

[Advanced Search](#)

Search...

- [Explore](#)
- [Gist](#)
- [Blog](#)
- [Help](#)

[pmq20](#)

- [Notifications](#)
- [Account Settings](#)
- [Log Out](#)

[nesquena](#) / [cap-recipes](#)

- [Watch](#) [Unwatch](#)
- [Fork](#)
- - [331](#)
 - [27](#)
- [Code](#)
- [Network](#)
- [Pull Requests 0](#)
- [Issues 3](#)
- [Wiki 1](#)
- [Stats & Graphs](#)

1. [Ruby 100%](#)

Battle-tested capistrano recipes for ruby, rubygems, apache, passenger, delayed_job, juggernaut, thinking_sphinx, mongodb, whenever, among other popular tools — [Read more](#)

- [🍏 Clone in Mac](#)
- [ZIP](#)
- [HTTP](#)
- [Git Read-Only](#)

Read+Write access

- [Tags 47](#)
- [Downloads 0](#)
- [branch: master](#)
Switch Branches/Tags

Filter branches/tags

- [Branches](#)
- [Tags](#)

[el-changes](#)

[master](#)

- [Files](#)
- [Commits](#)
- [Branches 2](#)

Latest commit to the **master** branch










[Update simple example for cap deploy](#)

[commit 53b76b83cb](#)



[nesquena](#) authored a month ago

[cap-recipes](#) /

name	age	history message
 examples	a month ago	Update simple example for cap deploy [nesquena]
 lib	a month ago	whitespace [nesquena]
 spec	3 years ago	trying to add some tests [grosser]
 .gitignore	a month ago	Ignore Rubymine files [elmatou]
 LICENSE	3 years ago	updated to using jeweler for packaging my gem [nesquena]
 README.textile	2 years ago	(Thanks achiu) Added mongodb recipes to library [nesquena]
 Rakefile	2 years ago	Fixed up readme and rakefile text [nesquena]
 VERSION.yml	a month ago	Version bump to 0.3.39 [nesquena]
 cap-recipes.gemspec	a month ago	Regenerate gemspec for version 0.3.39 [nesquena]

README.textile

cap-recipes

A collection of useful capistrano recipes I have been using in production on a number of sites.

This gem is best suited for ruby (and rails) projects which deploy using Apache and Phusion Passenger. Worth noting is that the installation tasks are for systems using aptitude package management.

The overall vision for this project is simply to collect a number of classic, generic recipes for common issues which can then be used by the community.

Feel free to improve upon this recipe collection by adding your own!

Currently included recipes in this gem:

- Ruby Setup
- Rubygems Management
- Apache Server
- Phusion Passenger
- Memcached Management
- ThinkingSphinx Daemon
- Juggernaut Daemon
- Backgroundrb Server
- DelayedJob Worker
- Whenever Cron Scheduling
- MongoDB Management
- Aptitude Package Management
- Gitis Git Repository Hosting

Check out the USAGE section below for specific tasks included in the recipes.

INSTALLATION

To install the gem, execute:

```
$ sudo gem install cap-recipes --source http://gemcutter.org
```

Then, include into your `deploy.rb` configuration file for Capistrano:

```
# use the complete deployment process (NOT RECOMMENDED)
require 'cap_recipes'

# RECOMMENDED
# require necessary recipes according to your needs
# (use hooks which tie all the recipes into the deployment process,
# tasks for managing and tasks for installing):
require 'cap_recipes/tasks/memcache'
require 'cap_recipes/tasks/passenger'
require 'cap_recipes/tasks/thinking_sphinx'
require 'cap_recipes/tasks/rails'
require 'cap_recipes/tasks/delayed_job'
# or ruby, rubygems, apache, mongodb, juggernaut, backgroundrb, aptitude, gitosis, whenever

# OR
# only use managing tasks:
require 'cap_recipes/tasks/memcache/manage'

# OR
# only use install tasks:
require 'cap_recipes/tasks/memcache/install'
```

You can find full examples of what your `deploy.rb` file should look like in the `examples` folder.

USAGE

Apache

These recipes manage the apache web server

Configuration

- `apache_init_path` – the path to the `init.d` apache file [default: `‘/etc/init.d/apache2’`]

Tasks

`manage.rb`

- `apache:stop` – stops the apache server
- `apache:start` – starts the apache server
- `apache:restart` – restarts the apache server

`install.rb`

- `apache:install` – installs apache and dependencies onto a debian system

Ruby

Configuration

- None required

Tasks

install.rb

- ruby:setup – Installs ruby 1.8 using standard aptitude packages

Rubygems

Configuration

- rubygem_paths – a path or array of paths to your gem binaries [default ‘/usr/bin/gem’]
- rubygems_version – which version of rubygems to install [default ‘1.3.5’]

Tasks

manage.rb

- rubygems:full_update – Upgrades rubygems, updates all installed rubygems and performs a cleanup
- rubygems:upgrade – Upgrades rubygems version (if available)
- rubygems:update – Updates all installed rubygems
- rubygems:cleanup – Performs a cleanup for all outdated rubygems
- rubygems:install – Installs a specified gem from name inputted on console
- rubygems:uninstall – Removes a specified rubygem from name inputted on console

install.rb

- rubygems:install – Installs rubygems by downloading package and running setup.rb

Passenger

These recipes manage the passenger module for apache

Configuration

- base_ruby_path – the base path to the ruby installation [default: ‘/usr’]
- local_ping_path – the localhost path to ping to start passenger [default: “http://localhost”]

The following files and folders are expected to exist:

- “#{base_ruby_path}/lib/ruby”
- “#{base_ruby_path}/bin/ruby”
- “#{base_ruby_path}/bin/gem”

Tasks

manage.rb

- deploy:start – Starts the apache server (if needed)
- deploy:stop – Stops the apache server (if needed)

- `deploy:restart` – Touches the `restart.txt` file

`install.rb`

- `passenger:install` – Installs the `passenger` gem, compiles the module silently, and adds the necessary configuration

Aptitude

Tasks

`manage.rb`

- `aptitude:install` – Installs a specified aptitude package from name inputted on console
- `aptitude:remove` – Removes a specified aptitude package from name inputted on console
- `aptitude:updates` – Updates all installed aptitude packages

Rails

These recipes support rails-specific functionality

Tasks

`manage.rb`

- `rails:symlink_db_config` – Symlinks the `database.yml` file from `shared/config` to release path
- `rails:repair_permissions` – Forces the permissions on the rails app to make permission errors a thing of the past
- `rails:tail` – tail production log – Provides a stream tailing the production log
- `rails:ping` – pings the server to start a passenger instance (reduces downtime)
- `rails:sweep:cache` – performs a file cache sweep in production
- `rails:sweep:log` – performs a log sweep in production

`hooks.rb`

```
after "deploy:update_code", "rails:symlink_db_config" # copy database.yml file to release path
after "deploy:update_code", "rails:sweep:cache" # clear cache after updating code
after "deploy:restart"      , "rails:repair_permissions" # fix the permissions to work properly
after "deploy:restart"      , "rails:ping" # ping passenger to start the rails instance
```

DelayedJob

These recipes are for `tobi`'s `delayed_job` plugin for background job queue processing

Configuration

- `delayed_script_path` – the path to the delayed job script [default: `'#{current_path}/script/delayed_job'`]
- `delayed_job_env` – the rails environment [default: `'production'`]

Tasks

`manage.rb`

- `delayed_job:stop` – stops the `delayed_job` workers
- `delayed_job:start` – starts the `delayed_job` workers

- `delayed_job:restart` – restarts the `delayed_job` workers

hooks.rb

```
after "deploy:start",    "delayed_job:start"
after "deploy:stop",     "delayed_job:stop"
after "deploy:restart",  "delayed_job:restart"
```

Backgroundrb

These recipes are for backgroundrb job queue processing

Configuration

- `backgroundrb_log` – the path to the backgroundrb log file
- `backgroundrb_host` – the background host machine ip [default: 'localhost']
- `backgroundrb_env` – the rails environment [default: 'production']

Tasks

manage.rb

- `backgroundrb:stop` – stop backgroundrb workers
- `backgroundrb:start` – start backgroundrb workers
- `backgroundrb:restart` – restart backgroundrb workers
- `backgroundrb:symlink_config` – copy backgroundrb.yml from shared/config to release
- `backgroundrb:tail` – tails the backgroundrb log file in production

hooks.rb

```
after "deploy:update_code" , "backgroundrb:symlink_config" # copy backgroundrb config to release
after "deploy:restart"     , "backgroundrb:restart"        # restart backgroundrb daemon
after "backgroundrb:restart" , "backgroundrb:repair_permissions" # restart backgroundrb daemon
```

Juggernaut

These recipes are for managing the juggernaut push server

Configuration

- `juggernaut_config` – path to juggernaut config file [default: “#{current_path}/config/juggernaut.yml”]
- `juggernaut_pid` – path to juggernaut pid file [default: “#{current_path}/tmp/pids/juggernaut.pid”]
- `juggernaut_log` – path to juggernaut log file [default: “#{current_path}/log/juggernaut.log”]

Tasks

manage.rb

- `juggernaut:start` – starts the juggernaut push server
- `juggernaut:stop` – stop the juggernaut push server
- `juggernaut:restart` – restart the juggernaut push server
- `juggernaut:symlink_config` – copy juggernaut.yml from shared/config to release
- `juggernaut:tail` – tails the juggernaut log file in production

hooks.rb

```
after "deploy:update_code", "juggernaut:symlink_config" # copy juggernaut.yml to release
after "deploy:restart"      , "juggernaut:restart"      # restart juggernaut daemon
```

ThinkingSphinx

These recipes are for managing the thinking_sphinx search index daemon

Configuration

- None required

Tasks

manage.rb

- thinking_sphinx:start – starts the thinking_sphinx search daemon
- thinking_sphinx:index – executes the thinking_sphinx search indexer
- thinking_sphinx:stop – stop the thinking_sphinx search daemon
- thinking_sphinx:restart – restart (and index) the thinking_sphinx search daemon
- thinking_sphinx:symlink_config – copy sphinx.yml from shared/config to release
- thinking_sphinx:tail – tails the juggernaut log file in production

install.rb

- thinking_sphinx:install – installs sphinx, dependencies, and thinking_sphinx gem

hooks.rb

```
after "deploy:update_code", "thinking_sphinx:symlink_config" # sym thinking_sphinx.yml on update code
after "deploy:restart"      , "thinking_sphinx:restart"      # restart thinking_sphinx on app restart
```

Memcache

These recipes are for managing the memcached caching mechanism

Configuration

- memcache_init_path – path to memcache config file [default: “/etc/init.d/memcache”]
- memcache_size – the total size of memory to use [default: 64]
- memcache_port – the port to start memcache [default: ‘11211’]
- memcache_host – the host for memcache [default: ‘127.0.0.1’]
- memcache_user – the user to run memcache [default: ‘nobody’]

Tasks

manage.rb

- memcache:start – starts the memcache daemon
- memcache:stop – stops the memcache daemon
- memcache:restart – restarts the memcache daemon

install.rb

- memcache:install – Installs memcache aptitude package and rubygem

hooks.rb

```
after "deploy:restart", "memcache:restart" # clear cache after updating code
```

MongoDB

These recipes are for installing and managing the mongodb document-oriented database

Configuration

- mongodb_data_path – the location to store the mongodb data [default: “/data/db”]
- mongodb_bin_path – the location to install mongodb [default: “/opt/mongo”]
- mongodb_log – the path to the mongodb log file [default: “/var/log/mongodb.log”]

Tasks

install.rb

- mongodb:install – Performs the full installation of mongodb and dependencies

manage.rb

- mongodb:start – Starts the mongodb process
- mongodb:stop – Stops the mongodb process
- mongodb:restart – Restarts the mongodb process

Gitis

These recipes are for installing Gitis Git Repository Hosting

Configuration

- None required

Tasks

install.rb

- gitosis:install

Whenever

These recipes are for managing whenever cron job scheduling

Configuration

- None required

Tasks

manage.rb

- `whenever:update_crontab` – Performs an update applying the `schedule.rb` file to your server's cron

hooks.rb

```
after "deploy:symlink", "whenever:update_crontab"
```

EXAMPLE

Here is a sample `deploy.rb` file using `cap_recipes`:

```
# =====
# GENERAL SETTINGS
# =====

set :application, "app_name"
set :deploy_to, "/var/apps/#{application}"
set :scm, :git
set :repository, "git@repos.site.com:/home/git/repos.git"

# =====
# CAP RECIPES
# =====

# Note this happens after the general settings have been defined
require 'rubygems'

# PASSENGER
require 'cap_recipes/tasks/passenger'
set :base_ruby_path, '/opt/ruby-enterprise' # defaults to "/usr"
set :apache_init_path, '/etc/init.d/apache2' # defaults to "/etc/init.d/apache2"

# BACKGROUNDDRB
require 'cap_recipes/tasks/backgrounddrb'
set :backgrounddrb_log, "/var/log/backgrounddrb.log" # defaults to "#{release_path}/log/backgrounddrb.log"
set :backgrounddrb_host, "worker.site.com" # defaults to localhost
set :backgrounddrb_env, "staging" # defaults to production

# DELAYED_JOB
require 'cap_recipes/tasks/delayed_job'
set :delayed_script_path, 'script/djworker' # defaults to 'script/delayed_job'
set :delayed_job_env, 'staging' # defaults to production

# JUGGERNAUT
require 'cap_recipes/tasks/juggernaut'
set :juggernaut_config, "/some/path/juggernaut.yml" # defaults to "#{current_path}/config/juggernaut.yml"
set :juggernaut_pid, "/some/path/juggernaut.pid" # defaults to "#{current_path}/tmp/pids/juggernaut.pid"
set :juggernaut_log, "/var/log/juggernaut.log" # defaults to "#{release_path}/log/juggernaut.log

# MEMCACHE
require 'cap_recipes/tasks/memcache'
set :memcache_init_path, "/etc/init.d/memcache" # defaults to "/etc/init.d/memcache"
```

You can find more examples of what your `deploy.rb` file should look like in the `examples` folder.

UTILITIES

In addition to the recipes themselves, this gem provides a bunch of useful utility methods which make writing your own recipes or extending existing ones much easier by creating reusable commands for common subtasks in a recipe.

There are too many to mention all of them directly, but check out `lib/cap_recipes/tasks/utilities.rb` to browse all available utility methods. A few are listed below:

- `utilities.config_gsub(file, findPattern, replaceString)`
 - Replaces the `findPattern` with `replaceString` within specified file on server
 - `# utilities.config_gsub('/etc/example', /\#(option:.*?\n)/im, "\1")`
- `utilities.ask(question, default)`
 - Asks the user a question on the command-line and returns the response
 - `utilities.ask('What is your name?', 'John')`
- `utilities.yes?(question)`
 - Asks the user a yes or no question on the command-line and returns boolean result
 - `utilities.yes?('Proceed with installation?')`
- `utilities.apt_install(packages)`
 - Installs all specified aptitude packages silently without user intervention
 - `utilities.apt_install %w[package1 package2]`
- `utilities.sudo_upload(local_path, remote_dest, options)`
 - Uploads a file to a temporary location and then sudo moves it to specified remote destination
 - `utilities.sudo_upload('/local/path/to/file', '/remote/path/to/destination', :mode => 'u+rwX')`
- `utilities.with_role(role) { ... }`
 - Forces tasks in block to execute only within specific role(s)
 - This is useful because of certain limitations in the role declaration in capistrano
- `utilities.with_credentials(:user => 'jsmith', :password => 'secret') { ... }`
 - Forces tasks in block to execute using the given user/password credentials
 - This is useful because of certain limitations in capistrano

CONTRIBUTORS

The following people are the reason this library exists:

- nesquena [Nathan Esquenazi] – created and maintaining the library
- achiu [Arthur Chiu] – contributed gitosis, ruby, mongodb and other recipes
- hubertlepicki – contributed thinking_sphinx recipes

LICENSE

(The MIT License)

Copyright © 2008 Nathan Esquenazi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ‘AS IS’, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GitHub Links

GitHub

- [About](#)
- [Blog](#)
- [Features](#)
- [Contact & Support](#)
- [Training](#)
- [GitHub Enterprise](#)
- [Site Status](#)

Tools

- [Gauges: Analyze web traffic](#)
- [Speaker Deck: Presentations](#)
- [Gist: Code snippets](#)
- [GitHub for Mac](#)
- [Issues for iPhone](#)
- [Job Board](#)

Extras

- [GitHub Shop](#)
- [The Octodex](#)

Documentation

- [GitHub Help](#)
- [Developer API](#)
- [GitHub Flavored Markdown](#)
- [GitHub Pages](#)
- [Terms of Service](#)
- [Privacy](#)
- [Security](#)

© 2012 GitHub Inc. All rights reserved.



Powered by the [Dedicated Servers](#) and [Cloud Computing](#) of Rackspace Hosting®

Markdown Cheat Sheet

Format Text

Headers

```
# This is an <h1> tag
## This is an <h2> tag
##### This is an <h6> tag
```

Text styles

```
*This text will be italic*
_This will also be italic_
**This text will be bold**
__This will also be bold__

*You **can** combine them*
```

Lists

Unordered

```
* Item 1
* Item 2
  * Item 2a
  * Item 2b
```

Ordered

```
1. Item 1
2. Item 2
3. Item 3
  * Item 3a
  * Item 3b
```

Miscellaneous

Images

```
![GitHub Logo](/images/logo.png)
Format: ![Alt Text](url)
```

Links

```
http://github.com - automatic!
[GitHub](http://github.com)
```

Blockquotes

```
As Kanye West said:
```

```
> We're living the future so  
> the present is our past.
```

Code Examples in Markdown

Syntax highlighting with [GFM](#)

```
```javascript  
function fancyAlert(arg) {
 if(arg) {
 $.facebox({div:'#foo'})
 }
}
```
```

Or, indent your code 4 spaces

Here is a Python code example
without syntax highlighting:

```
def foo:  
    if not bar:  
        return true
```

Inline code for comments

I think you should use an
`<addr>` element here instead.

Something went wrong with that request. Please try again. [Dismiss](#)