

# Capistrano

Paul Gross

<http://www.pgrs.net>

11/3/2008

## What is Capistrano?

- Capistrano is a tool for automating tasks on one or more remote servers over ssh
- Capistrano comes with a bunch of recipes to make deploying rails apps easier
- Capistrano lets you write deploy scripts in ruby

## Basic Recipe for Rails

```
set :application, "myapp"
set :repository, "https://www.example.com/svn/myapp"
set :deploy_to, "/var/www/#{application}"
set :user, "paul"
set :use_sudo, true

role :app, "server1.mysite.com"

namespace :deploy do
  task :restart do
    sudo "touch #{deploy_to}/current/tmp/restart.txt"
  end
end

after :deploy do
  deploy.cleanup
end
```

## The Filesystem

```
1 [deploy_to]
2 [deploy_to]/releases
3 [deploy_to]/releases/20080819001122
4 [deploy_to]/releases/...
5 [deploy_to]/shared
6 [deploy_to]/shared/log
7 [deploy_to]/shared/pids
8 [deploy_to]/shared/system
9 [deploy_to]/current -> [deploy_to]/releases/20080819001122
```

## Roles

```
role :app, "web01.mysite.com", "web02.mysite.com"
role :background, "bg.mysite.com"

task :configure_monit, :roles => [:app, :background] do
  ...
end

task :configure_apache, :roles => [:app] do
  ...
end
```



## ERBing and Installing Config Files

```
after :'deploy:symlink' do
  add_cron_jobs
end

task :add_cron_jobs, :roles => :background do
  put(run_template("../cron_jobs/jobs.txt.erb"), "#{deploy_to}/current/cron_jobs/jobs.txt")
  run "crontab #{deploy_to}/current/cron_jobs/jobs.txt"
end

def run_template(base_name)
  template = File.read(File.dirname(__FILE__) + "../" + base_name)
  ERB.new(template, nil, "<>").result(binding)
end
```

## Multiple Environments

- Capistrano multistage extension basically does this:

```
task :staging do
  set :email_address, 'test@example.com'
  role :app, staging.example.com
end

task :production do
  set :email_address, 'alert@example.com'
  role :app, www.example.com
end
```

- Now, you can run:

```
% cap staging deploy
% cap production deploy
```

## Deploying Trunk or Tags

```
set :repository_root, "http://www.example.com/svn/myproject"
set(:tag) { Capistrano::CLI.ui.ask("Tag to deploy (or type 'trunk' to deploy from trunk): ") }
set(:repository) {
  (tag == "trunk") ? "#{repository_root}/trunk" : "#{repository_root}/tags/#{tag}"
}
```

```
namespace :deploy do
  task :trunk do
    sh "cap -s tag=trunk deploy"
  end
end
```



## Deploy Verification

```
after :deploy do
  verify.apache
end

namespace :verify do
  task :apache, :roles => :app do
    verify_status_code '200', "http://localhost:8080/stylesheets/base.css"
  end
end

def verify_status_code expected, url
  errors = []
  run %{curl -s -o /dev/null -w '%{http_code}' #{url} } do |ssh_channel, stream, output|
    if output != expected
      errors << "Expected: #{expected.inspect} but was #{output.inspect} " +
        "on #{ssh_channel.connection.host}"
    end
  end
  raise errors.join("\n") unless errors.empty?
end
```

## Capistrano Shell

```
% HOSTS=localhost,pgrs.net cap -s user=paul shell
* executing `shell`
```

```
=====
Welcome to the interactive Capistrano shell! This is an experimental
feature, and is liable to change in future releases. Type 'help' for
a summary of how to use the shell.
-----
```

```
cap> gem list | grep rails
[establishing connection(s) to localhost, pgrs.net]
Password:
** [out :: localhost] rails (2.1.0, 2.0.2, 1.2.6)
** [out :: pgrs.net] rails (2.1.1)
cap> which svn
** [out :: localhost] /usr/bin/svn
** [out :: pgrs.net] /usr/bin/svn
cap>
```

## Shell One Liners

```
% HOSTS=localhost,pgrs.net COMMAND="svn --version | head -n 1" cap -s user=paul invoke
* executing `invoke`
* executing "svn --version | head -n 1"
  servers: ["localhost", "pgrs.net"]
Password:
  [localhost] executing command
** [out :: localhost] svn, version 1.4.4 (r25188)
  [pgrs.net] executing command
** [out :: pgrs.net] svn, version 1.5.1 (r32289)
  command finished
```

## Using the Shell in Tasks

```
task :pairing_stations do
  role :app, "green.local", "blue.local", "red.local"
  shell
end
```



## Using the Shell with Existing Tasks

- With our existing example of environments:

```
task :staging do
  set :email_address, 'test@example.com'
  role :app, staging.example.com
end
```

```
task :production do
  set :email_address, 'alert@example.com'
  role :app, www.example.com
end
```

- You can run:

```
% cap staging shell
% cap production shell
```



## More Information

- <http://www.capify.org>
- <http://github.com/jamis/capistrano/tree/master>
- <http://groups.google.com/group/capistrano>