ANSHIKA SRIVASTAVA
ROLL NUMBER – R2142220907
SAP ID – 500107049
DEVSECOPS BATCH B1 HONS.

EXPERIMENT – 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.

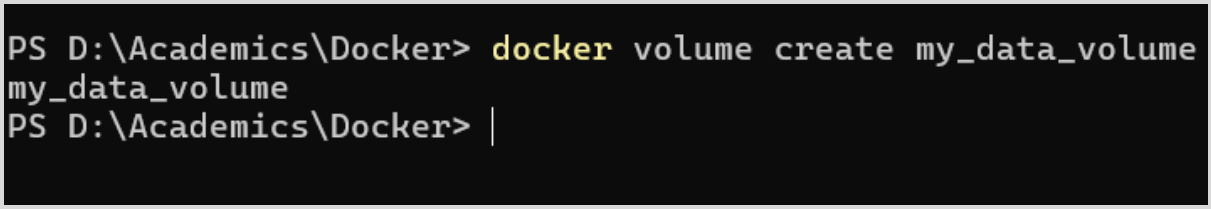
Prerequisites:

- 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
```



This command creates a Docker volume named my_data_volume.

Verify that the volume was created:

```
docker volume ls
```

```
PS D:\Academics\Docker> docker volume ls
DRIVER    VOLUME NAME
local     c0d4c403d74bb835395f73da488c38a1a1a0efbeb95621e98227fa6c2157728d
local     d5ffef87cc04e3e192b77a19bf74809be30e762524f9fbedcd497341a1fc77d3
local     my_data_volume
local     my_vol
PS D:\Academics\Docker>
```

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 D:\Academics\Docker> docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
a2318d6c47ec: Already exists
095d327c79ae: Pull complete
bbfaa25db775: Pull complete
7bb6fb0cfb2b: Pull complete
0723edc10c17: Pull complete
24b3fdc4d1e3: Pull complete
3122471704d5: Pull complete
Digest: sha256:04ba374043ccd2fc5c593885c0eacddebabd5ca375f9323666f28dfd5a9710e3
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview nginx
PS D:\Academics\Docker> docker run -d --name nginx_anshika -v my_data_volume:/usr/share/nginx/html -p 8008:80 nginx
371e99f7e390ed83991a3d1509b63142635b2e00fa295bcea49f74a8d2889d28
PS D:\Academics\Docker>
```

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 D:\Academics\Docker> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                    NAMES
371e99f7e390   nginx     "/docker-entrypoint.…"   43 seconds ago   Up 41 seconds   0.0.0.0:8008->80/tcp     nginx_anshika
43bbadd13889   redis     "docker-entrypoint.s…"   21 minutes ago   Up 21 minutes   6379/tcp                 vol2
53cfc1e01d1f   redis     "docker-entrypoint.s…"   37 minutes ago   Up 37 minutes   6379/tcp                 Anshika_redis
PS D:\Academics\Docker>
```

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

```
PS D:\Academics\Docker> docker exec -it nginx_anshika bash
root@371e99f7e390:/# echo "<h1>Hello, Docker Volume! from Anshika Srivastava</h1>" > /usr/share/nginx/html/index.html
root@371e99f7e390:/# exit
exit
PS D:\Academics\Docker>
```

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

```
PS D:\Academics\Docker> docker stop nginx_anshika
nginx_anshika
PS D:\Academics\Docker>
```

docker rm my_nginx

```
PS D:\Academics\Docker> docker rm nginx_anshika
nginx_anshika
PS D:\Academics\Docker>
```

Run a new Nginx container using the same volume:

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

```
PS D:\Academics\Docker> docker run -d -p 8011:80 -v my_data_volume:/usr/share/nginx/html nginx
05f5cc49cdb9808bbc2932a44ed1ecefcb1f2267c2dd3dd9738bc6222810e638
PS D:\Academics\Docker>
```

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

```
PS D:\Academics\Docker> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                  NAMES
05f5cc49cdb9   nginx     "/docker-entrypoint.…"   2 minutes ago    Up 2 minutes    0.0.0.0:8011->80/tcp   jovial_buck
43bbadd13889   redis     "docker-entrypoint.s…"   32 minutes ago   Up 32 minutes   6379/tcp               vol2
53cfc1e01d1f   redis     "docker-entrypoint.s…"   48 minutes ago   Up 48 minutes   6379/tcp               Anshika_redis
PS D:\Academics\Docker> docker stop jovial_buck
jovial_buck
```

docker rm new_nginx

```
PS D:\Academics\Docker> docker rm jovial_buck
jovial_buck
PS D:\Academics\Docker>
```

Remove the Docker volume:

docker volume rm my_data_volume

```
PS D:\Academics\Docker> docker volume rm my_data_volume
my_data_volume
PS D:\Academics\Docker>
```

Verify that the volume is removed:

docker volume ls

```
PS D:\Academics\Docker> docker volume ls
DRIVER    VOLUME NAME
local     c0d4c403d74bb835395f73da488c38a1a1a0efbeb95621e98227fa6c2157728d
local     d5ffef87cc04e3e192b77a19bf74809be30e762524f9fbedcd497341a1fc77d3
local     my_vol
PS D:\Academics\Docker>
```

Ensure that my_data_volume is no longer listed.