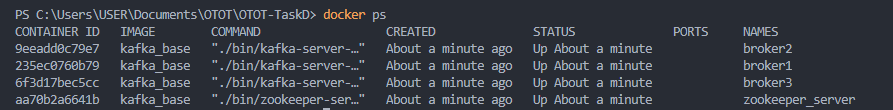
OTOT\_Task\_D

Goh Ee Liang, A0202170B

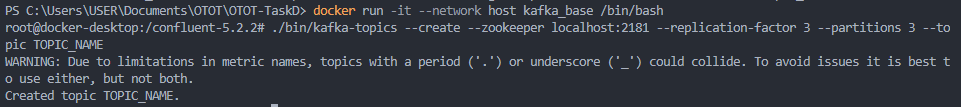
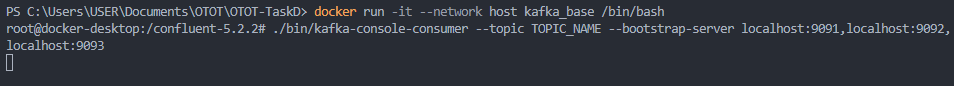
GitHub link: <https://github.com/Elgoh/PubSubApp>

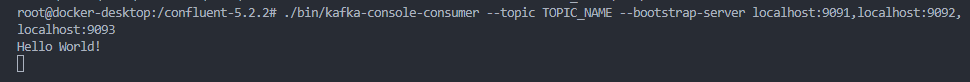
Steps to begin:

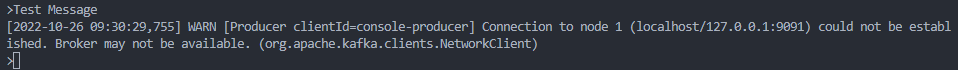
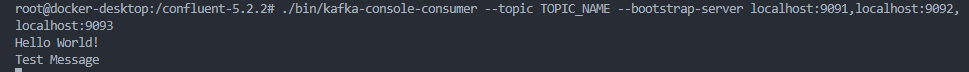
1. Ensure that docker engine and docker-compose is installed.
   1. <https://docs.docker.com/get-docker/>
2. Clone the repo from the GitHub link above
3. Open a new terminal and run `docker build -t kafka\_base .`
   1. The build is based on the Dockerfile in the repo
   2. Open-jdk-8 & configuration files for kafka broker are installed and downloaded
   3. The image will be the be created
4. Using the same terminal, run `docker-compose up`
   1. This launches the cluster with 4 containers as seen in the docker-compose.yml file
   2. Text

      Description automatically generated
   3. 

Steps to test:

1. Creation of topic
   1. First run a client with the command:
      1. Open a new terminal
      2. `docker run -it --network host kafka\_base /bin/bash`
   2. Next create a topic with the command:
      1. `./bin/kafka-topics --create --zookeeper localhost:2181 --replication-factor 3 --partitions 3 --topic TOPIC\_NAME`
      2. You can change TOPIC\_NAME to whatever you like
      3. NOTE: message published on `TOPIC\_NAME` can only be consumed by consumer that subscribes to `TOPIC\_NAME`
      4. 
2. Creation of Publisher/Producer
   1. Using the above terminal,
   2. Run `./bin/kafka-console-producer --broker-list localhost:9091,localhost:9092,localhost:9093 --topic TOPIC\_NAME`
   3. 
3. Creation of Subscriber/Consumer
   1. Open a new terminal
   2. Run `docker run -it --network host kafka\_base /bin/bash`
   3. Run `./bin/kafka-console-consumer --topic TOPIC\_NAME --bootstrap-server localhost:9091,localhost:9092,localhost:9093`
   4. 
4. Test that message can be passed from Publisher to Subscriber
   1. Type a message in the Publisher’s terminal as below
   2. Text

      Description automatically generated
   3. Check Subscriber’s terminal and it should contain the message as it is subscribed to the same topic as the Publisher
   4. 
5. Test that if a Kafka broker is down, messages will still be able to pass through other brokers
   1. Open a new terminal
   2. Run `docker stop broker1`
   3. Run `docker ps` and check that broker 1 has indeed stopped
   4. Graphical user interface, text

      Description automatically generated
   5. Send a message from the publisher
   6. 
   7. Check that message is received from subscriber
   8. 
   9. This can be done on any broker and the message will still pass through, if time permits, do try it.