

UNIVERSIDAD LUTERANA SALVADOREÑA

FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA



GUIA #1

Configuración de openVPN de acceso remoto con clave estática compartida

GUIA #2

Configuración de openVPN basada en SSL/TLS

Docente: Licda. Ana Lissette Girón de Bermúdez

Asignatura: Administración de sistemas informáticos

Ciclo 1-2020

Integrantes:

Irene Magaly Beltrán Guzmán.

Dina Raquel Guzmán.

Katherine Esmeralda Pocasangre Serrano.

Luis Alberto Carranza Muñoz

GUIA #1

Configuración de openVPN de acceso remoto con clave estática compartida

Actualizamos la lista de paquetes utilizando apt-get update

```
root@servidor-VirtualBox:/home/servidor# apt-get update
Obj:1 http://sv.archive.ubuntu.com/ubuntu xenial InRelease
Des:2 http://sv.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Des:3 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Des:4 http://sv.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Des:5 http://sv.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [1,
141 kB]
77% [5 Packages 747 kB/1,141 kB 66%] 143 kB/s 2s
```

Instalamos openvpn utilizando apt-get install openvpn

```
root@servidor-VirtualBox:/home/servidor# apt-get install openvpn
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
libpkcs11-helper1
Paquetes sugeridos:
easy-rsa
Se instalarán los siguientes paquetes NUEVOS:
libpkcs11-helper1 openvpn
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 464 kB de archivos.
Se utilizarán 1,168 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://sv.archive.ubuntu.com/ubuntu xenial/main amd64 libpkcs11-helper1 am
d64 1.11-5 [44.0 kB]
Des:2 http://sv.archive.ubuntu.com/ubuntu xenial-updates/main amd64 openvpn amd6
4 2.3.10-1ubuntu2.2 [420 kB]
Descargados 464 kB en 4s (109 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete libpkcs11-helper1:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 95%
```

Se genera la llave compartida del lado del servidor, esta misma llave debemos de copiarla para nuestro cliente.

```
root@servidor-VirtualBox:/home/servidor# openvpn --genkey --secret static.key
root@servidor-VirtualBox:/home/servidor#
```

Se copia el archivo static.key hacia la siguiente ruta /etc/openvpn

```
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@ubuntuvpn:/home/user1# cp static.key /etc/openvpn
```

Una vez copiado la llave compartida, creamos el archivo de configuración nano **server.conf** y editamos el archivo que creamos anteriormente con el siguiente contenido.

```
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@ubuntuvpn:/etc/openvpn# nano server.conf
```



```
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.5.3           Archivo: /etc/openvpn/server.conf
```



```
dev tun
ifconfig 10.10.0.1 10.10.0.2
secret static.key
```

Ejecutamos el archivo de configuración de la vpn si todo está correcto deberíamos de ver un mensaje como el siguiente: **secuencia inicializada**

```
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@ubuntuvpn:/etc/openvpn# openvpn --config /etc/openvpn/server.conf
```

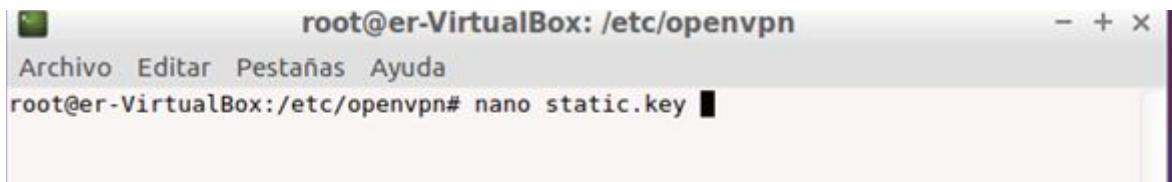
Verificamos que el túnel este creado correctamente con **ip add**

```
valid_lft forever preferred_lft forever
1: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 100
    link/none
    inet 10.10.0.1 peer 10.10.0.2/32 scope global tun0
        valid_lft forever preferred_lft forever
root@ubuntuvpn:/etc/openvpn#
```

Configuración del cliente

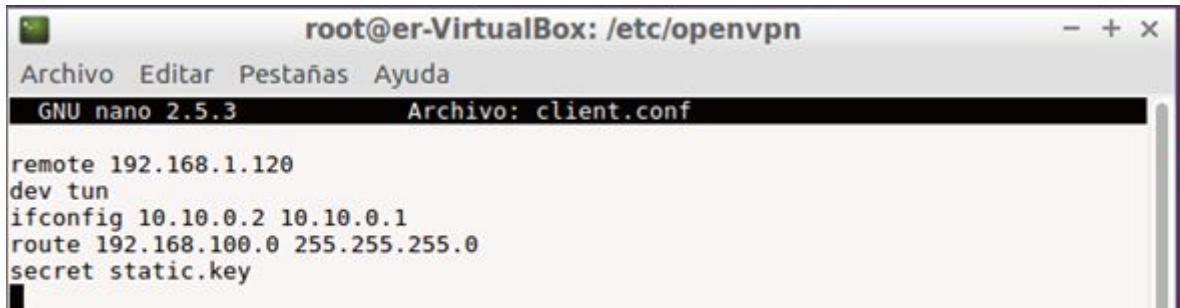
- Actualizamos la lista de paquetes apt-get update
- Instalamos openvpn apt-get install openvpn

Creamos el archivo que contendrá la llave, el contenido de este debe ser el mismo que tiene el servidor.



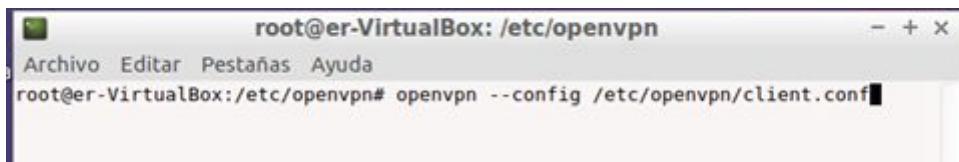
```
root@er-VirtualBox: /etc/openvpn
Archivo Editar Pestañas Ayuda
root@er-VirtualBox:/etc/openvpn# nano static.key
```

Dentro del directorio /etc/openvpn creamos el archivo de configuración del cliente con el siguiente contenido.



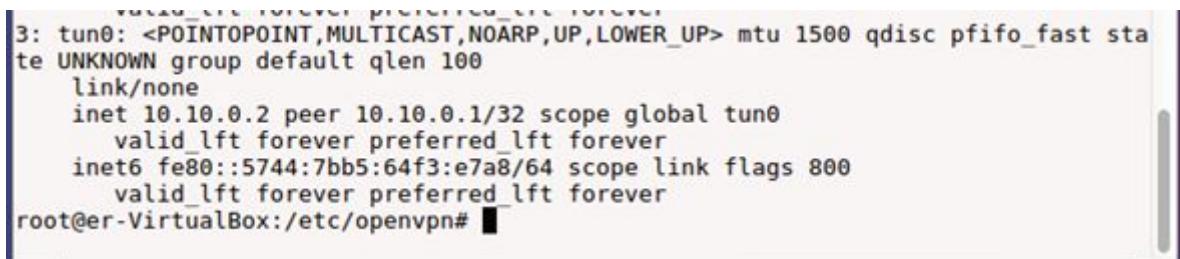
```
root@er-VirtualBox: /etc/openvpn
Archivo Editar Pestañas Ayuda
GNU nano 2.5.3           Archivo: client.conf
remote 192.168.1.120
dev tun
ifconfig 10.10.0.2 10.10.0.1
route 192.168.100.0 255.255.255.0
secret static.key
```

Ejecutamos el archivo de configuración del cliente.



```
root@er-VirtualBox: /etc/openvpn
Archivo Editar Pestañas Ayuda
root@er-VirtualBox:/etc/openvpn# openvpn --config /etc/openvpn/client.conf
```

Verificamos con **ip add** que el túnel se cree correctamente.



```
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 100
    link/none
    inet 10.10.0.2 peer 10.10.0.1/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::5744:7bb5:64f3:e7a8/64 scope link flags 800
        valid_lft forever preferred_lft forever
root@er-VirtualBox:/etc/openvpn#
```

Verificamos la conexión con nuestro servidor openvpn utilizando el comando **ping 10.10.0.1**

```
er@er-VirtualBox: ~
Archivo Editar Pestañas Ayuda
er@er-VirtualBox:~$ ping 10.10.0.1
PING 10.10.0.1 (10.10.0.1) 56(84) bytes of data.
64 bytes from 10.10.0.1: icmp_seq=1 ttl=64 time=0.560 ms
64 bytes from 10.10.0.1: icmp_seq=2 ttl=64 time=1.13 ms
64 bytes from 10.10.0.1: icmp_seq=3 ttl=64 time=1.31 ms
64 bytes from 10.10.0.1: icmp_seq=4 ttl=64 time=1.25 ms
64 bytes from 10.10.0.1: icmp_seq=5 ttl=64 time=1.34 ms
^C
--- 10.10.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4020ms
rtt min/avg/max/mdev = 0.560/1.121/1.341/0.291 ms
er@er-VirtualBox:~$
```

Verificamos conexión con nuestra red interna utilizando el comando **ping 192.168.100.2**

```
er@er-VirtualBox:~$ ping 192.168.100.2
PING 192.168.100.2 (192.168.100.2) 56(84) bytes of data.
64 bytes from 192.168.100.2: icmp_seq=1 ttl=64 time=0.487 ms
64 bytes from 192.168.100.2: icmp_seq=2 ttl=64 time=1.16 ms
64 bytes from 192.168.100.2: icmp_seq=3 ttl=64 time=1.28 ms
64 bytes from 192.168.100.2: icmp_seq=4 ttl=64 time=1.40 ms
^C
--- 192.168.100.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3015ms
rtt min/avg/max/mdev = 0.487/1.086/1.401/0.356 ms
er@er-VirtualBox:~$
```

GUIA #2

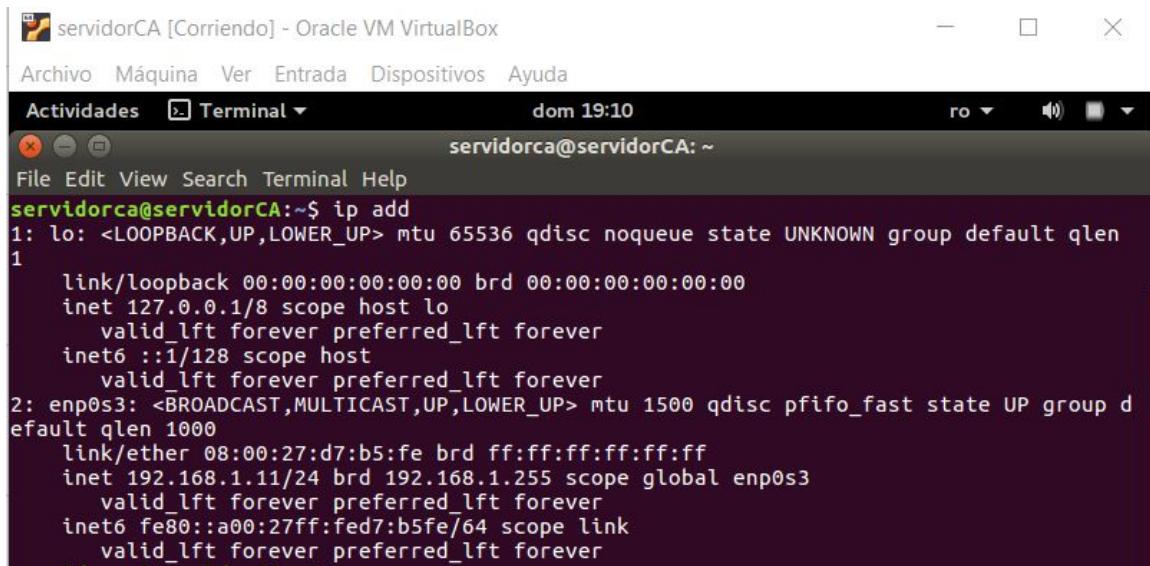
Configuración de openVPN basada en SSL/TLS

Requisitos previos

- Una máquina con Ubuntu server 16.04.3 que servirá como servidor VPN.
- Una máquina con Ubuntu server 16.04.3 que servirá como autoridad de certificación (CA).
- Un cliente
- OpenVPN instalado
- Tener SSH instalado

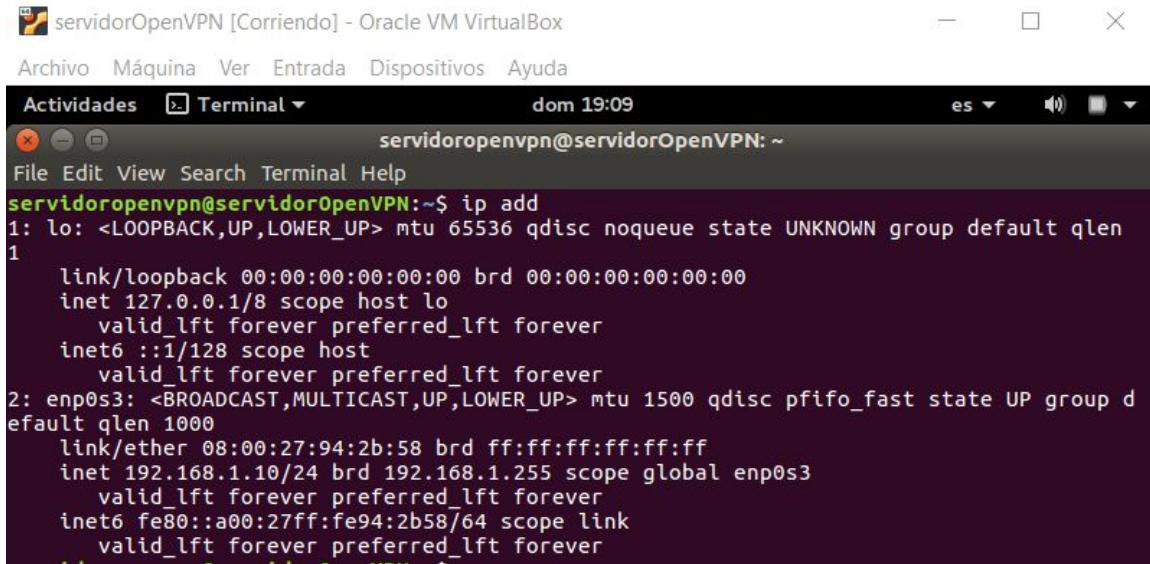
Paso 1: Instalar OpenVPN y EasyRSA

Colocamos el comando **ip add** para poder ver nuestra ip en este caso sera la **192.168.1.11** que esta sera la ip de nuestro **CA**.



```
servidorCA [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal dom 19:10
servidorca@servidorCA: ~
File Edit View Search Terminal Help
servidorca@servidorCA:~$ ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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 pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:d7:b5:fe brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fed7:b5fe/64 scope link
        valid_lft forever preferred_lft forever
```

Colocamos el comando **ip add** para poder ver nuestra ip en este caso sera la **192.168.1.10** que esta sera la ip de nuestro servidor **VPN**.



A screenshot of a terminal window titled "servidorOpenVPN [Corriendo] - Oracle VM VirtualBox". The window shows a list of network interfaces and their configurations. The output of the "ip add" command is displayed, including details for the loopback interface (lo) and the ethernet interface (enp0s3). The IP address 192.168.1.10 is listed under the enp0s3 interface.

```
servidoropenvpn@servidorOpenVPN:~$ ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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 pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:94:2b:58 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe94:2b58/64 scope link
        valid_lft forever preferred_lft forever
```

Instalamos OpenVPN en nuestro servidor VPN.

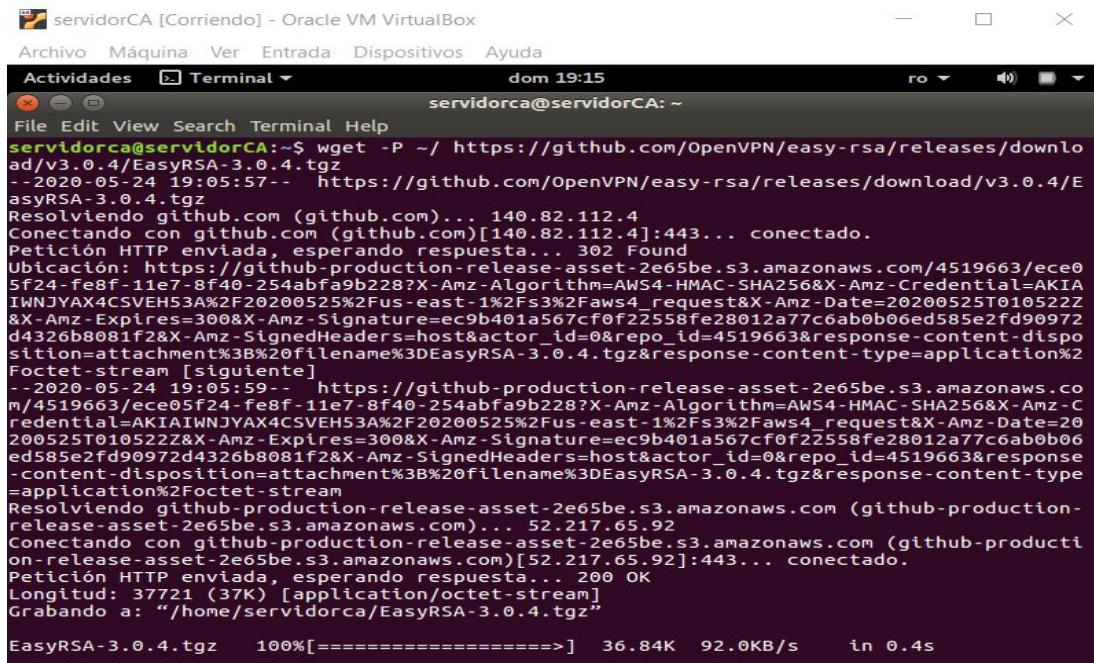


A screenshot of a terminal window titled "servidorOpenVPN [Corriendo] - Oracle VM VirtualBox". The window shows the root user executing the "apt install openvpn" command. The output indicates that the package is already at its latest version (2.3.10-1ubuntu2.2) and no updates are available.

```
root@servidorOpenVPN:/home/servidoropenvpn# apt install openvpn
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openvpn ya está en su versión más reciente (2.3.10-1ubuntu2.2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 200 no actualizados.
root@servidorOpenVPN:/home/servidoropenvpn#
```

A fin de comenzar a crear la infraestructura de CA y PKI, descargamos la última versión de EasyRSA tanto en la máquina de CA como en la del servidor de OpenVPN.

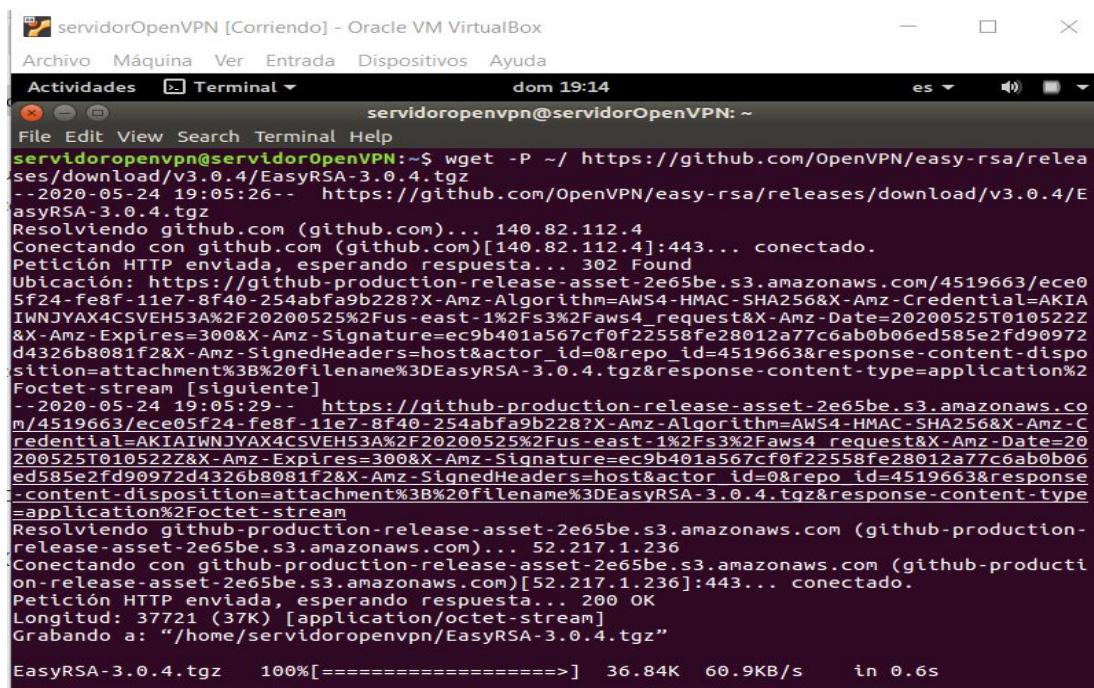
Maquina CA.



```
servidorCA [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal dom 19:15
servidorca@servidorCA: ~
File Edit View Search Terminal Help
servidorca@servidorCA:~$ wget -P ~/ https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.4/EasyRSA-3.0.4.tgz
--2020-05-24 19:05:57-- https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.4/EasyRSA-3.0.4.tgz
Resolviendo github.com (github.com)... 140.82.112.4
Conectando con github.com (github.com)[140.82.112.4]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Ubicación: https://github-production-release-asset-2e65be.s3.amazonaws.com/4519663/ece05f24-fe8f-11e7-8f40-254abfa9b2287X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20200525%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200525T010522Z&X-Amz-Expires=3008X-Amz-Signature=ec9b401a567cf0f22558fe28012a77c6ab0b06ed585e2fd90972d4326b8081f2&X-Amz-SignedHeaders=host&actor_id=0&repo_id=4519663&response-content-disposition=attachment%3B%20filename%3DEasyRSA-3.0.4.tgz&response-content-type=application%2Foctet-stream [siguiente]
--2020-05-24 19:05:59-- https://github-production-release-asset-2e65be.s3.amazonaws.com/4519663/ece05f24-fe8f-11e7-8f40-254abfa9b2287X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20200525%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200525T010522Z&X-Amz-Expires=3008X-Amz-Signature=ec9b401a567cf0f22558fe28012a77c6ab0b06ed585e2fd90972d4326b8081f2&X-Amz-SignedHeaders=host&actor_id=0&repo_id=4519663&response-content-disposition=attachment%3B%20filename%3DEasyRSA-3.0.4.tgz&response-content-type=application%2Foctet-stream
Resolviendo github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 52.217.65.92
Conectando con github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)[52.217.65.92]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 37721 (37K) [application/octet-stream]
Grabando a: "/home/servidorca/EasyRSA-3.0.4.tgz"

EasyRSA-3.0.4.tgz 100%[=====] 36.84K 92.0KB/s in 0.4s
```

Servidor VPN.

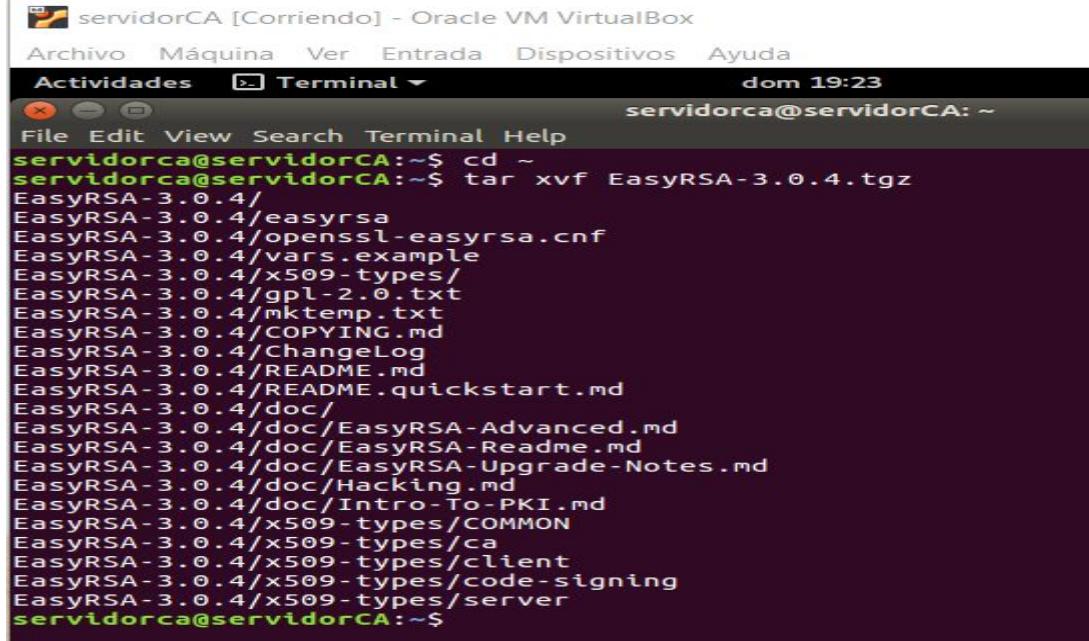


```
servidorOpenVPN [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal dom 19:14
servidoropenvpn@servidorOpenVPN: ~
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~$ wget -P ~/ https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.4/EasyRSA-3.0.4.tgz
--2020-05-24 19:05:26-- https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.4/EasyRSA-3.0.4.tgz
Resolviendo github.com (github.com)... 140.82.112.4
Conectando con github.com (github.com)[140.82.112.4]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Ubicación: https://github-production-release-asset-2e65be.s3.amazonaws.com/4519663/ece05f24-fe8f-11e7-8f40-254abfa9b2287X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20200525%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200525T010522Z&X-Amz-Expires=3008X-Amz-Signature=ec9b401a567cf0f22558fe28012a77c6ab0b06ed585e2fd90972d4326b8081f2&X-Amz-SignedHeaders=host&actor_id=0&repo_id=4519663&response-content-disposition=attachment%3B%20filename%3DEasyRSA-3.0.4.tgz&response-content-type=application%2Foctet-stream [siguiente]
--2020-05-24 19:05:29-- https://github-production-release-asset-2e65be.s3.amazonaws.com/4519663/ece05f24-fe8f-11e7-8f40-254abfa9b2287X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20200525%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200525T010522Z&X-Amz-Expires=3008X-Amz-Signature=ec9b401a567cf0f22558fe28012a77c6ab0b06ed585e2fd90972d4326b8081f2&X-Amz-SignedHeaders=host&actor_id=0&repo_id=4519663&response-content-disposition=attachment%3B%20filename%3DEasyRSA-3.0.4.tgz&response-content-type=application%2Foctet-stream
Resolviendo github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 52.217.1.236
Conectando con github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)[52.217.1.236]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 37721 (37K) [application/octet-stream]
Grabando a: "/home/servidoropenvpn/EasyRSA-3.0.4.tgz"

EasyRSA-3.0.4.tgz 100%[=====] 36.84K 60.9KB/s in 0.6s
```

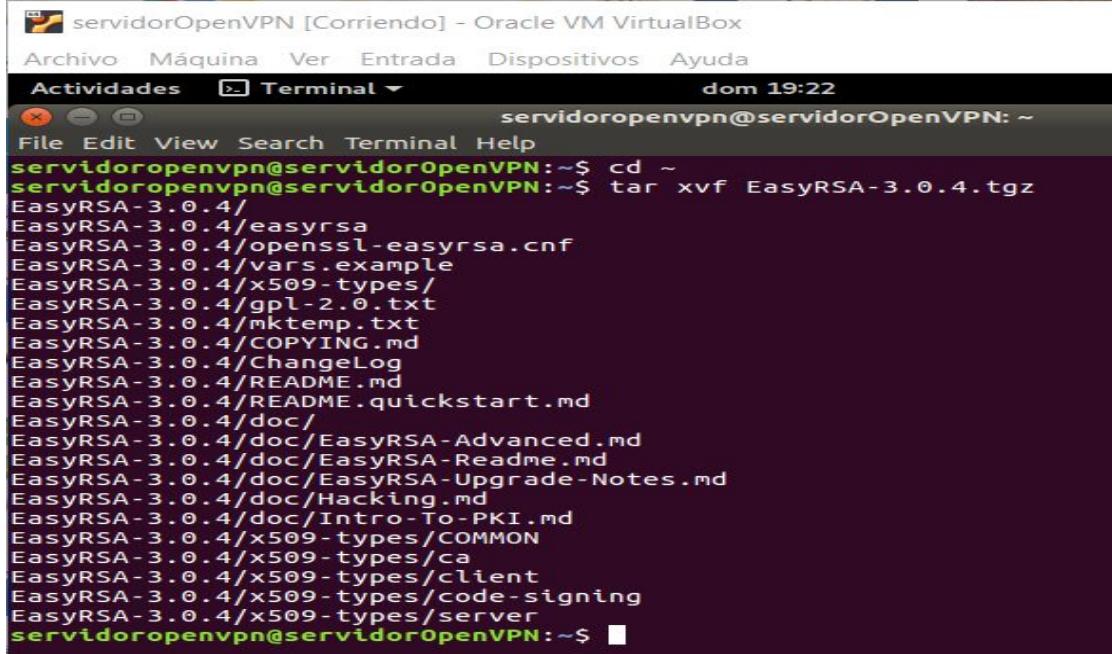
Luego descomprimimos el archivo .tgz de EasyRSA tanto en la máquina de CA como en la del servidor de OpenVPN.

Maquina CA.



```
servidorCA [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal dom 19:23
servidorca@servidorCA: ~
File Edit View Search Terminal Help
servidorca@servidorCA:~$ cd ~
servidorca@servidorCA:~$ tar xvf EasyRSA-3.0.4.tgz
EasyRSA-3.0.4/
EasyRSA-3.0.4/easyrsa
EasyRSA-3.0.4/openssl-easyrsa.cnf
EasyRSA-3.0.4/vars.example
EasyRSA-3.0.4/x509-types/
EasyRSA-3.0.4/gpl-2.0.txt
EasyRSA-3.0.4/mktemp.txt
EasyRSA-3.0.4/COPYING.md
EasyRSA-3.0.4/ChangeLog
EasyRSA-3.0.4/README.md
EasyRSA-3.0.4/README.quickstart.md
EasyRSA-3.0.4/doc/
EasyRSA-3.0.4/doc/EasyRSA-Advanced.md
EasyRSA-3.0.4/doc/EasyRSA-Readme.md
EasyRSA-3.0.4/doc/EasyRSA-Upgrade-Notes.md
EasyRSA-3.0.4/doc/Hacking.md
EasyRSA-3.0.4/doc/Intro-To-PKI.md
EasyRSA-3.0.4/x509-types/COMMON
EasyRSA-3.0.4/x509-types/ca
EasyRSA-3.0.4/x509-types/client
EasyRSA-3.0.4/x509-types/code-signing
EasyRSA-3.0.4/x509-types/server
servidorca@servidorCA:~$
```

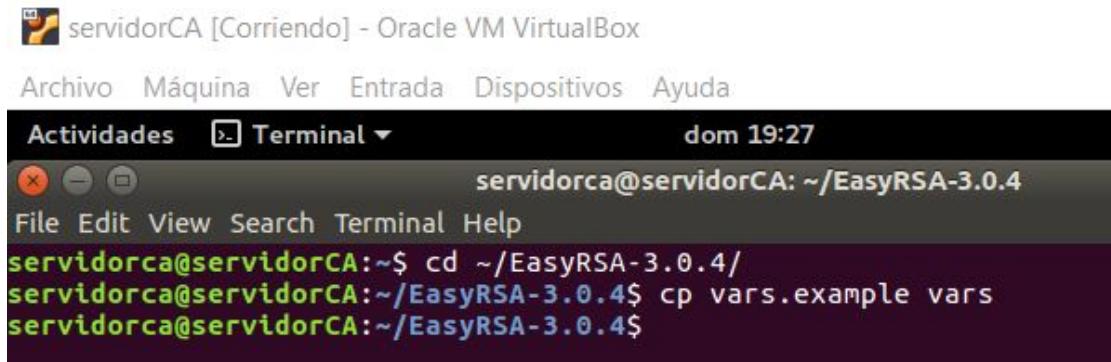
Servidor VPN.



```
servidorOpenVPN [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal dom 19:22
servidoropenvpn@servidorOpenVPN: ~
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~$ cd ~
servidoropenvpn@servidorOpenVPN:~$ tar xvf EasyRSA-3.0.4.tgz
EasyRSA-3.0.4/
EasyRSA-3.0.4/easyrsa
EasyRSA-3.0.4/openssl-easyrsa.cnf
EasyRSA-3.0.4/vars.example
EasyRSA-3.0.4/x509-types/
EasyRSA-3.0.4/gpl-2.0.txt
EasyRSA-3.0.4/mktemp.txt
EasyRSA-3.0.4/COPYING.md
EasyRSA-3.0.4/ChangeLog
EasyRSA-3.0.4/README.md
EasyRSA-3.0.4/README.quickstart.md
EasyRSA-3.0.4/doc/
EasyRSA-3.0.4/doc/EasyRSA-Advanced.md
EasyRSA-3.0.4/doc/EasyRSA-Readme.md
EasyRSA-3.0.4/doc/EasyRSA-Upgrade-Notes.md
EasyRSA-3.0.4/doc/Hacking.md
EasyRSA-3.0.4/doc/Intro-To-PKI.md
EasyRSA-3.0.4/x509-types/COMMON
EasyRSA-3.0.4/x509-types/ca
EasyRSA-3.0.4/x509-types/client
EasyRSA-3.0.4/x509-types/code-signing
EasyRSA-3.0.4/x509-types/server
servidoropenvpn@servidorOpenVPN:~$
```

Paso 2: Configurar las variables de EasyRSA y crear la CA.

En la máquina CA nos dirigimos al directorio de EasyRSA ,dentro de este directorio, hay un archivo llamado vars.example. Hacemos una copia de este archivo y asignamos a esta el nombre vars sin agregar una extensión.



```
servidorca@servidorCA:~/EasyRSA-3.0.4$ cp vars.example vars
```

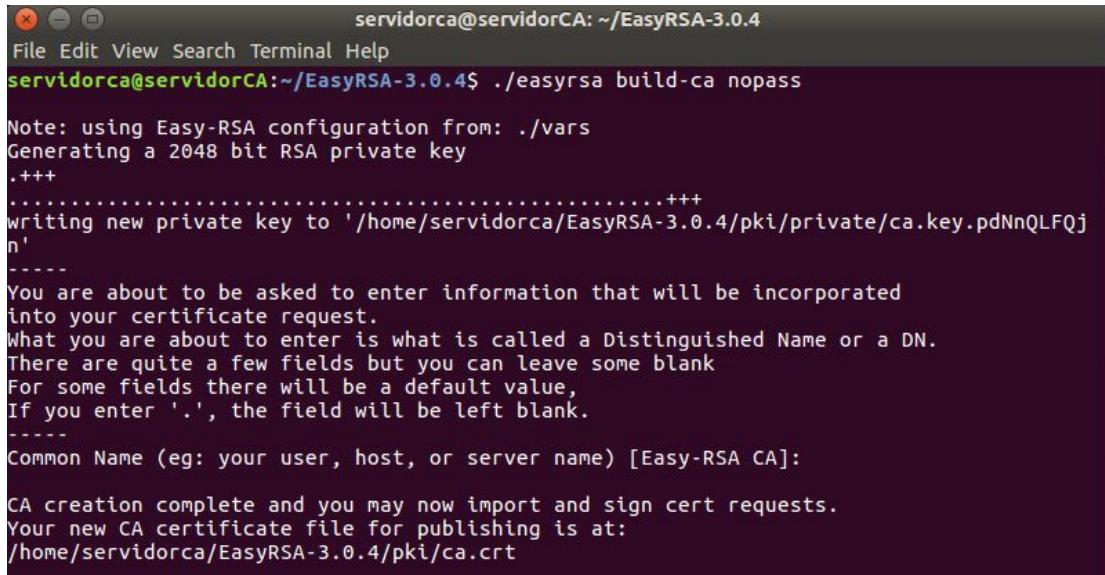
Abrimos el archivo con nano vars y buscaremos los ajustes que establecen los valores de campos predeterminados para nuevos certificados.

```
set_var EASYRSA_REQ_COUNTRY "ES"
set_var EASYRSA_REQ_PROVINCE "La Paz"
set_var EASYRSA_REQ_CITY "San Juan Nonualco"
set_var EASYRSA_REQ_ORG "Copyleft Certificate"
set_var EASYRSA_REQ_EMAIL "server@gmail.com"
set_var EASYRSA_REQ_OU "ServidorCA"
```

Dentro del directorio EasyRSA hay una secuencia de comandos llamada easyrsa, que se usan para llevar a cabo varias tareas relacionadas con la creación y administración de la CA. Ejecutamos la secuencia de comandos con **./easyrsa init-pki** para iniciar la infraestructura de clave pública en el servidor de CA.

```
servidorca@servidorCA:~/EasyRSA-3.0.4$ ./easyrsa init-pki
Note: using Easy-RSA configuration from: ./vars
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/servidorca/EasyRSA-3.0.4/pki
servidorca@servidorCA:~/EasyRSA-3.0.4$
```

Luego ejecutamos la secuencia de comandos easyrsa nuevamente, seguida de la opción build-ca. Con esto, se crearán la CA y dos archivos importantes, ca.crt y ca.key, que representarán los lados públicos y privados de un certificado SSL y si no deseamos que nos pida contraseña le colocamos nopass al final.



```
servidorca@servidorCA: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidorca@servidorCA:~/EasyRSA-3.0.4$ ./easyrsa build-ca nopass

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
.+++
.....+
writing new private key to '/home/servidorca/EasyRSA-3.0.4/pki/private/ca.key.pdNnQLFQj
n'
-----
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/servidorca/EasyRSA-3.0.4/pki/ca.crt
```

Paso 3: Crear los archivos de certificado, clave y cifrado del servidor

Nos dirigiremos al directorio EasyRSA en nuestro **servidor de OpenVPN** de allí ejecutaremos la secuencia de comandos easyrsa con el comando ./easyrsa init-pki. Aunque ya ejecutó este comando en la máquina de CA, es necesario ejecutarlo aquí porque su servidor y su CA tendrán directorios de PKI independientes.



```
Actividades Terminal ▾ dom 19:56
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~$ cd EasyRSA-3.0.4/
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ ./easyrsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/servidoropenvpn/EasyRSA-3.0.4/pki
```

Luego ejecutaremos la secuencia de comandos easyrsa nuevamente, esta vez con la opción gen-req seguida de un nombre común para la máquina en este caso será **server** seguida de la opción nopass para que no nos pida una contraseña.

```

servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ ./easyrsa gen-req server nopass
Generating a 2048 bit RSA private key
.....+
.....+
writing new private key to '/home/servidoropenvpn/EasyRSA-3.0.4/pki/private/server.key.
XtSacFMQ3j'
-----
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) [server]:
Keypair and certificate request completed. Your files are:
req: /home/servidoropenvpn/EasyRSA-3.0.4/pki/reqs/server.req
key: /home/servidoropenvpn/EasyRSA-3.0.4/pki/private/server.key

```

Con el paso anterior se creó una clave privada para el servidor y un archivo de solicitud de certificado llamado server.req, copiamos la clave del servidor al directorio /etc/openvpn/.

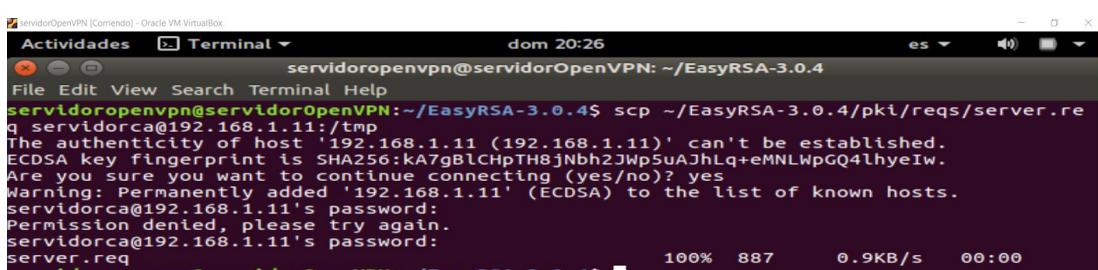


```

Actividades Terminal dom 20:10
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo cp ~/EasyRSA-3.0.4/pki/private/se
rver.key /etc/openvpn/
[sudo] password for servidoropenvpn:
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ 

```

Luego transferiremos el archivo server.req a nuestra máquina de CA utilizando SSH pondremos scp ~/EasyRSA-3.0.4/pki/reqs/server.req seguido del nombre y dirección ip de nuestra máquina **CA**.

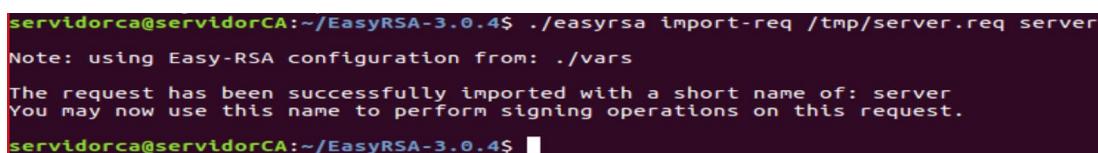


```

Actividades Terminal dom 20:26
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ scp ~/EasyRSA-3.0.4/pki/reqs/server.re
q servidorca@192.168.1.11:/tmp
The authenticity of host '192.168.1.11 (192.168.1.11)' can't be established.
ECDSA key fingerprint is SHA256:kA7g8lCHpTH8jNbh2JWp5uAJhLq+eMNLWpGQ4lhyeIw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.11' (ECDSA) to the list of known hosts.
servidorca@192.168.1.11's password:
Permission denied, please try again.
servidorca@192.168.1.11's password:
server.req
100% 887 0.9KB/s 00:00

```

Luego en nuestra **máquina CA** usando nuevamente la secuencia de comandos easyrsa, importamos el archivo server.req y seguiremos la ruta de este con su nombre común en este caso **server**.

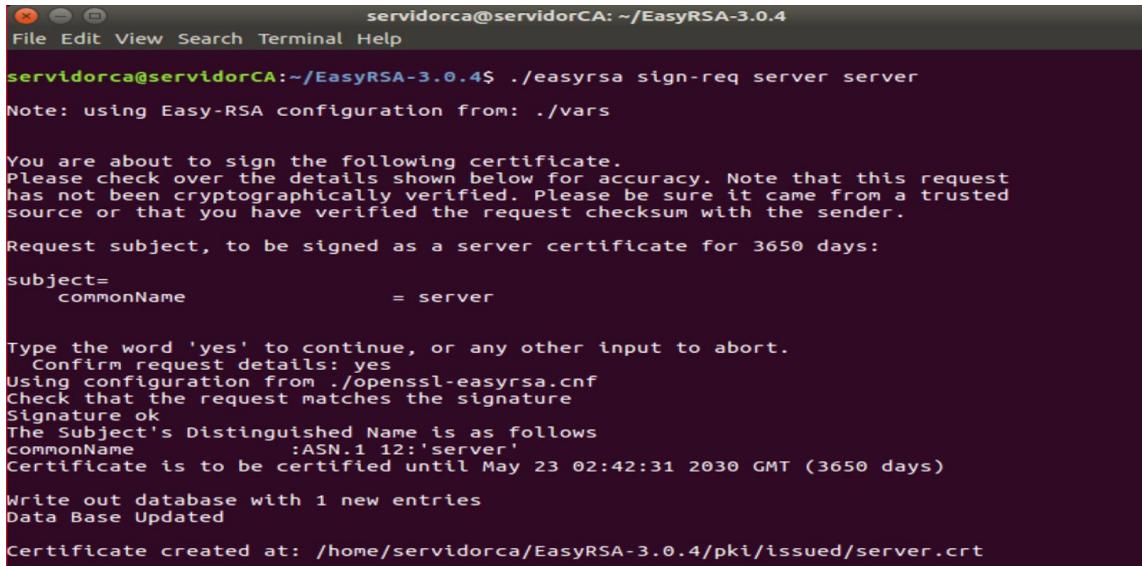


```

servidorca@servidorCA:~/EasyRSA-3.0.4$ ./easyrsa import-req /tmp/server.req server
Note: using Easy-RSA configuration from: ./vars
The request has been successfully imported with a short name of: server
You may now use this name to perform signing operations on this request.
servidorca@servidorCA:~/EasyRSA-3.0.4$ 

```

Luego firmaremos la solicitud ejecutando la secuencia easyrsa con la opción sign-req ,seguida del tipo de solicitud y el nombre común. El tipo de solicitud puede ser client o server. Por ello, para la solicitud de certificado del servidor de OpenVPN asegúrese de usar el tipo de solicitud server.



```
servidorca@servidorCA: ~/EasyRSA-3.0.4$ ./easyrsa sign-req server server
Note: using Easy-RSA configuration from: ./vars

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.

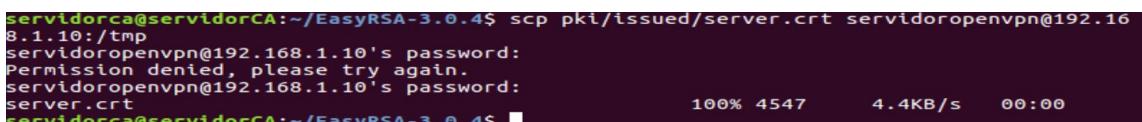
Request subject, to be signed as a server certificate for 3650 days:
subject=
    commonName      = server

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
Using configuration from ./openssl-easyrsa.cnf
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 May 23 02:42:31 2030 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

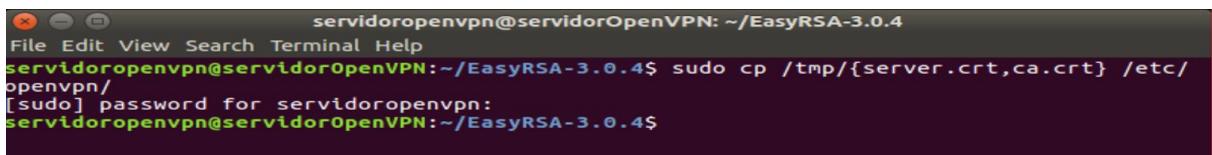
Certificate created at: /home/servidorca/EasyRSA-3.0.4/pki/issued/server.crt
```

Luego transferiremos el certificado firmado de vuelta a nuestro servidor de VPN con un método seguro en nuestro caso por ssh, pondremos scp pki/issued/server.crt seguido del nombre y la dirección ip de nuestro servidor VPN.



```
servidorca@servidorCA:~/EasyRSA-3.0.4$ scp pki/issued/server.crt servidoropenvpn@192.168.1.10:/tmp
servidoropenvpn@192.168.1.10's password:
Permission denied, please try again.
servidoropenvpn@192.168.1.10's password:
server.crt                                         100% 4547      4.4KB/s   00:00
servidorca@servidorCA:~/EasyRSA-3.0.4$
```

Nuevamente en nuestra máquina **servidor VPN** copiamos los archivos server.crt y ca.cart a su directorio /etc/openvpn/



```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4$ File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo cp /tmp/{server.crt,ca.crt} /etc/openvpn/
[sudo] password for servidoropenvpn:
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```

Creamos una clave segura Diffie-Hellman para usarla durante el intercambio de claves escribiendo el comando ./easyrsa gen-dh

```
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ ./easyrsa gen-dh
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.
.
.
DH parameters of size 2048 created at /home/servidoropenvpn/EasyRSA-3.0.4/pki/dh.p
```

Una vez que se complete, generamos una firma HMAC para fortalecer las capacidades de verificación de integridad TLS del servidor con el comando `openvpn --genkey --secret ta.key`

```
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ openvpn --genkey --secret ta.key  
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```

Cuando el comando se aplique, copiamos los dos nuevos archivos a su directorio /etc/openvpn/. Con esto, se generarán todos los archivos de certificados y claves necesarios para nuestro servidor

```
Actividades Terminal dom 21:22 es ▾
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo cp ~/EasyRSA-3.0.4/tap.key /etc/openvpn/
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo cp ~/EasyRSA-3.0.4/pki/dh.pem /etc/openvpn/
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ █
```

Paso 4: Generar un par de certificado y clave de cliente

Crearemos una estructura de directorios dentro de nuestro directorio de inicio para almacenar los archivos de certificado y clave de cliente.

```
Actividades Terminal dom 21:24 es ⓘ ⓘ ⓘ
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ mkdir -p ~/client-configs/keys
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ █
```

Debido a que almacenará los pares de certificado y clave de los clientes y los archivos de configuración en este directorio, se deben bloquear sus permisos ahora como medida de seguridad con el comando chmod -R 700 ~/client-configs.

```
servidoropenvpn@servidorOpenVPN: ~/client-configs  
File Edit View Search Terminal Help  
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ chmod -R 700 ~/client-configs
```

Luego, nos dirigimos al directorio EasyRSA y ejecute la secuencia de comandos easyrsa con las opciones gen-req y nopass, junto con el nombre común para el cliente en este caso es **client1**.

```

servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ ./easyrsa gen-req client1 nopass
Generating a 2048 bit RSA private key
.....+ ++
writing new private key to '/home/servidoropenvpn/EasyRSA-3.0.4/pki/private/client1.key'
.TDwKFCVF9B'
-----
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) [client1]:
Keypair and certificate request completed. Your files are:
req: /home/servidoropenvpn/EasyRSA-3.0.4/pki/reqs/client1.req
key: /home/servidoropenvpn/EasyRSA-3.0.4/pki/private/client1.key

```

Luego, copiamos el archivo client1.key al directorio /client-configs/keys/ que creamos antes

```

servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ cp pki/private/client1.key ~/client-co
nfigs/keys/
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ █

```

Después transferimos el archivo client1.req a nuestra máquina de CA usando SSH poniendo scp pki/reqs/client1.req seguido del nombre de nuestra máquina CA y nuestra dirección ip

```

Actividades Terminal dom 21:32
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ scp pki/reqs/client1.req servidorca@19
2.168.1.11:/tmp
servidorca@192.168.1.11's password:
client1.req                                100%   887      0.9KB/s   00:00
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ █

```

Iniciamos sesión en nuestra máquina de CA por medio de SSH, y luego nos vamos hasta el directorio EasyRSA e importe la solicitud de certificado

```

Actividades Terminal dom 21:36
servidorca@servidorCA: ~
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ ssh servidorca@192.168.1.11
servidorca@192.168.1.11's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

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

Pueden actualizarse 210 paquetes.
134 actualizaciones son de seguridad.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat May 23 23:11:27 2020

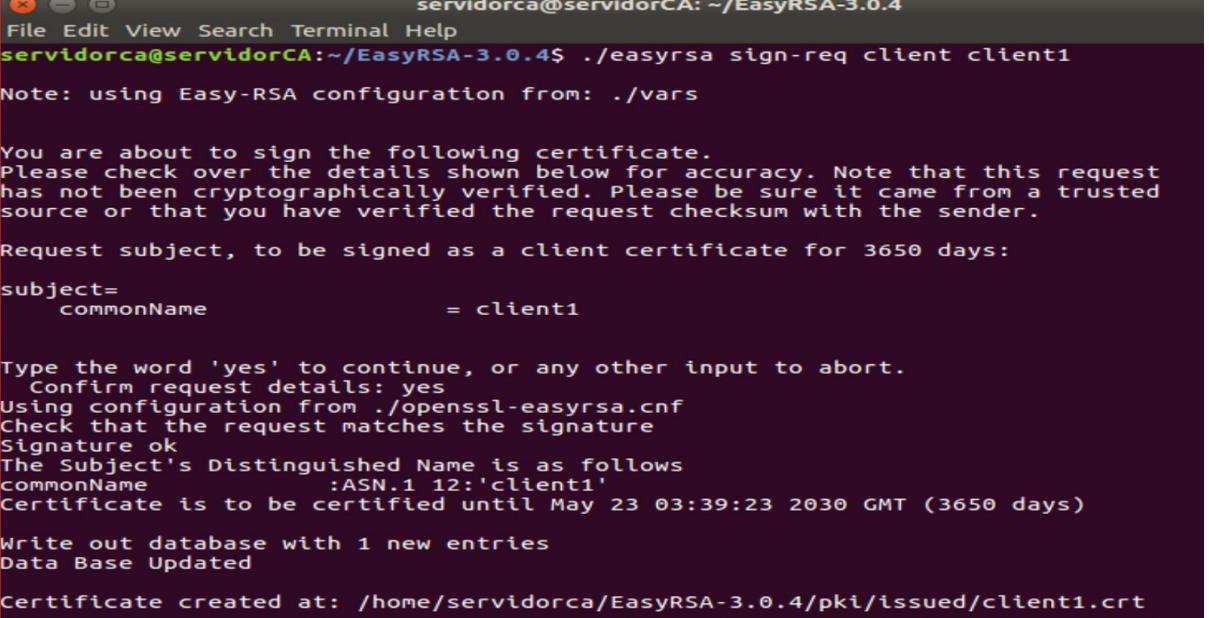
```

```

servidorca@servidorCA:~$ cd EasyRSA-3.0.4/
servidorca@servidorCA:~/EasyRSA-3.0.4$ ./easysrsa import-req /tmp/client1.req client1
Note: using Easy-RSA configuration from: ./vars
The request has been successfully imported with a short name of: client1
You may now use this name to perform signing operations on this request.
servidorca@servidorCA:~/EasyRSA-3.0.4$ █

```

Luego se firmará la solicitud como se hizo en el caso del servidor en el paso anterior. Esta vez, nos aseguraremos de especificar el tipo de solicitud client



```

servidorca@servidorCA: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidorca@servidorCA:~/EasyRSA-3.0.4$ ./easysrsa sign-req client client1
Note: using Easy-RSA configuration from: ./vars

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 client certificate for 3650 days:
subject=
    commonName          = client1

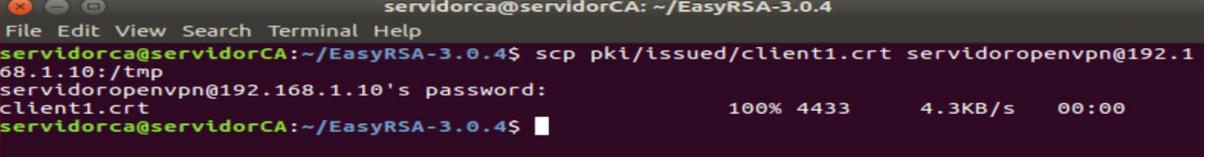
Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
Using configuration from ./openssl-easysrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName          :ASN.1 12:'client1'
Certificate is to be certified until May 23 03:39:23 2030 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/servidorca/EasyRSA-3.0.4/pki/issued/client1.crt

```

Con el siguiente, se creará un archivo de certificado de cliente llamado client1.crt. vamos a transferir este archivo de vuelta al servidor VPN con ssh.



```

servidorca@servidorCA: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidorca@servidorCA:~/EasyRSA-3.0.4$ scp pki/issued/client1.crt servidoropenvpn@192.168.1.10:/tmp
servidoropenvpn@192.168.1.10's password: client1.crt
100% 4433      4.3KB/s   00:00
servidorca@servidorCA:~/EasyRSA-3.0.4$ █

```

Luego retornamos de ssh a nuestro servidor VPN y copiamos el certificado de cliente al directorio **/client-configs/keys/** después de hacer esto copiamos también los archivos **ca.crt** y **ta.key** al directorio **/client-configs/keys/**.

```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ cp /tmp/client1.crt ~/client-configs/keys/
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ cp ~/EasyRSA-3.0.4/ta.key ~/client-configs/keys/
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo cp /etc/openvpn/ca.crt ~/client-configs/keys/
[sudo] password for servidoropenvpn:
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```

Paso 5: Configurar el servicio de OpenVPN

Ahora que generamos los certificados y las claves vamos a configurar nuestro servicio de OpenVPN vamos a copiar el archivo de configuración de OpenVPN al directorio de configuración y luego lo vamos a extraer.

```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```



```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo gzip -d /etc/openvpn/server.conf.gz
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```

Abrimos el archivo de configuración del servidor con el comando nano **/etc/openvpn/server.conf** y modificamos lo siguiente.

Buscamos la directiva **tls-auth** para encontrar la sección HMAC. Los comentarios de esta línea no deberían existir, pero si esto no sucede elimine ";" para quitar los comentarios

```
# 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 ta.key 0 # This file is secret
```

Luego buscamos las líneas **cipher** y el código **AES-256-CBC** una vez encontrados nos aseguramos de que no estén comentados si es el caso se elimina ";". Y agregamos una directiva **auth** con **SHA256** esto para seleccionar el algoritmo de codificación de mensajes HMAC.

```
# Select a cryptographic cipher.  
# This config item must be copied to  
# the client config file as well.  
;cipher BF-CBC           # Blowfish (default)  
;cipher AES-128-CBC      # AES  
;cipher DES-EDE3-CBC     # Triple-DES  
cipher AES-256-CBC  
auth SHA256
```

Después buscamos la línea que contenga la directiva **dh** que define los parámetros Diffie-Hellman. Debido a algunos cambios recientes realizados en EasyRSA, el nombre de archivo de la clave Diffie-Hellman puede ser distinto del que figura en el ejemplo del archivo de configuración del servidor. Si es necesario, se tiene que cambiar el nombre de archivo que aparece eliminando **2048** para que coincida con la clave que generó en el paso anterior.

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

Luego verificamos estas líneas de código para ver si está correcta.

```
cert server.crt  
key server.key # This file should be kept secret
```

Paso 6: Ajustar la configuración de redes del servidor

Para ajustar el enrutamiento de IP predeterminado de su servidor, modificaremos el archivo **/etc/sysctl.conf** luego buscamos la línea con comentarios que configura **net.ipv4.ip_forward** y se elimina el carácter “#” y sólo se guarda.

```
# Uncomment the next line to enable packet forwarding for IPv4  
net.ipv4.ip_forward=1
```

Para leer el archivo y modificar los valores de la sesión actual, escribiremos el comando **sudo sysctl -p**.

```
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo sysctl -p
net.ipv4.ip_forward = 1
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ █
```

Antes de abrir el archivo de configuración de firewall para agregar las reglas de enmascaramiento, primero debemos encontrar la interfaz de red pública de nuestra máquina.

```
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ ip route | grep default
default via 192.168.1.1 dev enp0s3
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ █
```

Ahora **sudo nano /etc/ufw/before.rules** y colocamos todo este código con el nombre de nuestra red en el caso de nosotros es la **enp0s3**.

```
# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to enp0s3 (change to the interface you discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o enp0s3 -j MASQUERADE
COMMIT
# END OPENVPN RULES
```

Luego abrimos el archivo **nano /etc/default/ufw** dentro de este, encontraremos la directiva **DEFAULT_FORWARD_POLICY** y le cambiaremos el valor de **DROP** a **ACCEPT**.

```
# Set the default forward policy to ACCEPT, DROP or REJECT. Please note that
# if you change this you will most likely want to adjust your rules
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Luego modificaremos el firewall para permitir el tráfico hacia OpenVPN. Si no se cambio el puerto y el protocolo en el archivo **/etc/openvpn/server.conf**, deberá abrir el tráfico UDP al puerto **1194** y en caso que no hayamos agregado el ssh lo podemos agregar.

```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo ufw allow 1194/udp
Reglas actualizadas
Reglas actualizadas (v6)
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo ufw allow OpenSSH
Reglas actualizadas
Reglas actualizadas (v6)
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```

Luego de agregar esas reglas, deshabilitamos y volvemos a habilitar UFW para reiniciarlo y que cargue los cambios de todos los archivos que se hayan modificado.

```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo ufw disable
El cortafuegos está detenido y deshabilitado en el arranque del sistema
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```

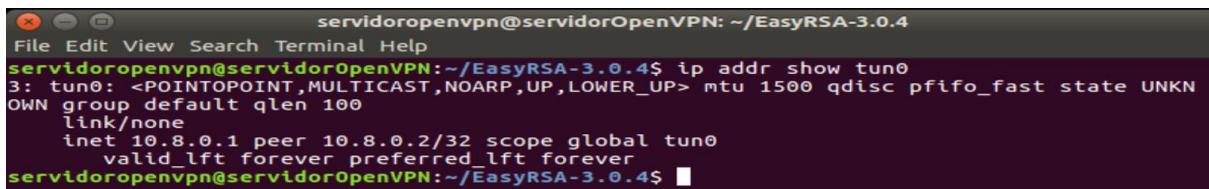
Paso 7: Iniciar y habilitar el servicio de OpenVPN

Luego iniciamos los servicios de OpenVPN con **systemctl start openvpn@server** y para ver que el servicio esté funcionando correctamente colocamos **systemctl status openvpn@server** si se le coloco otro nombre al archivo deberá cambiarse pero en el caos de nosotros en **server**.

```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo systemctl start openvpn@server
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo systemctl status openvpn@server
● openvpn@server.service - OpenVPN connection to server
  Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)
  Active: active (running) since dom 2020-05-24 22:58:54 CST; 24s ago
    Docs: man:openvpn(8)
          https://community.openvpn.net/openvpn/wiki/Openvpn23ManPage
          https://community.openvpn.net/openvpn/wiki/HOWTO
   Process: 6703 ExecStart=/usr/sbin/openvpn --daemon ovpn-%i --status /run/openvpn/%i.s
   Main PID: 6707 (openvpn)
      CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
              └─6707 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.s

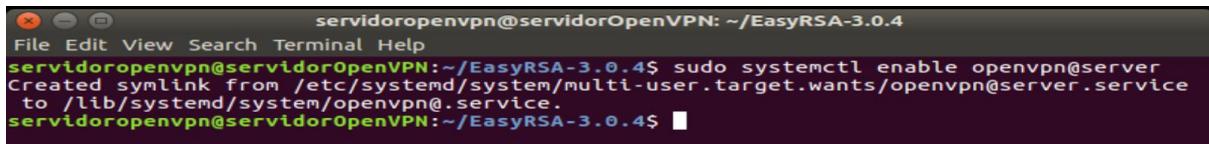
may 24 22:58:54 servidorOpenVPN ovpn-server[6707]: /sbin/ip addr add dev tun0 local 10.
may 24 22:58:54 servidorOpenVPN ovpn-server[6707]: /sbin/ip route add 10.8.0.0/24 via 1
may 24 22:58:54 servidorOpenVPN ovpn-server[6707]: GID set to nogroup
may 24 22:58:54 servidorOpenVPN ovpn-server[6707]: UID set to nobody
may 24 22:58:54 servidorOpenVPN ovpn-server[6707]: UDPv4 link local (bound): [undef]
may 24 22:58:54 servidorOpenVPN ovpn-server[6707]: UDPv4 link remote: [undef]
may 24 22:58:54 servidorOpenVPN ovpn-server[6707]: MULTI: multi_init called, r=256 v=25
may 24 22:58:54 servidorOpenVPN ovpn-server[6707]: IFCONFIG POOL: base=10.8.0.4 size=62
may 24 22:58:54 servidorOpenVPN ovpn-server[6707]: IFCONFIG POOL LIST
may 24 22:58:54 servidorOpenVPN ovpn-server[6707]: Initialization Sequence Completed
lines 1-21/21 (END)
```

También se puede controlar que la interfaz tun0 de OpenVPN esté disponible con el comando **ip addr show tun0**.



```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ ip addr show tun0
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKN
OWN group default qlen 100
    link/none
        inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
            valid_lft forever preferred_lft forever
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```

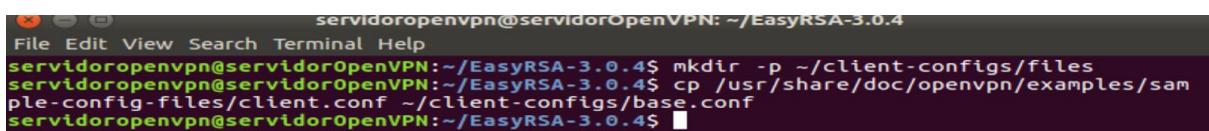
Luego de iniciar el servicio, lo habilitaremos para que se cargue de manera automática en el inicio.



```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ sudo systemctl enable openvpn@server
Created symlink from /etc/systemd/system/multi-user.target.wants/openvpn@server.service
to /lib/systemd/system/openvpn@.service.
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```

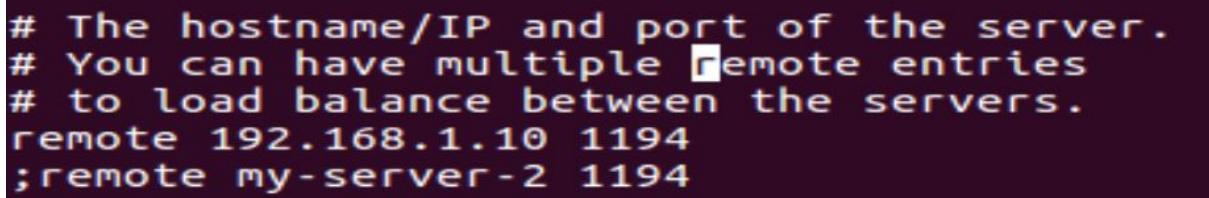
Paso 8: Crear la infraestructura de configuración de clientes

Comenzaremos creando un nuevo directorio con el comando `mkdir -p ~/client-configs/files` en el que se almacenarán los archivos de configuración de clientes dentro del directorio **client-configs** creado anteriormente; después de eso copiaremos un archivo de configuración de cliente al directorio client-configs para usarlo como su configuración de base.



```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ mkdir -p ~/client-configs/files
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ cp /usr/share/doc/openvpn/examples/sam
ple-config-files/client.conf ~/client-configs/base.conf
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```

Luego abriremos el archivo `nano ~/client-configs/base.conf` y modificamos estas líneas de código

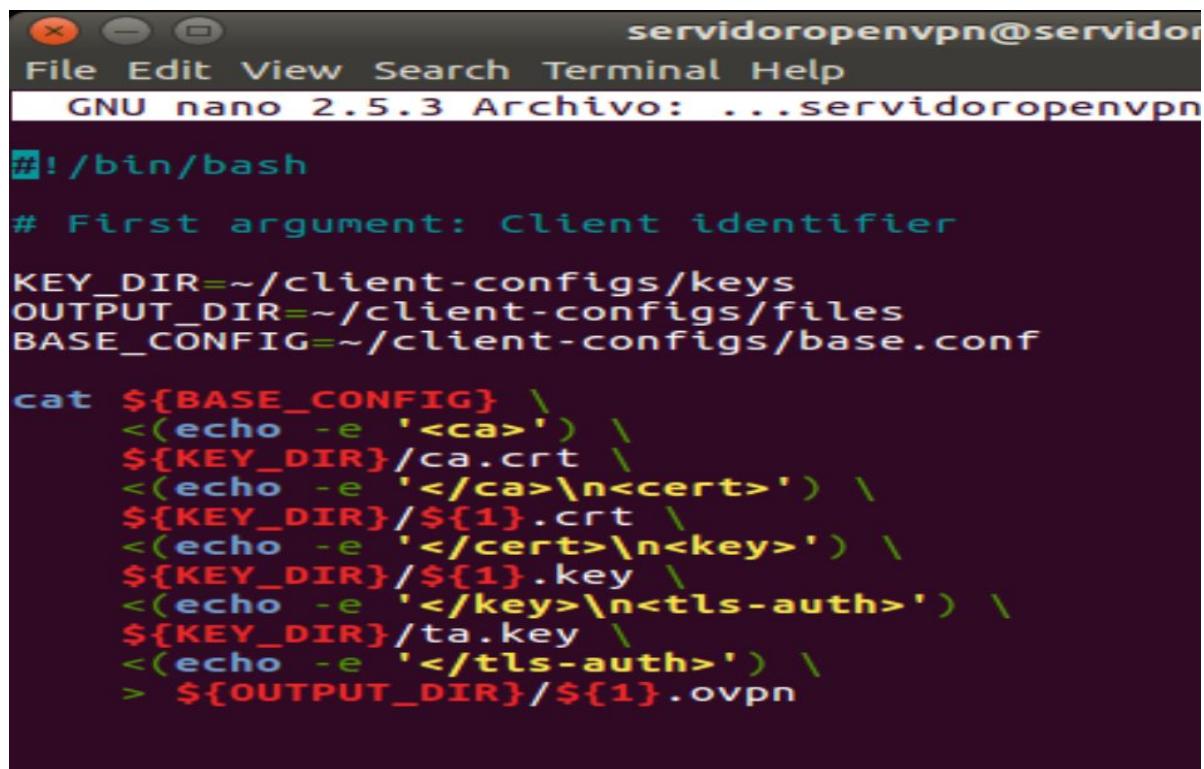


```
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 192.168.1.10 1194
;remote my-server-2 1194
```

Reflejaremos los ajustes de cipher y auth establecidos en el archivo `/etc/openvpn/server.conf`

```
# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher x
cipher AES-256-CBC
auth SHA256
```

Ahora abriremos el archivo **nano ~client-configs/make_config.sh** y colocaremos estas líneas de código ya que este archivo por defecto está vacío.



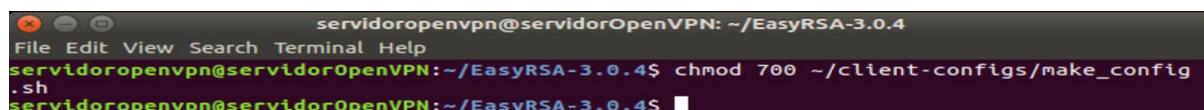
```
servidoropenvpn@servidor
File Edit View Search Terminal Help
GNU nano 2.5.3 Archivo: ...servidoropenvpn
#!/bin/bash

# First argument: Client identifier

KEY_DIR=~/client-configs/keys
OUTPUT_DIR=~/client-configs/files
BASE_CONFIG=~/client-configs/base.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>' ) \
${KEY_DIR}/${1}.key \
<<(echo -e '</key>\n<tls-auth>' ) \
${KEY_DIR}/ta.key \
<<(echo -e '</tls-auth>' ) \
> ${OUTPUT_DIR}/${1}.ovpn
```

Y le daremos los permisos correspondientes con el comando **chmod 700 ~client-configs/make_config.sh**



```
servidoropenvpn@servidorOpenVPN: ~/EasyRSA-3.0.4
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ chmod 700 ~/client-configs/make_config.sh
servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$
```

Paso 9: Generar las configuraciones de clientes

Ya que generamos los archivos del certificado y de la clave del cliente llamados **client1.crt** y **client1.key**, vamos a generar un archivo de configuración para estas credenciales si se dirige al directorio **~/client-configs** y ejecuta la secuencia de comandos que realizó al final del paso anterior .

```

servidoropenvpn@servidorOpenVPN:~/EasyRSA-3.0.4$ cd ~/client-configs
servidoropenvpn@servidorOpenVPN:~/client-configs$ 

```

```

servidoropenvpn@servidorOpenVPN:~/client-configs
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/client-configs$ sudo ./make_config.sh client1
[sudo] password for servidoropenvpn:
servidoropenvpn@servidorOpenVPN:~/client-configs$ 

```

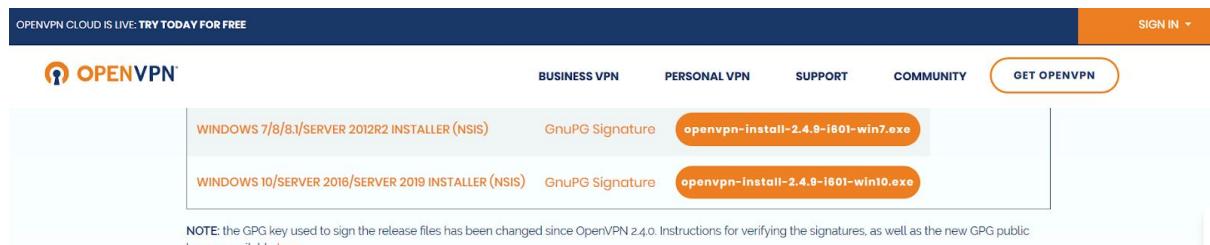
Con esto se creará un archivo llamado **client1.ovpn** en el directorio **~/client-configs/files**

```

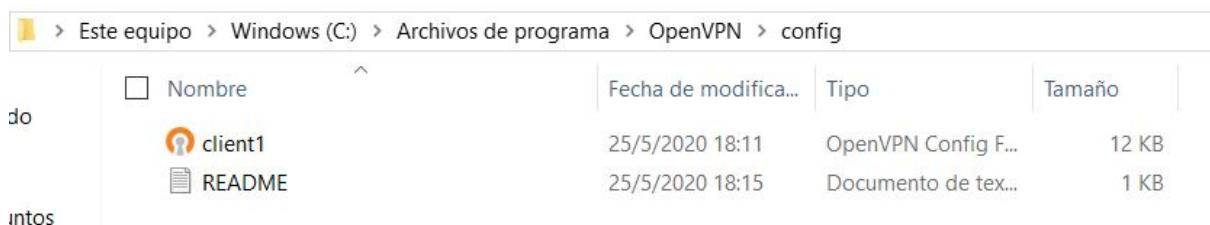
servidoropenvpn@servidorOpenVPN:~/client-configs
File Edit View Search Terminal Help
servidoropenvpn@servidorOpenVPN:~/client-configs$ ls ~/client-configs/files
client1.ovpn
servidoropenvpn@servidorOpenVPN:~/client-configs$ 

```

Ahora vamos a instalar OpenVPN en nuestra máquina que servirá como cliente, nos vamos a la página oficial de openVPN y descargamos la versión que mejor nos convenga. Mientras tanto guardamos el archivo **client1.ovpn** en nuestro drive para poderlo descargar en nuestro cliente.



Una vez descargado nuestro archivo nos vamos a descargas y lo copiamos luego nos vamos a ir a **Este equipo** al disco de **windows(C:)** después a **Archivos de programas** luego a **OpenVPN** y a **Config** una vez ahí pegamos el archivo **client1.ovpn**.



Una vez hecho esto ejecutamos OpenVPN como administrador y lo conectamos con el cliente y nos mostrara esto sin ningún error si se logró conectar con nuestro

cliente al final les aparecera que se conectó y la dirección ip de nuestro servidor VPN

```

Mon May 25 18:35:22 2020 OpenVPN 2.4.9 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [PKCS11] [AEAD] built on Apr 16 2020
Mon May 25 18:35:22 2020 Windows version 6.2 (Windows 8 or greater) 64bit
Mon May 25 18:35:22 2020 library versions: OpenSSL 1.1.1f  31 Mar 2020, LZO 2.10
Enter Management Password:
Mon May 25 18:35:22 2020 MANAGEMENT: TCP Socket listening on [AF_INET]127.0.0.1:25340
Mon May 25 18:35:22 2020 Need hold release from management interface, waiting...
Mon May 25 18:35:23 2020 MANAGEMENT: Client connected from [AF_INET]127.0.0.1:25340
Mon May 25 18:35:23 2020 MANAGEMENT: CMD 'state on'
Mon May 25 18:35:23 2020 MANAGEMENT: CMD 'log all on'
Mon May 25 18:35:23 2020 MANAGEMENT: CMD 'echo all on'
Mon May 25 18:35:23 2020 MANAGEMENT: CMD 'bytecount 5'
Mon May 25 18:35:23 2020 MANAGEMENT: CMD 'hold off'
Mon May 25 18:35:23 2020 MANAGEMENT: CMD 'hold release'
Mon May 25 18:35:23 2020 Outgoing Control Channel Authentication: Using 256 bit message hash 'SHA256' for HMAC authentication
Mon May 25 18:35:23 2020 Incoming Control Channel Authentication: Using 256 bit message hash 'SHA256' for HMAC authentication
Mon May 25 18:35:23 2020 TCP/UDP: Preserving recently used remote address: [AF_INET]192.168.1.10:1194
Mon May 25 18:35:23 2020 Socket Buffers: R=[65536->65536] S=[65536->65536]
Mon May 25 18:35:23 2020 UDP link remote: [AF_INET]192.168.1.10:1194
Mon May 25 18:35:23 2020 MANAGEMENT: >STATE:1590453323,WAIT,,,
Mon May 25 18:35:23 2020 MANAGEMENT: >STATE:1590453323,AUTH,,,
Mon May 25 18:35:23 2020 TLS: Initial packet from [AF_INET]192.168.1.10:1194, sid=34797dba 608a2e0f
Mon May 25 18:35:23 2020 VERIFY OK: depth=1, CN=Easy-RSA CA
Mon May 25 18:35:23 2020 VERIFY KU OK
Mon May 25 18:35:23 2020 Validating certificate extended key usage
Mon May 25 18:35:23 2020 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
Mon May 25 18:35:23 2020 VERIFY EKU OK
Mon May 25 18:35:23 2020 VERIFY OK: depth=0, CN=server
Mon May 25 18:35:23 2020 Control Channel: TLSv1.2, cipher TLSv1.2 DHE-RSA-AES256-GCM-SHA384, 2048 bit RSA
Mon May 25 18:35:23 2020 [server] Peer Connection Initiated with [AF_INET]192.168.1.10:1194
Mon May 25 18:35:24 2020 MANAGEMENT: >STATE:1590453324,GET_CONFIG,,,
Mon May 25 18:35:24 2020 SENT CONTROL [server]: "PUSH_REQUEST" (status=)
Mon May 25 18:35:24 2020 PUSH: Received control message: "PUSH_REPLY,redirect-gateway def1 bypass-dhcp,dhcp-option DNS 208.67.222.222,dhcp-option DNS 208.67.220.220,route 10.8.0.1,topology
Mon May 25 18:35:24 2020 OPTIONS IMPORT: timers and/or timeouts modified
Mon May 25 18:35:24 2020 OPTIONS IMPORT: --ifconfig/up options modified
Mon May 25 18:35:24 2020 OPTIONS IMPORT: route options modified
Mon May 25 18:35:24 2020 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified
Mon May 25 18:35:24 2020 Outgoing Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
Mon May 25 18:35:24 2020 Outgoing Data Channel: Using 256 bit message hash 'SHA256' for HMAC authentication
Mon May 25 18:35:24 2020 Incoming Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
Mon May 25 18:35:24 2020 Incoming Data Channel: Using 256 bit message hash 'SHA256' for HMAC authentication
Mon May 25 18:35:24 2020 interactive service msg_channel=0

Mon May 25 18:35:24 2020 ROUTE GATEWAY 192.168.1.1/255.255.255.0 I=5 HWADDR=18:56:80:52:ba:75
Mon May 25 18:35:24 2020 open_tun
Mon May 25 18:35:24 2020 TAP-WIN32 device [Local Area Connection] opened: \\.\Global\{5ED6EBF4-F884-4CDB-BEA7-8F696676B104}.tap
Mon May 25 18:35:24 2020 TAP-Windows Driver Version 9.24
Mon May 25 18:35:24 2020 Notified TAP-Windows driver to set a DHCP IP/netmask of 10.8.0.6/255.255.255.252 on interface {5ED6EBF4-F884-4CDB-BEA7-8F696676B104} [DHCP-serv: 10.8.0.5, lease-tim
Mon May 25 18:35:24 2020 Successful ARP flush on interface [401] {5ED6EBF4-F884-4CDB-BEA7-8F696676B104}
Mon May 25 18:35:24 2020 MANAGEMENT: >STATE:1590453324,ASSIGN_IP,,10.8.0.6,,,
Mon May 25 18:35:29 2020 TEST ROUTES: 2/2 succeeded len=1 ret=1 a=0 u/d=up
Mon May 25 18:35:29 2020 C:\windows\system32\route.exe ADD 192.168.1.11 MASK 255.255.255.255 192.168.1.1 IF 5
Mon May 25 18:35:29 2020 ROUTE: CreateIphForwardEntry succeeded with dwForwardMetric1=50 and dwForwardType=4
Mon May 25 18:35:29 2020 Route addition via IPAPI succeeded [adaptive]
Mon May 25 18:35:29 2020 C:\windows\system32\route.exe ADD 0.0.0.0 MASK 128.0.0.0 10.8.0.5
Mon May 25 18:35:29 2020 ROUTE: CreateIphForwardEntry succeeded with dwForwardMetric1=25 and dwForwardType=4
Mon May 25 18:35:29 2020 Route addition via IPAPI succeeded [adaptive]
Mon May 25 18:35:29 2020 C:\windows\system32\route.exe ADD 128.0.0.0 MASK 128.0.0.0 10.8.0.5
Mon May 25 18:35:29 2020 ROUTE: CreateIphForwardEntry succeeded with dwForwardMetric1=25 and dwForwardType=4
Mon May 25 18:35:29 2020 Route addition via IPAPI succeeded [adaptive]
Mon May 25 18:35:29 2020 MANAGEMENT: >STATE:1590453329,ADD_ROUTES,,,
Mon May 25 18:35:29 2020 C:\windows\system32\route.exe ADD 10.8.0.1 MASK 255.255.255.255 10.8.0.5
Mon May 25 18:35:29 2020 ROUTE: CreateIphForwardEntry succeeded with dwForwardMetric1=25 and dwForwardType=4
Mon May 25 18:35:29 2020 Route addition via IPAPI succeeded [adaptive]
Mon May 25 18:35:29 2020 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
Mon May 25 18:35:29 2020 Initialization Sequence Completed
Mon May 25 18:35:29 2020 MANAGEMENT: >STATE:1590453329,CONNECTED,SUCCESS,10.8.0.6,192.168.1.10,1194,,
```

Para comprobar los saltos que dio podemos poner en el navegador de nuestro cliente <https://www.dnsleaktest.com/> y nos saldra esto



IP	Hostname	ISP	Country
146.112.137.64	r1.compute.atl1.edc.strin.net.	Cisco OpenDNS, LLC	Atlanta, United States 
146.112.137.65	r2.compute.atl1.edc.strin.net.	Cisco OpenDNS, LLC	Atlanta, United States 
146.112.137.66	r3.compute.atl1.edc.strin.net.	Cisco OpenDNS, LLC	Atlanta, United States 
146.112.137.67	r4.compute.atl1.edc.strin.net.	Cisco OpenDNS, LLC	Atlanta, United States 
146.112.137.68	r5.compute.atl1.edc.strin.net.	Cisco OpenDNS, LLC	Atlanta, United States 
146.112.137.69	r6.compute.atl1.edc.strin.net.	Cisco OpenDNS, LLC	Atlanta, United States 
146.112.137.70	r7.compute.atl1.edc.strin.net.	Cisco OpenDNS, LLC	Atlanta, United States 
146.112.137.71	r8.compute.atl1.edc.strin.net.	Cisco OpenDNS, LLC	Atlanta, United States 
146.112.137.72	r9.compute.atl1.edc.strin.net.	Cisco OpenDNS, LLC	Atlanta, United States 
146.112.137.73	r10.compute.atl1.edc.strin.net.	Cisco OpenDNS, LLC	Atlanta, United States 
201.247.149.27	None	Claro El Salvador	Soyapango, El Salvador 
216.230.139.4	4.139.intelnet.net.gt.	Telgua	Escuintla, Guatemala 