

Centos: Mise en oeuvre de RUBY

Table of Contents

- [Centos: Mise en oeuvre de RUBY](#)
- [Installation de ruby2](#)
 - [Activer le dépôt Software Collections \(SCol\),](#)
 - [Installer ruby2](#)
- [Gestion des versions](#)
 - [Configuration de la version par défaut de ruby.](#)
 - [Utiliser scl](#)
 - [Exécution d'une commande dans un contexte](#)
 - [Exécution du shell bash dans un contexte](#)
 - [Utiliser un manager](#)
 - [Présentation des Manager](#)
 - [Intégration de RVM](#)
- [Mise en oeuvre d'un entrepôt local de gems](#)
 - [Utilisation du serveur gem intégré](#)
 - [Le projet Gem in a Box](#)
 - [installer geminabox:](#)
 - [Créer un répertoire de données pour stocker les gems](#)
 - [Créer un fichier config.ru à la racine du dépôt \(rubygems\)](#)
 - [Lancer le serveur](#)

:

Ruby est un langage de programmation interprété, orienté objet. Il existe de nombreuses bibliothèques de fonctionnalités adjoignables au langage qui contiennent de nombreuses fonctionnalités supplémentaires, mais aussi des extensions.

Ruby on Rails, également appelé RoR ou Rails, est un framework web libre écrit en Ruby. Il suit le motif de conception modèle-vue-contrôleur (MVC). Il propose une structure qui permet de développer rapidement et intuitivement. Cependant, il impose un grand niveau d'abstraction dans la programmation qui apporte en contre-partie l'économie d'écrire soi-même la plupart des routines obligatoires d'une application web.

Installation de ruby2

Activer le dépôt Software Collections (SCol),

Red Hat a mis à disposition pour CentOS une sélection de paquets du projet COPR initié par Fedora au travers du nouveau dépôt Software Collections, pour activer ces dépôt :

- Activer le dépôt extras dans CentOS à partir du fichier `/etc/yum.repos.d/CentOS-Base.repo` en passant la directive `enabled` à 1,
- Installer le paquet **centos-release-scl**

```
yum install centos-release-scl-rh
```

Installer ruby2

```
yum install ruby200 ruby200-devel
```

Gestion des versions

Une installation de Ruby correspond à un ensemble de programmes (gems et utilitaires liés à Ruby) associés à une version spécifique de Ruby.

Il est donc possible pour les applications d'utiliser une version embarquée de Ruby indépendante de celle du système, deux « installations » totalement indépendantes de Ruby seront ainsi présentes sur le même système.

Plusieurs outils/ astuces permettent de gérer ces multiples versions

Configuration de la version par défaut de ruby.

Il est possible de configurer une version spécifique de Ruby en tant que version par défaut.

Par défaut la version de ruby est celle installée sur la station:

```
which ruby

/usr/bin/ruby.
```

Pour utiliser la version Ruby du kit de développement Chef par défaut, modifier les variables d'environnement `$PATH` et `GEM` pour inclure les chemins d'accès au kit de développement Chef.

- Si Bash est le shell du poste de travail, exécuter: `eval "$ (chef shell-init bash)"`
- Si zsh est le shell du poste de travail, exécuter: `eval "$ (chef shell-init zsh)"`

La commande **which** doit retourner la version de de ruby embarquée avec chef

```
which ruby

/opt/chefdk/embedded/bin/ruby
```

Pour rendre ces modifications permanentes

- Si Bash est le shell du poste de travail, exécuter: `echo 'eval "$ (chef shell-init bash)"' >> ~/.bash_profile`
- Si zsh est le shell du poste de travail, exécuter: `echo 'eval "$ (chef shell-init zsh)"' >> ~/.zshrc`

Utiliser scl

L'utilitaire **scl** permet d'exécuter une application qui ne se trouve pas dans la hiérarchie racine du système de fichiers mais qui est présente dans un autre emplacement. Cette application peut toujours utiliser et bénéficier de logiciels installés dans le système de fichiers racine. Afin de laisser une application visible au système, il faut utiliser l'utilitaire **scl** comme interface.

Exécution d'une commande dans un contexte

```
scl enable example 'less --version'
```

exécute la commande **less --version** dans l'environnement avec la collection 'example' activée
par exemple

```
scl enable ruby200 'gem install geminabox-1.1.0.gem'
```

exécute **gem install** dans l'environnement **ruby2.0.0**

Exécution du shell bash dans un contexte

```
scl enable à foo bar bash
```

exécute l'instance bash avec les collections de logiciels foo et bar activées
par exemple pour exécuter l'instance bash dans l'environnement ruby2.0.0

```
scl enable ruby200 bash
```

la commande **ruby -v** doit retourner la version de ruby2

```
ruby 2.0.0p645 (2015-04-13) [x86_64-linux]
```

la commande **which ruby** doit retourner le path de la library ruby200

```
/opt/rh/ruby200/root/usr/bin/ruby
```

Utiliser un manager

L'utilisation d'un manager pour alterner entre les versions de Ruby, permet d'installer des gems différents pour ces versions de Ruby. Des «installations» totalement indépendantes de Ruby seront ainsi présentes sur le système.

Présentation des Manager

- **chruby**: permet de basculer entre différentes installation de Ruby. chruby peut gérer les installations de ruby-install ou même les installations faites à partir des sources.
- **rbenv**: permet de gérer différentes installations de Ruby. Il ne permet pas d'installer Ruby, mais il existe un plugin très utilisé s'appelant ruby-build qui s'occupe de l'installation de Ruby. Tous deux sont disponibles pour les systèmes d'exploitation OS X, Linux, ou autres systèmes basés sur UNIX.
- **RVM ("Ruby Version Manager")**: permet d'installer et de gérer plusieurs installations de Ruby sur un système. Il peut aussi gérer différent gemsets. Il est disponible pour les systèmes d'exploitation OS X, Linux, ou autres systèmes basés sur UNIX.
- **uru**: est un outil en ligne de commande léger et multi-plateforme qui aide à utiliser différentes installation de Ruby sur les systèmes OS X, Linux, ou Windows.

Intégration de RVM

Télécharger l'archive tar de rvm sur <https://github.com/rvm/rvm/tarball/stable>

Transférer l'archive sur le serveur

```
scp rvm-rvm-1.29.3-0-gad6e58e.tar.gz root@xx.xx.xxx.xx:/usr/local/
```

Créer le répertoire rvm:

```
mkdir rvm
```

Déballer le fichier tar

```
tar -xzf ../rvm-stable.tar.gz
```

Installer rvm:

```
cd ../rvm-rvm-ad6e58e/  
./install --auto-dotfiles
```

```
Installing RVM to /usr/local/rvm/
```

```
Installation of RVM in /usr/local/rvm/ is almost complete:
```

```
* First you need to add all users that will be using rvm to 'rvm' group,
```

```
and logout - login again, anyone using rvm will be operating with `umask
u=rwx,g=rwx,o=rx`.
* To start using RVM you need to run `source /etc/profile.d/rvm.sh`
  in all your open shell windows, in rare cases you need to reopen all shell
  windows.
```

Charger rvm

```
source /usr/local/rvm//scripts/rvm
```

Mise en oeuvre d'un entrepôt local de gems

Diverses méthodes permettent d'héberger et partager des gems RUBY.

Utilisation du serveur gem intégré

L'installation de RubyGems, intègre un serveur gem. C'est le moyen le plus rapide pour héberger des gemmes. Il suffit de lancer la commande:

```
gem server
```

Cela servira toutes les gems installés dans la machine locale dans. L'URL <http://localhost:8808> affiche un index de documentation HTML fournit par la commande gem server.

Lorsqu'on installe de nouvelles gemmes, elles sont automatiquement disponibles via le serveur gem intégré.

Le projet Gem in a Box

Pour un serveur avec plus de fonctionnalités, y compris la possibilité de pousser des gems.

installer geminabox:

```
gem install geminabox
```

Créer un répertoire de données pour stocker les gems

```
mkdir -p /usr/share/nginx/html/repo/pub/rubygems/gems
```

Créer un fichier config.ru à la racine du dépôt (rubygems)

```
vi /usr/share/nginx/html/repo/pub/rubygems/config.ru

require "rubygems"
require "geminabox"
Geminabox.data = "/usr/share/nginx/html/repo/pub/rubygems"
run Geminabox::Server
```

Lancer le serveur

```
rackup -d /usr/share/nginx/html/repo/pub/rubygems

[2018-05-28 16:21:22] INFO WEBrick 1.3.1
[2018-05-28 16:21:22] INFO ruby 1.8.7 (2013-06-27) [x86_64-linux]
[2018-05-28 16:21:27] INFO WEBrick::HTTPServer#start: pid=18565 port=9292
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=centos:ruby-install>

Last update: **2025/02/19 10:59**

