Home assignment for (Senior) Data Engineer position

Scenario:
You have a stream of sensor data arriving in real time via a Kafka topic. Each message includes a sensor ID, a measurement value, and a timestamp in milli seconds. Your task is to create a streaming application that reads the data from Kafka, calculate average value of each sensor in each **1 minute** and publish the results in to another Kafka topic.

Sample Input Data (Kafka)

- Each incoming message to the topic of Kafka named `sensor-input` is a JSON record. Kafka producer code is attached and explained later of this document.

```
{
  "sensorId": "sensor-123",
  "value": 42.7,
  "timestamp": 1741032900000
}
```

Windowed Aggregation

- You must aggregate the data based on a 1 minute time window.

Sample Output Data (Kafka)

If you're computing **1 minute** window, a few example outputs might be:

```
{
  "sensorId": "sensor-123",
  "windowStart": 1741032900000,
  "windowEnd": 1741032960000,
  "averageValue": 45.2
}
```

```
{
  "sensorId": "sensor-123",
  "windowStart": 1741032960000,
  "windowEnd": 1741033020000,
  "averageValue": 46.7
}
```

```
{
  "sensorId": "sensor-456",
  "windowStart": 1741032900000,
  "windowEnd": 1741032960000,
  "averageValue": 39.8
}
```

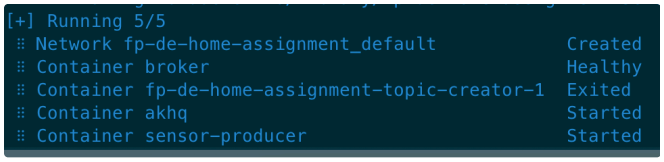
Prepare your development environment

- Install Docker in your local machine

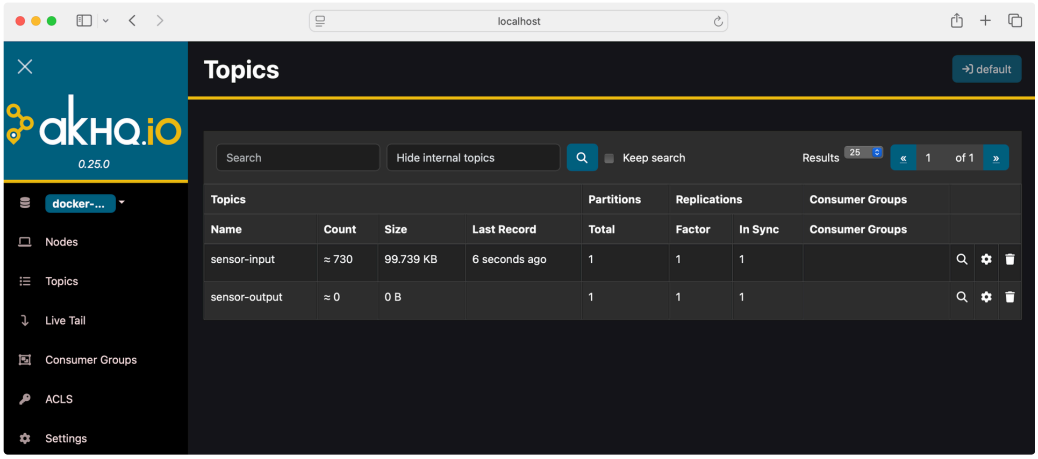
- Run Kafka and Kafka Producer
  - Unzip the attached folder `fp-de-home-assignment.zip` .
  - Run following commands in your terminal where you unzipped it.

```
# Go to the project directory
cd fp-de-home-assignment/

# Run Kafka with the `sensor-input` and `sensor-output` topics
# Run the producer to produce messages to `sensor-input`
# Run Kafka UI (http://localhost:8080/)

docker compose up --build -d
```

- Wait for 2-3 minutes until all of the docker containers are running. (see below)



- Check the Kafka UI at http://localhost:8080/ui/docker-kafka-server/topic
  - Verify `sensor-input` topic with messages from the producer (see below)



Implementation guideline

- Use python
- Use of any streaming tool (Apache Flink/ Apache Spark/ Apache Beam/ Kafka Streams/ … …) which is comfortable to you.
  - Read from `sensor-input` topic.
  - Do the aggregation by 1 minute for the sensors.
  - Write the average value of the incoming sensors to `sensor-output` topic.
- Spend not more than **4 hours.**
- Upload your solution to a public Github repository and share the link with us.

Bonus Points

- Write the result to any NoSQL database.

Happy coding 🙂