Ensuring Your Web Server Starts on Boot

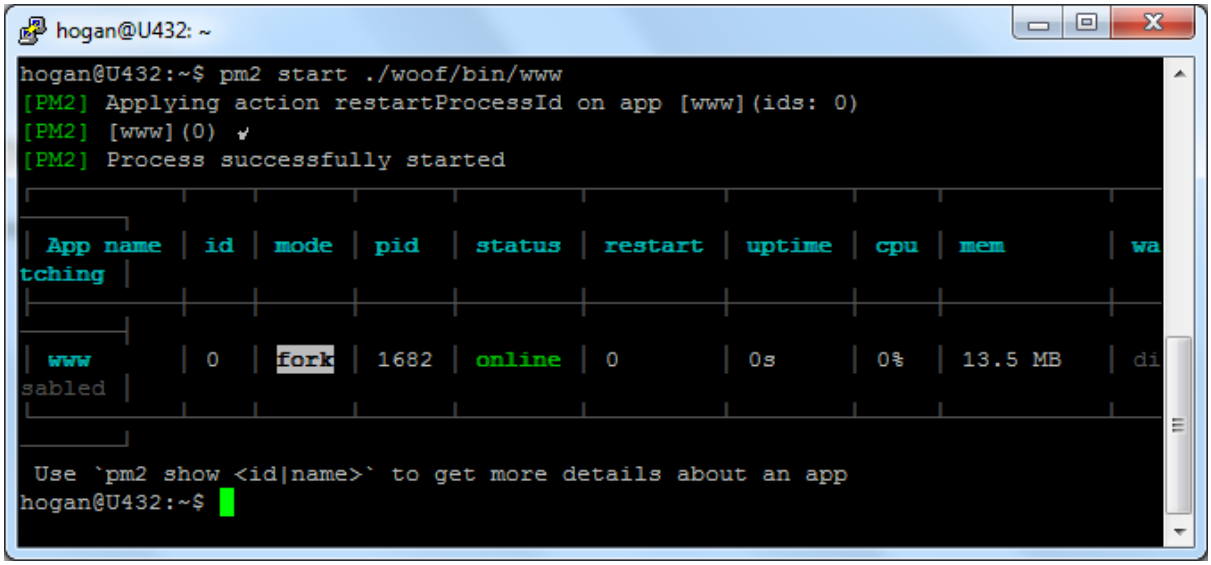
There are a number of ways of ensuring that the server starts when the VM boots. This is essential when we scale, but there are variations between operating system variants. My comments below refer only to Ubuntu. The approach is similar for others, but I have not looked at them.

Earlier in the semester we found that we could use a very simple script in the directory `/etc/int.d`. This is fine for earlier versions of Ubuntu, but the requirements for this script have become much tighter for later versions of the OS. While the linux sysadmin gurus will have their own solutions, I found pm2 to be a very good solution.

Install the application using npm, making sure to use the global option:

```
sudo npm install -g pm2
```

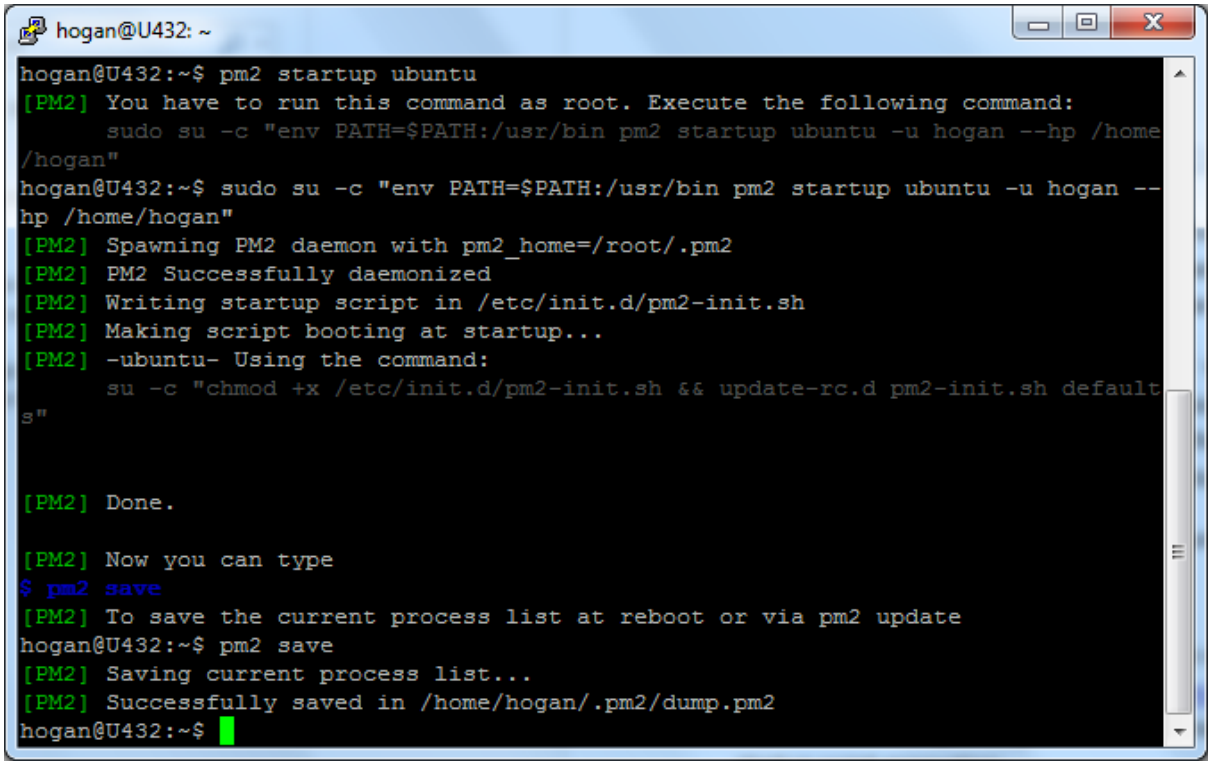
We then start the application in much the same we as we do with node. Make sure that the working directory chosen here matches the one used below. To save hassle, I did everything from my home directory `/home/hogan` connecting to the express app called woof:



What you see may differ slightly as I am restarting the app after stopping it. You should confirm that you can access the page at port 3000, using the IP address for the instance.

We then want to make sure that the app is set up to run on system startup. This is nicely covered at the site: https://futurestud.io/tutorials/pm2-restart-processes-after-system-reboot and elsewhere. If you look at this guide, ignore the material at the beginning and jump straight to the stuff about startup script generation.

In our case, we will indeed use the platform setting for Ubuntu, and follow their instructions to generate the startup script and to save the current set of processes. The approach is shown on the following page.



Confirm that you can still serve the page at port 3000 as before, and then use the console/portal or the appropriate CLI to reboot the machine. If all is well, you should still be able to find your web app at the same port as before.

More details of how to set up a node app for production may be found here:

https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-16-04

Note that the approach here is set up using node as a localhost server and then using nginx as a reverse proxy. This is not required for CAB432 but is very common in practice. You can see some of the arguments – though not from an entirely unbiased source – here:

https://www.nginx.com/blog/5-performance-tips-for-node-js-applications/