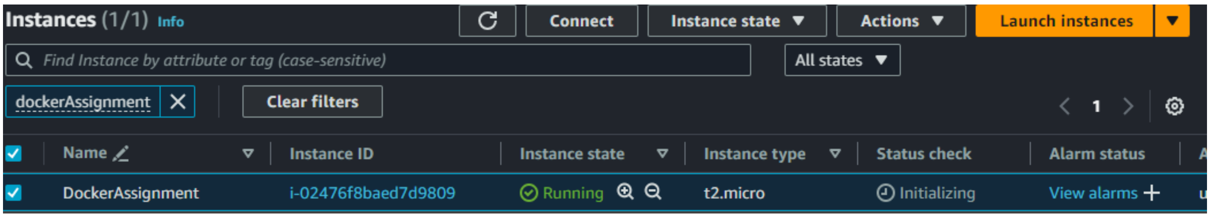
Module-3: Docker – I Assignment - 1

You have been asked to:

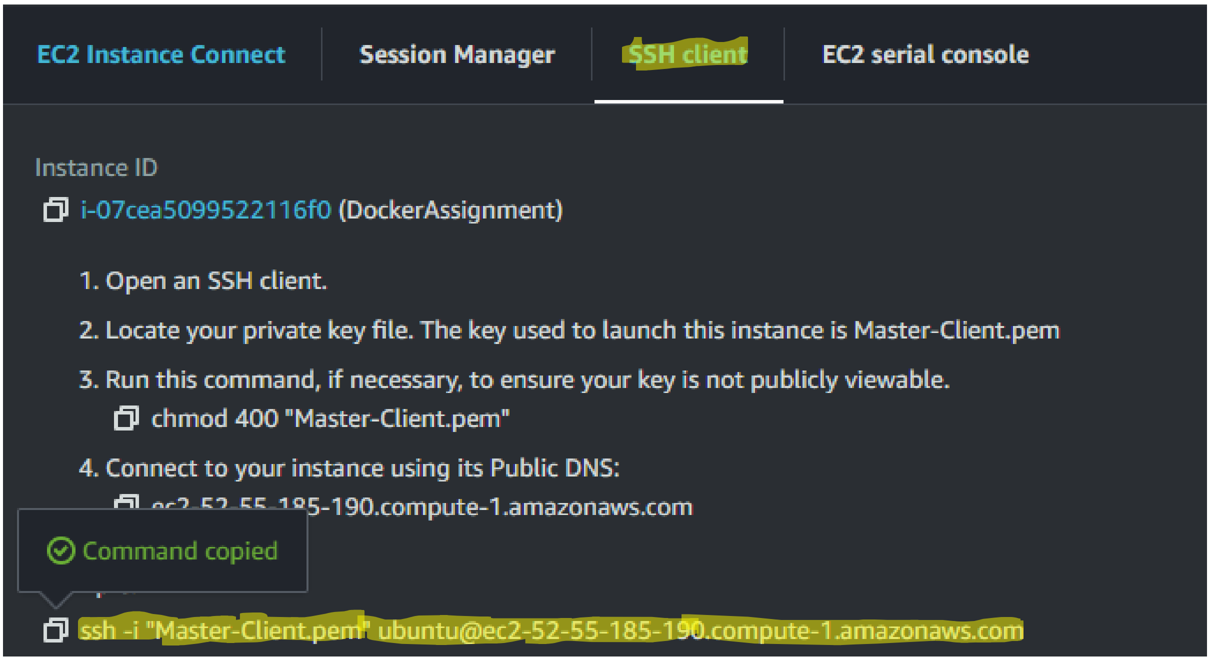
- Pull ubuntu container
- Run this container, and map port 80 on the local
- Install apache2 on this container
- Check if you are able to access the apache page on your browser

Docker Assignments - Step-by-Step Process and Before Starting the Assignment need to install Dockers on Instances(Machine) so launch Ubuntu Instances and Connect



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	A
☑	DockerAssignment	i-02476f8baed7d9809	Running	t2.micro	Initializing	View alarms +	u

Copy the SSH Client



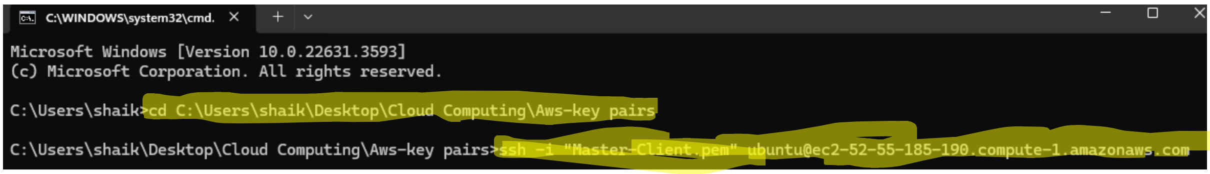
Instance ID

i-07cea5099522116f0 (DockerAssignment)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Master-Client.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   `chmod 400 "Master-Client.pem"`
4. Connect to your instance using its Public DNS:
   `ec2-52-55-185-190.compute-1.amazonaws.com`

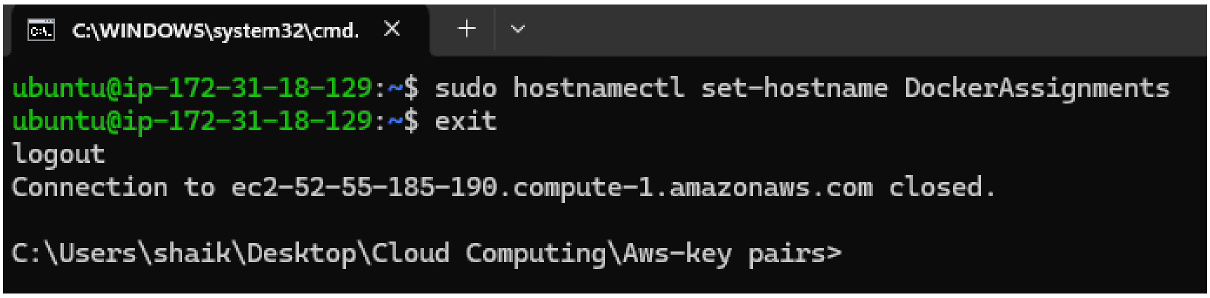
```
ssh -i "Master-Client.pem" ubuntu@ec2-52-55-185-190.compute-1.amazonaws.com
```

Go to Ur Key Location path and paste the SSH path and login it



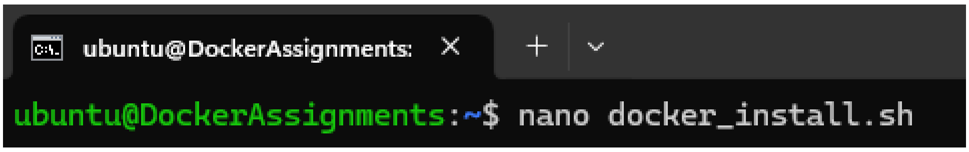
```
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shaik>cd C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-52-55-185-190.compute-1.amazonaws.com
```

Change the Hostname and Exit it and Again Login, it will Change the Hostname as u given



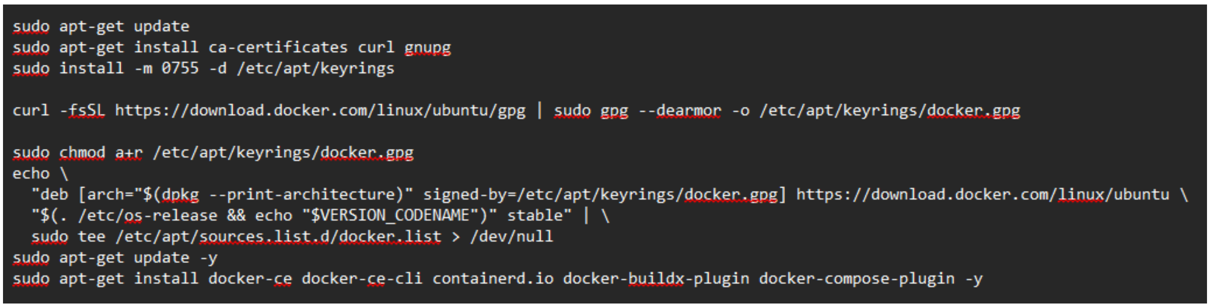
```
ubuntu@ip-172-31-18-129:~$ sudo hostnamectl set-hostname DockerAssignments
ubuntu@ip-172-31-18-129:~$ exit
logout
Connection to ec2-52-55-185-190.compute-1.amazonaws.com closed.

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>
```

Create Docker_install.sh



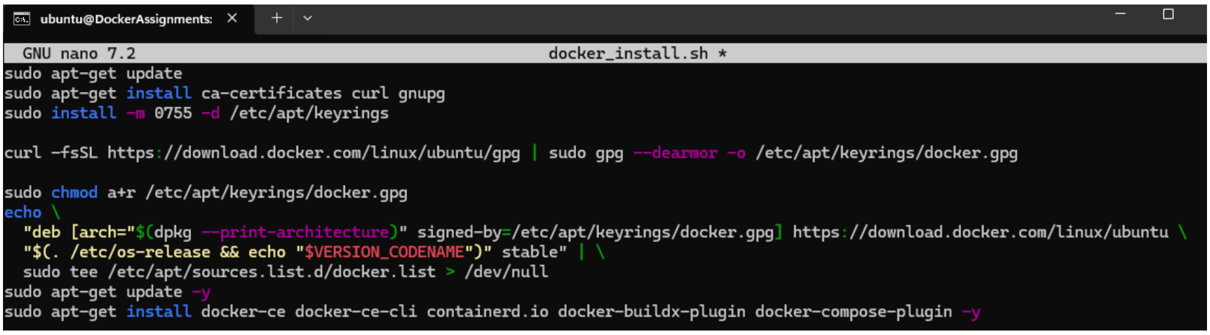
```
ubuntu@DockerAssignments:~$ nano docker_install.sh
```

Paste the Docker Installation Commands



```
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update -y
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
```

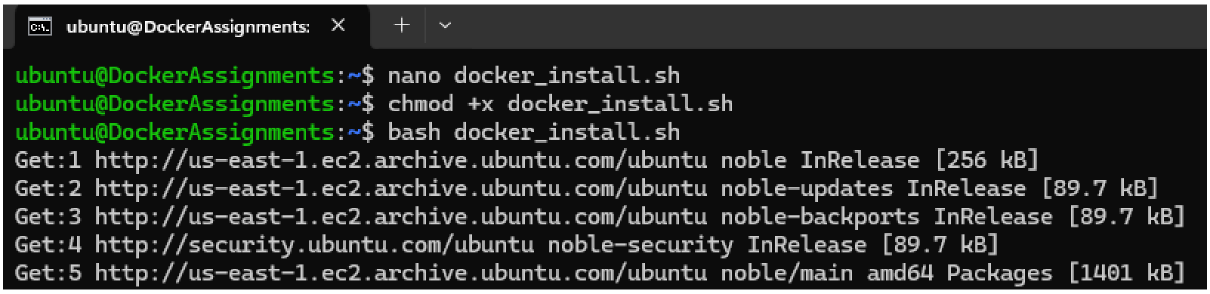
Paste it and Save it control s and control x



```
  GNU nano 7.2                                        docker_install.sh *
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update -y
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
```

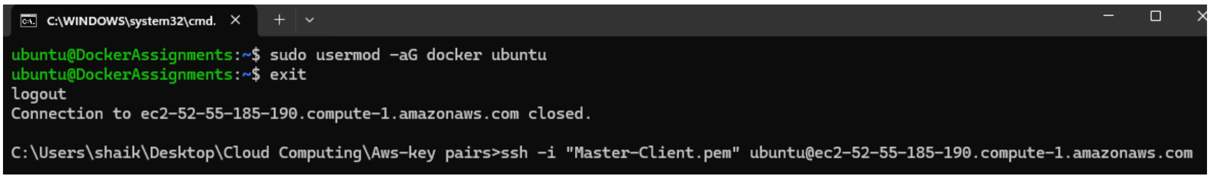
Give the Execution Permission and Now run the bash file and it will install Dockers in your machine

The command `chmod +x docker_install.sh` makes the file named "docker_install.sh" executable.



```
ubuntu@DockerAssignments:~$ nano docker_install.sh
ubuntu@DockerAssignments:~$ chmod +x docker_install.sh
ubuntu@DockerAssignments:~$ bash docker_install.sh
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [89.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [89.7 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [89.7 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 Packages [1401 kB]
```

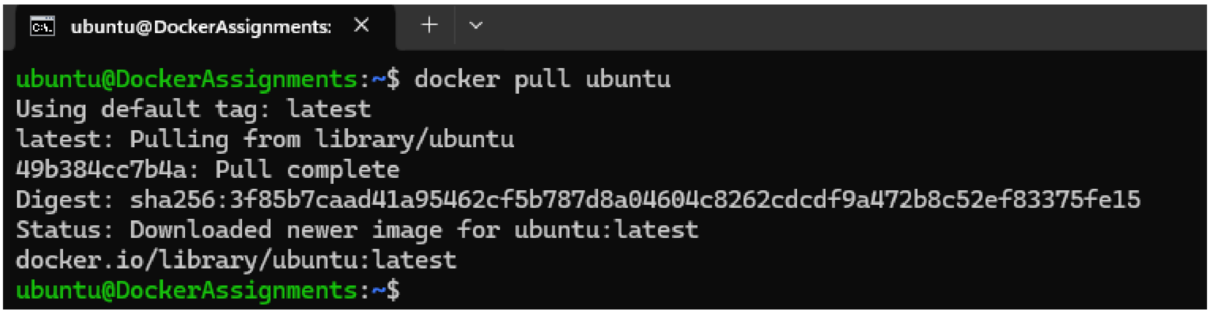
This command adds the user "ubuntu" to the "docker" group, granting them permissions associated with the Docker daemon



```
ubuntu@DockerAssignments:~$ sudo usermod -aG docker ubuntu
ubuntu@DockerAssignments:~$ exit
logout
Connection to ec2-52-55-185-190.compute-1.amazonaws.com closed.

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-52-55-185-190.compute-1.amazonaws.com
```

Step1:

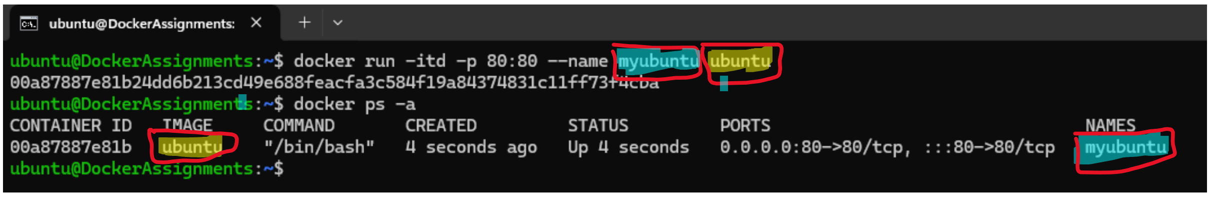
Pull the Ubuntu Container



```
ubuntu@DockerAssignments:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
49b384cc7b4a: Pull complete
Digest: sha256:3f85b7caad41a95462cf5b787d8a04604c8262cdcdf9a472b8c52ef83375fe15
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
ubuntu@DockerAssignments:~$
```

Step2:

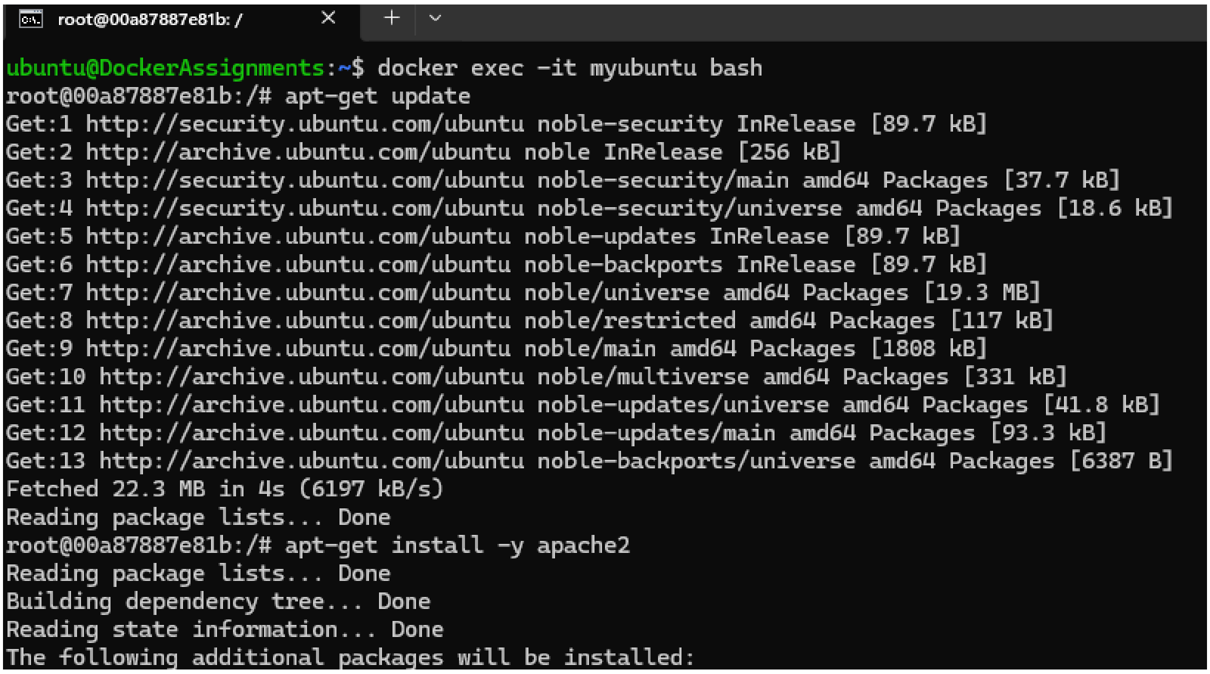
Run this Container and map Port 80 on Local here we Used name as myubuntu and Image we used ubuntu that is default image



```
ubuntu@DockerAssignments:~$ docker run -itd -p 80:80 --name myubuntu ubuntu
00a87887e81b24dd6b213cd49e688feacfa3c584f19a84374831c11ff73f4cba
ubuntu@DockerAssignments:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS         PORTS                                 NAMES
00a87887e81b   ubuntu    "/bin/bash"   4 seconds ago   Up 4 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp   myubuntu
ubuntu@DockerAssignments:~$
```

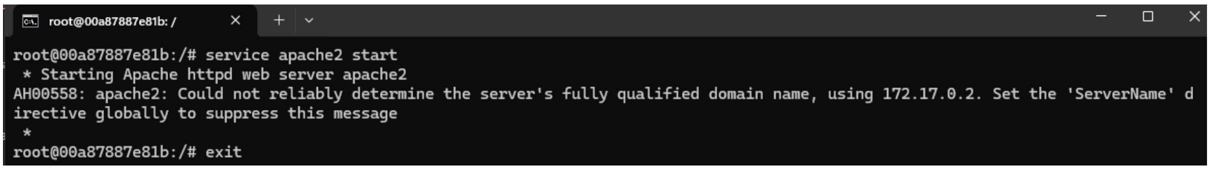
Step3:

Docker exec -it myubuntu bash means going inside the myubuntu. And Need to update the package



```
ubuntu@DockerAssignments:~$ docker exec -it myubuntu bash
root@00a87887e81b:/# apt-get update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [89.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [37.7 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [18.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease [89.7 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease [89.7 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:8 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [41.8 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [93.3 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [6387 B]
Fetched 22.3 MB in 4s (6197 kB/s)
Reading package lists... Done
root@00a87887e81b:/# apt-get install -y apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

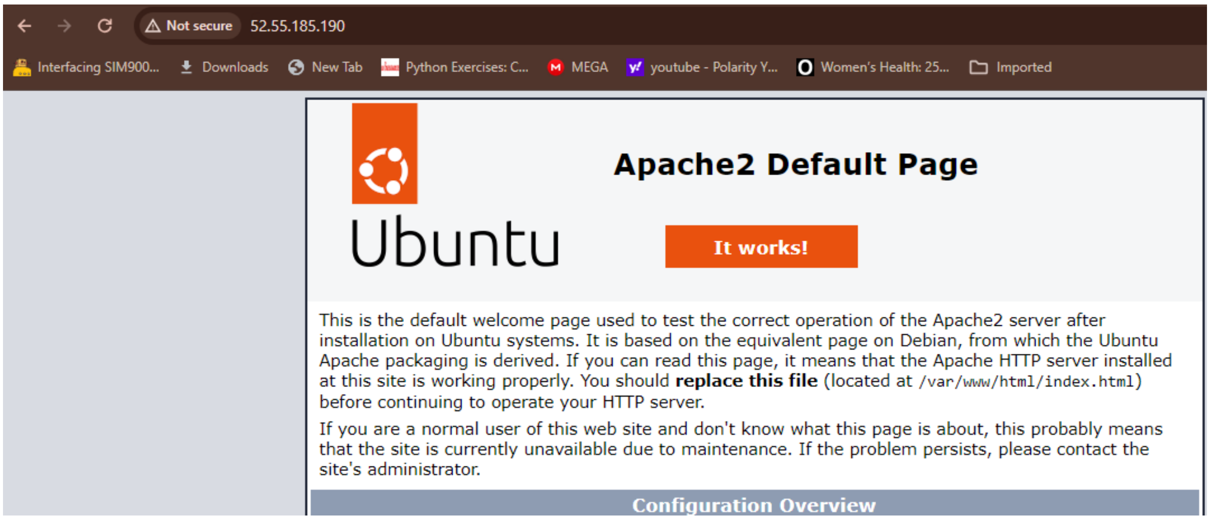
We start apache2 Service and Exited from the my ubuntu container



```
root@00a87887e81b:/# service apache2 start
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' d
irective globally to suppress this message
 *
root@00a87887e81b:/# exit
```

Step 4:

Copy Instances public IP and paste on the browser and it show the default apache2 page



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Module-3: Docker – I Assignment - 2

You have been asked to:

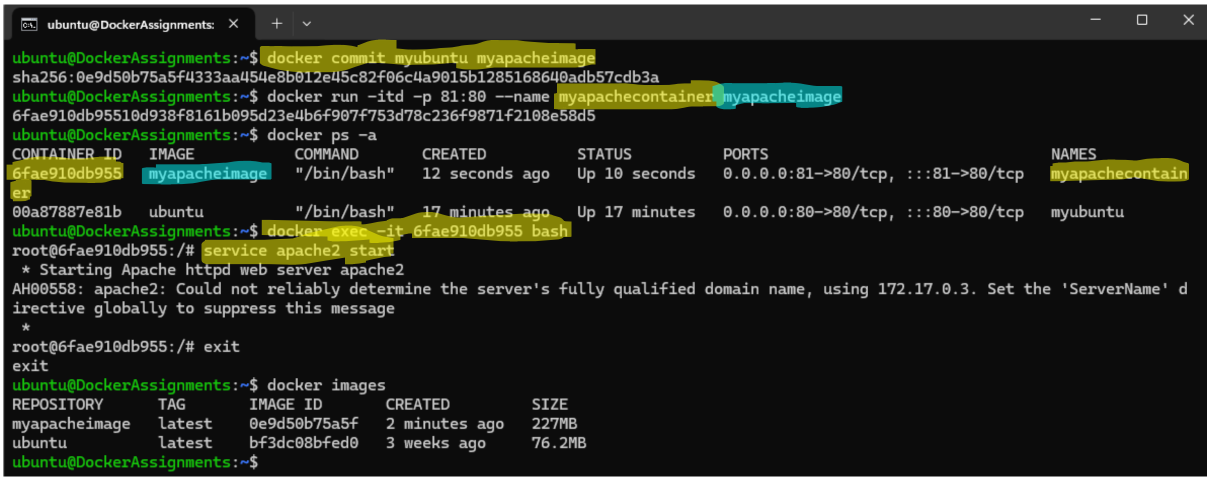
- Save the image created in Assignment 1 as a Docker image
- Launch container from this new image and map the port to 81
- Go inside the container and start the apache2 service
- Check if you are able to access it on the browser

Step 1,2 & 3:

Step1: Save the myubuntu as a myapacheimage

Step2: Launching New Container using of Assignment 1 docker image(myapacheimage) and port to 81 and name given as myapachecontainer

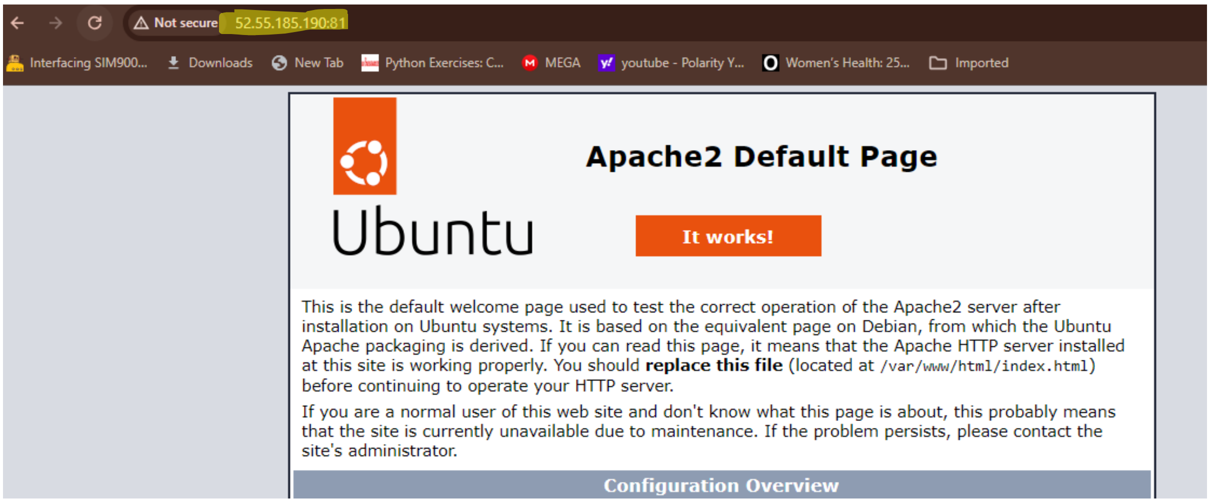
Step3: Go to inside the Container and started the apache2 service and exit and check images(2 images available)



```
ubuntu@DockerAssignments:~$ docker commit myubuntu myapacheimage
sha256:0e9d50b75a5f4333aa454e8b012e45c82f06c4a9015b1285168640adb57cdb3a
ubuntu@DockerAssignments:~$ docker run -itd -p 81:80 --name myapachecontainer myapacheimage
6fae910db95510d938f8161b095d23e4b6f907f753d78c236f9871f2108e58d5
ubuntu@DockerAssignments:~$ docker ps -a
CONTAINER ID   IMAGE           COMMAND       CREATED          STATUS          PORTS                                 NAMES
6fae910db955   myapacheimage   "/bin/bash"   12 seconds ago   Up 10 seconds   0.0.0.0:81->80/tcp, :::81->80/tcp   myapachecontain
er
00a87887e81b   ubuntu          "/bin/bash"   17 minutes ago   Up 17 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp   myubuntu
ubuntu@DockerAssignments:~$ docker exec -it 6fae910db955 bash
root@6fae910db955:/# service apache2 start
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' d
irective globally to suppress this message
 *
root@6fae910db955:/# exit
exit
ubuntu@DockerAssignments:~$ docker images
REPOSITORY      TAG       IMAGE ID       CREATED         SIZE
myapacheimage   latest    0e9d50b75a5f   2 minutes ago   227MB
ubuntu          latest    bf3dc08bfed0   3 weeks ago     76.2MB
ubuntu@DockerAssignments:~$
```

Step4:

We can see in the Port 81 it successfully showing default Apache2 web page



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

DevOps Certification Training

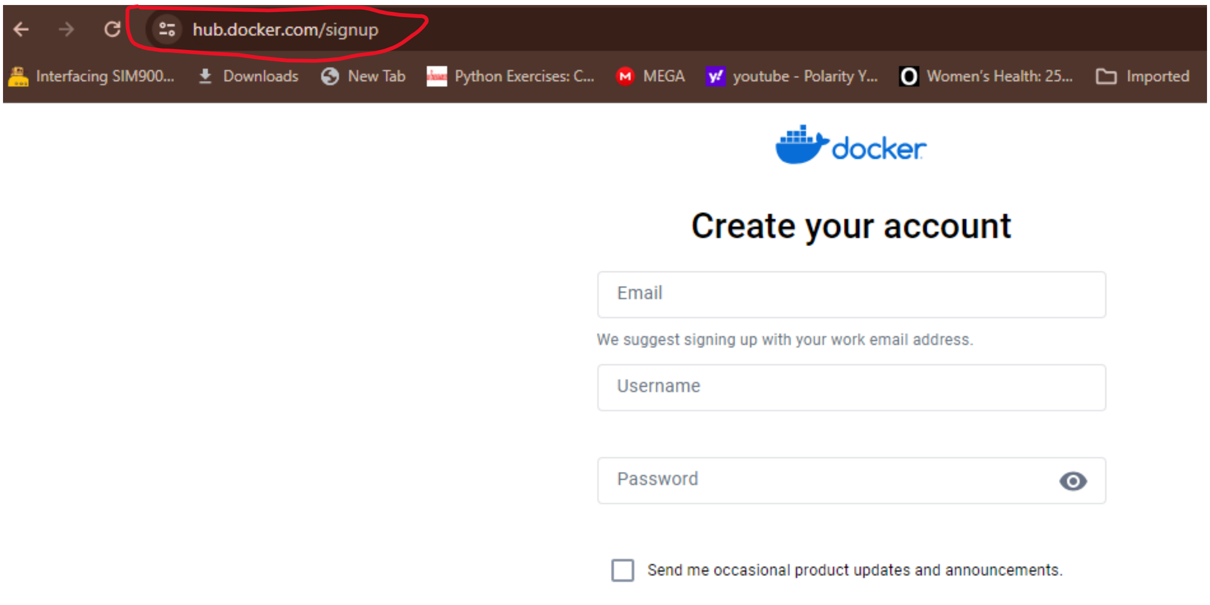
Module-3: Docker – I Assignment - 3

You have been asked to:

- Use the saved image in the previous assignment
- Upload this image on Dockerhub
- On a separate machine pull this dockerhub image, and launch it on port 80
- Start the apache2 service
- Verify if you are able to see the apache2 service

Before Going to Perform the Assignment 3 Task we Need DockerHub Account

Create Account


hub.docker.com/signup

Create your account

Email

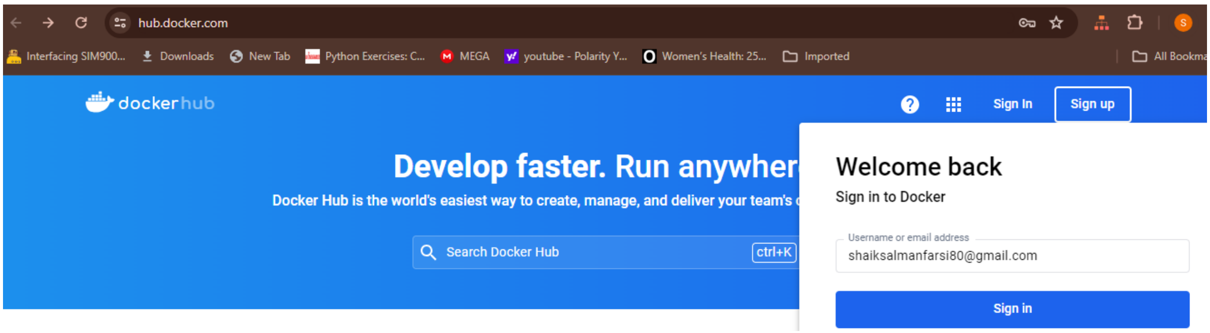
We suggest signing up with your work email address.

Username

Password

□ Send me occasional product updates and announcements.

After Creating the Account go sign in


hub.docker.com

Develop faster. Run anywher

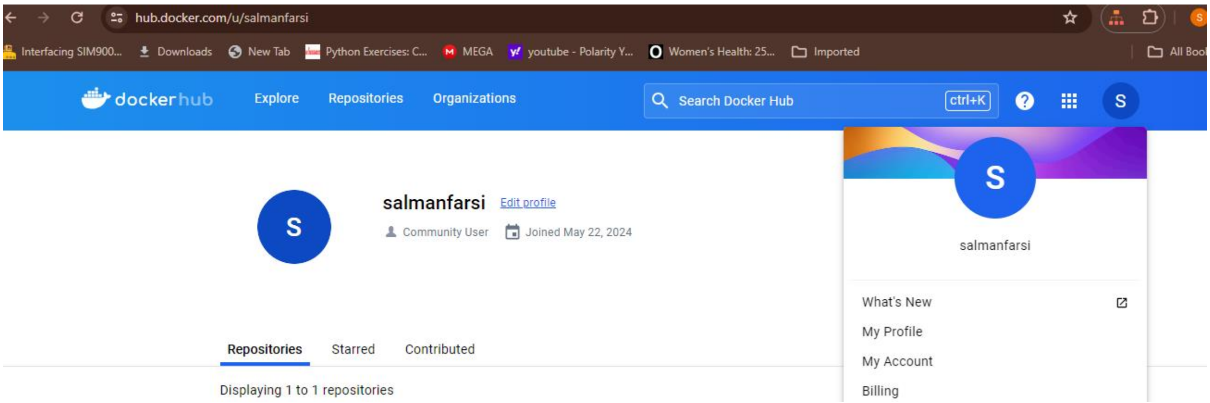
Docker Hub is the world's easiest way to create, manage, and deliver your team's c

Welcome back

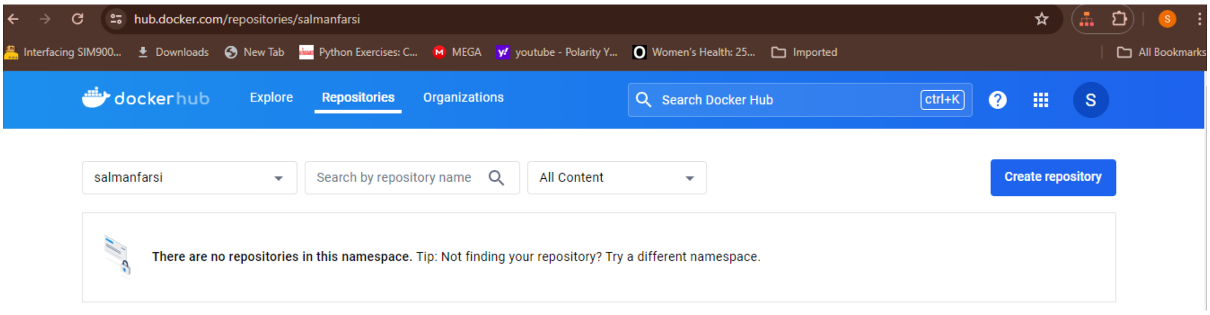
Sign in to Docker

Username or email address: shaiksalmanfarsi80@gmail.com

Salmanfarsi is my username



Currently there is no Repository

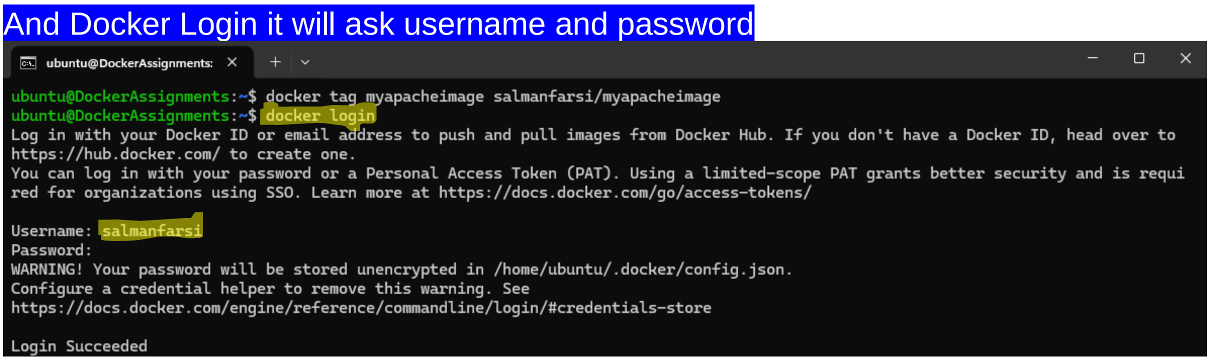


There are no repositories in this namespace. Tip: Not finding your repository? Try a different namespace.

Step1:

The command `docker tag myapacheimage salmanfarsi/myapacheimage` creates a new "tag" for an existing Docker image

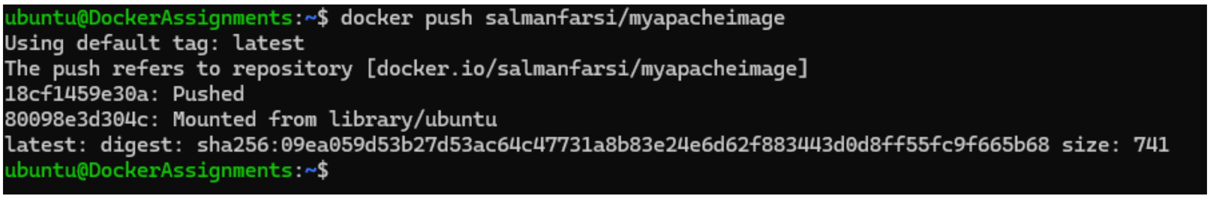
And Docker Login it will ask username and password



```
ubuntu@DockerAssignments:~$ docker tag myapacheimage salmanfarsi/myapacheimage
ubuntu@DockerAssignments:~$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to
https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is requi
red for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: salmanfarsi
Password:
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

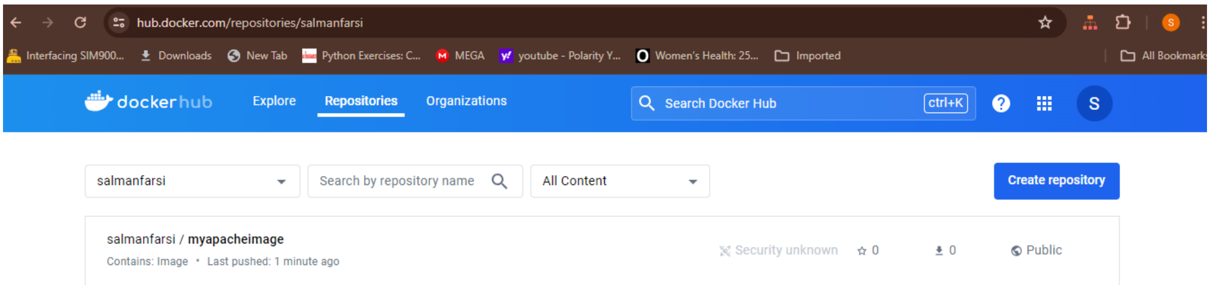
Step2:

Image Upload in dockerhub means docker push username/Repository = docker push salmanfarsi/myapacheimage and Its Successfully Pushed.



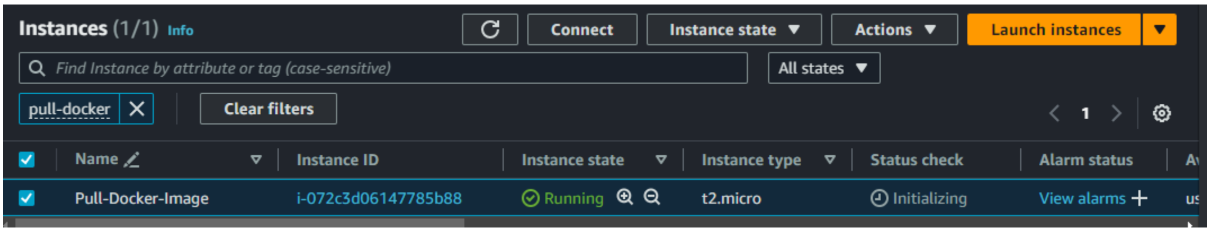
```
ubuntu@DockerAssignments:~$ docker push salmanfarsi/myapacheimage
Using default tag: latest
The push refers to repository [docker.io/salmanfarsi/myapacheimage]
18cf1459e30a: Pushed
80098e3d304c: Mounted from library/ubuntu
latest: digest: sha256:09ea059d53b27d53ac64c47731a8b83e24e6d62f883443d0d8ff55fc9f665b68 size: 741
ubuntu@DockerAssignments:~$
```

Now its Uploaded successfully



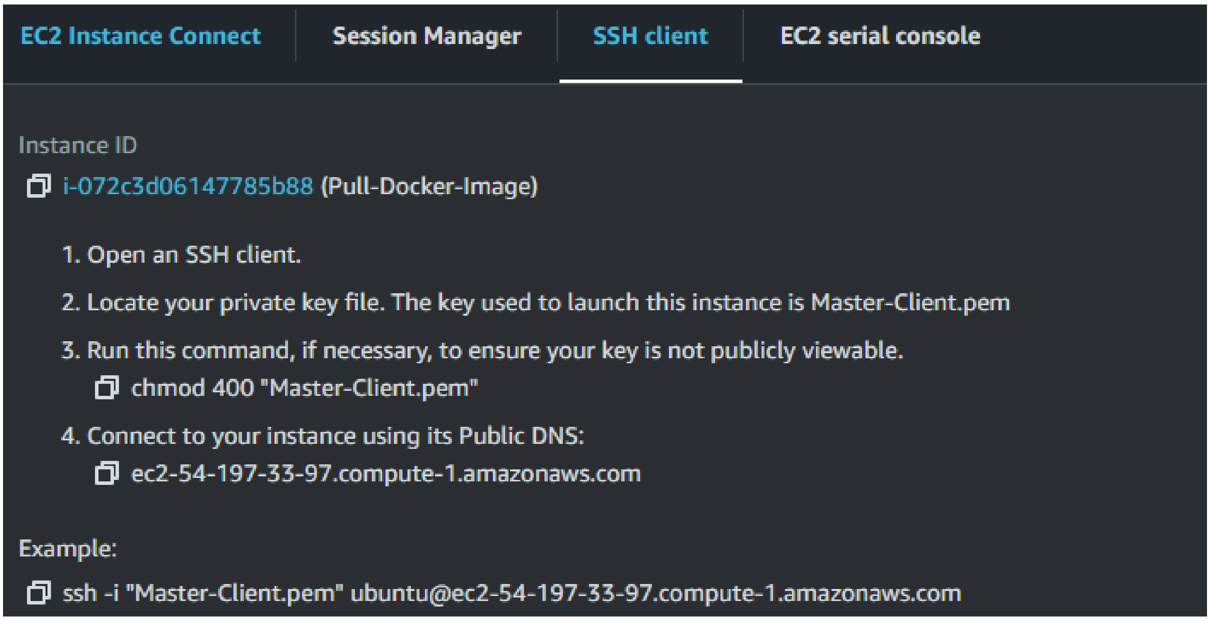
salmanfarsi / myapacheimage
Contains: Image • Last pushed: 1 minute ago
Security unknown ☆ 0 ⬇ 0 Public

Step3: Launch Another Instances to Pull Our Image



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Av
☑	Pull-Docker-Image	i-072c3d06147785b88	✓ Running	t2.micro	Initializing	View alarms +	us

Copy ssh client



Instance ID

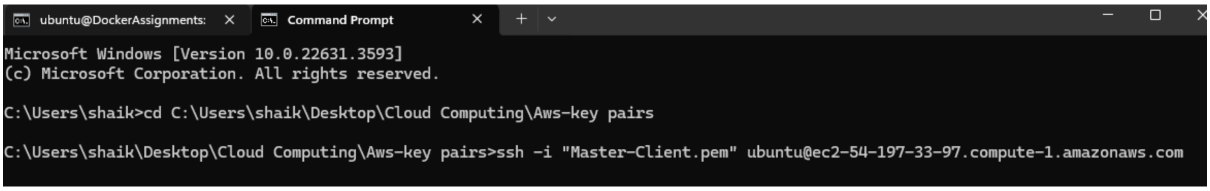
i-072c3d06147785b88 (Pull-Docker-Image)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Master-Client.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "Master-Client.pem"
4. Connect to your instance using its Public DNS:
   ec2-54-197-33-97.compute-1.amazonaws.com

Example:

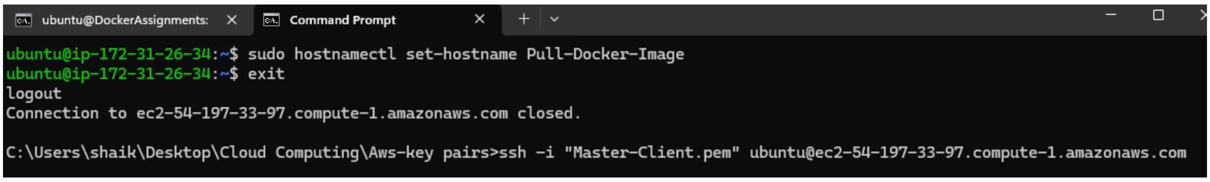
ssh -i "Master-Client.pem" ubuntu@ec2-54-197-33-97.compute-1.amazonaws.com

Change directory to Private Key path which we personally saved and enter and paste ssh client path and login it



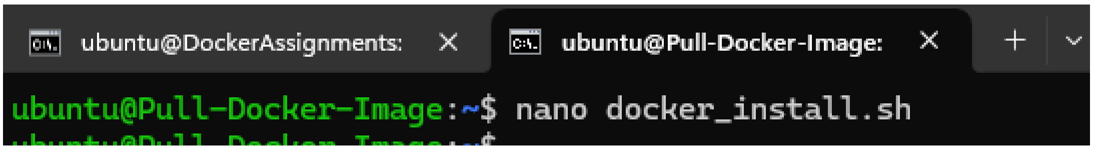
```
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shaik>cd C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-197-33-97.compute-1.amazonaws.com
```

Change hostname



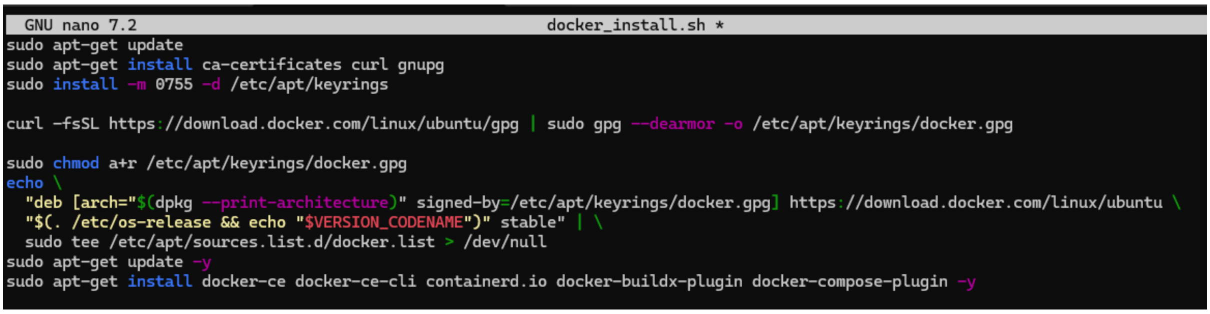
```
ubuntu@ip-172-31-26-34:~$ sudo hostnamectl set-hostname Pull-Docker-Image
ubuntu@ip-172-31-26-34:~$ exit
logout
Connection to ec2-54-197-33-97.compute-1.amazonaws.com closed.

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-197-33-97.compute-1.amazonaws.com
```

Install dockers in this pulling instances



```
ubuntu@Pull-Docker-Image:~$ nano docker_install.sh
ubuntu@Pull-Docker-Image:~$
```

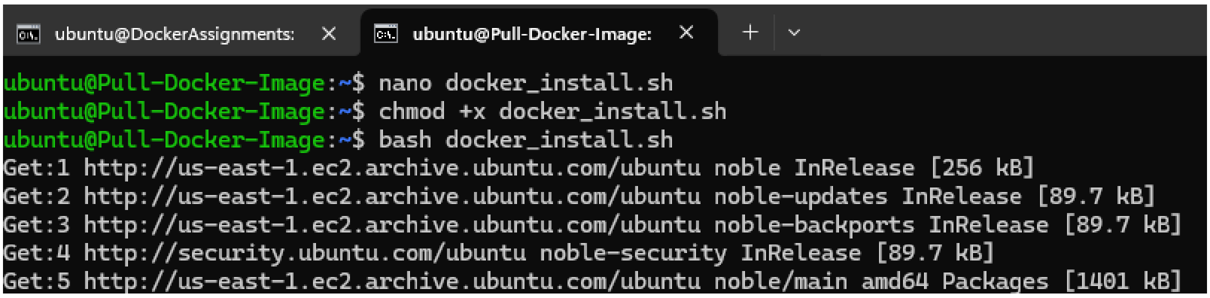
Paste the Installation stuff and save it



```
  GNU nano 7.2                                              docker_install.sh *
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update -y
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
```

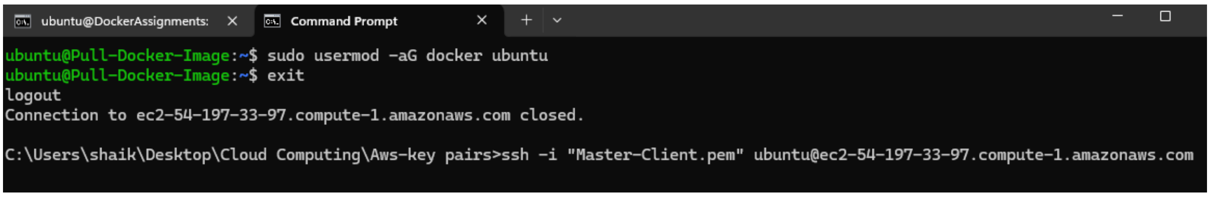
The command `chmod +x docker_install.sh` makes the file named "docker_install.sh" executable.

The command `bash docker_install.sh` executes a script named `docker_install.sh` using the bash shell, likely with the intention of installing Docker on your system.



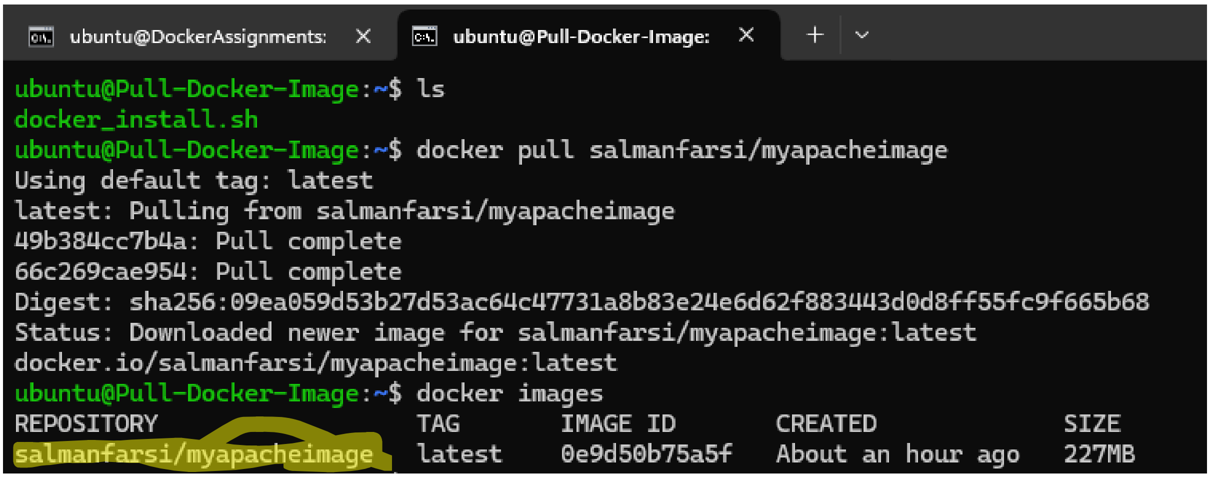
```
ubuntu@Pull-Docker-Image:~$ nano docker_install.sh
ubuntu@Pull-Docker-Image:~$ chmod +x docker_install.sh
ubuntu@Pull-Docker-Image:~$ bash docker_install.sh
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [89.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [89.7 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [89.7 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 Packages [1401 kB]
```

This command adds the user "ubuntu" to the "docker" group, granting them permissions associated with the Docker daemon



```
ubuntu@Pull-Docker-Image:~$ sudo usermod -aG docker ubuntu
ubuntu@Pull-Docker-Image:~$ exit
logout
Connection to ec2-54-197-33-97.compute-1.amazonaws.com closed.

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-197-33-97.compute-1.amazonaws.com
```

Pull the Image from salmanfarsi/myapacheimage

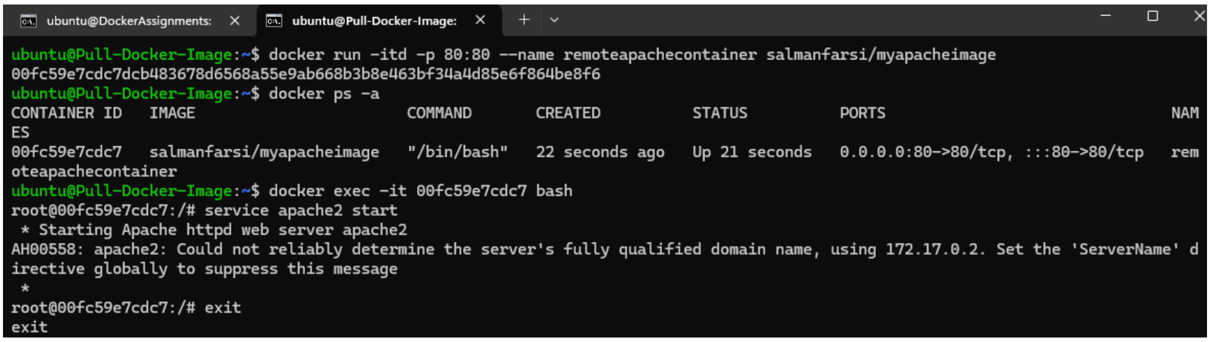


```
ubuntu@Pull-Docker-Image:~$ ls
docker_install.sh
ubuntu@Pull-Docker-Image:~$ docker pull salmanfarsi/myapacheimage
Using default tag: latest
latest: Pulling from salmanfarsi/myapacheimage
49b384cc7b4a: Pull complete
66c269cae954: Pull complete
Digest: sha256:09ea059d53b27d53ac64c47731a8b83e24e6d62f883443d0d8ff55fc9f665b68
Status: Downloaded newer image for salmanfarsi/myapacheimage:latest
docker.io/salmanfarsi/myapacheimage:latest
ubuntu@Pull-Docker-Image:~$ docker images
REPOSITORY                  TAG       IMAGE ID       CREATED              SIZE
salmanfarsi/myapacheimage   latest    0e9d50b75a5f   About an hour ago    227MB
```

And Running New container named as “remoteapachecontainer” using salmanfarsi/myapacheimage which is already apache2 installed.

Step4:

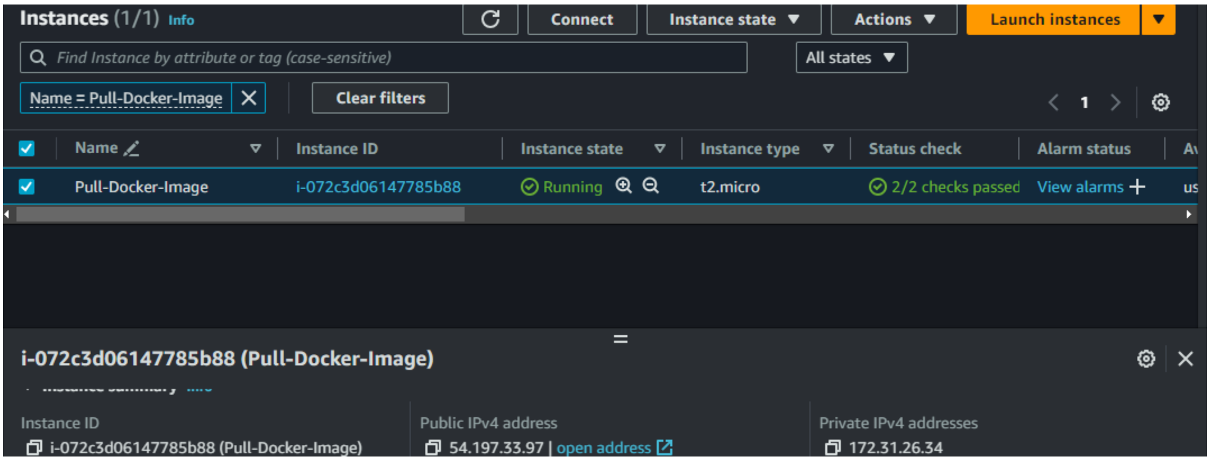
Go to Inside the container and Start the apache2 Service and Exit



```
ubuntu@Pull-Docker-Image:~$ docker run -itd -p 80:80 --name remoteapachecontainer salmanfarsi/myapacheimage
00fc59e7cdc7dcb483678d6568a55e9ab668b3b8e463bf34a4d85e6f864be8f6
ubuntu@Pull-Docker-Image:~$ docker ps -a
CONTAINER ID   IMAGE                       COMMAND       CREATED          STATUS          PORTS                               NAM
ES
00fc59e7cdc7   salmanfarsi/myapacheimage   "/bin/bash"   22 seconds ago   Up 21 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp   rem
oteapachecontainer
ubuntu@Pull-Docker-Image:~$ docker exec -it 00fc59e7cdc7 bash
root@00fc59e7cdc7:/# service apache2 start
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' d
irective globally to suppress this message
 *
root@00fc59e7cdc7:/# exit
exit
```

Step5:

Go to Pull Docker Image Instances and copy IP and Paste in the Web Browser



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Av
☑	Pull-Docker-Image	i-072c3d06147785b88	Running	t2.micro	2/2 checks passed	View alarms +	us

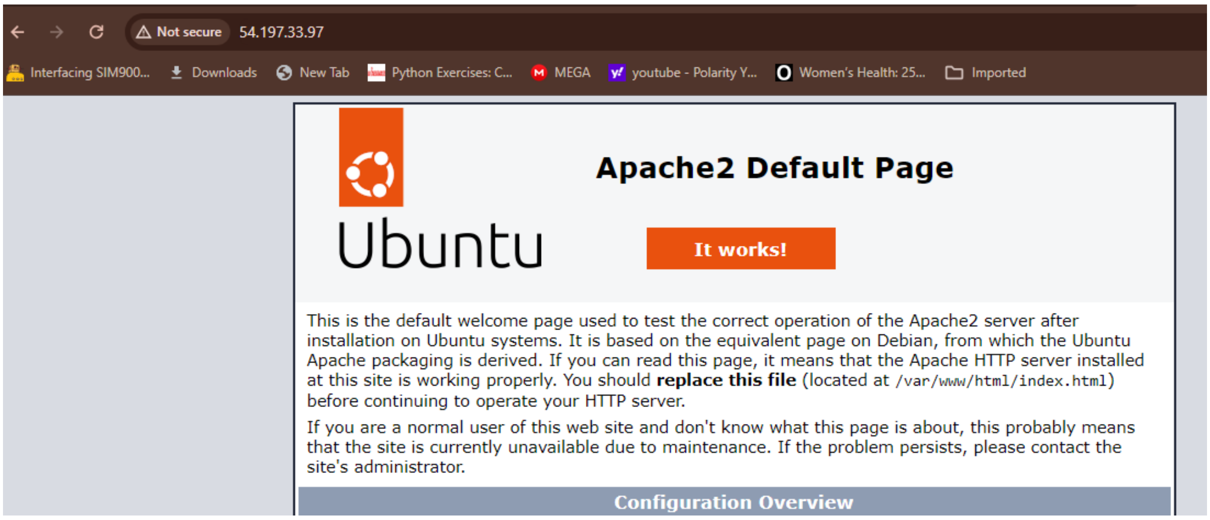
i-072c3d06147785b88 (Pull-Docker-Image)

Instance ID: i-072c3d06147785b88 (Pull-Docker-Image)

Public IPv4 address: 54.197.33.97 | open address

Private IPv4 addresses: 172.31.26.34

Successfully we can the Apache2 Default Page in the Pull Docker Image



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

DevOps Certification Training

IntelliPaat

Module-3: Docker – I Assignment – 4

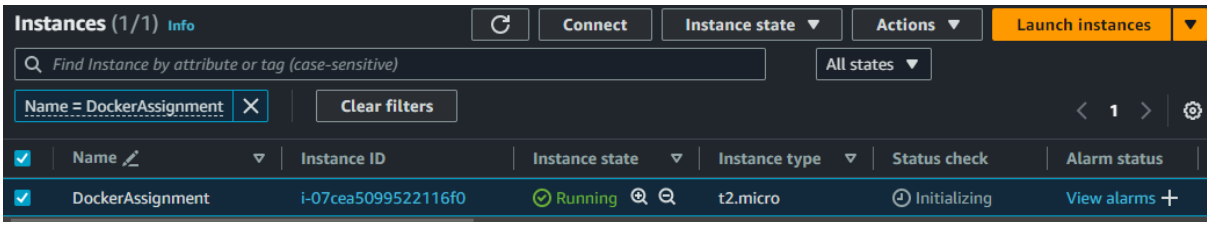
You have been asked to:

Create a dockerfile with the following specs:

- Ubuntu container
- Apache2 installed
- Apache2 should automatically run once the container starts

Submit the dockerfile, for assignment completion

Launch Instances:

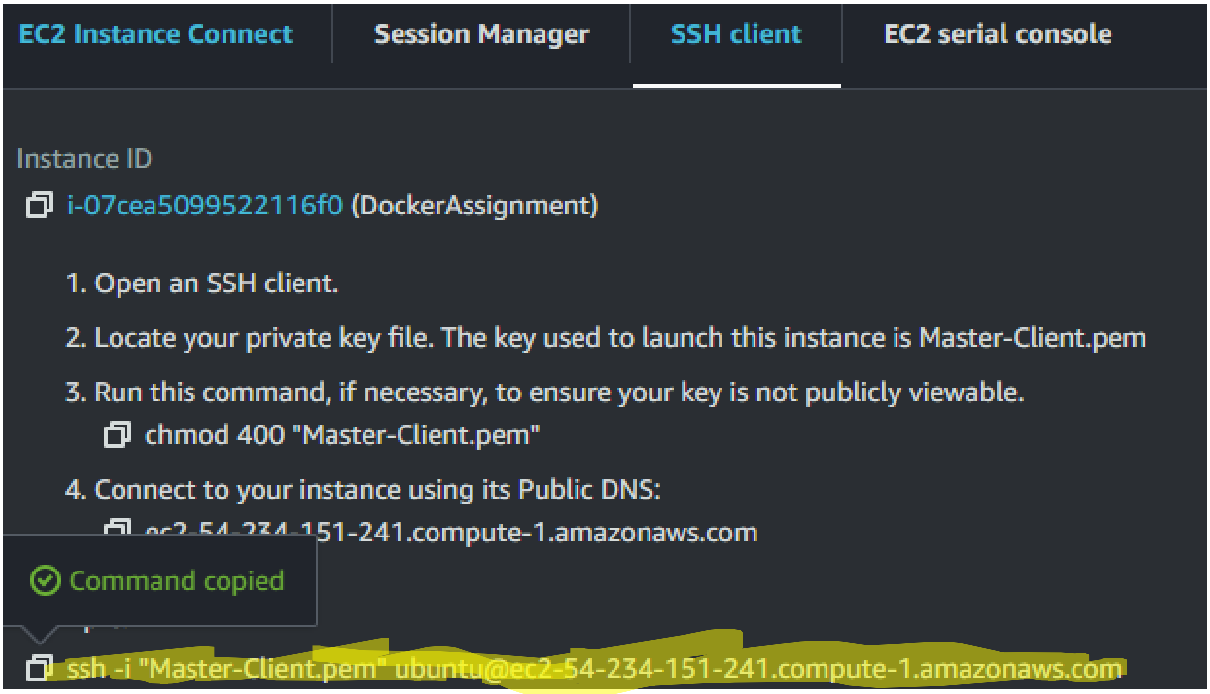


Instances (1/1) Info

Name = DockerAssignment

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☑	DockerAssignment	i-07cea5099522116f0	Running	t2.micro	Initializing	View alarms

Copy SSH Client



Instance ID

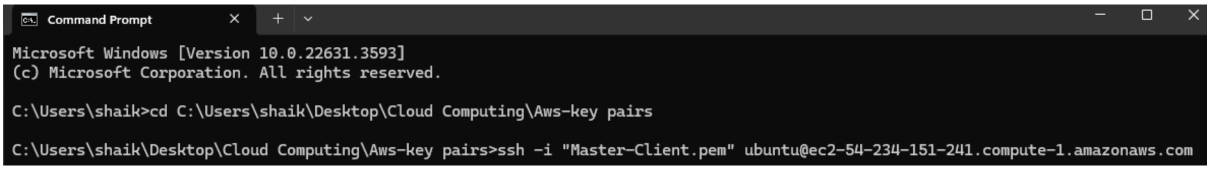
i-07cea5099522116f0 (DockerAssignment)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Master-Client.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   ```
   chmod 400 "Master-Client.pem"
   ```
4. Connect to your instance using its Public DNS:
   ```
   ec2-54-234-151-241.compute-1.amazonaws.com
   ```

Command copied

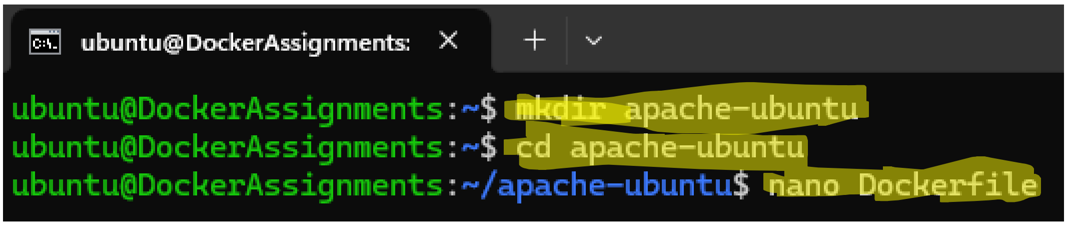
```
ssh -i "Master-Client.pem" ubuntu@ec2-54-234-151-241.compute-1.amazonaws.com
```

Change Directory, And Copy Key-Pair path, Enter, And After Paste SSH Client



```
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shaik>cd C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-234-151-241.compute-1.amazonaws.com
```

Step 1,2 & 3: Create a New Directory and go to inside the Directory and Create Dockerfile

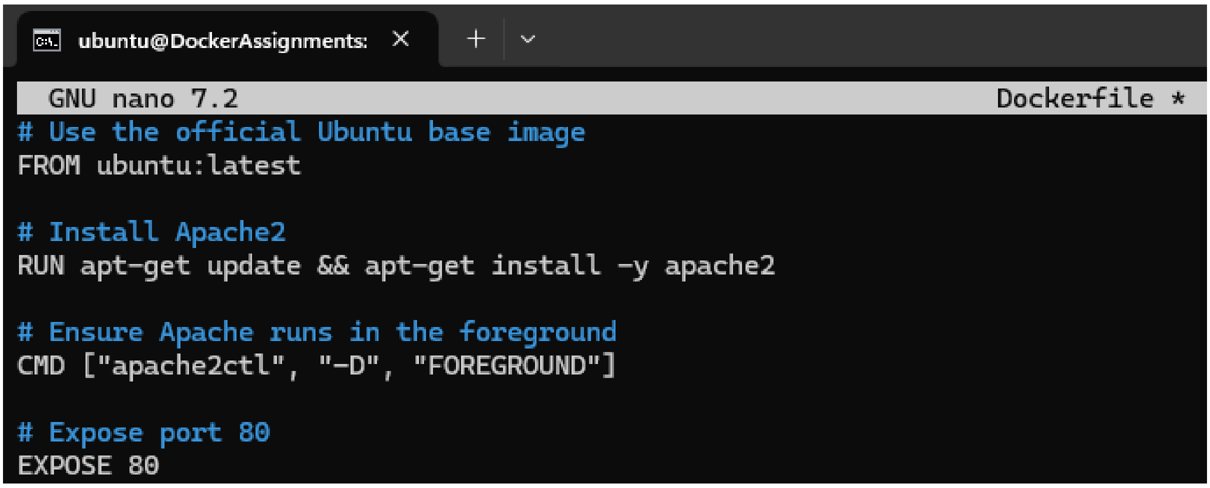


```
ubuntu@DockerAssignments:~$ mkdir apache-ubuntu
ubuntu@DockerAssignments:~$ cd apache-ubuntu
ubuntu@DockerAssignments:~/apache-ubuntu$ nano Dockerfile
```

Edit the Dockerfile Open the `Dockerfile` in your preferred text editor and add the following content

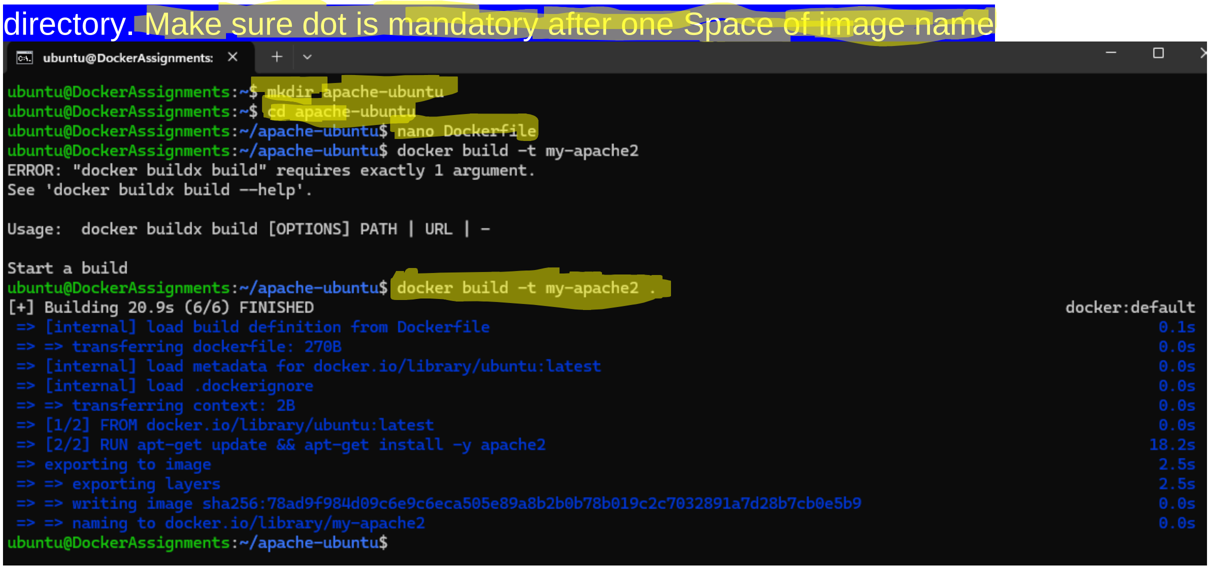
This Dockerfile creates a container that automatically starts and runs an Apache2 web server, exposing port 80 for web traffic.

by setting the default command (`CMD`) to run `apache2ctl -D FOREGROUND` which keeps the container running in the foreground with Apache2 active.



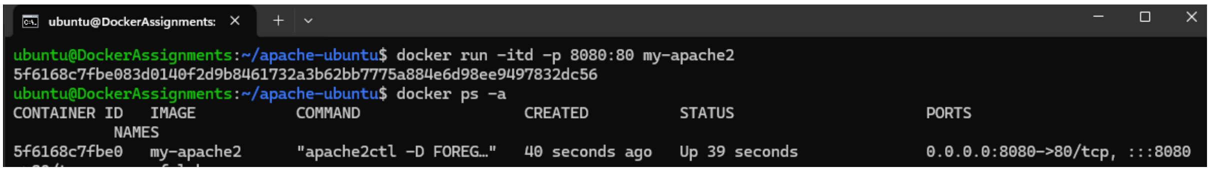
```
  GNU nano 7.2                                              Dockerfile *
# Use the official Ubuntu base image
FROM ubuntu:latest

# Install Apache2
RUN apt-get update && apt-get install -y apache2

# Ensure Apache runs in the foreground
CMD ["apache2ctl", "-D", "FOREGROUND"]

# Expose port 80
EXPOSE 80
```

Builds a Docker image named "my-apache2" from the Dockerfile in the current directory. Make sure dot is mandatory after one Space of image name



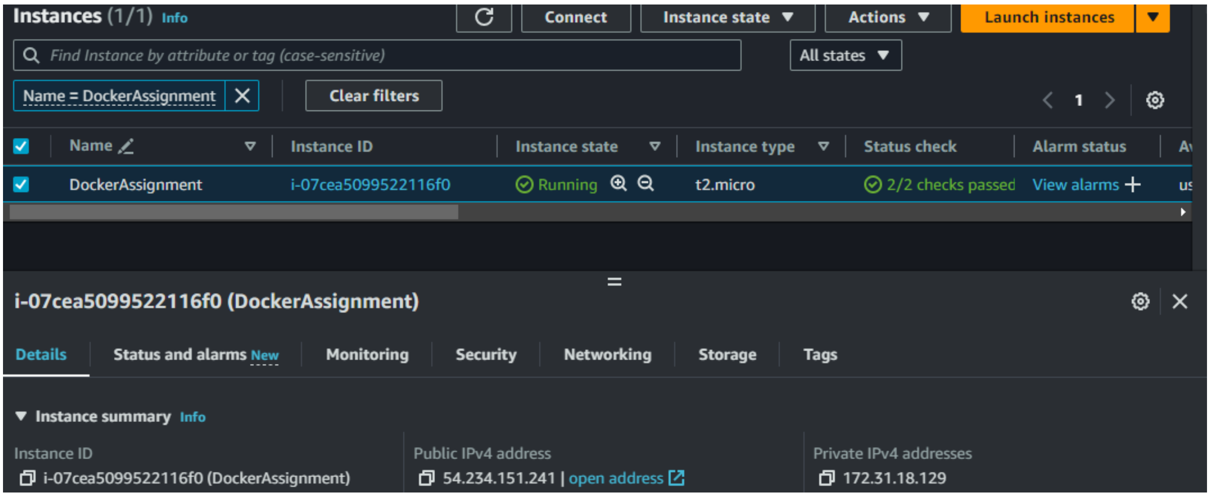
```
ubuntu@DockerAssignments:~$ mkdir apache-ubuntu
ubuntu@DockerAssignments:~$ cd apache-ubuntu
ubuntu@DockerAssignments:~/apache-ubuntu$ nano Dockerfile
ubuntu@DockerAssignments:~/apache-ubuntu$ docker build -t my-apache2
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage:  docker buildx build [OPTIONS] PATH | URL | -

Start a build
ubuntu@DockerAssignments:~/apache-ubuntu$ docker build -t my-apache2 .
[+] Building 20.9s (6/6) FINISHED                                                           docker:default
 => [internal] load build definition from Dockerfile                                                  0.1s
 => => transferring dockerfile: 270B                                                                  0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                      0.0s
 => [internal] load .dockerignore                                                                     0.0s
 => => transferring context: 2B                                                                       0.0s
 => [1/2] FROM docker.io/library/ubuntu:latest                                                        0.0s
 => [2/2] RUN apt-get update && apt-get install -y apache2                                           18.2s
 => exporting to image                                                                                2.5s
 => => exporting layers                                                                               2.5s
 => => writing image sha256:78ad9f984d09c6e9c6eca505e89a8b2b0b78b019c2c7032891a7d28b7cb0e5b9          0.0s
 => => naming to docker.io/library/my-apache2                                                         0.0s
ubuntu@DockerAssignments:~/apache-ubuntu$
```

Run the Docker container And Port to 8080



```
ubuntu@DockerAssignments:~/apache-ubuntu$ docker run -itd -p 8080:80 my-apache2
5f6168c7fbe083d0140f2d9b8461732a3b62bb7775a884e6d98ee9497832dc56
ubuntu@DockerAssignments:~/apache-ubuntu$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS                        PORTS
          NAMES
5f6168c7fbe0   my-apache2     "apache2ctl -D FOREG…"   40 seconds ago   Up 39 seconds                 0.0.0.0:8080->80/tcp, :::8080
```

Go to Instances and Copy the Public IP



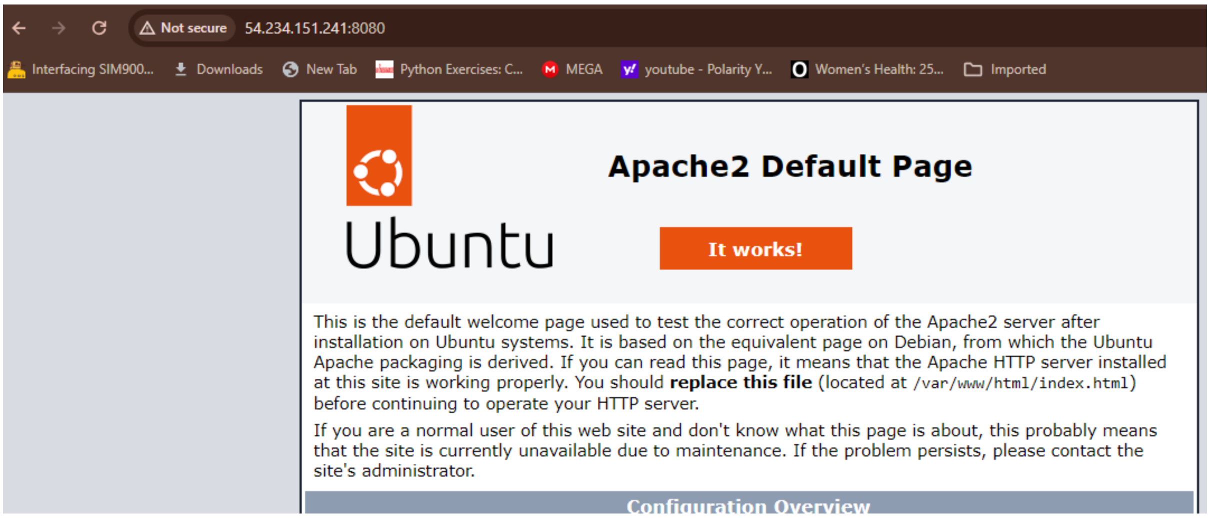
☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Av
☑	DockerAssignment	i-07cea5099522116f0	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us

i-07cea5099522116f0 (DockerAssignment)

▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-07cea5099522116f0 (DockerAssignment)	54.234.151.241 | open address	172.31.18.129

Successfully showing Default Browser



Apache2 Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Dockerfile:

```
FROM ubuntu:latest

RUN apt-get update && apt-get install -y apache2

CMD ["apache2ctl", "-D", "FOREGROUND"]

EXPOSE 80
```

DevOps Certification Training

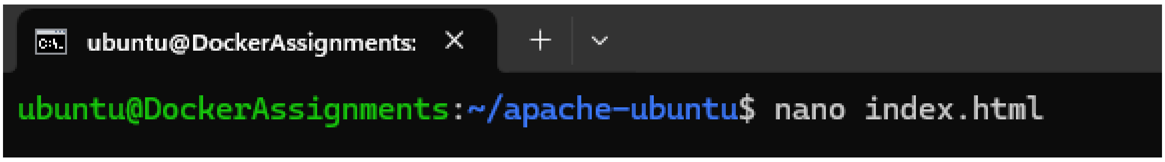
IntelliPaat

Module-3: Docker – I Assignment – 5

You have been asked to:

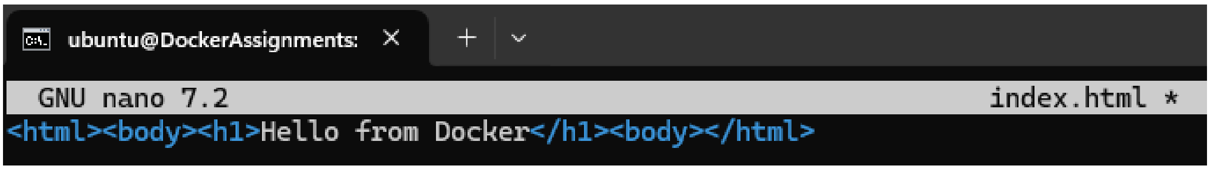
- Create a sample HTML file
- Use the Dockerfile from the previous task
- Replace this sample HTML file inside the docker container with the default page

Step1: Create a Sample HTML file



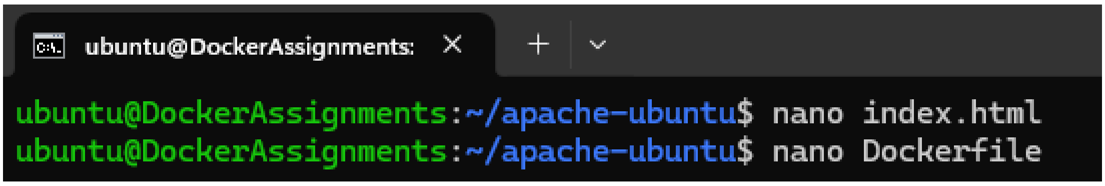
```
ubuntu@DockerAssignments:~/apache-ubuntu$ nano index.html
```

Type the Sample Content and Save it.



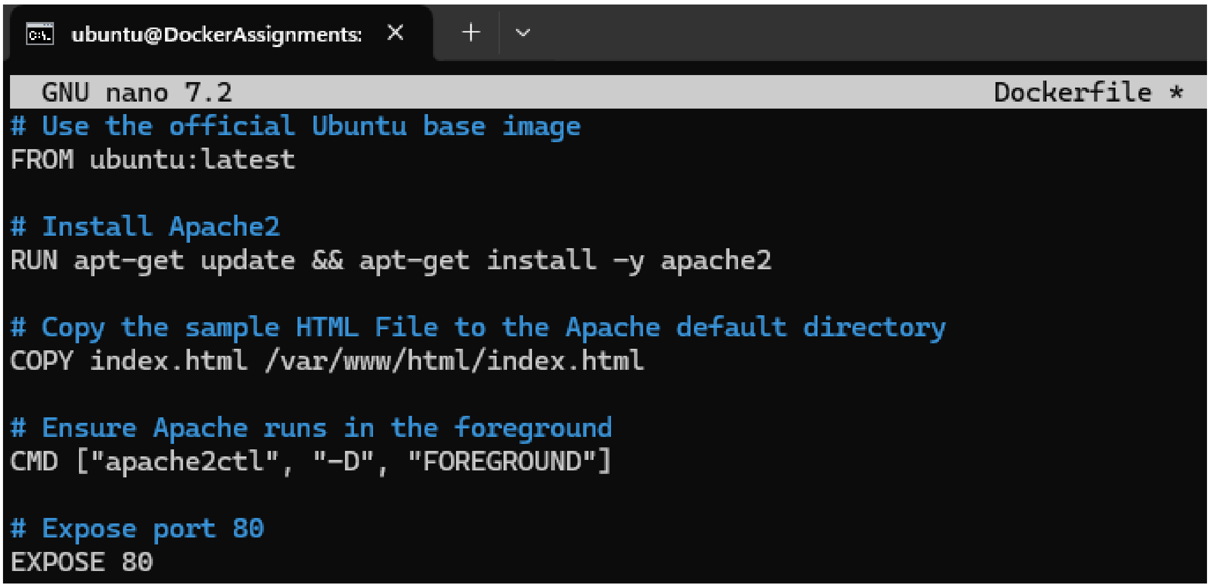
```
 GNU nano 7.2                                              index.html *
<html><body><h1>Hello from Docker</h1><body></html>
```

Step2: Using Previous Dockerfile



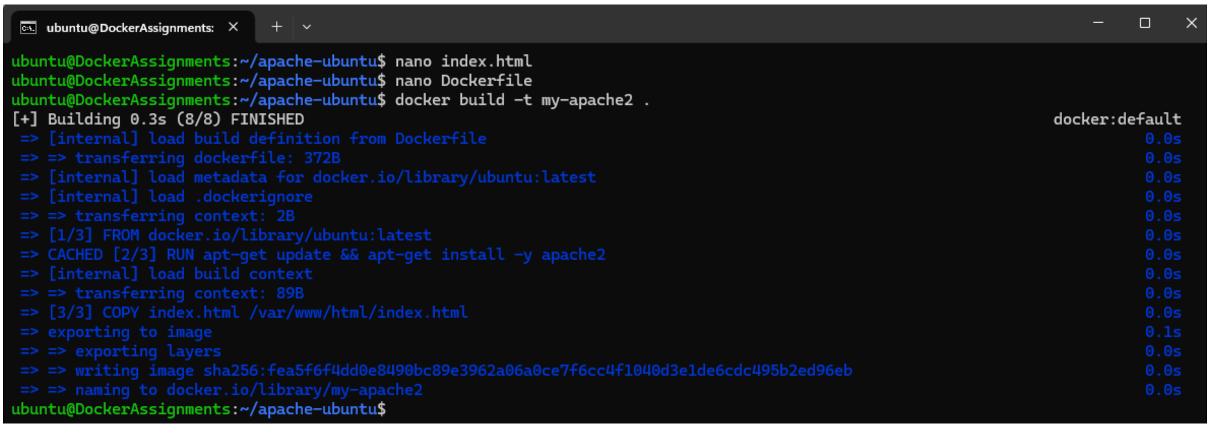
```
ubuntu@DockerAssignments:~/apache-ubuntu$ nano index.html
ubuntu@DockerAssignments:~/apache-ubuntu$ nano Dockerfile
```

Step3: Do Changes Copy the Sample HTML File and Paste /var/www/html/index.html



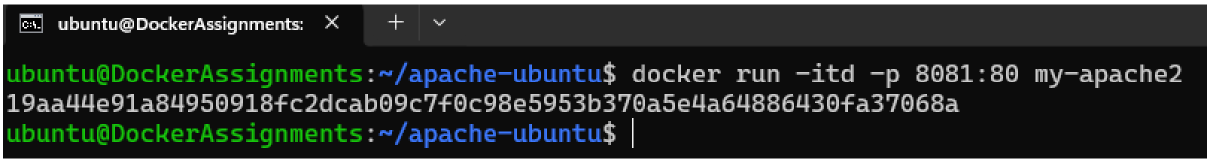
```
GNU nano 7.2                                                    Dockerfile *
# Use the official Ubuntu base image
FROM ubuntu:latest

# Install Apache2
RUN apt-get update && apt-get install -y apache2

# Copy the sample HTML File to the Apache default directory
COPY index.html /var/www/html/index.html

# Ensure Apache runs in the foreground
CMD ["apache2ctl", "-D", "FOREGROUND"]

# Expose port 80
EXPOSE 80
```

Build Again docker build -t my-apache2 .

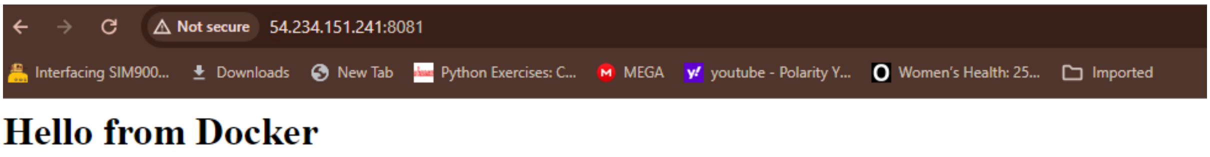


```
ubuntu@DockerAssignments:~/apache-ubuntu$ nano index.html
ubuntu@DockerAssignments:~/apache-ubuntu$ nano Dockerfile
ubuntu@DockerAssignments:~/apache-ubuntu$ docker build -t my-apache2 .
[+] Building 0.3s (8/8) FINISHED                                                                docker:default
 => [internal] load build definition from Dockerfile                                                      0.0s
 => => transferring dockerfile: 372B                                                                      0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                          0.0s
 => [internal] load .dockerignore                                                                         0.0s
 => => transferring context: 2B                                                                           0.0s
 => [1/3] FROM docker.io/library/ubuntu:latest                                                            0.0s
 => CACHED [2/3] RUN apt-get update && apt-get install -y apache2                                         0.0s
 => [internal] load build context                                                                         0.0s
 => => transferring context: 89B                                                                          0.0s
 => [3/3] COPY index.html /var/www/html/index.html                                                        0.0s
 => exporting to image                                                                                    0.1s
 => => exporting layers                                                                                   0.0s
 => => writing image sha256:fea5f6f4dd0e8490bc89e3962a06a0ce7f6cc4f1040d3e1de6cdc495b2ed96eb              0.0s
 => => naming to docker.io/library/my-apache2                                                             0.0s
ubuntu@DockerAssignments:~/apache-ubuntu$
```

Now its port to 8081:80



```
ubuntu@DockerAssignments:~/apache-ubuntu$ docker run -itd -p 8081:80 my-apache2
19aa44e91a84950918fc2dcab09c7f0c98e5953b370a5e4a64886430fa37068a
ubuntu@DockerAssignments:~/apache-ubuntu$
```



Hello from Docker

Dockerfile:

```
FROM ubuntu:latest
RUN apt-get update && apt-get install -y apache2
COPY index.html /var/www/html/index.html
CMD ["apache2ctl", "-D", "FOREGROUND"]
EXPOSE 80
```

Index.html:

```
<html><body><h1>Hello from Docker!</h1></body></html>
```