Fiche Pratique N°20a : Configurer WireGuard sur YunoHost pour un accès VPN sécurisé V1.1

7 nov, 2026 · @Gabriel

Objectif : *Installer et configurer WireGuard sur votre serveur YunoHost pour créer un tunnel VPN sécurisé, qui donne un accès chiffré à vos services locaux depuis l'extérieur.*

- ***Public visé :*** *Intermédiaire (quelques bases en ligne de commande sont utiles, mais la configuration est guidée)*
- ***Temps estimé :*** *45 à 90 minutes*
- ***Difficulté :*** *★★★☆☆ (Moyen)*
- ***Prérequis :*** *un serveur YunoHost 12 à jour, un accès SSH administrateur, une box qui permet la redirection de ports*
- ***Fiches associées :*** *N°20 (YunoHost, section 4.5 sur l'IPv4), fiches suivantes (AdGuard Home, SearXNG)*

1. Pourquoi installer WireGuard en premier ?

WireGuard est la fondation de votre infrastructure sécurisée. C'est le tunnel VPN qui permet d'accéder à vos services locaux (AdGuard Home, SearXNG, Nextcloud...) depuis l'extérieur, sans les exposer directement sur Internet.

Les bénéfices concrets :

- **Accès sécurisé à distance :** accédez à vos services en 4G ou sur un Wi-Fi public, comme si vous étiez chez vous.
- **Chiffrement intégral :** tout le trafic du tunnel est chiffré.
- **Surface d'attaque réduite :** seul le port UDP de WireGuard est ouvert vers votre serveur pour l'usage VPN.
- **Simplicité et rapidité :** WireGuard est léger, plus rapide qu'OpenVPN et intégré au noyau Linux.

Cas d'usage typiques :

- Vous êtes en déplacement et voulez accéder à vos services locaux en toute sécurité.
- Vous voulez administrer votre serveur à distance sans ouvrir d'autres ports.
- Vous voulez protéger vos communications sur un réseau public (café, hôtel).
- Vous voulez rendre accessibles AdGuard Home, SearXNG et d'autres services depuis l'extérieur (les fiches suivantes s'appuient sur ce tunnel).

2. Architecture cible et prérequis

Architecture cible :

Élément	IP / Domaine	Rôle
Serveur	192.168.1.200 (IP locale fixe) ; 10.10.10.1 dans le tunnel	Serveur WireGuard, héberge les services locaux
Réseau VPN	10.10.10.0/24	Réseau privé des clients VPN

gabriel.ananda@mailo.com
CC BY SA

Fiche Pratique N°20a : Configurer WireGuard sur YunoHost pour un accès VPN sécurisé V1.1

Élément	IP / Domaine	Rôle
Client Linux	10.10.10.2/32	Poste de travail Linux (Mint, Ubuntu…)
Client Android	10.10.10.3/32	Téléphone

Prérequis :

- Un serveur sous **YunoHost 12** (la version stable actuelle, sur Debian 12), avec une IP fixe sur votre réseau local (ex. 192.168.1.200).
- Un accès administrateur en SSH sur ce serveur.
- Un ordinateur client sous Linux pour la configuration et les tests.
- Un téléphone Android pour tester l'accès distant (optionnel).
- **Une adresse IPv4 publique dédiée** sur votre box (chez Free : option « IPv4 Full Stack », voir fiche N°20, section 4.5). Avec une IPv4 partagée, le client ne pourra pas joindre votre serveur depuis l'extérieur.
- Une box qui permet la redirection de ports.

⚠️ **Le paquet YunoHost « WireGuard » (wireguard_ynh)** existe aussi, avec une interface web. Mais son installation a échoué sur YunoHost 12.1 (il appelait encore iptables) : cette fiche configure WireGuard à la main.

3. Installation et configuration du serveur

3.1. Installer WireGuard sur le serveur

Depuis un terminal sur le serveur YunoHost :

```
sudo apt update
sudo apt install wireguard wireguard-tools -y
```

3.2. Générer les clés du serveur

Ouvrez une session administrateur, puis générez les clés avec des droits restreints :

```
sudo -i
cd /etc/wireguard
umask 077
wg genkey | tee privatekey | wg pubkey > publickey
exit
```

- Vous devez voir : deux fichiers, **privatekey** et **publickey**, dans **/etc/wireguard** (vérifiez avec **sudo ls -l /etc/wireguard**).
- ⚠️ **umask** est une commande interne du shell : **sudo umask 077** (version précédente de la fiche) n'a aucun effet. D'où le passage par **sudo -i**.
- La clé **privée** ne doit jamais quitter le serveur. Seule la clé **publique** est donnée aux clients.

gabriel.ananda@mailo.com
CC BY SA

Fiche Pratique N°20a : Configurer WireGuard sur YunoHost pour un accès VPN sécurisé V1.1

3.3. Repérer le nom de votre interface réseau

```
ip route | grep default
```

- Vous devez voir : une ligne du type **default via 192.168.1.1 dev enp3s0** ; le nom après **dev** (ici **enp3s0**) est votre interface.

3.4. Créer le fichier de configuration /etc/wireguard/wg0.conf

Affichez la clé privée du serveur (**sudo cat /etc/wireguard/privatekey**), puis ouvrez le fichier :

```
sudo nano /etc/wireguard/wg0.conf
```

Copiez le contenu suivant, en remplaçant **CLE_PRIVEE_DU_SERVEUR** par la clé affichée et **enp3s0** par le nom de votre interface :

```
[Interface]
Address = 10.10.10.1/24
ListenPort = 51820
PrivateKey = CLE_PRIVEE_DU_SERVEUR

# Routage et NAT avec nftables (YunoHost 12.1 n'utilise plus iptables)
PostUp = nft add table inet wg-vpn
PostUp = nft add chain inet wg-vpn forward '{ type filter hook forward
priority 0; }'
PostUp = nft add rule inet wg-vpn forward iifname "wg0" accept
PostUp = nft add rule inet wg-vpn forward oifname "wg0" ct state
established,related accept
PostUp = nft add chain inet wg-vpn postrouting '{ type nat hook postrouting
priority 100; }'
PostUp = nft add rule inet wg-vpn postrouting ip saddr 10.10.10.0/24
oifname "enp3s0" masquerade
PostDown = nft delete table inet wg-vpn
```

- Les lignes **PostUp** créent, au démarrage du tunnel, une table nftables dédiée (**wg-vpn**) qui autorise le trafic du tunnel et le fait sortir sous l'adresse du serveur ; **PostDown** la supprime à l'arrêt. Elles ne touchent pas aux règles de YunoHost.
- Enregistrez (**Ctrl+O**, **Entrée**) et quittez (**Ctrl+X**).

3.5. Activer le routage IP (de façon permanente)

```
echo "net.ipv4.ip_forward=1" | sudo tee /etc/sysctl.d/99-wireguard.conf
sudo sysctl --system
```

- Vous devez voir : **net.ipv4.ip_forward = 1** dans la sortie.
- Un fichier dans **/etc/sysctl.d/** est la méthode la plus sûre (le fichier **/etc/sysctl.conf** n'est plus lu par défaut sur les Debian récentes).

gabriel.ananda@mailo.com
CC BY SA

Fiche Pratique N°20a : Configurer WireGuard sur YunoHost pour un accès VPN sécurisé V1.1

3.6. Ouvrir le port dans le pare-feu de YunoHost

YunoHost gère lui-même son pare-feu (nftables depuis YunoHost 12.1) : passez par sa commande plutôt que par des règles nft manuelles, qui seraient écrasées au rechargement.

```
sudo yunohost firewall allow UDP 51820
```

- Vous devez voir : « Succès ! Pare-feu rechargé ».
- Vous pouvez aussi passer par l'interface d'administration web de YunoHost (**Outils › Pare-feu**).

3.7. Démarrer le tunnel et l'activer au démarrage

```
sudo systemctl enable --now wg-quick@wg0
sudo wg show
```

- Vous devez voir : **interface: wg0** et **listening port: 51820**.

Les clients (peers) seront ajoutés dans les parties 4 et 5.

4. Configuration du client Linux (poste de travail)

4.1. Installer WireGuard sur le client

```
sudo apt update
sudo apt install wireguard wireguard-tools -y
```

4.2. Générer les clés du client

```
sudo -i
cd /etc/wireguard
umask 077
wg genkey | tee client-privatekey | wg pubkey > client-publickey
exit
```

Affichez la clé publique du client : **sudo cat /etc/wireguard/client-publickey**

4.3. Déclarer le client sur le serveur

Sur le **serveur**, ouvrez **/etc/wireguard/wg0.conf** et ajoutez à la fin :

```
[Peer]
# Client Linux
PublicKey = CLE_PUBLIQUE_DU_CLIENT_LINUX
AllowedIPs = 10.10.10.2/32
```

Rechargez la configuration sans couper le tunnel :

```
sudo systemctl reload wg-quick@wg0
```

4.4. Créer le fichier de configuration du client

Sur le **client**, créez **/etc/wireguard/wg0.conf** (**sudo nano /etc/wireguard/wg0.conf**) en remplaçant :

gabriel.ananda@mailo.com
CC BY SA

- **CLE_PRIVEE_DU_CLIENT** par le contenu de **client-privatekey** ;
- **CLE_PUBLIQUE_DU_SERVEUR** par le contenu du fichier **publickey** du serveur ;
- **votre-domaine.fr** par votre nom de domaine YunoHost (ou l'IP publique de votre box).

```
[Interface]
Address = 10.10.10.2/32
PrivateKey = CLE_PRIVEE_DU_CLIENT
DNS = 9.9.9.9
MTU = 1420

[Peer]
PublicKey = CLE_PUBLIQUE_DU_SERVEUR
AllowedIPs = 0.0.0.0/0, ::/0
Endpoint = votre-domaine.fr:51820
PersistentKeepalive = 25
```

- **DNS :** en attendant d'installer AdGuard Home sur le serveur (fiche suivante), utilisez un résolveur public respectueux comme **Quad9 (9.9.9.9)**. Vous remplacerez ensuite cette valeur par **10.10.10.1**. ⚠️ N'utilisez plus le DNS public de Mullvad : il a fermé le 2 novembre 2026.
- **AllowedIPs = 0.0.0.0/0, ::/0** fait passer **tout** le trafic par le tunnel. Pour ne faire passer que l'accès à votre réseau local, mettez **AllowedIPs = 10.10.10.0/24, 192.168.1.0/24**.
- **Endpoint** est l'adresse publique que le client utilise pour joindre le serveur depuis Internet.
- **MTU = 1420** est la valeur par défaut de WireGuard ; descendez à 1380 ou 1280 si certaines pages ne se chargent pas en 4G.

4.5. Démarrer le tunnel sur Linux Mint

Méthode recommandée sur Linux Mint : l'importer dans le gestionnaire de réseau

```
sudo nmcli connection import type wireguard file /etc/wireguard/wg0.conf
```

- Vous devez voir : « Connexion "wg0" ajoutée avec succès ». Le tunnel apparaît dans l'applet réseau, où vous pouvez l'activer ou le désactiver d'un clic.
- Le gestionnaire de réseau ignore les lignes **PostUp/PostDown** (il n'y en a pas côté client) et active la connexion automatiquement : désactivez-la quand vous êtes chez vous.

Méthode alternative : wg-quick

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

- Si vous obtenez **resolvconf: command not found** : n'installez pas **openresolv** (il entre en conflit avec systemd-resolved). Remplacez la ligne **DNS = 9.9.9.9** par ces deux lignes dans **[Interface]** :

```
PostUp = resolvectl dns %i 9.9.9.9; resolvectl domain %i ~.
PreDown = resolvectl revert %i
```

4.6. Tester le tunnel

⚠️ Testez **depuis l'extérieur** (partage de connexion 4G du téléphone, par exemple) : depuis votre propre réseau, beaucoup de box ne savent pas renvoyer une connexion vers leur propre IP publique.

gabriel.ananda@mailo.com
CC BY SA

```
ping -c 3 10.10.10.1
ping -c 3 192.168.1.200
sudo wg show
```

- Vous devez voir : des réponses aux deux pings, et dans **wg show** une ligne **latest handshake** récente.

5. Configuration du client Android (téléphone)

5.1. Installer l'application

Installez **WireGuard** depuis F-Droid (ou depuis Aurora Store).

5.2. Créer la configuration sur le téléphone (et générer ses clés)

1. Ouvrez l'application WireGuard et appuyez sur **+**.
2. Choisissez **Créer à partir de zéro**.
3. Donnez un nom au tunnel (ex. **maison**), puis appuyez sur l'icône de génération à côté de **Clé privée**.
    - Vous devez voir : une clé privée et une **clé publique** remplies automatiquement. Notez la clé publique (vous en aurez besoin à l'étape 5.3).
4. Remplissez la partie **Interface** :
    - **Adresses :** 10.10.10.3/32
    - **Serveurs DNS :** 9.9.9.9 (puis 10.10.10.1 après la fiche AdGuard Home)
5. Appuyez sur **Ajouter un pair** et remplissez :
    - **Clé publique :** la clé publique du serveur
    - **Point de terminaison :** votre-domaine.fr:51820
    - **Adresses IP autorisées :** 0.0.0.0/0, ::/0
    - **Keepalive persistant :** 25
6. Enregistrez (icône de disquette).

5.3. Déclarer le téléphone sur le serveur

Sur le **serveur**, ajoutez à la fin de **/etc/wireguard/wg0.conf** :

```
[Peer]
# Téléphone Android
PublicKey = CLE_PUBLIQUE_DU_TELEPHONE
AllowedIPs = 10.10.10.3/32
```

Puis rechargez : **sudo systemctl reload wg-quick@wg0**

5.4. Activer et tester

1. Coupez le Wi-Fi du téléphone (pour tester en 4G).
2. Activez le tunnel dans l'application WireGuard.

gabriel.ananda@mailo.com
CC BY SA

3. Dans le navigateur du téléphone, ouvrez **https://192.168.1.200** (ou l'adresse d'un de vos services).
    - Vous devez voir : la page de YunoHost (après un éventuel avertissement de certificat, normal quand on passe par une adresse IP).

6. Rendre le VPN accessible depuis Internet

Prérequis indispensable : une IPv4 publique dédiée (voir fiche N°20, section 4.5).

6.1. Configurer la redirection de port sur votre box

1. Connectez-vous à l'interface de votre box (souvent **http://192.168.1.1** ou **http://192.168.0.1**).
2. Cherchez la section **Redirection de ports** (Port Forwarding, NAT/PAT).
3. Créez une règle :
    - **Nom :** WireGuard
    - **Protocole :** UDP
    - **Port externe :** 51820
    - **IP interne :** 192.168.1.200 (l'IP de votre serveur)
    - **Port interne :** 51820 (identique au **ListenPort** de wg0.conf)

Si vous choisissez un autre port, modifiez-le aussi dans **ListenPort**, dans la commande **yunohost firewall allow** et dans l'**Endpoint** de tous les clients.

6.2. Vérifier que le serveur écoute

```
sudo ss -ulpn | grep 51820
```

- Vous devez voir : une ligne contenant **0.0.0.0:51820** (et souvent **[::]:51820**). WireGuard fonctionnant dans le noyau, aucun nom de programme n'est affiché : c'est normal.

7. Dépannage et vérifications

Problèmes fréquents :

Problème	Cause possible	Solution
Pas de « latest handshake » dans wg show	Endpoint faux, port non redirigé, IPv4 partagée	Vérifier l'Endpoint, la redirection 51820/UDP sur la box, l'IPv4 dédiée ; tester depuis la 4G
Ça marche en 4G mais pas depuis chez vous	La box ne gère pas le « NAT loopback »	Normal : chez vous, inutile d'activer le VPN
Ping vers 10.10.10.1 OK, mais pas vers 192.168.1.200	Routage ou NAT absent	Vérifier **sysctl net.ipv4.ip_forward** (doit valoir 1) et la présence de la table : **sudo nft list table inet wg-vpn**

gabriel.ananda@mailo.com
CC BY SA

Problème	Cause possible	Solution
Toujours bloqué malgré la table wg-vpn	Une chaîne « forward » du pare-feu YunoHost refuse le trafic	Regarder **sudo nft list ruleset** et chercher une chaîne **forward** avec **policy drop** ; demander de l'aide sur le forum YunoHost
Erreur iptables au démarrage du tunnel	Ancienne configuration (iptables) sur YunoHost 12.1	Remplacer les lignes PostUp/PostDown par la version nftables (§ 3.4)
resolvconf: command not found (client Linux)	wg-quick a besoin de resolvconf pour la ligne DNS	Utiliser l'import dans le gestionnaire de réseau, ou les lignes resolvectl (§ 4.5)
Pages qui ne se chargent pas en 4G	MTU trop élevée	Mettre **MTU = 1380** (ou 1280) côté client
La connexion se coupe en 4G	Box ou opérateur qui ferme les connexions inactives	**PersistentKeepalive = 25** côté client

Commandes utiles :

Action	Serveur	Client Linux (wg-quick)
Démarrer	sudo systemctl start wg-quick@wg0	sudo wg-quick up wg0
Arrêter	sudo systemctl stop wg-quick@wg0	sudo wg-quick down wg0
Recharger les pairs	sudo systemctl reload wg-quick@wg0	–
Statut	sudo wg show	sudo wg show
Journaux	sudo journalctl -u wg-quick@wg0 -f	–

8. Sécurité et maintenance

8.1. Sauvegarder les configurations

- **Serveur : /etc/wireguard/** (wg0.conf et clés), à copier dans un endroit **chiffré** : ces fichiers contiennent des clés privées.
- **Clients :** fichiers .conf, ou export depuis l'application Android (⋮ › **Exporter les tunnels**, archive zip).

8.2. Retirer un appareil perdu

Supprimez sa section **[Peer]** dans **/etc/wireguard/wg0.conf**, puis **sudo systemctl reload wg-quick@wg0** : l'appareil ne peut plus se connecter.

gabriel.ananda@mailo.com
CC BY SA

8.3. Surveiller les journaux

```
sudo journalctl -u wg-quick@wg0 -n 50 --no-pager
```

8.4. Mettre à jour

WireGuard est mis à jour avec le système. Utilisez de préférence l'outil de mise à jour de YunoHost (interface web **Mises à jour du système**, ou **sudo yunohost tools update** puis **sudo yunohost tools upgrade system**).

C'est terminé ! Vous avez un VPN WireGuard opérationnel, base sécurisée des fiches suivantes (AdGuard Home et SearXNG).

À retenir absolument

- **Une seule ouverture vers l'extérieur :** le port UDP 51820, redirigé par la box et ouvert avec **yunohost firewall allow UDP 51820**.
- **YunoHost 12.1 utilise nftables :** les anciennes règles iptables ne fonctionnent plus.
- **La clé privée ne quitte jamais l'appareil** qui l'a générée ; on n'échange que les clés publiques.
- **Testez depuis la 4G**, pas depuis votre propre réseau.

Test final :

1. **sudo wg show** sur le serveur affiche **listening port: 51820**.
2. Depuis la 4G, le client Linux ou Android active le tunnel.
3. **ping 192.168.1.200** répond depuis le client.
4. **sudo wg show** sur le serveur affiche un **latest handshake** récent pour chaque client.
5. Si tout fonctionne : **votre VPN est opérationnel** ✅

Références

- wg-quick(8) – page de manuel (PostUp, DNS et resolvconf, rechargement par syncconf)
- Forum YunoHost – Ouverture d'un port UDP pour WireGuard (yunohost firewall allow)
- Forum YunoHost – Serveur VPN sur YunoHost 12.1 (passage d'iptables à nftables)
- Forum YunoHost – nftables et yunohost-firewall sur YunoHost 12.1
- Forum YunoHost – YunoHost 13.0 alpha (Debian 13), déconseillé en production
- Paquet YunoHost wireguard_ynh (README)
- GNOME – WireGuard dans NetworkManager (import d'un fichier wg-quick)
- LumaDock – Corriger « resolvconf: command not found » avec systemd-resolved
- Mullvad – Fermeture du DNS public, parrainage de Quad9
- WireGuard pour Android sur F-Droid

gabriel.ananda@mailo.com
CC BY SA