

How To Install Rails, Apache, and MySQL on Ubuntu with Passenger

Authored by: **ASPHostServer Administrator** [asphostserver@gmail.com]

Saved From: <http://faq.asphosthelpdesk.com/article.php?id=152>

Intro

Installing Ruby on Rails and MySQL on an apache virtual private server is the first step toward getting Ruby applications live and online. Three useful installers make the task of building this server easier than ever before.

Setup

This tutorial requires you to have a server up and running. Additionally, the rails ready script needs to be performed by a user with sudo privileges. If you don't have a user like that on your server, you can check out how to do that in steps 3 and 4 of this tutorial.

Step One—Install Rails Ready

Once you are logged in on your virtual server with your user with root privileges, type in the command to install Rails Ready:

```
wget --no-check-certificate
https://raw.githubusercontent.com/joshfng/railsready/master/railsready.sh && bash
railsready.sh
```

Rails Ready can be installed either from the source or with RVM, the Ruby Version Manager. I would recommend using RVM—it's an easy installation and will later let you to switch between multiple versions of Ruby if needed.

Overall the installation does take a while—be prepared to wait.

However, once Rails Ready finishes the process, your server will be fully equipped with Ruby, Gems, and Rails. Then, following the instructions on screen, "logout and back in to access Ruby"

Step Two—Install Apache with Phusion Passenger

Once RVM is set up, you can use it to install rails:

```
rvm install 1.9.3
```

And set RVM to use Ruby 1.9.3 by default:

```
rvm use --default 1.9.3
```

Then install the passenger gem:

```
gem install passenger
```

As a useful bonus, RailsReady comes packaged with Phusion Passenger, which we can use to then automatically install and configure Apache on our server.

Use this command to start the apache installation:

```
passenger-install-apache2-module
```

Step Three—Update the Apache configuration

Passenger will display this text after Apache installs:

```
The Apache 2 module was successfully installed.  
Please edit your Apache configuration file, and add these lines:  
  LoadModule passenger_module
```

```
/home/username/.rvm/gems/ruby-1.9.3-p194/gems/passenger-3.0.12/ext/apache2/mod_passenger.so  
  PassengerRoot /home/username/.rvm/gems/ruby-1.9.3-p194/gems/passenger-3.0.12  
  PassengerRuby /home/username/.rvm/wrappers/ruby-1.9.3-p194/ruby
```

To finish the process, open up the Apache config and paste the three required lines into the file:

```
sudo nano /etc/apache2/apache2.conf
```

Save and Exit.

Step Four—Install MySQL

Before we conclude the installation, we should add one more useful program to our virtual server.

MySQL is a powerful database management system used for organizing and retrieving data.

To install MySQL, open terminal and type in these commands:

```
sudo aptitude update  
sudo aptitude install mysql-server
```

During the installation, MySQL will ask you to set a root password. If you miss the chance to set the password while the program is installing, it is very easy to set the password later from within the MySQL shell with this command:

```
UPDATE mysql.user SET Password = PASSWORD('password') WHERE User = 'root';
```

Congratulations! With the help of three useful installers, we now have Ruby on Rails, Apache, and MySQL on our Ubuntu server!