KIP-220: Add AdminClient into Kafka Streams' ClientSupplier

Status

Current state: *Accepted*

Discussion thread: *here*

JIRA:

key	summary	type	created	updated	due	assignee	reporter	priority	status	resolution

⚠ JQL and issue key arguments for this macro require at least one Jira application link to be configured

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

Motivation

Kafka Streams was added before the Java AdminClient was introduced (KIP-117), and hence an internal StreamsKafkaClient is added for 1) managing the internal topics and 2) check compatibility between clients and brokers.

After 1.0.0 release, there are increasing number of requests to add more functionalities to the Streams client, such as purging internal data, list consumer groups etc. In addition, we also want to consolidate such duplicated code for the same functionality. So in this KIP we are proposing to add the Java AdminClient to Kafka Streams, with the goal to eventually replace StreamsKafkaClient for administrative functionalities.

Note that the scope of this KIP is only for the user facing public APIs that are needed for adding this client. We are going to replace the whole StreamsKafkaClient in steps alongside with this KIP.

Public Interfaces

Add a new function in the org.apache.kafka.streams.KafkaClientSupplier, as the following:

```
import org.apache.kafka.clients.admin.AdminClient;

public interface KafkaClientSupplier {
    // other existing APIs

    /**
     * Create a {@link AdminClient} which is used for internal topic management.
     *
     * @param config Supplied by the {@link StreamsConfig} given to the {@link KafkaStreams}
     * @return an instance of {@link AdminClient}
     */
    AdminClient getAdminClient(final Map<String, Object> config);
}
```

Proposed Changes

The newly added API allows the Streams library to create an instance of AdminClient. We are going to create one instance within each thread, and pass in that instance into the created InternalTopicManager for the leader of the group only. Here are a list of changes we are going to make:

- Purge repartition data on commit: this is summarized in ⚠ Unable to render Jira issues macro, execution error. . The AdminClient's deleteRecords API (adding in KIP-204) will be used upon committing intervals.
- Create internal topic within InternalTopicManager: we will use the create topic API of AdminClient.
- We'll also remove the endless-loop checking after the creation within StreamPartitionAssignor (⚠ Unable to render Jira issues macro, execution error.); instead after the rebalance we will let the restoration retry within the main loop if the metadata is not known yet.
- We'll add an ADMIN_PREFIX (String contant valued "admin.") in the StreamsConfig for admin client usage which will allow certain config parameters for the admin client only, also add the adminClientPrefix function; this is similar to PRODUCER_PREFIX and CONSUMER_PREFIX.
- Compatibility check: we will remove this functionality and defer to the runtime to detect the returned error code and throw exception.

In addition, the following new public APIs will be added as part of this KIP (details of these changes can be found in https://github.com/apache/kafka/pull/4315):

- Overloaded KafkaStreams constructor to allow overriding the `Time` object.
- Overloaded AbstractConfig#originalsWithPrefix() to allow specifying whether or not to strip the prefix in the key of the returned map.

Regarding the changes to replace StreamsKafkaClient with the KafkaAdminClient, the only differences are the configs. Here is a list of configs defined in StreamsConfig that will be used in StreamsKafkaClient:

- StreamsConfig.CLIENT_ID_CONFIG: still used in KafkaAdminClient.
- StreamsConfig.RETRY_BACKOFF_MS_CONFIG: still used in KafkaAdminClient.
- StreamsConfig.METADATA_MAX_AGE_CONFIG: still used in KafkaAdminClient.
- StreamsConfig.CONNECTIONS_MAX_IDLE_MS_CONFIG: still used in KafkaAdminClient.
- StreamsConfig.RECONNECT_BACKOFF_MS_CONFIG: still used in KafkaAdminClient.
- StreamsConfig.RECONNECT_BACKOFF_MAX_MS_CONFIG: still used in KafkaAdminClient.
- StreamsConfig.SEND_BUFFER_CONFIG: still used in KafkaAdminClient.
- StreamsConfig.RECEIVE_BUFFER_CONFIG: still used in KafkaAdminClient.
- StreamsConfig.REQUEST_TIMEOUT_MS_CONFIG: still used in KafkaAdminClient.
- StreamsConfig.BOOTSTRAP_SERVERS_CONFIG: still used in KafkaAdminClient.

So none of the defined properties in StreamsConfig need to be deprecated or removed. No Changes here.

Compatibility, Deprecation, and Migration Plan

- *Since we are only adding a new function in the public API, it is binary compatible though not source compatible; users are only required to make one-line change and recompile if they customize the KafkaClientSupplier.*
- *Note that the admin metrics would be a bit different:*
    - *"clientID" : [user-value] -> "[clientId]-admin"*
    - *"type": "kafka-client-metrics" -> "admin-client-metrics"*

Rejected Alternatives

None