UPES

UNIVERSITY OF THE FUTURE

Containers & Docker Security LAB

SUBMITTED TO

Dr. Hitesh Kumar Sharma

SUBMITTED BY

Siddharth Agarwal

500107594

R2142220663

Btech CSE DevOps B1

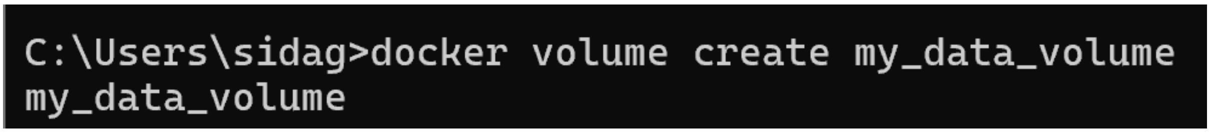
Experiment-3

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.

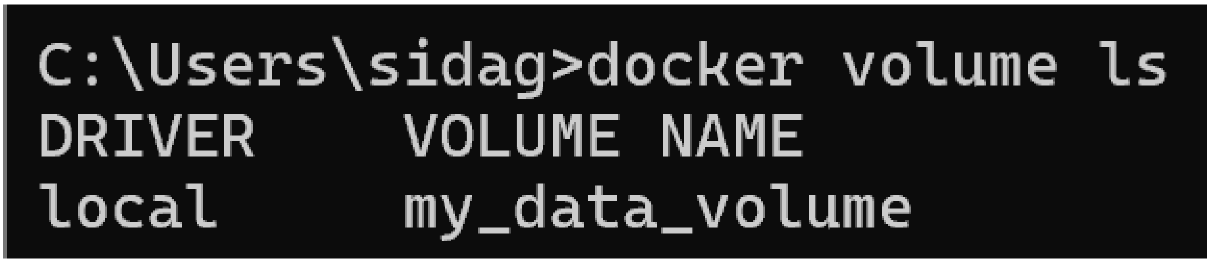


```
C:\Users\sidag>docker volume create my_data_volume
my_data_volume
```

Verify that the volume was created:

```
docker volume ls
```

You should see my_data_volume listed among the volumes.



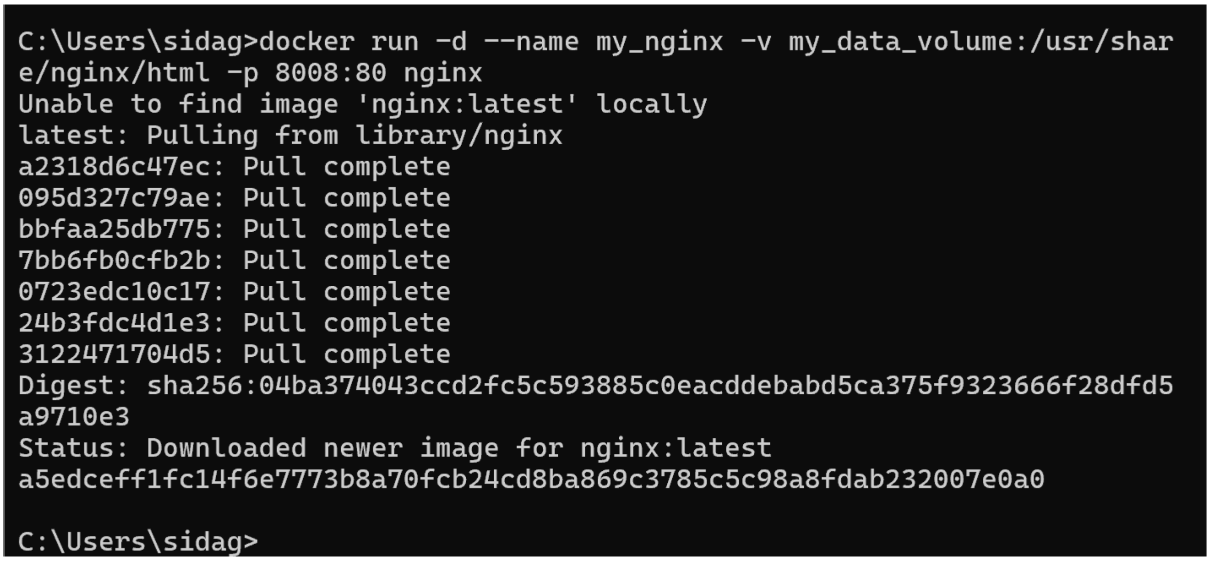
```
C:\Users\sidag>docker volume ls
DRIVER    VOLUME NAME
local     my_data_volume
```

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.

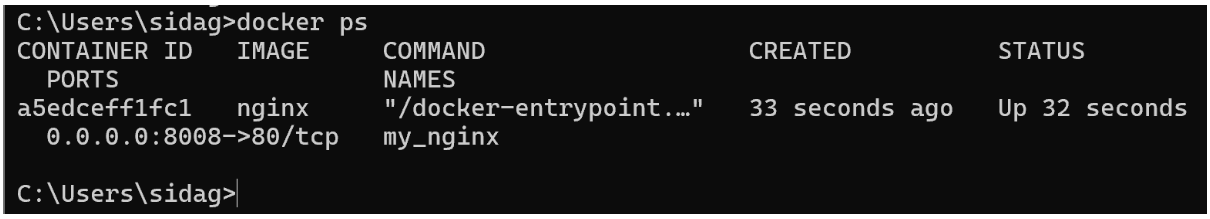


```
C:\Users\sidag>docker run -d --name my_nginx -v my_data_volume:/usr/shar
e/nginx/html -p 8008:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a2318d6c47ec: Pull complete
095d327c79ae: Pull complete
bbfaa25db775: Pull complete
7bb6fb0cfb2b: Pull complete
0723edc10c17: Pull complete
24b3fdc4d1e3: Pull complete
3122471704d5: Pull complete
Digest: sha256:04ba374043ccd2fc5c593885c0eacddebabd5ca375f9323666f28dfd5
a9710e3
Status: Downloaded newer image for nginx:latest
a5edceff1fc14f6e7773b8a70fcb24cd8ba869c3785c5c98a8fdab232007e0a0

C:\Users\sidag>
```

Verify that the container is running:

```
docker ps
```

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



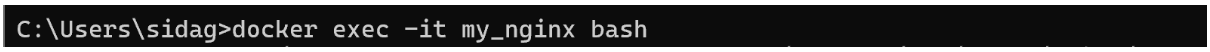
```
C:\Users\sidag>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
  PORTS                  NAMES
a5edceff1fc1   nginx     "/docker-entrypoint.…"   33 seconds ago   Up 32 seconds
  0.0.0.0:8008->80/tcp   my_nginx

C:\Users\sidag>
```

Step 3: Interact with the Volume

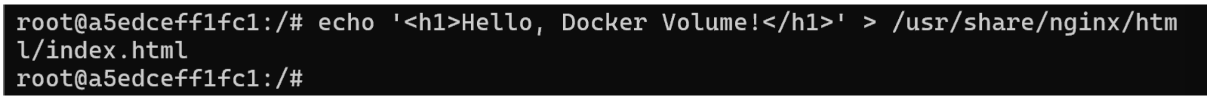
Create a simple HTML file in the volume:

```
docker exec -it my_nginx bash

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

exit
```

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

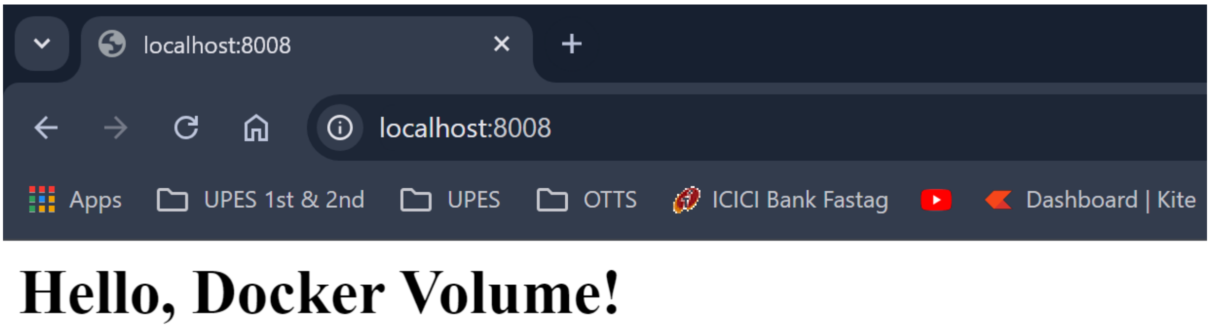


```
C:\Users\sidag>docker exec -it my_nginx bash
```



```
root@a5edceff1fc1:/# echo '<h1>Hello, Docker Volume!</h1>' > /usr/share/nginx/htm
l/index.html
root@a5edceff1fc1:/#
```

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.

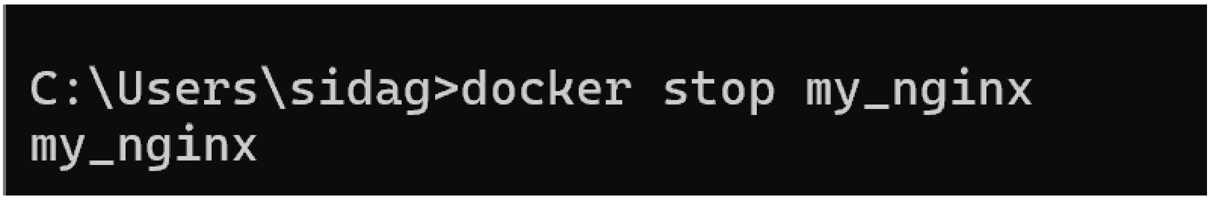


Hello, Docker Volume!

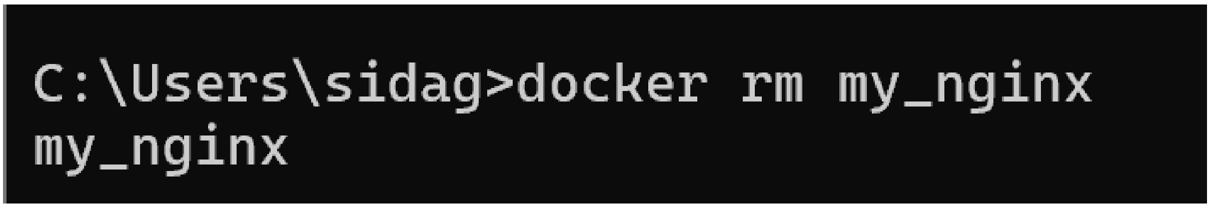
Step 4: Test Data Persistence

Stop and remove the container:

```
docker stop my_nginx

docker rm my_nginx
```



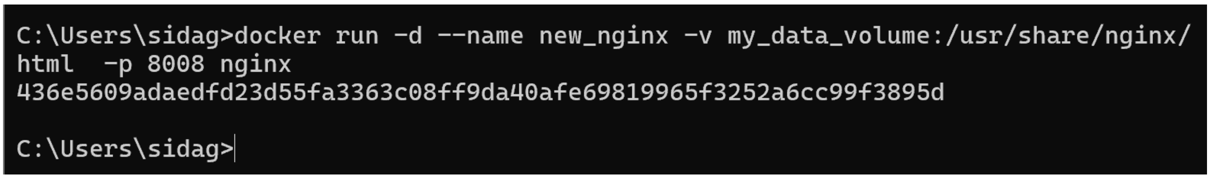
```
C:\Users\sidag>docker stop my_nginx
my_nginx
```



```
C:\Users\sidag>docker rm my_nginx
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
```



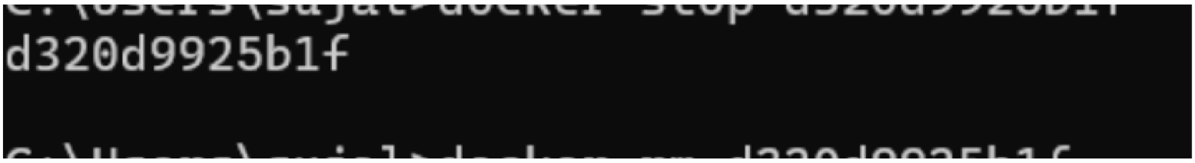
```
C:\Users\sidag>docker run -d --name new_nginx -v my_data_volume:/usr/share/nginx/
html  -p 8008 nginx
436e5609adaedfd23d55fa3363c08ff9da40afe69819965f3252a6cc99f3895d

C:\Users\sidag>
```

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.

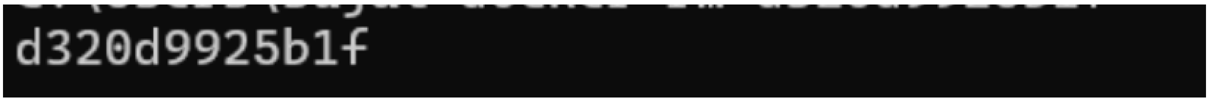
Step 5: Clean Up

Stop and remove the container:

```
docker stop new_nginx

docker rm new_nginx
```



```
d320d9925b1f
```



```
d320d9925b1f
```

Remove the Docker volume:

```
docker volume rm my_data_volume
```

Verify that the volume is removed:

```
docker volume ls
```

Ensure that my_data_volume is no longer listed.