How To Setup a Rails 4 App With Apache and Passenger on CentOS 6

Authored by: ***ASPHostServer Administrator*** *[asphostserver@gmail.com]*
Saved From: *http://faq.asphosthelpdesk.com/article.php?id=151*

When the user creates a server instance from a vanilla image (in our case: CentOS 6.4 x64), additional system administration work is needed before a Ruby on Rails application can be run and become available online. This tutorial provides the necessary steps to make that happen.

In order to go through this tutorial, a virtual server with at least 1 GB of RAM is a minimum requirement, because Passenger commands in step #3 are memory intensive.

Also, the tutorial assumes that the user has root access to the VPS (either as a user with root privileges or as the system root).

This tutorial is not intended for production node setup. It involves development library installation and ad hoc compilation on the machine, which is not a good practice when you're setting up a production environment."

Step One â€“ Apache Setup

It all starts with the web server and the simplest way to install Apache is to pull it from the yum repository:

yum install httpd

After the installation is done, the system should be configured to automatically run Apache on system boot:

chkconfig httpd on

Without this setting, the httpd service needs to be started manually each time the server instance is rebooted.

The next step is to configure Apache to expect incoming requests by editing its configuration file:

nano /etc/httpd/conf/httpd.conf

In the editor, near the bottom, locate and uncomment the line containing: NameVirtualHost *:80

Save and exit the editor. Apache is ready to be started now:

service httpd start

VoilÃ ! Accessing server instance"s IP address in a web browser will show Apache"s welcome page.

Step Two â€“ Ruby 2.0 and Rails 4.0 Setup

Setting up Apache was an easy stroll. It"s even simpler to setup Ruby, followed by Rails.

The quickest way is to use Ruby Version Manager (RVM) to maintain multiple Ruby environments on the machine. The latest stable version of RVM is downloaded and installed like this:

curl -L get.rvm.io | bash -s stable

If RVM is supposed to be used by all users, there is a shell script available for the job:

source /etc/profile.d/rvm.sh

Additionally, installing RVM"s requirements is necessary and this command will install various development packages and dependencies:

rvm requirements

At this point, RVM is fully initialized and ready for use.

Currently, the latest Ruby version is 2.0.0 and the corresponding RVM command that installs it is:

rvm install 2.0.0

It will download and install the Ruby environment. After the installation and even though it is the only version installed at this point, it should be configured as the default environment:

rvm use 2.0.0 --default

Rails is distributed as a Ruby gem and adding it to the local system is extremely simple:

gem install rails

A quick version check will yield output similar to this one:

[user@server ~]$ ruby -v

ruby 2.0.0p353 (2013-11-22 revision 43784) [x86_64-linux]

[user@server ~]$ rails -v

Rails 4.0.1

Nice job!

Step Three â€“ Phusion Passenger Setup

Phusion Passenger (commonly shortened to Passenger or referred to as mod_passenger) is an application server and it is often used to power Ruby sites. Its code is distributed in form of a Ruby gem, which is then compiled on the target machine and installed into Apache as a module.

First, the gem needs to be installed on the system:

gem install passenger

Before the Apache module is compiled, two dependency packages need to be installed as well:

yum install curl-devel httpd-devel

The environment is now ready for the compilation. The process takes a few minutes and it"s started by the following command:

passenger-install-apache2-module

Note that this script will not install the module really. It will compile module"s binary and place it under gem"s path. The path will be printed on screen and it needs to be copy-pasted into Apache"s config file (/etc/httpd/conf/httpd.conf) manually.

The output will be similar to this one:

LoadModule passenger_module /usr/local/rvm/gems/ruby-2.0.0-p353/gems/passenger-

4.0.26/buildout/apache2/mod_passenger.so

PassengerRoot /usr/local/rvm/gems/ruby-2.0.0-p353/gems/passenger-4.0.26

PassengerDefaultRuby /usr/local/rvm/wrappers/ruby-2.0.0-p353/ruby

Placing those lines at the bottom of the file will do just fine. Save the change and restart Apache:

service httpd restart

Two green OK messages mark completion of the environment setup. Congrats!

Finally â€“ A Rails App Example

It"s always good to do a smoke test and using Rails" app skeleton generator is an ideal tool for that task. It requires sqlite"s development package to be installed:

yum install sqlite-devel

The simplest location to place the test application code is Apache"s web root folder:

cd /var/www/html

In this folder, test application code is generated under the "helloappâ€• folder:

rails new helloapp

cd helloapp

Since CentOS doesn"t come with a JavaScript execution environment, one must be installed manually. It can be installed as a gem, hence it simply needs to be listed in Gemfile as follows:

gem 'therubyracer'

Gems required by the Ruby application are installed by positioning in application''s root and running the bundle installer:

bundle install

Also, the development database needs to be initialized by the migration tool:

rake db:migrate

One final visit to Apache''s config file (/etc/httpd/conf/httpd.conf) is needed. A virtual host section is missing and it must be added at the bottom:

RackEnv development

RackEnv development

<VirtualHost *:80>

ServerName www.yourhost.com

!!! Be sure to point DocumentRoot to 'public'!

DocumentRoot /var/www/html/helloapp/public

<Directory /var/www/html/helloapp/public>

This relaxes Apache security settings.

AllowOverride all

MultiViews must be turned off.

Options -MultiViews

</Directory>

</VirtualHost>

Save the change and restart Apache once again:

service httpd restart

Go to the virtual server''s IP address in your browser and you should see the "Welcome aboard â€“ You''re riding Ruby on Rails!â€• welcome message.