capacity Scheduler: <value>u:%user:default</value> Supported by Queue Manager UI.

Use the fs2cs conversion utility

You can use the fs2cs conversion utility to automatically convert certain Fair Scheduler configuration to Capacity Scheduler configuration as part of the Upgrade Cluster Wizard in Cloudera Manager.

About this task

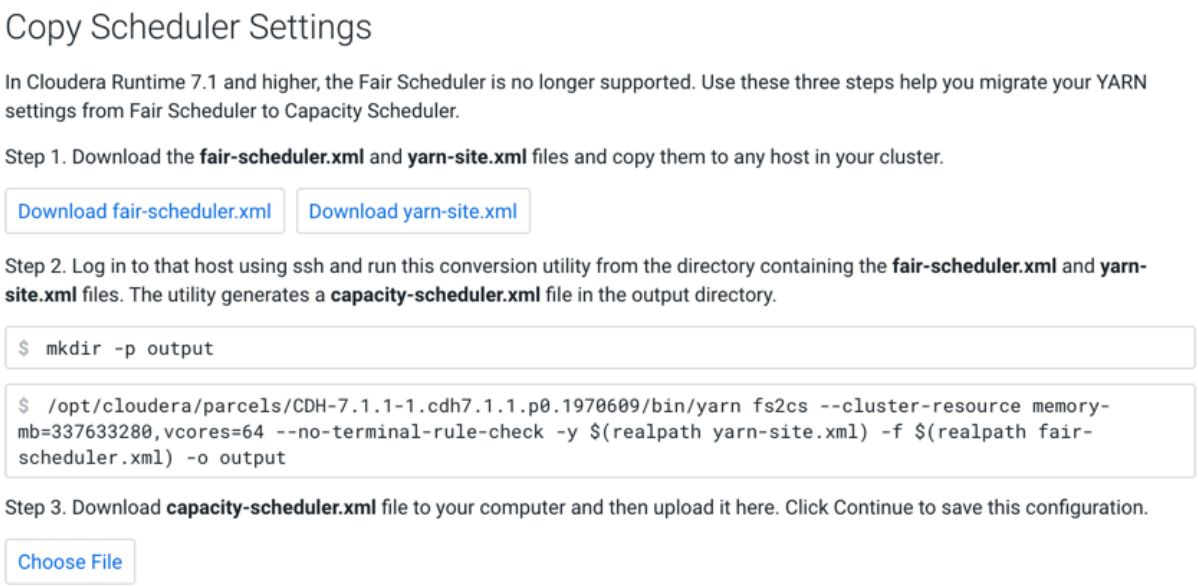
From the Cloudera Base on premises 7.1 release, Cloudera provides a conversion tool, called fs2cs conversion utility. This utility is a CLI application that is part of the yarn CLI command. It generates capacity-scheduler.xml and yarn-site.xml as output files.

Important: The features of Capacity Scheduler are not exactly the same as the features of Fair Scheduler. Hence, the fs2cs conversion utility cannot convert every Fair Scheduler configuration into a Capacity Scheduler configuration. After the automatic conversion and once the upgrade is completed, you must manually tune the scheduler configurations to ensure that the resulting scheduling configuration fits your organization's internal goals and SLAs after conversion.

Before you begin

- Be aware of the Fair Scheduler properties that are auto-converted, those that require manual configuration, and those that do not have an equivalent feature in Capacity Scheduler.
- You must have downloaded and distributed parcels for the target version of CDP.
- In VPC, to use your current Compute Cluster queue configurations in your new installation after the upgrade, you must have manually saved them before starting the update process and then added the configurations to your new installation. Else, your Compute Cluster queue configurations will be lost because the Upgrade Wizard transitions only the queues from your Base Cluster.

  1. In Cloudera Manager, navigate to Host All Hosts .
  2. Find the host with the ResourceManager role and click the YARN ResourceManager role.
  3. Click the Processes tab.
  4. Find and save the fair-scheduler.xml and yarn-site.xml configuration files for future reference.
- Ensure that the configuration is not stale, there is no unsaved changes. Ensure that there is no unsaved changes on the Dynamic Resource Pools view, meaning that the Refresh Dynamic Resource Pools button is inactive.
- Reach the Copy Scheduler Settings part of the upgrade process using the Upgrade Cluster Wizard in Cloudera Manager. That is the first step when you add YARN Queue Manager service:



Procedure

1. Download the Fair Scheduler configuration files from the Cloudera Manager data store:
   a) In the Copy Scheduler Settings window during the upgrade process, click Download fair-scheduler.xml and Download yarn-site.xml to download the fair-scheduler.xml and yarn-site.xml files.
   b) Copy the downloaded configuration files to any host in your cluster.

2. Use the fs2cs conversion utility:
   a) Log in to the host machine where you downloaded the fair-scheduler.xml and yarn-site.xml files using ssh.
   b) Create a new directory to save the capacity-scheduler.xml file that is generated by the fs2cs conversion utility:

```
$ mkdir -p output
```

   c) Use the fs2cs conversion utility to auto-convert the structure of resource pools. Options listed between braces [] are optional:

```
$ yarn fs2cs [--cluster-resource ***VCORES/MEMORY***][--no-terminal-rule
-check] --yarnsiteconfig ***FULL PATH TO YARN-SITE.XML***> [--fsconfi
g ***FULL PATH TO FAIR-SCHEDULER.XML***] --output-directory ***OUTPUT
 PATH*** [--print] [--skip-verification]
```

Important: You have to provide absolute path for the yarn-site.xml and the fair-scheduler.xml configuration file. If only the file names are provided the command fails.

   For example:

```
yarn fs2cs --yarnsiteconfig /home/hadoop/yarn-site.xml --fsconfig /home/
hadoop/fair-scheduler.xml --output-directory /tmp
```

3. Upload the generated Capacity Scheduler configuration files to save the configuration in Cloudera Manager:
   Click Choose File and select the generated capacity-scheduler.xml file to save the configuration.

Note: The configurations in the generated yarn-site.xml output file have to be manually configured using Cloudera Manager Advanced configuration snippet (Safety Valves) once the upgrade is completed.

If the fs2cs conversion utility command fails, check if you provided the correct full path for the yarn-site.xml and the fair-scheduler.xml configuration file.

What to do next

Proceed with the CDP upgrade.

After the upgrade is completed, manually add the yarn-site.xml configurations using Cloudera Manager Advanced configuration snippet (Safety Valves), and tune the configuration generated by the fs2cs conversion utility using Queue Manager UI and Cloudera Manager Advanced configuration snippet (Safety Valves).

CLI options of the fs2cs conversion tool

Before generating the scheduler output files, you must understand the CLI options available for the fs2cs conversion tool. These options help you to complete the scheduler conversion.

Option	Description

-c,--cluster-resource <arg>	Needs to be specified if maxResources is defined as percentages for any queue, otherwise this parameter can be omitted. The cluster resource setting is optional, but it can be necessary in some cases. In Fair Scheduler, you can define the maximum capacity of the queue as a percentage of the total cluster resource. Capacity scheduler does not accept a vector of capacities (work is ongoing under YARN-9936), therefore, percentages are converted to absolute resources. The acceptable formats are:
	`--cluster-resource vcores=10,memory-mb=1024`
	`--cluster-resource 1024 mb,10 vcores`

-d,--dry-run	Performs a dry-run of the conversion. Outputs whether the conversion is possible or not.

Option	Description
-f,--fsconfig <arg>	Absolute path to a valid fair-scheduler.xml configuration file. By default, yarn-site.xml contains the property which defines the path of fair-scheduler.xml. Therefore, the -f / --fsconfig settings are optional.
-h,--help	Displays the list of options
-o,--output-directory <arg>	Output directory for yarn-site.xml and capacity-scheduler.xml files. Must have write permission for the user who is running this script. If -p or --print is specified, the xml files are emitted to the standard output, so the -o / --output-directory is ignored.
-p,--print	If defined, the converted configuration will only be emitted to the console. If -p or --print is specified, the xml files are emitted to the standard output, so the -o / --output-directory is ignored.
-r,--rulesconfig <arg>	Optional parameter. If specified, should point to a valid path to the conversion rules file (property format).
-s, --skip-verification	It does not validate the converted Capacity Scheduler configuration. By default, the utility starts an internal Capacity Scheduler instance to see whether it can start up properly or not. This switch disables this behaviour.
-t,--no-terminal-rule-check	Disables checking whether a placement rule is terminal to maintain backward compatibility with configs that were made before YARN-8967. By default, Fair Scheduler performs a strict check of whether a placement rule is terminal or not. This means that if you use a <reject> rule which is followed by a <specified> rule, then this is not allowed, because the latter is unreachable. However, before YARN-8967, Fair Scheduler was more lenient and allowed certain sequence of rules that are no longer valid. Inside the tool, a Fair Scheduler instance is instantiated to read and parse the allocation file. To have Fair Scheduler accept such configurations, the -t or --no-terminal-rule-c heck argument must be supplied to avoid the Fair Scheduler instance throwing an exception.
-y,--yarnsiteconfig <arg>	Path to a valid yarn-site.xml configuration file.

Option	Description

-c,--cluster-resource <arg>	Needs to be specified if maxResources is defined as percentages for any queue, otherwise this parameter can be omitted. The cluster resource setting is optional, but it can be necessary in some cases. In Fair Scheduler, you can define the maximum capacity of the queue as a percentage of the total cluster resource. Capacity scheduler does not accept a vector of capacities (work is ongoing under YARN-9936), therefore, percentages are converted to absolute resources. The acceptable formats are:
	`--cluster-resource vcores=10,memory-mb=1024`
	`--cluster-resource 1024 mb,10 vcores`

-d,--dry-run	Performs a dry-run of the conversion. Outputs whether the conversion is possible or not.

Option	Description
-f,--fsconfig <arg>	Absolute path to a valid fair-scheduler.xml configuration file. By default, yarn-site.xml contains the property which defines the path of fair-scheduler.xml. Therefore, the -f / --fsconfig settings are optional.
-h,--help	Displays the list of options
-o,--output-directory <arg>	Output directory for yarn-site.xml and capacity-scheduler.xml files. Must have write permission for the user who is running this script. If -p or --print is specified, the xml files are emitted to the standard output, so the -o / --output-directory is ignored.
-p,--print	If defined, the converted configuration will only be emitted to the console. If -p or --print is specified, the xml files are emitted to the standard output, so the -o / --output-directory is ignored.
-pc,-percentage	By default the fs2cs conversion utility converts into weight mode. Using -pc you can change it to relative (percentage) mode. Note that there are some scheduler transition limitations that are resolved in weight mode but not in relative (percentage) mode. Relative mode can be considered the “legacy” mode of Capacity Scheduler, where capacities are expressed in percentages.
-r,--rulesconfig <arg>	Optional parameter. If specified, should point to a valid path to the conversion rules file (property format).
-s, --skip-verification	It does not validate the converted Capacity Scheduler configuration. By default, the utility starts an internal Capacity Scheduler instance to see whether it can start up properly or not. This switch disables this behaviour.
-t,--no-terminal-rule-check	Disables checking whether a placement rule is terminal to maintain backward compatibility with configs that were made before YARN-8967. By default, Fair Scheduler performs a strict check of whether a placement rule is terminal or not. This means that if you use a <reject> rule which is followed by a <specified> rule, then this is not allowed, because the latter is unreachable. However, before YARN-8967, Fair Scheduler was more lenient and allowed certain sequence of rules that are no longer valid. Inside the tool, a Fair Scheduler instance is instantiated to read and parse the allocation file. To have Fair Scheduler accept such configurations, the -t or --no-terminal-rule-check argument must be supplied to avoid the Fair Scheduler instance throwing an exception.
-y,--yarnsiteconfig <arg>	Path to a valid yarn-site.xml configuration file.

Manual configuration of scheduler properties

After upgrading to Cloudera Base on premises, you must manually add the content of the yarn-site.xml output file, and then fine-tune the scheduler configurations using the YARN Queue Manager UI to ensure that the resulting configurations suit your requirements. You can use Cloudera Manager Advanced configuration snippet (Safety Valve) to configure a property that is missing from the YARN Queue Manager UI.

The features of Capacity Scheduler are not exactly the same as the features of Fair Scheduler. Hence, the conversion utility cannot convert every Fair Scheduler configuration into a Capacity Scheduler configuration. Therefore, you must manually tune the scheduler configurations to ensure that the resulting scheduling configuration fits your organization’s internal goals and SLAs after conversion. If needed, further change the scheduler properties in the capacity-scheduler.xml and yarn-site.xml output files generated by the fs2cs conversion utility. For information about the Fair Scheduler properties that are auto-converted by the fs2cs conversion utility, see *Auto-converted Fair Scheduler properties*.

You can configure the properties manually using the YARN Queue Manager UI. If you see any properties that are unavailable in the Queue Manager UI, you can use Cloudera Manager configuration snippet (Safety Valves) to configure them.

Important: You must not use the Queue Manager UI and Cloudera Manager Safety Valves at the same time as safety valves overwrite the configuration set using Queue Manager UI.

Manually add the configurations of yarn-site.xml

Once the upgrade to Cloudera Base on premises is completed, you have to manually add the yarn-site.xml configurations using Cloudera Manager Advanced configuration snippet (Safety Valves).

About this task

The fs2cs conversion utility generates two output files: capacity-scheduler.xml and yarn-site.xml. The capacity-scheduler.xml file can be uploaded using the Upgrade Wizard, but the configurations in the yarn-site.xml have to be manually added once the upgrade is completed.

Before you begin

- Use the fs2cs conversion utility to generate the capacity-scheduler.xml and yarn-site.xml output files.
- Complete the upgrade process.

Procedure

1. In Cloudera Manager, select the YARN service.
2. Click the Configuration tab.
3. Search for yarn-site, and find the YARN Service Advanced Configuration Snippet (Safety Valve) for yarn-site.xml.
4. Copy the content of the yarn-site.xml output file.
5. Click View as XML and paste the copied content.

   The yarn-site.xml file can contain the following configurations:

   - Continuous scheduling enabled/disabled
   - Continuous scheduling interval
   - Preemption
     - Preemption enabled/disabled
     - Wait time before preemption
     - Wait time before next starvation check
   - Assign multiple enabled/disabled
   - Maximum number of heartbeat assignments
   - Locality threshold per node
   - Locality threshold per rack
   - Size-based weight enabled/disabled
   - Resource calculator class
   - Async scheduling enabled/disabled
6. Manually remove the following invalid tags:
   - header
   - configuration tags
   - final tags
   - source tags
7. Click Save Changes.
8. Restart the YARN service.

What to do next

Manually tune the configuration generated by the fs2cs conversion utility using Queue Manager UI and Cloudera Manager Advanced configuration snippet (Safety Valves).

Use YARN Queue Manager UI to configure scheduler properties

After upgrading to Cloudera Base on premises, you must configure the Capacity Scheduler properties using the output files generated by the fs2cs conversion utility. You can configure the properties manually using the YARN Queue Manager UI service.

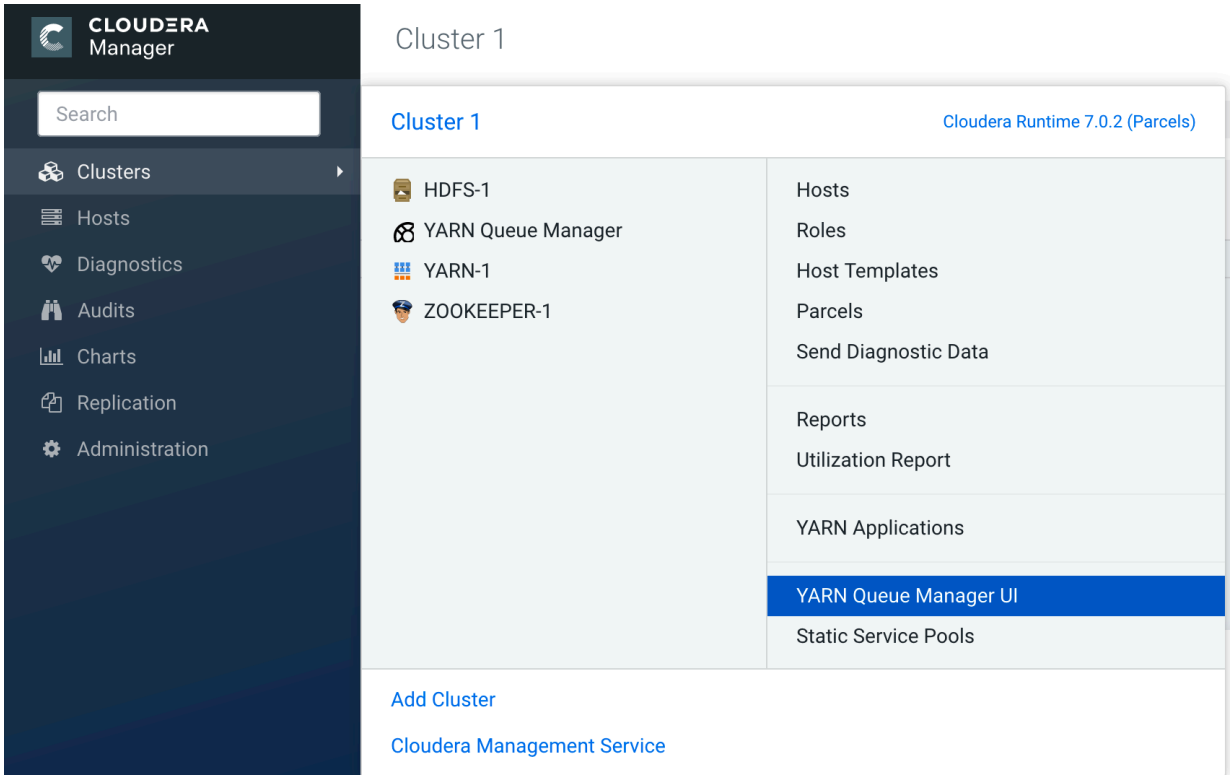
Before you begin

- Use the fs2cs conversion utility to generate the capacity-scheduler.xml and yarn-site.xml output files.
- Complete the upgrade process.
- Identify properties that require manual configuration and can be configured using the Queue Manager UI.

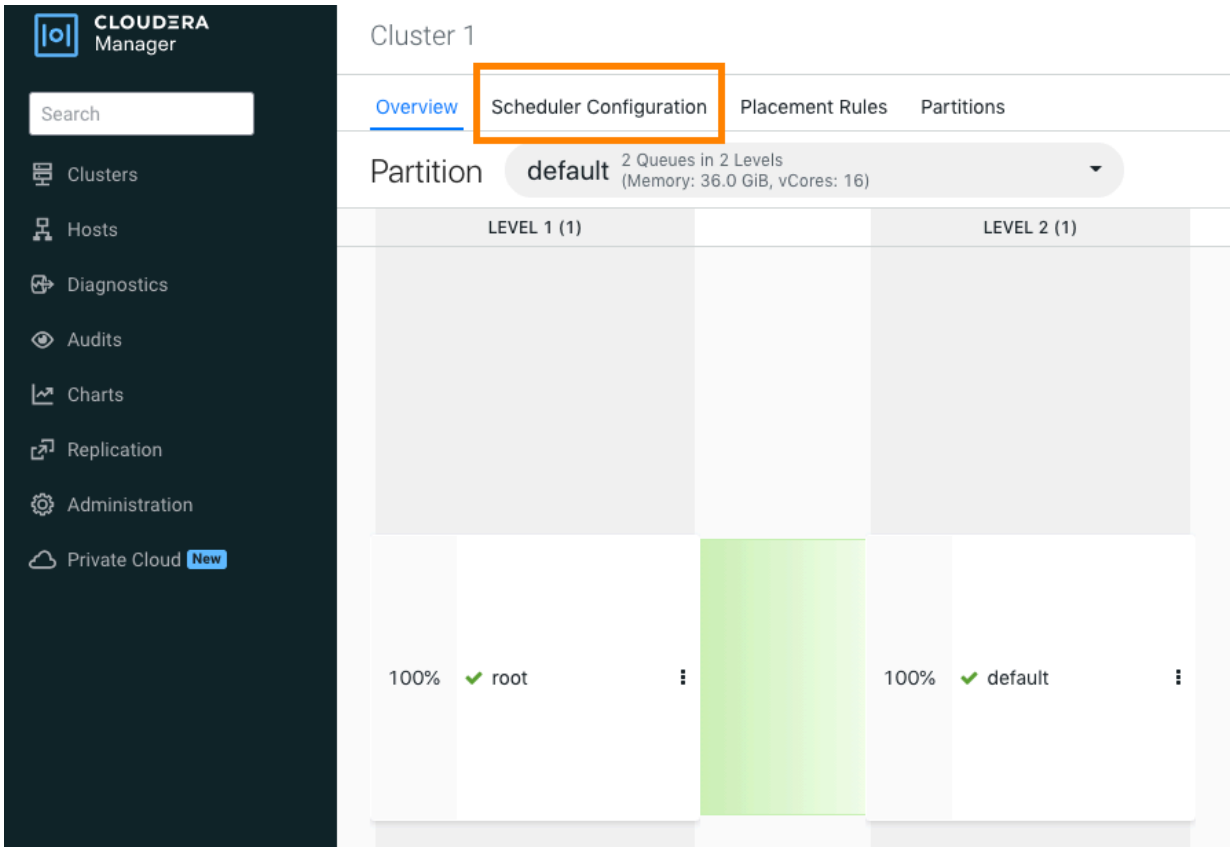
For more information about scheduler properties, see *Fair Scheduler feature and conversion details*.

Procedure

1. In Cloudera Manager, click Clusters and select the YARN Queue Manager UI service.



2. In the YARN Queue Manager window, click the Scheduler Configuration tab.



3. In the Scheduler Configuration window, enter the value of the property and click Save.

Use Cloudera Manager Safety Valves to configure scheduler properties

Certain scheduler properties can neither be converted by the fs2csconversion utility nor be configured using the YARN Queue Manager UI service. After upgrading to Cloudera Base on premises you must manually configure these properties using the Cloudera Manager advanced configuration snippet (Safety Valves).

Before you begin

- Use the fs2cs conversion utility to generate the capacity-scheduler.xml and yarn-site.xml output files.
- Complete the upgrade process.
- Identify the scheduler properties that need to be configured manually and not supported by the Queue Manager UI.

Procedure

1. In Cloudera Manager, select the YARN service.
2. Click the Configuration tab.
3. Search for capacity-scheduler, and find the Capacity Scheduler Configuration Advanced Configuration Snippet (Safety Valve).
4. Click View as XML, and insert the complete capacity-scheduler.xml file, generated by the converter tool.
5. Add the necessary configuration properties.
6. Click Save Changes.
7. Search for yarn-site, and find the YARN Service Advanced Configuration Snippet (Safety Valve) for yarn-site.xml.
8. Click View as XML and add the required configuration in an XML format.

   Optionally, use + and - to add and remove properties.

9. Click Save Changes.
10. Restart the YARN service.

Configure TLS/SSL for Ranger in a manually configured TLS/SSL environment

How to manually configure TSL/SSL for Ranger in a manually configured TLS/SSL environment.

About this task

If you use manual TLS encryption on CDH and you plan to enable TLS for Ranger during the CDP upgrade, you must ensure that the Ranger Admin certificate is imported into the truststore file configuration of all services that support Ranger plugins.

Note:

If you have existing CA-signed certificates for keystores and truststores deployed across all the cluster hosts, then continue to use the same for the Ranger service. In this case you can skip the steps in this topic that create a self-signed certificate. Proceed to Upgrade the Cluster Step 4 : Add Ranger service from CM Upgrade Wizard and update the TLS/SSL configurations.

Procedure

1. Create a keystore file for Ranger Admin.

```
mkdir -p /etc/security/serverKeys

${JAVA_HOME}/bin/keytool -genkeypair -alias {ALIAS}
-keyalg RSA -keysize 2048 -validity 360
-keystore /etc/security/serverKeys/ranger-admin-keystore.jks
-storepass {STOREPASS}
```

Where

ALIAS

A unique alias specified for creating the keystore file. This can be Ranger Admin host FQDN/ UQDN where it is going to be installed or can use custom text.

Note: If upgrading to CDP <= 7.1.4, it is recommended to use the host FQDN.

STOREPASS

A password which is used to protect the keystore.

```
chown -R ranger:ranger /etc/security/serverKeys/ranger-admin-keystore.jks
```

2. Export the certificate from the Ranger Admin keystore and create a certificate file.

```
${JAVA_HOME}/bin/keytool -exportcert -keystore
/etc/security/serverKeys/ranger-admin-keystore.jks -alias {ALIAS} -file
/etc/security/serverKeys/ranger-admin-trust.cer
```

Where

ALIAS

Use the same alias used for creating the keystore file ranger-admin-keystore.jks, as we are exporting the certificate associated with that alias.

Note: No need to provide a -storepass password to secure the certificate file.

3. Import Ranger Admin certificate into a truststore file for creating a "trusted certificate".

```
${JAVA_HOME}/bin/keytool -importcert -file /etc/security/serverKeys/ranger-
admin-trust.cer
-alias {ALIAS} -keystore /etc/security/serverKeys/ranger-truststore.jks -
storepass
{STOREPASS}
```

Where
ALIAS

Assign any alias name.

STOREPASS

A password to secure the truststore file.

Note: You must also import Solr Server certificate into /etc/security/serverKeys/ranger-truststore.jks, if Solr (CDP-INFRA-SOLR) Server has TLS enabled. This will help Ranger Admin to connect Solr for creating ranger_audits collection.

4. On Review Changes, set the TLS/SSL configurations:

Configuration Property	Description
Enable TLS/SSL for Ranger Admin (ranger.service.https.attrib.ssl.enabled)	Select this check box
Ranger Admin TLS/SSL Server JKS Keystore File Location (ranger.https.attrib.keystore.file)	The path to the keystore file created in Step 1 : Creating a keystore file for Ranger Admin OR The path to the existing keystore file if using existing CA-signed certificates
Ranger Admin TLS/SSL Server JKS Keystore File Password (ranger.service.https.attrib.keystore.pass)	The password of the keystore file used for Ranger Admin
Ranger Admin TLS/SSL Keystore File Alias (ranger.service.https.attrib.keystore.keyalias) **Note:** This config will be shown from CM-7.3.1 for CDP-7.1.5+ onwards in the initial wizard setup. If using CM <=7.3.1 & upgrading to CDP < 7.1.5, continue the use of alias name to FQDN of the host while creating Ranger Admin keystore or using the existing keystore.	Enter the alias used for the keystore file created in Step 1 : Creating a keystore file for Ranger Admin. The {{RANGER_ADMIN_HOST}} is a placeholder value which will be replaced with the host FQDN where Ranger Admin will be installed in the current cluster when Auto-TLS is enabled which uses host FQDN as alias while creating keystore file. The placeholder can be replaced to have custom alias value in case of manual TLS/SSL setup. If using a custom alias value which is the same as the host short name then use {{RANGER_ADMIN_HOST_UQDN}} placeholder as a value.
Ranger Admin TLS/SSL Trust Store File (ranger.truststore.file)	The path to the truststore file created in Step 3 : Import Ranger Admin certificate into a truststore file for creating a "trusted certificate" entry OR The path to the existing truststore file if using existing CA-signed certificates
Ranger Admin TLS/SSL Trust Store Password (ranger.truststore.password)	The password used for creating the trust store file
Ranger Tagsync TLS/SSL Trust Store File (xasecure.policymgr.clientssl.truststore)	The path to the truststore file created in Step 3 : Import Ranger Admin certificate into a truststore file for creating a "trusted certificate" entry OR The path to the existing truststore file if using existing CA-signed certificates

Ranger Tagsync TLS/SSL Trust Store Password (xasecure.policymgr.clientssl.truststore.password)	The password used for creating the trust store file
Ranger Usersync TLS/SSL Trust Store File (ranger.usersync.truststore.file)	The path to the truststore file created in Step 3 : Import Ranger Admin certificate into a truststore file for creating a "trusted certificate" entry OR The path to the existing truststore file if using existing CA-signed certificates
Ranger Usersync TLS/SSL Trust Store Password (ranger.usersync.truststore.password)	The password used for creating the trust store file

5. Import the Ranger Admin certificate into the truststore file configs for the following services, if present in the cluster. Ranger plugin is being enabled for these services during upgrade.
   a) Search for the following configuration properties:

      HDFS

      HDFS NameNode TLS/SSL Trust Store File (namenode_truststore_file)

      HDFS NameNode TLS/SSL Trust Store Password (namenode_truststore_password)

      Hive (shown under Hive Configuration in CM-7.3.1+)

      Hive Metastore TLS/SSL Trust Store File (hive.metastore.dbaccess.ssl.truststore.path)

      Hive Metastore TLS/SSL Trust Store Password (hive.metastore.dbaccess.ssl.truststore.password)

      Kafka

      Kafka Broker TLS/SSL Trust Store File (ssl.truststore.location)

      Kafka Broker TLS/SSL Trust Store Password (ssl.truststore.password.generator)

      Impala

      impala TLS/SSL Trust Store File (impala_truststore_file)

      impala TLS/SSL Trust Store Password (impala_truststore_password)

      Atlas

      Atlas Server TLS/SSL Trust Store File (truststore.file)

      Atlas Server TLS/SSL Trust Store Password (truststore.password)

   b) If a service has an existing truststore file, use that to import the Ranger Admin certificate. If not, then add a new truststore file and update the above configs.
   c) Import the Ranger Admin certificate into the existing/new truststore file.

```
${JAVA_HOME}/bin/keytool -importcert -file
/etc/security/serverKeys/ranger-admin-trust.cer -alias {ALIAS} -keystore
{TRUSTSTORE} -storepass {STOREPASS}
```

      Where

      ALIAS

      Assign any alias name.

      STOREPASS

      A password to secure the truststore file.

      TRUSTSTORE

      EITHER an existing truststore file used by the service to import the Ranger Admin certificate OR a new truststore file which will have Ranger Admin certificate.

   d) If using existing CA-signed certificates then add the path of the existing truststore file.
   e) Proceed to upgrade wizard for further other tasks.

6. Migrate Key Trustee KMS to Ranger KMS KTS.
   a) If you have Key Trustee service present in the cluster, Ranger KMS KTS will be added from the backend as part of the upgrade flow.
   b) To successfull start Ranger KMS KTS service in the upgrade flow:
   a) Change the alias of the keystore file used by Key Trustee KMS to point to the hostname where it is installed. Check Key Management Server Proxy TLS/SSL Server JKS Keystore File Location config to get the keystore file.

```
${JAVA_HOME}/bin/keytool -changealias -alias {EXISTING_ALIAS} -destalias
`hostname -f` -keystore {KT_KMS_KEYSTORE}
```

   Note: In a High Availability envrionment, perform this step on all KT KMS servers.

   b) Import Ranger Admin certificate into truststore file used by KT KMS. Check Key Management Server Proxy TLS/SSL Trust Store File config to get the truststore file getting used.

Step 11: Finalize the HDFS or Ozone Upgrade

Steps to finalize the HDFS or Ozone upgrade. This step is required only when Apache Hadoop version changes.

Note: Before upgrading the dependent services such as HBase, you must verify and ensure that the HDFS safemode is off.

To determine if you can finalize the upgrade, run important workloads and ensure that they are successful. After you have finalized the upgrade, you cannot roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time.

Make sure you have enough free disk space, keeping in mind that the following behavior continues until the upgrade is finalized:

- Deleting files does not free up disk space.
- Using the balancer causes all moved replicas to be duplicated.
- All on-disk data representing the NameNodes metadata is retained, which could more than double the amount of space required on the NameNode and JournalNode disks.

If you have not performed a rolling upgrade:

1. Go to the HDFS service.
2. Click the Instances tab.
3. Click the link for the NameNode instance. If you have enabled high availability for HDFS, click the link labeled NameNode (Active).

   The NameNode instance page displays.
4. Select Actions Finalize Metadata Upgrade and click Finalize Metadata Upgrade to confirm.

Finalize Ozone upgrade

To complete the upgrade of Ozone services, you must finalize the upgrade. For more information, see Upgrading Ozone parcels.

Step 12: Complete Post-Upgrade steps for upgrades to Cloudera Base on premises

Steps to perform after upgrading a cluster.

Several components require additional steps after you complete the upgrade to Cloudera Base on premises:

- HBase See Apache HBase post-upgrade tasks
- Apache Hive See Hive Post-Upgrade Tasks.
- Kafka
  1. Remove the following properties from the Kafka Broker Advanced Configuration Snippet (Safety Valve) for kafka.properties configuration property.
     - inter.broker.protocol.version
     - log.message.format.version
  2. Save your changes.
  3. Perform a rolling restart:
     a. Select the Kafka service.
     b. Click Actions Rolling Restart .
     c. In the pop-up dialog box, select the options you want and click Rolling Restart.
     d. Click Close once the command has finished.
- Kudu See Upgrade Notes for Apache Kudu 1.15 / CDP 7.1 on page 326
- Stream Messaging Manager See Configure SMM to monitor SRM replications on page 328
- YARN
  - Scheduler: If you are using Fair Scheduler, you must migrate to Capacity Scheduler during the upgrade process, and once the upgrade is finished you need to manually fine tune it. For more information, see Manual configuration of scheduler properties on page 177.
  - Considering logical processors in the calculation: The yarn.nodemanager.resource.count-logical-processors-as-cores property was not present in CDH 5. In Cloudera Runtime 7.1.1 (and in CDH 6), it is set to false by default, meaning that YARN does not consider logical processors in the calculation which can result in a performance degradatio if Linux Container Executor and CGroups are enabled. The extent of such degradation depends on the CPU manufacturer. To solve this issue, do the following:
    1. In Cloudera Manager, navigate to YARN Configuration .
    2. Find the YARN Service Advanced Configuration Snippet (Safety Valve) for yarn-site.xml property.
    3. Add the following configuration:

       ```
       yarn.nodemanager.resource.count-logical-processors-as-cores=true
       ```

       Using this configuration snippet ensures that all nodes that need the configuration receive it. This also ensures different NodeManager groups are consistently configured.
    4. Restart the NodeManager.
  - NodeManager recovery: The default value of the yarn.nodemanager.recovery.enabled property is true. However, if in you source cluster you used safety-valves to set this property to false it will stay false after upgrading from CDH 6 to CDP. Cloudera recommends to have this feature enabled and set the yarn.nodemanager.recovery.enabled property to true.
  - Log aggregation:: In order to see the history of applications that were launched before upgrade, do the following:
    1. In Cloudera Manager, navigate to YARN Configuration Category: Log aggreation .
    2. See the following configurations:

       ```
       yarn.log-aggregation.TFile.remote-app-log-dir-suffix=logs
       yarn.log-aggregation.IFile.remote-app-log-dir-suffix=logs-ifile
       ```
  - Maximum capacity: Set the yarn.scheduler.capacity.<queuepath>.user-limit-factor to a value that is greater than 1. This configuration will help to grow the queue usage beyond its configured capacity till its maximum capacity configured.

- YARN Queue Manager

  1. In Cloudera Manager, navigate to HDFS Configuration .
  2. Find the hadoop.http.staticuser.user property.
  3. If it is set, then remove the configuration.
  4. Restart the required services.
- Ranger Plugins

  The following Ranger plugins are not enabled by default after the upgrade. If these services are configured in the cluster, you will need to manually enable the plugins in order for them to use Ranger:

  - HBase
  - Kudu
  - Solr
  - YARN: If you want to enable the Ranger YARN plugin, you have to migrate the ACLs manually. For more information, see Configure a resource-based policy: YARN.
  - YARN

  The following Ranger plugins are enabled after an upgrade:

  - Atlas
  - HDFS
  - Hive
  - Hive on Tez
  - Impala
  - Kafka

  Important: For every service in your cluster, you should verify that Ranger is enabled. You should also create a plugin audit directory for every service on the cluster as follows:

  1. To enable a Ranger plugin for each service, from Cloudera Manager Service Name Configuration Ranger Service tick Ranger-1.
  2. To collect Ranger audit logs for each service in a unique directory, from Cloudera Manager Service Name Actions click Create <ServiceName> Plugin Audit Directory.

  After making any configuration changes, remember to save changes, restart the service and respond to (restart) any stale configurations.
- ZooKeeper

  Ensure, that QuorumSSL (Secure ZooKeeper) is enabled only if QuorumSASL (Server to server SASL authentication) is also enabled. Note, that QuorumSSL is enabled by default if AutoTLS is enabled. If QuorumSSL is enabled without QuorumSASL, then the ZooKeeper cluster can be slow to start due to some known ZooKeeper limitations.
- Sentry – See Importing Sentry privileges into Ranger policies on page 318.

  Important: If HDFS ACL sync is enabled (hdfs_sentry_sync_enable=true) on the CDH cluster, then you must install Ranger RMS to support the same functionality.

  - For more information about Ranger RMS, see Ranger Hive-HDFS ACL Sync Overview.
  - For steps to install Ranger RMS, see Installing Ranger RMS.
- Impala – See Apache Impala changes in CDP on page 305

Step 13: Exit Maintenance Mode

If you enabled Maintenance Mode before the upgrade, you must exit Maintenance Mode to complete the upgrade.

If you entered maintenance mode during this upgrade, exit maintenance mode.

On the HomeStatus tab, click next to the cluster name and select Exit Maintenance Mode.

Troubleshooting Upgrades

Cloudera Runtime upgrade failure

Warning: If there is any failure during the Cloudera Runtime upgrade process, you must not delete or modify the records in the upgrade_state table. If you still want to delete or modify the upgrade_state table, you must contact the Cloudera development team for modification or deletion approval. If you proceed without approval from the Cloudera development team, it can cause additional issues while resolving the runtime upgrade failure.

"Access denied" in install or update wizard

"Access denied" in install or update wizard during database configuration for Activity Monitor or Reports Manager.

Possible Reasons

Hostname mapping or permissions are not set up correctly.

Possible Solutions

- 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' IDENTI
  FIED 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_p
  assword';
  ```

  the value you enter for the database hostname can be either the FQDN or localhost.

The Hive on Tez service and the RangerAdmin role of the Ranger service kept going down

Possible Reasons

Both issues were caused by their respective Java heap size configuration being very low.

Possible Solutions

- For Hive on Tez: Look up the hiveserver2_java_heapsize config key under Hive on Tez > Configuration . Later, change its value from 256 MiB to 2 GiB.
- **Note:** The heap size can be verified on hive_on_tez configuration page, search for "heap".
- For Ranger: Look up the ranger_admin_max_heap_size config key under Ranger > Configuration. Later, change its value from 1 GiB to 4 GiB.

Both the services must be restarted.

Cluster hosts do not appear

Some cluster hosts do not appear when you click Find Hosts in install or update wizard.

Possible Reasons

You might have network connectivity problems.

Possible Solutions

- 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.

Cannot start services after upgrade

You have upgraded the Cloudera Manager Server, but now cannot start services.

Possible Reasons

You might have mismatched versions of the Cloudera Manager Server and Agents.

Possible Solutions

Make sure you have upgraded the Cloudera Manager Agents on all hosts. (The previous version of the Agents will heartbeat with the new version of the Server, but you cannot start HDFS and MapReduce with this combination.)

HDFS DataNodes fail to start

After upgrading, HDFS DataNodes fail to start with exception:

```
Exception in secureMainjava.lang.RuntimeException: Cannot start datanode bec
ause the configured max locked memory size (dfs.datanode.max.locked.memory)
of 4294967296 bytes is more than the datanode's available RLIMIT_MEMLOCK uli
mit of 65536 bytes.
```

Possible Reasons

HDFS caching, which is enabled by default in CDH 5 and higher, requires new memlock functionality from Cloudera Manager Agents.

Possible Solutions

Do the following:

1. Stop all CDH and managed services.
2. On all hosts with Cloudera Manager Agents, hard-restart the Agents. Before performing this step, ensure you understand the semantics of the hard_restart command by reading Cloudera Manager Agents.
   RHEL 7, SLES 12, Ubuntu 18.04 and higher

   ```
   sudo systemctl stop cloudera-scm-supervisord.service
   sudo systemctl restart cloudera-scm-agent
   ```
3. Start all services.

Cloudera services fail to start

Cloudera services fail to start.

Possible Reasons

Java might not be installed or might be installed at a custom location.

Possible Solutions

See Configuring a Custom Java Home Location on page 53 for more information on resolving this issue.

Host Inspector Fails

If you see the following message in the Host Inspector:

There are mismatched versions across the system, which will cause failures. See below for details on which hosts are running what versions of components.

When looking at the results, some hosts report Supervisord vX.X.X, while others report X.X.X-cmY.Y.Y (where X and Y are version numbers). During the upgrade, an old file on the hosts may cause the Host Inspector to indicate mismatched Supervisord versions.

This issue occurs because these hosts have a file on them at /var/run/cloudera-scm-agent/supervisor/__STARTING_CM_VERSION__ that contains a string for the older version of Cloudera Manager.

To resolve this issue:

1. Remove or rename the /var/run/cloudera-scm-agent/supervisor/__STARTING_CM_VERSION__ file
2. Perform a hard restart of the agents:

```
sudo systemctl stop cloudera-scm-supervisord.service
sudo systemctl start cloudera-scm-agent
```

3. Run the Host inspector again. It should pass without the warning.

Manual upgrade to Cloudera Base on premises

Manual steps to follow for upgrading a Cloudera Runtime cluster to a higher version of Cloudera Runtime if the Upgrade Wizard fails.

Warning: The upgrade wizard does not allow a manual upgrade option. There is no route or an option from Cloudera Manager to skip restart of custom services. In this scenario, the users must proceed with manual upgrade steps.

Important:

If Cloudera Base on premises Upgrade Wizard fails with any error, search for the Knowledge Base article from the MyCloudera website to find the failure resolution for that particular error. Perform the steps that are detailed in the Knowledge Base article to resolve the error and continue with the manual upgrade process. If you do not find the Knowledge Base article for any specific failure, contact Cloudera Support to open a support case.

Important:

Perform the steps in this section only if the upgrade wizard reports a failure, or if you selected Manual Upgrade from the Upgrade Wizard (the Manual Upgrade option is only available for minor or maintenance upgrades). Manual upgrades allow you to selectively stop and restart services to prevent or mitigate downtime for services or clusters where rolling restarts are not available.

All the following steps assume the starting Cloudera Runtime version is at least 7.0.3, because those are the lowest versions that Cloudera Manager 7.1 supports. The following steps are required when you are upgrading Cloudera Runtime cluster from Cloudera Runtime 7.0.3 to Cloudera Runtime 7.1.1 or higher versions.

The following steps should be executed roughly in the order that they are listed, and should only be executed if the service is configured.

Upgrade Ranger database and apply patches

1. Go to the RANGER service.
2. Select ActionsUpgrade Ranger Database and apply patches and click Upgrade Ranger Database and apply patches to confirm.

Setup Ranger Admin Component

1. Go to the Ranger service.
2. Select ActionsSetup Ranger Admin Component and click Setup Ranger Admin Component to confirm.

Start Ranger

1. Go to the Ranger service.
2. Select ActionsStart.

Set up the Ranger Plugin service

1. Go to the Ranger service.
2. Select ActionsSetup Ranger Plugin Service and click Setup Ranger Plugin Service to confirm.

Start Kudu

1. Go to the KUDU service.
2. Select ActionsStart.

Start ZooKeeper

1. Go to the ZooKeeper service.
2. Select ActionsStart.

Upgrade HDFS Metadata

1. Go to the HDFS service.
2. Select ActionsUpgrade HDFS Metadata and click Upgrade HDFS Metadata to confirm.

Start HDFS

1. Go to the HDFS service.
2. Select ActionsStart.

Start YARN QueueManager

1. Go to the QueueManager service.
2. Select ActionsStart.

Import Sentry Polices to Ranger

1. Go to the HDFS service.
2. Select ActionsImport Sentry Policies into Ranger and click Import Sentry Policies into Ranger to confirm.

Start HBASE

1. Go to the HBASE service.
2. Select ActionsStart.

Start YARN QueueManager

1. Go to the QueueManager service.
2. Select ActionsStart.

Clean NodeManager Recovery Directory (YARN)

1. Go to the YARN service.
2. Select ActionsClean NodeManager Recovery Directory and click Clean NodeManager Recovery Directory to confirm.

Reset ACLs on YARN Zookeeper nodes

Any other upgrade if Enable ResourceManager Recovery is enabled for a Resource Manager group (for example, ResourceManager Default Group) and ZooKeeper is a dependency of YARN. Note that when YARN is running in High Availability mode, ResourceManager recovery is always enabled.

1. Go to the YARN service.
2. Select Actions Reset ACLs on YARN Zookeeper nodes
3. Click Reset ACLs on YARN Zookeeper nodes to confirm.

Install YARN MapReduce Framework Jars

1. Go to the YARN service.
2. Select ActionsInstall YARN MapReduce Framework JARs and click Install YARN MapReduce Framework JARs to confirm.

Start YARN

1. Go to the YARN service.
2. Select ActionsStart.

Deploy Client Configuration Files

1. On the Home page, click to the right of the cluster name and select Deploy Client Configuration.
2. Click the Deploy Client Configuration button in the confirmation pop-up that appears.

Reinitialize Solr State for Upgrade

Warning: Performing this step deletes the entire Solr collection.

1. Go to the SOLR service.
2. Select ActionsReinitialize Solr State for Upgrade and click Reinitialize Solr State for Upgrade to confirm.

Bootstrap Solr Configuration

1. Go to the SOLR service.
2. Select Actions Bootstrap Solr Configuration and click Bootstrap Solr Configuration to confirm.

Start Solr

1. Go to the SOLR service.
2. Select ActionsStart.

Bootstrap Solr Collections

1. Go to the SOLR service.
2. Select Actions Bootstrap Solr Collections and click Bootstrap Solr Collections to confirm.

Create HDFS Home directory

1. Go to the infrastructure SOLR service.
2. Select ActionsCreate HDFS Home Dir and click Create HDFS Home Dir to confirm.

Create Ranger Plugin Audit Directory

1. Go to the Solr service.
2. Select ActionsCreate Ranger Plugin Audit Directory and click Create Ranger Plugin Audit Directory to confirm.

Start infrastructure Solr

1. Go to the infrastructure SOLR service.
2. Select ActionsStart.

Start HBASE

1. Go to the HBASE service.
2. Select ActionsStart.

Start KAFKA

1. Go to the KAFKA service.
2. Select ActionsStart.

Create Ranger Kafka Plugin Audit Directory

1. Go to the KAFKA service.
2. Select ActionsCreate Ranger Kafka Plugin Audit Directory and click Create Ranger Kafka Plugin Audit Directory to confirm.

Create HBase tables for Atlas

1. Go to the ATLAS service.
2. Select ActionsCreate HBase tables for Atlas and click Create HBase tables for Atlas to confirm.

Start Atlas

1. Go to the ATLAS service.
2. Select ActionsStart.

Create Ranger Atlas Plugin Audit Directory

1. Go to the ATLAS service.
2. Select ActionsCreate Ranger Atlas Plugin Audit Directory and click Create Ranger Atlas Plugin Audit Directory to confirm.

Start Phoenix

1. Go to the PHOENIX service.
2. Select ActionsStart.

Install MapReduce Framework Jars

1. Go to the YARN service.
2. Select ActionsInstall YARN MapReduce Framework JARs and click Install YARN MapReduce Framework JARs to confirm.

Start YARN

1. Go to the YARN service.
2. Select ActionsStart.

Deploy Client Configuration Files

1. On the Home page, click to the right of the cluster name and select Deploy Client Configuration.
2. Click the Deploy Client Configuration button in the confirmation pop-up that appears.

Upgrade the Hive Metastore Database

Warning: Your upgrade will fail if you do not complete this step.

1. Go to the Hive service.
2. If the Hive service is running, stop it:
   a. Select ActionsStop and click Stop to confirm.
3. Select ActionsUpgrade Hive Metastore Database Schema and click Upgrade Hive Metastore Database Schema to confirm.
4. If you have multiple instances of Hive, perform the upgrade on each metastore database.
5. Select ActionsValidate Hive Metastore Schema and click Validate Hive Metastore Schema to check that the schema is now valid.

Start Hive

1. Go to the Hive service.
2. Select ActionsStart.

Create Hive Warehouse Directory

1. Go to the HIVE service.
2. Select Actions Create Hive Warehouse Directory and click Create Hive Warehouse Directory to confirm.

Create Hive Warehouse External Directory

1. Go to the HIVE service.
2. Select Actions Create Hive Warehouse External Directory and click Create Hive Warehouse External Directory to confirm.

Create Hive Sys database

1. Go to the HIVE service.
2. Select ActionsCreate Hive Sys database and click Create Hive Sys database to confirm.

Create Ranger Plugin Audit Directory

1. Go to the HIVE service.
2. Select Actions Create Ranger Plugin Audit Directory and click Create Ranger Plugin Audit Directory to confirm.

Start Impala

1. Go to the Impala service.
2. Select ActionsStart.

Create Ranger Plugin Audit Directory

1. Go to the Impala service.
2. Select Actions Create Ranger Plugin Audit Directory and click Create Ranger Plugin Audit Directory to confirm.

Create Spark Driver Log Dir

1. Go to the SPARK_ON_YARN service.
2. Select ActionsCreate Spark Driver Log Dir and click Create Spark Driver Log Dir to confirm.

Start Spark

1. Go to the SPARK_ON_YARN service.
2. Select ActionsStart.

Start Livy

1. Go to the LIVY service.
2. Select ActionsStart.

Upgrade Oozie Database Schema

1. Go to the OOZIE service.
2. If the OOZIE service is running, stop it:

   Select Actions > Stop and click Stop to confirm.
3. Select Actions Upgrade Oozie Database Schema and click Upgrade Oozie Database Schema to confirm.

Updating column types in Oozie Database

If the Oracle database is used for Oozie, then you must update the APP_PATH column type to store values with more than 255 characters. This ensures Oozie does not get stuck in PREP state when your application path exceeds the 255 character limit. If the APP_PATH column type is not updated, then Oozie fails to run the jobs with the following database error message Data too long for column 'app_path'. This scenario is also applicable to coordinator and bundle jobs. For database types other than Oracle, this update is not mandatory for using Oozie. It works without the

update if the APP_PATH value does not exceed 255 characters. Also, Oozie's internal database schema validation fails with an unexpected APP_PATH column type. However, this validation does not have any effect. It just logs it's result.

You must update the APP_PATH column type in the WF_JOBS, BUNDLE_JOBS, and COORD_JOBS tables in the Oozie database for the following conditions:

- When you use the **Oracle** database for Oozie service.
- When you are upgrading Cloudera Runtime from earlier versions to Cloudera Base on premises 7.1.9 SP1 CHF1 version or later.

If you do not execute the following statements on the Oozie Oracle database, then the Oozie service fails to run the jobs with a database persistence error. This update is not mandatory, but highly recommended for other database types, such as MySQL, MariaDB, and PostgreSQL, due to the internal database schema validation.

Examples:

- On the Oozie Oracle database - The following example uses the Oracle sqlplus command-line tool:

```
sqlplus <OOZIE_DB_USERNAME>@localhost/<SERVICE_NAME>

SQL> ALTER TABLE <TABLE_NAME> ADD (APP_PATH_TMP CLOB);

Table altered.

SQL> UPDATE <TABLE_NAME> SET APP_PATH_TMP = APP_PATH;

X rows updated.

SQL> ALTER TABLE <TABLE_NAME> DROP COLUMN APP_PATH;

Table altered.

SQL> ALTER TABLE <TABLE_NAME> RENAME COLUMN APP_PATH_TMP TO APP_PATH;

Table altered.
```

- On the Oozie MySQL database - The following example uses the MySQL mysql command-line tool:

```
$ mysql -u root -p
Enter password:

mysql> use <OOZIE_DATABASE_NAME>;
Database changed

mysql> ALTER TABLE <TABLE_NAME> MODIFY COLUMN app_path text;
Query OK, X rows affected (0.03 sec)
Records: X Duplicates: 0 Warnings: 0
mysql> exit
Bye
```

- On the Oozie MariaDB database - The following example uses the MariaDB mysql command-line tool:

```
$ mysql -u root -p
Enter password:

MariaDB [(none)]> use <OOZIE_DATABASE_NAME>;
Database changed

MariaDB [OOZIE_DATABASE_NAME]> ALTER TABLE <TABLE_NAME> MODIFY COLUMN a
pp_path text;
Query OK, X rows affected (2.11 sec)
Records: X Duplicates: 0 Warnings: 0
```

```
MariaDB [OOZIE_DATABASE_NAME]> exit
Bye
```

- On the Oozie PostgreSQL database - The following example uses the PostgreSQL psql command-line tool:

```
$ psql -U postgres
Password for user postgres: *****

postgres=# \c <OOZIE_DATABASE_NAME>;
You are now connected to database "<OOZIE_DATABASE_NAME>" as user "postgre
s".
OOZIE_DATABASE_NAME=# ALTER TABLE <TABLE_NAME> ALTER COLUMN app_path type
text;
ALTER TABLE

OOZIE_DATABASE_NAME=# \q
```

Upgrade Oozie SharedLib

1. Go to the Oozie service.
2. If the OOZIE service is stopped, start it:

   Select ActionsStart and click Start to confirm.
3. Select ActionsInstall Oozie SharedLib and click Install Oozie SharedLib to confirm.

Upload Tez tar file to HDFS

1. Go to the TEZ service.
2. Select Actions > Upload Tez tar file to HDFS and click Upload Tez tar file to HDFS to confirm.

Migrate Hive tables for Cloudera Base on premises upgrade

1. Go to the HIVE_ON_TEZ service.
2. Select ActionsMigrate Hive tables for CDP upgrade and click Migrate Hive tables for CDP upgrade to confirm.

Create Ranger Plugin Audit Directory

1. Go to the Hive-on-Tez service.
2. Select Actions Create Ranger Plugin Audit Directory and click Create Ranger Plugin Audit Directory to confirm.

Start Hive on Tez

1. Go to the Hive-on-Tez service.
2. Select ActionsStart.

Start Hue

1. Go to the HUE service.
2. Select ActionsStart.

Start the Remaining Cluster Services

1. Use rolling restart or full restart.
2. Ensure that all services are started or restarted. You can use Cloudera Manager to start the cluster, or you can restart the services individually. The Cloudera Manager Home page indicates which services have stale configurations and require restarting.
3. To start or restart the cluster:
   a. On the Home Status page, click the down arrow to the right of the cluster name and select Start or Restart.
   b. Click Start that appears in the next screen to confirm. The Command Details window shows the progress of starting services.
   c. When All services successfully started appears, the task is complete and you can close the Command Details window.

Validate the Hive Metastore Database Schema

Warning: Your upgrade will fail if you do not complete this step.

1. Select ActionsValidate Hive Metastore Schema and click Validate Hive Metastore Schema to confirm.
2. If you have multiple instances of Hive, perform the validation on each metastore database.
3. Select ActionsValidate Hive Metastore Schema and click Validate Hive Metastore Schema to check that the schema is now valid.

Test the Cluster and Finalize HDFS Metadata

To determine if you can finalize the upgrade, run important workloads and ensure that they are successful. After you have finalized the upgrade, you cannot roll back to a previous version of HDFS without using backups. Verifying that you are ready to finalize the upgrade can take a long time.

When you are ready to finalize the upgrade, do the following:

- 1. Go to the HDFS service.
  2. Click the Instances tab.
  3. Click the link for the NameNode instance. If you have enabled high availability for HDFS, click the link labeled NameNode (Active).

     The NameNode instance page displays.
  4. Select Actions Finalize Metadata Upgrade and click Finalize Metadata Upgrade to confirm.

Clear the Upgrade State Table

After completing all of the previous steps, do the following to complete the upgrade:

1. Log in to the Cloudera Manager server host.
2. Stop the Server.

```
sudo systemctl stop cloudera-scm-server
```

3. Log in to the command-line environment for the Cloudera Manager database. (mysql, sqlplus, or postgres psql).

4. Run the following command:

```
DELETE FROM UPGRADE_STATE;
```

5. Start the Server.

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

Rolling Back a Cloudera Private Cloud Base Upgrade from version 7.1.9 to CDH 6

You can roll back an upgrade from Cloudera Private Cloud Base 7 to CDH 6. The rollback restores your CDH cluster to the state it was in before the upgrade, including Kerberos and TLS/SSL configurations.

Important: Any data created after the upgrade is lost.

In a typical upgrade, you first upgrade Cloudera Manager from version 6.x to version 7.x, and then you use the upgraded version of Cloudera Manager 7 to upgrade CDH 6 to Cloudera Private Cloud Base 7. (See Upgrading a CDH 6 Cluster on page 121.) If you want to roll back this upgrade, follow these steps to roll back your cluster to its state prior to the upgrade.

You can roll back to CDH 6 after upgrading to Cloudera Private Cloud Base 7 only if the HDFS upgrade has not been finalized. The rollback restores your CDH cluster to the state it was in before the upgrade, including Kerberos and TLS/SSL configurations.

Important: Follow all of the steps in the order presented in this topic. Cloudera recommends that you read through the backup and rollback steps before starting the backup process. You may want to create a detailed plan to help you anticipate potential problems.

Important:

These rollback steps depend on complete backups taken before upgrading Cloudera Manager and CDH. See Step 2: Backing Up Cloudera Manager 6 on page 71 and Step 3: Backing Up the Cluster on page 134.

For steps where you need to restore the contents of a directory, clear the contents of the directory before copying the backed-up files to the directory. If you fail to do this, artifacts from the original upgrade can cause problems if you attempt the upgrade again after the rollback.

Review Limitations

The rollback procedure has the following limitations:

- HDFS – If you have finalized the HDFS upgrade, you cannot roll back your cluster.
- Compute clusters – Rollback for Compute clusters is not supported. You must remove any compute clusters before rolling back.
- Configuration changes, including the addition of new services or roles after the upgrade, are not retained after rolling back Cloudera Manager.

  Cloudera recommends that you not make configuration changes or add new services and roles until you have finalized the HDFS upgrade and no longer require the option to roll back your upgrade.
- HBase – If your cluster is configured to use HBase replication, data written to HBase after the upgrade might not be replicated to peers when you start your rollback. This topic does not describe how to determine which, if any, peers have the replicated data and how to roll back that data. For more information about HBase replication, see HBase Replication.

- Sqoop 2 – As described in the upgrade process, Sqoop2 had to be stopped and deleted before the upgrade process and therefore will not be available after the rollback.
- Kafka – Once the Kafka log format and protocol version configurations (the inter.broker.protocol.version and log.message.format.version properties) are set to the new version (or left blank, which means to use the latest version), Kafka rollback is not possible.

Stop the Cluster

1. If HBase is deployed in the cluster do the following before stopping the cluster:

   The HBase Master procedures changed between the two versions, so if a procedure was started by HBase 2.2 (CDP 7.x) then the older HBase 2.1 won't be able to continue the procedure after the rollback. For this reason the Procedure Store in HBase Master must be cleaned before the rollback. If CDP 7.x HBase Master was never started, then the rollback should be fine. But if HBase Master was running with the new version and there is any ongoing (or stuck) HBase Master Procedure present in the CDP 7 HBase Master, then the older CDH 6 HBase Master will fail to start after the rollback. If this happens, HBase will need manual fix after the rollback (e.g. the sidelining of the HBase Master Procedure WAL files and the potential fixing of inconsistencies in HBase).

   To avoid this problem, you should try to verify that no unfinished procedure is present before stopping HBase Master on the CDP 7.x Cluster. Please follow these steps:

   a. Make sure there was no traffic running against the HBase Cluster recently (in the last 10 minutes) that can trigger e.g. table creation or deletion, region assignment or split or merge, etc.
   b. Disable automatic Balancer and Normalizer in HBase. Also disable Split and Merge procedures, before stopping the CDP 7 Cluster. All these tools in HBase can cause the starting of new HBase Master Procedures, which we want to avoid now. Issue the following commands in HBase Shell:

      ```
      balance_switch false
      normalizer_switch false
      splitormerge_switch 'SPLIT', false
      splitormerge_switch 'MERGE', false
      ```

   c. Check the list of procedures on the HBase Master Web UI (In Cloudera Manager, go to the HBase service and open the HBase Web UI Procedures & Locks tab). Wait until you see procedures only with final states like 'SUCCESS', 'FAILED' or 'ROLLEDBACK'.
   d. Get the list of procedures from HBase shell using the 'list_procedures' command. Wait until you see procedures only with final states like 'SUCCESS', 'FAILED' or 'ROLLEDBACK'. The State appears in the third column of the table returned by the 'list_procedures' command.

   If the HBase Master doesn't start after the rollback and some procedure-related exceptions are found in the role logs (like "BadProcedureException" or decode errors in the "ProcedureWALFormatReader" class, or "ClassNotFoundException" for procedure classes), then this is most likely caused by CDP 7 procedures that still remain in the procedure WAL files. In this case, please open a ticket for Cloudera customer support, who will help you to sideline the procedure WAL files and fix any potential inconsistencies in HBase.
2. On the Home Status tab, click the Actions menu and select Stop.
3. 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.
4. Go to the YARN service and click Actions Clean NodeManager Recovery Directory . The CDH 6 NodeManager will not start up after the downgrade if it finds CDP 7.x data in the recovery directory. The format and content of the NodeManager's recovery state store was changed between CDH 6.x and CDP 7.x. The recovery directory used by CDP 7.x must be cleaned up as part of the downgrade to CDH 6.

(Parcels) Downgrade the Software

Follow these steps only if your cluster was upgraded using Cloudera parcels.

1. Log in to the Cloudera Manager Admin Console.
2. Select Hosts Parcels .

   A list of parcels displays.
3. Locate the CDH 6 parcel and click Activate. (This automatically deactivates the Cloudera Private Cloud Base 7 parcel.) See Activating a Parcel for more information. If the parcel is not available, use the Download button to download the parcel.

   Important: If you added new services that are not part of CDH 6 during the upgrade, a list of services that need to be deleted displays. You must delete these services before activating the CDH 6 parcel.
4. If you include any additional components in your cluster, such as Search or Impala, click Activate for those parcels.
5. If the Ranger service is deployed in the cluster, disable the Ranger plugin from the services below, if they are deployed in the cluster:
   - HDFS: Go to the HDFS service > Configurations and disable the Enable Ranger Authorization configuration property.
   - Hive: Go to the Hive service > Configurations and delete the Ranger Service configuration property.
   - Kafka: Go to the Kafka service > Configurations and delete the Ranger Service configuration property.
   - Impala Go to the Impala service > Configurations > delete the Ranger Service configuration property.
6. After performing the above steps to disable the plugin, Stop the Ranger service and delete it.
7. The Sentry service will be added when you perform the Restore Cloudera Manager Databases steps, later in this rollback procedure. The Sentry service will be added in Cloudera Manager and will continue to use the database configuration saved in Cloudera Manager.

Important:

Do not start any services. (Select the Activate Only option.)

If you accidentally restart services, stop your cluster before proceeding.

Stop Cloudera Manager

1. Stop the .
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Stop .
2. Stop the Server.

```
sudo systemctl stop cloudera-scm-server
```

3. Hard stop the Cloudera Manager agents. Run the following command on all hosts:

```
sudo systemctl stop cloudera-scm-supervisord.service
```

Restore Cloudera Manager Databases

Restore the Cloudera Manager databases from the backup of Cloudera Manager that was taken before upgrading the cluster toCloudera Private Cloud Base 7. See the procedures provided by your database vendor.

- MariaDB 5.5: http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/
- MySQL 5.5: http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html
- MySQL 5.6: http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html

- PostgreSQL 8.4: https://www.postgresql.org/docs/8.4/static/backup.html
- PostgreSQL 9.2: https://www.postgresql.org/docs/9.2/static/backup.html
- PostgreSQL 9.3: https://www.postgresql.org/docs/9.3/static/backup.html
- Oracle 11gR2: https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm
- HyperSQL: http://hsqldb.org/doc/guide/management-chapt.html#mtc_backup

Restore Cloudera Manager Server

Use the backup of CDH that was taken before the upgrade to restore Cloudera Manager Server files and directories. Substitute the path to your backup directory for *CM7_CDH6* in the following steps:

1. On the host where the Event Server role is configured to run, restore the Events Server directory from the CM 7/CDH 6 backup.

```
cp -rp /var/lib/cloudera-scm-eventserver /var/lib/cloudera-scm-eventserv
er-CM<ph outputclass="cdoc-specific-cm-from"/>-CDH<ph outputclass="cdoc-
specific-cdh-from"/>
rm -rf /var/lib/cloudera-scm-eventserver/*
cp -rp /var/lib/cloudera-scm-eventserver_CM7_CDH6/* /var/lib/cloudera-scm-
eventserver/
```

2. Remove the Agent runtime state. Run the following command on all hosts:

```
rm -rf /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent/response.
avro
```

   This command may return a message similar to: rm: cannot remove ‘/var/run/cloudera-scm-agent/process’: Device or resource busy. You can ignore this message.
3. On the host where the Service Monitor is running, restore the Service Monitor directory:

```
rm -rf /var/lib/cloudera-service-monitor/*
cp -rp /var/lib/cloudera-service-monitor_CM7_CDH6/* /var/lib/cloudera-ser
vice-monitor/
```

4. On the host where the Host Monitor is running, restore the Host Monitor directory:

```
rm -rf /var/lib/cloudera-host-monitor/*
cp -rp /var/lib/cloudera-host-monitor_CM7_CDH6/* /var/lib/cloudera-host-mo
nitor/
```

Start Cloudera Manager

1. Log in to the Cloudera Manager server host.
2. Start the Server.

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

3. Start the Cloudera Manager Agent.

   Run the following commands on all cluster hosts:

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

4. Start the Cloudera Management Service.
   a. Log in to the Cloudera Manager Admin Console.
   b. Select Clusters Cloudera Management Service .
   c. Select Actions Start .

   The cluster page may indicate that services are in bad health. This is normal.
5. Stop the cluster. In the Cloudera Manager Admin Console, click the Actions menu for the cluster and select Stop.

Optional Step

1. Add and start Navigator Audit Server and Navigator Metadata Server role instances.

Roll Back ZooKeeper

1. Using the backup of Zookeeper that you created when backing up your CDH 6.x cluster, restore the contents of the *DATADIR* on each ZooKeeper server. These files are located in a directory specified with the dataDir property in the ZooKeeper configuration. The default location is /var/lib/zookeeper. For example:

   ```
   rm -rf /var/lib/zookeeper/*
   cp -rp /var/lib/zookeeper_CM7_CDH6/* /var/lib/zookeeper/
   ```
2. Using the backup of Zookeeper that you created when backing up your CDH 6.x cluster, restore the contents of the *TRANSACTION LOG DIRECGTORY* on each ZooKeeper server. These files are located in a directory specified with the Transaction Log Direcgtory property in the ZooKeeper configuration. For example:

   ```
   rm -rf /var/lib/zookeeper/*
   cp -rp /var/lib/zookeeper_CM7_CDH6/* /var/lib/zookeeper/
   ```
3. Make sure that the permissions of all the directories and files are as they were before the upgrade.
4. Start ZooKeeper using Cloudera Manager.

Roll Back HDFS

You cannot roll back HDFS while high availability is enabled. The rollback procedure in this topic creates a temporary configuration without high availability. Regardless of whether high availability is enabled, follow the steps in this section.

1. Roll back all of the Journal Nodes. (Only required for clusters where high availability is enabled for HDFS). Use the JournalNode backup you created when you backed up HDFS before upgrading to Cloudera Private Cloud Base.
   a. Log in to each Journal Node host and run the following commands:

      ```
      rm -rf /dfs/jn/ns1/current/*
      ```

      ```
      cp -rp <JOURNAL_NODE_BACKUP_DIRECTORY>/ns1/current/* /dfs/jn/ns1/curren
      t/
      ```
   b. Start the JournalNodes using Cloudera Manager:
      1. Go to the HDFS service.
      2. Select the Instances tab.
      3. Select all JournalNode roles from the list.
      4. Click Actions for Selected Start .

2. Roll back all of the NameNodes. Use the NameNode backup directory you created before upgrading to Cloudera Private Cloud Base. (/etc/hadoop/conf.rollback.namenode) to perform the following steps on all NameNode hosts:

   a. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file on all NameNode hosts (located in the temporary rollback directory) and update the keystore passwords with the actual cleartext passwords.

      The passwords will have values that look like this:

      ```
      <property>
          <name>ssl.server.keystore.password</name>
          <value>********</value>
        </property>
        <property>
          <name>ssl.server.keystore.keypassword</name>
          <value>********</value>
        </property>
      ```

   b. (TLS only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file and remove the hadoop.security.credential.provider.path property.

   c. (TLS only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file and update the ssl.server.keystore.location property:

      ```
      # Original version of the keystore.location property:
      <property>
        <name>ssl.server.keystore.location</name>
        <value>/var/run/cloudera-scm-agent/process/879-hdfs-NAMENODE/cm-auto-host_keystore.jks</value>
      </property>
      ```

      ```
      # New version of the keystore.location property:
      <property>
        <name>ssl.server.keystore.location</name>
        <value>/etc/hadoop/conf.rollback.namenode/cm-auto-host_keystore.jks</value>
      </property>
      ```

3. Edit the /etc/hadoop/conf.rollback.namenode/hdfs-site.xml file on all NameNode hosts and make the following changes:

   a. Update the dfs.namenode.inode.attributes.provider.class property. If Sentry was installed prior to the upgrade, change the value of the property from org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer to "org.apache.sentry.hdfs.SentryINodeAttributesProvider. If Sentry was not installed, remove this property.

   b. Change the path in the dfs.hosts property to the value shown in the example below. The file name, dfs_all_hosts.txt, may have been changed by a user. If so, substitute the correct file name.

      ```
      # Original version of the dfs.hosts property:
      <property>
      <name>dfs.hosts</name>
      <value>/var/run/cloudera-scm-agent/process/63-hdfs-NAMENODE/dfs_all_hosts.txt</value>
      </property>
      ```

      ```
      # New version of the dfs.hosts property:
      <property>
      <name>dfs.hosts</name>
      <value>/etc/hadoop/conf.rollback.namenode/dfs_all_hosts.txt</value>
      </property>
      ```

c. Remove the property that has the following value:

```
com.cloudera.navigator.audit.hdfs.HdfsAuditLoggerCdh5
```

4. Edit the /etc/hadoop/conf.rollback.namenode/core-site.xml and change the value of the net.topology.script.file .name property to /etc/hadoop/conf.rollback.namenode. For example:

```
# Original property
<property>
<name>net.topology.script.file.name</name>
<value>/var/run/cloudera-scm-agent/process/63-hdfs-NAMENODE/topology.py</
value>
</property>
```

```
# New property
<property>
<name>net.topology.script.file.name</name>
<value>/etc/hadoop/conf.rollback.namenode/topology.py</value>
</property>
```

5. Edit the /etc/hadoop/conf.rollback.namenode/topology.py file and change the value of MAP_FILE to /etc/hadoop/conf.rollback.namenode. For example:

```
MAP_FILE = '/etc/hadoop/conf.rollback.namenode/topology.map'
```

6. (TLS-enabled clusters only) Run the following command:

```
sudo -u hdfs kinit hdfs/<NAMENODE HOST NAME> -l 7d -kt /etc/hadoop/conf.
rollback.namenode/hdfs.keytab
```

7. Run the following command:

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.namenode namenode -
rollback
```

8. Restart the NameNodes and JournalNodes using Cloudera Manager:
   a. Go to the HDFS service.
   b. Select the Instances tab, and then select all Failover Controller, NameNode, and JournalNode roles from the list.
   c. Click Actions for Selected Restart .
9. Rollback the DataNodes.

   Use the DataNode rollback directory you created before upgrading to Cloudera Private Cloud Base (/etc/hadoop/conf.rollback.datanode) to perform the following steps on all DataNode hosts:

   a. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.datanode/ssl-server.xml file on all DataNode hosts (Located in the temporary rollback directory.) and update the keystore passwords (ssl.server.keystore.password and ssl.server.keystore.keypassword) with the actual passwords.

      The passwords will have values that look like this:

```
<property>
    <name>ssl.server.keystore.password</name>
    <value>********</value>
  </property>
  <property>
    <name>ssl.server.keystore.keypassword</name>
    <value>********</value>
  </property>
```

b. (TLS only) Edit the /etc/hadoop/conf.rollback.datanode/ssl-server.xml file and update the ssl.server.keystore.location property:

```
# Original version of the keystore.location property:
<property>
  <name>ssl.server.keystore.location</name>
  <value>/var/run/cloudera-scm-agent/process/879-hdfs-NAMENODE/cm-auto-host_keystore.jks</value>
</property>
```

```
# New version of the keystore.location property:
<property>
  <name>ssl.server.keystore.location</name>
  <value>/etc/hadoop/conf.rollback.namenode/cm-auto-host_keystore.jks</value>
</property>
```

c. (TLS only) Edit the /etc/hadoop/conf.rollback.datanode/ssl-server.xml file and remove the hadoop.security.credential.provider.path property.

d. Edit the /etc/hadoop/conf.rollback.datanode/hdfs-site.xml file and remove the dfs.datanode.max.locked.memory property.

e. Run one of the following commands:

- Run as root if the DataNodes use reserved ports.

  Search in the logs for completed rollback line. It will not be shown in the command line of the rollback.
- If the DataNode is running with privileged ports (usually 1004 and 1006):

  ```
  cd /etc/hadoop/conf.rollback.datanode
  export HADOOP_SECURE_DN_USER=hdfs
  export JSVC_HOME=/opt/cloudera/parcels/<PARCEL_FILENAME>/lib/bigtop-utils
  hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
  ```
- If the DataNode is not running on privileged ports:

  ```
  cd /etc/hadoop/conf.rollback.datanode
  sudo hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
  ```

  You may see the following error after issuing these commands:

  ```
  ERROR datanode.DataNode: Exception in secureMain
  java.io.IOException: The path component: '/var/run/hdfs-sockets' in
   '/var/run/hdfs-sockets/dn' has permissions 0755 uid 39998 and gid 1
  006.
  ```

```
It is not protected because it is owned by a user who is not root
and not the effective user: '0'.
```

The error message will also include the following command to run:

```
chown root /var/run/hdfs-sockets
```

After running this command, rerun the DataNode rollback command:

```
sudo hdfs --config /etc/hadoop/conf.rollback.datanode datanode -roll
back
```

The DataNodes will now restart successfully.

When the rolling back of the DataNodes is complete, terminate the console session by typing Control-C. Look for output from the command similar to the following that indicates when the DataNode rollback is complete:

```
Rollback of /dataroot/ycloud/dfs/dn/current/BP-<Block Group number> is c
omplete
```

f. If High Availability for HDFS is enabled, restart the HDFS service. In the Cloudera Manager Admin Console, go to the HDFS service and select Actions Restart .

g. If high availability is not enabled for HDFS, use the Cloudera Manager Admin Console to restart all NameNodes and DataNodes.

1. Go to the HDFS service.
2. Select the Instances tab
3. Select all DataNode and NameNode roles from the list.
4. Click Actions for Selected Restart .

10. If high availability is not enabled for HDFS, roll back the Secondary NameNode.

a. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.secondarynamenode/ssl-server.xml file on all Secondary NameNode hosts (Located in the temporary rollback directory.) and update the keystore passwords with the actual cleartext passwords.

The passwords will have values that look like this:

```
<property>
    <name>ssl.server.keystore.password</name>
    <value>********</value>
  </property>
  <property>
    <name>ssl.server.keystore.keypassword</name>
    <value>********</value>
  </property>
```

b. (TLS only) Edit the /etc/hadoop/conf.rollback.secondarynamenode/ssl-server.xml file and remove the hadoop.security.credential.provider.path property.

c. Log in to the Secondary NameNode host and run the following commands:

```
rm -rf /dfs/snn/*
cd /etc/hadoop/conf.rollback.secondarynamenode/
```

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.secondarynamenode
 secondarynamenode -format
```

When the rolling back of the Secondary NameNode is complete, terminate the console session by typing Control-C. Look for output from the command similar to the following that indicates when the Secondary NameNode rollback is complete:

```
2020-12-21 17:09:36,239 INFO namenode.SecondaryNameNode: Web server init
 done
```

11. Restart the HDFS service. Open the Cloudera Manager Admin Console, go to the HDFS service page, and select Actions Restart .

    The Restart Command page displays the progress of the restart. Wait for the page to display the Successfully restarted service message before continuing.

Start the HBase Service

Restart the HBase Service. Open the Cloudera Manager Admin Console, go to the HBase service page, and select Actions Start .

If you have configured any HBase coprocessors, you must revert them to the versions used before the upgrade.

If CDP 7.x HBase Master was started after the upgrade and there was any ongoing (or stuck) HBase Master Procedure present in the HBase Master before stopping the CDP 7 Cluster, then it is expected for the CDH 6 HBase Master to fail with warnings and errors in the role log from the classes like 'ProcedureWALFormatReader' and 'WALProcedureStore' or 'TransitRegionStateProcedure'. These errors mean that the HBase Master Write-Ahead Log files are incompatible with the CDH 6 HBase version. The only way to fix this problem is to sideline the log files (all the files placed under /hbase/MasterProcWALs by default), then restart the HBase Master. After the HBase Master has started, Use the HBCK command to find out if there are any inconsistencies that will need to be fixed manually.

You my encounter other errors when starting HBase (for example, replication-related problems, region assignment related issues, and meta region assignment problems). In this case you should delete the znode in ZooKeeper and then start HBase again. (This will delete replication peer information and you will need to re-configure your replication schedules.):

1. In Cloudera Manager, look up the value of the zookeeper.znode.parent property. The default value is /hbase.
2. Connect to the ZooKeeper ensemble by running the following command from any HBase gateway host:

   ```
   zookeeper-client -server ZOOKEEPER_ENSEMBLE
   ```

   To find the value to use for *ZOOKEEPER_ENSEMBLE*, open the /etc/hbase/conf.cloudera.<*HBASE SERVICE NAME*>/hbase-site.xml file on any HBase gateway host. Use the value of the hbase.zookeeper.quorum property.

   Note:

   If you have deployed a secure cluster, you must connect to ZooKeeper using a client jaas.conf file. You can find such a file in an HBase process directory (/var/run/cloudera-scm-agent/process/). Specify the jaas .conf using the JVM flags by running the following commands in the ZooKeeper client:

   ```
   CLIENT_JVMFLAGS=
    "-Djava.security.auth.login.config=/var/run/cloudera-scm-agent/pro
   cess/HBASE_PROCESS_DIRECTORY/jaas.conf"
   zookeeper-client -server <zookeeper_ensemble>
   ```

   The ZooKeeper command-line interface opens.
3. Enter the following command:

   ```
   rmr /hbase
   ```

4. After HBase is healthy, make sure you restore the states of the Balancer and Normalizer (enable them if they were enabled before the rollback). Also re-enable the Merge and Split operations you disabled before the rollback to avoid the Master Procedure incompatibility problem. Run the following commands in HBase Shell:

```
balance_switch true
normalizer_switch true
splitormerge_switch 'SPLIT', true
splitormerge_switch 'MERGE', true
```

Fixing tableinfo file format

When you are rolling back from CDP Private Cloud Base 7.1.8 to CDH 6 if you encounter a change in the tableinfo file name format from the new tableinfo file name that was created during the 7.1.8 upgrade can prevent HBase from functioning normally.

After the rollback, if HDFS rollback was not successful and Hbase is unable to read the tableinfo files then use the HBCK2 tool to verify the list of tableinfo files that need to be fixed.

Follow these steps to execute the HBCK2 command on the HBCK2 tool to fix the tableinfo file format:

1. Contact Cloudera support to request the latest version of HBCK2 tool.
2. Use the following HBCK2 command and run the HBCK2 tool without the –fix option:

```
hbase --config /path/to/client/conf hbck -j
~/path/to/hbck/hbase-hbck2-1.0.0-<build>.jar shortenTableinfo
```

   For example:

```
hbase --config /etc/hbase/conf hbck -j
~/hbase-operator-tools/hbase-hbck2/target/hbase-hbck2-1.0.0-SNAPSHOT.jar
shortenTableinfo
```

   The command displays the following message and the list of files to be fixed:

   Found the following tableinfo file names containing file size

   If the list is empty, no additional steps are needed. Go to Step 11.
3. Use the following HBCK2 command and run the HBCK2 tool with the –fix option:

```
hbase --config /etc/hbase/conf hbck -j
~/hbase-operator-tools/hbase-hbck2/target/hbase-hbck2-1.0.0-SNAPSHOT.jar
shortenTableinfo –fix
```

4. Check the output and verify whether all the tableinfo files are fixed.

Restore CDH Databases

Restore the following databases from the CDH 6 backups:

- Hive Metastore
- Hue
- Oozie
- Sentry Server

The steps for backing up and restoring databases differ depending on the database vendor and version you select for your cluster and are beyond the scope of this document.

Important: Restore the databases to their exact state as of when you took the backup. Do not merge in any changes that may have occurred during the subsequent upgrade.

See the following vendor resources for more information:

- MariaDB 5.5: http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/
- MySQL 5.5: http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html
- MySQL 5.6: http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html
- MySQL 5.7: http://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html
- PostgreSQL 8.4: https://www.postgresql.org/docs/8.4/static/backup.html
- PostgreSQL 9.2: https://www.postgresql.org/docs/9.2/static/backup.html
- PostgreSQL 9.3: https://www.postgresql.org/docs/9.3/static/backup.html
- Oracle 11gR2: https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm

Start the Sentry Service

Roll Back Cloudera Search

1. Start the HDFS, Zookeeper and Sentry services.
2. Delete the instancedir created during the upgrade process:

   a. If the cluster is secured with Kerberos, run this command Otherwise skip this step.

   ```
   export ZKCLI_JVM_FLAGS="-Djava.security.auth.login.config=~/solr-jaas.co
   nf
   -DzkACLProvider=org.apache.solr.common.cloud.ConfigAwareSaslZkACLProvide
   r"
   ```

   b.
   ```
   sudo -u solr solrctl instancedir --delete localFSTemplate
   ```

   3. Start the Solr service.

   Note: If the state of one or more Solr core is down and the Solr log contains an org.apache.lucene.store. LockObtainFailedException: Lock obtain timed out: org.apache.solr.store.hdfs.HdfsLockFactory error message, it is necessary to clean up the HDFS locks in the index directories.

   For each of the affected Solr nodes:

   a. 1. Stop the Solr node using Cloudera Manager.
   b. 2. Remove the HdfsDirectory<id>-write.lock file from the index directory.

   ```
   hdfs dfs -rm "/solr/<collection_name>/<core>/data/<index_directory_name>
   /HdfsDirectory@ <hex_id> lockFactory=org.apache.solr.store.hdfs.HdfsLock
   Factory@<hex_id>-write.lock"
   ```

   For example:

   ```
   hdfs dfs -rm "/solr/testCollection/core_node1/data/index/HdfsDirectory@5
   d07feac lockFactory=org.apache.solr.store.hdfs.HdfsLockFactory@7df08aad-
   write.lock"
   ```

   c. 3. Start the Solr node using Cloudera Manager.

Roll Back Atlas

Rollback Atlas Solr Collections

Atlas has several collections in Solr that must be restored from the pre-upgrade backup - vertex_index, edge_index, and fulltext_index. These collections may already have been restored using the Roll Back Cloudera Search on page 210 documentation. If the collections are not yet restored, you must restore collections now using the Roll Back Cloudera Search on page 210 documentation.

Rollback Atlas HBase Tables

1. From a client host, start the HBase shell `hbase shell`
2. Within the HBase shell, list the snapshots, that must contain the pre-upgrade snapshots `list_snapshots`
3. Within the HBase shell, disable the atlas_janus table, restore the snapshot, and enable the table

   ```
   disable 'atlas_janus'
   restore_snapshot '<name of atlas_janus snapshot from list_snapshots>'
   enable 'atlas_janus'
   ```
4. Within the HBase shell, disable the ATLAS_ENTITY_AUDIT_EVENTS table, restore the snapshot, and enable the table

   ```
   disable 'ATLAS_ENTITY_AUDIT_EVENTS'
   restore_snapshot '<name of ATLAS_ENTITY_AUDIT_EVENTS snapshot from list_snapshots>'
   enable 'ATLAS_ENTITY_AUDIT_EVENTS'
   ```
5. Restart Atlas.

Roll Back Hue

1. Restore the file, app.reg, from your backup:
   - Parcel installations

     ```
     rm -rf /opt/cloudera/parcels/CDH/lib/hue/app.reg
     cp -rp app.reg_CM7_CDH6_BACKUP /opt/cloudera/parcels/CDH/lib/hue/app.reg
     ```
   - Package Installations

     ```
     rm -rf /usr/lib/hue/app.reg
     cp -rp app.reg_CM7_CDH6_BACKUP /usr/lib/hue/app.reg
     ```

Roll Back Kafka

A Cloudera Private Cloud Base 7 cluster that is running Kafka can be rolled back to the previous CDH5/CDK versions as long as theinter.broker.protocol.version and log.message.format.version properties have not been set to the new version or removed from the configuration.

To perform the rollback using Cloudera Manager:

1. Activate the previous CDK parcel. Please note, that when rolling back Kafka from CDP Private Cloud Base 7 to CDH 6/CDK, the Kafka cluster will restart. Rolling restart is not supported for this scenario. See Activating a Parcel.

2. Remove the following properties from the Kafka Broker Advanced Configuration Snippet (Safety Valve) configuration property.
   - Inter.broker.protocol.version
   - log.message.format.version

Deploy the Client Configuration

1. On the Cloudera Manager Home page, click the Actions menu and select Deploy Client Configuration.
2. Click Deploy Client Configuration.

Restart the Cluster

1. On the Cloudera Manager Home page, click the Actions menu and select Restart.
2. Click Restart that appears in the next screen to confirm. If you have enabled high availability for HDFS, you can choose Rolling Restart instead to minimize cluster downtime. The Command Details window shows the progress of stopping services.

   When All services successfully started appears, the task is complete and you can close the Command Details window.

Roll Back Cloudera Navigator Encryption Components

If you are rolling back any encryption components (Key Trustee Server, Key Trustee KMS, HSM KMS, Key HSM, or Navigator Encrypt), first refer to:

- Backing up and Restoring Key Trustee Server and Clients
- HSM KMS High Availability Backup and Recovery
- Manually Backing Up Navigator Encrypt

Roll Back Key Trustee Server

Note: If rolling back multiple encryption product components, it is recommended that you begin with the Key Trustee Server.

To roll back Key Trustee Server, replace the currently used parcel (for example, the parcel for version 7.1.4) with the parcel for the version to which you wish to roll back (for example, version 5.14.0). See Parcels for detailed instructions on using parcels.

The Keytrustee Server 7.x upgrades the bundled Postgres engine from version 9.3 to 12.1. The upgrade happens automatically, however, downgrading to CDH 6 requires manual steps to roll back the database engine to version 9.3. Because the previously upgraded database is left unchanged, the database server will fail start. Follow these steps to recreate the Postgres 9.3 compatible database:

1. Open the Cloudera Manager Admin Console and go to the Key Trustee Server service. If you see that Key Trustee Server has stale configurations, click the yellow or blue button and follow the prompts.
2. Make sure that the Keytrustee Server database roles are stopped. Then rename the folder containing Keytrustee Postgres database data (both on master and slave hosts):

   ```
   mv /var/lib/keytrustee/db /var/lib/keytrustee/db-12_1
   ```
3. Open the Cloudera Manager Admin Console and go to the Key Trustee Server service.
4. Select the Instances tab.
5. Select the Active Database role type.

6. Click Actions for Selected Set Up the Key Trustee Server Database .
7. Click Set Up the Key Trustee Server Database to confirm.

   Cloudera Manager sets up the Key Trustee Server database.
8. On the master KTS node: running as user keytrustee, restore the keytrustee database from the dump created during the upgrade by running the following commands:

   ```
   sudo -su keytrustee
   export HOME=/opt/cloudera/parcels/KEYTRUSTEE_SERVER
   export JAVA_HOME=... # Set this to your Java Home folder
   export PATH="/opt/cloudera/parcels/KEYTRUSTEE_SERVER/bin:/opt/cloudera/
   parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/9.3/bin:$PATH"
   source /opt/cloudera/parcels/KEYTRUSTEE_SERVER/meta/keytrustee_env.sh
   dropdb -p 11381 keytrustee
   ```

   If you see the message: could not change directory to "/root: Permission denied on the console, run the following command to check the exit code of the last command:

   ```
   $?
   ```

   You can use the exit code to debug any issues.
9. Run the following command to import a database dump that was created during upgrade:

   ```
   psql -p 11381 postgres -f /var/lib/keytrustee/.keytrustee/kt93dump.pg
   ```
10. Start the Active Database role in Cloudera Manager by clicking Actions for Selected Start .
11. Click Start to confirm.
12. Select the Active Database.
13. Click Actions for Selected Setup Enable Synchronous Replication in HA mode .
14. Start the Passive Database instance: select the Passive Database, click Actions for Selected Start .
15. In the Cloudera Manager Admin Console, start the active KTS instance.
16. In the Cloudera Manager Admin Console, start the passive KTS instance.

Start the Key Management Server

Restart the Key Management Server. Open the Cloudera Manager Admin Console, go to the KMS service page, and select Actions Start .

Roll Back Key HSM

To roll back Key HSM:

1. Install the version of Navigator Key HSM to which you wish to roll back

   Install the Navigator Key HSM package using yum:

   ```
   sudo yum downgrade keytrustee-keyhsm
   ```

   Cloudera Navigator Key HSM is installed to the /usr/share/keytrustee-server-keyhsm directory by default.
2. Rename Previously-Created Configuration Files

   For Key HSM major version rollbacks, previously-created configuration files do not authenticate with the HSM and Key Trustee Server, so you must recreate these files by re-executing the setup and trust commands. First, navigate to the Key HSM installation directory and rename the applications.properties, keystore, and truststore files:

   ```
   cd /usr/share/keytrustee-server-keyhsm/
   mv application.properties application.properties.bak
   mv keystore keystore.bak
   ```

```
mv truststore truststore.bak
```

3. Initialize Key HSM

   Run the service keyhsm setup command in conjunction with the name of the target HSM distribution:

   ```
   sudo service keyhsm setup [keysecure|thales|luna]
   ```

   For more details, see Initializing Navigator Key HSM.
4. Establish Trust Between Key HSM and the Key Trustee Server

   The Key HSM service must explicitly trust the Key Trustee Server certificate (presented during TLS handshake). To establish this trust, run the following command:

   ```
   sudo keyhsm trust /path/to/key_trustee_server/cert
   ```

   For more details, see Establish Trust from Key HSM to Key Trustee Server.
5. Start the Key HSM Service

   Start the Key HSM service:

   ```
   sudo service keyhsm start
   ```
6. Establish Trust Between Key Trustee Server and Key HSM

   Establish trust between the Key Trustee Server and the Key HSM by specifying the path to the private key and certificate:

   ```
   sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 \
   --client-certfile /etc/pki/cloudera/certs/mycert.crt \
   --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
   ```

   For a password-protected Key Trustee Server private key, add the --passphrase argument to the command (enter the password when prompted):

   ```
   sudo ktadmin keyhsm --passphrase \
   --server https://keyhsm01.example.com:9090 \
   --client-certfile /etc/pki/cloudera/certs/mycert.crt \
   --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
   ```

   For additional details, see Integrate Key HSM and Key Trustee Server.
7. Remove Configuration Files From Previous Installation

   After completing the rollback, remove the saved configuration files from the previous installation:

   ```
   cd /usr/share/keytrustee-server-keyhsm/
   rm application.properties.bak
   rm keystore.bak
   rm truststore.bak
   ```

Roll Back Key Ranger KMS Parcels

Enable the desired parcel that you wish to roll back to (for example, version 6.3.4 of Key Trustee KMS). See Parcels for detailed instructions on using parcels. See Parcels for detailed instructions on using parcels.

Roll Back HSM KMS Parcels

To roll back the HSM KMS parcels, replace the currently used parcel (for example, the parcel for version 6.0.0) with the parcel for the version to which you wish to roll back (for example, version 5.14.0). See Parcels for detailed instructions on using parcels.

See Upgrading HSM KMS Using Packages for detailed instructions on using packages.

Roll Back Navigator Encrypt

To roll back Cloudera Navigator Encrypt:

1. If you have configured and are using an RSA master key file with OAEP padding, then you must revert this setting to its original value:

```
navencrypt key --change
```

2. Stop the Navigator Encrypt mount service:

```
sudo /etc/init.d/navencrypt-mount stop
```

3. Confirm that the mount-stop command completed:

```
sudo /etc/init.d/navencrypt-mount status
```

4. Backup the NavEncrypt control directory:

```
sudo mkdir navencryptBAK
sudo cp -rp /etc/navencrypt/ navencryptBAK/
```

5. Clean the dkms/navencryptfs directory:

```
sudo rm -rf /var/lib/dkms/navencryptfs/
```

6. If rolling back to a release lower than NavEncrypt 6.2:
   a. Print the existing ACL rules and save that output to a file:

```
sudo navencrypt acl --print+ vim acls.txt
```

   b. Delete all existing ACLs, for example, if there are a total of 7 ACL rules run:

```
sudo navencrypt acl --del --line=1,2,3,4,5,6,7
```

7. To fully downgrade Navigator Encrypt, manually downgrade all of the associated Navigator Encrypt packages (in the order listed):
   a. navencrypt
   b. navencrypt-kernel-module (Only required for operating systems other than SLES)
   c. cloudera-navencryptfs-kmp (Only required for the SLES operating system)
   d. libkeytrustee
8. If rolling back to a release less than NavEncrypt 6.2
   a. Reapply the ACL rules:

```
sudo navencrypt acl --add --file=acls.txt
```

   b. Recompute process signatures:

```
sudo navencrypt acl --update
```

9. Restart the Navigator Encrypt mount service

```
sudo /etc/init.d/navencrypt-mount start
```

(Optional) Cloudera Manager Rollback Steps

After you complete the rollback steps, your cluster is using Cloudera Manager 7 to manage your CDH 6 or CDH 6 cluster. You can continue to use Cloudera Manager 7 to manage your CDH 6 cluster, or you can downgrade to Cloudera Manager 6 by following these steps:

Stop Cloudera Manager

1. Stop the .
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Stop .
2. Stop the Server.

```
sudo systemctl stop cloudera-scm-server
```

3. Hard stop the Cloudera Manager agents. Run the following command on all hosts:

```
sudo systemctl stop cloudera-scm-supervisord.service
```

4. Back up the repository directory. You can create a top-level backup directory and an environment variable to reference the directory using the following commands. You can also substitute another directory path in the backup commands below:

```
export CM_BACKUP_DIR="`date +%F`-CM"
mkdir -p $CM_BACKUP_DIR
```

5. Back up the existing repository directory.
   RHEL / CentOS

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
```

   SLES

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
```

   Ubuntu

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources
.list.d
```

Restore the Cloudera Manager 6 Repository Files

Copy the repository directory from the backup taken before upgrading to Cloudera Manager 7.x.

```
rm -rf /etc/yum.repos.d/*
tar -xf cm6cdh6_backedUp_dir/repositary.tar -C CM6CDH6/
cp -rp /etc/yum.repos.d_CM6CDH6/* /etc/yum.repos.d/
```

Restore Packages

1. Run the following commands on all hosts:

Operating System	Command

RHEL	`sudo yum remove cloudera-manager-daemons cloudera-manager-agent`
	`sudo yum clean all`
	`sudo yum install cloudera-manager-agent`

SLES	`sudo zypper remove cloudera-manager-daemons cloudera-manager-agent`
	`sudo zypper refresh -s`
	`sudo zypper install cloudera-manager-agent`

Ubuntu or Debian	`sudo apt-get purge cloudera-manager-daemons cloudera-manager-agent`
	`sudo apt-get update`
	`sudo apt-get install cloudera-manager-agent`

2. Run the following commands on the Cloudera Manager server host:

Operating System	Command
RHEL	`sudo yum remove cloudera-manager-server` `sudo yum install cloudera-manager-server`
SLES	`sudo zypper remove cloudera-manager-server` `sudo zypper install cloudera-manager-server`
Ubuntu or Debian	`sudo apt-get purge cloudera-manager-server` `sudo apt-get install cloudera-manager-server`

Restore Cloudera Manager Databases

Restore the Cloudera Manager databases from the backup of Cloudera Manager that was taken before upgrading to Cloudera Manager 7. See the procedures provided by your database vendor.

These databases include the following:

- Cloudera Manager Server
- Reports Manager
- Activity Monitor (Only used for MapReduce 1 monitoring).

- MariaDB 5.5: http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/
- MySQL 5.5: http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html

- MySQL 5.6: http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html
- PostgreSQL 8.4: https://www.postgresql.org/docs/8.4/static/backup.html
- PostgreSQL 9.2: https://www.postgresql.org/docs/9.2/static/backup.html
- PostgreSQL 9.3: https://www.postgresql.org/docs/9.3/static/backup.html
- Oracle 11gR2: https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm
- HyperSQL: http://hsqldb.org/doc/guide/management-chapt.html#mtc_backup

Here is an sample command to restore a MySQL database:

```
mysql -u username -ppassword --host=hostname cm < backup.sql
```

Restore Cloudera Manager Server

Use the backup of Cloudera Manager 6.x taken before upgrading to Cloudera Manager 7.x for the following steps:

1. If you used the backup commands provided in Step 2: Backing Up Cloudera Manager 6 on page 71, extract the Cloudera Manager 6 backup archives you created:

   ```
   tar -xf CM6CDH6/cloudera-scm-agent.tar -C CM6CDH6/
   tar -xf CM6CDH6/cloudera-scm-server.tar -C CM6CDH6/
   ```

2. On the host where the Event Server role is configured to run, restore the Events Server directory from the Cloudera Manager 6 backup.

   ```
   cp -rp /var/lib/cloudera-scm-eventserver /var/lib/cloudera-scm-eventserv
   er-CM
   rm -rf /var/lib/cloudera-scm-eventserver/*
   cp -rp /var/lib/cloudera-scm-eventserver_CM6CDH6/* /var/lib/cloudera-scm-
   eventserver/
   ```

3. Remove the Agent runtime state. Run the following command on all hosts:

   ```
   rm -rf /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent/response.
   avro
   ```

4. On the host where the Service Monitor is running, restore the Service Monitor directory:

   ```
   rm -rf /var/lib/cloudera-service-monitor/*
   cp -rp /var/lib/cloudera-service-monitor_CM6CDH6/* /var/lib/cloudera-ser
   vice-monitor/
   ```

5. On the host where the Host Monitor is running, restore the Host Monitor directory:

   ```
   rm -rf /var/lib/cloudera-host-monitor/*
   cp -rp /var/lib/cloudera-host-monitor_CM6CDH6/* /var/lib/cloudera-host-mo
   nitor/
   ```

6. Restore the Cloudera Navigator Solr storage directory from the CM6/CDH6 backup.

   ```
   rm -rf /var/lib/cloudera-scm-navigator/*
   cp -rp /var/lib/cloudera-scm-navigator_CM6CDH6/* /var/lib/cloudera-scm-nav
   igator/
   ```

7. On the Cloudera Manager Server, restore the /etc/cloudera-scm-server/db.properties file.

   ```
   rm -rf /etc/cloudera-scm-server/db.properties
   cp -rp CM6CDH6/etc/cloudera-scm-server/db.properties /etc/cloudera-scm-ser
   ver/db.properties
   ```

8. On each host in the cluster, restore the /etc/cloudera-scm-agent/config.ini file from your backup.

```
rm -rf /etc/cloudera-scm-agent/config.ini
cp -rp CM6CDH6/etc/cloudera-scm-agent/config.ini /etc/cloudera-scm-agent/c
onfig.ini
```

Start the Cloudera Manager Server and Agents

- Start the Server.

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

- Hard Restart the Agent.
  RHEL 7, SLES 12, Ubuntu 18.04 and higher

```
sudo systemctl stop cloudera-scm-supervisord.service
sudo systemctl restart cloudera-scm-agent
```

- Start the .

  1. Log in to the Admin Console.
  2. Select Clusters .
  3. Select Actions Start .

Rolling Back a Cloudera Private Cloud Base Upgrade from versions 7.1.1 - 7.1.7 to CDH 6

You can roll back an upgrade from Cloudera Private Cloud Base 7 to CDH 6. The rollback restores your CDH cluster to the state it was in before the upgrade, including Kerberos and TLS/SSL configurations.

Important: Any data created after the upgrade is lost.

In a typical upgrade, you first upgrade Cloudera Manager from version 6.x to version 7.x, and then you use the upgraded version of Cloudera Manager 7 to upgrade CDH 6 to Cloudera Private Cloud Base 7. (See Upgrading a CDH 6 Cluster on page 121.) If you want to roll back this upgrade, follow these steps to roll back your cluster to its state prior to the upgrade.

You can roll back to CDH 6 after upgrading to Cloudera Private Cloud Base 7 only if the HDFS upgrade has not been finalized. The rollback restores your CDH cluster to the state it was in before the upgrade, including Kerberos and TLS/SSL configurations.

Important: Follow all of the steps in the order presented in this topic. Cloudera recommends that you read through the backup and rollback steps before starting the backup process. You may want to create a detailed plan to help you anticipate potential problems.

Important:

These rollback steps depend on complete backups taken before upgrading Cloudera Manager and CDH. See Step 2: Backing Up Cloudera Manager 6 on page 71 and Step 3: Backing Up the Cluster on page 134.

For steps where you need to restore the contents of a directory, clear the contents of the directory before copying the backed-up files to the directory. If you fail to do this, artifacts from the original upgrade can cause problems if you attempt the upgrade again after the rollback.

Review Limitations

The rollback procedure has the following limitations:

- HDFS – If you have finalized the HDFS upgrade, you cannot roll back your cluster.
- Compute clusters – Rollback for Compute clusters is not supported. You must remove any compute clusters before rolling back.
- Configuration changes, including the addition of new services or roles after the upgrade, are not retained after rolling back Cloudera Manager.

  Cloudera recommends that you not make configuration changes or add new services and roles until you have finalized the HDFS upgrade and no longer require the option to roll back your upgrade.
- HBase – If your cluster is configured to use HBase replication, data written to HBase after the upgrade might not be replicated to peers when you start your rollback. This topic does not describe how to determine which, if any, peers have the replicated data and how to roll back that data. For more information about HBase replication, see HBase Replication.
- Sqoop 2 – As described in the upgrade process, Sqoop2 had to be stopped and deleted before the upgrade process and therefore will not be available after the rollback.
- Kafka – Once the Kafka log format and protocol version configurations (the inter.broker.protocol.version and log.message.format.version properties) are set to the new version (or left blank, which means to use the latest version), Kafka rollback is not possible.

Stop the Cluster

1. If HBase is deployed in the cluster do the following before stopping the cluster:

   The HBase Master procedures changed between the two versions, so if a procedure was started by HBase 2.2 (CDP 7.x) then the older HBase 2.1 won't be able to continue the procedure after the rollback. For this reason the Procedure Store in HBase Master must be cleaned before the rollback. If CDP 7.x HBase Master was never started, then the rollback should be fine. But if HBase Master was running with the new version and there is any ongoing (or stuck) HBase Master Procedure present in the CDP 7 HBase Master, then the older CDH 6 HBase Master will fail to start after the rollback. If this happens, HBase will need manual fix after the rollback (e.g. the sidelining of the HBase Master Procedure WAL files and the potential fixing of inconsistencies in HBase).

   To avoid this problem, you should try to verify that no unfinished procedure is present before stopping HBase Master on the CDP 7.x Cluster. Please follow these steps:

   a. Make sure there was no traffic running against the HBase Cluster recently (in the last 10 minutes) that can trigger e.g. table creation or deletion, region assignment or split or merge, etc.
   b. Disable automatic Balancer and Normalizer in HBase. Also disable Split and Merge procedures, before stopping the CDP 7 Cluster. All these tools in HBase can cause the starting of new HBase Master Procedures, which we want to avoid now. Issue the following commands in HBase Shell:

      ```
      balance_switch false
      normalizer_switch false
      splitormerge_switch 'SPLIT', false
      splitormerge_switch 'MERGE', false
      ```

   c. Check the list of procedures on the HBase Master Web UI (In Cloudera Manager, go to the HBase service and open the HBase Web UIProcedures & Locks tab). Wait until you see procedures only with final states like 'SUCCESS', 'FAILED' or 'ROLLEDBACK'.
   d. Get the list of procedures from HBase shell using the 'list_procedures' command. Wait until you see procedures only with final states like 'SUCCESS', 'FAILED' or 'ROLLEDBACK'. The State appears in the third column of the table returned by the 'list_procedures' command.

   If the HBase Master doesn't start after the rollback and some procedure-related exceptions are found in the role logs (like "BadProcedureException" or decode errors in the "ProcedureWALFormatReader" class, or "ClassNotFoundException" for procedure classes), then this is most likely caused by CDP 7 procedures that still remain in the procedure WAL files. In this case, please open a ticket for Cloudera customer support, who will help you to sideline the procedure WAL files and fix any potential inconsistencies in HBase.

2. On the HomeStatus tab, click the Actions menu and select Stop.
3. 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.
4. Go to the YARN service and click ActionsClean NodeManager Recovery Directory. The CDH 6 NodeManager will not start up after the downgrade if it finds CDP 7.x data in the recovery directory. The format and content of the NodeManager's recovery state store was changed between CDH 6.x and CDP 7.x. The recovery directory used by CDP 7.x must be cleaned up as part of the downgrade to CDH 6.

(Parcels) Downgrade the Software

Follow these steps only if your cluster was upgraded using Cloudera parcels.

1. Log in to the Cloudera Manager Admin Console.
2. Select HostsParcels.

   A list of parcels displays.
3. Locate the CDH 6 parcel and click Activate. (This automatically deactivates the Cloudera Private Cloud Base 7 parcel.) See Activating a Parcel for more information. If the parcel is not available, use the Download button to download the parcel.

   Important: If you added new services that are not part of CDH 6 during the upgrade, a list of services that need to be deleted displays. You must delete these services before activating the CDH 6 parcel.
4. If you include any additional components in your cluster, such as Search or Impala, click Activate for those parcels.

   Important:

   Do not start any services. (Select the Activate Only option.)

   If you accidentally restart services, stop your cluster before proceeding.

Stop Cloudera Manager

1. Stop the .
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Stop .
2. Stop the Server.

```
sudo systemctl stop cloudera-scm-server
```

3. Hard stop the Cloudera Manager agents. Run the following command on all hosts:

```
sudo systemctl stop cloudera-scm-supervisord.service
```

Restore Cloudera Manager Databases

Restore the Cloudera Manager databases from the backup of Cloudera Manager that was taken before upgrading the cluster toCloudera Private Cloud Base 7.

Important: Cloudera recommends that you run the TRUNCATE statement on your database before you start the restore procedure to make sure that there are no remanent tables in the database from a failed upgrade attempt. The truncate statement/procedure may differ slightly depending on the backend database being used in your environment.

See the procedures provided by your database vendor.

- MariaDB 5.5: http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/
- MySQL 5.5: http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html
- MySQL 5.6: http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html
- PostgreSQL 8.4: https://www.postgresql.org/docs/8.4/static/backup.html
- PostgreSQL 9.2: https://www.postgresql.org/docs/9.2/static/backup.html
- PostgreSQL 9.3: https://www.postgresql.org/docs/9.3/static/backup.html
- Oracle 11gR2: https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm
- HyperSQL: http://hsqldb.org/doc/guide/management-chapt.html#mtc_backup

Restore Cloudera Manager Server

Use the backup of CDH that was taken before the upgrade to restore Cloudera Manager Server files and directories. Substitute the path to your backup directory for *CM7_CDH6* in the following steps:

1. On the host where the Event Server role is configured to run, restore the Events Server directory from the CM 7/CDH 6 backup.

```
cp -rp /var/lib/cloudera-scm-eventserver /var/lib/cloudera-scm-eventserv
er-CM<ph outputclass="cdoc-specific-cm-from"/>-CDH<ph outputclass="cdoc-
specific-cdh-from"/>
rm -rf /var/lib/cloudera-scm-eventserver/*
cp -rp /var/lib/cloudera-scm-eventserver_CM7_CDH6/* /var/lib/cloudera-scm-
eventserver/
```

2. Remove the Agent runtime state. Run the following command on all hosts:

```
rm -rf /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent/response.
avro
```

   This command may return a message similar to: rm: cannot remove ‘/var/run/cloudera-scm-agent/process’: D evice or resource busy. You can ignore this message.

3. On the host where the Service Monitor is running, restore the Service Monitor directory:

```
rm -rf /var/lib/cloudera-service-monitor/*
cp -rp /var/lib/cloudera-service-monitor_CM7_CDH6/* /var/lib/cloudera-ser
vice-monitor/
```

4. On the host where the Host Monitor is running, restore the Host Monitor directory:

```
rm -rf /var/lib/cloudera-host-monitor/*
cp -rp /var/lib/cloudera-host-monitor_CM7_CDH6/* /var/lib/cloudera-host-mo
nitor/
```

5. Restore the Cloudera Navigator storage directory from the CM 7/CDH 6 backup.

```
rm -rf /var/lib/cloudera-scm-navigator/*
cp -rp /var/lib/cloudera-scm-navigator_CM7_CDH6/* /var/lib/cloudera-scm-
navigator/
```

Start Cloudera Manager

1. Log in to the Cloudera Manager server host.
2. Start the Server.

   ```
   sudo systemctl start cloudera-scm-server
   ```
3. Start the Cloudera Manager Agent.

   Run the following commands on all cluster hosts:

   ```
   sudo systemctl start cloudera-scm-agent
   ```
4. Start the Cloudera Management Service.
   a. Log in to the Cloudera Manager Admin Console.
   b. Select ClustersCloudera Management Service.
   c. Select ActionsStart.

   The cluster page may indicate that services are in bad health. This is normal.
5. Stop the cluster. In the Cloudera Manager Admin Console, click the Actions menu for the cluster and select Stop.

Roll Back ZooKeeper

1. Using the backup of Zookeeper that you created when backing up your CDH 6.x cluster, restore the contents of the *DATADIR* on each ZooKeeper server. These files are located in a directory specified with the dataDir property in the ZooKeeper configuration. The default location is /var/lib/zookeeper. For example:

   ```
   rm -rf /var/lib/zookeeper/*
   cp -rp /var/lib/zookeeper_CM7_CDH6/* /var/lib/zookeeper/
   ```
2. Using the backup of Zookeeper that you created when backing up your CDH 6.x cluster, restore the contents of the *TRANSACTION LOG DIRECGTORY* on each ZooKeeper server. These files are located in a directory specified with the Transaction Log Direcgtory property in the ZooKeeper configuration. For example:

   ```
   rm -rf /var/lib/zookeeper/*
   cp -rp /var/lib/zookeeper_CM7_CDH6/* /var/lib/zookeeper/
   ```
3. Make sure that the permissions of all the directories and files are as they were before the upgrade.
4. Start ZooKeeper using Cloudera Manager.

Roll Back HDFS

You cannot roll back HDFS while high availability is enabled. The rollback procedure in this topic creates a temporary configuration without high availability. Regardless of whether high availability is enabled, follow the steps in this section.

1. Roll back all of the Journal Nodes. (Only required for clusters where high availability is enabled for HDFS). Use the JournalNode backup you created when you backed up HDFS before upgrading to Cloudera Private Cloud Base.
   a. Log in to each Journal Node host and run the following commands:

   ```
   rm -rf /dfs/jn/ns1/current/*
   ```

   ```
   cp -rp <JOURNAL_NODE_BACKUP_DIRECTORY>/ns1/current/* /dfs/jn/ns1/curren
   t/
   ```

   b. Start the JournalNodes using Cloudera Manager:
      1. Go to the HDFS service.
      2. Select the Instances tab.
      3. Select all JournalNode roles from the list.
      4. Click Actions for Selected Start .
2. Roll back all of the NameNodes. Use the NameNode backup directory you created before upgrading to Cloudera Private Cloud Base. (/etc/hadoop/conf.rollback.namenode) to perform the following steps on all NameNode hosts:
   a. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file on all NameNode hosts (located in the temporary rollback directory) and update the keystore passwords with the actual cleartext passwords.

      The passwords will have values that look like this:

      ```
      <property>
          <name>ssl.server.keystore.password</name>
          <value>********</value>
        </property>
        <property>
          <name>ssl.server.keystore.keypassword</name>
          <value>********</value>
        </property>
      ```

   b. (TLS only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file and remove the hadoop.security.credential.provider.path property.
3. Edit the /etc/hadoop/conf.rollback.namenode/hdfs-site.xml file on all NameNode hosts and make the following changes:
   a. Update the dfs.namenode.inode.attributes.provider.class property. If Sentry was installed prior to the upgrade, change the value of the property from org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer to "org.apache.sentry.hdfs.SentryINodeAttributesProvider. If Sentry was not installed, remove this property.
   b. Change the path in the dfs.hosts property to the value shown in the example below. The file name, dfs_all_hosts.txt, may have been changed by a user. If so, substitute the correct file name.

      ```
      # Original version of the dfs.hosts property:
      <property>
      <name>dfs.hosts</name>
      <value>/var/run/cloudera-scm-agent/process/63-hdfs-NAMENODE/dfs_all_host
      s.txt</value>
      </property>
      ```

      ```
      # New version of the dfs.hosts property:
      <property>
      <name>dfs.hosts</name>
      <value>/etc/hadoop/conf.rollback.namenode/dfs_all_hosts.txt</value>
      </property>
      ```

c. Remove the property that has the following value:

```
com.cloudera.navigator.audit.hdfs.HdfsAuditLoggerCdh5
```

4. Edit the /etc/hadoop/conf.rollback.namenode/core-site.xml and change the value of the net.topology.script.file.name property to /etc/hadoop/conf.rollback.namenode. For example:

```
# Original property
<property>
<name>net.topology.script.file.name</name>
<value>/var/run/cloudera-scm-agent/process/63-hdfs-NAMENODE/topology.py</value>
</property>
```

```
# New property
<property>
<name>net.topology.script.file.name</name>
<value>/etc/hadoop/conf.rollback.namenode/topology.py</value>
</property>
```

5. Edit the /etc/hadoop/conf.rollback.namenode/topology.py file and change the value of MAP_FILE to /etc/hadoop/conf.rollback.namenode. For example:

```
MAP_FILE = '/etc/hadoop/conf.rollback.namenode/topology.map'
```

6. (TLS-enabled clusters only) Run the following command:

```
sudo -u hdfs kinit hdfs/<NAMENODE HOST NAME> -l 7d -kt /etc/hadoop/conf.rollback.namenode/hdfs.keytab
```

7. Run the following command:

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.namenode namenode -rollback
```

8. Restart the NameNodes and JournalNodes using Cloudera Manager:
   a. Go to the HDFS service.
   b. Select the Instances tab, and then select all Failover Controller, NameNode, and JournalNode roles from the list.
   c. Click Actions for SelectedRestart.
9. Rollback the DataNodes.

   Use the DataNode rollback directory you created before upgrading to Cloudera Private Cloud Base (/etc/hadoop/conf.rollback.datanode) to perform the following steps on all DataNode hosts:

   a. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.datanode/ssl-server.xml file on all DataNode hosts (Located in the temporary rollback directory.) and update the keystore passwords (ssl.server.keystore.password and ssl.server.keystore.keypassword) with the actual passwords.

      The passwords will have values that look like this:

```
<property>
    <name>ssl.server.keystore.password</name>
    <value>********</value>
  </property>
  <property>
    <name>ssl.server.keystore.keypassword</name>
    <value>********</value>
  </property>
```

b. (Clusters with TLS only) Edit the /etc/hadoop/conf.rollback.datanode/ssl-server.xml file and remove the hadoop.security.credential.provider.path property.

c. (Clusters with Kerberos enabled only) Modify the Kerberos file (hdfs.keytab) path in /etc/hadoop/conf.rollback.datanode/core-site.xml and /etc/hadoop/conf.rollback.datanode/hdfs-site.xml files to the absolute path. For example, the absolute path of the Kerberos file can be /etc/hadoop/conf.rollback.datanode/hdfs.keytab.

d. Edit the /etc/hadoop/conf.rollback.datanode/hdfs-site.xml file and remove the dfs.datanode.max.locked.memory property.

e. Run one of the following commands:

- Run as root if the DataNodes use reserved ports.

  Search in the logs for completed rollback line. It will not be shown in the command line of the rollback.
- If the DataNode is running with privileged ports (usually 1004 and 1006):

```
cd /etc/hadoop/conf.rollback.datanode
export HADOOP_SECURE_DN_USER=hdfs
export JSVC_HOME=/opt/cloudera/parcels/<PARCEL_FILENAME>/lib/bigtop-utils
hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
```

- If the DataNode is not running on privileged ports:

```
cd /etc/hadoop/conf.rollback.datanode
sudo hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
```

  You may see the following error after issuing these commands:

```
ERROR datanode.DataNode: Exception in secureMain
java.io.IOException: The path component: '/var/run/hdfs-sockets' in
 '/var/run/hdfs-sockets/dn' has permissions 0755 uid 39998 and gid 1006.
It is not protected because it is owned by a user who is not root
and not the effective user: '0'.
```

  The error message will also include the following command to run:

```
chown root /var/run/hdfs-sockets
```

  After running this command, rerun the DataNode rollback command:

```
sudo hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
```

  The DataNodes will now restart successfully.

When the rolling back of the DataNodes is complete, terminate the console session by typing Control-C. Look for output from the command similar to the following that indicates when the DataNode rollback is complete:

```
Rollback of /dataroot/ycloud/dfs/dn/current/BP-<Block Group number> is complete
```

f. If High Availability for HDFS is enabled, restart the HDFS service. In the Cloudera Manager Admin Console, go to the HDFS service and select Actions Restart .

g. If high availability is not enabled for HDFS, use the Cloudera Manager Admin Console to restart all NameNodes and DataNodes.

1. Go to the HDFS service.
2. Select the Instances tab
3. Select all DataNode and NameNode roles from the list.
4. Click Actions for Selected Restart .

10. If high availability is not enabled for HDFS, roll back the Secondary NameNode.

a. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.secondarynamenode/ssl-server.xml file on all Secondary NameNode hosts (Located in the temporary rollback directory.) and update the keystore passwords with the actual cleartext passwords.

The passwords will have values that look like this:

```
<property>
    <name>ssl.server.keystore.password</name>
    <value>********</value>
  </property>
  <property>
    <name>ssl.server.keystore.keypassword</name>
    <value>********</value>
  </property>
```

b. (TLS only) Edit the /etc/hadoop/conf.rollback.secondarynamenode/ssl-server.xml file and remove the hadoop.security.credential.provider.path property.

c. Log in to the Secondary NameNode host and run the following commands:

```
rm -rf /dfs/snn/*
cd /etc/hadoop/conf.rollback.secondarynamenode/
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.secondarynamenode
 secondarynamenode -format
```

When the rolling back of the Secondary NameNode is complete, terminate the console session by typing Control-C. Look for output from the command similar to the following that indicates when the Secondary NameNode rollback is complete:

```
2020-12-21 17:09:36,239 INFO namenode.SecondaryNameNode: Web server init
 done
```

11. Restart the HDFS service. Open the Cloudera Manager Admin Console, go to the HDFS service page, and select ActionsRestart.

The Restart Command page displays the progress of the restart. Wait for the page to display the Successfully restarted service message before continuing.

Start the HBase Service

Restart the HBase Service. Open the Cloudera Manager Admin Console, go to the HBase service page, and select ActionsStart.

If you have configured any HBase coprocessors, you must revert them to the versions used before the upgrade.

If CDP 7.x HBase Master was started after the upgrade and there was any ongoing (or stuck) HBase Master Procedure present in the HBase Master before stopping the CDP 7 Cluster, then it is expected for the CDH 6 HBase Master to fail with warnings and errors in the role log from the classes like 'ProcedureWALFormatReader' and 'WALProcedureStore' or 'TransitRegionStateProcedure'.

These errors mean that the HBase Master Write-Ahead Log files are incompatible with the CDH 6 HBase version. The only way to fix this problem is to sideline the log files (all the files placed under /hbase/MasterProcWALs by

default), then restart the HBase Master. After the HBase Master has started, Use the HBCK command to find out if there are any inconsistencies that will need to be fixed manually.

Attention: You must follow the below instructions if you are rolling back from the 7.1.7.2000 version.

You may encounter other errors when starting HBase (for example, replication-related problems, region assignment related issues, and meta region assignment problems). In this case you should delete the znode in ZooKeeper and then start HBase again. (This will delete replication peer information and you will need to re-configure your replication schedules.):

1. In Cloudera Manager, look up the value of the zookeeper.znode.parent property. The default value is /hbase.
2. Connect to the ZooKeeper ensemble by running the following command from any HBase gateway host:

   ```
   zookeeper-client -server ZOOKEEPER_ENSEMBLE
   ```

   To find the value to use for *ZOOKEEPER_ENSEMBLE*, open the /etc/hbase/conf.cloudera.<*HBASE SERVICE NAME*>/hbase-site.xml file on any HBase gateway host. Use the value of the hbase.zookeeper.quorum property.

   Note: If you have deployed a secure cluster, you must connect to ZooKeeper using a client jaas.conf file. You can find such a file in an HBase process directory (/var/run/cloudera-scm-agent/process/). Specify the jaas.conf using the JVM flags by running the following commands in the ZooKeeper client:

   ```
   CLIENT_JVMFLAGS=
    "-Djava.security.auth.login.config=/var/run/cloudera-scm-agent/pro
   cess/HBASE_PROCESS_DIRECTORY/jaas.conf"
   ```

   Attention: If you have deployed a secure cluster 7.1.7.2000, you must connect to ZooKeeper using a client jaas.conf file. You can find such a file in an HBase process directory (/var/run/cloudera-scm-agent/process/). If the JAAS config file has "useTicketCache=false", make a copy of this file and replace it with "useTicketCache=true". Make sure the kerberos principal name matches with the name expected by the jaas.conf file. Specify the modified jaas.conf using the JVM flags by running the following commands in the ZooKeeper client:

   ```
   CLIENT_JVMFLAGS=
   "-Djava.security.auth.login.config=/var/run/cloudera-scm-agent/pro
   cess/HBase_process_directory/<modified
            jaas.conf file name>"
   ```

   zookeeper-client -server <zookeeper_ensemble>

   The ZooKeeper command-line interface opens.
3. Enter the following command:

   ```
   rmr /hbase
   ```

   Note: If you have deployed a secure cluster 7.1.7.2000 , enter the following command:

   ```
   deleteall /hbase
   ```

   If you see the message “Node not empty: /hbase/tokenauth”, you must re-run the same command.
4. Restart HBase service.
5. After HBase is healthy, make sure you restore the states of the Balancer and Normalizer (enable them if they were enabled before the rollback). Also re-enable the Merge and Split operations you disabled before the rollback to avoid the Master Procedure incompatibility problem. Run the following commands in HBase Shell:

   ```
   balance_switch true
   normalizer_switch true
   splitormerge_switch 'SPLIT', true
   ```

```
splitormerge_switch 'MERGE', true
```

Restore CDH Databases

Restore the following databases from the CDH 6 backups:

- Hive Metastore
- Hue
- Oozie
- Sentry Server

The steps for backing up, truncating, and restoring databases differ depending on the database vendor and version you select for your cluster and are beyond the scope of this document.

Important: Cloudera recommends that you run the TRUNCATE statement on your database before you start the restore procedure to make sure that there are no remanent tables in the database from a failed upgrade attempt. The truncate statement/procedure may differ slightly depending on the backend database being used in your environment.

Important: Restore the databases to their exact state as of when you took the backup. Do not merge in any changes that may have occurred during the subsequent upgrade.

See the following vendor resources for more information:

- MariaDB 5.5: http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/
- MySQL 5.5: http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html
- MySQL 5.6: http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html
- MySQL 5.7: http://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html
- PostgreSQL 8.4: https://www.postgresql.org/docs/8.4/static/backup.html
- PostgreSQL 9.2: https://www.postgresql.org/docs/9.2/static/backup.html
- PostgreSQL 9.3: https://www.postgresql.org/docs/9.3/static/backup.html
- Oracle 11gR2: https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm

Start the Sentry Service

Roll Back Cloudera Search

1. Start the HDFS, Zookeeper and Sentry services.
2. Delete the instancedir created during the upgrade process:
   a. If the cluster is secured with Kerberos, run this command Otherwise skip this step.

```
export ZKCLI_JVM_FLAGS="-Djava.security.auth.login.config=~/solr-jaas.conf
```

```
-DzkACLProvider=org.apache.solr.common.cloud.ConfigAwareSaslZkACLProvide
r"
```

b.

```
sudo -u solr solrctl instancedir --delete localFSTemplate
```

3. Start the Solr service.

Note: If the state of one or more Solr core is down and the Solr log contains an org.apache.lucene.store. LockObtainFailedException: Lock obtain timed out: org.apache.solr.store.hdfs.HdfsLockFactory error message, it is necessary to clean up the HDFS locks in the index directories.

For each of the affected Solr nodes:

a. 1. Stop the Solr node using Cloudera Manager.

b. 2. Remove the HdfsDirectory<id>-write.lock file from the index directory.

```
hdfs dfs -rm "/solr/<collection_name>/<core>/data/<index_directory_name>
/HdfsDirectory@ <hex_id> lockFactory=org.apache.solr.store.hdfs.HdfsLock
Factory@<hex_id>-write.lock"
```

For example:

```
hdfs dfs -rm "/solr/testCollection/core_node1/data/index/HdfsDirectory@5
d07feac lockFactory=org.apache.solr.store.hdfs.HdfsLockFactory@7df08aad-
write.lock"
```

c. 3. Start the Solr node using Cloudera Manager.

Roll Back Hue

1. Restore the file, app.reg, from your backup:
   - Parcel installations

     ```
     rm -rf /opt/cloudera/parcels/CDH/lib/hue/app.reg
     cp -rp app.reg_CM7_CDH6_BACKUP /opt/cloudera/parcels/CDH/lib/hue/app.reg
     ```

   - Package Installations

     ```
     rm -rf /usr/lib/hue/app.reg
     cp -rp app.reg_CM7_CDH6_BACKUP /usr/lib/hue/app.reg
     ```

Roll Back Kafka

A Cloudera Private Cloud Base 7 cluster that is running Kafka can be rolled back to the previous CDH5/CDK versions as long as theinter.broker.protocol.version and log.message.format.version properties have not been set to the new version or removed from the configuration.

To perform the rollback using Cloudera Manager:

1. Activate the previous CDK parcel. Please note, that when rolling back Kafka from CDP Private Cloud Base 7 to CDH 6/CDK, the Kafka cluster will restart. Rolling restart is not supported for this scenario. See Activating a Parcel.
2. Remove the following properties from the Kafka Broker Advanced Configuration Snippet (Safety Valve) configuration property.
   - Inter.broker.protocol.version
   - log.message.format.version

Deploy the Client Configuration

1. On the Cloudera Manager Home page, click the Actions menu and select Deploy Client Configuration.
2. Click Deploy Client Configuration.

Restart the Cluster

1. On the Cloudera Manager Home page, click the Actions menu and select Restart.
2. Click Restart that appears in the next screen to confirm. If you have enabled high availability for HDFS, you can choose Rolling Restart instead to minimize cluster downtime. The Command Details window shows the progress of stopping services.

   When All services successfully started appears, the task is complete and you can close the Command Details window.

Roll Back Cloudera Navigator Encryption Components

If you are rolling back any encryption components (Key Trustee Server, Key Trustee KMS, HSM KMS, Key HSM, or Navigator Encrypt), first refer to:

- Backing up and Restoring Key Trustee Server and Clients
- HSM KMS High Availability Backup and Recovery
- Manually Backing Up Navigator Encrypt

Roll Back Key Trustee Server

Note: If rolling back multiple encryption product components, it is recommended that you begin with the Key Trustee Server.

To roll back Key Trustee Server, replace the currently used parcel (for example, the parcel for version 7.1.4) with the parcel for the version to which you wish to roll back (for example, version 5.14.0). See Parcels for detailed instructions on using parcels.

The Keytrustee Server 7.x upgrades the bundled Postgres engine from version 9.3 to 12.1. The upgrade happens automatically, however, downgrading to CDH 6 requires manual steps to roll back the database engine to version 9.3. Because the previously upgraded database is left unchanged, the database server will fail start. Follow these steps to recreate the Postgres 9.3 compatible database:

1. Open the Cloudera Manager Admin Console and go to the Key Trustee Server service. If you see that Key Trustee Server has stale configurations, click the yellow or blue button and follow the prompts.
2. Make sure that the Keytrustee Server database roles are stopped. Then rename the folder containing Keytrustee Postgres database data (both on master and slave hosts):

   ```
   mv /var/lib/keytrustee/db /var/lib/keytrustee/db-12_1
   ```
3. Open the Cloudera Manager Admin Console and go to the Key Trustee Server service.
4. Select the Instances tab.
5. Select the Active Database role type.
6. Click Actions for SelectedSet Up the Key Trustee Server Database.
7. Click Set Up the Key Trustee Server Database to confirm.

   Cloudera Manager sets up the Key Trustee Server database.

8. On the master KTS node: running as user keytrustee, restore the keytrustee database from the dump created during the upgrade by running the following commands:

```
sudo -su keytrustee
export HOME=/opt/cloudera/parcels/KEYTRUSTEE_SERVER
export JAVA_HOME=... # Set this to your Java Home folder
export PATH="/opt/cloudera/parcels/KEYTRUSTEE_SERVER/bin:/opt/cloudera/
parcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/9.3/bin:$PATH"
source /opt/cloudera/parcels/KEYTRUSTEE_SERVER/meta/keytrustee_env.sh
dropdb -p 11381 keytrustee
```

   If you see the message: could not change directory to "/root: Permission denied on the console, run the following command to check the exit code of the last command:

```
$?
```

   You can use the exit code to debug any issues.
9. Run the following command to import a database dump that was created during upgrade:

```
psql -p 11381 postgres -f /var/lib/keytrustee/.keytrustee/kt93dump.pg
```

10. Start the Active Database role in Cloudera Manager by clicking Actions for SelectedStart.
11. Click Start to confirm.
12. Select the Active Database.
13. Click Actions for SelectedSetup Enable Synchronous Replication in HA mode .
14. Start the Passive Database instance: select the Passive Database, click Actions for SelectedStart.
15. In the Cloudera Manager Admin Console, start the active KTS instance.
16. In the Cloudera Manager Admin Console, start the passive KTS instance.

Start the Key Management Server

Restart the Key Management Server. Open the Cloudera Manager Admin Console, go to the KMS service page, and select ActionsStart.

Roll Back Key HSM

To roll back Key HSM:

1. Install the version of Navigator Key HSM to which you wish to roll back

   Install the Navigator Key HSM package using yum:

```
sudo yum downgrade keytrustee-keyhsm
```

   Cloudera Navigator Key HSM is installed to the /usr/share/keytrustee-server-keyhsm directory by default.
2. Rename Previously-Created Configuration Files

   For Key HSM major version rollbacks, previously-created configuration files do not authenticate with the HSM and Key Trustee Server, so you must recreate these files by re-executing the setup and trust commands. First, navigate to the Key HSM installation directory and rename the applications.properties, keystore, and truststore files:

```
cd /usr/share/keytrustee-server-keyhsm/
mv application.properties application.properties.bak
mv keystore keystore.bak
mv truststore truststore.bak
```

3. Initialize Key HSM

   Run the service keyhsm setup command in conjunction with the name of the target HSM distribution:

   ```
   sudo service keyhsm setup [keysecure|thales|luna]
   ```

   For more details, see Initializing Navigator Key HSM.
4. Establish Trust Between Key HSM and the Key Trustee Server

   The Key HSM service must explicitly trust the Key Trustee Server certificate (presented during TLS handshake). To establish this trust, run the following command:

   ```
   sudo keyhsm trust /path/to/key_trustee_server/cert
   ```

   For more details, see Establish Trust from Key HSM to Key Trustee Server.
5. Start the Key HSM Service

   Start the Key HSM service:

   ```
   sudo service keyhsm start
   ```
6. Establish Trust Between Key Trustee Server and Key HSM

   Establish trust between the Key Trustee Server and the Key HSM by specifying the path to the private key and certificate:

   ```
   sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 \
   --client-certfile /etc/pki/cloudera/certs/mycert.crt \
   --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
   ```

   For a password-protected Key Trustee Server private key, add the --passphrase argument to the command (enter the password when prompted):

   ```
   sudo ktadmin keyhsm --passphrase \
   --server https://keyhsm01.example.com:9090 \
   --client-certfile /etc/pki/cloudera/certs/mycert.crt \
   --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
   ```

   For additional details, see Integrate Key HSM and Key Trustee Server.
7. Remove Configuration Files From Previous Installation

   After completing the rollback, remove the saved configuration files from the previous installation:

   ```
   cd /usr/share/keytrustee-server-keyhsm/
   rm application.properties.bak
   rm keystore.bak
   rm truststore.bak
   ```

Roll Back Key Ranger KMS Parcels

Enable the desired parcel that you wish to roll back to (for example, version 6.3.4 of Key Trustee KMS). See Parcels for detailed instructions on using parcels. See Parcels for detailed instructions on using parcels.

Roll Back HSM KMS Parcels

To roll back the HSM KMS parcels, replace the currently used parcel (for example, the parcel for version 6.0.0) with the parcel for the version to which you wish to roll back (for example, version 5.14.0). See Parcels for detailed instructions on using parcels.

See Upgrading HSM KMS Using Packages for detailed instructions on using packages.

Roll Back Navigator Encrypt

To roll back Cloudera Navigator Encrypt:

1. If you have configured and are using an RSA master key file with OAEP padding, then you must revert this setting to its original value:

```
navencrypt key --change
```

2. Stop the Navigator Encrypt mount service:

```
sudo /etc/init.d/navencrypt-mount stop
```

3. Confirm that the mount-stop command completed:

```
sudo /etc/init.d/navencrypt-mount status
```

4. If rolling back to a release lower than NavEncrypt 6.2:
   a. a. Print the existing ACL rules and save that output to a file:

```
sudo navencrypt acl --print+ vim acls.txt
```

   b. b. Delete all existing ACLs, for example, if there are a total of 7 ACL rules run:

```
sudo navencrypt acl --del --line=1,2,3,4,5,6,7
```

5. To fully downgrade Navigator Encrypt, manually downgrade all of the associated Navigator Encrypt packages (in the order listed):
   a. navencrypt
   b. (Only required for operating systems other than SLES) navencrypt-kernel-module
   c. (Only required for the SLES operating system) cloudera-navencryptfs-kmp-<*KERNEL_FLAVOR*>

   Note: Replace kernel_flavor with the kernel flavor for your system. Navigator Encrypt supports the default, xen, and ec2 kernel flavors.d. libkeytrustee
6. If rolling back to a release less than NavEncrypt 6.2
   a. Reapply the ACL rules:

```
sudo navencrypt acl --add --file=acls.txt
```

7. Recompute process signatures:

```
sudo navencrypt acl --update
```

8. Restart the Navigator Encrypt mount service

```
sudo /etc/init.d/navencrypt-mount start
```

(Optional) Cloudera Manager Rollback Steps

After you complete the rollback steps, your cluster is using Cloudera Manager 7 to manage your CDH 6 or CDH 6 cluster. You can continue to use Cloudera Manager 7 to manage your CDH 6 cluster, or you can downgrade to Cloudera Manager 6 by following these steps:

Stop Cloudera Manager

1. Stop the .
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Stop .
2. Stop the Server.

```
sudo systemctl stop cloudera-scm-server
```

3. Hard stop the Cloudera Manager agents. Run the following command on all hosts:

```
sudo systemctl stop cloudera-scm-supervisord.service
```

4. Back up the repository directory. You can create a top-level backup directory and an environment variable to reference the directory using the following commands. You can also substitute another directory path in the backup commands below:

```
export CM_BACKUP_DIR="`date +%F`-CM"
mkdir -p $CM_BACKUP_DIR
```

5. Back up the existing repository directory.
   RHEL / CentOS

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
```

   SLES

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
```

   Ubuntu

```
sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources
.list.d
```

Restore the Cloudera Manager 6 Repository Files

Copy the repository directory from the backup taken before upgrading to Cloudera Manager 7.x.

```
rm -rf /etc/yum.repos.d/*
tar -xf cm6cdh6_backedUp_dir/repositary.tar -C CM6CDH6/
cp -rp /etc/yum.repos.d_CM6CDH6/* /etc/yum.repos.d/
```

Restore Packages

1. Run the following commands on all hosts:

Operating System	Command

RHEL	`sudo yum remove cloudera-manager-daemons cloudera-manager-agent`
	`sudo yum clean all`
	`sudo yum install cloudera-manager-agent`

SLES	`sudo zypper remove cloudera-manager-daemons cloudera-manager-agent`
	`sudo zypper refresh -s`
	`sudo zypper install cloudera-manager-agent`

Ubuntu or Debian	`sudo apt-get purge cloudera-manager-daemons cloudera-manager-agent`
	`sudo apt-get update`
	`sudo apt-get install cloudera-manager-agent`

2. Run the following commands on the Cloudera Manager server host:

Operating System	Command
RHEL	`sudo yum remove cloudera-manager-server` `sudo yum install cloudera-manager-server`
SLES	`sudo zypper remove cloudera-manager-server` `sudo zypper install cloudera-manager-server`
Ubuntu or Debian	`sudo apt-get purge cloudera-manager-server` `sudo apt-get install cloudera-manager-server`

Restore Cloudera Manager Databases

Restore the Cloudera Manager databases from the backup of Cloudera Manager that was taken before upgrading to Cloudera Manager 7. See the procedures provided by your database vendor.

These databases include the following:

- Cloudera Manager Server
- Reports Manager
- Navigator Audit Server
- Navigator Metadata Server
- Activity Monitor (Only used for MapReduce 1 monitoring).

- MariaDB 5.5: http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/
- MySQL 5.5: http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html
- MySQL 5.6: http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html
- PostgreSQL 8.4: https://www.postgresql.org/docs/8.4/static/backup.html
- PostgreSQL 9.2: https://www.postgresql.org/docs/9.2/static/backup.html
- PostgreSQL 9.3: https://www.postgresql.org/docs/9.3/static/backup.html
- Oracle 11gR2: https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm
- HyperSQL: http://hsqldb.org/doc/guide/management-chapt.html#mtc_backup

Here is an sample command to restore a MySQL database:

```
mysql -u username -ppassword --host=hostname cm < backup.sql
```

Restore Cloudera Manager Server

Use the backup of Cloudera Manager 6.x taken before upgrading to Cloudera Manager 7.x for the following steps:

1. If you used the backup commands provided in Step 2: Backing Up Cloudera Manager 6 on page 71, extract the Cloudera Manager 6 backup archives you created:

   ```
   tar -xf CM6CDH6/cloudera-scm-agent.tar -C CM6CDH6/
   tar -xf CM6CDH6/cloudera-scm-server.tar -C CM6CDH6/
   ```

2. On the host where the Event Server role is configured to run, restore the Events Server directory from the Cloudera Manager 6 backup.

   ```
   cp -rp /var/lib/cloudera-scm-eventserver /var/lib/cloudera-scm-eventserv
   er-CM
   rm -rf /var/lib/cloudera-scm-eventserver/*
   cp -rp /var/lib/cloudera-scm-eventserver_CM6CDH6/* /var/lib/cloudera-scm-
   eventserver/
   ```

3. Remove the Agent runtime state. Run the following command on all hosts:

   ```
   rm -rf /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent/response.
   avro
   ```

4. On the host where the Service Monitor is running, restore the Service Monitor directory:

   ```
   rm -rf /var/lib/cloudera-service-monitor/*
   cp -rp /var/lib/cloudera-service-monitor_CM6CDH6/* /var/lib/cloudera-ser
   vice-monitor/
   ```

5. On the host where the Host Monitor is running, restore the Host Monitor directory:

   ```
   rm -rf /var/lib/cloudera-host-monitor/*
   cp -rp /var/lib/cloudera-host-monitor_CM6CDH6/* /var/lib/cloudera-host-mo
   nitor/
   ```

6. Restore the Cloudera Navigator Solr storage directory from the CM6/CDH6 backup.

   ```
   rm -rf /var/lib/cloudera-scm-navigator/*
   cp -rp /var/lib/cloudera-scm-navigator_CM6CDH6/* /var/lib/cloudera-scm-nav
   igator/
   ```

7. On the Cloudera Manager Server, restore the /etc/cloudera-scm-server/db.properties file.

   ```
   rm -rf /etc/cloudera-scm-server/db.properties
   cp -rp CM6CDH6/etc/cloudera-scm-server/db.properties /etc/cloudera-scm-ser
   ver/db.properties
   ```

8. On each host in the cluster, restore the /etc/cloudera-scm-agent/config.ini file from your backup.

```
rm -rf /etc/cloudera-scm-agent/config.ini
cp -rp CM6CDH6/etc/cloudera-scm-agent/config.ini /etc/cloudera-scm-agent/c
onfig.ini
```

Start the Cloudera Manager Server and Agents

- Start the Server.

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

- Hard Restart the Cloudera Manager Agent.
  RHEL 7, SLES 12, Ubuntu 18.04 and higher

```
sudo systemctl stop cloudera-scm-supervisord.service
sudo systemctl restart cloudera-scm-agent
```

  xg
- Start the .

  1. Log in to the Admin Console.
  2. Select Clusters .
  3. Select Actions Start .

Rolling Back a Cloudera Private Cloud Base Upgrade from version 7.1.8 to CDH 6

You can roll back an upgrade from Cloudera Private Cloud Base 7 to CDH 6. The rollback restores your CDH cluster to the state it was in before the upgrade, including Kerberos and TLS/SSL configurations.

Important: Any data created after the upgrade is lost.

In a typical upgrade, you first upgrade Cloudera Manager from version 6.x to version 7.x, and then you use the upgraded version of Cloudera Manager 7 to upgrade CDH 6 to Cloudera Private Cloud Base 7. (See Upgrading a CDH 6 Cluster on page 121.) If you want to roll back this upgrade, follow these steps to roll back your cluster to its state prior to the upgrade.

You can roll back to CDH 6 after upgrading to Cloudera Private Cloud Base 7 only if the HDFS upgrade has not been finalized. The rollback restores your CDH cluster to the state it was in before the upgrade, including Kerberos and TLS/SSL configurations.

Important: Follow all of the steps in the order presented in this topic. Cloudera recommends that you read through the backup and rollback steps before starting the backup process. You may want to create a detailed plan to help you anticipate potential problems.

Important:

These rollback steps depend on complete backups taken before upgrading Cloudera Manager and CDH. See Step 2: Backing Up Cloudera Manager 6 on page 71 and Step 3: Backing Up the Cluster on page 134.

For steps where you need to restore the contents of a directory, clear the contents of the directory before copying the backed-up files to the directory. If you fail to do this, artifacts from the original upgrade can cause problems if you attempt the upgrade again after the rollback.

Review Limitations

The rollback procedure has the following limitations:

- HDFS – If you have finalized the HDFS upgrade, you cannot roll back your cluster.
- Compute clusters – Rollback for Compute clusters is not supported. You must remove any compute clusters before rolling back.
- Configuration changes, including the addition of new services or roles after the upgrade, are not retained after rolling back Cloudera Manager.

  Cloudera recommends that you not make configuration changes or add new services and roles until you have finalized the HDFS upgrade and no longer require the option to roll back your upgrade.
- HBase – If your cluster is configured to use HBase replication, data written to HBase after the upgrade might not be replicated to peers when you start your rollback. This topic does not describe how to determine which, if any, peers have the replicated data and how to roll back that data. For more information about HBase replication, see HBase Replication.
- Sqoop 2 – As described in the upgrade process, Sqoop2 had to be stopped and deleted before the upgrade process and therefore will not be available after the rollback.
- Kafka – Once the Kafka log format and protocol version configurations (the inter.broker.protocol.version and log.message.format.version properties) are set to the new version (or left blank, which means to use the latest version), Kafka rollback is not possible.

Stop the Cluster

1. If HBase is deployed in the cluster do the following before stopping the cluster:

   The HBase Master procedures changed between the two versions, so if a procedure was started by HBase 2.2 (CDP 7.x) then the older HBase 2.1 won't be able to continue the procedure after the rollback. For this reason the Procedure Store in HBase Master must be cleaned before the rollback. If CDP 7.x HBase Master was never started, then the rollback should be fine. But if HBase Master was running with the new version and there is any ongoing (or stuck) HBase Master Procedure present in the CDP 7 HBase Master, then the older CDH 6 HBase Master will fail to start after the rollback. If this happens, HBase will need manual fix after the rollback (e.g. the sidelining of the HBase Master Procedure WAL files and the potential fixing of inconsistencies in HBase).

   To avoid this problem, you should try to verify that no unfinished procedure is present before stopping HBase Master on the CDP 7.x Cluster. Please follow these steps:

   a. Make sure there was no traffic running against the HBase Cluster recently (in the last 10 minutes) that can trigger e.g. table creation or deletion, region assignment or split or merge, etc.
   b. Disable automatic Balancer and Normalizer in HBase. Also disable Split and Merge procedures, before stopping the CDP 7 Cluster. All these tools in HBase can cause the starting of new HBase Master Procedures, which we want to avoid now. Issue the following commands in HBase Shell:

      ```
      balance_switch false
      normalizer_switch false
      splitormerge_switch 'SPLIT', false
      splitormerge_switch 'MERGE', false
      ```

   c. Check the list of procedures on the HBase Master Web UI (In Cloudera Manager, go to the HBase service and open the HBase Web UIProcedures & Locks tab). Wait until you see procedures only with final states like 'SUCCESS', 'FAILED' or 'ROLLEDBACK'.

d. Get the list of procedures from HBase shell using the 'list_procedures' command. Wait until you see procedures only with final states like 'SUCCESS', 'FAILED' or 'ROLLEDBACK'. The State appears in the third column of the table returned by the 'list_procedures' command.

If the HBase Master doesn't start after the rollback and some procedure-related exceptions are found in the role logs (like "BadProcedureException" or decode errors in the "ProcedureWALFormatReader" class, or "ClassNotFoundException" for procedure classes), then this is most likely caused by CDP 7 procedures that still remain in the procedure WAL files. In this case, please open a ticket for Cloudera customer support, who will help you to sideline the procedure WAL files and fix any potential inconsistencies in HBase.

2. On the HomeStatus tab, click the Actions menu and select Stop.
3. 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.
4. Go to the YARN service and click ActionsClean NodeManager Recovery Directory. The CDH 6 NodeManager will not start up after the downgrade if it finds CDP 7.x data in the recovery directory. The format and content of the NodeManager's recovery state store was changed between CDH 6.x and CDP 7.x. The recovery directory used by CDP 7.x must be cleaned up as part of the downgrade to CDH 6.

(Parcels) Downgrade the Software

Follow these steps only if your cluster was upgraded using Cloudera parcels.

1. Log in to the Cloudera Manager Admin Console.
2. Select HostsParcels.

   A list of parcels displays.
3. Locate the CDH 6 parcel and click Activate. (This automatically deactivates the Cloudera Private Cloud Base 7 parcel.) See Activating a Parcel for more information. If the parcel is not available, use the Download button to download the parcel.

   Important: If you added new services that are not part of CDH 6 during the upgrade, a list of services that need to be deleted displays. You must delete these services before activating the CDH 6 parcel.
4. If you include any additional components in your cluster, such as Search or Impala, click Activate for those parcels.
5. If the Ranger service is deployed in the cluster, disable the Ranger plugin from the services below, if they are deployed in the cluster:
   - HDFS: Go to the HDFS service > Configurations and disable the Enable Ranger Authorization configuration property.
   - Hive: Go to the Hive service > Configurations and delete the Ranger Service configuration property.
   - Kafka: Go to the Kafka service > Configurations and delete the Ranger Service configuration property.
   - Impala Go to the Impala service > Configurations > delete the Ranger Service configuration property.
6. After performing the above steps to disable the plugin, Stop the Ranger service and delete it.
7. The Sentry service will be added when you perform the Restore Cloudera Manager Databases steps, later in this rollback procedure. The Sentry service will be added in Cloudera Manager and will continue to use the database configuration saved in Cloudera Manager.

Important:

Do not start any services. (Select the Activate Only option.)

If you accidentally restart services, stop your cluster before proceeding.

Stop Cloudera Manager

1. Stop the .
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Stop .
2. Stop the Server.

```
sudo systemctl stop cloudera-scm-server
```

3. Hard stop the Cloudera Manager agents. Run the following command on all hosts:

```
sudo systemctl stop cloudera-scm-supervisord.service
```

Restore Cloudera Manager Databases

Restore the Cloudera Manager databases from the backup of Cloudera Manager that was taken before upgrading the cluster toCloudera Private Cloud Base 7. See the procedures provided by your database vendor.

- MariaDB 5.5: http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/
- MySQL 5.5: http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html
- MySQL 5.6: http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html
- PostgreSQL 8.4: https://www.postgresql.org/docs/8.4/static/backup.html
- PostgreSQL 9.2: https://www.postgresql.org/docs/9.2/static/backup.html
- PostgreSQL 9.3: https://www.postgresql.org/docs/9.3/static/backup.html
- Oracle 11gR2: https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm
- HyperSQL: http://hsqldb.org/doc/guide/management-chapt.html#mtc_backup

Restore Cloudera Manager Server

Use the backup of CDH that was taken before the upgrade to restore Cloudera Manager Server files and directories. Substitute the path to your backup directory for *CM7_CDH6* in the following steps:

1. On the host where the Event Server role is configured to run, restore the Events Server directory from the CM 7/ CDH 6 backup.

```
cp -rp /var/lib/cloudera-scm-eventserver /var/lib/cloudera-scm-eventserv
er-CM<ph outputclass="cdoc-specific-cm-from"/>-CDH<ph outputclass="cdoc-
specific-cdh-from"/>
rm -rf /var/lib/cloudera-scm-eventserver/*
cp -rp /var/lib/cloudera-scm-eventserver_CM7_CDH6/* /var/lib/cloudera-scm-
eventserver/
```

2. Remove the Agent runtime state. Run the following command on all hosts:

```
rm -rf /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent/response.
avro
```

   This command may return a message similar to: rm: cannot remove ‘/var/run/cloudera-scm-agent/process’: D evice or resource busy. You can ignore this message.
3. On the host where the Service Monitor is running, restore the Service Monitor directory:

```
rm -rf /var/lib/cloudera-service-monitor/*
cp -rp /var/lib/cloudera-service-monitor_CM7_CDH6/* /var/lib/cloudera-ser
vice-monitor/
```

4. On the host where the Host Monitor is running, restore the Host Monitor directory:

```
rm -rf /var/lib/cloudera-host-monitor/*
cp -rp /var/lib/cloudera-host-monitor_CM7_CDH6/* /var/lib/cloudera-host-mo
nitor/
```

5. Restore the Cloudera Navigator storage directory from the CM 7/CDH 6 backup.

```
rm -rf /var/lib/cloudera-scm-navigator/*
cp -rp /var/lib/cloudera-scm-navigator_CM7_CDH6/* /var/lib/cloudera-scm-
navigator/
```

Start Cloudera Manager

1. Log in to the Cloudera Manager server host.
2. Start the Server.

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

3. Start the Cloudera Manager Agent.

   Run the following commands on all cluster hosts:

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

4. Start the Cloudera Management Service.
   a. Log in to the Cloudera Manager Admin Console.
   b. Select ClustersCloudera Management Service.
   c. Select ActionsStart.

   The cluster page may indicate that services are in bad health. This is normal.
5. Stop the cluster. In the Cloudera Manager Admin Console, click the Actions menu for the cluster and select Stop.

Roll Back ZooKeeper

1. Using the backup of Zookeeper that you created when backing up your CDH 6.x cluster, restore the contents of the *DATADIR* on each ZooKeeper server. These files are located in a directory specified with the dataDir property in the ZooKeeper configuration. The default location is /var/lib/zookeeper. For example:

```
rm -rf /var/lib/zookeeper/*
cp -rp /var/lib/zookeeper_CM7_CDH6/* /var/lib/zookeeper/
```

2. Using the backup of Zookeeper that you created when backing up your CDH 6.x cluster, restore the contents of the *TRANSACTION LOG DIRECGTORY* on each ZooKeeper server. These files are located in a directory specified with the Transaction Log Direcgtory property in the ZooKeeper configuration. For example:

```
rm -rf /var/lib/zookeeper/*
cp -rp /var/lib/zookeeper_CM7_CDH6/* /var/lib/zookeeper/
```

3. Make sure that the permissions of all the directories and files are as they were before the upgrade.
4. Start ZooKeeper using Cloudera Manager.

Roll Back HDFS

You cannot roll back HDFS while high availability is enabled. The rollback procedure in this topic creates a temporary configuration without high availability. Regardless of whether high availability is enabled, follow the steps in this section.

1. Roll back all of the Journal Nodes. (Only required for clusters where high availability is enabled for HDFS). Use the JournalNode backup you created when you backed up HDFS before upgrading to Cloudera Private Cloud Base.
   a. Log in to each Journal Node host and run the following commands:

      ```
      rm -rf /dfs/jn/ns1/current/*
      ```

      ```
      cp -rp <JOURNAL_NODE_BACKUP_DIRECTORY>/ns1/current/* /dfs/jn/ns1/curren
      t/
      ```

   b. Start the JournalNodes using Cloudera Manager:
      1. Go to the HDFS service.
      2. Select the Instances tab.
      3. Select all JournalNode roles from the list.
      4. Click Actions for Selected Start .
2. Roll back all of the NameNodes. Use the NameNode backup directory you created before upgrading to Cloudera Private Cloud Base. (/etc/hadoop/conf.rollback.namenode) to perform the following steps on all NameNode hosts:
   a. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file on all NameNode hosts (located in the temporary rollback directory) and update the keystore passwords with the actual cleartext passwords.

      The passwords will have values that look like this:

      ```
      <property>
          <name>ssl.server.keystore.password</name>
          <value>********</value>
        </property>
        <property>
          <name>ssl.server.keystore.keypassword</name>
          <value>********</value>
        </property>
      ```

   b. (TLS only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file and remove the hadoop.security.credential.provider.path property.
   c. (TLS only) Edit the /etc/hadoop/conf.rollback.namenode/ssl-server.xml file and update the ssl.server.keystore.location property:

      ```
      # Original version of the keystore.location property:
      <property>
        <name>ssl.server.keystore.location</name>
        <value>/var/run/cloudera-scm-agent/process/879-hdfs-NAMENODE/cm-auto-
      host_keystore.jks</value>
      </property>
      ```

      ```
      # New version of the keystore.location property:
      <property>
        <name>ssl.server.keystore.location</name>
        <value>/etc/hadoop/conf.rollback.namenode/cm-auto-host_keystore.jks</v
      alue>
      </property>
      ```

3. Edit the /etc/hadoop/conf.rollback.namenode/hdfs-site.xml file on all NameNode hosts and make the following changes:
   a. Update the dfs.namenode.inode.attributes.provider.class property. If Sentry was installed prior to the upgrade, change the value of the property from org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer to "org.apache.sentry.hdfs.SentryINodeAttributesProvider. If Sentry was not installed, remove this property.
   b. Change the path in the dfs.hosts property to the value shown in the example below. The file name, dfs_all_hosts.txt, may have been changed by a user. If so, substitute the correct file name.

```
# Original version of the dfs.hosts property:
<property>
<name>dfs.hosts</name>
<value>/var/run/cloudera-scm-agent/process/63-hdfs-NAMENODE/dfs_all_hosts.txt</value>
</property>
```

```
# New version of the dfs.hosts property:
<property>
<name>dfs.hosts</name>
<value>/etc/hadoop/conf.rollback.namenode/dfs_all_hosts.txt</value>
</property>
```

   c. Remove the property that has the following value:

```
com.cloudera.navigator.audit.hdfs.HdfsAuditLoggerCdh5
```

4. Edit the /etc/hadoop/conf.rollback.namenode/core-site.xml and change the value of the net.topology.script.file.name property to /etc/hadoop/conf.rollback.namenode. For example:

```
# Original property
<property>
<name>net.topology.script.file.name</name>
<value>/var/run/cloudera-scm-agent/process/63-hdfs-NAMENODE/topology.py</value>
</property>
```

```
# New property
<property>
<name>net.topology.script.file.name</name>
<value>/etc/hadoop/conf.rollback.namenode/topology.py</value>
</property>
```

5. Edit the /etc/hadoop/conf.rollback.namenode/topology.py file and change the value of MAP_FILE to /etc/hadoop/conf.rollback.namenode. For example:

```
MAP_FILE = '/etc/hadoop/conf.rollback.namenode/topology.map'
```

6. (TLS-enabled clusters only) Run the following command:

```
sudo -u hdfs kinit hdfs/<NAMENODE HOST NAME> -l 7d -kt /etc/hadoop/conf.rollback.namenode/hdfs.keytab
```

7. Run the following command:

```
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.namenode namenode -rollback
```

8. Restart the NameNodes and JournalNodes using Cloudera Manager:
   a. Go to the HDFS service.
   b. Select the Instances tab, and then select all Failover Controller, NameNode, and JournalNode roles from the list.
   c. Click Actions for SelectedRestart.
9. Rollback the DataNodes.

   Use the DataNode rollback directory you created before upgrading to Cloudera Private Cloud Base (/etc/hadoop/conf.rollback.datanode) to perform the following steps on all DataNode hosts:

   a. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.datanode/ssl-server.xml file on all DataNode hosts (Located in the temporary rollback directory.) and update the keystore passwords (ssl.server.keystore.password and ssl.server.keystore.keypassword) with the actual passwords.

      The passwords will have values that look like this:

      ```
      <property>
          <name>ssl.server.keystore.password</name>
          <value>********</value>
        </property>
        <property>
          <name>ssl.server.keystore.keypassword</name>
          <value>********</value>
        </property>
      ```

   b. (TLS only) Edit the /etc/hadoop/conf.rollback.datanode/ssl-server.xml file and update the ssl.server.keystore.location property:

      ```
      # Original version of the keystore.location property:
      <property>
        <name>ssl.server.keystore.location</name>
        <value>/var/run/cloudera-scm-agent/process/879-hdfs-NAMENODE/cm-auto-host_keystore.jks</value>
      </property>
      ```

      ```
      # New version of the keystore.location property:
      <property>
        <name>ssl.server.keystore.location</name>
        <value>/etc/hadoop/conf.rollback.namenode/cm-auto-host_keystore.jks</value>
      </property>
      ```

   c. (TLS only) Edit the /etc/hadoop/conf.rollback.datanode/ssl-server.xml file and remove the hadoop.security.credential.provider.path property.
   d. Edit the /etc/hadoop/conf.rollback.datanode/hdfs-site.xml file and remove the dfs.datanode.max.locked.memory property.
   e. Run one of the following commands:

      - Run as root if the DataNodes use reserved ports.

        Search in the logs for completed rollback line. It will not be shown in the command line of the rollback.
      - If the DataNode is running with privileged ports (usually 1004 and 1006):

        ```
        cd /etc/hadoop/conf.rollback.datanode
        export HADOOP_SECURE_DN_USER=hdfs
        export JSVC_HOME=/opt/cloudera/parcels/<PARCEL_FILENAME>/lib/bigtop-utils
        hdfs --config /etc/hadoop/conf.rollback.datanode datanode -rollback
        ```

- If the DataNode is not running on privileged ports:

```
cd /etc/hadoop/conf.rollback.datanode
sudo hdfs --config /etc/hadoop/conf.rollback.datanode datanode -ro
llback
```

You may see the following error after issuing these commands:

```
ERROR datanode.DataNode: Exception in secureMain
java.io.IOException: The path component: '/var/run/hdfs-sockets' in
 '/var/run/hdfs-sockets/dn' has permissions 0755 uid 39998 and gid 1
006.
It is not protected because it is owned by a user who is not root
and not the effective user: '0'.
```

The error message will also include the following command to run:

```
chown root /var/run/hdfs-sockets
```

After running this command, rerun the DataNode rollback command:

```
sudo hdfs --config /etc/hadoop/conf.rollback.datanode datanode -roll
back
```

The DataNodes will now restart successfully.

When the rolling back of the DataNodes is complete, terminate the console session by typing Control-C. Look for output from the command similar to the following that indicates when the DataNode rollback is complete:

```
Rollback of /dataroot/ycloud/dfs/dn/current/BP-<Block Group number> is c
omplete
```

f. If High Availability for HDFS is enabled, restart the HDFS service. In the Cloudera Manager Admin Console, go to the HDFS service and select Actions Restart .

g. If high availability is not enabled for HDFS, use the Cloudera Manager Admin Console to restart all NameNodes and DataNodes.

1. Go to the HDFS service.
2. Select the Instances tab
3. Select all DataNode and NameNode roles from the list.
4. Click Actions for Selected Restart .

10. If high availability is not enabled for HDFS, roll back the Secondary NameNode.

a. (Clusters with TLS enabled only) Edit the /etc/hadoop/conf.rollback.secondarynamenode/ssl-server.xml file on all Secondary NameNode hosts (Located in the temporary rollback directory.) and update the keystore passwords with the actual cleartext passwords.

The passwords will have values that look like this:

```
<property>
    <name>ssl.server.keystore.password</name>
    <value>********</value>
  </property>
  <property>
    <name>ssl.server.keystore.keypassword</name>
    <value>********</value>
  </property>
```

b. (TLS only) Edit the /etc/hadoop/conf.rollback.secondarynamenode/ssl-server.xml file and remove the hadoop.security.credential.provider.path property.

c. Log in to the Secondary NameNode host and run the following commands:

```
rm -rf /dfs/snn/*
cd /etc/hadoop/conf.rollback.secondarynamenode/
sudo -u hdfs hdfs --config /etc/hadoop/conf.rollback.secondarynamenode
 secondarynamenode -format
```

When the rolling back of the Secondary NameNode is complete, terminate the console session by typing Control-C. Look for output from the command similar to the following that indicates when the Secondary NameNode rollback is complete:

```
2020-12-21 17:09:36,239 INFO namenode.SecondaryNameNode: Web server init
 done
```

11. Restart the HDFS service. Open the Cloudera Manager Admin Console, go to the HDFS service page, and select ActionsRestart.

The Restart Command page displays the progress of the restart. Wait for the page to display the Successfully restarted service message before continuing.

Start the HBase Service

Restart the HBase Service. Open the Cloudera Manager Admin Console, go to the HBase service page, and select ActionsStart.

If you have configured any HBase coprocessors, you must revert them to the versions used before the upgrade.

If CDP 7.x HBase Master was started after the upgrade and there was any ongoing (or stuck) HBase Master Procedure present in the HBase Master before stopping the CDP 7 Cluster, then it is expected for the CDH 6 HBase Master to fail with warnings and errors in the role log from the classes like 'ProcedureWALFormatReader' and 'WALProcedureStore' or 'TransitRegionStateProcedure'. These errors mean that the HBase Master Write-Ahead Log files are incompatible with the CDH 6 HBase version. The only way to fix this problem is to sideline the log files (all the files placed under /hbase/MasterProcWALs by default), then restart the HBase Master. After the HBase Master has started, Use the HBCK command to find out if there are any inconsistencies that will need to be fixed manually.

You my encounter other errors when starting HBase (for example, replication-related problems, region assignment related issues, and meta region assignment problems). In this case you should delete the znode in ZooKeeper and then start HBase again. (This will delete replication peer information and you will need to re-configure your replication schedules.):

1. In Cloudera Manager, look up the value of the zookeeper.znode.parent property. The default value is /hbase.
2. Connect to the ZooKeeper ensemble by running the following command from any HBase gateway host:

```
zookeeper-client -server ZOOKEEPER_ENSEMBLE
```

To find the value to use for *ZOOKEEPER_ENSEMBLE*, open the /etc/hbase/conf.cloudera.<*HBASE SERVICE NAME*>/hbase-site.xml file on any HBase gateway host. Use the value of the hbase.zookeeper.quorum property.

Note:

If you have deployed a secure cluster, you must connect to ZooKeeper using a client jaas.conf file. You can find such a file in an HBase process directory (/var/run/cloudera-scm-agent/process/). Specify the jaas .conf using the JVM flags by running the following commands in the ZooKeeper client:

```
CLIENT_JVMFLAGS=
 "-Djava.security.auth.login.config=/var/run/cloudera-scm-agent/pro
cess/HBASE_PROCESS_DIRECTORY/jaas.conf"
zookeeper-client -server <zookeeper_ensemble>
```

The ZooKeeper command-line interface opens.

3. Enter the following command:

```
rmr /hbase
```

4. After HBase is healthy, make sure you restore the states of the Balancer and Normalizer (enable them if they were enabled before the rollback). Also re-enable the Merge and Split operations you disabled before the rollback to avoid the Master Procedure incompatibility problem. Run the following commands in HBase Shell:

```
balance_switch true
normalizer_switch true
splitormerge_switch 'SPLIT', true
splitormerge_switch 'MERGE', true
```

Fixing tableinfo file format

When you are rolling back from CDP Private Cloud Base 7.1.8 to CDH 6 if you encounter a change in the tableinfo file name format from the new tableinfo file name that was created during the 7.1.8 upgrade can prevent HBase from functioning normally.

After the rollback, if HDFS rollback was not successful and Hbase is unable to read the tableinfo files then use the HBCK2 tool to verify the list of tableinfo files that need to be fixed.

Follow these steps to execute the HBCK2 command on the HBCK2 tool to fix the tableinfo file format:

1. Contact Cloudera support to request the latest version of HBCK2 tool.
2. Use the following HBCK2 command and run the HBCK2 tool without the –fix option:

```
hbase --config /path/to/client/conf hbck -j
~/path/to/hbck/hbase-hbck2-1.0.0-<build>.jar shortenTableinfo
```

   For example:

```
hbase --config /etc/hbase/conf hbck -j
~/hbase-operator-tools/hbase-hbck2/target/hbase-hbck2-1.0.0-SNAPSHOT.jar
shortenTableinfo
```

   The command displays the following message and the list of files to be fixed:

   Found the following tableinfo file names containing file size

   If the list is empty, no additional steps are needed. Go to Step 11.
3. Use the following HBCK2 command and run the HBCK2 tool with the –fix option:

```
hbase --config /etc/hbase/conf hbck -j
~/hbase-operator-tools/hbase-hbck2/target/hbase-hbck2-1.0.0-SNAPSHOT.jar
shortenTableinfo –fix
```

4. Check the output and verify whether all the tableinfo files are fixed.

Restore CDH Databases

Restore the following databases from the CDH 6 backups:

- Hive Metastore
- Hue
- Oozie
- Sentry Server

The steps for backing up and restoring databases differ depending on the database vendor and version you select for your cluster and are beyond the scope of this document.

Important: Restore the databases to their exact state as of when you took the backup. Do not merge in any changes that may have occurred during the subsequent upgrade.

See the following vendor resources for more information:

- MariaDB 5.5: http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/
- MySQL 5.5: http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html
- MySQL 5.6: http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html
- MySQL 5.7: http://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html
- PostgreSQL 8.4: https://www.postgresql.org/docs/8.4/static/backup.html
- PostgreSQL 9.2: https://www.postgresql.org/docs/9.2/static/backup.html
- PostgreSQL 9.3: https://www.postgresql.org/docs/9.3/static/backup.html
- Oracle 11gR2: https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm

Start the Sentry Service

Roll Back Cloudera Search

1. Start the HDFS, Zookeeper and Sentry services.
2. Delete the instancedir created during the upgrade process:
   a. If the cluster is secured with Kerberos, run this command Otherwise skip this step.

```
export ZKCLI_JVM_FLAGS="-Djava.security.auth.login.config=~/solr-jaas.co
nf
-DzkACLProvider=org.apache.solr.common.cloud.ConfigAwareSaslZkACLProvide
r"
```

   b.

```
sudo -u solr solrctl instancedir --delete localFSTemplate
```

   3. Start the Solr service.

Note: If the state of one or more Solr core is down and the Solr log contains an org.apache.lucene.store. LockObtainFailedException: Lock obtain timed out: org.apache.solr.store.hdfs.HdfsLockFactory error message, it is necessary to clean up the HDFS locks in the index directories.

For each of the affected Solr nodes:

   a. 1. Stop the Solr node using Cloudera Manager.
   b. 2. Remove the HdfsDirectory<id>-write.lock file from the index directory.

```
hdfs dfs -rm "/solr/<collection_name>/<core>/data/<index_directory_name>
/HdfsDirectory@ <hex_id> lockFactory=org.apache.solr.store.hdfs.HdfsLock
Factory@<hex_id>-write.lock"
```

For example:

```
hdfs dfs -rm "/solr/testCollection/core_node1/data/index/HdfsDirectory@5
d07feac lockFactory=org.apache.solr.store.hdfs.HdfsLockFactory@7df08aad-
write.lock"
```

   c. 3. Start the Solr node using Cloudera Manager.

Roll Back Hue

1. Restore the file, app.reg, from your backup:
   - Parcel installations

     ```
     rm -rf /opt/cloudera/parcels/CDH/lib/hue/app.reg
     cp -rp app.reg_CM7_CDH6_BACKUP /opt/cloudera/parcels/CDH/lib/hue/app.reg
     ```

   - Package Installations

     ```
     rm -rf /usr/lib/hue/app.reg
     cp -rp app.reg_CM7_CDH6_BACKUP /usr/lib/hue/app.reg
     ```

Roll Back Kafka

A Cloudera Private Cloud Base 7 cluster that is running Kafka can be rolled back to the previous CDH5/CDK versions as long as theinter.broker.protocol.version and log.message.format.version properties have not been set to the new version or removed from the configuration.

To perform the rollback using Cloudera Manager:

1. Activate the previous CDK parcel. Please note, that when rolling back Kafka from CDP Private Cloud Base 7 to CDH 6/CDK, the Kafka cluster will restart. Rolling restart is not supported for this scenario. See Activating a Parcel.
2. Remove the following properties from the Kafka Broker Advanced Configuration Snippet (Safety Valve) configuration property.
   - Inter.broker.protocol.version
   - log.message.format.version

Deploy the Client Configuration

1. On the Cloudera Manager Home page, click the Actions menu and select Deploy Client Configuration.
2. Click Deploy Client Configuration.

Restart the Cluster

1. On the Cloudera Manager Home page, click the Actions menu and select Restart.
2. Click Restart that appears in the next screen to confirm. If you have enabled high availability for HDFS, you can choose Rolling Restart instead to minimize cluster downtime. The Command Details window shows the progress of stopping services.

   When All services successfully started appears, the task is complete and you can close the Command Details window.

Roll Back Cloudera Navigator Encryption Components

If you are rolling back any encryption components (Key Trustee Server, Key Trustee KMS, HSM KMS, Key HSM, or Navigator Encrypt), first refer to:

- Backing up and Restoring Key Trustee Server and Clients
- HSM KMS High Availability Backup and Recovery
- Manually Backing Up Navigator Encrypt

Roll Back Key Trustee Server

Note: If rolling back multiple encryption product components, it is recommended that you begin with the Key Trustee Server.

To roll back Key Trustee Server, replace the currently used parcel (for example, the parcel for version 7.1.4) with the parcel for the version to which you wish to roll back (for example, version 5.14.0). See Parcels for detailed instructions on using parcels.

The Keytrustee Server 7.x upgrades the bundled Postgres engine from version 9.3 to 12.1. The upgrade happens automatically, however, downgrading to CDH 6 requires manual steps to roll back the database engine to version 9.3. Because the previously upgraded database is left unchanged, the database server will fail start. Follow these steps to recreate the Postgres 9.3 compatible database:

1. Open the Cloudera Manager Admin Console and go to the Key Trustee Server service. If you see that Key Trustee Server has stale configurations, click the yellow or blue button and follow the prompts.
2. Make sure that the Keytrustee Server database roles are stopped. Then rename the folder containing Keytrustee Postgres database data (both on master and slave hosts):

   ```
   mv /var/lib/keytrustee/db /var/lib/keytrustee/db-12_1
   ```
3. Open the Cloudera Manager Admin Console and go to the Key Trustee Server service.
4. Select the Instances tab.
5. Select the Active Database role type.
6. Click Actions for SelectedSet Up the Key Trustee Server Database.
7. Click Set Up the Key Trustee Server Database to confirm.

   Cloudera Manager sets up the Key Trustee Server database.
8. Start the Active Database role in Cloudera Manager by clicking Actions for Selected Start .
9. Click Start to confirm.
10. On the master KTS node: running as user keytrustee, restore the keytrustee database from the dump created during the upgrade by running the following commands:

    ```
    sudo -su keytrustee
    cd
    export HOME=/opt/cloudera/parcels/KEYTRUSTEE_SERVER
    export JAVA_HOME=... # Set this to your Java Home folder
    export PATH="/opt/cloudera/parcels/KEYTRUSTEE_SERVER/bin:/opt/cloudera/p
    arcels/KEYTRUSTEE_SERVER/PG_DB/opt/postgres/9.3/bin:$PATH"
    source /opt/cloudera/parcels/KEYTRUSTEE_SERVER/meta/keytrustee_env.sh
    dropdb -p 11381 keytrustee
    ```

    If you see the message: could not change directory to "/root: Permission denied on the console, run the following command to check the exit code of the last command:

    ```
    $?
    ```

    You can use the exit code to debug any issues.
11. Run the following command to import a database dump that was created during upgrade:

    ```
    psql -p 11381 postgres -f /var/lib/keytrustee/.keytrustee/kt93dump.pg
    ```
12. Enable synchronous replication by navigating to KTS Actions Enable Synchronous Replication in HA Mode .
13. Start the Passive Database instance: select the Passive Database, click Actions for SelectedStart.
14. In the Cloudera Manager Admin Console, start the active KTS instance.
15. In the Cloudera Manager Admin Console, start the passive KTS instance.

Start the Key Management Server

Restart the Key Management Server. Open the Cloudera Manager Admin Console, go to the KMS service page, and select ActionsStart.

Roll Back Key HSM

To roll back Key HSM:

1. Install the version of Navigator Key HSM to which you wish to roll back

   Install the Navigator Key HSM package using yum:

   ```
   sudo yum downgrade keytrustee-keyhsm
   ```

   Cloudera Navigator Key HSM is installed to the /usr/share/keytrustee-server-keyhsm directory by default.
2. Rename Previously-Created Configuration Files

   For Key HSM major version rollbacks, previously-created configuration files do not authenticate with the HSM and Key Trustee Server, so you must recreate these files by re-executing the setup and trust commands. First, navigate to the Key HSM installation directory and rename the applications.properties, keystore, and truststore files:

   ```
   cd /usr/share/keytrustee-server-keyhsm/
   mv application.properties application.properties.bak
   mv keystore keystore.bak
   mv truststore truststore.bak
   ```
3. Initialize Key HSM

   Run the service keyhsm setup command in conjunction with the name of the target HSM distribution:

   ```
   sudo service keyhsm setup [keysecure|thales|luna]
   ```

   For more details, see Initializing Navigator Key HSM.
4. Establish Trust Between Key HSM and the Key Trustee Server

   The Key HSM service must explicitly trust the Key Trustee Server certificate (presented during TLS handshake). To establish this trust, run the following command:

   ```
   sudo keyhsm trust /path/to/key_trustee_server/cert
   ```

   For more details, see Establish Trust from Key HSM to Key Trustee Server.
5. Start the Key HSM Service

   Start the Key HSM service:

   ```
   sudo service keyhsm start
   ```
6. Establish Trust Between Key Trustee Server and Key HSM

   Establish trust between the Key Trustee Server and the Key HSM by specifying the path to the private key and certificate:

   ```
   sudo ktadmin keyhsm --server https://keyhsm01.example.com:9090 \
   --client-certfile /etc/pki/cloudera/certs/mycert.crt \
   --client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
   ```

   For a password-protected Key Trustee Server private key, add the --passphrase argument to the command (enter the password when prompted):

   ```
   sudo ktadmin keyhsm --passphrase \
   --server https://keyhsm01.example.com:9090 \
   --client-certfile /etc/pki/cloudera/certs/mycert.crt \
   ```

```
--client-keyfile /etc/pki/cloudera/certs/mykey.key --trust
```

For additional details, see Integrate Key HSM and Key Trustee Server.

7. Remove Configuration Files From Previous Installation

   After completing the rollback, remove the saved configuration files from the previous installation:

```
cd /usr/share/keytrustee-server-keyhsm/
rm application.properties.bak
rm keystore.bak
rm truststore.bak
```

Roll Back Key Ranger KMS Parcels

Enable the desired parcel that you wish to roll back to (for example, version 6.3.4 of Key Trustee KMS). See Parcels for detailed instructions on using parcels. See Parcels for detailed instructions on using parcels.

Roll Back HSM KMS Parcels

To roll back the HSM KMS parcels, replace the currently used parcel (for example, the parcel for version 6.0.0) with the parcel for the version to which you wish to roll back (for example, version 5.14.0). See Parcels for detailed instructions on using parcels.

See Upgrading HSM KMS Using Packages for detailed instructions on using packages.

Roll Back Navigator Encrypt

To roll back Cloudera Navigator Encrypt:

1. If you have configured and are using an RSA master key file with OAEP padding, then you must revert this setting to its original value:

```
navencrypt key --change
```

2. Stop the Navigator Encrypt mount service:

```
sudo /etc/init.d/navencrypt-mount stop
```

3. Confirm that the mount-stop command completed:

```
sudo /etc/init.d/navencrypt-mount status
```

4. Backup the NavEncrypt control directory:

```
sudo mkdir navencryptBAK
sudo cp -rp /etc/navencrypt/ navencryptBAK/
```

5. Clean the dkms/navencryptfs directory:

```
sudo rm -rf /var/lib/dkms/navencryptfs/
```

6. If rolling back to a release lower than NavEncrypt 6.2:

   a. Print the existing ACL rules and save that output to a file:

```
sudo navencrypt acl --print+ vim acls.txt
```

   b. Delete all existing ACLs, for example, if there are a total of 7 ACL rules run:

```
sudo navencrypt acl --del --line=1,2,3,4,5,6,7
```

7. To fully downgrade Navigator Encrypt, manually downgrade all of the associated Navigator Encrypt packages (in the order listed):
   a. navencrypt
   b. navencrypt-kernel-module (Only required for operating systems other than SLES)
   c. cloudera-navencryptfs-kmp (Only required for the SLES operating system)
   d. libkeytrustee
8. If rolling back to a release less than NavEncrypt 6.2
   a. Reapply the ACL rules:

      ```
      sudo navencrypt acl --add --file=acls.txt
      ```

   b. Recompute process signatures:

      ```
      sudo navencrypt acl --update
      ```

9. Restart the Navigator Encrypt mount service

   ```
   sudo /etc/init.d/navencrypt-mount start
   ```

(Optional) Cloudera Manager Rollback Steps

After you complete the rollback steps, your cluster is using Cloudera Manager 7 to manage your CDH 6 or CDH 6 cluster. You can continue to use Cloudera Manager 7 to manage your CDH 6 cluster, or you can downgrade to Cloudera Manager 6 by following these steps:

Stop Cloudera Manager

1. Stop the .
   a. Log in to the Admin Console.
   b. Select Clusters .
   c. Select Actions Stop .
2. Stop the Server.

   ```
   sudo systemctl stop cloudera-scm-server
   ```

3. Hard stop the Cloudera Manager agents. Run the following command on all hosts:

   ```
   sudo systemctl stop cloudera-scm-supervisord.service
   ```

4. Back up the repository directory. You can create a top-level backup directory and an environment variable to reference the directory using the following commands. You can also substitute another directory path in the backup commands below:

   ```
   export CM_BACKUP_DIR="`date +%F`-CM"
   mkdir -p $CM_BACKUP_DIR
   ```

5. Back up the existing repository directory.
   RHEL / CentOS

   ```
   sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/yum.repos.d
   ```

   SLES

   ```
   sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/zypp/repos.d
   ```

   Ubuntu

   ```
   sudo -E tar -cf $CM_BACKUP_DIR/repository.tar /etc/apt/sources
   .list.d
   ```

Restore the Cloudera Manager 6 Repository Files

Copy the repository directory from the backup taken before upgrading to Cloudera Manager 7.x.

```
rm -rf /etc/yum.repos.d/*
tar -xf cm6cdh6_backedUp_dir/repositary.tar -C CM6CDH6/
cp -rp /etc/yum.repos.d_CM6CDH6/* /etc/yum.repos.d/
```

Restore Packages

1. Run the following commands on all hosts:

Operating System	Command

RHEL	`sudo yum remove cloudera-manager-daemons cloudera-manager-agent`
	
	`sudo yum clean all`
	`sudo yum install cloudera-manager-agent`

SLES	`sudo zypper remove cloudera-manager-daemons cloudera-manager-agent`
	
	`sudo zypper refresh -s`
	`sudo zypper install cloudera-manager-agent`

Ubuntu or Debian	`sudo apt-get purge cloudera-manager-daemons cloudera-manager-agent`
	
	`sudo apt-get update`
	`sudo apt-get install cloudera-manager-agent`

2. Run the following commands on the Cloudera Manager server host:

Operating System	Command
RHEL	`sudo yum remove cloudera-manager-server` `sudo yum install cloudera-manager-server`
SLES	`sudo zypper remove cloudera-manager-server` `sudo zypper install cloudera-manager-server`
Ubuntu or Debian	`sudo apt-get purge cloudera-manager-server` `sudo apt-get install cloudera-manager-server`

Restore Cloudera Manager Databases

Restore the Cloudera Manager databases from the backup of Cloudera Manager that was taken before upgrading to Cloudera Manager 7. See the procedures provided by your database vendor.

These databases include the following:

- Cloudera Manager Server
- Reports Manager
- Navigator Audit Server
- Navigator Metadata Server
- Activity Monitor (Only used for MapReduce 1 monitoring).

- MariaDB 5.5: http://mariadb.com/kb/en/mariadb/backup-and-restore-overview/
- MySQL 5.5: http://dev.mysql.com/doc/refman/5.5/en/backup-and-recovery.html
- MySQL 5.6: http://dev.mysql.com/doc/refman/5.6/en/backup-and-recovery.html
- PostgreSQL 8.4: https://www.postgresql.org/docs/8.4/static/backup.html
- PostgreSQL 9.2: https://www.postgresql.org/docs/9.2/static/backup.html
- PostgreSQL 9.3: https://www.postgresql.org/docs/9.3/static/backup.html
- Oracle 11gR2: https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm
- HyperSQL: http://hsqldb.org/doc/guide/management-chapt.html#mtc_backup

Here is an sample command to restore a MySQL database:

```
mysql -u username -ppassword --host=hostname cm < backup.sql
```

Restore Cloudera Manager Server

Use the backup of Cloudera Manager 6.x taken before upgrading to Cloudera Manager 7.x for the following steps:

1. If you used the backup commands provided in Step 2: Backing Up Cloudera Manager 6 on page 71, extract the Cloudera Manager 6 backup archives you created:

```
tar -xf CM6CDH6/cloudera-scm-agent.tar -C CM6CDH6/
tar -xf CM6CDH6/cloudera-scm-server.tar -C CM6CDH6/
```

2. On the host where the Event Server role is configured to run, restore the Events Server directory from the Cloudera Manager 6 backup.

```
cp -rp /var/lib/cloudera-scm-eventserver /var/lib/cloudera-scm-eventserv
er-CM
rm -rf /var/lib/cloudera-scm-eventserver/*
cp -rp /var/lib/cloudera-scm-eventserver_CM6CDH6/* /var/lib/cloudera-scm-
eventserver/
```

3. Remove the Agent runtime state. Run the following command on all hosts:

```
rm -rf /var/run/cloudera-scm-agent /var/lib/cloudera-scm-agent/response.
avro
```

4. On the host where the Service Monitor is running, restore the Service Monitor directory:

```
rm -rf /var/lib/cloudera-service-monitor/*
cp -rp /var/lib/cloudera-service-monitor_CM6CDH6/* /var/lib/cloudera-ser
vice-monitor/
```

5. On the host where the Host Monitor is running, restore the Host Monitor directory:

```
rm -rf /var/lib/cloudera-host-monitor/*
cp -rp /var/lib/cloudera-host-monitor_CM6CDH6/* /var/lib/cloudera-host-mo
nitor/
```

6. Restore the Cloudera Navigator Solr storage directory from the CM6/CDH6 backup.

```
rm -rf /var/lib/cloudera-scm-navigator/*
cp -rp /var/lib/cloudera-scm-navigator_CM6CDH6/* /var/lib/cloudera-scm-nav
igator/
```

7. On the Cloudera Manager Server, restore the /etc/cloudera-scm-server/db.properties file.

```
rm -rf /etc/cloudera-scm-server/db.properties
cp -rp CM6CDH6/etc/cloudera-scm-server/db.properties /etc/cloudera-scm-ser
ver/db.properties
```

8. On each host in the cluster, restore the /etc/cloudera-scm-agent/config.ini file from your backup.

```
rm -rf /etc/cloudera-scm-agent/config.ini
cp -rp CM6CDH6/etc/cloudera-scm-agent/config.ini /etc/cloudera-scm-agent/c
onfig.ini
```

Start the Cloudera Manager Server and Agents

- Start the Server.

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

- Hard Restart the Agent.
  RHEL 7, SLES 12, Ubuntu 18.04 and higher

```
sudo systemctl stop cloudera-scm-supervisord.service
sudo systemctl restart cloudera-scm-agent
```

- Start the .
  1. Log in to the Admin Console.
  2. Select Clusters .
  3. Select Actions Start .

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.

Note: This internal web repository can also be used when your Cloudera Manager server or cluster does not have access to internet. You must download the installable separately from archive.cloudera.com and place the installable in the internal repository.

Important: Select a supported operating system for the versions of or CDH that you are downloading. See CDH and Supported Operating Systems.

Warning: Upgrades from 5.12 and lower to 7.1.1 or higher are not supported

Important: Upgrading to version 7.7.1 or higher from clusters where CDH 5.x is deployed is not supported. To upgrade such clusters:

1. Upgrade to version 6.3.4.
2. Upgrade CDH to version 6.3.4
3. Upgrade to version 7.6.5 or higher

Warning: For upgrades from CDH clusters with Navigator to 7.1.1 (or higher) clusters where Navigator is to be migrated to Apache Atlas, the cluster must have Kerberos enabled before upgrading.

Warning: Before upgrading CDH 5 clusters with Sentry to Cloudera Runtime 7.1.x clusters where Sentry privileges are to be transitioned to Apache Ranger:

- The cluster must have Kerberos enabled.
- Verify that HDFS gateway roles exist on the hosts that runs the Sentry service.

Important: If HDFS ACL sync is enabled (hdfs_sentry_sync_enable=true) on the CDH cluster, then you must install Ranger RMS to support the same functionality. For steps to install Ranger RMS, see Installing Ranger RMS.

Note: If the cluster you are upgrading will include Atlas, Ranger, or both, the upgrade wizard deploys one infrastructure Solr service to provide a search capability of the audit logs through the Ranger Admin UI and/or to store and serve Atlas metadata. recommends that you do not use this service for customer workloads to avoid interference with audit and timeline performance.

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. 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 for Cloudera Manager on page 259.

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 8

   ```
   sudo systemctl start httpd
   ```

   SLES 12, Ubuntu 16 or later

   ```
   sudo systemctl start apache2
   ```

Downloading and publishing the package repository for Cloudera Manager

1. Download the package repository for the product you want to install:
   Cloudera Manager 7

   Do the following steps to download the files for a Cloudera Manager release:

   a. Run the following command to create a local repository directory to hold the Cloudera package repository:

   ```
   sudo mkdir -p /var/www/html/cloudera-repos/cm7
   ```

   b. Run the following command to download the repository tarball for your operating system:

   ```
   wget https://[USERNAME]:[PASSWORD]@archive.cloudera.com/p/
   cm7/7.0.3/repo-as-tarball/cm7.0.3-redhat7.tar.gz
   ```

   c. Run the following command to unpack the tarball into the local repository directory:

   ```
   tar xvfz cm7.0.3-redhat7.tar.gz -C /var/www/html/cloudera-r
   epos/cm7 --strip-components=1
   ```

   d. Run the following command to modify the file permission that allows you to download the files under the local repository directory:

   ```
   sudo chmod -R ugo+rX /var/www/html/cloudera-repos/cm7
   ```
2. 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.

Important:

If you downloaded the rpm files individually instead of using repo-as-tarball, ensure that the allkeys*.asc files (allkeys.asc and the allkeyssha256.asc for Cloudera Manager 7.11.2 and later versions) are present at the top level of the package repository. The allkeys*.asc files are included in the repo-as-tarball file. Ensure to include allkeys*.asc files, if you are manually copying the package files between hosts.

The allkeys*.asc files are used to validate the signatures of the package files during host installation. If allkeys*.asc files are not available in the repository, then you cannot add a host in the Cloudera Manager.

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. This example uses Python SimpleHTTPServer as the Web server to host the /var/www/html directory, but you can use a different directory.

1. Download the repository you need following the instructions in Downloading and publishing the package repository for Cloudera Manager on page 259.
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 establishing the repository, modify the client configuration to use it:

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>/cm/5`
	`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>`

OS	Procedure

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 Runtime 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.

Using an Internally Hosted Remote Parcel Repository

The following sections describe how to use an internal Web server to host a parcel repository:

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. 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 on page 263.

1. Install Apache HTTP Server:
   RHEL / CentOS

   ```
   sudo yum install httpd
   ```

   SLES

   ```
   sudo zypper install httpd
   ```

   Ubuntu

   ```
   sudo apt-get install httpd
   ```

2. **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.

   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.

   ```
   <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>
   ```

3. Start Apache HTTP Server:
   RHEL 8

   ```
   sudo systemctl start httpd
   ```

   SLES 12, Ubuntu 16 or later

   ```
   sudo systemctl start apache2
   ```

Downloading and Publishing the Parcel Repository

1. Look up the Cloudera Runtime *VERSION* number for your deployment on the Cloudera Runtime Download Information page. You will need this version number in the next step.
2. Download manifest.json and the parcel files for the product you want to install:

   To download the files for the latest Cloudera Runtime 7 release, run the following commands on the Web server host:

   ```
   sudo mkdir -p /var/www/html/cloudera-repos
   sudo wget --recursive --no-parent --no-host-directories https://
   [username]:[password]@archive.cloudera.com/p/cdh7/CLOUDERA RUNTIME
    VERSION/parcels/ -P /var/www/html/cloudera-repos

   sudo chmod -R ugo+rX /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.

Configuring Cloudera Manager to Use an Internal Remote Parcel Repository

1. Use one of the following methods to open the parcel settings page:
   - Navigation bar
     a. Click the parcel icon in the top navigation bar or click Hosts and click the Parcels tab.
     b. Click the Configuration button.
   - Menu
     a. Select AdministrationSettings.
     b. Select CategoryParcels.
2. In the Remote Parcel Repository URLs list, click the addition symbol to open an additional row.
3. Enter the path to the parcel. For example: http://<*WEB_SERVER*>/cloudera-parcels/cdh7/7.2.16.0.0/
4. Enter a Reason for change, and then click Save Changes to commit the changes.

Using a Local Parcel Repository

To use a local parcel repository, complete the following steps:

1. Open the Cloudera Manager Admin Console and navigate to the Parcels page.
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 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 on page 263 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 CDH-6.1.0-1.cdh6.1.0.p0.770702-el7.parcel:

   ```
   sha1sum CDH-6.1.0-1.cdh6.1.0.p0.770702-el7.parcel | awk '{ print $1 }'
    > CDH-6.1.0-1.cdh6.1.0.p0.770702-el7.parcel.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, navigate to the Parcels page.
9. Click Check for New Parcels and verify that the new parcel appears.
10. Download, distribute, and activate the parcel.

CDH 6 to Cloudera Base on premises post-upgrade transition steps

You need to understand the upgrade configuration changes and perform tasks related to Hive, Impala, Solr, and other components. CDP does not support some features in CDH clusters, but alternatives might suffice.

Update permissions for Replication Manager service

After the upgrade process is complete, you must update the permissions in the Ranger audit log path in HDFS, so that the data replication using Replication Manager works as expected.

Before you begin

Attention: If your Cloudera Base on premises cluster has Ranger configured, the hdfs user should have access to all Hive datasets, including all operations. Else, Hive import fails during the replication process. For more information, see Providing access to hdfs user.

Perform the following steps to update the Ranger audit permissions:

Procedure

1. Add the user to the user-groups (supergroup, hdfs, hadoop) on all the hosts, including the source and target clusters.

   The user name you specify here is to be used in the Run as Username field when you create a replication policy to run the replication job.
2. Provide “hive” user permissions in HDFS in the Ranger UI.

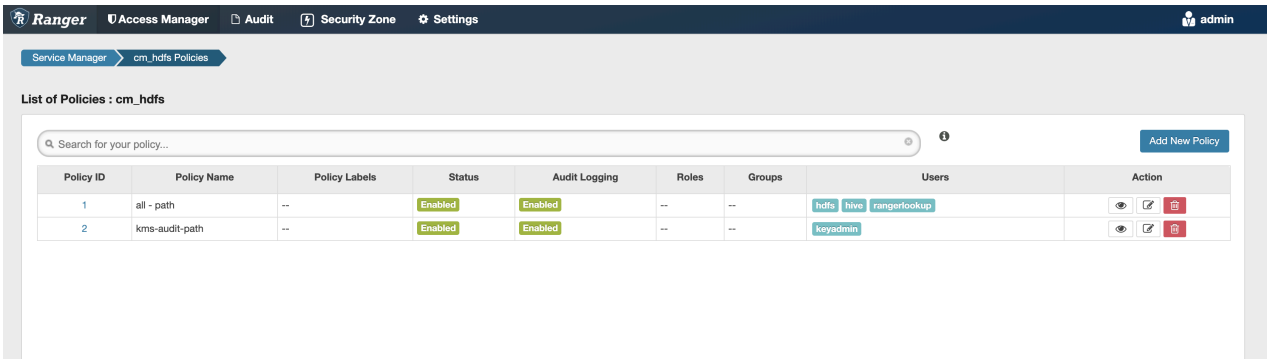
What to do next

1. To add the user to the user-groups, run the following commands:

   ```
   > sudo usermod -a -G hdfs [***USER***]
   > sudo usermod -a -G hadoop [***USER***]
   > id -Gn [***USER***]
   > sudo groupadd supergroup
   > sudo usermod -a -G supergroup [***USER***]
   > hdfs groups [***USER***]
   ```

2. To provide permissions for the Ranger audit log path in HDFS:

- Log in to Ranger Admin UI.
- Provide "hive" user permission to "all-path" in hdfs under the cm_hdfs section.



Migrating Spark workloads to Cloudera

Migrating Spark workloads from CDH or HDP to Cloudera involves learning the Spark semantic changes in your source cluster and the Cloudera target cluster. You get details about how to handle these changes.

Spark 2.3 to Spark 2.4 Refactoring

Because Spark 2.3 is not supported on Cloudera, you need to refactor Spark workloads from Spark 2.3 on CDH or HDP to Spark 2.4 on Cloudera.

This document helps in accelerating the migration process, provides guidance to refactor Spark workloads and lists migration. Use this document when the platform is migrated from CDH or HDP to Cloudera.

Handling prerequisites

You must perform a number of tasks before refactoring workloads.

About this task

Assuming all workloads are in working condition, you perform this task to meet refactoring prerequistes.

Procedure

1. Identify all the workloads in the cluster (CDH/HDP) which are running on Spark 1.6 - 2.3.

2. Classify the workloads.

   Classification of workloads will help in clean-up of the unwanted workloads, plan resources and efforts for workload migration and post upgrade testing.

   Example workload classifications:

   - Spark Core (scala)
   - Java-based Spark jobs
   - SQL, Datasets, and DataFrame
   - Structured Streaming
   - MLlib (Cloudera AI)
   - PySpark (Python on Spark)
   - Batch Jobs
   - Scheduled Jobs
   - Ad-Hoc Jobs
   - Critical/Priority Jobs
   - Huge data Processing Jobs
   - Time taking jobs
   - Resource Consuming Jobs etc.
   - Failed Jobs

Identify configuration changes

3. Check the current Spark jobs configuration.

   - Spark 1.6 - 2.3 workload configurations which have dependencies on job properties like scheduler, old python packages, classpath jars and might not be compatible post migration.
   - In Cloudera, Capacity Scheduler is the default and recommended scheduler. Follow Fair Scheduler to Capacity Scheduler transition guide to have all the required queues configured in the Cloudera cluster post upgrade. If any configuration changes are required, modify the code as per the new capacity scheduler configurations.
   - For workload configurations, see the Spark History server UI http://spark_history_server:18088/history/<application_number>/environment/.

4. Identify and capture workloads having data storage locations (local and HDFS) to refactor the workloads post migration.
5. Refer to unsupported Apache Spark features, and plan refactoring accordingly.

Spark 2.3 to Spark 2.4 changes

A description of the change, the type of change, and the required refactoring provide the information you need for migrating from Spark 2.3 to Spark 2.4.

Empty schema not supported

Writing a dataframe with an empty or nested empty schema using any file format, such as parquet, orc, json, text, or csv is not allowed.

Type of change: Syntactic/Spark core

Spark 1.6 - 2.3

Writing a dataframe with an empty or nested empty schema using any file format is allowed and will not throw an exception.

Spark 2.4

An exception is thrown when you attempt to write dataframes with empty schema. For example, if there are statements such as df.write.format("parquet").mode("overwrite").save(somePath), the following error occurs: org.apache.spark.sql.AnalysisException: Parquet data source does not support null data type.

Action Required

Make sure that DataFrame is not empty. Check whether DataFrame is empty or not as follows:

```
if (!df.isEmpty) df.write.format("parquet").mode("overwrite").save("somePath
")
```

CSV header and schema match

Column names of csv headers must match the schema.

Type of change: Configuration/Spark core changes

Spark 1.6 - 2.3

Column names of headers in CSV files are not checked against the against the schema of CSV data.

Spark 2.4

If columns in the CSV header and the schema have different ordering, the following exception is thrown:java.lang.Il legalArgumentException: CSV file header does not contain the expected fields.

Action Required

Make the schema and header order match or set enforceSchema to false to prevent getting an exception. For example, read a file or directory of files in CSV format into Spark DataFrame as follows: df3 = spark.read.option("delimiter", ";").option("header", True).option("enforeSchema", False).csv(path)

The default "header" option is true and enforceSchema is False.

If enforceSchema is set to true, the specified or inferred schema will be forcibly applied to datasource files, and headers in CSV files are ignored. If enforceSchema is set to false, the schema is validated against all headers in CSV files when the header option is set to true. Field names in the schema and column names in CSV headers are checked by their positions taking into account spark.sql.caseSensitive. Although the default value is true,you should disable the enforceSchema option to prevent incorrect results.

Table properties support

Table properties are taken into consideration while creating the table.

Type of change: Configuration/Spark Core Changes

Spark 1.6 - 2.3

Parquet and ORC Hive tables are converted to Parquet or ORC by default, but table properties are ignored. For example, the compression table property is ignored:

```
CREATE TABLE t(id int) STORED AS PARQUET TBLPROPERTIES (parquet.compression
'NONE')
```

This command generates Snappy Parquet files.

Spark 2.4

Table properties are supported. For example, if no compression is required, set the TBLPROPERTIES as follows: (parquet.compression 'NONE').

This command generates uncompressed Parquet files.

Action Required

Check and set the desired TBLPROPERTIES.

Managed table location

Creating a managed table with nonempty location is not allowed.

Type of change: Property/Spark core changes

Spark 1.6 - 2.3

You can create a managed table having a nonempty location.

Spark 2.4

Creating a managed table with nonempty location is not allowed. In Spark 2.4, an error occurs when there is a write operation, such as df.write.mode(SaveMode.Overwrite).saveAsTable("testdb.testtable"). The error side-effects are the cluster is terminated while the write is in progress, a temporary network issue occurs, or the job is interrupted.

Action Required

Set spark.sql.legacy.allowCreatingManagedTableUsingNonemptyLocation to true at runtime as follows:

```
spark.conf.set("spark.sql.legacy.allowCreatingManagedTableUsingNonemptyLocat
ion","true")
```

Precedence of set operations

Set operations are executed by priority instead having equal precedence.

Type of change: Property/Spark SQL changes

Spark 1.6 - 2.3

If the order is not specified by parentheses, equal precedence is given to all set operations.

Spark 2.4

If the order is not specified by parentheses, set operations are performed from left to right with the exception that all INTERSECT operations are performed before any UNION, EXCEPT or MINUS operations.

For example, if your code includes set operations, such as INTERSECT , UNION, EXCEPT or MINUS, consider refactoring.

Action Required

Change the logic according to following rule:

If the order of set operations is not specified by parentheses, set operations are performed from left to right with the exception that all INTERSECT operations are performed before any UNION, EXCEPT or MINUS operations.

If you want the previous behavior of equal precedence then, set spark.sql.legacy.setopsPrecedence.enabled=true.

HAVING without GROUP BY

HAVING without GROUP BY is treated as a global aggregate.

Type of change: Property/Spark SQL changes

Spark 1.6 - 2,3

HAVING without GROUP BY is treated as WHERE. For example, SELECT 1 FROM range(10) HAVING true is executed as SELECT 1 FROM range(10) WHERE true, and and returns 10 rows.

Spark 2.4

HAVING without GROUP BY is treated as a global aggregate. For example, SELECT 1 FROM range(10) HAVING true returns one row, instead of 10, as in the previous version.

Action Required

Check the logic where having and group by is used. To restore previous behavior, set spark.sql.legacy.parser.havingWithoutGroupByAsWhere=true.

CSV bad record handling

How Spark treats malformations in CSV files has changed.

Type of change: Property/Spark SQL changes

Spark 1.6 - 2.3

CSV rows are considered malformed if at least one column value in the row is malformed. The CSV parser drops malformed rows in the DROPMALFORMED mode or outputs an error in the FAILFAST mode.

Spark 2.4

A CSV row is considered malformed only when it contains malformed column values requested from CSV datasource, other values are ignored.

Action Required

To restore the Spark 1.6 behavior, set spark.sql.csv.parser.columnPruning.enabled to false.

Spark 2.4 CSV example

A CSV example illustrates the CSV-handling change in Spark 2.4.

In the following CSV file, the first two records describe the file. These records are not considered during processing and need to be removed from the file. The actual data to be considered for processing has three columns (jersey, name, position).

```
These are extra line1
These are extra line2
10,Messi,CF
7,Ronaldo,LW
9,Benzema,CF
```

The following schema definition for the DataFrame reader uses the option DROPMALFORMED. You see only the required data; all the description and error records are removed.

```
schema=Structtype([Structfield(“jersy”,StringType()),Structfield(“name”,Stri
ngType()),Structfi
eld(“position”,StringType())])
df1=spark.read\
.option(“mode”,”DROPMALFORMED”)\
.option(“delimiter”,”,”)\
.schema(schema)\
.csv(“inputfile”)
df1.select(“*”).show()
```

Output is:

jersy	name	position
10	Messi	CF
7	Ronaldo	LW
9	Benzema	CF

Select two columns from the dataframe and invoke show():

```
df1.select(“jersy”,”name”).show(truncate=False)
```

jersy	name
These are extra line1	null
These are extra line2	null
10	Messi
7	Ronaldo
9	Benzema

Malformed records are not dropped and pushed to the first column and the remaining columns will be replaced with null.This is due to the CSV parser column pruning which is set to true by default in Spark 2.4.

Set the following conf, and run the same code, selecting two fields.

```
spark.conf.set(“spark.sql.csv.parser.columnPruning.enabled”,False)
```

```
df2=spark.read\
   .option(“mode”,”DROPMALFORMED”)\
   .option(“delimiter”,”,”)\
   .schema(schema)\
   .csv(“inputfile”)
   df2.select(“jersy”,”name”).show(truncate=False)
```

jersy	name
10	Messi
7	Ronaldo
9	Benzema

Conclusion: If working on selective columns, to handle bad records in CSV files, set spark.sql.csv.parser.columnP runing.enabled to false; otherwise, the error record is pushed to the first column, and all the remaining columns are treated as nulls.

Configuring storage locations

To execute the workloads in Cloudera, you must modify the references to storage locations. In Cloudera, references must be changed from HDFS to a cloud object store such as S3.

About this task

The following sample query shows a Spark 2.4 HDFS data location.

```
scala> spark.sql("CREATE TABLE IF NOT EXISTS default.sales_spark_2(Region st
ring, Country string,Item_Type string,Sales_Channel string,Order_Priority st
ring,Order_Date date,Order_ID int,Ship_Date date,Units_sold string,Unit_Pric
e string,Unit_cost string,Total_revenue string,Total_Cost string,Total_Profi
t string) row format delimited fields terminated by ','")
scala> spark.sql("load data local inpath '/tmp/sales.csv' into table defau
lt.sales_spark_2")
scala> spark.sql("select count(*) from default.sales_spark_2").show()
```

The following sample query shows a Spark 2.4 S3 data location.

```
scala> spark.sql("CREATE TABLE IF NOT EXISTS default.sales_spark_2(Region st
ring, Country string,Item_Type string,Sales_Channel string,Order_Priority st
ring,Order_Date date,Order_ID int,Ship_Date date,Units_sold string,Unit_Pric
e string,Unit_cost string,Total_revenue string,Total_Cost string,Total_Profi
t string) row format delimited fields terminated by ','")
scala> spark.sql("load data inpath 's3://<bucket>/sales.csv' into table de
fault.sales_spark_2")
scala> spark.sql("select count(*) from default.sales_spark_2").show()
```

Querying Hive managed tables from Spark

Hive-on-Spark is not supported on Cloudera. You need to use the Hive Warehouse Connector (HWC) to query Apache Hive managed tables from Apache Spark.

To read Hive external tables from Spark, you do not need HWC. Spark uses native Spark to read external tables. For more information, see the Hive Warehouse Connector documentation.

The following example shows how to query a Hive table from Spark using HWC:

```
spark-shell --jars /opt/cloudera/parcels/CDH/jars/hive-warehouse-connector-a
ssembly-1.0.0.7.1.4.0-203.jar --conf spark.sql.hive.hiveserver2.jdbc.url=jdb
c:hive2://cdhhdp02.uddeepta-bandyopadhyay-s-account.cloud:10000/default --co
nf spark.sql.hive.hiveserver2.jdbc.url.principal=hive/cdhhdp02.uddeepta-band
yopadhyay-s-account.cloud@Uddeepta-bandyopadhyay-s-Account.CLOUD
scala> val hive = com.hortonworks.hwc.HiveWarehouseSession.session(spark).b
uild()
scala> hive.executeUpdate("UPDATE hive_acid_demo set value=25 where key=4")
scala> val result=hive.execute("select * from default.hive_acid_demo")
scala> result.show()
```

Compiling and running Spark workloads

After modifying the workloads, compile and run (or dry run) the refactored workloads on Spark 2.4.

You can write Spark applications using Java, Scala, Python, SparkR, and others. You build jars from these scripts using one of the following compilers.

- Java (with Maven/Java IDE),
- Scala (with sbt),
- Python (pip).
- SparkR (RStudio)

Post-migration tasks

After the workloads are executed on Spark 2.4, validate the output, and compare the performance of the jobs with CDH/HDP cluster executions.

After the workloads are executed on Spark 2.4, validate the output, and compare the performance of the jobs with CDH/HDP cluster executions. After you perform the post migration configurations, do benchmark testing on Spark 2.4.

Troubleshoot the failed/slow performing workloads by analyzing the application event logs/driver logs and fine tune the workloads for better performance.

For more information, see the following documents:

- https://spark.apache.org/docs/2.4.4/sql-migration-guide-upgrade.html

  https://spark.apache.org/releases/spark-release-2-4-0.html

  https://spark.apache.org/releases/spark-release-2-2-0.html

  https://spark.apache.org/releases/spark-release-2-3-0.html

  https://spark.apache.org/releases/spark-release-2-1-0.html

  https://spark.apache.org/releases/spark-release-2-0-0.html

  For additional information about known issues please also refer:

  Known Issues in Cloudera Manager 7.4.4 | CDP Private Cloud

  For additional information about known issues please also refer to:

  Known Issues in Cloudera Manager 7.4.4 | Cloudera on premises

Apache Hive Expedited Migration Tasks

If you chose to expedite the Hive upgrade process by postponing migration of your tables and databases, you need to identify any problems in tables and get help with fixing those problems before migrating the tables to CDP. You then need to migrate these tables to CDP before you can use them.

Preparing tables for migration

You download the Hive Upgrade Check tool and use it to identify problems in unmigrated tables. These problems can cause upgrade failure. It saves time to fix the problems and avoid failure. The tool provides help for fixing those problems before migrating the tables to CDP.

About this task

You use the Hive Upgrade Check community tool to help you identify tables that have problems affecting migration. You resolve problems revealed by the Hive Upgrade Check tool to clean up the Hive Metastore before migration. If you do not want to use the Hive Upgrade Check tool, you need to perform the tasks described in the following subtopics to migrate Hive data to CDP:

- Check SERDE Definitions and Availability
- Handle Missing Table or Partition Locations
- Manage Table Location Mapping
- Make Tables SparkSQL Compatible

Procedure

1. Obtain the Hive Upgrade Check tool.

   Download the Hive SRE Upgrade Check tool from the Cloudera labs github location.
2. Follow instructions in the github readme to run the tool.
   The Hive Upgrade Check (v.2.3.5.6+) will create a yaml file (hsmm_<name>.yaml) identifying databases and tables that require attention.
3. Follow instructions in prompts from the Hive Upgrade Check tool to resolve problems with the tables.

   At a minimum, you must run the following processes described in the github readme:

   - process ID 1 Table / Partition Location Scan - Missing Directories
   - process id 3 Hive 3 Upgrade Checks - Managed Non-ACID to ACID Table Migrations

Check SERDE Definitions and Availability

Ensure correct Serde definitions and a reference to a SERDE exists to ensure a successful upgrade.

About this task

You perform this step if you do not modify the HSMM process for expediting the Hive upgrade.

Procedure

1. Check Serde definitions for correctness and check for SERDE availability.
2. Correct any problems found as follows:

   - Remove the table having the problematic SERDE.
   - Ensure the SERDE is available during the upgrade, so the table can be evaluated.

Handle Missing Table or Partition Locations

You need to identify missing table or partition locations, or both, to prevent upgrade failure. If the table and partition locations do not exist in the file system, you must either create a replacement partition directory (recommended) or drop the table and partition.

About this task

You perform this step if you did not modify the HSMM process to expedite the Hive upgrade.

Procedure

Ensure the table and partition locations exist on the file system. If these locations don’t exist either create a replacement partition directory (recommended) or drop the table and partition.

Managed Table Location Mapping

A managed table location must map to one managed table only. If multiple managed tables point to the same location, upgrade problems occur.

Make Tables SparkSQL Compatible

Non-Acid, managed tables in ORC or in a Hive Native (but non-ORC) format that are owned by the POSIX user hive will not be SparkSQL-compatible after the upgrade unless you perform manual conversions.

About this task

If your table is a managed, non-ACID table, you can convert it to an external table using this procedure (recommended). After the upgrade, you can easily convert the external table to an ACID table, and then use the Hive Warehouse Connector to access the ACID table from Spark.

Take one of the following actions.

- Convert the tables to external Hive tables before the upgrade.

  ALTER TABLE ... SET TBLPROPERTIES('EXTERNAL'='TRUE','external.table.purge'='true')
- Change the POSIX ownership to an owner other than hive.

  You will need to convert managed, ACID v1 tables to external tables after the upgrade.

Creating a list of tables to migrate

To run Hive Strict Managed Migration process (HSMM) after upgrading, you need to know how to create a YAML file that specifies the tables for migration.

Procedure

Create a YAML file in the following format:

```
---
databaseIncludeLists:
  <database name>:
    - "<table name>"
    - "<table name2>"
    - ...
  <database name 2>:
    - ...
```

For example:

```
---
databaseIncludeLists:
  tpcds_bin_partitioned_orc_10:
    - "call_center"
    - "catalog_page"
    - "catalog_returns"
    - "customer"
    - "customer_address"
  bu_raw:
    - "cc_input"
    - "geo_regions"
```

Migrating tables to CDP

You set a Hive property to point to your YAML list of tables you want to migrate, and then migrate the tables by manually running the Hive Strict Managed Migration process on the tables. You perform this action to use the tables in CDP.

About this task

In this task you set the table migration control file URL property to the path of a YAML file that lists tables and databases you want to migrate. You run the HSMM process to migrate the tables and databases using Cloudera Manager.

Before you begin

- You completed the upgrade to CDP.
- You created a YAML file listing databases and tables to migrate.

Procedure

1. In Cloudera Manager, go to Clusters Hive-on-Tez .
2. Stop the Hive-on-Tez service.
3. In Configuration , search for table migration control file URL.
4. Set the value of the Table migration control file URL property to the absolute path and file name of your YAML include list.
5. Save configuration changes.
6. Click Clusters Hive-on-Tez , and in Actions, click Migrate Hive tables for CDP upgrade.
   HSMM migrates the Hive tables listed in the YAML.
7. To prevent problems with any subsequent HSMM run, remove the value you set for Table migration control file URL, leaving the value blank.
8. Save configuration changes.
9. Start the Hive-on-Tez service.
   The YAML-specified tables and databases are migrated.

Apache Hive Changes in CDP

You need to know where your tables are located and the property changes that the upgrade process makes. You need to perform some post-migration tasks before using Hive tables and handle semantic changes.

Understanding Apache Hive 3 major design features, such as default ACID transaction processing, can help you use Hive to address the growing needs of enterprise data warehouse systems.

If you are expediting the Hive upgrade process and modified the upgrade process to skip materializing every table in the metastore, you need to modify the Hive Strict Metastore Migration (HSMM) process by running the Hive Upgrade Check tool and provided scripts. Scripts are not included to address legacy Kudu storage handler classes.

Related Information

Apache Hive 3 Key Features

Apache Hive 3 Architectural Overview

Preparing tables for migration

You download the Hive Upgrade Check tool and use it to identify problems in unmigrated tables. These problems can cause upgrade failure. It saves time to fix the problems and avoid failure. The tool provides help for fixing those problems before migrating the tables to CDP.

About this task

You use the Hive Upgrade Check community tool to help you identify tables that have problems affecting migration. You resolve problems revealed by the Hive Upgrade Check tool to clean up the Hive Metastore before migration. If you do not want to use the Hive Upgrade Check tool, you need to perform the tasks described in the following subtopics to migrate Hive data to CDP:

- Check SERDE Definitions and Availability
- Handle Missing Table or Partition Locations
- Manage Table Location Mapping
- Make Tables SparkSQL Compatible

Procedure

1. Obtain the Hive Upgrade Check tool.

   Download the Hive SRE Upgrade Check tool from the Cloudera labs github location.
2. Follow instructions in the github readme to run the tool.
   The Hive Upgrade Check (v.2.3.5.6+) will create a yaml file (hsmm_<name>.yaml) identifying databases and tables that require attention.
3. Follow instructions in prompts from the Hive Upgrade Check tool to resolve problems with the tables.

   At a minimum, you must run the following processes described in the github readme:

   - process ID 1 Table / Partition Location Scan - Missing Directories
   - process id 3 Hive 3 Upgrade Checks - Managed Non-ACID to ACID Table Migrations

Check SERDE Definitions and Availability

Ensure correct Serde definitions and a reference to a SERDE exists to ensure a successful upgrade.

About this task

You perform this step if you do not modify the HSMM process for expediting the Hive upgrade.

Procedure

1. Check Serde definitions for correctness and check for SERDE availability.
2. Correct any problems found as follows:

   - Remove the table having the problematic SERDE.
   - Ensure the SERDE is available during the upgrade, so the table can be evaluated.

Handle Missing Table or Partition Locations

You need to identify missing table or partition locations, or both, to prevent upgrade failure. If the table and partition locations do not exist in the file system, you must either create a replacement partition directory (recommended) or drop the table and partition.

About this task

You perform this step if you did not modify the HSMM process to expedite the Hive upgrade.

Procedure

Ensure the table and partition locations exist on the file system. If these locations don’t exist either create a replacement partition directory (recommended) or drop the table and partition.

Managed Table Location Mapping

A managed table location must map to one managed table only. If multiple managed tables point to the same location, upgrade problems occur.

Make Tables SparkSQL Compatible

Non-Acid, managed tables in ORC or in a Hive Native (but non-ORC) format that are owned by the POSIX user hive will not be SparkSQL-compatible after the upgrade unless you perform manual conversions.

About this task

If your table is a managed, non-ACID table, you can convert it to an external table using this procedure (recommended). After the upgrade, you can easily convert the external table to an ACID table, and then use the Hive Warehouse Connector to access the ACID table from Spark.

Take one of the following actions.

- Convert the tables to external Hive tables before the upgrade.

  ALTER TABLE ... SET TBLPROPERTIES('EXTERNAL'='TRUE','external.table.purge'='true')

- Change the POSIX ownership to an owner other than hive.

  You will need to convert managed, ACID v1 tables to external tables after the upgrade.

Hive Configuration Property Changes

You need to know the property value changes made by the upgrade process as the change might impact your work. You might need to consider reconfiguring property value defaults that the upgrade changes.

Hive Configuration Property Values

The upgrade process changes the default values of some Hive configuration properties and adds new properties. The following list describes those changes that occur after upgrading from CDH or HDP to Cloudera.

datanucleus.connectionPool.maxPoolSize

Before upgrade: 30

After upgrade: 10

datanucleus.connectionPoolingType

Before upgrade: BONECP

After upgrade: HikariCP

hive.auto.convert.join.noconditionaltask.size

Before upgrade: 20971520

After upgrade: 52428800

Exception: Preserves pre-upgrade value if old default is overridden; otherwise, uses new default.

hive.auto.convert.sortmerge.join

Before upgrade: FALSE in the old CDH; TRUE in the old HDP.

After upgrade: TRUE

hive.auto.convert.sortmerge.join.to.mapjoin

Before upgrade: FALSE

After upgrade: TRUE

hive.cbo.enable

Before upgrade: FALSE

After upgrade: TRUE

hive.cbo.show.warnings

Before upgrade: FALSE

After upgrade: TRUE

hive.compactor.worker.threads

Before upgrade: 0

After upgrade: 5

hive.compute.query.using.stats

Before upgrade: FALSE

After upgrade: TRUE

hive.conf.hidden.list

Before upgrade:

```
javax.jdo.option.ConnectionPassword,hive.server2.keystore.passwo
rd,hive.metastore.dbaccess.ssl.truststore.password,fs.s3.awsAcce
ssKeyId,fs.s3.awsSecretAccessKey,fs.s3n.awsAccessKeyId,fs.s3n.aw
sSecretAccessKey,fs.s3a.access.key,fs.s3a.secret.key,fs.s3a.prox
y.password,dfs.adls.oauth2.credential,fs.adl.oauth2.credential,f
s.azure.account.oauth2.client.secret
```

After upgrade:

```
javax.jdo.option.ConnectionPassword,hive.server2.keystore.passwo
rd,hive.druid.metadata.password,hive.driver.parallel.compilation
.global.limit
```

hive.conf.restricted.list

Before upgrade:

```
hive.security.authenticator.manager,hive.security.authorization.
manager,hive.users.in.admin.role,hive.server2.xsrf.filter.enable
d,hive.spark.client.connect.timeout,hive.spark.client.server.con
nect.timeout,hive.spark.client.channel.log.level,hive.spark.clie
nt.rpc.max.size,hive.spark.client.rpc.threads,hive.spark.client.
secret.bits,hive.spark.client.rpc.server.address,hive.spark.clie
nt.rpc.server.port,hive.spark.client.rpc.sasl.mechanisms,hadoop.
bin.path,yarn.bin.path,spark.home,bonecp.,hikaricp.,hive.driver.
parallel.compilation.global.limit,_hive.local.session.path,_hive
.hdfs.session.path,_hive.tmp_table_space,_hive.local.session.pat
h,_hive.hdfs.session.path,_hive.tmp_table_space
```

After upgrade:

```
hive.security.authenticator.manager,hive.security.authorization.
manager,hive.security.metastore.authorization.manager,hive.secur
ity.metastore.authenticator.manager,hive.users.in.admin.role,hiv
e.server2.xsrf.filter.enabled,hive.security.authorization.enable
d,hive.distcp.privileged.doAs,hive.server2.authentication.ldap.b
aseDN,hive.server2.authentication.ldap.url,hive.server2.authenti
cation.ldap.Domain,hive.server2.authentication.ldap.groupDNPatte
rn,hive.server2.authentication.ldap.groupFilter,hive.server2.aut
hentication.ldap.userDNPattern,hive.server2.authentication.ldap.
userFilter,hive.server2.authentication.ldap.groupMembershipKey,h
ive.server2.authentication.ldap.userMembershipKey,hive.server2.a
uthentication.ldap.groupClassKey,hive.server2.authentication.lda
p.customLDAPQuery,hive.privilege.synchronizer.interval,hive.spar
k.client.connect.timeout,hive.spark.client.server.connect.timeou
t,hive.spark.client.channel.log.level,hive.spark.client.rpc.max.
size,hive.spark.client.rpc.threads,hive.spark.client.secret.bits
,hive.spark.client.rpc.server.address,hive.spark.client.rpc.serv
er.port,hive.spark.client.rpc.sasl.mechanisms,bonecp.,hive.druid
.broker.address.default,hive.druid.coordinator.address.default,h
ikaricp.,hadoop.bin.path,yarn.bin.path,spark.home,hive.driver.pa
rallel.compilation.global.limit,_hive.local.session.path,_hive.h
dfs.session.path,_hive.tmp_table_space,_hive.local.session.path,
_hive.hdfs.session.path,_hive.tmp_table_space
```

hive.default.fileformat.managed

Before upgrade: None

After upgrade: ORC

hive.default.rcfile.serde

Before upgrade: org.apache.hadoop.hive.serde2.columnar.ColumnarSerDe

After upgrade: org.apache.hadoop.hive.serde2.columnar.LazyBinaryColumnarSerDe

Not supported in Impala. Impala cannot read Hive-created RC tables.

hive.driver.parallel.compilation

Before upgrade: FALSE

After upgrade: TRUE

hive.exec.dynamic.partition.mode

Before upgrade: strict

After upgrade: nonstrict

In Cloudera Base on premises, accidental use of dynamic partitioning feature is not prevented by default.

hive.exec.max.dynamic.partitions

Before upgrade: 1000

After upgrade: 5000

In Cloudera Base on premises, fewer restrictions on dynamic paritioning occur than in the pre-upgrade CDH or HDP cluster.

hive.exec.max.dynamic.partitions.pernode

Before upgrade: 100

After upgrade: 2000

In Cloudera Base on premises, fewer restrictions on dynamic paritioning occur than in the pre-upgrade CDH or HDP cluster.

hive.exec.post.hooks

Before upgrade:

```
com.cloudera.navigator.audit.hive.HiveExecHookContext,org.apache
.hadoop.hive.ql.hooks.LineageLogger
```

After upgrade: org.apache.hadoop.hive.ql.hooks.HiveProtoLoggingHook

A prime number is recommended.

hive.exec.reducers.max

Before upgrade: 1099

After upgrade: 1009

Exception: Preserves pre-upgrade value if old default is overridden; otherwise, uses new default

hive.execution.engine

Before upgrade: mr

After upgrade: tez

Tez is now the only supported execution engine, existing queries that change execution mode to Spark or MapReduce within a session, for example, fail.

hive.fetch.task.conversion

Before upgrade: minimal

After upgrade: more

hive.fetch.task.conversion.threshold

Before upgrade: 256MB

After upgrade: 1GB

Exception: Preserves pre-upgrade value if old default is overridden; otherwise, uses new default.

hive.hashtable.key.count.adjustment

Before upgrade: 1

After upgrade: 0.99

Exception: Preserves pre-upgrade value if old default is overridden; otherwise, uses new default.

hive.limit.optimize.enable

Before upgrade: FALSE

After upgrade: TRUE

hive.limit.pushdown.memory.usage

Before upgrade: 0.1

After upgrade: 0.04

Exception: Preserves pre-upgrade value if old default is overridden; otherwise, uses new default.

hive.mapjoin.hybridgrace.hashtable

Before upgrade: TRUE

After upgrade: FALSE

hive.mapred.reduce.tasks.speculative.execution

Before upgrade: TRUE

After upgrade: FALSE

hive.metastore.aggregate.stats.cache.enabled

Before upgrade: TRUE

After upgrade: FALSE

hive.metastore.disallow.incompatible.col.type.changes

Before upgrade: FALSE

After upgrade: TRUE

Schema evolution is more restrictive in Cloudera Base on premises than in CDH to avoid data corruption. The new default disallows column type changes if the old and new types are incompatible.

hive.metastore.dml.events

Before upgrade: FALSE

After upgrade: TRUE

hive.metastore.event.message.factory

Before upgrade: org.apache.hadoop.hive.metastore.messaging.json.ExtendedJSONMessageFactory

After upgrade: org.apache.hadoop.hive.metastore.messaging.json.gzip.GzipJSONMessageEncoder

hive.metastore.uri.selection

Before upgrade: SEQUENTIAL

After upgrade: RANDOM

hive.metastore.warehouse.dir

Before upgrade from CDH: /user/hive/warehouse

Before upgrade from HDP: /apps/hive/warehouse

After upgrade from CDH: /warehouse/tablespace/managed/hive

After upgrade from HDP: /warehouse/tablespace/managed/hive

For information about the location of old tables and new tables, which you create after the upgrade, see Changes to CDH Hive Tables or Changes to HDP Hive tables.

hive.optimize.metadataonly

Before upgrade: FALSE

After upgrade: TRUE

hive.optimize.point.lookup.min

Before upgrade: 31

After upgrade: 2

hive.prewarm.numcontainers

Before upgrade: 10

After upgrade: 3

hive.script.operator.env.blacklist

Before upgrade: hive.txn.valid.txns,hive.script.operator.env.blacklist

After upgrade: hive.txn.valid.txns,hive.txn.tables.valid.writeids,hive.txn.valid.writeids,hive.script.operator.env.blacklist

hive.security.authorization.sqlstd.confwhitelist

Before upgrade:

```
hive\.auto\..*hive\.cbo\..*hive\.convert\..*hive\.exec\.dynamic\
.partition.*hive\.exec\..*\.dynamic\.partitions\..*hive\.exec\.c
ompress\..*hive\.exec\.infer\..*hive\.exec\.mode.local\..*hive\.
exec\.orc\..*hive\.exec\.parallel.*hive\.explain\..*hive\.fetch.
task\..*hive\.groupby\..*hive\.hbase\..*hive\.index\..*hive\.ind
ex\..*hive\.intermediate\..*hive\.join\..*hive\.limit\..*hive\.l
og\..*hive\.mapjoin\..*hive\.merge\..*hive\.optimize\..*hive\.or
c\..*hive\.outerjoin\..*hive\.parquet\..*hive\.ppd\..*hive\.prew
arm\..*hive\.server2\.proxy\.userhive\.skewjoin\..*hive\.smbjoin
\..*hive\.stats\..*hive\.strict\..*hive\.tez\..*hive\.vectorized
\..*mapred\.map\..*mapred\.reduce\..*mapred\.output\.compression
\.codecmapred\.job\.queuenamemapred\.output\.compression\.typema
pred\.min\.split\.sizemapreduce\.job\.reduce\.slowstart\.complet
edmapsmapreduce\.job\.queuenamemapreduce\.job\.tagsmapreduce\.in
put\.fileinputformat\.split\.minsizemapreduce\.map\..*mapreduce\
.reduce\..*mapreduce\.output\.fileoutputformat\.compress\.codecm
apreduce\.output\.fileoutputformat\.compress\.typeoozie\..*tez\.
am\..*tez\.task\..*tez\.runtime\..*tez\.queue\.namehive\.transpo
se\.aggr\.joinhive\.exec\.reducers\.bytes\.per\.reducerhive\.cli
ent\.stats\.countershive\.exec\.default\.partition\.namehive\.ex
ec\.drop\.ignorenonexistenthive\.counters\.group\.namehive\.defa
ult\.fileformat\.managedhive\.enforce\.bucketmapjoinhive\.enforc
e\.sortmergebucketmapjoinhive\.cache\.expr\.evaluationhive\.quer
y\.result\.fileformathive\.hashtable\.loadfactorhive\.hashtable\
.initialCapacityhive\.ignore\.mapjoin\.hinthive\.limit\.row\.max
\.sizehive\.mapred\.modehive\.map\.aggrhive\.compute\.query\.usi
ng\.statshive\.exec\.rowoffsethive\.variable\.substitutehive\.va
riable\.substitute\.depthhive\.autogen\.columnalias\.prefix\.inc
ludefuncnamehive\.autogen\.columnalias\.prefix\.labelhive\.exec\
.check\.crossproductshive\.cli\.tez\.session\.asynchive\.compath
ive\.exec\.concatenate\.check\.indexhive\.display\.partition\.co
ls\.separatelyhive\.error\.on\.empty\.partitionhive\.execution\.
```

```
enginehive\.exec\.copyfile\.maxsizehive\.exim\.uri\.scheme\.whit
elisthive\.file\.max\.footerhive\.insert\.into\.multilevel\.dirs
hive\.localize\.resource\.num\.wait\.attemptshive\.multi\.insert
\.move\.tasks\.share\.dependencieshive\.support\.quoted\.identif
iershive\.resultset\.use\.unique\.column\.nameshive\.analyze\.st
mt\.collect\.partlevel\.statshive\.exec\.schema\.evolutionhive\.
server2\.logging\.operation\.levelhive\.server2\.thrift\.results
et\.serialize\.in\.taskshive\.support\.special\.characters\.tabl
enamehive\.exec\.job\.debug\.capture\.stacktraceshive\.exec\.job
\.debug\.timeouthive\.llap\.io\.enabledhive\.llap\.io\.use\.file
id\.pathhive\.llap\.daemon\.service\.hostshive\.llap\.execution\
.modehive\.llap\.auto\.allow\.uberhive\.llap\.auto\.enforce\.tre
ehive\.llap\.auto\.enforce\.vectorizedhive\.llap\.auto\.enforce\
.statshive\.llap\.auto\.max\.input\.sizehive\.llap\.auto\.max\.o
utput\.sizehive\.llap\.skip\.compile\.udf\.checkhive\.llap\.clie
nt\.consistent\.splitshive\.llap\.enable\.grace\.join\.in\.llaph
ive\.llap\.allow\.permanent\.fnshive\.exec\.max\.created\.filesh
ive\.exec\.reducers\.maxhive\.reorder\.nway\.joinshive\.output\.
file\.extensionhive\.exec\.show\.job\.failure\.debug\.infohive\.
exec\.tasklog\.debug\.timeouthive\.query\.id
```

After upgrade:

```
hive\.auto\..*hive\.cbo\..*hive\.convert\..*hive\.druid\..*hive\
.exec\.dynamic\.partition.*hive\.exec\.max\.dynamic\.partitions.
*hive\.exec\.compress\..*hive\.exec\.infer\..*hive\.exec\.mode.l
ocal\..*hive\.exec\.orc\..*hive\.exec\.parallel.*hive\.exec\.que
ry\.redactor\..*hive\.explain\..*hive\.fetch.task\..*hive\.group
by\..*hive\.hbase\..*hive\.index\..*hive\.index\..*hive\.interme
diate\..*hive\.jdbc\..*hive\.join\..*hive\.limit\..*hive\.log\..
*hive\.mapjoin\..*hive\.merge\..*hive\.optimize\..*hive\.materia
lizedview\..*hive\.orc\..*hive\.outerjoin\..*hive\.parquet\..*hi
ve\.ppd\..*hive\.prewarm\..*hive\.query\.redaction\..*hive\.serv
er2\.thrift\.resultset\.default\.fetch\.sizehive\.server2\.proxy
\.userhive\.skewjoin\..*hive\.smbjoin\..*hive\.stats\..*hive\.st
rict\..*hive\.tez\..*hive\.vectorized\..*hive\.query\.reexecutio
n\..*reexec\.overlay\..*fs\.defaultFSssl\.client\.truststore\.lo
cationdistcp\.atomicdistcp\.ignore\.failuresdistcp\.preserve\.st
atusdistcp\.preserve\.rawxattrsdistcp\.sync\.foldersdistcp\.dele
te\.missing\.sourcedistcp\.keystore\.resourcedistcp\.liststatus\
.threadsdistcp\.max\.mapsdistcp\.copy\.strategydistcp\.skip\.crc
distcp\.copy\.overwritedistcp\.copy\.appenddistcp\.map\.bandwidt
h\.mbdistcp\.dynamic\..*distcp\.meta\.folderdistcp\.copy\.listin
g\.classdistcp\.filters\.classdistcp\.options\.skipcrccheckdistc
p\.options\.mdistcp\.options\.numListstatusThreadsdistcp\.option
s\.mapredSslConfdistcp\.options\.bandwidthdistcp\.options\.overw
ritedistcp\.options\.strategydistcp\.options\.idistcp\.options\.
p.*distcp\.options\.updatedistcp\.options\.deletemapred\.map\..*
mapred\.reduce\..*mapred\.output\.compression\.codecmapred\.job\
.queue\.namemapred\.output\.compression\.typemapred\.min\.split\
.sizemapreduce\.job\.reduce\.slowstart\.completedmapsmapreduce\.
job\.queuenamemapreduce\.job\.tagsmapreduce\.input\.fileinputfor
mat\.split\.minsizemapreduce\.map\..*mapreduce\.reduce\..*mapred
uce\.output\.fileoutputformat\.compress\.codecmapreduce\.output\
.fileoutputformat\.compress\.typeoozie\..*tez\.am\..*tez\.task\.
.*tez\.runtime\..*tez\.queue\.namehive\.transpose\.aggr\.joinhiv
e\.exec\.reducers\.bytes\.per\.reducerhive\.client\.stats\.count
ershive\.exec\.default\.partition\.namehive\.exec\.drop\.ignoren
onexistenthive\.counters\.group\.namehive\.default\.fileformat\.
managedhive\.enforce\.bucketmapjoinhive\.enforce\.sortmergebucke
tmapjoinhive\.cache\.expr\.evaluationhive\.query\.result\.filefo
rmathive\.hashtable\.loadfactorhive\.hashtable\.initialCapacityh
ive\.ignore\.mapjoin\.hinthive\.limit\.row\.max\.sizehive\.mapre
```

```
d\.modehive\.map\.aggrhive\.compute\.query\.using\.statshive\.ex
ec\.rowoffsethive\.variable\.substitutehive\.variable\.substitut
e\.depthhive\.autogen\.columnalias\.prefix\.includefuncnamehive\
.autogen\.columnalias\.prefix\.labelhive\.exec\.check\.crossprod
uctshive\.cli\.tez\.session\.asynchive\.compathive\.display\.par
tition\.cols\.separatelyhive\.error\.on\.empty\.partitionhive\.e
xecution\.enginehive\.exec\.copyfile\.maxsizehive\.exim\.uri\.sc
heme\.whitelisthive\.file\.max\.footerhive\.insert\.into\.multil
evel\.dirshive\.localize\.resource\.num\.wait\.attemptshive\.mul
ti\.insert\.move\.tasks\.share\.dependencieshive\.query\.results
\.cache\.enabledhive\.query\.results\.cache\.wait\.for\.pending\
.resultshive\.support\.quoted\.identifiershive\.resultset\.use\.
unique\.column\.nameshive\.analyze\.stmt\.collect\.partlevel\.st
atshive\.exec\.schema\.evolutionhive\.server2\.logging\.operatio
n\.levelhive\.server2\.thrift\.resultset\.serialize\.in\.taskshi
ve\.support\.special\.characters\.tablenamehive\.exec\.job\.debu
g\.capture\.stacktraceshive\.exec\.job\.debug\.timeouthive\.llap
\.io\.enabledhive\.llap\.io\.use\.fileid\.pathhive\.llap\.daemon
\.service\.hostshive\.llap\.execution\.modehive\.llap\.auto\.all
ow\.uberhive\.llap\.auto\.enforce\.treehive\.llap\.auto\.enforce
\.vectorizedhive\.llap\.auto\.enforce\.statshive\.llap\.auto\.ma
x\.input\.sizehive\.llap\.auto\.max\.output\.sizehive\.llap\.ski
p\.compile\.udf\.checkhive\.llap\.client\.consistent\.splitshive
\.llap\.enable\.grace\.join\.in\.llaphive\.llap\.allow\.permanen
t\.fnshive\.exec\.max\.created\.fileshive\.exec\.reducers\.maxhi
ve\.reorder\.nway\.joinshive\.output\.file\.extensionhive\.exec\
.show\.job\.failure\.debug\.infohive\.exec\.tasklog\.debug\.time
outhive\.query\.idhive\.query\.tag
```

hive.security.command.whitelist

Before upgrade: set,reset,dfs,add,list,delete,reload,compile

After upgrade: set,reset,dfs,add,list,delete,reload,compile,llap

hive.server2.enable.doAs

Before upgrade: TRUE (in case of an insecure cluster only)

After upgrade: FALSE (in all cases)

Affects only insecure clusters by turning off impersonation. Permission issues are expected to arise for affected clusters.

hive.server2.idle.session.timeout

Before upgrade: 12 hours

After upgrade: 24 hours

Exception:Preserves pre-upgrade value if old default is overridden; otherwise, uses new default.

hive.server2.max.start.attempts

Before upgrade: 30

After upgrade: 5

hive.server2.parallel.ops.in.session

Before upgrade: TRUE

After upgrade: FALSE

A Tez limitation requires disabling this property; otherwise, queries submitted concurrently on a single JDBC connection fail or execute slower.

hive.server2.support.dynamic.service.discovery

Before upgrade: FALSE

After upgrade: TRUE

hive.server2.tez.initialize.default.sessions

Before upgrade: FALSE

After upgrade: TRUE

hive.server2.thrift.max.worker.threads

Before upgrade: 100

After upgrade: 500

Exception: Preserves pre-upgrade value if the old default is overridden; otherwise, uses new default.

hive.server2.thrift.resultset.max.fetch.size

Before upgrade: 1000

After upgrade: 10000

hive.service.metrics.file.location

Before upgrade: /var/log/hive/metrics-hiveserver2/metrics.log

After upgrade: /var/log/hive/metrics-hiveserver2-hiveontez/metrics.log

This location change is due to a service name change.

hive.stats.column.autogather

Before upgrade: FALSE

After upgrade: TRUE

hive.stats.deserialization.factor

Before upgrade: 1

After upgrade: 10

hive.support.special.characters.tablename

Before upgrade: FALSE

After upgrade: TRUE

hive.tez.auto.reducer.parallelism

Before upgrade: FALSE

After upgrade: TRUE

hive.tez.bucket.pruning

Before upgrade: FALSE

After upgrade: TRUE

hive.tez.container.size

Before upgrade: -1

After upgrade: 4096

hive.tez.exec.print.summary

Before upgrade: FALSE

After upgrade: TRUE

hive.txn.manager

Before upgrade: org.apache.hadoop.hive.ql.lockmgr.DummyTxnManager

After upgrade: org.apache.hadoop.hive.ql.lockmgr.DbTxnManager

hive.vectorized.execution.mapjoin.minmax.enabled

Before upgrade: FALSE

After upgrade: TRUE

hive.vectorized.execution.mapjoin.native.fast.hashtable.enabled

Before upgrade: FALSE

After upgrade: TRUE

hive.vectorized.use.row.serde.deserialize

Before upgrade: FALSE

After upgrade: TRUE

Related Information

Custom Configuration (about Cloudera Manager Safety Valve)
Customizing critical Hive configurations
Changes to CDH Hive Tables
Changes to HDP Hive Tables

LOCATION and MANAGEDLOCATION clauses

Before upgrading, your Hive version might have supported using the LOCATION clause in queries to create either managed or external tables or databases for managed and external tables. After upgrading, Hive stores managed and external tables in separate HDFS locations. CREATE TABLE limits the use of the LOCATION clause, and consequently requires a change to your queries. Hive in Cloudera also supports a new location-related clause.

External table limitation for creating table locations

Hive assigns a default location in the warehouse for external tables—/warehouse/tablespace/external/hive. In Cloudera, Hive does not allow the LOCATION clause in queries to create a managed table. Using this clause, you can specify a location only when creating external tables. For example:

```
CREATE EXTERNAL TABLE my_external_table (a string, b string)
ROW FORMAT SERDE 'com.mytables.MySerDe'
WITH SERDEPROPERTIES ( "input.regex" = "*.csv")
LOCATION '/warehouse/tablespace/external/hive/marketing';
```

Table MANAGEDLOCATION clause

In Cloudera, Hive has been enhanced to include a MANAGEDLOCATION clause to specify the location of managed tables as shown in the following syntax:

```
CREATE (DATABASE|SCHEMA) [IF NOT EXISTS] database_name
  [COMMENT database_comment]
  [LOCATION external_table_path]
  [MANAGEDLOCATION managed_table_directory_path]
  [WITH DBPROPERTIES (property_name=property_value, ...)];
```

Hive assigns a default location in the warehouse for managed tables—/warehouse/tablespace/managed/hive. In the MANAGEDLOCATION clause, you specify a top level directory for managed tables when creating a Hive database.

Important: Do not use the LOCATION clause to specify the location of managed tables. This clause is only used to specify the location of external tables. Use the MANAGEDLOCATION clause if you are creating managed tables. You must also ensure that you do not set LOCATION and MANAGEDLOCATION to the same HDFS path.

Use DESCRIBE DATABASE db_name; to view the root location of the database on the filesystem.

Related Information

Create a default directory for managed tables

Handling table reference syntax

For ANSI SQL compliance, Hive 3.x rejects `db.table` in SQL queries as described by the Hive-16907 bug fix. A dot (.) is not allowed in table names. As a Data Engineer, you need to ensure that Hive tables do not contain these references before migrating the tables to Cloudera, that scripts are changed to comply with the SQL standard references, and that users are aware of the requirement.

About this task

To change queries that use such `db.table` references thereby preventing Hive from interpreting the entire db.table string incorrectly as the table name, you enclose the database name and the table name in backticks as follows:

A dot (.) is not allowed in table names.

Procedure

1. Find a table having the problematic table reference.
   For example, math.students appears in a CREATE TABLE statement.
2. Enclose the database name and the table name in backticks.

```
CREATE TABLE `math`.`students` (name VARCHAR(64), age INT, gpa DECIMAL(3
,2));
```

Related Information

Add Backticks to Table References

Add Backticks to Table References

CDP Private Cloud Base includes the Hive-16907 bug fix, which rejects `db.table` in SQL queries. A dot (.) is not allowed in table names. You need to change queries that use such references to prevent Hive from interpreting the entire db.table string as the table name.

Procedure

1. Find a table having the problematic table reference.

```
math.students
```

   appears in a CREATE TABLE statement.
2. Enclose the database name and the table name in backticks.

```
 CREATE TABLE `math`.`students` (name VARCHAR(64), age INT, gpa DECIMAL
(3,2));
```

Related Information

Handling table reference syntax

Key semantic changes and workarounds

As SQL Developer, Analyst, or other Hive user, you need to know potential problems with queries due to semantic changes. Some of the operations that changed were not widely used, so you might not encounter any of the problems associated with the changes.

Over the years, Apache Hive committers enhanced versions of Hive supported in legacy releases of CDH and HDP, with users in mind. Changes were designed to maintain compatibility with Hive applications. Consequently, few syntax changes occurred over the years. A number of semantic changes, described in this section did occur, however. Workarounds are described for these semantic changes.

Casting timestamps

Results of applications that cast numerics to timestamps differ from Hive 2 to Hive 3. Apache Hive changed the behavior of CAST to comply with the SQL Standard, which does not associate a time zone with the TIMESTAMP type.

Before Upgrade to Cloudera

Casting a numeric type value into a timestamp could be used to produce a result that reflected the time zone of the cluster. For example, 1597217764557 is 2020-08-12 00:36:04 PDT. Running the following query casts the numeric to a timestamp in PDT:

```
> SELECT CAST(1597217764557 AS TIMESTAMP);
| 2020-08-12 00:36:04 |
```

After Upgrade to Cloudera

Casting a numeric type value into a timestamp produces a result that reflects the UTC instead of the time zone of the cluster. Running the following query casts the numeric to a timestamp in UTC.

```
> SELECT CAST(1597217764557 AS TIMESTAMP);
| 2020-08-12 07:36:04.557  |
```

Action Required

Change applications. Do not cast from a numeral to obtain a local time zone. Built-in functions from_utc_timestamp and to_utc_timestamp can be used to mimic behavior before the upgrade.

Related Information

Apache Hive web site summary of timestamp semantics

Casting invalid dates

Casting of an invalid date differs from Hive 1 in CDH 5 to Hive 3 in CDP. Hive 3 uses a different parser formatter from the one used in Hive 1, which affects semantics. Hive 1 considers 00 invalid for date fields. Hive 3 considers 00 valid for date fields. Neither Hive 1 nor Hive 3 correctly handles invalid dates, and Hive-25056 addresses this issue.

Before Upgrade to CDP

Casting of invalid date (zero value in one or more of the 3 fields of date, month, year) returns a NULL value:

```
SELECT CAST ('0000-00-00' as date) , CAST ('000-00-00 00:00:00' AS TIMESTAMP
) ;
```

After Upgrade to CDP

Casting of an invalid date returns a result.

```
> SELECT CAST ('0000-00-00' as date) , CAST ('000-00-00 00:00:00' AS TIMESTA
MP) ;
...
00002-11-30 00:00:00.0
```

Action Required

Do not cast invalid dates in Hive 3.

Changing incompatible column types

A default configuration change can cause applications that change column types to fail.

Before Upgrade to Cloudera

In HDP 2.x and CDH 5.x and CDH 6 hive.metastore.disallow.incompatible.col.type.changes is false by default to allow changes to incompatible column types. For example, you can change a STRING column to a column of an incompatible type, such as MAP<STRING, STRING>. No error occurs.

After Upgrade to Cloudera

In Cloudera, hive.metastore.disallow.incompatible.col.type.changes is true by default. Hive prevents changes to incompatible column types. Compatible column type changes, such as INT, STRING, BIGINT, are not blocked.

Action Required

Change applications to disallow incompatible column type changes to prevent possible data corruption. Check ALTER TABLE statements and change those that would fail due to incompatible column types.

Related Information

HIVE-12320

Understanding CREATE TABLE behavior

Hive table creation has changed significantly since Hive 3 to improve useability and functionality. If you are upgrading from CDH or HDP, you must understand the changes affecting legacy table creation behavior.

Hive has changed table creation in the following ways:

- Creates ACID-compliant table, which is the default in Cloudera
- Supports simple writes and inserts
- Writes to multiple partitions
- Inserts multiple data updates in a single SELECT statement
- Eliminates the need for bucketing.

If you have an ETL pipeline that creates tables in Hive, the tables will be created as ACID. Hive now tightly controls access and performs compaction periodically on the tables. Using ACID-compliant, transactional tables causes no performance or operational overload. The way you access managed Hive tables from Spark and other clients changes. In Cloudera, access to external tables requires you to set up security access permissions.

You must understand the behavior of the CREATE TABLE statement in legacy platforms like CDH or HDP and how the behavior changes after you upgrade to Cloudera.

Before upgrading to CDP Private Cloud Base

In CDH 5, CDH 6, and HDP 2, by default CREATE TABLE creates a non-ACID managed table in plain text format.

In HDP 3 and CDP 7.1.0 through 7.1.7.x, by default CREATE TABLE creates either a full ACID transactional table in ORC format or insert-only ACID transactional tables for all other table formats.

After upgrading to CDP Private Cloud Base

- If you are upgrading from HDP 2, CDH 5, or CDH 6 to CDP 7.1.0 through CDP 7.1.8, by default CREATE TABLE creates a full ACID transactional table in ORC format or insert-only ACID transactional tables for all other table formats.
- If you are upgrading from HDP 3 or CDP 7.1.0 through 7.1.7.x to CDP 7.1.8, the existing behavior persists and CREATE TABLE creates either a full ACID transactional table in ORC format or insert-only ACID transactional tables for all other table formats.

Now that you understand the behavior of the CREATE TABLE statement, you can choose to modify the default table behavior by configuring certain properties. The order of preference for configuration is as follows:

Modify the default CREATE TABLE behavior

Override default behavior when creating the table

Irrespective of the database, session, or site-level settings, you can override the default table behavior by using the MANAGED or EXTERNAL keyword in the CREATE TABLE statement.

```
CREATE [MANAGED][EXTERNAL] TABLE foo (id INT);
```

Set the default table type at a database level

You can use the database property, defaultTableType=EXTERNAL or ACID to specify the default table type to be created using the CREATE TABLE statement. You can specify this property when creating the database or at a later point using the ALTER DATABASE statement. For example:

```
CREATE DATABASE test_db WITH DBPROPERTIES ('defaultTableType'='E
XTERNAL');
```

In this example, tables created under the test_db database using the CREATE TABLE statement creates external tables with the purge fucntionality enabled (external.table.purge = 'true').

You can also choose to configure a database to allow only external tables to be created and prevent creation of ACID tables. While creating a database, you can set the database property, EXTER NAL_TABLES_ONLY=true to ensure that only external tables are created in the database. For example:

```
CREATE DATABASE test_db WITH DBPROPERTIES ('EXTERNAL_TABLES_ONLY
'='true');
```

Set the default table type at a session level

You can configure the CREATE TABLE behavior within an existing beeline session by setting hive.create.as.external.legacy to true or false. Setting the value to true results in configuring the CREATE TABLE statement to create external tables by default.

When the session ends, the default CREATE TABLE behavior also ends.

Set the default table type at a site level

You can configure the CREATE TABLE behavior at the site level by configuring the hive.create.as.insert.only and hive.create.as.acid properties in Cloudera Manager under Hive configuration. When configured at the site level, the behavior persists from session to session. For more information, see Configuring CREATE TABLE behavior.

If you are a Spark user, switching to legacy behavior is unnecessary. Calling ‘create table’ from SparkSQL, for example, creates an external table after upgrading to Cloudera as it did before the upgrade. You can connect to Hive using the Hive Warehouse Connector (HWC) to read Hive ACID tables from Spark. To write ACID tables to Hive from Spark, you use the HWC and HWC API. Spark creates an external table with the purge property when you do not use the HWC API. For more information, see Hive Warehouse Connector for accessing Spark data.

Related Information

Hive Warehouse Connector for accessing Apache Spark data
Spark Direct Reader for accessing Spark data
HDFS ACLS
Apache Hive 3 Architectural Overview
Configure a Resource-based Policy: Hive
Apache Hive 3 Key Features
Apache Hive 3 Tables
Configuring legacy CREATE TABLE behavior

Configuring legacy CREATE TABLE behavior

After you upgrade to CDP Private Cloud Base and migrate old tables, the legacy CREATE TABLE behavior of Hive is no longer available by default and you might want to switch to the legacy behavior. Legacy behavior might solve compatibility problems with your scripts during data migration, for example, when running ETL.

About this task

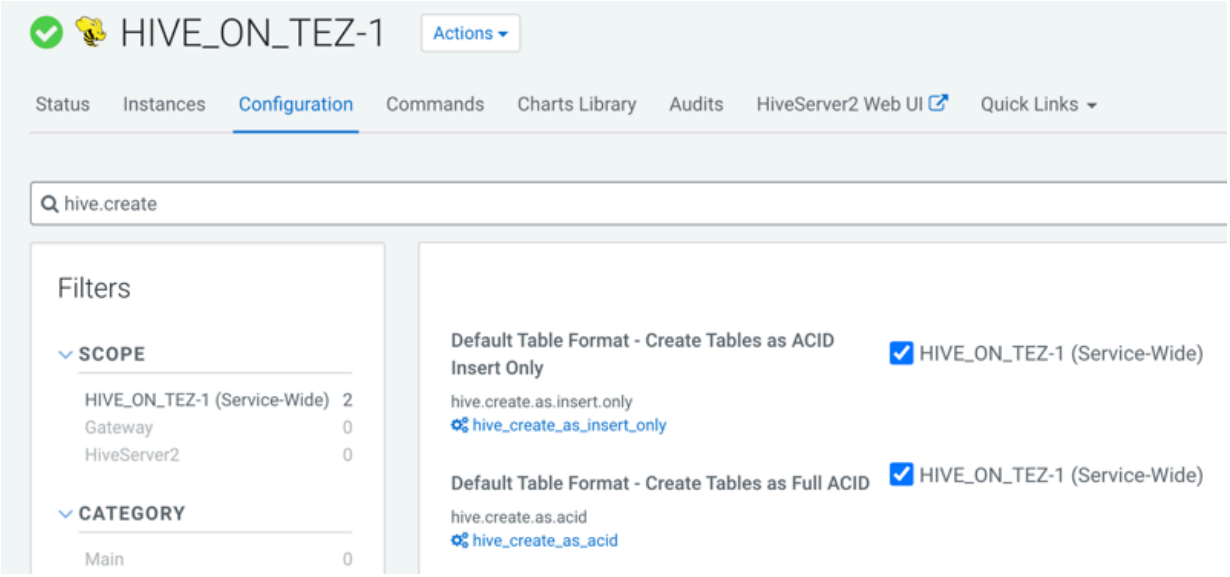
In Cloudera, running a CREATE TABLE statement by default creates a full ACID table for ORC file format and insert-only ACID table for other file formats. You can change the default behavior to use the legacy CREATE TABLE behavior. When you configure legacy behavior, CREATE TABLE creates external tables with the purge

functionality enabled (external.table.purge = 'true'). Therefore, when the table is dropped, data is also deleted from the file system.

You can configure legacy CREATE TABLE behavior at the site level by configuring properties in Cloudera Manager. When configured at the site level, the behavior persists from session to session.

Procedure

1. In Cloudera Manager, click Clusters and select the Hive on Tez service.
2. From the Hive on Tez service, go to the Configuration tab and search for hive.create.



3. If the following properties are selected, clear the selection to enable legacy CREATE TABLE behavior.
   - Default Table Format - Create Tables as ACID Insert Only (hive.create.as.insert.only)
   - Default Table Format - Create Tables as Full ACID (hive.create.as.acid)

Results

Legacy behavior is enabled and the CREATE TABLE statement now creates external tables with the external.table.purge table property set to true.

Handling the Keyword APPLICATION

If you use the keyword APPLICATION in your queries, you might need to modify the queries to prevent failure.

To prevent a query that uses a keyword from failing, enclose the query in backticks.

Before Upgrade to CDP Private Cloud Base

In CDH releases, such as CDH 5.13, queries that use the word APPLICATION in queries execute successfully. For example, you could use this word as a table name.

```
> select f1, f2 from application
```

After Upgrade to Cloudera Base on premises

A query that uses the keyword APPLICATION fails.

Action Required

Change applications. Enclose queries in backticks. SELECT field1, field2 FROM `application`;

Handling output of greatest and least functions

To calculate the greatest (or least) value in a column, you need to work around a problem that occurs when the column has a NULL value.

Before Upgrade to Cloudera

The greatest function returned the highest value of the list of values. The least function returned the lowest value of the list of values.

After Upgrade to Cloudera

Returns NULL when one or more arguments are NULL.

Action Required

Use NULL filters or the nvl function on the columns you use as arguments to the greatest or least functions.

```
SELECT greatest(nvl(col1,default value incase of NULL),nvl(col2,default valu
e incase of NULL));
```

TRUNCATE TABLE on an external table

Hive 3 does not support TRUNCATE TABLE on external tables. Truncating an external table results in an error. You can truncate an external table if you change your applications to set a table property to purge data.

Before Upgrade to Cloudera

Some legacy versions of Hive supported TRUNCATE TABLE on external tables.

After Upgrade to CDP Private Cloud Base

By default, TRUNCATE TABLE is supported only on managed tables. Attempting to truncate an external table results in the following error:

```
Error: org.apache.spark.sql.AnalysisException: Operation not allowed: TRUNCA
TE TABLE on external tables
```

Action Required

Change applications. Do not attempt to run TRUNCATE TABLE on an external table.

Alternatively, change applications to alter a table property to set external.table.purge to true to allow truncation of an external table:

```
ALTER TABLE mytable SET TBLPROPERTIES ('external.table.purge'='true');
```

Hive unsupported interfaces and features

You need to understand the interfaces that are not supported.

Unsupported Interfaces and features

The following interfaces are not supported in Cloudera Base on premises:

- Druid
- Hcat CLI (however HCatalog is supported)
- Hive CLI (replaced by Beeline)
- Hive View UI feature in Ambari
- Apache Hive Standalone driver
- Renaming Hive databases
- Multiple insert overwrite queries that read data from a source table.
- LLAP
- MapReduce execution engine (replaced by Tez)

- Pig
- S3 for storing tables (available in Cloudera on cloud only)
- Spark execution engine (replaced by Tez)
- Spark thrift server

  Spark and Hive tables interoperate using the Hive Warehouse Connector.
- SQL Standard Authorization
- Storage Based Authorization
- Tez View UI feature in Ambari
- WebHCat

You can use Hue in lieu of Hive View.

Storage Based Authorization

Storage Based Authorization (SBA) is no longer supported in Cloudera. Ranger integration with Hive metastore provides consistency in Ranger authorization enabled in HiveServer (HS2). SBA did not provide authorization support for metadata that does not have a file/directory associated with it. Ranger-based authorization has no such limitation.

Hive-Kudu integration

Cloudera does not support the integration of HiveServer (HS2) with Kudu tables. You cannot run queries against Kudu tables from HS2.

Partially unsupported interfaces

Apache Hadoop Distributed Copy (DistCP) is not supported for copying Hive ACID tables.

Unsupported Features

Cloudera does not support the following features that were available in HDP and CDH platforms:

- CREATE TABLE that specifies a managed table location

  Do not use the LOCATION clause to create a managed table. Hive assigns a default location in the warehouse to managed tables. That default location is configured in Hive using the hive.metastore.warehouse.dir configuration property, but can be overridden for the database by setting the CREATE DATABASE MANAGEDLOCATION parameter.
- CREATE INDEX and related index commands were removed in Hive 3, and consequently are not supported in Cloudera.

  In Cloudera, you use the Hive 3 default ORC columnar file formats to achieve the performance benefits of indexing. Materialized Views with automatic query rewriting also improves performance. Indexes migrated to Cloudera are preserved but render any Hive tables with an undroppable index. To drop the index, google the Known Issue for CDPD-23041.
- Hive metastore (HMS) high availablility (HA) load balancing in CDH

  You need to set up HMS HA as described in the documentation.
- Local or Embedded Hive metastore server

  Cloudera does not support the use of a local or embedded Hive metastore setup.

Unsupported Connector Use

Cloudera does not support the Sqoop exports using the Hadoop jar command (the Java API) that Teradata documents. For more information, see Migrating data using Sqoop.

Related Information

Configuring HMS for high availability
Hive Warehouse Connector for accessing Apache Spark data
Spark Direct Reader for accessing Spark data

Changes to CDH Hive Tables

As a Data Scientist, Architect, Analyst, or other Hive user you need to locate and use your Apache Hive 3 tables after an upgrade. You also need to understand the changes that occur during the upgrade process. The location of existing tables after a CDH to Cloudera upgrade does not change. Upgrading CDH to CDP Private Cloud Base converts Hive managed tables to external tables in Hive 3.

About this task

When the upgrade process converts a managed table to external, it sets the table property external.table.purge to true. The table is equivalent to a managed table having purge set to true in your old CDH cluster.

Managed tables on the HDFS in /user/hive/warehouse before the upgrade remain there after the conversion to external. Tables that were external before the upgrade are not relocated. You need to set HDFS policies to access external tables in Ranger, or set up HDFS ACLs.

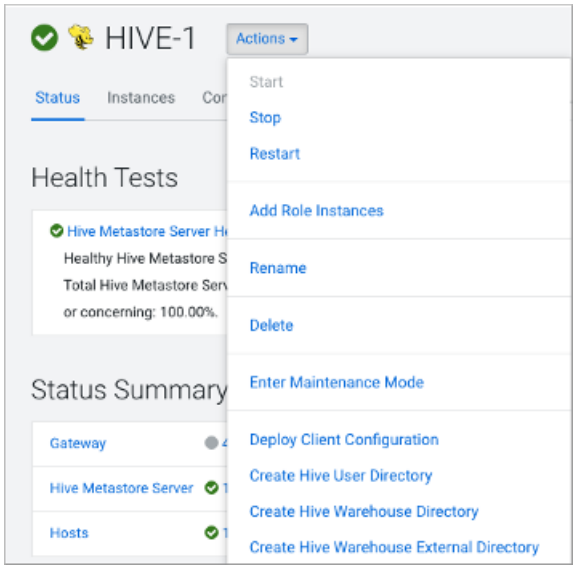
The upgrade process sets the hive.metastore.warehouse.dir property to /warehouse/tablespace/managed/hive, designating it the Hive warehouse location for managed tables. New managed tables that you create in Cloudera are stored in the Hive warehouse. New external tables are stored in the Hive external warehouse /warehouse/tablespace/external/hive.

To change the location of the Hive warehouses, you navigate to one of the following menu items in the first step below.

- Hive Action Menu Create Hive Warehouse Directory
- Hive Action Menu Create Hive Warehouse External Directory

Procedure

1. Set up directories for the Hive warehouse directory and Hive warehouse external directory from Cloudera Manager Actions.



2. In Cloudera Manager, click Clusters Hive (the Hive Metastore service) Configuration , and change the hive.metastore.warehouse.dir property value to the path you specified for the new Hive warehouse directory.
3. Change the hive.metastore.warehouse.external.dir property value to the path you specified for the Hive warehouse external directory.

4. Configure Ranger policies or set up ACL permissions to access the directories.

Related Information

HDFS ACLS

Set ACLs for Impala

Apache Hive Post-Upgrade Tasks

A successful upgrade requires performing a number of procedures that you can follow using step-by-step instructions. Important configuration tasks set up security on your cluster. You learn about semantic changes that might affect your applications, and see how to find your tables or move them. You find out about the Hive Warehouse Connector (HWC) to access files from Spark.

Customizing critical Hive configurations

As Administrator, you need property configuration guidelines. You need to know which properties you need to reconfigure after upgrading. You must understand which the upgrade process carries over from the old cluster to the new cluster.

The Cloudera upgrade process tries to preserve your Hive configuration property overrides. These overrides are the custom values you set to configure Hive in the old CDH or HDP cluster. The upgrade process does not perserve all overrides. For example, a custom value you set for hive.exec.max.dynamic.partitions.pernode is preserved. In the case of other properties, for example hive.cbo.enable, the upgrade ignores any override and just sets the Cloudera-recommended value.

The upgrade process does not preserve overrides to the configuration values of the following properties that you likely need to reconfigure to meet your needs:

- hive.conf.hidden.list
- hive.conf.restricted.list
- hive.exec.post.hooks
- hive.script.operator.env.blacklist
- hive.security.authorization.sqlstd.confwhitelist
- hive.security.command.whitelist

The Apache Hive Wiki describes these properties. The values of these properties are lists.

The upgrade process ignores your old list and sets a new generic list. For example, the hive.security.command.whitel ist value is a list of security commands you consider trustworthy and want to keep. Any overrides of this list that you set in the old cluster are not preserved. The new default is probably a shorter (more restrictive) list than the original default you were using in the old cluster. You need to customize this Cloudera to meet your needs.

Check and change each property listed above after upgrading as described in the next topic.

Consider reconfiguring more property values than the six listed above. Even if you did not override the default value in the old cluster, the Cloudera default might have changed in a way that impacts your work.

Related Information

Hive Configuration Property Changes

Apache Hive Wiki: Configuration Properties

Hive Configuration Requirements and Recommendations

Setting Hive Configuration Overrides

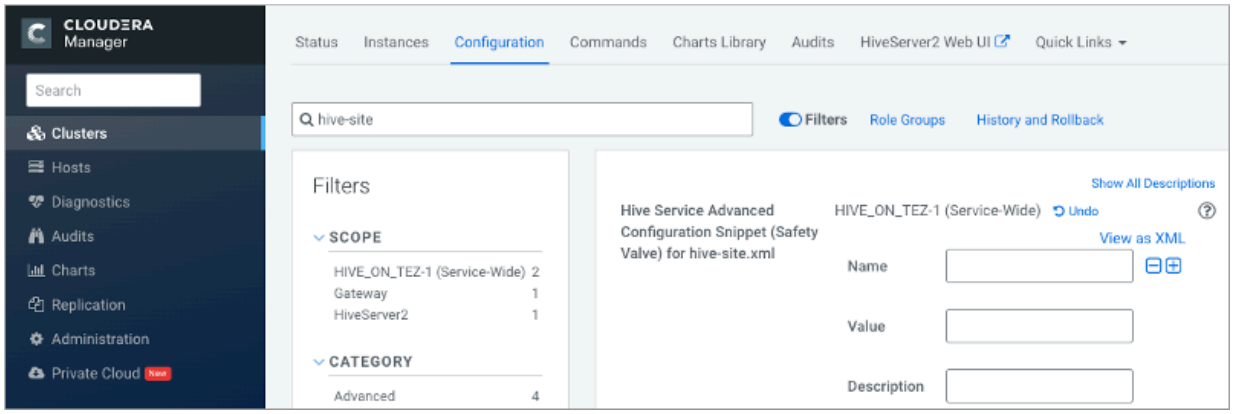
You need to know how to configure the critical customizations that the upgrade process does not preserve from your old Hive cluster. Referring to your records about your old configuration, you follow steps to set at least six critical property values.

About this task

By design, the six critical properties that you need to customize are not visible in Cloudera Manager, as you can see from the Visible in Cloudera Manager column of Configurations Requirements and Recommendations. You use the Safety Valve to add these properties to hive-site.xml as shown in this task.

Procedure

1. In Cloudera Manager Clusters select the Hive on Tez service. Click Configuration, and search for hive-site.xml.
2. In Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml, click +.



3. In Name, add the hive.conf.hidden.list property.
4. In Value, add your custom list.
5. Customize the other critical properties: hive.conf.restricted.list, hive.exec.post.hooks, hive.script.operator.env.blacklist, hive.security.authorization.sqlstd.confwhitelist, hive.security.command.whitelist.

   Use hive.security.authorization.sqlstd.confwhitelist.append, for example, to set up the list.
6. Save the changes and restart the Hive service.
7. Look at the Configurations Requirements and Recommendations to understand which overrides were preserved or not.

Related Information

Hive Configuration Requirements and Recommendations

Hive Configuration Requirements and Recommendations

You need to set certain Hive and HiveServer (HS2) configuration properties after upgrading. You review recommendations for setting up Cloudera Base on premises for your needs, and understand which configurations remain unchanged after upgrading, which impact performance, and default values.

Requirements and Recommendations

The following table includes the Hive service and HiveServer properties that the upgrade process changes. Other property values (not shown) are carried over unchanged from CDH or HDP to Cloudera

- Set After Upgrade column: properties you need to manually configure after the upgrade to Cloudera. Pre-existing customized values are not preserved after the upgrade.
- Default Recommended column: properties that the upgrade process changes to a new value that you are strongly advised to use.
- Impacts Performance column: properties changed by the upgrade process that you set to tune performance.
- Safety Value Overrides column: How the upgrade process handles Safety Valve overrides.
  - Disregards: the upgrade process removes any old CDH Safety Valve configuration snippets from the new CDP configuration.
  - Preserves means the upgrade process carries over any old CDH snippets to the new CDP configuration.
  - Not applicable means the value of the old parameter is preserved.

- Visible in CM column: property is visible in Cloudera Manager after upgrading. Cloudera Manager after upgrading.

  If a property is not visible, and you want to configure it, use the Cloudera Manager Safety Valve to safely add the parameter to the correct file, for example to a cluster-wide, hive-site.xml file.

Table 17:

Property	Set After Upgrade	Default Recomme	Impacts Performa	New Feature	Safety Valve Overrides	Visible in CM
datanucleus.connectionPool.maxPoolSize			#		Preserve	
datanucleus.connectionPoolingType			#		Disregard	
hive.async.log.enabled					Disregard	#
hive.auto.convert.join.noconditionaltask.size					Not applicable	#
hive.auto.convert.sortmerge.join					Preserve	
hive.auto.convert.sortmerge.join.to.mapjoin					Preserve	
hive.cbo.enable					Disregard	#
hive.cbo.show.warnings					Disregard	
hive.compactor.worker.threads				#	Disregard	#
hive.compute.query.using.stats			#		Disregard	#
hive.conf.hidden.list	#				Disregard	
hive.conf.restricted.list	#				Disregard	
hive.default.fileformat.managed					Disregard	#
hive.default.rcfile.serde		#			Preserve	
hive.driver.parallel.compilation					Disregard	#
hive.exec.dynamic.partition.mode					Disregard	
hive.exec.max.dynamic.partitions					Preserve	
hive.exec.max.dynamic.partitions.pernode					Preserve	
hive.exec.post.hooks	#				Disregard	
hive.exec.reducers.max		# or other prime number			Not applicable	#
hive.execution.engine					Disregard	
hive.fetch.task.conversion			#		Not applicable	#
hive.fetch.task.conversion.threshold			#		Not appliable	#
hive.hashtable.key.count.adjustment			#		Preserve	
hive.limit.optimize.enable		#			Disregard	
hive.limit.pushdown.memory.usage			#		Not Applicable	#
hive.mapjoin.hybridgrace.hashtable		#	#		Disregard	
hive.mapred.reduce.tasks.speculative.execution		#			Disregard	
hive.metastore.aggregate.stats.cache.enabled		#	#		Disregard	
hive.metastore.disallow.incompatible.col.type.changes					Disregard	
hive.metastore.dml.events					Disregard	#
hive.metastore.event.message.factory		#			Disregard	

Property	Set After Upgrade	Default Recomme	Impacts Performa	New Feature	Safety Valve Overrides	Visible in CM
hive.metastore.uri.selection		#			Disregard	
hive.metastore.warehouse.dir					Preserve	#
hive.optimize.metadataonly		#			Disregard	
hive.optimize.point.lookup.min					Disregard	
hive.prewarm.numcontainers					Disregard	
hive.script.operator.env.blacklist	#				Disregard	
hive.security.authorization.sqlstd.confwhitelist	#				Disregard	
hive.security.command.whitelist	#				Disregard	
hive.server2.enable.doAs					Disregard	#
hive.server2.idle.session.timeout					Not applicable	#
hive.server2.max.start.attempts					Preserve	
hive.server2.parallel.ops.in.session					Preserve	
hive.server2.support.dynamic.service.discovery				#	Disregard	#
hive.server2.tez.initialize.default.sessions				#	Disregard	
hive.server2.thrift.max.worker.threads					Not Applicable	#
hive.server2.thrift.resultset.max.fetch.size					Preserve	
hive.service.metrics.file.location					Disregard	#
hive.stats.column.autogather		#			Disregard	
hive.stats.deserialization.factor		#			Disregard	
hive.support.special.characters.tablename		#			Disregard	
hive.tez.auto.reducer.parallelism				#	Disregard	#
hive.tez.bucket.pruning				#	Disregard	#
hive.tez.container.size				#	Disregard	#
hive.tez.exec.print.summary				#	Disregard	#
hive.txn.manager				#	Disregard	#
hive.vectorized.execution.mapjoin.minmax.enabled		#			Disregard	
hive.vectorized.execution.mapjoin.native.fast.hashtable.enabled		#			Disregard	
hive.vectorized.use.row.serde.deserialize		#			Disregard	

Related Information

Custom Configuration (about Cloudera Manager Safety Valve)
Customizing critical Hive configurations
Setting Hive Configuration Overrides

Fixing the canary test after upgrading

After upgrading from CDH 6 to CDP, you need to make a change to a table by querying the Hive metastore backend database to run the Hive canary test.

About this task

The Hive canary test for Hive-on-Tez fails unless you make the changes described in the following procedure. When you run the test, you get the following error:

```
org.apache.hadoop.hive.metastore.RetryingHMSHandler: [pool-6-thread-69]: HMS
Handler Fatal error: javax.jdo.JDODataStoreException: Exception thrown flush
ing changes to datastore
 at org.datanucleus.api.jdo.NucleusJDOHelper.getJDOExceptionForNucleusExce
ption(NucleusJDOHelper.java:543)
NestedThrowablesStackTrace:
java.sql.BatchUpdateException: Unknown column 'MESSAGE_FORMAT' in 'field li
st'
```

Procedure

1. Backup the Hive metastore database.
2. Log into the Hive metastore database.
3. Use the metastore database.

   ```
   USE metastore;
   ```

4. Add a MESSAGE_FORMAT column and modify the data type of MESSAGE in the NOTIFICATION_LOG table.

   ```
   ALTER TABLE `NOTIFICATION_LOG` ADD `MESSAGE_FORMAT` varchar(16);

   ALTER TABLE `NOTIFICATION_LOG` MODIFY `MESSAGE` longtext;
   ```

Configuring HiveServer for ETL using YARN queues

You need to set several configuration properties to allow placement of the Hive workload on the Yarn queue manager, which is common for running an ETL job. You need to set several parameters that effectively disable the reuse of containers. Each new query gets new containers routed to the appropriate queue.

About this task

Hive configuration properties affect mapping users and groups to YARN queues. You set these properties to use with YARN Placement Rules.

To set Hive properties for YARN queues:

Procedure

1. In Cloudera Manager, click Clusters Hive-on-Tez Configuration .
2. Search for the Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml setting.
3. In the Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml setting, click +.
4. In Name enter the property hive.server2.tez.initialize.default.sessions and in value enter false.
5. In Name enter the property hive.server2.tez.queue.access.check and in value enter true.
6. In Name enter the property hive.server2.tez.sessions.custom.queue.allowed and in value enter true.

Removing Hive on Spark Configurations

Your scripts, or queries, include the Hive on Spark configuration, which is no longer supported, and you must know how to recognize and remove these configurations.

In Cloudera, there is no Hive-Spark dependency. The Spark site and libs are not in the classpath. This execution engine has been replaced by Apache Tez.

Before Upgrade to Cloudera

CDH supported Hive on Spark and the following configuration to enable Hive on Spark: set hive.execution.engine=spark

After Upgrade to Cloudera

Cloudera does not support Hive on Spark. Scripts that enable Hive on Spark do not work.

Action Required

Remove set hive.execution.engine=spark from your scripts.

Configuring authorization to tables

Although the upgrade process makes no change to the location of external tables, you need to set up access to external tables in HDFS. If you choose the recommended Ranger security model for authorization, you need to set up policies and configure Hive metastore (HMS).

About this task

Set up access to external tables in HDFS using one of the following methods.

- Set up a Hive HDFS policy in Ranger (recommended) to include the paths to external table data.
- Put an HDFS ACL in place. Store the external text file, for example a comma-separated values (CSV) file, in HDFS that will serve as the data source for the external table.

If you want to use Ranger to authorize access to your tables, you must configure a few HMS properties for authorization in addition to setting up Ranger policies. If you have not configured HMS, attempting to create a table using Spark SQL, Beeline, or Hue results in the following error:

```
org.apache.hadoop.hive.ql.ddl.DDLTask. MetaException(message:No privilege 'Create' found for outputs { database:DATABASE_NAME, table:TABLE_NAME})
```

Related Information

HDFS ACLS

Authorizing Apache Hive Access

Configuring HMS properties for authorization

Making the Hive plugin for Ranger visible

After upgrading from HDP or CDH clusters to Cloudera, the Hive plugin for the Hive Metastore and HiveServer2 appears in the Ranger Admin UI unless configuration property problems due to upgrading exist. You can rectify the incorrect properties to fix the problem.

About this task

If the Hive Metastore plugin does not appear in the Ranger Admin UI, you must remove the following property settings from Hive Metastore hive-site.xml safety valve:

- hive.security.authorization.enabled
- hive.security.authorization.manager
- hive.security.metastore.authorization.manager

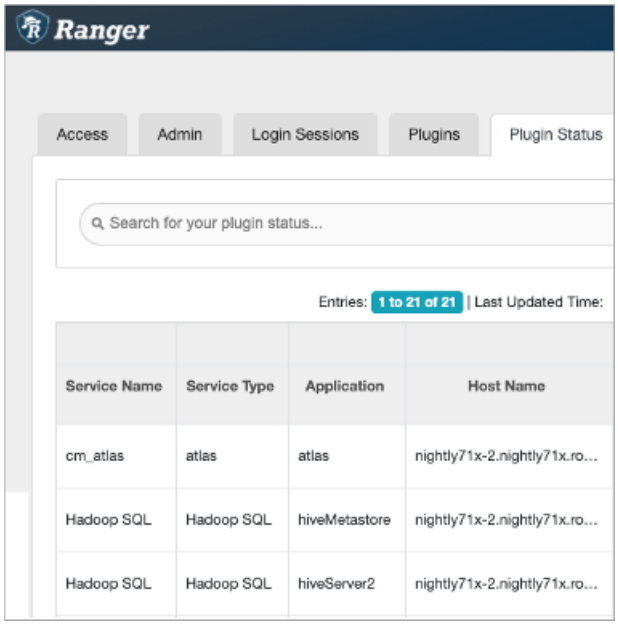
If the HiveServer2 plugin does not appear in the Ranger Admin UI, you must remove the following property settings from HiveServer2 hive-site.xml safety valve:

- hive.security.authorization.enabled
- hive.security.authorization.manager
- hive.security.metastore.authorization.manager
- hive.security.authenticator.manager

After removing these configuration properties, restart the Hive Metastore and HiveServer2 services from Cloudera Manager. Next, you must check whether the Ranger Hive Metastore and HiveServer2 plugins are enabled successfully. To do so:

Procedure

1. From Cloudera Manager, go to Clusters Ranger Ranger Admin Web UI Audit Plugin Status .



   The Hadoop SQL service type for the hiveMetastore and hiveServer2 applications should appear. If so, skip the next step. Your configuration is ok.

2. If, after removing the Hive Metastore and HiveServer2 configuration properties from the respective hive-ste.xml safety valves, the Hive Metastore and HiveServer2 plugins are NOT visible, you must confirm whether or not the following configuration properties appear in hive-site.xml:

   For Hive Metastore, confirm whether or not the following key-value pair appears in hive-site.xml:

   Key: hive.metastore.pre.event.listeners

   Value: org.apache.hadoop.hive.ql.security.authorization.plugin.metastore.HiveMetaStoreAuthorizer

   If this key-value pair does not appear in hive-site.xml, then add it to the Hive Metastore hive-site.xml safety valve.

   For HiveServer2, confirm whether or not the following key value pair appears in hive-site.xml:

   Key: hive.security.authenticator.manager

   Value: org.apache.hadoop.hive.ql.security.SessionStateUserAuthenticator

   If this key-value pair does not appear in hive-site.xml, then add it to the HiveServer2 hive-site.xml safety valve.

Setting up access control lists

Several sources of information about setting up HDFS ACLS plus a brief Ranger overview and pointer to Ranger information prepare you to set up Hive authorization.

In Cloudera Base on premises, HDFS supports POSIX ACLs (Access Control Lists) to assign permissions to users and groups. In lieu of Ranger policies, you use HDFS ACLs to check and make any necessary changes in HDFS permission changes. For more information, see HDFS ACLs, Apache Software Foundation HDFS Permissions Guide, and HDFS ACL Permissions.

In Ranger, you give multiple groups and users specific permissions based on your use case. You apply permissions to a directory tree instead of dealing with individual files. For more information, see Authorizing Apache Hive Access.

If possible, you should use Ranger policies over HDFS ACLs to control HDFS access. Controlling HDFS access through Ranger provides a single, unified interface for understanding and managing your overall governance framework and policy design. If you need to mimic the legacy Sentry HDFS ACL Sync behavior for Hive and Impala tables, consider using Ranger RMS.

Related Information

HDFS ACLS
Apache Hive 3 Architectural Overview
Configure a Resource-based Policy: Hive

Configure encryption zone security

Under certain conditions, you as Administrator, need to perform a security-related task to allow users to access to tables stored in encryption zones. You find out how to prevent access problems to these tables.

About this task

Hive on Tez cannot run some queries on tables stored in encryption zones under certain conditions. Perform the following procedure only when the cluster uses self-signed certificates.

Important: Skip this task for clusters where TLS certificates are properly signed by a Certificate Authority (CA), and the CA is in the truststore files.

Procedure

1. Copy the ssl-client.xml file to a directory that is available on all hosts.
2. In Cloudera Manager, click Clusters Hive on Tez Configuration .
3. Search for the Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml setting.
4. In the Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml setting, click + .
5. In Name enter the property tez.aux.uris and in value enter path-to-ssl-client.xml.

   Ensure that you include the file URI scheme in the path. For example:

   ```
   tez.aux.uris=file:///etc/hadoof/conf
   ```

Configure edge nodes as gateways

If you use command-line clients, such as Sqoop, to access Hive, you must configure these gateways to use defaults for your service. You can accomplish this task in a few steps.

About this task

By default, the HS2 instances configured in the migration already have the default beeline-site.xml file defined for the service. Other hosts do not. Configure these hosts as a gateway for that service.

Procedure

1. Find the notes you made before the upgrade about edge nodes and default, connected endpoints.
2. In Cloudera Manager, configure hosts other than HiveServer (HS2) hosts that you want to be Hive Gateway nodes as gateways for the default beeline-site.xml file for the gateway service.

Related Information

Capture Information about Multiple HiveServers
Adding a Service
Managing Roles
Adding a Host to a Cluster

Spark integration with Hive

You need to know a little about Hive Warehouse Connector (HWC) and how to find more information because to access Hive from Spark, you need to use HWC implicitly or explicitly.

You can use the Hive Warehouse Connector (HWC) to access Hive managed tables from Spark. HWC is specifically designed to access managed ACID v2 Hive tables, and supports writing to tables in Parquet, ORC, Avro, or Textfile formats. HWC is a Spark library/plugin that is launched with the Spark app.

You do not need HWC to read from or write to Hive external tables. Spark uses native Spark to access external tables.

Use the Spark Direct Reader and HWC for ETL jobs. For other jobs, consider using Apache Ranger and the HiveWarehouseConnector library to provide row and column, fine-grained access to the data.

HWC supports spark-submit and pyspark. The spark thrift server is not supported.

Related Information

Hive Warehouse Connector for accessing Apache Spark data

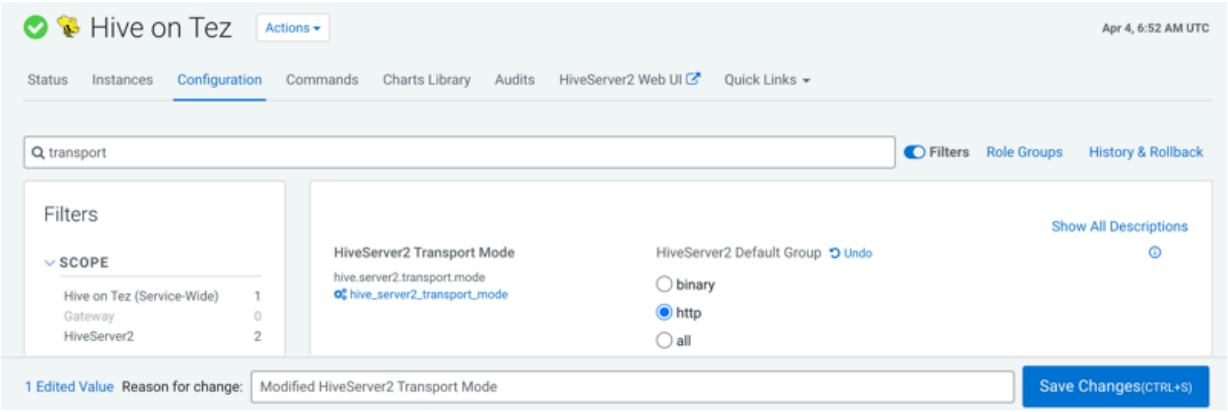
Spark Direct Reader for accessing Spark data

Configure HiveServer HTTP mode

If you use Knox, you might need to change the HTTP mode configuration. If you installed Knox on Cloudera Base on premises and want to proxy HiveServer with Knox, you need to change the default HiveServer transport mode (hive.server2.transport.mode).

Procedure

1. Click Cloudera Manager Clusters HIVE_ON_TEZ Configuration
2. In Search, type transport.
3. In HiveServer2 Transport Mode, select http.



4. Save and restart Hive on Tez.

Configuring HMS for high availability

To provide failover to a secondary Hive metastore if your primary instance goes down, you need to know how to add a Metastore role in Cloudera Manager and configure a property.

About this task

Multiple HMS instances run in active/active mode. No load balancing occurs. An HMS client always reaches the first instance unless it is down. In this case, the client scans the hive.metastore.uris property that lists the HMS instances for a replacement HMS. The second HMS is the designated replacement if hive.metastore.uri.selection is set to SEQUENTIAL (recommended and the default); otherwise, the replacement is selected randomly from the list if hive .metastore.uri.selection is set to RANDOM.

Before you begin

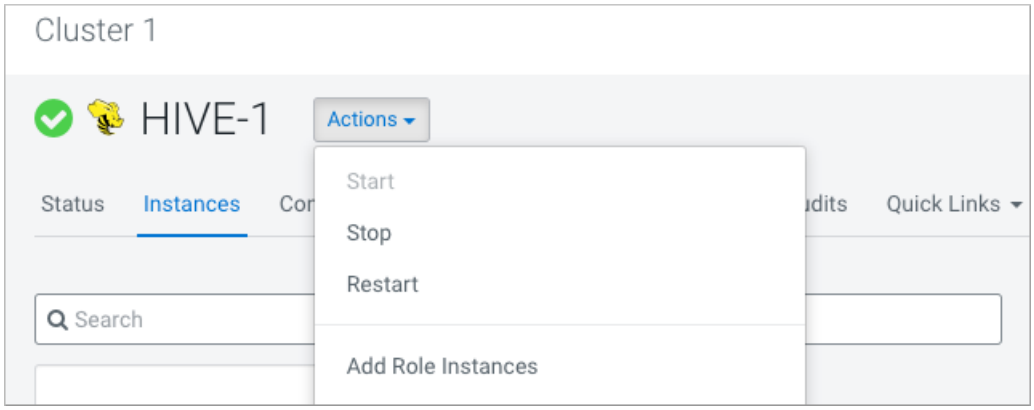
Minimum Required Role: Configurator (also provided by Cluster Administrator, Full Administrator)

Procedure

1. In Cloudera Manager, click Clusters Hive Configuration .
2. Take one of the following actions:
   - If you have a cluster secured by Kerberos, search for Hive Delegation Token Store, which specifies storage for the Kerberos token as described below.
   - If you have an unsecured cluster, skip the next step.
3. Select org.apache.hadoop.hive.thrift.DBTokenStore, and save the change.

   Storage for the Kerberos delegation token is defined by the hive.cluster.delegation.token.store.class property. The available choices are Zookeeper, the Metastore, and memory. Cloudera recommends using the database by setting the org.apache.hadoop.hive.thrift.DBTokenStore property.

4. Click Instances Actions Add Role Instances



5. In Assign Roles, in Metastore Server, click Select Hosts.
6. In Hosts Selected, scroll and select the host that you want to serve as the backup Metastore, and click OK.
7. Click Continue until you exit the wizard.
8. Start the Metastore role on the host from the Actions menu.

   The hive.metastore.uris property is updated automatically.

9. To check or to change the hive.metastore.uri.selection property, go to Clusters Hive Configurations , and search for Hive Service Advanced Configuration Snippet (Safety Valve) for hive-site.xml.
10. Add the property and value (SEQUENTIAL or RANDOM).

Installing Hive on Tez and adding a HiveServer role

Cloudera Runtime services include Hive on Tez and Hive Metastore (HMS). Hive on Tez is a SQL query engine using Apache Tez that performs the HiveServer (HS2) role in a Cloudera cluster. You must install the Hive service (HMS) before Hive on Tez and HMS in the correct order; otherwise, HiveServer will fail. You must add the HiveServer role additionally to Hive on Tez, not the Hive service; otherwise, HiveServer will fail.

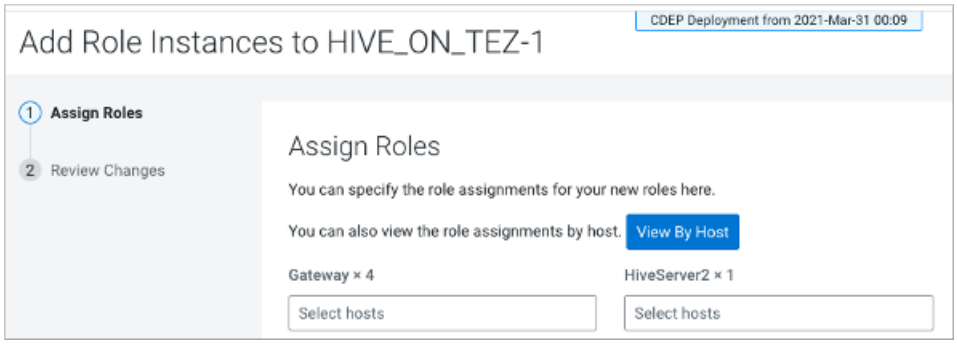
Procedure

1. Install the HMS, designated Hive service in Cloudera.
2. Install Hive on Tez
   HiveServer is installed automatically during this process.
3. Accept the default, or change the Hive warehouse location for managed and external tables as described below.

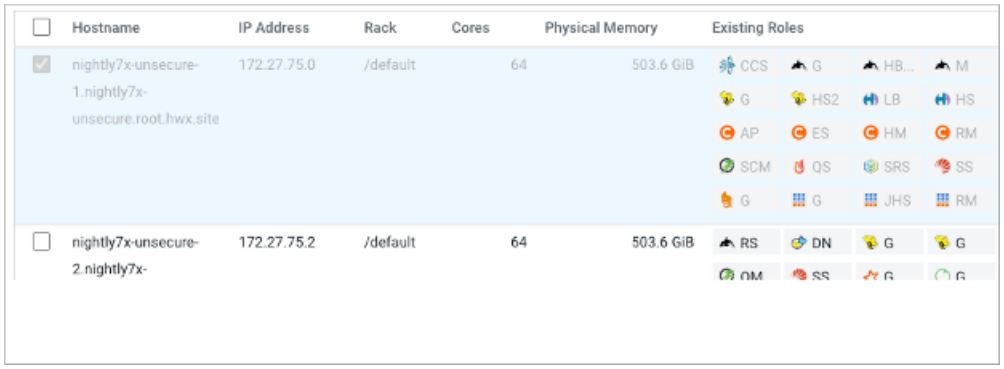
Adding a HiveServer role

Procedure

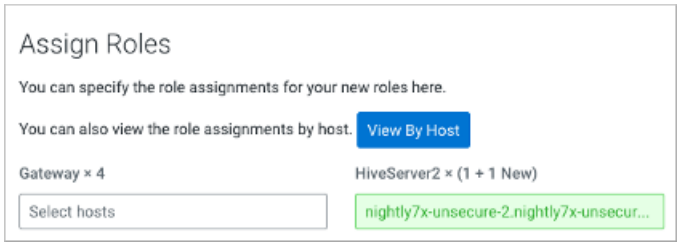
1. In Cloudera Manager, click Clusters Hive on Tez .
   Do not click Clusters Hive by mistake. This selects the Hive metastore and ultimately results in failure.
2. Click Actions Add Role Instances .



3. Click in the HiveServer2 box to select hosts.

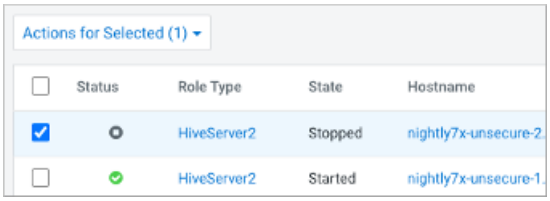


4. In the Host name column, select a host for the HiveServer2 role, and click OK.
   The selected host name you assigned the HiveServer2 role appears under HiveServer2.

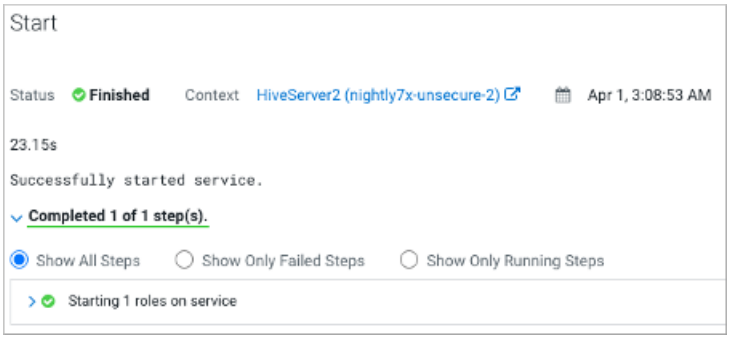


5. Click Continue.
   The new HiveServer2 role state is stopped.

6. Select the new HiveServer2 role.



7. In Actions for Selected, select Start, and then click Start to confirm.
You see that the service successfully started.



Changing the Hive warehouse location

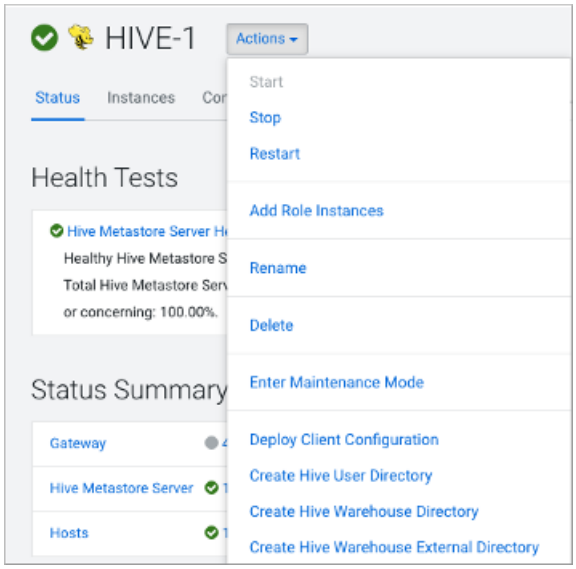
About this task

You use the Hive Metastore Action menu in Cloudera Manager, and navigate to one of the following menu items in the first step below.

- Hive Action Menu Create Hive Warehouse Directory
- Hive Action Menu Create Hive Warehouse External Directory

Procedure

1. Set up directories for the Hive warehouse directory and Hive warehouse external directory from Cloudera Manager Actions.



2. In Cloudera Manager, click Clusters Hive (the Hive Metastore service) Configuration , and change the hive.metastore.warehouse.dir property value to the path you specified for the new Hive warehouse directory.
3. Change the hive.metastore.warehouse.external.dir property value to the path you specified for the Hive warehouse external directory.

4. Configure Ranger policies or set up ACL permissions to access the directories.

Updating Hive and Impala JDBC/ODBC drivers

After upgrading, Cloudera recommends that you update your Hive and Impala JDBC and ODBC drivers. You follow a procedure to download a driver.

Before you begin

Configure authenticated users for running SQL queries through a JDBC or ODBC driver. For example, set up a Ranger policy.

Getting the JDBC driver

You learn how to download the Cloudera Hive and Impala JDBC drivers to give clients outside the cluster access to your SQL engines.

Procedure

1. Download the lastest Hive JDBC driver for CDP from the Hive JDBC driver download page.
2. Go to the Impala JDBC driver page, and download the latest Impala JDBC driver.
3. Follow JDBC driver installation instructions on the download page.

Getting the ODBC driver

You learn how to download the Cloudera ODBC drivers for Hive and Impala.

Procedure

1. Download the latest Hive ODBC driver for CDP from the Cloudera ODBC driver download page.
2. Go to the Impala ODBC driver page, and download the latest Impala ODBC driver.
3. Follow ODBC driver installation instructions on the download page.

Apache Impala changes in CDP

You need to understand changes that affect Impala after you upgrade from CDH 5.13-5.16 or CDH 6.1 or later to Cloudera Base on premises. The version of Impala you used in CDH 5.11 - 5.16 or 6.1 or later changes to Impala 3.4 when you upgrade to Cloudera Base on premises.

Table Changes

Upgrading CDH to Cloudera Base on premises converts Impala managed tables to external tables.

Location of Tables

If Impala managed tables were located on the HDFS in /user/hive/warehouse before the upgrade, the tables, converted to external, remain there. The upgrade process sets the hive.metastore.warehouse.dir property to this location, designating it the Hive warehouse location.

Changes to Impala Syntax or Service

- If you upgrade from 5.11-5.16 to Cloudera Base on premises, the following changes described in detail below, are relevant:
  - Decimal V2 Default
  - Behavior of Column Aliases
  - Default PARQUET_ARRAY_RESOLUTION
  - Enable Clustering Hint for Inserts
  - Deprecated Query Options Removed
  - refresh_after_connect Impala Shell Option Removed
  - Return Type Changed for EXTRACT and DATE_PART Functions
- If you upgrade from CDH 5.15 or 5.14, the ##compact_catalog_topicimpalad flag default value changed to true.
- If you upgrade from 6.1-6.3 to Cloudera Base on premises, the following changes, described in detail below, are relevant:
  - Port Change for SHUTDOWN Command
  - Change in Client Connection Timeout
- After upgrading from any CDH 5.x version to Cloudera Base on premises 7.1, recompute the statistics for Impala. Even though CDH 5.x statistics will be available after the upgrade, the queries will not benefit from the new features until the statistics are recomputed.
- Impala supports a number of file formats used in Apache Hadoop. It can also load and query data files produced by other Hadoop components such as hive. After upgrading from any CDH 5.x version to Cloudera Base on premises 7.1, if you create a RC file in Hive using the default LazyBinaryColumnarSerDe, Impala will not be able to read the RC file. However you can set the configuration option of hive.default.rcfile.serde to ColumnarSerDe to maintain the interoperability between hive and impala.
- After upgrading from CDH to CDP, the on-demand use_local_catalog mode is set to True by default on all the Impala coordinators so that the Impala coordinators pull metadata as needed from catalogd and cache it locally. This reduces memory footprint on coordinators and automate the cache eviction.
- In CDP, the catalog_topic_mode is set to minimal by default to enable on demand metadata for all coordinators.

Decimal V2 Default

In CDP, Impala uses DECIMAL V2 by default.

If you need to continue using the first version of the DECIMAL type for the backward compatibility of your queries, set the DECIMAL_V2 query option to FALSE:

```
SET DECIMAL_V2=FALSE;
```

Column Aliases Substitution

To conform to the SQL standard, Impala no longer performs alias substitution in the subexpressions of GROUP BY, HAVING, and ORDER BY.

The example below references to the actual column sum(ss_quantity) in the ORDER BY clause instead of the alias Total_Quantity_Purchased and also references to the actual column ss_item_sk in the GROUP BYclause instead of the alias Item since aliases are no longer supported in the subexpressions.

```
select
ss_item_sk as Item,
count(ss_item_sk) as Times_Purchased,
sum(ss_quantity) as Total_Quantity_Purchased
from store_sales
group by ss_item_sk
order by sum(ss_quantity) desc
limit 5;
+-------+-----------------+-------------------------+
```

```
| item  | times_purchased | total_quantity_purchased |
+-------+-----------------+--------------------------+
| 9325  | 372             | 19072                    |
| 4279  | 357             | 18501                    |
| 7507  | 371             | 18475                    |
| 5953  | 369             | 18451                    |
| 16753 | 375             | 18446                    |
+-------+-----------------+--------------------------+
```

Default PARQUET_ARRAY_RESOLUTION

The default value for the PARQUET_ARRAY_RESOLUTION is THREE_LEVEL to match the Parquet standard 3-level encoding.

Clustered Hint Default

The clustered hint is enabled by default, which adds a local sort by the partitioning columns HDFS and Kudu tables to a query plan. The noclustered hint, which prevents clustering in tables having ordering columns, is ignored with a warning.

Query Options Removed

The following query options have been removed:

- DEFAULT_ORDER_BY_LIMIT
- ABORT_ON_DEFAULT_LIMIT_EXCEEDED
- V_CPU_CORES
- RESERVATION_REQUEST_TIMEOUT
- RM_INITIAL_MEM
- SCAN_NODE_CODEGEN_THRESHOLD
- MAX_IO_BUFFERS
- RM_INITIAL_MEM
- DISABLE_CACHED_READS

Shell Option refresh_after_connect

The refresh_after_connect option for starting the Impala Shell is removed.

Deprecated Support to LZOCompressed Tables

In CDH, you might have used LZO compression for the text files for more compact data. And Impala supported text files that employ LZO compression. But in CDP, we deprecated support to LZO compressed tables since impala-lzo plugin is no longer shipped as part of GPL Extras.

EXTRACT and DATE_PART Functions

The EXTRACT and DATE_PART functions changed in the following way:

- The output type of the EXTRACT and DATE_PART functions was changed to BIGINT.
- Extracting the millisecond part from a TIMESTAMP returns the seconds component and the milliseconds component. For example, EXTRACT (CAST('2006-05-12 18:27:28.123456789' AS TIMESTAMP), 'MILLIS ECOND') will return 28123.

Port for SHUTDOWN Command

If you upgraded from CDH 6.1 or later and specified a port as part of the SHUTDOWN command, change the port number parameter to use the Kudu7: RPC (KRPC) port for communication between the Impala brokers.

Change in Client Connection Timeout

The default behavior of client connection timeout changes after upgrading.

In CDH 6.2 and lower, the client waited indefinitely to open the new session if the maximum number of threads specified by --fe_service_threads has been allocated.

After upgrading, the server requires a new startup flag, --accepted_client_cnxn_timeout to control treatment of new connection requests the configured number of server threads is insufficient for the workload.

If --accepted_client_cnxn_timeout > 0, new connection requests are rejected after the specified timeout.

If --accepted_client_cnxn_timeout=0, clients waits indefinitely to connect to Impala. This sets pre-upgrade behavior.

The default timeout is 5 minutes.

Advanced Array Handling in SQL with JOIN and UNNEST

Use the JOIN and UNNEST functions to query arrays in SQL. JOIN creates combinations of array elements, while UNNEST zips multiple arrays together, aligning their elements side by side, even if they have different lengths, with unmatched elements filled with NULL. For more information, see *Querying arrays*.

Related Information

Decimal Data Type
Impala Aliases
Impala Query Options

Set ACLs for Impala

To allow Impala to write to the Hive Warehouse Directory you must set ACLs for Impala.

About this task

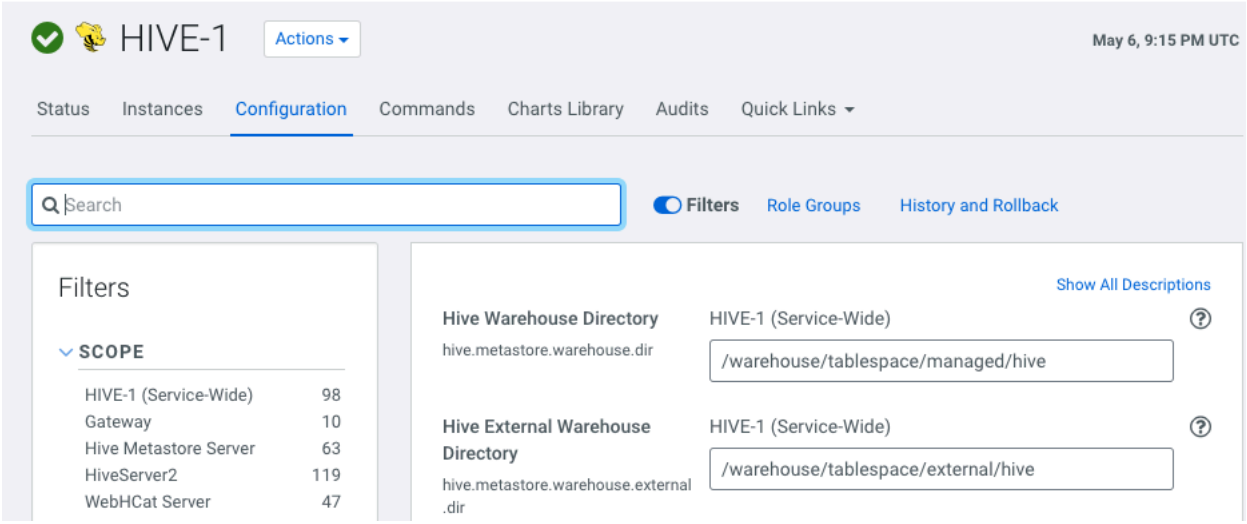
The location of existing tables after a CDH to CDP upgrade does not change. In CDP, there are separate HDFS directories for managed and external tables.

- The data files for managed tables are located in warehouse location specified by the Cloudera Manager configuration setting, Hive Warehouse Directory.
- The data files for external tables are located in warehouse location specified by the Cloudera Manager configuration setting, Hive Warehouse External Directory.

During the upgrade from CDH to CDP, the ACL settings are taken care automatically for the default warehouse directories. If you decide to change the default warehouse directories after upgrading to CDP then you must run the commands shown in Step 3.

After upgrading, the (hive.metastore.warehouse.dir) is set to /warehouse/tablespace/managed/hive where the Impala managed tables are located.

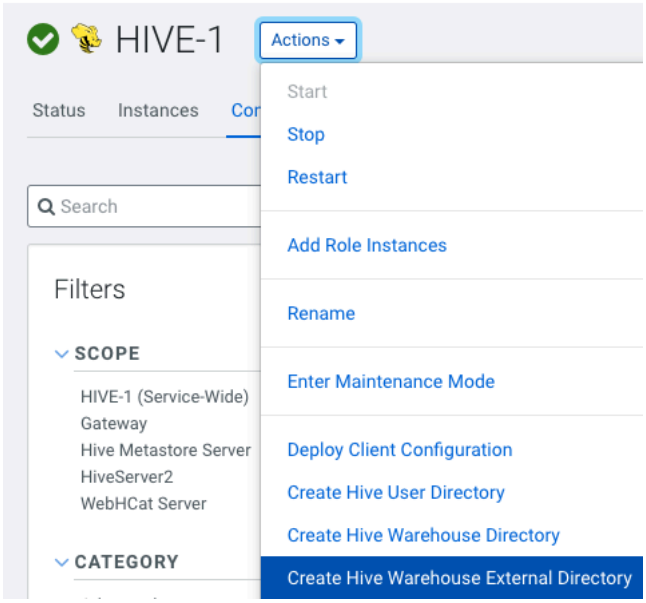
You can change the location of the warehouse using the Hive Metastore Action menu in Cloudera Manager.



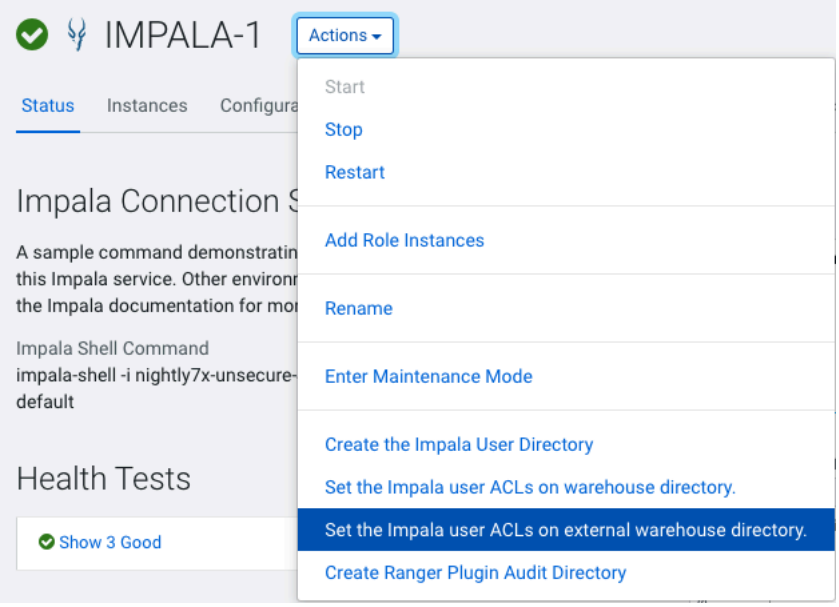
Complete the initial configurations in the free-form fields on the Hive/Impala Configuration pages in Cloudera Manager to allow Impala to write to the Hive Warehouse Directory.

Procedure

1. Create Hive Directories using the Hive Configuration page
   a) Hive Action Menu Create Hive User Directory
   b) Hive Action Menu Create Hive Warehouse Directory
   c) Hive Action Menu Create Hive Warehouse External Directory



2. Set Up Impala User ACLs using the Impala Configuration page
   a) Impala Action Menu Set the Impala user ACLs on warehouse directory
   b) Impala Action Menu Set the Impala user ACLs on external warehouse directory



3. Cloudera Manager sets the ACL for the user "Impala" however before starting the Impala service, verify permissions and ACLs set on the individual database directories using the sub-commands getfacl and setfacl.
   a) Verify the ACLs of HDFS directories for managed and external tables using getfacl.

   Example:

```
$ hdfs dfs -getfacl hdfs:///warehouse/tablespace/managed/hive
# file: hdfs:///warehouse/tablespace/managed/hive
# owner: hive
# group: hive
user::rwx
group::rwx
other::---
default:user::rwx
default:user:impala:rwx
default:group::rwx
default:mask::rwx
default:other::---
```

```
$ hdfs dfs -getfacl hdfs:///warehouse/tablespace/external/hive
# file: hdfs:///warehouse/tablespace/external/hive
# owner: hive
# group: hive
# flags: --t
user::rwx
group::rwx
other::rwx
default:user::rwx
default:user:impala:rwx
default:group::rwx
default:mask::rwx
default:other::--x
```

b) If necessary, set the ACLs of HDFS directories using setfacl

Example:

```
$ hdfs dfs -setfacl hdfs:///warehouse/tablespace/managed/hive
$ hdfs dfs -setfacl hdfs:///warehouse/tablespace/external/hive
```

For more information on using the sub-commands getfacl and setfacl, see Using CLI commands to create and list ACLs.

c) The above examples show the user Impala as part of the Hive group. If in your setup, the user Impala does not belong to the group Hive then ensure that the Group user Impala belongs to has WRITE privileges assigned on the directory.

To view the Group user Impala belongs to:

```
$ id -Gn impala
uid=973(impala) gid=971(impala) groups=971(impala),972(hive)
```

Related Information

HDFS ACLS

Changes to CDH Hive Tables

Impala Configuration Changes

The upgrade process to Cloudera Base on premises changes the default values of some Impala configuration properties and adds new properties.

Impala Configuration Property Values

The following list describes Impala configuration property value changes or additions that occur after upgrading from CDH or HDP to CDP. These changes in properties ensure that CDP Hive and Impala interoperate to the best of their abilities. The CDP default might have changed in a way that impacts your work.

Note: After an upgrade you may see some or all of the configurations listed here. If you do not see any of the following configurations by default these are the recommended configurations you must consider adding post upgrade in your CDP environment.

default_file_format

Before upgrade: text

After upgrade: parquet

In CDP the default Impala table file format changed from text to parquet. If the file format is not parquet, add stored as clause in the create table statements explicitly or change the query option default_file_format to text to revert to the behavior as CDH.

default_transactional_type

Before upgrade: N/A

After upgrade: insert_only

In CDP the default table type for managed tables is insert_only. If you must revert to the behavior as CDH, set default_transactional_type to none. These transactional tables cannot be currently altered in Impala using an alter statement. Similarly, Impala does not support compaction on transaction tables currently. You must use Hive to compact the tables as needed. Other operations like select, insert, insert overwrite, truncate are supported. For latest information, see SQL transactions in Impala.

Note: default_file_format and default_transactional_type can be set under Impala > Configuration > default_query_options.

hms_event_polling_interval_s

Before upgrade: 0

After upgrade: 2

When raw data is inserted ingested into Tables, new HMS metadata and filesystem metadata are generated. In CDH, in order to pick up this new information, you must manually issue an Invalidate or Refresh command. However in CDP, hms_event_polling_interval_s property is set to 2 seconds by default. This option automatically refreshes the tables as changes are detected in HMS. If only specific tables that are not supported by event polling need to be refreshed issue a table level Invalidate or Refresh command.

disconnected_session_timeout

Before upgrade: N/A

After upgrade: 900

In CDP Impala supports the ability to disconnect a connection to Impala while keeping the session running. Impala clients/drivers may support re-connecting to the same session even when the network connection is interrupted. By default disconnected sessions are not terminated until 15 minutes if you want to reconnect. You can adjust the disconnected_session_timeout flag to a lower value so that disconnected sessions are cleaned up more quickly.

enable_orc_scanner

Before upgrade: True (preview)

After upgrade: True

While using Impala to query ORC tables, set the command line argument enable_orc_scanner = true to re-enable ORC table support.

enable_insert_events

Before upgrade: N/A

After upgrade: True

If Impala inserts into a table it refreshes the underlying table/partition. When this configuration enab le_insert_eventsis set to True Impala will generate INSERT event types which when received by other Impala clusters will automatically refresh the tables or partitions. Event processing must be ON, for this property to work.

disable_hdfs_num_rows_estimate

Before upgrade: N/A

After upgrade: False

In CDP Impala, if there are no statistics available on a table, Impala will try to estimate the cardinality by estimating the size of table based on the number of rows in the table. This behavior is turned ON by default to use when stats are not present. However you can set the query option disa ble_hdfs_num_rows_estimate = true to disable this optimization.

use_local_catalog

Before upgrade: False

After upgrade: True

In CDP, the on-demand use_local_catalog mode is set to True by default on all the Impala coordinators so that the Impala coordinators pull metadata as needed from catalogd and cache it locally. This reduces memory footprint on coordinators and automate the cache eviction.

catalog_topic_mode

Before upgrade: full

After upgrade: minimal

In CDP, the catalog_topic_mode is set to minimal by default to enable on demand metadata for all coordinators.

Interoperability between Hive and Impala

This topic describes the changes made in CDP for the optimal interoperability between Hive and Impala for the improved user experience.

Statistics Interoperability Between Hive and Impala

New default behavior:

Statistics for tables are engine specific, namely, Hive or Impala, so that each engine could use its own statistics and not overwrite the statistics generated by the other engine.

When you issue the COMPUTE STATS statement on Impala, you need to issue the corresponding statement on Hive to ensure both Hive and Impala statistics are accurate.

Impala COMPUTE STATS command does not overwrite the Hive stats for the same table.

Steps to switch to the CDH behavior:

There is no workaround.

Hive Default File Format Interoperability

New default behavior:

The managed tables created by Hive are of ORC file format, by default, and support full transactional capabilities. If you create a table without specifying the STORED AS clause and load data from Hive, then such tables are not readable or writable by Impala. But Impala can continue to read non-transactional and insert-only transactional ORC tables.

Steps to switch to the CDH behavior:

- You must use the STORED AS PARQUET clause when you create tables in Hive if you want interoperability with Impala on those tables.
- If you want to change this default file format at the system level, in the Hive_on_Tez service configuration in Cloudera Manager, set the hive_default_fileformat_managed field to parquet.

Impala supports a number of file formats used in Apache Hadoop. It can also load and query data files produced by other Hadoop components such as hive. After upgrading from any CDH 5.x version to Cloudera Base on premises 7.1, if you create a RC file in Hive using the default LazyBinaryColumnarSerDe, Impala will not be able to read the RC file. However you can set the configuration option of hive.default.rcfile.serde to ColumnarSerDe to maintain the interoperability between hive and impala.

Managed and External Tablespace Directories

New default behavior:

In CDP, there are separate HDFS directories for managed and external tables.

- The data files for managed tables are located in warehouse location specified by the Cloudera Manager configuration setting, hive_warehouse_directory.
- The data files for external tables are located in warehouse location specified by the Cloudera Manager configuration setting, hive_warehouse_external_directory.

If you perform file system level operations for adding/removing files on the table, you need to consider if its an external table or managed table to find the location of the table directory.

Steps to switch to the CDH behavior:

Check the output of the DESCRIBE FORMATTED command to find the table location.

Revert to CDH-like Tables

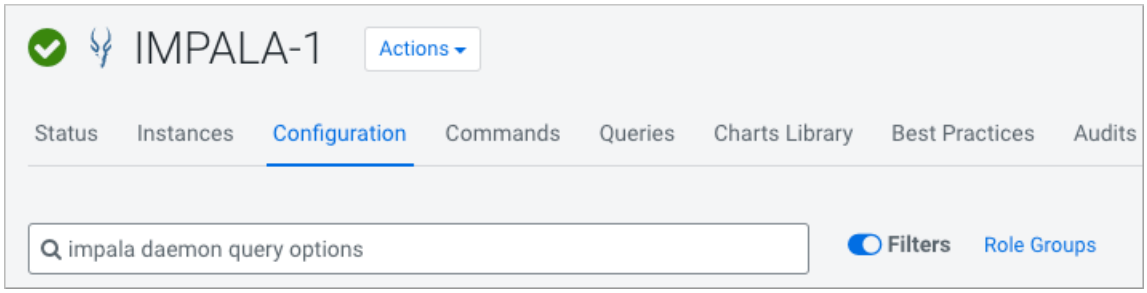
In CDH 5 and CDH 6, CREATE TABLE created managed non-ACID tables in text format. In CDP, CREATE TABLE creates an INSERT-ONLY table in parquet format. After upgrading, to avoid code changes due to new features in CDP, you might want want to disable the new transactional (ACID) table type and parquet file format defaults. The old table type and file format (managed, non-transactional and text) take effect when you disable the new defaults.

About this task

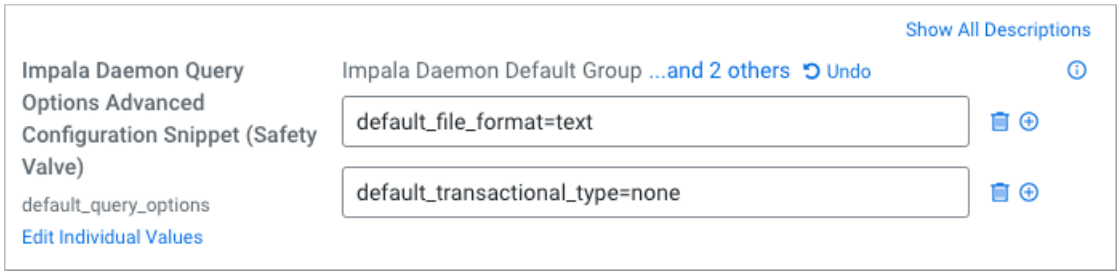
To disable the new defaults:

Procedure

1. In Cloudera Manager Clusters select the Impala service. Click Configuration and search for Impala Daemon Query Options.



2. In Impala Daemon Query Options Advanced Configuration Snippet (Safety Valve) for default_query_options edit the default_file_format=text and default_transactional_type=none properties.



Authorization Provider for Impala

In CDP, Ranger is the authorization provider instead of Sentry. There are some changes with how Ranger enforces a policy which may be different from using Sentry.

New behavior:

- The CREATE ROLE, GRANT ROLE, SHOW ROLE statements are not supported as Ranger currently does not support roles.
- When a particular resource is renamed, currently, the policy is not automatically transferred to the newly renamed resource.
- SHOW GRANT with an invalid user/group does not return an error.

The following table lists the different access type requirements to run SQL statements in Impala.

SQL Statement	Impala Access Requirement
DESCRIBE *VIEW*	VIEW_METADATA on the underlying tables
ALTER TABLE RENAME ALTER VIEW RENAME	ALL on the target table / view ALTER on the source table / view
SHOW DATABASES SHOW TABLES	VIEW_METADATA

where:

- VIEW_METADATA privilege denotes the SELECT, INSERT, or REFRESH privileges.
- ALL privilege denotes the SELECT, INSERT, CREATE, ALTER, DROP and REFRESH privileges.

For more information on the minimum level of privileges and the scope required to execute SQL statements in Impala, see Impala Authorization.

Migrating Sentry Policies

When upgrading from CDH to CDP, all SQL permissions and Kafka permissions will be migrated. However if you must migrate some sentry policies from your CDH environment to the new environment you can use the Replication Manager service available in CDH. This service migrates Sentry authorization policies into Ranger as part of the replication policy. Sentry policy migration takes place as part of a replication policy job. When you create the replication policy, choose the resources that you want to migrate and the Sentry policies will be migrated for those resources.

For more information on using the Replication Manager service to migrate Sentry policies, see Sentry Policy Replication.

Note: Since the authorization model in Ranger is different from Sentry's not all policies can be migrated using Replication Manager. For some resources you must manually create the permissions after upgrade.

Related Information

Impala Authorization

Data Governance Support by Atlas

Both CDH and CDP environments support governance functionality for Impala operations. The two environments collect similar information to describe Impala activities, including:

- Audits for Impala access requests
- Metadata describing Impala queries
- Metadata describing any new data assets created or updated by Impala operations

The services that support these operations are different in the two environment. Functionality is distributed across services as follows:

Feature	CDH	CDP
Auditing		
• Access requests	Audit tab in Navigator console	Audit page in Ranger console
• Service operations that create or update metadata catalog entries	Audit tab in Navigator console	Audit tab for each entity in Atlas dashboard
• Service operations in general	Audit tab in Navigator console	No other audits collected.
Metadata Catalog		

• Impala operations:
  • CREATE TABLE AS SELECT
  • CREATE VIEW
  • ALTER VIEW AS SELECT
  • INSERT INTO
  • INSERT
  • OVERWRITE
	Process and Process Execution entities
	Column- and table-level lineage
		Process and Process Execution entities
		Column- and table-level lineage

Migrating Navigator content to Atlas

As part of upgrading a CDH cluster to CDP, Atlas replaces Cloudera Navigator Data Management for your cluster. You can choose to migrate your Navigator metadata to Atlas as part of upgrading. Migrating content from Navigator to Atlas involves 3 steps:

- extracting the content from Navigator,
- transforming that content into a form Atlas can consume,
- importing the content into Atlas.

See Migrating Metadata from Navigator to Atlas for more information on the high-level migration process.

Related Information

Migrating Navigator Content to Atlas

Handling Data Files

You must know how to recursively load the data files, for transactional tables, that are not stored directly within the partition directories, but instead are stored within subdirectories corresponding to writeIds, compactions, etc.

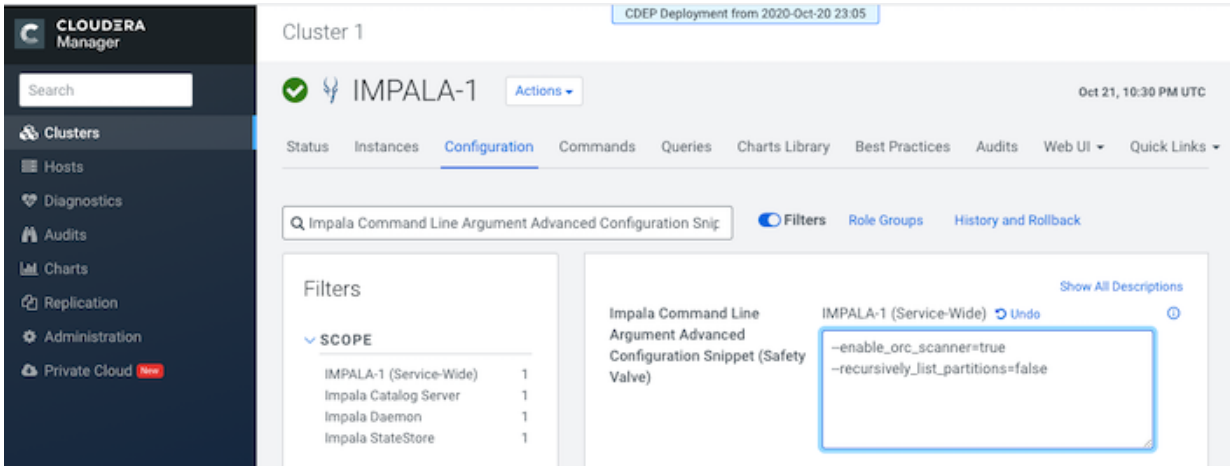
About this task

In CDP 7.x, Impala includes files within subdirectories. If you must restore to the old behavior and recursively load file lists within partition directories, you can use the Safety Valve to add the property --recursively_list_partitions in Impala service as shown in this task. This change can be done either per table or globally.

Procedure

To make the changes globally:

1. In Cloudera Manager Clusters select the Impala service. Click Configuration, and search for Impala Command Line Argument Advanced Configuration Snippet (Safety Valve).
2. In Impala Command Line Argument Advanced Configuration Snippet (Safety Valve) add the value --recursivel y_list_partitions=false.



3. Save the changes and restart the Impala service.

To make the changes on individual tables:

4. In CLI, enteralter table tablename set tblproperties('impala.disable.recursive.listing'='true'); refresh tablename;

Hue post-upgrade tasks

Review the changes for the Hive editor and the Security Browser after upgrading from CDH 5 or CDH 6 to CDP to avoid access issues.

You must install the Query Processor service on your CDP Private Cloud Base clusters manually to use features and functionality of DAS in Hue.

Updating group permissions for Hive query editor

In CDH 5 and CDH 6, the beeswax.access permission governs access to the Hive editor in Hue. In CDP, access to the Hive editor is governed by the hive.access permission and the permission is added to the default group, by default. If you upgrade from CDH 5 or CDH 6 to CDP, and if you are not a part of the default group, then you may not be able to access the Hive editor. You must manually grant the hive.access permission to your user groups.

Procedure

1. Log in to Hue as an Administrator.
2. Go to Admin Manage Users Permissions and click hive under Application.
3. On the **Edit hive** page, select the groups to which you want to grant access to the Hive editor.
4. Click Update permission to save the changes.

Adding Security Browser to the blocked list of applications

The Security Browser application is no longer supported in Hue on CDP. If you were using Sentry on your CDH cluster and a Security Browser in Hue, then you must manually add the Security Browser application to the list of blocked applications in the Hue Advanced Configuration Snippet after upgrading to CDP.

About this task

After upgrading from CDH to CDP, you may see the following error when accessing the Security Browser: “Failed to connect to Sentry server localhost”. To prevent this error:

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters Hue service Configuration Hue Service Advanced Configuration Snippet (Safety Valve) for hue_safety_valve.ini and append security to the app_blacklist property. For example:

```
[desktop]
app_blacklist=pig,spark,security
```

3. Click Save Changes.
4. Restart the Hue service.

Adding Query Processor service to a cluster

The Query Processor service indexes Hive and Tez events and provides APIs to access them. It is required if you want to view the Queries tab (query history and query details) on the Hue Job Browser. You must install the Query Processor service on your Cloudera Base on premises clusters manually.

About this task

Attention: This task is applicable only if you are upgrading to Cloudera 7.1.8 and higher. If you are upgrading to Cloudera 7.1.7 SP2, 7.1.7 SP1, 7.1.7, or lower, then you can skip this task.

Note: The Query Process service supports only MySQL.

You can either install the Query Processor service as a dependency while adding the Hue service or after adding the Hue service.

Before you begin

This task assumes that you already have a database installed on a host in your cluster. On Cloudera, Hue Query Processor supports PostgreSQL, MySQL, and Oracle databases. The supported PostgreSQL database version for Hue Query Processor is 9.6 and higher.

Next, you need to create a database for the Query Processor service with the required roles. To create the Query Processor database:

1. SSH into your database host as a root user.
2. Start the psql terminal by entering the following command:

```
sudo -u postgres psql
```

3. Run the following statement while specifying the username, password, and a database name for the Query Processor:

```
CREATE ROLE [***QP-USER***] WITH LOGIN PASSWORD '[***QP-PASSWORD***]';
ALTER ROLE [***QP-USER***] CREATEDB;
CREATE DATABASE [***QP-DATABASE***];
GRANT ALL PRIVILEGES ON DATABASE [***QP-DATABASE***] TO [***QP-USER***];
```

Procedure

1. Log in to Cloudera Manager as an Administrator.
2. Go to Clusters ⋮ Add Service .
3. Select Query Processor on the **Add Service to Cluster** page and click Continue.

   The required dependencies are automatically selected.
4. Select the host on which you want to install the Query Processor by clicking View By Host. Then click Continue.
5. Select the database type, and specify the database hostname, database name, and username and password to access the database on the **Setup Database** page and click Test Connection.

   After the connection is verified, click Continue.
6. On the **Review Changes** page, accept the default settings and click Continue.

   The **Command Details** page is displayed, which shows the status of the installation and configuration.
7. Click Continue if the operation is successful.
8. Click Finish on the **Summary** page.

Results

You are redirected to the Cloudera Manager home page. You can now see the Query Processor service listed within your cluster.

What to do next

(Optional) To view the **Queries** tab on the **Job Browser** page in Hue, you must enable the Query Processor Service option in Hue configurations using Cloudera Manager.

Importing Sentry privileges into Ranger policies

How to complete the process of translating Sentry privileges into Ranger policies.

About this task

No one-to-one mapping between Sentry privileges and Ranger service policies exists. Upgrading your platform involves translating Sentry privileges to their equivalents within Ranger service policies. After upgrading Cloudera Manager and your cluster, this post-upgrade step completes the translation process.

Procedure

1. In Ranger Actions , click Import Sentry Policies.
2. Read the following points that describe how Sentry privileges appear in Ranger after the migration:
   - Sentry permissions that are granted to roles are granted to groups in Ranger.
   - Sentry permissions that are granted to a parent object are granted to the child object as well. The migration process preserves the permissions that are applied to child objects. For example, a permission that is applied at the database level also applies to the tables within that database.
   - Sentry OWNER privileges are translated to the Ranger OWNER privilege.
   - Sentry OWNER WITH GRANT OPTION privileges are translated to Ranger OWNER with Delegated Admin checked.
   - Sentry does not differentiate between tables and views. When view permissions are migrated, they are treated as table names.
   - Sentry privileges on URIs use the object store location as the base location.
   - If your cluster contains the Kafka service and the Kafka sentry policy had "action": "ALL" permission, the migrated Ranger policy for "cluster" resource will be missing the "alter" permission. This is only applicable for "cluster" resource. You need to add the policy manually after the upgrade. This missing permission does not have any functional impact. Adding the "alter" permission post upgrade is needed only for completeness because the 'configure' permission allow alter operations.
   - Sentry "alter" permission on cluster and topic is translated to "configure" in Ranger.

   The following table shows how actions in Sentry translate to corresponding actions in Ranger:

Table 18: Sentry Actions to Ranger Actions

Sentry Action	Ranger Action
SELECT	SELECT
INSERT	UPDATE
CREATE	CREATE
REFRESH	REFRESH
ALL	ALL
SELECT with Grant	SELECT
INSERT with Grant	UPDATE
CREATE with Grant	CREATE
ALL with Grant	ALL with Delegated Admin Checked
ALTER	CONFIGURE

Apache Knox - create plugin audit directory

To create hdfs audit directories, you must create the Ranger Knox plugin audit directory manually, post-upgrade.

About this task

If you add the Knox service after upgrading a CDH 5.x/6.x cluster to CDP, you must create the Ranger Knox plugin audit dirctory manually in order for the hdfs audit directory to exist.

Before you begin

Install the Knox service. See related topics for more information about installing Knox.

Procedure

1. From Cloudera Manager, go to Knox Action Create Knox Plugin Audit Directory .
2. Restart Knox.

Apache Ranger TLS Post-Upgrade Tasks

For TLS/SSL enabled Ranger Service, to enable Yarn and Hbase plugin, you must:

1. Add Ranger Admin certificate in the following truststore files used by these services.
   Yarn

   TLS/SSL Client Truststore File Location (ssl.client.truststore.location)

   TLS/SSL Client Truststore File Password (ssl.client.truststore.password)

   Hbase

   HBase Master TLS/SSL Trust Store File (master_truststore_file)

   HBase Master TLS/SSL Trust Store Password (master_truststore_password)

   HBase Region Server TLS/SSL Trust Store File (regionserver_truststore_file)

   HBase Region Server TLS/SSL Trust Store Password (regionserver_truststore_password)
2. In Cloudera Manager UI Ranger KMS KTS Action , run Create Ranger Plugin Audit Directory command.

Migrating ACLs from Key Trustee KMS to Ranger KMS

You must perform the following procedures to migrate ACLs from Key Trustee Key Management Server (KMS) to Ranger KMS.

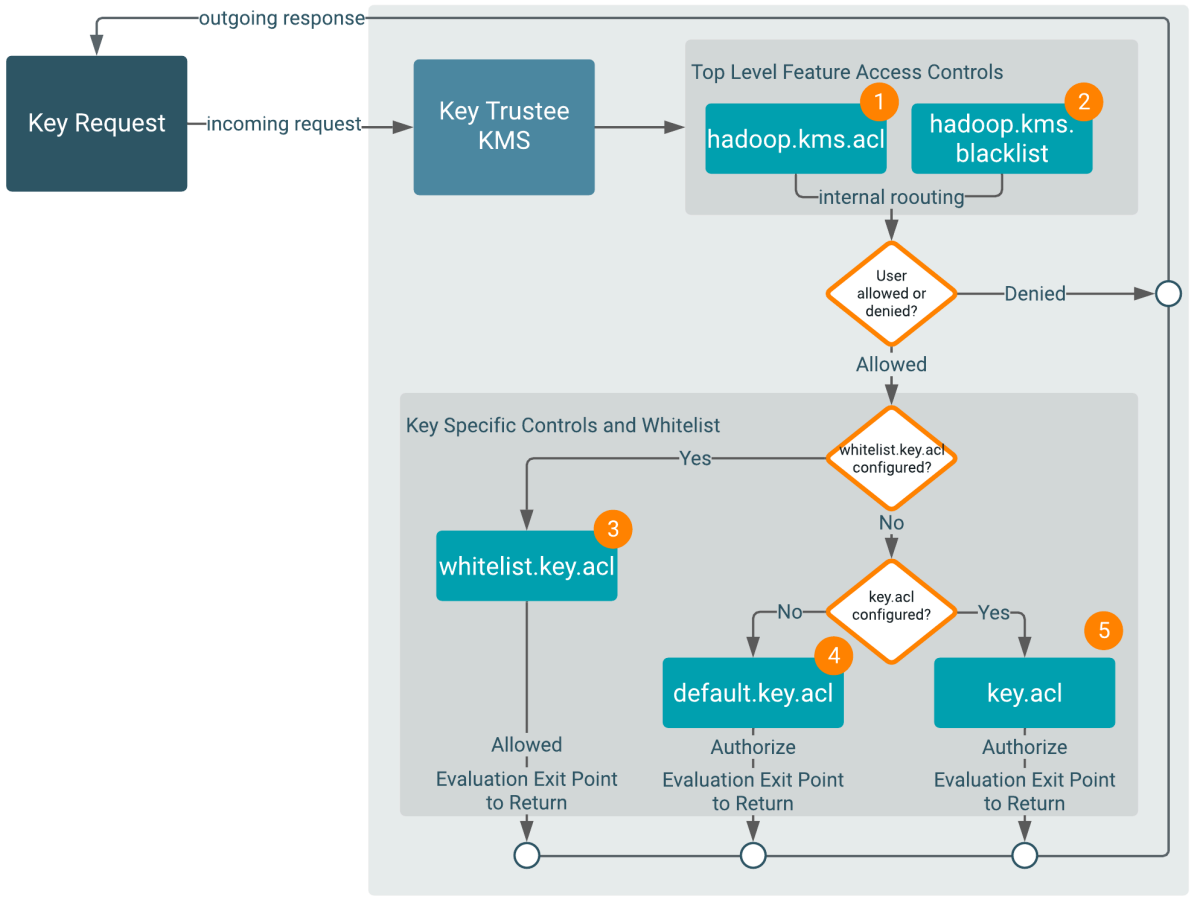
Key Trustee ACL evaluation

Before going into the details of how Key Trustee ACLs are evaluated, it is critical that you understand the key rules that the Key Trustee Key Management Server uses in performing this evaluation.

KMS ACL Flow Rules:

- The whitelist class bypasses key.acl and default.key.acl controls.
- The key.acl definitions override all default definitions.

Encryption key access is evaluated as follows:



1 and 2

The KMS evaluates the hadoop.kms.acl.<OPERATION> and hadoop.kms.blacklist.<OPERATION> classes to determine whether or not access to a specific KMS feature or function is authorized.

In other words, a user must be allowed by hadoop.kms.acl.<OPERATION>, and not be disallowed by hadoop.kms.blacklist.<OPERATION>.

If a user is denied access to a KMS-wide operation, then the flow halts and returns the result Denied.

If a user is allowed access to a KMS-wide operation, then the evaluation flow proceeds.

3

The KMS evaluates the whitelist.key.acl class.

The KMS ACL workflow evaluates the whitelist.key.acl.<OPERATION>, and if the user is allowed access, then it is granted (Allowed) . If not, then the flow continues with the evaluation.

4 and 5

The KMS evaluates the default.key.acl.<OPERATION> and key.acl.<OPERATION> classes.

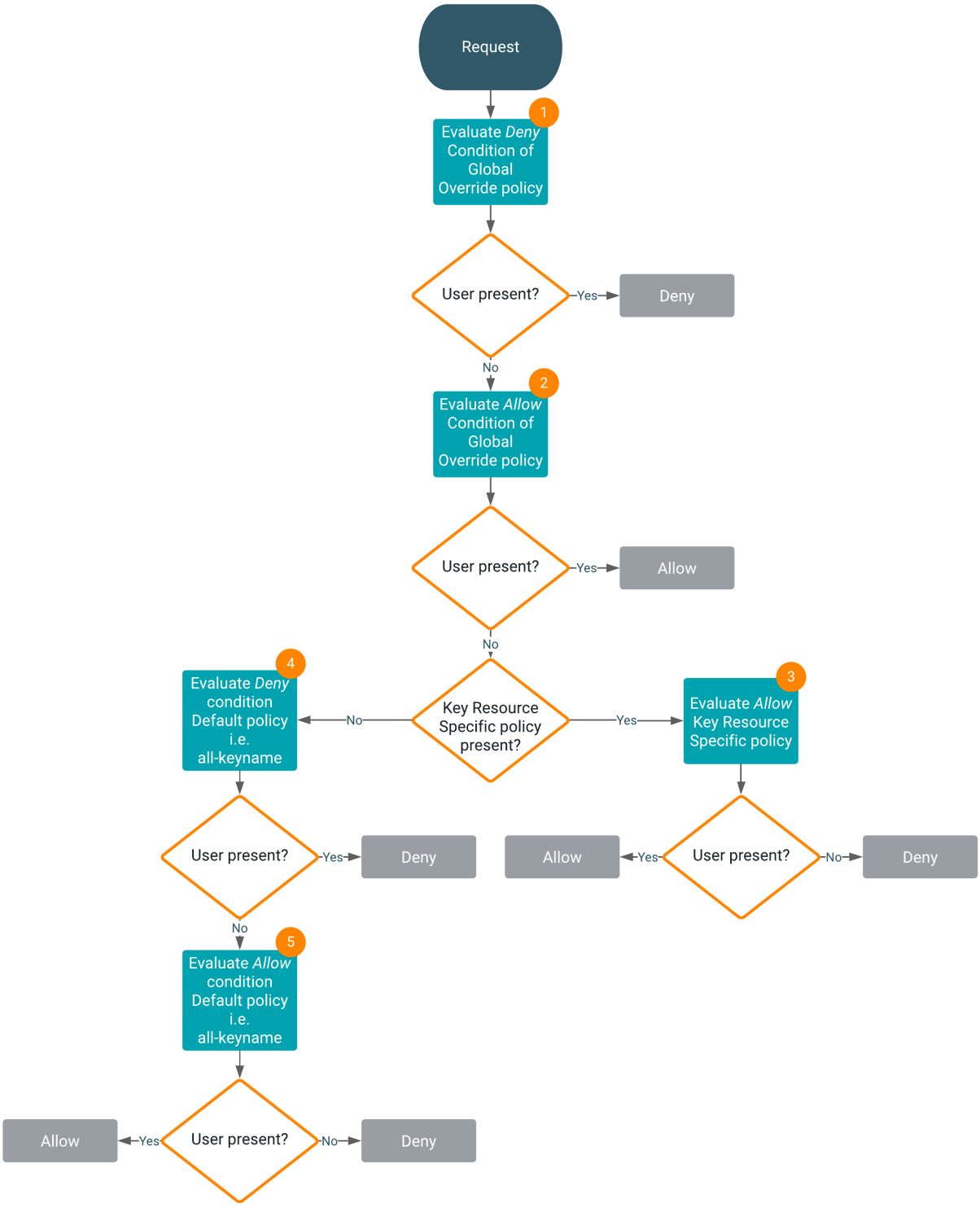
The KMS evaluates whether or not there is a key.acl.KEY.<OPERATION> class that matches the action the user is attempting to perform. If there is, it then evaluates that value to determine whether or not the user can perform the requested operation.

Note: Before evaulating the default.key.acl.<OPERATION> and key.acl.<OPERATION> classes, the flow logic determines which classes exist. Only one of these can exist and be used at any time (for example, key.acl.prodkey.READ overrides default.key.acl.READ for prodkey, so the flow logic is configured with it's own READ ACLs)

Depending on the result of the Key Trustee ACL evaluation, controls are applied to the key and results (Allowed or Denied).

Access evaluation with Ranger KMS policies

Access is evaluated with Ranger KMS policies as follows:



1

After the request is received, the Deny condition of the Global Override policy is evaluated. If the user is present, the flow halts and returns the result Deny. If the user is not present, the evaluation flow proceeds.

2

Now, the Allow condition of the Global Override policy is evaluated. If the user is present, the flow halts and returns the result Allow. If the user is not present, the evaluation flow proceeds.

3

If the Key Resource Specific policy is present, the Allow condition of the Key Resource Specific policy is evaluated. If the user is not present, the flow halts and returns the result Deny. If the user is present, the flow is complete and returns the result Allow.

4

If the Key Resource Specific policy is not present, the Deny condition of the Default policy, all-keyname, is evaluated. If the user is present, the flow halts and returns the result Deny. If the user is not present, the evaluation flow proceeds.

5

Now, the Allow condition of the Default policy, all-keyname, is evaluated. If the user is not present, the flow halts and returns the result Deny. If the user is present, the flow is complete and returns the result Allow.

Key Trustee KMS operations not supported by Ranger KMS

The following Key Trustee KMS operations are not supported by Ranger KMS.

- hadoop.kms.acl.<OPERATION>

  The ACLs mentioned below are ignored by Ranger KMS because these ACLs are not migrated to the Ranger KMS policy.

```
hadoop.kms.acl.CREATE
hadoop.kms.acl.DELETE
hadoop.kms.acl.ROLLOVER
hadoop.kms.acl.GET
hadoop.kms.acl.GET_KEYS
hadoop.kms.acl.GET_METADATA
hadoop.kms.acl.SET_KEY_MATERIAL
hadoop.kms.acl.GENERATE_EEK
hadoop.kms.acl.DECRYPT_EEK
```

- keytrustee.kms.acl.<OPERATION>

  The ACLs mentioned below are Key Trustee-specific ACLs. These ACLs are ignored by Ranger KMS because they are not migrated to the Ranger KMS policy. Also, these ACLs are not supported by Hadoop KMS.

```
keytrustee.kms.acl.UNDELETE
keytrustee.kms.acl.PURGE
```

Note: The KTS to Ranger KMS migration utility may exit with output similar to:

```
Following users do not exist in Ranger DB: [csso_xxxx, user01, csso_xxx2,
 ser2_old]
Following groups do not exist in Ranger DB: [usergroup1, usergroup2,
 unknown_test_group, usergroup_old]
To fix this problem, either add the users/groups to your user management
 system and re-sync the users/groups,
or, create the listed users/groups in Ranger, using the Ranger Admin Web
 UI: https://<servername>.root.hwx.site:6182/
```

The workaround is to add the required users and groups in the ranger database, as internal users and groups, using the Ranger Admin Web UI, then resume the upgrade process.

ACLs supported by Ranger KMS and Ranger KMS Mapping

The following ACLs are supported by Ranger KMS and Ranger KMS mapping.

- whitelist.key.acl.<operation> and hadoop.kms.blacklist.<Operation>

  In this case, you create a Global Override policy under the service cm_kms.

  Service : cm_kms

Policy	Key-resource	Priority	Key Trustee ACL	Ranger Policy Condition	Ranger Policy Permission
Global Override Policy	*	Override	whitelist.key.acl.MANAGEMENT	ALLOW	CREATE, DELETE, ROLLOVER
			whitelist.key.acl.GENERATE_EEK	ALLOW	GENERATE_EEK
			whitelist.key.acl.DECRYPT_EEK	ALLOW	DECRYPT_EEK
			whitelist.key.acl.READ	ALLOW	GET, GET KEYS, GET METADATA
			hadoop.kms.blacklist.CREATE	DENY	CREATE
			hadoop.kms.blacklist.DELETE	DENY	DELETE
			hadoop.kms.blacklist.ROLLOVER	DENY	ROLLOVER
			hadoop.kms.blacklist.GET	DENY	GET
			hadoop.kms.blacklist.GET_KEYS	DENY	GET KEYS
			hadoop.kms.blacklist.GET_METADATA	DENY	GET METADATA
			hadoop.kms.blacklist.SET_KEY_MATERIAL	DENY	SET KEY MATERIAL
			hadoop.kms.blacklist.GENERATE_EEK	DENY	GENERATE_EEK
			hadoop.kms.blacklist.DECRYPT_EEK	DENY	DECRYPT_EEK

- default.key.acl.<operation>

  Service : cm_kms

Policy	Key-resource	Priority	Key Trustee ACL	Ranger Policy Condition	Ranger Policy Permission
Default Policy all-keyname	*	Normal	default.key.acl.MANAGEMENT	ALLOW	CREATE, DELETE, ROLLOVER
			default.key.acl.GENERATE_EEK	ALLOW	GENERATE_EEK
			default.key.acl.DECRYPT_EEK	ALLOW	DECRYPT_EEK
			default.key.acl.READ	ALLOW	GET, GET KEYS, GET METADATA

- key.acl.<key-name>.<OPERATION> Key Specific ACL

  In this case, you create a Key Resource Specific policy under the service cm_kms.

  Service : cm_kms

Policy	Key-resource	Priority	Key Trustee ACL	Ranger Policy Condition	Ranger Policy Permission
Key Resource Specific policy <keyname>	<keyname>	Normal	key.acl.<key-name>.MANAGEMENT	ALLOW	CREATE, DELETE, ROLLOVER
			key.acl.<key-name>.GENERATE_EEK	ALLOW	GENERATE_EEK
			key.acl.<key-name>.DECRYPT_EEK	ALLOW	DECRYPT_EEK
			key.acl.<key-name>.READ	ALLOW	GET, GET KEYS, GET METADATA
			key.acl.<key-name>.ALL	ALLOW	SELECT ALL

Note: In Key Resource Specific policies, DENY ALL OTHER ACCESS flags are set to true.

Apache Hadoop YARN default value changes

A list of default value changes when upgrading from CDH to CDP.

Note: This is not a complete list. It only contains the default value changes that cause behaviour changes.

- Scheduler: In CDP, Capacity Scheduler is the supported and default scheduler. For more information about scheduler migration and post-upgrade fine tuning, see Manual configuration of scheduler properties on page 177.
- YARN and MapReduce daemons: YARN daemons (ResourceManager, NodeManager) and the JobHistory Server runs with unix group hadoop instead of yarn:yarn and mapred:mapred.
- Cross-Origin Resource Sharing: CORS is enabled for every role by default.
- YARN admin commands: By default YARN admin commands can be run only as yarn. In Cloudera Base on premises 7.1.7 and higher a placeholder value ${yarn_user} is also supported. In such cases Cloudera Manager replace the placeholder value with the collected principal name.
- ResourceManager recovery: ResourceManager recovery is enabled by default.
- Filter entity list by use (filter-entity-list-by-user): Enabled by default, meaning that users can see only those applications on the UI which they have access to.
- Log aggregation: IFile is the default file controller.
- MapReduce shuffle connection keep-alive (mapreduce.shuffle.connection-keep-alive.enabled):: Set to true by default because Auto TLS requires it.
- YARN Admin ACL: If the yarn.admin.acl property is not configured before the upgrade, its default value is changed from * to yarn. In Cloudera Base on premises 7.1.7 and higher a placeholder value ${yarn_user} is also supported. In such cases Cloudera Manager replace the placeholder value with the collected principal name.

Upgrade Notes for Apache Kudu 1.15 / CDP 7.1

Learn about the most important Kudu relatd changes when upgrading to Cloudera Base on premises 7.1.

World-readable Kerberos keytab files

To improve security, world-readable Kerberos keytab files are no longer accepted by default. You can override this behaviour by setting the ##allow_world_readable_credentials property to true using the Kudu Service Advanced Configuration Snippet (Safety Valve) for gflagfile advanced configuration snippet:

```
##allow_world_readable_credentials=true
```

Wire Protocol compatibility

Kudu 1.15.0 is wire-compatible with previous versions of Kudu:

- Kudu 1.15 clients may connect to servers running Kudu 1.0 or later. If the client uses features that are not available on the target server, an error will be returned.
- Rolling upgrade between Kudu 1.9 and Kudu 1.15 servers is believed to be possible though has not been sufficiently tested. Users are encouraged to shut down all nodes in the cluster, upgrade the software, and then restart the daemons on the new version.
- Kudu 1.6 and later clients may connect to servers running Kudu 1.15.

Client Library Compatibility

- The Kudu 1.15 Java client library is API- and ABI-compatible with Kudu 1.6 and later. Applications written against Kudu 1.6 and later will compile and run against the Kudu 1.15 client library and the other way around.
- The Kudu 1.15 C++ client is API- and ABI-forward-compatible with Kudu 1.6 and later. Applications written and compiled against the Kudu 1.6 client library will run without modification against the Kudu 1.15 client library. Applications written and compiled against the Kudu 1.6 or later client library will run without modification against the Kudu 1.15client library.
- The Kudu 1.15Python client is API-compatible with Kudu 1.6 and later. Applications written against Kudu 1.6 and later will continue to run against the Kudu 1.15 client and the other way around.

Integration with Apache Ranger

The integration with Apachy Sentry is replaced by the integration with Apache Ranger. Kudu 1.15 natively integrates with Apache Ranger for fine grain authorization and access control. This integration is disabled by default after the upgrade. If you want to enable fine grain authorization and access control with Kudu and Ranger, follow the steps described in *Enabling Ranger authorization*.

Related Information

Enabling Ranger authorization

Apache HBase post-upgrade tasks

After upgrading from CDH to Cloudera Base on premises, there are certain tasks that you need to perform before you can start using the HBase cluster.

Switch to relying on shaded artifacts

After upgrading from CDH 5 to CDP, the HBase client applications have to switch to hbase-shaded-client and hbase-shaded-mapreduce artifacts as dependencies. Cloudera recommends relying on the Maven coordinates org.apache.hbase:hbase-shaded-client for their runtime use.

Note:

- If you are upgrading from CDH 6, you should already use the shaded artifacts. Ensure that you have switched to hbase-shaded-client and hbase-shaded-mapreduce artifacts as dependencies.
- After upgrading from CDH 5 to CDP, use the updated ports for your HBase client application to access the HBase service properly. To check the latest HBase ports, see *Ports Used by Cloudera Runtime Components*.

Users of HBase's integration for Apache Hadoop MapReduce must switch to relying on the org.apache.hbase:hbase-shaded-mapreduce module for their runtime use. Neither org.apache.hbase:hbase-server nor org.apache.hbase:hbase-shaded-server artifacts are supported anymore.

Note that both artifacts expose some classes in the org.apache.hadoop package space (for example o.a.h.configuration.Configuration) to maintain source compatibility with the public API. Those classes are included so that they can be altered to use the same relocated third party dependencies as the rest of the HBase client code. In the event that you need to also use Hadoop in your code, you should ensure all Hadoop related jars precede the HBase client jar in your classpath.

Configure SMM to monitor SRM replications

Following a successful upgrade, if you want to use SMM to monitor SRM replications, you must reconnect the two services. This is done by enabling the STREAMS_REPLICATION_MANAGER Service SMM property which is disabled by default.

About this task

This configuration is only required if you are upgrading from CDH 5 or CDH 6 to Cloudera Runtime 7.1.6 and higher or from Cloudera Runtime 7.1.5 and lower to Cloudera Runtime 7.1.6 and higher.

Important: SMM can only connect to and monitor an SRM service that is running in the same cluster as SMM. Monitoring an SRM service that is running in a cluster that is external to SMM is no longer supported.

Procedure

1. In Cloudera Manager, select the SMM service.
2. Go to Configuration.
3. Find and enable the STREAMS_REPLICATION_MANAGER Service property.
4. Click Save Changes.
5. Restart the service.

Results

SMM is configured to monitor SRM replications. The Cluster Replications tab is available in the SMM UI.

Configure SMM's service dependency on Schema Registry

Following a successful upgrade, the integration between the SMM and Schema Registry services is disabled. If you have previously enabled integration you must re-enable it following an upgrade. This can be done by selecting the Schema Registry Service checkbox.

About this task

This configuration is only required if you are upgrading from CDH 5 or CDH 6 to Cloudera Runtime 7.1.1 or higher.

Procedure

1. In Cloudera Manager, select the SMM service.
2. Go to Configuration.
3. Find and select the Schema Registry Service property.
4. Click Save Changes.

5. Restart the service.

Apache Sqoop Changes

After upgrading from CDH to CDP, Sqoop action errors are not logged due to an change in the log4j configuration. You must configure Oozie to log Sqoop action errors in the Oozie launcher log.

About this task

Before Upgrade to CDP

The Sqoop action in an Oozie workflow log errors to the Oozie launcher log. For example:

```
>>> Invoking Sqoop command line now >>>
 2021-01-11 09:58:21,438 [main] WARN  org.apache.sqoop.tool.SqoopTool  - $SQ
OOP_CONF_DIR has not been set in the environment. Cannot check for additiona
l configuration.
 2021-01-11 09:58:21,489 [main] INFO  org.apache.sqoop.Sqoop  - Running Sqo
op version: 1.4.7.7.1.5.0-257
 2021-01-11 09:58:21,503 [main] WARN  org.apache.sqoop.tool.BaseSqoopTool  -
 Setting your password on the command-line is insecure. Consider using -P
instead.
 2021-01-11 09:58:21,516 [main] WARN  org.apache.sqoop.ConnFactory  - $SQOO
P_CONF_DIR has not been set in the environment. Cannot check for additional
configuration.
...
```

After Upgrade to CDP

The Sqoop action in an Oozie workflow does not log errors to the Oozie launcher log. For example:

```
>>> Invoking Sqoop command line now >>>
09:39:49.715 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation -
 mapred.jar is deprecated. Instead, use mapreduce.job.jar
09:39:49.738 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation -
 mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
09:39:49.974 [main] INFO  org.apache.hadoop.mapreduce.JobResourceUploader -
 Disabling Erasure Coding for path: /user/cloudera/.staging/job_1609912545
960_0013
09:39:50.347 [main] INFO  org.apache.hadoop.mapreduce.JobSubmitter - Cleani
ng up the staging area /user/cloudera/.staging/job_1609912545960_0013
<<< Invocation of Sqoop command completed <<<
No child hadoop job is executed.
Intercepting System.exit(1)
java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
...
```

Action Required

Configure the Sqoop action to use the log4j1 configuration for this Sqoop action and the Hue workspace only.

Procedure

1. Create a log4j.properties file and upload it to the lib directory in the workflow.xml path inside HDFS.

```
log4j.rootLogger=INFO , A
log4j.logger.org.apache.sqoop=INFO, A
log4j.additivity.org.apache.sqoop=false
log4j.appender.A=org.apache.log4j.ConsoleAppender
```

```
log4j.appender.A.layout=org.apache.log4j.PatternLayout
log4j.appender.A.layout.ConversionPattern=%d [%t] %-5p %c %x - %m%n
log4j.logger.org.apache.sqoop=INFO, A
```

2. In the workflow.xml file or in the Sqoop action in Hue interface, configure the Sqoop action to use the log4j.properties file by configuring SqoopAction > Properties > yarn.app.mapreduce.am.admin-command-opts: Dlog4j.configuration=log4j.properties.

Related Information

Sqoop Documentation 1.4.7.7.1.6.0

Check Parquet writer implementation property

You might need to set the Parquet writer implementation property to hadoop. In releases before CDP 7.1.6/Cloudera Manager 7.3.1, the Parquet writer implementation property (parquetjob.configurator.implementation) default was not hadoop.

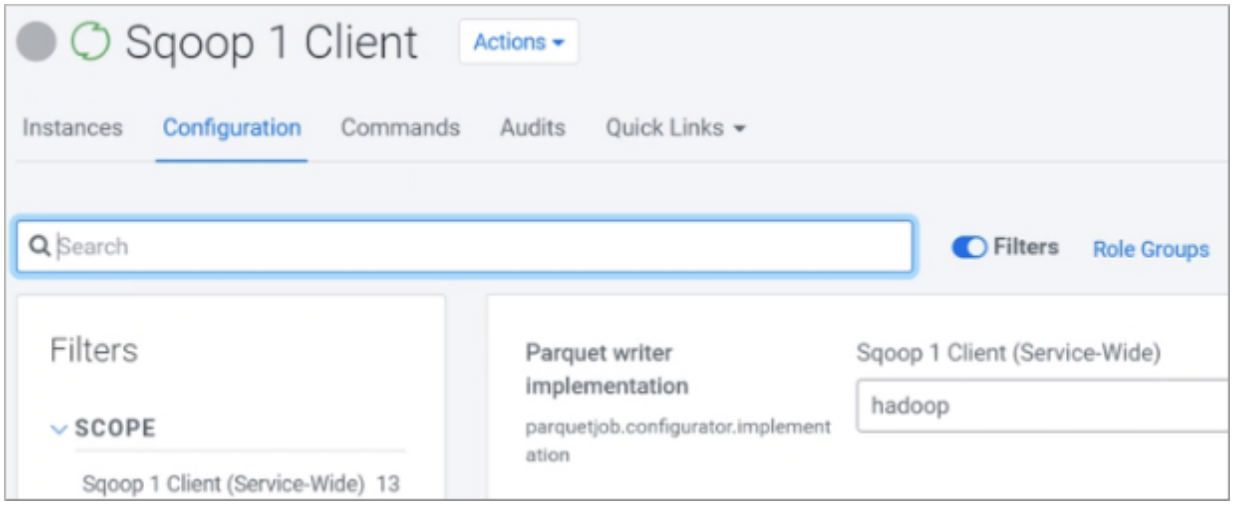
About this task

If you upgraded to a CDP release earlier than CDP 7.1.6, change the default value of the Parquet writer implementation property to hadoop. Making this change prevents encountering the following error when using the Sqoop client:

```
Post upgrade sqoop failed with the error "Invalid Parquet job configurator i
mplementation is set: kite. Supported values are: [HADOOP]" ,
```

Procedure

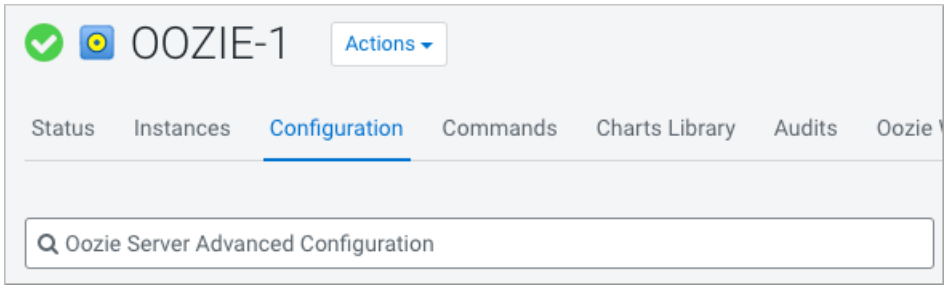
1. In Cloudera Manager, click Clusters Sqoop 1 Client Configuration , and search for Parquet writer implementation.
2. Change the value to hadoop.



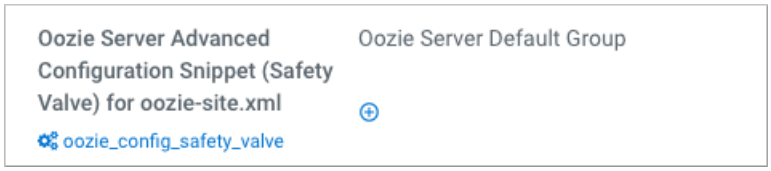
Configure a Sqoop Action globally and for all Hue workspaces

Procedure

1. In Cloudera Manager, click Clusters Oozie-1 Configuration , and search for Oozie Server Advanced Configuration.



2. Scroll down, locate the Oozie Server Advanced Configuration Snippet (Safety Valve) for oozie-site.xml, and click +.



3. Add the following property name and value.

```
<property>
  <name>oozie.service.HadoopAccessorService.action.configurations</name>
  <value>*=/var/lib/oozie/action-conf</value>
</property>
```

4. Login to the server running the Oozie service as root and run the following commands.

```
mkdir -p /var/lib/oozie/action-conf
chown -R oozie:oozie /var/lib/oozie
```

5. Create a sqoop.xml file that configures the yarn.app.mapreduce.am.admin-command-opts property.

```
<configuration>
<property>
<name>yarn.app.mapreduce.am.admin-command-opts</name>
<value>-Dlog4j.configuration=log4j.properties</value>
</property>
</configuration>
```

6. Copy the file to /var/lib/oozie/action-conf and ensure it is owned by oozie:oozie.
7. Copy the log4j.properties file to Oozie shared lib /user/oozie/share/lib/lib_<timestamp>/sqoop and restart Oozie service.

Cloudera Search changes

Learn about the most important Search related changes when upgrading to CDP Private Cloud Base 7.1.
Summary:

Admin API address has changed

Previous behavior:

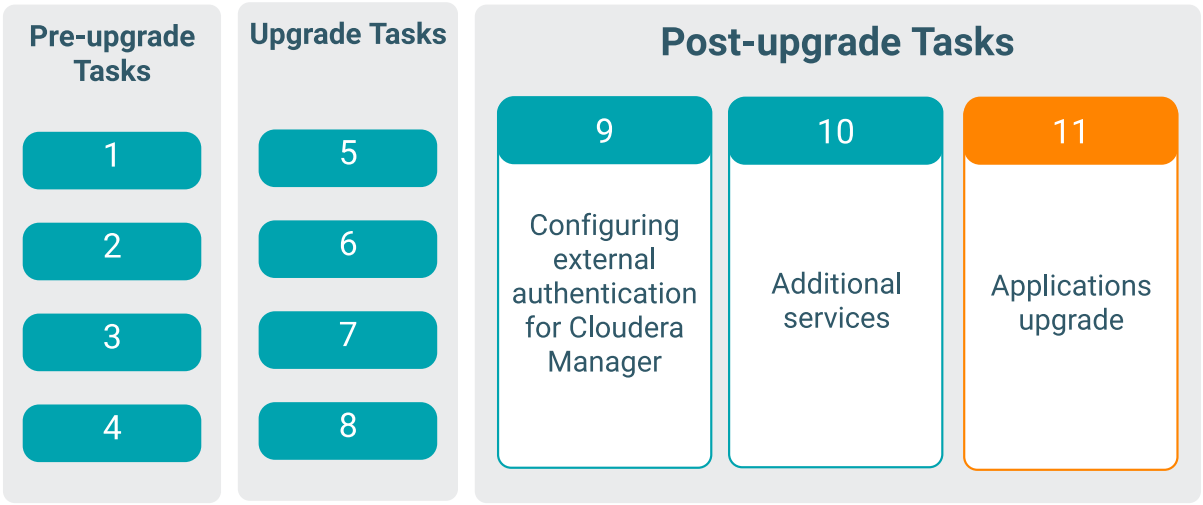
In Solr 7 both curl -k --negotiate -u: "https://`hostname -f`:8985/solr/?op=GETDELEGATION TOKEN" and curl -k --negotiate -u: "https://`hostname -f`:8985/solr/admin?op=GETDELEGATIO NTOKEN" commands worked.

New behavior

In Solr 8 only curl -k --negotiate -u: "https://`hostname -f`:8985/solr/admin?op=GETDELEGATIONTOKEN" command (with the 'admin' string added) works.

Applications Upgrade

After you upgrade, you must test all the services that run on your platform.



Ideally, you should have an indicative subset of jobs from your workloads. These are the tasks that you should have identified and run before the upgrade allowing you to compare pre-upgrade versus post-upgrade test results. These tests should also include any parts of the application that required code changes due to the changes in the platform. For example, to cater for changes in Hive managed versus external tables. The tests should also include a performance test. This can help to highlight missed or wrong configuration settings or point to other issues with the upgrade. Depending on your environment, perform these steps.

1. Update application code with changes required by the upgraded platform
2. Update the dependencies in the pom.xml file for custom jar files used in your applications to use the new dependencies for CDP Private Cloud Base.
3. Restart applications
4. Test the applications and verify they are functioning and performing as they were prior to upgrade

Note: After successfully upgrading from HDP to CDP Private Cloud Base, you can remove the HDP bits from the CDP Private Cloud Base cluster using the yum commands. Otherwise, when you run any security tool or security scanner for CVE detection, the HDP bits on the CDP Private Cloud Base cluster are detected as CVEs.