

[Get unlimited access](#)[Open in app](#)

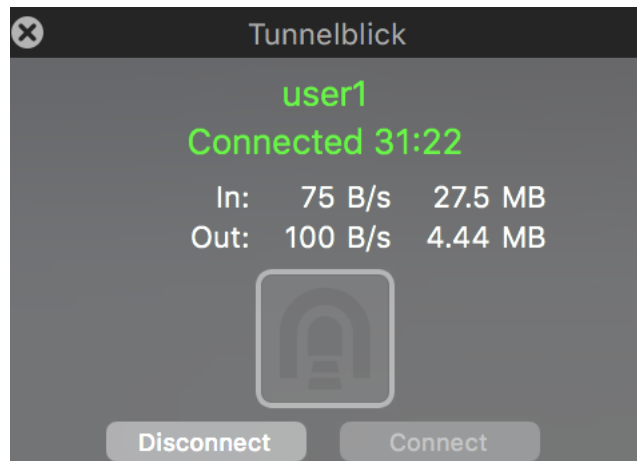
Guray Yildirim

[Following](#)Feb 13, 2018 · 4 min read · [Listen](#)

Save



## Set Up a VPN Server With Docker In 5 Minutes



**Update 2020-08:** Easy-RSA looks for a vars file and gives error otherwise. Creation of an empty file is added in the steps.

Many restricted environments make people need to use VPN servers. There are some VPN providers available for free or paid use but there are also many people who don't trust these providers. In this post, we will examine a method for creating your own OpenVPN server with Docker. We will also try to simplify it so that it can be completed in or less than 5 minutes.

Our OpenVPN server will also be capable of handling multiple user accounts and different port options thanks to Docker's easy port exporting options. We will start with UDP 3000 port which is different than its default port (UDP 1194). We will not use pre-built image and make our own image from a Dockerfile and other scripts written by [kylemanna](#) and licensed under [MIT](#). We will clone [this repository](#) and build our image.

First start with cloning the git repository into our server:

```
git clone https://github.com/kylemanna/docker-openvpn.git
```

Change current directory to our cloned repository:

```
cd docker-openvpn/
```

Build a new Docker image from these files. We will name it **myownvpn** in this example:

```
docker build -t myownvpn .
```

Afterwards, we need a volume or directory to store our config files and keys. This is a very significant step due to our keys will be put in that directory. As a result, I suggest you to keep that directory safe.

```
cd ..  
mkdir von-data && touch von-data/vars
```



[Get unlimited access](#)[Open in app](#)

```
$ docker run -v $PWD/vpn-data:/etc/openvpn --rm myownvpn ovpn_genconfig -u udp://IP_ADDRESS:3000

Processing PUSH Config: 'block-outside-dns'
Processing Route Config: '192.168.254.0/24'
Processing PUSH Config: 'dhcp-option DNS 8.8.8.8'
Processing PUSH Config: 'dhcp-option DNS 8.8.4.4'
Successfully generated config
Cleaning up before Exit ...
```

Be sure that your IP address is written in the command above. You can change your port on that stage. In this example, we used UDP 3000. You should get an output that looks like in that example.

We should init our PKI. This covers generating our CA certificate and we will have a private key belong to the PKI. We will be asked a password for protecting the private key. The command and truncated output will look like this:

```
$ docker run -v $PWD/vpn-data:/etc/openvpn --rm -it myownvpn ovpn_initpki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/pki

Generating a 2048 bit RSA private key
.....+++
....+++
writing new private key to '/etc/openvpn/pki/private/ca.key.XXXXCFGIEm'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
...
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:g

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/etc/openvpn/pki/ca.crt
👏 1.3K | 💬 22 | ...
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
... # Wait a while

Enter pass phrase for /etc/openvpn/pki/private/ca.key:
Check that the request matches the signature
...
Enter pass phrase for /etc/openvpn/pki/private/ca.key:

An updated CRL has been created.
CRL file: /etc/openvpn/pki/crl.pem
```

Finally, we can run the VPN server based on that config:

```
$ docker run -v $PWD/vpn-data:/etc/openvpn -d -p 3000:1194/udp --cap-add=NET_ADMIN myownvpn
```

We did not create any users yet. In order to connect to this OpenVPN server, we can create a user per connection with this command:

```
$ docker run -v $PWD/vpn-data:/etc/openvpn --rm -it myownvpn easyrsa build-client-full user1 nopass

Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/etc/openvpn/pki/private/user1.key.XXXXeoGIJE'
-----
Using configuration from /usr/share/easy-rsa/openssl-1.0.cnf
Enter pass phrase for /etc/openvpn/pki/private/ca.key:
...
```



[Get unlimited access](#)[Open in app](#)

to our VPN server without need of anything else. You will also be asked about certificate authority password during the user creation.

At the last step, we will generate a configuration file which will be sent to the user. In order to generate that file, we can run that command:

```
docker run -v $PWD/vpn-data:/etc/openvpn --rm myownvpn openvpn_getclient user1 > user1.ovpn
```

You can copy user1.ovpn file with SCP, SFTP, or any method you want. Users can connect to your server with that file. I am using Tunnelblick in MacOS. This is the result I got:

