

Streamlining your Pi

based on an article by Kenneth A Spencer
(Setting up a Web Server using Raspbian Wheezy on the Raspberry Pi)

Start by changing the password for user pi /password raspberry.

From the command line type:

```
passwd pi
set new password (t ex jordgubbar)
```

Set a root password

Once the machine is rebooted after stage A2, execute the command:

```
sudo passwd
```

and enter your preferred root password (t ex godis). As the installations and reconfigurations are best carried out with root privilege you should then issue the command:

```
sudo su
```

which is the easiest way to change to the privilege level required for the rest of your session.

Exit this mode by typing

```
exit
```

Back to old user!

Execute the following command to obtain and install the proftpd ftp sever:

```
sudo apt-get install proftpd
```

You will use proftpd later on to transfer your website files to the newly configured web server. standalone

Run Software Updates

```
sudo dpkg-reconfigure tzdata
sudo apt-get update
sudo apt-get upgrade
```

takes some time and you have to press yes a couple of times.

Change hostname (name of computer)

Edit /etc/hostname as follows:

```
cd /etc
```

```
sudo nano hostname
```

nano is a Linux editor. Change the line rpi1 to your chosen name for your server t ex rpi2.

Then save the file with [Ctrl]+O and exit nano with [Ctrl]+X.

Also check the file /etc/hosts and replace the current name with the name you want.

Finish the change with

```
sudo /etc/init.d/hostname.sh
```

```
sudo reboot
```

Network address is only occurring in file /etc/network/interfaces

Optional install Gnome partition editor, gparted

```
sudo apt-get install gparted
```

Setting up Apache2

Then execute the following command to obtain and install the Apache2 web server:

```
apt-get install apache2
```

Apache2 will be running after installation, but if you need to start or stop it, the commands are:

```
service apache2 start
```

```
service apache2 stop
```

A5.1 After the initial installation, Apache2 may report the following warning when started or stopped:

Could not determine the server's fully qualified domain name, using 127.0.0.1 for ServerName

A5.1a Edit `/etc/hostname` as follows:

```
cd /etc
```

```
nano hostname
```

nano is a Linux editor. Change the line `raspberrypi` to your chosen name for your server.

Mine is `rpi2`. This is essential anyway if you have more than one Raspberry Pi.

Then save the file with `[Ctrl]+O` and exit nano with `[Ctrl]+X`.

A5.1b Then edit `/etc/hosts` as follows:

```
nano hosts
```

Add the lines (the first line may already present but commented out with a `#` symbol):

```
127.0.0.1 localhost
```

```
192.168.0.202 localhost
```

and, if you wish, also add the following line to the `hosts` file:

```
192.168.0.202 rpi2
```

In the `/etc/hosts` file, `127.0.0.1` is the standard IP address and `localhost` the standard TC/IP hostname for local reference to the current physical machine.

Replace `192.168.0.202` by your actual IP address noted earlier, and `rpi2` by your chosen name for your Raspberry Pi machine.

Then save the file with `[Ctrl]+O` and exit nano with `[Ctrl]+X`.

A5.1c Next, create an extra file in the `/etc/apache2/conf.d` directory as follows:

```
cd /etc/apache2/conf.d
```

```
nano servername.conf
```

which will open the nano editor with a new empty file called `servername.conf`.

Type your `ServerName` into the file under the `ServerName` directive:

```
ServerName rpi2
```

Change the name to that chosen for your own machine.

Then save the file `[Ctrl]+O` and exit nano with `[Ctrl]+X.sudo nano`

As Apache2 loads data from the files in `conf.d` into its configuration, these changes will remove the error concerning Apache2 not finding your server name.

A6. Install the PHP programming environment

install the PHP programming environment version 5:

```
apt-get install php5 libapache2-mod-php5 php5-intl php5-mcrypt
php5-curl php5-gd php5-sqlite [NB: all on a single line]
```

Test and get PHP information

If you wish, you can create a simple web page to show the PHP configuration information:

```
cd /var/www
```

to get to the directory containing your web page. Then open `nano` with an empty file:

```
sudo nano phpinformation.php
```

When it opens, type in the following:

```
<?php
phpinfo();
?>
```

Then save the file with `[Ctrl]+O` and exit `nano` with `[Ctrl]+X`.

View your new page by typing `http://piserver/phpinformation.php` into your browser.

A7. Install the MySQL database management system

Execute the following command to obtain and install the MySQL database management system:

```
apt-get install mysql-server mysql-client php5-mysql
```

During the installation, you will be invited to enter a password for the top level MySQL user. The username is commonly `root`, but the password is up to you. Note that this is the managing user for MySQL and is not related to the `root` user of your Raspbian Wheezy operating system.

Then to take account of system and configuration changes, stop and start the Apache2 web server.

You can test your MySQL installation by logging in to MySQL. You will have to use the MySQL `root` user for the moment:

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

The `-u` parameter signifies that the next item is a username, and the `-p` parameter tells MySQL to expect a password. When prompted, enter the `root` password created earlier, and MySQL will respond with an Oracle copyright notice and the `mysql>` prompt. Exit MySQL by typing `exit`.

It is very much easier to manage a database, including the creation of tables and other database objects in MySQL using phpMyAdmin, and we will install that most useful tool very shortly. But we have two or three more steps to complete first.

After this make a reboot and disconnect/reconnect putty

A8. Install the APC support package for PHP

ACP is an alternative caching support system for PHP, which enables PHP intermediate code to be cached. This can improve the performance of Apache2 and other programs which may execute PHP code.

Execute the following command to obtain and install the APC support system for PHP:

```
sudo su
```

```
apt-get install php-pear php5-dev apache2-prefork-dev build-essential make && pecl install apc [NB: all on a single line]
```

Then edit the `php.ini` configuration file:

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

and add the following text to the file, in the `Dynamic Extension` section, some way down the file:

```
extension=apc.so
```

Then to take account of system and configuration changes, stop and start the Apache2 web server.

```
service apache2 stop
service apache2 start
```

A9. Install some additional items

There are a few additional items which may be usefully installed before we install the next major item.

Execute the following command to obtain and install them:

```
apt-get install nmap
```

It installs some additional fonts, the `nmap` network analysis tool some other utilities. The `nmap` program has a GUI interface, (on your Pi only) called `zenmap` which can be installed by:

```
apt-get install zenmap.
```

A10. Install phpMyAdmin

`phpMyAdmin` is one of the most useful tools for working with the `MySQL` database management system. It is a web-based control panel, but it is not only that. It also includes support for creating and amending your database tables, for entering data into them, and for designing and executing queries.

Execute the following command to obtain and install `phpMyAdmin`:

```
apt-get install libapache2-mod-auth-mysql php5-mysql phpmyadmin
```

When asked if `phpMyAdmin` should configure a `MySQL` database for itself, select `yes`.

When asked, enter the password of the `root MySQL` user, so as to authenticate yourself on `MySQL`.

Once authenticated on `MySQL` you will be asked to create and confirm a password for `phpMyAdmin`.

Then you'll be asked which server on which to install it: select `Apache2`.

During the installation, you will be offered several options: accept the default suggestions at this stage of your knowledge! After further processing, the `phpMyAdmin` installation will finish.

There are two adjustments to the configurations necessary before you can use `phpMyAdmin`.

First edit the `php.ini` file to include a `MySQL` library:

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

Then type the following text into the `Dynamic Extensions` section about two thirds down the page:

```
Shellextension=mysql.so
```

As things stand, there is no provision for showing the `phpMyAdmin` pages from your default web page directory. To such an create an entry make a symbolic link in the `Apache2` data directory, as follows:

```
cd /var/www
ln -s /usr/share/phpmyadmin
```

Now from a workstation, you'll be able to navigate to:

```
http://raspbianwheezy97/phpmyadmin
```

or directly on your Pi to:

```
http://localhost/phpmyadmin
```

If logging in from a workstation, remember to substitute your own IP address or if you have entered it into your DNS server or the workstation's `hosts` file, you may enter your `hostname`.

When invited to login on the `phpMyAdmin` opening page, respond with the username `root` and the password which you created during `phpMyAdmin` installation.

A11. Install the Webmin administration Tool

`Webmin` is an extremely useful tool for administering all aspects of a Linux server.

Before installing `Webmin`, it is necessary to edit the `sources.list` file:

```
nano /etc/apt/sources.list
```

Add the following two lines:

```
deb http://download.webmin.com/download/repository sarge contrib
deb http://webmin.mirror.somersettechsolutions.co.uk/repository
sarge contrib [NB: the second entry must be all on a single line]
```

Then you must obtain and install the GPG key with which the repository holding Webmin is signed:

```
cd /root
wget http://www.webmin.com/jcameron-key.asc
apt-key add jcameron-key.asc
```

Then get any system updates, following which you can obtain and install the Webmin package(s):

```
apt-get update
apt-get install webmin
```

It may take some time, but all dependencies should be resolved automatically.

Webmin runs its own secure sockets web server, separate from Apache2. It does this using its own port number (10000) which must be added to the Webmin address when typed into your web browser address bar: `https://10.0.0.97:10000/`. The `https://` in the address bar indicates that the web browser is asking for the page using secure sockets layer, which is a more secure protocol than the usual `http://`. Of course, you can also use the form `https://yourPiHostname:10000` for the Webmin address if you wish and if you have your Pi details in the workstation `hosts` file or in a DNS server..

When invited to login, you will normally respond as user `root` with your `root` password, or as any user who can use `sudo` to run commands as `root`.

Webmin install complete.

You can now login to <https://rpi2:10000/> to use Webmin as `root` with your `root` password, or as any user who can use `sudo` to run commands as `root`.

hit har jag gjort nu. Dags för backup av SD kortet Spencer A

Sen kommer webalizer etc Inte inlagt ännu

A12. Install the Webalizer Graphical Webstats package

By default, the Apache2 web server generates a line of data in a log file for each request it receives to serve a webpage. This data consists of a series of fields containing such items as the IP address of the client requesting the page, the date and time, the filename of the page, the browser and operating system, and more. You can read this data into a database for analysis, but it can also be converted into a graphical display which is useful for less formal study of the use of your web server activity.

You can obtain and install Webalizer by executing the following command:

```
apt-get install webalizer
```

The default configuration of Webalizer is not quite correct for Apache2 under Raspbian Wheezy. Therefore you must edit the Webalizer configuration file

```
/etc/webalizer/webalizer.conf
nano /etc/webalizer/webalizer.conf
```

In the `webalizer.conf` file, for the setup of Apache2 so far, you may need to locate and edit the Apache2 log file entry to:

```
logfile /var/log/apache2/access.log
```

Later, if you decide to implement multiple virtual hosts on Apache2 this entry will need further amendment.

In addition, it appears that `Webalizer` does not install the default geographical IP locator database, but instead installs `GeoIP`. Therefore edit the `webalizer.conf` file further:

Uncomment/edit the lines:

```
GeoIP yes
GeoIPDatabase /usr/share/GeoIP/GeoIP.dat
```

Back up the SD-card.

If you have already developed website, you can copy it onto your Raspberry Pi. To do this, follow this procedure:

Use an FTP Client such as `WS_FTP95` or similar, to connect to your Raspberry Pi..

You will need to enter the IP address of your Pi into your FTP client.

You will have to extend the default settings of the `ProFTPD` server on your Raspberry Pi to allow `root` login.

NB: if you have extended the `root` login to `ProFTPD` for a session, it would be wise to revert afterwards to the standard settings, which do not permit `root` to login via FTP remotely.

Allowing root log-in in ProFTPD

To permit `root` login to `ProFTPD` open `PuTTY` and login to your Pi as `root`. Then open the `ProFTPD` configuration file in the `nano` editor:

```
nano /etc/proftpd/proftpd.conf
```

For tidiness, first of all set the `ServerName` directive to your own host name for your Pi

```
ServerName "yourPiHostname" (set rpi2)
```

Then locate the `RootLogin` directive and change it to "On", or create the line at the end of the file:

```
RootLogin On
```

Then save the file `[Ctrl]+O` and exit `nano` with `[Ctrl]+X`.

Finally, you need to make a change to the `ftpusers` file, which contains a list of usernames prohibited from logging in via an FTP client:

```
nano /etc/ftpusers
```

The `root` user is commonly at the top of the file. Simply insert a hash symbol (`#`) at the very beginning of that line to remove the root user from the prohibited list.

Then save the file `[Ctrl]+O` and exit `nano` with `[Ctrl]+X`.

You will need to stop and restart the `ProFTPD` service in order for the changes to take effect:

```
service proftpd stop
service proftpd start
```

C1. Copy your Website Files

Login to your FTP client program with root privileges and navigate to your Apache2 data directory, usually `/var/www`. You must then create a directory for your website - we will name our site `oursite1`. Then navigate to the location where you hold your website files and content and start the process of transfer of all of your website files to your chosen site directory on your Raspberry Pi `/var/www/oursite1`.

You will now be able to view your website from your PC by starting a web browser and typing the following into the address bar:

http://10.0.0.97/oursite1

or *http://raspbianwheezy97/oursite1*

but remember to substitute my Pi IP address for yours, and my Pi hostname for yours, and to enter your Pi hostname into your Windows host file.

Your site is now available, but there are two further enhancements we can make to your setup: use a virtual hostname and make to site available outside your network - i.e. go global!

Hit har jag gjort

C2. Create a Virtual Host

First, let's stop the Apache2 service – `service apache2 stop`

Next, we need to navigate to the location from where Apache2 will read a new configuration file containing our setup for a virtual host.

```
cd /etc/apache2/sites-available
```

You can place all virtual host configuration files here. But each file will not be enabled unless a symbolic link to the file is created in the parallel directory `sites-enabled`. There is a tool for doing this, which we will use once we have created the file.

We will call our configuration file `oursites.conf`, and we will create it with the nano editor:

```
nano oursites.conf
```

Note that the `oursites` is plural, as we may have more than one virtual site! Then edit the file so that it contains the following text:

```
# oursites.conf
# KA Spencer 201211
#
# First, the listening port (if not specified elsewhere):
Listen 80
#
# Next, the IP address and port for the virtual host. This
assumes
# that you have only one IP address and port for this server.
# Be sure to substitute your own parameters throughout this file!
NameVirtualHost 10.0.0.97:80
```

```
# NB
```

```
# The apache2.conf - file contains a statement include ports.conf
# The ports.conf -file contains a statement NameVirtualHost *:80
```

```
# This will create a problem if your definition file for virtual
# hosts also contains a statement NameVirtualHost...
# Comment away the include ports.conf statement in the
# apache2.conf-file
```

```
# Next, add the default server, because creating a virtual host
# causes Apache2 to ignore the default server configured in the
# /etc/apache2/sites-available/default file.
# If you do not do this, any html files in /var/www will be
ignored!
```

```
<VirtualHost 10.0.0.97:80>
```

```
DocumentRoot /var/www
```

```
DirectoryIndex index.htm index.html index.php
```

```

</VirtualHost>
#
# Next your first virtual server details:
<VirtualHost 10.0.0.97:80>
ServerAdmin email@youraddress.com
ServerName oursitel
DocumentRoot /var/www/oursitel
DirectoryIndex index.htm index.html index.php
ErrorLog /var/www/oursitel/log/error.log
CustomLog /var/www/oursitel/log/access.log combined
</VirtualHost>
# This will allow you to access your "oursitel" website by that
name.

```

For the CustomLog directive, choose a file format defined in /etc/apache2/apache2.conf.

There are one or two other configuration adjustments to make before this will work.

First, create a directory for the Apache2 log files, as given in the oursites.conf:

```
mkdir /var/www/oursitel/log
```

Then we must make a link to our new virtual host file in sites-enabled. Do this using the a2ensite command:

```
a2ensite oursites.conf
```

This program creates the link for you. Its sister program a2dissite will remove the link if you wish to take your virtual host site offline.

OBS använd a2dissite default för att ta bort 000-default från etc/apache2/sites-enabled om du lägger in virtual hosts med en definitionsfil. Annars blir det dubbelt.

Finally, it is necessary to enter your new virtual host site name into your Raspberry Pi /etc/hosts file (see A5.1b), your PC workstation hosts file (see A5.2), or into your domain name server (DNS) if you have one.

Now if all is well, you can restart the Apache2 server in the usual way. There should be no errors on startup. If errors occur, stop the Apache2 server and examine each stage in turn for errors.

If everything is correct, when you type `http://oursitel` into your web browser address bar, you should see your website.

You have a virtual host working! Wouldn't it be good if the whole world could see you site - that's possible, let's see how.

Failed

```

root@rpi2:/etc/apache2/sites-available# cd /var/www/
root@rpi2:/var/www# ls
index.html phpinformation.php phpmyadmin taki
root@rpi2:/var/www# mv /var/www/taki /var/www/oursitel
root@rpi2:/var/www# ls
index.html oursitel phpinformation.php phpmyadmin
root@rpi2:/var/www# mkdir /var/www/oursitel/log
root@rpi2:/var/www# a2ensite oursites.conf
Enabling site oursites.conf.

```

To activate the new configuration, you need to run:


```
service apache2 reload
root@rpi2:/var/www# nano /etc/hosts
root@rpi2:/var/www# service apache2 reload
[ ok ] Reloading web server config: apache2 not running.
root@rpi2:/var/www# service apache2 start
[...] Starting web server: apache2(98)Address already in use: make_sock: could not bind to
address 0.0.0.0:80
no listening sockets available, shutting down
Unable to open logs
Action 'start' failed.
The Apache error log may have more information.
failed!
root@rpi2:/var/www#
```