

## Ruby on Rails 3.1 инструкция по установке

(с Apache2 and MySQL на Ubuntu Linux )

### 1. УСТАНОВКА RUBY ON RAILS

Ruby on Rails можно устанавливать двумя методами: Первый — от root-а но тогда теряется гибкость контроля использования разных версий Ruby для разных приложений; Второй от пользователя который будет заниматься разработкой на Ruby, в таком случае для разных пользователей можно будет использовать разные версии Ruby.

Мы покажем как устанавливать Ruby для конкретного пользователя.

1. Для начала необходимо получить право суперпользователя (**root**) если вы знаете его пароль или Ваш пользователь должен иметь права на использование команды **sudo**.

2. Подготовка к установке:

1. Выполнить:

```
sudo apt-get update.
```

```
sudo apt-get upgrade.
```

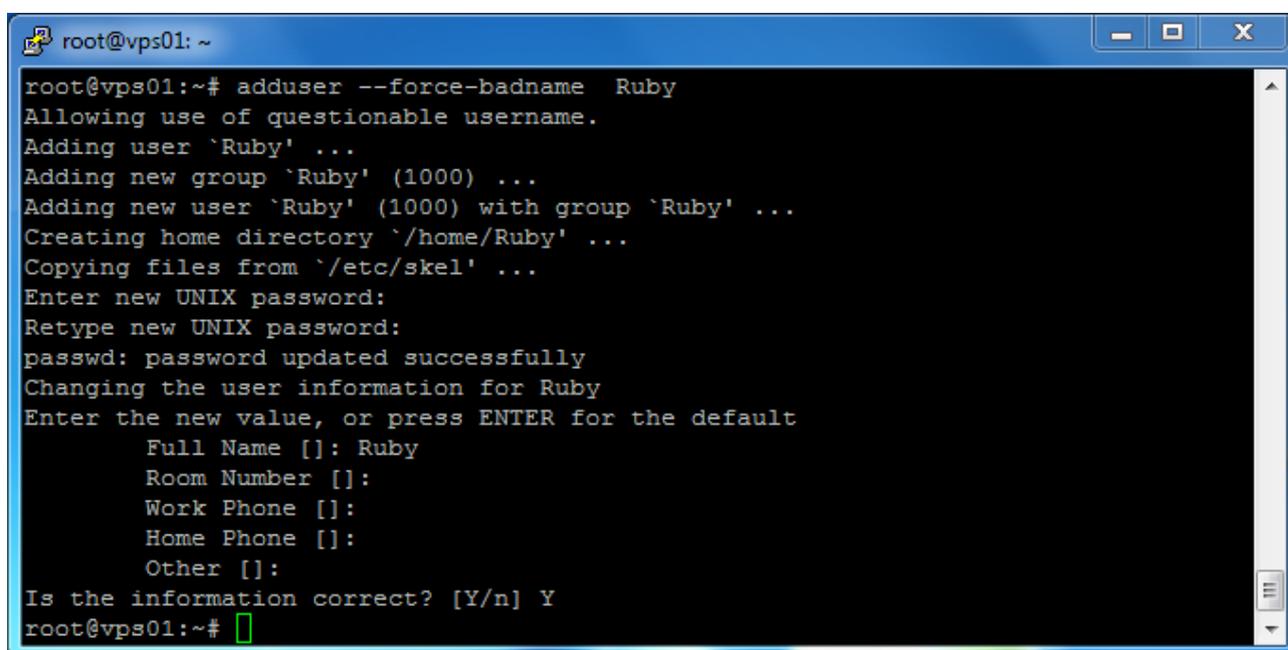
```
sudo apt-get install aptitude (опционально).
```

При необходимости Ubuntu Linux попросит Вас нажать «Y/n» для принятия решения об установке.

2. Создаем пользователя от которого будет осуществляться установка Ruby и создание приложений.

```
sudo adduser --force-badname Ruby
```

далее Ubuntu Linux задаст несколько несложных вопросов и создаст пользователя.



```
root@vps01: ~
root@vps01:~# adduser --force-badname Ruby
Allowing use of questionable username.
Adding user `Ruby' ...
Adding new group `Ruby' (1000) ...
Adding new user `Ruby' (1000) with group `Ruby' ...
Creating home directory `/home/Ruby' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for Ruby
Enter the new value, or press ENTER for the default
    Full Name []: Ruby
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
root@vps01:~#
```

3. Необходимо дать Вашему пользователю права на установку ПО (отредактируйте файл /etc/sudoers).

4. Устанавливаем необходимое для работы rvm (Ruby Version Manager) ПО.

```
cd ~ или cd /home/user
```

```
sudo aptitude install build-essential git-core curl
```

5. Устанавливаем сам RVM:

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

```
type rvm | head -1 (эта команда должна вывести: «rvm is a function»)
```

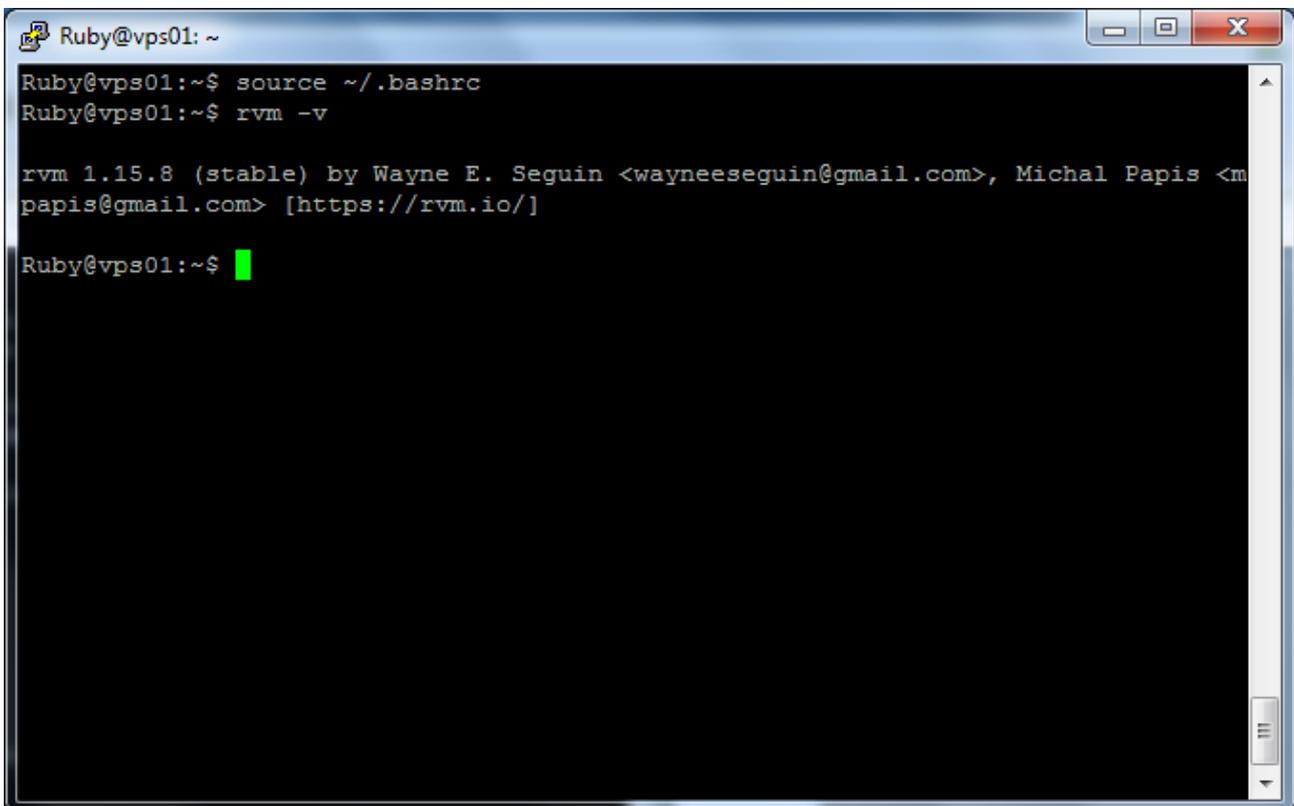
6. Редактируем файл ~/.bashrc командой:

```
echo '[[ -s "$HOME/.rvm/scripts/rvm" ]] && source "$HOME/.rvm/scripts/rvm"' >> ~/.bashrc
```

далее команда `source ~/.bashrc`

7. Для того чтоб узнать версию установленного rvm:

В командной строке набрать: `rvm -v`, команда должна вывести примерно следующее.



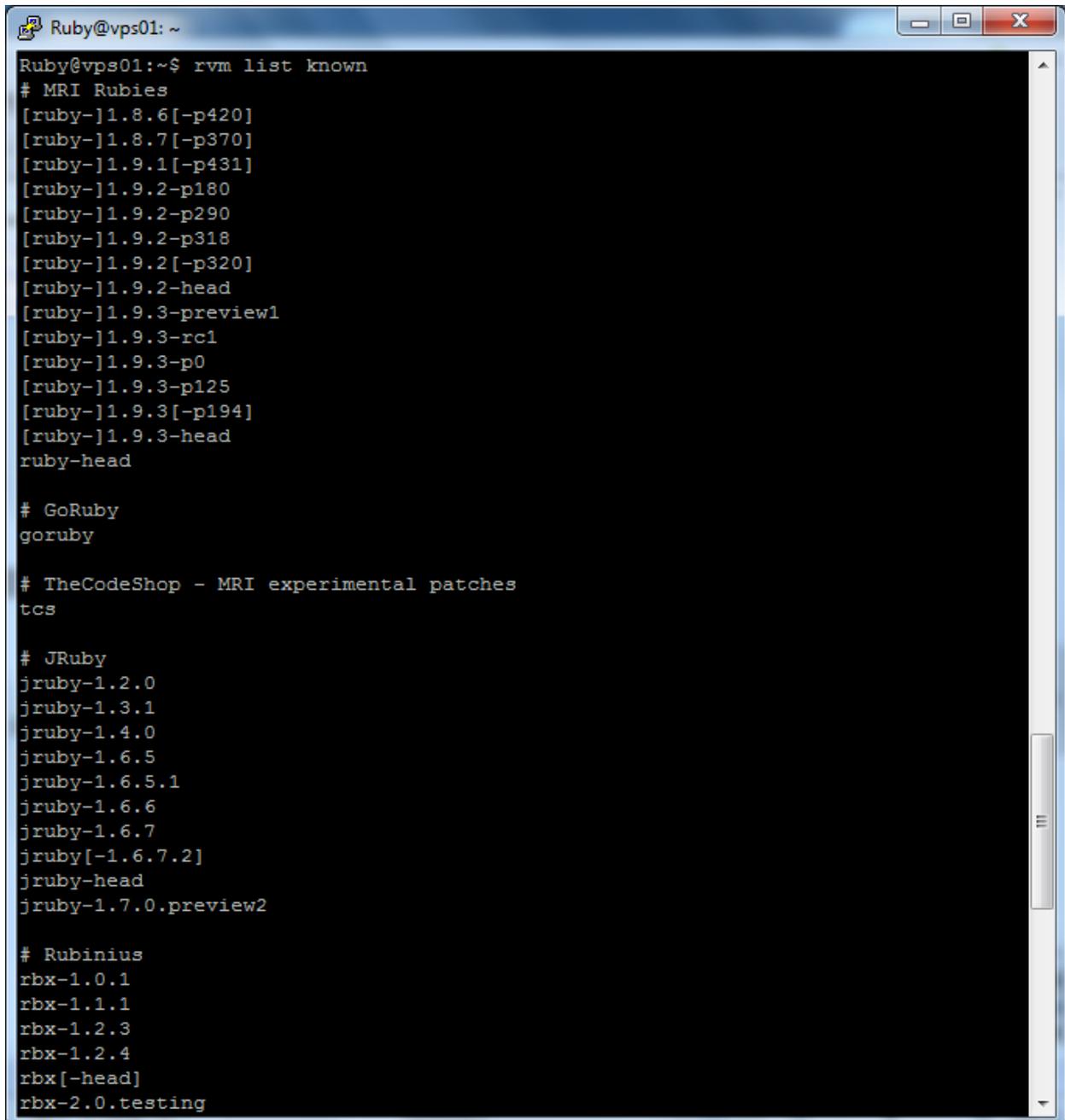
```
Ruby@vps01: ~  
Ruby@vps01:~$ source ~/.bashrc  
Ruby@vps01:~$ rvm -v  
  
rvm 1.15.8 (stable) by Wayne E. Seguin <wayneesequin@gmail.com>, Michal Papis <papis@gmail.com> [https://rvm.io/]  
  
Ruby@vps01:~$ █
```

8. Устанавливаем необходимые для Ruby пакеты:

```
sudo aptitude install build-essential openssl libreadline6 libreadline6-dev zlib1g zlib1g-dev zlib  
libssl-dev libyaml-dev libsqlite3-0 libsqlite3-dev sqlite3 libxml2-dev libxslt-dev autoconf libc6-
```

`dev ncurses-dev automake libtool bison`

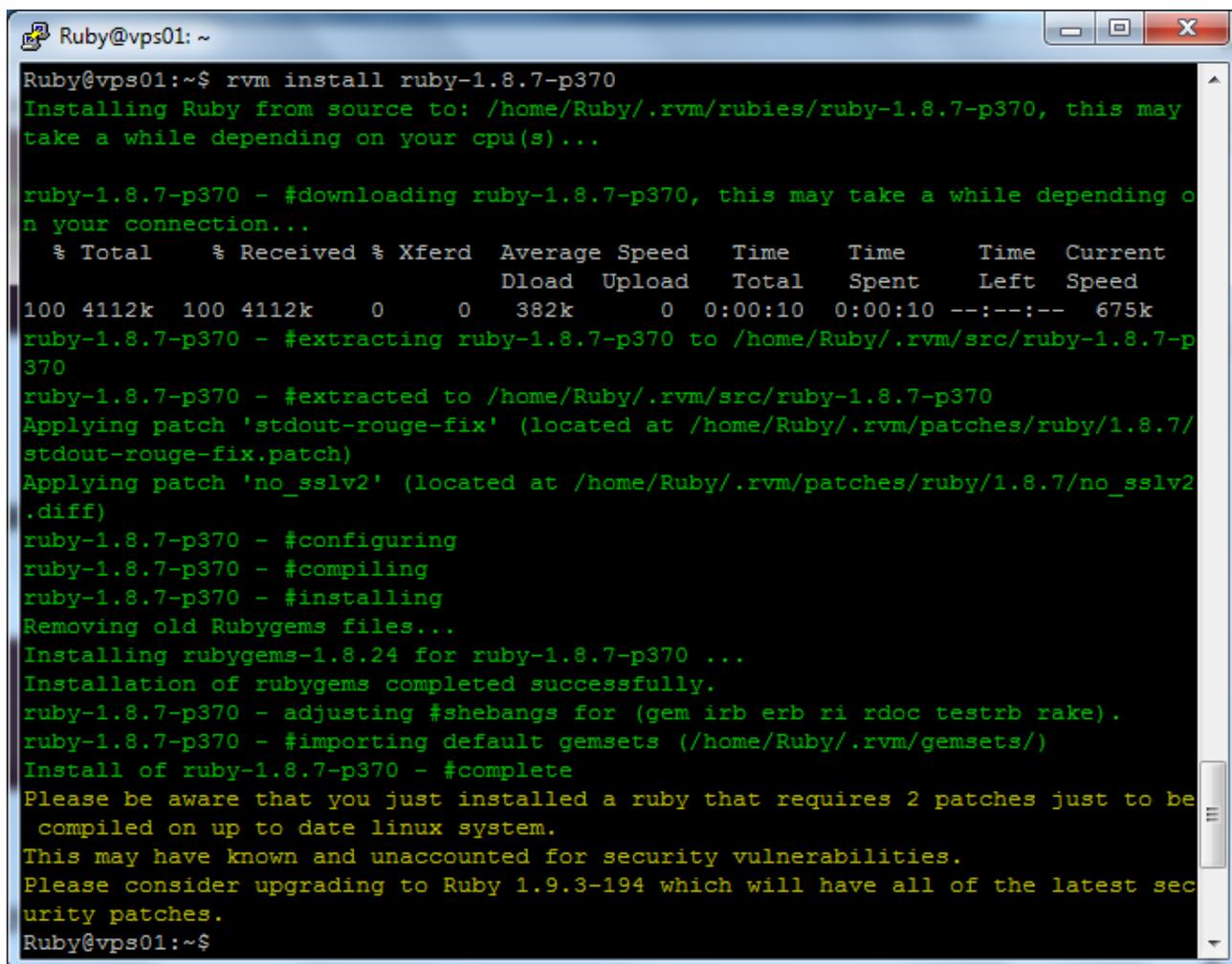
9. Для того чтоб узнать какие версии Ruby доступны для установки необходимо



```
Ruby@vps01: ~  
Ruby@vps01:~$ rvm list known  
# MRI Rubies  
[ruby-]1.8.6[-p420]  
[ruby-]1.8.7[-p370]  
[ruby-]1.9.1[-p431]  
[ruby-]1.9.2-p180  
[ruby-]1.9.2-p290  
[ruby-]1.9.2-p318  
[ruby-]1.9.2[-p320]  
[ruby-]1.9.2-head  
[ruby-]1.9.3-preview1  
[ruby-]1.9.3-rc1  
[ruby-]1.9.3-p0  
[ruby-]1.9.3-p125  
[ruby-]1.9.3[-p194]  
[ruby-]1.9.3-head  
ruby-head  
  
# GoRuby  
goruby  
  
# TheCodeShop - MRI experimental patches  
tcs  
  
# JRuby  
jruby-1.2.0  
jruby-1.3.1  
jruby-1.4.0  
jruby-1.6.5  
jruby-1.6.5.1  
jruby-1.6.6  
jruby-1.6.7  
jruby[-1.6.7.2]  
jruby-head  
jruby-1.7.0.preview2  
  
# Rubinius  
rbx-1.0.1  
rbx-1.1.1  
rbx-1.2.3  
rbx-1.2.4  
rbx[-head]  
rbx-2.0.testing
```

выполнить: `rvm list known`

10. Выполним установку Ruby: `rvm install 1.9.3` и `1.8.7`



```
Ruby@vps01: ~  
Ruby@vps01:~$ rvm install ruby-1.8.7-p370  
Installing Ruby from source to: /home/Ruby/.rvm/rubies/ruby-1.8.7-p370, this may  
take a while depending on your cpu(s) ...  
  
ruby-1.8.7-p370 - #downloading ruby-1.8.7-p370, this may take a while depending on  
your connection ...  
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
                                 Dload  Upload   Total   Spent    Left   Speed  
100 4112k  100 4112k    0     0   382k     0  0:00:10  0:00:10  --:--:--  675k  
ruby-1.8.7-p370 - #extracting ruby-1.8.7-p370 to /home/Ruby/.rvm/src/ruby-1.8.7-p  
370  
ruby-1.8.7-p370 - #extracted to /home/Ruby/.rvm/src/ruby-1.8.7-p370  
Applying patch 'stdout-rouge-fix' (located at /home/Ruby/.rvm/patches/ruby/1.8.7/  
stdout-rouge-fix.patch)  
Applying patch 'no_sslv2' (located at /home/Ruby/.rvm/patches/ruby/1.8.7/no_sslv2  
.diff)  
ruby-1.8.7-p370 - #configuring  
ruby-1.8.7-p370 - #compiling  
ruby-1.8.7-p370 - #installing  
Removing old Rubygems files ...  
Installing rubygems-1.8.24 for ruby-1.8.7-p370 ...  
Installation of rubygems completed successfully.  
ruby-1.8.7-p370 - adjusting #shebangs for (gem irb erb ri rdoc testrb rake).  
ruby-1.8.7-p370 - #importing default gemsets (/home/Ruby/.rvm/gemsets/)  
Install of ruby-1.8.7-p370 - #complete  
Please be aware that you just installed a ruby that requires 2 patches just to be  
compiled on up to date linux system.  
This may have known and unaccounted for security vulnerabilities.  
Please consider upgrading to Ruby 1.9.3-194 which will have all of the latest sec  
urity patches.  
Ruby@vps01:~$
```

11. Выполним следующие команды для установки версии руби которую мы будем использовать:

1. Узнаем какие версии Ruby установлены: `rvm list`
2. Выберем для использования версию 1.9.3: `rvm use 1.9.3`
3. Установим выбранную версию используемой по умолчанию:

`rvm list`

`rvm use 1.9.3`

`rvm -default use 1.9.3`

`rvm list`

`ruby -v`

```
Ruby@vps01: ~
Ruby@vps01:~$ rvm list

rvm rubies

=> ruby-1.8.7-p370 [ x86_64 ]
* ruby-1.9.3-p194 [ x86_64 ]

# => - current
# =* - current && default
# * - default

Ruby@vps01:~$ rvm use 1.9.3
Using /home/Ruby/.rvm/gems/ruby-1.9.3-p194
Running /home/Ruby/.rvm/hooks/after_use
Ruby@vps01:~$ rvm --default use 1.9.3
Using /home/Ruby/.rvm/gems/ruby-1.9.3-p194
Running /home/Ruby/.rvm/hooks/after_use
Ruby@vps01:~$ rvm list

rvm rubies

    ruby-1.8.7-p370 [ x86_64 ]
=*   ruby-1.9.3-p194 [ x86_64 ]

# => - current
# =* - current && default
# * - default

Ruby@vps01:~$ ruby -v
ruby 1.9.3p194 (2012-04-20 revision 35410) [x86_64-linux]
Ruby@vps01:~$
```

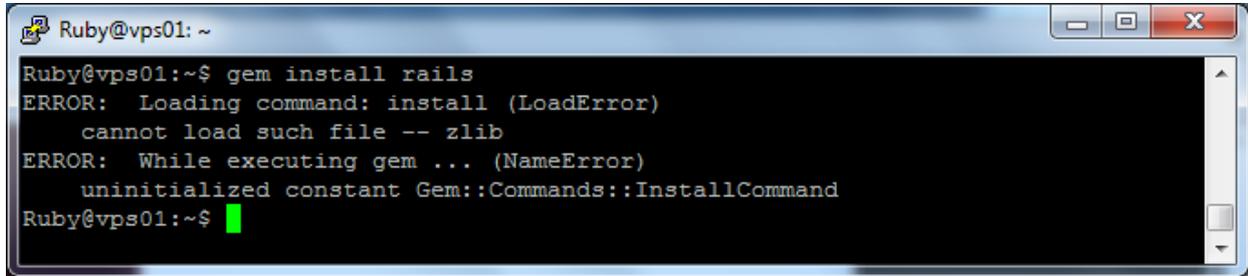
12. Установим Rails «рельсы» **ВНИМАНИЕ! Не используйте sudo!**

gem install rails

```
Ruby@vps01: ~
Ruby@vps01:~$ gem install rails
Fetching: i18n-0.6.0.gem (100%)
Fetching: multi_json-1.3.6.gem (100%)
Fetching: activesupport-3.2.8.gem (100%)
Fetching: builder-3.0.0.gem (100%)
Fetching: activemodel-3.2.8.gem (100%)
Fetching: rack-1.4.1.gem (100%)
Fetching: rack-cache-1.2.gem (100%)
Fetching: rack-test-0.6.1.gem (100%)
Fetching: journey-1.0.4.gem (100%)
Fetching: hike-1.2.1.gem (100%)
Fetching: tilt-1.3.3.gem (100%)
Fetching: sprockets-2.1.3.gem (100%)
Fetching: erubis-2.7.0.gem (100%)
Fetching: actionpack-3.2.8.gem (100%)
Fetching: arel-3.0.2.gem (100%)
Fetching: tzinfo-0.3.33.gem (100%)
Fetching: activerecord-3.2.8.gem (100%)
Fetching: activerecord-3.2.8.gem (100%)
Fetching: mime-types-1.19.gem (100%)
Fetching: polyglot-0.3.3.gem (100%)
Fetching: treetop-1.4.10.gem (100%)
Fetching: mail-2.4.4.gem (100%)
Fetching: actionmailer-3.2.8.gem (100%)
Fetching: rack-ssl-1.3.2.gem (100%)
Fetching: thor-0.16.0.gem (100%)
Fetching: json-1.7.5.gem (100%)
Building native extensions. This could take a while...
Fetching: rdoc-3.12.gem (100%)
Depending on your version of ruby, you may need to install ruby rdoc/ri data:

<= 1.8.6 : unsupported
= 1.8.7 : gem install rdoc-data; rdoc-data --install
= 1.9.1 : gem install rdoc-data; rdoc-data --install
>= 1.9.2 : nothing to do! Yay!
Fetching: railties-3.2.8.gem (100%)
Fetching: rails-3.2.8.gem (100%)
Successfully installed i18n-0.6.0
```

Если Вы получили ошибку: Смотрите приложение 1 пункт 1



```
Ruby@vps01: ~  
Ruby@vps01:~$ gem install rails  
ERROR: Loading command: install (LoadError)  
cannot load such file -- zlib  
ERROR: While executing gem ... (NameError)  
uninitialized constant Gem::Commands::InstallCommand  
Ruby@vps01:~$
```

Узнать версию рельсов: `rails -v`



```
Ruby@vps01: ~  
Ruby@vps01:~$ rails -v  
Rails 3.2.8  
Ruby@vps01:~$
```

**На этом все Ruby on Rails установлено!**

## 2. СОЗДАНИЕ ТЕСТОВОГО ПРЕКТА, УСТАНОВКА APACHE2+MYSQL

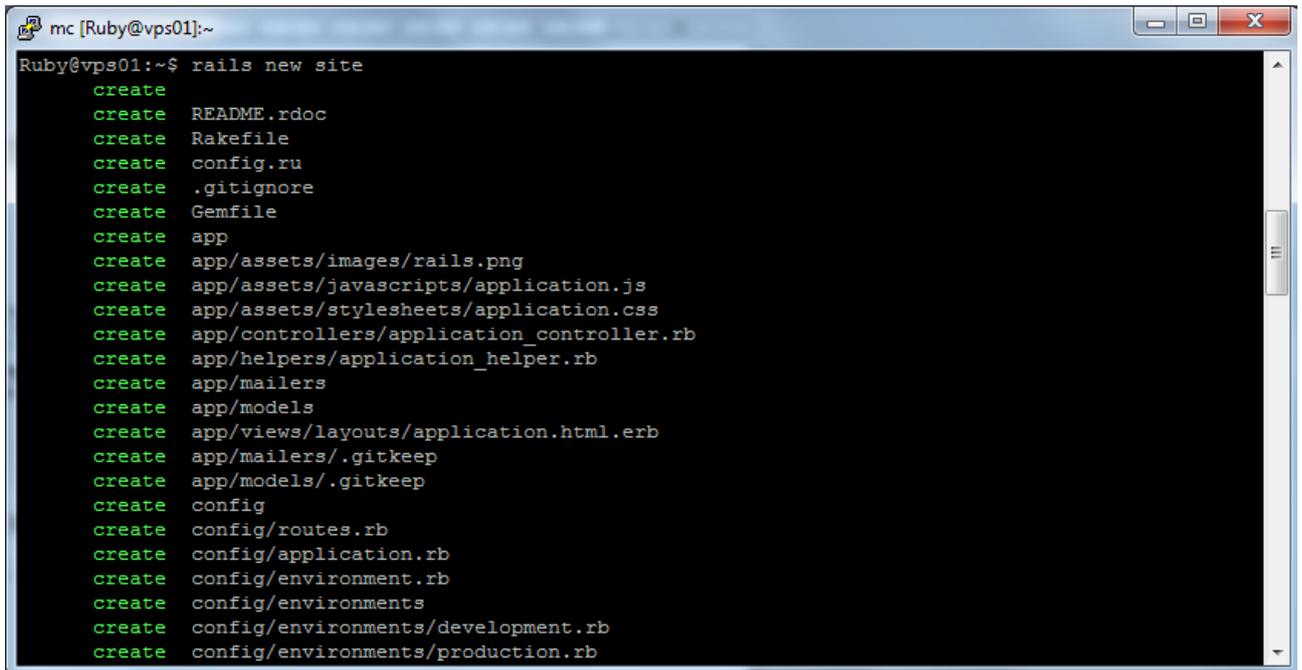
1. В домашней директории пользователя создайте каталог в котором будет храниться Ruby проект.

```
cd /home/Ruby
```

```
mkdir site
```

2. Создает Ruby проект:

```
rails new site
```



```
mc [Ruby@vps01]:~  
Ruby@vps01:~$ rails new site  
create  
create README.rdoc  
create Rakefile  
create config.ru  
create .gitignore  
create Gemfile  
create app  
create app/assets/images/rails.png  
create app/assets/javascripts/application.js  
create app/assets/stylesheets/application.css  
create app/controllers/application_controller.rb  
create app/helpers/application_helper.rb  
create app/mailers  
create app/models  
create app/views/layouts/application.html.erb  
create app/mailers/.gitkeep  
create app/models/.gitkeep  
create config  
create config/routes.rb  
create config/application.rb  
create config/environment.rb  
create config/environments  
create config/environments/development.rb  
create config/environments/production.rb
```

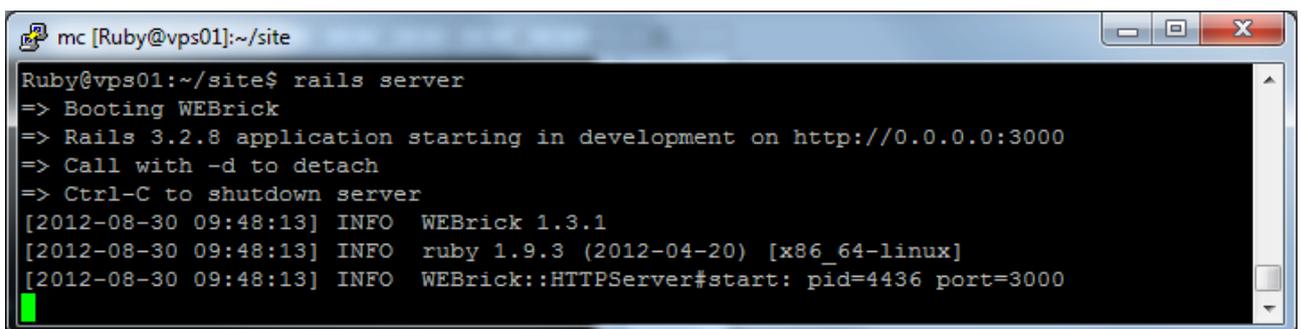
`cd site`

`bundle install`

3. Проверим работоспособность Rails.

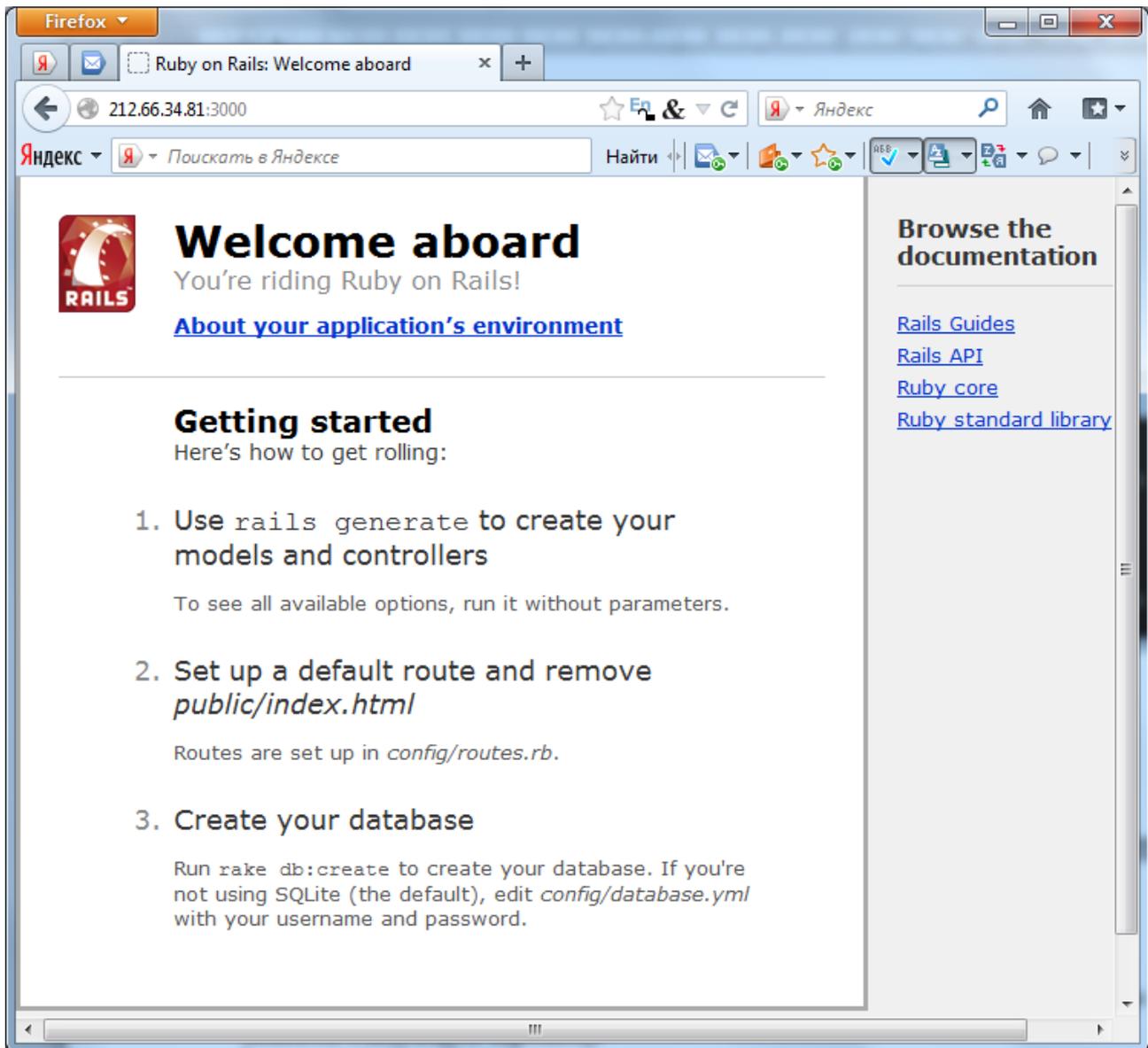
Для этого запустим встроенный Rails сервер:

`rails server` или `rails s`

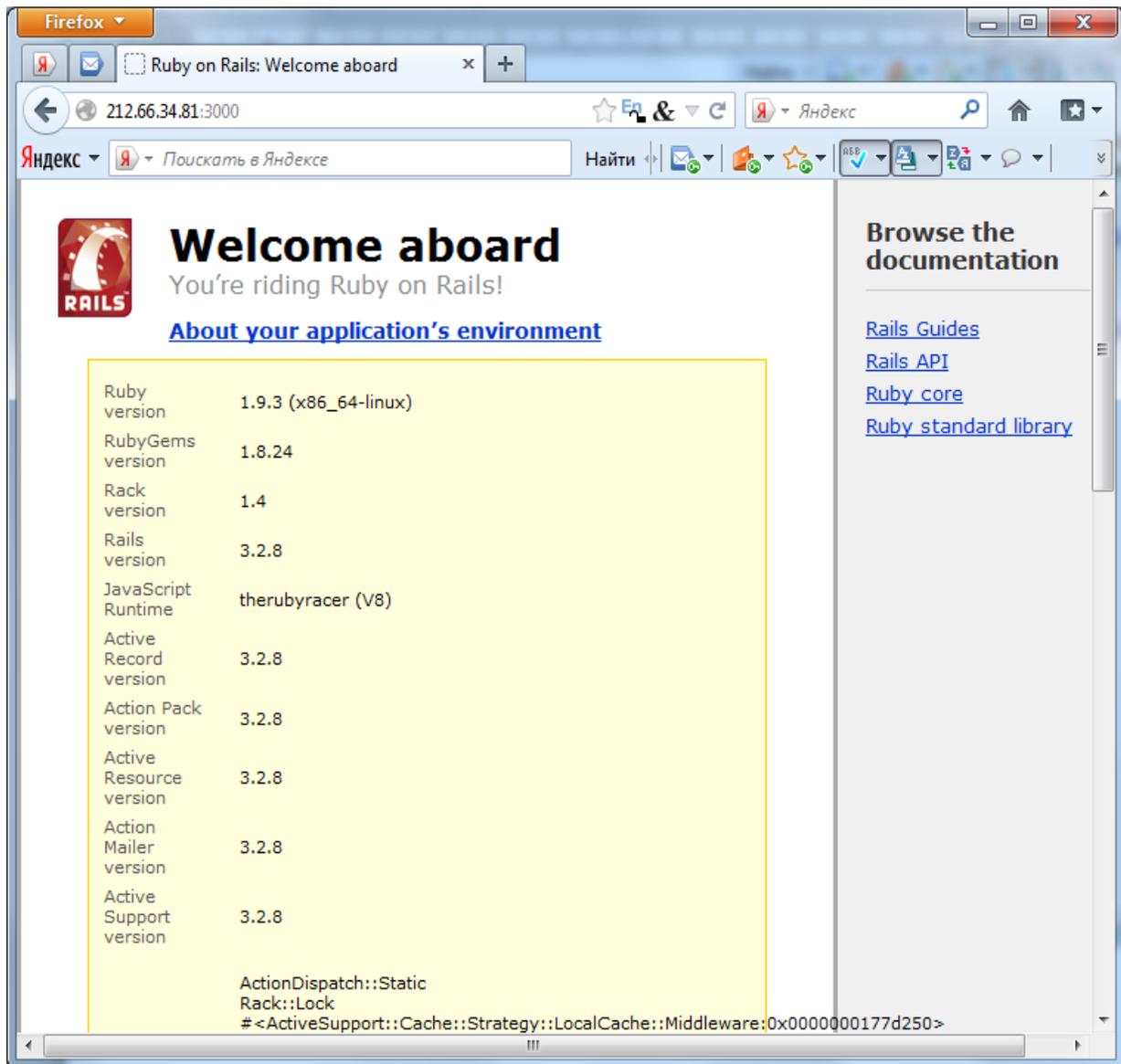


```
mc [Ruby@vps01]:~/site  
Ruby@vps01:~/site$ rails server  
=> Booting WEBrick  
=> Rails 3.2.8 application starting in development on http://0.0.0.0:3000  
=> Call with -d to detach  
=> Ctrl-C to shutdown server  
[2012-08-30 09:48:13] INFO WEBrick 1.3.1  
[2012-08-30 09:48:13] INFO ruby 1.9.3 (2012-04-20) [x86_64-linux]  
[2012-08-30 09:48:13] INFO WEBrick::HTTPServer#start: pid=4436 port=3000
```

После запуска **rails server** заходим по адресу **http://ip-adr\_of\_server:3000** и вы должны увидеть следующую картинку.



Перейдите по ссылке [About your application's environment](#) для получения сведения о Ваших переменных окружения созданного приложения.



Если при запуске выдало ошибку **ExecJS::RuntimeUnavailable**. Смотрите **Приложение 1 пункт 2**

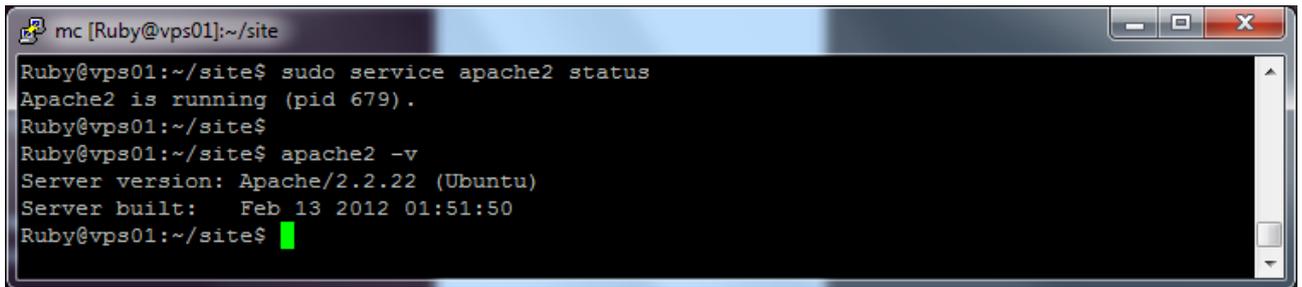
```
mc [Ruby@vps01]:~  
Ruby@vps01:~/site$ rails server  
/home/Ruby/.rvm/gems/ruby-1.9.3-p194/gems/execjs-1.4.0/lib/execjs/runtimes.rb:51:in `autodetect':  
ld not find a JavaScript runtime. See https://github.com/sstephenson/execjs for a list of availabl  
untimes. (ExecJS::RuntimeUnavailable)  
    from /home/Ruby/.rvm/gems/ruby-1.9.3-p194/gems/execjs-1.4.0/lib/execjs.rb:5:in `<module:Ex  
S>'  
    from /home/Ruby/.rvm/gems/ruby-1.9.3-p194/gems/execjs-1.4.0/lib/execjs.rb:4:in `<top (requ  
d)>'
```

#### 4. Установка Apache 2.

`sudo aptitude install apache2`

Чтоб узнать запущен ли Apache: `sudo service apache2 status/start/stop`

Для того чтоб узнать версию установленного Apache: `apache2 -v`



```
mc [Ruby@vps01]:~/site
Ruby@vps01:~/site$ sudo service apache2 status
Apache2 is running (pid 679).
Ruby@vps01:~/site$
Ruby@vps01:~/site$ apache2 -v
Server version: Apache/2.2.22 (Ubuntu)
Server built:   Feb 13 2012 01:51:50
Ruby@vps01:~/site$
```

5. После установки необходимо обновит набор gem-ов

`gem update`

6. Устанавливаем пассажира (Passanger) .

`gem install passenger`

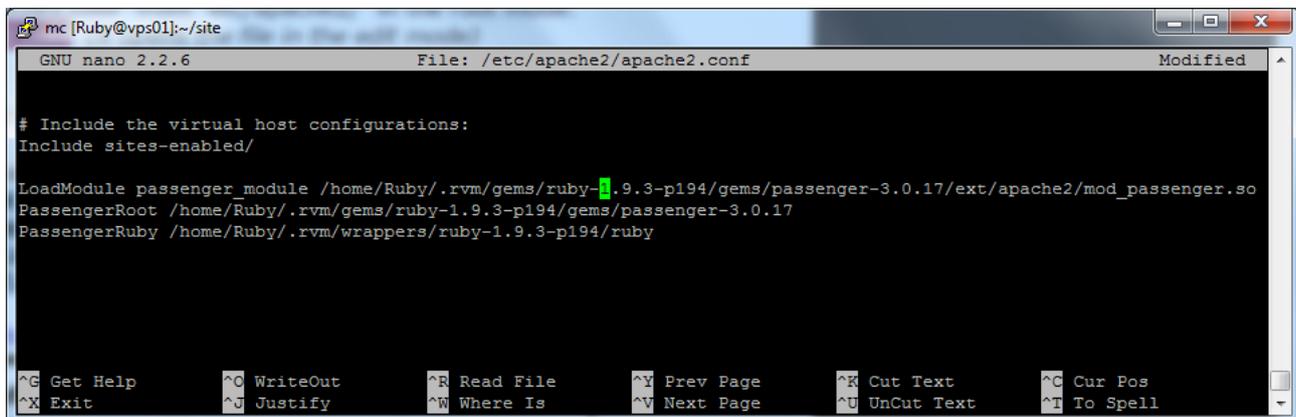
`passenger -v`

7. Установим Apache2 модуль.

`passenger-install-apache2-module`

если для установки `passenger-install-apache2-module` будет не хватать каких либо компонентов программа установки сообщит как их установить.

8. После установки необходимо отредактировать от пользователя root файл конфигурации Apache: `sudo nano /etc/apache2/apache2.conf`.

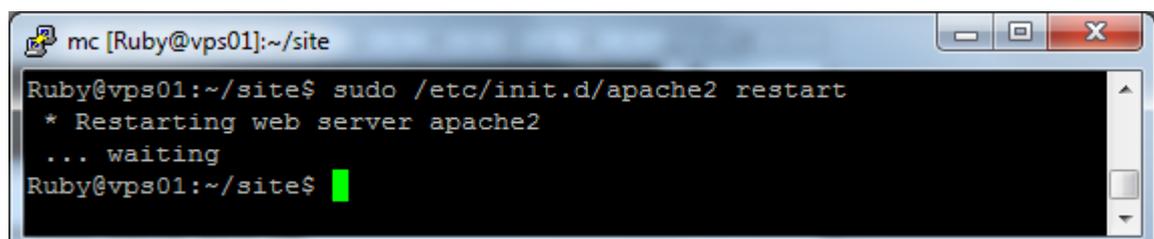


```
GNU nano 2.2.6 File: /etc/apache2/apache2.conf Modified
# Include the virtual host configurations:
Include sites-enabled/

LoadModule passenger_module /home/Ruby/.rvm/gems/ruby-1.9.3-p194/gems/passenger-3.0.17/ext/apache2/mod_passenger.so
PassengerRoot /home/Ruby/.rvm/gems/ruby-1.9.3-p194/gems/passenger-3.0.17
PassengerRuby /home/Ruby/.rvm/wrappers/ruby-1.9.3-p194/ruby

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit          ^U Justify      ^W Where Is    ^V Next Page    ^U UnCut Text  ^T To Spell
```

9. Перезапустить демон Apache2: `sudo /etc/init.d/apache2 restart` или `sudo service apache2 restart`



```
mc [Ruby@vps01]:~/site
Ruby@vps01:~/site$ sudo /etc/init.d/apache2 restart
* Restarting web server apache2
... waiting
Ruby@vps01:~/site$
```

10. Создадим файл /etc/apache2/sites-available/site

```
sudo nano /etc/apache2/sites-available/site
```

где Site – это имя вашей папки с приложением.

```
<VirtualHost *:80>
  ServerName localhost
  ServerAlias localhost
  DocumentRoot /home/user/site/public
</VirtualHost>
```

Включаем виртуальный хост: `sudo a2ensite site`

Включаем mod rewrite в Apache: `sudo a2enmod rewrite`

11. Установка Mysql:

```
sudo aptitude install mysql-server mysql-client mysql-admin libmysqlclient-dev
```

во время установки инсталлятор попросит ввести пароль пользователя root mysql.

```
sudo service mysql stop
sudo service mysql start
```

12. Устанавливаем ruby библиотеку для mysql

```
sudo aptitude install libmysql-ruby
```

13. Переходим в папку с проектом у устанавливаем gem для mysql.

```
gem install mysql2
```

Проверка:

```
irb ruby-1.9.2-p290 :003 >
      require 'mysql2' <нажать Enter>
      => true
```

`exit` для выхода

14. По умолчанию Ruby проект использует sqlite базу данных для того чтоб указать другую необходимо создать приложение следующей командой:

```
rails new site -d mysql
```

Если Вы уже создали приложение с sqlite то вы можете изменить используемую по умолчанию базу просто перейдя в директорию /home/Ruby и выполнить указанную выше команду, при этом инсталлятор задаст несколько вопросов.

Далее Вам необходимо отредактировать файл

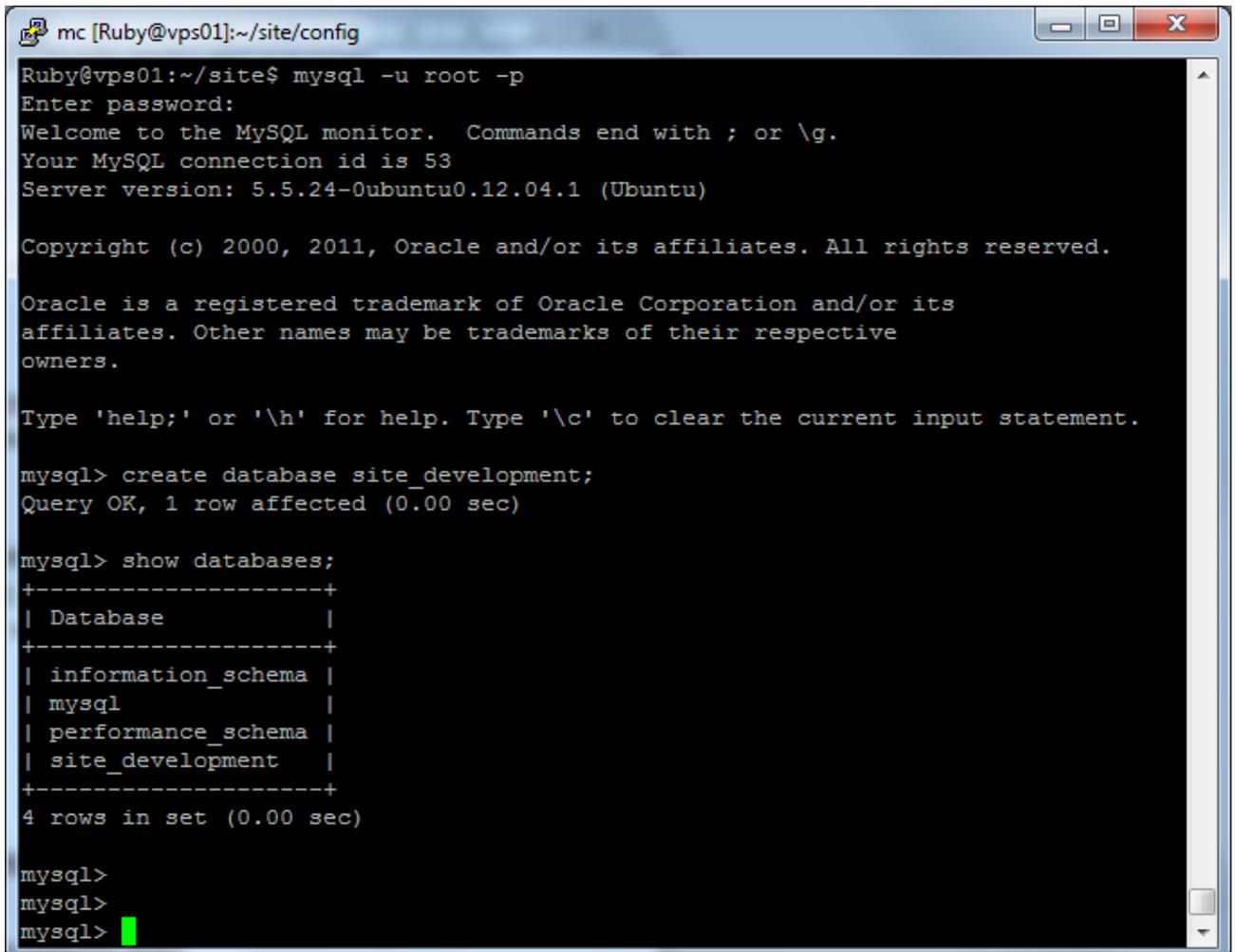
```
/home/Ruby/sait/config/database.yml указав в ней свой пароль от пользователя root mysql.
```

15. Создание базы в Mysql:

```
cd /home/Ruby/site
```

```
sudo mysql -u root -p (после этого введите свой пароль в mysql)
```

```
mysql> create database site_development;
```



```
mc [Ruby@vps01]:~/site/config
Ruby@vps01:~/site$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 53
Server version: 5.5.24-0ubuntu0.12.04.1 (Ubuntu)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database site_development;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| performance_schema      |
| site_development        |
+-----+
4 rows in set (0.00 sec)

mysql>
mysql>
mysql>
```

Для того чтоб узнать создана ли база данных введите команду mysql:

```
show databases;
```

#### 16. Создание тестового приложения Ruby on Rails+Apache+Mysql

```
rails generate scaffold products title:string description:text price:decimal
quantity:integer starts_at:datetime ends_at:datetime
```

```
rake db:migrate
```

```
mc [Ruby@vps01]:~/site
Ruby@vps01:~/site$ rails generate scaffold products title:string description:text price:decimal quantity:integer starts_at:datetime ends_at:datetime
Plural version of the model detected, using singularized version. Override with --force-plural.
  invoke  active_record
  create  db/migrate/20120830092226_create_products.rb
  create  app/models/product.rb
  invoke  test_unit
  create  test/unit/product_test.rb
  create  test/fixtures/products.yml
  invoke  resource_route
  route  resources :products
  invoke  scaffold_controller
  create  app/controllers/products_controller.rb
  invoke  erb
  create  app/views/products
  create  app/views/products/index.html.erb
  create  app/views/products/edit.html.erb
  create  app/views/products/show.html.erb
  create  app/views/products/new.html.erb
  create  app/views/products/_form.html.erb
  invoke  test_unit
  create  test/functional/products_controller_test.rb
  invoke  helper
  create  app/helpers/products_helper.rb
  invoke  test_unit
  create  test/unit/helpers/products_helper_test.rb
  invoke  assets
  invoke  coffee
  create  app/assets/javascripts/products.js.coffee
  invoke  scss
  create  app/assets/stylesheets/products.css.scss
  invoke  scss
  create  app/assets/stylesheets/scaffolds.css.scss
Ruby@vps01:~/site$ rake db:migrate
== CreateProducts: migrating =====
-- create_table(:products)
   -> 0.0925s
== CreateProducts: migrated (0.0926s) =====
Ruby@vps01:~/site$
```

17. Далее зайдите на сайт.

## ПРИЛОЖЕНИЕ 1

### 1. **no such file to load — zlib:**

`rm pkg install zlib` - после этого Вам необходимо переустановить все Ваши версии Ruby командами:

```
rm reinstall $(rm list strings | tr "\n" ',')
```

после этого Вам необходимо опять выполнить действия из пункта 11 основной установки. Далее делаем:

```
rm all-gemsets do rm gemset pristine
```

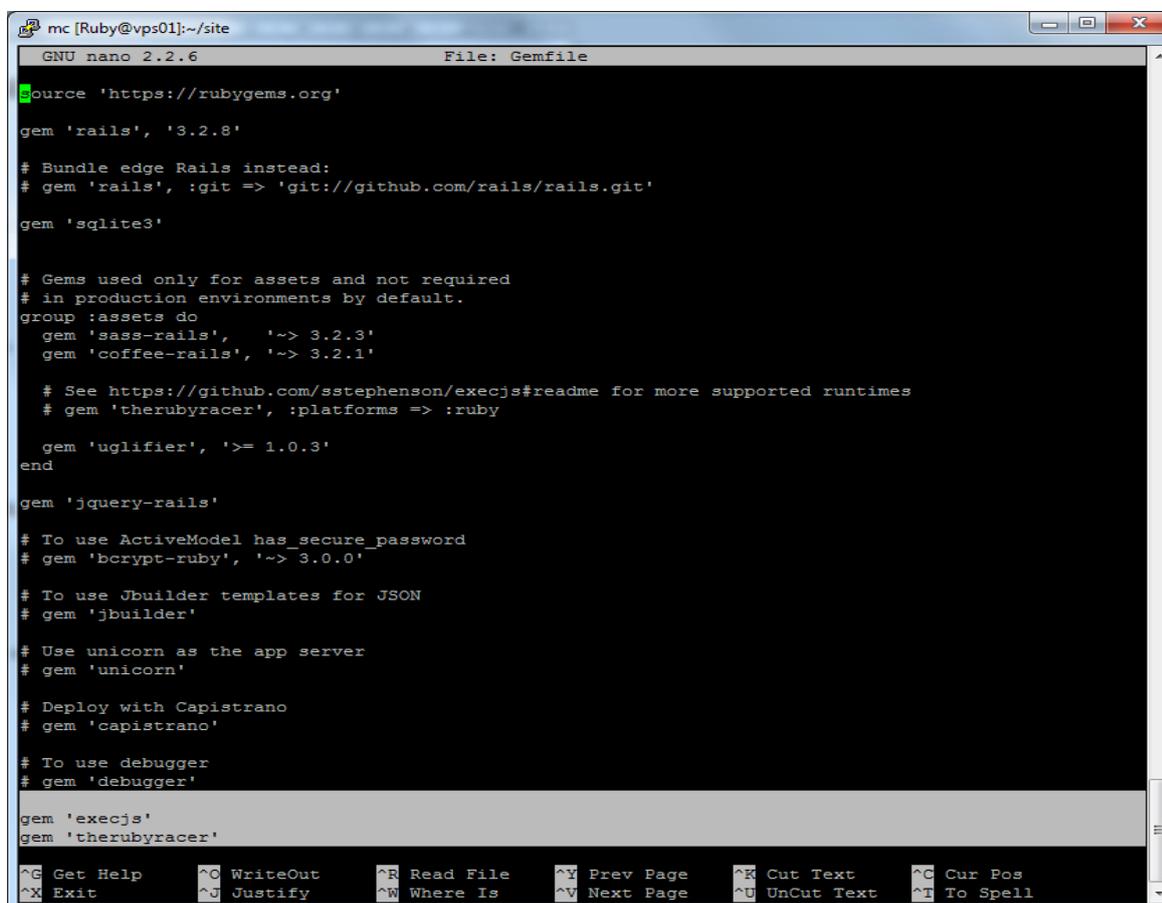
### 2. **ExecJS::RuntimeUnavailable:**

Необходимо отредактировать Gemfile вашего приложения и добавить в него необходимые гемы:

```
gem 'execjs'
```

```
gem 'therubyracer'
```

После сохранения GemFile необходимо выполнить **bundle install**.



```
mc [Ruby@vps01]:~/site
GNU nano 2.2.6 File: Gemfile
source 'https://rubygems.org'

gem 'rails', '3.2.8'

# Bundle edge Rails instead:
# gem 'rails', :git => 'git://github.com/rails/rails.git'

gem 'sqlite3'

# Gems used only for assets and not required
# in production environments by default.
group :assets do
  gem 'sass-rails', '~> 3.2.3'
  gem 'coffee-rails', '~> 3.2.1'

  # See https://github.com/sstephenson/execjs#readme for more supported runtimes
  # gem 'therubyracer', :platforms => :ruby

  gem 'uglifyer', '>= 1.0.3'
end

gem 'jquery-rails'

# To use ActiveRecord has_secure_password
# gem 'bcrypt-ruby', '~> 3.0.0'

# To use Jbuilder templates for JSON
# gem 'jbuilder'

# Use unicorn as the app server
# gem 'unicorn'

# Deploy with Capistrano
# gem 'capistrano'

# To use debugger
# gem 'debugger'

gem 'execjs'
gem 'therubyracer'
```