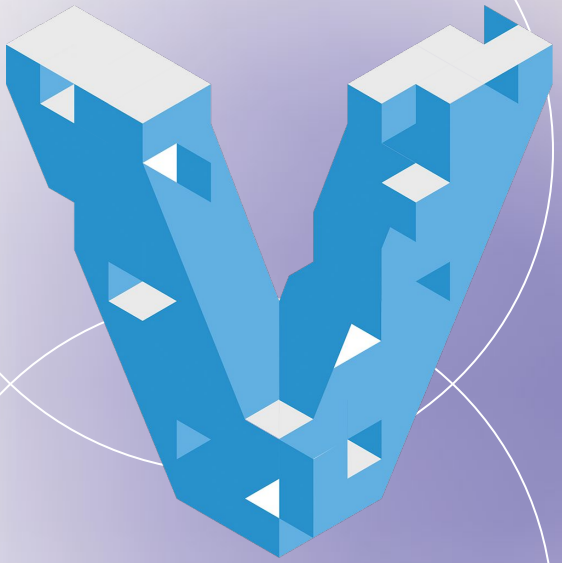

Vagrant

Table of contents

Introduction

Depuis quelques années, les VMs sont là pour répondre à des problématiques. Néanmoins, ces outils ne sont pas toujours simples à mettre en place. C’est à cet instant que Vagrant entre en scène.

01

Définition Vagrant

Définition Vagrant

Vagrant est un des outils en ligne de commande qui permet de virtualiser rapidement des machines en utilisant la plupart des hyperviseurs comme HyperV, Libvirt, Parallels, VMWare et VirtualBox.

Définition Vagrant

Concrètement, vous avez besoin rapidement de trois machines virtuelles tournant respectivement sous des OS différents, Vagrant vous permettra de les instancier en quelques minutes via un simple fichier Vagrantfile.

Intérêts Vagrant

Les intérêts de vagrant sont multiples :

- Disposer d’un environnement de développement sur un système d’exploitation différent de sa propre machine.
- Avoir plusieurs environnements de développement sans modifier la configuration de son propre système (certains projets nécessitent des configurations système qui les rendent incompatibles entre eux).

Intérêts Vagrant

- Monter rapidement un environnement de développement complet
- certains outils sont plus faciles à installer sous GNU/Linux

02

Avantages Vagrant

Avantages Vagrant

L’avantage principal de Vagrant est qu’il ne repose que sur un simple fichier de configuration qui définit le système à utiliser et la configuration à appliquer. Les autres avantages sont :

- Outils très simple
- Les VMs sont portables

Avantages Vagrant

- Moins lourd que les VMs classiques.
- Facilité à dupliquer et/ou reproduire un environnement autant de fois qu’on le souhaite.

03

Installation Vagrant

Installation Vagrant (debian/ubuntu)

1. **curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -**
2. **sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs) main"**
3. **sudo apt-get update**
4. **sudo apt-get install vagrant**

Installation Vagrant(redhat/oraclelinux/centos)

1. **sudo yum install -y yum-utils**
2. **sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo**
3. **sudo yum -y install vagrant**

04

Utilisation Vagrant

Utilisation Vagrant

Vagrant repose sur des boxes. Ce sont les images des OS qui sont utilisées pour la virtualisation (ex: VirtualBox).Les commandes pour gérer les boxes sont :

ajouter une box (l'URL est optionnelle pour les boxes officielle)

vagrant box add nom_de_ma_box url_de_la_box

Utilisation Vagrant

supprimer une box

vagrant box remove nom_de_ma_box

lister les boxes installées

vagrant box list

Utilisation Vagrant

Un des autres intérêts de Vagrant est qu'il existe déjà de nombreuses boxes toutes prêtes à être utilisées mais également de nombreux projets Vagrant proposant des boxes et/ou des configurations pour différents environnements de développement (Java EE, monitoring, PHP, etc...). Il est également possible de créer une propre box.

Utilisation Vagrant

Réutiliser une box officielle avec la commande suivante:

vagrant box add centos/8

Cette commande va prendre un peu de temps car elle va télécharger l'image de l'OS.

Utilisation Vagrant

Une fois l'image téléchargée, elle est utilisable. Pour cela, il faut créer un répertoire pour votre projet et s'y placer dedans et enfin initialiser Vagrant avec la commande suivante:

vagrant init centos/8

Cette commande va simplement créer le fichier de configuration Vagrant pour le projet .

Utilisation Vagrant

Une fois l’environnement initialisé (et éventuellement configuré), il existe quelques commandes pour manipuler l’instance de la VM :

Utilisation Vagrant

- Démarrer et configurer la VM

  vagrant up

- Se connecter en SSH à la VM

  vagrant ssh

- Vérifier l'état des VM

  vagrant status

Utilisation Vagrant

- Mettre en pause la VM

  vagrant suspend

- Sortir de pause la VM

  vagrant resume

- Arrêter la VM

  vagrant halt

- Pour détruire la VM

  vagrant destroy.

Vagrantfile

Ce fichier de déclaration Vagrantfile utilise un langage proche de ruby.

Syntaxe pour les fichiers Vagrantfile

- Les commentaires sur une seule ligne commencent avec un #
- Les commentaires sur plusieurs lignes sont encadrés par /* et */
- Les variables Vagrantfile sont assignées avec la syntaxe key = value (aucune importance concernant les espaces).

Syntaxe pour les fichiers Vagrantfile

- Les valeurs peuvent être des primitives string, number ou boolean ou encore une list ou une map.
- Les chaînes de caractères sont encadrées par des “doubles-quotes”. Les valeurs booléennes peuvent être true ou false.

Premières lignes de Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.box = "hashicorp/precise32"
end
```

On configure avec la version (“2”) de vagrant un objet config qui utilise une image ubuntu precise 32 bits.

Premières lignes de Vagrantfile

On peut configurer plus finement la vm avec différentes sections :

- **config.vm** : pour paramétrer la machine
- **config.ssh** : pour définir commande vagrant accédera à votre machine
- **config.vagrant** : pour modifier le comportement de vagrant en lui-même.

Paramétrage de la VM config.vm

```
Vagrant.configure("2") do |config|
  config.vm.box = "ol8"
  config.vm.provider "libvirt" do |hv|
    hv.cpus = "1"
    hv.memory = "512"
  end
  config.vm.synced_folder '.', '/vagrant', disabled: true

  config.vm.define "host1" do |host1|
    host1.vm.network :private_network, ip: "192.168.3.10"
    host1.vm.hostname = "host1"
  end
end
```

Paramétrage de la VM config.vm

provider

Indiquer l'hyperviseur utilisé. On peut utiliser une boucle pour configurer proprement chaque hyperviseur.

network

On configure ici la partie réseau. Dans notre exemple, on fixe l'adresse ip.

Paramétrage de la VM config.vm

libvirt: est la bibliothèque utilisée pour gérer les machines virtuelles. Elle peut fonctionner avec de nombreux hyperviseurs, dont VirtualBox, VMWare Workstation, QEMU... Elle fait le lien entre le noyau, responsable de l'allocation des ressources (CPU, RAM, réseau...), et l'hyperviseur de la machine virtuelle

Paramétrage de la VM config.vm

provision

Cette partie permet de définir ce qu'il se passe une fois que la machine est démarrée. On a le choix d'utiliser langage dont les plus courants sont le shell, ansible et bien d'autres encore.

Provisioning sur une VM déjà instanciée

Pour relancer le playbook sur une VM déjà démarrée il suffit d'utiliser l'option --provision.

```
vagrant up --provision
```

Merci ...