

Lab Exercise 2- Working with docker

volumes

Name:- Vansh Bhatt

Sap ID:- 500125395

Batch:- DevOps B1

To:- Hitesh Sharma Sir

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.

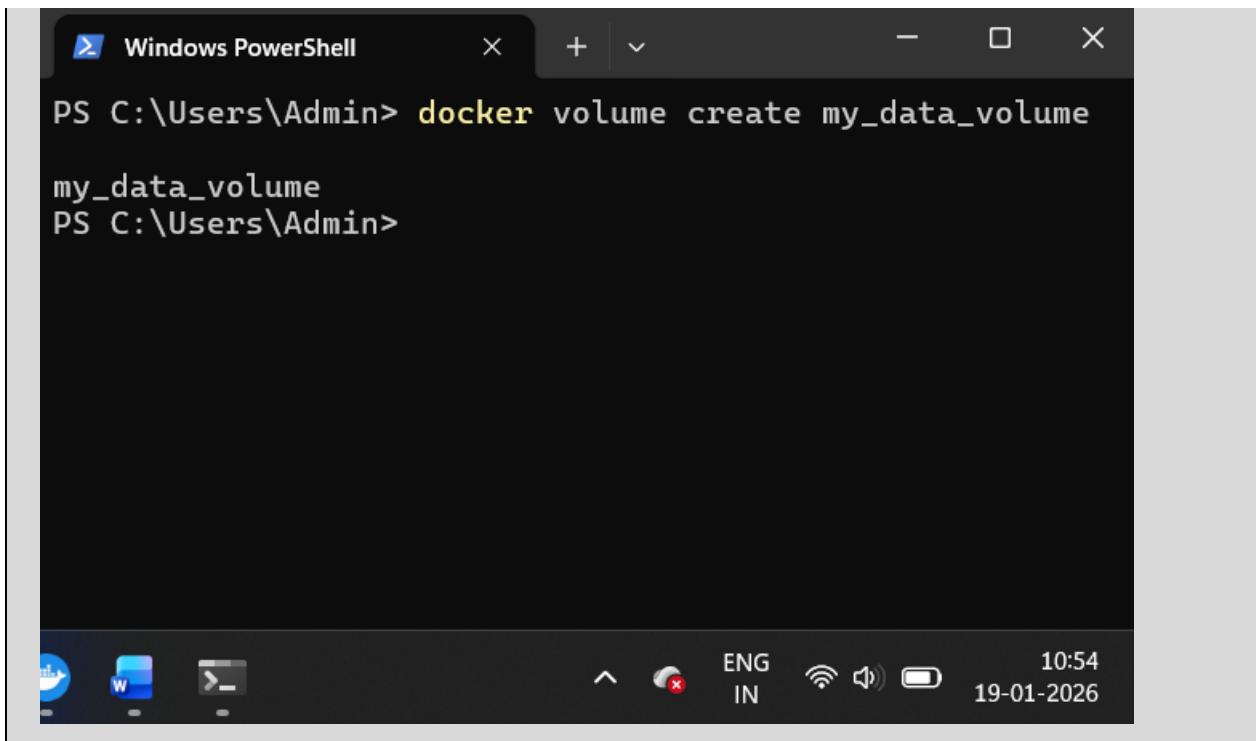
Prerequisites:

- 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
```

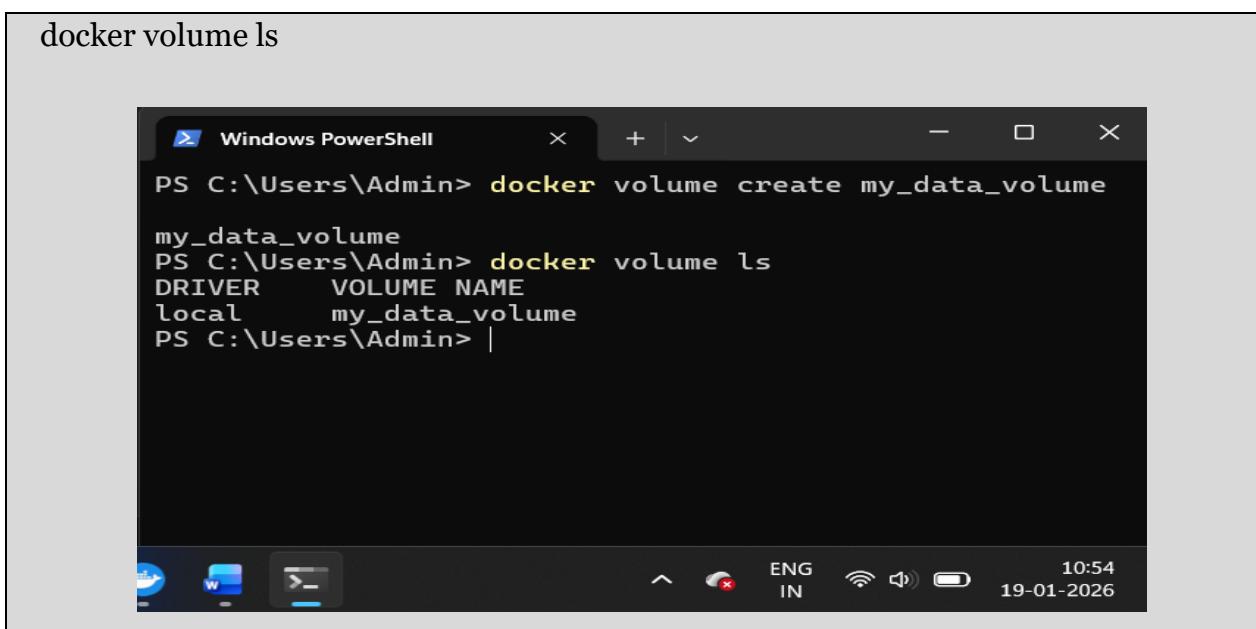


```
PS C:\Users\Admin> docker volume create my_data_volume
my_data_volume
PS C:\Users\Admin>
```

The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command "docker volume create my_data_volume" is run, and the output "my_data_volume" is displayed. The window has standard operating system icons at the bottom.

This command creates a Docker volume named my_data_volume.

Verify that the volume was created:



```
PS C:\Users\Admin> docker volume ls
```

```
PS C:\Users\Admin> docker volume create my_data_volume
my_data_volume
PS C:\Users\Admin> docker volume ls
DRIVER      VOLUME NAME
local      my_data_volume
PS C:\Users\Admin>
```

The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command "docker volume ls" is run, and the output shows a single volume named "my_data_volume" under the "local" driver. The window has standard operating system icons at the bottom.

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
```

```
PS C:\Users\Admin> docker run -d --name vb_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
119d43eecc815: Pull complete
700146c8ad64: Pull complete
d989100b8a84: Pull complete
500799c30424: Pull complete
10b68cfeef1: Pull complete
57f0dd1befe2: Pull complete
eaf8753feae0: Pull complete
Digest: sha256:c881927c4877710ac4b1da63b83aa163937fb47457950c267d92f7e4dedf4aec
Status: Downloaded newer image for nginx:latest
295c2d910bb04e1e9b2df028296acf5f9071c3139f7d9c1408b88b76ffa577
PS C:\Users\Admin>
```

PS C:\Users\Admin>

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
```

```
PS C:\Users\Admin> docker ps
CONTAINER ID        IMAGE       COMMAND
295c2d910bb04e1e9b2df028296acf5f9071c3139f7d9c1408b88b76ffa577   "docker-entrypoint..."
PS C:\Users\Admin>
```

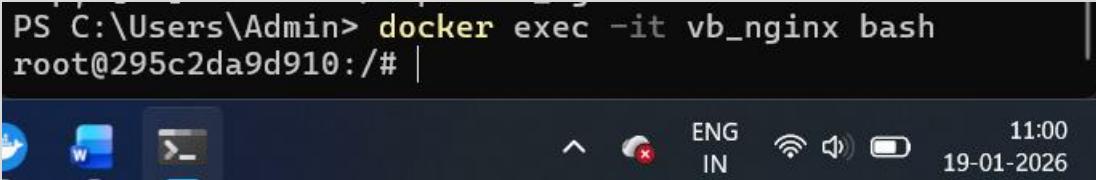
```
PS C:\Users\Admin>
```

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

Step 3: Interact with the Volume

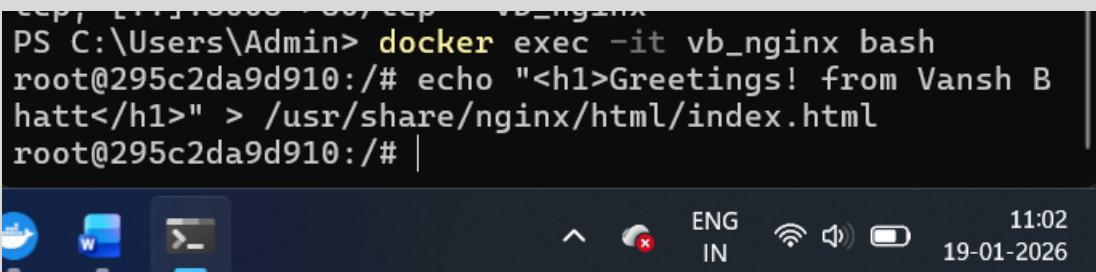
Create a simple HTML file in the volume:

```
docker exec -it my_nginx bash
```



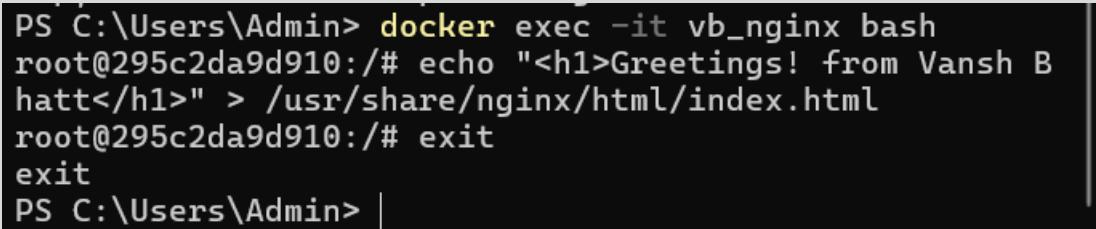
```
PS C:\Users\Admin> docker exec -it my_nginx bash
root@295c2da9d910:/# |
```

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



```
PS C:\Users\Admin> docker exec -it my_nginx bash
root@295c2da9d910:/# echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html
root@295c2da9d910:/# |
```

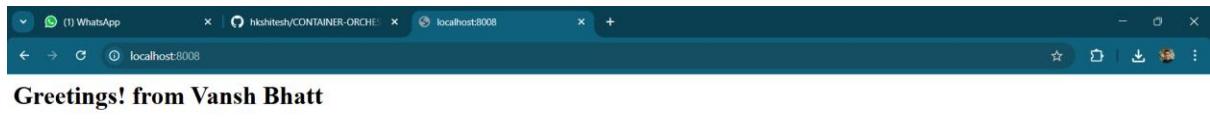
Exit



```
PS C:\Users\Admin> docker exec -it my_nginx bash
root@295c2da9d910:/# echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html
root@295c2da9d910:/# exit
exit
PS C:\Users\Admin> |
```

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:

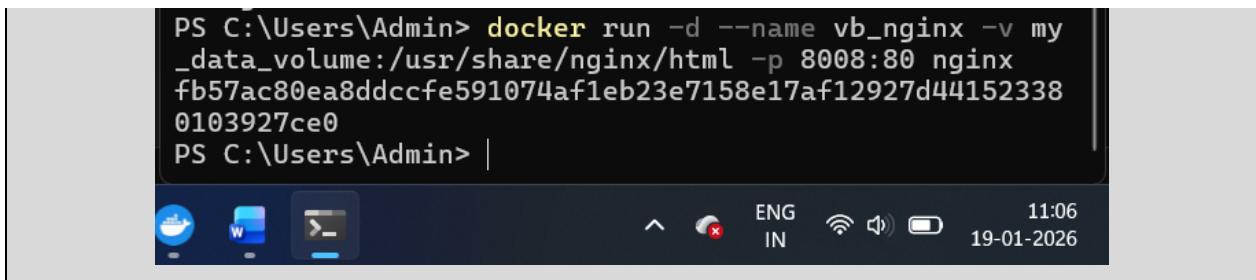
```
docker stop my_nginx
```

```
docker rm my_nginx
```

```
PS C:\Users\Admin> docker stop vb_nginx
vb_nginx
PS C:\Users\Admin> docker rm vb_nginx
vb_nginx
PS C:\Users\Admin>
```

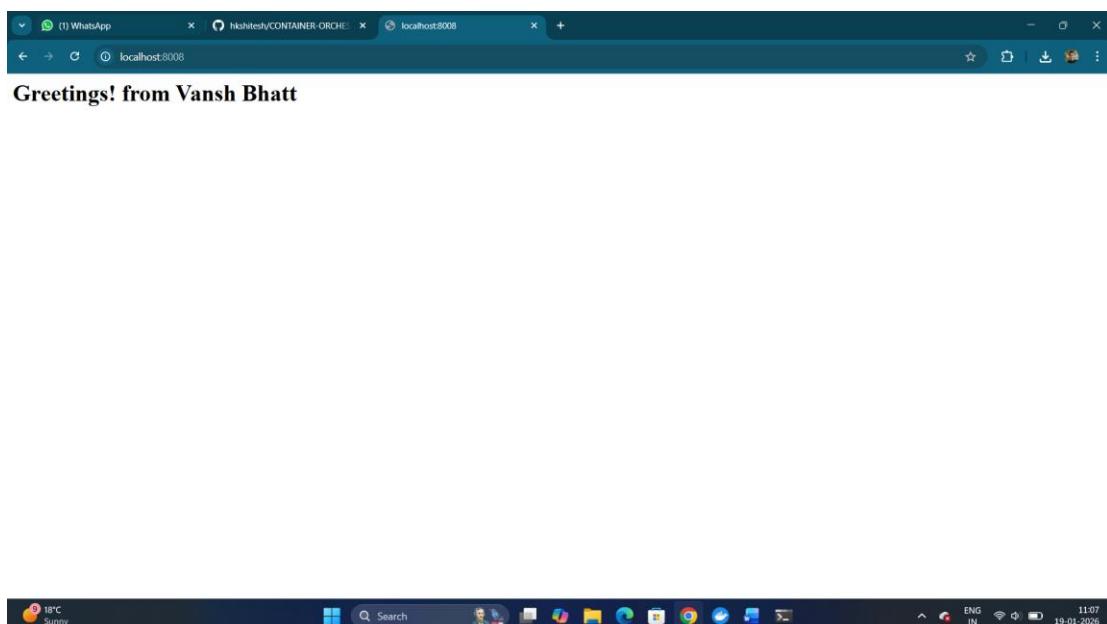
Run a new Nginx container using the same volume:

```
docker run -d --name my_nginx -v my_data_volume:/usr/share/nginx/html -p
8008:80 nginx
```



```
PS C:\Users\Admin> docker run -d --name vb_nginx -v my_data_volume:/usr/share/nginx/html -p 8008:80 nginx
fb57ac80ea8ddccfe591074af1eb23e7158e17af12927d44152338
0103927ce0
PS C:\Users\Admin>
```

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.

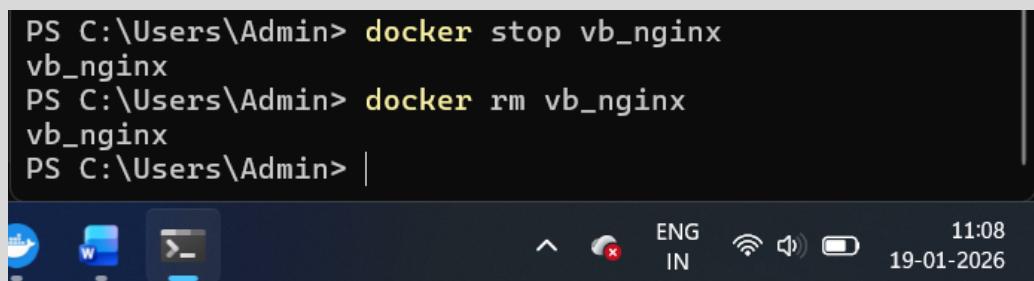


Step 5: Clean Up

Stop and remove the container:

```
docker stop my_nginx
```

```
docker rm my_nginx
```



```
PS C:\Users\Admin> docker stop vb_nginx
vb_nginx
PS C:\Users\Admin> docker rm vb_nginx
vb_nginx
PS C:\Users\Admin>
```

Remove the Docker volume:

```
docker volume rm my_data_volume
```

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



Verify that the volume is removed:

```
docker volume ls
```

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



Ensure that my_data_volume is no longer listed.

Thank You