

Universidad Nacional Autónoma de México
Facultad de Ciencias

Asignatura: Redes de computadoras
Semestre: 2024-1

Profesor: Javier León Cotonieto

Ayudantes: Magdalena Reyes Granados
Itzel Gómez Muñoz
Sandra Plata Velázquez

Practica 1. “Máquinas virtuales Linux y modos de configuración de NIC”

Equipo 5

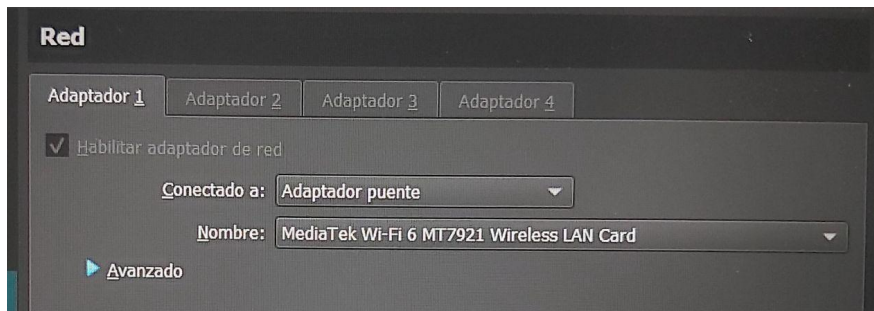
Integrantes:

- **Almanza Torres José Luis**
- **Jimenez Reyes Abraham**
- **Martínez Pardo Esaú**



Modos de configuración de la NIC

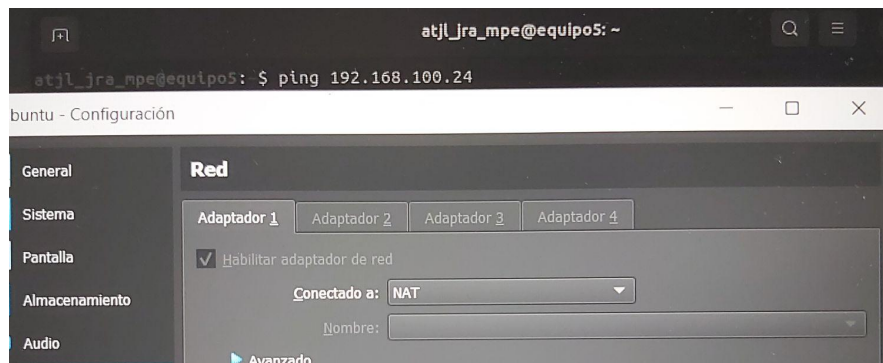
Modo Bridge.



```
atjl_jra_mpe@equipo5: ~  
atjl_jra_mpe@equipo5:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.100.27 netmask 255.255.255.0 broadcast 192.168.100.255  
inet6 fe80::8f7c:a124:7017:cf14 prefixlen 64 scopeid 0x20<link>  
inet6 2806:2f0:91a1:ae5b:3fff:a9ea:7d3d:3114 prefixlen 64 scopeid 0x0<global>  
inet6 2806:2f0:91a1:ae5b:9ae:d07f:266f:a55d prefixlen 64 scopeid 0x0<global>  
ether 08:00:27:1a:9e:fc txqueuelen 1000 (Ethernet)  
RX packets 245 bytes 27828 (27.8 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 143 bytes 18738 (18.7 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Bucle local)  
RX packets 119 bytes 9959 (9.9 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 119 bytes 9959 (9.9 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
atjl_jra_mpe@equipo5:~$
```

```
atjl_jra_mpe@equipo5: ~  
atjl_jra_mpe@equipo5:~$ ping 192.168.100.24  
PING 192.168.100.24 (192.168.100.24) 56(84) bytes of data.  
64 bytes from 192.168.100.24: icmp_seq=1 ttl=128 time=1.50 ms  
64 bytes from 192.168.100.24: icmp_seq=2 ttl=128 time=0.637 ms  
64 bytes from 192.168.100.24: icmp_seq=3 ttl=128 time=0.696 ms  
64 bytes from 192.168.100.24: icmp_seq=4 ttl=128 time=0.400 ms  
64 bytes from 192.168.100.24: icmp_seq=5 ttl=128 time=0.641 ms  
64 bytes from 192.168.100.24: icmp_seq=6 ttl=128 time=0.543 ms  
64 bytes from 192.168.100.24: icmp_seq=7 ttl=128 time=0.760 ms  
64 bytes from 192.168.100.24: icmp_seq=8 ttl=128 time=0.288 ms  
64 bytes from 192.168.100.24: icmp_seq=9 ttl=128 time=0.602 ms  
64 bytes from 192.168.100.24: icmp_seq=10 ttl=128 time=0.641 ms  
64 bytes from 192.168.100.24: icmp_seq=11 ttl=128 time=0.983 ms  
64 bytes from 192.168.100.24: icmp_seq=12 ttl=128 time=0.417 ms  
64 bytes from 192.168.100.24: icmp_seq=13 ttl=128 time=0.650 ms  
64 bytes from 192.168.100.24: icmp_seq=14 ttl=128 time=0.348 ms  
64 bytes from 192.168.100.24: icmp_seq=15 ttl=128 time=0.629 ms  
64 bytes from 192.168.100.24: icmp_seq=16 ttl=128 time=0.254 ms  
64 bytes from 192.168.100.24: icmp_seq=17 ttl=128 time=0.349 ms  
64 bytes from 192.168.100.24: icmp_seq=18 ttl=128 time=0.643 ms  
64 bytes from 192.168.100.24: icmp_seq=19 ttl=128 time=0.528 ms  
^C  
--- 192.168.100.24 ping statistics ---  
19 packets transmitted, 19 received, 0% packet loss, time 18385ms  
rtt min/avg/max/mdev = 0.254/0.605/1.501/0.274 ms  
atjl_jra_mpe@equipo5:~$
```

Modo NAT.



```
atjl_jra_mpe@equipo5: ~
atjl_jra_mpe@equipo5:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::8f7c:a124:7017:cf14 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1a:9e:fc txqueuelen 1000 (Ethernet)
    RX packets 18 bytes 3043 (3.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 73 bytes 8849 (8.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

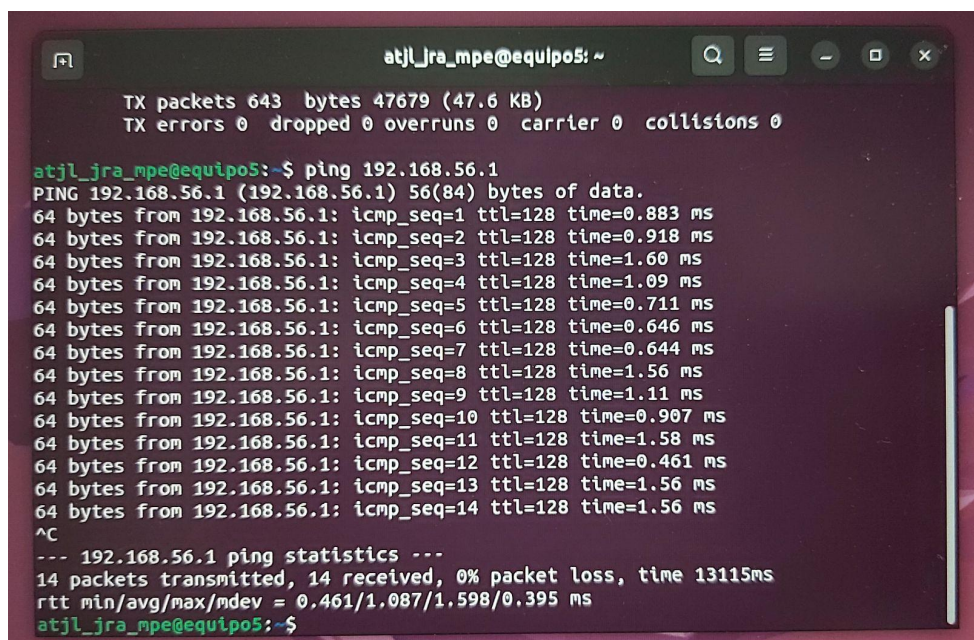
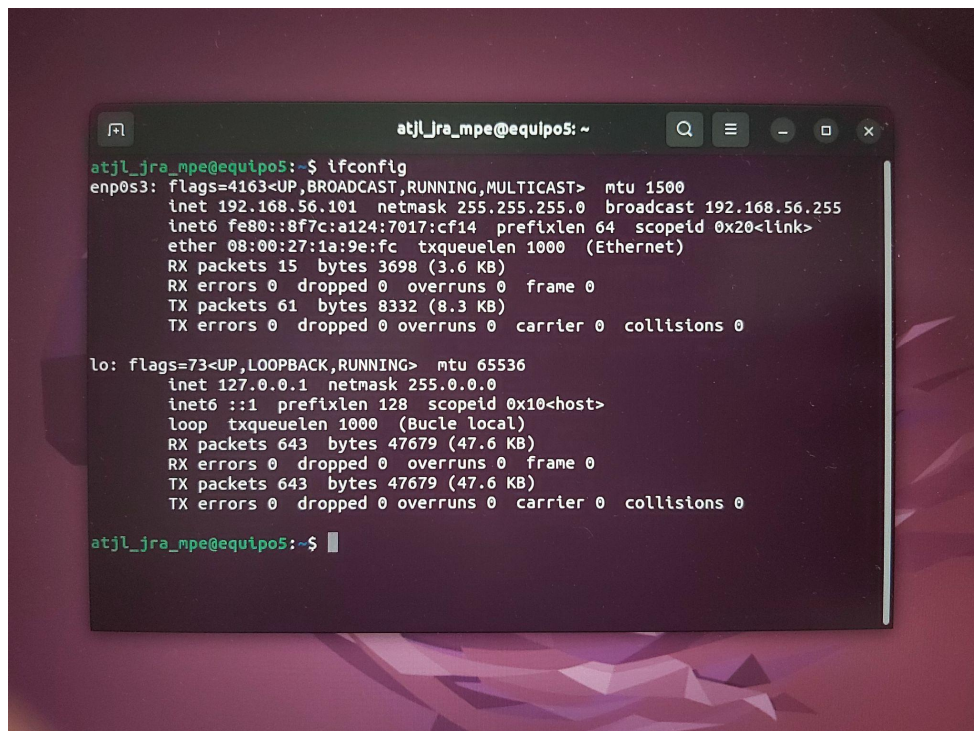
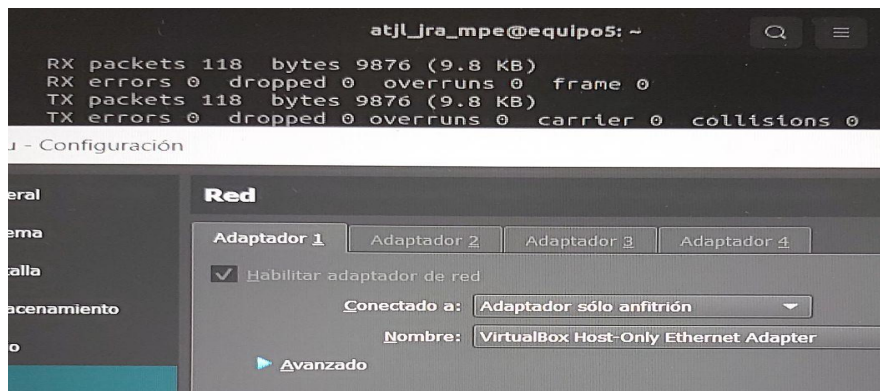
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 118 bytes 9876 (9.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 118 bytes 9876 (9.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

atjl_jra_mpe@equipo5:~$
```

```
atjl_jra_mpe@equipo5: ~
RX packets 118 bytes 9876 (9.8 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 118 bytes 9876 (9.8 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

atjl_jra_mpe@equipo5:~$ ping 192.168.100.24
PING 192.168.100.24 (192.168.100.24) 56(84) bytes of data.
64 bytes from 192.168.100.24: icmp_seq=1 ttl=127 time=1.29 ms
64 bytes from 192.168.100.24: icmp_seq=2 ttl=127 time=2.01 ms
64 bytes from 192.168.100.24: icmp_seq=3 ttl=127 time=1.64 ms
64 bytes from 192.168.100.24: icmp_seq=4 ttl=127 time=1.04 ms
64 bytes from 192.168.100.24: icmp_seq=5 ttl=127 time=1.90 ms
64 bytes from 192.168.100.24: icmp_seq=6 ttl=127 time=2.00 ms
64 bytes from 192.168.100.24: icmp_seq=7 ttl=127 time=0.721 ms
64 bytes from 192.168.100.24: icmp_seq=8 ttl=127 time=0.866 ms
64 bytes from 192.168.100.24: icmp_seq=9 ttl=127 time=0.920 ms
64 bytes from 192.168.100.24: icmp_seq=10 ttl=127 time=0.910 ms
64 bytes from 192.168.100.24: icmp_seq=11 ttl=127 time=2.02 ms
64 bytes from 192.168.100.24: icmp_seq=12 ttl=127 time=2.07 ms
^C
--- 192.168.100.24 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11086ms
rtt min/avg/max/mdev = 0.721/1.448/2.071/0.516 ms
atjl_jra_mpe@equipo5:~$
```


Modo Host-Only



```
atjl_jra_mpe@equipo5: ~  
atjl_jra_mpe@equipo5:~$ ping 192.168.56.1  
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.  
64 bytes from 192.168.56.1: icmp_seq=1 ttl=128 time=0.883 ms  
64 bytes from 192.168.56.1: icmp_seq=2 ttl=128 time=0.918 ms  
64 bytes from 192.168.56.1: icmp_seq=3 ttl=128 time=1.60 ms  
64 bytes from 192.168.56.1: icmp_seq=4 ttl=128 time=1.09 ms  
64 bytes from 192.168.56.1: icmp_seq=5 ttl=128 time=0.711 ms  
64 bytes from 192.168.56.1: icmp_seq=6 ttl=128 time=0.646 ms  
64 bytes from 192.168.56.1: icmp_seq=7 ttl=128 time=0.644 ms  
64 bytes from 192.168.56.1: icmp_seq=8 ttl=128 time=1.56 ms  
64 bytes from 192.168.56.1: icmp_seq=9 ttl=128 time=1.11 ms  
64 bytes from 192.168.56.1: icmp_seq=10 ttl=128 time=0.907 ms  
64 bytes from 192.168.56.1: icmp_seq=11 ttl=128 time=1.58 ms  
64 bytes from 192.168.56.1: icmp_seq=12 ttl=128 time=0.461 ms  
64 bytes from 192.168.56.1: icmp_seq=13 ttl=128 time=1.56 ms  
64 bytes from 192.168.56.1: icmp_seq=14 ttl=128 time=1.56 ms  
^C  
--- 192.168.56.1 ping statistics ---  
14 packets transmitted, 14 received, 0% packet loss, time 13115ms  
rtt min/avg/max/mdev = 0.461/1.087/1.598/0.395 ms  
atjl_jra_mpe@equipo5:~$ ping 192.168.100.24  
ping: connect: La red es inaccesible  
atjl_jra_mpe@equipo5:~$
```

Explica porque la red es inaccesible.

Observamos que el equipo anfitrión se pudo conectar con la máquina virtual, pero la máquina virtual con el equipo anfitrión no, esto sucede porque cuando está en modo Host-Only, la MV está totalmente aislada de la red de área local, ya que la red de la MV está dentro del propio equipo y es invisible e inaccesible para cualquier equipo de la red del equipo. Y en el caso que sí se puede es porque el modo Host-Only asigna una dirección IP a la MV y a través de esta el equipo anfitrión se conecta con ella.

Realiza la tabla colocando los diferentes datos obtenidos en las diferentes configuraciones.

Configuración	IP(inet)	Netmask	Broadcast
Bridge	192.168.100.27	255.255.255.0	192.168.100.255
Nat	10.0.2.15	255.255.255.0	10.0.2.255
Host-Only	192.168.56.101	255.255.255.0	192.168.56.255

Investiga qué archivo se debe modificar para configurar la tarjeta de red desde la terminal y que datos se deben colocar.

Para configurar la tarjeta de red en Ubuntu, usaremos Netplan, el cual se basa en un sistema de configuración basado en YML que se encargará de configurar la red fácilmente.

El YML se encuentra en /etc/netplan/

Al abrir el YML observaremos los parámetros de red que podemos editar.

Los datos que se pueden colocar son:

```
network:
  version: 2
  renderer: NetworkManager/networkd
  ethernets:
    DEVICE_NAME:
      dhcp4: yes/no
      addresses: [IP_ADDRESS/NETMASK]
      routes:
        to: default
        via: GATEWAY
      nameservers:
        addresses: [NAMESERVER_1, NAMESERVER_2]
```

Con el comando `sudo netplan try`, probaremos el archivo de configuración antes de aplicar los cambios.

Después, se deberá imprimir en pantalla “configuración aceptada”, para que así podamos aplicar la configuración. Para ello ejecutamos el comando `sudo netplan apply`.

Finalmente reiniciaremos el servicio, “Network-Manager” si estamos en la versión de escritorio con el comando `sudo systemctl restart network-manager` o “system-networkd” si estamos en la versión server con el comando `sudo systemctl restart system-networkd`

Conclusiones

José Luis: Como conclusión de esta práctica puedo decir que aprendí todo lo que conlleva la creación de una máquina virtual y lo útil que son para simular una computadora física en la que se ejecuta un cierto sistema operativo (Linux), así como darle configuraciones como si estuviéramos en una máquina independiente. Además, aprendí la importancia de la tarjeta de interfaz de red (NIC) que es la administración de redes y sistemas informáticos, entre las diferentes formas de configurar la NIC en el desarrollo de la práctica, notamos que el modo Bridge funciona para que la NIC funcione como un puente o enlace para conectar dos segmentos de red, como si estuvieran en la misma red local. Ahora, en el modo NAT vemos que la NIC realiza la traducción de direcciones de red, por lo que vimos que actúa como un intermediario entre una red local y externa provocando que diversos dispositivos en la red local compartan una dirección IP para acceder a la red externa. Finalmente el modo HostOnly vimos que tiene como comportamiento que se crea un tipo de red privada entre el anfitrión(nuestra máquina) y la máquina virtual, logrando que las máquinas virtuales puedan tener una comunicación entre sí, sin embargo, no pueden acceder a redes externas como el internet.

Abraham: En conclusión sobre los tres modos de configuración NIF. El modo *bridge* proporciona mayor visibilidad y acceso a la red esto ya que la máquina virtual tiene su propia dirección IP en la red local, el modo NAT ofrece una conexión a la máquina virtual que se conecta a la red a través de una interfaz virtual que está administrada por el software de virtualización, y el modo *host-only* las máquinas virtuales pueden comunicarse entre sí y con la máquina host, pero no pueden acceder a la red externa.

Esaú: en esta práctica aprendimos a instalar una máquina virtual, para simular nuestro sistema operativo en este caso Ubuntu. Probamos distintos modos de configuración de red de la tarjeta de interfaz de red (NIC), los cuales fueron el adaptador puente, NAT (el que viene predispuesto en la máquina) y el adaptador Host-Only e hicimos pruebas de conexión en los tres modos ejecutando en terminal el comando `ifconfig` y `ping` (ip del equipo anfitrión). En el modo Host-Only pasó algo particular, al conectar la máquina anfitrión con la máquina virtual si se pudo conectar pero al revés no, esto porque la máquina virtual se aísla y no detecta otros dispositivos, esto nos permitió ver cómo se comporta la máquina virtual cuando le otorgamos acceso a la red en estos 3 modos, obteniendo su ip, netmask y broadcast. De igual manera vimos la forma más actual de configurar la tarjeta de red en Ubuntu.

Referencias

- Galán, V. (2013). Tipos de conexiones de red en software de virtualización: VirtualBox y VMware | Tícarte. Recuperado el 29 de agosto del 2023, de <https://www.ticarte.com/sites/su/users/7/file/tipos-de-redes-en-virtualbox-y-vmware.pdf>
- Méndez, I. (2022). Cómo configurar tarjeta de red en Ubuntu | linuxbasico. Recuperado el 29 de agosto del 2023, de <https://linuxbasico.com/configurar-tarjeta-red-ubuntu>