

Configuring Apache Hive

Date published: 2019-08-21

Date modified: 2021-09-08



Legal Notice

© Cloudera Inc. 2024. 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

Configuring Hive in Cloudera Data Warehouse.....	4
Configuring legacy CREATE TABLE behavior.....	6
Session-level configuration.....	7
Site-level configuration.....	7
Limiting concurrent connections.....	8
Hive on Tez configurations.....	9
Configuring HiveServer high availability using a load balancer.....	10
Configuring the Hive Delegation Token Store.....	10
Adding a HiveServer role.....	11
Configuring the HiveServer load balancer.....	12
Configuring HiveServer high availability using ZooKeeper.....	13
Removing scratch directories.....	13

Configuring Hive in Cloudera Data Warehouse

Configuring Hive performance in the Cloud is rarely, if ever, required relative to the extensive performance tuning typical in a bare metal environment. You might occasionally want to configure Hive properties unrelated to performance, such as HiveServer (HS2) administration or logging level properties.

About this task

Changing one of the vast array of configuration parameters from Cloudera Data Warehouse is recommended only when following Cloudera instructions. For example, you follow instructions in a compaction alert or the product documentation that describes the property and value to change, and how to do it.

In Cloudera Data Warehouse, if you have required permissions, you can configure Hive, Hive metastore, or HiveServer properties from a Database Catalog or Virtual Warehouse. In this task, you see how to change a property and add a property from the Database Catalog, and then from a Virtual Warehouse.

Before you begin

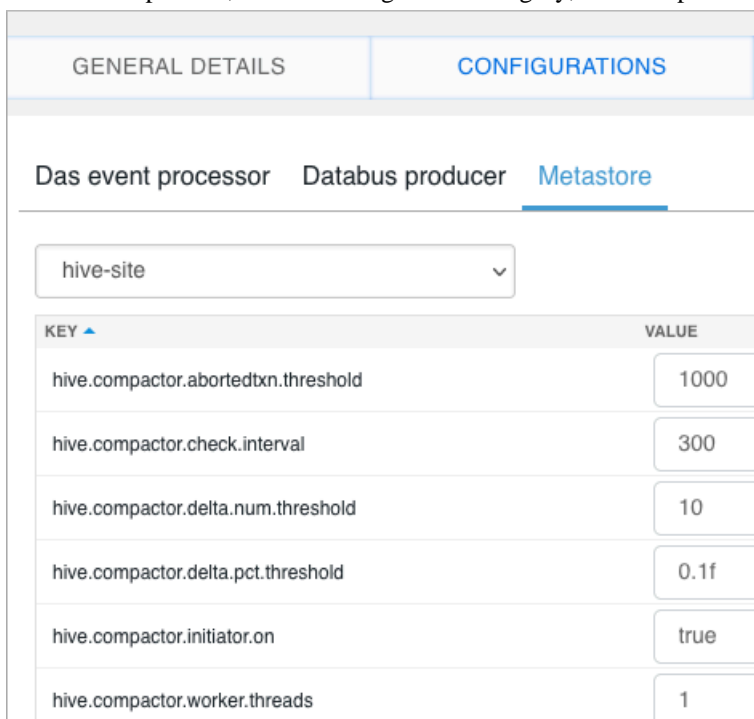
- You have the appropriate Admin role to make the configuration change.

Procedure

1.

Click  CDW Overview options  Edit Configurations Metastore .

2. From the drop-down, select a configuration category, for example hive-site.



GENERAL DETAILS	CONFIGURATIONS
Das event processor Databus producer <u>Metastore</u>	
hive-site	
KEY	VALUE
hive.compactor.abortedtxn.threshold	1000
hive.compactor.check.interval	300
hive.compactor.delta.num.threshold	10
hive.compactor.delta.pct.threshold	0.1f
hive.compactor.initiator.on	true
hive.compactor.worker.threads	1

3. Change, or set, the property value.

4. Click + to add a property to hive-site, for example.

Add Custom Configurations ×

Single and double quotes are not supported yet
Enter key values delimited by =
Comma separated for multiple values

CLOSE ADD

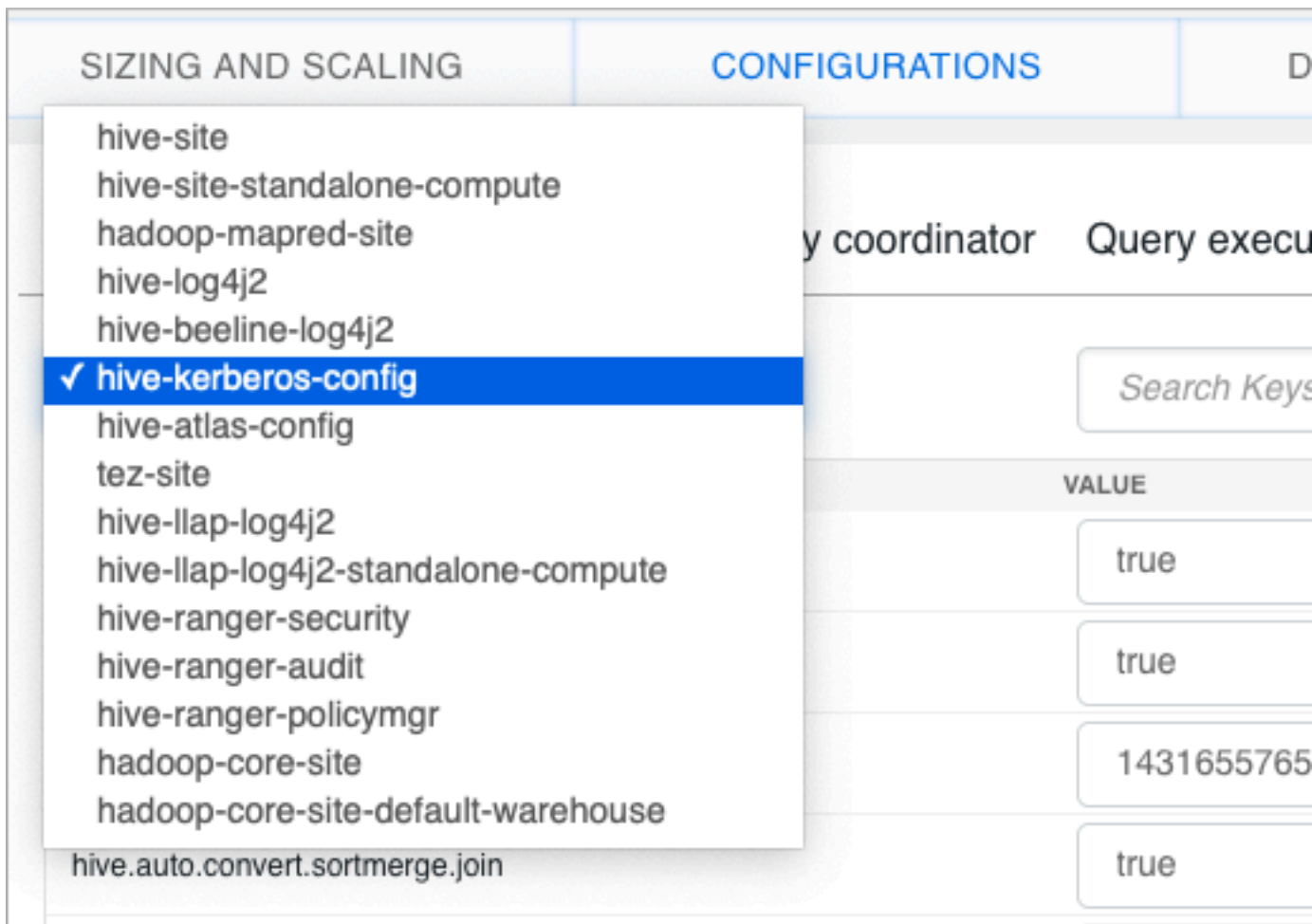
5. Enter the name of the property you want to add, the equals symbol, and one or more comma-separated values.
Do not use single or double quotation marks to enclose the property or value.
6. Click Add.
7. Save, and restart the Database Catalog.
8. Click Virtual Warehouses, and select a Hive Virtual Warehouse.
9. Click Actions Edit Metastore HiveServer2 .

SIZING AND SCALING CONFIGURATIONS

Das webapp Hiveserver2 Hue Query coordinator

hadoop-mapred-site ▼

10. Search for a specific property, or view categories of properties by selecting a category in the dropdown.



Configuring legacy CREATE TABLE behavior

After you upgrade to CDP and migrate old tables, you might want to briefly switch to Hive legacy behavior. Legacy behavior might solve compatibility problems with your scripts during data migration, for example, when running ETL.

About this task

By default, executing a CREATE TABLE statement creates a managed Apache Hive 3 table in the Hive metastore. You can change the default behavior to use the legacy CREATE TABLE behavior. When you configure legacy behavior, CREATE TABLE generates external tables. Legacy behavior is recommended only during upgrading due to the advantages of full ACID transactional tables over external tables.

Apache Hive full ACID (transactional) tables deliver better performance, security, and user experience than non-transactional tables. By default, executing a CREATE TABLE statement creates a managed Apache Hive 3 table in the Hive metastore. Hive 3 tables are ACID-compliant, transactional tables having the following full ACID capabilities on data in ORC format only:

- Insert
- Update
- Delete

Using ACID-compliant, transactional tables causes no performance or operational overload. Bucketing is not necessary.

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 CDP as it did before the upgrade.

Configure legacy CREATE TABLE behavior

When you configure legacy behavior, CREATE TABLE creates an external table in your specified warehouse, which is /warehouse/tablespace/external/hive by default. To configure legacy behavior at the session level, you can pass a property to HiveServer (HS2) in the Beeline connection string when you launch Hive. Alternatively, you can pass the property on the Hive command line to switch to the old behavior. You can also configure legacy create table behavior at the site level by configuring properties in Cloudera Manager. When configured at the site level, legacy behavior persists from session to session.

Related Information

[Change DROP behavior](#)

Session-level configuration

About this task

Step 1 describes two ways of configuring legacy CREATE TABLE behavior. You can override the configured legacy behavior as described in step 2 to create a managed table.

Procedure

1. Choose one of the following ways to configure legacy CREATE TABLE behavior:

- To configure legacy behavior in any JDBC client, include `hiveCreateAsExternalLegacy=true` in the connection string. For example, in Beeline, include the connection string to launch Hive:

```
beeline -u jdbc:hive2://10.65.13.98:10000/default;hiveCreateAsExternalLegacy=true \
-n <your user name> -p
```

- To configure legacy behavior within an existing beeline session, set `hive.create.as.external.legacy=true`. For example:

```
hive> SET hive.create.as.external.legacy=true;
```

You can purge the table from the file system and metastore. You can change the DROP behavior, to remove metadata only.

2. Override the configured legacy behavior at the session level (only) to create a managed table by using the MANAGED keyword.

```
CREATE MANAGED TABLE test (id INT);
```

When your session ends, the create legacy behavior also ends. If you issue a CREATE TABLE statement, Hive creates either an insert-only or full ACID table, depending on how you set the following table properties:

- `hive.create.as.insert.only`
- `hive.create.as.acid`

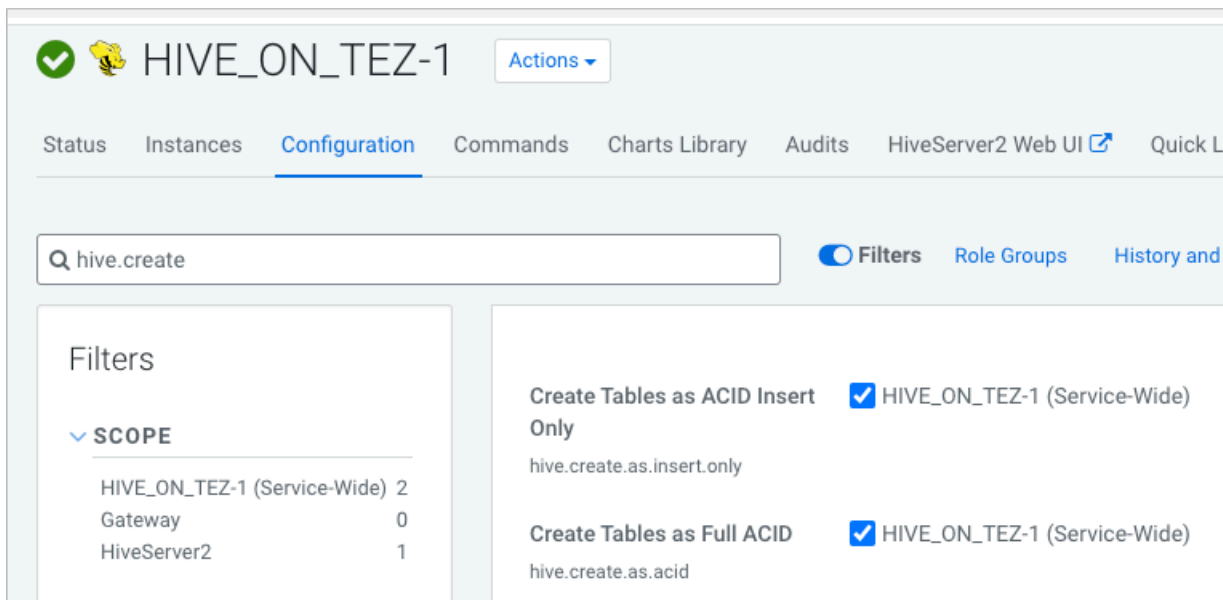
Site-level configuration

About this task

When you configure legacy create table behavior at the site level, the legacy behavior persists from session to session. You configure this behavior at the site level using Cloudera Manager as follows:

Procedure

1. In Cloudera Manager > Clusters > Hive On Tez, search for hive.create.



2. Configure properties in one of the following ways:

- If Create Tables as ACID Insert Only and Create Tables as Full ACID properties appear and are checked, uncheck the properties.
- If your version of Cloudera Manager does not expose these properties, in the HiveServer2 Advanced Configuration Snippet Safety Value for hive-site.xml, add the properties and values.

```
<property>
  <name>hive.create.as.insert.only</name>
  <value>>false</value>
</property>
<property>
  <name>hive.create.as.acid</name>
  <value>>false</value>
</property>
```

Limiting concurrent connections

To prevent a rogue application from repeatedly connecting to and monopolizing HiveServer, you can limit concurrent connections to HiveServer.

About this task

As administrator, you can limit concurrent connections using the Cloudera Manager Safety Valve to add one or more of the following properties to the hive-site.xml configuration file:

hive.server2.limit.connections.per.user

Maximum number of HiveServer concurrent connections per user

hive.server2.limit.connections.per.ipaddress

Maximum number of HiveServer concurrent connections per IP address

hive.server2.limit.connections.per.user.ipaddress

Maximum number of HiveServer concurrent connections per user and IP address combination

The default of each parameter is 0. You can change the value of each parameter to any number. You must configure concurrent connections on the server side; therefore, a `hive --hiveconf` command does not work.

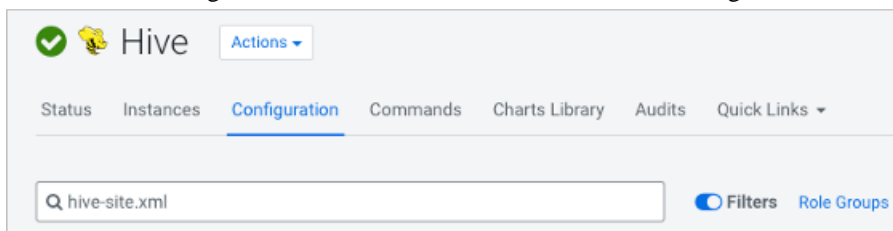
In this task, limit the number of connections per user to 25.

Before you begin

- The following components are running:
 - HiveServer
 - Hive Metastore
 - Hive client
- Minimum Required Role: Configurator (also provided by Cluster Administrator, Full Administrator)

Procedure

1. In Cloudera Manager Clusters select the Hive service. Click Configuration, and search for `hive-site.xml`.



2. In HiveServer2 Advanced Configuration Snippet (Safety Valve) for `hive-site.xml`, click + and add the `hive.server2.limit.connections.per.user` property.
3. Enter a value representing the maximum number of concurrent connections: for example 25.

4. Click Save.
5. Click Actions Deploy Client Configuration .
6. Restart HIVE.

Hive on Tez configurations

Understanding key Hive on Tez properties might help you tune performance or troubleshoot problems, such as running multiple TEZ Application Master (AM) when your default sessions configuration allows running only one. After upgrading, the number of default sessions allowed might be only one. Making Hive on Tez configuration changes is recommended for users who know what they are doing.

Property and Default Value	Description	How to Check and Configure
<code>hive.server2.tez.default.queues</code> (default: "default")	A list of comma separated values corresponding to YARN queues for which to maintain a Tez session pool	Use the Cloudera Manager Safety Valve. When specifying additional queues, they must already exist in YARN.

Property and Default Value	Description	How to Check and Configure
hive.server2.tez.sessions.per.default.queue (default:1)	<p>The number of Tez sessions (DAGAppMaster) to maintain in the pool per YARN queue</p> <p>The total number of concurrent Tez session running can be calculated with:</p> $(\text{Tez Sessions})_{\text{total}} = \text{HiveServer2instances} \times (\text{default.queues}) \times (\text{sessions.per.default.queue})$ <p>The pooled Tez Sessions are always running, even on an idle cluster.</p>	Use the Cloudera Manager Safety Valve. A value of 1 means only one query can run at a time
hive.server2.tez.initialize.default.sessions (default: true)	If enabled, HiveServer (HS2), at startup, will launch all necessary Tez sessions within the specified default.queues to meet the sessions .per.default.queue requirements.	Use the Cloudera Manager Safety Valve.

Related Information

[Custom Configuration \(about Cloudera Manager Safety Valve\)](#)

[Example of using the Cloudera Manager Safety Valve](#)

Configuring HiveServer high availability using a load balancer

To enable high availability for multiple HiveServer (HS2) hosts, you need to know how to configure a load balancer to manage them. First, you configure the Hive Delegation Token Store, next you add HS2 roles, and finally, you configure the load balancer.

About this task

HiveServer HA does not automatically fail and retry long-running Hive queries. If any of the HS2 instances fail, all queries running on that instance fail and are not retried. The client application must re-submit the queries.

After you enable HS2 high availability, ensure that all your clients reflect the load balancer's principal in the connection string. On Kerberos-enabled clusters, you must use the load balancer's principal to connect to HS2 directly; otherwise, after you enable HS2 high availability, direct connections to HS2 instances fail.

Before you begin

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

Configuring the Hive Delegation Token Store

You need to enable Hive Delegation Token Store implementation as the first step in configuring HiveServer high availability using a load balancer. You also need to understand the interaction between Oozie and HS2 with regard to the delegation token.

About this task

Oozie needs this implementation for secure HiveServer high availability (HA). Otherwise, the Oozie server can get a delegation token from one HS2 server, but the actual query might run against another HS2 server, which does not recognize the HS2 delegation token. Exception: If you enable HMS HA, do not enable Hive Delegation Token Store; otherwise, Oozie job issues occur.

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.

Hive Metastore Delegation Token Store
 hive.cluster.delegation.token.store.class
 ⚙️ [hive_metastore_delegation_token_store](#)

Hive Metastore Server Default Group

☐ org.apache.hadoop.hive.thrift.MemoryTokenStore
☒ org.apache.hadoop.hive.thrift.DBTokenStore
☐ org.apache.hadoop.hive.thrift.ZooKeeperTokenStore

[Show All Descriptions](#)

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. Add HiveServer (HS2) roles as described in the next topic.

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** .

Add Role Instances to HIVE_ON_TEZ-1 CDEP Deployment from 2021-Mar-31 00:09

1 Assign Roles
 2 Review Changes

Assign Roles
 You can specify the role assignments for your new roles here.
 You can also view the role assignments by host. [View By Host](#)

Gateway × 4 HiveServer2 × 1

- Click in the HiveServer2 box to select hosts.

<input type="checkbox"/>	Hostname	IP Address	Rack	Cores	Physical Memory	Existing Roles
<input checked="" type="checkbox"/>	nightly7x-unsecure-1.nightly7x-unsecure.root.hwx.site	172.27.75.0	/default	64	503.6 GiB	CCS G G HS2 LB HS AP ES HM RM SCM QS SRS SS G G JHS RM
<input type="checkbox"/>	nightly7x-unsecure-2.nightly7x-unsecure.root.hwx.site	172.27.75.2	/default	64	503.6 GiB	RS DN G G NM SS G G

- 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.

Assign Roles

You can specify the role assignments for your new roles here.

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

Gateway × 4 HiveServer2 × (1 + 1 New)

nightly7x-unsecure-2.nightly7x-unsecure-1.nightly7x-unsecure.root.hwx.site

- Click Continue.
The new HiveServer2 role state is stopped.

- Select the new HiveServer2 role.

Actions for Selected (1)				
<input type="checkbox"/>	Status	Role Type	State	Hostname
<input checked="" type="checkbox"/>		HiveServer2	Stopped	nightly7x-unsecure-2.nightly7x-unsecure.root.hwx.site
<input type="checkbox"/>		HiveServer2	Started	nightly7x-unsecure-1.nightly7x-unsecure.root.hwx.site

- In Actions for Selected, select Start, and then click Start to confirm.
You see that the service successfully started.

Start

Status **Finished** Context [HiveServer2 \(nightly7x-unsecure-2\)](#) Apr 1, 3:08:53 AM

23.15s

Successfully started service.

✓ **Completed 1 of 1 step(s).**

☒ Show All Steps
 ☐ Show Only Failed Steps
 ☐ Show Only Running Steps

> Starting 1 roles on service

Configuring the HiveServer load balancer

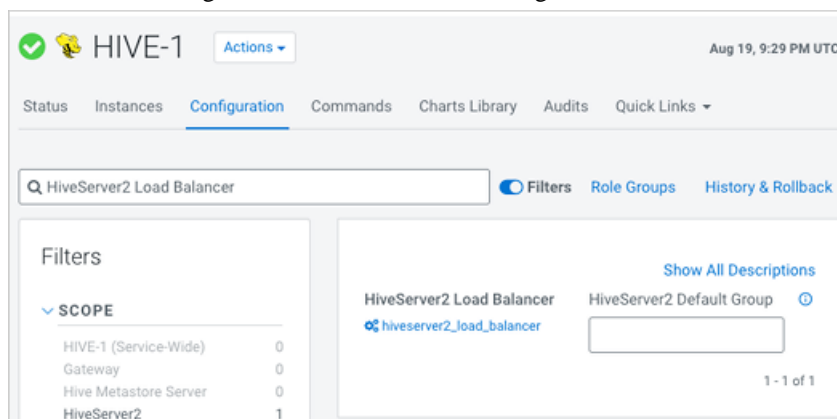
Cloudera Manager exposes the HiveServer load balancer property. You see how to access the property and set it.

Before you begin

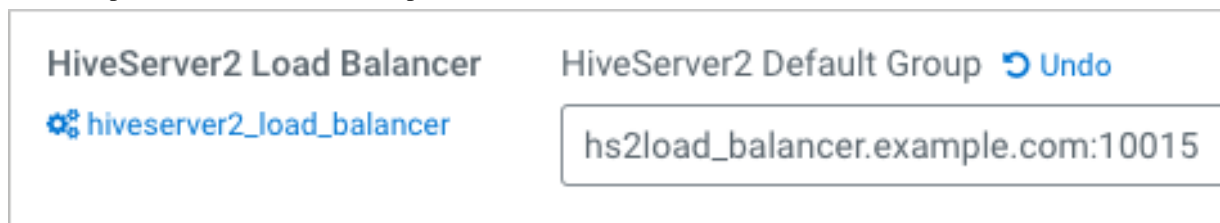
- You configured the Hive Delegation Token Store.
- You added one or more HiveServer roles.

Procedure

1. In Cloudera Manager, click **Clusters Hive Configuration**, and search for **HiveServer2 Load Balancer**.



2. Set the value, using the following format: `<hostname>:<port number>`.
For example, `hs2load_balancer.example.com:10015`.



3. Save the change.

Configuring HiveServer high availability using ZooKeeper

You need to know how to configure your Hive-on-Tez to use ZooKeeper for HiveServer high availability.

When you add one or more additional HiveServer (HS2) role instances to the Hive-on-Tez service, the multiple role instances can register themselves with ZooKeeper. The JDBC client (client driver) can find a HiveServer through ZooKeeper. Using Beeline, you connect to Hive, and the ZooKeeper discovery mechanism locates and connects to one of the running HiveServer instances.

If more than one HiveServer instance is registered with ZooKeeper, and all instances fail except one, ZooKeeper passes the link to the instance that is running and the client can connect successfully. Failed instances must be restarted manually.

Automatic failover does not occur. If an HS2 instance failed while a client is connected, the session is lost. Since this situation needs to be handled at the client, there is no automatic failover; the client needs to reconnect using ZooKeeper.

Using binary transport mode in HiveServer (HS2), Knox, and Dynamic Discovery, possibly supported on your platform before upgrading to CDP, are not supported on CDP. Use alternate solutions, such as HAProxy.

Related Information

[Adding a Role Instance](#)

Removing scratch directories

You need to know how to periodically clear scratch directories used by Apache Hive to prevent problems, such as failing jobs.

About this task

Scratch directories where Hive stores intermediate, or temporary, files accumulate too much data over time and overflow. You can configure Hive to remove scratch directories periodically and without user intervention. Using Cloudera Manager, you add the following properties as shown in the procedure:

hive.start.cleanup.scratchdir

Value: true

Cleans up the Hive scratch directory while starting the HiveServer.

hive.server2.clear.dangling.scratchdir

Value: true

Starts a thread in HiveServer to clear out the dangling directories from the file system, such as HDFS.

hive.server2.clear.dangling.scratchdir.interval

Example Value: 1800s

Procedure

1. In Cloudera Manager, click **Clusters** **Hive on Tez Configuration** . **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 name and in **value** enter the value.