Step I : create the EC2 Instance using http and ssh in security group.

Step II : then Install docker using “yum install docker -y”.

Step III : then Start Docker using “systemctl start docker”.

Step IV : then copy the link address for download css template and using curl -O <link_address> download it.

Step V : unzip it .

```
[root@ip-172-31-42-97 ec2-user]# vim Dockerfile
[root@ip-172-31-42-97 ec2-user]# curl -O https://www.free-css.com/assets/files/free-css-templates/download/p
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1316k  100 1316k    0     0  1029k      0  0:00:01  0:00:01 --:--:-- 1029k
[root@ip-172-31-42-97 ec2-user]# ls
browny.zip  index.html  krishna
[root@ip-172-31-42-97 ec2-user]# unzip browny.zip
Archive:  browny.zip
```

Step VI : create Dockerfile using “vim Dockerfile”.

```
# Use the official httpd (Apache HTTP Server) image as the base
FROM httpd:latest

# Copy the contents of the "browny-v1.0" directory from the local machine
# into the "htdocs" directory of the httpd container
COPY ./browny-v1.0 /usr/local/apache2/htdocs/browny-v1.0

# Expose port 80 to allow external access to the httpd server
EXPOSE 80
```

Step VII : Add above content.

Step VIII : then using “docker build -t <image_name> .” this command and build image from Dockerfile.

```
[root@ip-172-31-42-97 ec2-user]# docker build -t my-httpd-image .
Sending build context to Docker daemon  4.099MB
```

To see image use “docker images”.

```
[root@ip-172-31-42-97 ec2-user]# docker images
REPOSITORY       TAG       IMAGE ID       CREATED          SIZE
my-httpd-image   latest    6089bff39260   26 seconds ago   151MB
b31              latest    a6f03f82b6b7   3 days ago       187MB
redis            latest    7fc37b47acde   5 days ago       116MB
httpd            latest    fa0099f1c09d   5 days ago       148MB
postgres         latest    d60dc4bd84c0   7 weeks ago      431MB
postgres         9.6       027ccf656dc1   2 years ago      200MB
postgres         10.10     9a05a2b9e69f   4 years ago      211MB
```

Step IX : then run image using “docker -d -p 80:80 <image_name>

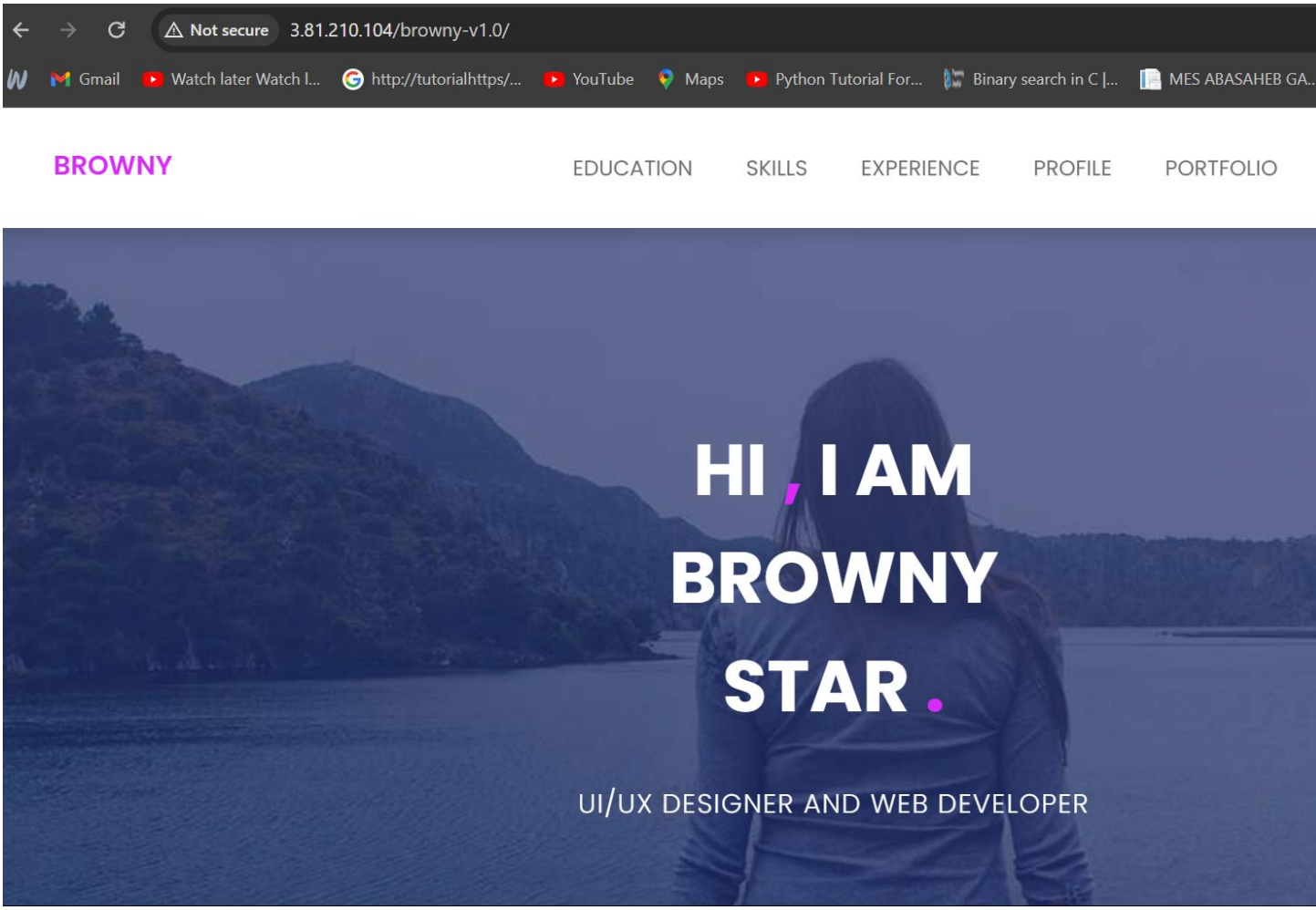
```
[root@ip-172-31-42-97 ec2-user]# docker run -d -p 80:80 --name my-httpd-container my-httpd-image
cca5fefb08e57f5b218d1fec96e1a6791786aa96084d90aaef6b498d1ce01582
```

To see Container use “docker ps”.

```
[root@ip-172-31-42-97 ec2-user]# docker ps
CONTAINER ID   IMAGE            COMMAND                  CREATED          STATUS          PORTS
cca5fefb08e5   my-httpd-image   "httpd-foreground"       6 seconds ago    Up 6 seconds    0.0.0.0:80->80/tcp,
5753e300fd10   redis            "docker-entrypoint.s…"   2 hours ago      Up 2 hours      6379/tcp
92d178cf59f3   postgres:10.10   "docker-entrypoint.s…"   2 hours ago      Up 2 hours      5432/tcp
3efe53a21106   postgres         "docker-entrypoint.s…"   2 hours ago      Up 2 hours      5432/tcp
```

Step X: copy the instance public ip and paste it using above format.

Step XI : here, template Hosted Successfully.




Step XI : here, template Hosted Successfully.