

Ubuntu 8.04 LAMP/Tomcat/FTP Server

Andrew Stein

Last Revision: 08/01/08

Goal

The goal of this project is to install a LAMP (Linux Apache MySQL PHP) server with FTP functionality on a machine installed with the Ubuntu 8.04 Linux Desktop. The guide will be very extensive and hopes to cover every step of the process both graphically and textually from the installing of the operating system to configuring the server components.

Audience

Anyone can use this guide to set up a server, but every step is specifically tailored for the target audience of those completely inexperienced with Ubuntu/Linux. The guide is written in such a detail that some of the steps will seem painfully obvious to even the somewhat experienced Windows user. More experienced users will want to skim over large portions of the guide to find the core commands to type in.

Approach

As mentioned above, this guide goes into very explicit detail and also tends to focus on actually learning the theory behind certain commands and concepts, rather than just blindly running them. For this reason, the guide is as much a Linux learning tool as it is a comprehensive way to install a server with Ubuntu. On a related note, I personally believe the Desktop version of Ubuntu is the best tool for easily configuring and maintaining a non-critical server machine and will make the guide more user friendly for the inexperienced. If you want to modify the guide for the Server edition of Ubuntu (a future goal of mine), I'd appreciate that initiative.

Fair Warning

Follow each command and step exactly as described with great attention to detail. If you mistype one thing carelessly, parts of your server will just not work. This point really cannot be stressed enough - in real world testing of this guide, small oversights ruined the process.

Software Used

Ubuntu Linux (8.04, Desktop Edition)

Ubuntu is a fairly new Linux distribution that has rapidly risen to prominence in the past few years due to its usability (see section I for more details)

Apache Web Server (Version 2)

Apache is the next generation release of the most popular open source web server in existence today.

Tomcat (version 5.5)

Apache Tomcat is the reference implementation for the Java Servlet and Java Server Pages technologies, permitting you to serve powerful JSP scripts from your machine

PHP v5

PHP is an open source, interpretive, HTML centric, server side scripting language.

MySQL and phpMyAdmin

phpMyAdmin is a web based front end for MySQL, an open source relational database.

ProFTPD and GPROFTPD

ProFTPD is a easy to use FTP server and GPROFTPD is a graphical tool to manage it.

Contents of Guide

- (1) [What is Ubuntu/Linux, and how do I acquire it?](#)
- (2) [Installing Ubuntu](#)
- (3) [Installing the server components](#)
- (4) [Configuring and Repairing the PHP, MySQL, Apache, Tomcat, and FTP Server Components](#)
- (5) [Setting up the work environment](#)
- (6) [Future Maintenance](#)
- (7) [Command line reference](#)
- (8) [Links Referenced](#)
- (9) [Contact Me](#)

(1) What is Ubuntu/Linux, and how do I acquire it?

Linux is the optimal operating system for a server as it is a fairly flexible platform and the software required to run a server is simple to install. It should also be reiterated that Linux is free, and for many schools this is a significant factor in choosing their software.

There are a lot of different distributions of Linux out there, and the choices can be overwhelming. A Linux distribution is simply an easily installable operating system that uses the Linux kernel at it's core, and generally comes packaged with hundreds of open source applications. More or less, a Linux distribution is the author's interpretation of what an operating system is supposed to be.

For the purposes of setting up a server, I strongly recommend Ubuntu 8.04 (Hardy Heron). Ubuntu is a fairly new Linux distribution that has rapidly risen to prominence in the past few years due to it's focus on usability - in fact, it claims to be "Linux for human beings". Ubuntu is the way to go as it is extremely easy to install, maintain, and update. There are basically two ways to acquire Ubuntu -you can either (a) burn the operating system to a CD or (b) order CD(s) from Canonical for free using their [ShipIt](#)¹ service.

(1a) Getting Ubuntu - Option 1 - Ordering Ubuntu CD's from Canonical

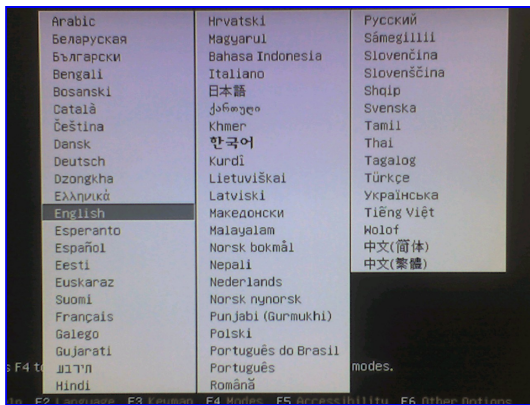
1. Using the [ShipIt](#)¹ service to order CD's with Ubuntu is fairly self explanatory (just visit the website), but the CD's may take up to 10 weeks to arrive in the mail (although in practice it turns out to be more like 2 weeks). If you do decide to order CD's from Canonical, when the CD's arrive, put the CD in the computer and skip to step 1 of the "Installing Ubuntu" section. My recommendation is to burn Ubuntu to a CD yourself, a process I explain in depth below.

(1b) Getting Ubuntu - Option 2 - Burning Ubuntu to a CD (recommended)

1. Download and install [ISO Recorder](#)² on a Windows machine. This will be used later to burn the OS onto a CD.
2. Visit the [Ubuntu download page](#)³.
3. Under the “Which release do you want” section, look under the “Desktop Edition” heading and make sure the “Ubuntu 8.04 -Supported to 2011” checkbox is selected.
4. Under the “What type of computer do you have” heading, select the type of architecture that best describes your computer. Certain server machines may be 64-bit AMD/Intel based, but determining whether this is the case is outside the scope of this guide. Most people will want to select the “Standard personal computer (x86 architecture, Pentium™, Celeron™, Athlon™, Sempron™)” checkbox. If you are unsure which option to choose, choose this option, as it will work in both cases.
5. Under the “Choose a location near you” heading, select a location nearest you (probably a North American location).
6. Click “Start Download” and select a location to save the “ubuntu8.04-desktop-i386.iso” file. The download will typically take up to a maximum of an hour using a broadband connection, but your mileage will vary considerably.
7. When the download is complete, insert a blank CD-R, CD-RW, or DVD-ROM in your CD-RW drive.
8. Locate the “ubuntu-8.04-desktop-i386.iso” file on your computer and right click on it. In the context menu, select “Copy image to CD”. Follow the steps given by the “CD Recording Wizard” pop-up. Wait until the wizard completes.

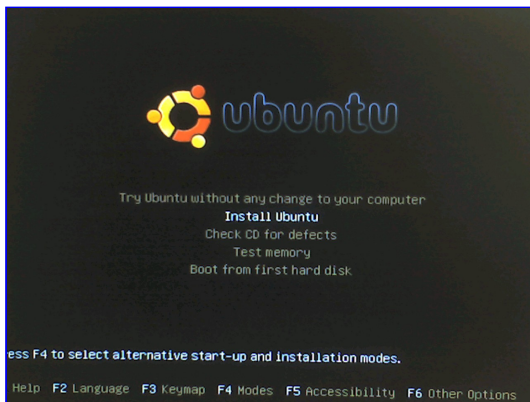
(2) Installing Ubuntu

1. Put the CD back in the CD-RW drive and restart your computer.
2. Your computer's BIOS must be instructed to boot from CD before it boots from your hard drive, if you want to boot from the Ubuntu installer CD. Most computers are configured to do this automatically. However, if you restart and the computer does not boot from the CD, but instead boots to the default Windows XP/Vista installation you must configure your machine to boot from the CD first. The exact process varies by computer, but the general process is as follows -
 1. Enter the “Del” key (on most computers) as the computer is first booting up to enter the BIOS.
 2. Navigate around the BIOS using your arrow keys, find the section entitled “Boot Priority”, and select the CD-RW drive to boot before the hard disk drive.
 3. To save all changes and exit, on most machines you press “F10”. Look around the BIOS for the exact keybindings for exiting and saving.
3. At this point, you should be presented with a screen with a variety of different languages. Select the language of your choice using the arrow keys, and press Enter.



(Note that the rest of the install process is fairly self explanatory, but for the sake of comprehensiveness it is broken down into steps here.)

4. Select “Install Ubuntu” from the menu using the arrow keys and press Enter.



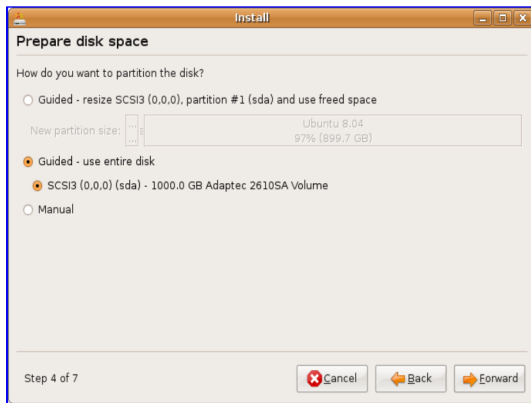
5. On the “Where are you” screen, select a city in your country and time zone. If you cannot find your city, select a city near you that shares the same time zone. Then, press Forward.



6. On the “Keyboard layout” screen, select the country of your residence on the left pane and then the keyboard that best reflects the layout of your keyboard on the right pane. You can test these settings in the provided textbox. Note that even if certain keyboard functions do not work during the install, it is likely that you can configure your keyboard correctly and easily when the operating system is up and running.



7. On the “Prepare disk space” screen, you will be presented with several options concerning the partitioning of your hard disk in order to make room for Ubuntu.

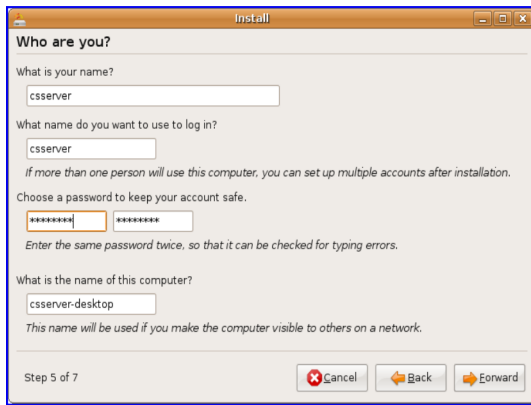


There are two common scenarios that may be applicable to you in regards to the partitioning process

1. If you want to completely dedicate your server machine to Ubuntu, and do not need to deal with any other operating system(s) already on the machine (such as Windows XP), select -“Guided – use entire disk” on the hard drive you want to install Linux to. (See the note below if you are unsure which disk is which). This option would best suit a dedicated server machine.
2. If you want to resize a hard disk and used the freed spaced to install Ubuntu on, selected “Guided – resize SCSI3 (0,0,0), partition #1 (sda) and used freed space” (this line may vary from computer for computer, but be sure to identify the line with “resize”). In most cases you will want to select this option if you have Windows installed on the server machine, but don't want to entirely get rid of it. This setup is called a dual boot (where two bootable operating systems exist on the same machine). (Note -If you have multiple hard drives on your machine, you may want to install Linux on a specific drive, and leave the others alone entirely. In this case, you might not immediately recognize which hard drive is which on the “Prepare hard disk space” screen. If you can't recognize the drive in question by the label given, a simple way to identify the drives is by the size of the drive – for example, if you have a 250 GB hard drive and a 350 GB hard drive, the size is a unique identifier and you can determine which drive you want to install in this manner.)

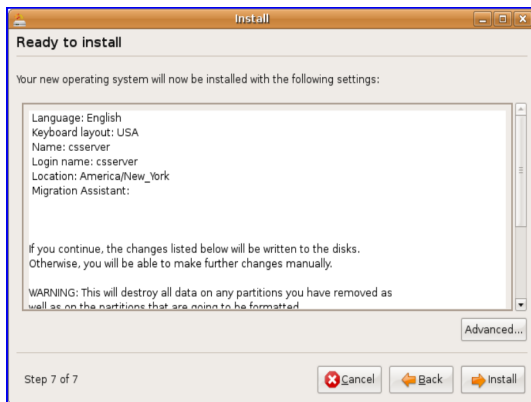
After selecting the option that best suits your situation, press Forward.

8. On the “Who are you?” screen, fill out the credentials that will be used to log into your server after it is installed.

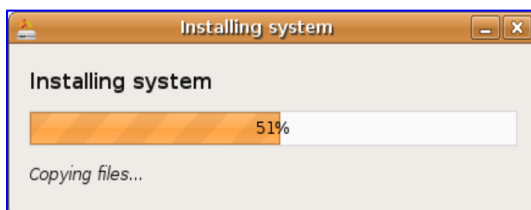
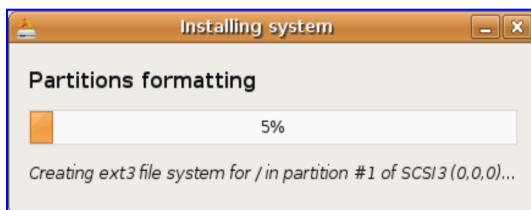


The only caution on this part of the install is to write down your username and password and store it in a safe place for later use, if you happen to forget your login credentials. Also, note that the "name" of the computer (in the image above, "csserverdesktop") will be the server's netbios identification on your Windows or Linux network (in more simplistic terms, it is the name that will appear in "My Network" if you choose to configure a network share later).

9. On the "Ready to Install" screen, press "Install". However, if you want to install the Grub bootloader on a different hard disk or partition, press Advanced to select the disk or partition you desire. (hd0 represents the MBR). For a normal dedicated server, you will not need to change any options here.



After pressing "Install," the operating system will install - specifically, it will first partition and format your hard disks and then copy files from the CD to the new partitions.



Depending on the size of your hard disk, the partitioning part of the install may take some time. If the installer appears to be frozen at a certain percent complete, it is probably just taking a long time to install - do not exit out of the installer or restart your computer until you are prompted to at the end of the process.

As this guide was made using the LiveCD installer approach, the "Installation Complete" dialog was not captured. However, simply follow the on screen instructions and continue to the next step while the computer is restarting (remember to take the CD out of the drive at the end of the install)

10. If you installed the server as a dedicated machine, the computer will simply restart and reach the ubuntu login screen, where you can easily log in using the credentials you created earlier. However, if you are dual booting, a black screen (grub bootloader menu) will appear (which is simply a list of bootable operating systems on your machine), just press Enter to load the default OS (Ubuntu 8.04). Eventually you will be presented with the screen to log in with.

(3) Installing the server components

At this point half the battle is over - we simply need to install the software to make the server run. There are two ways to approach this - by using the Synaptic package manager or by using the terminal. Each method has its pluses and minuses. Synaptic Package Manager will be more familiar to Windows users as it has a graphical interface - however, the tradeoff is the relatively large amount of time it takes to search for specific packages and check them off individually (you'll see this later). The terminal method is infinitely faster as you essentially just type out what packages you want to install in a list. Both methods are covered here, because both have their merits and you will find yourself using them extensively if you maintain or use your machine. However, the terminal method, which is listed second, is recommended.

Terminal Install Method (Recommended)

One of the older but faster ways to install software in Ubuntu is by using the terminal. The convention I use for terminal commands is as follows - each individual command is given a corresponding number. For example, the first command below is prefaced by " 1. ", meaning it's the first command. Obviously, each individual command should be treated separately and therefore go on its own line in the terminal.

1. From the main menu, select **Applications → Accessories → Terminal**.

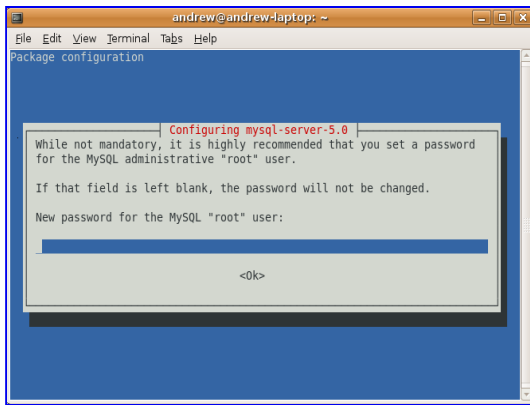
2. Type or copy and paste the following in the resulting window (you can right click in the terminal to paste)

```
1. sudo tasksel install lamp-server
```

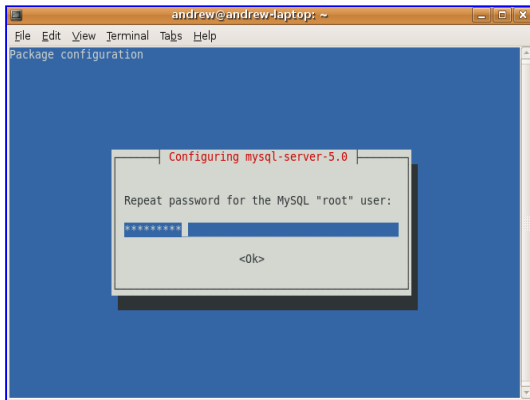
(*lamp-server* is a sort of metapackage that automatically installs MySQL, Apache, PHP, and their various dependencies automatically. It appears to be relatively new and extremely useful.)

3. After the packages are done installing and are configuring, you will be presented with a dialog to create a MySQL root password. You might need this later if you plan on creating databases and the like.

Type in a password and press Tab to focus OK, and press Enter to continue.



Repeat this process in the next password confirmation dialog.

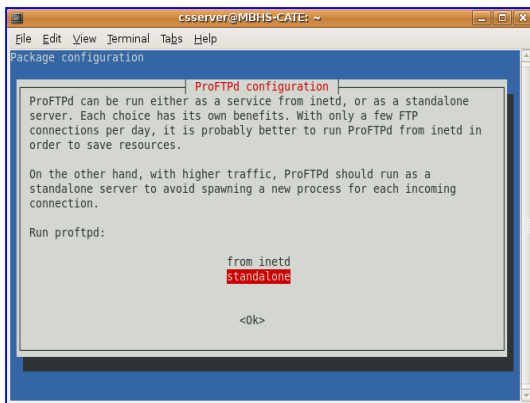


4. When the lamp-server package is done installing, type the following command in the terminal -

```
1. sudo apt-get install tomcat5.5 tomcat5.5-webapps gproftpd phpmyadmin libapache2-mod-jk
```

When prompted to confirm the installation of these packages, press "Y" and Enter to continue.

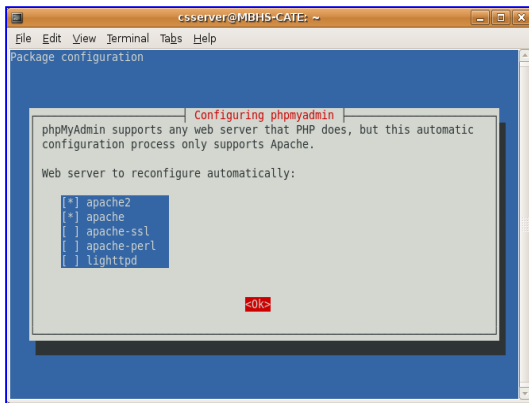
5. When prompted, select either the "standalone" or "inetd" mode for the FTP daemon (proftpd) using your arrow keys.



The differences are described adequately on the screen. **"inetd"** mode is usually best suited for low traffic sites that do not handle many FTP sessions. **"standalone"** mode is best suited for

high traffic, popular sites that handle many FTP sessions. For a dedicated server, either works fine, but "standalone" makes the most sense. For a multipurpose machine that is used as both a server and as a general desktop, "inetd" would be the logical choice. When you are done, press "Forward".

6. When prompted, select the versions of apache you want to configure with phpMyAdmin using your arrow keys to highlight the various options and space to select a particular option. To cover all your bases, it is recommended that you select both the apache2 and apache web servers. However, depending on your purposes, you may wish to select more web servers to configure. When you are done, press Tab to focus OK and press Enter to continue.

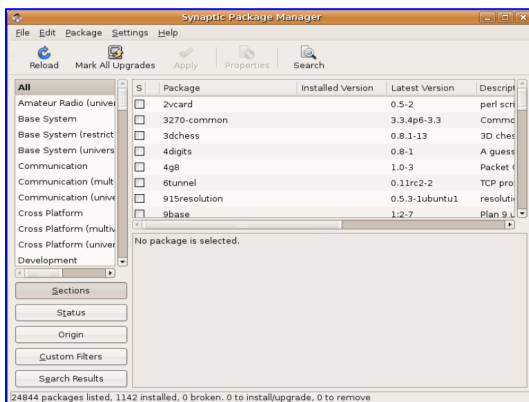


7. When the packages are done configuring (all activity ceases on the terminal), keep the terminal open and skip past the Synaptic Package Manager Method below and go straight to the section entitled "Configuring and Repairing the PHP, MySQL, Apache, Tomcat, and FTP Server Components".

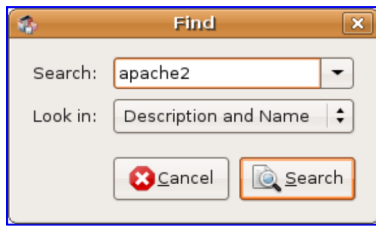
Synaptic Package Manager Method

(Remember, if you already used the terminal install method above, you can skip this section)

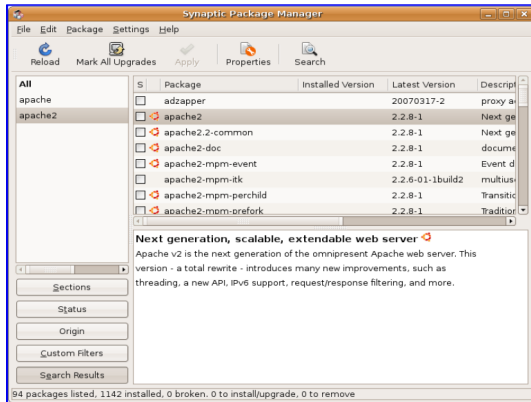
1. After logging into the server, select "System" from the menu at the top of the screen, select "Administration", and select "Synaptic Package Manager". When prompted, type in your password and click "OK".



2. Click Search and type "apache2" in the resulting dialog.



3. You will eventually be presented with a screen of search results.



Right click on "apache2" (the second item in the list) and select "Mark for Installation".

4. On the resulting "Mark additional required changes?" dialog box, select "Mark". (This step is just confirming that you want to install the various dependencies of apache2)

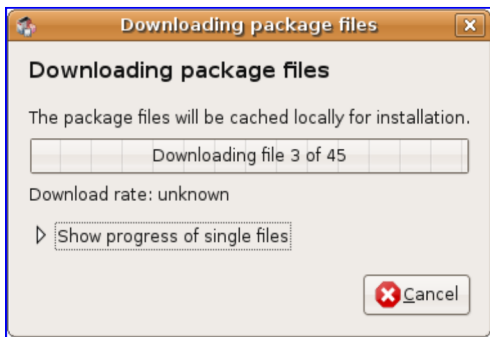
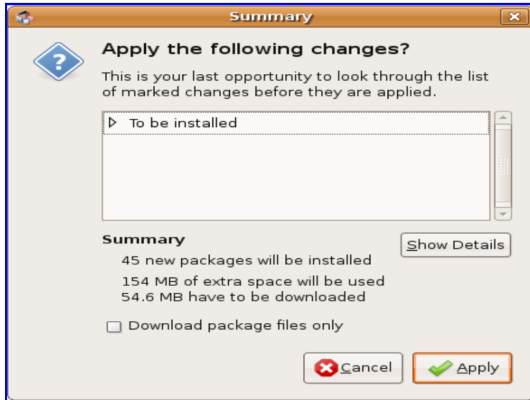
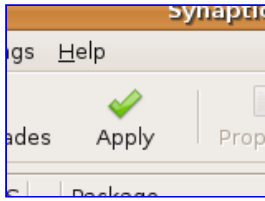


The checkbox next to apache2 should turn green at this point.

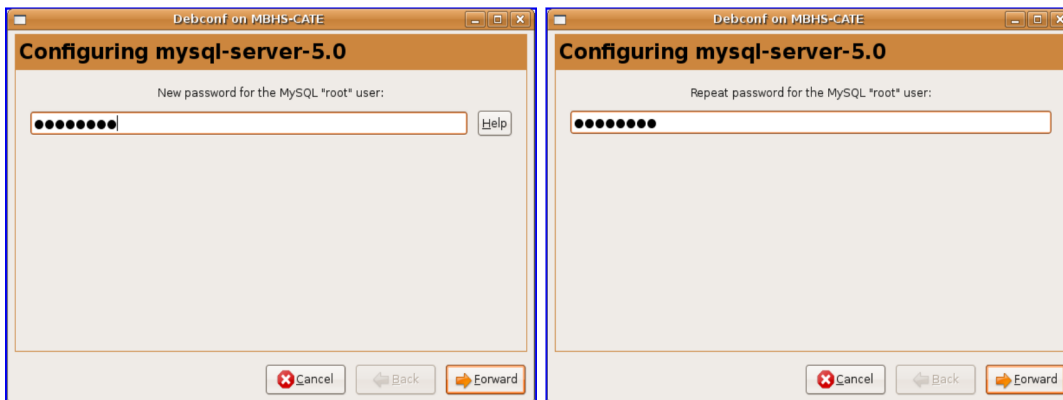
5. Repeat steps 2, 3 and (possibly) 4 for the following packages -

1. apache2
2. mysql-server
3. tomcat5.5
4. php5
5. phpmyadmin
6. gproftpd
7. proftpd
8. tomcat5.5-webapps
9. libapache2-mod-jk

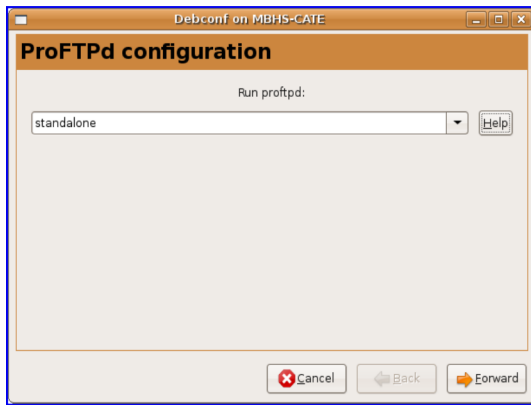
6. When you are complete selecting the last package, press "Mark All Changes". Follow the prompts.



7. When prompted, type in a password for the MySQL root user, press Forward, and repeat it in the next dialog. You might need this later if you plan on creating databases and the like.

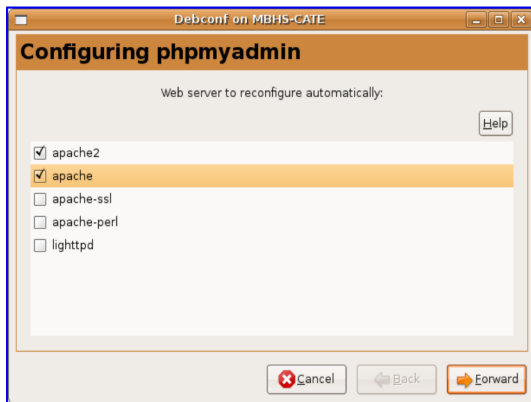


8. When prompted, select either the "standalone" or "inetd" mode for the FTP daemon (proftpd).



Clicking "Help" adequately describes the differences between each mode. Further information can be found in step 6 of "Terminal Install Method" section above.

9. When prompted, select the versions of apache you want to configure with phpMyAdmin. To cover all your bases, it is recommended that you select both the apache2 and apache webservers.



10. The components will now be automatically configured. When everything is done, close out of Synaptic.

(4) Configuring and Repairing the PHP, MySQL, Apache, Tomcat, and FTP Server Components

1. Open a terminal (**Applications → Accessories → Terminal**) and type in the following commands to make symbolic links to the Tomcat, Tomcat-webapps, and Java directories.

1. `sudo ln -s /usr/share/tomcat5.5 /usr/share/tomcat`
2. `sudo ln -s /usr/share/tomcat5.5-webapps /usr/share/tomcat-webapps`
3. `sudo ln -s /usr/lib/jvm/java-gcj /usr/lib/jvm/java`

You can change these lines as you see fit to reflect the actual locations of your Tomcat, Tomcat-webapps, and Java directories.

It should be noted that this step is not necessarily required, but will make future upgrades of either the Tomcat, Tomcat-webapps, or Apache packages much easier to manage (it saves you from having to manually update multiple configuration files, a lengthy process). We will also be using these links throughout the rest of the guide as we configure the server components.

It is important to understand the theory behind a symbolic link (also called a symlink, or soft

link). It is a special kind of file that points to another file, much like a shortcut in Windows. Unlike a hard link, a symbolic link does not contain the data in the target file. It simply points to another entry somewhere in the file system. This difference gives symbolic links certain qualities that hard links do not have, such as the ability to directly link directories as if they were one in the same. Think of it simply as a folder that automatically redirects to another folder that you specify.

If you upgrade or change Tomcat, Tomcat-webapps, or Java in the future, these symlinks will need to be changed from the terminal. (See the section entitled "Future Maintenance" for more info) However, as mentioned, this is much easier than manually editing configuration files to point to the new installation directories.

(Important side note - it is true that *java-gcj* is a symlink in itself to the current *java-gcj* folder, but I chose to symlink it separately just in case someone eventually decides to use the *java-6-sun* package or already had it installed before using this guide. If you know you are using *java-6-sun*, simply replace the text *java-gcj* in the third command above with *java-6-sun*)

2. As of the time this guide was being written, Tomcat 5.5 from the repositories has one minor setting that should be modified (Credit goes to [this](#)⁴ guide for discovering the issue). The modification of this setting is basically optional - Tomcat will run fine without changing it for the time being, but I really do suggest following it if you ever think you might install Java in a nonstandard prefix (location) at any point in the future. The issue is that the environmental variable `JAVA_HOME` (the location of the JDK on your computer) is not set by default in one of the Tomcat configuration files. The good news is that in our configuration, Tomcat will still run without explicitly setting `JAVA_HOME` because Tomcat is smart enough to search through typical Java installation directories, so it will find `/usr/lib/jvm/java-gcj` without complaint. The bad news is that this is not ideal as you may install Java in a nonstandard prefix in the future and Tomcat will complain accordingly if it can't find it. This could also be a problem if you're using another distro. To make sure Tomcat knows where Java is, after closing out of Synaptic, open a terminal (on the top menu, press **Applications** → **Accessories** → **Terminal**) and type in the following -

```
1. sudo gedit /etc/default/tomcat5.5
```

Type in your password if prompted, and in the resulting text editor, find the line

```
#JAVA_HOME=/usr/lib/jvm/java-6-sun
```

Delete the pound/comment sign (#) at the beginning and change "java-6-sun" to simply "java" (we symlinked this before). So the line should read

```
JAVA_HOME=/usr/lib/jvm/java
```

3. If you followed the optional step above, restart the Tomcat server daemon by opening a terminal and typing -

```
sudo /etc/init.d/tomcat5.5 restart
```

(This command may take a while to process - be patient)

If everything is configured correctly, you should be presented the message

```
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
```

4. To test your Tomcat installation, open Firefox and visit the address <http://localhost:8180/>. If everything is configured correctly, you will be presented with the Tomcat greeting page. If

something is amiss, you will get a "Page Not Found" error.

5. To test your Apache installation, open Firefox and visit the address <http://localhost>. If the message "It works!" is returned, your apache web server is running correctly.

6. As has been demonstrated, the Apache and Tomcat servers installed can be accessed individually from either port 80 (Apache) or port 8180 (Tomcat). This is not a problem for most people because Tomcat is fairly good at displaying static HTML files in addition to JSP files. So, if someone wanted to run a very basic server, they'd probably just use Tomcat so they would have HTML and JSP functionality. However, most people will agree that it is inconvenient managing two separate servers on the same machine -above all, it requires the specification of different ports in the web browser. Additionally, Apache has numerous modules (such as the cgi and php modules) that are built in or very easy to install, that provide it priceless functionality. Tomcat either doesn't have these modules or they are difficult/pointless to install. However, we can "combine" Apache and Tomcat with an apache module created for the purpose. It requires a little more configuration, and is not necessarily required if you understand the differences are fine with the default setup -but as I mentioned, it is a lot less messy and probably worth the extra time.

The connection module between Tomcat and Apache is provided by the packaged `libapache2-mod-jk`, which we installed above. However, a little modification of the configuration files for this package and apache itself is necessary to get things up and running. (Credit goes [here](#)⁵ for the basic steps)

The first file we need to edit is `mod-jk's worker.properties`. Type the following in a terminal

```
1. sudo gedit /etc/libapache2-mod-jk/workers.properties
```

Find the line

```
workers.tomcat_home=/usr/share/tomcat5
```

Modify the line to reflect the location of Tomcat on your machine. We symlinked it to `/usr/share/tomcat` earlier, so change the line to read

```
workers.tomcat_home=/usr/share/tomcat
```

Then find the line that begins with

```
workers.java_home=/usr/lib/jvm/java-gcj
```

Modify the line to reflect the location of the JVM (the JRE+JDK) on your machine. We symlinked it to `/usr/lib/jvm/java` earlier, so change the line to read

```
workers.java_home=/usr/lib/jvm/java
```

Save the file and exit `gedit`.

7. The second file we need to edit is Apache's `apache2.conf` file. We need to copy and paste a lot of text from one file to this file. The painless way to do this is by running the following commands from a terminal -

```
1. sudo -i
2. cat /usr/share/doc/libapache2-mod-jk/httpd_example_apache2.conf >> /etc/apache2
   /apache2.conf
```

(Yes, there is a space after the double shift arrows >>)

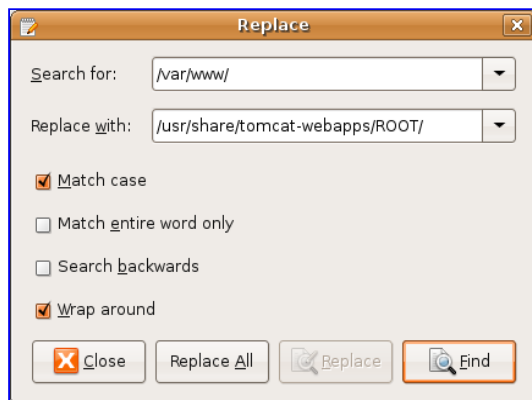
8. The final Apache/Tomcat related configuration file we need to modify is Apache's VirtualHost configuration file. The reason we need to edit it is because the way we have it set up currently, when a client requests a JSP file from your server (for example, `http://www.yoursite.com/test.jsp`), the `mod_jk` module will redirect this request to the folder `/usr/share/tomcat-webapps/ROOT` and look for the file "test.jsp" there - in other words, this folder is the JSP root. In a similar way, the HTML/PHP/CGI root (any Apache related file) works in the same manner but the requests gets sent to the root folder `/var/www`. This isn't necessarily a undesirable thing, but some people, especially me, consider it a hassle to have to deal with two separate directories to display all manner of web scripting. It is probably a better solution to consolidate everything into one folder and still have the module we loaded recognize which server (Apache or Tomcat) needs to load which type of file. Like the majority of this guide, this step is not required, but recommended. You can just as easily put your JSP files in the official Tomcat directory and other files in the Apache directory.

For the purposes of this guide, Apache's VirtualHost configuration will be modified so that Apache's new "root" folder is the same root folder as Tomcat (namely, `/usr/share/tomcat-webapps/ROOT`).

Open a terminal and type -

```
1. sudo gedit /etc/apache2/sites-enabled/000-default
```

9. The goal is to replace every reference in this file to the current Apache root, `/var/www/`, with the new one, `/usr/share/tomcat-webapps/ROOT/`. You can do this manually or type `Ctrl+H` to use `gedit`'s Replace function.



10. While editing this file, for security reasons, add the bolded text below into the correct position near the bottom of the text file (if all of the text is bolded below, the snippet is between the first `</Directory>` tag and the `</VirtualHost>` tag)

```
Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
Options Indexes MultiViews FollowSymLinks
AllowOverride None
Order deny,allow
Deny from all
```

```
Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
```

```
<Directory "/usr/share/tomcat-webapps/ROOT/WEB-INF/">
AllowOverride None
deny from all
</Directory>
```

```
</VirtualHost>
```

11. Let's restart Tomcat and Apache now with the following commands so that they use the modified configuration files.

1. `sudo /etc/init.d/tomcat5.5 restart`
2. `sudo /etc/init.d/apache2 restart`

12. We need to make a symbolic link to phpMyAdmin it in our new webserver root in order to test and use it.

Open a terminal and type -

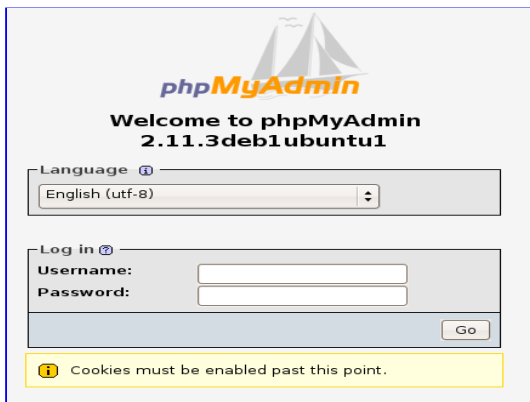
1. `sudo ln -s /usr/share/phpmyadmin /usr/share/tomcat-webapps/ROOT/phpmyadmin`

13. Let's make the webroot directory viewable to all -

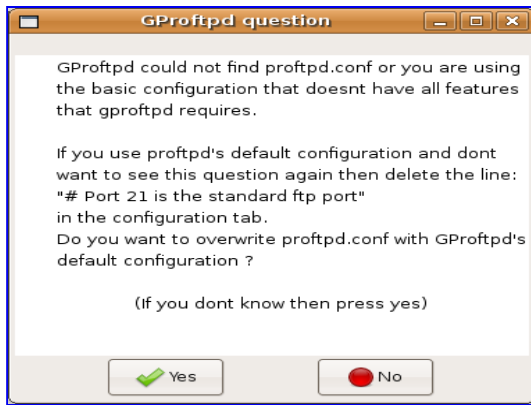
1. `sudo chmod -R +x /usr/share/tomcat-webapps/ROOT/`

From this point forward, you can place all your JSP/HTML/PHP files in the folder `/usr/share/tomcat-webapps/ROOT/` and view them locally via <http://localhost>. Note that this directory is not directly write able by the user. We handle this small issue in the "Setting up the work environment" section of this guide.

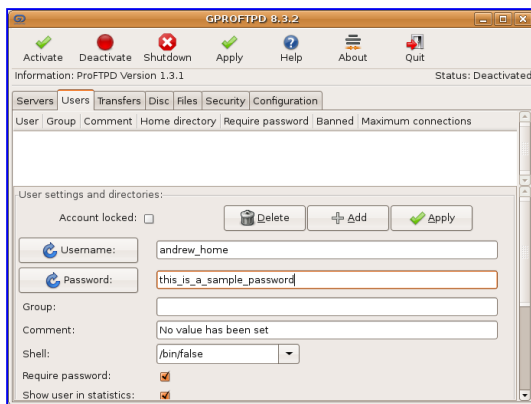
14. To test your PHP and MySQL installations, visit phpMyAdmin in Firefox at the address <http://localhost/phpmyadmin>. If php and MySQL are correctly installed, you will see the following page -

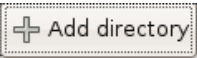



15. Now it is time to configure ProFTPD for use using the graphical configuration tool that was installed earlier, GPROFTPD. GPROFTPD can be found in **System → Administration → GPROFTPD**. After typing in your password, press "Yes" to the prompt that follows.



16. Select the "Users" tab near the top of the screen and type a username in the box labeled "Username" in the middle. Continue the process of creating a FTP user account by assigning a password to this account. Conventionally, the group "nobody" is assigned to new users. You should only worry about the username, password and group (nobody) - leave everything else to their defaults.





17. Scroll down to the bottom of this section and press . Click the resulting string , and type in a location of your choice for uploading files to your server remotely via FTP. The logical configuration here dictates that you make each user their own folder in the folder `/usr/share/tomcat-webapps/ROOT` and use the path to that folder here. That way, they would have immediate access to their own area of the website. (For example, make folders like `/usr/share/tomcat-webapps/ROOT/andrew`). Any folder/path you type in will be automatically created - you don't need to do that manually.

18. Assign the user as many privileges as you desire, using the appropriate checkboxes. The options are mostly self explanatory.



(Don't specify the directory `/var/www/name` like in the screenshot - This is an error in the screenshot, before I learned how to combine Tomcat and Apache)

You can safely assign a user all of these privileges and be fine.

19. Scroll up to the top and press  and then . Repeat steps 13, 14 and 15 for as many FTP users as you want.

20. When you are finished adding all users, press  to reactivate the server.

(5) Setting up the work environment

1. By default, the directory `/usr/share/tomcat-webapps/ROOT/` is not writeable by the user account you created using Ubuntu's file browser. Most consider it bad practice to directly work within this directory when developing files, and recommend creating a separate work directory to mess around with. We can easily handle this by using a Nautilus (Ubuntu's file browser) script to create a right click context menu in any folder to open this directory as root. (Credit goes [here](#)⁶ for the actual script).

Open a terminal and run the following commands -

```
1. cd ~/.gnome2/nautilus-scripts
2. wget http://g-scripts.sourceforge.net/nautilus-scripts/Execute/Misc/root-
nautilus-here
3. chmod +x root-nautilus-here
```

Now whenever you want to add a file to the web server root from the physical machine, copy it and browse to the folder using Nautilus (**Places** → **Computer**, and type `/usr/share/tomcat-webapps/ROOT/` in the "Location" bar). Right click within the folder, mouse over "Scripts", and select "root nautilus here". Type in your password, and an unprotected version of the directory will open for your usage. Then, you can simply paste the requisite files into the directory.

2. You may wish to install Kompozer for an easy graphical way to create websites. To install it, use Synaptic or open a terminal and type -

```
1. sudo apt-get install kompozer
```

3. In order to run applications like Kompozer as root, you may want to add an icon on your top panel (the one at the top of the screen) to graphically launch a program with root privileges.

To do this, follow these steps -

1. Right click on a blank/white part of your top panel and press "Add to Panel".
2. When the Add to Panel dialog appears, press "Custom Application Launcher" and click the "Add" button at the bottom of the screen.
3. In the "Name" box, type something like "Launch application as root".
4. In the "Command" box, type **gksu**.
5. Close out of the "Add to Panel" dialog.

Now, whenever you want to launch a graphical GUI application as root, you can simply click this icon and type in the program name.

(6) Future Maintenance

1. If you ever update the java, tomcat, or tomcat-webapps packages, you need to update the symbolic links that we made in the configuration above. This involves removing the older symbolic links and creating new ones to reflect the changes in the locations of the newly installed software. You would run commands like these, replacing the general locations (like

new_tomcat_location) with the new locations of the software on your machine -

```
1. sudo -i
2. rm /usr/share/tomcat; ln -s /usr/share/new_tomcat_location /usr/share/tomcat
3. rm /usr/lib/jvm/java; ln -s /usr/lib/jvm/new_java_location /usr/lib/jvm/java
4. rm /usr/share/tomcat-webapps; ln -s /usr/share/new_tomcat-webapps_location
   /usr/share/tomcat-webapps
```

Remember that some of the configuration files we edited depend on these symbolic links being up to date, so it is important to keep this in the back of your mind when you run a system update or the like - otherwise the software simply might fail.

(7) Command Cheat Sheet

Stop/Start/Restart Apache

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

Stop/Start/Restart Tomcat

```
sudo /etc/init.d/tomcat5.5 stop
sudo /etc/init.d/tomcat5.5 start
sudo /etc/init.d/tomcat5.5 restart
```

Stop/Start/Restart ProFTPD

```
sudo /etc/init.d/proftpd stop
sudo /etc/init.d/proftpd start
sudo /etc/init.d/proftpd restart
```

Edit a configuration file

```
sudo gedit /path/to/configuration_file
```

Run a single command as root

```
sudo programe_name
```

Pretend that you're the root user (use when sudo fails or you have many consecutive root commands to run)

```
sudo -i
```

Restart gnome/login/logout

CTRL+ALT+Backspace

Login to the MySQL CLI as root

```
mysql -u root -p
```

Install software

```
sudo apt-get install package_name
```

Fuzzy/Exact Search for a package

```
sudo aptitude search package_name
```

Kill a process by name

```
killall process_name
```

Copy a file to a folder

```
cp file_name /folder_name
```

Delete a file

```
rm file_name
```

Change directories

```
cd path_to_directory
```

Create a symbolic link to a directory or file

```
ln -s /path/to/real/directory /path/to/link
```

(8) Links Referenced

- 1 - shipit.ubuntu.com/
- 2 - <http://isorecorder.alexfeinman.com/>
- 3 - <http://www.ubuntu.com/getubuntu/download/>
- 4 - <http://ubuntuforums.org/showthread.php?t=436295>
- 5 - <http://ubuntuforums.org/showthread.php?t=422472>
- 6 - <https://answers.launchpad.net/ubuntu/+source/nautilus/+question/15598>

(9) Contact Me

I can be contacted at andrewlstein@gmail.com with questions, comments, concerns, praise, or general bug reports (there's bound to be mistakes the first time around)

You should also utilize the Ubuntu forums at <http://www.ubuntuforums.org> with any questions regarding Tomcat, Apache, or Linux in general - it's a wonderful resource.

If anyone is curious, I'm a junior (soon to be senior) from McKinney Boyd High School in McKinney, Texas. I hope you enjoyed this guide.