Lab Exercise 3: 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
```

```
C:\Users\an626>docker volume create my_data_volume
my_data_volume
```

This command creates a Docker volume named my_data_volume.

Verify that the volume was created:

```
docker volume ls
```

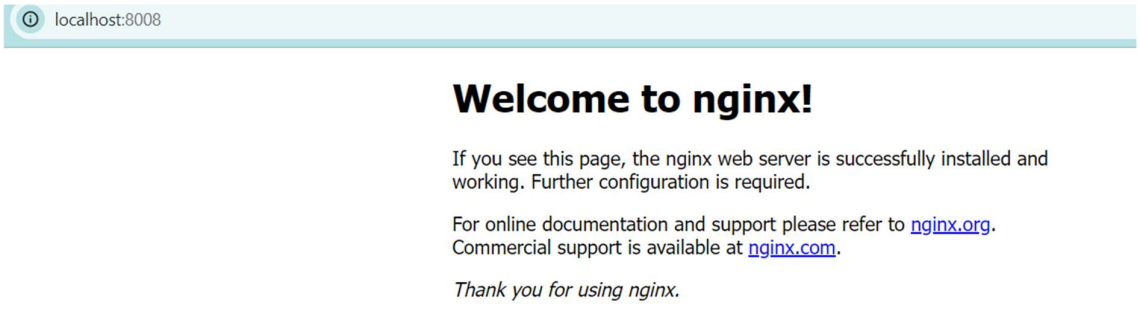
```
C:\Users\an626>docker volume ls
DRIVER    VOLUME NAME
local     f659b3dd47b135cdbdc2c0e233200d6eab53273f5b2fbc02d479b66a64064e89
local     my_data_volume
```

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

```
C:\Users\an626>docker exec -it my_nginx bash
```

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

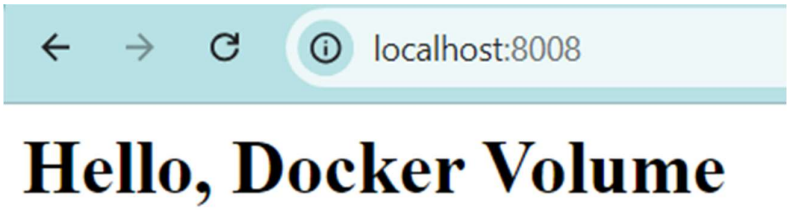
```
root@6813a3c3a221:/# echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html
```

Exit

```
root@6813a3c3a221:/# exit
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.



q

Step 4: Test Data Persistence

Stop and remove the container:

docker stop my_nginx

```
C:\Users\an626>docker stop my_nginx
my_nginx
```

docker rm my_nginx

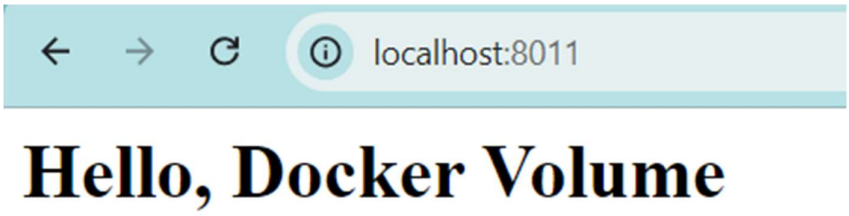
```
C:\Users\an626>docker rm my_nginx
my_nginx
```

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\an626>docker run -d -p 8011:80 -v my_data_volume:/usr/share/nginx/html nginx
bab104d6c71556edbcb0c0464c3d216bd36f4da4bec1f0c975b2330403cdbb88
```

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 new_nginx

```
C:\Users\an626>docker stop new_nginx
new_nginx
```

docker rm new_nginx

```
C:\Users\an626>docker rm new_nginx
new_nginx
```

Remove the Docker volume:

```
docker volume rm my_data_volume
```

```
C:\Users\an626>docker volume rm my_data_volume
my_data_volume
```

Verify that the volume is removed:

```
docker volume ls
```

```
C:\Users\an626>docker volume ls
DRIVER    VOLUME NAME
local     f659b3dd47b135cdbdc2c0e233200d6eab53273f5b2fbc02d479b66a64064e89
```

Ensure that my_data_volume is no longer listed.

```
C:\Users\an626>docker volume ls
DRIVER    VOLUME NAME
```