

Workshop 01 – Entorno base con Debian/Bookworm

Objetivo

Aprovisionar en un equipo de desarrollo un ambiente equiparable al que se encontraría en un servidor GNU/Linux de producción, usando Vagrant y Debian Bookworm.

INSTALACIÓN Y CONFIGURACIÓN INICIAL

1. Se instaló VirtualBox.
 2. Se instaló Vagrant.
 3. Se abrió Git Bash.
 4. Se abrió VirtualBox.
 5. Se fue a Administrador de red (Archivo > Herramientas > Administrador de red) y se verificó que en Host-only Networks hubiera una entrada con una IP asignada.
 6. Recomendación: No tener el escritorio sincronizado con OneDrive, ya que puede generar errores al usar Git Bash.
-

COMANDOS UTILIZADOS

Navegación y manejo de directorios

<code>pwd</code>	Imprime la ruta donde se está trabajando.
<code>mkdir ISW811</code>	Crea el directorio llamado ISW811.
<code>ls</code>	Muestra los archivos que hay en el directorio actual.
<code>ls -la</code>	Muestra los archivos con más detalles (permisos, usuario, etc.).
<code>cd ISW811/</code>	Entra al directorio ISW811.
<code>cd ..</code>	Vuelve al directorio padre.
<code>cd</code>	Va a la carpeta principal del usuario.

Estructura de carpetas y comandos utilizados

Dentro de la carpeta principal **K**, se creó una subcarpeta llamada **VMs** para organizar mejor los proyectos virtualizados.

Para comprobar que las carpetas se crearon correctamente y visualizar su contenido, se realizó un listado detallado.

Posteriormente, dentro de la carpeta **VMs**, se creó otra subcarpeta llamada **WebServer**, donde se alojarán los archivos específicos del servidor web.

Los comandos usados fueron:

```
Crear la carpeta principal K
mkdir K

Crear la carpeta VMs dentro de K
mkdir K/VMs

Verificar que la carpeta K contiene la carpeta VMs
ls -la K

Crear la carpeta WebServer dentro de VMs
mkdir K/VMs/WebServer
```

Manejo de archivos

```
Para mover un archivo a una carpeta
mv <archivo> <carpeta_destino>

Para eliminar una carpeta
rm -rf ./private # (ejemplo de carpeta a eliminar)
```

Inicialización de VAGRANT

Paso 1: Inicializar la máquina virtual con Debian

Ejecuta el siguiente comando en la terminal para crear el entorno Vagrant con la caja oficial de Debian Bookworm (64 bits):

```
vagrant init debian/bookworm64
```

Paso 2: Editar el archivo Vagrantfile para configurar la red

Abre el archivo Vagrantfile con tu editor de código. Por ejemplo:

```
code Vagrantfile
```

Una vez dentro del archivo:

1. Se descomenta la línea 35 (que usualmente configura la red privada).
2. Se cambia la IP, usando .56 en lugar de .33 y asegurándose de que termine en .10.

```
Vagrant.configure("2") do |config|

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "debian/bookworm64"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine and only allow access
  # via 127.0.0.1 to disable public access
  # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"

  # Create a private network, which allows host-only access to the machine
  # using a specific IP.
  config.vm.network "private_network", ip: "192.168.56.10"

  # Create a public network, which generally matched to bridged network.
  # Bridged networks make the machine appear as another physical device on
  # your network.
  # config.vm.network "public_network"

  # Share an additional folder to the guest VM. The first argument is
  # the path on the host to the actual folder. The second argument is
  # the path on the guest to mount the folder. And the optional third
  # argument is a set of non-required options.
  # config.vm.synced_folder "~/data", "/vagrant_data"
```

Paso 3: Levantar la máquina virtual

Guarda los cambios y levanta la máquina con:

```
vagrant up
```

Esto descargará la caja de Debian (si no está previamente descargada) y configurará la máquina con la IP privada indicada.

Verificar el estado de la máquina Y COMANDO

```
vagrant status
```

Verificar conectividad con ping usando la IP configurada (ej. 192.168.56.10) Y COMANDO

```
ping 192.168.56.10
```

Conectarse a la máquina virtual Y COMANDO

```
vagrant ssh
```