

Replicating Data

Date published: 2020-08-14

Date modified: 2022-09-29



Legal Notice

© Cloudera Inc. 2025. 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

- Cloudera Operational Database data replication.....4**
 - Configuring replication for Apache HBase clusters..... 4
 - Configuring replication with Apache HBase in EMR.....6
 - Using the CldrCopyTable utility to copy data.....7
 - Using HashTable and SyncTable tools to copy data between Cloudera Operational Database clusters.....8

Cloudera Operational Database data replication

Cloudera Operational Database enables you to replicate your data by making a copy of the data stored in your Cloudera Operational Database instance and store them in other locations to help you in disaster recovery scenarios.

Cloudera provides a replication plugin that allows you to securely enable data replication for Apache HBase data in Cloudera Data Hub and Cloudera Operational Database experience. The replication plugin simplifies SSL authenticated data replication between Apache HBase clusters in the cloud.

Cloudera Operational Database supports the following data replication scenarios:

- CDH or HDP to Cloudera Operational Database using the replication plugin
- Cloudera on premises Base to Cloudera Operational Database using Cloudera Replication Manager
- Cloudera Operational Database to Cloudera Operational Database using Cloudera Replication Manager
- External services like EMR that has Apache HBase using the replication plugin

Related Information

[HBase Migration through Cloudera Replication Manager](#)

Configuring replication for Apache HBase clusters

You can configure replication between two Apache HBase clusters.

Before you begin

Ensure that you have root access on your source HDP, CDH, or any other Apache HBase cluster.

Ensure that you have noted down your workload user name and password. To check your workload user name, navigate to the Cloudera Management Console > User Management > Users, find your user, and then find your Workload User Name. The workload user name is typically `srv_***WORKLOAD USER NAME***`.

About this task

You need the replication plugin to enable data replication between the clusters. A custom replication endpoint allows the Apache HBase clusters to specify a different Simple Authentication and Security Layer (SASL) ticket when establishing the connection with a remote cluster. Use the following instructions to configure data replication for the clusters.

Procedure

1. Create a machine user in your Cloudera environment that has your Cloudera Operational Database instances. This machine user is specific for your replication use case.
Note down the machine user name. You will need this information later.
2. Add your created machine user with `environmentUser` role in the destination environment using the [Assign environment role](#).
3. Do FreeIPA sync for the destination environment and wait for the sync to complete.
The sync can take about 5 to 15 minutes depending upon the number of servers and users in the environment.
4. Verify if the credentials are correct by running the following kinit command on any node of the destination environment.

```
kinit [***WORKLOAD USER NAME***]
```

- On any environment, generate the keystore using the following command and passing your workload password as a parameter.

```
sudo -u hbase hbase com.cloudera.hbase.security.token.CldrReplicationSecurityTool -sharedkey cloudera -password [***WORKLOAD PASSWORD***] -keystore localjceks://file/[***PATH***]/[CREDENTIALS.JCEKS]
```

- Add the generated keystore to HDFS on both the clusters using the following commands

```
kinit -kt /var/run/cloudera-scm-agent/process/`ls -1 /var/run/cloudera-scm-agent/process/ | grep -i "namenode\|datanode" | sort -n | tail -1`/hdfs.keytab hdfs/${hostname -f}
```

```
hdfs dfs -mkdir /hbase-replication
hdfs dfs -put /[***PATH***]/[CREDENTIALS.JCEKS] /hbase-replication
hdfs dfs -chown -R hbase:hbase /hbase-replication
```

- Configure the following properties in Cloudera Manager.

- Log in to Clouder Manager as an Administrator.
- Go to HBase Configuration and search for the property HBase Service Advanced Configuration Snippet (Safety Valve) for hbase-site.xml.
- Set the values for a Keystore file on HDFS and workload username that matches the CSSO Keystore setup by FreeIPA.

```
<property>
  <name>hbase.security.replication.credential.provider.path</name>
  <value>cdprepjceks://hdfs@[***NAMENODE_HOST***]:[***NAMENODE_PORT***]/hbase-replication/credentials.jceks</value>
</property>
<property>
  <name>hbase.security.replication.user.name</name>
  <value>srv_[***WORKLOAD USER NAME***]</value>
</property>
```

- Restart your cluster for the changes to take effect.
- Using the HBase shell define replication peer at source cluster, specifying the pluggable replication endpoint implementation and zookeeper_quorum in the following format: zk1:zk2:zk3:2181:/hbase.

```
hbase shell
add_peer '1',
ENDPOINT_CLASSNAME => 'com.cloudera.hbase.replication.CldrHBaseInterClusterReplicationEndpoint',
CLUSTER_KEY => '[***ZOOKEEPER_QUORUM***]:2181:/hbase'
```

What to do next

- Set the REPLICATION_SCOPE for the selected table and column family. Note that enable_table_replication is not supported.

```
hbase shell alter 'my-table', {NAME=>'cf', REPLICATION_SCOPE => '1'}
```

- Use replication plugin validation tool to make sure the setup is working. On one of the source cluster hosts, run the following command:

```
hbase com.cloudera.hbase.replication.CldrReplicationPluginValidator
[***PEER_ID***]
```

Configuring replication with Apache HBase in EMR

You can configure your Cloudera Operational Database experience for data replication with an Amazon EMR cluster with Apache HBase.

Before you begin

- Ensure that you have the replication plugin. Contact your Cloudera account team to get the replication plugin.
- Ensure that all the EC2 instances in the EMR cluster can communicate with Cloudera Operational Database. For example, you can configure this by placing the EMR cluster on the same VPC network and subnets used by the Cloudera Operational Database instance.
- Ensure that your Cloudera Operational Database cluster security group allows inbound TCP connections to ports 16020, 16010, and 2181 from all the EC2 instances in the EMR cluster. You can configure this using the AWS Cloudera Management Console. The port configuration is automatically done if the EMR EC2 instances are configured with the same worker, leader, and controller (also known as master) security groups from Cloudera Operational Database.

Procedure

1. Copy the replication plugin to /usr/lib/hbase/lib/ on all EMR cluster RegionServer. (requires root access);



Important: This step requires you to have root access to the EMR cluster.

2. Add the keystore generated on the destination Cloudera Operational Database cluster to HDFS using the following command:

```
hdfs dfs -mkdir /hbase-replication

hdfs dfs -put /[*PATH*]/[*CREDENTIALS.JCEKS*] /hbase-replication
hdfs dfs -chown -R hbase:hbase /hbase-replication
```

3. Edit the EMR /etc/hbase/conf.dist/hbase-site.xml file on each RegionServer host to add the following values:

```
<property>
  <name>[**HBASE.SECURITY.REPLICATION.CREDENTIAL.PROVIDER.PATH**]</na
me>
  <value>cdprepjceks://hdfs@[**NAMENODE_HOST**]:[**NAMENODE_PORT**]/
hbase-replication/credentials.jceks</value>
</property>
<property>
  <name>[**HBASE.SECURITY.REPLICATION.USER.NAME**]</name>
  <value>srv_[**WORKLOAD USER NAME**]</value>
</property>
```

4. Restart the EMR RegionServer by stopping the running processes. RegionServers autostarts and now uses the additional JAR files in the classpath.
5. Use _ReplicationSetupTool_ to add the peer. _ReplicationSetupTool_ is a command line tool that enables you to create the replication peer and make necessary configuration to be peer specific.

Example

```
sudo -u hbase hbase org.apache.hadoop.hbase.client.replication.ReplicationSe
tupTool
-clusterKey "ZK-HOST-1,ZK-HOST-2,ZK-HOST-3:2181:/hbase"
-endpointImpl "org.apache.hadoop.hbase.replication.regionserver.CldrHBase
InterClusterReplicationEndpoint" -peerId 1
```

Using the CldrCopyTable utility to copy data

You can use the CldrCopyTable utility to copy data from one Cloudera Operational Database cluster to another. You can use it to bring data in sync for replication.

About this task

CldrCopyTable is Cloudera's version of the upstream CopyTable utility. For more information about the CopyTable utility, see *Use CopyTable*.

The `--cldr.cross.domain` option of CldrCopyTable is an extension of the Cloudera Operational Database replication plugin that enables you to copy data cross-realm. The replication plugin uses PAM authentication to validate the machine user credentials. Cloudera Operational Database clusters are always provisioned with PAM authentication against the Cloudera environment FreeIPA security domain. Cloudera Single Sign on (CSSO) is used as the machine user credentials (for example, `csso_user`).

Procedure

1. Ensure that the following properties have the correct values in the `hbase-site.xml` configuration file of the target cluster:

```
<property>
<name>hbase.security.replication.credential.provider.path</name>
<value>cdprepjceks://hdfs@[***NAMENODE_HOST***]:[***NAMENODE_PORT***]/hbase-replication/credentials.jceks</value>
</property>

<property>
<name>hbase.security.replication.user.name</name>
<value>srv_[***WORKLOAD_USER_NAME***]</value>
</property>
```

2. Ensure that the source cluster can communicate with the target cluster:

- a) Get the ZooKeeper quorum address of the target cluster.
- b) Set the address as an environment parameter in the source cluster.
- c) Set a subnet that allows connection from the source cluster.

For example by enabling the port 2181 for the ZooKeeper client.

3. Issue the CldrCopyTable command from the source cluster to write to the target cluster.

Based on your target cluster setup you have to use either the `--cldr.cross.domain` or the `--cldr.unsecure.peer` option.

For Secure target cluster

Use the `--cldr.cross.domain` option:

```
hbase org.apache.hadoop.hbase.mapreduce.CldrCopyTable --cldr
.cross.domain --peer.adr=[***ZOOKEEPER_QUORUM***]:[***ZOOKEEPER
PORT***]:[***ZOOKEEPER_ROOT_FOR_HBASE***] --new.name="[***NEW_TABLE
NAME***]" "[***SOURCE_TABLE_NAME***]"
```

For Unsecure target cluster

Use the `--cldr.unsecure.peer` option:

```
hbase.org.apache.hadoop.hbase.mapreduce.CldrCopyTable --cldr
.unsecure.peer --peer.adr=[***ZOOKEEPER_QUORUM***]:[***ZOOKEEPER
```

```
PORT***]:[***ZOOKEEPER ROOT FOR HBASE***] --new.name="[***NEW TABLE
NAME***]" "[***SOURCE TABLE NAME***]"
```

4. Once the job is finished, check the target cluster and ensure that the copy was successful.

Related Information

[Use CopyTable](#)

Using HashTable and SyncTable tools to copy data between Cloudera Operational Database clusters

You can use the HashTable and SyncTable tools to copy data from one Cloudera Operational Database cluster to another. You can use these tools to synchronize data prior to replication.

About this task

You can use the HashTable and SyncTable CLI tools as a one way synchronization method for data in Cloudera Operational Database clusters. The CldrSyncTable job is an extension of the upstream SyncTable tool. For more information about the HashTable and SyncTable tools, see *Use HashTable and SyncTable tool*.

When you use these tools, ensure that you place the HashTable output directory and the source table at the same location where the CLI exists. This means that you cannot set the sourcezkcluster and the sourcehashdir properties to a remote cluster that the command-line executor cannot authenticate.

Before you begin

- Ensure that all RegionServers and DataNodes on the source cluster are accessible by NodeManagers on the target cluster where SyncTable job tasks are running.
- In case of secured clusters, users on the target cluster who execute the SyncTable job must be able to do the following on the HDFS and HBase services of the source cluster:
 - Authenticate: for example, using the centralized authentication or cross-realm setup.
 - Be authorized: having at least read permission.
- Ensure that the target table is created and enabled on the target cluster.

Procedure

1. Ensure that the following properties have the correct values in the hbase-site.xml configuration file of the target cluster:

```
<property>
  <name>hbase.security.replication.credential.provider.path</name>
  <value>cdprepjceks://hdfs@[***NAMENODE_HOST***]:[***NAMENODE_PORT***]/
hbase-replication/credentials.jceks</value>
</property>
<property>
  <name>hbase.security.replication.user.name</name>
  <value>srv_[***WORKLOAD USER NAME***]</value>
</property>
```

2. Ensure that the source cluster can communicate with the target cluster.
 - a) Get the ZooKeeper quorum address of the target cluster.
 - b) Set a subnet that allows connection from the source cluster.

For example, enabling the port 2181 for the ZooKeeper client.

3. Run the HashTable job from the source cluster to generate a hash of a table on the source cluster:

```
hbase org.apache.hadoop.hbase.mapreduce.HashTable
```



```

[***TABLE NAME***] [***HASH OUTPUT PATH***]
hdfs dfs -ls -R [***HASH OUTPUT PATH***]

```

4. Run the CldrSyncTable job from the source cluster to compare the generated hashes:

Use the `--cldr.cross.domain` or the `--cldr.unsecure.peer` option:

For Secure target cluster

Use the `--cldr.cross.domain` option when running the CldrSyncTable job in a secure cluster:

```

hbase org.apache.hadoop.hbase.mapreduce.CldrSyncTable
--cldr.cross.domain --targetzkcluster=[***TARGET ZOOKEEPER
QUORUM***]:[***TARGET ZOOKEEPER PORT***]:[***TARGET ZOOKEEPER ROOT FOR
HBASE***]
[***HASH OUTPUT PATH***] [***SOURCE TABLE NAME***]
[***TARGET TABLE NAME ON THE TARGET CLUSTER***]

```

For Unsecure target cluster

Use the `--cldr.unsecure.peer` option when running the CldrSyncTable job in an unsecure cluster::

```

hbase org.apache.hadoop.hbase.mapreduce.CldrSyncTable
--cldr.unsecure.peer
--targetzkcluster=[***TARGET ZOOKEEPER QUORUM***]:[***TARGET
ZOOKEEPER PORT***]:[***TARGET ZOOKEEPER ROOT FOR HBASE***]
[***HASH OUTPUT PATH***] [***SOURCE TABLE NAME***]
[***TARGET TABLE NAME ON THE TARGET CLUSTER***]

```

5. After the job is completed, check the data copied on the target cluster.

```

scan '[***TARGET TABLE NAME***]'

```

Related Information

[Use HashTable and SyncTable tool](#)