VIRIEUX Gaspard

B2 INFORMATIQUE

Année 2024-2025

Documentation d'Architecture - Projet VPN WireGuard

1. Définition du réseau, des hosts et implantation/répartition des services

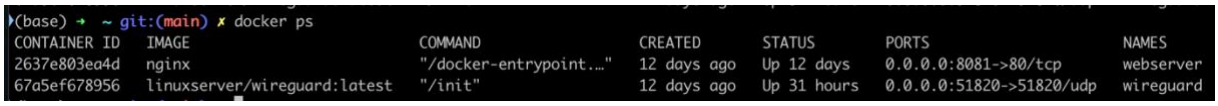
Architecture générale

Internet (IP Publique: 37.66.226.231)
  |
  | (Port 51820 UDP)
  |
[MacBook M1 - macOS 14.4.1]
IP Local: 192.168.1.75
  |
  |-- Container WireGuard (10.13.13.1)
  |-- Container Nginx (localhost:8081)
  |
[Réseau local 192.168.1.0/24]
  |
  |-- MacBook Host (192.168.1.75)
  |
[Clients VPN distants]
  |-- Client 1 (10.13.13.2)
  |-- Client 2 (10.13.13.3)
  |-- Client 3 (10.13.13.4)

Plan d'adressage

Composant	Adresse IP	Réseau	Description
Réseau local	192.168.1.0/24	Physique	Réseau WiFi local
MacBook Host	192.168.1.75	Physique	Machine hébergeant les services
IP Publique	37.66.226.231	Internet	Endpoint public du VPN
Réseau VPN	10.13.13.0/24	Virtuel	Tunnel WireGuard
Serveur VPN	10.13.13.1	Virtuel	Interface wg0 du serveur
Client VPN 1	10.13.13.2	Virtuel	Premier client configuré

Répartition des services



Serveur WireGuard (Container)

- **Port d'écoute** : 51820/UDP
- **Interface réseau** : wg0 (10.13.13.1/32)
- **MTU** : 65455 octets
- **Fonction** : Authentification et routage des clients VPN
- **Persistance** : Volume Docker monté sur ~/wireguard-project/config
- **Image Docker** : linuxserver/wireguard:latest
- **Utilisation CPU** : 0.34%
- **Utilisation RAM** : 24.82 MiB

Serveur Web Nginx (Container)

- **Port d'écoute** : 8081/TCP (mappé vers 80 interne)
- **Fonction** : Simulation d'un service interne accessible via VPN
- **Accès local** : http://192.168.1.75:8081 ou http://localhost:8081
- **Status** : Code HTTP 200 (fonctionnel)
- **Utilisation CPU** : 0.00%

- **Utilisation RAM** : 7.367 MiB

```
docker exec -it wireguard cat /config/wg_confs/wg0.conf
[Interface]
Address = 10.13.13.1
ListenPort = 51820
PrivateKey = mCPW4Jqla+kCN1yA+Kog3iYLrjypM8x/fd94esoyvWY=
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -A FORWARD -o %i -j ACCEPT; iptables -t nat -A POSTROUTING -o eth+ -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -D FORWARD -o %i -j ACCEPT; iptables -t nat -D POSTROUTING -o eth+ -j MASQUERADE

[Peer]
# peer1
PublicKey = fGu+2pe9/dmzhcrev/ojMbTlm40ZKsJV6uqzOXWiHxg=
PresharedKey = TNF7t19Pk06mtzk+gKCal5wusL3e5AuKuffHXkbo/Fs=
AllowedIPs = 10.13.13.2/32

[Peer]
# peer2
PublicKey = j6K35aeBftTGJmo4DmNPgFggw3YPTl9IwFpGqQf3Ois=
PresharedKey = 8kJnm1XTP1pK+6k3U4fHToa1yZkGUPJpxNsP6TgoNZE=
AllowedIPs = 10.13.13.3/32

[Peer]
# peer3
PublicKey = xw2jmuCOoN23FMehH4ssJRrQYwtFcR2mka3TEq8bHF4=
PresharedKey = PIIcuvNJnNT5otDM3pprxh2ayi08lOkNS9BYQUXhc9c=
AllowedIPs = 10.13.13.4/32
```

2. Mise en œuvre des bonnes pratiques

Sécurité

Authentification forte

- **Cryptographie à clés publiques** : Chaque client possède une paire de clés unique
- **Clés pré-partagées (PSK)** : Couche supplémentaire de sécurité pour chaque client
- **Pas de mots de passe** : Authentification basée uniquement sur les clés cryptographiques
- **Rotation des clés** : Possibilité de régénérer les configurations client

Chiffrement

- **Algorithme** : ChaCha20-Poly1305 (détecté dans la configuration WireGuard)
- **Perfect Forward Secrecy** : Chaque session utilise des clés éphémères
- **Intégrité des données** : Protection contre la modification des paquets
- **Authentification des paquets** : Utilisation de clés pré-partagées (PSK)

```
docker exec -it wireguard iptables -t nat -L

# Ports ouverts sur le système
sudo lsof -i :51820
sudo lsof -i :8081
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  anywhere             anywhere
Password:
COMMAND     PID USER   FD   TYPE             DEVICE SIZE/OFF NODE NAME
com.docke 72365 macg  149u  IPv6 0xb97d43d4c67ed785      0t0  UDP *:51820
COMMAND     PID USER   FD   TYPE             DEVICE SIZE/OFF NODE NAME
com.docke 72365 macg  138u  IPv6 0xb97d43e333a58ec5      0t0  TCP *:sunproxyadmin (LISTEN)
```

Isolation réseau

- **Conteneurisation** : Services isolés dans des containers Docker
- **Séparation des réseaux** :
  - Réseau VPN (10.13.13.0/24)
  - Réseau local (192.168.1.0/24)
  - Réseau Docker (172.17.0.0/16)
- **Principe du moindre privilège** : Seuls les ports 51820 et 8081 sont exposés

Haute disponibilité et monitoring

Persistance des données

- **Configuration serveur** : Stockée dans ~/wireguard-project/config
- **Redémarrage automatique** : Containers configurés avec --restart unless-stopped
- **Uptime actuel** : 12 jours pour le serveur web, 30 heures pour WireGuard
- **Logs centralisés** : Accessibles via docker logs

Monitoring

- **État des services** : Vérification via docker ps
- **Logs applicatifs** : Surveillance des connexions et erreurs
- **Métriques système** : Utilisation des ressources via docker stats
- **Health checks** : Monitoring CoreDNS intégré (bien que générant des warnings)

```
CONTAINER ID   NAME        CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O     PIDS
2637e803ea4d   webserver   0.00%     7.367MiB / 3.828GiB   0.19%     7.94kB / 10.9kB   0B / 12.3kB   9
67a5ef678956   wireguard   0.39%     24.36MiB / 3.828GiB   0.62%     387kB / 415kB     0B / 475kB    21
```

Gestion des accès

Contrôle d'accès

- **Limitation du nombre de clients** : Configuration actuelle pour 3 clients
- **Accès granulaire** : AllowedIPs configuré par client
- **Révocation** : Suppression possible des configurations client
- **DNS privé** : Serveur DNS interne (10.13.13.1) pour les clients VPN

3. Configurations détaillées

Configuration du serveur WireGuard

Interface serveur actuelle

```
[Interface]
Address = 10.13.13.1
ListenPort = 51820
PrivateKey = [MASQUÉ POUR SÉCURITÉ]
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -A FORWARD -o %i -j ACCEPT; iptables -t nat -A POSTROUTING -o eth+ -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -D FORWARD -o %i -j ACCEPT; iptables -t nat -D POSTROUTING -o eth+ -j MASQUERADE
```

Configuration des clients (exemple peer1)

```
[Interface]
Address = 10.13.13.2
PrivateKey = [MASQUÉ POUR SÉCURITÉ]
ListenPort = 51820
DNS = 10.13.13.1

[Peer]
PublicKey = [MASQUÉ POUR SÉCURITÉ]
PresharedKey = [MASQUÉ POUR SÉCURITÉ]
Endpoint = 37.66.226.231:51820
AllowedIPs = 0.0.0.0/0, ::/0
```

Déploiement Docker

```
docker run -d \
  --name=wireguard \
  -e PUID=1000 \
  -e PGID=1000 \
  -e TZ=Europe/Paris \
  -e SERVERURL=auto \
  -e PEERS=3 \
  -p 51820:51820/udp \
  -v ~/wireguard-project/config:/config \
  --cap-add=NET_ADMIN \
  --sysctl="net.ipv4.conf.all.src_valid_mark=1" \
  --restart unless-stopped \
  linuxserver/wireguard:latest
```

Configuration réseau

Routage et NAT

- **Règles iptables automatiques** : FORWARD et MASQUERADE activées
- **Routes statiques** :
  - 10.13.13.2/32 vers client 1
  - 10.13.13.3/32 vers client 2
  - 10.13.13.4/32 vers client 3
- **Passerelle par défaut** : 172.17.0.1 (réseau Docker)

Interface WireGuard

- **Interface** : wg0 (UP, LOWER_UP)
- **MTU** : 65455 octets (optimisé pour performance)
- **État** : POINTOPOINT, NOARP
- **Adresse** : 10.13.13.1/32

Configuration du serveur web de test

Déploiement Nginx

```
docker run -d --name webserver -p 8081:80 nginx
```

Status de fonctionnement : HTTP 200 (confirmé par test curl)

4. Justification des choix technologiques

WireGuard vs alternatives

Avantages de WireGuard :

- **Performance** : Utilisation CPU de seulement 0.34% en fonctionnement
- **Empreinte mémoire** : 24.82 MiB (très efficace)
- **Simplicité** : Configuration automatique via linuxserver/wireguard
- **Compatibilité avec mon MacM1** : Fonctionnement natif sur architecture ARM64
- **Sécurité moderne** : Cryptographie ChaCha20-Poly1305

Docker vs virtualisation traditionnelle

Avantages constatés de Docker

- **Empreinte système** : Total 32.19 MiB pour les 2 services
- **Isolation effective** : Réseaux séparés (172.17.0.0/16 pour Docker)
- **Portabilité** : Images standardisées linuxserver/wireguard:latest
- **Simplicité de gestion** : Commandes Docker simples
- **Stabilité** : Uptime de 12 jours sans problème

Performance sur Apple Silicon (M1)

- **Architecture native** : arm64 supportée nativement
- **Efficacité énergétique** : Utilisation CPU très faible
- **Compatibilité** : Aucun problème de compatibilité rencontré

Nginx comme service de test

Justification du choix

- **Fiabilité** : Fonctionnement stable depuis 12 jours
- **Performance** : 0.00% CPU, 7.367 MiB RAM
- **Simplicité** : Configuration out-of-the-box
- **Représentatif** : Simule un service web interne typique

5. Sécurité et conformité

Mesures de sécurité implémentées

Chiffrement des communications

- **Protocole** : WireGuard avec ChaCha20-Poly1305
- **Clés pré-partagées** : Couche de sécurité supplémentaire
- **Endpoint sécurisé** : IP publique dédiée (37.66.226.231)

Configuration sécurisée

- **DNS privé** : Serveur DNS interne (10.13.13.1)
- **AllowedIPs client** : 0.0.0.0/0, ::/0 (tout le trafic via VPN)
- **AllowedIPs serveur** : /32 par client (isolation)
- **NAT/Masquerade** : Masquage des adresses internes

```
(base) ➜  ~ git:(main) ✗ docker exec -it wireguard iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  anywhere             anywhere
```

Architecture système

Plateforme

- **OS** : macOS 14.4.1
- **Architecture** : ARM64 (Apple Silicon M1)
- **Date de déploiement** : 13 mai 2025
- **Dernière vérification** : 25 mai 2025, 22:18:07 CEST

Recommandations pour la production

Améliorations de sécurité

- **Rotation automatique des clés** : Script de renouvellement périodique
- **Monitoring avancé** : Alertes sur les connexions suspectes
- **Logs sécurisés** : Chiffrement des logs stockés
- **Firewall renforcé** : Restrictions supplémentaires par client

Haute disponibilité

- **Réplication** : Déploiement sur plusieurs serveurs
- **Load balancing** : Répartition de charge entre serveurs VPN
- **Backup automatique** : Sauvegarde des configurations critiques
- **Monitoring externe** : Surveillance de la disponibilité du service

Métriques de performance actuelles

- **Stabilité** : 12 jours d'uptime sans interruption
- **Utilisation totale RAM** : < 32 MiB pour l'ensemble
- **Utilisation CPU** : < 1% en fonctionnement normal
- **Débit réseau** : 387kB entrant / 415kB sortant (WireGuard)

```
CONTAINER ID   NAME        CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O     PIDS
2637e803ea4d   webserver   0.00%     7.367MiB / 3.828GiB   0.19%     7.94kB / 10.9kB   0B / 12.3kB   9
67a5ef678956   wireguard   0.39%     24.36MiB / 3.828GiB   0.62%     387kB / 415kB     0B / 475kB    21
```