

# How To Install Jekyll in Linux Mint 19

```
greys@xps:~/proj/glebreys.com$ bundle exec jekyll serve
Configuration file: /home/greys/proj/glebreys.com/_config.yml
   Source: /home/greys/proj/glebreys.com
  Destination: /home/greys/proj/glebreys.com/_site
 Incremental build: disabled. Enable with --incremental
  Generating...
   Jekyll Feed: Generating feed for posts
                  done in 0.614 seconds.
 Auto-regeneration: enabled for '/home/greys/proj/glebreys.com'
   Server address: http://127.0.0.1:4000/
  Server running... press ctrl-c to stop.
[2019-03-29 13:48:46] ERROR `./favicon.ico' not found.
```

I'm still fascinated with the [Jekyll](#) approach to website management, and am working on converting one of my blogs (not [Unix Tutorial](#) just yet!) to Jekyll website. This short post shows how to install Jekyll on a Linux Mint system.

## Install Ruby and Bundler

First things first: you need to install Ruby:

```
greys@xps:~/proj$ sudo apt install ruby
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
libssh-4
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
libruby2.5 rake ruby-test-unit ruby2.5
Suggested packages:
ri ruby-dev
Recommended packages:
fonts-lato libjs-jquery
The following NEW packages will be installed:
```

```
libruby2.5 rake ruby ruby-test-unit ruby2.5
0 upgraded, 5 newly installed, 0 to remove and 317 not
upgraded.
Need to get 3,227 kB of archives.
After this operation, 14.8 MB of additional disk space will be
used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/main
amd64 ruby2.5 amd64 2.5.1-1ubuntu1.1 [48.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby
amd64 1:2.5.1 [5,712 B]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 rake
all 12.3.1-1 [45.1 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-
test-unit all 3.2.5-1 [61.1 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates/main
amd64 libruby2.5 amd64 2.5.1-1ubuntu1.1 [3,066 kB]
Fetched 3,227 kB in 1s (3,576 kB/s)
Selecting previously unselected package ruby2.5.
(Reading database ... 272296 files and directories currently
installed.)
Preparing to unpack .../ruby2.5_2.5.1-1ubuntu1.1_amd64.deb ...
Unpacking ruby2.5 (2.5.1-1ubuntu1.1) ...
Selecting previously unselected package ruby.
Preparing to unpack .../ruby_1%3a2.5.1_amd64.deb ...
Unpacking ruby (1:2.5.1) ...
Selecting previously unselected package rake.
Preparing to unpack .../archives/rake_12.3.1-1_all.deb ...
Unpacking rake (12.3.1-1) ...
Selecting previously unselected package ruby-test-unit.
Preparing to unpack .../ruby-test-unit_3.2.5-1_all.deb ...
Unpacking ruby-test-unit (3.2.5-1) ...
Selecting previously unselected package libruby2.5:amd64.
Preparing to unpack .../libruby2.5_2.5.1-1ubuntu1.1_amd64.deb
...
Unpacking libruby2.5:amd64 (2.5.1-1ubuntu1.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up rake (12.3.1-1) ...
Setting up ruby2.5 (2.5.1-1ubuntu1.1) ...
Setting up ruby (1:2.5.1) ...
```

```
Setting up ruby-test-unit (3.2.5-1) ...
Setting up libruby2.5:amd64 (2.5.1-1ubuntu1.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
```

... and now use the gem command to install bundler gem:

```
greys@xps:~/proj$ sudo gem install bundler
Successfully installed bundler-2.0.1
Parsing documentation for bundler-2.0.1
Done installing documentation for bundler after 1 seconds
1 gem installed
```

## Install the Jekyll gem

Excellent, we're really close to getting this working.

Only small problem: when installing the jekyll gem, there's an error:

```
greys@xps:~/proj$ sudo gem install jekyll
Building native extensions. This could take a while...
ERROR: Error installing jekyll:
ERROR: Failed to build gem native extension.
```

```

           current           directory:
/var/lib/gems/2.5.0/gems/ffi-1.10.0/ext/ffi_c
/usr/bin/ruby2.5  -r  ./siteconf20190329-23694-7cpq9s.rb
extconf.rb
mkmf.rb  can't find header files for ruby at
/usr/lib/ruby/include/ruby.h
```

```
extconf failed, exit code 1
```

```
Gem files will remain installed in
/var/lib/gems/2.5.0/gems/ffi-1.10.0 for inspection.
```

```
Results logged to /var/lib/gems/2.5.0/extensions/x86_64-
linux/2.5.0/ffi-1.10.0/gem_make.out
```

I first thought I did something wrong or that I have an old version of Ruby (2.5.0 as you can see). But now, it seems the minimal required version for Jekyll is Ruby 2.1.0 so it should

all work.

The hint is in the error message:

```
mkmf.rb can't find header files for ruby at /usr/lib/ruby/include/ruby.h
```

This include file is indeed missing, because previously we simply installed Ruby binaries, but not the development packages.

Once we install ruby-dev package:

```
greys@xps:~/proj$ sudo apt install ruby-dev
```

```
Reading package lists... Done
```

```
Building dependency tree
```

```
Reading state information... Done
```

```
The following package was automatically installed and is no longer required:
```

```
libssh-4
```

```
Use 'sudo apt autoremove' to remove it.
```

```
The following additional packages will be installed:
```

```
ruby2.5-dev
```

```
Recommended packages:
```

```
ruby2.5-doc
```

```
The following NEW packages will be installed:
```

```
ruby-dev ruby2.5-dev
```

```
0 upgraded, 2 newly installed, 0 to remove and 317 not upgraded.
```

```
Need to get 68.3 kB of archives.
```

```
After this operation, 351 kB of additional disk space will be used.
```

```
Do you want to continue? [Y/n] y
```

```
Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 ruby2.5-dev amd64 2.5.1-1ubuntu1.1 [63.7 kB]
```

```
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-dev amd64 1:2.5.1 [4,604 B]
```

```
Fetches 68.3 kB in 0s (198 kB/s)
```

```
Selecting previously unselected package ruby2.5-dev:amd64.
```

```
(Reading database ... 273441 files and directories currently installed.)
```

```
Preparing to unpack .../ruby2.5-dev_2.5.1-1ubuntu1.1_amd64.deb
```

```
...
```

```
Unpacking ruby2.5-dev:amd64 (2.5.1-1ubuntu1.1) ...
```

```
Selecting previously unselected package ruby-dev:amd64.
```

```
Preparing to unpack .../ruby-dev_1%3a2.5.1_amd64.deb ...
Unpacking ruby-dev:amd64 (1:2.5.1) ...
Setting up ruby2.5-dev:amd64 (2.5.1-1ubuntu1.1) ...
Setting up ruby-dev:amd64 (1:2.5.1) ...
```

The header files for Ruby should appear now (although in a slightly different path):

```
/usr/include/ruby-2.5.0/ruby/ruby.h
/usr/include/ruby-2.5.0/ruby.h
```

and the Jekyll gem install will complete:

```
greys@xps:~/proj$ sudo gem install jekyll
Building native extensions. This could take a while...
Successfully installed ffi-1.10.0
Fetching: rb-inotify-0.10.0.gem (100%)
Successfully installed rb-inotify-0.10.0
Fetching: rb-fsevent-0.10.3.gem (100%)
Successfully installed rb-fsevent-0.10.3
Fetching: listen-3.1.5.gem (100%)
Successfully installed listen-3.1.5
Fetching: jekyll-watch-2.2.1.gem (100%)
Successfully installed jekyll-watch-2.2.1
Fetching: sass-listen-4.0.0.gem (100%)
Successfully installed sass-listen-4.0.0
Fetching: sass-3.7.3.gem (100%)
```

Ruby Sass is deprecated and will be unmaintained as of 26 March 2019.

\* If you use Sass as a command-line tool, we recommend using Dart Sass, the new primary implementation: <https://sass-lang.com/install>

\* If you use Sass as a plug-in for a Ruby web framework, we recommend using the sassc gem: <https://github.com/sass/sassc-ruby#readme>

\* For more details, please refer to the Sass blog: <http://sass.logdown.com/posts/7081811>

```
Successfully installed sass-3.7.3
```

Fetching: jekyll-sass-converter-1.5.2.gem (100%)  
Successfully installed jekyll-sass-converter-1.5.2  
Fetching: concurrent-ruby-1.1.5.gem (100%)  
Successfully installed concurrent-ruby-1.1.5  
Fetching: i18n-0.9.5.gem (100%)  
Successfully installed i18n-0.9.5  
Fetching: http\_parser.rb-0.6.0.gem (100%)  
Building native extensions. This could take a while...  
Successfully installed http\_parser.rb-0.6.0  
Fetching: eventmachine-1.2.7.gem (100%)  
Building native extensions. This could take a while...  
Successfully installed eventmachine-1.2.7  
Fetching: em-websocket-0.5.1.gem (100%)  
Successfully installed em-websocket-0.5.1  
Fetching: colorator-1.1.0.gem (100%)  
Successfully installed colorator-1.1.0  
Fetching: public\_suffix-3.0.3.gem (100%)  
Successfully installed public\_suffix-3.0.3  
Fetching: addressable-2.6.0.gem (100%)  
Successfully installed addressable-2.6.0  
Fetching: jekyll-3.8.5.gem (100%)  
Successfully installed jekyll-3.8.5  
Parsing documentation for ffi-1.10.0  
Installing ri documentation for ffi-1.10.0  
Parsing documentation for rb-inotify-0.10.0  
Installing ri documentation for rb-inotify-0.10.0  
Parsing documentation for rb-fsevent-0.10.3  
Installing ri documentation for rb-fsevent-0.10.3  
Parsing documentation for listen-3.1.5  
Installing ri documentation for listen-3.1.5  
Parsing documentation for jekyll-watch-2.2.1  
Installing ri documentation for jekyll-watch-2.2.1  
Parsing documentation for sass-listen-4.0.0  
Installing ri documentation for sass-listen-4.0.0  
Parsing documentation for sass-3.7.3  
Installing ri documentation for sass-3.7.3  
Parsing documentation for jekyll-sass-converter-1.5.2  
Installing ri documentation for jekyll-sass-converter-1.5.2  
Parsing documentation for concurrent-ruby-1.1.5  
Installing ri documentation for concurrent-ruby-1.1.5  
Parsing documentation for i18n-0.9.5

```
Installing ri documentation for i18n-0.9.5
Parsing documentation for http_parser.rb-0.6.0
Installing ri documentation for http_parser.rb-0.6.0
Parsing documentation for eventmachine-1.2.7
Installing ri documentation for eventmachine-1.2.7
Parsing documentation for em-websocket-0.5.1
Installing ri documentation for em-websocket-0.5.1
Parsing documentation for colorator-1.1.0
Installing ri documentation for colorator-1.1.0
Parsing documentation for public_suffix-3.0.3
Installing ri documentation for public_suffix-3.0.3
Parsing documentation for addressable-2.6.0
Installing ri documentation for addressable-2.6.0
Parsing documentation for jekyll-3.8.5
Installing ri documentation for jekyll-3.8.5
Done installing documentation for ffi, rb-inotify, rb-fsevent,
listen, jekyll-watch, sass-listen, sass, jekyll-sass-
converter, concurrent-ruby, i18n, http_parser.rb,
eventmachine, em-websocket, colorator, public_suffix,
addressable, jekyll after 26 seconds
17 gems installed
```

Great stuff! Now we can serve my project and browse to the Jekyll page (I skip the jekyll steps for starting new website because they're described in my [Jekyll website with GitHub Pages](#) article):

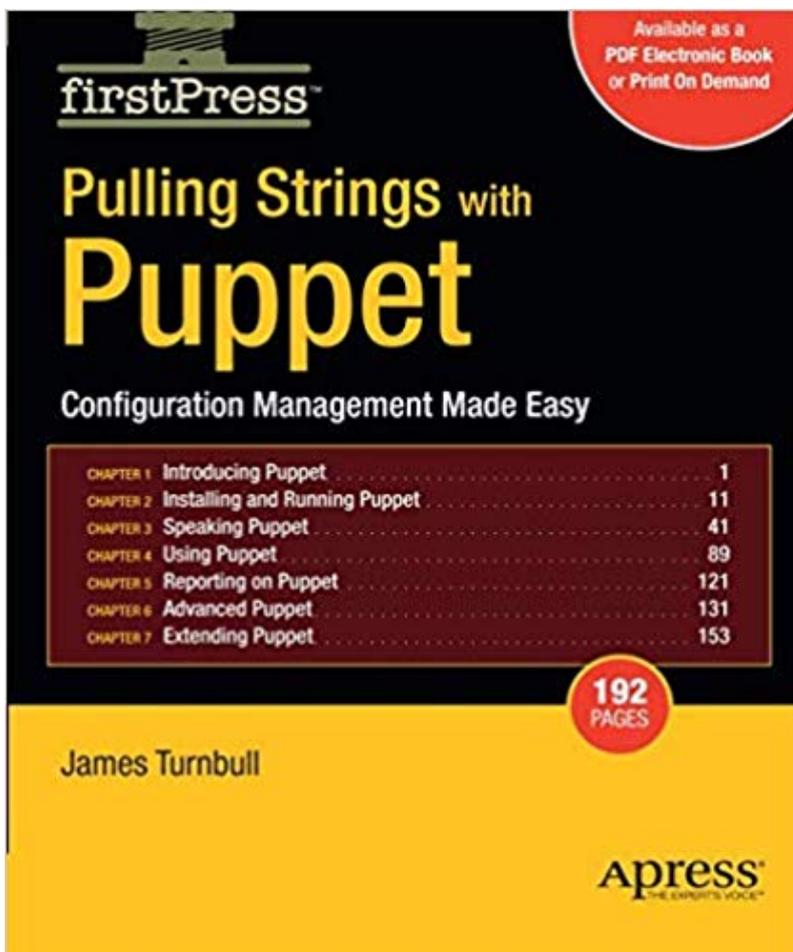
```
greys@xps:~/proj/glebreys.com$ bundle exec jekyll serve
Configuration file: /home/greys/proj/glebreys.com/_config.yml
Source: /home/greys/proj/glebreys.com
Destination: /home/greys/proj/glebreys.com/_site
Incremental build: disabled. Enable with --incremental
Generating...
Jekyll Feed: Generating feed for posts
done in 0.614 seconds.
Auto-regeneration: enabled for '/home/greys/proj/glebreys.com'
Server address: http://127.0.0.1:4000/
Server running... press ctrl-c to stop.
```

That's it for today!

## See Also

- [Unix Tutorial Projects: GitHub Pages with Jekyll](#)
- 

# Book Review: Pulling Strings with Puppet



## Pulling Strings with Puppet

**Puppet** is an incredibly popular Ruby-based configuration management tool. Gaining its popularity with the open-source edition, it has become popular enough to also appear as a Puppet PE – Puppet Enterprise edition.

**Update February 2019:** I'm migrating book reviews from another website of mine, so this isn't a recent review but still a great book I think you should read if you get a chance.

[Puppeting Strings with Puppet](#) is now a classic: written 10 years ago, it's still a really good introduction to configuration management with Puppet. James Turnbull has since written an even better book (**Pro Puppet**) and I'll probably find a few even more recent introductions to Puppet, but if you have [Pulling Strings with Puppet](#) on your colleague's desk I suggest you borrow it.

## What is Puppet?

Puppet is a framework for automatic configuration management of your systems. Originally oriented on Unix systems and servers specifically, it can now be used to manage quite a range of Unix-like systems as well as Windows environments.

Both open source and enterprise (paid) edition allow you to greatly optimize and automate the process of deploying and maintaining configurations of your environments via a scriptable core that can be configured to suit all of your needs.

Puppet has 400+ configurations already tested and available online on **Puppet Forge** – this is basically a collection of downloadable modules for automating all sorts of systems administration tasks.

The [Pulling Strings with Puppet book](#) gives a very good introduction into common tasks of configuration management and explains the multiple layers of Puppet-based solution very well: there's a declarative language used for defining classes and modules, then there's a transaction layer for creating and deploying updates and, finally, there is a resource abstraction layer – this piece of magic that makes it possible to use the same configuration stanza for deploying the same

change to vastly different Unix-like distros.

## Deploying a Puppet infrastructure

**Puppet** servers are called *puppet masters*. These are the servers which store all configurations along with list of *Puppet clients* (sometimes called *Puppet nodes*). Clients communicate back to **Puppet** servers at regular intervals using RESTful API over HTTPs, checking in and downloading configurational updates if necessary.

Self-signed certificates are a requirement for all the clients and the way these are managed on the Puppet master side ensures that no unauthorized access can be gained easily enough. You will find a good enough description of how this works in practice so it will be very easy to get started if you're new to **Puppet**.

The book goes into lots of detail when it comes to explaining the pre-requisites (Ruby, Facter) and installation process (compiling from sources or deploying from packages). The easiest way to deploy a Puppet client or server is probably to get Ruby installed and then deploy a Puppet Ruby gem.

I really liked the table with descriptions of all the common **Puppet** management commands – it's a neat little reference giving you a good idea of what's possible and where to look for more information once you become comfortable enough.

## Using Puppet for managing configurations

It all starts with the `[main]` configuration namespace, and before you know it the book effortlessly takes you through configuring resources and attributes and deploying classes and modules. **Puppet** relies heavily on the **Facter** framework which is an abstract way of documenting, configuring and presenting useful configuration elements about your environments like OS version or name of your Linux distro. The book gives a very

useful description of the approach to using facts and configurative definitions.

A good few pages are used to make sure you will have the full understanding of what variables are possible in Puppet and how variable scoping is probably different from most of the scripting and programming languages you already know.

You will learn about virtual resources and how they need to be realized before the changes are actually applied, and finally will be introduced to various default types available in a Puppet installation (cron/exec/file/filebucket/group/package/yumrepo and quite a few others).

## **Sample Puppet configuration management environment**

Perhaps the most valuable element of the whole book is a complete description of nodes, classes, users and groups needed to deploy and support a typical LAMP environment. In addition to sample configurations for managing nodes and users, you are given full code for modules managing MySQL, Apache and Postfix.

The language used for composing classes and modules in Puppet is pretty straightforward, but the complete examples will help you not only learn the syntax but also pick up some of the best practices when it comes to starting even the simplest of Linux configuration management environments.

For the most technically curious minds there are hints for deploying custom set of Facter facts and even if that doesn't impress enough you are given instructions for creating your own type for the most flexible resource/configuration management.

## Advanced Puppet usage

The last few chapters in the Pulling Strings with Puppet book show you some more advanced challenges you'll face if you chose to explore Puppet in depth.

In addition to getting plenty of hints for performance optimisation and scaling out of your **Puppet** infrastructure, you will find information about migrating nodes description information into an external storage (scripts or indeed proper datastores like LDAP).

For scaling, you'll get basic information about **Mongrel** (apparently it's a Ruby friendly webserver you can use instead of Apache) but also full configuration examples for configuring Apache as a proxy and a load balancing solution pointing to multiple Puppet instances.

Even if you are well versed in `mod_proxy` or `mod_ajp`, reading the chapter about Puppet scalability with Mongrel and Apache proxy will be a time well spent – it doesn't just show you what steps are needed for the desired configuration but gives you explanations of the typical challenges you'll be trying to solve.

## Summary for the Pulling Strings with Puppet book

I had thoroughly enjoyed this book the first time I had read it back in 2010 and I can still recommend it after reading it again in 2012. Majority of the topics in the book are still quite relevant and easily applicable even today, which is probably a testament to both the talents of the book author and the great planning/roadmapping of the founding geniuses behind the **Puppet** framework.

firstPress™

Available as a  
PDF Electronic Book  
or Print On Demand

# Pulling Strings with Puppet

Configuration Management Made Easy

CHAPTER 1	Introducing Puppet	1
CHAPTER 2	Installing and Running Puppet	11
CHAPTER 3	Speaking Puppet	41
CHAPTER 4	Using Puppet	89
CHAPTER 5	Reporting on Puppet	121
CHAPTER 6	Advanced Puppet	131
CHAPTER 7	Extending Puppet	153

192  
PAGES

James Turnbull

Apress®  
THE EXPERT'S VOICE™

Pulling Strings with Puppet