

Ensuring Your Web Server Starts on Boot

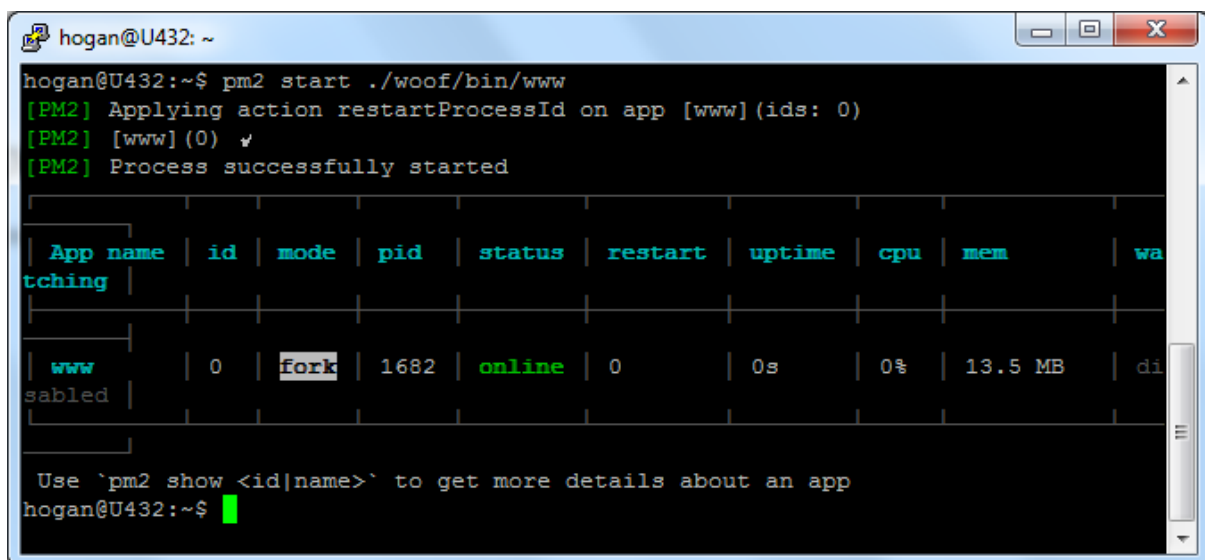
There are a number of ways of ensuring that the server starts when the VM boots. This is essential when we scale, but there are variations between operating system variants. My comments below refer only to Ubuntu. The approach is similar for others, but I have not looked at them.

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:



```
hogan@U432: ~$ pm2 start ./woof/bin/www
[PM2] Applying action restartProcessId on app [www] (ids: 0)
[PM2] [www] (0) ↗
[PM2] Process successfully started
```

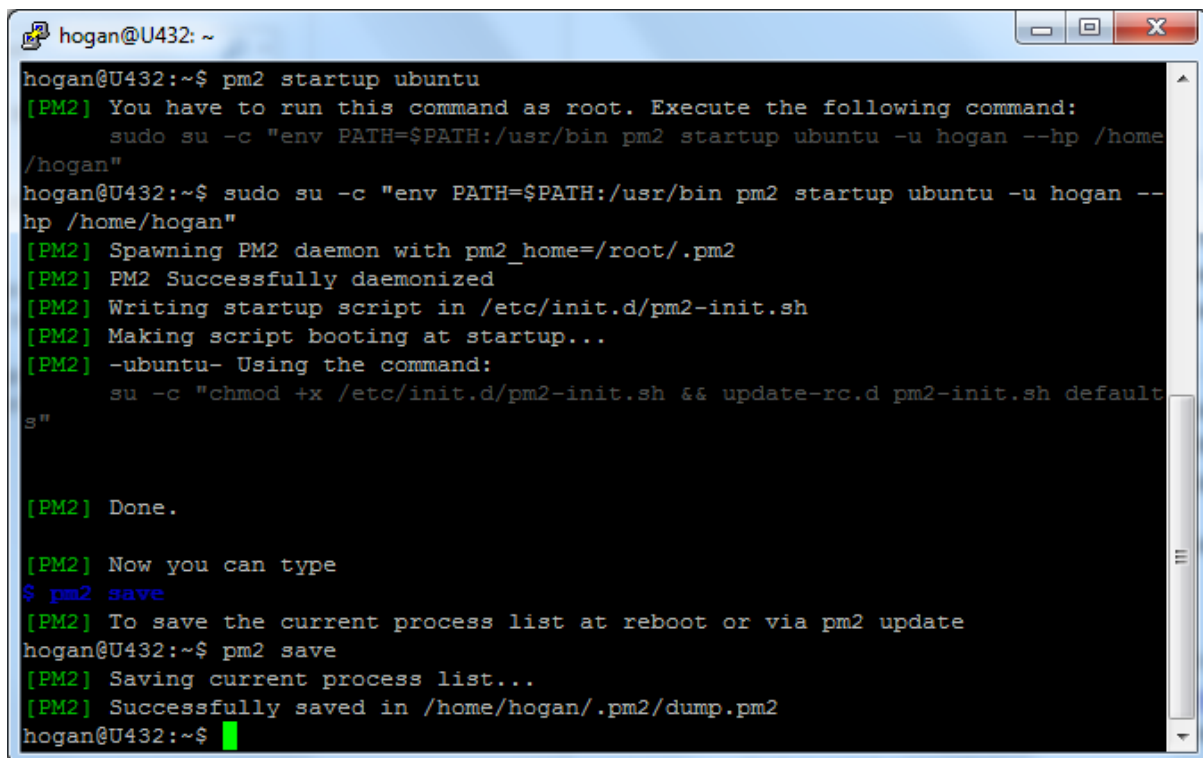
App name	id	mode	pid	status	restart	uptime	cpu	mem	watching
www	0	fork	1682	online	0	0s	0%	13.5 MB	disabled

```
Use `pm2 show <id|name>` to get more details about an app
hogan@U432:~$
```

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.

A terminal window titled 'hogan@U432: ~' with standard window controls. It shows the execution of 'pm2 startup ubuntu'. The output includes instructions to run the command as root, the command itself, confirmation of PM2 daemon spawning, script writing to /etc/init.d/pm2-init.sh, and the final 'pm2 save' command which saves the process list to /home/hogan/.pm2/dump.pm2.

```
hogan@U432:~$ pm2 startup ubuntu
[PM2] You have to run this command as root. Execute the following command:
      sudo su -c "env PATH=$PATH:/usr/bin pm2 startup ubuntu -u hogan --hp /home
/home/hogan"
hogan@U432:~$ sudo su -c "env PATH=$PATH:/usr/bin pm2 startup ubuntu -u hogan --
hp /home/hogan"
[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Writing startup script in /etc/init.d/pm2-init.sh
[PM2] Making script booting at startup...
[PM2] -ubuntu- Using the command:
      su -c "chmod +x /etc/init.d/pm2-init.sh && update-rc.d pm2-init.sh default
s"

[PM2] Done.

[PM2] Now you can type
$ pm2 save
[PM2] To save the current process list at reboot or via pm2 update
hogan@U432:~$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/hogan/.pm2/dump.pm2
hogan@U432:~$
```

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/>