

# Self-hosted

Servidores de serviços essenciais que você pode hospedar você mesmo

- Como ter sua própria VPN com OpenVPN e Docker

# Como ter sua própria VPN com OpenVPN e Docker

Nesse artigo mostraremos como você pode hospedar sua própria VPN usando os software OpenVPN e Docker.

## Pré-requisitos

- Acesso à shell de um VPS com um endereço IP público
- Acesso à algum usuário do grupo `docker` na VPS
- Familiaridade com a linha de comando

## Docker OpenVPN

Docker OpenVPN é um projeto que visa automatizar a configuração e *set up* do software OpenVPN usando Docker.

Como diz a própria página do GitHub

“ Out of the box stateless openvpn server docker image which starts in just a few seconds and doesn't require persistent storage

Estaremos utilizando esse software nesse tutorial.

## Instalando OpenVPN

Primeiro, faça `ssh` na sua VPS.

## Arquivo `.env`

Então, crie um diretório onde iremos armazenar algumas configurações que serão usadas posteriormente.

```
mkdir docker-openvpn && cd docker-openvpn  
touch .env
```

O arquivo `.env` terá algumas configurações básica sobre a VPN.

```
CONTAINER_NAME=openvpn  
PUBLIC_IP=127. 0. 0. 1
```

Substitua `127. 0. 0. 1` pelo seu endereço IP público. Se você não sabe qual é o endereço, execute o seguinte comando na shell da sua VPS:

```
curl -s https://api.ipify.org
```

Que irá imprimir o endereço IP da sua VPS. Então coloque esse endereço no arquivo `.env`.

## Usando docker

Há duas formas de instalar `Docker OpenVPN`. Usando `docker`, execute os seguintes comandos:

```
source .env  
docker run --detach \  
  --cap-add=NET_ADMIN \  
  -p publish 1194:1194/udp \  
  -e env HOST_ADDR=${PUBLIC_ip} \  
  -e name ${CONTAINER_NAME} \  
  alekslitvinenk/openvpn
```

O comando `source .env` irá colocar as variáveis especificadas em `.env` como variáveis de ambiente.

A flag `--detach` é para o container ser executado como um *background process*. A flag `--cap-add` adiciona *capacidades do Linux* ao *container*, nesse caso, ser administrador da rede. A flag `--env` especifica variáveis de ambiente do *container*. A flag `--name` especifica o nome do *container*. A flag `--publish` irá mapear portas do *host* para o *container* usando o protocolo especificado.

Por fim, temos o nome da imagem usada: `alekslitvinenk/openvpn`.

## Usando docker-compose

Eis um `docker-compose.yml` para facilitar o trabalho:

```
version: "2.0"

services:
  openvpn:
    container_name: ${CONTAINER_NAME}
    image: alekslitvinenk/openvpn
    restart: unless-stopped
    cap_add:
      - NET_ADMIN
    environment:
      - HOST_ADDR=${PUBLIC_IP}
    ports:
      - 1194:1194/udp
```

Coloque esse arquivo no mesmo diretório que o arquivo `.env`, ou seja, em `docker-openvpn/docker-compose.yml`.

Então, nesse mesmo diretório, rode o comando:

```
docker-compose up --detach
```

Novamente, a flag `detach` é para rodar a VPN no background.

# Configurando o cliente

## Obtendo o arquivo de configuração

Após ter configurado o servidor, precisamos obter o arquivo de configuração do OpenVPN. Para isso, usaremos `curl`.

Por padrão, essa imagem Docker não vem com `curl`. Temos que instala-lo como segue:

```
docker exec ${CONTAINER_NAME} apk add curl
```

Onde `${CONTAINER_NAME}` é o nome do *container* especificado em `.env`.

Após isso, execute:

```
docker exec ${CONTAINER_NAME} curl -s localhost:8080 > client.ovpn
```

Isso irá executar o comando `curl` dentro do *container*, que irá fazer um *request* para `localhost:8080`, o qual devolverá o arquivo de configuração do nosso OpenVPN e redirecionar a saída para o arquivo `client.ovpn`.

**Atenção:** Após o request ser feito, o servidor `http` que está ativo em `localhost:8080` será automaticamente desativado, o que significa que não será possível mais obter a configuração do OpenVPN. Entretanto, você pode gerar uma nova configuração usando o comando

```
docker exec ${CONTAINER_NAME} ./genclient.sh .
```

Depois disso você pode transferir o arquivo `client.ovpn` para sua máquina local usando `scp`, `rsync`, ou copiando e colando o conteúdo do arquivo caso seu terminal suporte operações de copiar-e-colar.

Usando `scp`:

```
scp username@host_address:/path/to/client.ovpn .
```

O comando acima irá usar o protocolo `ssh` para transferir o arquivo `client.ovpn` para o diretório

atual na sua máquina local.

Lembre-se de substituir `username`, `host_address`, e `/path/to/client.ovpn` pelo seu usuário de nome na VPS, o endereço IP da VPS, e o caminho absoluto do arquivo salvo anteriormente.

## Ativando a VPN com o Network Manager

Há inúmeros software clientes do `OpenVPN`. Essa parte do tutorial pressupõe que você está usando `network-manager`, que é um software gerenciador de rede disponível em várias distribuições Linux e BSD. Entretanto, mesmo que você não use `networkmanager`, é provável que tenha algum cliente que funciona para o seu gerenciador de redes.

Primeiro, instale os pacotes `openvpn`, `networkmanager`, e `networkmanager-openvpn`. É possível que os nomes dos pacotes sejam diferentes dependendo do seu Sistema Operacional.

Se você está usando alguma `Arch-based distro`, instale como segue:

```
pacman -S openvpn networkmanager networkmanager-openvpn
```

Após isso, e após habilitar o `networkmanager`, execute:

```
nmcli connection import type openvpn file /path/to/client.ovpn/
```

Onde `/path/to/client.ovpn/` é o caminho do arquivo `client.ovpn` na sua máquina local.

Esse comando irá adicionar uma conexão VPN na sua máquina. Para se conectar à VPN, execute o comando:

```
nmcli connection up client
```

Pronto :D, agora você tem sua própria VPN!