

Apache Kafka

Prerequisites:

Configure a Kafka cluster using Docker with the following parameters:

- Number of brokers - 3
- Number of partitions - 3
- Replication factor - 2
- Observe the Kafka broker logs to see how leaders/replicas for every partition are assigned

Tips

- if you're working on a machine with 8 Gb of RAM or less, you might need to fall back to just 2 brokers
- an example of a Docker Compose for a 2-node cluster based on the official Confluent Kafka image, can be found [here](#)

Practical Task:

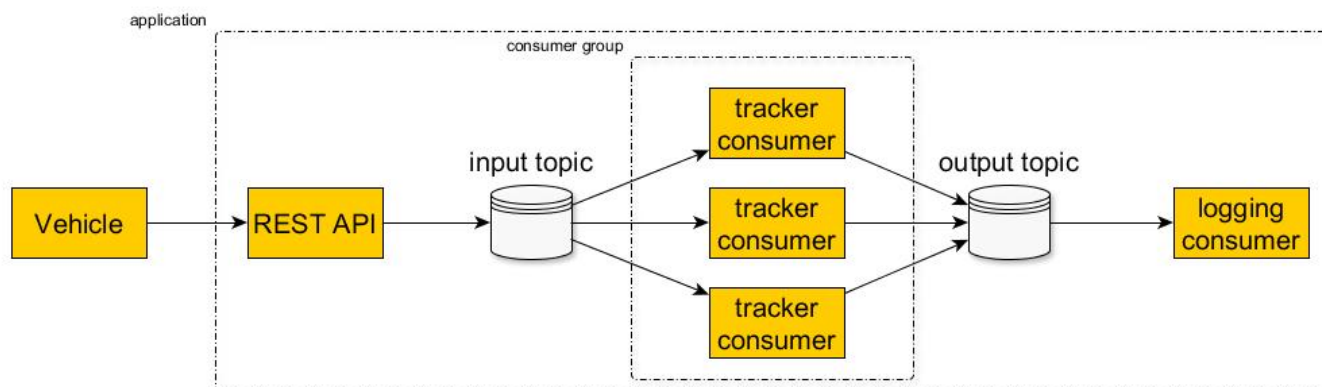
I. Implement a pair of "at least once" producer and "at most once" consumer.

1. No web application required
2. Write an integration test using the [Kafka Containers library](#)

II. Implement a taxi Spring Boot application. The application should consist of:

1. REST API which
 1. accepts vehicle signals
 2. validates that every signal carries a valid vehicle id and 2d coordinates
 3. puts the signals into the "input" Kafka topic
2. Kafka broker
3. 3 consumers which
 1. poll the signals from the "input" topic
 2. calculate the distance traveled by every unique vehicle so far
 3. store the latest distance information per vehicle in another "output" topic
4. Separate consumer that polls the "output" topic and logs the results in realtime

III. **Advanced task (optional):** Improve the taxi application by wrapping the poll(input) > calculate distance > publish(output) loop into Kafka transactions



Important

- Messages from every vehicle must be processed **sequentially!**

Tips

- the first two subtasks may be done as integration tests (for example, using the [Embedded Kafka from Spring Boot](#))

References

1. [Kafka Introduction](#)
2. [Kafka Quickstart Guide](#)
3. [Spring Kafka Introduction](#)
4. [Learn Apache Kafka for Beginners](#)