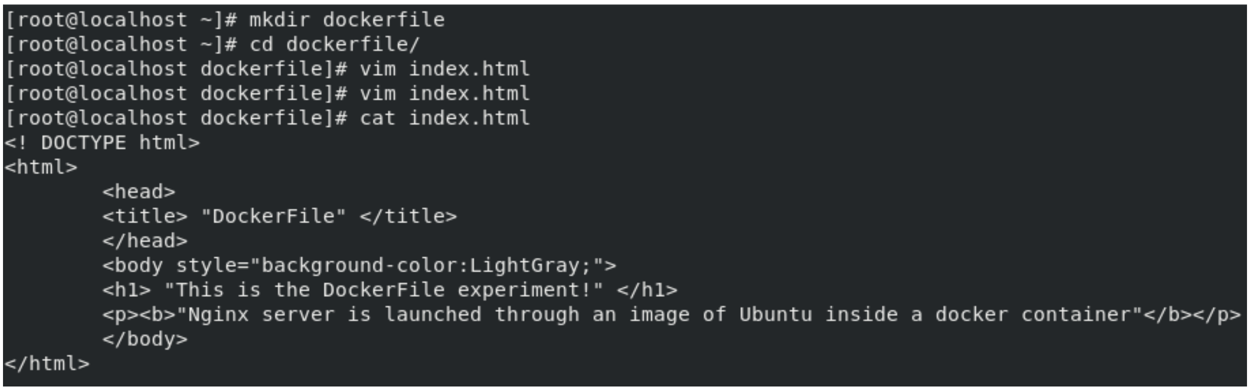
Experiment-4

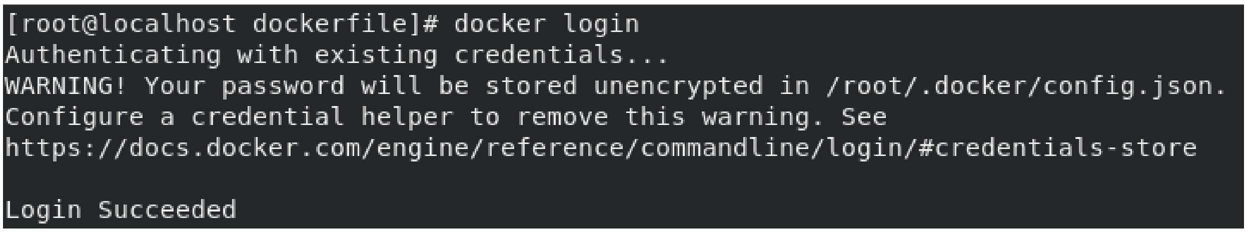
Aim- To create Docker image using Dockerfile and pushing to Docker Hub

- Creating a directory using mkdir command, here “mkdir dockerfile” and changing to it using cd command. Since our goal is to create an image containing webserver, we have to copy a html file with some contents, here “index.html”.



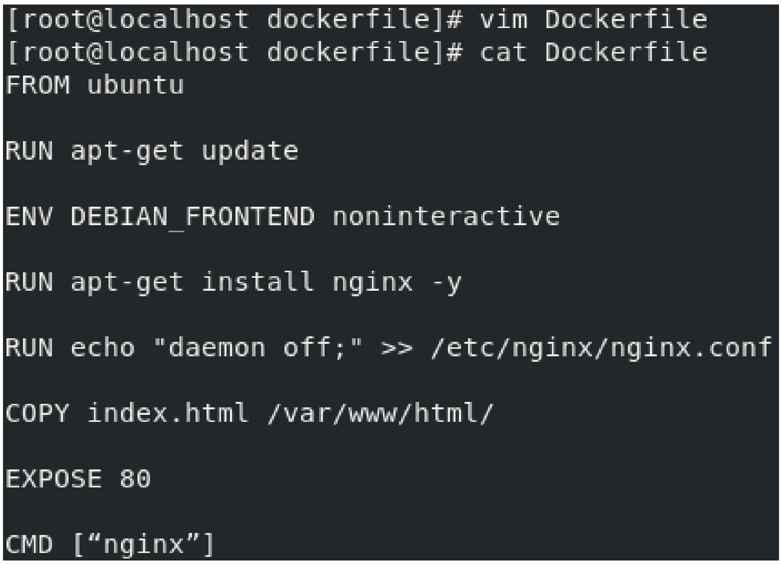
```
[root@localhost ~]# mkdir dockerfile
[root@localhost ~]# cd dockerfile/
[root@localhost dockerfile]# vim index.html
[root@localhost dockerfile]# vim index.html
[root@localhost dockerfile]# cat index.html
<! DOCTYPE html>
<html>
        <head>
        <title> "DockerFile" </title>
        </head>
        <body style="background-color:LightGray;">
        <h1> "This is the DockerFile experiment!" </h1>
        <p><b>"Nginx server is launched through an image of Ubuntu inside a docker container"</b></p>
        </body>
</html>
```

- Logging in to the DockerHub account using the command “docker login”.



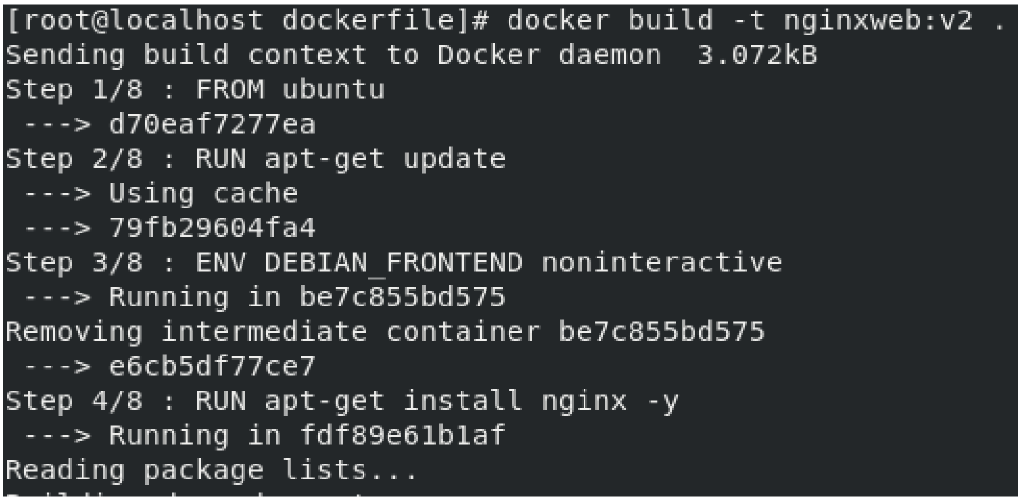
```
[root@localhost dockerfile]# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

- Creating a docker file and the image which we are creating is on top of ubuntu OS, so we use “FROM” keyword to fetch Ubuntu image. After that, we update the yum repository of the container using “RUN” keyword. Setting an environment variable using “ENV” keyword, setting DEBIAN_FRONTEND as noninteractive. Installing nginx webserver using “RUN” keyword and assigning “daemon off;” in the nginx config file using “RUN” keyword. Copying the index.html file using “COPY” keyword and exposing the 80-port using “EXPOSE” keyword. And after all that for starting services we use the “CMD” keyword.

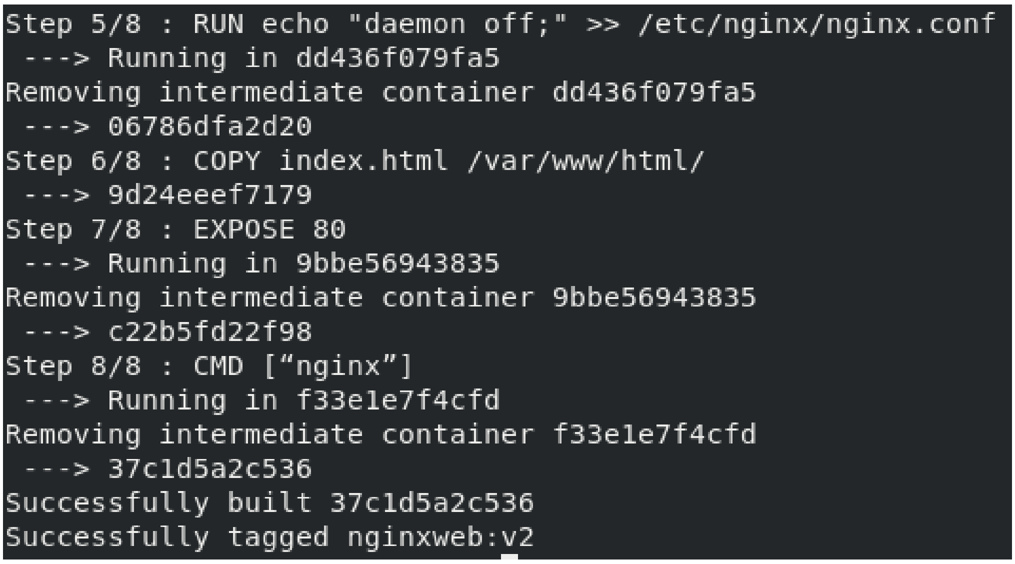


```
[root@localhost dockerfile]# vim Dockerfile
[root@localhost dockerfile]# cat Dockerfile
FROM ubuntu

RUN apt-get update

ENV DEBIAN_FRONTEND noninteractive

RUN apt-get install nginx -y

RUN echo "daemon off;" >> /etc/nginx/nginx.conf

COPY index.html /var/www/html/

EXPOSE 80

CMD [“nginx”]
```

- Creating the image from Dockerfile using the command docker build.

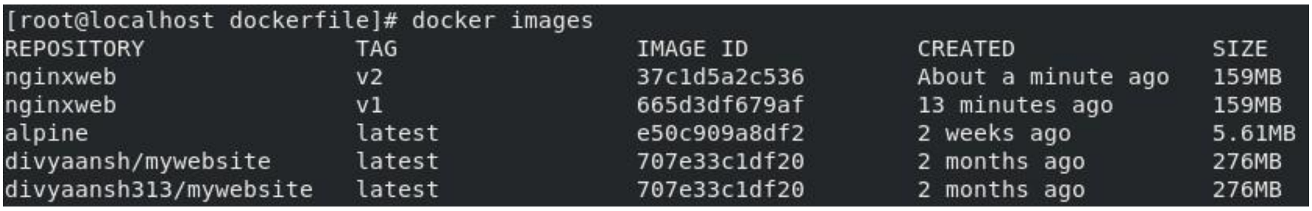


```
[root@localhost dockerfile]# docker build -t nginxweb:v2 .
Sending build context to Docker daemon  3.072kB
Step 1/8 : FROM ubuntu
 ---> d70eaf7277ea
Step 2/8 : RUN apt-get update
 ---> Using cache
 ---> 79fb29604fa4
Step 3/8 : ENV DEBIAN_FRONTEND noninteractive
 ---> Running in be7c855bd575
Removing intermediate container be7c855bd575
 ---> e6cb5df77ce7
Step 4/8 : RUN apt-get install nginx -y
 ---> Running in fdf89e61b1af
Reading package lists...
```



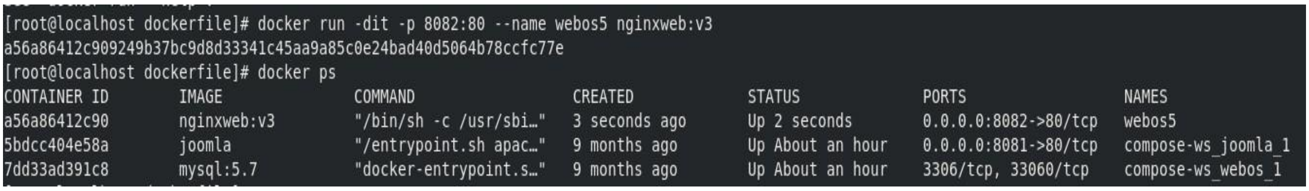
```
Step 5/8 : RUN echo "daemon off;" >> /etc/nginx/nginx.conf
 ---> Running in dd436f079fa5
Removing intermediate container dd436f079fa5
 ---> 06786dfa2d20
Step 6/8 : COPY index.html /var/www/html/
 ---> 9d24eeef7179
Step 7/8 : EXPOSE 80
 ---> Running in 9bbe56943835
Removing intermediate container 9bbe56943835
 ---> c22b5fd22f98
Step 8/8 : CMD [“nginx”]
 ---> Running in f33e1e7f4cfd
Removing intermediate container f33e1e7f4cfd
 ---> 37c1d5a2c536
Successfully built 37c1d5a2c536
Successfully tagged nginxweb:v2
```

- After it is built successfully, we can see the image using docker images.



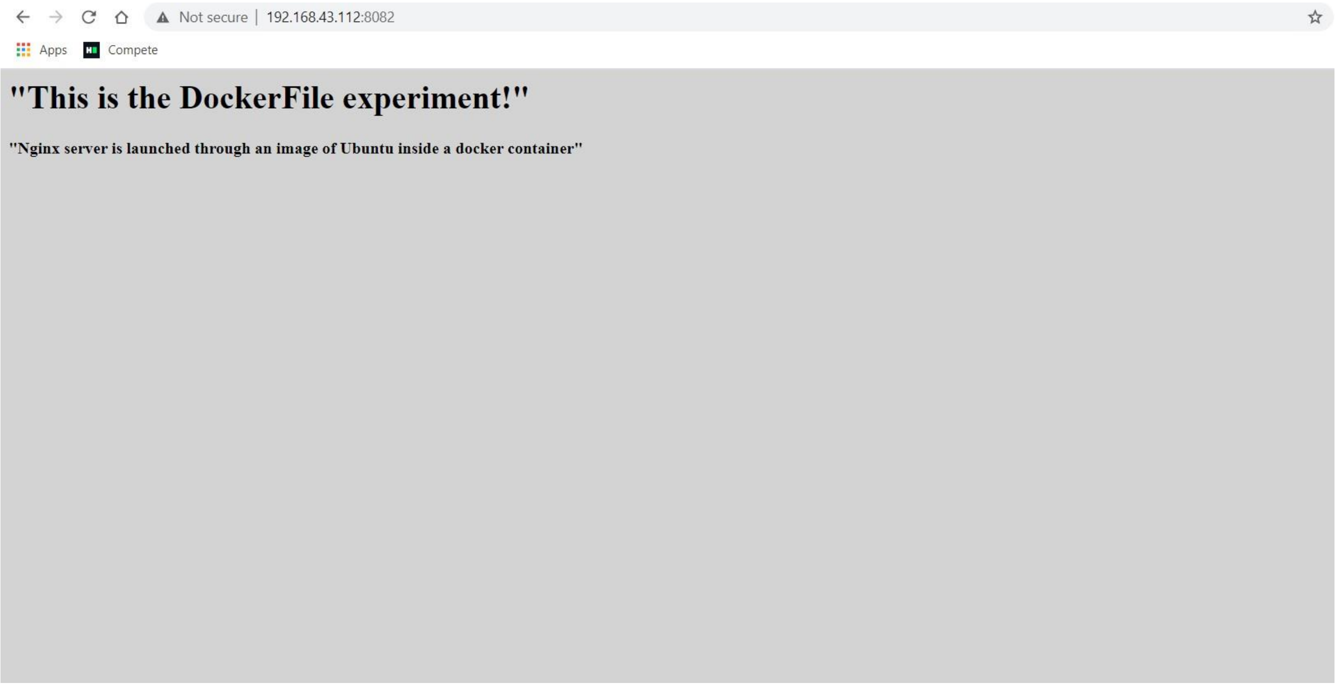
```
[root@localhost dockerfile]# docker images
REPOSITORY               TAG       IMAGE ID       CREATED              SIZE
nginxweb                 v2        37c1d5a2c536   About a minute ago   159MB
nginxweb                 v1        665d3df679af   13 minutes ago       159MB
alpine                   latest    e50c909a8df2   2 weeks ago          5.61MB
divyaansh/mywebsite      latest    707e33c1df20   2 months ago         276MB
divyaansh313/mywebsite   latest    707e33c1df20   2 months ago         276MB
```

- Running a container using this image with -p option to attach a port to the container for accessing the webserver.



```
[root@localhost dockerfile]# docker run -dit -p 8082:80 --name webos5 nginxweb:v3
a56a86412c909249b37bc9d8d33341c45aa9a85c0e24bad40d5064b78ccfc77e
[root@localhost dockerfile]# docker ps
CONTAINER ID   IMAGE         COMMAND                  CREATED         STATUS            PORTS                  NAMES
a56a86412c90   nginxweb:v3   "/bin/sh -c /usr/sbi…"   3 seconds ago   Up 2 seconds      0.0.0.0:8082->80/tcp   webos5
5bdcc404e58a   joomla        "/entrypoint.sh apac…"   9 months ago    Up About an hour  0.0.0.0:8081->80/tcp   compose-ws_joomla_1
7dd33ad391c8   mysql:5.7     "docker-entrypoint.s…"   9 months ago    Up About an hour  3306/tcp, 33060/tcp    compose-ws_webos_1
```

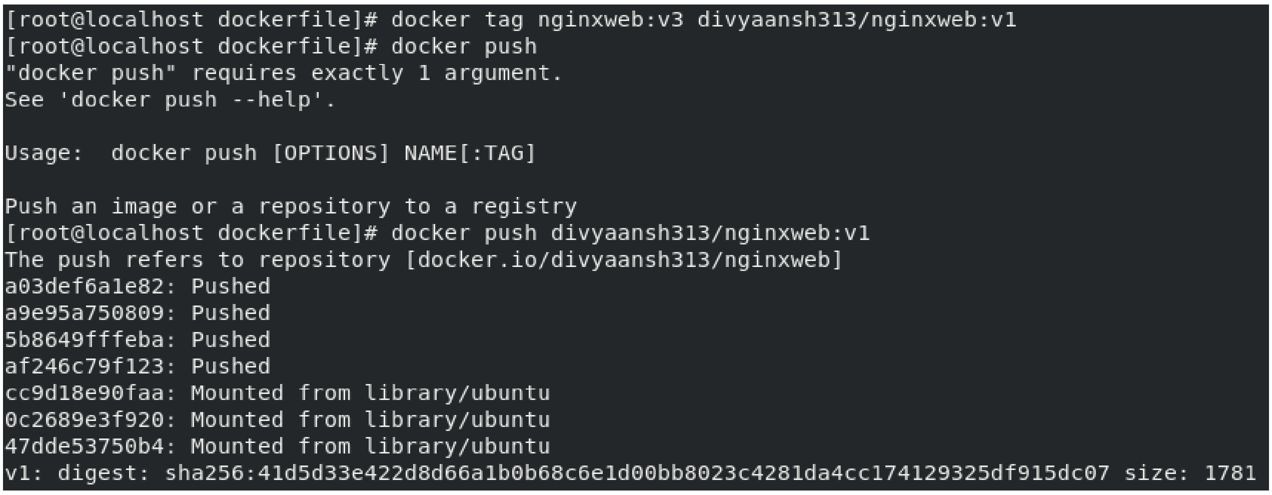
- Checking if the webserver is working fine from the browser.



"This is the DockerFile experiment!"

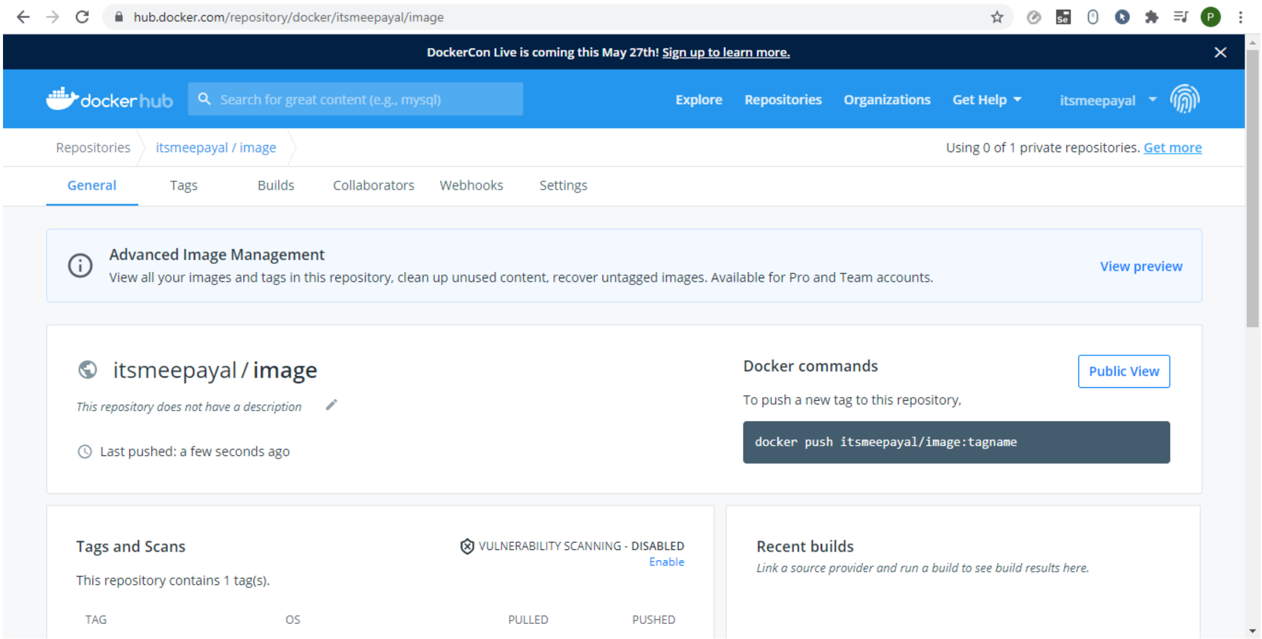
"Nginx server is launched through an image of Ubuntu inside a docker container"

- Tagging the file appropriately using the command "docker tag" and pushing it to the Dockerhub repository using the command "docker push".



```
[root@localhost dockerfile]# docker tag nginxweb:v3 divyaansh313/nginxweb:v1
[root@localhost dockerfile]# docker push
"docker push" requires exactly 1 argument.
See 'docker push --help'.

Usage:  docker push [OPTIONS] NAME[:TAG]

Push an image or a repository to a registry
[root@localhost dockerfile]# docker push divyaansh313/nginxweb:v1
The push refers to repository [docker.io/divyaansh313/nginxweb]
a03def6a1e82: Pushed
a9e95a750809: Pushed
5b8649fffeba: Pushed
af246c79f123: Pushed
cc9d18e90faa: Mounted from library/ubuntu
0c2689e3f920: Mounted from library/ubuntu
47dde53750b4: Mounted from library/ubuntu
v1: digest: sha256:41d5d33e422d8d66a1b0b68c6e1d00bb8023c4281da4cc174129325df915dc07 size: 1781
```

- Checking the image in the DockerHub.



DockerCon Live is coming this May 27th! Sign up to learn more.

Using 0 of 1 private repositories. Get more

Advanced Image Management

View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts.

itsmeepayal / image

This repository does not have a description

Last pushed: a few seconds ago

Docker commands

To push a new tag to this repository,

```
docker push itsmeepayal/image:tagname
```

Tags and Scans

This repository contains 1 tag(s).

Recent builds

Link a source provider and run a build to see build results here.