

DOCKER ASSIGNMENT SUBMISSION

Name: Vikram

Assignment: Create Dockerfile to Install & Auto-Start Apache2

Problem Statement

Create a Dockerfile that:

1. Uses Ubuntu as the base image
 2. Installs Apache2 inside the container
 3. Automatically starts Apache2 when the container launches
-

Tasks Performed

STEP 1 — Create a Directory for Your Dockerfile

`mkdir apache-docker`

`cd apache-docker`

```
ubuntu@ip-10-0-13-46:~$ mkdir apache-docker
cd apache-docker
ubuntu@ip-10-0-13-46:~/apache-docker$
```

STEP 2 — Create the Dockerfile

Run:

`nano Dockerfile`

Paste this inside:

`# Use Ubuntu as base image`

`FROM ubuntu:latest`

`# Install Apache2`

`RUN apt update -y && \`

`apt install apache2 -y && \`

`apt clean`

```
# Expose port 80
```

```
EXPOSE 80
```

```
# Start Apache automatically when container starts
```

```
CMD ["apachectl", "-D", "FOREGROUND"]
```

```
ubuntu@ip-10-0-13-46:~/apache-docker$ nano Dockerfile
ubuntu@ip-10-0-13-46:~/apache-docker$ cat Dockerfile
# Use Ubuntu as base image
FROM ubuntu:latest

# Install Apache2
RUN apt update -y && \
    apt install apache2 -y && \
    apt clean

# Expose port 80
EXPOSE 80

# Start Apache automatically when container starts
CMD ["apachectl", "-D", "FOREGROUND"]
ubuntu@ip-10-0-13-46:~/apache-docker$
```

STEP 3 — Build the Docker Image

Run:

```
sudo docker build -t ubuntu-apache-auto:v1
```

You should see:

Ubuntu downloading

Apache installing

Image built successfully

Check image:

```
sudo docker images
```

```

ubuntu@ip-10-0-13-46:~/apache-docker$ sudo docker build -t ubuntu-apache-auto:v1 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  2.048kB
Step 1/4 : FROM ubuntu:latest
--> c3a134f2ace4
Step 2/4 : RUN apt update -y &&      apt install apache2 -y &&      apt clean
--> Running in 9bee2f05a002

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:9 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [2105 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [35.9 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [3026 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1943 kB]

```

```

ubuntu@ip-10-0-13-46:~/apache-docker$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu-apache-auto  v1                 c5f8a23f0f2f       38 seconds ago     242MB
ubuntu-apache       v1                 36054fb52c4f       3 hours ago        242MB
vikky9387/vikky     v1                 36054fb52c4f       3 hours ago        242MB
ubuntu              latest             c3a134f2ace4       7 weeks ago        78.1MB
ubuntu@ip-10-0-13-46:~/apache-docker$

```

STEP 4 — Run Container From This Image

Run:

```
sudo docker run -p 80:80 --name auto-apache ubuntu-apache-auto:v1
```

This time Apache automatically starts because of CMD.

You will NOT need to run:

```
apachectl start
```

Docker will do it.

```

ubuntu@ip-10-0-13-46:~/apache-docker$ sudo docker run -p 80:80 --name auto-apache ubuntu-apache-auto:v1
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

```

STEP 5 — Test in Browser

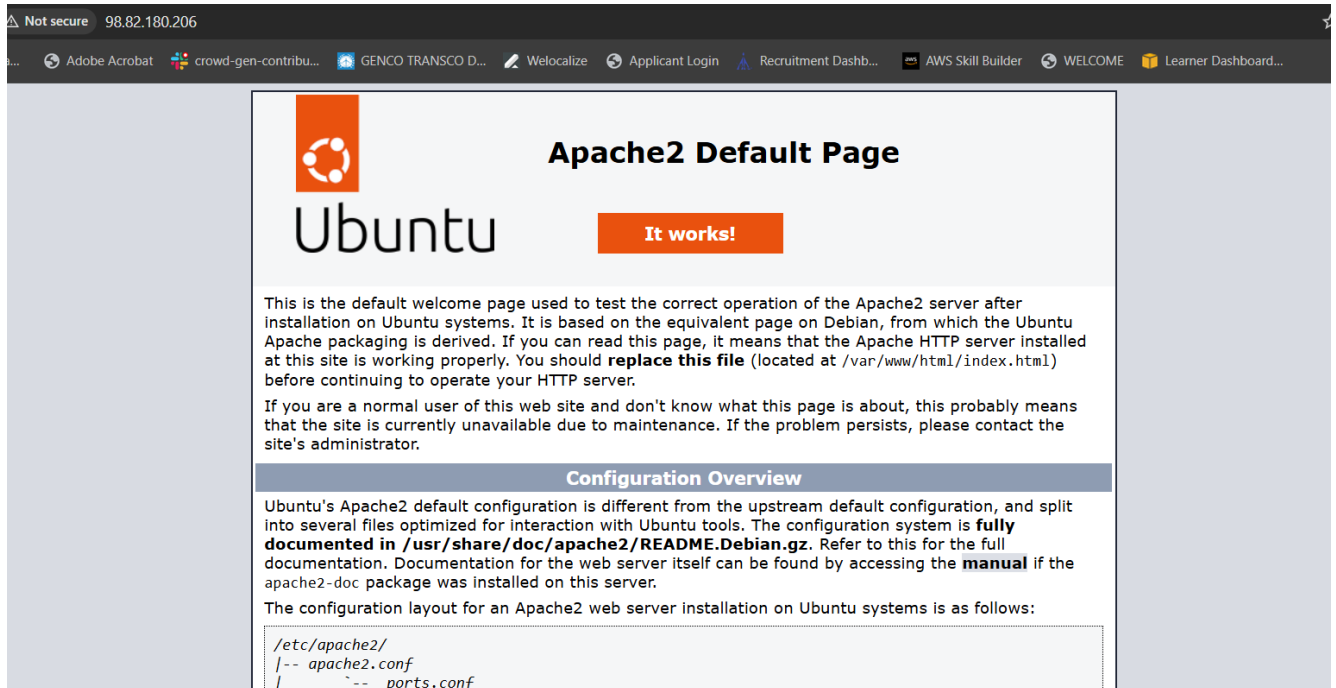
Open:

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

You MUST see:

Apache2 Ubuntu Default Page

If yes → your Dockerfile works perfectly.



Conclusion

Successfully created a Dockerfile that:

- Uses Ubuntu base image
- Installs Apache2
- Exposes port 80
- Automatically starts Apache on container launch

The Dockerfile fulfills all assignment requirements and demonstrates Docker image creation, customization, and container execution.