Distributed Software Systems exercise 4

Riccardo Benedetti
Leon Harz

Environmnet and Technologies

- Apache kafka was installed using a docker container, which includes a zookeeper service and the latest kafka release.
- A python virtual environment was used to implement the producer and consumers.
- A kafka topic is created used the command

  kafka-topics.sh --create --topic test-topic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1

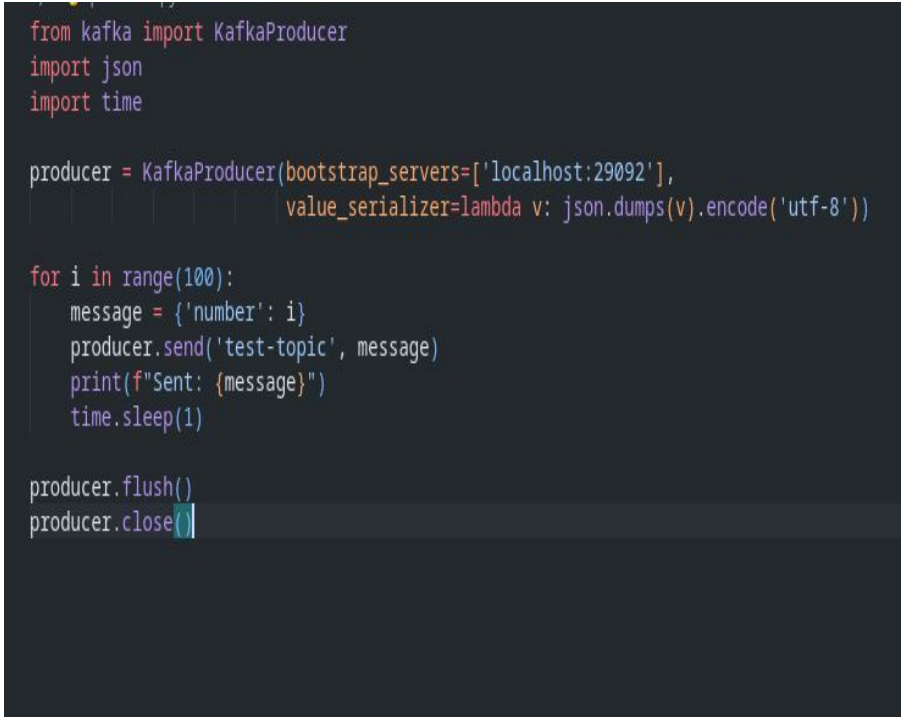
  (creates a topic named “test-topic”)

```
version: '3'
services:
  zookeeper:
    image: confluentinc/cp-zookeeper:latest
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000
    ports:
      - 22181:2181

  kafka:
    image: confluentinc/cp-kafka:latest
    depends_on:
      - zookeeper
    ports:
      - 29092:29092
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://kafka:9092,PLAINTEXT_HOST://localhost:29092
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT_HOST:PLAINTEXT
      KAFKA_INTER_BROKER_LISTENER_NAME: PLAINTEXT
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
```

producer.py

- produces and send 100 JSON messages to the “test-topic” kafka topic, using a rate of 1 message per second.



```python
from kafka import KafkaProducer
import json
import time

producer = KafkaProducer(bootstrap_servers=['localhost:29092'],
                         value_serializer=lambda v: json.dumps(v).encode('utf-8'))

for i in range(100):
    message = {'number': i}
    producer.send('test-topic', message)
    print(f"Sent: {message}")
    time.sleep(1)

producer.flush()
producer.close()
```

consumer.py

- a kafka consumer is created, with the instructions to read from the earliest message available in the topic
- value-serializer parameter deserializes from bytes to json.