

APRENDE VAGRANT CON 10 PRÁCTICAS

- parte 2 -

PRÁCTICA 8

¿QUÉ ES UN SCRIPT?

8- ¿QUÉ ES UN SCRIPT?

- Un script es un archivo de tipo texto plano que contiene ordenes/comandos que puede interpretar el sistema operativo al que está dirigido.
- Para UNIX o GNU/Linux usaremos la extensión .sh
- Para Windows usaremos la extensión .bat, si son comandos MS-DOS, o la extensión .ps1, si son cmd-lets de PowerShell.

8- ¿QUÉ ES UN SCRIPT?

- En un script GNU/Linux es importante que la primera línea del script sea:
 - `#!/bin/bash`
 - Se utiliza esa línea para indicar la SHELL con la que vamos a trabajar
- El resto de las líneas del script son los comandos **en orden** que queremos que se ejecuten.

8- ¿QUÉ ES UN SCRIPT?

- Generando un script que podamos utilizar con Vagrant:
 - 1- Creamos el script con los comandos que necesitemos para preparar nuestra máquina.
 - IDEA: Podemos añadir líneas de echo “Texto” para ir sabiendo por donde se está ejecutando el script.
 - 2- Si necesitamos archivos de configuración, páginas que aparezcan, documentos, paquetes...podemos dejarlos en la carpeta compartida por defecto.
 - 3- En el Vagrantfile añadimos:
 - **etiqueta**.vm.provision :shell, :path => **"/ruta/script.sh"**, privileged: false
 - ETIQUETA: El nombre de la etiqueta de la máquina que estamos configurando.
 - RUTA: Ruta de la carpeta con el script en caso de que sea necesario.

8- ¿QUÉ ES UN SCRIPT?

- Otra posibilidad es escribir el propio SCRIPT dentro del bloque provision del VagrantFile:
 - El siguiente script sencillo realiza un update de los paquetes y nos informa de que lo está realizando:

```
config.vm.provision "shell", inline: <<-SHELL
  echo "Vamos a actualizar los paquetes"
  sudo apt-get update
  echo "Paquetes actualizados"
SHELL
```

8- ¿QUÉ ES UN SCRIPT?

PRÁCTICA: NUESTRO PRIMER SCRIPT

- Vamos a crear un script para GNU/Linux, denominado, crear-carpeta.sh que se encargue de crear una carpeta nueva llamada “script” en nuestra máquina. Además copiará un archivo de texto que tendremos en la carpeta compartida a la carpeta “script”. Nos indicará con mensajes lo que hace.
- Piensa donde guardar el script, que será llamado desde el fichero VagrantFile.
- Forzaremos la ejecución del script: vagrant provision

PRÁCTICA 9

AÑADIENDO TARJETAS DE RED

9- AÑADIENDO TARJETAS DE RED

- Vamos a preparar el VagrantFile para que nuestras máquinas virtuales aparezcan con una o varias tarjetas de red completamente configuradas.
- Nos será útil a la hora de tener por ejemplo una red interna o sólo anfitrión, además de la tarjeta por defecto de tipo NAT.
- Podemos ver el manual en el siguiente enlace:
 - <https://www.vagrantup.com/docs/networking/>

9- AÑADIENDO TARJETAS DE RED

- LA PRÁCTICA 9-1 CONSISTE EN
 - REALIZAR LA TAREA QUE ESTA EN IKAS : “Tarea método network”



9- AÑADIENDO TARJETAS DE RED

- PRÁCTICA 9-2: Inicia tu máquina virtual debian-10 con dos tarjetas de red: una en local y otra la tarjeta por defecto que esta en modo NAT. Modifica el Vagrantfile como sea necesario.
- Ejemplo para la local:

```
server.vm.network "private_network",  
  ip: "192.168.10.10",  
  netmask: "255.255.255.0",  
  virtualbox__intnet: "intnet"
```

PRÁCTICA 10

ESCENARIO CON MULTIPLES MAQUINAS

10- EL VAGRANT FILE

- Tenemos multitud de opciones para añadir a la configuración, podemos ver el manual aquí:
 - <https://www.vagrantup.com/docs/vagrantfile/>
- En esta práctica vamos a ver como configurar un Vagrantfile para poder inicializar dos o más máquinas virtuales utilizando un único VagrantFile.

10- EL VAGRANT FILE

- `vagrant.configure("2") do |config|` → Inicializamos la configuración.
→ Fíjate en la etiqueta `config`
- `config.vm.define "servidor" do |servidor|` → Definimos la primera máquina virtual.
→ Fíjate en la etiqueta `servidor`
- `servidor.vm.box = "ubuntu/trusty64"` → Indicamos que box utiliza la primera máquina.
- `end` → Final de la primera máquina virtual

10- EL VAGRANT FILE

....

- `config.vm.define "cliente" do |cliente|` → Definimos la segunda máquina virtual.
→ Fíjate en la etiqueta **cliente**
- `cliente.vm.box = "ubuntu/trusty64"` → Indicamos que box utiliza la segunda máquina.
- `end` → Final de la segunda máquina virtual.
- `end` → Final de todo.

10- EL VAGRANT FILE

- A tener en cuenta el nombre que definimos entre | | sirve para referirnos a él a la hora de realizar la configuración. Se denomina etiqueta y es propio del lenguaje Ruby:
 - `config.vm...` estamos realizando una configuración sobre el total del vagrantfile.
 - `servidor.vm....` Estamos realizando una configuración para la máquina que hemos llamado servidor.
 - `cliente.vm....` Estamos realizando una configuración para la máquina que hemos llamado cliente.

10- EL VAGRANT FILE

- PRÁCTICA 10:
 - Crea un VagrantFile que inicie dos máquinas virtuales basadas en la box bento/debian-10, que estén en la red interna denominada “lan” de clase C (192.168.numPC.0)
 - La primera se llamará srv-tunombre
 - La segunda se llamara pc-tunombre
 - ¿Como accedes a cada una de las máquinas?
 - Comprueba que conectividad entre ellas.



10- EL VAGRANT FILE

- PRÁCTICA 10:
 - Sube el VagrantFile a tu repositorio de GitHub asignado en ikas a esta tarea.



**¡ENHORABUENA, AHORA PUEDES
EMPEZAR A UTILIZAR VAGRANT!**

PRÁCTICA FINAL SERVIDOR WEB

CREAMOS SERVIDOR WEB

PRÁCTICA FINAL:

- Crea un VagrantFile con un script, escrito dentro de él, para iniciar automáticamente una máquina que sea un servidor web basado en Apache.
- Utiliza la máquina bento/debian-10
- La máquina tendrá el nombre srv-web-tunombre
- Configura 2 tarjetas (NAT y Red Interna).
- Comprueba que al iniciar tu máquina todo funciona como esperabas

CREAMOS SERVIDOR WEB

- PRÁCTICA FINAL:
 - Sube el VagrantFile a tu repositorio de GitHub asignado en ikas a esta tarea.

