The Docker daemon

After we've installed Docker, we need to confirm that the Docker daemon is running. Docker runs as a `root`-privileged daemon process to allow it to handle operations that can't be executed by normal users (e.g., mounting filesystems). The `docker` binary runs as a client of this daemon and also requires `root` privileges to run. You can control the Docker daemon via the `dockerd` binary.

NOTE Prior to Docker 1.12 the daemon was controlled with the `docker daemon` subcommand.

The Docker daemon should be started by default when the Docker package is installed. By default, the daemon listens on a Unix socket at `/var/run/docker.sock` for incoming Docker requests. If a group named `docker` exists on our system, Docker will apply ownership of the socket to that group. Hence, any user that belongs to the `docker` group can run Docker without needing to use the `sudo` command.

WARNING Remember that although the `docker` group makes life easier, it is still a security exposure. The `docker` group is `root`-equivalent and should be limited to those users and applications who absolutely need it.

Configuring the Docker daemon

We can change how the Docker daemon binds by adjusting the `-H` flag when the daemon is run.

We can use the `-H` flag to specify different interface and port configuration; for example, binding to the network:

Listing 2.37: Changing Docker daemon networking

```
$ sudo dockerd -H tcp://0.0.0.0:2375
```

This would bind the Docker daemon to all interfaces on the host. Docker isn't automatically aware of networking changes on the client side. We will need to specify the `-H` option to point the `docker` client at the server; for example, `docker -H :4200` would be required if we had changed the port to `4200`. Or, if we don't want to specify the `-H` on each client call, Docker will also honor the content of the `DOCKER_HOST` environment variable.

Listing 2.38: Using the DOCKER_HOST environment variable

```
$ export DOCKER_HOST="tcp://0.0.0.0:2375"
```

WARNING By default, Docker client-server communication is not authenticated. This means that if you bind Docker to an exposed network interface, anyone can connect to the daemon. There is, however, some TLS authentication available in Docker 0.9 and later. You'll see how to enable it when we look at the Docker API in Chapter 8.

We can also specify an alternative Unix socket path with the `-H` flag; for example, to use `unix://home/docker/docker.sock`:

Listing 2.39: Binding the Docker daemon to a different socket

```
$ sudo dockerd -H unix://home/docker/docker.sock
```

Or we can specify multiple bindings like so:

Listing 2.40: Binding the Docker daemon to multiple places

```
$ sudo dockerd -H tcp://0.0.0.0:2375 -H unix://home/docker/docker
  .sock
```

TIP If you're running Docker behind a proxy or corporate firewall you can also use the `HTTPS_PROXY`, `HTTP_PROXY`, `NO_PROXY` options to control how the daemon connects.

We can also increase the verbosity of the Docker daemon by using the `-D` flag.

Listing 2.41: Turning on Docker daemon debug

```
$ sudo dockerd -D
```

If we want to make these changes permanent, we'll need to edit the various startup configurations. On SystemV-enabled Ubuntu and Debian releases, this is done by editing the `/etc/default/docker` file and changing the `DOCKER_OPTS` variable.

On systemd-enabled distributions we would add an override file at:

`/etc/systemd/system/docker.service.d/override.conf`

With content like:

Listing 2.42: The systemd override file

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H ...
```

In earlier Red Hat and Fedora releases, we'd edit the `/etc/sysconfig/docker` file.

NOTE On other platforms, you can manage and update the Docker daemon's starting configuration via the appropriate `init` mechanism.

Checking that the Docker daemon is running

On Ubuntu, if Docker has been installed via package, we can check if the daemon is running with the Upstart `status` command:

Listing 2.43: Checking the status of the Docker daemon

```
$ sudo status docker
docker start/running, process 18147
```

We can then start or stop the Docker daemon with the Upstart `start` and `stop` commands, respectively.