

KIP-785: Automatic storage formatting

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

This page is meant as a template for writing a [KIP](#). To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: *Under Discussion*

Discussion thread: [here](#)

JIRA: [KAFKA-13382](#)

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

Motivation

Configuring Kafka shouldn't be more complicated than it needs to be.

KIP-631 introduced a required step to run Kafka brokers in KRaft mode - running the command `kafka-storage.sh` to format the storage directories. The KIP mentions why the functionality was designed to be a command, separate from the broker process:

Before being used in KIP-500 mode, the storage directories on a node must be formatted. This requirement prevents system administrators from accidentally enabling KIP-500 mode by simply making a configuration change. Requiring formatting also prevents mistakes, since Kafka no longer has to guess if an empty storage directory is a newly directory or one where a system error prevented any data from showing up.

Two intentions are stated above to have the format operation as a pre-requisite separate step

1. Preventing accidental enabling of KRaft mode.
2. Avoiding the need to distinguish between a new storage directory and a faulty one.

Today, we can empirically verify that actually enabling KIP-500 (KRaft mode) is not so simple to the point it could easily be done by accident. There isn't a single and trivial configuration change that enables it, but rather there are several required configurations:

- `process.roles`
- `node.id`
- `controller.quorum.votes`
- `controller.listener.names`

As to the second intention, avoiding the need to distinguish between a new storage directory and a faulty one is not a good trade-off. The responsibility of automating this decision is passed on to the operator, increasing the complexity of a deployment. Kafka already has mechanisms to detect faulty storage directories and the same technique can be used to automatically distinguish between these two scenarios.

Not requiring running a separate process before launching Kafka brokers has significant benefits. A deployment setup for Kafka requires additional complexity to correctly identify the first time storage directories will be used and run the format storage command before launching Kafka. This overhead is somewhat mitigated by the `--ignore-formatted` flag in the `kafka-storage.sh` command which causes the command to have no effect if the storage is already formatted. But configuring and running this command as an initial step to starting a KRaft server still constitutes unnecessary complexity. It is typical to generate configuration for every broker based on a common base configuration or template. This same configuration is expected by `kafka-storage.sh` to be available via the `--config` parameter.

Operational simplicity is another reason why operators would want to run the command before every broker start. When using multiple log directories configured for different disks, the process to replace a failed disk becomes simpler. After restarting the broker, `kafka-storage.sh` would run automatically and not require additional interaction by the operator.

Public Interfaces

Two new configuration options will be introduced — `cluster.id` and `auto.storage.format`. When both options are present, during startup in a KRaft broker, the storage directories are "formatted" if necessary. The operation is mostly equivalent of the following command is run at startup:

```
bin/kafka-storage.sh format \
  --cluster-id <cluster.id> \
  --config <broker.properties> \
  --ignore-formatted
```

However, if some, but not all, of the storage directories are not accessible the operation does not fail, and Kafka starts with the available storage directories.

Proposed Changes

New configuration:

- `cluster.id: <string>` — Indicates the cluster ID. The value is the same that is currently used with the `--cluster-id` flag in `kafka-storage.sh`.
- `auto.format.storage: <boolean>` — This additional flag gates the automatic operation of formatting storage, if it needs to be formatted.

The `kafka-storage.sh` command does not require `--cluster-id` if the `cluster.id` is present in configuration.

Compatibility, Deprecation, and Migration Plan

The new functionality is gated by the presence of two new configuration properties, so the operator needs to take deliberate action to enable the feature.

The feature, once enabled is idempotent. It should have no effect if the storage directory is already formatted.

We can keep the recommendation to use the storage formatting command explicitly, and we can have that be a step in any eventual ZooKeeper KRaft migration routes, but still support this feature to simplify the operation of Kafka post-KRaft transition.

Rejected Alternatives

- Maintaining the current requirement, where an additional process is required to start before Kafka is launched. If configuring and operating Kafka can safely be made simpler, we should aim to make it so.
- Using only a new single configuration to gate the automatic formatting feature, so that the original goal of preventing accidental mistakes is avoided.