

UNIT 4.2 GRADED ASSIGNMENT

Group members

Ifra Saleem (2303.khi.deg.003)

Umaina Siddiqui (2023.KHI.DEG.033)

UNIT 4.2 GRADED ASSIGNMENT

Task:

Start Kafka using docker-compose and:

1. Create a topic.
2. List Kafka topics.
3. Inspect one of them to see the number of partitions.

Solution:

1. First, I created a docker-compose.yaml file in which I have two services, one is zookeeper and another one is kafka. Both are using Confluent Platform images. This docker-compose.yaml file is used for running Apache kafka and Zookeeper. In the zookeeper service we have two environment variables ZOOKEEPER_CLIENT_PORT and ZOOKEEPER_TICK_TIME. And we are mapping container port to the host port. In the kafka service we have four environment variables.

```
1 version: '3'
2 services:
3   zookeeper:
4     image: 'confluentinc/cp-zookeeper:latest'
5     environment:
6       ZOOKEEPER_CLIENT_PORT: 2181
7       ZOOKEEPER_TICK_TIME: 2000
8     ports:
9       - '2181:2181'
10
11  kafka:
12    image: 'confluentinc/cp-kafka:latest'
13    depends_on:
14      - zookeeper
15    environment:
16      KAFKA_BROKER_ID: 1
17      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
18      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://localhost:9092
19      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
20    ports:
21      - '9093:9093'
22
23
```

2. After creating docker-compose.yaml, we need to start the containers. We can use docker-compose up -d to start the containers in the detached mode.

```
all@localhost: ~/Desktop/kafka_assignment
(base) all@localhost:~/Desktop/kafka_assignment$ docker-compose up -d
Creating network "kafka_assignment_default" with the default driver
Creating kafka_assignment_zookeeper_1 ... done
Creating kafka_assignment_kafka_1 ... done
```

3. After starting the containers, I created a `create_kafka_topic.sh` file in which I save following commands to create the kafka topics. I created two topics by running the command `./create_kafka_topic.sh`. It will check `create_kafka_topic.sh` and will execute the commands saved in that file. Replica factor is 1 which means that each partition will have one replica.

```
1 docker-compose exec kafka kafka-topics --create --topic topic1 --partitions 3 --replication-factor 1 --if-not-exists --bootstrap-server localhost:9092
2 docker-compose exec kafka kafka-topics --create --topic topic2 --partitions 4 --replication-factor 1 --if-not-exists --bootstrap-server localhost:9092
3
```

4. After creating two topics I used **`docker-compose exec kafka kafka-topics --list --bootstrap-server localhost:9092`** command to get a list of topics which I created before.
5. After this I used **`docker-compose exec kafka kafka-topics --describe --topic topic2 --bootstrap-server localhost:9092`** command to see the number of partitions in topic2.

Overall working:

```
(base) all@localhost:~/Desktop/kafka_assignment$ docker-compose up -d
Creating network "kafka_assignment_default" with the default driver
Creating kafka_assignment_zookeeper_1 ... done
Creating kafka_assignment_kafka_1 ... done
(base) all@localhost:~/Desktop/kafka_assignment$ ./create_kafka_topic.sh
Created topic topic1.
Created topic topic2.
(base) all@localhost:~/Desktop/kafka_assignment$ docker-compose exec kafka kafka-topics --list --bootstrap-server localhost:9092
topic1
topic2
(base) all@localhost:~/Desktop/kafka_assignment$ docker-compose exec kafka kafka-topics --describe --topic topic2 --bootstrap-server localhost:9092

Topic: topic2 TopicId: 5fP6ox1Q00m8g8YGagKDcw PartitionCount: 4 ReplicationFactor: 1 Configs:
Topic: topic2 Partition: 0 Leader: 1 Replicas: 1 Isr: 1
Topic: topic2 Partition: 1 Leader: 1 Replicas: 1 Isr: 1
Topic: topic2 Partition: 2 Leader: 1 Replicas: 1 Isr: 1
Topic: topic2 Partition: 3 Leader: 1 Replicas: 1 Isr: 1
(base) all@localhost:~/Desktop/kafka_assignment$
```