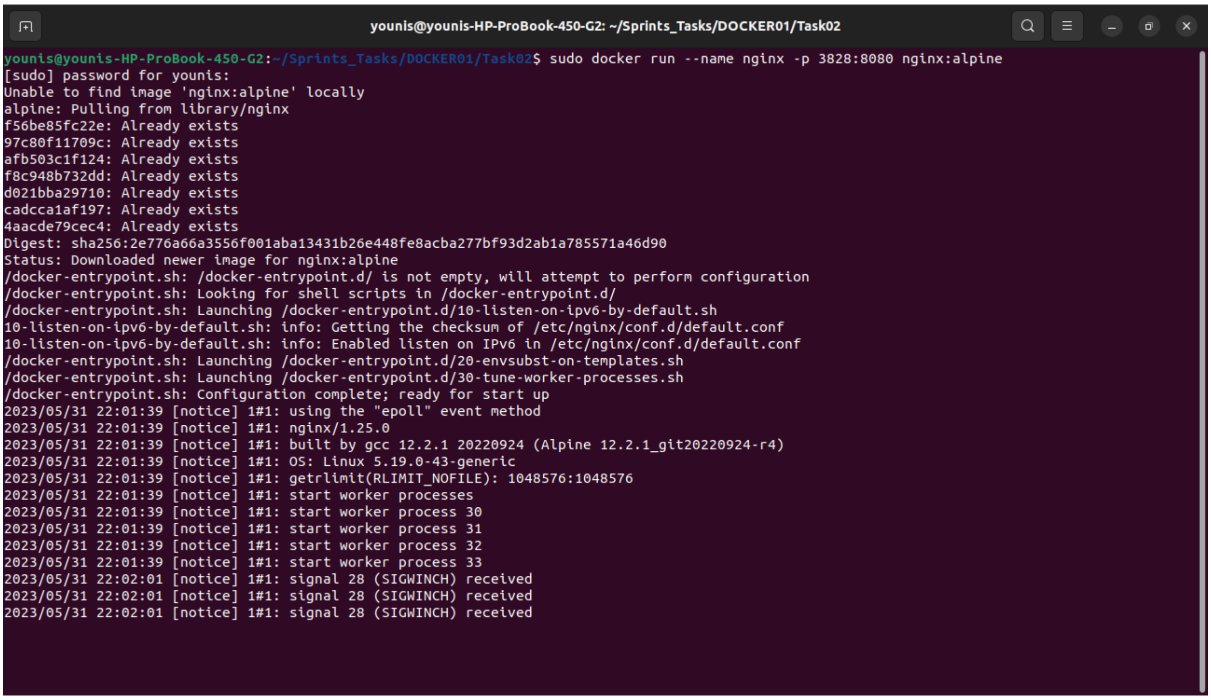
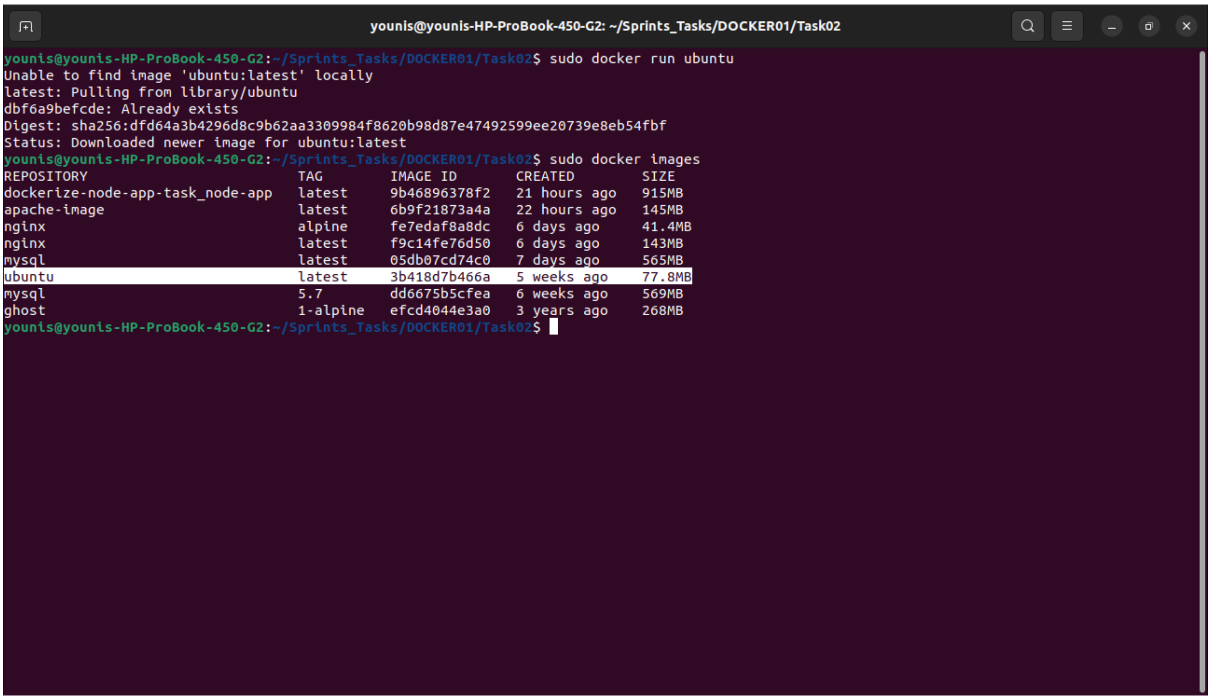
Run an instance of nginx:alpine with a name nginx and map port 8080 on the container to 3828 on the host



younis@younis-HP-ProBook-450-G2: ~/Sprints_Tasks/DOCKER01/Task02

```
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker run --name nginx -p 3828:8080 nginx:alpine
[sudo] password for younis:
Unable to find image 'nginx:alpine' locally
alpine: Pulling from library/nginx
f56be85fc22e: Already exists
97c80f11709c: Already exists
afb503c1f124: Already exists
f8c948b732dd: Already exists
d021bba29710: Already exists
cadcca1af197: Already exists
4aacde79cec4: Already exists
Digest: sha256:2e776a66a3556f001aba13431b26e448fe8acba277bf93d2ab1a785571a46d90
Status: Downloaded newer image for nginx:alpine
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/05/31 22:01:39 [notice] 1#1: using the "epoll" event method
2023/05/31 22:01:39 [notice] 1#1: nginx/1.25.0
2023/05/31 22:01:39 [notice] 1#1: built by gcc 12.2.1 20220924 (Alpine 12.2.1_git20220924-r4)
2023/05/31 22:01:39 [notice] 1#1: OS: Linux 5.19.0-43-generic
2023/05/31 22:01:39 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/05/31 22:01:39 [notice] 1#1: start worker processes
2023/05/31 22:01:39 [notice] 1#1: start worker process 30
2023/05/31 22:01:39 [notice] 1#1: start worker process 31
2023/05/31 22:01:39 [notice] 1#1: start worker process 32
2023/05/31 22:01:39 [notice] 1#1: start worker process 33
2023/05/31 22:02:01 [notice] 1#1: signal 28 (SIGWINCH) received
2023/05/31 22:02:01 [notice] 1#1: signal 28 (SIGWINCH) received
2023/05/31 22:02:01 [notice] 1#1: signal 28 (SIGWINCH) received
```

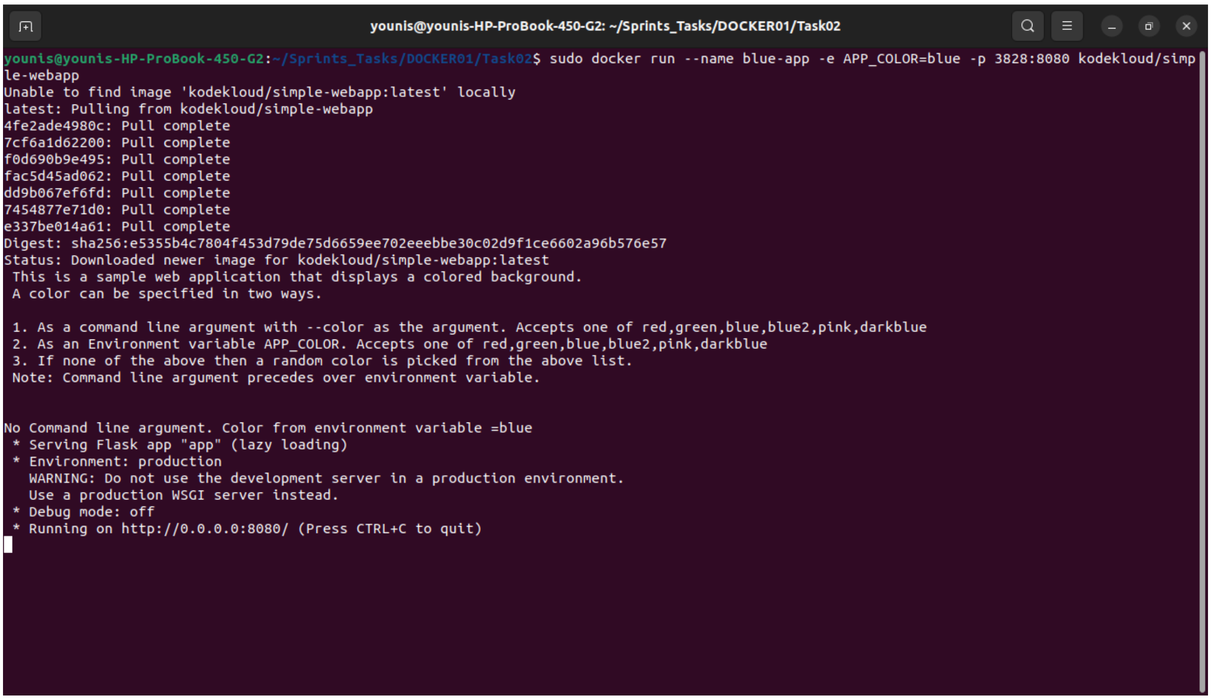
Create ubuntu image and check the size of it



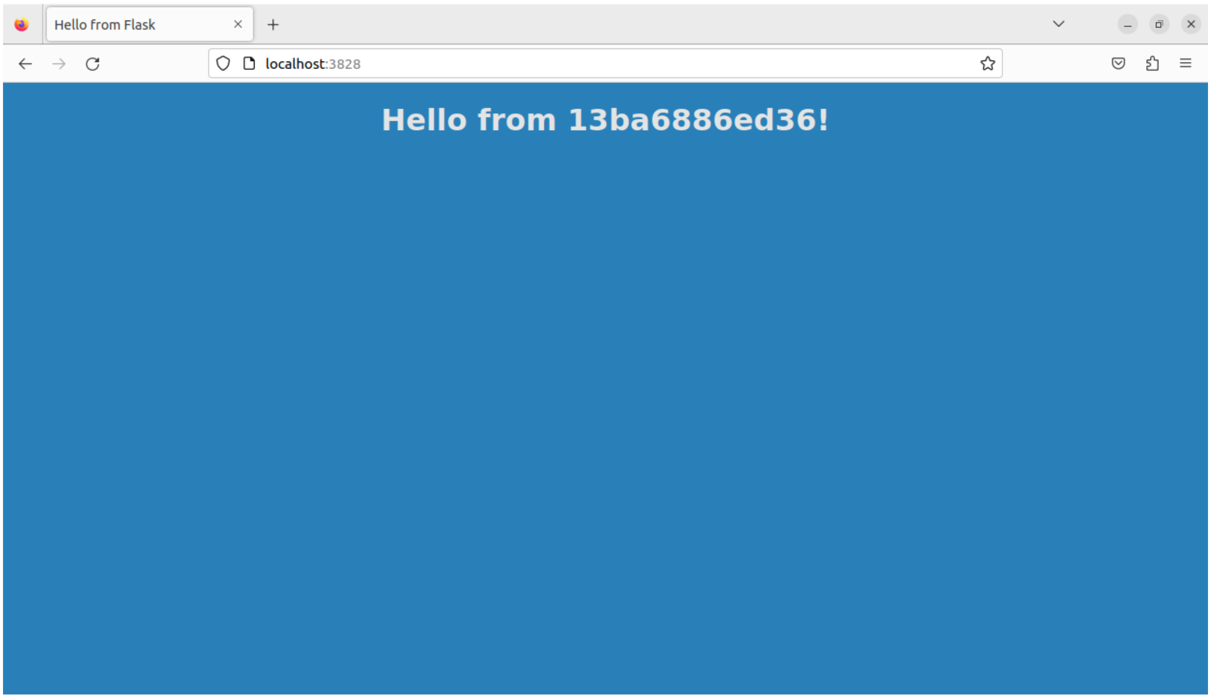
younis@younis-HP-ProBook-450-G2: ~/Sprints_Tasks/DOCKER01/Task02

```
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker run ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
dbf6a9befcde: Already exists
Digest: sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee20739e8eb54fbf
Status: Downloaded newer image for ubuntu:latest
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker images
REPOSITORY                        TAG        IMAGE ID       CREATED          SIZE
dockerize-node-app-task_node-app  latest     9b46896378f2   21 hours ago     915MB
apache-image                      latest     6b9f21873a4a   22 hours ago     145MB
nginx                             alpine     fe7edaf8a8dc   6 days ago       41.4MB
nginx                             latest     f9c14fe76d50   6 days ago       143MB
mysql                             latest     05db07cd74c0   7 days ago       565MB
ubuntu                            latest     3b418d7b466a   5 weeks ago      77.8MB
mysql                             5.7        dd6675b5cfea   6 weeks ago      569MB
ghost                             1-alpine   efcd4044e3a0   3 years ago      268MB
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$
```

Run a container named blue-app using image KodeKloud/simple-webapp and set the environmet variable APP_COLOR to blue. make the app available on port 3828 on the host. the app listens on port 8080

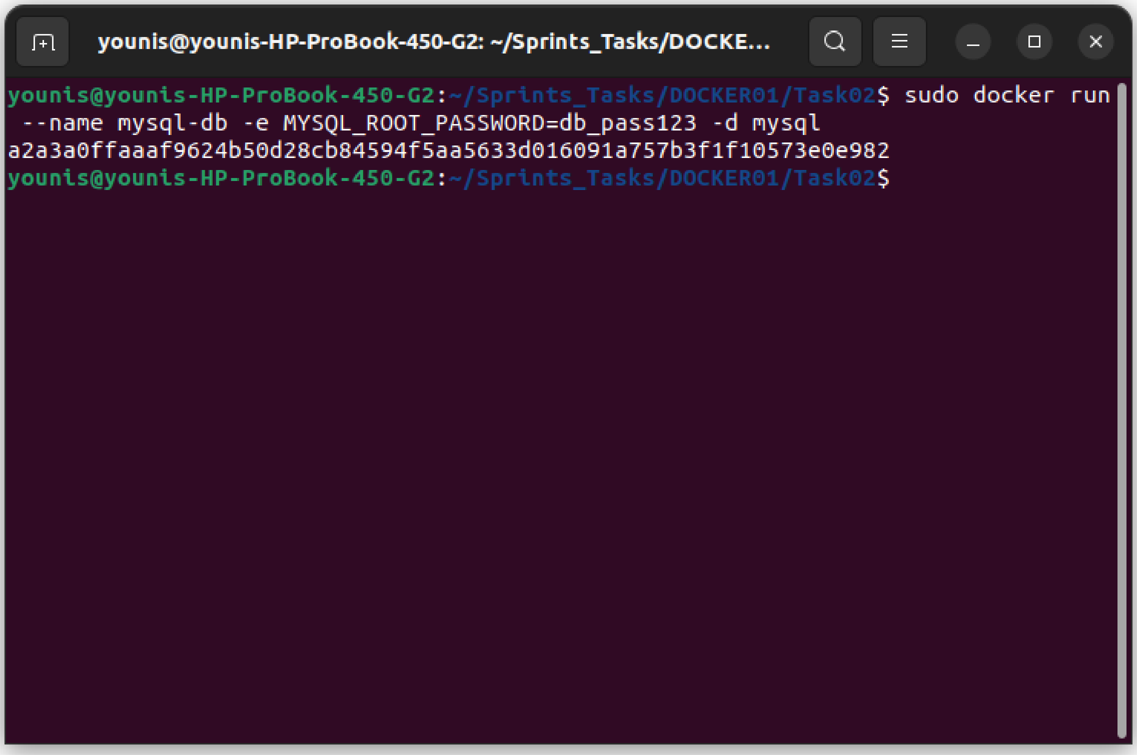

younis@younis-HP-ProBook-450-G2: ~/Sprints_Tasks/DOCKER01/Task02

```
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker run --name blue-app -e APP_COLOR=blue -p 3828:8080 kodekloud/simp
le-webapp
Unable to find image 'kodekloud/simple-webapp:latest' locally
latest: Pulling from kodekloud/simple-webapp
4fe2ade4980c: Pull complete
7cf6a1d62200: Pull complete
f0d690b9e495: Pull complete
fac5d45ad062: Pull complete
dd9b067ef6fd: Pull complete
7454877e71d0: Pull complete
e337be014a61: Pull complete
Digest: sha256:e5355b4c7804f453d79de75d6659ee702eeebbe30c02d9f1ce6602a96b576e57
Status: Downloaded newer image for kodekloud/simple-webapp:latest
 This is a sample web application that displays a colored background.
 A color can be specified in two ways.

 1. As a command line argument with --color as the argument. Accepts one of red,green,blue,blue2,pink,darkblue
 2. As an Environment variable APP_COLOR. Accepts one of red,green,blue,blue2,pink,darkblue
 3. If none of the above then a random color is picked from the above list.
 Note: Command line argument precedes over environment variable.


No Command line argument. Color from environment variable =blue
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

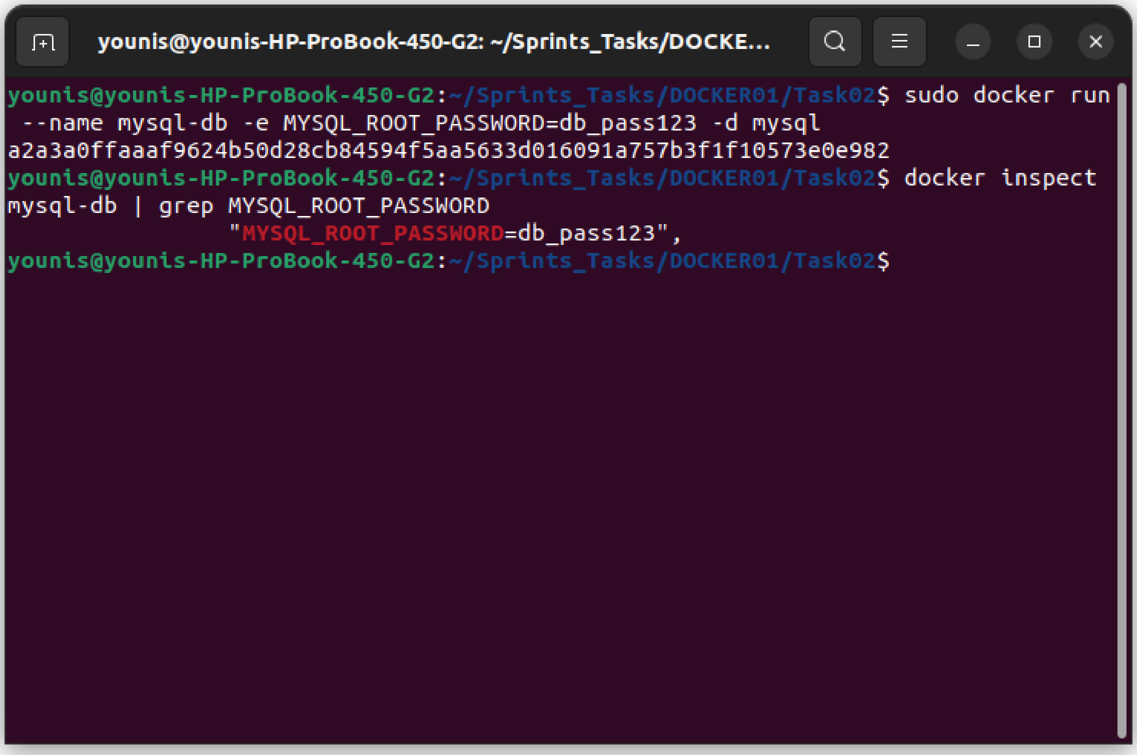


Hello from 13ba6886ed36!

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

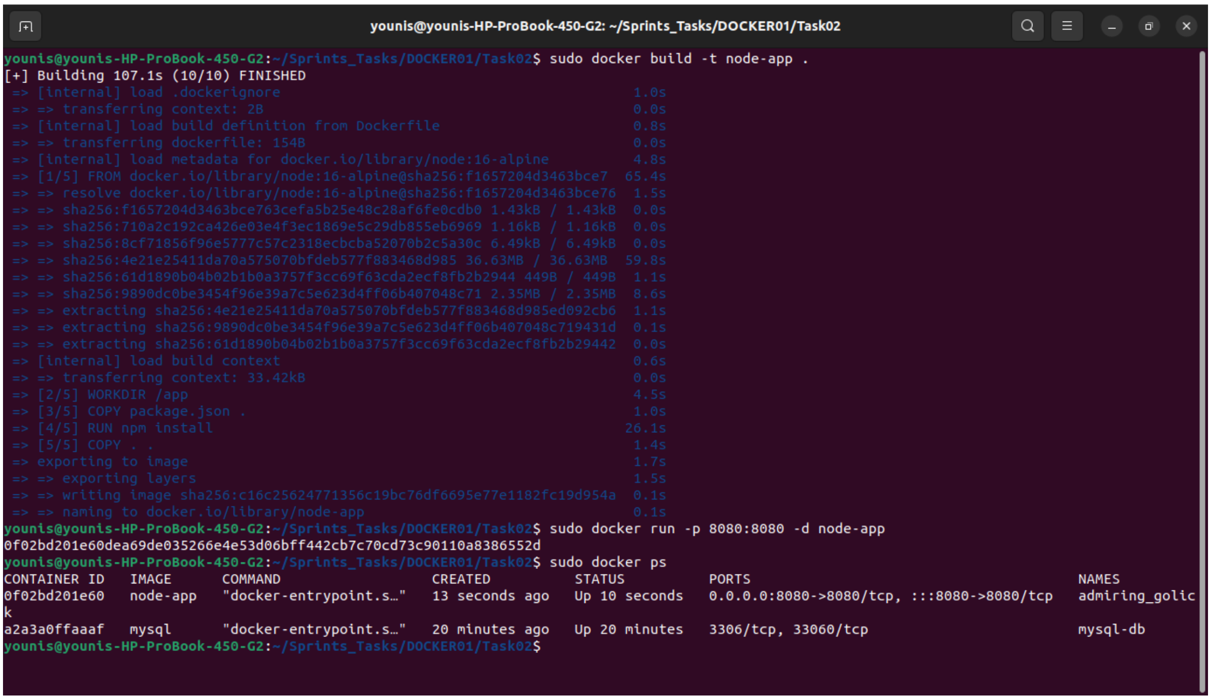


```
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker run
 --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 -d mysql
a2a3a0ffaaaf9624b50d28cb84594f5aa5633d016091a757b3f1f10573e0e982
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$
```



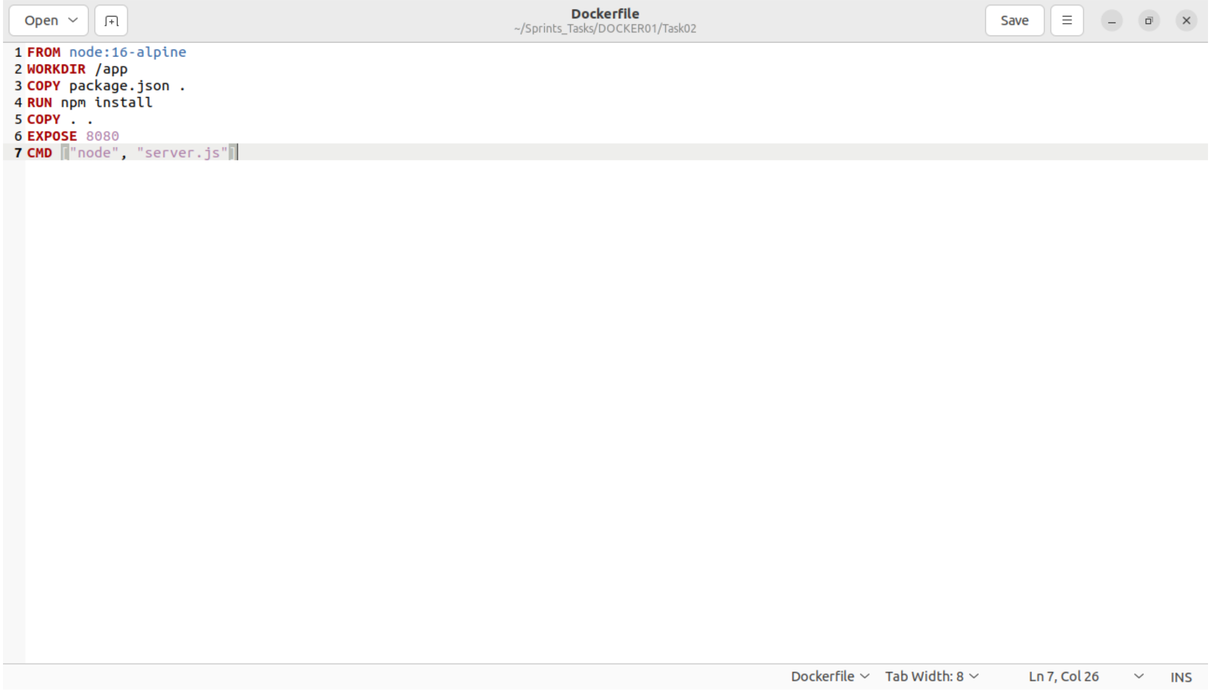
```
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker run
 --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 -d mysql
a2a3a0ffaaaf9624b50d28cb84594f5aa5633d016091a757b3f1f10573e0e982
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ docker inspect
mysql-db | grep MYSQL_ROOT_PASSWORD
                "MYSQL_ROOT_PASSWORD=db_pass123",
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$
```

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

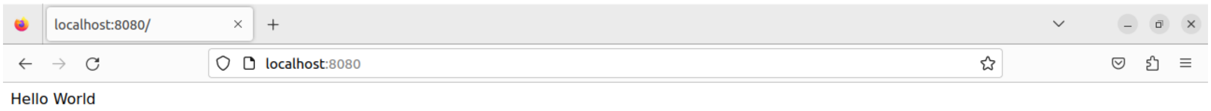


younis@younis-HP-ProBook-450-G2: ~/Sprints_Tasks/DOCKER01/Task02

```
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker build -t node-app .
[+] Building 107.1s (10/10) FINISHED
 => [internal] load .dockerignore                                          1.0s
 => => transferring context: 2B                                            0.0s
 => [internal] load build definition from Dockerfile                       0.8s
 => => transferring dockerfile: 154B                                       0.0s
 => [internal] load metadata for docker.io/library/node:16-alpine          4.8s
 => [1/5] FROM docker.io/library/node:16-alpine@sha256:f1657204d3463bce7  65.4s
 => => resolve docker.io/library/node:16-alpine@sha256:f1657204d3463bce76  1.5s
 => => sha256:f1657204d3463bce763cefa5b25e48c28af6fe0cdb0 1.43kB / 1.43kB  0.0s
 => => sha256:710a2c192ca426e03e4f3ec1869e5c29db855eb6969 1.16kB / 1.16kB  0.0s
 => => sha256:8cf71856f96e5777c57c2318ecbcba52070b2c5a30c 6.49kB / 6.49kB  0.0s
 => => sha256:4e21e25411da70a575070bfdeb577f883468d985 36.63MB / 36.63MB  59.8s
 => => sha256:61d1890b04b02b1b0a3757f3cc69f63cda2ecf8fb2b2944 449B / 449B  1.1s
 => => sha256:9890dc0be3454f96e39a7c5e623d4ff06b407048c71 2.35MB / 2.35MB  8.6s
 => => extracting sha256:4e21e25411da70a575070bfdeb577f883468d985ed092cb6  1.1s
 => => extracting sha256:9890dc0be3454f96e39a7c5e623d4ff06b407048c719431d  0.1s
 => => extracting sha256:61d1890b04b02b1b0a3757f3cc69f63cda2ecf8fb2b29442  0.0s
 => [internal] load build context                                          0.6s
 => => transferring context: 33.42kB                                       0.0s
 => [2/5] WORKDIR /app                                                     4.5s
 => [3/5] COPY package.json .                                              1.0s
 => [4/5] RUN npm install                                                 26.1s
 => [5/5] COPY . .                                                         1.4s
 => exporting to image                                                     1.7s
 => => exporting layers                                                    1.5s
 => => writing image sha256:c16c25624771356c19bc76df6695e77e1182fc19d954a  0.1s
 => => naming to docker.io/library/node-app                                0.1s
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker run -p 8080:8080 -d node-app
0f02bd201e60dea69de035266e4e53d06bff442cb7c70cd73c90110a8386552d
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS          PORTS                                       NAMES
0f02bd201e60   node-app   "docker-entrypoint.s…"   13 seconds ago   Up 10 seconds   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   admiring_golic
k
a2a3a0ffaaaf   mysql      "docker-entrypoint.s…"   20 minutes ago   Up 20 minutes   3306/tcp, 33060/tcp                         mysql-db
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$
```

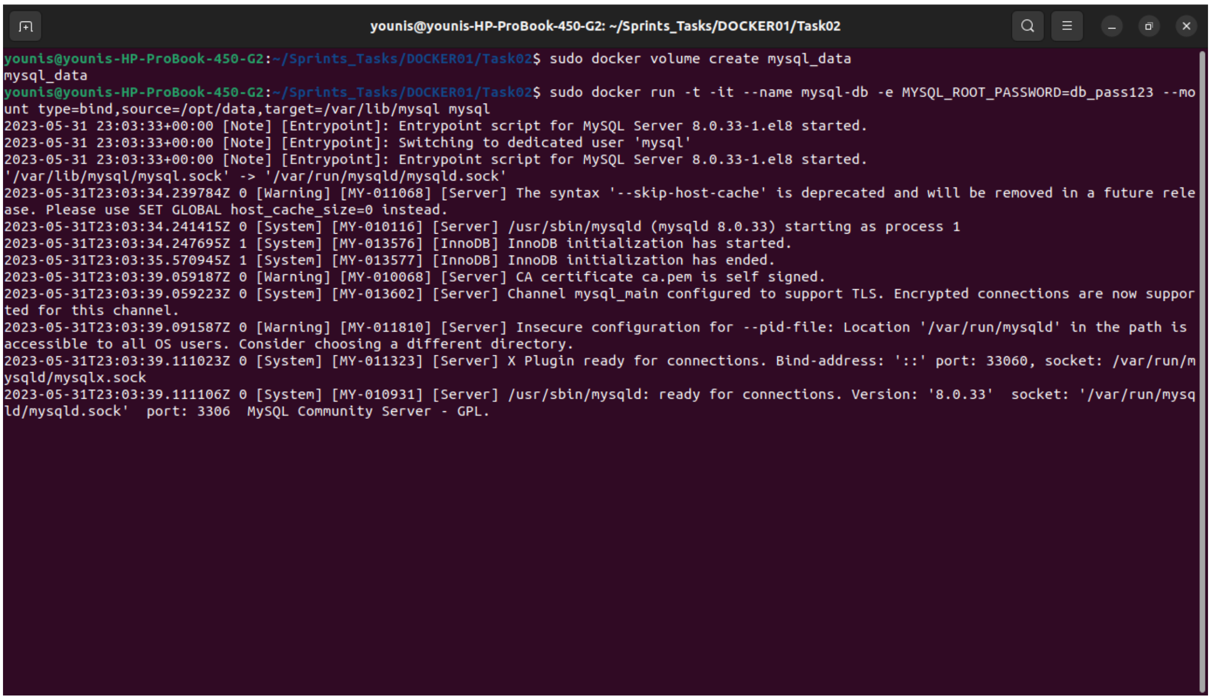


```
FROM node:16-alpine
WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
EXPOSE 8080
CMD ["node", "server.js"]
```

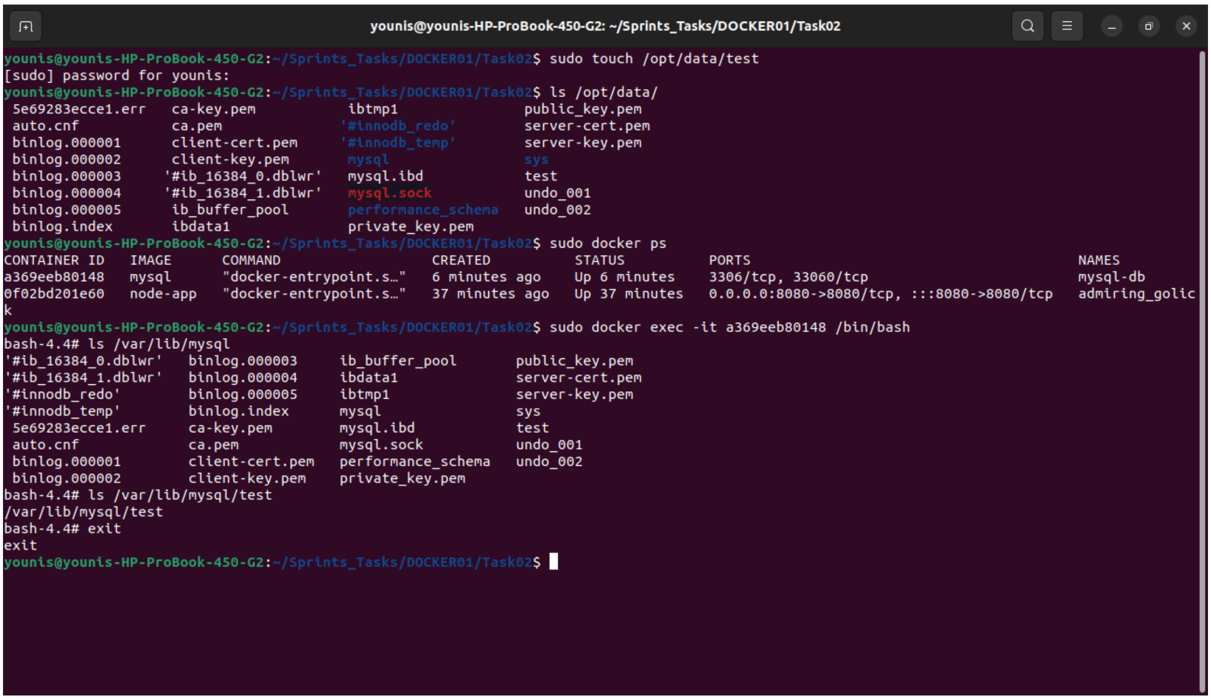


Hello World

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 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


younis@younis-HP-ProBook-450-G2: ~/Sprints_Tasks/DOCKER01/Task02

```
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker volume create mysql_data
mysql_data
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker run -t -it --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 --mo
unt type=bind,source=/opt/data,target=/var/lib/mysql mysql
2023-05-31 23:03:33+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.33-1.el8 started.
2023-05-31 23:03:33+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2023-05-31 23:03:33+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.33-1.el8 started.
'/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
2023-05-31T23:03:34.239784Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future rele
ase. Please use SET GLOBAL host_cache_size=0 instead.
2023-05-31T23:03:34.241415Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.33) starting as process 1
2023-05-31T23:03:34.247695Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2023-05-31T23:03:35.570945Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2023-05-31T23:03:39.059187Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2023-05-31T23:03:39.059223Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now suppor
ted for this channel.
2023-05-31T23:03:39.091587Z 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.
2023-05-31T23:03:39.111023Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/m
ysqld/mysqlx.sock
2023-05-31T23:03:39.111106Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.33'  socket: '/var/run/mysq
ld/mysqld.sock'  port: 3306  MySQL Community Server - GPL.
```



younis@younis-HP-ProBook-450-G2: ~/Sprints_Tasks/DOCKER01/Task02

```
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo touch /opt/data/test
[sudo] password for younis:
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ ls /opt/data/
 5e69283ecce1.err   ca-key.pem            ibtmp1               public_key.pem
 auto.cnf           ca.pem               '#innodb_redo'        server-cert.pem
 binlog.000001      client-cert.pem      '#innodb_temp'        server-key.pem
 binlog.000002      client-key.pem        mysql                sys
 binlog.000003     '#ib_16384_0.dblwr'    mysql.ibd            test
 binlog.000004     '#ib_16384_1.dblwr'    mysql.sock           undo_001
 binlog.000005      ib_buffer_pool        performance_schema   undo_002
 binlog.index       ibdata1               private_key.pem
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS          PORTS                                       NAMES
a369eeb80148   mysql      "docker-entrypoint.s…"   6 minutes ago    Up 6 minutes    3306/tcp, 33060/tcp                         mysql-db
0f02bd201e60   node-app   "docker-entrypoint.s…"   37 minutes ago   Up 37 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   admiring_golic
k
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$ sudo docker exec -it a369eeb80148 /bin/bash
bash-4.4# ls /var/lib/mysql
'#ib_16384_0.dblwr'   binlog.000003      ib_buffer_pool       public_key.pem
'#ib_16384_1.dblwr'   binlog.000004      ibdata1              server-cert.pem
'#innodb_redo'        binlog.000005      ibtmp1               server-key.pem
'#innodb_temp'        binlog.index       mysql                sys
 5e69283ecce1.err     ca-key.pem         mysql.ibd            test
 auto.cnf             ca.pem             mysql.sock           undo_001
 binlog.000001        client-cert.pem    performance_schema   undo_002
 binlog.000002        client-key.pem     private_key.pem
bash-4.4# ls /var/lib/mysql/test
/var/lib/mysql/test
bash-4.4# exit
exit
younis@younis-HP-ProBook-450-G2:~/Sprints_Tasks/DOCKER01/Task02$
```