OpenVPN: Securely connect to your device from any place

Check GitHub for helpful DevOps tools:

Michael Robotics

Hi, I'm Michal. I'm a Robotics Engineer and DevOps enthusiast. My mission is to create skill-learning platform that combats information overload by adhering to the set of principles: simplify, prioritize, and execute.

https://github.com/MichaelRobotics

Ask Personal AI Document assistant to learn interactively (FASTER)!

1 Download PDF

2 Go to website

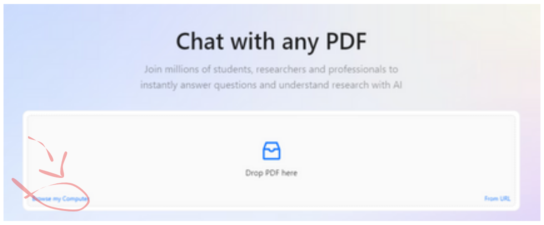
3 Browse file

4 Chat with Document

1 https://github.com/MichaelRobotics/DevOpsTools/blob/main/OpenVPN.pdf

2 Click there to go to ChatPdf website

3



Chat with any PDF

Join millions of students, researchers and professionals to instantly answer questions and understand research with AI

4 Ask questions about document!

Complety new to Linux and Networking?

Essential for this PDF is a thorough knowledge of networking. I highly recommend the HTB platform's networking module, which offers extensive information to help build a comprehensive understanding.

HTB - Your Cyber Performance Center

We provide a human-first platform creating and maintaining high performing cybersecurity individuals and organizations.

https://www.hackthebox.com/

What is OpenVPN?

OpenVPN is an open-source virtual private network (VPN) protocol that provides secure point-to-point or site-to-site connections using encrypted tunnels. It supports both remote access and private network-to-network communication, ensuring data security across the internet.

How OpenVPN works?

The process of converting human-readable domain names, like "example.com," into machine-readable IP addresses involves several DNS servers working in sequence. These include recursive resolvers, root servers, TLD (top-level domain) servers, and authoritative servers, which collaborate to efficiently locate and return the correct IP address for a requested domain.

OpenVPN: Why and When

OpenVPN is essential for securing remote access, especially for distributed workforces that need to access SaaS applications, corporate resources, or public cloud services safely. It is typically used when an organization needs scalable, secure, and flexible remote access that extends beyond the perimeter of traditional network security.

Typical Use Case
A company with remote or hybrid employees uses OpenVPN to allow secure access to internal resources and third-party cloud applications, ensuring that employees work securely from any location.

System Requirements

- 1 GB RAM (minimum, 2 GB recommended for better performance)
- 10 GB free storage (more may be required depending on the size of the DNS zone files and logs)
- Ubuntu 22.04

If you want to install it on a different Linux distro, ask in the comments and I will write an Ansible playbook or bash script.

OpenVPN: Main components & packages

- openvpn – Core VPN software for secure connections.
- openssl – Cryptography toolkit for encryption and certificates.
- ca-certificates - Trusted CA certificates for SSL/TLS validation.
- iptables - Firewall tool to control network traffic for VPN.

OpenVPN Server: How to setup on Linux (Using mostly CLI)

1) Download OpenVPN installer script

This script will let you set up your own VPN server in no more than a minute, even if you haven't used OpenVPN before. It has been designed to be as unobtrusive and universal as possible.

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

Shoutout to script creator! check GitHub:

openvpn-install

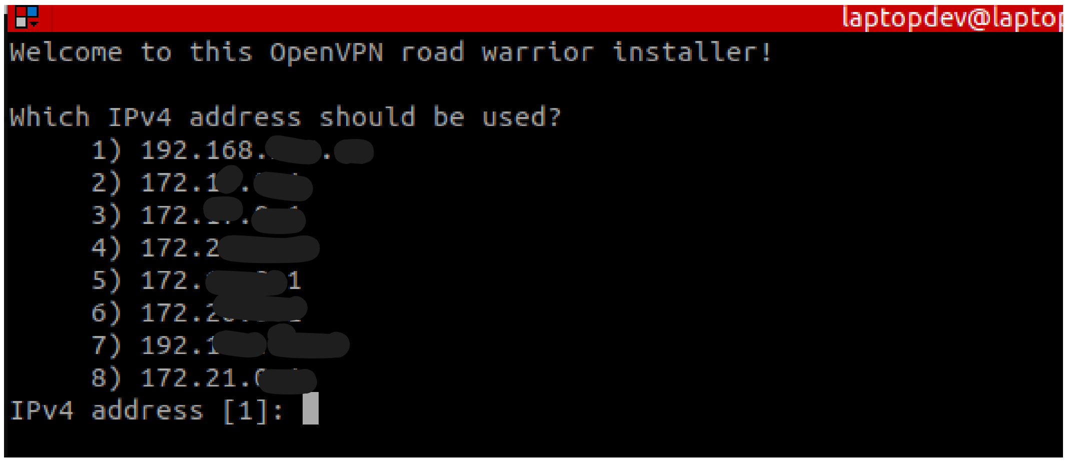
OpenVPN road warrior installer for Ubuntu, Debian, AlmaLinux,

https://github.com/Nyr/openvpn-install

Run script:

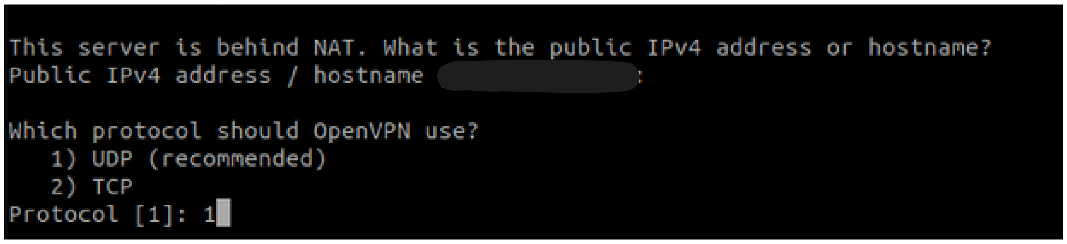
```
bash openvpn-install.sh
```

Select the appropriate IPv4 address to use (typically, choose an IP address from your WLAN or LAN network):



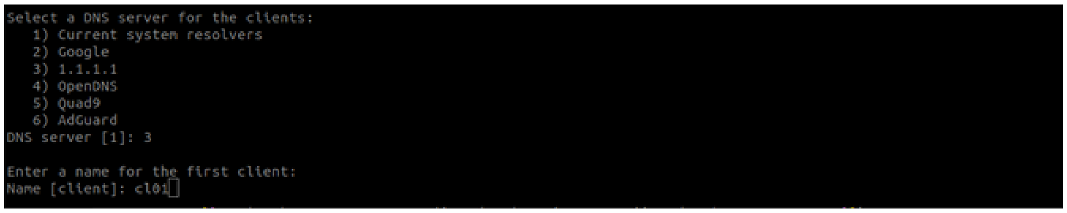
```
laptopdev@lapto
Welcome to this OpenVPN road warrior installer!

Which IPv4 address should be used?
     1) 192.168.[redacted].[redacted]
     2) 172.1[redacted].[redacted]
     3) 172.[redacted].[redacted]
     4) 172.2[redacted]
     5) 172.[redacted]1
     6) 172.2[redacted]
     7) 192.1[redacted]
     8) 172.21.[redacted]
IPv4 address [1]:
```

Select your global IP address (Typically, the default option is pre-selected; you can simply press Enter):



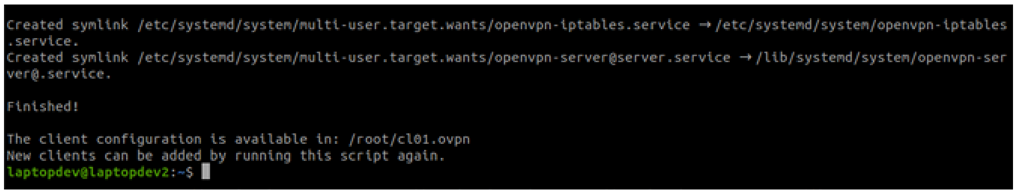
```
This server is behind NAT. What is the public IPv4 address or hostname?
Public IPv4 address / hostname [redacted]:

Which protocol should OpenVPN use?
   1) UDP (recommended)
   2) TCP
Protocol [1]: 1
```

Select a DNS server (choose according to your preference):



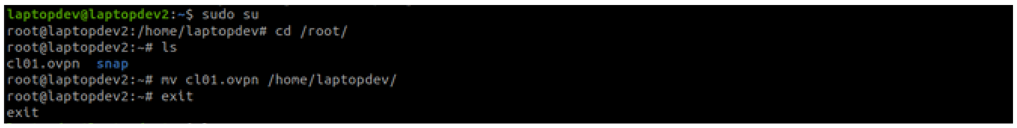
```
Select a DNS server for the clients:
   1) Current system resolvers
   2) Google
   3) 1.1.1.1
   4) OpenDNS
   5) Quad9
   6) AdGuard
DNS server [1]: 3

Enter a name for the first client:
Name [client]: cl01
```

Your VPN has been successfully configured!



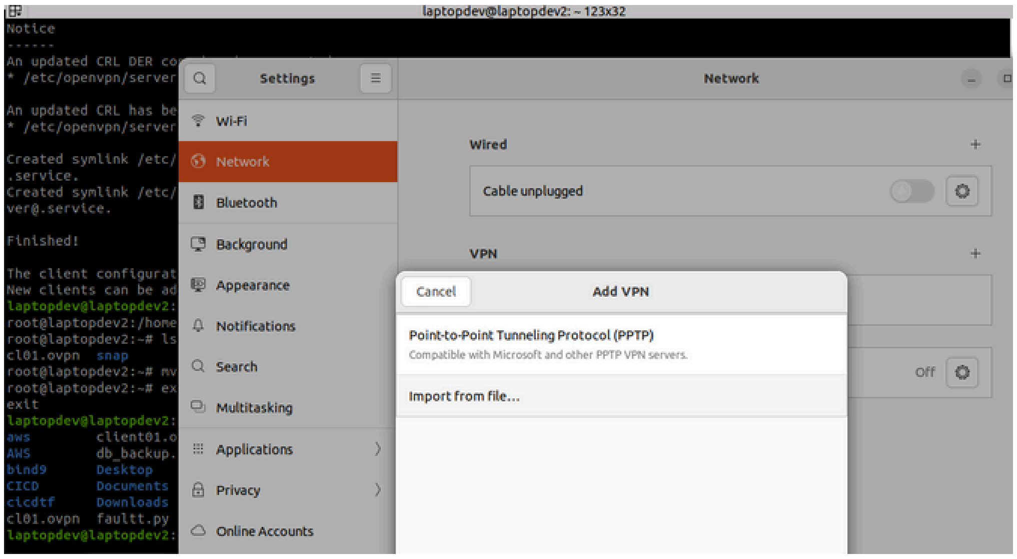
```
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-iptables.service → /etc/systemd/system/openvpn-iptables
.service.
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-server@server.service → /lib/systemd/system/openvpn-ser
ver@.service.

Finished!

The client configuration is available in: /root/cl01.ovpn
New clients can be added by running this script again.
laptopdev@laptopdev2:~$
```

Move ovpn file from root to your home directory



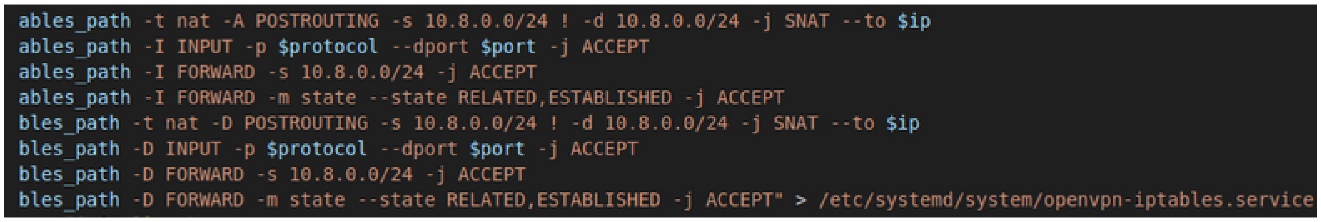
```
laptopdev@laptopdev2:~$ sudo su
root@laptopdev2:/home/laptopdev# cd /root/
root@laptopdev2:~# ls
cl01.ovpn  snap
root@laptopdev2:~# mv cl01.ovpn /home/laptopdev/
root@laptopdev2:~# exit
exit
```

Send it to client and load (Works on Linux OS)



OpenVPN Server: Understand important parts of openvpn-install.sh script

1) VPN routing



```
ables_path -t nat -A POSTROUTING -s 10.8.0.0/24 ! -d 10.8.0.0/24 -j SNAT --to $ip
ables_path -I INPUT -p $protocol --dport $port -j ACCEPT
ables_path -I FORWARD -s 10.8.0.0/24 -j ACCEPT
ables_path -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
bles_path -t nat -D POSTROUTING -s 10.8.0.0/24 ! -d 10.8.0.0/24 -j SNAT --to $ip
bles_path -D INPUT -p $protocol --dport $port -j ACCEPT
bles_path -D FORWARD -s 10.8.0.0/24 -j ACCEPT
bles_path -D FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT" > /etc/systemd/system/openvpn-iptables.service
```

```
$iptables_path -t nat -A POSTROUTING -s 10.8.0.0/24 ! -d 10.8.0.0/24 -j SNAT --to $ip
```

Adds a NAT (Network Address Translation) rule for the VPN traffic.

-t nat specifies that this is a NAT rule.

-A POSTROUTING appends a rule to the POSTROUTING chain, which is used to modify packets after routing.

-s 10.8.0.0/24 specifies the source IP range, which is the VPN subnet.

! -d 10.8.0.0/24 ensures that traffic is only NAT-ed if the destination is outside the VPN subnet.

-j SNAT --to $ip means that the source IP is replaced with the server's IP ($ip) for external traffic.

```
$iptables_path -I INPUT -p $protocol --dport $port -j ACCEPT
```

Inserts a rule to allow incoming traffic on the specified VPN protocol ($protocol, likely udp or tcp) and port ($port).

```
$iptables_path -I FORWARD -s 10.8.0.0/24 -j ACCEPT
```

Allows forwarding of packets from the VPN subnet (10.8.0.0/24)

```
$iptables_path -I FORWARD -m state --state RELATED,ESTABLISHED -j
ACCEPT
```

Ensures that forwarded traffic that is part of an established connection is allowed. This rule uses state matching to track connection states and is crucial for proper functioning of the VPN.
Without this rule, the VPN server might drop packets that are part of legitimate, ongoing connections. For example, when a VPN client requests a webpage, the server's response could be blocked if this rule wasn't in place because the system might not recognize it as part of an established session

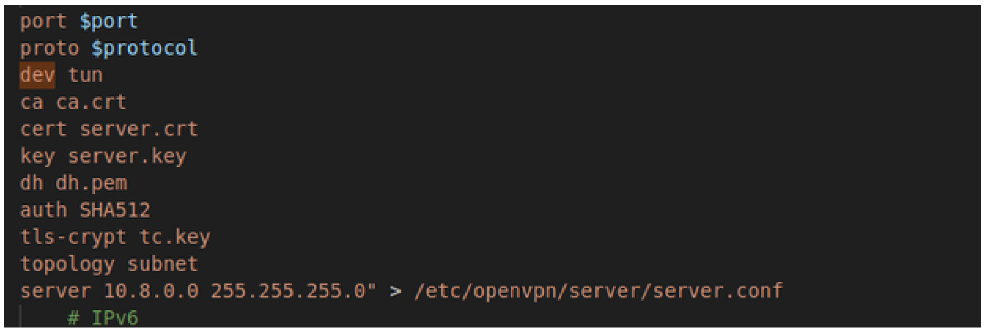
ExecStop Commands:

These are the inverse of the ExecStart rules. When the service is stopped, the corresponding iptables rules are deleted (-D):

IPv6 Support

The configuration also includes conditional rules for IPv6

2) VPN config



```
port $port
proto $protocol
dev tun
ca ca.crt
cert server.crt
key server.key
dh dh.pem
auth SHA512
tls-crypt tc.key
topology subnet
server 10.8.0.0 255.255.255.0" > /etc/openvpn/server/server.conf
    # IPv6
```

local $ip: Specifies the IP address where the OpenVPN server listens for connections.

port $port: Defines the port OpenVPN will use (commonly 1194).

proto $protocol: Sets the communication protocol (udp for lower latency or tcp for reliability).

dev tun: Uses a TUN device for routing IP packets (layer 3), ideal for VPNs.

ca ca.crt: Points to the Certificate Authority (CA) certificate, used to authenticate the server.

cert server.crt: Specifies the server's SSL certificate for client authentication.

key server.key: Defines the server's private key for encryption.

dh dh.pem: Uses Diffie-Hellman parameters for secure key exchange.

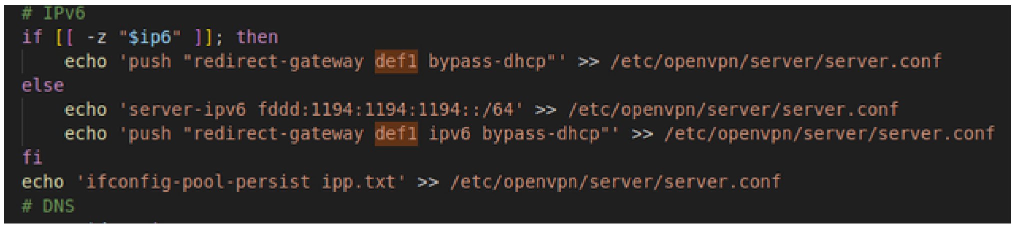
auth SHA512: Ensures secure connection integrity using the SHA512 hash algorithm.

tls-crypt tc.key: Encrypts the TLS control channel, adding protection against attacks.

topology subnet: Assigns a unique IP address to each VPN client using a subnet topology.

server 10.8.0.0 255.255.255.0: Configures the VPN subnet (10.8.0.0/24), assigning IPs to clients and enabling communication within this range.

3) Force all traffic to go through vpn



```
# IPv6
if [[ -z "$ip6" ]]; then
    echo 'push "redirect-gateway def1 bypass-dhcp"' >> /etc/openvpn/server/server.conf
else
    echo 'server-ipv6 fddd:1194:1194:1194::/64' >> /etc/openvpn/server/server.conf
    echo 'push "redirect-gateway def1 ipv6 bypass-dhcp"' >> /etc/openvpn/server/server.conf
fi
echo 'ifconfig-pool-persist ipp.txt' >> /etc/openvpn/server/server.conf
# DNS
```

If no IPv6:

push "redirect-gateway def1 bypass-dhcp":

Redirects all IPv4 traffic through the VPN, ensuring local DHCP traffic is unaffected.

If IPv6 is configured:

server-ipv6 fddd:1194:1194:1194::/64:

Defines the IPv6 subnet for VPN clients.

push "redirect-gateway def1 ipv6 bypass-dhcp":

Redirects both IPv4 and IPv6 traffic through the VPN.

i**fconfig-pool-persist ipp.txt:** Stores client-to-IP mappings so clients keep the same IP across sessions.

Purpose

Traffic Redirection: Ensures all IPv4/IPv6 traffic is routed securely through the VPN.

Persistent IPs: Keeps client IPs consistent across sessions.

Common troubleshooting

1) Connection Failure

Description: The client cannot connect to the OpenVPN server.
Causes: Incorrect server address, port issues, firewall blocking the connection, or server not running.

2) Authentication Errors

Description: The client fails to authenticate with the server.
Causes: Incorrect username/password, expired or incorrect certificates, or wrong client configuration.

3) Slow Speeds or High Latency

Description: VPN connection is slow or has high ping times.
Causes: Network congestion, poor server location, or using a heavy encryption method that may slow down the connection.

4) **Check the OpenVPN man page**

5) **If everything is a complete mess**

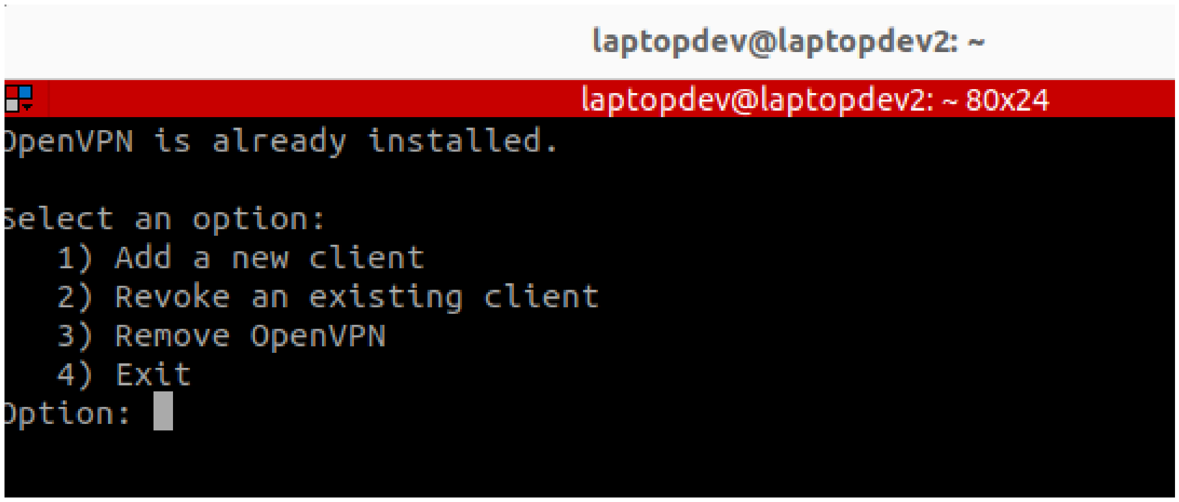
Remove the OpenVPN and revert the configuration to its previous state.

OpenVPN: How to remove

1) Run script

Afer running script, choose Remove VPN option

```
$ bash openvpn-install.sh
```



```
OpenVPN is already installed.

Select an option:
   1) Add a new client
   2) Revoke an existing client
   3) Remove OpenVPN
   4) Exit
Option: 
```

Learn more about OpenVPN

Check OpenVPN, they have great docs

OpenVPN

Deliver secure remote access with OpenVPN.

https://openvpn.net/as-docs/general.html

Share, comment, DM and check GitHub for scripts & playbooks created to automate process.

Check my GitHub

Michael Robotics

Hi, I'm Michal. I'm a Robotics Engineer and DevOps enthusiast. My mission is to create skill-learning platform that combats skill information overload by adhering to the set of principles: simplify, prioritize, and execute.

https://github.com/MichaelRobotics

PS.

If you need a playbook or bash script to manage KVM on a specific Linux distribution, feel free to ask me in the comments or send a direct message!