KIP-NEXT: Support dynamic update of delete.topic.enable config

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 [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: *here*

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

Motivation

Topic deletion in Kafka removes data and the operation is not reversible (i.e. there is no "undelete" operation). Hence, there is a valid use-case to keep the flag "delete.topic.enable" as "false" in `server.properties` configs, so that someone doesn't accidentally delete a topic, which can then lead to outages.

But sometimes, there are legit client use-cases, where the users want to deprecate a topic and those topics need to be deleted from the cluster. Currently, the process to do this operation is very cumbersome (with "delete.topic.enable" flag set as "false"):

```
1. Change the `server.properties` config on each broker with: `delete.topic.enable=true`
2. Roll the cluster
3. Run Kafka Admin commands to delete the topics
4. Revert the `server.properties` config change for `delete.topic.enable`
5. Roll the cluster again
```

There are other hacky workarounds as well, like:

```
1. Set Topic retention of the topic-to-be-deleted to a small value to flush the data
2. Invoke `zkcli.sh rmr /brokers/topic/topic-to-be-deleted`
3. When Topic metadata is no longer available, rm the topic dirs on the brokers
```

The above process is pretty risky and can lead to unavailability of one or more topics if any mistake happens in executing the above commands.

Proposed Changes

KIP-226 added support for dynamic update of broker configuration. In this KIP, we propose to extend the support to dynamic update of "delete.topic.enable" config flag.

Use case: To allow deletion of topics from the cluster dynamically without restarting the brokers (and otherwise keeping the config "delete.topic.enable" as "false" in the static server.properties config).

Config scope:

Default for whole cluster (/configs/brokers/<default>)

Broker-level config (for testing) (/configs/brokers/<broker-id>)

Config options:

delete.topic.enable

Dynamic update changes:

`TopicDeletionManager.isDeleteTopicEnabled` will be updated on dynamic flag reconfiguration. This variable flag will be used at all places in lieu of `KafkaConfig.deleteTopicEnable`.

Sample Workflow to delete a topic after the proposed changes:

1. `bin/kafka-configs --bootstrap-server localhost:9092 --entity-type brokers --entity-default --alter --add-config delete.topic.enable=true`

2.
   ```
   bin/kafka-topics.sh --zookeeper localhost:2181 --delete --topic TopicToBeDeleted
   ```
3.
   ```
   bin/kafka-configs --bootstrap-server localhost:9092 --entity-type brokers --entity-default --alter --add-config delete.topic.enable=false
   ```

Compatibility, Deprecation, and Migration Plan

- There won't be any impact on existing users.
- There won't be any change of current behavior.
- No migration tool required

Rejected Alternatives

- None