Configuración de red - Servidor QEMU/KVM

July 20, 2022

```
root@qemu:~# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP mode DEFAULT group default qlen 1000
    link/ether [redacted] ff:ff:ff:ff:ff:ff
    altname enp0s25
3: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
4: dmz@br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
5: lan1@br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
6: lan2@br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
9: vnet2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br0 state UNKNOWN mode DEFAULT group default qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
root@qemu:~#
```

HABILITAR IP-FORWARDING

Para permitir que los paquetes pasen por medio de las interfaces virtuales.

- */etc/sysctl.conf*

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Iproute2

La herramienta **IP** es la herramienta principal de **iproute2**, con ella se pueden ver y configurar direcciones ip, tablas de enrutamiento, túneles e interfaces.

Haciendo uso de **IP** Creare tres interfaces, junto con sus configuraciones. Estas interfaces estaran asociadas a la interfaz virtual **br0** que funciona como puente entre la interfaz fisica.

```
#! /bin/bash
echo "Creacion de las interfaces"

ip link add link br0 name dmz type vlan id 100
ip addr add 192.168.100.1/24 brd 192.168.100.255 dev dmz
ip link set dmz up

ip link add link br0 name lan1 type vlan id 101
ip addr add 192.168.101.1/24 brd 192.168.101.1 dev lan1
ip link set lan1 up

ip link add link br0 name lan2 type vlan id 102
ip addr add 192.168.102.1/24 brd 192.168.102.255 dev lan2
ip link set lan2 up

echo "Fin de las interfaces"
```

Este script en bash lo guardare dentro de un directorio y con **crontab** lo ejecutare cada vez que se inicie el sistema operativo.

IPTABLES

Iptables es un módulo del núcleo de Linux que se encarga de filtrar los paquetes de red. O sea, se encarga de determinar qué paquetes se aceptan y cuales no.

Como cualquier firewall, iptables funciona a través de reglas. Estas reglas deben especificar que hacer con cada paquete, que puertos deben recibiresos paquetes, el protocolo utilizado para el envío de datos y cualquier otra información relacionada con el intercambio de datos entre redes.

Configuracón de iptables

Una vez finalizadas las configuraciones de red definire una series de reglas en **IPTABLES** las cuales me serviran para poder acceder a los servicios desde internet, proteger las distintas redes y darle salida a internet a los host de cada red.

```
#! /bin/bash

#############################################################################################################
#SERVIDOR DE VIRTUALIZACION
#br0 => red local => 192.198.1.222
#dmz => vlan => dentro del servidor => 192.168.100.0/24
#lan1 => vlan => dentro del servidor => 192.168.101.0/24
#lan2 => vlan => dentro del servidor => 192.168.102.0/24

echo "Comienzo de las Reglas"

#############################################################################################################
#VARIABLES
MAC_PC1=""
MAC_NET=""
MAC_CEL=""

#############################################################################################################
#LIMPIAR REGLAS
/usr/sbin/iptables -F
/usr/sbin/iptables -X
/usr/sbin/iptables -Z
/usr/sbin/iptables -t nat -F

#############################################################################################################
#POLITICAS POR DEFECTO
/usr/sbin/iptables -P INPUT DROP
/usr/sbin/iptables -P FORWARD DROP
/usr/sbin/iptables -P OUTPUT DROP

#############################################################################################################
#LO QUE LLEGUE DE INTERNET
/usr/sbin/iptables -t nat -A PREROUTING -i br0 -p tcp --dport 22 -j DNAT --to 192.168.100.2:2222
/usr/sbin/iptables -t nat -A PREROUTING -i br0 -p tcp --dport 8096 -j DNAT --to 192.168.100.2:8096
/usr/sbin/iptables -t nat -A PREROUTING -i br0 -p tcp --dport 8080 -j DNAT --to 192.168.100.2:8080
/usr/sbin/iptables -t nat -A PREROUTING -i br0 -p tcp --dport 8765 -j DNAT --to 192.168.100.2:8765

#############################################################################################################
#RESPUESTA A LAS COMUNICACIONES YA ESTABLECIDAS
/usr/sbin/iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
/usr/sbin/iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
/usr/sbin/iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

#############################################################################################################
#PERMITEN LOS PING
/usr/sbin/iptables -A INPUT -i dmz -p icmp --icmp-type echo-request -j ACCEPT #PING
/usr/sbin/iptables -A INPUT -i lan1 -p icmp --icmp-type echo-request -j ACCEPT #PING
/usr/sbin/iptables -A INPUT -i lan2 -p icmp --icmp-type echo-request -j ACCEPT #PING
/usr/sbin/iptables -A OUTPUT -o dmz -p icmp --icmp-type echo-request -j ACCEPT #PING
/usr/sbin/iptables -A OUTPUT -o lan1 -p icmp --icmp-type echo-request -j ACCEPT #PING
/usr/sbin/iptables -A OUTPUT -o lan2 -p icmp --icmp-type echo-request -j ACCEPT #PING
#/usr/sbin/iptables -A FORWARD -i dmz -o br0 -p icmp --icmp-type echo-request -j ACCEPT
#/usr/sbin/iptables -A FORWARD -i lan1 -o br0 -p icmp --icmp-type echo-request -j ACCEPT
```

```
#/usr/sbin/iptables -A FORWARD -i lan2 -o br0 -p icmp --icmp-type echo-request -j ACCEPT

#################################################################################################
#ADMINISTRAR FIREWALL DESDE MI RED LAN (NETBOOK,CELULAR,PC DESKTOP)
/usr/sbin/iptables -A INPUT -i br0 -m mac --mac-source $MAC_CEL -p tcp --dport 2222 -j ACCEPT #CELULAR
/usr/sbin/iptables -A INPUT -i br0 -m mac --mac-source $MAC_PC1 -p tcp --dport 2222 -j ACCEPT #MI PC
/usr/sbin/iptables -A INPUT -i br0 -m mac --mac-source $MAC_PC1 -p tcp --dport 5900:5920 -j ACCEPT #SPICE
/usr/sbin/iptables -A INPUT -i br0 -m mac --mac-source $MAC_NET -p tcp --dport 2222 -j ACCEPT #NET
/usr/sbin/iptables -A INPUT -i br0 -m mac --mac-source $MAC_NET -p tcp --dport 5900:5920 -j ACCEPT #SPICE

#################################################################################################
#NFS => SOLO ES PRACTICA
/usr/sbin/iptables -A INPUT -s 192.168.100.2 -p tcp --dport 2049 -j ACCEPT #NFS
/usr/sbin/iptables -A INPUT -s 192.168.100.2 -p udp --dport 2049 -j ACCEPT #NFS

#################################################################################################
#CONEXION SSH DESDE EL FIREWALL A LAS DEMAS REDES
/usr/sbin/iptables -A OUTPUT -o dmz -p tcp --dport 2222 -j ACCEPT #SSH
/usr/sbin/iptables -A OUTPUT -o lan1 -p tcp --dport 2222 -j ACCEPT #SSH
/usr/sbin/iptables -A OUTPUT -o lan2 -p tcp --dport 2222 -j ACCEPT #SSH
/usr/sbin/iptables -A OUTPUT -o br0 -p tcp --dport 2222 -j ACCEPT #SSH

#################################################################################################
#PUERTOS QUE USA EL FIREWALL
/usr/sbin/iptables -A OUTPUT -o br0 -p tcp --dport 2222 -j ACCEPT #SSH
/usr/sbin/iptables -A OUTPUT -o br0 -p tcp --dport 80 -j ACCEPT #HTTP
/usr/sbin/iptables -A OUTPUT -o br0 -p tcp --dport 443 -j ACCEPT #HTTPS
/usr/sbin/iptables -A OUTPUT -o br0 -p tcp --dport 53 -j ACCEPT #DNS TCP
/usr/sbin/iptables -A OUTPUT -o br0 -p udp --dport 53 -j ACCEPT #DNS UDP

#################################################################################################
#PUERTOS POR LOS QUE SE PODRA SALIR DE LA RED
#dmz
/usr/sbin/iptables -A FORWARD -i dmz -o br0 -p tcp --dport 80 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i dmz -o br0 -p tcp --dport 443 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i dmz -o br0 -p tcp --dport 53 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i dmz -o br0 -p udp --dport 53 -j ACCEPT
#lan1
/usr/sbin/iptables -A FORWARD -i lan1 -o br0 -p tcp --dport 80 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i lan1 -o br0 -p tcp --dport 443 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i lan1 -o br0 -p tcp --dport 53 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i lan1 -o br0 -p udp --dport 53 -j ACCEPT
#lan2
/usr/sbin/iptables -A FORWARD -i lan2 -o br0 -p tcp --dport 80 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i lan2 -o br0 -p tcp --dport 443 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i lan2 -o br0 -p tcp --dport 53 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i lan2 -o br0 -p udp --dport 53 -j ACCEPT

#################################################################################################
#LAS REDIRECCIONES QUE REALIZARA EL FIREWALL A LOS SERVIDORES
/usr/sbin/iptables -A FORWARD -i br0 -o dmz -p tcp --dport 8096 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i br0 -o dmz -p tcp --dport 8080 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i br0 -o dmz -p tcp --dport 8765 -j ACCEPT
/usr/sbin/iptables -A FORWARD -i br0 -o dmz -p tcp --dport 2222 -j ACCEPT

#################################################################################################
#EL TRAFICO QUE SE ORIGINE EN LAS SIGUIENTES REDES SE ENMASCARA
/usr/sbin/iptables -t nat -A POSTROUTING -s 192.168.100.0/24 -j MASQUERADE
/usr/sbin/iptables -t nat -A POSTROUTING -s 192.168.101.0/24 -j MASQUERADE
/usr/sbin/iptables -t nat -A POSTROUTING -s 192.168.102.0/24 -j MASQUERADE
```

Este script en bash lo guardare dentro del mismo directorio donde guarde el script con las configuraciones de red y de la misma manera que el script anterior lo ejecutare en cada incio del sistema con **crontab**.

Link de referecnias:

- RedHat (https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-iptables.html)
- Manual Practico (http://redesdecomputadores.umh.es/iptables.htm)

DMZ

Para mi primer servidor que estara dentro de la DMZ le configurare la interfaz de red de la misma manera que lo hice con el servidor QEMU/KVM.

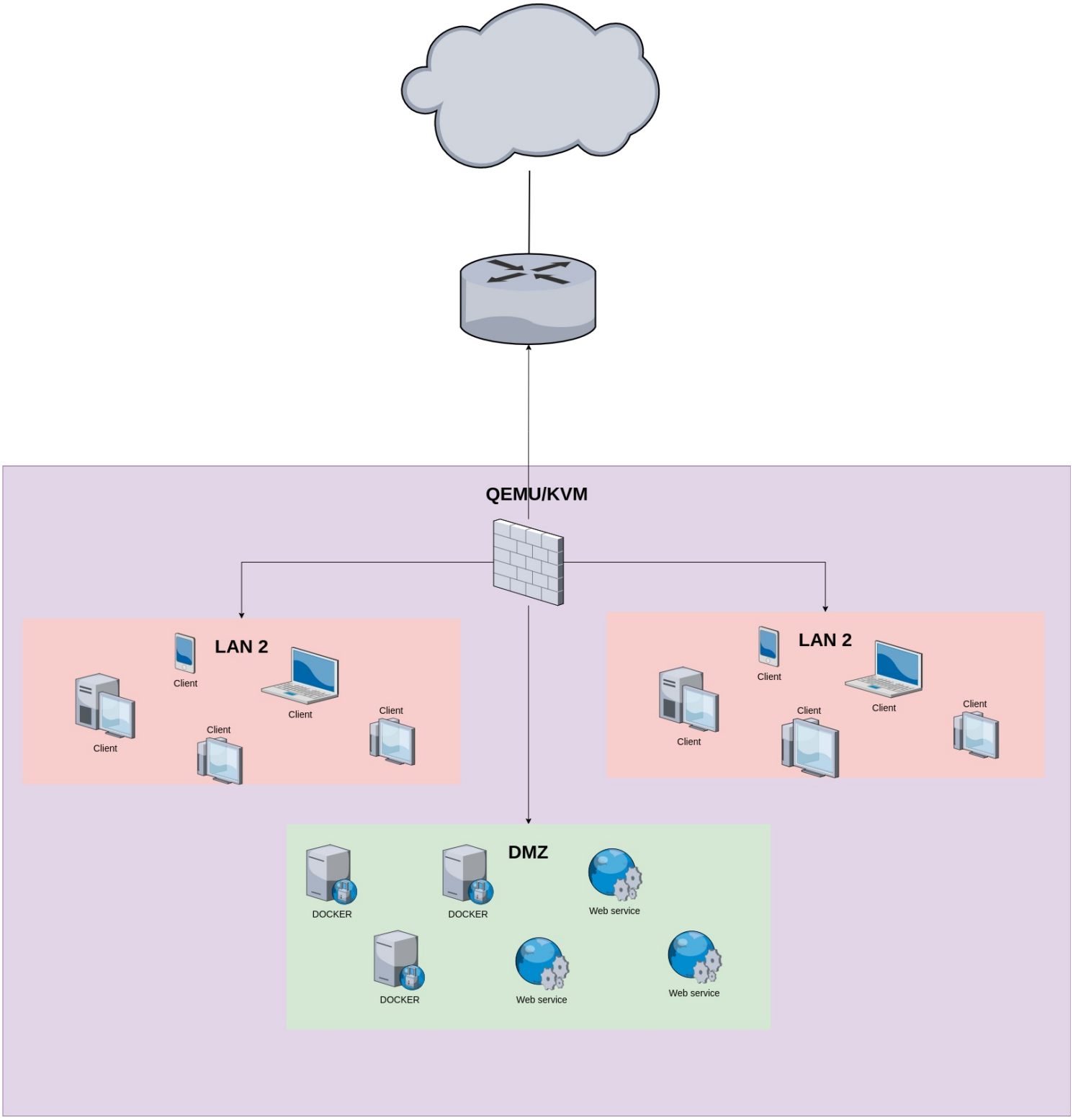
```
#! /bin/bash

ip link add link enp1s0 name dmz type vlan id 100 #creo la vlan
ip addr add 192.168.100.2/24 brd 192.168.100.255 dev dmz #configuro la ip
ip link set dmz up #activo la interfaz
ip route add default via 192.168.100.1 #nueva ruta de ruteo
echo "nameserver 1.1.1.1" > /etc/resolv.conf # dns
ip addr del 192.168.1.124/24 dev enp1s0 #elimino la ip estatica anterio
```

Es necesario que este script se ejecute en cada inicio del sistema. Se puede ejecutar desde las configuraciones de red dentro del archivo */etc/network/interfaces*, pero para mi gusto lo ejecutare desde **crontab**.

Diagrama de RED

El servidor virtualizacion emulara tres deres virtuales y los equipos de cada red.



Tags: Configuracón de Red

Categories: Debian dmz ip iptables vlan

Updated: July 20, 2022