Create a Docker Volume and Mount it to a Container

Step 1: Create a Docker Volume

docker volume create my-volume

Explanation:

- my-volume is the name of your volume (you can name it anything).
- This creates a volume managed by Docker in /var/lib/docker/volumes/.

Step 2: Verify the Volume

docker volume ls

You should see output like:

DRIVER VOLUME NAME

local my-volume

Step 3: Mount the Volume to a Container

docker run -dit --name vol-container -v my-volume:/data alpine

Explanation:

- -v my-volume:/data → Mounts the Docker volume my-volume to /data inside the container.
- alpine is a lightweight base image used here for demonstration.

Step 4: Inspect the Volume Mount

docker inspect vol-container

Look for the Mounts section to see details like:

```
"Mounts": [
 {
  "Type": "volume",
  "Name": "my-volume",
  "Destination": "/data",
  ...
 }
]
```

Step 5: Test Volume Functionality

Write a file from inside the container:

docker exec -it vol-container sh

echo "Hello from volume!" > /data/hello.txt

exit

Then, you can mount the same volume to another container:

docker run -it --rm -v my-volume:/data alpine cat /data/hello.txt

It will output:

Hello from volume!

This proves the volume is persistent and shared.

Step 6: Remove the Container and Volume (Optional)

docker rm -f vol-container

docker volume rm my-volume

Summary

Task	**Command**
Create Volume	docker volume create my-volume
List Volumes	docker volume ls
Mount Volume to Container	docker run -v my-volume:/data alpine
Inspect Container Mounts	docker inspect <container_name>
Write to Volume	echo "..." > /data/... (inside container)
Share Volume with Another Container	docker run -v my-volume:/data alpine
Delete Volume	docker volume rm my-volume