

# 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  
  
shaad@shaadiso:~$ docker volume create my_data_volume  
my_data_volume  
shaad@shaadiso:~$
```

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

Verify that the volume was created:

```
docker volume ls  
  
shaad@shaadiso:~$ docker volume ls  
DRIVER      VOLUME NAME  
local      my_data_volume  
shaad@shaadiso:~$
```

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
```

```
shaad@shaadiso:~$ 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
d989100b8a84: Pull complete
10b68cfefee1: Pull complete
700146c8ad64: Pull complete
500799c30424: Pull complete
57f0dd1befe2: Pull complete
eaf8753feae0: Pull complete
e2dd2dbe6277: Download complete
785250c9bf9e: Download complete
Digest: sha256:c881927c4077710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf4aec
Status: Downloaded newer image for nginx:latest
0bc84e55b3366aa1018789d1b5692af9095833db0bcfe39364c2a2f755ea1429
shaad@shaadiso:~$
```

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
```

```
shaad@shaadiso:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
0bc84e55b336   nginx    "/docker-entrypoint..." 34 seconds ago Up 34 seconds 0.0.0.0:8008->80/tcp, [::]:8008->80/tcp  my_nginx
shaad@shaadiso:~$
```

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

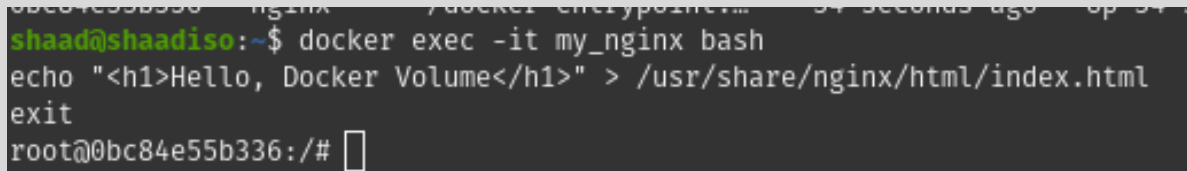
## Step 3: Interact with the Volume

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
```

A terminal window showing the execution of Docker commands. The user 'shaad@shaadiso' runs 'docker exec -it my\_nginx bash' to enter a container. Inside the container, they run 'echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html' to create a file. Finally, they run 'exit' to leave the container. The prompt changes from 'shaad@shaadiso:~\$' to 'root@0bc84e55b336:/#' when inside the container.

```
shaad@shaadiso:~$ docker exec -it my_nginx bash
echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html
exit
root@0bc84e55b336:/#
```

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

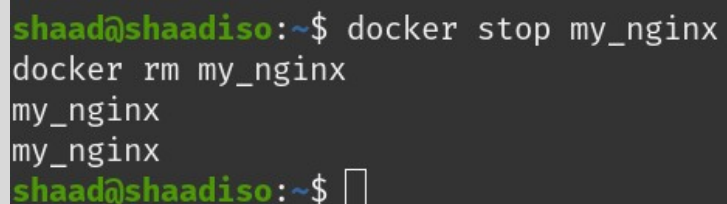
Access the Nginx server to see your file: Open a browser and navigate to `http://localhost:8008`. You should see the message "Hello, Docker Volume!" displayed on the page.

## Step 4: Test Data Persistence

Stop and remove the container:

```
docker stop my_nginx
```

```
docker rm my_nginx
```

A terminal window showing the user 'shaad@shaadiso' running 'docker stop my\_nginx' and 'docker rm my\_nginx'. The container 'my\_nginx' is shown as stopped and then removed. The prompt returns to 'shaad@shaadiso:~\$'.

```
shaad@shaadiso:~$ docker stop my_nginx
my_nginx
shaad@shaadiso:~$ docker rm my_nginx
my_nginx
shaad@shaadiso:~$
```

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
```

```
shaad@shaadiso:~$ docker run -d --name my_nginx -v my_data_volume:/usr/share/nginx/html -p 8008:80 nginx
09a8434e1eaf4cca13e84ec854ce6daebdd38d8b7244278c9ec30af1bde9e4e1
shaad@shaadiso:~$
```

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
```

```
shaad@shaadiso:~$ docker stop my_nginx
my_nginx
shaad@shaadiso:~$ docker rm my_nginx
my_nginx
shaad@shaadiso:~$
```

Remove the Docker volume:

```
docker volume rm my_data_volume
```

```
shaad@shaadiso:~$ docker volume rm my_data_volume
my_data_volume
shaad@shaadiso:~$
```

Verify that the volume is removed:

docker volume ls

```
shaad@shaadiso:~$ docker volume ls
DRIVER      VOLUME NAME
shaad@shaadiso:~$
```

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