

MANAGE, MONITOR AND REPLICATE APACHE KAFKA ACROSS THE ENTERPRISE AND CLOUD

With Cloudera Data Platform



DATA IN MOTION

Advanced Kafka Management, Monitoring, and Replication for All

Apache Kafka has emerged as the single central backbone of streaming architectures for large organizations, igniting data-in-motion innovations across financial services, telecommunications, manufacturing, and numerous other industries. Kafka is inherently scalable and fits nicely into a myriad of next generation use cases that require advanced analytics on data streams.

The problem that presents itself however, is that Kafka does not inherently have a user interface and so, IT teams across the enterprise struggle to understand what is going on within their Kafka clusters. The few open source and third-party tools in the market are either proprietary, too complex to use, or just plain lacking in features.

The dearth of good Kafka management and monitoring tools makes it virtually impossible to understand how data moves across clusters between producers, brokers, topics, and consumers, let alone provide automated failover and failback capabilities that underpin resilient disaster recovery and business continuity plans.

This paper introduces you to key management components of Cloudera DataFlow (CDF)'s Streams Messaging capabilities, which are now also available within the Cloudera Data Platform (CDP). This materially advances how Kafka management, monitoring, and replication is done through two fully integrated components:

1. **Streams Messaging Manager:** An advanced management and monitoring toolset with a single pane of glass dashboard that is revolutionizing how enterprises gain end-to-end visibility into their Kafka clusters.
2. **Streams Replication Manager:** An enterprise-grade cross cluster Kafka replication solution that realizes business critical use cases that go beyond disaster recovery.

The Streams Messaging capabilities within CDP are available on-premises, on the public cloud as well as on hybrid cloud architectures. These address the unique needs of Kafka platform operations, DevOps, application development, security, and governance teams.

Table of Contents

Advanced Kafka Management, Monitoring, and Replication for All	2
Kafka Blindness	4
Disparate Tools Hinder Teamwork	4
Data-in-Motion in Context	5
Cloudera Streams Messaging Manager (SMM)	6
Simplify Troubleshooting Kafka Environments	6
Visualize End-To-End Kafka Stream Flows	6
Extend Monitoring/Management Capabilities with REST	7
Track Data Lineage and Governance from Edge-To-Enterprise	7
Integration with Schema Registry	7
Topic Lifecycle Management	8
Alerting	8
DevOps Monitoring Capability with End-to-End Latency View	9
Adopt Hybrid Cloud Deployments for Kafka	9
Address Pain Points Across Your Team	10
Cloudera Streams Replication Manager (SRM)	11
MirrorMaker2	11
Remote Topics	12
Replication Flows	12
Replication Monitoring	12
HeartBeats	12
Checkpoints	12
Offset Syncs	13
Automatic Topic and Partition Detection	13
Advanced Disaster Recovery and High Availability	13
Other Business Critical Use Cases	13
Next Steps	14

Kafka Blindness

Enterprises are transforming their businesses to keep up with customer demand, provide better services, and proactively manage risk with a wide range of streaming data initiatives. Apache Kafka has emerged as a key component to many of those architectures.

Organizations and even groups within the same organization have different streaming data approaches but Kafka is constant—it's everywhere and an important component in many of our customers' architectures.

However, most organizations struggle to monitor and manage Kafka in production and often do so through a hodgepodge of disparate tools and methodologies. Based on the feedback and experience of our top Cloudera Kafka customers, it became clear that users need better visibility into how data moves across clusters between Kafka producers, brokers, and consumers. Visibility, or lack thereof, was a recurring theme to which customers coined the term, "Kafka Blindness".

Enterprises struggle to monitor, troubleshoot and see what's happening inside their Kafka clusters. They are also challenged to replicate those clusters in a way that comprehensively suffice stringent business continuity requirements.

Disparate Tools Hinder Teamwork

An indication of how well Kafka has been adopted is by the number of dedicated teams that have been created to manage and support Kafka clusters across their respective organizations. Each team needs to fulfill distinct responsibilities, but Kafka Blindness is something they all have in common.

- **Kafka platform operations teams** lack visibility at a cluster and broker level, the effects of the broker on the infrastructure, and vice versa.
- **Kafka DevOps and application teams** need to know how data flows between producers, brokers, topics, and consumers and to understand the key performance metrics (KPMs) of those entities.
- **Governance and security teams** require transparency around chain of custody, audit, metadata, access control, and data lineage.

Based on our customer interviews, there was an overwhelming consensus that the combination of native Kafka command line utilities, open source tools, and homegrown and proprietary solutions are ineffective in providing the complete picture. What's required is a centralized, single pane of glass dashboard that lets them see how data moves across Kafka entities and identify misconfigured aspects or failures. Additionally, they need first class support for REST services in order to integrate with third-party enterprise monitoring tools.

[Cloudera Data Platform's \(CDP\) Streams Messaging](#) capabilities address each Kafka team's unique visibility requirements through one comprehensive toolset. Streams Messaging Manager provides a single source of truth that each team can collaborate through. It also opens a window to replication functions through Streams Replication Manager, extending enterprise-wide Kafka visibility to the business continuity, disaster recovery, and shared services teams.

Data-in-Motion in Context

Before diving into the capabilities of managing, monitoring, and replicating Kafka clusters across the enterprise, it helps to understand how Streams Messaging Manager and Streams Replication Manager fit within the larger context of data-in-motion.

Cloudera DataFlow (CDF) is the data-in-motion platform that supports the entire streaming data journey from data capture and flow management at the edge (1) to provisioning that data directly to/from your Kafka messaging backbone (2) and/or stream processing and analytics (3). The diagram below illustrates how the data-in-motion platform is tightly integrated with Cloudera Data Platform (CDP)'s [Shared Data Experience \(SDX\)](#)—a common set of services that offer unified security and governance (4) across your enterprise's data center and cloud environments (5).



The Streams Messaging capabilities, in particular, are about Kafka and the entire ecosystem of components that support and enhance Kafka. The management and monitoring toolset within that provide a comprehensive way to support your Kafka clusters with Streams Messaging Manager and Streams Replication Manager.

The rest of this paper will walk you through the key capabilities and innovations of these two components and how they address Kafka Blindness and enhance business continuity.



Questions Answered by Streams Messaging Manager for the Platform Operations Teams

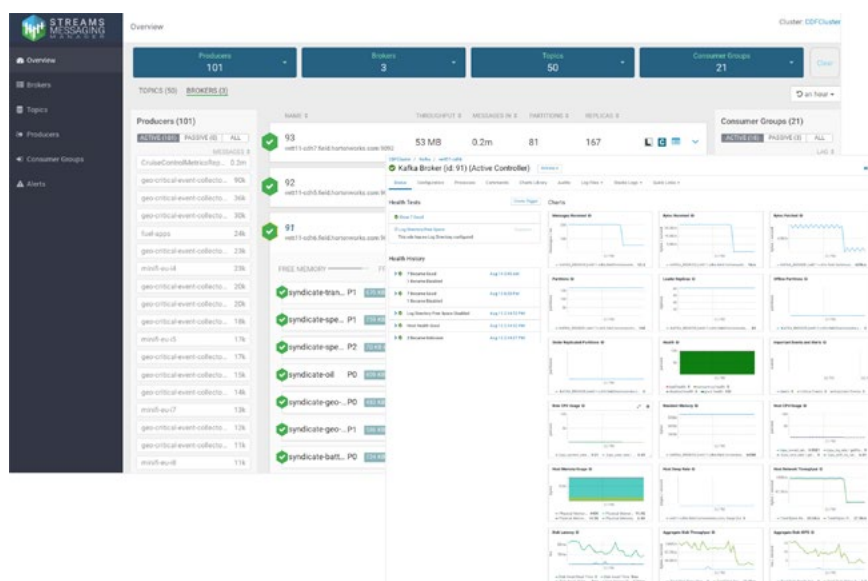
- Do I have any offline topic partitions?
- Which consumer group is falling behind the most?
- How many active producers/consumers are there?
- Which producers are generating the most data right now?
- Are any of my brokers down?
- What is the throughput in/out for a given partition on that broker?
- Are there any skewed partitions for a broker?
- How many active producers and consumers exist now?
- Are any of my brokers running out of disk space?
- How much capacity do I have available per broker in the cluster?
- Which partitions are located on each broker?
- What hosts are my brokers located on?
- Are all my replicas in my topic in-synch?
- Are any of my brokers running hot?
- Which broker has the highest throughput in/out rates?
- How many total topics does my Kafka cluster have?
- Which of my topics has produced or consumed the most messages over the last N minutes/hours?
- What is the end-to-end latency of my Kafka data pipeline?

Cloudera Streams Messaging Manager (SMM)

Cloudera Streams Messaging Manager (SMM) is an operations monitoring/management tool that provides end-to-end visibility into how data moves across Kafka clusters between producers, brokers, topics, and consumers. The following sections highlight just a few of the many innovative capabilities it provides. Also, see the sidebars to get a sense of how SMM helps each Kafka team.

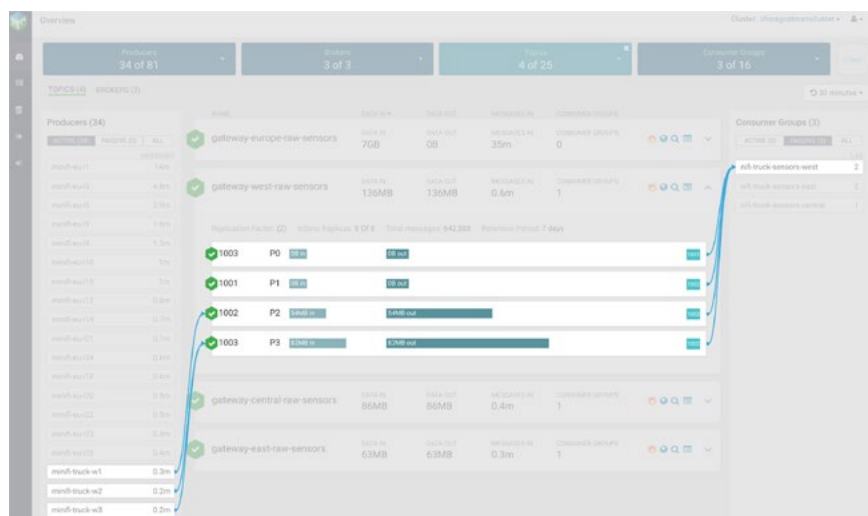
Simplify Troubleshooting Kafka Environments

SMM provides visibility and control to all Kafka teams with intelligent filtering and sorting. Based on a selected producer, broker, topic, or consumer, SMM will display only its related entities. This enables you to quickly hone in on root causes when troubleshooting and debugging Kafka issues. You can also integrate with JMX and Grafana for additional analytics and interactive visualization.



Visualize End-To-End Kafka Stream Flows

From the filtering capabilities described above, interactive visualizations enable you to fully understand how data flows across Kafka clusters. Select any set of entities to analyze throughput and stream dynamics between producers and consumers, for example.





Questions Answered by Streams Messaging Manager for the DevOps Teams

- What are all the entities associated with any application?
- Are any consumers in a consumer-group falling behind for any topic?
- What is the retention rate for my application topics?
- Did a consumer rebalance occur for a given topic?
- How many active consumers are there in a given consumer group?
- What is the replication factor for my application topics?
- Are any of my consumers or consumer-groups over-consuming?
- Are any of my consumers or consumer-groups under-consuming?
- From which topic(s) are the consumer groups consuming messages?
- What is the total number of messages into my topic over the last N minutes/hours?
- Who are all the producers and consumers that are connected to my app topics?
- What type of events are in my application topics?
- What does the event look like?

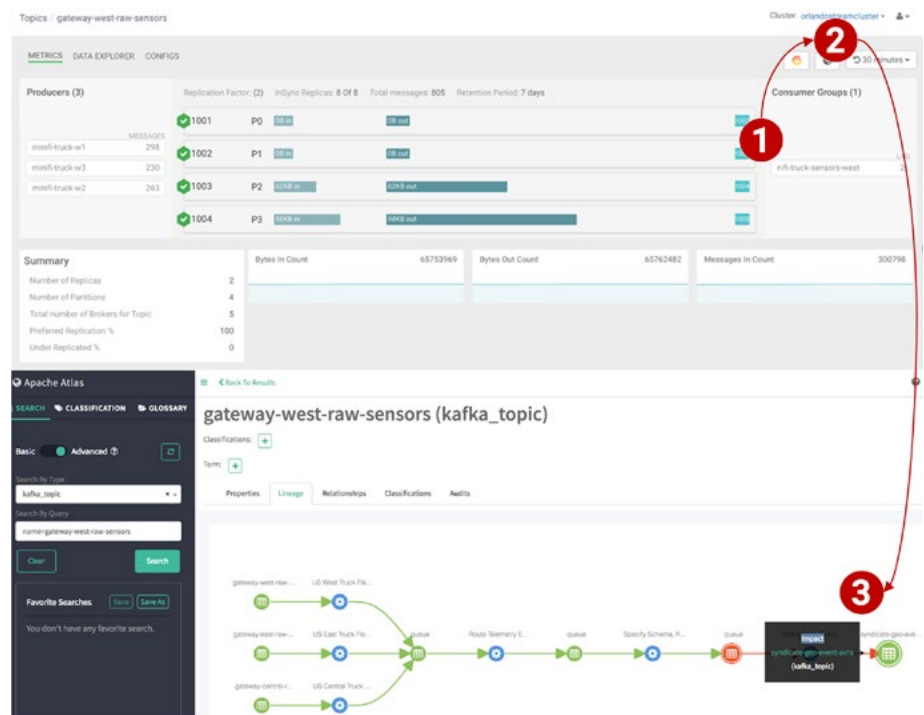
Extend Monitoring/Management Capabilities with REST

The user interface is powered by first class REST services and all SMM capabilities are exposed as REST endpoints, making the product completely extensible. This is a developer and DevOps friendly way to integrate with other enterprise tools such as application performance monitoring and case/ticketing systems.

Track Data Lineage and Governance from Edge-To-Enterprise

Like other integrated components of the Cloudera DataFlow platform, SMM enjoys SDX's unified data security and governance from edge environments across to your enterprise's data center and cloud platforms. This includes Ranger for security and Apache Atlas for end-to-end data governance. With that, you have access to the metadata and metrics about every Kafka topic and can produce complete data lineage and audit trails, even across multiple Kafka hops.

The example below shows how a user can drill down from an edge sensor consumer (1) and launch a data lineage diagram (2) to directly see related flows across Kafka topics (3).



Integration with Schema Registry

Schema Registry, another key component of CDP, has been integrated with SMM, providing the ability to view, create and modify the schema associated with any given Kafka topic. It allows the user to define schemas for a given Kafka topic and provides the following key benefits:

- **Data governance:** Provide reusable schema (centralized registry), define relationships between schemas (version management), and enable generic format conversion and generic routing (schema validation).
- **Operational efficiency:** Avoid attaching schemas to every piece of data (centralized registry), enable consumers and producers to evolve at different rates (version management), and ensure data quality (schema validation).

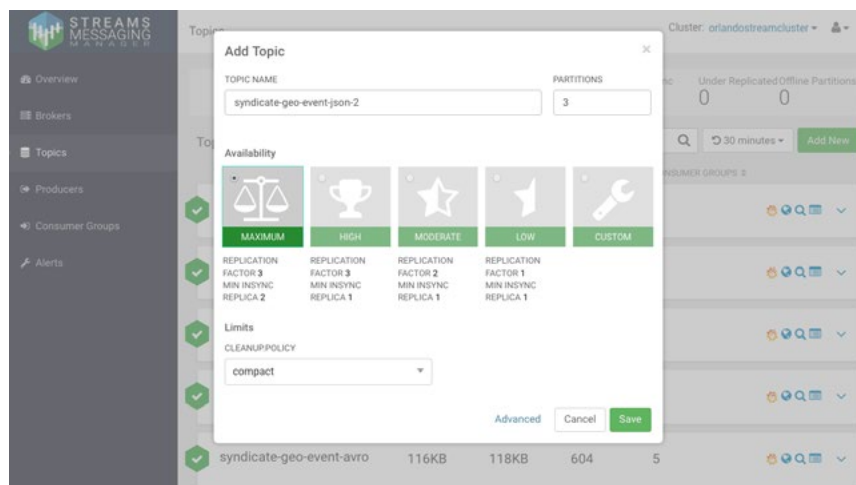


Questions Answered by Streams Messaging Manager for the Application Development Teams

- How can I build good monitoring apps for the operations and support teams?
- How can I streamline development tasks with better management and monitoring tools?
- What APIs are available for me to integrate with SMM?
- Do I have programmatic access to all the key features exposed by the monitoring engine?
- Can I integrate third-party service ticketing solutions with the monitoring tool?
- Can I integrate third-party application monitoring systems with the monitoring tool?
- How can I visualize my data flowing through my topics from producers to consumers?
- How can I introspect and filter the payload of a specific topic?
- How can I troubleshoot the flow across my topics to see where are the bottlenecks?

Topic Lifecycle Management

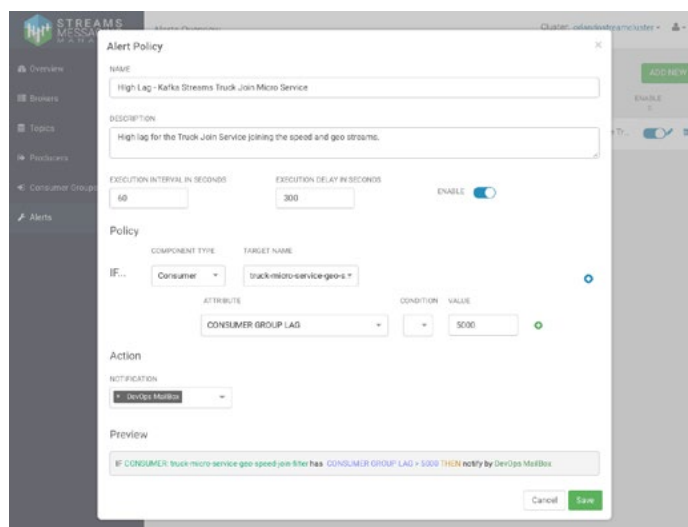
SMM enables users to create, update and delete topics directly through the user interface as well as via REST services. Topics can be created as a function of availability characteristics (replication factor, minimum in-sync replicas, etc.) or with custom settings. These operations are fully integrated with Kafka Ranger policies such that only authorized users can perform these topic lifecycle management actions.



Alerting

Platform Operations and DevOps teams need the ability to create alerts to manage the service level agreement (SLA) of their applications. SMM provides rich alert management features for the critical components of a Kafka cluster including brokers, topics, consumers, and producers by making use of two key constructs.

- **Alert Notifier:** An alert notifier tells SMM what to do when a configured alert is triggered. Out-of-the-box notifiers include sending alerts to a configured email inbox, an HTTP endpoint, or a Kafka topic to integrate alerts with other systems used across the enterprise (e.g: ticketing/case creation systems). The user is also able to configure custom alert notifiers.
- **Alert Policy:** An alert can be defined for any Kafka entity: cluster, broker, topic, producer, or consumer. A set of metrics can be selected to define a series of simple alerts while conditional operators can be used to compose complex alerts that monitor a variety of metrics across a number of entities. The alert policy is also configured with the notifier (above) when the alert fires.





Questions Answered by Streams Messaging Manager for the Security and Governance Teams

- When was a topic created?
- What is the schema for a given topic?
- What is the lineage of a Kafka topic?
- How has the schema evolved for a given topic?
- How does data flow across multiple Kafka hops?
- Who has edited the Access Control Lists (ACL) policies of a given topic?
- Which consumers have consumed from a topic?
- What are the ACL policies for a given topic?
- When were additional brokers added to the topic?
- Which user, group, or service accounts have read from a given topic?
- Which user, group, or service accounts have sent data to a topic?
- When was the topic configuration last modified?
- Which producers have sent data to the topic?

DevOps Monitoring Capability with End-to-End Latency View

Latency is the time taken by a consumer to consume a message that is produced in a topic. Through the SMM user interface, you can monitor the end-to-end latency of all topics as part of DevOps monitoring tasks. For instance, you can verify whether end-to-end processing time SLAs are met, identify slow or lagging consumers, or verify whether messages are over-consumed or under-consumed.



In the example above, the latency view gives you a powerful snapshot of the end-to-end latency. Here we include the number of consumer groups for a topic, number of clients inside a particular consumer group, and number of partitions in a topic along with the messages consumed and end-to-end latency graphs.

Adopt Hybrid Cloud Deployments for Kafka

SMM is available as a part of the Streams Messaging capabilities of CDP. Given that CDP supports different deployment models - on-premises, public cloud and private cloud, SMM can also be deployed flexibly to monitor your hybrid cloud Kafka deployments. SMM is available within CDP to be deployed within your data centers or your private cloud environments. SMM is also available within CDP Data Hub so that it can be spun up along with a Kafka cluster on your favorite public cloud in a matter of minutes.

Address Pain Points Across Your Team

Cloudera Streams Messaging Manager truly addresses the pain points of each of your Kafka teams with a holistic set of integrated capabilities.

PERSONAS	PAIN POINTS	BEN
Kafka Platform Operations teams (see sidebar page 6)	<ul style="list-style-type: none"> • Difficulty troubleshooting / monitoring • Lack of visibility into Kafka clusters • Not sure which brokers are down and if they are running out of disk space • Unable to measure throughput in/out of their Kafka clusters 	<ul style="list-style-type: none"> • Optimize your Kafka environment based on the key performance insights gathered from various brokers and topics • Troubleshoot your Kafka environment to identify bottlenecks, throughputs, consumer patterns, traffic flow etc. • Gain clear insights about your Kafka clusters
Kafka DevOps teams (see sidebar page 7)	<ul style="list-style-type: none"> • Difficulty seeing who is producing and consuming data • Difficulty understanding the flow of data from producers to topics to consumers • Unable to understand consumption patterns of consumers 	<ul style="list-style-type: none"> • Understand the end-to-end flow of message streams from producers to topics to consumers • Analyze the stream dynamics between producers and consumers using various filters
Kafka Application Developers (see sidebar page 8)	<ul style="list-style-type: none"> • Unable to integrate with monitoring applications for operations teams • Lack of Kafka management / monitoring APIs makes building applications cumbersome 	<ul style="list-style-type: none"> • Monitoring / management REST APIs delivered through SMM REST server • Monitoring REST endpoints can be used to integrate with APM / alerting / ticketing solutions • Visualize and search for messages in Kafka topics
Kafka Security and Governance Teams (see sidebar page 9)	<ul style="list-style-type: none"> • Unable to track the metadata about a particular Kafka topic • Difficulty understanding the lineage of data flows into and out of Kafka topics 	<ul style="list-style-type: none"> • Gain complete data lineage across multiple Kafka hops using SMM's powerful data flow visualization and integration with Atlas

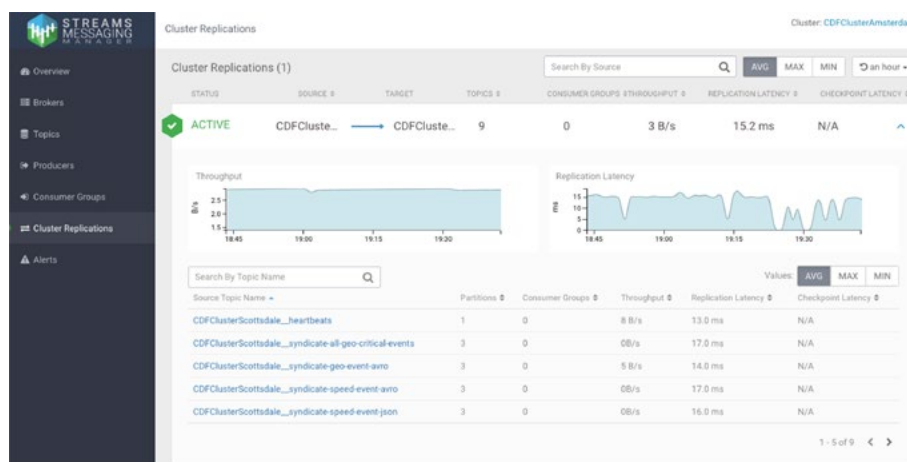
Cloudera Streams Replication Manager (SRM)

While we are focused on delivering key management and monitoring capabilities for important personas across the IT landscape, we are also deliberate about ensuring business continuity and high availability for your streaming architecture. Kafka, being the backbone of such architectures, cannot afford to have any downtime. So, as a part of our Streams Messaging capabilities within CDP, we also offer replication monitoring and management tooling to give you peace of mind.

Streams Replication Manager (SRM) is an enterprise-grade replication solution that enables fault tolerant, scalable and robust cross-cluster Kafka topic replication. As part of Cloudera's aim to deliver a unified toolset, it is directly incorporated into SMM as a window into your replication clusters called, "Cluster Replications View".

SRM provides replication monitoring, details, and metrics at cluster and topic levels including:

- **Status:** Active (replication running), Inactive (replication down), Warning (latency high)
- **All topics** being replicated
- **Throughput:** message brokers replicated between the clusters
- **Replication latency:** time it takes for a message to get replicated from origin to destination



More notably, SRM provides the ability to dynamically change configurations and keep 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.

In addition to overall visibility, SMM with SRM enables you to build and manage replication models that, in the case of an outage, ensure all events continue to be processed without interruption or data loss. The next few pages describe some of the capabilities and innovations of SRM that let all of this happen.

MirrorMaker2

SRM is built on the innovations that Cloudera has brought to MirrorMaker, the original Kafka open source messaging replication tool. MirrorMaker has been with Kafka from the beginning to copy messages from one Kafka cluster to another. Though it fits its original function as a messaging replication tool, it was not designed for disaster recovery and business continuity.

Cloudera addresses some of the severe shortcomings of the original MirrorMaker by unveiling MirrorMaker2, which infuses the concepts of clusters, global configuration, and global management APIs. Because MirrorMaker is used by a number of organizations, MirrorMaker2 retains key aspects that let customers plug into and leverage their existing investments.

Below we summarize some of SRM's key features and innovations.

Remote Topics

SRM innovates on the traditional notion of "remote topics" by formalizing the concept into a solid definition with a standardized naming convention.

In any replication flow, topics are replicated from a source cluster to a remote cluster, and it is at that point that SRM creates a remote topic. The naming convention identifies where the topic resides, its source, and its name, "RemoteClusterB.sourceClusterA.Topic1" for instance. This does two things. First, topics are easy to recognize from a human perspective and second, an automated and configurable "replication policy" enables tooling to distinguish remote topics from source topics across your organization.

Replication Flows

Remote topics are key to enabling users to set up powerful, topic specific replication flows. For example, you can have any number of replicating topologies across clusters but you will never have cycles where data goes from A to B and then copied back to A. Through the creation of remote topics and its naming convention, SRM knows that a topic that originated from A doesn't need to be replicated back to A.

The term replication flow is used to specify all replications set up in a system. Two examples are provided below:

- **Bi-directional replication flows:** As described above, SRM understands cycles and will never replicate records in an infinite loop. This enables bi-directional replication flows in which clusters are mutually replicated. In this case, records sent to one cluster will be replicated to the other and the other way around. You can configure any number of clusters in this way.
- **Fan-in and fan-out:** You can construct fan-in replication flows, where records from multiple source clusters are aggregated in a single target cluster. Similarly, you can construct fan-out replication flows as well, where a single cluster is replicated to multiple target clusters.

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 as needed.

HeartBeats

SRM emits heartbeat messages into a heartbeat topic in every cluster it can reach, confirming that it can replicate records to any one of them. This is how SRM provides a live view of the health of your clusters and other valuable insights. For example, you can look at a local cluster and understand if it is or isn't connected to any other cluster, how far away it is, and the latency between them.

Checkpoints

Checkpoints are another SRM innovation. It is a special message that encodes the progress of consumer groups in a cluster, making it the first replication solution that replicates the state of a consumer group as well. Across the cluster landscape and at configurable intervals, SRM queries each consumer group in each cluster and then emits a checkpoint to the other clusters. The result is that every cluster has a global view of consumer group state, therefore enabling consumer group migration, failover, fallback, and other scenarios.

Offset Syncs

Checkpoints are enabled by offset syncs by providing points of reference. An offset, also called a position, is the sequential order of records produced or consumed and are often out of sync with each other. During a cluster failure, it is difficult to know the position of where your streaming application should resume processing when it has failed over to a healthy cluster. SRM understands that and will keep track any time an offset becomes out of sync between clusters.

Automatic Topic and Partition Detection

SRM monitors Kafka clusters for new topics, partitions, and consumer groups as they are created. This important capability ensures that the platform teams are always aware of the current situation and never caught by surprise.

Advanced Disaster Recovery and High Availability

The capabilities described above establish an enterprise-grade replication solution that suffices demanding disaster recovery and business continuity and resilience requirements. SRM also addresses high availability requirements, ensuring that, in the case of a cluster outage, you continue to process events without data loss. Three bi-directional replication models are described below.

- **Active / Stand-by:** If there is a disaster, producers are directed from the primary cluster to a stand-by cluster to which you can easily migrate your consumer groups (refer also to Checkpoints above). Producers are still able to ingest while the primary cluster is down.
- **Active / Active:** In this case, your producers are load balanced to either your primary or secondary cluster with consumers reading from both clusters at the same time, essentially acting like a cross-cluster consumer group. If the primary cluster goes down, those producers are directed to the secondary cluster. They are still able to ingest and your consumers are still able to process messages. This results in zero downtime and hands-off failover in case of a disaster.
- **Cross data center replication:** You can use SRM to set up replication between Kafka clusters in different data centers so that messages are available to consumers in each of your data centers. If you are using more than two data centers, SRM creates a “replication circle”, ensuring that a single data center failure does not halt replication between the remaining clusters.

In all cases above, SRM automatically synchronizes clusters when all return to service.

Other Business Critical Use Cases

As described above, SRM provides the tools that enterprises need to ensure that data availability and durability guarantees span entire cluster and site failures. However, other business critical use cases are also enabled by SRM and are described below.

- **Aggregation for analytics:** Aggregate data from multiple streaming pipelines and 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 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-premises 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 use case, a policy driven replication is used to limit what data is accessible in a cluster to meet legal and compliance requirements.

About Cloudera

At Cloudera, we believe that data can make what is impossible today, possible tomorrow. We empower people to transform complex data into clear and actionable insights. Cloudera delivers an enterprise data cloud for any data, anywhere, from the Edge to AI. Powered by the relentless innovation of the open source community, Cloudera advances digital transformation for the world's largest enterprises.

Learn more at cloudera.com

Connect with Cloudera

About Cloudera DataFlow:
cloudera.com/cdf

Join the Cloudera Community:
community.cloudera.com

Read about our customers' successes:
cloudera.com/more/customers.html

Next Steps

Cloudera is dedicated to the Kafka ecosystem and continues to be actively involved with the Kafka community through deep engineering relationships with other Kafka committers. This relationship has led to critical innovations and product improvements, many of which have been described here.

This paper introduced two key management components of Cloudera DataFlow's Streams Messaging capabilities that respond to the feedback and experience of our top Cloudera Kafka customers.

First, Stream Messaging Manager provides visibility into how data moves across clusters between Kafka producers, brokers, and consumers while addressing the unique needs of Kafka platform operations, DevOps, application development, security, and governance teams.

Second, Streams Replication Manager enables highly available and resilient cross cluster Kafka replication solutions that address business critical use cases that go beyond disaster recovery.

Besides that, Cloudera has an entire ecosystem of components that support and enhance high performing Kafka environments and the entire streaming data journey, including flow management and stream processing through the Cloudera DataFlow data-in-motion platform.

Go ahead and [test out the Streams Messaging capabilities](#) and other Cloudera DataFlow components to find out how easy it is to set it up, configure, and gather key insights about your Kafka clusters that you had never been able to do before.