

Configure and Manage

Overview

Configuration Reference

Control Center

CLI Tools for Use with Confluent Platform

Change Configurations Without Restart

Manage Clusters

Overview

Kafka Metadata Management in Confluent Platform

Metadata Management of Kafka in Confluent Platform

KRaft Overview

Configure KRaft

Configure ZooKeeper

Manage Self-Balancing Clusters

Auto Data Balancing

Tiered Storage in Confluent Platform

Metadata Service (MDS) in Confluent Platform

Docker Operations for Confluent Platform

Run Kafka in Production

Production Best Practices

Monitor

Monitor with Health+

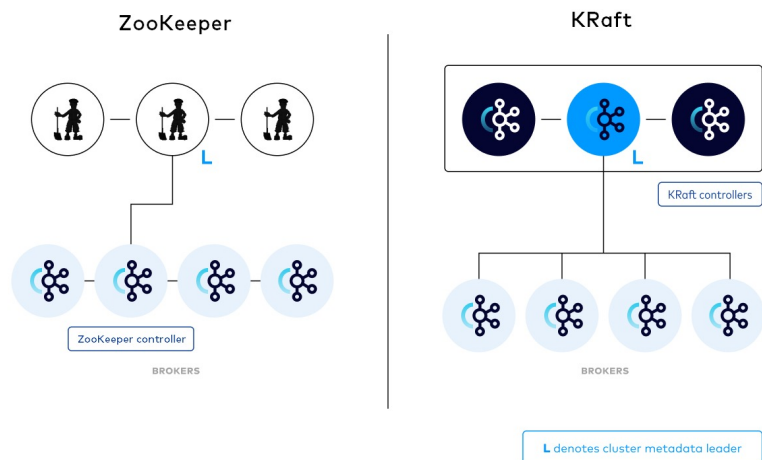
Confluent CLI

# KRaft Overview for Confluent Platform

Previously, when you ran Confluent Platform, you also always ran ZooKeeper for metadata storage. Starting with Confluent Platform version 7.4, KRaft (pronounced craft) mode is generally available.

Apache Kafka® Raft (KRaft) is the consensus protocol that was introduced to remove Kafka's dependency on ZooKeeper for metadata management. This greatly simplifies Kafka's architecture by consolidating responsibility for metadata into Kafka itself, rather than requiring and configuring two different systems: ZooKeeper and Kafka.

The following image provides a simple illustration of the difference between running with ZooKeeper and KRaft for managing metadata.



Note: This shows KRaft running in Isolated Mode

## Why move to KRaft?

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

Accept All Cookies

Reject All

Cookies Settings

er is a separate system, deployment patterns, administrators. Specifically:

d with the appropriate throughput and latency



- Metadata failover is near-instantaneous with KRaft.
- KRaft provides a single system to manage since KRaft is part of Kafka itself. Single management model for authentication, configuration, protocols, and networking.

## The controller quorum

The KRaft controller nodes comprise a Raft quorum which manages the Kafka metadata log. This log contains information about each change to the cluster metadata. Everything that is currently stored in ZooKeeper, such as topics, partitions, ISRs, configurations, and so on, is stored in this log.

Using the Raft consensus protocol, the controller nodes maintain consistency and leader election without relying on any external system. The leader of the metadata log is called the active controller. The active controller handles all RPCs made from the brokers. The follower controllers replicate the data which is written to the active controller, and serve as hot standbys if the active controller should fail. With the concept of a metadata log, brokers use offsets to keep track of the latest metadata stored in the KRaft controllers, which results in more efficient propagation of metadata and faster recovery from controller failovers.

Just like ZooKeeper, KRaft requires a majority of nodes to be running. For example, a three-node controller cluster can survive one failure. A five-node controller cluster can survive two failures, and so on.

Periodically, the controllers will write out a snapshot of the metadata to disk. This is conceptually similar to compaction, but state is read from memory rather than re-reading the log from disk.

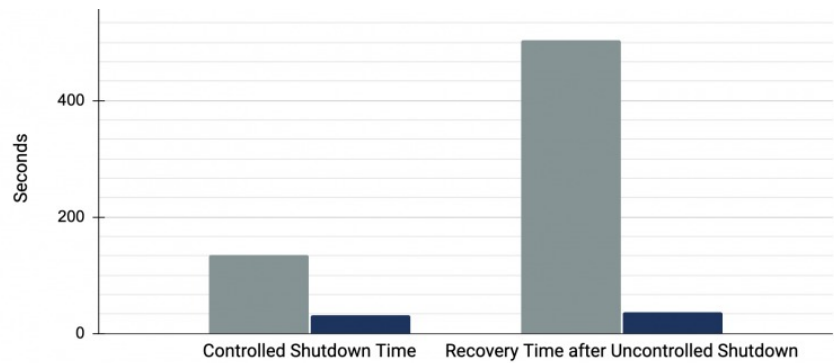
## Scaling Kafka with KRaft

There are two properties that determine the number of partitions an Kafka cluster can support: the per-node partition count limit and cluster-wide partition limit. Previously, metadata management with ZooKeeper was the main bottleneck for the cluster-wide limitation. KRaft mode is designed to handle a much larger number of partitions per cluster, however Kafka's scalability still primarily depends on adding nodes to get more capacity, so the cluster-wide limit still defines the upper bounds of scalability within the system.

With ZooKeeper, the upper bound of partitions was approximately 200,000, with the limiting factor being the time taken to move critical metadata between external consensus (ZooKeeper) to internal leader management (the Kafka controller). In KRaft, the new quorum controller serves both these roles. The result of this change is a near-instantaneous controller failover. The following image shows the results of a Confluent lab experiment on a Kafka cluster running 2 million partitions, which is 10 times the maximum number of partitions for a cluster running ZooKeeper. The experiment shows that controlled shutdown time and recovery time after uncontrolled shutdown are greatly improved with a quorum controller versus ZooKeeper.

**Timed Shutdown Operations In Apache Kafka with 2 Million Partitions**  
*Faster is better*





---

## Migrate from ZooKeeper to KRaft

If you want to migrate your existing Kafka clusters from ZooKeeper for metadata management to KRaft, see [Migrate from ZooKeeper to KRaft on Confluent Platform](#).

---

## Configure Confluent Platform with KRaft

For details on how to configure Confluent Platform with KRaft, see [KRaft Configuration for Confluent Platform](#).

Client configurations are not impacted by Confluent Platform moving to KRaft to manage metadata.

---

## Limitations and known issues

- Combined mode, where a Kafka node acts as a broker and also a KRaft controller, is not currently supported for production workloads. There are key security and feature gaps between combined mode and isolated mode in Confluent Platform.
- JBOD (just a bunch of disks) is not supported in KRaft mode, meaning you can only configure one directory for the `log.dirs` configuration.
- Source-initiated [Cluster Linking for Confluent Platform](#) between a source cluster running Confluent Platform 7.0.x or earlier and a destination Confluent Platform cluster running in KRaft mode is not supported. Source-initiated link creation may succeed, but the connection will ultimately fail. To work around this issue, make sure the source cluster is running Confluent Platform 7.1.0 or later; or use a default cluster link, not a source-initiated link. To avoid this issue if you have source-initiated links from a Confluent Platform source cluster to a Confluent Cloud destination cluster, you must upgrade your source clusters to Confluent Platform 7.1.0 or later, or use a default cluster link instead.
- There is currently no support for quorum reconfiguration, meaning you cannot add more KRaft controllers, or remove existing ones in your cluster. 3 or 5 controllers are generally recommended for production, but you should have at least 3. For more information, see [Hardware](#).

- You cannot currently use [Schema Registry Topic ACL Authorizer for Confluent Platform](#) for Schema Registry with Confluent Platform in KRaft mode. As an alternative, you can use [Schema Registry ACL Authorizer for Confluent Platform](#) or [Configure Role-Based Access Control for Schema Registry in Confluent Platform](#).
- Currently, Health+ reports KRaft controllers as brokers and as a result, alerts may not function as expected.


---

## Related content


- [Metadata Management of Kafka in Confluent Platform](#)
- [KRaft Configuration for Confluent Platform](#)
- [Migrate from ZooKeeper to KRaft on Confluent Platform](#)
- Watch the [Confluent blog](#) and [Release Notes for Confluent Platform 7.7](#) for additional announcements about the rollout of KRaft in Confluent Platform.
- [KRaft: Apache Kafka Without ZooKeeper](#)
- [The Truth About ZooKeeper Removal and the KIP-500 Release in Apache Kafka \(podcast\)](#)
- [A Kafkaesque Raft Protocol \(video\)](#)

---

Was this doc page  
helpful?

 [Give us feedback](#)

Do you still need help?

 [Confluent support portal](#)

 [Ask the community](#)

## Be the first to get updates and new content

Email

**SIGN UP**

By clicking "SIGN UP" you agree that your personal data will be processed in accordance with our [Privacy Policy](#).

Confluent

About

Careers

Contact

Professional Services

Product

Confluent Cloud

ksqlDB

Developer

Community

[Free Courses](#)

[Forum](#)

[Tutorials](#)

[Meetups](#)

[Event Streaming Patterns](#)

[Kafka Summit](#)

[Documentation](#)

[Catalysts](#)

[Blog](#)

[Podcast](#)



[Terms & Conditions](#) | [Privacy Policy](#) | [Do Not Sell My Information](#) | [Modern Slavery Policy](#) |

[Cookie Settings](#) | [Feedback](#)

Copyright © Confluent, Inc. 2014- 2024 Apache, Apache Kafka, Kafka, the Kafka logo, Apache Flink, Flink, the Flink logo, Apache Iceberg, Iceberg, and the Iceberg logo are trademarks of the **Apache Software Foundation**