

Vagrant sur Raspberry pi

Table of Contents

- [Vagrant sur Raspberry pi](#)
- [Préparation de Raspbian](#)
- [Installation de FOG](#)
 - [Installer fog-core](#)
 - [Installer fog](#)
- [Installation et configuration de vagrant](#)
 - [Installation de vagrant depuis les dépôts](#)
 - [Installation de Vagrant à partir de la source](#)
 - [Installer les plugins](#)
 - [Installation de vagrant-libvirt](#)
 - [Installation de knife-solo](#)

Vagrant est un logiciel libre et open-source pour la création et la configuration des environnements de développement virtuel.

Depuis la version 1.63.4, **Vagrant** fournit un support natif des conteneurs Docker à l'exécution, au lieu d'un système d'exploitation entièrement virtualisé. Cela permet de réduire les frais généraux puisque Docker utilise des conteneurs Linux légers.

Depuis la version 1.1, **Vagrant** n'impose plus VirtualBox, mais fonctionne également avec d'autres logiciels de virtualisation tels que KVM/libvirt.

Afin de permettre à **Vagrant** de contrôler et de pourvoir les machines à l'aide de Libvirt il faut installer :

- Le fournisseur de service cloud: **fog**
- Le greffon ajoutant un fournisseur Libvirt à Vagrant: **Vagrant-libvirt**

Préparation de Raspbian

Télécharger puis graver une image **Raspbian** sur une carte SD en utilisant l'une des méthodes décrites [ici](#) pour graver l'image sur la carte SD.

Afin de permettre la construction (build) de **fog** installer **libxml2-devel** et **libcurl-devel**

```
apt-get install libxml2-dev libcurl-dev ruby -y
```

Installation de FOG

Fog est la bibliothèque de services cloud de Ruby.

- Les collections fournissent une interface simplifiée, permettant le travail et la manipulation dans le cloud.
- Les requêtes permettent aux utilisateurs expérimentés de tirer le meilleur parti des fonctionnalités de chaque cloud.
- Les mocks (objets simulés qui reproduisent le comportement d'objets réels) permettent de tester le comportement d'objets réel.



Pour une intégration avec **chef**, l'installation doit être opérée dans l'environnement **ruby** embarqué dans **chef**. Pour cela basculer l'environnement de ruby, avec la commande suivante:

```
eval "$(chef shell-init bash)"
```

Installer fog-core

Fog-core nécessite Ruby 2.0.0 ou une version ultérieure.



La prise en charge de **Ruby 1.8** et **1.9** a été abandonnée dans **fog-v2.0.0** en tant que modification rétrocompatible. Il faut utiliser les dernières versions de **fog 1.x** lorsqu'on a besoin du support **1.8.7** ou **1.9.x**.

Afin de satisfaire les dépendances installer les versions 2.1 e 1.45.0 de fog-core

```
gem install fog-core -v 2.1
gem install fog-core -v 1.45.0
```

Installer fog

```
gem install fog
```

Installation et configuration de vagrant



Les plugins **Vagrant** sont configurés en fonction de l'environnement **Ruby** actuel. Si les plugins sont installés à l'aide de **Vagrant** à partir de la source, ils ne fonctionneront pas à partir de l'installation de **Vagrant** basée sur le package.

Installation de vagrant depuis les dépôts

```
apt install vagrant
```

Installation de Vagrant à partir de la source

Il faut avoir un Ruby moderne pour développer et construire **Vagrant**. La version spécifique de Ruby est documentée dans la gemspec de Vagrant.



Il est conseillé de ne pas utiliser le **Ruby** du système - utiliser un gestionnaire de version **Ruby** comme **rvm** ou **chruby**.

Cloner le référentiel de **Vagrant** de GitHub dans un répertoire spécifique:

```
git clone https://github.com/hashicorp/vagrant.git
```

Ensuite, cd dans ce chemin. Toutes les commandes seront exécutées à partir de ce chemin :

```
cd /chemin/vers/vagrant/clone
```

Exécuter la commande **bundle** pour installer la configuration requise :

```
bundle install
```

On peut maintenant exécuter **Vagrant** en exécutant `bundle exec vagrant` depuis ce répertoire.



Pour utiliser la version de **Vagrant** installée localement dans d'autres projets, il faut créer un **binstub** et l'ajouter au PATH.

Tout d'abord, exécuter la commande suivante à partir du dossier **Vagrant** : `bundle --`

```
binstubs exec
```

Cela générera des fichiers dans `exec/`, y compris **vagrant**. On peut désormais spécifier le chemin complet vers `exec/vagrant` n'importe où sur le système d'exploitation :
`/chemin/vers/vagrant/exec/vagrant init -m hashicorp/bionic64`



Il y aura des avertissements indiquant que l'exécution de Vagrant comme celle-ci n'est pas prise en charge.

Afin de ne pas spécifier le chemin complet vers **Vagrant** (pour simplement exécuter **vagrant**), créer un lien symbolique vers l'exec :

```
ln -sf /chemin/vers/vagrant/exec/vagrant /usr/local/bin/vagrant
```

Installer les plugins

Installation de vagrant-libvirt

```
vagrant plugin install vagrant-libvirt --plugin-version="0.0.42"
```

Installation de knife-solo

Installation avec ruby gem standard:

```
gem install knife-solo  
  
# ou si on utilise ChefDK  
chef gem install knife-solo
```



Les commandes de knife intégrées pour travailler avec des sacs de données ne fonctionnent pas bien sans un serveur Chef, il est donc recommandé d'utiliser la gem `knife-solo_data_bag`, pour fournir des versions "solo" de toutes les commandes de sac de données typiques. La structure par défaut générée par **knife solo init** doit être compatible avec toutes les opérations répertoriées dans la documentation de cette gem.

Installation à partir de la source git:

```
bundle && bundle exec rake install
```

From:

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

Permanent link:

<http://www.ouarte.garden/doku.php?id=prive:rpi-vagrant>

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

