**Module : Cloud computing**

**TP Vagrant**

**I) Création d’un réseau local :**

La création d'un réseau local est le processus d'attribution d'une adresse IP à une machine Vagrant.

Vagrant offre deux options :

* Un réseau local qui limite l'accès aux machines virtuelles (VM) s'exécutant sur l'ordinateur hôte. Le logiciel de l'hyperviseur spécifie généralement une plage d'adresses.
* Un réseau ponté (bridged) qui obtiendra une adresse IP en dehors de la plage locale. Cela signifie que la machine Vagrant est accessible comme n'importe quelle autre machine sur le réseau informatique hôte. Vous pouvez, par exemple, spécifier un réseau ponté si vous voulez que votre machine Vagrant soit une ressource partagée entre de nombreuses personnes différentes de votre Bureau.

Dans cette manipulation, nous allons créer une machine Vagrant simple exécutant Ubuntu 14.04 LTS et attribuer une adresse IP à la machine.

1. Commencez avec un simple Vagrantfile. Dans ce cas, nous allons commencer par une définition de base de la machine:

|  |
| --- |
| # -\*- mode: ruby -\*- # vi: set ft=ruby : VAGRANTFILE\_API\_VERSION = "2" Vagrant.configure(VAGRANTFILE\_API\_VERSION) do |config| config.vm.box = "ubuntu/trusty64" end |

1. Pour cette configuration, attribuez une adresse IP à la machine Vagrant à l'aide du

|  |
| --- |
| config.vm.network "private\_network", ip: "192.168.99.100" |

1. Démarrez la machine : vagrant up
2. A l'invite de commande de la machine Vagrant, vérifiez les plages IP de la machine à l'aide de la commande ifconfig. Cela affichera des informations sur l'environnement réseau de la machine.

Notez que la machine Vagrant a deux adresses IP distinctes sur différentes interfaces définies  
ici comme eth0 et eth1. La machine peut répondre à l'une ou l'autre des adresses IP (une  
par l'hyperviseur, tandis que l'autre est défini dans la configuration Vagrant).

**II) Environnement multi machines :**

La principale raison pour laquelle nous souhaitons créer des réseaux de machines Vagrant est souvent parce que nous souhaitons modéliser un environnement de plusieurs machines. Un exemple courant pourrait être le désir de modéliser une application Web avec des serveurs Web et une base de données dédiés machines, ou même un environnement qui crée un cluster de machines virtuelles identiques.

1. Créer un nouveau répertoire.
2. Commencez avec un simple Vagrantfile sans définition de machine ou de boîte:

|  |
| --- |
| # -\*- mode: ruby -\*- # vi: set ft=ruby : VAGRANTFILE\_API\_VERSION = "2" Vagrant.configure(VAGRANTFILE\_API\_VERSION) do |config| end |

1. Dans le | config | , définissez un serveur de base de données à l'aide d'une définition de machine Vagrant. Dans un environnement multimachine, nous voulons généralement  
   pour définir une adresse IP de machine (en particulier dans le cas où une machine Vagrant (un serveur web) devra se connecter à une autre (un serveur de base de données).

|  |
| --- |
| config.vm.define "database" do |db| db.vm.box = " ubuntu/trusty64" db.vm.hostname = "db01" db.vm.network "private\_network", ip: "192.168.55.100" end |

1. Créez une deuxième machine définie dans un bloc après le bloc de| db |. Ce sera la machine Web:

|  |
| --- |
| config.vm.define "web" do |web| web.vm.box = " ubuntu/trusty64" web.vm.hostname = "web01" web.vm.network "private\_network", ip:"192.168.55.101" end |

1. Démarrez les machines
2. Pinger sur les deux machines.
3. Taper la commande vagrant status.
4. Connecter-vous sur web et pinger sur la machine db.
5. Nous utiliserons le provisioner pour définir le fichier /etc /hosts sur le serveur Web. Cela permet au serveur Web de faire référence au serveur de base de données avec le nom d'hôte db01. La définition complète du serveur Web comprendra cette commande d'approvisionnement.

|  |
| --- |
| config.vm.define "web" do |web| web.vm.box = " ubuntu/trusty64" web.vm.hostname = "web01" web.vm.network "private\_network", ip:"192.168.55.101" web.vm.provision "shell", inline: "echo '127.0.0.1 localhost web01\n192.168.55.100 db01' > /etc/hosts" end |

1. Démarrez les machines
2. Accédez à la machine Web en utilisant la commande vagrant ssh.
3. Vérifier que le serveur web peut contacter la machine database avec son nom db01 : ping db01.

**III ) Exécuter un script dans un vagrantfile :**

Le provisionnement d'une machine Vagrant avec des arguments de chaîne en ***inline*** peut faciliter les tâches de provisionnement simple, mais des exigences plus complexes peuvent nécessiter des scripts plus compliqués.

Dans cette manipulation, nous allons fournir un nouvel environnement Vagrant, installer le serveur Web nginx et remplacer le répertoire Web par défaut par notre répertoire de travail. Le répertoire de travail qui contient le Vagrantfile est partagé, par défaut, avec le système d'exploitation invité comme dossier /vagrant.

A l’aide de vagrant, il est possible de créer un environnement virtuel composé de plusieurs machines. Dans cette manipulation, on va créer deux VM :

1. Ubuntu
2. Centos

Créer un nouveau répertoire contenant le fichier vagrantfile suivant :

|  |
| --- |
| # -\*- mode: ruby -\*-  # vi: set ft=ruby :  Vagrant.configure("2") do |config|  config.vm.define "ubuntu-vm" do |vm1|  vm1.vm.hostname ="ubuntu-vm"  vm1.vm.box = "bento/ubuntu-16.04"  vm1.vm.network "private\_network", ip: "192.168.33.10"  vm1.vm.provider "virtualbox" do |vb|  vb.name ="ubuntu-vm"  vb.gui=false  vb.memory = 1024  end  vm1.vm.provision "shell", run: "always", inline: <<-SHELL  echo "Hello from the Ubuntu VM"  #apt-get update  #apt-get install -y apache2  SHELL  end  config.vm.define "centos-vm" do |vm2|  vm2.vm.hostname ="centos-vm"  vm2.vm.box = "bento/centos-7.2"  vm2.vm.network "private\_network", ip: "192.168.33.20"  vm2.vm.provider "virtualbox" do |vb|  vb.name ="centos-vm"  vb.gui=false  vb.memory = 1024  end  vm2.vm.provision "shell", run: "always", inline: <<-SHELL  echo "Hello from the Centos VM"  SHELL  end  end |

1. Pinger sur les deux machines.
2. Taper la commande vagrant status.
3. Connecter-vous sur ubuntu et pinger sur la machine centos.