EN DESARROLLO

Instalación PXE

Este proyecto consiste enla instalación de un servidor PXE para instalar un sistema operativo Open SUSE Leap 15.5.

En mi caso he decidido hacerlo con Vagrant para facilitar el proceso de crear las máquinas virtuales con las características necesarias.

Máquinas que utilizaremos:

- Servidor PXE (Debian 11)
- Router-Firewall (Debian 11)
- Cliente (Máquina vacía)

La máquina de Servidor PXE será la que sirva los archivos necesarios para realizar la instalación a través de la red y el Router-Firewall redirigirá el tráfico de internet a la máquina cliente y viceversa para que así esta tenga conexión para realizar todas las descargas necesarias.

Vagrantfile que utilizaré es el siguiente:

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

#Definimos la máquina virtual del servidor
  config.vm.define "serverPXE" do |subconfig|
    #Indicamos el sistema operativo
    subconfig.vm.box = "debian/bullseye64"
    subconfig.vm.hostname = "serverPXE"
    #Indicamos la ip que tendrá dentro de nuestra lan
    subconfig.vm.network :private_network, ip: "192.168.1.10",
    virtualbox__intnet: "PXElan"

    subconfig.vm.provider :virtualbox do |vb|
      vb.name = "serverPXE"
      vb.gui = false
      vb.memory = "4096"
      vb.cpus = "4"
    end
  end

  #Creamos la máquina cliente
  config.vm.define "client", autostart: false do |cli|
    cli.vm.box = "TimGesekus/pxe-boot"
    cli.vm.hostname = "client"
    cli.ssh.connect_timeout = 1
    #Indicamos que obtendrá dirección IP por dhcp
    cli.vm.network "private_network", type: "dhcp",
```

PROF

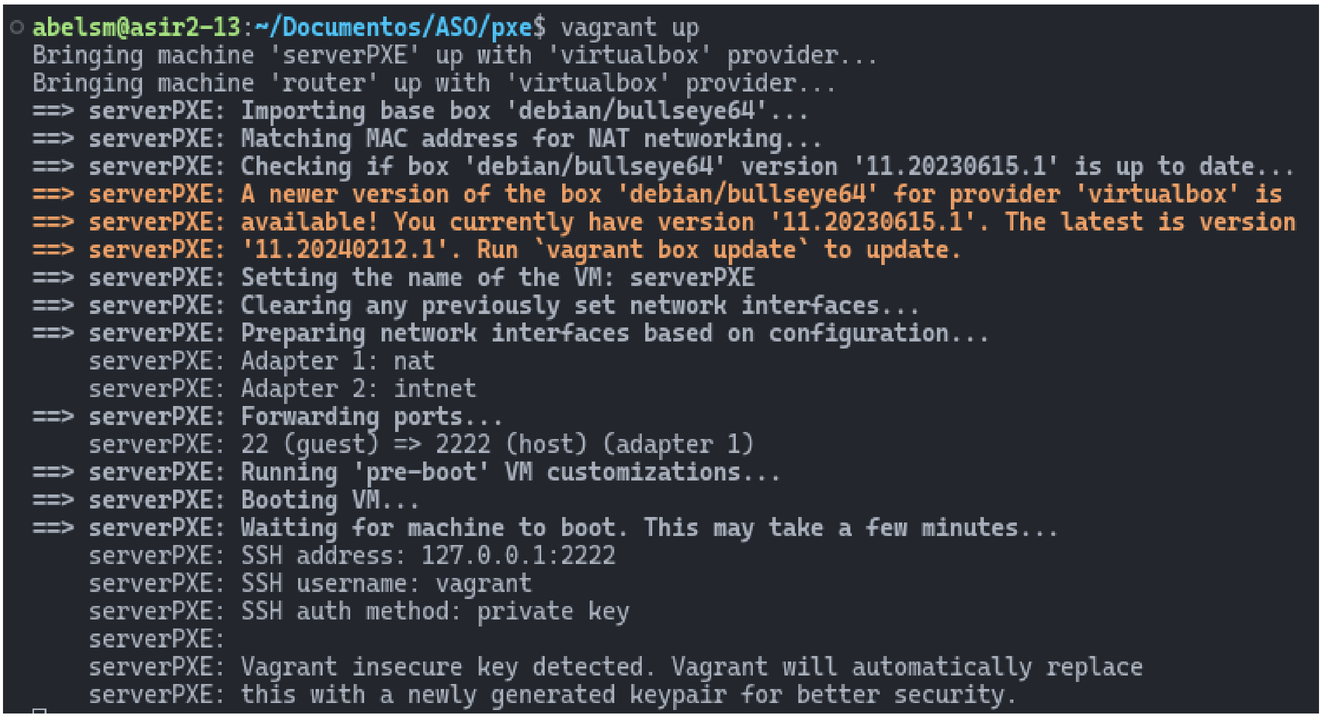
```
    :adapter => 1, virtualbox__intnet: "PXElan"

    cli.vm.provider :virtualbox do |vb|
      vb.name = "client"
      vb.gui = true
    end
  end

  #Creamos la máquina del router
  config.vm.define "router" do |subconfig|
    subconfig.vm.box = "debian/bullseye64"
    subconfig.vm.hostname = "router"
    subconfig.vm.network :private_network, ip: "192.168.1.1",
    virtualbox__intnet: "PXElan"

    #Ejecutamos en ella los comandos necesarios para convetirlo en un
router
    subconfig.vm.provision "shell", inline: <<-SHELL
      apt update && apt install -y iptables

      echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
      sysctl -p

      iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
      iptables-save > /etc/iptables.up.rules
    SHELL

    subconfig.vm.provider :virtualbox do |vb|
      vb.name = "RouterFirewall"
      vb.gui = false
      vb.memory = 512
      vb.cpus = 1
    end
  end
end
```

PROF

Pasos a seguir

Levantamos las máquinas

```
vagrant up serverPXE router
```



```
abelsm@asir2-13:~/Documentos/ASO/pxe$ vagrant up
Bringing machine 'serverPXE' up with 'virtualbox' provider...
Bringing machine 'router' up with 'virtualbox' provider...
==> serverPXE: Importing base box 'debian/bullseye64'...
==> serverPXE: Matching MAC address for NAT networking...
==> serverPXE: Checking if box 'debian/bullseye64' version '11.20230615.1' is up to date...
==> serverPXE: A newer version of the box 'debian/bullseye64' for provider 'virtualbox' is
==> serverPXE: available! You currently have version '11.20230615.1'. The latest is version
==> serverPXE: '11.20240212.1'. Run `vagrant box update` to update.
==> serverPXE: Setting the name of the VM: serverPXE
==> serverPXE: Clearing any previously set network interfaces...
==> serverPXE: Preparing network interfaces based on configuration...
    serverPXE: Adapter 1: nat
    serverPXE: Adapter 2: intnet
==> serverPXE: Forwarding ports...
    serverPXE: 22 (guest) => 2222 (host) (adapter 1)
==> serverPXE: Running 'pre-boot' VM customizations...
==> serverPXE: Booting VM...
==> serverPXE: Waiting for machine to boot. This may take a few minutes...
    serverPXE: SSH address: 127.0.0.1:2222
    serverPXE: SSH username: vagrant
    serverPXE: SSH auth method: private key
    serverPXE:
    serverPXE: Vagrant insecure key detected. Vagrant will automatically replace
    serverPXE: this with a newly generated keypair for better security.
```

Nos conectamos a la máquina de servidor

```
vagrant ssh serverPXE
```

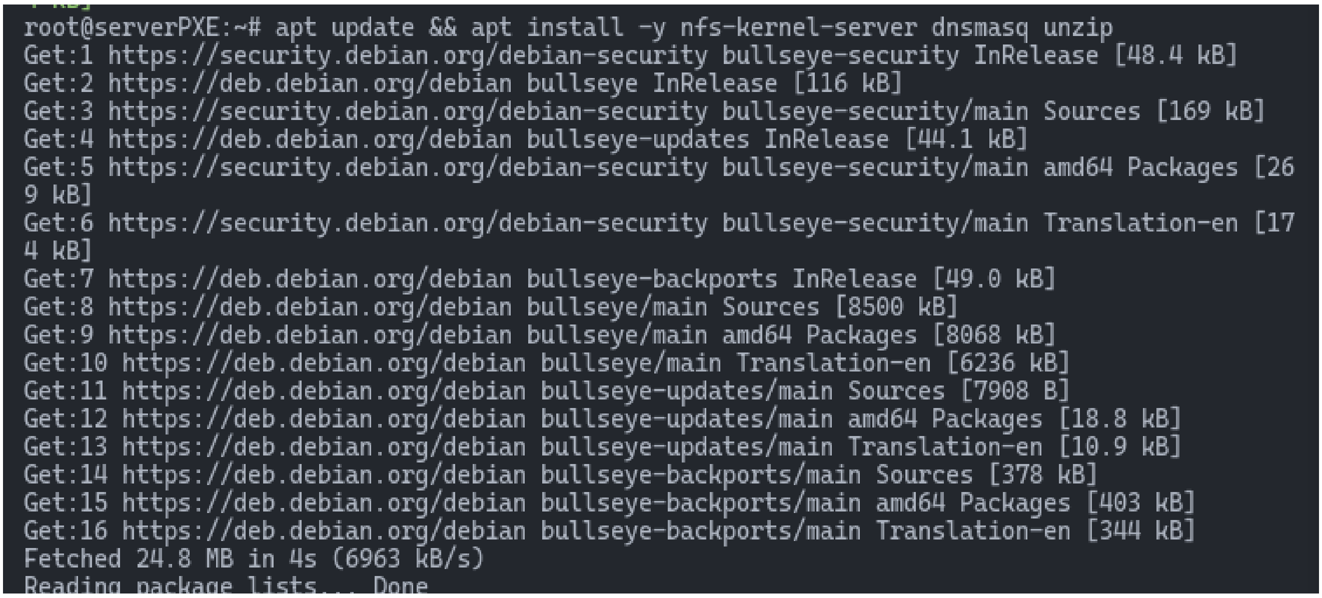
Ejecutamos los siguientes comandos:

```
sudo su -
```

Descargar paquetes

PROF

```
apt update && apt install -y nfs-kernel-server dnsmasq unzip
```



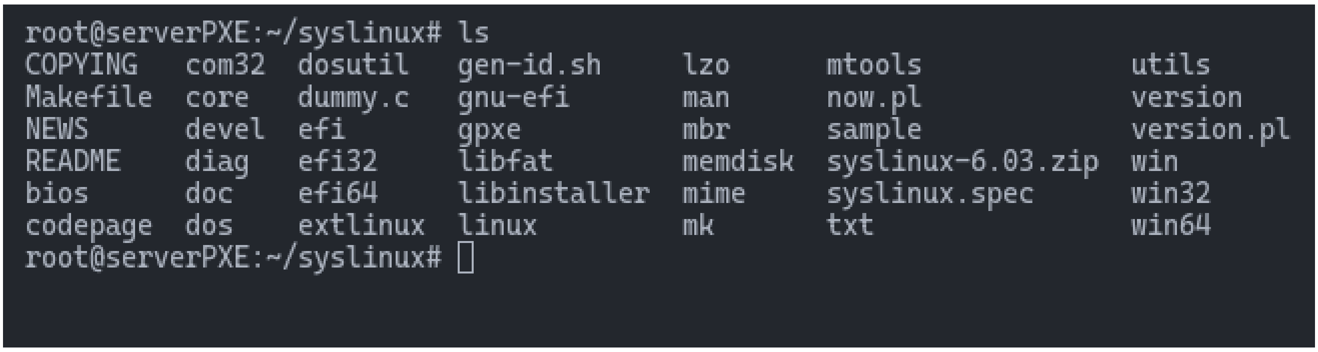
```
root@serverPXE:~# apt update && apt install -y nfs-kernel-server dnsmasq unzip
Get:1 https://security.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:2 https://deb.debian.org/debian bullseye InRelease [116 kB]
Get:3 https://security.debian.org/debian-security bullseye-security/main Sources [169 kB]
Get:4 https://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:5 https://security.debian.org/debian-security bullseye-security/main amd64 Packages [26
9 kB]
Get:6 https://security.debian.org/debian-security bullseye-security/main Translation-en [17
4 kB]
Get:7 https://deb.debian.org/debian bullseye-backports InRelease [49.0 kB]
Get:8 https://deb.debian.org/debian bullseye/main Sources [8500 kB]
Get:9 https://deb.debian.org/debian bullseye/main amd64 Packages [8068 kB]
Get:10 https://deb.debian.org/debian bullseye/main Translation-en [6236 kB]
Get:11 https://deb.debian.org/debian bullseye-updates/main Sources [7908 B]
Get:12 https://deb.debian.org/debian bullseye-updates/main amd64 Packages [18.8 kB]
Get:13 https://deb.debian.org/debian bullseye-updates/main Translation-en [10.9 kB]
Get:14 https://deb.debian.org/debian bullseye-backports/main Sources [378 kB]
Get:15 https://deb.debian.org/debian bullseye-backports/main amd64 Packages [403 kB]
Get:16 https://deb.debian.org/debian bullseye-backports/main Translation-en [344 kB]
Fetched 24.8 MB in 4s (6963 kB/s)
Reading package lists... Done
```

Crear carpeta para archivos de sistema

```
mkdir syslinux && cd syslinux
```

Descargamos los fichero de kernel

```
wget
https://mirrors.edge.kernel.org/pub/linux/utils/boot/syslinux/syslinux-
6.03.zip

unzip syslinux*
```

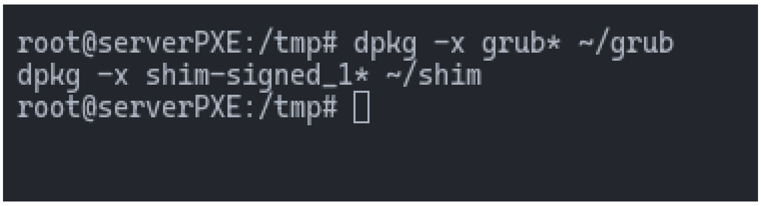


```
root@serverPXE:~/syslinux# ls
COPYING   com32  dosutil   gen-id.sh     lzo      mtools             utils
Makefile  core   dummy.c   gnu-efi       man      now.pl             version
NEWS      devel  efi       gpxe          mbr      sample             version.pl
README    diag   efi32     libfat        memdisk  syslinux-6.03.zip  win
bios      doc    efi64     libinstaller  mime     syslinux.spec      win32
codepage  dos    extlinux  linux         mk       txt                win64
root@serverPXE:~/syslinux# 
```

PROF

Descargamos archivos de grub y de shim

```
cd /tmp
apt-get download shim.signed grub-efi-amd64-signed
dpkg -x grub* ~/grub
dpkg -x shim-signed_1* ~/shim
```



```
root@serverPXE:/tmp# dpkg -x grub* ~/grub
dpkg -x shim-signed_1* ~/shim
root@serverPXE:/tmp# 
```

Creamos el directorio donde se alojarán los archivos del servidor tftp

```
mkdir -p /tftp/{bios,boot,grub}
```

Copiamos los ficheros de configuración

Para poder copiar los archivos de la carpeta /vagrant debemos tenerlos en el mismo directorio que nuestro Vagrantfile

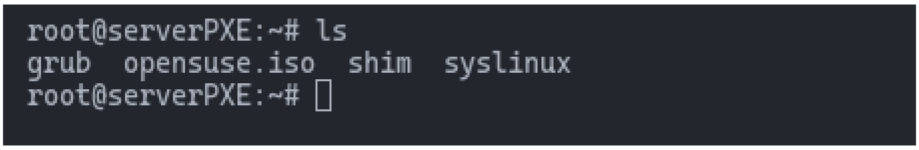
```
cp -v /vagrant/files/exports /etc/exports
systemctl restart nfs-kernel-server

cp -v /vagrant/files/dnsmasq.conf /etc/dnsmasq.conf

cd ~/syslinux

cp -v
bios/{com32/{elflink/ldlinux/ldlinux.c32,libutil/libutil.c32,menu/{menu.
c32,vesamenu.c32}},core/{pxelinux.0,lpxelinux.0}} /tftp/bios

cd ~

cp -v grub/usr/lib/grub/x86_64-efi-signed/grubnetx64.efi.signed
/tftp/grubx64.efi
cp -v shim/usr/lib/shim/shimx64.efi.signed  /tftp/grub/bootx64.efi

cp -v /boot/grub/{grub.cfg,unicode.pf2} /tftp/grub/


sudo ln -s /tftp/boot  /tftp/bios/boot


mkdir /tftp/bios/pxelinux.cfg
cp -v /vagrant/files/default /tftp/bios/pxelinux.cfg/default

cp /vagrant/files/dnsmasq.conf /etc/dnsmasq.conf
systemctl restart dnsmasq
```

PROF

Preparamos la imagen iso para la instalación

Descargamos la imagen iso

```
cd ~

wget https://download.opensuse.org/distribution/leap/15.5/iso/openSUSE-Leap-15.5-DVD-x86_64-Media.iso -O opensuse.iso
```



```
root@serverPXE:~# ls
grub  opensuse.iso  shim  syslinux
root@serverPXE:~# 
```

Creamos las carpetas para alojar la iso

```
mkdir -p /var/www/html/opensuse

mount opensuse.iso /mnt

cp -rfv /mnt/* /var/www/html/opensuse
cp -rfv /mnt/.disk /var/www/html/opensuse

umount /mnt

mkdir -p /tftp/boot/opensuse/loader

cp -rfv /var/www/html/opensuse/boot/x86_64/loader/linux /tftp/boot/opensuse/loader

cp -rfv /var/www/html/opensuse/boot/x86_64/loader/initrd /tftp/boot/opensuse/loader
```

Creamos el fichero default

`nano /tftp/bios/pxelinux.cfg/default`

PROF

```
...
...
...

LABEL OpenSUSE
        kernel /boot/opensuse/loader/linux
        append nfsroot=192.168.1.1:/var/www/html/opensuse netboot=nfs ip=dhcp boot=loader initrd=/boot/opensuse/loader/initrd splash=silent ramdisk_size=512000 ramdisk_blocksize=4096 language=es_ES keytable=es quiet quiet showopts
```

Reiniciamos los servicios

```
systemctl restart dnsmasq
systemctl restart nfs-kernel-server
```

Probamos si funciona

```
vagrant up client
```