

Name: Chen Hsiao Ting

Matriculation Number: A0222182R

Link to GitHub repository: <https://github.com/hsiaotingluv/CS3219-OTOT-TaskD>

Instructions on how to run Pub-Sub Messaging

Task D.1: Successful implementation of Pub-Sub messaging system using Apache Kafka

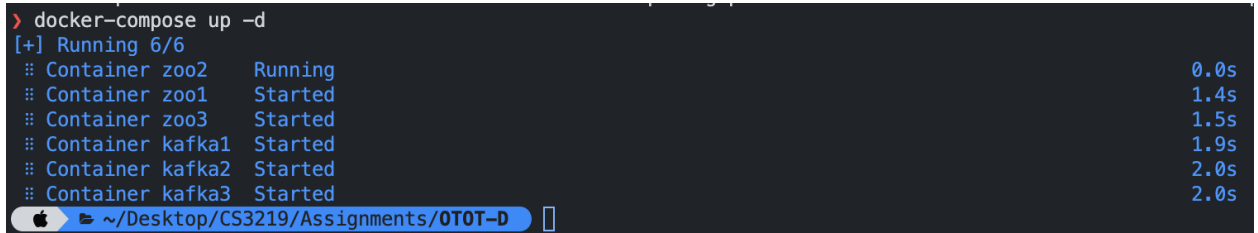
1. Download using Homebrew

- brew install kafka
- brew install zookeeper
- zkServer start
- kafka-server-start /usr/local/etc/kafka/server.properties

2. Run Docker Containers

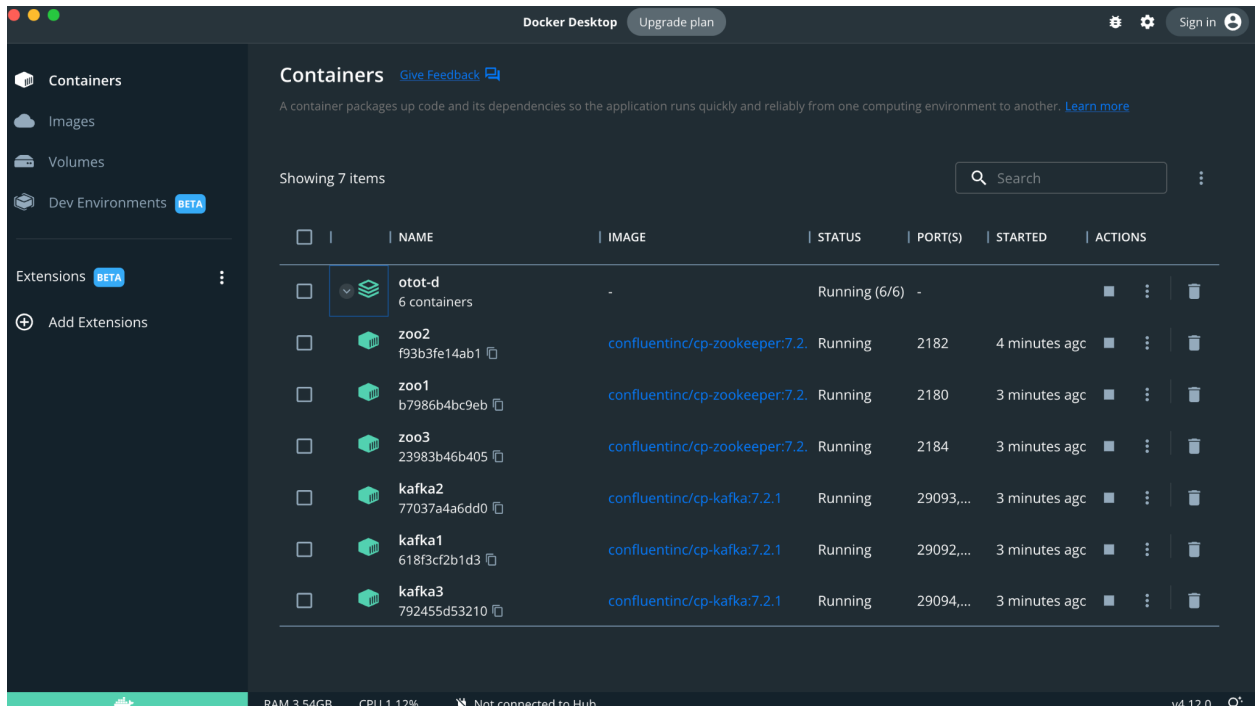
- run `docker-compose.yml` from my GitHub repo to start 3 zookeeper and 3 kafka nodes

```
> docker-compose up -d
[+] Running 6/6
  :: Container zoo2    Running           0.0s
  :: Container zoo1    Started           1.4s
  :: Container zoo3    Started           1.5s
  :: Container kafka1  Started           1.9s
  :: Container kafka2  Started           2.0s
  :: Container kafka3  Started           2.0s
```

A terminal window screenshot showing the execution of 'docker-compose up -d'. The output shows six containers being started: zoo2 (Running, 0.0s), zoo1 (Started, 1.4s), zoo3 (Started, 1.5s), kafka1 (Started, 1.9s), kafka2 (Started, 2.0s), and kafka3 (Started, 2.0s). The terminal has a dark background with light blue and white text. The prompt is a red '>' character. The file path at the bottom is '~/Desktop/CS3219/Assignments/OTOT-D'.

3. Verify Docker is running

- using Docker Desktop



- run `docker ps` on terminal

```
> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
792455d53210	confluentinc/cp-kafka:7.2.1	"/etc/confluent/dock..."	3 minutes ago	Up 3 minutes	0.0.0.0:9094->9094/tcp, 9092/tcp, 0.0.0.0:29094->29094/tcp	kafka3
618f3cf2b1d3	confluentinc/cp-kafka:7.2.1	"/etc/confluent/dock..."	3 minutes ago	Up 3 minutes	0.0.0.0:9092->9092/tcp, 0.0.0.0:29092->29092/tcp	kafka1
77837a4a6dd0	confluentinc/cp-kafka:7.2.1	"/etc/confluent/dock..."	3 minutes ago	Up 3 minutes	0.0.0.0:9093->9093/tcp, 9092/tcp, 0.0.0.0:29093->29093/tcp	kafka2
23983b46b405	confluentinc/cp-zookeeper:7.2.1	"/etc/confluent/dock..."	3 minutes ago	Up 3 minutes	2181/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2184->2184/tcp	zoo3
b7986b4bc9eb	confluentinc/cp-zookeeper:7.2.1	"/etc/confluent/dock..."	3 minutes ago	Up 3 minutes	2181/tcp, 2888/tcp, 0.0.0.0:2180->2180/tcp, 3888/tcp	zoo1
f93b3fe14ab1	confluentinc/cp-zookeeper:7.2.1	"/etc/confluent/dock..."	4 minutes ago	Up 4 minutes	2181/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2182->2182/tcp	zoo2

4. Create a topic

- run `kafka-topics --create --bootstrap-server localhost:9092 --replication-factor 3 --partitions 1 --topic otot_task_d`

```
> kafka-topics --create --bootstrap-server localhost:9092 --replication-factor 3 --partitions 1 --topic otot_task_d
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
Created topic otot_task_d.
```

5. Create a producer

- run `kafka-console-producer --bootstrap-server localhost:9092 --topic otot_task_d`

```
> kafka-console-producer --bootstrap-server localhost:9092 --topic otot_task_d
>
```

6. Create a consumer

- run `kafka-console-consumer --bootstrap-server localhost:9092 --topic otot_task_d`

```
> kafka-console-consumer --bootstrap-server localhost:9092 --topic otot_task_d
```

7. Testing Pub-Sub Messaging

- in the producer shell, type a message "hello world"

```
> kafka-console-producer --bootstrap-server localhost:9092 --topic otot_task_d  
>hello world  
>
```

- navigate to the consumer shell, we see the "hello world" message, pub-sub messaging is successfully implemented!

```
> kafka-console-consumer --bootstrap-server localhost:9092 --topic otot_task_d  
hello world
```

Task D.2: Successful management of the failure of the master node in the cluster, i.e. another node takes over as master node

1. Check master node

- run `kafka-topics --bootstrap-server localhost:9092 --topic otot_task_d -describe``

```
> kafka-topics --bootstrap-server localhost:9092 --topic otot_task_d -describe
Topic: otot_task_d      TopicId: Ajv_J-PrTxCV7TRECCBRfw PartitionCount: 1      ReplicationFactor: 3      Configs:
                        Topic: otot_task_d      Partition: 0      Leader: 3      Replicas: 3,1,2 Isr: 3,1,2
```

- we see that the leader is kafka3

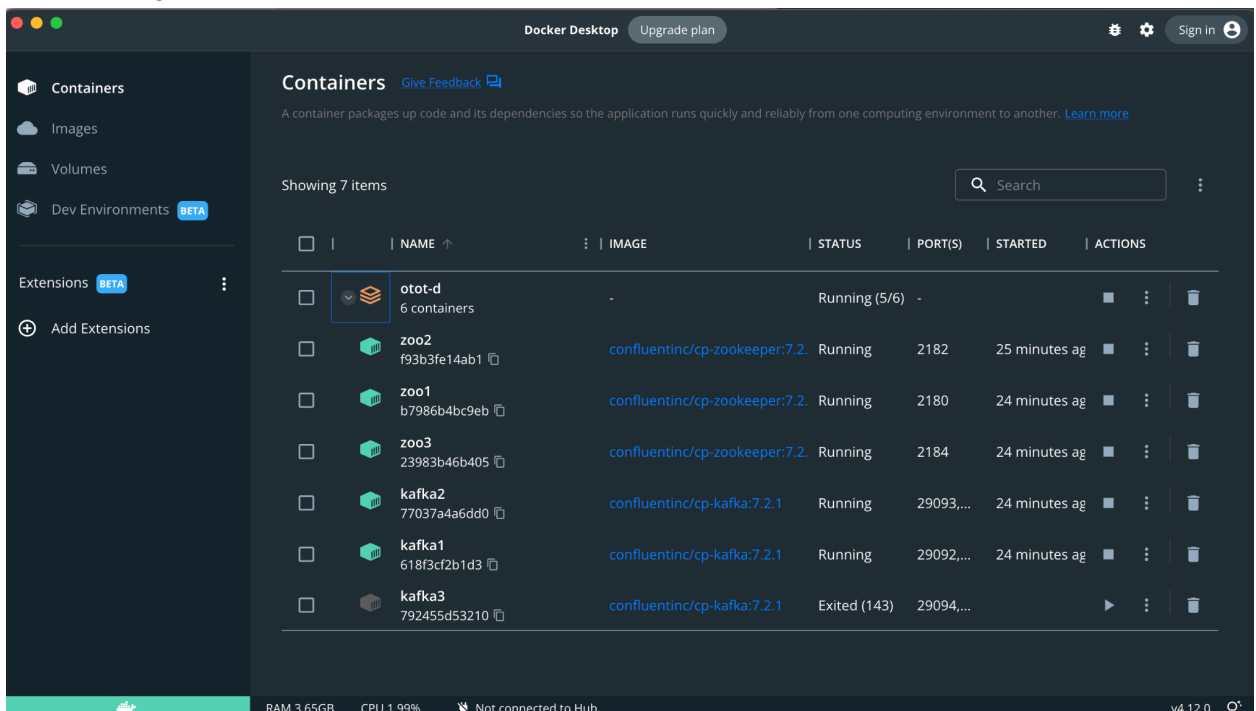
2. Stop master node

- run `docker stop kafka3``

```
> docker stop kafka3
kafka3
```

3. Verify master node is stopped

- using Docker Desktop



- using `docker ps``

```
> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                                                                                     NAMES
618f3cf2b1d3   confluentinc/cp-kafka:7.2.1         "/etc/confluent/dock..." 24 minutes ago Up 24 minutes 0.0.0.0:9092->9092/tcp, 0.0.0.0:29092->29092/tcp               kafka1
77037a4a6dd0   confluentinc/cp-kafka:7.2.1         "/etc/confluent/dock..." 24 minutes ago Up 24 minutes 0.0.0.0:9093->9093/tcp, 9092/tcp, 0.0.0.0:29093->29093/tcp       kafka2
23983b46b405   confluentinc/cp-zookeeper:7.2.1     "/etc/confluent/dock..." 24 minutes ago Up 24 minutes 2181/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2184->2184/tcp             zoo3
b7986b4bc9eb   confluentinc/cp-zookeeper:7.2.1     "/etc/confluent/dock..." 24 minutes ago Up 24 minutes 2181/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2180->2180/tcp, 3888/tcp  zoo1
f93b3fe14ab1   confluentinc/cp-zookeeper:7.2.1     "/etc/confluent/dock..." 25 minutes ago Up 25 minutes 2181/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2182->2182/tcp             zoo2
```

4. Check new master node

- run `kafka-topics --bootstrap-server localhost:9092 --topic otot_task_d -describe``

```
> kafka-topics --bootstrap-server localhost:9092 --topic otot_task_d -describe
Topic: otot_task_d      TopicId: Ajv_J-PrTxCV7TREcCBRfw PartitionCount: 1      ReplicationFactor: 3   Configs:
                        Topic: otot_task_d      Partition: 0      Leader: 1      Replicas: 3,1,2 Isr: 1,2
```

- we see that the new leader is kafka1

5. Testing Pub-Sub Messaging

- in the producer shell, type a message “test 2 after failure of previous master node!”

```
> kafka-console-producer --bootstrap-server localhost:9092 --topic otot_task_d
>hello world
>test 2 after failure of previous master node!
>
```

- navigate to the consumer shell, we see the “test 2 after failure of previous master node!” message, pub-sub messaging failure is successfully managed!

```
> kafka-console-consumer --bootstrap-server localhost:9092 --topic otot_task_d
hello world
test 2 after failure of previous master node!
```