

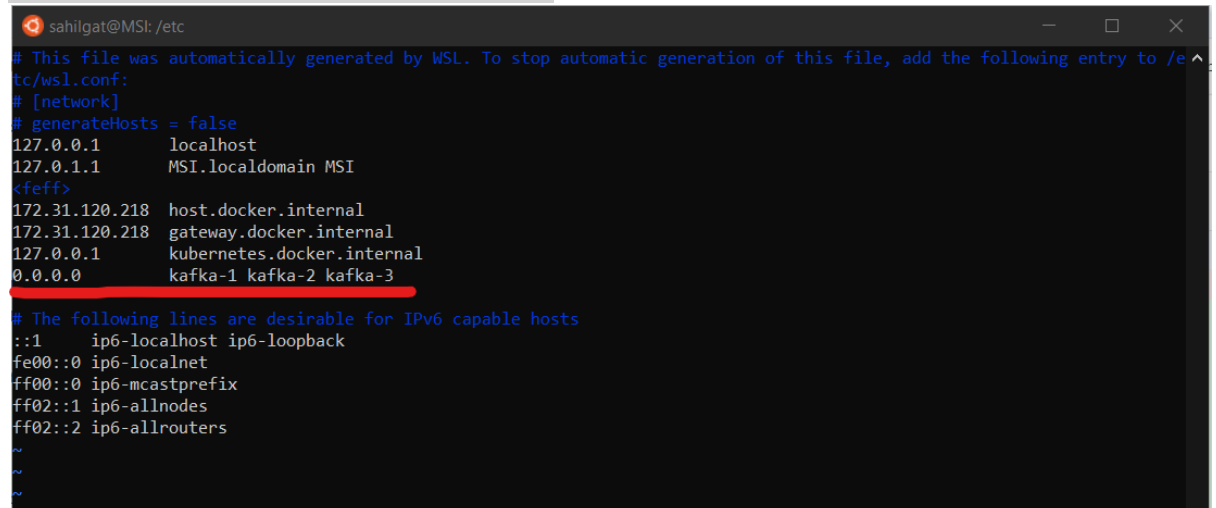
Set-up instructions:

- 1) Ensure you have docker and docker-compose installed
- 2) Ensure you have kafkacat installed. You can use the command below for installation:

```
sudo apt-get install kafkacat
```

- 3) Edit the /etc/hosts file and add the following line:

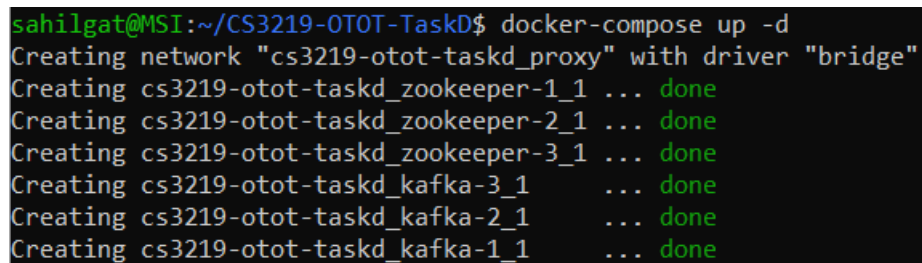
```
0.0.0.0 kafka-1 kafka-2 kafka-3
```



```
sahilgat@MSI: /etc
# This file was automatically generated by WSL. To stop automatic generation of this file, add the following entry to /etc/hosts:
# [network]
# generateHosts = false
127.0.0.1    localhost
127.0.1.1    MSI.localdomain MSI
<feff>
172.31.120.218 host.docker.internal
172.31.120.218 gateway.docker.internal
127.0.0.1    kubernetes.docker.internal
0.0.0.0      kafka-1 kafka-2 kafka-3
# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
~
~
~
```

- 4) Navigate to the root folder after cloning and run:

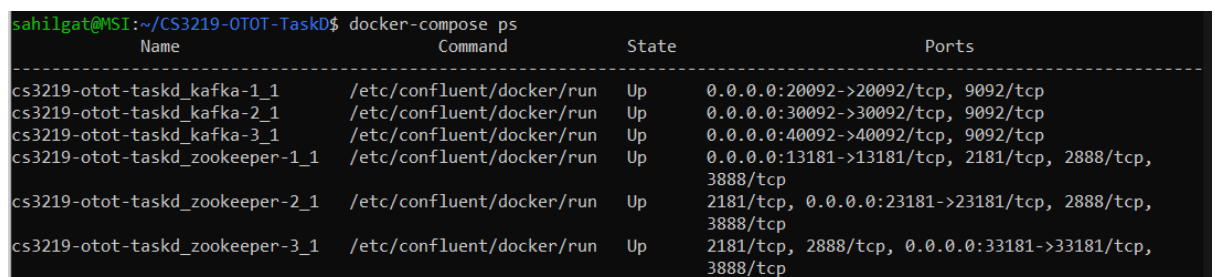
```
sudo docker-compose up -d
```



```
sahilgat@MSI:~/CS3219-OTOT-TaskD$ docker-compose up -d
Creating network "cs3219-otot-taskd_proxy" with driver "bridge"
Creating cs3219-otot-taskd_zookeeper-1_1 ... done
Creating cs3219-otot-taskd_zookeeper-2_1 ... done
Creating cs3219-otot-taskd_zookeeper-3_1 ... done
Creating cs3219-otot-taskd_kafka-3_1 ... done
Creating cs3219-otot-taskd_kafka-2_1 ... done
Creating cs3219-otot-taskd_kafka-1_1 ... done
```

- 5) Use the following command to ensure all services are running:

```
docker-compose ps
```



```
sahilgat@MSI:~/CS3219-OTOT-TaskD$ docker-compose ps
-----
Name                                Command                                State    Ports
-----
cs3219-otot-taskd_kafka-1_1         /etc/confluent/docker/run            Up       0.0.0.0:20092->20092/tcp, 9092/tcp
cs3219-otot-taskd_kafka-2_1         /etc/confluent/docker/run            Up       0.0.0.0:30092->30092/tcp, 9092/tcp
cs3219-otot-taskd_kafka-3_1         /etc/confluent/docker/run            Up       0.0.0.0:40092->40092/tcp, 9092/tcp
cs3219-otot-taskd_zookeeper-1_1     /etc/confluent/docker/run            Up       0.0.0.0:13181->13181/tcp, 2181/tcp, 2888/tcp, 3888/tcp
cs3219-otot-taskd_zookeeper-2_1     /etc/confluent/docker/run            Up       2181/tcp, 0.0.0.0:23181->23181/tcp, 2888/tcp, 3888/tcp
cs3219-otot-taskd_zookeeper-3_1     /etc/confluent/docker/run            Up       2181/tcp, 2888/tcp, 0.0.0.0:33181->33181/tcp, 3888/tcp
```

- 6) Identify a controller using any of the following commands:

```
sudo kafkacat -L -b kafka-1:20092
```

```
sudo kafkacat -L -b kafka-2:30092
```

```
sudo kafkacat -L -b kafka-3:40092
```

```
sahilgat@MSI:~/CS3219-OTOT-TaskD$ sudo kafkacat -L -b kafka-1:20092
Metadata for all topics (from broker 1: kafka-1:20092/1):
 3 brokers:
   broker 2 at kafka-2:30092
   broker 3 at kafka-3:40092
   broker 1 at kafka-1:20092
 0 topics:
```

- 7) Opening another terminal, set the controller you selected as a producer using the following command (change respectively):
`sudo kafkacat -P -b kafka-1:20092 -t test_topic`
- 8) Opening another terminal, set any other cluster as a consumer. For example:
`sudo kafkacat -C -b kafka-2:30092 -t test_topic`
- 9) Test the Pub-Sub messaging feature through the two terminals.

```
sahilgat@MSI:~$ sudo kafkacat -P -b kafka-1:20092 -t test_topic
hello!
This is working :)
Sahil says hi!

sahilgat@MSI:~$ sudo kafkacat -C -b kafka-2:30092 -t test_topic
[sudo] password for sahilgat:
% Reached end of topic test_topic [0] at offset 0
hello!
% Reached end of topic test_topic [0] at offset 1
This is working :)
% Reached end of topic test_topic [0] at offset 2
Sahil says hi!
% Reached end of topic test_topic [0] at offset 3
```

- 10) In the control terminal, stop the controller terminal using the following command (change respectively):
`sudo docker stop cs3219-otot-taskd_kafka-1_1`

```
sahilgat@MSI:~$ sudo docker stop cs3219-otot-taskd_kafka-1_1
cs3219-otot-taskd_kafka-1_1
sahilgat@MSI:~$
```

- 11) Test the Pub-Sub feature through the two terminals again to ensure it is still working. Note that you may have to press return in the consumer terminal to refresh it.

```
sahilgat@MSI:~$ sudo kafkacat -P -b kafka-1:20092 -t test_topic
hello!
This is working :)
Sahil says hi!
I'm back up!
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Connect to ipv4#0.0.0.0:20092 failed: Connection refused
Yay!
Final test!
Goodbye

sahilgat@MSI:~$ sudo kafkacat -C -b kafka-2:30092 -t test_topic
[sudo] password for sahilgat:
% Reached end of topic test_topic [0] at offset 0
hello!
% Reached end of topic test_topic [0] at offset 1
This is working :)
% Reached end of topic test_topic [0] at offset 2
Sahil says hi!
% Reached end of topic test_topic [0] at offset 3
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Receive failed: Disconnected
% ERROR: Local: Broker transport failure: kafka-1:20092/1: Connect to ipv4#0.0.0.0:20092 failed: Connection refused
I'm back up!
% Reached end of topic test_topic [0] at offset 4
Yay!
% Reached end of topic test_topic [0] at offset 5
Final test!
% Reached end of topic test_topic [0] at offset 6
Goodbye
% Reached end of topic test_topic [0] at offset 7
```

- 12) This can be verified by running any of the following commands and observing the new appointed controller cluster:

```
sudo kafkacat -L -b kafka-1:20092
sudo kafkacat -L -b kafka-2:30092
sudo kafkacat -L -b kafka-3:40092
```

```
sahilgat@MSI: ~  
sahilgat@MSI:~$ sudo kafkacat -L -b kafka-1:20092  
% ERROR: Failed to acquire metadata: Local: Broker transport failure  
sahilgat@MSI:~$
```

Note the failure above since the original controller was shut down.

```
sahilgat@MSI:~$ sudo kafkacat -L -b kafka-2:30092  
Metadata for all topics (from broker 2: kafka-2:30092/2):  
 2 brokers:  
   broker 2 at kafka-2:30092  
   broker 3 at kafka-3:40092  
 1 topics:  
   topic "test_topic" with 1 partitions:  
     partition 0, leader 3, replicas: 3, isrs: 3  
sahilgat@MSI:~$
```

- 13) Clean up everything with the following command:
`docker-compose down`

```
sahilgat@MSI:~/CS3219-OTOT-TaskD$ docker-compose down  
Stopping cs3219-otot-taskd_kafka-3_1      ... done  
Stopping cs3219-otot-taskd_kafka-2_1      ... done  
Stopping cs3219-otot-taskd_zookeeper-3_1  ... done  
Stopping cs3219-otot-taskd_zookeeper-2_1  ... done  
Stopping cs3219-otot-taskd_zookeeper-1_1  ... done  
Removing cs3219-otot-taskd_kafka-1_1      ... done  
Removing cs3219-otot-taskd_kafka-3_1      ... done  
Removing cs3219-otot-taskd_kafka-2_1      ... done  
Removing cs3219-otot-taskd_zookeeper-3_1  ... done  
Removing cs3219-otot-taskd_zookeeper-2_1  ... done  
Removing cs3219-otot-taskd_zookeeper-1_1  ... done  
Removing network cs3219-otot-taskd_proxy
```