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.

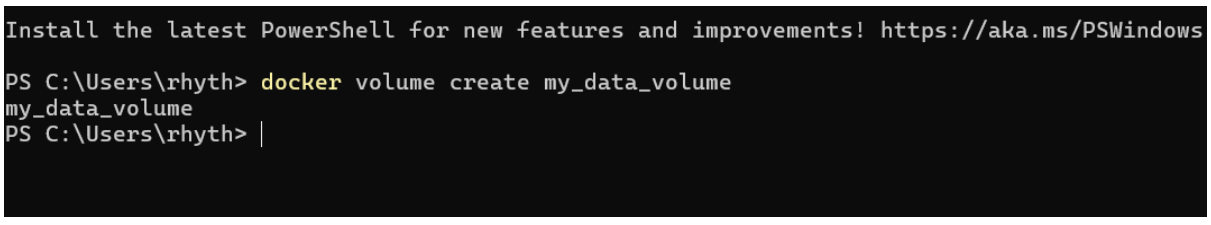
Step 1: Create a Docker Volume

Create a new Docker volume:

```
docker volume create my_data_volume
```

This command creates a Docker volume named my_data_volume.

Output:




```
docker volume ls
```

Output:

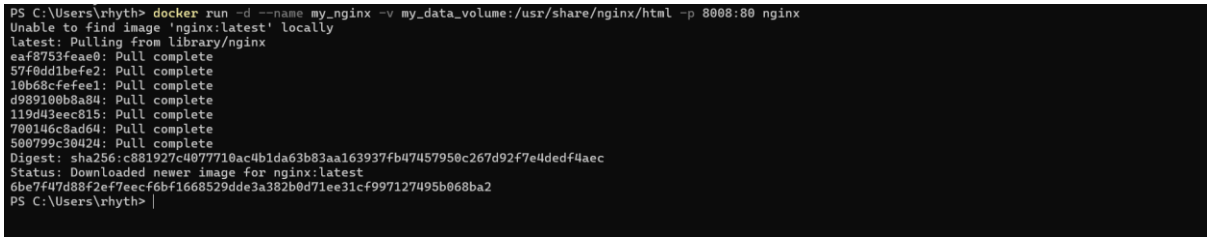
Step 2: Run a Container with the Volume Mounted

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.

Output:



Verify that the container is running:

```
docker ps
```

You should see my_nginx listed as one of the running containers.

Output:

```
PS C:\Users\rhyth> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                       NAMES
6be7f47d88f2   nginx     "/docker-entrypoint.…"   59 seconds ago   Up 59 seconds   0.0.0.0:8008->80/tcp, [::]:8008->80/tcp   my_nginx
PS C:\Users\rhyth>
```

Step 3: Interact with the Volume

Create a simple HTML file in the volume:

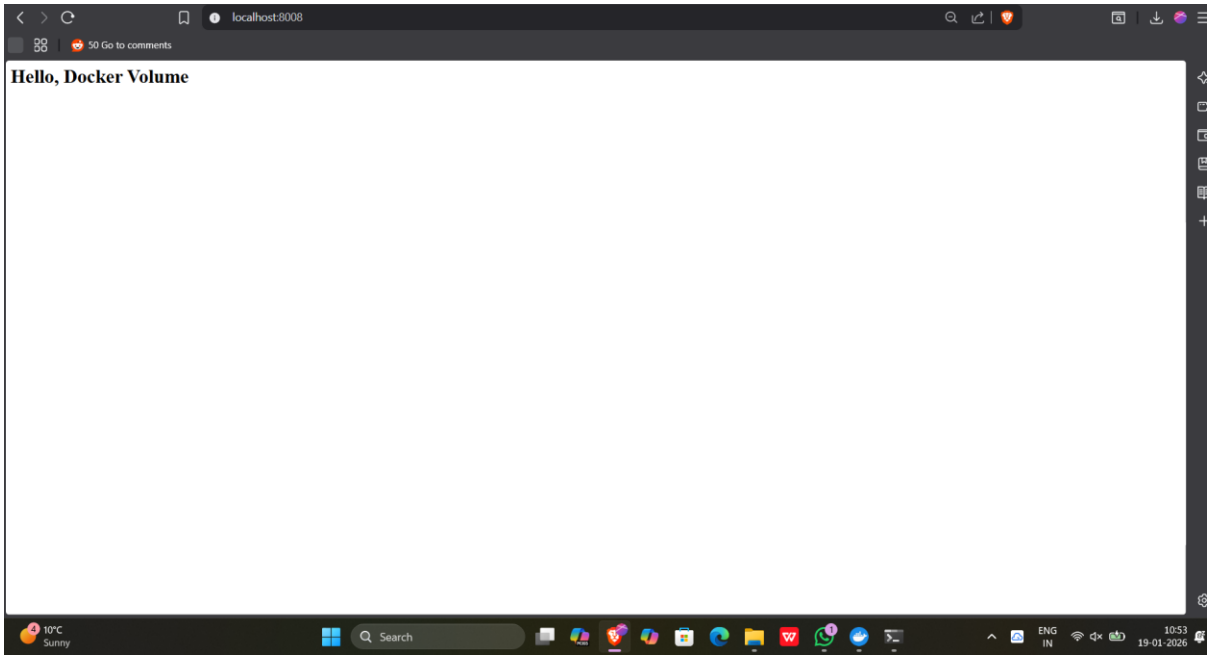
```
docker exec -it my_nginx bash

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.

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.

Output:

```
PS C:\Users\rhyth> docker exec -it my_nginx bash
root@6be7f47d88f2:/# echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html
root@6be7f47d88f2:/# exit
exit
PS C:\Users\rhyth>
```



Step 4: Test Data Persistence

Stop and remove the container:

```
docker stop my_nginx

docker rm my_nginx
```

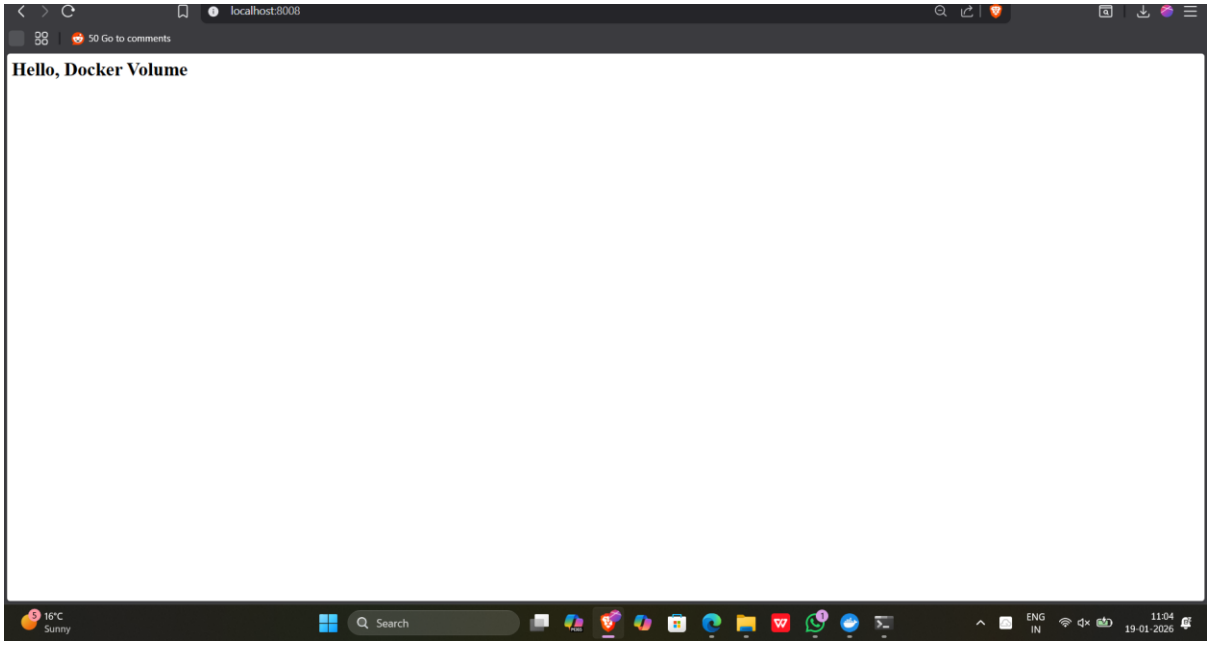
Output:

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
```

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.

Output:



Step 5: Clean Up

Stop and remove the container:

```
docker stop my_nginx

docker rm my_nginx
```

Output:

```
PS C:\Users\rhyth> docker stop my_nginx
my_nginx
PS C:\Users\rhyth> docker rm my_nginx
my_nginx
PS C:\Users\rhyth>
```

Remove the Docker volume:

```
docker volume rm my_data_volume
```

Output:

```
PS C:\Users\rhyth> docker volume rm my_data_volume
my_data_volume
PS C:\Users\rhyth>
```

Verify that the volume is removed:

```
docker volume ls
```

Output:

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

This ensure that my_data_volume is no longer listed.