Apuntes Examen Tema 02 de Optativa: DevOps.

Contenido

Comandos en CMD:

En CMD:

- Comprobar versión de Vagrant:

“vagrant --version”

- Descargarnos una BOX de Ubuntu:

vagrant box add ubuntu/trusty64

https://portal.cloud.hashicorp.com/vagrant/discover

- Listar cuantos BOX tenemos descargados en una máquina física:

vagrant box list

- Borrar una BOX del repositorio local:

vagrant box remove hashicorp/bionic64

- Actualizar una BOX del entorno actual:

vagrant box update

- Saber si la BOX del entorno actual está desactualizada:

vagrant box outdated

- Quitar versiones antiguas de una BOX. Si la BOX se está usando en se momento, pedirá confirmación:

vagrant box prune

- Crear un Vagrantfile:

vagrant init

- Arrancar una VM:

vagrant up

- Conectarnos a una VM:

https://developer.hashicorp.com/vagrant/docs/cli/ssh

“On a simple vagrant project, the instance created will be named default.

Vagrant will ssh into this instance without the instance name:”

vagrant ssh [name|id] [-- extra_ssh_args]

Ejemplo: **vagrant ssh**

vagrant ssh default

- Salir de la VM:

exit

- Apagar la VM:

vagrant halt

- Eliminar la VM:

vagrant destroy

No pregunta por confirmación: **vagrant destroy- f**

- Aplicar “provision” en una VM ya levantada:

vagrant provision

- Aplicar cambios de hardware en la VM:

vagrant reload

Para comprobar dentro de la VM la CPU y la RAM:

i. CPU: grep processor /proc/cpuinfo | wc -l
ii. RAM: grep MemTotal /proc/meminfo

- Comprobar cuantas maquinas hay en un proyecto activo:

vagrant status

- Comprobar cuantas máquinas hay en un equipo con varios proyectos:

vagrant global-status

SNAPSHOTS

- Crear un snapshot de una VM:

vagrant snapshot save [vm-name] NAME

Ejemplo: **vagrant snapshot save estado1**

- Restaurar una VM desde un snapshot:

vagrant snapshot restore [vm-name] NAME

Ejemplo: vagrant snapshot restore estado1

- Listar snapshots:

vagrant snapshot list

- Borrar snapshots:

vagrant snapshot delete [vm-name] NAME

Configuración VAGRANTFILE:

- Establecer el BOX que vamos a usar:

config.vm.box = "ubuntu/bionic64"

- Provisionar:

config.vm.provision "shell", inline: <<-SHELL

mkdir /home/vagrant/carpeta_nuevecita

SHELL

Provisionar con script: En este caso el script está en el mismo directorio que el Vagrantfile.

config.vm.provision "shell", path: "provision_nueva_carpeta.sh"

¿Qué hay en el script?

mkdir /home/vagrant/carpetaza

- Modifica los recursos que usa la VM

“Indica que cuando se use VirtualBox use solo una cpu y 1gb de ram”

```
 config.vm.provider "virtualbox" do |vb|
#   # Display the VirtualBox GUI when booting the machine
#   vb.gui = true
#
#   # Customize the amount of memory on the VM:
   vb.memory = "1024"
        vb.cpus = 1
 end
```

- Redes:

http://codigoelectronica.com/blog/configurar-redes-en-vagrant

a) NAT: Es la que existe por defecto. No hay que configurar nada.

NAT (Network Address Translation)

Propósito: Acceso a Internet desde la VM

Características:

- Configuración por defecto en Vagrant
- El host no puede acceder directamente a la VM
- Ideal para desarrollo aislado con conectividad externa

b) Red privada (Host-only):

https://developer.hashicorp.com/vagrant/docs/networking/private_network

Private Network (Host-Only)

Propósito: Comunicación host-VM y entre VMs

Características:

- **Red privada** entre host y máquinas virtuales
- Asignación de **IPs estáticas o DHCP**
- **Sin acceso directo a Internet** desde la red privada
- Perfecto para clusters y arquitecturas multi-máquina

Ejemplo de IP Estática:

config.vm.network "private_network", ip: "192.168.50.65"

Ejemplo de IP Dinámica:

config.vm.network "private_network", type: "dhcp"

Ejemplo de IP Estática en IPv6:

config.vm.network "private_network", ip: "fde4:8dba:82e1::c4"

c) Public Network (Bridged)

Propósito: VM como dispositivo de red real

Características:

- La VM aparece como **dispositivo físico en la red**
- Obtiene IP del router de la red local
- Accesible desde otros dispositivos de la red
- Útil para pruebas de producción y demos

config.vm.network "public_network"

Al ejecutar vagrant up, te preguntará a qué adaptador de red física conectarse.

Ventajas:

- Accesible desde otros dispositivos de la red.
- Útil para probar acceso externo o servicios públicos.

d) Mapeo de Puertos:

config.vm.network "forwarded_port", guest: 80, host: 8080

Si activamos auto_correct, nos modifica el puerto por si colisiona con alguno que ya esté en uso.

config.vm.network "forwarded_port", guest: 80, host: 80, auto_correct: true

- Carpetas sincronizadas:

Creamos la carpeta en el host y luego ponemos esto en el Vagrantfile:

config.vm.synced_folder "gimnasia_sincronizada", "/home/vagrant/gimnasia_sincronizada"

Otro ejemplo:

```
Vagrant.configure("2") do |config|
 config.vm.box = "hashicorp/bionic64"

 # Sincronización básica
 config.vm.synced_folder "./data", "/vagrant_data"

 # Sincronización con tipo rsync y opciones de propietario/grupo
 config.vm.synced_folder "./www", "/var/www/html",
  type: "rsync",
  owner: "www-data",
  group: "www-data"
end
```