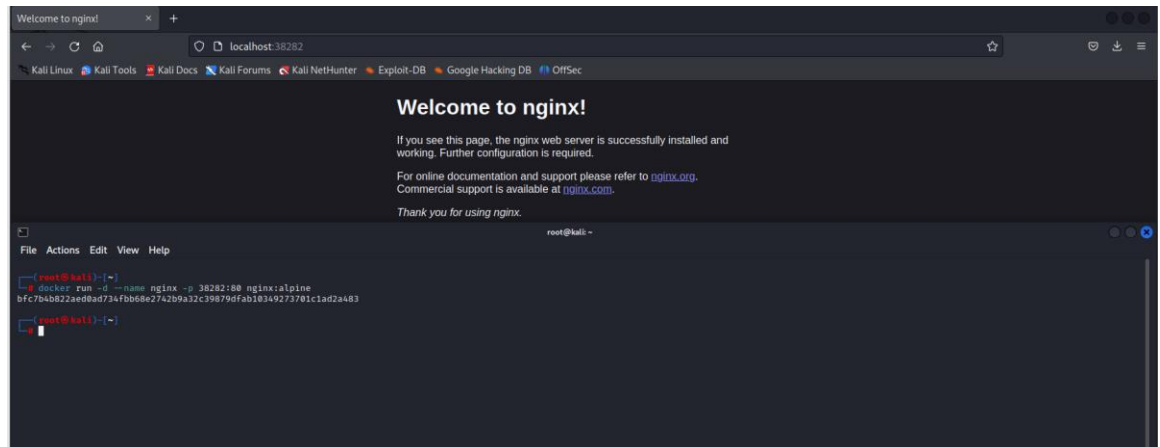
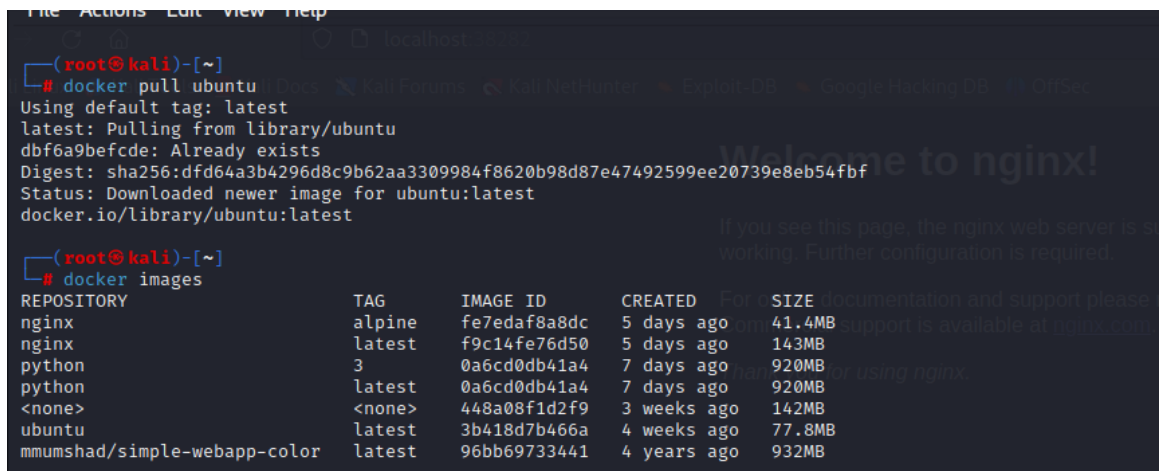


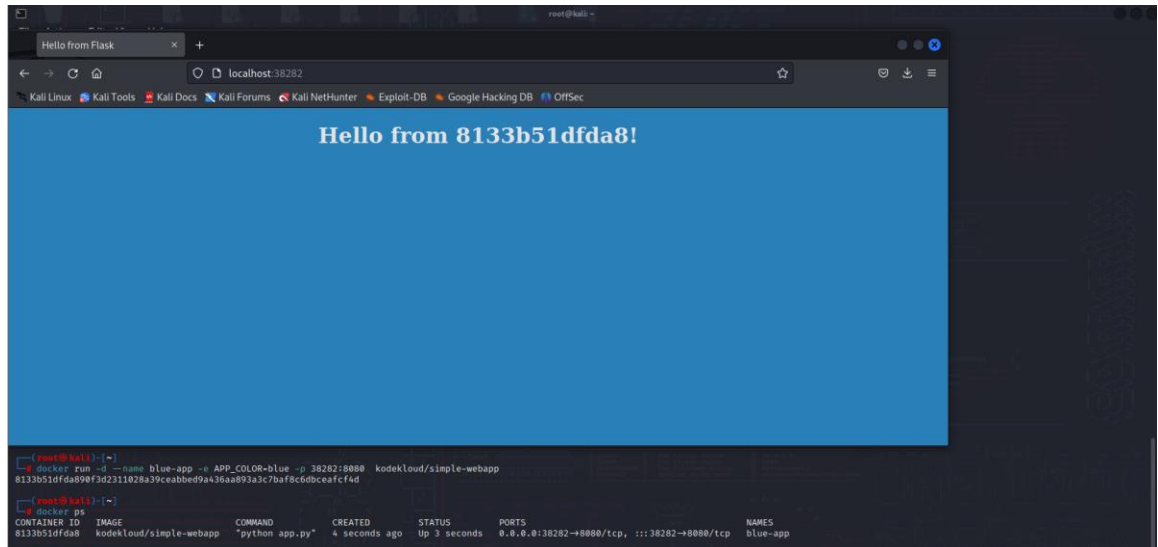
- 1- Run an instance of nginx:alpine with a name nginx and map port 8080 on the container to 38282 on the host.



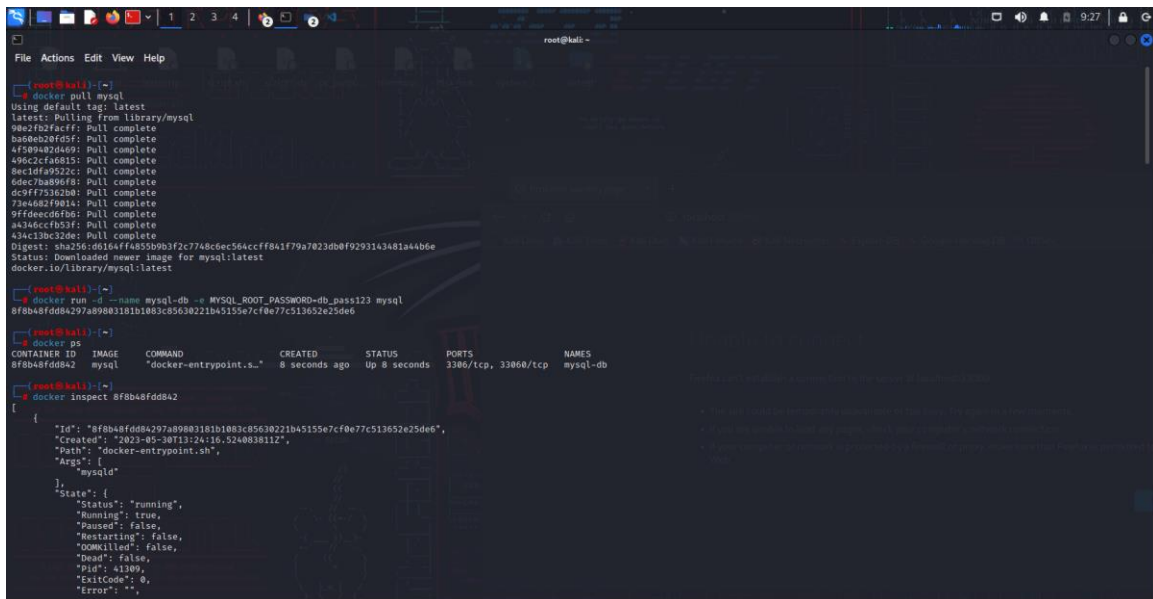
- 2- create ubuntu image and check the size of it



- 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.



- 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



```
"Env": [
  "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_VERSION=8.0.33-1.el8",
  "MYSQL_SHELL_VERSION=8.0.33-1.el8"
],
```

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

```
(root@kali) ~/home/kali/Desktop
$ git clone https://github.com/sabreensalama/dockerize-node-app-task.git
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 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (12/12), done.
Resolving deltas: 100% (1/1), done.

Cloning with GitHub CLI

(root@kali) ~/home/kali/Desktop
$ ls
disk_output.txt  dockerize-node-app-task  iletest  memory_output.txt  mysql-application  ps_output.txt  script1.sh  script.sh  system_info.tar.gz  testscript.sh  xfce4-terminal-emulator.desktop

(root@kali) ~/home/kali/Desktop
$ cd dockerize-node-app-task
$ ls
package.json  README.md  server.js

Cloning with Subversion

(root@kali) ~/home/kali/Desktop/dockerize-node-app-task
$ touch Dockerfile

File Actions Edit View Help

(root@kali) ~/home/kali/Desktop/dockerize-node-app-task
$ cat requirements.txt
express

(root@kali) ~/home/kali/Desktop/dockerize-node-app-task
$ cat Dockerfile
From node:14

WORKDIR / app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["node", "app.js"]

(root@kali) ~/home/kali/Desktop/dockerize-node-app-task
$
```

```
(root@kali)-[/home/kali/Desktop/dockerize-node-app-task]
# docker build -t my-nodeapp .
Sending build context to Docker daemon 77.82kB
Step 1/7 : From node:14
14: Pulling from library/node
2ff1d7c41c74: Pull complete
b253aaefaa7: Pull complete
3d2201bd995c: Pull complete
1de76e268b10: Pull complete
d9a8df589451: Pull complete
6f51ee005dea: Pull complete
5f32ed3c3f27: Pull complete
0c8cc2f24a4d: Pull complete
0d27a8e86132: Pull complete
Digest: sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa
Status: Downloaded newer image for node:14
Step 2/7 : WORKDIR / app
Removing intermediate container e8b0b59fb200
Step 3/7 : COPY package*.json ./
Step 4/7 : RUN npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN docker_web_app@1.0.0 No repository field.
npm WARN docker_web_app@1.0.0 No license field.
added 58 packages from 42 contributors and audited 58 packages in 9.007s
8 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
Removing intermediate container ea7dacd8b692
Step 5/7 : COPY . .
Step 6/7 : EXPOSE 3000
Removing intermediate container 2123b9e239d3
Step 7/7 : CMD ["node", "app.js"]
Removing intermediate container 6b483c32a54c
Successfully built ff2a24d13cbc

# docker run -d -p 3000:3000 my-nodeapp
6f84bc97810db11ea218944a36d0066891d0e442a9d11b33459d79336471fcc2
```

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.

```
(root@kali)-[~]  
# docker run -d --name mysql-db -v Mysql_data:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=db_pass123 mysql  
7d4193ed5f04d162894b86ae3ed981dba6ba1d94109f60e2e24541c943dc9945
```