

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
0d4a724c1cfb   confluentinc/cp-kafka:latest       "/etc/confluent/dock...  8 seconds ago Up 4 seconds  9092/tcp, 0.0
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->22181/tcp, :::22181->22181/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.

```
> @0d4a724c1cf:-
[2021-10-23 07:57:20,004] WARN [Producer clientId=console-producer] Error(467)
null) (org.apache.kafka.clients.NetworkClient)
java.net.UnknownHostException: kafka-3: Name or service not known
    at java.base/java.net.InetAddressImpl.lookupAllHostAddr(Native Method)
    at java.base/java.net.InetAddress$ImplFormNameService.lookupAllHostAddr(InetAddress$ImplFormNameService.java:118)
    at java.base/java.net.InetAddress$NameServiceAddresses.get(InetAddress$NameServiceAddresses.java:109)
    at java.base/java.net.InetAddress.getAllByName0(InetAddress.java:1531)
    at java.base/java.net.InetAddress.getAllByName(InetAddress.java:1513)
    at java.base/java.net.InetAddress.getAllByName(InetAddress.java:1513)
    at org.apache.kafka.clients.DefaultHostResolver.resolve(DefaultHostResolver.java:27)
    at org.apache.kafka.clients.ClientUtils.resolve(ClientUtils.java:111)
    at org.apache.kafka.clients.ClusterConnectionStates$NodeConnectionState.currentAddress(ClusterConnectionStates.java:172)
    at org.apache.kafka.clients.ClusterConnectionStates$NodeConnectionState.access$200(ClusterConnectionStates.java:195)
    at org.apache.kafka.clients.NetworkClient.initiateConnect(NetworkClient.java:311)
    at org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.trySend(ConsumerNetworkClient.java:498)
    at org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.poll(ConsumerNetworkClient.java:255)
    at org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.poll(KafkaConsumer.java:236)
    at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1296)
    at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1237)
    at org.apache.kafka.tools.ConsoleConsumer$ConsumerWrapper.receive(ConsoleConsumer.scala:443)
    at kafka.tools.ConsoleConsumer$.process(ConsoleConsumer.scala:102)
    at kafka.tools.ConsoleConsumer$.run(ConsoleConsumer.scala:76)
    at kafka.tools.ConsoleConsumer$.main(ConsoleConsumer.scala:53)
    at kafka.tools.ConsoleConsumer.main(ConsoleConsumer.scala)
[2021-10-23 07:56:11,766] WARN [consumer clientId=console-consumer-71150-1, groupId=console-consumer-71150] Err
or connecting to node kafka-3:9092 (id: 3 rack: null) (org.apache.kafka.clients.NetworkClient)
java.net.UnknownHostException: kafka-3
    at java.base/java.net.InetAddress$CachedAddresses.get(InetAddress.java:797)
    at java.base/java.net.InetAddress.getAllByName0(InetAddress.java:1509)
    at java.base/java.net.InetAddress.getAllByName(InetAddress.java:1368)
    at java.base/java.net.InetAddress.getAllByName(InetAddress.java:1302)
    at org.apache.kafka.clients.DefaultHostResolver.resolve(DefaultHostResolver.java:27)
    at org.apache.kafka.clients.ClientUtils.resolve(ClientUtils.java:111)
    at org.apache.kafka.clients.ClusterConnectionStates$NodeConnectionState.currentAddress(ClusterConnectionStates.java:172)
    at org.apache.kafka.clients.ClusterConnectionStates$NodeConnectionState.access$200(ClusterConnectionStates.java:195)
    at org.apache.kafka.clients.NetworkClient.initiateConnect(NetworkClient.java:311)
    at org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.trySend(ConsumerNetworkClient.java:498)
    at org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.poll(ConsumerNetworkClient.java:255)
    at org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.poll(KafkaConsumer.java:236)
    at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1296)
    at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1237)
    at org.apache.kafka.tools.ConsoleConsumer$ConsumerWrapper.receive(ConsoleConsumer.scala:443)
    at kafka.tools.ConsoleConsumer$.process(ConsoleConsumer.scala:102)
    at kafka.tools.ConsoleConsumer$.run(ConsoleConsumer.scala:76)
    at kafka.tools.ConsoleConsumer$.main(ConsoleConsumer.scala:53)
    at kafka.tools.ConsoleConsumer.main(ConsoleConsumer.scala)
wow, thats interesting!
1
HEY ARE YOU STILL ALIVE?
> HEY ARE YOU STILL ALIVE?
> NOW IT WORKS~>
```

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