OpenVPN and DNS Firewall or Sinkhole

OpenVPN is an open-source VPN protocol allowing secure online access from point-to-point completely free of charge.OpenVPN in particular has garnered a lot of attention, due to its open-source nature and the fact it's free.OpenVPN is unlike most other VPNs, as it's an open-source encryption protocol. This means users enjoy a notably secure network thanks to the vast OpenSSL Library that is completely unowned.

DNS Firewall is a Security technology solution, protecting against volumetric, exploit and stealth attacks for both public and private DNS infrastructures.

DNS Firewall is the root of the Internet and protects from the attack at the root.

DNS Firewall governs/inspects DNS Queries at port 53.

DNS Firewall - **Mitigate At The Source** - Thwart initial infection and phishing.

DNS Firewall - **Adapt To Evolving Threat Landscape** - Threat Intelligence services to keep pace with malicious domains/IPs.

DNS Firewall - **Proactively Prevent New Attacks** - Detect and block malware communication with C&C (Command and Control) server.

DNS Sinkhole is allowing very specific DNS queries and denying the rest by default. So thereby achieving the easiest and best DNS Security.

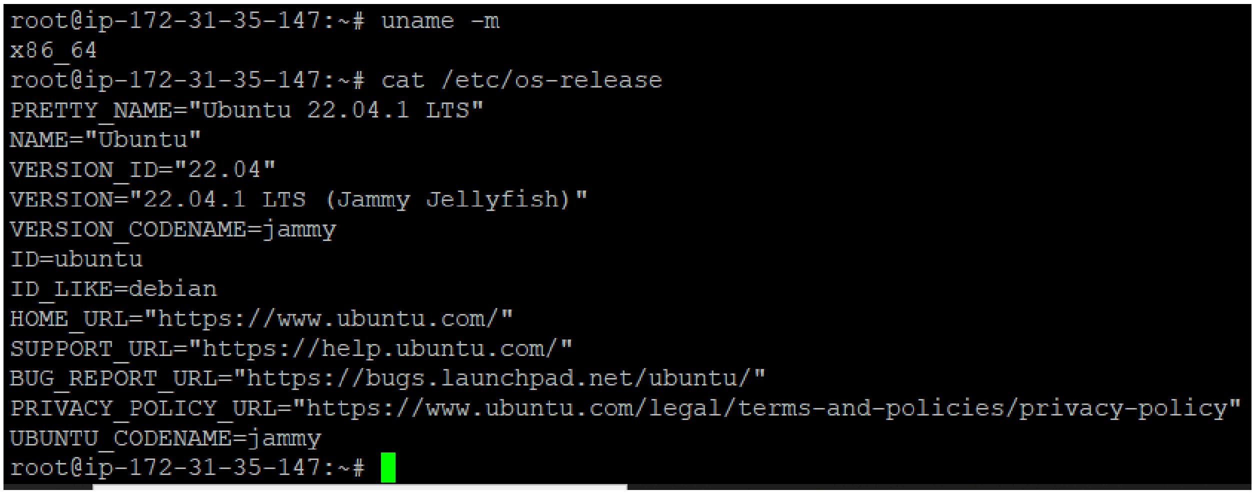
When we combine **OpenVPN server** with DNS Firewall or a DNS Sinkhole , we can have safe and secure internet surfing.

OpenVPN Client is available for free for Windows,MacOS,Linux,Android and iOS.

Use Case

In our use case the OpenVPN server will be deployed in AWS Cloud and tested using Windows OpenVPN client.

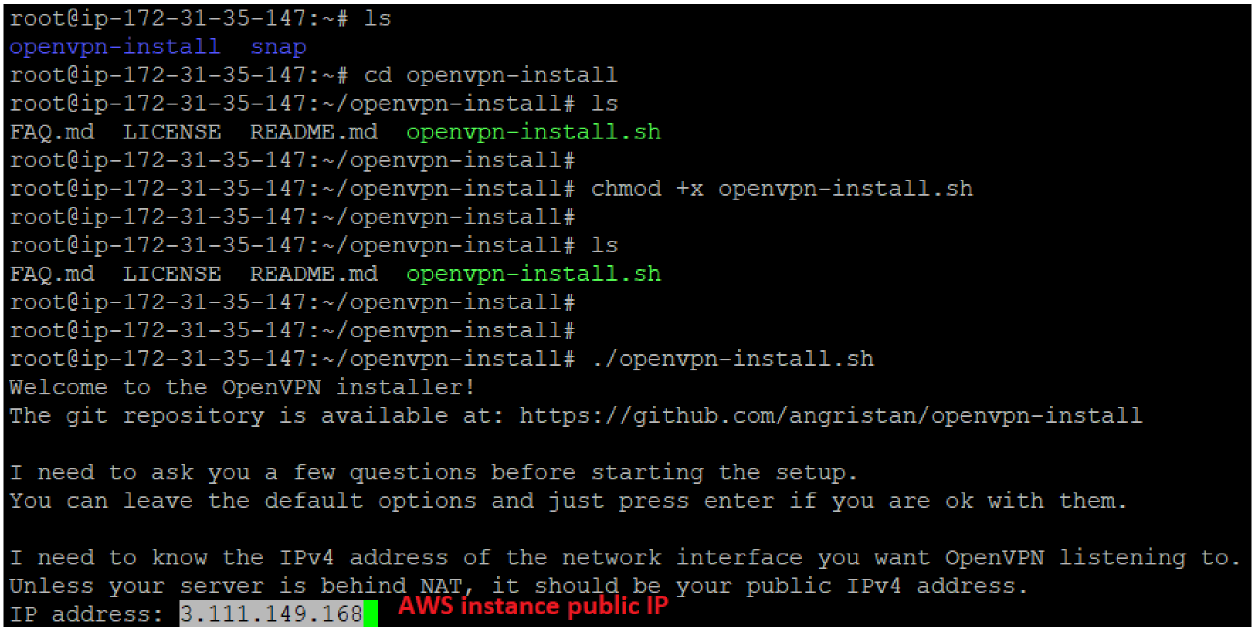
Tested OpenVPN server version,



```
root@ip-172-31-35-147:~# uname -m
x86_64
root@ip-172-31-35-147:~# cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.1 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
root@ip-172-31-35-147:~#
```

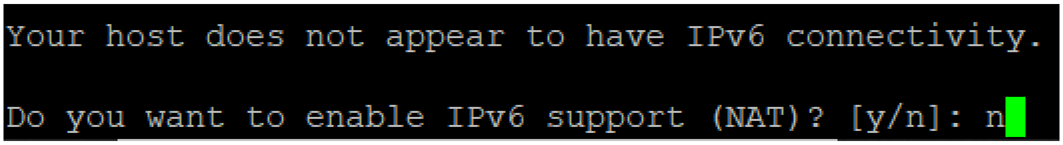
Login into server console and do the below,

git clone https://github.com/angristan/openvpn-install.git

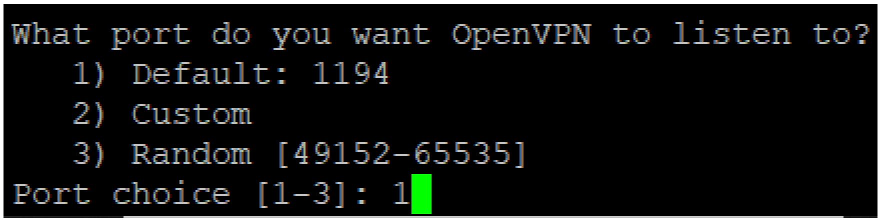


```
root@ip-172-31-35-147:~# ls
openvpn-install  snap
root@ip-172-31-35-147:~# cd openvpn-install
root@ip-172-31-35-147:~/openvpn-install# ls
FAQ.md  LICENSE  README.md  openvpn-install.sh
root@ip-172-31-35-147:~/openvpn-install#
root@ip-172-31-35-147:~/openvpn-install# chmod +x openvpn-install.sh
root@ip-172-31-35-147:~/openvpn-install#
root@ip-172-31-35-147:~/openvpn-install# ls
FAQ.md  LICENSE  README.md  openvpn-install.sh
root@ip-172-31-35-147:~/openvpn-install#
root@ip-172-31-35-147:~/openvpn-install#
root@ip-172-31-35-147:~/openvpn-install# ./openvpn-install.sh
Welcome to the OpenVPN installer!
The git repository is available at: https://github.com/angristan/openvpn-install

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.

I need to know the IPv4 address of the network interface you want OpenVPN listening to.
Unless your server is behind NAT, it should be your public IPv4 address.
IP address: 3.111.149.168
```

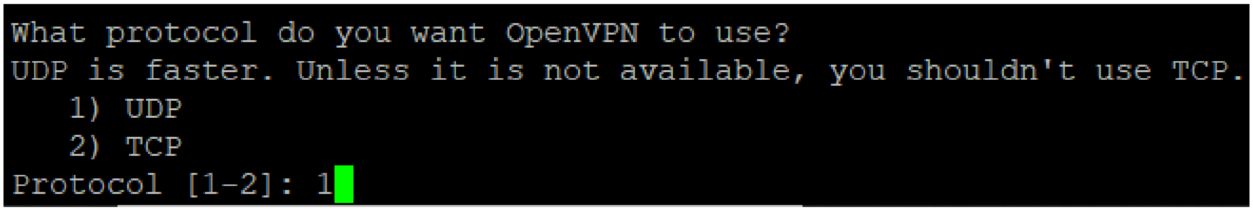
AWS instance public IP



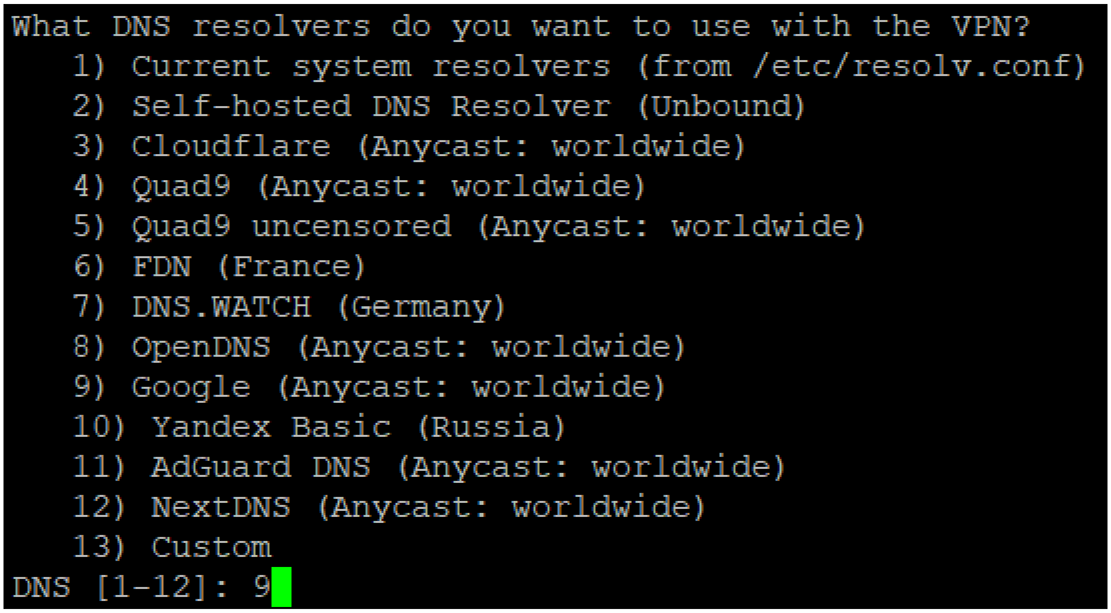
```
Your host does not appear to have IPv6 connectivity.

Do you want to enable IPv6 support (NAT)? [y/n]: n
```



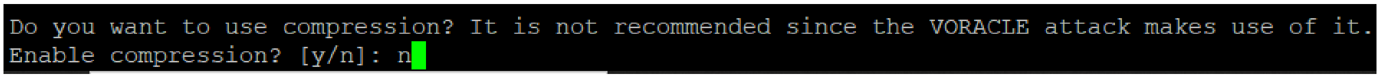
```
What port do you want OpenVPN to listen to?
   1) Default: 1194
   2) Custom
   3) Random [49152-65535]
Port choice [1-3]: 1
```



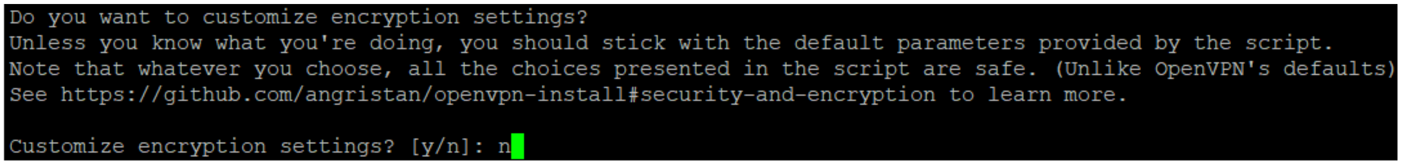
```
What protocol do you want OpenVPN to use?
UDP is faster. Unless it is not available, you shouldn't use TCP.
   1) UDP
   2) TCP
Protocol [1-2]: 1
```



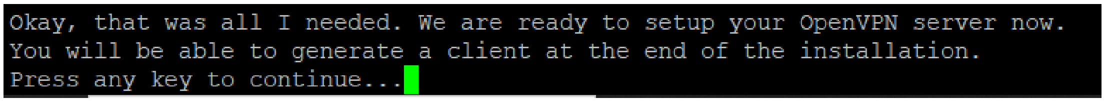
```
What DNS resolvers do you want to use with the VPN?
   1) Current system resolvers (from /etc/resolv.conf)
   2) Self-hosted DNS Resolver (Unbound)
   3) Cloudflare (Anycast: worldwide)
   4) Quad9 (Anycast: worldwide)
   5) Quad9 uncensored (Anycast: worldwide)
   6) FDN (France)
   7) DNS.WATCH (Germany)
   8) OpenDNS (Anycast: worldwide)
   9) Google (Anycast: worldwide)
   10) Yandex Basic (Russia)
   11) AdGuard DNS (Anycast: worldwide)
   12) NextDNS (Anycast: worldwide)
   13) Custom
DNS [1-12]: 9
```



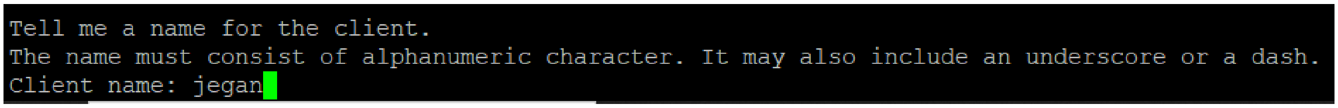
```
Do you want to use compression? It is not recommended since the VORACLE attack makes use of it.
Enable compression? [y/n]: n
```



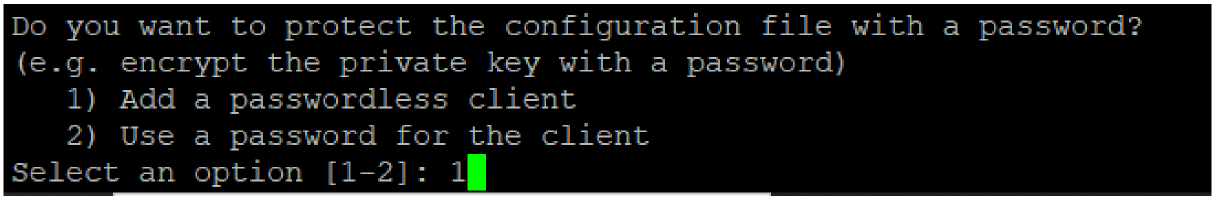
```
Do you want to customize encryption settings?
Unless you know what you're doing, you should stick with the default parameters provided by the script.
Note that whatever you choose, all the choices presented in the script are safe. (Unlike OpenVPN's defaults)
See https://github.com/angristan/openvpn-install#security-and-encryption to learn more.

Customize encryption settings? [y/n]: n
```



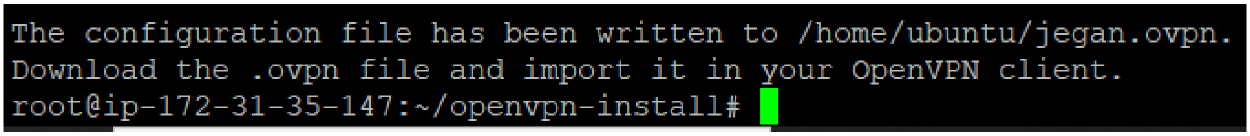
```
Okay, that was all I needed. We are ready to setup your OpenVPN server now.
You will be able to generate a client at the end of the installation.
Press any key to continue...
```



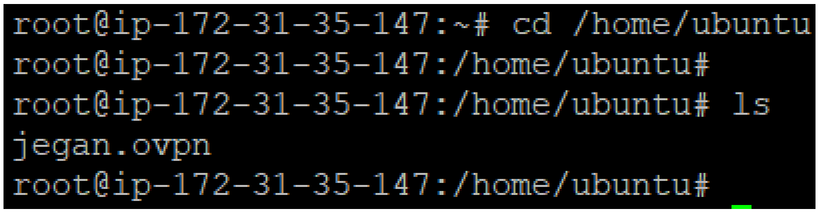
```
Tell me a name for the client.
The name must consist of alphanumeric character. It may also include an underscore or a dash.
Client name: jegan
```



```
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]: 1
```



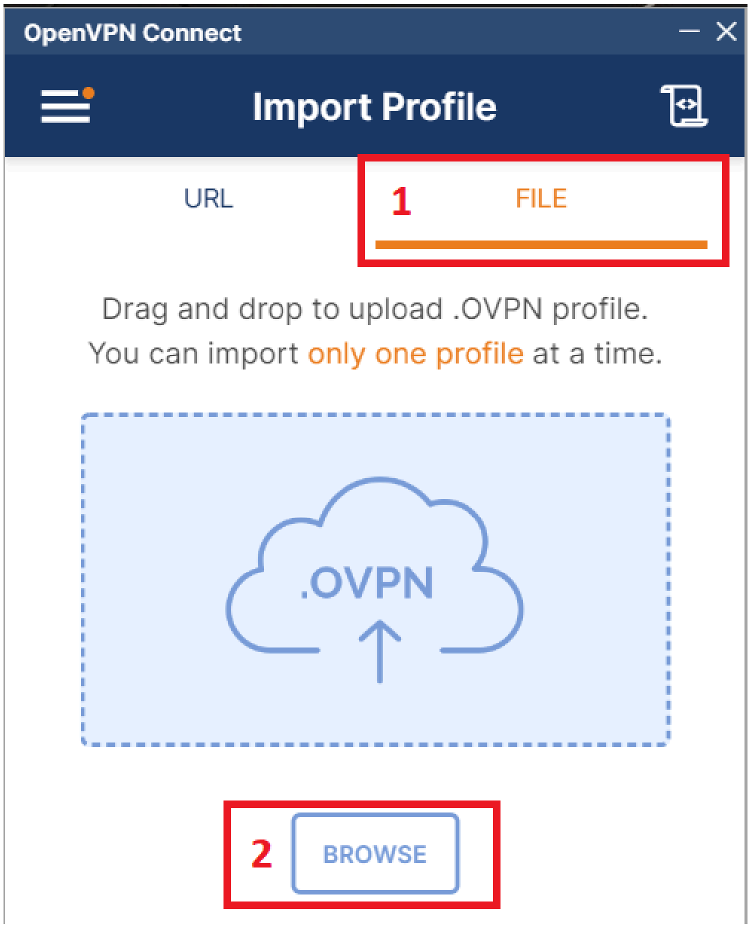
```
The configuration file has been written to /home/ubuntu/jegan.ovpn.
Download the .ovpn file and import it in your OpenVPN client.
root@ip-172-31-35-147:~/openvpn-install#
```



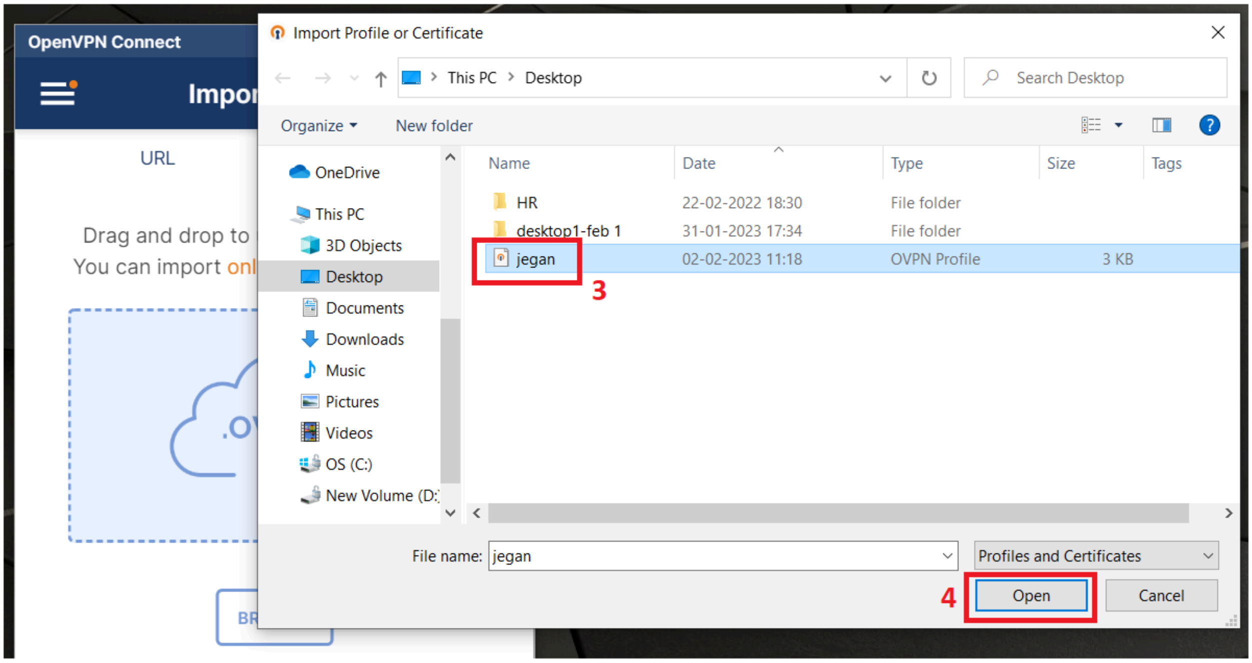
```
root@ip-172-31-35-147:~# cd /home/ubuntu
root@ip-172-31-35-147:/home/ubuntu#
root@ip-172-31-35-147:/home/ubuntu# ls
jegan.ovpn
root@ip-172-31-35-147:/home/ubuntu#
```

Copy the file /home/ubuntu/jegan.ovpn to your host and import the same in your openvpn client application as shown below to establish a secure connection between your host and VPN server.

Open OpenVPN connect client,

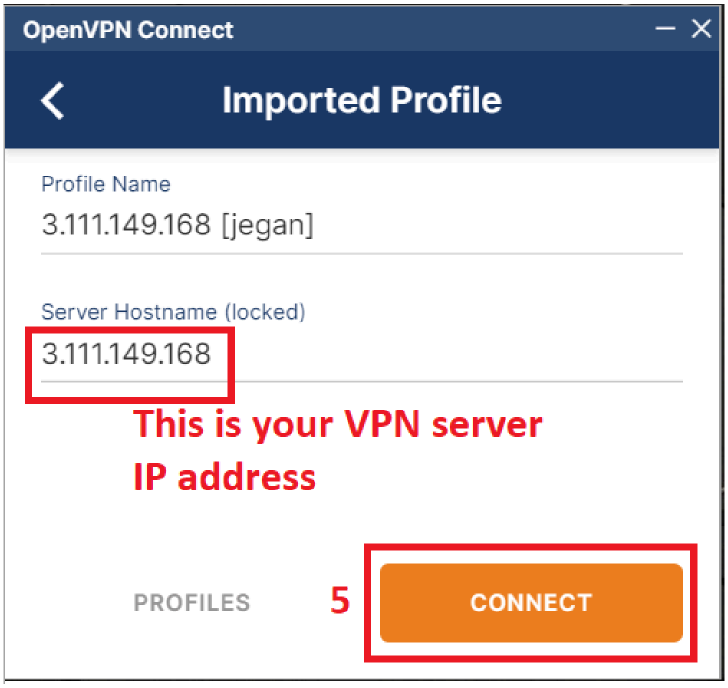


Drag and drop to upload .OVPN profile.
You can import only one profile at a time.



Name	Date	Type	Size	Tags
HR	22-02-2022 18:30	File folder		
desktop1-feb 1	31-01-2023 17:34	File folder		
jegan	02-02-2023 11:18	OVPN Profile	3 KB	

File name: jegan



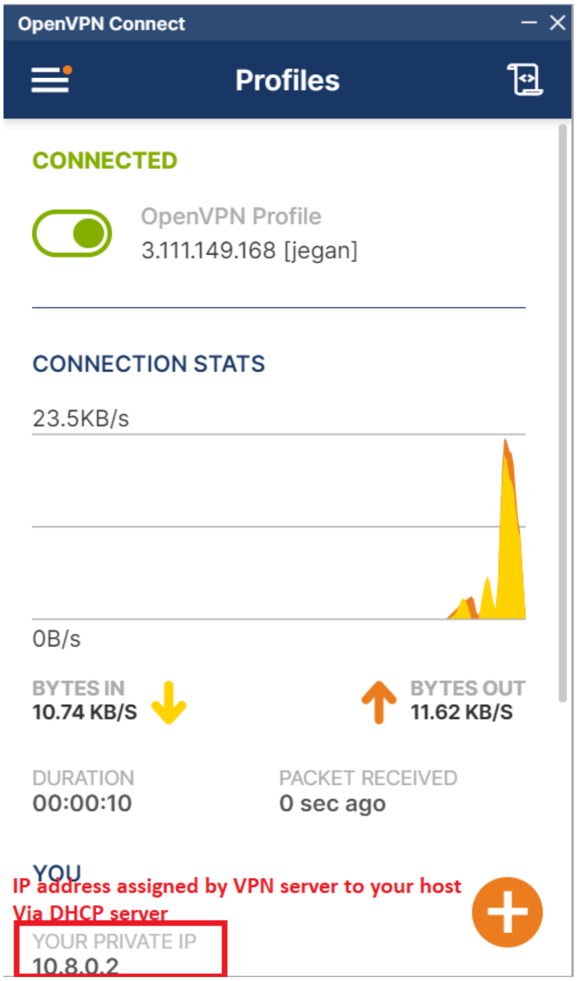
OpenVPN Connect

Imported Profile

Profile Name
3.111.149.168 [jegan]

Server Hostname (locked)
3.111.149.168

This is your VPN server IP address



OpenVPN Connect

Profiles

CONNECTED

OpenVPN Profile
3.111.149.168 [jegan]

CONNECTION STATS

BYTES IN
10.74 KB/S

BYTES OUT
11.62 KB/S

DURATION
00:00:10

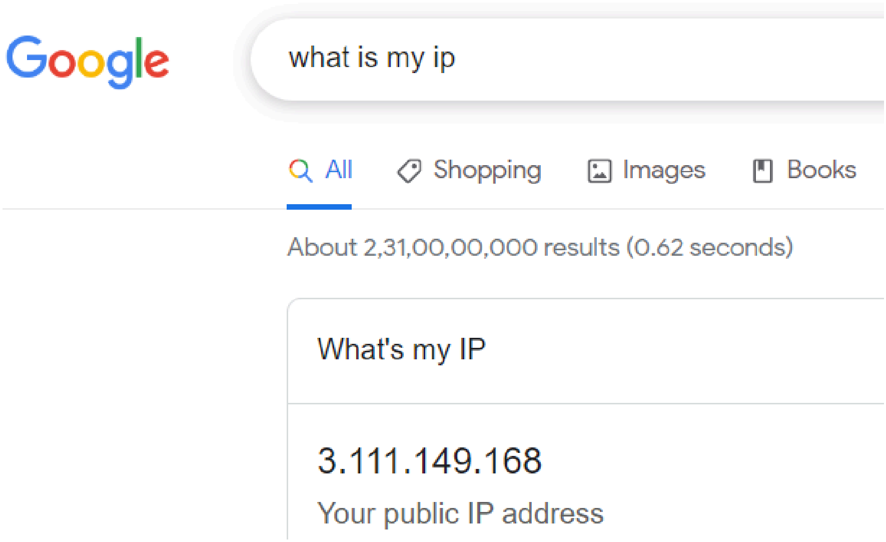
PACKET RECEIVED
0 sec ago

YOU

IP address assigned by VPN server to your host
Via DHCP server

YOUR PRIVATE IP
10.8.0.2

Now browse google.com and find your IP address, google should provide AWS Instance public IP address as shown below,



About 2,31,00,00,000 results (0.62 seconds)

What's my IP

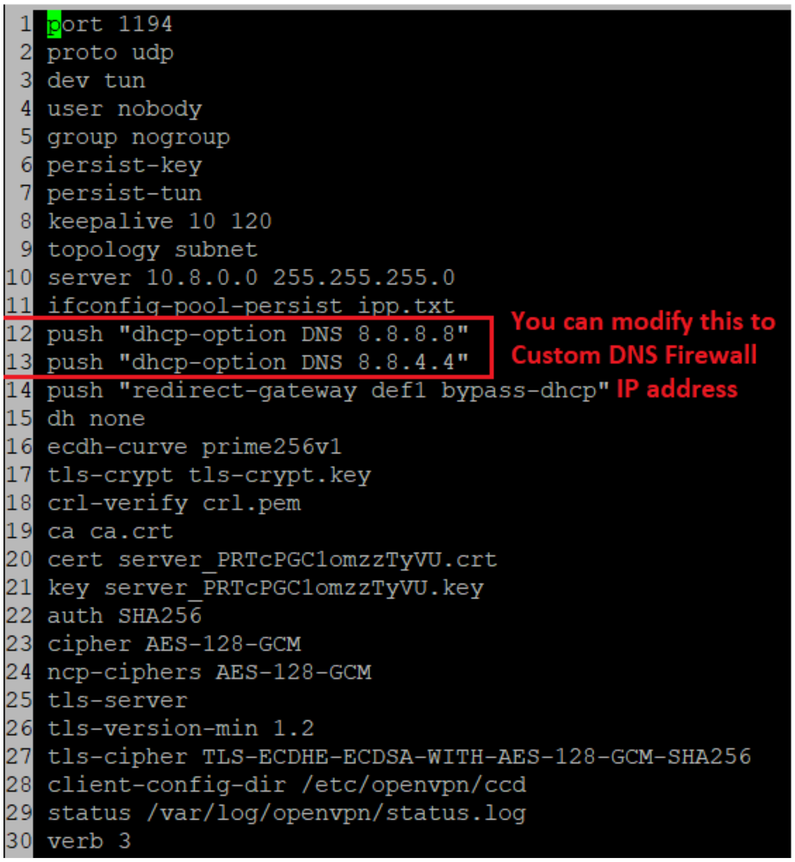
3.111.149.168

Your public IP address

Now your connection is private and secure , now we need to make the internet traffic to your host secure.

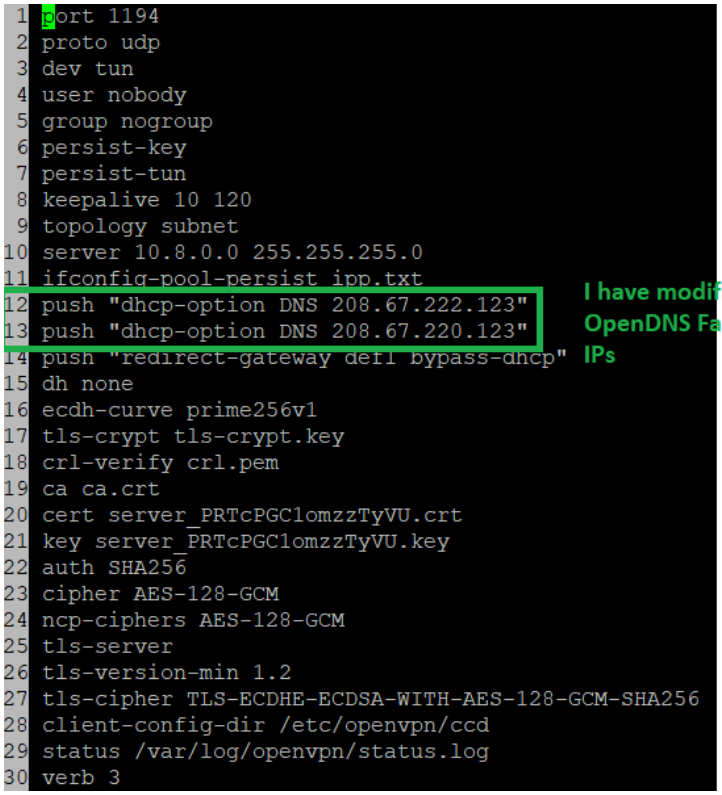
Now we will integrate DNS Firewall or DNS Sinkhole IP address to VPN server, so that when you reset your VPN client, your internet traffic will flow through DNS Firewall IP and thereby keeping your host internet traffic clean through the VPN tunnel.

Now login to your server console and edit the file /etc/openvpn/server.conf so that you configure the DNS Firewall IP address from OpenDNS Family Shield.



```
port 1194
proto udp
dev tun
user nobody
group nogroup
persist-key
persist-tun
keepalive 10 120
topology subnet
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"
push "redirect-gateway def1 bypass-dhcp"
dh none
ecdh-curve prime256v1
tls-crypt tls-crypt.key
crl-verify crl.pem
ca ca.crt
cert server_PRTcPGC1omzzTyVU.crt
key server_PRTcPGC1omzzTyVU.key
auth SHA256
cipher AES-128-GCM
ncp-ciphers AES-128-GCM
tls-server
tls-version-min 1.2
tls-cipher TLS-ECDHE-ECDSA-WITH-AES-128-GCM-SHA256
client-config-dir /etc/openvpn/ccd
status /var/log/openvpn/status.log
verb 3
```

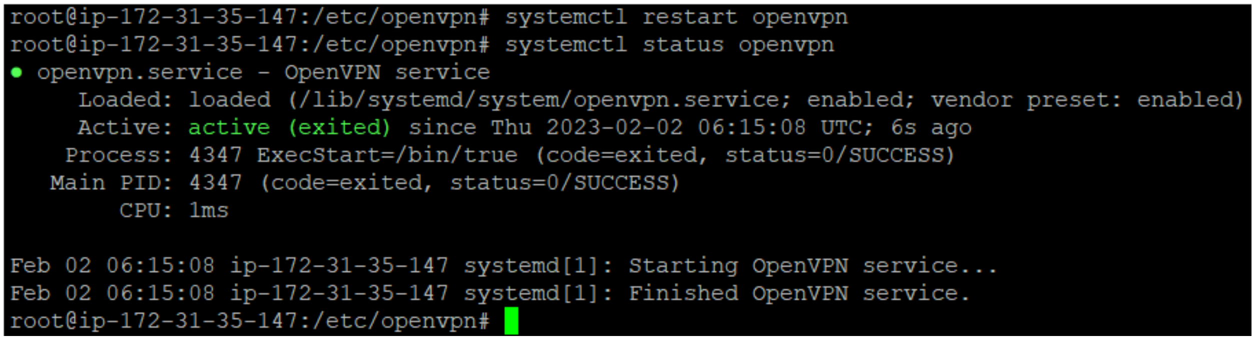
You can modify this to Custom DNS Firewall IP address



```
port 1194
proto udp
dev tun
user nobody
group nogroup
persist-key
persist-tun
keepalive 10 120
topology subnet
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "dhcp-option DNS 208.67.222.123"
push "dhcp-option DNS 208.67.220.123"
push "redirect-gateway def1 bypass-dhcp"
dh none
ecdh-curve prime256v1
tls-crypt tls-crypt.key
crl-verify crl.pem
ca ca.crt
cert server_PRTcPGC1omzzTyVU.crt
key server_PRTcPGC1omzzTyVU.key
auth SHA256
cipher AES-128-GCM
ncp-ciphers AES-128-GCM
tls-server
tls-version-min 1.2
tls-cipher TLS-ECDHE-ECDSA-WITH-AES-128-GCM-SHA256
client-config-dir /etc/openvpn/ccd
status /var/log/openvpn/status.log
verb 3
```

I have modified this to OpenDNS Family Shield DNS Firewall IPs

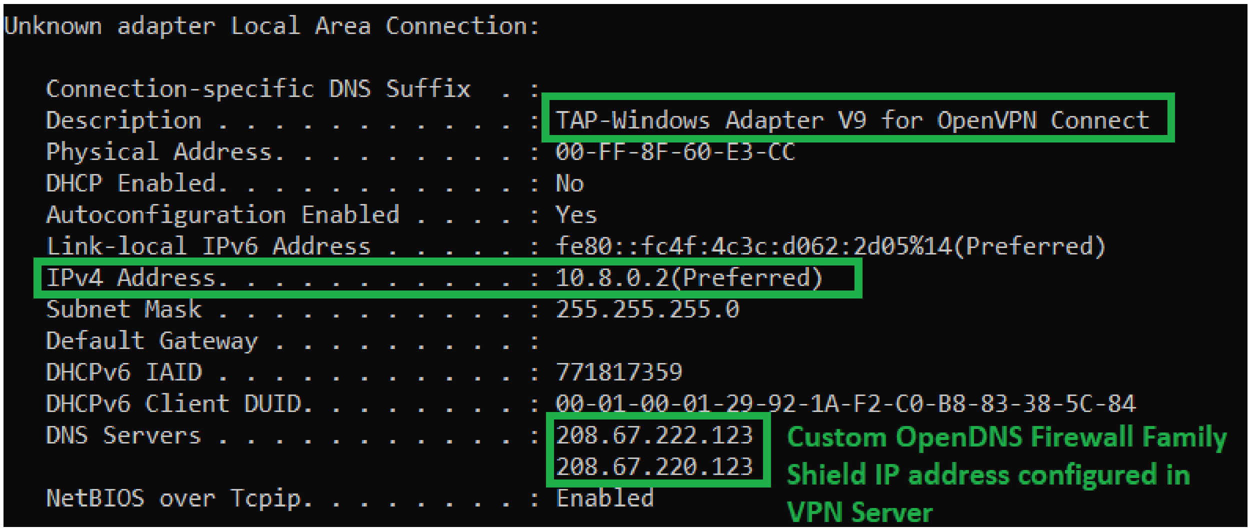
Now restart OpenVPN service in VPN Server,



```
root@ip-172-31-35-147:/etc/openvpn# systemctl restart openvpn
root@ip-172-31-35-147:/etc/openvpn# systemctl status openvpn
● openvpn.service - OpenVPN service
     Loaded: loaded (/lib/systemd/system/openvpn.service; enabled; vendor preset: enabled)
     Active: active (exited) since Thu 2023-02-02 06:15:08 UTC; 6s ago
    Process: 4347 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 4347 (code=exited, status=0/SUCCESS)
        CPU: 1ms

Feb 02 06:15:08 ip-172-31-35-147 systemd[1]: Starting OpenVPN service...
Feb 02 06:15:08 ip-172-31-35-147 systemd[1]: Finished OpenVPN service.
root@ip-172-31-35-147:/etc/openvpn#
```

Reset VPN Client in your host by turning off and on switch.

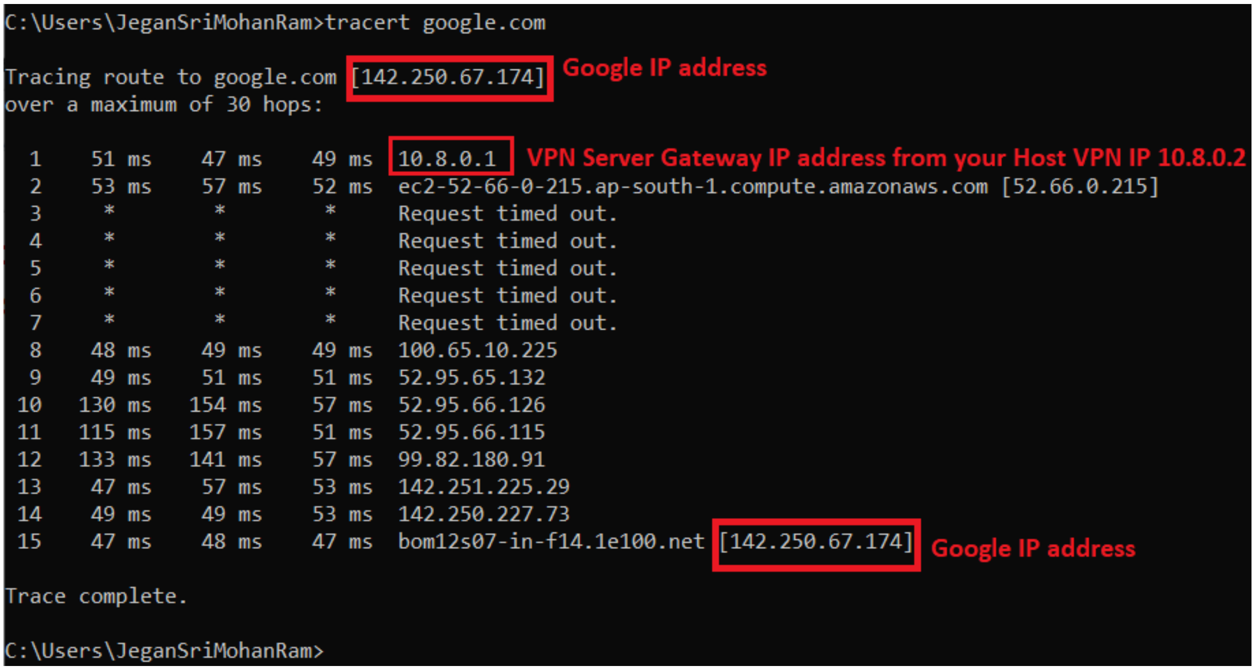
Now validate your host OpenVPN adapter with ipconfig /all in your windows host command prompt,



```
Unknown adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : TAP-Windows Adapter V9 for OpenVPN Connect
   Physical Address. . . . . . . . . : 00-FF-8F-60-E3-CC
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::fc4f:4c3c:d062:2d05%14(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.8.0.2(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 771817359
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-29-92-1A-F2-C0-B8-83-38-5C-84
   DNS Servers . . . . . . . . . . . : 208.67.222.123
                                       208.67.220.123
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Custom OpenDNS Firewall Family Shield IP address configured in VPN Server

Now check the traffic path to ensure the internet traffic flows through VPN Server from your windows host.Use tracert google.com from windows command prompt.



```
C:\Users\JeganSriMohanRam>tracert google.com

Tracing route to google.com [142.250.67.174]
over a maximum of 30 hops:

  1    51 ms    47 ms    49 ms  10.8.0.1
  2    53 ms    57 ms    52 ms  ec2-52-66-0-215.ap-south-1.compute.amazonaws.com [52.66.0.215]
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8    48 ms    49 ms    49 ms  100.65.10.225
  9    49 ms    51 ms    51 ms  52.95.65.132
 10   130 ms   154 ms    57 ms  52.95.66.126
 11   115 ms   157 ms    51 ms  52.95.66.115
 12   133 ms   141 ms    57 ms  99.82.180.91
 13    47 ms    57 ms    53 ms  142.251.225.29
 14    49 ms    49 ms    53 ms  142.250.227.73
 15    47 ms    48 ms    47 ms  bom12s07-in-f14.1e100.net [142.250.67.174]

Trace complete.

C:\Users\JeganSriMohanRam>
```

Google IP address

VPN Server Gateway IP address from your Host VPN IP 10.8.0.2

Google IP address

Done!!!