Cloudera Runtime 7.2.6

Configuring Streams Replication Manager

Date published: 2019-09-13
Date modified: 2020-12-11

https://docs.cloudera.com/

Legal Notice

Contents

Enable high availability for Streams Replication Manager

Streams Replication Manager is capable of running in high availability mode. This can be enabled by deploying multiple instances of the driver and service role in a cluster.

Streams Replication Manager (SRM) is capable of running in high availability mode. By enabling high availability mode for SRM, you can ensure that the replication of data and the monitoring of replication continues even in the case of host failure. To enable SRM high availability, you have to deploy multiple instances of the driver and service roles on the hosts in a cluster. In CDP Public Cloud this can be done when provisioning a new cluster with Data Hub.

Note: Note: Expect an increased load when running SRM in high availability mode.

Enable high availability mode in CDP Public Cloud

In CDP Public Cloud, high availability mode can be enabled by provisioning a Data Hub cluster that has multiple instances of the SRM driver and service. In terms of high availability, the default Streams Messaging cluster definitions behave in the following way:

Streams Messaging Light Duty

High availability mode is enabled by default. Any cluster that you provision with this definition will start SRM in high availability mode. Note that it is only the driver that is started in high availability mode, the service role is only provisioned on a single host. This however, is still considered as a highly available deployment of SRM.

Streams Messaging Heavy Duty

In the heavy duty definition, SRM has its own host group. However, by default, the SRM host group is not provisioned. When provisioning a cluster with the heavy duty definition, you can decide how many nodes this host group should have. To provision SRM in high availability mode with this cluster definition, you have to set the instance count of the SRM nodes host group to at least 2.

For more information on creating a Streams Messaging cluster with Data Hub, see Creating your first Streams Messaging cluster.

Related Information

Creating your first Streams Messaging cluster

Configuring clusters and replications

You can expand an existing deployment of Streams Replication Manager by adding new clusters and replications to the configuration. To do this, you need to specify cluster aliases and cluster connection information, as well as add and enable replications.

About this task

Specifying your clusters and enabling replications does not start replication of data itself. When clusters and replications are added with the following method to the configuration, SRM will connect and set up communication with them, but will not automatically replicate any data. To start replicating data you need to specify which topics to replicate with the srm-control command line tool.

Use the following steps as reference when you want to add new clusters or replications to your deployment.

Before you begin

- If you are planning on replicating data to or from a Kafka service running in either a CDH 5.x or 6.x cluster and you are using Sentry for authorization, make sure that the streamsrepmgr user is added to the Kafka Super users property. You can find the Super users property by going to Kafka service Configuration . Do this on all CDH 5.x or 6.x clusters where data replication will happen.
- If you are planning on replicating data to or from a Kafka service running in Runtime 7.x and you are using Ranger for authorization, make sure that the streamsrepmgr user has all required permissions assigned to it in Ranger. Do this on all Runtime 7.x clusters where data replication will happen.

Procedure

1. In Cloudera Manager, select Streams Replication Manager.
2. Go to Configuration.
3. Specify cluster aliases:
   a) Find the Streams Replication Manager Cluster alias property.
   b) Add a comma delimited list of cluster aliases. For example:

   ```
   primary, secondary
   ```

   Cluster aliases are arbitrary names defined by the user. Aliases specified here are used in other configuration properties and with the srm-control tool to refer to the clusters added for replication.
4. Specify cluster connection information:
   a) Find the Streams Replication Manager's Replication Configs property.
   b) Click the add button and add new lines for each cluster alias you have specified in the Streams Replication Manager Cluster alias property
   c) Add connection information for your clusters. For example:

   ```
   primary.bootstrap.servers=primary_host1:9092,primary_host2:9092,primary_
   host3:9092
   secondary.bootstrap.servers=secondary_host1:9092,secondary_host2:9092
   ,secondary_host3:9092
   ```

   Each cluster has to be added to a new line. If a cluster has multiple hosts, add them to the same line but delimit them with commas.
5. Add and enable replications:
   a) Find the Streams Replication Manager's Replication Configs property.
   b) Click the add button and add new lines for each unique replication you want to add and enable.
   c) Add and enable your replications. For example:

   ```
   primary->secondary.enabled=true
   secondary->primary.enabled=true
   ```
6. Enter a Reason for change, and then click Save Changes to commit the changes.
7. Restart Streams Replication Manager.

Results

Replicating data to or from the specified clusters is now possible.

What to do next

Use the srm-control tool to kick off replication by adding topics or groups to the allowlist.

Related Information

srm-control

Configuring the driver role target clusters

The Streams Replication Manager Driver role's target clusters are the clusters that the driver is writing data to. You can configure these target clusters for each instance of the driver with the Streams Replication Manager Driver Target Cluster property. Custom configuration of these targets is only recommended in advanced deployments.

About this task

The Streams Replication Manager Driver role is responsible for connecting to the specified clusters and performing replication between them. The driver can be installed on one or more hosts within a cluster.

The clusters the driver connects to are the clusters that you specify with the Streams Replication Manager Cluster alias and Streams Replication Manager's Replication Configs properties.

Target clusters of the driver are clusters that the driver writes data to. By default when the driver is started it will connect to all clusters, gather data from them, and write to all of them. In other words, by default a driver targets all clusters in your configuration. You can limit the number of clusters that each driver targets. This can be done with the Streams Replication Manager Driver Target Cluster property, which allows you to specify which cluster or clusters the driver targets.

When you specify a driver target, the driver still connects to all clusters and gathers data from them, but will only write to the clusters specified.

However, in order for monitoring to function correctly, the driver has to target all clusters taking part in the replication. That is, it has to contain the actual target, the cluster you want to write data to, as well as the source clusters for that target, where data is being pulled from. If the source clusters are not specified, you will not be able to monitor your replications. As a result of this, configuring driver targets and limiting the number of clusters each instance of the driver writes to is considered an advanced configuration practice. This practice is only viable in complex replication scenarios that involve a high number clusters and replications. Therefore, Cloudera recommends that you either leave this property empty or add all clusters taking part in the replication.

By default the Streams Replication Manager Driver Target Cluster property is left empty, meaning that all clusters are targeted. The property accepts any cluster alias that is specified in Streams Replication Manager Cluster alias. When adding multiple cluster aliases, delimit them with a comma.

Procedure

1. In Cloudera Manager, select Streams Replication Manager.
2. Go to Configuration.
3. Find the Streams Replication Manager Driver Target Cluster property.
4. Add the cluster aliases that you want the driver role to target. For example:

   ```
   primary, secondary
   ```

5. Enter a Reason for change, and then click Save Changes to commit the changes.
6. Restart Streams Replication Manager.

Results

Driver targets are configured. Drivers only write data to the targeted clusters.

Configuring the service role target cluster

The Streams Replication Manager Service role's target cluster is the cluster from which metrics are gathered and exposed. A single target can be configured for each instance of the service with the Streams Replication Manager Service Target Cluster property. Configuration is mandatory.

About this task

The Streams Replication Manager Service role consists of a REST API and a Kafka Streams application that aggregates and exposes cluster, topic, and consumer group metrics. With the help of these metrics, users can monitor replications. The service can only be installed on one host per cluster.

Each instance of the service is associated with a single target cluster. The target is the cluster that the service gathers and exposes metrics from. Because each instance of the service can only target and expose metrics from a single cluster, monitoring multiple clusters requires the deployment of multiple instances of the service.

The target cluster of the service is configured with the Streams Replication Manager Service Target Cluster property. The property accepts any cluster alias that is specified in Streams Replication Manager Cluster alias as long as data is being replicated to that cluster. Configuring a service target is mandatory.

Procedure

1. In Cloudera Manager, select Streams Replication Manager.
2. Go to Configuration.
3. Find the Streams Replication Manager Service Target Cluster property.
4. Add the target cluster alias. For example:

   ```
   secondary
   ```

5. Enter a Reason for change, and then click Save Changes to commit the changes.
6. Restart Streams Replication Manager.

Results

The service target is set. The service gathers and exposes metrics from the specified cluster.

Configuring properties not exposed in Cloudera Manager

There are number of configuration properties that Streams Replication Manager accepts, but are not exposed directly in Cloudera Manager. You can configure these properties with the Streams Replication Manager's Replication Configs property. Additionally these properties can be configured on a top, cluster, or replication level.

About this task

In addition to the configuration properties exposed directly for configuration through Cloudera Manager, Streams Replications Manager accepts a number of additional Streams Replication Manager specific properties as well as Kafka properties available in the version of Kafka that you are using. Properties not exposed directly in Cloudera Manager can be set through the Streams Replication Manager's Replication Configs property. For a comprehensive list of SRM properties not available in Cloudera Manager, see Configuration Properties Reference for Properties not Available in Cloudera Manager. For a comprehensive list of Kafka client properties, see the upstream Apache Kafka documentation.

The configuration properties that you add to Streams Replication Manager's Replication Configs can be prefixed. These prefixes allow you to exercise control over when and where a configuration should be used by SRM. The prefixes and the levels of configuration they correspond to are the following:

- Top level (no prefix): Top level or global configuration is achieved by adding the property on its own, without a prefix. A configuration like this will be used For example:

  ```
  replication.factor=3
  ```

- Cluster level (prefix with cluster alias): Cluster level configuration can be achieved by prefixing the configuration property with a cluster alias specified in Streams Replication Manager Cluster alias. For example:

```
primary.replication.factor=3
```

- Replication level (prefix with replication):Replication level configuration can be achieved by prefixing the configuration property with the name of the replication. For example:

```
primary->secondary.replication.factor=3
```

In addition to these prefixes, there also exist two other prefixes that can be used to set configuration properties for the dedicated Kafka Connect REST servers set up by the SRM driver for each replication. These are ***ALIAS***->** *ALIAS***.worker. and listeners.https.. For more information on the usage of these prefixes and the configuration of the dedicated REST servers, see Configuring replication specific Kafka Connect REST servers.

Before you begin

Make sure that cluster aliases and replications are configured. Otherwise cluster or replication level configuration is not possible.

Procedure

1. In Cloudera Manager, select Streams Replication Manager.
2. Go to Configuration.
3. Configure properties not exposed in Cloudera Manager:
   a) Find the Streams Replication Manager's Replication Configs property.
   b) Click the add button and add new lines for each additional property you want to configure.
   c) Add configuration properties. For example:

```
replication.factor=3
```

4. Enter a Reason for change, and then click Save Changes to commit the changes.
5. Restart Streams Replication Manager.

Results

Configuration properties not directly exposed in Cloudera Manager are configured.

Related Information

Configuration Properties Reference for Properties not Available in Cloudera Manager

Configuring replication specific Kafka Connect REST servers

Configuring replication specific Kafka Connect REST servers

The SRM Driver role starts a dedicated Kafka Connect REST server for each replication that you set up and configure. These REST servers make communication possible between the different instances of the SRM driver. Communication between the driver instances in turn ensures that replication does not fail when there are multiple driver instances present in a single cluster.

About this task

If required, you can configure each replication’s dedicated REST server that is set up by the driver.

REST server properties can be configured through the Streams Replication Manager's Replication Configs property by using two specific prefixes. These prefixes are the following:

*****ALIAS***->***ALIAS***.worker.**

This prefix can be used to set any Kafka Connect REST server property for a replication's dedicated REST server. The first element of the prefix determines which replication's REST server is being configured. For example, the primary->secondary.worker. prefix can be used to configure the prim ary->secondary replication's REST server. While configuration of all REST server properties is supported, the main use case for this prefix is to set the ports of the REST servers. The reason for this is that by default the REST servers will bind to port 0, that is, any available port. This behaviour may not be suitable for your deployment.

Important: The rest.host.name and rest.port properties should not be used with this prefix. Use the listeners property instead if you want to configure ports.

listeners.https.

This prefix can be used to configure SSL related properties. You can use this prefix to override the SSL settings that the REST server inherits from the service configuration. For example, if the REST server needs to use a different keystore location than the one provided in the service configuration, you can use the following property:

```
listeners.https.ssl.keystore.location=***CUSTOM KEYSTORE LOCATIO
N***
```

Procedure

1. In Cloudera Manager, select the Streams Replication Manager service.
2. Go to Configuration.
3. Find the Streams Replication Manager's Replication Configs property.
4. Click the add button and add new lines for each additional property you want to configure.
5. Add configuration properties. For example:

```
primary->secondary.worker.listeners=HTTPS://myhost:8084
listeners.https.ssl.keystore.location=***CUSTOM KEYSTORE LOCATION***
```

6. Enter a Reason for change, and then click Save Changes to commit the changes.
7. Restart Streams Replication Manager.

Results

Custom configuration of the Kafka Connect REST server is complete.

Configuring automatic group offset synchronization

Automatic group offset synchronization is a feature in Streams Replication Manager (SRM) that automates the export and application of translated consumer group offsets. Enabling this feature can simplify the manual steps that you need to take to migrate consumer groups in a failover or failback scenario.

About this task

SRM automatically translates consumer group offsets between clusters. While the offset mappings are created by SRM, they are not applied by default to the consumer groups of the target cluster. As a result, by default, migrating consumer groups from one cluster to another involves running the srm-control offsets and kafka-consumer-groups tools. The srm-control offsets tool exports translated offsets, kafka-consumer-groups resets and applies the translated offsets on the target cluster.

This process can be automated by enabling automatic consumer group synchronization. If automatic group offset synchronization is enabled, the translated group offsets of the source cluster are automatically exported from the

source cluster and are applied on the target cluster (they are written to the __consumer_offsets topic). If you choose to enable this feature, running srm-control offsets and kafka-consumer-groups is not required to migrate consumer groups. You only need to restart and redirect consumers to consume from the new cluster.

Although automatic consumer group synchronization can simplify migrating consumer groups, ensure that you understand the following about its behavior:

- Automatic consumer group offset synchronization does not fully automate a failover or failback process. It only allows you to skip certain manual steps required in the process. Consumers must be restarted and redirected to the new cluster even if the feature is enabled.
- Offsets are synced at a configured interval. As a result, it is not guaranteed that the latest translated offsets are applied. If you want to have the latest offsets applied, Cloudera recommends that you export and apply consumer group offsets manually instead. The exact interval depends on sync.group.offsets.interval.seconds and emit.checkpoints.interval.seconds.
- The checkpointing frequency configured for SRM can have an effect on the group offset synchronization frequency. The frequency of group offset synchronization can be configured with sync.group.offsets.interval.seconds. However, specifying an interval using this property might not result in the offsets being synchronized at the set frequency. This depends on how emit.checkpoints.interval is configured. The emit.checkpoints.interval property specifies how frequently offset information is fetched (checkpointing). Because offset synchronization can only happen after offset information is available, the frequency configured with the emit.checkpoints.interval might introduce additional latency. For example, if you set offset synchronization to 60 seconds (default), but have checkpointing set to 120 seconds, offsets will only be synchronized every two minutes.
- Offsets are only synched for the consumers that are inactive in the target cluster. This is done so that the SRM does not override the offsets in the target cluster.

Procedure

1. In Cloudera Manager, select Streams Replication Manager.
2. Go to Configuration.
3. Find the Streams Replication Manager's Replication Configs and add the following configuration entries:

   ```
   sync.group.offsets.enabled = true
   sync.group.offsets.interval.seconds = [***TIME IN SECONDS***]
   ```

   In this example, all properties are set on a global level. This means that automatic group offset synchronization is applied to all replications. Depending on your setup, you can also choose to set these properties for specific replications only using appropriate replication prefixes.

   The backup.sync.group.offsets.enabled property enable automatic group offset synchronization. The sync.group.offsets.interval.seconds property controls how frequently offsets are synched. You only need to specify this property if you want to customize synchronization frequency.
4. Enter a Reason for change, and then click Save Changes to commit the changes.
5. Restart Streams Replication Manager.

Results

Auto group offset synchronization is enabled. From now on, SRM automatically exports the translated offsets of the configured source clusters and applies them to the configured target clusters.

What to do next

New topic and consumer group discovery

Kafka topics or consumer groups may not get replicated instantly when they are added to white and blacklists. This is due to the default behaviour of how topics and consumer groups are discovered by Streams Replication Manager.

The discovery and replication of newly created topics or consumer groups is not instantaneous. Streams Replication Manager checks source clusters for new topics and consumer groups periodically, as controlled by the Refresh Topics Interval Seconds and Refresh Groups Interval Seconds properties. By default both properties are set to 10 minutes. As a result, the discovery and replication of new topics or groups can take up to 10 minutes.

Cloudera does not recommend using a refresh interval lower than the default value for production environments as it can lead to severe performance degradation.

Related Information

srm-control Topics and Groups Subcommand

Configuration examples

These configuration examples give step-by-step instructions on how you can set up and configure typical deployments of Streams Replication Manager. Reviewing these examples can help you gain a better understanding of how your specific setup can be configured.

Bidirectional replication example of two active clusters

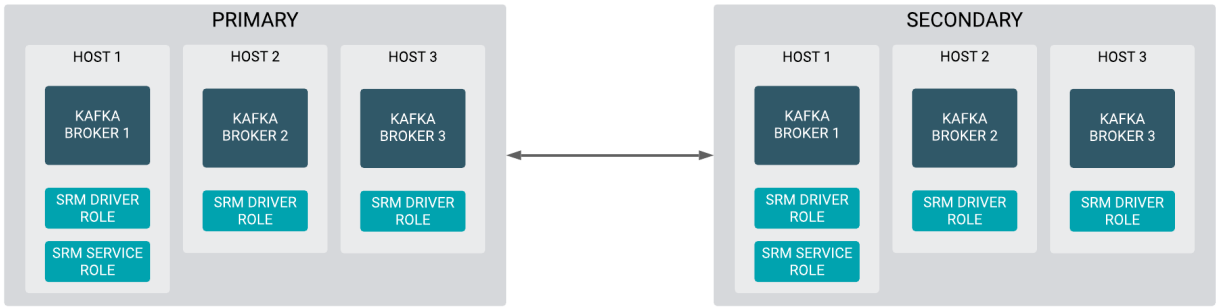
Review the bidirectional replication example to learn how you can configure and start replication with Streams Replication Manager in a deployment with two active clusters configured with bidirectional replication.

About this task

In a typical scenario, you may have two active Kafka clusters within the same region but in separate availability zones. With bidirectional replication, clients can connect to either cluster in case one is temporarily unavailable.

This example demonstrates the steps required to configure the deployment shown below. Additionally, it also provides example commands to start replication between clusters.

Figure 1: Bidirectional Replication of Active Clusters



The steps shown here have to be carried out on all clusters in a given deployment. Configuration properties presented in Steps 3-5 are configured identically on all clusters. The configuration property presented in Step 7 will differ for each cluster.

Note: The following list of steps assume that the Streams Replication Manager Service role is running on 1 host on each cluster and is targeting the cluster it is running on.

Procedure

1. In Cloudera Manager, select Streams Replication Manager.
2. Go to Configuration.
3. Specify cluster aliases:
   a) Find the Streams Replication Manager Cluster alias property.
   b) Add a comma delimited list of cluster aliases. For example:

   ```
   primary, secondary
   ```

   Cluster aliases are arbitrary names defined by the user. Aliases specified here are used in other configuration properties and with the srm-control tool to refer to the clusters added for replication.
4. Specify cluster connection information:
   a) Find the Streams Replication Manager's Replication Configs property.
   b) Click the add button and add new lines for each cluster alias you have specified in the Streams Replication Manager Cluster alias property
   c) Add connection information for your clusters. For example:

   ```
   primary.bootstrap.servers=primary_host1:9092,primary_host2:9092,primary_
   host3:9092
   secondary.bootstrap.servers=secondary_host1:9092,secondary_host2:9092
   ,secondary_host3:9092
   ```

   Each cluster has to be added to a new line. If a cluster has multiple hosts, add them to the same line but delimit them with commas.
5. Add and enable replications:
   a) Find the Streams Replication Manager's Replication Configs property.
   b) Click the add button and add new lines for each unique replication you want to add and enable.
   c) Add and enable your replications. For example:

   ```
   primary->secondary.enabled=true
   secondary->primary.enabled=true
   ```
6. Enter a Reason for change, and then click Save Changes to commit the changes.

7. Add Streams Replication Manager Driver role instances to all Kafka broker hosts:
   a) Go to Instances.
   b) Click Add Role Instances.
   c) Click Select Hosts.
   d) Select all Kafka broker hosts and click Ok.
   e) Click Continue.
   f) Find the Streams Replication Manager Driver Target Cluster property.
   g) Add the cluster aliases that you want the driver role to target. For example:
      - On the primary cluster:

        ```
        primary
        ```

      - On the secondary cluster:

        ```
        secondary
        ```

      The Streams Replication Manager Driver Target Cluster property allows you to specify which clusters the driver should write to. In this example, the drivers read data from all clusters, but only write to the cluster they are running on. This allows you to distribute replication workloads.
   h) Click Continue.
8. Restart Streams Replication Manager.
9. Replicate data between clusters with the following commands:

   ```
   srm-control topics --source primary --target secondary --add ".*"
   ```

   ```
   srm-control topics --source secondary --target primary --add ".*"
   ```

Cross data center replication example of multiple clusters

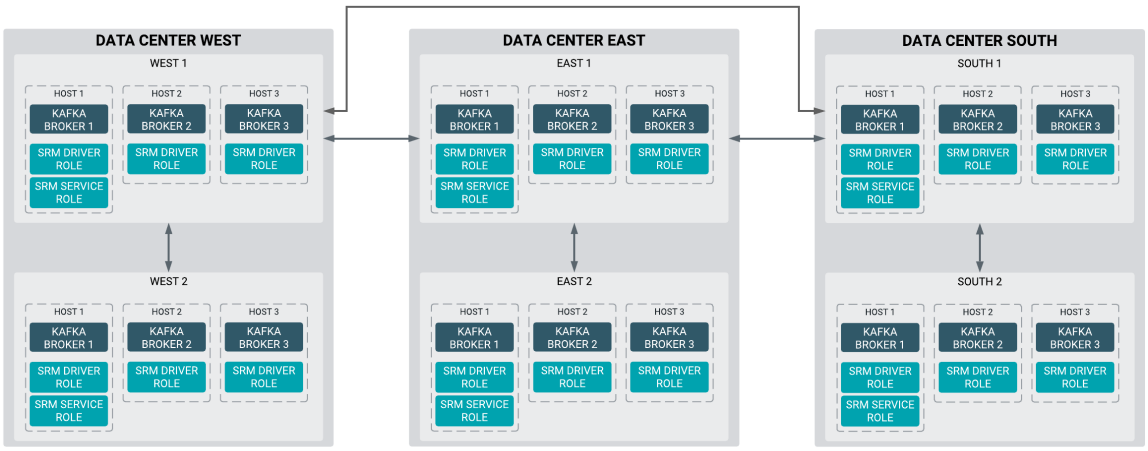
Review the cross data center replication example to understand how you can configure and start replication with Streams Replication Manager in a deployment with three data centers that each have two Kafka clusters.

About this task

In more advanced deployments, you may have multiple Kafka clusters in each of several data centers. To prevent creating a fully-connected mesh of all Kafka clusters, Cloudera recommends leveraging a single Kafka cluster in each data center for cross data center replication.

This example demonstrates the steps required to configure the deployment shown below. Additionally, it also provides example commands to start bidirectional replication of all topics within each data center and an example on how to replicate a single topic across all data centers.

Figure 2: Cross Data Center Replication of Multiple Clusters



The steps shown here have to be carried out on all clusters in a given deployment. Configuration properties presented in Steps 3-5 are configured identically on all clusters. The configuration property presented in Step 7 will differ for each cluster.

Note: The following list of steps assume that the Streams Replication Manager Service role is running on 1 host on each cluster and is targeting the cluster it is running on.

Procedure

1. In Cloudera Manager, select Streams Replication Manager.
2. Go to Configuration.
3. Specify cluster aliases:
   a) Find the Streams Replication Manager Cluster alias property.
   b) Add a comma delimited list of cluster aliases. For example:

   ```
   west1, west2, east1, east2, south1, south2
   ```

   Cluster aliases are arbitrary names defined by the user. Aliases specified here are used in other configuration properties and with the srm-control tool to refer to the clusters added for replication.
4. Specify cluster connection information:
   a) Find the Streams Replication Manager's Replication Configs property.
   b) Click the add button and add new lines for each cluster alias you have specified in the Streams Replication Manager Cluster alias property
   c) Add connection information for your clusters. For example:

   ```
   west1.bootstrap.servers=west1_host1:9092,west1_host2:9092,west1_host3:90
   92
   west2.bootstrap.servers=west2_host1:9092,west2_host2:9092,west2_host3
   :9092

   east1.boostrap.servers=east1_host1:9092,east1_host2:9092,east1_host3:
   9092
   east2.boostrap.servers=east2_host1:9092,east2_host2:9092,east2_host3:90
   92

   south1.boostrap.servers=south1_host1:9092,south1_host2:9092,south1_host3
   :9092
   ```

```
south2.boostrap.servers=south2_host1:9092,south2_host2:9092,south2_hos
t3:9092
```

Each cluster has to be added to a new line. If a cluster has multiple hosts, add them to the same line but delimit them with commas.

5. Add and enable replications:
   a) Find the Streams Replication Manager's Replication Configs property.
   b) Click the add button and add new lines for each unique replication you want to add and enable.
   c) Add and enable your replications. For example:

      Enable cross data center replication by adding the following replications:

      ```
      west1->east1.enabled=true
      west1->south1.enabled=true
      east1->west1.enabled=true
      east1->south1.enabled=true
      south1->west1.enabled=true
      south1->east1.enabled=true
      ```

      Enable bidirectional replication within each data center by adding the following replications:

      ```
      west1->west2.enabled=true
      west2->west1.enabled=true
      east1->east2.enabled=true
      east2->east1.enabled=true
      south1->south2.enabled=true
      south2->south1.enabled=true
      ```

6. Enter a Reason for change, and then click Save Changes to commit the changes.
7. Add Streams Replication Manager Driver role instances to all Kafka broker hosts:
   a) Go to Instances.
   b) Click Add Role Instances.
   c) Click Select Hosts.
   d) Select all Kafka broker hosts and click Ok.
   e) Click Continue.
   f) Find the Streams Replication Manager Driver Target Cluster property.
   g) Add the cluster aliases that you want the driver role to target. For example:
      - In the west data center:

        ```
        west1, west2
        ```

      - In the east data center:

        ```
        east1, east2
        ```

      - In the south data center:

        ```
        south1, south2
        ```

      The Streams Replication Manager Driver Target Cluster property allows you to specify which clusters the driver should write to. In this example, the drivers read data from all clusters, but only write to the cluster they are running on. This allows you to distribute replication workloads.
   h) Click Continue.
8. Restart Streams Replication Manager.

9. Replicate topics between hosts within each data center:

```
srm-control topics --source west1 --target west2 --add ".*"
```

```
srm-control topics --source west2 --target west1 --add ".*"
```

```
srm-control topics --source east1 --target east2 --add ".*"
```

```
srm-control topics --source east2 --target east1 --add ".*"
```

```
srm-control topics --source south1 --target south2 --add ".*"
```

```
srm-control topics --source south2 --target south1 --add ".*"
```

10. Replicate topic1 across all data centers:

```
srm-control topics --source west1 --target east1 --add topic1,west2.topic1
```

```
srm-control topics --source west1 --target south1 --add topic1,west2.top
ic1
```

```
srm-control topics --source east1 --target west1 --add topic1,east2.topic1
```

```
srm-control topics --source east1 --target south1 --add topic1,east2.top
ic1
```

```
srm-control topics --source south1 --target west1 --add topic1,south2.to
pic1
```

```
srm-control topics --source south1 --target east1 --add topic1,south2.to
pic1
```