Streams Replication Manager for HDF and HDP 1.0.0

# SRM Overview

**Date published: 2019-09-20**
**Date modified: 2019-09-20**

## CLOUDERA

# Legal Notice

# Contents

# Overview

Get familiar with Streams Replication Manager and its components.

Streams Replication Manager (SRM) is an enterprise-grade replication solution that enables fault tolerant, scalable and robust cross-cluster Kafka topic replication. SRM provides the ability to dynamically change configurations and keeps the topic properties in sync across clusters at high performance. SRM also delivers custom extensions that facilitate installation, management and monitoring making SRM a complete replication solution that is built for mission critical workloads. Streams Replication Manager consists of two main components. The Stream Replication Engine and the Stream Replication Management Services.

**Figure 1: Streams Replication Manager Overview**
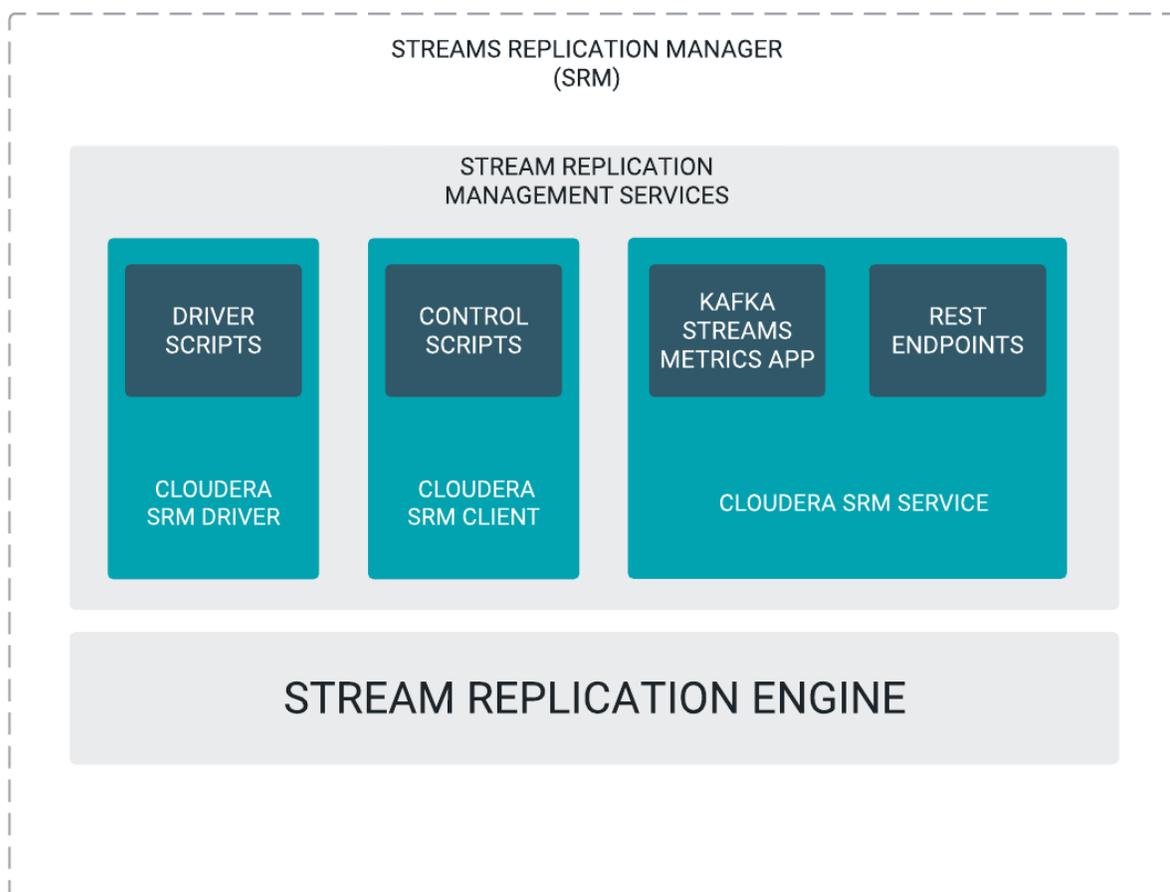


**Stream Replication Engine**

> The Stream Replication Engine is a next generation multi-cluster and cross-datacenter replication engine for Kafka clusters.

**Stream Replication Management Services**

> Stream Replication Management Services are services powered by open source Cloudera extensions which utilize the capabilities of the Stream Replication Engine. These services provide:

> - Easy installation
> - Lifecycle management
> - Management and monitoring of replication flows across clusters

The Stream Replication Management Services includes the following custom extensions:

**Cloudera SRM Driver**

> The Cloudera SRM Driver is a small wrapper around the Stream Replication Engine that adds the extensions provided by Cloudera. It provides the ability to spin up SRM clusters and has a metrics reporter. The command line tool associated with the Cloudera SRM Driver is called `srm-driver`.

**Cloudera SRM Client**

> The Cloudera SRM Client provides users with command line tools that enable replication management for topics and consumer groups. The command line tool associated with the Cloudera SRM Client is called `srm-control`.

**Cloudera SRM Service**

> The Cloudera SRM Service consists of a REST API and a Kafka Streams application to aggregate and expose cluster, topic and consumer group metrics. The command line tool associated with the Cloudera SRM Service is called `srm-service`.

# Key Features

SRM has the following main features.

### Remote topics

Remote topics are replicated topics referencing a source cluster via a naming convention. For example, the "topic1" topic from the "us-west" source cluster creates the "us-west.topic1" remote topic on the target cluster. SRM automatically applies this configurable "replication policy" across your organization, enabling tooling to distinguish remote topics from source topics. For more information regarding remote topics, see Understanding Replication Flows.

### Consistent semantics

Partitioning and record offsets are synchronized between replicated clusters to ensure consumers can migrate from one cluster to another without losing data or skipping records.

### Cross cluster configuration

Topic-level configuration properties are synced across clusters. For example, the cleanup policy (`cleanup.poli cy`), or the log segment file size (`segment.bytes`), as well as other topic-level configurations are automatically synched to remote topics. This simplifies managing topics across multiple Kafka clusters.

### Consumer group checkpoints

In addition to data and configuration, SRM replicates consumer group progress via periodic checkpoints. At configurable intervals, checkpoint records are emitted to downstream clusters, encoding the latest offsets for whitelisted consumer groups and topic-partitions. As with topics, groups are matched against an allowlist which can be updated dynamically with `srm-control`. Normally, consumer group offsets are not portable between Kafka clusters, as offsets are not consistent between otherwise identical topic-partitions on different clusters. SRM's checkpoint records account for this by including offsets which are automatically translated from one cluster to another. This offset translation feature works in both directions; a consumer group can be migrated from one cluster to another (failover) and then back again (failback) without skipping records or losing progress.

### Automatic topic and partition detection

SRM monitors Kafka clusters for new topics, partitions, and consumer groups as they are created. These are compared with configurable whitelists, which may include regular expressions.

### Tooling to automate consumer migration

SRM tooling enables operators to translate offsets between clusters and to migrate consumer groups while preserving state.

### Centralized configuration for multi-cluster environments

SRM leverages a single top-level configuration file to enable replication across multiple Kafka clusters. Moreover, command-line tooling can alter which topics and consumer groups are replicated in real-time.

### Replication monitoring

Since cluster replication will mainly be used for highly critical Kafka applications, it is crucial for customers to be able to easily and reliably monitor the Kafka cluster replications. The custom extensions included with SRM collect and aggregate Kafka replication metrics and make them available through a REST API. This REST API is used by Streams Messaging Manager (SMM) to display metrics. Customers could also use the REST API to implement their own monitoring solution or plug it into third party solutions. The metrics make the state of cluster replication visible to end users who then can take corrective action if needed.

### Related Information

Monitoring Cluster Replications Overview
Understanding Replication Flows