killian goncalves & cristopher boni fuentes

AP 11 : SSH, VPN et Fail2ban

Lien vers le GANTT :

Lien

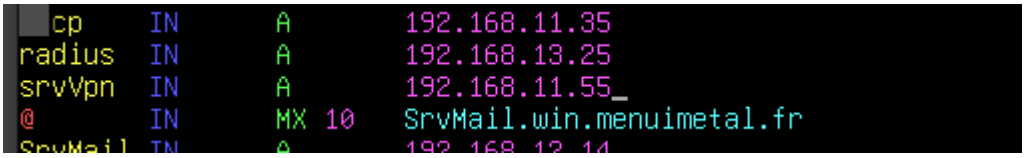
Lien vers le schéma réseau :

lien

PRÉREQUIS

référencement Dns

Le référencement du srvVpn se passe dans le fichier db.menuimetal.fr dans le srv DNS et puis ajouter la ligne de commande srvVpn IN A ip du serveur.



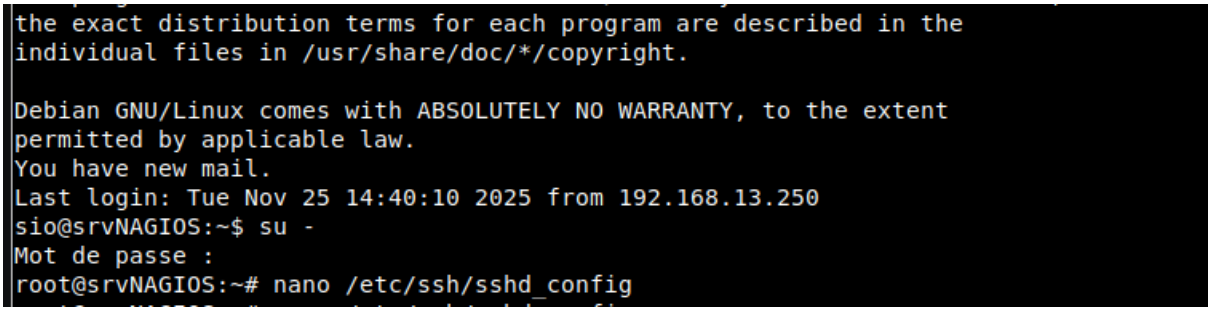
Ensuite un systemctl restart bind9

On aura ce beau résultat avec un nslookup

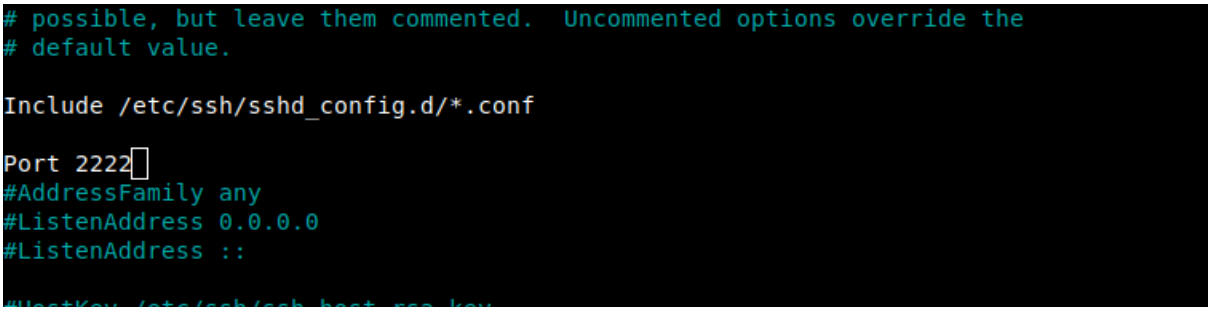
Partie 1 SSH

srvNAGIOS :

On va dans le fichier configuration sshd pour changer le port 22 mis par défaut

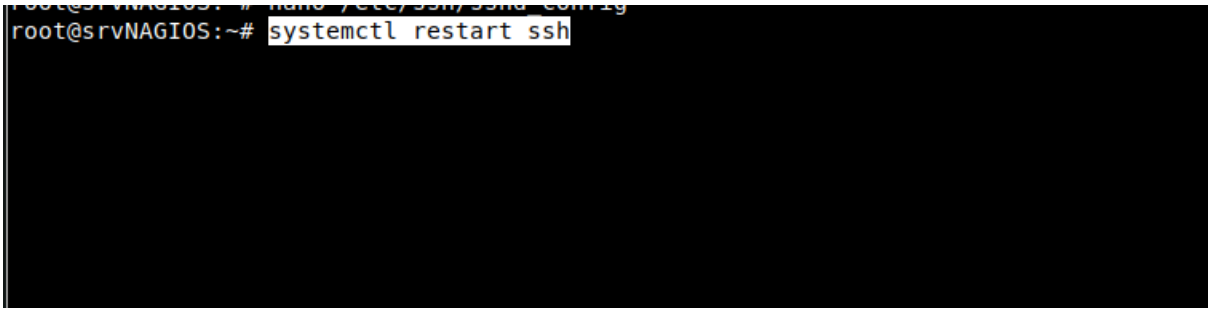


Puis on va modifier le port 22 en port 2222

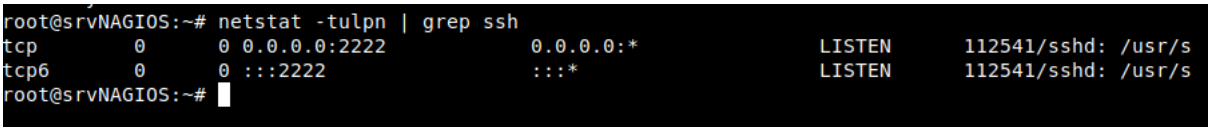


on décommente et on mais yes

Ensuite on va un restart ssh pour que les modification soit pris en compte



on vérifie grace a la commande nestat -tulpn | grep ssh si le changement de port a bien etait effectué



La configuration de l'accès grâce à une clef public pour l'utilisateur sio sur le serveur srvNAGIOS qui est finalisée sur le répertoire de configuration **.ssh** qui est créé et ses droits d'accès ont été définis à 700 sur le fichier **authorized_keys** est aussi et ses droits fixés à 600. La clé publique **tmp_id_rsa.pub** a été ajoutée au fichier **authorized_keys** qui autorise ainsi la connexion et j'ai supprime une autre cléé **tmp_id_rsa.pub** (qui etait problematique) et l'état final des droits sur le fichier **authorized_keys** a été vérifié.

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Tue Nov 25 17:08:33 2025 from 192.168.13.16
sio@srvNAGIOS:~$ ls -l ~/.ssh/
total 4
-rw-r--r-- 1 sio sio 573 25 nov.  17:10 tmp_id_rsa.pub
sio@srvNAGIOS:~$ cat ~/.ssh/tmp_id_rsa.pub >> ~/.ssh/authorized_keys
sio@srvNAGIOS:~$ chmod 600 ~/.ssh/authorized_keys
sio@srvNAGIOS:~$ rm ~/.ssh/tmp_id_rsa.pub
sio@srvNAGIOS:~$ ls -l ~/.ssh/
total 4
-rw------- 1 sio sio 573 25 nov.  17:11 authorized_keys
sio@srvNAGIOS:~$
```

```
sio@srvNAGIOS:~$ mkdir -p ~/.ssh
sio@srvNAGIOS:~$ chmod 700 ~/.ssh
sio@srvNAGIOS:~$ touch ~/.ssh/authorized_keys
sio@srvNAGIOS:~$ chmod 600 ~/.ssh/authorized_keys
```

srvGLPI :

changement de port

```
Include /etc/ssh/sshd_config.d/*.conf

Port 2222
#AddressFamily any
#ListenAddress 0.0.0.0
```

j'ai relancé le service ssh

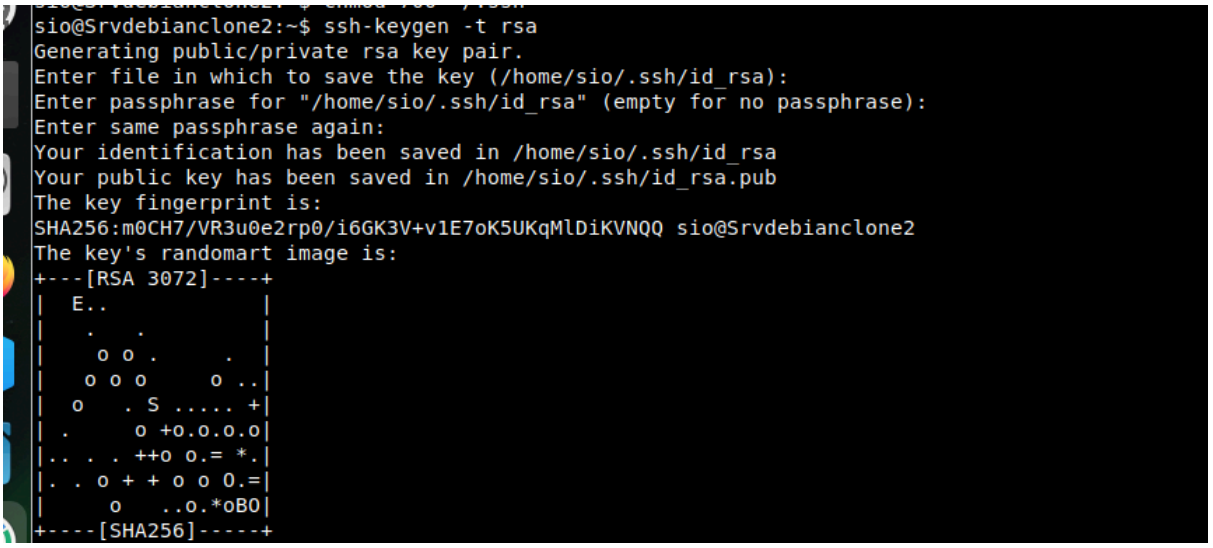
```
root@glpi:~# systemctl restart ssh
root@glpi:~#
```

et j'ai verifier si tout était bien modifié (est même procédé que nagios)

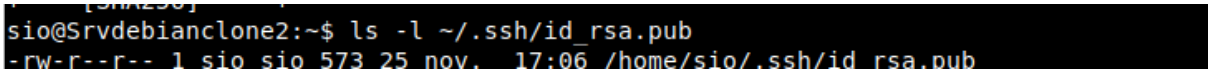
```
root@glpi:~# netstat -tulpn | grep ssh
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN      23596/sshd: /usr/sb
tcp6       0      0 :::2222                 :::*                    LISTEN      23596/sshd: /usr/sb
root@glpi:~#
```

srvdebianclone2 :

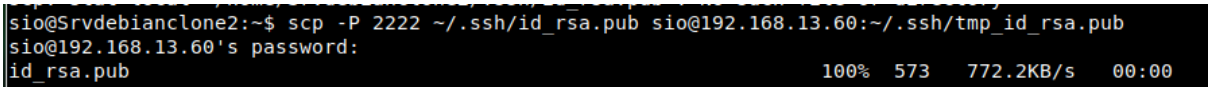
création de la cléé avec la commande ssh-keygen -trsa



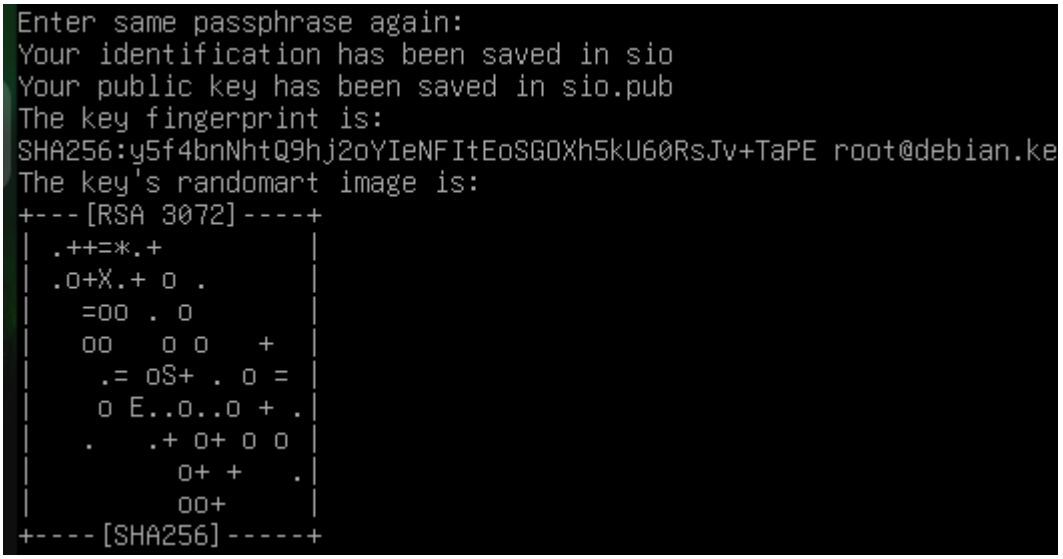
ainsi que la création du fichier **authorized_keys** avec des droits stricts fixés à **600**.



cette commande va me permettre d’envoyer ma clef a mon serveur nagios sur le fichier que j’ai créé



debian.key



```
sio@debian:~$ ssh -p 2222 sio@192.168.13.19
Enter passphrase for key '/home/sio/.ssh/id_rsa':
Linux glpi 6.12.41+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.41-1 (2025-08-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov 27 12:59:11 2025 from 192.168.13.15
sio@glpi:~$
```

Rediriger les logs d'authentification vers le serveur central de log Rsyslog dans un fichier spécifique

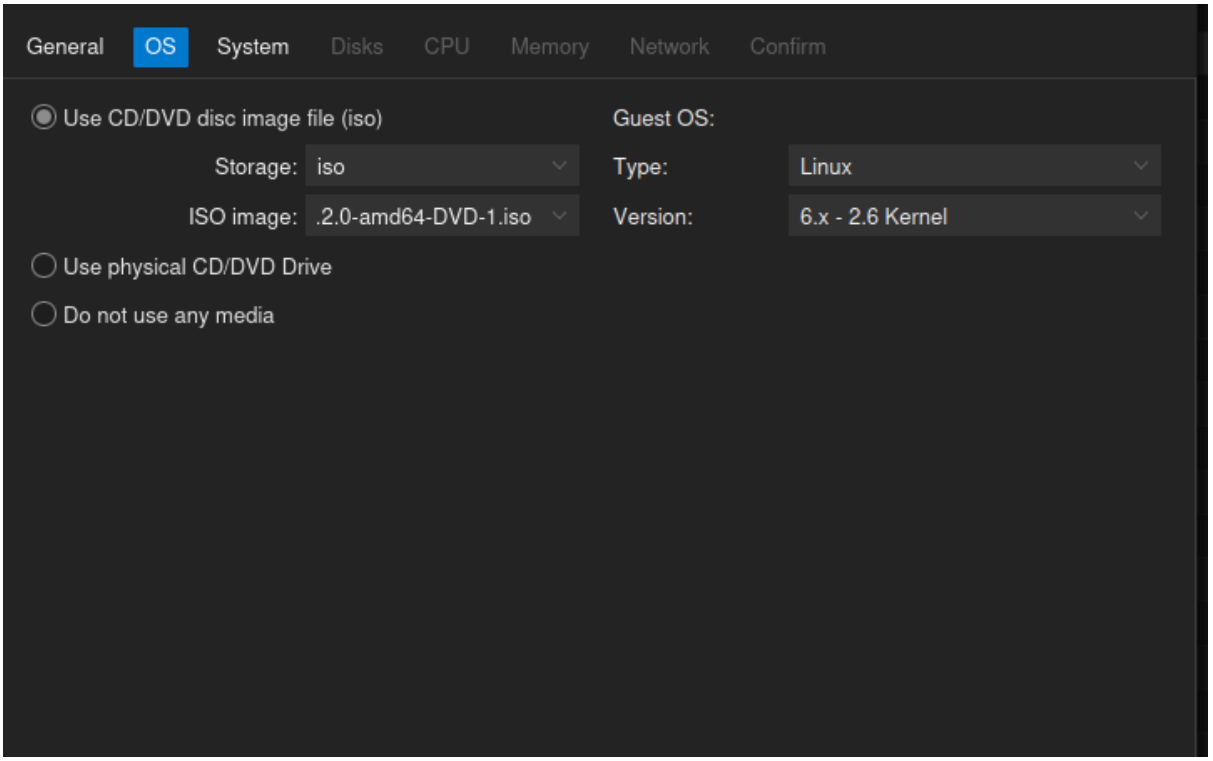
```
root@SrvRsyslog:~# nano /etc/rsyslog.conf
```

j’ai décommenter module et input

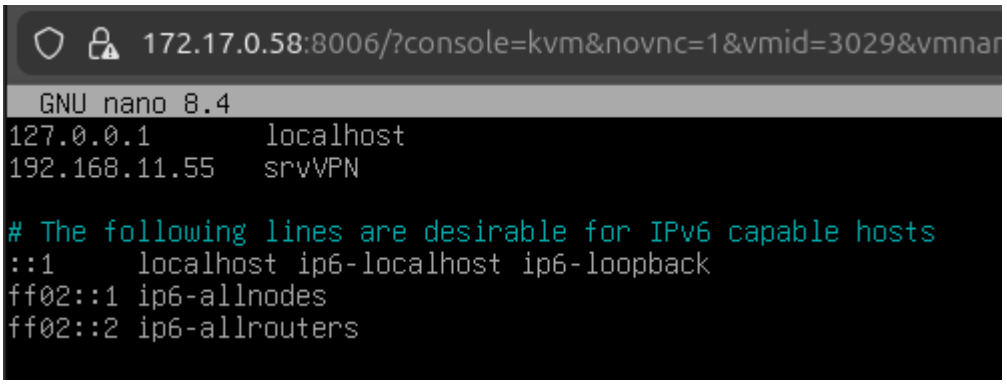
```
# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")
```

Partie 2 VPN

Création d'une machine Debian nommé SrvVpn



Modifications dans le fichier hosts pour changement noms



Puis modification du fichier interface pour mettre une ip physique et non en dhcp car le dhcp envoie automatiquement des ips dans le VLAN Lan

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

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens18
iface ens18 inet static
 address 192.168.11.55
 netmask 255.255.255.0
 broadcast 192.168.11.255
 gateway 192.168.11.254
```

Installation du serveur VPN (srvVPN)

- Installation OpenVPN & Easy-RSA

```
root@srvVpn:~#  apt install openvpn easy-rsa -y
```

Préparation de l'infrastructure PKI

cp -r /usr/share/easy-rsa/* /etc/openvpn/easy-rsa
cd /etc/openvpn/easy-rsa/
📌 `cp -r` signifie **copier récursivement** → permet de copier tout le dossier.

```
~# cp -r /usr/share/easy-rsa /etc/openvpn/
```

puis vérification du contenu :

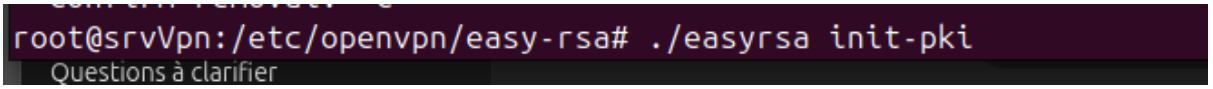
```
root@srvVpn:/etc/openvpn/easy-rsa# ls -lh
total 212K
-rwxr-xr-x 1 root root 182K 25 nov.  15:08 easyrsa
-rw-r--r-- 1 root root 5,1K 25 nov.  15:08 openssl-easyrsa.cnf
drwx------ 8 root root 4,0K 26 nov.  09:47 pki
-rw-r--r-- 1 root root 8,9K 25 nov.  15:08 vars.example
drwxr-xr-x 2 root root 4,0K 25 nov.  15:08 x509-types
root@srvVpn:/etc/openvpn/easy-rsa#
```

Initialisation PKI + génération CA

Déplacement dans le répertoire :

```
root@srvVpn:~# cd /etc/openvpn/easy-rsa/
```

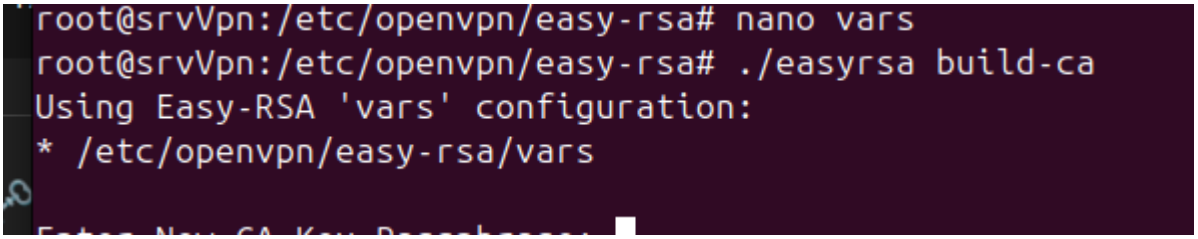
Crée l'architecture cryptographique PKI (Public Key Infrastructure) → dossier `pki/`
`./easyrsa init-pki`



`Génération Autorité Certificatrice (CA)`
avec la commande ./easyrsa build-ca crée les fichiers :

✔ `pki/ca.crt` → certificat racine (public)
✔ `pki/private/ca.key` → clé maître du VPN (NE JAMAIS DIFFUSER)

💀 *Si cette clé fuite → le VPN est compromi !*



📍 3) Certificat serveur OpenVPN

Avec la commande ./easyrsa gen-req SrvVPN → crée une clé privée + un fichier CSR (`SrvVPN.req`)

```
root@srvVpn:/etc/openvpn/easy-rsa# ./easyrsa sign-req server SrvVPN
```

puis on ajoute les informations demandées

```
....+....+...+...+.........+..............+.....+.+..+.+.....+....+........+..
......+.........+.+.........+..+....+.....+.+.........+......+......+.........
..+.....+........................+.......+...++++++
-----
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.
-----
Country Name (2 letter code) [US]:fr
State or Province Name (full name) [California]:seine et marne
Locality Name (eg, city) [San Francisco]:melun
Organization Name (eg, company) [Copyleft Certificate Co]:menuimetal.fr
Organizational Unit Name (eg, section) [My Organizational Unit]:
Common Name (eg: your user, host, or server name) [SrvVPN]:SrvVpn
Email Address [me@example.net]:Srvvpn@menuimetal.fr
```

Signature du certificat serveur :
📌 On valide que ce certificat pourra agir comme **serveur VPN**.
puis on appuis sur yes quand sa demande de confirmé

```
root@srvVpn:/etc/openvpn/easy-rsa# ./easyrsa sign-req server SrvVPN
Using Easy-RSA 'vars' configuration:
* /etc/openvpn/easy-rsa/vars
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.
You are about to sign the following certificate:

  Requested CN:     'SrvVpn'
  Requested type:   'server'
  Valid for:        '365' days


subject=
    countryName               = fr
    stateOrProvinceName       = seine et marne
    localityName              = melun
    organizationName          = menuimetal.fr
    organizationalUnitName    = My Organizational Unit
    commonName                = SrvVpn
    emailAddress              = Srvvpn@menuimetal.fr

Type the word 'yes' to continue, or any other input to abort.
  Confirm requested details:
```

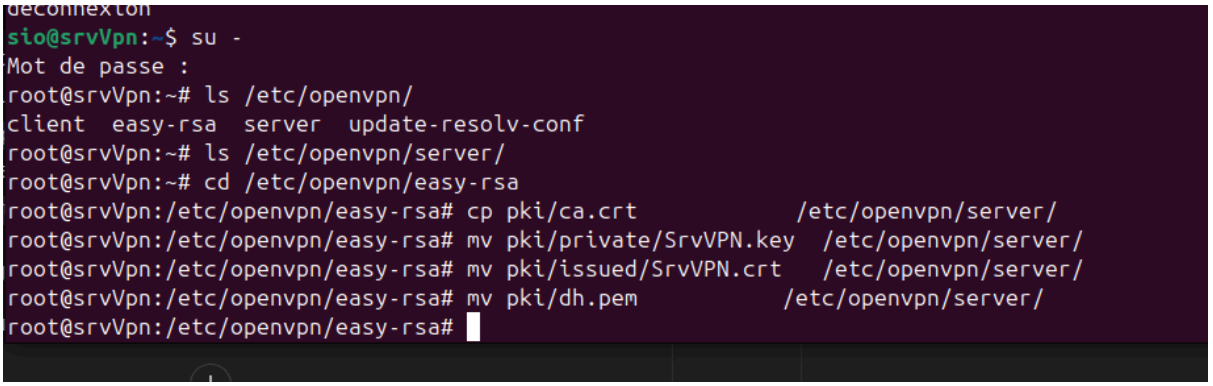
Résultat:

```
-----
Country Name (2 letter code) [US]:fr
State or Province Name (full name) [California]:seine et marne
Locality Name (eg, city) [San Francisco]:melun
Organization Name (eg, company) [Copyleft Certificate Co]:menuimetal.fr
Organizational Unit Name (eg, section) [My Organizational Unit]:
Common Name (eg: your user, host, or server name) [SrvVPN]:SrvVpn
Email Address [me@example.net]:Srvvpn@menuimetal.fr
Serial-number (eg, device serial-number) []:

Notice
------
Private-Key and Public-Certificate-Request files created.
Your files are:
* req: /etc/openvpn/easy-rsa/pki/reqs/SrvVPN.req
* key: /etc/openvpn/easy-rsa/pki/private/SrvVPN.key

root@srvVpn:/etc/openvpn/easy-rsa#
```

Organisation fichiers :

`cp ... SrvVPN.crt` : **Copie** le certificat public (l'identité) du serveur VPN vers le dossier de configuration `/etc/openvpn/server/`.

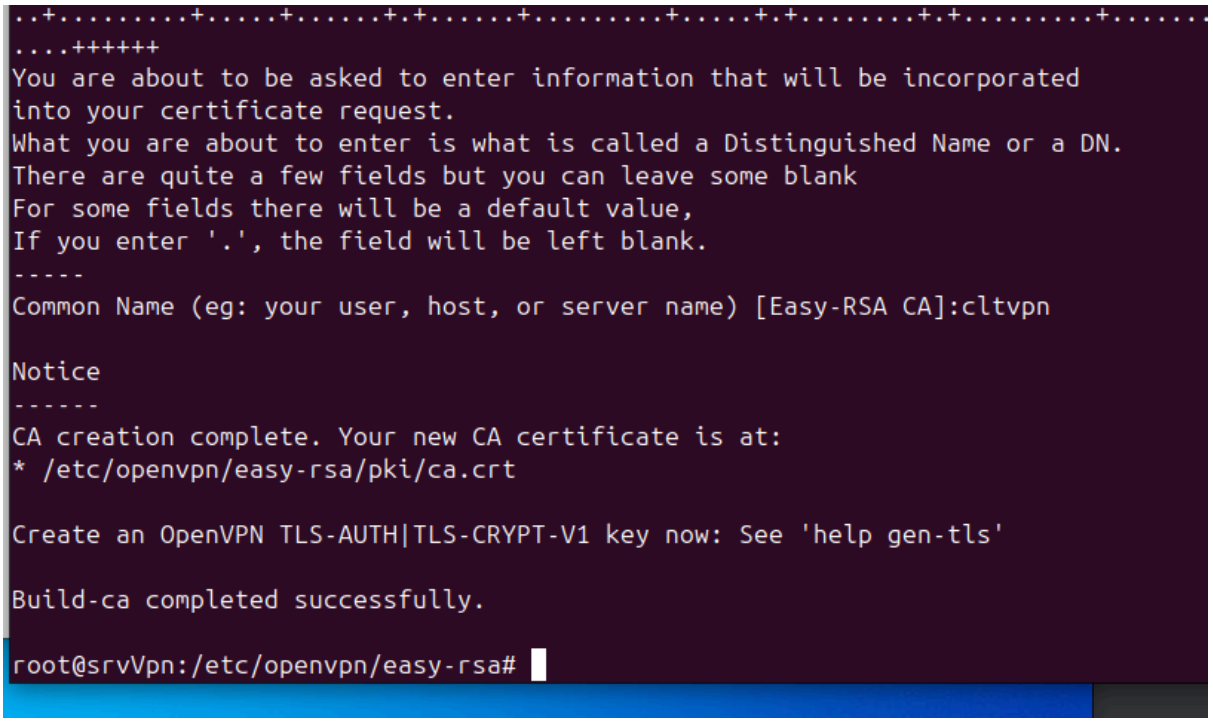
`mv ... SrvVPN.key` : **Déplace** la clé privée (fichier très sensible) vers ce même dossier ; l'original est supprimé pour éviter les doublons non sécurisés.
`cp ... ca.crt` : **Copie** le certificat de l'Autorité de Certification (CA) nécessaire pour valider la chaîne de confiance des clients



Génération certificat Client (CltVPN)

📌 Celui-ci aura le droit de se connecter mais pas d'émettre des certificats.

```
./easyrsa gen-req CltVPN
```



./easyrsa sign-req client CltVPN

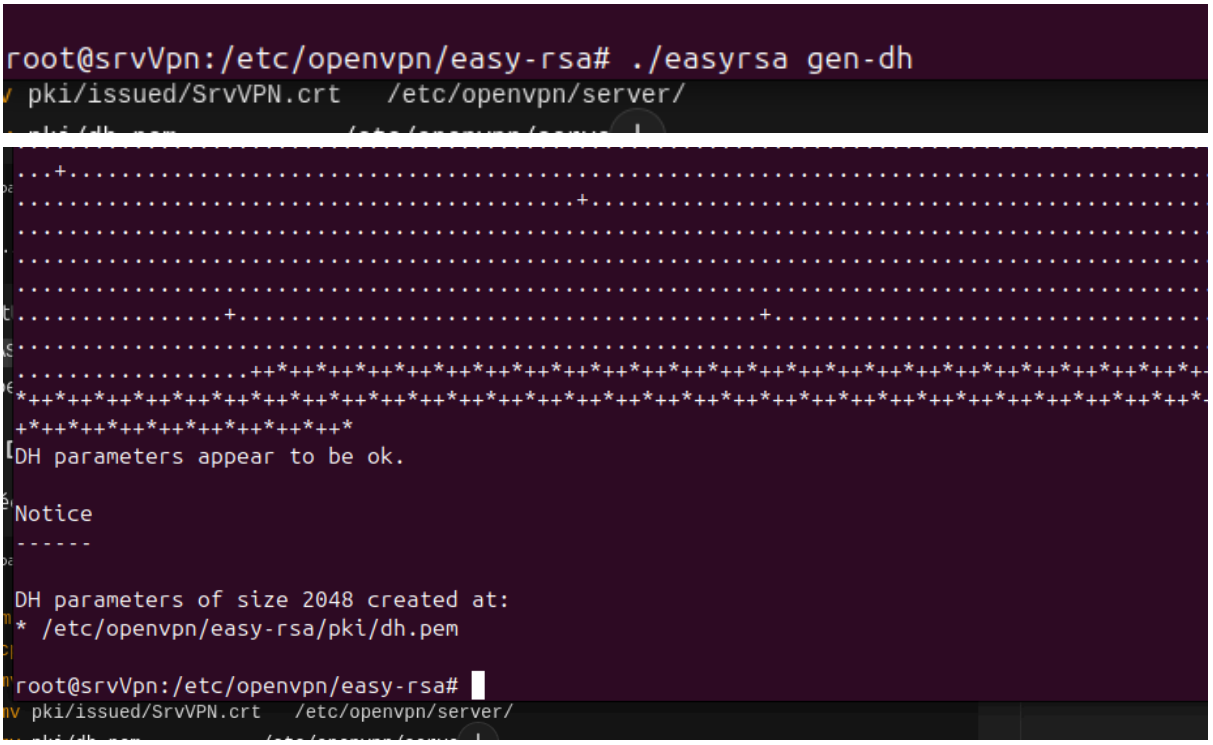
```
root@srvVpn:/etc/openvpn/easy-rsa# ./easyrsa sign-req server SrvVPN
Please check over the details shown below for accuracy. Note that this reque
has not been cryptographically verified. Please be sure it came from a truste
source or that you have verified the request checksum with the sender.
You are about to sign the following certificate:

  Requested CN:     'srvVpn'
  Requested type:   'server'
  Valid for:        '825' days


subject=
    commonName                = srvVpn

Type the word 'yes' to continue, or any other input to abort.
  Confirm requested details:
```

puis un ajoute Yes

```
root@srvVpn:/etc/openvpn/easy-rsa# ./easyrsa sign-req client cltVpn
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.
You are about to sign the following certificate:

  Requested CN:     'cltvpn'
  Requested type:   'client'
  Valid for:        '825' days


subject=
    commonName                = cltvpn

Type the word 'yes' to continue, or any other input to abort.
  Confirm requested details:
```

et pui

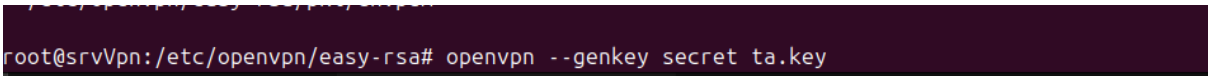
Lance la génération DH (utilise la taille défini dans `vars`, ici 2048) : crée `pki/dh.pem` (paramètres DH nécessaires au handshake pour l'échange de clés).



🔥 *Génération clé TLS HMAC*

Cette Clé supplémentaire empêchant les attaques « bruteforce » sur le port 1194.

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



📍 *Configuration serveur VPN*

Création du fichier server.conf dans etc/openvpn/

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

voici le contenu:

mode server
tls-server
local 192.168.11.55
port 1194

```
proto udp
dev tun

ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/SrvVPN.crt
key /etc/openvpn/server/SrvVPN.key
dh /etc/openvpn/server/dh.pem

server 10.8.0.0 255.255.255.0

push "route 10.8.0.1 255.255.255.255"
push "dhcp-option DNS 192.168.12.1"
push "redirect-gateway def1"

keepalive 10 120
persist-key
persist-tun

client-to-client
max-clients 10

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

chroot /etc/openvpn/jail

log /var/log/openvpn/openvpn.log
status /var/log/openvpn/openvpn-status.log
verb 3
mute 20

tls-version-min 1.2
cipher AES-256-GCM
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384
auth SHA256
```

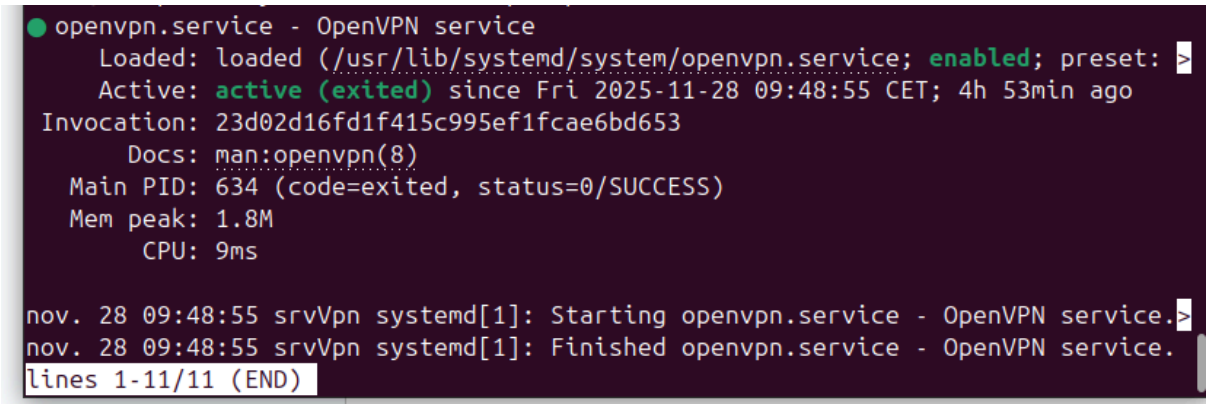
Explication :

1. Ce fichier est le **cœur** de la configuration et utilise le mode `server` avec l'écoute sur l'IP `192.168.11.55`, port `1194` UDP.
2. Il crée une interface virtuelle (`dev tun`) et définit le réseau VPN interne pour les clients, ici le sous-réseau `10.8.0.0/24`.
3. La partie **cryptographie** est cruciale : on spécifie les chemins vers les fichiers (`ca`, `cert`, `key`) qui servent à identifier le serveur et à chiffrer les données.
4. Le serveur est configuré pour **pousser automatiquement** des paramètres aux clients qui se connectent, c'est ce qu'on appelle les directives `push`.

5. L'une de ces directives force le client à utiliser le VPN comme **passerelle par défaut** (`redirect-gateway def1`), pour que tout son trafic soit chiffré.
6. Il impose également un **serveur DNS** spécifique (`192.168.12.1`) pour les requêtes des clients, ce qui est typique dans un réseau d'entreprise.
7. Pour la sécurité, on utilise une clé supplémentaire avec **`tls-auth`** pour bloquer les tentatives d'attaques non-authentifiées.
8. On s'assure que le chiffrement est fort en forçant le protocole **`AES-256-GCM`** et la version TLS 1.2 minimum.
9. La commande **`chroot`** est un mécanisme de sécurité qui isole le processus OpenVPN dans un dossier restreint (une "prison virtuelle").
10. Enfin, `keepalive` et les directives de `log` garantissent que le service est stable et que l'on peut facilement le surveiller et le déboguer.

Activation du service :

systemctl enable openvpn-server@server.service
systemctl start openvpn-server@server.service
systemctl status openvpn-server@server.service



Commande	Rôle
`enable`	démarre au boot
`start`	lance immédiatement
`status`	affiche si le service fonctionne

Puis avec un la commande tail -f /var/log/openvpn/openvpn.log
a doit

Routage et translation NAT

Activer routage IP permanent :
dans le fichier sysctl.conf dans /etc/sysctl.d on active le routage ip permanent

```
root@srvVpn:~# nano /etc/sysctl.d/sysctl.conf
```

vérification:

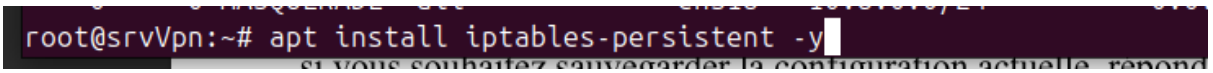
```
root@srvVpn:~# cat /etc/sysctl.d/sysctl.conf
net.ipv4.ip_forward=1
```

Vérification

Puis avec un la commande tail -f /var/log/openvpn/openvpn.log montre les logs et si y a la phrase “Initialization sequence completed” sa veut dire que les configuration sont bonnes et le openvpn est fonctionnel

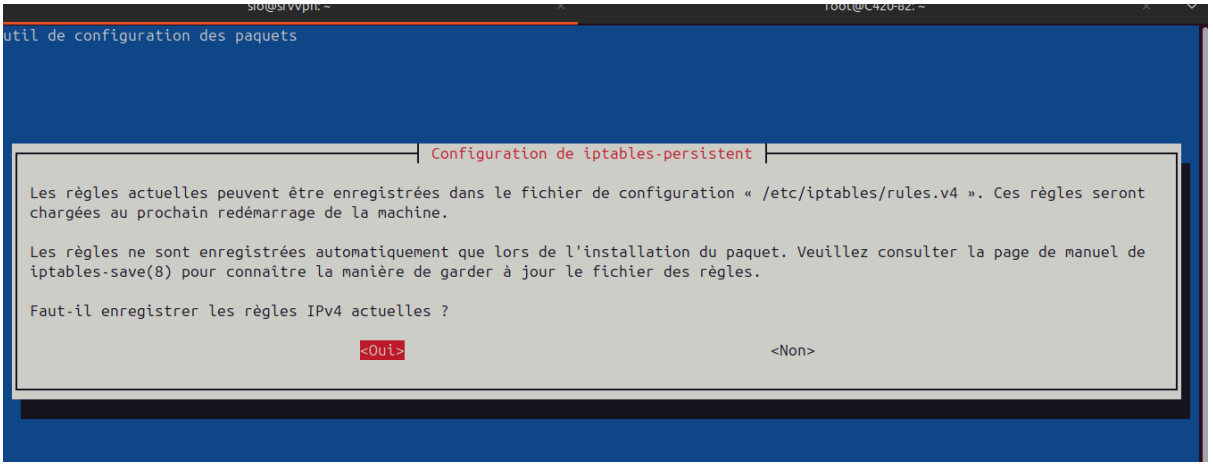
```
~
lines 1785-1808/1808 (END)
root@srvVpn:~# nano /etc/openvpn/server.conf
root@srvVpn:~# nano /etc/openvpn/server.conf
root@srvVpn:~# systemctl restart openvpn
root@srvVpn:~# tail -f /var/log/openvpn/openvpn.log
2025-11-25 15:20:05 net_addr_ptp_v4_add: 10.8.0.1 peer 10.8.0.2 dev tun0
2025-11-25 15:20:05 net_route_v4_add: 10.8.0.0/24 via 10.8.0.2 dev [NULL] table 0 metric -1
2025-11-25 15:20:05 Could not determine IPv4/IPv6 protocol. Using AF_INET
2025-11-25 15:20:05 Socket Buffers: R=[212992->212992] S=[212992->212992]
2025-11-25 15:20:05 UDPv4 link local (bound): [AF_INET]192.168.11.55:1194
2025-11-25 15:20:05 UDPv4 link remote: [AF_UNSPEC]
2025-11-25 14:20:05 chroot to '/etc/openvpn/jail' and cd to '/' succeeded
2025-11-25 14:20:05 MULTI: multi_init called, r=256 v=256
2025-11-25 14:20:05 IFCONFIG POOL IPv4: base=10.8.0.4 size=62
2025-11-25 14:20:05 Initialization Sequence Completed
```

📌 3) Installation de **iptables-persistent**

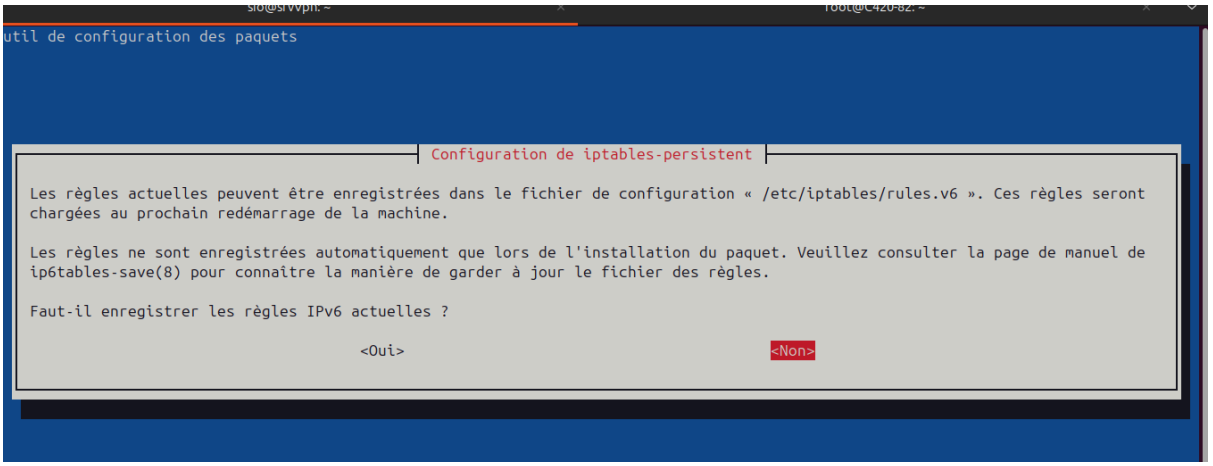


L'installation demande :

🔶 ***Save current IPv4 rules ?*** → **YES**



🔶 ***Save IPv6 rules ?*** → **NON pas obligatoire**

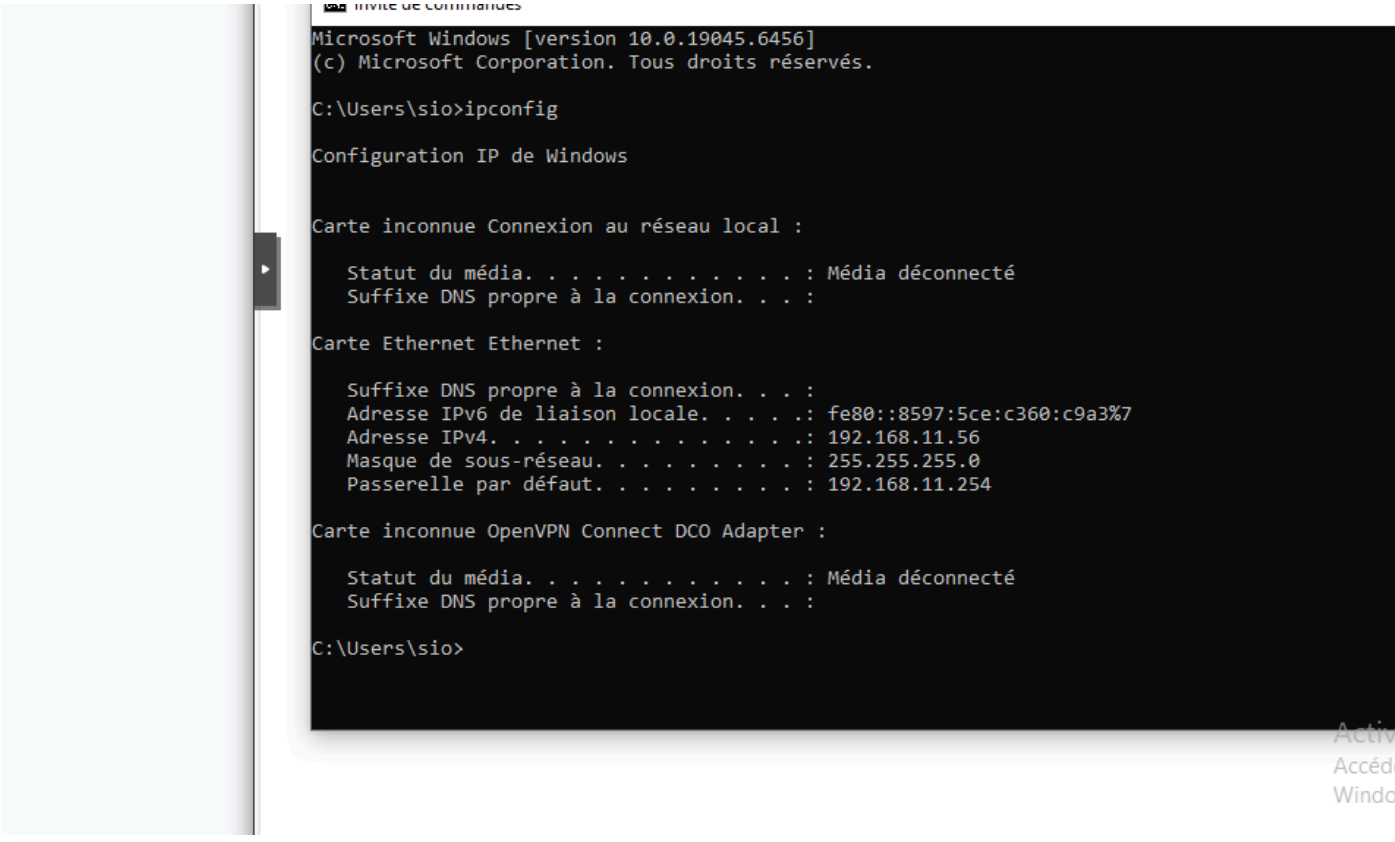


Ensuite on exécute la commande iptables -t nat -L -n -V pour vérifier on constate l'ip de routage 10.08.0.0 de l'interface ens18

```
root@srvVpn:~# iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o ens18 -j MASQUERADE
root@srvVpn:~# iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 1 packets, 112 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MASQUERADE  all  --  *      ens18   10.8.0.0/24          0.0.0.0/0
root@srvVpn:~#
```

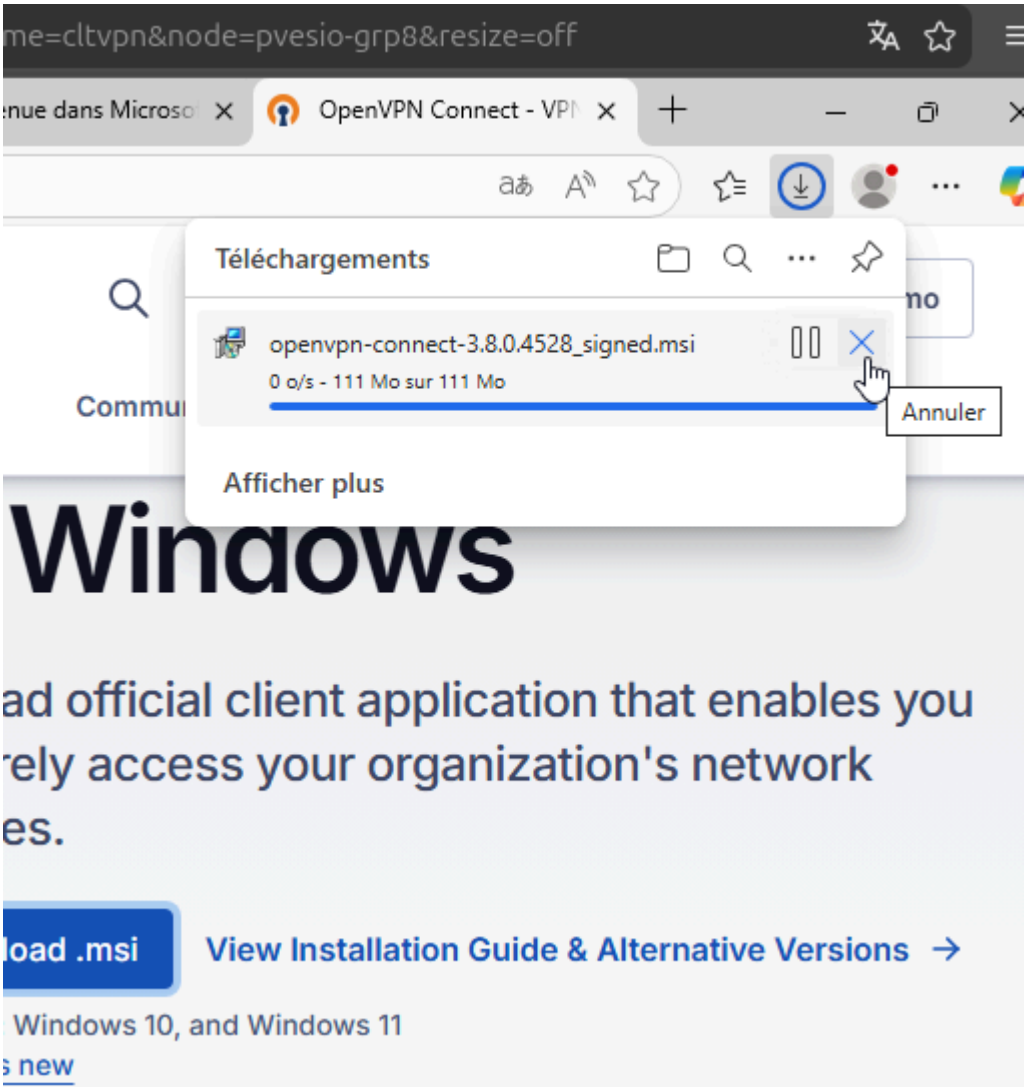
Installation du logiciel client OpenVPN sur une client Windows

On vérifie l'ip du client windows 192.168.11.56 avec un ipconfig dans un cmd

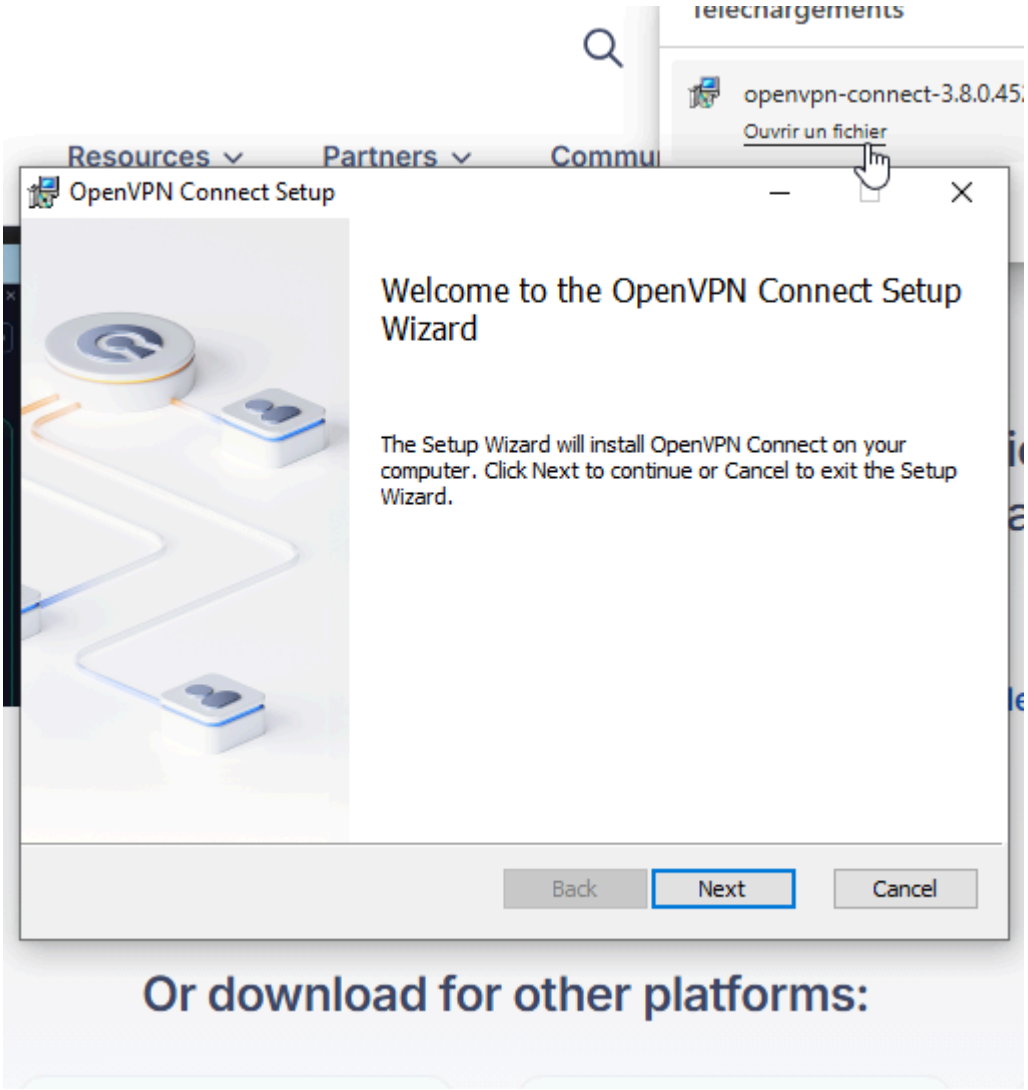


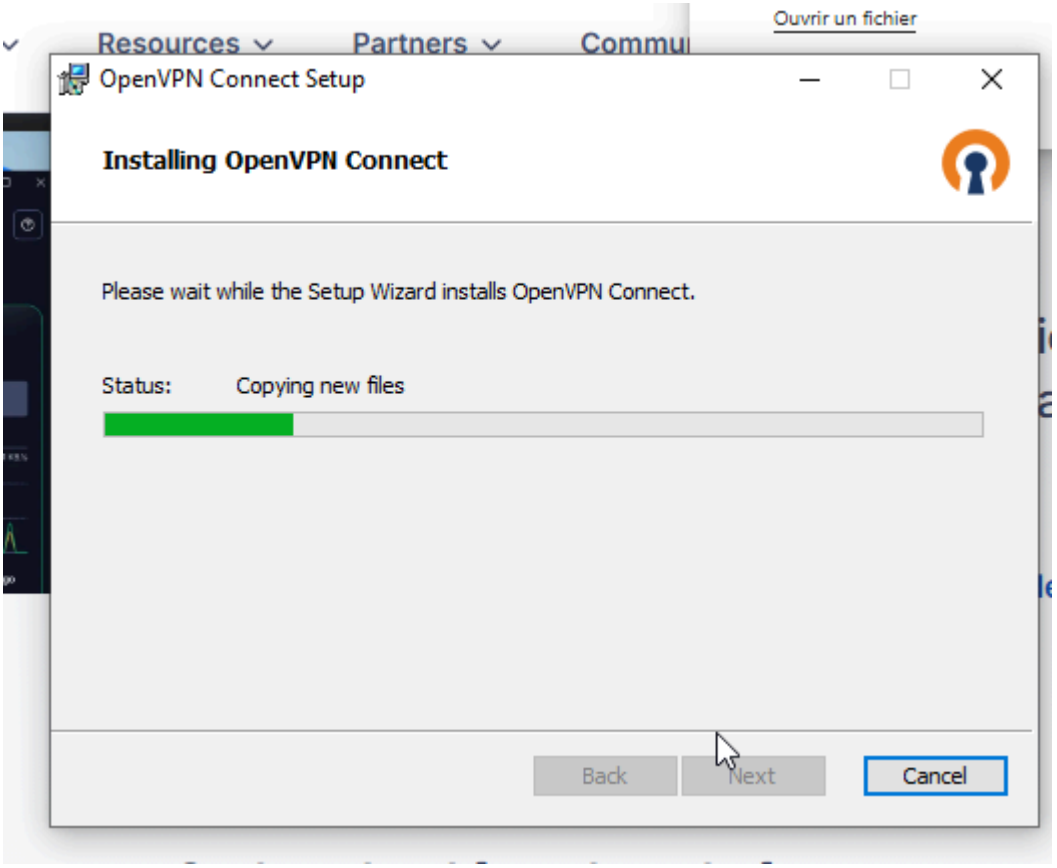
Sur Windows on télécharge et installe :

✔ **OpenVPN Connect** (interface moderne)

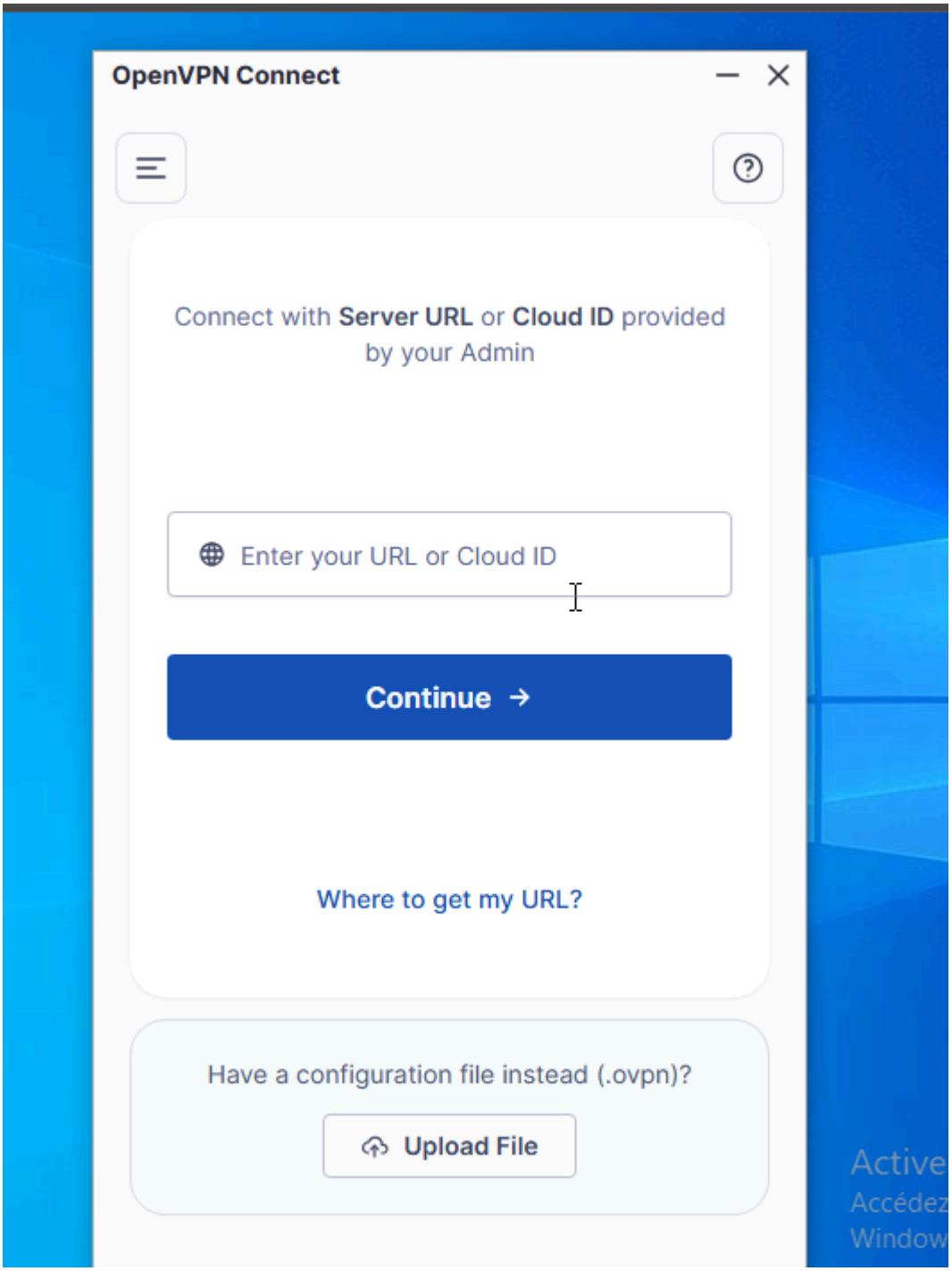


Puis on installe


Téléchargements
openvpn-connect-3.8.0.45
Ouvrir un fichier
Resources
Partners
OpenVPN Connect Setup
Welcome to the OpenVPN Connect Setup Wizard
The Setup Wizard will install OpenVPN Connect on your computer. Click Next to continue or Cancel to exit the Setup Wizard.
Back
Next
Cancel
Or download for other platforms:

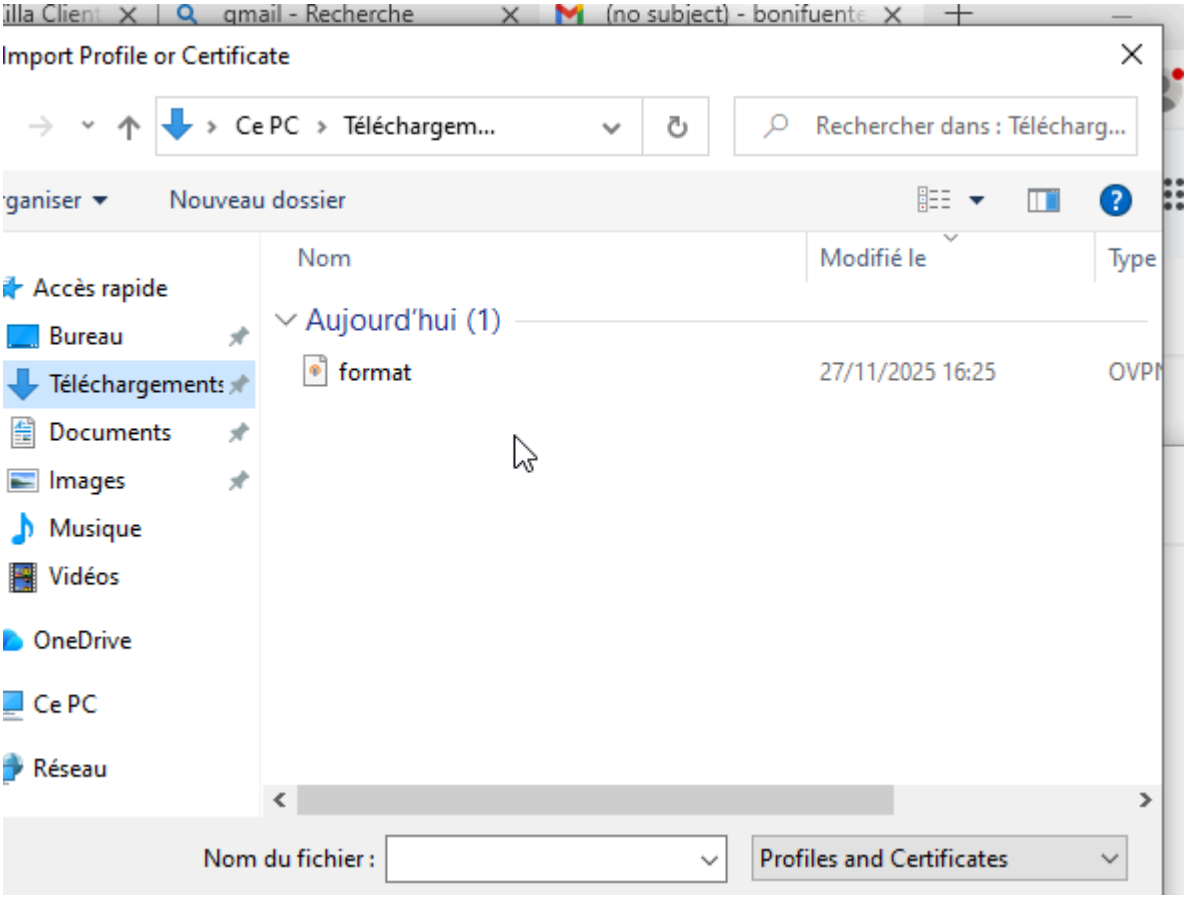


Au final on ouvre l'application et on a 2 moyen de se connecter en vpn soit par url, cloud ou en important un fichier



. Nous on va importer un fichier ovpn que on va crée nous même pour qu'il se connecte en vpn

le fichier se nommera Format.ovpn



voici les modification du fichier:
client
dev tun
proto udp
remote 192.168.11.55 1194
nobind
auth-nocache
remote-cert-tls server

cipher AES-256-GCM
auth SHA256
tls-version-min 1.2
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384

IMPORTANT POUR WINDOWS
key-direction 1
<ca>
-----BEGIN CERTIFICATE-----
MIIEkjCCA3qgAwIBAgIUJcUqzgr0m4U9gI2TfOJ46NImZ5gwDQYJKoZIhvcNAQEL
BQAwgYAxCzAJBgNVBAYTAmZyMRUwEwYDVQQIDAxzZWluZWV0bWFybmUxDjAMBgN
V
BAcMBW1lbHVuMQwwCgYDVQQKDANzaW8xDDAKBgNVBAsMA3NpbzEPMA0GA1UEA
wwG

c3J2dnBuMR0wGwYJKoZIhvcNAQkBFg5tZUBleGFtcGxlLm5ldDAeFw0yNTExMjUx
NDEwMjRaFw0zNTExMjMxNDEwMjRaMIGAMQswCQYDVQQGEwJmcjEVMBMGA1UECA
wM
c2VpbmVldG1hcm5lMQ4wDAYDVQQHDAVtZWx1bjEMMAoGA1UECgwDc2lvMQwwCgYD
VQQLDANzaW8xDzANBgNVBAMMBnNydnZwbjEdMBsGCSqGSIb3DQEJARYObWVAZXh
h
bXBsZS5uZXQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDYgR5tHCuN
vAN40Hj3FWVObGzn8zwzMQQ2u8+hN++P1wt//4/b0VFwMHRqC0M1fPYikFCsna2n
lMLw7ZCeJXna9sPJFRfbkVgoFi5ep5ReF62Pl999jmw3a+4NOH36DqFfPKp489aE
8vJvDFhHgnUXTgmHSoceG8nx4cCmQjhw3yUT3+/FhbLP2jaW8i/cFa5fEd5ykxE8
HiUbuCp+8GbhP78DOobQjpG6KWVf2RVgmOHT5t0xIO5FulaioCW5evwvb7RzAMHT
PA67LOLj7oclfeHvpWRyH39HT//4mWBXLPJQOOQuuQ/NVPn30ThrxA2ApFlgwJfG
yow1T+xN++jVAgMBAAGjggEAMIH9MAwGA1UdEwQFMAMBAf8wHQYDVR0OBBYEFJyb
+gPv18s20bCtLxMUmo01lM4aMIHABgNVHSMEgbgwgbWAFJyb+gPv18s20bCtLxMU
mo01lM4aoYGGpIGDMIGAMQswCQYDVQQGEwJmcjEVMBMGA1UECAwMc2VpbmVldG1h
cm5lMQ4wDAYDVQQHDAVtZWx1bjEMMAoGA1UECgwDc2lvMQwwCgYDVQQLDANzaW8x
DzANBgNVBAMMBnNydnZwbjEdMBsGCSqGSIb3DQEJARYObWVAZXhhbXBsZS5uZXSC
FCXFKs4K9JuFPYCNk3zieOjSJmeYMAsGA1UdDwQEAwIBBjANBgkqhkiG9w0BAQsF
AAOCAQEAguF/tHCd17jWhzSBtCC0BCZFnssYbwEC7GWmUA8Z1D0Q2apPfiBMhyb4
Rnsd0/1IwMLHdW9jTiFfSaS5Re3mDq1wBiG0eOwZMMoDKyMO+H29nfnHpSua/Jv+
648ApJrpiDfakaRASOghZP9Rn4sgy3txvAP9+AEUz/7Hqm2X2YHKskAfa3xKkiM2
JMGUXz4ZoOH3Kc7OMeza6z8/i3BieLBx25pz4t/vl845CE44im4vYIsaU7DMH/bA
l4HROZosOj9JLgWMV01ugDxpRAOxGB4n8L/R6MS9DKU+2zm11yM8GsYnyBKVpYya
lFAmgyYSB2hXiu3TIGMLEbqLGTElGg==
-----END CERTIFICATE-----
</ca>
<cert>
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            1b:73:85:a8:b3:a7:9c:80:cb:b2:cd:87:4c:fa:b1:1d
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=fr, ST=seineetmarne, L=melun, O=sio, OU=sio,
CN=srvvpn/emailAddress=me@example.net
        Validity
            Not Before: Nov 25 15:54:58 2025 GMT
            Not After : Nov 25 15:54:58 2026 GMT
        Subject: C=fr, ST=seineetmarne, L=melun, O=sio, OU=sio,
CN=cltvpn/emailAddress=me@example.net
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ae:8b:22:f5:30:38:c4:89:71:a8:cd:34:af:3a:
                    3e:bc:ff:bd:3d:7a:e6:ee:92:c5:4c:61:22:f8:3d:

```
                98:cb:b0:ee:97:98:06:82:79:97:32:bf:6d:3e:c0:
                66:23:fb:64:8f:08:46:92:07:8b:85:28:26:90:bc:
                8a:09:af:25:77:66:88:63:e4:8b:bb:2a:05:13:9c:
                fd:f5:e3:ad:00:c5:12:04:37:38:c1:18:13:5f:4e:
                cf:e8:60:0c:90:d2:7f:39:65:37:78:80:cd:e4:d4:
                47:17:16:bd:61:0c:44:03:75:91:32:95:36:1d:b7:
                67:4d:0a:90:be:65:90:7a:cd:5c:71:6f:36:43:e6:
                67:6a:29:37:7f:02:3b:16:bd:2c:84:7f:06:ce:d4:
                52:60:2b:1e:be:55:34:e7:a1:da:f1:27:6b:f6:db:
                fb:3f:b1:b4:6b:f1:53:31:9f:9e:b2:1c:7c:57:af:
                98:fa:ff:e6:1c:fa:55:f3:c1:c6:c7:25:0d:fd:03:
                4e:fe:35:cb:db:f4:a3:f0:f2:24:0c:4c:39:19:b2:
                32:e1:83:97:a2:ae:c6:c7:4e:6e:50:67:73:ab:39:
                59:99:44:79:e1:cb:d1:7d:0f:a1:60:88:d0:e0:e8:
                6a:44:80:ca:7d:47:69:06:65:43:41:1d:d8:e2:ae:
                fd:53
            Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                CF:41:51:FE:8A:CF:51:E8:20:44:BA:DE:AD:B2:AF:B9:BC:E3:3F:F5
            X509v3 Authority Key Identifier:
                keyid:9C:9B:FA:03:EF:D7:CB:36:D1:B0:AD:2F:13:14:9A:8D:35:94:CE:1A

DirName:/C=fr/ST=seineetmarne/L=melun/O=sio/OU=sio/CN=srvvpn/emailAddress=me@ex
ample.net
                serial:25:C5:2A:CE:0A:F4:9B:85:3D:80:8D:93:7C:E2:78:E8:D2:26:67:98
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
            X509v3 Key Usage:
                Digital Signature
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        04:61:fc:6f:ff:86:b3:75:e1:5d:31:68:e5:b8:80:bb:76:9a:
        60:e0:32:ab:d2:e1:97:30:b0:ea:6e:fd:ba:88:ff:9e:fd:a1:
        d2:81:88:8e:e4:c1:e1:39:7e:cb:2f:32:fb:4d:ff:c7:86:fd:
        43:30:29:8c:f3:ac:bc:8d:90:b4:0d:2c:07:d7:f7:d2:5c:f6:
        6f:59:c9:a0:f8:eb:59:a9:ef:f7:0c:20:83:be:98:77:c1:8c:
        ff:21:ef:95:21:1f:86:3b:d4:7e:85:3c:a3:6f:2a:42:5d:cf:
        57:d7:54:fa:2e:25:99:fb:26:61:b6:77:a9:d0:dd:d6:33:2e:
        4d:8d:2e:2b:ef:6c:94:74:8c:c3:69:43:4d:96:3c:2e:dd:e7:
        ad:db:69:f3:b2:fd:14:37:f3:15:40:b4:5d:a1:e2:fc:3a:03:
        95:69:c9:3d:bd:d1:b3:bc:14:b6:43:a2:e4:20:df:7f:bf:53:
        34:f0:18:5e:5e:19:8c:eb:6d:b5:e5:66:97:b1:ce:d0:c8:71:
        21:cc:86:49:31:8a:e2:33:93:54:69:5f:1d:93:74:8f:0e:36:
        e5:e7:ff:b0:40:57:f3:88:89:10:97:3b:aa:be:74:5b:03:a6:
        7e:82:25:c1:93:2a:91:1b:92:ed:9c:d0:3d:f4:94:39:74:44:
```

41:76:98:c9
-----BEGIN CERTIFICATE-----
MIIEoTCCA4mgAwIBAgIQG3OFqLOnnIDLss2HTPqxHTANBgkqhkiG9w0BAQsFADCB
gDELMAkGA1UEBhMCZnIxFTATBgNVBAgMDHNlaW5lZXRtYXJuZTEOMAwGA1UEBwwF
bWVsdW4xDDAKBgNVBAoMA3NpbzEMMAoGA1UECwwDc2lvMQ8wDQYDVQQDDAZzcn
Z2
cG4xHTAbBgkqhkiG9w0BCQEWDm1lQGV4YW1wbGUubmV0MB4XDTI1MTEyNTE1NTQ1
OFoXDTI2MTEyNTE1NTQ1OFowgYAxCzAJBgNVBAYTAmZyMRUwEwYDVQQIDAxzZWlu
ZWV0bWFybmUxDjAMBgNVBAcMBW1lbHVuMQwwCgYDVQQKDANzaW8xDDAKBgNVB
AsM
A3NpbzEPMA0GA1UEAwwGY2x0dnBuMR0wGwYJKoZIhvcNAQkBFg5tZUBleGFtcGxl
Lm5ldDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAK6LIvUwOMSJcajN
NK86Prz/vT165u6SxUxhIvg9mMuw7peYBoJ5lzK/bT7AZiP7ZI8IRpIHi4UoJpC8
igmvJXdmiGPki7sqBROc/fXjrQDFEgQ3OMEYE19Oz+hgDJDSfzlIN3iAzeTURxcW
vWEMRAN1kTKVNh23Z00KkL5lkHrNXHFvNkPmZ2opN38COxa9LIR/Bs7UUmArHr5V
NOeh2vEna/bb+z+xtGvxUzGfnrIcfFevmPr/5hz6VfPBxsclDf0DTv41y9v0o/Dy
JAxMORmyMuGDl6KuxsdOblBnc6s5WZlEeeHL0X0PoWCI0ODoakSAyn1HaQZlQ0Ed
2OKu/VMCAwEAAaOCARMwggEPMAkGA1UdEwQCMAAwHQYDVR0OBBYEFM9BUf6Kz
1Ho
lES63q2yr7m84z/1MIHABgNVHSMEgbgwgbWAFJyb+gPv18s20bCtLxMUmo01lM4a
oYGGpIGDMIGAMQswCQYDVQQGEwJmcjEVMBMGA1UECAwMc2VpbmVldG1hcm5lMQ
4w
DAYDVQQHDAVtZWx1bjEMMAoGA1UECgwDc2lvMQwwCgYDVQQLDANzaW8xDzANBgN
V
BAMMBnNydnZwbjEdMBsGCSqGSIb3DQEJARYObWVAZXhhbXBsZS5uZXSCFCXFKs4K
9JuFPYCNk3zieOjSJmeYMBMGA1UdJQQMMAoGCCsGAQUFBwMCMAsGA1UdDwQEAw
IH
gDANBgkqhkiG9w0BAQsFAAOCAQEABGH8b/+Gs3XhXTFo5biAu3aaYOAyq9LhlzCw
6m79uoj/nv2h0oGIjuTB4Tl+yy8y+03/x4b9QzApjPOsvI2QtA0sB9f30lz2b1nJ
oPjrWanv9wwgg76Yd8GM/yHvlSEfhjvUfoU8o28qQl3PV9dU+i4ImfsmYbZ3qdDd
1jMuTY0uK+9slHSMw2lDTZY8Lt3nrdtp87L9FDfzFUC0XaHi/DoDlWnJPb3Rs7wU
tkOi5CDff79TNPAYXl4ZjOttteVml7HO0MhxIcyGSTGK4jOTVGlfHZN0jw425ef/
sEBX84iJEJc7qr50WwOmfoIlwZMqkRuS7ZzQPfSUOXREQXaYyQ==
-----END CERTIFICATE-----

</cert>
<key>
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCuiyL1MDjEiXGo
zTSvOj68/709eubuksVMYSL4PZjLsO6XmAaCeZcyv20+wGYj+2SPCEaSB4uFKCaQ
vIoJryV3Zohj5Iu7KgUTnP31460AxRIENzjBGBNfTs/oYAyQ0n85ZTd4gM3k1EcX
Fr1hDEQDdZEylTYdt2dNCpC+ZZB6zVxxbzZD5mdqKTd/AjsWvSyEfwbO1FJgKx6+
VTTnodrxJ2v22/s/sbRr8VMxn56yHHxXr5j6/+Yc+lXzwcbHJQ39A07+Ncvb9KPw
8iQMTDkZsjLhg5eirsbHTm5QZ3OrOVmZRHnhy9F9D6FgiNDg6GpEgMp9R2kGZUNB
Hdjirv1TAgMBAAECggEACmeZME6Ilg/T/NNP4a9Wvcyjytm3bHGigh07GyRWk9go
YzArPXVQqMZ/FV1B5y7rFGH3Mda7TW4Jb6MLTKOhMyFaS8Igs/etbF2+bUZr6LVo
/WOhU8KME/Fgoo/6ohEn9TgSsbqvVIW90f6sBxTW0IRC6kFwOoxnleMcWOimdA3E
zujS87fSjLeJ2Kkp0h3e3hnnD0KzxCAIO2OLyt5SvwnMk5lSe4OoR0UhNjBr/QI+

xSqRd0IGN+WdQItpuEkkUx30VUrZ1GOzzwjdMVzmhdOtuj6O26yOn3qU9f6Rethx
Yp6cR+bNBYKBNNLsQS8IgQvdP62k52xu0+FV+6SeqQKBgQDsP/hBmE8WqI4ni6+g
DOXPw+VOqjwl5dXtsdqM+mcGmH7ZdTwqTCRB0zdtiAAJFlbCZbVUH9kEj4tblC9b
R/eOcSjqwy0SfNnJJj4AFOEAcUNouettP0ObvvSrDNQfOdQdTb1vMDagAp/2h5LA
3ktlhl1ih+jMDYRc/EwHRGIrnQKBgQC9IpOQLhT2WSEWNZvHR5uXngRB9PFksZOx
BDfohQquoAxqbH5Op5FuLGsxurQVRqn8JggZzlnjEJXMY+QDZs0JxAoOpNmeu4oq
LE5VCqZhzw2aY7/SeMNST0t+3vR8ia7y8jZ8bSzjAuM7HnzNV6GipEvKlYiDWKFu
2LcmAFjRrwKBgQCS+Pz1uc5B91xwZ6s7s6pohaMrPCatIP3koMeo+sKLTOVlXVSv
r6zCsqcB4iC7hoU07NdQ3XtCJhb5QwfvDqeImwdsfT2WCcvnIn5hyJttjgj0pu06
iqgBxVsq6pEwV+aI0GswN9tT0i43zYHTsLdRErUX7hs6jpIw0thtkPJN1QKBgEBS
UfzmrxpYlXgII0JSiAe4JZqIws+L1L6OwLTDCi75Hz/SReJZ+ouNb6cp57SUo5qt
FHQyST9/lHI+Rn0hqPl5QjJYvenNvd850in5xTvePrIS2lmp2ENV1EcBbAQujuDw
nrcSKgHKKZ0hcTAkbUPyvBOx88s3iS85Zl/37jzhAoGADhHinphunpMWrbFdKCr/
z4snyzar0KxShw6DfSBHIsnLP3o/kGXr5vew3U4INqAXFIKBiVbqADpr6Hcd4kod
vIENq9ORp9vWhLKKHvoDI7wmfuaQEWDsATW0mFKifgVzsV7V5PcVDHPnwdkKLzsG
W3YSpmPuSWSXdfG7ODYAvqk=
-----END PRIVATE KEY-----
</key>
<tls-auth>
-----BEGIN OpenVPN Static key V1-----
8ea9857f7ce6f9b2e637e5b326d9d687
db089dcb6781d1925224441d3926d469
170b9eadd6fc9b23a2dd37ad830e8a18
4a5b41a6944c8121f7ff38bc88137617
7045d3729ef4f77249e505dc135fa89b
7ab83ada74ec9bef649ae5764109a2ef
19a8935e8b2d91ac801a64a74380efd0
c4e45afaaaf87abd70c63a94b564715d
228d3e5f954980bf3377e7ffb30040a6
3572585ff23b90dcc63346340b6a1ea5
2912da541ca5f8c2c82217bc1dade4bf
b46823ff58bf2c9bd5f643ae78b859c1
931e46674560ca6660213c741970aaae
77a3d9da1d5e146f0f44c1682d2a3b29
0afc3ae209844b13882a4f1b0065f877
7483a5d3a44c876a588c4589ab0ea15f
-----END OpenVPN Static key V1-----

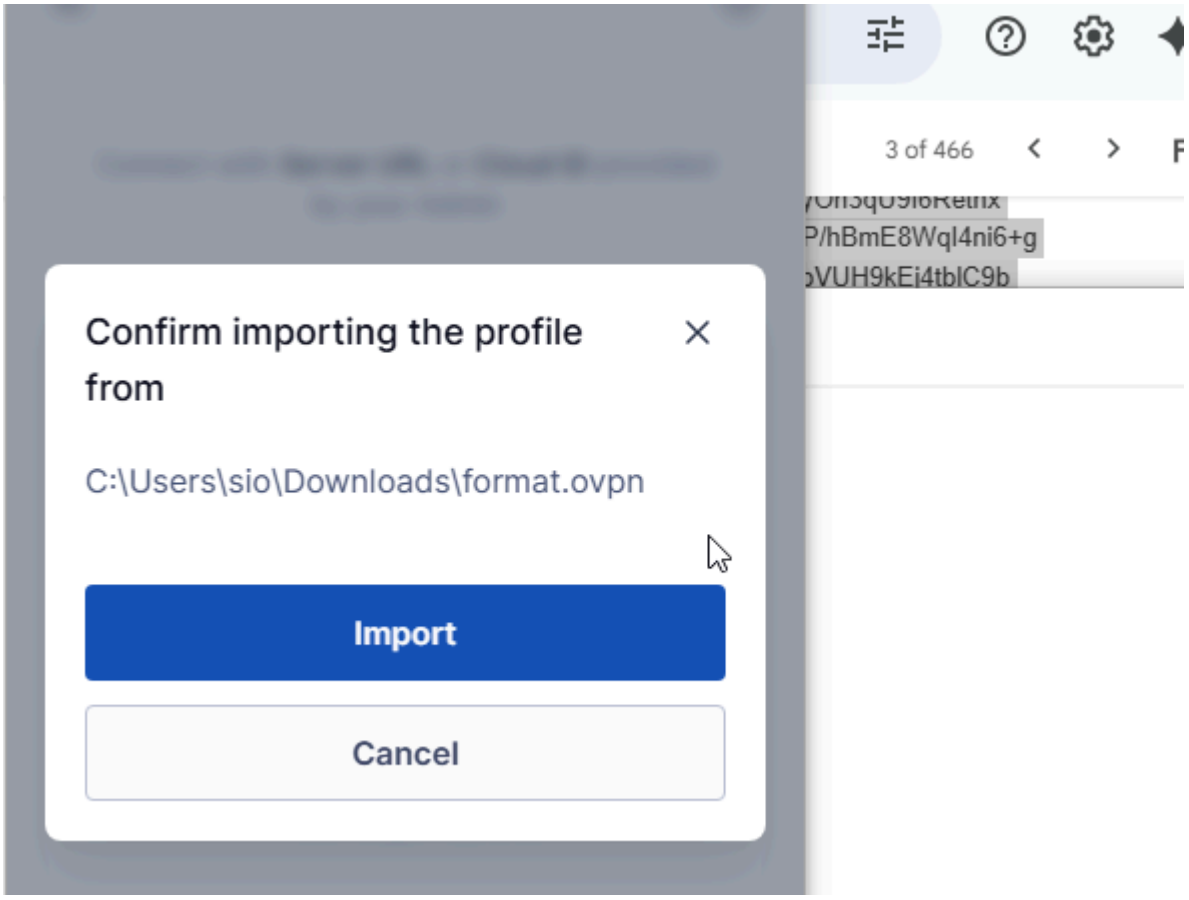
</tls-auth>

explication :

1. **Mode Client** : La directive `client` indique qu'il s'agit d'un poste client, et non d'un serveur.
2. **Tunnel & Protocole** : `dev tun` crée l'interface réseau virtuelle, et `proto udp` utilise le protocole UDP pour des connexions rapides.

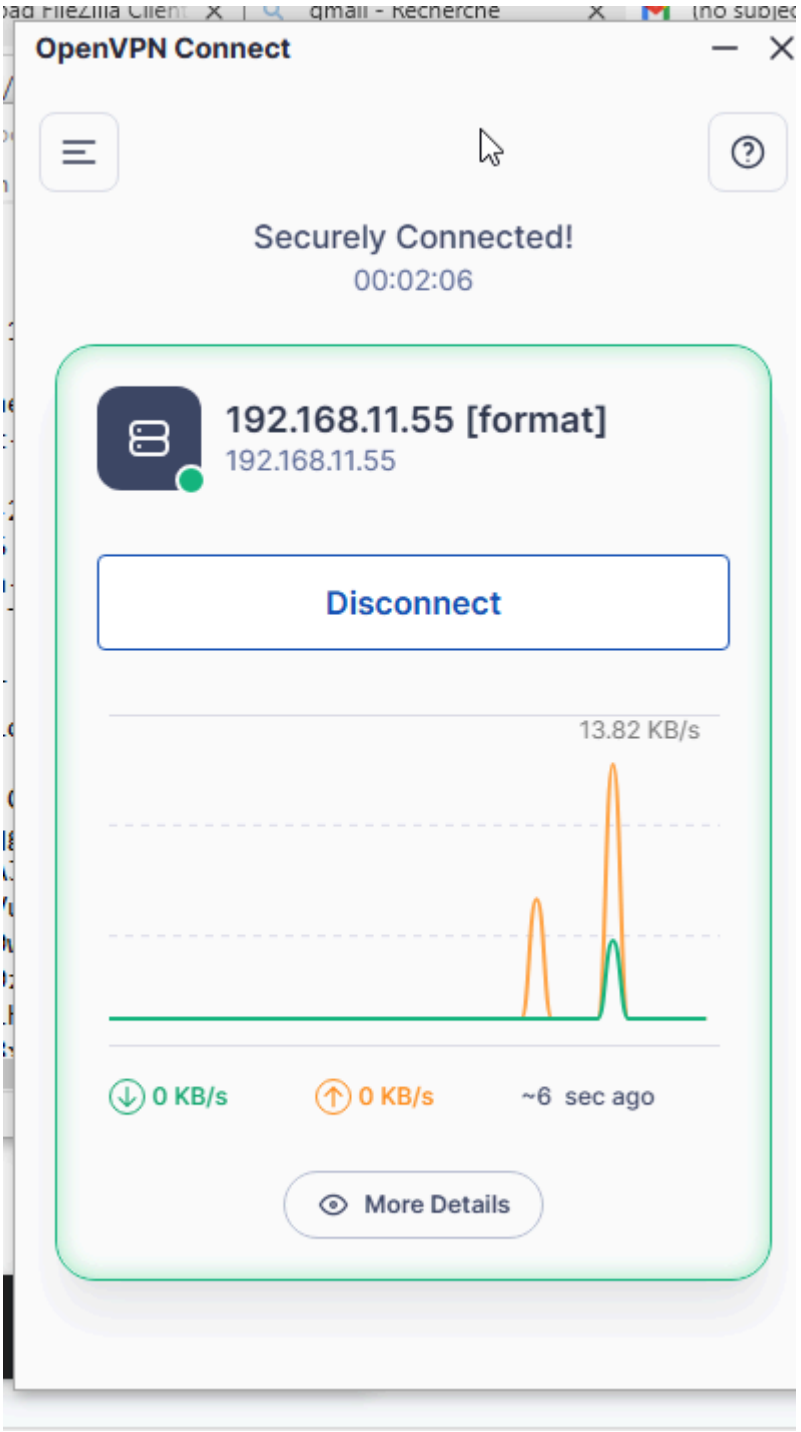
3. **Adresse du Serveur** : `remote 192.168.11.55 1194` définit l'adresse IP et le port **exacts** où joindre le serveur VPN.
4. **Connexion** : `nobind` autorise le client à utiliser n'importe quel port source disponible (plus flexible).
5. **Sécurité TLS** : `remote-cert-tls server` confirme que le client n'acceptera que les certificats qui sont explicitement signés comme étant des certificats de serveur.
6. **Chiffrement/Hachage** : `cipher AES-256-GCM` et `auth SHA256` spécifient les algorithmes cryptographiques modernes pour chiffrer les données.
7. **Version TLS** : `tls-version-min 1.2` force l'utilisation d'une version récente et sécurisée du protocole TLS.
8. **Clé HMAC** : `key-direction 1` (avec la balise `<tls-auth>`) est nécessaire pour les clients utilisant Windows et ajoute une clé secrète partagée pour éviter les attaques (HMAC Firewall).
9. **Certificat CA** : La balise `<ca>` contient le certificat de l'**Autorité de Certification** (CA), permettant au client de vérifier l'authenticité du serveur.
10. **Certificat Client** : La balise `<cert>` contient le certificat **public** unique du client (`CN=cltvpn`), utilisé pour s'authentifier auprès du serveur.
11. **Clé Privée Client** : La balise `<key>` contient la clé **privée** unique du client, essentielle pour établir le tunnel sécurisé.
12. **Clé TLS** : La balise `<tls-auth>` contient la clé statique additionnelle utilisée conjointement avec `key-direction 1` pour la couche HMAC de protection.

Puis on emporte le fichier



on aura ce résultat:

il est en vert et il arrive a se connecter en vpn



Vérification du bon fonctionnement : *

Après que le client windows est connecter en vpn au serveur , on va au SrvVpn et on fait la commande tail -f /openvpn.log dans le dossier /var/log/openvpn

```
root@srvVpn:~# cat /var/log/openvpn/openvpn.log
```

on constate la connexion du client 192.168.11.56

```
025-11-25 16:30:54 192.168.11.56:49800 TLS: move_session: dest=TM_ACTIVE src=TM_INITIAL reinit_src=1
025-11-25 16:30:54 192.168.11.56:49800 TLS: tls_multi_process: initial untrusted session promoted to tr
025-11-25 16:30:54 192.168.11.56:49800 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384,
, signature: RSA-SHA256, peer temporary key: 253 bits X25519
025-11-25 16:30:54 192.168.11.56:49800 [cltvpn] Peer Connection Initiated with [AF_INET]192.168.11.56:4
025-11-25 16:30:54 cltvpn/192.168.11.56:49800 MULTI_sva: pool returned IPv4=10.8.0.6, IPv6=(Not enabled
025-11-25 16:30:54 cltvpn/192.168.11.56:49800 MULTI: Learn: 10.8.0.6 -> cltvpn/192.168.11.56:49800
025-11-25 16:30:54 cltvpn/192.168.11.56:49800 MULTI: primary virtual IP for cltvpn/192.168.11.56:49800:
025-11-25 16:30:54 cltvpn/192.168.11.56:49800 SENT CONTROL [cltvpn]: 'PUSH_REPLY,route 10.8.0.1 255.255
.12.1,redirect-gateway def1,route 10.8.0.0 255.255.255.0,topology net30,ping 10,ping-restart 120,ifconf
ipher AES-256-GCM,protocol-flags cc-exit tls-ekm dyn-tls-crypt,tun-mtu 1500' (status=1)
025-11-25 16:30:54 cltvpn/192.168.11.56:49800 PUSH: Received control message: 'PUSH_REQUEST'
025-11-25 16:30:55 cltvpn/192.168.11.56:49800 Data Channel: cipher 'AES-256-GCM', peer-id: 0
025-11-25 16:30:55 cltvpn/192.168.11.56:49800 Timers: ping 10, ping-restart 240
025-11-25 16:30:55 cltvpn/192.168.11.56:49800 Protocol options: protocol-flags cc-exit tls-ekm dyn-tls-
oot@srvVpn:~#
```

puis un tail -f open-status on regard l'etat des connexion

```
root@srvVpn:~# tail -f /var/log/openvpn/openvpn-status.log
OpenVPN CLIENT LIST
Updated,2025-11-25 16:36:37
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
cltvpn,192.168.11.56:49800,607684,338169,2025-11-25 16:30:54
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
10.8.0.6,cltvpn,192.168.11.56:49800,2025-11-25 16:36:35
GLOBAL STATS
Max bcast/mcast queue length,1
END
```

Puis on revient sur le client windows on ouvre un cmd et on fait un ping au serveur glpi 192.168.13.19 en utilisant le nom dns glpi.menuimetal.fr

```
*** dns ne parvient pas à trouver glpi : Query refused
>
C:\Windows\system32>ping glpi.menuimetal.fr

Envoi d'une requête 'ping' sur glpi.menuimetal.fr [192.168.13.19] avec 32 octets de données :
Réponse de 192.168.13.19 : octets=32 temps=2 ms TTL=62
Réponse de 192.168.13.19 : octets=32 temps=2 ms TTL=62
Réponse de 192.168.13.19 : octets=32 temps=2 ms TTL=62
Réponse de 192.168.13.19 : octets=32 temps=2 ms TTL=62

Statistiques Ping pour 192.168.13.19:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 2ms, Maximum = 2ms, Moyenne = 2ms
```

on fait un ping vers le serveur nagios aussi 192.168.13.60

```
^C
C:\Windows\system32>ping 192.168.13.60

Envoi d'une requête 'Ping'  192.168.13.60 avec 32 octets de données :
Réponse de 192.168.13.60 : octets=32 temps=2 ms TTL=62
Réponse de 192.168.13.60 : octets=32 temps=2 ms TTL=62
Réponse de 192.168.13.60 : octets=32 temps=2 ms TTL=62
Réponse de 192.168.13.60 : octets=32 temps=2 ms TTL=62

Statistiques Ping pour 192.168.13.60:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 2ms, Maximum = 2ms, Moyenne = 2ms

C:\Windows\system32>
```

puis au final on fait un TRACER sur le serveur glpi

```
    Minimum = 2ms, Maximum = 3ms, Moyenne = 2ms

C:\Users\sio>tracert 192.168.13.19

Détermination de l'itinéraire vers 192.168.13.19 avec un maximum de 30 sauts.

  1    <1 ms    <1 ms    <1 ms  10.8.0.1
  2     1 ms    <1 ms     1 ms  192.168.11.254
  3     2 ms     1 ms     1 ms  192.168.13.19

Itinéraire déterminé.

C:\Users\sio>
```

La commande `tracert` a été utilisée pour vérifier l'itinéraire réseau vers le serveur GLPI (`192.168.13.19`).
Le premier saut crucial à **10.8.0.1** confirme que votre trafic utilise activement la passerelle du **tunnel OpenVPN**.
Cela prouve que la configuration client/serveur VPN fonctionne pour acheminer les paquets.

Connexion vpn sur un poste client

Tout d'abord on modifie le switch cisco , on le branche dans le port 2 du cisco et puis dans le port gi0/46 du switch du lycée pour avoir accès à tous les vlans (invités, lan ,gestion etc..)

gi0/13-14	5x4[1]	Invités (INVITES)
gi0/21-24	50	Réseau salles SIO (access)
gi0/45-47	99	Réseau salles SIO (access)
gi0/48	Trunk	uplink 1Gbits/s vers le cœur de réseau

Puis , On rentre on mode config , puis on accède à fastethernet0/2

```
moncisco(config)#interface fastEthernet0/2
moncisco(config-if)#swi
moncisco(config-if)#switchport mode
moncisco(config-if)#switchport mode tru
moncisco(config-if)#switchport mode trunk
Command rejected: An interface whose trunk encapsulation is "Auto" can not be co
nfigured to "trunk" mode.
moncisco(config-if)#switchport mode trunk
Command rejected: An interface whose trunk encapsulation is "Auto" can not be co
nfigured to "trunk" mode.
moncisco(config-if)#
moncisco(config-if)#
```

u

Cette commande configure le Port 2 en **lien Trunk**, lui permettant de transporter le trafic de **plusieurs VLANs** différents. Elle impose le standard **802.1Q** pour l'encapsulation (le taggage) des trames

```
moncisco(config-if)#switchport trunk encapsulation dot
moncisco(config-if)#switchport trunk encapsulation dot1q
```

Cette commande switchport trunk allowed vlan 581.582 etc.. **filtre** le trafic en spécifiant que seuls les VLANs listés (580, 581, 582, 583) sont autorisés à passer par ce lien **Trunk**. Cela garantit la sécurité et l'efficacité en bloquant le trafic de tous les autres VLANs du switch sur ce port.

```
moncisco(config-if)#switchport trunk allowed vlan 580,581,582,583
moncisco(config-if)#
```

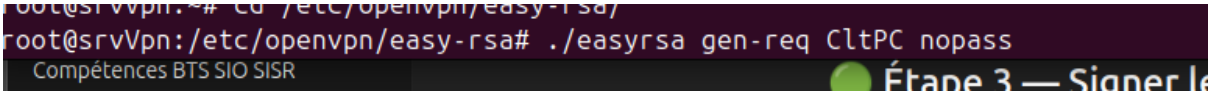
puis on fait un show ip interface brief pour regarder les ip . l'ip de 192.168.11.254 est celui du port2 fastethernet0/2

```
Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      YES NVRAM  administratively down down
Vlan580                192.168.13.29   YES NVRAM  up                    up
Vlan581                192.168.11.254  YES manual up                    up
FastEthernet0/1        unassigned      YES unset  up                    up
FastEthernet0/2        unassigned      YES unset  up                    up
FastEthernet0/3        unassigned      YES unset  down                  down
FastEthernet0/4        unassigned      YES unset  down                  down
FastEthernet0/5        unassigned      YES unset  down                  down
FastEthernet0/6        unassigned      YES unset  down                  down
FastEthernet0/7        unassigned      YES unset  down                  down
FastEthernet0/8        unassigned      YES unset  down                  down
FastEthernet0/9        unassigned      YES unset  down                  down
FastEthernet0/10       unassigned      YES unset  down                  down
```

Srv Vpn configuration nécessaires

on va dans le dossier easyrsa et on écrite la commande ./easyrsa gen-req CltPC nopass

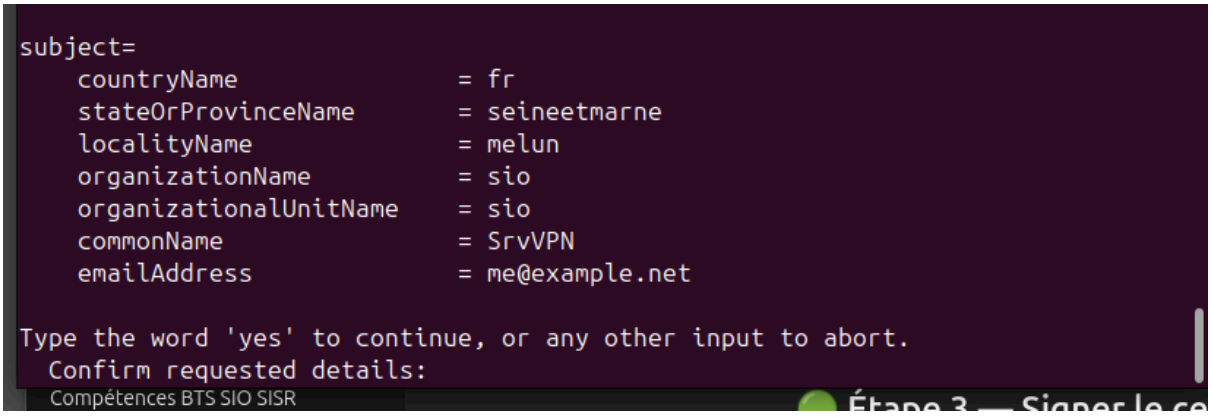
Cette commande consiste à utiliser l'outil **Easy-RSA** pour **générer la demande de certificat** et la clé privée pour le nouvel utilisateur `CltPC`. L'option `nopass` indique que la clé privée du client sera créée **sans mot de passe**, simplifiant la connexion mais réduisant la sécurité



Cette commande génère la **demande de certificat (CSR)** et la clé privée sans mot de passe pour le client nommé `CltPC` en utilisant Easy-RSA.

```
root@srvVpn:~# ./easyrsa gen-req CltPC nopass
```

puis on fait yes

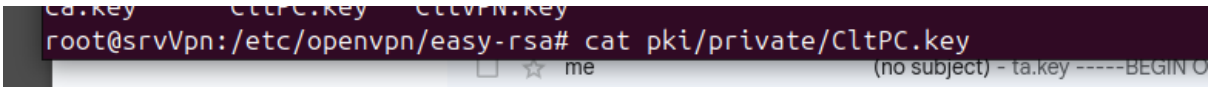


CAT

Cette commande affiche le contenu du certificat client `CltPC.crt` et confirme qu'il a été **correctement émis et signé** par le serveur VPN (l'émetteur) avec une validité d'un an.

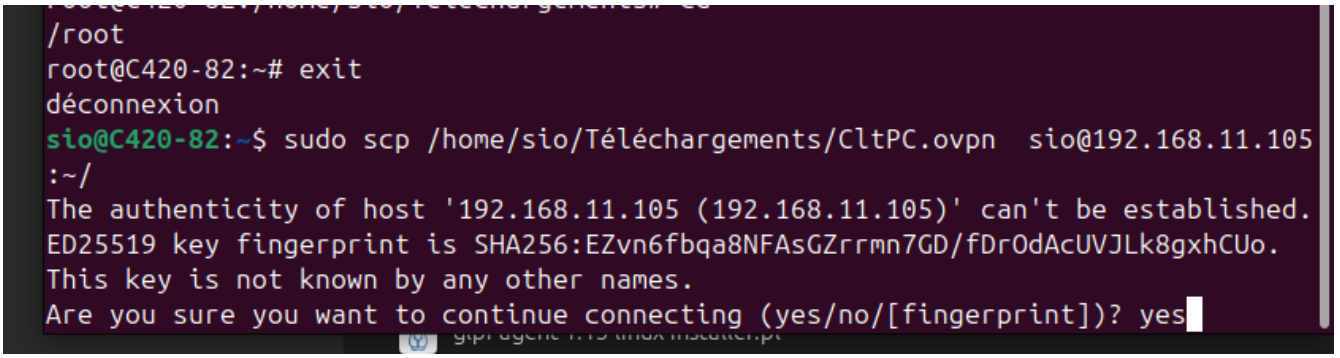
```
CltPC.crt   CltVPN.crt
root@srvVpn:/etc/openvpn/easy-rsa# cat pki/issued/CltPC.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            6e:70:51:2d:3c:8a:ab:df:bf:6d:99:eb:3e:62:85:e1
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=fr, ST=seineetmarne, L=melun, O=sio, OU=sio, CN=srvvpn/emailAd
dress=me@example.net
        Validity
            Not Before: Nov 26 08:47:05 2025 GMT
            Not After : Nov 26 08:47:05 2026 GMT
```

Cette commande affiche le contenu de la **clé privée** du client (`CltPC.key`) stockée dans le dossier `pki/private/`, un fichier **sensible et confidentiel** nécessaire pour le chiffrement.

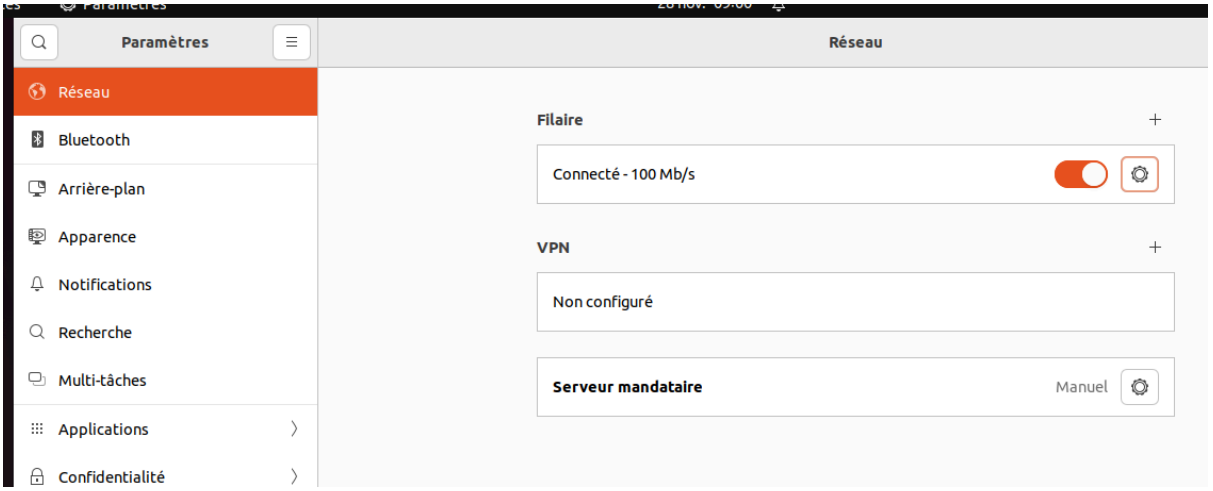


Puis avec les les clés récupérés on partage dans le clientPC poste

La commande **`scp`** est utilisée pour **copier en toute sécurité** le fichier de configuration VPN (`CltPC.ovpn`) vers le répertoire de l'utilisateur `sio` sur le serveur distant (`192.168.11.105`).
Le système alerte l'utilisateur que la **clé d'hôte** (fingerprint) du serveur est inconnue, demandant confirmation (`yes`) pour l'enregistrer et établir la connexion sécurisée (SSH/SCP).

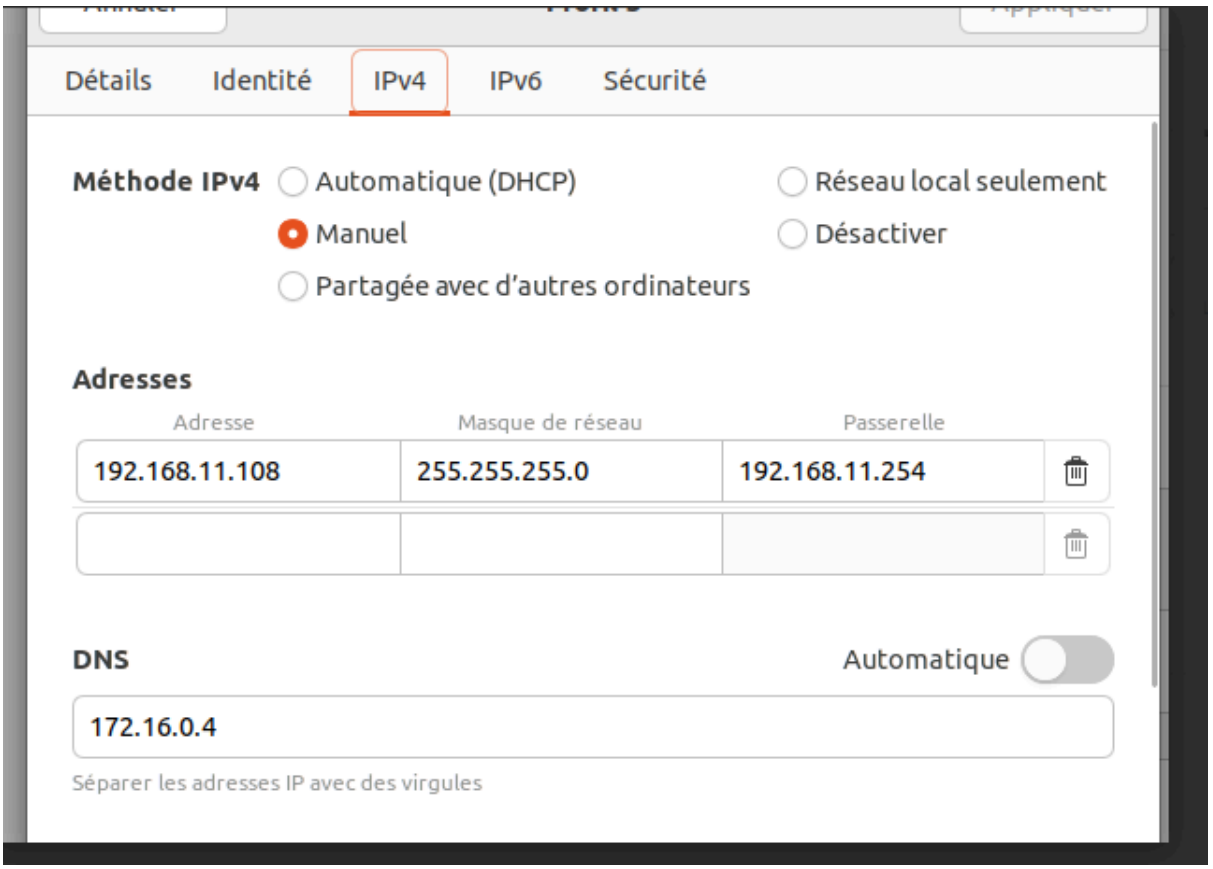


Dans le cltpc on modifie l'interface graphique :
dans réseau on modifie les paramètres réseaux



Explication:
Méthode d'adressage : La sélection de l'option "Manuel" désactive l'attribution automatique par DHCP.
Adresse Locale : L'interface réseau se voit attribuer une adresse IP statique unique, **192.168.11.108** (Masque `255.255.255.0`).
Routage : La communication vers l'extérieur du sous-réseau est assurée par la **passerelle par défaut 192.168.11.254**.
Résolution de Noms : La résolution DNS est forcée manuellement sur l'adresse **172.16.0.4**, ignorant tout DNS fourni par le réseau local.

Synthèse : Cette configuration garantit que le poste conserve une **identité réseau fixe** et utilise un serveur DNS spécifique, ce qui est typique pour les serveurs ou les machines critiques.



Proxy: ajout du proxy

J’ai refais des clés et certificats pour le client PC (“M.guillien me la demander”)

```
root@srvVpn:/etc/openvpn/easy-rsa# ./easyrsa sign-req client CltPC
Using Easy-RSA 'vars' configuration:
```

```
subject=
    countryName               = fr
    stateOrProvinceName       = seineetmarne
    localityName              = melun
    organizationName          = sio
    organizationalUnitName    = sio
    commonName                = SrvVPN
    emailAddress              = me@example.net

Type the word 'yes' to continue, or any other input to abort.
  Confirm requested details:
```


Puis d'après un apt install openvpn ,
Importation du fichier client OpenVPN
Cette commande importe le fichier VPN *CltPC.ovpn* dans NetworkManager.
Elle crée automatiquement une connexion VPN prête à être utilisée sur le client Linux

```
root@sio-proto:~# nmcli connection import type openvpn file ~/CltPC.ovpn
Connexion « CltPC » (9a2edb96-239b-426b-8576-28169214d3a0) ajoutée avec succès.
root@sio-proto:~#
```

Démarrage / Connexion au VPN
➡️ Active la connexion VPN *CltPC*.
➡️ À partir de ce moment, ton PC est connecté au serveur VPN

```
root@sio-proto:~# nmcli connection up C
CltPC                  Connexion\ filaire\ 1  Connexion\ filaire\ 2
root@sio-proto:~# nmcli connection up CltPC
```

Puis dans l’interface dans réseau on aura ce beau résultat

voici un ping vers le serveur Glpi

```
root@sio-proto:~# ping 192.168.13.19
PING 192.168.13.19 (192.168.13.19) 56(84) bytes of data.
64 bytes from 192.168.13.19: icmp_seq=3 ttl=62 time=2.67 ms
64 bytes from 192.168.13.19: icmp_seq=4 ttl=62 time=2.70 ms
```

et aussi un ping au nom dns du glpi

```
root@sio-proto:~# ping glpi.menuimetal.fr
PING glpi.menuimetal.fr (192.168.13.19) 56(84) bytes of data.
64 bytes from 192.168.13.19 (192.168.13.19): icmp_seq=1 ttl=62 time=1.81 ms
64 bytes from 192.168.13.19 (192.168.13.19): icmp_seq=3 ttl=62 time=2.26 ms
64 bytes from 192.168.13.19 (192.168.13.19): icmp_seq=7 ttl=62 time=2.11 ms
^C
```

coté serveur vpn on va dans /openvpn-log et on aura ce résultat

```
2025-11-28 09:54:55 SrvVPN/192.168.11.106:47989 MULTI: primary virtual IP for Sr
vVPN/192.168.11.106:47989: 10.8.0.6
2025-11-28 09:54:55 SrvVPN/192.168.11.106:47989 SENT CONTROL [SrvVPN]: 'PUSH_REP
LY,route 10.8.0.1 255.255.255.255,dhcp-option DNS 192.168.12.1,redirect-gateway
def1,route 10.8.0.0 255.255.255.0,topology net30,ping 10,ping-restart 120,ifconf
ig 10.8.0.6 10.8.0.5,peer-id 0,cipher AES-256-GCM' (status=1)
2025-11-28 09:54:56 SrvVPN/192.168.11.106:47989 Data Channel: cipher 'AES-256-GC
M', peer-id: 0
2025-11-28 09:54:56 SrvVPN/192.168.11.106:47989 Timers: ping 10, ping-restart 24
0
```

puis on install ryslogs

```
root@srvVpn:~# apt install rsyslog -y
Les paquets suivants ont été installés automatiquement et ne sont plus
es :
  libcbor0.8                libpython3.11-minimal   python3-jaraco.fun
  libcurl3t64-gnutls        libpython3.11-stdlib    python3-more-itert
```

un status pour regarder l'état

```
root@srvVpn:~# systemctl status rsyslog
● rsyslog.service - System Logging Service
     Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; preset:
     Active: active (running) since Fri 2025-11-28 11:38:39 CET; 9s ago
 Invocation: a952d2f368fc4370b26459a2909d716d
TriggeredBy: ● syslog.socket
       Docs: man:rsyslogd(8)
             man:rsyslog.conf(5)
             https://www.rsyslog.com/doc/
   Main PID: 1076 (rsyslogd)
      Tasks: 4 (limit: 2303)
     Memory: 1.8M (peak: 2.4M)
        CPU: 89ms
     CGroup: /system.slice/rsyslog.service
             └─1076 /usr/sbin/rsyslogd -n -iNONE

nov. 28 11:38:38 srvVpn systemd[1]: Starting rsyslog.service - System Logging S
nov. 28 11:38:39 srvVpn systemd[1]: Started rsyslog.service - System Logging Se
nov. 28 11:38:39 srvVpn rsyslogd[1076]: imuxsock: Acquired UNIX socket '/run/sy
```

puis dans le fichier rysylog.conf on ajoute cette ligne

```
*.emerg
*.* @192.168.13.20:514
```

puis dans le serveur rysylog apres d'un tail -f on aura ce beau résultat

```
root@SrvRsyslog:~# tail -f /var/log/srvVpn/rsyslogd.log
2025-11-28T11:39:46+01:00 srvVpn rsyslogd: [origin software="rsyslogd" swVersion
="8.2504.0" x-pid="1076" x-info="https://www.rsyslog.com"] exiting on signal 15.
2025-11-28T11:39:46+01:00 srvVpn rsyslogd: imuxsock: Acquired UNIX socket '/run/
systemd/journal/syslog' (fd 3) from systemd.  [v8.2504.0]
2025-11-28T11:39:46+01:00 srvVpn rsyslogd: [origin software="rsyslogd" swVersion
="8.2504.0" x-pid="1096" x-info="https://www.rsyslog.com"] start
c^C
root@SrvRsyslog:~# tail -f /var/log/srvVpn/
root.log      rsyslogd.log  systemd.log
```

Partie 3 FAIL2BAN

installation de fail2ban

```
root@SrvMail:~#  apt-get install fail2ban
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  python3-autocommand python3-inflect python3-jaraco.context python3-jaraco.functools
  python3-jaraco.text python3-more-itertools python3-pkg-resources python3-pyasyncore
  python3-pyinotify python3-setuptools python3-systemd python3-typeguard python3-typing-extensions
  python3-zipp whois
Paquets suggérés :
  monit sqlite3 python-pyinotify-doc python-setuptools-doc
Les NOUVEAUX paquets suivants seront installés :
  fail2ban python3-autocommand python3-inflect python3-jaraco.context python3-jaraco.functools
  python3-jaraco.text python3-more-itertools python3-pkg-resources python3-pyasyncore
  python3-pyinotify python3-setuptools python3-systemd python3-typeguard python3-typing-extensions
  python3-zipp whois
0 mis à jour, 16 nouvellement installés, 0 à enlever et 24 non mis à jour.
Il est nécessaire de prendre 1 861 kB dans les archives.
Après cette opération, 9 106 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n]
```

On copie le fichier de configuration principal de Fail2ban (`jail.conf`) vers un nouveau fichier nommé **`jail.local`** pour éviter qu'une mise à niveau écrase ce que j'ai fait

```
root@SrvMail:~# cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
root@SrvMail:~#
```

aussi l'installation de iptables

```
root@SrvMail:~# apt install iptables
Installation de :
  iptables
```

dans le fichier de configuration de fail2ban j'ai rajouter les information que je voulais pour permettre de ban une ip

```
[sshd]

# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and de
#mode   = normal
enabled = true
port    = 22
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
bantime = 60
findtime = 600
banaction = iptables-multiport
ignorereip = 192.168.14.100 192.168.11.100
backend = %(sshd_backend)s
```

on essaye plusieur tentative que l'on fait echoué

```
root@Srvdebianclone2:~# ssh sio@192.168.12.14 -p 22
sio@192.168.12.14's password:
Permission denied, please try again.
sio@192.168.12.14's password:
Permission denied, please try again.
sio@192.168.12.14's password:
sio@192.168.12.14: Permission denied (publickey,password).
root@Srvdebianclone2:~#
```

L'exécution de la commande **`fail2ban-client status sshd`** sur le serveur **SrvMail** a confirmé que la règle de protection SSH (**sshd**) est active et fonctionnelle.

```
             └─15793 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

nov. 06 11:23:04 SrvMail systemd[1]: Started fail2ban.service - Fail2Ban Se
nov. 06 11:23:05 SrvMail fail2ban-server[15793]: Server ready
root@SrvMail:~# fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     3
|  `- Journal matches:  _SYSTEMD_UNIT=ssh.service + _COMM=sshd
`- Actions
   |- Currently banned: 1
   |- Total banned:     1
   `- Banned IP list:   192.168.13.16
root@SrvMail:~#
```

Pour ensuite pouvoir envoyer un mail si une ip a etait bloque j’ai modifier les ligne destmail, sender, action

```
# Destination email address used solely for the interpolations in
# jail.{conf,local,d/*} configuration files.
destemail = SrvMail@menuimetal.fr

# Sender email address used solely for some actions
sender = fail2ban

# E-mail action. Since 0.8.1 Fail2Ban uses sendmail MTA for the
# mailing. Change mta configuration parameter to mail if you want to
# revert to conventional 'mail'.
mta = sendmail
```

```
# Choose default action.  To change, just override value of 'action' with the
# interpolation to the chosen action shortcut (e.g.  action_mw, action_mwl, etc) in jail.l
# globally (section [DEFAULT]) or per specific section
action = %(action_mwl)s


#
```