

Pratik Agrawal

500123601

Devops 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.

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

Verify that the volume was created:

```
docker volume ls
```

You should see my_data_volume listed among the volumes.

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

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.

```
C:\Users\prati>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
bae5a1799a80: Pull complete
4f4efe02d542: Pull complete
7b6cb8ccac7b: Pull complete
0c8d55a45c0d: Pull complete
f73400a233fd: Pull complete
46bf3a120c8e: Pull complete
47cd406a84ef: Pull complete
2e02dba24409: Download complete
a5d78d617315: Download complete
Digest: sha256:341bf0f3ce6c5277d6002cf6e1fb0319fa4252add24ab6a0e262e0056d313208
Status: Downloaded newer image for nginx:latest
c72f0d848cbdaaae4b5f48ab2117a5cbb81920827d0b8039e04cbb1b799f4415
```

Verify that the container is running:

```
docker ps
```

```
C:\Users\prati>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
NAMES
c72f0d848cbd   nginx    "/docker-entrypoint..." 31 seconds ago Up 30 seconds 0.0.0.0:8008->80/tcp, [::]:8008->80/tcp
cp   my_nginx
```

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
```

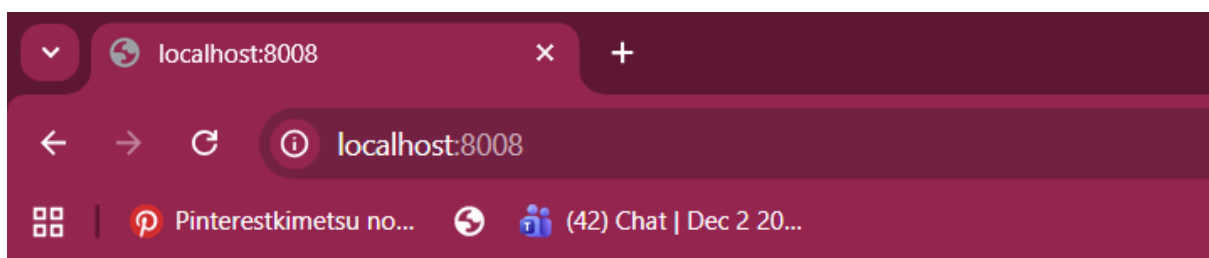
```
echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html
```

```
exit
```

```
C:\Users\prati>docker exec -it my_nginx bash
root@c72f0d848cbd:/# echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html
root@c72f0d848cbd:/# 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.



Hello, Docker Volume

Step 4: Test Data Persistence

Stop and remove the container:

```
docker stop my_nginx
```

```
docker rm my_nginx
```

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

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

```
C:\Users\prati>docker run -d --name my_nginx -v my_data_volume:/usr/share/nginx/html -p 8008:80 nginx
699f23f2c3f439ea0336bfbcb859b2de9519ecb9a5063c74d3ec305754697093c
```

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
```

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

C:\Users\prati>docker rm my_nginx
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.

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

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