

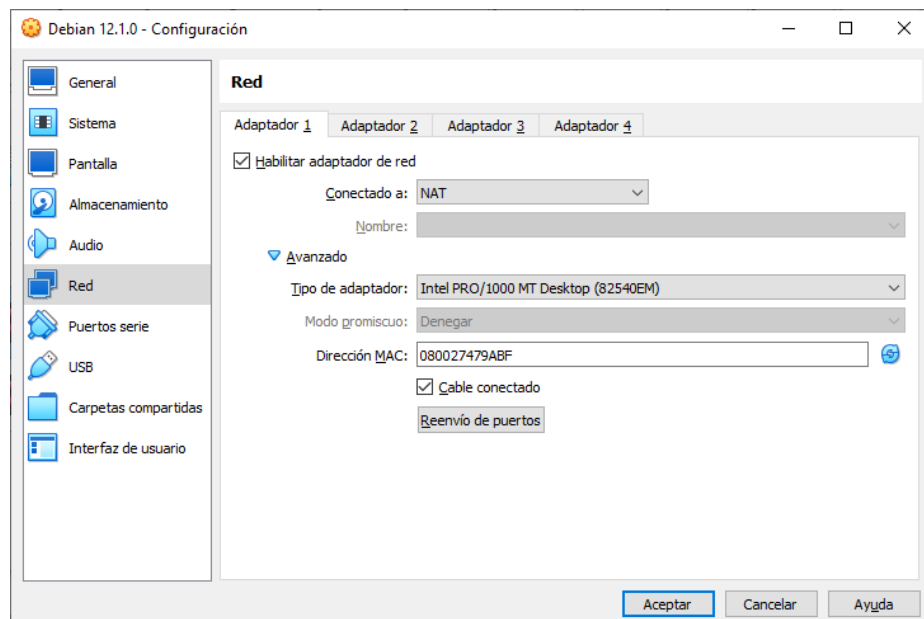
Ejercicio guiado. Configuración de dispositivos Linux. Configuración final.

NOTA: Todos los comandos deben ejecutarse como administrador. Si no estás como administrador en un terminal Linux (dependiendo de la versión) ejecuta alguno los los siguientes comandos:

```
Su -  
su  
sudo y_el_comando_que_quieras_ejecutar
```

Cuando instalas la máquina Debian

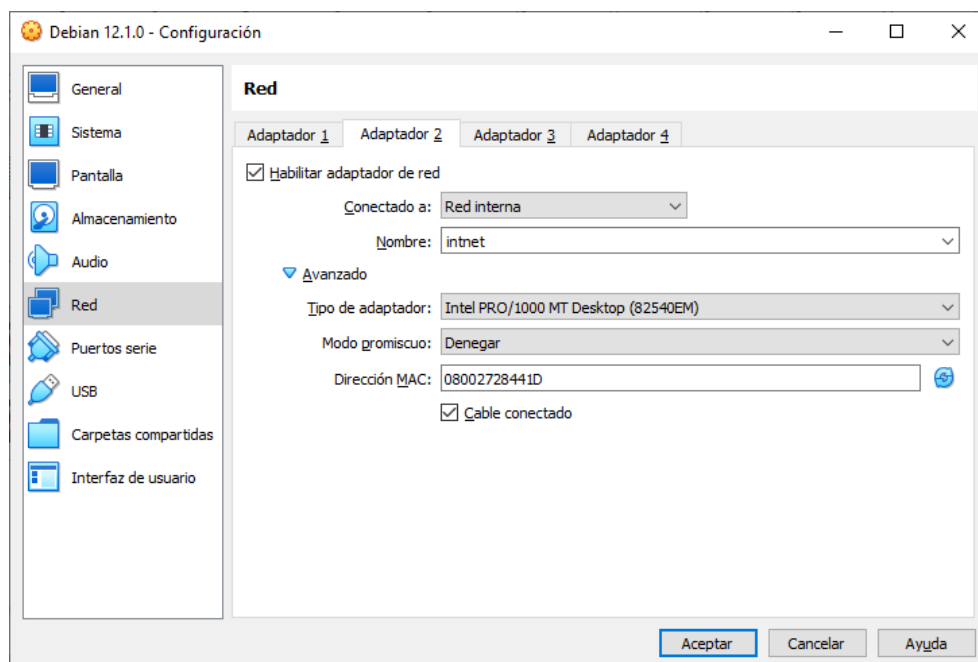
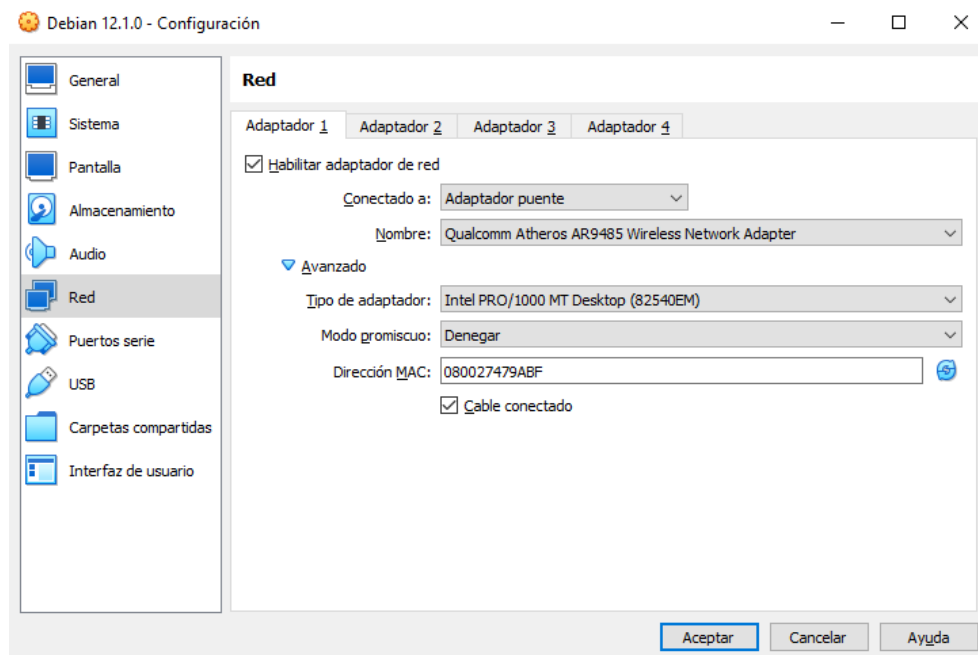
Deja que el adaptador esté en modo NAT porque al tratarse de una instalación a través de la red garantizas tener conectividad a Internet.



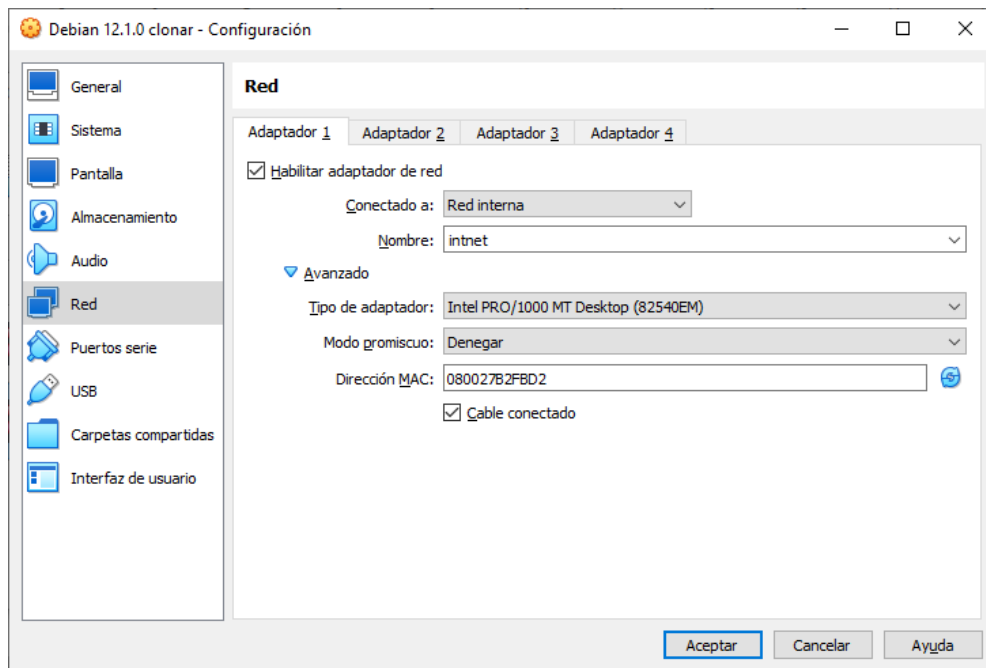
Una vez está instalado el sistema, apaga la máquina Debian y haz una clonación. Añade los adaptadores necesarios y elije el modo correcto para cada adaptador.

Así debe quedar la configuración final.

Asegurarse de que tenéis esta configuración en Virtual Box, que los cables estén conectados y que las direcciones MAC sean distintas en cada adaptador:



En la máquina clonada:



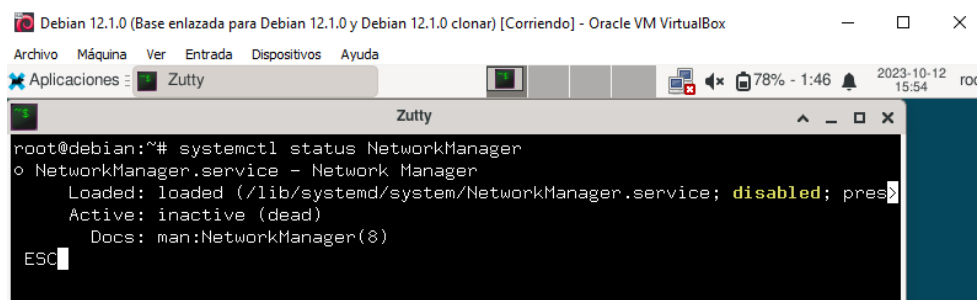
En la máquina Debian

Debe estar deshabilitado Network Manager.

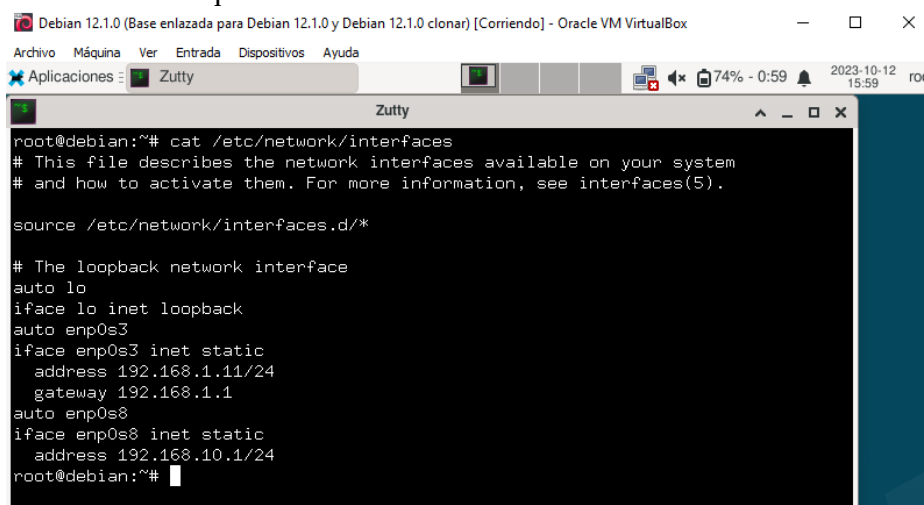
En un terminal linux:

systemctl stop NetworkManager //para Network Manager en esta sesión

systemctl disable NetworkManager //Aunque reiniciemos la máquina no se inicia el servicio.



La configuración de red debe quedar así:



Sino hay que editarlo con algún editor.

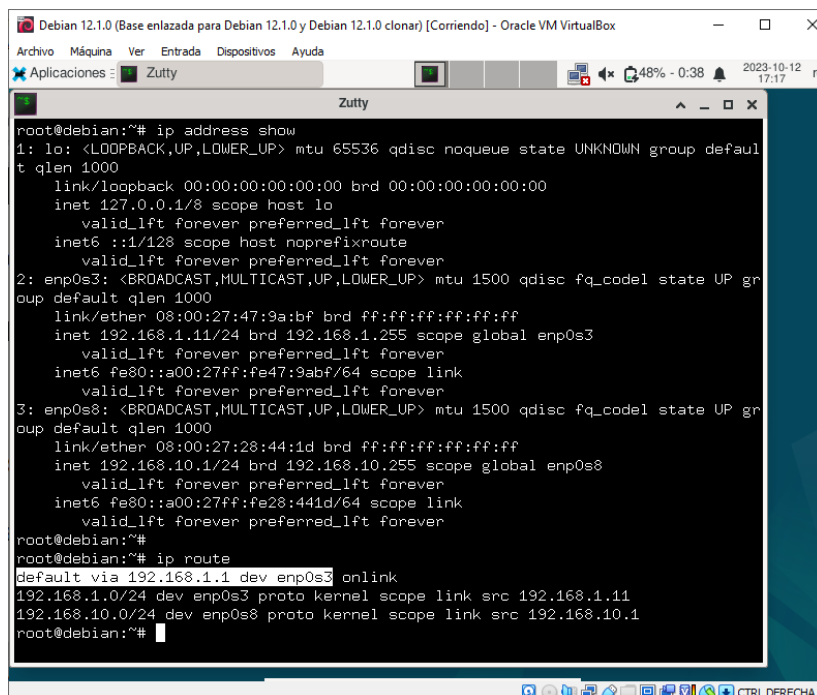
Después tendremos que reiniciar el servicio de red para que los cambios sean efectivos:

systemctl restart networking

No debe dar ninguna respuesta, si la da es que tenemos errores en el archivo “interfaces”. Toca corregirlo.

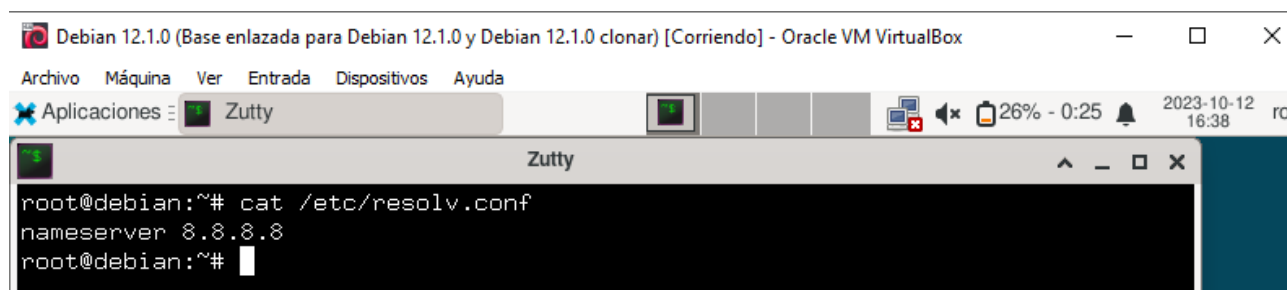
Ojo: la IP que elegimos para la interfaz enp0s3 no puede estar repetida en vuestra Red doméstica, si fuera el caso tenéis que elegir otra que no esté siendo usada por algún otro dispositivo.

Comprobamos que la configuración de los adaptadores es correcta, y que la puerta de enlace es la correcta:



```
root@debian:~# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    link/ether 08:00:27:47:9a:bf brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe47:9abf/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    link/ether 08:00:27:28:44:1d brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.1/24 brd 192.168.10.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe28:441d/64 scope link
        valid_lft forever preferred_lft forever
root@debian:~# ip route
default via 192.168.1.1 dev enp0s3 onlink
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.11
192.168.10.0/24 dev enp0s8 proto kernel scope link src 192.168.10.1
root@debian:~#
```

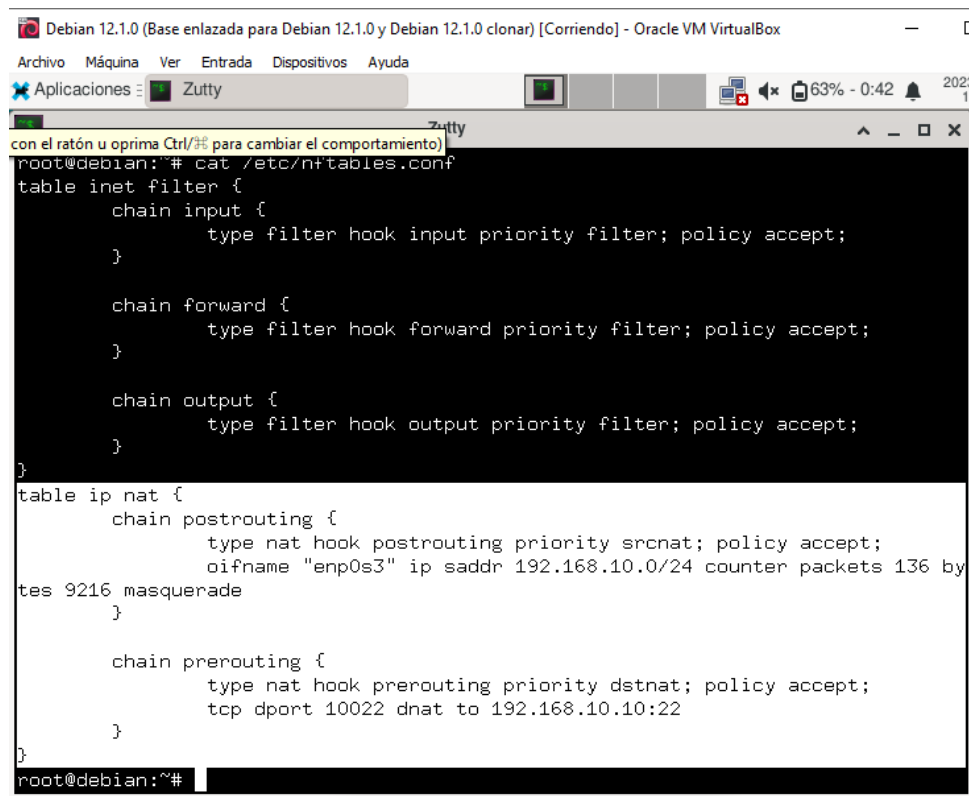
Hay que configurar también el servidor DNS, eso se hace en el archivo /etc/resolv.conf. Debe quedar así:



```
root@debian:~# cat /etc/resolv.conf
nameserver 8.8.8.8
root@debian:~#
```

En este caso estamos indicando a los servicios de red del sistema que, quien resuelve nuestros nombres de host.dominio es el servidor de google 8.8.8.8, puede ser cualquier otro. Este cambio no requiere el reinicio de los servicios de red. Es efectivo una vez se guarda el contenido del fichero.

La configuración nftables debe quedar así, sobre todo lo resaltado en blanco.

A screenshot of a virtual machine window titled "Debian 12.1.0 (Base enlazada para Debian 12.1.0 y Debian 12.1.0 clonar) [Corriendo] - Oracle VM VirtualBox". The window shows a terminal window with the prompt "root@debian:~#". The user has run the command "cat /etc/nftables.conf", and the output is displayed in the terminal. The configuration defines two tables: "inet filter" and "ip nat". The "inet filter" table has three chains: "input", "forward", and "output", each with a "filter" hook and "policy accept". The "ip nat" table has two chains: "postrouting" and "prerouting". The "postrouting" chain has a "nat" hook with "srcnat" priority, "policy accept", and a rule to masquerade traffic from the "enp0s3" interface. The "prerouting" chain has a "nat" hook with "dstnat" priority, "policy accept", and a rule to redirect traffic from port 10022 to 192.168.10.10:22.

```
con el ratón u oprima Ctrl/⇧ para cambiar el comportamiento)
root@debian:~# cat /etc/nftables.conf
table inet filter {
    chain input {
        type filter hook input priority filter; policy accept;
    }

    chain forward {
        type filter hook forward priority filter; policy accept;
    }

    chain output {
        type filter hook output priority filter; policy accept;
    }
}

table ip nat {
    chain postrouting {
        type nat hook postrouting priority srcnat; policy accept;
        oifname "enp0s3" ip saddr 192.168.10.0/24 counter packets 136 bytes 9216 masquerade
    }

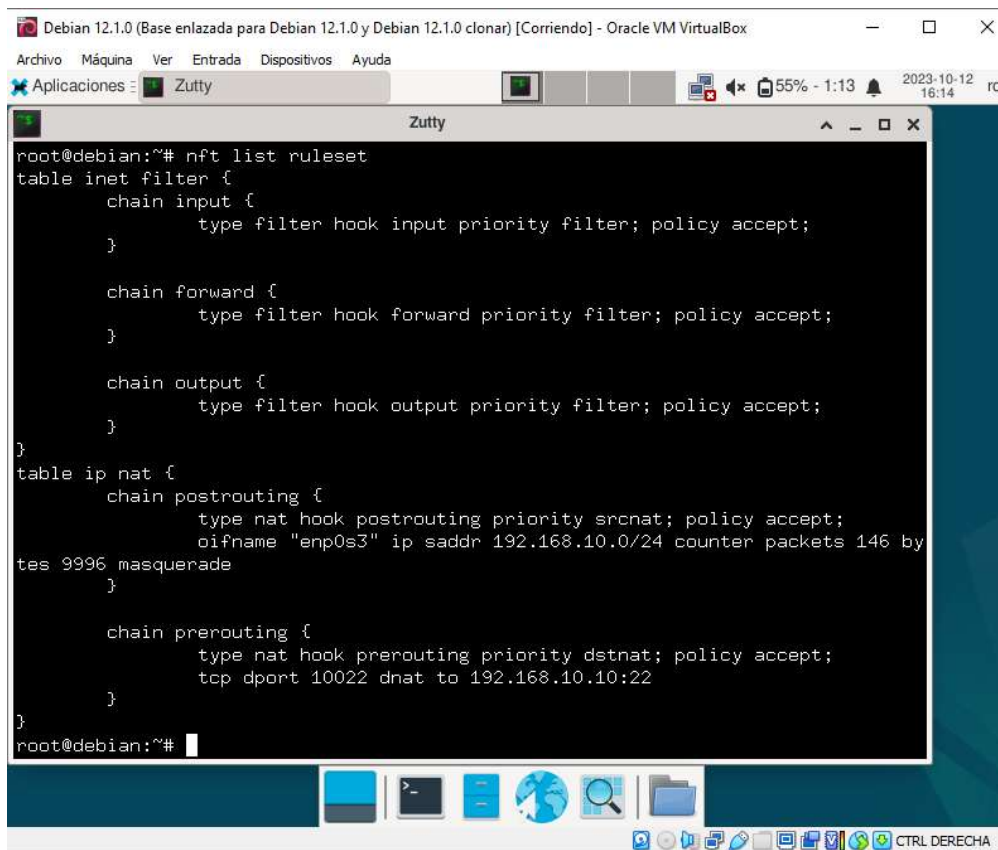
    chain prerouting {
        type nat hook prerouting priority dstnat; policy accept;
        tcp dport 10022 dnat to 192.168.10.10:22
    }
}
root@debian:~#
```

Si no esta así podéis editar directamente el archivo `/etc/nftables.conf` y reiniciar el servicio para que coja los cambios:

systemctl restart nftables

Podéis comprobar que todo está OK utilizando el comando:

nft list ruleset



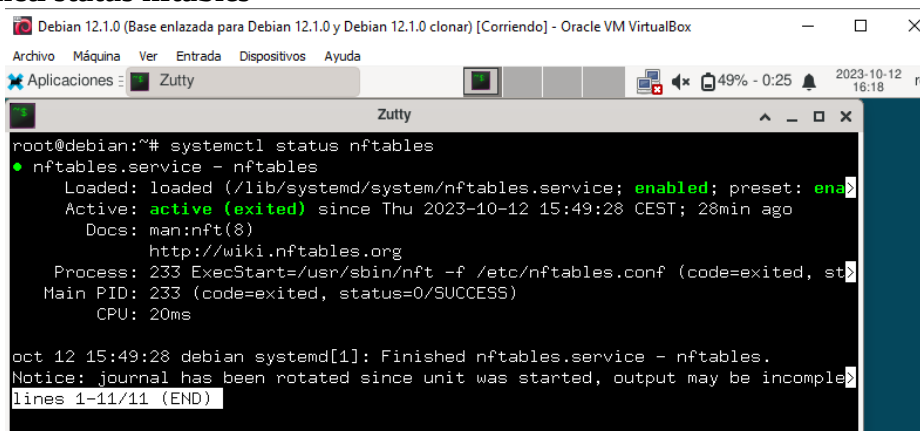
```
root@debian:~# nft list ruleset
table inet filter {
    chain input {
        type filter hook input priority filter; policy accept;
    }

    chain forward {
        type filter hook forward priority filter; policy accept;
    }

    chain output {
        type filter hook output priority filter; policy accept;
    }
}
table ip nat {
    chain postrouting {
        type nat hook postrouting priority srcnat; policy accept;
        oifname "enp0s3" ip saddr 192.168.10.0/24 counter packets 146 by
tes 9996 masquerade
    }

    chain prerouting {
        type nat hook prerouting priority dstnat; policy accept;
        tcp dport 10022 dnat to 192.168.10.10:22
    }
}
root@debian:~#
```

Comprobar que el servicio nftables está activo:
systemctl status nftables

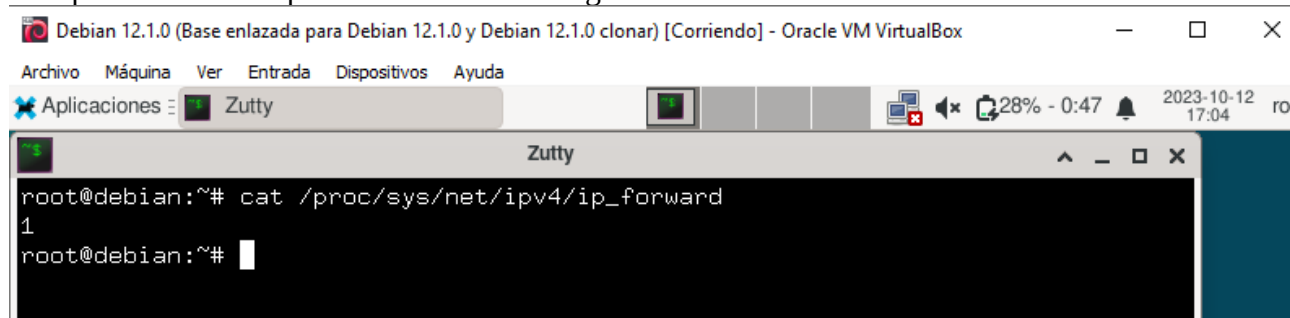


```
root@debian:~# systemctl status nftables
● nftables.service - nftables
   Loaded: loaded (/lib/systemd/system/nftables.service; enabled; preset: ena
   Active: active (exited) since Thu 2023-10-12 15:49:28 CEST; 28min ago
     Docs: man:nft(8)
           http://wiki.nftables.org
   Process: 233 ExecStart=/usr/sbin/nft -f /etc/nftables.conf (code=exited, st
   Main PID: 233 (code=exited, status=0/SUCCESS)
      CPU: 20ms

oct 12 15:49:28 debian systemd[1]: Finished nftables.service - nftables.
Notice: journal has been rotated since unit was started, output may be incomple
lines 1-11/11 (END)
```

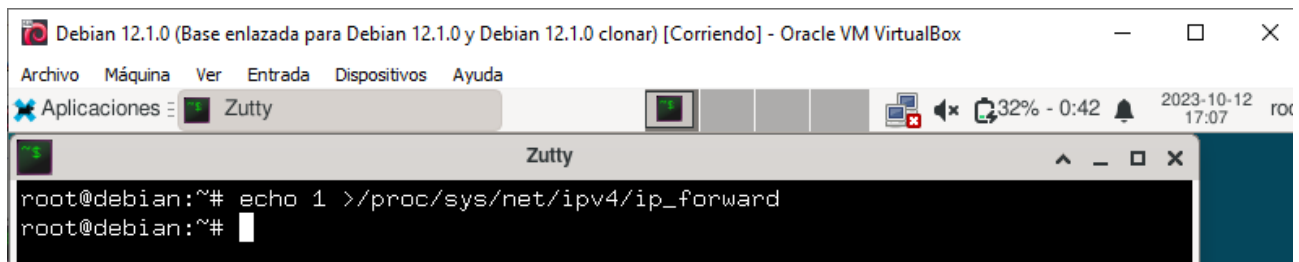
Para que se inicie automáticamente cada vez que iniciamos la máquina:
systemctl enables nftables

Comprueba también que el bit de forwarding está activo en el fichero:

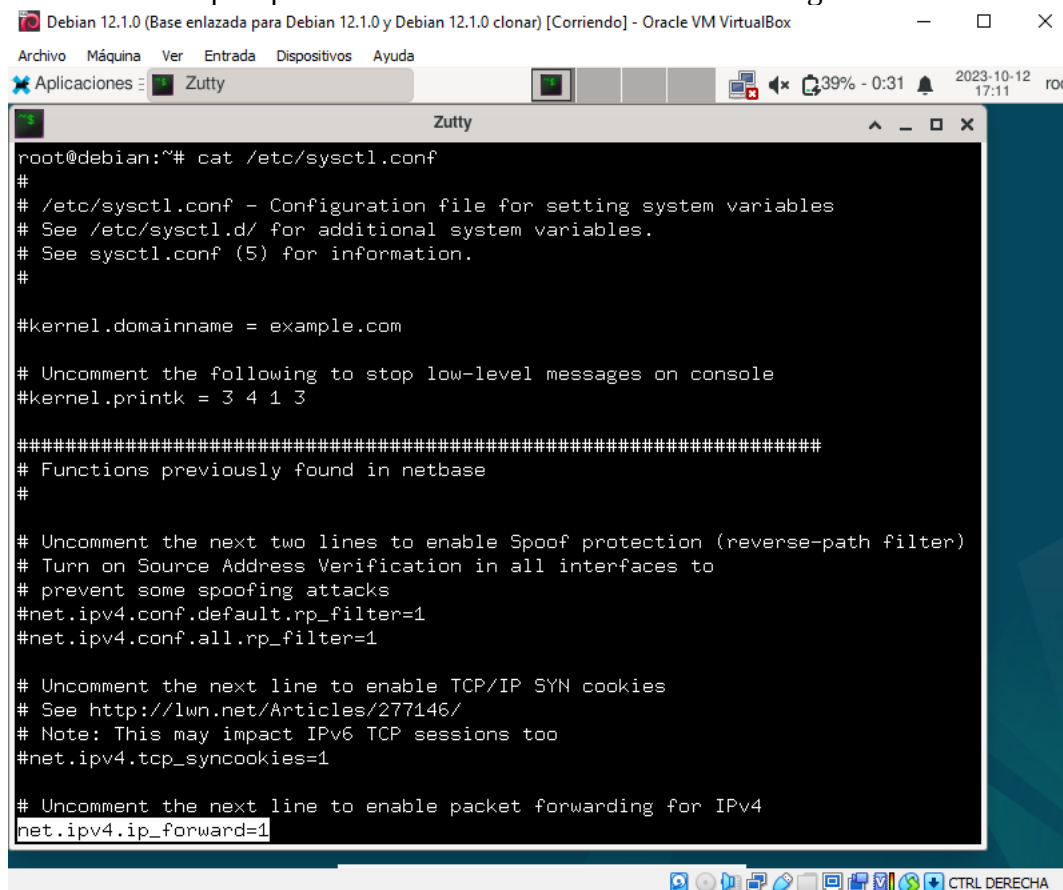


```
root@debian:~# cat /proc/sys/net/ipv4/ip_forward
1
root@debian:~#
```

Si no fuera así, actívalo. Cuidado después del “echo 1” hay un espacio:



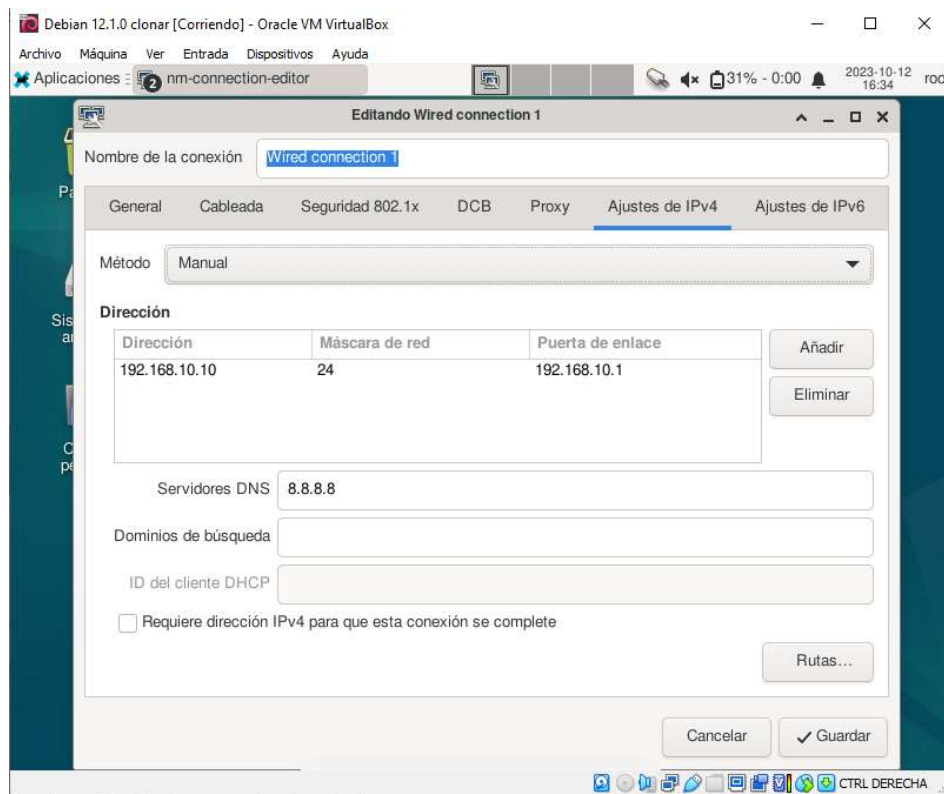
Si queremos que siga activo en futuros reinicios del sistema hay que editar el archivo descomentando la línea que aparece con fondo blanco al final de la imagen:



En la máquina clonada

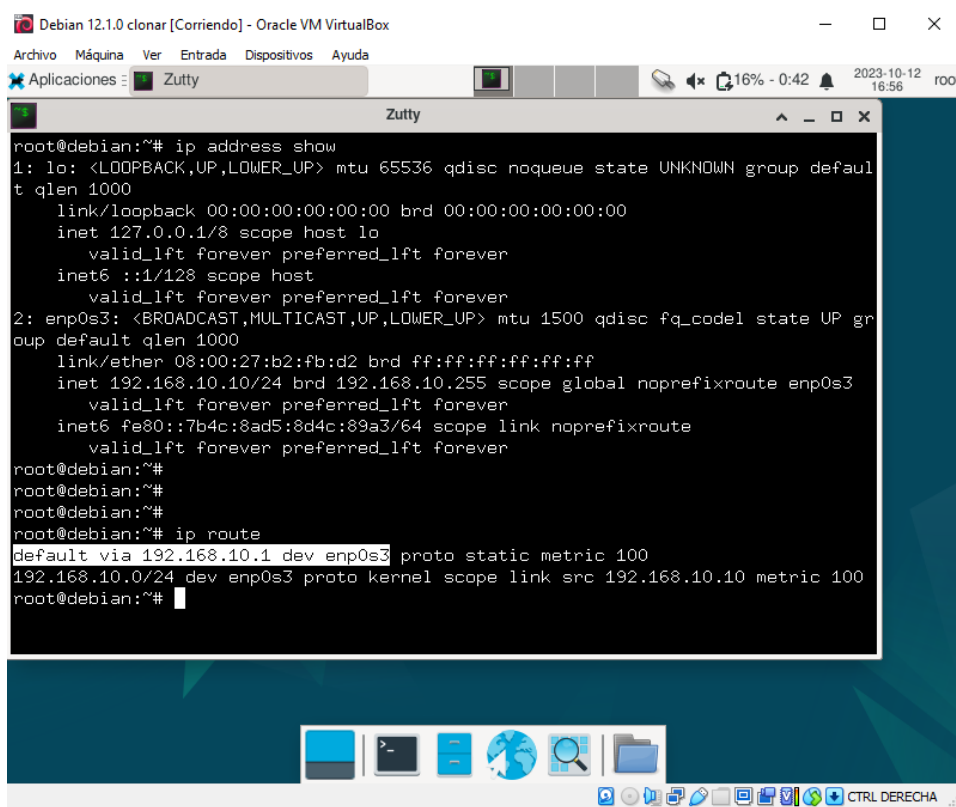
Vamos utilizar el entorno gráfico para configurar el adaptador de red, es otra forma de hacerlo.

Aquí no paramos Network Manager, corre por detrás del entorno gráfico y gracias a él la red se configurará como nosotros le indicamos en la pantalla gráfica:



Para que coja los cambios es necesario desactivar la casilla “Activar Red” y volver a activarla. Botón derecho del ratón encima del icono de red y clip.

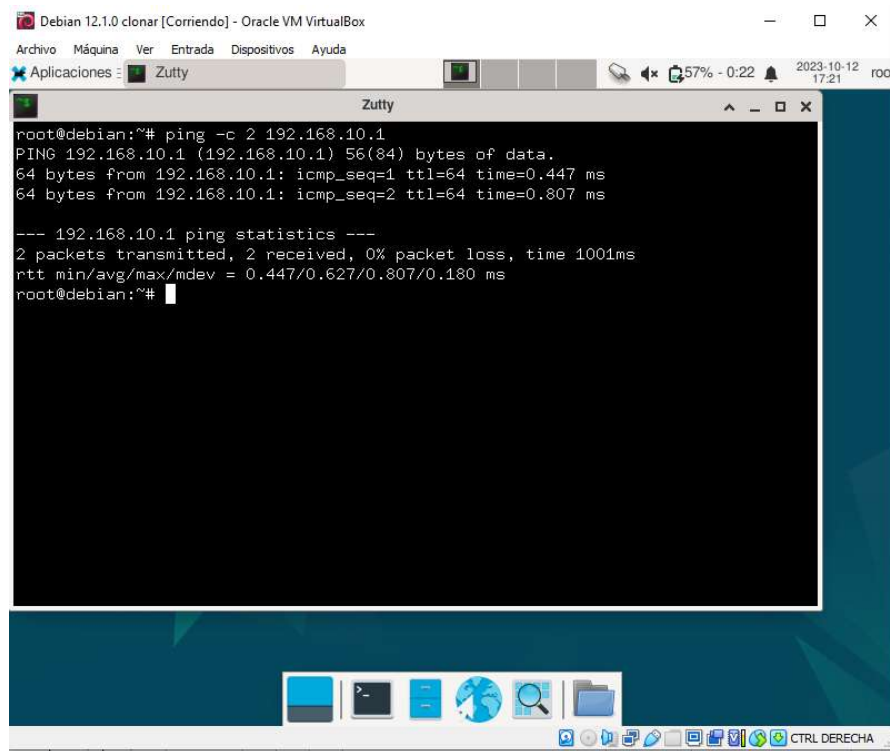
Abrimos un terminal y comprobamos que la configuración es correcta:



Además de la IP y máscara del adaptador enp0s3 (192.168.10.10/24) también he cogido bien la puerta de enlace por defecto. No es otra que la IP de la otra máquina Debian (192.168.10.1).

Comprobaciones de conectividad.

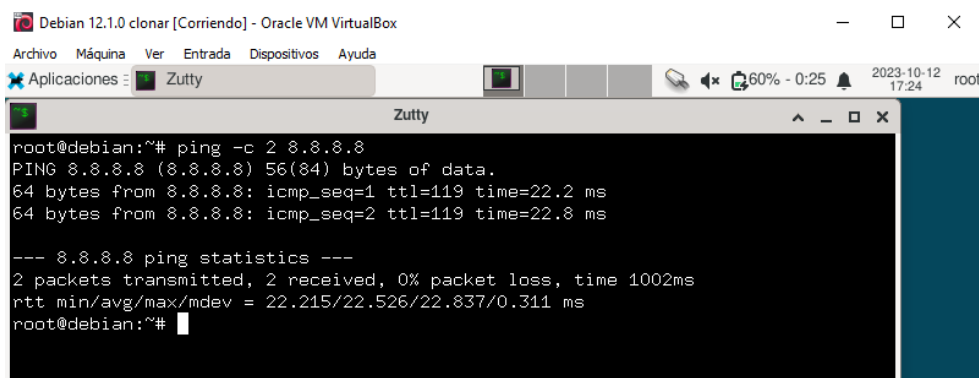
Desde Debian Clonar llegamos a la puerta de enlace(El otro Debian) a través de la red interna:



```
Debian 12.1.0 clonar [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Aplicaciones: Zutty
root@debian:~# ping -c 2 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.447 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.807 ms

--- 192.168.10.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.447/0.627/0.807/0.180 ms
root@debian:~#
```

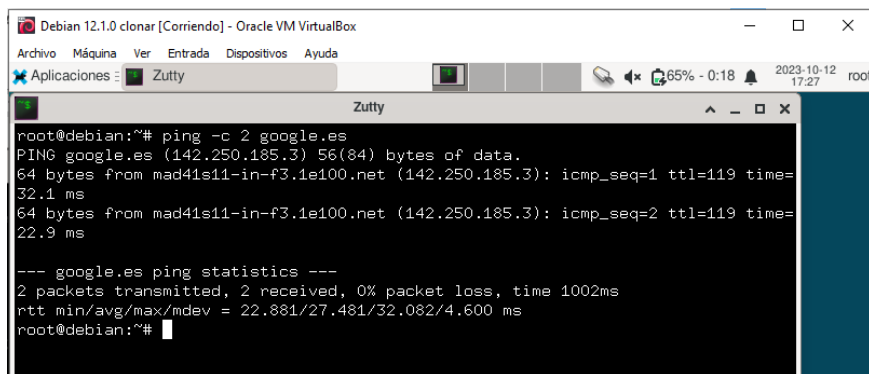
Desde Debian Clonar llegamos a Internet a través de nuestra puerta de enlace(El otro Debian):



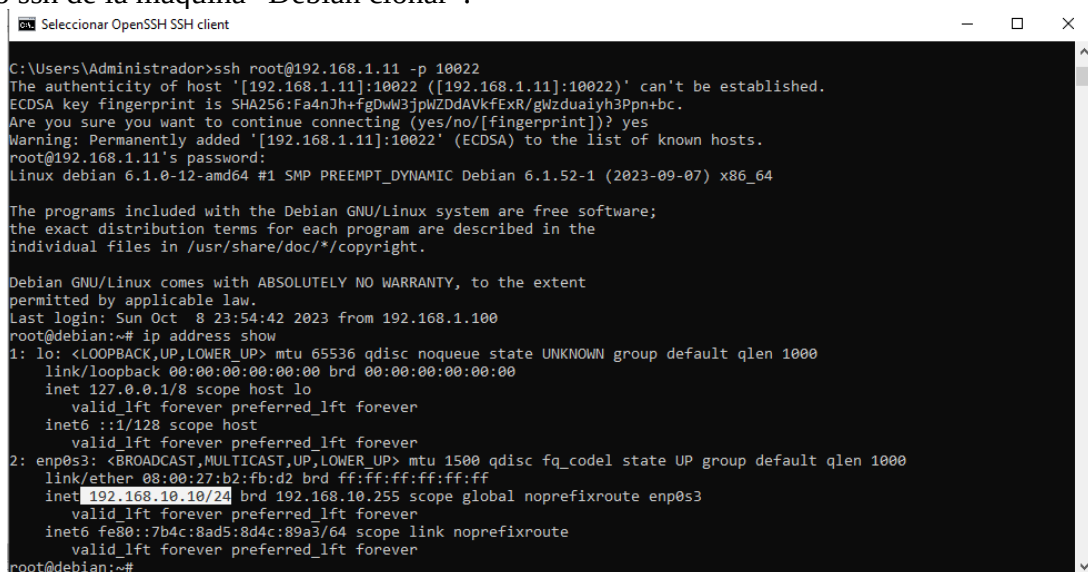
```
Debian 12.1.0 clonar [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Aplicaciones: Zutty
root@debian:~# ping -c 2 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=119 time=22.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=119 time=22.8 ms

--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 22.215/22.526/22.837/0.311 ms
root@debian:~#
```

Desde Debian Clonar comprobamos que el servidor DNS está bien configurado en /etc/resolv.conf :

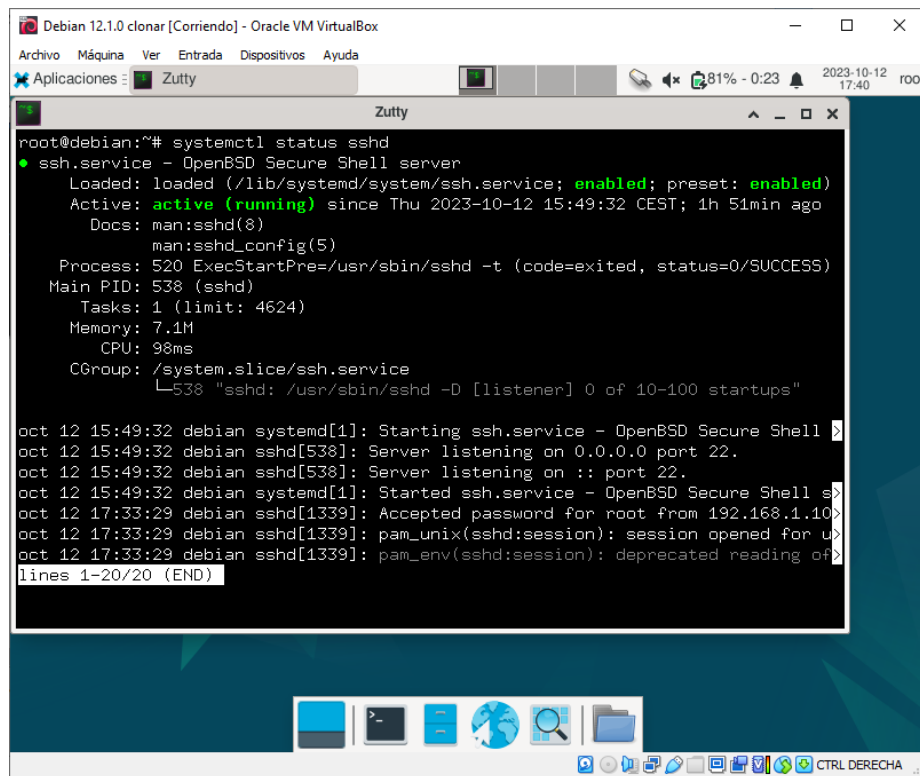


Comprobamos que desde una línea de comando(CMD) del PC anfitrión nos podemos conectar al servicio ssh de la máquina “Debian clonar”:



La IP en color de fondo blanco demuestra que estamos en Debian clonar.
Sino consigues esta conexión ssh, mira si tienes instalado el servicio ssh en la máquina Debian Clonar:

systemctl status sshd



```
root@debian:~# systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Thu 2023-10-12 15:49:32 CEST; 1h 51min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 520 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 538 (sshd)
      Tasks: 1 (limit: 4624)
     Memory: 7.1M
        CPU: 98ms
   CGroup: /system.slice/ssh.service
           └─538 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

oct 12 15:49:32 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell
oct 12 15:49:32 debian sshd[538]: Server listening on 0.0.0.0 port 22.
oct 12 15:49:32 debian sshd[538]: Server listening on :: port 22.
oct 12 15:49:32 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell s
oct 12 17:33:29 debian sshd[1339]: Accepted password for root from 192.168.1.10
oct 12 17:33:29 debian sshd[1339]: pam_unix(sshd:session): session opened for u
oct 12 17:33:29 debian sshd[1339]: pam_env(sshd:session): deprecated reading of
lines 1-20/20 (END)
```

Sino esta instalado, instalalo:

apt install openssh-server

Y configuramos el servicio para que el usuario root pueda acceder utilizando su password editando el archivo `/etc/ssh/sshd_config`(las líneas en blanco):

```
#PermitRootLogin prohibit-password
PermitRootLogin yes
```

y

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
```

Reinicia el servicio para que los cambios se hagan efectivos:

systemctl restart sshd