

How to Create an OpenVPN Server on Ubuntu 20.04

Learn how to set up and configure your own OpenVPN server on Ubuntu 20.04 with our step-by-step guide for secure remote access to your network.

Contents

01	Introduction	3
02	Install OpenVPN and EasyRSA	3
03	Set up a Certificate Authority	3
04	Create the Server public/private keys	5
05	Create Client Public/Private Keys	6
06	Configure the OpenVPN Server	6
07	Start the OpenVPN Server	8

Introduction

OpenVPN is a full-featured, open-source secure socket layer (SSL) virtual private network (VPN) that offers a broad range and rapid development of features that allow you to access the Internet safely and securely through your private server. In this guide, we install OpenVPN on a Ubuntu 20.04 server.

Prerequisites

- A Ubuntu 20.04 Server
- Non-root user with sudo Privileges

Install OpenVPN and EasyRSA

First, update your server, and install OpenVPN from default Ubuntu sources.

```
$ sudo apt-get install openvpn
```

Since OpenVPN is an SSL VPN, it uses certificates to encrypt traffic between the server and connected clients. So, we need to install the easy-rsa hosted certificate authority to create and sign new certificates on the server.

```
$ sudo apt-get install easy-rsa
```

Set up a Certificate Authority

We need to utilize the easy-rsa template to create our OpenVPN server's Certificate Authority by copying it to a new directory.

```
$ make-cadir ~/openvpn-ca
```

Enter the created directory

```
$ cd openvpn-ca
```

Now, open the file named `vars` through nano or any other editor.

```
$ nano vars
```

Locate the following entries, uncomment them by removing the `#` sign, and enter your details.

```
#set_var EASYRSA_REQ_COUNTRY    "US"  
#set_var EASYRSA_REQ_PROVINCE   "California"  
#set_var EASYRSA_REQ_CITY       "San Francisco"  
#set_var EASYRSA_REQ_ORG        "Copyleft Certificate Co"  
#set_var EASYRSA_REQ_EMAIL      "me@example.net"  
#set_var EASYRSA_REQ_OU         "My Organizational Unit"
```

Particularly, you should enter your desired country, province, city, company, email, and department. After editing, your file should now look similar to the one below.

```
set_var EASYRSA_REQ_COUNTRY "US"  
set_var EASYRSA_REQ_PROVINCE "California"  
set_var EASYRSA_REQ_CITY "San Francisco"  
set_var EASYRSA_REQ_ORG "My Example Company"  
set_var EASYRSA_REQ_EMAIL "user@example.com"  
set_var EASYRSA_REQ_OU "Marketing"
```

Save and close the file.

Next, within the `openvpn-ca` directory resides a script `easyrsa` that lets you perform a series of tasks and commands for building the certificate authority. Execute the script with the argument `init-pki` to ready public key infrastructure on the server.

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

Note: using Easy-RSA configuration from: `./vars`

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

Build the CA to create two important files (`ca.crt` and `ca.key`) that make up an SSL certificate.

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

During the build process, you will be asked to enter a common name for your certificate authority, enter a simple name or click enter for a default name.

```
If you enter '.', the field will be left blank.  
-----  
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:
```

Create the Server public/private keys

Now that a CA has been created, you need to build a server certificate and key pair. To do so, run the command below with a custom name for your server. In this case, we use `vpnsrvr`. Replace it with a desired simpler name since it will be required for reference.

```
$ ./easymrsa gen-req vpnsrvr nopass
```

The command will create a new private server key and certificate request file. Now, create a strong Diffie-Hellman key that will be used during the key exchange process.

```
$ ./easymrsa gen-dh
```

This may take a few minutes to complete. Once ready, create an HMAC signature to strengthen the TLS certificate integrity verification capabilities:

```
$ openvpn --genkey --secret ta.key
```

Finally, copy the created `vpnsrv`, `dh`, and `hmac` keys to the OpenVPN directory.

```
$ sudo cp ~/openvpn-ca/pki/private/vpnsrv.key /etc/openvpn/  
$ sudo cp ~/openvpn-ca/ta.key /etc/openvpn/  
$ sudo cp ~/openvpn-ca/pki/dh.pem /etc/openvpn/  
$ sudo cp ~/openvpn-ca/pki/ca.crt /etc/openvpn/
```

Create Client Public/Private Keys

Navigate back to the CA directory and run the `easyrsa` script with `gen-req`, and a simple name for your client.

```
$ cd ~/openvpnca/  
./easyrsa gen-req client nopass
```

Press enter to confirm the common name. If you wish to create a user protected with a password, remove the `nopass` option.

Configure the OpenVPN Server

Now that we've created a certificate authority, we must configure the server. First, copy and extract the sample OpenVPN configuration file to the default directory.

```
$ gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz | sudo tee /etc/openvpn/server.conf
```

Open the configuration file located at `/etc/openvpn/server.conf` and make some changes to it.

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

Uncomment the following lines:

```
push "redirect-gateway def1 bypass-dhcp"  
user nobody  
group nogroup  
push "dhcp-option DNS 208.67.222.222"  
push "dhcp-option DNS 208.67.220.220"  
tls-auth ta.key 0
```

Change the user directive to listen for a non-privileged user instead of root.

```
user openvpn
```

Now, ensure that OpenVPN is pointing to the right `.cert` and `.key` files. Then, change the entries depending on the VPN server name prescribed earlier in this guide.

```
ca ca.crt  
cert vpnserver.crt  
key vpnserver.key # This file should be kept secret
```

Save and close the file.

Next, to allow connected clients to access the Internet through the OpenVPN server, we need to modify `/etc/sysctl.conf`

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

Uncomment the line:

```
net.ipv4.ip_forward=1
```

Save and close the file, then apply the changes.

```
$ sysctl -p
```

Start the OpenVPN Server

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

To provide internet access and properly direct traffic, we need to set up a Network Address Translation (NAT) rule with the following command.

```
$ sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/16 -o eth0 -j MASQUERADE
```

Congratulations, you have successfully installed OpenVPN on your Ubuntu 20.04 server.



VULTR

