AWS Tutorial 09 - Deploying Rails Apps With Capistrano Take 1

Sep 23, 2016

In this tutorial we're going to use the latest version of Capistrano to deploy your Rails app onto one or more EC2 instances. I'm dividing this tutorial into two parts:

- Take 1 where a static set of boxes is encoded into your deploy/production.rb file
- Take 2 where we call AWS apis to figure out what boxes to deploy to

The reason for separating them is that take 2 will bring us much deeper into AWS API programming and concepts like tagging / IAM roles I wanted that to be highly separate from the guts of Capistrano.

Tools

In this tutorial I'm going to be specific to:

- Capistrano
- Passenger
- Bundler
- RVM

That's my standard Rails stack and while other configurations will work, NGINX is a favorite of people, that's the only stack I personally understand end to end.

Getting Started - Add Capistrano to Your Gemfile

You're going to want to use Capistrano 3.6 or later. Earlier versions of Capistrano should work but, in my experience, Capistrano 3.6 is a much better release - it took me away from Vlad finally and for that I was glad.

If your Gemfile you want something like this:

```
group :development do
  # Use Capistrano for deployment
  gem 'capistrano', "3.6.1"
  gem 'capistrano3-puma'
  gem 'capistrano-rails', require: false
  gem 'capistrano-bundler', require: false
  gem 'capistrano-rvm'
  gem 'capistrano-passenger'
end
```

After that you want to run bundle install:

```
bundle install
```

As long as this runs correctly you should be fine to move onto the next step.

Run the Cap install Command

Capistrano requires a one time setup step. Do this now:

```
Cap install
```

This generates a file in the root of your Rails app called Capfile. Open that file and uncomment these lines:

```
require 'capistrano/rvm'
require 'capistrano/bundler'
require 'capistrano/rails/assets'
```

Modify config/deploy.rb

The file config/deploy.rb covers the application level configuration for deployment. In this file you need to add lines like the ones below but specific to your code base / toolset:

```
set :application, 'banks'
set :repo_url, 'git@github.com:appdatallc/banks.git'

set :ssh_options, {
  forward_agent: true,
  auth_methods: ["publickey"],
  keys: ["#{Dir.home}/.ssh/fi_nav_sitecrawl.pem"]
}

set :deploy_to, '/var/www/apps/banks'
```

Modifying config/deploy/production.rb

The file config/deploy/production.rb covers the servers you are deploying to in the production environment. If you're setting up a staging server then there is a similar file in config/deploy/staging.rb.

Before you start does this section I strongly, strong, strongly advice you to read my AWS Tutorial on the SSH Config file. You can use the approach in that file to identify your boxes. If you do then this line:

```
server 'ec2-52-41-237-52.us-west-2.compute.amazonaws.com', user: 'ub
```

would be replaced by this line:

```
server 'ficrawler1', roles: %w{web app}
```

Not only is this simpler to understand but it centralizes all your EC2 machine addresses in one place. That's a huge win.

Regardless of the approach you decide to take, you need one line in this file for each EC2 instance along with the roles the instance provides.

Doing a Deploy

Once all of this configured then you should be able to do a:

```
cap production deploy
```

to get your current code base onto all your boxes. Please keep in mind that Capistrano executes everything via SSH and it generates a long stream of commands. If you see errors then there are two things to understand:

- Not every error may be significant and need to be addressed
- If you are trying to track down an error then I would strongly recommend reducing the boxes you deploy to to just one and then redeploying. This is much easier to troubleshoot exactly what is happening

References

Here are some of the research sources I used in writing this:

- http://jyaasa.com/blog/deploying-rails-web-application-using-capistrano-bundler-and-rvm <== This is the best source
- https://www.sitepoint.com/deploy-your-rails-app-to-aws/
- https://www.digitalocean.com/community/tutorials/how-to-deploy-a-rails-app-with-puma-and-nginx-on-ubuntu-14-04
- https://github.com/capistrano/bundler
- https://www.digitalocean.com/community/tutorials/deploying-a-rails-app-on-ubuntu-14-04-with-capistrano-nginx-and-puma
- http://stackoverflow.com/questions/21259826/ruby-on-rails-capistrano-3-deployment-error

- https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-an-ubuntu-14-04-server
- http://railsapps.github.io/installing-rails.html
- http://stackoverflow.com/questions/12967918/deploy-with-capistrano-using-a-pem-file
- http://stackoverflow.com/questions/9217868/deploying-ruby-on-rails-app-using-capistrano-cap-deploysetup-failing
- http://stackoverflow.com/questions/9277731/cap-generates-cannot-load-such-file-bundler-capistrano-loaderror
- http://stackoverflow.com/questions/15282532/capistrano-deploy-using-passenger-not-changing-release

- http://stackoverflow.com/questions/29241053/incomplete-response-received-from-application-from-nginx-passenger
- https://semaphoreci.com/community/tutorials/how-to-deploy-node-js-applications-with-capistrano

FuzzyBlog

FuzzyBlog
fuzzygroup@gmail.com

fuzzygroup
fuzzygroup

Scott Johnson writing about the usual array of nerd stuff: AWS / Ansible / Ruby / Rails / Elixir / Misc / Hyde.