**R00171867 Dipendra Ale ITM-B**

**Internet & Network Services Assignment 1**

In this assignment, as a systems administrator we have been asked to build a secure server to host an application. The application that I chose to host is Joomla.

**Required software required for the Assignment**

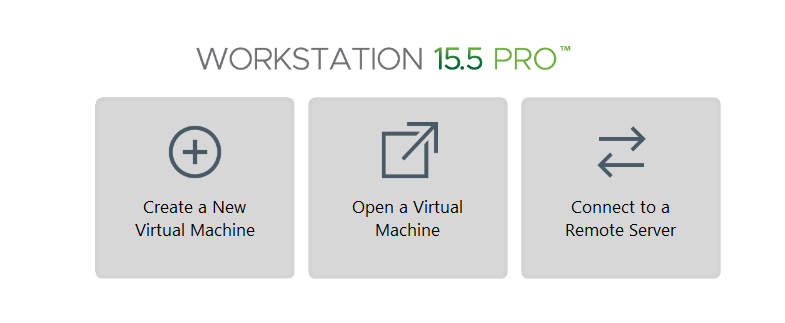
1. VMware Workstation 15.5

<http://mcom.cit.ie/kms/software-licenses/vmware-academy>

1. Ubuntu Desktop client 18.04 LTS & Ubuntu Server 18.04 LTS

<http://releases.ubuntu.com/18.04>

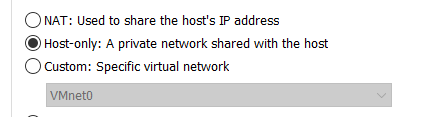
After these required software’s have been downloaded, we are going to create Ubuntu Client and Server Virtual Machine, so we are able to communicate with each other.



**Configuring the VMs**

After the installation of the server and client VM’s are complete we need to configure the networks settings for both to enable communication between two.

First off, On the server You need to change the VM settings so that VMs Network Adapter is set to Host-Only rather than NAT.



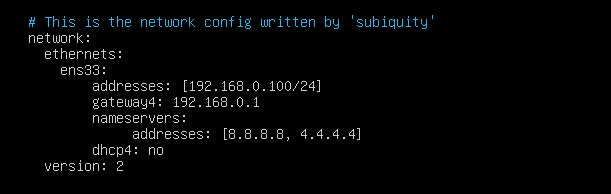
Now, we need to type these commands on the cmdlet to give the server the correct IP settings:



(create a backup of the config file)



Change the nano file so it looks like this.

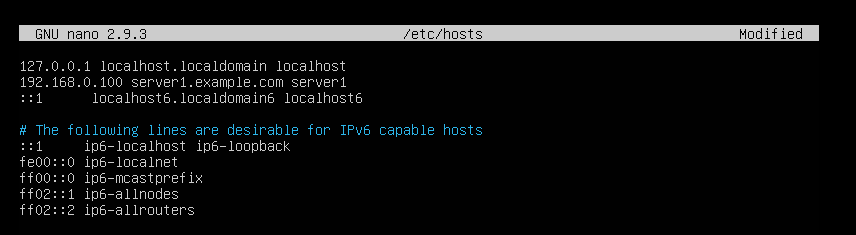


(To apply changes)

Now we assign a hostname to the server:

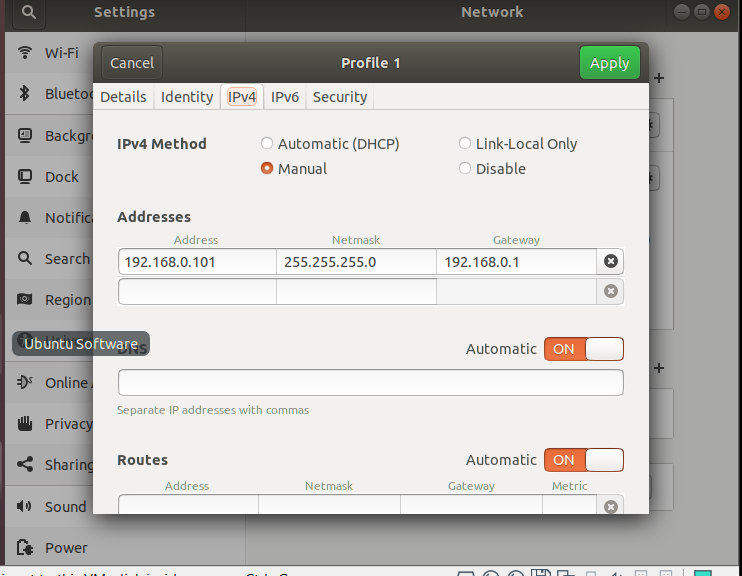


Change the nano file so it looks like this.



Now on Client VM we again need to change VMs Network Adapter to Host-Only rather than NAT like we did on server VM. We are also going to give the client the correct IP settings:

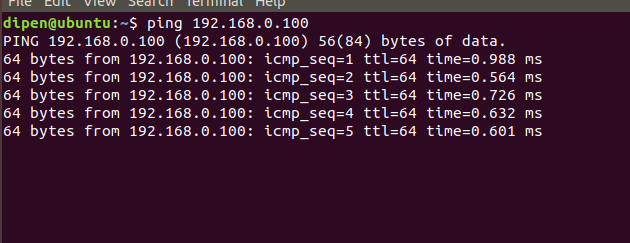
* Click the networking icon on the top right-hand corner of the screen & then go to Wired connected.
* Choose the Wired settings and click on the gearwheel.
* Click on IPv4 settings & then changed the method to manual.
* Click add & enter the following IP information like below and click apply.



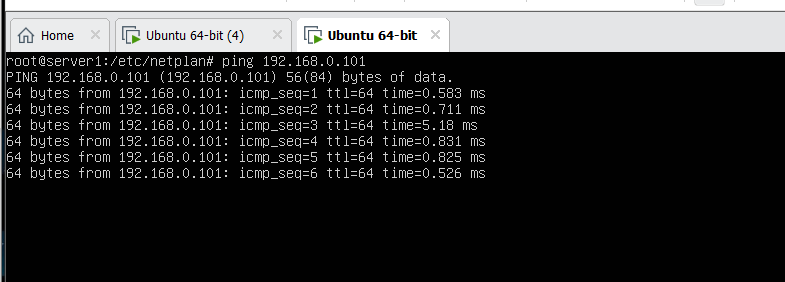
* Disable and re-enable the interface.

Lets now test that we have connectivity:

On client VM:



On server VM:

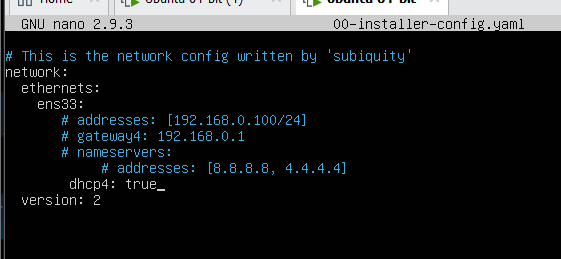


Every time you create a new VM after you have finished your install this should be the first thing to do and if u want to connect to web to download packages then u need to set the VMs Network adapter to NAT again. You also have to change the 00-installer-config.yaml file & comment out static IP settings & put back in the setting for DHCP.

However, when you have the adapter set to NAT you VMs will no longer be able to communicate. Therefore, u will need to switch back & forth between NAT and Host-Only and also change the 00-installer-config.yaml file.

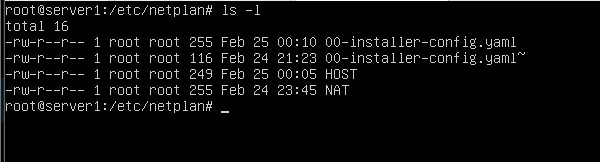
This is why its worthwhile creating two files HOST and NAT with different configuration in the /etc/netplan folder.

  
Now change the 00-installer-config.yaml file like following:

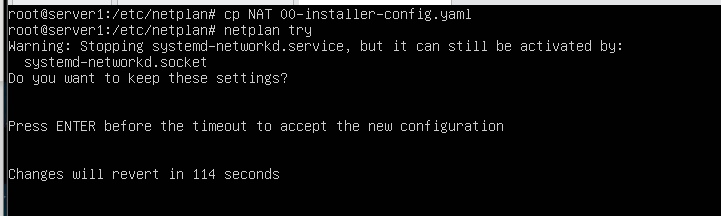




You should see the two files HOST and NAT files if u do ls.



Finally, the last part to test out if we can connect to the internet by changing to NAT configuration:



We can confirm this by pinging 8.8.8.8 in the server VM.

To invoke the changes between HOST-ONLY and NAT type in netplan try and ifconfig can be used to confirm IP address settings. This concludes configuring the client and server VMs.

**Securing the Linux server using Ubuntu**

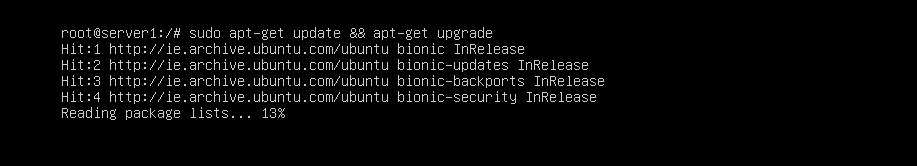
First off, we are going to discuss why it is important and worthwhile to secure your Linux Server. Well, the main reason and the most generic answer is to protect our data from the hackers. By 2020, the cost of a data breach amounts to a whooping $150 million and no industry is safe from such violation. Securing server doesn’t require to be complicated. We should adopt a method that will protect our servers from the most frequent attacks along with efficient administration. However, don’t take things for granted. Even the most hardened servers can be hijacked by exploiting any vulnerable component running on that server.

**Update your package list and upgrade your OS**

Software updates and patches are often distributed to fix security vulnerabilities as they’re discovered. Running outdated software puts you at risk as soon as the details of the vulnerability are published. For that reason, it’s vital to make sure your packages and OS are constantly updated and as secure as they can be.

Type this command as a root:

apt-get update && apt-get upgrade

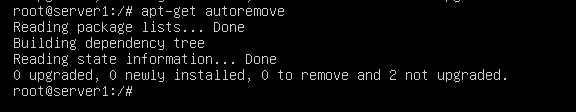


**Install what u need**

The first rule is to always keep your server lean and mean. Install only those packages that you really need. If there are unwanted packages you may want to get rid of them. The fewer the packages the less chance of unpatched code.

Type in:

apt-get autoremove



Next, check apt-cache pkgnames & delete anything you don’t need with sudo apt-get purge –auto remove [packagename]

Note: Install deborphan to automatically detect unused packages.

**User Management**

User management is another very important aspects of any security plan. Balancing your user’s access requirements against their everyday needs, versus the overall security of the server will demand a clear view of those goals to ensure users have the tools they need to get the job done as well as protect the other user’s privacy. There are three types or levels of user access:

1. Root: This is the main administrator of the server. The root account has full access to everything on the server.
2. A sudoer(user): This is a user who has been granted special access to a Linux application called sudo and it has elevated rights to run a function or program as another user.
3. A user: This is a regular user who has been set up using adduser command and given access to and, who owns the files and folders within the user /home/user/directory.

**Setup SSH Keys**

SSH keys allow you to connect to the server securely with a stored key pair. This would be an extra step in securing the server to disallow additional access. SSH into your server as the root user.

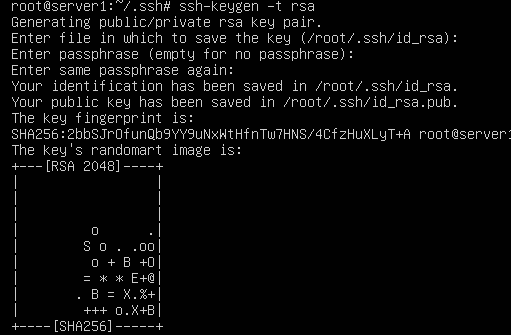
Cd to shh folder



Next, run to create ssh key to login into the server:

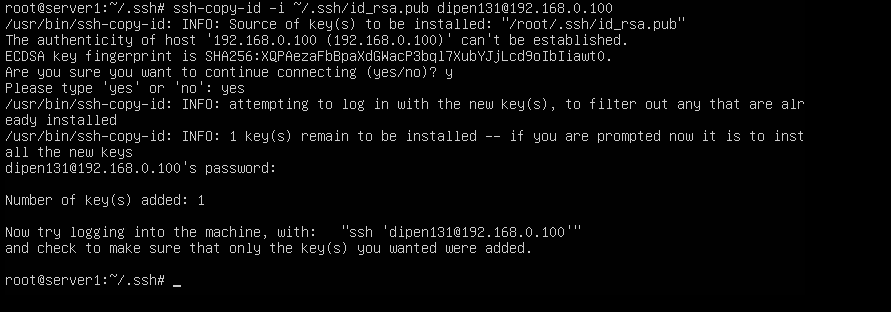


Press enter key to accept the default locations and file names, then enter, and then re-enter a passphrase for your user(You can leave the password empty if u don’t want to set it).

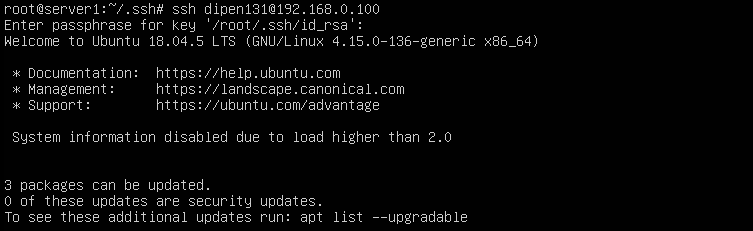


Next copy the new public key to your users SSH directory on the server.

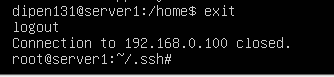




Try logging in to the SSH server as instructed (dipen131@192.168.0.100):



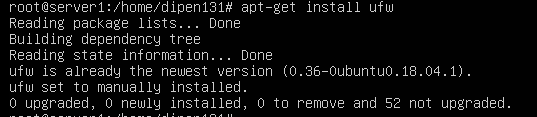
We have now successfully logged into the ssh server



**Install and configure UFW**

Uncomplicated firewall (UFW) is an interface for iptables that offers an easier way to regulate incoming and outgoing traffic. With a few quick commands, you can block all traffic that isn’t going to your web server, through SSH, FTP etc. This drastically reduces your risk while connected to the network.

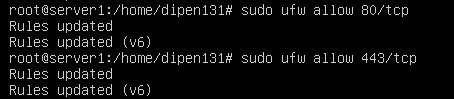
Install ufw if u have not already: apt-get install ufw



Enable port 22 , 80 and 443:

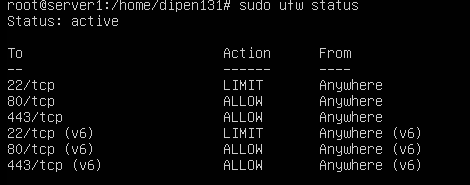
The limit command limit how many connections can be applied so u don’t have to worry about brute force attacks on these port. This is why we are doubling up the firewall.





Now finally, enable ufw



You can check the ufw status: 

We are now going to setup the defaults and and add in extra rules to deny incoming connections and allow outgoing connections:





And reload ufw to apply these changes



**Configure sysctl securely**

You default sysctl settings may leave you open to syn flood attacks and IP spoofing, or may not log suspicious packages.

To harden your sysctl settings, open /etc/sysctl.config in a text editor, and do the following:

* Uncomment this for spoof protection



* Uncomment this to block ping requests which prevents MITM(man in the middle attacks) and also block when you are not router.

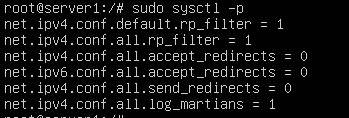




* Uncomment Martian packets to log it.



Now check the changes you made:



**Disable interactive hotkey startup at boot**

Another way you can secure your server from an attacker with physical access to disable interactive hotkey startup. With this turned off, you limit the possibility of an attacker disabling system services including firewalls.

The interactive boot settings are set in /etc/sysconfig/init. Simply add this line to the file or edit the variable to match: PROMPT=no

**Secure any Apache server**

Apache servers make for a wonderfully predictable entry point for attackers- you only have to check the number of Apache vulnerabilities that have surfaced over the years to be wary about the security of yours.

Open the Apache config file – located at /etc/apache2/apache2/conf and modify the default settings to :

* ServerTokens Prod
* ServerSignature Off
* Header always unset X-Powered-By

These three lines prevent Apache from broadcasting its versions number and other identifiable details which makes it harder for attackers to exploit known vulnerabilities.

**Installing moodle application**

Before we begin the installation, we need to make sure we change the network configuration setting to NAT rather than HOST so we can connect to the internet to be able install this application.

We will first download vim (heavyweight editor) to edit or create new files.

The basic vim commands:

Insert Key or I = to edit the file

ESC Key = to finish editing

Type: q = to exit the editor

Type: wq = to save and exit the editor





**Step 1: Install apache/MySQL/PHP**



**Step 2: Install Additional Software (php7.2)**



Restart Apache so that the modules are loaded correctly



We will be using Git to install/update the Moodle Core Application



**Step 3: Download Moodle**

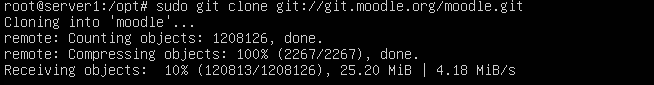
Setup your local repository and download Moodle, We will use /opt for this installation.

We are going to use Git (version control system) which will be much easier to update the moodle core application.

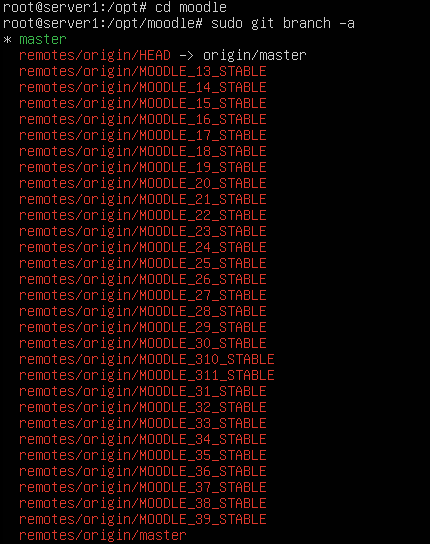
Having your local repository outside of the webroot, like we have in this step, you will be able to prepare and stage your upgrades in a more efficient manner.



Download the Moodle Code and Index



Change directory into the downloaded Moodle folder and retrieve a list of each branch available like shown in the snippet.

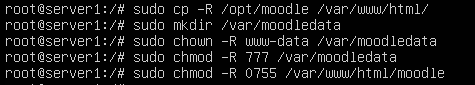


We now tell git which branch to track or use

Finally, Check out the Moodle version specified



**Step 4: Copy local repository to /var/www/html/**



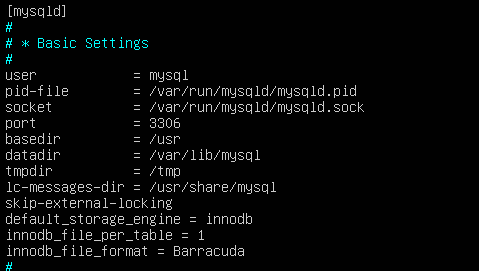
Since we setup a local repository in the previous step, you will copy it to your webroot after any updates and making changes.

**Step 5: Setup MySQL Server**

Firstly, we need to change the default storage engine to innodb and change the default file format to Barracuda, this is a new setting compared to previous versions. You also need to set innodb\_file\_per\_table in order for Barracuda to work properly.

Open the mysqld.conf file using vim and scroll down to [mysqld] section and under Basic Settings add the following as shown on the snippet under the last statement. Press I key to write, press ESC and type: wq to save and exit vim.

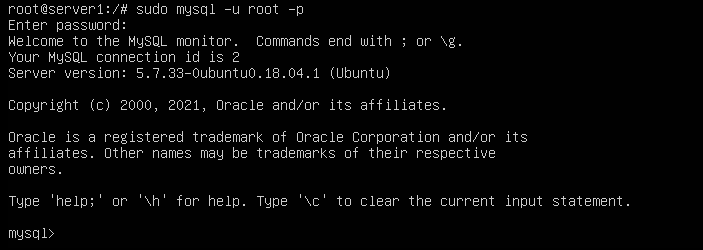




Then restart MySQL Server for changes to take affect



Now we will create the Moodle database and the Moodle MySQL User with the correct permissions. Use your root password here.

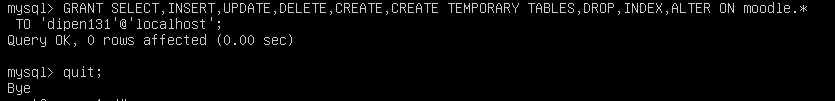


Now lets create database moodle.



You should replace ‘dipen131’ and ‘123’ to your preferred username and password.





**Step 6: Complete Setup**

Note : If you are not comfortable using terminal to create the config.php file that needs to be created when going through the installer, you should temporarily make the webroot writable so that it will create the file automatically for you.



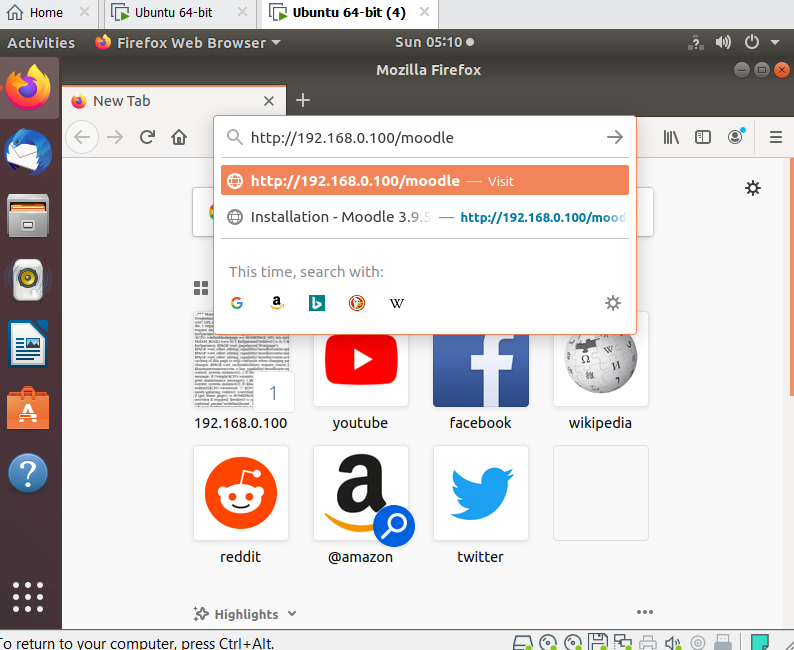
And after you have ran the installer and you have the moodle setup, you will need to revert permissions so that it is no longer writable using the below command.



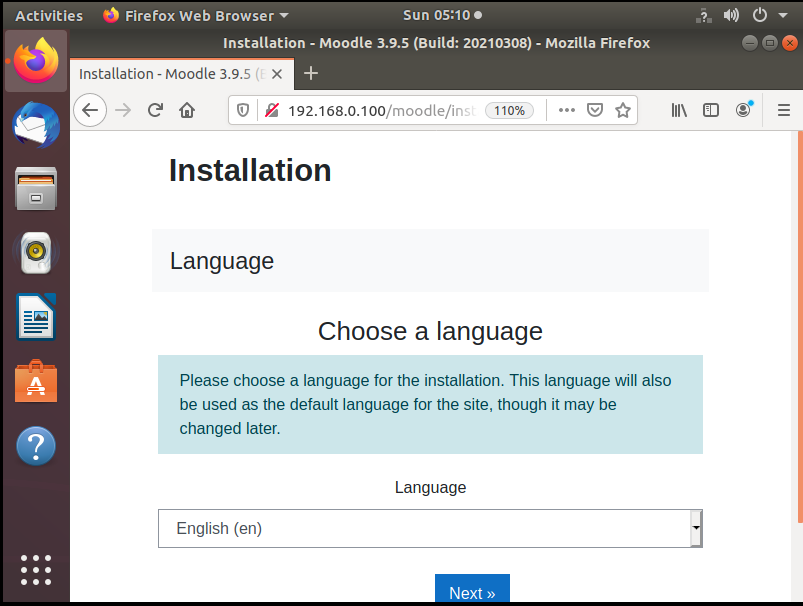
Now go to the client ubuntu and open up Firefox.

Before you do this make sure you change your network settings configurations to HOST so that the server and client can communicate.

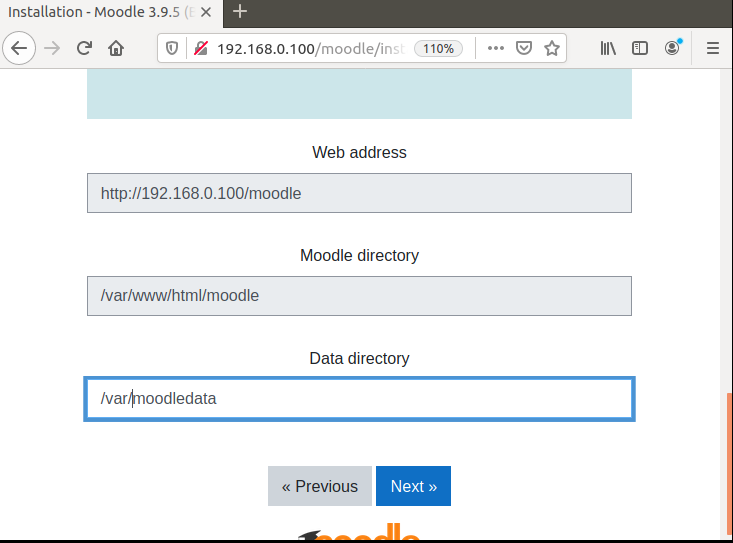
Type in http://youraddress/moodle



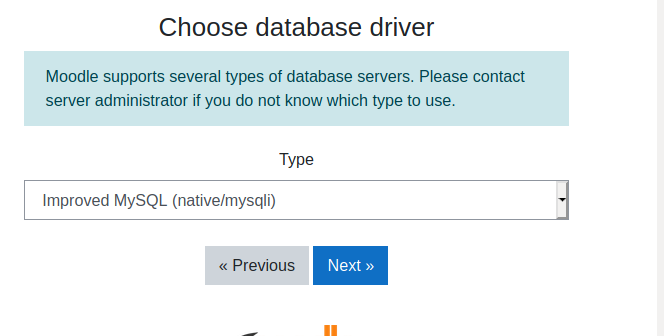
Choose your preferred language

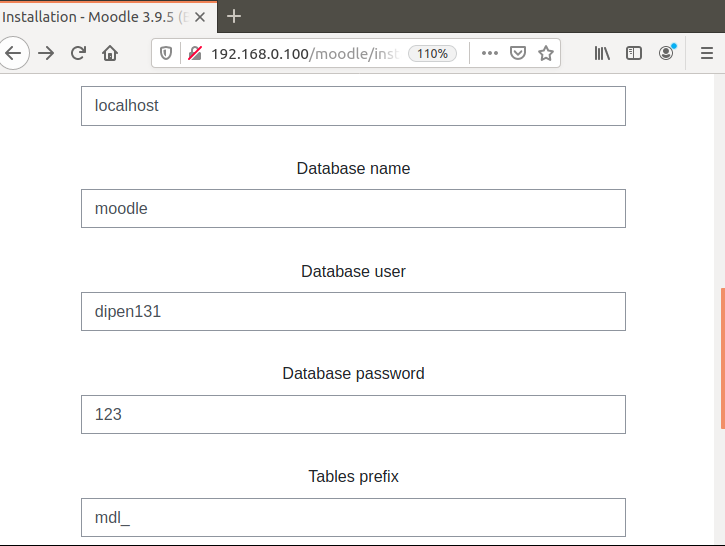


Change the path for moodledata to /var/moodledata



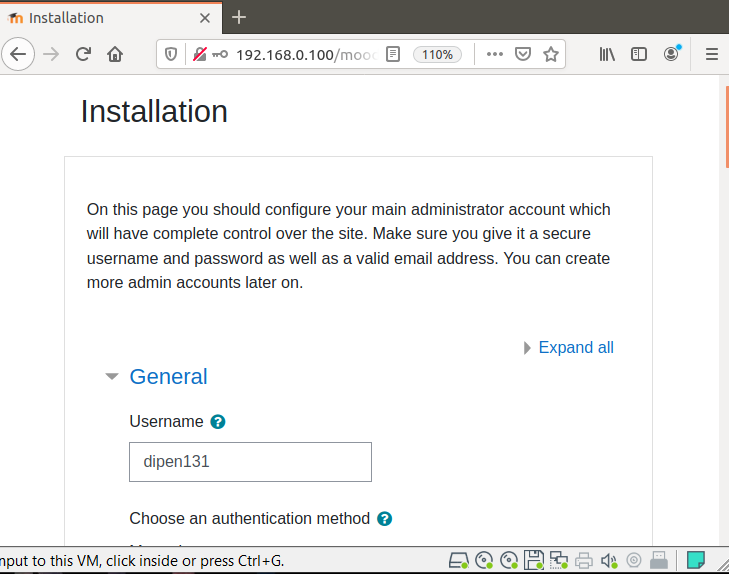
Choose Database Type : mysqli

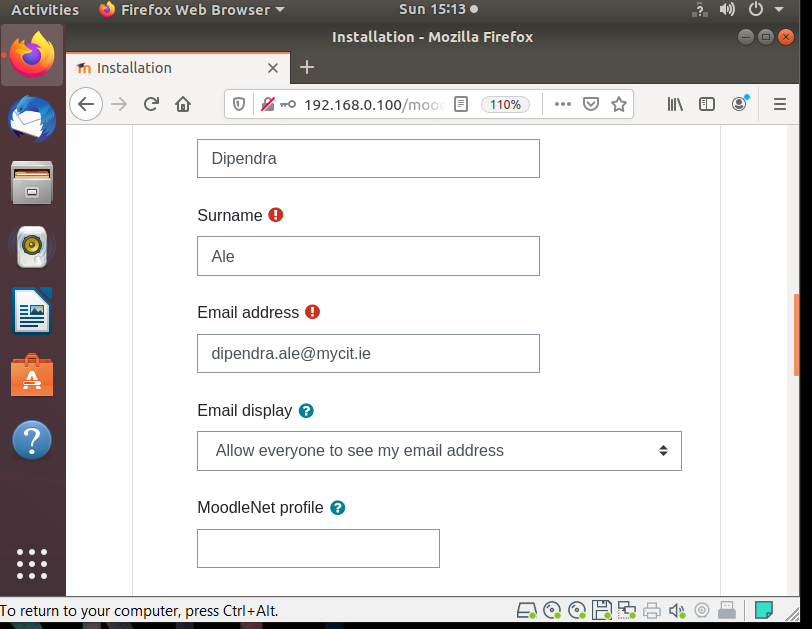


Change Database Settings as shown : Type the user that you created when setting up the database and also enter the password that you created.

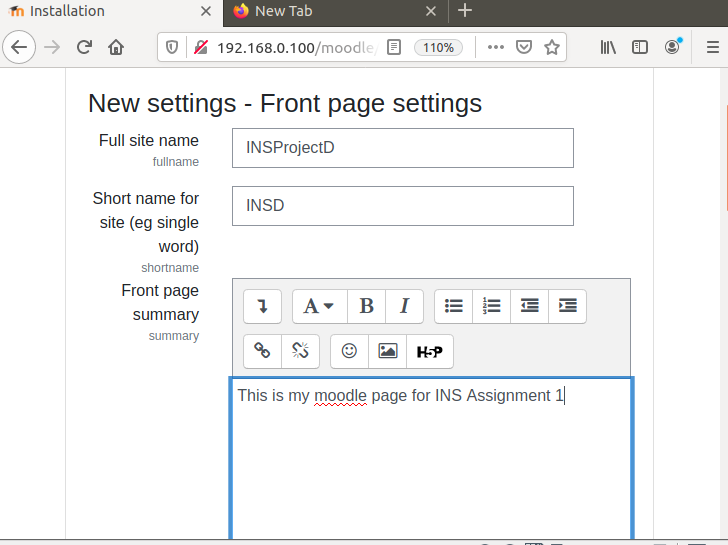
Enter Next Next Next and follow prompts and confirm installation

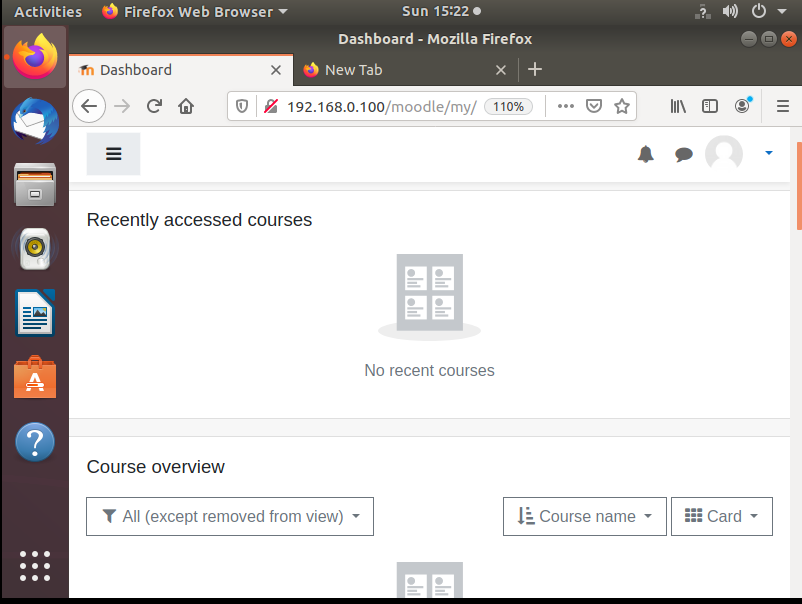
Create a Site Administrator Account





Create a new settings





**Installation complete**

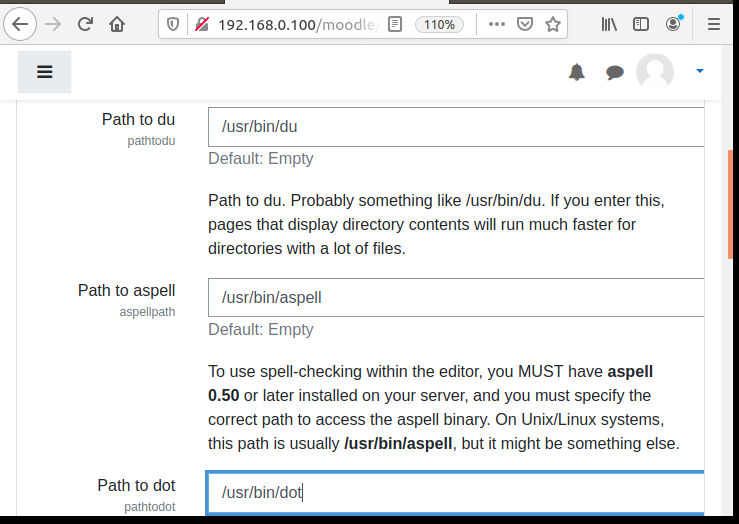
Don’t forget to revert permissions if you made the webroot writable.

You can now customize your Moodle to your likings.

**System Paths After Install**

After installing Moodle you should set the system paths, this will provide better performance VS not setting them. Each entry in Moodle will have its explanation. To do this: go to moodle webpage -> Site Administration -> Server ->System Paths

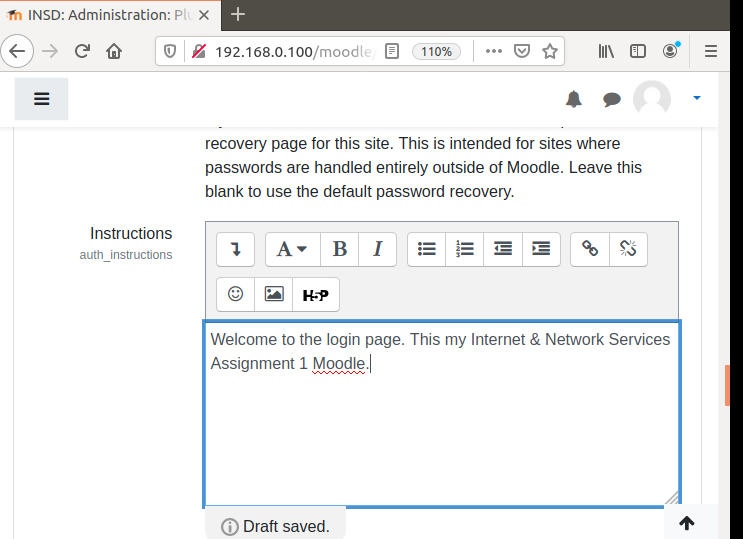
Input these to the paths and save changes



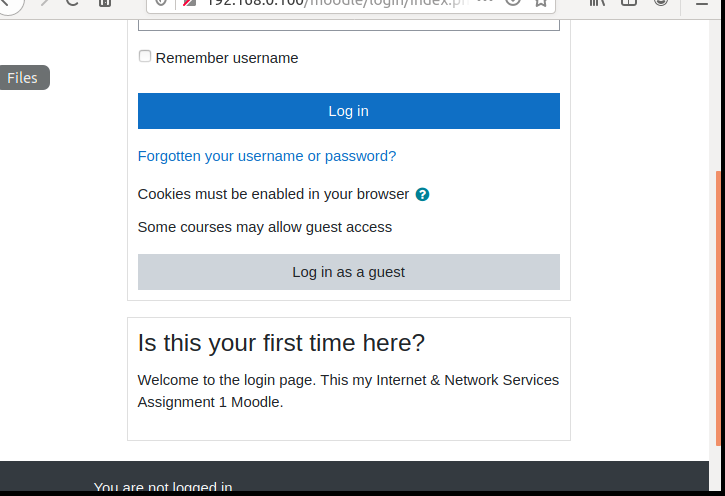
**Adding custom welcome message to Moodle’s login page**

Navigate to the Manage authentication option: Settings >Site administration >plugins >Authentication >Manage authentication >Scroll down to this editor:

Add in your preferred message and apply changes:

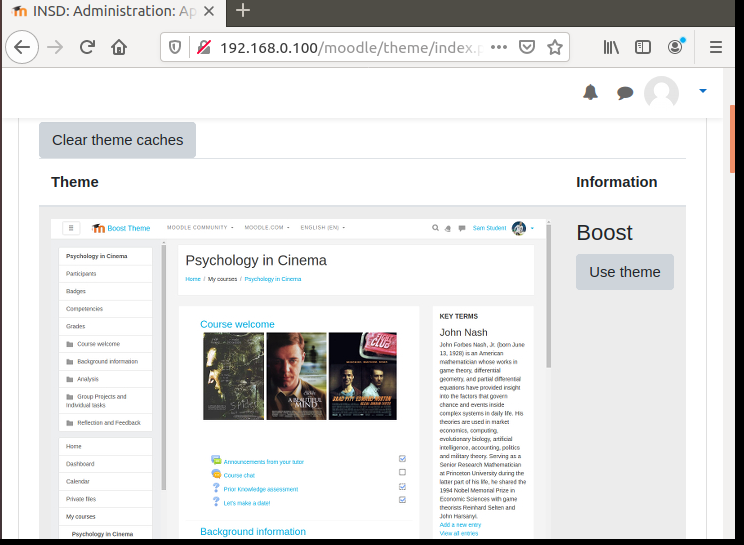


Your message will appear below the login box:

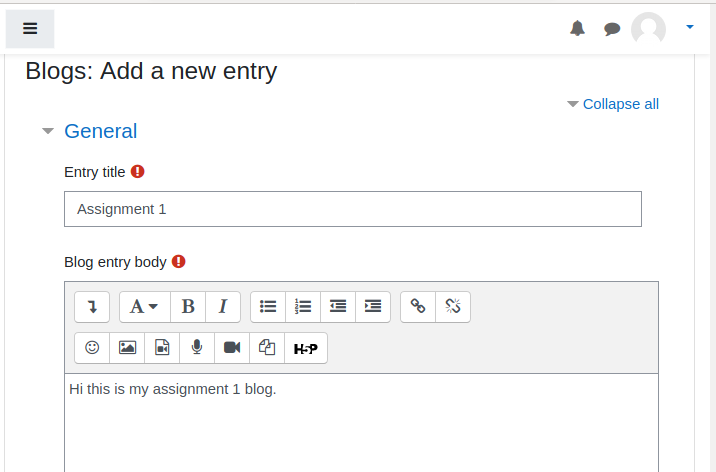


**Changing the theme**

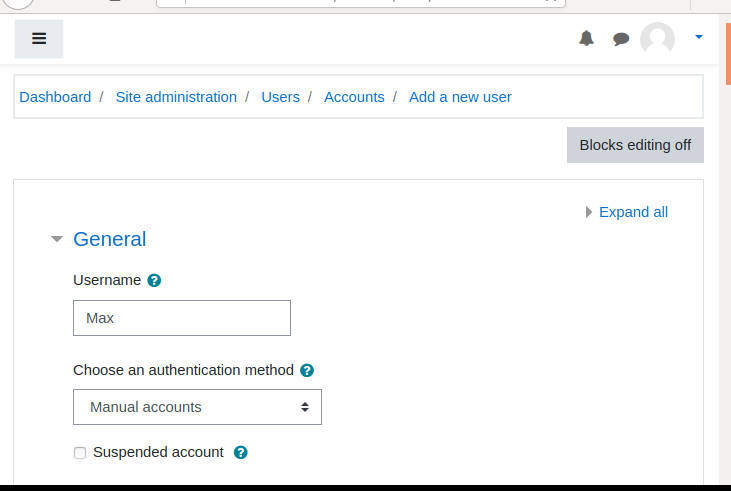
* Navigate to Administration > Site administration > Appearance > Themes > Theme Selector.
* Click Change theme button on the right
* Scroll down to find the theme of your liking and click on use theme.

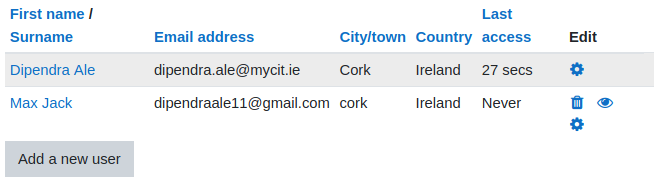


**Writing a blog**

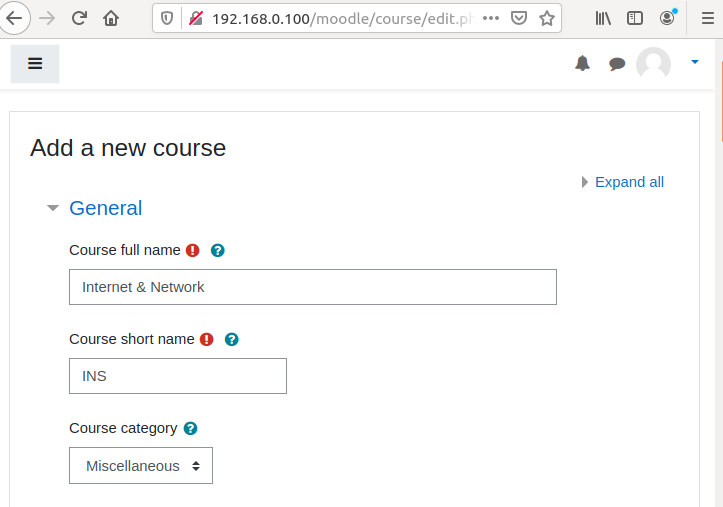


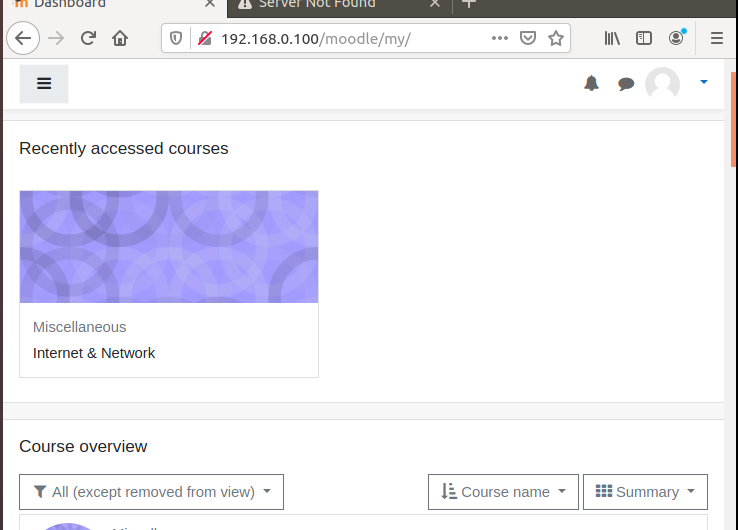
**Creating a new user**



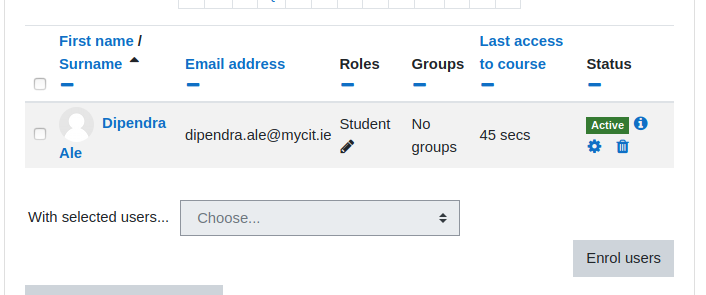


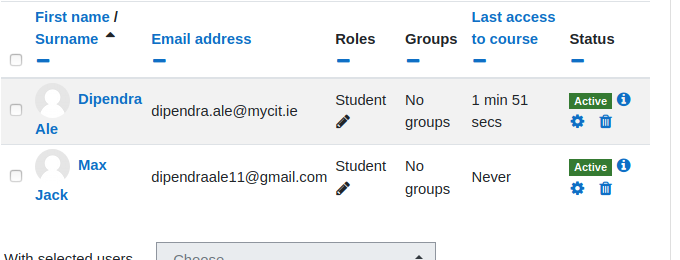
**Adding a new course**



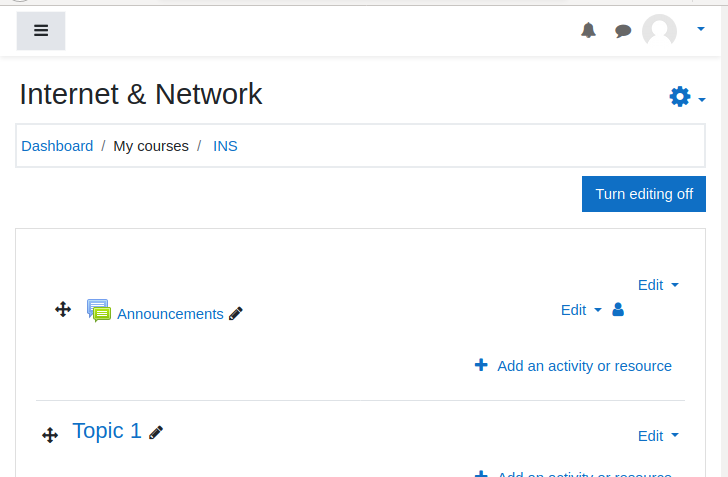


Enrolled a users





You can edit all the course contents to your linkings.



**Summary & Conclusion**

Overall, this project was very informative and helped me to gain insight knowledge about how systems administrators have to deal with building a secure servers as part of their job. This has definitely helped me to be prepared for next year 3rd year where we may have to do something similar in our placements.

I learned how to install the ‘perfect’ Ubuntu desktop & server and to be able to communicate with each other. The way that VMs Network Adapters settings affect how the server reacts was very informative, for example HOST is only set when communicating with client meanwhile NAT is used to access to internet and install the software’s. I also gained knowledge how to set IP settings in both server/client and was able to see the different settings for HOST and NAT config files. I also learned how to use editors especially VIM which I had never used before. The commands were bit different to nano which I used previously but I got hang of it after a while.

Security aspect of this project was the most interesting for me, I found out the many ways that you can secure the Linux server using Ubuntu. Most importantly, I realized how important security actually is to maintain the server and keep it lean and tidy. I learned how to set up SSH keys, configure & install UFW and configure sysctl securely. All these security ways to harden my server was new to me as I had not came across these before and this helped me to understand how and why they are used.

The application that I chose to host was Moodle. I had researched all of the options that were available for the assignment beforehand but Moodle definitely was the most interesting one for me. As Moodle is a learning platform similar to Canvas which is what we use, I wanted to see how other learning flatforms are like and get hands-on experience. The initial installing part of Moodle took me some time due to some versions compatibility problems which I will list on the Project Progress Documentation part of the assignment. But after I successfully managed to install Moodle, it was definitely worth the time as there were a lot of fun customization features like themes, colors etc.

If I were to start this project again I would definitely research more into all of applications and also the installation guides because some guides may not be actually applicable to you due to the compatibility of different versions. I also would give more time researching/learning many other ways you can secure your server. I would also make sure to double check when editing/making changes to the system configuration files as it can be very head wrecking if u mess up these files. Lastly I would definitely make at least 2 or 3 snapshots so that even when u mess up u can always go back/revert to the snapshot.

**Project Progress Documentation**

20th Feb – I followed and re did the lab 1 which was creating server & client VM’s. I managed to get the two VM’s pinging to each other which is very important aspect of the Assignment. I had no issues doing this part as we had already done it before in the lab. I then took all the screenshots of the progress for the assignment in a word file.

23rd Feb – I went on to research the security aspect of server and how we can harden and secure it. There were a lot of guide on the web but the two that I found most useful and simple to understand were these website.

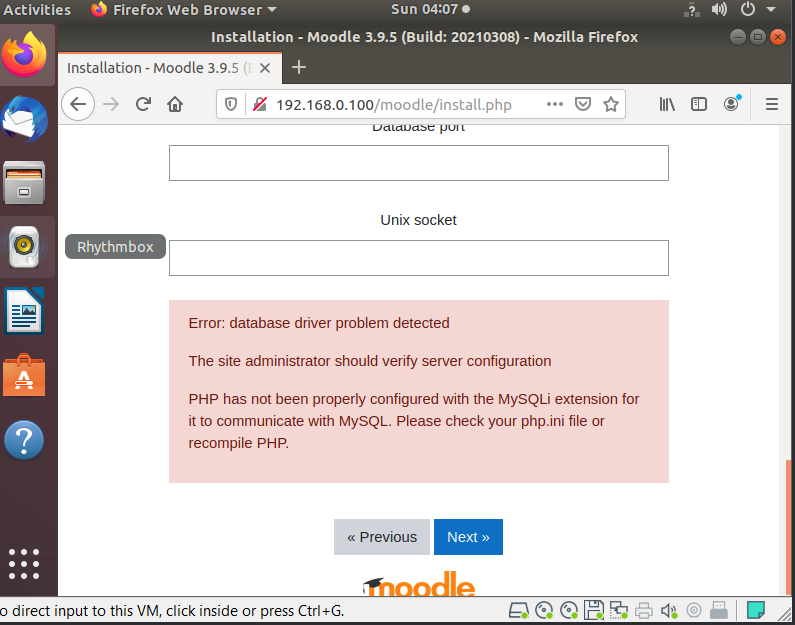
Links : <https://www.process.st/server-security/>

<https://www.liquidweb.com/kb/security-for-your-linux-server/>

25th Feb - I went on and used these guides to test them on my server and kept a record/screenshots of the changes that I made. While I was editing a system config file for setting up SHH keys I messed up because of some typo error and I had to revert to my saved snapshot and had to redo it.

3rd Mar – I went on to research the installation of Moodle guides on the web and found a lot of them. I decided use this website : <https://cloudcone.com/docs/article/how-to-install-moodle-on-ubuntu-18-04/> as a guide. I went on to face few issues as there were some steps missing and the application was not correctly installed as the apache wasn’t able to locate.

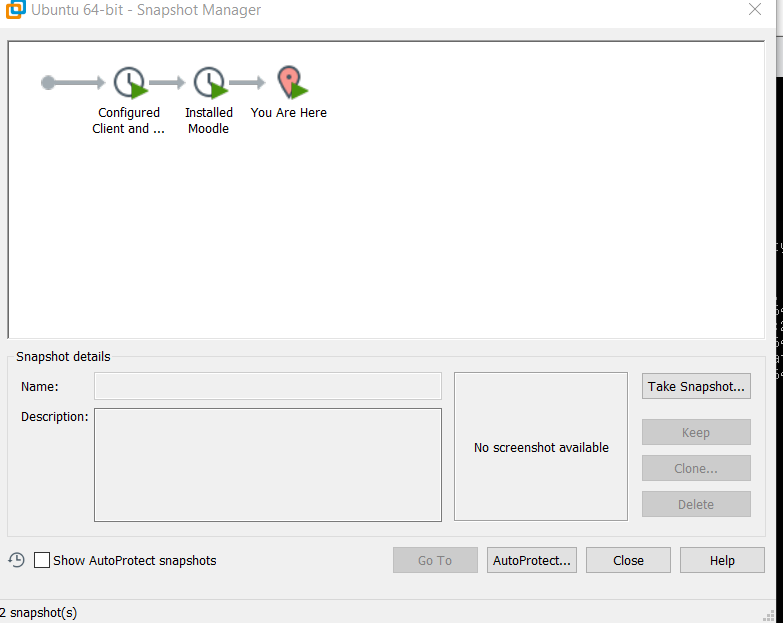
4th Mar – I decided revert to my earlier snapshot and use another guide which was provided by the moodle’s main website:<https://docs.moodle.org/310/en/Step-by-step_Installation_Guide_for_Ubuntu> . I also found some issues with this guide as php was not configured properly with MySQLI as shown in the snippet. I finally resolved the issue by downloading a previous version of php7.2 as the one that was on the guide 7.4 didn’t have MySQLi extension enabled.



6th Mar – I went on the moodle website and played around a bit to customize the app to my likings.

7th Mar – I then finished up the report by doing a Summary & conclusion.

**My Snapshots:**



**Reference :**

<https://www.process.st/server-security/>

<https://www.liquidweb.com/kb/security-for-your-linux-server/>

<https://cloudcone.com/docs/article/how-to-install-moodle-on-ubuntu-18-04/>

<https://docs.moodle.org/310/en/Step-by-step_Installation_Guide_for_Ubuntu>

<https://cit.instructure.com/courses/36916/files/1070631?module_item_id=323823>

**A**