

VPN Setup – The Easy Way



By
Deepayan Patra,
Nicklaus Choo,
Zhiyuan Xu

June 23, 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.

Throughout this booklet we will be typing in commands into two separate machines - the VPN server and the Certificate Authority (CA). The VPN server hides your personal devices IP as you access the internet. If you visit `www.google.com` from your personal device, your request first goes through the VPN server and then the actual Google webpage. Information coming back to your device (so that the webpage is displayed on your device) also goes through the VPN server before reaching your device. 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).

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 and file text will have a green background:

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

Prerequisites

You need the 2 resources below in order to completely install VPN:

- 1 machine running Ubuntu 18.04 as the VPN server
- 1 machine running Ubuntu 18.04 as the CA

Both machines require you to have sudo access. If you do not have these machines set up, follow the instructions in “0. Setting Up Your Servers with Digital Ocean”, else skip to Section 1. There are 10 sections in total.

Table of Contents

Step 0. Setting Up Your Servers with Digital Ocean	3
Step 1. Build CA with EasyRSA	5
Step 2. Install OpenVPN and EasyRSA on VPN server	7
Step 3. Create Diffie-Hellman and HMAC Keys	8
Step 4. Create Server Certificate and Private Key	9
Step 5. Configure the OpenVPN Service	11
Step 6. Start and Enable the OpenVPN Service	14
Step 7. Create Client Configuration Infrastructure	15
Step 8. Create Client Certificate and Private Key Configuration	17
Step 9. Connect Clients	19


Step 0. Setting Up Your Servers with Digital Ocean

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


1. Create your Digital Ocean account
2. Click on [+ New Project](#)
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 [Skip for now](#)
5. Click on [Get Started with a Droplet](#)
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

Plan	Price	CPU	Disk	Transfer
Starter	\$5/mo (\$0.007/hour)	1 GB / 1 CPU	25 GB SSD disk	1000 GB transfer
Basic	\$10/mo (\$0.015/hour)	2 GB / 1 CPU	50 GB SSD disk	2 TB transfer
Professional	\$15/mo (\$0.022/hour)	1 GB / 3 CPUs	60 GB SSD disk	3 TB transfer
Professional	\$15/mo (\$0.022/hour)	2 GB / 2 CPUs	60 GB SSD disk	3 TB transfer
Professional	\$15/mo (\$0.022/hour)	3 GB / 1 CPU	60 GB SSD disk	3 TB transfer
Enterprise	\$20/mo (\$0.030/hour)	4 GB / 2 CPUs	80 GB SSD disk	4 TB transfer

Show all plans

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. Follow instructions in the pop-up window

15. 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

16. Click [Create Droplet](#)

Step 1. Build CA with EasyRSA

1. Log in to your CA

(You can log in via SSH)

2. Download and Install EasyRSA

```
$ cd && wget
https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.4/EasyRSA-nix-3.0.4.tgz
$ tar -xvf EasyRSA-3.0.4.tgz
$ cd ~/EasyRSA-3.0.4/
$ cp vars.example vars
```

3. Change commented lines in ~/EasyRSA-3.0.4/vars in your favorite text editor to the ones below

(remember to replace each field in double quotes with a sensible value)

```
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 public key infrastructure

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

6. Create public certificate `ca.crt` and 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

```
...
Enter PEM pass phrase:
```

Verifying - Enter PEM pass phrase:

...

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/causer/EasyRSA-3.0.4/pki/ca.crt

Step 2. Install OpenVPN and EasyRSA on VPN server

1. Open a terminal window with a shell to your VPN server
2. Install OpenVPN on your VPN server

(OpenVPN is a free open-source implementation of a VPN service)

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

3. Install EasyRSA on your VPN server

```
$ cd && wget -P ~/https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.4/EasyRSA-3.0.4.tgz  
$ tar -xvf EasyRSA-3.0.4.tgz
```

4. Navigate to EasyRSA directory and run `easyrsa` script on your VPN server

```
$ cd EasyRSA-3.0.4/  
$ ./easyrsa init-pki
```


Step 3. Create Diffie-Hellman and HMAC Keys

1. Navigate to the EasyRSA directory on your VPN server.

```
$ cd EasyRSA-3.0.4/
```

2. Generate a Diffie-Hellman key.

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

3. Check output (it might take some time to generate the cryptographic key)

Output

```
DH parameters of size 2048 created at  
/home/serveruser/EasyRSA-3.0.4/pki/dh.pem
```

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

```
$ sudo cp ~/EasyRSA-3.0.4/pki/dh.pem /etc/openvpn/
```

5. Generate an HMAC signature and move to the `/etc/openvpn` directory

```
$ openvpn --genkey --secret ta.key  
sudo cp ~/EasyRSA-3.0.4/ta.key /etc/openvpn/
```

Step 4. Create Server Certificate and Private Key

1. Generate new private key for server and a certificate request file

```
$ cd ~/EasyRSA-3.0.4/  
$ ./easyrsa gen-req server1 nopass
```

2. Check output

Output

Common Name (eg: your user, host, or server name) [server1]:

Keypair and certificate request completed. Your files are:

req: /home/serveruser/EasyRSA-3.0.4/pki/reqs/server1.req

key: /home/serveruser/EasyRSA-3.0.4/pki/private/server1.key

3. Copy the private key to the `/etc/openvpn` directory

```
$ sudo cp ~/EasyRSA-3.0.4/pki/private/server1.key /etc/openvpn/
```

4. Transfer the certificate request file to your CA

```
$ scp ~/EasyRSA-3.0.4/pki/reqs/server1.req  
your_ca_user@your_ca_ip:/tmp
```

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

```
$ cd ~/EasyRSA-3.0.4  
$ ./easyrsa import-req /tmp/server1.req server1
```

6. Sign the request file

```
$ cd ~/EasyRSA-3.0.4  
$ ./easyrsa sign-req server server1
```

7. Transfer signed certificate back to your VPN server

```
$ scp pki/issued/server1.crt your_server_user@your_server_ip:/tmp  
$ scp pki/ca.crt your_server_user@your_server_ip:/tmp
```

8. Login to your VPN server, and copy the `server1.crt` and `ca.crt` files into the `/etc/openvpn/` directory:

```
$ sudo cp /tmp/{server1,ca}.cert /etc/openvpn/
```

Step 5. Configure the OpenVPN Service

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 nano /etc/openvpn/server1.conf
```

3. Update the configuration file `server1.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 nano /etc/sysctl.conf
```

5. Update the system configuration file.

- Update the line containing `net.ipv4.ip_forward=1` to be uncommented by removing the `#` at the beginning of this line.

6. Run the following command in the terminal and save the output.

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

7. Open the firewall rules file.

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

8. Update `before.rules` by adding the following bolded lines as shown.
Replace the `eth0` with the previously saved output from 6.

```
# 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
```

9. Update `/etc/default/ufw` as follows.

```
$ sudo nano /etc/default/ufw
```

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

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

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

11. Finally, reload ufw with the following commands.

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

Step 6. Start and Enable the OpenVPN Service

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

1. Start the VPN server.

```
$ sudo systemctl start openvpn@server1
```

2. The server should be up and running. To check its status, run the following command.

```
$ sudo systemctl status openvpn@server1
```

If the server is running correctly, you can expect a similar message as follows.

```
● openvpn@server1.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
```

3. To set the OpenVPN server to be activated at boot, run the following command.

```
$ sudo systemctl enable openvpn@server1
```

Step 7. Create Client Configuration Infrastructure

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

1. We need to create a directory storing the client-related files and two sub-directories for keys and generated configuration files.

```
$ mkdir -p ~/client-configs/{keys, files}
$ chmod -R 700 ~/client-configs
$ cp ~/EasyRSA-3.0.4/ta.key ~/client-configs/keys/
$ sudo cp /etc/openvpn/ca.crt ~/client-configs/keys/
```

2. Start with the provided starter configuration by copying the file.

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

3. Update the `~/client-configs/client.conf` with the following instructions.
 - update the line containing `remote my-server-1 1194` to be of the form `remote YOUR_SERVER_IP 443` to update the port and use the public IP address of the VPN Server
 - Change the protocol from `proto udp` by adding a `;` at the start of this line and remove the `;` from the line containing `proto tcp`
 - Comment out the lines containing `ca ca.crt`, `cert client.crt`, `key client.key` by adding a `#` before these lines
 - 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`
 - 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 8. Create Client Certificate and Private Key Configuration

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-3.0.4/  
$ ./easyrsa gen-req client1 nopass
```

2. Copy the private key to the correct directory

```
cp ~/EasyRSA-3.0.4/pki/private/client1.key  
~/openvpn-clients/files/
```

3. Transfer certificate request to your CA machine

```
$ scp ~/EasyRSA-3.0.4/pki/reqs/client1.req  
your_ca_user@your_ca_ip:/tmp
```

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

```
$ cd ~/EasyRSA-3.0.4  
$ ./easyrsa import-req /tmp/client1.req client1
```

5. Sign the request

```
./easyrsa sign-req client client1
```

6. Transfer the signed request back to your VPN server

```
$ scp ~/EasyRSA-3.0.4/pki/issued/client1.crt  
your_server_user@your_server_ip:/tmp
```

7. Login to your VPN server and complete configuration

```
$ cp /tmp/client1.crt ~/openvpn-clients/files  
$ cd ~/openvpn-clients  
$ ./gen_config.sh client1
```

8. Transfer the config file to your local machine via `scp` (there are other ways to transfer the file but `scp` is used for linux machines)

```
$ scp ~/client-configs/files/client1.ovpn  
your_user@your_local_ip:~
```

Step 9. Connect Clients

Linux - Ubuntu

1. Install OpenVPN on Ubuntu and Debian

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

2. Run OpenVPN

```
$ sudo openvpn --config client1.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 (`\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 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 `.ovp` file.

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