

## CS3219 OTOT Assignment Task D

**Student Name:** Seah Zhi Xuan

**Matriculation Number:** A0227546Y

Link to github repo: <https://github.com/joszx/OTOT-D>

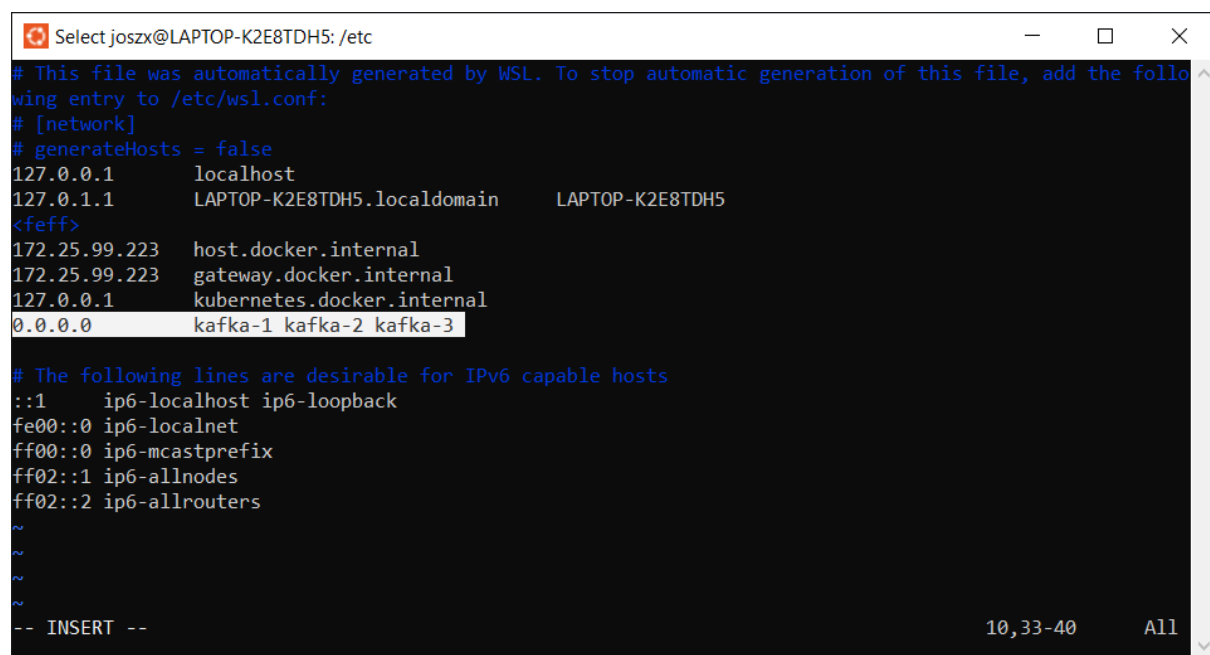
Prerequisites:

Have docker, docker-compose and kafkacat installed.

Linux system. For this I used Ubuntu.

1. Add the following line to /etc/hosts on linux using “sudo vim hosts”:

```
0.0.0.0 kafka-1 kafka-2 kafka-3
```



```
Select joszx@LAPTOP-K2E8TDH5: /etc
# This file was automatically generated by WSL. To stop automatic generation of this file, add the following entry to /etc/wsl.conf:
# [network]
# generateHosts = false
127.0.0.1    localhost
127.0.1.1    LAPTOP-K2E8TDH5.localdomain    LAPTOP-K2E8TDH5
<feff>
172.25.99.223 host.docker.internal
172.25.99.223 gateway.docker.internal
127.0.0.1    kubernetes.docker.internal
0.0.0.0      kafka-1 kafka-2 kafka-3

# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
~
~
~
-- INSERT --                               10,33-40    All
```

2. Cd to repository and create a “docker-compose.yml” file with the following lines:

```
version: '2'
services:
  zookeeper-1:
    image: confluentinc/cp-zookeeper:latest
    hostname: zookeeper-1
    ports:
      - "12181:12181"
    environment:
      ZOOKEEPER_SERVER_ID: 1
```

```
ZOOKEEPER_CLIENT_PORT: 12181
ZOOKEEPER_TICK_TIME: 2000
ZOOKEEPER_INIT_LIMIT: 5
ZOOKEEPER_SYNC_LIMIT: 2
ZOOKEEPER_SERVERS: zookeeper-1:12888:13888;zookeeper-2:22888:23888;zookeeper-
3:32888:33888

zookeeper-2:
  image: confluentinc/cp-zookeeper:latest
  hostname: zookeeper-2
  ports:
    - "22181:22181"
  environment:
    ZOOKEEPER_SERVER_ID: 2
    ZOOKEEPER_CLIENT_PORT: 22181
    ZOOKEEPER_TICK_TIME: 2000
    ZOOKEEPER_INIT_LIMIT: 5
    ZOOKEEPER_SYNC_LIMIT: 2
    ZOOKEEPER_SERVERS: zookeeper-1:12888:13888;zookeeper-2:22888:23888;zookeeper-
3:32888:33888

zookeeper-3:
  image: confluentinc/cp-zookeeper:latest
  hostname: zookeeper-3
  ports:
    - "32181:32181"
  environment:
    ZOOKEEPER_SERVER_ID: 3
    ZOOKEEPER_CLIENT_PORT: 32181
    ZOOKEEPER_TICK_TIME: 2000
    ZOOKEEPER_INIT_LIMIT: 5
    ZOOKEEPER_SYNC_LIMIT: 2
    ZOOKEEPER_SERVERS: zookeeper-1:12888:13888;zookeeper-2:22888:23888;zookeeper-
3:32888:33888

kafka-1:
  image: confluentinc/cp-kafka:latest
  hostname: kafka-1
  ports:
    - "19092:19092"
  depends_on:
    - zookeeper-1
    - zookeeper-2
    - zookeeper-3
  environment:
    KAFKA_BROKER_ID: 1
    KAFKA_ZOOKEEPER_CONNECT: zookeeper-1:12181,zookeeper-2:12181,zookeeper-3:12181
    KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://kafka-1:19092

kafka-2:
  image: confluentinc/cp-kafka:latest
```

```
hostname: kafka-2
ports:
  - "29092:29092"
depends_on:
  - zookeeper-1
  - zookeeper-2
  - zookeeper-3
environment:
  KAFKA_BROKER_ID: 2
  KAFKA_ZOOKEEPER_CONNECT: zookeeper-1:12181,zookeeper-2:12181,zookeeper-3:12181
  KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://kafka-2:29092
```

```
kafka-3:
  image: confluentinc/cp-kafka:latest
  hostname: kafka-3
  ports:
    - "39092:39092"
  depends_on:
    - zookeeper-1
    - zookeeper-2
    - zookeeper-3
  environment:
    KAFKA_BROKER_ID: 3
    KAFKA_ZOOKEEPER_CONNECT: zookeeper-1:12181,zookeeper-2:12181,zookeeper-3:12181
    KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://kafka-3:39092
```

3. Run “docker-compose.yml” with the following command:

```
docker-compose -f docker-compose.yml up -d
```

4. Check docker containers are running with the following command:

```
docker-compose container ls
```

```
9642x0LAPTOP-K26ST0M5:/mnt/c/Users/ZK/Documents/VScode_projects/OTOT-D$ docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                                                                               NAMES
3a792fdbfb1a   confluentinc/cp-kafka:latest        "/etc/confluent/dock..." 2 minutes ago  Up 2 minutes  9092/tcp, 0.0.0.0:20092->20092/tcp                       otot-d-kafka-1-1
43b534f76918   confluentinc/cp-kafka:latest        "/etc/confluent/dock..." 2 minutes ago  Up 2 minutes  9092/tcp, 0.0.0.0:40092->40092/tcp                       otot-d-kafka-3-1
c5b4d679a61e   confluentinc/cp-kafka:latest        "/etc/confluent/dock..." 2 minutes ago  Up 2 minutes  9092/tcp, 0.0.0.0:30092->30092/tcp                       otot-d-kafka-2-1
9462840a3087   confluentinc/cp-zookeeper:latest    "/etc/confluent/dock..." 2 minutes ago  Up 2 minutes  2181/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:13181->13181/tcp    otot-d-zookeeper-1-1
c68297d08fc7   confluentinc/cp-zookeeper:latest    "/etc/confluent/dock..." 2 minutes ago  Up 2 minutes  2181/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:33181->33181/tcp    otot-d-zookeeper-3-1
8eb585da347d   confluentinc/cp-zookeeper:latest    "/etc/confluent/dock..." 2 minutes ago  Up 2 minutes  2181/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:23181->23181/tcp    otot-d-zookeeper-2-1
```

5. Check which broker is the controller with the following command:

```
kafkacat -L -b kafka-1:20092
```

```
jsozxx@LAPTOP-K2E8TDH5:/mnt/c/Users/ZX/Documents/VScode projects/OTOT-D$ kafkacat -L -b kafka-1:20092
Metadata for all topics (from broker 1: kafka-1:20092/1):
3 brokers:
  broker 2 at kafka-2:30092
  broker 3 at kafka-3:40092 (controller)
  broker 1 at kafka-1:20092
2 topics:
  topic "test_topic" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "topic1" with 1 partitions:
    partition 0, leader 3, replicas: 3, isrs: 3
```

6. To test that the Pub-Sub messaging system is working, open 2 other terminals.

7. In the first terminal, set it as the producer with the following command:

```
kafkacat -P -b kafka-2:30092 -t topic1
```

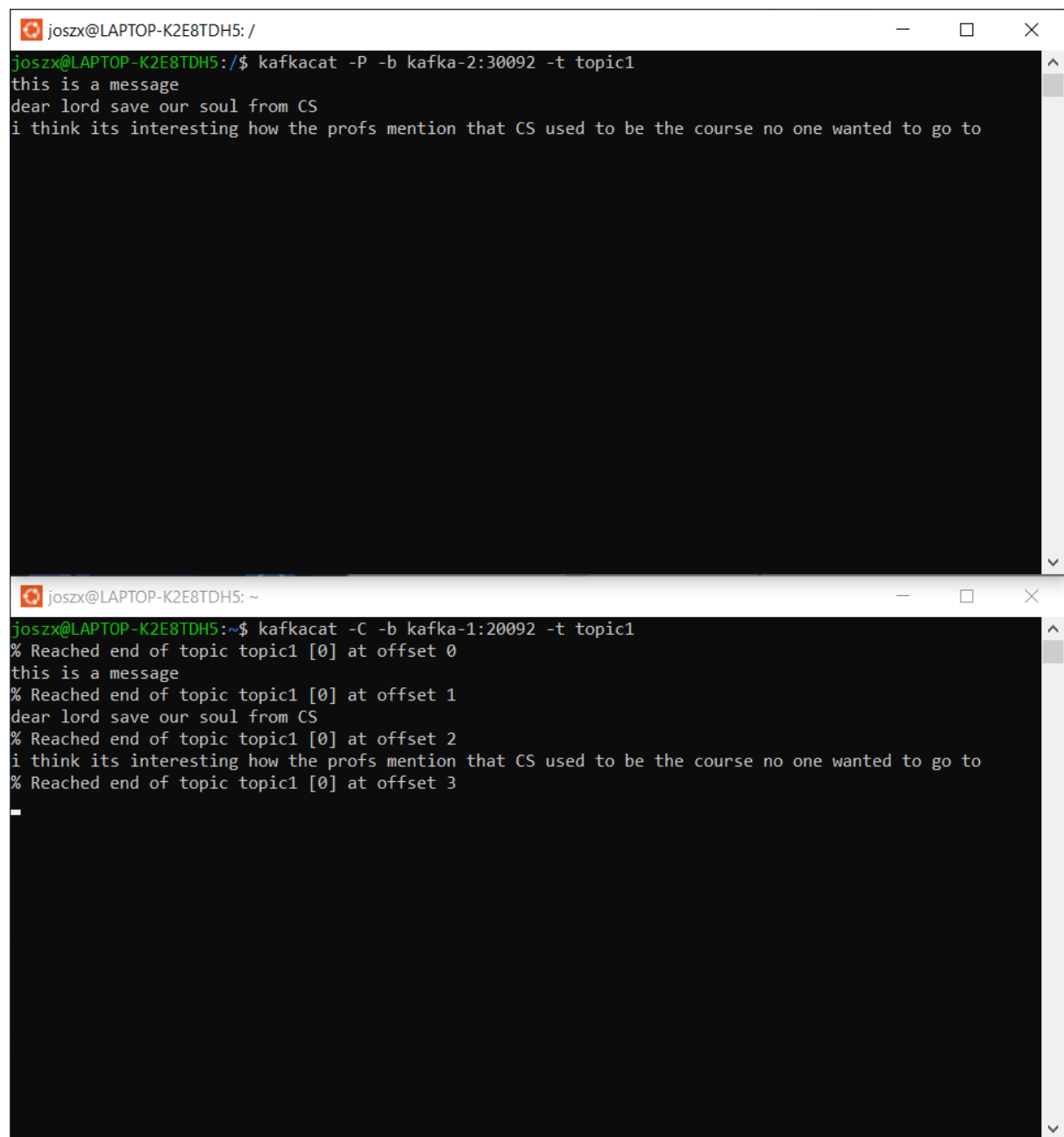
8. In the second terminal, set it as the consumer with the following command:

```
kafkacat -C -b kafka-1:20092 -t topic1
```

9. Test Pub-Sub messaging system is working by typing into the producer and pressing enter like in the pictures below.

```
josex@LAPTOP-K2E8TDH5: /  
josex@LAPTOP-K2E8TDH5:/$ kafkacat -P -b kafka-2:30092 -t topic1  
  
josex@LAPTOP-K2E8TDH5: ~  
josex@LAPTOP-K2E8TDH5:~$ kafkacat -C -b kafka-1:20092 -t topic1  
% Reached end of topic topic1 [0] at offset 0
```

Pub-sub working:



The image shows two terminal windows from a laptop named LAPTOP-K2E8TDH5. The top window shows a Kafka consumer (kafkacat -P) reading messages from a topic named 'topic1' at broker 'kafka-2:30092'. It displays three messages: 'this is a message', 'dear lord save our soul from CS', and 'i think its interesting how the profs mention that CS used to be the course no one wanted to go to'. The bottom window shows a Kafka producer (kafkacat -C) writing the same three messages to 'topic1' at broker 'kafka-1:20092'. It shows the messages being sent at offsets 0, 1, 2, and 3, with the producer reaching the end of the topic after each message.

```
joszx@LAPTOP-K2E8TDH5: /  
joszx@LAPTOP-K2E8TDH5:/$ kafkacat -P -b kafka-2:30092 -t topic1  
this is a message  
dear lord save our soul from CS  
i think its interesting how the profs mention that CS used to be the course no one wanted to go to  
  
joszx@LAPTOP-K2E8TDH5: ~  
joszx@LAPTOP-K2E8TDH5:~$ kafkacat -C -b kafka-1:20092 -t topic1  
% Reached end of topic topic1 [0] at offset 0  
this is a message  
% Reached end of topic topic1 [0] at offset 1  
dear lord save our soul from CS  
% Reached end of topic topic1 [0] at offset 2  
i think its interesting how the profs mention that CS used to be the course no one wanted to go to  
% Reached end of topic topic1 [0] at offset 3  
-
```

10. To test successful management of the failure of the master node in the cluster, stop the control node by stopping its docker container using the following command:

```
docker stop otot-d-kafka-3-1
```

From step 5 we know that broker 3 is the controller, and from step 4 we can get its name “otot-d-kafka-3-1” to stop it.

```
joszx@LAPTOP-K2E8TDH5:/mnt/c/Users/ZX/Documents/VScode projects/OTOT-D$ docker stop otot-d-kafka-3-1  
otot-d-kafka-3-1
```

11. Check the new controller with the following command:

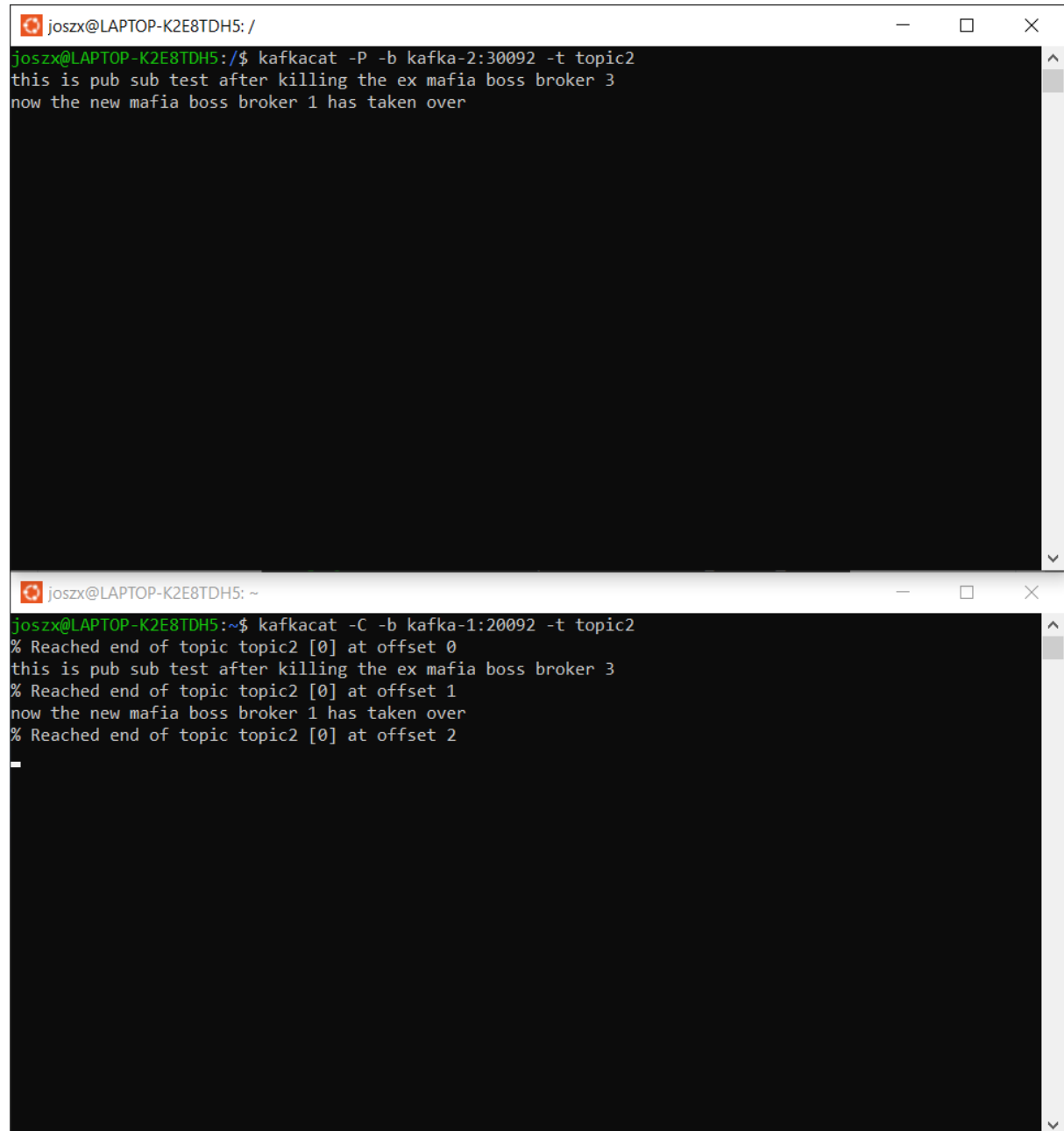
```
kafkacat -L -b kafka-1:20092
```

```
joszx@LAPTOP-K2E8TDH5:/mnt/c/Users/ZX/Documents/VScode projects/OTOT-D$ kafkacat -L -b kafka-1:20092  
Metadata for all topics (from broker 1: kafka-1:20092/1):  
2 brokers:  
  broker 2 at kafka-2:30092  
  broker 1 at kafka-1:20092 (controller)  
2 topics:  
  topic "test_topic" with 1 partitions:  
    partition 0, leader 1, replicas: 1, isrs: 1  
  topic "topic1" with 1 partitions:  
    partition 0, leader -1, replicas: 3, isrs: 3, Broker: Leader not available
```

Broker 1 is now the controller

12. Check Pub-Sub messaging system is still working by repeating steps 7 to 9.

Pub-Sub still working:



The image shows two terminal windows from a user named 'joszx' on a machine named 'LAPTOP-K2E8TDH5'. The top window shows a Kafka consumer command being executed, which outputs a message about a mafia boss broker taking over. The bottom window shows the same command being executed again, but with additional output indicating that the consumer has reached the end of the topic at offset 0, 1, and 2.

```
joszx@LAPTOP-K2E8TDH5: /  
joszx@LAPTOP-K2E8TDH5:/$ kafkacat -P -b kafka-2:30092 -t topic2  
this is pub sub test after killing the ex mafia boss broker 3  
now the new mafia boss broker 1 has taken over  
  
joszx@LAPTOP-K2E8TDH5: ~  
joszx@LAPTOP-K2E8TDH5:~$ kafkacat -C -b kafka-1:20092 -t topic2  
% Reached end of topic topic2 [0] at offset 0  
this is pub sub test after killing the ex mafia boss broker 3  
% Reached end of topic topic2 [0] at offset 1  
now the new mafia boss broker 1 has taken over  
% Reached end of topic topic2 [0] at offset 2  
-
```