Compte rendu TD Serveur VPN

Table des matières

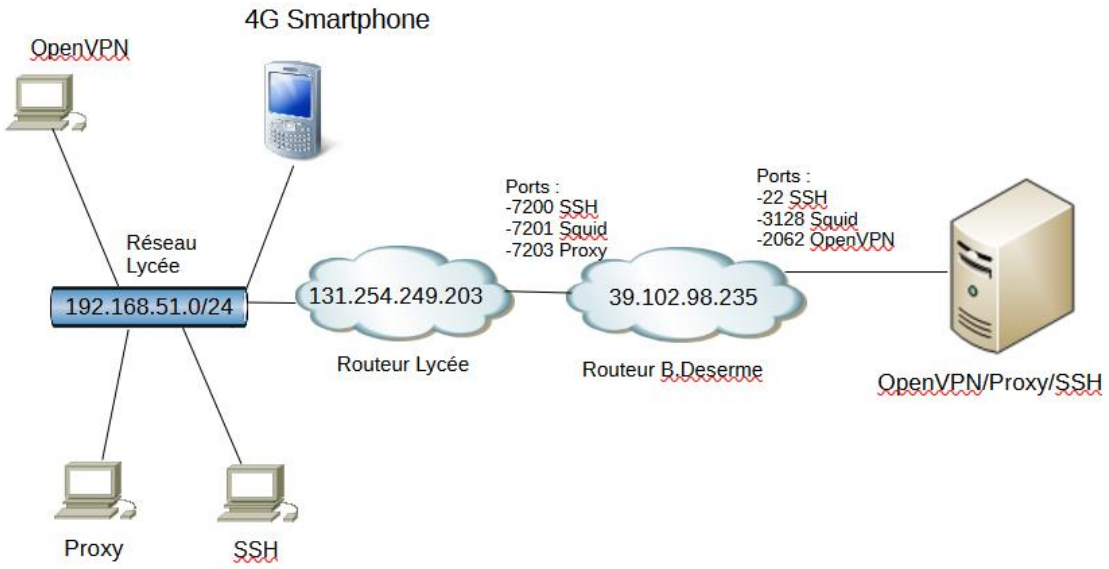
Mise en Œuvre :

1. Tableau Comparatif :

Caractéristique	OpenVPN	IPsec	WireGuard	L2TP/IPsec
Type	Open-source	Standardisé	Nouveau	Tunneling
Flexibilité	Très élevée	Moyenne	Moyenne	Limitée
Sécurité	Robuste	Solide	En cours de vérification	Robuste, quelques vulnérabilités connues
Performance	Variable, généralement performant	Rapide, efficace à grande échelle	Rapide, efficace	Variable, peut être lent dans certains cas
Compatibilité	Large	Large	En croissance	Large
Nombre de connexions	Variable	Variable	Plusieurs, généralement élevé	Variable
Prix	Gratuit (open-source), tarifs commerciaux disponibles	Souvent inclus dans les équipements réseau ou les systèmes d'exploitation, coûts supplémentaires pour les solutions commerciales	Gratuit (open-source), tarifs commerciaux disponibles	Souvent inclus dans les équipements réseau ou les systèmes d'exploitation, coûts supplémentaires pour les solutions commerciales

Notre groupe est le Groupe 2 et nous allons récupérer la plage de port ci-contre : 7200 ->7210

2. Schéma Infrastructure :



3. Solution :

Serveur Proxy :

Un proxy est un serveur intermédiaire qui relaye les requêtes entre un client et un serveur. Il peut masquer l'adresse IP du client pour protéger son anonymat. De plus, il permet de filtrer certains contenus en bloquant l'accès à des sites spécifiques et offre un contrôle sur l'accès aux ressources selon des règles prédéfinies.

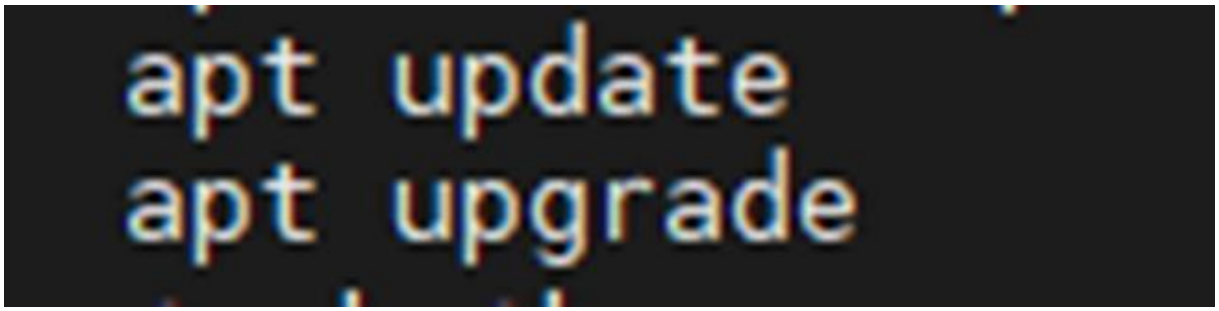
Modes de mise en œuvre de cette solution :

On peut l'installer via un logiciel avec Squid sur Ubuntu en version Graphique, avec une vue d'ensemble.

On peut aussi l'installer via un serveur Ubuntu en CLI (Ligne de commandes), c'est ce que nous allons faire.

Installation du Serveur Proxy sur Ubuntu Linux

On met à jour notre Serveur



Je commence par regarder quelle adresse IP mon serveur Ubuntu possède : (on en aura besoin pour la configuration)

```
root@g2:/etc/squid# 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
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:3e:9f:5b brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.10.202/24 metric 100 brd 192.168.10.255 scope global dynamic ens33
       valid_lft 78808sec preferred_lft 78808sec
    inet6 2a01:cb19:143:9400:250:56ff:fe3e:9f5b/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86349sec preferred_lft 549sec
    inet6 fe80::250:56ff:fe3e:9f5b/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 10.8.0.1/24 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fddd:1194:1194:1194::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::fbf7:87bf:ff73:3272/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Ensuite on installe Squid (notre solution de proxy) avec :

```
apt-get install squid
```

A la fin de l'installation, nous allons commencer la configuration de notre Serveur Proxy :

On se met dans le dossier squid avec :

```
cd /etc/squid/
```

Dans ce dossier nous voyons qu'il y a des dossiers de configuration par défaut, nous allons éditer ce fichier de configuration et repartir de zéro afin de faire quelque chose de propre :

```
root@g2:/etc/squid# ls
conf.d  domain.txt  passwd  squid.conf
root@g2:/etc/squid#
```

Et on peut commencer la configuration : (nous allons détailler chaque ligne avec les « # »

```
  GNU nano 6.2                                        squid.conf
# Squid a besoin de savoir le nom de la machine, notre machine s'appelle ubuntuserv, >
visible_hostname g2

# restreindre à écouter sur l'interface du réseau local (LAN)
http_port 192.168.10.202:3128

# Changer la taille du cache de squid
cache_dir ufs /var/spool/squid 100 16 256

# ACL pour autoriser tous les réseaux
acl all src all

# ACL pour autoriser notre réseau
acl lan src 192.168.10.0/24
acl lan src 192.168.51.0/24
acl lan src 10.8.0.0/24
#on dit que tous les ports suivants sont sûrs :
acl Safe_ports port 80
acl Safe_ports port 443
acl Safe_ports port 21
acl Safe_ports port 7201
acl Safe_ports port 3128

# Désactiver tous les protocoles sauf les ports sures
http_access deny !Safe_ports

# Désactiver l'accès pour tous les réseaux sauf les clients de l'ACL Lan
http_access deny !lan

#Comme demandé dans le tp on change le port par défaut en 3140
http_port 3128


# Déclarer un fichier qui contient les domaines à bloquer
acl deny_domain url_regex -i "/etc/squid/domain.txt"

# Refuser les domaines déclarés dans le fichier définit dans l'ACL deny_domain
http_access deny deny_domain

# configuration de l'authentification avec basic_ncsa_auth  /etc/squid/passwd file
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 2 hours

# Déclarer une ACL pour les utilisateurs authentifiés
acl auth_users proxy_auth REQUIRED

# Autoriser l'accès pour les utilisateurs authentifiés
http_access allow auth_users

# Refuser l'accès pour les utilisateurs non authentifiés
http_access deny all
```

Avec cette commande on crée un système d'authentification (pour faire fonctionner ce qui est spécifié dans le commentaire numéro 12)

```
apt-get install apache2-utils -y
```

Avec cette commande on veut créer un utilisateur (dans notre cas user1 avec comme mdp user1234!) : `htpasswd -c /etc/squid/passwd user1`

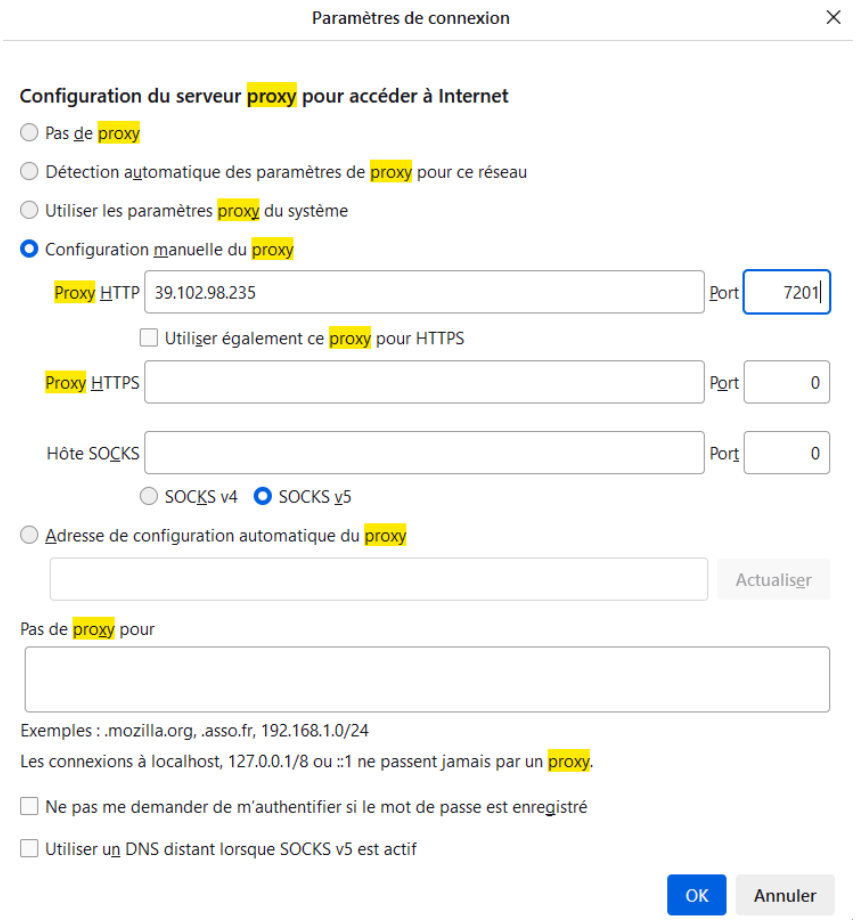
Et on rentre deux fois le mot de passe user1234 !

Ensuite nous créons le fichier dans lequel on veut spécifier les noms de domaines/site web à exclure dans le fichier de configuration squid, dans le fichier /etc/squid/domain.txt, dans notre cas :

Et à la fin on redémarre notre serveur Proxy pour qu'il prenne en compte notre configuration.

Ensuite, nous allons sur notre machine cliente afin de vérifier que cela fonctionne

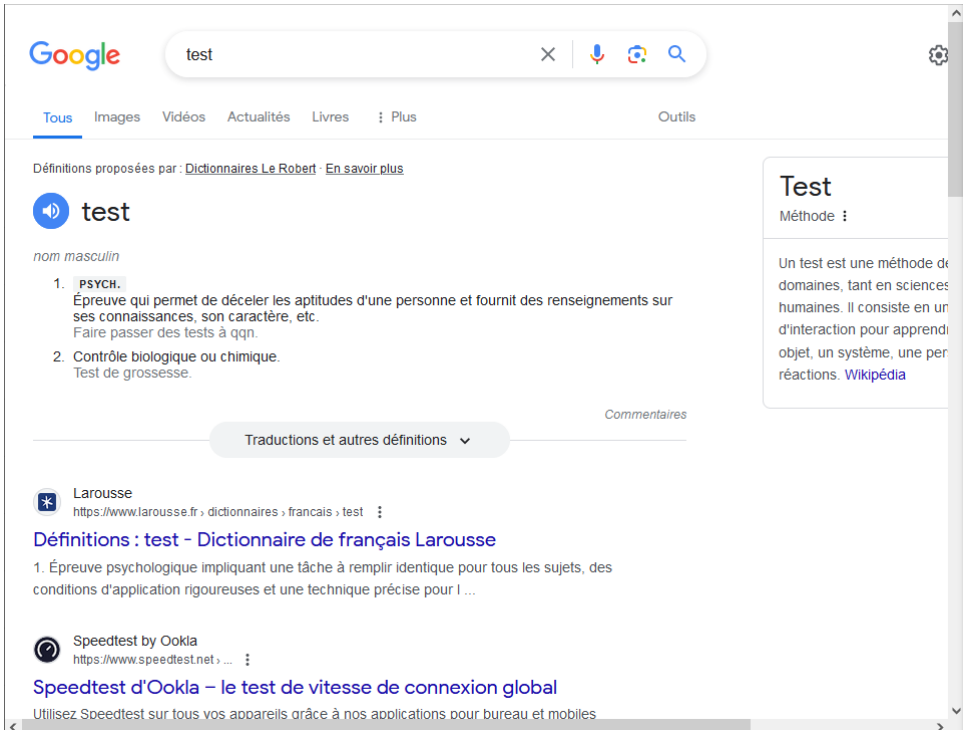
Tout d'abord nous mettons les paramètres du proxy dans notre machine cliente :



Ici, nous mettons l'adresse IP Publique du professeur suivi du port externe sur lequel le flux arrive

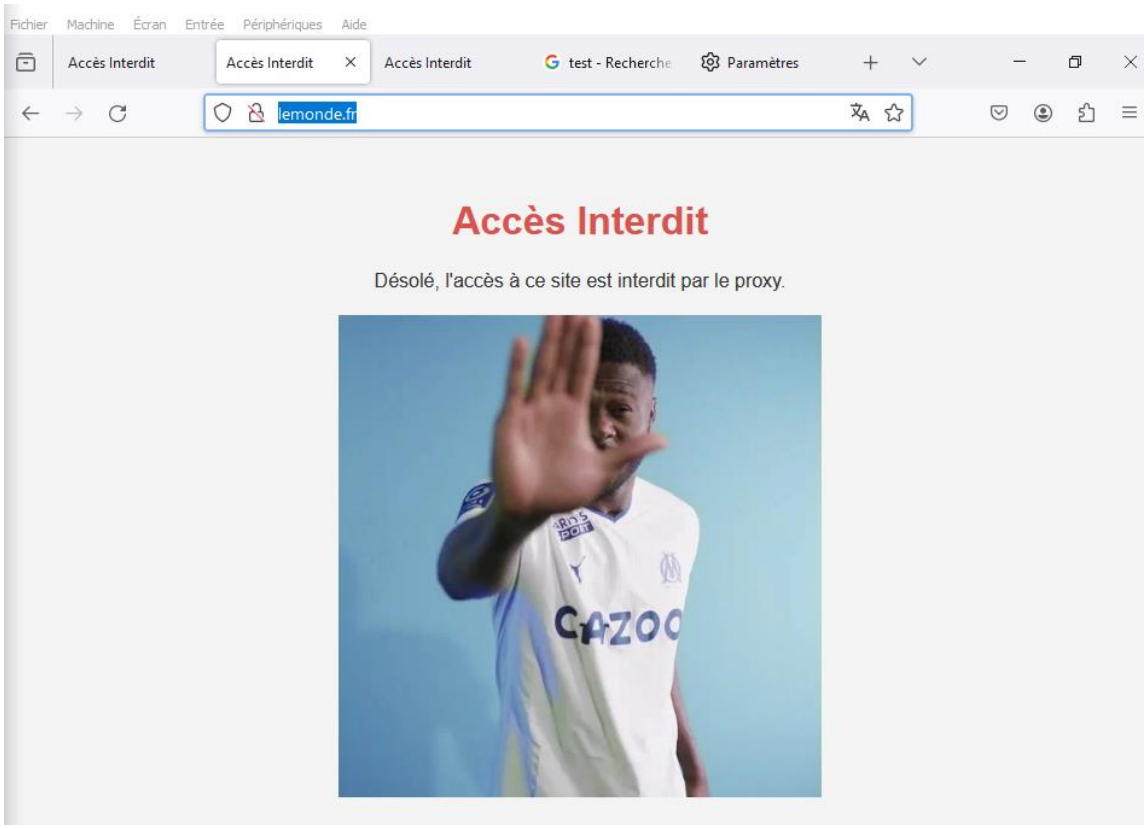
Et nous nous rendons sur Firefox afin de tester que cela fonctionne. (Après vérification avec le professeur, l'authentification ne fonctionnera pas sur ce tp)

On peut voir qu'on peut accéder à internet et ses sites sans soucis mais dès que nous nous rendons sur un des sites bloqués par le proxy, cela ne fonctionne pas.

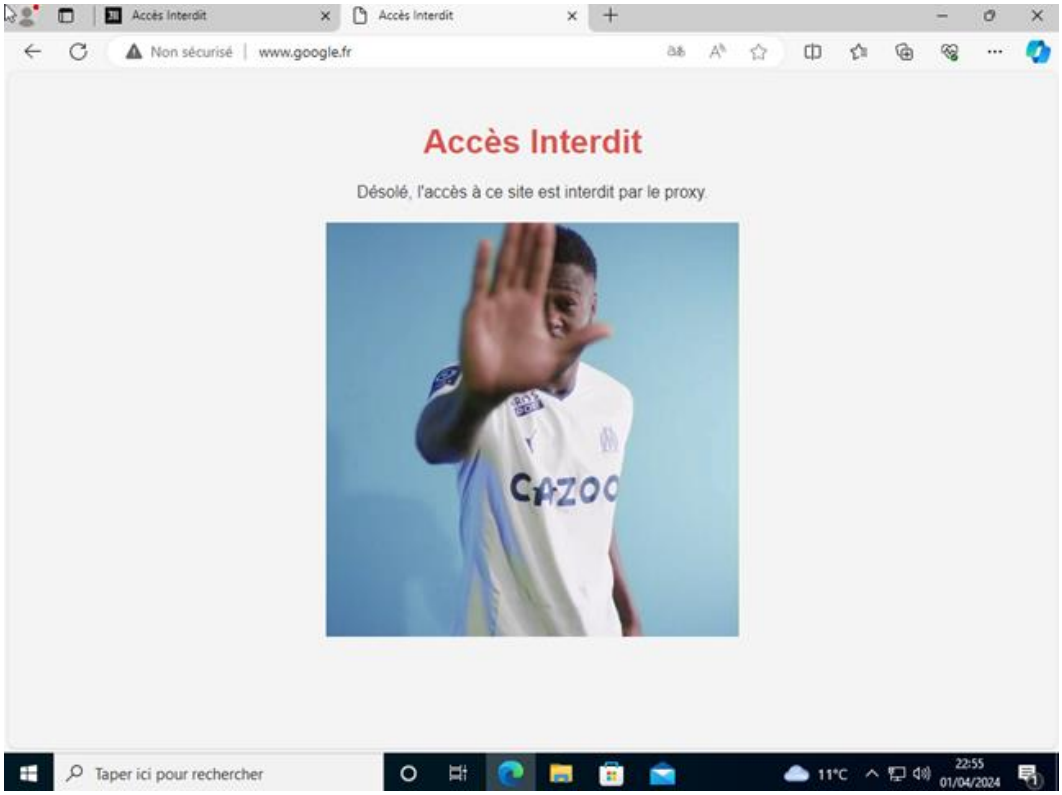


On peut voir que les sites www.google.fr, www.lemonde.fr et www.orange.fr sont bloqués :

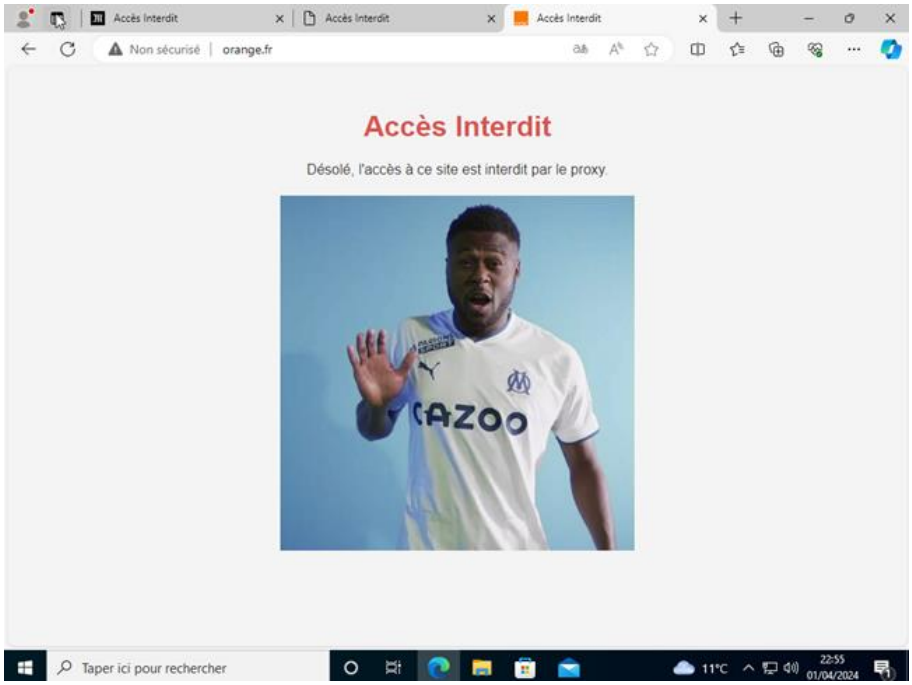
lemonde.fr :



google.fr



orange.fr



Pour avoir une page du proxy modifiée comme ci-dessus, nous nous rendons dans ce chemin de dossiers afin de modifier le contenu par défaut, par une page HTML créée par moi-même :

```
root@ubuntuserv:~# cd /usr/share/squid-langpack/fr/ERR_ACCESS_DENIED
```

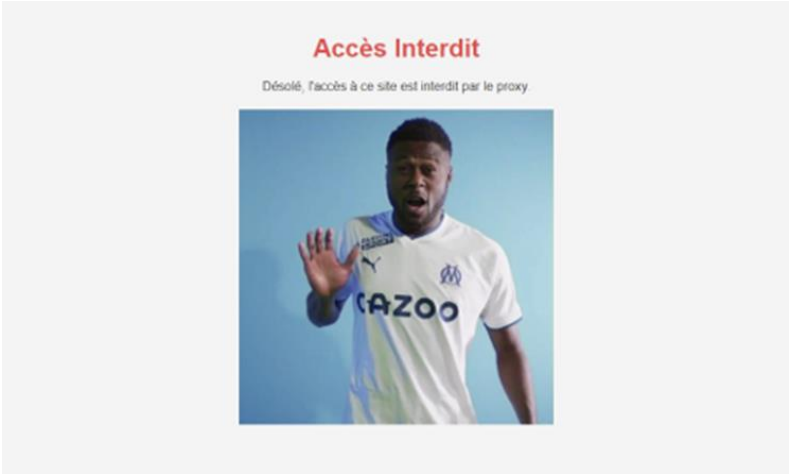
Avant :

```
  GNU nano 4.8                                                    /usr/share/squid-langpack/fr/ERR_ACCESS_DENIED
<html><head>
<meta type="copyright" content="Copyright (C) 1996-2019 The Squid Software Foundation and contributors">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>ERREUR : l'URL demandée n'a pas pu être chargée</title>
<style type="text/css"><!--
 %l

body
:lang(fa) { direction: rtl; font-size: 100%; font-family: Tahoma, Roya, sans-serif; float: right; }
:lang(he) { direction: rtl; }
 --></style>
</head><body id="%c">
<div id="titles">
<h1>ERREUR</h1>
<h2>L'URL demandée n'a pas pu être trouvée</h2>
</div>
<hr>

<div id="content">
<p>L'erreur suivante s'est produite en essayant d'accéder à l'URL : <a href="%U">%U</a></p>

<blockquote id="error">
<p><b>Accès interdit.</b></p>
</blockquote>

<p>La configuration du contrôle d'accès, empêche votre requête d'être acceptée. Si vous pensez que c'est une erreur, contactez votre fournisseur d'accès.</p>

<p>Votre administrateur proxy est <a href="mailto:%w%W">%w</a>.</p>
<br>
</div>

<hr>
<div id="footer">
<p>Générée le %T par %h (%s)</p>
<!-- %c -->
</div>
</body></html>
```

Après :

```
 GNU nano 4.8                                    /usr/share/squid-langpack/fr/ERR_ACCESS_DENIED
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Accès Interdit</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            color: #333;
            text-align: center;
            margin-top: 50px;
        }
        h1 {
            color: #d9534f;
        }
        .gif-container {
            margin-top: 20px;
        }
    </style>
</head>
<body>
    <h1>Accès Interdit</h1>
    <p>Désolé, l'accès à ce site est interdit par le proxy.</p>
    <div class="gif-container">
        <img src="https://media1.giphy.com/media/aZ5f8KiaIt8u5b1dul/giphy.webp?cid=790b7611ioirtazcg7h67zmsspyitvluxngotv49ck0t4lh1&ep=v1_gifs_search&rid=giphy.webp&ct=g" alt="GIF" width="400">
    </div>
</body>
</html>
```

Ce qui donne donc :



Serveur VPN OpenVPN

A. Télécharger le script d'installation

On se connecte sur notre serveur où tourne le Squid et on va mettre en place la solution VPN. On met aussi à jour les packages et on installe curl.

```
sudo apt-get update
sudo apt-get install curl
```

Ensuite on télécharge notre script d'installation depuis github

```
curl -O https://raw.githubusercontent.com/Angristan/openvpn-install/master/openvpn-install.sh
```

Dès que le script est téléchargé, on doit ajouter les droits en exécution sur ce fichier

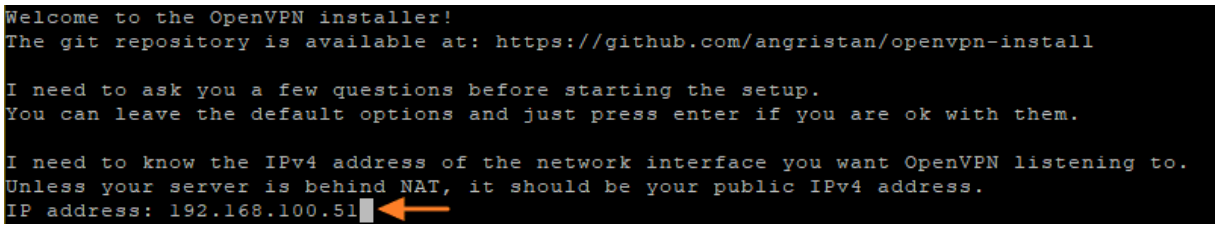
```
chmod +x openvpn-install.sh
```

Ensuite, on exécute le script pour commencer la configuration pas à pas d'OpenVPN Server :

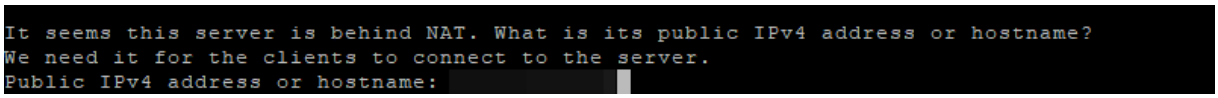
```
sudo ./openvpn-install.sh
```

B. Configurer le VPN

Le message "Welcome ton the OpenVPN installer!" s'affiche et les étapes de configuration vont s'enchaîner. Tout d'abord, il faut indiquer l'adresse IPv4 du serveur VPN, mais la bonne nouvelle, c'est qu'elle remonte automatiquement. S'il s'agit de l'adresse IP locale, cela signifie qu'il y a un NAT et dans ce cas, c'est logique. Sinon, l'adresse IP publique de votre serveur, par exemple de votre serveur VPS, s'affichera ici. Ici, le script remonte bien "192.168.100.51" Validez.



D'ailleurs, le script détecte la présence du NAT et indique l'adresse IP publique. Il suffit de valider (l'adresse IP public est cachée pour des raisons de confidentialité.

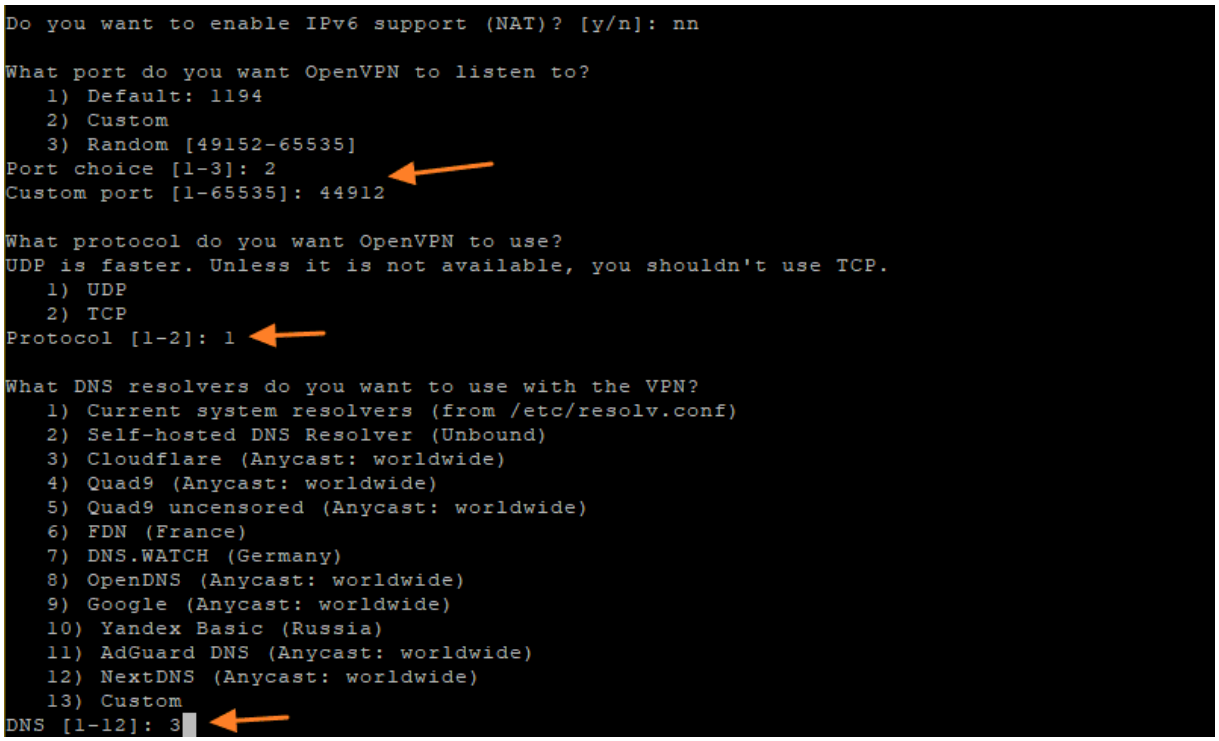


On ne veut pas activer le support IPv6 alors on indique non

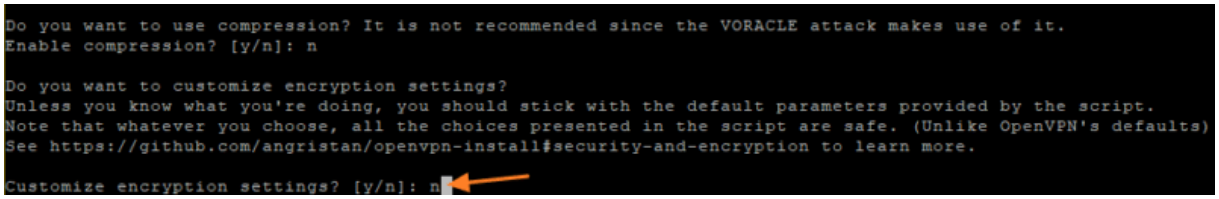
Il faut choisir un port, par défaut le VPN écoute sur le port 1194, or nous voulons qu'il soit sur le port 7202, on indique 2 et on met notre port personnalisé.

OpenVPN est plus rapide avec le protocole de transport UDP, et d'ailleurs c'est son mode de fonctionnement par défaut.

On a utilisé le protocole UDP.Une fois arrivé à cette question : « quel serveur DNS voulez-vous utiliser pour la résolution de noms. » . On peut choisir un serveur DNS personnalisé avec le choix 13, ou en choisir un dans la liste en indiquant son numéro. Nous avons choisi nous le DNS de Cloudflare dans le choix 3.



Le script est déjà préconfiguré pour utiliser certains paramètres pour le chiffrement du tunnel VPN et sa sécurité dans son ensemble. On a la possibilité de définir nos propres paramètres en indiquant "y", sinon il suffit de faire "n".



Ci-dessous, voici les différentes options proposées (*ainsi que les choix recommandés et correspondants à la configuration automatique*) si on besoin de définir nous même nos paramètres

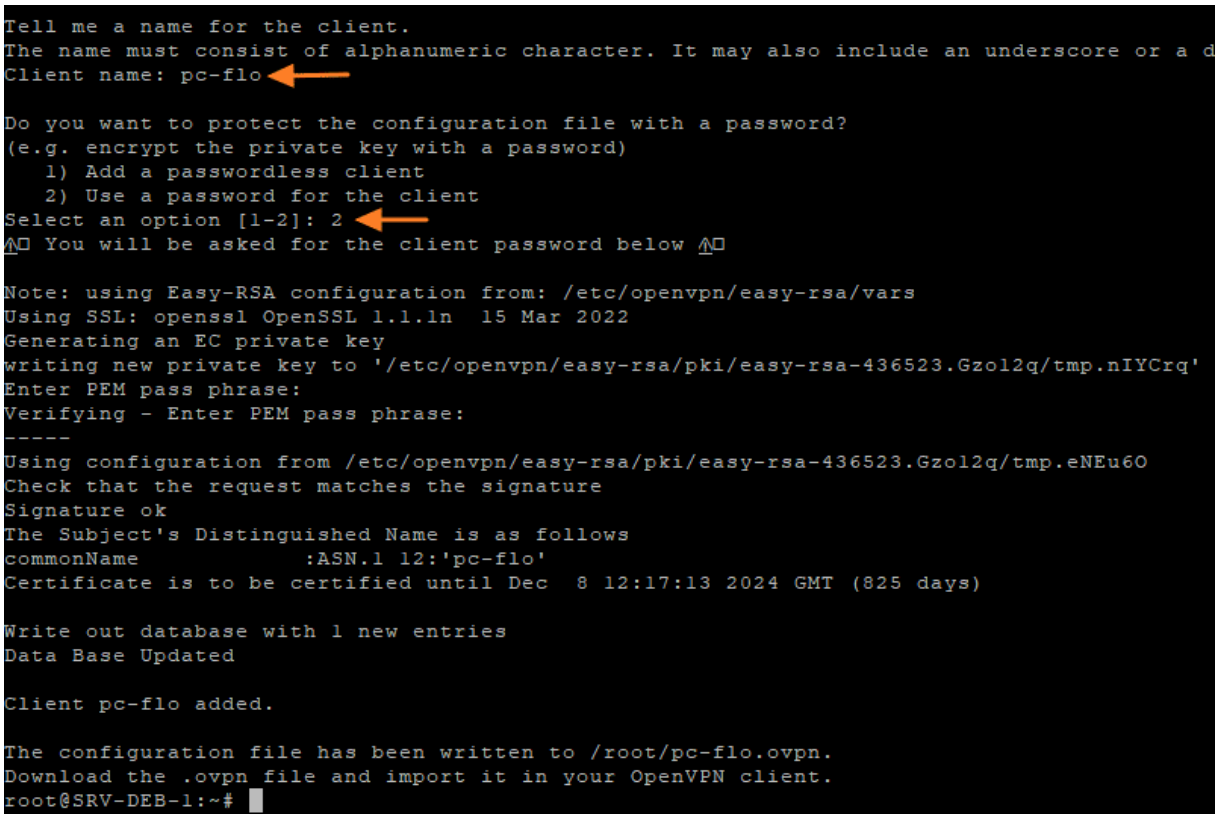
```
Choose which cipher you want to use for the data channel:
   1) AES-128-GCM (recommended)
   2) AES-192-GCM
   3) AES-256-GCM
   4) AES-128-CBC
   5) AES-192-CBC
   6) AES-256-CBC
Cipher [1-6]: 1

Choose what kind of certificate you want to use:
   1) ECDSA (recommended)
   2) RSA
Certificate key type [1-2]: 1

Choose which curve you want to use for the certificate's key:
   1) prime256v1 (recommended)
   2) secp384r1
   3) secp521r1
Curve [1-3]: 1

Choose which cipher you want to use for the control channel:
   1) ECDHE-ECDSA-AES-128-GCM-SHA256 (recommended)
   2) ECDHE-ECDSA-AES-256-GCM-SHA384
Control channel cipher [1-2]: 1

Choose what kind of Diffie-Hellman key you want to use:
   1) ECDH (recommended)
   2) DH
DH key type [1-2]: 1

Choose which curve you want to use for the ECDH key:
   1) prime256v1 (recommended)
   2) secp384r1
   3) secp521r1
Curve [1-3]: 1

The digest algorithm authenticates tls-auth packets from the control channel.
Which digest algorithm do you want to use for HMAC?
   1) SHA-256 (recommended)
   2) SHA-384
   3) SHA-512
Digest algorithm [1-3]: 1

You can add an additional layer of security to the control channel with tls-auth and tls-crypt
tls-auth authenticates the packets, while tls-crypt authenticate and encrypt them.
   1) tls-crypt (recommended)
   2) tls-auth
Control channel additional security mechanism [1-2]: 1
```

La première partie de l'interrogatoire est terminé ! Jusqu'à présent, le script n'a pas encore modifié la machine locale. En revanche, à ce moment précis si on appuie sur la touche "Entrée" (*ou une autre touche*), l'installation du serveur OpenVPN débutera.

```
Okay, that was all I needed. We are ready to setup your OpenVPN server now.
You will be able to generate a client at the end of the installation.
Press any key to continue...
```

C. Création d'un premier client

Pour donner suite à la configuration du serveur VPN, l'installation via le script se poursuit avec la création d'un premier client VPN. Il faut indiquer le nom du PC qui va utiliser le VPN (histoire de s'y retrouver), par exemple "*pc-flo*". Ensuite, la question "*Do you want to protect the configuration file with a password?*" s'affiche, il faut taper "2" pour oui afin de définir un mot de passe qui sera nécessaire pour établir la connexion VPN.



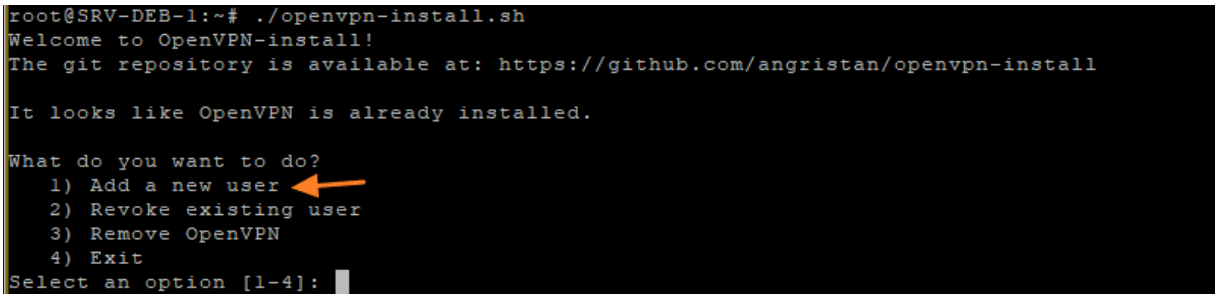
Ceci va générer un fichier de configuration OVPN dans le profil de l'utilisateur en cours d'utilisation. Ici, je suis connecté en tant que root alors la configuration est générée dans "/root/". D'un point de vue du serveur VPN, l'ajout de ce client va générer deux fichiers :

- Le certificat du client dans /etc/openvpn/easy-rsa/pki/issued/<nom du client>.crt
- La clé privée du client dans /etc/openvpn/easy-rsa/pki/private/<nom du client>.key

D. Ajouter un nouveau client OpenVPN

À tout moment, on peut ajouter un nouveau client pour que chaque machine qui se connecte dispose de son propre certificat. Que ce soit pour ajouter ou supprimer un nouveau client, il suffit de réexécuter le script et de faire le choix "1".

```
sudo ./openvpn-install.sh
```



Test de la connexion VPN

Le fichier de configuration généré précédemment (*/root/pc-flo.ovpn*) dans le profil de l'utilisateur doit être transféré sur l'ordinateur qui doit se connecter au VPN. On peut utiliser WinSCP pour le Transférer

A. Sur Windows

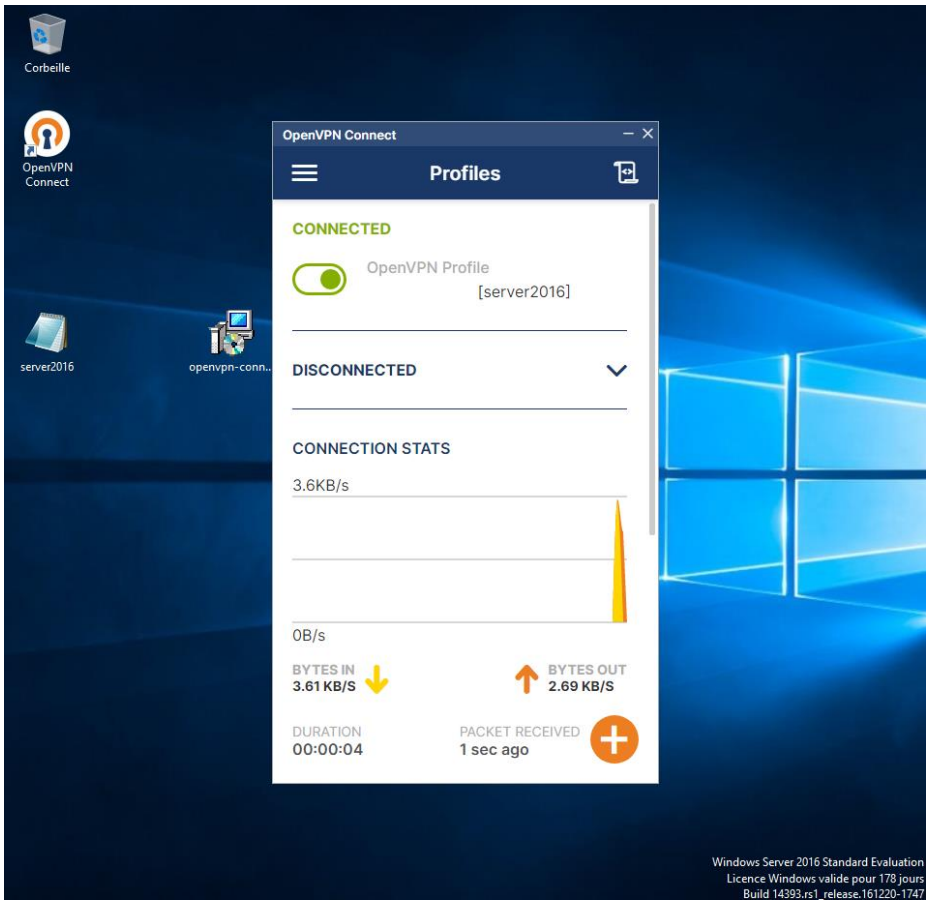
Sur Windows, il faudra installer OpenVPN GUI ou OpenVPN Connect. Personnellement, j'utilise OpenVPN GUI donc je dois copier-coller le fichier OVPN dans le répertoire suivant:

```
C:\Program Files\OpenVPN\config
```

Pour Tester la Connexion et que les autres appareils puissent accéder à des dossiers, on va créer un dossier partagé.

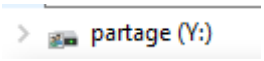
Ici on va se connecter à notre serveur VPN via un fichier OpenVPN.

On peut voir qu'on est connecté ici :



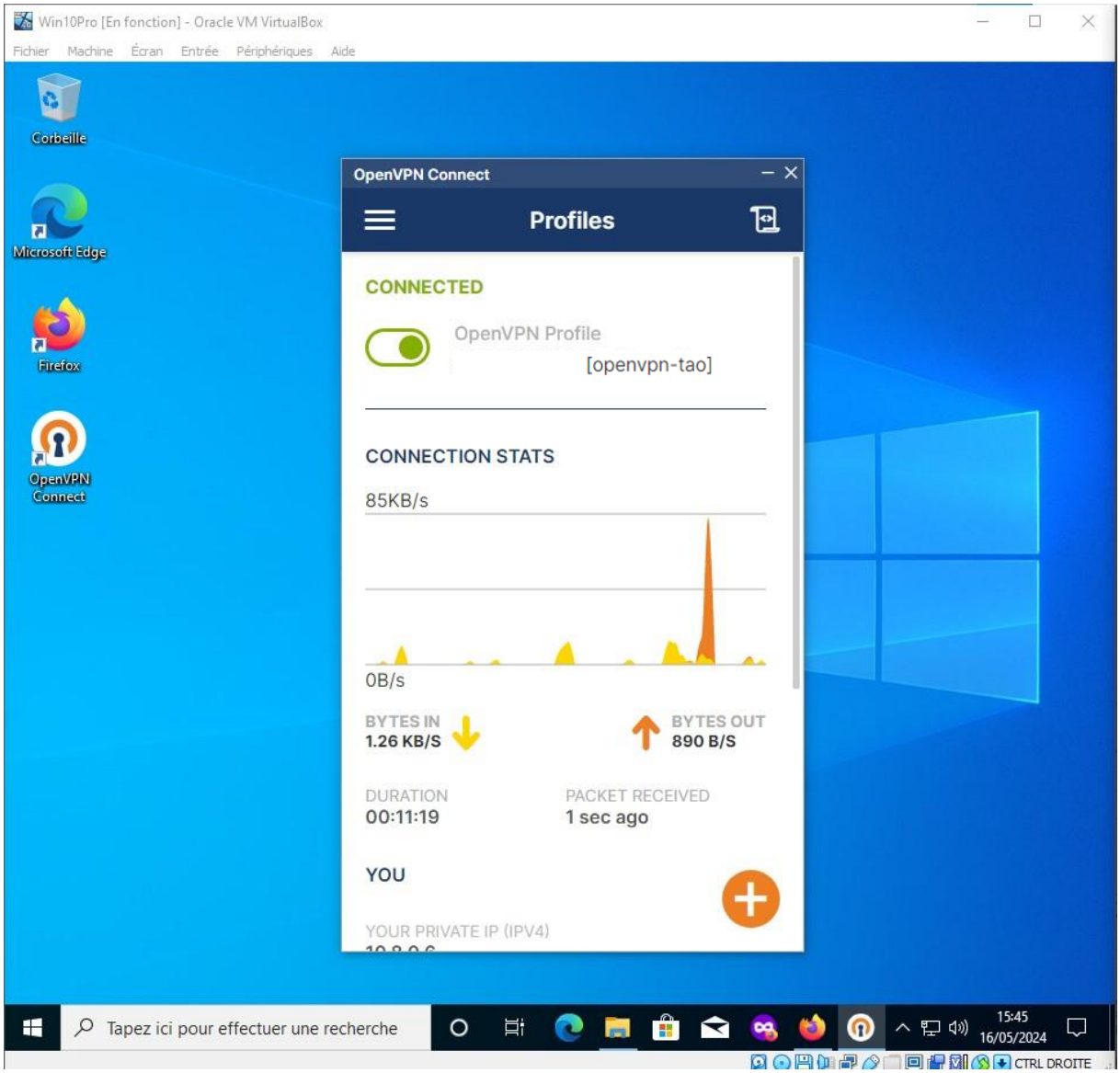
On va créer un dossier partagé afin que tout le monde puisse y accéder.

On peut voir notre dossier partagé créé ici :

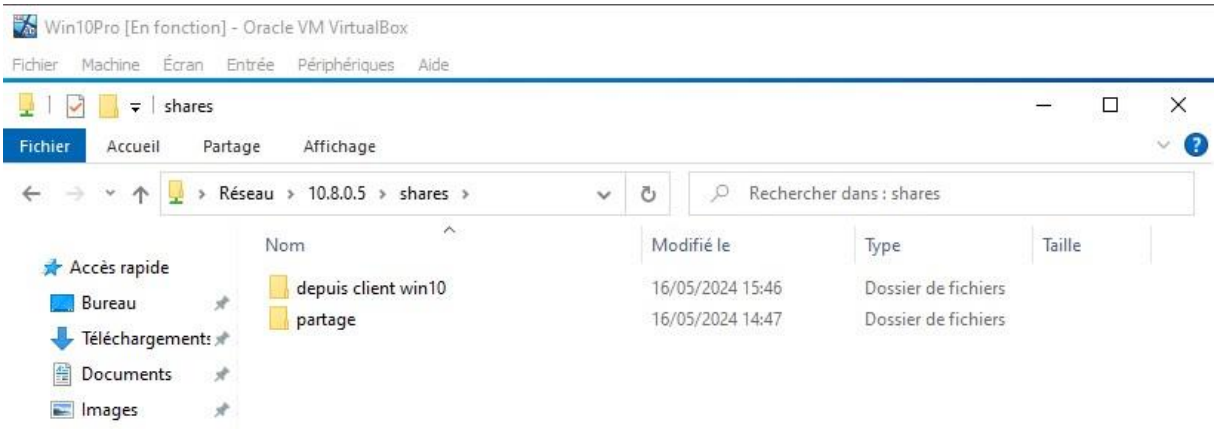


B. Test sur machine Windows 10 :

On ouvre le tunnel VPN depuis la machine Windows 10

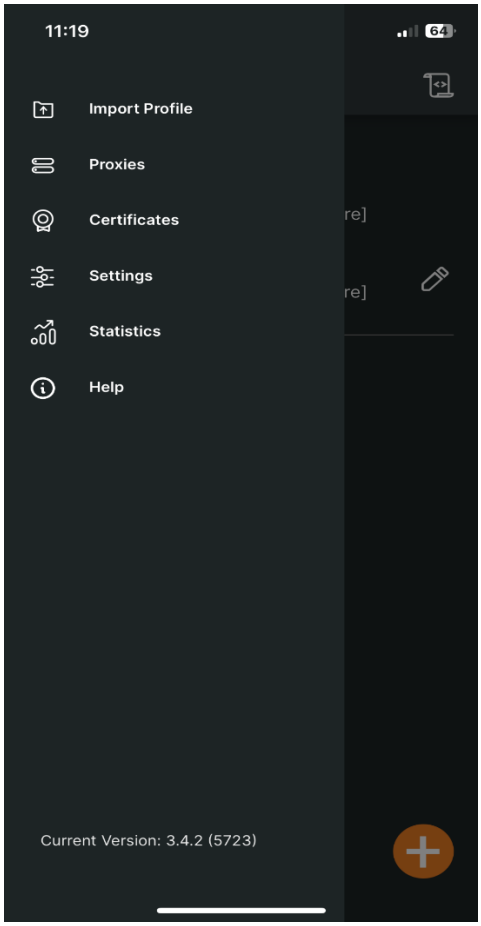


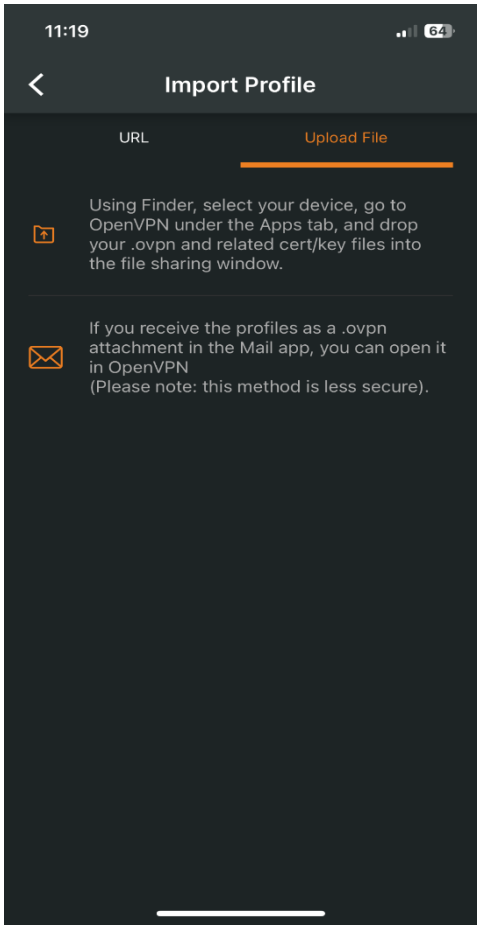
Maintenant depuis la VM Windows 10 on se connecte au serveur de fichier :



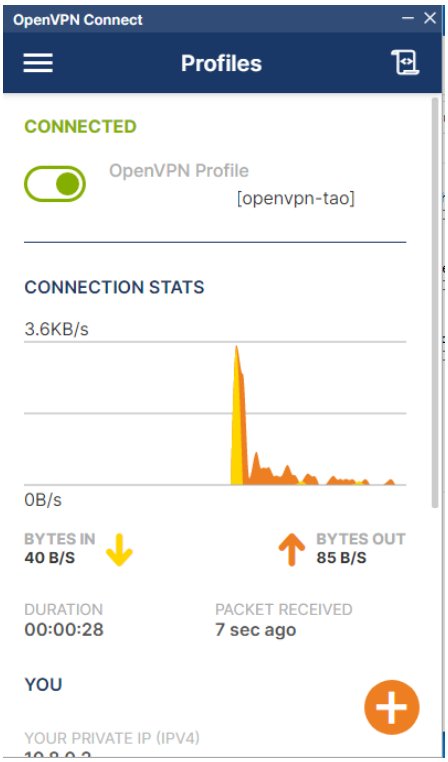
C. Test OpenVPN depuis le téléphone :

Il faut installer l'application OpenVPN sur le Play store ou l'App store et importer un profil créé auparavant, il faut télécharger au préalable le fichier de connexion au serveur VPN



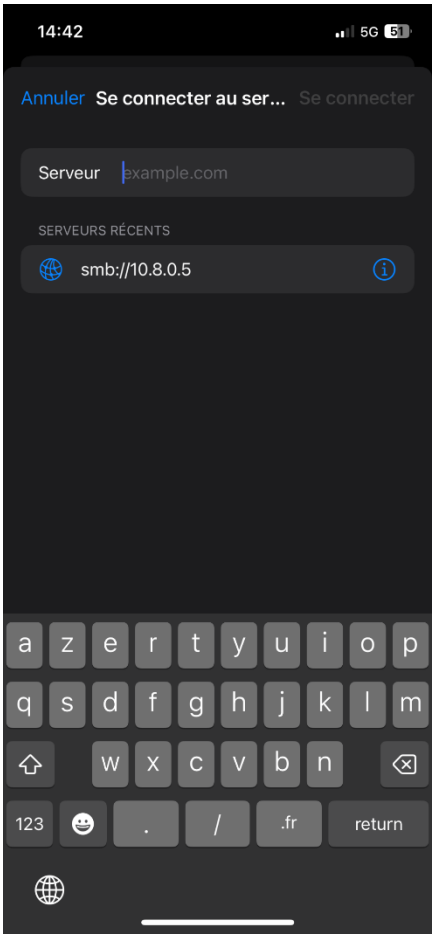


À la suite de cela nous pouvons voir que nous sommes connectés :



D. Test d’accès du serveur de fichier depuis le téléphone :

Depuis le téléphone on peut se connecter sur un serveur de fichier de cette manière :



Comme nous le voyons ci-dessous : on voit que nous avons notre dossier « partage » et donc on a accès au serveur de fichier depuis un téléphone par le biais d'un tunnel VPN.