Justin Gnoh Kit Peow

A0202054Y

Github Repository: https://github.com/justgnoh/Task-D-CS3219

Instructions

1. Enter the working directory `cd OTOT_Task_D/`
2. Ensure Docker Desktop is running
3. Start ZooKeeper and Kafka using `docker-compose up` command
4. Observe the pulling and building of images

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker-compose up
Pulling zookeeper-1 (confluentinc/cp-zookeeper:latest)...
latest: Pulling from confluentinc/cp-zookeeper
Digest: sha256:cd05266dabba8fdfd89c4a62804e69bbf3f0c0b0d0fb2fa56b74663d892c5d0c
Status: Downloaded newer image for confluentinc/cp-zookeeper:latest
Pulling kafka-1 (confluentinc/cp-kafka:latest)...
latest: Pulling from confluentinc/cp-kafka
Digest: sha256:9b3f922f03bed5bab9cd62df8eaad7fd72d26a8b42d87bfcbde3905a4295ec25
Status: Downloaded newer image for confluentinc/cp-kafka:latest
Creating otot_task_d_zookeeper-2_1 ...
Creating otot_task_d_zookeeper-1_1 ...
Creating otot_task_d_zookeeper-3_1 ...
Creating otot_task_d_zookeeper-1_1 ... done
Creating otot_task_d_zookeeper-3_1 ... done
Creating otot_task_d_zookeeper-2_1 ... done
Creating otot_task_d_kafka-2_1     ...
Creating otot_task_d_kafka-3_1     ...
Creating otot_task_d_kafka-1_1     ...
Creating otot_task_d_kafka-2_1     ... done
Creating otot_task_d_kafka-3_1     ... done
Creating otot_task_d_kafka-1_1     ... done
Attaching to otot_task_d_zookeeper-1_1, otot_task_d_zookeeper-3_1, otot_task_d_z
```

5. Open another terminal to observe the status of services
6. Enter `docker-compose ps`

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker-compose ps
          Name                        Command            State   Ports
-------------------------------------------------------------------------
otot_task_d_kafka-1_1       /etc/confluent/docker/run   Up
otot_task_d_kafka-2_1       /etc/confluent/docker/run   Up
otot_task_d_kafka-3_1       /etc/confluent/docker/run   Up
otot_task_d_zookeeper-1_1   /etc/confluent/docker/run   Up
otot_task_d_zookeeper-2_1   /etc/confluent/docker/run   Up
otot_task_d_zookeeper-3_1   /etc/confluent/docker/run   Up
```

7. Observe above a kafka cluster and its zookeepers
8. Check the ZooKeeper logs to verify that ZooKeeper is healthy

a. Run **`docker-compose logs zookeeper-1 | findstr "FOLLOWING Created`**

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker-compose logs zookeeper-1 | findstr "FOLLOWING Created"
zookeeper-1_1  | [2021-11-12 04:06:51,230] INFO FOLLOWING (org.apache.zookeeper.
server.quorum.QuorumPeer)
zookeeper-1_1  | [2021-11-12 04:06:51,261] INFO Created server with tickTime 200
0 minSessionTimeout 4000 maxSessionTimeout 40000 clientPortListenBacklog -1 data
dir /var/lib/zookeeper/log/version-2 snapdir /var/lib/zookeeper/data/version-2 (
org.apache.zookeeper.server.ZooKeeperServer)
zookeeper-1_1  | [2021-11-12 04:06:51,264] INFO FOLLOWING - LEADER ELECTION TOOK
 - 1416 MS (org.apache.zookeeper.server.quorum.Learner)
```

b. Run **`docker-compose logs zookeeper-2 | findstr "FOLLOWING Created`**

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker-compose logs zookeeper-2 | findstr "FOLLOWING Created"
zookeeper-2_1  | [2021-11-12 04:06:51,232] INFO FOLLOWING (org.apache.zookeeper.serve
r.quorum.QuorumPeer)
zookeeper-2_1  | [2021-11-12 04:06:51,246] INFO Created server with tickTime 2000 min
SessionTimeout 4000 maxSessionTimeout 40000 clientPortListenBacklog -1 datadir /var/l
ib/zookeeper/log/version-2 snapdir /var/lib/zookeeper/data/version-2 (org.apache.zook
eeper.server.ZooKeeperServer)
zookeeper-2_1  | [2021-11-12 04:06:51,248] INFO FOLLOWING - LEADER ELECTION TOOK - 43
4 MS (org.apache.zookeeper.server.quorum.Learner)
```

c. Run **`docker-compose logs zookeeper-3 | findstr "FOLLOWING Created`**

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker-compose logs zookeeper-3 | findstr "FOLLOWING Created"
zookeeper-3_1  | [2021-11-12 04:06:51,258] INFO Created server with tickTime 2000 minSessionTi
meout 4000 maxSessionTimeout 40000 clientPortListenBacklog -1 datadir /var/lib/zookeeper/log/v
ersion-2 snapdir /var/lib/zookeeper/data/version-2 (org.apache.zookeeper.server.ZooKeeperServe
r)
```

9. Next, we will verify that the Kafka broker has booted up successfully
   a. Run **`docker-compose logs kafka-1 | findstr started`**

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker-compose logs kafka-1 | findstr started
kafka-1_1      | [2021-11-12 04:07:04,323] INFO [KafkaServer id=1] started (kafka.server.Kafka
Server)
```

   b. Run **`docker-compose logs kafka-2 | findstr started`**

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker-compose logs kafka-2 | findstr started
kafka-2_1      | [2021-11-12 04:07:03,889] INFO [KafkaServer id=2] started (kafka.server.Kafka
Server)
```

   c. Run **`docker-compose logs kafka-3 | findstr started`**

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker-compose logs kafka-3 | findstr started
kafka-3_1      | [2021-11-12 04:07:05,208] INFO [KafkaServer id=3] started (kafka.server.Kafka
Server)
```

10. Next, we will test if the brokers are working as expected by creating a topic
11. Enter ` **docker run --net=host --rm confluentinc/cp-kafka:5.0.0 kafka-topics --create --topic bar --partitions 3 --replication-factor 3 --if-not-exists --zookeeper localhost:32181**`
    a. If you do not 'confluentinc/cp-kafka:5.0.0' locally, it will pull the necessary images.
    b. Verify the topic **`bar`** has been created

```
Digest: sha256:9bc4b7a7234338cae2eaf5da6b2e563668f6ca6cd3544bde009029517035168b
Status: Downloaded newer image for confluentinc/cp-kafka:5.0.0
Created topic "bar".
```

12. Verify topic `**bar**` by entering:
    ` docker run --net=host --rm confluentinc/cp-kafka:5.0.0 kafka-topics --describe --topic bar --zookeeper localhost:32181`

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker run --net=host --rm confluentinc/cp-kafka:5.0.0 kafka-topics --describe --topic bar -
-zookeeper localhost:32181
Topic:bar       PartitionCount:3        ReplicationFactor:3     Configs:
        Topic: bar      Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: bar      Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: bar      Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
```

    a. Observe in `Replicas` the nodes that will take over the Master node (Leader) in the event the Master node (Leader) fails

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker run --net=host --rm confluentinc/cp-kafka:5.0.0 kafka-topics --describe --topic bar -
-zookeeper localhost:32181
Topic:bar       PartitionCount:3        ReplicationFactor:3     Configs:
        Topic: bar      Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: bar      Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: bar      Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
```

13. Generate data to topic `**bar**` by entering:
    `docker run --net=host --rm confluentinc/cp-kafka:5.0.0 bash -c "seq 33 | kafka-console-producer --broker-list localhost:29092 --topic bar && echo 'Produced 33 messages.'"`

14. Observe the production of 33 messages

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker run --net=host --rm confluentinc/cp-kafka:5.0.0 bash -c "seq 33 | kafka-console-produ
cer --broker-list localhost:29092 --topic bar && echo 'Produced 33 messages.'"
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Produced 33 messages.
```

15. Verify that the consumer receives all generated messages. This can be done by outputting the messages using the Console Consumer. Enter the following command:
    ` docker run --net=host --rm confluentinc/cp-kafka:5.0.0 kafka-console-consumer --bootstrap-server localhost:29092 --topic bar --from-beginning --max-messages 33`

```
Justin@JustinsWorkstation MINGW64 ~/Desktop/New folder (2)/OTOT_Task_D (main)
$ docker run --net=host --rm confluentinc/cp-kafka:5.0.0 kafka-console-consumer --bootstrap-se
rver localhost:29092 --topic bar --from-beginning --max-messages 33
1
4
7
10
13
16
19
22
25
28
31
2
5
8
11
14
17
20
23
26
29
32
3
6
9
12
15
18
21
24
27
30
33
Processed a total of 33 messages
```

Observing the image above, all 33 messages generated has been processed, showing that it is successful.

Documenting Initial Steps:

Using the docker-compose.yml from: https://github.com/confluentinc/cp-docker-images/blob/5.3.3-post/examples/kafka-cluster/docker-compose.yml