

Cloudera Runtime ..

## Sqoop Troubleshooting

Date published: 2023-05-16

Date modified:

# 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

<b>Unable to read Sqoop metastore created by an older HSQLDB version.....</b>	<b>4</b>
<b>Merge process stops during Sqoop incremental imports.....</b>	<b>5</b>
<b>Sqoop Hive import stops when HS2 does not use Kerberos authentication.....</b>	<b>6</b>

## Unable to read Sqoop metastore created by an older HSQLDB version

After upgrading to certain Cloudera Data Platform (CDP) versions, you may encounter issues in reading the Sqoop metastore that was created using an older version of HyperSQL Database (HSQLDB). Learn how to upgrade the Sqoop metastore to resolve this issue.

Cloudera upgraded the HSQLDB dependency in the following versions to the newest HSQLDB 2.7.1 version:

- CDP Private Cloud Base 7.1.7 SP2
- CDP Private Cloud Base 7.1.8 Cumulative hotfix 4
- CDP Public Cloud 7.2.16

The version upgrade causes incompatibility issues in Sqoop jobs that are stored in HSQLDB. Therefore, the Sqoop metastore should be upgraded after you upgrade to the above-mentioned CDP versions or higher.



**Note:** Sqoop stores its metastore files in the current user's home directory. If the Sqoop metastore is on-demand, then its location is `~/.sqoop/metastore.db`. If the Sqoop metastore is a service, then its location is `~/sqoop/shared-metastore.db`.

Choose one of the following procedures based on how the Sqoop metastore is created:

### For Sqoop metastore created using Sqoop

Perform the following steps if the Sqoop metastore is created using Sqoop:

1. If the Sqoop metastore is started as a service then stop the service by running this command on the host where the service is running:

```
/opt/cloudera/parcels/CDH/lib/sqoop/stop-metastore.sh -p <path to Sqoop metastore pid file>
```

2. Download HSQLDB 2.3.6 and SqlTool JAR files to the host by running the following commands:

```
wget https://repol.maven.org/maven2/org/hsqldb/sqltool/2.3.6/sqltool-2.3.6.jar
wget https://repol.maven.org/maven2/org/hsqldb/hsqldb/2.3.6/hsqldb-2.3.6.jar
```

3. Run the following command to upgrade and convert the Sqoop metastore database files using SqlTool:

For on-demand Sqoop metastore:

```
$JAVA_HOME/bin/java -cp "sqltool-2.3.6.jar:hsqldb-2.3.6.jar" org.hsqldb.cmdline.SqlTool --driver org.hsqldb.jdbc.JDBCdriver --inlineRc="url=jdbc:hsqldb:file:~/.sqoop/metastore.db;shutdown=true;user=SA,password=" --sql="SELECT * FROM SQOOP_ROOT;"
```

For Sqoop metastore as a service:

```
$JAVA_HOME/bin/java -cp "sqltool-2.3.6.jar:hsqldb-2.3.6.jar" org.hsqldb.cmdline.SqlTool --driver org.hsqldb.jdbc.JDBCdriver --inlineRc="url=jdb
```

```
c:hsqldb:file:~/ .sqoop/shared-metastore.db;shutdown=true;user=SA,password=" --sql="SELECT * FROM SQOOP_ROOT;"
```

The Sqoop metastore is upgraded and the database files are converted to a format that can easily be read by HSQLDB 2.7.1.



**Note:** As part of the CDP upgrade, Apache Sqoop installation has the dependent HSQLDB 2.7.1 version by default.

4. If the Sqoop metastore service was stopped earlier, restart the service:

```
/opt/cloudera/parcels/CDH/lib/sqoop/start-metastore.sh -p <path to Sqoop metastore pid file> -l <path to Sqoop metastore logs dir>
```

5. You can choose to remove the downloaded HSQLDB JAR files from the host.

### For Sqoop metastore created in a separate HSQLDB instance

Perform the following steps if the Sqoop metastore is created in a separate HSQLDB instance:

1. Login to the HSQLDB service host and stop the service. For more information, see the HSQLDB documentation.
2. Work with your IT team to determine the HSQLDB store file that is created to store the Sqoop metastore information.
3. Download HSQLDB 2.3.6 and SqlTool JAR files to the host by running the following commands:

```
wget https://repo1.maven.org/maven2/org/hsqldb/sqltool/2.3.6/sqltool-2.3.6.jar
wget https://repo1.maven.org/maven2/org/hsqldb/hsqldb/2.3.6/hsqldb-2.3.6.jar
```

4. Run the following command to upgrade and convert the Sqoop metastore database files using SqlTool:

```
$JAVA_HOME/bin/java -cp "sqltool-2.3.6.jar:hsqldb-2.3.6.jar" org.hsqldb.cmdline.SqlTool --driver org.hsqldb.jdbc.JDBCdriver --inlineRc="url=jdbc:c:hsqldb:file:<PATH OF SQOOP METASTORE FILE>;shutdown=true;user=SA,password=" --sql="SELECT * FROM SQOOP_ROOT;"
```

5. If the HSQLDB service hosts multiple databases then each of the store files must be upgraded as described in the previous step.

The Sqoop metastore is upgraded and the database files are converted to a format that can easily be read by HSQLDB 2.7.1.



**Note:** As part of the CDP upgrade, Apache Sqoop installation has the dependent HSQLDB 2.7.1 version by default.

6. Restart the HSQLDB service. For more information, see the HSQLDB documentation.
7. You can choose to remove the downloaded HSQLDB JAR files from the host.

### Related Information

[HyperSQL Documentation](#)

## Merge process stops during Sqoop incremental imports

During Sqoop incremental import operations, if the target directory is located outside of Hadoop Distributed File System (HDFS), such as in Amazon S3 or Azure Blob Storage, the merge phase of the import process does not take effect.

### Condition

Sqoop, by default, creates temporary directories within HDFS. However, you must be aware of certain considerations in choosing the target directory location while working with Sqoop's incremental import modes. By default, Sqoop operates seamlessly when the target directory resides within HDFS. However, the merge phase of the import process does not take effect outside the box if the target directory is located outside of HDFS.

### Cause

During an import operation, Sqoop generally imports data to a target directory. If this target directory is a non-HDFS location, the merge process tries to acquire the temporary directory required for the merge on the same non-HDFS file system. Since Sqoop creates the temporary directory in HDFS by default, the merge process checks if the temporary directory exists in the target directory's file system and when it does not find it, the merge process simply stops.

### Solution

If the target directory is present outside of HDFS, you must modify the default path of the temporary directory by adding the `--temporary-rootdir` Sqoop option and pointing to a path on the same file where the target directory is located. By aligning the temporary directory path with the file system of the target directory, Sqoop can effectively complete the import process.

#### Example:

Include the `--temporary-rootdir` Sqoop option as shown below:

```
sqoop-import --connect jdbc:mysql://.../transaction --username [***USER NAME ***] --table [***TABLE NAME***] --password [***PASSWORD***] --target-dir abfs://foo@bar/targetdir -m 1 --temporary-rootdir abfs://foo@bar/_sqoop
```

## Sqoop Hive import stops when HS2 does not use Kerberos authentication

Learn how to resolve the issue related to Sqoop Hive imports when either LDAP authentication or no authentication mechanism is enabled for the cluster.

### Condition

When running Sqoop commands to import data into Hive from either the CLI or Oozie, the import job stops after the Sqoop import is done and while trying to connect to HiveServer (HS2) through JDBC. The following log is displayed and you will notice that the job stops on the last line:

```
23/07/24 18:10:17 INFO hive.HiveImport: Loading uploaded data into Hive
23/07/24 18:10:17 INFO hive.HiveImport: Collecting environment variables which need to be preserved for beeline invocation
...
23/07/24 18:10:20 INFO hive.HiveImport: SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
23/07/24 18:10:21 INFO hive.HiveImport: Connecting to jdbc:hive2://HOSTNAME/default;serviceDiscoveryMode=zooKeeper;ssl=true;sslTrustStore=/var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks;trustStorePassword=changeit;zooKeeperNamespace=hiveserver2
```

### Cause

This issue occurs when Kerberos is not used in the JDBC connection string, which Sqoop uses to connect to HS2. The issue affects unsecure clusters and clusters where LDAP authentication is enabled, and the `beeline-site.xml` configuration file does not use Kerberos authentication.

The underlying issue is that Beeline prompts for the username and password for a successful connection and since the Sqoop Hive import is a non-interactive session, you are unable to provide the credentials and therefore the import job stops.

## Solution

### Procedure

Perform one of the following steps to resolve this issue:

#### If...

**No authentication is enabled for the cluster**

#### Then...

Include the `--hs2-url` option in the Sqoop import command and provide the JDBC connection string.

```
--hs2-url <HS2 JDBC string>
```

This allows for a successful connection without prompting for the credentials.

**LDAP authentication is enabled for the cluster**

Include the `--hs2-user` and `--hs2-password` options in the Sqoop import command and provide the credentials.

```
--hs2-user <username>  
--hs2-password <password>
```