How to Install Ruby on Rails on Ubuntu Linux

launchschool.com/blog/how-to-install-ruby-on-rails-development-environment-for-linux

By Juan Lebrijo

(Related article: How to Install Ruby on Rails for Mac OSX Mavericks)

For this post we are based on a clean “Ubuntu 14.04 LTS” installation, nevertheless I have installed these tools in previous versions with the same steps. And similar steps should work in other Linux ditros.

For day to day Ruby on Rails development work we need several tools to be porperly installed and running:

- Git: Right now this is the standard Version Control System in the industry. Even more!! I work with git in all my client projects since a few years ago.
- Rbenv (and ruby): lightweight Ruby Version Manager to isolate Ruby versions and project dependencies.
- Rails: Our favourite framework to build Web apps.
- Sublime Text Editor: lightweight modern editor.
- PostgreSQL: Probably the best Database in the open source market (in fact my favourite)

For Ubuntu my first recommendation, before to make a big installation, is updating system packages:

```
1  sudo apt-get
   update
```

Step 1. Installing Git

Git is the standard VCS for the Ruby community, and probably for the IT industry at this moment. Also it is pretty simple to install at Ubuntu:

```
1  sudo apt-get install
   git
```

Git will need your email and name at least to identify your commits, but I have added some other configurations interesting for my day-to-day work. Execute each line in prompt:

```
1  git config --global user.name "Name Surname"
2  git config --global user.email
3  nick@company.com
4  git config --global color.ui true
5  git config --global core.editor vi
6  git config --global color.branch auto
7  git config --global color.diff auto
   git config --global color.status auto
```

Probably you will use Github to deliver code to your clients, and it will need to identify you through your ssh-keys. You

can follow this guide in order to create ssh keys (if you do not have them yet) and configure your account.

This process is similar with other Git providers like Bitbucket or your Company owned Git servers.

Step 2. Dynamic prompt with Git and ANSI colors

Tobias Sjösten wrote a pretty useful article explaining this.

This hack will:

- Put our current branch in prompt
- Change color if there are files to commit

This gives you a quick status of your changes in project in a quick eye-shot. You just have to add the following code to ~/.bash_profile file:

```
## GIT colored prompt
############################
# Configure colors, if available.
if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
c_reset='\[\e[0m\]'
c_user='\[\e[1;33m\]'
c_path='\[\e[0;33m\]'
c_git_cleancleann='\[\e[0;36m\]'
c_git_dirty='\[\e[0;35m\]'
else
c_reset=
c_user=
c_git_cleancleann_path=
c_git_clean=
c_git_dirty=
fi

# Function to assemble the Git parsingart of our prompt.
git_prompt ()
{
if ! git rev-parse --git-dir > /dev/null 2>&1; then
return 0
```

```
fi

git_branch=$(git branch 2>/dev/null | sed -n '/^\*/s/^\* //p')

if git diff --quiet 2>/dev/null >&2; then
git_color="$c_git_clean"
else
git_color="$c_git_dirty"
fi

echo "[$git_color$git_branch${c_reset}]"
}

# Thy holy prompt.
PROMPT_COMMAND='PS1="${c_user}\u${c_reset}@${c_user}\h${c_reset}:${c_path}\w${c_reset}$(git_prompt)\$
"'
```

You should restart terminal for these changes to take effect.

Step 3. Install Ruby with RBENV

RVM and RBENV are the most used Ruby Version Managers among developers. Rbenv is lighter and solves the dependency isolation problem working with Bundle (the default dependency management gem for Rails).

RBENV give us several advantages:

- The obvious ‘apt-get install ruby’ is outdated, and you probably want to work with latest ruby versions.
- Allow different versions of ruby in the same machine, which will be useful when we write code in different projects.
- Dependency isolation working with bundle.

You need Git that was installed previously to install RBENV.

```
git clone git://github.com/sstephenson/rbenv.git .rbenv
echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
echo 'eval "$(rbenv init -)"' >> ~/.bashrc

git clone git://github.com/sstephenson/ruby-build.git
~/.rbenv/plugins/ruby-build
echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >>
~/.bashrc

git clone https://github.com/sstephenson/rbenv-gem-rehash.git
~/.rbenv/plugins/rbenv-gem-rehash

git clone https://github.com/ianheggie/rbenv-binstubs.git
~/.rbenv/plugins/rbenv-binstubs
```

With commands above you have installed Rbenv and the ruby versions builder, and give access to them directly from command prompt. Please, restart your terminal to load Rbenv.

The only disadvantage of BINSTUBS is that you have to remember to run the following command

```
bundle install --binstubs
.bundle/bin
```

Now, you can install every ruby version with a command:

```
# list all available versions:
$ rbenv install -l

# install a Ruby version:
$ rbenv install 2.1.2

# choose default ruby version on your
system
$ rbenv global 2.1.2
```

Step 4. Your first Rails App

First you need to install Rails gem:

```
gem install
rails
```

Create your new app:

```
1  rails new name_of_your_new_app
2  cd name_of_your_new_app
```

Go to Gemfile and uncomment this line:

```
1  gem 'therubyracer',  platforms:
   :ruby
```

The gem “Therubyracer” is Google V8 engine wrapped in a gem, in order to have the capability of evaluating JavaScript code from Ruby. Pretty interesting in Rails templates and assets.

After that re-install gems:

```
1  bundle
   install
```

And start it in your browser at http://localhost:3000

```
1  rails s
```

Step 5. Set Up Sublime Text as Code Editor

There are several code editors in the market to edit ruby code: SublimeText, RubyMine, Aptanta,…. Or Opensource like vim, gedit or emacs. Sublime Text is an excellent choice if you want a lightweight editor with great features and a modern user interface.

Download and unzip last version from the web page. In order to give access from command prompt:

```
1  sudo mv Sublime\ Text\ 2 /opt/
2  sudo ln -s /opt/Sublime\ Text\ 2/sublime_text
   /usr/bin/subl
```

Now you can open your Rails app by typing:

```
1  cd name_of_your_new_app
2  subl .
```

The convention for Ruby programs is to use two spaces as indentation. You can follow `Sublime Text 2 => Preferences => Settings - User` and add these lines.

```
{
"tab_size": 2,
"translate_tabs_to_spaces":
true
}
```

Step 6. PostgreSQL installation

This is a pretty good Database and maybe this will be your Production relational database. So worth it to take a look at how to integrate it with rails:

```
sudo apt-get install postgresql libpq-
dev
```

You can add the gem to your Gemfile:

```
gem 'pg'
```

And run bundle again: bundle install, to make PG gem installation.

After that your Database config file should be like:

```
common: &common
  adapter: postgresql
  username: foxwordy
  password: foxwordy # from psql setup, see
Postgresql

development:
  <<: *common
  database: development

test: &test
  <<: *common
  database: test

production:
  <<: *common
  database: production
```

Extra Ball. Set up RubyMine

Here at Tealeaf Academy we recommend Sublime for our students, because is lighter and easy to use. But if you need a more complete editor: with debugger, Git or Ticketing System integration, plugins … . RubyMine is a great choice.

You need a Java Runtime installation before:

```
1  sudo apt-get install openjdk-7-
   jdk
```

Download and install:

```
1  tar xvf RubyMine-6.3.2.tar.gz
2  mv RubyMine-6.3.2 ~
3  ~/RubyMine-6.3.2/bin/rubymine.sh
```

They allow you a 30 day trial version, after that you should purchase a license.

Then you can open every Rails application within the project folder with the command:

```
1  mine
   .
```

Conclusion

Well, this is just the beginning. Here we just installed our Dev environment with a bunch of really useful tools, which you will use in your day-to-day work.

If you want a structured, instructor led program to level up to a professional level Ruby on Rails developer, check out our curriculum!