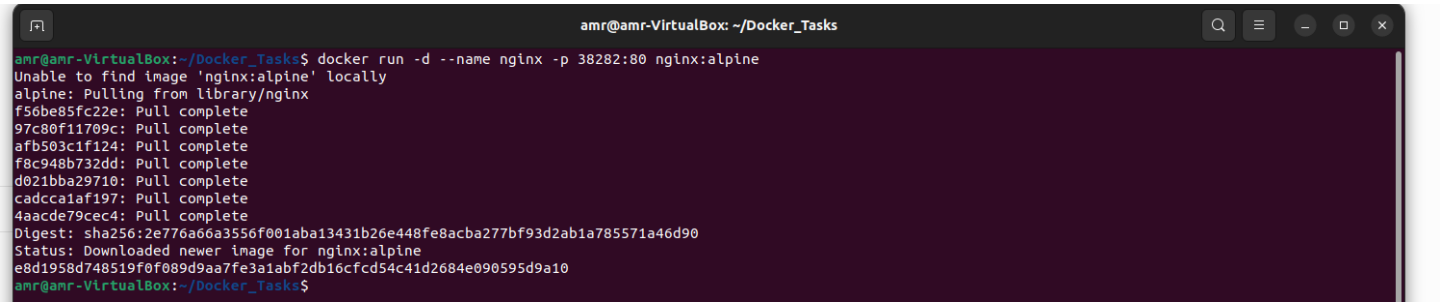
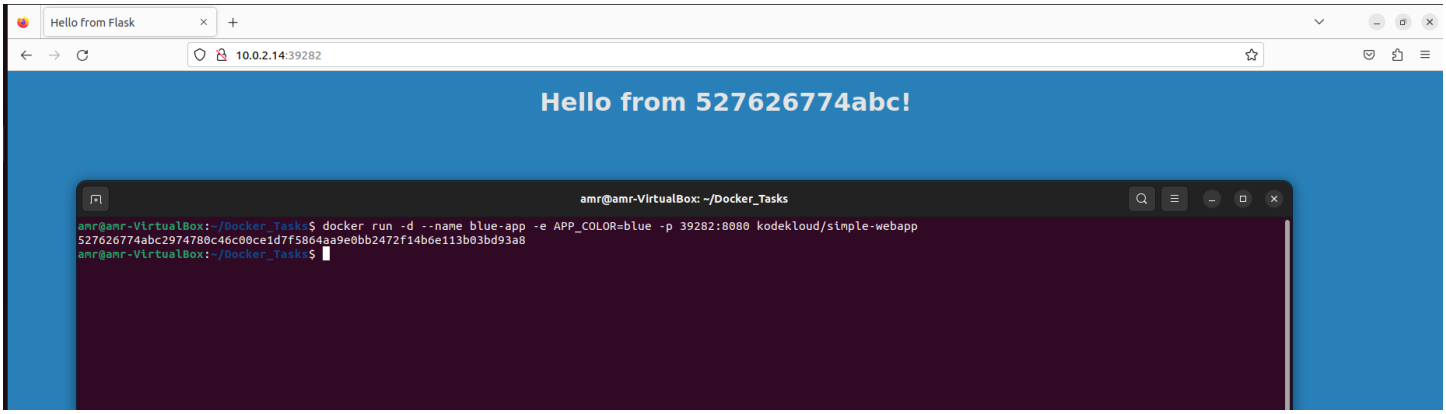
1- Run an instance of nginx:alpine with a name ,nginx and map port 80 on the container to 38282 on the host.



2- create ubuntu image and check the size of it

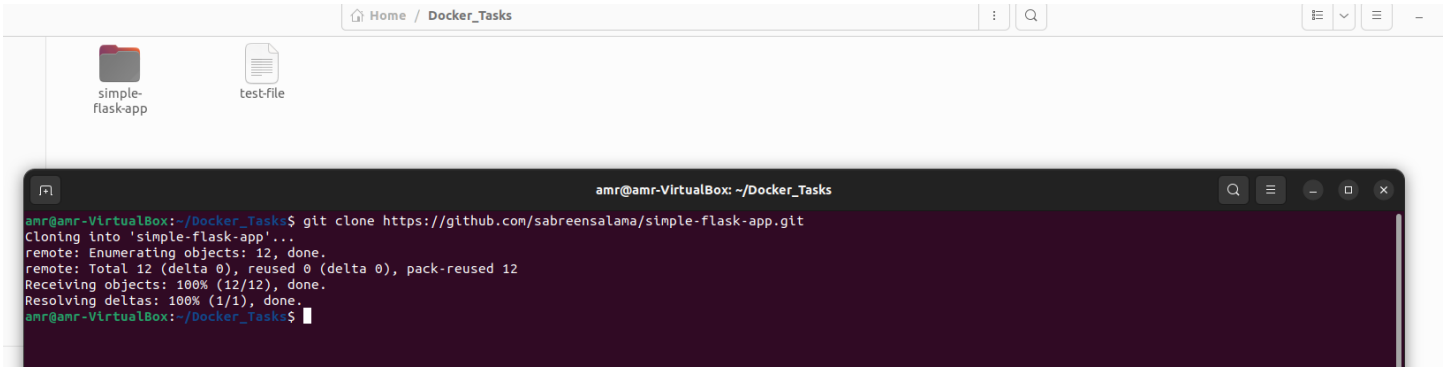
3 - Run a container named blue-app using image kodekloud/simple-webapp and set the environment variable APP COLOR to blue. Make the application available on port 38282 on the host. The application listens on port 8080

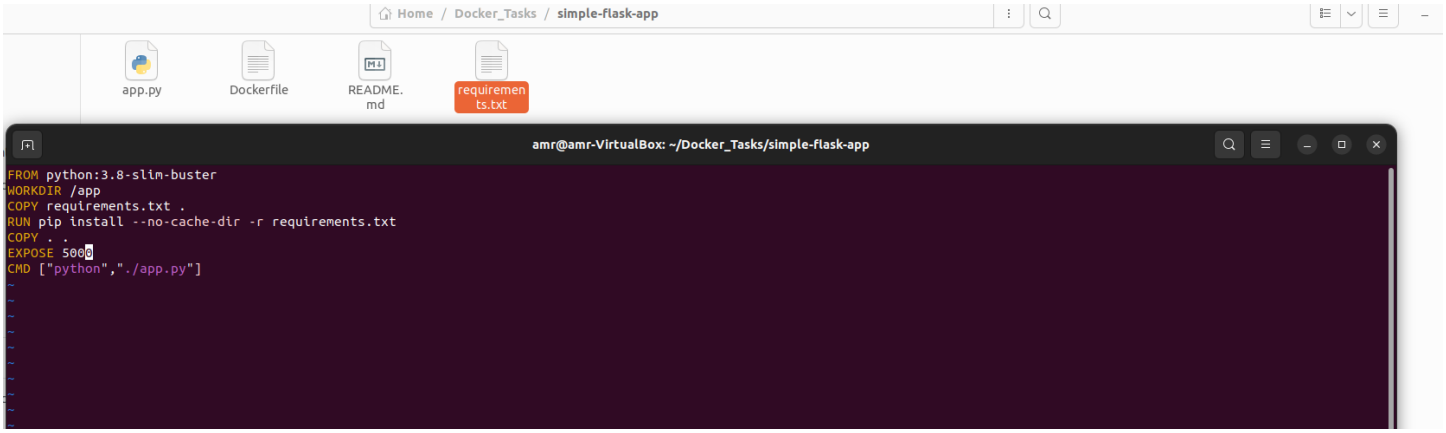


4 - Deploy a mysql database using the mysql image and name it mysql-db Set the database password to use db_pass123 then inspect it to check the value

```
amr@amr-VirtualBox:~/Docker_Tasks$ docker run -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql
Unable to find image 'mysql:latest' locally
latest: 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:latest
352175fbb6dcb92c7586e2f4f60d2fea0e7807dc3918740108f83727dd2171cf
amr@amr-VirtualBox:~/Docker_Tasks$ docker inspect mysql-db | grep Password
amr@amr-VirtualBox:~/Docker_Tasks$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS          PORTS                                          NAMES
352175fbb6dc   mysql                    "docker-entrypoint.s…"   39 seconds ago   Up 37 seconds   3306/tcp, 33060/tcp                            mysql-db
527626774abc   kodekloud/simple-webapp  "python app.py"          3 minutes ago    Up 3 minutes    0.0.0.0:39282->8080/tcp, :::39282->8080/tcp    blue-app
e8d1958d7485   nginx:alpine             "/docker-entrypoint.…"   9 minutes ago    Up 9 minutes    0.0.0.0:38282->80/tcp, :::38282->80/tcp        nginx
amr@amr-VirtualBox:~/Docker_Tasks$ dokcer inspect 3521
Command 'dokcer' not found, did you mean:
  command 'docker' from snap docker (20.10.24)
  command 'docker' from snap docker (20.10.17)
  command 'docker' from deb docker.io (20.10.21-0ubuntu1~22.04.3)
  command 'docker' from deb podman-docker (3.4.4+ds1-1ubuntu1)
See 'snap info <snapname>' for additional versions.
amr@amr-VirtualBox:~/Docker_Tasks$ docker inspect 3521
[
    {
        "Id": "352175fbb6dcb92c7586e2f4f60d2fea0e7807dc3918740108f83727dd2171cf",
        "Created": "2023-05-26T12:39:27.270499331Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "mysqld"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
```

5 - pull the code from https://github.com/sabreensalama/simple-flask-app and create a docker file for this flask app





6 - Create a volume called mysql_data, Run a mysql Container again, but this time map a volume to the container so that the data stored by container is stored at /opt/data on the host .Use the same name : mysql-db and same password; db_pass123 as before Mysql stores data at /var/lib/mysql inside the container

```
amr@amr-VirtualBox:~/Docker_Tasks/simple-flask-app$ vim Dockerfile
amr@amr-VirtualBox:~/Docker_Tasks/simple-flask-app$ docker volume create mysql_data
mysql_data
amr@amr-VirtualBox:~/Docker_Tasks/simple-flask-app$ docker run -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 -v mysql_data:/var/lib/mysql mysql
docker: Error response from daemon: Conflict. The container name "/mysql-db" is already in use by container "352175fbb6dcb92c7586e2f4f60d2fea0e7807dc3918740108f83727dd2171c
f". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
amr@amr-VirtualBox:~/Docker_Tasks/simple-flask-app$ docker run -d --name mysql-dbs -e MYSQL_ROOT_PASSWORD=db_pass123 -v mysql_data:/var/lib/mysql mysql
b972d3690d8393441d546845f572699b380c311beded9a070efd27869c5bbaf5
amr@amr-VirtualBox:~/Docker_Tasks/simple-flask-app$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS          PORTS                                                  NAMES
b972d3690d83   mysql                    "docker-entrypoint.s…"   30 seconds ago   Up 29 seconds   3306/tcp, 33060/tcp                                    mysql-dbs
352175fbb6dc   mysql                    "docker-entrypoint.s…"   24 minutes ago   Up 24 minutes   3306/tcp, 33060/tcp                                    mysql-db
527626774abc   kodekloud/simple-webapp  "python app.py"          26 minutes ago   Up 26 minutes   0.0.0.0:39282->8080/tcp, :::39282->8080/tcp            blue-app
e8d1958d7485   nginx:alpine             "/docker-entrypoint.…"   33 minutes ago   Up 33 minutes   0.0.0.0:38282->80/tcp, :::38282->80/tcp                nginx
amr@amr-VirtualBox:~/Docker_Tasks/simple-flask-app$
```

127.0.0.1:5000/

127.0.0.1:5000

Hello World

amr@amr-VirtualBox: ~/Docker_Tasks/simple-flask-app

```
amr@amr-VirtualBox:~/Docker_Tasks/simple-flask-app$ vim Dockerfile
amr@amr-VirtualBox:~/Docker_Tasks/simple-flask-app$ docker build -t flask-app .
[+] Building 1.5s (10/10) FINISHED
 => [internal] load .dockerignore                                                                                0.0s
 => => transferring context: 2B                                                                                  0.0s
 => [internal] load build definition from Dockerfile                                                             0.1s
 => => transferring dockerfile: 202B                                                                             0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim-buster                                        0.9s
 => [1/5] FROM docker.io/library/python:3.8-slim-buster@sha256:eb48d017c5e117d9fbcbe991b4dbc61339734e01578d8d350b38fe2033a67199  0.0s
 => [internal] load build context                                                                                0.0s
 => => transferring context: 2.14kB                                                                              0.0s
 => CACHED [2/5] WORKDIR /app                                                                                    0.0s
 => CACHED [3/5] COPY requirements.txt .                                                                         0.0s
 => CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt                                              0.0s
 => [5/5] COPY . .                                                                                               0.2s
 => exporting to image                                                                                           0.1s
 => => exporting layers                                                                                          0.1s
 => => writing image sha256:5077273e9a8464cd0e0d6c372925cd07f2ad509eed4ba89a142548e7ad8e4861                     0.0s
 => => naming to docker.io/library/flask-app                                                                     0.0s
amr@amr-VirtualBox:~/Docker_Tasks/simple-flask-app$ docker run -p 5000:5000 flask-app
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.6:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 100-665-108
172.17.0.1 - - [26/May/2023 13:13:15] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [26/May/2023 13:13:15] "GET /favicon.ico HTTP/1.1" 404 -
```