

# CS3219 OTOT Task D

- **Name:** Ryan Aidan
- **Matric. Number:** A0218327E
- **Repo Link:** <https://github.com/aidanaden/otot-d>

## Implement 1-Zookeeper + 3-Node Kafka cluster with Docker

To implement a pub-sub messaging system, we'll be using kafka with 3 nodes (or brokers) and 1 zookeeper via Docker.

### Requirements

To set up the cluster, you will need to install Docker.

### Dockerizing

To create the dockerised cluster, we'll be using `docker-compose`.

1. Create a file named `docker-compose.yml` with the following below:

```
version: "3"

services:
  zoo1:
    image: confluentinc/cp-zookeeper:7.3.0
    hostname: zoo1
    container_name: zoo1
    ports:
      - "2181:2181"
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_SERVER_ID: 1
      ZOOKEEPER_SERVERS: zoo1:2888:3888

  kafka1:
    image: confluentinc/cp-kafka:7.3.0
    hostname: kafka1
    container_name: kafka1
    ports:
      - "9092:9092"
      - "29092:29092"
    environment:
      KAFKA_ADVERTISED_LISTENERS:
INTERNAL://kafka1:19092,EXTERNAL://${DOCKER_HOST_IP:-127.0.0.1}:9092,DOCKER://host.docker.internal:29092

      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INTERNAL:PLAINTEXT,EXTERNAL:PLAINTEXT,DOCKER:PLAINTEXT
      KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
      KAFKA_ZOOKEEPER_CONNECT: "zoo1:2181"
      KAFKA_BROKER_ID: 1
      KAFKA_LOG4J_LOGGERS:
"kafka.controller=INFO,kafka.producer.async.DefaultEventHandler=INFO,state.change.logger=INFO"
      KAFKA_AUTHORIZER_CLASS_NAME: kafka.security.authorizer.AclAuthorizer
      KAFKA_ALLOW_EVERYONE_IF_NO_ACL_FOUND: "true"
    depends_on:
      - zoo1

  kafka2:
    image: confluentinc/cp-kafka:7.3.0
    hostname: kafka2
    container_name: kafka2
    ports:
      - "9093:9093"
```

```

- "29093:29093"
environment:
  KAFKA_ADVERTISED_LISTENERS:
INTERNAL://kafka2:19093,EXTERNAL://${DOCKER_HOST_IP:-127.0.0.1}:9093,DOCKER://host.docker.internal:29093

  KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INTERNAL:PLAINTEXT,EXTERNAL:PLAINTEXT,DOCKER:PLAINTEXT
  KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
  KAFKA_ZOOKEEPER_CONNECT: "zoo1:2181"
  KAFKA_BROKER_ID: 2
  KAFKA_LOG4J_LOGGERS:
"kafka.controller=INFO,kafka.producer.async.DefaultEventHandler=INFO,state.change.logger=INFO"
  KAFKA_AUTHORIZER_CLASS_NAME: kafka.security.authorizer.AclAuthorizer
  KAFKA_ALLOW_EVERYONE_IF_NO_ACL_FOUND: "true"
depends_on:
- zoo1

kafka3:
  image: confluentinc/cp-kafka:7.3.0
  hostname: kafka3
  container_name: kafka3
  ports:
  - "9094:9094"
  - "29094:29094"
  environment:
    KAFKA_ADVERTISED_LISTENERS:
INTERNAL://kafka3:19094,EXTERNAL://${DOCKER_HOST_IP:-127.0.0.1}:9094,DOCKER://host.docker.internal:29094

    KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INTERNAL:PLAINTEXT,EXTERNAL:PLAINTEXT,DOCKER:PLAINTEXT
    KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
    KAFKA_ZOOKEEPER_CONNECT: "zoo1:2181"
    KAFKA_BROKER_ID: 3
    KAFKA_LOG4J_LOGGERS:
"kafka.controller=INFO,kafka.producer.async.DefaultEventHandler=INFO,state.change.logger=INFO"
    KAFKA_AUTHORIZER_CLASS_NAME: kafka.security.authorizer.AclAuthorizer
    KAFKA_ALLOW_EVERYONE_IF_NO_ACL_FOUND: "true"
  depends_on:
  - zoo1

```

3. Create a topic using the `create-topic.sh` script

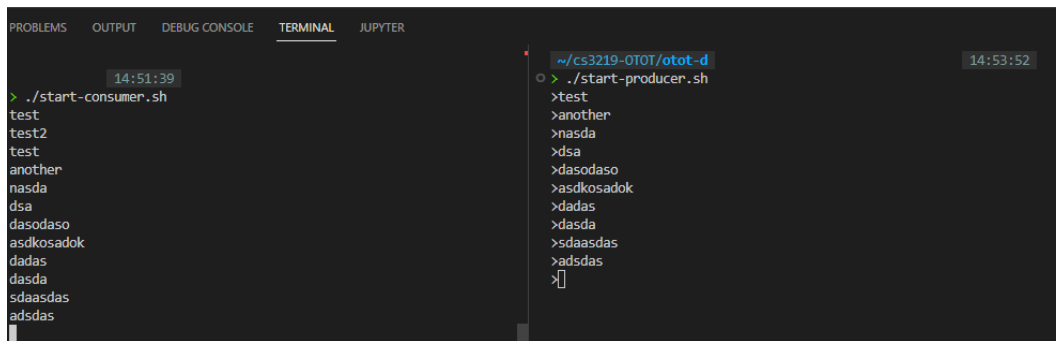
```
bash create-topic.sh
```

4. Verify that publishing and subscribing of messages work with 2 terminals

```
# terminal 1 (for publishing messages)
bash start-producer.sh
<insert-messages-to-publish>
```

```
# terminal 2 (for reading messages)
bash start-consumer.sh # reads message from kafka1
```

5. If successful, should see the following output.



```
14:51:39
> ./start-consumer.sh
test
test2
test
another
nasda
dsa
dasodaso
asdkosadok
dadas
dasda
sdaasdas
adsdas

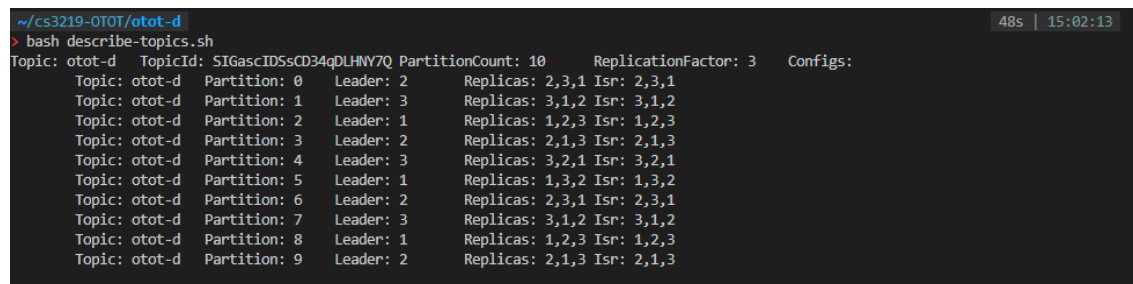
~/cs3219-OTOT/otot-d 14:53:52
> ./start-producer.sh
test
another
nasda
dsa
dasodaso
asdkosadok
ddadas
dasda
sdaasdas
adsdas
>
```

## Test management of master node failure

1. Verify current leader nodes with the `describe-topics.sh` script

```
bash describe-topics.sh
```

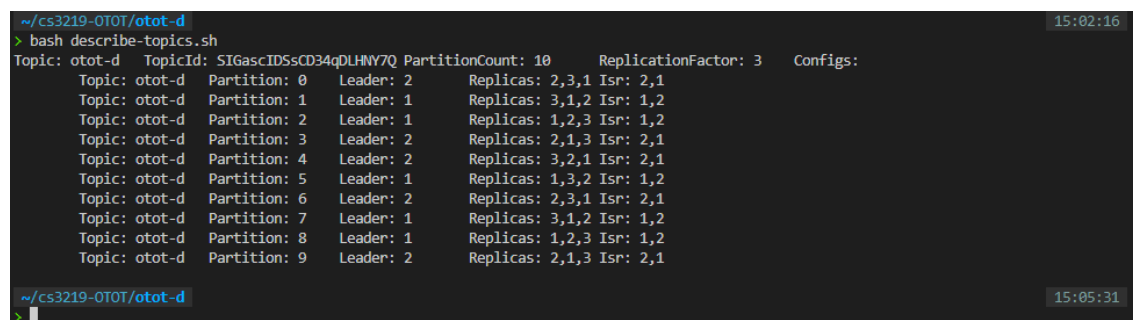
output:



```
~/cs3219-OTOT/otot-d 48s | 15:02:13
> bash describe-topics.sh
Topic: otot-d TopicId: SIGascIDSsCD34qDLHNY7Q PartitionCount: 10 ReplicationFactor: 3 Configs:
  Topic: otot-d Partition: 0 Leader: 2 Replicas: 2,3,1 Isr: 2,3,1
  Topic: otot-d Partition: 1 Leader: 3 Replicas: 3,1,2 Isr: 3,1,2
  Topic: otot-d Partition: 2 Leader: 1 Replicas: 1,2,3 Isr: 1,2,3
  Topic: otot-d Partition: 3 Leader: 2 Replicas: 2,1,3 Isr: 2,1,3
  Topic: otot-d Partition: 4 Leader: 3 Replicas: 3,2,1 Isr: 3,2,1
  Topic: otot-d Partition: 5 Leader: 1 Replicas: 1,3,2 Isr: 1,3,2
  Topic: otot-d Partition: 6 Leader: 2 Replicas: 2,3,1 Isr: 2,3,1
  Topic: otot-d Partition: 7 Leader: 3 Replicas: 3,1,2 Isr: 3,1,2
  Topic: otot-d Partition: 8 Leader: 1 Replicas: 1,2,3 Isr: 1,2,3
  Topic: otot-d Partition: 9 Leader: 2 Replicas: 2,1,3 Isr: 2,1,3
```

2. Repeat the command above to AFTER shutting down kafka3 to re-verify leader nodes

output:



```
~/cs3219-OTOT/otot-d 15:02:16
> bash describe-topics.sh
Topic: otot-d TopicId: SIGascIDSsCD34qDLHNY7Q PartitionCount: 10 ReplicationFactor: 3 Configs:
  Topic: otot-d Partition: 0 Leader: 2 Replicas: 2,3,1 Isr: 2,1
  Topic: otot-d Partition: 1 Leader: 1 Replicas: 3,1,2 Isr: 1,2
  Topic: otot-d Partition: 2 Leader: 1 Replicas: 1,2,3 Isr: 1,2
  Topic: otot-d Partition: 3 Leader: 2 Replicas: 2,1,3 Isr: 2,1
  Topic: otot-d Partition: 4 Leader: 2 Replicas: 3,2,1 Isr: 2,1
  Topic: otot-d Partition: 5 Leader: 1 Replicas: 1,3,2 Isr: 1,2
  Topic: otot-d Partition: 6 Leader: 2 Replicas: 2,3,1 Isr: 2,1
  Topic: otot-d Partition: 7 Leader: 1 Replicas: 3,1,2 Isr: 1,2
  Topic: otot-d Partition: 8 Leader: 1 Replicas: 1,2,3 Isr: 1,2
  Topic: otot-d Partition: 9 Leader: 2 Replicas: 2,1,3 Isr: 2,1

~/cs3219-OTOT/otot-d 15:05:31
>
```

3. If successful, new leaders should have been elected as seen in the output above and pub-sub messages continue to be sent:

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER

dasodaso  
asdkosadok  
dadas  
dasda  
sdaasdas  
adsdas  
hello  
123  
456  
789  
sdads  
123  
12345  
this is a new message after shutting down k  
afka3  
█

>dasodaso  
>asdkosadok  
>dadas  
>dasda  
>sdaasdas  
>adsdas  
>hello  
>123  
>456  
>789  
>sdads  
>123  
>12345  
>this is a new message after shutting down  
kafka3  
>█

dadas  
dasda  
sdaasdas  
adsdas  
hello  
test  
test2  
123  
456  
789  
sdads  
123  
12345  
this is a new message after shutting down  
kafka3  
█