

ITIS 6240

Parag Mhatre

Project 2 – Setting up a VPN Server

The objective of the project is to setup a VPN server and connect to it. For this, we are using the Strongswan tool available for Linux. We have to generate keys, certificate, export the certificate to the client machine and then connect using IKEv2 with a predefined username and password.

## Step 1: Install Strongswan and other needed utilities

```
conf plugins/sshkey.conf plugins/pem.conf plugins/openssl.conf plugins/fips-prf.conf plugins/gmp.conf
plugins/curve25519.conf plugins/xcbc.conf plugins/cmac.conf plugins/hmac.conf plugins/attr.conf plug
as/kernel-libipsec.conf plugins/kernel-netlink.conf plugins/resolve.conf plugins/socket-default.conf
plugins/stroke.conf plugins/vici.conf plugins/updown.conf plugins/xauth-generic.conf plugins/dhchap.co
f plugins/counters.conf '/usr/share/strongswan/templates/config/plugins'
/bin/mkdir -p '/usr/share/strongswan/templates/config'
/usr/bin/install -c -m 644 strongswan.conf '/usr/share/strongswan/templates/config'
ake[4]: Leaving directory '/home/ubuntu/strongswan-5.7.1/conf'
ake[3]: Leaving directory '/home/ubuntu/strongswan-5.7.1/conf'
ake[2]: Leaving directory '/home/ubuntu/strongswan-5.7.1/conf'
aking install in init
ake[2]: Entering directory '/home/ubuntu/strongswan-5.7.1/init'
aking install in systemd
ake[3]: Entering directory '/home/ubuntu/strongswan-5.7.1/init/systemd'
ake[4]: Entering directory '/home/ubuntu/strongswan-5.7.1/init/systemd'
ake[4]: Nothing to be done for 'install-exec-am'.
/bin/mkdir -p '/lib/systemd/system'
/usr/bin/install -c -m 644 strongswan.service '/lib/systemd/system'
ake[4]: Leaving directory '/home/ubuntu/strongswan-5.7.1/init/systemd'
ake[3]: Leaving directory '/home/ubuntu/strongswan-5.7.1/init/systemd'
ake[3]: Entering directory '/home/ubuntu/strongswan-5.7.1/init'
ake[4]: Entering directory '/home/ubuntu/strongswan-5.7.1/init'
ake[4]: Nothing to be done for 'install-exec-am'.
ake[4]: Nothing to be done for 'install-data-am'.
ake[4]: Leaving directory '/home/ubuntu/strongswan-5.7.1/init'
ake[3]: Leaving directory '/home/ubuntu/strongswan-5.7.1/init'
ake[2]: Leaving directory '/home/ubuntu/strongswan-5.7.1/init'
aking install in testing
ake[2]: Entering directory '/home/ubuntu/strongswan-5.7.1/testing'
ake[3]: Entering directory '/home/ubuntu/strongswan-5.7.1/testing'
ake[3]: Nothing to be done for 'install-exec-am'.
ake[3]: Nothing to be done for 'install-data-am'.
ake[3]: Leaving directory '/home/ubuntu/strongswan-5.7.1/testing'
ake[2]: Leaving directory '/home/ubuntu/strongswan-5.7.1/testing'
aking install in scripts
ake[2]: Entering directory '/home/ubuntu/strongswan-5.7.1/scripts'
ake[3]: Entering directory '/home/ubuntu/strongswan-5.7.1/scripts'
ake[3]: Nothing to be done for 'install-exec-am'.
ake[3]: Nothing to be done for 'install-data-am'.
ake[3]: Leaving directory '/home/ubuntu/strongswan-5.7.1/scripts'
ake[2]: Leaving directory '/home/ubuntu/strongswan-5.7.1/scripts'
ake[2]: Entering directory '/home/ubuntu/strongswan-5.7.1'
ake[3]: Entering directory '/home/ubuntu/strongswan-5.7.1'
ake[3]: Nothing to be done for 'install-exec-am'.
ake[3]: Leaving directory '/home/ubuntu/strongswan-5.7.1'
ake[2]: Leaving directory '/home/ubuntu/strongswan-5.7.1'
ake[1]: Leaving directory '/home/ubuntu/strongswan-5.7.1'
ubuntu@ip-172-31-28-71:~/strongswan-5.7.1$
```

I used the following commands to install the needed tools:

```
sudo apt-get install build-essential
sudo apt-get install libgmp3-dev
sudo apt-get install libssl-dev
sudo apt-get install -y libgcrypt11-dev
sudo apt install -y iptables-persistent
```

Then I downloaded the Strongswan from the following link

wget <https://download.strongswan.org/strongswan-5.7.1.tar.bz2>

```
tar xjvf strongswan-5.7.1.tar.bz2; cd strongswan-5.7.1
```

```
./configure --prefix=/usr --sysconfdir=/etc -with-random-device=/dev/urandom --enable-all
```

```
Sudo make
```

```
Sudo make install
```

## Step 2: Create a Certificate Authority and Keys

I created a folder for the VPN Certificates and then used the following commands:

```
mkdir vpn-certs
```

```
cd vpn-certs
```

```
ipsec pki --gen --type rsa --size 4096 --outform pem > server-root-key.pem
```

Then for securing the key,

```
chmod 600 server-root-key.pem
```

```
buntu@ip-172-31-13-9:~$ mkdir vpn-certs
buntu@ip-172-31-13-9:~$ cd vpn-certs
buntu@ip-172-31-13-9:~/vpn-certs$ ipsec pki --gen --type rsa --size 4096 --outform pem > server-root-key.pem
buntu@ip-172-31-13-9:~/vpn-certs$ ipsec pki --self --ca --lifetime 3650 \
--in server-root-key.pem \
--type rsa --dn "C=US, O=VPN Server, CN=VPN Server Root CA" \
--outform pem > server-root-ca.pem
buntu@ip-172-31-13-9:~/vpn-certs$ ipsec pki --gen --type rsa --size 4096 --outform pem > server-root-key.pem
buntu@ip-172-31-13-9:~/vpn-certs$ chmod 600 server-root-key.pem
buntu@ip-172-31-13-9:~/vpn-certs$ ipsec pki --self --ca --lifetime 3650 \
--in server-root-key.pem \
--type rsa --dn "C=US, O=VPN Server, CN=VPN Server Root CA" \
--outform pem > server-root-ca.pem
buntu@ip-172-31-13-9:~/vpn-certs$
```

```
ipsec pki --self --ca --lifetime 3650 --in server-root-key.pem --type rsa --dn "C=US, O=VPN Server,
CN=VPN Server Root CA" --outform pem > server-root-ca.pem
```

```
ubuntu@ip-172-31-13-9: ~/vpn-certs
ubuntu@ip-172-31-13-9:~/vpn-certs$ ipsec pki --gen --type rsa --size 4096 --outform pem > server-root-key.pem
ubuntu@ip-172-31-13-9:~/vpn-certs$ ipsec pki --self --ca --lifetime 3650 \
> --in server-root-key.pem \
> --type rsa --dn "C=US, O=VPN Server, CN=VPN Server Root CA" \
> --outform pem > server-root-ca.pem
ubuntu@ip-172-31-13-9:~/vpn-certs$ ipsec pki --gen --type rsa --size 4096 --outform pem > server-root-key.pem
ubuntu@ip-172-31-13-9:~/vpn-certs$ chmod 600 server-root-key.pem
ubuntu@ip-172-31-13-9:~/vpn-certs$ ipsec pki --self --ca --lifetime 3650 \
> --in server-root-key.pem \
> --type rsa --dn "C=US, O=VPN Server, CN=VPN Server Root CA" \
> --outform pem > server-root-ca.pem
ubuntu@ip-172-31-13-9:~/vpn-certs$ ipsec pki --gen --type rsa --size 4096 --outform pem > vpn-server-key.pem
ubuntu@ip-172-31-13-9:~/vpn-certs$ ipsec pki --pub --in vpn-server-key.pem \
> --type rsa | ipsec pki --issue --lifetime 1825 \
> --cacert server-root-ca.pem \
> --cakey server-root-key.pem \
> --dn "C=US, O=VPN Server, CN=ec2-18-221-9-7.us-east-2.compute.amazonaws.com" \
> --san ec2-18-221-9-7.us-east-2.compute.amazonaws.com \
> --flag serverAuth --flag ikeIntermediate \
> --outform pem > vpn-server-cert.pem
ubuntu@ip-172-31-13-9:~/vpn-certs$
```

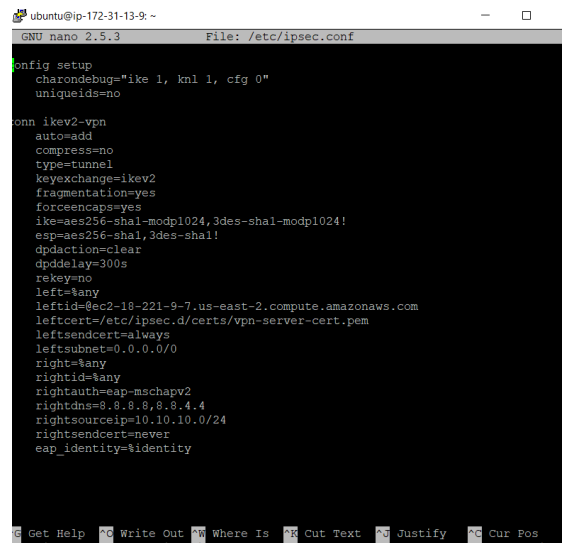
```
ipsec pki --gen --type rsa --size 4096 --outform pem > vpn-server-key.pem
```

```
$ ipsec pki --pub --in vpn-server-key.pem --type rsa | ipsec pki --issue --lifetime 1825 \
```

```
$ --cacert server-root-ca.pem --cakey server-root-key.pem \
$ --dn "C=US, O=VPN Server, CN= ec2-18-221-9-7.us-east-2.compute.amazonaws.com" \
$ --san ec2-18-221-9-7.us-east-2.compute.amazonaws.com \
$ --flag serverAuth --flag ikeIntermediate --outform pem > vpn-server-cert.pem
```

### Step 3 : Configuring Strong Swan

I did this by configuring ipsec.conf and ipsec.secrets



```
ubuntu@ip-172-31-13-9: ~
GNU nano 2.5.3 File: /etc/ipsec.conf

#config setup
charondebug="ike 1, knl 1, cfg 0"
uniqueids=no

conn ikev2-vpn
    auto=add
    compress=no
    type=tunnel
    keyexchange=ikev2
    fragmentation=yes
    forceencaps=yes
    ike=aes256-sha1-modp1024,3des-sha1-modp1024!
    esp=aes256-sha1,3des-sha1!
    dpdaction=clear
    dpddelay=300s
    rekey=no
    left=%any
    leftid=@ec2-18-221-9-7.us-east-2.compute.amazonaws.com
    leftcert=/etc/ipsec.d/certs/vpn-server-cert.pem
    leftsendcert=always
    leftsubnet=0.0.0.0/0
    right=%any
    rightid=%any
    rightauth=eap-mschapv2
    rightdns=8.8.8.8,8.8.4.4
    rightsourcips=10.10.10.0/24
    rightsendcert=never
    eap_identity=identity
```

Then I reloaded the settings using `sudo ipsec reload`.

### Step 4: Configuration of the firewall and some other stuff

I used the following commands to configure the firewall

```
$sudo iptables -P INPUT ACCEPT
$ sudo iptables -P FORWARD ACCEPT
$ sudo iptables -F
$ sudo iptables -Z
$ sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
$ sudo iptables -A INPUT -i lo -j ACCEPT
$ sudo iptables -A INPUT -p udp --dport 500 -j ACCEPT
$ sudo iptables -A INPUT -p udp --dport 4500 -j ACCEPT
$ sudo iptables -A FORWARD --match policy --pol ipsec --dir in --proto esp -s 10.10.10.10/24 -j ACCEPT
$ sudo iptables -A FORWARD --match policy --pol ipsec --dir out --proto esp -d 10.10.10.10/24 -j ACCEPT
$ sudo iptables -t nat -A POSTROUTING -s 10.10.10.10/24 -o eth0 -m policy --pol ipsec --dir out -j ACCEPT
$ sudo iptables -t nat -A POSTROUTING -s 10.10.10.10/24 -o eth0 -j MASQUERADE
$ sudo iptables -t mangle -A FORWARD --match policy --pol ipsec --dir in -s 10.10.10.10/24 -o eth0 -p tcp
-m tcp --tcp-flags SYN,RST SYN -m tcpmss --mss 1361:1536 -j TCPMSS --set-mss 1360
$ sudo iptables -A INPUT -j DROP
$ sudo iptables -A FORWARD -j DROP
```

I edited the following file:

```
sudo nano /etc/syctl.conf
```

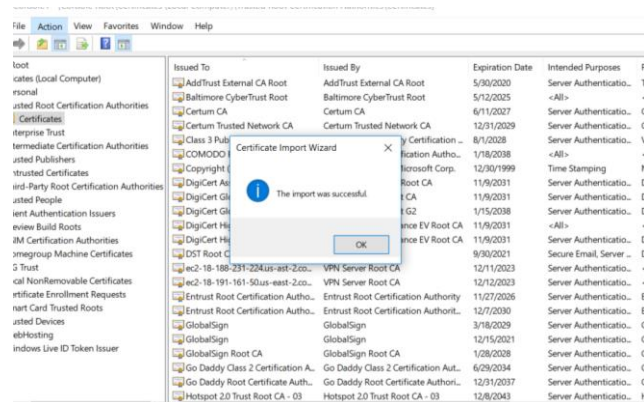
# Uncomment below lines to enable packet forwarding for IPv4 and prevent mitm attacks.

```
net.ipv4.ip_forward=1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.ip_no_pmtu_disc = 1
```

## Step 5: Setting up the Client side

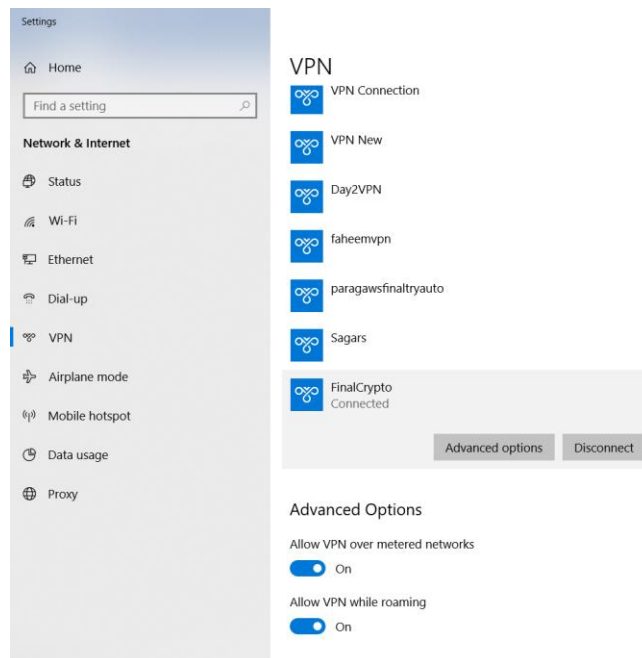
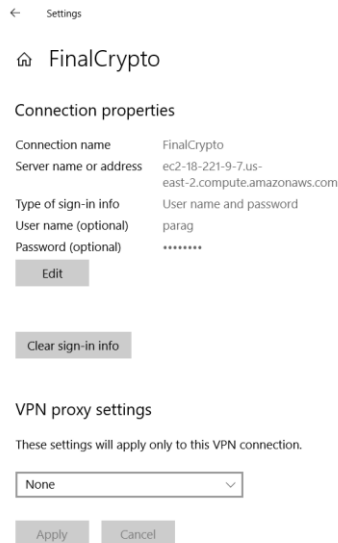
Then I copied the CA key to my local system and installed the certificate.

```
proj2 - Notepad
File Edit Format View Help
-----BEGIN CERTIFICATE-----
MIIFQjCCAYqgAwIBAgIIGSwTqPb3tfeWQY3KozIhvcNAQEMBAwPZELMAGAIUE
BhMCVPMKxZARBgNBaoTC1ZQ1B1ZXJ2ZXIwZGZBZGhVBAITE1ZQ1B1ZXJ2ZXIw
UMpvcuBDQTAeFw00ODEyMTQ0MzHSTVUwFw0ODEyMTQ0MzHSTVUwDQExCjA3BgNV
BAYTAVTRMhEYQYVQVQKEwpmUE4gU2YyZmVudVhRswQYVQVQVQKEwpmUE4gU2YyZmV
IFJ1b3Q0Q0EwggE1MAAGCSqGSIb3DQEBAQUAA4ICDwAwggE1KAo1CAQOpYbpxSfY
fq1UQL8d/ZSCBAH3st3DVh+2XKQjghwlnpn0XvJG6W12K75rGrE1Qd3mbVLuK5P+
kpdz0zmynr1F1QRTNugKcgo5puV1aohjQbVPF3JG6HYK6U41uaf3k/dTywlp3y
KC/09vXHrDD0S22M486MUTCfX0J3Q5yc04dnck17ENDd47ZUSEcyHHG6XezTmI
oy9rA/7j9TYXLkovhNNUeuHFCvXpVcb3n35wm130vu3JEr3eKXe1EjreV9r731R
TCpLm1CFHY3oUmGFM7H7GP1724VXy3ke29Ku/V1yJmUF5soHCDaehQ5ESJ7UorS
Is3YHwktTL1QYrT3V1/ZhxFQ0W1E2ZrB+Lahn85mWcm012+u1215Mu00eH4gt
yAZGy/vWavygdgK5Xdh3q35vtfBaeAeoYvgJp/YhAkDalG3N0304t171K/JH
E2F0MC3+dgmpu9JRLSLBQ118jbn3mntzoEKKoiz5Xdhzbo3J3H5s1LXog2yAxG0
2xpBN7qM2T5hES50Ddp7/BWk7FfTnhpCo/TvyvICogxtF1dvYKhqX116chWw5NY
NdAadaG6J82RUG8F7KCCv0FHSq091MR5unosX6hLANLncdCi/BSA+PgrAXCTTKA
x2Ly1w4wI1aJY4ZuHzWGN1KOR1f+hq3p6wIDAQAB01wQDAPBgNVHRBBAF8EBTAD
AQH/MA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUAwn4Toc2a8g1uW28AY81G67
YN4wQYK0Z3IhvcNAQEMBAQAgIBAHYKTO4Ea6e9tenEfyxh7KXfGgk4w5n8q2g
970pBC+y70111oLQQM01VRXJ8wv116ms0301vaJv9cSG9XJIFqEx3QFNZr0n0wy
i6R1kK2M15NqWJ789gVzsqXupF78yBUS1gy8Xo+LzFOTWuCIahtxG5bf1hQo/+
QIL1R8F2B15YwKXp1wJrC27yhgqbwSm6zuUq2BFZIQGMO/XK01sEAGtUc35yVtB
A3nqT5j3fBmCJ1Wc1sncf5Y7TnvLYC/9AKQh2fuddrnbmZvBE+HF5oPGFnTuvJHF
KtdXXQXQj9T+RNA0eAaICfGvUBurwMC3/ClQJ6fCG3PTL+36j8x31V6a8JegOGj
Vov+YvuxiaG9RUXSCKqRqKv39A115NUubeK5acFbvcvob0mdmZK2UBKcWc
2FvMuVat9LigFFXwE/tA1mbT6g2130ZCY6ATvpKwYatcfrVhfaoM8NMNRDq1u
fZKF1GXVPq6W2ML1TTUPpooQ10x+C093zeZmPHV0ebvFQ6b288ZeJNNVhcl0eW
1s6+dhHm1KtdPHdQ3kffVKOhtcVom5ohF2st4X3a5EgTb7m5w8Wq63JAF1BEhB
/zN1IYEHYb0GBBj+hXIAH9ZWRfN1LYYmAZMF8qwwhDophrm7B5uRqt/1H+4V1w
VOfXNHNW
-----END CERTIFICATE-----
```



## Step 6: Establishing Connection

Then I created a VPN connection and configured it as seen in the screenshots.



After connecting it to the VPN server, my system was assigned a different IP as seen in the Screenshots:

10.10.10.1

```

Autoconfiguration Enabled . . . . . : Yes

Wireless LAN adapter Local Area Connection* 3:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . : 
Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
Physical Address. . . . . : 74-40-B8-38-6D-F5
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes

Wireless LAN adapter FinalCrypto:

Connection-specific DNS Suffix . . : 
Description . . . . . : FinalCrypto
Physical Address. . . . . : 
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . . : Yes
IPv4 Address. . . . . : 10.10.10.1(Preferred)
Subnet Mask . . . . . : 255.255.255.255
Default Gateway . . . . . : 
DNS Servers . . . . . : 8.8.8.8
                        8.8.4.4
NetBIOS over Tcpip. . . . . : Enabled

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . . : 
Description . . . . . : Realtek RTL8822BE 802.11ac PCIe Adapter
Physical Address. . . . . : 74-40-B8-38-6D-F5
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::1168:ab0:9407:1470%9(Preferred)
IPv4 Address. . . . . : 172.20.1.31(Preferred)
Subnet Mask . . . . . : 255.255.0.0
Lease Obtained. . . . . : Thursday, December 13, 2018 1:05:33 PM
Lease Expires . . . . . : Friday, December 14, 2018 11:09:59 PM
Default Gateway . . . . . : 172.20.1.1
DHCP Server . . . . . : 172.20.1.1
DHCPv6 IAID . . . . . : 74727611
DHCPv6 Client DUID. . . . . : 00-01-00-01-23-73-B5-4F-B4-B6-86-E9-D8-ED
DNS Servers . . . . . : 172.20.1.1
NetBIOS over Tcpip. . . . . : Enabled

Users\p>

```

The assignment of the IPs is seen from the Server logs below.

```

[NET] sending packet: from 172.31.13.9[4500] to 173.95.57.197[4500] (544 bytes)
[NET] sending packet: from 172.31.13.9[4500] to 173.95.57.197[4500] (544 bytes)
[NET] sending packet: from 172.31.13.9[4500] to 173.95.57.197[4500] (544 bytes)
[NET] sending packet: from 172.31.13.9[4500] to 173.95.57.197[4500] (176 bytes)
[NET] received packet: from 173.95.57.197[4500] to 172.31.13.9[4500] (76 bytes)
[ENC] parsed IKE_AUTH request 2 [ EAP/RES/ID ]
[IKE] received EAP identity 'parag'
[IKE] initiating EAP_MSCHAPV2 method (id 0xB0)
[ENC] generating IKE_AUTH response 2 [ EAP/REQ/MSCHAPV2 ]
[NET] sending packet: from 172.31.13.9[4500] to 173.95.57.197[4500] (108 bytes)
[NET] received packet: from 173.95.57.197[4500] to 172.31.13.9[4500] (140 bytes)
[ENC] parsed IKE_AUTH request 3 [ EAP/RES/MSCHAPV2 ]
[ENC] generating IKE_AUTH response 3 [ EAP/REQ/MSCHAPV2 ]
[NET] sending packet: from 172.31.13.9[4500] to 173.95.57.197[4500] (140 bytes)
[NET] received packet: from 173.95.57.197[4500] to 172.31.13.9[4500] (76 bytes)
[ENC] parsed IKE_AUTH request 4 [ EAP/RES/MSCHAPV2 ]
[IKE] EAP method EAP_MSCHAPV2 succeeded, MSK established
[ENC] generating IKE_AUTH response 4 [ EAP/SUCC ]
[NET] sending packet: from 172.31.13.9[4500] to 173.95.57.197[4500] (76 bytes)
[NET] received packet: from 173.95.57.197[4500] to 172.31.13.9[4500] (92 bytes)
[ENC] parsed IKE_AUTH request 5 [ AUTH ]
[IKE] authentication of '172.20.1.31' with EAP successful
[IKE] authentication of 'ec2-18-221-9-7.us-east-2.compute.amazonaws.com' (myse
) with EAP
[IKE] IKE_SA ikv2-vpn[1] established between 172.31.13.9[ec2-18-221-9-7.us-ea
-2.compute.amazonaws.com]...173.95.57.197[172.20.1.31]
[IKE] peer requested virtual IP many
[IKE] assigning virtual IP 10.10.10.1 to peer 'parag'
[IKE] CHILD_SA ikv2-vpn[1] established with SPIs c5eac45f_i has68eae_o and TS
.0.0.0/0 ==> 10.10.10.1/32
[ENC] generating IKE_AUTH response 5 [ AUTH CPRP(ADDR DNS DNS) SA TS1 TSr N(MO
DE_SUPP) N(MO_ADD_ADDR) ]
[NET] sending packet: from 172.31.13.9[4500] to 173.95.57.197[4500] (236 bytes)
[IKE] sending keep alive to 173.95.57.197[4500]

```

Thus I got the new IP in the Class A of the ip addresses as seen in the Screenshots and successfully setup a VPN server.