

Internet & Network Services

Assignment 1

Adam Lloyd – R00117318

- Web Development -

YouTube Demo: <https://www.youtube.com/watch?v=i4OE7j1kbd4>



ubuntu



Table of Contents

Overview	3
Operating System Installation.....	4
The Super User.....	5
Network Configuration	6
Network Services	8
PhPMyAdmin	10
Joomla	11
Joomla – The Database	11
Joomla – The Installation	13
Joomla – The Setup.....	15
Security	17
Firewall.....	17
Hardening PHP	18
Nmap.....	19
Conclusion.....	20
References	21

Overview

(<http://www.ubuntu.com/>, 2015)

For this project a secure server was required to serve your web applications using the Joomla Content Management System (CMS).

The operating system selected for this project is Ubuntu server version 14.04.3 LTS. This is an “Open Source” Linux based operating system which means that your server will be extremely reliable and secure as well as being open source which means that people all over the world contribute to make it more secure every day.

The reason behind choosing this version of the newer 15.10 version is that this is the long Term Support (LTS) version of the operating system (O/S). LTS means that the Ubuntu project will provide support for this version until 2019 and, as a client, to you it means that the server O/S will not need to be upgraded anytime soon saving you time and money.

The server is now set up and is running smoothly so your only interaction with it should be creating your websites using the Joomla interface from your web browser on your computer. However, I have included the following documentation for your reference should you need to modify anything in the future.

Operating System Installation

(<http://www.ubuntu.com/>, 2015)

The Ubuntu O/S can be obtained by visiting <http://www.ubuntu.com/download/server> and choosing to download the LTS version. Clicking the Download button will start the .iso file to your download folder. This is a disk image file and you will need to insert a blank DVD disk into your computer and use either your native O/S functionality or download some software to copy the disk image onto your DVD.

If your O/S supports it, you can browse to your download directory and right click on the .iso file, you may get an option called “Burn Disk Image” or something similar. If so, follow the on screen instructions until the disk is prepared and ejects from the computer.

If you do not have access to such an option you may download a program to do it for you such as IMG Burn from <http://www.imgburn.com/> and proceed that way.

Once your disk is prepared you may insert it into the computer you wish to install your Linux server on and power it on. Ubuntu Server should automatically load and begin prompting you for installation instructions.

Choose the “basic server install” option and, if prompted just accept the default values suggested by the installer.

When prompted for usernames and passwords you may choose any that you wish but be sure to keep a record of them as they will be essential in the later steps. Be sure to also take note of the Super User username and password as this is equally as important.

The computer should reboot once this is complete and once you log in with the username and password you selected, you will have a working Ubuntu server with some basic functionality ready to be configured.

The Super User

(Regan, 2015)

Once successfully logged in, you will be presented with a command line interface (CLI) and some minor setup is required to get the network settings correct and ready to use.

Since you logged in with your chosen username you are using the system as a user and part of Linux security is that users cannot make sensitive changes to the system, as such you need to do things as a Super User. This is achieved with the “sudo” command, which means “Super User Do”. After issuing this command you will be prompted for the super user password that you set up in the O/S installation.

If you want to do anything relating to the system settings on the server you need to preface any command you use with “sudo”, failure to do this may result in incorrect configuration of the server.

For example, if I wished to open a text file using the Nano text editor that comes with Ubuntu I might use the following command:

nano myTextFile

This is fine but if you wanted to open up a protected file such as the network interface configuration file detailed below, you would have to use the following command:

sudo nano /etc/network/interface

This effectively says to the system, “I would like to open this file as a Super User since my user account isn’t allowed”.

The file is called “interface” and the words and slashes preceding it are the directory location but don’t worry about that now.

Network Configuration

(Regan, 2015)

Out of the “box” the server comes configured to use Dynamic Host Configuration Protocol (DHCP) which just means that it looks at whatever network it is connected to and just auto configures the settings so it can connect to that network. The server will get an address from the network that other computers can use to communicate with it.

For this project we need to modify these settings to use a static (fixed) address on the network so we always know exactly what address it has which will not change.

To do this we issue a command like so:

sudo nano /etc/network/interfaces

This tells the system that I wish to open a file called “interfaces” which is location in the directory “/etc/network”

Modify the file to look like the following screenshot. If the file that opens is blank, it means there was either a typo in the command or the sudo prefix was not entered.

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.0.100
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
    dns-nameservers 8.8.8.8 8.8.4.4
```

The above settings tell the server to connect to the network with an address of 192.168.0.100 each time it connects. Other notable settings here are the gateway which is effectively the address of the router or device that connects the network to the internet. The dns-nameservers tell the server how to convert IP addresses into internet addresses like www.google.com.

To save the file press “CTRL + X” then press “Y” and you will be returned to the command prompt again. Bear in mind that you were only a “Super User” for that one command and now you are using the system as a regular user again.

Next we issue a few other commands to make sure that those settings are applied:

sudo ifdown eth0

sudo ifup eth0

This tells the server to apply the newly configured settings to the “eth0” interface which you configured in the steps above. If you are having any connectivity issues throughout this process then issue these commands as they may resolve the issue.

Next we need to set up the servers “hosts” file, we open this will the following command:

sudo nano /etc/hosts

Change the file that opens to look like the following screenshot.

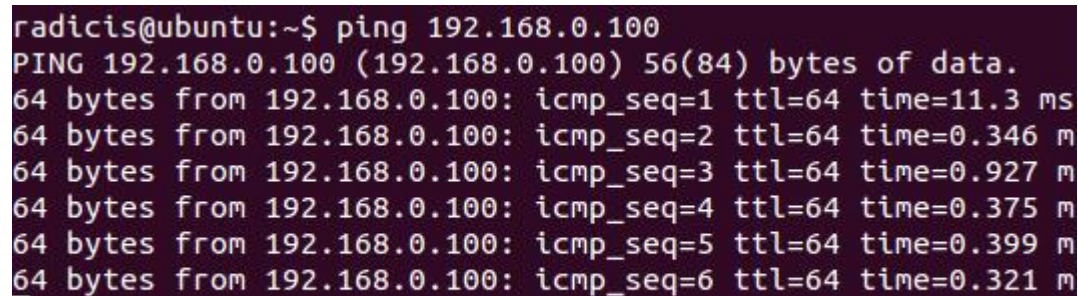
```
127.0.0.1      localhost.localdomain  localhost
192.168.0.100  server1.example.com    server1

# The following lines are desirable for IPv6 capable
::1           ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
```

At this point you should be able to test that your settings are correct by “pinging” the server from another computer on the same network. You can do this by opening a command prompt and typing the following:

ping 192.168.0.100

If this has worked you will see something like the following screenshot:



```
radicis@ubuntu:~$ ping 192.168.0.100
PING 192.168.0.100 (192.168.0.100) 56(84) bytes of data.
64 bytes from 192.168.0.100: icmp_seq=1 ttl=64 time=11.3 ms
64 bytes from 192.168.0.100: icmp_seq=2 ttl=64 time=0.346 m
64 bytes from 192.168.0.100: icmp_seq=3 ttl=64 time=0.927 m
64 bytes from 192.168.0.100: icmp_seq=4 ttl=64 time=0.375 m
64 bytes from 192.168.0.100: icmp_seq=5 ttl=64 time=0.399 m
64 bytes from 192.168.0.100: icmp_seq=6 ttl=64 time=0.321 m
```

If you see a message saying “Destination Host Unreachable” it means there has been an error in one of the above steps or the computer you are using to run the “ping” command is not correctly connected to the same network as the server.

Network Services

(Regan, 2015)

Now our server is connected and ready to use but, so far, it does not have any capabilities to host websites or databases which are required for our Joomla installation.

To enable this functionality we need to install a Linux, Apache, MySQL, and PHP (LAMP) service bundle. This is a bundle of programs that are commonly used on a web server. Linux refers to the operating system we are using, Apache is a very popular and reliable web server which allows people on the network to access files on the server, MySQL is a database that allows us to store data and PHP is a programming language which is used to connect these together. Joomla will make use of all of these to provide a simple and effective application for you to use.

Before LAMP is installed we need to make sure that the O/S knows where all of the files are on the internet and that it is looking for the latest versions. We do this by using the following command:

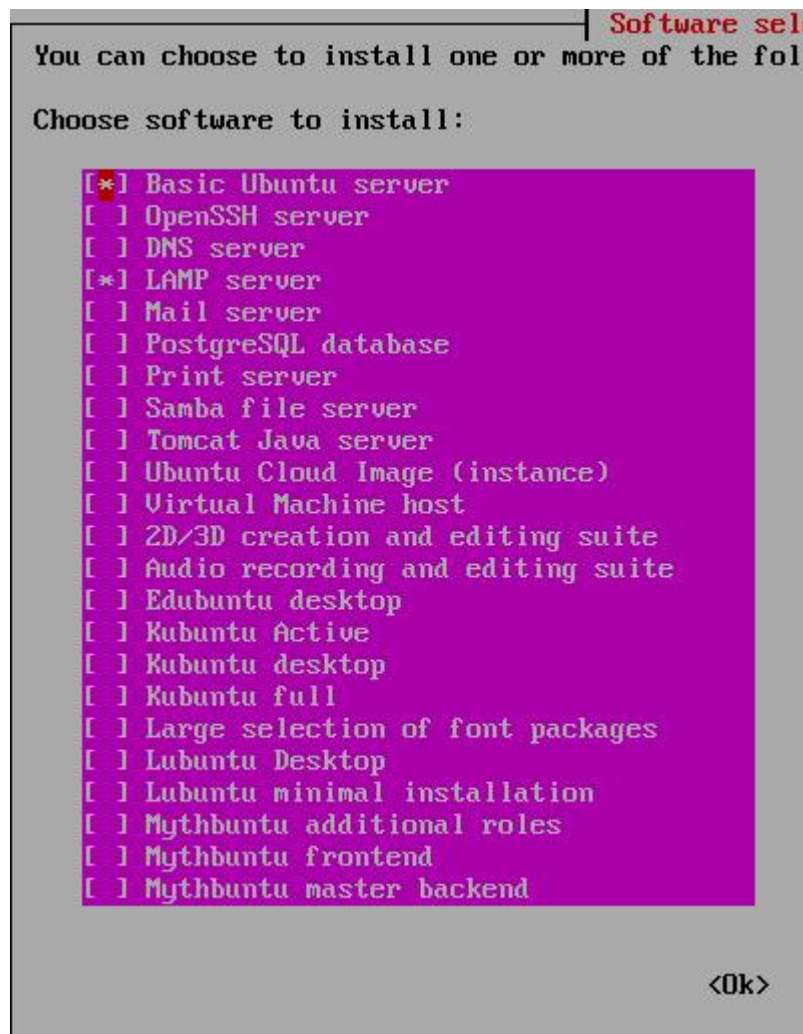
sudo apt-get update

“apt” is what Ubuntu uses to find files on the internet and install programs. This command tells it to update itself and make sure it has all the latest information before we install the rest of the things we need.

Next type the command:

sudo tasksel

This will present you with the following screen:



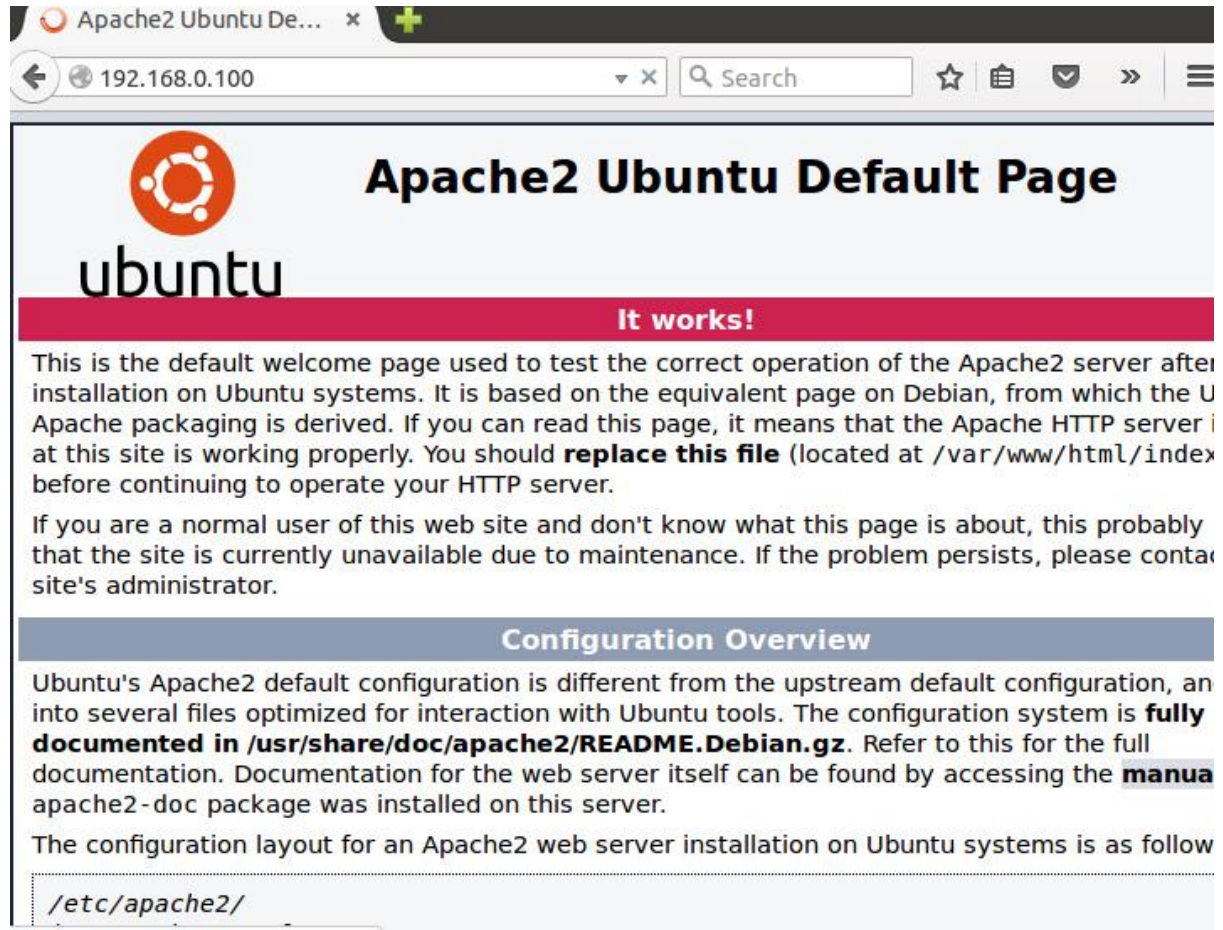
Navigate down to “LAMP Server” using the arrow keys and press “Space” to select it. Then press “Enter” to begin installation.

During this installation you will be prompted to choose a root password for MySQL. Be sure to make a note of this as it will be required later.

Issue the following command to restart the Apache server:

```
sudo /etc/init.d/apache2 restart
```

Once this command completes your apache web server will be accessible and you can access it from any computer on the network by typing “192.168.0.100” into a web browser. You will be greeted by the following screen:



Your web server is live and ready to install Joomla.

PhPMyAdmin

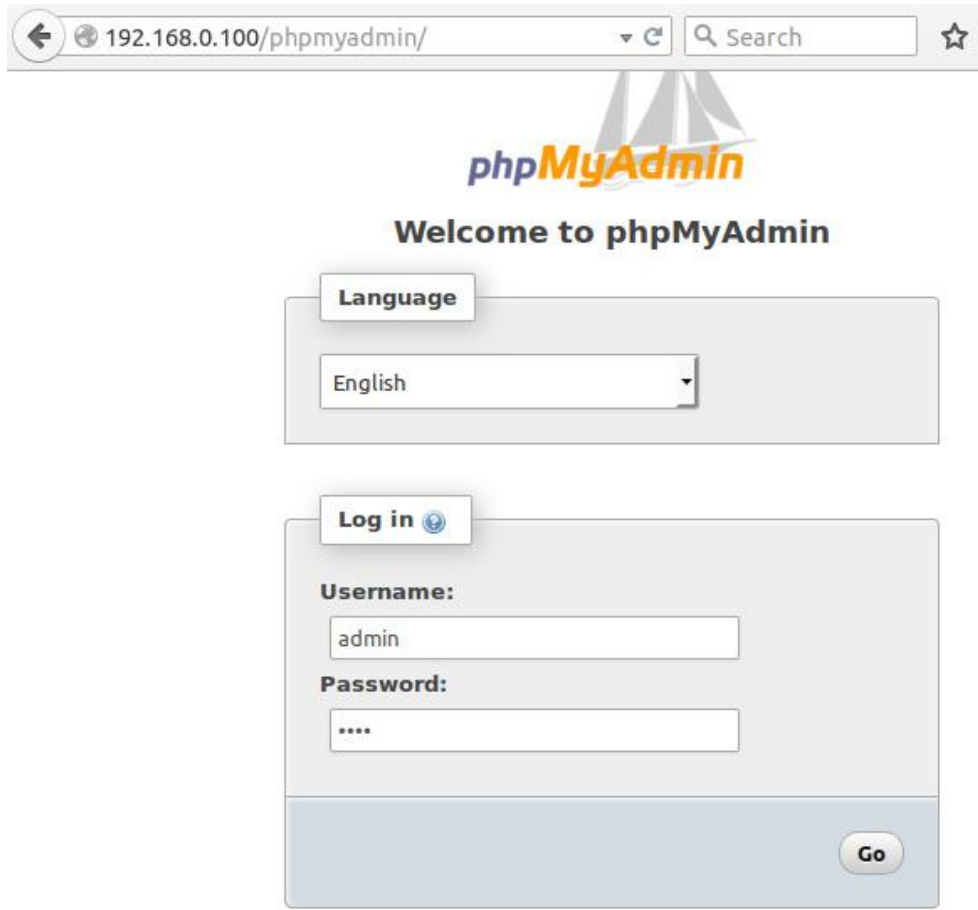
(Regan, 2015)

Since we install PHP and MySQL earlier and Joomla relies on these to function we must next install PhPMyAdmin which is a web based interface which lets you look at your databases and php configuration information from your browser.

To install use the command:

```
sudo apt-get install phpmyadmin
```

Once complete you can open a browser on another computer again and go to <http://192.168.0.100/phpmyadmin> and you will be greeted by the following screen:



The screenshot shows a web browser window with the address bar displaying "192.168.0.100/phpmyadmin/". The page features the phpMyAdmin logo, which includes a stylized sailboat. Below the logo, the text "Welcome to phpMyAdmin" is displayed. There are two main sections: a "Language" section with a dropdown menu currently set to "English", and a "Log in" section. The "Log in" section contains fields for "Username:" (with "admin" entered) and "Password:" (with four dots for masking). A "Go" button is located at the bottom right of the login section.

You can visit this page in the future to modify or view any of your created databases once we have created them.

Joomla

Joomla – The Database

(<https://www.howtoforge.com>, 2015)

To set up Joomla we first need to create a database for it to use by using the MySQL service we installed earlier.

We log into the MySQL service using the following command:

mysql -u root -p

Note that sudo is not needed here as we aren't accessing anything sensitive and mysql has its own login details that you need to enter before you can make any changes.

This command tells the mysql service that you wish to login to it as the user “root” and to prompt you for a password. Enter the command then the root password you configured earlier. You will be presented with the following screen:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 52
Server version: 5.5.44-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

From here on we are not using the Ubuntu command line anymore, we are using MySQL so the commands are different.

First we create the database for Joomla to use:

CREATE DATABASE joomladb;

This tells MySQL to create a blank database called “joomladb”

Next we need to create a user account which we will use to access/modify this database:

CREATE USER joomlauser@localhost;

..and then set the password for that user:

SET PASSWORD FOR joomlauser@localhost= PASSWORD("joomlapassword");

You may set the password here to anything you like but be sure to take note of it.

Next we need to tell MySQL that the user we created “joomlauser” is allowed to modify the database:

GRANT ALL PRIVILEGES ON joomladb.* TO joomlauser@localhost IDENTIFIED BY 'joomlapassword';

This tells MySQL that joomlauser is allowed to do everything it wants on the joomladb database.

The privileges we just set may not yet be active, to ensure that everything is updated we issue the following command:

FLUSH PRIVILEGES;

And then we exit MySQL by typing:

exit

It is also possible that the Apache web server and the MySQL service will need a restart after this so be sure to issue the following commands after making any changes like this

service apache2 restart

and then:

service mysql restart

Note: These 2 commands can be used if you have any issues with connectivity to your server or your database at any stage as a simple restart may resolve them rather than editing any configuration files.

Joomla – The Installation

(<https://www.howtoforge.com>, 2015)

Now the database is ready to be used we need to install the core Joomla files on the server.

We can't just download files into any folder we like on Ubuntu as this is prevented by inbuilt security so it is necessary create a temporary folder to store the installation files in first.

In your command line once more create a new directory called "temp" by issuing the following command:

mkdir temp

This tells the O/S to make (mk) and directory (dir) for us called "temp".

Note: that we do not need sudo here as we are just making a folder in a non-secure location.

Then we navigate into that folder like so:

cd temp

This Change(c)s the Directory(d) to "temp" just like clicking on the folder in a windows based system.

So now we have that directory open we need to download the Joomla files and save them in it. We do this by issuing the following command:

wget http://joomlancode.org/gf/download/frsrelease/19665/160049/Joomla_3.4.5-Stable-Full_Package.zip

Note: The latest version of Joomla can be found here: <https://www.joomla.org/download.html> just replace the 3.4.5 with the most recent version listed on this page.

Now we have the installation files but they are "zipped" up in a .zip file. To "unzip" them we need to install some other software to do this using the following command:

apt-get install unzip

Next we need to make a directory for the joomla installation to reside and this needs to be inside the **/var/www/html** directory on the server as this is where the apache web server looks when it serves files to people accessing it with a web browser.

mkdir -p /var/www/html/joomla

Again with the mkdir command we are creating a directory called joomla

Then, using the newly install unzip application we "unzip" the installation files into this directory:

unzip -q Joomla_3.4.5-Stable-Full_Package.zip -d /var/www/html/joomla

This command will "unzip" the zip file we downloaded into the directory we just created.

Next we have to tell the server to change the permissions of this folder so it can be accessed

chown -R www-data.www-data /var/www/html/joomla

This tells the server that the web server "owns" these files so it can access them and use them as it wishes.

chmod -R 755 /var/www/html/joomla

This allows everyone that can access this location to read and execute the files in it but only the owner of the files can modify them. Simple it just means that the files can be accessed on your web server by using your joomla installation.

After all of this is complete you may wish to run the commands to restart your apache web server and MySQL again:

service apache2 restart

service mysql restart

Joomla – The Setup

(<https://www.joomla.org/>, 2015)

The server will now have Joomla installed and you will be able to access the interface for it from a web browser by going to the following address:

<http://192.168.0.100/joomla>

You will be presented with a screen like the following:

Joomla! is free software released under the GNU General Public License.

1 Configuration 2 Database 3 FTP 4 Overview

Select Language English (United States) Next

Main Configuration

Next

<p>Site Name * <input type="text" value="joomla_test_site"/></p> <p>Enter the name of your Joomla! site.</p> <p>Description <input type="text" value="joomla_test_site"/></p> <p>Enter a description of the overall Web site that is to be used by search engines. Generally, a maximum of 20 words is optimal.</p>	<p>Admin Email * <input type="text" value="admin@example.com"/></p> <p>Enter an email address. This will be the email address of the Web site Super Administrator.</p> <p>Admin Username * <input type="text" value="admin"/></p> <p>Set the username for your Super Administrator account.</p> <p>Admin Password * <input type="password" value="....."/></p> <p>Set the password for your Super Administrator account and confirm it in the field below.</p> <p>Confirm Admin Password * <input type="password" value="....."/></p>
---	---

From here you can set your “Site Name” as you wish and enter a short description of the site.

The admin details on the right should be set and recorded for use later.

Clicking “Next” will present you with the Database Configuration screen:

Database Configuration

[< Previous](#)
[Next >](#)

Database Type *
This is probably "MySQL"

Host Name *
This is usually "localhost"

Username *
Either something as "root" or a username given by the host

Password
For site security using a password for the database account is mandatory

Database Name *
Some hosts allow only a certain DB name per site. Use table prefix in this case for distinct Joomla! sites.

Table Prefix *
Choose a table prefix or use the **randomly generated**. Ideally, three or four characters long, contain only alphanumeric characters, and **MUST** end in an underscore. **Make sure that the prefix chosen is not used by other tables.**

Old Database Process *
Any existing backup tables from former Joomla! installations will be replaced

Complete the details like so:

Database Type = MySQLi *MySQL Improved, the default for use with a MySQL database*
hostname = localhost *Tells Joomla that the database is on the same machine as Joomla*
username = joomlauser *As Set earlier*
password = joomlapassword *As Set earlier*
Database Name = joomladb *As you created it earlier*
Table Prefix = jml_ *To indicate joomla tables from others in your database*

The next few screens are there to configure FTP access, set up a website template if desired and to remove the installation files from the server. Simple accept the default values for all of these and choose to remove the installation files at the end.

You can now view your site at <http://192.168.0.100/joomla> and enter the administration section at <http://192.168.0.100/joomla/administrator/> using the administrator details you created earlier in this section.

From here you can create pages and content and design your web site as you wish.

Security

In order to add some increased security to the Ubuntu server and the Joomla install several steps can be taken.

Firewall

(www.thefanclub.co.za, 2015)

UFW or Uncomplicated Firewall is a basic firewall that will prevent many malicious attacks on the server by default without complicated configuration and is extremely simple to install.

Just install the application with the following command:

```
sudo apt-get install ufw
```

Once complete adjust a few settings to allow for http and ssh(should this be required) using the following 2 commands:

```
sudo ufw allow ssh  
sudo ufw allow http
```

You can then activate the firewall with the command:

```
sudo ufw enable
```

..and check the status of it using:

```
sudo ufw status verbose
```

```
radicis@ubuntu-server:~$ sudo ufw status verbose
[sudo] password for radicis:
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
22 ALLOW IN Anywhere
80 ALLOW IN Anywhere
22 (v6) ALLOW IN Anywhere (v6)
80 (v6) ALLOW IN Anywhere (v6)
```

As you can see in the above screenshot, after using the ufw status verbose command you are shown a list of the open ports on the server. Port 80 is for basic web requests and anyone is “allowed in” so they can access your website and port 22 is open to allow for SSH or Secure Shell activity which lets a user securely log into the server from a remote location.

Hardening PHP

(www.thefanclub.co.za, 2015)

Since PHP is a programming language, it is possible that malicious users could use this language to exploit vulnerabilities in your server. This risk of this can be reduced by removing their ability to see sensitive information such as errors and even to prevent them from knowing your server uses PHP at all so they will not even know what language to try attacking with.

We open the php initialization file using the following command:

```
sudo nano /etc/php5/apache2/php.ini
```

Once this file opens modify the following lines to match these:

```
disable_functions = exec,system,shell_exec,passthru Removes ability to execute sensitive commands  
register_globals = Off  
expose_php = Off Hides the face that the server has PHP installed  
display_errors = Off Stops errors displaying which may expose sensitive code  
track_errors = Off As above  
html_errors = Off As above  
magic_quotes_gpc = Off There may be a vulnerability in this php module that may enable SQL injection
```

Save the file again by pressing "CTRL+X" and then "Y" and restarting the apache web server:

```
sudo /etc/init.c/apache2 restart
```

Nmap

(www.thefanclub.co.za, 2015)

While the UFW firewall status command was able to show open ports, a more advanced application called NMAP may be used to scan your server to ensure there are no other ports open. This is a popular application used to exploit vulnerabilities in systems so using it to check your own server may be beneficial.

Issue the following command to start the scan:

nmap -v -sT localhost

You will be presented with something that looks like the following screen:

```
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
587/tcp    open  submission
3306/tcp   open  mysql
```

As you can see, much like the UFW status report, port 80 is open but also ports 25, 587 and 3306.

Port 3306 is open which may seem troubling but as you can see in the "SERVICE" column, it is only open to the MySQL service which is on the server itself so there is no threat of outside entities accessing it.

Port 25 allows for the server to send mail, so if you have lost your Joomla password it can be used to retrieve it or send emails from your website. This is fine but it is a well-known port which is widely used to distribute malware (viruses) so, as you can see, we also have port 587 open which can be used instead if the application supports it. This port can be set in the Joomla settings.

As you can see below, it is possible to change the mail settings from PHP mail which may be unreliable to SMTP and change the port to 587 for added security and reliability.

Mail Settings

Mailer *	<input type="text" value="SMTP"/>
From email	<input type="text" value="adam.lloyd@mycit.ie"/>
From Name	<input type="text" value="INS"/>
SMTP Authentication	<input type="button" value="Yes"/> <input checked="" type="button" value="No"/>
SMTP Security	<input type="text" value="None"/>
SMTP Port *	<input type="text" value="25"/>
SMTP Username	<input type="text"/>
SMTP Password	<input type="password"/>
SMTP Host	<input type="text" value="localhost"/>

Conclusion

Choosing a Linux based operating system to base the project on ensures that the website will remain as stable as possible and, with the added security tweaks, be as secure as possible. The LTS(Long Term Support) version of the operating system selected will ensure that the server gets the required stability and security updates as they become available in the future.

The Joomla installation is set up with its own security and database user rather than just using the root database user to improve the security there and Joomla is very well supported itself and updated regularly for security and stability.

The combination of these 2 will make for a very robust web application solution.

References

<http://www.ubuntu.com/>. (2015, 10 23). *Unubtu*. Retrieved from Ununtu: <http://www.ubuntu.com/>

<https://www.howtoforge.com/>. (2015, 10 23). *How To Forge*. Retrieved from How To Forge:
<https://www.howtoforge.com/how-to-install-joomla-on-ubuntu-14.04>

<https://www.joomla.org/>. (2015, 10 22). Retrieved from Joomla: <https://www.joomla.org/>

www.thefanclub.co.za. (2015, 10 22). Retrieved from The Fan Club:
<https://www.thefanclub.co.za/how-to/how-secure-ubuntu-1204-lts-server-part-1-basics>