INSTALLATION D'UN SERVEUR PIVPN SOUS RASPBERRY PI

RaspberryPi - Debian Buster
Configuration de base

Tutoriel **WIREGUARD** - RASPBERRYPI

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.

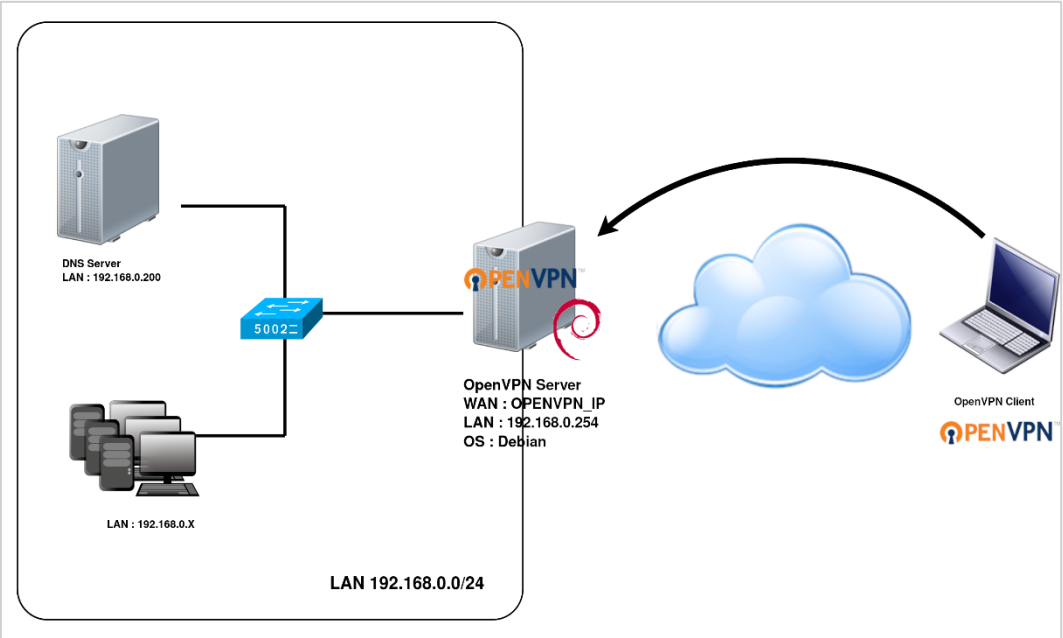
Il fournit le cryptage et l'anonymat, protège nos activités en ligne, nos achats en ligne, l'envoi d'e-mails et aide également à garder notre navigation Web anonyme.

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, Sfr…)
- Un Raspberry 3B+ avec l'OS Raspian Buster installé avec Etcher
- Le logiciel WireGuard 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**

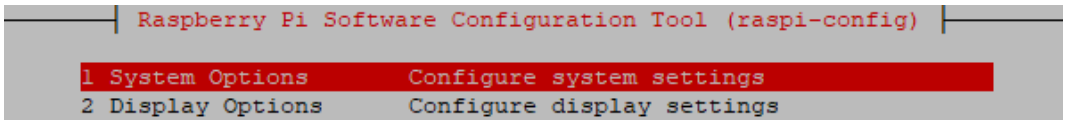
```
# Interface reseau de bouclage
auto lo
iface lo inet loopback
# Interface reseau principale
allow-hotplug eth0
iface eth0 inet static
address 192.xxx.xxx.xxx
netmask 255.255.255.0
gateway 192.xxx.xxx.xxx
```

d) Rebooter le serveur

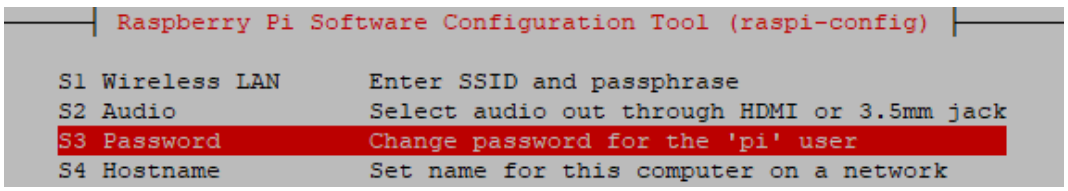
```
# /etc/init.d/networking restart
# reboot
```

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 que WireGuard 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 :

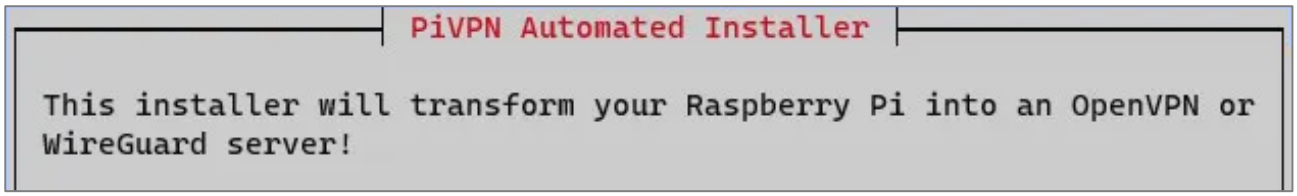
```
# sysctl -p
```

5. Installer le serveur PIVPN WireGuard

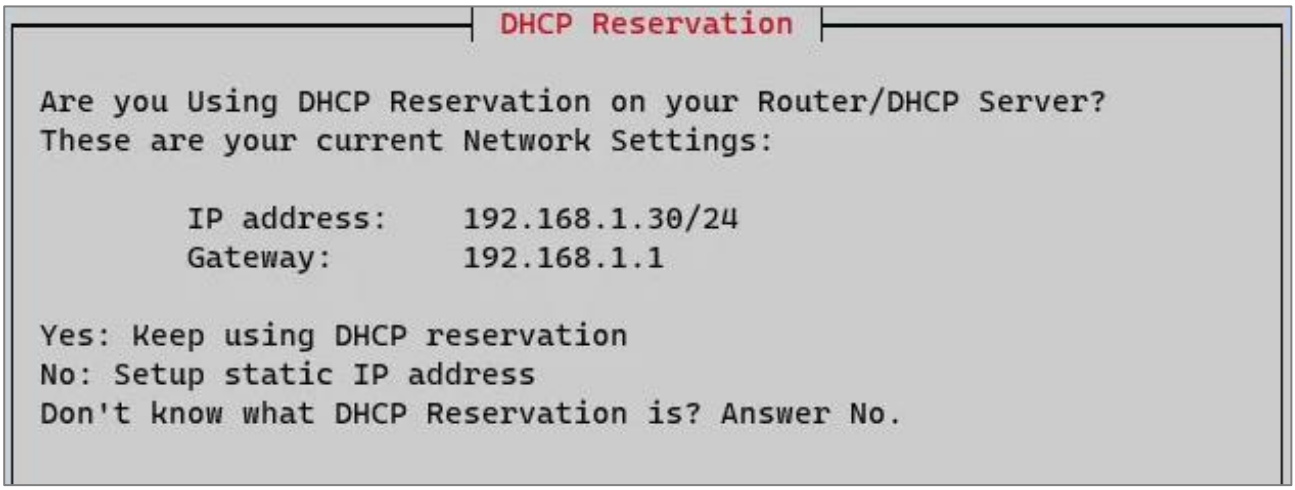
Par défaut, le paquet PIVPN WireGuard n'est pas disponible dans le référentiel par défaut Debian 10. Il faut l'installer avec la commande suivante :

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

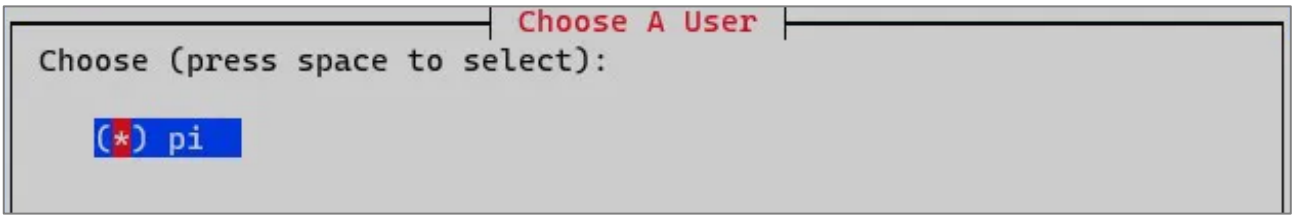
a) L'installation démarre :



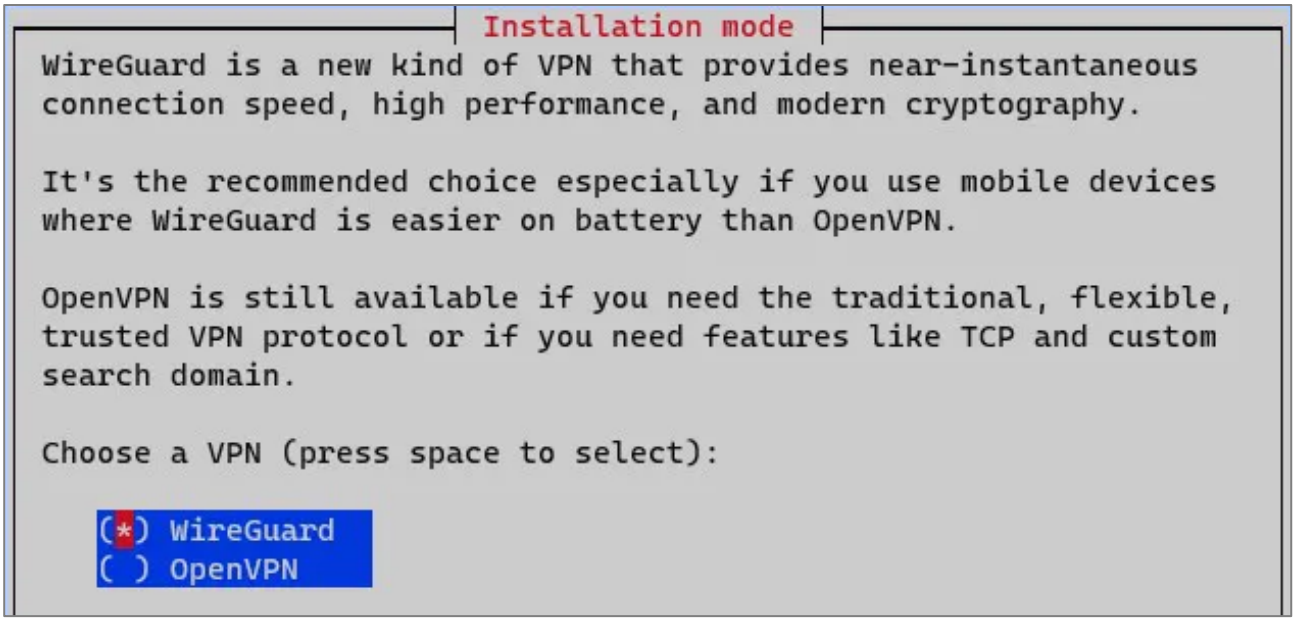
b) Définir une adresse IP statique. Il est préférable de définir une adresse IP statique dans les paramètres du routeur, car n s'assure que DHCP n'essaye pas de donner cette adresse à d'autres périphériques. Notre serveur possède déjà une IP fixe défini plus haut, on peut donc sauter cette étape.



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



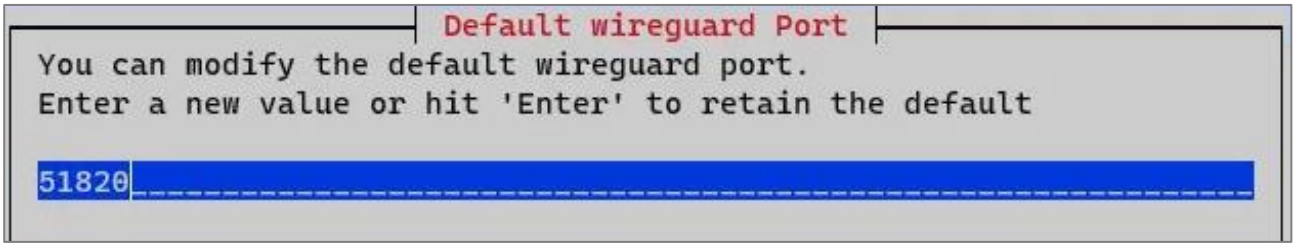
d) Choisir le mode **WireGuard**. C'est un logiciel libre qui permet d'établir des tunnels chiffrés de bout en bout (VPN) avec des outils et protocoles robustes et modernes comme le framework Noise, Curve25519, ChaCha20, Poly1305, BLAKE2, SipHash24, HKDF…etc., le tout avec des performances de dingue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



e) Sélectionner **Oui** pour procéder à la mise à jour du noyau. Il faudra redémarrer le serveur à la fin de l'installation.

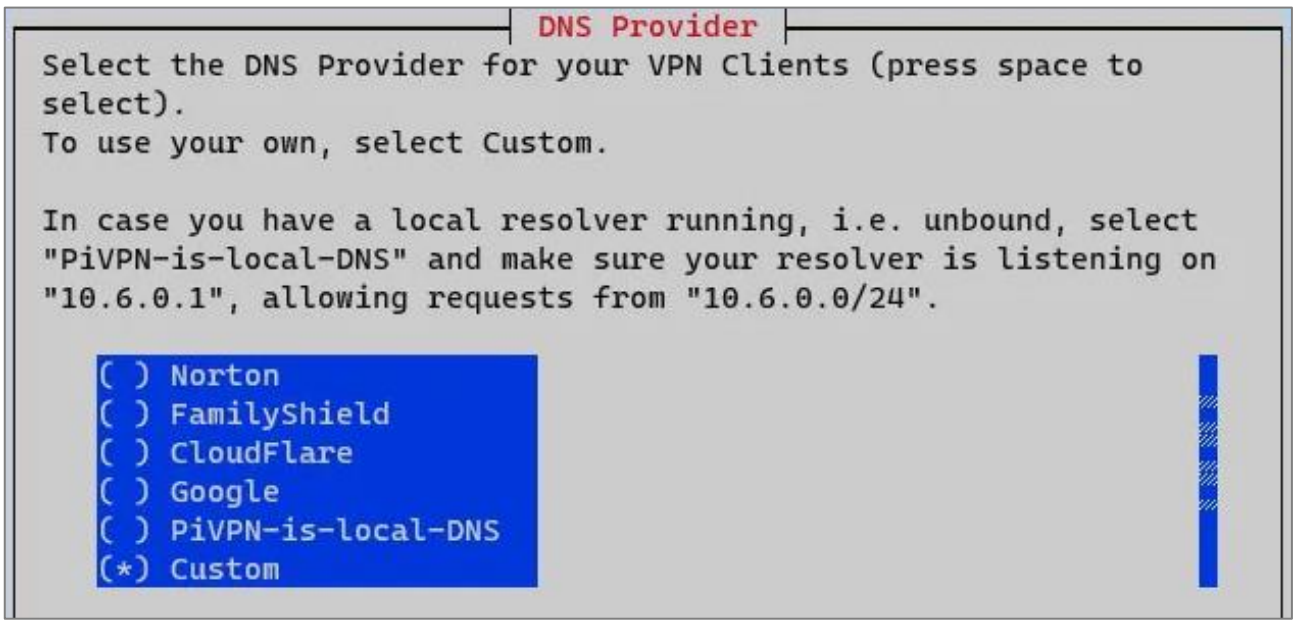
f) WireGuard s'installe.

g) Par défaut le port **51820** est sélectionné pour le serveur WireGuard. Si on n'a pas un besoin particulier, laisser tel quel et **valider**.

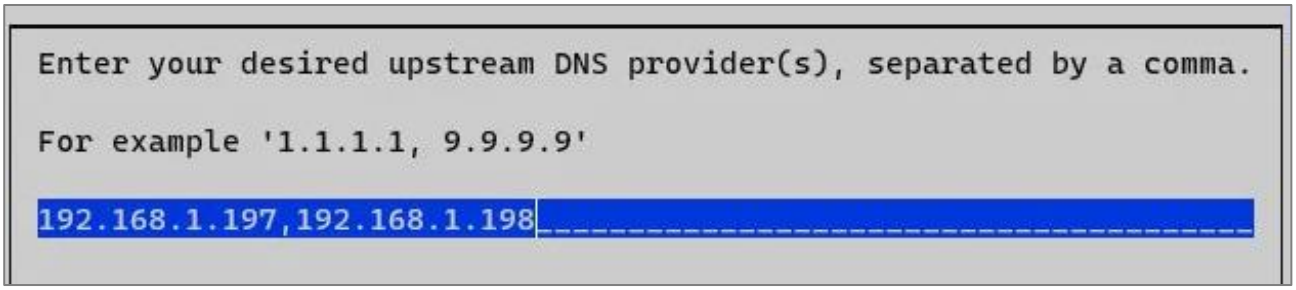


h) Confirmation du choix du port. Cliquer sur le bouton **Yes**.

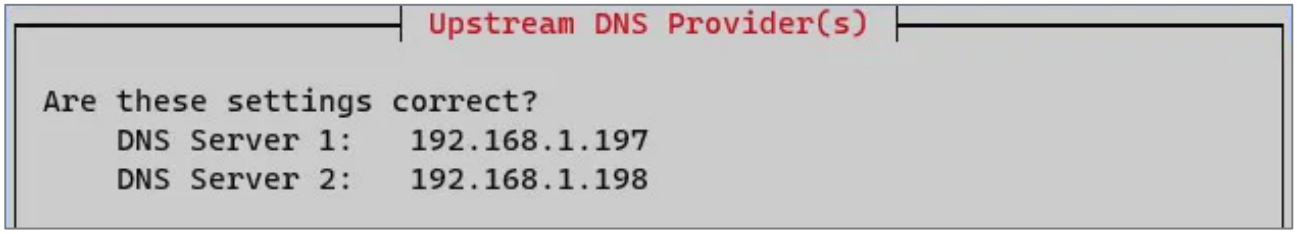
i) On doit maintenant sélectionner le fournisseur DNS que l'on souhaite utiliser. Sélectionner **Custom** si l'on veut utiliser son propre serveur DNS ou l'un des fournisseurs DNS publics ou si l'on ne souhaite pas utiliser de serveur DNS local.



j) Les IP ci-dessous sont des DNS locaux, les remplacer par des IP publics. Puis cliquer sur **Ok**.

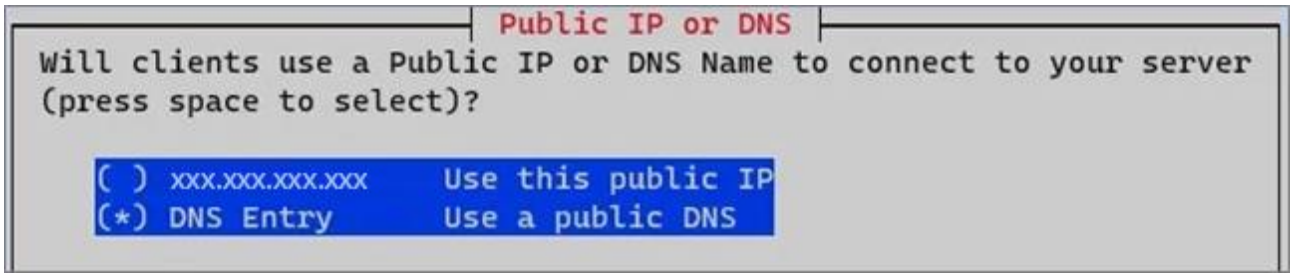


k) Les serveurs DNS que l'on sélectionne seront désormais répertoriés. Cliquer sur le bouton **Yes**

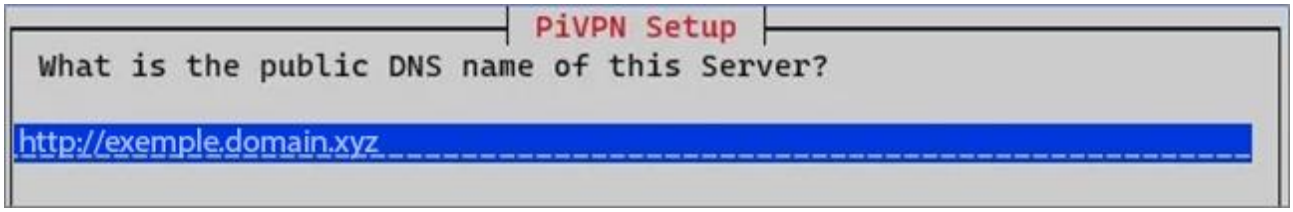


l) Maintenant, WireGuard demande si les clients vont se connecter en utilisant l'adresse IP publique de notre Raspberry Pi ou un nom de domaine (référencé sur DNS public).

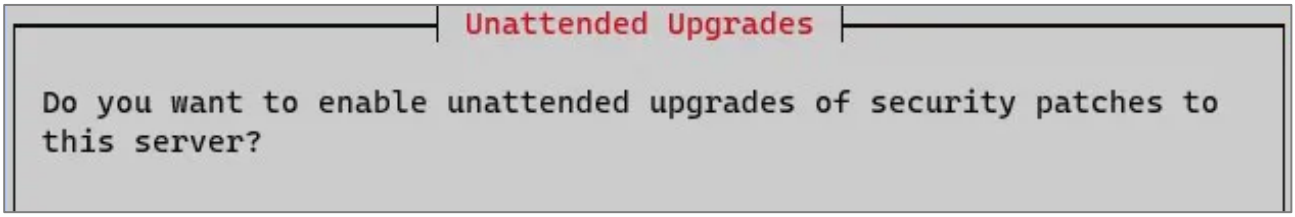
- Choisir **Use this public IP** pour utiliser L'IP du routeur ou de la box
- Choisir **DNS Entry** pour utiliser une IP dynamique, tel que sur NoIP (inscription obligatoire).



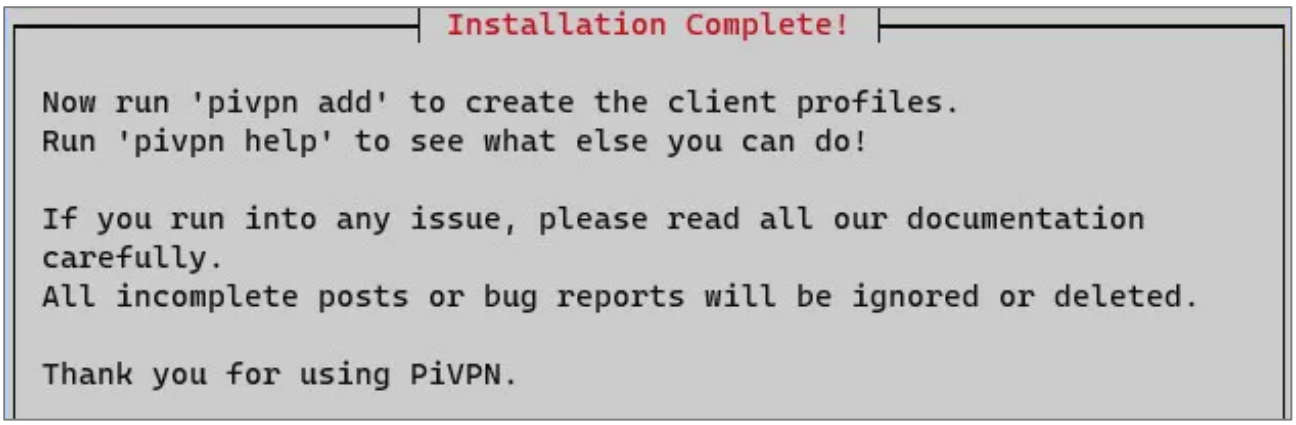
m) Saisir une **Url de DNS public** pour le serveur. Cliquer sur le bouton **Yes** à l'étape suivante pour confirmer qu'il est correct.



n) Nous sommes maintenant invité à indiquer que les clés du serveur seront générées. Cliquer sur **Ok**. L'étape suivante nous indiquera que le serveur VPN vérifiera les mises à niveau sans surveillance et qu'un redémarrage périodique sera nécessaire. Activer les mises à niveau sans assistance. Les packages vont maintenant s'installer.



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



6. PostUp et PostDown

Cette section est délicate car ces lignes seront obligatoires si l'on est intéressé par l'utilisation **d'un profil VPN à tunnel partagé**. Si l'on utilise uniquement le tunnel complet, il est possible que tout fonctionne sans ajouter les lignes PostUp et PostDown au fichier de configuration.

Si on a uniquement l'intention d'utiliser le tunnel complet, on n'aura peut-être pas besoin d'ajouter ces lignes car tout le trafic sera acheminé via le Raspberry Pi.

Si vous ne les ajoute pas et que l'on ne peut pas accéder aux ressources locales ou se connecte à Internet lorsque l'on est connecté au VPN, ajouter ces deux lignes.

- **PostUp** : commande exécutée lorsque l'on se connecte au VPN WireGuard.

- **IPTables** : ce que le système doit faire avec certains paquets. Une table est créée avec ces règles afin que le système sache quoi faire lorsqu'il reçoit un paquet. Voici ce que signifient les différents paramètres de ligne de commande.

- **Masquerade** : l'adresse IP sera réécrite de la source (wg0) à la destination (eth0). En termes simples, le trafic semble provenir du Raspberry Pi par opposition à l'appareil source. Lorsque le trafic arrive et est envoyé à votre appareil client (où vous êtes connecté au VPN), l'adresse IP de destination du trafic sera réécrite de eth0 (Raspberry Pi) à wg0 (réseau WireGuard).

Remarque : On a toujours besoin d'une route statique si l'on souhaite accéder à nos clients VPN !

- **PostDown** : commande exécutée lorsque l'on se déconnecte du VPN WireGuard pour annuler tout ce que nous avons fait dans la commande **PostUp**. Ajouter ces lignes au fichier de configuration pour créer une table IP lorsque l'on se connecte à WireGuard et masquer son adresse IP.

```
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -A FORWARD -o wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -D FORWARD -o wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE
```

7. 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) WireGuard génère un certificat que l'on utilisera côté client. Pour récupérer ce certificat, saisir la commande suivante ou utiliser FileZilla :

```
# scp -p root@ip:/home/user/configs/client1.conf   /ip/dossier-posteclient
```

Remplacer **client1.conf** par le nom exact du fichier de certificat généré.

8. Modifier le fichier de configuration du client

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

```
[Interface]
PrivateKey = bGMzxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
Address = 10.6.0.2/24
DNS = 208.67.222.222, 208.67.220.220
[Peer]
PublicKey = EG20xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
PresharedKey = GwiFxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
Endpoint = xxx.xxx.xxx.xxx:51820
AllowedIPs = 0.0.0.0/0, ::0/0/
PersistentKeepalive = 25
```

a) Modifier la ligne **AllowedIPs** du fichier de conf. Par défaut, AllowedIPs est défini sur 0.0.0.0/0, cela qui signifie que tout le trafic passera par le réseau domestique (VPN à tunnel complet).

Pour changer cela afin que seul le trafic local soit envoyé via WireGuard, nous devons changer cette ligne par la plage IP locale. Pour la plupart, ce sera 192.168.1.0/24 ou 192.168.0.0/24. Si l'on souhaite que les clients VPN se parlent, on doit également ajouter le sous-réseau VPN (10.6.0.0/24).

b) Ajouter la ligne **PersistentKeepalive = 25**, pour que la connexion reste active même si elle n'est pas utilisée

Utiliser la commande suivante pour éditer le fichier.

```
$ nano /configs/client1.conf
```

9. Modifier le fichier de configuration du serveur

Un fichier de configuration pour le serveur WireGuard a été généré automatiquement lors de la création d'un utilisateur avec PiVPN : **wg0.conf**.

a) Ouvrir le fichier de configuration

```
# nano /etc/WireGuard/wg0.conf
```

b) Ajouter les lignes **PostUp**, **PostDown**.

c) On doit obtenir un fichier comme ci-dessous

```
[Interface]
PrivateKey = 3GyWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
Address = 10.6.0.1/24
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -A FORWARD -o wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -D FORWARD -o wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE
ListenPort = 51820
### begin client1 ###
[Peer]
PublicKey = sG3Bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
PresharedKey = GwiFxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
AllowedIPs = 10.6.0.2/32
### end client1 ###
```

10. 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
```

11. Démarrer le service WireGuard

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

```
# systemctl enable wg-quick@wg0
```

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

```
# wg show
```

On doit obtenir la sortie suivante :

```
interface: wg0
  public key: EG20xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
  private key: (hidden)
  listening port: 1194

peer: sG3Bxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=
  preshared key: (hidden)
  endpoint: xxx.xxx.xxx.xxx:42395
  allowed ips: 10.6.0.2/32
  latest handshake: 15 hours, 8 minutes, 4 seconds ago
  transfer: 272.38 MiB received, 468.53 MiB sent
```

12. 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/pi/configs/client1.conf
```

c) Copier les fichiers dans le dossier souhaité **C:\Documents\keys**
d) Ouvrir le client WireGuard
e) Importer le fichier **client1.conf**, puis cliquer sur le bouton **Activer**

13. 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 10.6.0.0/24 -o tun -j MASQUERADE
```

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

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

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 10.6.0.0/24 -o tun -j MASQUERADE
```

14. Commandes RaspberryPi

a) Liste des commandes basique à la gestion du serveur RaspberryPi

```
# wg-quick up ./wg0.conf   # activer WireGuard
# wg-quick down ./wg0.conf   # désactiver WireGuard
# systemctl start wg-quick@wg0   #démarrer WireGuard
# systemctl stop wg-quick@wg0   #arrête WireGuard
# 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
```

b) Changer le nom IP/DNS public du PiVPN après l'installation. Ouvrir le fichier :

```
# nano /etc/pivpn/wireguard/setupVars.conf
```

c) Modifier la ligne **pivpnHOST=[...]**

d) Les nouveaux clients que l'on génère utiliseront le nouveau point de terminaison, mais on doit modifier manuellement les clients existants

e) Ouvrir le fichier **client1.conf** et modifier la ligne **Endpoint = [...]:1194**
Enregistrez le fichier et reconnectez-vous

f) Autre méthode d'installation de **WireGuard**

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

15. Utilisation local du VPN

Une fois connecté au VPN **via un PC local**, impossible d'accéder aux périphériques réseaux locaux sans ajouter une route spécifique.

a) Ouvrir une invite de commande

b) Saisir la commande : **route -p add 192.168.1.X/24 10.0.0.2** (@IP du périphérique / @IP du VPN)

c) Mapper les périphériques via leur adresse IP

16. Conclusion

WireGuard 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 WireGuard** fonctionne aussi parfaitement sur une distribution Debian, Fedora ou une Ubuntu en mode VPS ou sur un ordinateur personnel.