

Cours Majeur Virtualisation

Nantes Ynov Campus – 2022-2023

Activité Pratique 5

Découverte et déploiement de VM avec Vagrant



Objectif :

Tester et découvrir pratiquement un outil de déploiement et provisionnement de machines virtuelles

Présentation :

Vagrant est une solution Open Source créée pour la configuration automatique de machine virtuelle et de conteneur. Vagrant permet par exemple de configurer la quantité de ram, la quantité de CPU, les paramètres réseaux et le nom d'hôte. Il peut ensuite automatiquement exécuter des scripts de gestion de configuration (Ansible, chef, Puppet, etc...). Ces scripts vont permettre l'installation et la configuration des paquets via le protocole SSH. La première version est sortie le 8 mars 2010 et était développée par Mitchell Hashimoto et John Bender en Ruby.

Vagrant est par défaut compatible avec les services de virtualisations suivants :

- Virtual box
- Hyper-V
- Docker
- VMWare
- Amazon EC2

Il est possible d'ajouter des providers grâce à des plugins.

Installation

Pour installer Vagrant, il faut se rendre sur le site officiel (<https://www.vagrantup.com/downloads.html>), télécharger l'exécutable de votre système

et l'exécuter. Le but est d'avoir accès à la commande « Vagrant » via la console (sur Windows il faudra donc ajouter Vagrant à votre « global path »).

Les commandes

Pour utiliser Vagrant il faut obligatoirement créer un « projet », ces derniers sont appelés « boxes ». Une liste de box partagé par la communauté est accessible sur le site :

<https://app.vagrantup.com/boxes/search>

Ci-dessous une liste non exhaustive de quelques commandes utiles pour commencer ces premiers pas :

```
# Afficher l'aide
vagrant -h
# Afficher l'aide sur une commande
vagrant COMMANDE -h
# Lister les boxes présentent sur le système
vagrant box list
# Ajouter une box
vagrant box add nom_de_ma_box url_de_la_box
# Supprimer une box
vagrant box remove nom_de_ma_box
# Finalisation du projet vagrant
vagrant package
# initialisation d'un nouveau projet vagrant
vagrant init nom_de_la_box
# Lancer la machine virtuelle
vagrant up
# Pour tester si la machine fonctionne, on se connecte en SSH
vagrant ssh
```

Configuration

La configuration du projet Vagrant se fait dans le fichier « Vagrantfile ».
Voici quelques exemples de configurations :

```
NODES = [  
  {:hostname => "node1", :ip => "192.168.72.10", :cpus => 1, :mem => 1024 },  
  {:hostname => "node2", :ip => "192.168.72.20", :cpus => 1, :mem => 1024}  
]
```

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

```
  config.vm.box = "debian/stretch64"
```

```
  NODES.each do |node|
```

```
    config.vm.define node[:hostname] do |cfg|
```

```
      cfg.vm.hostname = node[:hostname]
```

```
      cfg.vm.network "private_network", ip: node[:ip]
```

```
      cfg.vm.provider "virtualbox" do |v|
```

```
        v.customize ["modifyvm", :id, "--cpus", node[:cpus] ]
```

```
        v.customize ["modifyvm", :id, "--memory", node[:mem] ]
```

```
        v.customize ["modifyvm", :id, "--natdnshostresolver1", "on"]
```

```
        v.customize ["modifyvm", :id, "--natdnsproxy1", "on"]
```

```
        v.customize ["modifyvm", :id, "--name", node[:hostname] ]
```

```
      end
```

```
    end
```

```
  end
```

```
  #provisionnement de la VM
```

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

```
    apt-get update
```

```
    apt-get install -y ansible
```

```
  SHELL
```

```
end
```

La partie « nodes » permet de configurer un tableau de machine à créer avec des variables personnalisées, ici, on configure leurs noms d'hôtes, leurs IP, la RAM et l'image à utiliser.

```
NODES = [
  {:hostname => "node1", :ip => "192.168.72.10", :cpus => 1, :mem => 1024 },
  {:hostname => "node2", :ip => "192.168.72.20", :cpus => 1, :mem => 1024}
]
```

La partie centrale, « vagrant.configure » applique les paramètres du tableau de node indiqué auparavant et crée chaque machine avec les bons paramètres associés.

```
config.vm.box = "debian/stretch64"
NODES.each do |node|
  config.vm.define node[:hostname] do |cfg|
    cfg.vm.hostname = node[:hostname]
    cfg.vm.network "private_network", ip: node[:ip]
    cfg.vm.provider "virtualbox" do |v|
      v.customize ["modifyvm", :id, "--cpus", node[:cpus] ]
      v.customize ["modifyvm", :id, "--memory", node[:mem] ]
      v.customize ["modifyvm", :id, "--natdnshostresolver1", "on"]
      v.customize ["modifyvm", :id, "--natdnsproxy1", "on"]
      v.customize ["modifyvm", :id, "--name", node[:hostname] ]
    end
  end
end
```

La partie « config.vm.provision » permet de configurer un shell à utiliser sur les machines précédemment créées.

```
config.vm.provision "shell", inline: <<-SHELL
  apt-get update
  apt-get install -y ansible
SHELL
```

Gestion de configuration

Le principe de la gestion de configuration est d'avoir un script exécuté via SSH sur les machines virtuelles à la fin de la création de ces dernières par Vagrant. Dans cet exemple nous utiliserons un petit shell pour la configuration mais il tout aussi possible d'utiliser des gestionnaires comme terraform ou ansible.

Exemple :

```
Vagrant.configure("2") do |config|  
  #  
  # Run Ansible from the Vagrant Host  
  #  
  config.vm.provision "ansible" do |ansible|  
    ansible.playbook = "playbook.yml"  
  end  
end
```

La ligne « config.vm.provision "ansible" do |ansible| » indique que cette partie va utiliser le module Ansible de Vagrant.

La ligne « ansible.playbook = "playbook.yml" » indique à Vagrant quel script Ansible exécuté.

Exercice 1 :

- 1 - Télécharger une box centos7
- 2 - Initialiser un nouveau projet vagrant
- 3 - Démarrer un projet vagrant en utilisant la box que vous venez de télécharger
- 4 - Accéder à la box en ssh
- 5 - arrêter la box
- 6 - modifier le fichier afin de lui donner accès à internet
- 7 - redémarrer la box et vérifier que vous êtes maintenant bien connecté
- 8 - mettez à jour les packages de votre distribution
- 9 - installer un serveur web
- 10 - rendre ce serveur web accessible depuis votre poste sur un port spécifique

Exercice 2 :

- 1 - Modifier le fichier précédent pour créer automatiquement deux VM (ubuntu/trusty64)
- 2 - Configurer le ssh afin que chacune d'elles soient accessibles en ssh à partir du nom
- 3 - Installer un serveur web et rendez le accessible depuis votre VM via le port 8080
- 4 - Configurer votre serveur web afin que celui-ci puisse servir une page web qui sera accessible depuis votre répertoire de travail.

Exercice 3 :

- 1 - Modifier votre précédent fichier Vagrantfile pour créer et provisionner automatiquement via un script trois VM (centos7) avec les caractéristiques suivantes (512 Mo, 1 VCPU), des IP fixes avec les services suivants installés (ssh, nginx) sur deux nœuds et (ssh, haproxy) sur un nœud maître
- 2 - Configurer le haproxy pour servir à tour de rôle vos serveurs web sur lequel vous personnaliserez les pages pour identifier le nœud sur lequel on se trouve.