

Practica Aprovisionamiento

Oscar H. Mondragón

1. Objetivos

- Comprender el funcionamiento de los Aprovisionadores SHELL y Puppet con Vagrant.

2. Herramientas a utilizar

- Vagrant
- VirtualBox
- Provisioners: Shell, Puppet

3. Aprovisionamiento con Vagrant Shell

3.1. Configuración de Vagrant

Esta práctica la desarrollaremos usando boxes de Ubuntu 22.04 en Vagrant. El Vagrantfile que usaremos es el siguiente (con el que venimos trabajando):

```
# -*- mode: ruby -*-  
# vi: set ft=ruby :
```

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

```
  if Vagrant.has_plugin? "vagrant-vbguest"  
    config.vbguest.no_install = true  
    config.vbguest.auto_update = false  
    config.vbguest.no_remote = true  
  end
```

```
  config.vm.define :clienteUbuntu do |clienteUbuntu|  
    clienteUbuntu.vm.box = "bento/ubuntu-22.04"  
    clienteUbuntu.vm.network :private_network, ip: "192.168.100.2"  
    clienteUbuntu.vm.hostname = "clienteUbuntu"  
  end
```

```
  config.vm.define :servidorUbuntu do |servidorUbuntu|  
    servidorUbuntu.vm.box = "bento/ubuntu-22.04"  
    servidorUbuntu.vm.network :private_network, ip: "192.168.100.3"
```

```
servidorUbuntu.vm.hostname = "servidorUbuntu"  
servidorUbuntu.vm.provision "shell", path: "script.sh"  
end  
end
```

3.2. Pruebas básicas de aprovisionamiento

Note que el servidor esta siendo aprovisionado usando a línea:

```
servidorUbuntu.vm.provision "shell", path: "script.sh"
```

Cree un archivo script.sh con el siguiente contenido:

```
#!/bin/bash  
  
echo "configurando el resolv.conf con cat"  
cat <<TEST> /etc/resolv.conf  
nameserver 8.8.8.8  
TEST  
  
echo "instalando un servidor vsftpd"  
sudo apt-get install vsftpd -y  
  
echo "Modificando vsftpd.conf con sed"  
sed -i 's/#write_enable=YES/write_enable=YES/g' /etc/vsftpd.conf  
  
echo "configurando ip forwarding con echo"  
sudo echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
```

Verifique el aprovisionamiento y explique su funcionamiento.

4. Aprovisionamiento con Puppet

Los archivos usados en esta parte de la practica se encuentran en el repositorio <https://github.com/omondragon/testPuppet>

4.1. Configuración de Vagrant

Con el fin de usar puppet, cree el Vagrantfile como sigue (también puede solo agregar la parte de aprovisionamiento a su Vagrantfile existente):

```
# -*- mode: ruby -*-
```

```
# vi: set ft=ruby :
```

```
$install_puppet = <<-PUPPET
sudo apt-get update -y
sudo apt-get install -y puppet
PUPPET
```

```
Vagrant.configure("2") do |config|
  config.vm.box = "bento/ubuntu-20.04"
  config.vm.hostname = "puppetServer"
  config.vm.network :private_network, ip: "192.168.90.3"
  config.vm.provision "shell", inline: $install_puppet
  config.vm.provision :puppet do |puppet|
    puppet.manifests_path = "puppet/manifests"
    puppet.manifest_file = "site.pp"
    puppet.module_path = "puppet/modules"
  end
end
```

Construya la siguiente estructura de directorios:

```
omondragon@Oscars-MacBook-Pro test_puppet % tree
.
├── Vagrantfile
├── puppet
│   ├── manifests
│   │   └── site.pp
│   └── modules
│       └── baseconfig
│           ├── files
│           │   └── index.html
│           └── manifests
│               └── init.pp
```

Cree los archivos con el siguiente contenido:

site.pp

```
include baseconfig
```

index.html

```
<!DOCTYPE html>
<html>
<body>

<h1>Aprovisionando con Puppet</h1>

<p>Probando puppet</p>

</body>
</html>
```

init.pp

```

class baseconfig {
  exec { 'apt-get update':
    command => '/usr/bin/apt-get update';
  }

  package { ['apache2', 'tree']:
    ensure => present;
  }

  file { ['/var/www/html/index.html']:
    ensure => present,
    owner   => 'root',
    group   => 'root',
    mode    => '0644',
    source  => 'puppet:///modules/baseconfig/index.html',
    path    => '/var/www/html/index.html';
  }

  service { "apache2":
    ensure => running,
    enable => true,
    require => Package['apache2'];
  }
}

```

"modules/baseconfig/manifests/init.pp" 24L, 496C

5. Ejercicio

1. Implemente el aprovisionamiento basado en Shell propuesto en la sección 3. Explique su funcionamiento.
2. Implemente el aprovisionamiento basado en Puppet propuesto en la sección 4. Explique su funcionamiento.
3. Aprovisiona Jupiter notebooks usando el aprovisionador que mas le guste.

6. Desafío [Hasta 0.5 Puntos en una nota de practicas]

Implemente uno

4. Aprovisione una maquina Vagrant con un contenedor LXD corriendo un sitio web personalizado
5. Implemente aprovisionamiento de un servicio/aplicación usando Chef Provisioner.

6. Bibliografía

- Sitio oficial Ansible: <https://www.ansible.com/>
- Vagrant Provisioning with Puppet.
<https://medium.com/@Joachim8675309/vagrant-provisioning-with-puppet-553a59f0c48e>
- Automation testing with ansible, molecule and Vagrant.
<https://www.trustedsec.com/blog/automation-testing-with-ansible-molecule-and-vagrant/>
- Correr Ansible en Windows.
https://docs.ansible.com/ansible/2.5/user_guide/windows_faq.html
- Instalar WSL en Windows. <https://docs.microsoft.com/en-us/windows/wsl/install-win10>
- Aprovisionamiento en Vagrant:
<https://www.vagrantup.com/docs/provisioning>
- Vagrant Provisioning with Puppet.
<https://medium.com/@Joachim8675309/vagrant-provisioning-with-puppet-553a59f0c48e>
- G.ST.02 Guía de Computación en la nube (2018):
https://estrategia.gobiernoenlinea.gov.co/623/articles-75246_recurso_2.pdf