

[Commands](#)[Subscribe](#)[ISPConfig](#)[News](#)

with PostgreSQL on Ubuntu 16.04

How to Install Ruby on Rails (RoR) with PostgreSQL on Ubuntu 16.04

Ruby on Rails (RoR) is an open source web application framework, published under MIT License. Ruby on Rails, or rails is written in Ruby, it is a server-side web application framework that follows the MVC (Model-View-Controller) concept. Rails is providing default structures for the database, web service, and web pages. More than 3000 developers have been contributed code to the Rails framework and there are many well-known applications based on Rails, such as Github, Airbnb, Soundcloud etc.

In this tutorial, I will show you how to install Ruby on Rails on ubuntu 16.04 LTS. I will show you how to install and configure Rails with a PostgreSQL database, and how to create a new first project with Rails.

On this page

- [Step 1 - Install RVM](#)
- [Step 2 - Install Ruby](#)
- [Step 3 - Install Nodejs](#)
- [Step 4 - Configure Ruby Gem](#)
- [Step 5 - Install Ruby on Rails 5 Stable](#)
- [Step 6 - Setting up the PostgreSQL Database for Rails Development](#)
- [Step 7 - Create the First Application with Rails and PostgreSQL](#)
- [Reference](#)

Prerequisite

- Ubuntu 16.04 Server.
- Root privileges.

Step 1 - Install RVM

RVM or Ruby Version Manager is a command-line tool based on Bash and Ruby to manage the ruby installation. RVM allows you to install and configure multiple ruby versions on one system.

In this step, we will install the stable RVM version and add the repository key with the gpg command.

```
gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3
```

Install RVM stable. Make sure curl command is installed on your system.

```
curl -sSL https://get.rvm.io | bash -s stable --ruby
```

That command will install the requires packages for the RVM installation, then download the latest stable RVM version and install it.

Now we can start using RVM by executing the command below at the first:

```
source /usr/local/rvm/scripts/rvm
```

```

Searching for binary rubies, this might take some time.
Found remote file https://rvm_io.global.ssl.fastly.net/binaries/ubuntu/16.04/x86_64/ruby-2.3.0.tar.bz2
Checking requirements for ubuntu.
Installing requirements for ubuntu.
Updating system..
Installing required packages: g++, libreadline6-dev, zlib1g-dev, libssl-dev, libyaml-dev, libsqlite3-dev, sqlite3,
git, bison, pkg-config, libffi-dev.....
Requirements installation successful.
ruby-2.3.0 - #configure
ruby-2.3.0 - #download
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 24.8M  100 24.8M    0     0  248k      0  0:01:42  0:01:42 --:--:-- 278k
No checksum for downloaded archive, recording checksum in user configuration.
ruby-2.3.0 - #validate archive
ruby-2.3.0 - #extract
ruby-2.3.0 - #validate binary
ruby-2.3.0 - #setup
ruby-2.3.0 - #gemset created /usr/local/rvm/gems/ruby-2.3.0@global
ruby-2.3.0 - #importing gemset /usr/local/rvm/gemsets/global.gems.....
ruby-2.3.0 - #generating global wrappers.....
ruby-2.3.0 - #gemset created /usr/local/rvm/gems/ruby-2.3.0
ruby-2.3.0 - #importing gemsetfile /usr/local/rvm/gemsets/default.gems evaluated to empty gem list
ruby-2.3.0 - #generating default wrappers.....
Creating alias default for ruby-2.3.0...

* To start using RVM you need to run 'source /usr/local/rvm/scripts/rvm'
  in all your open shell windows, in rare cases you need to reopen all shell windows.
root@asuka:~# source /usr/local/rvm/scripts/rvm

```

Step 2 - Install Ruby

The current stable version of Ruby is 2.3.1. Install it with the rvm command and then make it the default ruby version for your system.

Update rvm to the latest stable version, then install Ruby 2.3.1 version.

```

rvm get stable --autolibs=enable
rvm install ruby-2.3.1

```

Next, make 2.3.1 the default ruby version on the system.

```

rvm --default use ruby-2.3.1

```

Now check the ruby version with command below:

```

ruby -v

```

```

root@asuka:~# rvm install ruby-2.3.1
Searching for binary rubies, this might take some time.
Found remote file https://rvm_io.global.ssl.fastly.net/binaries/ubuntu/16.04/x86_64/ruby-2.3.1.tar.bz2
Checking requirements for ubuntu.
Installing requirements for ubuntu.
Requirements installation successful.
ruby-2.3.1 - #configure
ruby-2.3.1 - #download
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 24.8M  100 24.8M    0     0  240k      0  0:01:45  0:01:45 --:--:-- 252k
No checksum for downloaded archive, recording checksum in user configuration.
ruby-2.3.1 - #validate archive
ruby-2.3.1 - #extract
ruby-2.3.1 - #validate binary
ruby-2.3.1 - #setup
ruby-2.3.1 - #gemset created /usr/local/rvm/gems/ruby-2.3.1@global
ruby-2.3.1 - #importing gemset /usr/local/rvm/gemsets/global.gems.....
ruby-2.3.1 - #generating global wrappers.....
ruby-2.3.1 - #gemset created /usr/local/rvm/gems/ruby-2.3.1
ruby-2.3.1 - #importing gemsetfile /usr/local/rvm/gemsets/default.gems evaluated to empty gem list
ruby-2.3.1 - #generating default wrappers.....
root@asuka:~# rvm --default use ruby-2.3.1
Using /usr/local/rvm/gems/ruby-2.3.1
root@asuka:~# ru
ruby               ruby-rvm-env               run-one             run-this-one
ruby-percutable-hooks  run                    run-one-constantly  runuser
rubygems_detect_ruby_lib_gem_path  run_gems_wrappers_regenerate
rubygems_detect_ruby_lib_gem_path_fallback  runlevel
rubygems_detect_ruby_lib_gem_path_from  run-mailcap
run-parts
root@asuka:~# ruby -v
ruby 2.3.1p112 [2016-04-26 revision 54768] [x86_64-linux]
root@asuka:~#

```

Step 3 - Install Nodejs

Rails requires a JavaScript runtime to compile the Rails asset pipeline. For Rails development on Ubuntu Linux, it's best to install Nodejs as the Javascript runtime.

Install nodejs from the nodesource repository:

```

curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash -
apt-get install -y nodejs

```

Step 4 - Configure Ruby Gem

RubyGems is a Ruby Package Manager. It's coming with the gem command-line tool and is automatically installed when we install Ruby on the system.

Update the gem version and check it:

```
gem update --system
gem -v
```

This is optional, you can disable to install the documentation on every gem installation by adding a new line to the .gemrc file below.

```
echo "gem: --no-document" >> ~/.gemrc
```

Step 5 - Install Ruby on Rails 5 Stable

Install Ruby on Rails 5 with the gem command below:

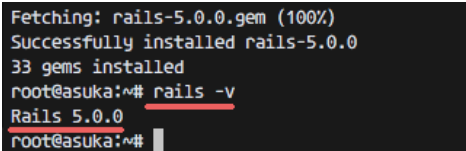
```
gem install rails -v 5.0.0
```

When the installation finished successfully, check the rails version:

```
rails -v
```

And you will see the results of rails version:

Rails 5.0.0



```
Fetching: rails-5.0.0.gem (100%)
Successfully installed rails-5.0.0
33 gems installed
root@asuka:~# rails -v
Rails 5.0.0
root@asuka:~#
```

Rails 5.0 has been installed on Ubuntu 16.04 with Ruby 2.3.1.

Step 6 - Setting up the PostgreSQL Database for Rails Development

In this step, we will prepare PostgreSQL for rails development. Ruby on Rails supports many databases such as MySQL, SQLite (Default) and PostgreSQL. We will use PostgreSQL as the database for this guide.

Install PostgreSQL and some other required packages with the apt command:

```
apt-get -y install postgresql postgresql-contrib libpq-dev
```

When the installation is done, login to the postgres user and access the postgresql shell.

```
su - postgres
psql
```

Give the postgres user a new password with command below:

```
\password postgres
Enter new password:
```

Next, create a new role named 'rails-dev' for the rails development with the command below:

```
create role rails_dev with createdb login password 'aqwe123';
```

Set a new password for the user and check that the user has been created.

Now check the new role and you will see new role has been created:

```
\du
```

The PostgreSQL database has been prepared for the Rails Development.

```
root@asuka:~# su - postgres
postgres@asuka:~# psql
psql (9.5.4)
Type "help" for help.

postgres=# \password postgres
Enter new password:
Enter it again:
postgres=# create role rails_dev with createdb login password 'aqwe123';
CREATE ROLE
postgres=# \du

          List of roles
Role name | Attributes                                     | Member of
-----+-----+-----
postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
rails_dev | Create DB                                         | {}

postgres=#
```

Step 7 - Create the First Application with Rails and PostgreSQL

Ruby on Rails or Rails ships with the command "rails" and we can bootstrap our first application with that command.

Create a new application "myapp" with PostgreSQL as the default database.

```
rails new myapp -d postgresql
```

That command will create a new directory 'myapp' and install new gem including pg gem that is needed by rails to connect to the PostgreSQL database into that directory.

Next, go to the 'myapp' directory and edit the database.yml file in the config directory.

```
cd myapp/
vim config/database.yml
```

In the development section, uncomment line 32 and type the role we've created in step 6.

```
username: rails_dev
```

Set the rails_dev user password on line 35.

```
password: aqwe123
```

Uncomment line 40 and 44 for the database host configuration.

```
host: localhost
port: 5432
```

Now go to the test section and add the new configuration below:

```

database: myapp_test
host: localhost
port: 5432
username: rails_dev
password: aqwe123

```

Save and exit.

Next, generate the database with the rails command:

```

rails db:setup
rails db:migrate

```

And then start the rails server with the command below:

```

rails s -b 192.168.1.110 -p 8080

```

```

root@asuka:~# cd myapp/
root@asuka:~/myapp# vim config/database.yml
root@asuka:~/myapp# rails db:setup
Created database 'myapp_development'
Created database 'myapp_test'
/root/myapp/db/schema.rb doesn't exist yet. Run 'rails db:migrate' to create it, then try again.
g/application.rb to limit the frameworks that will be loaded.
root@asuka:~/myapp# rails db:migrate
root@asuka:~/myapp# rails s -b 192.168.1.110 -p 8080
=> Booting Puma
=> Rails 5.0.0.1 application starting in development on http://192.168.1.110:8080
=> Run 'rails server -h' for more startup options
Puma starting in single mode...
* Version 3.6.0 (ruby 2.3.1-p112), codename: Sleepy Sunday Serenity
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://192.168.1.110:8080
Use Ctrl-C to stop

```

Open your web browser and visit the server IP address on port 8080 - 192.168.1.110:8080.



Back to the terminal and create a new scaffold to interact with the PostgreSQL database.

Type the command below:

```

rails g scaffold Post title:string body:text
rake db:migrate

```

Run the rails server again.

```

rails s -b 192.168.1.110 -p 8080

```

Visit the server : 192.168.1.110:8080/posts

You will see simple CRUD on Ruby on Rails with PostgreSQL.

← → ↻

🔒 192.168.1.110:8080/posts

Posts

Title	Body
Ruby on Rails and PostgreSQL	This is Ruby on Rails with PostgreSQL on Ubuntu 16.04

[Show](#) [Edit](#) [Destroy](#)

[New Post](#)

Reference

- <http://railsapps.github.io/installrubyonrails-ubuntu.html>
- <https://gorails.com/setup/ubuntu/16.04>

 [view as pdf](#) |  [print](#)

Share this page:

Tweet

Follow @howtoforgecom

29K followers

Recommend 27

G+1

3

Suggested articles


2 Comment(s)

Add comment

Name *

Email *

↶ ↷

B *I* 

p

I'm not a robot

reCAPTCHA
Privacy - Terms

Submit comment

Comments

From: LJ Nissen

Reply

This was exactly what I was looking for!

What I found however was that I needed to modify the pg_hba.conf file located at /etc/postgresql/<version>/pg_hba.conf to something like:

local all all trust host all all 127.0.0.1/32 trust host all all ::1/128 trust

and then restart Postgresql with:

\$ /etc/init.d/postgresql restart

From: ylluminate

Reply

How well does this work with Windows 10 and the WSL Ubuntu 16.04 system?

Tutorials

How to Install Ruby on Rails (RoR) with PostgreSQL on Ubuntu 16.04

Sign up now!



Tutorial Info

Author: Muhammad Arul
Tags: linux, programming, server, ubuntu

Share This Page

Tweet Follow 29K followers

Recommend 27

G+ 3

Xenforo skin by Xenfocus

Howtoforge © projektfarm GmbH.

[Contribute](#) [Contact](#) [Help](#) [Imprint](#)

[Terms](#)