



Chapters

- 1. [Introduction](#)
 - [What is Capistrano?](#)
 - [What can it do?](#)
 - [What assumptions does it make?](#)
- 2. [Quick Start](#)
 - [Getting started](#)
 - [Installing Capistrano](#)
 - [Deployment Recipe](#)
 - [Setup](#)
 - [Apache Configuration](#)
 - [Deploying](#)
 - [Rolling back a release](#)
- 3. [A More Complicated Example](#)
 - [Getting started](#)
 - [Deployment Recipe](#)
 - [Spinner](#)
 - [Spawner](#)
 - [Reaper](#)
 - [Deploying](#)
- 4. [Recipes](#)
 - [Introduction](#)
 - [Variables](#)
 - [Roles](#)
 - [Tasks](#)
 - [Extending Tasks](#)
- 5. [Standard Tasks](#)
 - [Overview](#)
 - [cleanup](#)
 - [cold_deploy](#)
 - [deploy](#)
 - [diff from last_deploy](#)
 - [disable_web](#)
 - [enable_web](#)
 - [invoke](#)
 - [migrate](#)
 - [restart](#)
 - [rollback](#)
 - [rollback_code](#)
 - [setup](#)
 - [show_tasks](#)
 - [spinner](#)
 - [symlink](#)
 - [update_code](#)
- 6. [Creating Tasks](#)
 - [Overview](#)
 - [run](#)
 - [sudo](#)
 - [put](#)
 - [delete](#)
 - [render](#)
 - [transaction](#)
 - [on_rollback](#)
- 7. [Extending Capistrano](#)
 - [Task Libraries](#)
 - [Writing a Task Library](#)
 - [Using a Task Library](#)
 - [Extension Libraries](#)

Options

- [exports](#)
- [recent changes](#)
- [rss 2.0](#) | [atom](#)

Authors

- [Login](#) [Signup](#)

About "Capistrano: Automating Application Deployment"

Application deployment is one of those things that becomes more and more complicated as the scale of your application increases. With just a single box running your database and your application, it’s quite simple. But when you start putting your database on a different server, and then separating your web servers from your app servers, and eventually splitting your database into master and slave servers... It can get to where you almost don’t want to deploy your application any more.

Capistrano is a standalone utility that can also integrate nicely with Rails. You simply provide Capistrano with a deployment “recipe” that describes your various servers and their roles, and voila! You magically have single-command deployment. It even allows you to roll a bad version out of production and revert back to the previous release.

It should be stated that the concepts that Capistrano uses and encourages are *not* specific to Rails, or even Ruby. They are common-sense practices that are applicable to a variety of environments. In fact, you’ll find that there is very little that is Rails-specific about Capistrano, aside from the fact that it is in Rails that it found its genesis. No matter where you are or what environment you are using, Capistrano can probably help ease your deployment pains.

Of course, we hope you’re using Ruby on Rails...

Contents

- 1. [Introduction](#)
 - [What is Capistrano?](#)
 - [What can it do?](#)
 - [What assumptions does it make?](#)
- 2. [Quick Start](#)
 - [Getting started](#)
 - [Installing Capistrano](#)
 - [Deployment Recipe](#)
 - [Setup](#)
 - [Apache Configuration](#)
 - [Deploying](#)
 - [Rolling back a release](#)
- 3. [A More Complicated Example](#)
 - [Getting started](#)
 - [Deployment Recipe](#)
 - [Spinner](#)
 - [Spawner](#)
 - [Reaper](#)
 - [Deploying](#)
- 4. [Recipes](#)
 - [Introduction](#)
 - [Variables](#)
 - [Roles](#)
 - [Tasks](#)
 - [Extending Tasks](#)
- 5. [Standard Tasks](#)
 - [Overview](#)
 - [cleanup](#)
 - [cold_deploy](#)
 - [deploy](#)
 - [diff from last_deploy](#)
 - [disable_web](#)
 - [enable_web](#)
 - [invoke](#)
 - [migrate](#)
 - [restart](#)
 - [rollback](#)
 - [rollback_code](#)
 - [setup](#)
 - [show_tasks](#)
 - [spinner](#)
 - [symlink](#)
 - [update_code](#)
- 6. [Creating Tasks](#)
 - [Overview](#)
 - [run](#)
 - [sudo](#)
 - [put](#)
 - [delete](#)
 - [render](#)
 - [transaction](#)
 - [on_rollback](#)
- 7. [Extending Capistrano](#)
 - [Task Libraries](#)
 - [Writing a Task Library](#)
 - [Using a Task Library](#)
 - [Extension Libraries](#)