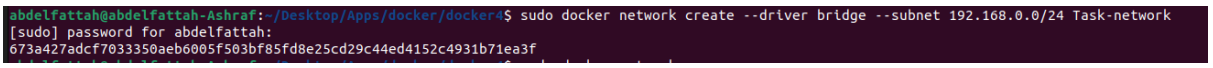
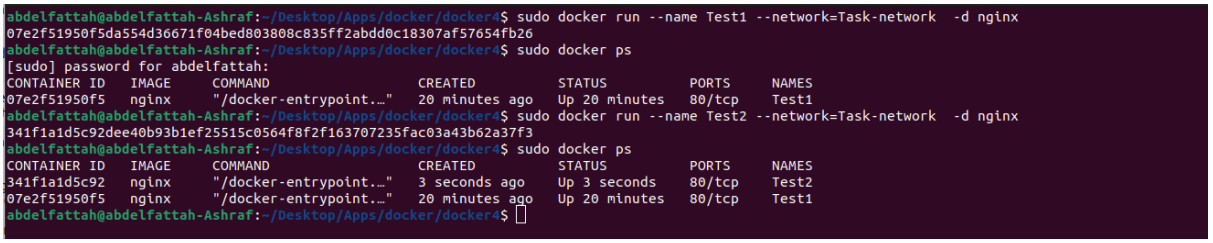
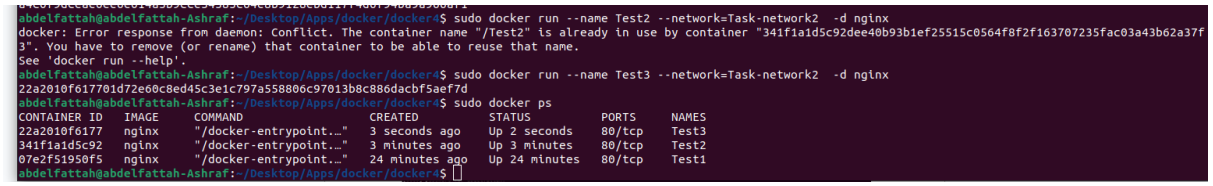
1. Problem 1:
   Create bridge network with subnet 192.168.0.0/24.
   Run 2 containers and attach containers to this network.
   Create another bridge network with subnet 10.5.0.0/24.
   Run any container and attach it to the new network.
   Make sure that the containers at different network can’t ping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







2. Problem 2:
   Create static html file
   Write Dockerfile to build image based on https 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

```
abdelfattah@abdelfattah-Ashraf:~/Desktop/Apps/docker/docker4$ sudo docker build -t my-nginx-image .
[sudo] password for abdelfattah:
[+] Building 0.7s (8/8) FINISHED
 => [internal] load .dockerignore                                                0.0s
 => => transferring context: 2B                                                  0.0s
 => [internal] load build definition from Dockerfile                             0.1s
 => => transferring dockerfile: 490B                                             0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                  0.0s
 => [internal] load build context                                                0.0s
 => => transferring context: 211B                                                0.0s
 => CACHED [1/3] FROM docker.io/library/nginx                                    0.0s
 => [2/3] COPY index.html .                                                      0.1s
 => [3/3] COPY nginx.conf .                                                      0.1s
 => exporting to image                                                           0.2s
 => => exporting layers                                                          0.1s
 => => writing image sha256:6f3f148cf499163d5fdee9c151331ff94023e7f5498e757d69c0e6e64a34ec9e  0.0s
 => => naming to docker.io/library/my-nginx-image                                0.0s
abdelfattah@abdelfattah-Ashraf:~/Desktop/Apps/docker/docker4$ sudo docker run --name my-nginx-container -p 9999:9999 -d my-nginx-image
033f492d61e8cffe69e01eeb67e9793275b89a92e16f6f20bee1ccf280cb5d92
abdelfattah@abdelfattah-Ashraf:~/Desktop/Apps/docker/docker4$
```

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

```
abdelfattah@abdelfattah-Ashraf:~/Desktop/Apps/docker/Task Docker 3/dockerize-node-app-task-main$ sudo docker-compose up -d
Creating network "dockerize-node-app-task-main_default" with the default driver
Creating volume "dockerize-node-app-task-main_mysql-data" with default driver
Building node-app
[+] Building 1.9s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 428B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/node:14-alpine
 => [auth] library/node:pull token for registry-1.docker.io
 => [1/5] FROM docker.io/library/node:14-alpine@sha256:434215b487a329c9e867202ff89e704d3a75e554822e07f3e0c0f9e606121b33
 => [internal] load build context
 => => transferring context: 659B
 => CACHED [2/5] WORKDIR /app
 => CACHED [3/5] COPY package*.json ./
 => CACHED [4/5] RUN npm install
 => [5/5] COPY . .
 => exporting to image
 => => exporting layers
 => => writing image sha256:80531cb5105375b7df2f721e9a5ab810f41190378175acf1e416bfaa9dccdfb
 => => naming to docker.io/library/dockerize-node-app-task-main_node-app
WARNING: Image for service node-app was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compose up --build`.
Creating dockerize-node-app-task-main_mysql_1 ... done
Creating dockerize-node-app-task-main_node-app_1 ... done
abdelfattah@abdelfattah-Ashraf:~/Desktop/Apps/docker/Task Docker 3/dockerize-node-app-task-main$
```

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

```
abdelfattah@abdelfattah-Ashraf:~/Desktop/Apps/docker/docker4/ghost$ sudo docker-compose up -d
[sudo] password for abdelfattah:
Creating network "ghost_default" with the default driver
Creating volume "ghost_ghost-data" with default driver
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
ef529ab4d1ec: Pull complete
96e7ecd230d9: Pull complete
59586d3e4b30: Pull complete
089ba083e7d4: Pull complete
Digest: sha256:0a9957f8831db9fe6a87fe95217053939601fdcd1db047cb8b106f0ec4b7506b
Status: Downloaded newer image for ghost:1-alpine
Creating ghost_mysql_1 ... done
Creating ghost_ghost_1 ... done
abdelfattah@abdelfattah-Ashraf:~/Desktop/Apps/docker/docker4/ghost$
```