FIX Antenna Java Kafka Adapter quick start

Kafka Intro

Apache Kafka® (https://kafka.apache.org) is a distributed streaming platform with three key capabilities:

- Publish and subscribe to streams of records, similar to a message queue or enterprise messaging system.
- Store streams of records in a fault-tolerant durable way.
- Process streams of records as they occur.

Kafka is generally used for two broad classes of applications:

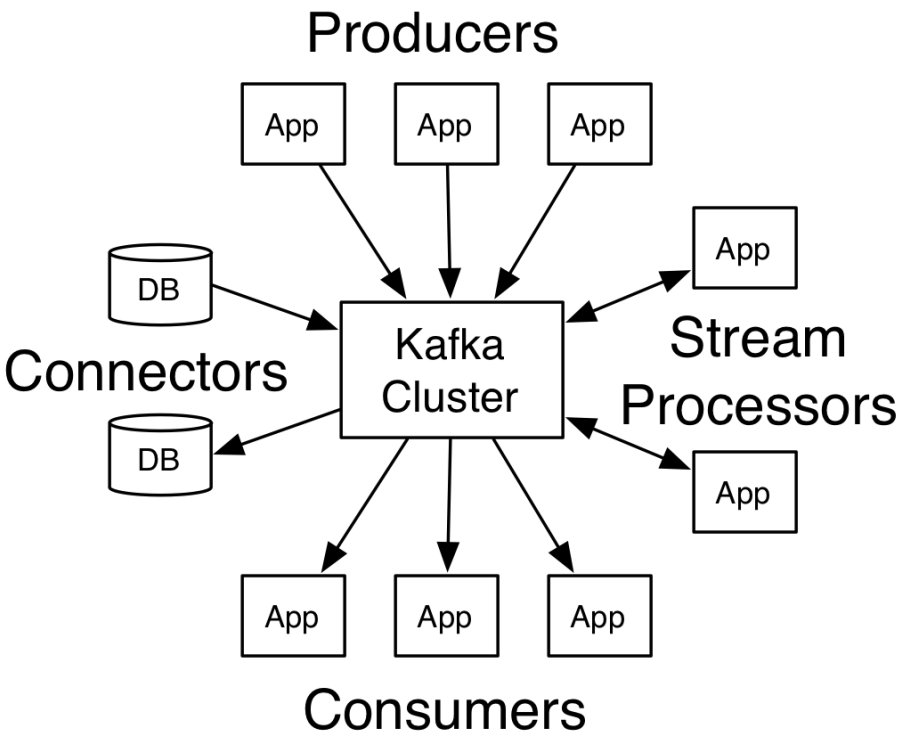
- Building real-time streaming data pipelines that reliably get data between systems or applications.
- Building real-time streaming applications that transform or react to the streams of data.

Kafka concepts:

- Kafka is run as a cluster on one or more servers that can span multiple datacenters.
- The Kafka cluster stores streams of *records* in categories called *topics.*
- Each record consists of a key, a value, and a timestamp.

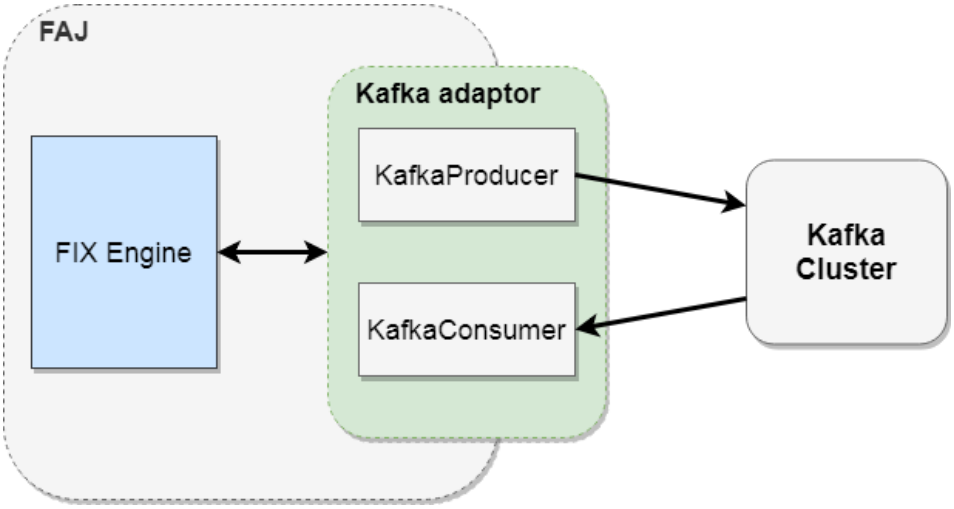
Kafka has four core APIs:

- The Producer API allows an application to publish a stream of records to one or more Kafka topics.
- The Consumer API allows an application to subscribe to one or more topics and process the stream of records produced to them.
- The Streams API allows an application to act as a *stream processor*, consuming an input stream from one or more topics and producing an output stream to one or more output topics, effectively transforming the input streams to output streams.
- The Connector API allows building and running reusable producers or consumers that connect Kafka topics to existing applications or data systems. For example, a connector to a relational database might capture every change to a table.



FAJ Kafka adapter

FAJ Kafka adapter is designed to send and receive FIX messages to or from Kafka and connect FIX sources with Kafka. It uses Kafka Producer and Consumer API to communicate with Kafka cluster.



FAJ Kafka adapter provides a few main interfaces.

Producer interface is responsible for sending FIX messages to Kafka topics and *Consumer* - for receiving. Both of them can be instantiated via a factory, which is provided with the adapter:

```
// prepare Kafka client factory
Config config = new Config("kafka", "kafka-adapter.propeties");
ClientFactory clientFactory = ClientFactory.getInstance();

// Create producer instance with id `KProducer`
Producer producer = clientFactory.createProducerClient("KProducer", outConfig);
producer.init();
producer.connect();
// send FIX message to Kafka
producer.sendMessage(fixMessage);

// Create producer instance with id `KConsumer`
Consumer consumer = clientFactory.createConsumerClient("KConsumer", inConfig);
// register a callback for received messages
consumer.setMessageListener((sessionID, message) ->
                    System.out.println("Message received by client '" + sessionID + "' - " + new String
(message)));
consuner.init();
consuner.connect();
```

The minimal kafka-adapter.propeties file should include a list of Kafka adapter clients and link them to corresponding Kafka topics:

```
# list of kafka clients (consumers and producer)
kafka.clients = KProducer, KConsumer

# main properties to specify producer's client id and topic
kafka.producer.KProducer.client.id = KProducer
kafka.producer.KProducer.topic = PTopic1

# main properties to specify consumer's client id and topic
kafka.consumer.KConsumer.client.id = KConsumer
kafka.consumer.KConsumer.topics = KTopic2
```

Each Kafka adapter client (producer or consumer) can be additionally configured with original Kafka Producer or Consumer configuration options if they are added after the client' prefix:

```
# Apply batch size for KProducer producer client only
kafka.producer.KProducer.batch.size = 16384

# Apply memory bufer size for every producer
kafka.producer.buffer.memory = 33554432
```

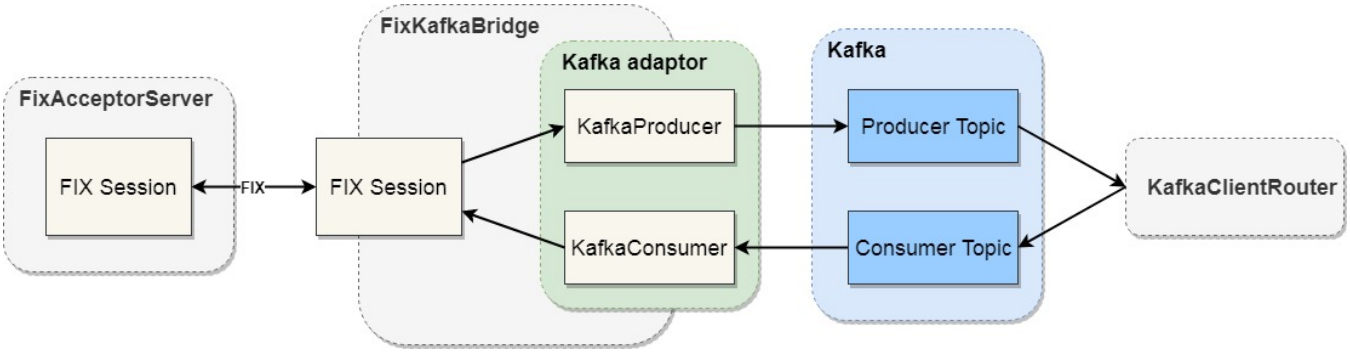
Pair of producers and consumers may be linked with FixClientAdaptor interface for easier communication management. In this case, a pair of producer and consumer may be initialized and closed at the same time:

```
Producer producer = clientFactory.createProducerClient("KProducer", outConfig);
Consumer consumer = clientFactory.createConsumerClient("KConsumer", inConfig);
FixClientAdaptor adaptor = new FixClientAdaptorImpl(consumer, producer);
adaptor.setMessageListener((sessionID, message) -> System.out.println("Message received by client '" +
sessionID + "' - " + message.toPrintableString()));
adaptor.init();
adaptor.connect();
adaptor.sendMessage(fixMsg);
```

FAJ Kafka Adapter Samples

FAJ package includes samples to demonstrate communication between FIX engine and Kafka. The quick start demo consists of a few components:

- FixAcceptorServer - simplest FIX server, which accepts incoming FIX connection, sends FIX messages to it and print all incoming FIX messages. In this demo it represents a generic external FIX message system.
- FixKafkaBridge - a sample of configurable router server, designed to transfer messages between FIX and Kafka environments. The main goal of this server is to demonstrate how FIX Engine can communicate with Kafka cluster through the FAJ Kafka adapter.
- Kafka - the instance of Kafka server.
- KafkaClientRouter - simplest Kafka client app (based on FAJ Kafka Adapter as well). Its goal - route messages back from producer topic to consumer topic and enable a roundtrip of FIX message back to the FIX server.



Steps to run FAJ Kafka adapter demo

1. Install and start Kafka:
   - Download and extract the latest version Kafka (https://kafka.apache.org/downloads)
   - Start the ZooKeeper server.
     Kafka uses ZooKeeper so you need to first start a ZooKeeper server if you don't already have one.
     You can use the convenience script packaged with Kafka to get a quick-and-dirty single-node ZooKeeper instance:
     > bin/windows/zookeeper-server-start.bat ../../config/zookeeper.properties (Windows)
     or
     > bin/zookeeper-server-start.sh ../config/zookeeper.properties (Linux)
   - Start the Kafka server:
     > bin/windows/kafka-server-start.bat ../../config/server.properties (Windows)
     or
     > bin/kafka-server-start.sh ../config/server.properties (Linux)
   - ZooKeeper/Kafka properties and performance tuning are out of the scope of this document, all further examples will be executed on default settings for Kafka.

2. Download and unpack the latest FAJ Kafka distribution package (fixaj-kafka-distribution-<version>-bin.zip).

   It will contain the following folders/parts:
   - tools - scripts, properties, and libs to start Kafka bridge;
   - examples - scripts, properties, libs and sources for FIX/Kafka samples/clients.
3. Start FIX server sample to accept FIX connection from FixKafkaBridge (server sends 10 test FIX messages to KafkaBridge after establishing FIX connection):
   > examples/bin/runFixAcceptorServer.bat (Windows)
   or
   > examples/bin/runFixAcceptorServer (Linux)
4. Start KafkaClientRouter to route messages from 'producerTopic' to 'consumerTopic' to provide messages back to KafkaBridge:
   > examples/bin/runKafkaClientRouter.bat (Windows)
   or
   > examples/bin/runKafkaClientRouter (Linux)
5. Start FixKafkaBridge with properties from "tools/etc" folder:
   > tools/bin/startKafkaBridge.bat (Windows)
   or
   > tools/bin/startKafkaBridge (Linux)

As a result of these steps, you will be able to see that:

- FIX server sends and receives FIX messages back.
- Kafka transfers messages between FixKafkaBridge and KafkaClientRouter (you can check Kafka server log or use Kafka Monitoring capabilities).