

2023/2024

Instalación y configuración del laboratorio virtual

Jesús Lorenzo Limon

<i>Configurar las redes del laboratorio virtual</i>	2
Adaptadores de red de las MVs:	2
Configuración de los nombres de las maquinas	2
Netplans y archivo .yaml	2
Configurar enrutamiento	3
Configurar DNS	4
Configurar nombres y dominios	5
Configurar SSH en router	6
Configurar SSH en servidor	6
Configurar APACHE	6

Configurar las redes del laboratorio virtual

Importante: *toda la configuración aquí albergada esta cogiendo de referencia una dirección personal, en caso de querer utilizar otra u tener otra deberás buscar la tuya propia para poder configurarlo a gusto.*

Adaptadores de red de las MVs:

En el **router**, instalación de 3 adaptadores virtuales de red

- 1) El externo, en modo puente.
- 2) Dos en modo red interno:
 - a. **emp**, que representa la red de empleados de la empresa
 - b. **dmz**, que representa la red de **servidores** de la empresa o DMZ (Demilitarized Zone).

En el **cliente**, un adaptador en modo red interna. La red se llama **emp**.

En el **servidor**, un adaptador en modo red interna. La red se llama **dmz**.

Configuración de los nombres de las maquinas

Modificamos el archivo encontrado en `/etc/hostname` en cada máquina. (Anotación, el cambio también deberíamos hacerlo en el archivo con ruta `/etc/hosts`)

Netplans y archivo .yaml

Empezando por el **router**, entramos en `/etc/netplans/` y modificamos el archivo `00-network-manager-all.yaml` y empezamos a configurar la red de la siguiente forma:

ROUTER

Empezando por el **router**, entramos en `/etc/netplans/` y modificamos el archivo `00-network-manager-all.yaml` y empezamos a configurar la red de la siguiente forma:

#Let NetworkManager manage all devices on system

network:

version: 2

renderer: NetworkManager

etehrnets:

enp0s3:

dhcp: no

addresses:

- 172.28.5.1/16 *#en este caso esta red pertenece a la red de la maquina real*

routes:

- to: default

via: 172.28.0.5

enp0s8:

addresses:

- 192.168.1.1/24 *#red emp*

enp0s9:

addresses:

- 10.0.1.254/16 #red dmz

CLIENTE

En el caso del **cliente y el servidor**, entramos en `/etc/netplans/` y modificamos el archivo `01-network-manager-all.yaml` y empezamos a configurar la red de la siguiente forma:

#Let NetworkManager manage all devices on system

network:

version: 2

renderer: NetworkManager

eternets:

enp0s3:

dhcp: no

addresses:

- 192.168.1.10/24 *#en este caso esta red pertenece a la red de la maquina real*

routes:

- to: default

via: 192.168.1.1

SERVIDOR

#Let NetworkManager manage all devices on system

network:

version: 2

renderer: NetworkManager

eternets:

enp0s3:

dhcp: no

addresses:

- 10.0.1.1/16 *#en este caso esta red pertenece a la red de la maquina real*

routes:

- to: default

via: 10.0.1.254

una vez cambiado el archivo `.yaml` de cada máquina ejecutamos el comando `netplan apply` para que la red se actualice en las máquinas y usando `ip a` verificamos que los cambios se hayan llevado a cabo

En principio todos los dispositivos deberían poder hacerse ping entre ellos ya que las peticiones llegan al **router** y sabe resolver las peticiones.

Configurar enrutamiento

Para configurar el **router** para que enrute tenemos que seguir los siguientes pasos:

1º Cambiamos el archivo `/etc/sysctl.conf`, y des comentamos la línea que ponga `net.ipv4.ip_forward=1` (existen alguna forma más pero no son persistentes y solo sirven para la sesión actual por lo que si reiniciamos volvería al valor default).

2º creamos los siguientes archivos:

`nano /etc/iptables/iptables.reglas`

`iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE`

```
nano /etc/iptables/iptables.reglas
```

```
#!/bin/sh
```

```
iptables -t filter -F
```

```
iptables -t nat -F
```

```
iptables -X iptables -Z
```

```
iptables -P INPUT ACCEPT
```

```
iptables -P FORWARD ACCEPT
```

```
iptables -P OUTPUT ACCEPT
```

```
exit 0
```

les damos permisos de ejecución con `chmod +x /etc/iptables/iptables.re*`
y ahora creamos el servicio cortafuegos

```
nano /etc/systemd/system/cortafuegos.service
```

```
[Unit]
```

```
Description=Packet Filtering Framework
```

```
[Service]
```

```
Type=oneshot
```

```
ExecStart=/etc/iptables/iptables.reglas
```

```
ExecReload=/etc/iptables/iptables.reglas
```

```
ExecStop=/etc/iptables/iptables.reseteo
```

```
RemainAfterExit=yes
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Una vez realizado estos pasos lanzamos los siguientes comandos:

```
systemctl enable cortafuegos // para que se inicia cuando se inicia el dispositivo
```

```
systemctl start cortafuegos // para iniciarlo en la sesión actual
```

```
systemctl status cortafuegos // para comprobar que este activo
```

una vez hecho esto nos va a resolver 3 de las 5 pruebas propuestas, las 2 que no va a saber resolver son **cliente** -> www.google.com y **servidor** -> www.google.com ya que no tenemos ningún DNS configurado aún

Configurar DNS

Para configurar los DNS añadimos lo siguiente al archivo que ya modificamos en su momento, el `.yam` Importante, solo en cliente y servidor ya que el router tiene en su tarjeta de red configurado el default del instituto

CLIENTE

```
#Let NetworkManager manage all devices on system
```

```
network:
```

```
version: 2
```

```
renderer: NetworkManager
```

```
etehrnets:
```

```
enp0s3:
```

```
dhcp: no
```

```
addresses:
```

- 192.168.1.10/24 *#en este caso esta red pertenece a la red de la maquina real*

routes:

- to: default
- via: 192.168.1.1

nameservers:

addresses:

- 192.168.1.1 # Router propio
- 172.28.0.5 # Instituto
- 8.8.8.8 # Google

SERVIDOR

#Let NetworkManager manage all devices on system

network:

version: 2

renderer: NetworkManager

etehrnets:

enp0s3:

dhcp: no

addresses:

- 10.0.1.1/16 *#en este caso esta red pertenece a la red de la maquina real*

routes:

- to: default

via: 10.0.1.254

nameservers:

addresses:

- 10.0.1.254 # Router propio

- 172.28.0.5 # Instituto

- 8.8.8.8 # Google

Una vez hecho esto hacemos *netplan apply* y ya podemos realizar todas las pruebas que se nos proponen, tanto el ping a Google con ambas maquinas, como las actualizaciones

Configurar nombres y dominios

En el archivo */etc/hosts* añadimos u modificamos lo siguiente

ROUTER

127.0.1.1	router
192.168.1.10	cliente
10.0.1.1	servidor

127.0.1.1	router.iesclaradeltey.org
192.168.1.10	cliente.iesclaradeltey.org
10.0.1.1	servidor.iesclaradeltey.org

CLIENTE

127.0.1.1	cliente
192.168.1.10	router
10.0.1.1	servidor

127.0.1.1	cliente.iesclaradeltey.org
192.168.1.1	router.iesclaradeltey.org
10.0.1.1	servidor.iesclaradeltey.org

SERVIDOR

127.0.1.1	servidor
-----------	----------

192.168.1.10	cliente
10.0.1.254	router
127.0.1.1	servidor.iesclaradeltey.org
192.168.1.10	cliente.iesclaradeltey.org
10.0.1.254	router.iesclaradeltey.org

Ya deberían de funcionar todas las conexiones entre si con los nuevos nombres.

Configurar SSH en router

La instalación de ssh seria *apt install openssh-server*

Una vez instalado modificamos 2 archivos, el primero sería */etc/hosts.deny* y añadimos la línea *sshd: ALL* y el archivo */etc/hosts.allow* al cual añadimos la línea *sshd: 172.28.254.145* (esta dirección será personal para cada maquina real)

Hacemos ssh alumno@172.28.5.1 para probar la conexión desde la maquina real

Configurar SSH en servidor

La instalación de ssh seria *apt install openssh-server*

Con *htop* vemos las conexiones y buscamos ssh para ver que el puerto que se nos habilita para esa conexión es la 22

Configurar APACHE

Instalamos apache con *apt install apache2*

En la maquina real entramos en los adaptadores de red y añadimos una ruta en ipv4 poniendo que cada vez que se haga una petición con una red en *10.0.0.0* con mascara de red *255.255.0.0* se la envíe a la maquina con ip *172.28.5.1* para que se lo solucione y listo ya podemos hacer las pruebas pertinentes, cono dato si buscamos <http://servidor> no nos va a saber solucionar el nombre porq en la maquina real no tenemos ese nombre dentro del DNS, pero si ponemos <http://10.0.1.1:80> si que nos aparecerá el servidor