

## 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

```
root@kali: ~  
File Actions Edit View Help  
(root@kali)-[~]  
# docker network create bridge_network1 --subnet 192.168.0.0/24  
88aa339ae29e9c45c285d406730a221e9010c81466eb902357e60bba5fc55790  
(root@kali)-[~]  
# docker ps  
CONTAINER ID   IMAGE     COMMAND                  CREATED      STATUS      PORTS      NAMES  
(root@kali)-[~]  
# docker run -d --name container1 --network bridge_network1 nginx  
67a21430b6a890d3b974f5b54058fa7348b4e10f889a239222e114fd1fad4a38  
(root@kali)-[~]  
# docker run -d --name container2 --network bridge_network1 nginx  
c1ecbcd7ae98fae4fef75a07869405016b876833aab97c9ba38e2de45b2a6d5c  
(root@kali)-[~]  
# docker ps  
CONTAINER ID   IMAGE     COMMAND                  CREATED      STATUS      PORTS      NAMES  
c1ecbcd7ae98   nginx     "/docker-entrypoint..." 8 seconds ago  Up 7 seconds  80/tcp     container2  
67a21430b6a8   nginx     "/docker-entrypoint..." 21 seconds ago  Up 19 seconds  80/tcp     container1  
(root@kali)-[~]  
# docker network create bridge_network2 --subnet 10.5.0.0/24  
05c84e326f585c0ed3b704d4960d3bd8b6dc7d720ac0247b39b4829e6e38f0bb  
(root@kali)-[~]  
# docker run -d --name container3 --network bridge_network2 nginx  
fc808adf6abc1b98fcb1e59fe35a66021b839ceb1e36c11338963bfbcb85ac94  
(root@kali)-[~]  
# iptables -I DOCKER-USER -s 192.168.0.0/24 -d 10.5.0.0/24 -p icmp -j DROP  
(root@kali)-[~]  
# iptables -I DOCKER-USER -s 10.5.0.0/24 -d 192.168.0.0/24 -p icmp -j DROP  
(root@kali)-[~]  
# docker ps  
CONTAINER ID   IMAGE     COMMAND                  CREATED      STATUS      PORTS      NAMES  
fc808adf6abc   nginx     "/docker-entrypoint..." 12 minutes ago  Up 12 minutes  80/tcp     container3  
c1ecbcd7ae98   nginx     "/docker-entrypoint..." 15 minutes ago  Up 15 minutes  80/tcp     container2  
67a21430b6a8   nginx     "/docker-entrypoint..." 15 minutes ago  Up 15 minutes  80/tcp     container1  
(root@kali)-[~]
```

## 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@kali)-[/home/kali/Desktop/Docker3]
# vim Dockerfile

(root@kali)-[/home/kali/Desktop/Docker3]
# cat Dockerfile
FROM httpd:latest

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

COPY httpd.conf /usr/local/apache2/conf/httpd.conf

EXPOSE 9999

ENV CONTAINER=docker

CMD echo "The value of CONTAINER is: $CONTAINER"

(root@kali)-[/home/kali/Desktop/Docker3]
# vim httpd.conf

(root@kali)-[/home/kali/Desktop/Docker3]
# cat httpd.conf
Listen 9999
```

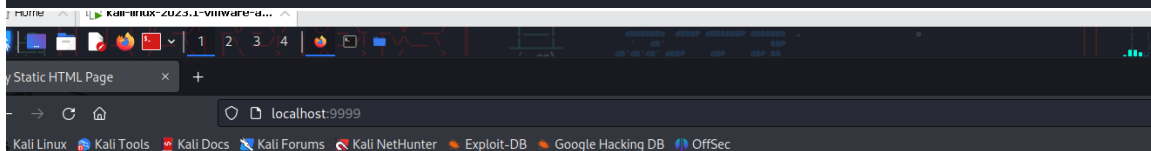
```

(root@kali)-[/home/kali/Desktop/Docker3]
# docker build -t my-httpd-image .
Sending build context to Docker daemon 4.096kB
Step 1/6 : FROM httpd:latest
latest: Pulling from library/httpd
f03b40093957: Already exists
abaf8619eb1c: Pull complete
e3fe37d0c2ad: Pull complete
52a1e37affe5: Pull complete
49d8a68fd903: Pull complete
Digest: sha256:1bb3f7669a85713906e695599d29c58ab40d4e6409907946609d92a428e95b49
Status: Downloaded newer image for httpd:latest
   -> d1676199e605
Step 2/6 : COPY index.html /usr/local/apache2/htdocs/
   -> 22a83737eed7
Step 3/6 : COPY httpd.conf /usr/local/apache2/conf/httpd.conf
   -> a180eb07ffe7
Step 4/6 : EXPOSE 9999
   -> Running in 8eda0f11196d
Removing intermediate container 8eda0f11196d
   -> 8330d114a429
Step 5/6 : ENV CONTAINER=docker
   -> Running in 66da1fe63c70
Removing intermediate container 66da1fe63c70
   -> 22a9e0df183d
Step 6/6 : CMD echo "The value of CONTAINER is: $CONTAINER"
   -> Running in c9b1cddb1d8b
Removing intermediate container c9b1cddb1d8b
   -> 514b7d0d7b43
Successfully built 514b7d0d7b43
Successfully tagged my-httpd-image:latest

(root@kali)-[/home/kali/Desktop/Docker3]
# docker run -d -p 9999:9999 my-httpd-image

9df5e3ef7525c6e693ec07555698680449dfd864088d180bc2fe5bbe9e5cba0d

```



**Welcome to my static HTML page!**

### 3. Problem 3:

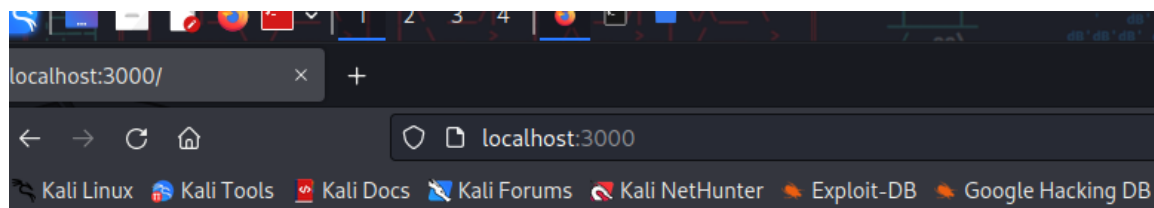
create a docker compose to up mysql container, and

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

Add volume for mysqldb

```
(root@kali)-[/home/kali/Desktop/dockerize-node-app-task]
# cat docker-compose.yml
version: '3'
services:
  mysql:
    image: mysql:latest
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: passwd
      MYSQL_DATABASE: mydbpw
      MYSQL_USER: mohamed
      MYSQL_PASSWORD: mohamed
    volumes:
      - mysql_data:/var/lib/mysql
  app:
    build: .
    restart: always
    depends_on:
      - mysql
    ports:
      - "3000:8080"
volumes:
  mysql_data:
```

```
(root@kali)-[/home/kali/Desktop/dockerize-node-app-task]
# docker-compose up
Creating dockerize-node-app-task_mysql_1 ... done
Creating dockerize-node-app-task_app_1 ... done
Attaching to dockerize-node-app-task_mysql_1, dockerize-node-app-task_app_1
mysql_1 | 2023-06-05 19:40:31+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.33-1.el8 started.
mysql_1 | 2023-06-05 19:40:32+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql_1 | /var/lib/mysql/mysql.sock → /var/run/mysqld/mysqld.sock
app_1 | Running on http://0.0.0.0:8080
mysql_1 | 2023-06-05T19:40:33.040811Z 0 [Warning] [MY-011866] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
mysql_1 | 2023-06-05T19:40:33.043819Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.33) starting as process 1
mysql_1 | 2023-06-05T19:40:33.054211Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql_1 | 2023-06-05T19:40:33.256565Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql_1 | 2023-06-05T19:40:33.406921Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
mysql_1 | 2023-06-05T19:40:33.407147Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
mysql_1 | 2023-06-05T19:40:33.490042Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
mysql_1 | 2023-06-05T19:40:33.523696Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqld.sock
mysql_1 | 2023-06-05T19:40:33.524209Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.33' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server - GPL.
Gracefully stopping... (press Ctrl+C again to force)
Stopping dockerize-node-app-task_app_1 ...
Stopping dockerize-node-app-task_mysql_1 ...
```



Hello World

#### 4. 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

```
(root@kali) ~ /home/kali/Desktop/dockerize-node-app-task
$ docker-compose up --remove-orphans
Pulling db (mysql:5.7) ...
5.7: Pulling from library/mysql
e83e8f2e82cc: Pull complete
9f23a0b01b84: Pull complete
f5bda3b184ea: Pull complete
ed17eddc6684: Pull complete
2da9a4a6cf77: Pull complete
f153bd2953e4: Pull complete
ab532edf8813: Pull complete
c76b8f64f280: Pull complete
8a7ffe2f2551: Pull complete
857ada4fbccc: Pull complete
37c580a84c3c: Pull complete
Digest: sha256:f57ee421808aaf8332a91ab0bdc96b3c83ed2a981c29e6528b21ce10197cd16
Status: Downloaded newer image for mysql:5.7
Pulling ghost (ghost:1-alpine) ...
1-alpine: Pulling from library/ghost
aad63a933944: Pull complete
976f86839970: Pull complete
c20b7938f4f9: Pull complete
18316e90c190: Pull complete
7aba797547c3: Pull complete
e452ba84d1ec: Pull complete
9667ecd23809: Pull complete
5958d63e4b30: Pull complete
809ba882e7d4: Pull complete
Digest: sha256:8a9957f8831db9f6ea87fe95217053939601fcd1db047c8b106f0ec4b7586b
Status: Downloaded newer image for ghost:1-alpine
Creating dockerize-node-app-task-db_1 ... done
Creating dockerize-node-app-task-ghost_1 ... done
Attaching to dockerize-node-app-task-db_1, dockerize-node-app-task-ghost_1
db_1 | 2023-06-06 00:26:10+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.42-1.el7 started.
db_1 | 2023-06-06 00:26:10+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
db_1 | 2023-06-06 00:26:10+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.42-1.el7 started.
db_1 | 2023-06-06 00:26:10+00:00 [Note] [Entrypoint]: Initializing database files
db_1 | mysql: Can't create/write to file '/var/lib/mysql/is_writable' (Errcode: 13 - Permission denied)
db_1 | 2023-06-06T00:26:10.960195Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
db_1 | 2023-06-06T00:26:10.961946Z 0 [ERROR] --initialize specified but the data directory exists and is not writable. Aborting.
db_1 | 2023-06-06T00:26:10.961952Z 0 [ERROR] Aborting
ghost_1 | (node:56) [DEP0096] DeprecationWarning: timers.unenroll() is deprecated. Please use clearTimeout instead.
ghost_1 | [2023-06-06 00:26:11]
```

```
(root@kali) ~ /home/kali/Desktop/dockerize-node-app-task
$ cat docker-compose.yml
version: '3'
services:
  ghost:
    image: ghost:1-alpine
    restart: always
    ports:
      - 2368:2368
    environment:
      - database__client=mysql
      - database__connection__host=db
      - database__connection__user=root
      - database__connection__password=mysecretpassword
      - database__connection__database=ghost
    depends_on:
      - db
    volumes:
      - ./data:/var/lib/ghost/content

  db:
    image: mysql:5.7
    restart: always
    environment:
      - MYSQL_ROOT_PASSWORD=mysecretpassword
      - MYSQL_DATABASE=ghost
    volumes:
      - ./data/mysql:/var/lib/mysql
```

