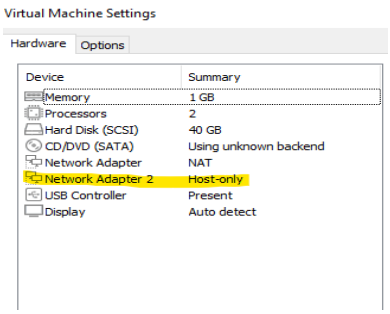
Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

SPD P01 – VPN

1. Creamos el entorno

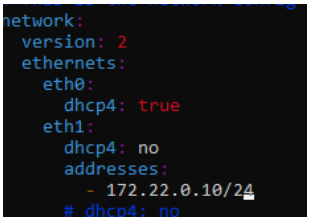
Empezamos descargando todo lo necesario de cada maquina, para esto tiene que estar las redes en modo NAT. (OpenVPN en PC1 y el Server VPN, easy-rsa en el Server y Apache en todas porque no)

Apagamos el Gateway y le metemos otra interfaz de red en Host-Only

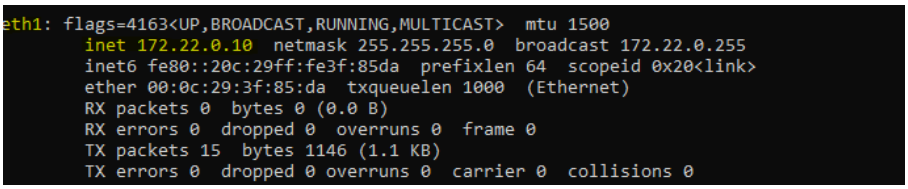


Despues encendemos y levantamos eth1 con '*sudo ip link set eth1 up*'

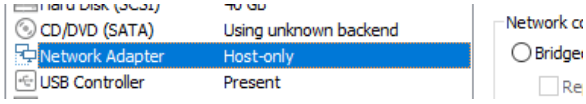
Y configuramos el direccionamiento del Gateway para eth1 ('*vim /etc/netplan/00-installer-config.yaml*'):



Aplicamos con '*sudo netplan apply*' y haciendo un ipconfig ya sale la direccion para eth1:



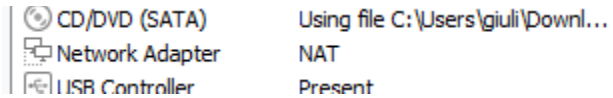
Cambiamos la NIC de PC1 para que este en Host-Only (se puede hacer en caliente):



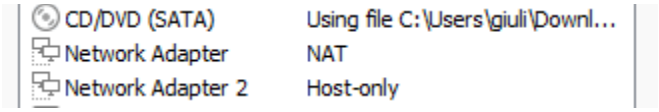
Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

Una vez tenemos todo descargado correctamente, ajustamos las tarjetas de red de la siguiente manera:

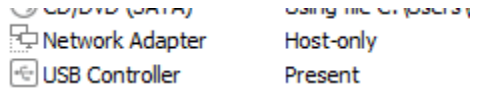
PC1 al estar en la red externa lo dejamos en modo NAT:



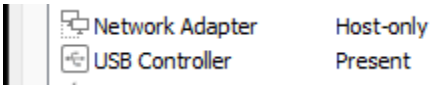
El **gateway** estará a su vez en la red externa y la red interna, por tanto, la tarjeta que esté en la red externa se configurará en modo NAT y la otra interna en modo H.O:



El **server**:



El **PC2**:



Y ponemos el direccionamiento correcto de PC1 y en todas las máquinas correspondientes como se muestra a continuación, todo esto siempre manipulando el fichero que se encuentra en (/etc/netplan/00-cloud-init.yaml) o como se llame el fichero en su máquina correspondiente:

PC1 (por favor, comentar el default gateway para simular que está en red externa):

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

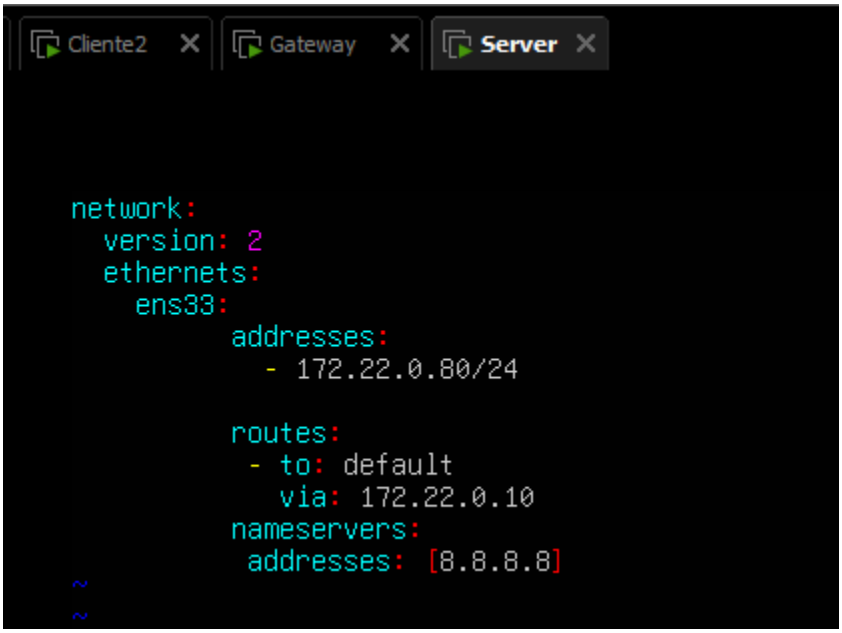
```
network:
  version: 2
  ethernets:
    ens33:
       addresses: [40.40.40.30/24]

       routes:
        - to: default
          via: 40.40.40.10
       nameservers:
         addresses: [8.8.8.8]
```

Ejecutar **(netplan apply)** para guardar los cambios en la configuración. Ahora seguimos la misma lógica en función el esquema de la práctica para el resto de las máquinas.

Gateway:

Server:

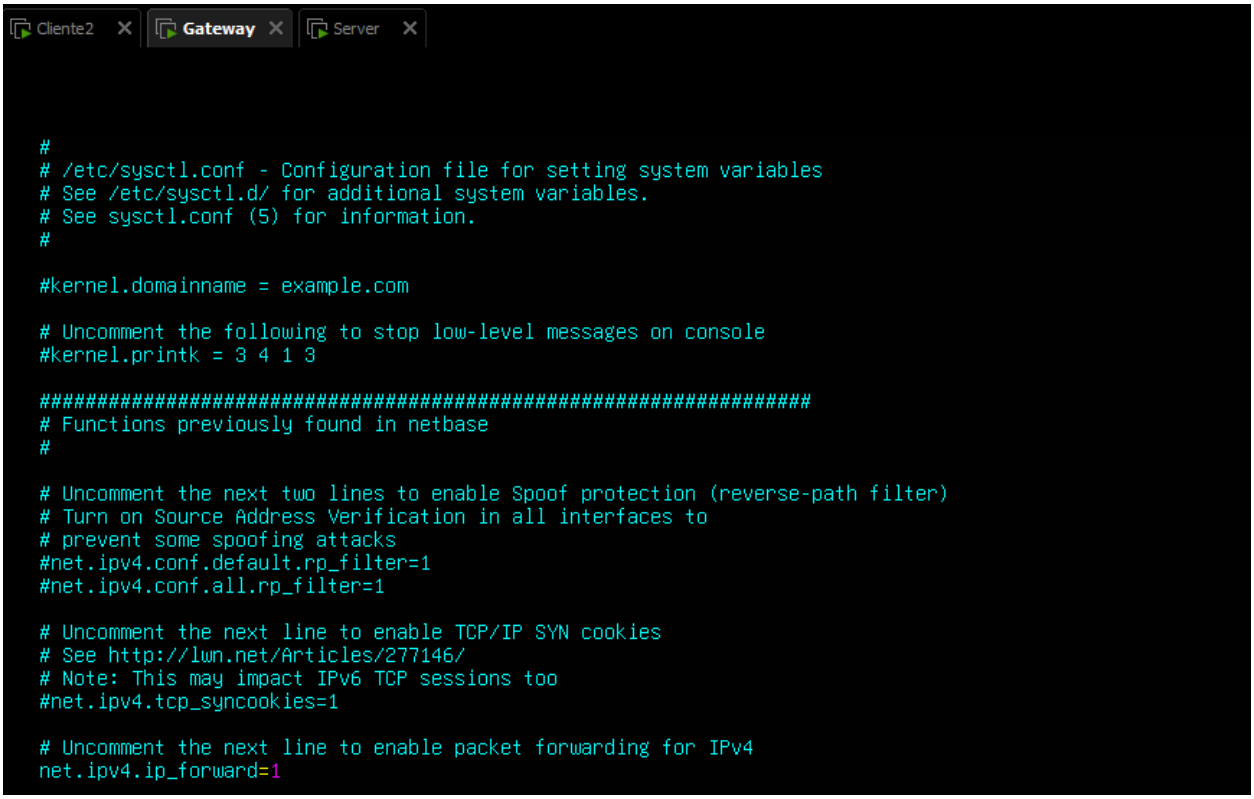
Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco



PC2:

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

Activar el forwarding en el gateway (ir al fichero /etc/sysctl.conf):



Descomentar el campo net.ipv4.ip_forward = 1. Para aplicar el cambio en la configuración con (**sudo sysctl –p**).

Ahora comprobamos la conectividad entre los equipos (naturalmente, sí que tendremos que ser capaces dentro de la red interna hacer ping a cualquier máquina dentro de la red interna).

Desde **PC2** al resto de máquinas:

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

```
PING 172.22.0.80 (172.22.0.80) 56(84) bytes of data.
64 bytes from 172.22.0.80: icmp_seq=1 ttl=64 time=1.36 ms
64 bytes from 172.22.0.80: icmp_seq=2 ttl=64 time=0.567 ms
64 bytes from 172.22.0.80: icmp_seq=3 ttl=64 time=0.576 ms
64 bytes from 172.22.0.80: icmp_seq=4 ttl=64 time=0.519 ms

--- 172.22.0.80 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3022ms
rtt min/avg/max/mdev = 0.519/0.754/1.356/0.347 ms
cliente2@cliente2:~$ ping -c 4 172.22.0.10
PING 172.22.0.10 (172.22.0.10) 56(84) bytes of data.
64 bytes from 172.22.0.10: icmp_seq=1 ttl=64 time=0.634 ms
64 bytes from 172.22.0.10: icmp_seq=2 ttl=64 time=0.664 ms
64 bytes from 172.22.0.10: icmp_seq=3 ttl=64 time=0.511 ms
64 bytes from 172.22.0.10: icmp_seq=4 ttl=64 time=0.503 ms

--- 172.22.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3032ms
rtt min/avg/max/mdev = 0.503/0.578/0.664/0.071 ms
cliente2@cliente2:~$ ping -c 4 40.40.40.10
PING 40.40.40.10 (40.40.40.10) 56(84) bytes of data.
64 bytes from 40.40.40.10: icmp_seq=1 ttl=64 time=0.682 ms
64 bytes from 40.40.40.10: icmp_seq=2 ttl=64 time=0.592 ms
64 bytes from 40.40.40.10: icmp_seq=3 ttl=64 time=0.568 ms
64 bytes from 40.40.40.10: icmp_seq=4 ttl=64 time=0.528 ms

--- 40.40.40.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3089ms
rtt min/avg/max/mdev = 0.528/0.592/0.682/0.056 ms
cliente2@cliente2:~$ ping 40.40.40.30
PING 40.40.40.30 (40.40.40.30) 56(84) bytes of data.
^C
--- 40.40.40.30 ping statistics ---
99 packets transmitted, 0 received, 100% packet loss, time 100293ms

cliente2@cliente2:~$
```

Desde **PC1**:

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

```
cliente1@cliente1:~$ ping -c 4 40.40.40.10
PING 40.40.40.10 (40.40.40.10) 56(84) bytes of data.
64 bytes from 40.40.40.10: icmp_seq=1 ttl=64 time=0.996 ms
64 bytes from 40.40.40.10: icmp_seq=2 ttl=64 time=0.603 ms
64 bytes from 40.40.40.10: icmp_seq=3 ttl=64 time=0.613 ms
64 bytes from 40.40.40.10: icmp_seq=4 ttl=64 time=0.581 ms

--- 40.40.40.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.581/0.698/0.996/0.172 ms
cliente1@cliente1:~$ ping -c 4 172.22.0.10
ping: connect: Network is unreachable
cliente1@cliente1:~$ ping -c 4 172.22.0.80
ping: connect: Network is unreachable
cliente1@cliente1:~$ ping -c 4 172.22.0.50
ping: connect: Network is unreachable
cliente1@cliente1:~$ _
```

Desde **server**:

```
server@server:~$ ping -c 4 172.22.0.50
PING 172.22.0.50 (172.22.0.50) 56(84) bytes of data.
64 bytes from 172.22.0.50: icmp_seq=1 ttl=64 time=0.635 ms
64 bytes from 172.22.0.50: icmp_seq=2 ttl=64 time=0.467 ms
64 bytes from 172.22.0.50: icmp_seq=3 ttl=64 time=0.510 ms
64 bytes from 172.22.0.50: icmp_seq=4 ttl=64 time=0.599 ms

--- 172.22.0.50 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3066ms
rtt min/avg/max/mdev = 0.467/0.552/0.635/0.067 ms
server@server:~$ ping -c 4 172.22.0.10
PING 172.22.0.10 (172.22.0.10) 56(84) bytes of data.
64 bytes from 172.22.0.10: icmp_seq=1 ttl=64 time=0.809 ms
64 bytes from 172.22.0.10: icmp_seq=2 ttl=64 time=0.557 ms
64 bytes from 172.22.0.10: icmp_seq=3 ttl=64 time=0.533 ms
64 bytes from 172.22.0.10: icmp_seq=4 ttl=64 time=0.543 ms

--- 172.22.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3081ms
rtt min/avg/max/mdev = 0.533/0.610/0.809/0.114 ms
server@server:~$ ping -c 4 40.40.40.10
PING 40.40.40.10 (40.40.40.10) 56(84) bytes of data.
64 bytes from 40.40.40.10: icmp_seq=1 ttl=64 time=0.756 ms
64 bytes from 40.40.40.10: icmp_seq=2 ttl=64 time=0.540 ms
64 bytes from 40.40.40.10: icmp_seq=3 ttl=64 time=0.621 ms
64 bytes from 40.40.40.10: icmp_seq=4 ttl=64 time=0.656 ms

--- 40.40.40.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3110ms
rtt min/avg/max/mdev = 0.540/0.643/0.756/0.077 ms
server@server:~$ ping -c 4 40.40.40.30
PING 40.40.40.30 (40.40.40.30) 56(84) bytes of data.
^C
--- 40.40.40.30 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2027ms
```

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

2. **Crear CA y configurar valores DH y claves/certificados del servidor y cliente:**

Para la creaciónde la CA, certificados y claves nos vamos al servidor y ejecutamos los siguientes comandos:

1. cd ~
2. /usr/share/easy-rsa/easyrsa init-pki
3. /usr/share/easy-rsa/easyrsa build-ca
4. /usr/share/easy-rsa/easyrsa gen-dh

Con esto generamos tres ficheros importantes:

- ca.crt: certificado raíz de la CA (se comparte con todos)
- ca.key: clave privada de la CA (se guarda en la CA)
- dh.pem: claves Diffie-Helmann para el intercambio de claves seguro

Creamos certificados para el servidor, cliente PC1 y la lista de certificados revocados:

1. /usr/share/easy-rsa/easyrsa build-server-full servidor nopass

2. /usr/share/easy-rsa/easyrsa build-client-full cliente-01 nopass

3. /usr/share/easy-rsa/easyrsa gen-crl

Opcionalmente, se puede generar una clave compartida para aumentar la seguridad, yo no lo he hecho:

- openvpn --genkey secret ta.key

Luego, movemos los certificados correspondientes donde les corresponde usando la siguiente lógica:

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

Archivo	Contenido	Utilizado por	Secreto
ca.key	Clave privada de la CA	Equipo de generación de certificados	Si
ca.crt	Certificado raíz de la CA	servidor y todos los clientes	No
dh.pem	Parámetros Diffie Hellman	Servidor	No
crl.pem	Certificados revocados	Servidor	No
servidor.crt	Certificado del servidor	Servidor	No
servidor.key	Clave priv. del servidor	Servidor	Si
cliente-01.crt	Certificado de cliente1	Cliente1	No
cliente-01.key	Clave priv. de Cliente1	Cliente1	Si

Para transferir los certificados y claves de manera segura usaremos **SCP**:

Archivos que corresponden a PC1, para ello ejecutar los siguientes comandos:

- scp /usr/share/easy-rsa/pki/issued/cliente-01.crt cliente1@40.40.40.30:/home/cliente1/
- scp /usr/share/easy-rsa/pki/private/cliente-01.key cliente1@40.40.40.30:/home/cliente1/
- scp /usr/share/easy-rsa/pki/ca.crt cliente1@40.40.40.30:/home/cliente1/

Luego, dentro del servidor creamos el siguiente directorio:

sudo mkdir -p /etc/openvpn/pki y copiamos cada certificado y llave relacionado con el servidor al nuevo directorio **pki.**

- **sudo cp ~/easy-rsa/pki/ca.crt /etc/openvpn/pki/**
- **sudo cp ~/easy-rsa/pki/issued/servidor.crt /etc/openvpn/pki/**
- **sudo cp ~/easy-rsa/pki/private/servidor.key /etc/openvpn/pki/**
- **sudo cp ~/easy-rsa/pki/dh.pem /etc/openvpn/pki/**
- **sudo cp ~/easy-rsa/pki/crl.pem /etc/openvpn/pki/**

```
server@server:/etc/openvpn$ ls pki/
ca.crt  crl.pem  dh.pem  servidor.crt  servidor.key
```

Con esto concluimos el apartado 2.

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

3) **Configurar el servidor VPN en modo TUN, configurar su arranque automático y su servicio. Habilitar la función de forwarding en el servidor.**

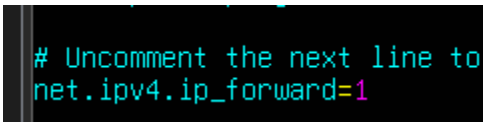
Antes de realizar la configuración del servidor en modo TUN, vamos a crear un directorio donde guardaremos todos los archivos esenciales para el openvpn-server. Por tanto, dentro del directorio (/etc/openvpn) creamos un directorio **server**:

Dentro del directorio server creamos un fichero llamado **server.conf** en el que configuraremos el lado del servidor para openvpn tal que así:

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

```
# Puerto donde se activa el servicio openvpn:
port 1194

# Protocolo que se usará para la comunicación
proto udp

# Modo de configuración de la VPN (TUN /TAP)
dev tun

# Rutas para encontrar los certificados del servidor:
ca /etc/openvpn/pki/ca.crt
cert /etc/openvpn/pki/servidor.crt
key /etc/openvpn/pki/servidor.key
dh /etc/openvpn/pki/dh.pem
crl-verify /etc/openvpn/pki/crl.pem

# Red que OpenVpn usará para asignar IPs a los clientes:
server 172.16.0.0 255.255.255.0

# Fichero donde se guardan las IPs asociadas a clientes que ya se conectaron

ifconfig-pool-persist ipp.txt

# Anunciar/mostrar a los clientes la red interna real detrás del servidor VPN (LAN 172.22.0.0/24)

push "route 172.22.0.0 255.255.255.0"

# Configuración de sesión:

# manda un ping cada 10 segundos, cierra la sesión si tras 120 seg no hay respuesta
keepalive 10 120

# Clave TLS para proteger contra escaneos y ataques DoS:

#tls-auth /etc/openvpn/server/ta.key 0

# Algoritmo de cifrado simétrico usado en el tunel:

data-ciphers AES-256-CBC


# Mantener claves y tunel si el servicio se reinicia:
persist-key
persist-tun


~
```

Activamos también el modo forward en el servidor, arrancar el servicio y dejarlo en enable para que se encienda solo cada vez que se encienda la máquina:



y guardar los cambios con sudo sysctl –p

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

```
server@server:~$ sudo cat /proc/sys/net/ipv4/ip_forward
1
server@server:~$
```

```
Home | Cliente1 | Cliente2 | Gateway | Server
erver@server:~$ sudo systemctl start openvpn-server@server
erver@server:~$ sudo systemctl enable openvpn-server@server
erver@server:~$ sudo systemctl status openvpn-server@server
 openvpn-server@server.service - OpenVPN service for server
    Loaded: loaded (/usr/lib/systemd/system/openvpn-server@.service; enabled; preset: enabled)
    Active: active (running) since Sat 2025-09-27 13:13:04 UTC; 7h ago
      Docs: man:openvpn(8)
            https://openvpn.net/community-resources/reference-manual-for-openvpn-2-6/
            https://community.openvpn.net/openvpn/wiki/HOWTO
  Main PID: 986 (openvpn)
    Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 4548)
    Memory: 3.0M (peak: 3.2M)
       CPU: 142ms
    CGroup: /system.slice/system-openvpn\x2dserver.slice/openvpn-server@server.service
            └─986 /usr/sbin/openvpn --status /run/openvpn-server/status-server.log --status-version 2 --suppress-timestamps --config server.conf

ep 27 13:13:04 server openvpn[986]: TUN/TAP device tun0 opened
ep 27 13:13:04 server openvpn[986]: net_iface_mtu_set: mtu 1500 for tun0
ep 27 13:13:04 server openvpn[986]: net_iface_up: set tun0 up
ep 27 13:13:04 server openvpn[986]: net_addr_ptp_v4_add: 172.16.0.1 peer 172.16.0.2 dev tun0
ep 27 13:13:04 server openvpn[986]: Could not determine IPv4/IPv6 protocol. Using AF_INET
ep 27 13:13:04 server openvpn[986]: UDPv4 link local (bound): [AF_INET][undef]:1194
ep 27 13:13:04 server openvpn[986]: UDPv4 link remote: [AF_UNSPEC]
ep 27 13:13:04 server openvpn[986]: ifconfig_pool_read(), in='cliente-01,172.16.0.4,'
ep 27 13:13:04 server openvpn[986]: succeeded -> ifconfig_pool_set(hand=0)
ep 27 13:13:04 server openvpn[986]: Initialization Sequence Completed
erver@server:~$
```

4. Configurar en el Gateway un port-forwarding para reenviar al servidor interno todo el tráfico recibido en el puerto udp/1194 de su interface externo (es necesaria una regla iptables nat en PREROUTING).

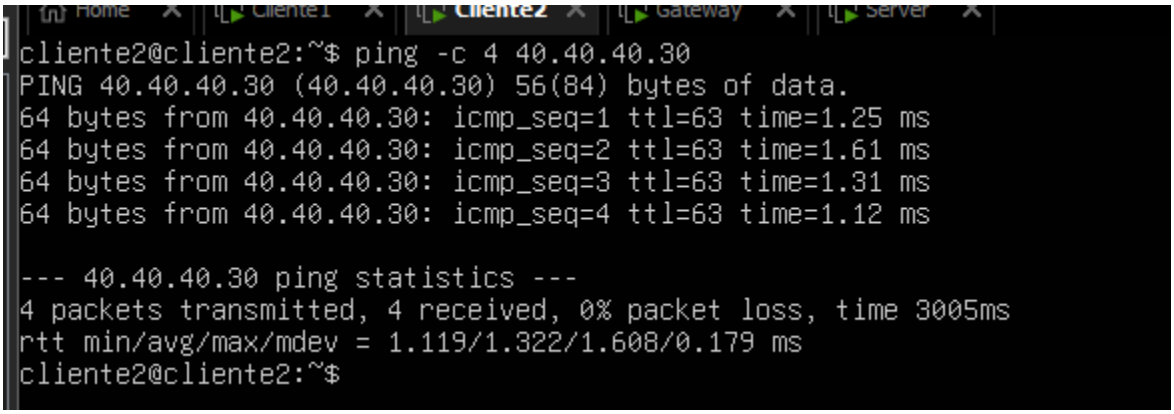
En el **gateway:**

```
gateway@gateway:~$ sudo iptables -t nat -A PREROUTING -p udp --dport 1194 -i ens33 -j DNAT --to-destination 172.22.0.80:1194
gateway@gateway:~$
```

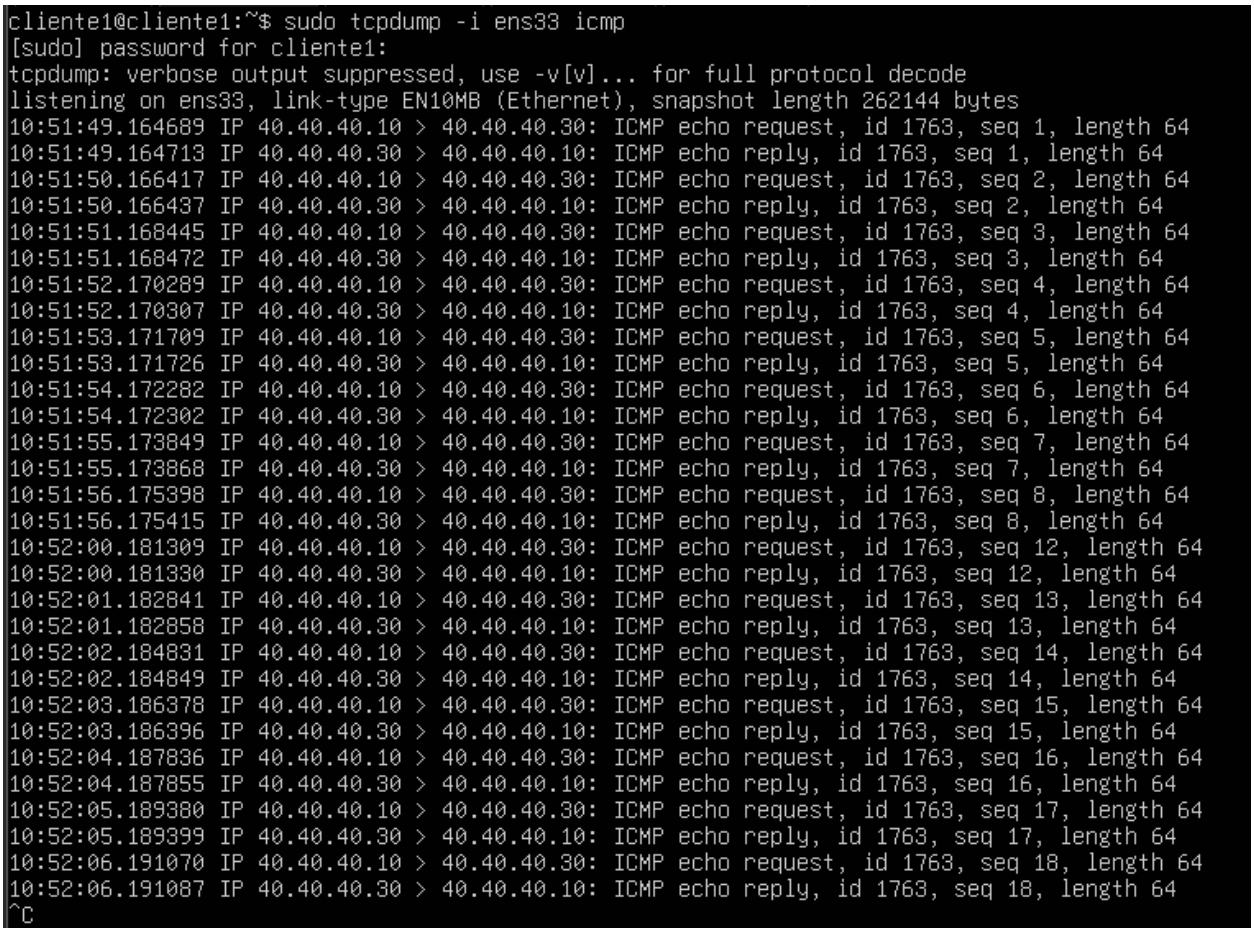
5. **Configurar en el Gateway el enmascaramiento para todo el tráfico saliente Comprobar que siguen pudiendo alcanzarse recursos externos desde la red interna (PC2 a PC1). Verificar que se realiza el enmascaramiento usando tcpdump.**

```
Home | Cliente1 | Cliente2 | Gateway | Server
gateway@gateway:~$ sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
```

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco



Enmascaramiento correcto en PC1:



Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

6 . Configurar el cliente OpenVPN en PC1 (client.conf) para conectar al servidor (a través de la IP del Gateway). Para completar esta tarea será necesario copiar los archivos necesarios obtenidos en el apartado 2.

```
cliente1@cliente1:~$ ls /etc/openvpn
client  server  update-resolv-conf
cliente1@cliente1:~$ ls /etc/openvpn/server/
cliente1@cliente1:~$ ls /etc/openvpn/client
ca.crt  client.conf  cliente-01.crt  cliente-01.key
cliente1@cliente1:~$ _
```

Copiar las claves y certificados al directorio de cliente1 como se muestra en la captura de arriba. Luego, crear un fichero de configuración en el directorio (**/etc/openvpn/client**)

Y añadir la siguiente configuración:

```
# Dirección y puerto del servidor VPN:
client
dev tun
proto udp

# IP externa del gateway

remote 40.40.40.10 1194

# Certificados y claves:

ca /etc/openvpn/client/ca.crt
cert /etc/openvpn/client/cliente-01.crt
key /etc/openvpn/client/cliente-01.key

# Si se usa un HMAC para protección TLS:

#remote-cert-tls server
# tls-auth ta.key 1

# Algoritmo de cifrado:

data-ciphers AES-256-CBC

# Mantener la conexión:
resolv-retry infinite
persist-key
persist-tun
```

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

7. Iniciar el cliente y comprobar que se conecta al servidor. Verificar la configuración que se establece en el interface virtual tun del cliente, su tabla de encaminamiento y que es posible alcanzar desde PC1 los recursos ofrecidos por el Servidor a través de la VPN (puede hacer una captura del tráfico intercambiado para verificar que los paquetes se transportan sobre tun0, que a su vez se envía encriptado sobre eth0). Desactivar el router por defecto en PC1 y comprobar que sigue siendo posible acceder al Servidor.

Iniciamos el cliente:

```
cliente1@cliente1:~$ sudo systemctl start openvpn-client@client
cliente1@cliente1:~$ sudo systemctl status openvpn-client@client
• openvpn-client@client.service - OpenVPN tunnel for client
     Loaded: loaded (/usr/lib/systemd/system/openvpn-client@.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-09-28 10:59:59 UTC; 36s ago
       Docs: man:openvpn(8)
             https://openvpn.net/community-resources/reference-manual-for-openvpn-2-6/
             https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 1802 (openvpn)
     Status: "Initialization Sequence Completed"
      Tasks: 1 (limit: 7564)
     Memory: 2.8M (peak: 3.1M)
        CPU: 31ms
     CGroup: /system.slice/system-openvpn\x2dclient.slice/openvpn-client@client.service
             └─1802 /usr/sbin/openvpn --suppress-timestamps --nobind --config client.conf

sep 28 10:59:59 cliente1 systemd[1]: Started openvpn-client@client.service - OpenVPN tunnel for
sep 28 10:59:59 cliente1 openvpn[1802]: TCP/UDP: Preserving recently used remote address: [AF_I
sep 28 10:59:59 cliente1 openvpn[1802]: UDPv4 link local: (not bound)
sep 28 10:59:59 cliente1 openvpn[1802]: UDPv4 link remote: [AF_INET]40.40.40.10:1194
sep 28 10:59:59 cliente1 openvpn[1802]: [servidor] Peer Connection Initiated with [AF_INET]40.4
sep 28 10:59:59 cliente1 openvpn[1802]: TUN/TAP device tun0 opened
sep 28 10:59:59 cliente1 openvpn[1802]: net_iface_mtu_set: mtu 1500 for tun0
sep 28 10:59:59 cliente1 openvpn[1802]: net_iface_up: set tun0 up
sep 28 10:59:59 cliente1 openvpn[1802]: net_addr_ptp_v4_add: 172.16.0.6 peer 172.16.0.5 dev tun
sep 28 10:59:59 cliente1 openvpn[1802]: Initialization Sequence Completed
cliente1@cliente1:~$ s
```

Verificar configuración interfaz virtual:

```
cliente1@cliente1:~$ ip addr show tun0
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 172.16.0.6 peer 172.16.0.5/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::8f52:df0f:2bff:6760/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
cliente1@cliente1:~$
```

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

Verificar tabla de encaminamiento:

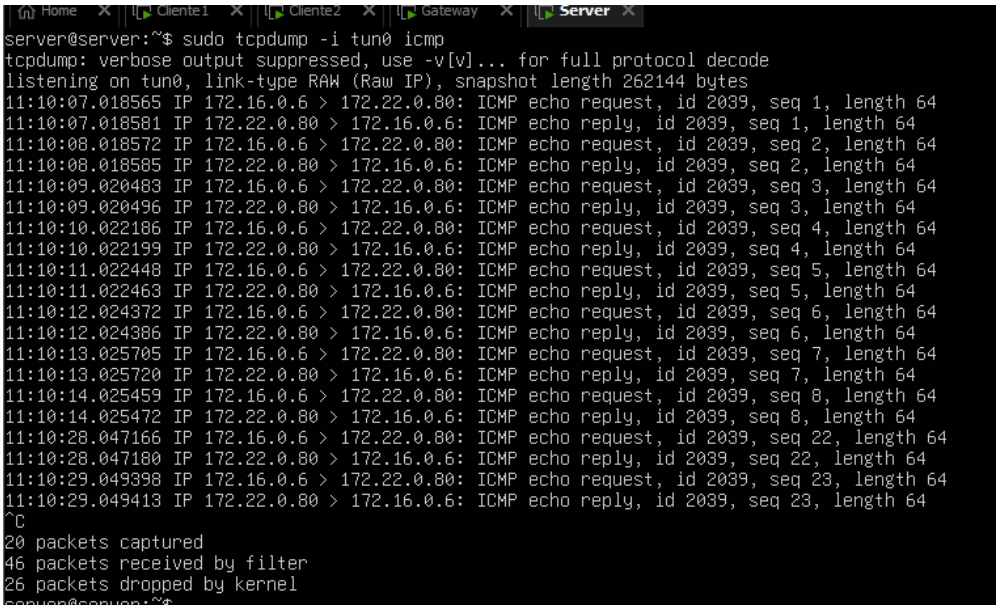
```
cliente1@cliente1:~$ ip route
40.40.40.0/24 dev ens33 proto kernel scope link src 40.40.40.30
172.16.0.1 via 172.16.0.5 dev tun0
172.16.0.5 dev tun0 proto kernel scope link src 172.16.0.6
172.22.0.0/24 via 172.16.0.5 dev tun0
cliente1@cliente1:~$
```

Alcanzando recursos red privada desde PC1 (PC1 – Server):

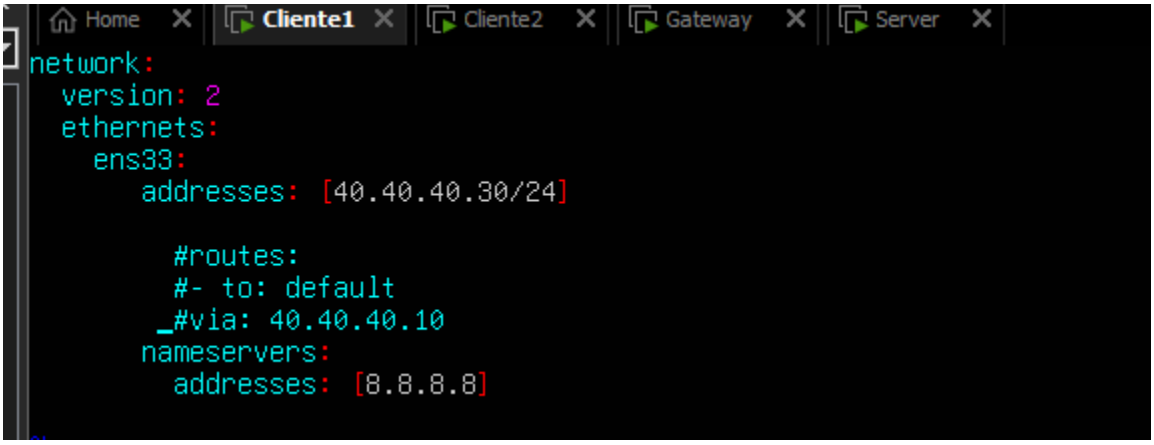
Home | Cliente1 | Cliente2 | Gateway | Server

```
cliente1@cliente1:~$ ping 172.22.0.80
PING 172.22.0.80 (172.22.0.80) 56(84) bytes of data.
64 bytes from 172.22.0.80: icmp_seq=1 ttl=64 time=1.79 ms
64 bytes from 172.22.0.80: icmp_seq=2 ttl=64 time=2.07 ms
64 bytes from 172.22.0.80: icmp_seq=3 ttl=64 time=2.12 ms
64 bytes from 172.22.0.80: icmp_seq=4 ttl=64 time=2.02 ms
64 bytes from 172.22.0.80: icmp_seq=5 ttl=64 time=1.89 ms
64 bytes from 172.22.0.80: icmp_seq=6 ttl=64 time=1.59 ms
64 bytes from 172.22.0.80: icmp_seq=7 ttl=64 time=1.68 ms
64 bytes from 172.22.0.80: icmp_seq=8 ttl=64 time=1.99 ms
64 bytes from 172.22.0.80: icmp_seq=9 ttl=64 time=1.58 ms
64 bytes from 172.22.0.80: icmp_seq=10 ttl=64 time=1.50 ms
64 bytes from 172.22.0.80: icmp_seq=11 ttl=64 time=1.53 ms
64 bytes from 172.22.0.80: icmp_seq=12 ttl=64 time=1.74 ms
64 bytes from 172.22.0.80: icmp_seq=13 ttl=64 time=1.63 ms
64 bytes from 172.22.0.80: icmp_seq=14 ttl=64 time=1.60 ms
64 bytes from 172.22.0.80: icmp_seq=15 ttl=64 time=1.78 ms
64 bytes from 172.22.0.80: icmp_seq=16 ttl=64 time=1.68 ms
64 bytes from 172.22.0.80: icmp_seq=17 ttl=64 time=1.60 ms
64 bytes from 172.22.0.80: icmp_seq=18 ttl=64 time=1.72 ms
64 bytes from 172.22.0.80: icmp_seq=19 ttl=64 time=1.82 ms
64 bytes from 172.22.0.80: icmp_seq=20 ttl=64 time=1.68 ms
64 bytes from 172.22.0.80: icmp_seq=21 ttl=64 time=1.62 ms
64 bytes from 172.22.0.80: icmp_seq=22 ttl=64 time=1.69 ms
64 bytes from 172.22.0.80: icmp_seq=23 ttl=64 time=1.79 ms
64 bytes from 172.22.0.80: icmp_seq=24 ttl=64 time=1.76 ms
^C
--- 172.22.0.80 ping statistics ---
24 packets transmitted, 24 received, 0% packet loss, time 23031ms
rtt min/avg/max/mdev = 1.495/1.744/2.124/0.165 ms
cliente1@cliente1:~$ _
```

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco



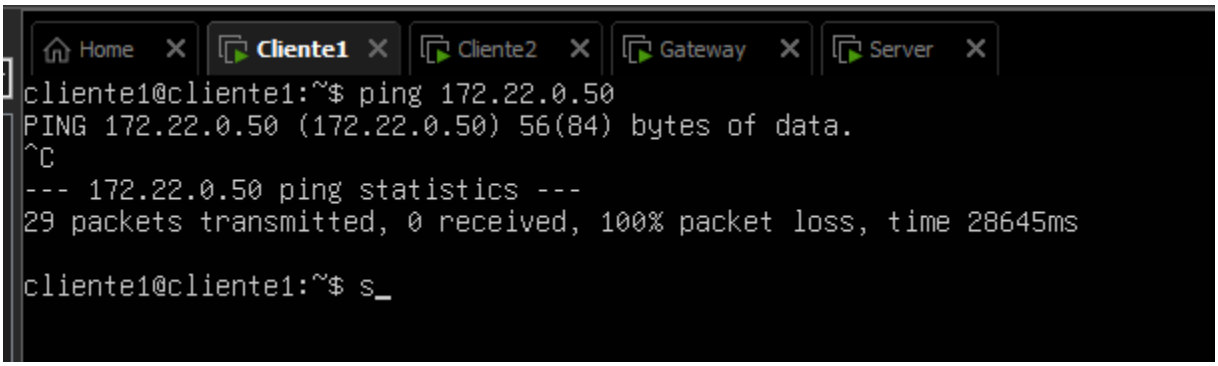
Desde el comienzo de la práctica tenemos el router por defecto desactivado en PC1:



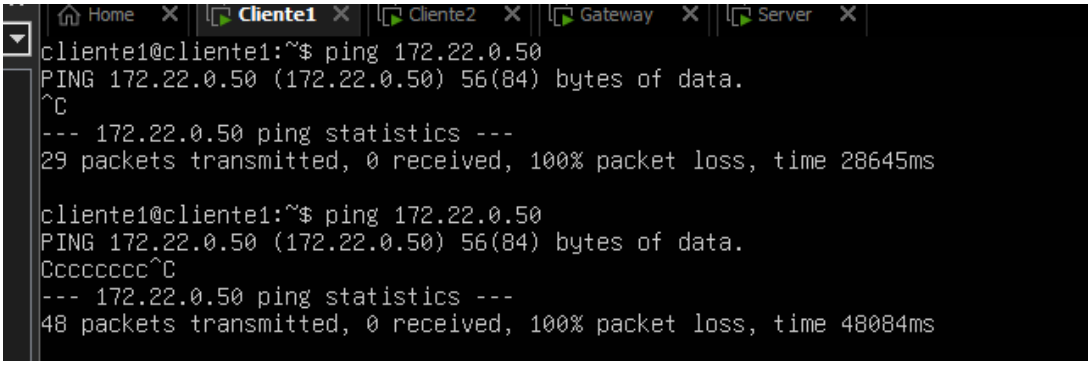
8. Compruebe si es posible alcanzar el equipo PC2 desde PC1 a través de la conexión VPN. ¿Qué está sucediendo?

Cuando se hace un ping desde PC1 – PC2 se puede ver claramente que no se obtiene una respuesta:

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco



Esto NO es porque PC1 no llegue al PC2 a través de la conexión VPN, sino más bien que PC2 no sabe como llegar a PC1 y por tanto no hay respuesta. Eso se puede contrastar con lo siguiente:



Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

```
cliente2@cliente2:~$ sudo tcpdump -i ens33 icmp
[sudo] password for cliente2:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
11:24:40.515628 IP 172.22.0.10 > 172.22.0.80: ICMP net 8.8.8.8 unreachable, length 68
11:24:40.515628 IP 172.22.0.10 > 172.22.0.80: ICMP net 8.8.8.8 unreachable, length 68
11:24:44.540377 IP 172.22.0.10 > 172.22.0.50: ICMP net 8.8.8.8 unreachable, length 68
11:24:47.225257 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 1, length 64
11:24:47.225273 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 1, length 64
11:24:47.225778 IP 172.22.0.10 > 172.22.0.50: ICMP net 172.16.0.6 unreachable, length 92
11:24:47.662247 IP 172.22.0.10 > 172.22.0.80: ICMP net 8.8.8.8 unreachable, length 68
11:24:48.228512 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 2, length 64
11:24:48.228528 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 2, length 64
11:24:48.229028 IP 172.22.0.10 > 172.22.0.50: ICMP net 172.16.0.6 unreachable, length 92
11:24:49.229419 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 3, length 64
11:24:49.229435 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 3, length 64
11:24:50.276153 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 4, length 64
11:24:50.276171 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 4, length 64
11:24:51.300238 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 5, length 64
11:24:51.300293 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 5, length 64
11:25:10.732191 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 24, length 64
11:25:10.732208 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 24, length 64
11:25:20.996227 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 34, length 64
11:25:20.996242 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 34, length 64
11:25:31.236471 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 44, length 64
11:25:31.236487 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 44, length 64
11:25:32.238151 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 45, length 64
11:25:32.238168 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 45, length 64
11:25:33.284140 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 46, length 64
11:25:33.284157 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 46, length 64
11:25:33.765892 IP 172.22.0.10 > 172.22.0.80: ICMP net 8.8.8.8 unreachable, length 68
11:25:33.765892 IP 172.22.0.10 > 172.22.0.80: ICMP net 8.8.8.8 unreachable, length 68
11:25:34.308702 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 47, length 64
11:25:34.308722 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 47, length 64
11:25:35.309397 IP 172.16.0.6 > 172.22.0.50: ICMP echo request, id 2089, seq 48, length 64
11:25:35.309412 IP 172.22.0.50 > 172.16.0.6: ICMP echo reply, id 2089, seq 48, length 64
11:25:40.974374 IP 172.22.0.10 > 172.22.0.80: ICMP net 8.8.8.8 unreachable, length 68
11:25:42.908724 IP 172.22.0.10 > 172.22.0.50: ICMP net 8.8.8.8 unreachable, length 68
^C
34 packets captured
```

9. Modificar la configuración de PC2 para que pueda comunicar con PC1 a través de la VPN. Nota: también es posible configurar el Gateway para el retorno de los paquetes a través de la VPN, sin modificar el encaminamiento en PC2. Compruebe ambas opciones.

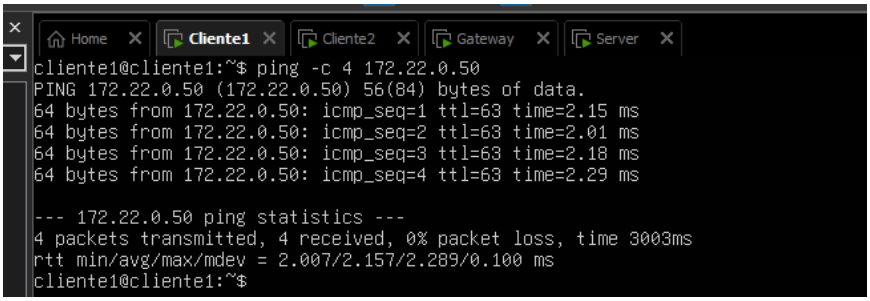
Lo que podemos hacer para PC2 es ponerle una ruta estática para que reenvíe el tráfico hacia la conexión VPN redirigiéndolo hacia el servidor:

```
Home | Cliente1 | Cliente2 | Gateway | Server
cliente2@cliente2:~$ sudo ip route add 172.16.0.0/24 via 172.22.0.80
cli Home cliente2:~$ _
```

O también se puede hacer poniendo una ruta en el Gateway (más cómodo si hay mas dispositivos en la red y no se quiere tener que configurar todos, o no los quieres tocar)

```
Last login: Sun Sep 28 10:50:46 2025 from 172.22.0.1
root@GW:~# sudo ip route add 172.16.0.0/24 via 172.22.0.80 dev eth0_
```

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco



Ahora sí que funciona, pero está solución no es persistente, por tanto:

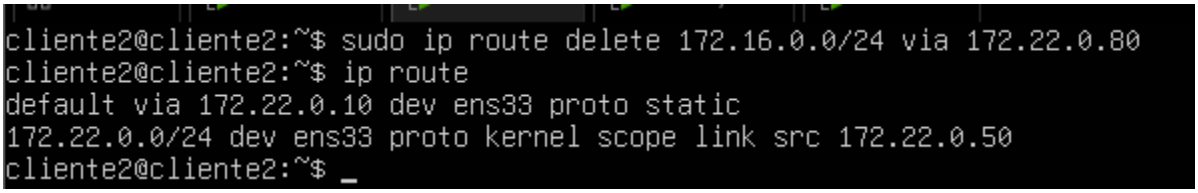
Añadimos la ruta estática a (**etc/netplan/50-cloud-init.yml**) y aplicamos los cambios netplan apply para que sea persistente.

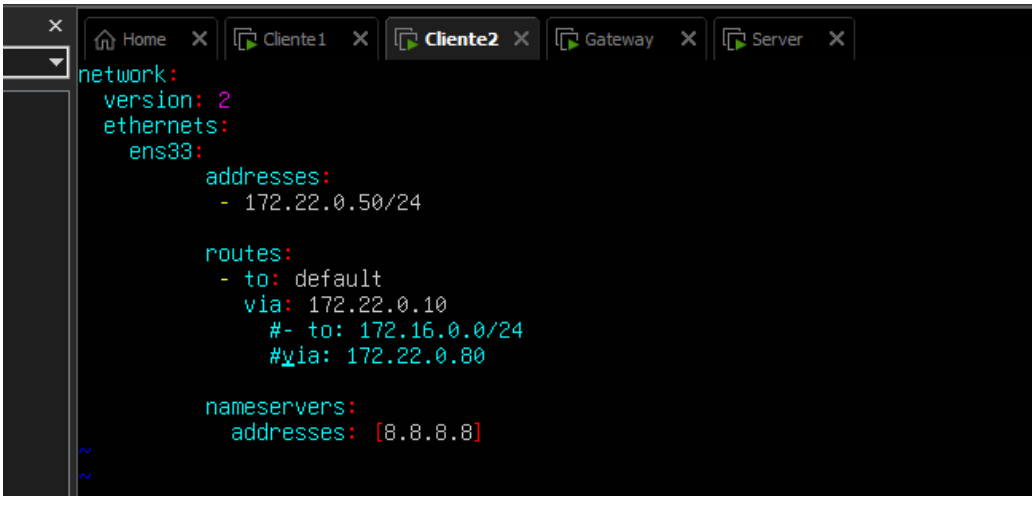
También podríamos añadir la ruta estática en el gateway (temporal) o configurar la misma ruta estática en el propio netplan del gateway que es lo que vamos a realizar:

Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

(Utilizar netplan apply para guardar la configuración)

Ahora comprobamos que funcione, para ello eliminamos la ruta estática temporal y comentamos aquella configurada de forma persistente en el PC2:





Daniel Gómez Obraztsov
Giulio Francesco Tizzano
Adolfo Blanco

Comprobamos que funcione:

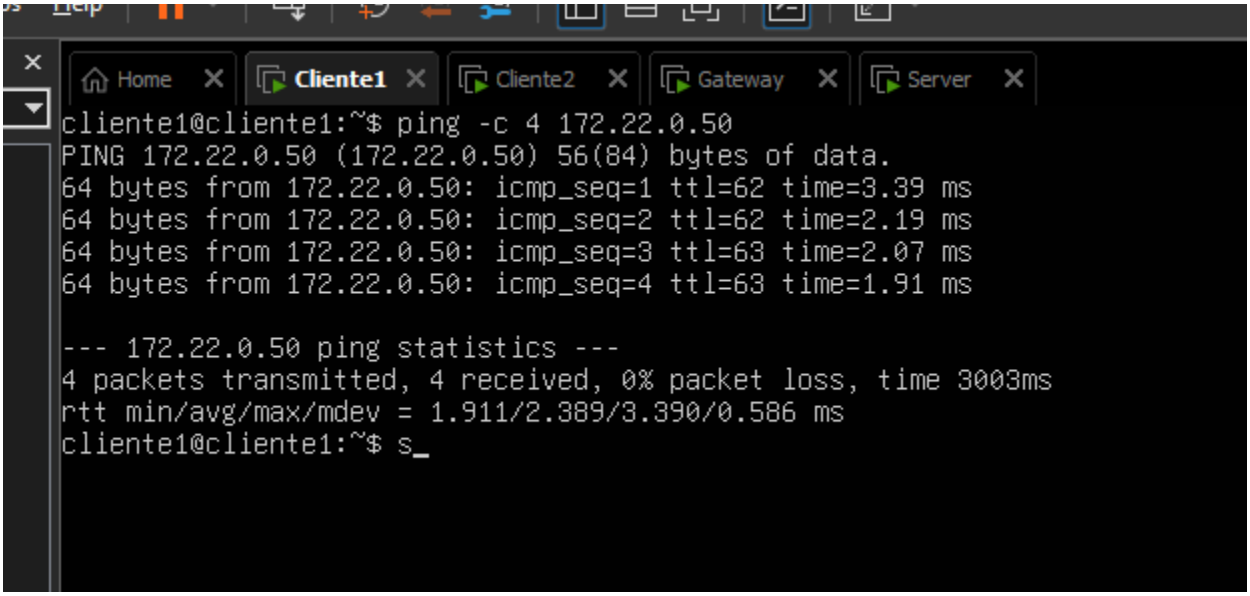


Tabla de encaminamiento para confirmarlo

Para hacer persistentes las relgas NAT (gateway):

- Persistencia de las reglas (iptables-save e iptables-restore)
  Salvamos las reglas en un archivo de configuración

```
iptables-save > /etc/iptables-rules
iptables-restores < /etc/iptables-rules
```