Programación de Aplicaciones para Internet y la Nube **2016-I**

TALLER 4

Nombre del Estudiante: **Duván Pardo, Wilson López**

1. INTRODUCCIÓN

Vagrant es un software informático que crea y configura entornos de desarrollo virtuales. Puede verse como un contenedor de nivel superior de todo el software de virtualización como VirtualBox, VMware, KVM y Contenedores de Linux (LXC), y alrededor de software de gestión de configuración, tales como Ansible, Chef, Salty y Pupplet.(tomado de Introducing Vagrant)

Vagrant fue originalmente ligado a VirtualBox, pero la versión 1.1 agregó soporte a otro software de virtualización como VMware y KVM, y para entornos de servidores como Amazon EC2.(tomado de Vagrant: Up and Running, Mitchell Hashimoto (2013) y O'Reilly Media)

Vagrant está escrito en Ruby, pero se puede utilizar en proyectos escritos en otros lenguajes de programación tales como PHP, Python, Java, C # y JavaScript. (tomado dw Vagrant: EC2-Like Virtual Machine Building and Provisioning from Ruby y Vagrant - Getting Started - Project Setup)

Desde la versión 1.6, Vagrant soporta de forma nativa contenedores estibador, que en algunos casos puede servir como un sustituto de un sistema operativo completamente virtualizado. Vagrant 1.6, Mitchell Hashimoto (2014-05-06).

Vagrant se inició en enero de 2010 por Mitchell Hashimoto. Durante casi tres años, vagrant era un proyecto paralelo para Mitchell, un proyecto que trabajó en en sus horas libres después de su trabajo a tiempo completo. Durante este tiempo, vagrant llegó a ser de confianza y utilizado por una amplia gama de individuos en los equipos de desarrollo en las grandes empresas.

En noviembre de 2012, se formó HashiCorp por Mitchell para respaldar el desarrollo de Vagrant a tiempo completo. HashiCorp construye adiciones comerciales y proporciona apoyo profesional y entrenamiento para Vagrant.

Vagrant sigue siendo y siempre será un proyecto con licencia de código abierto. Cada versión de Vagrant es el trabajo de cientos de contribuciones individuales al proyecto de código abierto.(tomado de Vagrant-Hashicorp)

2. OBJETIVO

Realizar el despliegue de entornos de desarrollo sencillos mediante la tecnología Vagrant utilizando como proveedor de infraestructura la tecnología VirtualBox.

3. ACTIVIDADES

1. Abrir una consola de comandos.
2. Validar la correcta instalación del software Vagrant ejecutando el comando `vagrant -h`

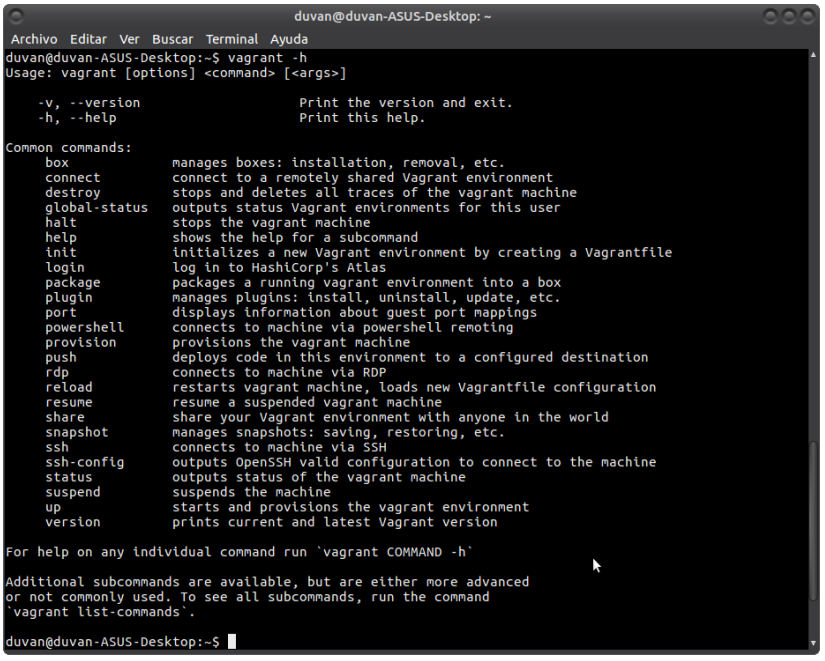

Figura 1: Verificación por consola de la instalación de Vagrant

3. Adicionar la imagen del sistema operativo Ubuntu Trusty de 64 bits. Para ello, ejecutar el comando `vagrant box add ubuntu/trusty64` . La descarga debe tomar alrededor de 5 minutos.

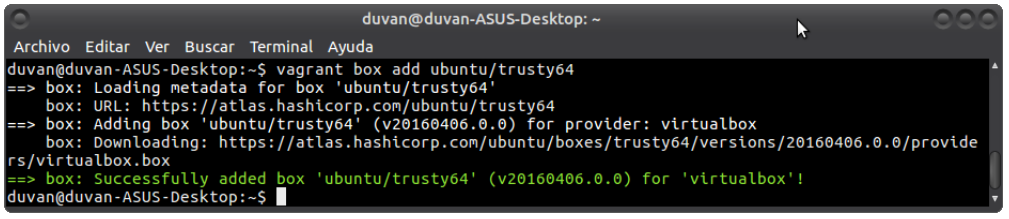

Figura 2: Adición del sistema operativo Ubuntu Trusty de 64 bits

4. Crear un directorio de trabajo clouapps (o cualquier otro nombre). Ingresar a ese directorio en la consola y ejecutar el comando `vagrant init ubuntu/trusty64`. Después de ejecutar el comando, verificar que se haya creado un archivo denominado `Vagrantfile`.

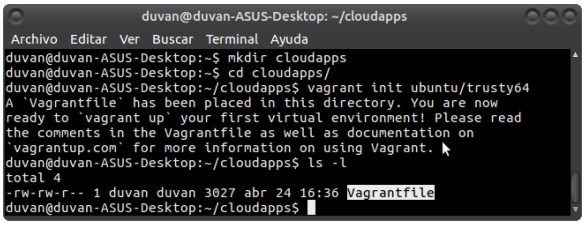

Figura 3: Creación del directorio clouapps

5. Editar el archivo `Vagrantfile` de forma que contenga la siguiente información

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

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure(2) do |config|

config.vm.box = "ubuntu/trusty64"
config.vm.network :forwarded_port, guest: 4000, host: 8100, host_ip: "127.0.0.1"
config.vm.provision "puppet"
config.vm.hostname = "www.cecad-example.com"

end
```

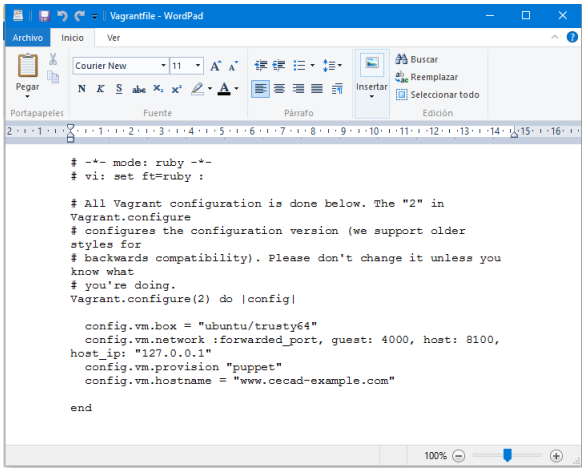

Figura 4: Archivo `Vagrantfile` editado.

6. Crear un archivo "`default.pp`" en un directorio `manifests` que se encuentra en el mismo lugar del archivo `Vagrantfile`.

7. Escribir el siguiente contenido en el archivo "`default.pp`"

```
exec { 'apt-get update':
command => '/usr/bin/apt-get update -y'
}

package { 'nodejs':
require => Exec['apt-get update']
}

package { 'lynx-cur':
require => Exec['apt-get update']
}

package { 'ruby1.9.1-dev':
require => Exec['apt-get update']
}
```

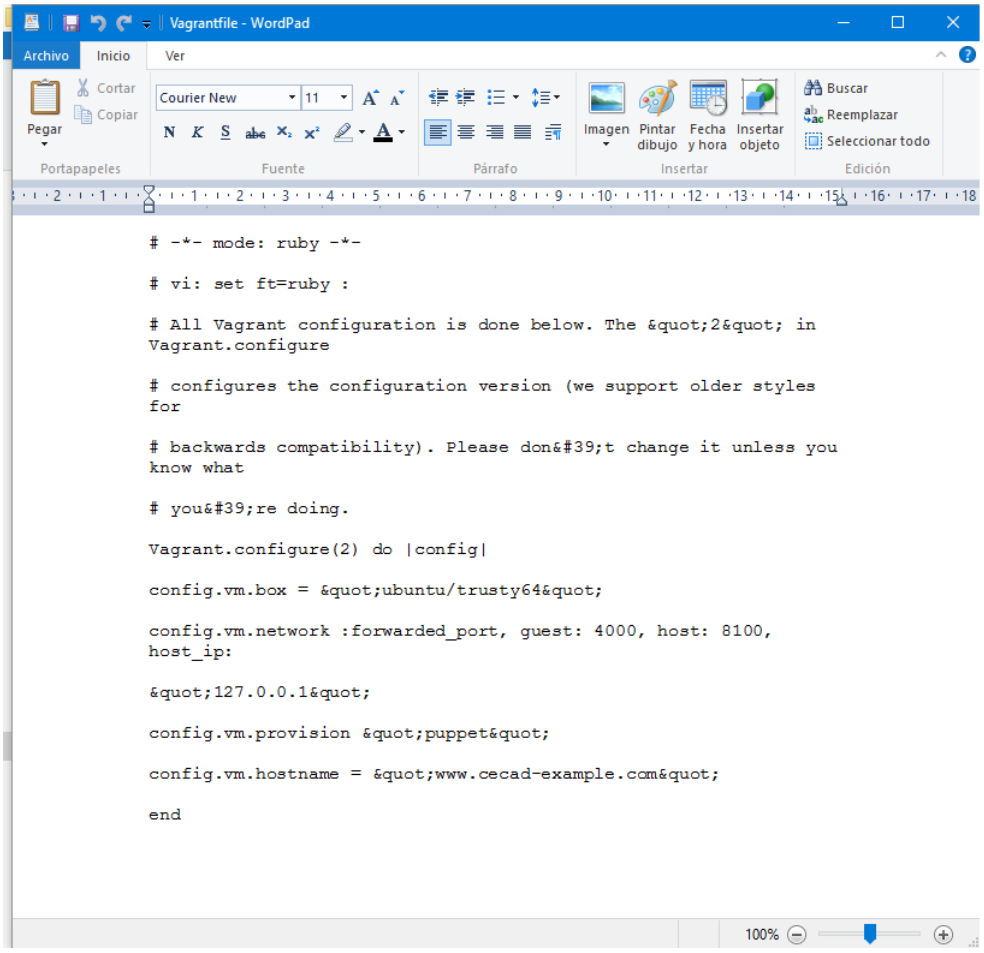

Figura 5: Archivo `script.sh` editado.

8. Ejecutar el comando `vagrant up --provision`.

   con lo que obtenemos la siguiente respuesta:

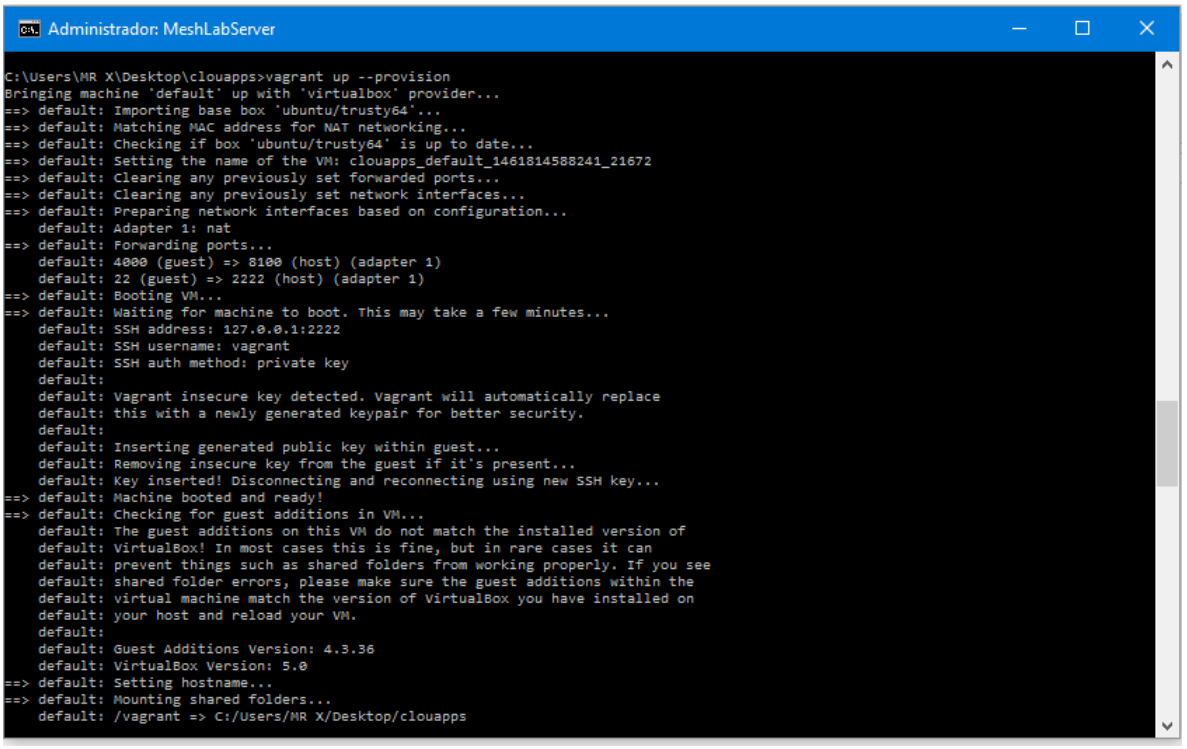

Figura 6: Ejecución del comando `vagrant up --provision`.

A continuación se nuestra con mas detalle el resultado al ejecutar el comando `vagrant up --provision`:

```
C:\Users\MR X\Desktop\clouapps>vagrant up --provision
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/trusty64' is up to date...
==> default: Setting the name of the VM: clouapps_default_1461814588241_21672
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
==> default: Forwarding ports...
default: 4000 (guest) => 8100 (host) (adapter 1)
default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
default: SSH address: 127.0.0.1:2222
default: SSH username: vagrant
default: SSH auth method: private key
default:
default: Vagrant insecure key detected. Vagrant will automatically replace
default: this with a newly generated keypair for better security.
default:
default: Inserting generated public key within guest...
default: Removing insecure key from the guest if it's present...
default: Key inserted! Disconnecting and reconnecting using new SSH key...
```

```
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
default: The guest additions on this VM do not match the installed version of
default: VirtualBox! In most cases this is fine, but in rare cases it can
default: prevent things such as shared folders from working properly. If you see
default: shared folder errors, please make sure the guest additions within the
default: virtual machine match the version of VirtualBox you have installed on
default: your host and reload your VM.
default:
default: Guest Additions Version: 4.3.36
default: VirtualBox Version: 5.0
==> default: Setting hostname...
==> default: Mounting shared folders...
default: /vagrant => C:/Users/MR X/Desktop/clouapps
default: /tmp/vagrant-puppet/manifests-a11d1078b1b1f2e3bdea27312f6ba513 =>
C:/Users/MR X/Desktop/clouapps/manifests
==> default: Running provisioner: puppet...
==> default: Running Puppet with default.pp...
==> default: stdin: is not a tty
==> default: Notice: Compiled catalog for www.cecad-example.com in environment
 production in 0.25 seconds
==> default: Notice: /Stage[main]/Main/Exec[apt-get update]/returns: executed
 successfully
==> default: Notice: /Stage[main]/Main/Package[lynx-cur]/ensure: ensure changed
 'purged' to 'present'
==> default: Notice: /Stage[main]/Main/Package[nodejs]/ensure: ensure changed
 'purged' to 'present'
==> default: Notice: /Stage[main]/Main/Package[ruby1.9.1-dev]/ensure: ensure
changed 'purged' to 'present'
==> default: Notice: Finished catalog run in 53.66 seconds
```

4. BIBLIOGRAFÍA

- http://www.linuxjournal.com/content/introducing-vagrant
- http://cdn.oreillystatic.com/oreilly/booksamplers/9781449335830_sampler.pdf
- https://en.wikipedia.org/wiki/O
- http://www.rubyinside.com/vagrant-ruby-powered-virtualbox-vm-building-and-provisioning-3059.html
- https://www.vagrantup.com/docs/getting-started/project_setup.html
- https://www.hashicorp.com/blog/vagrant-1-6.html
- https://www.vagrantup.com/about.html