

VAGRANT: Provisionnement Chef Server

Table of Contents

- [VAGRANT: Provisionnement Chef Server](#)
- [Créer et initialiser Vagrantfile](#)
 - [Création de l'objet config](#)
 - [Définir les options de de soclage](#)
 - [Définir les options réseau](#)
 - [Création et configuration du bloc chef_client](#)
 - [Définition de l'URL du serveur Chef](#)
 - [Définition des options de validation](#)
 - [Définition du chemin de la clé de validation](#)
 - [Personnaliser les éléments d'identification du noeud](#)
- [Définir les options de provisionnement](#)
 - [Adapter le réseau](#)
 - [Spécifier la liste d'exécution \(run list\)](#)
 - [Spécifier l'environnement](#)
- [Nettoyer la configuration](#)

Chef Server permet de provisionner une machine virtuelle sans avoir à conserver les livres de recettes dans le référentiel lui-même. Cette approche présente divers avantages, tels que la possibilité d'utiliser des livres de recettes de production à partir du serveur Chef pour approvisionner un environnement de développement.

Créer et initialiser Vagrantfile

Le fichier Vagrant est écrit dans un langage DSL (Ruby Domain Specific Language) pour la configuration des machines virtuelles Vagrant.

Le minimum requis pour provisionner à l'aide du client Chef consiste à fournir l'URL du serveur Chef, ainsi que le chemin d'accès à la clé de validation, afin que le nœud puisse s'enregistrer auprès du serveur Chef. Le nœud sera enregistré avec le serveur Chef spécifié.

Création de l'objet config

Vagrant utilisera cet objet config pour configurer la machine virtuelle:

```
Vagrant.configure ("2") do |config|  
  ...  
end
```

Définir les options de de soclage

Dans le bloc de configuration, indiquer à Vagrant quelle image de machine virtuelle utiliser pour démarrer le nœud:

```
config.vm.box = "dgfip/centos7"  
config.vm.box_url =  
"http://xx.xx.xxx.xxx:8089/boxes/vagrant/boxes/dgfip-rhel7-0.4.0.box"
```

Définir les options réseau

```
config.vm.define "dgfip-rhel7"  
config.vm.network "private_network", libvirt__network_name: "br0"
```

Création et configuration du bloc chef_client

Après avoir sélectionné l'image de la machine virtuelle à démarrer, configurer la mise en service de la boîte à l'aide de Chef. La configuration de Chef se fait dans un bloc Ruby imbriqué:

```
config.vm.provision: chef_client do |chef|  
  
end
```

Dans ce bloc on doit indiquer à Vagrant comment connecter le nœud virtuel au serveur Chef.

Tout d'abord, indiquer à Vagrant où stocker tous les éléments Chef sur le nœud:

```
chef.provisioning_path = "/etc/chef"
```

Définition de l'URL du serveur Chef

La première étape du provisionnement avec le serveur chef consiste à indiquer à Vagrant l'emplacement du serveur. Ceci est fait ci-dessous:

Indiquer à vagrant le point de terminaison de l'API du serveur Chef en utilisant le nom de l'organisation qui a été créée dans le compte sur Hosted Chef.

```
chef.chef_server_url = "https://xx.xx.xxx.xxx:10443/organizations/dgfip/"
```

Définition des options de validation

Adapter les options de validation à celles définies lors de la création de l'organization:

```
chef.validation_client_name = "jacques"
```

```
chef.client_key_path = "/etc/chef"
```

Définition du chemin de la clé de validation

Le serveur Chef utilise des paires de clés afin de vérifier et d'enregistrer des nœuds auprès du serveur Chef (comme pour l'authentification basée sur une clé SSH). La clé de validation est utilisée par un client non enregistré pour se vérifier et s'enregistrer auprès du serveur Chef. Vagrant doit connaître le chemin d'accès à cette clé de validation afin de configurer le client pour le serveur Chef.

Indiquer à vagrant où trouver la clé privée téléchargée lors de la création de l'organisation sur Hosted Chef:

```
chef.validation_key_path = "/etc/chef/jacques-validator.pem"
```

Personnaliser les éléments d'identification du nœud

Indiquer à Vagrant comment nommer le nœud:

```
chef.node_name = "serveur"
```

Définir les options de provisionnement

Adapter le réseau

Le nœud est fourni avec deux interfaces:

- eth0 pour le management en dhcp sur le réseau 192.168.121.0/24
- eth1 pour le wan avec l'adresse par défaut xx.xx.xxx.xxx/24

Il faut modifier l'adresse de l'interface eth1 pour lui donner une autre adresse (sinon toutes les machines auront la même adresse) ainsi que la route par défaut (GATEWAY).

```
config.vm.provision "shell", run: "always", inline: <<-SHELL
```

```
ifconfig eth1 xx.xx.xxx.xxx netmask 255.255.255.0 up  
route del default gw 192.168.121.1  
route add default gw xx.xx.xxx.x
```

```
SHELL
```



La syntaxe du script shell est importante :

* les délimiteurs **EOF** et **SHELL** ne doivent ni être précédés ni suivis d'aucun caractère



* le saut de ligne après «-**SHELL** et après «-**EOF** est nécessaire

Spécifier la liste d'exécution (run list)

La liste d'exécution est la liste des tâches à exécuter sur le nœud, qui sont des recettes et/ou des rôles. Habituellement, la liste d'exécution est gérée par le serveur Chef. Dans ce cas, vous n'avez rien à faire, car par défaut, Vagrant extraira la liste des exécutions du serveur du chef. Mais on peut spécifier directement la liste des tâches en utilisant les aides fournies par la configuration, qui s'expliquent assez bien:

```
Vagrant::Config.run do |config|
  config.vm.provision :chef_client do |chef|
    # Provision with the apache2 recipe
    chef.add_recipe("apache2")

    # Provision with the database role
    chef.add_role("database")
  end
end
```

Si on doit accéder directement à la liste d'exécution, on peut également utiliser l'accesseur `run_list` :

```
Vagrant::Config.run do |config|
  config.vm.provision :chef_client do |chef|
    # Modifying the run list directly
    chef.run_list = ["recipe[foo]", "role[bar]"]
  end
end
```

Spécifier l'environnement

Depuis Chef 0.10 on peut configurer l'environnement dont un nœud fait partie, en utilisant l'option de configuration de l'environnement :

```
Vagrant::Config.run do |config|
  config.vm.provision :chef_client do |chef|
    # Set the environment for the chef server
    chef.environment = "development"
  end
end
```

Nettoyer la configuration

Lorsqu'on configure une machine virtuelle Vagrant avec le serveur Chef, il crée une nouvelle entrée de nœud Chef et une entrée Client Chef sur le serveur Chef, à l'aide du nom d'hôte de la machine (par exemple, vagrant-ubuntu-oneiric). Après avoir démonté la machine virtuelle, on doit explicitement supprimer ces entrées du serveur Chef avant d'en configurer une nouvelle avec le serveur Chef. Par exemple, en utilisant knife :

```
$ knife node delete vagrant-ubuntu-oneiric  
$ knife client delete vagrant-ubuntu-oneiric
```

Si on ne le fait pas, on obtiendra le message d'erreur suivant lorsqu'on tentera de provisionner une nouvelle machine virtuelle avec le même nom d'hôte:

```
HTTP Request Returned 409 Conflict: Client already exists.
```

From:

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

Permanent link:

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

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

