LAMP Server with Drupal or Moodle Setup Guide

Steve Hayes

Version 1.0, August 2012

This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of the license, visit http://creativecommons.org/licenses/by/3.0/deed.en_GB.

In essence, you can modify, distribute, copy, print and use the guide as you wish for commercial or non-commercial purposes so long as you include an attribution to the original author. Please also include a reference to drenewydd.com.

For further information, please contact steve@drenewydd.com.

TRADEMARKS/REGISTERED TRADEMARKS
Any computer hardware and software brand names mentioned in this guide are protected by their respective trademarks and are acknowledged here.

Table of Contents

1 Introduction....3
1.1 What is Open Source?....4
1.2 Six server systems....5
1.3 Advantages and disadvantages....6
1.4 Conventions....7
1.5 Placemarkers....7
2 Download Ubuntu....10
3 Make a VirtualBox server....12
3.1 Install VirtualBox on a Windows PC....12
3.2 Install VirtualBox on a Ubuntu PC....12
3.3 Create a virtual machine....13
3.4 Start the virtual machine....14
4 Set up a server at your location....16
5 Install Ubuntu....17
6 Starting with a hosted server....22
6.1 Selecting and ordering the server....22
7 Set up the server....24
7.1 Set up ssh access to your server....24
7.2 Log in to your server with SSH....24
7.3 Using SSH from a Windows PC....25
7.4 Using pscp....27
7.5 Bring the system up to date....27
7.6 Make sure programs are installed....28
7.7 Secure your server with a firewall....29
7.8 Block attacks....31
7.9 More tweaks....31
7.10 Set up PHP....32
7.11 Set up Apache....32
7.12 Install Midnight Commander....34
7.13 Database survival guide....35
7.14 Quit your ssh connection....37
7.15 Shut down and restart the server....37
7.16 Set up the server domain....38
8 Install Drupal....40
8.1 Placemarkers....40
8.2 Download Drupal....40
8.3 Install Drupal....41
8.4 Finish the installation....43
8.5 Ensure secure logins....44
8.6 Set up cron....46
8.7 Move Drupal to a different server....47

9 Install Moodle....49
9.1 Placemarkers....49
9.2 Download Moodle....49
9.3 Install Moodle....50
9.4 Finish the installation....52
9.5 Set up cron....54
9.6 Move Moodle to a different server....55
10 More Apache tweaks....57

1 Introduction

If you want to set up a server for a simple web site or a **Content Management System (CMS)** such as Drupal or Moodle, you need to start with a basic server. This guide shows, step by step, how to set up a LAMP server in a few hours.

A LAMP server uses the following open-source software components:

- *Linux – the operating system*
- *Apache – the web server. Apache on its own can serve up static web pages.*
- *MySQL – the database used to store the data needed to create dynamic web pages.*
- *PHP – the programming language that glues it all together and creates dynamic pages.*

Across the internet, LAMP is the most common server software.

The guide uses Ubuntu 10.04 as the Linux distribution. Ubuntu is very popular and is well supported if you run into problems.

The guide continues with installation details for Drupal and Moodle.

Drupal is a CMS suitable for constructing and maintaining many different kinds of website, including e-commerce sites. Moodle is a CMS intended for use in education.

You can, of course, use your server for something different, e.g. to support another CMS, a simple static web site, a file server or an e-mail server. One day, some of these options may be covered in the guide too.

The guide doesn't cover the configuration or use of either Drupal or Moodle. There are many books available covering each system but, for understandable reasons, they tend to gloss

over the installation steps. This guide should be a useful addition.

1.1 What is Open Source?

Computer programmers write programs as **source code** in any of a number of programming languages. The source code is translated into a program that the computer can execute – this process is done by an existing program known as a compiler.

Once a program is written, it costs next to nothing to make additional copies of it.

The business model of a proprietary software company such as Microsoft is based on charging people who want to use their software via licensing fees (they may pretend to sell their programs but the fine print shows differently). They often resort to legal and technical means such as activation to prevent unauthorised copies being used. They also strive mightily to keep the source code for their programs secret.

Although this approach helps to ensure their continuing income, it has many disadvantages:

- The user has to pay licensing fees.
- Additional fees may be due for every updated version of the program. This can deter users from updating, leading to an extra burden of supporting out of date systems.
- Additional licenses and fees may be needed for evaluation or standby systems.
- Activation steps can present hurdles and roadblocks when a computer is replaced (including during recovery from a system failure).
- The user may not be able to test and evaluate the program before committing to it.
- Neither the user nor anyone else outside the software company can make improvements to the program or incorporate it into a different product.
- No-one outside the software company can examine details of how the program works, e.g. if they want to make something

that interfaces with it.

- No-one outside the software company can review the program for security lapses or fix any lapses that emerge.
- In a situation where competing programs have different desirable features, there's little prospect of these being brought together.
- If the software company loses interest in the program or goes out of business, no-one else can take over maintenance (e.g. so the program can run on new computer systems). Users' data tied to the program can become inaccessible.

Because of all these disadvantages, another approach has emerged. This makes the source code for the program freely available to anyone who wants to examine, use or modify it (usually subject to a legal requirement to publish any improvements). Income comes not from licensing the program but from providing services such as installation and maintenance, training and custom modifications. This is the open-source approach.

> *As well as the Linux operating system, examples of open-source systems include Android (Linux based and used on many mobile phones) and BSD (similar to Linux but with a different license and used as the basis for all current Apple devices although they themselves aren't open-source).*
>
> ***By choosing an open-source system, you not only save on licensing fees but also benefit from the extra flexibility, security and long-term support that it offers.***

1.2 Six server systems

The server software needs a system to run on. Here are six options:

1. Server software added to an existing computer located at your own premises.
2. A server set up as a new virtual machine on an existing computer at your premises.
3. A dedicated server using a separate computer located at your premises.

4. A dedicated server using a separate computer located at a hosting company.
5. A server set up as a virtual machine at a hosting company.
6. A cloud server.

This guide doesn't cover options 1 or 6 though it might still be helpful in parts.

Unless you're reasonably familiar with Linux, it's strongly suggested that you start out by installing VirtualBox on your own computer and set up a server as a virtual machine on it (option 2 above). This is comprehensively covered in this guide. It can be done easily in a day and costs nothing other than your time.

Once you've done this, you'll understand the jargon, will be much more confident about the whole idea and will be able to make a more informed decision about whether and how to proceed. Your time won't be wasted: the virtual machine will be invaluable as a test bed where you can try out and demonstrate changes before implementing them on the real server.

1.3 Advantages and disadvantages

Whichever type of server you choose eventually, you'll almost certainly want to have it on line 24 hours a day. This isn't a problem with a hosted or cloud server (options 4-6) and it's only a minor problem (power consumption) with a dedicated server at your location.

If your users are going to be mainly or exclusively at your location (e.g. for e-mail, groupware or an intranet), it makes sense to have the server there too. If they're going to be accessing the server over the internet (e.g. for a public website or where your organisation has many different locations), the upload speed from the server location (less than a megabit per second for most DSL connections) is going to slow things right down. A faster connection is likely to be expensive and a hosted or cloud server may be more economical.

Hosted virtual servers are pretty reliable but a virtual server on one of your computers can't be more reliable than its host operating system. If that's Windows and/or if it's being used for other things too, you should think twice.

1.4 Conventions

This guide assumes you're reasonably familiar with computers in general, e.g. finding and opening a file or going to a web page. **It doesn't assume any Linux knowledge.**

You should be able to follow the guide exactly if you use Windows 7 or Ubuntu on your computer (screenshots are from Windows 7). It hasn't been tested yet on other versions of Windows but there shouldn't be problems with XP.

When an important term is introduced, it's shown in **bold**.

Sometimes you'll need to type in a specific phrase or look for a phrase on a web page or in a dialogue box. These are shown in italics, e.g. “type *virtualbox* into the search box”.

The guide concentrates on working with the command line. Even where a graphical tool such as *Webmin* might be available, the command line is usually quicker and simpler. Commands that you need to type in are shown `in a different font like this`.

> *There are paragraphs formatted in italics like this one. They contain information that may be interesting, useful and helps you to understand the steps being described.*

Sometimes there are steps that you may only need to do if a particular thing happens: e.g. when you're using VirtualBox or if you encounter a known bug. These steps are indented like this paragraph.

If you don't need to do the first indented step, you don't need to do any of the ones afterwards either.

1.5 Placemarkers

At various points in this guide, you need to type in information that's special to you. These items are shown in this guide as placemarkers in *italics and underlined*. Here's a list:

dbrootpassword	This is the root password for the MySQL database. You'll set it when you install Ubuntu or MySQL. It can be the same as *loginpassword*.

domain	This is the address (e.g. *example.com*) that a user types into their web browser to use your server. You should confirm now that the domain you want to use is available and you'll need to pay to register it. Once this is done, you can set it to point to the numeric IP address on the Internet of your server (see *serveraddress* below). Until then, you can add the *domain* to your computer's hosts file (Section 7.16).
hostname	This is a more-or-less arbitrary name for your server. You might want to use the same name as *domain*, minus *.com*, *.co.uk* or whatever (e.g. *example*). At least make sure it isn't already used on your own home or company network.
loginpassword	This is the password that you'll use to log in to, set up and maintain your server. Choose it now. It needs to be secure: 8 characters or more and not a name or dictionary word.
numericserveraddress	This consists of four numbers separated by dots, e.g. *123.45.67.89*. Chapter 5 shows how to find out what it is for a local server. You'll need to forward ports on your router to make the server accessible from the Internet and you should point *domain* at the router's public address. For a hosted server, the hosting company should tell you what it is or you can use a ping command (Section 7.16) to get it from *serveraddress*.
serveraddress	If you know *numericserveraddress*, you can use that as *serveraddress*. You can use the URL of a hosted server (e.g *server123456.hostingcompany.com*) instead if the hosting company gives you this.
smarthost	If your Internet connection is a consumer one, you may find that emails can only be sent via your ISP's own mail server. This is

	the ISP mail server's address (e.g. *mail.myisp.net* or perhaps a numeric address). Check with your ISP to find out what to use.
username	This is the name that you use along with *loginpassword* to log in to your server. You can usually use your own first name (e.g. *john*).
you@youremaildomain	This is your usual e-mail address that you check regularly. The server will send any urgent messages to you there.

There are placemarkers specific to Drupal and Moodle that are listed in the relevant chapters.

2 Download Ubuntu

A new version of Ubuntu is released each April and October. A version is given a number consisting of the year, a dot and the month (04 for April or 10 for October). Each version also has a name consisting of the name of an animal and an adjective starting with the same letter as the animal's name. The letters are in alphabetic order.

For example, the latest version is 12.04 Precise Pangolin. The previous version, released in October 2011, was 11.10 Oneiric Ocelot.

Ubuntu provide updates for most versions for two years from the release date but one version every two years is designated as a Long Term Support (LTS) version. Updates for these are provided for five years from the date of release. 10.04 (Lucid Lynx) and 12.04 are LTS versions.

If you're using this guide to select and set up a hosted server (Section 1.2, option 4 or 5), the hosting provider will install Ubuntu for you. You can proceed directly to Chapter 6.

If you're planning to use a hosted server eventually but want to set up a local (physical or VirtualBox) server first, look at what Ubuntu version the planned hosting provider (Chapter 6) is offering. It's likely to be an LTS version. There can be slight variations in setting up different versions and it'll be simpler if you install the same one on your local server.

If you're only planning to use a local server, this guide is based on version 10.04 and you'll have less grief if you choose that version too.

Start your web browser and go to releases.ubuntu.com.

Click the version that you want to use. Look for the *Server install CD* versions lower down the page. For safety, click on the *Intel x86* version. The file is downloaded and saved – it's big and this may take some time.

If the computer you're planning to use as the server has a reasonably recent Intel or AMD processor, the AMD64 version

should give slightly better performance. You can try it but it may not work at all.

Note where the file is saved – it's usually in your *Downloads* folder.

If you're setting up a VirtualBox server (Section 1.2, option 2), continue with Chapter 3. If you're setting up a physical one (option 3), continue with Chapter 4.

3 Make a VirtualBox server

A virtual computer is a simulation of a computer that runs on another computer. Programs installed on the simulated computer run unaware that it's only simulated.

You can use VirtualBox to create a simulated computer on your own real computer (Section 1.2, option 2). You'll install and run the LAMP server software and any CMS that you want on this simulated virtual server.

If you've already installed VirtualBox on your computer, you can go directly to Section 3.3 to create a new virtual server.

3.1 Install VirtualBox on a Windows PC

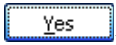
Start your web browser and go to www.virtualbox.org/wiki/Downloads

Click the VirtualBox version for Windows hosts. A *.exe* file is downloaded: this may take several minutes.

The file is released under the GPL2 license.

Once the file is downloaded, your web browser may start it running. If not, click on the file to run it.

The VirtualBox installer runs. The feature selection on the first two screens should be OK: click

each time. You may be warned about temporary network disconnection: click 

, then click

.

VirtualBox is installed. Windows may require you to allow this to happen.

Once it's installed, it may start automatically. If not, or when you need to start it again in future, you can start it like any other Windows application (e.g. *Excel*).

3.2 Install VirtualBox on a Ubuntu PC

Open the Ubuntu Software Centre and type *virtualbox* into the search

box. Find VirtualBox in the list that appears, click on it, then click Install.

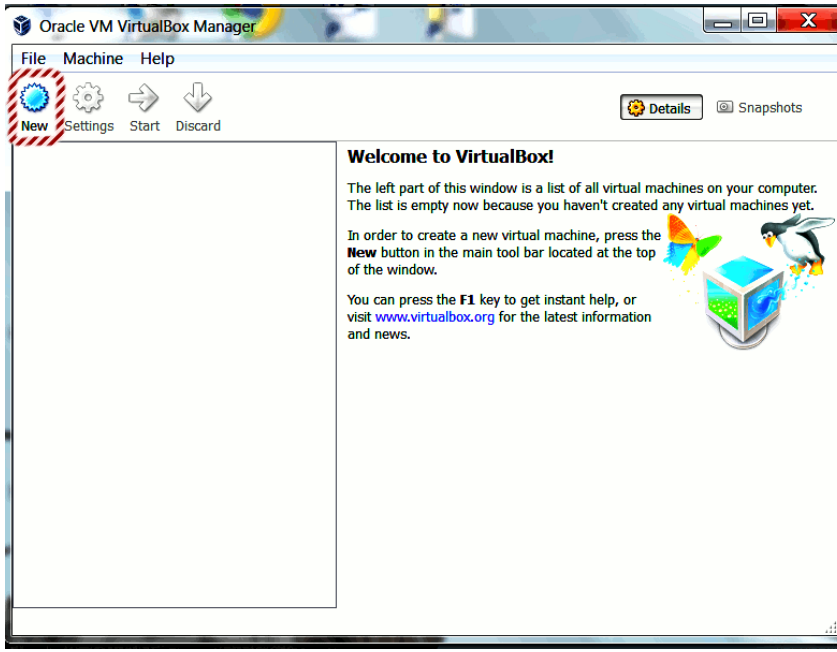
Once it's installed, there should be a new icon on the Launcher which you can click to start VirtualBox.

> *If the icon is missing, click on the Launcher, type vir into the search box, then click on VirtualBox to start it.*

3.3 Create a virtual machine

> *This procedure is the same whether your PC uses Windows or Ubuntu.*

Start VirtualBox.



Click New (circled). The *New Virtual Machine Wizard* starts. Click Next.

Give the new machine a name, e.g. *LAMP*. Use the dropdown boxes to select *Linux* as the Operating System and *Ubuntu* as the Version.

This is the operating system that you're installing on the virtual machine. Your PC can be using a different system, e.g. Windows.

Click Next. The next screen has a slider letting you choose how much of your computer's RAM will be used by the virtual machine. Unless your computer is short of RAM, you should set this to *1024 MB*. Click Next.

If you're very short of RAM (e.g. less than 1GB total), you're likely to have problems.

On the next screen, make sure *Start-up Disk* is ticked and *Create new hard disk* is selected.

The wizard is going to make a file on your computer that the new virtual machine will use to store the data in what it sees as its own hard disk. The size of this file and disk is shown.

Click Next. *VDI* should be selected as the *File type* for the disk. Click Next.

Choose *Fixed size* on the next screen – this keeps things simple. Click Next.

The settings on the next screen should be OK though you can change the virtual disk size here if you want. *8.00 GB* is a reasonable choice.

Click Next, then click Create. The virtual disk file is created. This takes a minute or two. Once it's done, click Create again to set up the virtual machine itself.

The VirtualBox manager appears again with the new machine listed on the left side.

3.4 Start the virtual machine

Click on the new machine in the VirtalBox manager to highlight it, then click Start to start it.

A new window appears along with a message about capturing of the keyboard. Make a note of which key is currently set as the **host key**.

When you press a key, your computer and VirtualBox need to decide whether the host system (Windows or Linux) sees it or whether the new virtual machine does. Usually, while the virtual machine window is active, it sees all key presses. If you need the host system to see it (e.g. so that pressing PRINT SCREEN makes a screenshot), you can press the host key first.

If you use the mouse to make a different window active, key presses are seen by the host system again: you don't need to use the host key in that case.

Click OK. The *First Run Wizard* starts. Click Next.

Click . A file selector box appears. Find the Ubuntu ISO file you downloaded in Chapter 2 and select it. Click Open.

To the new virtual machine, it looks as if this file had been burned to a disk and inserted in its CD/DVD drive. You can choose a different ISO file any time in the VirtualBox Manager – see Chapter 5.

Click Next, then Start. Make sure the new virtual machine window is active and at the front. You may see a message about display modes. These don't matter because you'll only be using text mode. Tick *Do not show this message again*, then click OK.

What you see now in the virtual machine window are the same messages you'd see if you were installing Ubuntu on a real computer. Continue with Chapter 5.

4 Set up a server at your location

Ubuntu is a very efficient operating system. Unless you're expecting very heavy usage of your server (you'd need a network connection good enough to handle it), you can reuse an old computer that you may already have. If you need a new computer for your server, you might be able to save money by buying one without an operating system (e.g. Windows) installed.

This guide assumes that, if the computer has an existing operating system and data on it, you don't care about losing that. If you want to keep it, you could fit a new hard disk and safely store the old disk. It may also be possible to install Ubuntu in a dual-boot configuration but this depends on many factors and isn't covered here.

Burn the Ubuntu ISO that you downloaded in Chapter 2 to a CD or DVD.

You can do this with any computer, not necessarily the one you'll use as the server. There are CD/DVD burning programs in both Windows and Ubuntu.

If the server machine doesn't have a CD/DVD drive, you can use a USB thumbdrive instead. See unetbootin.sourceforge.net.

Place the disk in the server's CD/DVD drive and power it up. After a short delay, the installer starts and you see the screen shown at the start of Chapter 5.

If you don't see this (e.g. if Windows starts up instead), you need to change BIOS settings so the machine boots from the CD/DVD drive ahead of the hard disk. Details vary from one computer to another but you need to look for a message that flashes up quickly after power is switched on and press the key that it mentions. Timing is critical and you may have to try several times before you see and can change the settings.

5 Install Ubuntu

In this chapter, you'll install Ubuntu on your new virtual or physical machine. If you're using a hosted server, it should already have Ubuntu installed on it and you can skip this chapter.

After installing and starting the virtual machine (Chapter 3) or the real server (Chapter 4), the Ubuntu Installer starts and you see a screen like this:

Make sure your language is selected, then press RETURN. Make sure *Install Ubuntu Server* is highlighted and press RETURN.

> *You can change the various installation selections with the keyboard arrow keys.*

Check that the correct installation language is selected, then press RETURN. Choose your location and press RETURN.

You may be asked whether to *Detect keyboard layout*. Use the left arrow key to highlight <*Yes*>, then press RETURN. Follow the

instructions that are shown. Note that you're asked about accented letter keys, not about the $ key.

> *As well as the £ sign, there are other differences between US and UK (gb) standard keyboards. If the wrong type is used, you'll have problems typing quote marks, etc.*

There's a delay, then you're asked for the machine's *hostname*. (Section 1.5). You'll also be asked to confirm or change your timezone.

When asked about partitioning, choose *Guided – use entire disk*. You need to confirm the disk before it's partitioned.

> *You might be able to make a different choice here so that an existing operating system and data is also kept in a dual boot configuration but, as mentioned in Chapter 4, this depends on many factors.*

You see a description of the partitions that the installer will delete and make. Check that these are OK, then select <*Yes*> and press RETURN.

> *If something looks wrong here, select <No> to cancel. The disk is left unchanged and you can start over with installation.*
>
> *A VirtualBox virtual machine only sees the disk that's saved in the virtual disk file made earlier (Section 7.3). You can safely let the Ubuntu installer tramp all over it.*
>
> *The same of course applies to a physical server if it has a new disk or you don't want to keep anything on the old disk*

The disk partitions are set up and Ubuntu is installed. This takes a while.

> You may see a message *No installable kernel found*. This is apparently a bug that affects VirtualBox installations on Vista or Windows 7. If it happens, press ALT+F2 and press RETURN to activate a new command line console in the virtual machine. You see a # prompt. Type `chroot /target` and press RETURN. Type `apt-get install linux-generic` and press RETURN. Press *Y* when prompted, then RETURN.
>
> After some delay (be patient), you may see a message about *grub-pc*. Press SPACEBAR so the highlighted choice is marked, then

press RETURN.

When you see the # prompt again, type `exit` and press RETURN. Another # prompt appears. Type `exit` and press RETURN again.

Press ALT+F1. The *No installable kernel* message reappears. Select <*Yes*> and press RETURN.

You're asked for your full name (e.g. *John Smith*), shown a *username* (e.g. *john*) that you can change if you want, then asked for a *loginpassword*. You need to type this in again to guard against typing errors.

> *You'll be using this user name and password to log in to the virtual machine*

When asked whether to encrypt your home directory, choose <*No*>. You may be asked for HTTP proxy information: you can normally leave this blank.

You may be asked how to manage updates on the system. Choose *No automatic updates*.

> *You'll be able to update the machine manually (Section 7.5) and can check afterwards to see if anything stops working as a result of the update.*
>
> *If you don't trust yourself to keep things up to date on a production (as opposed to a development, e.g. VirtualBox) server, choose Install security updates automatically.*

You're asked which software to install. Highlight *LAMP server* and press SPACEBAR to mark it. Do the same for *Mail server* and *OpenSSH server*. Press RETURN.

You're asked (twice again) for a password for the MySQL root user. Type in *dbrootpassword* (it can be the same as *loginpassword*).

You're asked to choose a mail server configuration. If you're setting up a server with a good quality network connection, choose *Internet Site*.

If you're setting something up that'll use a consumer/home grade internet connection, outgoing emails from the server might be blocked by your ISP or by some recipients. Find the address of your ISP's outgoing SMTP mail server (*smarthost*) and choose *Internet with smarthost*.

You might also want to choose Internet with smarthost to route all outgoing emails via another SMTP server on a company network. Note that you can't set up a smarthost server that needs a user name and password here.

If in doubt, choose Internet Site.

Check and adjust the System mail name (*domain*) that's shown (e.g. by adding *.com* or *.co.uk* to the machine hostname).

When your server receives an email, it'll only accept it if it's addressed to this domain.

You may only need the server to be able to send e-mails, e.g. to warn you of a problem. A separate guide is planned which will cover setting up a full email system.

If you chose *Internet with smarthost* above, you're asked for the mail server (*smarthost)* address.

You may be asked whether to install the GRUB boot loader to the master boot record. You can safely choose <*Yes*>. Wait for the installation to complete.

If you've installed Ubuntu on a real physical machine, remove the installation disk or thumbdrive.

If you don't do this, the installer will start again.

Choose <*Continue*>. The new server starts and, after a delay, you should see *hostname login:*.

Type your *username* and press RETURN. You're asked for your *loginpassword*. Type it and press RETURN.

After a delay, you see a **command prompt** showing your *username*, the *hostname* and your current directory. The prompt ends with *$*.

If it's a VirtualBox machine, type `sudo shutdown -h now` and press RETURN. You're asked for *loginpassword*. Type it and press RETURN. The virtual machine shuts down.

In VirtualBox Manager, click

, then

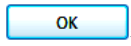
. If you see the ubuntu installation disk listed under IDE Controller, highlight it, click and choose *Remove disk from virtual drive* in the list that appears.

If you don't do this, the installer will start again when you restart the virtual machine. If you ever need to, you can select a different disk here too.

Click

. Make sure *Adapter 1* is enabled. Change its *Attached to:* from *NAT* to *Bridged Adapter*.

NAT allows the server to access the internet via the host using Network Address Translation but there's no way to initiate a connection to it from outside the host. Bridged Adapter gives it its own address on the network.

Click 

, then click

and log in to the virtual machine again.

At the command prompt, type `ifconfig` and press RETURN. You see a lot of information. Look for *eth0* on the left side. The line just below should start *inet addr:*. Note the four numbers that follow.

They should be something like 192.168.1.101. This is the ***numeric-serveraddress*.** Make a note of it: you can use it as the ***serveraddress*** too.

Continue with Chapter 7.

6 Starting with a hosted server

Details of how to purchase, access and use your server depend on which provider you choose. This guide can only give you a general idea of what is involved.

6.1 Selecting and ordering the server

Before you start, think about these questions:

- Do you want a virtual or dedicated server? *A virtual server would be appropriate unless you're expecting a very high usage level.*
- How much processing power and RAM is needed? *Ubuntu is an efficient operating system and, unless you're expecting a high usage level, a basic processor and 1GB of RAM should be sufficient.*
- How much disk space is needed? *Ubuntu and the software you'll install might need 1-2GB in total but you'll need space for the data (web pages, databases, emails) you'll have on the server too.*
- How much network traffic (often referred to as bandwidth) will you need? *This depends on the usage level – you'll incur extra charges if you go over the allowance.*

When you're looking at options, you'll want to consider the following:

- Can you get the server with the suggested Ubuntu version (10.04 server) preinstalled? If not, can you get a similar Linux version (another Ubuntu version or perhaps Debian)? *Other versions of Linux such as Red Hat, Fedora or SuSE are significantly different and you'll have problems following this guide.*
- How do you access the server's command line? *SSH is simple. KVM may be offered instead or additionally but isn't specifically covered in this guide.* ***Some servers may not have any command line access at all: you wouldn't be able to use this guide.***

- How do you restart and back up the server? *These tasks usually involve going to a web page where you may also be able to set up a firewall.*

Choose a hosting provider and order the server. The provider will tell you how to access it. You may be able to choose *username* and *login-password* or the provider may choose them for you.

The server will have a *numericserveraddress* (something like *123.45.67.89*) and will probably have a URL such as *server123456.hostingprovider.com*. You can use either of these as the *serveraddress* (Section 1.5).

> *When you're ready to point your domain to the hosted server, you'll use the numericserveraddress. Keep a note of it.*

If the hosting provider tells you that you can use *ssh* to access the server, you can continue with Chapter 7 now.

If you have to use KVM to access the server's command line, follow the provider's instructions.

7 Set up the server

While following this guide, you'll often need to access the command line prompt on your server.

If you're using a VirtualBox or physical server, you've already seen a command line prompt in Chapter 5. If you've logged out from the server, you can get back to it by logging in again.

With a hosted server, the provider should have told you how to get to the command line prompt. If this requires ssh, you can continue with Section 7.2 now.

7.1 Set up ssh access to your server

At the server's command line prompt, type `which sshd` and press RETURN. You should see something like */usr/bin/sshd*, then see a new command prompt.

> *sshd is the daemon program that looks for and handles incoming ssh connections. The* `which` *command tells you what folder it's in.*

If you only see the command prompt again, OpenSSH isn't installed. Type `sudo apt-get install openssh-server` and press RETURN. You'll be asked for *loginpassword* and whether to continue.

With a hosted server or if you want to be able to access the server from outside your local network, you'll need to open or forward port 22 to the server. This may involve changing a setting on a home router or company firewall or using the hosting company's control panel for the server.

7.2 Log in to your server with SSH

> *SSH (Secure Shell) allows you to access the command line of your server securely from anywhere. No-one snooping on the internet can see your login password or any of your commands.*

You'll need scp (secure copy) in Chapter 8 and/or 9 to copy files to the server, e.g. to set up a web site on it. So long as you can access the server with ssh, scp should work too.

If you're using Windows on your PC, go to Section 7.3.

If you're using any version of Linux on your PC, open a terminal session.

To do this in Ubuntu, you can click *, type term into the search box, then click on Terminal.*

You see the command line prompt for your own computer. **Note that this is different from the command line prompt for the server.**

Type `ssh` *username*@*serveraddress*. (Section 1.5). Press RETURN.

You'll probably be warned that the host authenticity can't be established and asked whether you want to continue connecting. Answer *yes*.

This is normal the first time you connect to the server. You shouldn't see this message or any others regarding its key unless you're connecting from a different computer or have reinstalled the PC or server's software. If you see one of these messages unexpectedly, be suspicious.

After a short delay, you're asked for the login password. Type *login-password* and press RETURN.

After a short delay, you should see the command prompt from the server. It shows the user name, host name and current directory and it ends with either $ or #. Continue with Section 7.5.

7.3 Using SSH from a Windows PC

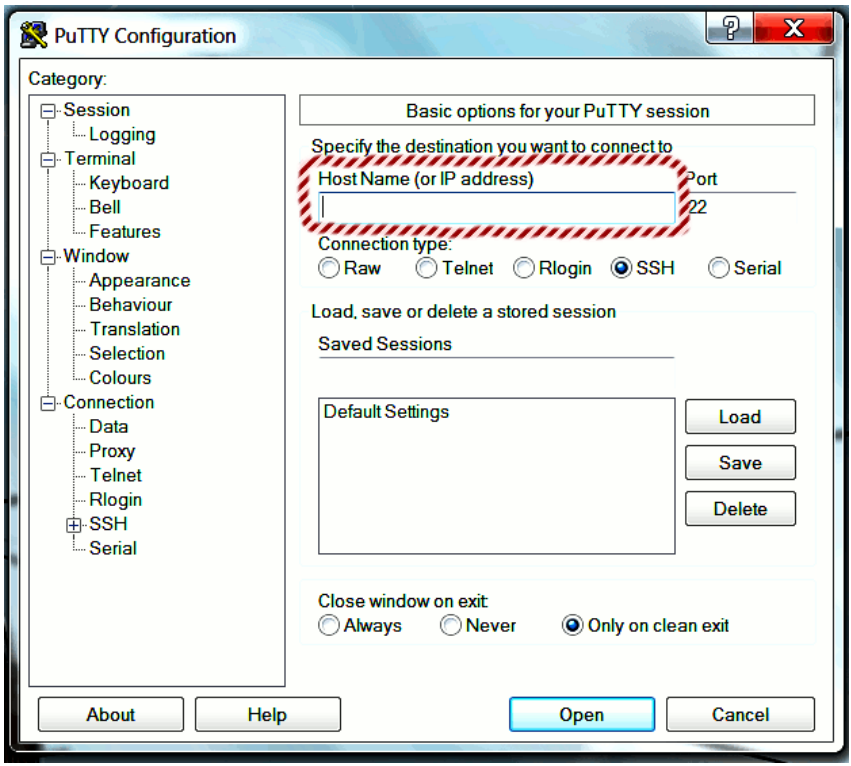
Windows doesn't come with an ssh program – you need to install PuTTY. Go to
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
and click on *putty-0.62-installer.exe*.

The version of PuTTY may have changed since this was written.

Once the file is downloaded, your web browser may start it running. If not, click on the file to run it.

Windows may require you to allow this to happen.

You can leave the various installation options as they are. Once PuTTY is installed, you can start it like any other Windows application (e.g. *Excel*). You see something like



Type *serveraddress* or *numericserveraddress* into the *Host Name* box (circled above) and make sure SSH is selected just below. Click Open.

You'll be warned about the server's host key. Click Yes.

> *This is normal the first time you connect to the server. You shouldn't see this message or any others regarding its key unless you're connecting from a different computer or have reinstalled the PC or server's software. If you see one of these messages unexpectedly, be suspicious.*

You see *login as:*. Type *username* (e.g. *john*) and press RETURN.

You're asked for the password. Type *loginpassword* and press RETURN.

After a short delay, you should see the command prompt from the server. It shows the user name, host name and current directory and it ends with either $ or #.

7.4 Using pscp

If you're installing a CMS such as Drupal or Moodle, you'll need to copy files from your own computer to the server. Chapters 8 and 9 show how to do it using *scp* but that program isn't available in Windows.

Instead, you can use *pscp*. It's included with PuTTY but it needs to be started from the Windows command line.

Click the Start button or orb, click *All programs*, then *Accessories*, then *Command Prompt*. A new window opens and you see the command line prompt for your own computer. **Note that this is different from the command line prompt for the server.** When the guide says to type `scp`, replace `scp` with `"\Program Files\PuTTY\pscp"` in the command.

> *This assumes that PuTTY was installed in \Program Files\PuTTY. The quotes are needed because of the space in Program Files. If you installed PuTTY in a different folder, use that location instead.*
>
> *Of course you can add the folder to your system path: then you'll only need to type* `pscp`.

7.5 Bring the system up to date

Type `sudo apt-get update` and press RETURN. You're asked for *loginpassword*. Type it in and press RETURN. The machine downloads information about the latest versions of programs from Ubuntu.

> *This assumes you're logged in as a normal user and can only make limited changes to the system. When you type sudo followed by some command, you're asked for your password,*

then the command runs as if you were the root user who is allowed to see and change everything.

You must be careful when you use sudo – you could break something important or unwittingly install malware. If someone tells you to use sudo, don't do it unless you trust them.

If you're using a hosted server (rather than a local VirtualBox or physical one) and username is root, you'll be logged in as the root user and the command prompt will end with # instead of $. In that case, leave out sudo from all commands where it's shown. E.g. just type `apt-get update`*. You won't be asked for your password.*

Command-line commands in this guide are shown in a different font `like this`*. Whenever you see a command (whether it's for the server or for your own computer), you'll need to make sure the command prompt is showing before you start typing it and you'll need to press RETURN after typing it.*

Type `sudo apt-get upgrade`. You may be asked for *loginpassword* again, then the Ubuntu installation on the machine is brought right up to date. You'll be asked to press *y*, then RETURN to confirm that you want to do this.

apt is a program used on Debian, Ubuntu and Mint versions of Linux to install, upgrade and remove programs. Red Hat, Fedora and SuSE use rpm instead – this isn't covered here. They also keep some files in different places.

Unless you chose *Install security updates automatically* when you installed Ubuntu, you'll need to repeat these two commands (`sudo apt-get update` and `sudo apt-get upgrade`) regularly to keep your system up to date and secure. E.g. you might want to do this weekly or even daily.

7.6 Make sure programs are installed

When you see the command prompt again, type `sudo apt-get install apache2 mysql-server php5 postfix`

If you installed Ubuntu yourself (Chapter 5), these programs should already be installed and you'll see messages to that effect. You can

continue with Section 7.7.

If you're using a hosted server, some programs may be missing. They'll be installed after you press *y* and RETURN.

You'll be asked (twice) for a password for the MySQL root user. Type in *dbrootpassword* (it can be the same as *loginpassword*).

You're asked to choose a mail server configuration. If you're setting up a server with a good quality network connection, choose *Internet Site*.

If things get stuck before you can make this choice, press TAB so <OK> is highlighted, then press RETURN.

If you're setting something up that'll use a consumer/home grade internet connection, outgoing emails from the server might be blocked by your ISP or by some recipients. Find the address of your ISP's outgoing SMTP mail server (*smarthost*) and choose *Internet with smarthost*.

You might also want to choose Internet with smarthost to route all outgoing emails via another SMTP server on a company network. Note that you can't set up a smarthost server that needs a user name and password here.

If in doubt, choose Internet Site.

Check and adjust the mail name (*domain*) that's shown (e.g. by adding *.com* or *.co.uk* to the machine hostname).

When your server receives an email, it'll only accept it if it's addressed to this domain.

You may only need the server to be able to send e-mails, e.g. to warn you of a problem. A separate guide is planned which will cover setting up a full email system.

If you chose *Internet with smarthost* above, you're asked for the mail server (*smarthost)* address.

7.7 Secure your server with a firewall

Home and company networks are normally protected from outside attack by an existing firewall, e.g. in a home router). A

hosted server may also be protected by an existing firewall – if so, the hosting provider will give you a way to administer it via a web interface.

The main function of a firewall is to prevent incoming connections to network ports other than the ones you know need to be allowed. There could be programs on the server that listen on other ports and have vulnerabilities.

Unless you're only anticipating connections (including viewing web pages) from inside your home or company network, you'll want to make sure that ports 22 (ssh), 25 (smtp), 80 (http) and 443 (https) are open at the firewall or forwarded to your server by the router.

If your server isn't protected externally from attacks from the internet (including some hosted servers or ones in a router's DMZ) or if you think it might be attacked from inside your home or company network, you can set up a firewall on the server itself by going to its command prompt and typing these commands in sequence:

```
sudo ufw disable
```

This temporarily disables the firewall, leaving all ports accessible. It's important if you're using ssh to access the server. You could lock yourself out otherwise.

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```

```
sudo ufw allow ssh
```

```
sudo ufw allow smtp
```

```
sudo ufw allow www
```

```
sudo ufw allow https
```

These four rules let your server accept ssh connections (port 22), send and receive e-mail (port 25) and show unsecured (port 80) and secured (port 443) web pages.

```
sudo ufw enable
```

The `sudo ufw status numbered` *command shows the rules that you've set. It should show rules numbered 1 to 4. You can*

delete a rule using `sudo ufw delete` *n where n is the rule number (1 to 4 in this case).*

7.8 Block attacks

You may get unwelcome attackers on the internet who try to guess login passwords or try to use your mail server to relay spam. If you've chosen secure passwords these attacks shouldn't be a real threat but they're still a nuisance. They cause extra load on your server and internet connection and fill up your log files with extraneous stuff. The fail2ban program is a good way to reduce these problems. It identifies attacks and sets your system firewall to reject all connections from the attacking address for a period of time.

Type `sudo apt-get install fail2ban`. Again, you may see *Do you want to continue [Y/n]?*. You can just press RETURN to select the capitalised default option (*Y*).

Fail2ban is installed and set up to block attackers who keep trying ssh passwords to see if they can log in to your server. They will be barred for 10 minutes after 6 tries.

7.9 More tweaks

Type the following commands:

```
sudo apt-get install nano php5-cli mailutils
```

This installs the nano text editor and useful programs for administering PHP and emails.

Type `mail -s "Test 1"` *you@youremaildomain* replacing *you@youremaildomain* with your normal email address. Press RETURN twice.

Type a message to yourself, pressing RETURN as needed. When it's finished, press RETURN, then CTRL+D. The message is sent and you see the command prompt again.

`mail` *is a very old and basic command line email program. It's still useful for testing.*

You can continue setting up the server. After you've finished, check that the e-mail you sent here has arrived in your usual inbox. If so, you'll know that the server and programs running on it can let you know if something is wrong.

7.10 Set up PHP

Type the following commands:

```
sudo apt-get install php5-curl php5-xmlrpc php5-gd
php5-intl php5-mysql
```

These PHP extensions are needed for Drupal and Moodle. If you find that other extensions are needed too (e.g. for a different CMS), you can install them the same way.

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

This ensures that the Apache web server knows about the new PHP extensions.

```
php -v
```

This shows you information about your PHP installation, including its version number.

7.11 Set up Apache

Type the following commands:

```
sudo a2enmod ssl
```

```
sudo a2enmod rewrite
```

These enable the SSL and Rewrite modules in Apache.

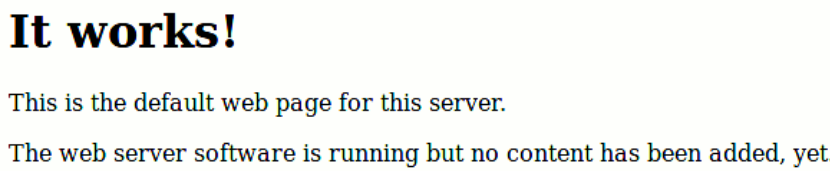
```
sudo a2ensite default-ssl
```

This puts a link to /etc/apache2/sites-available/default-ssl in /etc/apache2/sites-enabled making https (ssl) available.

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

Apache doesn't notice changes you've make to its configuration until you do this.

Start a web browser on your own computer. Type *serveraddress* into its address box and press RETURN. You should see something like



> *This is the default web page (/var/www/index.html) served up by Apache when it's freshly installed.*
>
> *You can create and upload your own static web pages to /var/www on the server. For example, if you create a web page called mypage.html and save it in /var/www, it'll be displayed when someone goes to serveraddress/mypage.html. If you call it index.html and replace the existing /var/www/index.html, the new page will be displayed whenever someone goes to serveraddress alone.*

Type *https://serveraddress* into the browser address box.

You'll probably see a warning about the security certificate. Exactly what appears and what options you have depend on your particular web browser. Go ahead and proceed to the site but, if given the choice (e.g. with Firefox), don't permanently store the exception.

> *You want to continue seeing what other users see when they go to your site.*

You should again see the default Apache web page.

This shows that secure web access (*ssl/https:*) is working although you'll need to buy a certificate to avoid your visitors seeing warnings.

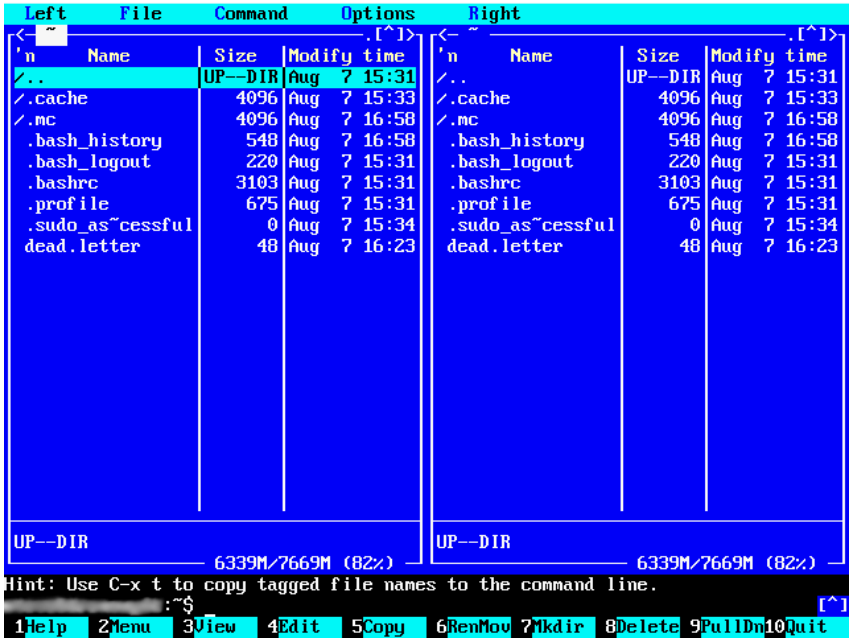
> *You might have problems if you're using Microsoft Internet Explorer. Try Firefox or Chrome instead.*
>
> *A separate guide covering encryption and certificates is planned.*

7.12 Install Midnight Commander

Midnight Commander *is a command-line program that makes it much easier to work with files (e.g. view, copy and move them) It gives you most of the advantages of a graphical file manager without giving up the speed and simplicity of the command line.*

Type `sudo apt-get install mc`.

Type `mc`. Midnight Commander starts.



You can use the ***Function keys*** *F1-F10 on your keyboard. The list across the bottom shows what each one does. For example, F10 exits Midnight Commander (type* `mc` *again to restart it).*

Above the function keys is a command prompt. You can type a command here and press RETURN to execute it. For example, `cd ..` *takes you up one directory level.*

The main part of the screen has two panes containing lists of files and directories. One shows the current directory and the other shows the contents of another directory. You can switch

between the two directories with the TAB key. Notice that the command prompt changes as you do this – TAB is equivalent to a `cd` *command.*

One of the files or directories in the active pane is highlighted. You can select which one with the up and down arrow keys on the keyboard.

When the command line is empty, you can switch to a directory by highlighting it and pressing RETURN. .. (shown highlighted above) is the directory containing the current one.

You can highlight a file or directory in one pane and press F5 to copy it or F6 to move it to the other directory/pane.

You can view the contents of a text file by highlighting it and pressing F3. Pressing F4 opens it for editing.

While you're typing on the command line, you can insert the name of the highlighted directory or file by holding down the Alt key and pressing RETURN.

Press *F9*, then *O*, then *C*. *Configuration options* open. If the option to *show Hidden files* isn't enabled (marked *[x]*), use the arrow keys to move to it and press the *Space bar*. Press *TAB* until <*OK*> is highlighted, then press *RETURN*. This ensures that files whose names start with a dot are shown by Midnight Commander.

Midnight Commander has many other useful features. Pressing F1 starts a help system which may help you to find them.

7.13 Database survival guide

Content Management Systems such as Drupal and Moodle store most of their information in MySQL databases. When you install one of these systems, you'll need to create a database for it.

It's handy to know a bit about MySQL so you understand what you're doing and can troubleshoot or undo any mistakes you make. This section demonstrates some useful SQL commands – you can skip it if you already know them.

Start the MySQL command-line tool by typing `mysql -u root -p`.

You'll be asked for the *dbrootpassword* that you chose when you installed Ubuntu or MySQL, then you're logged in as the MySQL root user and you see the prompt (*mysql>*). Type

```
status
```

You see information which might be useful sometimes.

You can type SQL commands: note that these need to end with a semi-colon (*;*). Type

```
show databases;
```

All your current MySQL databases are listed – there may only be one or two. One of them should be mysql – this is where MySQL keeps its own data such as its authorised users.

```
show tables from mysql;
```

The names of the tables in the mysql database are listed. One of them should be user – this is where MySQL keeps data about its individual users.

```
show columns from mysql.user;
```

The names of the columns in the mysql.user table are listed. Three of these are host, user and password – host specifies which computer(s) the user is allowed to log in from.

```
select host,user,password from mysql.user;
```

Authorised users are shown.

```
create database testxyz default character set utf8
collate utf8_unicode_ci;
```

A new database called testxyz is created.

```
grant select,insert,create,create temporary tables
on testxyz.* to testxxxuser@localhost identified by
'secretxyz';
```

A new user called testxxx is created. They are allowed to make changes to tables in the testxyz database but can only log in from the computer that the database is installed on (localhost). Their password is secretxyz.

```
use mysql;
```

This avoids having to specify the mysql database when working with its tables.

```
select host,user,password from user;
```

You can see the new testxxx user. Oops, we should have given them the name testxyzuser!

```
update user set user='testxyzuser' where user='testxxxuser';
```

```
select host,user,password from user;
```

The user's name is changed. You can of course use update to change any field of any record in any table if you have the necessary permissions.

Although passwords are encrypted, the update command can be used to set one user's password to be the same as another user. If a password is forgotten, you can create a new user, then set the existing user to have the same password before deleting the new user.

```
drop user testxyzuser@localhost;
```

```
drop database testxyz;
```

This removes the junk we just added.

```
quit
```

This quits mysql and returns you to the normal command prompt. Note that there's no semicolon this time.

7.14 Quit your ssh connection

To end the ssh session leaving the server running (what you'll usually want), just type `exit` and press RETURN.

If you were using Midnight Commander, `exit` *quits that. Type* `exit` *again to quit the session.*

7.15 Shut down and restart the server

When you need to, you can shut down the server from the command

line by typing

```
sudo shutdown -h now
```

> *You'll need access to the server's power button (or VirtualBox* *) to start it up again. With a hosted server, you should be able to restart it from the hosting provider's control panel but be sure to check this first.*

If you just want to restart the server (e.g. because it's misbehaving), type

```
sudo shutdown -r now
```

> *The server shuts down, then restarts immediately. You don't need access to its power button or control panel.*

7.16 Set up the server domain

If you're planning to make the server publicly accessible (right away or later on), you'll want to register *domain*. You can register the name (if it's not already taken) via any of a vast number of companies though this will cost something.

Once you've done this, you'll be able to point *domain* to *numericserveraddress*. E.g. if you register *example.com*, have a hosted server and the hosting company tells you its address is *123.45.67.89*, you'll use the registering company's control panel to point *example.com* to *123.45.67.89*.

> *If you know serveraddress but not numericserveraddress, use* `ping` *serveraddress at the command line of any computer. You'll see four numbers separated with dots in the response. This is numericserveraddress.*

If you want to test your server right away (after you've confirmed that the domain name is available but before you actually pay for it), you can put *domain* in your computer's hosts file and point it at the server's *numericserveraddress*.

> *Until you've paid for the domain, there's a risk of someone else nabbing it.*

In Windows 7, click and type *notepad* into the search box. Right

click on Notepad and choose *Run as administrator*. You'll need to confirm.

Notepad starts. Choose *File>Open* and navigate to *\Windows\System32\drivers\etc\hosts*. Choose *All files (*.*)* instead of *Text Documents (*.txt)*. Open *hosts*.

Add *numericserveraddress domain* (e.g. *123.45.67.89 example.com*) as a new line in *hosts*. Save *hosts* with *File>Save*.

In Linux, edit the */etc/hosts* file on your computer as root and add the same line as above.

On the same computer (Linux or Windows), type `ping` *domain*. You should see *numericserveraddress* in the response.

> *A computer needs the numeric IP address of a server to connect to it. Computers can look in their hosts file to find the numeric IP address corresponding to a domain. They consult the Internet's Domain Name System (DNS) to get it if the domain isn't listed in hosts.*
>
> *Other users won't have domain in their hosts files. They won't be able to reach the server until you've properly registered domain and DNS knows about it.*

8 Install Drupal

Only follow this chapter if you want to use Drupal.

8.1 Placemarkers

The following are in addition to the list in Section 1.5:

drupal	This is the name of Drupal's MySQL database. It's arbitrary: you can use *drupal*.
drupaladminuser	This is the user name of the first person who can administer and modify the Drupal installation and website. They can authorise additional administrators.
drupaladminpassword	This is the first administrator's password. Be sure it's strong and kept securely so the site doesn't get hacked.
drupalfilename.tar.gz	This is the name of the file that you download in Section 8.2.
drupaldbpassword	You'll choose this password which Drupal uses to access its MySQL database. The password isn't stored securely so don't reuse an important one.
drupaldbuser	This is the user name which Drupal will use to access its MySQL database. It's quite arbitrary: you can use *drupaldbuser*.
drupalfileowner	You'll find this out in Section 8.5: it seems to usually be *6226.*

8.2 Download Drupal

Go to *drupal.org/project/drupal*. Scroll down the page until you see the available downloads. Click to download and save the *tar.gz* package for the current version,

This is 7.14 at the time of writing.

The file is downloaded to your computer, usually to your *Downloads* folder. This may take a few minutes.

When it's complete, start the command line prompt for your own computer (Section 7.2 or 7.4) and type the following commands:

```
cd Downloads
```

> *If the downloaded file is somewhere else, go to that folder instead.*

`scp` *drupalfilename.tar.gz username@serveraddress*:.

> *Replace drupalfilename.tar.gz with the exact name of the file you downloaded. You can probably just type drupal, then press TAB to let your computer fill in the rest.*
>
> *Replace username@serveraddress with the login details for your server (Section 1.5). Be sure to include the colon and dot at the end.*
>
> *If you're using Windows, replace* `scp` *with* `pscp` *– see Section 7.4.*

You're asked for *loginpassword*, then the file is copied from your computer to your home folder on the server. This may take a minute or two.

> *You can use the same method in future whenever you need to copy a file (e.g. a logo) from your computer to the server.*
>
> *If you can't get scp or pscp to work with a hosted server, there will be another way to copy files to it. Consult the hosting provider.*

8.3 Install Drupal

Choose *drupaldbpassword* and make a note of it.

> *Drupal stores this password unencrypted in /var/www/drupal/sites/default/settings.php.*

Also choose and note *drupaladminpassword*. This must include at least one non-alphanumeric character, e.g. ! or &.

Log in to your server (locally or using ssh or KVM). At the server command prompt, type

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

Enter *dbrootpassword* (Section 1.5).

The *mysql>* prompt appears (Section 7.13). Type the following commands, waiting after each one for *mysql>* to reappear.

```
CREATE DATABASE drupal DEFAULT CHARACTER SET UTF8
COLLATE utf8_unicode_ci;
```

> *Words shown in capitals are SQL key words. You can type them in lower case – it's easier.*

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE
TEMPORARY TABLES,DROP,INDEX,ALTER ON drupal.* TO
drupaldbuser@localhost IDENTIFIED BY 'drupaldbpassword';
```

> *Remember, replace drupaldbpassword with the password you chose earlier. It needs to be in quotes as shown.*

```
FLUSH PRIVILEGES;
```

```
quit
```

The normal server command prompt appears. Type `sudo mc` and enter *loginpassword*.

> *If you'd already logged in as root (hosted server), leave out* `sudo`*. If you'd logged in as an ordinary user, this command opens mc as the root user so you can copy and edit system files and can issue commands without having to use* `sudo` *each time.*

Type `cd /var/www`

Press TAB to move the highlight to the right Midnight Commander pane and type `cd /home/`*username*.

You see the files in your home directory (not root's home directory) listed in the right pane. *drupalfilename.tar.gz* should be there.

Use the up and down arrow keys to highlight it, then press RETURN. Midnight Commander opens the compressed file and shows its contents in the pane. There should be a folder there called *drupal-X.XX* where *X.XX* is the version you downloaded (e.g. 7.14).

If you're copying an existing Drupal installation to a new server (Section 8.7), you'll should highlight and open the drupal.tgz file that you made on the old server and may need to open the nested var and www folders in it in turn to see the drupal folder. It's name won't include the -X.XX.

Highlight the folder and press F5. It's extracted and copied. This takes a few seconds, then it appears in the left hand pane (*/var/www*) too.

If you're copying an existing Drupal installation, skip the rest of this section.

Press TAB to move the highlight to the left pane. Highlight the *drupal-X.XX* folder and press F6. Type *drupal* as its new name (this replaces whatever was in the second line). Press RETURN. The folder is renamed simply as *drupal*.

Type the following commands:

```
cd drupal/sites/default

cp default.settings.php settings.php

chmod 666 settings.php

mkdir files

chmod 666 files
```

These permissions changes let Drupal itself make changes to its settings during initial setup. After these changes are complete, you'll change the permissions again so that the settings are protected from any hacking attempts.

8.4 Finish the installation

Before you start this step, you should set up the correct domain name. This makes sure Drupal sets itself up to use it. Even if you haven't registered the name yet, you can put it in your computer's hosts file. See Section 7.16.

Point your web browser to *http://domain/drupal*. The Drupal installation script appears.

> *Using domain (Section 1.5) to connect makes sure Drupal sets itself up correctly. Even if you haven't registered the name yet, you can put it in your computer's hosts file. See Section 7.16.*

The script is fairly self-explanatory. If there are problems at the *Verify requirements* stage, you may need to install additional PHP modules (Section 7.10), then click *proceed with the installation* at the bottom.

At the *Database configuration* step, set the *Database name* to *drupal*, the *Database username* to *drupaldbuser* and the *Database password* to *drupaldbpassword*.

When warned to remove write permissions on *sites/default* and *sites/default/settings.php*, go to the server command prompt and type

```
chmod 644 settings.php
```

> *If you followed this procedure, you won't have changed the permissions on /sites/default itself. They'll still be 755.*

In the SITE INFORMATION box, set the site name to what you want to appear at the top of each web page that Drupal generates. This might be the domain name but it could be something different. Can it be changed later???

The SITE MAINTENANCE ACCOUNT box sets up *drupaladminuser* and *drupaladminpassword*. Set the e-mail address so the administrator will receive and see any notifications about the site. Be sure to use a secure password.

8.5 Ensure secure logins

Log out from your new site, then go to *https://domain/drupal/?q=user*.

> *If, when you're setting up your Drupal site, you accidentally hide the normal login form from anyone not logged in (including yourself), you can log in using this address and fix the problem.*

You may see warnings about the site certificate (Section 7.11). Proceed anyway and confirm that you can log in as *drupaladminuser* with *drupaladminpassword*. This means you can install Drupal's *Secure Login* module without risking being locked out from your site.

Go to the server command prompt and type the following commands:

```
cd /var/www/drupal/sites
```

```
ls -l
```

The files and folders in the directory are listed in long format. The third column on each line shows the owner of the file or folder whose name is shown at the end of the line. This is *drupalfileowner*. Note it.

Type

```
chown www-data default
```

```
chown -R www-data all
```

Open a new tab in your web browser and go to drupal.org/project/securelogin. Right-click on the *tar.gz* download of the recommended version of the module and choose *Copy link address*, *Copy shortcut* or your web browser's equivalent.

Go to the tab where you're logged in to your Drupal site. Click on *Modules*, then + *Install new module*.

Click in the *Install from a URL* box and paste the address of the *tar.gz* file that you copied to the clipboard. Click Install.

After a few seconds, you should see a message that the module was installed successfully. Click *Enable newly added modules*. Scroll to the bottom of the page, tick the Enabled checkbox next to *Secure Login* and click Save configuration.

Log out of Drupal, then point your browser to *domain/drupal* without specifying *https://*. You should see the normal home page with the login boxes but the address bar should show the address starting with *https://*. This means that *Secure Login* is doing its job: logins are encrypted and users' passwords can't be sniffed on the network.

> *If you have a book on Drupal, it may suggest modules to install. You can install them now.*

The permission changes you made before installing *Secure Login* are needed to get the installation process to work but they make your site less secure. After you've installed all the modules you want, go to the server's command prompt again and type:

```
cd /var/www/drupal/sites
```

```
chown drupalfileowner default
```

```
chown -R drupalfileowner all
```

In future when you want to install other modules, you'll need to repeat the same *chown* commands before and after.

> *If you forget to do this, you'll see a very misleading message about FTP access when you try to install a module.*

8.6 Set up cron

While logged in to your Drupal site as *drupaladminuser*, click *Reports* (at the top), then *Status Report*. Look for the line that says *To run cron from outside the site, go to*. Highlight the line from *drupal* onwards and press CRTL+C to copy it to the clipboard.

> *You may be able to highlight most easily going from right to left.*

If you're not logged in to your server using ssh, do this and type sudo mc.

> *You need a root prompt and you may not be able to paste the key unless you're using ssh.*

Type the following commands:

```
sudo export EDITOR=nano
```

> *If you don't do this, you'll find yourself using the vile vi editor.*

```
sudo crontab –e
```

The nano editor starts and displays the system crontab. Use the down arrow key to move the cursor to a new line below any that are already in the crontab, then type

*0 * * * * wget -O - -q -t 1 http://localhost/*

> *Don't press RETURN. Note that -O is the capital letter O.*

Right-click with the mouse to paste the text you copied from the Status Report, then press RETURN.

The line should now read
*0 * * * * wget -O - -q -t 1 http://localhost/drupal/cron.php?cron_key=XXX*
where XXX is replaced with the long string of letters from the Status Report.

Press CTRL+X, then press Y to save the changes and RETURN to save to the file name that's shown.

> *Drupal has tasks such as checking for updates that it should carry out regularly. These are performed when a web browser is pointed to its cron.php file. For security, the cron.php file needs to see the correct key as displayed in the Status Report.*
>
> *Cron is a Unix service which can start a program on a regular schedule according to an entry in the crontab. wget is a program that can fetch a web page just as if you'd pointed your browser at it.*
>
> *You've added a crontab entry that makes wget fetch the page every hour on the hour.*

8.7 Move Drupal to a different server

> *You've set up a VirtualBox server and installed Drupal on it. You and others have spent ages adding modules and content and tweaking its behaviour and appearance. Now you're ready for the big time – you'll set up Drupal on a proper server. But what about all that work? You don't want to have to do it all again!*
>
> *The Drupal installation on the existing server consists of the various files (including added modules) in /var/www/drupal and the data in the MySQL moodle database. The installation can be copied to another server by copying the files and the database.*

Start by following this guide up to and including Chapter 7 to set up the new server. You can use new passwords or use the same ones as when you set up the old server.

> *It can be a different type of server (e.g. a WAMP server using Windows instead of Linux) so long as it runs similar versions of Apache, MySQL and PHP. However, the more differences in any of these components, the greater the risk of problems.*

Choose a nice quiet time to start transferring the server. On the old server, issue the following command:

```
/etc/init.d/apache2 stop
```

This prevents anyone accessing the old server and making changes that may not get carried over to the new server. ***Note that, because of delays in DNS propagation, some users might not be able to reach either server for a day or more.***

On the old server, issue the following commands:

```
mysqldump –u root –prootpassword drupal >drupal.sql
sudo tar –cz /var/www/drupal >drupal.tgz
```

The first command creates a sequence of SQL commands in drupal.sql that can be run to load a new database with the existing data. The second command saves all the files and folders in /var/www/drupal in a single archive file.

Copy the *drupal.sql* and *drupal.tgz* files to your home directory on the new server, e.g. using *scp* or *pscp*.

Follow Section 8.3, using the *drupal.tgz* you copied instead of downloading Drupal again. Be sure to use the same *drupaldbuser* and *drupaldbpassword* as when you set up the old server.

Log on to the new server, go to your home directory and type the command:

```
mysql –u root –pdbrootpassword drupal < drupal.sql
```

This loads the new database with the data from the old server.

Make any required changes to Apache on the new server so that the correct page will load (Chapter 10).

Set up cron (Section 8.6).

Put an entry in your hosts file pointing to the new server (Section 7.16) and use the web browser on your computer to confirm that the new server is working. You can now change the DNS record for your domain to point to the new server.

If you encounter a problem, e.g. because drupaldbpassword has changed, you can usually fix it by editing /var/www/drupal/sites/default/settings.php.

9 Install Moodle

Only follow this chapter if you want to use Moodle.

9.1 Placemarkers

The following are in addition to the list in Section 1.5:

moodle	This is the name of Moodle's MySQL database. It's easiest to use the word *moodle*.
moodleadminpassword	This is the password to administer the Moodle installation. Be sure it's strong and kept securely so the site doesn't get hacked.
moodlefilename.tgz	This is the name of the file that you download in Section 9.2.
moodledata	This is the name of a folder you'll create in Section 9.3. It's easiest to use *moodledata*.
moodledbpassword	You'll choose this password which Moodle uses to access its MySQL database. The password isn't stored securely so don't reuse an important one.
moodledbuser	This is the user name which Moodle will use to access its MySQL database. It's quite arbitrary: you can use *moodledbuser*.

9.2 Download Moodle

Moodle is available in the Ubuntu software repositories and you're supposed to be able to install it using `apt-get install moodle`. *Unfortunately, the version that's installed if you do this is an old one that's incompatible with the PHP version in Ubuntu 10.04. It doesn't work.*

By following this guide, you can install the latest version which does work correctly.

Go to moodle.org. Click on *Downloads*, then on *Moodle standard*

packages. Choose the version you want to download based on the recommendations on the page. Click to download and save the *.tgz* version.

> *Both .tgz and .zip versions are available but it's not clear which is which. If you move the mouse pointer over the icon for one of them, the file name usually appears at the bottom of the browser window. Choose the one that ends .tgz.*

The file is downloaded to your computer, usually to your Downloads folder. This may take a few minutes.

When it's complete, start the command line prompt for your own computer (Section 7.2 or 7.4) and type the following commands:

```
cd Downloads
```

`scp` *moodlefilename.tgz username@serveraddress*:.

> *Replace moodlefilename.tgz with the exact name of the file you downloaded. You can probably just type moodle, then press TAB to let your computer fill in the rest.*
>
> *Replace username@serveraddress with the login details for your server (Section 1.5). Be sure to include the colon and dot at the end.*
>
> *If you're using Windows, replace* `scp` *with* `pscp` *– see Section 7.4.*

You're asked for *loginpassword*, then the file is copied from your computer to your home folder on the server. This may take a minute or two.

> *You can use the same method in future whenever you need to copy a file (e.g. a logo) from your computer to the server.*
>
> *If you can't get scp or pscp to work with a hosted server, there will be another way to copy files to it. Consult the hosting provider.*

9.3 Install Moodle

Choose *moodledbpassword* and make a note of it.

Moodle stores this password unencrypted in /var/www/moodle/config.php.

Also choose and note *moodleadminpassword*.

Log in to your server (locally or using *ssh* or KVM). At the server command prompt, type

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

Enter *dbrootpassword* (Section 1.5).

The *mysql>* prompt appears (Section 7.13). Type the following commands, waiting after each one for *mysql>* to reappear.

```
CREATE DATABASE moodle DEFAULT CHARACTER SET UTF8
COLLATE utf8_unicode_ci;
```

Words shown in capitals are SQL key words. You can type them in lower case – it's easier.

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE
TEMPORARY TABLES,DROP,INDEX,ALTER ON moodle.* TO
moodledbuser@localhost IDENTIFIED BY 'moodledbpassword';
```

Remember, replace moodledbpassword with the password you chose earlier. It needs to be in quotes as shown.

```
FLUSH PRIVILEGES;
```

```
quit
```

The normal server command prompt appears. Type `sudo mc` and enter your password.

If you'd already logged in as root (hosted server), leave out sudo. If you'd logged in as an ordinary user, this command opens mc as the root user so you can copy and edit system files and can issue commands without having to use sudo again.

Type the following commands

```
cd /var
```

```
mkdir moodledata
```

```
chmod 777 moodledata
```

This creates a new directory (folder) called /var/moodledata and allows everyone (including Moodle) to write to it.

```
cd www
```

Press TAB to move the highlight to the right Midnight Commander pane and type `cd /home/`*username*.

You see the files in your home directory (not root's home directory) listed in the right pane. *moodlefilename.tgz* should be there.

Use the up and down arrow keys to highlight it, then press RETURN. Midnight Commander opens the compressed file and shows its contents in the pane. There should be a folder there called *moodle*.

If you're copying an existing Moodle installation to a new server (Section 9.6), you'll should highlight and open the moodle.tgz file that you made on the old server. You may need to open the nested var and www folders in it in turn to see the moodle folder.

Highlight the *moodle* folder and press F5. The folder is extracted and copied. This takes a few minutes, then it appears in the left hand pane (*/var/www*) too.

Press TAB so the highlight is in the left (*/var/www*) pane. Type `chmod -R 777 moodle`

This permission change lets Moodle itself make changes to its settings during initial setup. After these changes are complete, you'll change the permissions again so that the site is protected from hacking attempts.

9.4 Finish the installation

Point your web browser to *http://domain/moodle*. The Moodle installation script appears.

Using domain (Section 1.5) to connect makes sure Moodle sets itself up correctly. Even if you haven't registered the name yet, you can put it in your computer's hosts file. See Section 7.16.

Choose your language and click Next ».

Confirm that the Data directory is set to */var/moodledata*. Change it if

needed. Click Next ».

Confirm that the database driver is *Improved MySQL (native/mysqli)*. Click Next ».

Confirm that *Database host* is *localhost* and *Database name* is *moodle*. Set *Database user* to *moodledbuser* and *Database password* to *moodledbpassword*. Click Next ».

> *If you see a warning that Moodle can't write config.php, you missed the chmod -R 777 moodle earlier.*

Look at the very simple terms and conditions, then click Continue.

Moodle checks to see if your system meets its requirements. You might have to add more PHP extensions (Section 7.10), then reload the page and check again.

Once you see *Your server environment meets all minimum requirements* at the bottom of the page, click Continue.

Moodle is fully installed, this takes a few minutes: be patient. Once it's done, click Continue.

You're now asked for details for the Administrator's account. Fields that you must set are marked in red.

Enter *moodleadminpassword*.

> *Tick Unmask to make sure you've typed it correctly.*

Set the administrator's real name and email address.

> *In future the same person might want to have a second non-administrator account too. Unfortunately Moodle won't let them use the same email address again when they set this up.*

Set their *City/town* and *Country* and the *Timezone*. Make any other changes you want, then click Update profile.

> *You can change these settings later. While you're logged in, click My profile settings.*

Set up the site front page and choose whether to allow E-mail based self registration for additional users. Click Save changes.

You can change these settings later. While you're logged in, click Front page settings > Edit settings.

Moodle now starts with you logged in as administrator. Click *Logout* at the top right.

At the server's command prompt, type

```
chmod -R 755 /var/www/moodle
```

This helps to protect the Moodle installation from hackers.

Point your browser to *https://domain/moodle*. Click *Login* and log in as administrator.

This confirms that you can log in securely. You can force secure logins for everyone without locking yourself out.

You may see warnings about the site certificate (Section 7.11). Proceed anyway.

Click *Site administration* > *Security* > *HTTP security* on the left hand side. Tick *Use HTTPS for logins* and click Save changes.

9.5 Set up cron

Go to the command line prompt for the server and type the following commands:

```
export EDITOR=nano
```

If you don't do this, you'll find yourself using the vile vi editor.

```
crontab -e
```

The *nano* editor starts and displays the system crontab. Use the down arrow key to move the cursor to a new line below any that are already in the crontab, then type

**/15 * * * * php /var/www/moodle/admin/cli/cron.php*

Press RETURN. Press CTRL+X, then press Y to save the changes and RETURN to save to the file name that's shown.

Moodle has tasks such as checking for updates that it should carry out regularly. Cron is a Unix service which can start a program on a regular schedule according to an entry in the

crontab. You've added a crontab entry that runs Moodle's cron.php tasks every 15 minutes.

9.6 Move Moodle to a different server

You've set up a VirtualBox server and installed Moodle on it. Users have been using it and have added all sorts of information including courses. Now you're ready for the big time – you'll set up Moodle on a proper server. But what about all that work your users have done? They'll curse you if they have to do it all again!

The Moodle installation on the existing server consists of the various files in /var/www/moodle and the data in the moodle database. The database data includes everything that the users have added. The installation can be copied to another server by copying the files and the database.

Start by following this guide up to and including Chapter 7 to set up the new server. You can use new passwords or use the same ones as when you set up the old server.

It can be a different type of server (e.g. a WAMP server using Windows instead of Linux) so long as it runs similar versions of Apache, MySQL and PHP. However, the more differences in any of these components, the greater the risk of problems.

Choose a nice quiet time to start transferring the server. On the old server, issue the following command:

```
/etc/init.d/apache2 stop
```

This prevents users accessing the old server and making changes to the database that may not get carried over to the new server. ***Note that, because of delays in DNS propagation, some users might not be able to reach either server for a day or more.***

On the old server, issue the following commands:

```
mysqldump –u root –prootpassword moodle >moodle.sql
sudo tar –cz /var/www/moodle >moodle.tgz
```

The first command creates a sequence of SQL commands in moodle.sql that can be run to load a new database with the existing data. The second command saves all the files and folders in /var/www/moodle in a single archive file.

Copy the *moodle.sql* and *moodle.tgz* files to your home directory on the new server, e.g. using *scp* or *pscp*.

Follow Section 9.3, using the *moodle.tgz* you copied instead of downloading Moodle again. Be sure to use the same *moodledbuser* and *moodledbpassword* as when you set up the old server.

Log on to the new server, go to your home directory and type the command:

```
mysql -u root -prootpassword moodle < moodle.sql
```

This loads the new database with the data from the old server.

Make any required changes to Apache on the new server so that the correct page will load (Chapter 10).

Set up cron (Section 9.5).

Put an entry in your hosts file pointing to the new server (Section 7.16) and use the web browser on your computer to confirm that the new server is working. You can now change the DNS record for your domain to point to the new server.

If you encounter a problem, e.g. because moodledbpassword has changed, you can usually fix it by editing /var/www/moodle/config.php.

10 More Apache tweaks

If you point a web browser to *domain* alone (e.g. *example.com* rather than *example.com/drupal* or *example.com/moodle*), you'll still see Apache's default web page (Section 7.11). This isn't good at all.

The simple way to fix this is as follows:

Go to */etc/apache2/sites-available* in Midnight Commander (having started it with `sudo mc`). Highlight *default* and press F4 to edit it. Look for the line near the top saying *DocumentRoot /var/www* and change it to *DocumentRoot /var/www/drupal* or *DocumentRoot /var/www/moodle* as appropriate. Press F2 to save your changes and F10 to exit the editor.

Highlight *default-ssl* and make the same change to it, then type the command

```
/etc/init.d/apache2 reload
```

Users should now see the site you specified when they go to your domain alone but they can't specify it or a different CMS explicitly, E.g. if you install both Drupal and Moodle, then change *Document-Root* to */var/www/drupal*, neither *example.com/drupal* or *example.com/moodle* would work because Apache would look for */var/www/drupal/drupal* or */var/www/drupal/moodle* and wouldn't find it.

> *Because of this, you'll need to repeat the* `export EDITOR=nano` *and* `crontab -e` *steps in Section 8.6 or 9.5, remove /drupal or /moodle from the line that was added and press CTRL+X to save the modified crontab.*

The following method is more flexible but it's also more complicated and may be less secure so only use it if you need to:

Go to */etc/apache2/sites-available* in Midnight Commander (having started it with `sudo mc`). Highlight *default* and press F4 to edit it. Look for the line near the top saying *AllowOverride None* and change it to *AllowOverride All*. Do the same with the similar line slightly lower down relating to */var/www/*. Press F2 to save your changes and F10 to exit the editor.

Until you make these changes, Apache ignores .htaccess files.

Highlight *default-ssl* and make the same changes to it, then type the commands

```
/etc/init.d/apache2 reload
```

```
cd /var/www
```

```
touch .htaccess
```

> *If a file already exists,* `touch` *changes its modify time to the current time. If the file doesn't exist,* `touch` *creates it as an empty file.*

Highlight *.htaccess* in Midnight Commander and press F4 to edit it. Add the following lines:

RewriteEngine On
RewriteCond %{HTTP_HOST} ^(www.)?domain$
RewriteRule ^(/)?$ drupal [L]

Obviously, change *domain* (Section 1.5) and, if appropriate, change *drupal* to whichever CMS you want the user to see when they go to your domain alone.

Press F2 to save your changes and F10 to exit the editor. Now, users should see the site you specified when they go to *domain* alone. They can still specify it or a different CMS though, e.g. by going to *domain/moodle*.