

Cloudera Stream Processing 2.0.0

## Using SRM

Date published: 2019-09-13

Date modified: 2019-09-13

# CLOUDERA

<https://docs.cloudera.com/>

# 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>SRM Command Line Tools.....</b>	<b>4</b>
srm-control.....	4
Topics and Groups Subcommand.....	4
Offsets Subcommand.....	6
Options reference.....	6
<b>Monitoring Replication with Streams Messaging Manager.....</b>	<b>7</b>
<b>Replicating Data.....</b>	<b>8</b>
<b>How to Set up Failover and Failback.....</b>	<b>8</b>
Configure SRM for Failover and Failback.....	8
Migrating Consumer Groups Between Clusters.....	10

## SRM Command Line Tools

Overview of the command line tools shipped with SRM.



**Important:** SRM in CSP includes the `srm-control`, `srm-driver`, `srm-service`, and `srm.sh` command line tools. Use the `srm-control` tool to manage replication of topics and consumer groups. Do not use the `srm-driver`, `srm-service`, and `srm.sh` tools, instead use Cloudera Manager 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.

The `srm-control` command line tool is located in `/opt/cloudera/parcels/STREAMS_REPLICATION_MANAGER/bin`. By default no alias is provided for the tool. This means that in order to use it, you need to reference its path every time you want to pass a command. For ease of use, Cloudera recommends that you either create an alias, or add the SRM bin directory to your `PATH` environment variable.



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

### Topics and Groups Subcommand

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

The `topics` and `groups` subcommands are used to manipulate the topic or group white and blacklists. Both subcommands support the same set of command options.

Add topics or groups to a whitelist:

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

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

Remove topics or groups from a whitelist:

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

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

Add topics or groups to a blacklist:

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

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

Remove topics or groups from a blacklist:

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

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

Specifying topics or groups is also possible with regular expressions. The following example adds all topics to the whitelist, 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 black or whitelist.

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

### 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 --source [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 --source [SOURCE_CLUSTER] --list
```

## Offsets Subcommand

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

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] --group [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 --source [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 --source [SOURCE_CLUSTER] --group [GROUP] --export > out.csv
```

## Related Information

[How to Set up Failover and Failback](#)

[Migrating Consumer Groups Between Clusters](#)

## Options reference

A collection of all options and their descriptions for the `srm-control` command line tool.

**Table 1: Topics and groups subcommand properties**

Options	Description
-h, --help	Shows the help message.
--source	Specifies the source cluster
--target	Specifies the target cluster
--config	Specifies the SRM configuration file to use.
--add	Specifies topics or groups to add to the allowlist
--remove	Specifies topics or groups to remove from the allowlist
--add-blacklist	Specifies topics or groups to add to the denylist
--remove-blacklist	Specifies topics or groups to remove from the denylist
--list	Lists current allowlist and denylist
--bootstrap-servers	Specifies the bootstraps servers
--producer-props	Specifies producer configuration properties
--consumer-props	Specifies consumer configuration properties
--props	Specifies client configuration properties

**Table 2: Offsets subcommand properties**

Option	Description
-h, --help	Shows the help message.
--source	Specifies the source cluster
--target	Specifies the target cluster
--config	Specifies the SRM configuration file to use.
--export	Export translated offsets
--group	Specifies the groups translated offsets should be exported for
--bootstrap-servers	Specifies the bootstraps servers
--props	Specifies client configuration properties

## 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] --target
t [SECONDARY_CLUSTER] --add [GROUP1],[GROUP2]
```

- a) Whitelist topics:

```
srm-control topics --source [PRIMARY_CLUSTER] --target
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] --target
t [PRIMARY_CLUSTER] --add [GROUP1],[GROUP2]
```

#### b) Whitelist remote topics:

```
srm-control topics --source [SECONDARY_CLUSTER] --target
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] --target
t [SECONDARY_CLUSTER] --list
```

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

#### b) Verify topics:

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

```
srm-control topics --source [SECONDARY_CLUSTER] --target
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-offsets --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](#)