

Documentation détaillée pour le déploiement de Blacklight avec RubyOnRails

Pour avoir une vue d'ensemble de cette structure complexe :

- Git, Capistrano sont installés sur le serveur de développement disposant d'un environnement Ruby avec Blacklight. Les commandes de déploiement seront exécutées sur ce dernier également.
- Apache2, Passenger sont installés sur le serveur de production équipés d'un environnement ruby identique au premier et d'une base de données prête à l'emploi.

Ce document décrit l'installation de passenger et de capistrano. Ainsi que le déploiement de blacklight.

Installation de passenger sous debian pour ruby 1.9.1 sur le serveur de production

Installation :

A partir de ce tutoriel :

http://t37.net/installer-passenger-mod_rails-sous-debian.html

- \$ gem install passenger
- \$ cd [/var/lib/gems/1.9.1/gems/passenger-3.0.17/bin](#)
- \$./passenger-install-apache2-module
- \$ echo "LoadModule passenger_module [/var/lib/gems/1.9.1/gems/passenger-3.0.17/ext/apache2/mod_passenger.so](#)" > "/etc/apache2/mods-available/passenger.load"
- \$ echo "PassengerRoot [/var/lib/gems/1.9.1/gems/passenger-3.0.17](#)
PassengerRuby [/usr/bin/ruby1.9.1](#)
PassengerDefaultUser www-data" > "/etc/apache2/mods-available/passenger.conf"

Configuration du virtualHost :

```
<VirtualHost *:80>
  ServerName www.moncatalogue.fr
  DocumentRoot /var/www/blacklight/current/public
  RailsBaseUri /

  <Directory />
    Order deny,allow
    Deny from All
    AllowOverride None
    Options FollowSymLinks
  </Directory>
```

```
<Directory /var/www/blacklight/current/public>
  AllowOverride none
  Options -MultiViews -Indexes -FollowSymLinks +SymLinksIfOwnerMatch
  Order allow,deny
  Allow from all
</Directory>
</VirtualHost>
```

Pour redémarrer l'application :

- pour redémarrer l'application sans relancer apache il faut créer un fichier restart.txt dans le répertoire tmp du serveur web.

Installation de Capistrano

Installation :

On l'installe en décommentant dans le fichier Gemfile de Blacklight la ligne suivante :

```
gem 'capistrano'
```

Puis en exécutant cette commande dans le répertoire de BL :

```
$ bundle install
```

Configuration :

Dans le répertoire de blacklight, taper cette commande qui crée les fichiers de configuration de capistrano :

```
$ capify .
```

Il faut maintenant renseigner les fichiers de configuration de Capistrano (en exemples à la fin de ce fichier) dans lesquels on renseigne l'entrepôt Git du serveur de développement et l'IP du serveur de production.

Déploiement de Blacklight

On suppose que votre entrepôt git est créé dans le répertoire de Blacklight sur le serveur de développement. Le fichier « database.yml » contient également une entrée production avec les coordonnées de la base de données utilisée.

Si vous utilisez le gem zoom, il y a un problème de compilation avec ruby 1.9.1. Si ce gem est installé correctement sur votre environnement de développement, vous devrez faire ces commandes suivantes pour activer zoom :

```
$ bundle package
```

```
$ bundle install --local
```

```
$ bundle exec rake assets:precompile (pas nécessaire si ligne décommentée dans capfile.)
```

Déploiement :

Dans le répertoire de blacklight :

```
$ cap deploy:setup (pour créer la structure de répertoire sur le serveur d'arrivée).
```

```
$ cap deploy
```

Durant ce processus, capistrano vous demandera de vous identifier et signalera d'éventuelles erreurs ou gem manquants.

Pour les mises à jour suivantes :

Cette commande suffira à mettre à jour Blacklight :

```
$ cap deploy
```

Exemple de fichier Capfile :

```
load 'deploy'  
# Uncomment if you are using Rails' asset pipeline  
load 'deploy/assets'  
load 'config/deploy' # remove this line to skip loading any of the  
default tasks
```

Exemple de fichier deploy.rb présent dans le répertoire config de BL :

```
require 'bundler/capistrano'  
set :application, "ip du serveur d'arrivée"  
set :scm, :git  
set :repository, "file://."  
set :branch, "master"  
set :deploy_via, :copy  
set :checkout, 'export'  
set :keep_releases, 5  
# Or: `accurev`, `bZR`, `cvs`, `darcs`, `git`, `mercurial`, `perforce`,  
`subversion` or `none`  
  
role :web, "ip du serveur d'arrivée" # Your HTTP  
server, Apache/etc  
  
role :app, "ip du serveur d'arrivée" # This may be the  
same as your `Web` server  
  
role :db, "ip du serveur d'arrivée", :primary => true # This is where Rails  
migrations will run  
  
default_run_options[:pty] = true  
ssh_options[:forward_agent] = true  
set :deploy_to, "/var/www/blacklight"
```

```
set :user, "user"
set :use_sudo, false
# If you are using Passenger mod_rails uncomment this:
namespace :deploy do
  task :start do ; end
  task :stop do ; end
  task :restart, :roles => :app, :except => { :no_release => true } do
    run "#{try_sudo} touch #{File.join(current_path, 'tmp', 'restart.txt')}"
  end
end
end
```