

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`

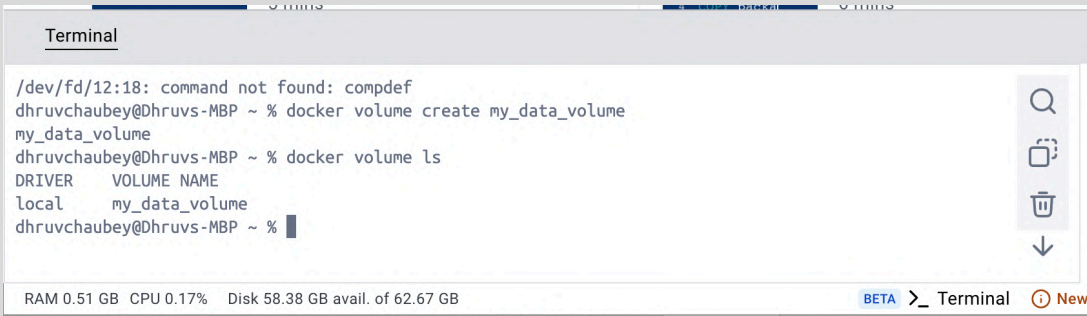


A terminal window titled 'Terminal' showing the command `docker volume create my_data_volume` being executed. The output shows the command was successful. The terminal also displays system information at the bottom: RAM 0.51 GB, CPU --%, Disk 58.38 GB avail. of 62.67 GB. There are tabs at the top for '5 mins' and '6 mins'.

This command creates a Docker volume named `my_data_volume`.

Verify that the volume was created:

`docker volume ls`



A terminal window titled 'Terminal' showing the command `docker volume ls` being executed. The output lists the volume: `DRIVER VOLUME NAME` and `local my_data_volume`. The terminal also displays system information at the bottom: RAM 0.51 GB, CPU 0.17%, Disk 58.38 GB avail. of 62.67 GB. There are tabs at the top for '5 mins' and '6 mins'.

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
```

Terminal

```
DRIVER    VOLUME NAME
local     my_data_volume
dhruvchaubey@Dhruvs-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
aa6fbc30c84e: Pull complete
c28fc33dc48c: Pull complete
08fb08230766: Pull complete
d2983a84b0c4: Pull complete
9f4e03394720: Pull complete
522e88f665e7: Pull complete
d964500c63bd: Pull complete
Digest: sha256:447a8665cc1dab95b1ca778e162215839ccbb9189104c79d7ec3a81e14577add
Status: Downloaded newer image for nginx:latest
0eceaadf7c04be4e2187a3f776befdd5327beb2640fdb0244569c157677dacbb
dhruvchaubey@Dhruvs-MBP ~ %
```

Verify that the container is running:

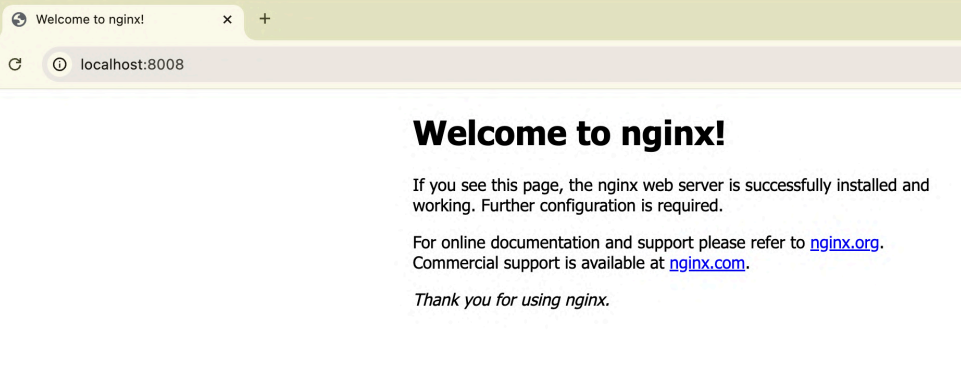
```
docker ps
```

Terminal

```
d964500c63bd: Pull complete
Digest: sha256:447a8665cc1dab95b1ca778e162215839ccbb9189104c79d7ec3a81e14577add
Status: Downloaded newer image for nginx:latest
0eceaadf7c04be4e2187a3f776befdd5327beb2640fdb0244569c157677dacbb
dhruvchaubey@Dhruvs-MBP ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
0eceaadf7c04   nginx    "/docker-entrypoint. ..." About a minute ago Up About a minute 0.0.0.0:8008->80/tcp  my_nginx
dhruvchaubey@Dhruvs-MBP ~ %
```

RAM 0.87 GB CPU -- % Disk 58.17 GB avail. of 62.67 GB

BETA >_ Term



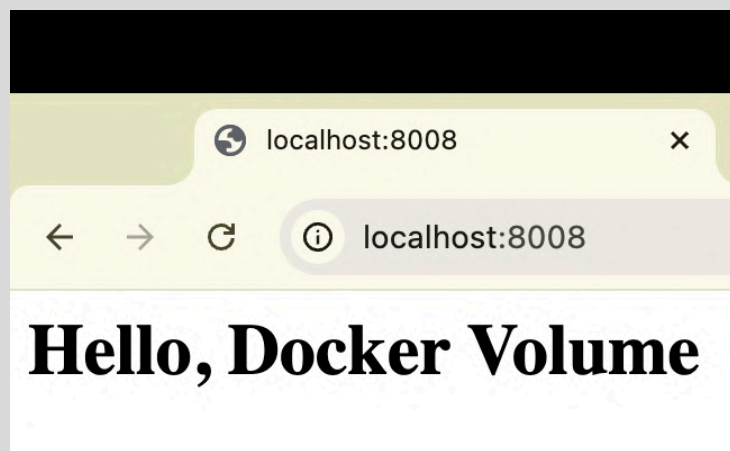
Step 3: Interact with the Volume

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@Dhruvs-MBP Downloads % mkdir usr  
dhruvchaubey@Dhruvs-MBP Downloads % cd usr  
dhruvchaubey@Dhruvs-MBP usr % mkdir share  
dhruvchaubey@Dhruvs-MBP usr % cd share  
dhruvchaubey@Dhruvs-MBP share % mkdir nginx  
dhruvchaubey@Dhruvs-MBP share % cd nginx  
dhruvchaubey@Dhruvs-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@Dhruvs-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`.

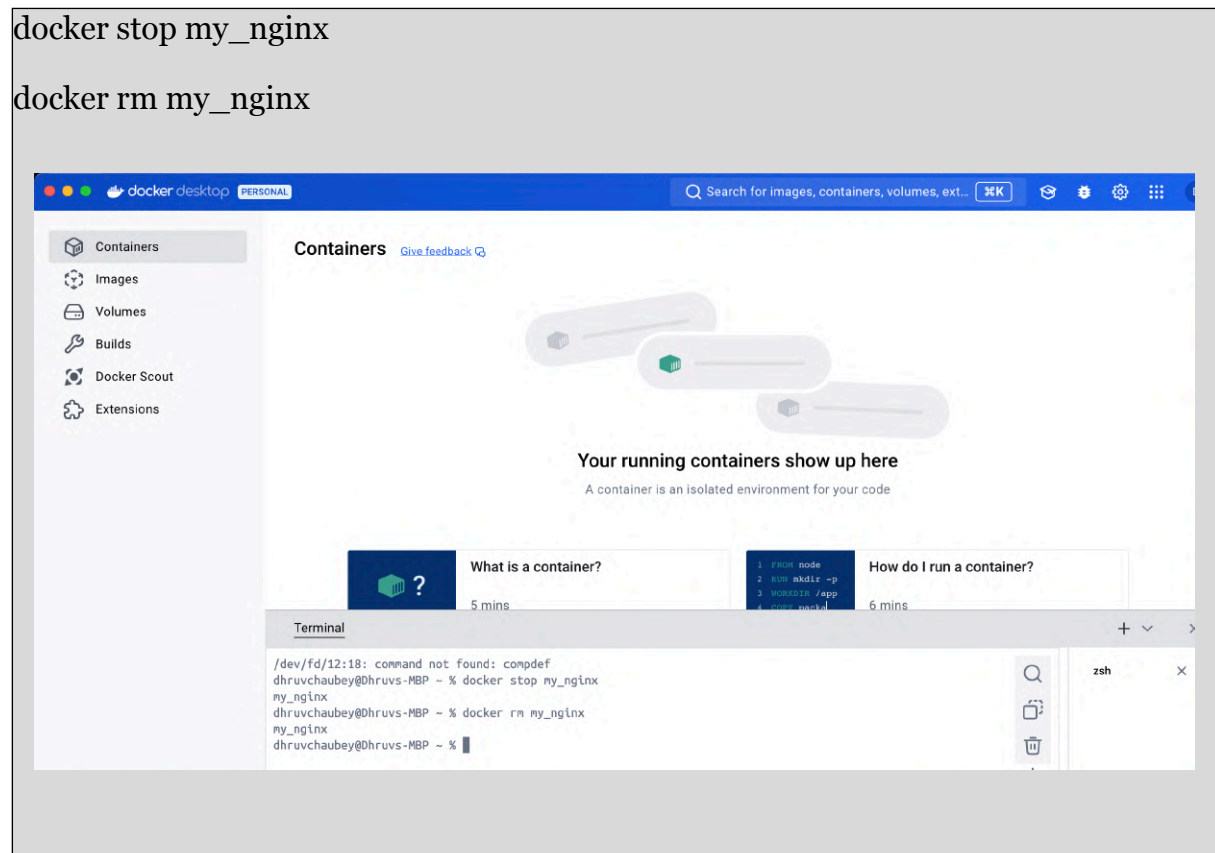
Access the Nginx server to see your file: Open a browser and navigate to `http://localhost:8008`. You should see the message "Hello, Docker Volume!" displayed on the page.

Step 4: Test Data Persistence

Stop and remove the container:

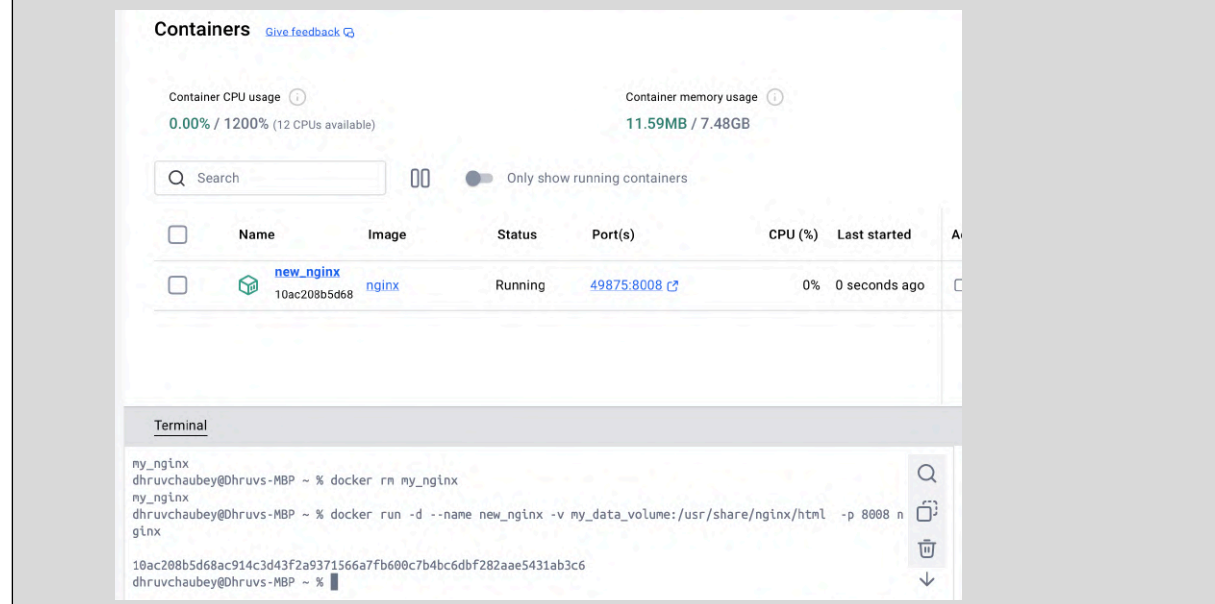
```
docker stop my_nginx
```

```
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
```

<input type="checkbox"/>	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	new_nginx 10ac208b5d68	nginx	Exited	0:8008	N/A	2 minutes ago	 

Terminal
dhruvchaubey@Dhruvs-MBP ~ % docker stop new_nginx
new_nginx
dhruvchaubey@Dhruvs-MBP ~ %

```
docker rm new_nginx
```

Your running containers show up here
A container is an isolated environment for your code

Terminal
dhruvchaubey@Dhruvs-MBP ~ % docker stop new_nginx
new_nginx
dhruvchaubey@Dhruvs-MBP ~ % docker rm new_nginx
new_nginx
dhruvchaubey@Dhruvs-MBP ~ %

Remove the Docker volume:

```
docker volume rm my_data_volume
```

Verify that the volume is removed:

```
docker volume ls
```

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

Ensure that `my_data_volume` is no longer listed.