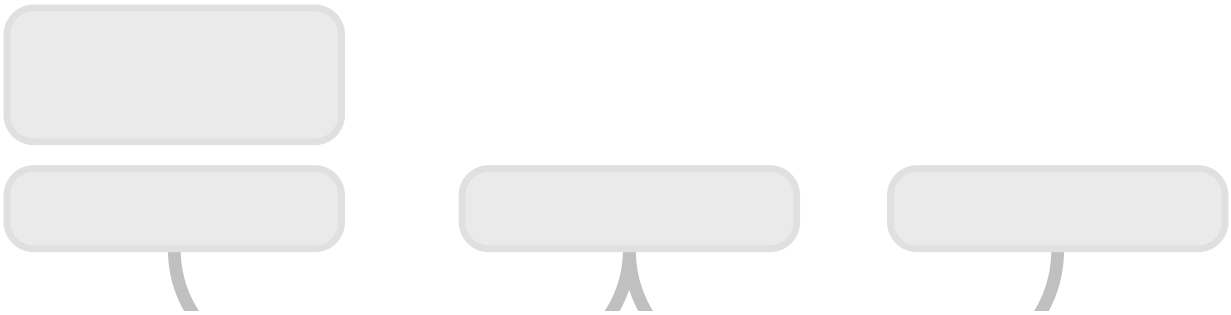
3 machines(CentOS 9):

1. VPN Server: Two Network Cards : 1st NAT(DHCP-automatic), 2nd Host Only(Manually set to 172.16.10.1, DNS: Automatic, gateway: not any)
2. VPN Client1: NAT connected with Main VPN(DHCP-automatic)
3. LAN Client2: Host Only(Manually set to 172.16.10.10/24, Gateway: 172.16.10.1, DNS: automatic) Connected with Main VPN

After Configurations VPN client should be able to ping LAN Client

**Untitled 2**



ON Main VPN and Client1

```
yum update -y
```

ON VPN Server:

To rename: It will be useful when certificate gets created

```
hostnamectl set-hostname vpnserver
```

```
hostname vpnserver
```

ON VPN Client

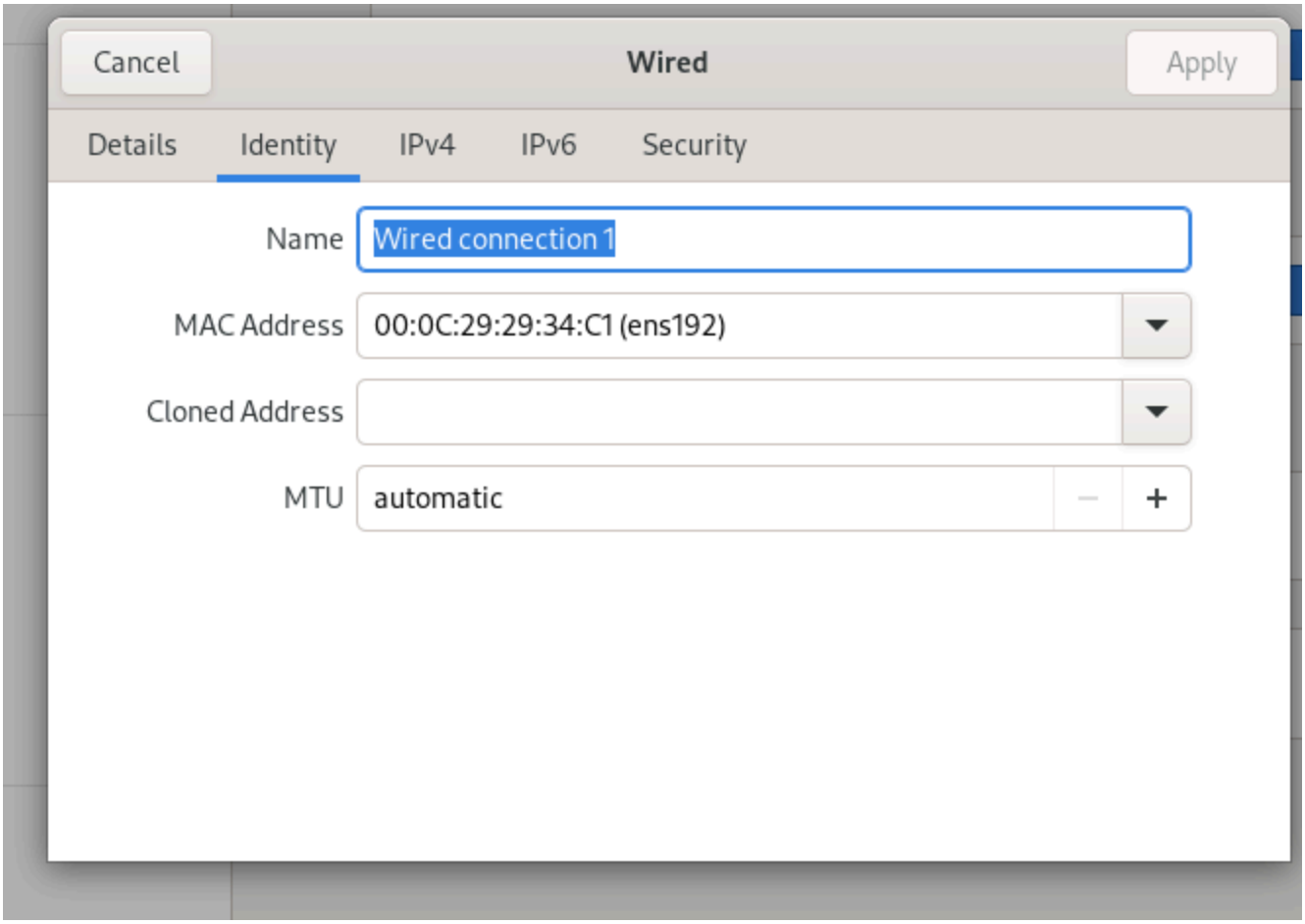
```
hostnamectl set-hostname client1
```

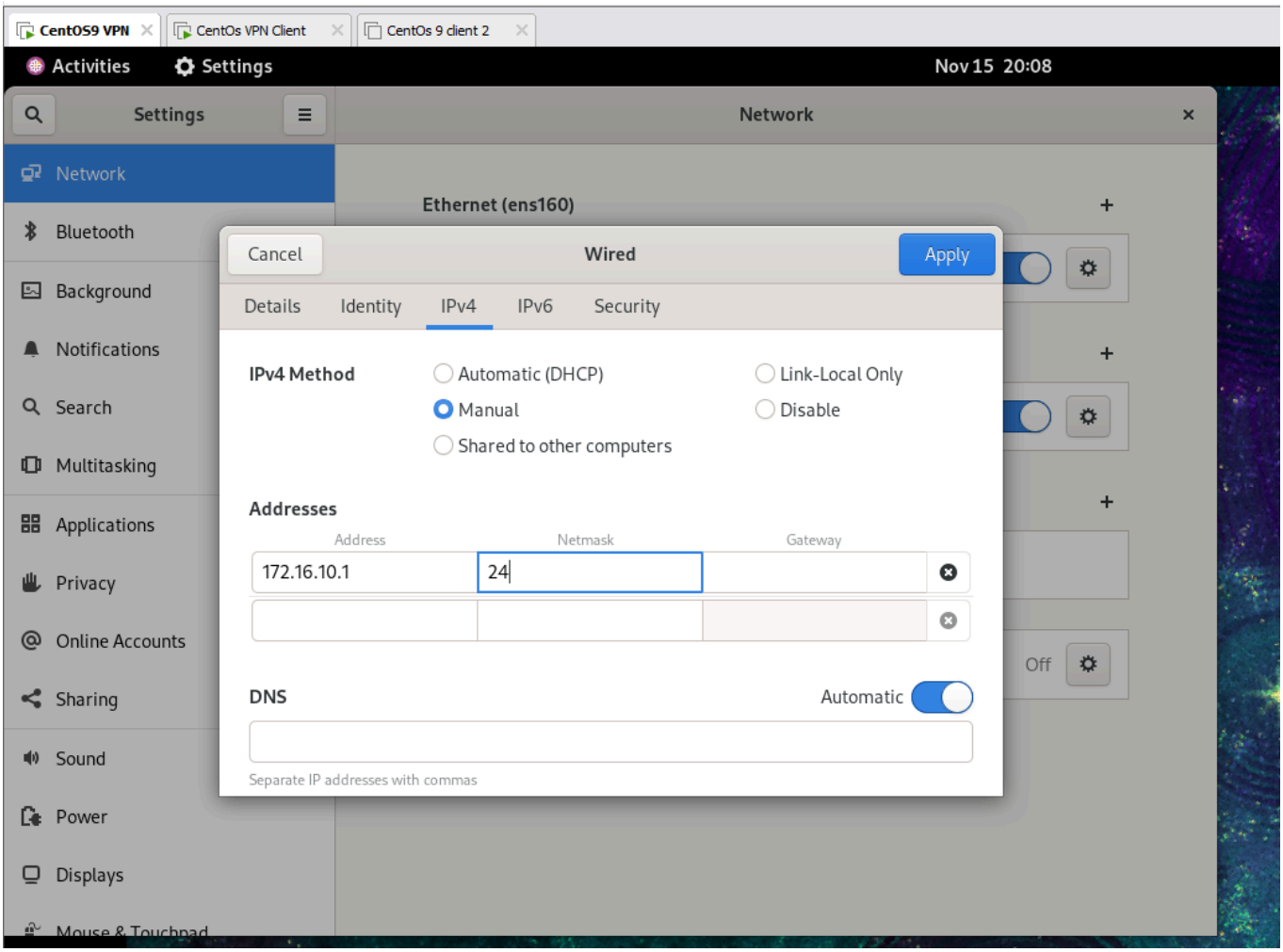
```
hostname client1
```

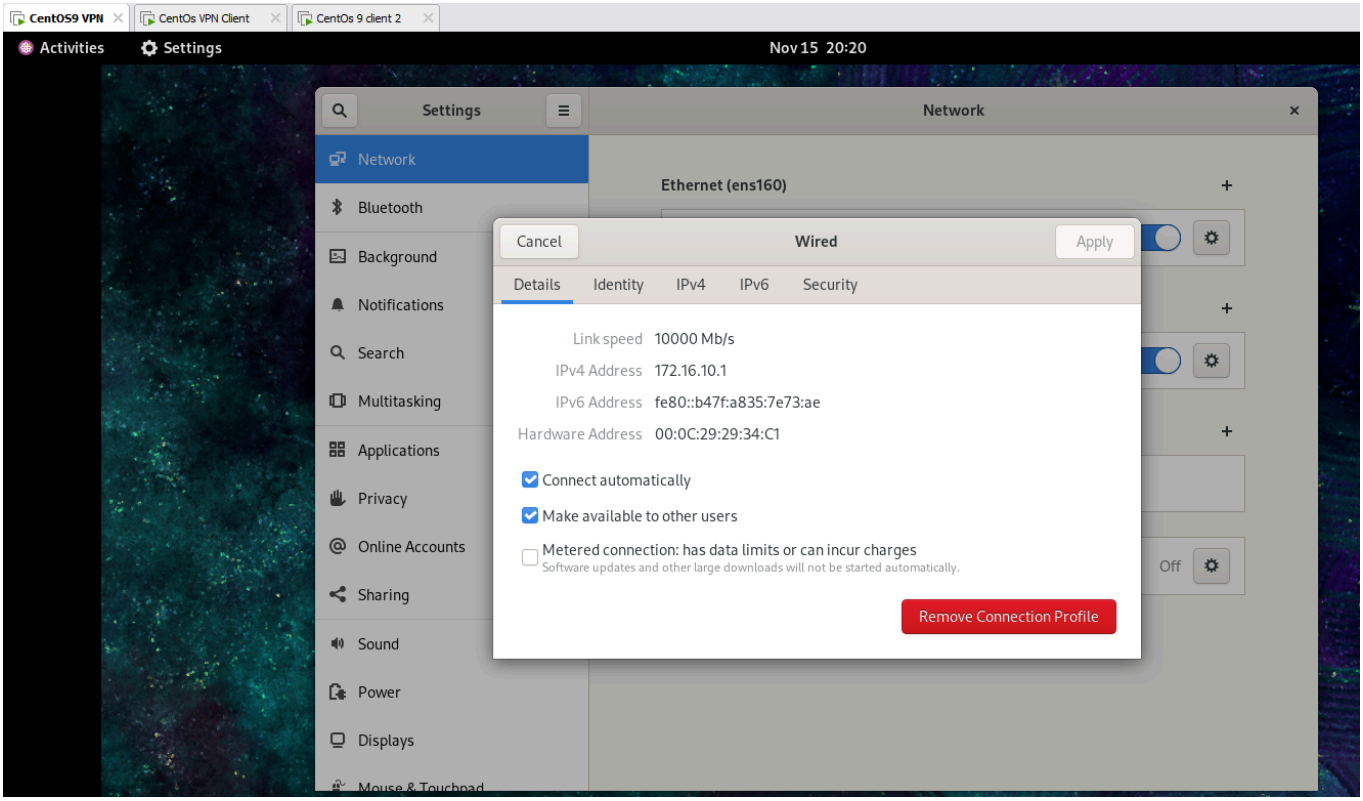
***Ping to VPN server IP (NAT)***

ON VPN Server:

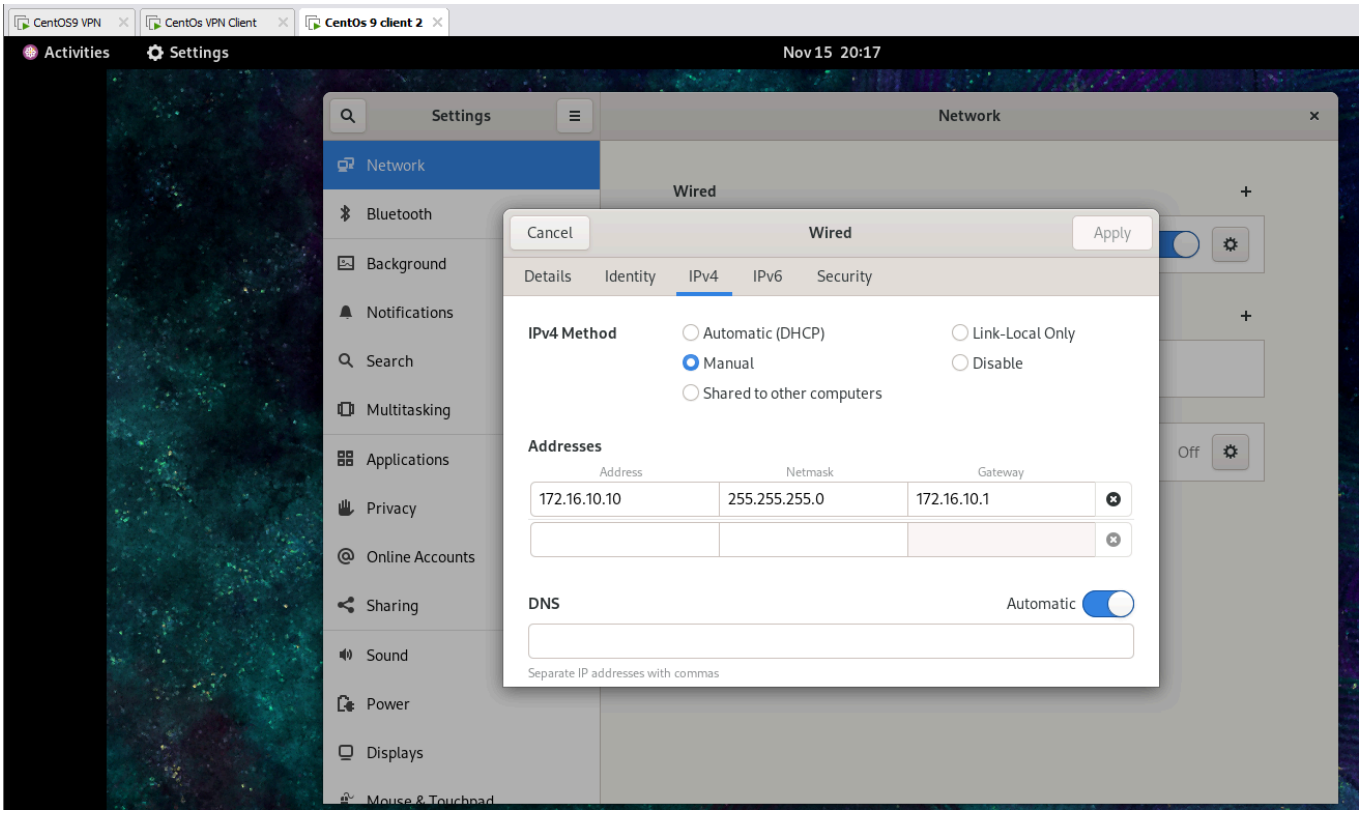
For Host Only: Select second Adapter in Identity Settings

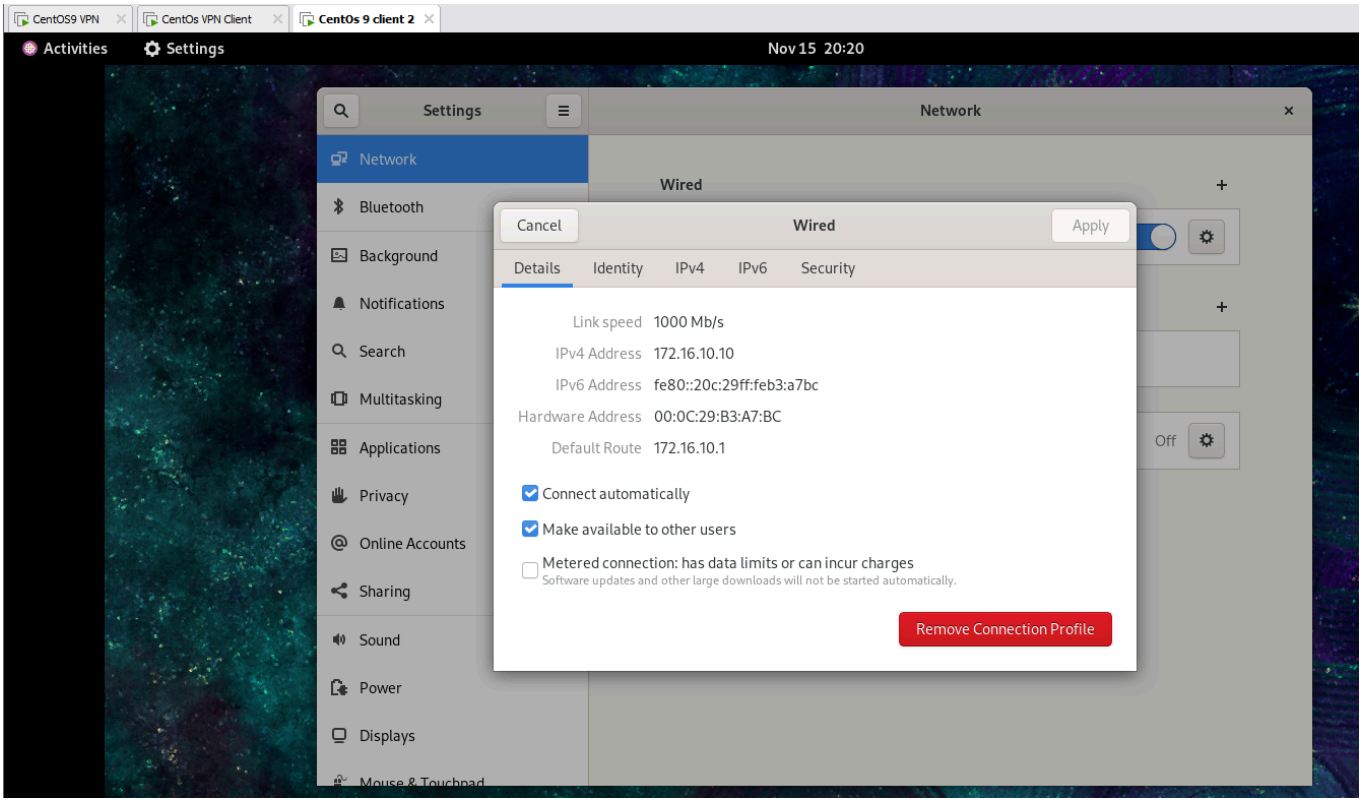





ON LAN Client2:



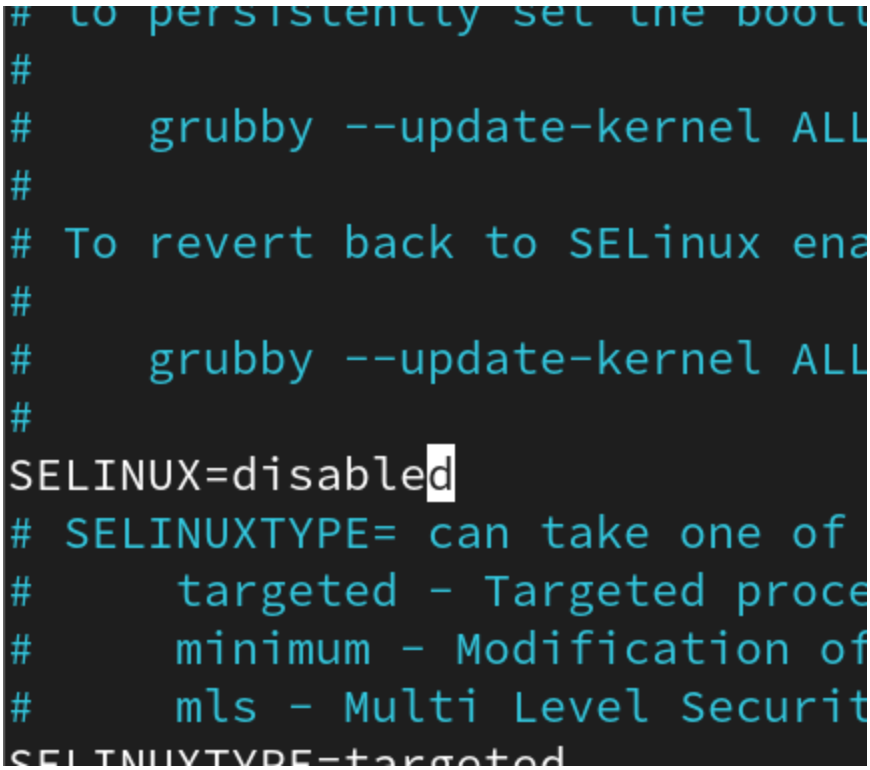




Ping Server(NAT only IP) From vpnclient

Ping Server(Host only IP) From Lan client

# ON VPN Server:

```
vi /etc/selinux/config
```




```
setenforce 0
```

To enable Ip forwarding:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

For permanent Enable:

```
vi /etc/sysctl.conf
```

```
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.ip_forward = 1
```

```
yum install epel-release -y
```

```
yum install openvpn wget tar -y
```

```
cd /etc/openvpn
```

There are client and server directories here

Now download Easy RSA Package:

```
sudo wget https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.6/EasyRSA-unix-v3.0.6.tgz
```

```
sudo tar xvzf EasyRSA-unix-v3.0.6.tgz
```

Now two New directories get created:

```
[root@vpnserver openvpn]# ls
client  EasyRSA-unix-v3.0.6.tgz  EasyRSA-v3.0.6  server
'[root@vpnserver openvpn]# '
[root@vpnserver openvpn]# mv EasyRSA-v3.0.6/ easy-rsa
[root@vpnserver openvpn]#
```

now to simplify rename file:

```
mv EasyRSA-v3.0.6/ easy-rsa
```

```
cd easy-rsa/
```

Now create variables file which will be useful while generating certificate:

```
vi vars
```

set_var EASYRSA "$PWD"
set_var EASYRSA_PKI "EASYRSA/pki"
set_var EASYRSA_DN "cn_only"
set_var EASYRSA_REQ_COUNTRY "IN"
set_var EASYRSA_REQ_PROVINCE "Maharastra"
set_var EASYRSA_REQ_CITY "Pune"
set_var EASYRSA_REQ_ORG "Demo Labs"
set_var EASYRSA_REQ_EMAIL ""
set_var EASYRSA_REQ_OU "Demo Labs CA"
set_var EASYRSA_KEY_SIZE 2048

set_var EASYRSA_ALGO rsa
set_var EASYRSA_CA_EXPIRE 7500
set_var EASYRSA_CERT_EXPIRE 365
set_var EASYRSA_NS_SUPPORT "no"
set_var EASYRSA_NS_COMMENT "Demo Labs"
set_var EASYRSA_EXT_DIR "$EASYRSA/x509-types" "set_var EASYRSA_SSL_CONF"
EASYRSA/openssl-easyrsa.cnf"
set_var EASYRSA_DIGEST "sha256"

:wq

init-pki : This is script used to initialize PKI

```
sudo ./easyrsa init-pki
```

To Build CA:

```
sudo ./easyrsa build-ca
```

Password: ditiss

Enter ditiss Again

CN: demo-ca ---> We can give any name

Common Name (eg: your user, host, or server name) [Easy-RSA CA]:demo-ca

Now generate the certificate for the vpnserver(vpnserver is the hostname here of our machine)

```
sudo ./easyrsa gen-req vpnserver nopass
```

No Name: ENTER

Now get the certificate signed from CA

```
sudo ./easyrsa sign-req server vpnserver
```

Confirmation details: yes

```
sudo ./easyrsa gen-dh
```

The server certificates are generated . Copy them to the server directory as OpenVPN server requires these files in that directory.

CA certificate i.e CA public key

```
sudo cp pki/ca.crt /etc/openvpn/server
```

VPN Server Diffie-Hellman Symmetric key:

```
sudo cp pki/dh.pem /etc/openvpn/server
```

VPN Server Private Key

```
sudo cp pki/private/vpnserver.key /etc/openvpn/server
```

VPN Server Certificate: This will be given to VPN clients

```
sudo cp pki/issued/vpnserver.crt /etc/openvpn/server
```

```
sudo ./easyrsa gen-req client1 nopass
```

Common Name (eg: your user, host, or server name) [client1]: Enter

Next get the certificate signed from the CA.

```
sudo ./easyrsa sign-req client client1
```

yes

give password: ditiss

Copy client certificates to the client directory

```
sudo cp pki/ca.crt /etc/openvpn/client/
```

```
sudo cp pki/issued/client1.crt /etc/openvpn/client
```

```
sudo cp pki/private/client1.key /etc/openvpn/client
```

# Now we will Start Configuration of OpenVPN Server

Create server.conf File:

```
sudo vi /etc/openvpn/server/server.conf
```

Add following to the file.

```
port 1194
proto udp
dev tun
ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/vpnserver.crt
key /etc/openvpn/server/vpnserver.key
dh /etc/openvpn/server/dh.pem
server 10.8.0.0 255.255.255.0
#push "redirect-gateway def1"
push "route 172.16.10.0 255.255.255.0" ### match this address to your LAN side
network address.
#push "dhcp-option DNS 208.67.222.222"
#push "dhcp-option DNS 208.67.220.220"
duplicate-cn
cipher AES-256-CBC
tls-version-min 1.2
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-CBC-
SHA256:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-128-CBC-SHA256
auth SHA512
auth-nocache
keepalive 20 60
persist-key
persist-tun
compress lz4
daemon
user nobody
group nobody
log-append /var/log/openvpn.log
verb 3
```

OpenVPN give diff network IP and not Internal LAN IP to client(so that other should not know Internal LAN IP) Like other VPN's they usually give internal LAN IP.

server 10.8.0.0 255.255.255.0: This is used to give IP's to clients after connection is established.

#push "redirect-gateway def1": Used if employees use vpn to access Internet so it will redirect to gateway
also enable push dns along with it if asked 8.8.8.8
push "route 172.16.10.0 255.255.255.0" ### match this address to your LAN side network address.

Here 172.16.10.0 is our LAN IP

Now start and enable the OpenVPN server service.

To check Errors:

```
journalctl -xe
```

OR

```
tail /var/log/messages
```

```
sudo systemctl start openvpn-server@server
```

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

```
sudo systemctl enable openvpn-server@server
```

# Create client configuration file:

```
sudo vi /etc/openvpn/client/client1.ovpn
```

Add following to the file.

```
client
dev tun
proto udp
```

```
remote 192.168.75.145 1194
ca ca.crt
cert client1.crt
key client1.key
cipher AES-256-CBC
auth SHA512
auth-nocache
tls-version-min 1.2
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-CBC-
SHA256:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-128-CBC-SHA256
resolv-retry infinite
compress lz4
nobind
persist-key
persist-tun
mute-replay-warnings
verb 3
```

Save the file.

remote 192.168.75.145 1194 : Here 192.168.75.145 Is NAT side IP of VPN Server

Add following rules to firewalld on the OpenVPN Server

```
sudo firewall-cmd --permanent --add-service=openvpn
```

```
sudo firewall-cmd --permanent --zone=trusted --add-service=openvpn
```

```
sudo firewall-cmd --permanent --zone=trusted --change-interface=tun0
```

```
sudo firewall-cmd --add-masquerade
```

```
sudo firewall-cmd --permanent --add-masquerade
```

```
sudo firewall-cmd --permanent --direct --passthrough ipv4 -t nat -A POSTROUTING -s
10.8.0.0/24 -o ens160 -j MASQUERADE
```

```
sudo firewall-cmd --reload
```

From the Server copy client configuration files to the VPN (Remote) client machine:

Server will send its certificate to client
and client will send certificate to server(For that we need to trnasfer clients certifacte we made here to vpnclient)
and they will also share key
so authentication is completed

```
scp /etc/openvpn/client/* vpnclient1@192.168.75.146:/home/vpnclient1
```

Yes
Enter Password

---

# ON VPNClient1:

```
sudo dnf install epel-release -y
```

```
sudo dnf install openvpn -y
```

Once the packages are installed copy the files copied from the VPN server as below.

```
sudo cp /home/vpnclient1/ca.crt /etc/openvpn/client
```

```
sudo cp /home/vpnclient1/client1.crt /etc/openvpn/client
```

```
sudo cp /home/vpnclient1/client1.key /etc/openvpn/client
```

Check Whether files have been Copied:

```
sudo ls /etc/openvpn/client/
```

Now start the VPN connection with command,
Client1 is file name not client1 hostname.

Give this Command From directory Where Client1 file is present

```
sudo openvpn --config client1.ovpn
```

```
-SHA256
2024-11-16 14:05:14 [vpnserver] Peer Connection Initiated with [AF_INET]192.168.75.145:1194
2024-11-16 14:05:14 PUSH: Received control message: 'PUSH_REPLY,route 172.16.10.0 255.255.255.0,rout
20,ping-restart 60,ifconfig 10.8.0.6 10.8.0.5,peer-id 0,cipher AES-256-GCM'
2024-11-16 14:05:14 OPTIONS IMPORT: timers and/or timeouts modified
2024-11-16 14:05:14 OPTIONS IMPORT: --ifconfig/up options modified
2024-11-16 14:05:14 OPTIONS IMPORT: route options modified
2024-11-16 14:05:14 OPTIONS IMPORT: peer-id set
2024-11-16 14:05:14 OPTIONS IMPORT: adjusting link_mtu to 1625
2024-11-16 14:05:14 OPTIONS IMPORT: data channel crypto options modified
2024-11-16 14:05:14 Data Channel: using negotiated cipher 'AES-256-GCM'
2024-11-16 14:05:14 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2024-11-16 14:05:14 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2024-11-16 14:05:14 net_route_v4_best_gw query: dst 0.0.0.0
2024-11-16 14:05:14 net_route_v4_best_gw result: via 192.168.75.2 dev ens160
2024-11-16 14:05:14 ROUTE_GATEWAY 192.168.75.2/255.255.255.0 IFACE=ens160 HWADDR=00:0c:29:af:bf:4c
2024-11-16 14:05:14 TUN/TAP device tun0 opened
2024-11-16 14:05:14 net_iface_mtu_set: mtu 1500 for tun0
2024-11-16 14:05:14 net_iface_up: set tun0 up
2024-11-16 14:05:14 net_addr_ptp_v4_add: 10.8.0.6 peer 10.8.0.5 dev tun0
2024-11-16 14:05:14 net_route_v4_add: 172.16.10.0/24 via 10.8.0.5 dev [NULL] table 0 metric -1
2024-11-16 14:05:14 net_route_v4_add: 10.8.0.1/32 via 10.8.0.5 dev [NULL] table 0 metric -1
2024-11-16 14:05:14 Initialization Sequence Completed
```

Now PIng VPN SERVER:

```
ping 192.168.75.145
```

Ping Host Only LAN Client From Nat Client

```
route -n
```

# ON VPN SERVER:

Tunnel Gets Created of IP 10.8.0.1

```
ip a
```

```
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::669e:6279:4d8f:3444/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
[root@vpnserver ~]#
```

VPN Certificate Connection Establishment

The connection between a Client and a VPN server is established using certificates through a process called mutual authentication. Here's a step-by-step breakdown:

1. **Certificate Generation**: The VPN server generates a server certificate and key, and the clients generate their own client certificates and keys. These certificates are signed by a trusted Certificate Authority (CA).
2. **Client Certificate Installation**: The client installs its own client certificate and private key.
3. **VPN Server Configuration**: The VPN server is configured to use the server certificate and key.
4. **Connection Establishment**: When a client initiates a VPN connection, it sends its client certificate to the VPN server.
5. **Server Verification**: The VPN server verifies the client certificate by checking its validity, ensuring it was signed by a trusted CA, and matching the client's identity information (e.g., Common Name).
6. **Client Verification**: The client verifies the VPN server's certificate by checking its validity, ensuring it was signed by a trusted CA, and matching the server's identity information.
7. **Authentication**: If both verifications succeed, the client and VPN server establish a secure TLS connection. This connection is encrypted and authenticated using the certificates and private keys.
8. **Key Exchange**: The client and VPN server exchange cryptographic keys for encrypting and decrypting data during the VPN session.
9. **VPN Session Establishment**: Once the secure connection is established, the client and VPN server negotiate and agree on VPN settings, such as IP addresses, subnet masks, and encryption algorithms.
10. **Data Encryption and Decryption**: The client and VPN server use the exchanged cryptographic keys to encrypt and decrypt data transmitted over the VPN connection.

In summary, the connection between a Client and a VPN server using certificates involves:

- Certificate generation and installation
- Mutual authentication and verification of client and server certificates
- Establishment of a secure TLS connection
- Key exchange for data encryption and decryption
- VPN session establishment and data transmission

This process ensures a secure and trusted connection between the client and VPN server, protecting data transmitted over the VPN.

1.

```
cd /etc/firewalld
```

```
vi direct.xml
```

rules are added in these file(IF rules are given wrong we can edit form here)

2. Certificate Problem

```
cd /etc/openvpn/easy-rsa/
```

After generating certificates they are stored in:

```
ls pki/issued/
```

Cat certificate name

Not Before: Usually this is not immediate
so check system date and time if it matches then it's ok otherwise take system and clients time ahead of Not Before date and time
Not After

on vpnclient1:

```
route -n
```

IF Que Comes to redirect traffic

```
sudo vi /etc/openvpn/server/server.conf
```

then uncomment:

#push "redirect-gateway def1": Used if employees use vpn to access Internet so it will redirect to gateway

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

Now check on VPNClient1

```
route -n
```

0.0.0.0 should get gateway 10.0.8.1