

# docker compose vpn Wireguard

## Datacode

---

La livraison du code contient deux docker compose:

- **docker-compose.yml**: Contenant une configuration générique de l'implémentation d'un container wireguard avec un container apache-php et un container postgresql
- **docker-compose-datacode.yml**: Adaptation du docker-compose fourni pour y intégrer le vpn Wireguard.

### Principe de fonctionnement

Le principe de fonctionnement est simple, le docker compose va créer 3 containers ayant un réseau commun appelé **wgnet**.

Ensuite lors de la creation du container de la solution VPN WIREGUARD, ce dernier disposera d'un acces publique depuis l'adresse ip du serveur hôte (cf variable **SERVERURL** à renseigner dans le docker-compose)

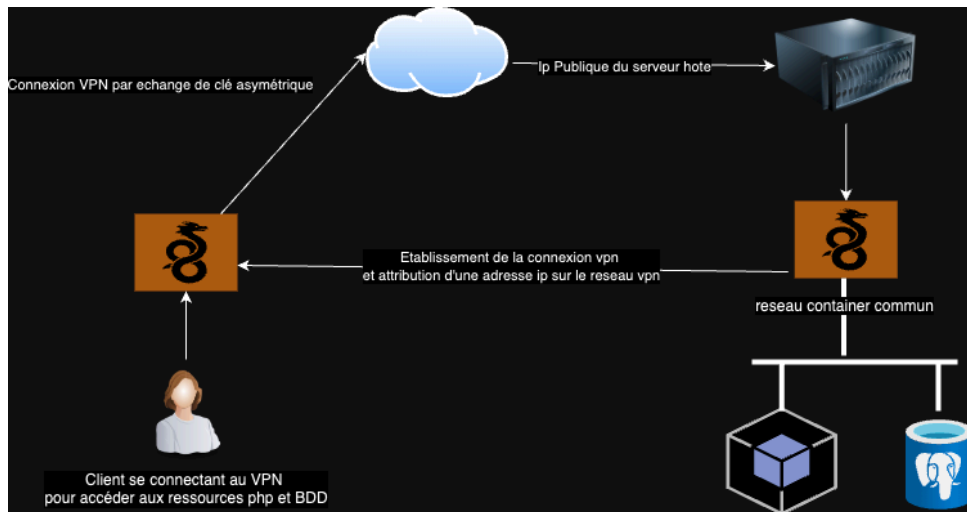
Puis ce dernier va créer un réseau dédié à la connexion VPN (cf variable **INTERNAL\_SUBNET** dans le docker-compose)

La configuration permettra le transfère des flux du réseau dédié vpn vers le réseau dédié des containers docker.

Afin de maitriser le plan ip sur le reseau des containers, il a etait décider de fixer les ips. aussi nous aurons par exemple :

- ip 10.0.0.2 == Ip du serveur VPN Wireguard sur le reseau docker
- ip 10.0.0.3 == Ip du serveur Apache-PHP
- ip 10.0.0.4 == Ip du serveurs Postgresql

### Schémas de fonctionnement



## Eléments important de paramétrage du container WireGuard

Ci-dessous les éléments importants à paramétrer dans le docker-compose pour la mise en place de Wireguard:

- sur la section environment:
  - SERVERURL : Correspond à l'adresse ip publique où le nom fqdn du serveur hôte
  - SERVERPORT: Port d'écoute du serveur vpn
  - PEERS: il s'agit d'une liste des clients du vpn, cela permet la génération de leur configuration
  - INTERNAL\_SUBNET: Il s'agit du subnet réseau dédié au vpn, pour chaque clients et serveur une adresse ip sera attribuée dans ce réseau.
  - ALLOWEDIPS: Définit les subnets autorisés via le réseau VPN
- sur la section sysctls:
  - net.ipv4.ip\_forward : permet d'autoriser le forward de trames vers d'autres réseaux, dans notre cas pour accéder au réseau commun docker.
  - net.ipv4.conf.all.src\_valid\_mark : permet de marquer comme valide les sources de trafic arrivant sur le container docker.
- sur la section cap\_add: Cette section permet de donner des privilèges au containers docker, sur le serveur hôte.
  - NET\_ADMIN : permet de donner des droits d'opération sur des éléments de gestion du réseau, cette directive est

nécessaire pour gérer les transferts de flux entre les différents réseaux utilisés

- SYS\_MODULE: permet de gérer des modules kernels

## Connexion au vpn avec un client Wireguard

La documentation officielle de Wireguard ci dessous explique comment effectuer la connexion au vpn via les différentes versions de client disponible:

[documentation\\_officielle\\_wireguard](#)

- MacOS
- Linux
- Windows
- IOS
- Android

Il y a généralement deux possibilités pour charger la configuration vpn, pour cela il faudra récupérer au niveau du volume du container Wireguard le fichier de configuration correspondant au client que l'on souhaite configurer, par exemple le client peer\_client1. La configuration se trouve dans le dossier peer\_client1 qui se trouve dans le répertoire configs du container Wireguard.

il y a deux fichiers permettant de récupérer la configuration :

- Soit un fichier png contenant le QRCODE de la configuration (recommandé pour Android et IOS)



- Soit un fichier de configuration à importer dans le client Wireguard (recommandé pour linux, windows et macOS)

[exemple d'un fichier de configuration](#)