

DEAW -DW2

Reto #1

Router SSH
29/09/2020

Aketza Egusquiza
Iker Navarro
Aritz Romo

Contenido

Introducción:.....	1
Esquema de red:.....	2
Debian 9 - router instalación:.....	3
Instalaciones iniciales:.....	3
Configuración de red:.....	4
Configuración ssh:.....	8
Xubuntu 20.04 – configuración:.....	9
Pruebas:.....	11
Bibliografía:	17

Introducción:

En este reto se busca instalar y configurar un router con SSH mediante el cual se pueda conectar nuestra red interna con la red de la clase y a su vez a otras redes internas de otros compañeros de clase. Además se podrá acceder al router via SSH mediante un usuario en concreto.

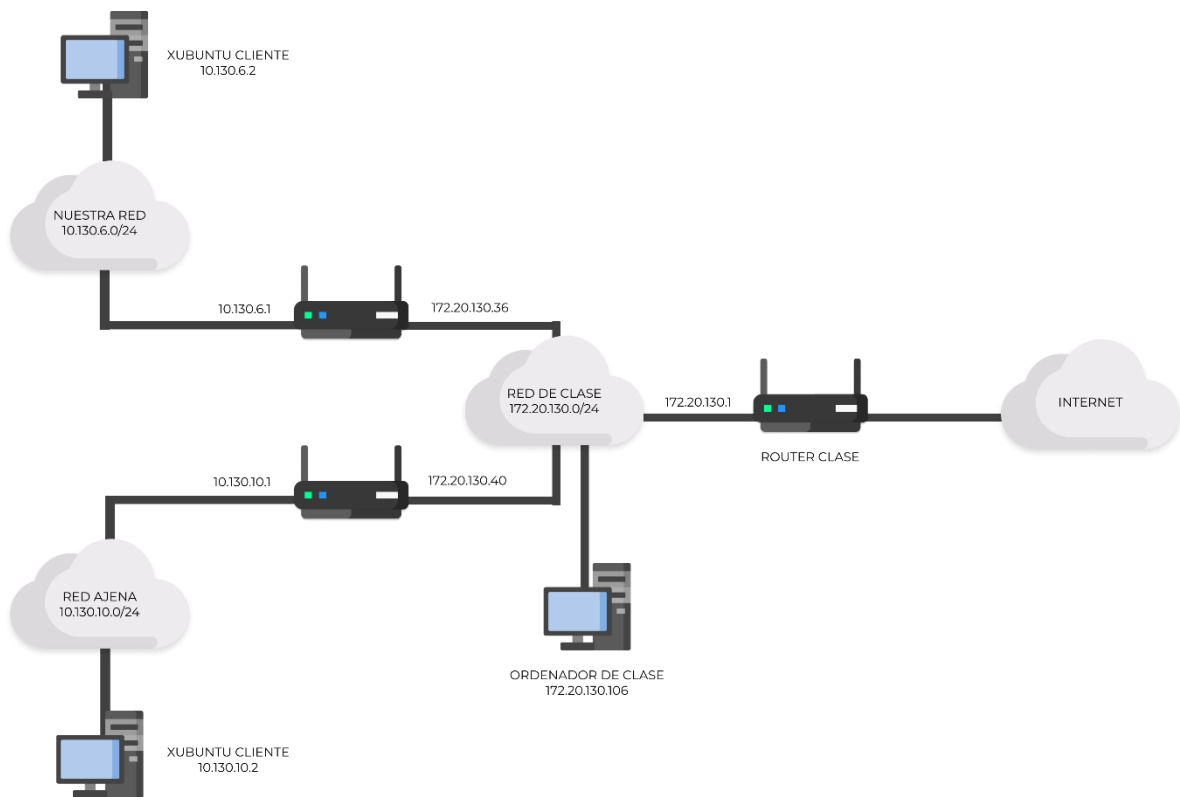
Para configurar las diferentes máquinas, hemos utilizado el software de virtualización VirtualBox.

Sistemas operativos utilizados:

Router: Debian 9

Cliente: Xubuntu 20.04

Esquema de red:



Debian 9 - router instalación:

Como hemos mencionado en la introducción, utilizaremos una máquina virtual con debían 9 para redirigir las peticiones de los equipos de nuestra red interna a la red de clase y viceversa. Además, necesitaremos poder conectarnos en remoto a esta máquina mediante SSH.

Instalaciones iniciales:

Después de realizar la instalación del sistema operativo, nos faltarán bastantes utilidades básicas que probablemente necesitemos. Para ello instalaremos los siguientes programas:

NOTA!

En este apartado de la instalación/configuración del router, asumiremos que todos los comandos están siendo ejecutados como root.

Para iniciar sesión como root: su -

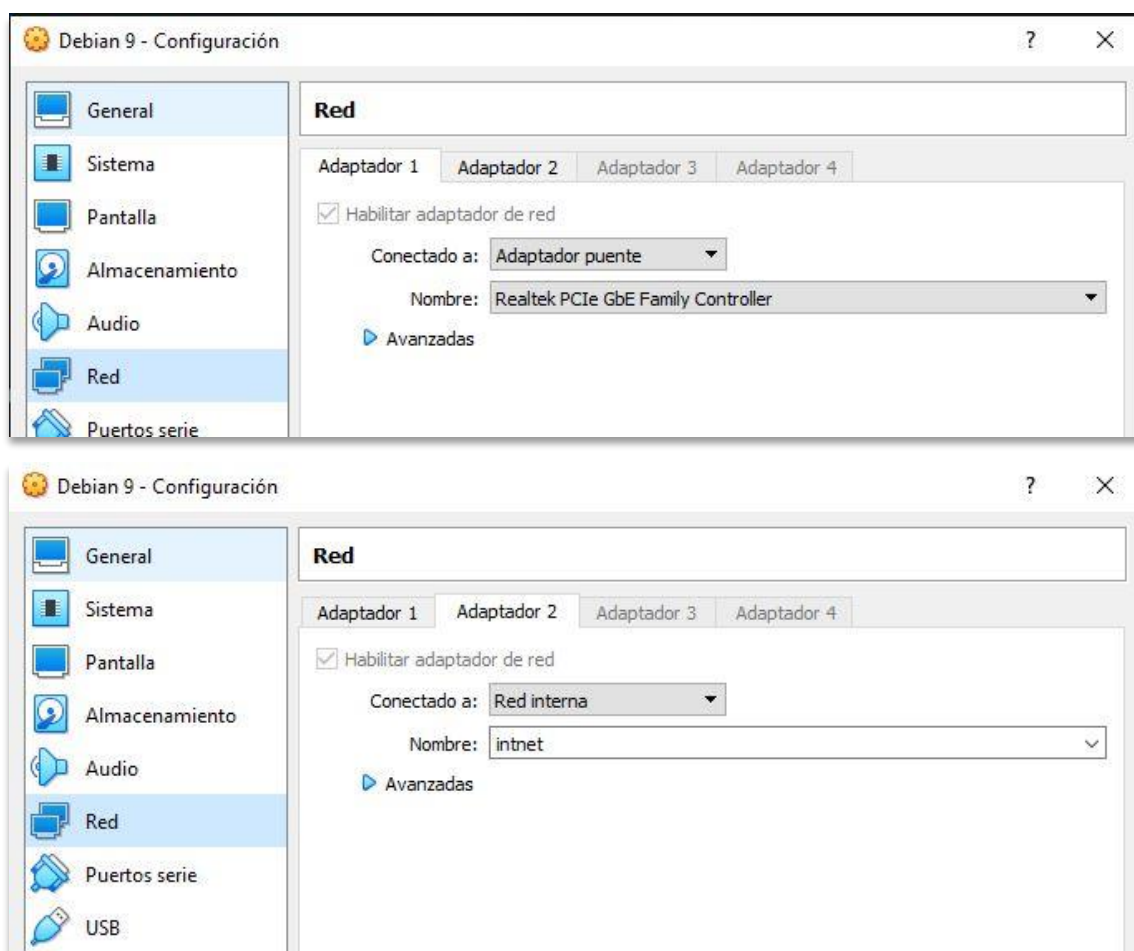
```
# apt-get install inetutils-tools inetutils-ping net-tools
```

Con estas instalaciones tendremos acceso a utilidades básicas de red como ping e ifconfig.

Configuración de red:

Hecho esto, tendremos que comenzar con la configuración de la red del equipo, pero para ello primero tendremos que acceder a la configuración de VirtualBox y añadir un nuevo adaptador de red.

Un adaptador (el que esté conectado a la red de clase) tendrá que estar configurado como adaptador puente, y la otra interfaz tendrá que estar en red interna.



Por defecto, Debian utiliza Network Manager para configurar la red automáticamente. Para nuestros propósitos necesitaremos deshabilitar esta funcionalidad y añadir la configuración de red de manera manual.

Para deshabilitar Network Manager y evitar que se reactive al reiniciar el sistema:

```
# systemctl stop NetworkManager
# systemctl disable NetworkManager
```

Ahora podremos configurar la red de manera manual y que se aplique correctamente.

Todas las configuraciones de las interfaces de red se harán en el archivo `/etc/network/interfaces`.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
source /etc/network/interfaces.d/*
```

```
# The Loopback network interface
auto lo
iface lo inet loopback
```

```
auto enp0s3
iface enp0s3 inet static
    address 172.20.130.136
    netmask 255.255.255.0
    gateway 172.20.130.1
    dns-nameservers 194.30.0.1 8.8.8.8
```

```
auto enp0s8
iface enp0s8 inet static
    address 10.130.6.1
    netmask 255.255.255.0
```

```
up route add -net 10.130.10.0 netmask 255.255.255.0 gw 172.20.130.140
```

Cómo podemos observar, en este archivo estamos configurando 2 interfaces además de la interfaz generada por defecto "lo" (localhost).

La interfaz enp0s3 será la que esté unida a la red de clase (172.20.130.0/24) y por lo tanto además de una IP que no esté siendo utilizada, necesitará tener configurados el DNS y gateway de clase.

La interfaz enp0s8, apuntará a la red interna, que en nuestro caso será la 10.130.6.0/24,

La última línea añade la ruta para que nuestro router conozca la interfaz de red por la que tienen que ser redirigidas las peticiones que se dirijan a la red interna de otro grupo.

Concretamente, en esa última línea especificamos que las peticiones dirigidas a la red 10.130.10.0 tendrán que ser redireccionadas a la IP 172.20.130.140, que es la máquina router de otro equipo.

#NOTA

Los cambios que realicemos en este archivo no se aplicarán hasta que reiniciemos el equipo o ejecutemos los siguientes comandos.

```
# systemctl restart networking
```

Hecho esto tendremos la configuración de red lista, pero el equipo todavía no está actuando como un router. Para obtener este resultado crearemos un script que se ejecute al inicio del equipo.

```
#!/bin/bash
```

```
### BEGIN INIT INFO
# Provides:          fw.sh
# Required-Start:    $all
# Required-Stop:     $all
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Firewall
# Description: Establece el firewall en este router
### END INIT INFO

#Activar enrutamiento
echo 1 > /proc/sys/net/ipv4/ip_forward

#Fuerza el vaciado de reglas del firewall iptables
iptables -F
```



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

```
#Enmascarar la salida a Internet desde la red interna
```

```
iptables -t nat -A POSTROUTING -s 10.130.6.1/24 -o enp0s3 -  
j MASQUERADE
```

Este archivo llamado fw.sh se encarga de 3 configuraciones principalmente:

Habilitar el redireccionamiento de peticiones para que actúe como un router, desactivar cualquier regla de firewall que pudiese estar configurada y finalmente habilitar NAT (Network Address Translation).

Mediante NAT enmascaramos todas las peticiones que sean originarias del adaptador en0s3 con la IP del router, para que las respuestas puedan ser recibidas por sus emisores iniciales.

Finalmente moveremos el archivo y le daremos los permisos necesarios (de ejecución) para que nuestro script se ejecute en el inicio del equipo.

```
# mv /home/debian/fw.sh /etc/init.d/  
# chmod +x /etc/init.d/fw.sh  
# update-rc.d fw.sh defaults
```

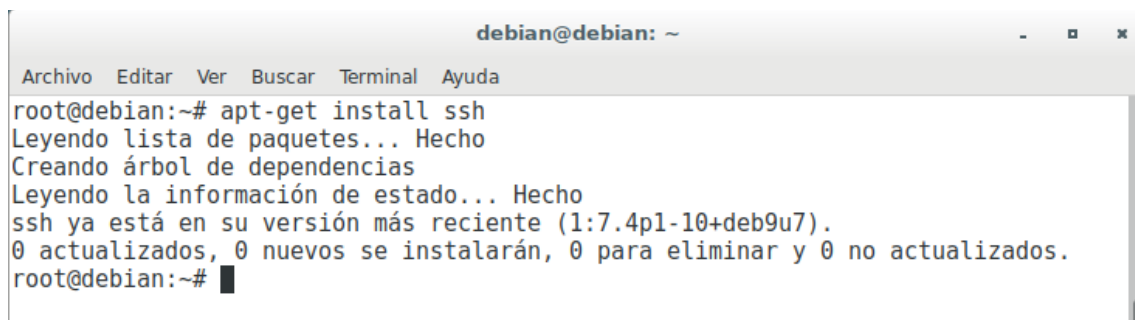
Para aplicar todos los cambios realizados y comprobar que todo funcione correctamente, tendremos que reiniciar el equipo para que se aplique el script creado.

Configuración ssh:

Procederemos con la instalación del servicio SSH, para ello ejecutamos los siguientes comandos.

```
# su -  
# apt-get install ssh
```

En este caso, Debian viene con el servicio SSH instalado por defecto.



```
debian@debian: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
root@debian:~# apt-get install ssh  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
ssh ya está en su versión más reciente (1:7.4p1-10+deb9u7).  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.  
root@debian:~#
```

Después procedemos a la configuración; para ello, tendremos que editar el archivo que contiene la configuración del servicio `/etc/ssh/sshd_config`.

Descomentaremos la línea 13 y modificamos el puerto 22 por el 222 "Port 222".

Para dar acceso únicamente al usuario `dw2` tendremos que añadir una nueva línea al final del archivo "AllowUsers `dw2`".

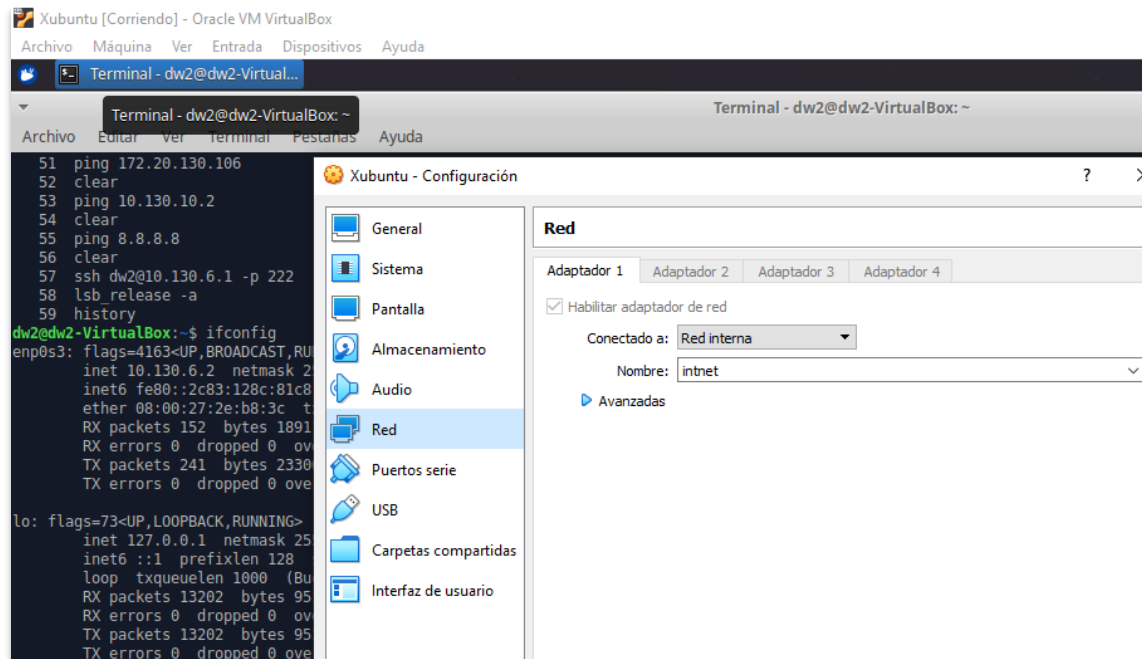
El usuario `dw2` es un usuario que previamente hemos creado en la máquina del router ejecutando el comando "adduser `dw2`".

Para que los cambios surtan efecto reiniciamos el servicio SSH con el comando "systemctl restart ssh".

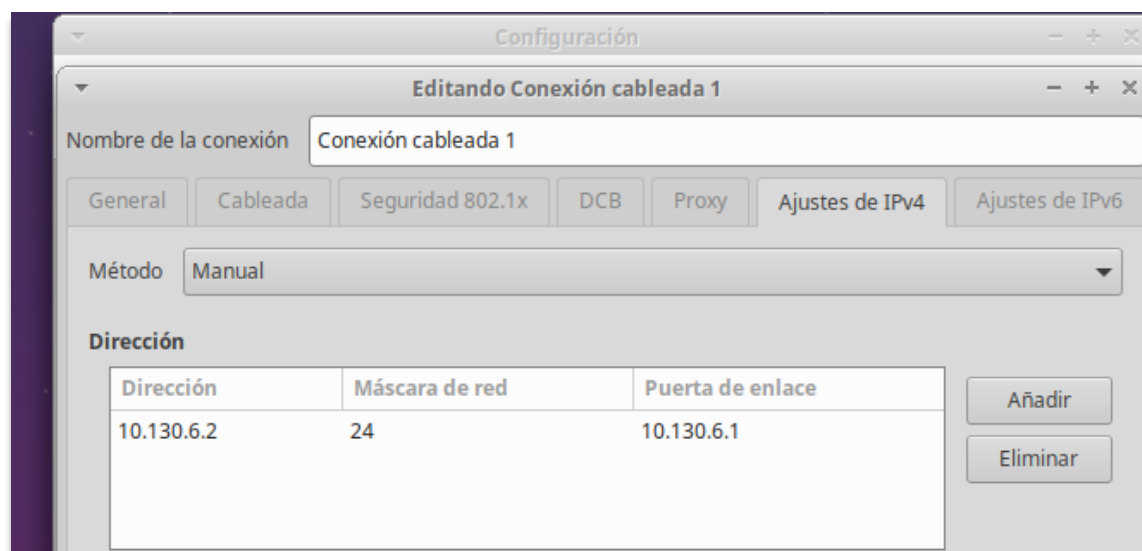
Con esta configuración solamente será posible la conexión a través del puerto 222 y accediendo como el usuario `dw2`.

Xubuntu 20.04 – configuración:

Para la máquina cliente hemos utilizado una máquina con Xubuntu 20.04, en la cual hemos configurado su tarjeta de red en VirtualBox como red interna.



Después, hemos configurado los parámetros de la red para que la máquina se encuentre en la misma red interna que el router. Hemos editado los parámetros de forma gráfica.



Además tendremos que instalar net-tools, ya que no vienen instaladas por defecto y no podemos utilizar comandos como ifconfig.

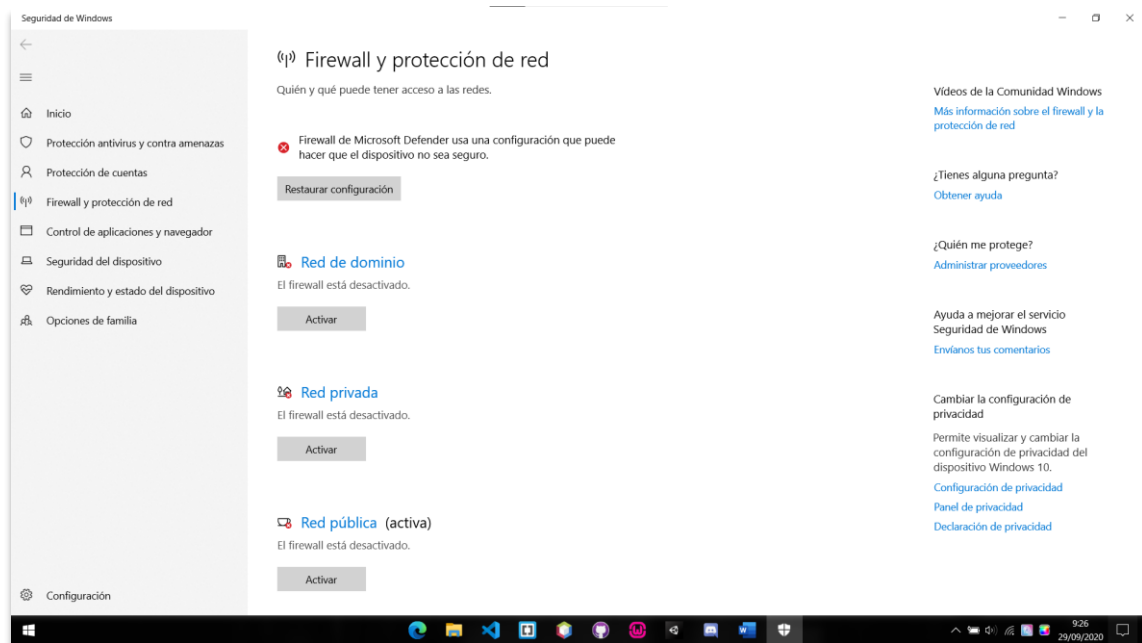
apt-get install net-tools

Una vez configurada la red en el cliente, y añadida la ruta en el router, podemos proceder a realizar las pruebas pertinentes para comprobar que todo funciona correctamente.

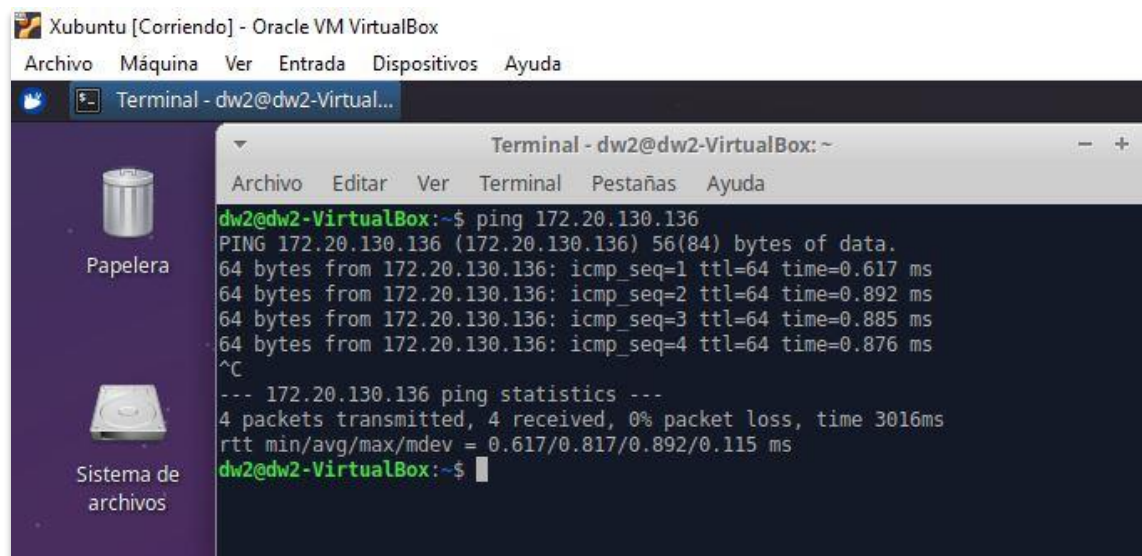
Pruebas:

#NOTA

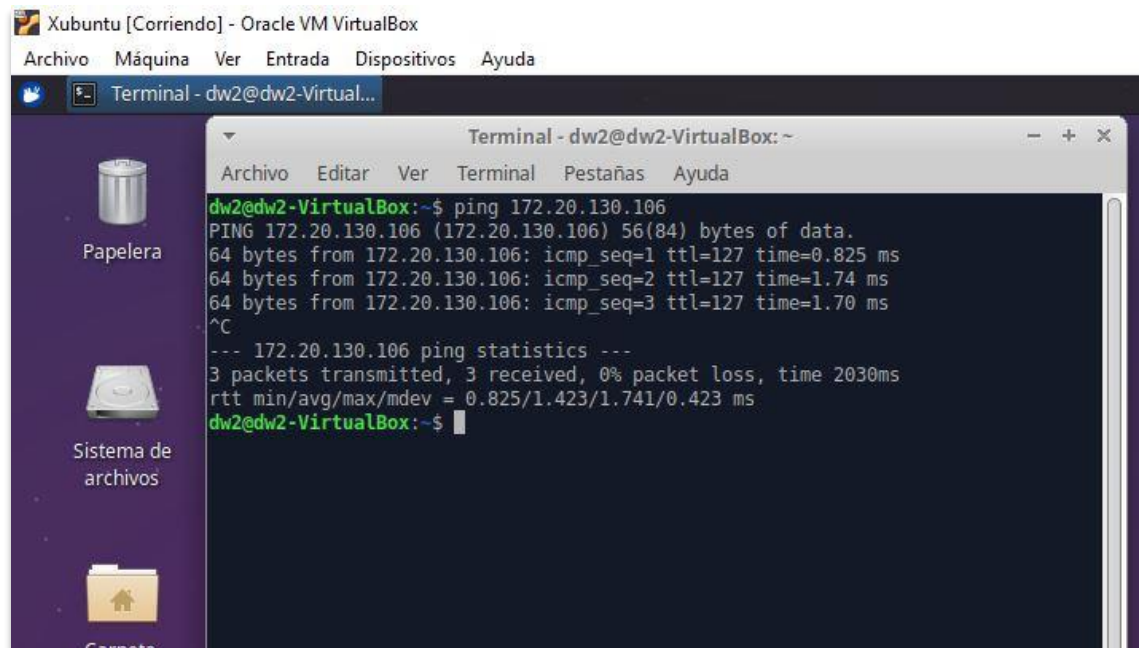
Antes de realizar acciones entre las máquinas, es conveniente desactivar el firewall de Windows desde el nuevo panel de control, ya que alguna prueba podría fallar.



Realizamos ping desde el cliente al router:



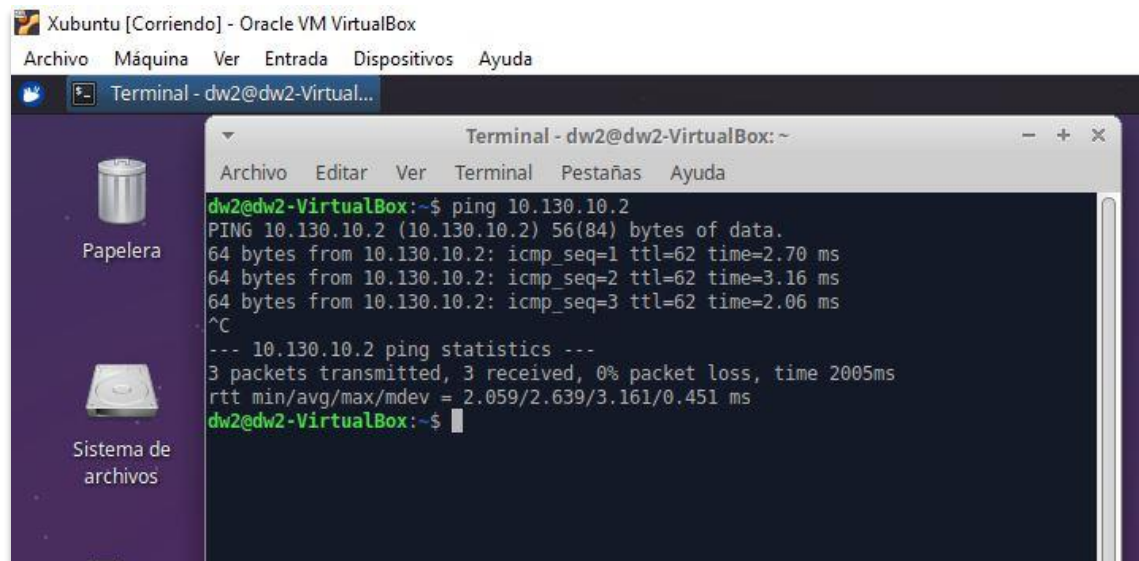
Realizamos ping desde el cliente a una máquina de la red de clase:



The screenshot shows a terminal window titled "Terminal - dw2@dw2-VirtualBox: ~" within a virtual machine environment. The terminal output shows a successful ping command to the IP address 172.20.130.106. The output includes the command, the ping statistics, and the results of three successful pings.

```
dw2@dw2-VirtualBox:~$ ping 172.20.130.106
PING 172.20.130.106 (172.20.130.106) 56(84) bytes of data:
64 bytes from 172.20.130.106: icmp_seq=1 ttl=127 time=0.825 ms
64 bytes from 172.20.130.106: icmp_seq=2 ttl=127 time=1.74 ms
64 bytes from 172.20.130.106: icmp_seq=3 ttl=127 time=1.70 ms
^C
--- 172.20.130.106 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.825/1.423/1.741/0.423 ms
dw2@dw2-VirtualBox:~$
```

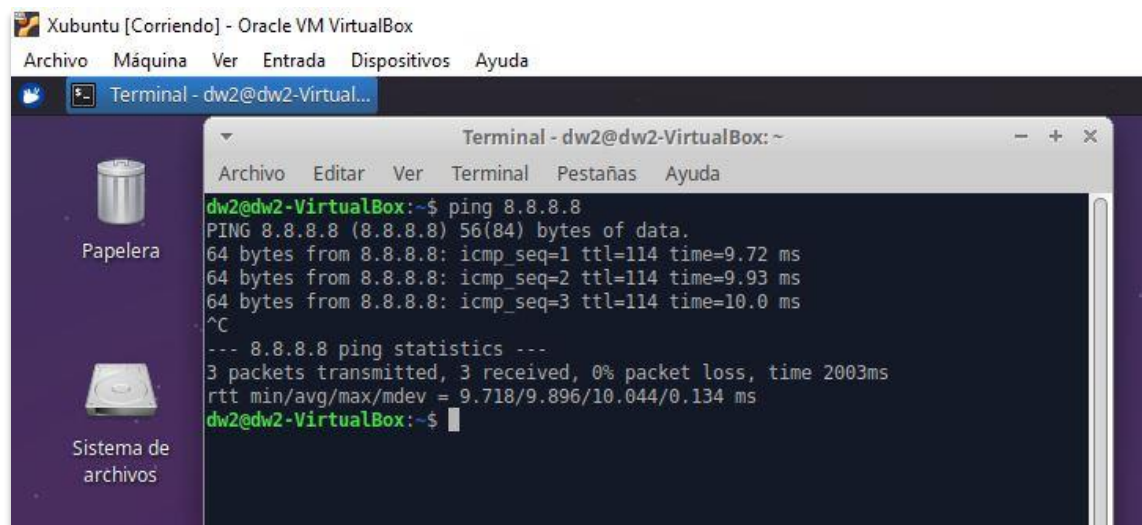
Realizamos ping desde el cliente a una máquina de otra red interna:



The screenshot shows a terminal window titled "Terminal - dw2@dw2-VirtualBox: ~" within a virtual machine environment. The terminal output shows a successful ping command to the IP address 10.130.10.2. The output includes the command, the ping statistics, and the results of three successful pings.

```
dw2@dw2-VirtualBox:~$ ping 10.130.10.2
PING 10.130.10.2 (10.130.10.2) 56(84) bytes of data:
64 bytes from 10.130.10.2: icmp_seq=1 ttl=62 time=2.70 ms
64 bytes from 10.130.10.2: icmp_seq=2 ttl=62 time=3.16 ms
64 bytes from 10.130.10.2: icmp_seq=3 ttl=62 time=2.06 ms
^C
--- 10.130.10.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 2.059/2.639/3.161/0.451 ms
dw2@dw2-VirtualBox:~$
```

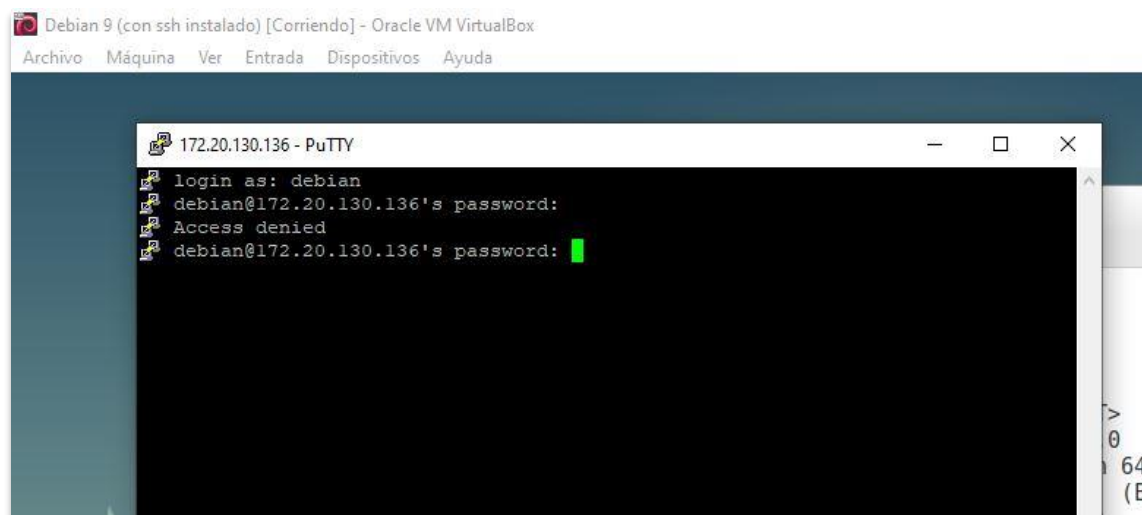
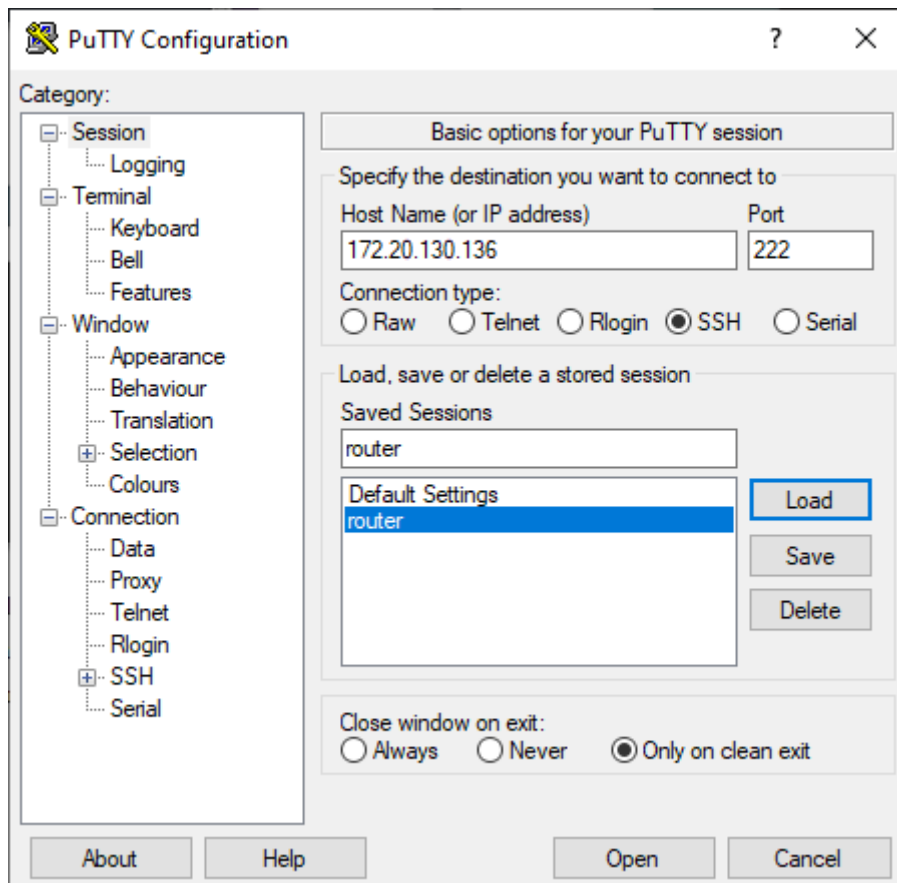
Realizamos ping desde el cliente a 8.8.8.8:

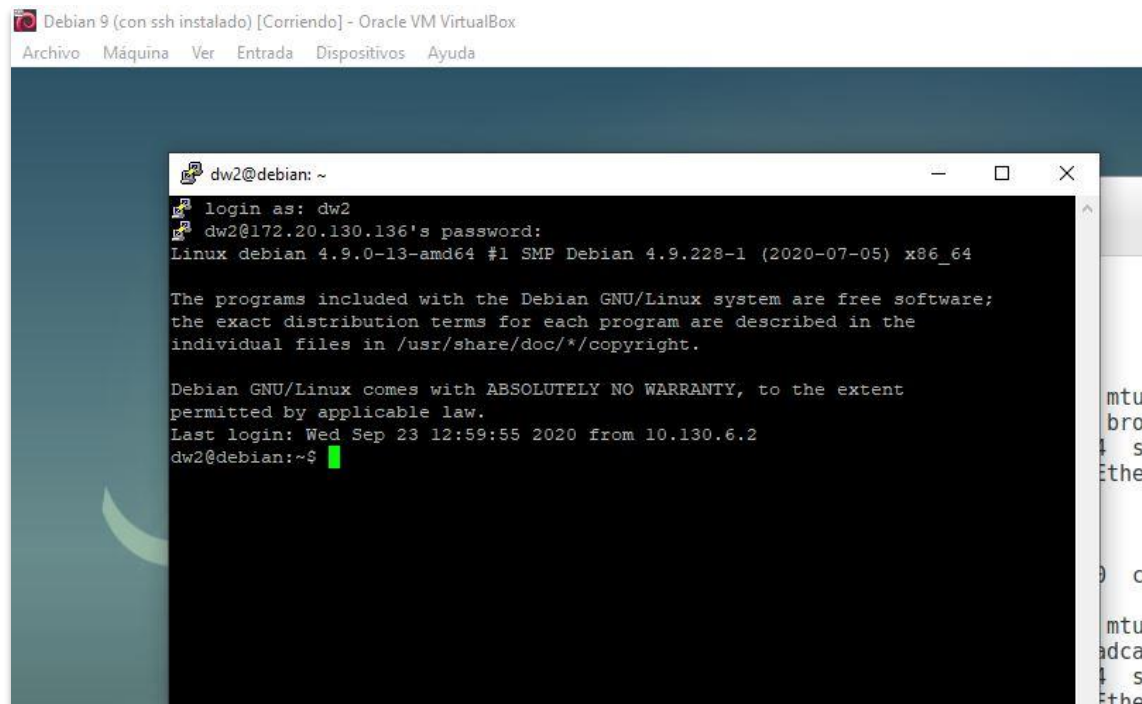


The screenshot shows a terminal window titled "Terminal - dw2@dw2-VirtualBox: ~" within an Oracle VM VirtualBox environment. The terminal displays the output of a ping command to 8.8.8.8. The output shows three successful pings with response times of 9.72 ms, 9.93 ms, and 10.0 ms. The statistics show 3 packets transmitted, 3 received, 0% packet loss, and a total time of 2003ms. The round-trip times (rtt) are 9.718/9.896/10.044/0.134 ms.

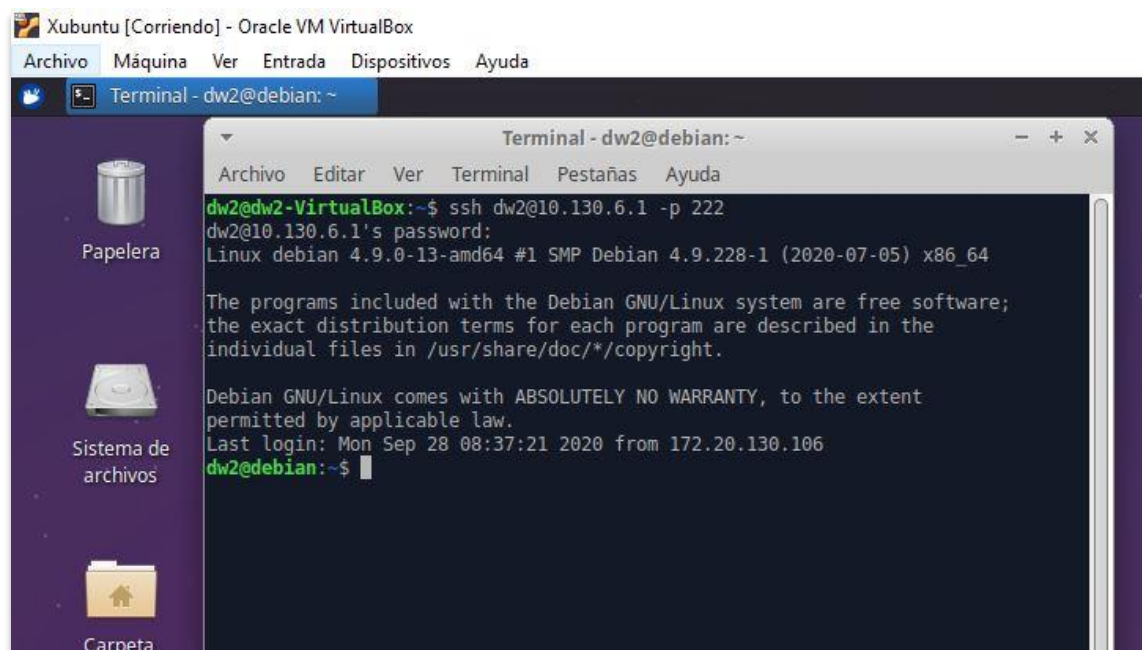
```
dw2@dw2-VirtualBox:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=9.72 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=9.93 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=10.0 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 9.718/9.896/10.044/0.134 ms
dw2@dw2-VirtualBox:~$
```

Conectamos al router vía SSH desde un equipo de la red del aula. Primero intentamos acceder con el usuario debían y comprobamos que no lo permite, ya que solo hemos permitido el acceso al usuario dw2. Seguidamente nos identificamos con el usuario dw2 y comprobamos que si tenemos acceso.

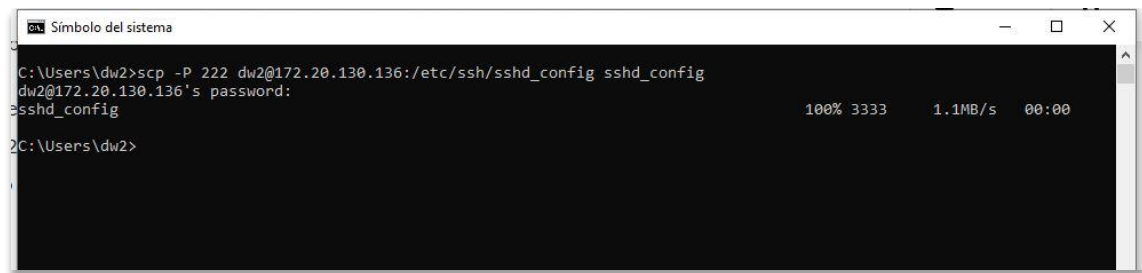




Conectamos al router via SSH desde el ordenador de nuestra red interna con Xubuntu instalado:



Copiamos el archivo de configuración de SSH mediante scp desde el router al equipo del aula:



```
C:\Users\dw2>scp -P 222 dw2@172.20.130.136:/etc/ssh/sshd_config sshd_config
dw2@172.20.130.136's password:
sshd_config 100% 3333 1.1MB/s 00:00
C:\Users\dw2>
```

Bibliografía:

- DOCUMENTACIÓN APORTADA EN MOODLE
- [O'REILLY.com](https://oreilly.com) (29/09/2020)
- [Arsys.es](https://arsys.es) (29/09/2020)