

## **Lab Exercise 3: Working with Docker Volumes**

**Name** – Sujal Bhandari

**SAP\_ID** – 500106865

**Roll\_No** – R2142220181

### **Objective:**

- Learn how to create and manage Docker volumes.
- Understand how Docker volumes can be used to persist data across container restarts.
- Practice mounting Docker volumes to containers.

### **Step 1: Create a Docker Volume**

Create a new Docker volume:

```
docker volume create my_data_volume
```

```
C:\Users\sujal>docker volume create my_data_volume
my_data_volume

C:\Users\sujal>docker volume ls
DRIVER      VOLUME NAME
local       my_data_volume

C:\Users\sujal>
```

This command creates a Docker volume named my\_data\_volume.

Verify that the volume was created:

```
docker volume ls
```

```
C:\Users\sujal>docker volume create my_data_volume
my_data_volume

C:\Users\sujal>docker volume ls
DRIVER      VOLUME NAME
local       my_data_volume

C:\Users\sujal>
```

You should see my\_data\_volume listed among the volumes.

## Step 2: Run a Container with the Volume Mounted

Run an Nginx container with the volume mounted:

```
docker run -d --name my_nginx -v my_data_volume:/usr/share/nginx/html -p
8008:80 nginx
```

This command starts an Nginx container named my\_nginx and mounts the my\_data\_volume volume to the /usr/share/nginx/html directory inside the container.

Verify that the container is running:

```
docker ps
```

You should see my\_nginx listed as one of the running containers.

```
C:\Users\sujal>docker run -d --name my_nginx -v my_data_volume:/usr/share/nginx/html -p 8008:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a2318d6c47ec: Pull complete
095d327c79ae: Pull complete
bbfaa25db775: Pull complete
7bb6fb0cfb2b: Pull complete
0723edc10c17: Pull complete
24b3fdc4d1e3: Pull complete
3122471704d5: Pull complete
Digest: sha256:04ba374043ccd2fc5c593885c0eacddebabd5ca375f9323666f28dfd5a9710e3
Status: Downloaded newer image for nginx:latest
c7a087fe18b4500146fa7fdc3c1a16e8d891caa8d21b422b351db091459756ff

C:\Users\sujal>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
c7a087fe18b4   nginx    "/docker-entrypoint...."   About a minute ago   Up About a minute   0.0.0.0:8008->80/tcp      my_nginx
```

### Step 3: Interact with the Volume

Create a simple HTML file in the volume:

```
docker exec -it my_nginx bash

echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html

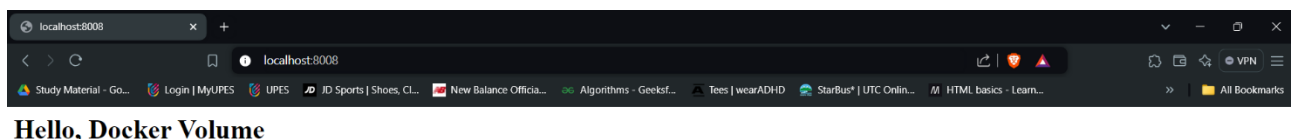
exit
```

```
C:\Users\sujal>docker exec -it my_nginx bash
root@c7a087fe18b4:/# echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html
root@c7a087fe18b4:/# exit
exit

C:\Users\sujal>|
```

This command creates an HTML file inside the `/usr/share/nginx/html` directory, which is backed by `my_data_volume`.

Access the Nginx server to see your file: Open a browser and navigate to `http://localhost:8008`. You should see the message "Hello, Docker Volume!" displayed on the page.



## Step 4: Test Data Persistence

Stop and remove the container:

```
docker stop my_nginx
```

```
docker rm my_nginx
```

```
C:\Users\sujal>docker stop my_nginx  
my_nginx
```

```
C:\Users\sujal>docker rm my_nginx  
my_nginx
```

```
C:\Users\sujal>
```

Run a new Nginx container using the same volume:

```
docker run -d -p 8011:80 -v my_data_volume:/usr/share/nginx/html nginx
```

```
C:\Users\sujal>docker run -d -p 8011:80 -v my_data_volume:/usr/share/nginx/html nginx  
d320d9925b1f2279c7db15e6d7ec87a601656a7fde9096842621b71376ad2398
```

```
C:\Users\sujal>|
```

Access the Nginx server again: Navigate to <http://localhost> in your browser. You should still see the "Hello, Docker Volume!" message, demonstrating that the data persisted across container instances.

Hello, Docker Volume

## Step 5: Clean Up

Stop and remove the container:

```
docker stop new_nginx
```

```
docker rm new_nginx
```

```
C:\Users\sujal>docker stop d320d9925b1f
d320d9925b1f
```

```
C:\Users\sujal>docker rm d320d9925b1f
d320d9925b1f
```

```
C:\Users\sujal>|
```

Remove the Docker volume:

```
docker volume rm my_data_volume
```

Verify that the volume is removed:

```
docker volume ls
```

Ensure that my\_data\_volume is no longer listed.

```
C:\Users\sujal>docker volume rm my_data_volume  
my_data_volume
```

```
C:\Users\sujal>docker volume ls  
DRIVER      VOLUME NAME
```

```
C:\Users\sujal>|
```