

- SEGUNDA PARTE Evaluación Práctica (0.5 Puntos): SCP con Vagrant PUNTAJE

Instale y demuestre el funcionamiento de vagrant-scp como opción de transferencia de archivos desde el anfitrión a una máquina invitada.

Instalación de Vagrant scp

1. Instalamos el plugin de vagrant-scp:

```
juanguzman@juan-g:~/vagrantProject$ vagrant plugin install vagrant-scp
```

2. Hacemos un reload a la máquina:

```
juanguzman@juan-g:~/vagrantProject$ vagrant reload
```

3. Creamos un archivo prueba y lo pasamos por medio de scp de la máquina host al servidor:

```
juanguzman@juan-g:~/vagrantProject$ vagrant scp prueba.txt servidor:~/
```

4. Probamos que el archivo se haya pasado:

```
juanguzman@juan-g:~/vagrantProject$ vagrant ssh servidor
Last login: Tue Mar  1 16:43:40 2022 from 10.0.2.2

[vagrant@servidor ~]$ ls

archivo          archivo_prueba  prueba.txt      prueba3
archivo_cliente  archivo_servidor  prueba2        repoPrueba

[vagrant@servidor ~]$
```

- TERCERA PARTE Evaluación Práctica (0.5 Puntos): Túnel hacia servidor web usando vagrant + ngrok PUNTAJE

Realice un túnel hacia el servidor web implementado en clase, de manera que los recursos de su servidor web puedan ser visualizados desde cualquier lugar por fuera de su red local. Para efectos de prueba, agregar una página personalizada en su sitio web.

Se sugiere usar:

- Port forwarding en Vagrant
- Vagrant share
- ngrok (agregarlo al path --variables de entorno-- una vez instalado)

Ajuste VagrantFile

```
servidor.vm.network "forwarded_port", guest: 80, host: 1234
```

Configuración de Ngrok

1. Instalamos ngrok

```
snap install ngrok
```

2. Agregamos en token

```
ngrok authtoken 25kpWqF5UgubQeAl2dUH8p3lzyV_22GK1faHVDqrkwwjcMLnq
```

Configuración vagrant-share

1. Instalamos el plugin de vagrant-share

```
juanguzman@juan-g:~/vagrantProject$ vagrant plugin install vagrant-share
```

2. Hacemos un reload del servidor

```
juanguzman@juan-g:~/vagrantProject$ vagrant reload
```

3. Ejecutamos el comando para el puente

```
juanguzman@juan-g:~/vagrantProject$ vagrant share servidor
==> servidor: Detecting network information for machine...
servidor: Local machine address: 127.0.0.1
servidor:
servidor: Note: With the local address (127.0.0.1), Vagrant Share can only
servidor: share any ports you have forwarded. Assign an IP or address to your
servidor: machine to expose all TCP ports. Consult the documentation
servidor: for your provider ('virtualbox') for more information.
servidor:
servidor: Local HTTP port: 1234
servidor: Local HTTPS port: disabled
servidor: Port: 2222
servidor: Port: 1234
==> servidor: Creating Vagrant Share session...
==> servidor: HTTP URL: http://5d15-2800-e2-4b80-56e-e9b2-b113-628f-8bca.ngrok.io
```

- CUARTA PARTE Evaluación Práctica (2.0 Puntos): Implementación de un Servidor PXE PUNTAJE
Implementación de un servidor PXE

Un servidor PXE (Preboot eXecution Environment) permite la instalación automatizada de un sistema operativo a través de la red. Los clientes que usan PXE no necesitan una unidad de arranque para iniciar el sistema operativo, ya que este se inicia a través de un ISO disponible en la red. Una vez configurado un servidor PXE, se pueden instalar cientos de sistemas al mismo tiempo a través de la red.

Este sistema funciona basado en la arquitectura cliente/servidor. Para el correcto funcionamiento de un servidor PXE, se deben instalar en el servidor servicios como dhcp y tftp. Para instalar el sistema operativo en los clientes, estos se deben iniciar usando la opción PXE de booteo.

Requerimiento

1. Configure un servicio PXE como servidor de imágenes Linux
2. Configure un cliente para iniciar en red, a través de la opción PXE de booteo y usando el servidor PXE configurado.

1. Creamos una máquina nueva por medio de vagrant init y editamos el VagrantFile

```
Vagrant.configure("2") do |config|
  if Vagrant.has_plugin? "vagrant-vbguest"
    config.vbguest.no_install = true
    config.vbguest.auto_update = false
    config.vbguest.no_remote = true
  end
  config.vm.define :servidor do |servidor|
    servidor.vm.box = "bento/centos-7.9"
    servidor.vm.network :private_network, ip: "192.168.30.11",
virtualbox__intnet: "lan2"
    servidor.vm.hostname = "servidor"
  end
end
```

2. Instalamos las siguientes servicios:

```
[root@servidor ~]# yum install dhcp tftp tftp-server syslinux vsftpd xinetd
```

3. Configuramos el servidor DCHP para PXE: Cuando instalamos el paquete dhcp, se crea un archivo de configuración de muestra del servidor dhcp en "/usr/share/doc/dhcp*/dhcpd.conf.example", aunque el archivo de configuración de dhcp está en "/etc/dhcp/dhcpd.conf".

```
[root@servidor ~]# vim /etc/dhcp/dhcpd.conf
ddns-update-style interim;
ignore client-updates;
authoritative;
allow booting;
allow bootp;
```

```
allow unknown-clients;

subnet 192.168.30.0 netmask 255.255.255.0 {
    range 192.168.30.21 192.168.30.151;
    option domain-name-servers 192.168.30.11;
    option domain-name "pxe.example.com";
    option routers 192.168.30.11;
    option broadcast-address 192.168.30.255;
    default-lease-time 600;
    max-lease-time 7200;
    IP del PXE Server
    next-server 192.168.30.11;
    filename "pxelinux.0";
}
```

4. Editamos y configuramos el servidor TFTP: TFTP (Protocolo trivial de transferencia de archivos) se utiliza para transferir archivos desde el servidor de datos a sus clientes sin ningún tipo de autenticación. En el caso de la configuración del servidor PXE, se utiliza tftp para la carga de arranque. Para configurar servidor tftp, edite su archivo de configuración ‘/etc/xinetd.d/tftp’, cambie el parámetro ‘disable = yes’ a ‘disable = no’ y deja los demás parámetros como están.

```
[root@servidor ~]# vim /etc/xinetd.d/tftp

service tftp
{
    socket_type = dgram
    protocol = udp
    wait = yes
    user = root
    server = /usr/sbin/in.tftpd
    server_args = -s /var/lib/tftpboot
    disable = no
    per_source = 11
    cps = 100 2
    flags = IPv4
}
```

Todos los archivos relacionados con el arranque de red deben colocarse en el directorio raíz de tftp “/var/lib/tftpboot”. Ejecute los siguientes comandos para copiar los archivos de arranque de red requeridos en ‘/var/lib/tftpboot’.

```
[root@servidor ~]# cp -v /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot
[root@servidor ~]# cp -v /usr/share/syslinux/menu.c32 /var/lib/tftpboot
[root@servidor ~]# cp -v /usr/share/syslinux/memdisk /var/lib/tftpboot
[root@servidor ~]# cp -v /usr/share/syslinux/mboot.c32 /var/lib/tftpboot
[root@servidor ~]# cp -v /usr/share/syslinux/chain.c32 /var/lib/tftpboot
[root@servidor ~]# mkdir /var/lib/tftpboot/pxelinux.cfg
```

```
[root@servidor ~]# mkdir /var/lib/tftpboot/networkboot
```

5. Montamos el archivo ISO CentOS 7.x lo pasamos de la máquina host al servidor por vagrant scp: Corre por debajo comandos para montar el archivo iso y luego copiar su contenido en el directorio del servidor ftp '/var/ftp/pub'.

```
juanguzman@juan-g:~/parcialPxe$ vagrant scp CentOS-7-x86_64-DVD-2009.iso  
servidor:~/
```

```
[root@servidor ~]# mount -o loop CentOS-7-x86_64-DVD-2009.iso /mnt/  
mount: /dev/loop0 is write-protected, mounting read-only  
[root@servidor ~]# cd /mnt/  
[root@servidor ~ mnt]# cp -av * /var/ftp/pub/
```

Copie el archivo Kernel (vmlinuz) y el archivo initrd del archivo iso montado a '/var/lib/tftpboot/networkboot/'.

```
[root@servidor ~]# cp /mnt/images/pxeboot/vmlinuz /var/lib/tftpboot/networkboot/  
[root@servidor ~]# cp /mnt/images/pxeboot/initrd.img  
/var/lib/tftpboot/networkboot/
```

Desmontamos el archivo iso

```
[root@servidor ~]# umount /mnt/
```

6. Cree el archivo de menú kickStart y PXE. Antes de crear el archivo kickstart, primero creamos la contraseña de root en una cadena cifrada porque usaremos esa cadena de contraseña cifrada en el archivo kickstart.

```
[root@servidor ~]# openssl passwd -1
```

El archivo kickstart predeterminado del sistema se coloca en /root con el nombre 'anaconda-ks.cfg'. Crearemos un nuevo kickstart en la carpeta /var/ftp/pub con el nombre 'centos7.cfg' Copie el siguiente contenido en el nuevo archivo kickstart. Modifique el archivo kickstart según sus necesidades.

```
[root@servidor ~]# vim /var/ftp/pub/centos7.cfg  
#platform=x86, AMD64, or Intel EM64T  
#version=DEVEL  
# Firewall configuration  
firewall --disabled  
# Install OS instead of upgrade  
install  
# Use FTP installation media  
url --url="ftp://192.168.30.11/pub/"  
# Root password  
rootpw --iscrypted $1$e2wrcGGX$tZPQKPsXVhNmbiGg53MN41  
# System authorization information  
auth useshadow passalgo=sha512  
# Use graphical install  
graphical  
firstboot disable
```

```
# System keyboard
https://www.linuxtechi.com/configure-pxe-installation-server-centos-7/
4/2831/8/2021
Configure PXE (Network Boot) installation Server on CentOS 7.x
keyboard us
# System language
lang en_US
# SELinux configuration
selinux disabled
# Installation logging level
logging level=info
# System timezone
timezone Europe/Amsterdam
# System bootloader configuration
bootloader location=mbr
clearpart --all --initlabel
part swap --asprimary --fstype="swap" --size=1024
part /boot --fstype xfs --size=300
part pv.01 --size=1 --grow
volgroup root_vg01 pv.01
logvol / --fstype xfs --name=lv_01 --vgname=root_vg01 --size=1 --grow
%packages
@^minimal
@core
%end
%addon com_redhat_kdump --disable --reserve-mb='auto'
%end
```

Cree un archivo de menú PXE (/var/lib/tftpboot/pxelinux.cfg/default), copie los siguientes contenidos en el archivo de menú pxe.

```
[root@servidor ~]# vim /var/lib/tftpboot/pxelinux.cfg/default
default menu.c32
prompt 0
timeout 30
MENU TITLE LinuxTechi.com PXE Menu
LABEL centos7_x64
MENU LABEL CentOS 7 X64
KERNEL /networkboot/vmlinuz
APPEND initrd=/networkboot/initrd.img inst.repo=ftp://192.168.30.11/pub
ks=ftp://192.168.30.11/pub/centos7.cfg
```

7. Inicie y habilite el servicio xinetd, dhcp, vsftpd y firewallld. Utilice los siguientes comandos para iniciar y habilitar xinetd, dhcp, vsftpd y firewallld.

```
[root@servidor ~]# systemctl start xinetd
[root@servidor ~]# systemctl enable xinetd
[root@servidor ~]# systemctl start dhcpd.service
[root@servidor ~]# systemctl enable dhcpd.service
Created symlink from /etc/systemd/system/m
[root@servidor ~]# systemctl start vsftpd
[root@servidor ~]# systemctl enable vsftpd
Created symlink from /etc/systemd/system/multi-user.target.wants/vsftpd.service to
/usr/lib/systemd/system/vsftpd.service.
[root@servidor ~]# systemctl start firewallld
[root@servidor ~]# systemctl enable firewallld
```

En caso de que SELinux esté habilitado, configure la siguiente regla de selinux para el servidor ftp.

```
[root@servidor ~]# setsebool -P allow_ftp_full_access 1
```