

# Wireguard - Mise en place

## Configuration serveur Wireguard

Documentation de mise en place et configuration du VPN wireguard

### Installation

Sous debian 12, wireguard est déjà installé dans les packages de base. S'il n'est pas installé :

```
apt update && apt install wireguard
```

### Configuration Serveur

Génération de fichier contenant la clé publique du serveur, ainsi que la clé privée :

```
root@wireguard:~# wg genkey |sudo tee /etc/wireguard/wg-srv-private.key | wg pubkey | sudo tee  
/etc/wireguard/wg-srv-public.key
```

Cela va générer 2 fichiers :

- **wg-srv-private.key** → Clé privée pour le serveur
- **wg-srv-public.key** → Clé publique, commune aux clients, à renseigner sur les configurations des clients

Ensuite, on va créer le fichier de configuration, contenant les paramètres du serveur, ainsi que la configuration des clients autorisés à se connecter dessus.

On va donc créer le fichier `/etc/wireguard/wg0.conf` :

```
root@wireguard:~# vi /etc/wireguard/wg0.conf  
[Interface]
```

```
Address = X.X.X./24
SaveConfig = true
ListenPort = 51820
PrivateKey = CléPrivéeServeurWg
```

Détail des options :

- **Address** : Adresse IP du serveur, utilisée pour la communication entre les clients et le serveur
- **SaveConfig** : Pour conserver la config quand le VPN est actif, protégeant la config après arrêt du tunnel
- **ListenPort** : port d'écoute de Wireguard
- **PrivateKey** : clé privée du serveur, pour le chiffrement des échanges

Ensuite on démarre notre tunnel wireguard :

```
sudo wg-quick up wg0
```

Après ça, on devrait voir apparaître notre tunnel, via la commande « ip -a » :

```
root@wireguard:~# ip a show dev wg0

13: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default qlen 1000
    link/none
    inet 10.8.8.254/24 scope global wg0
        valid_lft forever preferred_lft forever
```

Et aussi avec la commande « wg show », pour voir les interfaces de wireguard actives :

```
root@wireguard:~# wg show

interface: wg0
  public key: snWqZfuc54sEVrWBghBkxi1LkO1X4W1t0sjtJD5S9n8=
  private key: (hidden)
  listening port: 41820
```

Pour que notre configuration soit persistante et active après redémarrage du serveur, on crée un service associé :

```
sudo systemctl enable wg-quick@wg0.service
```

Pour que les paquets soient routés d'une interface à une autre, il faut activer l'IP Forwarding :

Dans le fichier **/etc/sysctl.conf**, décommenter la ligne suivante et modifier la valeur :

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

Il faut aussi activer l'IP masquerade, pour mettre en place le NAT sur la machine.

Configuration du parefeu nftable :

```
table inet filter {

    chain input {
        type filter hook input priority filter; policy accept;
        ip saddr 10.8.8.0/24 accept
        udp dport 41820 accept
        ct state vmap { invalid : drop, established : accept, related : accept }
        drop
    }

    chain output {
        type filter hook output priority filter; policy accept;
        ct state invalid drop
    }

    chain prerouting {
        type nat hook prerouting priority filter; policy accept;
    }

    chain postrouting {
        type nat hook postrouting priority filter; policy accept;
        ip saddr 10.8.8.0/24 oif "ens256" snat ip to 192.168.1.1
        ip saddr 10.8.8.0/24 masquerade
    }
}
```

## Ajout d'un client

Après avoir créer un couple de clé privée / publique sur le client, via wireguard sous windows, il faut ajouter le client sur le serveur.

Arrêt du service wgà:

```
systemctl stop wg-quick@wg0.service
```

Ajout du client dans le fichier /etc/wireguard.wg0.conf:

```
# Config VPN Exemple
[Peer]
PublicKey = 4eLau52rcERa1CTCbISUs3ysCnkpjqb1j0K2fhuKBTE=
AllowedIPs = 10.8.8.1/32
```

On relance le service de wireguard:

```
systemctl start wg-quick@wg0.service
```

On peut aussi utiliser la commande suivante sur le serveur, pour ajouter le client (évitant de redémarrer le service):

```
sudo wg set wg0 peer 4eLau52rcERa1CTCbISUs3ysCnkpjqb1j0K2fhuKBTE= allowed-ips 10.8.8.1/32
```

la configuration côté client, doit être la suivante:

```
[Interface]
PrivateKey = client_priv_key
Address = 10.8.8.1/24

[Peer]
PublicKey = server_pub_key
AllowedIPs = 10.8.8.0/24
Endpoint = ip_publique_serverur_wg:41820
```

Si tout est fonctionnel, notre client doit pouvoir pinguer le serveur.

## Docs supplémentaires

→ [Digital Ocean: wireguard](#)

→ [IT-Connect: Mise en place wireguard debian 11](#)

---

Revision #4

Created 19 June 2023 06:39:45 by Lauris\_Adm

Updated 4 July 2024 08:35:57 by Lauris\_Adm