KIP-839: Provide builders for KafkaProducer /KafkaConsumer and KafkaStreams

Status

Current state: *"Under Discussion"*

Discussion thread: https://www.mail-archive.com/dev@kafka.apache.org/msg124881.html

JIRA: KAFKA-13913

Motivation

To have more flexibility, builders should be provided for the following objects

- `KafkaAdminClient`
- KafkaProducer
- KafkaConsumer
- KafkaStreams

These builders will give an easy way to construct these objects using different arguments/combinations without having to add a new constructor every time a new parameter is required.

They will also allow using already configured dependencies coming from an injection framework such as Spring (see https://github.com/spring-projects/spring-kafka/issues/2244).

From a user point of view, builders would be used as follow

```
package org.apache.kafka.clients.admin;

KafkaAdminClient kafkaAdminClient = new KafkaAdminClientBuilder(<MAP_OR_PROPERTIES_OR_CONFIG>)
  .withMetricsReporter(<METRICS_REPORTER>)
  ...
  .build();
```

```
package org.apache.kafka.clients.producer;

KafkaProducer kafkaProducer = new KafkaProducerBuilder<String, MyPojo>(<MAP_OR_PROPERTIES_OR_CONFIG>)
  .withKeySerializer(<KEY_SERIALIZER>)
  .withValueSerializer(<VALUE_SERIALIZER>)
  .withInterceptors(<LIST_OF_INTERCEPTORS>)
  .withPartitioner(<PARTITIONER>)
  .withMetricsReporter(<METRICS_REPORTER>)
  ...
  .build();
```

```
package org.apache.kafka.clients.consumer;

KafkaConsumer consumer = new KafkaConsumerBuilder<String, MyPojo>(<MAP_OR_PROPERTIES_OR_CONFIG>)
  .withKeyDeserializer(<KEY_DESERIALIZER>)
  .withValueDeserializer(<VALUE_DESERIALIZER>)
  .withInterceptors(<LIST_OF_INTERCEPTORS>)
  .withMetricsReporter(<METRICS_REPORTER>)
  ...
  .build();
```

```
package org.apache.kafka.clients.streams;

KafkaStreams kafkaStreams = new KafkaStreamsBuilder(<TOPOLOGY>, <MAP_OR_PROPERTIES_OR_CONFIG>)
  .withProducerInterceptors(<LIST_OF_PRODUER_INTERCEPTORS>)
  .withConsumerInterceptors(<LIST_OF_CONSUMER_INTERCEPTORS>)
  .withTime(<TIME>)
  .withKafkaClientSupplier(<KAFKA_CLIENT_SUPPLIER>)
  .withMetricsReporter(<METRICS_REPORTER>)
  ...
  .build();
```

To ease the integration with existing clients, a static builder may be added on related clients.

This KIP can be seen as the continuity of the KIP-832.

Proposed Changes

Four new builders will be added

- KafkaAdminClientBuilder
- KafkaProducerBuilder
- KafkaConsumerBuilder
- KafkaStreamsBuilder

Requirements:

- The builder will always
  - override single value
  - complete the configuration parameters list values
- Any closeable objects provided to a builder will still be closed once the client is closed, and as a result, a builder should only be used once. An exception will be thrown if build() is invoked multiple times.
- Any configurable object provided to a builder will be configured by the client after instantiation.

To finalize KIP-378, two new constructors will be added in the TopologyTestDriver as follow

```
public TopologyTestDriver(Topology topology, StreamsConfig config)
public TopologyTestDriver(Topology topology, StreamsConfig config, Instant initialWallClockTime)
```

Compatibility, Deprecation, and Migration Plan

No compatibility issue and no migration plan are required as this KIP will create new builders to give more flexibility.

Kafka users will simply have more ways to build producers, consumers and a topology test driver without any impacts on the existing.

Rejected Alternatives

The KIP-832 was the original proposal but won't be implemented in benefit of the current KIP.