Docker Lab2

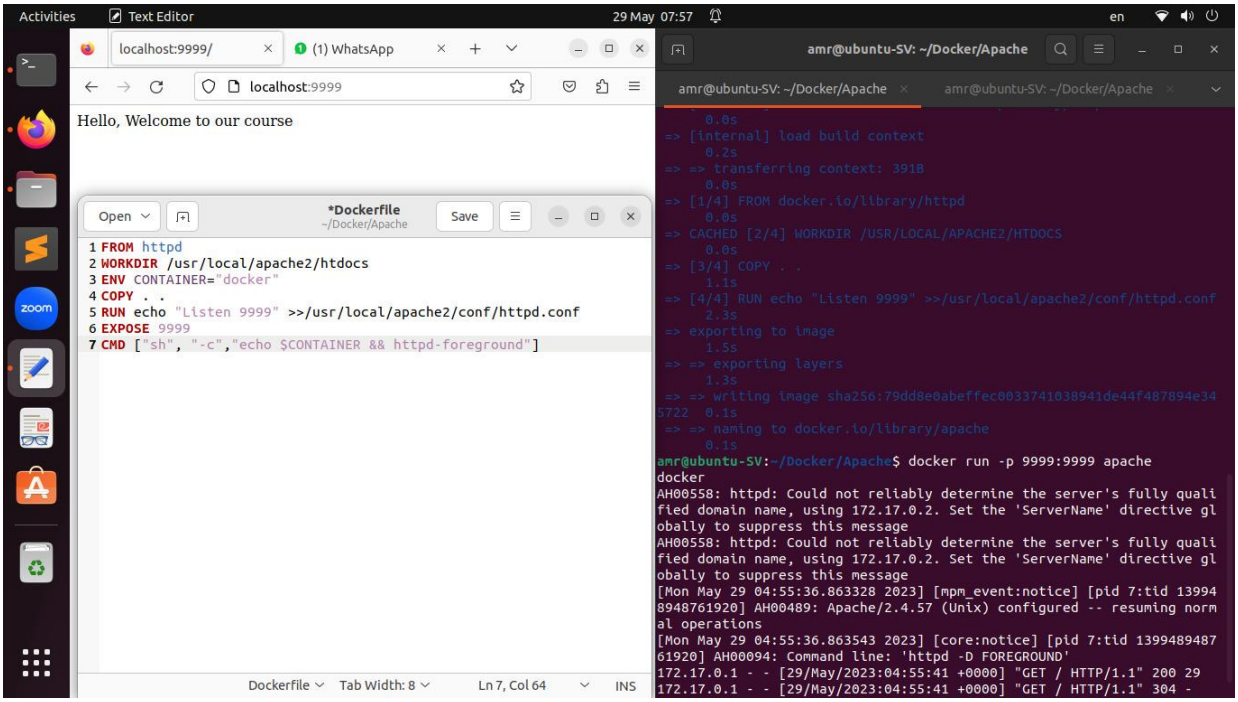
1. Problem 1:
   Create bridge network with subnet 192.168.0.0/24.
   Run 2 containers and attach containers to this network.
   Create another bridge network with subnet 10.5.0.0/24.
   Run any container and attach it to the new network.
   Make sure that the containers at different network can't ping each other

Sol:
1- create network net 1 and append ubuntu1 & ubuntu2 to net1
2- create network net 2 and append ubuntu3 to net2
3-try to ping from ubuntu1 to ubuntu2 success
4-try to ping from ubuntu1 to ubuntu3 fail

```
amr@ubuntu-SV:~/Docker$ docker network create --subnet=192.168.0.0/24 net1
91fb8812515e13da6447bf0929bfd18c141af701a49dc8cdde6a3eab2756cf6e
amr@ubuntu-SV:~/Docker$ docker run -d --network net1 --name ubuntu1 ubuntu sleep 1000
035f0083eb96feb3bb664356c883cb70d125cb2d4d1d1f3c6749c6e1189aa6e5
amr@ubuntu-SV:~/Docker$ docker run -d --network net1 --name ubuntu2 ubuntu sleep 1000
0f8723be5d0c5bcf3d3200fde70ed7c2248d23200a6a98d4df2304d512f57b3e
amr@ubuntu-SV:~/Docker$ docker network create --subnet=10.5.0.0/24 net2
393db3d9236a8f5fa68b880c23f7aba5bafd77a31e3061176fcd89304c8cbe4e
amr@ubuntu-SV:~/Docker$ docker run -d --network net2 --name ubuntu3 ubuntu sleep 1000
576371d2eb7fa85c9dd36a6350b2f0d2f5316fab296e936bef12f6b0d59440f5
amr@ubuntu-SV:~/Docker$

amr@ubuntu-SV:~$ docker exec -it ubuntu1 bash
root@035f0083eb96:/# ping 192.168.0.3
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
64 bytes from 192.168.0.3: icmp_seq=1 ttl=64 time=0.145 ms
64 bytes from 192.168.0.3: icmp_seq=2 ttl=64 time=0.070 ms
64 bytes from 192.168.0.3: icmp_seq=3 ttl=64 time=0.082 ms
64 bytes from 192.168.0.3: icmp_seq=4 ttl=64 time=0.096 ms
64 bytes from 192.168.0.3: icmp_seq=5 ttl=64 time=0.124 ms
64 bytes from 192.168.0.3: icmp_seq=6 ttl=64 time=0.138 ms
64 bytes from 192.168.0.3: icmp_seq=7 ttl=64 time=0.136 ms
^C
--- 192.168.0.3 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6132ms
rtt min/avg/max/mdev = 0.070/0.113/0.145/0.027 ms
root@035f0083eb96:/# ping 10.5.0.2
PING 10.5.0.2 (10.5.0.2) 56(84) bytes of data.
^C
--- 10.5.0.2 ping statistics ---
18 packets transmitted, 0 received, 100% packet loss, time 17404ms
```

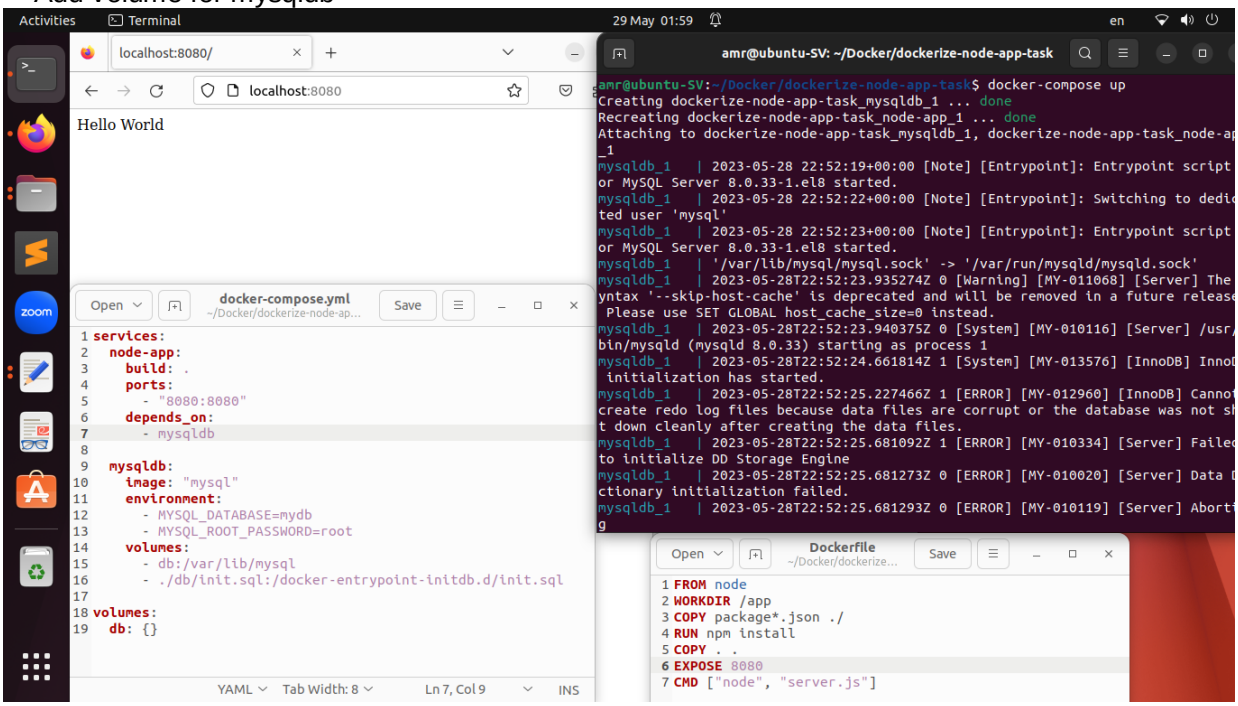
2. Problem 2:
   Create static html file
   Write Dockerfile to build image based on httpd to host the html file and specify the following
   Copy the html file.
   Copy a new configuration file to listen on port 9999 instead of 80
   Open the port 9999 in the container
   Add environment variable CONTAINER with value docker .
   Add startup command to echo the variable



3. Problem 3:

Create a docker compose to up mysql container, and https://github.com/sabreensalama/dockerize-node-app-task which depend on mysqldb.

Add volume for mysqldb



4. Problem4:

Use docker compose to deploy ghost platform (image: ghost:1-alpine)(Ghost is a free and open source blogging platform written in JavaScript)

Use mysql database instead of sqlite