

Docker containers are designed so that their internal storage can be easily destroyed. However, sometimes you might need more permanent data. Docker volumes and bind mounts allow you to attach external storage to containers. In this lesson, we will discuss bind mounts and volumes. We will also demonstrate how to create them using both the `--mount` and `-v` flags. We will talk about sharing volumes between multiple containers and go over some commands you can use to manage volumes.

Relevant Documentation

- <https://docs.docker.com/storage/>
- <https://docs.docker.com/storage/bind-mounts/>
- <https://docs.docker.com/storage/volumes/>

Lesson Reference

1. Create a directory on the host with some test data.

```
cd ~/
mkdir message
echo Hello, world! > message/message.txt
```

2. Mount the directory to a container with a bind mount.

```
docker run --mount type=bind,source=/home/cloud_user/message,destination=/root,readonly busybox cat /root
```

3. Run a container with a mounted volume.

```
docker run --mount type=volume,source=my-volume,destination=/root busybox sh -c 'echo hello > /root/messa
```

4. Use the `-v` syntax to create a bind mount and a volume.

```
docker run -v /home/cloud_user/message:/root:ro busybox cat /root/message.txt
docker run -v my-volume:/root busybox sh -c 'cat /root/message.txt'
```

5. Use a volume to share data between containers.

```
docker run --mount type=volume,source=shared-volume,destination=/root busybox sh -c 'echo I wrote this! >
docker run --mount type=volume,source=shared-volume,destination=/anotherplace busybox cat /anotherplace/m
```

6. Create and manage volumes using `docker volume` commands.

```
docker volume create test-volume
docker volume ls
docker volume inspect test-volume
docker volume rm test-volume
```