Aiven Kafka Quickstart for Java Developers

Author: https://github.com/whynotkafka
4 Jul 2022

Introduction

This quickstart tutorial will enable developers to integrate their Java applications with Kafka. By the end of this document, you will have an overview of how to produce messages, commit those messages to Kafka and deploy cloud native data infrastructure in no time at all.

Project Overview

The objective of this project is to retrieve tweets from twitter, commit them to a kafka topic and verify they have landed on the kafka service hosted the cloud data platform aiven.io.

Prerequisites

Before continuing some familiarity is expected with the following:

- Development environment
  - IDE, for example Intellij
  - Java 8 JDK
  - Git/bitbucket source control
  - Twitter Account
  - Aiven.io Account
    - https://console.aiven.io/signup

Twitter API

To retrieve tweets from twitter, please register to access the twitter development portal. You will need credentials to authenticate with the twitter API and in particular an auth bearer token. The auth bearer token is a random string that will prove your identity to twitter and this will need to be exported as a system variable before starting the application.

To learn more about the twitter API:
https://github.com/twitterdev/twitter-api-java-sdk

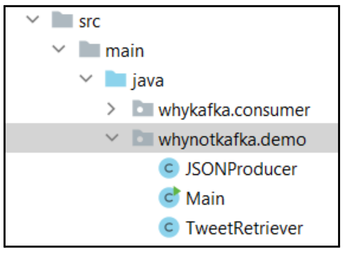
Implementation

In this section, implementation of the project is documented, this can be used as an example to incorporate in your existing applications.

1. Clone the following git repo
   ```
   cd $HOME/my_development_dir
   git clone https://github.com/whynotkafka/kafka_demo.git
   ```

2. Open Intellji and open the project in the my_development_dir/kafka_demo
   - Wait for the Maven dependencies to be retrieved

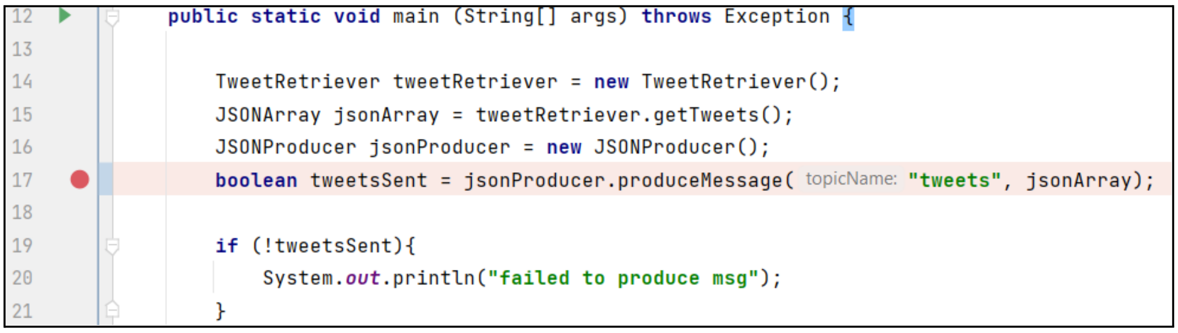
3. In the left hand side menu, navigate to whynotkafka.demo folder:

   ```
   src
     main
       java
         whykafka.consumer
         whynotkafka.demo
           JSONProducer
           Main
           TweetRetriever
   ```

   - The following example classes are provided:
     i. JSONProducer - Sends and commits our messages to Kafka in JSON format.
     ii. Main - This where the application will be started and all objects in the Java application instantiated.
     iii. TweetRetriever - Retrieves data from the Twitter API.

   Learn more about our Java examples here:
   https://help.aiven.io/en/articles/5344053-java-examples-for-testing-aiven-for-apache-kafka

4. For this exercise we will open the Main class.
   - Before we continue take note of the Kafka topicName:
   - 

     ```
     12      public static void main (String[] args) throws Exception {
     13
     14          TweetRetriever tweetRetriever = new TweetRetriever();
     15          JSONArray jsonArray = tweetRetriever.getTweets();
     16          JSONProducer jsonProducer = new JSONProducer();
     17          boolean tweetsSent = jsonProducer.produceMessage( topicName: "tweets", jsonArray);
     18
     19          if (!tweetsSent){
     20              System.out.println("failed to produce msg");
     21          }
     ```

     In the picture above, on line 17, we can see the topicName “tweets”, note this for later, it's important as this is the topic where the kafka messages will be sent.

5. Open the web browser and head to aiven.io.

6. Provision the following services:
   - Kafka
   - InfluxDB
   - Grafana
7. Setup Integrations between services, so that metrics can been synced from Kafka to InfluxDB and then displayed in Grafana.
   - https://help.aiven.io/en/articles/1456441-getting-started-with-service-integrations

8. Retrieve certificates from Kafka
   - https://developer.aiven.io/docs/products/kafka/howto/keystore-truststore

9. Set variables in the project settings. Including:
   - `TWITTER_BEARER_TOKEN`
   - `KAFKA_BOOTSTRAP_SERVERS`
   - `KAFKA_SECURITY_PROTOCOL`
   - `KAFKA_SSL_TRUSTSTORE_LOCATION`
   - `KAFKA_SSL_TRUSTSTORE_PASSWORD`
   - `KAFKA_SSL_KEYSTORE_TYPE`
   - `KAFKA_SSL_KEYSTORE_LOCATION`
   - `KAFKA_SSL_KEYSTORE_PASSWORD`
   - `KAFKA_SSL_KEY_PASSWORD`

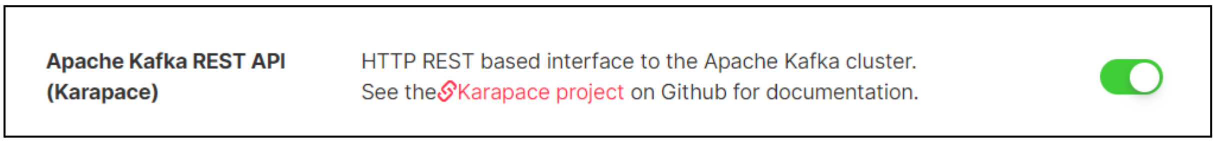
   Example here: https://www.twilio.com/blog/set-up-env-variables-intellij-idea-java

10. Head back to Intellij, go to the left hand side menu, right click the Main class and click "Run 'Main.main()'".

11. Providing the message was sent successfully, in the Run tab for intellij you will see the following message.

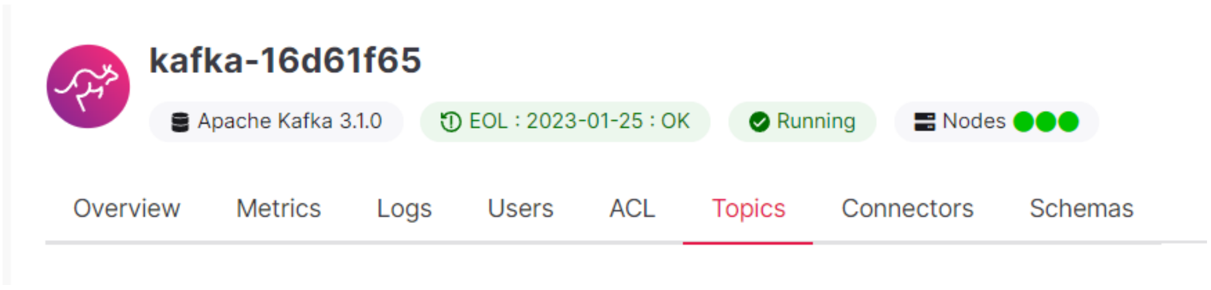
```
"Message sent successfully"
```

12. Go back to aiven.io, we will now verify the messages have landed on the tweets kafka topic. Go to the Kafka service and make sure to enable the REST API:



Apache Kafka REST API (Karapace) HTTP REST based interface to the Apache Kafka cluster. See the Karapace project on Github for documentation.

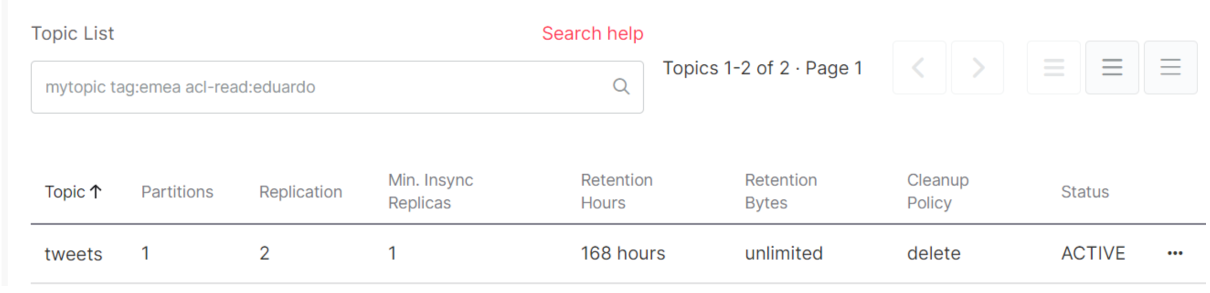
13. Now we will check the messages. Go to topics:



kafka-16d61f65

Apache Kafka 3.1.0 | EOL : 2023-01-25 : OK | Running | Nodes ●●●

14. Click the “tweets” topic:



Topic List

Search help

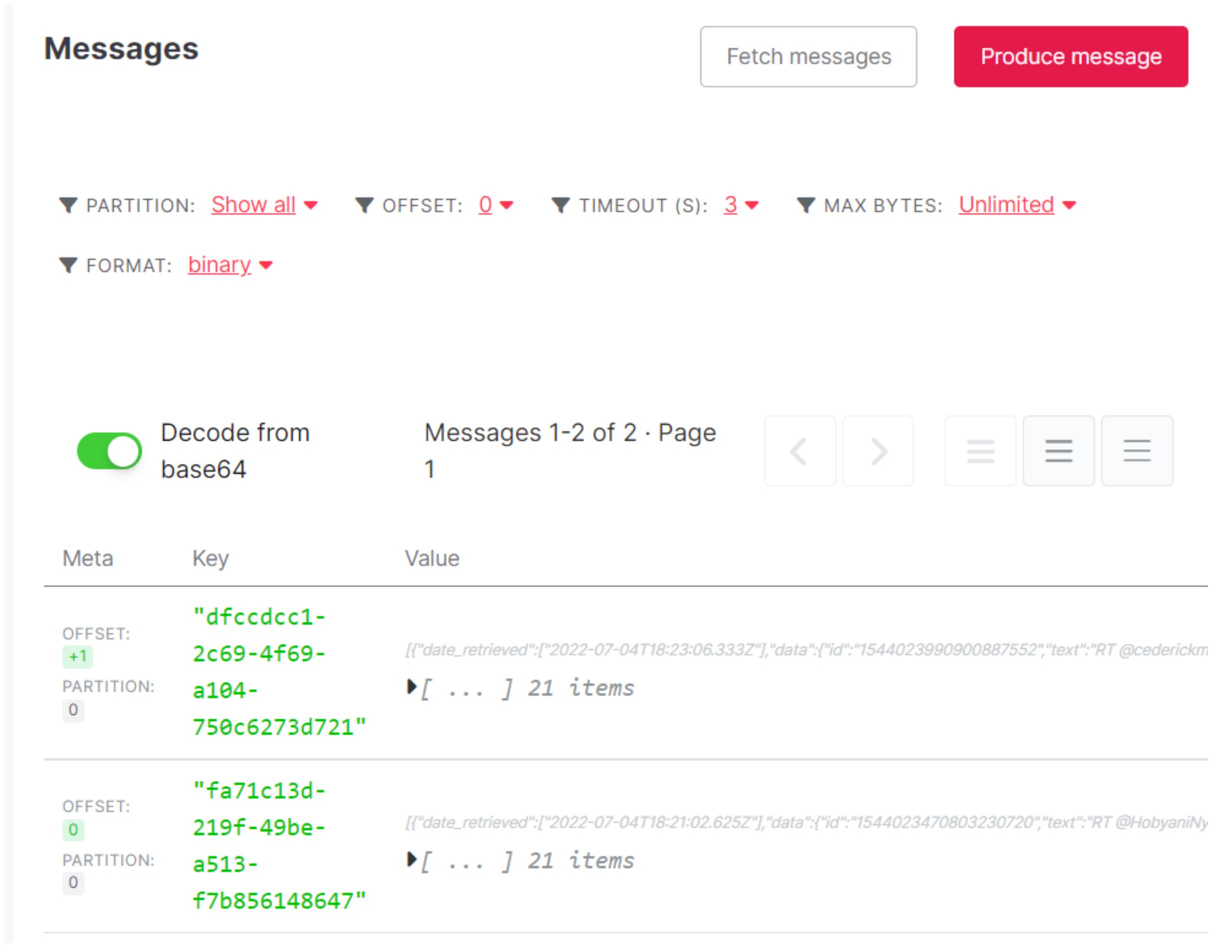
mytopic tag:emea acl-read:eduardo

Topics 1-2 of 2 · Page 1

Topic ↑	Partitions	Replication	Min. Insync Replicas	Retention Hours	Retention Bytes	Cleanup Policy	Status	
tweets	1	2	1	168 hours	unlimited	delete	ACTIVE	···

15. Click “messages”:

16. Click “Fetch messages”
17. Now we should see messages, remember to decode from base64.



Messages

PARTITION: Show all   OFFSET: 0   TIMEOUT (S): 3   MAX BYTES: Unlimited

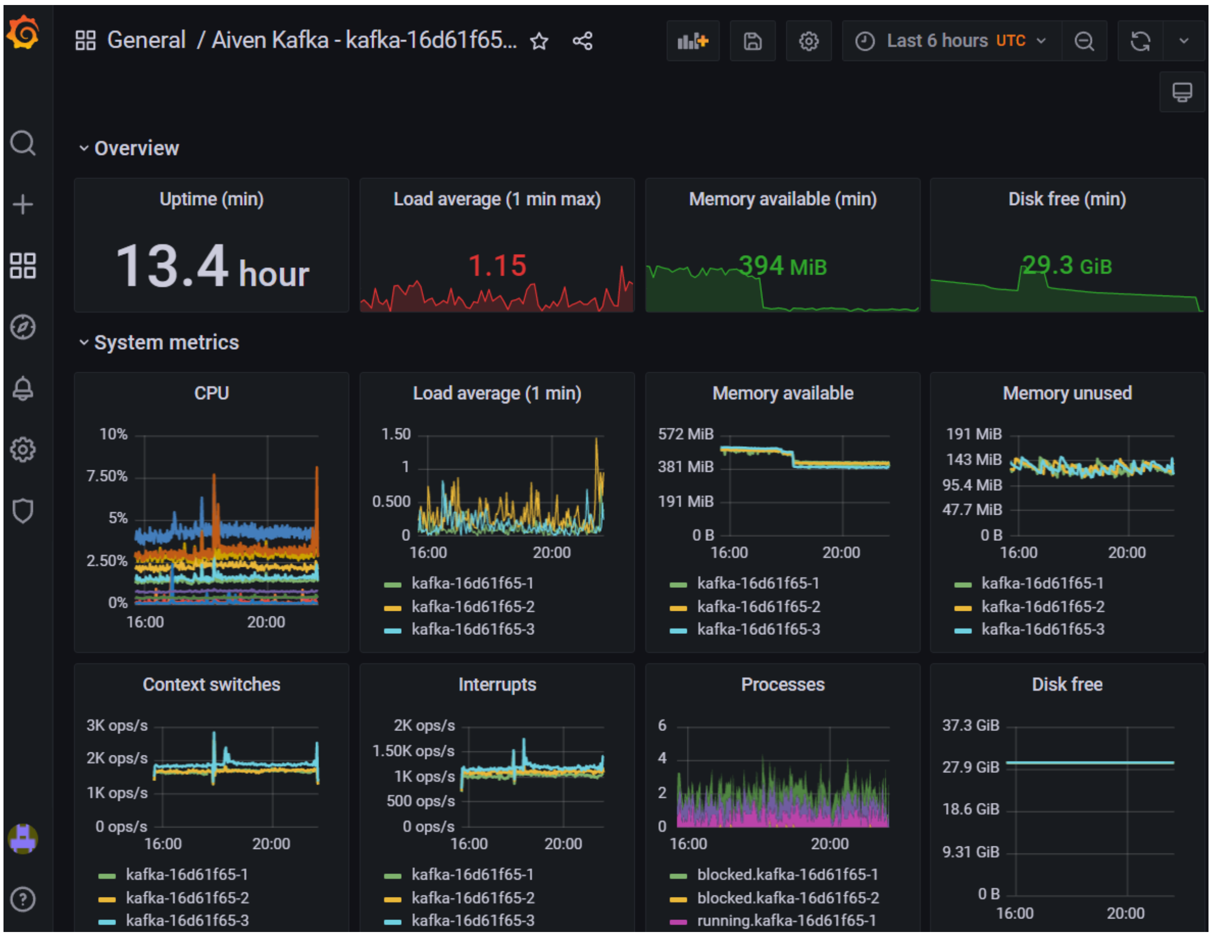
FORMAT: binary

Decode from base64

Messages 1-2 of 2 · Page 1

Meta	Key	Value
OFFSET: +1 PARTITION: 0	"dfccdcc1-2c69-4f69-a104-750c6273d721"	[{"date_retrieved":["2022-07-04T18:23:06.333Z"],"data":{"id":"1544023990900887552","text":"RT @cederickm [...] 21 items
OFFSET: 0 PARTITION: 0	"fa71c13d-219f-49be-a513-f7b856148647"	[{"date_retrieved":["2022-07-04T18:21:02.625Z"],"data":{"id":"1544023470803230720","text":"RT @HobyaniNy [...] 21 items

18. Finally, head to grafana service, lets now verify the health of the Kafka service.
    Beautiful metrics:



General / Aiven Kafka - kafka-16d61f65...
Last 6 hours UTC
Overview
Uptime (min)
13.4 hour
Load average (1 min max)
1.15
Memory available (min)
394 MiB
Disk free (min)
29.3 GiB
System metrics
CPU
Load average (1 min)
Memory available
Memory unused
Context switches
Interrupts
Processes
Disk free
kafka-16d61f65-1
kafka-16d61f65-2
kafka-16d61f65-3
blocked.kafka-16d61f65-1
blocked.kafka-16d61f65-2
running.kafka-16d61f65-1