# Deployment to AWS EC2 Ubuntu instance

## Step 1. Install necessary libs

Install necessary libs

$ sudo apt-get update

$ sudo apt-get install build-essential libssl-dev libyaml-dev libreadline-dev openssl curl git-core zlib1g-dev bison libxml2-dev libxslt1-dev libcurl4-openssl-dev libsqlite3-dev sqlite3

## Step 2. Install Ruby

Create a temporary folder for the Ruby source files:

$ mkdir ~/ruby

Move to the new folder:

$ cd ~/ruby

Download the latest stable Ruby source code. At the time of this writing, this is version 2.1.4. You can get the current latest version from the Ruby website. If a newer version is available, you will need to replace the link in the following command:

$ wget http://cache.ruby-lang.org/pub/ruby/2.2/ruby-2.2.2.tar.gz

Decompress the downloaded file:

$ tar -xzf ruby-2.2.2.tar.gz

Select the extracted directory:

$ cd ruby-2.2.2

Run the configure script. This will take some time as it checks for dependencies and creates a new Makefile, which will contain steps that need to be taken to compile the code:

$ ./configure

Run the make utility, which will use the Makefile to build the executable program. This step can take a bit longer:

$ make

Now, run the same command with the install parameter. It will try to copy the compiled binaries to the /usr/local/bin folder. This step requires root access to write to this directory:

$ sudo make install

Ruby should now be installed on the system. We can check it with the following command, which should print the Ruby version:

$ ruby -v

If your Ruby installation was successful, you should see output like the following:

$ ruby 2.2.2p95 (2015-04-13 revision 50295) [x86\_64-linux]

Finally, we can delete the temporary folder:

$ rm -rf ~/ruby

##Step 3 - Install Mysql

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

$ sudo apt-get install mysql-server libapache2-mod-auth-mysql php5-mysql

In this step you should enter your mySQL root password.

Once you have installed MySQL, we should activate it with this command:

$ sudo mysql\_install\_db

Finish up by running the MySQL set up script:

$ sudo /usr/bin/mysql\_secure\_installation

##Step 4 — Install Apache

To install Apache, type this command:

$ sudo apt-get install apache2

Then you should be able to see default apache2 web page.

##Step 5 — Install Passenger

First, install the PGP key for the repository server:

$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 561F9B9CAC40B2F7

Create an APT source file:

$ sudo nano /etc/apt/sources.list.d/passenger.list

Insert the following line to add the Passenger repository to the file:

deb https://oss-binaries.phusionpassenger.com/apt/passenger trusty main

Press CTRL+X to exit, type Y to save the file, and then press ENTER to confirm the file location.

Change the owner and permissions for this file to restrict access to ubuntu:

$ sudo chown ubuntu: /etc/apt/sources.list.d/passenger.list

$ sudo chmod 600 /etc/apt/sources.list.d/passenger.list

Update the APT cache:

$ sudo apt-get update

Finally, install Passenger:

$ sudo apt-get install libapache2-mod-passenger

Make sure the Passenger Apache module; it maybe enabled already:

$ sudo a2enmod passenger

Restart Apache:

$ sudo service apache2 restart

This step will overwrite our Ruby version to an older one. To resolve this, simply remove the incorrect Ruby location and create a new symlink to the correct Ruby binary file:

$ sudo rm /usr/bin/ruby

$ sudo ln -s /usr/local/bin/ruby /usr/bin/ruby

##Step 6 — Deploy

At this point you can deploy your own Rails application if you have one ready. If you want to deploy an existing app, you can upload your project to the server and skip to the /etc/apache2/sites-available/default step.

Now let's see how to do it.

Go to make workspace directory

$ mkdir ~/workspace

$ cd ~/workspace

Now clone the app

$ git clone https://github.com/adam-phillipps/saws.git

You should enter your github username and password

Move to the app root directory

$ cd saws/

Now we need to install a JavaScript execution environment. It can be installed as the therubyracer gem. To install it, first open the Gemfile:

$ nano Gemfile

Find the following line:

# gem 'therubyracer', platforms: :ruby

Uncomment it:

gem 'therubyracer', platforms: :ruby

Save the file, and install bundler gem

$ sudo gem install bundler

Run Bundler:

$ bundle install

Perhaps this might be failed due to a lib.

Please install a lib - libmysqlclient-dev

$ sudo apt-get install libmysqlclient-dev

Now, we need to create a virtual host file for our project. We'll do this by copying the default Apache virtual host:

$ sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/saws.conf

Open the config file:

$ sudo nano /etc/apache2/sites-available/saws.conf

Edit it or replace the existing contents so your final result matches the file shown below. Changes you need to make are highlighted in red. Remember to use your own domain name, and the correct path to your Rails app:

<VirtualHost \*:80>

#ServerName yourdomain.com

#ServerAlias www.example.com

#ServerAdmin webmaster@localhost

DocumentRoot /home/ubuntu/workspace/saws/public

RailsEnv production

ErrorLog ${APACHE\_LOG\_DIR}/saws-error.log

CustomLog ${APACHE\_LOG\_DIR}/saws-access.log combined

<Directory "/home/ubuntu/workspace/saws/public">

Options FollowSymLinks

Require all granted

</Directory>

</VirtualHost>

Basically, this file enables listening to our domain name on port 80, sets an alias for the www subdomain, sets the mail address of our server administrator, sets the root directory for the public directory of our new project, and allows access to our site. You can learn more about Apache virtual hosts by following the link.

To test our setup, we want to see the Rails Welcome aboard page. However, this works only if the application is started in the development environment. Passenger starts the application in the production environment by default, so we need to change this with the RailsEnv option. If your app is ready for production you'll want to leave this setting out.

If you don't want to assign your domain to this app, you can skip the ServerName and ServerAlias lines, or use your IP address.

Save the file (CTRL+X, Y, ENTER).

Don't forget to set username and password in <RAILS ROOT>/config/database.yml

Similar to following

production:

username: root

password: <Your password>

adapter: mysql2

encoding: utf8

reconnect: false

database: saws\_production

pool: 5

host: localhost

Now you should create database and migrate via rake

$ RAILS\_ENV=production rake db:create

$ RAILS\_ENV=production rake db:migrate

Disable the default site, enable your new site, and restart Apache:

$ sudo a2dissite 000-default

$ sudo a2ensite saws

$ sudo service apache2 restart

That's all.

Now you should see the app

Open http://ec2-52-26-121-108.us-west-2.compute.amazonaws.com/

> Written by [Paul Lin](https://www.upwork.com/users/~01d8caaa8e0f732462).