KIP-833: Mark KRaft as Production Ready

Status

Current state: *Accepted*

Discussion thread: https://lists.apache.org/thread/90zkqvmmw3y8j6tkgbg3md78m7hs4yn6

JIRA: n/a

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Proposed Changes

For several years, we have been developing a new way to run Kafka with self-managed metadata. This new mode, called KRaft mode, addresses many urgent scalability and performance issues in Kafka. It also eliminates the need to run an Apache ZooKeeper cluster alongside every Kafka cluster. See KIP-500 for more details.

In Kafka 2.8, KRaft mode was released in early access. By Kafka 3.0, it was in preview.

This KIP proposes to:

1. Mark KRaft as production-ready for new clusters in the upcoming Kafka 3.3 release.
2. Deprecate ZooKeeper mode in the upcoming Kafka 3.5 release
3. Plan to remove ZooKeeper mode entirely in Kafka 4.0.

Rationale

Over the last year and a half, we have gained a lot of confidence in the KRaft implementation. Our JUnit test coverage has grown as we have converted existing tests and written brand new tests. We have added support for KRaft mode to Kafka's "ducktape" system test framework. We have also run load tests and stress tests on KRaft clusters.

KIP-746: Revise KRaft Metadata Records fixed several issues with the original record definitions established in Kafka 2.8. KIP-778: KRaft upgrades spelled out a clear path for handling changes to KRaft metadata in a backwards-compatible way. Overall, we feel that the next release of Kafka, the 3.3 release, will be a good time to remove the "only for testing" warning from KRaft and mark it as ready for production.

The rationale for deprecating ZK in the 3.4 release is so that we can remove it in the 4.0 release. (In general, Kafka requires features to be deprecated for at least one release before they can be removed in the following major release.) During the deprecation period, ZK mode would continue to be fully supported. However, we would let people know that KRaft mode is the future.

Once ZK mode is removed in Kafka 4.0, we will be able to avoid maintaining parallel codebases for important infrastructure such as the Kafka controller.

Release Timeline

Note: this timeline is very rough and subject to change. However, it's intended to help people associated dates with release numbers. We are assuming that Kafka will continue to use time-based releases that happen every 4 months.

Kafka 3.3 (released)

- August 2022
- KRaft mode declared production-ready

Kafka 3.4 (released)

- January 2023
- Migration from ZK mode supported as Early Access (EA)

Kafka 3.5

- May 2023
- SCRAM support added for KRaft
- ZK mode deprecated
- Migration from ZK mode supported as Preview.

Kafka 3.6

- September 2023
- Migration from ZK mode supported as GA.
- Delegation token support in KRaft (planned)
- JBOD support in KRaft (planned)

Kafka 3.7

- January 2024
- Final release with ZK mode

Kafka 4.0

- April 2024
- Only KRaft mode supported.

Missing Features

As of the date of writing, the following features have not yet been implemented in KRaft mode.

- Supporting JBOD configurations with multiple storage directories Unable to render Jira issues macro, execution error.
- Modifying certain dynamic configurations on the standalone KRaft controller Unable to render Jira issues macro, execution error.
- Delegation tokens Unable to render Jira issues macro, execution error.

As outlined above, we expect to close these gaps soon.

Compatibility, Deprecation, and Migration Plan

As discussed in KIP-500 and the follow-on KIPs, KRaft clusters expose the same public APIs as ZK-based Kafka clusters. Since the modern Kafka producer and consumer do not directly access ZooKeeper, no changes are needed to existing Kafka clients. While some administrative tools still have options which enable them to perform access to ZooKeeper, this mode of access has been deprecated since mid-2020, as described in KIP-555. Therefore, we think the transition for clients should be smooth.

Bridge Releases

Starting with 3.5, all the 3.x releases will be "bridge releases" as described in KIP-500. What this means is that they can act as a bridge from the ZK world to the KRaft world.

Users who have a ZooKeeper-based Kafka and want to upgrade to a post-ZK version (4.0 and beyond) will have to first upgrade to one of these bridge releases. Then, once they are in KRaft mode, they will be able to upgrade to any 4.x release.

Potential Objections and Responses

One possible objection is that we should not declare KRaft to be production ready until it has reached full feature parity with ZooKeeper mode. My response to this is that we expect to close the feature gaps relatively quickly. There is still a lot of value in marking KRaft as production ready now. In addition to enabling more usage, it will ensure that we do not end up with more feature gaps to fill over the next few months.

Another possible objection is that we should keep ZooKeeper mode around in the 4.x series of releases. My response here is that supporting both modes is a burden for people maintaining and adding to the code. Also, it's worth keeping in mind that even if we commit to removing ZK in 4.0, we always have the option of doing more 3.x releases.