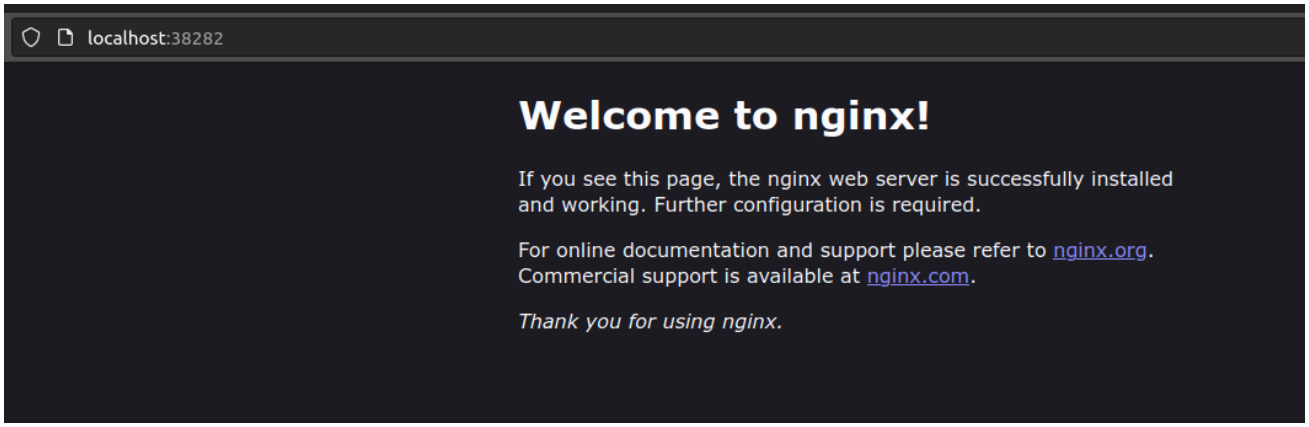
1- Run an instance of nginx:alpine with a name nginx and map port 8080 on the container to 38282 on the host.

```
heba@heba-HP-ProBook-450-G4:~$ docker run --name nginx -p 38282:8080 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
2023/05/28 04:31:25 [notice] 1#1: using the "epoll" event method
2023/05/28 04:31:25 [notice] 1#1: nginx/1.23.4
2023/05/28 04:31:25 [notice] 1#1: built by gcc 12.2.1 20220924 (Alpine 12.2.1_git20220924-r4)
2023/05/28 04:31:25 [notice] 1#1: OS: Linux 5.19.0-41-generic
2023/05/28 04:31:25 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/05/28 04:31:25 [notice] 1#1: start worker processes
2023/05/28 04:31:25 [notice] 1#1: start worker process 30
2023/05/28 04:31:25 [notice] 1#1: start worker process 31
2023/05/28 04:31:25 [notice] 1#1: start worker process 32
```

```
/etc/nginx/conf.d # vi default.conf
/etc/nginx/conf.d # vi default.conf
/etc/nginx/conf.d # nginx -s reload
2023/05/28 04:34:28 [notice] 46#46: signal process started
/etc/nginx/conf.d #
```

node@heba-HP-ProBook-450-G4: /home/heba

```
server {
    listen       8080;
    listen  [::]:8080;
    server_name  localhost;

    #access_log  /var/log/nginx/host.access.log  main;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
```

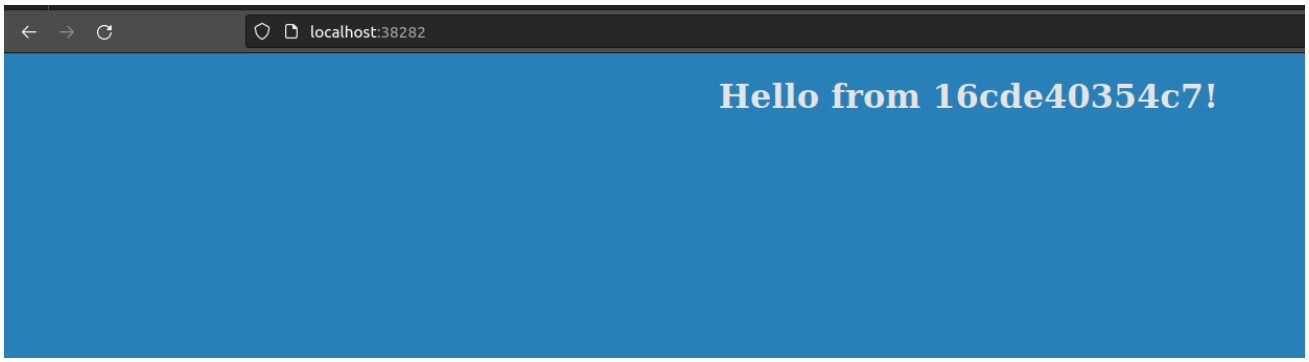


2 - create ubuntu image and check the size of it

```
heba@heba-HP-ProBook-450-G4:~$ docker run --name my-ubuntu -d ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
dbf6a9befcde: Pull complete
Digest: sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee20739e8eb54fbf
Status: Downloaded newer image for ubuntu:latest
ec58bc63594811312cfedd42590cd12f3600b3e3e402fed61f8f0dd8ac2624e6
heba@heba-HP-ProBook-450-G4:~$ docker images ubuntu
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
ubuntu       latest    3b418d7b466a   4 weeks ago   77.8MB
heba@heba-HP-ProBook-450-G4:~$
```

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.

```
heba@heba-HP-ProBook-450-G4:~$ docker run -d --name blue-app -e APP_COLOR=blue -p 38282:8080 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: 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
16cde40354c76633d58521ea9355fd83b179441ddbecb024c5736f14096324da
```



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

```
heba@heba-HP-ProBook-450-G4:~$ 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
d3e79391f7ba48993dd7f56df3c05f6355bbad05a6d45b304fd054abd3342a26
```

```
heba@heba-HP-ProBook-450-G4:~$ docker inspect -f '{{.Config.Env}}' mysql-db
[MYSQL_ROOT_PASSWORD=db_pass123 PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin GOSU_VERSION=1.16 MYSQL_MAJOR=8.0 MYSQL_VE
heba@heba-HP-ProBook-450-G4:~$ docker inspect  mysql-db
[
    {
        "Id": "d3e79391f7ba48993dd7f56df3c05f6355bbad05a6d45b304fd054abd3342a26",
        "Created": "2023-05-28T04:54:09.004386149Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "mysqld"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 58732,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2023-05-28T04:54:14.293766706Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        }
```

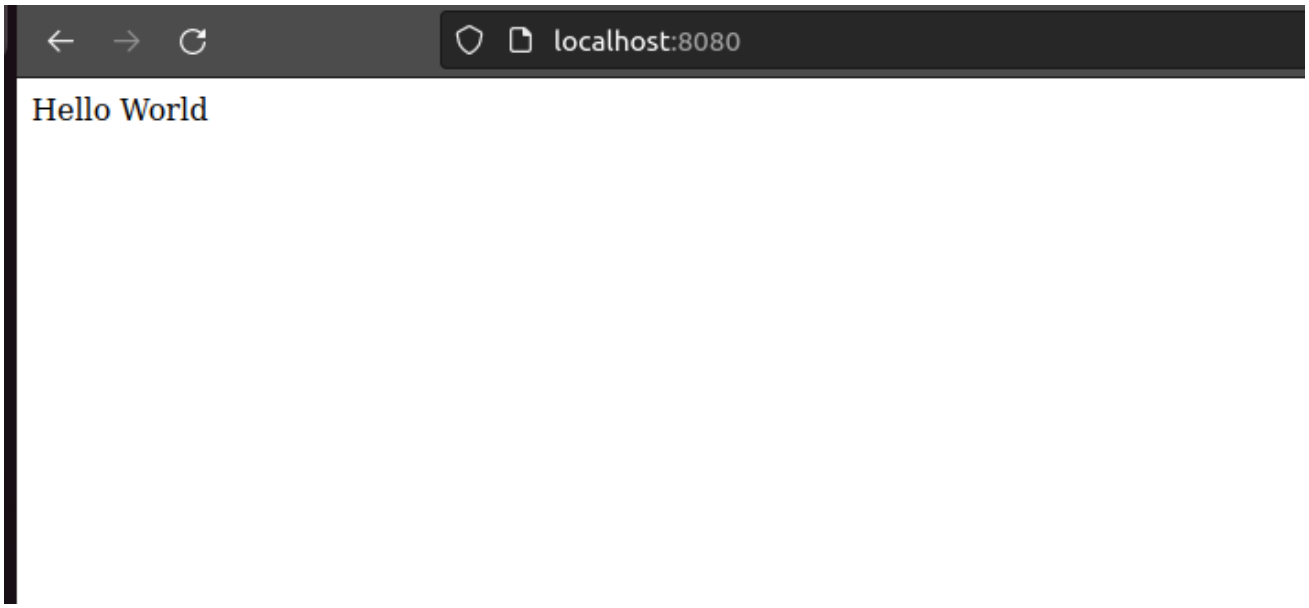
5- pull the code from https://github.com/sabreensalama/dockerize-node-app-task and create a docker file for this flask app

```
]
heba@heba-HP-ProBook-450-G4:~$ git clone https://github.com/sabreensalama/dockerize-node-app-task.git
cd dockerize-node-app-task
Cloning into 'dockerize-node-app-task'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (12/12), done.
Resolving deltas: 100% (1/1), done.
heba@heba-HP-ProBook-450-G4:~/dockerize-node-app-task$ touch Dockerfile
heba@heba-HP-ProBook-450-G4:~/dockerize-node-app-task$ vim Dockerfile
```

```
# Use an official Node.js runtime as a parent image
FROM node:14-alpine

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Install any needed packages specified in package.json
RUN npm install

# Make port 8080 available to the world outside this container
EXPOSE 8080

# Run the app when the container launches
CMD ["npm", "start"]
~
```

```
heba@heba-HP-ProBook-450-G4:~/dockerize-node-app-task$ docker build -t node-app .
[+] Building 62.7s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 452B
 => [internal] load .dockerignore
 => => transferring context: 2B
```

```
 => exporting to image
 => => exporting layers
 => => writing image sha256:1cb0e55f4455252185e116114120d9cef2b459b7bd401b50dbcf663e270bcf7d
 => => naming to docker.io/library/node-app
heba@heba-HP-ProBook-450-G4:~/dockerize-node-app-task$ docker run -d -p 8080:8080 node-app
617e0521336f37b680bd4ba75cfe2a101943251449b297f97ce5d9bec25988b4
```



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 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.

```
heba@heba-HP-ProBook-450-G4:~$ docker run -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 -v mysql_data:/var/lib/mysql -v /opt/data:/opt/data mysql
f39eff0a1e19d06ec41a9fdb2ed870ba2925442a172b63143d3ce9d422b0dde1
```


```
    "Mounts": [
        {
            "Type": "volume",
            "Name": "mysql_data",
            "Source": "/var/lib/docker/volumes/mysql_data/_data",
            "Destination": "/var/lib/mysql",
            "Driver": "local",
            "Mode": "z",
            "RW": true,
            "Propagation": ""
        },
        {
            "Type": "bind",
            "Source": "/opt/data",
            "Destination": "/opt/data",
            "Mode": "",
            "RW": true,
            "Propagation": "rprivate"
        }
    ],
```