

Procedure for installing VPN connection between HF Lab and telecommunication Lab

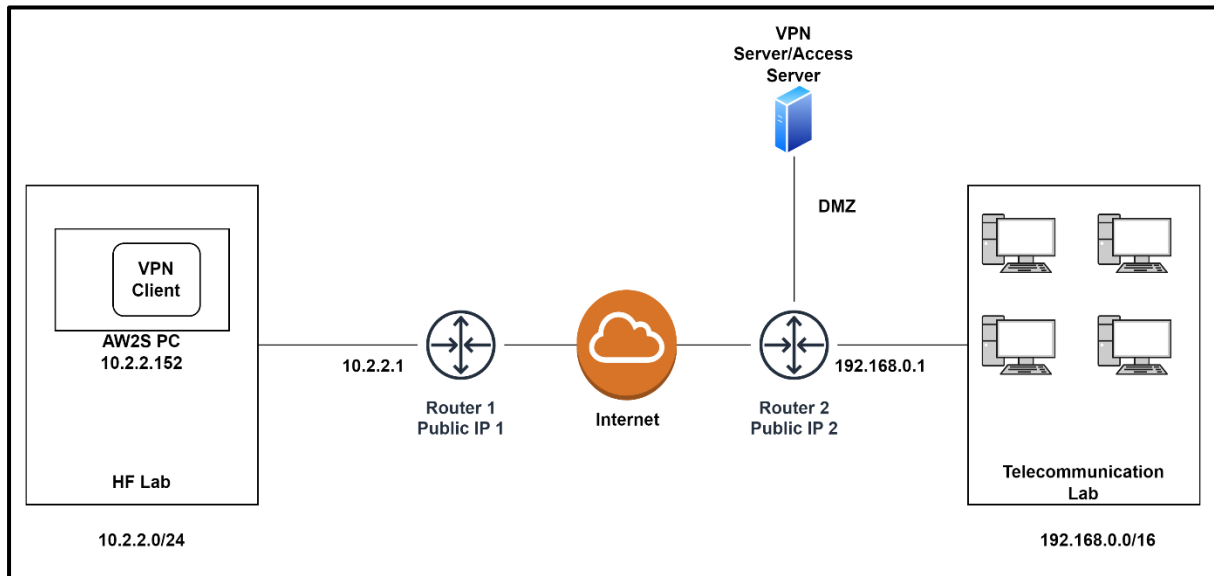


Figure 1: Architecture Overview to set up the VPN connection.

For installing the site-to-site VPN, I want to use OpenVPN. The installation procedure is referenced from the documentation “Site-to-site VPN routing with Access Server” from the official website of OpenVPN available at <https://openvpn.net/vpn-server-resources/site-to-site-routing-explained-in-detail/#a-simple-site-to-site-vpn-setup>.

Network1 (HF Lab): 10.2.2.0/24 → Branch Network

Network2 (Telecommunication Lab): 192.168.0.0/16 → Headquarter Network

Headquarter Network contains the VPN server or access server whereas the branch network contains the site-to-site connector or VPN client.

Table 1: IP Address Details for Access Server and Client

Device	IP address
Access Server (OpenVPN server)	192.168.abc.xyz
Site-to-site connector (OpenVPN client)	10.2.2.152

Table 2: Network Subnet Details

Network	Subnet
Headquarters (Telecommunication Lab)	192.168.0.0/16
OpenVPN virtual network	172.16.0.0/24
Branch office (HF Lab)	10.2.2.0/24

In a site-to-site VPN, devices in one network can reach devices in the other network and vice versa.

Steps taken for VPN installation:

- Install an Access Server (VPN server) on a headquarters (HQ) network server.
- Connect the Access Server to the same router as the other devices and servers in the HQ network.
- Provide internet access through the router to the HQ network.
- At a branch office, connect the network to the internet through a router.
- Install the OpenVPN client software (VPN client) on a Linux server on the branch network.
- Connect the OpenVPN client to the Access Server (VPN tunnel) to start an active tunnel for secure data communication.
- Allow traffic between the networks through each network's routers, firewalls, or internet gateways.

Set up the OpenVPN Access Server in the Headquarter network.

1. Installing an Access Server on the HQ network

We need to create a free account on the access server portal of OpenVPN to install a self-hosted VPN solution i.e., Access Server. Then, we can sign in to the Access Server portal.

Ensure we have the following for our Linux system:

- Root-level access to be supported on Linux OS.
- The correct date and time for certificate generation and verification, and properly handling TOTP for 2FA.
- Internet access.

- Forwarding through our firewall for ports **TCP 443, TCP 943, TCP 945, and UDP 1194.**
- Proper DNS resolution.

To install the Access Server, use the official repository. Log in to our Linux system with root privileges and enter the commands as mentioned in Listing 1 to add the repository and install the package 'openvpn-as'. These steps are used for Ubuntu 22, x86_64 version. For other versions, steps can be found at <https://as-portal.openvpn.com/instructions/ubuntu/installation>.

```
#apt update && apt -y install ca-certificates wget net-tools gnupg

#wget https://as-repository.openvpn.net/as-repo-public.asc -qO /etc/apt/trusted.gpg.d/as-repository.asc

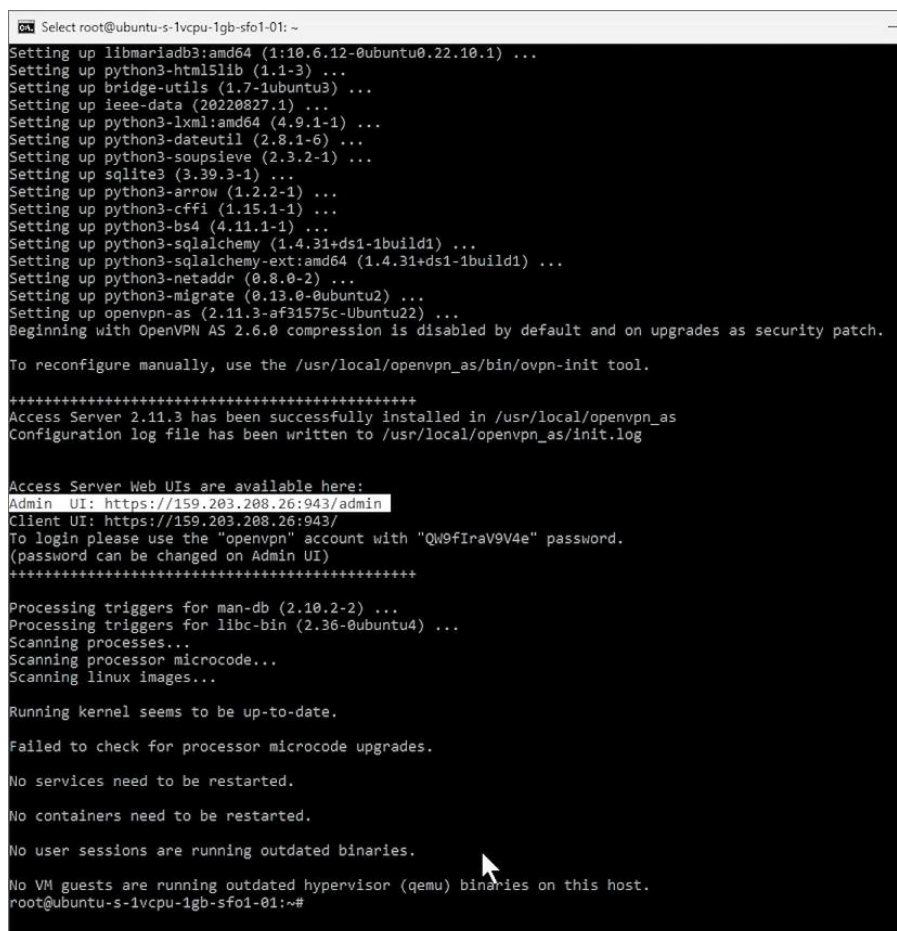
#echo "deb [arch=amd64 signed-by=/etc/apt/trusted.gpg.d/as-repository.asc] http://as-repository.openvpn.net/as/debian jammy main">/etc/apt/sources.list.d/openvpn-as-repo.list

#apt update && apt -y install openvpn-as
```

Listing 1: CLI commands to install OpenVPN server on Ubuntu 22, x86_64

Reference Video URL for installing Access Server: <https://vimeo.com/806106858>

After installing, we'll receive the URLs for your admin and client UIs along with a randomly generated admin username and password. The Screenshot shown in Figure 2 for the reference.



```
Select root@ubuntu-s-1vcpu-1gb-sfo1-01: ~
Setting up libmariadb3:amd64 (1:10.6.12-0ubuntu0.22.10.1) ...
Setting up python3-html5lib (1.1-3) ...
Setting up bridge-utils (1.7-1ubuntu3) ...
Setting up ieee-data (20220827.1) ...
Setting up python3-lxml:amd64 (4.9.1-1) ...
Setting up python3-dateutil (2.8.1-6) ...
Setting up python3-soupsieve (2.3.2-1) ...
Setting up sqlite3 (3.39.3-1) ...
Setting up python3-arrow (1.2.2-1) ...
Setting up python3-cffi (1.15.1-1) ...
Setting up python3-bs4 (4.11.1-1) ...
Setting up python3-sqlalchemy (1.4.31+ds1-1build1) ...
Setting up python3-sqlalchemy-ext:amd64 (1.4.31+ds1-1build1) ...
Setting up python3-netaddr (0.8.0-2) ...
Setting up python3-migrate (0.13.0-0ubuntu2) ...
Setting up openvpn-as (2.11.3-af31575c-Ubuntu22) ...
Beginning with OpenVPN AS 2.6.0 compression is disabled by default and on upgrades as security patch.

To reconfigure manually, use the /usr/local/openvpn_as/bin/ovpn-init tool.

*****
Access Server 2.11.3 has been successfully installed in /usr/local/openvpn_as
Configuration log file has been written to /usr/local/openvpn_as/init.log

Access Server Web UIs are available here:
Admin UI: https://159.203.208.26:943/admin
Client UI: https://159.203.208.26:943/
To login please use the "openvpn" account with "QW9fIraV9V4e" password.
(password can be changed on Admin UI)
*****

Processing triggers for man-db (2.10.2-2) ...
Processing triggers for libc-bin (2.36-0ubuntu4) ...
Scanning processes...
Scanning processor microcode...
Scanning linux images...

Running kernel seems to be up-to-date.

Failed to check for processor microcode upgrades.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ubuntu-s-1vcpu-1gb-sfo1-01:~#
```

Figure 2: Output of the commands of Listing 1

Now login to the OpenVPN admin UI of the access server from the URL highlighted in Figure 2, which we got after executing the commands from listing 1. The screenshot for the OpenVPN admin UI is shown in Figure 3.

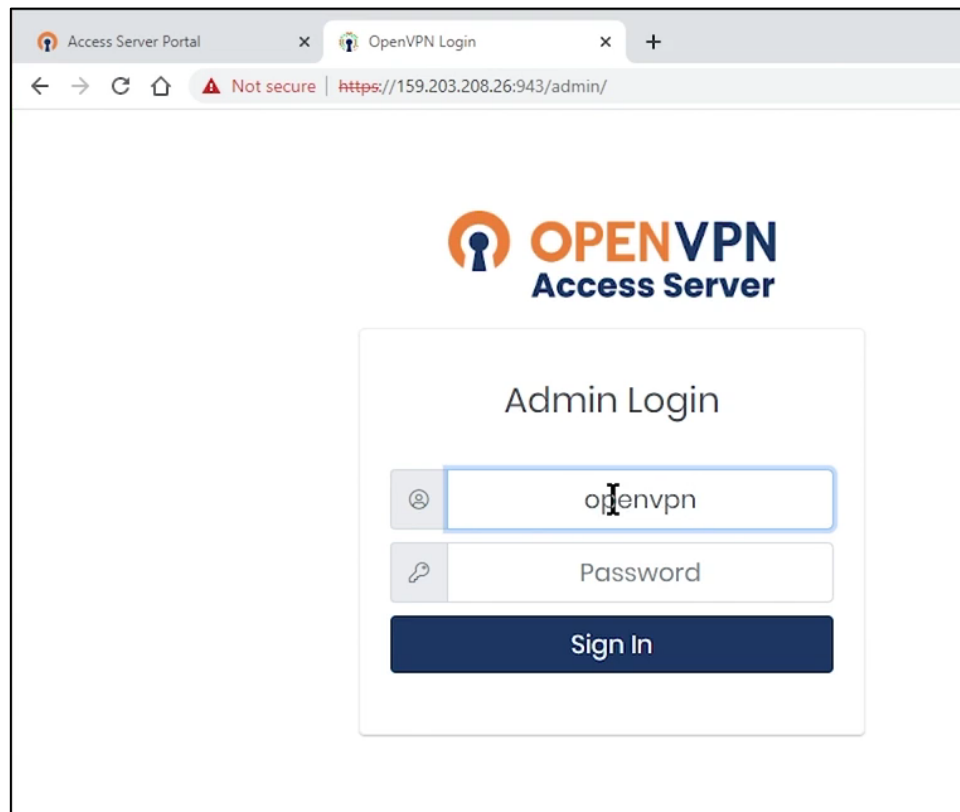


Figure 3: OpenVPN Admin GUI

Next, activate the subscription by entering the activation key as shown in Figure 4, which can be generated from the Access Server Portal of OpenVPN (Refer Figure 5). Only 2 VPN connections are allowed for the free use.

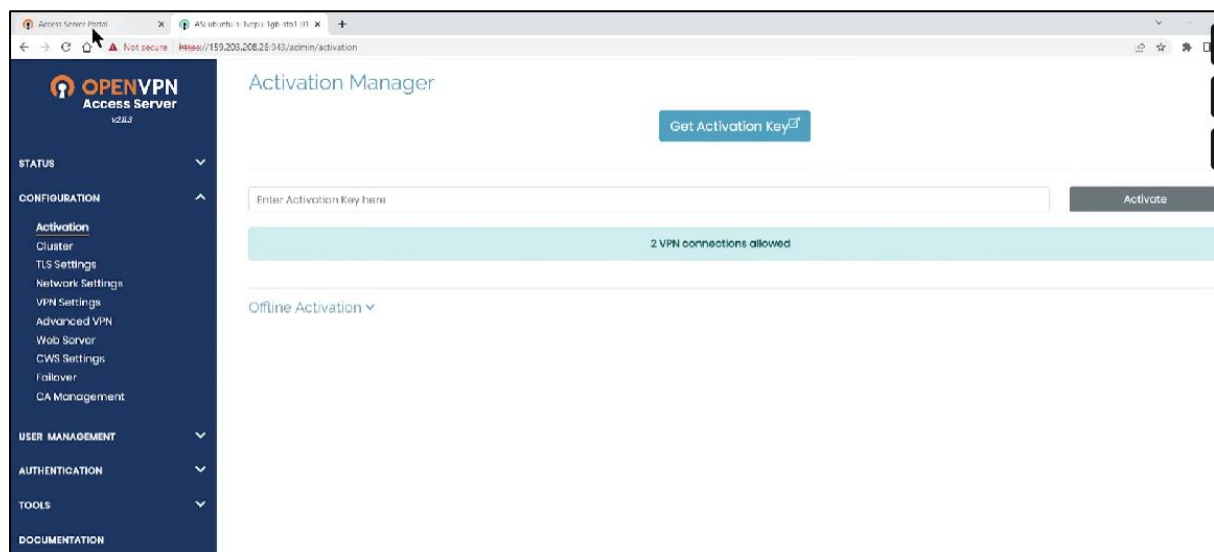


Figure 4: Activation Manager Tab in OpenVPN Admin GUI

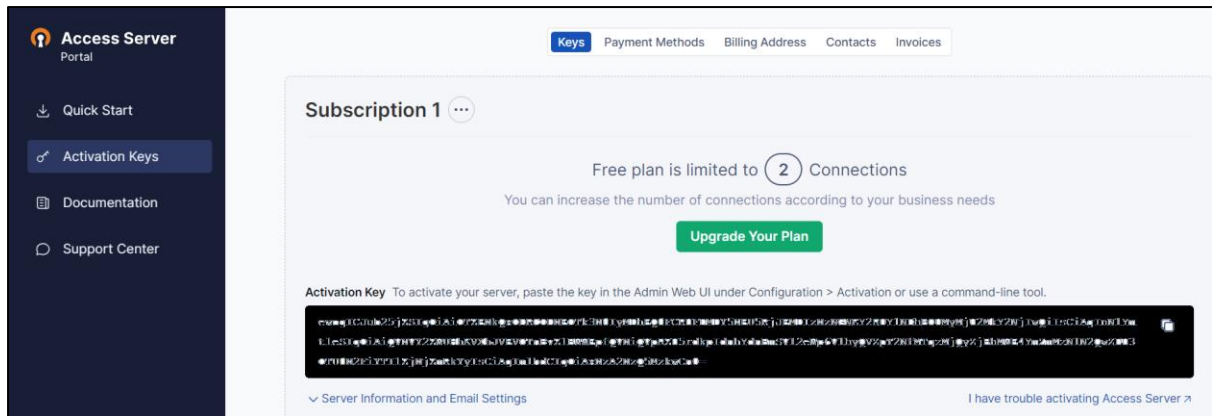


Figure 5: Activation Key from the Access Server Portal

Next, change the OpenVPN password under **USER Management → User Permissions** in Admin Web UI and try to again login with the new password.

2. Enabling Routing and VPN settings in Access Server

- Sign into the Admin Web UI of the Access Server.
- Click **Configuration > VPN Settings**.
- When a VPN Client connects to the Access Server, it is assigned a unique IP address on the virtual VPN IP network. This is managed by the “**VPN IP Network**” as shown below. We can define Dynamic, Static, or Group Default IP Address Networks. For our case, we will take a “Static IP Address Network” for the VPN client (AW2S PC), and the “Network Address” will be 172.16.0.0/24 which is our OpenVPN virtual network. I am planning to give a static tunneled IP address to the client (AW2S PC) and later in the user permission section, I will give the static IP address to the client. (Refer Figure 6)

VPN IP Network
Specify the addresses and netmasks for the virtual networks created for VPN clients

Dynamic IP Address Network
When a user does not have a specific VPN IP address configured on the **User Permissions** page, the user's VPN client is assigned an address from this network.

Network Address: # of Netmask bits:

Static IP Address Network (Optional)
Any static VPN IP addresses specified for particular users on the **User Permissions** page must be within this network

Network Address: # of Netmask bits:

Group Default IP Address Network (Optional)
When a group does not have a specific Dynamic IP Address pool setting, the dynamic IP address pool for the group will be allocated from this list of subnets.

Figure 6: VPN IP Network setting in OpenVPN Admin Web UI

- Under **Routing**, and “Should VPN clients have access to private subnets (non-public networks on the server side)?”, I will choose “Yes, using Routing”.
- Now, click “Specify the private subnets to which all clients should be given access (one per line)”. I will enter the network’s subnet where our Access Server is located i.e., 192.168.0.0/16. (Refer Figure 7)
- Click Save Settings and Update Running Server.

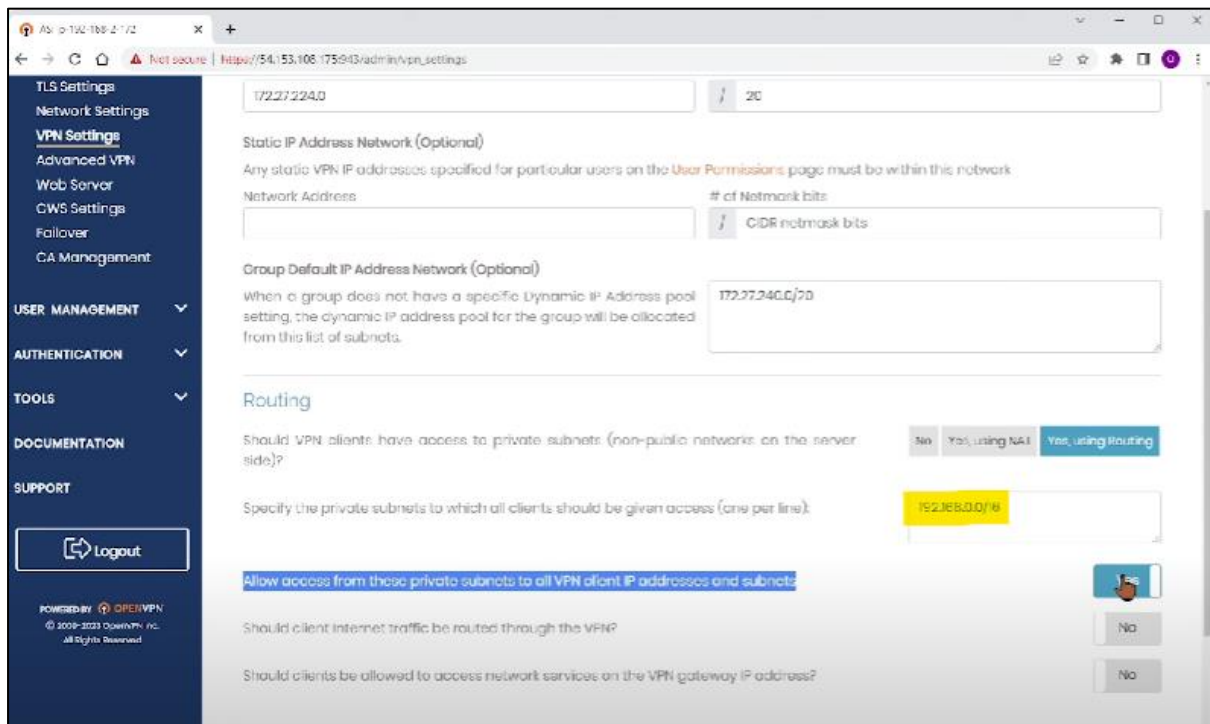


Figure 7: Routing setting in OpenVPN Admin Web UI

3. Create a client user with Auto Login for OpenVPN Client Gateway and other access control.

- Click **User Management > User Permissions**.
- Enter a new username for your OpenVPN client.
- Click **Allow Auto-login**. Screenshot shown below in which “client-gateway” user is created with allow Auto-login. (Refer Figure 8)

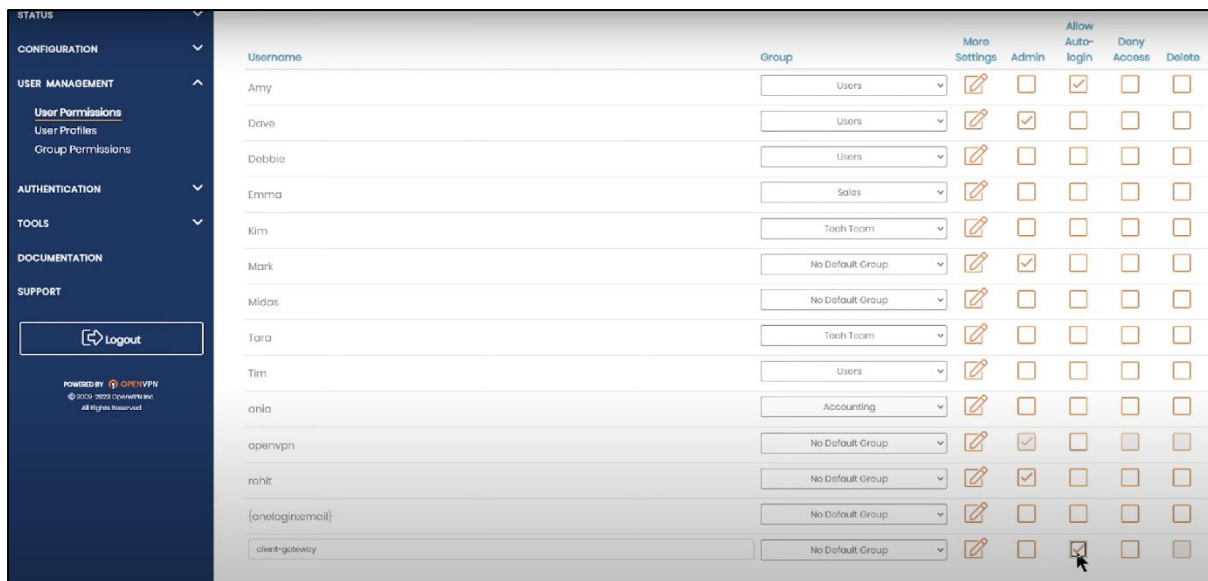


Figure 8: User “Client-gateway” created in Admin Web UI

- Click **More Settings**.
- Select **Default (Local)** for the **Auth method**. Generally, it is selected by default.
- We can manage password options for each user set to local authentication. Enter the local password in the **Password** field for authentication when connecting to the Access Server or signing into the Client Web UI. This password is for users authenticating with local authentication.
- Select **IP Addressing**. **Two options are there Dynamic or Static.**
 - **Dynamic:** Access Server dynamically assigns the user’s IP address from the subnets configured in VPN Settings.
 - **Static:** Access Server assigns the static IP address, defined in the VPN Static IP Address field that displays when we select **Use Static**. Ensure the IP address is within the subnet defined in VPN Settings.

As discussed in Section 2, I am planning to give the static IP address to the client PC i.e., our AW2S PC within the subnet 172.16.0.0/24.

- In the Access Control, NAT is chosen because for our setup NAT is used. Moreover, I will tick the option “**Allow Access From all server-side private subnets**” so that the server-side private subnets can also access the client i.e., our AW2S PC which is our main target to access it from the telecommunication Lab. (Refer Figure 9)
- Moreover, Set **Configure VPN Gateway** to **Yes**. (Refer Figure 9)
- Enter the subnet of the remote network of the OpenVPN client into the box for “**Allow client to act as VPN gateway for these client-side subnets**”. In our case, it is 10.2.2.0/24. (Refer Figure 9)
- Click **Save Settings** and **Update Running Server**.

Require MFA: ☒ Default (disabled) ☐ Enabled ☐ Disabled

Local Password

Password:

Allow password change from CWS: ☒ Default ☐ Yes ☐ No

Enable password strength checking in CWS: ☒ Default ☐ Yes ☐ No

IP Addressing

Select IP Addressing: ☒ Use Dynamic ☐ Use Static

Access Control

Select addressing method: ☒ Use NAT ☐ Use Routing

Allow Access To these Networks:

Allow Access From: ☐ all server-side private subnets

Allow Access From: ☐ all other VPN clients

VPN Gateway

Configure VPN Gateway: ☐ No ☒ Yes

Allow client to act as VPN gateway for these client-side subnets:

DMZ settings

Configure DMZ IP address: ☒ No ☐ Yes

Figure 9: More setting tab for client-gateway user in Admin Web UI

- Now download the client user profile for the client user (i.e., client-gateway) which we create. For this example, “client-gateway” user is created. Hence, we have to download the profile for this user by going into **User Management > User Profile > New Profile (for Client-gateway user) > Autologin > Create Profile**. A file i.e., a client configuration file is downloaded with “.ovpn” extension which we will use to configure our OpenVPN client as described in section 5. (Refer Figure 10)

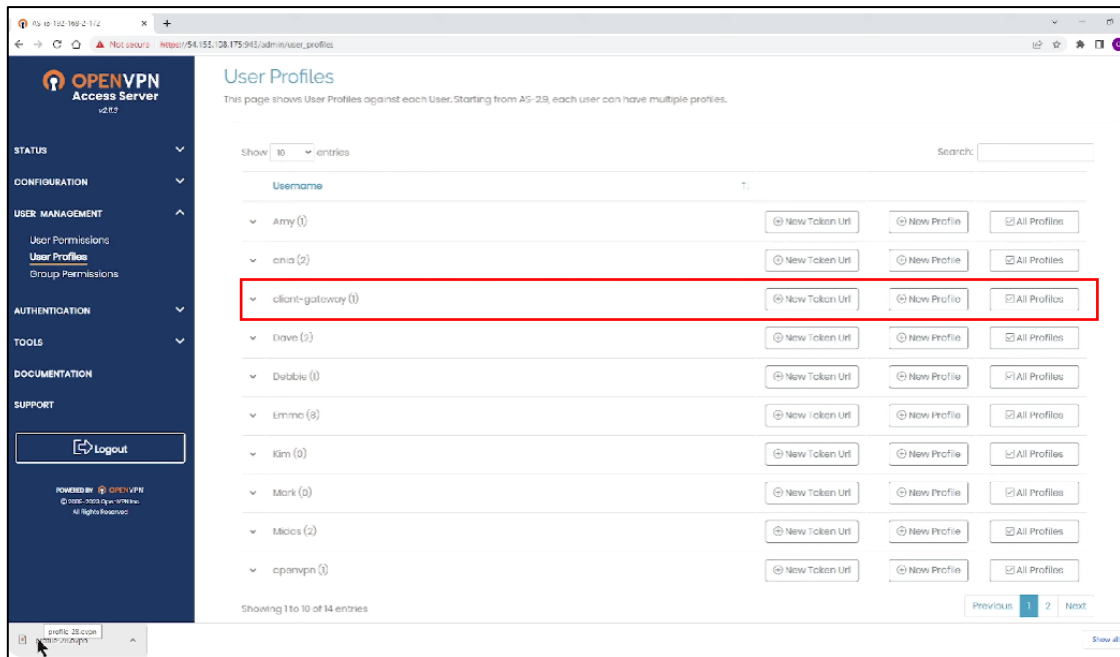


Figure 10: User Profile Section in Admin Web UI

4. To check Openvpn service status and VPN tunnel interface in Access Server

Next, confirm that the **OpenVPN** service is up and running by checking its status using the following systemctl command.

```
$ sudo systemctl status openvpn
```

Moreover, if “ip a” is executed in the access server, a new interface has been created for a VPN tunnel. In our setup, the interface will get the IP address within the subnet defined in VPN settings i.e., 172.16.0.0/24. A sample screenshot is shown in Figure 11 for reference purposes.

```
tecmint@openvpn-server:~$ ip ad
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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:90:63:46 brd ff:ff:ff:ff:ff:ff
   inet 10.42.0.24/24 brd 10.42.0.255 scope global dynamic enp0s3
       valid_lft 3288sec preferred_lft 3288sec
   inet6 fe80::a00:27ff:fe90:6346/64 scope link
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen :
   link/none
   inet 10.8.0.1/24 brd 10.8.0.255 scope global tun0
       valid_lft forever preferred_lft forever
   inet6 fe80::109a:492d:9153:d0b6/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
tecmint@openvpn-server:~$
```

Figure 11: “ip a” command output in Access Server CLI

5. To enable Firewall Ports at Telecommunication Lab Gateway Router

For the OpenVPN, forwarding for ports TCP 443, TCP 943, TCP 945, and UDP 1194 must be allowed.

Moreover, the HQ network router i.e., Router 2 needs to know:

- There are two additional subnets for the OpenVPN client network and the branch network. (Refer Table 2)
- These subnets are accessed by contacting Access Server's private IP address.

Configure the route table to include the routes to these additional subnets through Access Server. In our case, we can add the following static routes:

- Network 172.16.0.0 with subnet mask 255.255.255.0 through gateway 192.168.abc.xyz.
- Network 10.2.2.0 with subnet mask 255.255.255.0 through gateway 192.168.abc.xyz.

Set up the OpenVPN client in the branch network.

- Install the open-source OpenVPN client on an Ubuntu OS by running the following command with root privileges:

```
#sudo apt-get install OpenVPN
```

- After installing the above packages, start the OpenVPN service, for now, enable it to automatically start at system boot and check its status to confirm that it's up and running.

```
$ sudo systemctl start openvpn
```

```
$ sudo systemctl enable openvpn
```

```
$ sudo systemctl status openvpn
```

- Now, the “.ovpn” client file which we downloaded in the Access Server as discussed in Section 3 is put in this client to the /etc/openvpn/ directory.
- The next step is to enable IPv4 forwarding by uncommenting the below line in the /etc/sysctl.conf file.

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

- Now reboot the client machine and when the machine comes up, a virtual interface is created in the client machine. We can check by using “ip a” command.

If everything is set up correctly, then our client should have connected automatically to our OpenVPN access server. In the **Status > Status Overview**, it will show Current Active user is 1. (Refer 12).

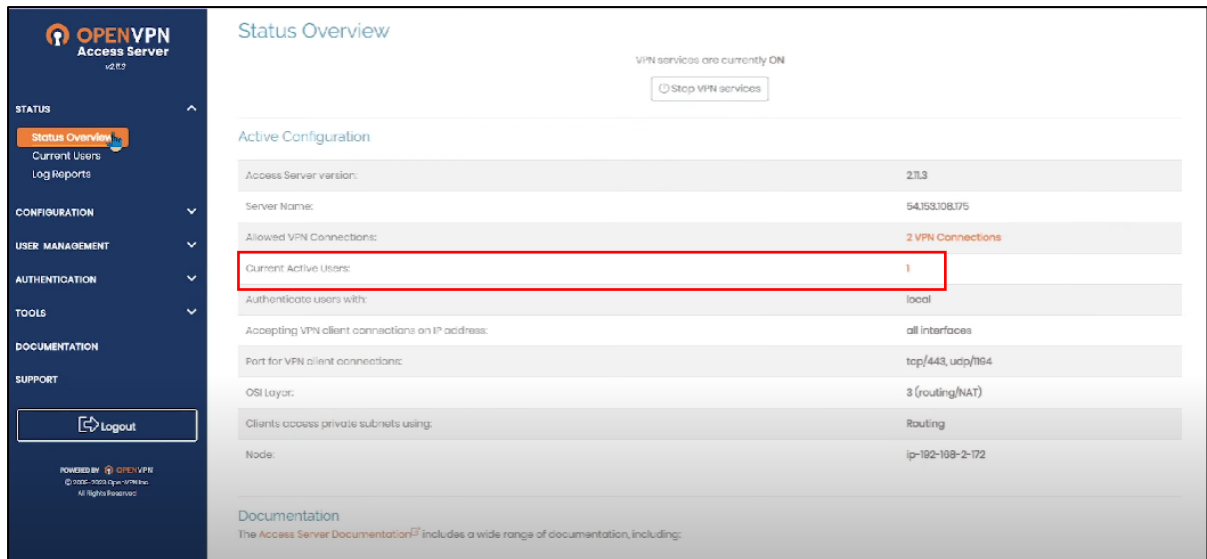


Figure 12: Status Overview in Admin Web UI

In the **Status > Current Users**, it will show the details of current Active user.

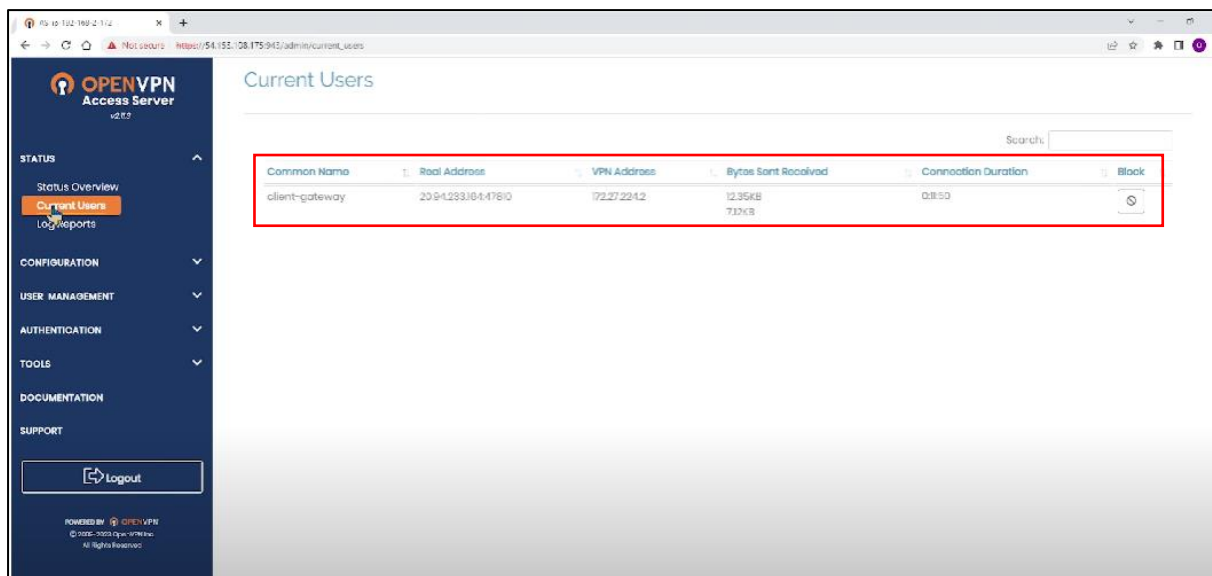


Figure 13: Current Users info in Admin Web UI

Now try to ping the access server from the client machine. If everything is set up correctly, then the ping should happen.