Project Details

Task: D
Done by: Tan Wei Jie (A0202017B)
Repo Link: https://github.com/tanweijie123/CS3219_Sandbox/tree/main/Task_D

Implementation of Pub-Sub System using Apache Kafka

1. In this docker compose file, it will create 1 Zookeeper and 3 Kafka cluster setup.
2. To start the system, run `docker-compose up -d` on this directory. You should see a network is being created.

```
PS C:\Users\tanwe\Desktop\Git\CS3219_sandbox\Task_D> docker-compose up -d
Creating network "task_d_default" with the default driver
Creating task_d_zk-1_1 ... done
Creating task_d_kafka-1_1 ... done
Creating task_d_kafka-2_1 ... done
Creating task_d_kafka-3_1 ... done
PS C:\Users\tanwe\Desktop\Git\CS3219_sandbox\Task_D> docker ps
CONTAINER ID   IMAGE                             COMMAND                  CREATED          STATUS         PORTS
                                                 NAMES
0d4a724c1cfb   confluentinc/cp-kafka:latest      "/etc/confluent/dock…"   8 seconds ago    Up 4 seconds   9092/tcp, 0.0
.0.0:19092->19092/tcp, :::19092->19092/tcp       task_d_kafka-1_1
f571b19f4924   confluentinc/cp-kafka:latest      "/etc/confluent/dock…"   8 seconds ago    Up 4 seconds   9092/tcp, 0.0
.0.0:39092->39092/tcp, :::39092->39092/tcp       task_d_kafka-3_1
c6f2b6dfe59c   confluentinc/cp-kafka:latest      "/etc/confluent/dock…"   8 seconds ago    Up 5 seconds   9092/tcp, 0.0
.0.0:29092->29092/tcp, :::29092->29092/tcp       task_d_kafka-2_1
45656c0deec7   confluentinc/cp-zookeeper:latest  "/etc/confluent/dock…"   10 seconds ago   Up 7 seconds   2888/tcp, 388
8/tcp, 0.0.0.0:22181->2181/tcp, :::22181->2181/tcp   task_d_zk-1_1
PS C:\Users\tanwe\Desktop\Git\CS3219_sandbox\Task_D>
```

3. To create a messaging channel, open bash in 1 of the nodes with `docker exec -it task_d_kafka-1_1 bash`.
4. I will be creating a topic called `interesting`. To do this, run `kafka-topics --create --zookeeper zk-1:2181 --replication-factor 2 --partitions 3 --topic interesting`.
5. To check if it is set up correctly, you can run `kafka-topics --list --zookeeper zk-1:2181` to list all the open topics.

```
PS C:\Users\tanwe\Desktop\Git\CS3219_sandbox\Task_D> docker exec -it task_d_kafka-1_1 bash
[appuser@0d4a724c1cfb ~]$ kafka-topics --create --zookeeper zk-1:2181 --replication-factor 2 --partitions 3 --topic inte
resting
Created topic interesting.
[appuser@0d4a724c1cfb ~]$ kafka-topics --list --zookeeper zk-1:2181
interesting
[appuser@0d4a724c1cfb ~]$
```

6. To send message over `interesting` topic, run `kafka-console-producer --broker-list localhost:9092 --topic interesting`.
7. You can receive the message in another node. To receive message from `interesting` topic, run `kafka-console-consumer --bootstrap-server localhost:9092 --topic interesting --from-beginning`.

```
[appuser@0d4a724c1cfb ~]$ kafka-console-producer --broker-list localhost:9092 --topic interesting
>wow. thats interesting!
>1
>2
>3
>
```

Kafka 1 (Producer)

```
PS C:\Users\tanwe> docker exec -it task_d_kafka-2_1 bash
[appuser@c6f2b6dfe59c ~]$ kafka-console-consumer --bootstrap-server localhost:9092 --topic interesting --from-beginning
wow. thats interesting!
1
2
3
```

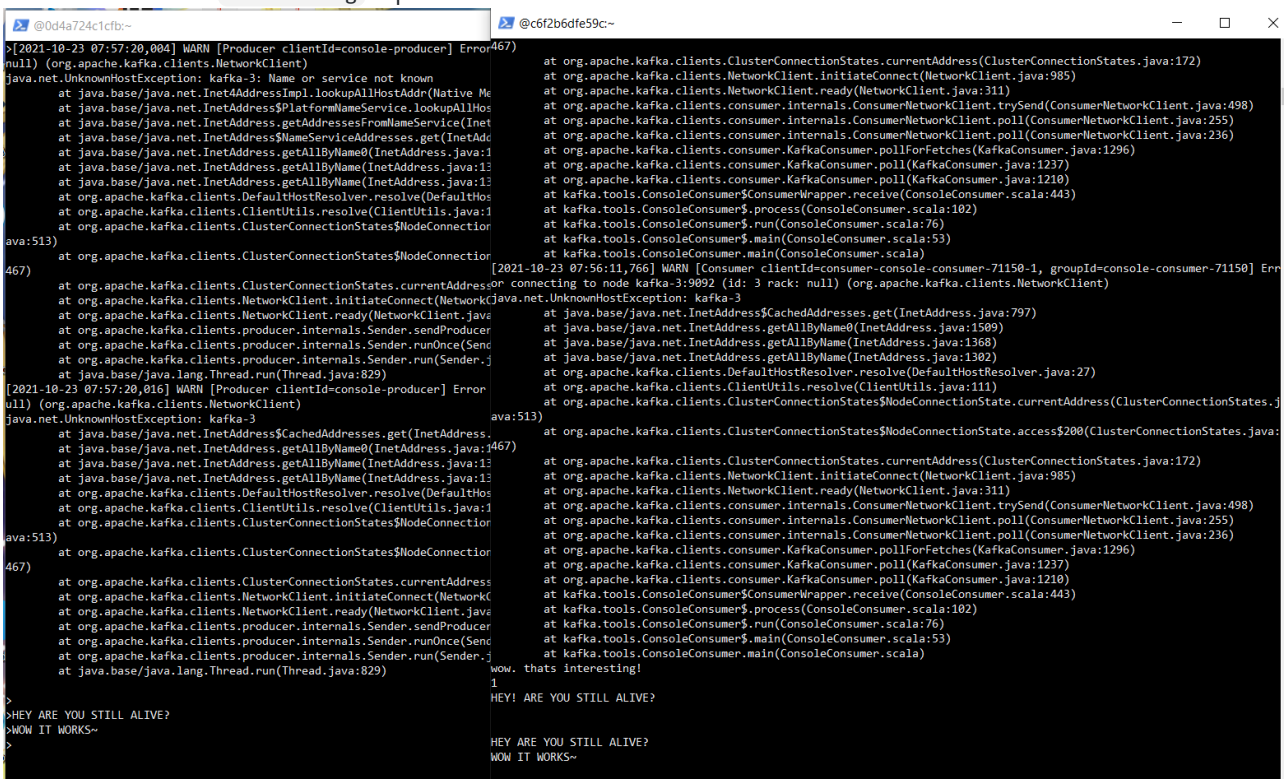
Kafka 2 (Consumer)

Another node takeover

1. Using this command `kafka-topics --zookeeper localhost:2181 --describe --topic interesting`, we can find the leader of the topic.
2. To check if another node will take over its place, I had killed off the leader 3.

```
PS C:\Users\tanwe> docker exec -it task_d_zk-1_1 bash
[appuser@45656c0deec7 ~]$ kafka-topics --zookeeper localhost:2181 --describe --topic interesting
Topic: interesting      TopicId: JDyLRmY1SwKC7TM1PuG4ZQ PartitionCount: 3       ReplicationFactor: 2    Configs:
        Topic: interesting      Partition: 0    Leader: 3       Replicas: 3,2   Isr: 3,2
        Topic: interesting      Partition: 1    Leader: 1       Replicas: 1,3   Isr: 1,3
        Topic: interesting      Partition: 2    Leader: 2       Replicas: 2,1   Isr: 2,1
[appuser@45656c0deec7 ~]$ exit
exit
PS C:\Users\tanwe> docker kill task_d_kafka-3_1
task_d_kafka-3_1
PS C:\Users\tanwe> docker exec -it task_d_zk-1_1 bash
[appuser@45656c0deec7 ~]$ kafka-topics --zookeeper localhost:2181 --describe --topic interesting
Topic: interesting      TopicId: JDyLRmY1SwKC7TM1PuG4ZQ PartitionCount: 3       ReplicationFactor: 2    Configs:
        Topic: interesting      Partition: 0    Leader: 2       Replicas: 3,2   Isr: 2
        Topic: interesting      Partition: 1    Leader: 1       Replicas: 1,3   Isr: 1
        Topic: interesting      Partition: 2    Leader: 2       Replicas: 2,1   Isr: 2,1
[appuser@45656c0deec7 ~]$
```

3. After displaying the error of not being able to connect to `kafka-3`, the producer and consumer are still able to communicate over `interesting` topic.



Resources

Resources that are used and referred to during the creation of this project.

Desc	Link
Kafka Docker Setup	https://www.baeldung.com/ops/kafka-docker-setup
Kafka Command Line Tutorial	http://cloudurable.com/blog/kafka-tutorial-kafka-from-command-line/index.html