Lab Exercise: Installing Docker

Preamble

This lab exercise involves installing Docker onto your local machine (either directly, or onto a virtual machine), ready for use in forthcoming lab exercises.

Step 1 – install Docker

Depending on the platform you are using, use the course notes to install Docker. If you're installing Docker locally on Linux, use the Docker Project's repositories so that you obtain a current release.

Step 2 – start the Docker server

Windows/OS X: the Docker Toolbox installer will have installed the toolbox contents, but will not have configured Docker for use. A virtual machine needs to be created with the Docker server component installed, which then needs to be booted. Following this, SSH keys need to be generated to allow secure communication between the local Docker client and the Docker host. Finally, the local Docker environment needs to pointed at the virtual machine.

The simplest way to achieve this is to double click the 'Docker Quickstart' icon installed by Docker Toolbox. This opens a terminal window, executes some Docker specific commands, and leaves you at a prompt ready to use Docker.

Linux: the installation commands will have installed Docker, but not necessarily enabled it to start at boot time, or started it at this moment. First, check whether the Docker server is running:

```
$ ps ax | grep '[d]ocker'
```

If the process listing returns nothing, start the Docker server component by executing the following command based on your Linux distribution:

Systems with `systemd` init:

```
$ sudo systemctl enable docker.service
$ sudo systemctl start docker.service
```

Systems with `upstart`:

```
$ sudo service docker start
```

Step 3 – (optional step for Linux platform ONLY) make your user a member of the docker group

The Docker server listens for client requests on a UNIX socket, which is owned by the user `root`, and the group `docker`, with permissions set to `660`. Therefore, in order to send client requests to the server, you must either have an effective UID of `0` (logged in as `root`, or using `sudo`), or be a member of the group `docker`. To avoid having to prepend every docker command with `sudo`, you can become a member of the group `docker` by executing the following command, substituting `<username>` for your real username:

```
$ sudo gpasswd -a <username> docker
```

It will be necessary to logout and back in again in order for the change to take effect.

Step 4 – check that Docker is running correctly

In order to make sure that Docker has installed correctly and is running, execute the following command:

```
$ docker version
```

If all is well, you will get the following output, or similar:

```
Client:
 Version:      1.9.0
 API version:  1.21
 Go version:   go1.4.2
 Git commit:   76d6bc9
 Built:        Tue Nov  3 17:43:42 UTC 2015
 OS/Arch:      linux/amd64

Server:
 Version:      1.9.0
 API version:  1.21
 Go version:   go1.4.2
 Git commit:   76d6bc9
 Built:        Tue Nov  3 17:43:42 UTC 2015
 OS/Arch:      linux/amd64
```