

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

```
abdefattah@abdefattah-Ashraf:~$ sudo docker run --name nginx -p 38282:8080 nginx:alpine
[sudo] password for abdefattah:
/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
2023/05/27 16:58:48 [notice] 1#1: using the "epoll" event method
2023/05/27 16:58:48 [notice] 1#1: nginx/1.25.0
2023/05/27 16:58:48 [notice] 1#1: built by gcc 12.2.1 20220924 (Alpine 12.2.1_git20220924-r4)
2023/05/27 16:58:48 [notice] 1#1: OS: Linux 5.19.0-41-generic
2023/05/27 16:58:48 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/05/27 16:58:48 [notice] 1#1: start worker processes
2023/05/27 16:58:48 [notice] 1#1: start worker process 30
2023/05/27 16:58:48 [notice] 1#1: start worker process 31
2023/05/27 16:58:48 [notice] 1#1: start worker process 32
2023/05/27 16:58:48 [notice] 1#1: start worker process 33
2023/05/27 16:58:48 [notice] 1#1: start worker process 34
```

2- create ubuntu image and check the size of it

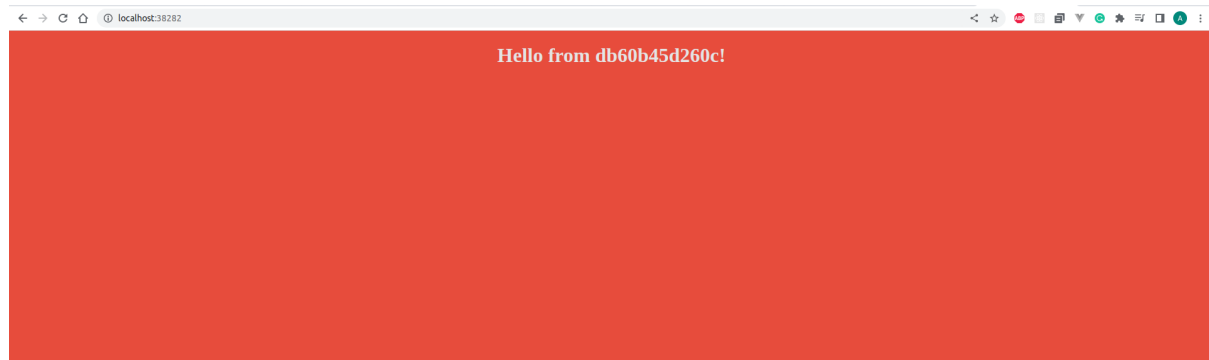
```
abdefattah@abdefattah-Ashraf:~/Desktop/Apps/docker/Dock-Task$ sudo touch Dockerfile
[sudo] password for abdefattah:
abdefattah@abdefattah-Ashraf:~/Desktop/Apps/docker/Dock-Task$ vim Dockerfile
abdefattah@abdefattah-Ashraf:~/Desktop/Apps/docker/Dock-Task$ sudo docker build -t ubuntu-image .
[+] Building 12.6s (5/5) FINISHED
=> [internal] load build definition from Dockerfile                                0.2s
=> => transferring dockerfile: 56B                                                0.0s
=> [internal] load .dockerignore                                                  0.1s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest                 2.5s
=> [1/1] FROM docker.io/library/ubuntu:latest@sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee207  9.7s
=> => resolve docker.io/library/ubuntu:latest@sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee207  0.0s
=> => sha256:3b418d7b466ac6275a6bfc0c86fbc4422ff6ea0af444a294f82d3bf5173ce74 2.30kB / 2.30kB      0.0s
=> => sha256:dbf6a9befcdeecbb8813406afb62ce81394e3869d84599f19f941aa5c74f3d1 29.53MB / 29.53MB    8.5s
=> => sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee20739e8eb54fbf 1.13kB / 1.13kB      0.0s
=> => sha256:ca5534a51dd04bbcebe9b23ba05f389466cf0c190f1f8f182d7eea92a9671d00 424B / 424B      0.0s
=> => extracting sha256:dbf6a9befcdeecbb8813406afb62ce81394e3869d84599f19f941aa5c74f3d1          0.8s
=> => exporting to image                                                         0.0s
=> => exporting layers                                                            0.0s
=> => writing image sha256:bb59a1c2f7b943b56c8b224b0d0039412d8594a5eebf58f58ff8512874d4db8e          0.0s
=> => naming to docker.io/library/ubuntu-image                                0.0s
abdefattah@abdefattah-Ashraf:~/Desktop/Apps/docker/Dock-Task$ sudo docker images ubuntu-image
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu-image  latest   bb59a1c2f7b9  4 weeks ago   77.8MB
abdefattah@abdefattah-Ashraf:~/Desktop/Apps/docker/Dock-Task$
```

3- Run a container named blue-app using image kodekloud/simplewebapp 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.

```

abdefattah@abdefattah-Ashraf:~$ sudo docker run -d -p 38282:8080 -e ALL_COLOR=blue 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:e5355b4c7804f453d79de75d6659ee702eebbe30c02d9f1ce6602a96b576e57
Status: Downloaded newer image for kodekloud/simple-webapp:latest
db60b45d260c49bffa31c2a75b047035f76e8a797c11b8774a59e943a265213dd
abdefattah@abdefattah-Ashraf:~$

```



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

```

abdefattah@abdefattah-Ashraf:~$ sudo docker run --name mysqlldb -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
ed742b97f9dca3c8b8f75c6db7b835709caf2f6111659dd2bc92b13572ee2acf

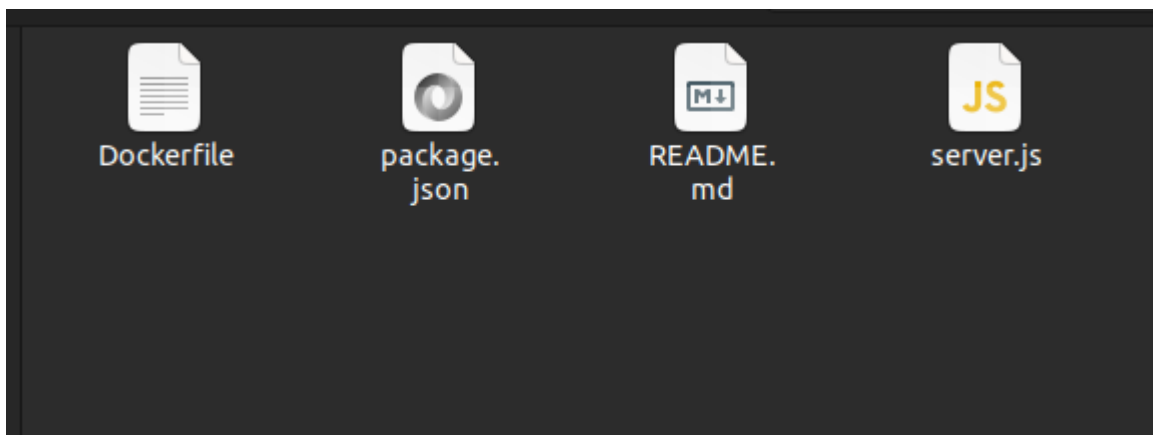
```

5- pull the code from <https://github.com/sabreensalama/dockerizenode-app-task> and create a docker file for this app

```

abdelfattah@abdelfattah-Ashraf: ~/Desktop/Apps/docker/dockerize-node-app-task-main$ sudo docker build -t node-app .
[sudo] password for abdelFattah:
[+] Building 24.5s (11/11) FINISHED
=> [internal] load build definition from Dockerfile                                0.2s
=> => transferring dockerfile: 428B                                              0.0s
=> [internal] load .dockerignore                                                 0.2s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/node:14-alpine                3.2s
=> [auth] library/node:pull token for registry-1.docker.io                     0.0s
=> [1/5] FROM docker.io/library/node:14-alpine@sha256:434215b487a329c9e8672b2ff89e704d3a75e554822e07f3e0c0f9e060121b33 15.3s
=> => resolve docker.io/library/node:14-alpine@sha256:434215b487a329c9e8672b2ff89e704d3a75e554822e07f3e0c0f9e060121b33 0.1s
=> => sha256:434215b487a329c9e8672b2ff89e704d3a75e554822e07f3e0c0f9e060121b33 1.43kB / 1.43kB 0.0s
=> => sha256:4e04c355cd270eaf9ed14a8b25939734364c2b0942405e09e1b97175e73df0548 1.16kB / 1.16kB 0.0s
=> => sha256:80ac3dc27b1ad570e0c3a7f7cd29e88e7130ff0cad31b2ec5a0f22f2f5e971b0b 6.44kB / 6.44kB 0.0s
=> => sha256:e5fca0c395a02ec277102af9e5203f0ed043b3e4f20f7798e3ce7e425be220ba6 2.37MB / 2.37MB 1.2s
=> => sha256:561cb09653d56a9725be56e02128e4e96fb434a8b4b4decf2bdeb479a225feaf 448B / 448B 0.3s
=> => sha256:8f065085b215c7daf9164545f1bbdd74d800af77d9d267db31fe0345c0c8fb8b 37.17MB / 37.17MB 13.3s
=> => extracting sha256:8f065085b215c7daf9164545f1bbdd74d800af77d9d267db31fe0345c0c8fb8b 1.3s
=> => extracting sha256:45fca0c395a02ec277102af9e5203f0ed043b3e4f20f7798e3ce7e425be220ba6 0.0s
=> => extracting sha256:561cb09653d56a9725be56e02128e4e96fb434a8b4b4decf2bdeb479a225feaf 0.0s
=> [internal] load build context                                                0.1s
=> => transferring context: 1.22kB                                              0.0s
=> [2/5] WORKDIR /app                                                         0.5s
=> [3/5] COPY package*.json ./                                                0.1s
=> [4/5] RUN npm install                                                       4.5s
=> [5/5] COPY -                                                                0.2s
=> exporting to image                                                         0.3s
=> => exporting layers                                                         0.3s
=> => writing image sha256:db6722f8170730eb79f1fa8fd650ec0bd9364061d93420ee344dba31a92763 0.0s
=> => naming to docker.io/library/node-app                                     0.0s
abdelfattah@abdelfattah-Ashraf: ~/Desktop/Apps/docker/dockerize-node-app-task-main$

```



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.

```

abdelfattah@abdelfattah-Ashraf:~$ sudo docker volume create mysql_data
mysql_data
abdelfattah@abdelfattah-Ashraf:~$ sudo docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 -v mysql_data:/var/lib/mysql -d mysql
0022d49d1b382fec54a0615464ef72960c7e630b51e26cb1772132f93205573e
abdelfattah@abdelfattah-Ashraf:~$ sudo docker ps

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0022d49d1b38	mysql	"docker-entrypoint.s..."	30 seconds ago	Up 29 seconds	3306/tcp, 33060/tcp	mysql-db