

# Migrating Streaming Workloads to CDP Private Cloud

Date published: 2019-08-22

Date modified: 2023-09-07



# 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

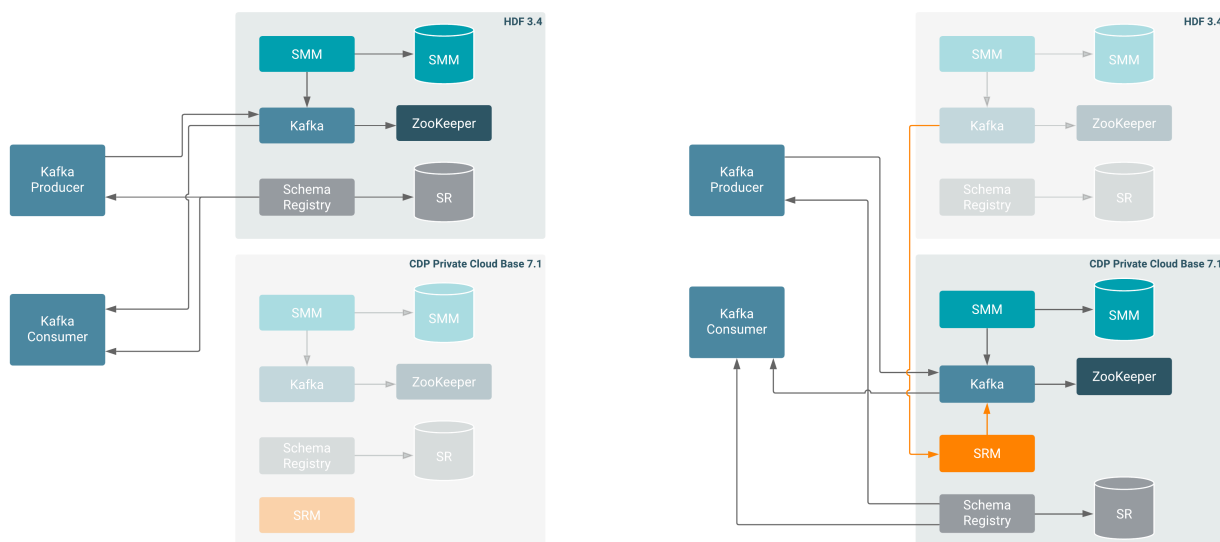
<b>Migrating Streaming workloads from HDF to CDP Private Cloud Base.....</b>	<b>4</b>
Set Up a New Streaming Cluster in CDP Private Cloud Base.....	4
Migrate Ranger Policies.....	4
Migrate Schema Registry.....	5
Copy Raw Data.....	5
Reuse Existing Storage.....	6
Migrate Streams Messaging Manager.....	6
Copy Raw Data.....	7
Reuse Existing Database.....	7
Migrate Kafka Using Streams Replication Manager.....	8
Migrate Kafka Using the DefaultReplicationPolicy.....	9
Migrate Kafka Using the IdentityReplicationPolicy.....	10
Migrate Kafka Using the MigratingReplicationPolicy.....	11

# Migrating Streaming workloads from HDF to CDP Private Cloud Base

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

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

Multiple methods are provided for migrating SMM and Schema Registry. Kafka is migrated with SRM set up on the target CDP Private Cloud Base 7.1 cluster. Multiple configuration methods are provided for the setup and configuration of SRM.



Complete the following steps in order to migrate your streaming workloads:

## Set Up a New Streaming Cluster in CDP Private Cloud Base

How to set up a new Streaming cluster in CDP 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 CDP Private Cloud Base. See the CDP 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 CDP 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](#)

## Migrate Schema Registry

Overview of the methods that can be used to migrate Schema Registry from HDF to CDP 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 CDP 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 CDP Private Cloud Base by reusing existing storage.

### Before you begin

Make sure that the existing database is compatible with and supported by CDP 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 CDP Private Cloud Base.

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



**Warning:** SMM 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 SMM 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 SMM documentation.

### Related Information

[Managing Alert Policies and Notifiers](#)

## Copy Raw Data

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

### Procedure

1. Stop the HDF Streams Messaging Manager (SMM).
2. Backup/restore the SMM 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.
3. Configure CDP SMM to connect to the new database:
  - a) In Cloudera Manager select the SMM 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

SMM 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 CDP Private Cloud Base by reusing existing storage.

## Before you begin

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

## Procedure

1. Stop the HDF Streams Messaging Manager (SMM).
2. Configure CDP SMM to connect to the database previously owned by HDF SMM:
  - a) In Cloudera Manager select the SMM 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

SMM 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 CDP Private Cloud Base using Streams Replication Manager (SRM).

Kafka data is migrated from HDF to CDP Private Cloud Base using SRM. SRM 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 `IdentityReplicationPolicy`, 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 SRM 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 SRM service you set up for migration and continue using it for replication.
- If you want to continue using SRM after migration, review the limitations of this policy in the SRM 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 SRM 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 SRM service configured with this policy for any other scenario than data migration is not supported. Once migration is complete, the SRM instance you set up must be reconfigured to use the `IdentityReplicationPolicy` or `DefaultReplicationPolicy`. Alternatively, you can delete SRM from the cluster.

## Migrate Kafka Using the DefaultReplicationPolicy

How to migrate Kafka with Streams Replication Manager (SRM) 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 CDP 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 SRM in the CDP Private Cloud Base cluster.

For more information, see [Add and Configure SRM](#) and the [SRM 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] --add ".*"
```

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

2. Validate that data is being migrated.

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

3. Wait until replication is caught up.
4. Stop producers.
5. Stop consumers.
6. Reconfigure all consumers to read from CDP Private Cloud Base Kafka and apply offset translation using SRM.
7. Start consumers.
8. Reconfigure all producers to write to CDP Private Cloud Base Kafka.

The HDF instances of Kafka and SMM are no longer required.

9. Start producers.

## Results

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

## Migrate Kafka Using the IdentityReplicationPolicy

How to migrate Kafka with Streams Replication Manager (SRM) 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 CDP Private Cloud Base cluster. For more information, see [Add and Configure SRM](#). Setup instructions might differ in different versions. Ensure that you view the version of the documentation that matches your Runtime version.
- Ensure that SRM 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.IdentityReplicationPolicy
```

- **7.1.9 or higher:** In Cloudera Manager, select the Enable Prefixless Replication property. This property configures SRM to use the IdentityReplicationPolicy.

- If you are on Cloudera Runtime 7.1.8, monitoring features provided by the SRM Service are not supported when this policy is in use. This means that you cannot use the SMM UI or SRM 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] --add ".*"
```

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

2. Validate that data is being migrated.

Use the Cluster Replications page on the SMM UI to monitor and validate the status of the migration.

Alternatively, you can use the various endpoints available on the SRM 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 CDP Private Cloud Base Kafka and apply offset translation using SRM.
6. Start consumers.
7. Reconfigure all producers to write to CDP Private Cloud Base Kafka.

The HDF instances of Kafka and SMM are no longer required.

8. Start producers.

## Results

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

## Migrate Kafka Using the MigratingReplicationPolicy

How to migrate Kafka with Streams Replication Manager (SRM) 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 SRM in the CDP 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 SRM](#). 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 SRM's 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*.

```
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.SOURCE_CLUSTER_ALIAS);
    }

    @Override
    public String formatRemoteTopic(String sourceClusterAlias, String topic) {
        // 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 SRM driver to include the compiled artifact when the SRM 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 SRM 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 SRM 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 SRM 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 CDP Private Cloud Base Kafka and apply offset translation using SRM.
8. Start consumers.

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

The HDF instances of Kafka and SMM are no longer required.

10. Start producers.

### Results

Kafka is migrated. Kafka clients produce and consume from the CDP 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