# KIP-934: Add DeleteTopicPolicy

## Status

**Current state**: Under Discussion

**Discussion thread**: https://lists.apache.org/thread/hqpg2sbmkxp8c8prhjt0cgt5n2xd4ocw

**JIRA**: ⚠ Unable to render Jira issues macro, execution error.

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

## Motivation

Currently, it's only possible to add policies for Topic creation and configuration updates via `CreateTopicPolicy` (introduced in KIP-108) and `AlterConfigPolicy` (introduced in KIP-133), but not for Topic deletion.

Topic deletion policies would enable operators to control how to proceed when topic deletions are requested.

Implementations of this policy could be used:

- As additional safeguard when deletion of internal topics is requested, even when a requester is authorized (i.e. has an ACL allowing to deleted).
  - A policy would be useful to avoid the deletion in this case, e.g. deleting Kafka internal topics `__consumer_offsets` or application internal topics like Connect internal topics.
  - Authorizer implementations tend to prefer coarse grained permissions to reduce the number of ACLs (e.g. Confluent RBAC which includes Delete as part of ResourceOwner) which includes both permissions to create *and* delete, making it harder to rely only on the Authorizer.
- To validate against a registry when a topic is deleted, e.g. for cross domain usage.
- To implement retry-based or flag-based mechanisms to safeguard topic deletion.

There has been related KIPs that included this proposal:

- KIP-170: Enhanced TopicCreatePolicy and introduction of TopicDeletePolicy (retired and superseded by KIP-201)
- KIP-201: Rationalising Policy interfaces (called abandoned: https://github.com/apache/kafka/pull/4281#issuecomment-1035154386)

This KIP is intended to reduce the scope of the proposal for topic deletion only, following approach from existing policies.

This KIP borrows parts of KIP-170. If KIP-201 is resurrected, this changes shouldn't increase complexity of the KIP as it follows TopicCreatePolicy approach and same migration should apply.

## Public Interfaces

1. New interface on clients module:

```
package org.apache.kafka.server.policy;

public interface DeleteTopicPolicy extends Configurable, AutoCloseable {

    class RequestMetadata {
        private final String topic;
        private final Uuid id;

        public RequestMetadata(String topic, Uuid id) {
            this.topic = topic;
            this.id = id;
        }

        public String topic() {
            return topic;
        }

        public Uuid id() {
            return id;
        }
    }

    void validate(RequestMetadata requestMetadata) throws PolicyViolationException;

}
```

2. New configuration for brokers:

`delete.topic.policy.class.name` : The delete topic policy class that should be used for validation. The class should implement the `org.apache.kafka.server.policy.DeleteTopicPolicy` interface.

3. New version of DeleteTopicsRequest protocol message:

```
DeleteTopics Request (Version: 7) => [topics] timeout validate_only
  topics => STRING
  timeout => INT32
  validate_only => BOOLEAN
```

## Proposed Changes

Apart from the Interfaces proposed, the changes will follow the same approach as `TopicCreatePolicy`.

Changes:

- DeleteTopicsRequest:
  - Bump to version 7
  - Add `validateOnly` flag to `DeleteTopicsRequest.json`
  - Add POLICY_VIOLATION as possible error code on `DeleteTopicsResponse`
  - Add options to `DeleteTopicsOptions`
  - Use new flag on `KafkaAdminClient`
  - Extend `Controller#deleteTopics` interface to include `DeleteTopicsRequestData` and update implementations
- DeleteTopicPolicy:
  - Add policy config to `KafkaConfig`
  - Load policy and pass it to `Controllers` and `ZKAdminManager` .
  - Use policy on `ReplicationControlManager`

## Compatibility, Deprecation, and Migration Plan

- *What impact (if any) will there be on existing users?*

No impact to existing users. All are new APIs, and should not have any compatibility issues apart from validating that `validateOnly` flag on protocol is only requested for version 7 of `DeleteTopicsRequest` message.

## Test Plan

Fairly similar to `TopicCreatePolicy`, checking that config is loaded properly, and policy validation returns proper exception.

## Rejected Alternatives

- Protecting Topics with ACLs:
    - As stated on the motivation, this approach is insufficient, as even when authorization is there topic deletion errors can occur.