SETTING UP A LAMP SERVER REMOTELY

It's been said a million times over — Linux is awesome on servers! With over 60 per cent of the Web's servers gunning away on the mighty penguin, the robust, resilient, scalable and stable Linux plays a major role in keeping the Internet running like a well-oiled machine. In this article, I will describe how to set up a LAMP (Linux, Apache, MySQL and PHP) server from scratch, remotely. The only step requiring physical access is installing Ubuntu Server. The rest can be done the geeky way, via SSH! Read on if getting your hands dirty gives you a kick!

Let us first install Ubuntu Server, and then, from another system, connect using the PuTTY SSH client to install FTP support and phpMyAdmin, and get a phpBB website up and running.

Setting up Ubuntu Server

Download the ISO from the [Ubuntu website](). Burn a CD, or prepare a USB boot disk, as per your preference. Boot the server machine from it. The boot menu is shown in Figure 1. Select Install to the Hard Disk. The installer will boot and ask you to select your language. Select English and continue.
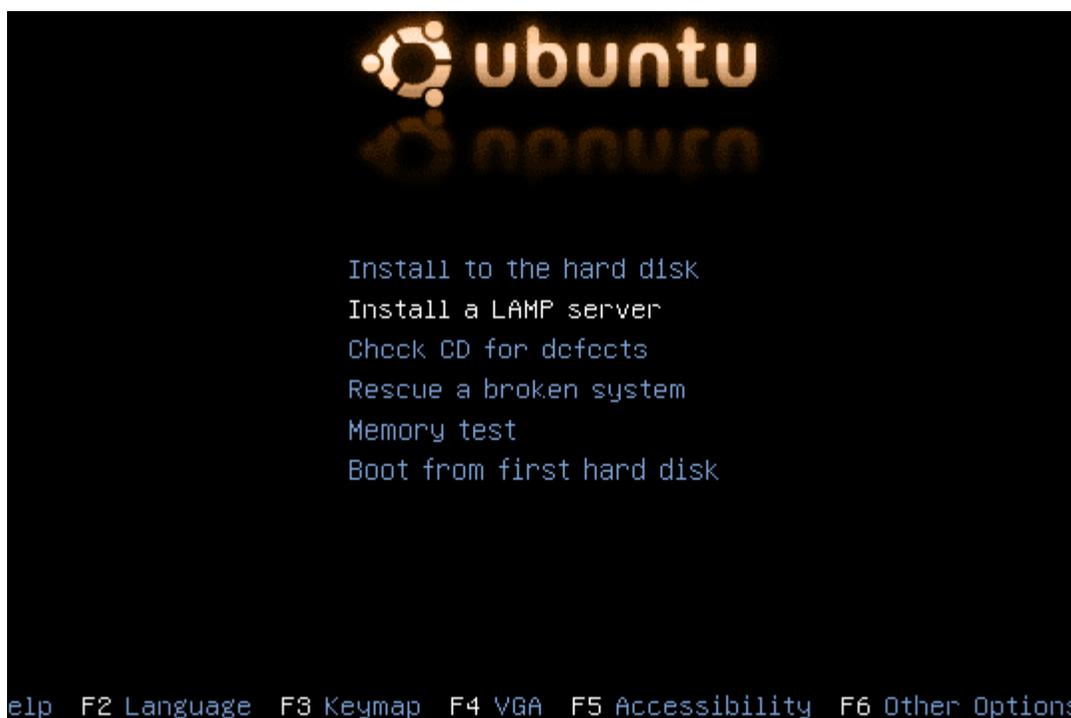
Figure 1: Ubuntu Server boot menu

I will fly through the next few screens, since they're all what you usually see during any standard OS installation. For the keyboard layout, the default US layout works for us. The installer will then load components and configure hardware. It will also try to configure the network interface using DHCP. If that fails, you will be asked to manually enter network information like Host IP, Gateway, Subnet Mask, etc. It will also prompt you to enter the hostname; use a name you would like the server to be

known by on the network. For this article, I have used 'mylamp' with IP address 210.22.23.24.

Next comes partitioning. I used the Erase Entire Disk option; if you have any specific partitioning requirements, feel free to use Manual partitioning. After this stage, under Clock Configuration, you will be asked about your system time setting. The new user (non-root) creation screen is next; enter your full name, user name (I used 'lampuser') and password.

The installer will now show you a software selection screen; select LAMP Server and OpenSSH Server.
Wait as the progress bars fill up — first for the base system, then for the package installation, and then for the GRUB bootloader. Once all these are done, it will tell you that it is time to reboot and take your new server for a ride!


```
       ┌─────┤ [!!] Finish the installation ├─────┐
                  Installation complete
   Installation is complete, so it is time to boot into your new system.
   Make sure to remove the installation media (CD-ROM, floppies), so
   that you boot into the new system rather than restarting the
   installation.

      <Go Back>                                    <Continue>
```
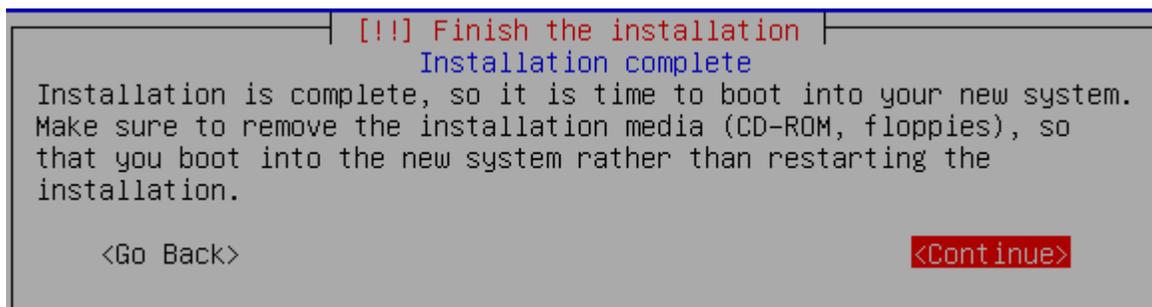
Figure 2: Reboot prompt

Server installation is now complete. Remove the disc and return to your desktop. Let us now remotely configure the server for a LAMP website.

Preparing for deployment

Whether your desktop runs Windows or Linux, you can use the PuTTY SSH client. [Download it](#), fire it up, and enter the connection details for your server (Figure 3).
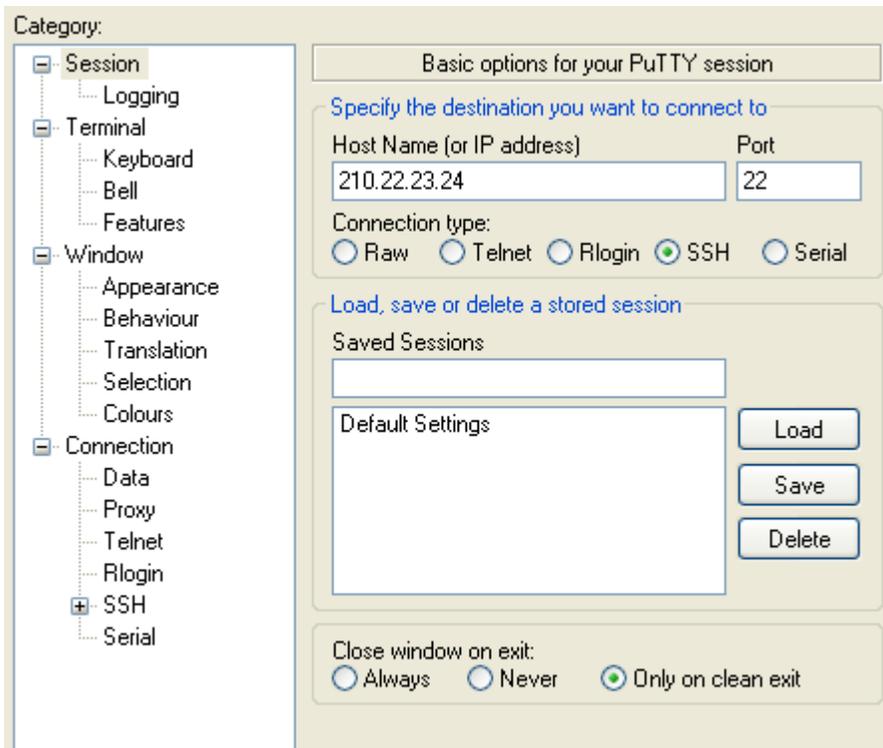
Figure 3: PuTTY connection screen

Click Open to connect to the server. This will bring up a terminal prompt that asks you to enter the username. It may throw up an error similar to the one in Figure 4. Just click on Yes. Once the password is accepted, you will be greeted with a shell prompt. The first thing to do is set up a FTP server so we can upload files to the server. Next, let us install phpMyAdmin to manipulate MySQL databases.

Figure 4: Server host key alert

FTP service

FTP support is by the `vsftp` daemon; installation is straightforward: `sudo apt-get install vsftpd`. After installation, tweak the configuration file to point the default path to `/var/www` (the Apache 'Webroot' for hosted content/sites).
Edit `/etc/vsftpd.conf` as the root user, with a text editor and Make sure the settings match those shown below:

```
write_enable=YES
local_enable=YES
local_root=/var/www
file_open_mode=0777
```

Restart the FTP service to bring the configuration changes into effect: `sudo restart vsftpd`.

Installing phpMyAdmin

phpMyAdmin is a great tool for administering MySQL databases. Install it using the following command: `sudo apt-get install phpmyadmin`.
We can now exit the PuTTY session, using the exit command.

Deploying a LAMP website

For the purpose of this article, let us deploy a standard version of phpBB — the most famous AMP-powered bulletin board system and forum content management system (CMS). I will not go into the installation details, since the aim here is to learn how to generally deploy a LAMP website remotely — however, you can refer to this article for installation instructions. Download the archive from the phpBB home page and extract its contents.
Now, using an FTP client, connect to the server with the username and the password used in the PuTTY session. Create a folder called phpBB on the server,

and copy the extracted files and folders to the root of that directory. Open a browser and navigate to http://210.22.23.24/phpmyadmin (replace the IP address with yours) to bring up the phpMyAdmin page (Figure 5).
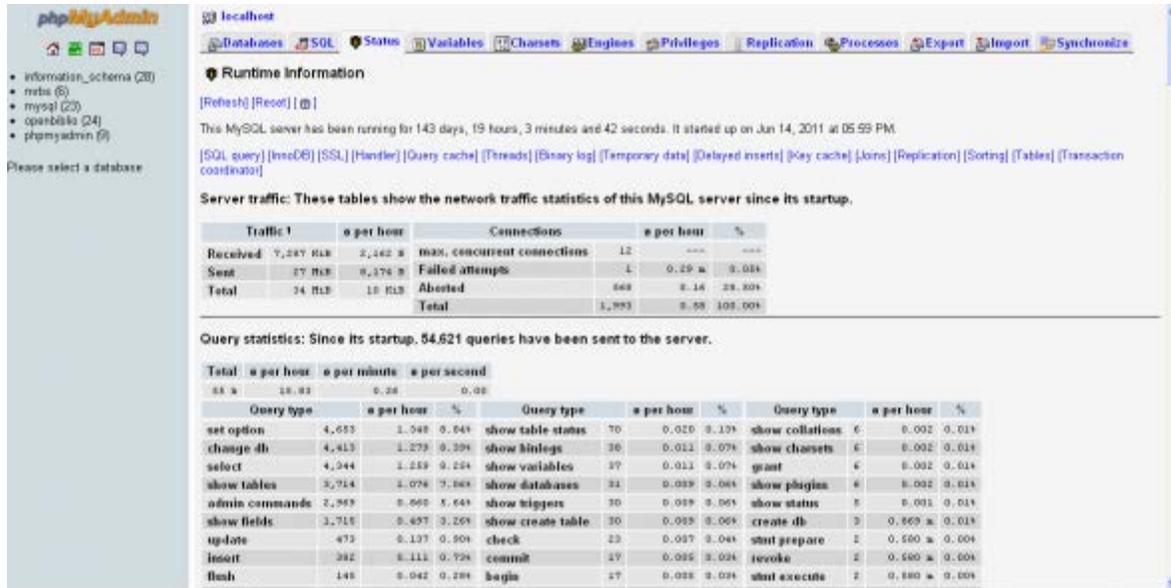
Figure 5: phpMyAdmin status page

Create a new database named phpbb — go to the Database tab and use the Create New Database section at the bottom. You can now proceed to a site-specific installation by navigating to http://210.22.23.24/phpbb. Follow the instructions on the page.

We have now successfully set up our server, and learnt how to remotely control it! I hope you had fun learning, during this process. Until next time, cheers!

Source : http://www.opensourceforu.com/2011/12/setting-up-a-lamp-server-remotely/