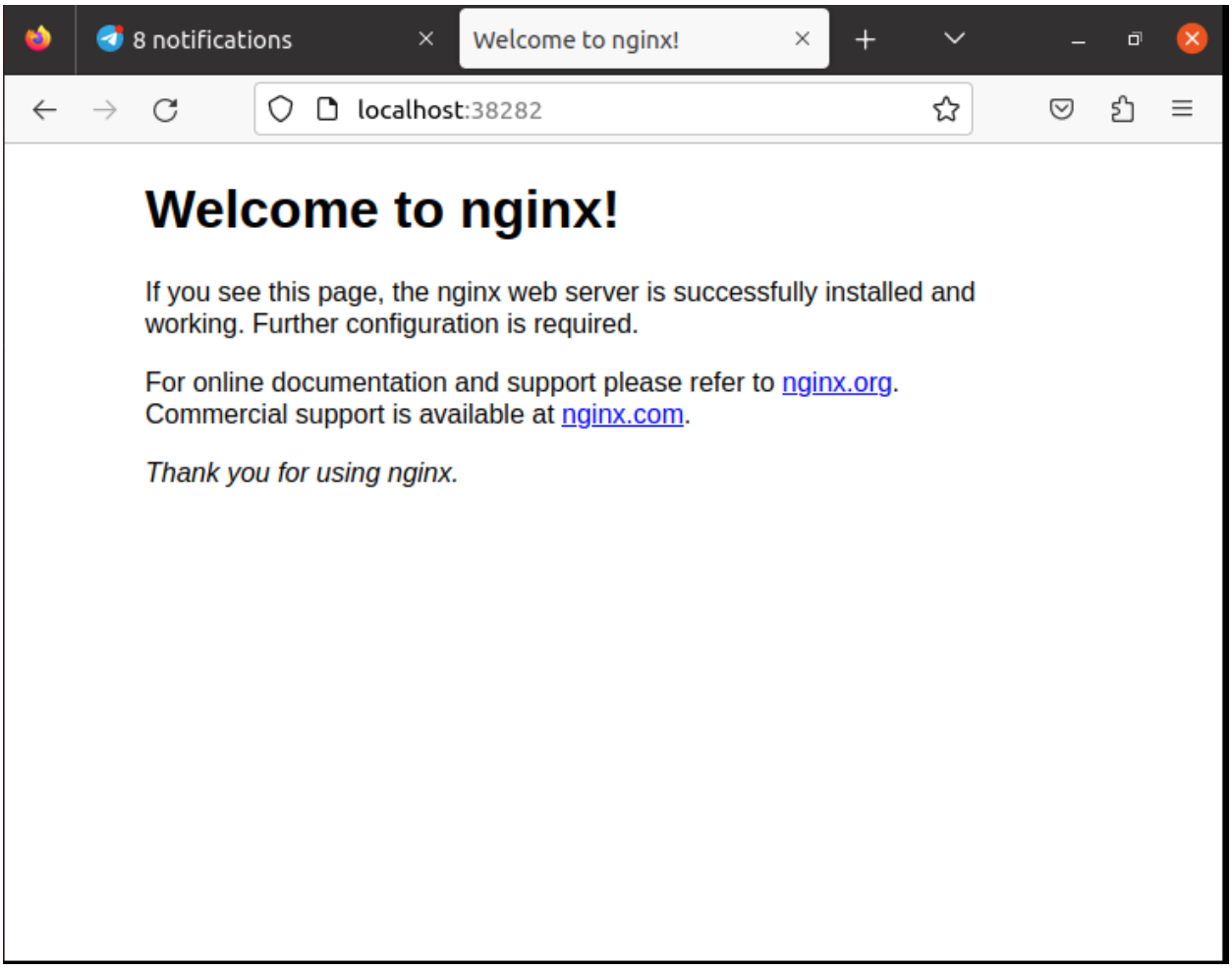
11- Run an instance of nginx:alpine with a name nginx and map port 8080 on the container to 38282 on the host

```
doha@ubuntu:~$ sudo docker run -d -p 38282:80 nginx:alpine
Unable to find image 'nginx:alpine' locally
alpine: Pulling from library/nginx
f56be85fc22e: Pull complete
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
de0b496056c574a86b2adbd3c54084c233824e0c8c7fc19e83737d6f602d504a
doha@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
         PORTS                                     NAMES
de0b496056c5   nginx:alpine   "/docker-entrypoint.…"   30 seconds ago   Up 23 s
econds   0.0.0.0:38282->80/tcp, :::38282->80/tcp   happy_sutherland
doha@ubuntu:~$
```



12- Create ubuntu image and check the size of it

```
doha@ubuntu:~$ sudo docker run -d ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
837dd4791cdc: Pull complete
Digest: sha256:ac58ff7fe25edc58bdf0067ca99df00014dbd032e2246d30a722fa348fd799a5
Status: Downloaded newer image for ubuntu:latest
19eca7c830fbc4dbb41cf46380b586b1a9690c65493c4e3657971a55b3ced83c
doha@ubuntu:~$ sudo docker images
REPOSITORY   TAG        IMAGE ID       CREATED       SIZE
<none>       <none>     4845f34b2d85   4 days ago    909MB
nginx        alpine     fe7edaf8a8dc   10 days ago   41.4MB
mysql        8.0        05db07cd74c0   10 days ago   565MB
node         16         4fbbe8e45ea1   12 days ago   909MB
ubuntu       latest     1f6ddc1b2547   12 days ago   77.8MB
mysql        5.7        dd6675b5cfea   6 weeks ago   569MB
ghost        1-alpine   efcd4044e3a0   3 years ago   268MB
doha@ubuntu:~$
```

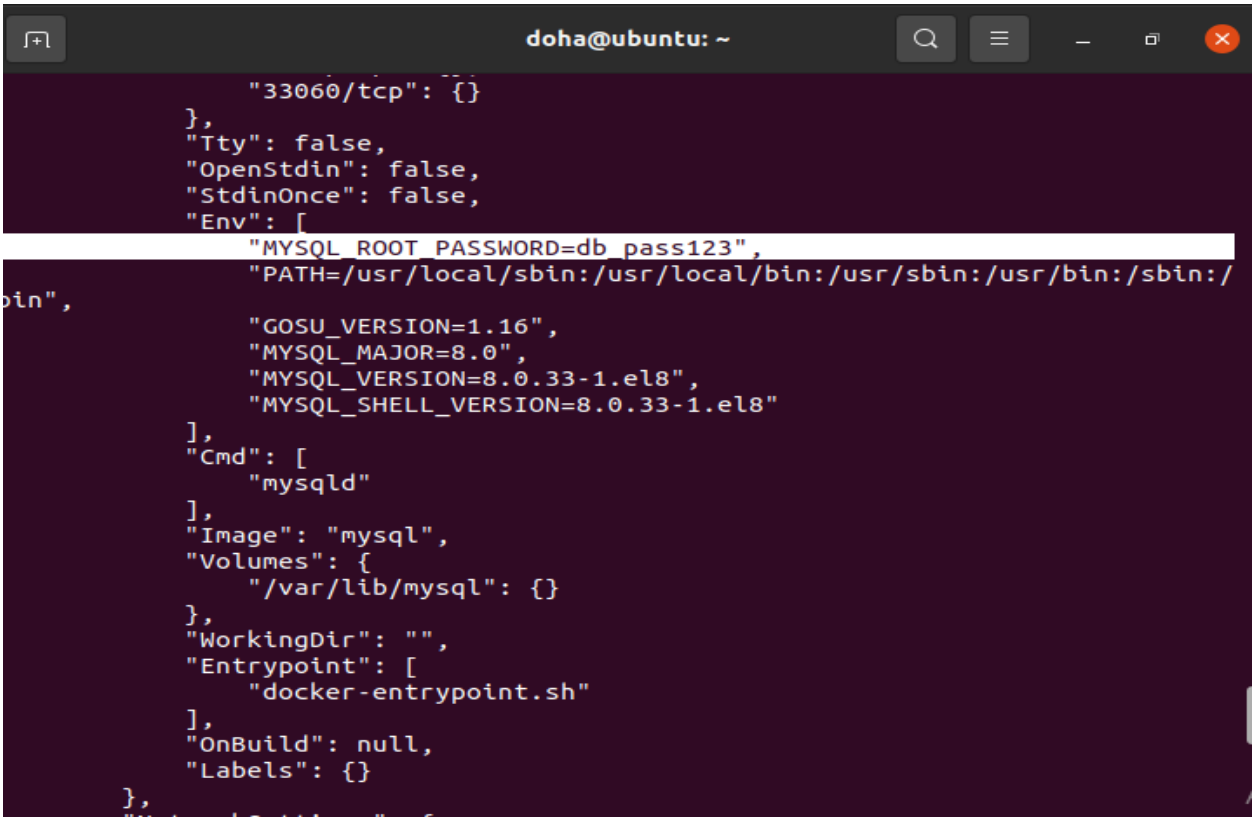
13- Run a container named blue-app using image kodekloud/simple-webapp and set the environment variable APP_COLOR to blue. Make the application available on port 39282 on the host. The application listens on port 8080.

```
doha@ubuntu:~$ sudo docker run --name blue-app -e APP_COLOR=blue -p 38283:8080
-d kodekloud/simple-webapp
a6a412e4c492ee03812edaf402e7def09f71be0811d125292a677a0a98facd4d
doha@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED
    STATUS          PORTS                                          NAMES
a6a412e4c492   kodekloud/simple-webapp   "python app.py"          48 seconds ag
o   Up 43 seconds   0.0.0.0:38283->8080/tcp, :::38283->8080/tcp   blue-app
de0b496056c5   nginx:alpine             "/docker-entrypoint.…"   16 minutes ag
o   Up 16 minutes   0.0.0.0:38282->80/tcp, :::38282->80/tcp       happy_sutherl
and
doha@ubuntu:~$
```

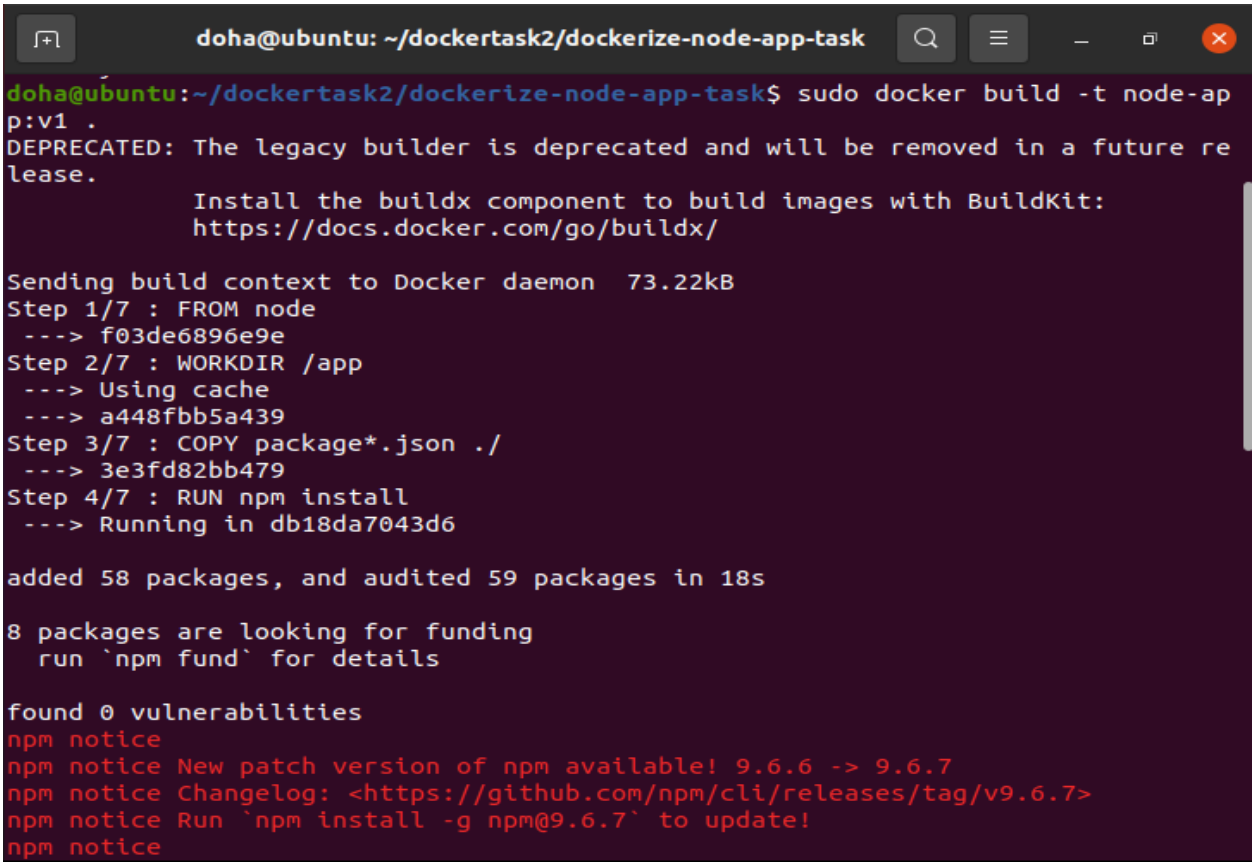
14- deploy a mysql database using the mysql image and name it mysql-db. Set the database password to use db_pass123 then inspect it to check the value.

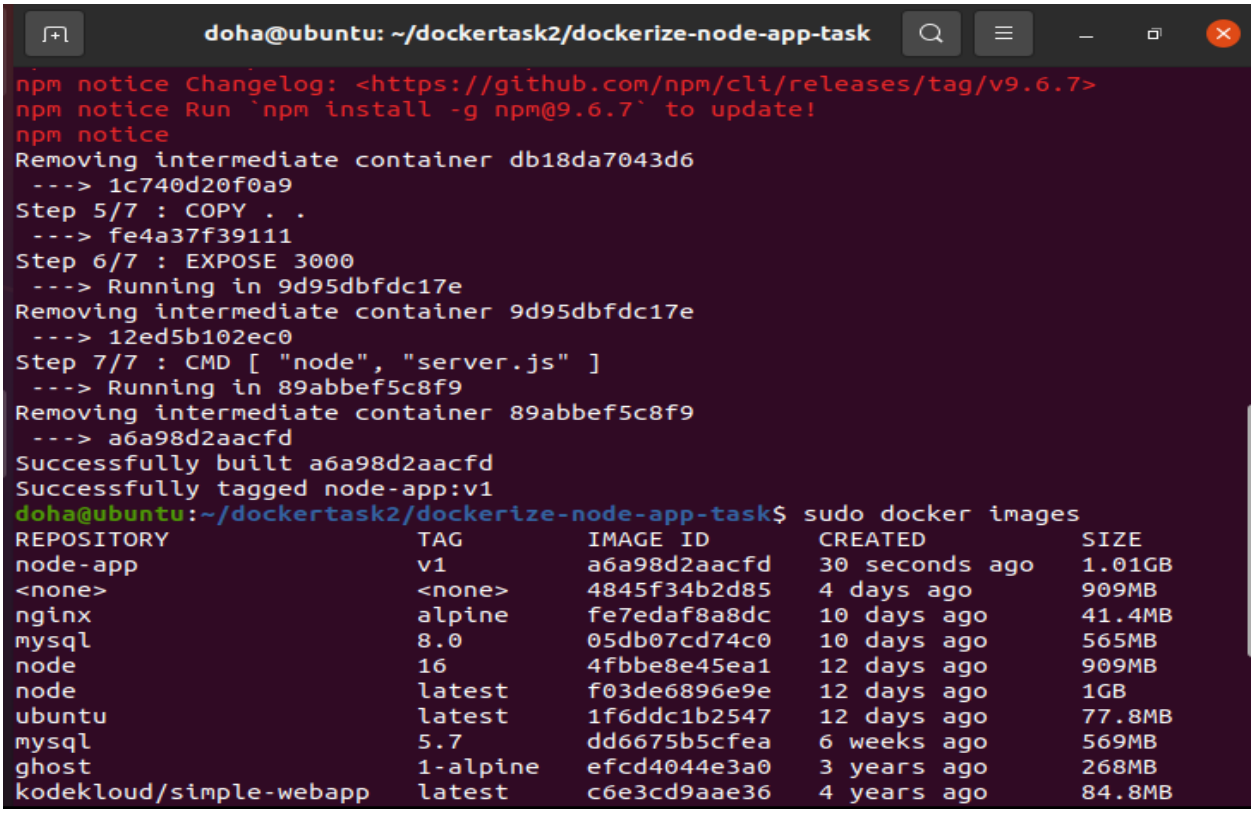
```
doha@ubuntu:~$ sudo docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass12
3 -d mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
90e2fb2facff: Pull complete
ba60eb20fd5f: Pull complete
4f509402d469: Pull complete
496c2cfa6815: Pull complete
8ec1dfa9522c: Pull 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
e93344c81a9a046aabf0e9d1a077205b3d697ff0f3cfd8c996bb5da9ebe7acb2
doha@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED
    STATUS          PORTS                                          NAMES
e93344c81a9a   mysql                    "docker-entrypoint.s…"   18 seconds ag
o   Up 12 seconds   3306/tcp, 33060/tcp                           mysql-db
9ca87f36f199   kodekloud/simple-webapp   "python app.py"          7 minutes ago
    Up 7 minutes    0.0.0.0:38282->8080/tcp, :::38282->8080/tcp   blue-app
doha@ubuntu:~$
```

```
doha@ubuntu:~$ sudo docker inspect e93344c81a9a
[sudo] password for doha:
[
    {
        "Id": "e93344c81a9a046aabf0e9d1a077205b3d697ff0f3cfd8c996bb5da9ebe7acb2
",
        "Created": "2023-05-26T23:55:49.437873957Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "mysqld"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 23851,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2023-05-26T23:55:54.984536358Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:05db07cd74c0520c8ffe5f7638063719a886f9115cecacc0654d98
1caf5d27f7",
        "ResolvConfPath": "/var/lib/docker/containers/e93344c81a9a046aabf0e9d1a
077205b3d697ff0f3cfd8c996bb5da9ebe7acb2/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/e93344c81a9a046aabf0e9d1a07
```

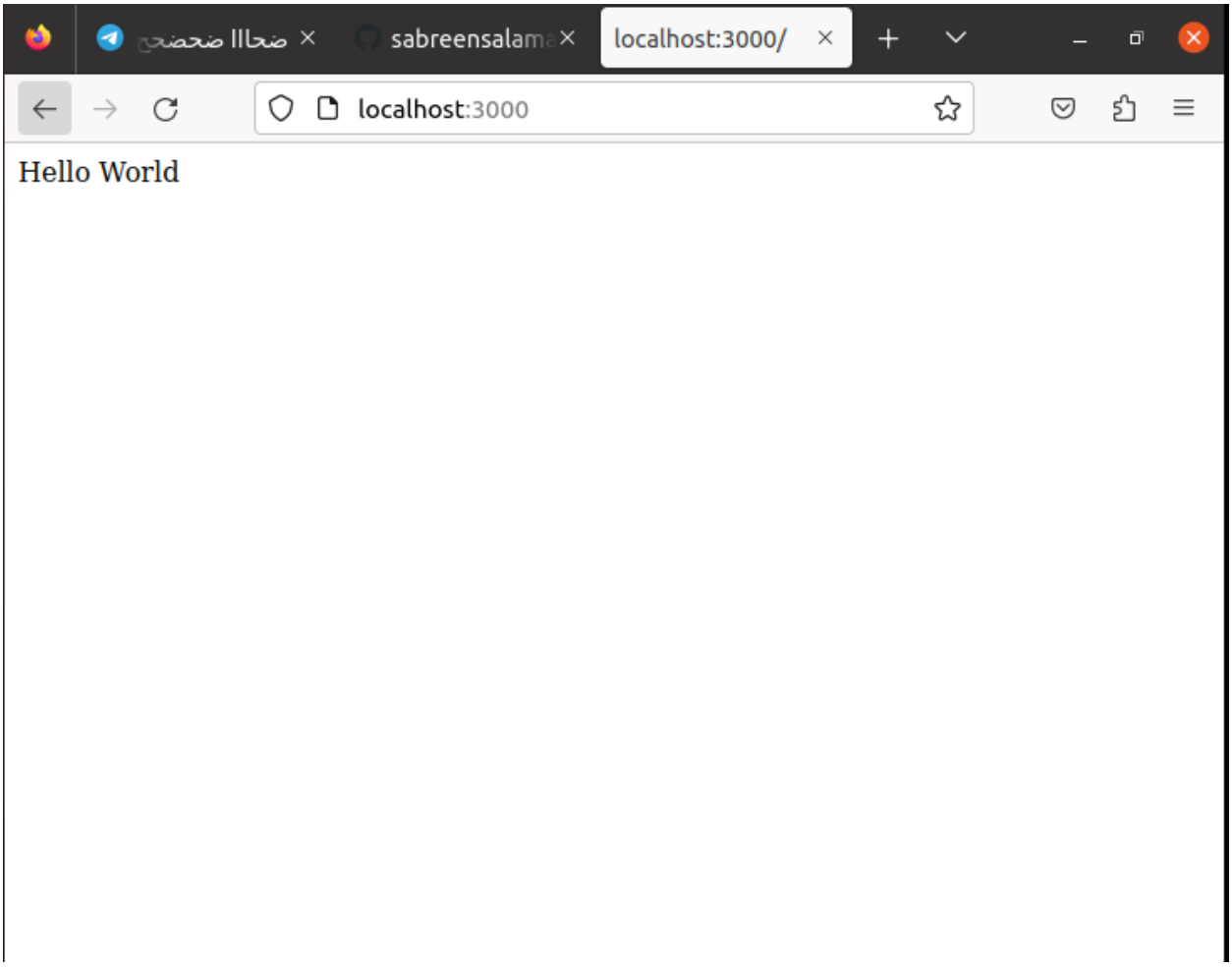


5- Pull the code from https://github.com/sabreensalama/dockerize-node-app-task and create a docker file for this node app.


doha@ubuntu: ~/dockertask2/dockerize-node-app-task
doha@ubuntu:~/dockertask2/dockerize-node-app-task$ sudo docker build -t node-app:v1 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/
Sending build context to Docker daemon 73.22kB
Step 1/7 : FROM node
---> f03de6896e9e
Step 2/7 : WORKDIR /app
---> Using cache
---> a448fbb5a439
Step 3/7 : COPY package*.json ./
---> 3e3fd82bb479
Step 4/7 : RUN npm install
---> Running in db18da7043d6
added 58 packages, and audited 59 packages in 18s
8 packages are looking for funding
run `npm fund` for details
found 0 vulnerabilities
npm notice
npm notice New patch version of npm available! 9.6.6 -> 9.6.7
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.7>
npm notice Run `npm install -g npm@9.6.7` to update!
npm notice


doha@ubuntu: ~/dockertask2/dockerize-node-app-task
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.7>
npm notice Run `npm install -g npm@9.6.7` to update!
npm notice
Removing intermediate container db18da7043d6
---> 1c740d20f0a9
Step 5/7 : COPY . .
---> fe4a37f39111
Step 6/7 : EXPOSE 3000
---> Running in 9d95dbfdc17e
Removing intermediate container 9d95dbfdc17e
---> 12ed5b102ec0
Step 7/7 : CMD ["node", "server.js"]
---> Running in 89abbef5c8f9
Removing intermediate container 89abbef5c8f9
---> a6a98d2aacfd
Successfully built a6a98d2aacfd
Successfully tagged node-app:v1
doha@ubuntu:~/dockertask2/dockerize-node-app-task$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
node-app v1 a6a98d2aacfd 30 seconds ago 1.01GB
<none> <none> 4845f34b2d85 4 days ago 909MB
nginx alpine fe7edaf8a8dc 10 days ago 41.4MB
mysql 8.0 05db07cd74c0 10 days ago 565MB
node 16 4fbbe8e45ea1 12 days ago 909MB
node latest f03de6896e9e 12 days ago 1GB
ubuntu latest 1f6ddc1b2547 12 days ago 77.8MB
mysql 5.7 dd6675b5cfea 6 weeks ago 569MB
ghost 1-alpine efcd4044e3a0 3 years ago 268MB
kodekloud/simple-webapp latest c6e3cd9aae36 4 years ago 84.8MB

```
doha@ubuntu:~/dockertask2/dockerize-node-app-task$ sudo docker run -p 3000:8080
 -d node-app:v1
407f4bc96943e787cd95db3654fa2bfd92e7cce283af24146ecca4a9ac5f45e6
doha@ubuntu:~/dockertask2/dockerize-node-app-task$ sudo docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED
    STATUS          PORTS                                                  NAMES
407f4bc96943   node-app:v1              "docker-entrypoint.s…"   13 seconds ag
o   Up 11 seconds   3000/tcp, 0.0.0.0:3000->8080/tcp, :::3000->8080/tcp   seren
e_albattani
a6a412e4c492   kodekloud/simple-webapp   "python app.py"          22 minutes ag
o   Up 22 minutes   0.0.0.0:38283->8080/tcp, :::38283->8080/tcp           blue-
app
de0b496056c5   nginx:alpine             "/docker-entrypoint.…"   38 minutes ag
o   Up 38 minutes   0.0.0.0:38282->80/tcp, :::38282->80/tcp               happy
_sutherland
doha@ubuntu:~/dockertask2/dockerize-node-app-task$
```



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

```
FROM python:3.9-alpine

WORKDIR /app

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY . .

EXPOSE 5000

CMD { python , "app.py"}
~
~
~
~
~
~
~
~
~
~
~
~
~
"Dockerfile" 13L, 150C                                    7,35          All
```

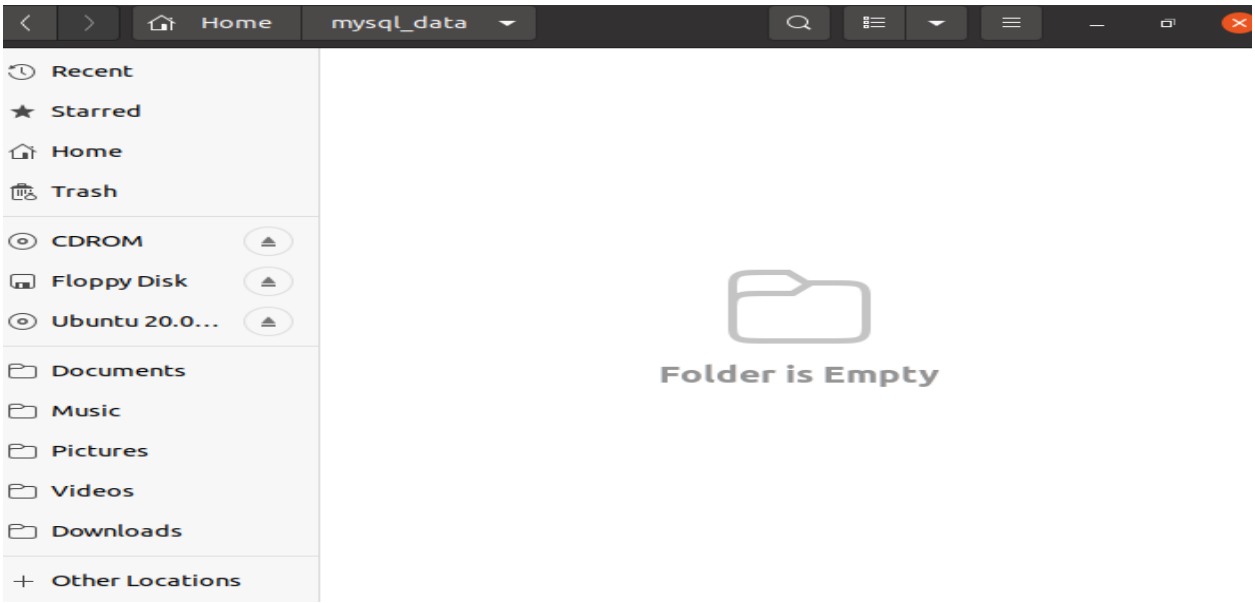
```
doha@ubuntu:~/simple-flask-app$ sudo docker build -t simple-flash-app .
Sending build context to Docker daemon  73.73kB
Step 1/7 : FROM python:3.9-alpine
 ---> 3e25ee8eaf10
Step 2/7 : WORKDIR /app
 ---> Using cache
 ---> 4dba60632bee
Step 3/7 : COPY requirements.txt .
 ---> Using cache
 ---> 4e996f193ed4
Step 4/7 : RUN pip install -r requirements.txt
 ---> Using cache
 ---> 1f00e3bfe4d0
Step 5/7 : COPY . .
 ---> Using cache
 ---> d0ba9ce4ce2f
Step 6/7 : EXPOSE 5000
 ---> Using cache
 ---> d53b430ac336
Step 7/7 : CMD { python , "app.py"}
 ---> Using cache
 ---> dc1151128dc8
Successfully built dc1151128dc8
Successfully tagged simple-flash-app:latest
doha@ubuntu:~/simple-flask-app$
```

```
doha@ubuntu:~/simple-flask-app$ sudo docker images
REPOSITORY                TAG             IMAGE ID       CREATED          SIZE
simple-flash-app          latest          dc1151128dc8   7 minutes ago    61MB
nginx                     alpine          fe7edaf8a8dc   2 days ago       41.4MB
mysql                     latest          05db07cd74c0   2 days ago       565MB
python                    3.9-alpine      3e25ee8eaf10   2 weeks ago      48.2MB
hello-world               latest          9c7a54a9a43c   3 weeks ago      13.3kB
nginx                     latest          448a08f1d2f9   3 weeks ago      142MB
ubuntu                    latest          3b418d7b466a   4 weeks ago      77.8MB
nginx                     1.24.0-alpine   1266a3a46e96   6 weeks ago      41.1MB
image1                    1.0             0dcda3f22e52   3 months ago     119MB
doha                      2.0             0dcda3f22e52   3 months ago     119MB
doha                      3.0             0dcda3f22e52   3 months ago     119MB
doha                      latest          0dcda3f22e52   3 months ago     119MB
ubuntu                    <none>          58db3edaf2be   4 months ago     77.8MB
nginx                     1.14-alpine     8a2fb25a19f5   4 years ago      16MB
kodekloud/simple-webapp   latest          c6e3cd9aae36   4 years ago      84.8MB
doha@ubuntu:~/simple-flask-app$
```

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.

1-create mysql_data

```
doha@ubuntu:~$ sudo docker volume create mysql_data
[sudo] password for doha:
mysql_data
```



2-Run the container , and create a directory called ‘From_container’ in it

```
doha@ubuntu:~$ sudo docker volume create mysql_data
[sudo] password for doha:
mysql_data
doha@ubuntu:~$ sudo docker run -d --name mysql_data -e MYSEQ_ROOT_PASSWORD=db_p
ass123 -v /opt/data/mysql_data:/var/lib/mysql mysql sleep 1000
e6627220dc63e0d23217f97bca2031c6832799552f53692ac0101030e53ce3ff
doha@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED
    STATUS          PORTS                                       NAMES
e6627220dc63   mysql                    "docker-entrypoint.s…"   19 seconds ag
o   Up 13 seconds   3306/tcp, 33060/tcp                         mysql_data
41c2d3ea2b13   kodekloud/simple-webapp   "python app.py"          2 hours ago
    Up 2 hours      0.0.0.0:38282->8080/tcp, :::38282->8080/tcp   blue-app
e93344c81a9a   mysql                    "docker-entrypoint.s…"   3 hours ago
    Up 3 hours      3306/tcp, 33060/tcp                         mysql-db
doha@ubuntu:~$ sudo docker stop e93344c81a9a
e93344c81a9a
doha@ubuntu:~$ sudo docker rm e93344c81a9a
e93344c81a9a
doha@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED
        STATUS              PORTS                                       NAMES
e6627220dc63   mysql                    "docker-entrypoint.s…"   About a minut
e ago   Up About a minute   3306/tcp, 33060/tcp                         mysql
_data
41c2d3ea2b13   kodekloud/simple-webapp   "python app.py"          2 hours ago
        Up 2 hours          0.0.0.0:38282->8080/tcp, :::38282->8080/tcp   blue-
app
doha@ubuntu:~$
```

3-The directory 'From_container' was created in mysql_data directory.

```
doha@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED
        STATUS              PORTS                                       NAMES
e6627220dc63   mysql                    "docker-entrypoint.s…"   About a minut
e ago   Up About a minute   3306/tcp, 33060/tcp                         mysql
_data
41c2d3ea2b13   kodekloud/simple-webapp   "python app.py"          2 hours ago
        Up 2 hours          0.0.0.0:38282->8080/tcp, :::38282->8080/tcp   blue-
app
doha@ubuntu:~$ sudo docker exec -it e6627220dc63 /bin/bash
bash-4.4# ls
bin   docker-entrypoint-initdb.d  home   media  proc  sbin  tmp
boot  entrypoint.sh               lib    mnt    root  srv   usr
dev   etc                         lib64  opt    run   sys   var
bash-4.4# cd /var/lib/mysql
bash-4.4# mkdir From_container
bash-4.4# ls
From_container
bash-4.4# exit
exit
doha@ubuntu:~$
```

4-Back agian and stop the container then delete it.

```
doha@ubuntu:~$ sudo docker exec -it e6627220dc63 /bin/bash
bash-4.4# ls
bin   docker-entrypoint-initdb.d  home   media  proc  sbin  tmp
boot  entrypoint.sh               lib    mnt    root  srv   usr
dev   etc                         lib64  opt    run   sys   var
bash-4.4# cd /var/lib/mysql
bash-4.4# mkdir From_container
bash-4.4# ls
From_container
bash-4.4# exit
exit
doha@ubuntu:~$ sudo docker stop e6627220dc63
e6627220dc63
doha@ubuntu:~$ sudo docker rm e6627220dc63
e6627220dc63
doha@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE                     COMMAND            CREATED       STATUS
      PORTS                                          NAMES
41c2d3ea2b13   kodekloud/simple-webapp   "python app.py"    2 hours ago   Up 2 h
ours   0.0.0.0:38282->8080/tcp, :::38282->8080/tcp   blue-app
```

5-After delete the container , The directory 'From_container' still in mysql_data directory.

```
doha@ubuntu:~$ sudo docker run -d --name mysql_db_2 -e MYSEQ_ROOT_PASSWORD=db_p
ass123 -v /opt/data/mysql_data:/var/lib/mysql mysql sleep 1000
6cc804cc575db1dab1cceedd9f40063e8a9a47054c668a2feca47ce135399658
doha@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE                     COMMAND                  CREATED
   STATUS         PORTS                                          NAMES
6cc804cc575d   mysql                     "docker-entrypoint.s…"   7 seconds ago
   Up 5 seconds   3306/tcp, 33060/tcp                            mysql_db_2
41c2d3ea2b13   kodekloud/simple-webapp   "python app.py"          3 hours ago
   Up 3 hours     0.0.0.0:38282->8080/tcp, :::38282->8080/tcp   blue-app
doha@ubuntu:~$ sudo docker exec -it 6cc804cc575d /bin/bash
bash-4.4# ls
bin   docker-entrypoint-initdb.d  home   media  proc  sbin  tmp
boot  entrypoint.sh               lib    mnt    root  srv   usr
dev   etc                         lib64  opt    run   sys   var
bash-4.4# cd /var/lib/mysql/
bash-4.4# ls
From_container
bash-4.4#
```