

## Docker-Tas 2

11- Run an instance of nginx:alpine with a name nginx and map port 8080 on the container to 38282 on the host.

```
saady@Linux: ~  
saady@Linux:~$ sudo docker run --name nginx -d -p 38282:8080 nginx:alpine  
Unable to find image 'nginx:alpine' locally  
alpine: Pulling from library/nginx  
f56be85fc22e: Already exists  
97c80f11709c: Pull complete  
afb503c1f124: Pull complete  
f8c948b732dd: Pull complete  
d021bba29710: Pull complete  
cadcca1af197: Pull complete  
4aacde79cec4: Pull complete  
Digest: sha256:2e776a66a3556f001aba13431b26e448fe8acba277bf93d2ab1a785571a46d90  
Status: Downloaded newer image for nginx:alpine  
869bfe89e7c23567087a540ea818f0d5edee574fd69a7e935b73f49b2087c22d  
saady@Linux:~$
```

12- Create ubuntu image and check the size of it

```
Dockerfile x requirements.txt  
Dockerfile  
1 FROM ubuntu  
saady@Linux: ~/Documents/Docker files  
nginx latest f9c14fe76d50 25 hours ago 143MB  
hello-world latest 9c7a54a9a43c 3 weeks ago 13.3kB  
mmumshad/simple-webapp-color latest 96bb69733441 4 years ago 932MB  
saady@Linux:~/Documents/Docker files$ sudo docker build -t ubuntu-image .  
[+] Building 4.5s (5/5) FINISHED  
=> [internal] load .dockerignore 0.0s  
=> => transferring context: 2B 0.0s  
=> [internal] load build definition from Dockerfile 0.0s  
=> => transferring dockerfile: 48B 0.0s  
=> [internal] load metadata for docker.io/library/ubuntu:latest 4.2s  
=> [1/1] FROM docker.io/library/ubuntu@sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee20739e8eb54fbf 0.1s  
=> => resolve docker.io/library/ubuntu@sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee20739e8eb54fbf 0.0s  
=> => sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee20739e8eb54fbf 1.13kB / 1.13kB 0.0s  
=> => sha256:ca5534a51dd04bbcebe9b23ba05f389466cf0c190f1f8f182d7eea92a9671d00 424B / 424B 0.0s  
=> => sha256:3b418d7b466ac6275a6bfc0c86fbc4422ff6ea0af444a294f82d3bf5173ce74 2.30kB / 2.30kB 0.0s  
=> exporting to image 0.0s  
=> => exporting layers 0.0s  
=> => writing image sha256:bb59a1c2f7b943b56c8b224b0d0039412d8594a5eebf58f58ff8512874d4db8e 0.0s  
=> => naming to docker.io/library/ubuntu-image 0.0s  
saady@Linux:~/Documents/Docker files$ sudo docker images  
REPOSITORY TAG IMAGE ID CREATED SIZE  
nginx alpine fe7edaf8a8dc 25 hours ago 41.4MB  
nginx latest f9c14fe76d50 25 hours ago 143MB  
hello-world latest 9c7a54a9a43c 3 weeks ago 13.3kB  
ubuntu-image latest bb59a1c2f7b9 4 weeks ago 77.8MB  
mmumshad/simple-webapp-color latest 96bb69733441 4 years ago 932MB  
saady@Linux:~/Documents/Docker files$
```

13- 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 39282 on the host. The application listens on port 8080.

Hello from 5d8e22f23fd2!

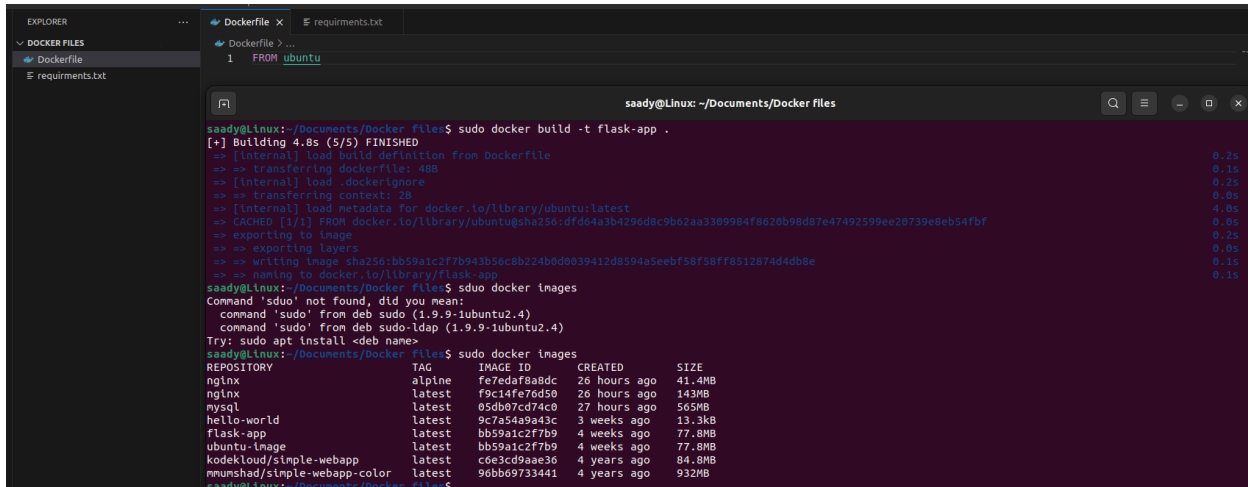
```
saady@Linux: ~/Documents/Docker files
saady@Linux:~/Documents/Docker files$ sudo docker run --name blue-app -e APP_COLOR=blue -p 39282:8080
0 kodekloud/simple-webapp
Unable to find image 'kodekloud/simple-webapp:latest' locally
latest: Pulling from kodekloud/simple-webapp
4fe2ade4980c: Pull complete
7cf6a1d62200: Pull complete
f0d690b9e495: Downloading [=====> ] 22.33MB/23.13MB
fac5d45ad062: Download complete
dd9b067ef6fd: Download complete
f0d690b9e495: Downloading [=====> ] 22.58MB/23.13MB
f0d690b9e495: Downloading [=====> ] 22.82MB/23.13MB
f0d690b9e495: Downloading [=====> ] 23.07MB/23.13MB
f0d690b9e495: Extracting [=====> ] 2.621MB/23.13MB
f0d690b9e495: Extracting [=====> ] 14.94MB/23.13MB
f0d690b9e495: Extracting [=====> ] 23.13MB/23.13MB
f0d690b9e495: Pull complete
```

14- 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.

```
saady@Linux: ~/Documents/Docker files
saady@Linux:~/Documents/Docker files$ sudo docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 -d 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
f7710c1c89fe6062c9caad72b4578b378701296f128cc6c93933871db739df4f
```

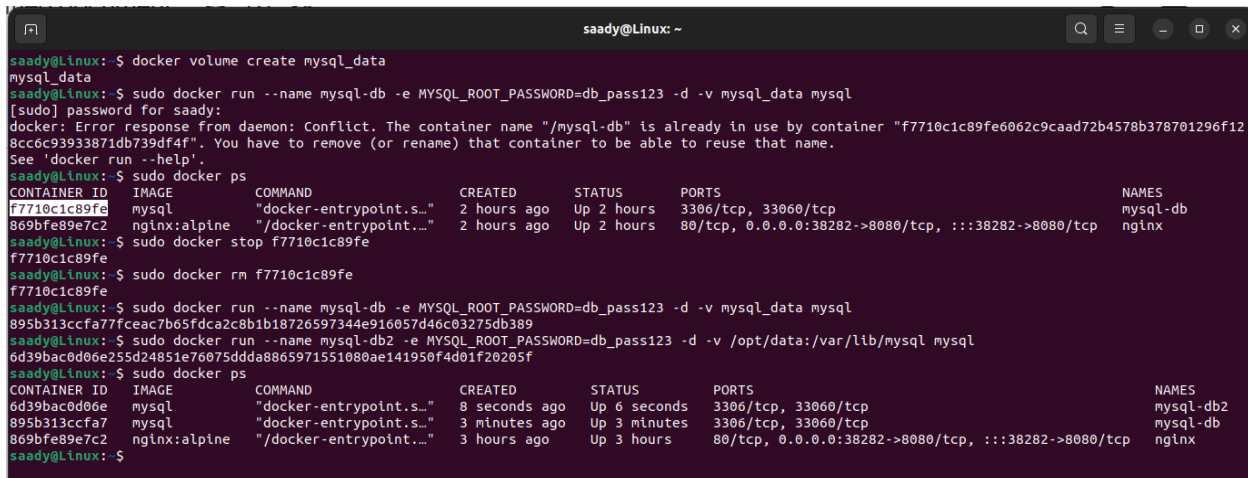
```
saady@Linux: ~/Documents/Docker files
saady@Linux:~/Documents/Docker files$ sudo docker ps
[sudo] password for saady:
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
f7710c1c89fe   mysql     "docker-entrypoint.s..." 19 minutes ago Up 18 minutes 3306/tcp, 33060/tcp               mysql
869bfe89e7c2   nginx:alpine "/docker-entrypoint..." 47 minutes ago Up 47 minutes 80/tcp, 0.0.0.0:38282->8080/tcp, :::38282->8080/tcp   nginx
saady@Linux:~/Documents/Docker files$ sudo docker inspect f7710c1c89fe
[
  {
    "Id": "f7710c1c89fe6062c9caad72b4578b378701296f128cc6c93933871db739df4f",
    "Created": "2023-05-26T00:06:53.919196076Z",
    "Path": "docker-entrypoint.sh",
    "Args": [
      "mysqld"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 12719,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2023-05-26T00:06:59.237202468Z",
      "FinishedAt": "2023-05-26T00:06:59.237202468Z"
    }
  }
]
```

15- pull the code from <https://github.com/sabreensalama/simple-flask-app/tree/main> and create a docker file for this flask app



```
saady@Linux: ~/Documents/docker files$ sudo docker build -t flask-app .
[+] Building 4.8s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 400B 0.2s
=> [internal] load .dockerignore 0.1s
=> => transferring context: 2B 0.2s
=> [internal] load metadata for docker.io/library/ubuntu:latest 0.0s
=> CACHED [1/1] FROM docker.io/library/ubuntu:sha256:dfd64a3b4296d8c9b2aa3389984f9670b98d87e47492599ee20739e8eb54fbf 4.8s
=> exporting to image 0.0s
=> => exporting layers 0.2s
=> => writing image sha256:bb59a1c2f7b943b5c8b224b0d0039412d8594a5eebf58f58f8512874d4db8e 0.1s
saady@Linux: ~/Documents/docker files$ sudo docker images
Command 'sduo' not found, did you mean:
  command 'sudo' from deb sudo (1.9.9-1ubuntu2.4)
  command 'sudo' from deb sudo-ldap (1.9.9-1ubuntu2.4)
Try: sudo apt install <deb name>
saady@Linux: ~/Documents/docker files$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx                alpine             fe7edaf8a8dc       26 hours ago       41.4MB
nginx                latest             f9c14fe76d50       26 hours ago       143MB
mysql                latest             05db07cd74c0       27 hours ago       565MB
hello-world          latest             9c7a5a9a943c       3 weeks ago        13.3kB
flask-app            latest             bb59a1c2f7b9       4 weeks ago        77.8MB
ubuntu-image         latest             bb59a1c2f7b9       4 weeks ago        77.8MB
kodekloud/simple-webapp latest             c6e3cd9aae36       4 years ago        84.8MB
munshad/simple-webapp-color latest             96bb69733441       4 years ago        932MB
saady@Linux: ~/Documents/docker files$
```

16- Create a volume called mysql\_data, run a mysql container again, but this time map 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 the 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.



```
saady@Linux: ~$ docker volume create mysql_data
mysql_data
saady@Linux: ~$ sudo docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 -d -v mysql_data mysql
[sudo] password for saady:
docker: Error response from daemon: Conflict. The container name "/mysql-db" is already in use by container "f7710c1c89fe6062c9caad72b4578b378701296f128cc6c93933871db739df4f". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
saady@Linux: ~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS                               NAMES
f7710c1c89fe   mysql    "docker-entrypoint.s..." 2 hours ago Up 2 hours 3306/tcp, 33060/tcp                mysql-db
869bfe89e7c2   nginx:alpine "/docker-entrypoint...." 2 hours ago Up 2 hours 80/tcp, 0.0.0.0:38282->8080/tcp, :::38282->8080/tcp  nginx
saady@Linux: ~$ sudo docker stop f7710c1c89fe
f7710c1c89fe
saady@Linux: ~$ sudo docker rm f7710c1c89fe
f7710c1c89fe
saady@Linux: ~$ sudo docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 -d -v mysql_data mysql
895b313ccfa77f7ceac7b65fdca2c8b1b18726597344e916057d46c03275db389
saady@Linux: ~$ sudo docker run --name mysql-db2 -e MYSQL_ROOT_PASSWORD=db_pass123 -d -v /opt/data:/var/lib/mysql mysql
6d39bac0d06e255d24851e76075ddda8865971551080ae141950f4d01f20285f
saady@Linux: ~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS                               NAMES
6d39bac0d06e   mysql    "docker-entrypoint.s..." 8 seconds ago Up 6 seconds 3306/tcp, 33060/tcp                mysql-db2
895b313ccfa7   mysql    "docker-entrypoint.s..." 3 minutes ago Up 3 minutes 3306/tcp, 33060/tcp                mysql-db
869bfe89e7c2   nginx:alpine "/docker-entrypoint...." 3 hours ago Up 3 hours 80/tcp, 0.0.0.0:38282->8080/tcp, :::38282->8080/tcp  nginx
saady@Linux: ~$
```