

# Doing Bad Things with Capistrano

I've cut out the bad  
things; it's all good

Who Am I?

What is Capistrano?

Capistrano allows you  
to execute commands  
on remote servers

Commands are  
specified in ruby

Commands are  
executed using SSH

# Requirements

On local machine:

- \* Ruby

On remote machine:

- \* SSH Server (OpenSSH or similar)
- \* Unix-like OS (or Cygwin)
- \* No ruby required!

# The basic Capistrano Commands

# run

# runs a command on the remote machine

```
run 'echo "something"'
```

```
run 'mv ~/config.file /var/apps/config.file'
```

```
run %Q{  
    cp #{template} #{template_destination}  
}
```

# sudo

# Runs commands as super user

```
sudo '/etc/init.d/apache2 stop'
```

```
sudo "chown -R :mongrel #{current_release}"
```

# put

```
# Puts a file on the remote machine, using  
# SFTP if available
```

```
put "Hello world", "~/helloworld.txt"
```

```
put File.read('web.xml'),  
    "#{release_path}/WEB-INF/web.xml"
```

# delete

```
# Deletes a file on the remote machine, using  
# rm -f
```

```
delete “#{release_path}/db.yml”
```

```
delete “#{release_path}/cache”,  
      :recursive => true
```

# render

```
# Simple interface to render erb templates  
# and return the results. Does nothing on  
# any remote machine!
```

```
render :template => "Hello <%= t %>",  
       :t => "world"
```

```
render "database.yml.template",  
       :db => 'herzog'
```

# Using commands in tasks

# A simple task

```
desc "Reload Apache Configuration"  
  
task :reload_apache, :roles => :web do  
  sudo "/etc/init.d/apache2 reload"  
end
```

# Task description

```
desc "Reload Apache Configuration"
```

```
task :reload_apache, :roles => :web do  
  sudo "/etc/init.d/apache2 reload"  
end
```

# Task declaration

```
desc "Reload Apache Configuration"  
  
task :reload_apache, :roles => :web do  
  sudo "/etc/init.d/apache2 reload"  
end
```

# Task body

```
desc "Reload Apache Configuration"

task :reload_apache, :roles => :web do
  sudo "/etc/init.d/apache2 reload"
end
```

The final ingredients:  
variables and roles

# Setting variables

```
set :domain, "mephisto.popdog.net"
```

```
set :repository,  
    "svn+ssh://svn.popdog.net/sites/mephisto"
```

# Setting roles

```
role :web, "kurosawa.popdog.net"
```

```
role :db, "bergman.popdog.net"
```

Putting it all together

# config/deploy.rb

```
role :web, "goddard.popdog.net"  
role :web, "truffaut.popdog.net"  
  
desc "Reload Apache Configuration"  
  
task :reload_apache, :roles => :web do  
  sudo "/etc/init.d/apache2 reload"  
end
```

# Finally, running the command

```
kurosawa:~/tomw/demo$ cap reload_apache
```

```
loading configuration ./config/deploy.rb
```

```
* executing task reload_apache
```

```
* executing "sudo /etc/init.d/apache2 reload"
```

```
servers: ["goddard.popdog.net",  
          "truffaut.popdog.net"]
```

```
[goddard.popdog.net] Text executing command
```

```
[truffaut.popdog.net] executing command
```

```
** [out :: goddard.popdog.net] * Reloading apache  
2.0 configuration...
```

```
** [out :: truffaut.popdog.net] * Reloading apache  
2.0 configuration...
```

```
** [out :: goddard.popdog.net] [ ok ]
```

```
** [out :: truffaut.popdog.net] [ ok ]
```

Putting our task into a  
library

# popdog/capistrano/recipes/apache.rb

```
Capistrano.configuration(:must_exist).load do |a|
  task :reload_apache, :roles => :web do
    sudo "/etc/init.d/apache2 reload"
  end
end
```

# Change config/deploy.rb

```
require "popdog/capistrano/recipes/apache"
```

```
role :web, "goddard.popdog.net"
```

```
role :web, "truffaut.popdog.net"
```

More tasks I've written

# Apache Tasks

```
Capistrano.configuration(:must_exist).load do |a|
  task :start_apache, :roles => :web do
    sudo "/etc/init.d/apache2 start"
  end

  task :stop_apache, :roles => :web do
    sudo "/etc/init.d/apache2 stop"
  end

  task :restart_apache, :roles => :web do
    sudo "/etc/init.d/apache2 restart"
  end

  task :reload_apache, :roles => :web do
    sudo "/etc/init.d/apache2 reload"
  end
end
```

# Postgresql Tasks

```
Capistrano.configuration(:must_exist).load do |a|
  task :create_db_user, :roles => :db do
    sudo "createuser -D -A -q #{app}"
  end

  task :create_db, :roles => :db do
    sudo "createdb --owner #{app} #{app}_#{env}"
  end

  task :setup_db, :roles => :db do
    create_db_user
    create_db
  end
end
```

# Mongrel Tasks

```
Capistrano.configuration(:must_exist).load do |a|
  task :config_cluster, :roles => :app do
    put render(File.read("mongrel_cluster.yml")),
        "#{current}/config/mongrel_cluster.yml"
  end

  task :update_cluster_config, :roles => :app do
    stop_mongrel_cluster
    config_mongrel_cluster
    start_mongrel_cluster
  end

  task :start_mongrel_cluster, :roles => :app do
    sudo "mongrel_rails cluster::start -C
#{current}/config/mongrel_cluster.yml"
  end

  task :stop_mongrel_cluster, :roles => :app do
    sudo "mongrel_rails cluster::stop -C #
{current_release}/config/mongrel_cluster.yml"
  end
end
```

# mongrel\_config.yml

```
---  
cwd: <%= current_path %>  
port: <%= cluster_port %>  
environment: <%= env %>  
address: 127.0.0.1  
pid_file: log/mongrel.pid  
servers: <%= cluster_size %>  
user: mongrel  
group: mongrel
```

# config/deploy.rb

```
require "popdog/capistrano/recipes/apache"  
require "popdog/capistrano/recipes/postgres"  
require "popdog/capistrano/recipes/mongrel"  
  
set :app, "mephisto"  
set :domain, "fassbinder.popdog.net"  
set :cluster_port, 8301  
set :cluster_size, 1  
set :env, "production"  
  
set :repository, "http://svn.techno-  
weenie.net/projects/mephisto/tags/rel-0.6.1/"  
  
role :web, domain  
role :db, domain
```

Some of the Tasks  
included with  
Capistrano

# setup

Sets up a server and creates application directory tree.

```
<release_path>/releases  
<release_path>/shared  
<release_path>/shared/log  
<release_path>/shared/system
```

# deploy

Checks out code into something like:

`<path>/20060906134554`

Creates symbolic link to above called current

`<path>/current -> <path>/20060906134554`

N.B. Also does some other linking to shared log and system directory, and executes restart task

# rollback

Deletes latest release, and points current link to previous release

# More Capistrano Features

- \* Cap Shell, command-line prompt into many machines
- \* Task extensions, using either `before_<name>` or `after_<name>` to extend tasks

# The slide with the Pros and ~~Cons~~

- \* Capistrano makes it easy to repeat tasks on one or more remote machines
- \* Useful for far more than just rails deployment, limited only by imagination
- \* Doesn't require ruby on remote machines, so maybe a way to slip ruby into your company?

# Further reading

\* The railsmachine gem - a great set of tasks, built for their environment but easy to adapt

\* The Capistrano Manual:

<http://manuals.rubyonrails.com/read/book/17>

\* The mailing list:

<http://groups.google.com/group/capistrano>

\* Jamis Buck's Blog:

<http://weblog.jamisbuck.org>