

Docker Volumes

Virtualisation

Otago Polytechnic
Dunedin, New Zealand

UNION FILE SYSTEMS

- ▶ Files in a Docker container are presented in a *Union File System*
- ▶ An image is stored as a set of read-only layers.
- ▶ Only the very top layer in a running container is read-write.
- ▶ When we modify a file from an image, we copy it from a lower, read-only layer into the read-write layer.
- ▶ The modified copy saved on the top layer masks the unmodified version on the lower layer.
- ▶ When we destroy a container, the read-write layer associated with it is destroyed and our changes are lost.

DOCKER VOLUMES

- ▶ Sometimes we want to persist data created inside a container, or we want to share data between containers.
- ▶ A Docker *volume* lets us do this.
- ▶ Volumes are files or directories that sit outside the union file system and are saved directly on the host file system.
- ▶ Other containers can access volumes, and we can delete a container without deleting its volumes.

CREATING VOLUMES

We can create volumes from the command line:

```
sudo docker run -it --name vol-test \  
--mount type=volume,target=/data \  
ubuntu /bin/bash
```

CREATING VOLUMES

We can create volumes from the in a Dockerfile:

```
FROM ubuntu:18.04  
VOLUME /data
```

CREATING VOLUMES

We can mount volumes that are associated with another container:

```
docker run -it --name vol-test-2  
--volumes-from vol-test  
ubuntu /bin/bash
```

Note that

- ▶ The container `vol-test` *doesn't have to be running*.
- ▶ Changes to the volume are immediately visible in both containers (and the host).

CREATING VOLUMES

We can create volumes without associating them with a particular container:

```
docker volume create --name Volume1
```

And then use them in a container later

```
docker run -ti \  
--mount type=volume,source=Volume1,target=/volume1 \  
ubuntu
```

BIND MOUNTS

We can create “volumes” (actually *bind mounts*) from directories that already exist on the host:

```
sudo docker run -it --name vol-test3 \  
--mount type=bind,source=/home/tclark/data,target=/data \  
ubuntu /bin/bash
```