

# Ultimate Open Source Web Server Part 2

de [Articles and Tutorials | Lehi Sanchez](#) de Lehi Sanchez

In [Part I](#) we covered the simple installation of Ubuntu 8.04.1 Server Edition and set up SSH keys and established a simple but effective firewall. In part 2, we're going to be installing all of the software we need in order to host data-driven web applications ([Ruby on Rails](#), [CakePHP](#), etc.), static html files, and popular content management systems like [Wordpress](#) and [Joomla!](#).

There are a lot of tools and a lot of software that we could use to configure our web server. The following list is what I chose to utilize in my web server. I have no evidence that my setup is the fastest in the world or the best. Personally, I just feel good with what I have and that makes all the difference.

DISCLAIMER: Follow this tutorial at your own risk. You are responsible for Your computer.

## Software

- **MySQL, SQLite3** - Databases
- **Ruby, RubyGems, PHP** - Server-Side Languages
- **Ruby on Rails** - Framework
- **NGINX, Thin, FastCGI** - Web Servers
- **Subversion, Capistrano, GIT** - Version Control

## Databases

### MySQL

```
sudo aptitude install mysql-server mysql-client libmysqlclient15-dev libmysql-ruby1.8
```

Enter a password when prompted and then confirm   **SQLite3**

```
sudo aptitude install sqlite3
```

Easy right?

## Server-Side Languages

Let's dive right in. **Ruby**

```
sudo aptitude install ruby1.8-dev ruby1.8 ri1.8 rdoc1.8 irb1.8 libreadline-ruby1.8 libruby1.8  
libopenssl-ruby libsqlite3-ruby1.8
```

```
sudo ln -s /usr/bin/ruby1.8 /usr/bin/ruby
```

```
sudo ln -s /usr/bin/ri1.8 /usr/bin/ri
```

```
sudo ln -s /usr/bin/rdoc1.8 /usr/bin/rdoc
```

```
sudo ln -s /usr/bin/irb1.8 /usr/bin/irb
```

**RubyGems** We'll be installing RubyGems from the [latest release source](#). Let's create a source folder for all of our source files while we configure our web server and let's download the latest version.

```
sudo mkdir ~/src
```

```
cd ~/src
```

```
sudo wget http://rubyforge.org/frs/download.php/43985/rubygems-1.3.0.tgz
```

```
sudo tar xzvf rubygems-1.3.0.tgz
```

```
cd rubygems-1.3.0
```

```
sudo ruby setup.rb
```

```
sudo ln -s /usr/bin/gem1.8 /usr/bin/gem
```

```
sudo gem update
sudo gem update --system
```

```
cd ~
```

### Ruby on Rails

```
sudo gem install rails --no-rdoc --no-ri
```

### PHP

```
sudo aptitude install php5-cli php5-cgi php5-mysql php5-xcache -y
```

## Web Servers

### NGINX

```
sudo aptitude install libpcre3 libpcre3-dev libpcrecpp0 libssl-dev zlib1g-dev
```

```
cd ~/src
```

```
sudo wget http://sysoev.ru/nginx/nginx-0.6.31.tar.gz
```

```
sudo tar -zxvf nginx-0.6.31.tar.gz
```

```
cd nginx-0.6.31
```


```
./configure --sbin-path=/usr/local/sbin --with-http_ssl_module
```

```
make
```

```
sudo make install
```

Start NGINX

```
sudo /usr/local/sbin/nginx
```

Navigate to your server's ip address. You should see the following  **Spawn-fcgi** We'll need this for our PHP files.

```
cd ~/src
```

```
sudo wget http://www.lighttpd.net/download/lighttpd-1.4.18.tar.bz2
```

```
sudo tar -xvjf lighttpd-1.4.18.tar.bz2
```

```
cd lighttpd-1.4.18
```

```
./configure
```

```
make
```

```
sudo cp src/spawn-fcgi /usr/bin/spawn-fcgi
```

```
cd ~
```

**Thin** This is the equivalent of a Mongrel cluster.

```
sudo gem install thin --no-rdoc --no-ri
```

And that will do it!

## Version Control

### Subversion

```
sudo aptitude install subversion
```

**GIT** I'm going to install Git, which is another version control utility, because I haven't made up my mind as to which version control I'm going to use. So, I'm going to install it in case I end up going the Git way.

```
sudo aptitude install git-core
```

### Capistrano

```
sudo gem install capistrano --no-rdoc --no-ri
```

This wraps things up for part II. In Part III we'll be setting up our "Hello World!" websites using Ruby on Rails, CakePHP, Wordpress, and Joomla!. While we're at it we'll set up PHPMysqlAdmin for easy MySQL management.