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:

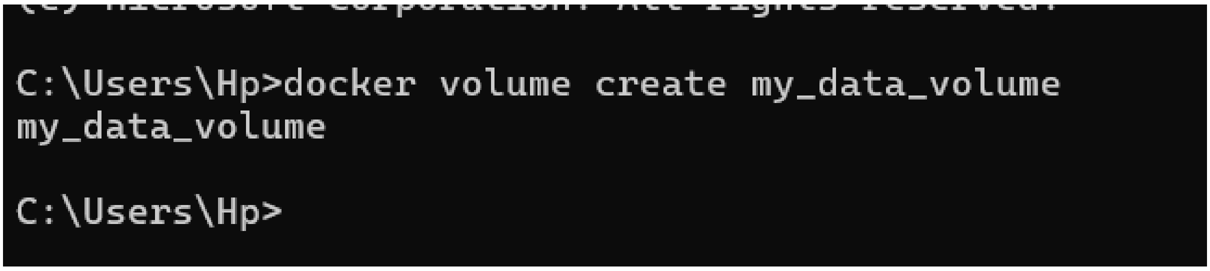
- Docker installed on your system.
- Basic understanding of Docker commands and container concepts.

Step 1: Create a Docker Volume

Create a new Docker volume:

```
docker volume create my_data_volume
```

This command creates a Docker volume named my_data_volume.

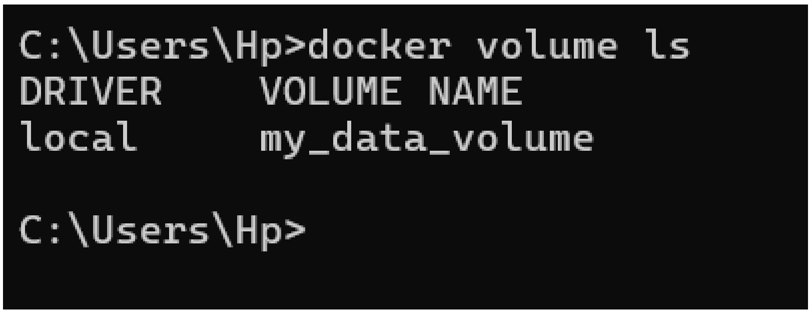


```
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hp>docker volume create my_data_volume
my_data_volume

C:\Users\Hp>
```

Verify that the volume was created:

```
docker volume ls
```

You should see my_data_volume listed among the volumes.



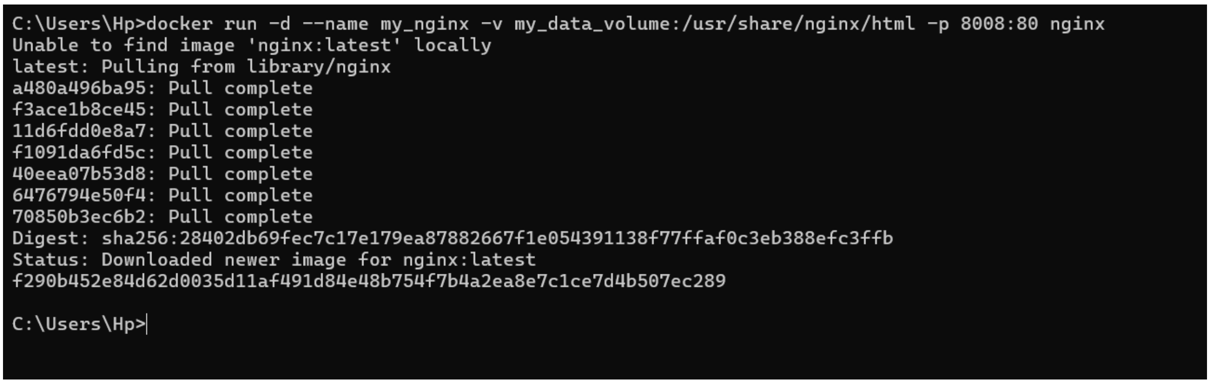
```
C:\Users\Hp>docker volume ls
DRIVER    VOLUME NAME
local     my_data_volume

C:\Users\Hp>
```

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.

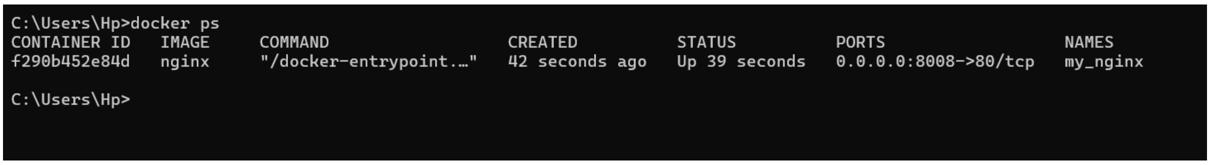


```
C:\Users\Hp>docker run -d --name my_nginx -v my_data_volume:/usr/share/nginx/html -p 8008:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a480a496ba95: Pull complete
f3ace1b8ce45: Pull complete
11d6fdd0e8a7: Pull complete
f1091da6fd5c: Pull complete
40eea07b53d8: Pull complete
6476794e50f4: Pull complete
70850b3ec6b2: Pull complete
Digest: sha256:28402db69fec7c17e179ea87882667f1e054391138f77ffaf0c3eb388efc3ffb
Status: Downloaded newer image for nginx:latest
f290b452e84d62d0035d11af491d84e48b754f7b4a2ea8e7c1ce7d4b507ec289

C:\Users\Hp>
```

Verify that the container is running:

```
docker ps
```

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

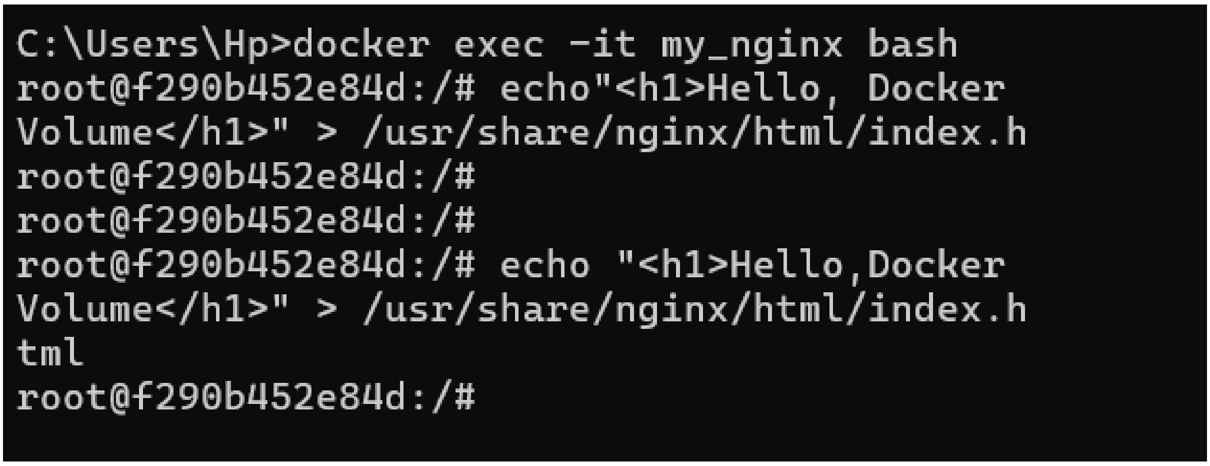


```
C:\Users\Hp>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                  NAMES
f290b452e84d   nginx     "/docker-entrypoint.…"   42 seconds ago   Up 39 seconds   0.0.0.0:8008->80/tcp   my_nginx

C:\Users\Hp>
```

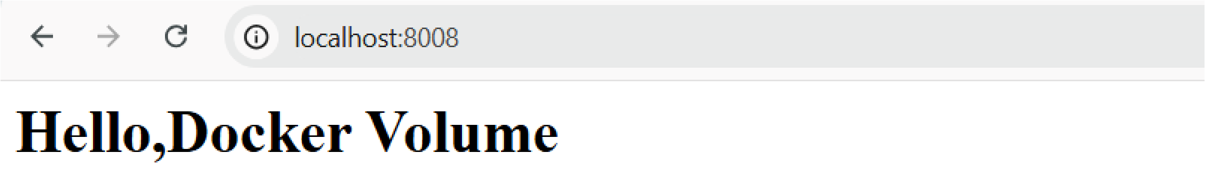
Step 3: Interact with the Volume

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\Hp>docker exec -it my_nginx bash
root@f290b452e84d:/# echo"<h1>Hello, Docker
Volume</h1>" > /usr/share/nginx/html/index.h
root@f290b452e84d:/#
root@f290b452e84d:/#
root@f290b452e84d:/# echo "<h1>Hello,Docker
Volume</h1>" > /usr/share/nginx/html/index.h
tml
root@f290b452e84d:/#
```



localhost:8008

Hello,Docker Volume

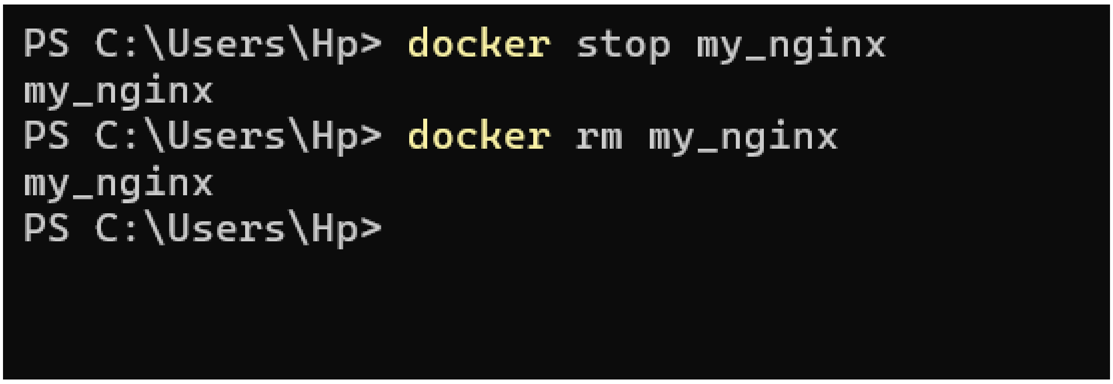
This command creates an HTML file inside the /usr/share/nginx/html directory, which is backed by my_data_volume.

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:

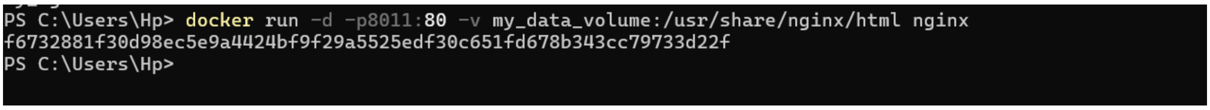
```
docker stop my_nginx

docker rm my_nginx
```



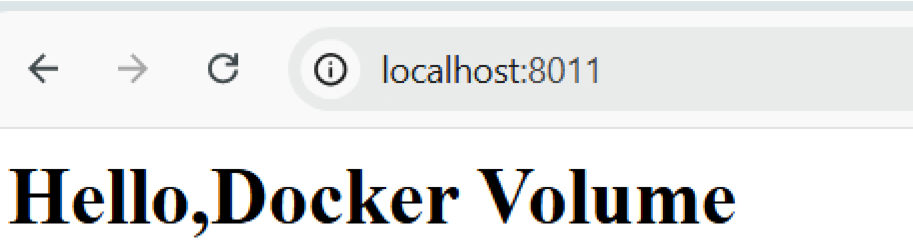
```
PS C:\Users\Hp> docker stop my_nginx
my_nginx
PS C:\Users\Hp> docker rm my_nginx
my_nginx
PS C:\Users\Hp>
```

Run a new Nginx container using the same volume:

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



```
PS C:\Users\Hp> docker run -d -p8011:80 -v my_data_volume:/usr/share/nginx/html nginx
f6732881f30d98ec5e9a4424bf9f29a5525edf30c651fd678b343cc79733d22f
PS C:\Users\Hp>
```



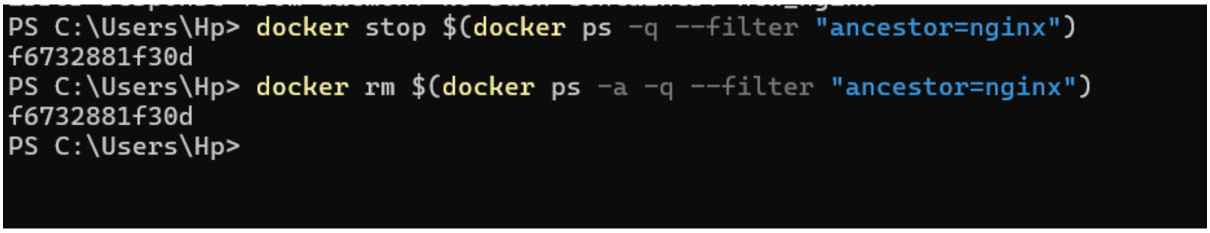
Hello,Docker Volume

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:

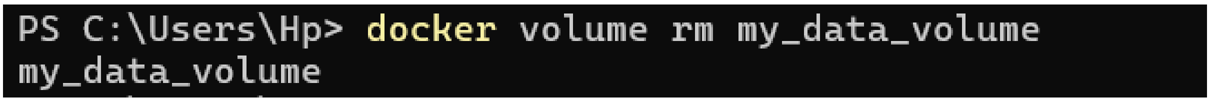
```
docker stop new_nginx
docker rm new_nginx
```



```
PS C:\Users\Hp> docker stop $(docker ps -q --filter "ancestor=nginx")
f6732881f30d
PS C:\Users\Hp> docker rm $(docker ps -a -q --filter "ancestor=nginx")
f6732881f30d
PS C:\Users\Hp>
```

Remove the Docker volume:

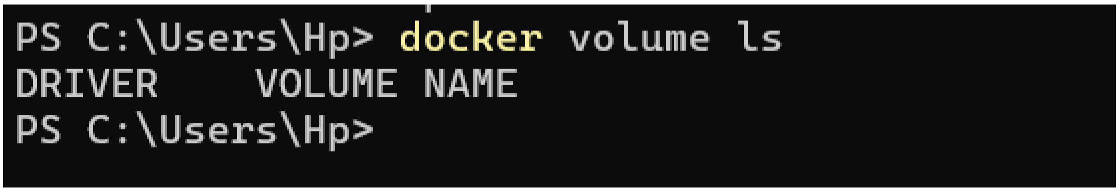
```
docker volume rm my_data_volume
```



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

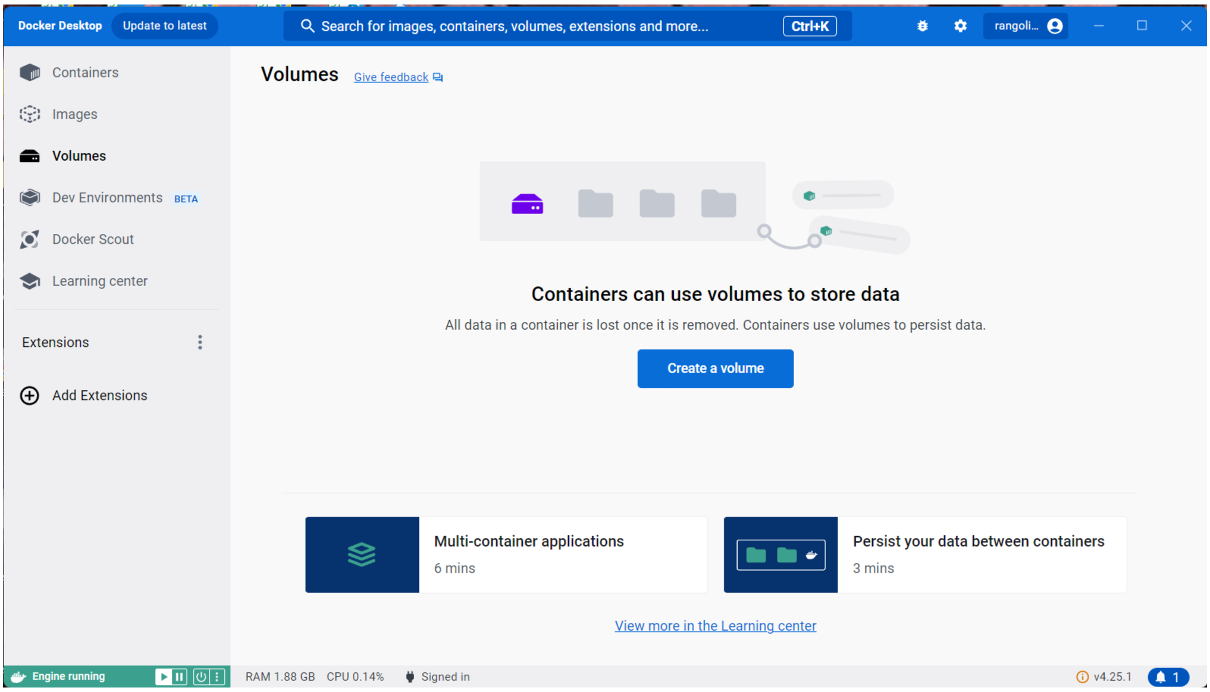
Verify that the volume is removed:

```
docker volume ls
```



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

Ensure that my_data_volume is no longer listed.



Containers can use volumes to store data

All data in a container is lost once it is removed. Containers use volumes to persist data.

Multi-container applications
6 mins

Persist your data between containers
3 mins

also removed from the docker desktop.