KIP-823: Update Admin::describeConfigs to allow fetching specific configurations

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

`AdminClient::describeConfigs` api takes a `Collection` of `ConfigResource` objects as an argument to get all the configurations of the entities specified. Here is the api signature:

> *DescribeConfigsResult describeConfigs(Collection<ConfigResource> resources, DescribeConfigsOptions options);*

The `ConfigResource` class is made up of two fields, name of the entity ("topic-1", "broker-1" etc) and type of the entity (BROKER, TOPIC etc). `DescribeConfigsOptions` allows users to specify whether to fetch config synonyms and their documentation. The api response contains all configurations for the specified entities.

This admin api in turn calls `DescribeConfigsRequest` kafka api to get the configuration for specified entities. `DescribeConfigsRequest` api takes a collection of `DescribeConfigsResource` objects to specify entities whose configuration needs to be fetched. So to make this Kafka API call, `AdminClient::describeConfigs` converts `ConfigResource` collection passed to it to a `DescribeConfigsResource` collection. In addition to name and type of entity whose configuration to get, Kafka `DescribeConfigsResource` structure also lets users provide `ConfigurationKeys`, a list of String, which allows users to specify only the configurations that they are interested in. As this information isn't present in the `ConfigResource` class, it is set to `null` when `DescribeConfigsResource` object is created from it. Here is the code doing this (KafkaAdminClient.java)

```
.map(config ->
    new DescribeConfigsRequestData.DescribeConfigsResource()
        .setResourceName(config.name())
        .setResourceType(config.type().id())
        .setConfigurationKeys(null))
```

This means that all configurations of all the entities specified are returned by Kafka. Then the user of the `AdminClient::describeConfigs` iterates over the returned list and filters out the configuration keys that they are interested in.

This results in boilerplate code for all users of `AdminClient::describeConfigs` api, in addition to being wasteful use of resource. It becomes painful for large clusters where to fetch one configuration of all topics, we need to fetch all configuration of all topics, which can result in huge response. Alternatively, request can be batched, but then one api request gets broken down in tens if not hundred of api requests depending on batch size, complicating error handling and retries.

This is also a usability issue when `kafka-topics` is used with `--describe` option and no topic name is provided. We get all configurations of all the topics, where a user may only be interested in only one or few of the configurations. For `kafka-topics` command we also need a way to skip getting configuration so that partition information of all topics can be fetched w/o fetching their configurations.

We need to a way to specify `ConfigurationKeys` parameter that `DescribeConfigsResource` takes to bring `AdminClient::describeConfigs` api to parity with `DescribeConfigsResource` and allow AdminClient's users to specify configuration keys that they are interested in.

In addition we need to add same option to `kafka-topics` command line utility so that users of the tool don't need to fetch all configurations when they are interested in only a few or none of them.

Public Interfaces

We will modify the `DescribeConfigsOptions` argument of the `AdminClient::describeConfigs` api to take an additional parameter `List<String> configurationKeys`, which will be same as one that the `DescribeConfigsResource` takes so that `AdminClient::describeConfigs` users can specify configurations to fetch. An empty or null value for this will behave as it behaves today by fetching all configuration values. Here is how the new `DescribeConfigsOptions` structure will look like after this change:

```
@InterfaceStability.Evolving
public class DescribeConfigsOptions extends AbstractOptions<DescribeConfigsOptions> {

    private boolean includeSynonyms = false;
    private boolean includeDocumentation = false;
    private List<String> configurationKeys = null;   // This is the newly introduced field
...
...
    public List<String> configurationKeys() {
      return configurationKeys;
    }
...
...
    public void configurationKeys(List<String> configurationKeys) {
      this.configurationKeys = configurationKeys;
    }
...
}
```

In addition to this we propose two changes to the `kafka-topics` command line tool :

1. Allow `--config` option to be specified when using `--describe` option. When used in such a way, only the configuration(s) specified will be fetched for the topic.
2. Add --partition-only option that when used with `–describe` will skip fetching configuration of the topics and will only return partition information.

Proposed Changes

Previous section lists the most of the changes that are needed. On the implementation side, when we convert `DescribeConfigsOptions` argument to Kafka apis `DescribeConfigsResource` object, we will fetch the list of configuration keys from the `DescribeConfigsOptions` and put that in the `DescribeConfigsResource`, so the new code will look like this:

```
.map(config ->
    new DescribeConfigsRequestData.DescribeConfigsResource()
        .setResourceName(config.name())
        .setResourceType(config.type().id())
        .setConfigurationKeys(options.configurationKeys()))  // This is the change, instead of passing `null`
we will pass the keys specified by user
```

Couple of things to note here:

1. A user can specify a configuration key that isn't valid for the resource type specified. In this case the response will be empty and no configuration will be returned. This matches current behavior of `DescribeConfigsRequest` Kafka Api. In that sense AdminClient will just reflect the existing behavior of underlying Kafka api and will not be modifying it.
2. A user can specify a mix of resource types as first argument to `AdminClient::describeConfigs` api, i.e. the `resources` argument can contain a resource of `BROKER` type and also a resource of `TOPIC` type. When fetching all configurations this behavior has no significance. However when user specifies configurations to fetch, then unless somehow the configuration key happen to exist for all specified resources, all except one of the resource will have empty configurations returned. In that sense this new feature is mostly useful when clients of `AdminClient::describeConfigs` are fetching configuration for resource of only one type.

Compatibility, Deprecation, and Migration Plan

There is no compatibility issue as we are adding a new field to an existing data structure (`DescribeConfigsOptions`). If the field is not specified then the code will behave in same manner as existing code. The same applies to changes to the `kafka-topics` tool, as we are adding new options and existing options will continue to behave as is.

Rejected Alternatives

We considered adding `configurationKeys` collection directly to the `ConfigResource` object so that user can specify different type of resources and for each one of them they can also specify the configurations to fetch. This avoid the issue #2 mentioned above in the `Proposed Changes` section. However, `ConfigResource` class is widely used outside of this api (over 300 usages) and is meant to represent a resource that has configurations. The new configuraiton key field will be of no use in all these other cases, so we decided to not pursue this approach.