DOCKER ASSIGNMENT SUBMISSION

Name: Vikram

Assignment: Upload Custom Docker Image to Docker Hub & Run on Separate Machine

Problem Statement

Using the Docker image created in the previous assignment:

1. Upload the custom image to Docker Hub
2. Pull the uploaded image from a separate EC2 machine
3. Run the container on port 80
4. Start Apache2 inside the container
5. Verify the Apache webpage loads successfully

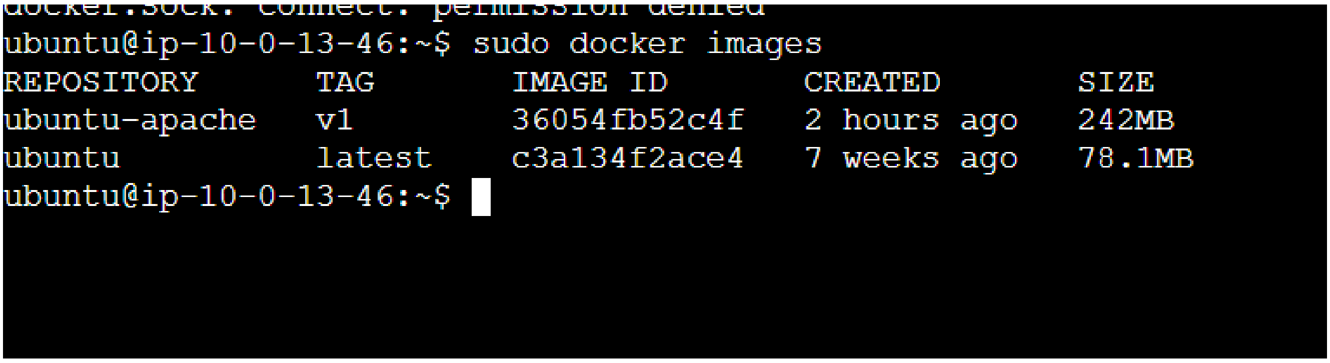
Task 1: Use the Saved Image From Previous Assignment

From Assignment 2, the saved image is:

ubuntu-apache:v1

Verify:

```
sudo docker images
```



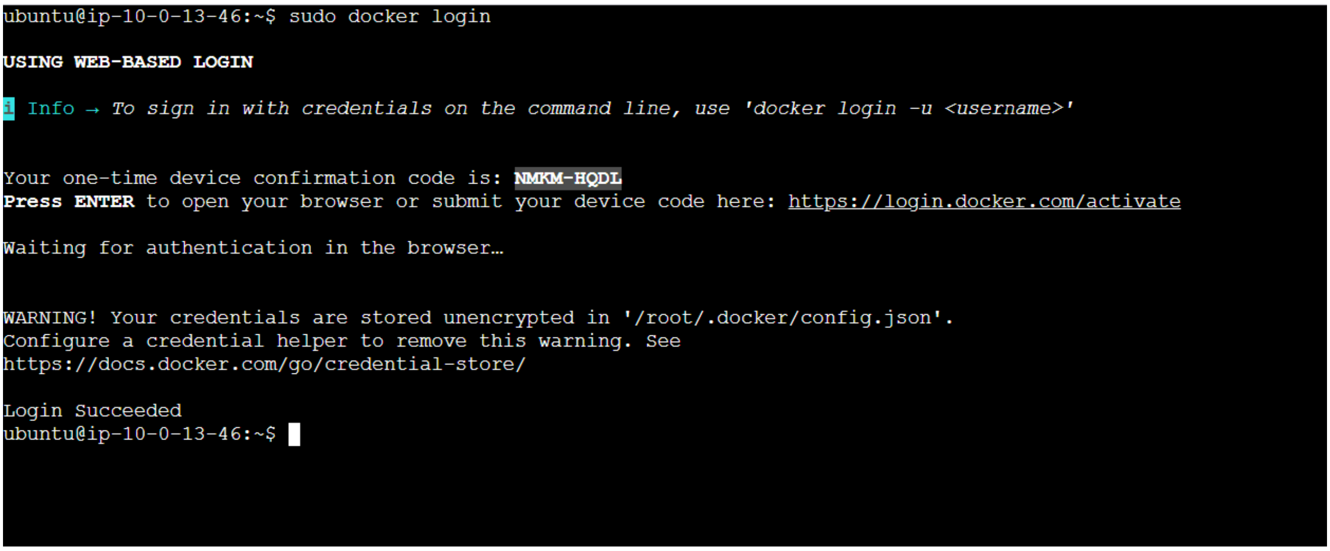
```
docker.sock: connect: permission denied
ubuntu@ip-10-0-13-46:~$ sudo docker images
REPOSITORY       TAG       IMAGE ID       CREATED         SIZE
ubuntu-apache    v1        36054fb52c4f   2 hours ago     242MB
ubuntu           latest    c3a134f2ace4   7 weeks ago     78.1MB
ubuntu@ip-10-0-13-46:~$
```

Task 2: Upload This Image to Docker Hub

1. Login to Docker Hub

```
sudo docker login
```

Enter your Docker Hub username and password.



```
ubuntu@ip-10-0-13-46:~$ sudo docker login

USING WEB-BASED LOGIN

i Info → To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: NMKM-HQDL
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser…


WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
ubuntu@ip-10-0-13-46:~$
```

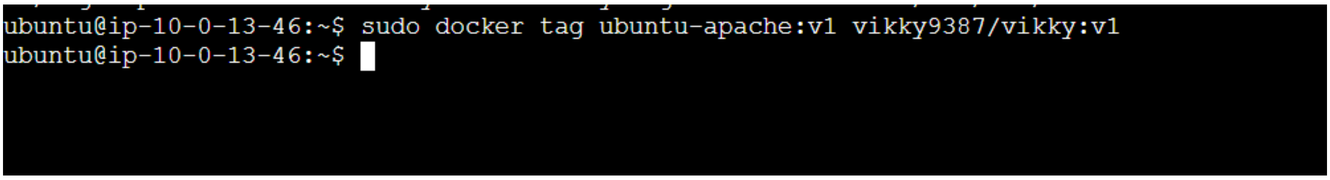
2. Tag the Image for Docker Hub

Format:

sudo docker tag <local-image> <dockerhub-username>/<repo-name>:tag

Example:

sudo docker tag ubuntu-apache:v1 vikky9387/vikky:v1

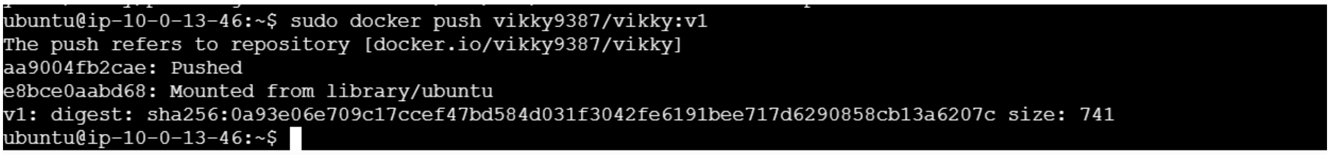


```
ubuntu@ip-10-0-13-46:~$ sudo docker tag ubuntu-apache:v1 vikky9387/vikky:v1
ubuntu@ip-10-0-13-46:~$
```

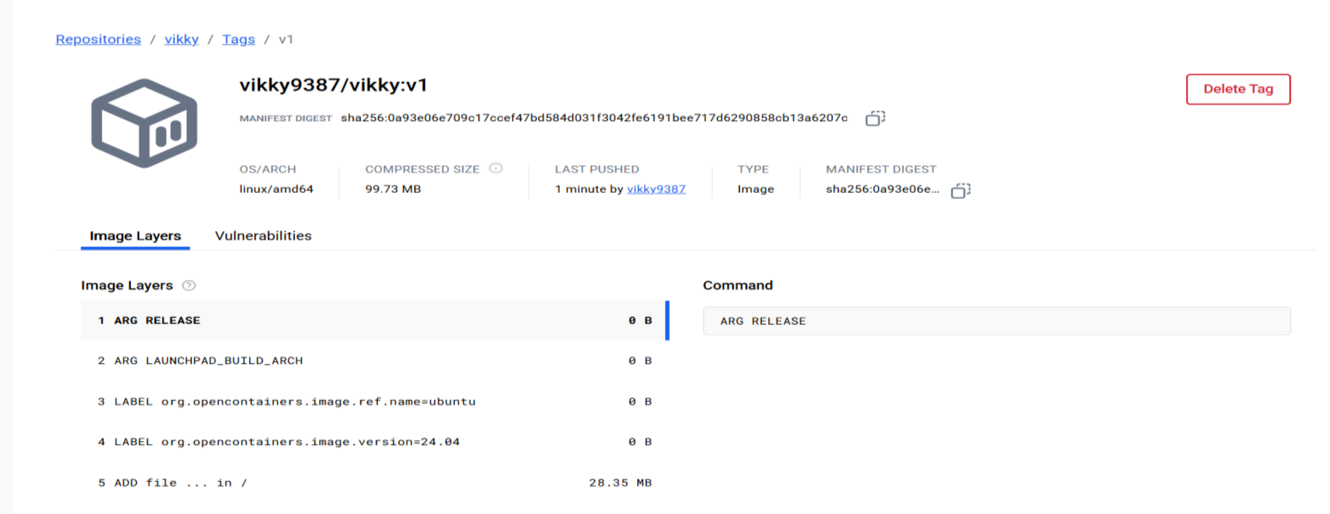
3. Push the Image to Docker Hub

sudo docker push vikky9387/vikky:v1

Your image is now available publicly on Docker Hub.



```
ubuntu@ip-10-0-13-46:~$ sudo docker push vikky9387/vikky:v1
The push refers to repository [docker.io/vikky9387/vikky]
aa9004fb2cae: Pushed
e8bce0aabd68: Mounted from library/ubuntu
v1: digest: sha256:0a93e06e709c17ccef47bd584d031f3042fe6191bee717d6290858cb13a6207c size: 741
ubuntu@ip-10-0-13-46:~$
```



Repositories / vikky / Tags / v1

vikky9387/vikky:v1

MANIFEST DIGEST sha256:0a93e06e709c17ccef47bd584d031f3042fe6191bee717d6290858cb13a6207c

OS/ARCH	COMPRESSED SIZE	LAST PUSHED	TYPE	MANIFEST DIGEST
linux/amd64	99.73 MB	1 minute by vikky9387	Image	sha256:0a93e06e...

Image Layers

```
1 ARG RELEASE                                          0 B
2 ARG LAUNCHPAD_BUILD_ARCH                             0 B
3 LABEL org.opencontainers.image.ref.name=ubuntu       0 B
4 LABEL org.opencontainers.image.version=24.04         0 B
5 ADD file ... in /                                28.35 MB
```

Command

```
ARG RELEASE
```

Task 3: Pull the Image on a Separate EC2 Machine

Launch a second Ubuntu EC2 instance → connect via EC2 Instance Connect.

Install Docker :

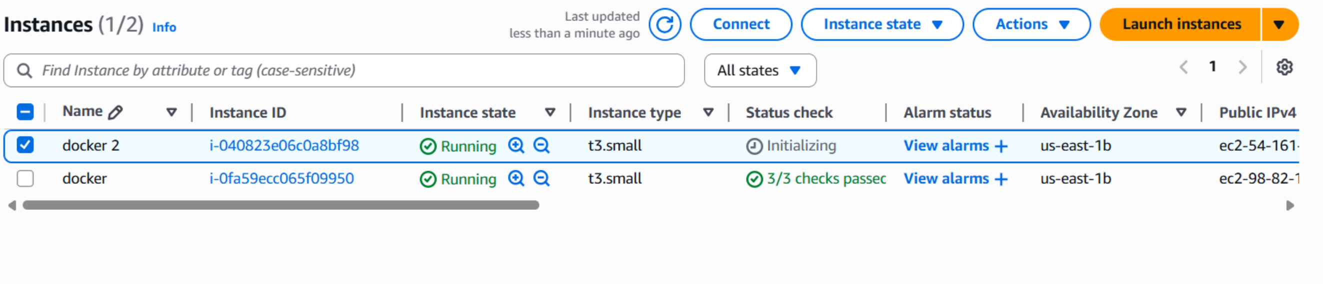
sudo apt update -y

sudo apt install docker.io -y

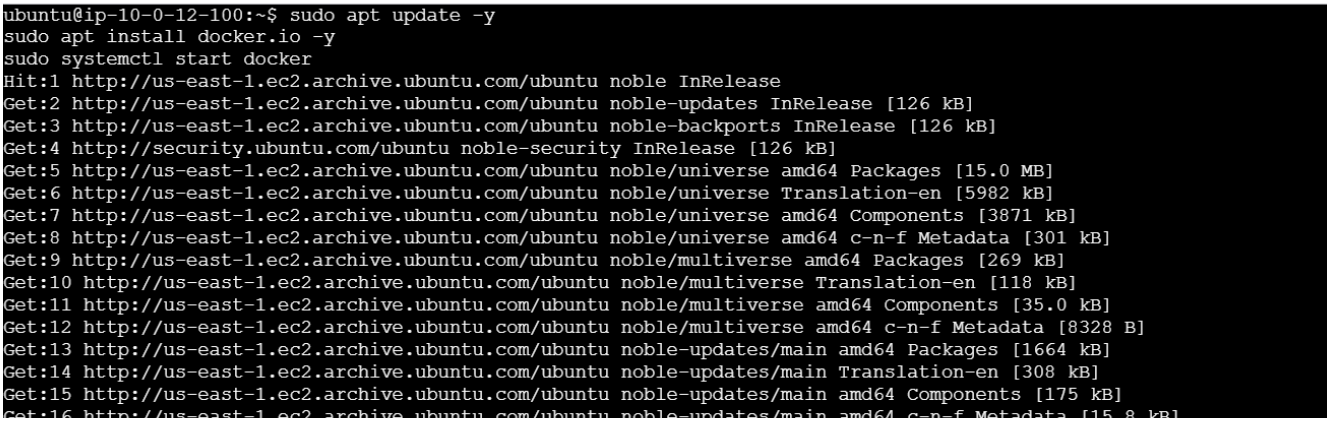
sudo systemctl start docker

Now pull the image:

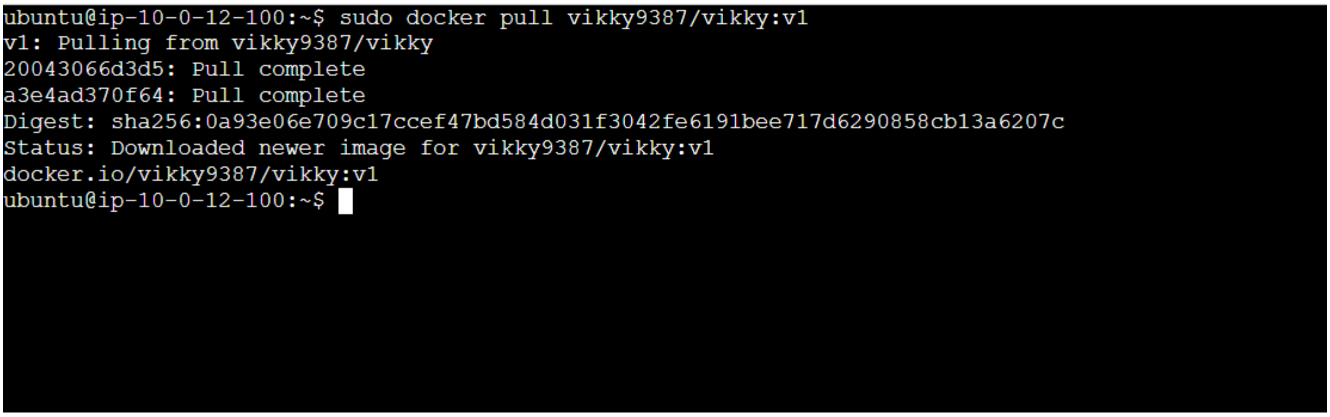
sudo docker pull vikky9387/vikky:v1



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑	docker 2	i-040823e06c0a8bf98	✓ Running	t3.small	Initializing	View alarms +	us-east-1b	ec2-54-161-
☐	docker	i-0fa59ecc065f09950	✓ Running	t3.small	✓ 3/3 checks passec	View alarms +	us-east-1b	ec2-98-82-1



```
ubuntu@ip-10-0-12-100:~$ sudo apt update -y
sudo apt install docker.io -y
sudo systemctl start docker
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1664 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [308 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.8 kB]
```



```
ubuntu@ip-10-0-12-100:~$ sudo docker pull vikky9387/vikky:v1
v1: Pulling from vikky9387/vikky
20043066d3d5: Pull complete
a3e4ad370f64: Pull complete
Digest: sha256:0a93e06e709c17ccef47bd584d031f3042fe6191bee717d6290858cb13a6207c
Status: Downloaded newer image for vikky9387/vikky:v1
docker.io/vikky9387/vikky:v1
ubuntu@ip-10-0-12-100:~$ 
```

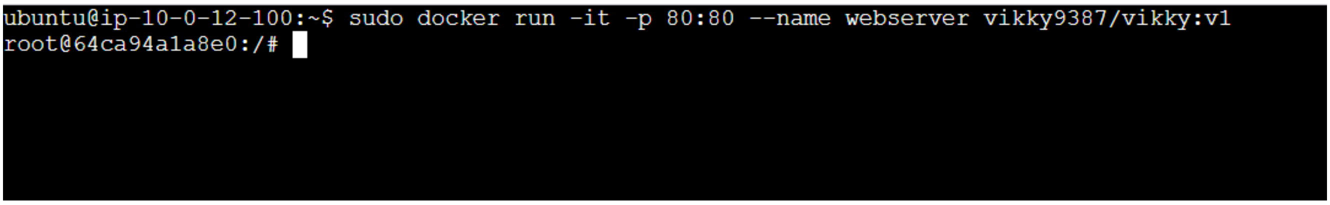
Task 4: Launch Container on Port 80 and Start Apache

Run the container:

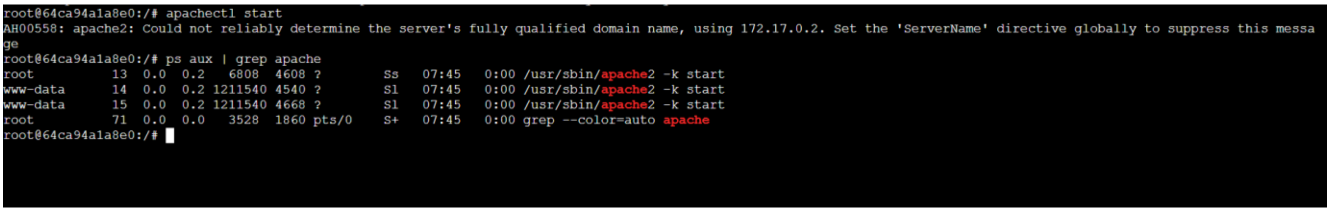
sudo docker run -it -p 80:80 --name webserver vikky9387/vikky:v1

Now inside the container, start Apache:

apachectl start



```
ubuntu@ip-10-0-12-100:~$ sudo docker run -it -p 80:80 --name webserver vikky9387/vikky:v1
root@64ca94a1a8e0:/#
```



```
root@64ca94a1a8e0:/# apachectl start
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this messa
ge
root@64ca94a1a8e0:/# ps aux | grep apache
root          13  0.0  0.2   6808  4608 ?        Ss   07:45   0:00 /usr/sbin/apache2 -k start
www-data      14  0.0  0.2 1211540 4540 ?        Sl   07:45   0:00 /usr/sbin/apache2 -k start
www-data      15  0.0  0.2 1211540 4668 ?        Sl   07:45   0:00 /usr/sbin/apache2 -k start
root          71  0.0  0.0   3528  1860 pts/0    S+   07:45   0:00 grep --color=auto apache
root@64ca94a1a8e0:/#
```

Task 5: Verify Apache2 Service in the Browser

Open:

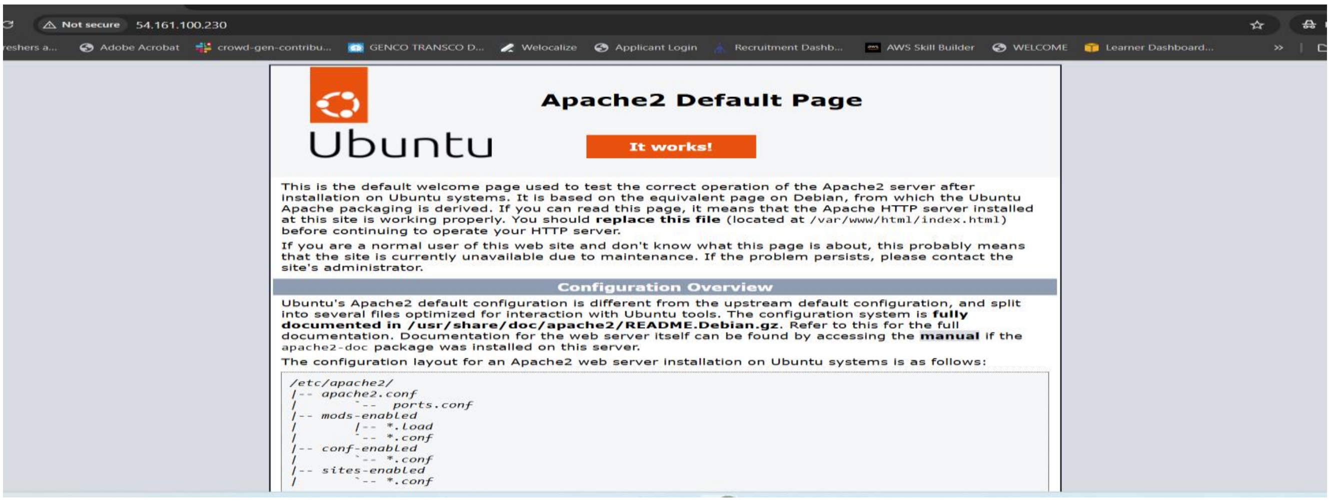
http://<SECOND-EC2-PUBLIC-IP>

You should see:

Apache2 Ubuntu Default Page

This confirms successful:

- Image upload
- Image pull
- Container launch
- Apache startup
- Web access



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

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

Conclusion

Successfully completed the assignment by:

- Using the custom image created earlier
- Uploading the image to Docker Hub
- Pulling the image on a separate EC2 machine
- Running the container and starting Apache
- Verifying Apache through the browser

This demonstrates knowledge of Docker Hub usage, image distribution, and deploying containers across different systems.