

Atelier 6 : Automatisation avec Vagrant

Objectifs de l'atelier :

- Installer Vagrant et VirtualBox pour créer et gérer des machines virtuelles.
- Créer une machine virtuelle Ubuntu avec un Vagrantfile simple.
- Se connecter à la machine virtuelle via SSH.
- Utiliser un script de provisioning pour installer automatiquement Nginx au démarrage de la VM.

Exercice 1 : Installer Vagrant et VirtualBox

Objectif de l'exercice :

Installer Vagrant et VirtualBox sur votre machine afin de préparer l'environnement nécessaire pour les exercices suivants.

Étapes :

1. Installer VirtualBox :

- Téléchargez VirtualBox depuis <https://www.virtualbox.org/>.
- Suivez les instructions d'installation spécifiques à votre système d'exploitation (Windows, Mac, Linux).
- Vérifiez l'installation de VirtualBox avec la commande suivante :

```
virtualbox --help
```

- Si vous obtenez des informations sur l'outil, cela signifie que l'installation s'est bien déroulée.

2. Installer Vagrant :

- Téléchargez la dernière version de Vagrant depuis <https://www.vagrantup.com/downloads>.
- Suivez les instructions d'installation pour votre système d'exploitation.
- Vérifiez l'installation de Vagrant avec la commande :

```
vagrant --version
```

- Cette commande devrait retourner la version de Vagrant installée.

3. Vérification de l'installation :

- Une fois les deux outils installés, assurez-vous que vous avez accès à **VirtualBox** et **Vagrant** depuis votre terminal (commande `vagrant` et `virtualbox`).

Exercice 2 : Lancer une première VM Ubuntu avec un Vagrantfile simple

Objectif de l'exercice :

Apprendre à lancer une machine virtuelle Ubuntu à partir d'un fichier `Vagrantfile` simple en utilisant Vagrant et VirtualBox.

Étapes :

1. Créer un nouveau dossier pour votre projet :

- Dans votre terminal, créez un répertoire pour votre projet et placez-vous dedans :

```
mkdir vagrant_project  
cd vagrant_project
```

2. Initialiser un projet Vagrant :

- Initialisez un projet Vagrant en utilisant la commande suivante :

```
vagrant init ubuntu/bionic64
```

- Cette commande crée un fichier `Vagrantfile` dans votre dossier avec une configuration par défaut pour une machine virtuelle Ubuntu.

3. Vérification et modification du Vagrantfile :

- Ouvrez le fichier `Vagrantfile` généré et vérifiez la configuration par défaut. Il devrait ressembler à cela :

```
Vagrant.configure("2") do |config|  
  config.vm.box = "ubuntu/bionic64"  
end
```

- Ce fichier indique à Vagrant de créer une machine virtuelle basée sur la box `ubuntu/bionic64`.

4. Lancer la machine virtuelle :

- Démarrez la machine virtuelle avec la commande suivante :

```
vagrant up
```

- Cette commande va télécharger l'image de la box `ubuntu/bionic64` (si elle n'est pas déjà téléchargée) et démarrer la machine virtuelle via VirtualBox.

5. Vérification de l'état de la machine virtuelle :

- Une fois le processus terminé, vous pouvez vérifier l'état de la machine virtuelle avec :

```
vagrant status
```

- Cela devrait afficher l'état de la machine virtuelle, indiquant qu'elle est en cours d'exécution.

Exercice 3 : Se connecter en SSH à la machine virtuelle

Objectif de l'exercice :

Se connecter à la machine virtuelle via SSH pour interagir directement avec elle.

Étapes :

1. Accéder à la machine virtuelle :

- Utilisez la commande suivante pour vous connecter à la machine virtuelle en utilisant SSH :

```
vagrant ssh
```

- Une fois connecté, vous devriez être dans un shell sur la machine virtuelle Ubuntu.

2. Vérification de l'accès SSH :

- Vous pouvez vérifier que vous êtes bien connecté à la VM en exécutant une commande comme :

```
hostname
```

- Cela devrait afficher le nom de la machine virtuelle, par défaut "vagrant".

3. Quitter la machine virtuelle :

- Pour quitter la session SSH, tapez `exit` dans le terminal.
- Vous serez de retour dans le terminal de votre machine hôte.

Exercice 4 : Utiliser un script shell de provisioning pour installer Nginx automatiquement

Objectif de l'exercice :

Utiliser un script de provisioning pour automatiser l'installation de Nginx sur la machine virtuelle au démarrage de celle-ci.

Étapes :

1. Modifier le Vagrantfile pour ajouter le provisioning :

- Ouvrez le `Vagrantfile` dans un éditeur de texte.
- Ajoutez un bloc de provisioning pour installer Nginx automatiquement lorsqu'on démarre la machine virtuelle. Le fichier devrait ressembler à ceci :

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/bionic64"
  config.vm.provision "shell", inline: <<-SHELL
    sudo apt-get update
    sudo apt-get install -y nginx
```

```
SHELL  
end
```

2. Lancer la machine virtuelle avec provisioning :

- Après avoir sauvegardé le `Vagrantfile`, exécutez la commande suivante pour démarrer la machine virtuelle et appliquer le provisioning :

```
vagrant up
```

- Cette commande va démarrer la machine virtuelle et exécuter le script shell, qui mettra à jour les paquets et installera Nginx.

3. Vérification de l'installation de Nginx :

- Une fois la machine virtuelle en cours d'exécution, vous pouvez vérifier l'installation de Nginx en vous connectant à la machine virtuelle via SSH :

```
vagrant ssh
```

- Vérifiez que Nginx fonctionne en accédant à l'adresse `http://localhost` dans un navigateur ou via `curl` dans la machine virtuelle :

```
curl http://localhost
```

- Vous devriez voir la page par défaut de Nginx.

4. Recharger la machine virtuelle :

- Si vous avez apporté des modifications au `Vagrantfile`, vous pouvez les appliquer avec :

```
vagrant reload
```

Conclusion de l'atelier :

À l'issue de cet atelier, les participants auront acquis les compétences nécessaires pour :

- Installer et configurer Vagrant et VirtualBox.
- Créer et configurer une machine virtuelle Ubuntu via un `Vagrantfile` simple.
- Se connecter à une machine virtuelle Vagrant en SSH.
- Automatiser le provisioning de la machine virtuelle pour installer des logiciels comme Nginx.

Ce processus d'automatisation est essentiel dans le cadre du **déploiement de machines virtuelles**, offrant ainsi aux développeurs un environnement de travail reproductible et cohérent. Les concepts et compétences abordés dans cet atelier sont directement applicables à des scénarios d'Infrastructure as Code (IaC).