

Utilisation de Vagrant

JAVA & DEVOPS BOOTCAMP

FORMATEUR M. Ngor SECK

Samane corporation: <https://www.samanecorporation.com>
nseck@samanecorporation.com

ngorsecka@gmail.com

Définition

- Vagrant est un outil open-source de gestion de machines virtuelles (VM).
- Il permet de créer et de gérer des environnements de développement **reproductibles**.

Principes de base

A. Vagrantfile

- Fichier de configuration écrit en Ruby.
- Définit la configuration de la machine virtuelle.

B. Boîtes (Boxes)

- Images de machines virtuelles préconfigurées.
- Vagrant utilise des boîtes comme base pour créer des environnements.
- Pour trouver les boxes vagrant : <https://app.vagrantup.com/boxes/search>

C. Commandes principales

- `vagrant init`: initialise un nouveau Vagrantfile dans le répertoire courant.
- `vagrant validate` : fait la invalidation du fichier Vagrantfile
- `vagrant up`: Crée et configure la machine virtuelle.
- `vagrant ssh`: Se connecte à la machine virtuelle via SSH.
- `vagrant halt`: Arrête la machine virtuelle.
- `vagrant destroy`: Supprime la machine virtuelle.
- `vagrant --help` : Pour voir la documentation vagrant

III. Configuration avancée

A. Provisionnement

- Installation automatique de logiciels et de configurations.
- Types de provisionnement: Shell, Ansible, Chef, Puppet, etc.

B. Réseau

- Configuration des ports, IP, et redirections.
- Permet l'accès depuis l'hôte vers la machine virtuelle.

C. Partage de fichiers

- Partage des fichiers entre l'hôte et la machine virtuelle.
- Utile pour le développement de code.

D. Vagrant plugin

- Étendre les fonctionnalités de Vagrant en installant des plugins :
- Support d'autres hyperviseurs

<https://github.com/hashicorp/vagrant/wiki/Available-Vagrant-Plugins>

IV. Cas d'utilisation

A. Environnements de développement

- Création rapide d'environnements isolés.

B. Tests automatisés

- Configuration facile d'environnements de test.

C. Déploiement

- Assure la portabilité des configurations dans les environnements de production.

V. Exemples pratiques

A. Configuration de base

Unset

```
Vagrant.configure("2") do |config|  
  
  config.vm.box = "ubuntu/jammy64"  
  
  config.vm.network "forwarded_port", guest: 80, host: 8080  
  
end
```

B. Exemple de fichier Vagrantfile

Python

```
Vagrant.configure("2") do |config|  
  
  config.vm.box = "ubuntu/jammy64"  
  
  config.vm.network "forwarded_port", guest: 80, host: 8080  
  
  config.vm.network "private_network", ip: "192.168.33.10"  
  
  config.vm.synced_folder "./web", "/var/www/html"  
  
  config.vm.provider "virtualbox" do |vb|  
  
    # Display the VirtualBox GUI when booting the machine  
  
    vb.gui = false  
  
    # Customize the amount of memory on the VM:  
  
    vb.memory = "2048"  
  
    vb.name = "vagrant_test"
```

```
end

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

  apt-get update

  apt-get install -y apache2

SHELL

end
```

C. Exemple 2 de fichier Vagrantfile

```
servers = [

  { :hostname => "node01", :ip => "192.168.1.10", :memory => "2048",
:disks => 2 },

  { :hostname => "node02", :ip => "192.168.1.20", :memory => "2048",
:disks => 2 },

  { :hostname => "node03", :ip => "192.168.1.30", :memory => "2048",
:disks => 2 },

]

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

  config.vm.define "ansible" do |ansible|

    ansible.vm.box = "ubuntu/jammy64"

    ansible.vm.network "private_network", type: "static", ip:
"192.168.99.10"
```

```
    ansible.vm.hostname = "ansible"

    ansible.vm.provider "virtualbox" do |v|

      v.name = "ansible"

      v.memory = 2048

      v.cpus = 2

    end
  end

  ram_client=2048

  cpu_client=2

  config.vm.define "client1" do |client|

    client.vm.box = "ubuntu/jammy64"

    client.vm.network "private_network", type: "static", ip:
"192.168.99.11"

    client.vm.hostname = "client1"

    client.vm.provider "virtualbox" do |v|

      v.name = "client1"

      v.memory = ram_client

      v.cpus = cpu_client

    end
  end

  (1..3).each do |i|
```

```
config.vm.define "Node-#{i}" do |node|

  node.vm.box = "centos/7"

  node.vm.network "private_network", ip: "192.168.100.2#{i}"

  node.vm.synced_folder ".", "/vagrant", disabled: true

  node.vm.provider "virtualbox" do |vb|

    # Customize the amount of memory on the VM:

    vb.name = "Name of your project - Node #{i}"

    vb.memory = 512

    vb.cpus = 1

  end

end

end

end

servers.each do |conf|

  config.vm.define conf[:hostname] do |node|

    node.vm.box = "centos/7"

    node.vm.hostname = conf[:hostname]

    node.vm.network "private_network", ip: conf[:ip]

    node.vm.provider "virtualbox" do |vb|

      vb.customize ['storagectl', :id, '--name', 'SATA
Controller', '--portcount', conf[:disks]+1]
```

```
        (1..conf[:disks]).each do |x|

            file_to_disk =
'./disk_'+conf[:hostname]+'_'+x.to_s()+'.vdi'

            unless File.exist?(file_to_disk)

                vb.customize ['createhd', '--filename',
file_to_disk, '--size', 20 * 1024]

                end

                vb.customize ['storageattach', :id, '--storagectl',
'SATA Controller', '--port', x, '--device', 0, '--type', 'hdd',
'--medium', file_to_disk]

                end

                vb.memory = conf[:memory]

            end

        end

    end

end
```

VI. Ressources supplémentaires

- [Site officiel de Vagrant](#)