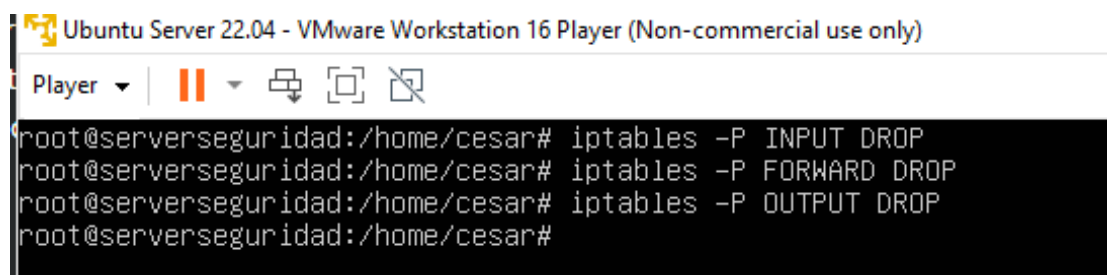


Configuración de NAT en Iptables y ufw.

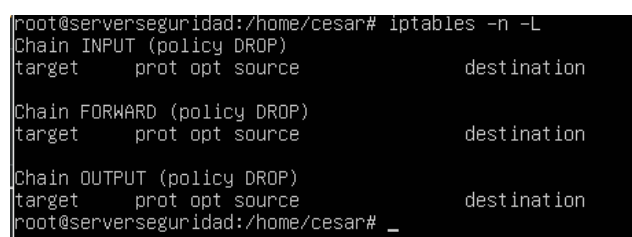
NAT con iptables.

Lo primero será establecer las políticas por defecto que debe seguir iptables cuando reciba un paquete.



```
root@serverseguridad:/home/cesar# iptables -P INPUT DROP
root@serverseguridad:/home/cesar# iptables -P FORWARD DROP
root@serverseguridad:/home/cesar# iptables -P OUTPUT DROP
root@serverseguridad:/home/cesar#
```

A continuación vamos a listar las reglas iptables que tenemos actualmente, es decir ninguna, para comprobar que las políticas por defecto efectivamente están en drop.

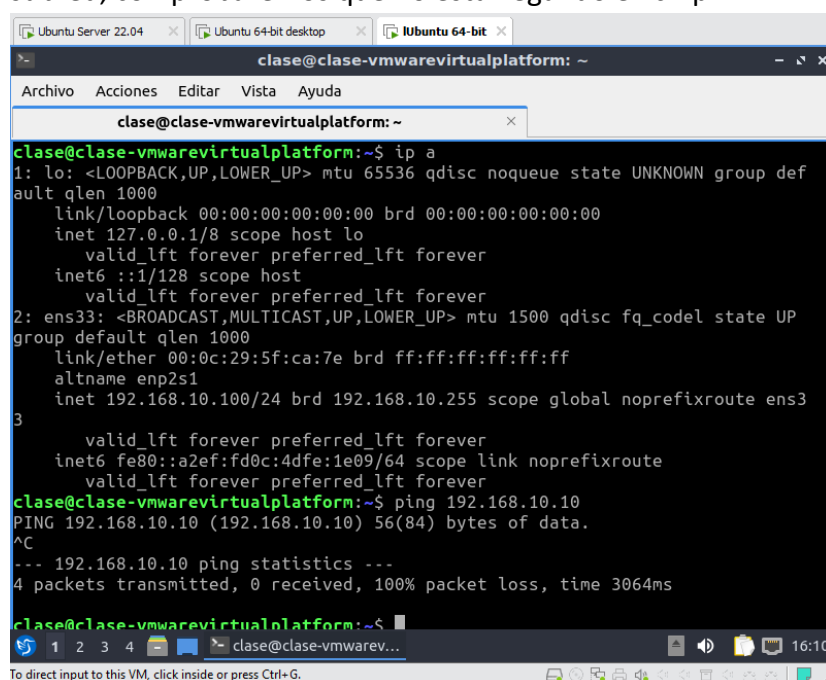


```
root@serverseguridad:/home/cesar# iptables -n -L
Chain INPUT (policy DROP)
target     prot opt source                destination

Chain FORWARD (policy DROP)
target     prot opt source                destination

Chain OUTPUT (policy DROP)
target     prot opt source                destination
root@serverseguridad:/home/cesar# _
```

Si ahora probamos a hacer ping desde una maquina cliente al servidor con la ip en su misma subred, comprobaremos que no esta llegando el icmp.



```
clase@clase-vmwarevirtualplatform: ~
Archivo Acciones Editar Vista Ayuda
clase@clase-vmwarevirtualplatform: ~
clase@clase-vmwarevirtualplatform:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 00:0c:29:5f:ca:7e brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.10.100/24 brd 192.168.10.255 scope global noprefixroute ens3
3
        valid_lft forever preferred_lft forever
    inet6 fe80::a2ef:fd0c:4dfe:1e09/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
clase@clase-vmwarevirtualplatform:~$ ping 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
^C
--- 192.168.10.10 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3064ms

clase@clase-vmwarevirtualplatform:~$
```

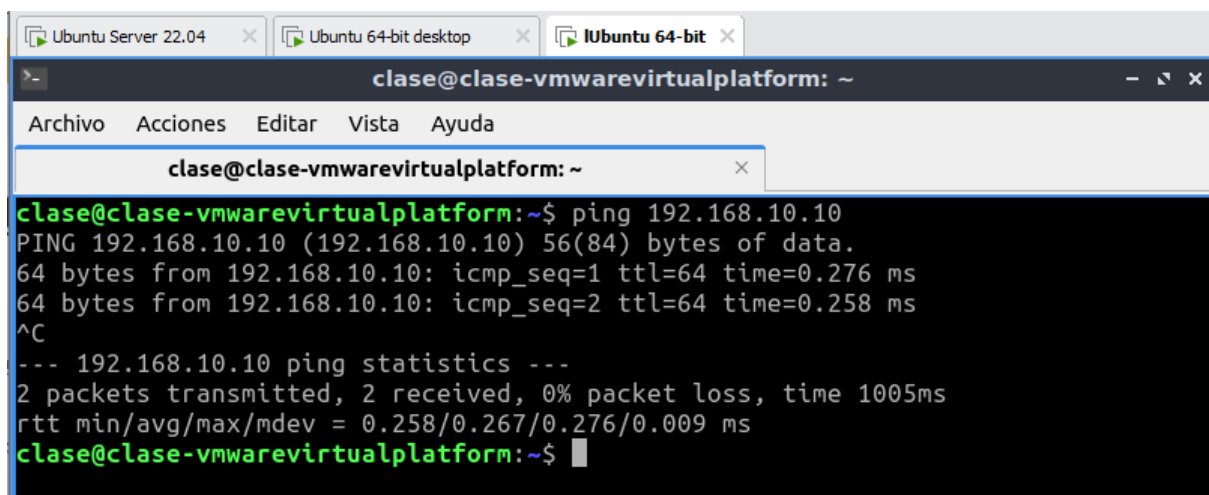
Para ello hay que permitir el ping al servidor desde cada una de las máquinas, haciendo uso de las cadenas INPUT y OUTPUT.

```
root@serverseguridad:/etc# iptables -L -n
Chain INPUT (policy DROP)
target     prot opt source                destination
ACCEPT     all  --  192.168.20.100         0.0.0.0/0
ACCEPT     all  --  192.168.10.100        0.0.0.0/0

Chain FORWARD (policy DROP)
target     prot opt source                destination

Chain OUTPUT (policy DROP)
target     prot opt source                destination
ACCEPT     all  --  0.0.0.0/0             192.168.10.100
ACCEPT     all  --  0.0.0.0/0             192.168.20.100
root@serverseguridad:/etc#
```

Podemos ver que ahora el ping se realiza correctamente.

A screenshot of a terminal window titled 'clase@clase-vmwarevirtualplatform: ~'. The window shows the output of a 'ping 192.168.10.10' command. The output indicates that 2 packets were transmitted and 2 were received with 0% packet loss. The terminal text is as follows:

```
clase@clase-vmwarevirtualplatform:~$ ping 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
64 bytes from 192.168.10.10: icmp_seq=1 ttl=64 time=0.276 ms
64 bytes from 192.168.10.10: icmp_seq=2 ttl=64 time=0.258 ms
^C
--- 192.168.10.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1005ms
rtt min/avg/max/mdev = 0.258/0.267/0.276/0.009 ms
clase@clase-vmwarevirtualplatform:~$
```

Ahora vamos a permitir el ping entre clientes en diferentes subredes mediante la cadena de FORWARD.(el comando iptables está omitido porque es una captura del iptables -S)

```
-A FORWARD -s 192.168.10.100/32 -d 192.168.20.100/32 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A FORWARD -s 192.168.20.100/32 -d 192.168.10.100/32 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A FORWARD -s 192.168.20.100/32 -d 192.168.10.100/32 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A FORWARD -s 192.168.10.100/32 -d 192.168.20.100/32 -p icmp -m icmp --icmp-type 8 -j ACCEPT
```

Pero el comando original es el siguiente:

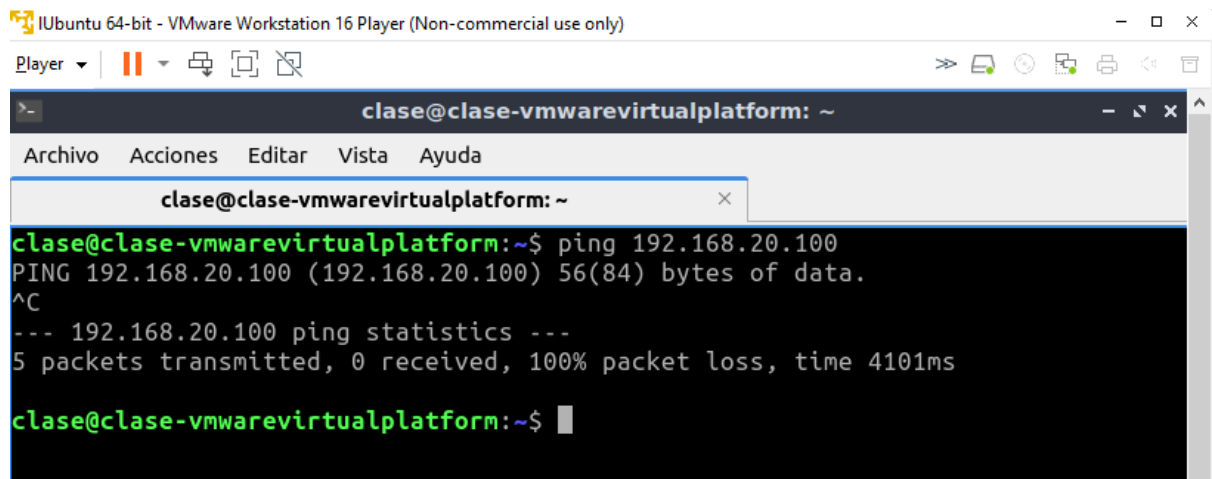
```
iptables -I FORWARD -p icmp --icmp-type echo-request -s 192.168.10.100/32 -d
192.168.20.100/32 -j ACCEPT
```

```
iptables -I FORWARD -p icmp --icmp-type echo-reply -d 192.168.10.100/32 -s
192.168.20.100/32 -j ACCEPT
```

```
iptables -I FORWARD -p icmp --icmp-type echo-request -d 192.168.10.100/32 -s
192.168.20.100/32 -j ACCEPT
```

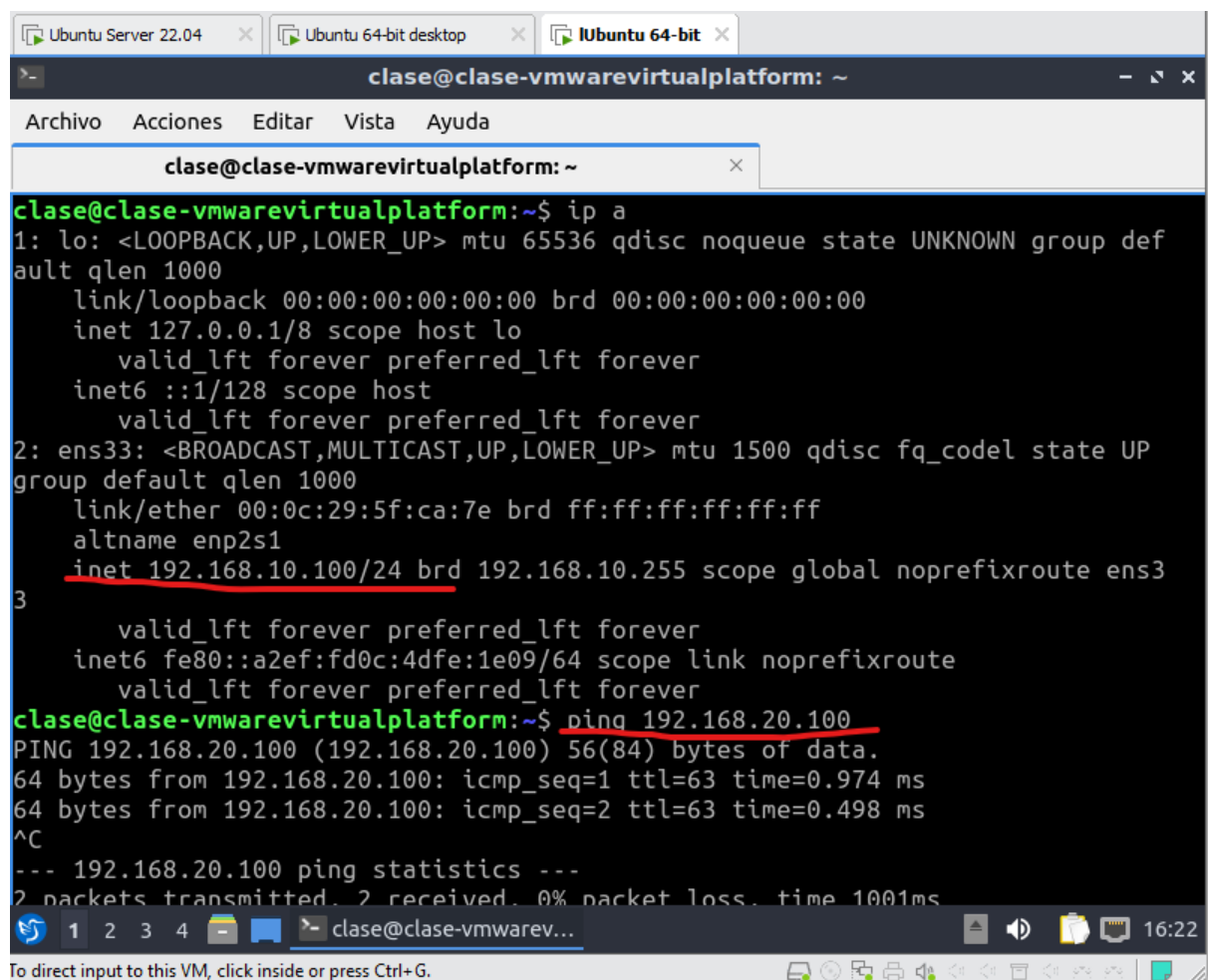
```
iptables -I FORWARD -p icmp --icmp-type echo-reply -s 192.168.10.100/32 -d 192.168.20.100/32 -j ACCEPT
```

Si probamos a hacer ping antes de configurar el FORWARD, veríamos que no funciona.



The screenshot shows a terminal window titled 'clase@clase-vmwarevirtualplatform: ~'. The user has entered the command 'ping 192.168.20.100'. The output shows 'PING 192.168.20.100 (192.168.20.100) 56(84) bytes of data.' followed by a control character '^C'. Then, it shows '--- 192.168.20.100 ping statistics ---' and '5 packets transmitted, 0 received, 100% packet loss, time 4101ms'. The prompt returns to 'clase@clase-vmwarevirtualplatform:~\$'.

Pero después de incluir las reglas, sí.

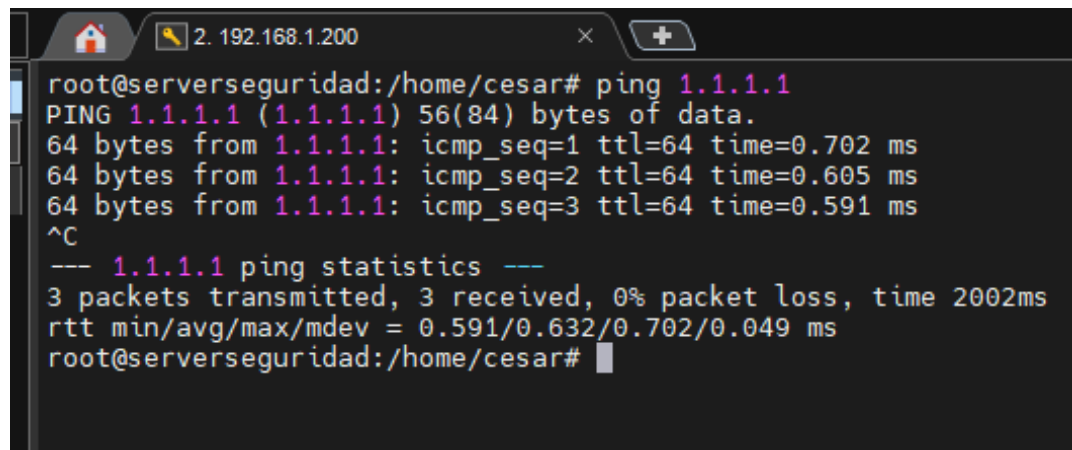


The screenshot shows a terminal window titled 'clase@clase-vmwarevirtualplatform: ~'. The user has entered the command 'ip a'. The output shows the configuration for the 'ens33' interface, including the IP address '192.168.10.100/24'. The user then enters the command 'ping 192.168.20.100'. The output shows 'PING 192.168.20.100 (192.168.20.100) 56(84) bytes of data.' followed by two successful ping results: '64 bytes from 192.168.20.100: icmp_seq=1 ttl=63 time=0.974 ms' and '64 bytes from 192.168.20.100: icmp_seq=2 ttl=63 time=0.498 ms'. Then, it shows '--- 192.168.20.100 ping statistics ---' and '2 packets transmitted, 2 received, 0% packet loss, time 1001ms'. The prompt returns to 'clase@clase-vmwarevirtualplatform:~\$'.

Por último solo falta tener conectividad a internet. Para ello usaremos la cadena FORWARD de la tabla filter y la cadena POSTROUTING de la tabla nat.

```
root@serverseguridad:/home/cesar# #PING INTERNET
iptables -A OUTPUT -p icmp --icmp-type echo-request -o ens33 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -i ens33 -j ACCEPT
```

Esto sirve para permitir el ping del servidor a internet.

A terminal window titled '2. 192.168.1.200' showing a successful ping command. The user enters 'ping 1.1.1.1' and the output shows three successful pings with decreasing times. The statistics show 3 packets transmitted, 3 received, and 0% packet loss.

```
root@serverseguridad:/home/cesar# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data:
64 bytes from 1.1.1.1: icmp_seq=1 ttl=64 time=0.702 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=64 time=0.605 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=64 time=0.591 ms
^C
--- 1.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.591/0.632/0.702/0.049 ms
root@serverseguridad:/home/cesar#
```

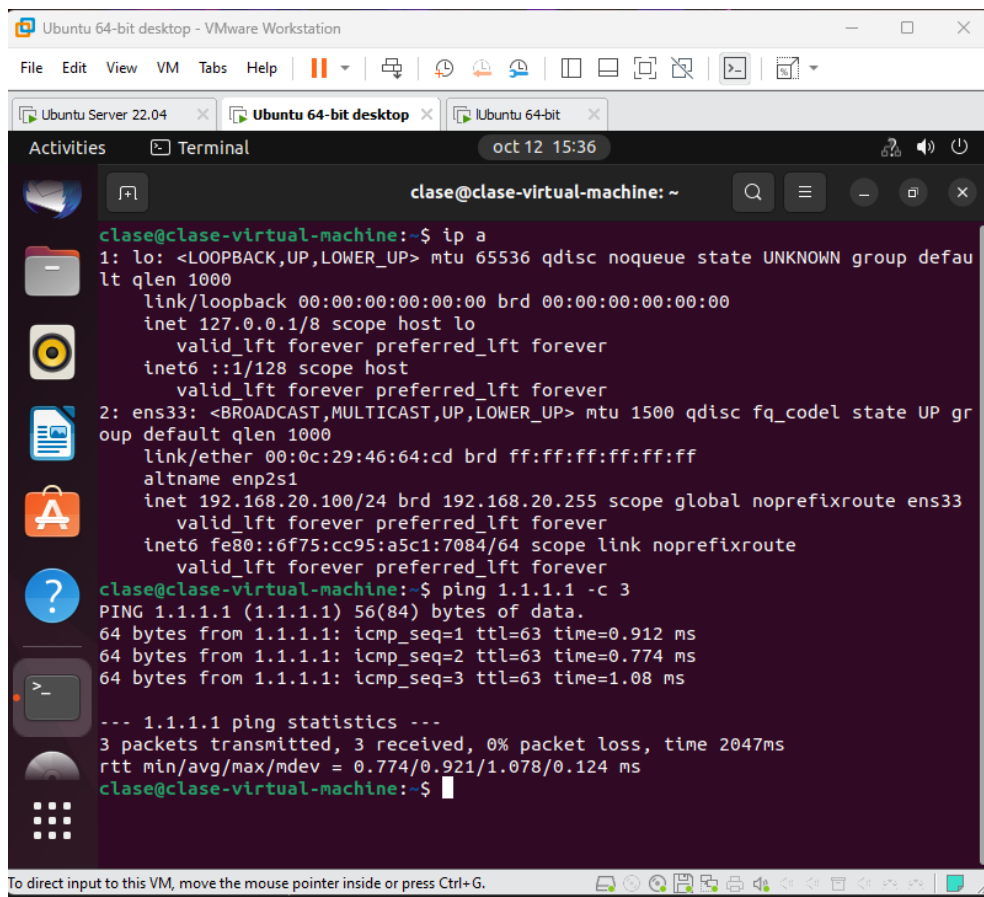
Pero permitir el tráfico web de un cliente requiere mas ordenes (el comando iptables está omitido porque es una captura del iptables -S).

```
-A FORWARD -s 192.168.10.100/32 -j ACCEPT
-A FORWARD -d 192.168.10.100/32 -j ACCEPT
-A FORWARD -s 192.168.20.100/32 -j ACCEPT
-A FORWARD -d 192.168.20.100/32 -j ACCEPT
```

Aquí hemos permitido el paso de cualquier red con origen o destino las ips de los clientes. Pero aun falta convertir las ips privadas de los clientes a la ip publica de internet del router. Dado que la ip publica no es estática y depende de nuestro proveedor telefónico, vamos a usar enmascaramiento (el comando iptables está omitido porque es una captura del iptables -S).

```
-A POSTROUTING -s 192.168.10.0/24 -o ens33 -j MASQUERADE
-A POSTROUTING -s 192.168.20.0/24 -o ens33 -j MASQUERADE
root@serverseguridad:/home/cesar#
```

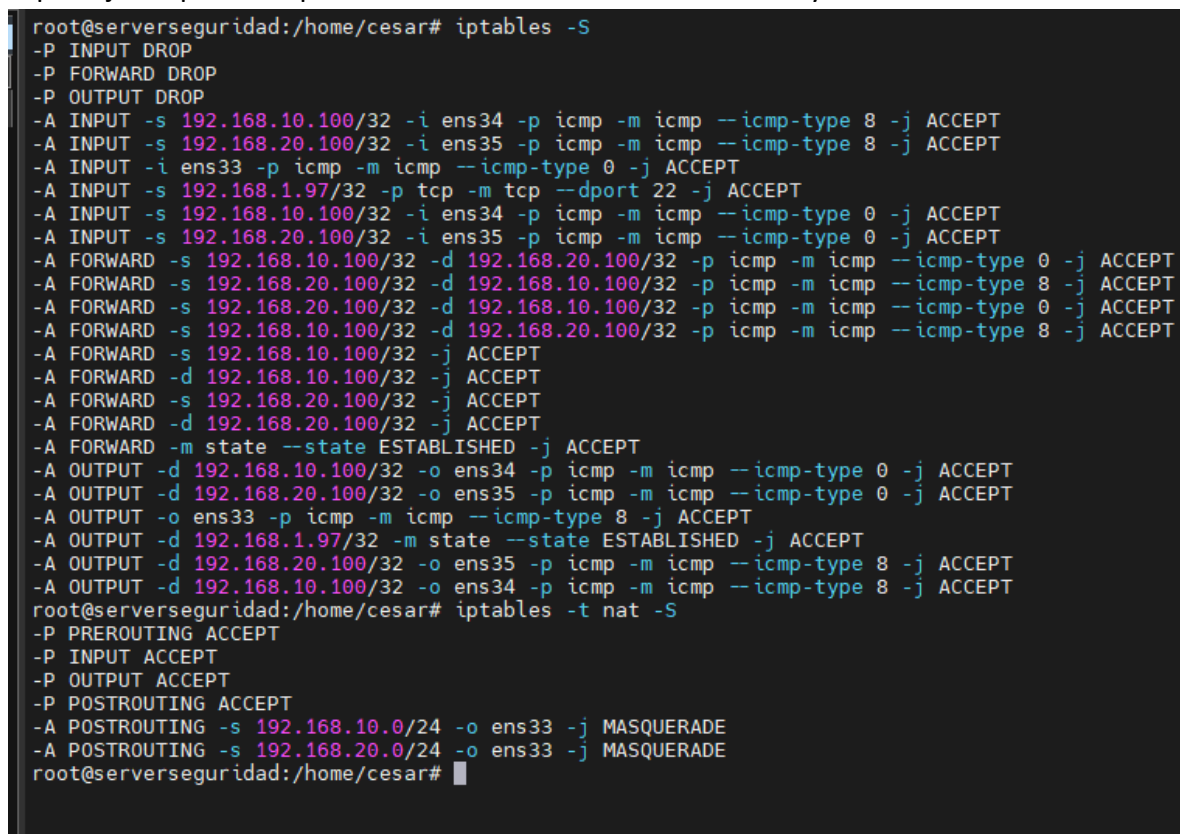
Ahora ya sí, podemos hacer ping a internet y navegar sin problema desde nuestros clientes.



```
clase@clase-virtual-machine:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:46:64:cd brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.20.100/24 brd 192.168.20.255 scope global noprefixroute ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::6f75:cc95:a5c1:7084/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
clase@clase-virtual-machine:~$ ping 1.1.1.1 -c 3
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data:
64 bytes from 1.1.1.1: icmp_seq=1 ttl=63 time=0.912 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=63 time=0.774 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=63 time=1.08 ms

--- 1.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2047ms
rtt min/avg/max/mdev = 0.774/0.921/1.078/0.124 ms
clase@clase-virtual-machine:~$
```

Aquí dejo un par de capturas con todos los comandos usados y las diferentes tablas.



```
root@serverseguridad:/home/cesar# iptables -F
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT DROP
-A INPUT -s 192.168.10.100/32 -i ens34 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -s 192.168.20.100/32 -i ens35 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -i ens33 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A INPUT -s 192.168.1.97/32 -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -s 192.168.10.100/32 -i ens34 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A INPUT -s 192.168.20.100/32 -i ens35 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A FORWARD -s 192.168.10.100/32 -d 192.168.20.100/32 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A FORWARD -s 192.168.20.100/32 -d 192.168.10.100/32 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A FORWARD -s 192.168.10.100/32 -d 192.168.20.100/32 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A FORWARD -s 192.168.10.100/32 -j ACCEPT
-A FORWARD -d 192.168.10.100/32 -j ACCEPT
-A FORWARD -s 192.168.20.100/32 -j ACCEPT
-A FORWARD -d 192.168.20.100/32 -j ACCEPT
-A FORWARD -m state --state ESTABLISHED -j ACCEPT
-A OUTPUT -d 192.168.10.100/32 -o ens34 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A OUTPUT -d 192.168.20.100/32 -o ens35 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A OUTPUT -o ens33 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A OUTPUT -d 192.168.1.97/32 -m state --state ESTABLISHED -j ACCEPT
-A OUTPUT -d 192.168.20.100/32 -o ens35 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A OUTPUT -d 192.168.10.100/32 -o ens34 -p icmp -m icmp --icmp-type 8 -j ACCEPT
root@serverseguridad:/home/cesar# iptables -t nat -F
-P PREROUTING ACCEPT
-P INPUT ACCEPT
-P OUTPUT ACCEPT
-P POSTROUTING ACCEPT
-A POSTROUTING -s 192.168.10.0/24 -o ens33 -j MASQUERADE
-A POSTROUTING -s 192.168.20.0/24 -o ens33 -j MASQUERADE
root@serverseguridad:/home/cesar#
```

```

root@serverseguridad:/home/cesar# iptables -L -nv
Chain INPUT (policy DROP 7 packets, 663 bytes)
pkts bytes target prot opt in out source destination icmptype
0 0 ACCEPT icmp -- ens34 * 192.168.10.100 0.0.0.0/0 icmptype 8
0 0 ACCEPT icmp -- ens35 * 192.168.20.100 0.0.0.0/0 icmptype 8
3 252 ACCEPT icmp -- ens33 * 0.0.0.0/0 0.0.0.0/0 icmptype 0
822 56080 ACCEPT tcp -- * * 192.168.1.97 0.0.0.0/0 tcp dpt:22
2 168 ACCEPT icmp -- ens34 * 192.168.10.100 0.0.0.0/0 icmptype 0
0 0 ACCEPT icmp -- ens35 * 192.168.20.100 0.0.0.0/0 icmptype 0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination icmptype
0 0 ACCEPT icmp -- * * 192.168.10.100 192.168.20.100 icmptype 0
0 0 ACCEPT icmp -- * * 192.168.20.100 192.168.10.100 icmptype 8
0 0 ACCEPT icmp -- * * 192.168.20.100 192.168.10.100 icmptype 0
0 0 ACCEPT icmp -- * * 192.168.10.100 192.168.20.100 icmptype 8
14 1176 ACCEPT all -- * * 192.168.10.100 0.0.0.0/0
14 1176 ACCEPT all -- * * 0.0.0.0/0 192.168.10.100
11 924 ACCEPT all -- * * 192.168.20.100 0.0.0.0/0
11 924 ACCEPT all -- * * 0.0.0.0/0 192.168.20.100
0 0 ACCEPT all -- * * 0.0.0.0/0 0.0.0.0/0 state ESTABLISHED

Chain OUTPUT (policy DROP 21 packets, 1688 bytes)
pkts bytes target prot opt in out source destination icmptype
0 0 ACCEPT icmp -- * ens34 0.0.0.0/0 192.168.10.100 icmptype 0
0 0 ACCEPT icmp -- * ens35 0.0.0.0/0 192.168.20.100 icmptype 0
3 252 ACCEPT icmp -- * ens33 0.0.0.0/0 0.0.0.0/0 icmptype 8
535 48360 ACCEPT all -- * * 0.0.0.0/0 192.168.1.97 state ESTABLISHED
0 0 ACCEPT icmp -- * ens35 0.0.0.0/0 192.168.20.100 icmptype 8
2 168 ACCEPT icmp -- * ens34 0.0.0.0/0 192.168.10.100 icmptype 8

root@serverseguridad:/home/cesar# iptables -t nat -L -nv
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
3 252 MASQUERADE all -- * ens33 192.168.10.0/24 0.0.0.0/0
4 336 MASQUERADE all -- * ens33 192.168.20.0/24 0.0.0.0/0
root@serverseguridad:/home/cesar#

```


Para guardar las normas iptables es tan sencillo como crear un fichero /etc/iptables y redirigir las tablas actuales a un archivo en ese fichero, luego, tras reiniciar el servidor, bastará con hacer un iptables restore y listo.

```
root@serverseguridad:/home/cesar# iptables-save > /etc/iptables/rules.v4
root@serverseguridad:/home/cesar# cat /etc/iptables/rules.v4
# Generated by iptables-save v1.8.7 on Wed Oct 12 13:45:12 2022
*filter
:INPUT DROP [9:727]
:FORWARD DROP [0:0]
:OUTPUT DROP [21:1688]
-A INPUT -s 192.168.10.100/32 -i ens34 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -s 192.168.20.100/32 -i ens35 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -i ens33 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A INPUT -s 192.168.1.97/32 -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -s 192.168.10.100/32 -i ens34 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A INPUT -s 192.168.20.100/32 -i ens35 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A FORWARD -s 192.168.10.100/32 -d 192.168.20.100/32 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A FORWARD -s 192.168.20.100/32 -d 192.168.10.100/32 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A FORWARD -s 192.168.20.100/32 -d 192.168.10.100/32 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A FORWARD -s 192.168.10.100/32 -d 192.168.20.100/32 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A FORWARD -s 192.168.10.100/32 -j ACCEPT
-A FORWARD -d 192.168.10.100/32 -j ACCEPT
-A FORWARD -s 192.168.20.100/32 -j ACCEPT
-A FORWARD -d 192.168.20.100/32 -j ACCEPT
-A FORWARD -m state --state ESTABLISHED -j ACCEPT
-A OUTPUT -d 192.168.10.100/32 -o ens34 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A OUTPUT -d 192.168.20.100/32 -o ens35 -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A OUTPUT -o ens33 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A OUTPUT -d 192.168.1.97/32 -m state --state ESTABLISHED -j ACCEPT
-A OUTPUT -d 192.168.20.100/32 -o ens35 -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A OUTPUT -d 192.168.10.100/32 -o ens34 -p icmp -m icmp --icmp-type 8 -j ACCEPT
COMMIT
# Completed on Wed Oct 12 13:45:12 2022
# Generated by iptables-save v1.8.7 on Wed Oct 12 13:45:12 2022
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 192.168.10.0/24 -o ens33 -j MASQUERADE
-A POSTROUTING -s 192.168.20.0/24 -o ens33 -j MASQUERADE
COMMIT
# Completed on Wed Oct 12 13:45:12 2022
```

NAT con ufw.

Lo primero es acceder al archivo con las políticas por defecto de ufw. Podemos comprobar que están en drop.

```
GNU nano 0.2
# /etc/default/ufw
#
# Set to yes to apply rules to support IPv6 (no means only IPv6 on loopback
# accepted). You will need to 'disable' and then 'enable' the firewall for
# the changes to take affect.
IPV6=yes
# Set the default input policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_INPUT_POLICY="DROP"
# Set the default output policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_OUTPUT_POLICY="DROP"
# Set the default forward policy to ACCEPT, DROP or REJECT. Please note that
# if you change this you will most likely want to adjust your rules
DEFAULT_FORWARD_POLICY="DROP"
# Set the default application policy to ACCEPT, DROP, REJECT or SKIP. Please
# note that setting this to ACCEPT may be a security risk. See 'man ufw' for
# details
DEFAULT_APPLICATION_POLICY="SKIP"
# By default, ufw only touches its own chains. Set this to 'yes' to have ufw
# manage the built-in chains too. Warning: setting this to 'yes' will break
# non-ufw managed firewall rules
MANAGE_BUILTINS=no
```

Ahora accedemos a las before rules y terminamos de arreglar el fichero para permitir los pings tanto entre clientes como a internet.

```
# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-reply -j ACCEPT

# añadimos el output para que funcionen los pings
-A ufw-before-output -p icmp --icmp-type echo-request -j ACCEPT

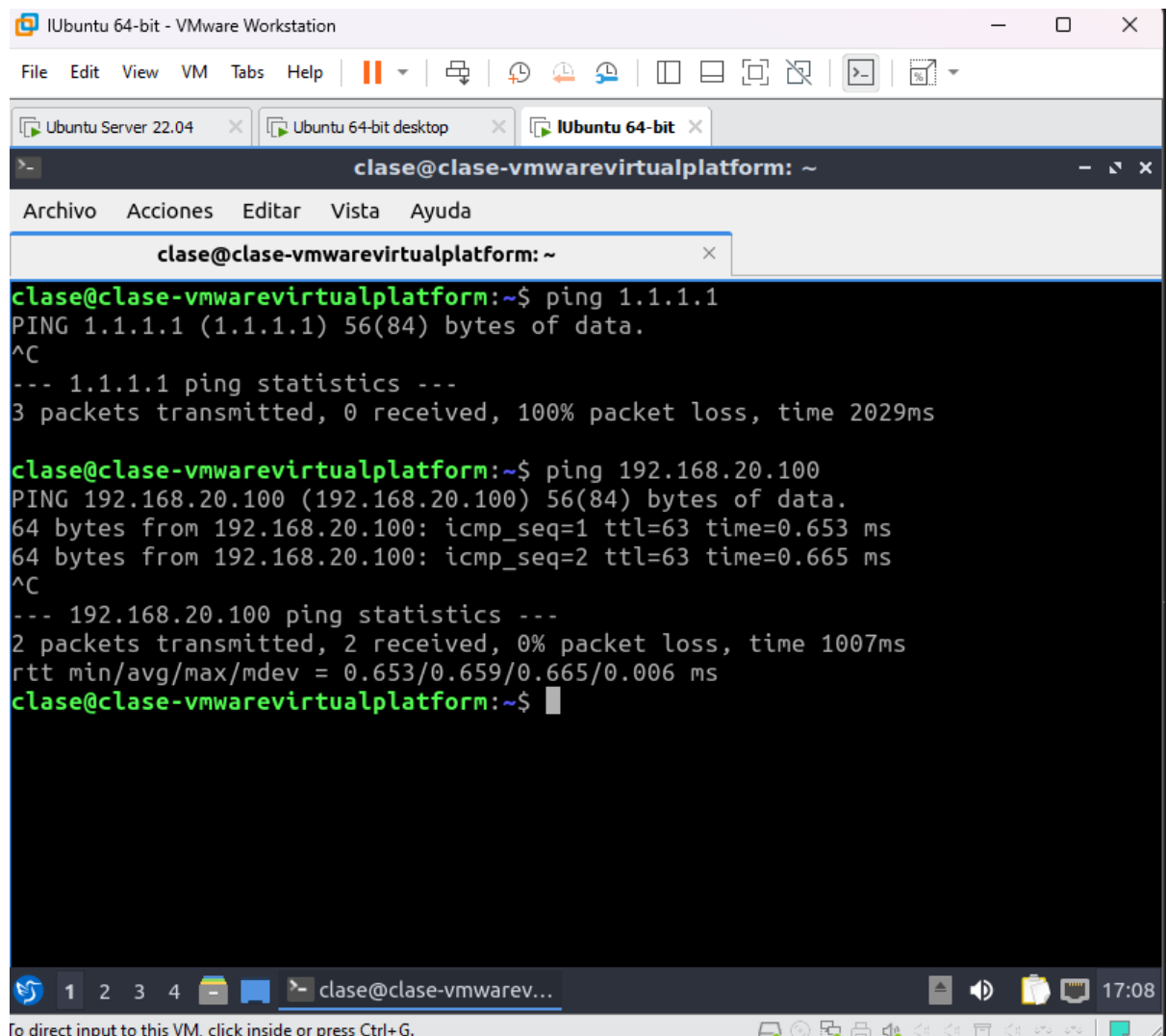
# ok icmp code for FORWARD
-A ufw-before-forward -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type echo-request -j ACCEPT

# lo mismo con el forward
-A ufw-before-forward -p icmp --icmp-type echo-reply -j ACCEPT
```


Habilitamos el servicio ufw.

```
root@serverseguridad:/home/cesar# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@serverseguridad:/home/cesar#
```

Vemos que no hay ping a internet pero si entre clientes, justo cómo hemos establecido.



```
clase@clase-vmwarevirtualplatform: ~
Archivo Acciones Editar Vista Ayuda

clase@clase-vmwarevirtualplatform: ~
clase@clase-vmwarevirtualplatform:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
^C
--- 1.1.1.1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2029ms

clase@clase-vmwarevirtualplatform:~$ ping 192.168.20.100
PING 192.168.20.100 (192.168.20.100) 56(84) bytes of data.
64 bytes from 192.168.20.100: icmp_seq=1 ttl=63 time=0.653 ms
64 bytes from 192.168.20.100: icmp_seq=2 ttl=63 time=0.665 ms
^C
--- 192.168.20.100 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.653/0.659/0.665/0.006 ms
clase@clase-vmwarevirtualplatform:~$
```

Ahora añadimos la tabla nat, con su correspondiente regla.

```
# Agregamos la tabla nat, para poder hacer ping a internet desde los clientes.
*nat

:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o ens33 -j MASQUERADE

COMMIT
```

Vemos que ya tenemos conexión icmp a internet.

```

root@serverseguridad:/home/cesar# ufw reload
Firewall reloaded
root@serverseguridad:/home/cesar# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=64 time=0.860 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=64 time=0.584 ms
^C
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.584/0.722/0.860/0.138 ms
root@serverseguridad:/home/cesar#

```

```

clase@clase-vmwarevirtualplatform: ~
Archivo Acciones Editar Vista Ayuda
clase@clase-vmwarevirtualplatform: ~
clase@clase-vmwarevirtualplatform:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
^C
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1008ms

clase@clase-vmwarevirtualplatform:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=63 time=0.946 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=63 time=1.08 ms
^C
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1008ms
rtt min/avg/max/mdev = 0.946/1.011/1.077/0.065 ms
clase@clase-vmwarevirtualplatform:~$

```

Pero supongamos que queremos ser más específicos y permitir solo el tráfico a internet de una de las redes. En ese caso lo que habría que hacer es especificar el origen en la regla nat.

```

# Agregamos la tabla nat, para poder hacer ping a internet desde los clientes.
*nat

:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 192.168.10.100/32 -o ens33 -j MASQUERADE

COMMIT

```

Pero hay que tener cuidado de borrar las tablas nat de iptables porque aunque cambiemos la configuración de ufw la de iptables se mantiene y puede darnos problemas.

```

root@serverseguridad:/home/cesar# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE all  --  anywhere              anywhere
MASQUERADE all  --  192.168.10.100        anywhere
root@serverseguridad:/home/cesar# iptables -t nat -F
root@serverseguridad:/home/cesar# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
root@serverseguridad:/home/cesar# ufw reload
Firewall reloaded
root@serverseguridad:/home/cesar# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE all  --  192.168.10.100        anywhere
root@serverseguridad:/home/cesar#

```

Si vamos a la máquina cliente de la subred 192.168.20.0/24, vemos que efectivamente ya no tiene acceso a internet.

```

Ubuntu 64-bit desktop - VMware Workstation
File Edit View VM Tabs Help
Ubuntu Server 22.04 Ubuntu 64-bit desktop Ubuntu 64-bit
Activities Terminal oct 12 17:38
clase@clase-virtual-machine: ~
clase@clase-virtual-machine:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
^C
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1013ms
clase@clase-virtual-machine:~$

```

