**Lab Experiment 3: Docker Volume**

In this lab experiment, you will learn how to work with Docker volumes, which are used to persist data across containers. Volumes enable data to be stored outside the container filesystem and are crucial for managing data consistency and sharing data between containers.

Prerequisites:

Docker installed and running on your machine.

Objective:

Create a Docker volume, use it with a container, and observe how data persists across container instances.

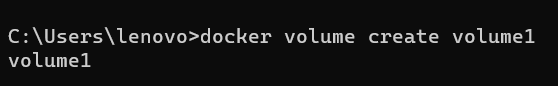
Steps:

**Step 1: Create a Docker Volume**

Open a terminal on your machine.

Run the following command to create a Docker volume named "my\_volume":

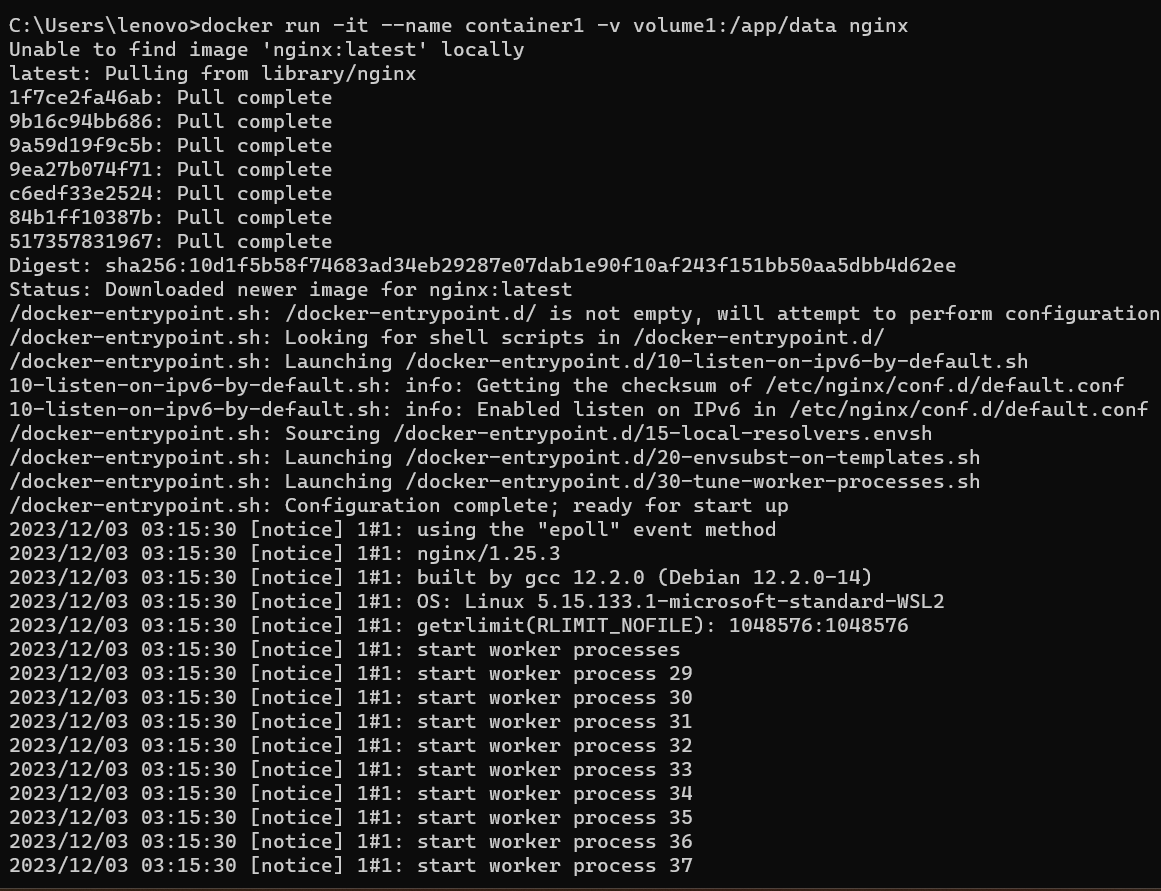
docker volume create my\_volume

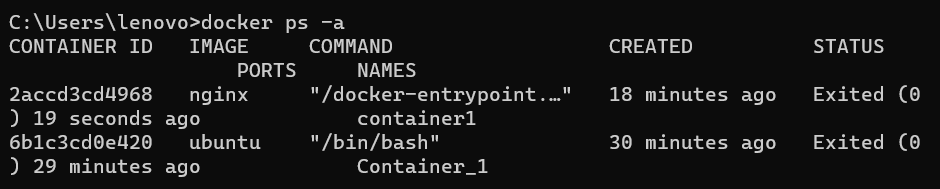


Step 2: Launch Containers with the Volume

Run a container using the volume you created:

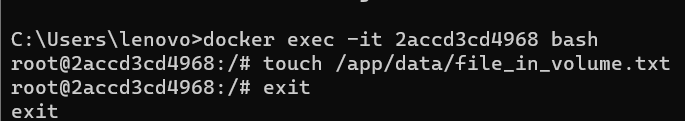
docker run -it --name container1 -v my\_volume:/app/data nginx

Enter the container to observe the volume and create a file inside it:



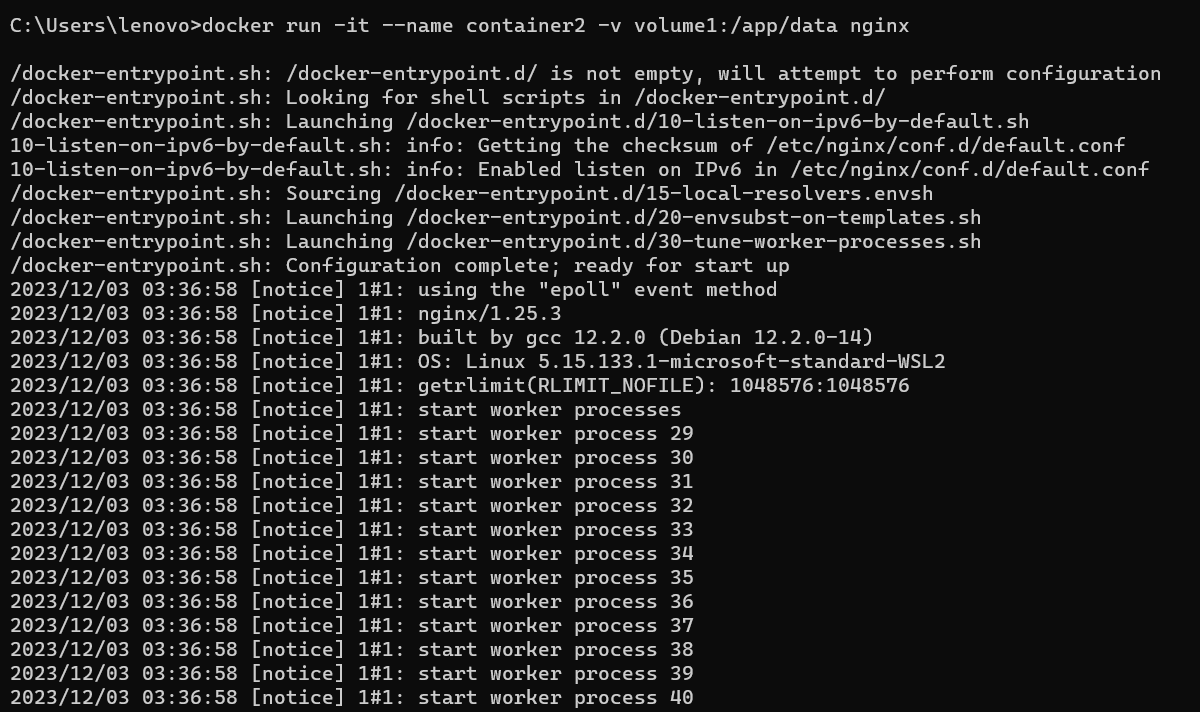
touch /app/data/file\_in\_volume.txt

exit



Run a second container, using the same volume, to verify data persistence:

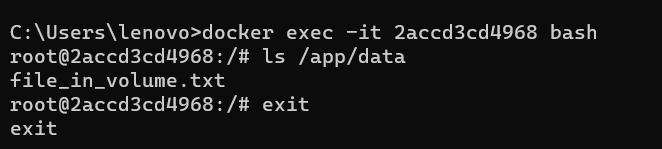
docker run -it --name container2 -v my\_volume:/app/data nginx



Enter the second container and check if the file exists:

ls /app/data

exit

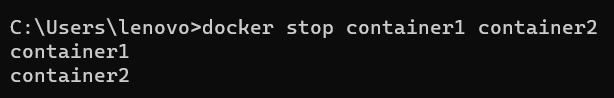


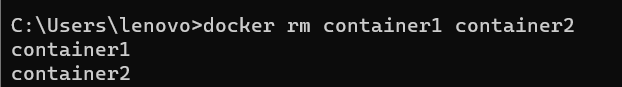
Step 3: Cleanup

Stop and remove the containers:

docker stop container1 container2

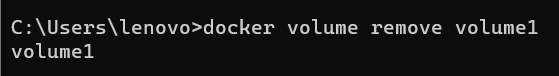
docker rm container1 container2





Remove the volume:

docker volume rm my\_volume



**Conclusion:**

In this experiment, you learned how to create a Docker volume, associate it with containers, and observed how data persisted between different container instances. Docker volumes are essential for maintaining data integrity, sharing data between containers, and ensuring data persistence even when containers are removed or replaced. This skill is crucial for managing stateful applications and databases within a Dockerized environment.