

Cloudera Runtime 7.2.11

Integrating Apache Hive with BI tools

Date published: 2019-08-21

Date modified: 2021-09-08

CLOUDERA

<https://docs.cloudera.com/>

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

Connecting Hive to BI tools using a JDBC/ODBC driver in Cloudera Data

Warehouse.....	4
Getting the JDBC driver.....	4
Getting the ODBC driver.....	4
Downloading a JDBC driver from Cloudera Data Warehouse.....	4
Specify the JDBC connection string.....	6
JDBC connection string syntax.....	6
 Using JdbcStorageHandler to query RDBMS.....	8
 Setting up JdbcStorageHandler for Postgres.....	9

Connecting Hive to BI tools using a JDBC/ODBC driver in Cloudera Data Warehouse

To query, analyze, and visualize data stored in CDP, you use drivers provided by Cloudera to connect Apache Hive to Business Intelligence (BI) tools.

About this task

How you connect to Hive depends on a number of factors: the location of Hive inside or outside the cluster, the HiveServer deployment, the type of transport, transport-layer security, and authentication. HiveServer is the server interface that enables remote clients to run queries against Hive and retrieve the results using a JDBC or ODBC connection.

Before you begin

- Choose a Hive authorization model.
- Configure authenticated users for querying Hive through 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 latest 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.

Downloading a JDBC driver from Cloudera Data Warehouse

To use third-party BI tools, your client users need a JDBC JAR to connect your BI tool and the service. You learn how to download the JDBC JAR to give to your client, and general instructions about how to use the JDBC JAR.

Before you begin

Before you can use your BI tool with the Data Warehouse service:


- You created a Database Catalog.

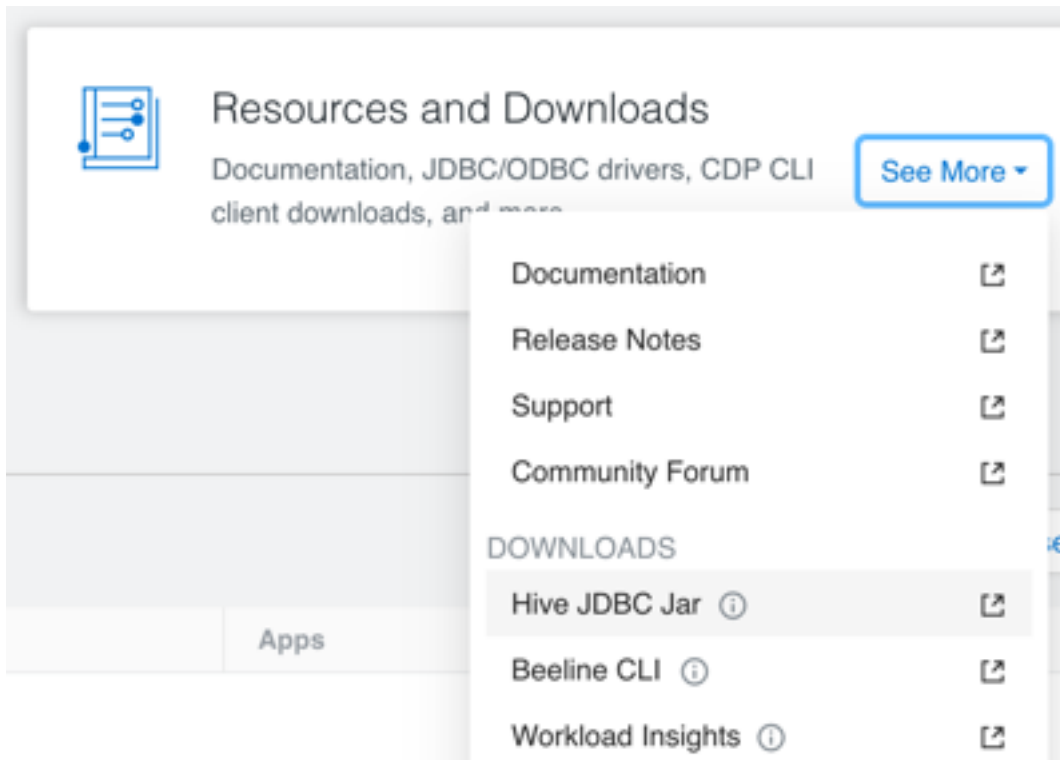
You have the option to populate your Database Catalog with sample data when you create it.


- You created a Virtual Warehouse and configured it to connect to the Database Catalog.

Of course, to query tables in the Virtual Warehouse from a client, you must have populated the Virtual Warehouse with some data.

Procedure

- Log in to the CDP web interface and navigate to the Data Warehouse service.
- In the **Overview** page of the Data Warehouse service, click See More in the Resources and Downloads tile.
- Select Hive JDBC Jar and click  to download the Apache Hive JDBC JAR file.



- Provide the JAR file you downloaded to your JDBC client.
On most clients, add the JAR file under the Libraries folder. Refer to your client documentation for information on the location to add the JAR file.
- In the Data Warehouse service **Overview** page, for the Virtual Warehouse you want to connect to the client, click  and select Copy JDBC URL.

A URL is copied to your system clipboard in the following format:

```
jdbc:hive2://<your_virtual_warehouse>.<your_environment>.<dwx.company.com>/default;transportMode=http;httpPath=cliservice;ssl=true;retries=3
```

- Paste the URL into a text editor and configure your client BI tool to connect to the Virtual Warehouse using the following portion of the URL, represents the server name of the Virtual Warehouse:

```
<your_virtual_warehouse>.<your_environment>.<dwx.company.com>
```

- In your JDBC client, set the following other options:

Authentication: Username and Password

Username: Username you use to connect to the CDP Data Warehouse service.

Password: Password you use to connect to the CDP Data Warehouse service.

Specify the JDBC connection string

You construct a JDBC URL to connect Hive to a BI tool.

About this task

In CDP Private Cloud Base, if HiveServer runs within the Hive client (embedded mode), not as a separate process, the URL in the connection string does not need a host or port number to make the JDBC connection. If HiveServer does not run within your Hive client, the URL must include a host and port number because HiveServer runs as a separate process on the host and port you specify. The JDBC client and HiveServer interact using remote procedure calls using the Thrift protocol. If HiveServer is configured in remote mode, the JDBC client and HiveServer can use either HTTP or TCP-based transport to exchange RPC messages.

Procedure

1. Create a minimal JDBC connection string for connecting Hive to a BI tool.
 - Embedded mode: Create the JDBC connection string for connecting to Hive in embedded mode.
 - Remote mode: Create a JDBC connection string for making an unauthenticated connection to the Hive default database on the localhost port 10000.

Embedded mode: "jdbc:hive://"

Remote mode: "jdbc:hive://myserver:10000/default", "", """);

2. Modify the connection string to change the transport mode from TCP (the default) to HTTP using the transportMode and httpPath session configuration variables.

```
jdbc:hive2://myserver:10000/default;transportMode=http;httpPath=myendpoint.com;
```

You need to specify httpPath when using the HTTP transport mode. <http_endpoint> has a corresponding HTTP endpoint configured in [hive-site.xml](#).

3. Add parameters to the connection string for Kerberos authentication.

```
jdbc:hive2://myserver:10000/default;principal=prin.dom.com@APRINCIPAL.DOM.COM
```

JDBC connection string syntax

The JDBC connection string for connecting to a remote Hive client requires a host, port, and Hive database name. You can optionally specify a transport type and authentication.

```
jdbc:hive2://<host>:<port>/<dbName>;<sessionConfs>?<hiveConfs>#<hiveVars>
```

Connection string parameters

The following table describes the parameters for specifying the JDBC connection.

JDBC Parameter	Description	Required
host	The cluster node hosting HiveServer.	yes
port	The port number to which HiveServer listens.	yes
dbName	The name of the Hive database to run the query against.	yes
sessionConfs	Optional configuration parameters for the JDBC/ODBC driver in the following format: <key1>=<value1>;<key2>=<key2>...;	no

JDBC Parameter	Description	Required
hiveConfs	Optional configuration parameters for Hive on the server in the following format: <key1>=<value1>;<key2>=<key2>; ... The configurations last for the duration of the user session.	no
hiveVars	Optional configuration parameters for Hive variables in the following format: <key1>=<value1>;<key2>=<key2>; ... The configurations last for the duration of the user session.	no

TCP and HTTP Transport

The following table shows variables for use in the connection string when you configure HiveServer. The JDBC client and HiveServer can use either HTTP or TCP-based transport to exchange RPC messages. Because the default transport is TCP, there is no need to specify transportMode=binary if TCP transport is desired.

transportMode Variable Value	Description
http	Connect to HiveServer2 using HTTP transport.
binary	Connect to HiveServer2 using TCP transport.

The syntax for using these parameters is:

```
jdbc:hive2://<host>:<port>/<dbName>;transportMode=http;httpPath=<http_endpoint>; \
<otherSessionConfs>?<hiveConfs>#<hiveVars>
```

User Authentication

If configured in remote mode, HiveServer supports Kerberos, LDAP, Pluggable Authentication Modules (PAM), and custom plugins for authenticating the JDBC user connecting to HiveServer. The format of the JDBC connection URL for authentication with Kerberos differs from the format for other authentication models. The following table shows the variables for Kerberos authentication.

User Authentication Variable	Description
principal	A string that uniquely identifies a Kerberos user.
saslQop	Quality of protection for the SASL framework. The level of quality is negotiated between the client and server during authentication. Used by Kerberos authentication with TCP transport.
user	Username for non-Kerberos authentication model.
password	Password for non-Kerberos authentication model.

The syntax for using these parameters is:

```
jdbc:hive2://<host>:<port>/<dbName>;principal=<HiveServer2_kerberos_principal>; \
<otherSessionConfs>?<hiveConfs>#<hiveVars>
```

Transport Layer Security

HiveServer2 supports SSL and Sasl QOP for transport-layer security. The format of the JDBC connection string for SSL uses these variables:

SSL Variable	Description
ssl	Specifies whether to use SSL
sslTrustStore	The path to the SSL TrustStore.
trustStorePassword	The password to the SSL TrustStore.

The syntax for using the authentication parameters is:

```
jdbc:hive2://<host>:<port>/<dbName>; \
ssl=true;sslTrustStore=<ssl_truststore_path>;trustStorePassword=<truststo
re_password>; \
<otherSessionConfs>?<hiveConfs>#<hiveVars>
```

When using TCP for transport and Kerberos for security, HiveServer2 uses Sasl QOP for encryption rather than SSL.

Sasl QOP Variable	Description
principal	A string that uniquely identifies a Kerberos user.
saslQop	The level of protection desired. For authentication, checksum, and encryption, specify auth-conf. The other valid values do not provide encryption.

The JDBC connection string for Sasl QOP uses these variables.

```
jdbc:hive2://fqdn.example.com:10000/default;principal=hive/_H
OST@EXAMPLE.COM;saslQop=auth-conf
```

The `_HOST` is a wildcard placeholder that gets automatically replaced with the fully qualified domain name (FQDN) of the server running the HiveServer daemon process.

Using JdbcStorageHandler to query RDBMS

Using the JdbcStorageHandler, you can connect Apache Hive to a MySQL, PostgreSQL, Oracle, DB2, or Derby data source. You can then create an external table to represent the data, and query the table.

About this task

This task assumes you are a CDP Private Cloud Base user. You create an external table that uses the JdbcStorageHandler to connect to and read a local JDBC data source.

Procedure

1. Load data into a supported SQL database, such as MySQL, on a node in your cluster, or familiarize yourself with existing data in the your database.
2. Create an external table using the JdbcStorageHandler and table properties that specify the minimum information: database type, driver, database connection string, user name and password for querying hive, table name, and number of active connections to Hive.

```
CREATE EXTERNAL TABLE mytable_jdbc(
  col1 string,
  col2 int,
  col3 double
)
STORED BY 'org.apache.hive.storage.jdbc.JdbcStorageHandler'
TBLPROPERTIES (
```



```
"hive.sql.database.type" = "MYSQL",
"hive.sql.jdbc.driver" = "com.mysql.jdbc.Driver",
"hive.sql.jdbc.url" = "jdbc:mysql://localhost/sample",
"hive.sql.dbcp.username" = "hive",
"hive.sql.dbcp.password" = "hive",
"hive.sql.table" = "MYTABLE",
"hive.sql.dbcp.maxActive" = "1"
);
```

3. Query the external table.

```
SELECT * FROM mytable_jdbc WHERE col2 = 19;
```

Setting up JdbcStorageHandler for Postgres

If you use Enterprise PostgreSQL as the backend HMS database, you need to put the JdbcStorageHandler JAR in a central place.

About this task

The Postgres Enterprise server comes with its own JDBC driver. The driver file is installed in the Hive lib directory. When you run a query as a YARN application, the Class not found exception is thrown on worker nodes. The YARN container cannot include the jar file in the classpath unless you place the JAR in a central location.

Place the JAR in aux jars or provide the path to aux jars.

Procedure

1. In CDP Private Cloud Base, click **Cloudera Manager Clusters** and select the Hive service, for example, HIVE.
2. Click **Configuration** and search for **Hive Auxiliary JARs Directory**.
3. Specify a directory value for the **Hive Aux JARs** property if necessary, or make a note of the path.
4. Upload the JAR to the specified directory on all HiveServer instances.