

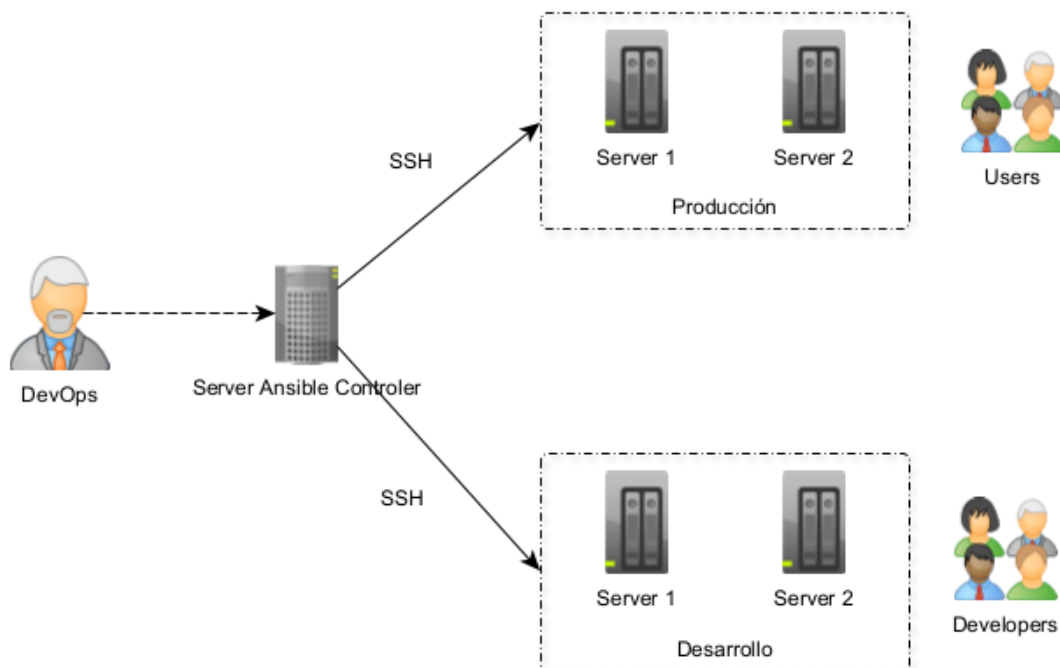


Infraestructura I

Ansible en acción

Creación del ambiente de trabajo

Vamos a crear nuestro servidor Linux con Ansible y dos entornos de trabajos (Desarrollo y Producción), cada uno con dos servidores. Tal como podemos observar en la siguiente imagen:



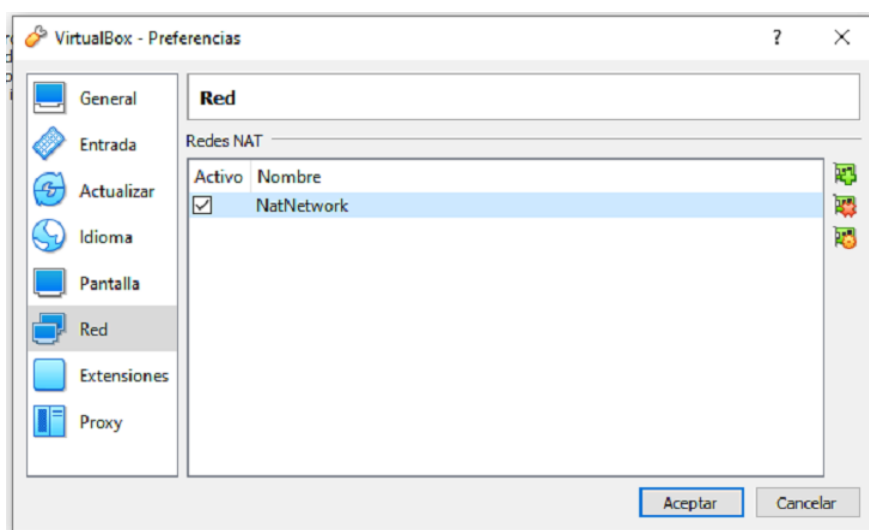
Generación de las MV

Hasta ahora hemos visto la virtualización a través VirtualBox y Vagrant. Ahora, vamos a agilizar la creación de máquinas virtuales y vamos a descargarlas completas en el siguiente link: <https://descargarmaquinasvirtuales.com>.

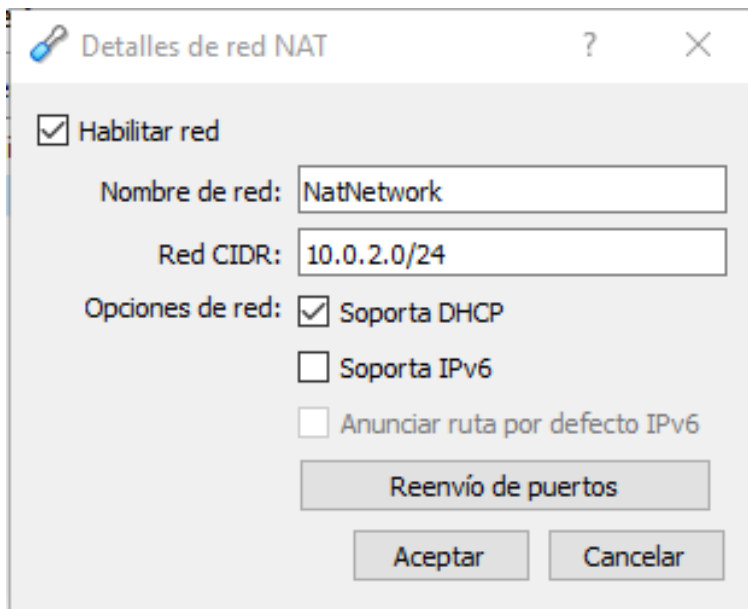
Descargamos la imagen de **Debian Gnome v10 de 2612 MB** para el server donde vamos a instalar **Ansible**, y la imagen de **Debian Server v10 de 676 MB** para los servers de "Producción" y "Desarrollo".

Distribución	Tipo	Versión	Tamaño (MB)	Descarga VirtualBox	Fecha
CentOS	Escritorio (Gnome)	8	2752	Descargar	05/12/2020
CentOS	Servidor	8	1060	Descargar	05/12/2020
CentOS	Escritorio (Gnome)	7	1930	Descargar	25/10/2017
CentOS	Servidor	7	1031	Descargar	28/03/2020
Debian	Escritorio (Gnome)	10	2612	Descargar	28/11/2020
Debian	Servidor	10	676	Descargar	28/11/2020
Debian	Escritorio (Gnome)	9	2588	Descargar	25/10/2017
Debian	Servidor	9	476	Descargar	24/10/2017
Fedora	Escritorio (Gnome)	33	5643	Descargar	30/11/2020

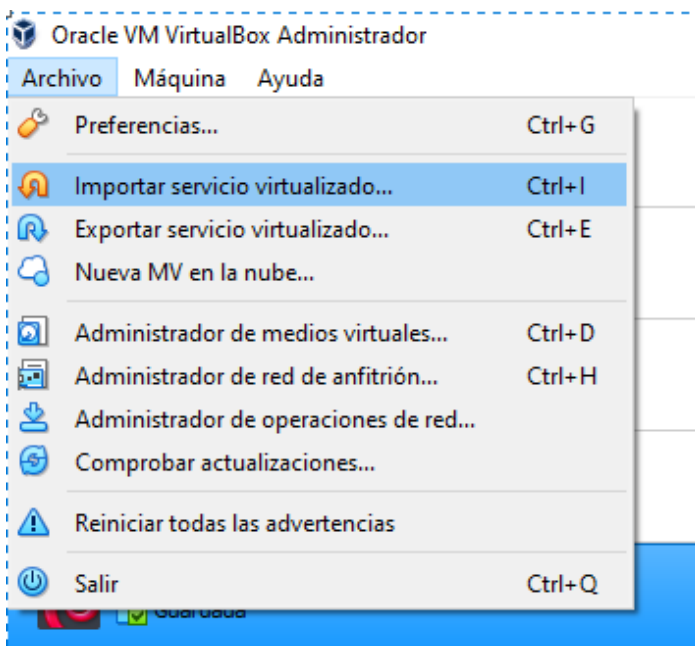
El próximo paso es abrir **Oracle VM VirtualBox**.



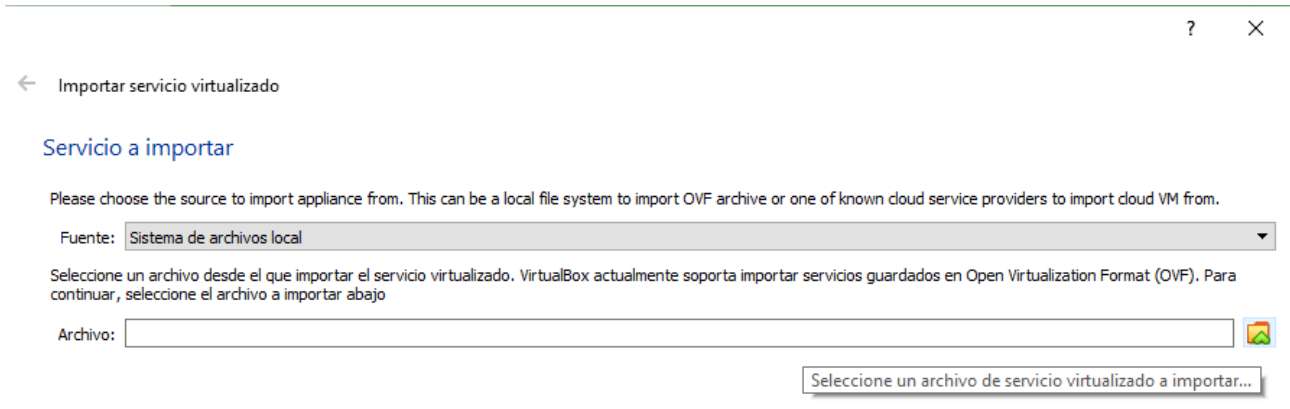
Vamos a “Preferencias” y creamos una nueva **Red NAT**. En este caso, creamos la **NatNetwork**.



Con IPs en la red: **10.0.2.0/24**. Luego, aceptamos y continuamos. Ahora, en Oracle **VM VirtualBox**, vamos a “Archivo” y seleccionamos “**Importar servicio virtualizado**”.



Vamos a “Archivo”, “**Importar servicio virtualizado**”. El próximo paso es **seleccionar** el archivo de VM descargado.



← Importar servicio virtualizado

Servicio a importar

Please choose the source to import appliance from. This can be a local file system to import OVF archive or one of known cloud service providers to import cloud VM from.

Fuente: Sistema de archivos local

Seleccione un archivo desde el que importar el servicio virtualizado. VirtualBox actualmente soporta importar servicios guardados en Open Virtualization Format (OVF). Para continuar, seleccione el archivo a importar abajo

Archivo:

Seleccione un archivo de servicio virtualizado a importar...

Luego, cambiamos el nombre del VM para que podamos identificarlo.



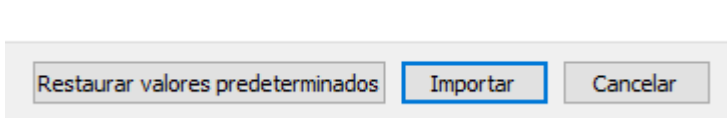
← Importar servicio virtualizado

Preferencias de servicio

Estas son las máquinas virtuales contenidas en el servicio y las preferencias sugeridas de las máquinas virtuales importadas de VirtualBox. Puede cambiar varias de las propiedades mostradas haciendo doble clic en los elementos y deshabilitar otras usando las casillas de verificación de abajo.

Sistema virtual 1	
 Nombre	debian-10-amd64-Ansible
 Tipo de SO invitado	 Debian (64-bit)
 CPU	2
 RAM	2048 MB
 DVD	<input checked="" type="checkbox"/>
 Tarjeta de sonido	<input checked="" type="checkbox"/> ICH AC97
 Adaptador de red	<input checked="" type="checkbox"/> Intel PRO/1000 MT Desktop (82540EM)
 Controlador de almacenamiento (IDE)	PIIX4
 Controlador de almacenamiento (IDE)	PIIX4
 Controlador de almacenamiento (SATA)	AHCI
 Imagen de disco virtual	debian-10-desktop-amd64-disk001.vmdk
 Carpeta base	C:\Users\david\VirtualBox VMs
 Grupo primario	/

Seleccionamos la opción de “Generar nuevas direcciones MAC ...”. Una vez finalizado hacemos clic en “Importar”.



Restaurar valores predeterminados **Importar** Cancelar

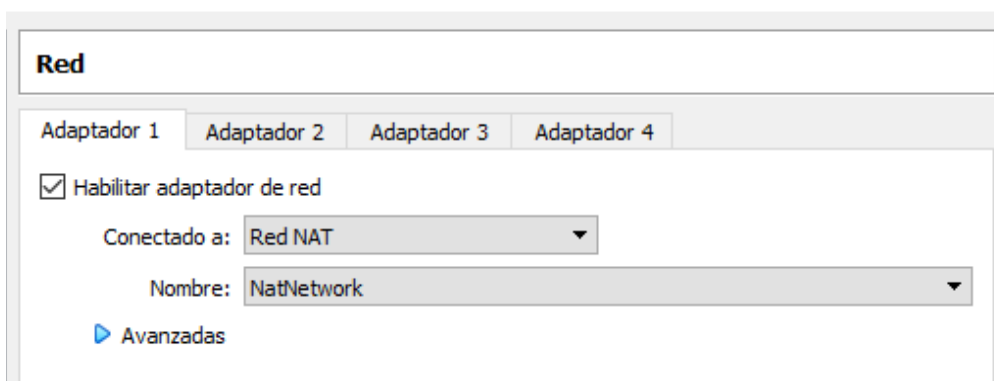
Repetimos estos pasos con cada VM que vamos a utilizar en nuestros entornos. Podemos hacer una prueba más reducida de acuerdo al **hardware disponible** con el que contamos, ya que puede ser que no tengamos disponible una máquina real que soporte todo este entorno. La versión reducida son dos VM (**un servidor de Ansible y un cliente**). Nuestro **VirtualBox** debería verse así.



Login:

- Usuario: **root**
- Contraseña: **toor**
- Usuario: **usuario**
- Contraseña: **usuario**

En la configuración de la red, tenemos que colocar la Red NAT para que tener acceso a la red.



Ayuda de comandos

Login:

user: **usuario**

user: **root**

pass: **usuario**

pass: **toor**

Para saber la IP de nuestro servidor debemos correr el siguiente comando:

```
ip a
```

```
root@debian-10-server-amd64:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:e6:f2:7c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.8/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 495sec preferred_lft 495sec
    inet6 fe80::a00:27ff:fe66:f27c/64 scope link
        valid_lft forever preferred_lft forever
root@debian-10-server-amd64:~# _
```

NOTA:

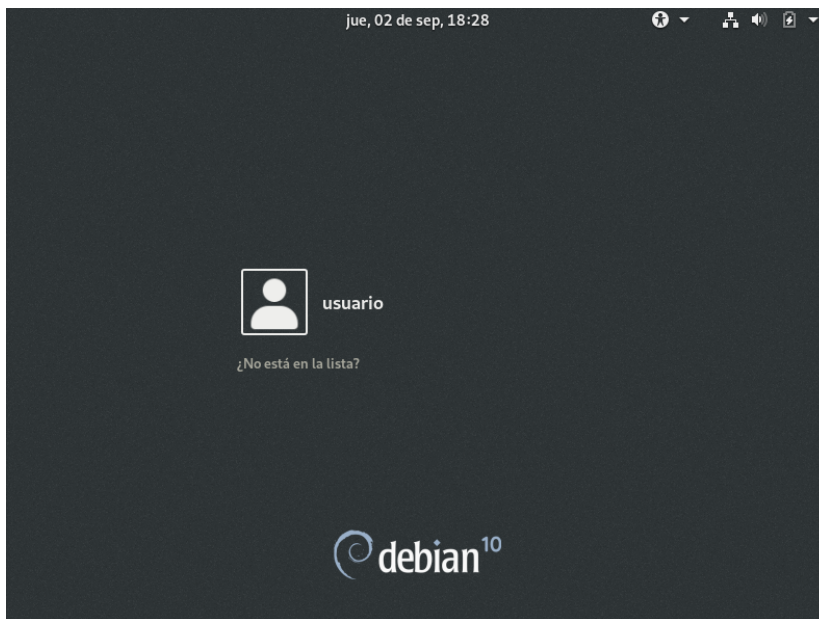
Ver el contenido de **“Configuración de IP en MV”** para configurar las IP de los hosts.

Instalar Ansible en nuestro servidor

Levantamos nuestro servidor de **Ansible**.

user: *usuario*

pass: *usuario*



Ansible

Abrimos una terminal:

```
sudo apt-get install ansible
```

Inventario

Ansible trabaja con un inventario de servidores. Este inventario se configura en el fichero (hay que crearlo si no existe) **'/etc/ansible/hosts'** y tiene la siguiente forma:

```
cd /etc/ansible/  
sudo nano hosts
```

```
[desarrollo]  
10.0.2.10 ansible_ssh_user=root  
10.0.2.8  ansible_ssh_user=root  
  
[producción]  
10.0.2.7 ansible_ssh_user=root  
10.0.2.9 ansible_ssh_user=root
```

Tal como se ve es un listado de IPs que se agrupan. Por ejemplo, aquí el grupo de servidores “**desarrollo**” tiene los servidores 10.0.2.10 y 10.0.2.8. (los IPs dependen de la red que tengan configurada).

Configuración de los certificados de seguridad

Primero tenemos que habilitar el usuario root en los hosts para que se pueda acceder por ssh, ya que por defecto viene bloqueado por seguridad.

```
root@debian-10-server-amd64: cd /etc/ssh
root@debian-10-server-amd64:/etc/ssh# nano sshd_config
```

Ahora tenemos que agregar la siguiente línea:

```
PermitRootLogin yes
```

Grabamos con presionando **Ctrl+O** y salimos con **Ctrl+X**, y luego reiniciamos el servicio SSH con el siguiente comando.

```
root@debian-10-server-amd64:/etc/ssh# systemctl restart ssh
```

Configuración de los certificados de seguridad

Ansible se comunica con los otros servidores vía SSH y un certificado de seguridad.

Ahora toca crear este certificado de seguridad (*recomendación no agregar frase*):

```
ssh-keygen
```

Después tendremos que copiar el certificado de seguridad a los distintos servidores:

```
ssh-copy-id root@10.0.2.8
```


Esto nos pedirá la clave del usuario SSH del servidor y copiará el certificado. Una vez todo instalado y configurado podemos probar de hacer un ping, para chequear que funcione:

```
ansible all -m ping
```

Debemos habilitar el usuario ssh y copiar el certificado en todos los hosts. Deberíamos ver:

```
usuario@debian-10:/etc/ansible$ ansible all -m ping
10.0.2.8 | SUCCES => {
  "changed": false,
  "ping": "pong"
}
10.0.2.7 | SUCCES => {
  "changed": false,
  "ping": "pong"
}
10.0.2.10 | SUCCES => {
  "changed": false,
  "ping": "pong"
}
```

Ejecutando comandos Ansible

Ahora que ya tenemos todo instalado y configurado ya podemos empezar a ejecutar comandos. La sintaxis de Ansible es la siguiente:

```
ansible <servidor/grupo/pattern> -m <módulo> -a <argumentos>
```

Podés encontrar más información en el siguiente link de la documentación oficial de Ansible: <https://docs.ansible.com>.

