

Ignacio Padrón Díaz

INSTALACION SERVIDOR OPENVPN

En una VPS Ubuntu server 22.04

1. Actualizar sistema e instalar paquete de openvpn

```
sudo apt update  
sudo apt upgrade -y
```

```
ubuntu@vps-813eb74d:~$ sudo apt update && apt upgrade -y
```

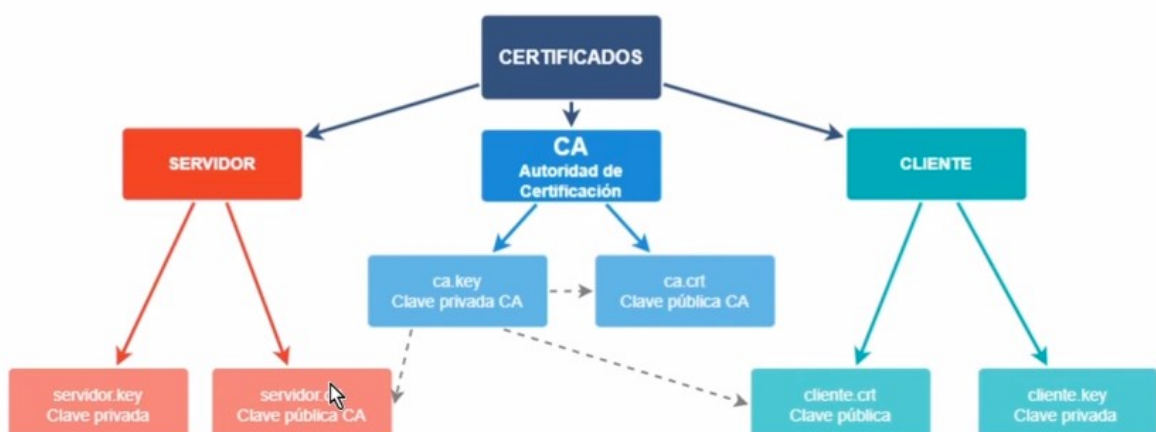
```
sudo apt install openvpn easy-rsa
```

```
ubuntu@vps-813eb74d:~$ sudo apt install openvpn easy-rsa
```

```
ubuntu@vps-813eb74d:~$ openvpn --version  
OpenVPN 2.5.9 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Sep 29 2023  
library versions: OpenSSL 3.0.2 15 Mar 2022, LZO 2.10  
Originally developed by James Yonan  
Copyright (C) 2002-2022 OpenVPN Inc <sales@openvpn.net>
```

Generación de claves y certificados

Inicializa el directorio PKI (Public Key Infrastructure) y crea un certificado de autoridad (CA).



1: copiar el directorio easy-rsa que se encuentra en *usr/share / al directorio etc/openvpn/*

Con esto conseguimos que cuando haya actualizaciones los certificados que haya generado o la configuración no se nos reemplace:

```
sudo cp -r /usr/share/easy-rsa /etc/openvpn/
```

```
ubuntu@vps-813eb74d:~$ sudo cp -r /usr/share/easy-rsa /etc/openvpn/
```

```
ubuntu@vps-813eb74d:/etc/openvpn$ ls
client  easy-rsa  server  update-resolv-conf
```

2: Entramos al directorio que hemos movido a `/etc/openvpn/easy-rsa/` y ejecutamos el script `./easyrsa init-pki` (la infraestructura de clave pública)

```
sudo ./easyrsa init-pki
```

```
ubuntu@vps-813eb74d:/etc/openvpn/easy-rsa$ sudo ./easyrsa init-pki
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/easy-rsa/pki
```

Ahora, nos aconseja crear una CA (Autoridad de Certificación) que lo haremos con el siguiente script en la misma carpeta `easy-rsa`:

```
sudo ./easyrsa build-ca nopass
```

(`nopass`) es para no tener que poner una contraseña al CA

```
ubuntu@vps-813eb74d:/etc/openvpn/easy-rsa$ sudo ./easyrsa build-ca nopass
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/etc/openvpn/easy-rsa/pki/ca.crt
```

SERVIDOR

Genera una solicitud de certificado y la clave privada para el servidor.

En este punto vamos a generar las claves para el servidor con el siguiente script:

```
sudo ./easyrsa gen-req server nopass
```

Donde pone `server` en el script anterior, es el nombre que quieras ponerle.

```
Keypair and certificate request complete
req: /etc/openvpn/easy-rsa/pki/reqs/serv
key: /etc/openvpn/easy-rsa/pki/private/s
```

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

Diffie-Hellman (DH) y TLS (Transport Layer Security)

```
ubuntu@vps-813eb74d:/etc/openvpn/easy-rsa$ sudo ./easyrsa gen-dh
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)
Generating DH parameters, 2048 bit long safe prime
.....+.....+.....+.....+
```

```
DH parameters of size 2048 created at /etc/openvpn/easy-rsa/pki/dh.pem
```

Luego el tls-auth (ta.key):

1: Accedemos a la carpeta /etc/openvpn/server para tener ordenadas las claves generadas en una carpeta.

2: Ejecutamos el siguiente script:

```
sudo openvpn --genkey secret ta.key
```

```
ubuntu@vps-813eb74d:/etc/openvpn/server$ sudo openvpn --genkey secret ta.key
ubuntu@vps-813eb74d:/etc/openvpn/server$ ls
ca.crt  server-cerrojo.crt  server-cerrojo.key  ta.key
```

(Opcional) Lo siguiente, será ordenar los certificados generados ya que cada uno se ha almacenado en una carpeta diferente. Por lo tanto, vamos a meter todo en la carpeta /etc/openvpn/server/

```
sudo cp /etc/openvpn/easy-rsa/pki/issued/server-cerrojo.crt /etc/openvpn/server/
```

```
sudo cp /etc/openvpn/easy-rsa/pki/ca.crt /etc/openvpn/server/
```

```
sudo cp /etc/openvpn/easy-rsa/pki/private/server-cerrojo.key /etc/openvpn/server/
```

CLIENTE

Configurar el cliente OpenVPN.

Primero, crea el certificado y la clave privada del cliente de la misma manera que hicimos con el servidor:

1: Nos centraremos en el directorio client para organizar estas claves

```
cd /etc/openvpn/client/
```

2: Modificamos los permisos para quitarles los privilegios de usuario y del grupo al directorio /client/

```
sudo chmod -R 700 /etc/openvpn/client
```

Luego, vamos al directorio easy-rsa para generar el requerimiento o certificado y la clave privada para el cliente ejecutando el siguiente script

```
sudo ./easyrsa gen-req client1 nopass
```

Donde pone client1 podemos poner el nombre que queramos

En este momento, hay que firmar el requerimiento para obtener la clave pública con el siguiente script:

```
sudo ./easyrsa sign-req client cliente1
```

```
Certificate created at: /etc/openvpn/easy-rsa/
```

Ahora vamos a organizar todas las claves en el directorio /client/:

```
sudo cp /etc/openvpn/easy-rsa/pki/issued/client1.crt /etc/openvpn/client/
```

```
sudo cp /etc/openvpn/easy-rsa/pki/private/client1.key /etc/openvpn/client/
```

```
sudo cp /etc/openvpn/easy-rsa/pki/ca.crt /etc/openvpn/client/
```

```
sudo cp /etc/openvpn/server/ta.key /etc/openvpn/client/
```

```
root@vps-813eb74d:/etc/openvpn/client# ls -la
total 36
-rwx----- 1 root root 1204 May 28 12:04 client1.key
-rwx----- 1 root root 8293 May 28 12:04 client1.pki
-rwx----- 1 root root 4494 May 28 12:04 client1.pki
-rwx----- 1 root root 1704 May 28 12:04 client1.pki
```

Configura el servidor OpenVPN

Crear un nuevo archivo de configuración, luego hay que configurar el archivo para la VPN, por tanto entro en el archivo con un nano y agrego las líneas como viene a continuación:

```
sudo nano /etc/openvpn/server/server.conf
```

Agrega el siguiente contenido al archivo:

```
port 1194
proto udp
dev tun
ca /etc/openvpn/server/ca.crt
```

```
cert /etc/openvpn/server/server.crt
key /etc/openvpn/server/server.key
dh /etc/openvpn/server/dh.pem
auth SHA256
tls-auth /etc/openvpn/server/ta.key 0
key-direction 0
cipher AES-256-CBC
user nobody
group nogroup
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"
keepalive 10 120
comp-lzo
persist-key
persist-tun
status openvpn-status.log
verb 3
explicit-exit-notify 1
client-to-client
```

A continuación, se explica qué es o para qué sirve cada línea:

#puerto a utilizar: tcp o udp. Por defecto es 1194
port 1194

#protocolo a utilizar: tcp o udp
proto udp
#tipo de tunel: dev tun (enrutamiento IP) o dev tap (puente Ethernet)
dev tun

#Llas claves y certificados que hemos utilizado
ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/server.crt
key /etc/openvpn/server/server.key

especifica el algoritmo de autenticación HMAC (Hash-based Message Authentication Code) algoritmo de hash criptográfico que produce un hash de 256 bits
auth SHA256

#Directiva de seguridad Diffie-Hellman (DH) y TLS-AUTH
dh /etc/openvpn/server/dh.pem
tls-auth /etc/openvpn/server/ta.key 0

#directorio de clave por defecto 0
key-direction 0

#tipo de cifrado CBC
cipher AES-256-CBC

#Sin permiso de usuario y grupo para mayor seguridad
user nobody
group nogroup

#dirección de subred de la VPN (ip_servidor=10.8.0.1)solo para dev tun
server 10.8.0.0 255.255.255.0

#configuramos para que los clientes tengan la misma IP siempre
ifconfig-pool-persist ipp.txt

#Esta directiva configurará todos los clientes para redirigir su puerta y los clientes puedan acceder a internet a través de la VPN
push "redirect-gateway def1 bypass-dhcp"

#Para pasar a los clientes unos servidores DNS para la resolución de nombres
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"

#keepalive para saber si el tunel se ha caido, hace ping cada 10 segundos y si no hay #respuesta a los 120 segundo es que estaría caido
keepalive 10 120

#el tipo de compresor
comp-lzo
#la clave y el tunel sean persistentes
persist-key

persist-tun

#En qué fichero queremos guardar las conexiones actuales
status openvpn-status.log

#El nivel de verbosidad o de información que queremos que se registre en los
logs
verb 3

#Notificación de reinicio
explicit-exit-notify 1

#Para que los clientes se puedan conectar entre ellos
client-to-client

Crear y configurar el archivo del cliente client.conf

Hay que crear y configurar el archivo del cliente de acuerdo a las
configuraciones que la del servidor, las claves y las certificaciones.

Las configuraciones quedarían así:

client
dev tun
proto udp
remote 149.56.45.220 1194
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
auth SHA256
auth-nocache
cipher AES-256-CBC
tls-client
tls-auth ta.key 1
key-direction 1
comp-lzo
verb 3

Justo después de las configuraciones, hay que ir haciendo un cat de cada clave y certificación del directorio /client/

```
root@vps-813eb74d:/etc/openssl/client# ls
ca.crt client.conf client1.crt client1.key dh.pem ta.key
```

Luego, ir copiando el contenido de los archivos: ca.crt, client1.crt, client1.key y ta.key en el archivo client.conf. Debería tener una estructura como la siguiente:

<ca>

-----BEGIN CERTIFICATE-----

```
MIIDSzCCAjOgAwIBAgIUTrsbWdOWEdjvhyAG7meltrM/jXQwDQYJKoZIhvcNAQELBQAw
FjEUMBIGA1UEAwwLRWFzeS1SU0EgQQEwHhcNMjQwNTI4MTQxODE2WhcNMzQwNTI
2MTQxODE2WjAwwMRQwEgYDVQQDDAtFYXN5LVJTSBDQTCCASlwDQYJKoZIhvcNAQE
BBQADggEPADCCAQoCggEBBAK4UtmVuZXUvleHmkdiCxHliAdK97tHIH3R5BiR7KAp5Ylt7
nLTppK0z0EupAZD0gQQZJC+I970WYgqADoMUf1j8zHJSZHRkaFKH8280Banyl0jkl8aaVY
TcQ5+4lRwYrHA3q6FXSE4clMHgsqStWooiPmZ7Ok8enR72vjC8akrzl7N6HCe3QBCXkDAi
ulifv/4sKRpCik1QA1bHeKJYRk2L8rMcZJ9RdaZq2MIYkst2jfKgmDqsE1pUf7PfA6agtOHO
ZDvpGo2gig5I5n6SZ0LI156YlxEoASTCcQcTTFI3dpdzy/viVLjwuYzR1WqTtjiw76yUVHkzP1
sQ1MTJECaAwEAAaOBkDCBjTAdBgNVHQ4EFgQUOMUFqoLpohhBINGw+n97cudIAWMw
UQYDVR0jBEowSIAUOMUFqoLpohhBINGw+n97cudIAWOHqQYMBYxFDASBgNVBAM
MC0Vhc3ktUINBIENBghROyttZ05YR2O+HIAbuZ6W2sz+NdDAMBgNVHRMEBTADAQH/
MAsgA1UdDwQEAwIBBjANBgkqhkiG9w0BAQsFAAOCAQEANFT75D3TEWzfwM69oqtze
O0dYe2FAtd81ccFgUx9mdpilErwYaaVvPUX8CNiS0sdPjUvsXSps6mjjl94W55QxtjLiqC5LP
YrHnY207Tv/fkRqzFRqp9DLm5h5Gf9Cn3EQzzeUYdeBtWD7CKc8WVBzi66JQfyL4D4uPo
N95DDDMGnZi5YIFnwt5vGmxrzjyEJ4rVho+t2Pdyk2KRr41vS3fIYvK73udYDinmNqh6h
WZNO5cztVlbdGACuOnlelBr6tRX5aMCx4QQpj1niBhi9CYK9XVVWRUewX3kvZ/M/bWa
wjisYJ7RAKIXJsu3Zy8uCnUpZBqsYqAFGI1JZI2uRA==
```

-----END CERTIFICATE-----

</ca>

<cert>

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

6e:cd:8b:3d:52:94:aa:bd:9a:7d:12:6b:b3:2f:b5:b8

Signature Algorithm: sha256WithRSAEncryption

Issuer: CN=Easy-RSA CA

Validity

Not Before: May 28 14:38:27 2024 GMT

Not After : Aug 31 14:38:27 2026 GMT

Subject: CN=client1

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

Modulus:

00:b5:83:28:f3:4a:34:6a:37:e1:ed:e1:d6:21:bc:
a3:95:ef:56:dc:cb:9f:ed:89:41:9d:71:b1:ff:12:
9e:17:a1:17:de:d5:9b:63:77:83:40:c1:a0:93:7a:
48:42:1c:ba:86:9c:92:87:1c:16:9d:33:ac:ec:7b:
ee:8d:97:ee:db:10:66:ec:0b:b6:0c:8b:8f:1f:5a:
b2:3e:11:c4:d4:88:ad:b0:8d:04:f9:03:b2:48:05:
4e:dc:6c:47:c1:45:7d:fa:5f:0b:7a:8f:c8:05:22:
ac:69:fb:e7:f7:8a:ef:51:f4:c3:6a:6f:f7:dc:b9:
9d:ca:b7:c8:29:7f:11:23:91:2d:85:ff:aa:0d:e1:
47:57:3a:1d:20:bd:8b:4e:82:00:22:2b:8b:8f:73:
b7:c0:d7:22:72:39:1e:52:fe:54:1e:00:0d:c8:db:
06:3d:8e:87:5c:30:6c:4f:24:14:89:1a:70:25:80:
3c:2a:44:f1:38:2e:ea:48:b0:74:2c:ff:5a:0b:c6:
27:fe:1a:a2:26:aa:1c:da:61:3f:db:35:46:f3:ef:
6d:a7:82:32:38:cc:ec:6c:20:a8:1e:b7:a9:d2:56:
42:3b:49:2f:10:bc:f1:a5:60:ec:1e:0f:4e:ed:31:
48:ff:60:97:4b:94:df:0f:4c:a6:97:12:3d:46:b5:
c7:93

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

X509v3 Subject Key Identifier:

F8:DF:7C:69:38:CE:32:61:DB:DA:D0:61:B7:53:FD:C4:0C:86:9F:9A

X509v3 Authority Key Identifier:

keyid:38:C5:05:AA:82:E9:A2:18:41:20:D1:B0:FA:7F:7B:72:E7:65:01:63

DirName:/CN=Easy-RSA CA

serial:4E:CA:DB:59:D3:96:11:D8:EF:87:20:06:EE:67:A5:B6:B3:3F:8D:74

X509v3 Extended Key Usage:

TLS Web Client Authentication

X509v3 Key Usage:

Digital Signature

Signature Algorithm: sha256WithRSAEncryption

Signature Value:

9b:00:2b:a8:18:1e:d7:15:15:0c:37:82:3c:70:c9:8c:5f:9d:
3a:b0:67:c9:20:4a:1f:0d:50:aa:ea:3a:27:6a:a6:c2:ea:fc:
56:43:bc:a8:c3:57:fb:00:e7:b8:1d:7c:9e:4e:7a:2b:b8:5d:
7e:86:91:f5:18:ab:c9:35:2a:3e:7c:69:5f:82:24:68:17:13:
76:d1:d9:73:ea:28:23:c4:82:65:fb:50:56:d8:9b:1b:38:4c:
09:d7:be:7d:1c:3b:75:9c:dc:b5:f3:f3:73:9f:ee:49:d3:89:
3c:32:35:8a:64:d1:4a:6a:bb:8a:e0:f1:00:66:f0:21:39:c7:
58:61:62:cc:54:59:04:e8:b6:5e:04:b4:c2:d5:d5:a2:a1:dc:
af:0d:f5:f3:a6:50:75:7f:34:ff:9a:50:18:5b:6e:55:f0:41:
23:79:78:02:59:c7:ea:72:22:96:d4:1a:22:81:d6:9e:51:f8:
d7:b3:b4:b5:99:b2:93:e5:51:7b:11:15:3c:e7:35:39:ed:a3:

bb:d5:0c:f2:fd:14:83:0d:9a:a0:c0:09:4a:37:32:fb:3c:82:
c3:f9:d9:78:52:a0:d0:7c:fd:5b:51:c2:db:02:3c:99:94:ab:
9b:c1:0f:96:b1:57:6f:8b:89:86:58:98:cb:cc:9b:4f:05:e5:
0c:62:aa:d9

-----BEGIN CERTIFICATE-----

MIIDVTCCAj2gAwIBAgIQbs2LPVKUqr2afRJrsy+1uDANBgkqhkiG9w0BAQsFADAWMRQw
EgYDVQQDDAtFYXN5LVJTSBDQTAEFw0yNDA1MjgxNDM4MjdaFw0yNjA4MzExNDM4
MjdaMBIxEDAOBgNVBAMMB2NsaWVudDEwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwA
wggEKAoIBAQC1gyjzSjRqN+Ht4dYhvKOV71bcy5/tiUGdcbH/Ep4XoRfe1Ztjd4NAwaCTek
hCHLqGnJKHHBadM6zse+6NI+7bEGbsC7YMi48fWrl+EcTUiK2wjQT5A7JIBU7cbEfBRX36
Xwt6j8gFlqxp++f3iu9R9Mnqb/fcuZ3Kt8gpfxEjKS2F/6oN4UdXOh0gvYtOggAiK4uPc7fA1
yJyOR5S/IQeAA3I2wY9jodcMGxPJBSJGnAlgDwqRPE4LuplsHQs/1oLxif+GqImqhzaYT/bN
Ubz722ngjl4zOxslKget6nSVkl7SS8QvPGIYOweD07tMUj/YJdLIN8PTKaXEj1GtceTAgMBA
AGjgalwgZ8wCQYDVR0TBAlwADAdBgNVHQ4EFgQU+N98aTjOMmHb2tBht1P9xAyGn5o
wUQYDVR0jBEowSIAUOMUFqoLpohhBINGw+n97cudlAWOhGqQYMBYxFDASBgNVBA
MMCOVhc3ktUINBIENBghROYttZ05YR2O+HIAbuZ6W2sz+NdDATBgNVHSUEDDAKBggrB
gEFBQCDAjALBgNVHQ8EBAMCB4AwDQYJKoZIhvcNAQELBQADggEBAJsAK6gYHtcVFQw
3gjxwyYxfnTqwZ8kgSh8NUKrqOidqpsLq/FZDvKjDV/sA57gdfJ5Oeiu4XX6GkfUYq8k1Kj58
aV+CJGgXE3bR2XPqKCEgmX7UFbYmxs4TAnXvn0cO3Wc3LXz83Of7knTiTwyNYpk0Upq
u4rg8QBm8CE5x1hhYsxUWQTotl4EtMLV1aKh3K8N9fOmUHV/NP+aUBhbbIXwQSN5eA
JZx+pylpbUGiKB1p5R+NeztLWZspPIUXsRFTznNTnto7vVDPL9FIMNmQDACUo3Mvs8gsP
52XhSoNB8/VtRwtsCPJmUq5vBD5axV2+LiZYmMvMm08F5Qxiqtk=

-----END CERTIFICATE-----

</cert>

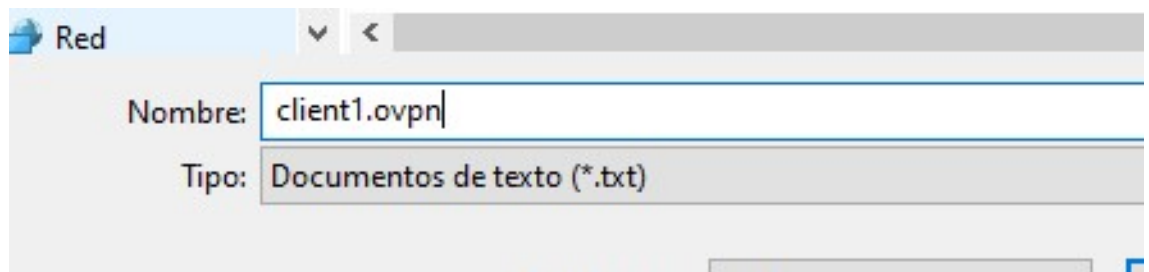
<key>

-----BEGIN PRIVATE KEY-----

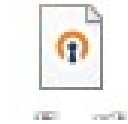
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCkwwggSjAgEAAoIBAQC1gyjzSjRqN+Ht4dYhvK
OV71bcy5/tiUGdcbH/Ep4XoRfe1Ztjd4NAwaCTekhCHLqGnJKHHBadM6zse+6NI+7bEGbs
C7YMi48fWrl+EcTUiK2wjQT5A7JIBU7cbEfBRX36Xwt6j8gFlqxp++f3iu9R9Mnqb/fcuZ3Kt
8gpfxEjKS2F/6oN4UdXOh0gvYtOggAiK4uPc7fA1yJyOR5S/IQeAA3I2wY9jodcMGxPJBSJG
nAlgDwqRPE4LuplsHQs/1oLxif+GqImqhzaYT/bNUbz722ngjl4zOxslKget6nSVkl7SS8QvP
GIYOweD07tMUj/YJdLIN8PTKaXEj1GtceTAgMBAAECggEACIsu9yCTkpssZ9i6BoFzGKEd
CzuTWZoidpDYDFzWSliaXRXtbB3PSHgE3duCOHRTDf44OMxk7S0IW0MwLcMelzYFPd4J
wVqqmjA4XfbC/LjbV+cT4K2oZOeYqgchVYACzwhQWOcxqEbL664lhS0FYEWS0XgWBAXa
f+8kp3bvXcqBJYtM0OeqKljmCORvWTOIQ0L2OtwEfMGz72zy91oNuh+sVIFzDnzRDh9bG
kvmfHCDO8wcWltNMgs0F84w2NeetGLSVsl4W9NmRdgFO3XD3W4JM1j6IAjR6vNZuU6
aYdDzkXEKzpzcknnmtvlZuqx9/F3ZrJBeeBkZwhR7+AQKBgQDgWtdn2uayaUBajE/Qrxdg
NNdmJSUPsR5seOFJBfKpftLSECZtUcma+aOGc+0HZWe/z6crLnyPwqKfE0/ai2608LeMEr
QZVE1pfTPzWM0d8iG90zQhihaRXKApUz1gPrIkYNUPMysx0hMGEfLdP2tI1SVrVnKcJUC
tlqKxgVb+MRQKBgQDPHVT9bqOFPT8ke7gOLx/2r6RTClY5od+r1J1GRG+AAAN79g3A7f8H
aGRqfpriahrAg/TMAbfcPJr9yBID1v5Vp9xrtbLPVVYlyfhOVx7P3QhL+FpB1CSEXnbG3Ee4
hri5VZei2cSk+5ShAOtl9ZErctjV7lx2kUe64V/gnZqU99wKBgDWpJJJo/f5XNEA0DScRqdql6
NsSiVG+gFNbei7YEFk/7za1DCvbgwfhK56N/aRmObFj/kzSupl78F9x5AiWju0SliSLmK3yPu
27OQ67r5u4/R8E9BQNjiZlbDpWIXjypfMTggI3r94jq88o3djTQVrTODb6u7rkBBNA0z3f5
letAoGBAL4TDDmdfQI9aCJiTFLwGqR5lXr2kQxnq271q/Sdv7o4km0UYD6qi8xNd6qHBio

```
Ca8eKt3CwesDO6eHqPAicb1IVTigQt6WUX5SscermoJnNkMKRELnze/vb/YzxVgxkylaaYx
to82t3te7anfyP+X4bnJ4kq6H5s5VJU97mzCXAoGAFFysek+FLsMGqqsXMTV4EXgHpzW
X806LSDHwLIPBa0qvNERoKAnD8sMK2CJIKyHzQLgUniNqZdS9q4HgoD6COYcJNndfhCO
qMYA4C58txnY5PytW38QqVqQs1yRIZI6Sp1de8/FBioD0Sk4fS+x5S2KzkWs9uJDy253W3
OxLoU=
-----END PRIVATE KEY-----
</key>
<tls-auth>
-----BEGIN OpenVPN Static key V1-----
64387df44100e24b13d6f866e7f70363
c54b7cc1e475b9195c3d7bce69380d7c
652cf565f7654961a4452d15b392d4ec
69e5365d0047711a406a5027db0b063d
8f5171eda41fc4e11fade3da03cd7a7b
4010f040fcd216ebf2b614514c2ef7cf
7f2263e4423c2701cc3436734d5c0689
cb8d3a2163b7035501acf6c2a90ff598
967a5c5d0385bc25b7c149d4e871bc14
59f465cbdddeaedaf531bc60fc77c82d
6b413b12dbdc1632971a1e32b3f563d5
429c3814cf21c683735bab6545f00db
1b853abe2dc4ec91ebbf0371dfd03ef6
ff0b617417a3bcbff28b90f98b6c30c3
e59e88fdbaa763b16146459c1a51ed8c
c5245125ab422dcb4a4f4cf436f65db6
-----END OpenVPN Static key V1-----
</tls-auth>
```

Para sacar el archivo al pc del cliente local, pegar en un bloc de notas el contenido del archivo client1.conf y cambiarlo a .ovpn



También será necesario además de sacar el archivo al pc del cliente local, sacar el archivo ta.key, lo que habrá que cambiarle los permisos para hacerlo.



Habilitar el reenvío de paquetes IP entre interfaces

1:

```
sudo nano /etc/sysctl.conf
```

2: Descomenta o agrega la siguiente línea: (quitar # de la línea)

```
net.ipv4.ip_forward=1
```

Para permitir el reenvío en la sesión actual ponemos si no te deja hacerlo desde el root:

```
sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

Añadir reglas al cortafuego para permitir puertos y protocolos para la conexión a internet:

Aviso → ir poniendo cada línea de una en una, copiarlo todo a la vez puede dar error.

A continuación, establece las políticas predeterminadas para INPUT, OUTPUT y FORWARD:

```
sudo iptables -P INPUT DROP  
sudo iptables -P OUTPUT ACCEPT  
sudo iptables -P FORWARD DROP
```

Permite el tráfico entrante para conexiones establecidas y relacionadas:

```
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Si necesitas permitir tráfico específico, como SSH (puerto 22), HTTP (puerto 80) y HTTPS (puerto 443), ejecuta:

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

Permitir tráfico entrante y saliente en la interfaz de bucle invertido (loopback):

```
sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A OUTPUT -o lo -j ACCEPT
```

Permitir conexiones SSH entrantes (cambie 22 por el puerto SSH personalizado si es necesario):

```
sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Permitir tráfico OpenVPN entrante (suponiendo que estés utilizando el puerto UDP 1194):

```
sudo iptables -A INPUT -p udp --dport 1194 -j ACCEPT
```

Permitir tráfico HTTP y HTTPS entrante (si estás ejecutando un servidor web):

```
sudo iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

```
sudo iptables -A INPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
```

Permitir tráfico ICMP entrante (ping):

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

Permitir tráfico relacionado y establecido:

```
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Permitir el reenvío de paquetes para el tráfico VPN:

```
sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -s 10.8.0.0/24 -j ACCEPT
```

Configurar NAT para la subred VPN, (en vez de poner eth0 poner la interfaz de la red pública mirándolo con el comando ip a, en mi caso (ens3):

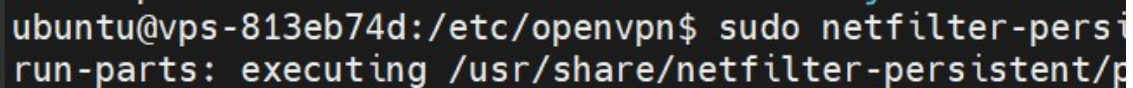
```
sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o ens3 -j MASQUERADE
```

Después de configurar las reglas, asegúrate de guardarlas para que persistan después de reiniciar el sistema. En sistemas basados en Debian y Ubuntu, puedes instalar el paquete iptables-persistent para lograrlo:

```
sudo apt-get install iptables-persistent
```

Para volver a guardar las reglas de forma persistente:

```
sudo netfilter-persistent save
```



```
ubuntu@vps-813eb74d:/etc/openvpn$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/persistent.rules
```

Para ver las reglas del cortafuego:

```
sudo iptables -L -nv
```

```

ubuntu@vps-813eb74d:/etc/openvpn$ sudo iptables -L -nv
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source    destination
    0     0 ACCEPT    udp  --  eth0    *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    all  --  tun0    *       0.0.0.0/0  0.0.0.0/0
  152 14380 ACCEPT    tcp  --  *       *       0.0.0.0/0  0.0.0.0/0
    7    340 ACCEPT    tcp  --  *       *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    tcp  --  *       *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    tcp  --  *       *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    tcp  --  *       *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    tcp  --  *       *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    all  --  *       *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    all  --  lo      *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    tcp  --  *       *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    tcp  --  *       *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    tcp  --  *       *       0.0.0.0/0  0.0.0.0/0
   49  1600 ACCEPT    icmp --  *       *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT    all  --  *       *       0.0.0.0/0  0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source    destination
    0     0 ACCEPT    all  --  tun0    eth0    0.0.0.0/0  0.0.0.0/0

```

Para ver las reglas NAT:

```
sudo iptables -t nat -L -nv
```

```

root@vps-813eb74d:/etc/openvpn/client# sudo iptables
Chain PREROUTING (policy ACCEPT 37652 packets, 1780K
 pkts bytes target    prot opt in     out     source
Chain INPUT (policy ACCEPT 15316 packets, 548K bytes)
 pkts bytes target    prot opt in     out     source
Chain OUTPUT (policy ACCEPT 63 packets, 5711 bytes)
 pkts bytes target    prot opt in     out     source
Chain POSTROUTING (policy ACCEPT 1884 packets, 124K b
 pkts bytes target    prot opt in     out     source

```

Iniciar o reiniciar Openvpn:


```
sudo service openvpn-server@server start
```

```
sudo systemctl restart openvpn-server@server.service
```

Comprobar el estado del servicio Openvpn:

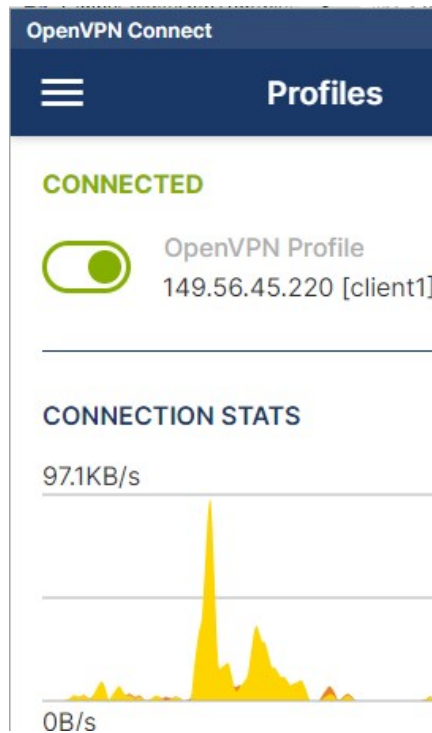
```
sudo service openvpn-server@server status
```

```
root@vps-813eb74d:/etc/openvpn/client# sudo service openvpn-server@server status
● openvpn-server@server.service - OpenVPN service for server
   Loaded: loaded (/lib/systemd/system/openvpn-server@.service; enabled; vendor preset: e
   Active: active (running) since Tue 2024-05-28 14:54:33 UTC; 2h 38min ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 96949 (openvpn)
   Status: "Initialization Sequence Completed"
    Tasks: 1 (limit: 2252)
   Memory: 2.1M
      CPU: 28.432s
   CGroup: /system.slice/system-openvpn\x2dserver.slice/openvpn-server@server.service
           └─96949 /usr/sbin/openvpn --status /run/openvpn-server/status-server.log --sta

May 28 17:14:17 vps-813eb74d openvpn[96949]: client1/79.116.208.86:61823 peer info: IV_PROT
May 28 17:14:17 vps-813eb74d openvpn[96949]: client1/79.116.208.86:61823 peer info: IV_MTU=
```


Finalmente para probar si funciona la vpn, descargar en el pc local la aplicación Openvpn connect y ejecutar el archivo client1.ovpn





IP Details For: 149.56.45.220

Decimal:	2503486940
Hostname:	vps-813eb74d.vps.ovh.ca
ASN:	16276
ISP:	OVH Hosting Inc.
Services:	Datacenter
Assignment:	Likely Static IP
Country:	Canada



(Opcional) Crear el fichero .ovpn mediante un script

Hay otra forma de hacer el fichero cliente .ovpn

Copiar el fichero plantilla.conf:

```
cp /etc/openvpn/client/client.conf /etc/openvpn/client/plantilla.conf
```

```
root@vps-813eb74d:/etc/openvpn/client# cp /etc/openvpn/client/client.conf /etc/openvpn/client/plantilla.conf
root@vps-813eb74d:/etc/openvpn/client# ls
```

Editar el fichero y dejar las claves comentadas:

```
sudo nano /etc/openvpn/client/plantilla.conf
```

```
# This file can be used to create a client configuration file
;ca ca.crt
;cert client.crt
;key client.key
```

crear el script que automatice el proceso de meter en un fichero un .ovpn:

```
sudo nano /etc/openvpn/client/make_config.sh
```

```
GNU nano 6.2
#!/bin/bash

# First argument: Client identifier

KEY_DIR=/etc/openvpn/client/keys
OUTPUT_DIR=/etc/openvpn/client/files
BASE_CONFIG=/etc/openvpn/client/plantilla.conf

cat ${BASE_CONFIG} \
  <(echo -e '<ca>') \
  ${KEY_DIR}/ca.crt \
  <(echo -e '</ca>\n<cert>') \
  ${KEY_DIR}/${1}.crt \
  <(echo -e '</cert>\n<key>')
```

crear el directorio files para guardar los ficheros .ovpn:

```
sudo mkdir /etc/openvpn/client/files
```

```
root@vps-813eb74d:/etc/openvpn/client# ls
client.conf  files  keys  make_config.sh  openvpn-st
```

Dar permisos de ejecución:

```
chmod 700 /etc/openvpn/client/make_config.sh
```

```
-rw-r--r-- 1 root root 3598 May 23 18:57
drwxr-xr-x 2 root root 4096 May 23 20:13
drwx----- 2 root root 4096 May 15 20:26
-rwx----- 1 root root 479 May 23 20:09
```

Generar el fichero ovpn poniéndole el mismo nombre que le habíamos dado a los certificados:

```
sudo ./make_config.sh cliente-padron
```

```
root@vps-813eb74d:/etc/openvpn/cl
cliente-padron.ovpn
```

Como este fichero tiene permisos de root, lo vamos a copiar en el directorio home/Ubuntu y le cambiamos los permisos al archivo para llevarlo a nuestro escritorio Windows y le cambiamos de propietario → Este archivo sería el que se le daría al cliente para que se conectara con openvpn:

```
cp /etc/openvpn/client/files/cliente-padron.ovpn /home/ubuntu/
```

```
ubuntu@vps-813eb74d:~$ ls
Asir cliente-padron.ovpn
```

```
cd /home/ubuntu/
```

```
sudo chmod 444 cliente-padron.ovpn
```

```
ubuntu@vps-813eb74d:~$ sudo chmod 444 cliente-padron.ovpn
```

```
sudo chown ubuntu:ubuntu cliente-padron.ovpn
```

```
ubuntu@vps-813eb74d:~$ sudo chown ubuntu:ubuntu cliente-padron.ovpn
```

```
ubuntu@vps-813eb74d:~$ ls -l
total 16
drwxrwxr-x 8 ubuntu ubuntu 4096 May 20 00:5
```