Lab 11.1: Docker Volumes
IN720 Virtualisation

Introduction

In this lab we are going to play around with Docker volumes and bind mounts to see how we can create and seed volumes with data that we can then share across other containers.

1 Build a data container

Build a new container from a Dockerfile that with the following:

1. Create a directory, `/data` inside your container.
2. In `/data`, create files named `foo`, `bar`, and `baz`.
3. Make `/data` a volume.

Build your image and launch a container called `vol_lab_1` from it.

2 Find your volume on the host system

In another ssh session, inspect your new container to find its volumes with the command

```
sudo docker inspect -f {{.Mounts}} vol_lab_1
```

Once you find the directory on the host that stores your volume, modify one of the files inside it. Attach a terminal to `vol_lab_1` and see if the changes on the host system are visible inside the container.

3 Using the volume in another container

Shut down `vol_lab_1`. Start a new container with the command

```
sudo docker run -it  --volumes-from=vol_lab_1 ubuntu /bin/bash
```

When your new container starts, inspect the `/data` volume inside it. Make some changes inside the volume and then shut down this container. Then, check the volume directory on the host system and see if your changes are visible there.

4 Creating a bind mount from an exisiting directory on the host

Often, especially when developing applications, we have a directory on the host system that contains code or other resources we want to use inside a container while still keeping access to them directly on the host. Modify the container image you created earlier by doing the following:

1. Create a directory, `/home/user/mydata` on your host system. Place some files in it.
2. In your Dockerfile, remove the earlier commands that created the files in `/data`, but do create the `/data` directory.
3. Rebuild your image and start a new container from it. Create a bind mount of `/home/user/mydata` onto `/data`
4. On your host system, modify the contents of `/home/user/mydata` and then observe how those changes are visible inside your running container.

5 One final note

The important property of volumes is that they exist independently of the containers that use them. You can see what volumes are present on your system with the command

```
docker volume ls
```