

Securing Streams Replication Manager

Date published: 2019-09-13

Date modified: 2021-08-05



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

| | |
|---|----------|
| Streams Replication Manager security overview..... | 4 |
| Enabling TLS/SSL for the SRM service..... | 5 |
| Enabling Kerberos for the SRM service..... | 6 |
| Configuring custom Kerberos principal for Streams Replication Manager..... | 7 |
| SRM security example..... | 8 |

Streams Replication Manager security overview

Configuring Streams Replication Manager (SRM) security involves enabling and setting security-related features and properties for the SRM service (Driver and Service roles) and the srm-control command line tool. This permits SRM to access source and target clusters and replicate data between them. In addition, it also enables the subcomponents of SRM that act as servers to function in a secure way.

Streams Replications Manager functions both as a client and a server. When SRM replicates data and connects to Kafka clusters it functions as a client. In addition however, some processes and components of SRM act as servers. For example the SRM Service role spins up a REST server. Similarly, the SRM Driver role has replication specific Connect REST servers which provide background functionality.

As a result of this, configuring security for SRM has two distinct aspects as you can configure security for SRM not only when it acts as client, but also when it acts as a server.

Server configuration

Security for SRM processes and components that act as servers can be configured by enabling the TLS/SSL and/or Kerberos feature toggles as well as configuring key and truststore related properties available in Cloudera Manager. For more information see, *Enable TLS/SSL for the SRM service* or *Enable Kerberos authentication for the SRM service*



Note: While the security feature toggles enable security for the components of SRM that function as servers, they are multi-purpose and also configure the security properties that are needed for SRM to connect, as a client, to the co-located clusters. As a result of this, if you have followed *Defining and adding clusters for replication* and have a secured co-located cluster, it is likely that security is already set up for the components that act as servers.

Client configuration

Configuring security for SRM when it functions as a client involves enabling and setting security-related features and properties for the SRM service and the srm-control command line tool. This permits both the service and the tool to access your source and target clusters and replicate data between them. The configuration workflow and the steps you need to take differ for the service and tool.

SRM service

Before you can start replicating data with SRM, you must define the clusters that take part in the replication process and add them to SRM's configuration. When you define a cluster, you also specify the required keys, certificates, and credentials needed to access the clusters that you define. This means that when you define a cluster for replication, at the same time, you also configure the necessary security-related properties needed for SRM to connect as a client.

The clusters and their security properties can be defined in multiple ways. What method you use depends on the clusters in your deployment and the type of security these clusters use. For more information regarding the workflow and the available options for configuration, see *Defining and adding clusters for replication*.

srm-control tool

In addition to configuring security for the SRM service, you also need to configure security related properties for the srm-control tool. This is because the srm-control tool also functions as a client and must have access to the clusters in the deployment. The exact configuration steps you need to take depends on how your deployment is set up, how you configured the service, and the type of security used on the clusters. For more information regarding the workflow and the available options for configuration, see *Configuring srm-control*.

Related Information

[Enabling TLS/SSL for the SRM service](#)

[Defining and adding clusters for replication](#)[Configuring srm-control](#)[Enabling Kerberos for the SRM service](#)

Enabling TLS/SSL for the SRM service

TLS/SSL can be enabled and configured for the Streams Replication Manager (SRM) service (Driver and Service roles) with various configuration properties available in Cloudera Manager. Configuring these properties affects the security configuration of SRM in multiple ways.

About this task

Both the Driver and Service roles of SRM have a number of TLS/SSL related properties associated with them. A dedicated TLS/SSL feature toggle exists for both roles. These are the Enable TLS/SSL for SRM Driver and Enable TLS/SSL for SRM Service properties. In addition to the feature toggles, there are a number of other properties that can be used to configure key and truststore information.

Configuring the feature toggles and the key/truststore related properties have the following effects on SRM's security configuration:

- The SRM Service role's REST server becomes secured and uses HTTPS.
- The SRM Driver role's replication specific Connect REST servers become secured and use HTTPS. In addition, client authentication will also be required from any client connecting to these servers.



Note: The replication specific Connect REST servers are for internal use only. They enable communication between Driver role instances. Third party clients should not interface with them.

- If the deployment has a co-located Kafka cluster and that cluster was configured using a service dependency, both the Service and Driver roles will use the keystore and truststore information when they establish a connection with the co-located Kafka cluster.
- Both the Driver and Service roles will use these properties as fallback configurations when establishing a connection to a Kafka cluster.

That is, if there is a Kafka cluster in your configuration that has its protocol specified as SSL, but no trust or keystore information is set for it, the roles will use the truststore and keystore configured with these properties.



Important: Configuring the feature toggles and key/truststore properties on their own do not enable SRM to connect to or replicate a TLS/SSL enabled external Kafka cluster. They also do not have an effect on the srm-control tool's security configuration.

Configuring these properties is part of the process of defining and adding clusters described in *Defining and adding clusters for replication*. Depending on how you set up your co-located cluster, these properties might already be configured.

If you configured the co-located cluster with a service dependency, then these properties are configured in your deployment and no additional steps are needed to enable or configure TLS/SSL.

However, if you chose to set up the co-located cluster with a Kafka credential, these properties might not be configured. Although in a case like this the properties required by SRM to access the co-located cluster will be set, the server functionality of the roles will not be TLS/SSL enabled. Additionally, while not crucial, no fallback properties will be set up for security either. In a case like this Cloudera recommends that you complete the following steps.

Procedure

1. In Cloudera Manager, go to Clusters and select the Streams Replication Manager service.
2. Go to Configuration.

- Find and configure the following properties based on your cluster and requirements:

Table 1:

| Cloudera Manager Property | Description |
|---|---|
| Enable TLS/SSL for SRM Driver | Encrypt communication between clients and SRM Driver using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)). |
| SRM Driver TLS/SSL Server JKS Keystore File Location | The path to the TLS/SSL keystore file containing the server certificate and private key used for TLS/SSL. Used when SRM Driver is acting as a TLS/SSL server. The keystore must be in JKS format. |
| SRM Driver TLS/SSL Server JKS Keystore File Password | The password for the SRM Driver JKS keystore file. |
| SRM Driver TLS/SSL Server JKS Keystore Key Password | The password that protects the private key contained in the JKS keystore used when SRM Driver is acting as a TLS/SSL server. |
| SRM Driver TLS/SSL Trust Store File | The location on disk of the trust store, in .jks format, used to confirm the authenticity of TLS/SSL servers that SRM Driver might connect to. This trust store must contain the certificate(s) used to sign the service(s) connected to. If this parameter is not provided, the default list of well-known certificate authorities is used instead. |
| SRM Driver TLS/SSL Trust Store Password | The password for the SRM Driver TLS/SSL Trust Store File. This password is not required to access the trust store; this field can be left blank. This password provides optional integrity checking of the file. The contents of trust stores are certificates, and certificates are public information. |
| Enable TLS/SSL for SRM Service | Encrypt communication between clients and SRM Service using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)). |
| SRM Service TLS/SSL Server JKS Keystore File Location | The path to the TLS/SSL keystore file containing the server certificate and private key used for TLS/SSL. Used when SRM Service is acting as a TLS/SSL server. The keystore must be in JKS format. |
| SRM Service TLS/SSL Server JKS Keystore File Password | The password for the SRM Service JKS keystore file. |
| SRM Service TLS/SSL Server JKS Keystore Key Password | The password that protects the private key contained in the JKS keystore used when SRM Service is acting as a TLS/SSL server. |
| SRM Service TLS/SSL Trust Store File | The location on disk of the trust store, in .jks format, used to confirm the authenticity of TLS/SSL servers that SRM Service might connect to. This trust store must contain the certificate(s) used to sign the service(s) connected to. If this parameter is not provided, the default list of well-known certificate authorities is used instead. |
| SRM Service TLS/SSL Trust Store Password | The password for the SRM Service TLS/SSL Trust Store File. This password is not required to access the trust store; this field can be left blank. This password provides optional integrity checking of the file. The contents of trust stores are certificates, and certificates are public information. |

- Click Save Changes.
- Restart the SRM service.

Related Information

[Defining and adding clusters for replication](#)

Enabling Kerberos for the SRM service

Kerberos can be enabled and configured for the Streams Replication Manager (SRM) service with the Enable Kerberos Authentication Cloudera Manager property.

About this task

Kerberos authentication can be enabled for the SRM service (Driver and Service roles) with the Enable Kerberos Authentication property. Enabling this property generates a JAAS configuration that is used by default for all SASL connections made by the SRM service. That is, SRM will connect to the co-located Kafka cluster using this JAAS configuration. Additionally, SRM will fall back to using this JAAS configuration for all other SASL connections where a connection (cluster) specific configuration is not configured.



Important: Configuring the Enable Kerberos Authentication on its own does not enable SRM to connect to or replicate a Kerberos enabled external Kafka cluster. Additionally, configuring the property does not have an effect on the srm-control tool's security configuration.

Configuring this property is part of the process of defining and adding clusters described in *Defining and adding clusters for replication*. Depending on how you set up your co-located cluster, the property might already be enabled.

Procedure

1. In Cloudera Manager, go to Clusters and select the Streams Replication Manager service.
2. Go to Configuration.
3. Find and enable the Enable Kerberos Authentication property
4. Click Save Changes.
5. Restart the SRM service.

Related Information

[Defining and adding clusters for replication](#)

Configuring custom Kerberos principal for Streams Replication Manager

In a Kerberos-enabled cluster, the Streams Replication Manager (SRM) service uses the streamsrepmgr principal by default. The principal can be configured in Cloudera Manager with the Kerberos Principal property.

About this task



Important: Cloudera Manager configures CDP services to use the default Kerberos principal names. Cloudera recommends that you do not change the default Kerberos principal names. If it is unavoidable to do so, contact Cloudera Professional Services because it requires extensive additional custom configuration.

The principal specified in the Kerberos Principal property is used by the SRM service (Driver and Service roles) when it establishes a connection with its co-located Kafka cluster. Additionally, this principal is also used for Kerberos-enabled clusters defined with a Kafka credential, but only if there is no specific JAAS configuration set for that cluster.

For example, if you have a Kerberos-enabled external Kafka cluster that you defined with a Kafka credential, but did not specify a JAAS configuration, SRM falls back to using the default JAAS configuration which contains the principal defined in the Kerberos Principal property.



Important: The principal set in the Kerberos Principal property is only used by the SRM service. Configuring the property does not affect the principal used by the srm-control tool. For more information on how to configure the srm-control tool, see *Configuring srm-control*.

Procedure

1. In Cloudera Manager select the SRM service.
2. Go to Configuration.
3. Find the Kerberos Principal property and enter the custom Kerberos principal.

4. Click Save Changes.
5. Restart the SRM service.

Results

The custom principal used by SRM is configured.

What to do next

If you use Ranger for authorization, update all resource-based services and policies that use the old principal and add the new principal. For more information on updating resource-based services and policies, see *Using Ranger to Provide Authorization in CDP*.

Related Information

[Using Ranger to Provide Authorization in CDP](#)

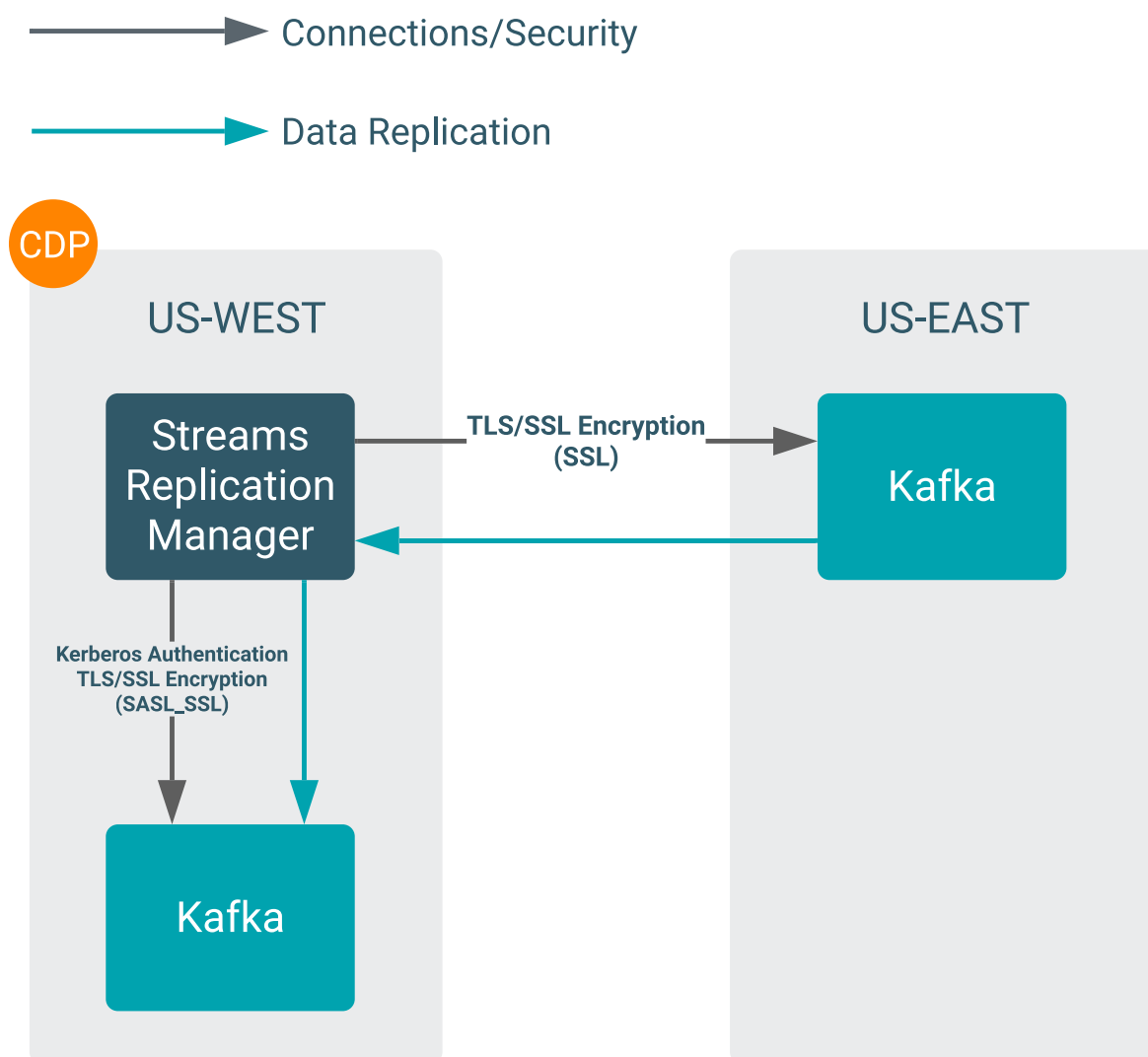
[Configuring srm-control](#)

SRM security example

Streams Replication Manager (SRM) is capable of replicating data between CDP and non-CDP clusters that are secured. A setup like this involves configuring security for the SRM service (Driver and Service roles) and the srm-control command line tool.

About this task

Consider the following replication environment.



There are two clusters, US-West and US-East. US-West is a CDP cluster that has both Kafka and SRM deployed on it. The Kafka service on this cluster is both Kerberos and TLS/SSL enabled. That is, clients connecting to this cluster (including SRM) use the SASL_SSL protocol.

US-East has Kafka deployed on it but not SRM. Kafka on this cluster has TLS/SSL encryption enabled. More importantly, the platform that this cluster is running on is not defined. This is because this example is meant to demonstrate that SRM can connect to and replicate data to or from non-CDP clusters. As long as that cluster is running Kafka, SRM will be able to connect to it.

Data replication is unidirectional. The SRM service deployed in US-West is replicating Kafka data from US-East to US-West. From the perspective of the SRM service, US-West is its co-located Kafka cluster, while US-East is an external Kafka cluster.

The following example walks you through the steps required to set up both the SRM service and the srm-control tool for this replication environment.

Before you begin

The following steps assume that the SRM service is already installed and available on US-West. Additionally, it is also assumed that key and truststore files as well as other credentials required to establish a connection with each cluster are known and are available. The instructions do not go into detail on how you can generate or acquire these credentials.

Procedure

1. Define the external Kafka cluster (US-East):

- a) Log into the Cloudera Manager instance managing US-West.
- b) Go to AdministrationExternal Accounts.
- c) Go to the Kafka Credentials tab.
- d) Click Add Kafka credentials.
- e) Create a Kafka credential for US-East:

The security configuration of the external cluster determines which of the available properties you must set. Because in this example US-East has TLS/SSL encryption enabled, the following properties must be set:

- Name
- Bootstrap servers
- Security protocol
- Truststore Password
- Truststore Path
- Truststore Type



Note: Click the i icon next to each property in Cloudera Manager to reveal additional information about each property.

- f) Click Add.

If credential creation is successful, a new entry corresponding to the Kafka credential you specified appears on the page.

2. Define the co-located Kafka cluster (US-West):

Co-located Kafka clusters can be defined in two separate ways. Either with a service dependency or with Kafka credentials. Because in this example US-West is Kerberos and TLS/SSL enabled, the service dependency method is used. This is in-line with Cloudera best practices.

- a) In Cloudera Manager, go to Clusters and select the Streams Replication Manager service.
- b) Go to Configuration.
- c) Find and enable the Kafka Service property.
- d) Find and configure the Streams Replication Manager Co-located Kafka Cluster Alias property.

The alias you configure represents the co-located cluster. Enter an alias that is unique and easily identifiable. For example:

```
uswest
```

- e) Enable relevant security feature toggles

Because US-West is both TLS/SSL and Kerberos enabled, you must enable all feature toggles for both the Driver and Service roles. The feature toggles are the following:

- Enable TLS/SSL for SRM Driver
- Enable TLS/SSL for SRM Service
- Enable Kerberos Authentication

3. Add both clusters to SRM's configuration:

- a) Find and configure the External Kafka Accounts property.

Add the name of the Kafka credential you created for US-East to this property. This can be done by clicking the add button to add a new line to the property and then entering the name of the Kafka credential. For example:

```
useast
```

- b) Find and configure the Streams Replication Manager Cluster alias property.

Add all cluster aliases to this property. This includes the aliases present in both the External Kafka Accounts and Streams Replication Manager Co-located Kafka Cluster Alias properties. Delimit the aliases with commas. For example:

```
useast , uswest
```

4. Configure replications:

In this example, data is replicated unidirectionally from US-West to US-East. As a result, only a single replication must be configured.

- 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:

```
useast->uswest.enabled=true
```

5. Configure Driver and Service role targets:

- a) Find and configure the Streams Replication Manager Service Target Cluster property.

Add the co-located cluster's alias to the property. In the case of this example:

```
useast
```

- b) Find and configure the Streams Replication Manager Driver Target Cluster property.

This property should contain all cluster aliases. In the case of this example, you can either add the aliases of both US-West and US-East or leave the property empty. Leaving the property empty is the same as adding all aliases.

6. Click Save Changes.**7. Restart the SRM service.**

8. Configure and run the srm-control tool:

- a) Click Gateway in the Filters pane.
- b) Find and configure the following properties:

SRM Client's Secure Storage Type

The keystore type of the secure storage. Must be a valid Java keystore type. Cloudera recommends that you use the default, PKCS12.

SRM Client's Secure Storage Password

The password used to access the secure storage. Take note of the password you configure. You need to provide it in your CLI session before running the tool.

Environment Variable Holding SRM Client's Secure Storage Password

The name of the environment variable that stores the secure storage password. Take note of the name that you configure. You need to set it in your CLI session before running the tool.

Gateway TLS/SSL Trust Store File

The path to the TLS/SSL truststore file containing the server (co-located cluster's) certificate and public key. Ensure that this file is available on all SRM hosts.

Gateway TLS/SSL Truststore Password

The password used to access the truststore file specified in the Gateway TLS/SSL Trust Store File property.

SRM Client's Kerberos Principal Name

The kerberos principal that the tool uses for authentication when connecting to the co-located Kafka cluster.

SRM Client's Kerberos Keytab Location

The path to the Kerberos keytab file that the tool uses for authentication when connecting to the co-located Kafka cluster.

- c) Click Save Changes.
- d) Re-deploy client configuration.
- e) SSH into one of the SRM hosts in your cluster.
- f) Set the secure storage password as an environment variable.

```
export [***SECURE STORAGE ENV VAR***]="[***SECURE STORAGE PASSWORD***]"
```

Replace [***SECURE STORAGE ENV VAR***] with the name of the environment variable you specified in Environment Variable Holding SRM Client's Secure Storage Password. Replace [***SRM SECURE STORAGE PASSWORD***] with the password you specified in SRM Client's Secure Storage Password. For example:

```
export SECURESTOREPASS="mypassword"
```

- g) Run the tool.

For example:

```
srm-control topics --source useast --target uswest --add topic1,topic2
```