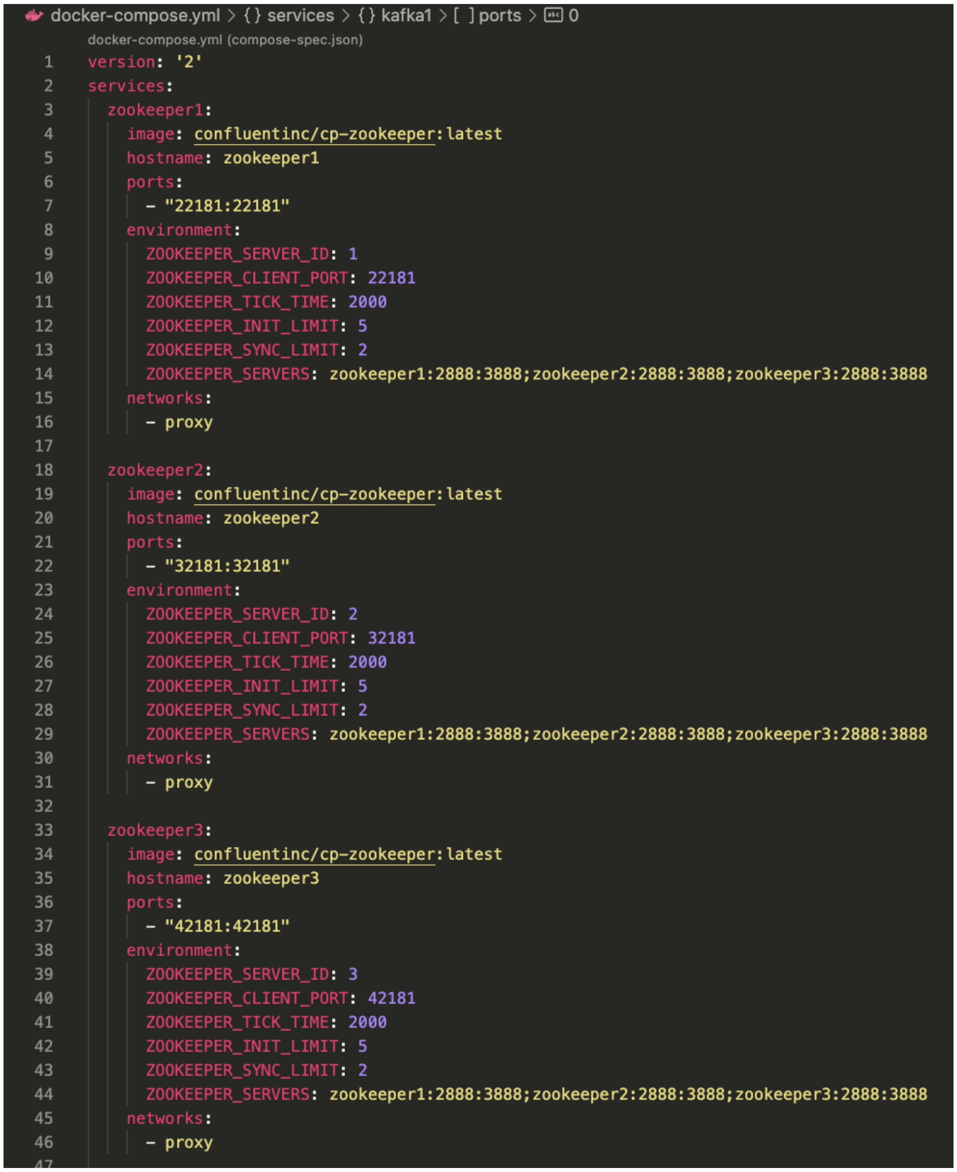
[3 nodes Kafka cluster]

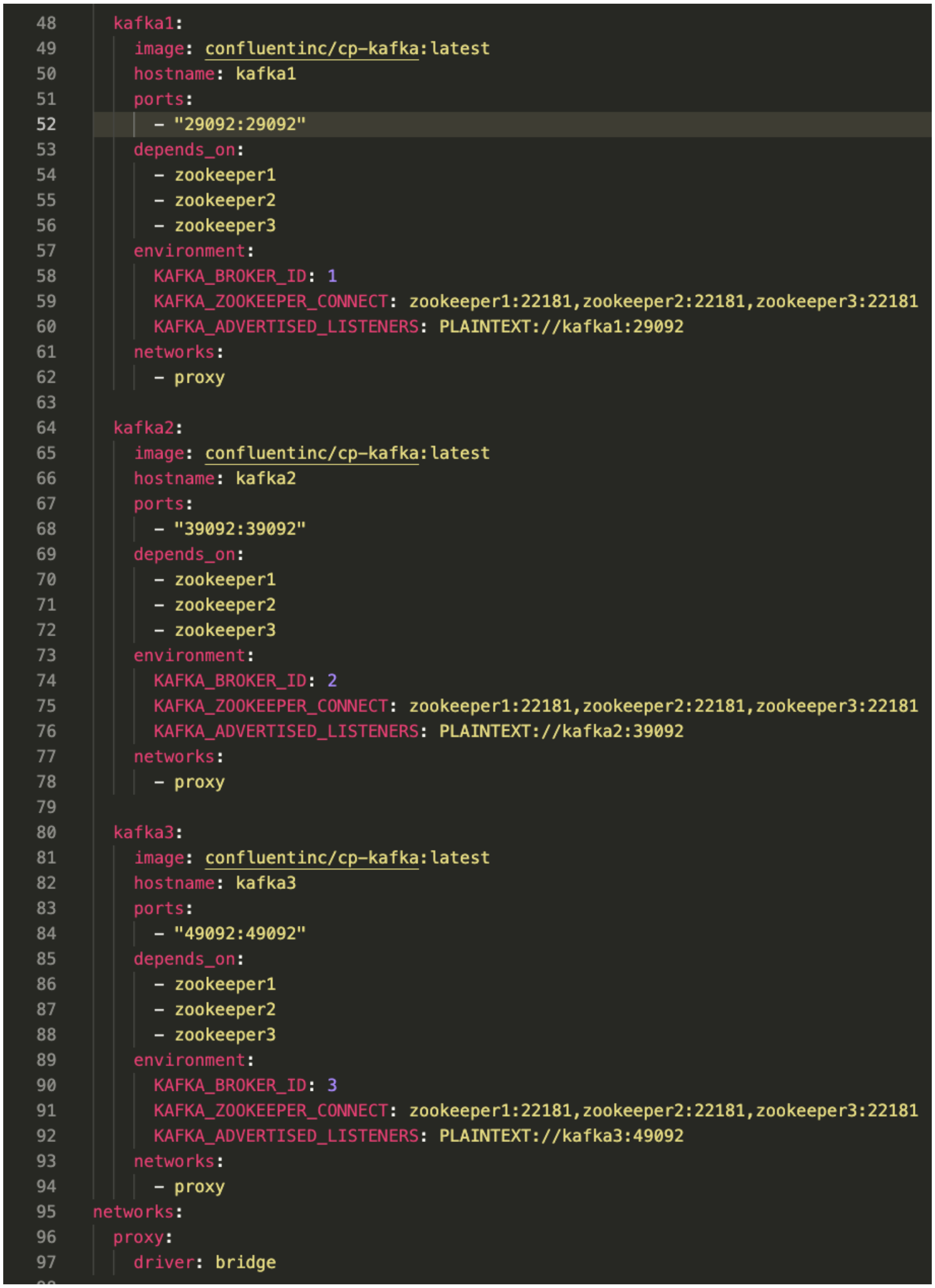
1. Install kafkacat
    a. `brew install kcat`
2. Create a `docker-compose.yml` file
3. This should ensure that Zookeeper always starts before the Kafka server and ends after it
4. `docker-compose.yml`
    a. Should have 6 services - 3 zookeeper services and 3 kafka services
    b. **`image: confluentinc/cp-kafka:latest`** is the docker image for deploying and running of the community version of Kafka
    c. **`image: confluentinc/cp-zookeeper:latest`** is the docker image for deploying and running Zookeeper
    d. Ensure that the service names and KAFKA_BROKER_ID are unique
    e. **`depends_on:`** Kafka depends on Zookeeper to run, so the Zookeeper keys are included in this property to ensure Docker will start the Zookeepers first before the Kafka servers
    f. Zookeeper 1, 2 and 3 are listening to connections by client such as Kafka servers on port 2181 by the ZOOKEEPER_CLIENT_PORT property
    g. All servers must expose unique ports to the host machine
        i. The 3 zookeeper servers are exposed to the host at port 22181, 32181 and 42181 respectively by the ports property
        ii. Same for kafka servers where they will be listening on ports 29092, 39092 and 49092 respectively
    h. Every machine in the Zookeeper ensemble should know about other machines in the ensemble using the ZOOKEEPER_SERVERS property
        i. **`ZOOKEEPER_SERVERS: zookeeper1:2888:3888; zookeeper2:2888:3888;zookeeper3:2888:3888;`**


docker-compose.yml > { } services > { } kafka1 > [ ] ports > 0
docker-compose.yml (compose-spec.json)

```yaml
version: '2'
services:
  zookeeper1:
    image: confluentinc/cp-zookeeper:latest
    hostname: zookeeper1
    ports:
      - "22181:22181"
    environment:
      ZOOKEEPER_SERVER_ID: 1
      ZOOKEEPER_CLIENT_PORT: 22181
      ZOOKEEPER_TICK_TIME: 2000
      ZOOKEEPER_INIT_LIMIT: 5
      ZOOKEEPER_SYNC_LIMIT: 2
      ZOOKEEPER_SERVERS: zookeeper1:2888:3888;zookeeper2:2888:3888;zookeeper3:2888:3888
    networks:
      - proxy

  zookeeper2:
    image: confluentinc/cp-zookeeper:latest
    hostname: zookeeper2
    ports:
      - "32181:32181"
    environment:
      ZOOKEEPER_SERVER_ID: 2
      ZOOKEEPER_CLIENT_PORT: 32181
      ZOOKEEPER_TICK_TIME: 2000
      ZOOKEEPER_INIT_LIMIT: 5
      ZOOKEEPER_SYNC_LIMIT: 2
      ZOOKEEPER_SERVERS: zookeeper1:2888:3888;zookeeper2:2888:3888;zookeeper3:2888:3888
    networks:
      - proxy

  zookeeper3:
    image: confluentinc/cp-zookeeper:latest
    hostname: zookeeper3
    ports:
      - "42181:42181"
    environment:
      ZOOKEEPER_SERVER_ID: 3
      ZOOKEEPER_CLIENT_PORT: 42181
      ZOOKEEPER_TICK_TIME: 2000
      ZOOKEEPER_INIT_LIMIT: 5
      ZOOKEEPER_SYNC_LIMIT: 2
      ZOOKEEPER_SERVERS: zookeeper1:2888:3888;zookeeper2:2888:3888;zookeeper3:2888:3888
    networks:
      - proxy
```

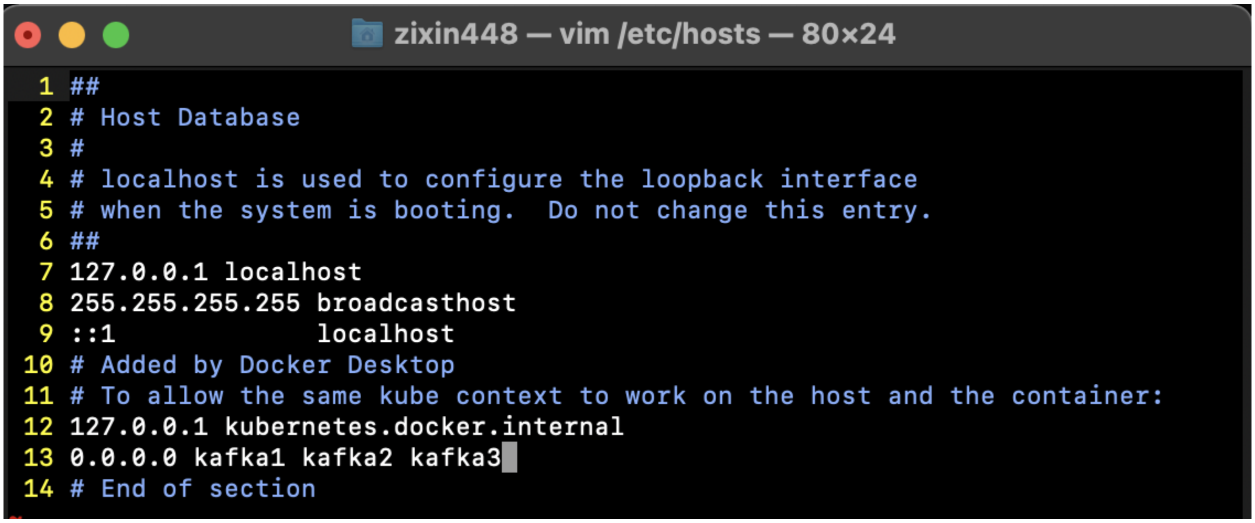


```yaml
  kafka1:
    image: confluentinc/cp-kafka:latest
    hostname: kafka1
    ports:
      - "29092:29092"
    depends_on:
      - zookeeper1
      - zookeeper2
      - zookeeper3
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: zookeeper1:22181,zookeeper2:22181,zookeeper3:22181
      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://kafka1:29092
    networks:
      - proxy

  kafka2:
    image: confluentinc/cp-kafka:latest
    hostname: kafka2
    ports:
      - "39092:39092"
    depends_on:
      - zookeeper1
      - zookeeper2
      - zookeeper3
    environment:
      KAFKA_BROKER_ID: 2
      KAFKA_ZOOKEEPER_CONNECT: zookeeper1:22181,zookeeper2:22181,zookeeper3:22181
      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://kafka2:39092
    networks:
      - proxy

  kafka3:
    image: confluentinc/cp-kafka:latest
    hostname: kafka3
    ports:
      - "49092:49092"
    depends_on:
      - zookeeper1
      - zookeeper2
      - zookeeper3
    environment:
      KAFKA_BROKER_ID: 3
      KAFKA_ZOOKEEPER_CONNECT: zookeeper1:22181,zookeeper2:22181,zookeeper3:22181
      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://kafka3:49092
    networks:
      - proxy
networks:
  proxy:
    driver: bridge
```

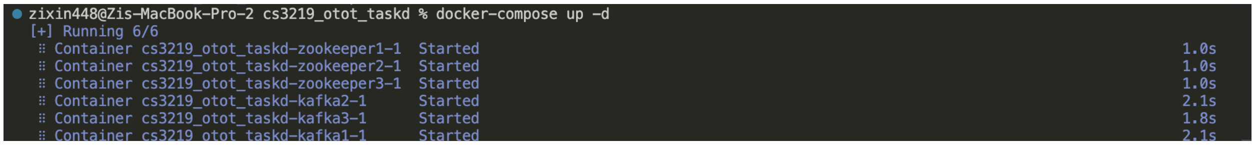
5. Update the /etc/hosts file
    a. `sudo nano /etc/hosts`
    b. Enter password
    c. Add the following line
        i. `0.0.0.0 kafka1 kafka2 kafka3`



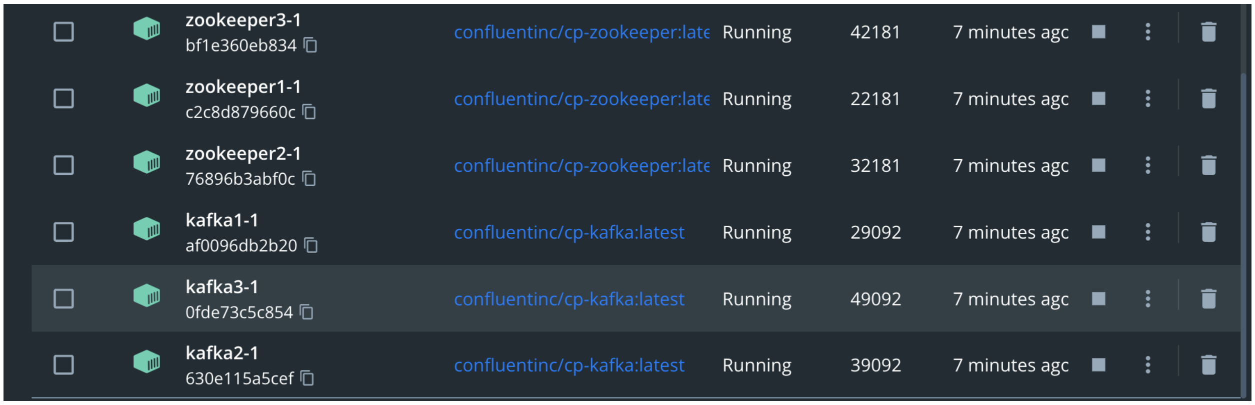
zixin448 — vim /etc/hosts — 80×24

```
##
# 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
# 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
0.0.0.0 kafka1 kafka2 kafka3
# End of section
```

6. Start the cluster using the docker compose command
    a. `docker-compose up -d`

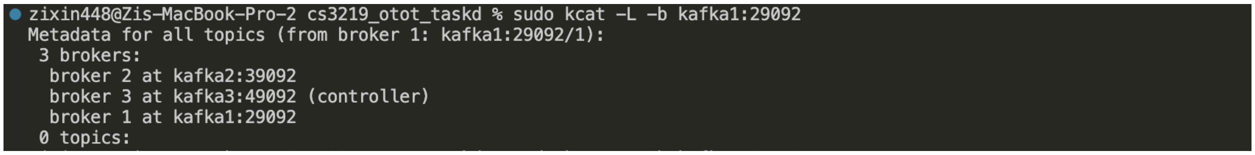


```
● zixin448@Zis-MacBook-Pro-2 cs3219_otot_taskd % docker-compose up -d
 [+] Running 6/6
  ⠿ Container cs3219_otot_taskd-zookeeper1-1  Started                                              1.0s
  ⠿ Container cs3219_otot_taskd-zookeeper2-1  Started                                              1.0s
  ⠿ Container cs3219_otot_taskd-zookeeper3-1  Started                                              1.0s
  ⠿ Container cs3219_otot_taskd-kafka2-1      Started                                              2.1s
  ⠿ Container cs3219_otot_taskd-kafka3-1      Started                                              1.8s
  ⠿ Container cs3219_otot_taskd-kafka1-1      Started                                              2.1s
```



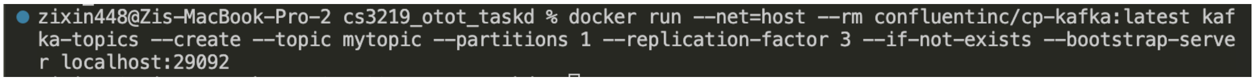
□	zookeeper3-1 bf1e360eb834	confluentinc/cp-zookeeper:late	Running	42181	7 minutes agc
□	zookeeper1-1 c2c8d879660c	confluentinc/cp-zookeeper:late	Running	22181	7 minutes agc
□	zookeeper2-1 76896b3abf0c	confluentinc/cp-zookeeper:late	Running	32181	7 minutes agc
□	kafka1-1 af0096db2b20	confluentinc/cp-kafka:latest	Running	29092	7 minutes agc
□	kafka3-1 0fde73c5c854	confluentinc/cp-kafka:latest	Running	49092	7 minutes agc
□	kafka2-1 630e115a5cef	confluentinc/cp-kafka:latest	Running	39092	7 minutes agc

7. Check that the services are running
    a. `docker-compose ps`
8. Pick a controller
    a. `sudo kcat -L -b kafka1:29092` (or kafka:39092 or kafka:49092)
    b. Enter password



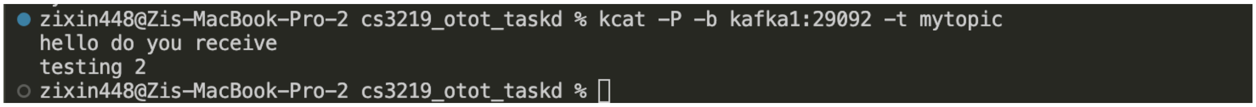
```
● zixin448@Zis-MacBook-Pro-2 cs3219_otot_taskd % sudo kcat -L -b kafka1:29092
 Metadata for all topics (from broker 1: kafka1:29092/1):
  3 brokers:
   broker 2 at kafka2:39092
   broker 3 at kafka3:49092 (controller)
   broker 1 at kafka1:29092
  0 topics:
```

9. Create a topic
    a. `docker run --net=host --rm confluentinc/cp-kafka:latest kafka-topics --create --topic mytopic --partitions 1 --replication-factor 3 --if-not-exists --bootstrap-server localhost:29092`

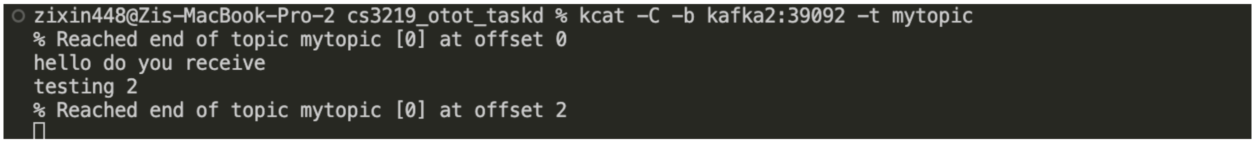


```
● zixin448@Zis-MacBook-Pro-2 cs3219_otot_taskd % docker run --net=host --rm confluentinc/cp-kafka:latest kaf
  ka-topics --create --topic mytopic --partitions 1 --replication-factor 3 --if-not-exists --bootstrap-serve
  r localhost:29092
```

10. Open new terminal and run a kafka server as producer
    a. `kcat -P -b kafka1:29092 -t mytopic`
    b. -P: flag for producer
    c. -t: topic name flag
    d. -b: broker chosen, in this case is kafka1 server
11. In a separate terminal, run another server as consumer
    a. `kcat -C -b kafka2:39092 -t mytopic`
    b. -C: flag for consumer
12. Check the Pub-Sub messaging between the two terminals
    a. Send messages at producer terminal, click `return` to enter next line
    b. When done, `Ctrl-D` to publish
    c. Visit consumer terminal to see messages published by producer



```
● zixin448@Zis-MacBook-Pro-2 cs3219_otot_taskd % kcat -P -b kafka1:29092 -t mytopic
  hello do you receive
  testing 2
○ zixin448@Zis-MacBook-Pro-2 cs3219_otot_taskd % 
```



```
○ zixin448@Zis-MacBook-Pro-2 cs3219_otot_taskd % kcat -C -b kafka2:39092 -t mytopic
  % Reached end of topic mytopic [0] at offset 0
  hello do you receive
  testing 2
  % Reached end of topic mytopic [0] at offset 2
```