

Vagrant



VAGRANT

- Vagrant
- Que é?
- Instalación de vagrant
- Que é un box?
- Crear unha máquina virtual sinxela.
- Configuración do escenario básico
- Vagrant redirección de portos

Que é?

- Vagrant é unha ferramenta deseñada para configurar e compartir o entorno de traballo coa mesma configuración.
- Emprega as tecnoloxías como os hipervisores de VirtualBox, VMWare, libvirt , ...
- Para adaptar unha máquina (box) podemos empregar:
 - Script shell
 - Ferramentas como: Chef, Puppet, Ansible.

Instalación de vagrant

- Documentación
- Linux

```
wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg  
  
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /  
etc/apt/sources.list.d/hashicorp.list  
  
sudo apt update && sudo apt install vagrant
```

- Windows

```
winget install --id=Hashicorp.Vagrant -e
```

Que é un box?

- Un box é unha máquina virtual "empaquetada", podemos velo como un modelo que vamos clonar ou replicar.
- Podemos consultar e usar Boxes publicados en <https://app.vagrantup.com/boxes/search>
- **Como podemos empregalos?**

```
vagrant box add {title} {url}
```

```
#Exemplo 01
```

```
vagrant.exe box add centos https://app.vagrantup.com/centos/boxes/7/versions/2004.01/providers/virtualbox.box
```

Xestión dun box

- Onde se atopan por defecto os box? No directorio do usuario:
 - Linux: /home/usuario/.vagrant.d/boxes
 - Windows: C:\Users\usuario\vagrant\

- Como consultamos os box existentes?

```
vagrant box list
```

- Como eliminamos o box creado chamado **centos** ?

- ```
vagrant.exe box remove centos
```

# Opciones con vagrant box

```
$ vagrant box [opción]
```

Subcomandos disponibles:

- add
- list
- outdated
- prune
- remove
- repackage
- update

# Crear unha máquina virtual sinxela.

- Para cada proxecto crearemos un directorio.
- En cada proxecto teremos un ficheiro chamado **Vagrantfile**

```
$ mkdir project
$ cd project
```

```
Creamos o ficheiro Vagrantfile co box ubuntu16.04
$ vagrant init bento/ubuntu-16.04
```

```
#Levantamos a máquina
$ vagrant up
```



# Como manipular a máquina (I)?

- Arrincar a máquina, a primeira vez tardará algo mais ao ter que descargar o **box**

```
$ vagrant up
```

- Deter a execución da máquina

```
$ vagrant halt
```

- Destruir os ficheiros da máquina virtual.

```
$ vagrant destroy
```

- Pausar a máquina virtual.

```
$ vagrant pause
```

- Reanudar a máquina virtual.

```
$ vagrant resume
```

## Como manipular a máquina? (II)

- Conectarse por SSH

```
$ vagrant ssh
```

- Sair da máquina virtual. **exit**

```
ubuntu@ubuntu-xenial:~$ exit
```

- Validar unha configuración do ficheiro Vagrantfile

```
$ vagrant validate
```

- Obter axuda sobre algún comando

```
$ vagrant COMMAND -h
```

- Vagrant modo debug

```
$ vagrant up --debug
```

# Configuración do escenario básico

Precisamos crear unha máquina virtual con Debian Bullseye 64 bits. Que pasamos debemos seguir?

```
#1 Crear o directorio do escenario
mkdir escenario-debian
cd escenario-debian
#2- Baixar o box
vagrant box add "deb/bull"
https://app.vagrantup.com/debian/boxes/bullseye64/versions/11.20221219.1/providers/virtualbox.box
#3- Crear o Vagrantfile plantilla
vagrant init deb/bull

Revisa o ficheiro **Vagrantfile** creado
cat Vagrantfile
```

## Configuración do escenario básico (II)

Precisamos as seguintes características:

- Proveedor de virtualización **virtualbox**
- Nome da MV: "probas"
- N° de CPUS: 2
- RAM : 2GiB
- Hostname: p2

```
config.vm.provider :virtualbox do |vb|
 vb.memory = "2048"
 vb.cpus = 2
 vb.name = "probas"
end
end
```

## ex02

### Configuración do escenario básico (III)

- Levanta a máquina e accede por ssh

- ```
vagrant up  
vagrant ssh
```

- Como comprobamos os portos empregados entre a MV e o host?

- ```
vagrant port
```

- Como paramos a máquina virtual?

## Configura o escenario para que arrinque a interfaz gráfica.

- Recorda que estamos empregando de proveedor Virtualbox.

- ```
Vagrant.configure("2") do |config|  
  config.vm.box = "deb/bull"  
  config.vm.hostname = "p2"  
  
  config.vm.provider "virtualbox" do |vb|  
    vb.memory = "2048"  
    vb.cpus = 2  
    vb.name = "probas"  
    vb.gui=true  
  end  
end
```

ex03

Configuración do escenario básico: que instale un aplicativo.

- Adapta o vagrantfile para que instale o servidor web apache2.
- O ficheiro de instalación debe facerse dentro do propio ficheiro Vagrantfile

- ```
$ sudo apt update
$ sudo apt install apache2
$ echo "<h1>o meu servidor web</h1>" >>/var/www/index.html
```

ex04

## Solución

```
Vagrant.configure("2") do |config|
 config.vm.box = "deb/bull"
 config.vm.hostname = "p2"

 config.vm.provider "virtualbox" do |vb|
 vb.memory = "2048"
 vb.cpus = 2
 vb.name = "probas"
 vb.gui=true
 end

 config.vm.provision "shell", inline: <<-SHELL
 sudo apt update
 sudo apt install apache2 -y
 SHELL
end
```



## Adapta o escenario cun ficheiro script externo.

- Neste caso o ficheiro externo chamarase **script.sh**
- O ficheiro **script.sh** ten que mostrar a seguinte mensaxe **OLA MUNDO**

ex05

- ```
config.vm.provision "shell", path: "script.sh"
```

- Arquivo **script.sh**

```
#!/bin/bash  
echo "ola mundo"
```

Vagrant redirección de portos

```
Vagrant::Config.run do |config|  
  # Redirecciona o porta 80 do guest para a porto 4567 do host  
  config.vm.forward_port 80, 4567  
end
```

- forward_port é um método que recolle dous argumentos:
 - guest port - O porto na máquina virtual
 - host port - O porto na máquina local que imos empregar para acceder.
- A redirección de portos aplícase durante o **vagrant up** , mais tamén podemos forzar isto unha vez lanzada con **vagrant reload**

Adapta o escenario facendo nat.

- Neste caso vamos redireccionar o porto 80 do servidor web apache ao porto 8080 no anfitrión.
- ```
config.vm.network "forwarded_port", guest: 80, host: 8080
```
- [ex06](#)

```
Vagrant.configure("2") do |config|
 config.vm.box = "deb/bull"
 config.vm.hostname = "p2"

 config.vm.provider "virtualbox" do |vb|
 vb.memory = "2048"
 vb.cpus = 2
 vb.name = "probas"
 # vb.gui=true
 end
 #config.vm.provision "shell", path: "script.sh"
 config.vm.provision "shell", inline: <<-SHELL
 sudo apt update
 sudo apt install apache2 -y
 SHELL
 #Configuración portos
 config.vm.network "forwarded_port", guest: 80, host: 8080

end
```

# Configuración do escenario para compartir directorios.

- Documentación
- Na máquina de referencia tes que:
  - i. Crear un directorio chamado **tmp** no directorio do proxecto vagrant.
  - ii. Editar o **Vagrantfile** para que compartas o directorio creado no host dentro da máquina virtual no directorio **/tmp/src**
  - iii. Levanta a máquina e accede nela para crear un ficheiro de texto chamado **oTeuNome.txt**
  - iv. Sae da MV e revisa o directorio **tmp** no host. Apareceu algo novo?

## ex07

- `config.vm.synced_folder "tmp", "/tmp/src"`

# Discos

- [Documentación](#)
- Existen 3 tipos de discos que nos permite Vagrant, estos son: **disk**, **dvd**, **floppy**
- Exemplo:

```
config.vm.disk :disk, name: "backup", size: "10GB"
config.vm.disk :dvd, name: "installer", file: "./installer.iso"
config.vm.disk :floppy, name: "cool_files"
```

