



UNIVERSIDAD DE COLIMA

Facultad de Ingeniería Mecánica y Eléctrica

Seguridad en la Red

Práctica #4 Servidor VPN

Elaborado por:

Puga Hernández Emilio Israel

Profesor: Dr. Leonel Soriano Equigua

Coquimatlán, Colima, 29 de Noviembre del 2023

Índice

| | |
|---|-----------|
| 1. Objetivo(s) | 5 |
| 2. Marco teórico | 6 |
| 3. Desarrollo | 7 |
| 3.1. Configuración de la Máquina Virtual del Servidor. | 7 |
| 3.2. Configuración del Adaptador de Red, en maquina virtual | 9 |
| 3.3. Instalación del Sistema Operativo | 11 |
| 3.4. SSH (Servidor) | 12 |
| 3.5. Instalación de OpenVPN | 14 |
| 3.6. Instalación de Easy-RSA | 14 |
| 3.7. Crear la PKI para OpenVPN | 15 |
| 3.8. Crear una entidad de certificación (CA) | 16 |
| 3.9. Crear una solicitud de certificado de servidor de OpenVPN y una clave privada | 17 |
| 3.10. Generar el parámetro Diffie-Hellman para el servidor | 18 |
| 3.11. Firma del Certificado del Servidor OpenVPN por la Autoridad de Certificación (CA) | 18 |
| 3.12. Generación de Clave de Autenticación TLS (tls-auth) | 19 |
| 3.13. Configuración del entorno de ejecución del servidor | 20 |
| 3.13.1. Direcciones del Servidor | 20 |
| 3.14. Tránsito de claves al servidor | 23 |
| 3.15. Configuración de la Máquina Virtual del cliente | 25 |
| 3.16. Instalación del Sistema Operativo del Cliente | 27 |
| 3.17. SSH (Cliente) | 28 |
| 3.18. Instalación de OpenVPN en el Cliente | 30 |
| 3.19. Configuración del entorno de ejecución del cliente | 30 |
| 3.19.1. Direcciones del Cliente | 31 |
| 3.20. Creación de los requerimientos necesarios para el cliente | 33 |
| 3.21. Tránsito de claves al cliente | 34 |
| 3.21.1. Organización de claves para el cliente en el servidor | 34 |
| 3.21.2. Tránsito de claves mediante scp a la maquina cliente | 35 |
| 3.22. Pruebas de funcionamiento | 37 |
| 4. Conclusiones | 40 |
| 5. Referencias | 41 |

Índice de figuras

| | | |
|-----|--|----|
| 1. | Configuración básica de la máquina del servidor | 7 |
| 2. | Características del Hardware del servidor | 8 |
| 3. | Pestaña de asignación de memoria del servidor | 8 |
| 4. | Pestaña de confirmación del servidor | 9 |
| 5. | Selección de la pestaña de configuración | 9 |
| 6. | Pestaña de la configuración de Red | 10 |
| 7. | Elección del adaptador de red correcto | 10 |
| 8. | Creación de usuario y contraseña del servidor | 11 |
| 9. | Paquetes requeridos para el sistema del servidor | 11 |
| 10. | Error en la autenticación del servidor | 12 |
| 11. | Asignación de contraseña al usuario root del servidor | 12 |
| 12. | IP del sistema del servidor | 13 |
| 13. | Conexión vía SSH al servidor | 13 |
| 14. | Instalación de OpenVPN en el servidor | 14 |
| 15. | Instalación de Easy-RSA en el servidor | 15 |
| 16. | Carpeta para la generación de llaves privadas del servidor | 15 |
| 17. | Enlace simbólico para la estructura PKI del servidor | 15 |
| 18. | Carpeta para la generación de llaves privadas del servidor | 16 |
| 19. | Contenido de la carpeta pk i del servidor | 16 |
| 20. | Creación de la Autoridad de Certificación (CA) en el servidor | 17 |
| 21. | Solicitud de certificado del servidor | 17 |
| 22. | Generación del parámetro Diffie-Hellman para el servidor | 18 |
| 23. | Solicitud de firma del Certificado para el Servidor OpenVPN | 19 |
| 24. | Generación de la Clave de Autenticación TLS para el servidor y los clientes | 19 |
| 25. | Copiado del archivo de configuración del servidor | 20 |
| 26. | Crear carpeta para organizar los archivos del servidor | 20 |
| 27. | Retroceder al directorio /etc/openvpn en el servidor | 20 |
| 28. | Configuración de la dirección IP del servidor por la que escuchara OpenVPN | 21 |
| 29. | Configuración de las dirección de certificados y llave privada del servidor | 22 |
| 30. | Configuración de la dirección para el parámetro Diffie-Hellman del servidor | 22 |
| 31. | Configuración de la dirección para la llave privada ta.key en el servidor | 23 |
| 32. | Trasferencia del certificado ca.crt en el servidor | 23 |
| 33. | Trasferencia del certificado para el servidor | 24 |
| 34. | Trasferencia de la llave del servidor a la carpeta /etc/openvpn/server/keys/ en el servidor | 24 |
| 35. | Ubicación de la clave de autenticación TLS | 24 |
| 36. | Nombre e ISO de la maquina cliente | 25 |
| 37. | Asignación de hardware de la maquina cliente | 25 |
| 38. | Tamaño del disco virtual de la maquina cliente | 26 |
| 39. | Resumen de configuraciones de la maquina cliente | 26 |
| 40. | Creación de usuario y contraseña para el cliente | 27 |
| 41. | Paquetes requeridos para el sistema del cliente | 27 |
| 42. | Error de autenticación del cliente | 28 |
| 43. | Asignación de contraseña para el usuario root del cliente | 28 |
| 44. | IP del sistema del cliente | 29 |
| 45. | Conexión vía SSH al cliente | 29 |
| 46. | Instalación de OpenVPN en el cliente | 30 |
| 47. | Copiado del archivo de configuración del cliente | 30 |
| 48. | Crear carpeta para organizar los archivos del cliente | 30 |
| 49. | Configuración de la dirección IP del servidor en el archivo de configuración del cliente | 31 |
| 50. | Configuración de las dirección de certificados y llave privada del cliente | 32 |
| 51. | Configuración de la dirección para la llave privada ta.key en el cliente | 32 |
| 52. | Solicitud de certificado del cliente | 33 |
| 53. | Solicitud de firma del Certificado para el Cliente OpenVPN | 34 |
| 54. | Trasferencia del certificado ca.crt en el servidor a la carpeta cliente | 34 |
| 55. | Trasferencia del certificado para el cliente a la carpeta cliente | 35 |

| | | |
|-----|---|----|
| 56. | Trasferencia de la llave del cliente a la carpeta <code>/etc/openvpn/client/</code> en el servidor | 35 |
| 57. | Trasferencia de la clave de autenticación TLS a la carpeta <code>/etc/openvpn/client/</code> en el servidor | 35 |
| 58. | Trasferencia de las clave a la maquina cliente remotamente con el comando “ <code>scp</code> ” | 36 |
| 59. | Movimiento las claves del cliente a su carpeta en la maquina cliente | 36 |
| 60. | Asignación de ruta especifica en la tabla de enrutamiento del servidor | 37 |
| 61. | Asignación de una dirección IP al servidor dentro del rango de red que se utilizará para las conexiones VPN | 38 |
| 62. | Ejercicio del servicio OpenVPN en el servidor | 38 |
| 63. | Ejecución del servicio OpenVPN en el cliente | 39 |
| 64. | Ping del cliente al servidor | 39 |

1. Objetivo(s)

1. Interconectar redes privadas utilizando OpenVPN en Ubuntu, permitiendo la comunicación segura y protegida de datos entre ambas redes.
2. Configurar el servidor VPN para permitir a usuarios remotos conectarse de manera segura a la red local, proporcionando un acceso seguro a recursos internos sin comprometer la integridad de los datos.
3. Optimizar la configuración del servidor OpenVPN en Ubuntu 22.04.3 para garantizar una alta disponibilidad y estabilidad, minimizando posibles interrupciones en la conexión y asegurando un servicio confiable.

2. Marco teórico

Servidor VPN

Un servidor VPN es un servidor físico o virtual que está configurado para alojar y entregar servicios VPN a usuarios de todo el mundo. El servidor es una combinación de hardware VPN y software VPN que permite a los clientes VPN conectarse a una red privada segura. A diferencia de la mayoría de los servidores, un servidor VPN generalmente tiene más puertos de comunicaciones lógicos y físicos [1].

Las tecnologías utilizadas para la elaboración de este proyecto son las siguientes

VirtualBox

VirtualBox es una aplicación que sirve para hacer máquinas virtuales con instalaciones de sistemas operativos. Esto quiere decir que si tienes un ordenador con Windows, GNU/Linux o incluso macOS, puedes crear una máquina virtual con cualquier otro sistema operativo para utilizarlo dentro del que estés usando [2].

Ubuntu

Ubuntu es un sistema operativo basado en Linux, lo que llamamos comúnmente una distribución. Como toda distro (como se conocen comúnmente las distribuciones de Linux), es de código abierto. Además el sistema operativo es totalmente gratuito para cualquier uso, tanto en sus versiones para ordenadores personales como para servidores [3].

OpenVPN

OpenVPN es tanto un protocolo VPN como un software que utiliza técnicas VPN para asegurar conexiones punto a punto y de sitio a sitio. Actualmente, es uno de los protocolos VPN más populares entre los usuarios de VPN. El protocolo OpenVPN es responsable de manejar las comunicaciones cliente-servidor [4].

3. Desarrollo

3.1. Configuración de la Máquina Virtual del Servidor.

1. Nombre y sistema de la máquina virtual

Para el desarrollo de este proyecto utilizaremos el gestor de máquinas virtuales para Windows, Virtual Box, se creará una máquina con un sistema operativo Linux para este caso Ubuntu Server en su versión 22.04.3. Como primeros pasos para configurar la máquina son (véase en la Figura1):

- Asignar un nombre (este solo sera como se llamara la máquina para Virtual Box, no tiene nada que ver de como se llamara en el sistema a instalar).
- Asignar si así se desea una ubicación específica donde se guardara la máquina virtual en el sistema base.
- Indicar cual es la Imagen ISO del sistema operativo que deseamos virtualizar.

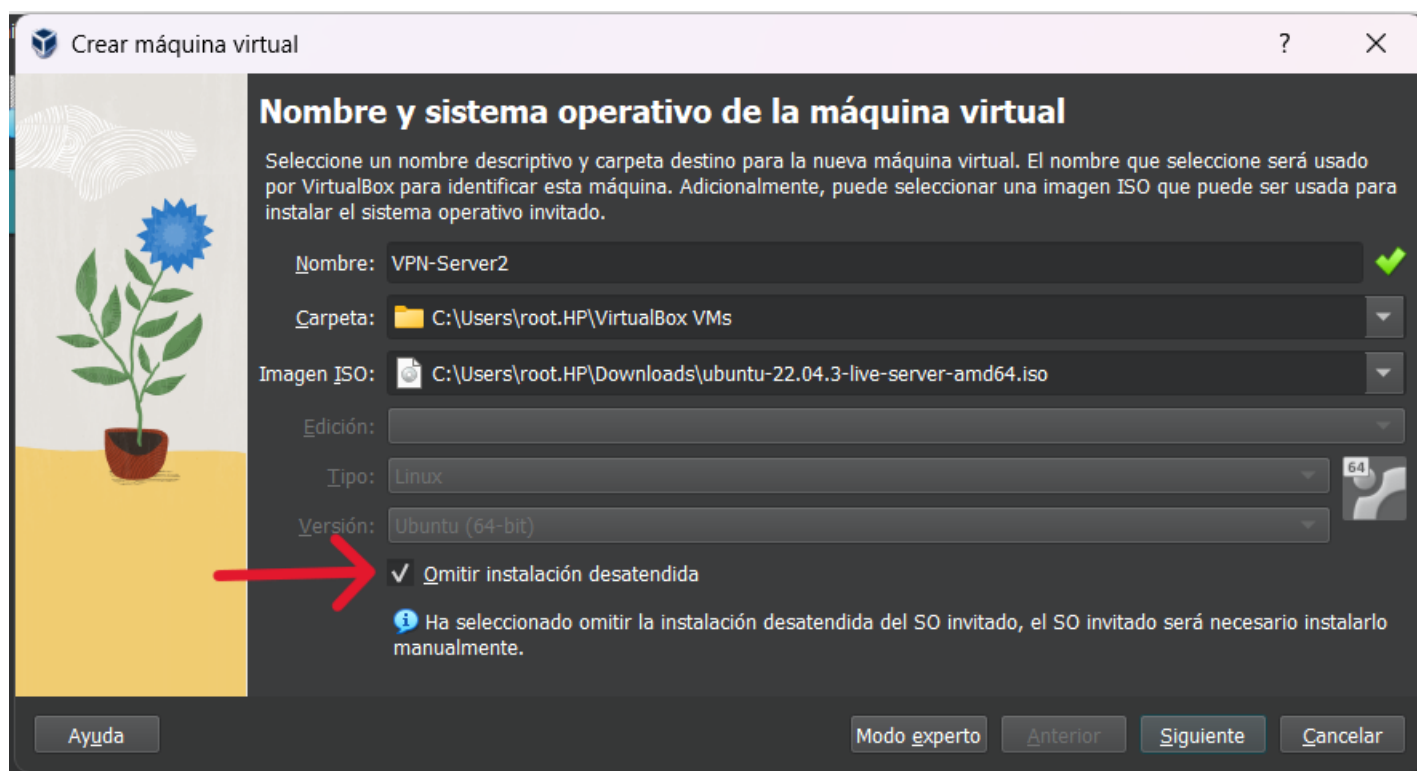


Figura 1: Configuración básica de la máquina del servidor

Es muy importante seleccionar la casilla “Omitir instalación desatendida”

2. Hardware

Se recomienda utilizar 4GB (4096 MB) de Ram y 4 núcleos del procesador (Figura 2).

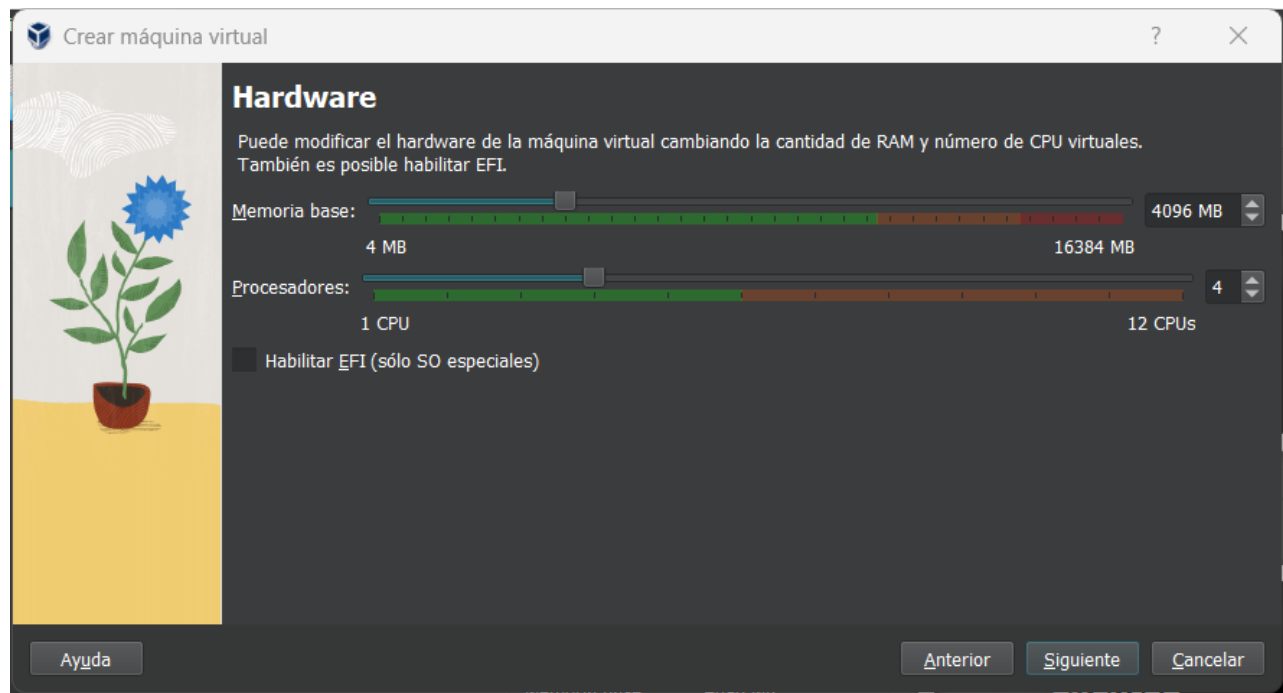


Figura 2: Características del Hardware del servidor

3. Disco duro virtual

Se recomienda asignar 10 GB como mínimo de memoria virtual base para el sistema operativo (Figura 3).

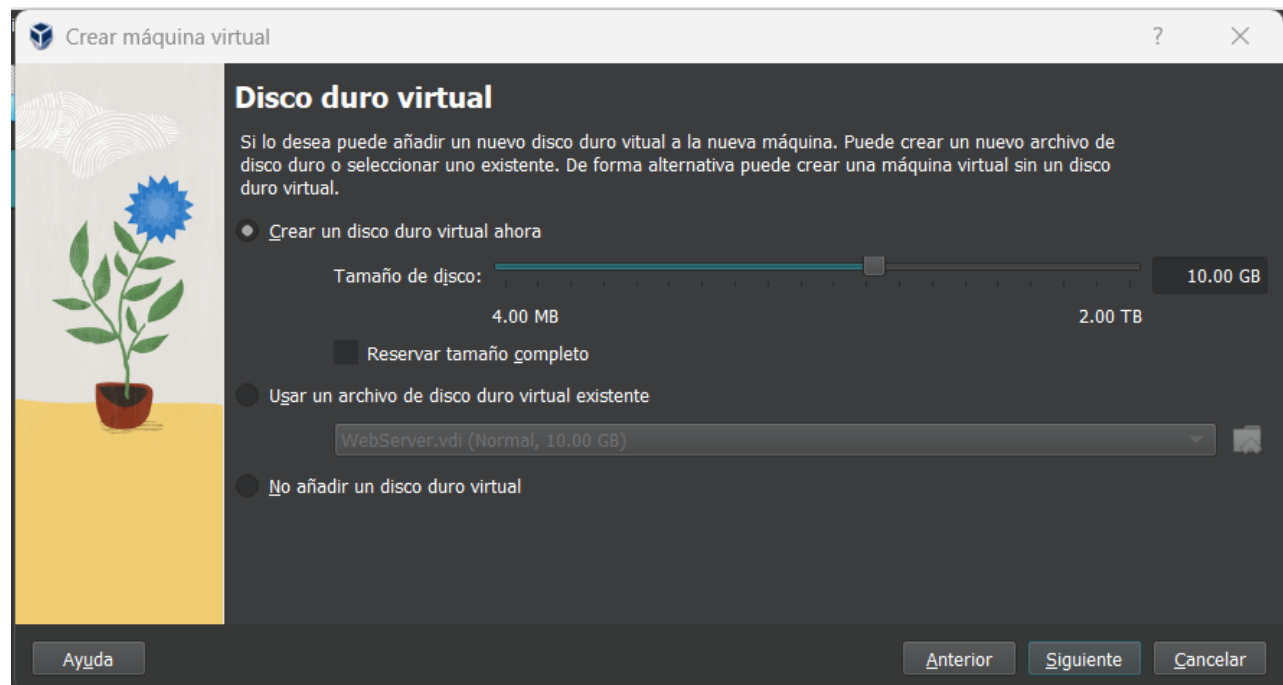


Figura 3: Pestaña de asignación de memoria del servidor

4. Resumen de configuración

Una vez creada la máquina virtual, y avernos corroborado los ajustes que se piden en este manual son los correctos terminaremos con la configuración (Figura4).



Figura 4: Pestaña de confirmación del servidor

3.2. Configuración del Adaptador de Red, en maquina virtual

1. En la pantalla de inicio seleccionamos nuestra maquina virtual y entramos en “Configuración”, véase en la Figura 5:

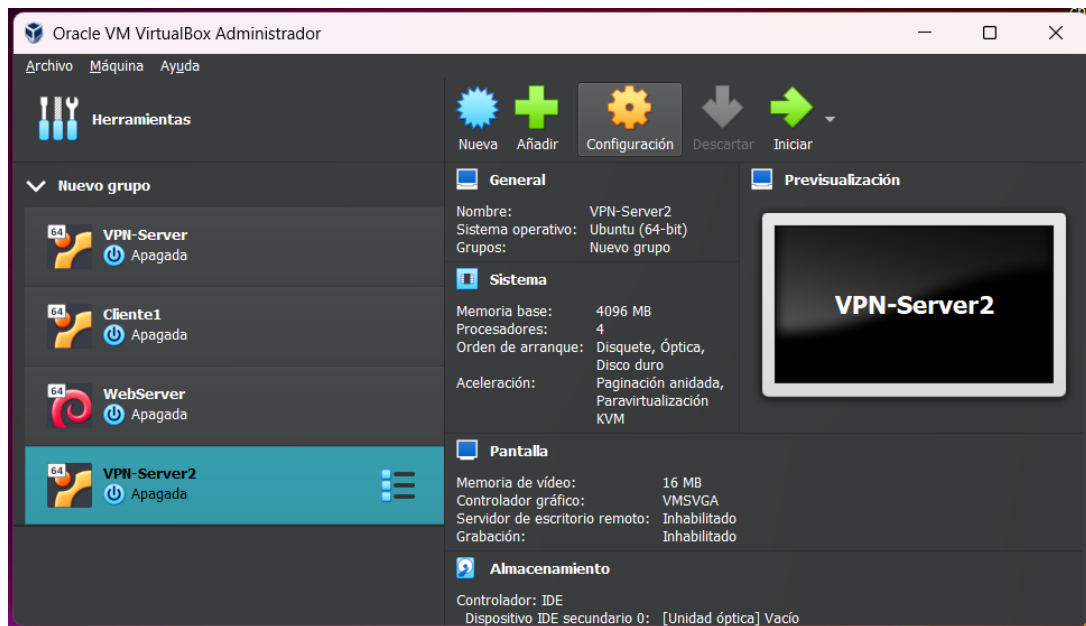


Figura 5: Selección de la pestaña de configuración

2. Entramos en el apartado de red, vease en la Figura 6:

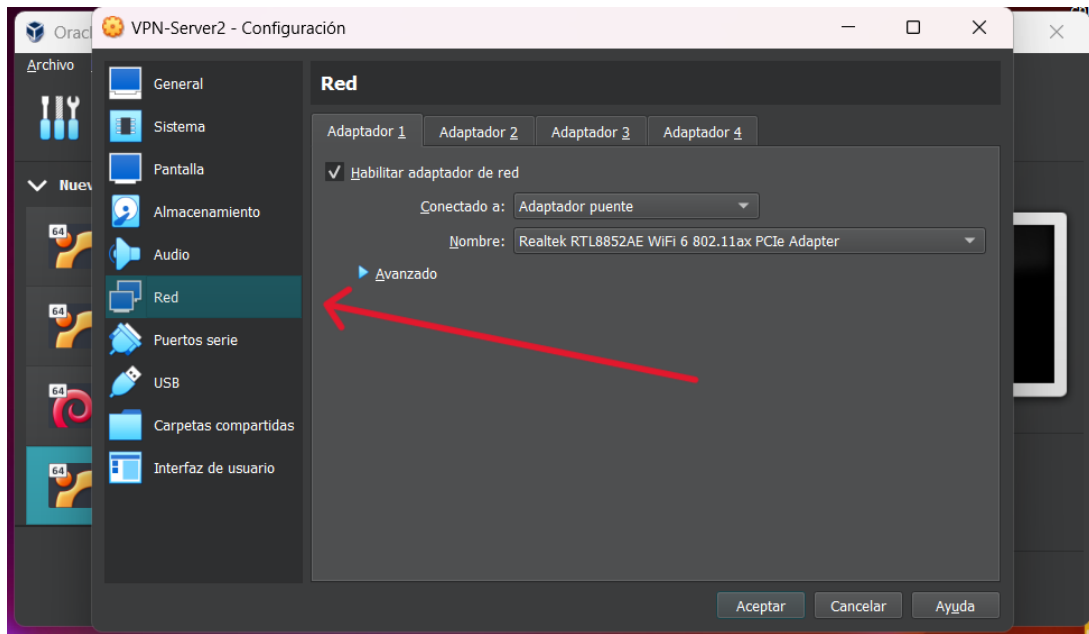


Figura 6: Pestaña de la configuración de Red

3. Y nos aseguramos de que estemos utilizando el adaptador de red correcto (Tarjeta de Red), en mi caso es un adaptador WiFi, pero para la demostración presencial sera el Ethernet y esto va variar según cuales sean las circunstancias (Figura 7).

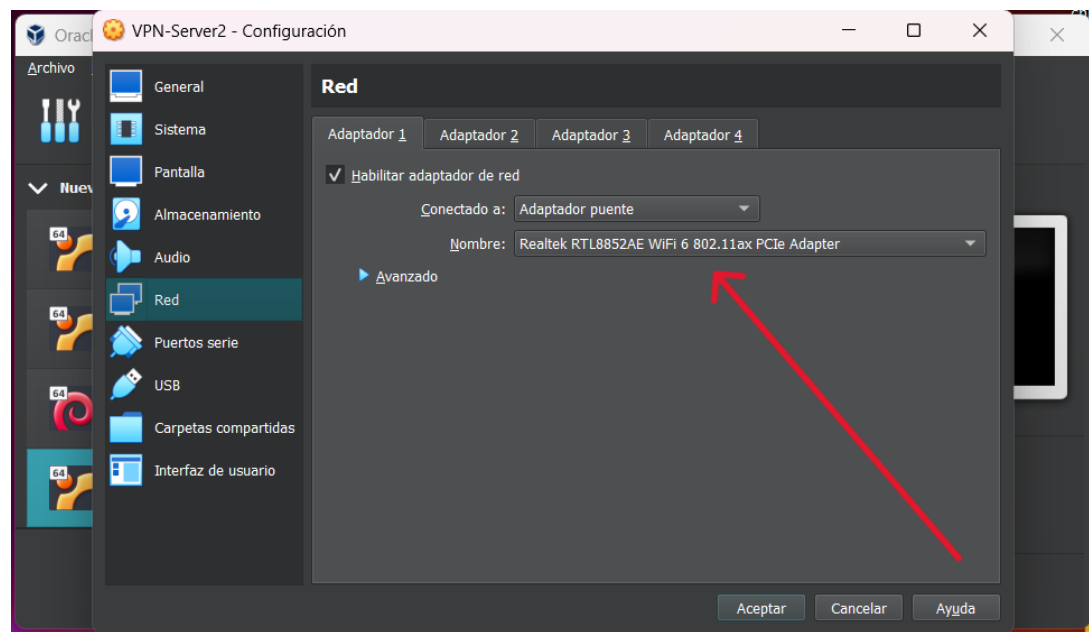


Figura 7: Elección del adaptador de red correcto

3.3. Instalación del Sistema Operativo

Para esta sección no considero necesario hacer un proceso de instalación del sistema operativo debido a que no incluyo ningún ajuste adicional o fuera de lo normal, como en la sección anterior 3.1, sin embargo este fue mi elecciones del nombre de sistema y usuario (véase en la Figura 8):

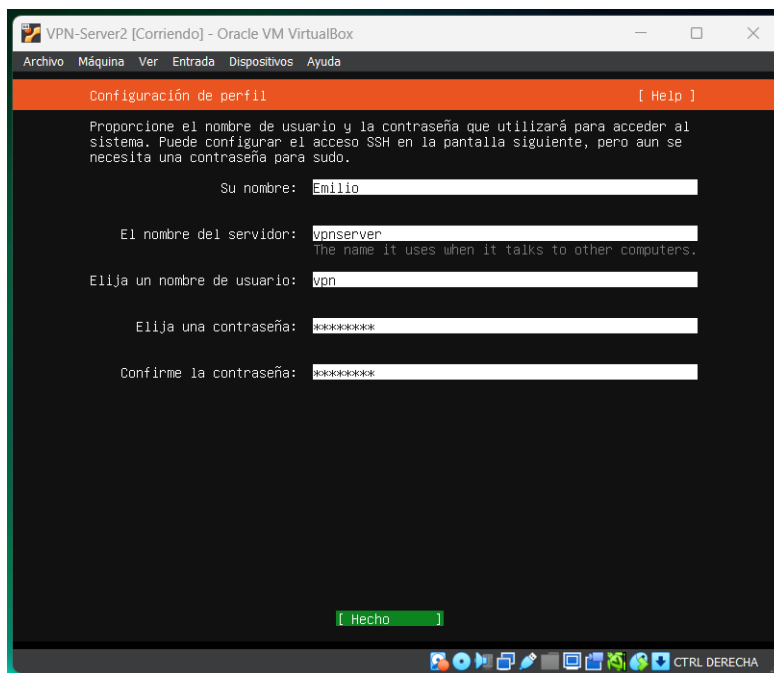


Figura 8: Creación de usuario y contraseña del servidor

No es necesario instalar ningún paquete ni dependencia extra, esto a pesar de que trabajare mediante SSH por comodidad (Figura 9).

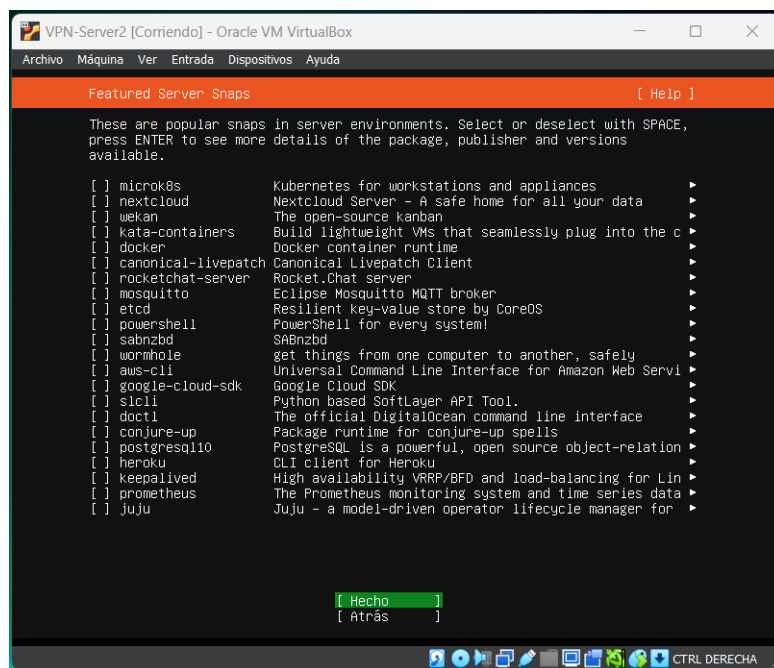


Figura 9: Paquetes requeridos para el sistema del servidor

3.4. SSH (Servidor)

Trabaje con mediante SSH por comodidad pero no es ningún requisito para el desarrollo de esta practica, para este caso donde el servidor esta siendo virtualizado en una red domestica es un entorno controlado, no es necesario instalar alguna dependencia como OpenSSL para añadir una capa extra de seguridad a la comunicación entre el administrador y del servidor mismo, de ser otro caso si lo tomaría en consideración.

Una vez dentro de la maquina virtual:

1. Asignación de contraseña al usuario “root”

Por defecto no esta asignada contraseña al usuario root en el sistema y si tratamos de entrar con el comando “su” e intentamos ingresar con la contraseña del usuario previamente creado (“vpn”) tendremos un error de autenticación, véase en la Figura 10:

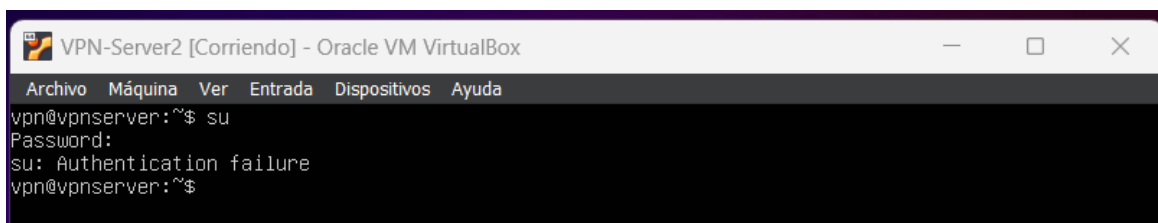


Figura 10: Error en la autenticación del servidor

Ya que en la creación de usuario no pidió una contraseña para el usuario root, para asignarle una utilizaremos el comando (Figura 11):

```
sudo passwd root
```

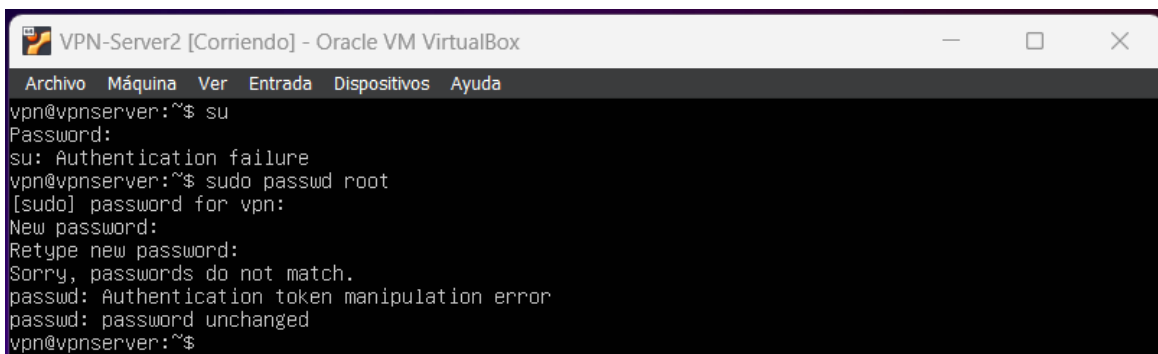


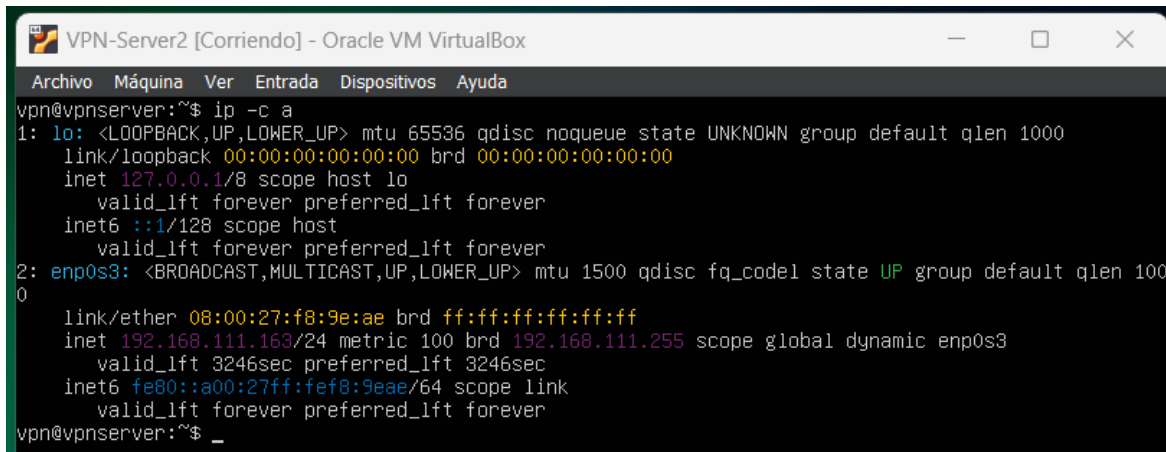
Figura 11: Asignación de contraseña al usuario root del servidor

Nos preguntara la contraseña del usuario con el que estamos en la sesión (“vpn”), después no pedirá una nueva contraseña (“New password”) y volvemos a confirmarla, con esto ya habremos asignado contraseña al usuario root. Esto es necesario hacerlo desde aquí ya que mediante vía ssh que es como yo trabajare en el desarrollo de esta practica este proceso no se puede hacer por esa vía, como se muestra en la Figura 11.

2. IP en mi Red

Para conocer que IP tiene el sistema, utiliza el comando (Figura 12):

```
ip -c a
```



```
VPN-Server2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
vpn@vpnserver:~$ ip -c 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f8:9e:ae brd ff:ff:ff:ff:ff:ff
    inet 192.168.111.163/24 metric 100 brd 192.168.111.255 scope global dynamic enp0s3
        valid_lft 3246sec preferred_lft 3246sec
    inet6 fe80::a00:27ff:fef8:9eae/64 scope link
        valid_lft forever preferred_lft forever
vpn@vpnserver:~$ _
```

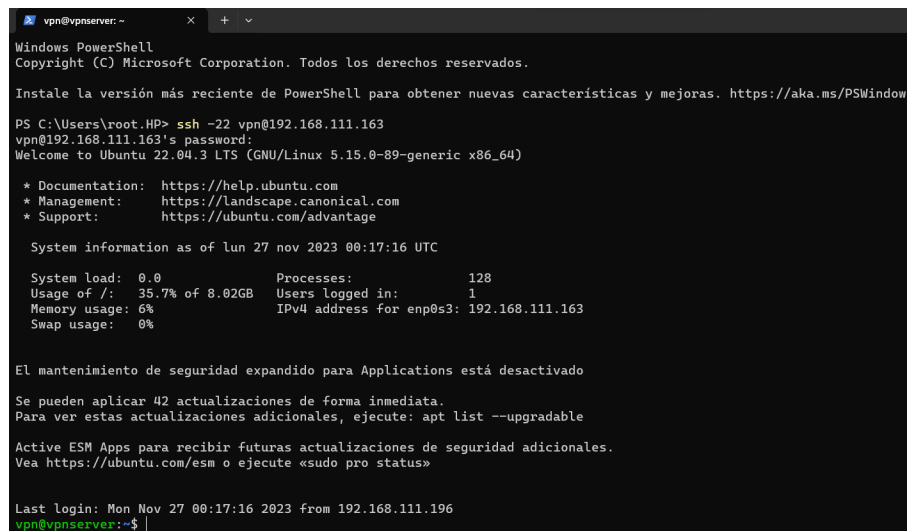
Figura 12: IP del sistema del servidor

Nota: Debo aclarar que esto la practica se realiza dentro de mi red domestica esto quiere decir que mi router (Modem) me asignara alguna IP dentro del rango de IPs que tiene mi red domestica de esto se encarga el DHCP, si es el caso contrario y no estas en este escenario tendrás que asignar tu manualmente una IP estática.

3. Conexión SSH:

En una terminal de sistema Windows, utilizaremos el comando (Figura 13):

```
ssh -22 vpn@192.168.111.163
```



```
vpn@vpnserver: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\root.HP> ssh -22 vpn@192.168.111.163
vpn@192.168.111.163's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-89-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of lun 27 nov 2023 00:17:16 UTC

System load:  0.0               Processes:    128
Usage of /:   35.7% of 8.02GB    Users logged in: 1
Memory usage: 6%               IPv4 address for enp0s3: 192.168.111.163
Swap usage:  0%

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 42 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Mon Nov 27 00:17:16 2023 from 192.168.111.196
vpn@vpnserver:~$ |
```

Figura 13: Conexión vía SSH al servidor

Este comando funciona solo para la Terminal que tiene Windows 11, en un CMD o PowerShell esto puede cambiar.

3.5. Instalación de OpenVPN

OpenVPN es una solución de software de código abierto que se utiliza para crear conexiones seguras punto a punto o de sitio a sitio en redes privadas virtuales (VPN). Funciona mediante la creación de un túnel cifrado a través del cual se pueden transmitir datos de manera segura a través de redes no seguras, como Internet.

Para la instalación del paquete OpenVPN en Ubuntu utilizamos el comando (Figura 14):

```
apt install openvpn -y
```

```
root@vpnsrvr:/home/vpn x + v
vpnsrvr:~$ su
Password:
root@vpnsrvr:/home/vpn# apt install openvpn -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  libpks11-helper1
Paquetes sugeridos:
  resolvconf openvpn-systemd-resolved easy-rsa
Se instalarán los siguientes paquetes NUEVOS:
  libpks11-helper1 openvpn
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 41 no actualizados.
Se necesita descargar 663 kB de archivos.
Se utilizarán 1.825 kB de espacio de disco adicional después de esta operación.
Des:1 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpks11-helper1 amd64 1.28-lubuntu0.22.04.1 [50,3 kB]
Des:2 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openvpn amd64 2.5.5-lubuntu3.1 [612 kB]
Descargados 663 kB en 3s (202 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete libpks11-helper1:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 74280 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libpks11-helper1_1.28-lubuntu0.22.04.1_amd64.deb ...
Desempaquetando libpks11-helper1:amd64 (1.28-lubuntu0.22.04.1) ...
Seleccionando el paquete openvpn previamente no seleccionado.
Preparando para desempaquetar .../openvpn_2.5.5-lubuntu3.1_amd64.deb ...
Desempaquetando openvpn (2.5.5-lubuntu3.1) ...
Configurando libpks11-helper1:amd64 (1.28-lubuntu0.22.04.1) ...
Configurando openvpn (2.5.5-lubuntu3.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn.service → /lib/systemd/system/openvpn.service.
Procesando disparadores para man-db (2.10.2-1) ...
Procesando disparadores para libc-bin (2.35-0ubuntu3.4) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.
```

Figura 14: Instalación de OpenVPN en el servidor

Previamente entre a usuario root con el comando `su` en la Figura 14.

3.6. Instalación de Easy-RSA

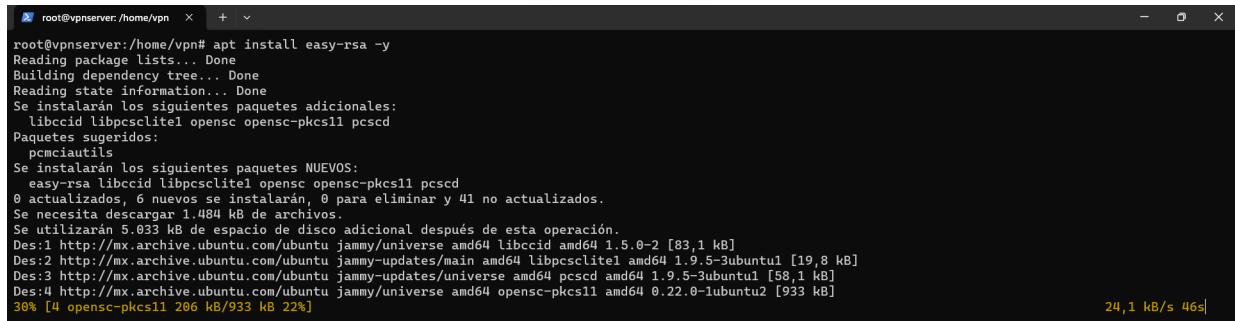
El paquete `easy-rsa` es una herramienta que simplifica la gestión de claves públicas y certificados para implementaciones de VPN, especialmente cuando se utiliza con OpenVPN.

Proporciona scripts que facilitan la generación, gestión y distribución de certificados digitales para autenticar clientes y servidores en una red VPN. Estos certificados son fundamentales para establecer conexiones seguras y verificar la identidad de los dispositivos que intentan acceder a la red protegida por la VPN.

En resumen, `easy-rsa` es una herramienta complementaria a OpenVPN que facilita la administración de certificados digitales, permitiendo una implementación más sencilla y segura de una infraestructura de clave pública para tu VPN.

Para la instalación del paquete de Easy-RSA utilizamos el comando (Figura15):

```
apt install easy-rsa -y
```



```
root@vpnserver: /home/vpn# apt install easy-rsa -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  libccid libpcsclite1 opensc opencsc-pkcs11 pcscd
Paquetes sugeridos:
  pcmciautils
Se instalarán los siguientes paquetes NUEVOS:
  easy-rsa libccid libpcsclite1 opensc opencsc-pkcs11 pcscd
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 41 no actualizados.
Se necesita descargar 1.484 kB de archivos.
Se utilizarán 5.833 kB de espacio de disco adicional después de esta operación.
Des:1 http://mx.archive.ubuntu.com/ubuntu jammy/universe amd64 libccid amd64 1.5.0-2 [83,1 kB]
Des:2 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpcsclite1 amd64 1.9.5-3ubuntu1 [19,8 kB]
Des:3 http://mx.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 pcscd amd64 1.9.5-3ubuntu1 [58,1 kB]
Des:4 http://mx.archive.ubuntu.com/ubuntu jammy/universe amd64 opencsc-pkcs11 amd64 0.22.0-1ubuntu2 [933 kB]
38% [4 opencsc-pkcs11 206 kB/933 kB 22%] 24,1 kB/s 46s
```

Figura 15: Instalación de Easy-RSA en el servidor

Se recomienda hacer una carpeta en `/home/vpn/` para en ella poder crear los certificados, firmas y llaves necesarias para el servidor y los clientes (Figura 16).

```
mkdir easy-rsa
```



```
root@vpnserver: /home/vpn# mkdir easy-rsa
root@vpnserver: /home/vpn# ls -l
total 4
drwxr-xr-x 2 root root 4096 nov 27 00:55 easy-rsa
root@vpnserver: /home/vpn#
```

Figura 16: Carpeta para la generación de llaves privadas del servidor

Posteriormente se hará un enlace simbólico (symlink) desde la secuencia de comandos de easysrsa que el paquete instaló, en el directorio `/easy-rsa` que acabo de crear, entro a la carpeta y declaro el enlace con (Figura 17):

```
ln -s /usr/share/easy-rsa/* .
```



```
root@vpnserver: /home/vpn# cd easy-rsa/
root@vpnserver: /home/vpn/easy-rsa# ln -s /usr/share/easy-rsa/* .
root@vpnserver: /home/vpn/easy-rsa# ls
easysrsa openssl-easysrsa.cnf vars.example x509-types
root@vpnserver: /home/vpn/easy-rsa#
```

Figura 17: Enlace simbólico para la estructura PKI del servidor

Al hacer esto se trasfieren los archivos de configuración necesarios del paquete que se instaló (Easy-RSA) a la carpeta de trabajo que hice (`/etc/openvpn/easy-rsa`). Este enlace se recomienda por que cuando se actualice el paquete y sus archivos de igual manera se actualice el contenido de esa carpeta, pero fácilmente también se puede copiar todos esos archivos a la carpeta que creamos.

3.7. Crear la PKI para OpenVPN

En el contexto de OpenVPN, la PKI es una parte fundamental que permite establecer un sistema de confianza y autenticación robusto entre los diferentes componentes de la red VPN. La PKI se encarga de generar, almacenar y distribuir los certificados y claves criptográficas necesarias para garantizar la seguridad y la autenticidad de la comunicación.

¿Qué es una PKI?

Una Infraestructura de Clave Pública (PKI) es un conjunto de hardware, software, políticas y procedimientos que se utilizan para crear, administrar, distribuir, utilizar, almacenar y revocar certificados digitales y claves criptográficas. En el contexto de OpenVPN, la PKI permite:

- Establecer una red de confianza entre los diferentes nodos de la red VPN.
- Facilitar la autenticación de usuarios y dispositivos.
- Proporcionar un marco de seguridad para el intercambio de datos cifrados.

Para crear la PKI en OpenVPN, se utiliza el comando `./easyrsa init-pki`, el cual inicializa la estructura de directorios y archivos necesarios para gestionar la PKI. Este paso es fundamental antes de generar los certificados y claves para los diferentes componentes de la VPN (Figura 18).

`./easyrsa init-pki`



```
root@vpnserver: /home/vpn/  x  +  v
root@vpnserver: /home/vpn/easy-rsa# ./easyrsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/vpn/easy-rsa/pki

root@vpnserver: /home/vpn/easy-rsa# |
```

Figura 18: Carpeta para la generación de llaves privadas del servidor

Al ejecutar este comando, se establecerá la infraestructura básica de la PKI, incluyendo directorios como `private`, `issued`, `reqs`, entre otros, donde se almacenarán las claves y certificados necesarios (Figura 19).



```
root@vpnserver: /home/vpn/  x  +  v
root@vpnserver: /home/vpn/easy-rsa# cd pki/
root@vpnserver: /home/vpn/easy-rsa/pki# ls
openssl-easyrsa.cnf  private  reqs  safessl-easyrsa.cnf
root@vpnserver: /home/vpn/easy-rsa/pki# |
```

Figura 19: Contenido de la carpeta `pki` del servidor

3.8. Crear una entidad de certificación (CA)

La Autoridad de Certificación (CA) raíz es un componente crítico en la infraestructura de clave pública de una red VPN. El comando `./easyrsa build-ca` se utiliza para generar esta CA raíz, que será responsable de firmar y emitir los certificados para los servidores, clientes y otros componentes de la red VPN. Para crear la Autoridad de Certificación raíz, se utiliza el siguiente comando:

`./easyrsa build-ca`

Al ejecutar este comando nos pedirá que ingresemos una contraseña, como se muestra en la Figura 20, esta contraseña debe de ser segura y privada, ya que se pedirá en repetidas ocasiones cuando queramos firmar los requerimientos necesarios para crear nuestros certificados y llaves privadas. También se le solicitará confirmar el nombre común (CN) de su CA, como también se muestra en la Figura 20. El nombre común es el que se usa para hacer referencia a esta máquina en el contexto de la entidad de certificación. Puede ingresar cualquier secuencia de caracteres para el nombre común de la CA, sin embargo, para hacerlo más simple presione ENTER para aceptar el nombre predeterminado.


```
root@vpnserver:/home/vpn/ - + v
root@vpnserver:/home/vpn/easy-rsa# ./easyrsa build-ca
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:
CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/vpn/easy-rsa/pki/ca.crt

root@vpnserver:/home/vpn/easy-rsa#
```

- **gen-req**: Indica que se está generando una solicitud de certificado.
- **server**: Es el nombre común (Common Name, CN) del servidor para el cual se está solicitando el certificado. Aquí se identifica como "server", pero este nombre puede variar dependiendo de la configuración específica.
- **nopass**: Este parámetro indica que no se requerirá una frase de contraseña para el certificado, lo que simplifica el proceso y facilita la automatización de la generación del certificado.

Al ejecutar este comando, se creará la solicitud de certificado para el servidor OpenVPN en la PKI configurada previamente. Esta solicitud será utilizada para generar el certificado que se asociará al servidor y que permitirá la autenticación dentro de la red VPN.

3.10. Generar el parámetro Diffie-Hellman para el servidor

Los parámetros de Diffie-Hellman son elementos cruciales en la configuración de seguridad de una red privada virtual (VPN). Estos parámetros consisten en un conjunto de valores matemáticos, como un número primo grande (p) y un número generador (g), utilizados para facilitar un intercambio seguro de claves entre un servidor y un cliente de VPN. La función principal de estos parámetros es permitir la generación de una clave secreta compartida sin necesidad de transmitirla directamente a través de la red, garantizando así la confidencialidad y la integridad de la comunicación. La elección adecuada de estos parámetros es fundamental para mantener la seguridad de la VPN, equilibrando la robustez criptográfica con la eficiencia computacional. Una actualización periódica de estos parámetros es recomendable para adaptarse a las evoluciones en las técnicas de cifrado y mantener un nivel óptimo de seguridad en la red.

Para este paso no ubicaremos en el directorio `/etc/openvpn/server/keys/` y utilizaremos el siguiente comando para generar el parámetros Diffie-Hellman (DH) de 2048 bits y guardarlos en un archivo llamado `dh2048.pem`, como se muestra en la Figura 22.

```
openssl dhparam -out dh2048.pem 2048
```



Figura 22: Generación del parámetro Diffie-Hellman para el servidor

3.11. Firma del Certificado del Servidor OpenVPN por la Autoridad de Certificación (CA)

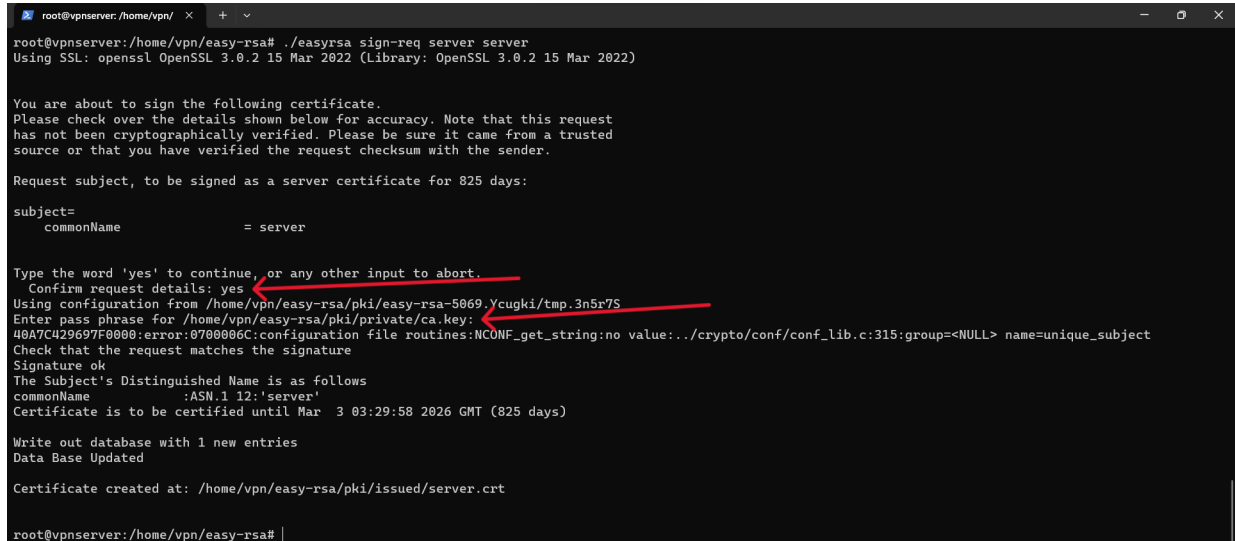
Una vez generada la solicitud de certificado del servidor OpenVPN, el siguiente paso es firmar dicha solicitud por parte de la Autoridad de Certificación (CA) para obtener el certificado válido para el servidor. Con el comando:

```
./easyrsa sign-req server server
```

- El primer **server** se refiere al nombre de la solicitud de certificado que se desea firmar.
- El segundo **server** es el nombre con el que se creará el archivo de certificado. Puede coincidir con el nombre de la solicitud de certificado o ser diferente según la organización y convenciones de nomenclatura establecidas.

Al ejecutar este comando, como se muestra en la Figura 23, la Autoridad de Certificación (CA) raíz valida la solicitud de certificado del servidor y firma el certificado, lo que lo convierte en un certificado válido y confiable para el servidor OpenVPN identificado por el nombre especificado.

Se le solicitará que verifique que la solicitud provenga de una fuente de confianza. Escriba yes y ingrese la contraseña para confirmar, esta contraseña sera la cual se hablo en la sección 3.8 (Figura 23).



```
root@vpnserver:/home/vpn/ X + v
root@vpnserver:/home/vpn/easy-rsa# ./easyrsa sign-req server server
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 825 days:

subject=
  commonName = server

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
Using configuration from /home/vpn/easy-rsa/pki/easy-rsa-5069.Ycugki/tmp.3n5r7S
Enter pass phrase for /home/vpn/easy-rsa/pki/private/ca.key:
40A7C29697F0080:error:0700006C:configuration file routines:NCNF_get_string:no value:../crypto/conf/conf_lib.c:315:group=<NULL> name=unique_subject
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :ASN.1 12:'server'
Certificate is to be certified until Mar  3 03:29:58 2026 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/vpn/easy-rsa/pki/issued/server.crt

root@vpnserver:/home/vpn/easy-rsa#
```

Figura 23: Solicitud de firma del Certificado para el Servidor OpenVPN

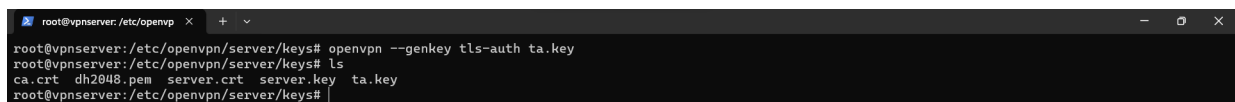
Una vez firmado, este certificado puede ser utilizado por el servidor para autenticarse con éxito dentro de la red VPN, asegurando así una comunicación segura y autenticada entre los diferentes componentes.

3.12. Generación de Clave de Autenticación TLS (tls-auth)

Es fundamental establecer una capa adicional de seguridad mediante la generación de una clave de autenticación TLS (tls-auth). Esta clave, generada con el comando `openvpn --genkey tls-auth ta.key`, desempeña un papel crucial en la protección contra ciertos ataques y en la verificación de la autenticidad de las conexiones entrantes. Al implementar esta medida de seguridad, fortalecemos la integridad y la confidencialidad de nuestra red VPN.

Esto lo podemos generar dicha llave nos aseguramos de estar ubicados en el directorio `/etc/openvpn/server/keys/` y ejecutamos el comando (Figura 24):

`openvpn --genkey tls-auth ta.key`



```
root@vpnserver:/etc/openvpn X + v
root@vpnserver:/etc/openvpn/server/keys# openvpn --genkey tls-auth ta.key
root@vpnserver:/etc/openvpn/server/keys# ls
ca.crt dh2048.pem server.crt server.key ta.key
root@vpnserver:/etc/openvpn/server/keys#
```

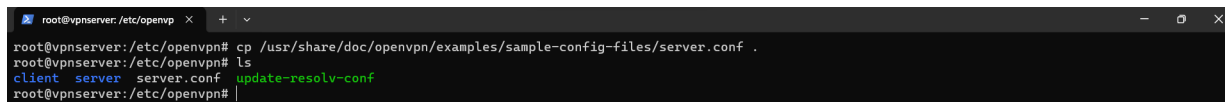
Figura 24: Generación de la Clave de Autenticación TLS para el servidor y los clientes

3.13. Configuración del entorno de ejecución del servidor

Existen muchas maneras de configurar un servidor y yo haré la que considero mejor en el aspecto de entender el procedimiento, utilizare el archivo para la configuración del servidor que tiene OpenVPN como ejemplo. Este posteriormente lo editare según mi configuración.

Entramos a la carpeta de ejecución del servidor vasado en OpenVPN `/etc/openvpn/` y copiamos el archivo de ejemplo con el que viene OpenVPN:(Figura 25):

```
cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf .
```



```
root@vpnserver:/etc/openvpn# cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf .
root@vpnserver:/etc/openvpn# ls
client  server  server.conf  update-resolv-conf
root@vpnserver:/etc/openvpn#
```

Figura 25: Copiado del archivo de configuración del servidor

Yo recomiendo ordenar todas las llaves y certificados que se creen en una directorio “/keys” dentro de la carpeta `/etc/openvpn/server` del entorno de ejecución del servidor, esta carpeta estará vacía, esta pensada para nosotros añadir allí nuestras llaves que vallamos creando (Figura 26):

```
mkdir keys
```



```
root@vpnserver:/etc/openvpn# ls
client  server  server.conf  update-resolv-conf
root@vpnserver:/etc/openvpn# cd server
root@vpnserver:/etc/openvpn/server# mkdir keys
root@vpnserver:/etc/openvpn/server# ls
keys
root@vpnserver:/etc/openvpn/server#
```

Figura 26: Crear carpeta para organizar los archivos del servidor

3.13.1. Direcciones del Servidor

Aprovecharemos esta sección para configurar las direcciones de nuestras llaves y certificados que futura mente estarán en este directorio (`/etc/openvpn/server/keys`) regresa dos directorios atrás para estar en `/etc/openvpn/` donde se encuentra el archivo `server.conf`, véase en la Figura 27:



```
root@vpnserver:/etc/openvpn/server/keys# cd ../../
root@vpnserver:/etc/openvpn# ls
client  server  server.conf  update-resolv-conf
root@vpnserver:/etc/openvpn#
```

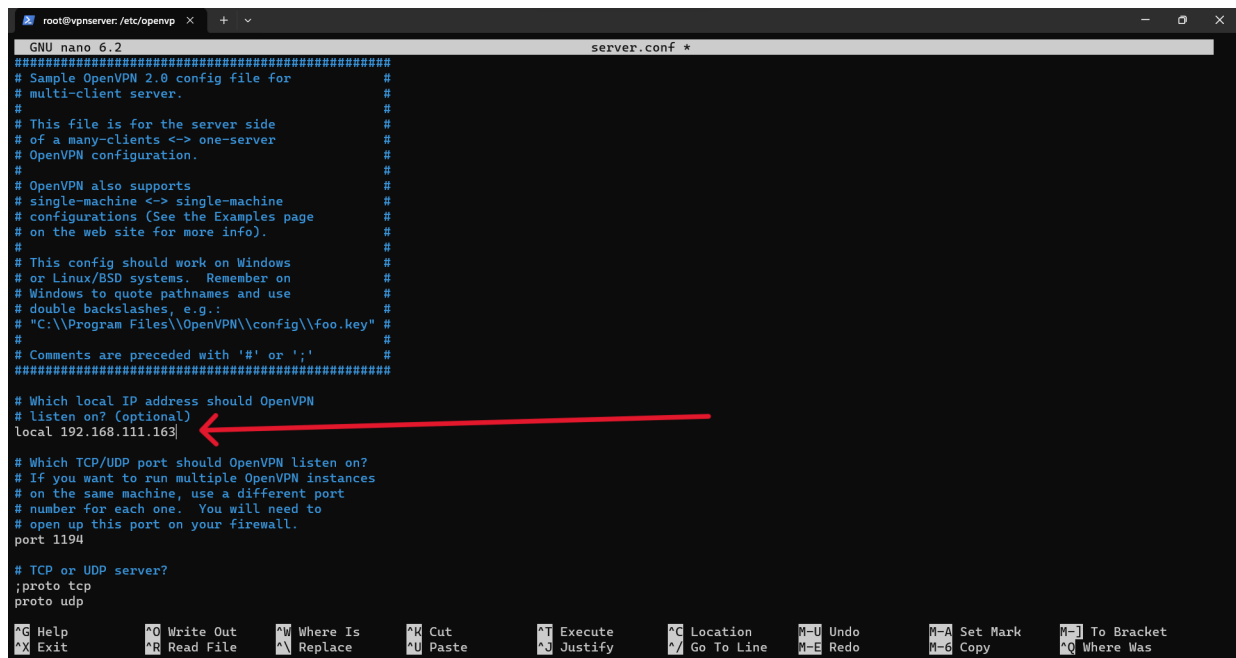
Figura 27: Retroceder al directorio `/etc/openvpn` en el servidor

Asegurándonos de ser usuarios root abriremos el archivo `server.conf` con el comando:

```
nano server.conf
```

1. Dirección IP

Asignaremos la dirección IP por la cual escuchara OpenVPN, la cual sera una dirección IP que nos asigno la red en la que estamos o la que nosotros configuramos como IP del sistema (Linux), buscamos la linea que esta comentada el archivo como “`;local a.b.c.d`”, la descomentaremos borrando los `;` que tiene al inicio y posteriormente de `local` le asignaremos nuestra dirección IP, como se muestra en la Figura 28:



```
GNU nano 6.2 server.conf *
#####
# Sample OpenVPN 2.0 config file for
# multi-client server.
#
# This file is for the server side
# of a many-clients <-> one-server
# OpenVPN configuration.
#
# OpenVPN also supports
# single-machine <-> single-machine
# configurations (See the Examples page
# on the web site for more info).
#
# This config should work on Windows
# or Linux/BSD systems. Remember on
# Windows to quote pathnames and use
# double backslashes, e.g.:
# "C:\\Program Files\\OpenVPN\\config\\foo.key"
#
# Comments are preceded with '#' or ';'
#####
# Which local IP address should OpenVPN
# listen on? (optional)
local 192.168.111.163
# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one. You will need to
# open up this port on your firewall.
port 1194
# TCP or UDP server?
;proto tcp
proto udp

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-J To Bracket
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^Q Justify    ^/ Go To Line M-E Redo      M-G Copy      ^Q Where Was
```

Figura 28: Configuración de la dirección IP del servidor por la que escuchara OpenVPN

2. Direcciones de los Certificados y Llaves privadas

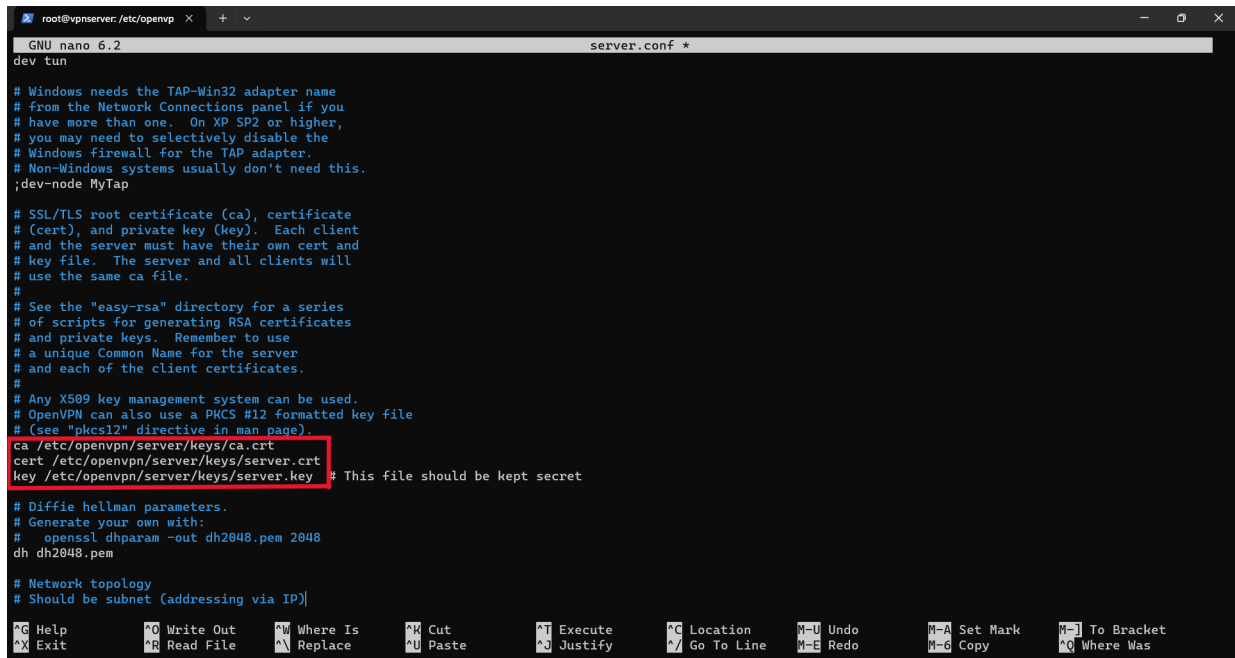
Posteriormente buscaremos la sección “SSL/TLS certificación root (ca), certificación (cert) y llaves privadas (key)”, si esto se te complica dentro del gestor de archivos para sistemas operativos Linux, nano, existe la función de buscar (Where is) presionando **Ctrl + W** y posteriormente busca “ca”, una vez hecho encontraras el apartado de configuración que menciono y observarás estas 3 líneas:

```
ca ca.crt
cert server.crt
key server.key
```

■ Certificados y Llave del servidor

Asignar las rutas de cada uno de esos certificados y llave, no están en ese directorio donde se encuentra el archivo en cuestión, pero recuerda que los certificados y llave privada estarán en la carpeta `/etc/openvpn/server/keys/`, así que asigna las direcciones de la siguiente manera (véase en la Figura 29):

```
ca /etc/openvpn/server/keys/ca.crt
cert /etc/openvpn/server/keys/server.crt
key /etc/openvpn/server/keys/server.key
```



```
GNU nano 6.2 server.conf *
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one. On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key). Each client
# and the server must have their own cert and
# key file. The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys. Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca /etc/openvpn/server/keys/ca.crt
cert /etc/openvpn/server/keys/server.crt
key /etc/openvpn/server/keys/server.key # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh2048.pem 2048
dh dh2048.pem

# Network topology
# Should be subnet (addressing via IP)

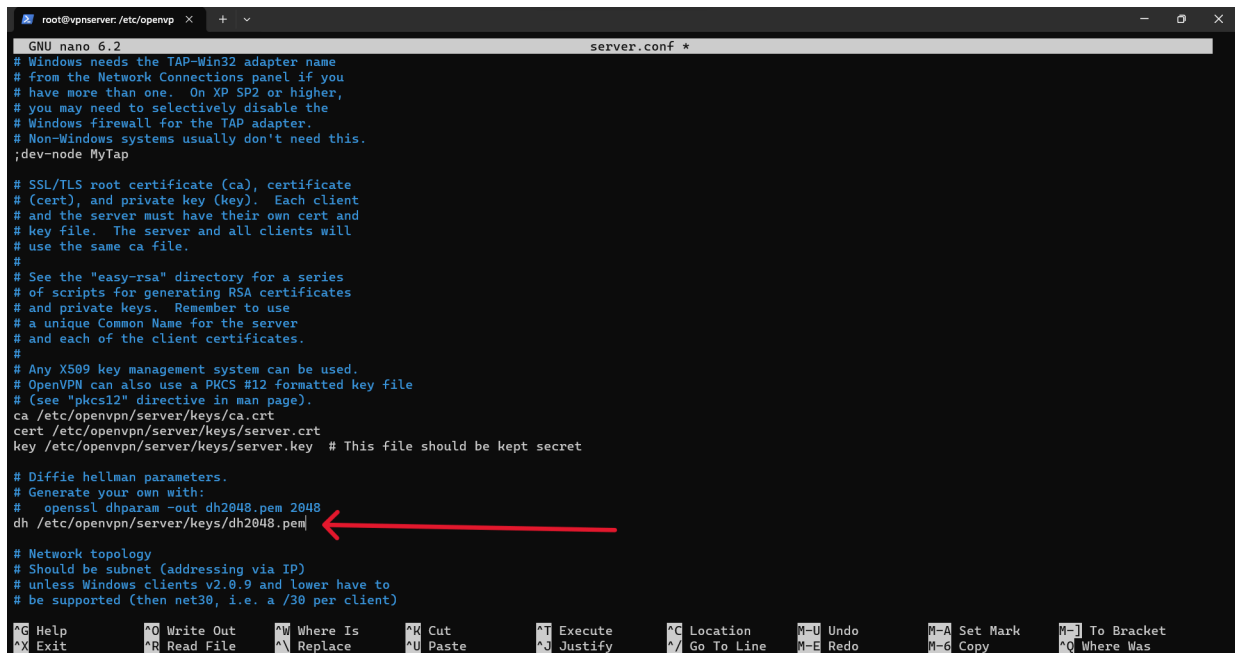
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark  M-J To Bracket
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_/ Go To Line M-E Redo     M-G Copy      ^Q Where Was
```

Figura 29: Configuración de las dirección de certificados y llave privada del servidor

■ Diffie-Hellman

También utilizaremos el parámetros Diffie-Hellman, así que pondremos la dirección del fichero keys, como se muestra en la Figura 30:

```
dh /etc/openvpn/server/keys/dh2048.pem
```



```
GNU nano 6.2 server.conf *
# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one. On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key). Each client
# and the server must have their own cert and
# key file. The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys. Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca /etc/openvpn/server/keys/ca.crt
cert /etc/openvpn/server/keys/server.crt
key /etc/openvpn/server/keys/server.key # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh2048.pem 2048
dh /etc/openvpn/server/keys/dh2048.pem

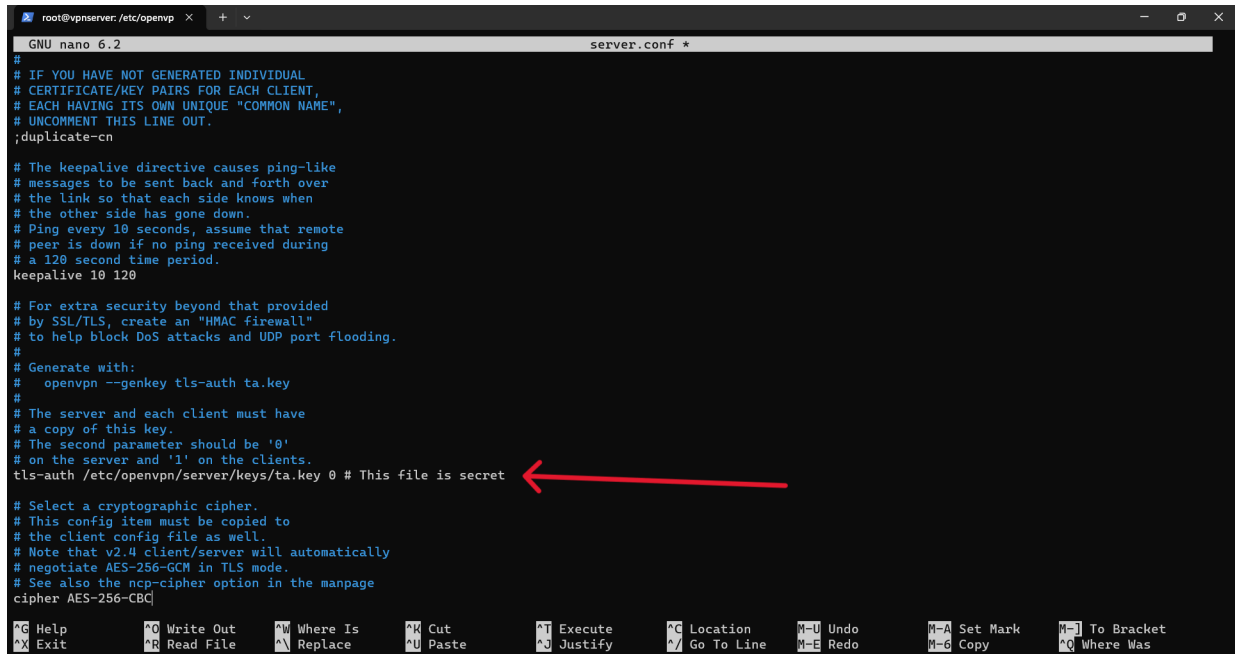
# Network topology
# Should be subnet (addressing via IP)
# unless Windows clients v2.0.9 and lower have to
# be supported (then net30, i.e. a /30 per client)

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark  M-J To Bracket
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_/ Go To Line M-E Redo     M-G Copy      ^Q Where Was
```

Figura 30: Configuración de la dirección para el parámetro Diffie-Hellman del servidor

■ tls-auth

De igual manera buscaremos en el archivo la línea `tls-auth ta.key 0` y asignaremos su dirección, recordemos lo que venimos hablando todas estas se encontraran en el directorio `/etc/openvpn/server/keys/` así que le indicamos la ruta, como se muestra en la Figura 31:



```
GNU nano 6.2 server.conf
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey tls-auth ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
tls-auth /etc/openvpn/server/keys/ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  ^U Undo      ^A Set Mark  ^J To Bracket
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_/ Go To Line ^E Redo      ^M Copy      ^Q Where Was
```

Figura 31: Configuración de la dirección para la llave privada `ta.key` en el servidor

■

3.14. Tránsito de claves al servidor

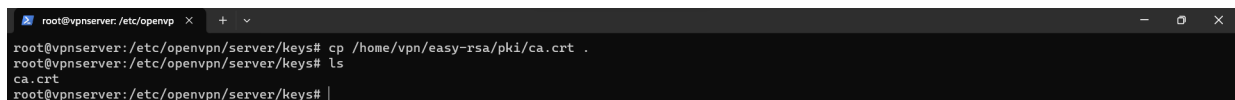
1. ca (ca.crt)

En la sección 3.8 creamos una entidad certificadora (CA) y nos generaron los certificados y claves necesarios para establecer la CA raíz, transferiremos el archivo que nos valida la solicitud de certificado del servidor y firma el certificado, dicho archivo se aloja en:

`/home/vpn/easy-rsa/pki/ca.crt`

Nos aseguramos de estar en el directorio `/etc/openvpn/server/keys/`, y copiamos el archivo con el comando (Figura 32):

```
cp /home/vpn/easy-rsa/pki/ca.crt .
```



```
root@vpnserver:/etc/openvpn X + v
root@vpnserver:/etc/openvpn/server/keys# cp /home/vpn/easy-rsa/pki/ca.crt .
root@vpnserver:/etc/openvpn/server/keys# ls
ca.crt
root@vpnserver:/etc/openvpn/server/keys# |
```

Figura 32: Tránsito del certificado `ca.crt` en el servidor

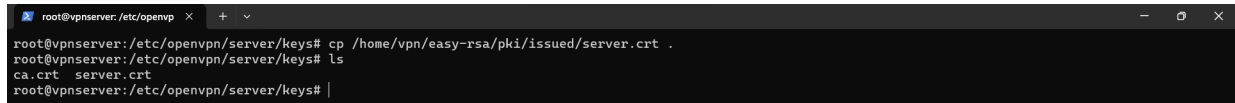
2. crt (server.crt)

En la sección 3.11 donde validamos la solicitud de certificado del cliente y firmamos dicho certificado, el cual se aloja en:

Certificate created at: /home/vpn/easy-rsa/pki/issued/server.crt

Necesitamos transferirla al directorio /etc/openvpn/server/keys/, así que nos aseguramos de estar en el directorio en cuestión y la copiamos, con el comando (Figura 33):

```
cp /home/vpn/easy-rsa/pki/issued/server.crt .
```



```
root@vpnserver:/etc/openvp X + v
root@vpnserver:/etc/openvpn/server/keys# cp /home/vpn/easy-rsa/pki/issued/server.crt .
root@vpnserver:/etc/openvpn/server/keys# ls
ca.crt  server.crt
root@vpnserver:/etc/openvpn/server/keys# |
```

Figura 33: Traslferencia del certificado para el servidor

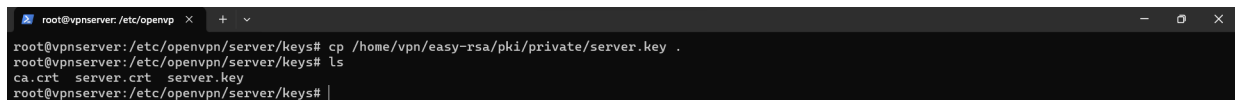
3. key (server.key)

En la sección 3.9 generamos el requerimiento necesario para el servidor y si recuerdas nos dio las direcciones donde se generaron el requerimiento y la llave del servidor:

```
req: /home/vpn/easy-rsa/pki/reqs/server.req
key: /home/vpn/easy-rsa/pki/private/server.key
```

Nos aseguraremos de estar en la carpeta /etc/openvpn/server/keys/ y copiaremos la llave del servidor a la carpeta en cuestión (Figura 34):

```
cp /home/vpn/easy-rsa/pki/private/server.key .
```



```
root@vpnserver:/etc/openvp X + v
root@vpnserver:/etc/openvpn/server/keys# cp /home/vpn/easy-rsa/pki/private/server.key .
root@vpnserver:/etc/openvpn/server/keys# ls
ca.crt  server.crt  server.key
root@vpnserver:/etc/openvpn/server/keys# |
```

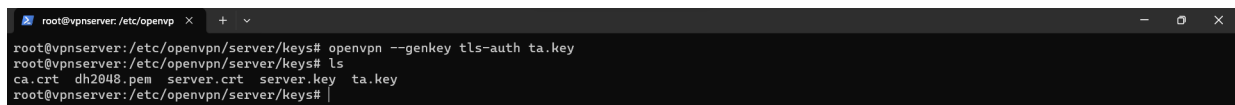
Figura 34: Traslferencia de la llave del servidor a la carpeta /etc/openvpn/server/keys/ en el servidor

4. tls-auth (ta.key)

En la sección 3.12 generamos la clave de autenticación necesario para que se conecte el cliente a nuestro servidor de forma segura y privada, esta fue generada en el directorio:

/etc/openvpn/server/keys/

Así que no requerimos copiar dicha llave a otro lugar, esta llave es privada, solamente la puede tener el servidor y el cliente y no vede ser publica de ninguna manera (Figura 35):



```
root@vpnserver:/etc/openvp X + v
root@vpnserver:/etc/openvpn/server/keys# openvpn --genkey tls-auth ta.key
root@vpnserver:/etc/openvpn/server/keys# ls
ca.crt  dh2048.pem  server.crt  server.key  ta.key
root@vpnserver:/etc/openvpn/server/keys# |
```

Figura 35: Ubicación de la clave de autenticación TLS

3.15. Configuración de la Máquina Virtual del cliente

Para la configuración del cliente lo aremos para uno en Linux, esta configuración puede ser general para cualquier distribución pero en este caso se ejemplificara con uno en Ubuntu, no mostrare configuración de la maquina para el cliente, porque esta sera según tus necesidades o según como configuraste tu sistema, sin embargo esta es mi configuración para esta ejemplificación:

1. Nombre y ISO de la maquina cliente

Utilizare la misma ISO pero no es necesario que tu utilices la misma, si ya cuentas con otra maquina con sistema operativo Linux, puedes obviar esta sección (Figura 36).

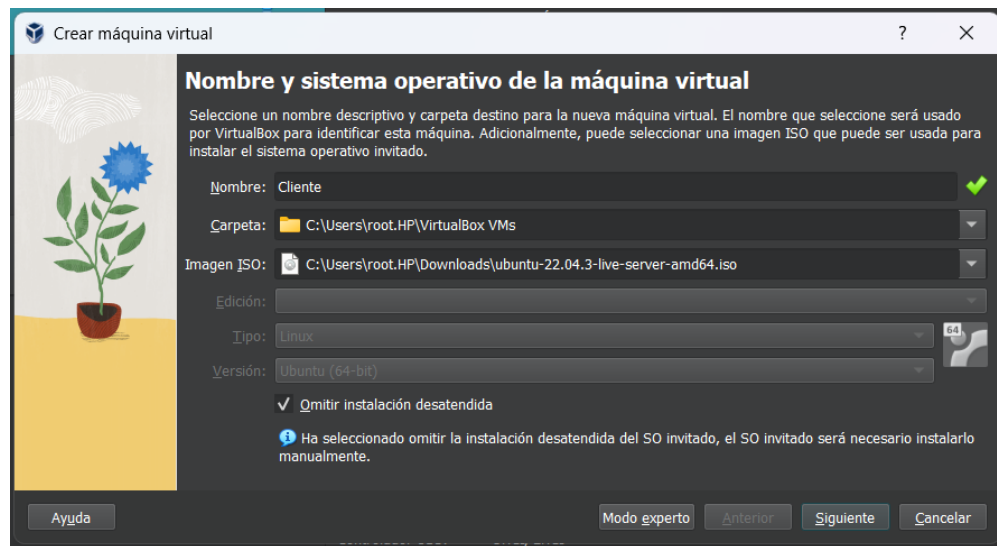


Figura 36: Nombre e ISO de la maquina cliente

2. Hardware de la maquina cliente

Para el cliente utilizó los recursos mínimos por fines prácticos nada mas, 2 GB de RAM (2048MB) y 1 solo proceso del CPU (Figura 37).

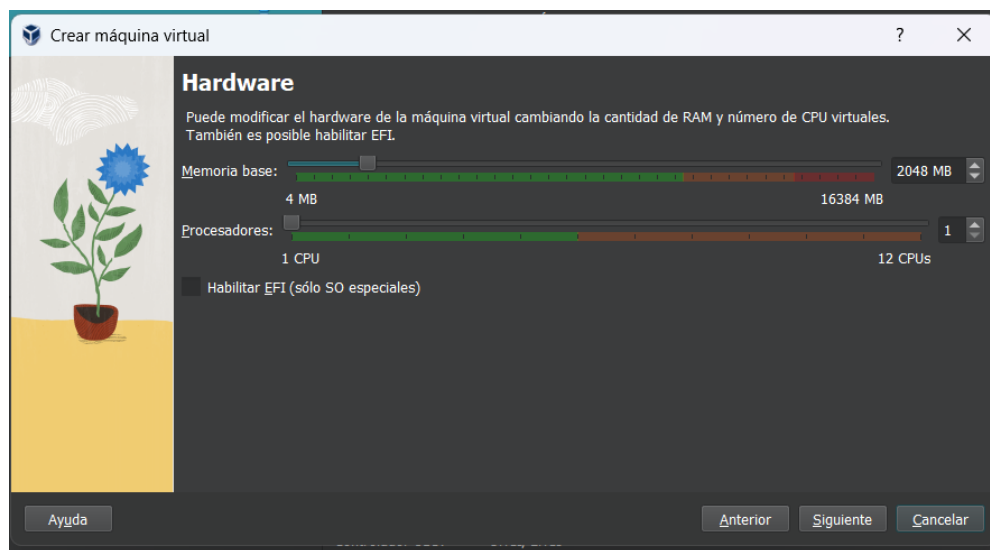


Figura 37: Asignación de hardware de la maquina cliente

3. **Tamaño del disco de la maquina cliente**

Asigne lo mínimo requerido para este sistema operativo (Ubuntu 22.04.3) como se muestra en la Figura 38

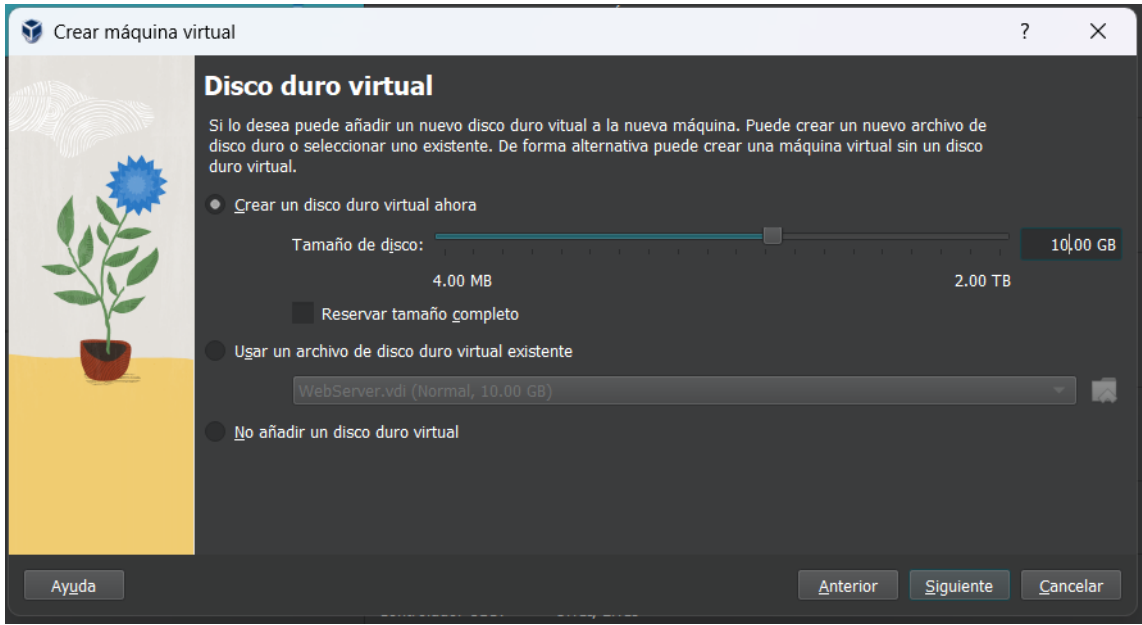


Figura 38: Tamaño del disco virtual de la maquina cliente

4. **Resumen de configuración de la máquina cliente**

Este seria el resumen de configuraciones necesarias y mínimas para esta ejemplificación de un cliente para un servidor de VPNs (Figura 39).

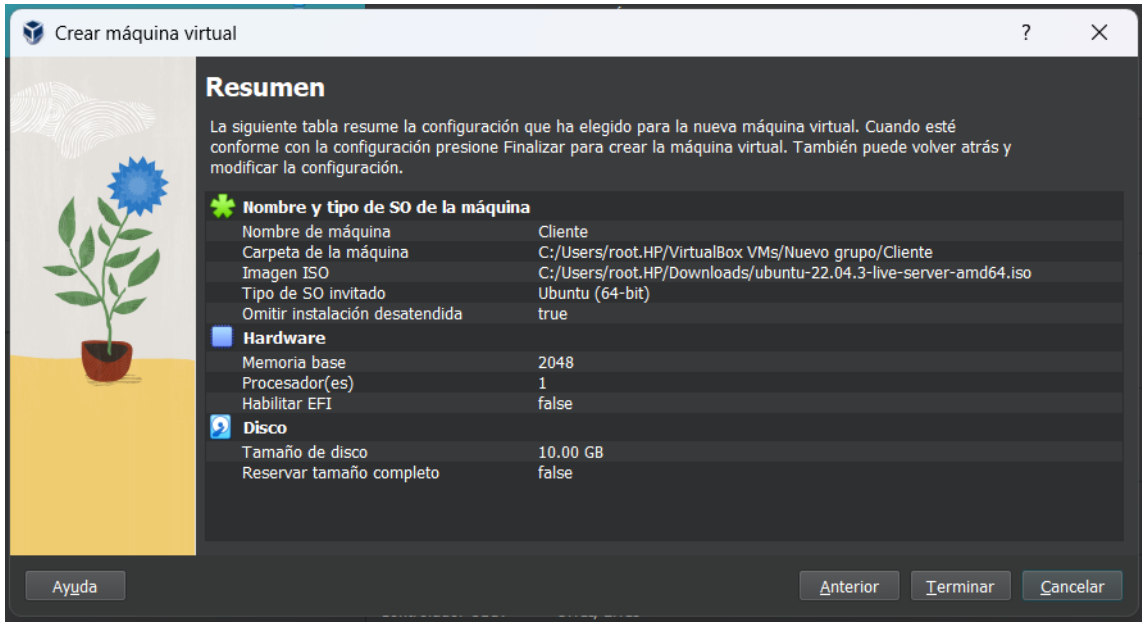


Figura 39: Resumen de configuraciones de la maquina cliente

Nota: asegurate de que si estas virtualizando un sistema operativo en Windows como yo, asegurate de configurar correctamente tu adaptador en modo puente como se mostró en la sección 3.2.

3.16. Instalación del Sistema Operativo del Cliente

Para esta sección también considero que no es necesario hacer un proceso de instalación del sistema operativo debido a que no incluyo ningún ajuste adicional o fuera de lo normal, como en la sección anterior 3.1, sin embargo este fue mi elecciones del nombre de sistema y usuario (Figura 40).

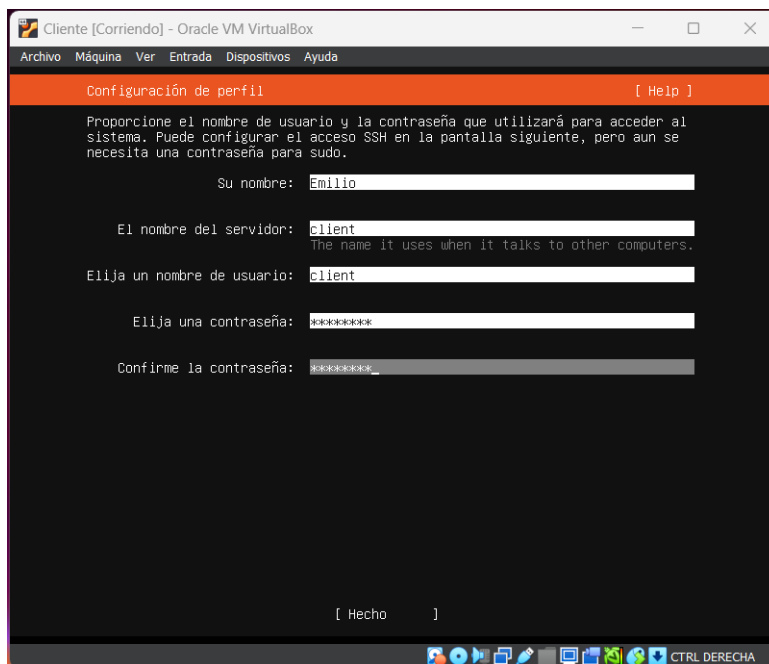


Figura 40: Creación de usuario y contraseña para el cliente

No es necesario instalar ningún paquete ni dependencia extra, esto a pesar de que trabajare mediante SSH por comodidad (Figura 41).

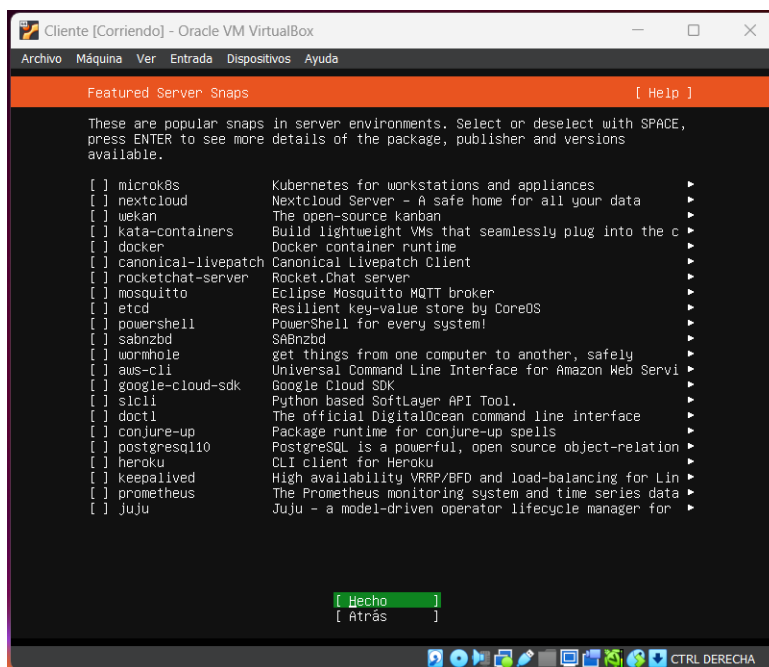


Figura 41: Paquetes requeridos para el sistema del cliente

3.17. SSH (Cliente)

También trabaje mediante SSH por comodidad, para empezar a trabajar vía SSH con la maquina del cliente, como en la sección 3.4, debemos de primero asignar una contraseña al usuario `root` en la maquina cliente.

Una vez dentro de la maquina virtual:

1. Asignación de contraseña al usuario “root”

Por defecto no esta asignada contraseña al usuario `root` en el sistema y si tratamos de entrar con el comando “`su`” e intentamos ingresar con la contraseña del usuario previamente creado (“`vpn`”) tendremos un error de autenticación, véase en la Figura 42

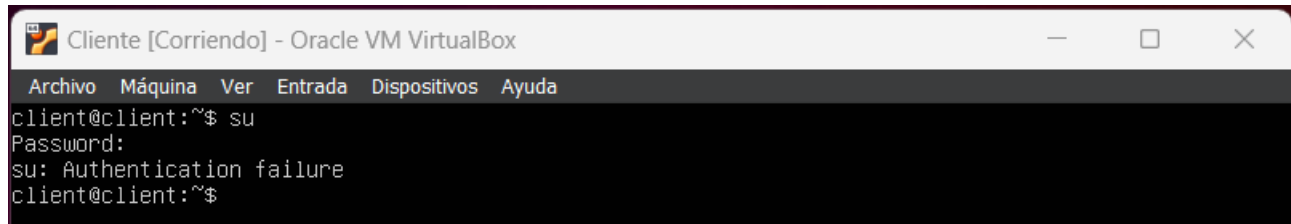


Figura 42: Error de autenticación del cliente

Ya que en la creación de usuario no pidió una contraseña para el usuario `root`, para asignarle una utilizaremos el comando (Figura 43):

```
sudo passwd root
```

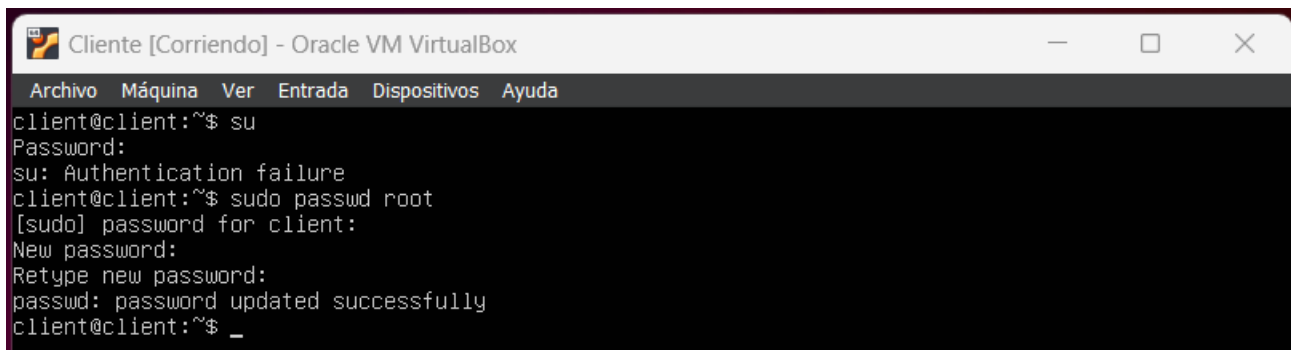


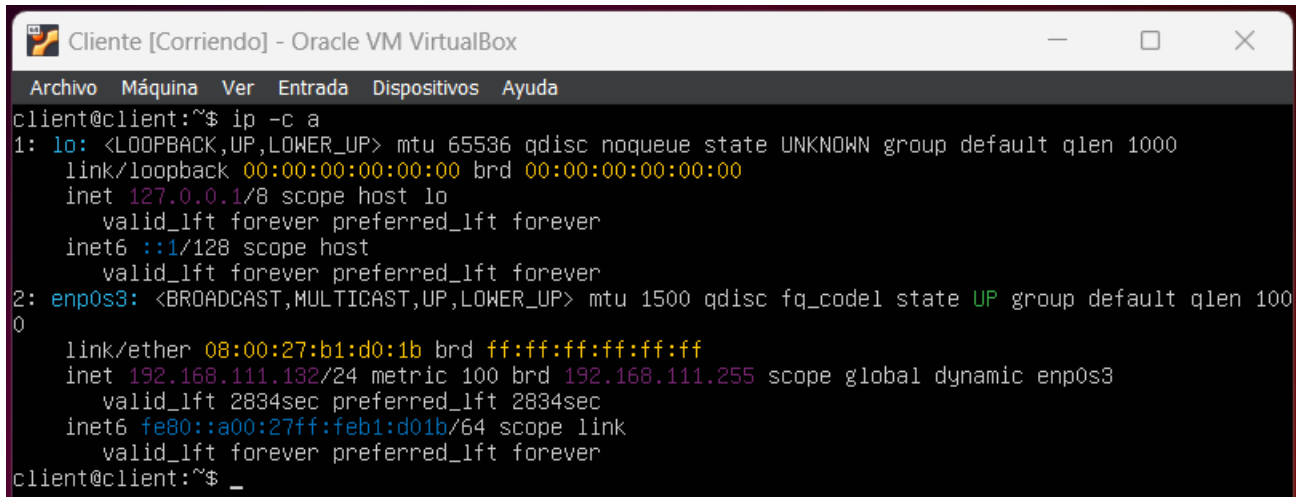
Figura 43: Asignación de contraseña para el usuario `root` del cliente

Nos preguntara la contraseña del usuario con el que estamos en la sesión (“`client`”), después no pedirá una nueva contraseña (“`New password`”) y volvemos a confirmarla, con esto ya habremos asignado contraseña al usuario `root`. Esto es necesario hacerlo desde aquí ya que mediante vía ssh que es como yo trabajare en el desarrollo de esta practica este proceso no se puede hacer por esa vía, como se muestra en la Figura 43.

2. IP en mi Red

Para conocer que IP tiene el sistema, utiliza el comando (Figura 44):

```
ip -c a
```



```
cliente@cliente:~$ ip -c 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b1:d0:1b brd ff:ff:ff:ff:ff:ff
    inet 192.168.111.132/24 metric 100 brd 192.168.111.255 scope global dynamic enp0s3
        valid_lft 2834sec preferred_lft 2834sec
    inet6 fe80::a00:27ff:feb1:d01b/64 scope link
        valid_lft forever preferred_lft forever
cliente@cliente:~$ _
```

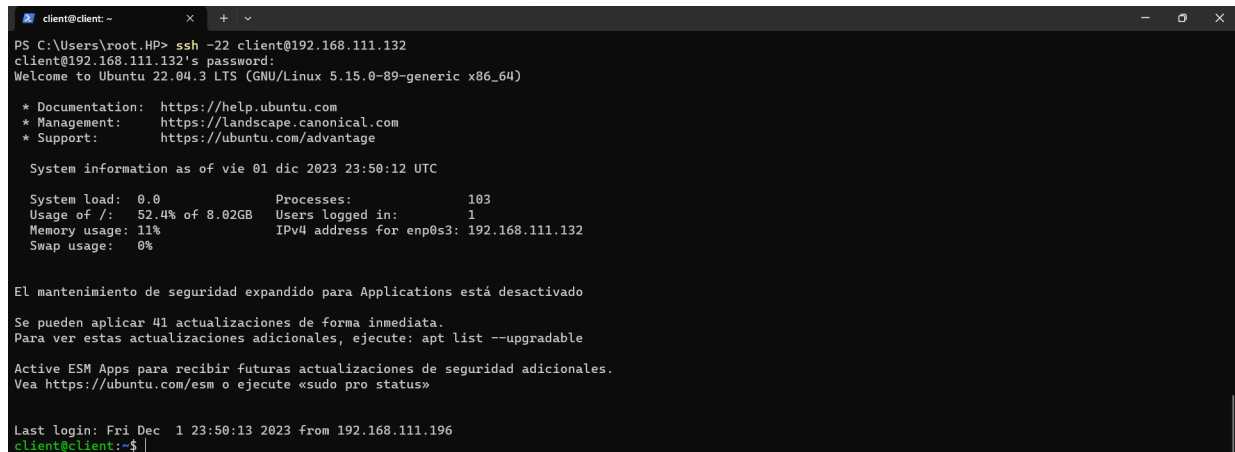
Figura 44: IP del sistema del cliente

Nota: Como en la sección 3.4, debo aclarar que esto la practica se realiza dentro de mi red domestica esto quiere decir que mi router (Modem) me asignara alguna IP dentro del rango de IPs que tiene mi red domestica de esto se encarga el DHCP, si es el caso contrario y no estas en este escenario tendrás que asignar tu manualmente una IP estática.

3. Conexión SSH

En una terminal de sistema Windows, utilizaremos el comando (Figura 45):

```
ssh -22 cliente@192.168.111.132
```



```
PS C:\Users\root.HP> ssh -22 cliente@192.168.111.132
cliente@192.168.111.132's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-89-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of vie 01 dic 2023 23:50:12 UTC

System load:  0.0               Processes:    103
Usage of /:   52.4% of 8.02GB   Users logged in: 1
Memory usage: 11%              IPv4 address for enp0s3: 192.168.111.132
Swap usage:   0%

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 41 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Fri Dec  1 23:50:13 2023 from 192.168.111.196
cliente@cliente:~$
```

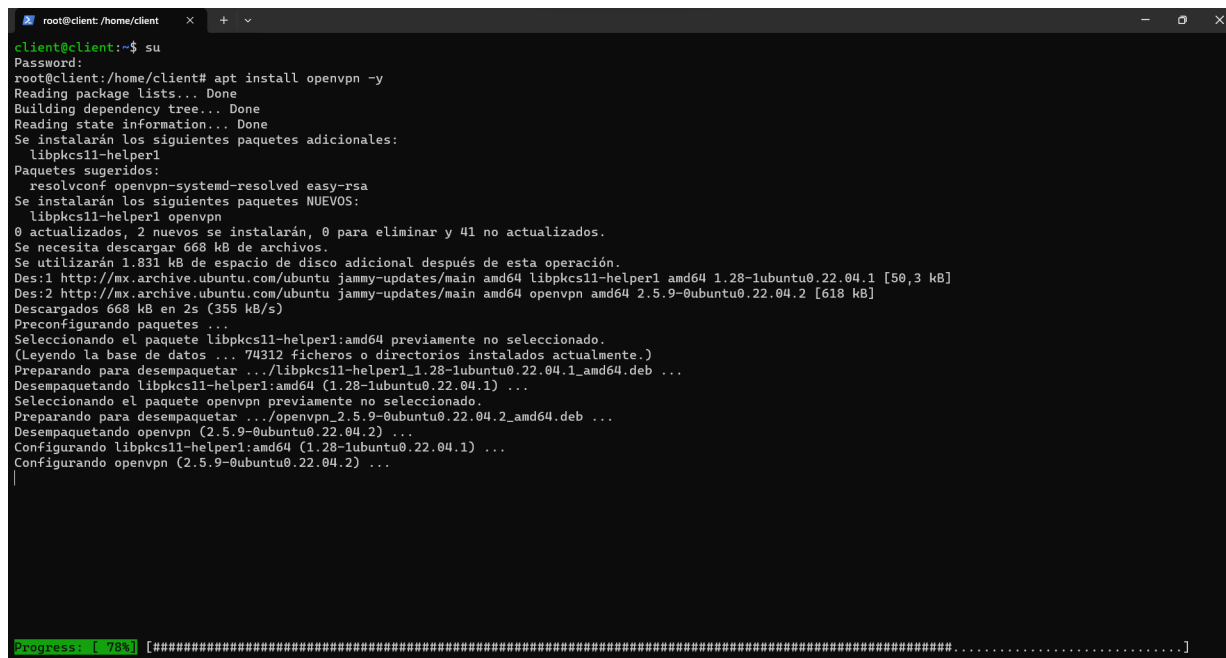
Figura 45: Conexión vía SSH al cliente

Este comando funciona solo para la Terminal que tiene Windows 11, en un CMD o PowerShell esto puede cambiar.

3.18. Instalación de OpenVPN en el Cliente

Para la instalación del paquete OpenVPN en Ubuntu utilizamos el comando (Figura 46):

```
apt install openvpn -y
```



```
root@client:/home/client x + v
client@client:~$ su
Password:
root@client:/home/client# apt install openvpn -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  libpks11-helper1
Paquetes sugeridos:
  resolvconf openvpn-systemd-resolved easy-rsa
Se instalarán los siguientes paquetes NUEVOS:
  libpks11-helper1 openvpn
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 41 no actualizados.
Se necesita descargar 668 kB de archivos.
Se utilizarán 1.831 kB de espacio de disco adicional después de esta operación.
Des:1 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpks11-helper1 amd64 1.28-1ubuntu0.22.04.1 [50,3 kB]
Des:2 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openvpn amd64 2.5.9-0ubuntu0.22.04.2 [618 kB]
Descargados 668 kB en 2s (355 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete libpks11-helper1:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 74312 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libpks11-helper1_1.28-1ubuntu0.22.04.1_amd64.deb ...
Desempaquetando libpks11-helper1:amd64 (1.28-1ubuntu0.22.04.1) ...
Seleccionando el paquete openvpn previamente no seleccionado.
Preparando para desempaquetar .../openvpn_2.5.9-0ubuntu0.22.04.2_amd64.deb ...
Desempaquetando openvpn (2.5.9-0ubuntu0.22.04.2) ...
Configurando libpks11-helper1:amd64 (1.28-1ubuntu0.22.04.1) ...
Configurando openvpn (2.5.9-0ubuntu0.22.04.2) ...
Progress: [ 78%] [#####.....]
```

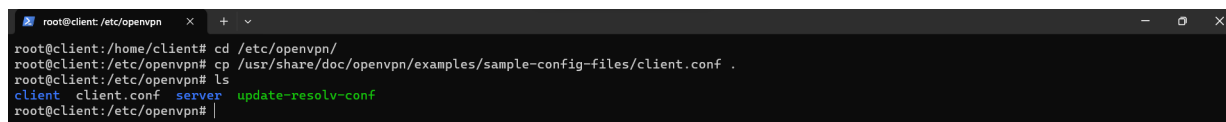
Figura 46: Instalación de OpenVPN en el cliente

Previamente entre a usuario root con el comando `su` en la Figura 46.

3.19. Configuración del entorno de ejecución del cliente

Una vez instalado el paquete de OpenVPN en el cliente, de manera similar a el servidor, no dirigiremos a la carpeta del paquete `/etc/openvpn/` y copiaremos el archivo de configuración del cliente (`client.conf`) que viene de ejemplo al instalar el paquete de OpenVPN, con el comando (Figura 47):

```
cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf .
```



```
root@client:/etc/openvpn x + v
root@client:/home/client# cd /etc/openvpn/
root@client:/etc/openvpn# cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf .
root@client:/etc/openvpn# ls
client client.conf server update-resolv-conf
root@client:/etc/openvpn#
```

Figura 47: Copiado del archivo de configuración del cliente

Nos ubicaremos en la carpeta `/etc/openvpn/client` y crearemos una carpeta llamada `/keys` para de igual manera que con el servidor mantener ordenado el entorno del alojamiento de dichas llaves (Figura 48).



```
root@client:/etc/openvpn/cli x + v
root@client:/etc/openvpn# cd client
root@client:/etc/openvpn/client# mkdir keys
root@client:/etc/openvpn/client# ls
keys
root@client:/etc/openvpn/client#
```

Figura 48: Crear carpeta para organizar los archivos del cliente

3.19.1. Direcciones del Cliente

Para configurar las direcciones de nuestras llaves y certificados que futura mente estarán en este directorio (/etc/openvpn/client/keys) regresa dos directorios atrás para estar en /etc/openvpn/ donde se encuentra el archivo `client.conf` con los comandos:

```
cd ..  
nano client.conf
```

1. Dirección IP del Servidor

En el archivo de configuración del cliente debemos de especificar la dirección IP del servidor al cual se quiere conectar, así que se lo asignaremos en la línea:

```
;remote my-server-1 1194
```

Como se muestra en la Figura 49.

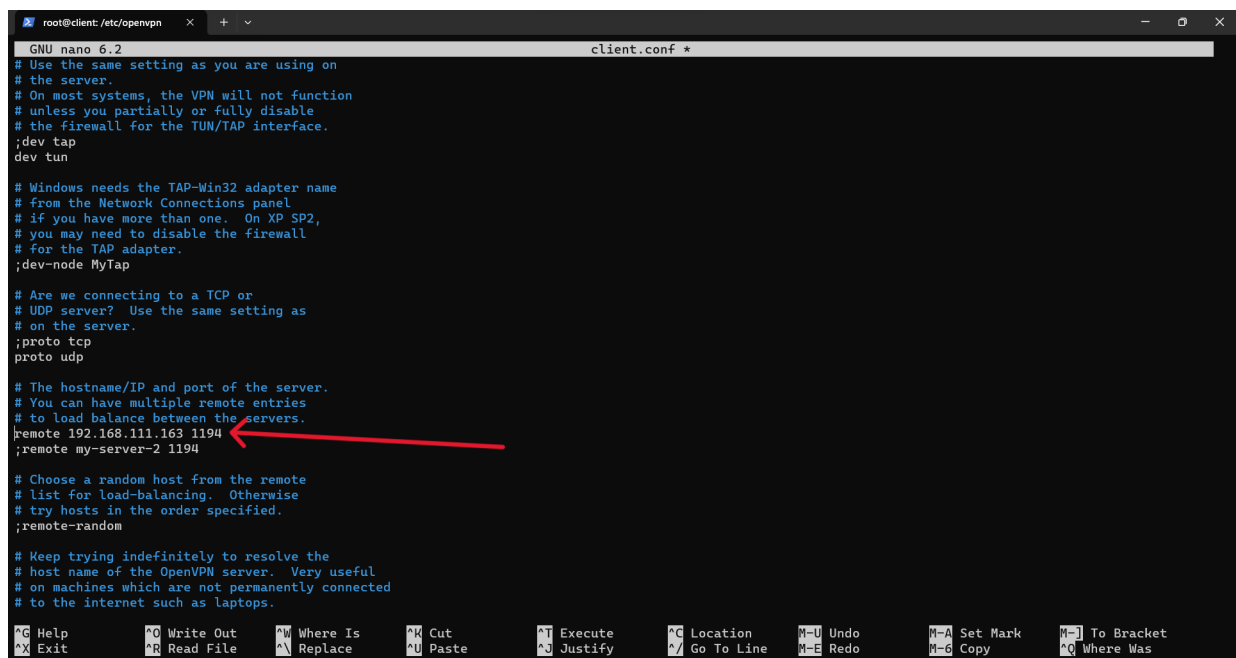
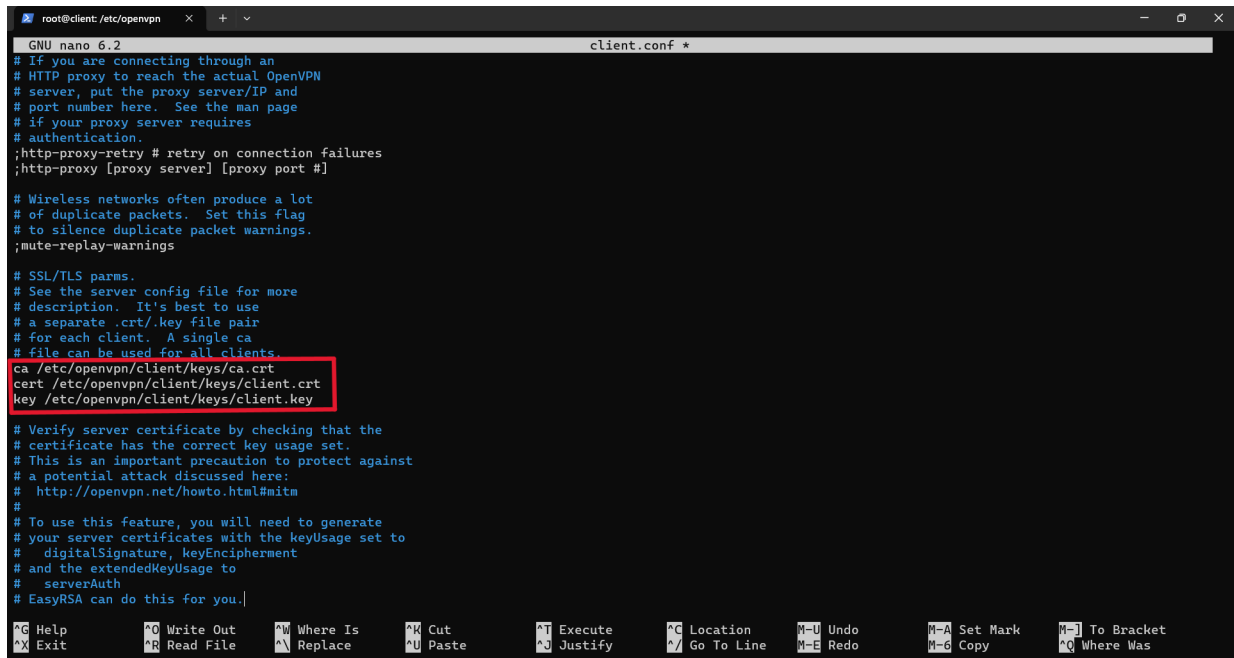


Figura 49: Configuración de la dirección IP del servidor en el archivo de configuración del cliente

2. Certificados y Llave del cliente

Asignaras las rutas de cada uno de esos certificados y llave, no están en ese directorio donde se encuentra el archivo en cuestión, pero recuerda que los certificados y llave privada estarán en la carpeta /etc/openvpn/client/keys/, así que asigna las direcciones de la siguiente manera (véase en la Figura 50):

```
ca /etc/openvpn/client/keys/ca.crt  
cert /etc/openvpn/client/keys/server.crt  
key /etc/openvpn/client/keys/server.key
```



```
GNU nano 6.2 client.conf *
# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here. See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets. Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
# description. It's best to use
# a separate .crt/.key file pair
# for each client. A single ca
# file can be used for all clients.
ca /etc/openvpn/client/keys/ca.crt
cert /etc/openvpn/client/keys/client.crt
key /etc/openvpn/client/keys/client.key

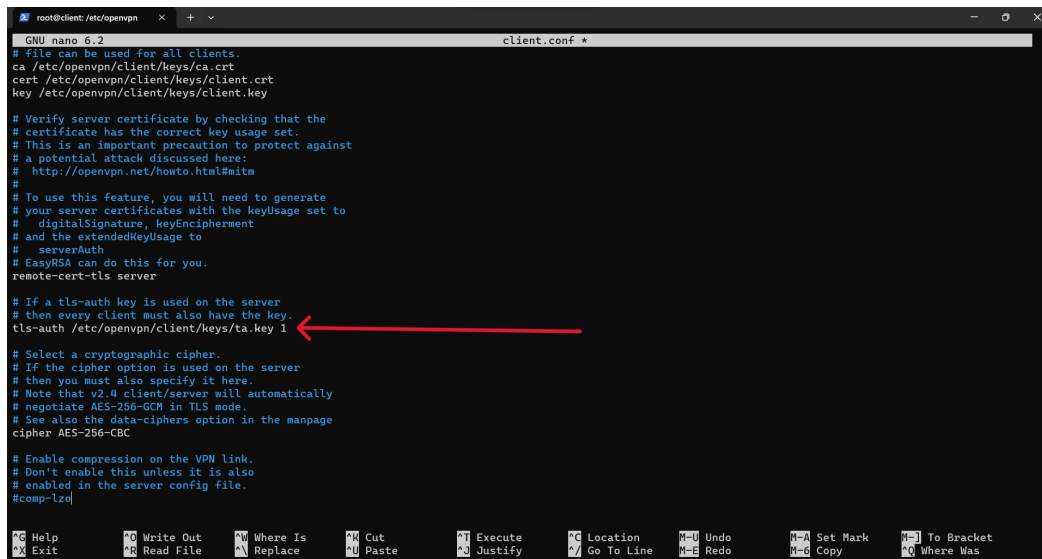
# Verify server certificate by checking that the
# certificate has the correct key usage set.
# This is an important precaution to protect against
# a potential attack discussed here:
# http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the keyUsage set to
# digitalSignature, keyEncipherment
# and the extendedKeyUsage to
# serverAuth
# EasyRSA can do this for you.

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark  M-J To Bracket
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo     M-C Copy      ^Q Where Was
```

Figura 50: Configuración de la dirección de certificados y llave privada del cliente

3. tls-auth

De igual manera buscaremos en el archivo la línea `tls-auth ta.key 0` y asignaremos su dirección, recordemos lo que venimos hablando todas estas se encontraran en el directorio `/etc/openvpn/client/keys/` así que le indicamos la ruta, como se muestra en la Figura 51:



```
GNU nano 6.2 client.conf *
# file can be used for all clients.
ca /etc/openvpn/client/keys/ca.crt
cert /etc/openvpn/client/keys/client.crt
key /etc/openvpn/client/keys/client.key

# Verify server certificate by checking that the
# certificate has the correct key usage set.
# This is an important precaution to protect against
# a potential attack discussed here:
# http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the keyUsage set to
# digitalSignature, keyEncipherment
# and the extendedKeyUsage to
# serverAuth
# EasyRSA can do this for you.
remote-cert-tls server

# If a tls-auth key is used on the server
# then every client must also have the key.
tls-auth /etc/openvpn/client/keys/ta.key 1

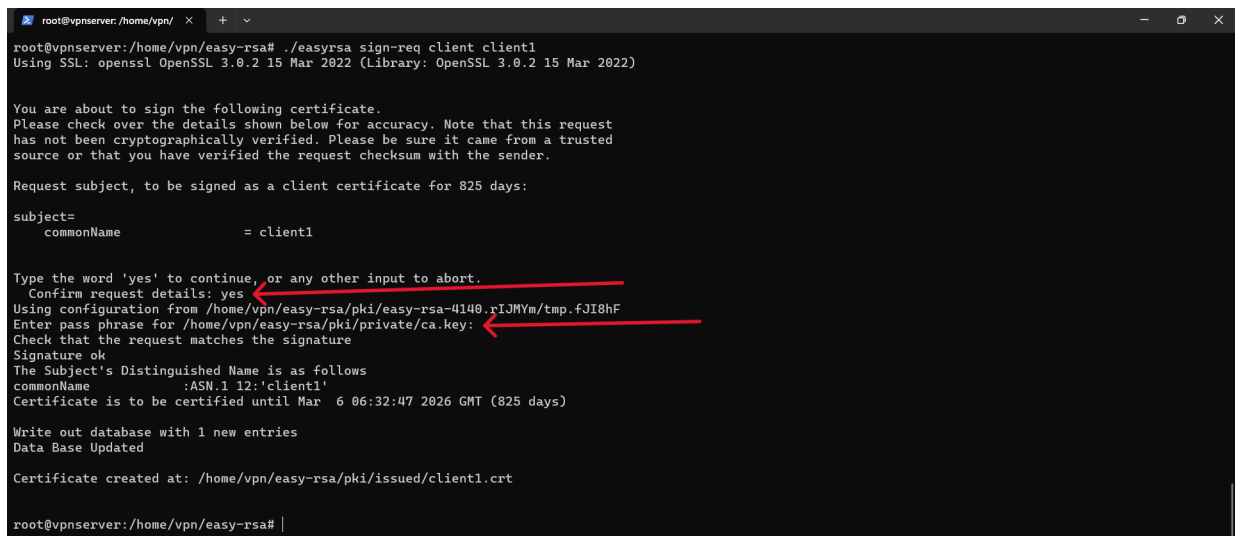
# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
# Note that v2.0 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the data-ciphers option in the manpage
cipher AES-256-CBC

# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
#comp-lzo

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark  M-J To Bracket
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo     M-C Copy      ^Q Where Was
```

Figura 51: Configuración de la dirección para la llave privada `ta.key` en el cliente

y ingrese la contraseña para confirmar, esta contraseña sera la cual se hablo en la sección 3.8 (Figura 53).



```
root@vpnserver: /home/vpn/
root@vpnserver:/home/vpn/easy-rsa# ./easyrsa sign-req client client1
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

You are about to sign the following certificate. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a client certificate for 825 days:

subject=
  commonName          = client1

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
Using configuration from /home/vpn/easy-rsa/pki/easy-rsa-4140.rIjMvM/tmp.fJI8hF
Enter pass phrase for /home/vpn/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'client1'
Certificate is to be certified until Mar  6 06:32:47 2026 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/vpn/easy-rsa/pki/issued/client1.crt
root@vpnserver:/home/vpn/easy-rsa#
```

Figura 53: Solicitud de firma del Certificado para el Cliente OpenVPN

3.21. Trasferencia de claves al cliente

Este servidor esta desarrollado y pensado para la seguridad entorno a la navegación segura, por esta razón no considero necesario hacer el típico archivo `.sh` (`make_config.sh`) el cual compila todas la llaves y certificados necesarios del cliente en un formato `.ovpn`, trasferiré cada uno de esos archivos a la carpeta `/etc/openvpn/client/keys` del cliente, desde el servidor a mi maquina cliente.

3.21.1. Organización de claves para el cliente en el servidor

Para que la explicación sea clara primero nos ubicaremos en el servidor, y organizaremos todos los archivos del cliente en el directorio `/etc/openvpn/client/`.

1. **ca (ca.crt)**

En la sección 3.8 creamos una entidad certificadora (CA) y nos generaron los certificados y claves necesarios para establecer la CA raíz, trasferiremos el archivo que nos valida la solicitud de certificado del servidor y firma del certificado, dicho archivo se alojo en:

`/home/vpn/easy-rsa/pki/ca.crt`

Nos aseguramos de estar en el directorio `/etc/openvpn/client/`, y copiamos el archivo con el comando (Figura 54):

```
cp /home/vpn/easy-rsa/pki/ca.crt .
```



```
root@vpnserver:/etc/openvpn
root@vpnserver:/home/vpn# cd /etc/openvpn/client/
root@vpnserver:/etc/openvpn/client# cp /home/vpn/easy-rsa/pki/ca.crt .
root@vpnserver:/etc/openvpn/client# ls
ca.crt
root@vpnserver:/etc/openvpn/client#
```

Figura 54: Trasferencia del certificado `ca.crt` en el servidor a la carpeta cliente

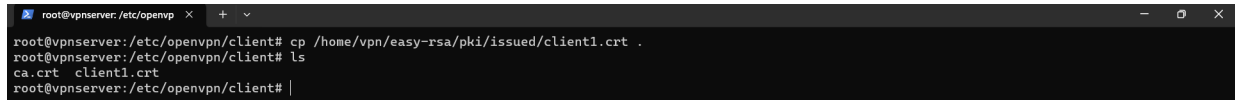
2. **crt (client1.crt)**

En la sección 3.20 donde validamos la solicitud de certificado del cliente y firmamos dicho certificado, el cual se alojo en:

Certificate created at: /home/vpn/easy-rsa/pki/issued/client1.crt

Necesitamos transferirla al directorio /etc/openvpn/client/, así que nos aseguramos de estar en el directorio en cuestión y la copiamos, con el comando(Figura 55):

```
cp /home/vpn/easy-rsa/pki/issued/client1.crt .
```



```
root@vpnserver: /etc/openvp X + v
root@vpnserver: /etc/openvpn/client# cp /home/vpn/easy-rsa/pki/issued/client1.crt .
root@vpnserver: /etc/openvpn/client# ls
ca.crt client1.crt
root@vpnserver: /etc/openvpn/client# |
```

Figura 55: Traslferencia del certificado para el cliente a la carpeta cliente

3. key (client.key)

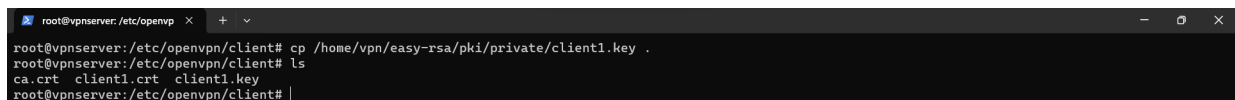
En la sección 3.20 generamos el requerimiento necesario para el cliente y si recuerdas nos dio las direcciones donde se generaron el requerimiento y la llave del cliente:

```
req: /home/vpn/easy-rsa/pki/reqs/client1.req
```

```
key: /home/vpn/easy-rsa/pki/private/client1.key
```

Nos aseguraremos de estar en la carpeta /etc/openvpn/client/ y copiaremos la llave del servidor a la carpeta en cuestión (Figura 56):

```
cp /home/vpn/easy-rsa/pki/private/client1.key .
```



```
root@vpnserver: /etc/openvp X + v
root@vpnserver: /etc/openvpn/client# cp /home/vpn/easy-rsa/pki/private/client1.key .
root@vpnserver: /etc/openvpn/client# ls
ca.crt client1.crt client1.key
root@vpnserver: /etc/openvpn/client# |
```

Figura 56: Traslferencia de la llave del cliente a la carpeta /etc/openvpn/client/ en el servidor

4. tls-auth (ta.key)

En la sección 3.12 generemos la clave de autenticación TLS en la carpeta /etc/openvpn/server/keys/

Nos aseguraremos de estar en la carpeta /etc/openvpn/client/ y copiaremos la llave del servidor a la carpeta en cuestión (Figura 57):

```
cp /etc/openvpn/server/keys/ta.key .
```



```
root@vpnserver: /etc/openvp X + v
root@vpnserver: /etc/openvpn/client# cp /etc/openvpn/server/keys/ta.key .
root@vpnserver: /etc/openvpn/client# ls
ca.crt client1.crt client1.key ta.key
root@vpnserver: /etc/openvpn/client# |
```

Figura 57: Traslferencia de la clave de autenticación TLS a la carpeta /etc/openvpn/client/ en el servidor

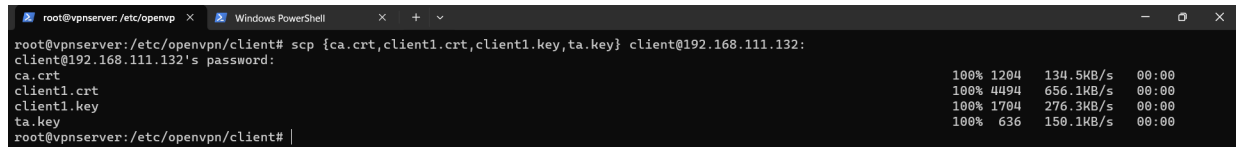
3.21.2. Traslferencia de claves mediante scp a la maquina cliente

Una vez organizadas nuestras claves para el cliente deberemos de contar con 4 archivos, en el directorio /etc/openvpn/client/:

```
ca.crt
client1.crt
client1.key
ta.key
```

Ubicados en el dicho directorio previamente mencionado transferiremos los archivos vía SSH con el comando (Figura 58):

```
scp {ca.crt,client1.crt,client1.key,ta.key} client@192.168.111.132:
```



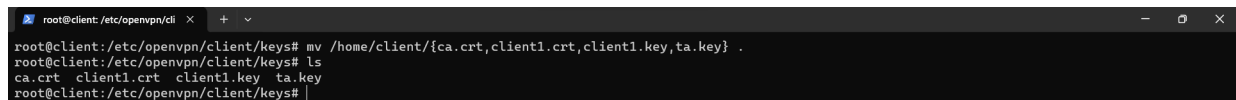
```
root@vpnserver:/etc/openvpn/client# scp {ca.crt,client1.crt,client1.key,ta.key} client@192.168.111.132:
client@192.168.111.132's password:
ca.crt                                100% 1204      134.5KB/s   00:00
client1.crt                          100% 4494      656.1KB/s   00:00
client1.key                          100% 1704      276.3KB/s   00:00
ta.key                                100% 636       150.1KB/s   00:00
root@vpnserver:/etc/openvpn/client#
```

Figura 58: Transferencia de las clave a la maquina cliente remotamente con el comando “scp”

El comando `scp` transfiere los archivos de manera segura al cliente, remotamente, identificado por la dirección IP `192.168.111.132`, utilizando el usuario `client`. Al nosotros poner : al final del comando copiará los archivos listados al directorio `home` del usuario `client` en la máquina con la dirección IP proporcionada.

Ahora debemos mover estas claves al directorio `/etc/openvpn/client/keys/`, nos ubicamos en dicho directorio y copiamos todos los archivos con el comando (Figura 59):

```
mv /home/client/{ca.crt,client1.crt,client1.key,ta.key} .
```



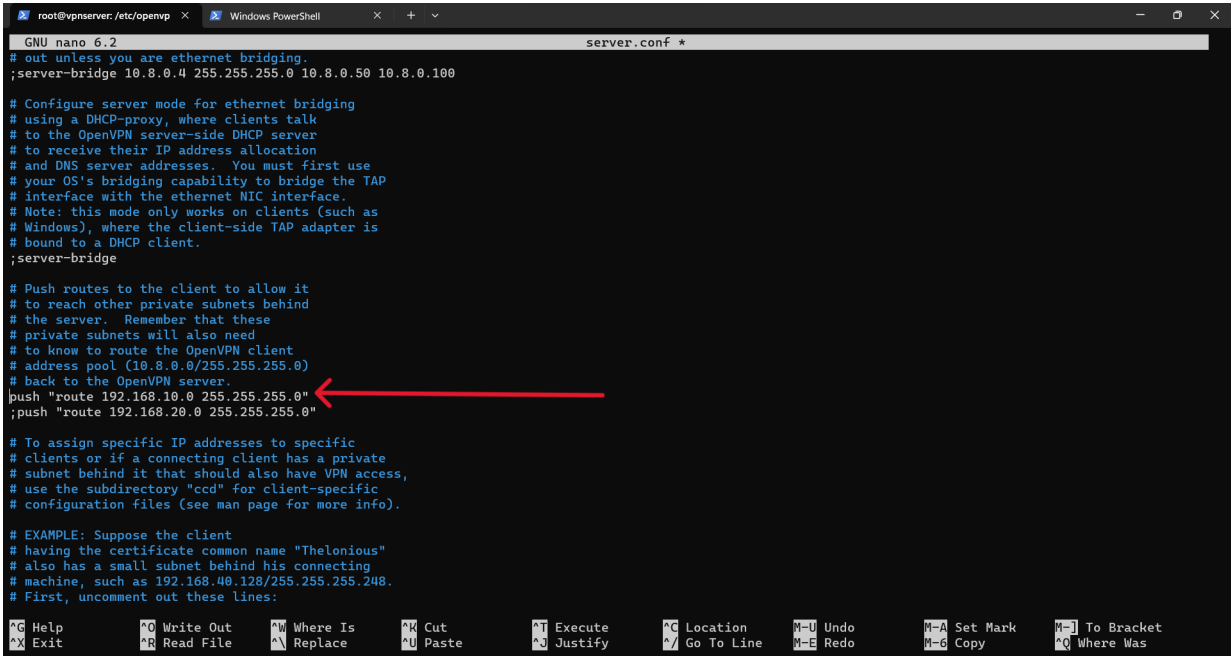
```
root@client:/etc/openvpn/client/keys# mv /home/client/{ca.crt,client1.crt,client1.key,ta.key} .
root@client:/etc/openvpn/client/keys# ls
ca.crt  client1.crt  client1.key  ta.key
root@client:/etc/openvpn/client/keys#
```

Figura 59: Movimiento las claves del cliente a su carpeta en la maquina cliente

3.22. Pruebas de funcionamiento

Previamente a iniciar el servicio por ambas partes tanto desde el servidor con el comando `openvpn --config server.conf` en el directorio `/etc/openvpn/`. Por parte del cliente con el comando `openvpn --config client.conf` en el directorio `/etc/openvpn/`. Nosotros podríamos el servicio funcionando pero el servicio los clientes no recibirán automáticamente la información de enrutamiento para esa red específica a través de la conexión VPN. Esto significa que, aunque estén conectados al servidor VPN, no tendrán una ruta predeterminada configurada para alcanzar la red. Por ende tenemos que entrar al archivo de configuración del servidor y asignar una ruta de enlace, en el directorio `/etc/openvpn/` en el servidor, abrimos el archivo de configuración del servidor y descomentamos la línea que se muestra en la Figura 60.

push ‘‘route 192.168.10.0 255.255.255.0’’



```
GNU nano 6.2 server.conf *
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Configure server mode for ethernet bridging
# using a DHCP-proxy, where clients talk
# to the OpenVPN server-side DHCP server
# to receive their IP address allocation
# and DNS server addresses. You must first use
# your OS's bridging capability to bridge the TAP
# interface with the ethernet NIC interface.
# Note: this mode only works on clients (such as
# Windows), where the client-side TAP adapter is
# bound to a DHCP client.
;server-bridge

# Push routes to the client to allow it
# to reach other private subnets behind
# the server. Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
push "route 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelionius"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
```

Figura 60: Asignación de ruta específica en la tabla de enrutamiento del servidor

`push ‘‘route 192.168.10.0 255.255.255.0’’`, es una directiva específica que se utiliza en el archivo de configuración del servidor OpenVPN. Esta directiva se emplea para indicar al cliente que agregue una ruta específica a su tabla de enrutamiento después de establecer la conexión VPN con el servidor.

En términos simples, esta línea indica al servidor que empuje la ruta de red `192.168.10.0/24` a los clientes cuando se conecten a la VPN. La notación `192.168.10.0` representa la dirección IP de la red y `255.255.255.0` es la máscara de subred que define qué parte de la dirección IP pertenece a la red y cuál a los hosts.

Cuando los clientes se conecten al servidor VPN y reciban esta instrucción, configurarán automáticamente una ruta en sus tablas de enrutamiento para dirigir el tráfico destinado a la red `192.168.10.0/24` a través de la conexión VPN. Esto permite que los dispositivos conectados a la VPN puedan comunicarse con los dispositivos en la red `192.168.10.0/24` a través de la conexión segura establecida por la VPN.

Es útil para permitir el acceso a recursos o dispositivos específicos que se encuentran en esa red remota, facilitando así la comunicación entre diferentes redes a través de la conexión VPN establecida. Para nosotros reconocer como clientes al servidor dentro de la red donde nos encontramos le asignare-

mos una IP dentro del rango del push que mencionamos anteriormente, por ejemplo la 192.168.10.5, utilizamos el comando (Figura 61):

```
root@vpnserver: /etc/openvpn# ip a add 192.168.10.5/24 dev enp0s3
root@vpnserver: /etc/openvpn# ip -c 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f8:9e:ae brd ff:ff:ff:ff:ff:ff
    inet 192.168.111.163/24 metric 100 brd 192.168.111.255 scope global dynamic enp0s3
        valid_lft 2807sec preferred_lft 2807sec
    inet 192.168.10.5/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a80:27ff:fe9e:9eae/64 scope link
        valid_lft forever preferred_lft forever
root@vpnserver: /etc/openvpn#
```

Figura 61: Asignación de una dirección IP al servidor dentro del rango de red que se utilizará para las conexiones VPN

Una vez realizada esta configuración podemos lanzar los dos servicios, por parte del servidor nos aseguraremos de estar ubicados en el directorio `/etc/openvpn/` y ejecutaremos el siguiente comando para lanzar el servicio de OpenVPN (Figura 62):

`openvpn --config server.conf`

```
root@vpnserver: /etc/openvpn# openvpn --config server.conf
2023-12-03 05:19:47 WARNING: --topology net30 support for server configs with IPv4 pools will be removed in a future release. Please migrate to --topology subnet as soon as possible.
2023-12-03 05:19:47 DEPRECATED OPTION: --cipher set to 'AES-256-CBC' but missing in --data-ciphers (AES-256-GCM:AES-128-GCM). Future OpenVPN version will ignore --cipher for cipher negotiations. Add 'AES-256-CBC' to --data-ciphers or change --cipher 'AES-256-CBC' to --data-ciphers-fallback 'AES-256-CBC' to silence this warning.
2023-12-03 05:19:47 OpenVPN 2.5.5 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PTINFO] [AEAD] built on Jul 14 2022
2023-12-03 05:19:47 library versions: OpenSSL 3.0.2 15 Mar 2022, LZO 2.10
2023-12-03 05:19:47 net_route_v4_best_gw query: dst 0.0.0.0
2023-12-03 05:19:47 net_route_v4_best_gw result: via 192.168.111.253 dev enp0s3
2023-12-03 05:19:47 Diffie-Hellman initialized with 2048 bit key
2023-12-03 05:19:47 Outgoing Control Channel Authentication: Using 160 bit message hash 'SHA1' for HMAC authentication
2023-12-03 05:19:47 Incoming Control Channel Authentication: Using 160 bit message hash 'SHA1' for HMAC authentication
2023-12-03 05:19:47 net_route_v4_best_gw query: dst 0.0.0.0
2023-12-03 05:19:47 net_route_v4_best_gw result: via 192.168.111.253 dev enp0s3
2023-12-03 05:19:47 ROUTE_GATEWAY 192.168.111.253/255.255.255.0 IFACE=enp0s3 HWADDR=08:00:27:f8:9e:ae
2023-12-03 05:19:47 TUN/TAP device tun0 opened
2023-12-03 05:19:47 net_iface_mtu_set: mtu 1500 for tun0
2023-12-03 05:19:47 net_iface_up: set tun0 up
2023-12-03 05:19:47 net_addr_pton_v4_add: 10.8.0.1 peer 10.8.0.2 dev tun0
2023-12-03 05:19:47 net_route_v4_add: 10.8.0.0/24 via 10.8.0.2 dev [NULL] table 0 metric -1
2023-12-03 05:19:47 Could not determine IPv4/IPv6 protocol. Using AF_INET
2023-12-03 05:19:47 Socket Buffers: R=[212992->212992] S=[212992->212992]
2023-12-03 05:19:47 UDPv4 link local (bound): [AF_INET]192.168.111.163:1194
2023-12-03 05:19:47 UDPv4 link remote: [AF_UNSPEC]
2023-12-03 05:19:47 MULTI: multi init called, r=256 v=256
2023-12-03 05:19:47 IFCONFIG POOL IPv4: base=10.8.0.4 size=62
2023-12-03 05:19:47 IFCONFIG POOL LIST
2023-12-03 05:19:47 Initialization Sequence Completed

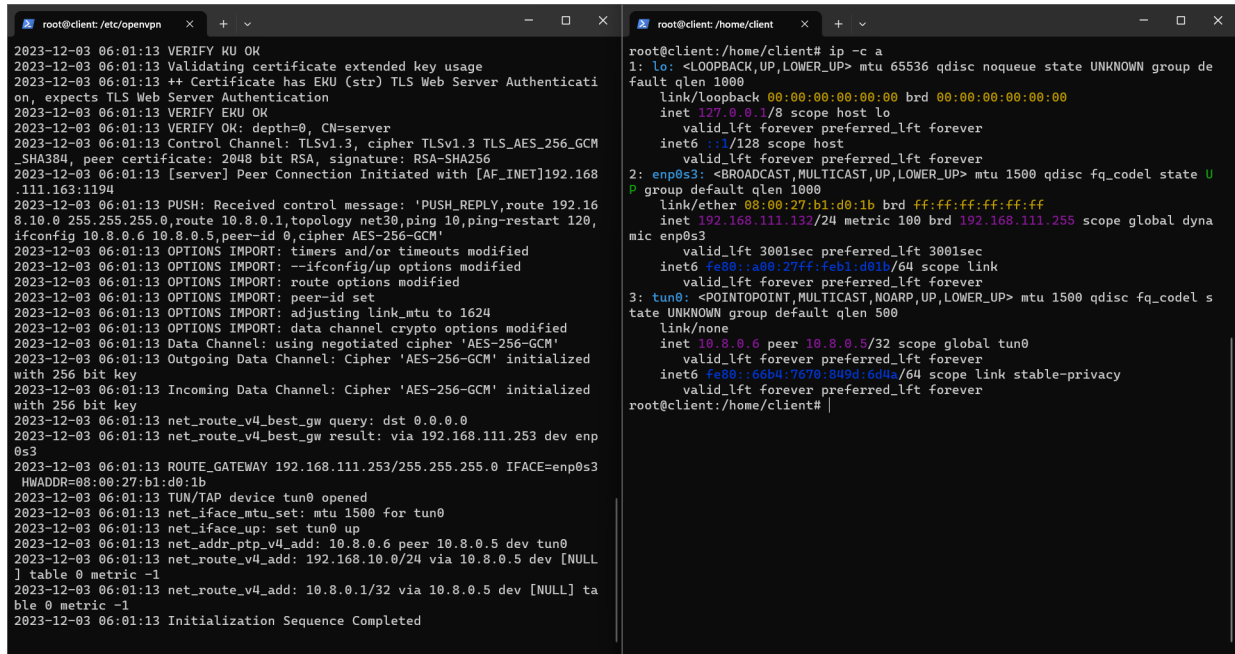
root@vpnserver: /home/vpn# ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f8:9e:ae brd ff:ff:ff:ff:ff:ff
    inet 192.168.111.163/24 metric 100 brd 192.168.111.255 scope global dynamic enp0s3
        valid_lft 2223sec preferred_lft 2223sec
    inet 192.168.10.5/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a80:27ff:fe9e:9eae/64 scope link
        valid_lft forever preferred_lft forever
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::655b:7efe:6025:a86a/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
root@vpnserver: /home/vpn#
```

Figura 62: Ejercicio del servicio OpenVPN en el servidor

Si hacemos un `ip -c a` en otra pestaña del servidor para ver la tabla de red podemos ver como ahora tenemos una tercera interfaz virtual la cual es de tipo `tun0`, y del otro lado en la última línea de ejecución un `2023-12-03 05:19:47 Initialization Sequence Completed` indicándonos que el servicio se ejecutó de manera correcta.

Ahora nos vamos a la parte del cliente, en su maquina, y ejecutamos el servicio de igual manera, nos ubicamos en el directorio `/etc/openvpn/` y ejecutamos el comando (Figura 63):

```
openvpn --config client.conf
```



```
root@client:/etc/openvpn x + - □ X
2023-12-03 06:01:13 VERIFY KU OK
2023-12-03 06:01:13 Validating certificate extended key usage
2023-12-03 06:01:13 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
2023-12-03 06:01:13 VERIFY ECU OK
2023-12-03 06:01:13 VERIFY OK: depth=0, CN=server
2023-12-03 06:01:13 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, peer certificate: 2048 bit RSA, signature: RSA-SHA256
2023-12-03 06:01:13 [server] Peer Connection Initiated with [AF_INET]192.168.111.163:1194
2023-12-03 06:01:13 PUSH: Received control message: 'PUSH_REPLY,route 192.168.10.0 255.255.255.0,route 10.8.0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5,peer-id 0,cipher AES-256-GCM'
2023-12-03 06:01:13 OPTIONS IMPORT: timers and/or timeouts modified
2023-12-03 06:01:13 OPTIONS IMPORT: --ifconfig/up options modified
2023-12-03 06:01:13 OPTIONS IMPORT: route options modified
2023-12-03 06:01:13 OPTIONS IMPORT: peer-id set
2023-12-03 06:01:13 OPTIONS IMPORT: adjusting link_mtu to 1624
2023-12-03 06:01:13 OPTIONS IMPORT: data channel crypto options modified
2023-12-03 06:01:13 Data Channel: using negotiated cipher 'AES-256-GCM'
2023-12-03 06:01:13 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2023-12-03 06:01:13 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2023-12-03 06:01:13 net_route_v4_best_gw query: dst 0.0.0.0
2023-12-03 06:01:13 net_route_v4_best_gw result: via 192.168.111.253 dev enp0s3
2023-12-03 06:01:13 ROUTE_GATEWAY 192.168.111.253/255.255.255.0 IFACE=enp0s3 HWADDR=08:00:27:b1:d0:1b
2023-12-03 06:01:13 TUN/TAP device tun0 opened
2023-12-03 06:01:13 net_ifconfig: mtu 1500 for tun0
2023-12-03 06:01:13 net_ifconfig: set tun0 up
2023-12-03 06:01:13 net_addr_pton_v4_add: 10.8.0.6 peer 10.8.0.5 dev tun0
2023-12-03 06:01:13 net_route_v4_add: 192.168.10.0/24 via 10.8.0.5 dev [NULL] table 0 metric -1
2023-12-03 06:01:13 net_route_v4_add: 10.8.0.1/32 via 10.8.0.5 dev [NULL] table 0 metric -1
2023-12-03 06:01:13 Initialization Sequence Completed

root@client:/home/client x + - □ X
root@client:/home/client# ip -c 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b1:d0:1b brd ff:ff:ff:ff:ff:ff
    inet 192.168.111.132/24 metric 100 brd 192.168.111.255 scope global dynamic enp0s3
        valid_lft 3001sec preferred_lft 3001sec
    inet6 fe80::a00:27ff:feb1:d01b/64 scope link
        valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.6 peer 10.8.0.5/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::66b4:7670:849d:6d4a/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
root@client:/home/client#
```

Figura 63: Ejecución del servicio OpenVPN en el cliente

Si hacemos un `ip -c a` en otra pestaña del servidor para ver la tabla de red podemos ver como ahora tenemos una tercera interfaz virtual la cual es de tipo `tun0`, y del otro lado en la ultima linea de ejecución un `2023-12-03 06:01:13 Initialization Sequence Completed` indicándonos que el servicio se ejecuto de manera correcta.

Se puede realizar un Ping al servidor mediante la IP `192.168.10.5` como se muestra en la Figura 64.



```
root@client:/home/client x + - □ X
root@client:/home/client# ping 192.168.10.5
PING 192.168.10.5 (192.168.10.5) 56(84) bytes of data.
64 bytes from 192.168.10.5: icmp_seq=1 ttl=64 time=1.99 ms
64 bytes from 192.168.10.5: icmp_seq=2 ttl=64 time=2.10 ms
64 bytes from 192.168.10.5: icmp_seq=3 ttl=64 time=2.08 ms
64 bytes from 192.168.10.5: icmp_seq=4 ttl=64 time=2.71 ms
64 bytes from 192.168.10.5: icmp_seq=5 ttl=64 time=2.41 ms
```

Figura 64: Ping del cliente al servidor

Con esto terminaríamos la configuracion del servidor VPN y el cliente.

4. Conclusiones

En conclusión, la configuración de un servidor VPN seguro y confiable es una habilidad valiosa para aquellos que buscan mejorar la seguridad en su red y proteger su información y comunicaciones en línea. A través de la práctica de Seguridad en la Red, se puede aprender sobre los conceptos básicos de VPN, la instalación y configuración de OpenVPN en un servidor Linux, la creación de certificados y claves de seguridad, y la configuración de clientes VPN. Además, la práctica es muy útil para aquellos que buscan mejorar sus habilidades en seguridad informática y para aquellos que buscan implementar una solución de VPN en su red. La implementación de una solución de VPN puede ser muy útil para aquellos que buscan proteger su información y comunicaciones en línea. La práctica también proporciona una excelente oportunidad para aprender sobre la importancia de la seguridad en la red y cómo se pueden utilizar las soluciones de VPN para mejorarla. En general, la configuración de un servidor VPN es una habilidad valiosa para aquellos que buscan mejorar la seguridad en su red y proteger su información y comunicaciones en línea. Recomiendo encarecidamente aprender sobre seguridad en la red y VPN a cualquier persona interesada en mejorar sus habilidades en seguridad informática.

5. Referencias

Referencias

- [1] T. Mocan. (2019, Febrero, 26) “¿Qué es un Servidor VPN y Cómo Funciona un Servidor VPN?” CactusVPN. Accedido el 21 de noviembre de 2023. [En línea]. Disponible: <https://www.cactusvpn.com/es/la-guia-para-principiantes-de-vpn/que-es-un-servidor-vpn/>
- [2] Y. Fernández. (2020, Junio, 1) “VirtualBox: qué es y cómo usarlo para crear una máquina virtual con Windows u otro sistema operativo”. Accedido el 21 de noviembre de 2023. [En línea]. Disponible: <https://www.xataka.com/basics/virtualbox-que-como-usarlo-para-crear-maquina-virtual-windows-u-otro-sistema-operativo>
- [3] (2023, Septiembre, 12)¿Qué es Ubuntu? Guía sobre este sistema operativo - Blog Piensa Solutions”. Blog Piensa Solutions. Accedido el 22 de noviembre de 2023. [En línea]. Disponible: <https://www.piensasolutions.com/blog/que-es-ubuntu-guia-sobre-este-sistema-operativo>
- [4] T. Mocan. (2019, Agosto, 30) “¿Qué Es OpenVPN y Cómo Funciona OpenVPN?” CactusVPN. Accedido el 22 de noviembre de 2023. [En línea]. Disponible: <https://www.cactusvpn.com/es/la-guia-para-principiantes-de-vpn/que-es-openvpn/>