

**NAME: PARTHIB HALDER**

**SAP ID: 500126084**

**BATCH: B2**

## **Lab Exercise 2: Working with Docker Volumes**

### **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.

### **Prerequisites:**

- Docker installed on your system.
- Basic understanding of Docker commands and container concepts.

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

Create a new Docker volume:

```
docker volume create my_data_volume
```

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

Verify that the volume was created:

```
docker volume ls
```

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:

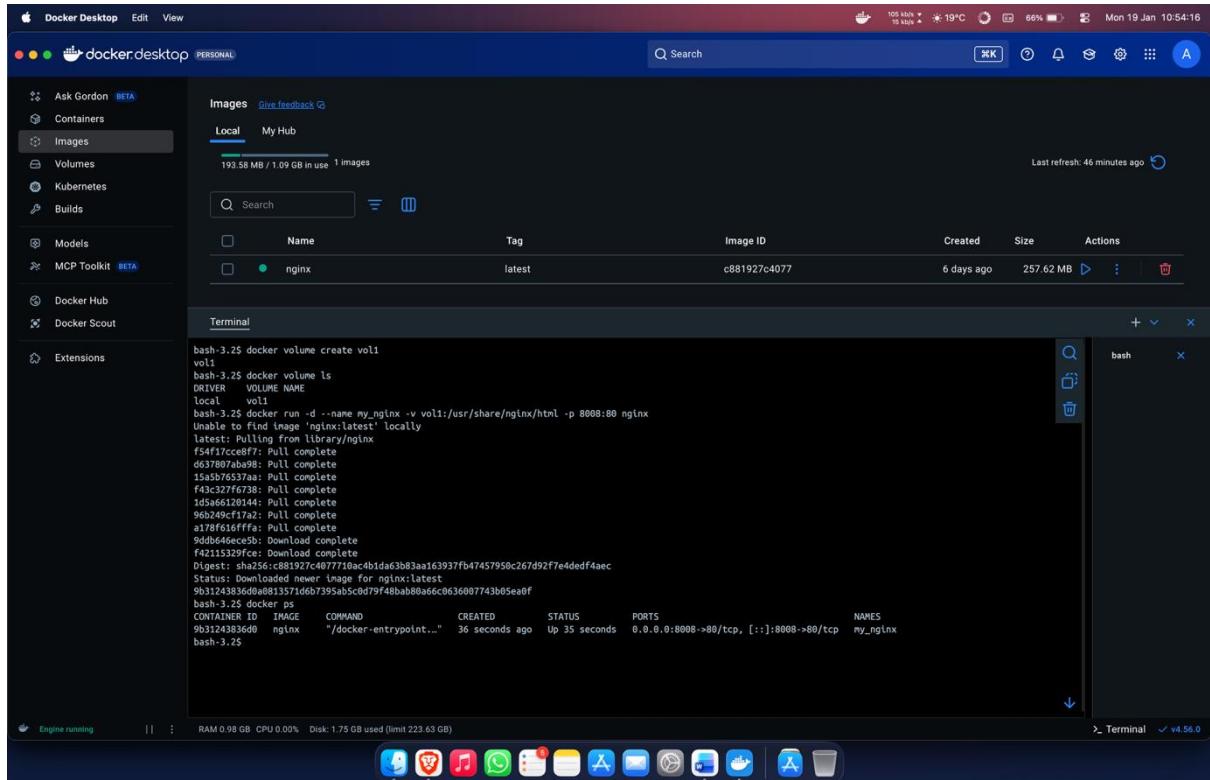
```
docker run -d --name my_nginx -v my_data_volume:/usr/share/nginx/html -p 8008:80 nginx
```

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.



## 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
```

This command creates an HTML file inside the /usr/share/nginx/html directory, which is backed by my\_data\_volume.

```
root@9b31243836d0:/# exit  
exit
```

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
```

Run a new Nginx container using the same volume:

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

```
bash-3.2$ docker stop my_nginx  
my_nginx  
bash-3.2$ docker rm my_nginx  
my_nginx  
bash-3.2$ docker run -d --name my_nginx -v vol1:/usr/share/nginx/html -p 8008:80 nginx  
3a11214f50d00ae5fa27cbf16a086cbf71f0096f084a851d4faf0797c91df547
```

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.

## Step 5: Clean Up

Stop and remove the container:

```
docker stop my_nginx
```

```
docker rm my_nginx
```

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.

```
bash-3.2$ docker stop my_nginx
my_nginx
bash-3.2$ docker rm my_nginx
my_nginx
bash-3.2$ docker volume rm vol1
vol1
bash-3.2$ docker volume ls
DRIVER      VOLUME NAME
bash-3.2$ █
```