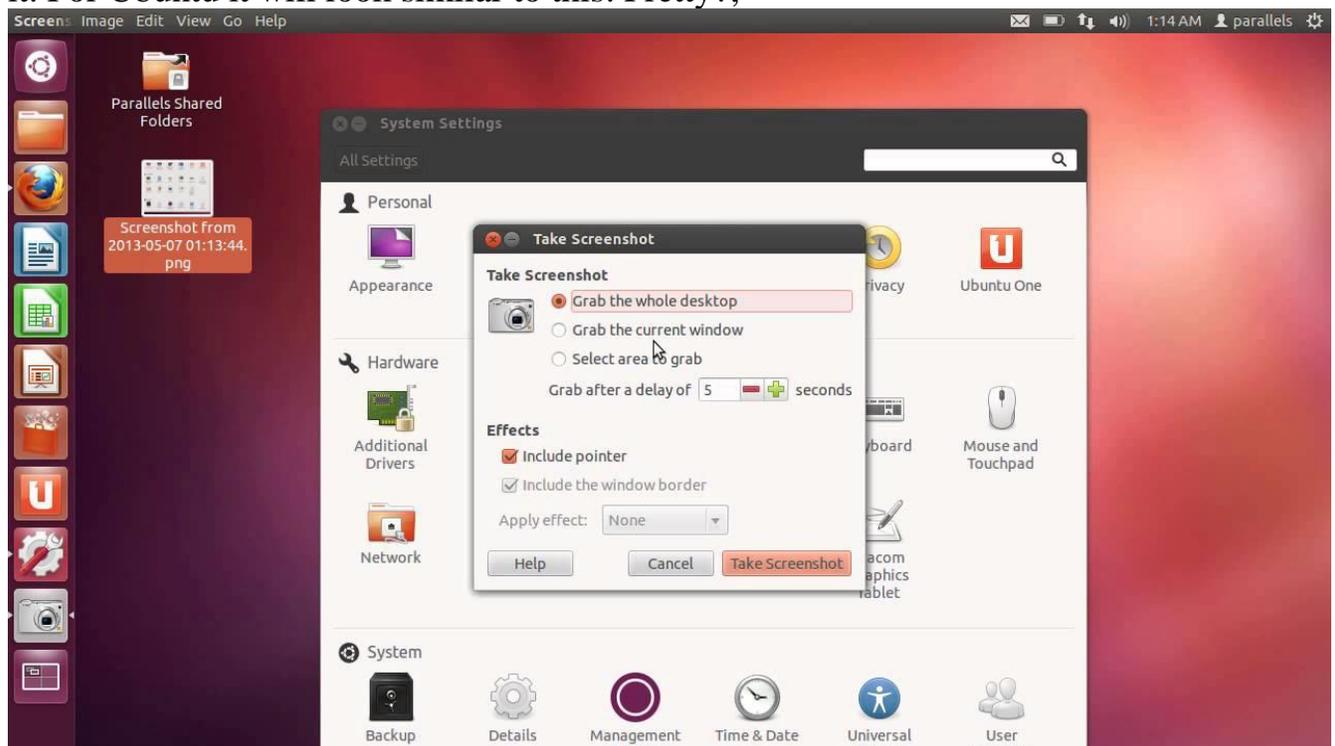


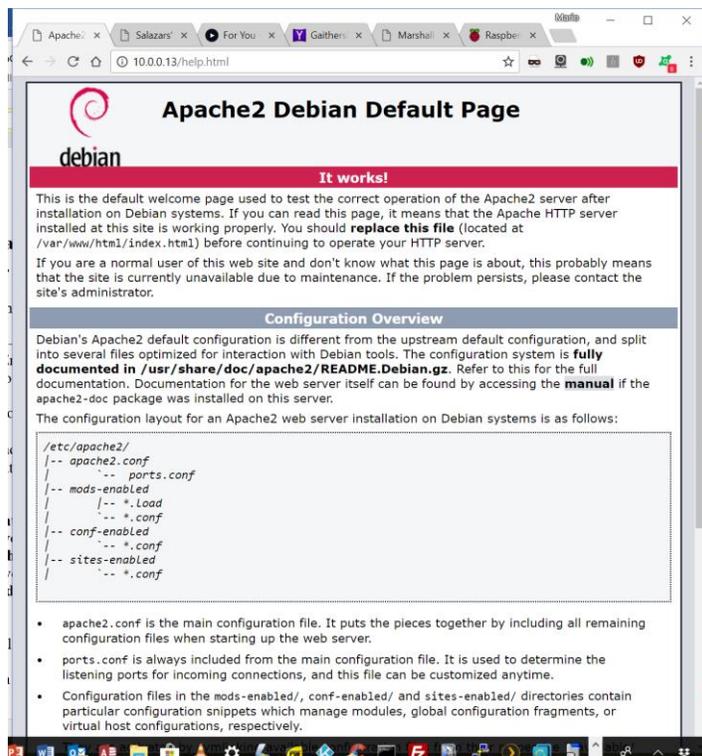
## Installing and configuring Apache 2 in Linux

Please note that dashes (-) are used to indicate what you should type, they should not be included in the command.

- Install Linux on an old desktop, dual core or better. I use Ubuntu. You can also install it in a single board computer like a Raspberry Pi;
- **PLEASE NOTE:** All the commands shown below can also be performed in the Ubuntu and most other Graphics User Interface (GUI). If you want to do it in this manner, please refer to the user manual for the specific distribution of Linux that you are using. However, if you follow the instructions below, you should be able to accomplish what you want and learn a lot on how to use the terminal commands in Linux. In Linux using terminal commands is much faster;
- Open your new Linux installation. It may take a few minutes the first time you do it. For Ubuntu it will look similar to this. Pretty?;



- Go to terminal (Ctrl-Alt-t or click black screen icon) and type `– sudo apt-get update` – press Enter, type your password if asked and press Enter, agree if prompted. This command updates the locations and files at those locations where you can get applications;
- When completed, type `- sudo apt-get install apache2*` - type your password if asked;
- Next go to a browser and type `– localhost` – in the address box;
- You should get a welcome screen from apache. It tells you that your installation worked;

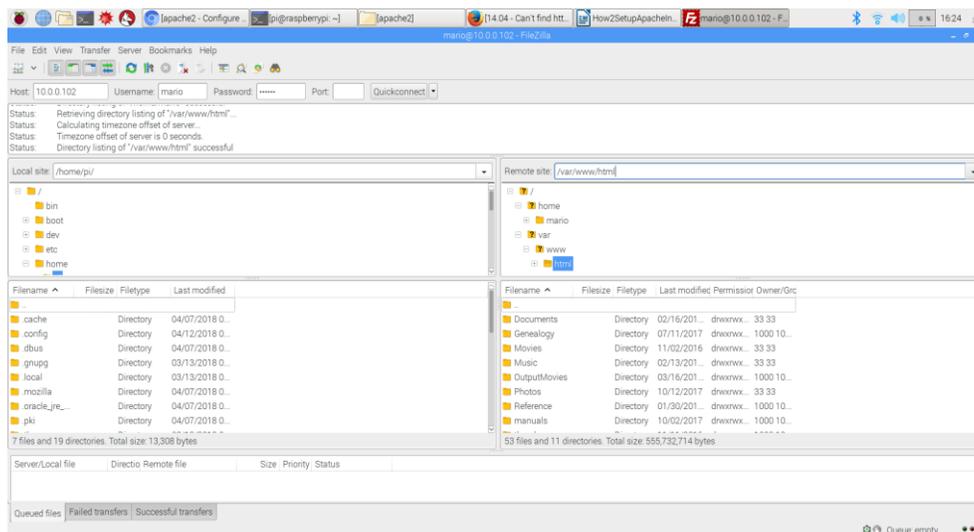


**Your newly created server is in the directory `/var/www/html/`.**

**Now you must get programs that will allow you to work with your web site remotely. You need a FTP server and a SSH server programs. The former allows you to upload files to your server and the latter to interact with your server remotely (headless), meaning you can use a remote computer and use the terminal in your server computer as if you were there.**

- Get a FTP server for your site. I use VSFTPD. On the terminal type – `sudo apt-get install vsftpd -;`
- Get a SSH server for your site. OpenSSH is a good choice. On the terminal type – `sudo apt-get install openssh-server openssh-client -.` If you are already using Putty, you can continue using it as a SSH client;
- Now you have to configure VSFTPD. The configuration file is in `/etc/vsftpd.config`. For that you need to open this file with a text editor in the terminal. I use nano as it is fairly easy to use and available in most Linux distros. Type – `sudo nano /etc/vsftpd.config -`, look for the line (use Ctrl-w to search):
  - `#Uncomment the second line to enable any form of FTP write command.`  
`#write_enable=YES`
  - Uncomment the second line by erasing the # so that it looks like `write_enable=YES;`
- Press Ctrl-o and Enter to save the file with your changes, press Ctrl-x to exit;

- Linux and Apache are very secure, so you must be authorized to copy files, view them or execute them. This is handled with “Permissions”. You must give yourself permission to upload and revise files in your server;
- In the terminal type – cd /var/www – and press Enter to go to the files that you need to change permissions.
- Type – sudo chmod -R 777 html – and press Enter. This will change all the files’ permissions so that anyone can read them, write to them and execute them. Don’t worry you have to login with a password to do it. Type your password and Enter to continue;
- In your Windows computer download FileZilla. This is a great program that allows you to upload files to your server, delete, rename and change permissions;
- Now you have to change the default page for your site to something more useful. What I recommend is to create a page with your current word-processor in Windows and save it as a html document with the name– index.html -. Later you can think about installing a WYSIWY (What you see is what you get) web site editor. The easiest to use that is free is Kompozer;
- In the computer where you hosted your site, go to the terminal and type – ifconfig – and press Enter. If you are connected to your network, which you have to be to be able to download files as instructed above, you should see *inet* and then a number in the form ###.###.###.### for example 10.0.0.13 or 192.68.1.25, depending on your router. This number is the *Internal IP* for your site;
- Using FileZilla connect to your site with your Windows computer. To do that you need to know the *Internal IP* for your site
- The image below shows such connection. Please note that FileZilla will connect you to the computer, not directly to your site. You must change the box titled *Remote site* to - /var/www/html – the directory where your site files are located;



- You can alternatively create a permanent site in the *Site Manager* under File or pressing Ctrl-s. Click new and add all the information. Click on *New site*, add information and click on the *Advanced* tab and changed the *Remote Default Directory* to - /var/www/html -. You can then connect by going to *Site Manager*, Ctrl-s, and selecting *Connect*;
- As you *may have noticed* the left panel in *FileZilla* is your Windows computer and the right side is the computer where your site is hosted. These are known as *Local* and *Remote*;
- It is a good idea to save the original *index.html* file as reference. Right click the *index.html* file on the right/remote and select *Rename*. Rename it as *help.html*;
- Grab *index.html* from the Local side and drag it to the directory *html* on the right side or *Remote*;
- Right click the new *index.html* and change *Permissions* to 777. Do the same with *help.html*;
- For the time being you are done with *FileZilla*;
- From your Windows computer type the *Internal IP* of your server in the address box of your browser to see your new site's main file (*index.html*);
- You can now add, delete or change files in your server using the same procedure in *FileZilla*.
- If you want to access your files remotely (from any browser in the world), you would have to forward three ports as indicated below. Once you do that, you should be able to access your site from anywhere by just typing your [\*External IP\*](#). Your Internet Service Provider assigns this address dynamically, so it may change with time.

## Other concerns

### Security

Apache has a file *.htaccess* (note the period preceding) in the *html* directory that can be configured to require a password for everyone trying to access your site. Go to <http://www.htaccess-guide.com/> to figure out how to configure and use.

### Accessing your server from remote locations

To do this you have to *forward* (open) some ports in your router. If you know how to *forward ports*, make sure you forward three ports for: *html*, *ftp* and *ssh* access. The default ports are 80, 21 and 22 respectively. If you don't, go to [https://www.pcworld.com/article/244314/how\\_to\\_forward\\_ports\\_on\\_your\\_router.html](https://www.pcworld.com/article/244314/how_to_forward_ports_on_your_router.html).

### Caveat emptor

Anything that goes on the Internet can be accessed by someone with sufficient knowledge and motive to do so. Don't give them the knowledge or motive.

Opening ports in your router could be used by people with nefarious intent to do you harm. There is always a calculated risk when you do something like this. For most of us this risk is minimal.