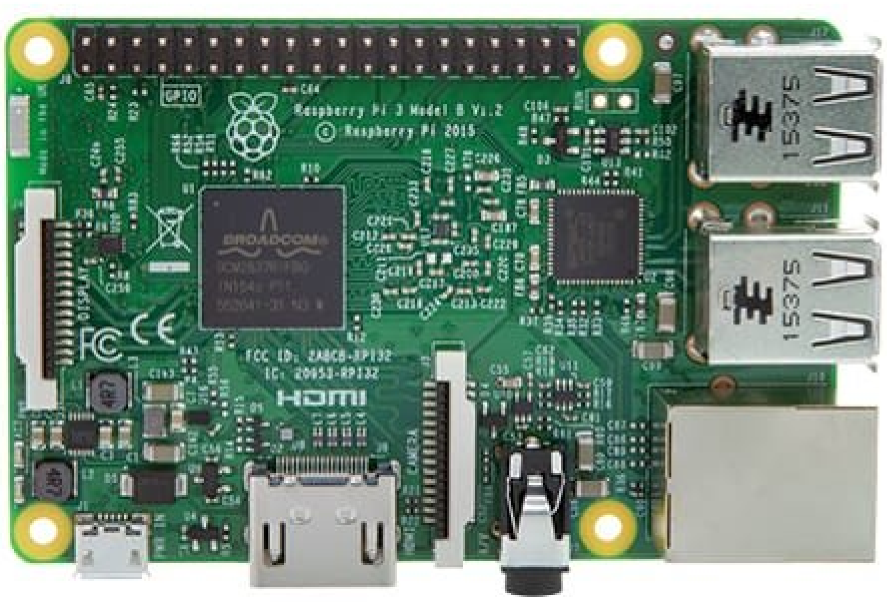
Raspberry Pi Dev Setup with Nginx + PHP7

How to develop Grav sites with a $35 computer

Andy Miller
posted on 04/04/2016 in tutorial + raspberrypi + nginx+ php7 12 mins

Last year I setup a Raspberry Pi 2 Model B as a development web server. I took note of the steps involved and over the ensuing months, I've provided those notes to individuals on our Gitter chat looking to do the same thing. I recently purchased the latest Raspberry Pi 3 Model B which has the same form-factor, but has a faster 1.2Ghz quad core processor, built-in WIFI, and Bluetooth 4.1. I thought I would take this opportunity to update my notes, and turn them into a full blown tutorial as this seems to be a popular subject.



In this tutorial, we cover the basics to get Raspbian OS running on your Raspberry Pi computer. We will install the high-performance nginx 1.9 webserver, along with PHP 7.0 for optimal performance. We'll also cover installing the latestNetatalk 3 with spotlight support for easy file sharing with your Mac. So read on dear listener if you would like to discover the joy of building your own full linux-powered server on a $35 computer!

Initial Raspbian Installation

Raspbian is the best option for operating systems on the Raspberry Pi because it's the most supported and is based on Debian, a very popular Linux distribution. Those of you that have used Debian or Ubuntu will be very much at home with Raspbian. The current version called Jessie is equivalent to the latest Debian 8 release.

Rather than rehashing all the information available, I suggest you follow the installation procedures outlined on the Raspberry Pi site and install NOOBS onto an SD card. After you have created your bootable NOOBS installer, you can insert it in your Raspberry Pi and power it on. It will boot very quickly and then you can simply pick Raspbian from the installer and let it do it's thing.

When the installation is complete, you should be greeted by a clean desktop environment. Time to get some things setup and configured!

Raspbian Updates

Throughout this tutorial, I'm going to assume you are using aterminal so if you have not already done so, start up a terminal by clicking the terminal icon from the top menu bar of the desktop and it will launch a terminal instance for you.

Even though you just installed Raspbian, there are often updates to packages available so the first thing to do is make sure you are running the latest versions. Follow these steps:

```
$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get upgrade
```

You probably won't have any extra updates to perform after the `dist-upgrade` step, but better safe than sorry!

Raspbian Configuration

We can use the RASPI Config command to setup some important configuration options for our new Pi.

```
$ sudo raspi-config
```

Read the official docs for full details, but I like to set the following:

- Change Password - pick a new unique password (default is `raspberry`)
- Boot Option - I like to select `console` as I don't use the desktop at all
- Internationalisation Options - Change Locale to suite (I use `en_US.UTF-8`), Change Timezone to `UTC` (why?)

Choose your Editor

By default Raspbian comes with the GNU Nano Editor. It's a very capable editor and is great for newbies because it has on-screen help for commands. However, I've been a Vi kind of guy for many years, and just feel more comfortable using that. Vi has been around for years, and Vim is a new and improved version that is easily installable on Raspbian:

```
$ sudo apt-get install vim
```

Throughtout this guide I commonly will have `vi` commands for editing files. You can simply replace this with `nano` if you wish to use the default editor.

Changing the default Hostname

By default, Raspbian installs with the hostname `raspberry`. If you wish to change this to something more memorable, you just need to edit this value and then restart the hostname service:

```
$ sudo vi /etc/hosts
$ sudo vi /etc/hostname
$ sudo /etc/init.d/hostname.sh
```

Accessing Remotely

It's all well and good to use your new Raspberry Pi machine via the keyboard and monitor you have plugged into it, but it's much easier to simply access it remotely from your regular desktop. To do this your best friend is SSH (Secure Shell). In order to reach your Raspberry Pi, you will need to know the IP address. To do so simply type:

```
$ hostname -I
```

This will report the current IP address of the machine.

On your regular computer, you can simply create a `/etc/hosts` entry with this IP and a suitable name. Usually this should match the hostname you just set. Then you can access your Pi remotely via:

```
$ ssh pi@raspberry
```

Enter the password you set in the configuration step, or if you have not set the password yet, it will be `raspberry`. More details can be found in the docs.

It is strongly advised to use SSH Keys rather than passwords. There is a great tutorial on how to set up Passwordless SSH Access in the offical docs.

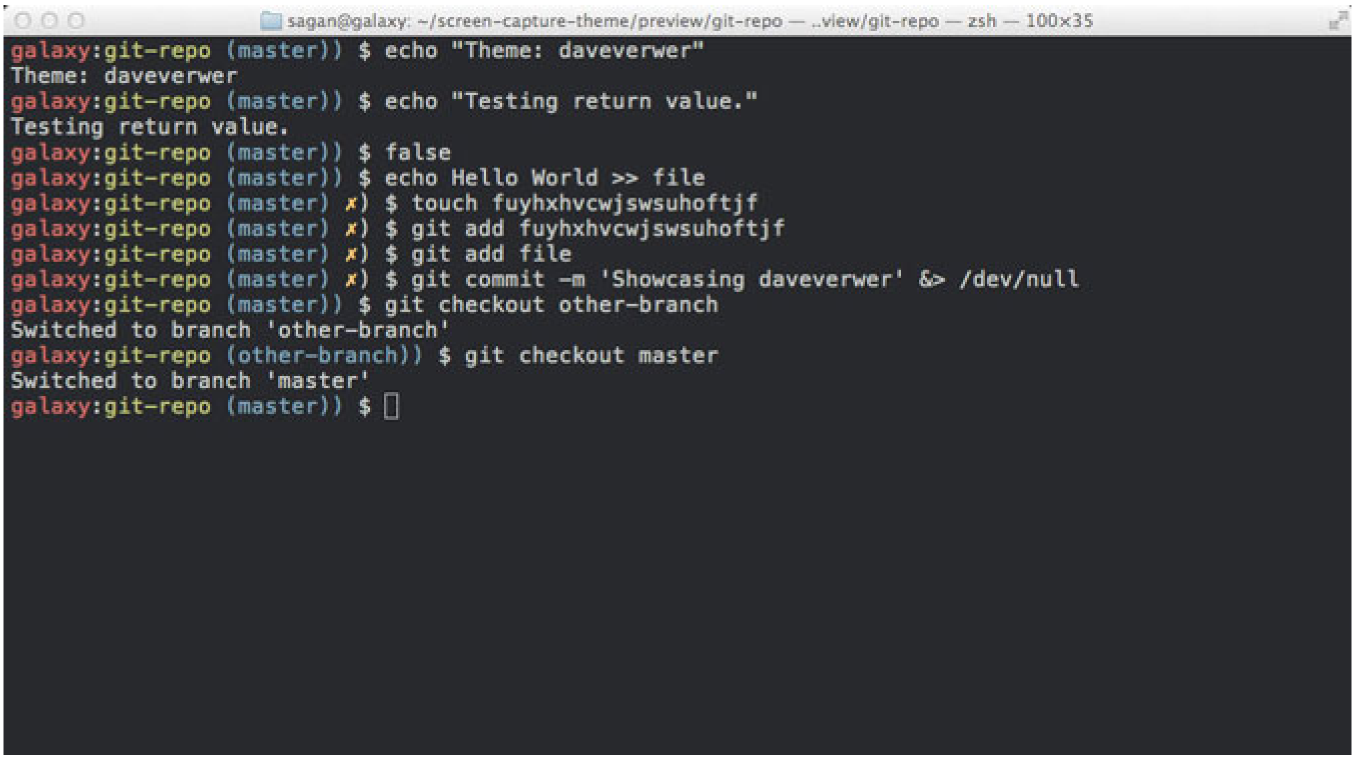
Updating Firmware (Optional)

Your Raspbian installation also included a pretty recent copy of the Raspberry Pi firmware. However, sometimes there are important updates, so it really doesn't hurt to update this to the latest version:

```
$ sudo rpi-update
$ sudo reboot
```

Change the Shell (Optional)

By default, Raspbian ships with the very capable bash shell. However, I personally prefer zsh which has many useful features. On top of this I like to use Oh My ZSH! which is a handy ZSH configuration tool that comes packed with custom plugins, themes, etc.


sagan@galaxy: ~/screen-capture-theme/preview/git-repo — ..view/git-repo — zsh — 100×35

```
galaxy:git-repo (master)) $ echo "Theme: daveverwer"
Theme: daveverwer
galaxy:git-repo (master)) $ echo "Testing return value."
Testing return value.
galaxy:git-repo (master)) $ false
galaxy:git-repo (master)) $ echo Hello World >> file
galaxy:git-repo (master) ✗) $ touch fuyhxhvcwjswsuhoftjf
galaxy:git-repo (master) ✗) $ git add fuyhxhvcwjswsuhoftjf
galaxy:git-repo (master) ✗) $ git add file
galaxy:git-repo (master) ✗) $ git commit -m 'Showcasing daveverwer' &> /dev/null
galaxy:git-repo (master)) $ git checkout other-branch
Switched to branch 'other-branch'
galaxy:git-repo (other-branch)) $ git checkout master
Switched to branch 'master'
galaxy:git-repo (master)) $ 
```

To install this just follow these steps:

```
$ sudo apt-get install zsh
$ wget https://github.com/robbyrussell/oh-my-zsh/raw/master/tools/install.sh -O -
$ chsh -s `which zsh`
```

Then simply log out and back in to see your new shell in all it's glory! I like to change the theme to nice clean multi-line version. To do so simply edit the `~/.zshrc` file and change the theme to one of the many supported themes (I personally like `candy`).

Install PHP 7.0

Raspbian being based on Debian Jessie ships with PHP 5.6 by default. This works fine, but there is a better option and that isPHP 7.0. Released in December of 2015, PHP 7.0 is a radical jump forwards for PHP in terms of performance, language features and reduction in memory usage.

As with most PHP-centric projects, Grav commonly sees 2X performance increases when running under PHP 7.0 and that combined with the reduction in memory, makes it an ideal candidate for the Raspberry Pi.

To install this newer PHP version however, we must tap into thetesting branch of Raspbian, commonly known by the codename stretch. We must edit the `sources.list` file used by Aptitude (`apt-get`):

```
$ sudo vi /etc/apt/sources.list
```

Add this line under the first jessie line:

```
deb http://mirrordirector.raspbian.org/raspbian/ stretch main contrib non-free rp
```

Now, by adding this all installs or updates will default to use the newer versions of files available in the stretch release which is not considered 100% stable. The way around this is to pin all packages to use the jessie release with a higher priority by default. To do this create a preferences file:

```
$ sudo vi /etc/apt/preferences
```

And paste in the follwing:

```
Package: *
Pin: release n=jessie
Pin-Priority: 600
```

Save this file and update:

```
$ sudo apt-get update
```

You can see which release versions are available with which priority by using `sudo apt-cache policy`. You can also see specific versions of packages with `sudo apt-cache policy <package_name>`

Now you are ready to install PHP 7.0 from the stretch release including all the common PHP packages:

```
$ sudo apt-get install -t stretch php7.0 php7.0-curl php7.0-gd php7.0-fpm php7.0-
```

Notice the use of `-t stretch` to specify the release? This is not strictly necessary as `php7.0` packages do not exist in the jessie release, but again, better to be specific here.

After this has been completed, you can quickly test to make sure things have been installed by simply typing:

```
$ php -v
PHP 7.0.4-7 (cli) ( NTS )
Copyright (c) 1997-2016 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies
    with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies
```

Installation of nginx 1.9

While it's trivial to install nginx 1.6 with Raspbian, we want to use a faster, more up to date, and secure version of nginx here, and luckily there's a nginx 1.9 version available in the stretchrelease.

```
$ sudo apt-get install -t stretch nginx
```

You can easily see what version would install by simulatingan install: `sudo apt-get install -t stretch -s`