Ques 7 : Create a Docker volume and mount it to a container.

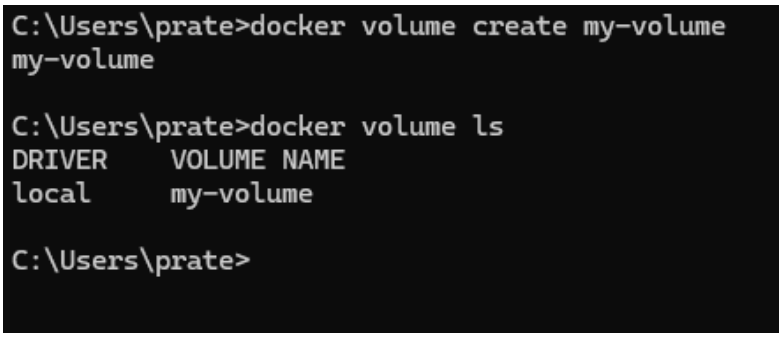
SOLN:

Step 1: Create a Docker Volume

```
docker volume create my-volume
```

You can list all volumes to confirm:

```
docker volume ls
```



Step 2: Run a Container and Mount the Volume

Mount the volume to a specific path in the container:

```
docker run -d -v my-volume:/app/data --name my-container nginx
```

- `my-volume`: your named volume.
- `/app/data`: directory inside the container where the volume will be mounted.
- `nginx`: image you're running (can be any image).
- `--name my-container`: optional name for your container.

Step 3: Check Volume Mount

To inspect where the volume is mounted:

```
docker inspect my-container
```

Search for the `Mounts` section in the output.

Volume Mount Confirmed

```
"Mounts": [
  {
    "Type": "volume",
    "Name": "my-volume",
    "Source": "/var/lib/docker/volumes/my-volume/_data",
    "Destination": "/app/data",
    "Driver": "local",
    "RW": true
  }
]
```

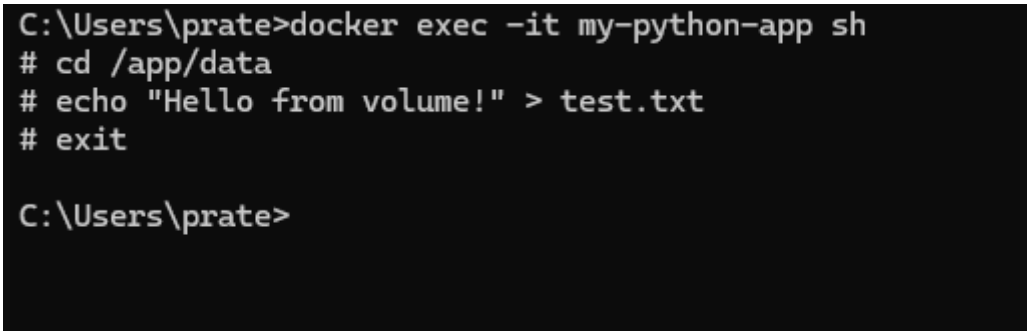
Step 4: Test the Volume

1. **Exec into the container**:

   ```
   docker exec -it my-container sh
   ```

2. **Go to the mounted directory**:

   ```
   cd /app/data
   echo "Hello from volume!" > test.txt
   ```

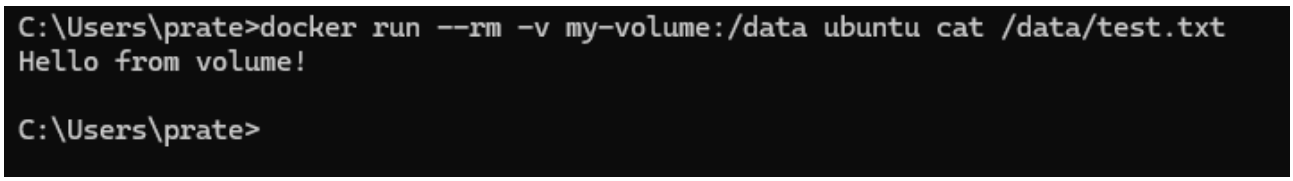


3. **Exit and check from host**:

   ```
   docker run --rm -v my-volume:/data alpine cat /data/test.txt
   ```

It should output:

`Hello from volume!`



`docker run -it --rm -v my-volume:/data ubuntu sh` → open a shell inside and explore `/data`

This is an extra verification step to:

- Check if the volume is mounted properly.
- See all the files present inside `/data`.
- Add, modify, or remove files manually inside the volume.