

## 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 Puppet.(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 de 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.

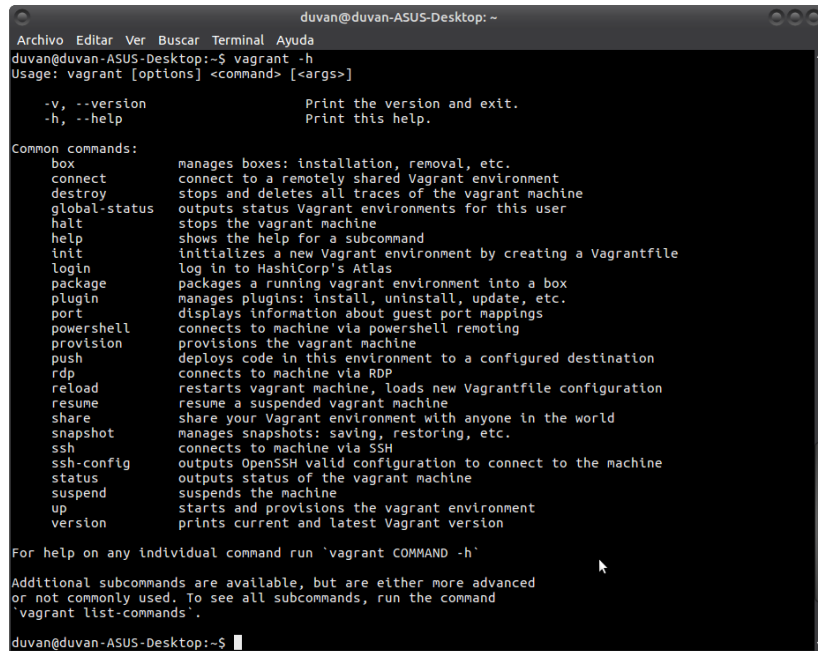
Vagrant sigue siendo y siempre será un proyecto con licencia de código abierto. Cada versión de Vagrant es el trabajo de cientos de contribuciones individuales al proyecto de código abierto.(tomado de Vagrant-Hashicorp)

### 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`



```
duvan@duvan-ASUS-Desktop: ~
Archivo Editar Ver Buscar Terminal Ayuda
duvan@duvan-ASUS-Desktop:~$ vagrant -h
Usage: vagrant [options] <command> [<args>]

-v, --version          Print the version and exit.
-h, --help             Print this help.

Common commands:
box                   manages boxes: installation, removal, etc.
connect              connect to a remotely shared Vagrant environment
destroy              stops and deletes all traces of the vagrant machine
global-status        outputs status Vagrant environments for this user
halt                 stops the vagrant machine
help                 shows the help for a subcommand
init                 initializes a new Vagrant environment by creating a Vagrantfile
login                log in to HashiCorp's Atlas
package              packages a running vagrant environment into a box
plugin               manages plugins: install, uninstall, update, etc.
port                 displays information about guest port mappings
powershell           connects to machine via powershell remoting
provision             provisions the vagrant machine
push                 deploys code in this environment to a configured destination
rdp                  connects to machine via RDP
reload               restarts vagrant machine, loads new Vagrantfile configuration
resume               resume a suspended vagrant machine
share                share your Vagrant environment with anyone in the world
snapshot             manages snapshots: saving, restoring, etc.
ssh                  connects to machine via SSH
ssh-config            outputs OpenSSH valid configuration to connect to the machine
status               outputs status of the vagrant machine
suspend              suspends the machine
up                   starts and provisions the vagrant environment
version              prints current and latest Vagrant version

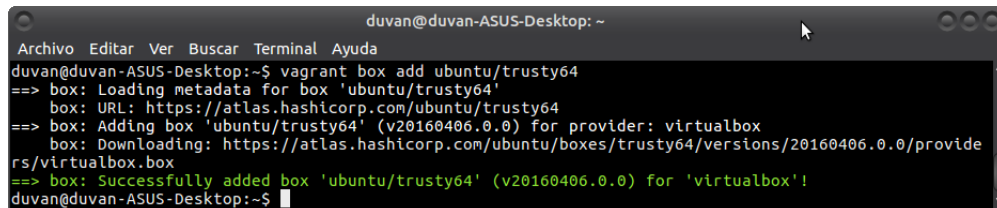
For help on any individual command run 'vagrant COMMAND -h'

Additional subcommands are available, but are either more advanced
or not commonly used. To see all subcommands, run the command
'vagrant list-commands'.

duvan@duvan-ASUS-Desktop:~$
```

Figura 1: Verificación por consola de la instalación de Vagrant

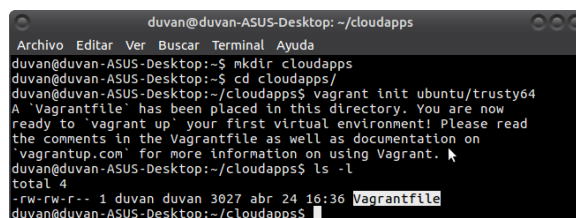
3. Adicionar la imagen del sistema operativo Ubuntu Trusty de 64 bits. Para ello, ejecutar el comando `vagrant box add ubuntu/trusty64`. La descarga debe tomar alrededor de 5 minutos.



```
duvan@duvan-ASUS-Desktop: ~
Archivo Editar Ver Buscar Terminal Ayuda
duvan@duvan-ASUS-Desktop:~$ vagrant box add ubuntu/trusty64
==> box: Loading metadata for box 'ubuntu/trusty64'
box: URL: https://atlas.hashicorp.com/ubuntu/trusty64
==> box: Adding box 'ubuntu/trusty64' (v20160406.0.0) for provider: virtualbox
box: Downloading: https://atlas.hashicorp.com/ubuntu/boxes/trusty64/versions/20160406.0.0/provide
rs/virtualbox.box
==> box: Successfully added box 'ubuntu/trusty64' (v20160406.0.0) for 'virtualbox'!
duvan@duvan-ASUS-Desktop:~$
```

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`.

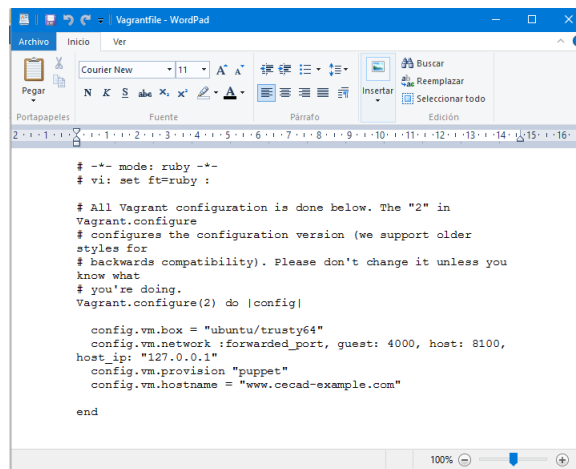


```
duvan@duvan-ASUS-Desktop: ~/cloudapps
Archivo Editar Ver Buscar Terminal Ayuda
duvan@duvan-ASUS-Desktop:~$ mkdir cloudapps
duvan@duvan-ASUS-Desktop:~$ cd cloudapps/
duvan@duvan-ASUS-Desktop:~/cloudapps$ vagrant init ubuntu/trusty64
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.
duvan@duvan-ASUS-Desktop:~/cloudapps$ ls -l
total 4
-rw-rw-r-- 1 duvan duvan 3027 abr 24 16:36 Vagrantfile
duvan@duvan-ASUS-Desktop:~/cloudapps$
```

Figura 3: Creación del directorio clouapps

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

```
1 # -*- mode: ruby -*-
2 # vi: set ft=ruby :
3
4 # All Vagrant configuration is done below. The "2" in Vagrant.configure
5 # configures the configuration version (we support older styles for
6 # backwards compatibility). Please don't change it unless you know what
7 # you're doing.
8 Vagrant.configure(2) do |config|
9
10   config.vm.box = "ubuntu/trusty64"
11   config.vm.network :forwarded_port, guest: 4000, host: 8100, host_ip: "127.0.0.1"
12   config.vm.provision "puppet"
13   config.vm.hostname = "www.cecad-example.com"
14
15 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”

```
1 exec { 'apt-get update':  
3   command => '/usr/bin/apt-get update -y'  
4 }  
5  
6 package { 'nodejs':  
7   require => Exec['apt-get update']  
8 }  
9  
10 package { 'lynx-cur':  
11   require => Exec['apt-get update']  
12 }  
13  
14 package { 'ruby1.9.1-dev':  
15   require => Exec['apt-get update']  
16 }
```

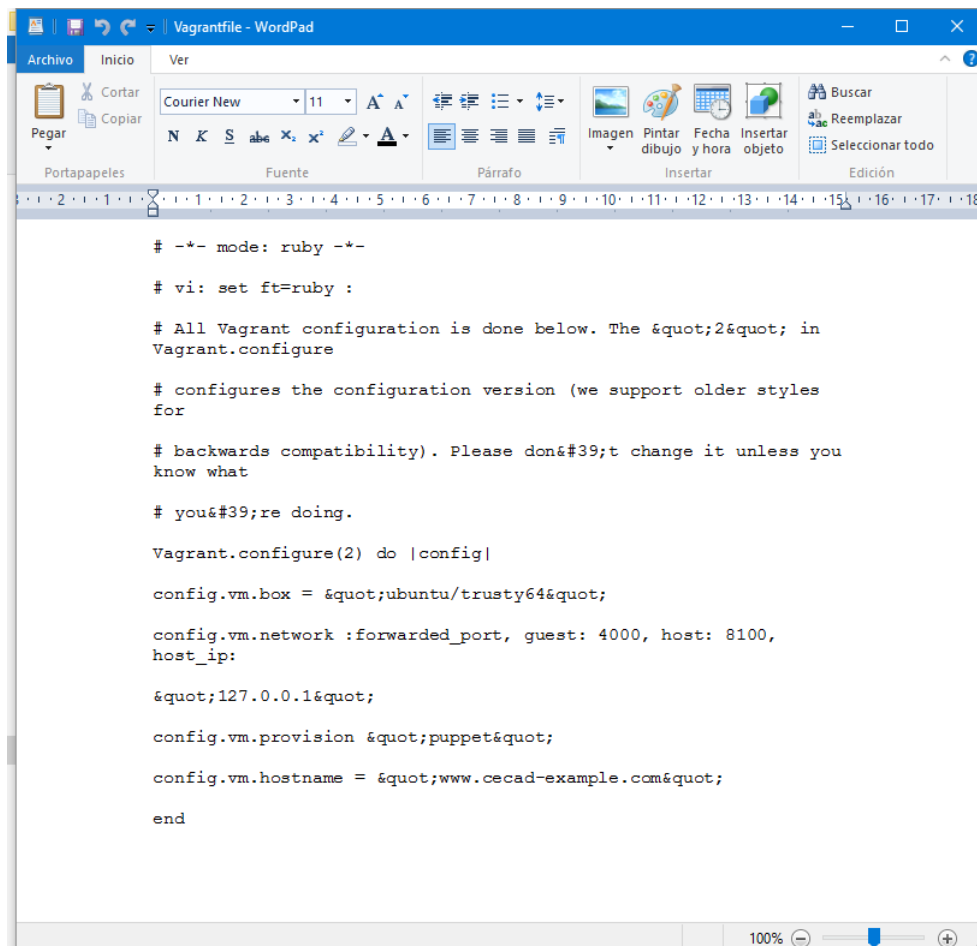


Figura 5: Archivo script.sh editado.

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

con lo que obtenemos la siguiente respuesta:

```
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
```

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

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

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

```
26 => default: Machine booted and ready!
=> default: Checking for guest additions in VM...
28 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
30 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
32 default: virtual machine match the version of VirtualBox you have installed on
default: your host and reload your VM.
34 default:
default: Guest Additions Version: 4.3.36
36 default: VirtualBox Version: 5.0
=> default: Setting hostname...
38 => default: Mounting shared folders...
default: /vagrant => C:/Users/MR X/Desktop/clouapps
40 default: /tmp/vagrant-puppet/manifests-a11d1078b1b1f2e3bdea27312f6ba513 =>
C:/Users/MR X/Desktop/clouapps/manifests
42 => default: Running provisioner: puppet...
=> default: Running Puppet with default.pp...
44 => default: stdin: is not a tty
=> default: Notice: Compiled catalog for www.cecad-example.com in environment
46 production in 0.25 seconds
=> default: Notice: /Stage[main]/Main/Exec[apt-get update]/returns: executed
48 successfully
=> default: Notice: /Stage[main]/Main/Package[lynx-cur]/ensure: ensure changed
'purged' to 'present'
50 => default: Notice: /Stage[main]/Main/Package[nodejs]/ensure: ensure changed
'purged' to 'present'
52 => default: Notice: /Stage[main]/Main/Package[ruby1.9.1-dev]/ensure: ensure
54 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](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.vagrantup.com/docs/getting-started/project_setup.html)
- <https://www.hashicorp.com/blog/vagrant-1-6.html>
- <https://www.vagrantup.com/about.html>