**Deploying with Capistrano**

**Capistrano?**

* It is a remote server automation tool used to deploy web applications on servers.
* It will just copy the code from git repository to server via SSH.
* Then it will perform pre and post-deploy functions like restarting a webserver, busting cache, renaming files, running database migrations and to deploy on many machines at once.

**Install Ruby on system: (From Terminal) (Optional – In case ruby is missing)**

* sudo apt install ruby
* password

**Install Capistrano: (From Terminal)**

* gem install capistrano
* gem install capistrano-ext

Next, Setup SSH Keys for the system ([Mac](http://guides.beanstalkapp.com/version-control/git-on-mac.html#creating-ssh-keys) or [Windows](http://guides.beanstalkapp.com/version-control/git-on-windows.html#installing-ssh-keys))

## Prepare your Project for Capistrano:

* Navigate to your application’s root directory in Terminal and run the following command:
* capify .
* This command creates a special file called Capfile in your project, and adds a template deployment recipe at config/deploy.rb in your Rails project.
* Open the deploy.rb file in any text editor.

**Working with Capistrano Recipes:**

* To create a Capistrano Recipe, enter the name of application in deploy.rb file
* set :application, “fancy\_shoes”
* Add Repository
* set :scm, :git
* set :repository, “[git@account.git.beanstalkapp.com:/account/repository.git](mailto:git@account.git.beanstalkapp.com:/account/repository.git)”
* set :scm\_passphrase, “”
* set :deploy\_via, :remote\_cache
* For subversion users,
* set :scm, :subversion
* set :repository, “<https://account.svn.beanstalkapp.com/repository>”
* Set the user on Server, who has access to directory specified in deploy\_to variable.
* set :user, “server-user-name”
* Next, try connecting to that repository
* svn co

or

* git clone
* Add information about server
* require ‘capistrano/ext/multistage’
* set :stages, [“staging”, “production”]
* set :default\_stage, “staging”
* Next, create a directory called deploy inside your app’s config directory.
* Then add production.rb and staging.rb files to it.
* You’ll need one Ruby file per stage you’ve configured, and they need to be named the same so that Capistrano can load the appropriate files when you specify which stage you want to deploy.
* Now, populate the production.rb settings:
* server "my\_fancy\_server.com", :app, :web, :db, :primary => true
* set :deploy\_to, "/var/www/fancy\_shoes"
* Then, staging.rb:
* server "my\_fancy\_server.com", :app, :web, :db, :primary => true
* set :deploy\_to, "/var/www/fancy\_shoes\_staging"

**Validating the installation:**

* Run the following command from the root directory of your application:
* cap deploy:setup
* cap deploy:check

**Deployment code for staging:**

* cap deploy

**Deployment code for production:**

* cap production deploy
* For every deployment instead of doing an entire clone/export, use this command:
* svn up

or

* git pull