CDP Private Cloud

Migrating Operational Database to Cloudera Base on premises

Date published: 2019-08-22
Date modified: 2022-04-08

https://docs.cloudera.com/

Legal Notice

Contents

Migrating HBase 4
Preparing for data migration 4
Removing PREFIX_TREE Data Block Encoding 4
Checking co-processor classes 6
Migrating HBase from CDH or HDP 7
Verifying and validating if your data is migrated 9
HBase unsupported features 11

Migrating Accumulo to Cloudera 11
Migrating to Operational Database powered by Apache Accumulo 11
Migrating to Accumulo 1.10.0 12
In-place data upgrade from Accumulo 1.7.0 in HDP 2 to Accumulo 1.10 12
In-place data upgrade from Accumulo 1.7.0 in HDP 3 to Accumulo 1.10 13
In-place data upgrade from Accumulo 1.7.2 in CDH 5 to Accumulo 1.10 14
In-place data upgrade from Accumulo 1.9.2 in CDH 6 to Accumulo 1.10 15
In-place data upgrade from CDH 6 to Operational Database powered by Apache Accumulo 16

Migrating HBase

You can migrate your Apache HBase workloads from CDH and HDP to Cloudera. To successfully migrate your Apache HBase workloads, you must first understand the data management differences between the two platforms and prepare your source data to be compatible with the destination Cloudera platform.

Migrating your workload means migrating your data to Cloudera and making your applications access the data in Cloudera.

When migrating your data to the Cloudera Base on premises deployment, you must use the Apache HBase replication and snapshot features, along with the HashTable/SyncTable tool.

Using the Apache HBase replication and snapshot feature ensures that you do not face any data migration bottlenecks even when you have large amounts of data in your source cluster. The HashTable/SyncTable tool ensures that the data migrated to the destination cluster is synchronized with your source cluster, and lets you verify if your migration is successful.

Preparing for data migration

Before you start the data migration from CDH 5.x or HDP 2.x to Cloudera, you must understand the requirements and complete certain tasks on CDH/HDP and Cloudera to ensure a successful migration.

Procedure

- If you are migrating from CDH, configure Ranger ACLs in Cloudera corresponding to the HBase ACLs in your existing CDH cluster.
- If you are migrating from HDP perform the following steps:
  a) Configure Ranger ACLs in Cloudera corresponding to the HBase or Ranger ACLs in your existing HDP cluster.

     For more information, see Configure a resource-based service: HBase.
  b) Migrate your applications to use the new HBase-Spark connector because the Spark-HBase connector that you were using in CDH or HDP is no longer supported in Cloudera.

     For more information, see Using the HBase-Spark connector.
- Review the deprecated APIs and incompatibilities for HBase, if any, when migrating from HDP 2.x or CDH 5.x to Cloudera.

  For more information, see Deprecation Notices for Apache HBase.
- Ensure that all data has been migrated to a supported encoding type before the migration.

  For more information, see Remove PREFIX_TREE Data Block Encoding.
- Ensure that you upgrade any external co-processors manually because they are not automatically upgraded during migration. You need to ensure that your co-processors classes are compatible with Cloudera.

  For more information, see Check co-processor classes.

Removing PREFIX_TREE Data Block Encoding

Before migrating to Cloudera Base on premises, ensure that you have transitioned all the data to a supported encoding type.

About this task

Important:

Ensure that you complete all the pre-migration steps if you have Apache HBase installed in your existing CDH cluster.

The PREFIX_TREE data block encoding code is removed in Cloudera Base on premises, meaning that HBase clusters with PREFIX_TREE enabled will fail. Therefore, before migrating to Cloudera Base on premises you must ensure that all data has been transitioned to a supported encoding type.

The following pre-upgrade command is used for validation: hbase pre-upgrade validate-dbe

If your cluster is Kerberized, ensure that you run the kinit command as a hbase user before running the pre-upgrade commands. Ensure you have valid Kerberos credentials. You can list the Kerberos credentials using the klist command, and you can obtain credentials using the kinit command.

Before you begin

1. Download and distribute parcels for the target version of Cloudera.

   

   Important: Do not activate the parcel yet.

   If the downloaded parcel version is higher than the current Cloudera Manager version, the following error message displayed:

   Error for parcel CDH-7.x.parcel : Parcel version 7.X is not supported by this version of Cloudera Manager. Upgrade Cloudera Manager to at least 7.X before using this version of the parcel.

   You can safely ignore this error message.
2. Find the installed parcel at /opt/cloudera/parcels.

   For example: /opt/cloudera/parcels/CDH-7.1.1-1.cdh7.1.1.p0.3224867/bin/hbase

Use the Cloudera parcel to run the pre-upgrade commands. Cloudera recommends that you run them on an HMaster host.

Procedure

1. Run the hbase pre-upgrade validate-dbe command using the new installation.

   For example, if you have installed the CDH-7.1.1-1 parcel, you must run the following command:

   ```
   /opt/cloudera/parcels/CDH-7.1.1-1.cdh7.1.1.p0.3224867/bin/hbase pre-upgr
   ade validate-dbe
   ```

   The commands check whether your table or snapshots use the PREFIX_TREE Data Block Encoding.

   This command does not take much time to run because it validates only the table-level descriptors.

   If PREFIX_TREE Data Block Encoding is not used, the following message is displayed:

   ```
   The used Data Block Encodings are compatible with HBase 2.0.
   ```

   If you see this message, your data block encodings are compatible, and you do not have to perform any more steps.

   If PREFIX_TREE Data Block Encoding is used, a similar error message is displayed:

   ```
   2018-07-13 09:58:32,028 WARN  [main] tool.DataBlockEncodingValidator: In
   compatible DataBlockEncoding for table: t, cf: f, encoding: PREFIX_TREE
   ```

   If you got an error message, continue with Step 2 and fix all your PREFIX_TREE encoded tables.

2. Fix your PREFIX_TREE encoded tables using the old installation.

   You can change the Data Block Encoding type to PREFIX, DIFF, or FAST_DIFF in your source cluster.

   Our example validation output reported column family f of table t is invalid. Its Data Block Encoding type is changed to FAST_DIFF in this example:

   ```
   hbase> alter 't', { NAME => 'f', DATA_BLOCK_ENCODING => 'FAST_DIFF' }
   ```

Checking co-processor classes

External co-processors are not automatically upgraded, you must upgrade them manually. Before migrating, ensure that your co-processors are compatible with themigration.

About this task

Important:

Ensure that you complete all the pre-migration steps if you have Apache HBase installed in your existing CDH cluster.

There are two ways to handle co-processor upgrade:

- Upgrade your co-processor jars manually before continuing the migration.
- Temporarily unset the co-processors and continue the migration.

  Once they are manually upgraded, they can be reset.

Attempting to migrate without upgrading the co-processor jars can result in unpredictable behaviour such as HBase role start failure, HBase role crashing, or even data corruption.

If your cluster is Kerberized, ensure that you run the kinit command as a hbase user before running the pre-upgrade commands.

Procedure

1. Download and distribute parcels for target version of Cloudera Base on premises.

   

   Important: Do not activate the parcel yet.

   If the downloaded parcel version is higher than the current Cloudera Manager version, the following error message displayed:

   Error for parcel CDH-7.X.parcel : Parcel version 7.X is not supported by this version of Cloudera Manager. Upgrade Cloudera Manager to at least 7.X before using this version of the parcel.

   You can safely ignore this error message.

2. Run the hbase pre-upgrade validate-cp commands to check if your co-processors are compatible with the migration.

   Use the Cloudera parcel to run the pre-upgrade commands. Cloudera recommends that you run them on an HMaster host.

   For example, you can check for co-processor compatibility on master:

   ```
   $ /opt/cloudera/parcels/CDH-7.1.1-1.cdh7.1.1.p0.3224867/bin/hbase pre-up
   grade validate-cp -jar /opt/cloudera/parcels/CDH-7.1.1-1.cdh7.1.1.p0.322
   4867/jars/ -config
   ```

   Or, you can validate every table level co-processors where the table name matches to the .* regular expression:

   ```
   $ /opt/cloudera/parcels/CDH-7.1.1-1.cdh7.1.1.p0.3224867/bin/hbase pre-up
   grade validate-cp -table '.*'
   ```

   Optinally, you can run the following command for a more detailed output:

   ```
   HBASE_ROOT_LOGGER=DEBUG,console hbase pre-upgrade validate-cp -table '.*'
   ```

   This way you can verify that all of the required tables were checked. The detailed output should contain lines like the following where test_table is a table on the server:

   ```
   21/05/10 11:07:58 DEBUG coprocessor.CoprocessorValidator: Validating table
    test_table
   ```

3. Check the output to determine if your co-processors are compatible with the upgrade.

   The output looks similar to the following:

   ```
   $ hbase pre-upgrade validate-cp -config
                           ... some output ...
                           $ echo $?
                           0
   ```

   If echo $? prints 0, the check was successful and your co-processors are compatible. A non-zero value means unsuccessful, your co-processors are not compatible.

Migrating HBase from CDH or HDP

Before you migrate your data, you must have an Apache HBase cluster created on Cloudera Base on premises. Your CDH or HDP cluster is your source cluster, and your Cloudera Base on premises cluster is your destination cluster.

Procedure

1. Deploy HBase replication on both the source and the destination cluster.

   For instructions, see Deploy HBase replication.

2. Enable replication on both the source and destination clusters by running the following commands in the HBase Shell.

   On the source cluster

   ```
   create 't1',{NAME=>'f1', REPLICATION_SCOPE=>1}
   ```

   On the destination cluster

   ```
   create 't1',{NAME=>'f1', KEEP_DELETED_CELLS=>'true'}
   ```

   Note: Cloudera recommends enabling KEEP_DELETED_CELLS on column families in the destination cluster, where REPLICATION_SCOPE=1 in the source cluster.

3. Run the add_peer <ID>, <DESTINATION_CLUSTER_KEY> command in the HBase Shell on the source cluster to add the destination cluster as a peer.

   You can get the DESTINATION_CLUSTER_KEY value from the HBase Master user interface that you can access using Cloudera Manager.

4. Monitor replication status and verify whether the queued writes on the source cluster are flowing to the destination.

   ```
   hbase shell
   hbase> status 'replication'
   hbase01.home:
   SOURCE: PeerID=1
   Normal Queue: 1
   AgeOfLastShippedOp=0, TimeStampOfLastShippedOp=Fri Jun 12 18:49:23 BST 20
   20, SizeOfLogQueue=1, EditsReadFromLogQueue=1, OpsShippedToTarget=1, Tim
   eStampOfNextToReplicate=Fri Jun 12 18:49:23 BST 2020, Replication Lag=0
   SINK: TimeStampStarted=1591983663458, AgeOfLastAppliedOp=0, TimeStampsOf
   LastAppliedOp=Fri Jun 12 18:57:18 BST 2020
   ```

   The Replication Lag=0 demonstrates that the data is replicated to the destination cluster successfully.

   For more information, see Monitoring Replication Status.

5. Run the disable_peer <ID1> command in the HBase Shell on the source cluster to disable the peer in the source cluster
   This stop the replication with the peer, but the logs are retained for future reference.

6. Take a snapshot in Cloudera Manager.

   a) Select the HBase service.
   b) Click the Table Browser tab.
   c) Click a table.
   d) Click Take Snapshot.
   e) Specify the name of the snapshot, and click Take Snapshot.

   To migrate HBase data from HDP, run the following sample HBase shell command to generate HBase table snapshot.

   ```
   hbase shell
   hbase> snapshot 'myTable', 'myTableSnapshot-122112'
   ```

   You must run this command for each table that you want to migrate.

7. Run the ExportSnapshot command in the HBase Shell on the source cluster to export a snapshot from the source to the destination cluster. You must run the ExportSnapshot command as the hbase user or the user that owns the files.

   The ExportSnapshot tool executes a MapReduce Job similar to distcp to copy files to the other cluster. ExportSnapshot works at the file-system level, so the HBase cluster can be offline.

   ```
   hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot <snapshot
    name> -copy-to hdfs://destination:hdfs_port/hbase -mappers 16
   ```

   Here, destination (hdfs://destination:hdfs_port/hbase) is the destination Cloudera Base on premises cluster. Replace the HDFS server path and port with the ones you have used for your cluster.

   Important: Snapshots must be enabled on the source and destination clusters. When you export a snapshot, the table's HFiles, logs, and the snapshot metadata are copied from the source cluster to the destination cluster.

8. Run the following commands in the HBase Shell on the destination cluster to restore the table. If the table already exists on the destination cluster, use restore_snapshot to restore it to the state recorded in the snapshot.

   ```
   disable <tablename>
   restore_snapshot <snapshotname>
   ```

   If the snapshot still does not exist, use the clone_snapshot command to recreate it.

   ```
   clone_snapshot <snapshotname> <tablename>
   ```

9. Run the enable_peer <ID1> command in the HBase Shell on the source cluster to enable the peer in the source and destination clusters.
10. Run the HashTable command on the source cluster and the SyncTable command on the destination cluster to synchronize the table data between your source and destination clusters.

    On the source cluster

    ```
    HashTable [options] <tablename> <outputpath>
    ```

    On the destination cluster

    ```
    SyncTable [options] <sourcehashdir> <sourcetable> <targettable>
    ```

    For more information and examples about using HashTable and SyncTable, see Verifying and validating if your data is migrated on page 9.

Verifying and validating if your data is migrated

You can use the SyncTable command with the --dryrun parameter to verify if the tables are in sync between your source and your destination clusters. The SyncTable --dryrun option makes this run of your SyncTable command as read-only.

About this task

The HashTable and SyncTable jobs compose a tool implemented as two map-reduce jobs that must be executed as individual steps. It is similar to the CopyTable tool, which can perform both partial or entire table data copy. Unlike CopyTable it only copies diverging data between target clusters, saving both network and computing resources during the copy procedure.

Procedure

1. Run the HashTable MapReduce job. This must be run on the cluster whose data is copied to the remote peer, usually the source cluster.

```
hbase org.apache.hadoop.hbase.mapreduce.HashTable --families=cf my-table /
hashes/test-tbl
…
20/04/28 05:05:48 INFO mapreduce.Job:  map 100% reduce 100%
20/04/28 05:05:49 INFO mapreduce.Job: Job job_1587986840019_0001 completed
 successfully
20/04/28 05:05:49 INFO mapreduce.Job: Counters: 68
…
File Input Format Counters
Bytes Read=0
File Output Format Counters
Bytes Written=6811788
```

   Once the HashTable job execution with the above command is completed, some output files are generated in the source hdfs /hashes/my-table directory. These files are needed as an input for the SyncTable execution.

```
hdfs dfs -ls -R /hashes/test-tbl
drwxr-xr-x   - root supergroup          0 2020-04-28 05:05 /hashes/test-
tbl/hashes
-rw-r--r--   2 root supergroup          0 2020-04-28 05:05 /hashes/test-
tbl/hashes/_SUCCESS
drwxr-xr-x   - root supergroup          0 2020-04-28 05:05 /hashes/test-
tbl/hashes/part-r-00000
-rw-r--r--   2 root supergroup    6790909 2020-04-28 05:05 /hashes/test-
tbl/hashes/part-r-00000/data
-rw-r--r--   2 root supergroup      20879 2020-04-28 05:05 /hashes/test-
tbl/hashes/part-r-00000/index
-rw-r--r--   2 root supergroup         99 2020-04-28 05:04 /hashes/test-t
bl/manifest
-rw-r--r--   2 root supergroup        153 2020-04-28 05:04 /hashes/test-
tbl/partitions
```

2. Launch the SyncTable at the target peer. The following command runs SyncTable for the output of HashTable from the previous step. It uses the --dryrun parameter.

```
hbase org.apache.hadoop.hbase.mapreduce.SyncTable --dryrun --sourcezkclu
ster=zk1.example.com,zk2.example.com,zk3.example.com:2181:/hbase hdfs://
source-cluster-active-nn/hashes/test-tbl test-tbl test-tbl
…
org.apache.hadoop.hbase.mapreduce.SyncTable$SyncMapper$Counter
BATCHES=97148
HASHES_MATCHED=97146
HASHES_NOT_MATCHED=2
MATCHINGCELLS=17
MATCHINGROWS=2
RANGESNOTMATCHED=2
ROWSWITHDIFFS=2
SOURCEMISSINGCELLS=1
TARGETMISSINGCELLS=1
```

   In the previous output, the SyncTable is reporting two rows diverging in both source and target (ROWSWITHDIFFS=2), where one row has a cell value in source not present in target (TARGETMISSINGCELLS=1), and another row has a cell value in the target that is not present in source (SOURCEMISSINGCELLS=1).

Note: You might replace the given parameters in the above examples with your actual environment values.

What to do next

- The HashTable or SyncTable jobs are designed to operate on individual tables. If multiple tables need to be migrated, you must execute these jobs separately for each table.
- If the data in a table is modified either through ingestion or deletion on the source or destination, the job reports mismatches. To narrow the scope of data being checked, you can use the --starttime or --endtime options. For more information, see the *Hashtable reference guide* section.

Related Information

Hashtable reference guide

HBase unsupported features

You need to know about some features you might have used in CDH or HDP are no longer supported in Cloudera.

- HBase cannot use an HDFS directory for storage that is configured for HDFS Erasure Coding.
- HBase in Cloudera Base on premises does not support using any Cloud infrastructure blob storage products (For example, AWS S3, Azure ABFS, and Google GCS) or any similar API-compatible offerings (example, Netapp) for storing HBase data.

Migrating Accumulo to Cloudera

As Operational Database powered by Apache Accumulo is handled as a different service from Accumulo, in-place upgrade is not supported. However, an in-place data upgrade is supported from CDH 6 to Cloudera, or you can migrate your data and configuration to Operational Database.

Cloudera does not support in-place upgrade because Operational Database is handled as a different service from Accumulo, so the configuration changes would be lost on upgrade. You can use the migration steps to migrate from Accumulo 1.x to Operational Database, or if you are migrating from CDH 6 to Cloudera, you can follow the in-place data upgrade process.

Migrating to Operational Database powered by Apache Accumulo

Follow these steps to migrate the Accumulo service's configuration and data to Operational Database powered by Apache Accumulo.

About this task

Cloudera does not support in-place upgrade because Operational Database is handled as a different service from Accumulo, so the configuration changes would be lost on upgrade. You can use these migration steps to migrate from Apache Accumulo 1.x to Operational Database.

In-place data upgrade is supported from CDH 6 to Operational Database. That is because CDH 6 is shipped with Accumulo 1.9.2 which uses a data version compatible with Operational Database. However, that is only an in-place data upgrade instead of in-place upgrade, meaning that the configuration changes are getting lost during the upgrade. If you do not want to lose your custom settings, use the migration steps instead of the in-place data upgrade. For more information about in-place data upgrade, see *In-place data upgrade from CDH 6 to Operational Database powered by Apache Accumulo*.

Procedure

1. Migrate Accumulo configuration.
   a) Export the service configuration on the source cluster.

      The way you can export the configuration depends on your environment. For more information, see the applicable documentation:

      - Cloudera Manager 5.1.6.x: Exporting the Cloudera Manager Configuration
      - Cloudera Manager 6.3.x: Exporting the Cloudera Manager Configuration
      - Ambari: Download cluster blueprints without hosts

   b) Manually add the configuration to the destination cluster.
2. Migrate Accumulo data.
   a) Export each table on the source cluster.
   b) Collect the exported table data for transfer.
   c) Move the collected data into the HDFS storage on the destination cluster.
   d) Install and configure the Accumulo on Cloudera service on the destination cluster.
   e) Import each table into the destination cluster.

Migrating to Accumulo 1.10.0

As the Accumulo 1.10 is handled as a different service from Accumulo, in-place upgrade is not supported. However, an in-place data upgrade is supported from any CDH 5, CDH 6, HDP 2, or HDP 3 releases.

Cloudera does not support in-place upgrades because Accumulo is handled as a different service from Accumulo, so the configuration changes would be lost on the upgrade. Please follow the in-place data upgrade process described in the following sections.

In-place data upgrade from Accumulo 1.7.0 in HDP 2 to Accumulo 1.10

HDP 2 has Accumulo 1.7.0 that still uses an older format for the root user name when it stores it in ZooKeeper. This is no longer compatible with Accumulo 1.10. To prevent such issues, before adding the new Accumulo-on-cdp service, you must first stop the old service and then change the stored name.

Procedure

1. Start a zookeeper-client.
2. Authenticate with the old Accumulo root user and password.
   a) addauth digest {accumulo_root_user}:{accumulo_root_user_pass}
   b) If you do not know the password, you can create a superuser for ZooKeeper that has the permissions to make the necessary change.

      Following are the steps to create a superuser for ZooKeeper.

      1. Add the following line to the ZooKeeper Server Java options.

         DigestAuthenticationProvider.superDigest=super:UdxDQl4f9v5oITwcAsO9bmWgHSI=
      2. Now you can authenticate with addauth digest super:super123.
3. Obtain the current InstanceId.

   get /accumulo/instances/{accumulo_instance_name}
4. Change the root user for this instance.

   set /accumulo/{accumulo_instance_id}/users {old_accumulo_principal}@{kerberos_realm_name}

What to do next

The old headless Accumulo principals, like the current root user, are no longer handled by Cloudera Manager and not added to the generated keytabs. The old keytabs generated by Ambari still exist on the nodes, but to ensure that

the necessary keytabs can be re-generated if needed, we first have to add the headless principal to Cloudera Manager through the Cloudera Manager API. To do so use the following call: Support -> API Explorer -> /cm/commands/generateCredentialsAdhoc

Use the following body:

```
{
  "items": [
 "accumulo@EXAMPLE.COM"
  ]
}
```

When the keytab needs to be regenerated it can be done using another call: Support -> API Explorer -> /cm/retrieveKeytab

With the following body:

```
{
  "items": [
 "accumulo@EXAMPLE.COM"
  ]
}
```

The above call returns a link using which the keytab can be downloaded.

In case you use the old principal, it might have trouble accessing HDFS after the upgrade. In that case change the permission in the HDFS -> Configuration -> extra_auth_to_local_rules file as follows:

RULE:[1:$1@$0](accumulo@EXAMPLE.COM)s/.*/accumulo/

This ensures that the old principal has the same permissions as the new Accumulo principals.

After the update the tracer and monitor roles automatically use the node specific Accumulo principals. These principals does not have permissions for the trace table. You must add the permissions as follows:

```
grant Table.READ -t trace -u accumulo/{hostname}@EXAMPLE.COM
grant Table.WRITE -t trace -u accumulo/{hostname}@EXAMPLE.COM
grant Table.ALTER_TABLE -t trace -u accumulo/{hostname}@EXAMPLE.COM
```

In-place data upgrade from Accumulo 1.7.0 in HDP 3 to Accumulo 1.10

HDP 3 has Accumulo 1.7.0 that still uses an older format for root user name when it stores it in ZooKeeper. This is no longer compatible with Accumulo 1.10. To prevent such issues, before adding the new Accumulo-on-Cloudera service, you must first stop the old service and then change the stored name.

Procedure

1. Start a zookeeper-client.
2. Authenticate with the old Accumulo root user and password.
   a) addauth digest {accumulo_root_user}:{accumulo_root_user_pass}
   b) If you do not know the password, you can create a superuser for ZooKeeper that has the permissions to make the necessary change.

      Following are the steps to create a superuser for ZooKeeper.

      1. Add the following line to the ZooKeeper Server Java options.

         DigestAuthenticationProvider.superDigest=super:UdxDQl4f9v5oITwcAsO9bmWgHSI=
      2. Now you can authenticate with addauth digest super:super123.
3. Obtain the current InstanceId.

   get /accumulo/instances/{accumulo_instance_name}

4. Change the root user for this instance.
   set /accumulo/{accumulo_instance_id}/users {old_accumulo_principal}@{kerberos_realm_name}

What to do next

The old headless Accumulo principals, like the current root user, are no longer handled by Cloudera Manager and not added to the generated keytabs. The old keytabs generated by Ambari still exist on the nodes, but to ensure that the necessary keytabs can be re-generated if needed, we first have to add the headless principal to Cloudera Manager through the Cloudera Manager API. To do so use the following call: Support -> API Explorer -> /cm/commands/generateCredentialsAdhoc

Use the following body:

```
{
  "items": [
 "accumulo@EXAMPLE.COM"
  ]
}
```

When the keytab needs to be regenerated it can be done using another call: Support -> API Explorer -> /cm/retrieveKeytab

With the following body:

```
{
  "items": [
 "accumulo@EXAMPLE.COM"
  ]
}
```

The above call returns a link using which the keytab can be downloaded.

In case you use the old principal, it might have trouble accessing HDFS after the upgrade. In that case change the permission in the HDFS -> Configuration -> extra_auth_to_local_rules file as follows:

RULE:[1:$1@$0](accumulo@EXAMPLE.COM)s/.*/accumulo/

This ensures that the old principal has the same permissions as the new Accumulo principals.

After the update the tracer and monitor roles automatically use the node specific Accumulo principals. These principals does not have permissions for the trace table. You must add the permissions as follows:

```
grant Table.READ -t trace -u accumulo/{hostname}@EXAMPLE.COM
grant Table.WRITE -t trace -u accumulo/{hostname}@EXAMPLE.COM
grant Table.ALTER_TABLE -t trace -u accumulo/{hostname}@EXAMPLE.COM
```

In-place data upgrade from Accumulo 1.7.2 in CDH 5 to Accumulo 1.10

All the supported Accumulo version (1.7+) has a data version compatible with Accumulo 1.10 so an in-place data upgrade is supported from all of them.

About this task

Only an in-place data upgrade is supported to Cloudera, which is not an in-place upgrade. That means that the configuration changes are getting lost during the upgrade. If you do not want to lose your custom settings, use the migration steps instead of the in-place data upgrade. For more information about how to migrate to Accumulo on Cloudera, see *Migrating Accumulo*.

Procedure

1. Migrate Accumulo configuration.

   As Cloudera Operational Database is handled as a different service from Accumulo, you have to migrate the configuration of Accumulo manually. For more information, see *Migrating Accumulo*.

   Note: Ensure that you take a note on which nodes each role is running.

2. Remove the Accumulo service.

   There is no automatic service upgrade for Accumulo, so during the in-place cluster upgrade process, it is removed if it still presents on the cluster at the time of the upgrade. Cloudera recommends removing the Accumulo service manually before starting the cluster upgrade.

   - In Cloudera Manager, stop the Accumulo service.
   - Delete the Accumulo service.

     The deleting process affects only configurations, it does not affect the data files or Zookeeper nodes.

3. Upgrade the cluster.

   For more information, see *Upgrading CDH 5 to Cloudera Base on premises*.

4. Install the new Cloudera Operational Database service.

   For more information, see *Install Accumulo 1.10*.

   When performing the installation, ensure that the following requirements are fulfilled:

   - Assign each role to the same node it was running on before the upgrade.
   - Use the same installation settings configurations the previous Accumulo service was using.
   - Clear Initialize Accumulo.
   - Clear Enable Kerberos Authentication. The main purpose of this function is to make a kerberized setup easier for a new service install. The changes it entails could clash with the previous Accumulo service configuration.

5. Re-apply the previous Accumulo configurations.

   Note: Major changes were introduced regarding how Accumulo handles the configurations. For more information, see *Migrating Accumulo*.

Related Information

Migrating Accumulo to Cloudera
Upgrading CDH 5 to Cloudera Base on premises
Installing Accumulo Parcel 1.10

In-place data upgrade from Accumulo 1.9.2 in CDH 6 to Accumulo 1.10

Every currently supported Accumulo version (1.7+) has a data version compatible with Accumulo 1.10 so an in-place data upgrade is supported from all of them.

About this task

Only an in-place data upgrade is supported to Cloudera, which is not an in-place upgrade. That means that the configuration changes are getting lost during the upgrade. If you do not want to lose your custom settings, use the migration steps instead of the in-place data upgrade. For more information about how to migrate to Accumulo on Cloudera, see *Migrating Accumulo*.

Procedure

1. Migrate Accumulo configuration.

   As Cloudera Operational Database is handled as a different service from Accumulo, you have to migrate the configuration of Accumulo manually. For more information, see *Migrating Accumulo*.

   Note: Ensure that you take a note on which nodes each role is running.

2. Remove the Accumulo service.

   There is no automatic service upgrade for Accumulo, so during the in-place cluster upgrade process, it is removed if it still presents on the cluster at the time of the upgrade. Cloudera recommends removing the Accumulo service manually before starting the cluster upgrade.

   - In Cloudera Manager, stop the Accumulo service.
   - Delete the Accumulo service.

     The deleting process affects only configurations, it does not affect the data files or Zookeeper nodes.

3. Upgrade the cluster.

   For more information, see *Upgrading CDH 6 to Cloudera Base on premises*.

4. Install the new Cloudera Operational Database service.

   For more information, see *Install Accumulo 1.10*.

   When performing the installation, ensure that the following requirements are fulfilled:

   - Assign each role to the same node it was running on before the upgrade.
   - Use the same installation settings configurations the previous Accumulo service was using.
   - Clear Initialize Accumulo.
   - Clear Enable Kerberos Authentication. The main purpose of this function is to make a kerberized setup easier for a new service install. The changes it entails could clash with the previous Accumulo service configuration.

5. Re-apply the previous Accumulo configurations.

   Note: Major changes were introduced regarding how Accumulo handles the configurations. For more information, see *Migrating Accumulo*.

Related Information

Migrating Accumulo to Cloudera
Upgrading CDH 6 to Cloudera Base on premises
Installing Accumulo Parcel 1.10

In-place data upgrade from CDH 6 to Operational Database powered by Apache Accumulo

As CDH 6 is shipped with Accumulo 1.9.2 which uses a data version compatible with Operational Datbase powered by Apache Accumulo, in-place data upgrade is supported from CDH 6 to Operational Database.

About this task

Only an in-place data upgrade is supported from CDH 6 to Operational Database, which is not an in-place upgrade. That means that the configuration changes are getting lost during the upgrade. If you do not want to lose your custom settings, use the migration steps instead of the in-place data upgrade. For more information about how to migrate to Operational Database, see *Migrating Operational Database powered by Apache Accumulo*.

Procedure

1. Migrate Accumulo configuration.

   As Operational Database is handled as a different service from Accumulo, you have to migrate the configuration of Accumulo manually. For more information, see *Migrating Operational Database powered by Apache Accumulo*.

   

   Note: Ensure that you note on which nodes each role is running.

2. Remove the Accumulo service.

   There is no automatic service upgrade for Accumulo, so during the in-place cluster upgrade process it is removed if it still presents on the cluster at the time of the upgrade. Cloudera recommends removing the Accumulo service manually before starting the cluster upgrade.

   a) In Cloudera Manager, stop the Accumulo service.
   b) Delete the Accumulo service.

      The deleting process affects only configurations, it does not affect data files or Zookeeper nodes.

3. Upgrade the cluster. For more information, see *Upgrading CDH 6 to Cloudera Base on premises*.
4. Install the new Operational Database service.

   For more information, see *Install Operational Database powered by Apache Accumulo*.

   When performing the installation, ensure that the following requirements are fulfilled:

   - Assign each role to the same node it was running on before the upgrade.
   - Use the same configurations the previous Accumulo service was using.
   - Clear Initialize Accumulo.
   - Clear Enable Kerberos Authentication. The main purpose of this function is to make a kerberized setup easier for a new service install. The changes it entails could clash with the previous Accumulo service configuration.

5. Re-apply the previous Accumulo configurations.

   

   Note: Major changes were introduced regarding how Accumulo handles configurations. Fore more information, see *Upgrading Accumulo*.