# OpenVPN

# Présentation d'OpenVPN

OpenVPN est une solution logicielle libre de **Réseau Privé Virtuel (VPN)** permettant de créer des tunnels sécurisés point à point ou site à site.
Contrairement aux vieux protocoles comme le **PPTP** de Microsoft (qui est à la sécurité ce qu'une passoire est à l'étanchéité), OpenVPN repose sur la bibliothèque **OpenSSL** pour assurer le chiffrement.

**Multiplateforme** : Fonctionne sur Linux, Windows, macOS, Android et iOS.

**Basé sur la PKI** : Utilise des certificats et des clés pour l'authentification mutuelle du serveur et du client.

**Sécurité** : Utilise l'autorisation et le chiffrement pour connecter des hôtes externes en toute sécurité.

# Les Modes de Fonctionnement

Le choix de la topologie est la décision la plus critique lors de la conception d'une infrastructure VPN. On distingue principalement deux approches basées sur le modèle OSI.

## Le Mode Routing (TUN) - Couche 3 (Réseau)

C’est la configuration la plus répandue pour les accès distants et l’interconnexion de sites.

**Fonctionnement** : On utilise des interfaces virtuelles de type **TUN** (Network TUNnel) qui opèrent au niveau de la couche réseau.

**Segmentation :** On définit un sous-réseau spécifique pour le VPN. Le client accède au sous-réseau distant via le routage IP, mais n'a pas forcément accès à l'intégralité d'Internet via le tunnel par défaut.

**Performance :** Ce mode est plus performant car il limite l'overhead en ne transmettant que les paquets IP, excluant le trafic de diffusion (broadcast).

## Le Mode Bridge (TAP) - Couche 2 (Liaison)

**Fonctionnement :** On utilise des interfaces **TAP** (Ethernet Tunnel) opérant au niveau de la couche liaison de données.

**Extension de segment :** On lie deux sites comme s'ils appartenaient à un seul segment Ethernet physique.

**Compatibilité :** On peut supporter tous les protocoles bâtis au-dessus d'Ethernet, facilitant la communication pour des périphériques ne supportant que le TCP/IP simplifié.

**Avantage** : Permet de voir les périphériques en broadcast comme s'ils étaient sur le même hub.

**Inconvénient** : Augmentation massive du trafic broadcast, ce qui peut saturer ta bande passante.

# Topologies de Déploiement

Selon les besoins d'interconnexion, on déploiera OpenVPN selon deux schémas types :

## VPN Nomade (Accès distant)

Un hôte autorisé se connecte à une passerelle de sécurité (VPN Gateway) via Internet. Le tunnel chiffre la communication entre le client et le réseau interne, mais le trafic non destiné au LAN reste généralement non chiffré sur Internet s'il ne passe pas par le VPN.

## VPN Site à Site

On interconnecte deux infrastructures fixes (ex: un siège social et une agence). On utilise un routeur local pour rediriger les requêtes vers une passerelle OpenVPN qui gère l'encapsulation et le chiffrement. À l'autre extrémité, le serveur VPN déchiffre le trafic et le transmet sur le réseau local cible.

# Infrastructure à Clé Publique (PKI)

La sécurité d'OpenVPN repose sur l'utilisation de certificats et de clés pour l'authentification mutuelle. On s'appuie sur une structure hiérarchique pour garantir l'intégrité de la solution.

**CA (Autorité de Certification) :** L'entité de confiance qui signe les certificats du serveur et des clients.

**RA (Autorité d'Enregistrement) :** Elle vérifie l'identité des demandeurs de certificats.

**VA (Autorité de Validation) :** Elle permet de vérifier la validité d'un certificat (notamment via les listes de révocation).

# Processus de sécurisation

Pour simplifier la gestion, on utilise souvent l'outil **Easy-RSA** pour automatiser les actions récurrentes.

**Initialisation :** On définit les variables par défaut (Pays, Organisation, etc.) dans un fichier vars pour assurer l'homogénéité des certificats.

**Génération de la CA :** On crée la clé privée (ca.key) et le certificat public (ca.crt) de l'autorité.

**Échange Diffie-Hellman :** On génère les paramètres DH pour permettre un échange de clés sécurisé sur un canal non sécurisé.

**Certificats d'entité :** On génère des requêtes de signature de certificat (CSR) pour le serveur et chaque client, que l'on signe ensuite avec la CA.

> **Note de sécurité :** Si la clé privée de la CA est compromise, l'intégralité de la confiance de l'infrastructure est rompue. Elle doit être conservée hors ligne ou protégée par une phrase secrète (Passphrase) forte.

# AUDIT ET DÉBOGAGE OPENVPN

Lorsqu'un tunnel VPN refuse de s'établir, on doit procéder par élimination en suivant les couches du modèle OSI. On ne panique pas, on analyse.

## Audit de la Couche Service et Système

Avant de chercher une erreur complexe, on vérifie si le "moteur" tourne et si le système autorise le passage des paquets.

### Vérification du service

On utilise les outils standards de gestion de services pour vérifier l'état du serveur:

**Commande :** `sudo systemctl status openvpn-server@server.service`

**Indicateur :** Le statut doit être active (running).

### Vérification du transfert IP (IP Forwarding)

Le routage VPN ne peut fonctionner si le noyau Linux bloque le transfert de paquets entre les interfaces.

**Vérification :** On consulte la valeur de `net.ipv4.ip_forward`

**Application :** Si elle est à 0, on l'active via `sudo sysctl -p` après modification du fichier `/etc/sysctl.conf`

## Audit de la PKI (Le "Cimetière" des Connexions)

La majorité des échecs de connexion OpenVPN provient d'un certificat mal signé, expiré ou d'une clé privée mal placée.

### Vérification de la validité des certificats

On utilise OpenSSL pour valider que le certificat du serveur ou du client est bien reconnu par l'Autorité de Certification (CA):

**Commande :** `openssl verify -CAfile pki/ca.crt pki/issued/user-vpn.lab-unsafe.lan.crt`

**Résultat attendu :** OK.

> **Note :** Si vous obtenez une erreur de type "Verification error: unable to get local issuer certificate", c'est que vous essayez d'utiliser un certificat signé par une CA que vous n'avez pas fournie. C'est le statut "Excommunicado" immédiat pour votre sécurité.

# Audit Réseau et Résolution

Si le service tourne et que les clés sont bonnes, le problème est "dans les tuyaux".

## Analyse des interfaces et des routes

On doit identifier l'interface publique pour s'assurer que les règles de firewall (nftables) pointent au bon endroit.

**Identification :** `ip route list default` permet de voir quelle interface (ex: enp0s3) porte la passerelle par défaut.

## Diagnostic DNS

Souvent, le tunnel monte mais "Internet ne marche plus". On vérifie alors la configuration de la résolution de noms.

**Vérification :** `cat /etc/resolv.conf`

**Indice :** Sur les systèmes utilisant systemd-resolved, l'adresse doit être 127.0.0.53.

> **Note :** Si après tout ça, ça ne marche toujours pas, vérifiez si vous n'avez pas simplement oublié d'ouvrir le port UDP 1194 sur votre routeur. C'est l'équivalent de chercher ses clés pendant une heure alors qu'elles sont sur la porte.

# INSTALLATION ET MISE EN PLACE DE LA PKI

La gestion des clés et des certificats étant une opération complexe, on utilise easy-rsa pour automatiser les actions récurrentes comme la génération de l'autorité de certification et des certificats d'entité.

## Préparation de l'environnement (sur le server)

On commence par installer les paquets nécessaires sur le serveur. OpenVPN s'appuie sur la bibliothèque OpenSSL pour le chiffrement.

```
admin@lab-debian-39:~$ sudo apt-get install openvpn easy-rsa
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libccid libduktape207 libeac3 liblzo2-2 libnl-3-200 libnl-genl-3-200 libpcsclite1 libpkcs11-helper1t64
libpolkit-agent-1-0 libpolkit-gobject-1-0
  opensc opensc-pkcs11 pcscd polkitd sgml-base xml-core
Suggested packages:
  pcmciautils resolvconf openvpn-dco-dkms openvpn-systemd-resolved sgml-base-doc debhelper
The following NEW packages will be installed:
  easy-rsa libccid libduktape207 libeac3 liblzo2-2 libnl-3-200 libnl-genl-3-200 libpcsclite1 libpkcs11-
helper1t64 libpolkit-agent-1-0
  libpolkit-gobject-1-0 opensc opensc-pkcs11 openvpn pcscd polkitd sgml-base xml-core
0 upgraded, 18 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,902 kB of archives.
After this operation, 9,438 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

## Création du répertoire de travail

On utilise un script pour créer une structure de répertoires dédiée à la CA afin de ne pas impacter d'autres configurations.

```
admin@lab-debian-39:~$ make-cadir ~/openvpn-ca
admin@lab-debian-39:~$ ls -l
total 4
drwx------ 2 admin admin 4096 Jan 27 10:45 openvpn-ca
admin@lab-debian-39:~$ cd openvpn-ca/
admin@lab-debian-39:~/openvpn-ca$
```

# Configuration des variables par défaut

On initialise le fichier vars à partir d'un modèle pour définir les informations d'identité qui seront communes à tous les certificats.

```
admin@lab-debian-39:~/openvpn-ca$ cp /usr/share/easy-rsa/vars.example vars
```

```
admin@lab-debian-39:~/openvpn-ca$ nano vars
```

# On édite ensuite ce fichier pour renseigner les variables telles que `EASYRSA_REQ_COUNTRY`, `EASYRSA_REQ_ORG`, ou `EASYRSA_REQ_EMAIL`

```
# Organizational fields (used with "org" mode and ignored in "cn_only" mode).
# These are the default values for fields which will be placed in the
# certificate.  Do not leave any of these fields blank, although interactively
# you may omit any specific field by typing the "." symbol (not valid for
# email).
#
# NOTE: The following characters are not supported
#       in these "Organizational fields" by Easy-RSA:
#       back-tick (`)
#
set_var EASYRSA_REQ_COUNTRY     "FR"
set_var EASYRSA_REQ_PROVINCE    "Occitanie"
set_var EASYRSA_REQ_CITY        "Auzat"
set_var EASYRSA_REQ_ORG         "CA OpenVPN lab-unsafe.lan"
set_var EASYRSA_REQ_EMAIL       "me@lab-unsafe.lan"
#set_var EASYRSA_REQ_OU         "My Organizational Unit"
```

# Initialisation de la PKI et création de la CA

Cette phase est cruciale : on prépare l'environnement technique et on crée l'entité de confiance suprême.

**Initialisation de la PKI :** Cette commande crée la structure de dossiers, notamment le répertoire pki.

```
admin@lab-debian-39:~/openvpn-ca$ ./easyrsa init-pki
Using Easy-RSA 'vars' configuration:
* /home/admin/openvpn-ca/vars

Notice
------
'init-pki' complete; you may now create a CA or requests.

Your newly created PKI dir is:
* /home/admin/openvpn-ca/pki

Using Easy-RSA configuration:
* /home/admin/openvpn-ca/vars
```

```
admin@lab-debian-39:~/openvpn-ca$ ls -l pki/
total 20
drwx------ 2 admin admin 4096 Jan 27 10:53 issued
drwx------ 2 admin admin 4096 Jan 27 10:53 private
drwx------ 2 admin admin 4096 Jan 27 10:53 reqs
-rw------- 1 admin admin 5826 Jan 27 10:53 vars.example
```

# Construction de l'Autorité de Certification (CA)

On génère ici le certificat public de la CA (ca.crt) et sa clé privée (ca.key).

```
admin@lab-debian-39:~/openvpn-ca$ ./easyrsa build-ca
Using Easy-RSA 'vars' configuration:
* /home/admin/openvpn-ca/vars

Enter New CA Key Passphrase:

Confirm New CA Key Passphrase:
.+...+++++++++++++++++++++++++++++++++++++*.+........+.......+...........++++++++++++++++++++++++++++++++++
+++*.+.........+..........+..+.+..+.............+.+......+..................+.....+..........
+.................+.+.....+.+......+..+.+...........+................+..+....+.....+..........+...+........
+.........+...+.......+........+......+.......+..+..........+...+.........+......+...+.....+......+.+..
+..........+........+............+......+...++++++
....+...+...+...+..+..........++++++++++++++++++++++++++++++++++++*......+....+...+.....+.......++++++++++++
++++++++++++++++++++++++*..+...+........+.+........+...+...+.........+...+....+......+...+..+...+......+.
+.....+............+..+..........+...+..............+......+..................+...+........+....+......
+............+...+..+.......+..................+...+...+....+..+...+.+.....+..........++++++
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
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_LAB-UNSAFE_OpenVPN

Notice
------
CA creation complete. Your new CA certificate is at:
* /home/admin/openvpn-ca/pki/ca.crt

Create an OpenVPN TLS-AUTH|TLS-CRYPT-V1 key now: See 'help gen-tls'

Build-ca completed successfully.
```

**Note :** Il est impératif de définir une passphrase robuste pour la clé de la CA, car elle sera nécessaire pour signer toutes les futures requêtes.

# Génération du certificat et de la clé du serveur

On crée maintenant l'identité numérique du serveur **lab-debian-39**.

## Génération de la requête (CSR)

On utilise l'option nopass pour éviter que le serveur ne demande un mot de passe à chaque redémarrage du service.

```
admin@lab-debian-39:~/openvpn-ca$ ./easyrsa gen-req lab-debian-39 nopass
Using Easy-RSA 'vars' configuration:
* /home/admin/openvpn-ca/vars
...........+...+..........+.....+.+......++++++++++++++++++++++++++++++++++++++*.................+.+...+.....
+......+.+.....+....+...........+.......+...+..+.+..+.......++++++++++++++++++++++++++++++++++++++++*..+..++++++
..........+......+.........++++++++++++++++++++++++++++++++++++*...+.+.....+++++++++++++++++++++++++++++++++
++++++*...+...+...+....+..+..................+..+.......+.........+..+...+.+...+...+..+.+.........+...........+.
+........+...........+.........+...+......+.....+.......+...+..+.......+..+......+.......+......+...........
+..+......+................+.......+.........+...+...+.....+.........+....+......+.....+...+...+.......+...+...
+......+........+.............+.........+...+.....+............+.........+....+...+..+...
+................................+...+............+.....++++++
-----
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) [lab-debian-39]:

Notice
------
Private-Key and Public-Certificate-Request files created.
Your files are:
* req: /home/admin/openvpn-ca/pki/reqs/lab-debian-39.req
* key: /home/admin/openvpn-ca/pki/private/lab-debian-39.key
```

# Signature du certificat par la CA

On demande à notre CA de valider l'identité du serveur.

```
admin@lab-debian-39:~/openvpn-ca$ ./easyrsa sign-req server lab-debian-39
Using Easy-RSA 'vars' configuration:
* /home/admin/openvpn-ca/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:     'lab-debian-39'
  Requested type:   'server'
  Valid for:        '825' days


subject=
    commonName                = lab-debian-39

Type the word 'yes' to continue, or any other input to abort.
  Confirm requested details: yes                                    <-- YES !

Using configuration from /home/admin/openvpn-ca/pki/e80c85f7/temp.1.1
Enter pass phrase for /home/admin/openvpn-ca/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'lab-debian-39'
Certificate is to be certified until May  1 10:05:18 2028 GMT (825 days)

Write out database with 1 new entries
Database updated

Notice
------
Inline file created:
* /home/admin/openvpn-ca/pki/inline/private/lab-debian-39.inline


Notice
------
Certificate created at:
* /home/admin/openvpn-ca/pki/issued/lab-debian-39.crt
```

# Génération du certificat et de la clé du client

On répète l'opération pour le client **lab-debian-41**

## Génération de la requête pour le client

```
admin@lab-debian-39:~/openvpn-ca$ ./easyrsa gen-req lab-debian-41 nopass
Using Easy-RSA 'vars' configuration:
* /home/admin/openvpn-ca/vars
....+....+.....+.+.....+.....+.....+.....+...+...+..+...+.+++++++++++++++++++++++++++++++
+++++++*..+++++++++++++++++++++++++++++++++++++*....+..........+.......+.........+...+...+..
+.................+..+.................+....+...+...........+.....+...+....+.....+...+.......+...+...+..
+......+....+..........+..........+.....+......+...+..........+........+..........+.....+....+.....+....+..
+...+.+...+...+.................+.+...........+.+.....+...+......+.+...........+.................+......++++++
...+...+..+.+..+.........+...+......+....+...+.........+......+...+...........+..++++++++++++++++++++++++++
++++++++++*.++++++++++++++++++++++++++++++++++++*......+..+...+.......+........+.+.....+.......+.....+....
+...+...........+......+...+.+..+...+......+.+.....+.........+.+.....+..........+.........+...........+....+...
+...+.....+...+.+......+..+.+..............+......++++++
-----
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) [lab-debian-41]:

Notice
------
Private-Key and Public-Certificate-Request files created.
Your files are:
* req: /home/admin/openvpn-ca/pki/reqs/lab-debian-41.req
* key: /home/admin/openvpn-ca/pki/private/lab-debian-41.key
```

# Signature par la CA

Le certificat résultant sera utilisé par le client pour s'authentifier auprès du serveur.

```
admin@lab-debian-39:~/openvpn-ca$ ./easyrsa sign-req client lab-debian-41
Using Easy-RSA 'vars' configuration:
* /home/admin/openvpn-ca/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:     'lab-debian-41'
  Requested type:   'client'
  Valid for:        '825' days


subject=
    commonName                = lab-debian-41

Type the word 'yes' to continue, or any other input to abort.
  Confirm requested details: yes                                    <-- YES !

Using configuration from /home/admin/openvpn-ca/pki/f2d21feb/temp.1.1
Enter pass phrase for /home/admin/openvpn-ca/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'lab-debian-41'
Certificate is to be certified until May  1 10:08:15 2028 GMT (825 days)

Write out database with 1 new entries
Database updated

Notice
------
Inline file created:
* /home/admin/openvpn-ca/pki/inline/private/lab-debian-41.inline


Notice
------
Certificate created at:
* /home/admin/openvpn-ca/pki/issued/lab-debian-41.crt
```

# Paramètres de Diffie-Hellman

Pour sécuriser l'échange de clés initial entre le serveur et le client sur un canal non sécurisé, on génère les paramètres de Diffie-Hellman.

## Génération du fichier DH

```
admin@lab-debian-39:~/openvpn-ca$ ./easyrsa gen-dh
Using Easy-RSA 'vars' configuration:
* /home/admin/openvpn-ca/vars
Generating DH parameters, 2048 bit long safe prime
.....................................................................................................
.....................................................................................
+....................................................................................................
....................................................................................................
.....................................................................
+.....................................................…

...

+....................................................................................................
....................................................................................................
.....................................................................................
+.....................................................
+....................................................................................................
+.....................................................+.................+.........................
+...................
+....................................................................................................
....................................................................................................
...........+.....................................................................
+....................................................................................................
.....................................................
+.....................................................
+.....................................................................+.....................................
+.......................
+.....................................................................................
+....................................................................................................
.............+.........................+.......
+....................................................................................................
+.............................................+................+...........
+....................................................................................................
.....................................................................
+....................................................................................................
.............+.........+........+..............................+.....................................
+...................
+....................................................................................................
.....................................................
+....................................................................................................
..+.......................................+.................................+.........
+....................................................................................................
....................................................................................................
....................................................................................................
....................................................................................................
............................+.....................................+.....................................
+....................................................................................................
+.....................................................................................
+.....................................................................................
+.....................................................................................
+....................................................................................................
.............................+......................................................................
+....................................................
+.....................................................
+.....................................................++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*+
+*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*++*+
+*++*++*++*++*++*++*++*++*
DH parameters appear to be ok.

Notice
------

DH parameters of size 2048 created at:
* /home/admin/openvpn-ca/pki/dh.pem
```

# Le fichier est généré par défaut avec une taille de 2048 bits dans le répertoire pki/dh.pem

# Organisation des Fichiers de Sécurité

Les certificats que tu as générés sont actuellement dans le répertoire personnel de l'utilisateur admin. Pour que le service système OpenVPN puisse les lire, on doit les déplacer dans le sanctuaire : `/etc/openvpn/server/`

```
admin@lab-debian-39:~/openvpn-ca$ sudo cp pki/ca.crt /etc/openvpn/server/
admin@lab-debian-39:~/openvpn-ca$ sudo cp pki/dh.pem /etc/openvpn/server/
```

# Copie du certificat de la CA et des paramètres DH

```
admin@lab-debian-39:~/openvpn-ca$ sudo cp pki/issued/lab-debian-39.crt /etc/openvpn/server/
admin@lab-debian-39:~/openvpn-ca$ sudo cp pki/private/lab-debian-39.key /etc/openvpn/server/
```

# Copie de l'identité du serveur (Certificat et Clé privée)

```
admin@lab-debian-39:~/openvpn-ca$ ls -l /etc/openvpn/server/
total 20
-rw------- 1 root root 1245 Jan 27 11:17 ca.crt
-rw------- 1 root root  428 Jan 27 11:17 dh.pem
-rw------- 1 root root 4575 Jan 27 11:19 lab-debian-39.crt
-rw------- 1 root root 1704 Jan 27 11:19 lab-debian-39.key
```

# **Vérification :** Assure-toi que les droits sont restrictifs sur la clé privée (.key)

# Configuration du service OpenVPN

On utilise le modèle officiel pour créer notre fichier de configuration.

```
admin@lab-debian-39:~/openvpn-ca$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf
/etc/openvpn/server/server.conf
```

# Copie du fichier d'exemple

```
admin@lab-debian-39:~/openvpn-ca$ sudo nano /etc/openvpn/server/server.conf
```

```
# If you do not want to maintain a CA
# and have a small number of clients
# you can also use self-signed certificates
# and use the peer-fingerprint option.
# See openvpn-examples man page for a
# configuration example.
ca ca.crt
cert lab-debian-39.crt
key lab-debian-39.key  # This file should be kept secret

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

...

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this on non-Windows
# systems after creating a dedicated user.
user nobody
group nogroup
```

# Dans le fichier, assure-toi que ces lignes sont correctes (ca, cert, key, dh)

# Et on décommente user nobody et group nogroup pour "enfermer" le service dans un environnement à privilèges restreints. C'est ce qu'on appelle le **principe du moindre privilège**.

# Vérification

```
admin@lab-debian-39:~/openvpn-ca$ ls -l /etc/openvpn/server/
total 32
-rw------- 1 root root  1245 Jan 27 11:17 ca.crt
-rw------- 1 root root   428 Jan 27 11:17 dh.pem
-rw------- 1 root root  4575 Jan 27 11:19 lab-debian-39.crt
-rw------- 1 root root  1704 Jan 27 11:19 lab-debian-39.key
-rw-r--r-- 1 root root 10780 Jan 27 11:32 server.conf
```

# Activation du routage (IP Forwarding)

Le serveur doit accepter de faire passer les paquets d'une interface à l'autre.

```
admin@lab-debian-39:~$ echo "net.ipv4.ip_forward = 1" | sudo tee /etc/sysctl.d/99-openvpn.conf
net.ipv4.ip_forward = 1
```

# Création manuelle du paramètre dans le répertoire sysctl.d

```
admin@lab-debian-39:~$ sudo sysctl -p /etc/sysctl.d/99-openvpn.conf
net.ipv4.ip_forward = 1
```

# Application immédiate sans redémarrage

# Vérification de l'état (La preuve par l'image)

```
admin@lab-debian-39:~$ cat /proc/sys/net/ipv4/ip_forward
1
```

> **Note :** Si la commande cat te renvoie 1, alors le noyau accepte maintenant de transférer les paquets. Ton serveur vient de gagner sa fonction de "passerelle".

# Configuration du Firewall (Masquerade)

On utilise nftables pour que le trafic provenant du VPN soit "masqué" par l'IP du serveur lors de sa sortie vers l'extérieur.

# Création du fichier de configuration

```
admin@lab-debian-39:~$ sudo nano /etc/nftables.conf
```

# Ajout des chaînes nécessaires

```
#!/usr/sbin/nft -f

flush ruleset

# Table pour le NAT (Masquerade)
# C'est ce qui permet au VPN de sortir vers la DMZ
table ip routingv4 {
    chain prerouting {
        type nat hook prerouting priority 0; policy accept;
    }

    chain postrouting {
        type nat hook postrouting priority 0; policy accept;
        # La régle magique qui cache l'IP du client VPN derrière l'IP du serveur
        masquerade
    }
}
```

# Activation

```
admin@lab-debian-39:~$ sudo systemctl enable --now nftables
```

# Démarrage et Validation

C'est le moment de vérité. On active le service pour qu'il survive au redémarrage et on le lance.

# Activation et démarrage

```
admin@lab-debian-39:~$ sudo systemctl enable --now openvpn-server@server.service
Created symlink '/etc/systemd/system/multi-user.target.wants/openvpn-server@server.service' →
'/usr/lib/systemd/system/openvpn-server@.service'.
```

# Vérification du statut

```
admin@lab-debian-39:~$ sudo systemctl status openvpn-server@server.service
● openvpn-server@server.service - OpenVPN service for server
     Loaded: loaded (/usr/lib/systemd/system/openvpn-server@.service; enabled; preset: enabled)
     Active: activating (auto-restart) (Result: exit-code) since Tue 2026-01-27 11:43:43 CET; 2s ago
 Invocation: 25db9e73517b47b89ca569ec436af153
       Docs: man:openvpn(8)
             https://openvpn.net/community-resources/reference-manual-for-openvpn-2-6/
             https://community.openvpn.net/openvpn/wiki/HOWTO
    Process: 2589 ExecStart=/usr/sbin/openvpn --status /run/openvpn-server/status-server.log --status-version 2
--suppress-timestamps --config server>
   Main PID: 2589 (code=exited, status=1/FAILURE)
     Status: "Pre-connection initialization successful"
   Mem peak: 2.1M
        CPU: 18ms
```

> **Note :** Si le statut est active (running), le tunnel est créé. Tu devrais voir une nouvelle interface tun0 en tapant ip addr.

```
admin@lab-debian-39:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:3f:55:ae brd ff:ff:ff:ff:ff:ff
    altname enx5254003f55ae
    inet 192.168.100.138/24 brd 192.168.100.255 scope global dynamic noprefixroute enp1s0
       valid_lft 2533sec preferred_lft 1898sec
    inet6 fe80::44af:efd6:de8f:7008/64 scope link
       valid_lft forever preferred_lft forever
96: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen
500
    link/none
    inet 10.8.0.1/24 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::7b90:1689:e3e:35f8/64 scope link stable-privacy proto kernel_ll
       valid_lft forever preferred_lft forever
```

# Audit Final

Si le service est bien lancé, on vérifie que le serveur "écoute" bien les clients qui vont arriver sur le port UDP 1194.

```
admin@lab-debian-39:~$ ss -tunlp | grep 1194
udp   UNCONN 0      0                                  0.0.0.0:1194      0.0.0.0:*
```

# Création du profil client (Sur le server)

On va faciliter la vie de l'utilisateur (nous-mêmes) en créant un fichier de configuration "tout-en-un" qui contient les certificats directement à l'intérieur.

```
admin@lab-debian-39:~$ mkdir -p client-configs/files
admin@lab-debian-39:~$ cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf
client-configs/base.conf
```

# Édition du modèle

```
admin@lab-debian-39:~$ nano client-configs/base.conf
```

```
# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 192.168.100.138 1194
;remote my-server-2 1194

...

# Downgrade privileges after initialization (non-Windows only)
user nobody
group nogroup

...

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

| | |
|---|---|
| **Proto** | Vérifie que c'est bien `proto udp` |
| **Remote** | Remplace `my-server-1` par l'IP de ton serveur `192.168.100.138` et le port `1194` |
| **User/Group** | Décommente `user nobody` et `group nogroup` |
| **Certificats** | Commente les lignes `ca`, `cert` et `key` |

# Automatisation du fichier .ovpn

Pour éviter de copier 4 fichiers sur le client, on utilise un script de concaténation.

```
admin@lab-debian-39:~$ nano client-configs/make_config.sh
```

```
#!/bin/bash
# Identifiant du client (ex: lab-debian-41)

KEY_DIR=~/openvpn-ca/pki
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>') \
    ${KEY_DIR}/issued/${1}.crt \
    <(echo -e '</cert>\n<key>') \
    ${KEY_DIR}/private/${1}.key \
    <(echo -e '</key>') \
    > ${OUTPUT_DIR}/${1}.ovpn
```

```
admin@lab-debian-39:~$ chmod +x client-configs/make_config.sh
admin@lab-debian-39:~$ ./client-configs/make_config.sh lab-debian-41
```

# Vérification

```
admin@lab-debian-39:~$ ls -l client-configs/
total 12
-rw-r--r-- 1 admin admin 3447 Jan 27 12:08 base.conf
drwxrwxr-x 2 admin admin 4096 Jan 27 12:12 files
-rwxrwxr-x 1 admin admin  415 Jan 27 12:11 make_config.sh
admin@lab-debian-39:~$ ls -l client-configs/files/
total 12
-rw-rw-r-- 1 admin admin 10992 Jan 27 12:12 lab-debian-41.ovpn
```

# Préparation du client (Sur le client)

Avant de lui envoyer les clés, on installe le moteur sur la machine cliente.

```
admin@lab-debian-41:~$ sudo apt update && sudo apt install openvpn
Get:1 http://security.debian.org/debian-security trixie-security InRelease [43.4 kB]
Hit:2 http://deb.debian.org/debian trixie InRelease
Get:3 http://deb.debian.org/debian trixie-updates InRelease [47.3 kB]
Get:4 http://security.debian.org/debian-security trixie-security/main Sources [119 kB]
Get:5 http://security.debian.org/debian-security trixie-security/main amd64 Packages [97.9 kB]
Fetched 308 kB in 0s (1,560 kB/s)
All packages are up to date.
Installing:
  openvpn

Installing dependencies:
  easy-rsa  libduktape207  liblzo2-2    libnl-genl-3-200  libpkcs11-helper1t64  libpolkit-gobject-1-0  opensc-
pkcs11  polkitd    xml-core
  libccid   libeac3        libnl-3-200  libpcsclite1      libpolkit-agent-1-0   opensc                 pcscd
sgml-base

Suggested packages:
  pcmciautils  resolvconf  openvpn-dco-dkms  openvpn-systemd-resolved  sgml-base-doc  debhelper

Summary:
  Upgrading: 0, Installing: 18, Removing: 0, Not Upgrading: 0
  Download size: 2,902 kB
  Space needed: 9,438 kB / 27.1 GB available

Continue? [Y/n]
```

# Transfert du fichier .ovpn

Depuis ton serveur, nous allons utiliser scp (Secure Copy) pour envoyer le fichier unifié vers le client.

```
admin@lab-debian-39:~$ scp ~/client-configs/files/lab-debian-41.ovpn admin@192.168.100.126:~/
The authenticity of host '192.168.100.126 (192.168.100.126)' can't be established.
ED25519 key fingerprint is SHA256:5EDP6Vq85vhLkp7WS3VJgljDPS9nQyvpaWXLCuF7NNA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.126' (ED25519) to the list of known hosts.
admin@192.168.100.126's password:
lab-debian-41.ovpn
100%   11KB  22.6MB/s   00:00
```

# Vérification sur le Client

```
admin@lab-debian-41:~$ ls -l
total 12
-rw-rw-r-- 1 admin admin 10992 Jan 27 12:17 lab-debian-41.ovpn
```

# Connexion (Sur le client)

Une fois le fichier reçu, il ne reste plus qu'à lancer le tunnel.

Contrairement au serveur qui tourne en arrière-plan via systemd, on va d'abord le lancer manuellement pour auditer la connexion.

```
admin@lab-debian-41:~$ sudo openvpn --config lab-debian-41.ovpn
2026-01-27 12:20:37 Note: --cipher is not set. OpenVPN versions before 2.5 defaulted to BF-CBC as fallback when
cipher negotiation failed in this case. If you need this fallback please add '--data-ciphers-fallback BF-CBC'
to your configuration and/or add BF-CBC to --data-ciphers.
2026-01-27 12:20:37 Note: Kernel support for ovpn-dco missing, disabling data channel offload.
2026-01-27 12:20:37 OpenVPN 2.6.14 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11]
[MH/PKTINFO] [AEAD] [DCO]
2026-01-27 12:20:37 library versions: OpenSSL 3.5.4 30 Sep 2025, LZO 2.10
2026-01-27 12:20:37 DCO version: N/A
2026-01-27 12:20:37 TCP/UDP: Preserving recently used remote address: [AF_INET]192.168.100.138:1194
2026-01-27 12:20:37 Socket Buffers: R=[212992->212992] S=[212992->212992]
2026-01-27 12:20:37 UDPv4 link local: (not bound)
2026-01-27 12:20:37 UDPv4 link remote: [AF_INET]192.168.100.138:1194
2026-01-27 12:20:37 NOTE: UID/GID downgrade will be delayed because of --client, --pull, or --up-delay
2026-01-27 12:20:37 TLS: Initial packet from [AF_INET]192.168.100.138:1194, sid=76858c6b b58dd76d
2026-01-27 12:20:37 VERIFY OK: depth=1, CN=CA_LAB-UNSAFE_OpenVPN
2026-01-27 12:20:37 VERIFY KU OK
2026-01-27 12:20:37 Validating certificate extended key usage
2026-01-27 12:20:37 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server
Authentication
2026-01-27 12:20:37 VERIFY EKU OK
2026-01-27 12:20:37 VERIFY OK: depth=0, CN=lab-debian-39
2026-01-27 12:20:37 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, peer certificate: 2048
bits RSA, signature: RSA-SHA256
2026-01-27 12:20:37 [lab-debian-39] Peer Connection Initiated with [AF_INET]192.168.100.138:1194
2026-01-27 12:20:37 TLS: move_session: dest=TM_ACTIVE src=TM_INITIAL reinit_src=1
2026-01-27 12:20:37 TLS: tls_multi_process: initial untrusted session promoted to trusted
2026-01-27 12:20:37 PUSH: Received control message: 'PUSH_REPLY,route-gateway 10.8.0.1,topology subnet,ping
10,ping-restart 120,ifconfig 10.8.0.2 255.255.255.0,peer-id 0,cipher AES-256-GCM,protocol-flags cc-exit tls-ekm
dyn-tls-crypt,tun-mtu 1500'
2026-01-27 12:20:37 OPTIONS IMPORT: --ifconfig/up options modified
2026-01-27 12:20:37 OPTIONS IMPORT: route-related options modified
2026-01-27 12:20:37 OPTIONS IMPORT: tun-mtu set to 1500
2026-01-27 12:20:37 TUN/TAP device tun0 opened
2026-01-27 12:20:37 net_iface_mtu_set: mtu 1500 for tun0
2026-01-27 12:20:37 net_iface_up: set tun0 up
2026-01-27 12:20:37 net_addr_v4_add: 10.8.0.2/24 dev tun0
2026-01-27 12:20:37 UID set to nobody
2026-01-27 12:20:37 GID set to nogroup
2026-01-27 12:20:37 Capabilities retained: CAP_NET_ADMIN
2026-01-27 12:20:37 Initialization Sequence Completed                <-- OK !
2026-01-27 12:20:37 Data Channel: cipher 'AES-256-GCM', peer-id: 0
2026-01-27 12:20:37 Timers: ping 10, ping-restart 120
2026-01-27 12:20:37 Protocol options: protocol-flags cc-exit tls-ekm dyn-tls-crypt
```

**Note :** Garde ce terminal ouvert. Si tu vois la ligne `Initialization Sequence Completed`, c'est que tu as officiellement hacké la réalité locale. Tu es dans le tunnel.

# Validation et Audit Final

On ouvre un **deuxième terminal** sur **le client** pendant que le VPN tourne

## Vérification de l'interface

```
admin@lab-debian-41:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:c3:ad:be brd ff:ff:ff:ff:ff:ff
    altname enx525400c3adbe
    inet 192.168.100.126/24 brd 192.168.100.255 scope global dynamic noprefixroute enp1s0
       valid_lft 2436sec preferred_lft 1787sec
    inet6 fe80::4023:c905:f0a1:7a21/64 scope link
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.2/24 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::1413:924:cd27:8247/64 scope link stable-privacy proto kernel_ll
       valid_lft forever preferred_lft forever
```

## Test de connectivité

On teste si le client peut "toucher" le serveur via l'adresse IP privée du tunnel.

```
admin@lab-debian-41:~$ ping 10.8.0.1 -c 3
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=0.486 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=0.620 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=1.07 ms

--- 10.8.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.486/0.724/1.066/0.247 ms
```

## Audit des routes

Vérifie que ton système sait maintenant que pour aller vers le réseau VPN, il doit passer par le tunnel.

```
admin@lab-debian-41:~$ ip route
default via 192.168.100.1 dev enp1s0 proto dhcp src 192.168.100.126 metric 1002
10.8.0.0/24 dev tun0 proto kernel scope link src 10.8.0.2
192.168.100.0/24 dev enp1s0 proto dhcp scope link src 192.168.100.126 metric 1002
```