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

# TALLER 3

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

## 1. INTRODUCCIÓN

Vagrant es una herramienta para crear entornos de desarrollo completos. Con un flujo de trabajo fácil de usar y centrado en la automatización, reduce el tiempo de configuración del entorno de desarrollo, aumenta la paridad entre desarrollo-producción, y hace que las .ºbras en mi máquina"sean una reliquia del pasado.

**Otto es el sucesor de Vagrant.**

Los creadores de Otto son también los creadores de vagrant. Después de trabajar en Vagrant más de seis años, los autores considera que Otto es una herramienta superior para el desarrollo y mucho más. Vagrant todavía llenará un papel importante para algunos usuarios, pero para la mayoría de los desarrolladores, Otto reemplazará Vagrant con el tiempo.

Aunque creen que Otto es un sucesor de Vagrant, El desarrollo de Vagrant continuará en los próximos años. Otto se basa en Vagrant, por lo que las mejoras de Vagrant beneficiarán a los usuarios Otto también. tomado de Vagran-Hashicorp y Otto-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`

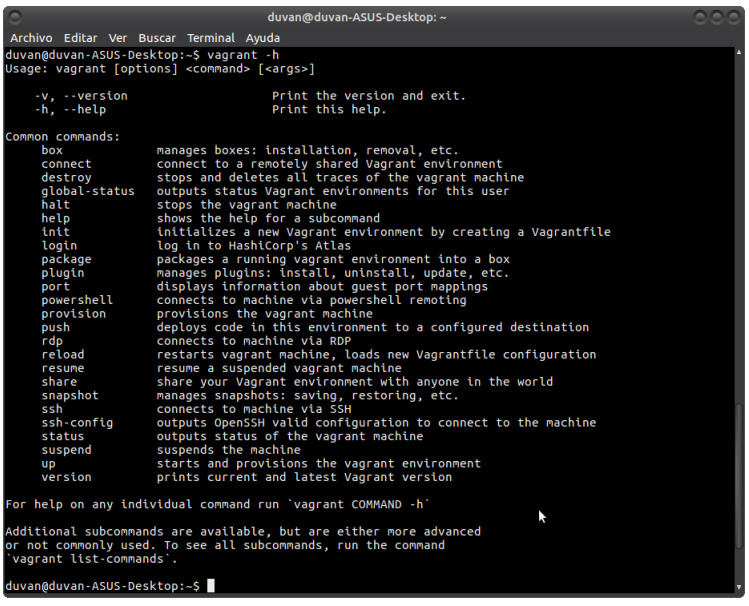


Figura 1: Verificación por consola de la instalación de Vagrant

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.

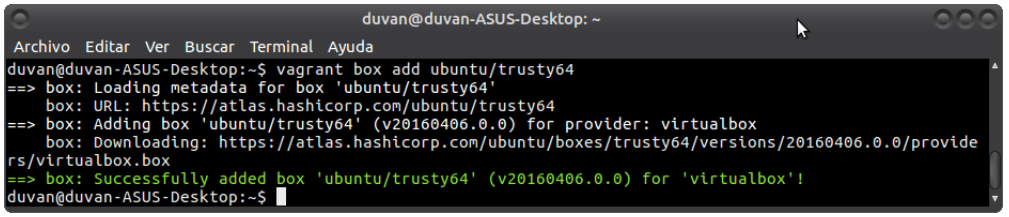


Figura 2: Adición del sistema operativo Ubuntu Trusty de 64 bits

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`.

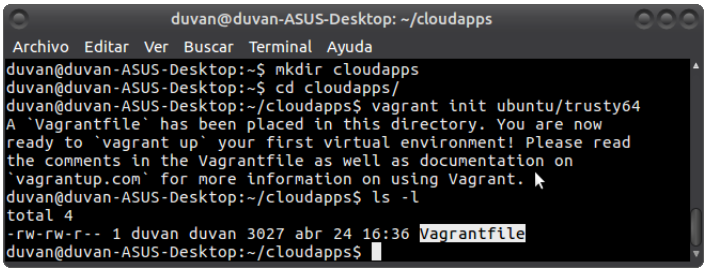


Figura 3: Creación del directorio clouapps

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

```
1
2
3 # -*- mode: ruby -*-
  # vi: set ft=ruby :
5 # All Vagrant configuration is done below. The "2" in Vagrant.configure
  # configures the configuration version (we support older styles for
7 # backwards compatibility). Please don't change it unless you know what
  # you're doing.
9 Vagrant.configure(2) do |config|
  config.vm.box = "ubuntu/trusty64"
11 config.vm.network :forwarded_port, guest: 4000, host: 8100, host_ip: "127.0.0.1"
  config.vm.provision "shell", path: "script.sh"
13 end
```

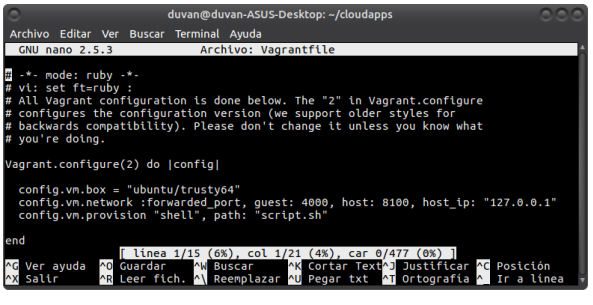


Figura 4: Archivo `Vagrantfile` editado.

6. Crear un archivo "`script.sh`" en el mismo directorio del archivo `Vagrantfile`.

7. Escribir el siguiente contenido en el archivo "`script.sh`"

```
#!/usr/bin/env bash
echo &quot;Installing: nodejs, lynx, ruby and jekyll...&quot;
apt-add- repository ppa:brightbox/ruby-ng &gt;&gt;/tmp/provision-script.log ... 2&gt;&amp;1
apt-get -y update &gt;&gt;/tmp/provision-script.log 2&gt;&amp;1
apt-get install -y nodejs &gt;&gt;/tmp/provision-script.log 2&gt;&amp;1
apt-get install -y lynx-cur &gt;&gt;/tmp/provision-script.log 2&gt;&amp;1
apt-get install -y ruby2.2 &gt;&gt;/tmp/provision-script.log 2&gt;&amp;1
apt-get install -y ruby2.2-dev &gt;&gt;/tmp/provision-script.log 2&gt;&amp;1
gem install jekyll &gt;&gt;/tmp/provision-script.log 2&gt;&amp;1
cd /vagrant
jekyll serve –H 0.0.0.0 – detach
```

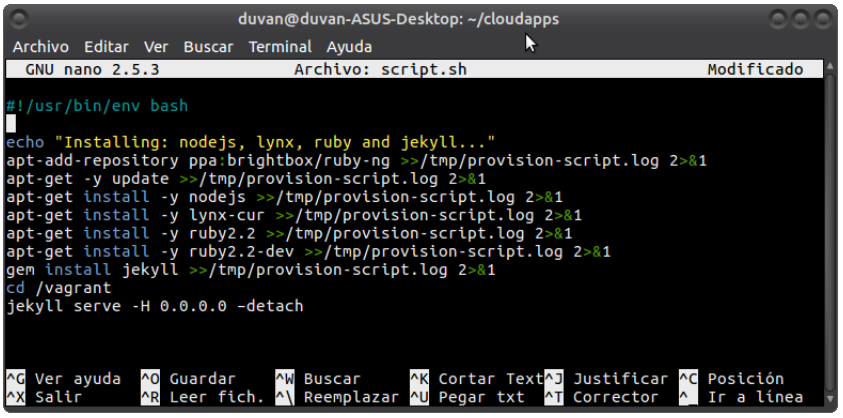


Figura 5: Archivo `script.sh` editado.

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

   Con lo que obtenemos la siguiente respuesta:

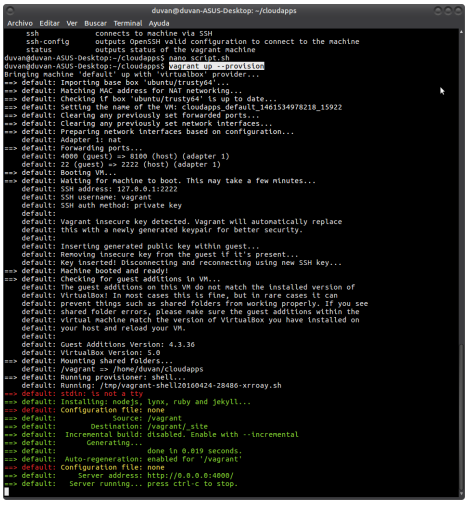

Figura 6: Ejecución del comando `vagrant up {provision`.

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

```
duvan@duvan-ASUS- Desktop:~/cloudapps$ vagrant up --provision
Bringing machine &#39;default&#39; up with &#39;virtualbox&#39; provider...
==&gt; default: Importing base box &#39;ubuntu/trusty64&#39;...
==&gt; default: Matching MAC address for NAT networking...
==&gt; default: Checking if box &#39;ubuntu/trusty64&#39; is up to date...
==&gt; default: Setting the name of the VM: cloudapps_default_1461534978218_15922
==&gt; default: Clearing any previously set forwarded ports...
==&gt; default: Clearing any previously set network interfaces...
==&gt; default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
==&gt; default: Forwarding ports...
default: 4000 (guest) =&gt; 8100 (host) (adapter 1)
default: 22 (guest) =&gt; 2222 (host) (adapter 1)
==&gt; default: Booting VM...
==&gt; 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&#39;s present...
default: Key inserted! Disconnecting and reconnecting using new SSH key...
==&gt; default: Machine booted and ready!
==&gt; 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
==&gt; default: Mounting shared folders...
default: /vagrant =&gt; /home/duvan/cloudapps
==&gt; default: Running provisioner: shell...
default: Running: /tmp/vagrant-shell20160424- 28486-xrroay.sh
==&gt; default: stdin: is not a tty
==&gt; default: Installing: nodejs, lynx, ruby and jekyll...
==&gt; default: Configuration file: none
==&gt; default: Source: /vagrant
==&gt; default: Destination: /vagrant/_site
==&gt; default: Incremental build: disabled. Enable with --incremental
==&gt; default: Generating...
==&gt; default: done in 0.019 seconds.
==&gt; default: Auto-regeneration: enabled for &#39;/vagrant&#39;
==&gt; default: Configuration file: none
==&gt; default: Server address: http://0.0.0.0:4000/
==&gt; default: Server running... press ctrl-c to stop.
```

9. Verificar el correcto funcionamiento del despliegue accediendo en un navegador (browser) a la dirección: `http://127.0.0.1:8100`.

Se observa la correcta creación de la imagen en VirtualBox:

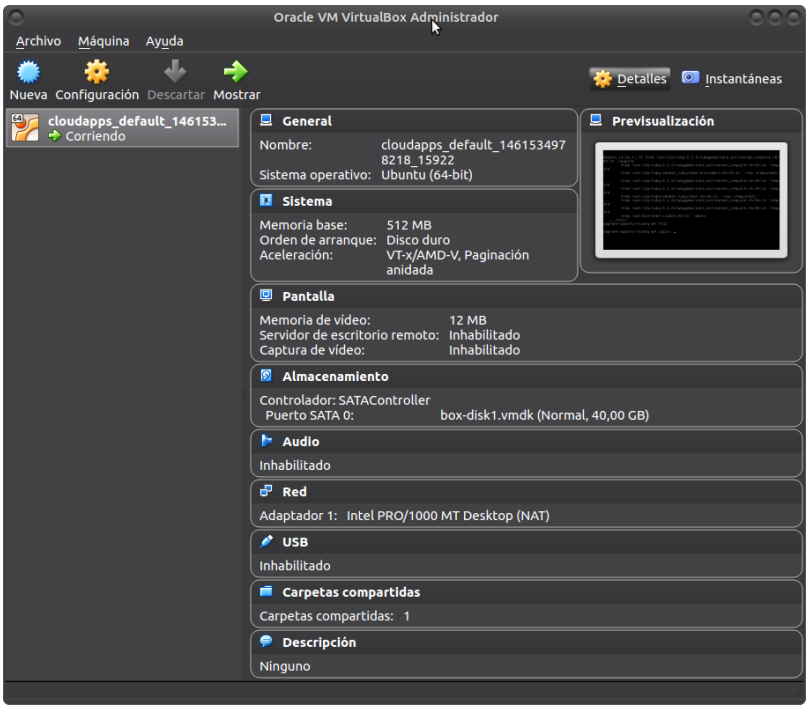


Figura 7: Imagen correctamente montana en VirtualBox.

Se procede a lanzar la maquina y verificar su funcionamiento en un navegador en la dirección: `http://127.0.0.1:8100`.

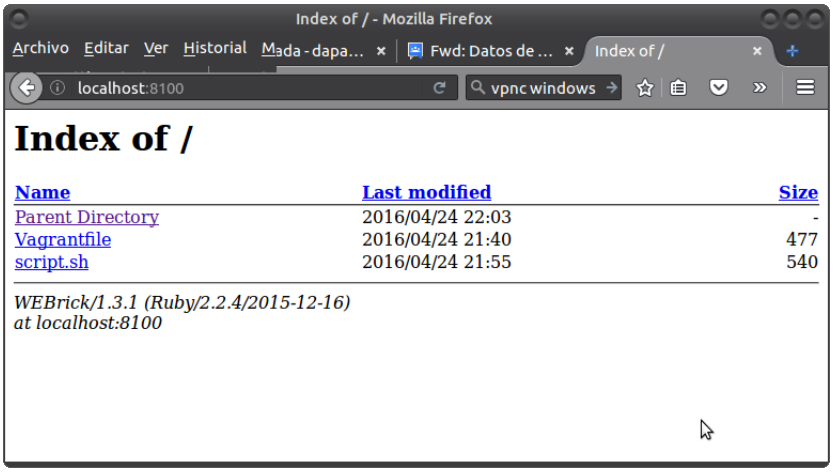


Figura 8: Validación del funcionamiento de la maquina virtual creada en Vagrant.

# 4. BIBLIOGRAFÍA


- https://www.vagrantup.com/about.html
- https://www.ottoproject.io/intro/vagrant-successor.html