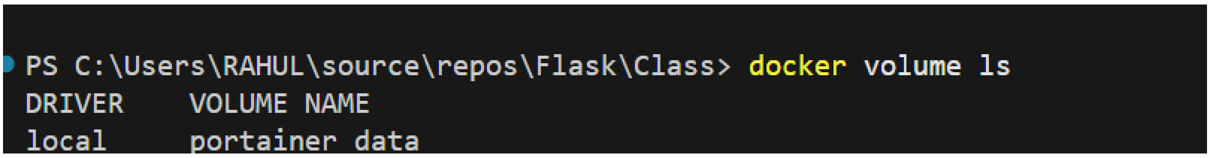
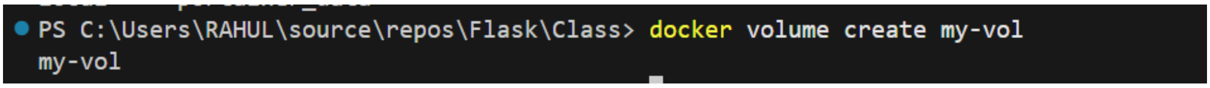
Create and manage volume

CMD : Docker volume ls



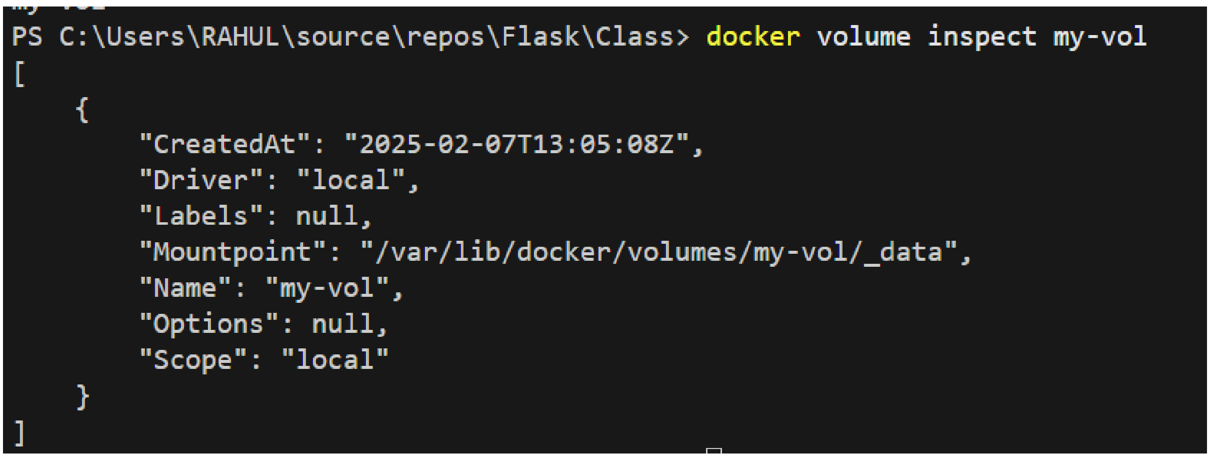
```
PS C:\Users\RAHUL\source\repos\Flask\Class> docker volume ls
DRIVER    VOLUME NAME
local     portainer_data
```

CMD : docker create volume my-vol



```
PS C:\Users\RAHUL\source\repos\Flask\Class> docker volume create my-vol
my-vol
```

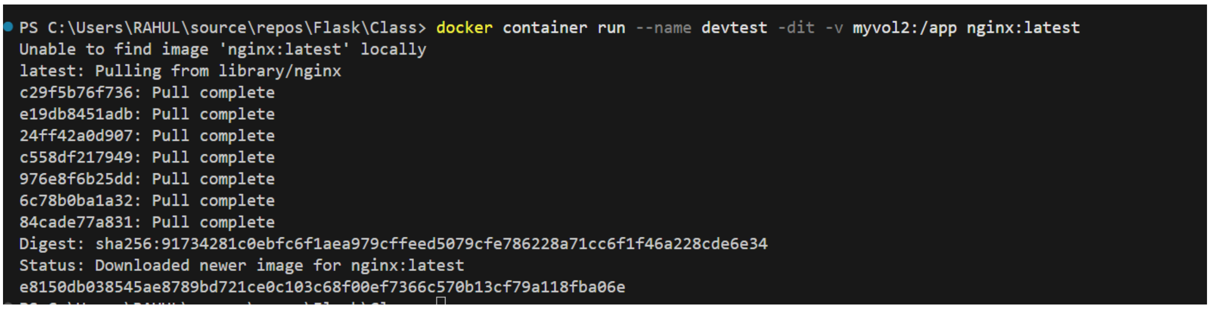
CMD : docker volume inspect my-vol



```
PS C:\Users\RAHUL\source\repos\Flask\Class> docker volume inspect my-vol
[
    {
        "CreatedAt": "2025-02-07T13:05:08Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/my-vol/_data",
        "Name": "my-vol",
        "Options": null,
        "Scope": "local"
    }
]
```

Start container with volume

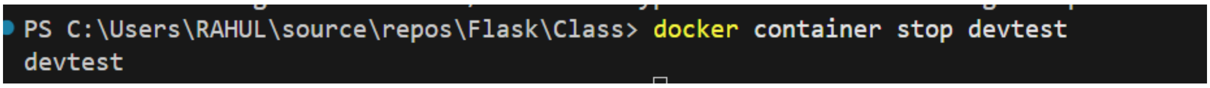
CMD : docker container run --name devtest -dit -v myvol2:/app nginx:latest



```
PS C:\Users\RAHUL\source\repos\Flask\Class> docker container run --name devtest -dit -v myvol2:/app nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
c29f5b76f736: Pull complete
e19db8451adb: Pull complete
24ff42a0d907: Pull complete
c558df217949: Pull complete
976e8f6b25dd: Pull complete
6c78b0ba1a32: Pull complete
84cade77a831: Pull complete
Digest: sha256:91734281c0ebfc6f1aea979cffeed5079cfe786228a71cc6f1f46a228cde6e34
Status: Downloaded newer image for nginx:latest
e8150db038545ae8789bd721ce0c103c68f00ef7366c570b13cf79a118fba06e
```

Stop the container and remove the volume

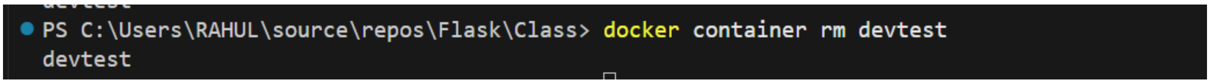
CMD : docker container stop devtest



```
PS C:\Users\RAHUL\source\repos\Flask\Class> docker container stop devtest
devtest
```

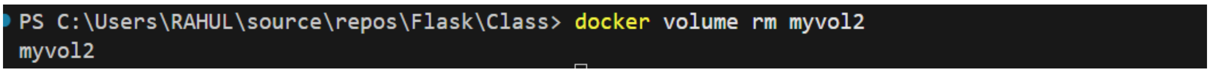
Remove the container

CMD : docker container rm devtest



```
PS C:\Users\RAHUL\source\repos\Flask\Class> docker container rm devtest
devtest
```

CMD : docker container rm devtest



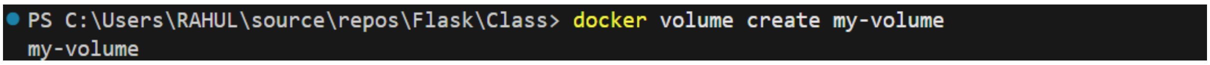
```
PS C:\Users\RAHUL\source\repos\Flask\Class> docker volume rm myvol2
myvol2
```

Persist a volume using container

Create a Docker volume:

CMD : docker volume create my-volume

This command creates a new Docker volume named my-volume.

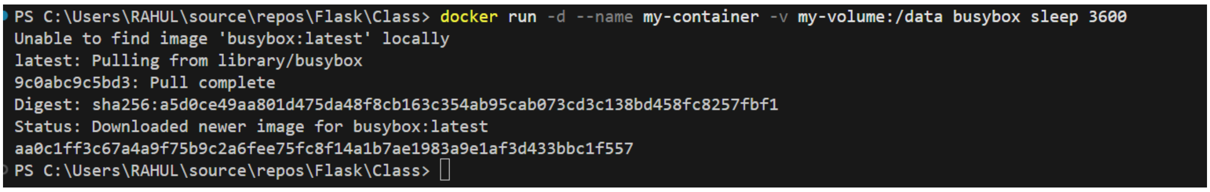


```
PS C:\Users\RAHUL\source\repos\Flask\Class> docker volume create my-volume
my-volume
```

Run a Docker container with the volume:

CMD : docker run -d --name my-container -v my-volume:/data busybox sleep 3600

This command runs a new container named my-container using the busybox image. The -v flag mounts the my-volume volume to the /data directory inside the container. The container will run for 3600 seconds (1 hour) before stopping, giving you plenty of time to interact with it.



```
PS C:\Users\RAHUL\source\repos\Flask\Class> docker run -d --name my-container -v my-volume:/data busybox sleep 3600
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
9c0abc9c5bd3: Pull complete
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Downloaded newer image for busybox:latest
aa0c1ff3c67a4a9f75b9c2a6fee75fc8f14a1b7ae1983a9e1af3d433bbc1f557
PS C:\Users\RAHUL\source\repos\Flask\Class>
```

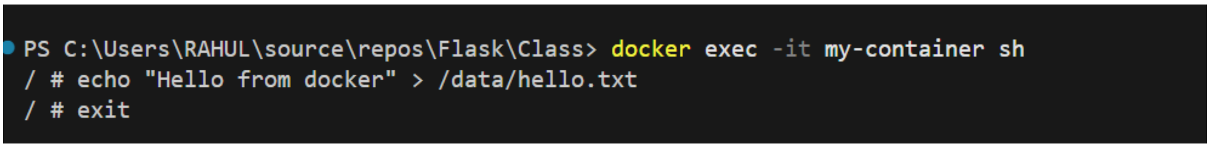
Add data to the volume: Now, let's add some data to the volume by logging into the container and creating a file in the /data directory:

CMD : docker exec -it my-container sh

Inside the container:

echo "Hello, Docker!" > /data/hello.txt

exit

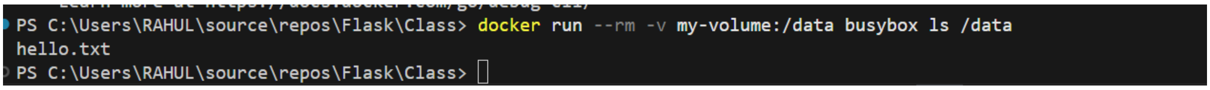


```
PS C:\Users\RAHUL\source\repos\Flask\Class> docker exec -it my-container sh
/ # echo "Hello from docker" > /data/hello.txt
/ # exit
```

Show the volume data: To verify that the data has been added to the volume, you can run another container and mount the same volume to inspect its contents:

docker run --rm -v my-volume:/data busybox ls /data

This command runs a temporary busybox container with the volume mounted to /data and lists the contents of the /data directory.



```
    Learn more at https://docs.docker.com/go/debug-cli/
PS C:\Users\RAHUL\source\repos\Flask\Class> docker run --rm -v my-volume:/data busybox ls /data
hello.txt
PS C:\Users\RAHUL\source\repos\Flask\Class>
```

Jenkins use case

Pull the Jenkins Docker image: Open Command Prompt or PowerShell and run the following command to pull the Jenkins Docker image:

CMD : docker pull jenkins/jenkins:lts

This command downloads the latest stable (LTS) Jenkins image.

Run the Jenkins container: Use the following command to start the Jenkins container:

CMD : docker run --name myjenkins1 -dit -p 8080:8080 -p 50000:50000 -v jenkins_data:/var/jenkins_home jenkins/jenkins:lts

This command starts a new Jenkins container with the following options:

- --name myjenkins1: Names the container myjenkins1.
- -dit: Runs the container in detached mode (-d), interactive mode (-i), and allocates a pseudo-TTY (-t).
- -p 8080:8080: Maps port 8080 of the container to port 8080 on the host.
- -p 50000:50000: Maps port 50000 of the container to port 50000 on the host.
- -v jenkins_data:/var/jenkins_home: Mounts the Docker volume jenkins_data to the /var/jenkins_home directory inside the container.

jenkins/jenkins:lts: Specifies the Jenkins image to use

Access Jenkins: Open your web browser and go to http://localhost:8080. You should see the Jenkins setup wizard.

Retrieve the initial admin password: To get the initial admin password, run the following command:

CMD docker exec -it myjenkins1 sh

Inside the container:

cat /var/jenkins_home/secrets/initialAdminPassword

This will display the initial admin password1.

Complete the setup: Use the initial admin password to log in to Jenkins and complete the setup wizard.