

Name: Rajat Kamboj

SAP ID: 500105058

Roll no. R2142220140

## 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.

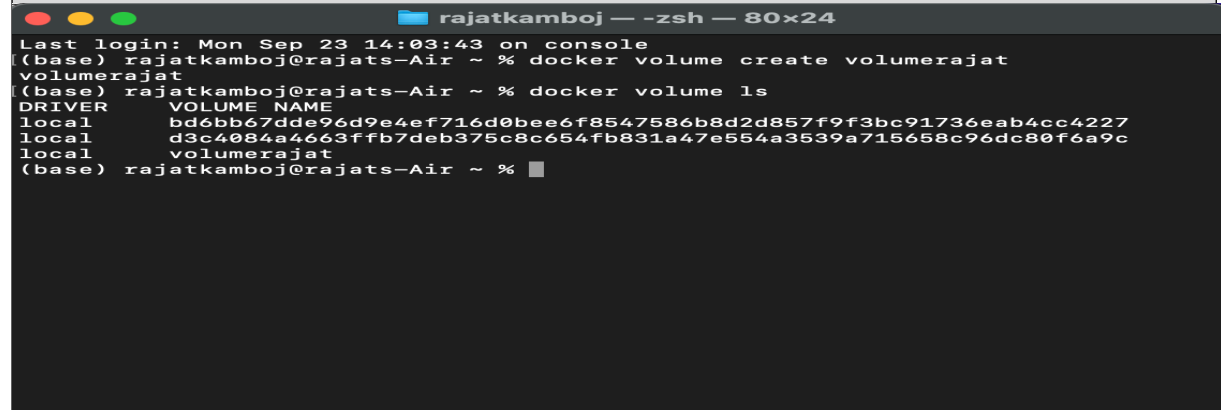
### Prerequisites:

- 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
```



The terminal screenshot shows a user named 'rajatkamboj' in a zsh shell. The prompt is '(base) rajatkamboj@rajats-Air ~ %'. The user enters 'docker volume create volumerajat'. The output is 'volumerajat'. The user then enters 'docker volume ls'. The output is a table with two columns: 'DRIVER' and 'VOLUME NAME'. The table lists three volumes: 'local' with ID 'bd6bb67dde96d9e4ef716d0bee6f8547586b8d2d857f9f3bc91736eab4cc4227', 'local' with ID 'd3c4084a4663fffb7deb375c8c654fb831a47e554a3539a715658c96dc80f6a9c', and 'local' with ID 'volumerajat'.

DRIVER	VOLUME NAME
local	bd6bb67dde96d9e4ef716d0bee6f8547586b8d2d857f9f3bc91736eab4cc4227
local	d3c4084a4663fffb7deb375c8c654fb831a47e554a3539a715658c96dc80f6a9c
local	volumerajat

Docker volume ls #(to see the list of volumes )

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

Verify that the volume was created:

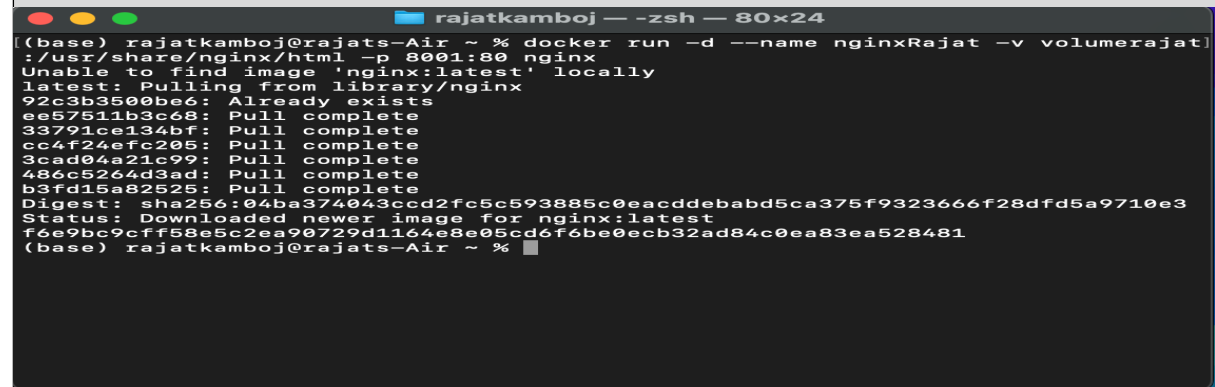
```
docker volume ls
```

You should see my\_data\_volume listed among the volumes.

## Step 2: Run a Container with the Volume Mounted

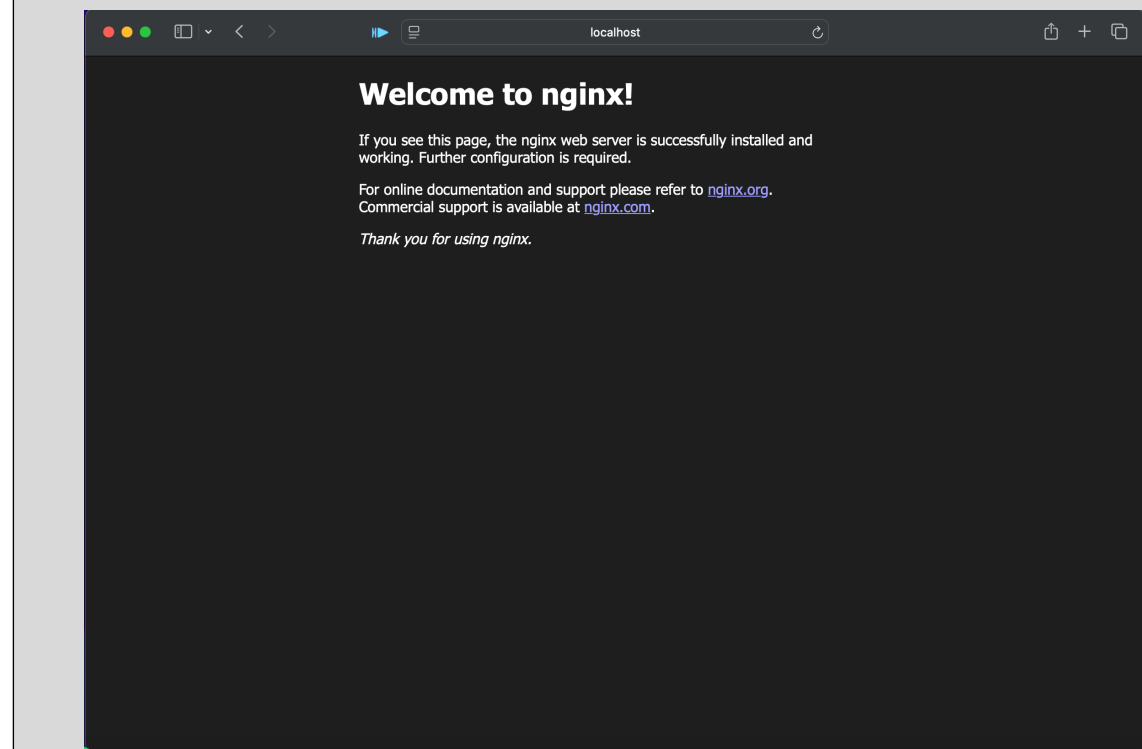
Run an Nginx container with the volume mounted:

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



```
rajatkamboj — zsh — 80x24
(base) rajatkamboj@rajats-Air ~ % docker run -d --name nginxRajat -v volumerajat:/usr/share/nginx/html -p 8001:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
92c3b3500be6: Already exists
ee57511b3c68: Pull complete
33791ce134bf: Pull complete
cc4f24efc205: Pull complete
3cad04a21c99: Pull complete
486c5264d3ad: Pull complete
b3fd15a82525: Pull complete
Digest: sha256:04ba374043ccd2fc5c593885c0eacddebabd5ca375f9323666f28dfd5a9710e3
Status: Downloaded newer image for nginx:latest
f6e9bc9cfff58e8c2ea907229d1164e8e05cd6f6be0ecb32ad84c0ea83ea528481
(base) rajatkamboj@rajats-Air ~ %
```

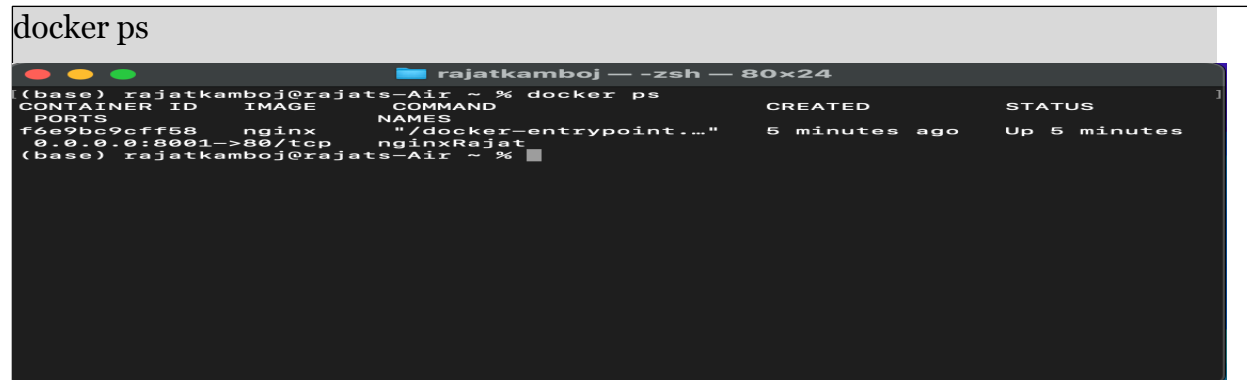
Nginx container running on port 8001



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
```



CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
f6e9bc9cff58	nginx	"/docker-entrypoint..."	5 minutes ago	Up 5 minutes

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

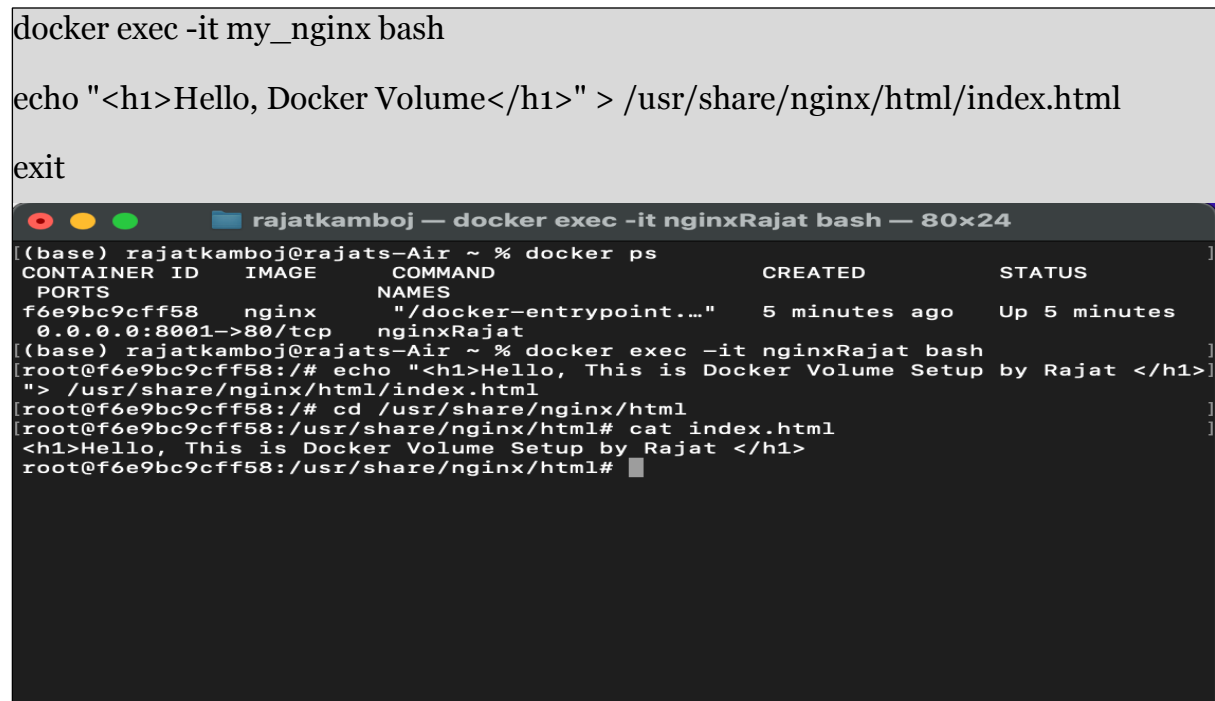
### Step 3: Interact with the Volume

Create a simple HTML file in the volume:

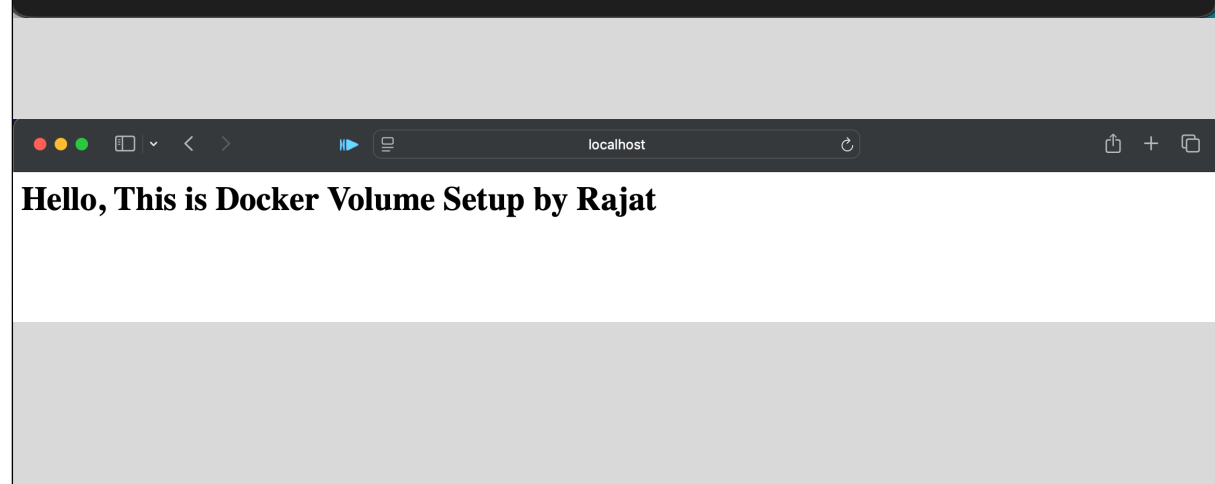
```
docker exec -it my_nginx bash
```

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

```
exit
```



```
[(base) rajatkamboj@rajats-Air ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS
f6e9bc9cff58   nginx    "/docker-entrypoint..." 5 minutes ago  Up 5 minutes
0.0.0.0:8001->80/tcp    nginxRajat
(base) rajatkamboj@rajats-Air ~ % docker exec -it nginxRajat bash
root@f6e9bc9cff58:/# echo "<h1>Hello, This is Docker Volume Setup by Rajat </h1>" > /usr/share/nginx/html/index.html
root@f6e9bc9cff58:/# cd /usr/share/nginx/html
root@f6e9bc9cff58:/usr/share/nginx/html# cat index.html
<h1>Hello, This is Docker Volume Setup by Rajat </h1>
root@f6e9bc9cff58:/usr/share/nginx/html#
```



localhost

**Hello, This is Docker Volume Setup by Rajat**

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
```

```
Last login: Wed Sep 25 12:02:24 on ttys000
(base) rajatkamboj@rajats-Air ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS         NAMES
f6e9bc9cff58   nginx    "/docker-entrypoint..." 16 minutes ago Up 16 minutes
0.0.0.0:8001->80/tcp    nginxRajat
(base) rajatkamboj@rajats-Air ~ % docker stop nginxRajat
nginxRajat
(base) rajatkamboj@rajats-Air ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS         NAMES
(base) rajatkamboj@rajats-Air ~ %
```

```
docker rm my_nginx
```

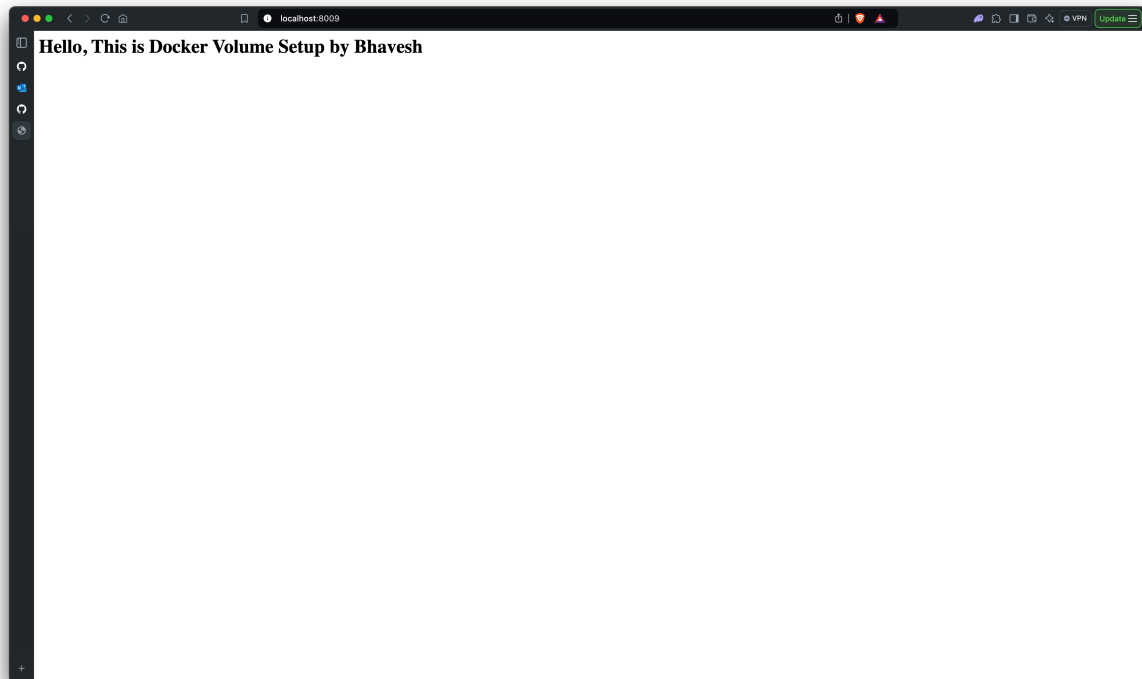
```
Last login: Wed Sep 25 12:02:24 on ttys000
(base) rajatkamboj@rajats-Air ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS         NAMES
f6e9bc9cff58   nginx    "/docker-entrypoint..." 16 minutes ago Up 16 minutes
0.0.0.0:8001->80/tcp    nginxRajat
(base) rajatkamboj@rajats-Air ~ % docker stop nginxRajat
nginxRajat
(base) rajatkamboj@rajats-Air ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS         NAMES
(base) rajatkamboj@rajats-Air ~ %
```

Run a new Nginx container using the same volume:

```
(base) ~ docker run -d -p 8009:80 -v volumeBhavesh:/usr/share/nginx/html nginx
3ad3ed772176bdf154ef402b5bc425c0a45516ca5f66578feff5dae171041e2f
(base) ~ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS         NAMES
3ad3ed772176   nginx    "/docker-entrypoint..." 3 seconds ago  Up 2 seconds  0.0.0.0:8009->80/tcp    youthful_newton
(base) ~ docker exec -it 3ad3ed772176 bash
root@3ad3ed772176:/# cd /usr/share/nginx/
root@3ad3ed772176:/usr/share/nginx# cd html/
root@3ad3ed772176:/usr/share/nginx/html# cat index.html
<h1>Hello, This is Docker Volume Setup by Bhavesh </h1>
root@3ad3ed772176:/usr/share/nginx/html#
```

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

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:

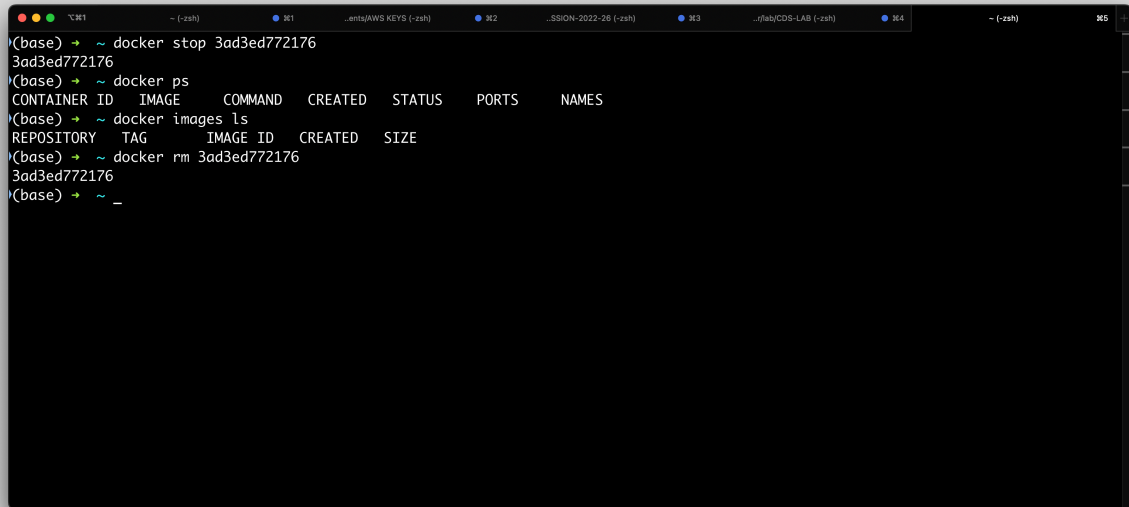
```
docker stop new_nginx  
docker rm new_nginx
```

Remove the Docker volume:

```
docker volume rm my_data_volume
```

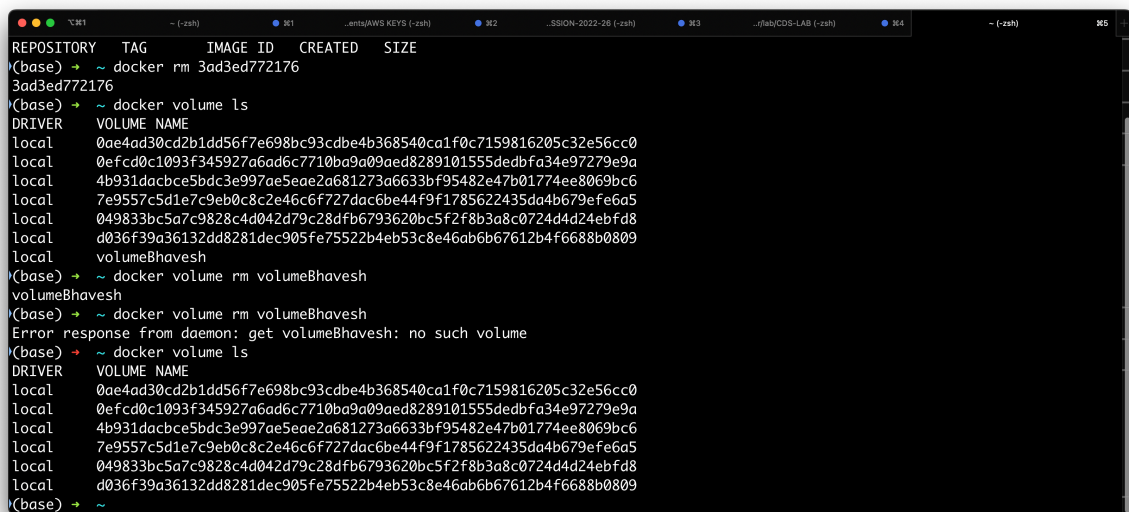
Verify that the volume is removed:

docker volume ls



```
(base) ➔ ~ docker stop 3ad3ed772176
3ad3ed772176
(base) ➔ ~ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
(base) ➔ ~ docker images ls
REPOSITORY    TAG        IMAGE ID      CREATED      SIZE
(base) ➔ ~ docker rm 3ad3ed772176
3ad3ed772176
(base) ➔ ~ _
```

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



```
REPOSITORY    TAG        IMAGE ID      CREATED      SIZE
(base) ➔ ~ docker rm 3ad3ed772176
3ad3ed772176
(base) ➔ ~ docker volume ls
DRIVER        VOLUME NAME
local         0ae4ad30cd2b1dd56f7e698bc93cdbc4b368540ca1f0c7159816205c32e56cc0
local         0efcd0c1093f345927a6ad6c7710ba9a09aed8289101555dedbfa34e97279e9a
local         4b931dacbce5bdc3e997ae5eae2a681273a6633bf95482e47b01774ee8069bc6
local         7e9557c5d1e7c9eb0c8c2e46c6f727dac6be44f9f1785622435da4b679efe6a5
local         049833bc5a7c9828c4d042d79c28dfb6793620bc5f2f8b3a8c0724d4d24ebfd8
local         d036f39a36132dd8281dec905fe75522b4eb53c8e46ab6b67612b4f6688b0809
local         volumeBhavesh
(base) ➔ ~ docker volume rm volumeBhavesh
volumeBhavesh
(base) ➔ ~ docker volume rm volumeBhavesh
Error response from daemon: get volumeBhavesh: no such volume
(base) ➔ ~ docker volume ls
DRIVER        VOLUME NAME
local         0ae4ad30cd2b1dd56f7e698bc93cdbc4b368540ca1f0c7159816205c32e56cc0
local         0efcd0c1093f345927a6ad6c7710ba9a09aed8289101555dedbfa34e97279e9a
local         4b931dacbce5bdc3e997ae5eae2a681273a6633bf95482e47b01774ee8069bc6
local         7e9557c5d1e7c9eb0c8c2e46c6f727dac6be44f9f1785622435da4b679efe6a5
local         049833bc5a7c9828c4d042d79c28dfb6793620bc5f2f8b3a8c0724d4d24ebfd8
local         d036f39a36132dd8281dec905fe75522b4eb53c8e46ab6b67612b4f6688b0809
(base) ➔ ~ _
```