Cloudera Runtime 7.3.3

Apache Hive Overview

Date published: 2019-08-21
Date modified: 2026-11-13



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

Apache Hive features....4
Spark integration with Hive....5
Hive on Tez introduction....5
Hive unsupported interfaces and features....5
Apache Hive 3 architectural overview....7
Installing Hive on Tez and adding a HiveServer role....8
Adding a HiveServer role....8
Changing the Hive warehouse location....10
Adding a HiveServer role....10
Changing the Hive warehouse location....12
Apache Hive content roadmap....13

Apache Hive features

Major changes to Apache Hive 2.x improve Apache Hive 3.x transactions and security. Knowing the major differences between these versions is critical for SQL users, including those who use Apache Spark and Apache Impala.

Hive is a data warehouse system for summarizing, querying, and analyzing huge, disparate data sets.

ACID transaction processing

Hive 3 tables are ACID (Atomicity, Consistency, Isolation, and Durability)-compliant. Hive 3 write and read operations improve the performance of transactional tables. Atomic operations include simple writes and inserts, writes to multiple partitions, and multiple inserts in a single SELECT statement. A read operation is not affected by changes that occur during the operation. You can insert or delete data, and it remains consistent throughout software and hardware crashes. Creation and maintenance of Hive tables is simplified because there is no longer any need to bucket tables.

Materialized views

Because multiple queries frequently need the same intermediate roll up or joined table, you can avoid costly, repetitious query portion sharing, by precomputing and caching intermediate tables into views.

Query results cache

Hive filters and caches similar or identical queries. Hive does not recompute the data that has not changed. Caching repetitive queries can reduce the load substantially when hundreds or thousands of users of BI tools and web services query Hive.

Scheduled Queries

Using SQL statements, you can schedule Hive queries to run on a recurring basis, monitor query progress, temporarily ignore a query schedule, and limit the number running in parallel. You can use scheduled queries to start compaction and periodically rebuild materialized views, for example.

Security improvements

Apache Ranger secures Hive data by default. To meet demands for concurrency improvements, ACID support, render security, and other features, Hive tightly controls the location of the warehouse on a file system, or object store, and memory resources.

With Apache Ranger and Apache Hive ACID support, your organization will be ready to support and implement GDPR (General Data Protection Regulation).

Connection Pooling

Hive supports HikariCP JDBC connection pooling.

Unsupported features

Cloudera does not support the following features that were available in HDP and CDH platforms:

- Multiple insert overwrite queries that read data from a source table.
- 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.
- CREATE INDEX

Hive builds and stores indexes in ORC or Parquet within the main table, instead of a different table, automatically. Set hive.optimize.index.filter to enable use (not recommended--use materialized views instead). Existing indexes are preserved and migrated in Parquet or ORC to Cloudera during upgrade.

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

Blog: Enabling high-speed Spark direct reader for Apache Hive ACID tables

Hive on Tez introduction

The Cloudera service provides an Apache Hive SQL database that Apache Tez executes.

The Hive on Tez service provides a SQL-based data warehouse system based on Apache Hive 3.x. The enhancements in Hive 3.x over previous versions can improve SQL query performance, security, and auditing capabilities. The Hive metastore (HMS) is a separate service, not part of Hive, not even necessarily on the same cluster. HMS stores the metadata on the backend for Hive, Impala, Spark, and other components.

Apache Tez is the Hive execution engine for the Hive on Tez service, which includes HiveServer (HS2) in Cloudera Manager. MapReduce is not supported. In a Cloudera cluster, if a legacy script or application specifies MapReduce for execution, an exception occurs. Most user-defined functions (UDFs) require no change to run on Tez instead of MapReduce.

With expressions of directed acyclic graphs (DAGs) and data transfer primitives, execution of Hive queries on Tez instead of MapReduce improves query performance. In Cloudera, Tez is usually used only by Hive, and launches and manages Tez AM automatically when Hive on Tez starts. SQL queries you submit to Hive are executed as follows:

- Hive compiles the query.
- Tez executes the query.
- Resources are allocated for applications across the cluster.
- Hive updates the data in the data source and returns query results.

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
- hms-mirror

You can use Data Explorer 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.
- Hive column masking and row-level filtering policies are applied only when you use the Hive Warehouse Connector (HWC) to access Hive from Spark.

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

DispCp causes Hive ACID jobs to fail

Apache Hive 3 architectural overview

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.

Data storage and access control

One of the major architectural changes to support Hive 3 design gives Hive much more control over metadata memory resources and the file system, or object store. The following architectural changes from Hive 2 to Hive 3 provide improved security:

- Tightly controlled file system and computer memory resources, replacing flexible boundaries: Definitive boundaries increase predictability. Greater file system control improves security.
- Optimized workloads in shared files and containers

Hive 3 is optimized for object stores in the following ways:

- Hive uses ACID to determine which files to read rather than relying on the storage system.
- In Hive 3, file movement is reduced from that in Hive 2.
- Hive caches metadata and data agressively to reduce file system operations

The major authorization model for Hive is Ranger. Hive enforces access controls specified in Ranger. This model offers stronger security than other security schemes and more flexibility in managing policies.

This model permits only Hive to access the Hive warehouse.

Transaction processing

You can deploy new Hive application types by taking advantage of the following transaction processing characteristics:

- Mature versions of ACID transaction processing:

  ACID tables are the default table type.

  ACID enabled by default causes no performance or operational overload.
- Simplified application development, operations with strong transactional guarantees, and simple semantics for SQL commands

  You do not need to bucket ACID tables.
- Materialized view rewrites
- Automatic query cache
- Advanced optimizations

Hive client changes

You can use the thin client Beeline for querying Hive from a client. You can run Hive administrative commands from the client. Beeline uses a JDBC connection to Hive to run commands. Hive parses, compiles, and runs operations. Beeline supports many of the command-line options that Hive CLI supported. Beeline does not support hive -e set key=value to configure the Hive Metastore.

You enter supported Hive CLI commands by invoking Beeline using the hive keyword, command option, and command. For example, hive -e set. Using Beeline instead of the thick client Hive CLI, which is no longer supported, has several advantages, including low overhead. Beeline does not use the entire Hive code base. A small number of daemons required to run queries simplifies monitoring and debugging.

Hive enforces allowlist and denylist settings that you can change using SET commands. Using the denylist, you can restrict memory configuration changes to prevent instability. Different Hive instances with different allowlists and denylists to establish different levels of stability.

Apache Hive Metastore sharing

Hive, Impala, and other components can share a remote Hive metastore.

Query execution of batch and interactive workloads

You can connect to Hive using a JDBC command-line tool, such as Beeline, or using an JDBC/ODBC driver with a BI tool, such as Tableau. You configure the settings file for each instance to perform either batch or interactive processing.

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.

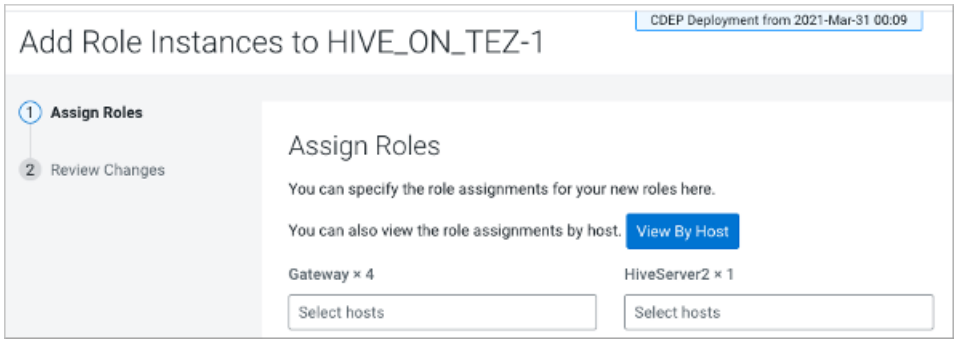
Procedure

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.

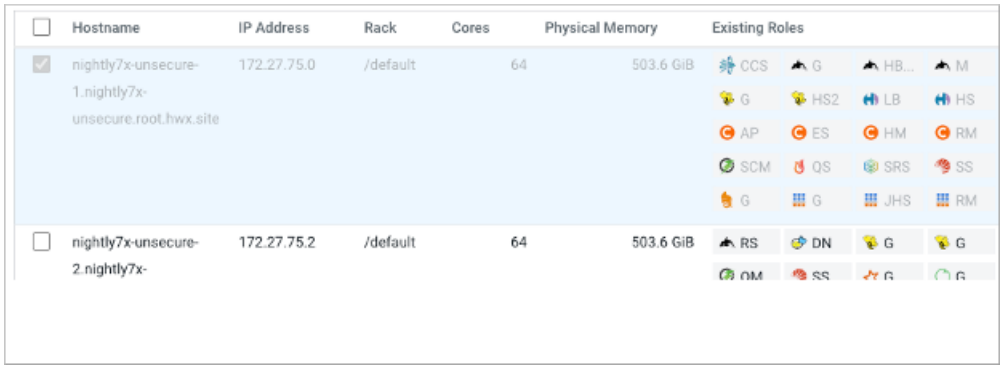
Adding a HiveServer role

Procedure

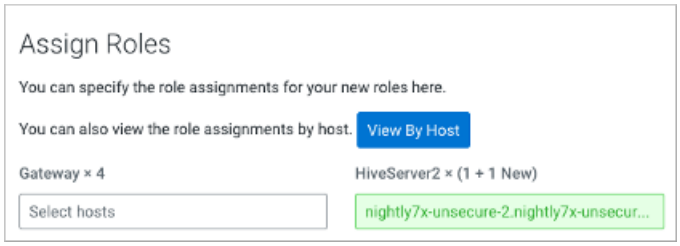
1. In Cloudera Manager, click Clusters Hive on Tez .
   Do not click Clusters Hive by mistake. This selects the Hive metastore and ultimately results in failure.
2. Click Actions Add Role Instances .



3. Click in the HiveServer2 box to select hosts.



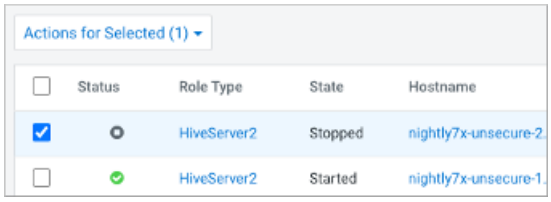
4. In the Host name column, select a host for the HiveServer2 role, and click OK.
The selected host name you assigned the HiveServer2 role appears under HiveServer2.



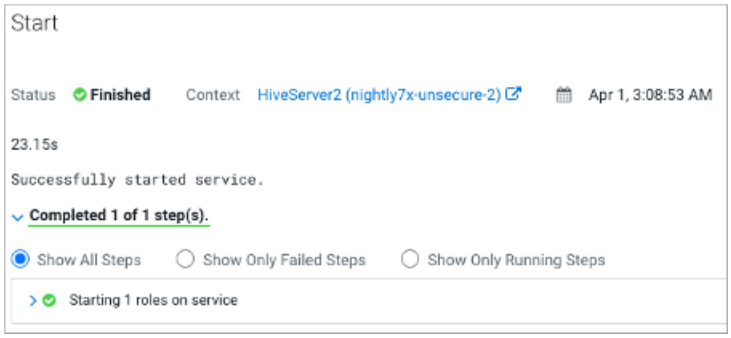
5. Click Continue.

   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

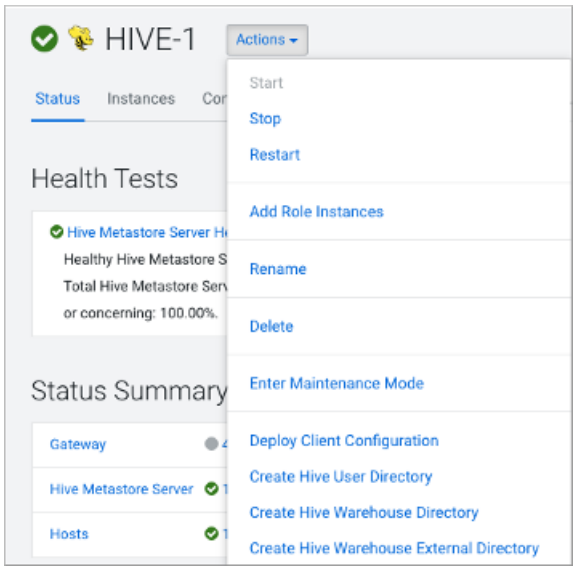
About this task

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.

Adding a HiveServer role

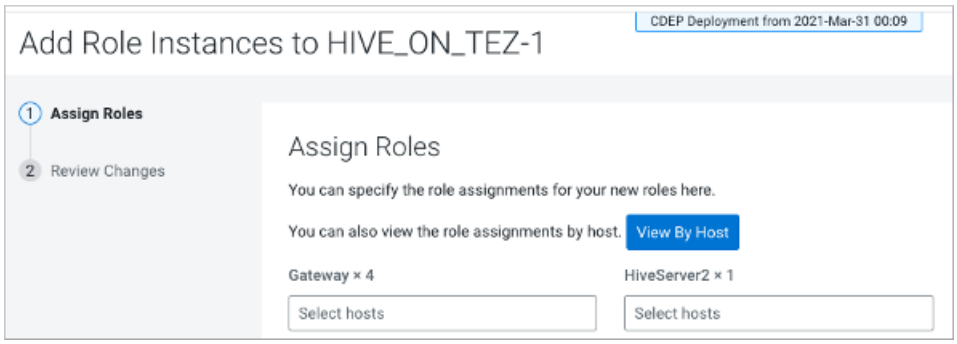
You can add a HiveServer (HS2) role to the Hive-on-Tez service, not to the Hive service.

Before you begin

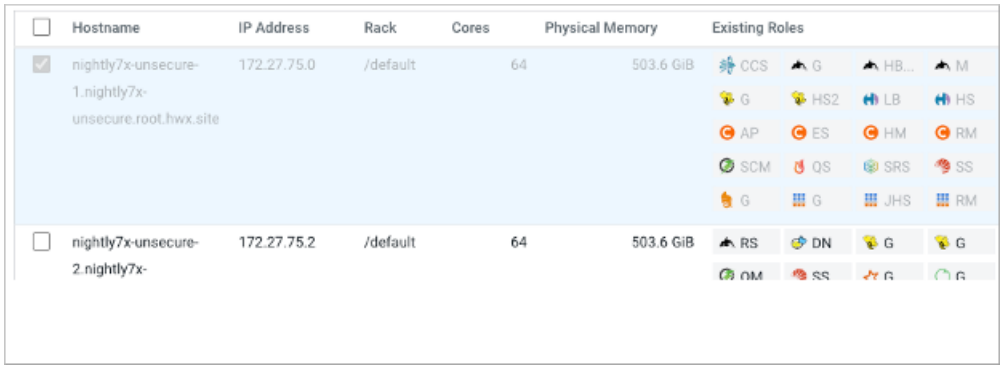
You configured the Hive Delegation Token Store.

Procedure

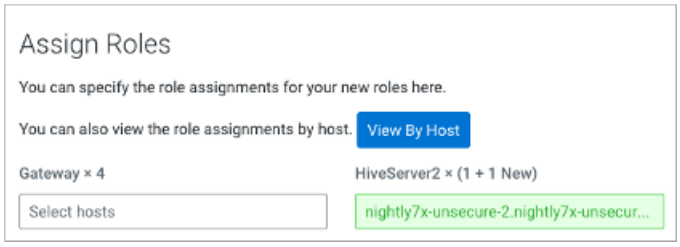
1. In Cloudera Manager, click Clusters Hive on Tez .
   Do not click Clusters Hive by mistake. Only the Hive on Tez service supports the HiveServer2 role.
2. Click Actions Add Role Instances .



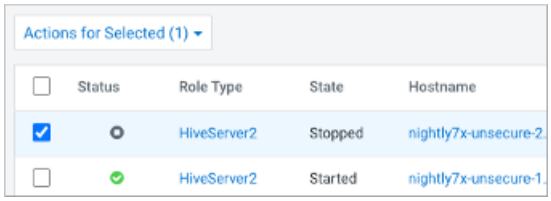
3. Click in the HiveServer2 box to select hosts.



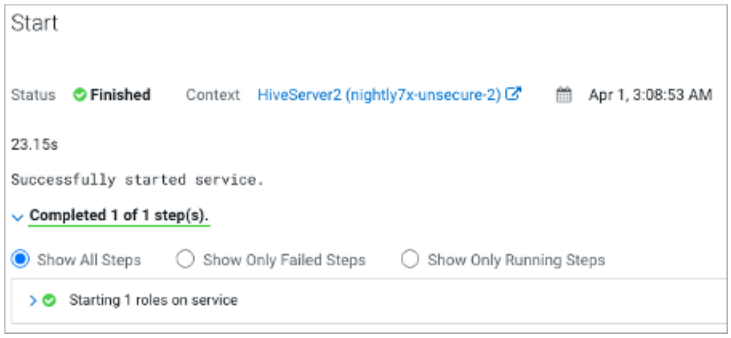
4. In the Host name column, select a host for the HiveServer2 role, and click OK.
   The selected host name you assigned the HiveServer2 role appears under HiveServer2.



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.



Important: After installing Hive on Tez or upgrading the Cloudera Runtime, ensure that you refresh the Tez binaries by **Upload the Tez tar file to HDFS**. This prevents job failures while processing the Tez DAG. To refresh the Tez binaries: Cloudera Manager Tez Actions Upload Tez tar file to HDFS

Changing the Hive warehouse location

You need to know how to use Cloudera Manager to change the location of the Hive warehouse.

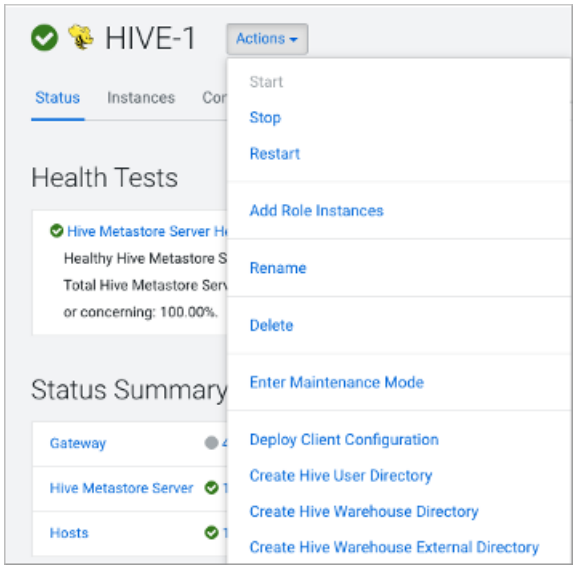
About this task

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.

Apache Hive content roadmap

The content roadmap provides links to the available content resources for Apache Hive.

Table 1: Apache Hive Content roadmap

Task	Resources	Source	Description
Understanding	Presentations and Papers about Hive	Apache wiki	Contains meeting notes, presentations, and whitepapers from the Apache community.
Getting Started	Hive Tutorial	Apache wiki	Provides a basic overview of Apache Hive and contains some examples on working with tables, loading data, and querying and inserting data.
Developing	Materialized Views	Apache wiki	Covers accelerating query processing in Cloudera Data Warehouse by pre-computing summaries using materialized views.
	JdbcStorageHandler	Apache wiki	Describes how to read from a JDBC data source in Hive.
	Hive transactions	Apache wiki	Describes ACID operations in Hive.
	Hive Streaming API	Apache wiki	Explains how to use an API for pumping data continuously into Hive using clients such as NiFi and Flume.
	Hive Operators and Functions	Apache wiki	Describes the Language Manual UDF.
	Beeline: HiveServer2 Client	Apache wiki	Describes how to use the Beeline client.
Reference	SQL Language Manual	Apache wiki	Language reference documentation available in the Apache wiki.
Contributing	Hive Developer FAQ	Apache wiki	Resources available if you want to contribute to the Apache community.
	How to Contribute	Apache wiki	
	Hive Developer Guide	Apache wiki	
	Plug-in Developer Kit	Apache wiki	
	Unit Test Parallel Execution	Apache wiki	
	Hive Architecture Overview	Apache wiki	
	Hive Design Docs	Apache wiki	
	Project Bylaws	Apache wiki	