KIP-920: Support upgrading KRaft MetadataVersion based on a static config

Status

Current state: Under Discussion

Discussion thread:

JIRA:

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

Motivation

Background

KRaft mode introduced a new way of handling the Kafka "inter-broker protocol." Previously, it had been determined by the value of the static configuration inter.broker.protocol. KRaft mode changed the name of the configuration to "metadata version" and made it so that it was managed and set dynamically, rather than statically.

While being able to set the metadata version dynamically is helpful, there are some cases where we would like to set a static configuration and have the metadata version be automatically applied. One example is when using configuration systems like ansible, puppet, chef, or certain kubernetes operators which manage state by means of updating configuration files. This KIP intends to add an alternate static configuration path for updating MetadataVersion, which can be used instead of the dynamic path if the operator so wishes.

Public Interfaces

metadata.version configuration

Now that the inter-broker protocol has been renamed to "metadata version," it is appropriate for the name of the static configuration to also be updated. Therefore, this KIP adds `metadata.version` as a synonym for `inter.broker.protocol` and deprecates the old configuration name.

auto.upgrade.metadata.version configuration

We will add a new dynamic broker-level configuration named `auto.upgrade.metadata.version` . It defaults to `false` unless otherwise specified.

auto.upgrade.metadata.version	Cluster Mode	Effect
false	ZK	none
false	KRaft	none
true	ZK	none
true	KRaft	Every 5 minutes, if the current MetadataVersion is less than the statically configured value, it will automatically upgrade the metadata version

Note that if the configured metadata version is less than or equal to the statically configured value, we will not perform any action.

RPC Changes

MANUAL_METADATA_VERSION_MANAGEMENT_DISABLED

This error code will be returned from UpdateFeaturesRequest to indicate that we can't dynamically change the metadata version because `auto.upgrade.metadata.version` is enabled.

The user can fix this by dynamically changing the value of `auto.upgrade.metadata.version` to false.

UpdateFeaturesRequest

There will be a new version of `UpdateFeaturesRequest` to reflect the possible presence of the new error code.

```
diff --git a/clients/src/main/resources/common/message/UpdateFeaturesRequest.json b/clients/src/main/resources
/common/message/UpdateFeaturesRequest.json
index 27ed8420fb..dcc31909ab 100644
--- a/clients/src/main/resources/common/message/UpdateFeaturesRequest.json
+++ b/clients/src/main/resources/common/message/UpdateFeaturesRequest.json
@@ -18,7 +18,8 @@
   "type": "request",
   "listeners": ["zkBroker", "broker", "controller"],
   "name": "UpdateFeaturesRequest",
-  "validVersions": "0-1",
+  // Version 2 adds support for the MANUAL_METADATA_VERSION_MANAGEMENT_DISABLED error.
+  "validVersions": "0-2",
   "flexibleVersions": "0+",
   "fields": [
     { "name": "timeoutMs", "type": "int32", "versions": "0+", "default": "60000",
diff --git a/clients/src/main/resources/common/message/UpdateFeaturesResponse.json b/clients/src/main/resources
/common/message/UpdateFeaturesResponse.json
index 033926b801..ef667a6fe9 100644
--- a/clients/src/main/resources/common/message/UpdateFeaturesResponse.json
+++ b/clients/src/main/resources/common/message/UpdateFeaturesResponse.json
@@ -17,7 +17,8 @@
   "apiKey": 57,
   "type": "response",
   "name": "UpdateFeaturesResponse",
-  "validVersions": "0-1",
+  // Version 2 adds support for the MANUAL_METADATA_VERSION_MANAGEMENT_DISABLED error.
+  "validVersions": "0-2",
   "flexibleVersions": "0+",
   "fields": [
     { "name": "ThrottleTimeMs", "type": "int32", "versions": "0+",
```

The `MANUAL_METADATA_VERSION_MANAGEMENT_DISABLED` error indicates that the user attempted to use `UpdateFeaturesRequest` to change the metadata version, but `auto.upgrade.metadata.version` was set to true. In versions of `UpdateFeaturesRequest` prior to 2, `INVALID_REQUEST` will be returned instead for this case.

Compatibility, Deprecation, and Migration Plan

There should be no compatibility impact since `auto.upgrade.metadata.version` defaults to false. In other words, the new behavior is opt-in.

Rejected Alternatives

Allowing automatic downgrade

One obvious question is why the proposed mechanism can only do upgrades, not downgrades. The reason is because we cannot guarantee that the statically configured value of `metadata.version` is the same on all controller nodes. If the value of the configuration differs between controllers nodes, and downgrades are possible, this could result in a "flip-flop" situation as different controller nodes become active. This would be very disruptive to the cluster. (While this case may seem rare, it is actually the common case during a controller cluster roll that happens to roll out a new static configuration!)

There are other ways to avoid the flip-flop situation, but only allowing upgrades is a simple one which accomplishes our main goals very well. We expect downgrades to be rare, so it is OK for them to be handled manually.