Student name and matriculation number: Shawn Lee Shi Jie (A0190131W)

Link to GitHub repository: https://github.com/shawnlsj97/CS3219_D

Task D

Pub-Sub messaging system using Apache Kafka

1. Ensure that docker and docker-compose are installed
2. Install kafkacat by running the command: `brew install kafkacat`
3. Edit the /etc/hosts file using the command: `sudo nano /etc/hosts`
   Add the following line: `0.0.0.0 kafka1 kafka2 kafka3`

```
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting.  Do not change this entry.
##
127.0.0.1       localhost
255.255.255.255 broadcasthost
::1             localhost
0.0.0.0 kafka1 kafka2 kafka3
# Added by Docker Desktop
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section


                [ Read 14 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

4. In a terminal, clone the repository by running the command: `git clone https://github.com/shawnlsj97/CS3219_D`
5. Navigate to the cloned repository and start the Kafka cluster by running the command: `docker-compose up -d`

```
shawn@Shawns-iMac files % docker-compose up -d
Docker Compose is now in the Docker CLI, try `docker compose up`

Creating network "files_default" with the default driver
Creating files_zoo2_1 ... done
Creating files_zoo1_1 ... done
Creating files_zoo3_1 ... done
Creating files_kafka1_1 ... done
Creating files_kafka2_1 ... done
Creating files_kafka3_1 ... done
```

6. Verify that the containers are up and running by running the command: `docker-compose ps`

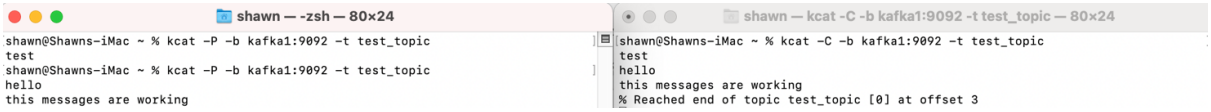
```
shawn@Shawns-iMac files % docker-compose ps
      Name                     Command               State             Ports
-------------------------------------------------------------------------------------
files_kafka1_1   /etc/confluent/docker/run    Up      0.0.0.0:9092->9092/tcp,:::
                                                      9092->9092/tcp
files_kafka2_1   /etc/confluent/docker/run    Up      9092/tcp, 0.0.0.0:9093->90
                                                      93/tcp,:::9093->9093/tcp
files_kafka3_1   /etc/confluent/docker/run    Up      9092/tcp, 0.0.0.0:9094->90
                                                      94/tcp,:::9094->9094/tcp
files_zoo1_1     /docker-entrypoint.sh zkSe   Up      0.0.0.0:2181->2181/tcp,:::
                 ...                                  2181->2181/tcp, 2888/tcp,
                                                      3888/tcp, 8080/tcp
files_zoo2_1     /docker-entrypoint.sh zkSe   Up      2181/tcp, 0.0.0.0:2182->21
                 ...                                  82/tcp,:::2182->2182/tcp,
                                                      2888/tcp, 3888/tcp,
                                                      8080/tcp
files_zoo3_1     /docker-entrypoint.sh zkSe   Up      2181/tcp, 0.0.0.0:2183->21
                 ...                                  83/tcp,:::2183->2183/tcp,
                                                      2888/tcp, 3888/tcp,
                                                      8080/tcp
```

7. In a new terminal, create a producer that connects to the “kafka1” node and publishes messages to “test_topic” by running the command: `kcat -P -b kafka1:9092 -t test_topic`

Note that you may choose to connect to any of the other Kafka nodes by replacing "kafka1" with "kafka2" or "kafka3" in the above command.

8. In a 3rd terminal, create a consumer that connects to the "kafka1" node and subscribes to "test_topic" by running the command: `kafkacat -C -b kafka1:9092 -t test_topic`

   Similar to the previous step, you may choose to connect to any of the other kafka nodes by replacing "kafka1" with "kafka2" or "kafka3" in the above command.
9. Test the pub sub messaging by entering your messages in the producer terminal and hitting enter. When you are done, press `ctrl-D` to send the message to the topic. You should see that the same message is displayed in the consumer terminal.



10. Clean up and stop all containers by running the command in the first terminal: `docker-compose down`

```
shawn@Shawns-iMac files % docker-compose down
Stopping files_kafka1_1 ... done
Stopping files_kafka2_1 ... done
Stopping files_kafka3_1 ... done
Stopping files_zoo2_1   ... done
Stopping files_zoo3_1   ... done
Stopping files_zoo1_1   ... done
Removing files_kafka1_1 ... done
Removing files_kafka2_1 ... done
Removing files_kafka3_1 ... done
Removing files_zoo2_1   ... done
Removing files_zoo3_1   ... done
Removing files_zoo1_1   ... done
Removing network files_default
```

Management of failure of master node

1. Start the Kafka cluster in a terminal by running the command: `docker-compose up -d`

```
shawn@Shawns-iMac files % docker-compose up -d
Docker Compose is now in the Docker CLI, try `docker compose up`

Creating network "files_default" with the default driver
Creating files_zoo2_1 ... done
Creating files_zoo1_1 ... done
Creating files_zoo3_1 ... done
Creating files_kafka1_1 ... done
Creating files_kafka2_1 ... done
Creating files_kafka3_1 ... done
```

2. In a second terminal, create a producer that connects to the "kafka1" node and publishes messages to "test_topic" by running the command: `kcat -P -b kafka1:9092 -t test_topic`
3. In a 3rd terminal, create a consumer that connects to the "kafka2" node and subscribes to "test_topic" by running the command: `kafkacat -C -b kafka2:9093 -t test_topic`
4. Get the name of the kafka1 cluster by running the command: `docker-compose ps`

```
shawn@Shawns-iMac files % docker-compose ps
      Name                     Command               State                Ports
-------------------------------------------------------------------------------------------
files_kafka1_1   /etc/confluent/docker/run    Up      0.0.0.0:9092->9092/tcp,:::
                                                      9092->9092/tcp
files_kafka2_1   /etc/confluent/docker/run    Up      9092/tcp, 0.0.0.0:9093->90
                                                      93/tcp,:::9093->9093/tcp
files_kafka3_1   /etc/confluent/docker/run    Up      9092/tcp, 0.0.0.0:9094->90
                                                      94/tcp,:::9094->9094/tcp
files_zoo1_1     /docker-entrypoint.sh zkSe   Up      0.0.0.0:2181->2181/tcp,:::
                 ...                                  2181->2181/tcp, 2888/tcp,
                                                      3888/tcp, 8080/tcp
files_zoo2_1     /docker-entrypoint.sh zkSe   Up      2181/tcp, 0.0.0.0:2182->21
                 ...                                  82/tcp,:::2182->2182/tcp,
                                                      2888/tcp, 3888/tcp,
                                                      8080/tcp
files_zoo3_1     /docker-entrypoint.sh zkSe   Up      2181/tcp, 0.0.0.0:2183->21
                 ...                                  83/tcp,:::2183->2183/tcp,
                                                      2888/tcp, 3888/tcp,
                                                      8080/tcp
```

5. In the first terminal, stop the producer node by running the command: `docker stop <kafka1 cluster name>`
   In this case, it would be: `docker stop files_kafka1_1`
6. Test the pub-sub messaging to ensure that it is still working

```
shawn@Shawns-iMac ~ % kcat -P -b kafka1:9092 -t test_topic
%6|1635241273.024|FAIL|rdkafka#producer-1| [thrd:kafka1:9092/bootstrap]: kafka1:
9092/1: Disconnected (after 23009ms in state UP)
this is still working!
```

```
shawn@Shawns-iMac ~ % kcat -C -b kafka2:9093 -t test_topic
% Reached end of topic test_topic [0] at offset 0
this is still working!
% Reached end of topic test_topic [0] at offset 1
```

7. Run the following command to observe the newly appointed controller node: `kcat -L -b kafka2:9093`

```
shawn@Shawns-iMac files % kcat -L -b kafka2:9093
Metadata for all topics (from broker 2: kafka2:9093/2):
 2 brokers:
  broker 2 at kafka2:9093
  broker 3 at kafka3:9094 (controller)
 1 topics:
  topic "test_topic" with 1 partitions:
    partition 0, leader 2, replicas: 2, isrs: 2
```

8. Clean up and stop all containers by running the command in the first terminal: `docker-compose down`

```
shawn@Shawns-iMac files % docker-compose down
Stopping files_kafka1_1 ... done
Stopping files_kafka2_1 ... done
Stopping files_kafka3_1 ... done
Stopping files_zoo2_1   ... done
Stopping files_zoo3_1   ... done
Stopping files_zoo1_1   ... done
Removing files_kafka1_1 ... done
Removing files_kafka2_1 ... done
Removing files_kafka3_1 ... done
Removing files_zoo2_1   ... done
Removing files_zoo3_1   ... done
Removing files_zoo1_1   ... done
Removing network files_default
```

References

https://hub.docker.com/_/zookeeper
https://github.com/wurstmeister/kafka-docker
https://docs.confluent.io/3.3.0/app-development/kafkacat-usage.html