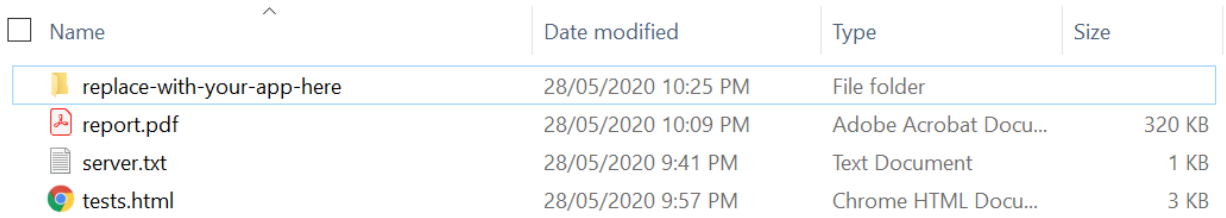
CAB230 Web Computing
Assignment 2 Submission Guide

This document presents a short guide to submitting the assignment on BB, and some instructions to ensure that your server remains up and running on the Linux VM. We will evaluate the functionality of your submission using automated tests and we will need you to make it available. We will talk about this below in Appendix A on PM2.

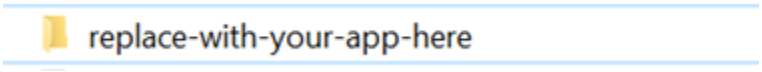
You need organise your code and supporting material into a standard directory structure. There is no demo video required for this assignment.

The Submission

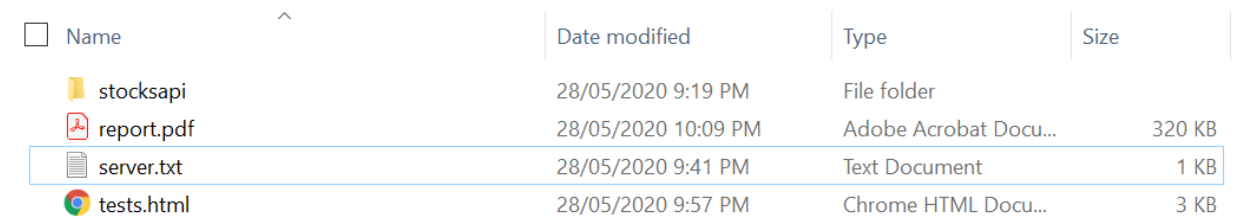
Start by downloading the file `assign.zip` which you will find nearby. This provides a template structure for you to follow. Unzip it, and you will see a structure like this:



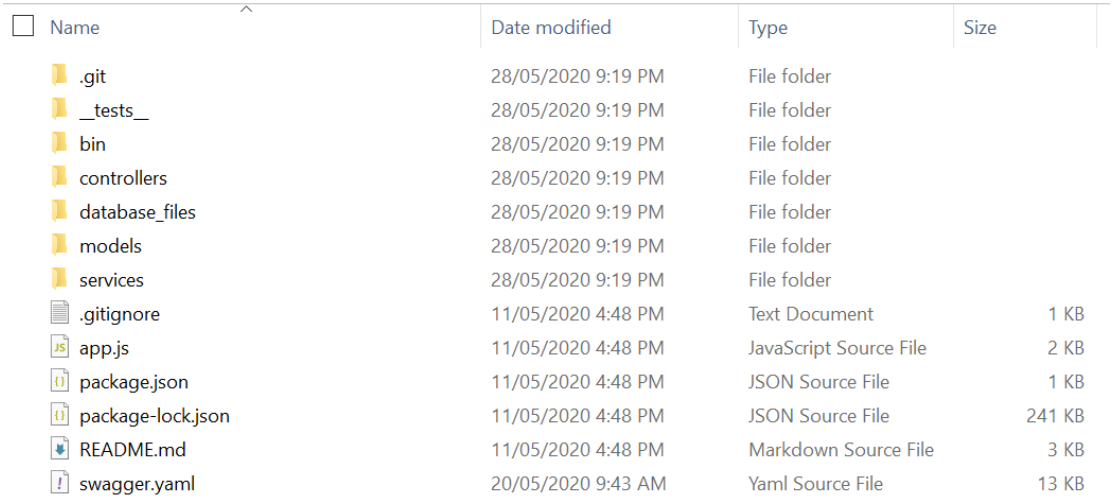
Now, each of these files is a dummy and it is your job to replace them. We will work through them in turn.



You should ***replace*** this folder with your app. *Don't* place your app inside this folder, just replace it. When we enter this folder we should see your `package.json` and `app.js` and the subdirectories that make up your application. In our case, we would have the following, with the folder `stocksapi`, but you can call your app what you like:



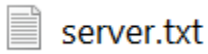
We now see what happens when we change directory into `stocksapi`:



Name	Date modified	Type	Size
.git	28/05/2020 9:19 PM	File folder	
__tests__	28/05/2020 9:19 PM	File folder	
bin	28/05/2020 9:19 PM	File folder	
controllers	28/05/2020 9:19 PM	File folder	
database_files	28/05/2020 9:19 PM	File folder	
models	28/05/2020 9:19 PM	File folder	
services	28/05/2020 9:19 PM	File folder	
.gitignore	11/05/2020 4:48 PM	Text Document	1 KB
app.js	11/05/2020 4:48 PM	JavaScript Source File	2 KB
package.json	11/05/2020 4:48 PM	JSON Source File	1 KB
package-lock.json	11/05/2020 4:48 PM	JSON Source File	241 KB
README.md	11/05/2020 4:48 PM	Markdown Source File	3 KB
swagger.yaml	20/05/2020 9:43 AM	Yaml Source File	13 KB

Again, your app may look different. But there are two things to note:

- This is the *top directory* of your app, not another empty layer…
- There is no `node_modules` directory. Just ***don't submit them***, please.



In this file we will get you to supply your student number and the *precise* URL of your server as it runs on the QUT Linux VMs. The instructions are in the file itself as shown:

```
1   #
2   # Lines that begin with a hash will be ignored.
3   #
4   # Please follow these instructions closely.
5   # You need to supply two lines only in this file.
6   # On the first line, you must give us your student number.
7   #
8   # On the second, please include the *precise* URL for the server on your QUT Linux VM
9   # Do *not* give us an entry for HTTP and HTTPS -- just one please
10  # Specify the port if it isn't the default: 80 for HTTP or 443 for HTTPS
11  #
12  # ****DO NOT USE ANY version of http://localhost or https://localhost****
13  # You must use an explicit ip address.
14  #
15  # Please edit the dummy entries below and save the file.
16  # Final check -- **don't** leave the student number as n1234567
17  #             -- make sure your URL is the correct one and not full of xxxs and yyys
18  #
19  n1234567
20  https://172.xxx.xxx.xxx:yyy
21
```

The main thing to remember is that you must edit the lines that have been highlighted or otherwise we won't be able to test your API. Note in particular that you ***must not use*** `http://localhost` or `https://localhost` or any variation of this address.

---

report.pdf

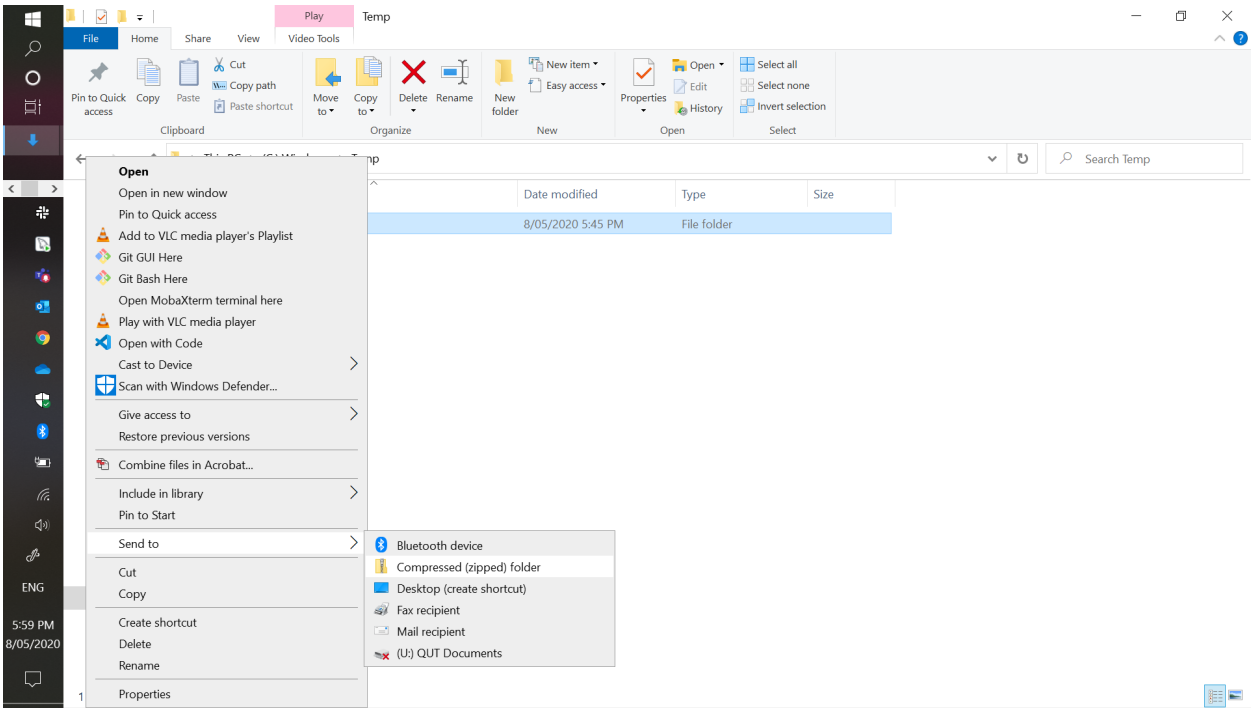
The report document is a single-page dummy version of the guide. There is nothing in it. You will need to replace it with your report based on the instructions.

---

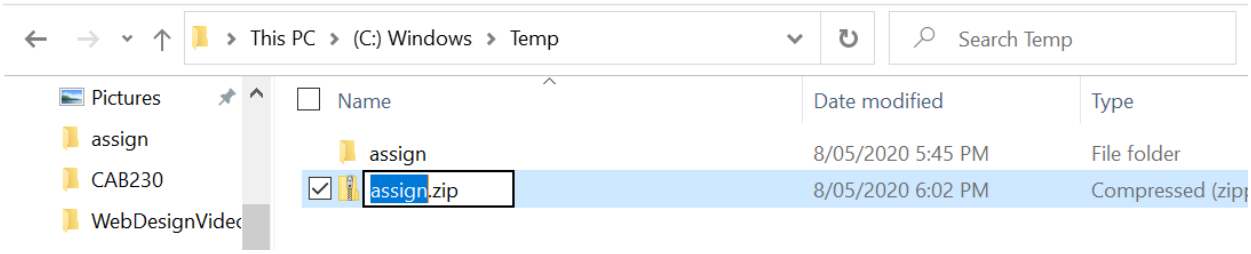
tests.html

Optionally, you may include `tests.html` as *your* copy of the `report.html` you obtained when running our tests. We will only look at it as a cross check if something odd is happening. And please do not fake it. We will know and we will *not* be happy.

Once you have performed a final check that you have deleted the `node_modules`, you should follow the same instructions as for assignment 1. Zip up the directory and submit the (now heavily revised) `assign.zip`. The instructions from last time are as follows:



The resulting `assign.zip` file is your submission file, and should be uploaded to BB via the submission link. You are allowed as many attempts as you wish for this assignment.



We will mark the last attempt received and we will ignore all the others.

The deadline for the submission is 11:59PM on Friday June 5. The submission link will remain available until 12:30 AM on Saturday June 6 in case of congestion on the site. After that time it will disappear. It is your responsibility to ensure that you have a submission successfully in the system before submissions close. Those with extensions will be allowed to submit some days later from a similar link.

Appendix A – Using pm2

During marking, we will need to fire our tests at your server. In the submission guide above, we require that you tell us your machine's IP address so that we can query your API. But we need your application to be available. In order to do this, there are two things to remember:

- Disconnect from your VM and do not logout or shutdown.
- Make sure that you run the server application as service.

It is more professional to use a process manager like pm2 to mange your node applications. In the case of a server, it is common to set things up so that it will start on boot – restarting if the server goes down.

Here we will give you two alternatives – a simple one for those in a hurry, and the longer story. The longer story is well covered in the Digital Ocean walk through linked below:

https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-18-04.

Whatever we do, we must begin with a global installation of `pm2`:

```
$ sudo npm install pm2@latest -g
```

You will need to ensure that your internet access is enabled via the IAClient app if the installation is to work properly. Once the application is installed, we have two options.

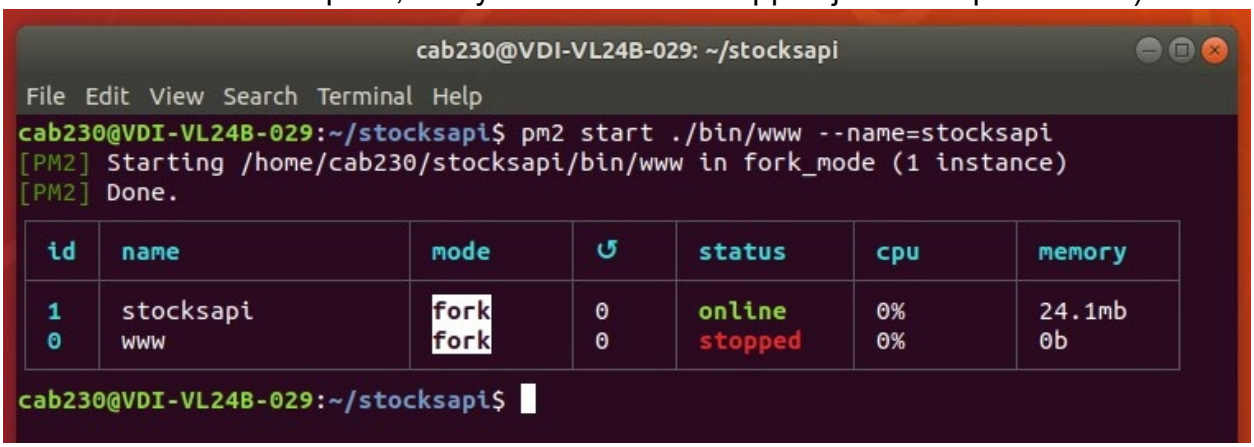
Running the app in the background – no restart on boot

This is by far the easier approach. We use `pm2` to start the app in the background. If you have deployed the secure version of the server then you need to remember that your application cannot serve https without elevated privileges. You will need to use `sudo`.

For Express applications, the executable is usually found in `./bin/www` and accessed via `npm start`. The simplest approach is to use something like:

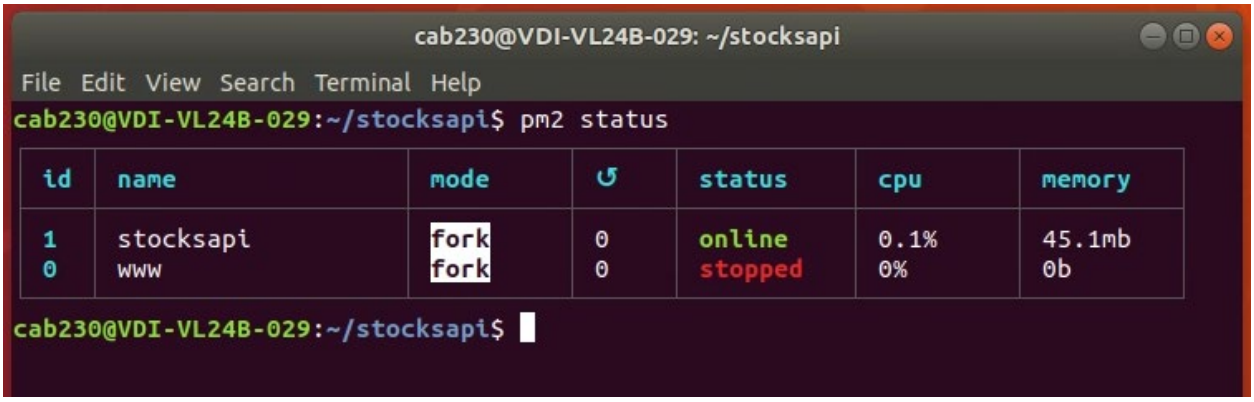
```
$ sudo pm2 start ./bin/www --name=stocksapi
```

Here we are using the `--name` option to choose a recognisable name for the process - you can choose anything you like. You will then see something like this (the previous run didn't have the name option, and you can see the stopped job in the process list):



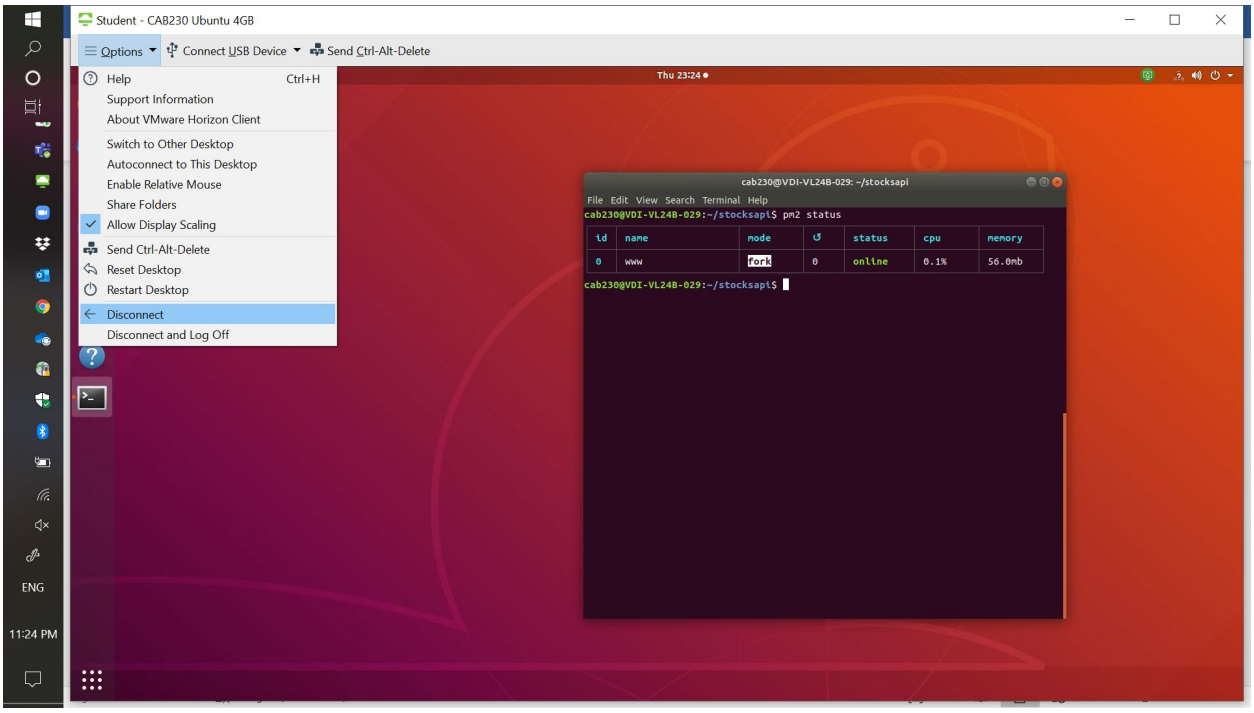
You should ensure that the application is still running:

```
$ pm2 status
```

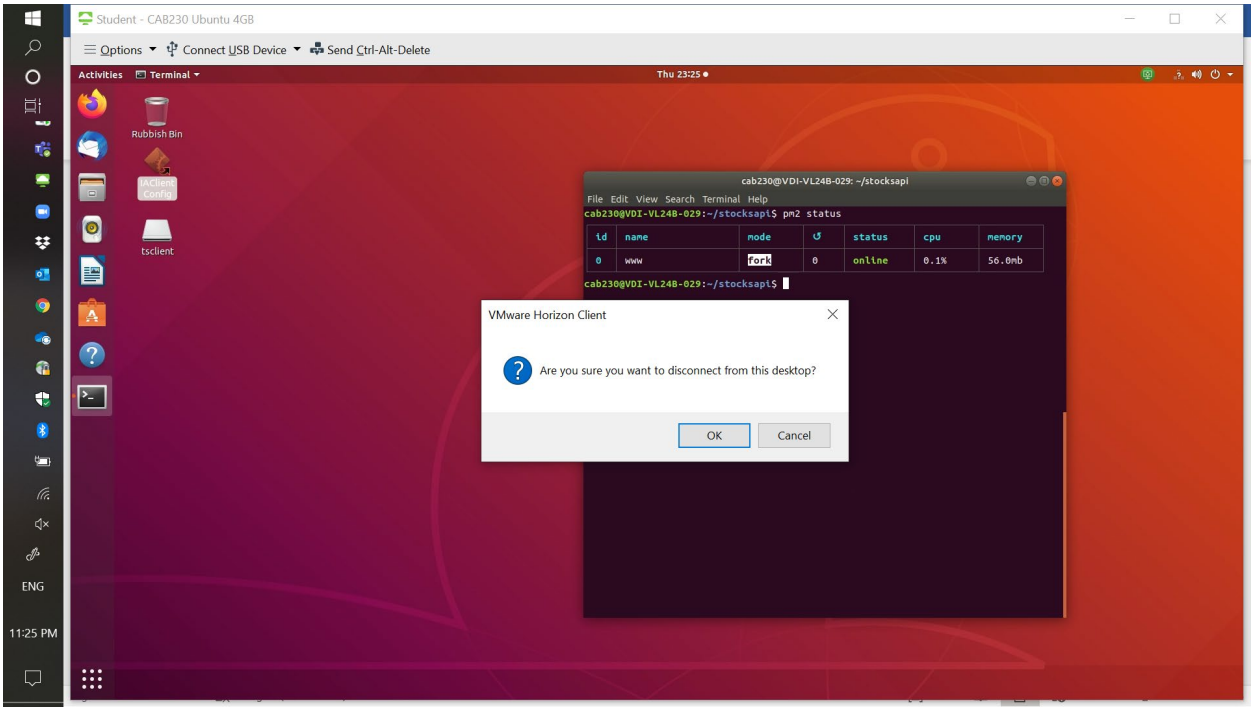


Also check directly in the browser – make sure that the Swagger docs are visible.

At this point, your task is to ensure that the machine remains running. Our advice from QUT IT is that the machines will remain up until we tell them to turn them off. So if you have reached this point, you may simply disconnect from the machine as shown:



This will ensure that the machine retains the same IP address and will be visible to us when we come to mark it. Click on the cross in the top right hand corner and you will see:



This has exactly the same effect. If you *logout* from the machine then it is likely that the IP address will be re-assigned. If you *reset* the machine then you will lose all your work. Disconnection is your friend. It is perfectly ok to leave it at this point, making occasional checks that the machine is running after submission.

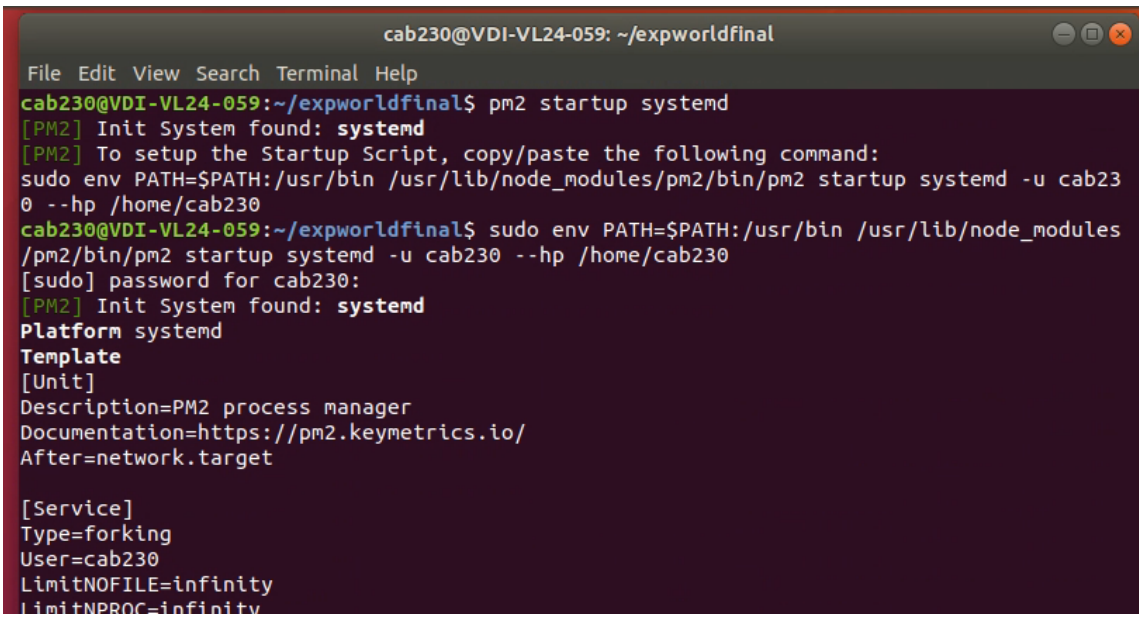
One final check that is definitely worth doing: edit the `integration.tests.js` file in your copy of the tests and make sure that the URL points to this server on your VM. Then go ahead and run the tests and make sure that the results are the same as you obtained locally. That will save us all a lot of time.

Optional: Enabling restart on system startup

If you want to do things a bit more professionally, you need to use pm2 to record the app in the list of processes to be executed on system startup. Once you have the application running as in our instructions above, you need to run the following command:

```
$ pm2 startup systemd
```

Your task is then to copy and paste whatever statement is generated (the screenshot shows an earlier version from within `expworldfinal`. The process is exactly the same):



The output continues here for some time… Yielding ultimately:

```
Target path
/etc/systemd/system/pm2-cab230.service
Command list
[ 'systemctl enable pm2-cab230' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-cab230.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-cab230...
Created symlink /etc/systemd/system/multi-user.target.wants/pm2-cab230.service → /etc/s
ystemd/system/pm2-cab230.service.
[PM2] [v] Command successfully executed.
+---------------------------------------+
[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup systemd
cab230@VDI-VL24-059:~/expworldfinal$
```

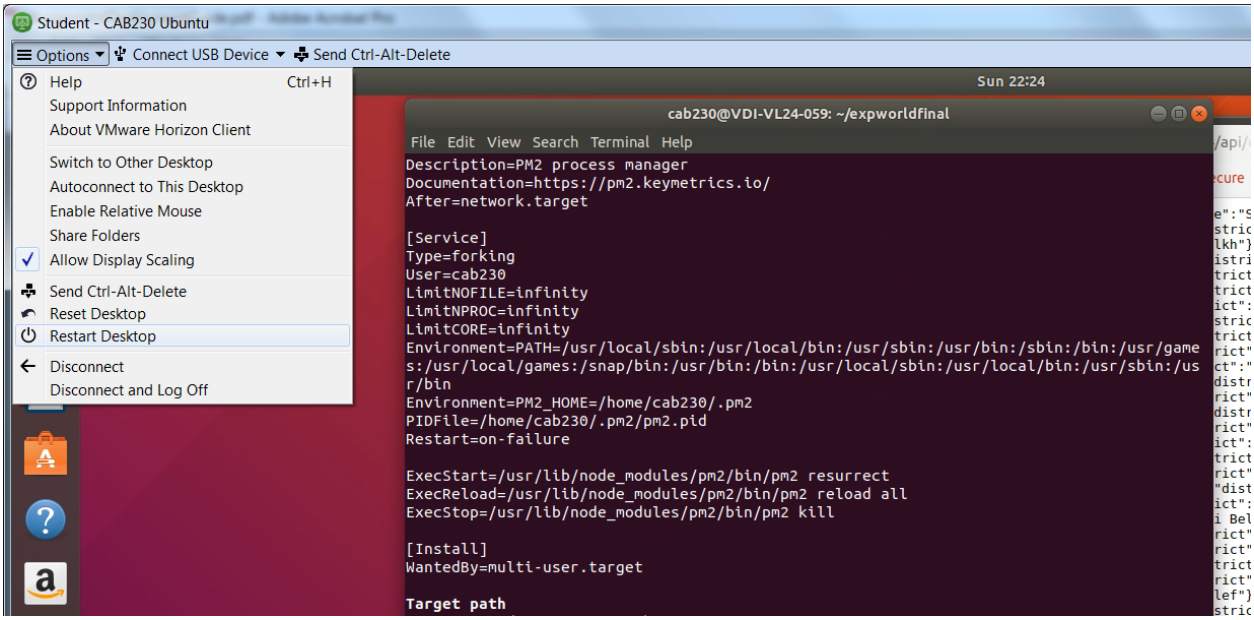
The result is that the process is ready to be executed. Preserve the list using:
`$ sudo pm2 save`

Here `sudo` is needed due to the earlier root level process. As noted in the Digital Ocean guide we can then:

- Start: `sudo systemctl start pm2-cab230`
- Check status: `systemctl status pm2-cab230`
- Check status: `systemctl status pm2-cab230`

And we can start or stop the app using the appname or ID.

Now, you can restart the desktop, but here you might get a different IP address. For now, please just look at the screenshots, or at least be prepared to update the IP address that you submit in the `server.txt` above:



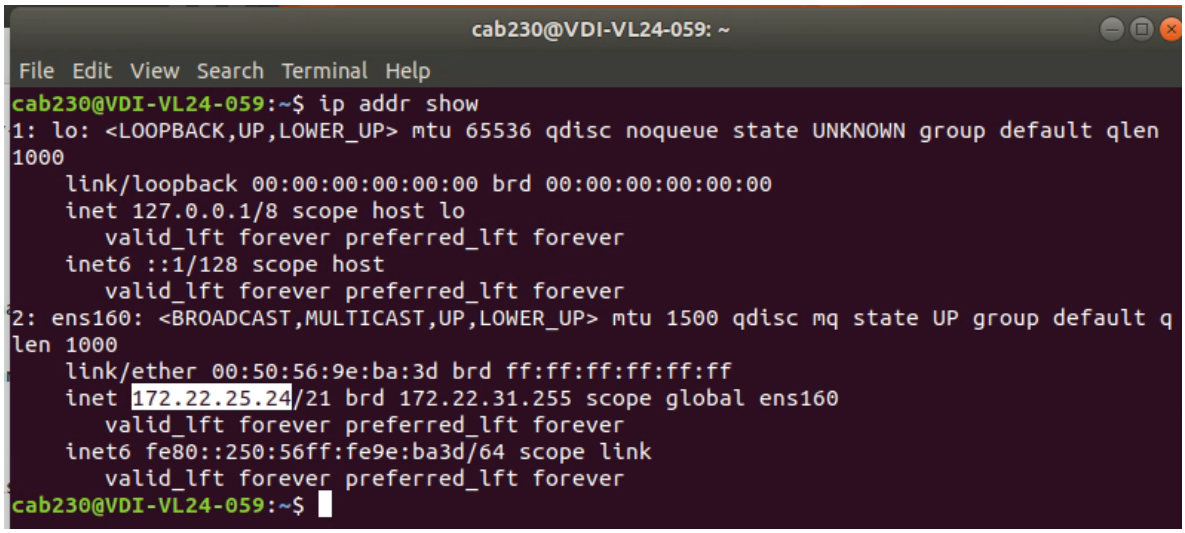
After a few minutes, we reconnect to the machine and continue. As noted above, for now it is better if you just disconnect and you do not logoff or restart.

Those who want a deeper understanding of these issues should take a look at the Digital Ocean guide linked above. If you have forgotten how to find your IP address, please take a look at Appendix B below.

Appendix B – Your IP Address

This appendix is copied directly from the Deployment Guide for your convenience. You will need to add the protocol (http or https) and if necessary, the port number.

This short appendix covers finding the IP address of your VM from within Ubuntu. As you will be within the QUT network, your machine is not internet facing and so the IP will be assigned within the ranges expected for a local network. However, your machine will be accessible elsewhere on the QUT network for the appropriate requests. Probably the quickest way to find your IP address is to use terminal:



The command shows the results for a number of network interfaces, with the top one obviously localhost. The highlighted address is the one you want.