

## Experiment No: 12

**AIM:** Creating Docker Volumes

### 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.

### 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
```

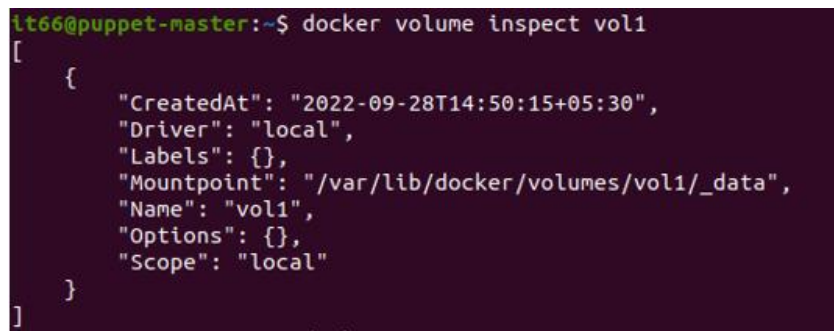
```
DRIVER VOLUME NAME
```

```
local vol1
```

```
lt66@puppet-master:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
lt66@puppet-master:~$ docker volume create vol1
vol1
lt66@puppet-master:~$ docker volume list
DRIVER      VOLUME NAME
local       RDK
local       ayush
local       dk
local       rdk
local       sahil
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"
}
]
```

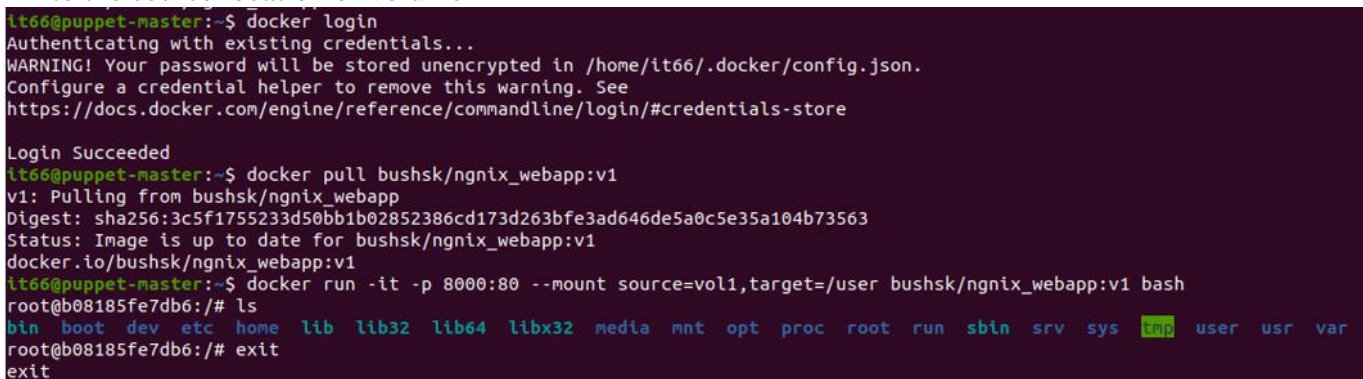


```
it66@puppet-master:~$ docker volume inspect vol1
[
  {
    "CreatedAt": "2022-09-28T14:50:15+05:30",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/vol1/_data",
    "Name": "vol1",
    "Options": {},
    "Scope": "local"
  }
]
```

Check the existing content of a created volume:

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

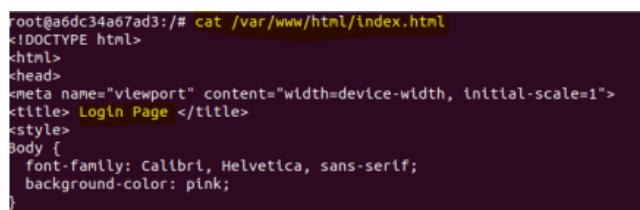
2. Select an existing image and run it using --mount option to attach the target location of container to the source location of volume



```
it66@puppet-master:~$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/it66/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
it66@puppet-master:~$ docker pull bushsk/nginx_webapp:v1
v1: Pulling from bushsk/nginx_webapp
Digest: sha256:3c5f1755233d50bb1b02852386cd173d263bfe3ad646de5a0c5e35a104b73563
Status: Image is up to date for bushsk/nginx_webapp:v1
docker.io/bushsk/nginx_webapp:v1
it66@puppet-master:~$ docker run -it -p 8000:80 --mount source=vol1,target=/user bushsk/nginx_webapp:v1 bash
root@b08185fe7db6:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  user  usr  var
root@b08185fe7db6:/# exit
exit
```

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;
}
```

```
<!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:

```
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:

```
it66@puppet-master:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
it66@puppet-master:~$ docker run -it -p 8000:80 --mount source=vol1,target=/user bushsk/nginx_webapp:v1 bash
root@c6b2d5c5ebd2:/# curl localhost:8989
```

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

The screenshot shows a web browser window with a pink background. At the top, there is a title 'Student Login Form'. Below the title, there is a light blue box containing the login form. The form has two input fields: 'Username' and 'Password'. Below the 'Password' field, there is a green 'Login' button. At the bottom of the light blue box, there is a 'Remember me' checkbox, a green 'Cancel' button, and a 'Forgot password?' link. The rest of the browser window has a pink background.

**Conclusion:** We are able to understand the concept of Docker volume and have successfully created it.