

Assignment: 8

Name: Divya Shah Branch: IT/V Roll No.: 115

Date: 26/08/2023

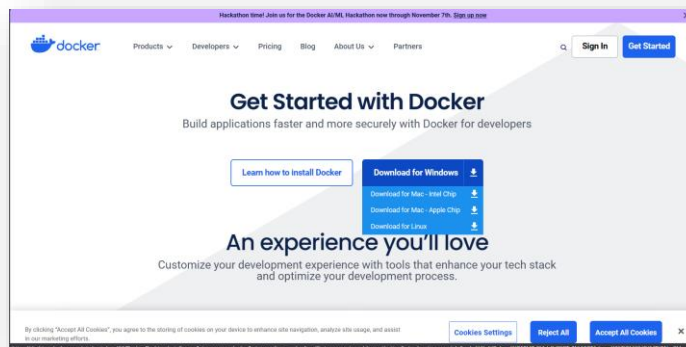
Aim: To build an image for sample web application using Dockerfile.

LO mapped: LO1, LO5

Theory:

The steps to build an image for sample web application using Dockerfile are as follows:

1. **Install Docker:** Ensure that Docker is installed on your system. You can download and install it from the [Docker website](https://www.docker.com/).



2. **Create a Project Directory:** Start by creating a directory for your project. Inside this directory, you will place your web application code and Dockerfile.



3. **Write a Dockerfile:** Create a Dockerfile in your project directory. This file contains instructions for building the Docker image. Here's a simple example of a Dockerfile for a Node.js web application:

```

root@ip-172-31-86-229 ec2-user:~# cd website
root@ip-172-31-86-229 ec2-user:~# cd website
root@ip-172-31-86-229 ec2-user:~# cd website/
root@ip-172-31-86-229 website:~# ls
root@ip-172-31-86-229 website:~# wget https://www.free-css.com/assets/files/free-css-templates/download/page273/spourmo.zip
--2021-12-09 16:10:44-- https://www.free-css.com/assets/files/free-css-templates/download/page273/spourmo.zip
Resolving www.free-css.com (www.free-css.com)... 217.160.0.242, 2001:698:100f:f000:28f
Connecting to www.free-css.com (www.free-css.com)[217.160.0.242]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1179909 (1.1M) [application/zip]
Saving to: 'spourmo.zip'

100%[=====] 1,179,909  1.61MB/s  in 0.7s

2021-12-09 16:10:45 (1.61 MB/s) - 'spourmo.zip' saved [1179909/1179909]

root@ip-172-31-86-229 website:~#

```

4. **Place Application Code:** Place your web application code, including HTML, JavaScript, CSS, and any other assets, in the same project directory.

```

root@ip-172-31-86-229 website:~# ls
spourmo.zip
root@ip-172-31-86-229 website:~# unzip spourmo.zip

```

5. **Build the Docker Image:** Open a terminal and navigate to your project directory. Run the following command to build the Docker image:

```

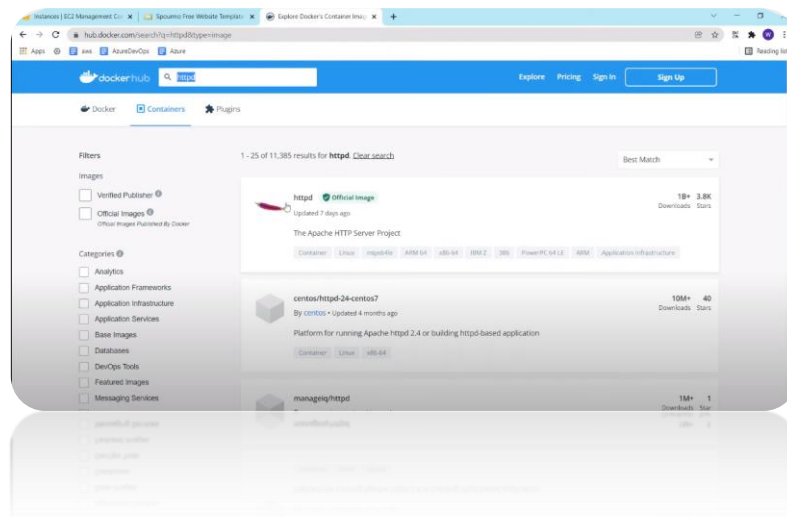
root@ip-172-31-86-229 website:~# cd spourmo/
root@ip-172-31-86-229 spourmo:~# ls
images index.html layout licence.txt pages
root@ip-172-31-86-229 spourmo:~#

```

```

[root@ip-172-31-86-128:/home/ec2-user]# cd spourmo/
[root@ip-172-31-86-229 spourmo]# ls
images index.html layout licence.txt pages
[root@ip-172-31-86-229 spourmo]# cp -R * ../
[root@ip-172-31-86-229 spourmo]# cd ..
[root@ip-172-31-86-229 website]# ls
images index.html layout licence.txt pages spourmo spourmo.zip
[root@ip-172-31-86-229 website]# rm -Rf spourmo.zip spourmo
[root@ip-172-31-86-229 website]# ls
images index.html layout licence.txt pages
[root@ip-172-31-86-229 website]#

```



```

[root@ip-172-31-86-128:/home/ec2-user]# cd spourmo/
[root@ip-172-31-86-229 spourmo]# ls
images index.html layout licence.txt pages
[root@ip-172-31-86-229 spourmo]# cp -R * ../
[root@ip-172-31-86-229 spourmo]# cd ..
[root@ip-172-31-86-229 website]# ls
images index.html layout licence.txt pages spourmo spourmo.zip
[root@ip-172-31-86-229 website]# rm -Rf spourmo.zip spourmo
[root@ip-172-31-86-229 website]# ls
images index.html layout licence.txt pages
[root@ip-172-31-86-229 website]# vim Dockerfile
[root@ip-172-31-86-229 website]# docker build -t website .
Sending build context to Docker daemon 2.8MB
Step 1/2 : FROM httpd
latest: Pulling from library/httpd
e5ae68f740261: Extracting [-----] 9.83MB/31.37MB
bc36ee1127ec: Download complete
d597d2b63171: Download complete
f1aa5f54b226: Download complete
aa379c0e0dc2: Download complete

```

```

root@ip-172-31-86-229 website| docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
website        latest    11e6bcb1b56    14 seconds ago 146MB
httpd          latest    ea28e1b82f31   7 days ago    143MB
hello-world    latest    feb5d9fea6a5   2 months ago   13.3kB
root@ip-172-31-86-229 website|

```

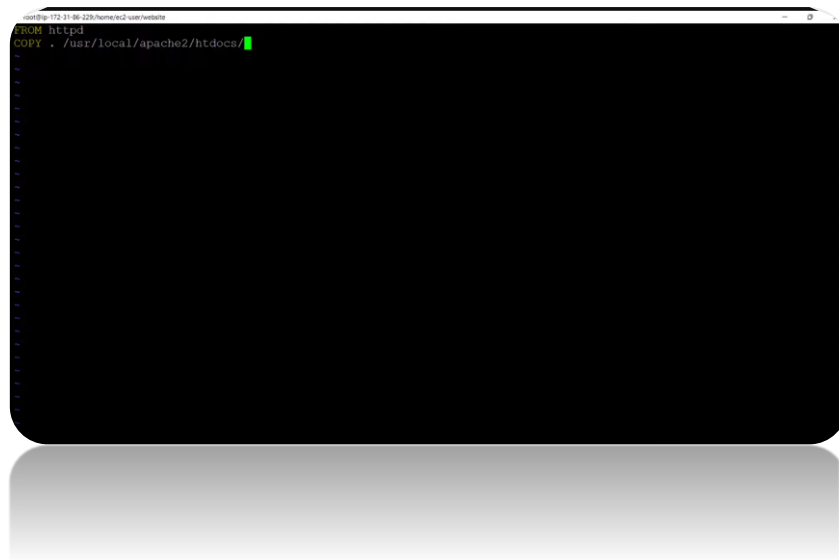
6. **Run a Container:** After successfully building the image, you can run a container based on that image with the following command:

```

root@ip-172-31-86-229 website| docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
website        latest    11e6bcb1b56    14 seconds ago 146MB
httpd          latest    ea28e1b82f31   7 days ago    143MB
hello-world    latest    feb5d9fea6a5   2 months ago   13.3kB
root@ip-172-31-86-229 website| docker run -itd -p 80:80 website website
10b56052b79f264bfd9116c69ea846cf2189bedc256c8e9759dcb6ee81dccc9b
docker: Error response from daemon: OCI runtime create failed: container_linux.go:380: starting container process caused: exec: "website": executable file not found in $PATH: unknown.
root@ip-172-31-86-229 website| docker run -itd -p 80:80 --name website website
a5a12ada297fed7aab0d4dad7bd6eb2f9811f3534b56567602916f9d25c69f8a
root@ip-172-31-86-229 website| docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED       STATUS       PORTS                               NAMES
a5a12ada297f   website   "httpd-foreground"      3 seconds ago Up 3 seconds 0.0.0.0:80->80/tcp, :::80->80/tcp   websa
te

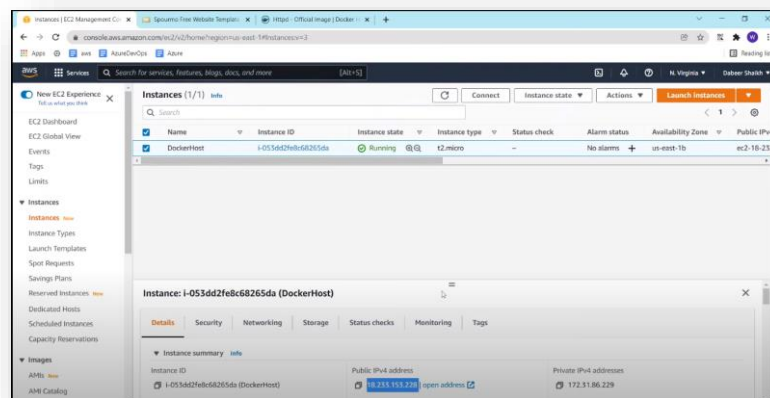
root@ip-172-31-86-229 website| cd spourmo/
root@ip-172-31-86-229 spourmo| ls
images index.html layout licence.txt pages
root@ip-172-31-86-229 spourmo| cp -R * ../
root@ip-172-31-86-229 spourmo| cd ..
root@ip-172-31-86-229 website| ls
images index.html layout licence.txt pages spourmo spourmo.zip
root@ip-172-31-86-229 website| rm -Rf spourmo.zip spourmo
root@ip-172-31-86-229 website| ls
images index.html layout licence.txt pages
root@ip-172-31-86-229 website| vim Dockerfile

```



7. **Access Your Web Application:** Open a web browser and navigate to <http://localhost:8080> to access your web application running in the Docker container.

8.



Your web application is now running inside a Docker container. You can further customize your Dockerfile and the application code as needed. Make sure to manage your containers and images appropriately, especially when dealing with production deployments.

Conclusion: We conclude by saying that we built a web app using DockerFile.