Mise en place de Wireguard Serveur

> WireGuard® est un VPN extrêmement simple, rapide et moderne qui utilise une cryptographie de pointe. Il se veut considérablement plus performant qu'OpenVPN.
>
> WireGuard est conçu comme un VPN à usage général fonctionnant aussi bien sur des interfaces embarquées que sur des superordinateurs, adapté à de nombreuses circonstances différentes.
>
> Initialement publié pour le noyau Linux, il est désormais multiplateforme (Windows, macOS, BSD, iOS, Android) et largement déployable. Il fait actuellement l'objet d'un développement intensif, mais il peut déjà être considéré comme la solution VPN la plus sûre, la plus facile à utiliser et la plus simple du secteur.

image docker : https://github.com/linuxserver/docker-wireguard

Utilisation de l'image docker

Mode serveur

Si la variable d'environnement `PEERS` est définie à un nombre ou à une liste de chaînes séparées par une virgule, le conteneur fonctionnera en mode serveur et les confs serveur et pair/client nécessaires seront générées.

Les qr codes de configuration des pairs/clients seront affichés dans le journal de docker si `LOG_CONFS` est défini à `true`. Ils seront également sauvegardés au format texte et png sous `/config/peerX` dans le cas où `PEERS` est une variable et un entier ou `/config/peer_X` dans le cas où une liste de noms a été fournie à la place d'un entier.

Les variables `SERVERURL`, `SERVERPORT`, `INTERNAL_SUBNET`, `PEERDNS`, `INTERFACE`, `ALLOWEDIPS` et `PERSISTENTKEEPALIVE_PEERS` sont des variables optionnelles utilisées pour le mode serveur. Toute modification de ces variables d'environnement déclenchera la régénération des confs du serveur et des pairs. Les confs pairs/clients seront recréées avec les clés privées/publiques existantes.
--> *Supprimez les dossiers des pairs pour que les clés soient recréées en même temps que les confs.*

Pour ajouter ultérieurement d'autres pairs/clients, vous devez incrémenter la variable d'environnement PEERS ou ajouter d'autres éléments à la liste et recréer le conteneur.

Pour afficher à nouveau les codes QR des pairs actifs, vous pouvez utiliser la commande suivante et lister les numéros de pairs comme arguments :
`docker exec -it wireguard /app/show-peer 1 4 5` ou `docker exec -it wireguard /app/show-peer myPC myPhone myTablet` (Gardez à l'esprit que les codes QR sont également stockés sous forme de PNG dans le dossier config).

Les modèles utilisés pour la configuration du serveur et des pairs sont enregistrés dans le dossier `/config/templates`. Les utilisateurs avancés peuvent modifier ces modèles et forcer la génération de confs en supprimant `/config/wg0.conf` et en redémarrant le conteneur.

docker-compose :

```
version: "2.1"


services:


  wireguard:
    image: lscr.io/linuxserver/wireguard:latest
    container_name: wireguard
    cap_add:
      - NET_ADMIN
      # - SYS_MODULE  # optional already active
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/Paris
      - SERVERURL=vpn.domaine-name.com
      - SERVERPORT=51820
      - PEERS=pcApple,pcGamer,telUser1,castHome
      - PEERDNS=auto
      #- INTERNAL_SUBNET=10.13.13.0 #optional
      #- ALLOWEDIPS=0.0.0.0/0 #optional
      #- PERSISTENTKEEPALIVE_PEERS= #optional
```

```
      - LOG_CONFS=true
    volumes:
      - /etc/container-conf/wireguard/appdata/config:/config
      - /etc/container-conf/wireguard/lib/modules:/lib/modules
    ports:
      - 51820:51820/udp
    restart: unless-stopped

volumes:
 wireguard:
```

Paramètres

Les images des conteneurs sont configurées à l'aide de paramètres transmis au moment de l'exécution (tels que ceux indiqués ci-dessus). Ces paramètres sont séparés par deux points et indiquent `<external>:<internal>` respectivement.

Par exemple, `-p 8080:80` expose le port `80` à l'intérieur du conteneur pour qu'il soit accessible à partir de l'IP de l'hôte sur le port `8080` à l'extérieur du conteneur.

Parameter	Function
`51820/udp`	wireguard port
`PUID=1000`	for UserID - see below for explanation
`PGID=1000`	for GroupID - see below for explanation
`TZ=Etc/UTC`	specify a timezone to use, see this list.
`SERVERURL=vpn.domain.com`	IP externe ou nom de domaine de l'hôte docker. Utilisé en mode serveur. Si la valeur est auto, le conteneur essaiera de déterminer et de définir l'IP externe automatiquement.
`SERVERPORT=51820`	Port externe pour l'hôte Docker. Utilisé en mode serveur.
`PEERS=1`	Nombre de pairs pour lesquels créer des confs. Requis pour le mode serveur. Peut également être une liste de noms : myPC,myPhone,myTablet (alphanumérique uniquement).
`PEERDNS=auto`	Serveur DNS défini dans les configurations homologue/client (peut être défini comme 8.8.8.8). Utilisé en mode serveur. La valeur par défaut est auto, ce qui utilise le DNS de l'hôte wireguard docker via la redirection CoreDNS incluse.

Parameter	Function
`INTERNAL_SUBNET=10.13.13.0`	Sous-réseau interne pour le wireguard, le serveur et les pairs (à ne modifier qu'en cas de conflit). Utilisé en mode serveur.
`ALLOWEDIPS=0.0.0.0/0`	Les IP/plages que les homologues pourront atteindre en utilisant la connexion VPN. Si elle n'est pas spécifiée, la valeur par défaut est : '0.0.0.0/0, ::0/0' Tout le trafic transitera par le VPN. Si vous souhaitez un tunnel divisé, réglez cette valeur sur les IP que vous souhaitez utiliser dans le tunnel ET sur l'IP du WG du serveur, par exemple 10.13.13.1.
`PERSISTENTKEEPALIVE_PEERS=`	Réglé sur tous ou sur une liste de pairs séparés par des virgules (par exemple 1,4,laptop) pour que le serveur wireguard envoie des paquets keepalive aux pairs listés toutes les 25 secondes. Utile si le serveur est accessible via un nom de domaine et possède une IP dynamique. Utilisé uniquement en mode serveur.
`LOG_CONFS=true`	Les codes QR générés seront affichés dans le journal du docker. Mettez false pour ignorer la sortie du journal.
`/config`	Contains all relevant configuration files.
`/lib/modules`	Host kernel modules for situations where they're not already loaded.

Ouverture des ports

Maintenant que le serveur est configuré il ne faut pas oublier d'ouvrir le port concerné.
Dans notre cas nous devons ouvrir le port 51820 sur le serveur et sur la box internet car le serveur se trouve derrière une box internet.

Sur le serveur (avec `ufw`) :

```
sudo ufw allow 51820
```

Sur la box internet : *(Voir comment configurer la redirection de port sur le panneau de configuration de la box internet de votre fournisseur)*

Accès Container Docker via Wireguard

Pour accéder aux containers Docker du serveur via Wireguard il est nécessaire d'ouvrir le par-feu comme ci dessous :

Adresse IP du container Wireguard : 192.168.128.2 (`docker inspect <containername>`)

Via UFW : `sudo ufw allow from 192.168.128.2`

Ce qui donne la configuration suivante :

```
> ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[1] 22/tcp                     ALLOW IN    Anywhere
[2] 80/tcp                     ALLOW IN    Anywhere
[3] 443                        ALLOW IN    Anywhere
[4] Anywhere                   ALLOW IN    192.168.128.2
[5] 51820                      ALLOW IN    Anywhere
[6] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[7] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[8] 443 (v6)                   ALLOW IN    Anywhere (v6)
[9] 51820 (v6)                 ALLOW IN    Anywhere (v6)
```

Revision #13
Created 2023-06-12 14:22:53 UTC by gpatruno
Updated 2024-12-02 15:46:24 UTC by gpatruno