Lab 9.2 Introduction to Docker
IN720 Virtualisation

Introduction

Last time we saw how we could use LXC to create and run containers on a host. While LXC is powerful and useful, its interface is a little low level. *Docker* is a tool that builds on the ideas we saw in LXC and extends our abilities to create and manage containers. In this lab we will start working with a more full featured and widely used container toolset *Docker*.

1 Setup

Use the same server that you used for the LXC lab. Install the Docker tools with the command

```
sudo apt-get install docker.io
```

Right now, the docker commands are only usable by `root`. The Docker package created a `docker` group, however, and any member of that groupc can run commands without `sudo`. Add your user to the `docker` group with the command `sudo adduser user docker`.

After installing, check to see that your Docker server is running with the command

```
docker info
```

You should see output like the following:

```
content..Containers: 0
Images: 0
Storage Driver: aufs
Root Dir: /var/lib/docker/aufs
Dirs: 6
Execution Driver: native-0.2
Kernel Version: 3.13.0-59-generic
WARNING: No swap limit support
.
```

2 Creating and running a container

Create a new container with the command

```
docker run -i -t --name fred ubuntu /bin/bash
```

This will create a new container named `fred` based on the ubuntu base image. We have told docker to run `bash` on the container, and the `-i` and `-t` options connect us to an interactive console on it.

Once the container is up you can interact with it normally. A few things about your container environment are interesting. Run `top` to see what is running inside the container. For comparison, you mnay want to run `top` on the host system when you exit the container. Also, use `ip a` to inspect the container's network interfaces.

Type `exit` to return to the host. Since this terminates the `bash shell` the container itself stops.

On the host system, type `docker ps -a` to list the containers on the system. (Without the `-a` it will only show running containers) You can get more information about your container with the command

```
docker inspect fred
```

Now restart your container with the command

```
docker start fred
```

and run `sudo docker ps`. You will see that the container is running, but we are not attached to the console. You can attach to is with the command

```
docker attach fred
```

3 Next steps

We don't usually want to create containers that we have to deal with interactively. To see how to create and use *daemonized containers*, work through Chapter 3, sections 7-17 of the *The Docker Book*.