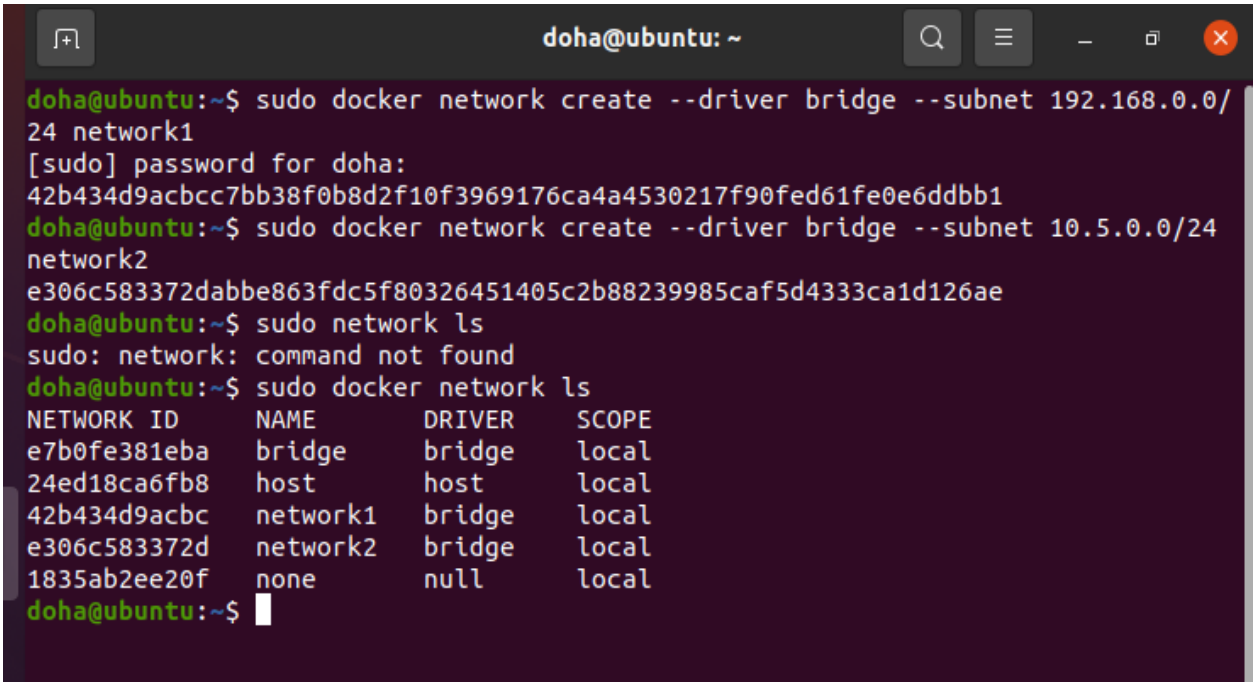
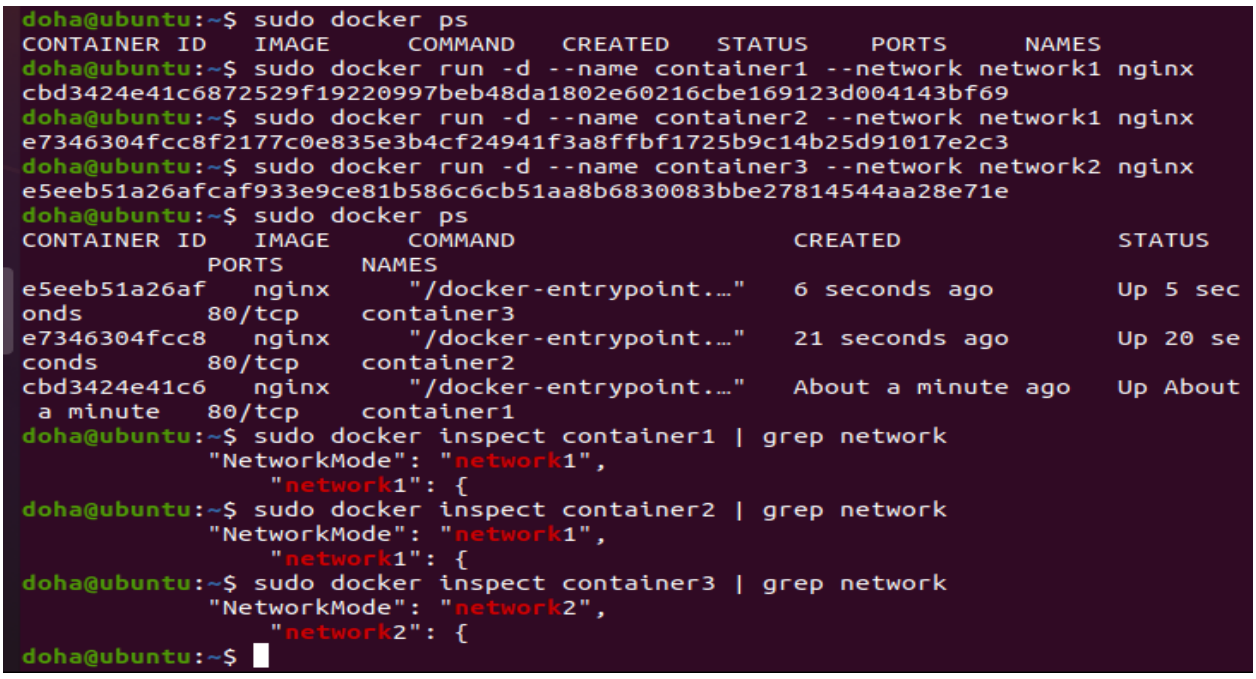
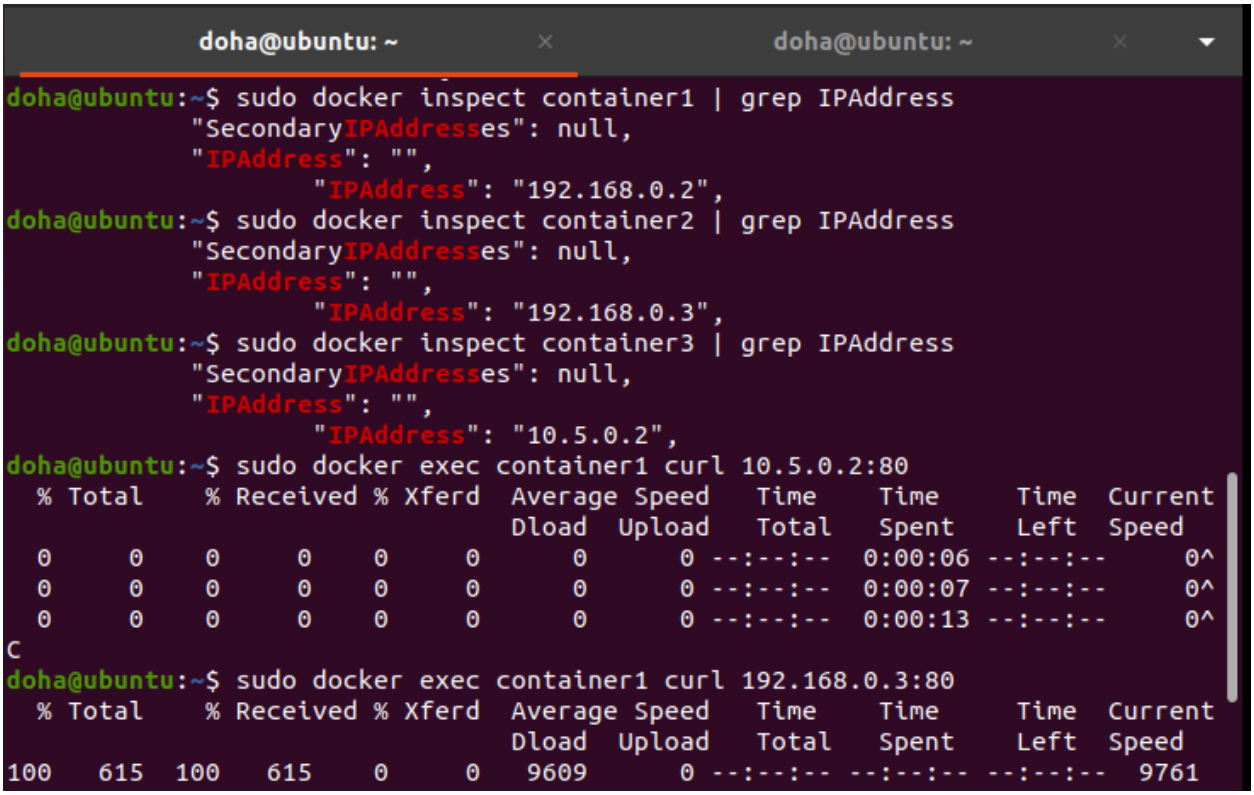
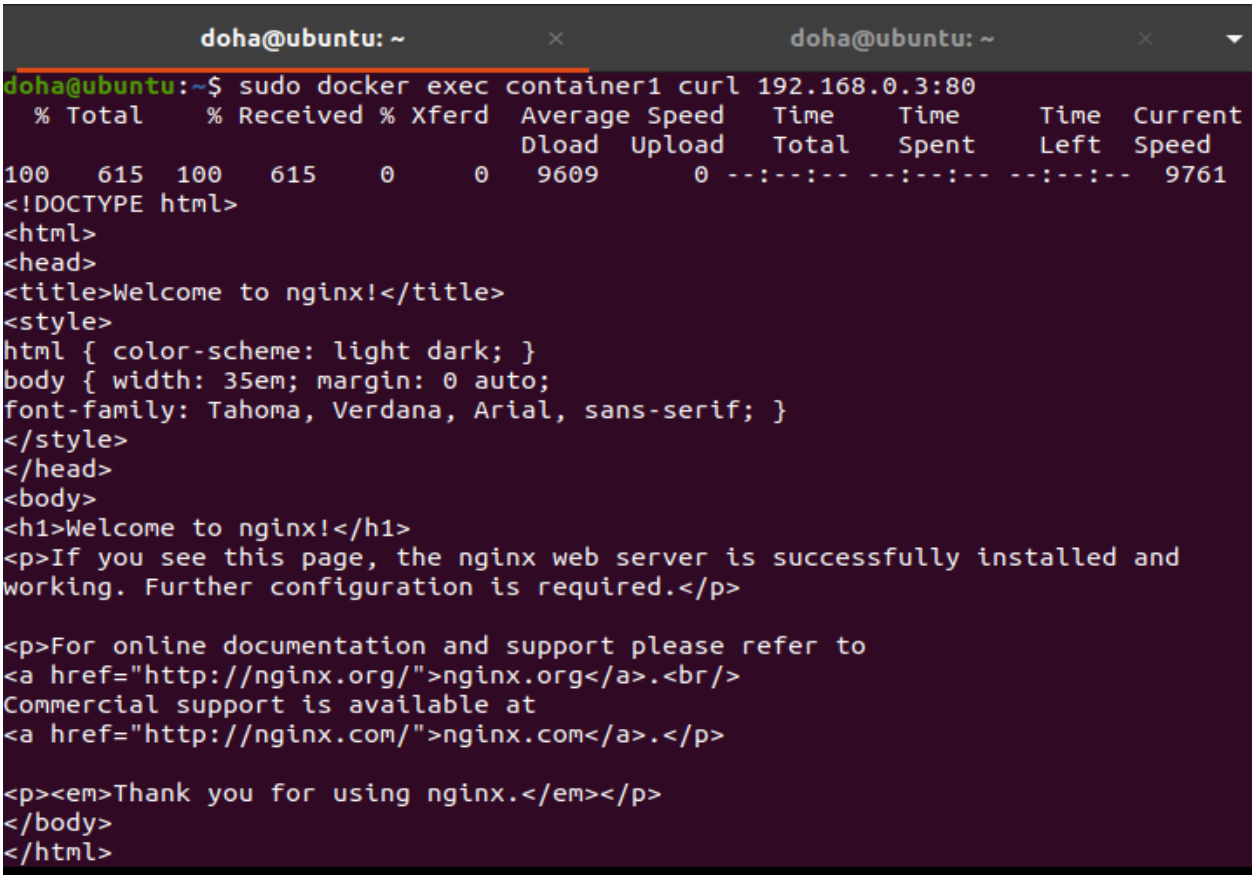
Create bridge network with subnet 192.168.0.0/24. Run 2 containers and attach containers to this network. Create another bridge network with subnet 10.5.0.0/24. Run any container and attach it to the new network. Make sure that the containers at different network can't ping each other

In this problem I used RUN apt-get update && apt-get install -y iputils-ping command inside the nginx container to be able to ping on the redis(ip=10.5.0.2) container.






doha@ubuntu: ~
doha@ubuntu: ~
doha@ubuntu:~$ sudo docker inspect container1 | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "",
"IPAddress": "192.168.0.2",
doha@ubuntu:~$ sudo docker inspect container2 | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "",
"IPAddress": "192.168.0.3",
doha@ubuntu:~$ sudo docker inspect container3 | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "",
"IPAddress": "10.5.0.2",
doha@ubuntu:~$ sudo docker exec container1 curl 10.5.0.2:80
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- 0:00:06 --:--:-- 0^
0 0 0 0 0 0 0 0 --:--:-- 0:00:07 --:--:-- 0^
0 0 0 0 0 0 0 0 --:--:-- 0:00:13 --:--:-- 0^
C
doha@ubuntu:~$ sudo docker exec container1 curl 192.168.0.3:80
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 615 100 615 0 0 9609 0 --:--:-- --:--:-- --:--:-- 9761


doha@ubuntu: ~
doha@ubuntu: ~
doha@ubuntu:~$ sudo docker exec container1 curl 192.168.0.3:80
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 615 100 615 0 0 9609 0 --:--:-- --:--:-- --:--:-- 9761
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.
For online documentation and support please refer to
nginx.org.
Commercial support is available at
nginx.com.
Thank you for using nginx.
</body>
</html>

Problem 2:

- Create static html file >> index.html

```
Open   index.html ~/proplem2   Save
Dockerfile | index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Lab 2</title>
5 </head>
6 <body>
7 <h1>from web-app</h1>
8 </body>
9 </html>
HTML  Tab Width: 8  Ln 1, Col 1  INS
```

- Write Dockerfile to build image based on httpd to host the html file >> Dockerfile

```
Open   Dockerfile ~/proplem2   Save
1 FROM httpd:2.4
2
3 COPY index.html /usr/local/apache2/htdocs/
4
5 COPY httpd.conf /usr/local/apache2/conf/httpd.conf
6
7 EXPOSE 9999
8
9 ENV CONTAINER="DOCKER"
10
11 CMD ["sh", "-c", "echo $CONTAINER && httpd-foreground"]
12
Dockerfile  Tab Width: 8  Ln 11, Col 38  INS
```

- Copy a new configuration file to listen on port 9999 instead of 80 >> [httpd.conf]

Open | httpd.conf ~/proplem2 | Save

```
Listen 9999
ServerName localhost

ServerRoot "/usr/local/apache2"
DocumentRoot "/usr/local/apache2/htdocs"

LoadModule mpm_prefork_module modules/mod_mpm_prefork.so
LoadModule authn_core_module modules/mod_authn_core.so
LoadModule authz_core_module modules/mod_authz_core.so
LoadModule dir_module modules/mod_dir.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
LoadModule unixd_module modules/mod_unixd.so

<Directory />
    AllowOverride none
    Require all denied
</Directory>

<Directory "/usr/local/apache2/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

ErrorLog "logs/error_log"
LogLevel warn

```

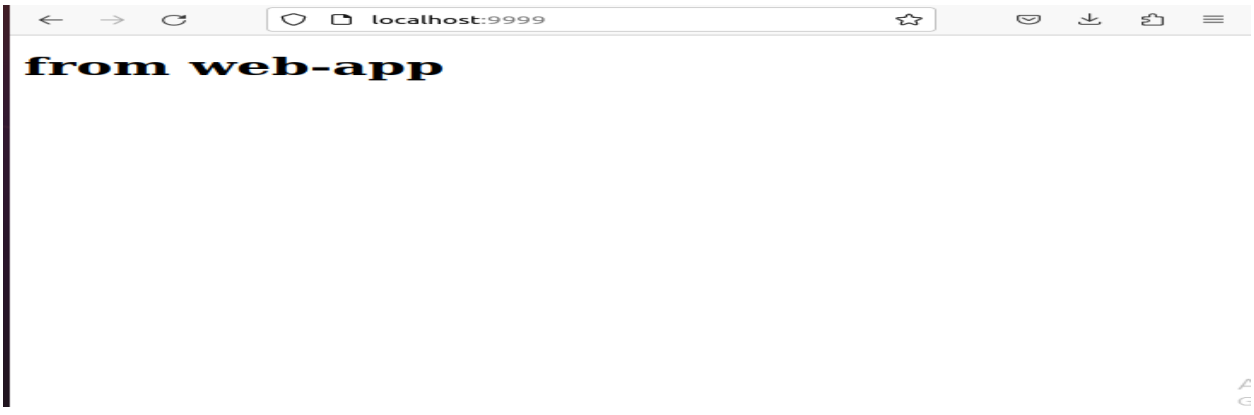
Plain Text | Tab Width: 8 | Ln 32, Col 12 | INS

- Build Dockerfile

```
doha@ubuntu:~/proplem2$ sudo docker build -t my-httpd-image .
[sudo] password for doha:
Sending build context to Docker daemon  30.21kB
Step 1/6 : FROM httpd:2.4
 ---> d1676199e605
Step 2/6 : COPY index.html /usr/local/apache2/htdocs/
 ---> 09f9902c2305
Step 3/6 : COPY httpd.conf /usr/local/apache2/conf/httpd.conf
 ---> 7875e6e3e8c8
Step 4/6 : EXPOSE 9999
 ---> Running in 4884755270cd
Removing intermediate container 4884755270cd
 ---> e7ae08738560
Step 5/6 : ENV CONTAINER="DOCKER"
 ---> Running in da819429b5ab
Removing intermediate container da819429b5ab
 ---> 6d2b431ff7d5
Step 6/6 : CMD ["sh", "-c", "echo $CONTAINER && httpd-foreground"]
 ---> Running in efd4af5dde4e
Removing intermediate container efd4af5dde4e
 ---> ce7178ada36b
Successfully built ce7178ada36b
Successfully tagged my-httpd-image:latest
```

```
doha@ubuntu:~/proplem2$ sudo docker images
REPOSITORY                TAG             IMAGE ID       CREATED          SIZE
my-httpd-image            latest          ce7178ada36b   2 minutes ago    145MB
apache                    latest          274fac7e8284   26 hours ago     145MB
simple-flash-app          latest          dc1151128dc8   3 days ago       61MB
nginx                     alpine          fe7edaf8a8dc   6 days ago       41.4MB
mysql                     latest          05db07cd74c0   6 days ago       565MB
httpd                     2.4             d1676199e605   7 days ago       145MB
python                    3.9-alpine      3e25ee8eaf10   2 weeks ago      48.2MB
hello-world               latest          9c7a54a9a43c   3 weeks ago      13.3kB
nginx                     latest          448a08f1d2f9   3 weeks ago      142MB
ubuntu                    latest          3b418d7b466a   5 weeks ago      77.8MB
nginx                     1.24.0-alpine   1266a3a46e96   7 weeks ago      41.1MB
doha                      2.0             0dcda3f22e52   3 months ago     119MB
doha                      3.0             0dcda3f22e52   3 months ago     119MB
doha                      latest          0dcda3f22e52   3 months ago     119MB
image1                    1.0             0dcda3f22e52   3 months ago     119MB
ubuntu                    <none>          58db3edaf2be   4 months ago     77.8MB
nginx                     1.14-alpine     8a2fb25a19f5   4 years ago      16MB
kodekloud/simple-webapp   latest          c6e3cd9aae36   4 years ago      84.8MB
doha@ubuntu:~/proplem2$ sudo docker run -p 9999:9999 --name my-httpd-container
my-httpd-image
DOCKER
```

```
doha@ubuntu:~/proplem2$ sudo docker ps -a
[sudo] password for doha:
CONTAINER ID   IMAGE                     COMMAND                  CREATED
       STATUS                     PORTS
        NAMES
bbccbf911dfe   my-httpd-image            "sh -c 'echo $CONTAI…"   About an hour
 ago   Up About an hour            80/tcp, 0.0.0.0:9999->9999/tcp, :::9999->999
9/tcp   my-httpd-container
19821fc24cbc   apache                    " httpd-foreground"      27 hours ago
       Created                    80/tcp, 0.0.0.0:9999->9999/tcp, :::9999->999
```



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

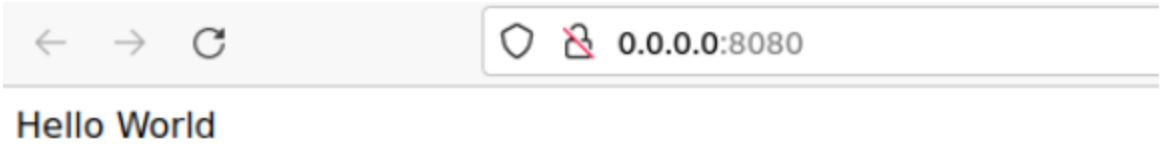
Open | docker-compose.yml ~/problem3 docker | Save

```yaml
version: "3"
services:
  db:
    image: mysql:8.0
    environment:
      MYSQL_ROOT_PASSWORD: heba
      MYSQL_DATABASE: heba-db
    volumes:
      - db_data:/var/lib/mysql

  app:
    build: .
    depends_on:
      - db
    ports:
      - "8080:8080"
    volumes:
      - app_data:/usr/src/app/data

volumes:
  db_data:
  app_data:
```

YAML | Tab Width: 8 | Ln 1, Col 12 | INS

doha@ubuntu: ~/problem3 docker/dockerize-node-app-task

```
doha@ubuntu:~/problem3 docker/dockerize-node-app-task$ sudo docker-compose up
Creating network "problem3docker_default" with the default driver
Creating volume "problem3docker_db_data" with default driver
Creating volume "problem3docker_app_data" with default driver
Pulling db (mysql:8.0)...
8.0: Pulling from library/mysql
90e2fb2facff: Pull complete
ba60eb20fd5f: Pull complete
4f509402d469: Pull complete
496c2cfa6815: Pull complete
8ec1dfa9522c: Pull complete
6dec7ba896f8: Pull complete
dc9ff75362b0: Pull complete
73e4682f9014: Pull complete
9ffdeecd6fb6: Pull complete
a4346ccfb53f: Pull complete
434c13bc32de: Pull complete
Digest: sha256:d6164ff4855b9b3f2c7748c6ec564ccff841f79a7023db0f9293143481a44b6e
Status: Downloaded newer image for mysql:8.0
Building app
Step 1/7 : FROM node:16
16: Pulling from library/node
c722db24a050: Pull complete
a3f8e4a0ed53: Pull complete
a739c67a76c3: Pull complete
51e0d706266f: Pull complete
f38700a2e11e: Pull complete
b84b1175afa3: Pull complete
f36686a8f4d2: Pull complete
```



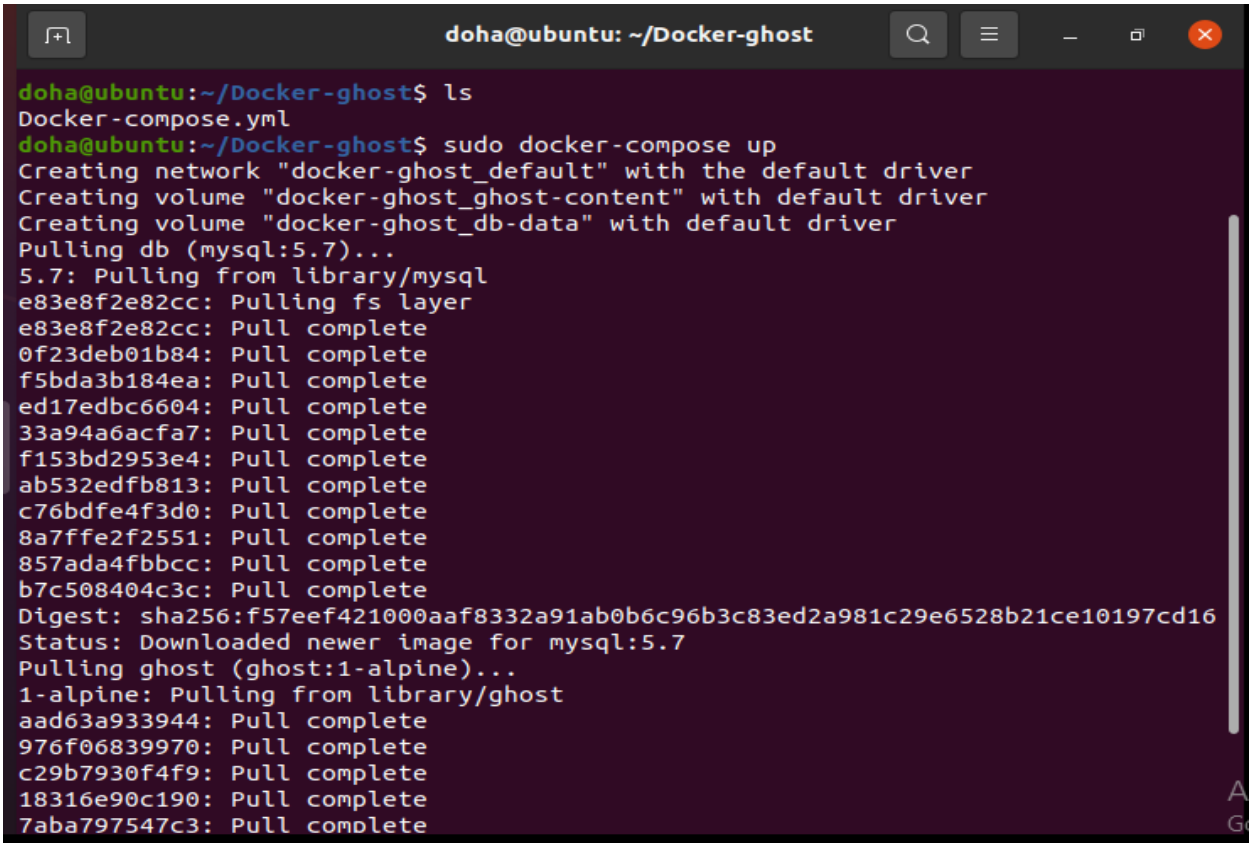
Problem 4:

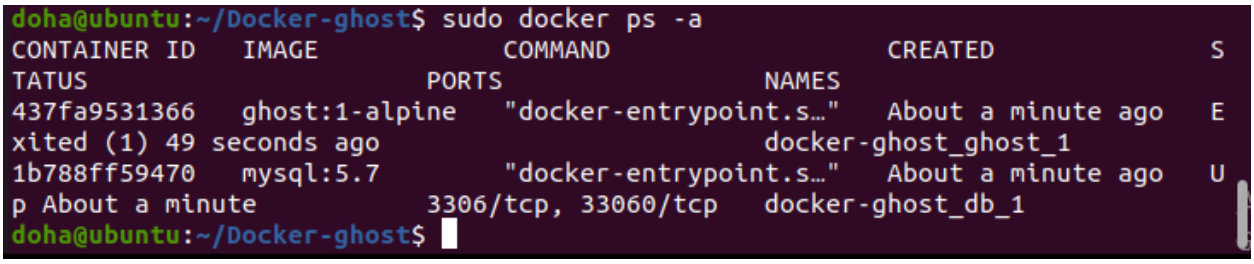
Use docker compose to deploy ghost platform (image: ghost:1-alpine)(Ghost is a free and open source blogging platform written in JavaScript) Use mysql database instead of sqlite >> [docker-compose.yml]

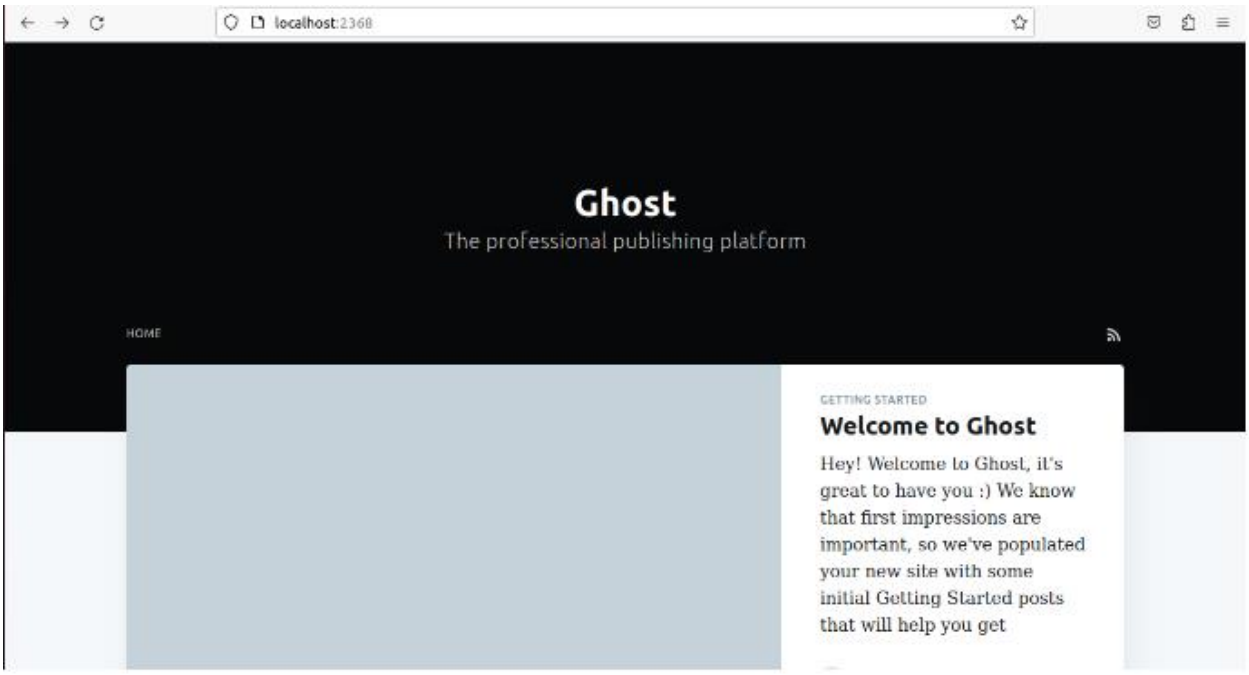
Docker-compose.yml | docker-compose.yml

```yaml
version: '3'
services:
  ghost:
    image: "ghost:1-alpine"
    ports:
      - "2368:2368"
    environment:
      database__client: mysql
      database__connection__host: db
      database__connection__user: root
      database__connection__password: heba
      database__connection__database: ghost
    depends_on:
      - db
    volumes:
      - ghost-content:/var/lib/ghost/content

  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: heba
      MYSQL_DATABASE: ghost
    volumes:
      - db-data:/var/lib/mysql

volumes:
```

YAML | Tab Width: 8 | Ln 11, Col 41 | INS


doha@ubuntu: ~/Docker-ghost
doha@ubuntu:~/Docker-ghost$ ls
Docker-compose.yml
doha@ubuntu:~/Docker-ghost$ sudo docker-compose up
Creating network "docker-ghost_default" with the default driver
Creating volume "docker-ghost_ghost-content" with default driver
Creating volume "docker-ghost_db-data" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
e83e8f2e82cc: Pulling fs layer
e83e8f2e82cc: Pull complete
0f23deb01b84: Pull complete
f5bda3b184ea: Pull complete
ed17edbc6604: Pull complete
33a94a6acfa7: Pull complete
f153bd2953e4: Pull complete
ab532edfb813: Pull complete
c76bdfe4f3d0: Pull complete
8a7ffe2f2551: Pull complete
857ada4fbbcc: Pull complete
b7c508404c3c: Pull complete
Digest: sha256:f57eef421000aaf8332a91ab0b6c96b3c83ed2a981c29e6528b21ce10197cd16
Status: Downloaded newer image for mysql:5.7
Pulling ghost (ghost:1-alpine)...
1-alpine: Pulling from library/ghost
aad63a933944: Pull complete
976f06839970: Pull complete
c29b7930f4f9: Pull complete
18316e90c190: Pull complete
7aba797547c3: Pull complete


doha@ubuntu:~/Docker-ghost$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED S
TATUS PORTS NAMES
437fa9531366 ghost:1-alpine "docker-entrypoint.s…" About a minute ago E
xited (1) 49 seconds ago docker-ghost_ghost_1
1b788ff59470 mysql:5.7 "docker-entrypoint.s…" About a minute ago U
p About a minute 3306/tcp, 33060/tcp docker-ghost_db_1
doha@ubuntu:~/Docker-ghost$


localhost:2368
Ghost
The professional publishing platform
HOME
GETTING STARTED
Welcome to Ghost
Hey! Welcome to Ghost, it's great to have you :) We know that first impressions are important, so we've populated your new site with some initial Getting Started posts that will help you get