INSTALLATION D'UN SERVEUR PIVPN SOUS RASPBERRY PI

Raspberry - Debian Buster
Configuration de base

Tutoriel **OPENVPN** - RASPBERRY

David GOÏTRÉ

Table des matières

Introduction

Un réseau privé virtuel (VPN) est un protocole utilisé pour ajouter la sécurité et la confidentialité aux réseaux privés et publics. Les VPN envoient du trafic entre deux ou plusieurs appareils sur un réseau dans un tunnel chiffré. Une fois la connexion VPN établie, tout le trafic réseau est chiffré du côté du client. Les VPN masquent votre adresse IP de sorte que nos actions en ligne sont pratiquement introuvables.

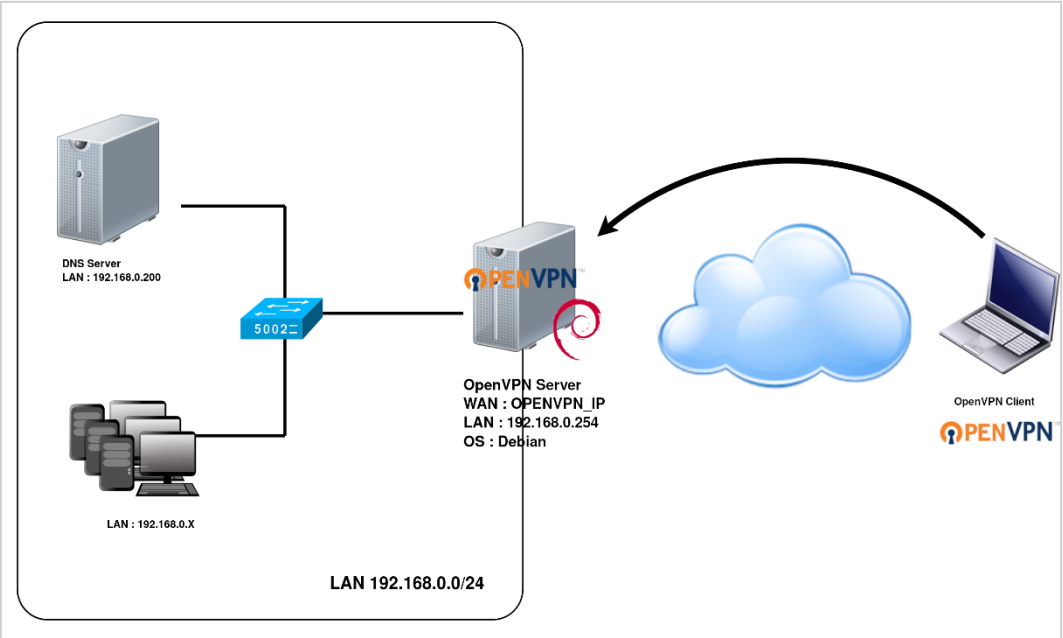
Il fournit le cryptage et l'anonymat, protège nos activités en ligne, nos achats en ligne, l'envoi d'e-mails et aide également à garder notre navigation Web anonyme.

1. Pré requis

On a besoin des différents matériels et logiciels pour la création d'un Serveur VPN avec un RaspberryPi.

- Un ou des PC client sous Windows
- Une Box (Free, Orange ou Sfr)
- Un Raspberry 3B+ avec l'OS Raspian Buster installé avec Etcher
- Le logiciel OpenVPN pour les clients
- Le logiciel Putty pour se connecter en SSH au serveur VPN
- Connaitre l'interface réseau (eth0, br0, ens3…) via la commande : **ip a**
  Pour notre test c'est **l'interface eth0** qui sera utilisée

Voici le schéma que l'on doit obtenir une fois le serveur VPN mise en place :



Ce schéma n'est qu'un exemple. Il n'est pas essentiel de posséder une machine Serveur DNS, ni d'avoir plusieurs PC Client sur le réseau LAN.

2. Paramétrage du serveur

Avant d'aller plus loin, il nous faut connaître l'interface réseau de notre serveur **RaspberryPi** et lui attribuer une adresse IP fixe.

a) Lister les interfaces

```
$ ip link | awk '{ print $2}'   # liste les interfaces
# ethtool <interface> | grep detected   # détecte l'interface connectée
```

b) Définir une adresse IP fixe

```
# nano /etc/network/interfaces   # ouvre le fichier des interfaces
```

c) Copier le texte ci-dessous dans le fichier **interfaces**

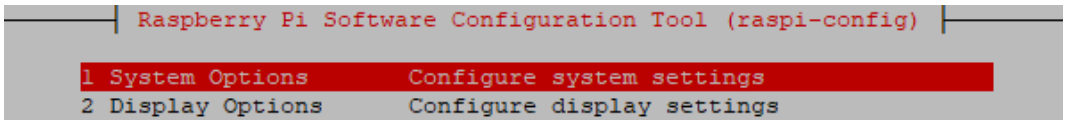
```
# Interface reseau de bouclage
auto lo
iface lo inet loopback
# Interface reseau principale
allow-hotplug eth0
iface eth0 inet static
address 192.xxx.xxx.xxx
netmask 255.255.255.0
gateway 192.xxx.xxx.xxx
```

d) Rebooter le serveur

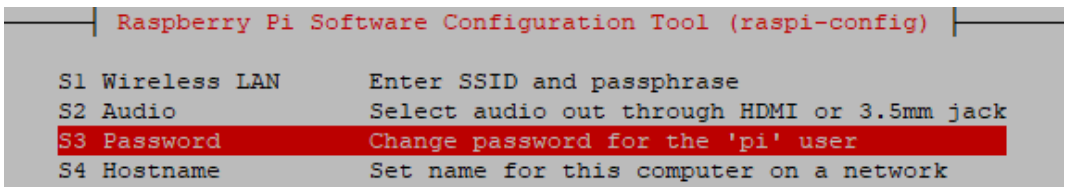
```
# /etc/init.d/networking restart
# reboot
```

e) Paramétrer le serveur

```
$ raspi-config   # ouvre l'utilitaire, sélectionner le menu System Options
```



Sélectionner le menu **S3 Password** pour modifier le mot de passe et **S4 Hostname** pour modifier le nom du serveur.



3. Paramétrage de connexion au serveur

a) Créer **une redirection de port** sur la box (Free, Orange…) vers votre serveur RaspberryPi.

- **port** : 1194
- **Protocole** : UDP

b) Activer le **SSH** sur le serveur. Pour ce faire, ouvrir le dossier **Boot**, de la carte SD du RaspberryPi via l'explorateur de Windows et créer un fichier **ssh** (sans extension) dans ce **dossier**.

c) Ouvrir **Putty** et se connecter au serveur VPN avec les identifiants (par défaut **pi/raspberry**)

b) Mettre à jour les packages du système vers la dernière version. Exécuter la commande suivante pour mettre à jour et mettre à niveau les packages de votre système :

```
# apt-get update -y
# apt-get upgrade -y
```

4. Activer le transfert IP

Certains aspects de la configuration réseau du serveur doivent être modifiés afin qu'OpenVPN puisse acheminer correctement le trafic à travers le VPN. Le premier d'entre eux est le transfert IP, une méthode permettant de déterminer où le trafic IP doit être acheminé. Ceci est essentiel pour la fonctionnalité VPN que notre serveur fournira. Editer le fichier **sysctl.conf** :

```
# nano /etc/sysctl.conf
```

Décommenter la ligne suivante :

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

Enregistrer le fichier lorsque l'on a terminé. Ensuite, exécuter la commande suivante pour appliquer les modifications :

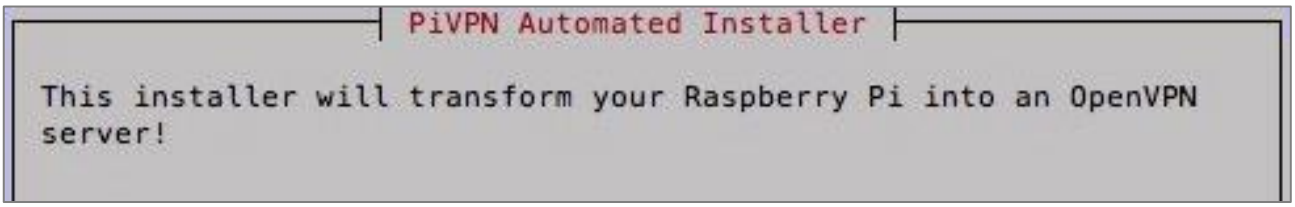
```
# sysctl -p
```

5. Installer le serveur PIVPN OpenVPN

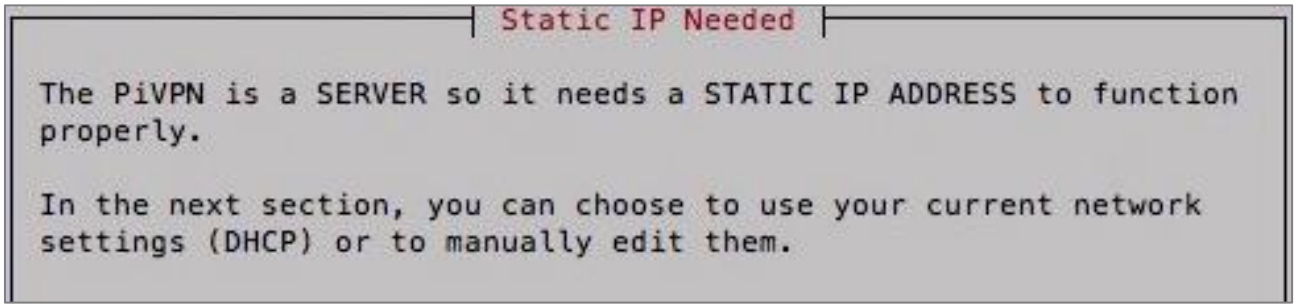
Par défaut, le paquet PIVPN (OpenVPN+easyrsa) n'est pas disponible dans le référentiel par défaut Debian 10. Il faut l'installer avec la commande suivante :

```
$ curl –L https://install.pivpn.io | bash
```

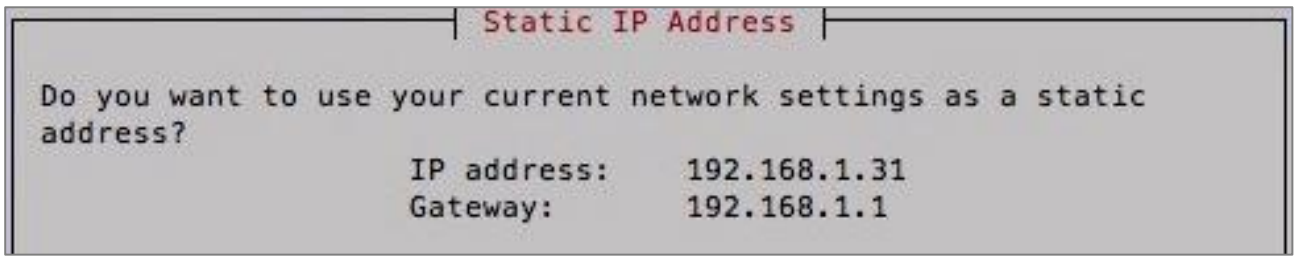
a) L'installation démarre :



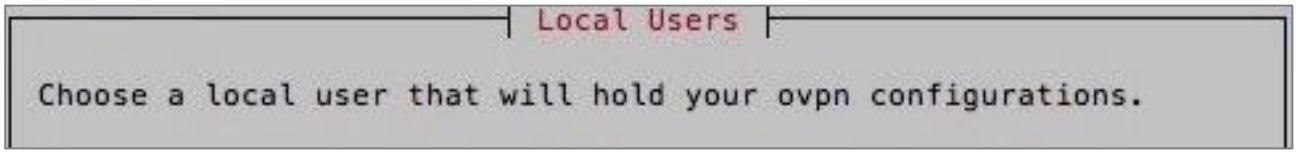
b) PiVPN utilise l'adresse IP attribuée via DHCP par le routeur ou la box. Il faut que celle-ci soit fixe. L'ip a été fixer à l'étape précédente.

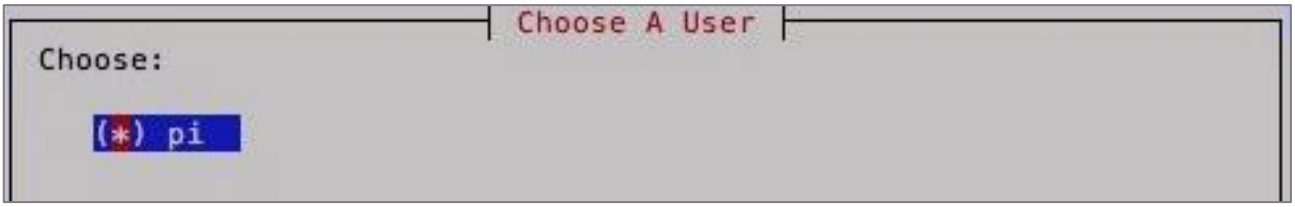


c) Valider l'adresse IP actuelle sera bien fixe



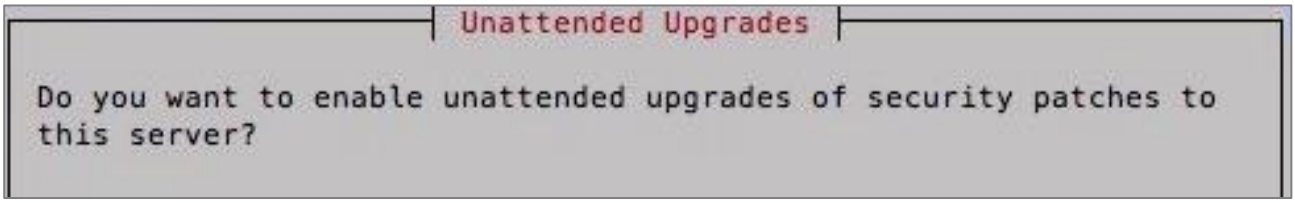
d) Choisir un utilisateur local qui gérera toutes les configs OpenVPN. Comme tout ordinateur on peut potentiellement avoir plusieurs utilisateurs. En l'occurrence là on ne peut en sélectionner qu'un seul : **pi**.



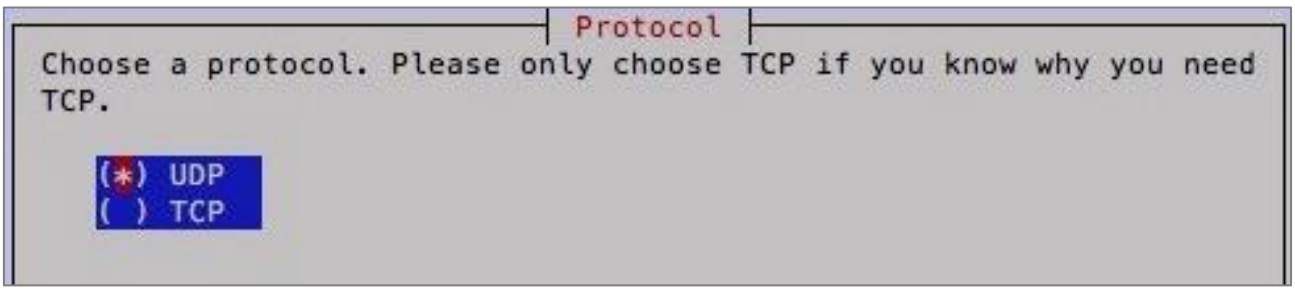


e) Choisir le mode **OpenVPN**. Le mode **WireGuard** est un logiciel libre qui permet d'établir des tunnels chiffrés de bout en bout (VPN) avec des outils et protocoles robustes et modernes comme le framework Noise, Curve25519, ChaCha20, Poly1305, BLAKE2, SipHash24, HKDF…etc., le tout avec des performances de dingue comparé à OpenVPN ou encore IPSec.

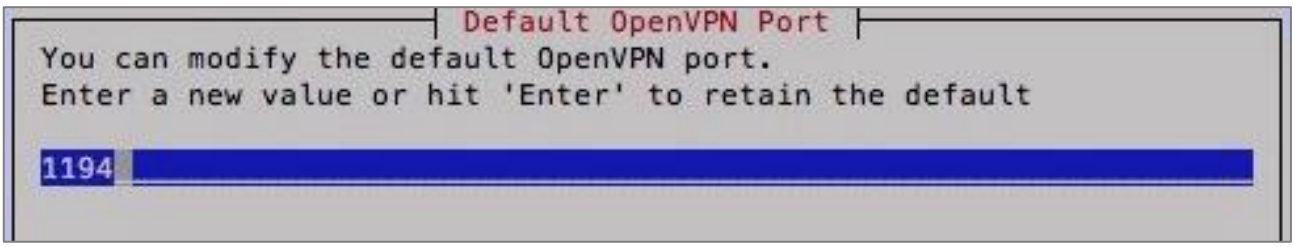
e) Donner l'autorisation au serveur d'installer lui-même les mises à jours de sécurité.



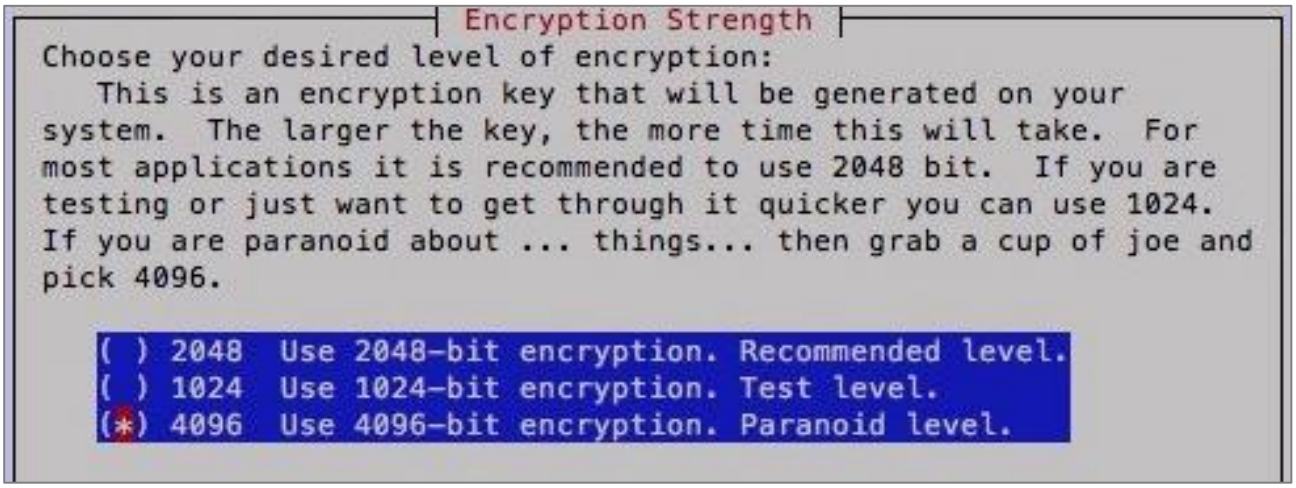
f) Sélectionner l'UDP comme protocole.



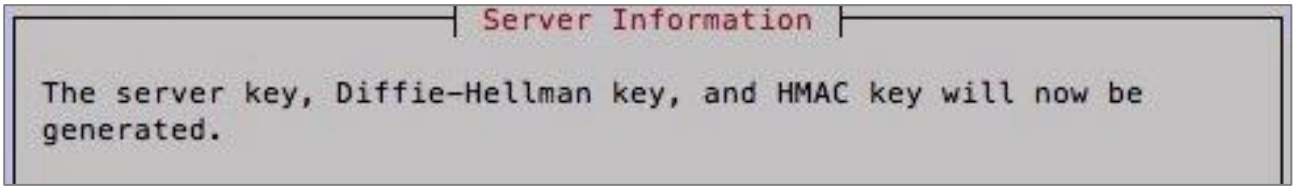
g) Par défaut le port 1194 est sélectionné pour le serveur OpenVPN. Si on n'a pas un besoin particulier, laisser tel quel et valider.



h) Maintenant on va définir le niveau de **chiffrement utilisé** par OpenVPN. Plus c'est élevé, plus le chiffrement sera dur à casser. Choisir un cryptage minimum de **2048-bit**.

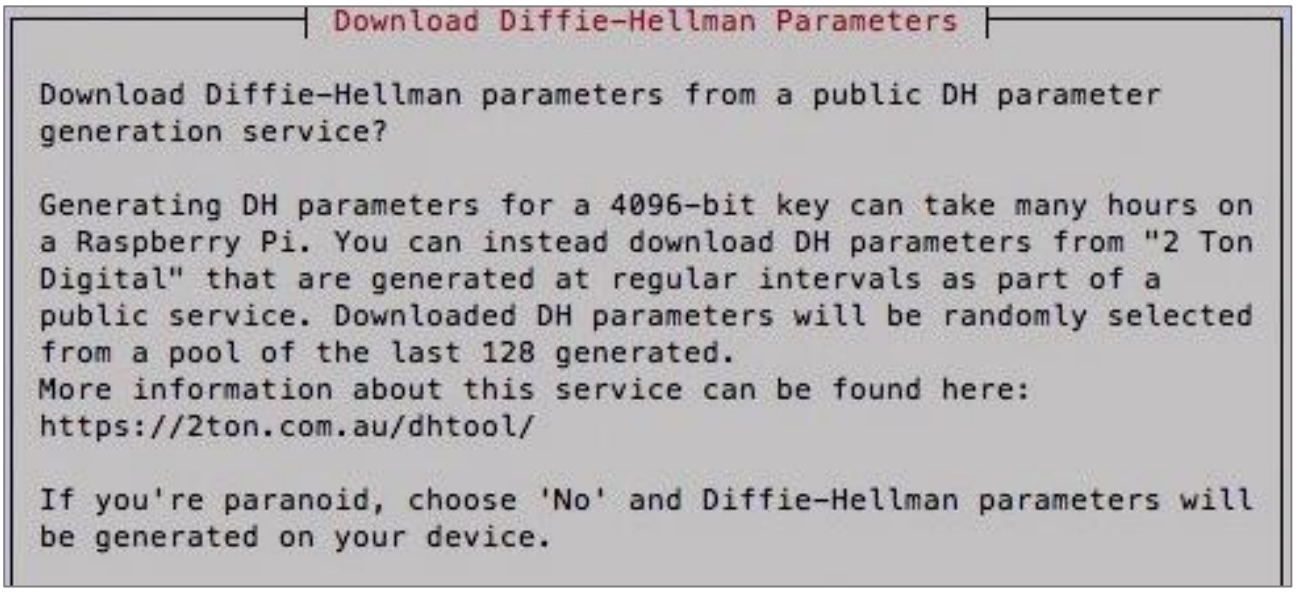


i) PiVPN va générer le certificat de chiffrement.



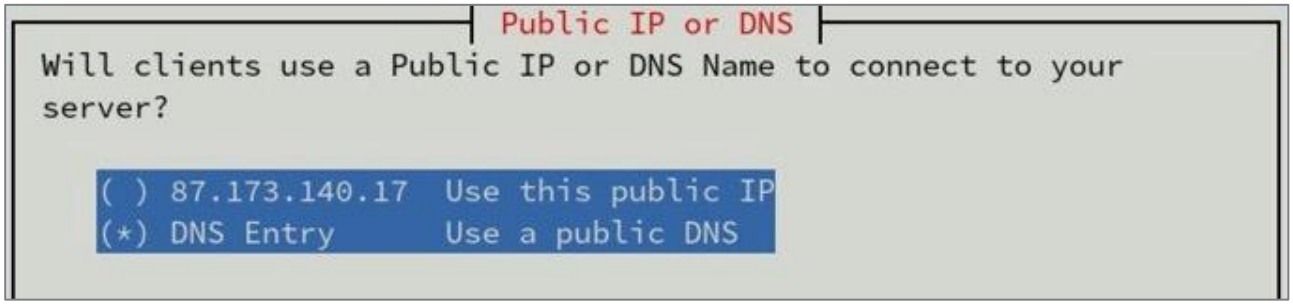

```
:::
::: Stopping OpenVPN service... done.
:::
::: Checking for existing base files...
:::    Checking /etc/.pivpn is a repo...:::    Cloning https://github.com/pivpn/pi
vpn.git into /etc/.pivpn... done!
:::
::: Installing scripts to /opt/pivpn... done.
::: Using protocol: udp
::: Building CA...
Generating a 4096 bit RSA private key
.....................................................................................
.................++
.....................................................................................
.....................++
writing new private key to '/etc/openvpn/easy-rsa/pki/private/ca.key.PqlQipCQhk'
-----

::: CA Complete.

Note: using Easy-RSA configuration from: ./vars
Generating a 4096 bit RSA private key
```

j) La génération des paramètres Diffie Hellman peut prendre beaucoup de temps à faire sur le Raspberry Pi (plusieurs heures). Choisir **No**. Mais si on veut aller plus vite, PiVPN nous propose de récupérer des paramètres générés aléatoirement sur un serveur public, choisir **Yes**.

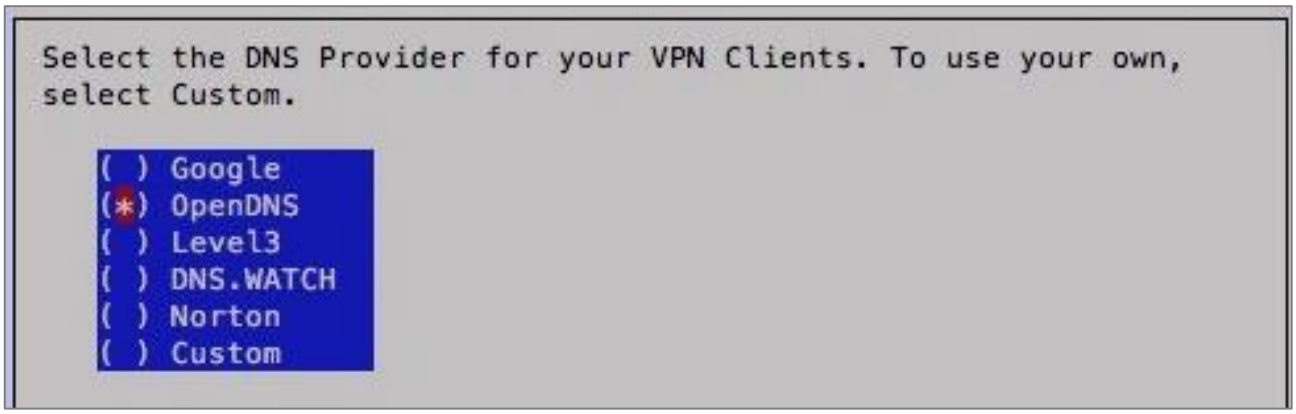


k) Maintenant, on PIVPN demande si les clients vont se connecter en utilisant l'adresse IP publique de notre Raspberry Pi ou un nom de domaine (référencé sur DNS public).

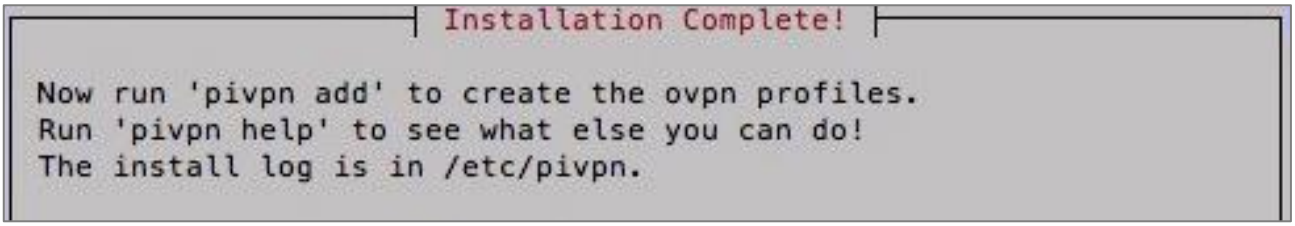
- Choisir **Use this public IP** pour utiliser le routeur ou la box
- Choisir DNS Entry pour utiliser un service en ligne **NoIP** (inscription obligatoire).



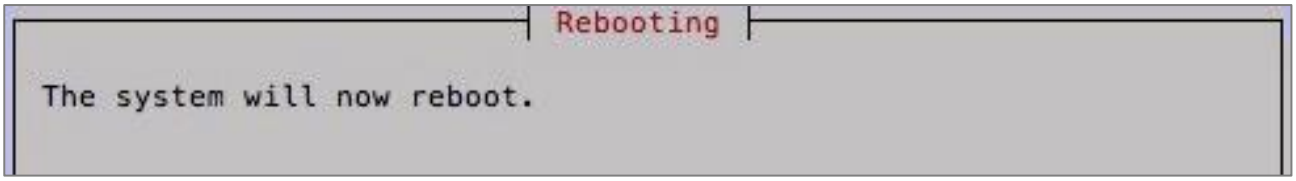
l) Choisir les serveurs DNS qui vont être utilisés : **OpenDNS**.



M) Explication sur la commande que l'on va utiliser pour créer un profile OpenVPN.



n) L'installation est terminée, il faut redémarrer le système avant de pouvoir ajouter des profils.



6. Ajouter un utilisateur

Une fois redémarré, se reconnecter en SSH à notre serveur Raspberry Pi. On va maintenant ajouter un utilisateur. Saisir la commande suivante :

```
$ pivpn -a
```

a) Saisir un nom d'utilisateur

b) Saisir un mot de passe que l'on reconfirme.

c) Notez bien toutes ces informations. PiVPN génère un certificat que l'on utilisera côté client. Pour récupérer ce certificat, voici une petite ligne de commande :

```
# scp pi@ADRESSE_IP:/home/pi/ovpns/user.ovpn /chemin-posteclient
```

Remplacer **user.ovpn** par le nom exacte du fichier de certificat généré. Cette commande ne se fait pas en étant connecté en SSH. Pour l'effectuer, ouvrir une nouvelle fenêtre dans le terminal.

7. Commandes PiVPN

Supprimer un client

```
$ pivpn -r
```

Lister tous les clients

```
$ pivpn -l
```

Afficher le code QR pour un client (nécessaire pour l'application mobile)

```
$ pivpn -qr
```

Afficher une liste de clients connectés

```
$ pivpn -c
```

Mettre à jour PiVPN

```
$ pivpn -up
```

Sauvegarde PiVPN

```
$ pivpn -bk
```

Déboguer PiVPN

```
$ pivpn -d
```

Désinstaller PiVPN

```
$ pivpn -u
```

8. Exemple du fichier de configuration du serveur

Fichier de configuration client OpenVPN généré automatiquement à la création d'un utilisateur avec PiVPN. Exemple : **server.conf**.

```
#Configuration server
Port 1194
proto udp
dev tun
#Cles certificats
ca /etc/openvpn/easy-rsa/pki/ca.crt
cert /etc/openvpn/easy-rsa/pki/issued/nomserver.crt
key /etc/openvpn/easy-rsa/pki/issued/nomserver.key
dh none
ciphers AES-256-CBC
tls-server
tls-version-min 1.2
tls-crypt /etc/openvpn/easy-rsa/pki/ta.key
auth SHA256
#Reseau
server 10.8.0.0 255.255.255.0
client-to-client
push "redirect-gateway def1 bypass-dhcp"
push "DNS option dhcp 208.67.222.222"
push "DNS option dhcp 208.67.220.220"
keepalive 15 120
#Securite
persist-key
persist-tun
user openvpn
group openvpn
#Logs
verb 3
```

9. Exemple de fichier de configuration du client

Fichier de configuration client OpenVPN généré automatiquement à la création d'un utilisateur avec PiVPN. Exemple : **client1.ovpn**.

```
#Client
client
dev tun
proto udp
remote xxx.xxx.xxx.xxx 1194
resolv-retry infinite
nobind
remote-cert-tls server
#Cles
ciphers AES-256-CBC
auth SHA256
auth-nocache
tls-version-min 1.2
tls-client
verify-x509-name server_xxxxxxxxxx name
#Securite
persist-key
persist-tun
key-direction 1
verb 3
<ca>
   -----BEGIN CERTIFICATE-----
   -----END CERTIFICATE-----
</ca>
<cert>
   -----BEGIN CERTIFICATE-----
   -----END CERTIFICATE-----
</cert>
<key>
   -----BEGIN ENCRYPTED PRIVATE KEY-----
   -----END ENCRYPTED PRIVATE KEY-----
</key>
<tls-crypt>
   ## 2048 bit OpenVPN static key#
   -----BEGIN OpenVPN Static key V1-----
   -----END OpenVPN Static key V1-----
</tls-crypt>
```

10. Démarrer le service OpenVPN

PiVPN est maintenant installé et configuré. On peut maintenant démarrer le service OpenVPN et l'activer après le redémarrage du système à l'aide de la commande suivante :

```
# systemctl start openvpn@server
# systemctl enable openvpn@server
```

Exécuter la commande suivante pour vérifier l'état du service OpenVPN :

```
# systemctl status openvpn@server
```

On doit obtenir la sortie suivante :

```
● openvpn@server.service - Connexion OpenVPN au serveur
Loaded: loaded (/lib/systemd/system/openvpn@.service; enableb; préréglage du fournisseur:
enable)
Active: active (en cours d'exécution) depuis ven 2020-02-21 15:38:31 UTC; Il y a 4s
  Documents: man: openvpn (8)
      https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
      https://community.openvpn.net/openvpn/wiki/HOWTO
PID principal: 3044 (openvpn)
Statut: "Séquence d'initialisation terminée"
 Tâches: 1 (limite: 2359)
Mémoire: 1,3 M
CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
      └─3044 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status  10 --
cd /etc/openvpn --config /etc/openvpn/server.
21 février 15:38:31 debian10 systemd [1]: Démarrage de la connexion OpenVPN au serveur ...
21 février 15:38:31 debian10 systemd [1]: Démarrage de la connexion OpenVPN au serveur.
```

11. Connecter le client Windows au VPN

Il faut transférer le fichier de configuration sur le PC Client à l'aide d'un logiciel FTP.

a) Se connecter au serveur via **FileZilla** avec les mêmes identifiants utilisés dans Putty.
b) Ouvrir le dossier et récupérer le fichier

```
$ /home/ovpns/client1.ovpn
```

c) Copier les fichiers dans le dossier **C:\Programmes\openvpn\config**
d) Ouvrir **OpenVPN** et se connecter
e) Ou avec **OpenVPN Connect**, importer le fichier de configuration **client1.ovpn** et se connecter

12. Configurer le routage à l'aide de UFW

Par défaut, le pare-feu UFW n'est pas installé dans Debian 10. On peut l'installer avec la commande suivante :

```
# apt-get install ufw -y
```

Après avoir installé le pare-feu UFW, vous devrez ajouter des règles de pare-feu pour activer le masquage afin que vos clients VPN accèdent à Internet.

Tout d'abord, vous devrez configurer UFW pour accepter les paquets transférés. On peut le faire en éditant le fichier **/etc/default/ufw** :

```
# nano /etc/default/ufw
```

Modifier la ligne suivante :

```
DEFAULT_FORWARD_POLICY = "ACCEPT"
```

Enregistrer et fermer le fichier. Ensuite, ouvrer le fichier **/etc/ufw/before.rules** :

```
# nano /etc/ufw/before.rules
```

Ajouter les lignes suivantes à la fin du fichier avant le COMMIT :

```
*nat
:POSTROUTING ACCEPT  [0: 0]
-A POSTROUTING -s 10.8.0.0/24 -o tun -j MASQUERADE
```

Enregistrer le fichier lorsque vous avez terminé. Ensuite, autoriser le port OpenVPN par défaut 1194 et OpenSSH avec la commande suivante :

```
# ufw allow 1194 /udp
# ufw allow OpenSSH
```

Ensuite, recharger le pare-feu UFW à l'aide de la commande suivante:

```
# ufw disable
# ufw enable
```

Si on ne veut pas installer le pare-feu, exécuter la commande suivante :

```
# iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o tun -j MASQUERADE
```

13. Commandes RaspberryPi

Liste des commandes utiles à la gestion du serveur RaspberryPI

```
# /etc/init.d/openvpn restart   # redémarre OpenVPN
# systemctl restart openvpn@server.service   # recharge le service
# shutdown -h now   # éteint le serveur en toute sécurité
# shutdown -r now   # redémarre le serveur en toute sécurité
# apt install xrdp   # install le bureau à distance RDP
# systemctl enable xrdp   # active xrdp en tant que service système
# apt install openssh-server  # installe le SSH
# systemctl enable sshd.service  # active le service SSH au démarrage
##### Désactive la mise en veille #####
# systemctl mask sleep.target suspend.target hibernate.target hybrid-sleep.target
```

Autre méthode d'installation de **PiVPN :**

```
$ curl https://raw.githubusercontent.com/pivpn/pivpn/master/auto_install/install.sh
| bash
```

14. Conclusion

PiVPN est installé et configuré avec succès sur le serveur **RaspberryPi Debian 10**. On peut désormais accéder à Internet en toute sécurité et protéger son identité.

Destiné au RaspberryPi (Raspbian), **PiVPN OpenVPN** fonctionne aussi parfaitement sur une Debian ou une Ubuntu en mode VPS ou un ordinateur personnel.

Informations :

- Le mode **TAP** correspond à du Open SSL
- Le mode **TUN**, bridgé, correspond à un tunnel IPsec, c'est du site à site