Department of Software Engineering

SE-211: Software Design and Architecture

Class: BESE 13AB

Lab 12: Docker

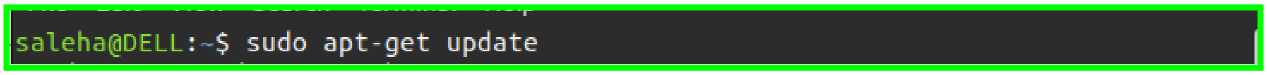
Date: May 6, 2024

Lab Instructor: Muhammad Danyal Sadiq

Name:	Saleha Zainab Fatima
CMS ID:	404329
Class:	SE-13B

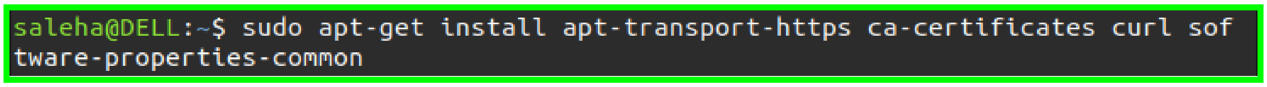
Installation of Docker Engine

Update Package Repository:



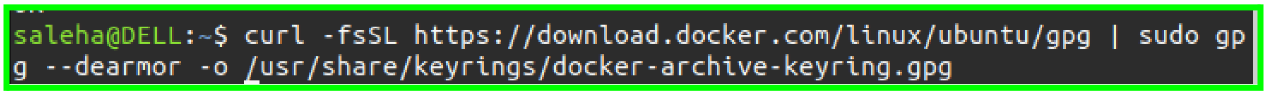
```
saleha@DELL:~$ sudo apt-get update
```

Install Dependencies:



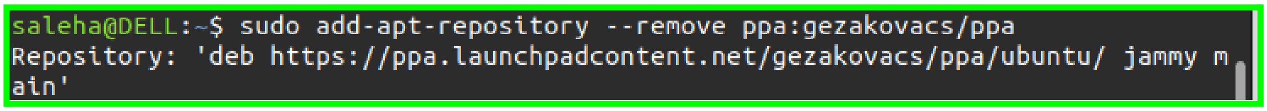
```
saleha@DELL:~$ sudo apt-get install apt-transport-https ca-certificates curl sof
tware-properties-common
```

Add Docker's Official GPG Key:



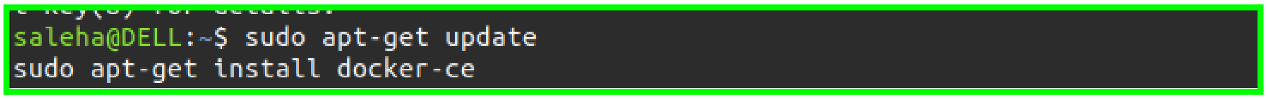
```
saleha@DELL:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gp
g --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Set Up Docker Repository:



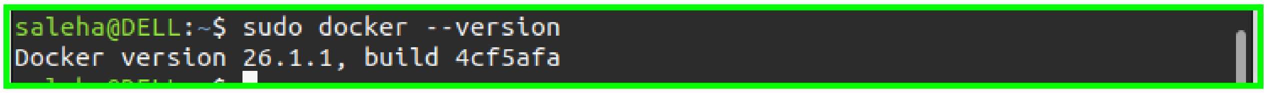
```
saleha@DELL:~$ sudo add-apt-repository --remove ppa:gezakovacs/ppa
Repository: 'deb https://ppa.launchpadcontent.net/gezakovacs/ppa/ubuntu/ jammy m
ain'
```

Install Docker Engine:



```
saleha@DELL:~$ sudo apt-get update
sudo apt-get install docker-ce
```

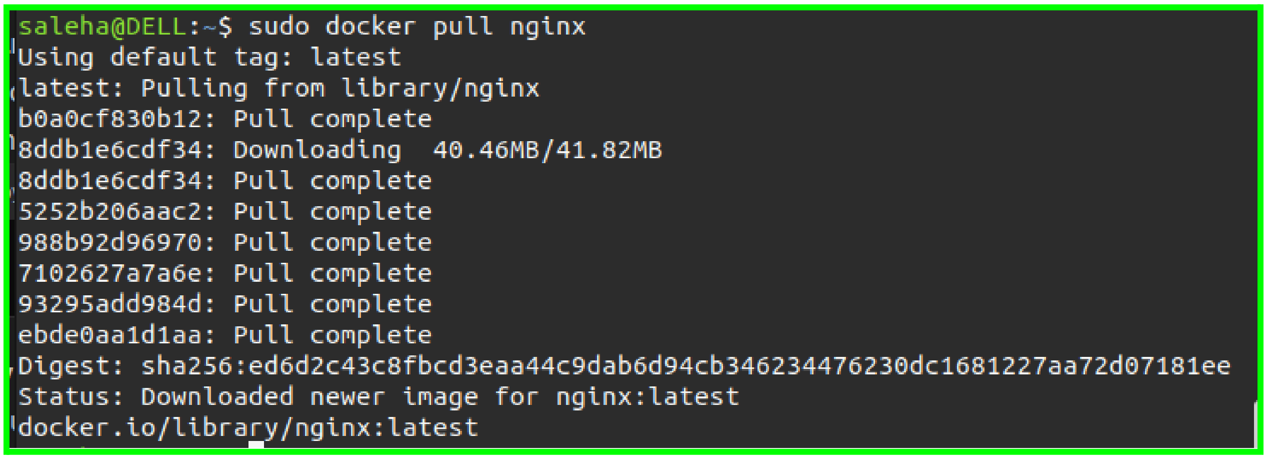
Verify Installation:



```
saleha@DELL:~$ sudo docker --version
Docker version 26.1.1, build 4cf5afa
```

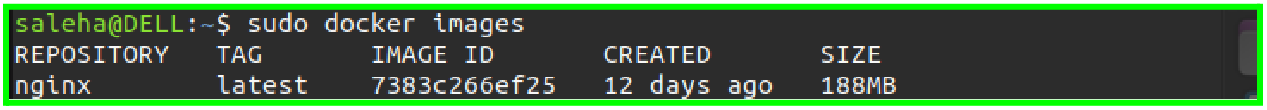
Task 1

1. Pull the Nginx image from the Docker registry:



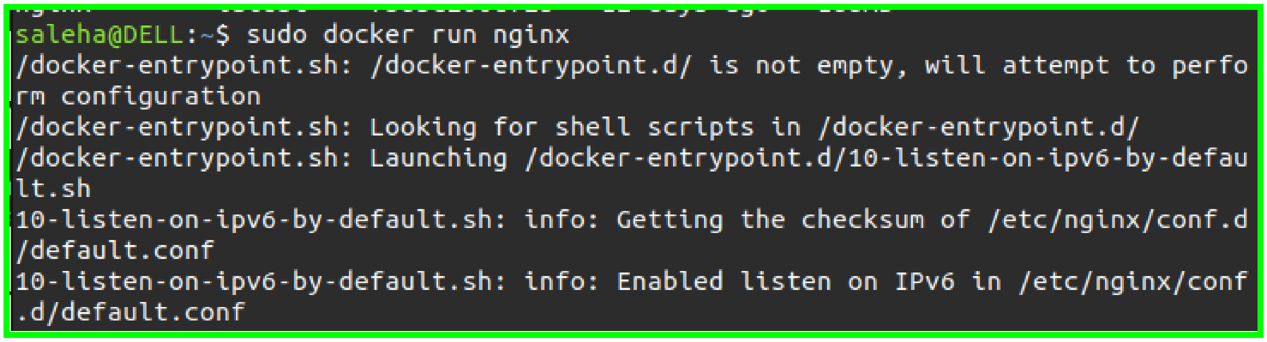
```
saleha@DELL:~$ sudo docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
b0a0cf830b12: Pull complete
8ddb1e6cdf34: Downloading  40.46MB/41.82MB
8ddb1e6cdf34: Pull complete
5252b206aac2: Pull complete
988b92d96970: Pull complete
7102627a7a6e: Pull complete
93295add984d: Pull complete
ebde0aa1d1aa: Pull complete
Digest: sha256:ed6d2c43c8fbcd3eaa44c9dab6d94cb346234476230dc1681227aa72d07181ee
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

2. Check the images available on the system:



```
saleha@DELL:~$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
nginx        latest    7383c266ef25   12 days ago   188MB
```

3. Create a container from the Nginx image:

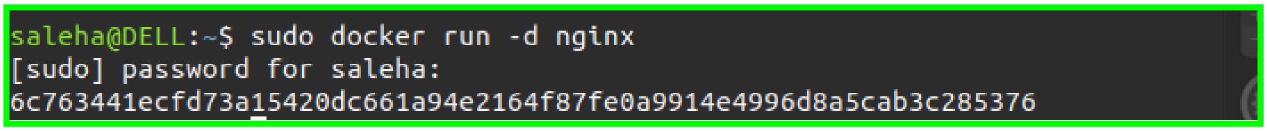


```
saleha@DELL:~$ sudo docker run nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perfo
rm configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-defau
lt.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d
/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf
.d/default.conf
```

4. Output of creating the container:

The Docker container running the Nginx web server is running successfully. The output shows that Nginx has started, and it's displaying some log messages indicating that it's ready to serve requests.

5. Create the container using -d tag:

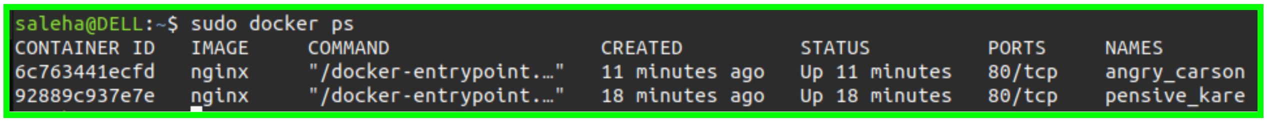


```
saleha@DELL:~$ sudo docker run -d nginx
[sudo] password for saleha:
6c763441ecfd73a15420dc661a94e2164f87fe0a9914e4996d8a5cab3c285376
```

What is returned in the output?

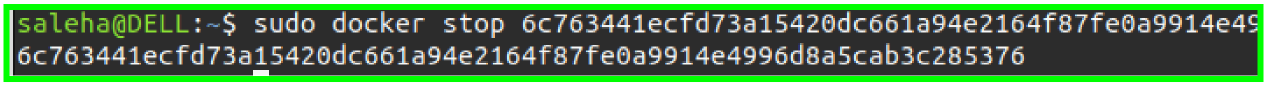
The output returned the container-id of the newly created container.

6. Check the containers on the system:



```
saleha@DELL:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
6c763441ecfd   nginx     "/docker-entrypoint.…"   11 minutes ago   Up 11 minutes   80/tcp    angry_carson
92889c937e7e   nginx     "/docker-entrypoint.…"   18 minutes ago   Up 18 minutes   80/tcp    pensive_kare
```

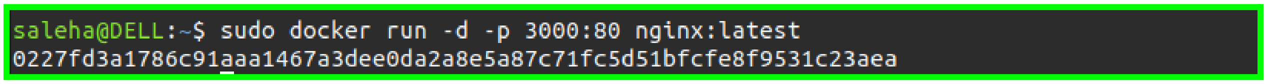
7. Stop the container:



```
saleha@DELL:~$ sudo docker stop 6c763441ecfd73a15420dc661a94e2164f87fe0a9914e49
6c763441ecfd73a15420dc661a94e2164f87fe0a9914e4996d8a5cab3c285376
```

After executing the **docker stop** command, Docker provides the container ID of the stopped container as confirmation that the container has been successfully stopped.

8. Run the Nginx container on port 3000:



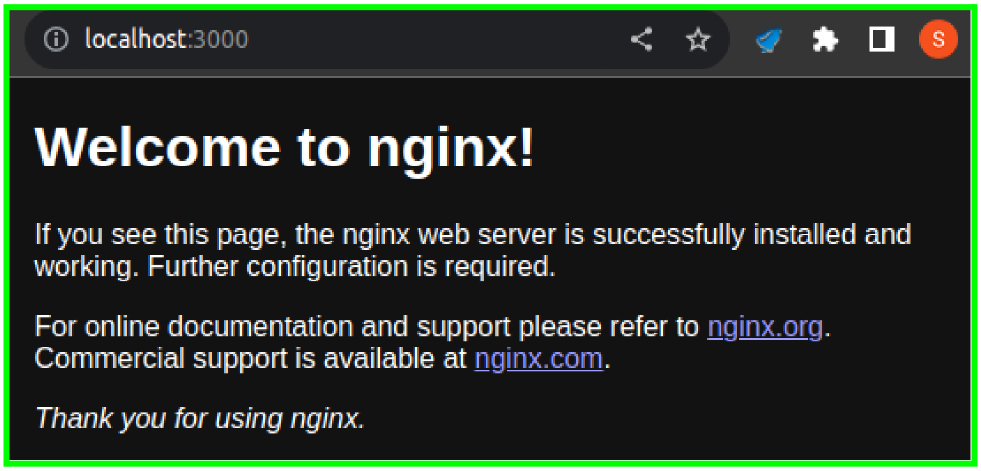
```
saleha@DELL:~$ sudo docker run -d -p 3000:80 nginx:latest
0227fd3a1786c91aaa1467a3dee0da2a8e5a87c71fc5d51bfcfe8f9531c23aea
```

The container ID provided confirms that the container has been successfully created.

Explanation of -d tag:

'**d**' implies that the container runs in **detached mode**, i.e. in the background.

9. Open the browser and go to localhost:3000.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Nginx welcome page is displayed after accessing localhost.

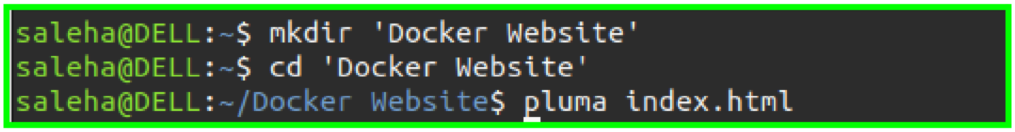
10. Explain what is happening by running the Docker container:

Running the Docker container makes the Nginx web server available on port 3000 of my local machine, I can access it through localhost on any web browser as displayed above.

Task 2

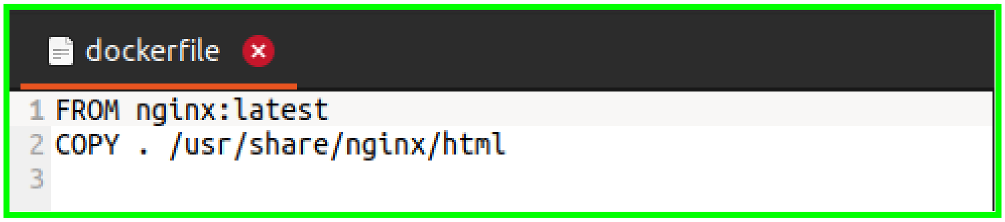
Creating a directory to store webpage and dockerfile

“Docker Website” is the directory, "index.html” is the simple webpage.



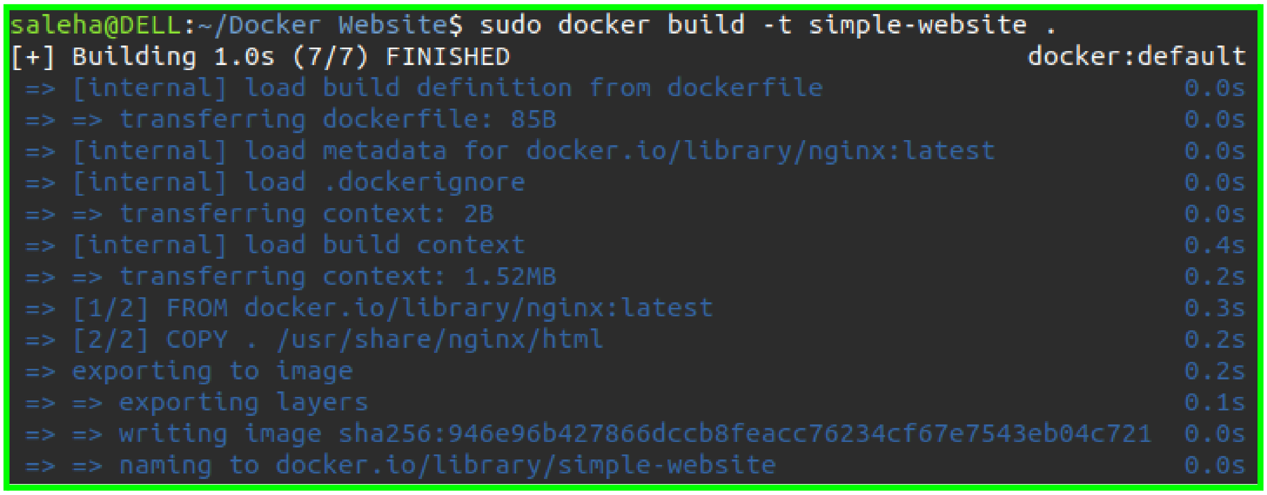
```
saleha@DELL:~$ mkdir 'Docker Website'
saleha@DELL:~$ cd 'Docker Website'
saleha@DELL:~/Docker Website$ pluma index.html
```

Contents of dockerfile:



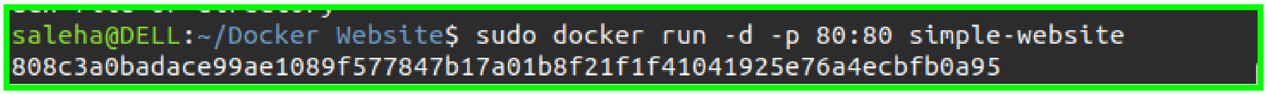
```
FROM nginx:latest
COPY . /usr/share/nginx/html

```

Build the Docker Image:



```
saleha@DELL:~/Docker Website$ sudo docker build -t simple-website .
[+] Building 1.0s (7/7) FINISHED                                 docker:default
 => [internal] load build definition from dockerfile                      0.0s
 => => transferring dockerfile: 85B                                       0.0s
 => [internal] load metadata for docker.io/library/nginx:latest           0.0s
 => [internal] load .dockerignore                                         0.0s
 => => transferring context: 2B                                           0.0s
 => [internal] load build context                                         0.4s
 => => transferring context: 1.52MB                                       0.2s
 => [1/2] FROM docker.io/library/nginx:latest                             0.3s
 => [2/2] COPY . /usr/share/nginx/html                                    0.2s
 => exporting to image                                                    0.2s
 => => exporting layers                                                   0.1s
 => => writing image sha256:946e96b427866dccb8feacc76234cf67e7543eb04c721 0.0s
 => => naming to docker.io/library/simple-website                         0.0s
```

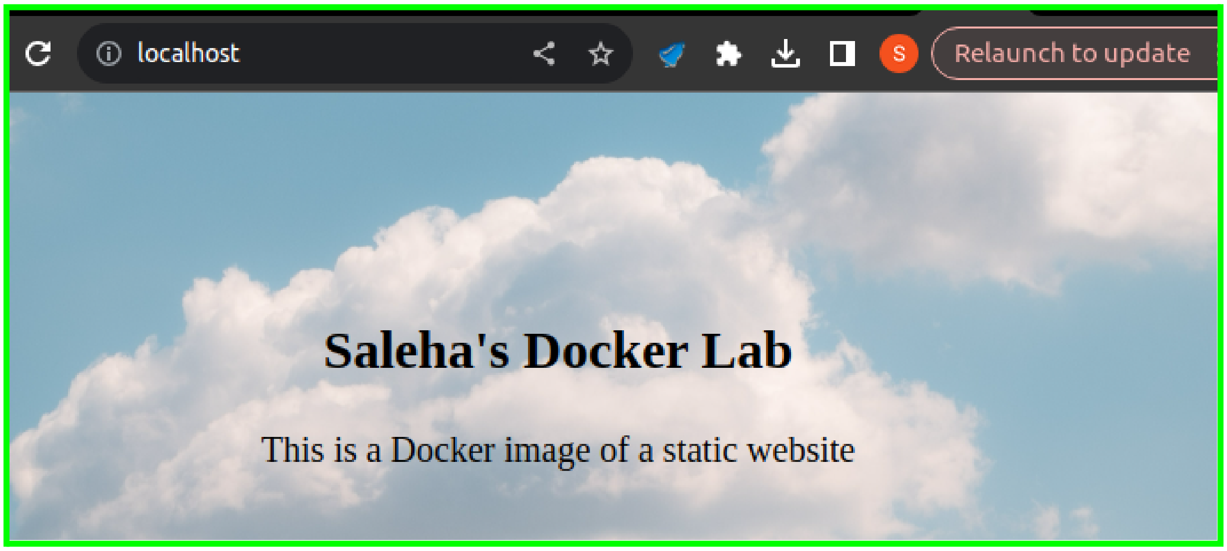
Run the Docker Container:



```
saleha@DELL:~/Docker Website$ sudo docker run -d -p 80:80 simple-website
808c3a0badace99ae1089f577847b17a01b8f21f1f41041925e76a4ecbfb0a95
```

The container ID provided confirms that the container has been successfully created.

Accessing the webpage using localhost:



Saleha's Docker Lab

This is a Docker image of a static website

Github Repository Link:

https://github.com/saleha-zf/simple-docker-webpage