

Streams Replication Manager for HDF and HDP 1.0.0

# SRM Main Use Cases and Use Case Architectures

Date published: 2019-09-20

Date modified: 2019-09-20

The Cloudera logo is displayed in a bold, orange, sans-serif font. The word "CLOUDERA" is written in all caps, with a stylized 'E' that has a horizontal bar extending to the right.

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

# Legal Notice

© Cloudera Inc. 2022. 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>Main Use Cases.....</b>	<b>4</b>
<b>Use Case Architectures.....</b>	<b>4</b>
Highly Available Kafka Architectures.....	4
Active / Stand-by Architecture.....	4
Active / Active Architecture.....	5
Cross Data Center Replication.....	6
Cluster Migration Architectures.....	7
On-premise to Cloud and Kafka Version Upgrade.....	7
Aggregation for Analytics.....	8

## Main Use Cases

Learn about the main use cases of SRM.

Apache Kafka has become an essential component of enterprise data pipelines and is used for tracking clickstream event data, collecting logs, gathering metrics, and being the enterprise data bus in a microservices based architectures. Kafka supports internal replication to support data availability within a cluster. However with Kafka based applications becoming critical, enterprises require that the data availability and durability guarantees span entire cluster and site failures.

Replication of data across clusters and sites is key for the following use cases:

### **Disaster Recovery**

Common enterprise use cases for cross-cluster replication is for guaranteeing business continuity in the presence of cluster or data center-wide outages.

### **Aggregation for Analytics**

Aggregate data from multiple streaming pipelines possibly across multiple data centers to run batch analytics jobs that provide a holistic view across the enterprise.

### **Data Deployment after Analytics**

This is the opposite of the aggregation use case in which the data generated by the analytics application in one cluster (say the aggregate cluster) is broadcast to multiple clusters possibly across data centers for end user consumption.

### **Isolation**

Due to performance or security reasons, data needs to be replicated between different environments to isolate access. In many deployments the ingestion cluster is isolated from the consumption clusters.

### **Geo Proximity**

In geographically distributed access patterns where low latency is required, replication is used to move data closer to the access location.

### **Cloud Migration**

As more enterprises have an on-premise and cloud presence, Kafka replication can be used to migrate data to the public or private cloud and back.

### **Legal and Compliance**

Much like the isolation uses case, a policy driven replication is used to limit what data is accessible in a cluster to meet legal and compliance requirements.

## Use Case Architectures

Highly available and cluster migration architecture examples for SRM.

### Highly Available Kafka Architectures

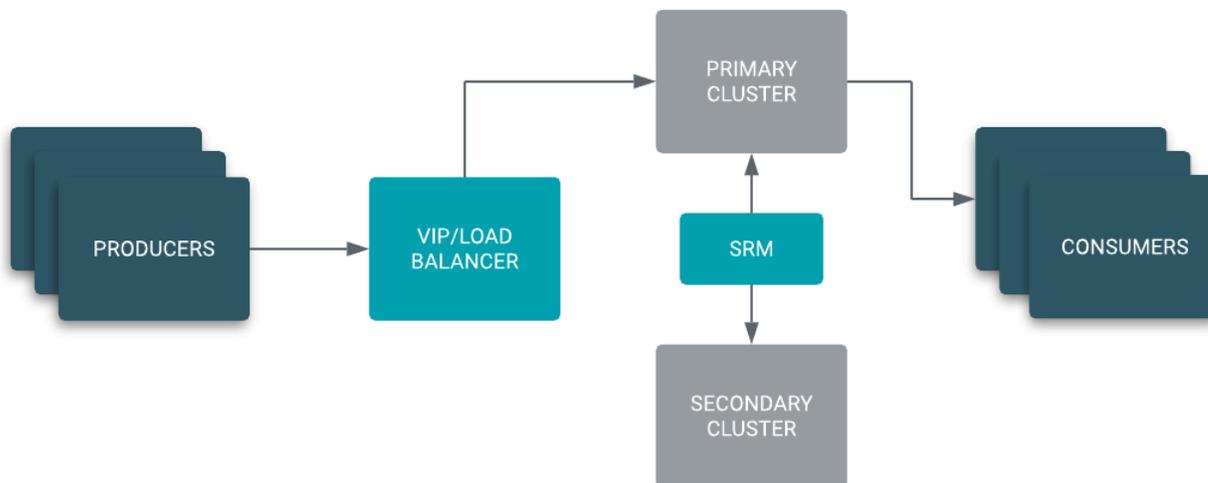
A highly available Kafka deployment must be able to survive a full single cluster outage while continuing to process events without data loss. With SRM, you can implement highly available Apache Kafka deployments which either follow an Active / Stand-by or an Active / Active model.

#### Active / Stand-by Architecture

Active / Stand-by architecture example for SRM.

In an Active / Stand-by scenario, you set up two Kafka clusters and configure SRM to replicate topics bi-directionally between both clusters. A VIP or load balancer directs your producers to ingest messages into the active cluster from which consumer groups are reading from.

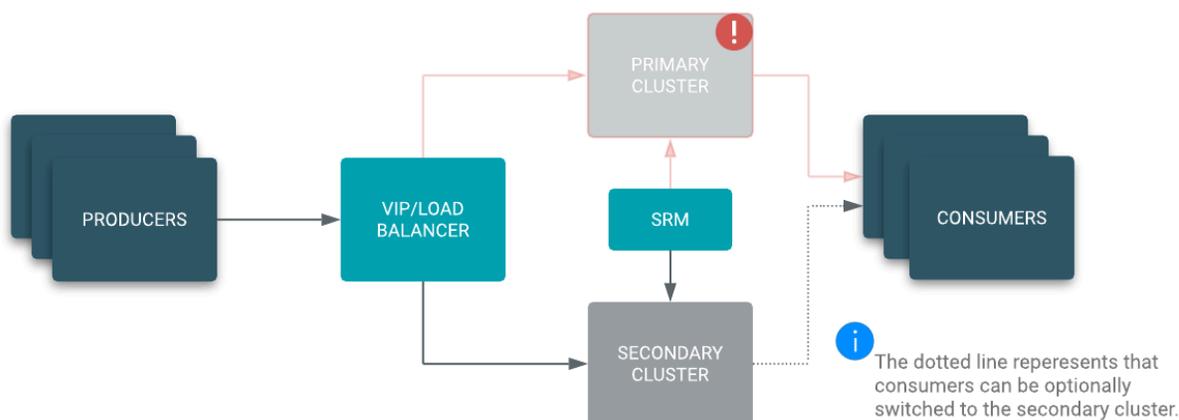
**Figure 1: Active / Stand-by Architecture Standard Operation**



In case of a disaster, the VIP or load balancer directs the producers to the stand-by cluster. You can easily migrate your consumer groups to start reading from the stand-by cluster or simply wait until the primary cluster is restored if the resulting consumer lag is acceptable for your use case.

While the primary cluster is down, your producers are still able to ingest. Once the primary cluster is restored, SRM automatically takes care of synchronizing both clusters making failback seamless.

**Figure 2: Active / Stand-by Architecture Cluster Failure**



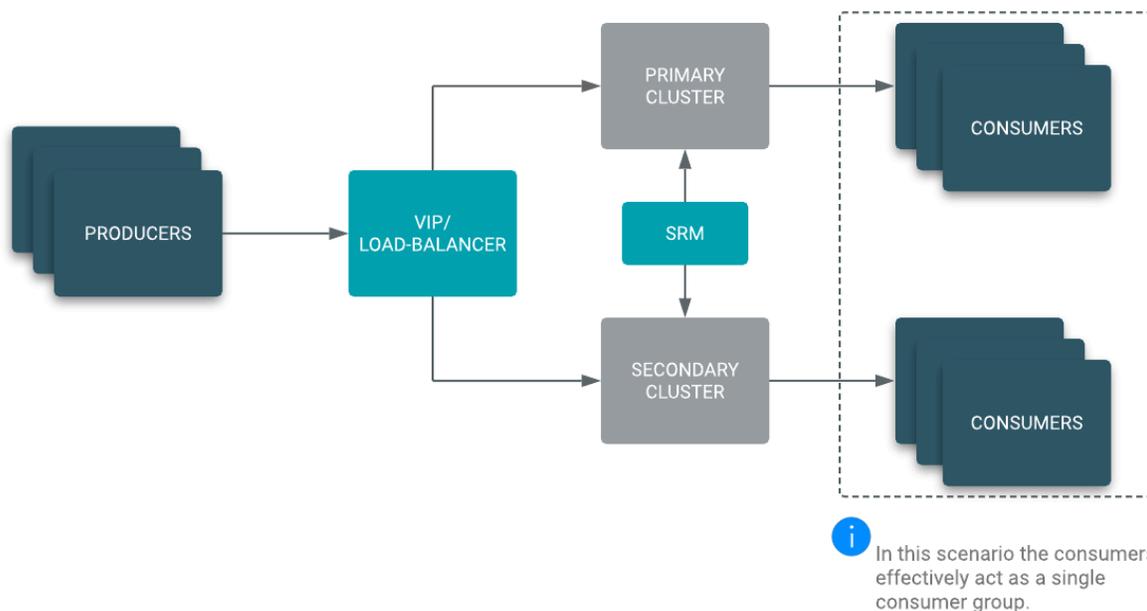
Implementing an Active/Stand-by architecture is the logical choice when an existing disaster recovery site with established policies is already available, and your goals include not losing ingest capabilities during a disaster and having a backup in your disaster recovery site.

### Active / Active Architecture

Active / Active architecture example for SRM.

In an Active / Active scenario, your producers can be load balanced to either your primary or secondary cluster. SRM is configured to replicate topics bi-directionally between both clusters. What makes this architecture Active / Active, is the fact that you now have consumers reading from both clusters at the same time, essentially acting like a cross-cluster consumer group. In case of a disaster the VIP or load balancer directs the producers to the secondary cluster and the secondary cluster consumer group is still able to process messages. While the primary cluster is down, your producers are still able to ingest and your consumers are still able to process messages. This results in a 0 downtime and hands-off failover in case of a disaster. Once the primary cluster is back online, SRM automatically takes care of synchronizing both clusters and your primary consumer group resumes processing messages.

**Figure 3: Active / Active Architecture**



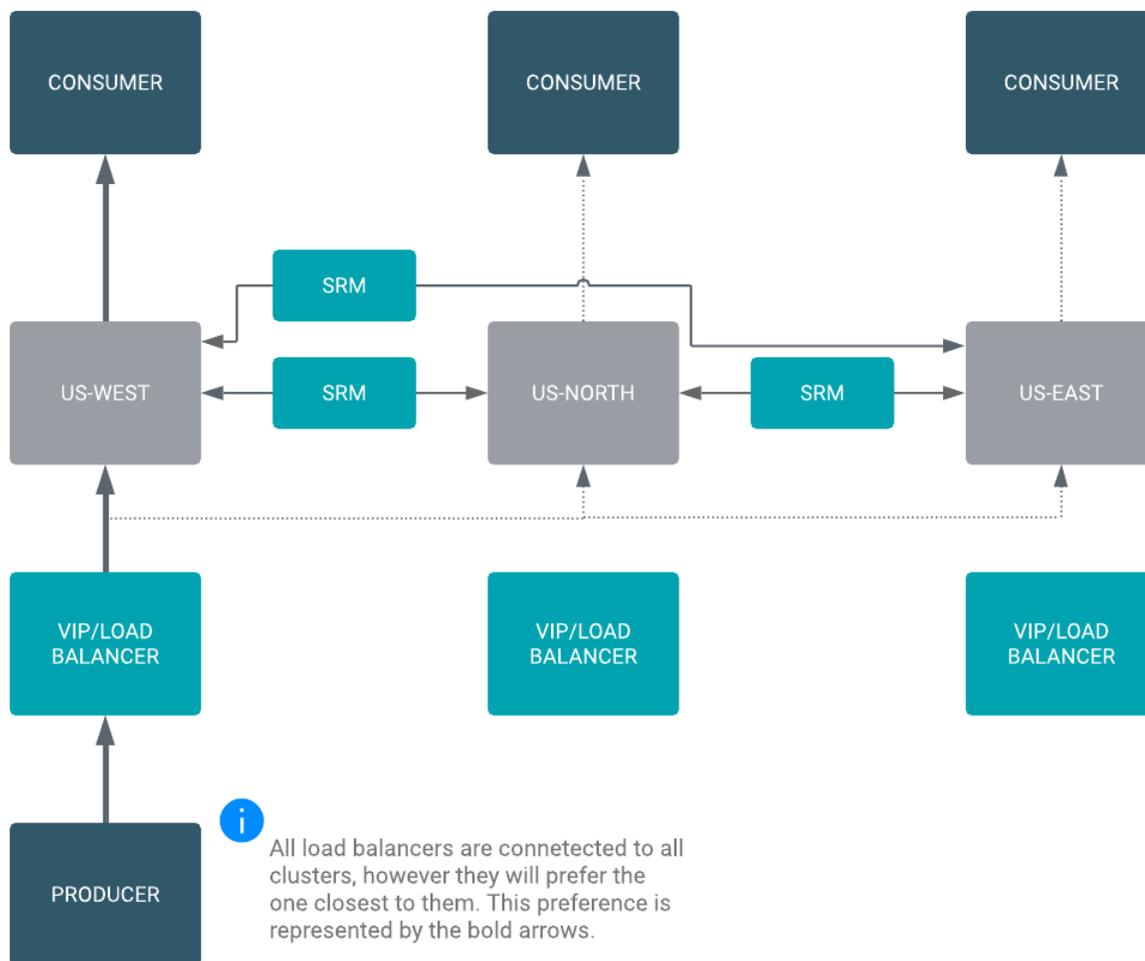
## Cross Data Center Replication

Cross data center replication architecture example for SRM.

Certain applications not only require local high availability within one data center or one availability zone, but have to be highly available across data centers too. You can use SRM to set up replication between Kafka clusters in different data centers which results in messages being available to consumers in each of your data centers.

A load balancer directs your producers to the local data center or closest data center if the primary data center is down. SRM is configured to replicate topics between all data centers. If you are using more than two data centers, SRM is configured to create a “replication circle”, ensuring a single data center failure (for example us-north in the example below) does not halt replication between the remaining clusters.

**Figure 4: Cross Data Center Replication Architecture**



## Cluster Migration Architectures

Example cluster migration architectures.

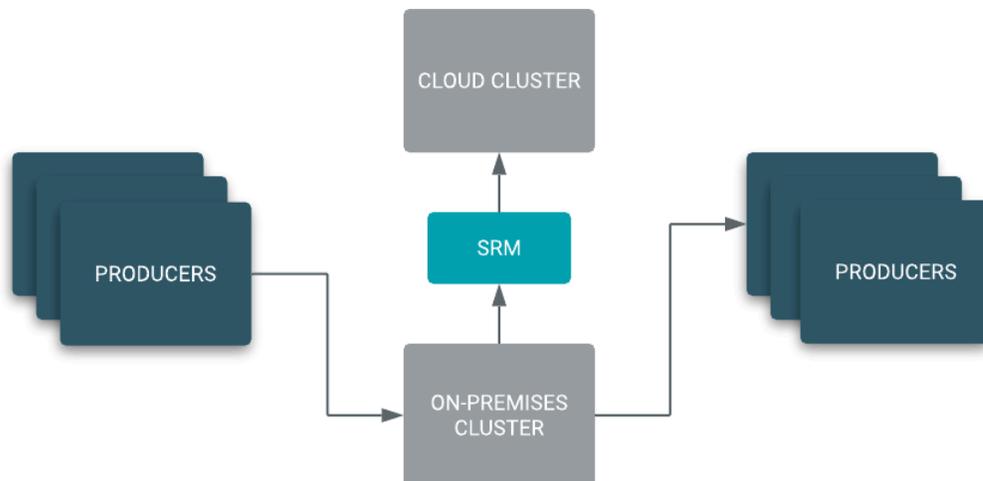
### On-premise to Cloud and Kafka Version Upgrade

On-premise to cloud and Kafka version upgrade example architectures for SRM.

If you have an on-premises Apache Kafka cluster that you want to migrate to the cloud, not only do you have to migrate consumers and producers, you also have to migrate topics and their messages to the new cloud based cluster.

After you have set up replication through SRM, you only need to point your consumers to the new brokers before you can start processing messages from the cloud cluster. This approach ensures that the historical data kept in the on-premises Kafka cluster is migrated to the cloud cluster allowing you to replay messages directly from the cloud without having to go back to your on-premises cluster.

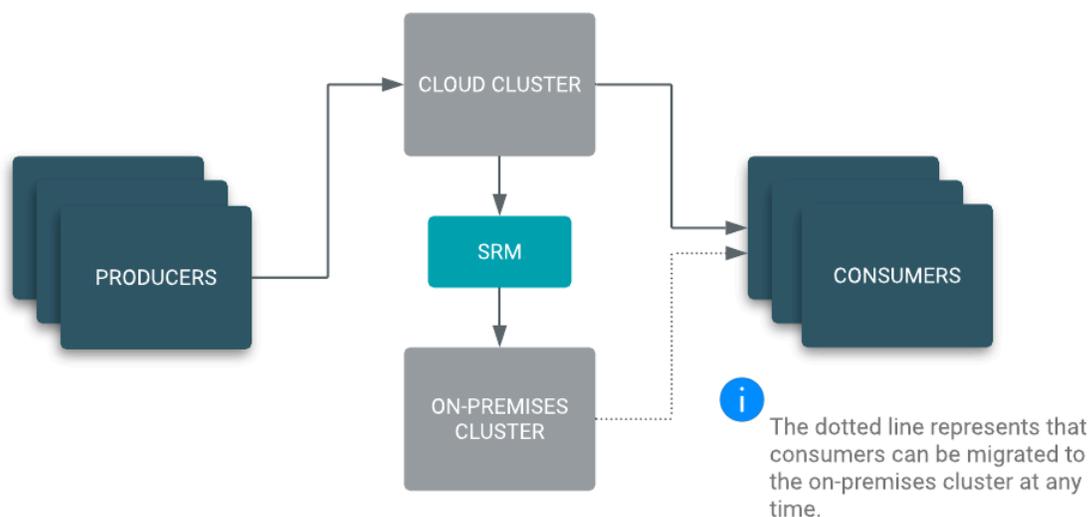
**Figure 5: Cluster Migration On-premise**



Producers and Consumers are using the on-premises cluster while SRM is replicating messages.

Once you have migrated your cluster, producers, and consumers to the cloud, you can use SRM to turn-around the replication direction and use the on-premises cluster as your DR cluster.

**Figure 6: Cluster Migration Cloud**



Producers and Consumers have been migrated to the cloud cluster and the on-premises cluster is used for disaster recovery.

### Kafka Version Upgrade

If you have to upgrade your Kafka cluster to a newer version and an in-place upgrade is not possible, you can use the same migration approach to provision a new cluster, use SRM to replicate all existing topics and messages before migrating your producers and consumers to interact with the new cluster.

### Aggregation for Analytics

Aggregation for analytics architecture example for SRM.

SRM can be used to aggregate data from multiple streaming pipelines, possibly across multiple data centers, to run batch analytics jobs that provide a holistic view across the enterprise.

**Figure 7: Aggregation for Analytics**

