Wireguard

*Création: lagrenouille *

- Objet : du tuto VPN: wireguard
- Niveau requis :
  débutant, avisé
- Suivi :
  à-placer
- Commentaires : *Contexte d'utilisation du sujet du tuto.* Fix Me!
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.
- sur le forum: https://debian-facile.org/viewtopic.php?pid=407983#p407983

Installation avec NetworkManager

```
apt install wireguard wireguard-tools
```

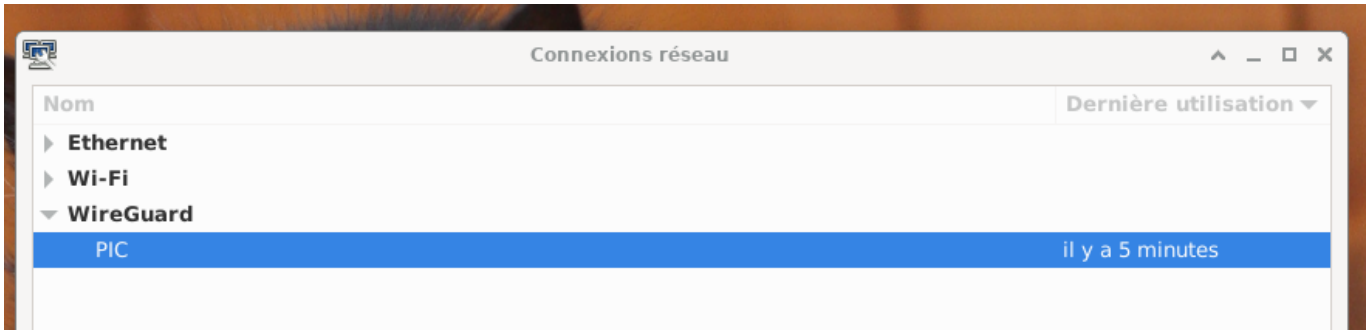
je pars du principe que vous avez déjà installé network-manager network-manager-gnome et fait en user nm-applet, et configurer vos réseaux, wifi etc...

création de vos clé, privée et public

```
mkdir wireguard
```

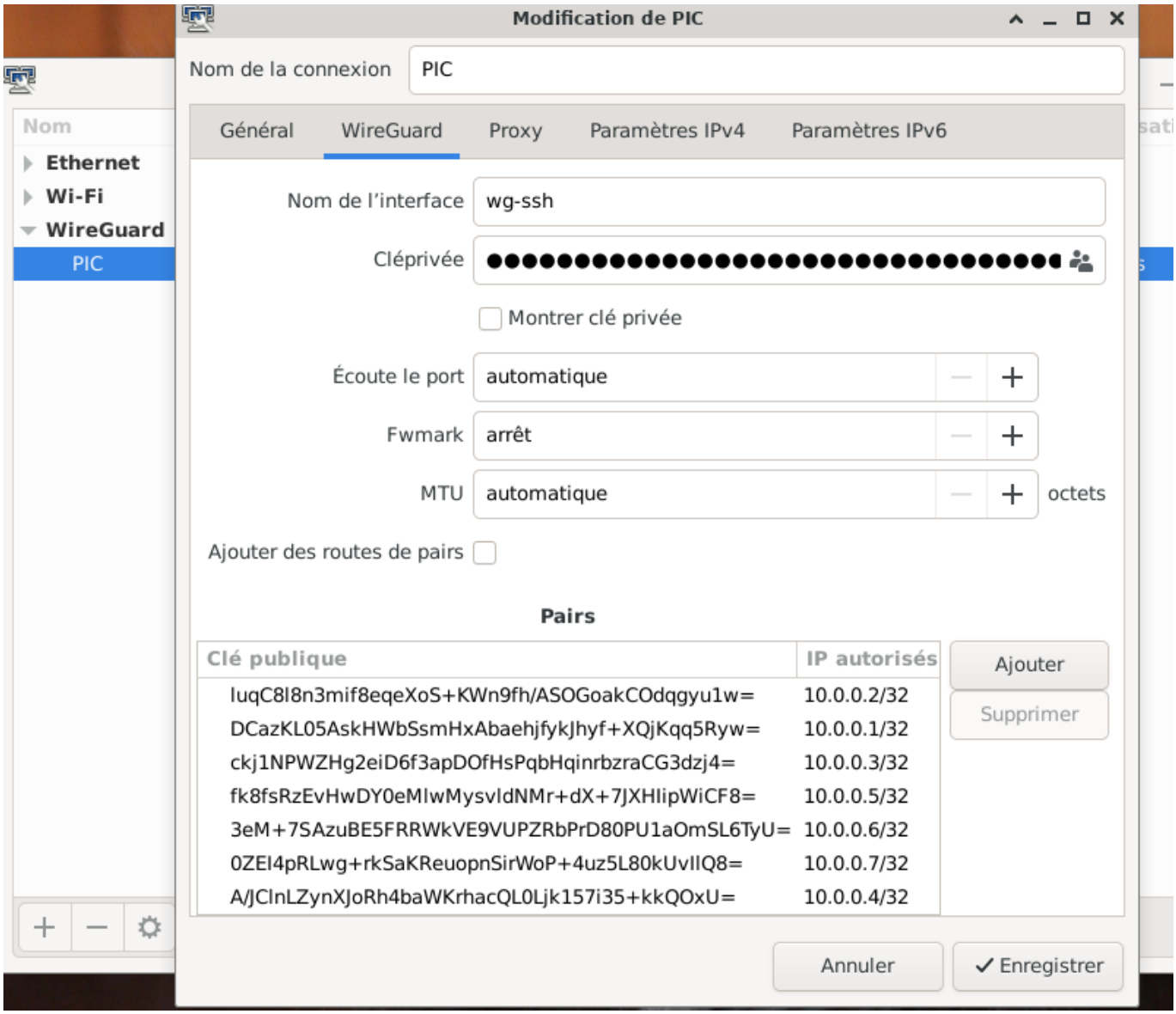
```
cd /home/user
mkdir wireguard
cd /home/user/wireguard
wg genkey | tee  wg-int.priv | wg pubkey > wg-int.pub
chmod 600 wg-int.priv
```

Un Endpoint est ce qu'on appelle une extrémité ou le points d'arrivée d'un canal de communication. une API, (interface de programmation d'application), permet d'échanger des informations et des données entre différents emplacements numériques, d'un point à un autre .



Donc, vous avez installé Network-manager, vous avez un logo dans le menu, un clique gauche ou milieu vous propose un petit menu dont un est nouvelle connexion ou Connexions VPN, mettez le nom que vous voulez pour votre connexion, puis cliquez sur : Configurez le VPN ici le nom est PIC, mais vous pouvez mettre n'importe quoi

tapez sur ce nom et une fenêtre va s'ouvrir pour la configuration



ici, il y a 7 entrées, simplement, c'est parce-que je me connecte via mon VPN sur 7 serveurs différents.

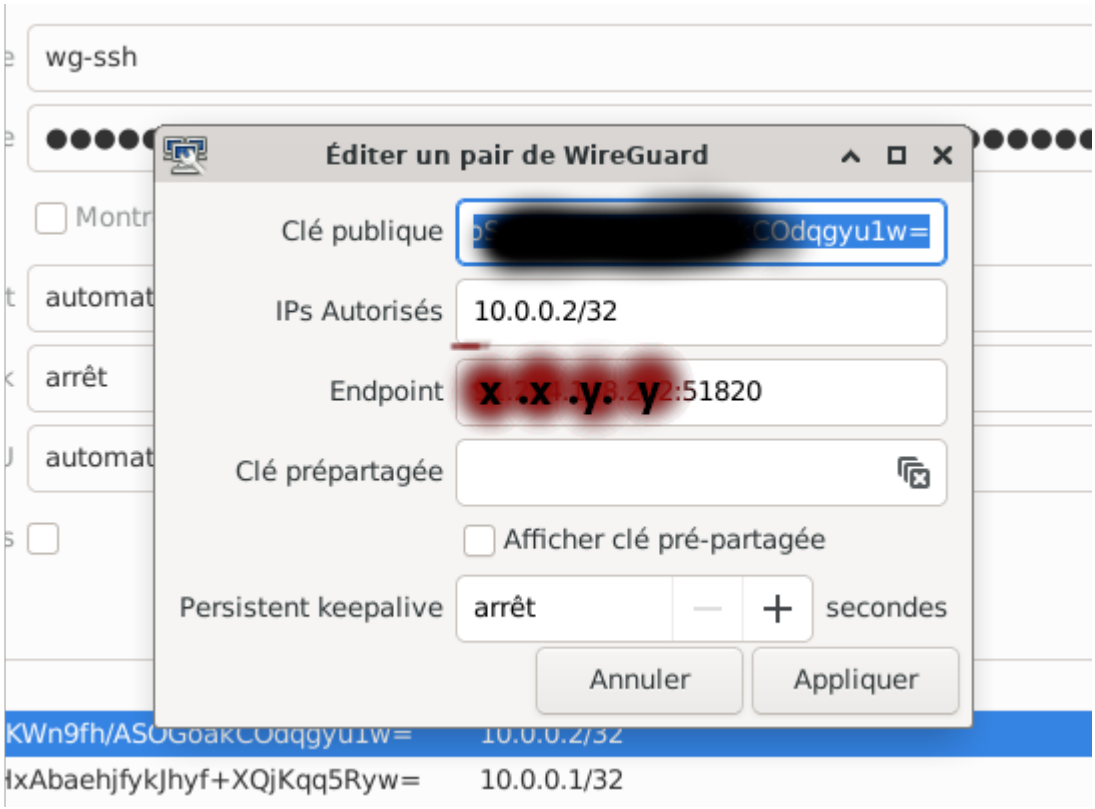
je pars du principe que le serveur est déjà paramétré, si non, on verra plus tard comment.

nom de l'interface: **wg-ssh**

mettez votre clé privée là où c'est demandé.

dans la case Pairs, cliquez sur Ajouter et mettez les clés publics et ip autorisées du serveur. et Enregistrer.

cliquez sur la ligne pour finir la conf, Entrez la clé publique du pair (serveur), l' IPs Autorisées et EndPoint



Dans proxy, ne faites rien, sauf si vous savez où vous allez

dans ipv4 vous mettez l'adresse, celle du client qui est sur votre serveur, ici 10.0.0.103 (103 c'est momo)

le masque sous réseau c'est 24 (soit l'équivalent de 255.255.255.0)

dans ipv6 ne mettez rien, laissé Désactivé

```
systemctl restart NetworkManager.service
```

vous voyez maintenant que votre applet network comporte un petit cadenas, qui vous dit que le vpn est installé et actif

ce qui ne veut pas dire qu'il est bien configuré

mettez vos clés dans .ssh la clé privé à 600 et la clé publique à 640

Vous pouvez simplifier la connexion en créant un fichier config dans votre .ssh

```
Host nom du serveur
```

```
  Hostname 10.0.0.2
  User   momo
  Port 2222
```

vous pouvez vous connecter avec

```
ssh nom du serveur
```

Installation en ligne de commande

Voici mes fichiers de conf sur mon PC de bureau

```
cat /etc/network/interfaces.d/wg-ssh
auto wg-ssh
iface wg-ssh inet static
address 10.0.0.104/24
pre-up ip link add dev wg-ssh type wireguard
pre-up wg setconf wg-ssh /etc/wireguard/wg-ssh.conf
post-down ip link delete wg-ssh
```

cat /etc/wireguard/wg-ssh.conf [Interface] PrivateKey = ma clé privée ListenPort = 51820

nom du serveur [Peer] Endpoint = 99.xxx.188.xxx:51820 PublicKey = celle du serveur AllowedIPs = 10.0.0.2/32

#nom du serveur [Peer] Endpoint = 99.xxx.188.xxx:51820 PublicKey = celle du serveur AllowedIPs =10.0.0.3/32

etc etc pour chaque serveur

mettre ses clés dans /etc/wireguard

et les mettre aussi dans .ssh

```
 chmod 600 pour clé privé , chmod 640 pour clé publique
```

et même fichier config dans li .ssh que la config avec NetworkManager

```
systemctl restart networking.service
```

Utilisation

créer un fichier config dans votre .ssh, style:

```
Host nom-du-serveur
   HostName 10.0.0.1
   User momo
```

```
Host nom-du-serveur
   HostName 10.0.0.2
   User momo
   Port 2222
```

Vous vous connecterez avec la commande simple:

```
ssh nom-du-serveur
```

pour plus de sécurité, utilisez une clé pour la connection ssh, avec ssh-keygen, ou autre, ça devrait ressembler à ceci:

```
ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key
(/home/myuser/.ssh/id_rsa):/home/momo/.ssh/id_rsa1
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/myuser/.ssh/id_rsa
Your public key has been saved in /home/myuser/.ssh/id_rsa.pub
```

N'oubliez pas de déposé la clé sur le serveur.

ensuite: ssh-add

```
ssh-add ~/.ssh/nom de la clé
paraphrase
xxxxxxxxx
un confirmation vous sera affiché sur la console.du style:
Identity added: /home/user/.ssh/nom de la clé
```

```
systemctl restart ssh.service
systemctl restartsshd.service
```

configuration du serveur

```
apt install wireguard wireguard-tools
```

Créez les clés et les déposées dans /etc/wireguard

```
vim /etc/wireguard/wg-ssh.conf
[Interface]
PrivateKey = clé privé du rerveur
ListenPort = 51820
```

dans le fichier /etc/wireguard/clients/, il faut mettre les gens qui ont accès, exemple: momo.conf

```
# momo arthur
[Peer]
PublicKey = clé publique de l'ordinateur à momo
AllowedIPs = 10.0.0.104/32
```

Certaines confs demandent à rajouter “Persistent-Keepalive”.. d'autres docs, disent : Le paramètre optionnel persistent-keepalive définit un intervalle en secondes dans lequel WireGuard envoie un paquet keep alive au serveur. Définissez ce paramètre si vous utilisez le client dans un réseau avec traduction d'adresse réseau (NAT) ou si un pare-feu ferme la connexion UDP après un certain temps d'inactivité.

```
**rappelez vous que ce 10.0.0.104 c'est dans le fichier
/etc/network/interfaces.d/wg-ssh, 104, c'est vous)**

auto wg-ssh
iface wg-ssh inet static
address 10.0.0.104/24
```

Si vous avez d'autres clients, alors re-belotte

PS: Endpoint = xxxxxxxx, si vous vous demandez ce que c'est, c'est l' ip du serveur (ip a)

```
Endpoint = ip serveur : port utilisé
```

On affiche les peers en root, avec:

```
#wg show

interface: wg-ssh
  public key: jWpHgcVEUIR9DhANXD/62sUu9fh/AZR6z0=
  private key: (hidden)
  listening port: 51820

peer: luqC8l8n89754Oh/ASOrkSaKReuopkCOdqgQL0Lyu1w=
  endpoint: x.x.x.x.252:51820
  allowed ips: 10.0.0.2/32
  latest handshake: 58 minutes, 9 seconds ago
  transfer: 167.01 KiB received, 190.57 KiB sent

peer: qinrbzraCG80PrsU1aOmSL6TyykJhyf+ghUgunaWKy=
  endpoint: x.x.x.x:51820
  allowed ips: 10.0.0.6/32
  latest handshake: 1 hour, 30 minutes, 19 seconds ago
  transfer: 63.13 KiB received, 81.73 KiB sent

peer: DCazKL05AskHWbSsmHxAbaehjfykJhyf+XQjKqq5Ryw=
  endpoint: x.x.x.x.165:51820
  allowed ips: 10.0.0.1/32

peer: DLSrtqiD6f3apDOfHsaWKrhacQL0Ljk15PqbHqgrzj4=
```

```
  endpoint: x.x.x.x.206:51820
  allowed ips: 10.0.0.3/32

peer: qA/lrs+tsgqilrsuhggiréucQL0Ljk157i35+kkQOxU=
  endpoint: x.x.x.x.248:51820
  allowed ips: 10.0.0.4/32

peer:GsRTnhutqev98tsffusaWKrhacQL NFQGtqin1HIipWiCF8=
  endpoint: x.x.x.x.243:51820
  allowed ips: 10.0.0.5/32

peer: 0ZEI4pRLwgirWoP+4uz5L8ytKCtqstM3hiU/jD0kUvIlQ8=
  endpoint: 192.168.0.2:51820
  allowed ips: 10.0.0.7/32
```

```
pour afficher le port
# wg show wg-ssh listen-port
51820
```

Si vous avez un fichier iptable, faites en sorte d'accepter le port 51820 et redémarrez les services:

```
systemctl restart iptables
systemctl restart fail2ban
```

petite explication

Si on lit Wikipèdia: *“VPN, virtual private network (réseau virtuel privé) La connexion entre les ordinateurs est gérée de façon transparente par un logiciel de VPN, créant un tunnel entre eux. Les ordinateurs connectés au VPN sont ainsi sur le même réseau local (virtuel), ce qui permet de passer outre d'éventuelles restrictions sur le réseau (comme des pare-feux ou des proxys)”*

WireGuard est un client vpn, un logiciel libre, un protocole réseau, de communication utilisant une technologie de chiffrement

le processus de tunnellisation est sécurisé, le VPN cache vos données derrière un code (il les chiffre) afin qu’aucune information ne s’échappe et que personne ne puisse vous identifier.

WireGuard est directement intégré dans le noyau Linux

ce protocole consiste à attribuer les plages d’adresses IP autorisées au sein d’un tunnel à la clé publique du partenaire de connexion.

Les paquets entrants du partenaire de connexion sont chiffrés à l’aide de la clé publique. Une fois cryptés, les paquets entrants sont

uniquement distribués s’ils proviennent d’une adresse IP correspondant à la clé. Dans le cas contraire, le paquet est rejeté.

Liens autre installation

https://blog.garamotte.net/posts/2020/08/29/fr-wireguard-with-systemd-networkd.html

https://www.malekal.com/utiliser-wg-quick-wireguard/

PS: Merci à l'équipe de notre chaton "https://le-pic.org" pour leurs aides