

Ubuntu 20.04 LTS Set Up OpenVPN Server In 5 Minutes

Author: Vivek Gite • Last updated: July 19, 2023 • [103 comments](#)

I am a new Ubuntu Linux 20.04 LTS server system administrator. How can I set up an OpenVPN



Server on an Ubuntu Linux version 20.04 LTS server

to shield my browsing activity from bad guys on public Wi-Fi, encrypt all traffic while connecting to 4G LTE network, and more?

Introduction OpenVPN is extremely popular and a full-featured SSL VPN (Virtual Private Network) software. It implements OSI layer 2 or 3 secure network extension using the SSL/TLS protocol.



Like much other popular software, it is open-source, free software and distributed under the GNU GPL. A VPN allows you to connect securely to an insecure public network such as wifi network at the airport or hotel. In many enterprises and government offices, VPN is needed to access your corporate server resources. Another widespread usage to bypass the geo-blocked sites/apps and increase your privacy or safety online. This tutorial provides step-by-step instructions for configuring an OpenVPN server on Ubuntu Linux 20.04 LTS server.

| Tutorial requirements | |
|-----------------------|------------------------------|
| Requirements | Ubuntu Linux 20.04 LTS |
| Root privileges | Yes |
| Difficulty level | Intermediate |
| Category | OpenVPN |
| Est. reading time | 13 minutes |

Tutorial requirements

Table of contents ↓

[1 Update system](#)

[2 Find public IP](#)

[3 Install OpenVPN server](#)

[4 Configuring OpenVPN clients](#)

[5 Adding or removing VPN clients](#)

[6 Verification](#)

[7 Conclusion](#)

nixCraft: Privacy First, Reader Supported

→ **nixCraft is a one-person operation.** I create all the content myself, with no help from AI or ML. I keep the content accurate and up-to-date.

→ **Your privacy is my top priority.** I don't track you, show you ads, or spam you with emails. Just pure content in the true spirit of Linux and FLOSS.

→ **Fast and clean browsing experience.** nixCraft is designed to be fast and easy to use. You won't have to deal with pop-ups, ads, cookie banners, or other distractions.

→ **Support independent content creators.** nixCraft is a labor of love, and it's only possible thanks to the support of our readers. If you enjoy the content, please support us on Patreon or share this page on social media or your blog. Every bit helps.

[Join Patreon →](#)

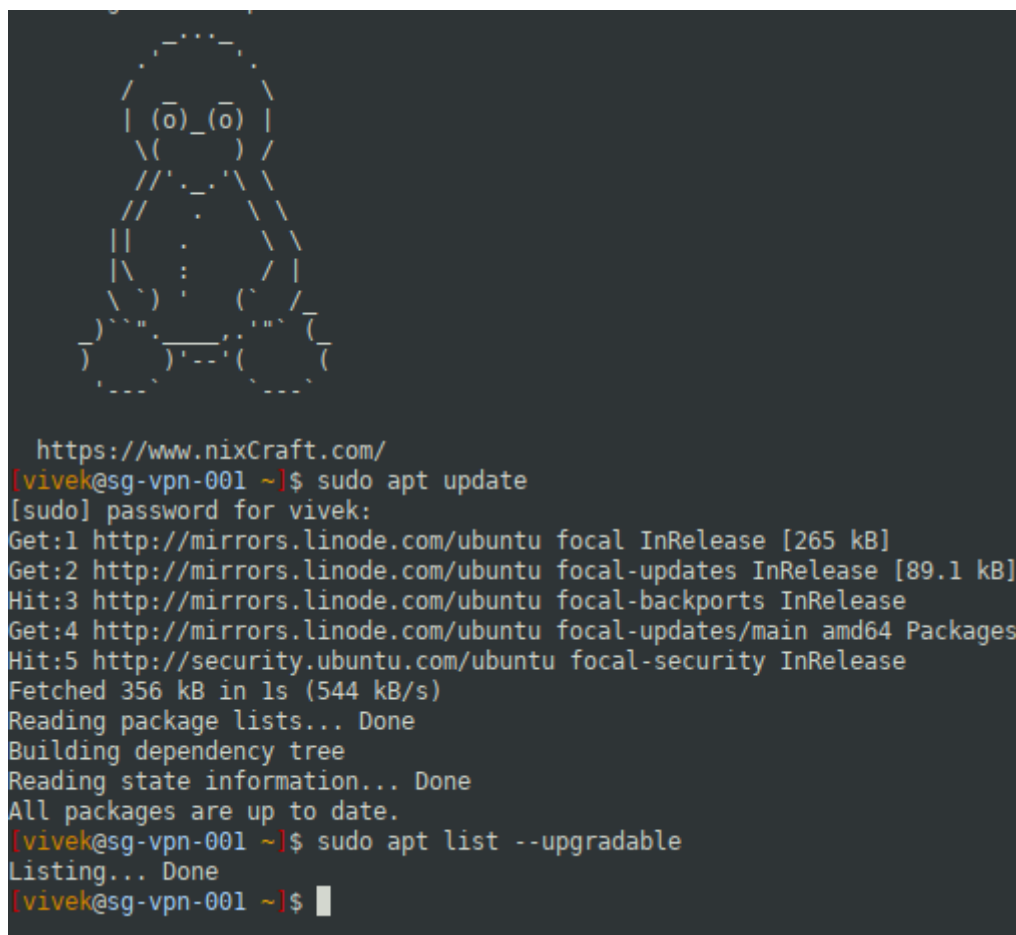
Procedure: Ubuntu 20.04 LTS Set Up OpenVPN Server In 5 Minutes

The steps are as follows:

Step 1 – Update your system

First, run the [apt command](#) to apply security updates:

```
sudo apt update
sudo apt upgrade
```

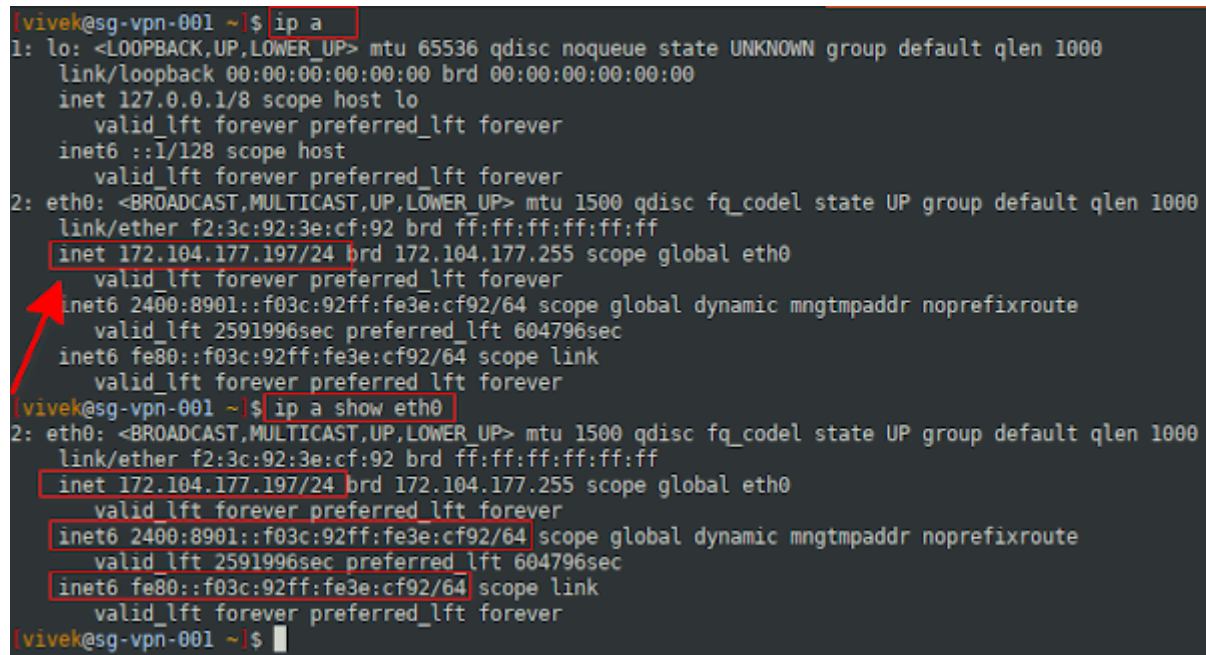
A terminal window screenshot showing the execution of two commands. The first command, 'sudo apt update', outputs information about package lists being updated from mirrors, including focal, focal-updates, and focal-security. The second command, 'sudo apt list --upgradable', outputs 'Listing... Done' and indicates that all packages are up to date. The terminal background is dark with light-colored text.

```
https://www.nixCraft.com/
[vivek@sg-vpn-001 ~]$ sudo apt update
[sudo] password for vivek:
Get:1 http://mirrors.linode.com/ubuntu focal InRelease [265 kB]
Get:2 http://mirrors.linode.com/ubuntu focal-updates InRelease [89.1 kB]
Hit:3 http://mirrors.linode.com/ubuntu focal-backports InRelease
Get:4 http://mirrors.linode.com/ubuntu focal-updates/main amd64 Packages
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Fetched 356 kB in 1s (544 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
[vivek@sg-vpn-001 ~]$ sudo apt list --upgradable
Listing... Done
[vivek@sg-vpn-001 ~]$
```

Step 2 – Find and note down your IP address

Use the [ip command](#) as follows:

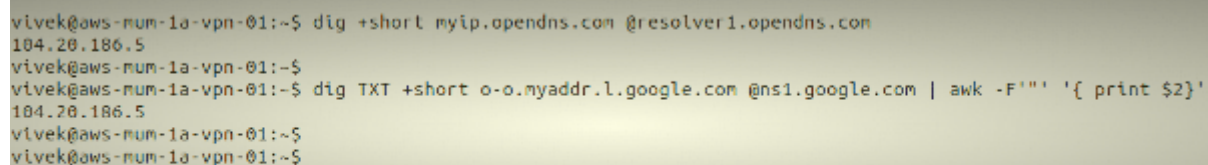
```
ip a
ip a show eth0
```



```
[vivek@sg-vpn-001 ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether f2:3c:92:3e:cf:92 brd ff:ff:ff:ff:ff:ff
    inet 172.104.177.197/24 brd 172.104.177.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 2400:8901::f03c:92ff:fe3e:cf92/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 2591996sec preferred_lft 604796sec
    inet6 fe80::f03c:92ff:fe3e:cf92/64 scope link
        valid_lft forever preferred_lft forever
[vivek@sg-vpn-001 ~]$ ip a show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether f2:3c:92:3e:cf:92 brd ff:ff:ff:ff:ff:ff
    inet 172.104.177.197/24 brd 172.104.177.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 2400:8901::f03c:92ff:fe3e:cf92/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 2591996sec preferred_lft 604796sec
    inet6 fe80::f03c:92ff:fe3e:cf92/64 scope link
        valid_lft forever preferred_lft forever
[vivek@sg-vpn-001 ~]$
```

Alternatively we can run the following [dig command](#)/[host command](#) to find out [our public IP address from Linux command line](#) itself:

```
dig +short myip.opendns.com @resolver1.opendns.com
## Get IPv4 ##
dig -4 +short myip.opendns.com @resolver1.opendns.com
## Find IPv6 ##
dig -6 +short myip.opendns.com @resolver1.opendns.com
## OR ##
dig TXT +short o-o.myaddr.l.google.com @ns1.google.com | awk -F'"' '{ print $2}'
```



```
vivek@aws-num-1a-vpn-01:~$ dig +short myip.opendns.com @resolver1.opendns.com
104.20.186.5
vivek@aws-num-1a-vpn-01:~$ dig -4 +short myip.opendns.com @resolver1.opendns.com
104.20.186.5
vivek@aws-num-1a-vpn-01:~$ dig -6 +short myip.opendns.com @resolver1.opendns.com
vivek@aws-num-1a-vpn-01:~$ dig TXT +short o-o.myaddr.l.google.com @ns1.google.com | awk -F'"' '{ print $2}'
104.20.186.5
vivek@aws-num-1a-vpn-01:~$
```

A note about IP address assigned to your server

Most cloud and bare-metal servers have two types of IP address provided by the ISP:

1. **Public static IP address directly** assigned to your box and routed from the Internet. For example, Linode, Digital Ocean, and others give you direct public IP address.
2. Private static IP address directly attached to your server and **your server is behind NAT with public IP** address. For example, AWS EC2/Lightsail give you this kind of NAT public IP address.

The script will automatically detect your networking setup. All you have to do is provide a correct IP address when asked for it.

Step 3 – Download and run openvpn-install.sh script

I am going to use the [wget command](#) as follows:

```
wget https://git.io/vpn -O openvpn-ubuntu-install.sh
```

```
[vivek@sg-vpn-001 ~]$ wget https://git.io/vpn -O openvpn-ubuntu-install.sh
--2020-04-24 18:35:00-- https://git.io/vpn
Resolving git.io (git.io)... 3.220.9.69, 34.233.35.85, 34.203.99.93, ...
Connecting to git.io (git.io)[3.220.9.69]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/Nyr/openvpn-install/master/openvpn-install.sh [following]
--2020-04-24 18:35:01-- https://raw.githubusercontent.com/Nyr/openvpn-install/master/openvpn-install.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.8.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)[151.101.8.133]:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://raw.githubusercontent.com/Nyr/openvpn-install/master/openvpn-install.sh [following]
--2020-04-24 18:35:02-- https://raw.githubusercontent.com/Nyr/openvpn-install/master/openvpn-install.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.8.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)[151.101.8.133]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21758 (21K) [text/plain]
Saving to: 'openvpn-ubuntu-install.sh'

openvpn-ubuntu-install.sh 100%[=====>] 21.25K --.-KB/s in 0s

2020-04-24 18:35:02 (75.5 MB/s) - 'openvpn-ubuntu-install.sh' saved [21758/21758]

[vivek@sg-vpn-001 ~]$
```

ATTENTION: Do you want **password authentication** along with certificates? Try the following:

```
wget https://raw.githubusercontent.com/angristan/openvpn-inst.
```

Now we downloaded the script and it is time to make it executable. Hence, set up permissions using the chmod command:

```
chmod -v +x openvpn-ubuntu-install.sh  
mode of 'openvpn-ubuntu-install.sh' changed from 0644 (rw-r--r--) to  
0755 (rwxr-xr-x)
```

One can view the script using a text editor such as nano/vim:

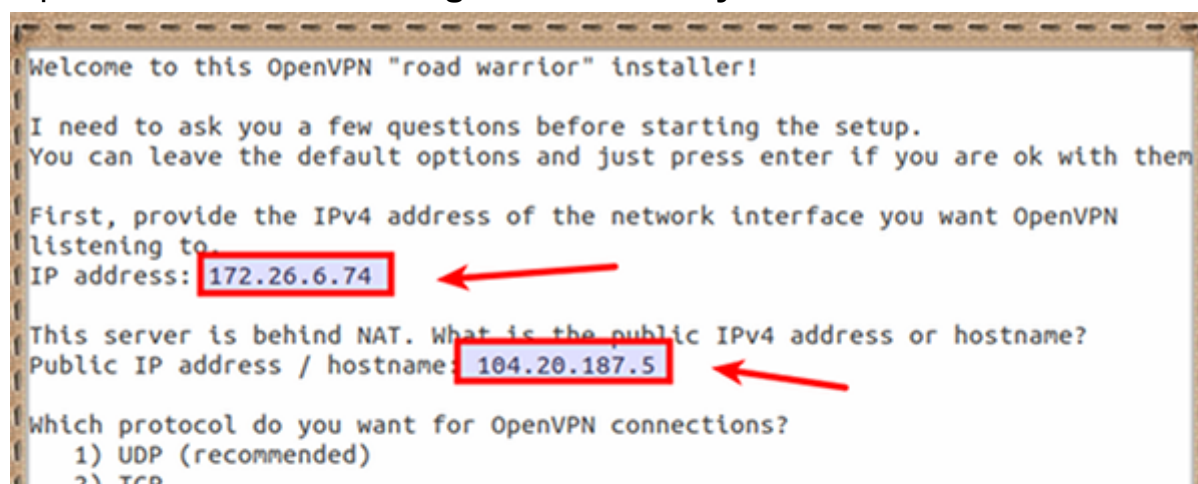
```
nano openvpn-ubuntu-install.sh
```

Run openvpn-ubuntu-install.sh script to install OpenVPN server

Now all you have to do is:

```
sudo ./openvpn-ubuntu-install.sh
```

Sample session **from AWS/Lightsail** where my cloud server is **behind NAT**:



```
Welcome to this OpenVPN "road warrior" installer!  
  
I need to ask you a few questions before starting the setup.  
You can leave the default options and just press enter if you are ok with them.  
  
First, provide the IPv4 address of the network interface you want OpenVPN  
listening to.  
IP address: 172.26.6.74  
  
This server is behind NAT. What is the public IPv4 address or hostname?  
Public IP address / hostname: 104.20.187.5  
  
Which protocol do you want for OpenVPN connections?  
1) UDP (recommended)  
2) TCP
```



```
Protocol [1-2]: 1
What port do you want OpenVPN listening to?
Port: 1194
Which DNS do you want to use with the VPN?
  1) Current system resolvers
  2) 1.1.1.1
  3) Google
  4) OpenDNS
  5) Verision
DNS [1-5]: 2
Finally, tell me your name for the client certificate.
Please, use one word only, no special characters.
Client name: desktop
Okay, that was all I needed. We are ready to set up your OpenVPN server now.
Press any key to continue...
```

Sample session from **Linode/DO** server where cloud server has **direct public IPv4 address**:

```
Welcome to this OpenVPN road warrior installer!

I need to ask you a few questions before starting setup.
You can use the default options and just press enter if you are ok with them.

Which protocol do you want for OpenVPN connections?
  1) UDP (recommended)
  2) TCP
Protocol [1]:
What port do you want OpenVPN listening to?
Port [1194]:
Which DNS do you want to use with the VPN?
  1) Current system resolvers
  2) 1.1.1.1
  3) Google
  4) OpenDNS
  5) NTT
  6) AdGuard
DNS [1]: 2
Finally, tell me a name for the client certificate.
Client name [client]: linuxdesktop
Okay, that was all I needed. We are ready to set up your OpenVPN server now.
Press any key to continue...
```

Hit the [Enter] key

© www.cyberciti.biz

If you downloaded the second script as per [step #3](#), you could set up a password for the client too. Here is how it will look on your screen:

Tell me a name for the client.

The name must consist of alphanumeric character. It may also include an underscore or a dash.

Client name: **linuxdesktop**



Do you want to protect the configuration file with a password?

(e.g. encrypt the private key with a password)

1) Add a passwordless client

2) Use a password for the client

Select an option [1-2]: **2**

 You will be asked for the client password below 

I strongly suggest that you always choose the DNS server option as 1.1.1.1 or Google DNS or any other DNS service provided that you trust as per your needs. Make sure you choose fast Geo-distributed DNS servers and reached from anywhere on the Internet. At the end we should see information as follows:

Your client configuration is available at: /root/linuxdesktop.ovpn

If you want to add more clients, just run this script again!

How do I start/stop/restart OpenVPN server on Ubuntu 20.04 LTS?

We need to use the systemctl command as follows:

Stop the OpenVPN server

```
sudo systemctl stop openvpn-server@server.service
## OR when using password to protect vpn ##
sudo systemctl stop openvpn@server.service
```

Start the OpenVPN server


```
sudo systemctl start openvpn-server@server.service
## OR when using password to protect vpn ##
sudo systemctl start openvpn@server.service
```

Restart the OpenVPN server after changing configuration options

```
sudo systemctl restart openvpn-server@server.service
## OR when using password to protect vpn ##
sudo systemctl restart openvpn@server.service
```

Show status of the OpenVPN server

```
sudo systemctl status openvpn-server@server.service
## OR when using password to protect vpn ##
sudo systemctl status openvpn@server.service
```

- openvpn-server@server.service - OpenVPN service for server
 - Loaded: loaded (/lib/systemd/system/openvpn-server@.service;
 - Active: **active (running)** since Tue 2020-06-16 09:01:19 UTC;
 - Docs: man:openvpn(8)
 - <https://community.openvpn.net/openvpn/wiki/Openvpn24>
 - <https://community.openvpn.net/openvpn/wiki/HOWTO>
 - Main PID: 1982 (openvpn)
 - Status: "Initialization Sequence Completed"
 - Tasks: 1 (limit: 2282)
 - Memory: 1.1M
 - CGroup: /system.slice/system-openvpn\x2dserver.slice/openvpn
 - └─1982 /usr/sbin/openvpn --status /run/openvpn-serve
- ```
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: Socket Buffers: R=[2129
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: UDPv4 link local (bound
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: UDPv4 link remote: [AF_
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: GID set to nogroup
```

```
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: UID set to nobody
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: MULTI: multi_init calle
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: IFCONFIG POOL IPv6: (IP
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: IFCONFIG POOL: base=10.
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: IFCONFIG POOL LIST
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: Initialization Sequence
```

**Warning:** AWS EC2/Lightsail users need to open the default OpenVPN port UDP/1194 using [Amazon EC2 security groups](#) for the Linux instances feature. Run the following ss command to see your OpenVPN port on EC2 cloud instance:

```
sudo ss -tulpn | grep -i openvpn
```

## Firewall ?

Create rules to open ports to the internet, or to a specific IP address or range.

[Learn more about firewall rules](#)

+ Add rule

| Application | Protocol | Port or range / Code | Restricted to                                 |   |   |
|-------------|----------|----------------------|-----------------------------------------------|---|---|
| SSH         | TCP      | 22                   | Any IP address<br>Lightsail browser SSH/RDP ? | ✍ | 🗑 |
| HTTP        | TCP      | 80                   | Any IP address                                | ✍ | 🗑 |
| Custom      | UDP      | 1194                 | Any IP address                                | ✍ | 🗑 |

AWS EC2/Lightsail open UDP port 1194

OpenVPN UDP port 1194 opened using AWS EC2/Lightsail Linux instance

## Step 4 – Connect an OpenVPN server using iOS/Android/Linux/Windows desktop client

**Note for Windows user:** Please download scp clients such as [PSCP](#) or [WinSCP](#) to copy the .ovpn file to your Windows machine. Some versions of [windows may come with both ssh/sftp/ssh clients](#).

On server your will find a client configuration file called /root/linuxdesktop.ovpn. All you have to do is copy this file to your local desktop using the scp command (replace 172.104.177.197 with your actual IP address):

```
scp root@172.104.177.197:/root/linuxdesktop.ovpn .
```

If root is not allowed to log in into the server, try the following scp command:

```
ssh vivek@172.104.177.197 "sudo -S cat
/root/linuxdesktop.ovpn" > linuxdesktop.ovpn
```

Next, provide this file to your OpenVPN client to connect:

1. [Apple iOS client](#)
2. [Android client](#)
3. [Apple MacOS \(OS X\) client](#)
4. [Windows 8/10 client](#)

**Tip:** Forgotten your .ovpn file location on the Ubuntu 20.04 LTS server? Try locating by typing the following command:

```
sudo find / -iname "*.ovpn"
```

## Unable to bind service to VPN port?

It would help if you [force Linux to bind an IP address that doesn't exist with net.ipv4.ip\\_nonlocal\\_bind](#) Linux kernel option. For example, during Ubuntu 20.04

LTS startup (boot) time, OpenVPN IP addresses such as 10.8.0.1/32 may not be available to services such as HTTPD or SSHD. Edit the following file:

```
$ sudo nano /etc/sysctl.d/1000-force-openvpn-bind.conf
OR when using password to protect vpn
$ sudo vim /etc/sysctl.d/1000-force-openvpn-bind.conf
```

Append the following:

```
net.ipv4.ip_nonlocal_bind=1
```

Reload changes using the sysctl command:

```
$ sudo sysctl -p /etc/sysctl.d/1000-force-openvpn-bind.conf
```

## Linux Desktop: OpenVPN client configuration

First, install the openvpn client for your desktop using the [yum command](#)/dnf command/[apt command](#):

```
sudo dnf install openvpn
```

OR

```
sudo apt install openvpn
```

Next, copy desktop.ovpn as follows:

```
sudo cp linuxdesktop.ovpn /etc/openvpn/client.conf
```

Test connectivity from the CLI:

```
sudo openvpn --client --config /etc/openvpn/client.conf
```

Your Linux system will automatically connect when computer restart using openvpn script/service:

```
sudo systemctl start openvpn@client # <--- start client service
```

## Step 5 – Verify/test the connectivity

Simply visit [this page to check your IP address](#) and it much change to your VPN server IP address. Next, execute the following commands after connecting to OpenVPN server from your Linux desktop:

```
ping 10.8.0.1 #Ping to the OpenVPN server gateway
ip route #Make sure routing setup working
the following must return public IP address of OpenVPN
server ##
dig TXT +short o-o.myaddr.l.google.com @ns1.google.com
```



```
[vivek@nixcraft-wks01 ~]$ ping -c 4 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=39.6 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=40.3 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=39.3 ms
64 bytes from 10.8.0.1: icmp_seq=4 ttl=64 time=39.2 ms

--- 10.8.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 39.237/39.624/40.318/0.427 ms
[vivek@nixcraft-wks01 ~]$ dig TXT +short o-o.myaddr.l.google.com @ns1.google.com
"172.104.177.197"
[vivek@nixcraft-wks01 ~]$ ip route
0.0.0.0/1 via 10.8.0.1 dev tun0
default via 192.168.2.254 dev enp0s31f6 proto static metric 100
10.8.0.0/24 dev tun0 proto kernel scope link src 10.8.0.2
10.83.200.0/24 dev lxdbr0 proto kernel scope link src 10.83.200.1
128.0.0.0/1 via 10.8.0.1 dev tun0
172.104.177.197 via 192.168.2.254 dev enp0s31f6
192.168.2.0/24 dev enp0s31f6 proto kernel scope link src 192.168.2.25 metric 100
[vivek@nixcraft-wks01 ~]$
```

Must get VPN server IP here

© www.cyberciti.biz

## Step 6 – How to add or remove a new VPN user

## with a certificate

You need to run the same script again for adding or removing a new VPN user to TLS certificate. For instance:

```
$ sudo ./openvpn-ubuntu-install.sh
```

You will see menu as follows:

OpenVPN is already installed.

Select an option:

- 1) Add a new client
- 2) Revoke an existing client
- 3) Remove OpenVPN
- 4) Exit

Option:

Choose option # 1 to add a new VPN client/user and option # 2 to remove the existing VPN client and user. Let us add a new client/user called iphone:

```
OpenVPN is already installed.
Select an option:
 1) Add a new client
 2) Revoke an existing client
 3) Remove OpenVPN
 4) Exit
Option: 1
Provide a name for the client:
Name: iphone
Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020
Generating a RSA private key
+++++
+++++
writing new private key to '/etc/openvpn/server/easy-rsa/pki/easy-rsa-962.qT5r09/tmp.3Zi0qZ'

Using configuration from /etc/openvpn/server/easy-rsa/pki/easy-rsa-962.qT5r09/tmp.1Hclfw
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :ASN.1 12:'iphone'
Certificate is to be certified until Oct 10 10:23:28 2030 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

iphone added. Configuration available in: /root/iphone.ovpn
```

The screenshot shows the terminal output of the script. Red arrows and numbers 1 through 5 highlight the following steps: 1. Selecting option 1. 2. Providing the client name 'iphone'. 3. Generating the RSA private key. 4. Writing the new private key to the specified path. 5. The final message indicating the configuration is available in /root/iphone.ovpn.



# A note about trouble shooting

## OpenVPN server and client issues

Type the following commands on your Ubuntu 20.04 Linux LTS server. First, check OpenVPN server for errors:

```
sudo journalctl --identifier openvpn
```

---

```
-- Logs begin at Tue 2020-06-16 08:53:36 UTC, end at Tue 2020-06-16 09:09:57 UTC. --
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: OpenVPN 2.4.7 x86_64-pc-linux-gnu [SSL (OpenS
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: library versions: OpenSSL 1.1.1f 31 Mar 2020
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: Diffie-Hellman initialized with 2048 bit key
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: Outgoing Control Channel Encryption: Cipher '
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: Outgoing Control Channel Encryption: Using 25
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: Incoming Control Channel Encryption: Cipher '
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: Incoming Control Channel Encryption: Using 25
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: TUN/TAP device tun0 opened
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: TUN/TAP TX queue length set to 100
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: /sbin/ip link set dev tun0 up mtu 1500
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: /sbin/ip addr add dev tun0 10.8.0.1/24 broadc
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: /sbin/ip -6 addr add fddd:1194:1194::1/6
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: Could not determine IPv4/IPv6 protocol. Using
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: Socket Buffers: R=[212992->212992] S=[212992-
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: UDPv4 link local (bound): [AF_INET]45.79.125.:
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: UDPv4 link remote: [AF_UNSPEC]
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: GID set to nogroup
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: UID set to nobody
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: MULTI: multi_init called, r=256 v=256
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: IFCONFIG POOL IPv6: (IPv4) size=252, size_ipv
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: IFCONFIG POOL: base=10.8.0.2 size=252, ipv6=1
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: IFCONFIG POOL LIST
Jun 16 09:01:19 sg-vpn-001 openvpn[1982]: Initialization Sequence Completed
```

---

Is firewall rule setup correctly on your server? Use the [cat command](#) to see rules:

```
sudo cat /etc/systemd/system/openvpn-iptables.service
OR when using password to protect vpn
sudo cat /etc/systemd/system/iptables-openvpn.service
```

## Config:

```

[Unit]
Before=network.target
[Service]
Type=oneshot
ExecStart=/sbin/iptables -t nat -A POSTROUTING -s 10.8.0.0/24 ! -d 10.8.0.0/24 -j SNAT
ExecStart=/sbin/iptables -I INPUT -p udp --dport 1194 -j ACCEPT
ExecStart=/sbin/iptables -I FORWARD -s 10.8.0.0/24 -j ACCEPT
ExecStart=/sbin/iptables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
ExecStop=/sbin/iptables -t nat -D POSTROUTING -s 10.8.0.0/24 ! -d 10.8.0.0/24 -j SNAT -
ExecStop=/sbin/iptables -D INPUT -p udp --dport 1194 -j ACCEPT
ExecStop=/sbin/iptables -D FORWARD -s 10.8.0.0/24 -j ACCEPT
ExecStop=/sbin/iptables -D FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
ExecStart=/sbin/ip6tables -t nat -A POSTROUTING -s fddd:1194:1194:1194::/64 ! -d fddd:11
ExecStart=/sbin/ip6tables -I FORWARD -s fddd:1194:1194:1194::/64 -j ACCEPT
ExecStart=/sbin/ip6tables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
ExecStop=/sbin/ip6tables -t nat -D POSTROUTING -s fddd:1194:1194:1194::/64 ! -d fddd:11
ExecStop=/sbin/ip6tables -D FORWARD -s fddd:1194:1194:1194::/64 -j ACCEPT
ExecStop=/sbin/ip6tables -D FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
RemainAfterExit=yes
[Install]
WantedBy=multi-user.target

```

Another option is to run [iptables command](#) and `sysctl` command commands to verify NAT rule setup on your server:

```

sudo iptables -t nat -L -n -v
sysctl net.ipv4.ip_forward
sudo cat /etc/sysctl.d/30-openvpn-forward.conf
OR when using password to protect vpn
sudo cat /etc/sysctl.d/99-openvpn.conf

```

```

[vivek@sg-vpn-001 ~]$ sudo iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 508 packets, 35215 bytes)
 pkts bytes target prot opt in out source destination
Chain INPUT (policy ACCEPT 16 packets, 1030 bytes)
 pkts bytes target prot opt in out source destination
Chain OUTPUT (policy ACCEPT 68 packets, 5525 bytes)
 pkts bytes target prot opt in out source destination
Chain POSTROUTING (policy ACCEPT 68 packets, 5525 bytes)
 20 5214 SNAT all -- * * 10.8.0.0/24 !10.8.0.0/24 to:172.104.177.197
[vivek@sg-vpn-001 ~]$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1
[vivek@sg-vpn-001 ~]$

```

**OpenVPN NAT Firewall rules**

**Linux Packet forwarding**

© www.cyberciti.biz

## NAT Firewall OpenVPN Rules Verification

Insert the rules if not inserted using the following command:

```
sudo systemctl start openvpn-iptables.service
OR when using password to protect vpn
sudo systemctl start iptables-openvpn.service
sudo sysctl -w net.ipv4.ip_forward=1
sudo sysctl -p -f /etc/sysctl.d/30-openvpn-forward.conf
OR when using password to protect vpn
sudo sysctl -p -f /etc/sysctl.d/99-openvpn.conf
```

Is OpenVPN server running and port is open? Use the ss command or netstat command and [pidof command](#)/ps command:

```
1194 is the openvpn server port
netstat -tulpn | grep :1194
1194 is the openvpn server port
ss -tulpn | grep :1194
is the openvpn server running?
ps aux | grep openvpn
is the openvpn server running?
ps -C openvpn
find the openvpn server PID
pidof openvpn
```

```
vivek@aws-mum-1a-vpn-01:~$ pidof openvpn
9130
vivek@aws-mum-1a-vpn-01:~$
vivek@aws-mum-1a-vpn-01:~$ ss -tulpn | grep 1194
udp UNCONN 44928 0 0.0.0.0:1194 0.0.0.0:*
vivek@aws-mum-1a-vpn-01:~$
vivek@aws-mum-1a-vpn-01:~$ netstat -tulpn | grep :1194
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
udp 44928 0 0.0.0.0:1194 0.0.0.0:*
vivek@aws-mum-1a-vpn-01:~$ ps -C openvpn
 PID TTY TIME CMD
 9130 ? 00:00:04 openvpn
vivek@aws-mum-1a-vpn-01:~$
```

If not running, restart the OpenVPN server:

```
sudo systemctl restart openvpn-server@server.service
```

Look out for errors:

```
sudo systemctl status openvpn-server@server.service
```

Can the Linux desktop client connect to the OpenVPN server machine? First you need to run a simple test to see if the OpenVPN server port (UDP 1194) accepts connections:

```
nc -vu 172.104.177.197 1194
```

```
Connection to 172.104.177.197 port [udp/openvpn] succeeded!
```

If not connected it means either a Linux desktop firewall or your router is blocking access to server. Make sure both client and server using same protocol and port, e.g. UDP port 1194.

## Conclusion

Congratulations. You successfully set up an OpenVPN server on Ubuntu Linux 20.04 LTS server running in the cloud. See the OpenVPN website [here](#), Ubuntu page [here](#) and Github [script page here](#) for additional information or use the [man command](#)/[help command](#) to read docs locally:

```
man openvpn
openvpn --help
Use the more command/less command as a filter
openvpn --help | more
```

This entry is **11** of **13** in the **OpenVPN Tutorial** series. Keep reading the rest of the series:

1. [How To Setup OpenVPN Server In 5 Minutes on Ubuntu Server](#)
2. [Install Pi-hole with an OpenVPN to block ads](#)
3. [Update/upgrade Pi-hole with an OpenVPN](#)
4. [OpenVPN server on Debian 9/8](#)
5. [Import a OpenVPN .ovpn file with Network Manager](#)
6. [Ubuntu 18.04 LTS Set Up OpenVPN Server In 5 Minutes](#)
7. [CentOS 7 Set Up OpenVPN Server In 5 Minutes](#)
8. [Pi-Hole and Cloudflare DoH config](#)
9. [Debian 10 Set Up OpenVPN Server In 5 Minutes](#)
10. [CentOS 8 OpenVPN server in 5 mintues](#)
11. [Ubuntu 20.04 LTS OpenVPN server in 5 mintues](#)
12. [Debian 11 set up OpenVPN server in 5 mintues](#)
13. [Ubuntu 22.04 LTS Set Up OpenVPN Server In 5 Minutes](#)

Did you notice? 🧐

nixCraft is ad-free to protect your privacy and security. We rely on reader support to keep the site running. Please consider subscribing to us on Patreon or supporting us with a one-time support through PayPal. Your support will help us cover the costs of hosting, CDN, DNS, and tutorial creation.

[Join Patreon →](#)

[PayPal →](#)

---

**About the author:** Vivek Gite is the founder of nixCraft, the oldest running blog about Linux and open source. He wrote more than 7k+ posts and helped numerous readers to master IT topics. Join the nixCraft community via [RSS Feed](#) or [Email Newsletter](#).

🙄 Was this helpful? Please add [a comment to show your appreciation or feedback](#).