Academic Year: 2025-26 **Semester: V Class /**
Branch: TE IT
Subject: DevOPs Lab (DL)
Subject Lab In-charge: Prof. Seema Jadhav

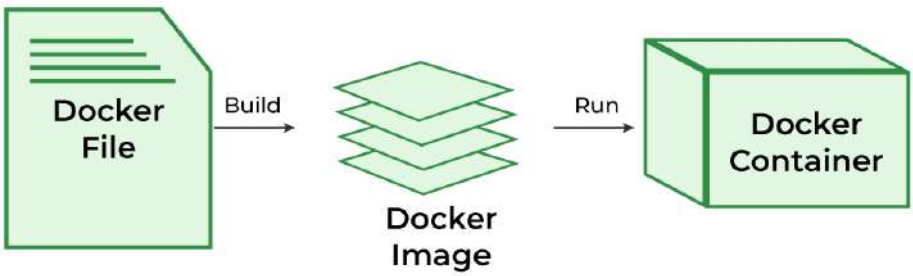
EXPERIMENT NO. 09

Aim: To build an image for a sample web application from a docker file using various docker file instructions

Theory: The Dockerfile uses DSL (Domain Specific Language) and contains instructions for generating a Docker image. Dockerfile will define the processes to quickly produce an image. While creating your application, you should create a Dockerfile in order since the Docker daemon runs all of the instructions from top to bottom.

An artifact with several layers and a lightweight, compact stand-alone executable package that contains all of the components required to run a piece of software, including the code, a runtime, libraries, environment variables, and configuration files is called a Docker image.

A container is a runtime instance of an image. Containers make development and deployment more efficient since they contain all the dependencies and parameters needed for the application it runs completely isolated from the host environment.



Dockerfile commands/Instructions

1. FROM

- Represents the base image(OS), which is the command that is executed first before any other commands.

Syntax

FROM <ImageName>

2. COPY

- *The copy command is used to copy the file/folders to the image while building the image.*

Syntax:

COPY <Source> <Destination>

3] RUN

- Scripts and commands are run with the RUN instruction. The execution of RUN commands or instructions will take place while you create an image on top of the prior layers (Image).

Syntax

RUN < Command + ARGS>

4] CMD

- *The main purpose of the CMD command is to start the process inside the container and it can be overridden.*

Syntax

CMD [command + args]

Stages of Creating Docker Image from Dockerfile
The following are the stages of creating docker image form Dockerfile:
1. Create a file named Dockerfile.
2. Add instructions in Dockerfile.
3. Build Dockerfile to create an image.
4. Run the image to create a container.

IMPLEMENTATION:

PART I: Containerize an application using docker CLI Commands:

Let's create an nginx webserver, it is a web server platform which helps to host your web applications.

STEP1: Download nginx official image and then containerized your web application in it.

#docker images

```
devasc@labvm:~/Desktop/DOCKER_LAB$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
mysql         latest    a82a8f162e18   5 weeks ago      586MB
hello-world   latest    d2c94e258dcb   16 months ago    13.3kB
```

#docker pull nginx

```
devasc@labvm:~/Desktop/DOCKER_LAB$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
e4fff0779e6d: Pull complete
2a0cb278fd9f: Pull complete
7045d6c32ae2: Pull complete
03de31afb035: Pull complete
0f17be8dcff2: Pull complete
14b7e5e8f394: Pull complete
23fa5a7b99a6: Pull complete
Digest: sha256:447a8665cc1dab95b1ca778e162215839ccbb9189104c79d7ec3a81e145
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

```
devasc@labvm:~/Desktop/DOCKER_LAB$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
nginx         latest    5ef79149e0ec   2 weeks ago      188MB
mysql         latest    a82a8f162e18   5 weeks ago      586MB
hello-world   latest    d2c94e258dcb   16 months ago    13.3kB
```

STEP2: Run the container from nginx image

root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run --name webserver1 5ef

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run --name webserver1 5ef
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform conf
iguration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/defaul
t.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/defa
ult.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/08/31 18:32:00 [notice] 1#1: using the "epoll" event method
2024/08/31 18:32:00 [notice] 1#1: nginx/1.27.1
2024/08/31 18:32:00 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/08/31 18:32:00 [notice] 1#1: OS: Linux 5.4.0-37-generic
2024/08/31 18:32:00 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/08/31 18:32:00 [notice] 1#1: start worker processes
2024/08/31 18:32:00 [notice] 1#1: start worker process 29
2024/08/31 18:32:00 [notice] 1#1: start worker process 30
```

In another terminal

#docker ps -a

```
devasc@labvm:~/Desktop/DOCKER_LAB$ sudo su
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATU
S                  PORTS     NAMES
34816ce8fdab   5ef            "/docker-entrypoint.…"   About a minute ago   Up Ab
out a minute       80/tcp    webserver1
bc8a1f0b194f   hello-world    "/hello"                 6 days ago           Exite
d (0) 6 days ago             peaceful_aryabhata
fbd148039aee   hello-world    "/hello"                 6 days ago           Exite
d (0) 6 days ago             infallible_cerf
```

In previous terminal: ctrl+C ie; exit from container

In another terminal

#docker ps -a

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
                   PORTS     NAMES
34816ce8fdab   5ef            "/docker-entrypoint.…"   3 minutes ago    Exited (0)
 10 seconds ago               webserver1
bc8a1f0b194f   hello-world    "/hello"                 6 days ago       Exited (0)
 6 days ago                   peaceful_aryabhata
fbd148039aee   hello-world    "/hello"                 6 days ago       Exited (0)
 6 days ago                   infallible_cerf
```

Remove the container:

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker rm 348
348
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker ps -a
CONTAINER ID   IMAGE          COMMAND    CREATED       STATUS
TS      NAMES
bc8a1f0b194f   hello-world    "/hello"   6 days ago    Exited (0) 6 days ago
        peaceful_aryabhata
fbd148039aee   hello-world    "/hello"   6 days ago    Exited (0) 6 days ago
        infallible_cerf
```

In terminal 1:

root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run -it -p 3031:80 --name server1 nginx:latest bash

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run -it -p 3031:80 --name nse
rver1 nginx:latest bash
```

In Another Terminal:

#docker ps -a

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
               PORTS                                   NAMES
595c2fa10371   nginx:latest   "/docker-entrypoint.…"   8 minutes ago    Up 8 minu
tes            0.0.0.0:3031->80/tcp, :::3031->80/tcp   nserver1
bc8a1f0b194f   hello-world    "/hello"                 6 days ago       Exited (0
) 6 days ago                                           peaceful_aryabhata
fbd148039aee   hello-world    "/hello"                 6 days ago       Exited (0
) 6 days ago                                           infallible_cerf
```

Lets create a static website inside container. I need to go to the location where my index.html file is:

cd /usr/share/nginx/html/

```
root@595c2fa10371:/# cd /usr/share/nginx/html/
```

#ls

```
root@595c2fa10371:/usr/share/nginx/html# ls
50x.html  index.html
```

Rename the default index.html to index.html_backup

#mv index.html index.html_backup

```
root@595c2fa10371:/usr/share/nginx/html# mv index.html index.html_backup
```

#nano index.html

```
root@595c2fa10371:/usr/share/nginx/html# nano index.html
```

Nano not found: Because the container that I am running inside the shell says that nano application is not available inside the container. So first install nano: apt install nano

```
root@595c2fa10371:/usr/share/nginx/html# nano index.html
bash: nano: command not found
```

```
root@595c2fa10371:/usr/share/nginx/html# apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package nano is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'nano' has no installation candidate
```

```
root@595c2fa10371:/usr/share/nginx/html# apt update
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB
]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8787 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [13.8 kB
]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Package
s [179 kB]
Fetched 9234 kB in 1s (6581 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
10 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@595c2fa10371:/usr/share/nginx/html# apt install nano
Reading package lists... Done
```

```
root@595c2fa10371:/usr/share/nginx/html# nano index.html
```

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Login Page </title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-color: pink;
}
button {
       background-color: #4CAF50;
       width: 100%;
        color: orange;
        padding: 15px;
        margin: 10px 0px;
        border: none;
        cursor: pointer;
         }
 form {
        border: 3px solid #f1f1f1;
    }
 input[type=text], input[type=password] {
        width: 100%;
        margin: 8px 0;
        padding: 12px 20px;
        display: inline-block;
        border: 2px solid green;
        box-sizing: border-box;
    }
 button:hover {
        opacity: 0.7;
```

```
    }
  .cancelbtn {
        width: auto;
        padding: 10px 18px;
        margin: 10px 5px;
    }


 .container {
        padding: 25px;
        background-color: lightblue;
    }
</style>
</head>
<body>
    <center> <h1> <b>Student Login Form Designed by Sujata Oak</b> </h1> </center>
    <form>
        <div class="container">
            <label>Username : </label>
            <input type="text" placeholder="Enter Username" name="username" required>
            <label>Password : </label>
            <input type="password" placeholder="Enter Password" name="password" required>
            <button type="submit">Login</button>
            <input type="checkbox" checked="checked"> Remember me
            <button type="button" class="cancelbtn"> Cancel</button>
            Forgot <a href="#"> password? </a>
        </div>
    </form>
</body>
</html>
```

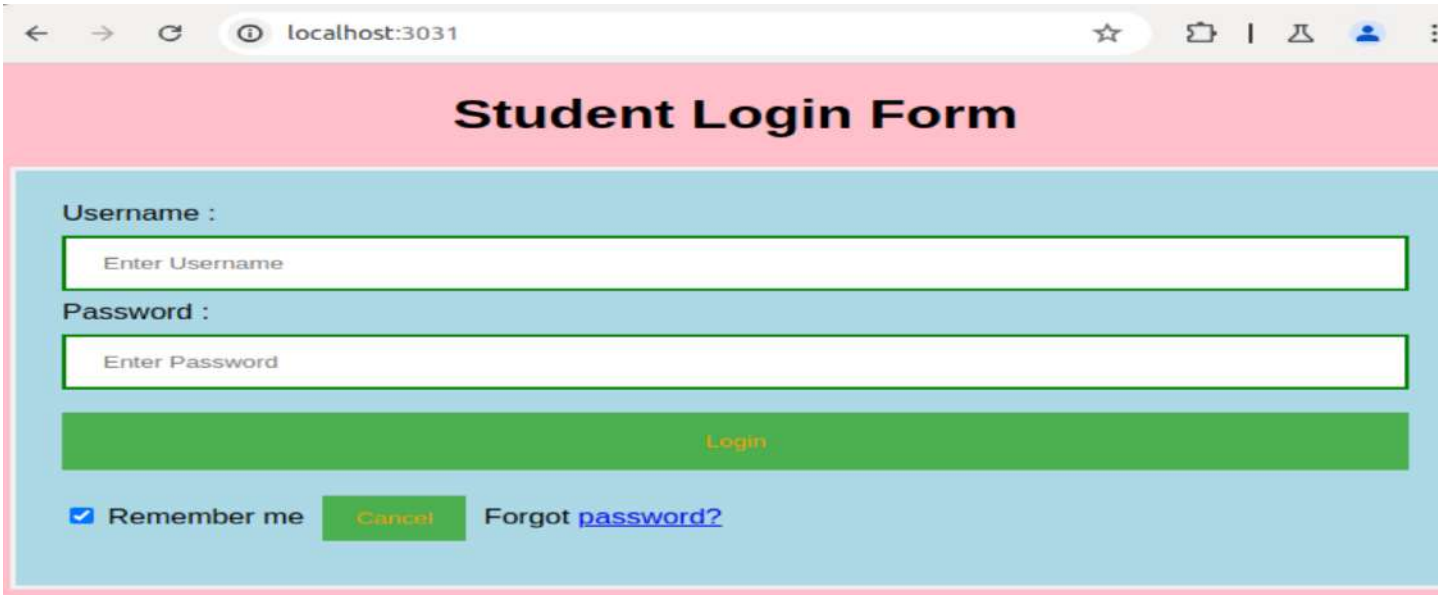
To check nginx service status:

```
root@595c2fa10371:/usr/share/nginx/html# service nginx status
nginx is not running ... failed!
```

```
root@595c2fa10371:/usr/share/nginx/html# service nginx start
2024/08/31 19:57:58 [notice] 170#170: using the "epoll" event method
2024/08/31 19:57:58 [notice] 170#170: nginx/1.27.1
2024/08/31 19:57:58 [notice] 170#170: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/08/31 19:57:58 [notice] 170#170: OS: Linux 5.4.0-37-generic
2024/08/31 19:57:58 [notice] 170#170: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/08/31 19:57:58 [notice] 171#171: start worker processes
root@595c2fa10371:/usr/share/nginx/html# 2024/08/31 19:57:58 [notice] 171#171: s
tart worker process 172
2024/08/31 19:57:58 [notice] 171#171: start worker process 173
172.17.0.1 - - [31/Aug/2024:19:58:36 +0000] "GET / HTTP/1.1" 200 1727 "-" "Mozil
la/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0
.0 Safari/537.36" "-"
```

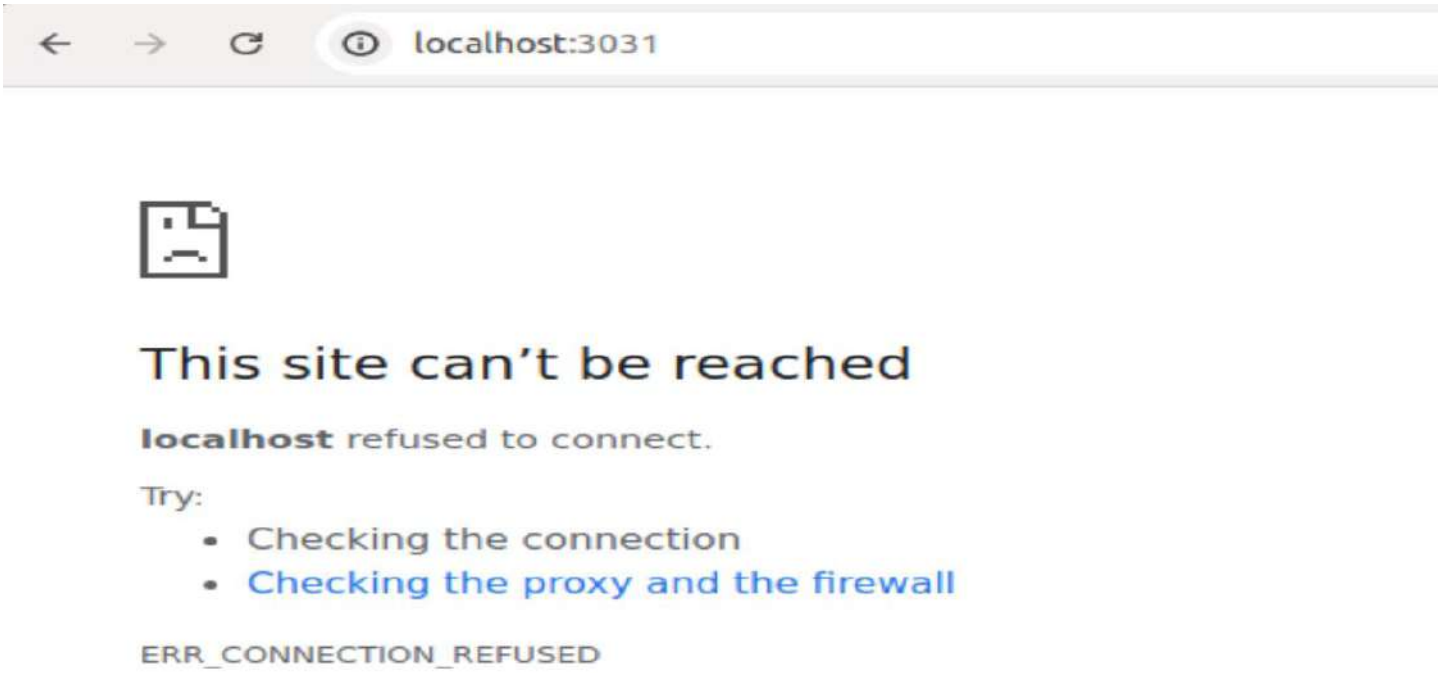
STEP 3:

GOTO BROWSER: localhost:3031




```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker stop 595
595
```

Goto browser→ Refresh page . Your Container is stopped now

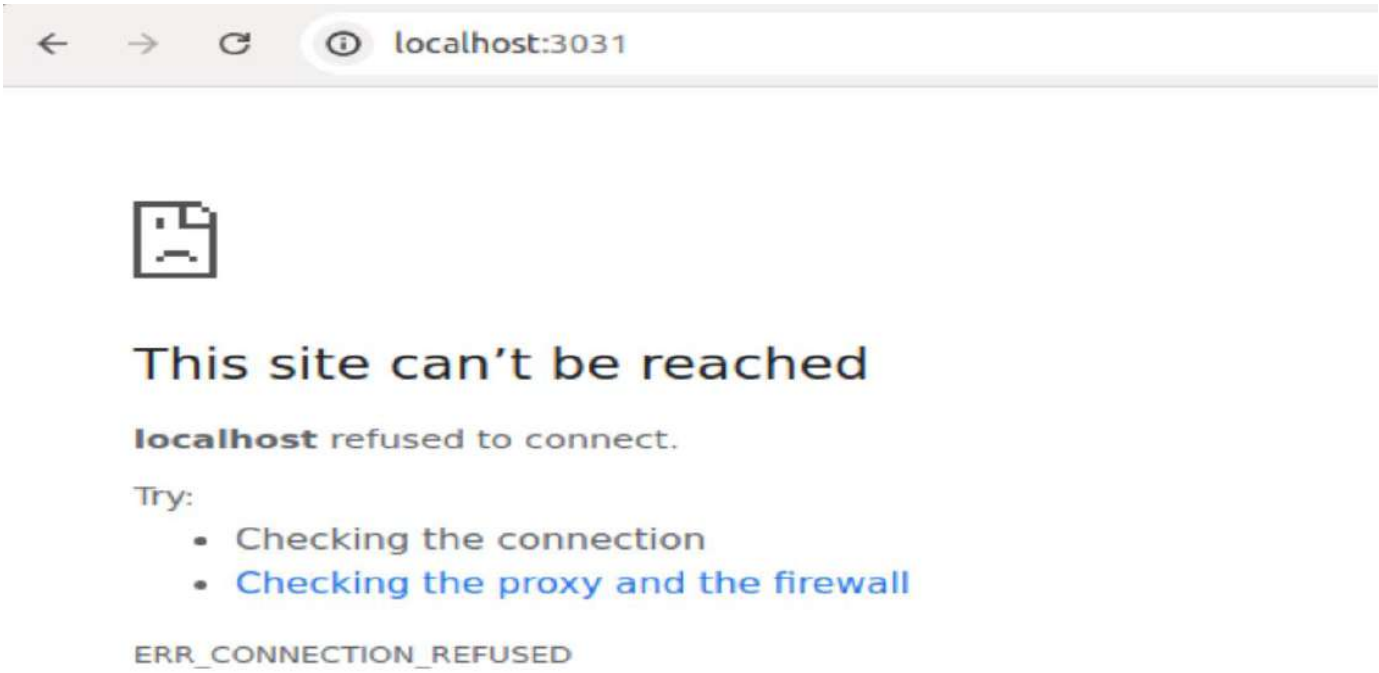



```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker start 595
595
```

Goto browser→ Refresh page . Your Container is not started

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker restart 595
595
```

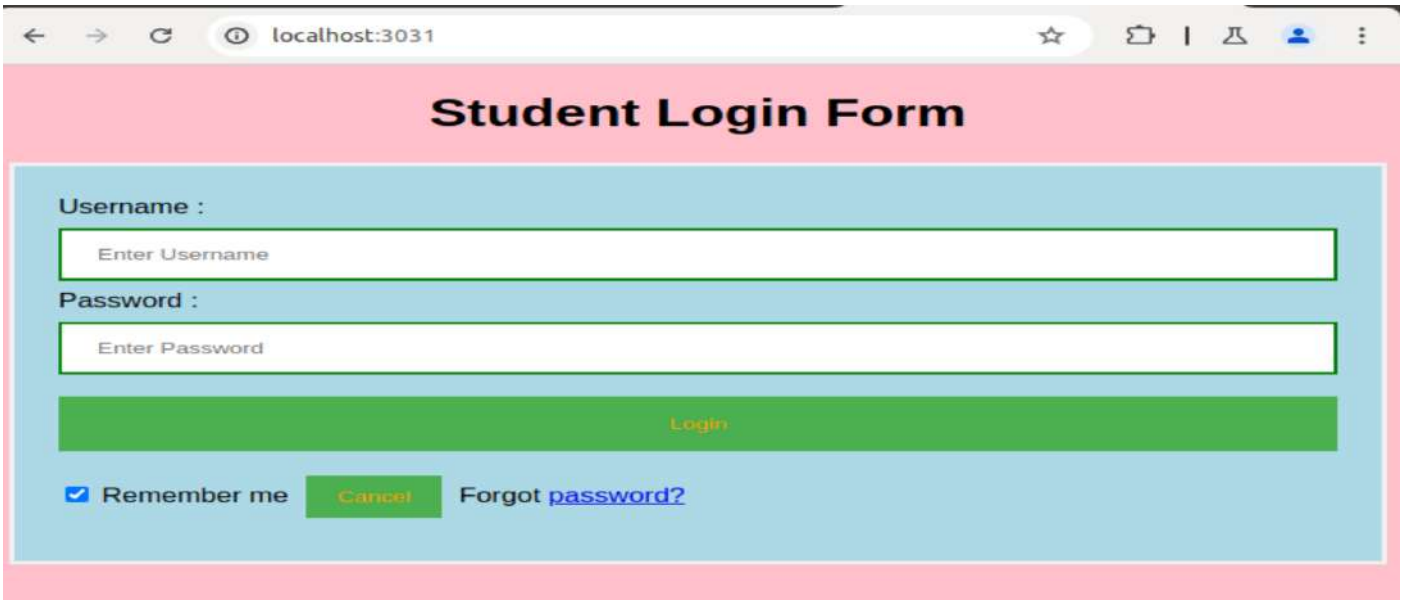
Goto browser→ Refresh page . Your Container is not restarted



docker exec 595 service nginx start

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker exec 595 service nginx start
2024/08/31 20:20:56 [notice] 18#18: using the "epoll" event method
2024/08/31 20:20:56 [notice] 18#18: nginx/1.27.1
2024/08/31 20:20:56 [notice] 18#18: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/08/31 20:20:56 [notice] 18#18: OS: Linux 5.4.0-37-generic
2024/08/31 20:20:56 [notice] 18#18: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/08/31 20:20:56 [notice] 19#19: start worker processes
2024/08/31 20:20:56 [notice] 19#19: start worker process 20
2024/08/31 20:20:56 [notice] 19#19: start worker process 21
```

Goto Browser and refresh it:



Firstly stop the container:

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker stop 595
595
```

#docker pause 595

#docker unpause 595

Then, Remove the Container

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker rm 595
595
```

To Verify container is removed or not:

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED       STATUS                     POR
TS      NAMES
bc8a1f0b194f   hello-world   "/hello"   6 days ago    Exited (0) 6 days ago
        peaceful_aryabhata
fbd148039aee   hello-world   "/hello"   6 days ago    Exited (0) 6 days ago
        infallible_cerf
```

PART II: DOCKERFILE

Creating a Docker Image for your Application:

This is the recommended workflow for creating your own Docker image for your application:

1. Write a Dockerfile for your application.
2. Build the image with docker build command.
3. Host your Docker image on a registry.
4. Pull and run the image on the target machine.

Docker builds images automatically by reading the instructions from a Dockerfile. It is a text file that contains all commands needed to build a given image.

STEP 1:

```
devasc@labvm:~$ cd Desktop/
devasc@labvm:~/Desktop$ cd DOCKER_LAB/
devasc@labvm:~/Desktop/DOCKER_LAB$ ls
docker-java  docker-nginx  get-docker.sh
devasc@labvm:~/Desktop/DOCKER_LAB$ cd docker-nginx/
devasc@labvm:~/Desktop/DOCKER_LAB/docker-nginx$ ls
Dockerfile  index.html  style.css
```

Step 2:

```
devasc@labvm:~/Desktop/DOCKER_LAB/docker-nginx$ nano Dockerfile
```

```
devasc@labvm: ~/Desktop/DOCKER_LAB/docker-nginx
File  Edit  View  Search  Terminal  Help
  GNU nano 4.8                                  Dockerfile
FROM ubuntu
LABEL author="Sujata Oak"
RUN apt-get update
RUN apt-get install nginx -y
COPY . /var/www/html/
EXPOSE 80
CMD ["nginx","-g","daemon off;"]
```

#docker build -t sujatadocker2024/websitetest .

```
devasc@labvm:~/Desktop/DOCKER_LAB/docker-nginx$ docker build -t sujatadocker2024/websit
etest .
[+] Building 1.2s (9/9) FINISHED                                    docker:default
 => [internal] load build definition from Dockerfile                          0.0s
 => => transferring dockerfile: 190B                                          0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest              0.8s
 => [internal] load .dockerignore                                             0.0s
 => => transferring context: 2B                                               0.0s
 => [1/4] FROM docker.io/library/ubuntu:latest@sha256:8a37d68f4f73ebf3d4e      0.0s
 => [internal] load build context                                             0.0s
 => => transferring context: 2.41kB                                           0.0s
 => CACHED [2/4] RUN apt-get update                                           0.0s
 => CACHED [3/4] RUN apt-get install nginx -y                                 0.0s
 => [4/4] COPY . /var/www/html/                                               0.1s
 => exporting to image                                                        0.1s
 => => exporting layers                                                       0.0s
```

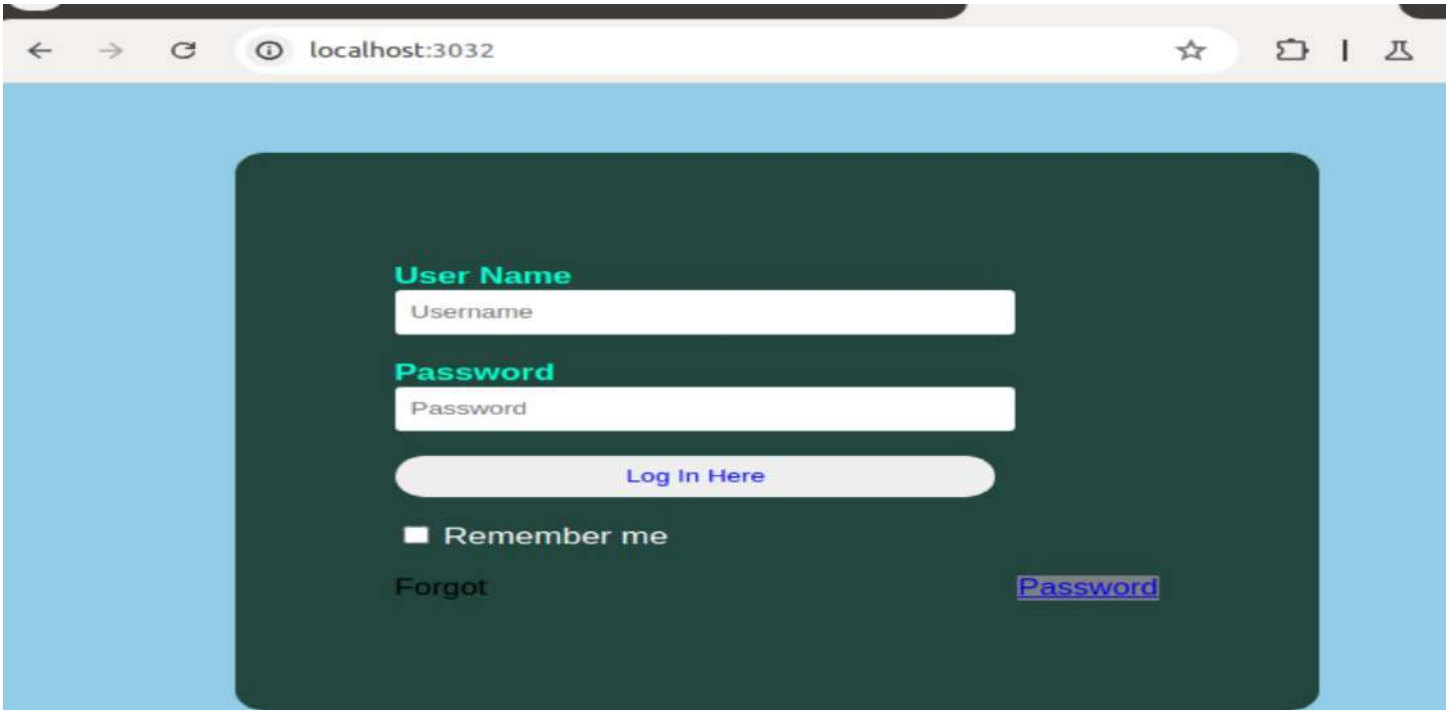
```
devasc@labvm:~/Desktop/DOCKER_LAB/docker-nginx$ docker images
REPOSITORY                        TAG       IMAGE ID       CREATED          SIZE
sujatadocker2024/websitetest      latest    da519d43aaa4   14 seconds ago   126MB
<none>                            <none>    5c8d66732fa0   3 minutes ago    126MB
<none>                            <none>    96e1fc59d3fa   8 minutes ago    126MB
nginx                             latest    5ef79149e0ec   2 weeks ago      188MB
mysql                             latest    a82a8f162e18   5 weeks ago      586MB
hello-world                       latest    d2c94e258dcb   16 months ago    13.3kB
```

Step 3: Run the container now:

```
devasc@labvm:~/Desktop/DOCKER_LAB/docker-nginx$ sudo su
root@labvm:/home/devasc/Desktop/DOCKER_LAB/docker-nginx# docker run -d -p 3032:80 --nam
e sujata_website da5
ff0cc2e19a4fc46780008104cbe41de8332724e133d4be93363b2c72a0dd9451
```

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB/docker-nginx# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORT
S                        NAMES
ff0cc2e19a4f   da5       "nginx -g 'daemon of…"   46 seconds ago   Up 46 seconds   0.0.
0.0:3032->80/tcp, :::3032->80/tcp   sujata_website
```

Step 4: Goto Browser: localhost:3032



STEP 5: How to push this image to your dockerhub :

```
devasc@labvm:~/Desktop/DOCKER_LAB/docker-nginx$ docker images
REPOSITORY                      TAG       IMAGE ID       CREATED          SIZE
sujatadocker2024/websitetest    latest    da519d43aaa4   14 seconds ago   126MB
<none>                          <none>    5c8d66732fa0   3 minutes ago    126MB
<none>                          <none>    96e1fc59d3fa   8 minutes ago    126MB
nginx                           latest    5ef79149e0ec   2 weeks ago      188MB
mysql                           latest    a82a8f162e18   5 weeks ago      586MB
hello-world                     latest    d2c94e258dcb   16 months ago    13.3kB
```

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB/docker-nginx# docker push sujatadocker2024/w
ebsitetest
Using default tag: latest
The push refers to repository [docker.io/sujatadocker2024/websitetest]
b4de1b3782f3: Preparing
7ddbe83467ff: Preparing
d17b9906ed77: Preparing
f36fd4bb7334: Preparing
denied: requested access to the resource is denied
```

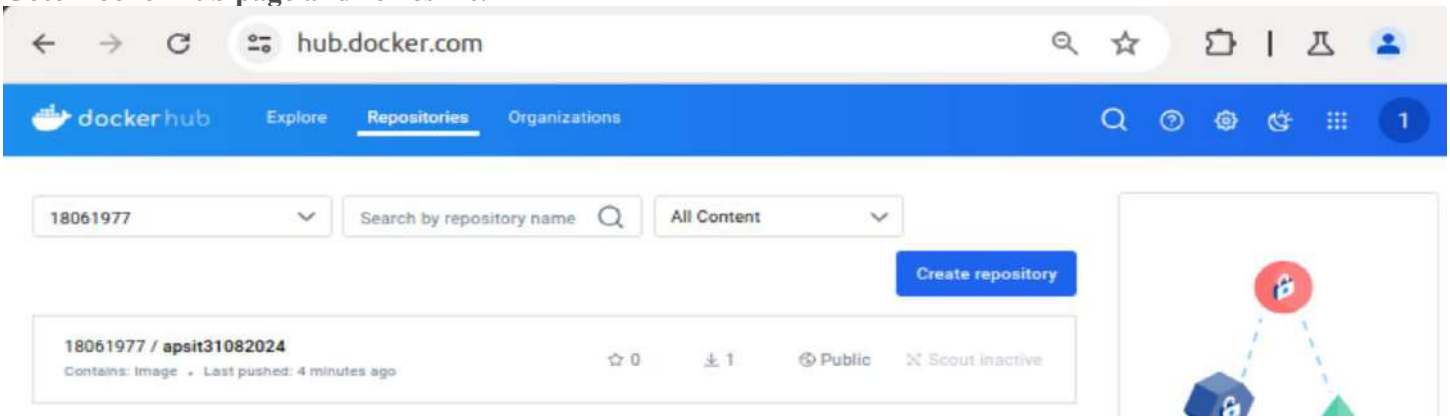
```
root@labvm:/home/devasc/Desktop/DOCKER_LAB/docker-nginx# docker tag da5 18061977/apsit3
1082024:v1
root@labvm:/home/devasc/Desktop/DOCKER_LAB/docker-nginx# docker images
REPOSITORY                      TAG       IMAGE ID       CREATED          SIZE
18061977/apsit31082024          v1        da519d43aaa4   19 minutes ago   126MB
sujatadocker2024/websitetest    latest    da519d43aaa4   19 minutes ago   126MB
<none>                          <none>    5c8d66732fa0   22 minutes ago   126MB
<none>                          <none>    96e1fc59d3fa   27 minutes ago   126MB
nginx                           latest    5ef79149e0ec   2 weeks ago      188MB
mysql                           latest    a82a8f162e18   5 weeks ago      586MB
hello-world                     latest    d2c94e258dcb   16 months ago    13.3kB
```

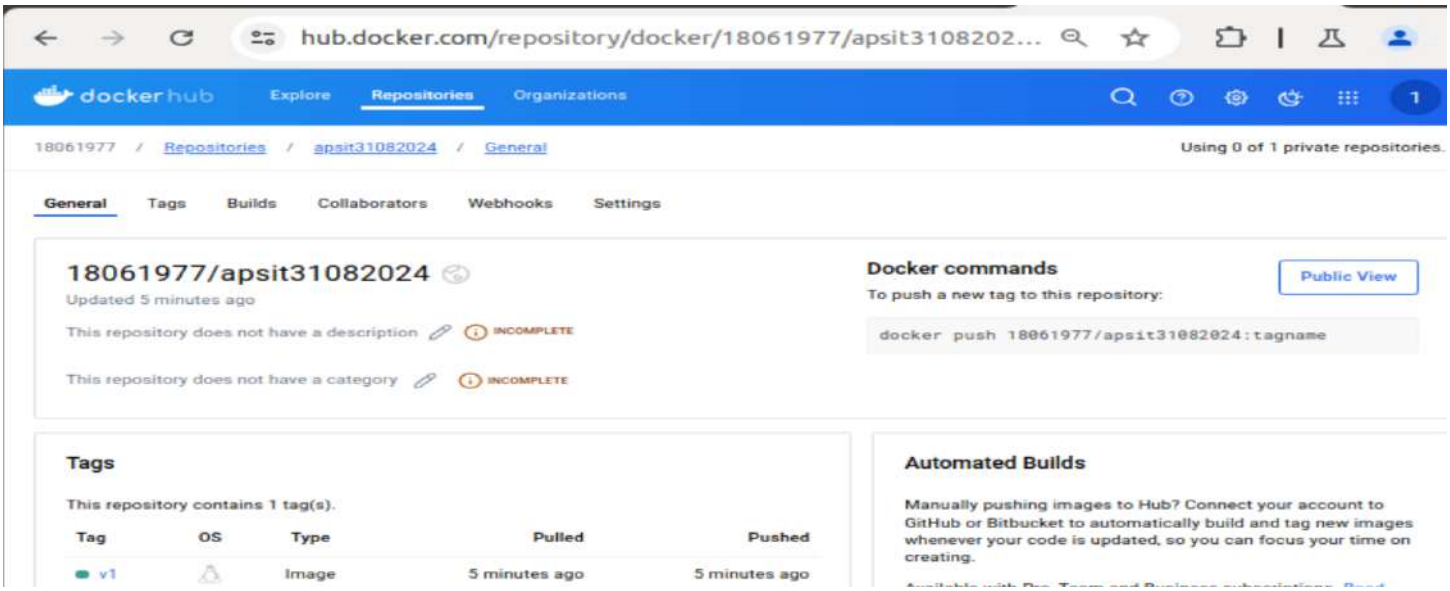
```
root@labvm:/home/devasc/Desktop/DOCKER_LAB/docker-nginx# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
```

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB/docker-nginx# docker push 18061977/apsit3108
2024:v1
The push refers to repository [docker.io/18061977/apsit31082024]
b4de1b3782f3: Pushed
7ddbe83467ff: Pushed
d17b9906ed77: Pushed
f36fd4bb7334: Pushed
v1: digest: sha256:754b62bd4141cff7bb3701128e61aa564e3544e66c8538bfc6e1e3e13ca1f9ae siz
e: 1161
```

Goto Docker hub page and refresh it:





Conclusion: In the experiment, we used various docker commands to pull images that were already built, also we created our own images by using docker file instructions for a sample web application and atlast we have pushed the image to docker hub account for others to use the repository.