Docker Lab3

1. Problem 1:

Create bridge network with subnet 192.168.0.0/24.

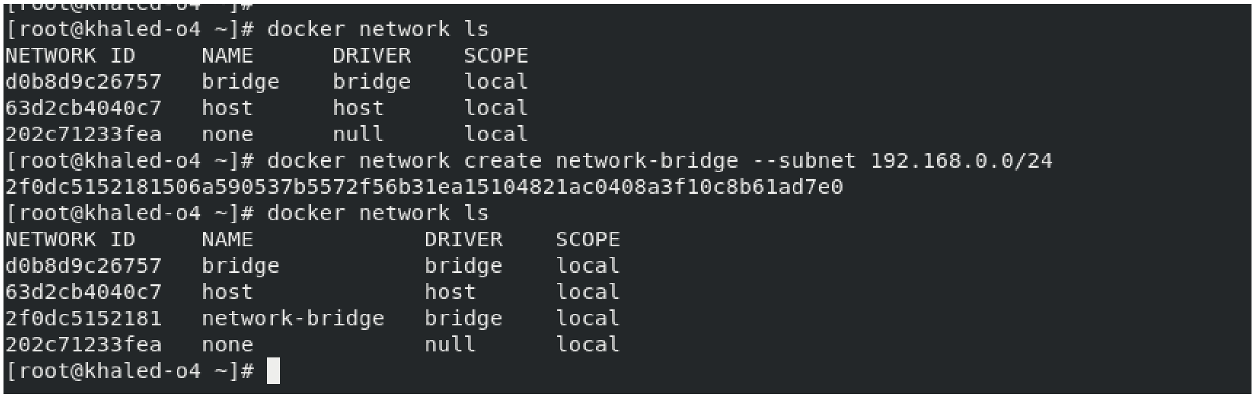
Run 2 containers and attach containers to this network.

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

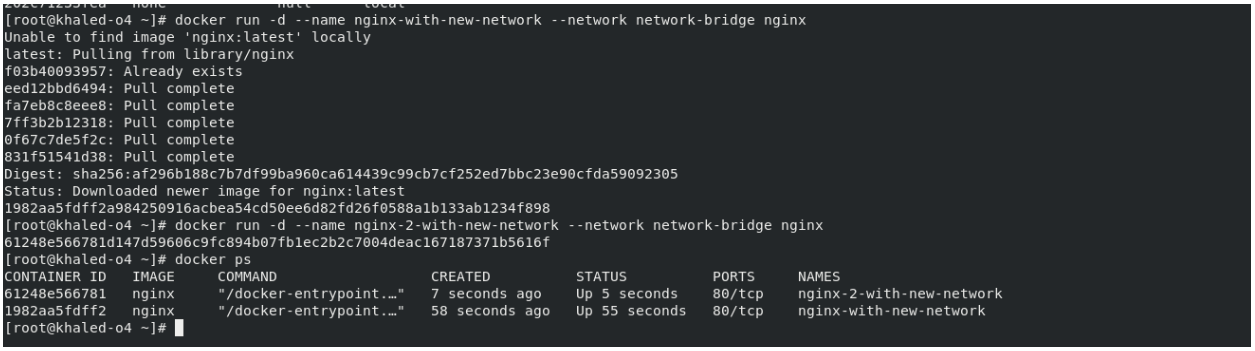
docker network ls



```
[root@khaled-o4 ~]# docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
d0b8d9c26757   bridge    bridge    local
63d2cb4040c7   host      host      local
202c71233fea   none      null      local
[root@khaled-o4 ~]# docker network create network-bridge --subnet 192.168.0.0/24
2f0dc5152181506a590537b5572f56b31ea15104821ac0408a3f10c8b61ad7e0
[root@khaled-o4 ~]# docker network ls
NETWORK ID     NAME             DRIVER    SCOPE
d0b8d9c26757   bridge           bridge    local
63d2cb4040c7   host             host      local
2f0dc5152181   network-bridge   bridge    local
202c71233fea   none             null      local
[root@khaled-o4 ~]# 
```

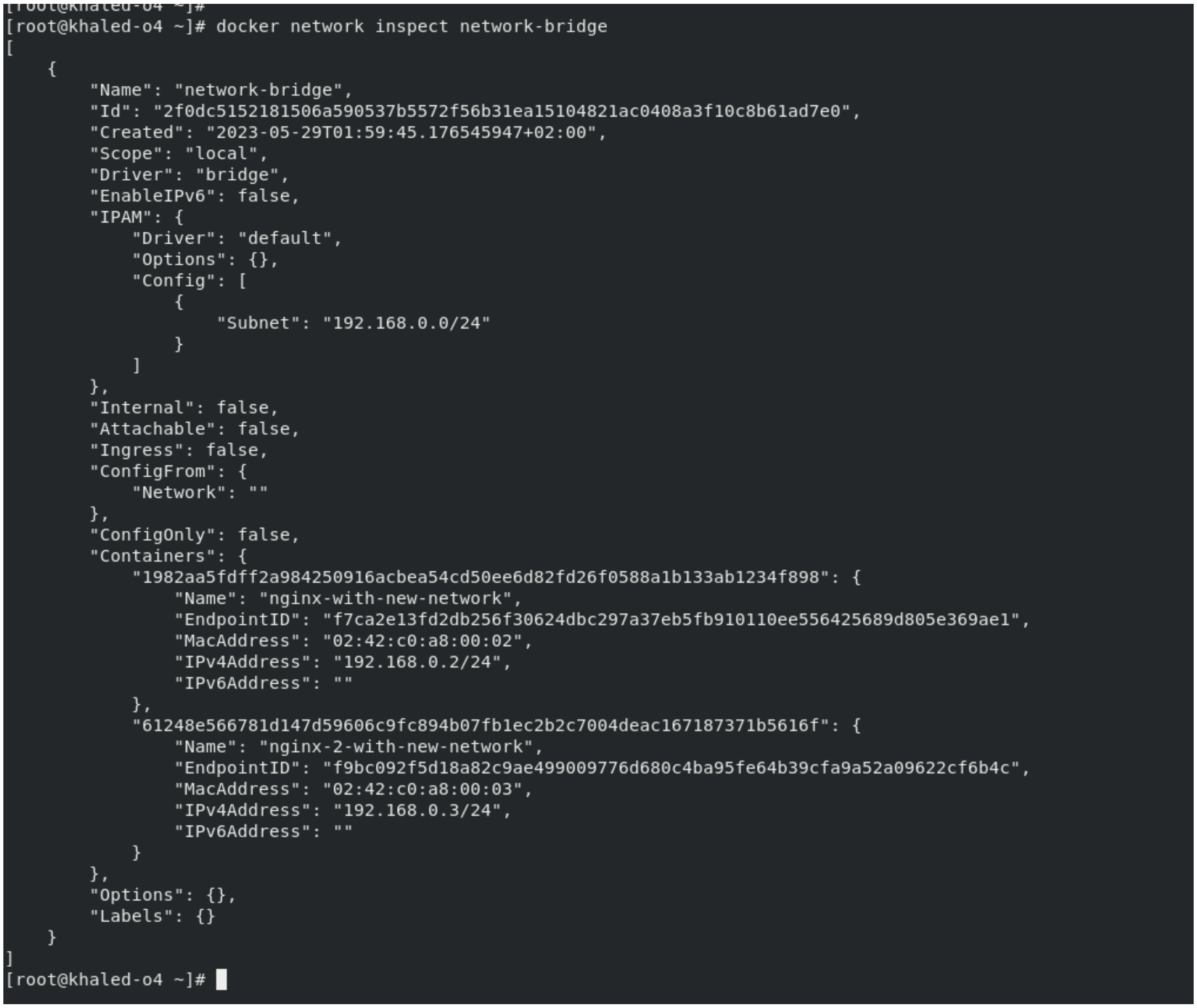
docker run -d --name nginx-with-new-network --network network-bridge nginx

docker ps



```
[root@khaled-o4 ~]# docker run -d --name nginx-with-new-network --network network-bridge nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
f03b40093957: Already exists
eed12bbd6494: Pull complete
fa7eb8c8eee8: Pull complete
7ff3b2b12318: Pull complete
0f67c7de5f2c: Pull complete
831f51541d38: Pull complete
Digest: sha256:af296b188c7b7df99ba960ca614439c99cb7cf252ed7bbc23e90cfda59092305
Status: Downloaded newer image for nginx:latest
1982aa5fdff2a984250916acbea54cd50ee6d82fd26f0588a1b133ab1234f898
[root@khaled-o4 ~]# docker run -d --name nginx-2-with-new-network --network network-bridge nginx
61248e566781d147d59606c9fc894b07fb1ec2b2c7004deac167187371b5616f
[root@khaled-o4 ~]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
61248e566781   nginx     "/docker-entrypoint.…"   7 seconds ago    Up 5 seconds    80/tcp    nginx-2-with-new-network
1982aa5fdff2   nginx     "/docker-entrypoint.…"   58 seconds ago   Up 55 seconds   80/tcp    nginx-with-new-network
[root@khaled-o4 ~]# 
```

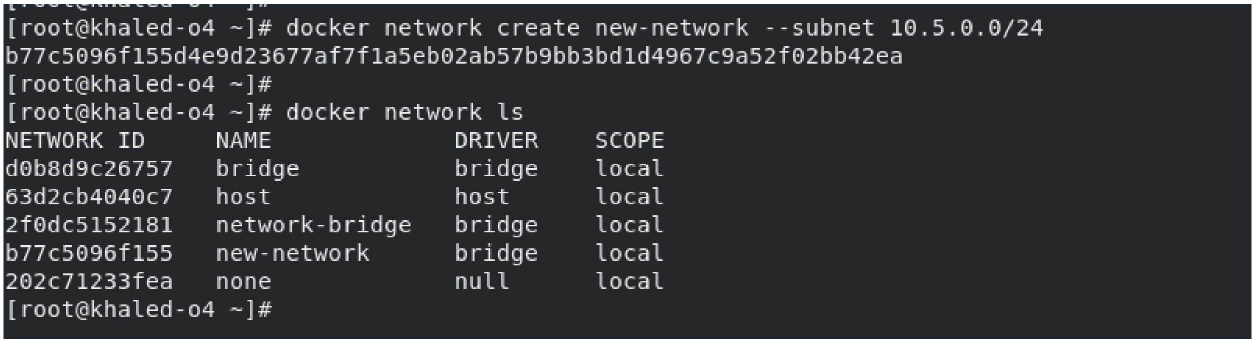
docker network inspect network-bridge



```
[root@khaled-o4 ~]#
[root@khaled-o4 ~]# docker network inspect network-bridge
[
    {
        "Name": "network-bridge",
        "Id": "2f0dc5152181506a590537b5572f56b31ea15104821ac0408a3f10c8b61ad7e0",
        "Created": "2023-05-29T01:59:45.176545947+02:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "192.168.0.0/24"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "1982aa5fdff2a984250916acbea54cd50ee6d82fd26f0588a1b133ab1234f898": {
                "Name": "nginx-with-new-network",
                "EndpointID": "f7ca2e13fd2db256f30624dbc297a37eb5fb910110ee556425689d805e369ae1",
                "MacAddress": "02:42:c0:a8:00:02",
                "IPv4Address": "192.168.0.2/24",
                "IPv6Address": ""
            },
            "61248e566781d147d59606c9fc894b07fb1ec2b2c7004deac167187371b5616f": {
                "Name": "nginx-2-with-new-network",
                "EndpointID": "f9bc092f5d18a82c9ae499009776d680c4ba95fe64b39cfa9a52a09622cf6b4c",
                "MacAddress": "02:42:c0:a8:00:03",
                "IPv4Address": "192.168.0.3/24",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
[root@khaled-o4 ~]#
```

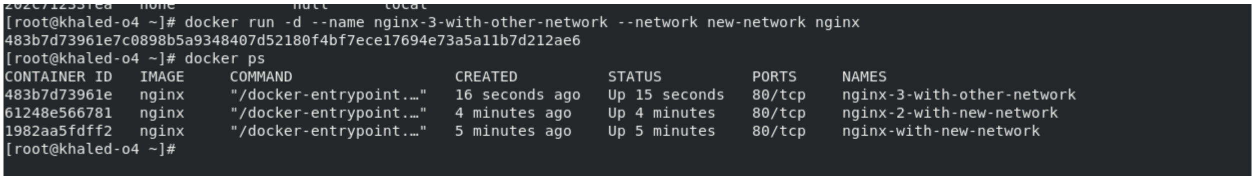
docker network create new-network --subnet 10.5.0.0/24

docker network ls



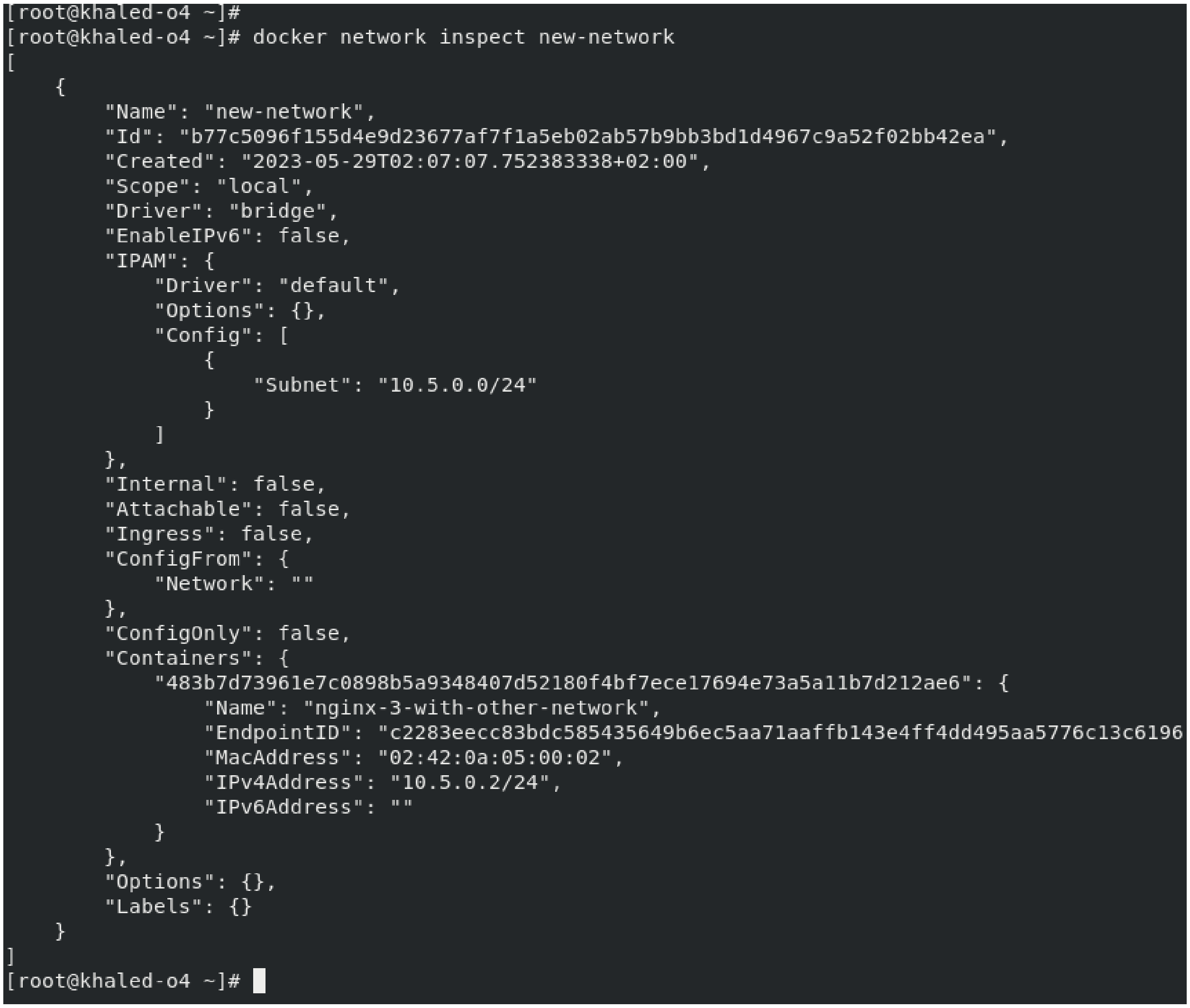
```
[root@khaled-o4 ~]#
[root@khaled-o4 ~]# docker network create new-network --subnet 10.5.0.0/24
b77c5096f155d4e9d23677af7f1a5eb02ab57b9bb3bd1d4967c9a52f02bb42ea
[root@khaled-o4 ~]#
[root@khaled-o4 ~]# docker network ls
NETWORK ID     NAME             DRIVER    SCOPE
d0b8d9c26757   bridge           bridge    local
63d2cb4040c7   host             host      local
2f0dc5152181   network-bridge   bridge    local
b77c5096f155   new-network      bridge    local
202c71233fea   none             null      local
[root@khaled-o4 ~]#
```

docker run -d --name nginx-3-with-other-network --network new-network nginx



```
202c71235fea   none             null      local
[root@khaled-o4 ~]# docker run -d --name nginx-3-with-other-network --network new-network nginx
483b7d73961e7c0898b5a9348407d52180f4bf7ece17694e73a5a11b7d212ae6
[root@khaled-o4 ~]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
483b7d73961e   nginx     "/docker-entrypoint.…"   16 seconds ago   Up 15 seconds   80/tcp    nginx-3-with-other-network
61248e566781   nginx     "/docker-entrypoint.…"   4 minutes ago    Up 4 minutes    80/tcp    nginx-2-with-new-network
1982aa5fdff2   nginx     "/docker-entrypoint.…"   5 minutes ago    Up 5 minutes    80/tcp    nginx-with-new-network
[root@khaled-o4 ~]#
```

docker network inspect new-network



```
[root@khaled-o4 ~]#
[root@khaled-o4 ~]# docker network inspect new-network
[
    {
        "Name": "new-network",
        "Id": "b77c5096f155d4e9d23677af7f1a5eb02ab57b9bb3bd1d4967c9a52f02bb42ea",
        "Created": "2023-05-29T02:07:07.752383338+02:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "10.5.0.0/24"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "483b7d73961e7c0898b5a9348407d52180f4bf7ece17694e73a5a11b7d212ae6": {
                "Name": "nginx-3-with-other-network",
                "EndpointID": "c2283eecc83bdc585435649b6ec5aa71aaffb143e4ff4dd495aa5776c13c6196
                "MacAddress": "02:42:0a:05:00:02",
                "IPv4Address": "10.5.0.2/24",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
[root@khaled-o4 ~]#
```

2. Problem 2:

Create static html file

Write Dockerfile to build image based on httpd to host the html file and specify the following

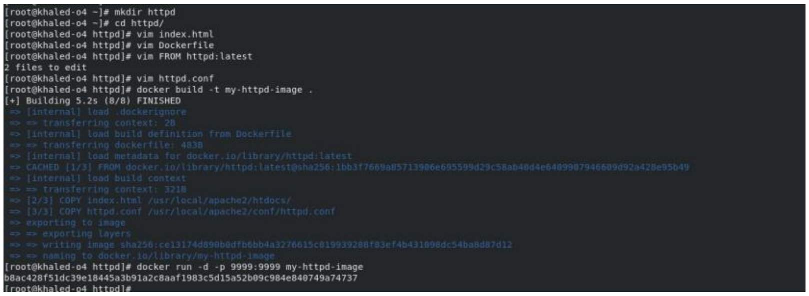
Copy the html file.

Copy a new configuration file to listen on port 9999 instead of 80

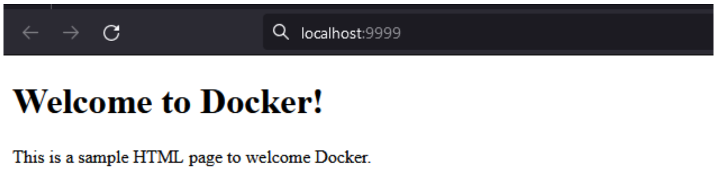
Open the port 9999 in the container

Add environment variable CONTAINER with value docker .

Add startup command to echo the variable

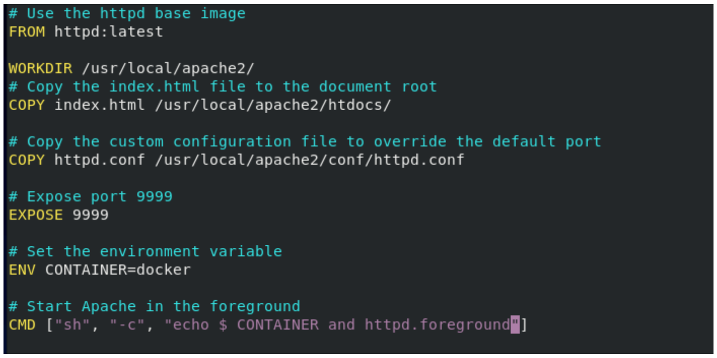


```
[root@khaled-o4 ~]# mkdir httpd
[root@khaled-o4 ~]# cd httpd/
[root@khaled-o4 httpd]# vim index.html
[root@khaled-o4 httpd]# vim Dockerfile
[root@khaled-o4 httpd]# vim FROM httpd:latest
2 files to edit
[root@khaled-o4 httpd]# vim httpd.conf
[root@khaled-o4 httpd]# docker build -t my-httpd-image .
[+] Building 5.2s (8/8) FINISHED
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 483B
 => [internal] load metadata for docker.io/library/httpd:latest
 => CACHED [1/3] FROM docker.io/library/httpd:latest@sha256:1bb3f7669a85713986e695599d29c58ab40d4e6409907946609d92a428e95b49
 => [internal] load build context
 => => transferring context: 321B
 => [2/3] COPY index.html /usr/local/apache2/htdocs/
 => [3/3] COPY httpd.conf /usr/local/apache2/conf/httpd.conf
 => exporting to image
 => => exporting layers
 => => writing image sha256:ce13174d890b0dfb6bb4a3276615c819939288f83ef4b431098dc54ba8d87d12
 => => naming to docker.io/library/my-httpd-image
[root@khaled-o4 httpd]# docker run -d -p 9999:9999 my-httpd-image
b8ac428f51dc39e18445a3b91a2c8aaf1983c5d15a52b09c984e840749a74737
[root@khaled-o4 httpd]#
```



Welcome to Docker!

This is a sample HTML page to welcome Docker.



```
# Use the httpd base image
FROM httpd:latest

WORKDIR /usr/local/apache2/
# Copy the index.html file to the document root
COPY index.html /usr/local/apache2/htdocs/

# Copy the custom configuration file to override the default port
COPY httpd.conf /usr/local/apache2/conf/httpd.conf

# Expose port 9999
EXPOSE 9999

# Set the environment variable
ENV CONTAINER=docker

# Start Apache in the foreground
CMD ["sh", "-c", "echo $ CONTAINER and httpd.foreground"]
```

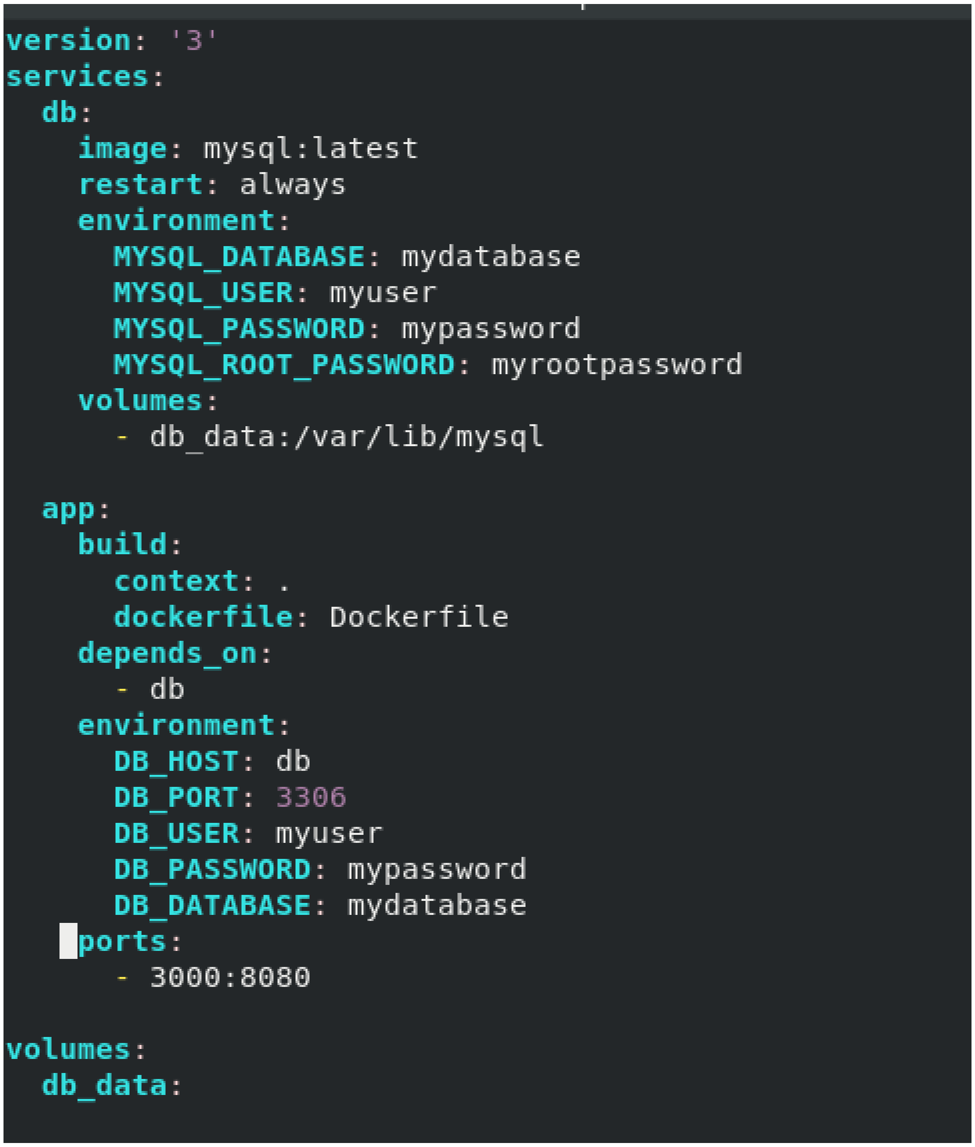
3. Problem 3:

Create a docker compose to up mysql container, and

https://github.com/sabreensalama/dockerize-node-app-task which depend on mysqldb.

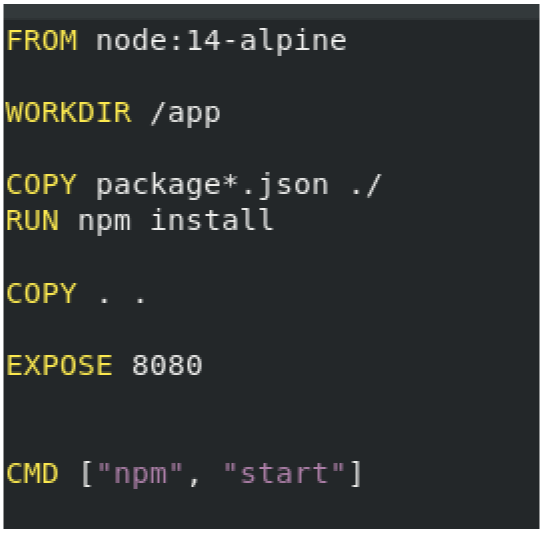
Add volume for mysqldb

vim docker-compose.yml



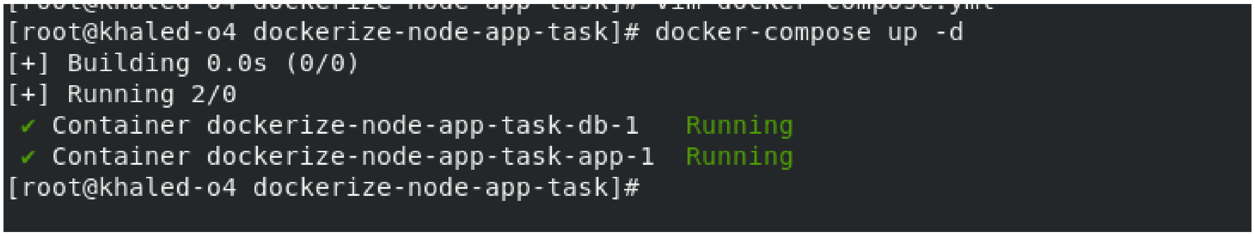
```
version: '3'
services:
  db:
    image: mysql:latest
    restart: always
    environment:
      MYSQL_DATABASE: mydatabase
      MYSQL_USER: myuser
      MYSQL_PASSWORD: mypassword
      MYSQL_ROOT_PASSWORD: myrootpassword
    volumes:
      - db_data:/var/lib/mysql

  app:
    build:
      context: .
      dockerfile: Dockerfile
    depends_on:
      - db
    environment:
      DB_HOST: db
      DB_PORT: 3306
      DB_USER: myuser
      DB_PASSWORD: mypassword
      DB_DATABASE: mydatabase
    ports:
      - 3000:8080

volumes:
  db_data:
```

vim Dockerfile

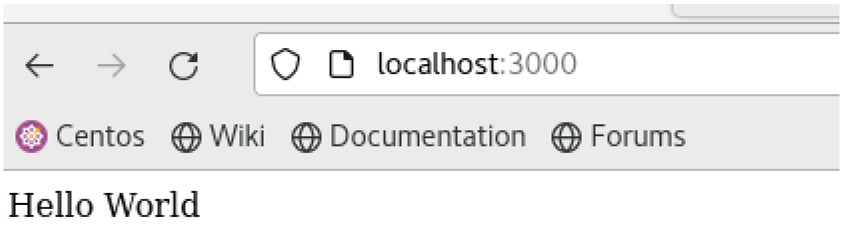


```
FROM node:14-alpine

WORKDIR /app

COPY package*.json ./
RUN npm install

COPY . .

EXPOSE 8080


CMD ["npm", "start"]
```

docker-compose up -d



```
[root@khaled-o4 dockerize-node-app-task]# vim docker-compose.yml
[root@khaled-o4 dockerize-node-app-task]# docker-compose up -d
[+] Building 0.0s (0/0)
[+] Running 2/0
 ✔ Container dockerize-node-app-task-db-1   Running
 ✔ Container dockerize-node-app-task-app-1  Running
[root@khaled-o4 dockerize-node-app-task]#
```

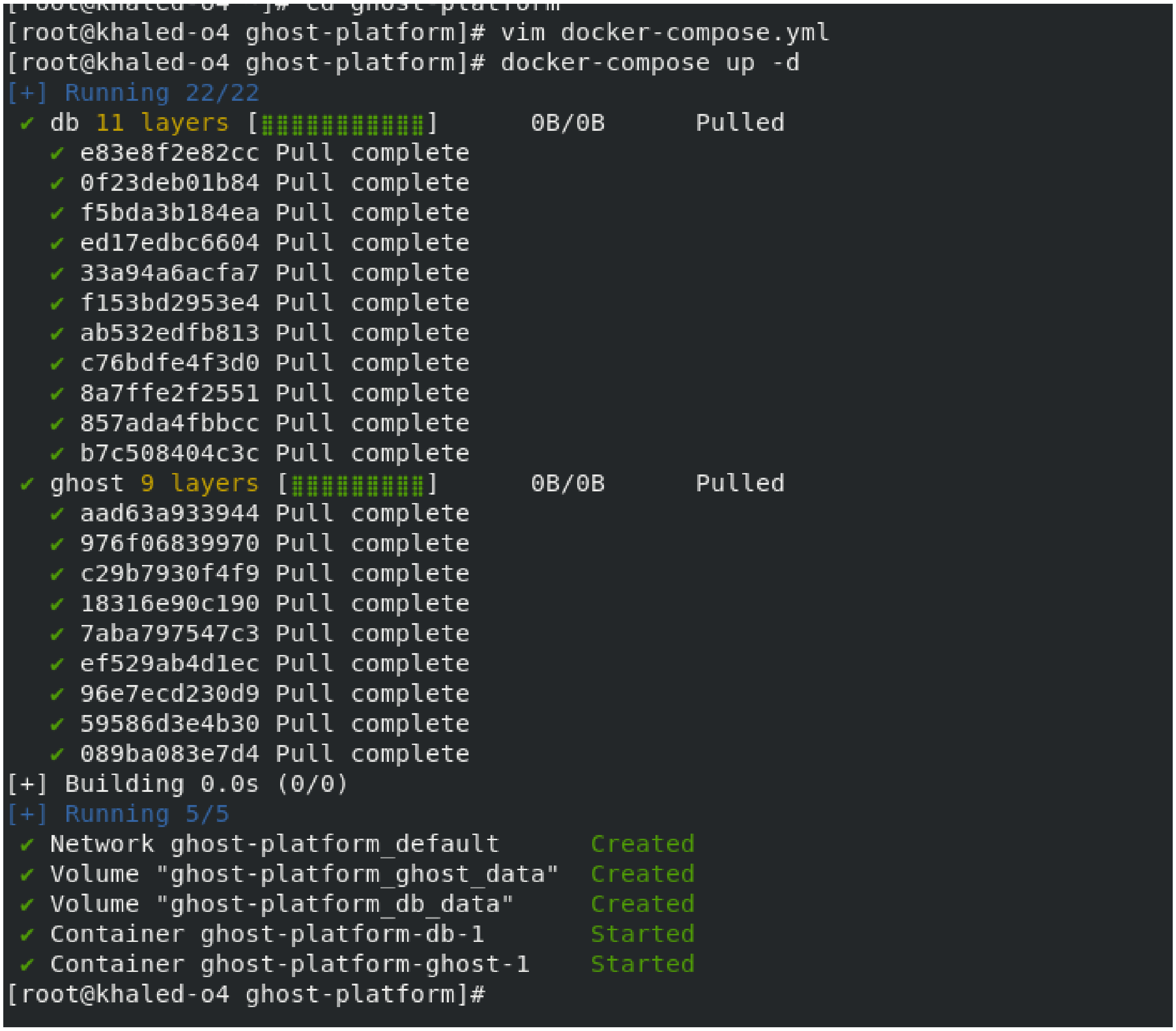


Hello World

4. Problem5:

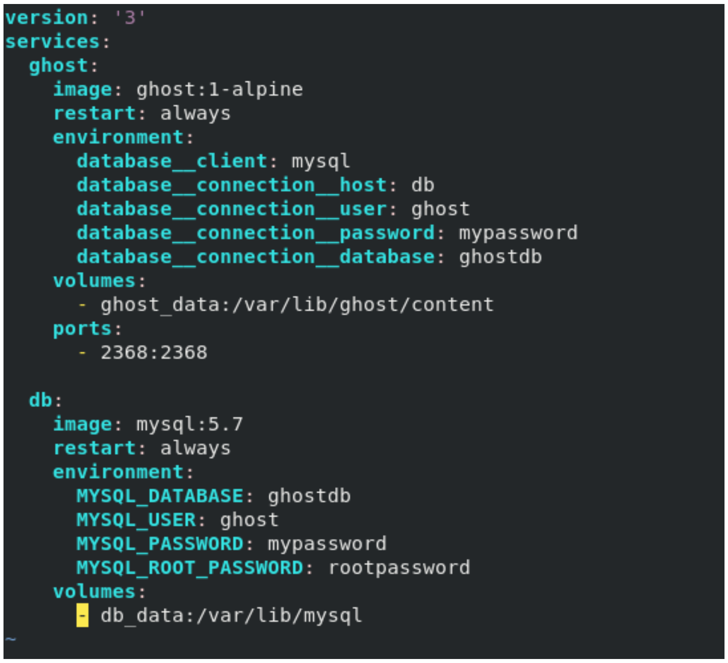
Use docker compose to deploy ghost platform (image: ghost:1-alpine)(Ghost is a free and open source blogging platform written in JavaScript)

Use mysql database instead of sqlite

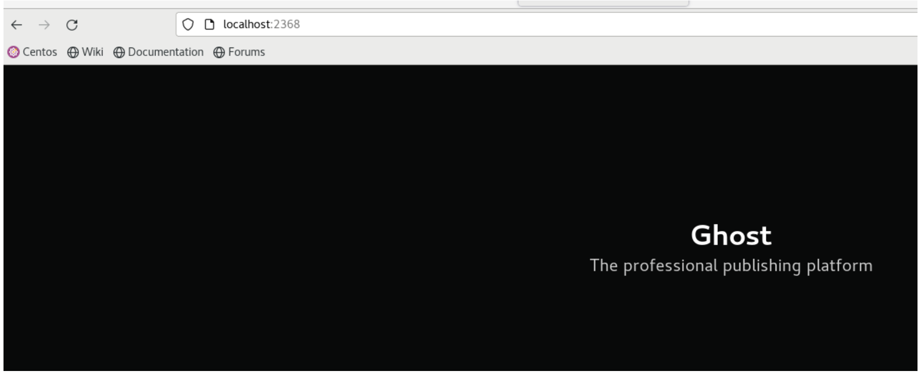


```
[root@khaled-o4 ~]# cd ghost-platform
[root@khaled-o4 ghost-platform]# vim docker-compose.yml
[root@khaled-o4 ghost-platform]# docker-compose up -d
[+] Running 22/22
 ✔ db 11 layers [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled
   ✔ e83e8f2e82cc Pull complete
   ✔ 0f23deb01b84 Pull complete
   ✔ f5bda3b184ea 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
 ✔ ghost 9 layers [⣿⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled
   ✔ aad63a933944 Pull complete
   ✔ 976f06839970 Pull complete
   ✔ c29b7930f4f9 Pull complete
   ✔ 18316e90c190 Pull complete
   ✔ 7aba797547c3 Pull complete
   ✔ ef529ab4d1ec Pull complete
   ✔ 96e7ecd230d9 Pull complete
   ✔ 59586d3e4b30 Pull complete
   ✔ 089ba083e7d4 Pull complete
[+] Building 0.0s (0/0)
[+] Running 5/5
 ✔ Network ghost-platform_default       Created
 ✔ Volume "ghost-platform_ghost_data"   Created
 ✔ Volume "ghost-platform_db_data"      Created
 ✔ Container ghost-platform-db-1        Started
 ✔ Container ghost-platform-ghost-1     Started
[root@khaled-o4 ghost-platform]#
```

vim docker-compose.yml



```
version: '3'
services:
  ghost:
    image: ghost:1-alpine
    restart: always
    environment:
      database__client: mysql
      database__connection__host: db
      database__connection__user: ghost
      database__connection__password: mypassword
      database__connection__database: ghostdb
    volumes:
      - ghost_data:/var/lib/ghost/content
    ports:
      - 2368:2368

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_DATABASE: ghostdb
      MYSQL_USER: ghost
      MYSQL_PASSWORD: mypassword
      MYSQL_ROOT_PASSWORD: rootpassword
    volumes:
      - db_data:/var/lib/mysql
~
```



Ghost

The professional publishing platform