

# Déploiement d'un Serveur VPN

Déploiement d'un serveur VPN sous OpenVPN UI



OpenVPN UI

Préparé par :  
Noah Louineau

# Description du Projet

Je me suis donné comme projet de déployer un serveur VPN sur une VM hébergé pour le moment sur mon ordinateurs personnel, afin d'avoir une solution d'accès à mon réseau local depuis internet.

L'idée de serveur VPN m'est venu après que j'ai souhaité posséder un serveur TrueNAS dans le but de stocker mes fichiers personnels et de pouvoir y accéder depuis internet.

## Introduction

Je recherchais un serveur VPN offrant une solution gratuite et open source, avec une interface web ou graphique pour faciliter la création et la gestion des connexions du serveur VPN. J'ai trouvé un projet GitHub appelé OpenVPN UI, créé par d3vilh, qui intègre une interface web à OpenVPN et répond parfaitement à mes besoins en matière d'administration.



# Objectifs et cibles

Je souhaite :

- Permettre un accès distant sécurisé à mon infrastructure, garantissant la confidentialité et l'intégrité des données transmises.
- Administrer mon serveur VPN depuis une interface web, facilitant la gestion et la surveillance des connexions VPN.

Je pourrais accéder à mon réseau interne depuis internet de manière sécurisée grâce au VPN, ainsi que gérer les connexions VPN depuis une interface WEB

Pour répondre à ces objectifs :

- Posséder un accès distant sécurisé à mon infrastructure
  - Utilisation de OpenVPN
  - Faciliter la gestion des connexions VPN
  - Utilisation du projet OpenVPN UI
-

# Préparation la VM du serveur VPN

La première étape :

Définir le nom de la machine virtuelle,  
son emplacement

son image ISO : debian-12.5

Type : Linux

Version Debian 64 bits

La deuxième étape :

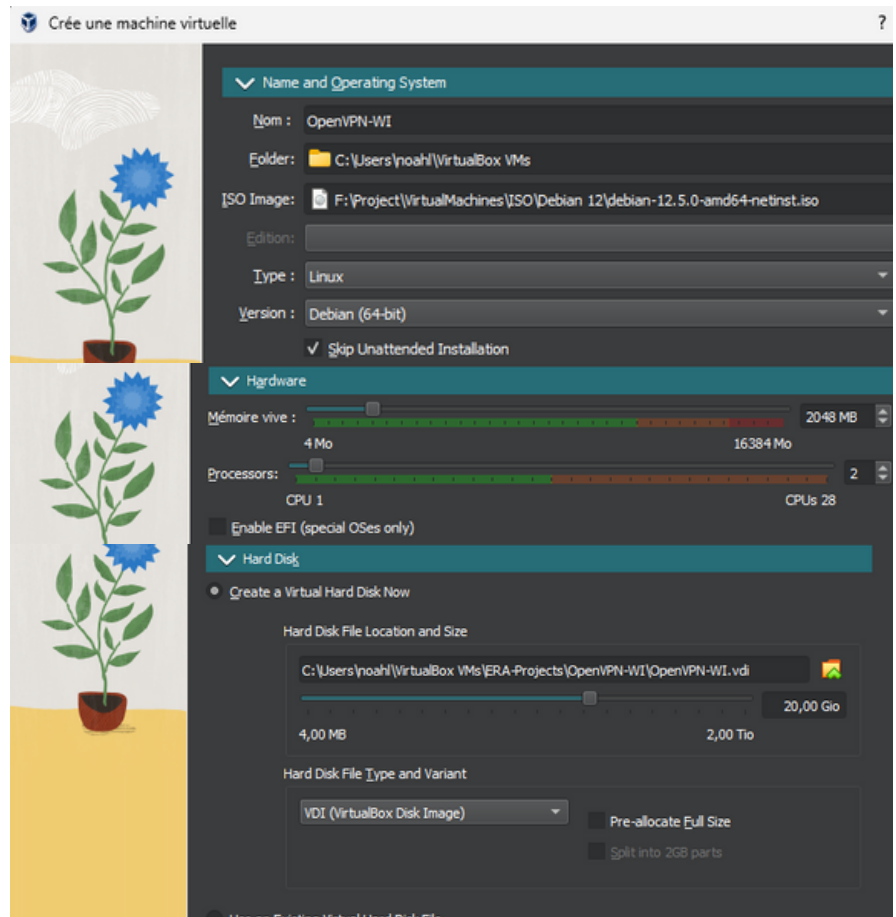
Indiquer la quantité de RAM qu'elle  
va utiliser de mon hôte

Le nombre de CPU qu'elle va  
également utiliser

La troisième étape :

Création du disque Virtuel au  
format VDI

La taille de ce disque sera la taille  
initié pour le système de la VM et  
non pour le stockage du cloud



**Nous pouvons confirmer la  
création de la VM**

# Installation de OpenVPN UI sur Debian 12

Prérequis :

- Git
- Docker

## Installation de Git



```
# apt install git
```

## Installation de Docker



```
# apt install docker-compose
```

# Installation du dépôt

```
# git clone https://github.com/d3vilh/openvpn-server
# mkdir /etc/openvpn
# mv openvpn-server /etc/openvpn
# cd /etc/openvpn
```

## Configuration du docker file

Nous allons devoir configurer le fichier docker-compose.yml afin d'adapter la configuration du serveur VPN à notre usage.

```
# nano /docker-compose.yml
```

```
version: "3.5"

services:
  openvpn:
    container_name: openvpn
    # If you want to build your own image with docker-compose, uncomment the next line, comment the
    # build: .
    image: d3vilh/openvpn-server:latest
    privileged: true
    ports:
      - "1194:1194/udp" # openvpn UDP port
      # - "1194:1194/tcp" # openvpn TCP port
      # - "2080:2080/tcp" # management port. uncomment if you would like to share it with the host
    environment:
      TRUST_SUB: "10.0.70.0/24"
      GUEST_SUB: "10.0.71.0/24"
      HOME_SUB: "192.168.88.0/24"
    volumes:
      - ./pki:/etc/openvpn/pki
      - ./clients:/etc/openvpn/clients
      - ./config:/etc/openvpn/config
      - ./staticclients:/etc/openvpn/staticclients
      - ./log:/var/log/openvpn
      - ./fw-rules.sh:/opt/app/fw-rules.sh
      - ./checkpsw.sh:/opt/app/checkpsw.sh
      - ./server.conf:/etc/openvpn/server.conf
    cap_add:
      - NET_ADMIN
    restart: always
    depends_on:
      - "openvpn-ui"

  openvpn-ui:
    container_name: openvpn-ui
    image: d3vilh/openvpn-ui:latest
    environment:
      - OPENVPN_ADMIN_USERNAME=admin
      - OPENVPN_ADMIN_PASSWORD=gagaZush
    privileged: true
    ports:
      - "8080:8080/tcp"
    volumes:
      - ./:/etc/openvpn
      - ./db:/opt/openvpn-ui/db
      - ./pki:/usr/share/easy-rsa/pki
      - /var/run/docker.sock:/var/run/docker.sock:ro
    restart: always
root@debianbase:/etc/openvpn#
```

## Description des variables du fichier

- `TRUST_SUB` est le sous-réseau de confiance, à partir duquel le serveur OpenVPN attribuera des adresses IP aux clients de confiance (sous-réseau par défaut pour tous les clients)
- `GUEST_SUB` est le sous-réseau Guest pour les clients ayant accès à Internet uniquement
- `HOME_SUB` est le sous-réseau où se trouve le serveur VPN, à travers lequel vous obtenez un accès Internet aux clients avec MASQUERADE
- `fw-rules.sh` est un fichier bash avec des règles de pare-feu supplémentaires que vous souhaitez appliquer lors du démarrage du conteneur
- `checkpsw.sh` est un script bash factice à utiliser avec `auth-user-pass-verify` une option dans `server.conf` le fichier. Il est utilisé pour vérifier les informations d'identification de l'utilisateur par rapport à certaines bases de données de mots de passe externes, comme LDAP ou Oath, ou MySQL. Si vous n'avez pas besoin de cette option, laissez-la telle quelle.

Je vais modifier les variables suivantes :

Permet de modifier le protocole réseau, ainsi que le port utilisé

```
ports:
- "62988:62988/udp" # openvpn UDP port
```

Permet de modifier le sous-réseau où se trouve le serveur VPN, à travers lequel vous obtenez un accès Internet aux clients avec MASQUERADE

```
environment:
  TRUST_SUB: "10.0.70.0/24"
  GUEST_SUB: "10.0.71.0/24"
  HOME_SUB: "192.168.1.0/24"
```

Je modifie les variables `OPENVPN_ADMIN` et `OPENVPN_PASSWORD` pour paramétrer le compte utilisateur de l'interface WEB

Je peux également modifier les ports de l'interface WEB

```
openvpn-ui:
  container_name: openvpn-ui
  image: d3vilh/openvpn-ui:latest
  environment:
    - OPENVPN_ADMIN_USERNAME=VADMIN
    - OPENVPN_ADMIN_PASSWORD=Ae2d4s@23
  privileged: true
  ports:
    - "80:80/tcp"
  volumes:
    - ./etc/openvpn
    - ./db:/opt/openvpn-ui/db
    - ./pki:/usr/share/easy-rsa/pki
    - /var/run/docker.sock:/var/run/docker.sock:ro
  restart: always
```

# Construction de l'image

Pour construire l'image du docker file que nous avons configuré nous utiliserons la commande :

```
# docker-compose up -d
```

```
root@debianbase:/etc/openvpn# docker-compose up -d
Creating network "openvpn_default" with the default driver
Pulling openvpn-ui (d3vilh/openvpn-ui:latest)...
latest: Pulling from d3vilh/openvpn-ui
4abcf2066143: Pull complete
54e611e34188: Pull complete
02882d0d3570: Pull complete
94420d7e0576: Pull complete
e55223947b2a: Pull complete
dc6ffdb07abd: Pull complete
07d0a6a6d136: Pull complete
dc512a0fc04c: Pull complete
e29ecdc190b4: Pull complete
82bd3569f6d7: Pull complete
29680421a2d3: Pull complete
dbb90151f380: Pull complete
4e3fb39c32ce: Pull complete
Digest: sha256:f803c8398a44780f54fe637a70b207c78ec99e1d54c02e493af121990ca52503
Status: Downloaded newer image for d3vilh/openvpn-ui:latest
Pulling openvpn (d3vilh/openvpn-server:latest)...
latest: Pulling from d3vilh/openvpn-server
4abcf2066143: Already exists
3b6476a8752b: Pull complete
f7c93359e974: Pull complete
5520fcd98056: Pull complete
7af7ae9c6aff: Pull complete
9a286ec80423: Pull complete
11bb192cc386: Pull complete
d92d52ada1f4: Pull complete
19066a2de6fa: Pull complete
Digest: sha256:ed1d50bc9f9b78a18cc0d3faf2f93d6dc29c4e66c0f590f27a0df2a1143141a1
Status: Downloaded newer image for d3vilh/openvpn-server:latest
Creating openvpn-ui ... done
Creating openvpn    ... done
root@debianbase:/etc/openvpn# |
```



# Aperçu finale de l'installation

OpenVPN UI

Home

Configuration

Certificates

Logs

Administrator

Light

Status

Clients: 0

In: 0 MB

Out: 0 MB

More info

Load Average

CPU count: 2

1m: 0 5m: 0.02 15m: 0

More info

OS uptime

1:12

More info

Server Time

2024-05-26 12:38:26

More info


Memory usage

Memory 33 / 1 967 MB - 1%

Swap 0 / 975 MB - 0%


Connected clients

Common Name	Real Address	Virtual Address	KB Received	KB Sent	Connected Since	Username
-------------	--------------	-----------------	-------------	---------	-----------------	----------



OpenVPN version:

OpenVPN 2.6.8 x86\_64 alpine Linux musl [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [MCRYPTMGR] [AEAD]



Operating system:

Linux

Architecture:

amd64

OpenVPN UI. Under the MIT License.

Version 0.9.5.3