CDP Private Cloud

Migrating Streaming Workloads to Cloudera Private Cloud Base

Date published: 2019-08-22
Date modified: 2023-09-07

CLOUDERA

Legal Notice

Contents

Migrating Streaming workloads from HDF to Cloudera Private Cloud Base

Learn how you can migrate streaming workloads from HDF to Cloudera Private Cloud Base.

In this scenario data is migrated from a HDF 3.4 cluster with Streams Messaging Manager, Kafka, and Schema Registry to a Cloudera Private Cloud Base 7.1 cluster with Streams Messaging Manager, Kafka, Schema Registry, and Streams Replication Manager.

Multiple methods are provided for migrating Streams Messaging Manager and Schema Registry. Kafka is migrated with Streams Replication Manager set up on the target Cloudera Private Cloud Base 7.1 cluster. Multiple configuration methods are provided for the setup and configuration of Streams Replication Manager.
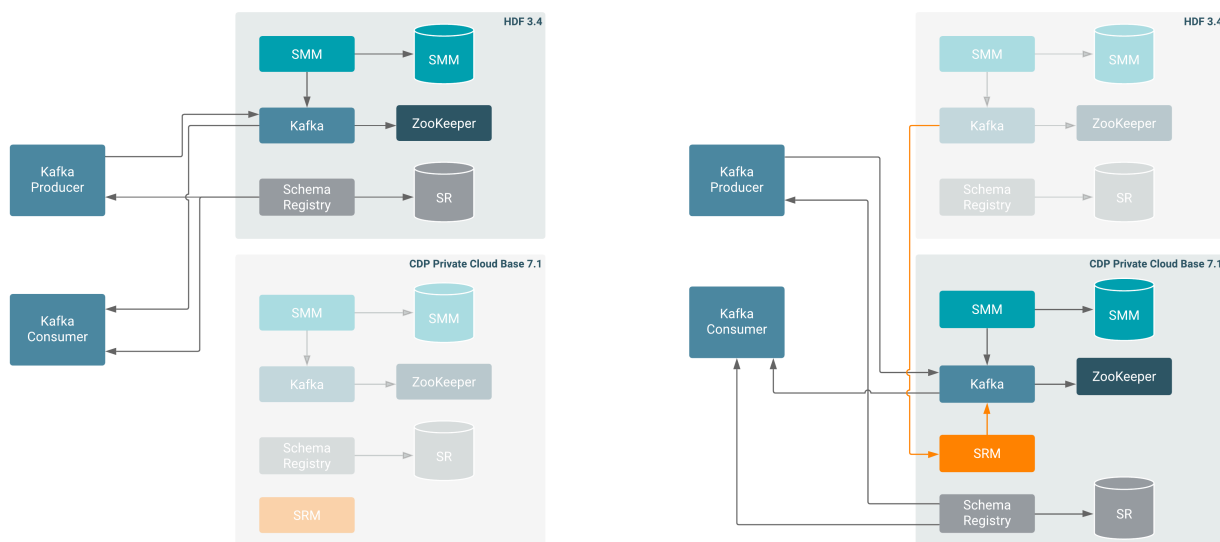
Complete the following steps in order to migrate your streaming workloads:

Set Up a New Streaming Cluster in Cloudera Private Cloud Base

How to set up a new Streaming cluster in Cloudera Private Cloud Base when migrating data from HDF.

In order to migrate your HDF workloads you need to set up a new Streaming Cluster in Cloudera Private Cloud Base. See the Cloudera Private Cloud Base Installation Guide as well as the Streaming Documentation for Runtime for installation, setup, and configuration instructions.

Related Information

CDP Private Cloud Base Installation Guide

Streaming Documentation for Runtime

Migrate Ranger Policies

How to migrate Ranger policies for Streaming clusters from HDF to Cloudera Private Cloud Base.

Use the Ranger Console (Ranger UI) or the Ranger REST API to import or export policies. For more information, see the Ranger documentation for Runtime or the Ranger documentation for HDP

Related Information

Ranger Documentation for Runtime

Ranger Documentation for HDP

Migrate Schema Registry

Overview of the methods that can be used to migrate Schema Registry from HDF to Cloudera Private Cloud Base.

There are two methods that you can use to migrate Schema Registry. You can either copy raw data or reuse existing storage. Review and choose one of the following methods:

Note: Cloudera recommends that you copy raw data.

Copy Raw Data

How to migrate Schema Registry from HDF to Cloudera Private Cloud Base by copying raw data.

Procedure

1. Stop existing Schema Registry clients.
2. Stop the HDF Schema Registry server.
3. Backup/restore the Schema Registry database from old database to new database:

   - MySQL - See the MySQL Backup and Recovery MySQL document.
   - PostgreSQL - See Chapter 24. Backup and Restore in the PostgreSQL documentation.
4. Copy all serdes from the HDF Schema Registry serdes jar location (local/HDFS) to the CDP Schema Registry serdes jar location (local/HDFS)
5. Configure CDP Schema Registry to connect to the new database:
   a) In Cloudera Manager select the Schema Registry service.
   b) Go to Configuration.
   c) Find and configure the following database related properties:

      - Schema Registry Database Type
      - Schema Registry Database Name
      - Schema Registry Database Host
      - Schema Registry Database Port
      - Schema Registry Database User
      - Schema Registry Database User Password
   d) Click Save Changes.
6. Start the CDP Schema Registry Server.
7. Reconfigure Schema Registry clients to point to the CDP Schema Registry Server.
8. Restart Schema Registry clients.

Results
Schema Registry is migrated. The HDF Schema Registry is no longer required.

What to do next
Migrate Streams Messaging Manager.
Related Information
MySQL Backup and Recovery

PostgreSQL Backup and Restore

Migrate Streams Messaging Manager

Reuse Existing Storage

How to migrate Schema Registry from HDF to Cloudera Private Cloud Base by reusing existing storage.

Before you begin

Make sure that the existing database is compatible with and supported by Cloudera Private Cloud Base. For more information, see Database Requirements in the CDP Release Guide.

Procedure

1. Stop existing Schema Registry clients.

2. Stop the HDF Schema Registry Server.

3. Configure CDP Schema Registry to connect to the database previously owned by HDF Schema Registry:

    a) In Cloudera Manager select the Schema Registry service.

    b) Go to Configuration.

    c) Find and configure the following database related properties:

- Schema Registry Database Type
- Schema Registry Database Name
- Schema Registry Database Host
- Schema Registry Database Port
- Schema Registry Database User
- Schema Registry Database User Password

4. Configure the CDP Schema Registry serdes jar location to point to the location used by the old HDF Schema Registry:

    a) In Cloudera Manager select the Schema Registry service.

    b) Go to Configuration.

    c) Find and configure the following properties:

- Schema Registry Jar Storage Type
- Schema Registry Jar Storage Directory Path
- Schema Registry Jar Storage HDFS URL

    d) Click Save Changes.

5. Start the CDP Schema Registry Server.

6. Reconfigure Schema Registry clients to point to the CDP Schema Registry Server.

7. Restart Schema Registry clients.

Results

Schema Registry is migrated. The HDF Schema Registry is no longer required.

What to do next
Migrate Streams Messaging Manager.
Related Information
CDP Database Requirements
Migrate Streams Messaging Manager

Migrate Streams Messaging Manager

Overview of the methods that can be used to migrate Streams Messaging Manager from HDF to Cloudera Private Cloud Base.

Streams Messaging Manager migration involves the migration of alert policies. The Streams Messaging Manager UI is stateless and no data migration is required.

⚠️ **Warning:** Streams Messaging Manager in HDF stores metrics in Ambari Metric Server (AMS). This data can not be migrated. Therefore, historic data is lost during the migration.

There are two methods you can use to migrate Streams Messaging Manager alert policies. You can either copy raw data or reuse existing storage.

In addition to these two migration methods, you can also choose to manually recreate alert policies in the new environment. Steps are provided for both storage reuse and data copy methods. For more information on manually recreating alert policies, see Managing Alert Policies and Notifiers in the Streams Messaging Manager documentation.

Related Information

Managing Alert Policies and Notifiers

Copy Raw Data

How to migrate Streams Messaging Manager alert policies from HDF to Cloudera Private Cloud Base by copying raw data.

Procedure

1. Stop the HDF Streams Messaging Manager.

2. Backup/restore the Streams Messaging Manager database from old database to new database:

   - MySQL - See the MySQL Backup and Recovery MySQL document.
   - PostgreSQL - See Chapter 24. Backup and Restore in the PostgresSQL documentation.

3. Configure CDP Streams Messaging Manager to connect to the new database:

   a) In Cloudera Manager select the Streams Messaging Manager service.
   b) Go to Configuration.
   c) Find and configure the following database related properties:

      - Streams Messaging Manager Database Type
      - Streams Messaging Manager Database Name
      - Streams Messaging Manager Database Host
      - Streams Messaging Manager Database Port
      - Streams Messaging Manager Database User
      - Streams Messaging Manager Database User Password

   d) Click Save Changes.
   e) Start the service.

Results

Streams Messaging Manager alert policies are migrated.

What to do next
Migrate Kafka using Streams Replication Manager.
Related Information
MySQL Backup and Recovery
PostgreSQL Backup and Restore
Migrate Kafka Using Streams Replication Manager

Reuse Existing Database

How to migrate Streams Messaging Manager alert policies from HDF to Cloudera Private Cloud Base by reusing existing storage.

Before you begin

Make sure that the existing database is compatible with and supported by Cloudera Private Cloud Base. For more information, see Database Requirements in the CDP Release Guide.

Procedure

1. Stop the HDF Streams Messaging Manager.
2. Configure CDP Streams Messaging Manager to connect to the database previously owned by HDF Streams Messaging Manager:
   a) In Cloudera Manager select the Streams Messaging Manager service.
   b) Go to Configuration.
   c) Find and configure the following database related properties:
      - Streams Messaging Manager Database Type
      - Streams Messaging Manager Database Name
      - Streams Messaging Manager Database Host
      - Streams Messaging Manager Database Port
      - Streams Messaging Manager Database User
      - Streams Messaging Manager Database User Password
   d) Click Save Changes.
   e) Start the service.

Results

Streams Messaging Manager alert policies are migrated.

What to do next
Migrate Kafka using Streams Replication Manager.
Related Information
CDP Database Requirements

Migrate Kafka Using Streams Replication Manager

Migrate Kafka Using Streams Replication Manager

Learn about the different options you have when migrating Kafka from HDF to Cloudera Private Cloud Base using Streams Replication Manager.

Kafka data is migrated from HDF to Cloudera Private Cloud Base using Streams Replication Manager. Streams Replication Manager can replicate data in various ways. How the data is replicated, and in this case migrated, is determined by the replication policy that is in use.

There are three replication policies that you can use when migrating data. These are the `DefaultReplicationPolicy`, the `IdenityReplicationPolicy`, and the `MigratingReplicationPolicy`. The following gives an overview of each policy and provides recommendations on which policy to use in different scenarios. Review the following sections and choose the policy that is best suited for your requirements.

DefaultReplicationPolicy

The `DefaultReplicationPolicy` is the default and Cloudera-recommended replication policy. This policy prefixes the remote (replicated) topics with the cluster name (alias) of the source topics. For example, the topic1 topic from the us-west source cluster creates the us-west.topic1 remote topic on the target cluster. Use this policy if topics getting renamed during the migration is acceptable for your deployment.

Additional notes:

• Remote topics will have different names in the target cluster. As a result, you must reconfigure existing Kafka clients to use the remote topic names.
• If you decide to, you can repurpose the Streams Replication Manager service you set up for migration and continue using it for replication.

IdentityReplicationPolicy

The `IdentityReplicationPolicy` does not change the names of remote topics. When this policy is in use, topics retain the same name on both source and target clusters. For example, the topic1 topic from the us-west source cluster creates the topic1 remote topic on the target cluster. Use this policy if you are on Cloudera Runtime 7.1.8 or higher and do not want remote topics to get renamed during migration.

Additional notes:

• In Cloudera Runtime 7.1.8 replication monitoring with this policy is not supported. This means that you will not be able to validate or monitor replications during the migration process. Support for replication monitoring is, however, available in Cloudera Runtime 7.1.9 or higher.
• If you decide to, you can repurpose the Streams Replication Manager service you set up for migration and continue using it for replication.
• If you want to continue using Streams Replication Manager after migration, review the limitations of this policy in the Streams Replication Manager Known Issues of the appropriate Cloudera Runtime version. Different limitations might apply depending on the Cloudera Runtime version.

MigratingReplicationPolicy

The `MigratingReplicationPolicy` is a custom replication policy that Cloudera provides the code for, but is not shipped with Streams Replication Manager like the `IdentityReplicationPolicy` or the `DefaultReplicationPolicy`. As a result, you must implement, compile, and package it as a JAR yourself.

This policy behaves similarly to the `IdentityReplicationPolicy` and does not rename replicated topics on target clusters. However, unlike the `IdentityReplicationPolicy`, this policy is only supported in data migration scenarios. Use this policy if you are using Cloudera Runtime 7.1.7 or lower and you do not want replicated topics to get renamed.

Additional notes:

• If you are using Cloudera Runtime 7.1.8 or later, Cloudera recommends that you use the `IdentityReplicationPolicy` instead.
• Other than implementing, compiling, and packaging the policy, you also need to carry out advanced configuration steps to use the policy.
• Replication monitoring with this policy is not supported. This means that you will not be able to validate or monitor replications during the migration process.
• This replication policy is only supported with a unidirectional data replication setup where replication happens from a single source cluster to a single target cluster. Configuring additional hops or bidirectional replication is not supported and can lead to severe replication issues.
• Using an Streams Replication Manager service configured with this policy for any other scenario than data migration is not supported. Once migration is complete, the Streams Replication Manager instance you set up must be reconfigured to use the `IdentityReplicationPolicy` or `DefaultReplicationPolicy`. Alternatively, you can delete Streams Replication Manager from the cluster.

Migrate Kafka Using the DefaultReplicationPolicy

How to migrate Kafka with Streams Replication Manager using the DefaultReplicationPolicy.

Before you begin

- Be aware that remote topics on the source cluster are renamed during the migration.

  As a result, you must make significant changes to all Kafka producers and consumers. Otherwise, they will not be able to connect to the correct topics in the Cloudera Private Cloud Base cluster. If this behavior is not suitable, review Migrate Kafka Using Streams Replication Manager on page 8 and use the `IdentityReplicationPolicy` or the `MigratingReplicationPolicy` instead.

- Setup and Configure Streams Replication Manager in the Cloudera Private Cloud Base cluster.

  For more information, see Add and Configure Streams Replication Manager and the Streams Replication Manager Configuration Examples. Setup instructions might differ in different versions, ensure that you view the version of the documentation that matches your Runtime version.

Procedure

1. Use the srm-control tool to include every topic and every consumer group.

Including consumer groups is required for checkpointing.

```
srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --a
dd ".*"
```

```
srm-control groups --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --a
dd ".*"
```

2. Validate that data is being migrated.

Use the Cluster Replications page on the Streams Messaging Manager UI to monitor and validate the status of the migration. Alternatively, you can use the various endpoints available on the Streams Replication Manager Service REST API.

3. Wait until replication is caught up.

4. Stop producers.

5. Stop consumers.

6. Reconfigure all consumers to read from Cloudera Private Cloud Base Kafka and apply offset translation using Streams Replication Manager.

7. Start consumers.

8. Reconfigure all producers to write to Cloudera Private Cloud Base Kafka.

The HDF instances of Kafka and Streams Messaging Manager are no longer required.

9. Start producers.

Results

Kafka is migrated. Kafka clients produce and consume from the Cloudera Private Cloud Base cluster. Migration is complete.

Migrate Kafka Using the IdentityReplicationPolicy

How to migrate Kafka with Streams Replication Manager using the IdentityReplicationPolicy.

Before you begin

- Ensure that you have reviewed Migrate Kafka Using Streams Replication Manager on page 8 and understand the limitations and use cases for this policy.

- Setup and configure Streams Replication Manager in the Cloudera Private Cloud Base cluster. For more information, see Add and Configure Streams Replication Manager. Setup instructions might differ in different versions. Ensure that you view the version of the documentation that matches your Runtime version.
- Ensure that Streams Replication Manager is set up to use the `IdentityReplicationPolicy`. This is done differently depending on the Cloudera Runtime version.
    - **7.1.8:** In Cloudera Manager, add the following entry to the Streams Replication Manager's Replication Configs property.

      ```
      replication.policy.class=org.apache.kafka.connect.mirror.IdentityReplica
      tionPolicy
      ```

    - **7.1.9 or higher:** In Cloudera Manager, select the Enable Prefixless Replication property. This property configures Streams Replication Manager to use the `IdentityReplicationPolicy`.
- If you are on Cloudera Runtime 7.1.8, monitoring features provided by the Streams Replication Manager Service are not supported when this policy is in use. This means that you cannot use the Streams Messaging Manager UI or Streams Replication Manager Service REST API to validate that data is being migrated.

Procedure

1. Use the srm-control tool to include every topic and every consumer group in the allowlist.

   Including consumer groups in the allowlist is required for checkpointing.

   ```
   srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --a
   dd ".*"
   ```

   ```
   srm-control groups --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --a
   dd ".*"
   ```

2. Validate that data is being migrated.

   Use the Cluster Replications page on the Streams Messaging Manager UI to monitor and validate the status of the migration. Alternatively, you can use the various endpoints available on the Streams Replication Manager Service REST API. Doing this is only possible if you are on Cloudera Runtime 7.1.9 or higher.

3. Stop producers.

4. Stop consumers.

5. Reconfigure all consumers to read from Cloudera Private Cloud Base Kafka and apply offset translation using Streams Replication Manager.

6. Start consumers.

7. Reconfigure all producers to write to Cloudera Private Cloud Base Kafka.

   The HDF instances of Kafka and Streams Messaging Manager are no longer required.

8. Start producers.

Results
Kafka is migrated. Kafka clients produce and consume from the Cloudera Private Cloud Base cluster. Migration is complete.

Migrate Kafka Using the MigratingReplicationPolicy

How to migrate Kafka with Streams Replication Manager using the `MigratingReplicationPolicy`, which is a custom replication policy that you must implement, compile, and package yourself.

Before you begin

- Ensure that you have reviewed Migrate Kafka Using Streams Replication Manager on page 8 and understand the limitations and use cases for this policy.

- Setup and Configure Streams Replication Manager in the Cloudera Private Cloud Base cluster for unidirectional replication.

  You can configure unidirectional replication by adding and enabling a single replication in the Streams Replication Manager's Replication Configs property. For example:

  ```
  HDF->CDP.enabled=true
  ```

  For more information on setup and configuration, see Add and Configure Streams Replication Manager. Setup instructions might differ in different versions. Ensure that you view the version of the documentation that matches your Runtime version.

Procedure

1. Implement, compile, and package (JAR) the following custom replication policy that overrides Streams Replication Manager default behavior.

   > ⚠ **Important:** Make sure that all required dependencies, including org.apache.kafka.connect.mirror, are added to your project. If you get a package org.apache.kafka.common does not exist error when creating the JAR, then the org.apache.kafka.connect.mirror dependency is missing. For more information, see the following Knowledge Base article: *ERROR: " :package org.apache.kafka.common does not exist:" when creating a jar for setting up unidirectional data replication.*

   ```java
   package com.cloudera.dim.mirror;
   import java.util.Map;
   import org.apache.kafka.common.Configurable;
   import org.apache.kafka.connect.mirror.ReplicationPolicy;
   import org.apache.kafka.connect.mirror.MirrorConnectorConfig;

   public class MigratingReplicationPolicy implements ReplicationPolicy, Configurable {
       private String sourceClusterAlias;

       @Override
       public void configure(Map<String, ?> props) {
           // The source cluster alias cannot be determined just by looking
    at the prefix of the remote topic name.
           // We extract this info from the configuration.
           sourceClusterAlias = (String) props.get(MirrorConnectorConfig.S
   OURCE_CLUSTER_ALIAS);
       }

       @Override
       public String formatRemoteTopic(String sourceClusterAlias, String to
   pic) {
           // We do not apply any prefix.
           return topic;
       }

       @Override
       public String topicSource(String topic) {
           // return from config
           return sourceClusterAlias;
       }

       @Override
       public String upstreamTopic(String topic) {
           return null;
       }
   }
   ```

2. Modify the classpath of the Streams Replication Manager driver to include the compiled artifact when the Streams Replication Manager Driver is started.

   This step is done differently depending on the Cloudera Runtime version.

   > **For 7.1.6 or lower**
   >
   > ⚠️ **Important:** Complete the following on all hosts that Streams Replication Manager is deployed on.
   >
   > **a.** Find the srm-driver script located at /opt/cloudera/parcels/CDH/lib/streams_replication_manager/bin/srm-driver.
   >
   > **b.** Modify the -cp flag in the srm-driver script to include the additional JAR file. For example:
   >
   > ```
   > exec $JAVA $SRM_HEAP_OPTS $SRM_JVM_PERF_OPTS $SRM_KERBEROS_OPTS
   > $GC_LOG_OPTS $SRM_JMX_OPTS -DdefaultConfig=$SRM_CONFIG_DIR/srm.prope
   > rties -DdefaultYaml=$SRM_CONFIG_DIR/srm-service.yaml -cp [***PATH TO
   >  POLICY JAR***]:$SRM_LIB_DIR/srm-driver-1.0.0.7.1.1.0-567.jar:$SRM_
   > LIB_DIR/srm-common-1.0.0.7.1.1.0-567.jar:...
   > ```
   >
   > **For 7.1.7 or higher**
   >
   > **a.** In Cloudera Manager, select the Streams Replication Manager service.
   >
   > **b.** Go to Configuration.
   >
   > **c.** Add the following key and value pair to both Streams Replication Manager Service Environment Advanced Configuration Snippet (Safety Valve) and SRM Driver Environment Advanced Configuration Snippet (Safety Valve).
   >
   > ```
   > Key:SRM_CLASSPATH
   > Value:[***PATH TO POLICY JAR***]
   > ```
   >
   > If you have other artifacts added to SRM_CLASSPATH, ensure that each path is delimited with a colon (:).

3. Configure the Streams Replication Manager service to use the custom replication policy:

   a) In Cloudera Manager, select the Streams Replication Manager service.

   b) Go to Configuration.

   c) Find the Streams Replication Manager's Replications Config property and add the following:

   ```
   replication.policy.class=com.cloudera.dim.mirror.MigratingReplicationPol
   icy
   ```

   Setting the replication.policy.class property configures Streams Replication Manager to use the custom replication policy instead of the default one.

   d) Click Save Changes.

   e) Restart the service.

4. Use the srm-control tool to include every topic and every consumer group in the allowlist.

   Including consumer groups in the allowlist is required for checkpointing.

   ```
   srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --a
   dd ".*"
   ```

   ```
   srm-control groups --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --a
   dd ".*"
   ```

5. Stop producers.

6. Stop consumers.

7. Reconfigure all consumers to read from Cloudera Private Cloud Base Kafka and apply offset translation using Streams Replication Manager.

8. Start consumers.

9. Reconfigure all producers to write to Cloudera Private Cloud Base Kafka.

   The HDF instances of Kafka and Streams Replication Manager are no longer required.

10. Start producers.

Results

Kafka is migrated. Kafka clients produce and consume from the Cloudera Private Cloud Base cluster. Migration is complete.

Related Information

ERROR: " :package org.apache.kafka.common does not exist:" when creating a jar for setting up unidirectional data replication