Cloudera Runtime 7.1.1

Using Streams Replication Manager

Date published: 2019-09-13
Date modified: 2020-05-22

https://docs.cloudera.com/

Legal Notice

Contents

SRM Command Line Tools

Overview of the command line tools shipped with SRM.

Important: SRM ships with the srm-control, srm-driver, srm-service, and srm command line tools. Use the srm-control tool to the manage replication of topics and consumer groups. Do not use the srm-driver, srm-service, and srm tools, use Cloudera Manager instead to manage the SRM driver and service.

srm-control

Learn how to use the srm-control command line tool which is used to manage replication of topics and consumer groups.

The srm-control tool enables users to manage replication of topics and consumer groups. The tool has three subcommands topics, groups, and offsets. The topics subcommand is used to control which topics are replicated. The groups subcommand is used to control which consumer groups are replicated. The offsets subcommand is used to export translated offsets for a source->target cluster pair.

A full list of the available options for the tool can be found in srm-control Options Reference.

The srm-control command line tool is located in /opt/cloudera/parcels/CDH/bin. An alternative is provided for it by default, it is called srm-control. Simply typing srm-control will invoke the tool, you do not need to reference its full path.

Important: Do not use the srm-driver, srm-service, and srm command line tools which are also located in the bin directory. Instead, use Cloudera Manager to manage the SRM driver and service.

Important: The srm-control tool requires security configuration before use. This configuration is required for both secured and unsecured environments. Without proper configuration, the tool will not run.

Related Information

srm-control Options Reference

Configuring srm-control

Security configuration is required before you can start using the srm-control-tool to manage the replication of topics and consumer groups. Configuration can be done with Cloudera Manager or by setting environment variables.

The srm-control tool requires security configuration prior to use. There are two methods of configuration, you can either use Cloudera Manager or environment variables.

The Cloudera Manager method involves logging into the Cloudera Manager instance managing your cluster and setting the required properties within the UI. The environment variable method involves logging into one of the Streams Replication Manager hosts and setting environment variables on the command line.

Which properties you have to configure depends on the security setup of your cluster.

Important: Configuration is required for both secured and unsecured environments. Without configuration, the tool will not be able to run and you will not be able to start replication.

Configure srm-control for unsecured environments using environment variables

Prior to using the srm-control tool, you must configure the properties required for your security setup. You can complete this task by setting environment variables.

About this task

To use the srm-control tool in unsecured environments, you need to set the security protocol to PLAINTEXT. Complete the following steps to configure the security protocol using environment variables.

Procedure

1. Log in to the Streams Replication Manager driver host.

```
ssh [MY_SRM_DRIVER_HOST]
```

2. Set the security_protocol environment variable to PLAINTEXT.

```
export security_protocol=PLAINTEXT
```

Results

The srm-control tool is configured.

What to do next

Use the srm-control tool to manage topic and consumer group replication.

Configure srm-control for unsecured environments using Cloudera Manager

Prior to using the srm-control tool, you must configure the properties required for your security setup. You can complete this task with Cloudera Manager.

About this task

To use the srm-control tool in unsecured environments, you need to set the security protocol to PLAINTEXT. Complete the following steps to configure the security protocol using Cloudera Manager.

Procedure

1. In Cloudera Manager select Streams Replication Manager.
2. Go to Configuration.
3. Find the Streams Replication Manager's Replication Configs property.
4. Add the following to the property in a new line:

```
security.protocol = PLAINTEXT
```

5. Click Save Changes.
6. Restart Streams Replication Manager.

Results

The srm-control tool is configured.

What to do next

Use the srm-control tool to manage topic and consumer group replication.

Configure srm-control for secured environments using environment variables

Prior to using the srm-control tool, you must configure the properties required for your security setup. You can complete this task by setting environment variables.

About this task

To use the srm-control tool in secured environments, you need to ensure that all relevant security properties are configured. Complete the following steps to configure security properties using environment variables.

Procedure

1. Log in to the Streams Replication Manager driver host.

```
ssh [MY_SRM_DRIVER_HOST]
```

2. Configure security properties with environment variables.

   The following example contains all possible environment variables with values. Use the variables required for your environment.

   Note: Instead of running the export commands individually, Cloudera recommends that you store them in a protected file and directory and source the file when the security properties need to be set.

```
export SRM_KERBEROS_OPTS="-Djavax.security.auth.useSubjectCredsOnly=false
 -Djava.security.auth.login.config=/path/to/jaas.conf"
export security_protocol=SASL_SSL
export ssl_truststore_location=/path/to/truststore.jks
export ssl_truststore_password=password123
export ssl_keystore_location=/path/to/keystore.jks
export ssl_keystore_password=password123
export ssl_key_password=password123
```

Results

The srm-control tool is configured.

What to do next

Use the srm-control tool to manage topic and consumer group replication.

Configure srm-control for secured environments using Cloudera Manager

Prior to using the srm-control tool, you must configure the properties required for your security setup. You can complete this task with Cloudera Manager

About this task

To use the srm-control tool in secured environments, you need to ensure that all relevant security properties are configured. Complete the following steps to configure security properties using Cloudera Manager.

Procedure

1. In Cloudera Manager select Streams Replication Manager.
2. Go to Configuration.
3. Find the Streams Replication Manager Cluster alias property.
4. Add security related properties.

   Note: Depending on how security was set up, the required properties may already be present. If they are, the tool is already configured and ready for use. Do not add duplicate properties to the configuration.

   Warning: Passwords entered into the following properties are stored in plaintext.

   The following example lists all possible security properties with values. Add and configure the properties required for your security setup. You may need to add the required properties prefixed with cluster aliases for each cluster that is taking part in replication.

```
security.protocol = SASL_SSL
ssl.truststore.location = /path/to/truststore.jks
```

```
ssl.truststore.password = test1234
ssl.keystore.location = /path/to/keystore.jks
ssl.keystore.password = test1234
ssl.key.password = test1234
sasl.kerberos.service.name = kafka
sasl.mechanism = GSSAPI
sasl.jaas.config = com.sun.security.auth.module.Krb5LoginModule required
 useKeyTab=true keyTab="/path/to/keytab file" storeKey=true useTicketCac
he=false principal="streamsrepmgr@STREAMANALYTICS.COM";
```

5. Click Save Changes.
6. Restart Streams Replication Manager.

Results

The srm-control tool is configured.

What to do next

Use the srm-control tool to manage topic and consumer group replication.

Topics and Groups Subcommand

Learn how to use the topics and groups subcommand of the srm-control command line tool.

Important: The srm-control tool requires security configuration before use. This configuration is required for both secured and unsecured environments. Without proper configuration, the tool will not run.

The topics and groups subcommands are used to manipulate the topic or group allowlist (whitelist) and denylist (blacklist). Both subcommands support the same set of command options.

If you are modifying allow and denylists which target newly created topics or consumer groups, changes made with the srm-control tool may not be instantaneous A topic or group needs to be discovered by SRM before it can be added to or removed from allow or denylists. New topic and group discovery happens every 10 minutes by default. As a result, you may need to wait up to 10 minutes until you can see the changes made.

Add topics or groups to an allowlist:

```
srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --a
dd [TOPIC1],[TOPIC2]
```

```
srm-control groups --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --a
dd [GROUP1],[GROUP2]
```

Remove topics or groups from an allowlist:

```
srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --r
emove [TOPIC1],[TOPIC2]
```

```
srm-control groups --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --r
emove [GROUP1],[GROUP2]
```

Add topics or groups to a denylist (blacklist):

```
srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --a
dd-blacklist [TOPIC1],[TOPIC2]
```

```
srm-control groups --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --a
dd-blacklist [GROUP1],[GROUP2]
```

Remove topics or groups from a denylist:

```
srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --r
emove-blacklist [TOPIC1],[TOPIC2]
```

```
srm-control groups --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --r
emove-blacklist [GROUP1],[GROUP2]
```

Specifying topics or groups is also possible with regular expressions. The following example adds all topics to the allowlist, meaning that every topic on the source cluster will be replicated to the target cluster.

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

In addition to adding or removing items, you can also use the tool to look at the contents of a deny or allowlist.

```
srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --l
ist
```

Client Override Options

The topics and groups subcommands support a number of client override options. Client override options allow users to temporarily specify or override configuration properties used for replication. These options also enable users to issue srm-control commands even if the SRM's configuration file is not available on the host that the command is being issued from. While it is possible to specify a range of properties with the client override options, and they can prove to be a powerful tool in certain scenarios, Cloudera recommends that you use Cloudera Manager to manage client configuration options.

The following client override options are available:

- --bootstrap-servers: Specifies the bootstraps servers.
- --producer-props: Specifies producer configuration properties.
- --consumer-props: Specifies consumer configuration properties.
- --props: Specifies client configuration properties.

Note:

Client override options always take precedence over the configuration set in Cloudera Manager. Additionally, the --producer-props and --consumer-props options take precedence over the --props option.

A simple example of using client override options is when you want to change the bootstrap server. This can be done in two ways.

You can specify the bootstrap server with the --bootstrap-servers option.

```
srm-control --bootstrap-servers localhost:9092 topics --sour
ce [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --list
```

Alternatively, you can also use the --props option together with the bootstrap.servers Kafka property to define the bootstrap server.

```
srm-control --props bootstrap.servers=localhost:9092 topics --so
urce [SOURCE_CLUSTER] --list
```

Related Information

New Topic and Consumer Group Discovery

Offsets Subcommand

Learn how to use the offsets subcommand of the srm-client command line tool.

Important: The srm-control tool requires security configuration before use. This configuration is required for both secured and unsecured environments. Without proper configuration, the tool will not run.

SRM automatically translates consumer group offsets between clusters. The offset mappings are created by SRM, but are not applied to the consumer groups of the target cluster directly. Consumers can be migrated from one cluster to another without losing any progress by using the offsets subcommand on the target cluster to export the translated offsets of the source cluster. For example:

```
srm-control offsets --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --g
roup [GROUP1] --export > out.csv
```

Exported offsets can then be applied to the consumer groups in the target cluster with the kafka-consumer-groups tool. For detailed steps on cluster migration, see *Migrating Consumer Groups Between Clusters*.

Client Override Options

The offset subcommand supports client override options. Client override options allow users to temporarily specify or override configuration properties. These options also enable users to issue srm-control commands even if the SRM's configuration file is not available on the host that the command is being issued from. While it is possible to specify a range of properties with the client override options, and they can prove to be a powerful tool in certain scenarios, Cloudera recommends that you use Cloudera Manager to manage client configuration options.

The following client override options are available:

- --bootstrap-servers: Specifies the bootstraps servers.
- --props: Specifies client configuration properties.

Note:

Client override options always take precedence over the configuration set in Cloudera Manager.

A simple example of using client override options is when you want to change the bootstrap server. This can be done in two ways.

You can specify the bootstrap server with the --bootstrap-servers option.

```
srm-control --bootstrap-servers localhost:9092 offsets --sourc
e [SOURCE_CLUSTER] --group [GROUP] --export > out.csv
```

Alternatively, you can use the --props option together with the bootstrap.servers Kafka property to define the bootstrap server.

```
srm-control --props bootstrap.servers=localhost:9092 offsets --sourc
e [SOURCE_CLUSTER] --group [GROUP] --export > out.csv
```

Related Information

How to Set up Failover and Failback

Migrating Consumer Groups Between Clusters

Monitoring Replication with Streams Messaging Manager

Learn about monitoring SRM replication with Streams Messaging Manager.

Users have the ability to connect SRM with Streams Messaging Manager (SMM) and monitor replications through the SMM UI.This is achieved with the Kafka Streams application and the REST API that come bundled with SRM. The Kafka Streams application calculates and aggregates replication metrics, the REST API exposes these metrics. SMM uses the REST API to display aggregated metrics to the end users, enabling monitoring as a result. Monitoring replication flows in SMM is available starting with version 2.0.0.

For more information regarding the requirements and setup of SRM with SMM, see Monitoring Kafka Cluster Replication using SMM in the SMM guide.

Related Information

Monitoring Cluster Replications Overview

Replicating Data

A step by step guide on how to start replicating data between Kafka clusters with SRM.

About this task

Installing and starting SRM on your cluster does not automatically start data replication. In order to kick off replication, you need to update the whitelists with the srm-control tool.

Before you begin

In Cloudera Manager, verify that the SRM driver role is started and is in good health.

Verify that SRM is configured correctly. Make sure that connection information for each Kafka cluster is added as well as at least one source->target replication is specified and enabled.

Procedure

1. Update the topics whitelist to start data replication.

```
srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --
add [TOPIC1],[TOPIC2]
```

Note: If required, instead of listing the topics that you want to add, you can also use regular expressions to add multiple topics with one command.

2. Verify that the topics have been added to the whitelist.

```
srm-control topics --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --
list
```

Results

The topics you specify with the --add option are added to the topic whitelist and are replicated to the specified target cluster.

How to Set up Failover and Failback

Learn how to prepare for failover and failback scenarios with SRM.

If a primary Kafka cluster is temporarily unavailable, you can migrate mission-critical workloads to a backup Kafka cluster (failover). When the primary cluster is restored, you can migrate back (failback). To prepare for this scenario, ensure SRM is configured with bidirectional replication of mission-critical consumer groups and topics. Then in the case of a disaster scenario you can migrate consumer groups between clusters.

Related Information

Offsets Subcommand

Configure SRM for Failover and Failback

Learn how to configure SRM for failover and failback.

About this task

To prepare for a failover or failback scenario you have to set up SRM with bidirectional replication. Additionally, you have to make sure that all mission critical topics and consumer groups are whitelisted on both the primary and backup clusters.

Procedure

1. In Cloudera Manager select Streams Replication Manager.
2. Go to Configuration.
3. Set up bidirectional replication between clusters:

   

   Note:

   The following example contains the minimum required properties only. For a more in-depth configuration example for a cluster setup with bidirectional replication, see Configuration Examples.

   a) Find the Streams Replication Manager Cluster alias property.
   b) Add a comma delimited list of cluster aliases. For example:

   ```
   primary, secondary
   ```

   c) Find the Streams Replication Manager's Replication Configs property.
   d) Click the add button and add new lines for each cluster alias you have specified in the Streams Replication Manager Cluster alias property
   e) 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.
   f) Click the add button and add new lines for each unique replication you want to add and enable.
   g) Add and enable your replications. For example:

   ```
   primary->secondary.enabled=true
   secondary->primary.enabled=true
   ```

4. Whitelist required consumer groups and topics on the primary cluster.
   a) Whitelist groups:

   ```
   srm-control groups --source [PRIMARY_CLUSTER] --targe
   t [SECONDARY_CLUSTER] --add [GROUP1],[GROUP2]
   ```

   a) Whitelist topics:

   ```
   srm-control topics --source [PRIMARY_CLUSTER] --targe
   t [SECONDARY_CLUSTER] --add [TOPIC1],[TOPIC2]
   ```

5. Whitelist required remote topics and consumer groups on the secondary cluster.

   Important:

   If remote topics and consumer groups are not whitelisted on the secondary cluster, a failback operation will be impossible to carry out.

   a) Whitelist remote groups:

   ```
   srm-control groups --source [SECONDARY_CLUSTER] --targe
   t [PRIMARY_CLUSTER] --add [GROUP1],[GROUP2]
   ```

   b) Whitelist remote topics:

   ```
   srm-control topics --source [SECONDARY_CLUSTER] --targe
   t [PRIMARY_CLUSTER] --a
   dd [PRIMARY_CLUSTER.TOPIC1],[PRIMARY_CLUSTER.TOPIC2]
   ```

6. Verify that all required topics and consumer groups are whitelisted.

   a) Verify consumer groups:

   ```
   srm-control groups --source [PRIMARY_CLUSTER] --targe
   t [SECONDARY_CLUSTER] --list
   ```

   ```
   srm-control groups --source [SECONDARY_CLUSTER] --targe
   t [PRIMARY_CLUSTER] --list
   ```

   b) Verify topics:

   ```
   srm-control topics --source [PRIMARY_CLUSTER] --targe
   t [SECONDARY_CLUSTER] --list
   ```

   ```
   srm-control topics --source [SECONDARY_CLUSTER] --targe
   t [PRIMARY_CLUSTER] --list
   ```

Results

SRM is set up with bidirectional replication and all mission critical topics and consumer groups are whitelisted on both the primary and secondary clusters.

Related Information

Configuration Examples

Migrating Consumer Groups Between Clusters

Learn how to migrate consumers between clusters.

About this task

If a primary Kafka cluster is temporarily unavailable, you can migrate mission-critical workloads to a secondary Kafka cluster (failover). When the primary cluster is restored, you can migrate back (failback). The steps for migrating consumers in a failover or failback scenario are identical. However, depending on the scenario, your source and target clusters will be different. During failover you migrate consumers from primary to secondary, while during failback you migrate consumers from secondary to primary.

Before you begin

- Make sure that the clusters that you are migrating consumers between are set up with bidirectional replication.

- Verify that all mission critical consumer groups and topics, including the ones on the secondary cluster are whitelisted.

Procedure

1. Export the translated consumer group offsets of the source cluster:

```
srm-control offsets --source [SOURCE_CLUSTER] --target [TARGET_CLUSTER] --
group [GROUP1] --export > out.csv
```

2. Reset consumer offsets on the target cluster:

```
kafka-consumer-groups --bootstrap-server [TARGET_BROKER:PORT] --reset-off
sets --group [GROUP1] --execute --from-file out.csv
```

3. Start consumers on the target cluster.

Results

Consumers automatically resume processing messages on the target cluster where they left off on the source cluster.

Related Information

Offsets Subcommand