

How to Make a Raspberry Pi Web Server

If you are looking for a way to make a cheap web server, primarily to be used as a testing environment or to store files, then the Raspberry Pi is perfect for you. What is a Raspberry Pi? It is an inexpensive mini computer, perfect for...

Part 1 of 7:

Starting the Raspberry Pi Operating System

1. **Obtain a copy of the Raspberry Pi OS (Operating System) from the link in the sources section.** There are a few different distributions available, but for this article, we will be using the "Raspbian" version.
2. **Extract the image to the SD card.** To do this, we need a tool called the Win32 Disc Imager. A link for this is also available in the sources section. Now open the tool, navigate to the drive letter that your SD card is in, select the location of the Raspberry Pi OS image, and select burn. Wait for it to finish.
3. **Open the SD card in Windows Explorer.** Just create a file with the name `ssh`. This is a security update introduced since Raspbian Jessie.
4. **Eject the SD card, and place it into your Raspberry Pi, then plug in the rest of the cords, being sure to plug in the mini USB last.**
5. **Log in once the operating system loads up.** The default username is "pi", and the default password is "raspberry".
6. **Start by changing the password.** From the command line type:

```
passwd pi
```

7. **Enter your new password and then confirm it.** Please note, the cursor will not move when typing passwords but you are entering text.

Part 2 of 7:

Running Software Updates

1. **Get started with updates.** Because you are running a fresh version of Debian, you will need to do some housecleaning, updating, and installing. First, we are going to update the clock, update our sources, then upgrade any pre-installed packages. Type the following at the command line (press return/enter after each line):

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

2. **Set the date and time.** From the command line type (replace parts as necessary):

```
sudo date --set="30 December 2013 10:00:00"
```

Part 3 of 7:

Keeping the Firmware Up To Date

1. **Install Hexxeh's RPI update tool to help keep Raspberry Pi up to date.** To do this, run the following commands (press return/enter after each line):

```
sudo apt-get install ca-certificates sudo apt-get install git-core sudo wget  
&&  
sudo chmod +x /usr/bin/rpi-update sudo rpi-update sudo shutdown -r now
```

Part 4 of 7:

Set Up SSH

1. **Set up SSH so that we can do everything else from a different computer.** To do this, first note the I.P. address of the Raspberry Pi:

```
ifconfig
```

1. You should see something like this:

```
eth0 Link encap:Ethernet HWaddr fe:fd:45:xx:xx:xx inet addr:69.164.xxx.xxx B  
(11.3 GB) TX bytes:33179556297 (33.1 GB)  
Interrupt:76 Make sure you are connected to internet!! The tag eth0 will ap  
if  
you have ethernet plugged in. Wireless might be wlan0, but could differ.
```

1. On the second line "inet addr:69.164.xxx.xxx" is the IP address of your Raspberry Pi.
2. **Enable SSH and reboot (press return/enter after each line):**

```
After noting the inet addr use: sudo /etc/init.d/ssh start for every  
time you boot up the pi. Hint: If an error occurs, use the command  
below, and then the command  
up above. sudo apt-get install ssh Then, restart your pi: sudo shutdown -r
```

3. **Unplug the cords for your USB keyboard and your monitor.** These are no longer necessary, as everything else will be done over SSH.
4. **Download an SSH client like PuTTY (www.putty.org) which can be downloaded for free from Google and connect to the IP address of your Raspberry Pi logging in with the username "pi" and the password you set earlier.**

Part 5 of 7:

Installing the Web Server

1. **Install Apache and PHP.** To do this, execute the following commands:

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```

2. **Restart the service:**

```
sudo service apache2 restart
```

OR

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

3. **Enter the I.P.** address of your Raspberry Pi into your web browser. You should see a simple page that says "It Works!"

Part 6 of 7:

Installing MySQL

1. **Install MySQL.** To do this, install a few packages with the following command:

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

Part 7 of 7:

Installing FTP

1. **Install FTP to allow transferring files to and from your Raspberry Pi.**

2. **Take ownership of the web root:**

```
sudo chown -R pi /var/www
```

3. **Install vsftpd:**

```
sudo apt-get install vsftpd
```

4. **Edit your vsftpd.conf file:**

```
sudo nano /etc/vsftpd.conf
```

5. **Make the following changes:**

1. **anonymous_enable=YES to anonymous_enable=NO**

2. **Uncomment local_enable=YES and write_enable=YES** by deleting the # symbol in front of each line

3. then go to the bottom of the file and add **force_dot_files=YES.**

6. **Save and exit the file by pressing CTRL-O, CTRL-X.**

7. **Restart vsftpd:**

```
sudo service vsftpd restart
```

8. Create a shortcut from the Pi user's home folder to /var/www:

```
ln -s /var/www/ ~/www
```

9. You can now FTP using the Pi user and access the /var/www folder via a shortcut that should appear on login.

You finished reading the article "**How to Make a Raspberry Pi Web Server**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.