

VPN Setup – The Easy Way



By:
Deepayan Patra
Nicklaus Choo
Zhiyuan Xu
June 23rd 2020

Overview

Introduction

This PDF booklet will go through the steps to set up a Virtual Private Network (VPN) from scratch. A VPN allows you to surf the web safely and securely from your personal devices when connected to an unsecured network such as the WiFi of a hotel or coffee shop.

How a VPN Works

Consider, for example, that you are in Singapore and are trying to access a website in the United States. Your request for the website to render on your browser goes through multiple intermediate devices before finally reaching the website's server. Through these intermediate devices, additional information such as your original location can be attached to your request. However, with a VPN, intermediate devices will be unable to attach information to your device's request as you will have a **direct, private connection** to the VPN server, which relays the content back to your device. This allows you to bypass surveillance from internet service providers and other third parties on the web.

Booklet Structure

Throughout this booklet we will be typing in commands into three separate machines - your local machine, the VPN server, and the Certificate Authority (CA). The VPN server hides your personal devices IP as you access the internet. The CA acts as a gatekeeper to your VPN server so only authorized users can access it (to prevent malicious attacks on your VPN server). Furthermore, extra information will be in gray parentheses: (extra information)

Commands typed into your local machine (the machine in front of you) will have a gray background:

```
$ Local machine commands look like this
```

Commands typed into the machine running our VPN server will have a yellow background:

```
$ VPN server commands look like this
```

Commands typed into the machine running our CA will have a blue background:

```
$ CA commands look like this
```

Output file text and text files will have a light green background:

```
$ Output and file text look like this
```

Important: When typing in the commands, ignore the front \$ sign. If you see \$ **./hello**, only type **./hello** into the terminal and press "Enter" to execute the command

Prerequisites

You need the 2 resources below to set up a VPN:

- 1 machine running Ubuntu 18.04 with sudo access as the VPN server
- 1 machine running Ubuntu 18.04 with sudo access as the CA

Note: if you do not have machines set up, but would like to host them on a commercial service, be sure to complete **Step 0** in the tutorial.

Table of Contents

Step 0. Deploy Digital Ocean Droplets	4
Step 1. Install Packages and Set Up the Servers	9
Step 2. Configure the Certificate Authority	10
Step 3. Set Up and Authenticate the VPN Server	11
Step 4. Setting Up OpenVPN	14
Step 5. Start the OpenVPN Server	17
Step 6. Set Up a Client Configuration Script	18
Step 7. Authenticate a Client Key and Configuration	20
Step 8. Connect the Client	22
Conclusion: Use Your VPN	23
References Used	24

Step 0. Deploy Digital Ocean Droplets


After following these steps, you should be able to SSH into your servers.

(SSH is a way of connecting directly to a machine/computer via a terminal/command prompt. You can then run commands on the remote machine from your local machine. Digital Ocean is a cloud hosting services where you can have a remote machine called a “droplet” for as low as \$5 a month)


1. Sign up for your Digital Ocean account via [this link](#)
2. Click on 
3. Fill up the form and click on “Create Project”

1 Create Project 2 Move Resources

Create new project




Name your project

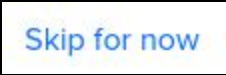
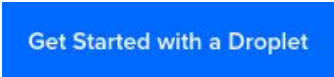


Add a description
Helpful for teams or differentiating between projects with similar names.











Tell us what it's for
This will help us to provide a more relevant experience.



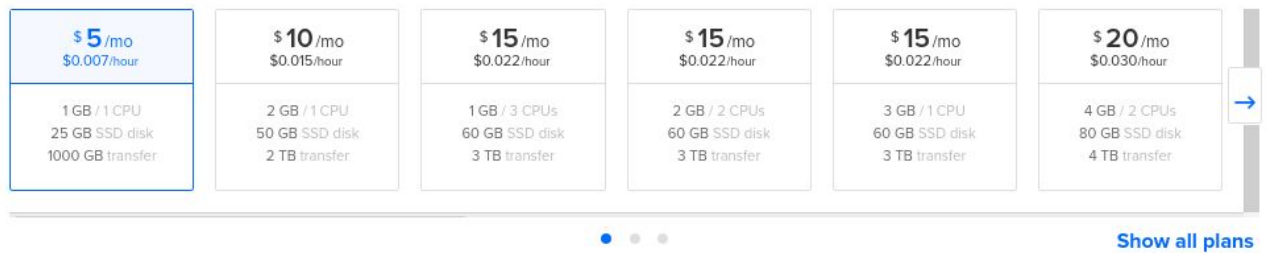
Create Project

4. When prompted to move resources, click 
5. Click on 
6. Choose Ubuntu as your droplet

Distributions Container distributions Marketplace Custom images

 Ubuntu 18.04.3 (LTS) x64 	 FreeBSD Select version 	 Fedora Select version 	 Debian Select version 	 CentOS Select version 
--	--	---	---	---

7. Choose the Standard plan
8. Make sure it is the cheapest plan



9. **Do not** add block storage and choose a datacenter region
10. **Do not** add a VPC Network

11. Check ☒ IPv6

12. Select SSH Keys via ☒ SSH keys
A more secure authentication method

13. Click [New SSH Key](#)

14. Go back to your terminal and type:

```
$ ssh-keygen
```

15. Press “Enter” when prompted for additional arguments mid-command

16. Check that you get the following output (with a different random art)

Output

```
Your identification has been saved in C:\Users\nickl/.ssh/id_rsa.
Your public key has been saved in C:\Users\nickl/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:0C0zh4fxQWwqoH+04Rz6k/eknxgsoTYEeCQyjUk7s9Q
nickl@DESKTOP-M7CAH89
The key's randomart image is:
+---[RSA 2048]---+
|==.      .oo    |
|==++ .   . *o.   |
|o=.E  .. Bo=    |
|.o=     ...*     |
| ....  .S       |
| .  .+o.        |
|  ++.Bo .       |
```

```
| ...*.0= . |  
| ..000+ |  
+-----[SHA256]-----+
```

17. Type the following into the terminal:

```
$ cat ~/.ssh/id_rsa.pub
```

18. Copy and paste the output into “SSH key content” on Digital Ocean and name your key “vpn-public-key” or whichever name you choose:

Add public SSH key

Copy your public SSH key and paste it in the space below. For instructions on how, follow the steps on the right.

SSH key content

tKRg/vs1TCn2VNus3UFggDZ6tr21QQ9WG7oFh5ErgcmwjrSMEwBVpT7mvJ7h728rH2ZiPX5hxdYEwYqgS5KJ7BS1vAOaz0g9ACqjssWuyqSnIC+klwOp6qAUtZoBaXtaXyE8tAAmK3KeCR59iPVdm/rRvVAKqjOgTb0KmWXUxzIJ5PF72ZnYuogUsoRltKRg/vs1TCn2VNus3UFggDZ6tr21QQ9WG7oFh5ErgcxwMTYh0sEqAHAKM19ra0wRWUKMi+Qyel4sjXjo4aQx5KSHldnyZGFKKsQlZr9lc4d0oeGR3xdmu8+g1FHRGGdq2i3kq8fK3KeCR59iPVdm/rRvVAKqjOxwMTYh0sEqAHAKM19ra0wRWUKMi+Qyel4sjXjo4aQx5KSHlz1Yj/ nickl@DESKTOP-M7CAH89

Name

vpn-public-key

Add SSH Key

19. Create two droplets. This example names the droplets **vpn-server**, **ca-server** for the VPN server and Certificate Authority respectively.

How many Droplets?

Deploy multiple Droplets with the same [configuration](#).

— 2 Droplets +

Choose a hostname

Give your Droplets an identifying name you will remember them by. Your Droplet name can only contain alphanumeric characters, dashes, and periods.

vpn-server

ca-server

20. Click

Create Droplet

21. On your terminal connect to your VPN server via SSH

```
$ ssh root@your_vpn_IP
```

22. Create a superuser **your_user** and set up a firewall

```
$ adduser your_user
$ usermod -aG sudo your_user
$ ufw allow OpenSSH
$ ufw enable
$ rsync --archive --chown=your_user:your_user ~/.ssh /home/your_user
```

23. Press “Ctrl + d” to logout

24. Login to your VPN server to test

```
$ ssh your_user@your_vpn_IP
```

25. Logout and repeat instructions 23 to 26 on your CA with **your_ca_IP**.

26. Generate a public key for your CA since you are logged on to it.

```
$ ssh-keygen
```

27. Logout and copy your CA's public key to your local machine and transfer it to your VPN server.

```
$ scp your_user@your_ca_IP:~/.ssh/id_rsa.pub ./id_rsa_ca.pub
$ scp ./id_rsa_ca.pub your_user@your_vpn_IP:~/.ssh
```

28. Generate a public key on your VPN server

```
$ ssh-keygen
```

29. Transfer that public key to your CA by typing in commands on your local machine.

```
$ scp your_user@your_vpn_IP:~/.ssh/id_rsa.pub ./id_rsa_vpn.pub
$ scp ./id_rsa_vpn.pub your_user@your_ca_IP:~/.ssh
```

30. Log on to your VPN server and append the public key to the list of authorized keys

```
$ cat ~/.ssh/id_rsa_ca.pub >> ~/.ssh/authorized_keys
$ chmod 600 ~/.ssh/authorized_keys
```

31. Log on to your CA and append the public key to the list of authorized keys

```
$ cat ~/.ssh/id_rsa_vpn.pub >> ~/.ssh/authorized_keys
```

```
$ chmod 600 ~/.ssh/authorized_keys
```

32. Move on to the next steps. You have successfully set up both servers and can now transfer files between them securely!

Step 1. Install Packages and Set Up the Servers

With the servers up and running, we will install and configure the needed packages.

1. Open a terminal and log in to your VPN Server.

(You can log in via SSH)

2. Execute the following commands to allow TCP connections through port 443.

```
$ sudo ufw allow OpenSSH
$ sudo ufw allow 443/tcp
```

3. Reload the firewall with the following commands. (You may need to re-login once firewall resets)

```
$ sudo ufw disable
$ sudo ufw enable
```

4. Download and Install OpenVPN and EasyRSA.

```
$ sudo apt update
$ sudo apt install openvpn

$ cd && wget
https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.5/EasyRSA-nix-3.0.5.tgz
$ tar -xvf EasyRSA-nix-3.0.5.tgz
```

5. Open another terminal and log in to the CA.

6. Download and Install EasyRSA.

```
$ cd && wget
https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.5/EasyRSA-nix-3.0.5.tgz
$ tar -xvf EasyRSA-nix-3.0.5.tgz
```

Step 2. Configure the Certificate Authority

Besides your VPN server, we also need to configure certificate authentication separately.

1. Switch to the terminal logged into your CA.
2. Create a file for your CA configuration.

```
$ cd EasyRSA-3.0.5
$ cp vars.example vars
```

3. Change the commented lines in `~/EasyRSA-3.0.5/vars` to the ones below in your favorite text editor. Remember to replace the bolded names with sensible values.

```
set_var EASYRSA_REQ_COUNTRY    "your_country"
set_var EASYRSA_REQ_PROVINCE   "your_state/province"
set_var EASYRSA_REQ_CITY       "your_city"
set_var EASYRSA_REQ_ORG        "Personal Certificate LTD"
set_var EASYRSA_REQ_EMAIL      "your_email@your_domain.com"
set_var EASYRSA_REQ_OU         "Personal"
```

4. Save and close the file.
5. Initialize the public key infrastructure.

```
$ ./easyrsa init-pki
```

6. Create a public certificate `ca.crt` and a private key `ca.key`. (Press “Enter” if you are prompted to enter extra information mid-command)

(The `nopass` option removes the need for a password prompt each time you sign your certificates. Pressing “Enter” removes the need for having an optional common name.)

```
$ ./easyrsa build-ca nopass
```

7. Check output.

Output

Common Name (eg: your user, host, or server name) [Easy-RSA CA]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
`/home/your_user/EasyRSA-3.0.5/pki/ca.crt`

Step 3. Set Up and Authenticate the VPN Server

Now we will use the keys and the certificates generated earlier to authenticate our VPN server.

1. Open the terminal window with your VPN server.
2. Navigate to the EasyRSA directory and run the `easyrsa` script on your VPN server.

```
$ cd ~/EasyRSA-3.0.5/  
$ ./easyrsa init-pki
```

3. Generate a Diffie-Hellman key (it may take a while).

```
$ ./easyrsa gen-dh
```

4. Check your output.

Output

```
DH parameters of size 2048 created at  
/home/your_user/EasyRSA-3.0.5/pki/dh.pem
```

5. Copy the generated file to the `/etc/openvpn` directory.

```
$ sudo cp ~/EasyRSA-3.0.5/pki/dh.pem /etc/openvpn/
```

6. Generate an HMAC signature and copy it to the `/etc/openvpn` directory.

```
$ openvpn --genkey --secret pki/ta.key  
$ sudo cp ~/EasyRSA-3.0.5/pki/ta.key /etc/openvpn/
```

7. Generate a new private key and certificate request file for the server (press “Enter” if prompted for additional information mid-command).

```
$ ./easyrsa gen-req server nopass
```

8. Check your output.

Output

```
-----  
Common Name (eg: your user, host, or server name) [server]:  
  
Keypair and certificate request completed. Your files are:  
req: /home/serveruser/EasyRSA-3.0.5/pki/reqs/server.req
```

```
key: /home/serveruser/EasyRSA-3.0.5/pki/private/server.key
```

9. Copy the private key to the /etc/openvpn directory.

```
$ sudo cp ~/EasyRSA-3.0.5/pki/private/server.key /etc/openvpn/
```

10. Transfer the certificate request file to your CA..

```
$ scp ~/EasyRSA-3.0.5/pki/reqs/server.req your_user@your_ca_ip:/tmp
```

11. Login to your CA and import the certificate request file.

```
$ cd ~/EasyRSA-3.0.5 && ./easyrsa import-req /tmp/server.req server
```

12. Sign the request file

```
$ ./easyrsa sign-req server server
```

13. Check your output. (Type “yes” and press “Enter” upon encountering a prompt)

Output

```
You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that
this request
has not been cryptographically verified. Please be sure it came
from a trusted
source or that you have verified the request checksum with the
sender.
```

```
Request subject, to be signed as a server certificate for 1080
days:
```

```
subject=
  commonName                = server
```

```
Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
...
```

14. Transfer the signed certificates back to your VPN server.

```
$ scp pki/issued/server.crt your_user@your_vpn_ip:/tmp
$ scp pki/ca.crt your_user@your_vpn_ip:/tmp
```

15. Switch back to your VPN server, and copy the `server.crt` and `ca.crt` files into the `/etc/openvpn/` directory:

```
$ sudo cp /tmp/{server,ca}.crt /etc/openvpn/
```

Step 4. Setting Up OpenVPN

With the generated certificate and key pair, we will proceed with the configurations.

1. Copy the sample configuration into the configuration directory and then extract it.

```
$ sudo cp
/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
/etc/openvpn/
$ sudo gzip -d /etc/openvpn/server.conf.gz
```

2. Open the configuration file in the editor of choice.

```
$ sudo vim /etc/openvpn/server.conf
```

3. Update the configuration file `server.conf` as follows.

- Update the line containing `port #` to be `port 443`
- Update the line containing `;proto tcp` to `proto tcp` by removing the `;` at the beginning of the line and change `proto udp` to `;proto udp`
- Update the line containing `dh filename` to be `dh dh.pem` to match the filename of the file you created in the previous section
- Update the line containing `push "redirect-gateway def1 bypass-dhcp"` to be uncommented by removing the `;` at the beginning of this line
- Update the lines containing `push "dhcp-option DNS 208.67.222.222"` and `push "dhcp-option DNS 208.67.220.220"` to be uncommented by removing the `;` at the beginning of both these lines
- Make sure the line containing `tls-auth ta.key 0` is not commented by removing the `;` if it exists
- Make sure the line containing `cipher AES-256-CBC` is not commented by removing the `;` if it exists
- Directly under this line add a line containing `auth SHA256`
- Update the lines containing `user nobody` and `group nogroup` to be uncommented by removing the `;` at the beginning of both these lines.
- Update the line containing `explicit-exit-notify 1` to be `explicit-exit-notify 0`

4. Open the system network configuration file.

```
$ sudo vim /etc/sysctl.conf
```

5. Update the system configuration file and remember to save and close the file
 - Update the line containing `net.ipv4.ip_forward=1` to be uncommented by removing the `#` at the beginning of this line.
6. Adjust the values for the current session.

```
$ sudo sysctl -p
```

7. Run the following command in the terminal and remember the output.

```
$ ip -o -4 route show to default | awk '{print $5}'
```

8. Open the firewall rules file.

```
$ sudo vim /etc/ufw/before.rules
```

9. Update `before.rules` by adding the following bolded lines as shown. Replace the `eth0` with the previously saved output from 7 (if it differs).

```
# ufw-before-input
# ufw-before-output
# ufw-before-forward
#

# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0 (change to the
interface you discovered!)
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT
# END OPENVPN RULES

# Don't delete these required lines, otherwise there will be
errors
```

10. Open the firewall file.

```
$ sudo vim /etc/default/ufw
```

11. Update the firewall configuration as follows.

- Update the line containing `DEFAULT_FORWARD_POLICY="DROP"` to be `DEFAULT_FORWARD_POLICY="ACCEPT"`

Step 5. Start the OpenVPN Server

With the server configured, we can start the OpenVPN service.

1. Start the VPN server.

```
$ sudo systemctl start openvpn@server
```

2. Check its status.

```
$ sudo systemctl status openvpn@server
```

3. Check the output. If the server is running correctly, you can expect a similar message to that below.

```
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; indirect;
   vendor preset: enabled)
   Active: active (running) since Sun 2020-06-21 01:13:47 UTC; 22h
   ago
     Docs: man:openvpn(8)

https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
https://community.openvpn.net/openvpn/wiki/HOWTO
Main PID: 2401 (openvpn)
  Status: "Initialization Sequence Completed"
    Tasks: 1 (limit: 1152)
   CGroup:
/system.slice/system-openvpn.slice/openvpn@server.service
└─2401 /usr/sbin/openvpn --daemon ovpn-server --status
/run/openvpn/server.status 10 --cd /etc/openvpn --script-security
2 --config /etc/openvpn/server.conf --writepid
/run/openvpn/server.pid
```

4. Activate OpenVPN on boot (the moment the machine starts up).

```
$ sudo systemctl enable openvpn@server
```

Step 6. Set Up a Client Configuration Script

Now that we are all set with the server, there are only a few steps left. First, we will set up the client configurations.

1. Create directories for storing keys and generated configuration files.

```
$ mkdir -p ~/client-configs/{keys,files}
$ chmod -R 700 ~/client-configs
$ cp ~/EasyRSA-3.0.5/pki/ta.key ~/client-configs/keys/
$ sudo cp /etc/openvpn/ca.crt ~/client-configs/keys/
$ sudo chown your_user ~/client-configs/keys/ca.crt
```

2. Copy the file example file. We will work on this file and edit it to our needs.

```
$ cp
/usr/share/doc/openvpn/examples/sample-config-files/client.conf
~/client-configs/
```

3. Update the ~/client-configs/client.conf with the following instructions.

- Change the protocol from UDP to TCP by adding `;` at the start of `proto udp` and remove the `;` from the line containing `proto tcp`
- Update the line containing `remote my-server-1 1194` to be of the form `remote YOUR_VPN_SERVER_IP 443` to update the port and use the public IP address of the VPN Server
- Comment out the lines containing `ca ca.crt`, `cert client.crt`, and `key client.key` by adding a `#` before these lines
- Comment out the line containing `tls-auth ta.key 1` by adding a `#` at the beginning of this line
- Make sure the line containing `cipher AES-256-CBC` is not commented by removing the `;` if it exists
- Add a line containing `auth SHA256` directly under this line.
- Add the line `key-direction 1` in the file as well

4. Create a new file `~/client-configs/generate_config.sh` in a text editor and write the following script.

```
#!/bin/bash

# First argument: Client identifier

KEY_DIR=~/client-configs/keys
OUTPUT_DIR=~/client-configs/files
BASE_CONFIG=~/client-configs/client.conf

cat ${BASE_CONFIG} \
  <(echo -e '<ca>' ) \
  ${KEY_DIR}/ca.crt \
  <(echo -e '</ca>\n<cert>' ) \
  ${KEY_DIR}/${1}.crt \
  <(echo -e '</cert>\n<key>' ) \
  ${KEY_DIR}/${1}.key \
  <(echo -e '</key>\n<tls-auth>' ) \
  ${KEY_DIR}/ta.key \
  <(echo -e '</tls-auth>' ) \
  > ${OUTPUT_DIR}/${1}.ovpn
```

5. Save the file and run the following command to grant execution permission.

```
$ chmod 700 ~/client-configs/generate_config.sh
```

Step 7. Authenticate a Client Key and Configuration

Now we will move on to generating the keys and certificates with EasyRSA.

1. Navigate to the EasyRSA directory on your VPN server and generate a new private key and a certificate request file for the client.

```
$ cd ~/EasyRSA-3.0.5/  
$ ./easyrsa gen-req client_name nopass
```

2. Copy the private key to the correct directory.

```
$ cp ~/EasyRSA-3.0.5/pki/private/client_name.key  
~/client-configs/keys/
```

3. Transfer your certificate request to your CA machine.

```
$ scp ~/EasyRSA-3.0.5/pki/reqs/client_name.req  
your_user@your_ca_ip:/tmp
```

4. Go back to your CA machine, switch to the EasyRSA directory and import the certificate request file.

```
$ cd ~/EasyRSA-3.0.5  
$ ./easyrsa import-req /tmp/client_name.req client_name
```

5. Sign the request.

```
$ ./easyrsa sign-req client client_name
```

6. Transfer the signed request back to your VPN server.

```
$ scp ~/EasyRSA-3.0.5/pki/issued/client_name.crt  
your_user@your_vpn_ip:/tmp
```

7. Switch over to your VPN server and complete the configuration.

```
$ cp /tmp/client_name.crt ~/client-configs/keys  
$ cd ~/client-configs  
$ ./generate_config.sh client_name
```

8. Transfer the config file to your local machine via scp in your local machine's terminal.

```
$ scp  
your_user@your_vpn_IP:~/client-configs/files/client_name.ovpn ./
```

Step 8. Connect the Client

With all the hard work done, you're only a few commands away from your own secure internet connection!

Linux - Ubuntu/Debian

1. Install OpenVPN.

```
$ sudo apt update  
$ sudo apt install openvpn
```

2. Run the configuration. Be sure to replace **client_name** with your actual client's file name.

```
$ sudo openvpn --config client_name.ovpn
```

macOS

1. Install the free open-source graphical user interface [Tunnelblick](#) for OpenVPN on macOS.
2. Double click on the .ovpn file and let the profile install.
3. Connect to the intended client through the app to start the VPN connection.

Windows

1. Download and install the latest build of OpenVPN application the [OpenVPN's Downloads page](#).
2. Copy the .ovpn file to the OpenVPN config folder as administrator (`\Users\<<Name>\OpenVPN\Config` or `\Program Files\OpenVPN\config`).
3. Launch the OpenVPN application.
4. Right click on the OpenVPN system tray icon and the name of the OpenVPN configuration file you copied will be listed on the menu. Click Connect.

Android & iOS

A VPN application developed by OpenVPN is available for both Android and iOS. Install the application and import the client .ovpn file.

- [Android OpenVPN Connect](#)
- [iOS OpenVPN Connect](#)

Conclusion: Use Your VPN

You're all set up! Go ahead and try out your VPN: check out the location listed [here](#) which should reflect the location of your VPN Server once it's running!

Before:

IP Location	Pittsburgh, Pennsylvania (US) [Details]
-------------	---

After:

IP Location	Columbus, Ohio (US) [Details]
-------------	---

References Used

[1]

<https://www.digitalocean.com/community/tutorials/how-to-set-up-an-openvpn-server-on-ubuntu-18-04>

[2] <https://linuxize.com/post/how-to-set-up-an-openvpn-server-on-ubuntu-18-04/>

[3] <https://community.openvpn.net/openvpn/wiki/GettingStartedwithOVPN>

[4] <https://community.openvpn.net/openvpn/wiki/HOWTO>