

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 screenshot shows a terminal window with a light gray background. At the top, there is a header bar with a progress bar indicating '5 mins' and some icons. Below the header, the word 'Terminal' is underlined. The main area of the terminal contains the command and its output:
/dev/fd/12:18: command not found: compdef
dhruvchaubey@Dhruv-MBP ~ % docker volume create my_data_volume
my_data_volume
dhruvchaubey@Dhruv-MBP ~ %

At the bottom of the terminal, there is a status bar showing 'RAM 0.51 GB CPU ---% Disk 58.38 GB avail. of 62.67 GB'. To the right of the status bar, there is a 'BETA' link and a 'Terminal' button. On the far right edge of the terminal window, there are several small icons: a magnifying glass, a square with a circle, a trash can, and a downward arrow.

This command creates a Docker volume named my_data_volume.

Verify that the volume was created:

```
docker volume ls
```

The screenshot shows a terminal window with a light gray background. The interface is similar to the previous one, with a header bar, the word 'Terminal' underlined, and a status bar at the bottom. The main terminal area shows the command and its output:
/dev/fd/12:18: command not found: compdef
dhruvchaubey@Dhruv-MBP ~ % docker volume create my_data_volume
my_data_volume
dhruvchaubey@Dhruv-MBP ~ % docker volume ls
DRIVER VOLUME NAME
local my_data_volume
dhruvchaubey@Dhruv-MBP ~ %

On the right side of the terminal window, there is a vertical toolbar with four icons: a magnifying glass, a square with a circle, a trash can, and a downward arrow. At the bottom right of the terminal window, there is a 'BETA' link, a 'Terminal' button, and a 'New' button with a red border.

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
```



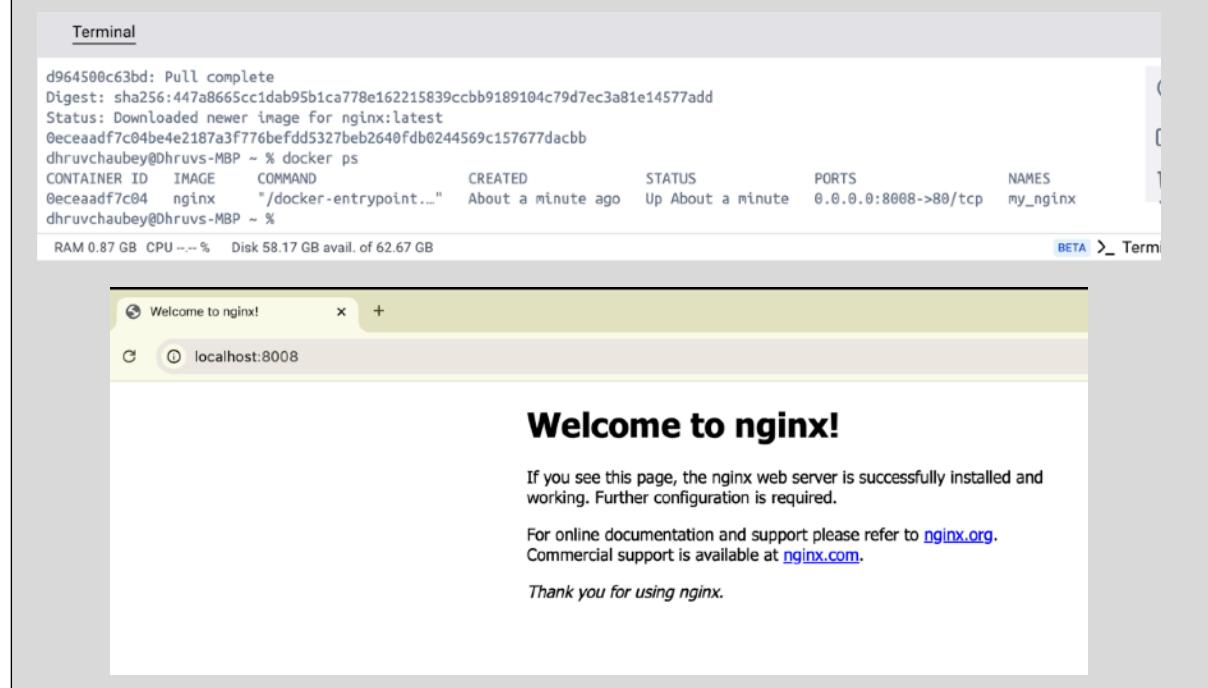
The terminal window shows the command being run:

```
Terminal
DRIVER      VOLUME NAME
local      my_data_volume
dhruvchaubey@Dhruv-MBP ~ % 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
aafdbc30c84e: Pull complete
c28fc33dc48c: Pull complete
08fb08230766: Pull complete
d2983a84b0c4: Pull complete
9f4e03394720: Pull complete
522e88f665e7: Pull complete
d964500c63bd: Pull complete
Digest: sha256:447a8665cc1dab95b1ca778e162215839ccb9189104c79d7ec3a81e14577add
Status: Downloaded newer image for nginx:latest
0eceaaadf7c04be4e2187a3f776befdd5327beb2640fdb0244569c157677dacbb
dhruvchaubey@Dhruv-MBP ~ %
```

Verify that the container is running:

```
docker ps
```



The terminal window shows the container is running:

```
Terminal
d964500c63bd: Pull complete
Digest: sha256:447a8665cc1dab95b1ca778e162215839ccb9189104c79d7ec3a81e14577add
Status: Downloaded newer image for nginx:latest
0eceaaadf7c04be4e2187a3f776befdd5327beb2640fdb0244569c157677dacbb
dhruvchaubey@Dhruv-MBP ~ % docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED      STATUS      PORTS          NAMES
0eceaaadf7c04   nginx    "/docker-entrypoint..."   About a minute ago   Up About a minute   0.0.0.0:8008->80/tcp   my_nginx
dhruvchaubey@Dhruv-MBP ~ %
RAM 0.87 GB CPU ... % Disk 58.17 GB avail. of 62.67 GB
```

The browser screenshot shows the Nginx welcome page at localhost:8008:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

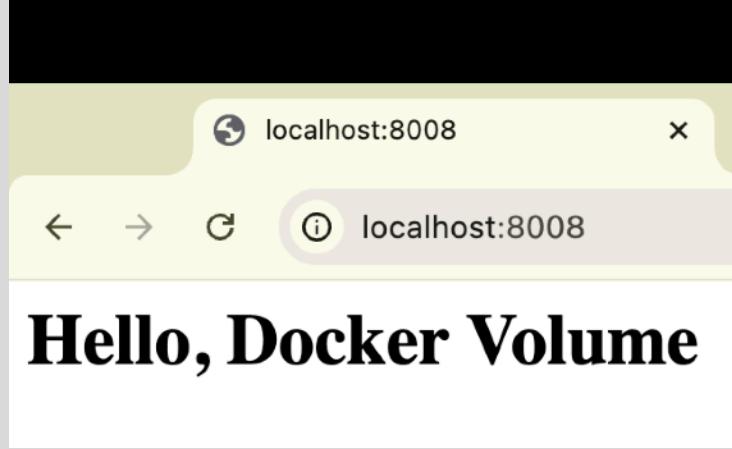
Step 3: Interact with the Volume

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
```

Terminal

```
desktop.ini  
dhruvchaubey@Dhruv-MBP Downloads % mkdir usr  
dhruvchaubey@Dhruv-MBP Downloads % cd usr  
dhruvchaubey@Dhruv-MBP usr % mkdir share  
dhruvchaubey@Dhruv-MBP usr % cd share  
dhruvchaubey@Dhruv-MBP share % mkdir nginx  
dhruvchaubey@Dhruv-MBP share % cd nginx  
dhruvchaubey@Dhruv-MBP nginx % docker exec -it my_nginx bash  
echo "<h1>Hello, Docker Volume!</h1>" > /usr/share/nginx/html/index.html  
exit  
  
zsh: event not found: </h1>  
dhruvchaubey@Dhruv-MBP nginx % docker exec -it my_nginx bash  
echo "<h1>Hello, Docker Volume</h1>" > /usr/share/nginx/html/index.html  
exit  
root@0eceaadf7c04:/#
```



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

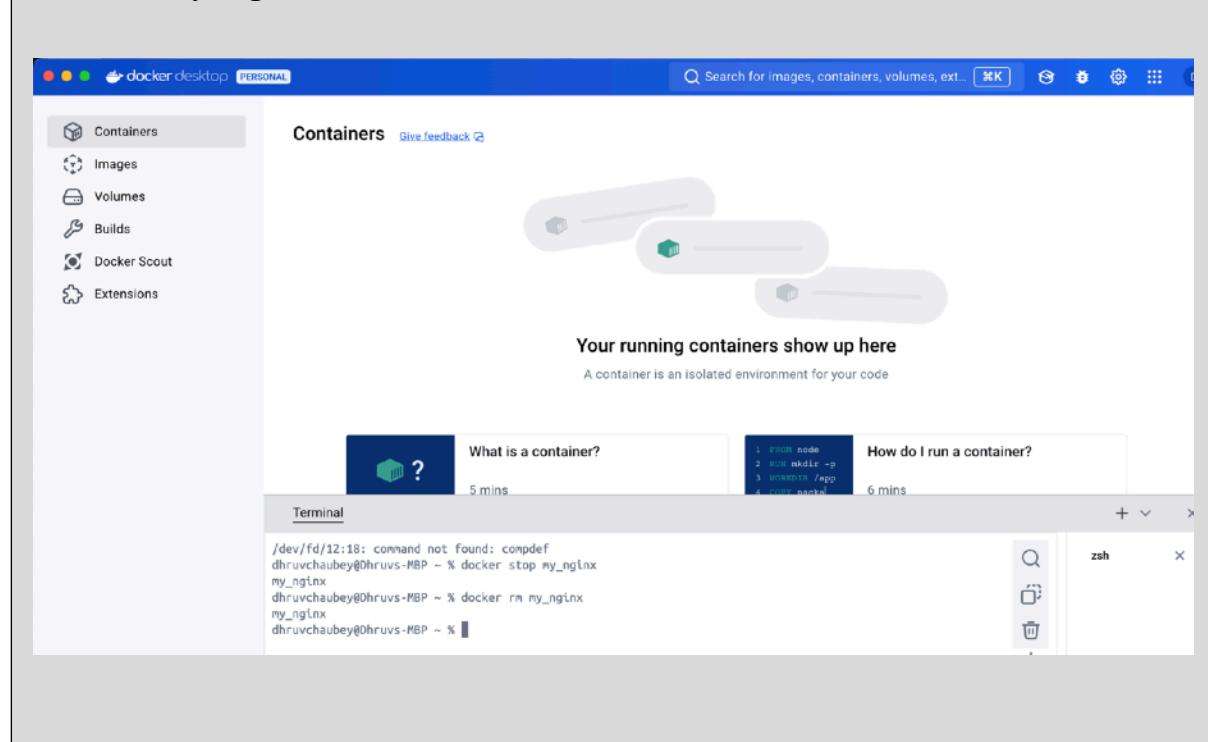
Access the Nginx server to see your file: Open a browser and navigate to http://localhost:8008. You should see the message "Hello, Docker Volume!" displayed on the page.

Step 4: Test Data Persistence

Stop and remove the container:

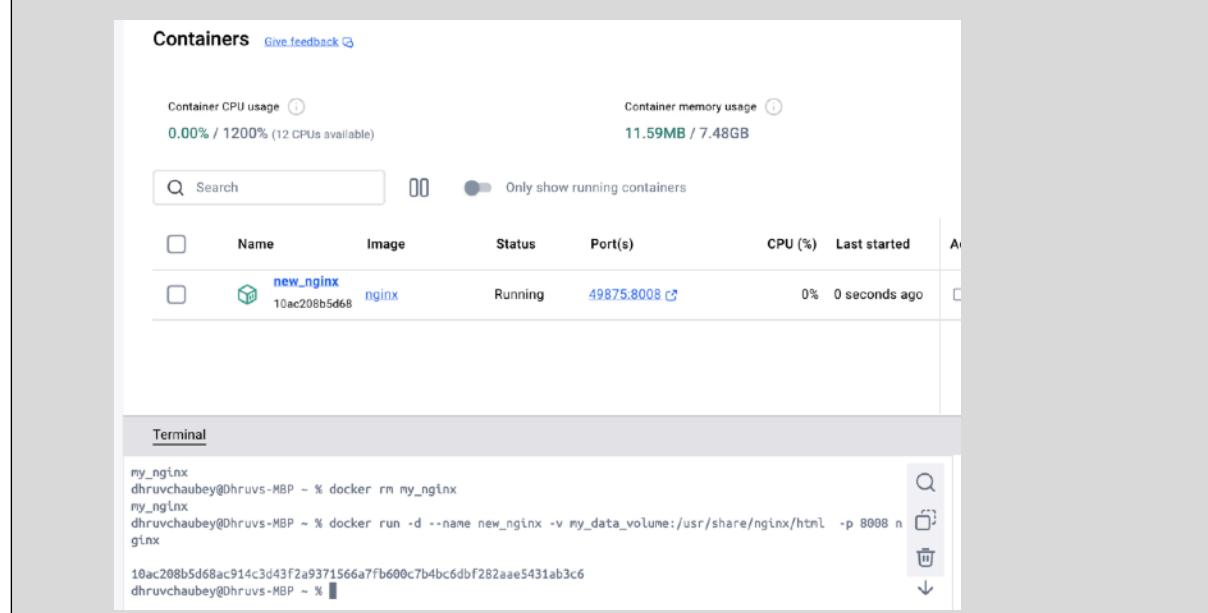
```
docker stop my_nginx
```

```
docker rm my_nginx
```



Run a new Nginx container using the same volume:

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

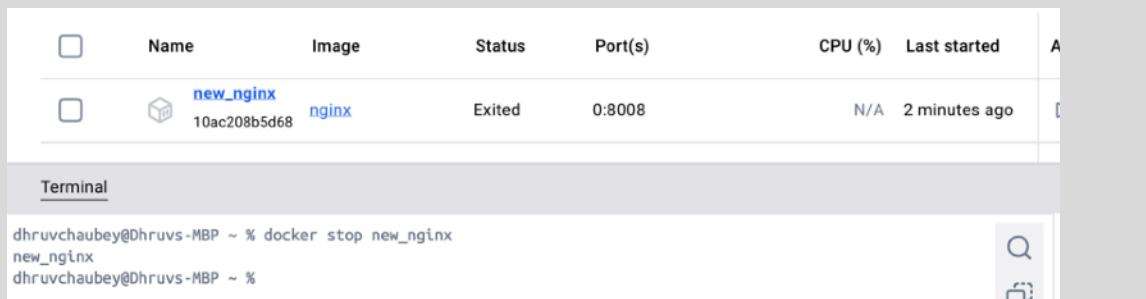


Access the Nginx server again: Navigate to `http://localhost` in your browser. You should still see the "Hello, Docker Volume!" message, demonstrating that the data persisted across container instances.

Step 5: Clean Up

Stop and remove the container:

```
docker stop new_nginx
```

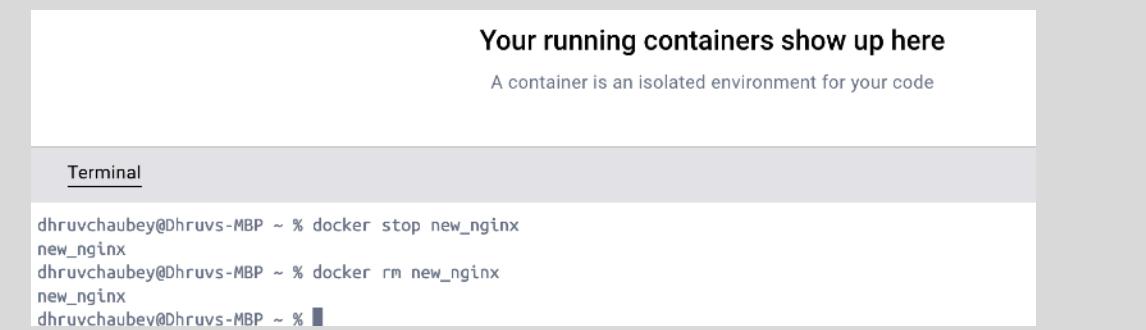


Name	Image	Status	Port(s)	CPU (%)	Last started	A
new_nginx	nginx	Exited	0:8008	N/A	2 minutes ago	[...]

Terminal

```
dhruvchaubey@Dhruv-MBP ~ % docker stop new_nginx
new_nginx
dhruvchaubey@Dhruv-MBP ~ %
```

```
docker rm new_nginx
```



Your running containers show up here

A container is an isolated environment for your code

Terminal

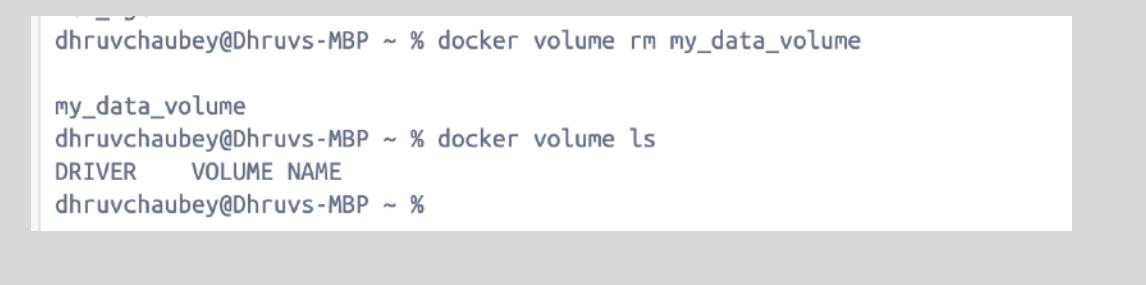
```
dhruvchaubey@Dhruv-MBP ~ % docker stop new_nginx
new_nginx
dhruvchaubey@Dhruv-MBP ~ % docker rm new_nginx
new_nginx
dhruvchaubey@Dhruv-MBP ~ %
```

Remove the Docker volume:

```
docker volume rm my_data_volume
```

Verify that the volume is removed:

```
docker volume ls
```



```
dhruvchaubey@Dhruv-MBP ~ % docker volume rm my_data_volume
my_data_volume
dhruvchaubey@Dhruv-MBP ~ % docker volume ls
DRIVER      VOLUME NAME
dhruvchaubey@Dhruv-MBP ~ %
```

Ensure that `my_data_volume` is no longer listed.