

Configuración entorno virtualizado con Vagrant

Elaborado por: Prof. Oscar Mondragón

Objetivos

- Configurar el ambiente de desarrollo Vagrant + VirtualBox + Ubuntu que se usará en el curso.
- Aprender cómo se publican boxes en Vagrant Cloud.

1. Instalar última versión de VirtualBox

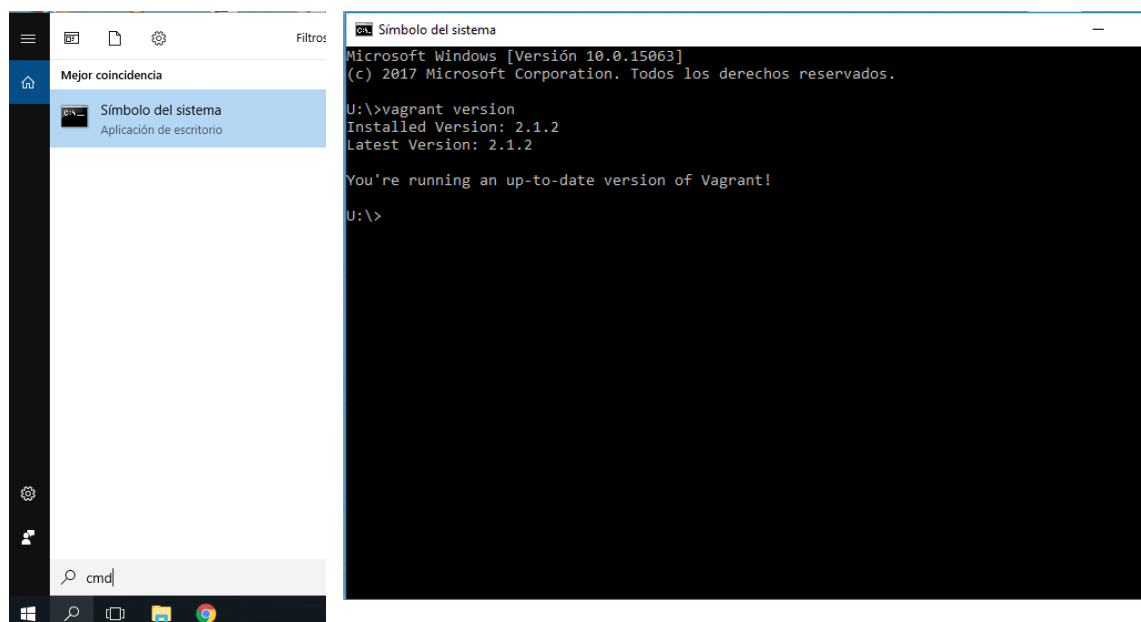
- Descarga e instala VirtualBox del sitio www.virtualbox.org

2. Instalar última versión de Vagrant

- Descarga e instala la ultima versión de Vagrant desde releases.hashicorp.com/vagrant/
- Por ejemplo: releases.hashicorp.com/vagrant/2.2.19/

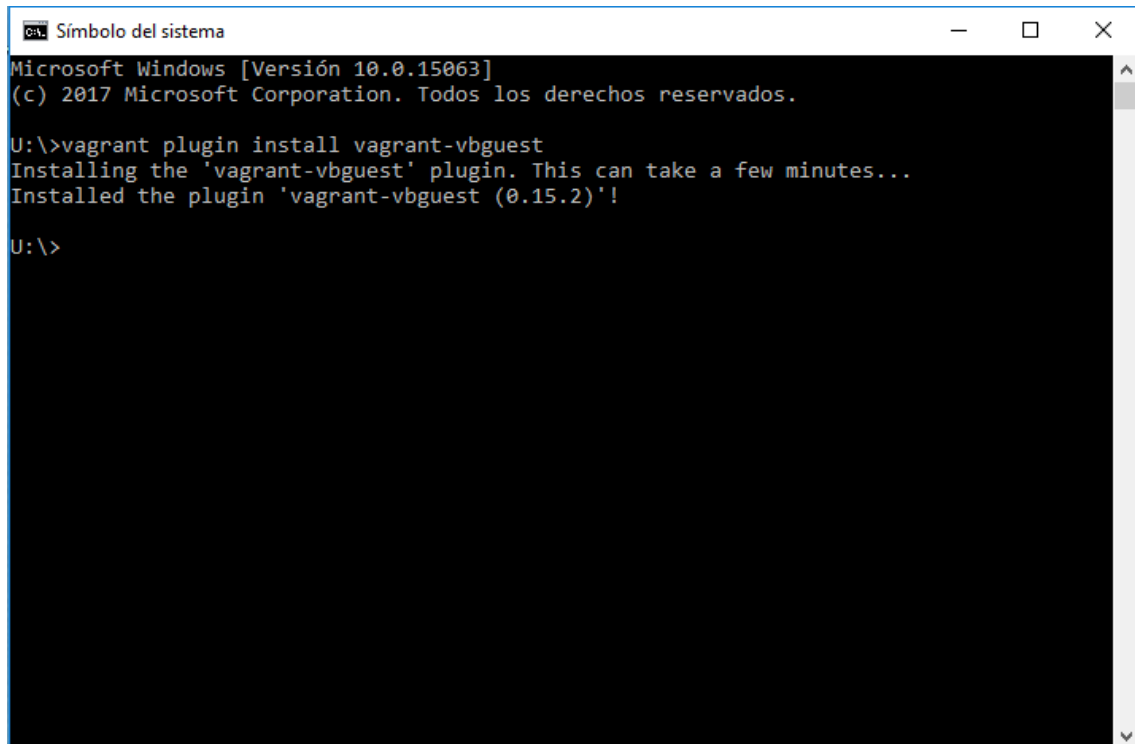
NOTA: En ese caso, usa la versión 2.2.19 para 64 bits (vagrant_2.2.19_x86_64.msi) (para usuarios Mac es el mismo archivo con extensión .dmg)

3. Abrir una consola de Windows y verificar la versión de Vagrant



- En la misma consola de Windows, ejecuta el siguiente comando para instalar el *plugin* vbguest

```
vagrant plugin install vagrant-vbguest
```



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.15063]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

U:\>vagrant plugin install vagrant-vbguest
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
Installed the plugin 'vagrant-vbguest (0.15.2)'\!

U:\>
```

Este es un *plugin* de Vagrant para mantener las adiciones de los *guest* de VirtualBox actualizadas.

4. Opcional. Powershell

Para algunos usuarios puede ser más cómodo usar Powershell, el cual es similar a los Shells de Unix. Este ya viene preinstalado en Windows 10.

Para ejecutarlo, puedes usar el buscador de aplicaciones de Windows o ejecutar “Powershell” directamente en una consola de Windows, como se muestra a continuación.

```

U:\>powershell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. Todos los derechos reservados.

PS U:\> pwd

Path
----
U:\

PS U:\> c:
PS C:\> pwd

Path
----
C:\

PS C:\> ls

    Directorio: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          24/07/2018   5:46 p. m.      HashiCorp
d-----          22/06/2017   3:44 p. m.      inetpub
d-----           1/02/2018   5:37 p. m.      Intel
d-----         20/03/2018  10:21 a. m.      PerfLogs
d-r-----        24/07/2018   5:37 p. m.      Program Files
d-r-----        23/07/2018   5:49 p. m.      Program Files (x86)
d-r-----        23/01/2018  10:22 a. m.      Users
d-----         24/07/2018   5:48 p. m.      Windows
d-----        23/01/2018  12:18 p. m.      Windows10Upgrade
d-----           4/07/2018   2:20 p. m.      Yopal
-a-----          11/07/2017   9:47 a. m.      52 network.lic

PS C:\>
  
```

Se puede ver en este ejemplo que se ejecutaron comandos como **pwd**, **ls** al estilo Unix. El comando **pwd** permite ver el directorio actual de trabajo, mientras que el **ls** el contenido de dicho directorio.

5. Configurar el entorno virtualizado

Muchos servicios de red usan el modelo cliente-servidor. Usaremos dos máquinas virtuales, una que aloja los servicios configurados y otra que los consume.

Configura el Vagrantfile de la siguiente manera (ver instrucciones abajo en “CONFIGURACIÓN”).

```

# -*- 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-20.04"
  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-20.04"
  servidorUbuntu.vm.network :private_network, ip: "192.168.100.3"
  servidorUbuntu.vm.hostname = "servidorUbuntu"
end
end
```

Este Vagrantfile define dos máquinas virtuales, una llamada **servidor** con dirección ip 192.168.100.3 y la otra **cliente** con dirección ip 192.168.100.2, ambas instanciadas desde un box en el repositorio de bento llamado bento/Ubuntu-20.04.

Nota: SOLO SI ES USUARIO MAC OSX:

En la máquina **anfitrión** crear el directorio /etc/vbox y dentro el archivo networks.conf (/etc/vbox/networks.conf): con el siguiente contenido:

```
* 192.168.100.0/24
```

CONFIGURACIÓN

1. Crea un directorio llamado "prueba"
2. Dentro del directorio prueba crea un archivo llamado "Vagrantfile".

Para crear un Vagrantfile de ejemplo puedes usar el siguiente comando dentro de la carpeta prueba:

```
vagrant init
```

Modifica el Vagrantfile para que tenga ÚNICAMENTE el contenido mostrado en el ejemplo de arriba (al inicio de la sección).

3. Crea y configura las máquinas mediante el comando `vagrant up` ejecutado desde la consola.

4. Verifica el puerto de reenvío para cada máquina virtual. En el ejemplo mostrado abajo, el puerto de reenvío es el 2222

```
$ vagrant up
```

5. Verifica el estado de las máquinas creadas con el comando

```
vagrant status
```

6. Establece una sesión ssh con la máquina servidor

```
vagrant ssh servidorUbuntu
```

7. Autenticate como super usuario

```
sudo -i
```

8. Instala algunas herramientas para la configuración de la red

```
apt-get install net-tools
```

9. Instala el editor Vim

```
apt-get install vim
```

10. Repite los pasos 5 al 8 para la máquina **cliente**.

11. Confirma la ip de las máquinas virtuales usando `ifconfig` y prueba la conectividad con el comando `ping`.

6. Detener o suspender una máquina virtual

vagrant suspend guarda el estado actual de la máquina y la detiene. Para volver a la máquina desde el punto en que la suspendió puede ejecutar **vagrant up**.

Vagrant halt apaga la máquina virtual de manera segura conservando los contenidos del disco y permitiendo un inicio seguro de nuevo. Para levantar la máquina de nuevo puede usar **vagrant up**.

Vagrant destroy remueve la máquina guest del sistema completamente. Para levantar la máquina de nuevo puede usar **vagrant up**.

7. Publicar Boxes en Vagrant Cloud

Una vez modificada la máquina virtual podemos crear un nuevo box y subirlo a Vagrant Cloud para usarlo posteriormente. Para eso seguiremos los siguientes pasos:

1. Re-empaquetar la máquina virtual en un nuevo Vagrant Box

```
vagrant package servidorUbuntu --output mynew.box
```

2. Agregar el box creado a su instalación de Vagrant.

El comando anterior creará un archivo `mynew.box`. Con el siguiente comando agregaremos el box a nuestra instalación de Vagrant:

```
vagrant box add mynewbox mynew.box
```

Esto permitirá usar el box desde cualquier ubicación en tu computador.

3. Publicar el box en Vagrant Cloud

- Diríjete a app.vagrantup.com/boxes/new
- Ingresa un nombre y una descripción para tu box.
- Crea la primera versión del box. Esta versión debe cumplir con el formato [0-9].[0-9].[0-9]. Por ejemplo 0.0.1.
- Crea un provider para el box. Virtualbox es el provider más común.
- Carga el archivo .box que corresponde al provider creado.
- Una vez cargado el box, lo puedes encontrar en la sección de boxes de <https://app.vagrantup.com/>
- Antes de usar una versión del box, debes liberarlo con “release”.
- Una vez creado y liberado un box, puedes liberar nuevas versiones haciendo clic en “create new version” en el menú de versiones de la página del box.

Referencias

Creating a New Vagrant Box.

<https://atlas.hashicorp.com/help/vagrant/boxes/create>

Ejercicios:

PARTE A

1. Crear el ambiente de trabajo inicial con una máquina cliente y una máquina servidor, de acuerdo al Vagrantfile mostrado en la sección 4.
2. Instalar net-tools y vim
3. Subir la imagen modificada a Vagrant Cloud
4. Realizar el taller Linux disponible en el sitio del curso
5. Investigar en qué consisten los **directorios sincronizados de Vagrant**. Demostrar su funcionamiento.

PARTE B

1. Investigar cómo funcionan los repositorios git.
 - a. Revisa el video de YouTube sobre los pasos básicos para empezar a usar GitHub: www.youtube.com/watch?v=SqbGliTKVoE&t=39s

- b. Si no lo tienes, crea un repositorio git en github (github.com/) y utilízalo desde tu máquina virtual "servidor" para subir y descargar archivos.
- c. En el repositorio, crea una estructura de directorios en la cual subirás los archivos relacionados con cada una de tus prácticas a lo largo del semestre. Por ejemplo:

```
.
├── compunube
│   ├── Practica1_Vagrant
│   └── Practica2_LXD
```

Nota: Recuerda instalar primero el cliente git en su máquina servidor.

REFERENCIAS

Vagrant. www.vagrantup.com/

GitHub. github.com/

Tutorial GIT. git-scm.com/docs/gittutorial

Ejercicios Linux. learnpythonthehardway.org/book/appendix-a-cli/ex1.html