

Getting Started with Capistrano in Ten Easy Steps

Keith Pitty

<http://keithpitty.org>
keith@keithpitty.org

What is Capistrano?

- A tool that aims to make it easy to deploy Rails applications!
- Written by Jamis Buck, one of the core Rails team

These ten steps are based on my experience

- Starting point:
 - “Agile Web Development with Rails”
 - 2nd edition, Section 27.5
- There may be better ways, but...
- The following steps worked for me

Step 1: Install Software

- Client and server
 - my client: OS X
 - my server: Fedora Core 6 on a VPS
- Native components
 - apache2, mysql5, ruby, rb-rubygems, rb-termios
- Ruby gems
 - rake, rails, termios, capistrano, mongrel, mongrel_cluster

```
$ sudo gem install --include-dependencies [gem]
```

Step 2: Set up Database

```
mysql> create database app;
```

```
mysql > grant all privileges on app.* to  
      'app'@'localhost identified by 'password'  
      with grant option;
```

Step 3: Add Mongrel to Project

- On development machine, this will create `config/mongrel_cluster.yml`:

```
$ mongrel_rails cluster::configure  
  -e production -p 8000 -a 127.0.0.1 -N -2  
  /deploy/path/current
```

Step 4: Apply Capistrano to Project

- On development machine, this will create `config/deploy.rb` and `lib/tasks/capistrano.rake`:

```
$ cap -apply-to /local/project/path [app]
```

Step 5: Configure Capistrano (1)

- Edit `config/deploy.rb`:
 - Add as first statement:

```
require 'mongrel_cluster/recipes'
```

Step 5: Configure Capistrano (2)

- Edit `config/deploy.rb`:

- Set variables:

```
set :application, "app"
```

```
set :repository,
```

```
  "http:svn.host.com/#{application}/trunk"
```

```
set :deploy_to,
```

```
  "/usr/local/railsapps/#{application}"
```

```
set :mongrel_conf,
```

```
  "#{current_path}/config/mongrel_cluster.yml"
```

Step 5: Configure Capistrano (3)

- Edit config/deploy.rb (continued):

```
set :web, "my.host.com"  
set :app, "my.host.com"  
set :db, "my.host.com", :primary => true  
set :user, "keith"
```

Step 6: Set up Directories

- On development machine, this will set up directories on target server:

```
$ cap setup
```

Step 7: Secure Database

- Copy `database.yml` to `{shared_path}/config` on server
- Rename to `database.yml.production` and set production username and password
- Add the following to `:after_update_code` task in `config/deploy.rb`:

```
db_config =  
  "{shared_path}/config/database.yml.production"  
run "cp #{db_config}  
    #{release_path}/config/database.yml"
```

Step 8: Ensure Subversion is OK

- Installed on client and server?
- Repository created on server?
- Apache configured for subversion?
- `svn import`?
- `svn checkout`?

Step 9: Deploy and Verify

- One more pre-req: edit sudoers file on server
- Deploy app, create tables (from client):

```
$ cap cold_deploy  
$ cap migrate
```

- Verify (on server):

```
$ curl -I http://127.0.0.1:8000
```

Step 10: Configure Apache

- Configure Apache to connect to Mongrel
 - Proxy for mongrel_cluster
 - VirtualHost with rewrite rules for app
 - see AWDwR, Section 27.5, Step Five for details
- Restart Apache
- App should now work via browser!

From now on it will be easy!

- When you've checked in some changes:

```
$ cap deploy
```

- If you need to back out your changes:

```
$ cap rollback
```

Resources

- “Agile Web Development with Rails”
 - Second Edition, section 27.5
- <http://manuals.rubyonrails.com/read/book/17>
 - when it's available
- <http://rubyforge.org/projects/capistrano/>
- <http://weblog.jamisbuck.org>
- capistrano@googlegroups.com
- rubyonrails-deployment@googlegroups.com

Thanks for listening!

Keith Pitty

<http://keithpitty.org>
keith@keithpitty.org