

Infogérance et virtualisation

Commande de base vagrant

Hassen CHEFFI

Institut Supérieur des Études Technologiques de SFAX

November 23, 2024

Introduction à Vagrant

- ▶ Qu'est-ce que Vagrant ?
 - ▶ Outil de gestion d'environnements de machines virtuelles (VM) reproductibles.
 - ▶ Permet de créer et gérer des VMs dans un environnement de développement unifié.
 - ▶ Fonctionne avec des hyperviseurs comme VirtualBox, VMware, Proxmox, etc.
- ▶ Avantages de Vagrant
 - ▶ Simplifie la gestion des environnements de développement.
 - ▶ Assure une configuration cohérente et portable entre les machines.

Démarrer avec Vagrant

► Vagrantfile

- Décrit la configuration de la machine virtuelle : système d'exploitation, ressources, réseau, etc.

```
1 Vagrant.configure("2") do |config|  
2   config.vm.box = "ubuntu/xenial64"  
3 end
```

► Créer un projet Vagrant

- Commande pour initialiser un projet :

```
1 vagrant init Nom_de_la_box
```

Commandes Vagrant usuelles

- ▶ **vagrant up** : Lance la ou les machines virtuelles définies dans le Vagrantfile. Si les machines n'existent pas encore, elles seront créées et démarrées.

```
1 vagrant up
```

- ▶ **vagrant halt** : Arrête la ou les machines virtuelles définies dans le Vagrantfile.

```
1 vagrant halt
```

- ▶ **vagrant destroy** : Supprime la ou les machines virtuelles, y compris toutes les modifications apportées.

```
vagrant destroy
```

Commandes Vagrant usuelles

- ▶ **vagrant reload** : Redémarre la ou les machines virtuelles définies dans le Vagrantfile.

```
vagrant reload
```

- ▶ **vagrant ssh** : Permet de se connecter à une machine virtuelle via SSH.

```
vagrant ssh
```

- ▶ Lister les boxes disponibles sur Vagrant

```
1 vagrant box list
```

Gestion des snapshots dans Vagrant

- ▶ Prendre un snapshot

```
vagrant snapshot save NOM_DU_SNAPSHOT
```

- ▶ Lister les snapshots

```
1 vagrant snapshot list
```

- ▶ Restaurer un snapshot

```
vagrant snapshot restore NOM_DU_SNAPSHOT
```

- ▶ Supprimer un snapshot

```
1 vagrant snapshot delete NOM_DU_SNAPSHOT
```

Les directives les plus utilisées dans le Vagrantfile

- ▶ `config.vm.box` : Définit la box utilisée pour la machine virtuelle.
- ▶ `config.vm.network` : Configure les paramètres réseau de la VM

```
1 config.vm.network "private_network",  
  type: "dhcp"  
2 config.vm.network "forwarded_port",  
  guest: 80, host: 8080
```

- ▶ `config.vm.provider` : Permet de configurer le fournisseur (hyperviseur) pour la VM.

```
1 config.vm.provider "virtualbox" do |vb|  
2   vb.memory = "1024"  
3   vb.cpus = 2  
4 end
```

Les directives les plus utilisées dans le Vagrantfile

- ▶ `config.vm.provision` : Permet de provisionner la machine virtuelle avec des scripts ou des configurations automatiques (ex. : shell, Ansible, Puppet).

```
1 config.vm.provision "shell", inline: <<-SHELL
2   apt-get update
3   apt-get install -y apache2
4 SHELL
```

- ▶ Le code de provision s'exécute uniquement lors du premier démarrage de la machine virtuelle. Pour forcer l'exécution du provisionnement à nouveau, il faut utiliser la commande suivante :

```
1 vagrant provision
```