Experiment No: 8

AIM: To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers.

THEORY: Docker Engine is an open source containerization technology for building and containerizing your applications. Docker Engine acts as a client-server application with:

- A server with a long-running daemon process dockerd.
- APIs which specify interfaces that programs can use to talk to and instruct the Docker daemon.
- A command line interface (CLI) client docker.

The CLI uses Docker APIs to control or interact with the Docker daemon through scripting or direct CLI commands. Many other Docker applications use the underlying API and CLI. The daemon creates and manage Docker objects, such as images, containers, networks, and volumes..

//Add the description of Docker Architecture & Docker Container Lifecycle Management here.

Installation of Docker:

To get started with Docker Engine on Ubuntu, make sure you meet the prerequisites, and then install Docker.

Prerequisites: OS requirements

To install Docker Engine, you need the 64-bit version of one of these Ubuntu versions:

- Ubuntu Hirsute 21.04
- Ubuntu Focal 20.04 (LTS)
- Ubuntu Bionic 18.04 (LTS)

Installation methods: You can install Docker Engine in different ways, depending on your needs:

1. Most users set up Docker's repositories and install from them
2. Some users download the DEB package and install it manually and manage upgrades completely manually.
3. In testing and development environments, some users choose to use automated convenience scripts to install Docker

Install using the convenience script: Docker provides a convenience script at get.docker.com to install Docker into development environments quickly and non-interactively. This example downloads the script from get.docker.com and runs it to install the latest stable release of Docker on Linux:

```
$ curl -fsSL https://get.docker.com -o get-docker.sh
```

```
$ sudo sh get-docker.sh
```

To get OS detail and version

```
it77@it77-OptiPlex-3050 :~$ lsb_release -a
```

Uninstall old versions

```
it77@it77-OptiPlex-3050 :~$ sudo su

t77@it77-OptiPlex-305i0 :~$ sudo apt-get remove docker docker-engine docker.io containerd runc

it77@it77-OptiPlex-3050 :~$ sudo apt install curl

root@it77-OptiPlex-3050:/home/it77# curl -fsSL https://get.docker.com -o get-docker.sh
```

Examine scripts downloaded from the internet

```
root@it77-OptiPlex-3050:/home/it77# ls

 root@it77-OptiPlex-3050:/home/it77# sudo sh get-docker.sh
```

Basic Docker Commands:

Check the version of Docker installed

```
root@it77-OptiPlex-3050:/home/it77# docker --version
```

Running existing Docker images: Go to Docker public repository at https://hub.docker.com to get the **official images available for testing purpose**

Run docker image

```
root@it77-OptiPlex-3050:/home/it77# docker run docker/whalesay cowsay hello_you

root@it77-OptiPlex-3050:/home/it77# docker run docker/whalesay cowsay hello_me
```

Check all pulled images

```
root@it77-OptiPlex-3050:/home/it77# docker images
```

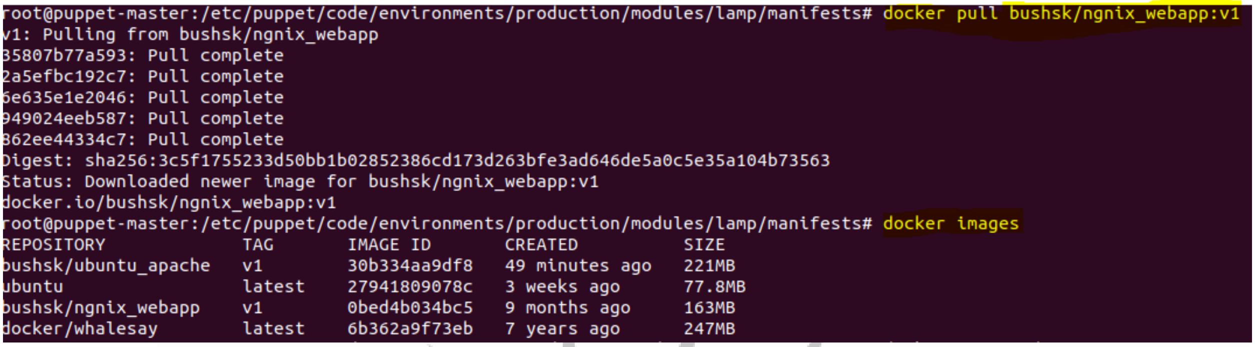
Pull the sample images

```
root@it77-OptiPlex-3050:/home/it77# sudo docker pull postgres

root@it77-OptiPlex-3050:/home/it77# docker images
```

Check all running container

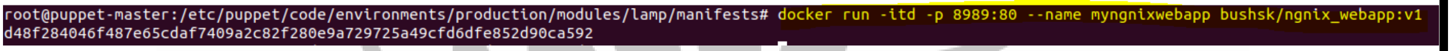
```
root@it77-OptiPlex-3050:/home/it77# docker ps // note the container id

root@it77-OptiPlex-3050:/home/it77# docker ps -a //previously ran containers
```

Pulling the created webapp project:



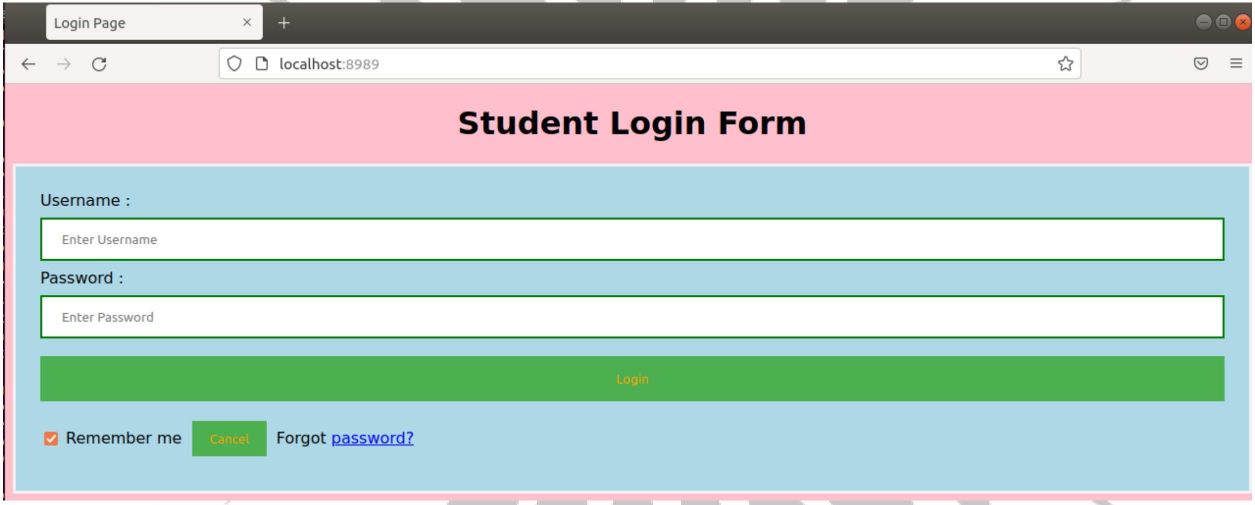
```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker pull bushsk/ngnix_webapp:v1
v1: Pulling from bushsk/ngnix_webapp
35807b77a593: Pull complete
2a5efbc192c7: Pull complete
6e635e1e2046: Pull complete
949024eeb587: Pull complete
862ee44334c7: Pull complete
Digest: sha256:3c5f1755233d50bb1b02852386cd173d263bfe3ad646de5a0c5e35a104b73563
Status: Downloaded newer image for bushsk/ngnix_webapp:v1
docker.io/bushsk/ngnix_webapp:v1
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker images
REPOSITORY             TAG       IMAGE ID       CREATED          SIZE
bushsk/ubuntu_apache   v1        30b334aa9df8   49 minutes ago   221MB
ubuntu                 latest    27941809078c   3 weeks ago      77.8MB
bushsk/ngnix_webapp    v1        0bed4b034bc5   9 months ago     163MB
docker/whalesay        latest    6b362a9f73eb   7 years ago      247MB
```

Running the pulled image



```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker run -itd -p 8989:80 --name myngnixwebapp bushsk/ngnix_webapp:v1
d48f284046f487e65cdaf7409a2c82f280e9a729725a49cfd6dfe852d90ca592
```

Running or testing webapp on port 8989:



Student Login Form

Username :

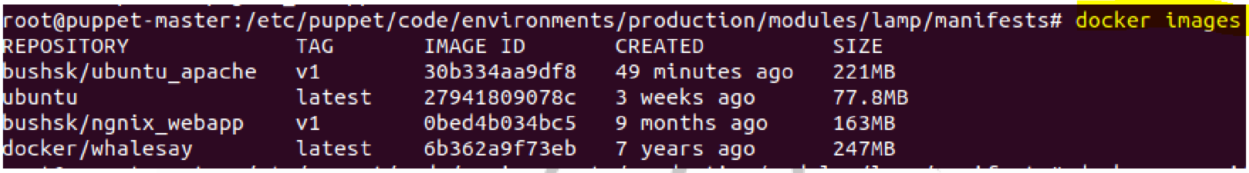
Enter Username

Password :

Enter Password

Login

☑ Remember me Cancel Forgot password?



```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker images
REPOSITORY             TAG       IMAGE ID       CREATED          SIZE
bushsk/ubuntu_apache   v1        30b334aa9df8   49 minutes ago   221MB
ubuntu                 latest    27941809078c   3 weeks ago      77.8MB
bushsk/ngnix_webapp    v1        0bed4b034bc5   9 months ago     163MB
docker/whalesay        latest    6b362a9f73eb   7 years ago      247MB
```

Creating a new image from the running containers:

Pull the Ubuntu as a base image :

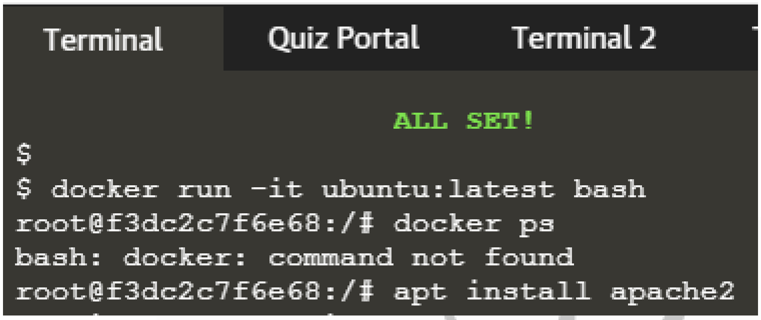
root@it77-OptiPlex-3050:/home/it77# docker pull ubuntu:latest

root@it77-OptiPlex-3050:/home/it77# docker images

Run the Ubuntu image with a command in a container: Getting a bash in Ubuntu

root@it77-OptiPlex-3050:/home/it77# docker run -it ubuntu:latest bash

root@it77-OptiPlex-3050:/home/it77# docker ps



```
              ALL SET!
$
$ docker run -it ubuntu:latest bash
root@f3dc2c7f6e68:/# docker ps
bash: docker: command not found
root@f3dc2c7f6e68:/# apt install apache2
```

Note the <12 digit hash value> is the id of the shell. Here it is 67e9bd16d77b

Create an Apache Server and host index.html in the Containers

root@67e9bd16d77b:/# apt update

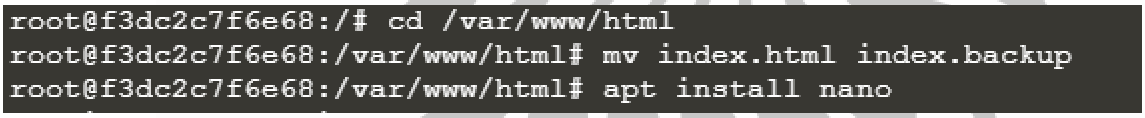
root@67e9bd16d77b:/# apt install apache2

root@67e9bd16d77b:/# cd /var/www/html

root@67e9bd16d77b:/var/www/html# mv index.html index.backup

root@67e9bd16d77b:/var/www/html# ls

root@67e9bd16d77b:/var/www/html# apt install nano



```
root@f3dc2c7f6e68:/# cd /var/www/html
root@f3dc2c7f6e68:/var/www/html# mv index.html index.backup
root@f3dc2c7f6e68:/var/www/html# apt install nano
```

root@67e9bd16d77b:/var/www/html# nano index.html

root@67e9bd16d77b:/var/www/html# cat index.html

<html>

<title> First page</title>

<body bgcolor="pink">

Our home Page

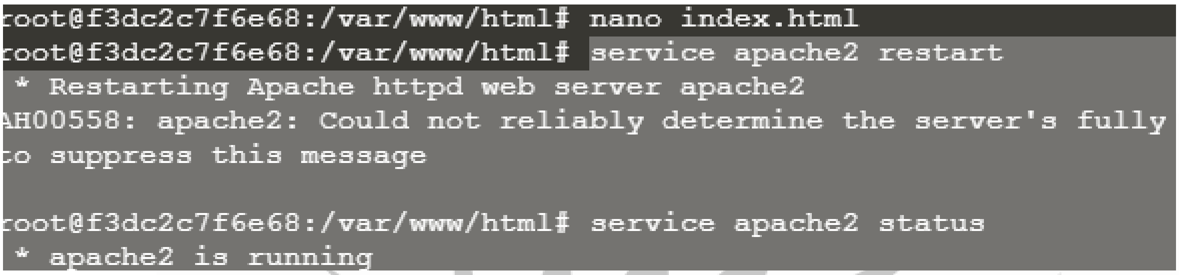
</body>

</html>

root@67e9bd16d77b:/var/www/html# service apache2 start

root@67e9bd16d77b:/var/www/html# service apache2 status

* apache2 is running



```
root@f3dc2c7f6e68:/var/www/html# nano index.html
root@f3dc2c7f6e68:/var/www/html# service apache2 restart
 * Restarting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully
to suppress this message

root@f3dc2c7f6e68:/var/www/html# service apache2 status
 * apache2 is running
```

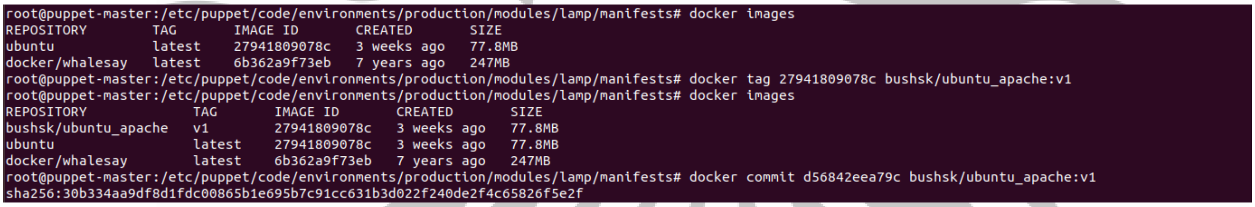
Committing an image:

Open a new terminal and run the foll. Commands:

Tagging <Ubuntu_apache:v1> image using image id:

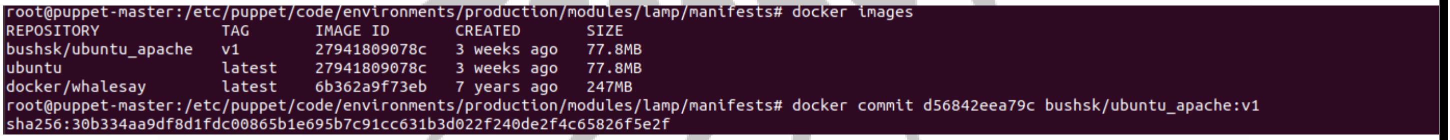
root@it77-OptiPlex-3050:/home/it77# docker tag 27941809078c bushsk/ubuntu_apache:v1

27941809078c is the image id of the running container



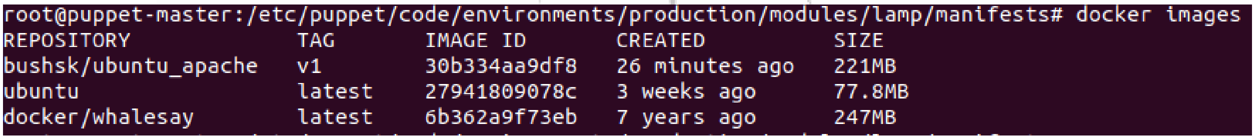
```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker images
REPOSITORY        TAG       IMAGE ID       CREATED       SIZE
ubuntu            latest    27941809078c   3 weeks ago   77.8MB
docker/whalesay   latest    6b362a9f73eb   7 years ago   247MB
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker tag 27941809078c bushsk/ubuntu_apache:v1
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker images
REPOSITORY             TAG       IMAGE ID       CREATED       SIZE
bushsk/ubuntu_apache   v1        27941809078c   3 weeks ago   77.8MB
ubuntu                 latest    27941809078c   3 weeks ago   77.8MB
docker/whalesay        latest    6b362a9f73eb   7 years ago   247MB
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker commit d56842eea79c bushsk/ubuntu_apache:v1
sha256:30b334aa9df8d1fdc00865b1e695b7c91cc631b3d022f240de2f4c65826f5e2f
```

Checking its size and committing an image



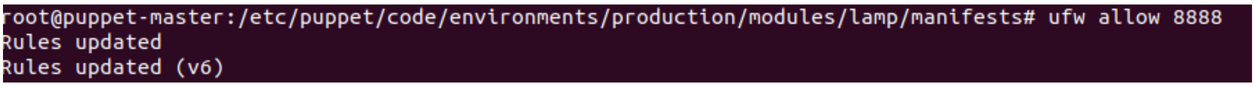
```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker images
REPOSITORY             TAG       IMAGE ID       CREATED       SIZE
bushsk/ubuntu_apache   v1        27941809078c   3 weeks ago   77.8MB
ubuntu                 latest    27941809078c   3 weeks ago   77.8MB
docker/whalesay        latest    6b362a9f73eb   7 years ago   247MB
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker commit d56842eea79c bushsk/ubuntu_apache:v1
sha256:30b334aa9df8d1fdc00865b1e695b7c91cc631b3d022f240de2f4c65826f5e2f
```

Checking the size of committed image



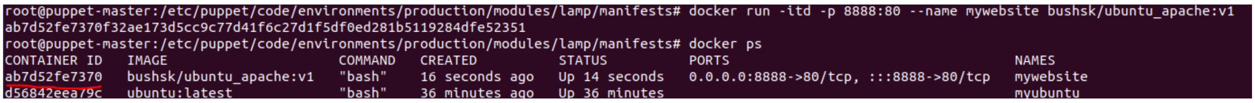
```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker images
REPOSITORY             TAG       IMAGE ID       CREATED          SIZE
bushsk/ubuntu_apache   v1        30b334aa9df8   26 minutes ago   221MB
ubuntu                 latest    27941809078c   3 weeks ago      77.8MB
docker/whalesay        latest    6b362a9f73eb   7 years ago      247MB
```

Allowing port 8888



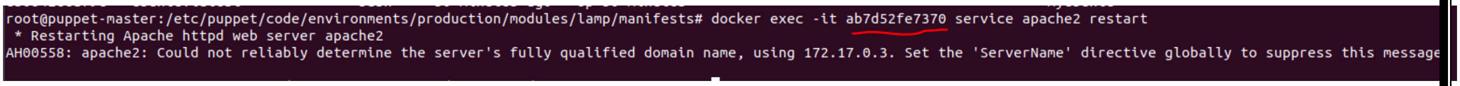
```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# ufw allow 8888
Rules updated
Rules updated (v6)
```

Running a named <mywebsite> image Ubuntu_apache:v1 image and note container id



```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker run -itd -p 8888:80 --name mywebsite bushsk/ubuntu_apache:v1
ab7d52fe7370f32ae173d5cc9c77d41f6c27d1f5df0ed281b5119284dfe52351
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker ps
CONTAINER ID   IMAGE                     COMMAND   CREATED          STATUS          PORTS                                   NAMES
ab7d52fe7370   bushsk/ubuntu_apache:v1   "bash"    16 seconds ago   Up 14 seconds   0.0.0.0:8888->80/tcp, :::8888->80/tcp   mywebsite
d56842eea79c   ubuntu:latest             "bash"    36 minutes ago   Up 36 minutes                                           myubuntu
```

Restarting an apache by attaching it into running containers using container id:



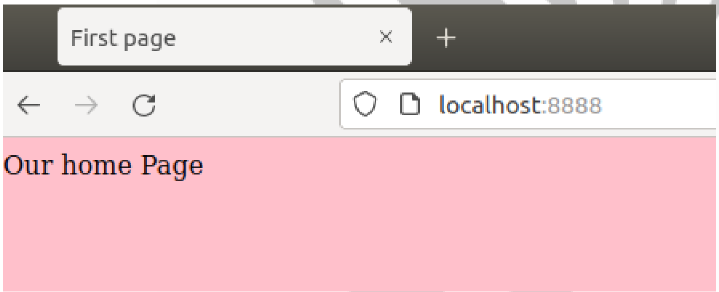
```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker exec -it ab7d52fe7370 service apache2 restart
 * Restarting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
```

Get the IP address of your system

root@it77-OptiPlex-3050:/home/it77# ifconfig

Test your web application

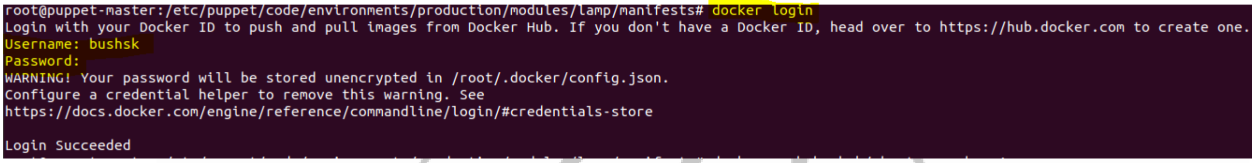
Open a browser and put <IP: port number> (as 8888) or type localhost:8888



Our home Page

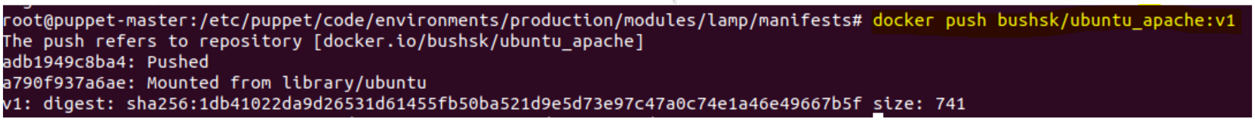
Login in hub

root@it77-OptiPlex-3050:/home/it77# docker login



```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: bushsk
Password:
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/#credentials-store

Login Succeeded
```

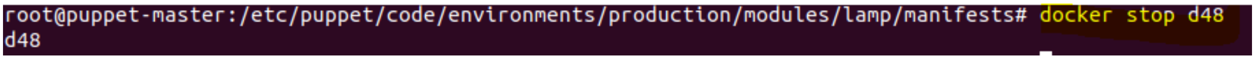
Pushing the image on Docker Hub



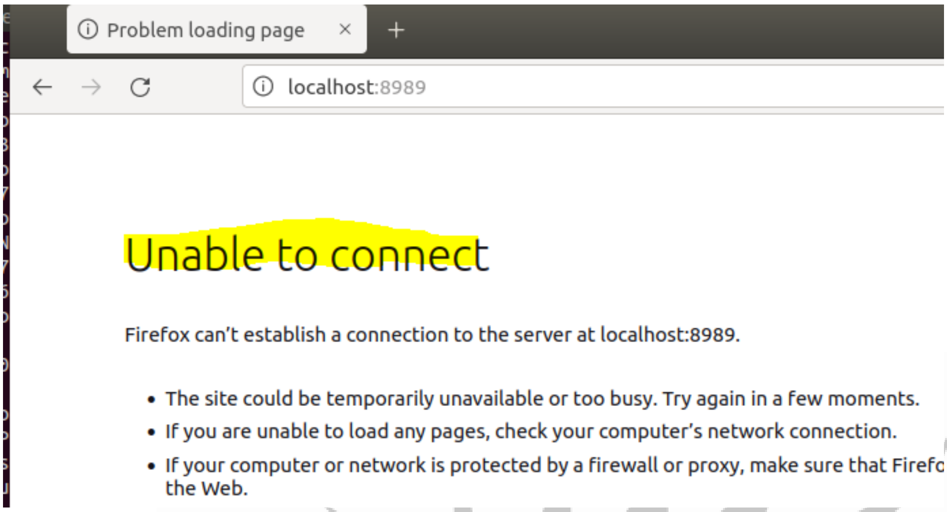
```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker push bushsk/ubuntu_apache:v1
The push refers to repository [docker.io/bushsk/ubuntu_apache]
adb1949c8ba4: Pushed
a790f937a6ae: Mounted from library/ubuntu
v1: digest: sha256:1db41022da9d26531d61455fb50ba521d9e5d73e97c47a0c74e1a46e49667b5f size: 741
```

Stopping and removing containers using container id

root@it77-OptiPlex-3050:/home/it77# docker stop bd9fdf66daaf



```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# docker stop d48
d48
```



Unable to connect

Firefox can’t establish a connection to the server at localhost:8989.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer’s network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefo the Web.

root@it77-OptiPlex-3050:/home/it77# docker kill bd9fdf66daaf -- > not advisable

root@it77-OptiPlex-3050:/home/it77# docker rm bd9fdf66daaf

root@it77-OptiPlex-3050:/home/it77# docker ps -a

Remove multiple containers using container ids:

root@it77-OptiPlex-3050:/home/it77# docker rm bd9 abc

root@it77-OptiPlex-3050:/home/it77# docker rm $(docker ps –aq)

root@it77-OptiPlex-3050:/home/it77# docker ps -a

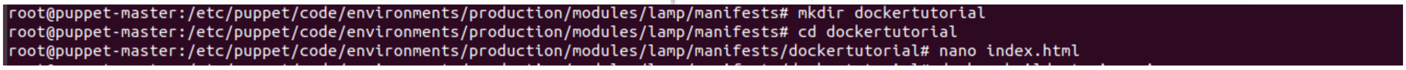
Deleting the images

root@it77-OptiPlex-3050:/home/it77# docker images

root@it77-OptiPlex-3050:/home/it77# docker rmi 5c6

Creating an image using docker file script:

DOCKER FILE



```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# mkdir dockertutorial
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests# cd dockertutorial
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests/dockertutorial# nano index.html
```

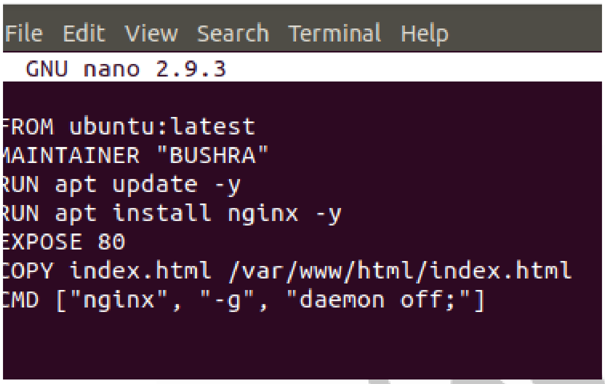
root@it77-OptiPlex-3050:/home/it77# mkdir dockertutorial

root@it77-OptiPlex-3050:/home/it77# cd dockertutorial

root@it77-OptiPlex-3050:/home/it77/dockertutorial# pwd

/home/it77/dockertutorial

```
root@it77-OptiPlex-3050:/home/it77/dockertutorial# nano index.html

root@it77-OptiPlex-3050:/home/it77/dockertutorial# ls

index.html
```



```
FROM ubuntu:latest
MAINTAINER "BUSHRA"
RUN apt update -y
RUN apt install nginx -y
EXPOSE 80
COPY index.html /var/www/html/index.html
CMD ["nginx", "-g", "daemon off;"]
```

```
root@it77-OptiPlex-3050:/home/it77/dockertutorial# nano Dockerfile
```

FROM ubuntu:latest

MAINTAINER "BUSHRA"

RUN apt update -y

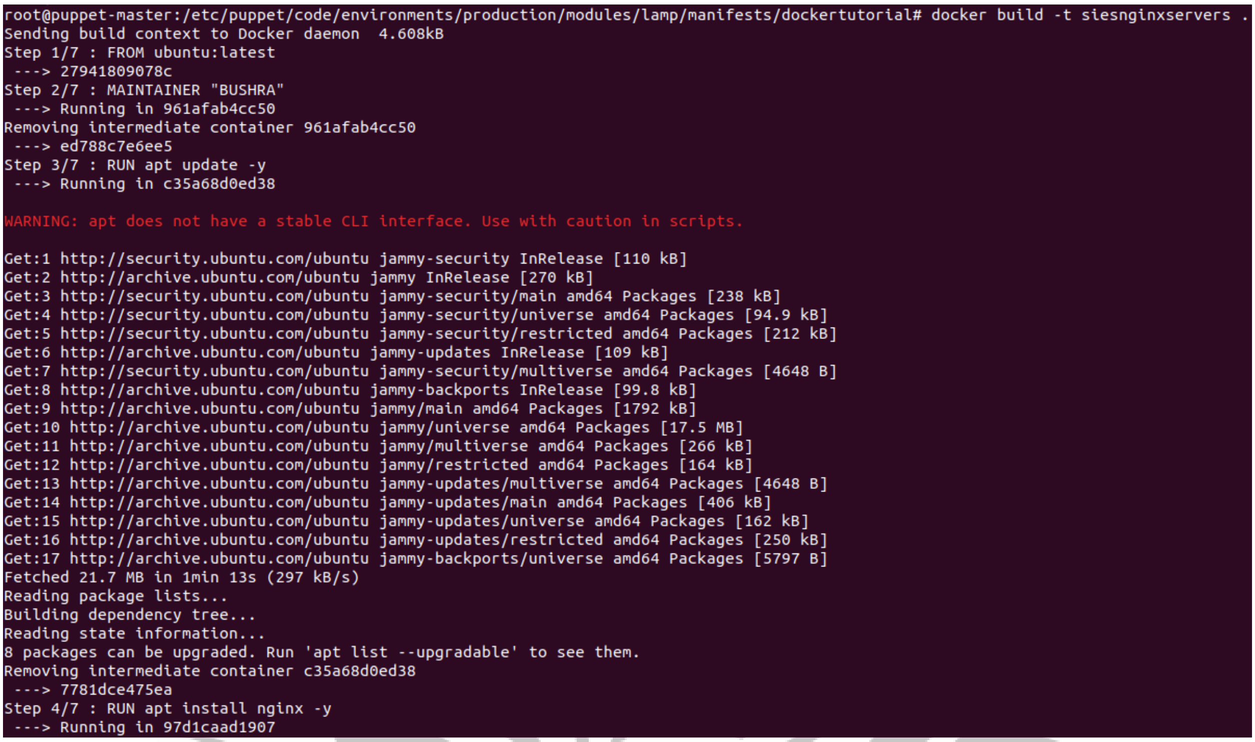
RUN apt install nginx -y

EXPOSE 80

COPY index.html /var/www/html/index.html

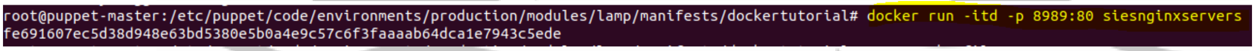
CMD ["nginx", "-g", "daemon off;"]

```
root@it77-OptiPlex-3050:/home/it77/dockertutorial# docker build -t siesnginxservers .
```



```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests/dockertutorial# docker build -t siesnginxservers .
Sending build context to Docker daemon  4.608kB
Step 1/7 : FROM ubuntu:latest
 ---> 27941809078c
Step 2/7 : MAINTAINER "BUSHRA"
 ---> Running in 961afab4cc50
Removing intermediate container 961afab4cc50
 ---> ed788c7e6ee5
Step 3/7 : RUN apt update -y
 ---> Running in c35a68d0ed38

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

Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [238 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [94.9 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [212 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [109 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4648 B]
Get:8 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [4648 B]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [406 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [162 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [250 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [5797 B]
Fetched 21.7 MB in 1min 13s (297 kB/s)
Reading package lists...
Building dependency tree...
Reading state information...
8 packages can be upgraded. Run 'apt list --upgradable' to see them.
Removing intermediate container c35a68d0ed38
 ---> 7781dce475ea
Step 4/7 : RUN apt install nginx -y
 ---> Running in 97d1caad1907
```

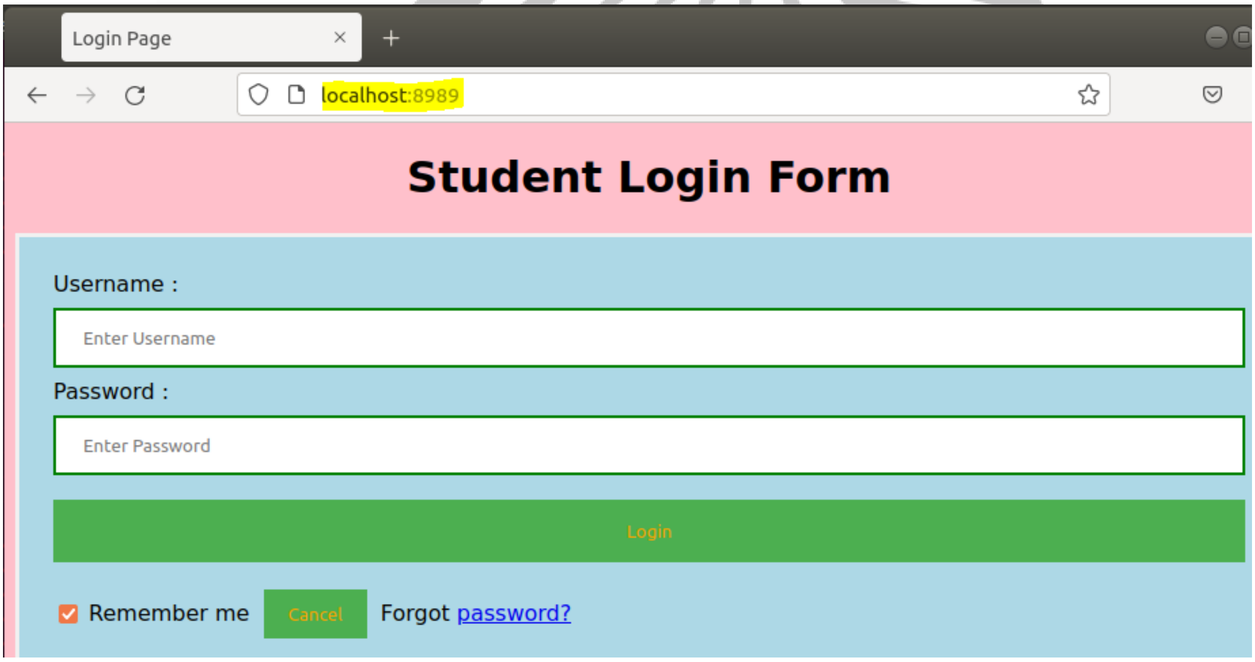
root@it77-OptiPlex-3050:/home/it77/dockertutorial# docker run -itd -p 8989:80 siesnginxservers



```
root@puppet-master:/etc/puppet/code/environments/production/modules/lamp/manifests/dockertutorial# docker run -itd -p 8989:80 siesnginxservers
fe691607ec5d38d948e63bd5380e5b0a4e9c57c6f3faaaab64dca1e7943c5ede
```

root@it77-OptiPlex-3050:/home/it77/dockertutorial# ufw allow 8989 //if required since already allowed

Open browser and put ip with port number as 8989



Login Page

localhost:8989

Student Login Form

Username :

Enter Username

Password :

Enter Password

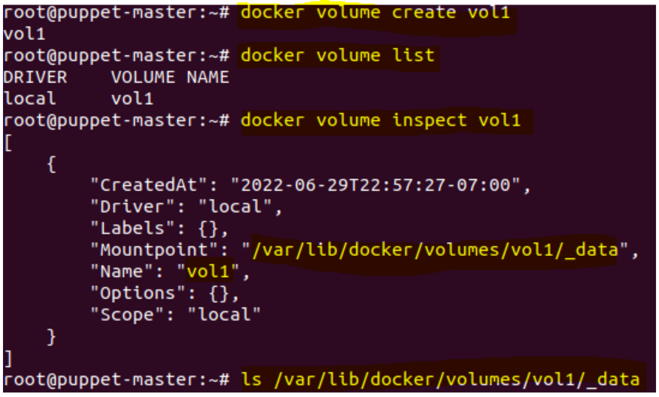
Docker Volume: Making data persistent across the containers and volume

1. Create a volume, List a volume and inspect a volume

```
root@puppet-master:~# docker volume create vol1

vol1

root@puppet-master:~# docker volume list

DRIVER    VOLUME NAME

local     vol1

root@puppet-master:~# docker volume inspect vol1

[

   {

      "CreatedAt": "2022-06-29T22:57:27-07:00",

      "Driver": "local",

      "Labels": {},

      "Mountpoint": "/var/lib/docker/volumes/vol1/_data",

      "Name": "vol1",

      "Options": {},

      "Scope": "local"

   }

]
```

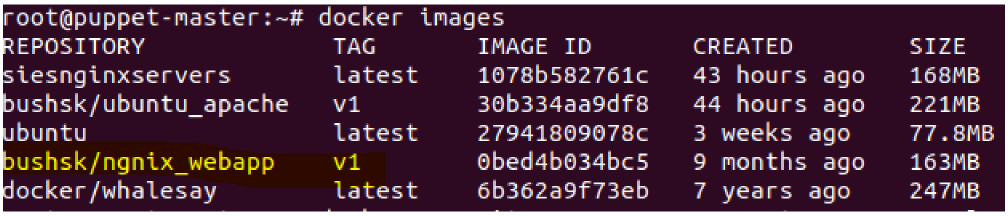
Check the existing content of a created volume :

```
root@puppet-master:~# ls /var/lib/docker/volumes/vol1/_data
```

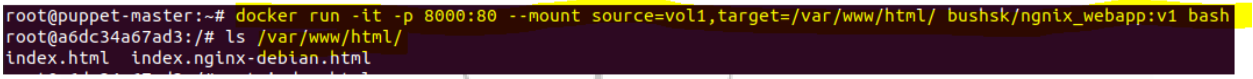


```
root@puppet-master:~# docker volume create vol1
vol1
root@puppet-master:~# docker volume list
DRIVER    VOLUME NAME
local     vol1
root@puppet-master:~# docker volume inspect vol1
[
    {
        "CreatedAt": "2022-06-29T22:57:27-07:00",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/vol1/_data",
        "Name": "vol1",
        "Options": {},
        "Scope": "local"
    }
]
root@puppet-master:~# ls /var/lib/docker/volumes/vol1/_data
```

2. **Select an existing image and run it using --mount option to attach the target location of container to the source location of volume**

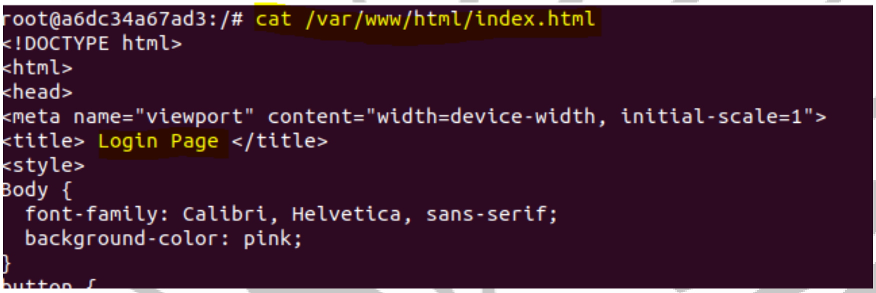


```
root@puppet-master:~# docker images
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
siesnginxservers         latest    1078b582761c   43 hours ago    168MB
bushsk/ubuntu_apache     v1        30b334aa9df8   44 hours ago    221MB
ubuntu                   latest    27941809078c   3 weeks ago     77.8MB
bushsk/ngnix_webapp      v1        0bed4b034bc5   9 months ago    163MB
docker/whalesay          latest    6b362a9f73eb   7 years ago     247MB
```

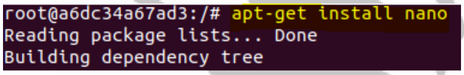


```
root@puppet-master:~# docker run -it -p 8000:80 --mount source=vol1,target=/var/www/html/ bushsk/ngnix_webapp:v1 bash
root@a6dc34a67ad3:/# ls /var/www/html/
index.html  index.nginx-debian.html
```

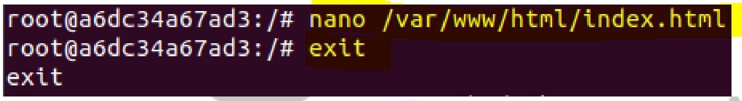
3. **Check the existing content of a container in the target location and make few changes in the running container by adding a layer of nano and modifying the index.html as shown**



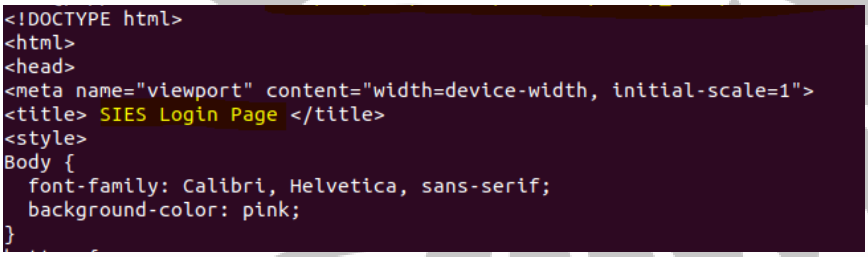
```
root@a6dc34a67ad3:/# cat /var/www/html/index.html
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Login Page </title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-color: pink;
}
button {
```



```
root@a6dc34a67ad3:/# apt-get install nano
Reading package lists... Done
Building dependency tree
```

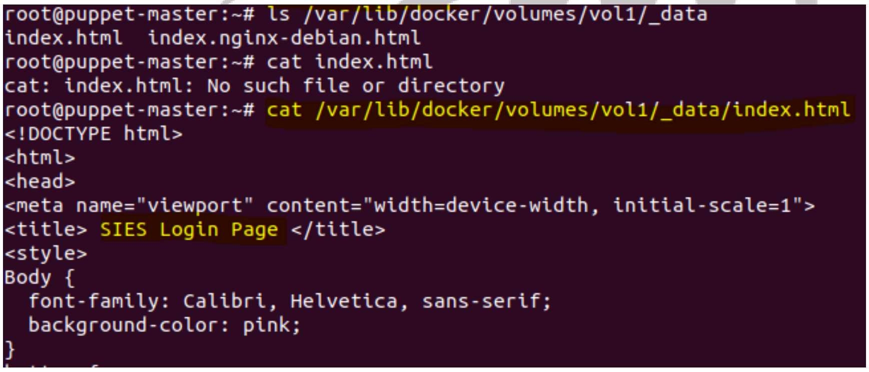


```
root@a6dc34a67ad3:/# nano /var/www/html/index.html
root@a6dc34a67ad3:/# exit
exit
```



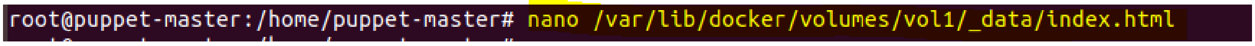
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> SIES Login Page </title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-color: pink;
}
```

4. **Now check the modified content visible in the source location of a volume:**

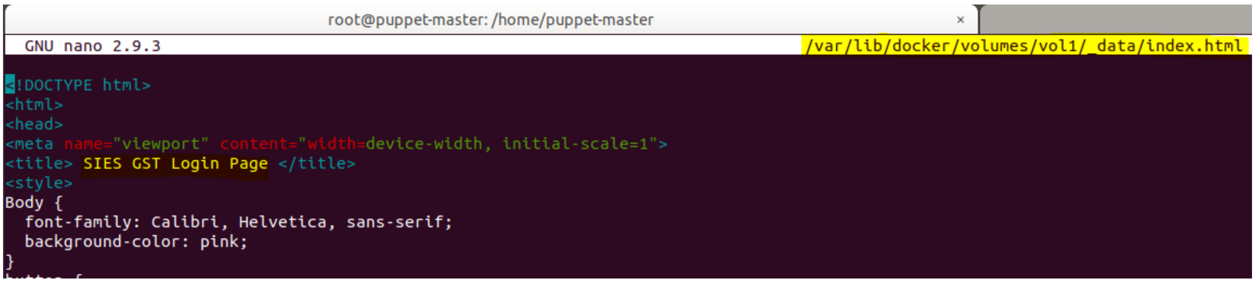


```
root@puppet-master:~# ls /var/lib/docker/volumes/vol1/_data
index.html  index.nginx-debian.html
root@puppet-master:~# cat index.html
cat: index.html: No such file or directory
root@puppet-master:~# cat /var/lib/docker/volumes/vol1/_data/index.html
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> SIES Login Page </title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-color: pink;
}
```

5. **Similarly, modify the content in the source location of a volume:**

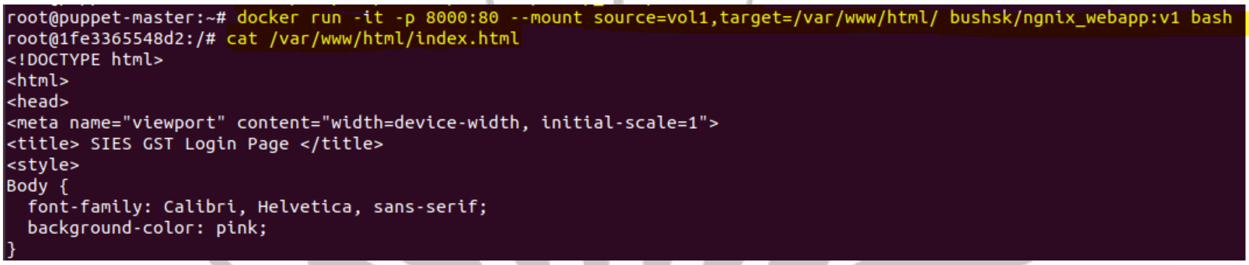


```
root@puppet-master:/home/puppet-master# nano /var/lib/docker/volumes/vol1/_data/index.html
```



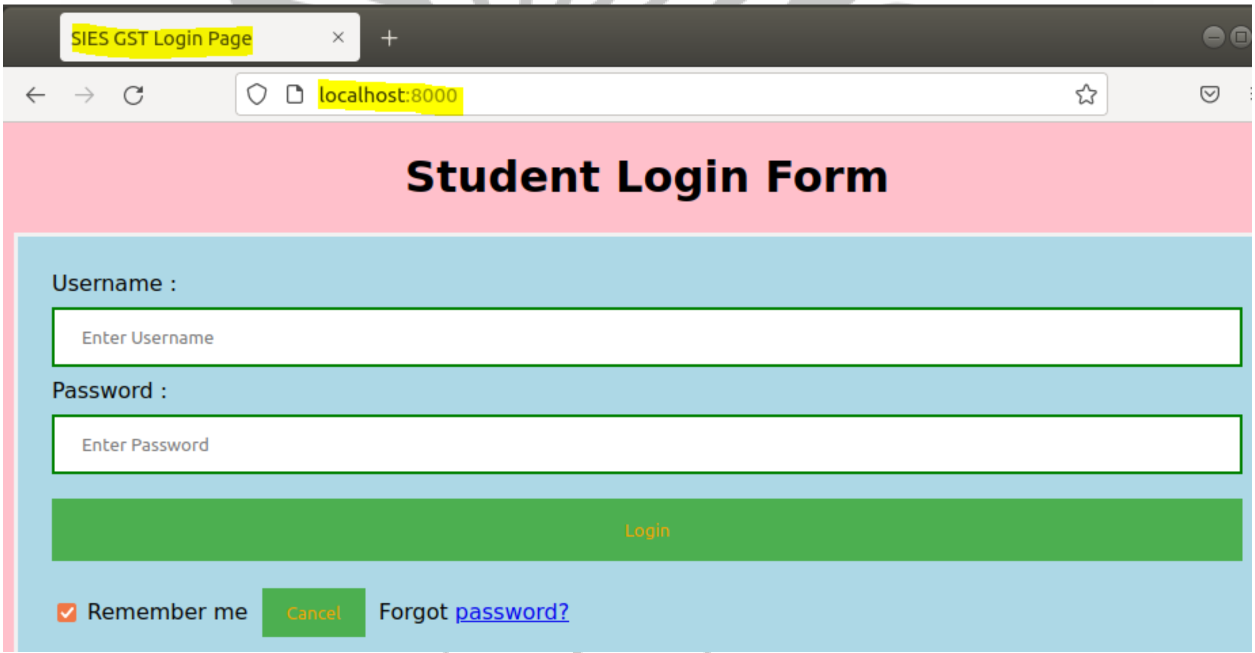
```
GNU nano 2.9.3                                    /var/lib/docker/volumes/vol1/_data/index.html

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> SIES GST Login Page </title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-color: pink;
}
```

6. **Now , run the container and check the modified content visible in the target location of a container:**



```
root@puppet-master:~# docker run -it -p 8000:80 --mount source=vol1,target=/var/www/html/ bushsk/ngnix_webapp:v1 bash
root@1fe3365548d2:/# cat /var/www/html/index.html
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> SIES GST Login Page </title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-color: pink;
}
```

7. **Open a web browser and got locathost:8000 and verify the output:**



Student Login Form

Username :

Enter Username

Password :

Enter Password

Login

☑ Remember me Cancel Forgot password?

Conclusion: Add conclusion here

Experiment No: 9

AIM: To learn Dockerfile instructions, build an image for a sample web application using Dockerfile.

THEORY:

Docker also gives you the capability to create your own Docker images, and it can be done with the help of Docker Files. A Docker File is a simple text file with instructions on how to build your images.

FROM keyword tells Docker, from which base image you want to base your image from. In our example, we are creating an image from the Ubuntu image.

The next command is the person who is going to maintain this image. Here you specify the **MAINTAINER** keyword and just mention the email ID.

The **RUN** command is used to run instructions against the image. In our case, we first update our Ubuntu system and then install the nginx server on our Ubuntu image.

The last **CMD** command is used to display a message to the user.

Create a file with a name as Dockerfile:

```
root@it77-OptiPlex-3050:/home/it77/dockertutorial# nano Dockerfile
```

```
FROM ubuntu:latest                          # Adding a layer of Ubuntu as a base image
MAINTAINER "BUSHRA"                         # Adding an author
RUN apt update –y                           # Adding a layer of ‘apt-update’ in our image
RUN apt install nginx –y                    # Adding a layer of ‘nginx’ webserver in our image
EXPOSE 80                                   # Exposing port 80 of web server
COPY index.html  /var/www/html/index.html # Hosting our web page / application
CMD ["nginx", "-g", "daemon off;"]          # Starting nginx webserver in foreground
```

```
root@it77-OptiPlex-3050:/home/it77/dockertutorial# docker build -t siesnginxservers .
root@it77-OptiPlex-3050:/home/it77/dockertutorial# docker run -itd -p 8989:80 siesnginxservers
root@it77-OptiPlex-3050:/home/it77/dockertutorial# ufw allow 8989
```

Open browser and put IP/localhost with port number as 8989 and run your web application.

Conclusion: Add conclusion here