

Lab3 Docker

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 1 with subnet 192.168.0.0/24 and attach it to ubuntu1 and ubuntu2
- 2- create network 2 with subnet 10.5.0.0/24 and attach it to ubuntu3
- 3- try to ping from ubuntu1 to ubuntu2 successfully
- 4- try to ping from ubuntu1 to ubuntu3 fail

Create a network

```
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3$ docker network create --driver bridge --subnet 192.168.0.0/24 network1
069de6ab32455497070b29f496004f7d34d9910f585f1734bb1e2df8dbbdfb1
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3$ docker network create --driver bridge --subnet 10.5.0.0/24 network2
ed9da3caa300b8e0aec8294bde3bd38e73bafbb58c63726016ce0b3bed8b9cdc
```

Run containers and attach to networks

```
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3$ docker run -d --name ubuntu1 --network network1 ubuntu sleep 1000
2874bf2d65c46dfd45250a9cbbfc5c36430ab88d0a848d6732cfe051a38505c7
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3$ docker run -d --name ubuntu2 --network network1 ubuntu sleep 1000
bea9d93a9f83ee9bed84d34b1b9aa68695d207df414852ba13fccf20864af38e
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3$ docker run -d --name ubuntu3 --network network2 ubuntu sleep 1000
07dfb0442f11abd9d24b74eef729d6b6dbb27b10470709947dfd8ecfa6c4b0e0
```

```
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3$ docker exec -it ubuntu1 /bin/bash
```

```
root@2874bf2d65c4:/# 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 ---
7 packets transmitted, 0 received, 100% packet loss, time 6148ms
```

```
root@2874bf2d65c4:/# 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=0.079 ms
64 bytes from 192.168.0.3: icmp_seq=2 ttl=64 time=0.108 ms
64 bytes from 192.168.0.3: icmp_seq=3 ttl=64 time=0.060 ms
64 bytes from 192.168.0.3: icmp_seq=4 ttl=64 time=0.074 ms
64 bytes from 192.168.0.3: icmp_seq=5 ttl=64 time=0.069 ms
64 bytes from 192.168.0.3: icmp_seq=6 ttl=64 time=0.078 ms
^C
--- 192.168.0.3 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5127ms
rtt min/avg/max/mdev = 0.060/0.078/0.108/0.014 ms
root@2874bf2d65c4:/#
```

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

Docker file

```
ahmed@ahmed-VirtualBox: ~/Desktop/docker_task3/Appachi

FROM httpd:2.4

COPY ./index.html /usr/local/apache2/htdocs/

RUN echo "Listen 9999" >>/usr/local/apache2/conf/httpd.conf

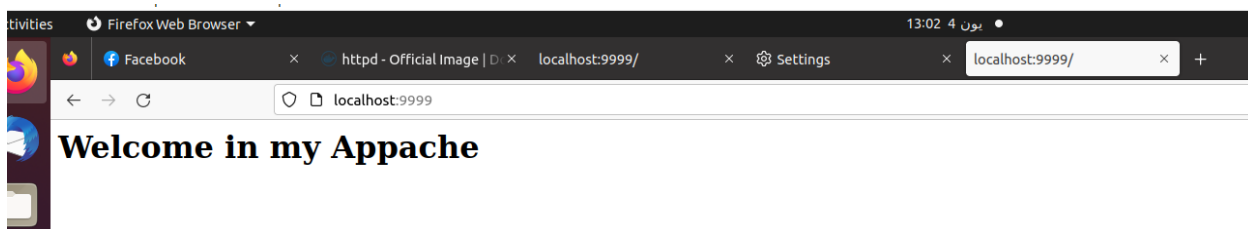
EXPOSE 9999

ENV CONTAINER="docker"
CMD ["sh", "-c", "echo $CONTAINER && httpd-foreground" ]
```

Build & Run

```
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3/Appachi$ docker build -t apache:v5 .
[+] Building 7.4s (8/8) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 28
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 258B
=> [internal] load metadata for docker.io/library/httpd:2.4
=> [internal] load build context
=> => transferring context: 68B
=> CACHED [1/3] FROM docker.io/library/httpd:2.4@sha256:1bb3f7669a85713906e695599d29c58ab40d4e6409907946609d92a428e95b49
=> [2/3] COPY ./index.html /usr/local/apache2/htdocs/
=> [3/3] RUN echo "Listen 9999" >>/usr/local/apache2/conf/httpd.conf
=> exporting to image
=> => exporting layers
=> => writing image sha256:262a7f2327bd3e7f9eac47820df5b63e5e1b1fbaa29c7121a4640db4d5363334
=> => naming to docker.io/library/apache:v5
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3/Appachi$ docker run --name apache-server5 -p 9999:9999 apache:v5
docker
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[Sun Jun 04 09:43:29.709970 2023] [mpm_event:notice] [pid 7:tid 140464921300288] AH00489: Apache/2.4.57 (Unix) configured -- resuming normal operations
[Sun Jun 04 09:43:29.728246 2023] [core:notice] [pid 7:tid 140464921300288] AH00094: Command line: 'httpd -D FOREGROUND'
172.17.0.1 - - [04/Jun/2023:09:43:36 +0000] "GET / HTTP/1.1" 200 31
172.17.0.1 - - [04/Jun/2023:09:43:36 +0000] "GET /favicon.ico HTTP/1.1" 404 196
172.17.0.1 - - [04/Jun/2023:09:44:58 +0000] "GET / HTTP/1.1" 304 -
```

Results



3. Problem 3:

Create a docker compose to up mysql container, and

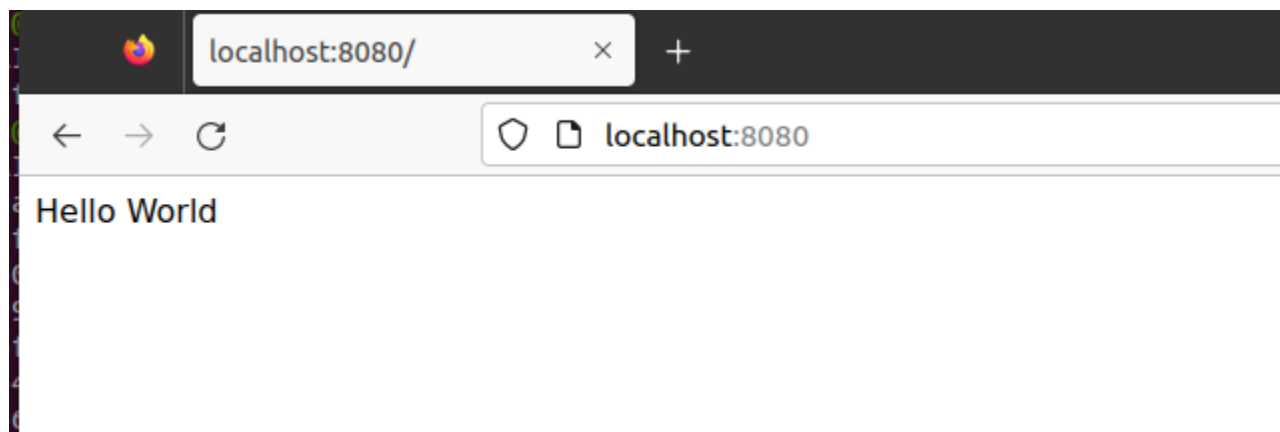
<https://github.com/sabreensalama/dockerize-node-app-task> which depend on mysqldb.

Add volume for mysqldb

```
file machine view input Devices Help
Activities Terminal 06:11 2 يون
ahmed@ahmed-VirtualBox: ~/Desktop/docker_task3

version: "3"
services:
  node-app:
    build: .
    ports:
      - "8080:8080"
  mysqldb:
    image: mysql
    volumes:
      - ./db_Volume
    environment:
      MYSQL_ROOT_PASSWORD: DB_123

ahmed@ahmed-VirtualBox:~/Desktop/docker_task3$ vim docker-compose.yml
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3$ docker compose up -d
[+] Building 0.0s (0/0)
[+] Running 2/2
✓ Container docker_task3-mysqldb-1 Started
✓ Container docker_task3-node-app-1 Running
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
31a4e69aec7d   mysql         "docker-entrypoint.s..." 14 seconds ago Up 12 seconds 3306/tcp, 33060/tcp              docker_task3-mysqldb-1
76e312960637   docker_task3-node-app "docker-entrypoint.s..." 17 minutes ago Up 17 minutes 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  docker_task3-node-app-1
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3$
```



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

```
ahmed@ahmed-VirtualBox: ~/Desktop
ahmed@ahmed-VirtualBox: ~/Desktop/docker_task3/docker_task3-4
version: "3"
services:
  ghost-app:
    image: ghost:1-alpine
    ports:
      - "4000:2368"
  mysqlldb:
    image: mysql
    volumes:
      - ./db_Volume
    environment:
      MYSQL_ROOT_PASSWORD: DB_123
```

```
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3/docker_task3-4$ vi docker-compose.yml
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3/docker_task3-4$ docker compose up -d
[+] Building 0.0s (0/0)
[+] Running 2/2
✔ Container docker_task3-4-mysqldb-1 Running
✔ Container docker_task3-4-ghost-app-1 Started
ahmed@ahmed-VirtualBox:~/Desktop/docker_task3/docker_task3-4$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
d8b42cbc9118   ghost:1-alpine "docker-entrypoint.s..." 13 seconds ago Up 9 seconds  0.0.0.0:4000->2368/tcp, :::4000->2368/tcp  docker_task3-4-ghost-app-1
e481a975b06a   mysql         "docker-entrypoint.s..." 8 minutes ago  Up 4 minutes  3306/tcp, 33060/tcp                docker_task3-4-mysqldb-1
```

