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:

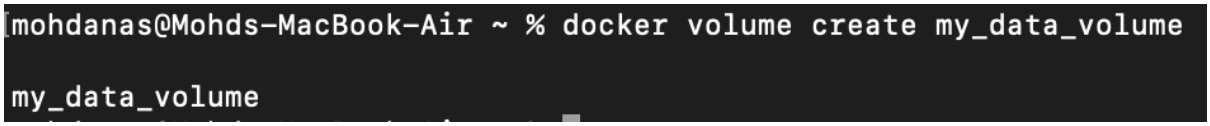
- Docker installed on your system.
- Basic understanding of Docker commands and container concepts.

Step 1: Create a Docker Volume

Create a new Docker volume:

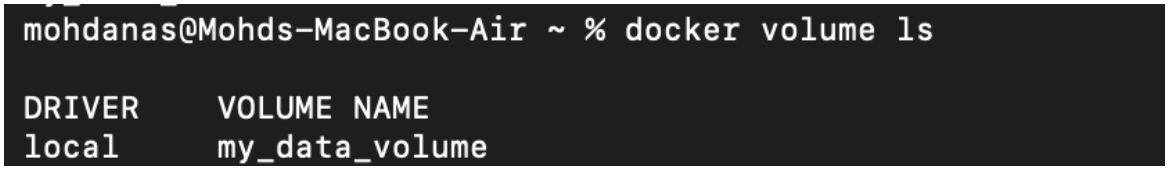
```
docker volume create my_data_volume
```

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.

```
mohdanas@Mohds-MacBook-Air ~ % docker run -d --name my_nginx -v my_data_volume
:/usr/share/nginx/html -p 8008:80 nginx
```

Verify that the container is running:

```
docker ps
```

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

```
mohdanas@Mohds-MacBook-Air ~ % docker ps

CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS
   PORTS                                   NAMES
5742f24bf46b   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes
   0.0.0.0:8008->80/tcp, [::]:8008->80/tcp   my_nginx
```

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
```

Hello, Docker Volume

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
```

```
mohdanas@Mohds-MacBook-Air ~ % docker stop my_nginx

my_nginx
mohdanas@Mohds-MacBook-Air ~ % 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
```

```
mohdanas@Mohds-MacBook-Air ~ % docker run -d --name my_nginx -v my_data_volume
:/usr/share/nginx/html -p 8008:80 nginx
e0d6ce86a9b877be14307e9194f55dc8ddf0994cf7aa1faf261b4a9ef78b45aa
```

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
```

```
mohdanas@Mohds-MacBook-Air ~ % docker stop my_nginx

my_nginx
```

Remove the Docker volume:

```
docker volume rm my_data_volume
```

```
mohdanas@Mohds-MacBook-Air ~ % docker volume rm my_data_volume

my_data_volume
```

Verify that the volume is removed:

```
docker volume ls
```

Ensure that my_data_volume is no longer listed.

```
mohdanas@Mohds-MacBook-Air ~ % docker volume rm my_data_volume

my_data_volume
mohdanas@Mohds-MacBook-Air ~ % docker volume ls

DRIVER    VOLUME NAME
mohdanas@Mohds-MacBook-Air ~ %
```