😊 AWS Client VPN

AWS Client VPN is a fully-managed VPN service provided by Amazon Web Services (AWS) that enables users to securely access resources within their AWS Virtual Private Cloud (VPC) or on-premises networks from anywhere using OpenVPN-based clients. It allows organizations to establish secure connections for remote users or devices to access resources in their AWS infrastructure or on-premises networks.

Key features of AWS Client VPN include:

1. **Secure Connectivity:** AWS Client VPN uses industry-standard encryption and authentication protocols to ensure secure communication between client devices and the AWS network. It supports both SSL/TLS and IKEv2 VPN protocols to accommodate different client configurations.
2. **Client Compatibility:** AWS Client VPN is compatible with a wide range of client devices and operating systems, including Windows, macOS, Linux, iOS, and Android. Users can easily connect to the VPN using native VPN clients or third-party OpenVPN clients.
3. **Centralized Management:** AWS Client VPN is fully managed and integrated with the AWS Management Console, allowing administrators to easily configure and manage VPN connections, user authentication, and access control policies from a centralized interface.
4. **Scalability:** AWS Client VPN is designed to scale with the needs of the organization, supporting thousands of simultaneous client connections. It can be easily deployed and managed across multiple AWS regions to provide high availability and redundancy.
5. **Integration with AWS Services:** AWS Client VPN seamlessly integrates with other AWS services, allowing remote users to securely access resources within their AWS VPCs, such as Amazon EC2 instances, Amazon RDS databases, or AWS Lambda functions. It also supports integration with AWS Directory Service for user authentication using Microsoft Active Directory.
6. **Monitoring and Logging:** AWS Client VPN provides detailed monitoring and logging capabilities, allowing administrators to monitor VPN usage, track connection logs, and troubleshoot connectivity issues using Amazon CloudWatch Logs and Amazon CloudWatch Metrics.

Overall, AWS Client VPN offers a convenient and secure solution for providing remote access to AWS resources and on-premises networks, enabling organizations to extend their network securely to remote users or devices while leveraging the scalability and reliability of the AWS cloud.

😄 Use cases of AWS Client VPN:

AWS Client VPN is employed in various use cases, including:

1. **Remote Workforce Access:** In today's distributed work environment, organizations use AWS Client VPN to provide secure access to corporate resources for remote employees, contractors, or partners. It enables remote workers to securely connect to

the organization's AWS infrastructure or on-premises networks from anywhere, ensuring productivity and collaboration while maintaining security.

2. **Secure Access to AWS Resources:** AWS Client VPN allows organizations to securely access resources within their AWS Virtual Private Cloud (VPC) from remote locations. This includes accessing Amazon EC2 instances, Amazon RDS databases, Amazon S3 buckets, or other AWS services securely over the internet, without exposing them to public access.
3. **Partner and Vendor Connectivity:** Organizations can use AWS Client VPN to provide secure connectivity for partners, vendors, or third-party service providers who need access to specific resources within their AWS infrastructure. This enables secure collaboration and data exchange while ensuring compliance with security and regulatory requirements.
4. **Temporary Site-to-Site Connectivity:** AWS Client VPN can be used as a temporary solution for site-to-site connectivity between on-premises networks and AWS VPCs. It provides a convenient way to establish secure connections for migration, disaster recovery, or temporary workload deployments without the need for dedicated hardware VPN appliances.
5. **BYOD (Bring Your Own Device) Support:** With the proliferation of mobile devices and remote work policies, AWS Client VPN enables organizations to support BYOD initiatives by allowing employees to securely connect to corporate resources using their personal devices. This enhances flexibility and productivity while maintaining security and compliance.
6. **Secure Access for IoT Devices:** Organizations deploying IoT (Internet of Things) devices can use AWS Client VPN to establish secure connections for remote monitoring, management, and control of IoT devices deployed in the field. It ensures that communication between IoT devices and cloud services is encrypted and protected from unauthorized access.
7. **Compliance and Regulatory Requirements:** AWS Client VPN helps organizations meet compliance and regulatory requirements, such as GDPR, HIPAA, or PCI DSS, by providing a secure and auditable solution for remote access to sensitive data and resources. It offers encryption, access control, and logging features necessary for demonstrating compliance with various security standards.

😄 Benefits of AWS Client VPN:

The benefits of AWS Client VPN include:

1. **Secure Remote Access:** AWS Client VPN provides a secure way for remote users to access resources within AWS VPCs or on-premises networks. It employs industry-standard encryption and authentication protocols to ensure that data transmitted over the VPN connection remains confidential and protected from unauthorized access.
2. **Scalability:** AWS Client VPN is designed to scale with the needs of the organization, supporting thousands of simultaneous client connections. It can be easily deployed and managed across multiple AWS regions, providing high availability and redundancy for remote access.

3. **Centralized Management:** AWS Client VPN is fully managed and integrated with the AWS Management Console, allowing administrators to centrally configure and manage VPN connections, user authentication, and access control policies. This simplifies management tasks and reduces operational overhead.
4. **Compatibility:** AWS Client VPN is compatible with a wide range of client devices and operating systems, including Windows, macOS, Linux, iOS, and Android. Users can easily connect to the VPN using native VPN clients or third-party OpenVPN clients, providing flexibility and convenience.
5. **Integration with AWS Services:** AWS Client VPN seamlessly integrates with other AWS services, allowing remote users to securely access resources within AWS VPCs, such as Amazon EC2 instances, Amazon RDS databases, or Amazon S3 buckets. It also supports integration with AWS Directory Service for user authentication using Microsoft Active Directory.
6. **Monitoring and Logging:** AWS Client VPN provides detailed monitoring and logging capabilities, allowing administrators to monitor VPN usage, track connection logs, and troubleshoot connectivity issues using Amazon CloudWatch Logs and Amazon CloudWatch Metrics. This helps ensure visibility into VPN activity and facilitates compliance with security and regulatory requirements.
7. **Cost-Effective:** AWS Client VPN is a cost-effective solution for providing secure remote access to AWS resources and on-premises networks. It eliminates the need for dedicated VPN appliances or infrastructure, reducing upfront capital expenses and ongoing maintenance costs associated with traditional VPN solutions.

AWS Client VPN provides secure and scalable remote access to AWS resources and on-premises networks, ensuring data privacy and compliance.

A multinational corporation uses AWS Client VPN to enable its remote workforce to securely access sensitive company data and applications from anywhere, facilitating productivity and collaboration while maintaining security standards.

🤓 Here's the technical summary of what you are going to do:

1. **Instance Setup:** Launch an EC2 instance on AWS.
2. **Certificate Installation:** Install CA, server, and client certificates on the instance for authentication.
3. **Authentication Process:** Utilize AWS documentation for authentication and certificate acquisition.
4. **VPN Endpoint Creation:** Establish a VPN endpoint in the VPC, associating it with the necessary components.
5. **Configuration Download:** Download configurations for OpenVPN and AWS VPN clients.
6. **Client Configuration:** Configure the .ovpn file with client certificate and key for mutual authentication.

7. **Connection Establishment:** Upload the configured .ovpn file to the OpenVPN client and establish a connection.
8. **Cleanup Procedure:** Disassociate network, delete VPN endpoint, certificates, and terminate the instance to conclude the setup.

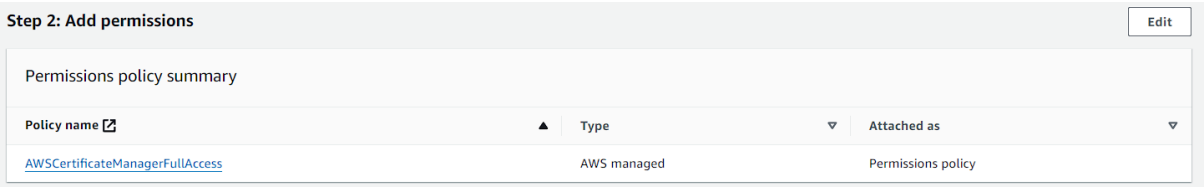
In this setup, you're configuring a Virtual Private Network (VPN) using AWS services. The end goal is to establish a secure connection between your local machine and AWS infrastructure, allowing encrypted communication and secure access to AWS resources from remote locations. This enables secure remote work, protects sensitive data during transmission, and enhances access control and network security within the AWS environment.

😁 To begin with the Lab:

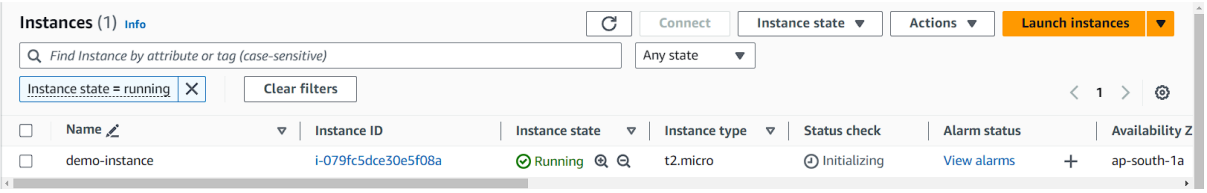
1. Login to AWS Console, then navigate to EC2 and create an instance.
2. Now in this instance we are going to install three certificates, the first is a CA Certificate, the second is a Server Certificate, and the third one is Client Certificate.
3. Server and client are straightforward to understand. The server certificate will be used by the client VPN endpoint. The client certificate is what we will be using as part of our OpenVPN client in our laptop/desktop to connect to the VPN, and the CA certificate is the one from which both of these certificates will be issued.
4. Now you can use the below AWS link for the authentication and downloading certificates over your instance.

   https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/client-auth-mutual-enable.html

5. Now navigate to EC2 and launch your instance in Amazon Linux AMI. Then you need to go to IAM and create a role for EC2, attach this Permission Policy which you can see in the snapshot, Certificate Manager Full Access.



6. Once your instance is launched and the role is attached to it then you have to SSH into it.



7. Once you are in, quickly switch to the root user.

   sudo su -

```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-32-226 ~]$ sudo su -
[root@ip-172-31-32-226 ~]#
```

8. Now the very first thing you have to do is install git in your instance.

   yum -y install git

```
[root@ip-172-31-32-226 ~]# yum -y install git
```

9. Once the git utility is installed go to the documentation page and run the commands one by one.

```
[root@ip-172-31-32-226 ~]# git clone https://github.com/OpenVPN/easy-rsa.git
Cloning into 'easy-rsa'...
remote: Enumerating objects: 6340, done.
remote: Counting objects: 100% (1116/1116), done.
remote: Compressing objects: 100% (408/408), done.
remote: Total 6340 (delta 741), reused 916 (delta 708), pack-reused 5224
Receiving objects: 100% (6340/6340), 52.09 MiB | 19.05 MiB/s, done.
Resolving deltas: 100% (2990/2990), done.
[root@ip-172-31-32-226 ~]# cd easy-rsa/easyrsa3
[root@ip-172-31-32-226 easyrsa3]# ./easyrsa init-pki

Notice
------
'init-pki' complete; you may now create a CA or requests.

Your newly created PKI dir is:
* /root/easy-rsa/easyrsa3/pki


Using Easy-RSA configuration:
* undefined
```

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

Notice
------
CA creation complete. Your new CA certificate is at:
* /root/easy-rsa/easyrsa3/pki/ca.crt
```

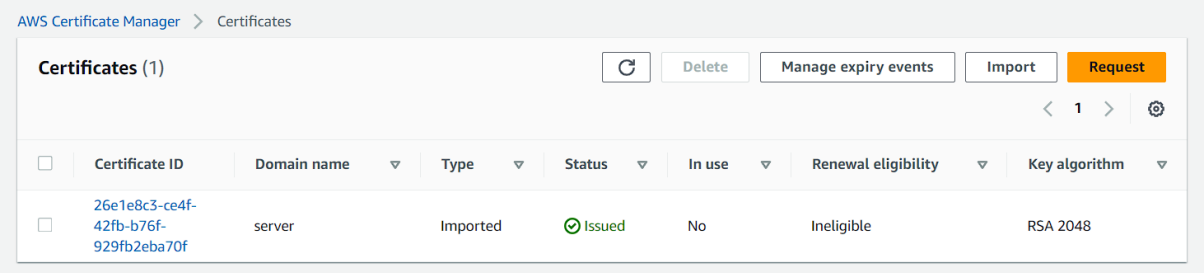
10. Once your certificates are generated then you have to follow step no. 6 of the documentation. Which says Copy the server certificate and key, the client certificate and key to a custom folder, then navigate into the custom folder.
11. So, copy all the commands from step no. 6 and paste them into your instance. Then if you do a listing of objects inside your custom folder, you can see the certificates.

```
[root@ip-172-31-32-226 easyrsa3]# mkdir ~/custom_folder/
[root@ip-172-31-32-226 easyrsa3]# cp pki/ca.crt ~/custom_folder/
[root@ip-172-31-32-226 easyrsa3]# cp pki/issued/server.crt ~/custom_folder/
[root@ip-172-31-32-226 easyrsa3]# cp pki/private/server.key ~/custom_folder/
[root@ip-172-31-32-226 easyrsa3]# cp pki/issued/client1.domain.tld.crt ~/custom_folder
[root@ip-172-31-32-226 easyrsa3]# cp pki/private/client1.domain.tld.key ~/custom_folder/
[root@ip-172-31-32-226 easyrsa3]# cd ~/custom_folder/
[root@ip-172-31-32-226 custom_folder]# ls
ca.crt  client1.domain.tld.crt  client1.domain.tld.key  server.crt  server.key
[root@ip-172-31-32-226 custom_folder]#
```

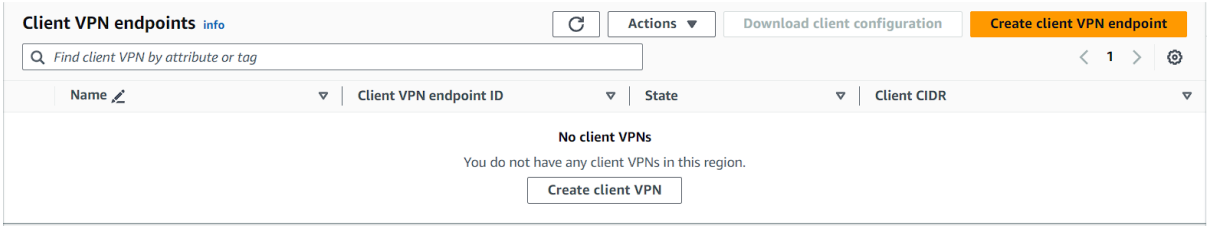
12. Once this is all done now you have to go and import the server certificate from the certificate manager.
13. For that you can follow step no. 7 in the AWS documentation which says to upload the certificate key. Also, you only have to upload the **Server Certificate.**
14. Then you can run the command. One important thing is that you should be in the custom folder while running this command.

```
[root@ip-172-31-32-226 custom_folder]# aws acm import-certificate --certificate fileb://server.crt --private-key fileb://server.key --certificate-chain fileb://ca.crt
{
    "CertificateArn": "arn:aws:acm:ap-south-1:878893308172:certificate/26e1e8c3-ce4f-42fb-b76f-929fb2eba70f"
}
[root@ip-172-31-32-226 custom_folder]#
```

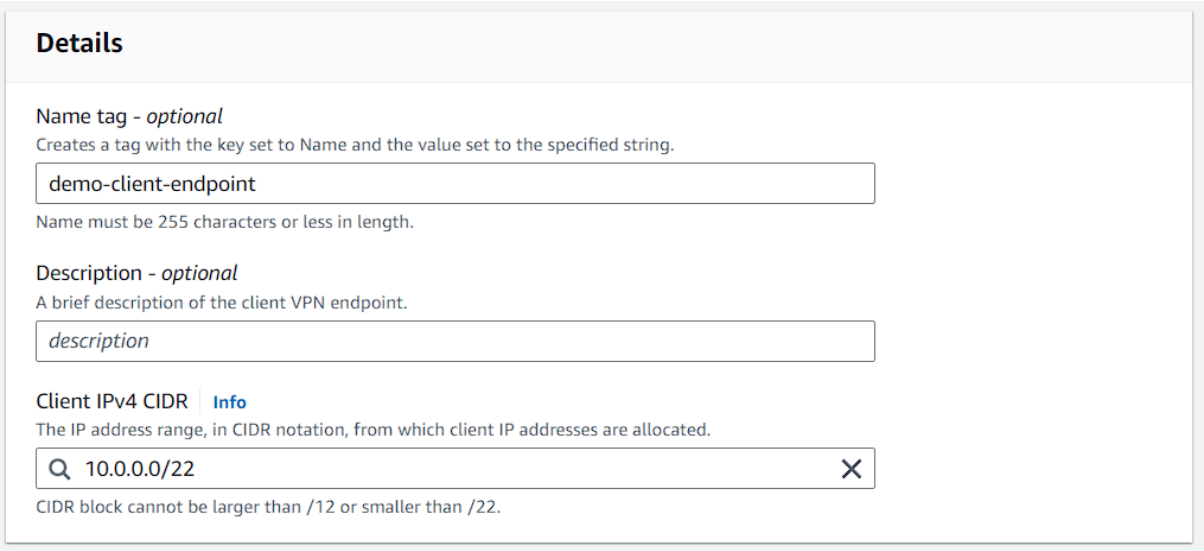
15. After that if you go into your certificate manager and list your certificates you can see your server certificate there.



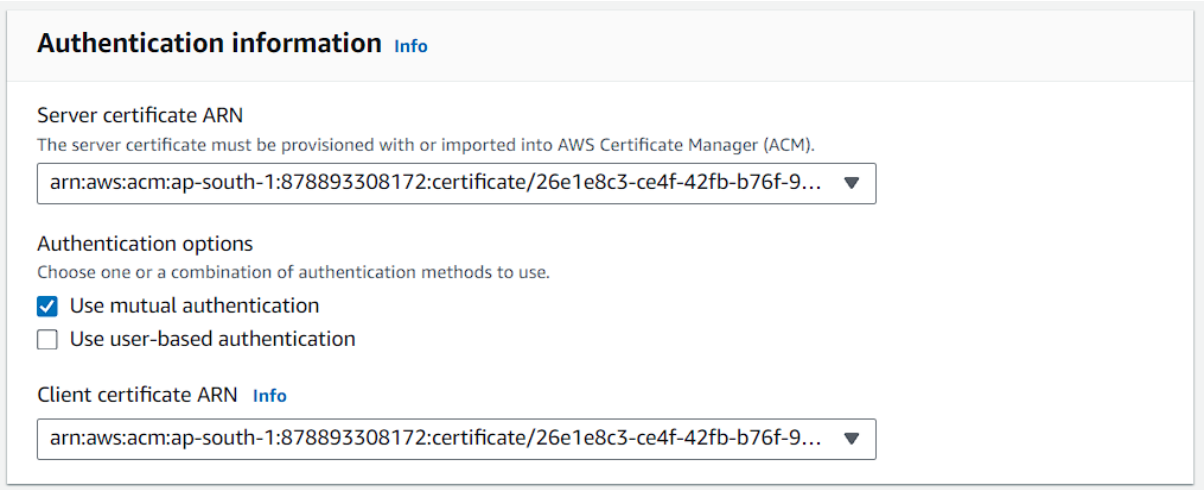
16. Then you have to navigate to VPC and here from the left pane go to VPN expand it, open client VPN endpoints and click on create.



17. Give a name to your client VPN, for the CIDR block you can use the same as shown below.



18. After that select your Server certificate ARN and in the authentication option choose mutual authentication.
19. Then in the client certificate choose the same certificate which is your Server Certificate.



20. After that give DNS server IP addresses, and use the same IP as shown below. Then select your default VPC and a security group. Create your endpoint.

DNS server 1 IP address

The IP address of the DNS server to use. There are no default DNS servers.

171.31.0.2

DNS server 2 IP address

The IP address of the DNS server to use. There are no default DNS servers.

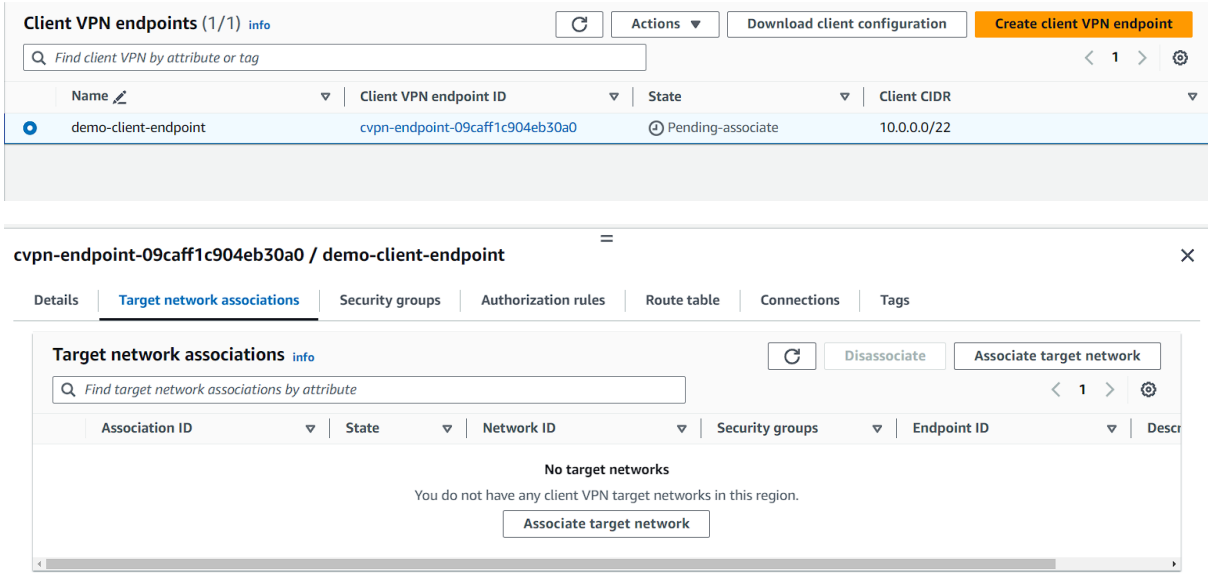
8.8.8.8

Transport protocol Info

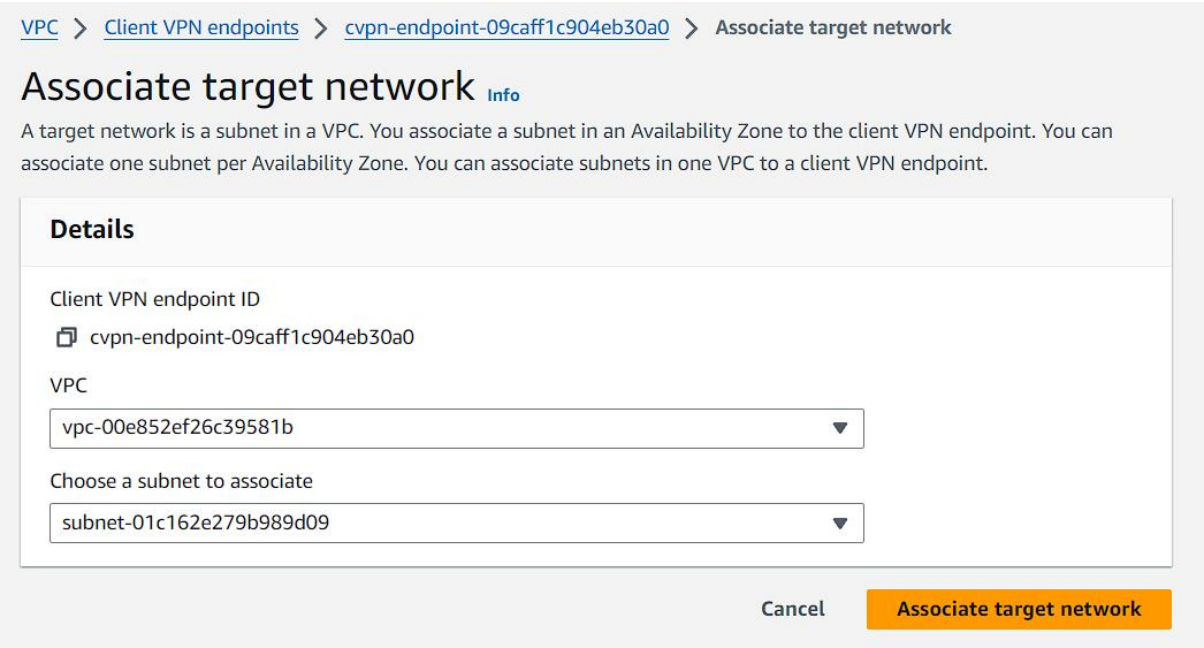
Transport protocol used by the TLS sessions.

- (•) UDP
- ( ) TCP

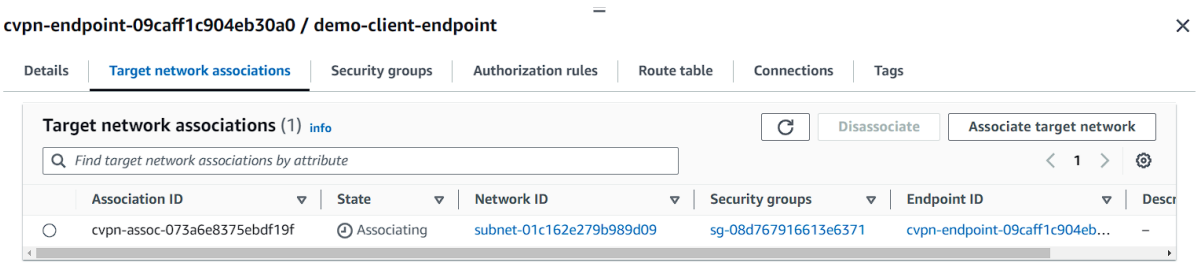
21. Currently you will see that it is in a pending associate state. For that select it and move to target association, click on Associate.



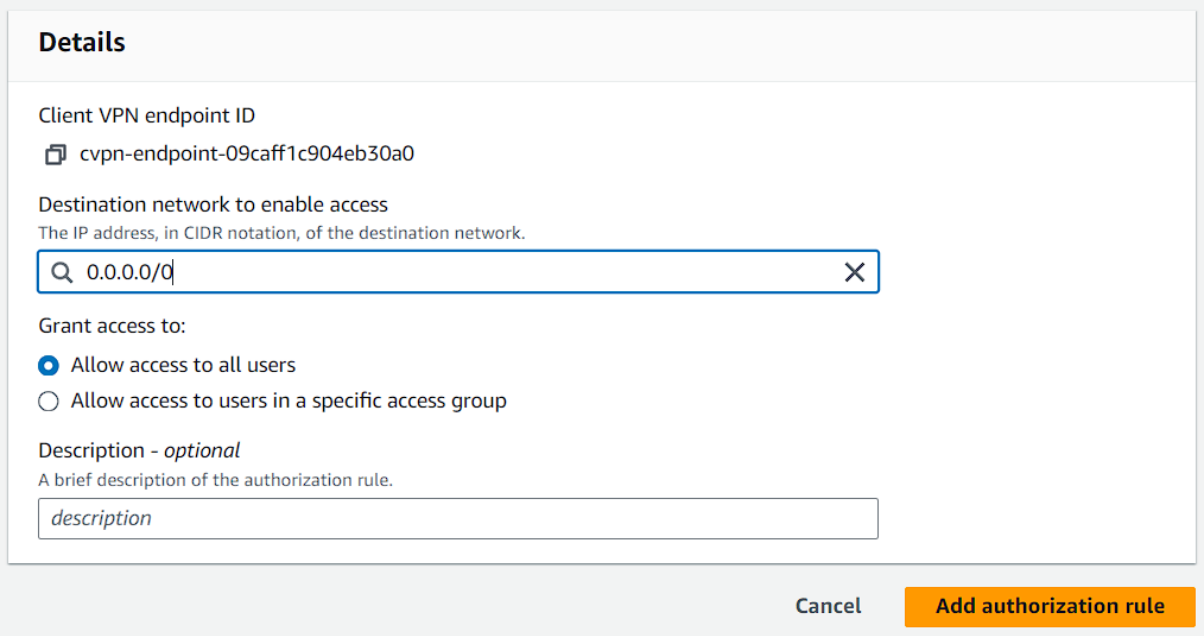
22. Here first choose your default VPC then any of the subnets of your choice would be enough or you can choose the same subnet as your instance.



23. Once you have associated it you will see that it is associating. You have to wait for some time until it gets associated. Might take 5-10 minutes.



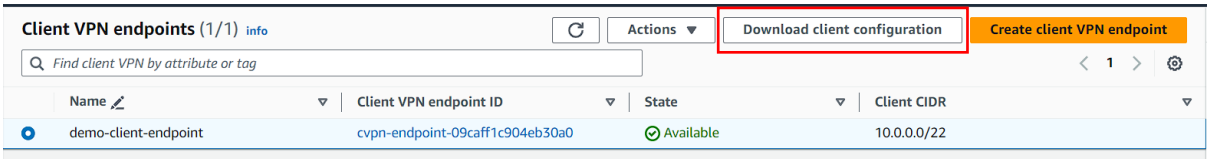
24. As it is associating with the target, you can go to **authorization rules** and add a rule for it.
25. Give the mentioned CIDR below and select Allow access to all users.



26. Once it is all done now just verify that the security group that you have associated with the endpoint should have **All traffic enabled from everywhere.**
27. You can use the below links to download the VPN software on your laptop.

    https://openvpn.net/client/
    https://aws.amazon.com/vpn/client-vpn-download/

28. Once they are downloaded just install them on your local machine.
29. Now you will notice that after installation there is an option to upload a file.
30. So, in your client VPN endpoint you will see an option for downloading client configuration. Click on it.
31. You will see that a file has been downloaded on your laptop.



32. Now if you will just open this file in Notepad you will see that here you have the protocol and you have the DNS associated with the VPN server.
33. So, whenever you are using the open VPN utility it will require all of these details to connect to our open VPN server.

```
client
dev tun
proto udp
remote cvpn-endpoint-09caff1c904eb30a0.prod.clientvpn.ap-south-1.amazonaws.com 443
remote-random-hostname
resolv-retry infinite
nobind
remote-cert-tls server
cipher AES-256-GCM
verb 3
<ca>
-----BEGIN CERTIFICATE-----
MIIDTjCCAjagAwIBAgIUaCeHDeN7TI2TjqWUEm10AkO9RQowDQYJKoZIhvcNAQEL
BQAwFzEVMBMGA1UEAwwMY2EuZGVtby5jZXJ0MB4XDTI0MDMwMjExMDM1MloXDTM0
MDIyODExMDM1MlowFzEVMBMGA1UEAwwMY2EuZGVtby5jZXJ0MIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAn0aE6Eh+qa14BrMcURSOQ78s8WT6ivMkvPBd
K0yJB10/PdG44QQWRFlIrZ03dOEAFtYK0nW3dOr+auEV0kU1v6R1wofx4qyq/WSf
+IcLJfSLrmGBi9e4it2N43Vcpr9lf27/IEkyrDkUSUwe8I8jHbVD7EV+2/gcNK1E
jpodPjlV9/0ustALbaaQRig6JqB4B6TiC6Tc0fJTN18+eDmyr/9u7Fdo8FEvpl7k
MAzZFNA5bF9/ys2++Gbf4VVXfFOqi7KtwEyotV3mNHKCkuV9B8DVszQW30Axwyj0
SqW9n1AmzgBtsxJ11RfBin+jp2iMK3F7DYqVuQrjwxpuhXhfJwIDAQABo4GRMIGO
MAwGA1UdEwQFMAMBAf8wHQYDVR0OBBYEFI7oGhJw/oASX4XTTt5Z6cI8jLOaMFIG
A1UdIwRLMEmAFI7oGhJw/oASX4XTTt5Z6cI8jLOaoRukGTAXMRUwEwYDVQQDDAxj
YS5kZW1vLmNlcnSCFGgnhw3je0yNk46llBJtdAJDvUUKMAsGA1UdDwQEAwIBBjAN
BgkqhkiG9w0BAQsFAAOCAQEAMT2adMlBqOFTLXnpvIf9k8Hv2c9RQ4mw6Weh/cIP
JbPrOORMzkhyKsW7VKqyFAejScButMNrg+XLpD1iCCczkLWt9z4ILK7SkNxuBwba
NPnUC1UcQwbYbnqZOXjP5a7MFZiYSisqlG0IPi8K6CwLpI1anSzz6WObXrt4u20l
aYjS7DIO3mjBGY5uq4jM3tik9vIT5Vx9oLV6NVlKiuCVUqWio9+rWTysYcTUMqN4
TVxRBh+CymsM+CVJeh67hNG5mE/RUG+o9dxbSQk7Nn+EtwZFVTIaviouJdWBNsrs
ViZuS4ZHSHBEeGx3SXtq11e3jEwPCxe9N3IPTOAWdYIjwg==
-----END CERTIFICATE-----

</ca>


reneg-sec 0

verify-x509-name server name
```

34. We have used mutual authentication. So, for the client to connect to the server the client will require the client certificate. Now this client certificate is stored in our EC2 instance. So, in the next step here, what we need to do in the **.ovpn** file we downloaded from the Client VPN endpoint, is to add our client certificate and the client key.
35. Also, in the documentation if you scroll down, you will see how to add a certificate and key. But if you don't find it there then you can use the format below and add the certificate and key.

```
<cert>
Contents of client certificate (.crt) file
</cert>

<key>
Contents of private key (.key) file
</key>
```

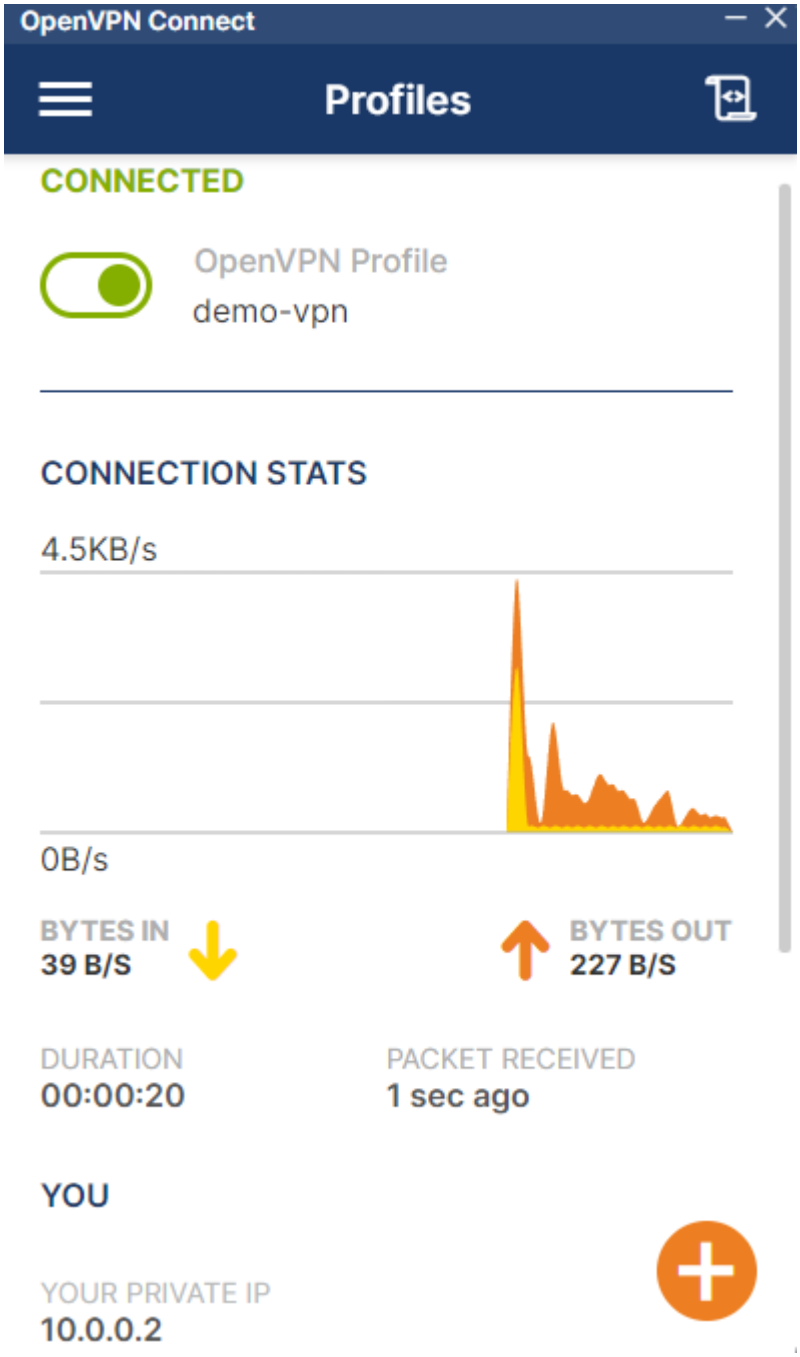
36. Now go back to your EC2 instance session and in the custom folder if you will do a listing of files in it. You can see your certificate and key there.

```
[root@ip-172-31-32-226 easyrsa3]# mkdir ~/custom_folder/
[root@ip-172-31-32-226 easyrsa3]# cp pki/ca.crt ~/custom_folder/
[root@ip-172-31-32-226 easyrsa3]# cp pki/issued/server.crt ~/custom_folder/
[root@ip-172-31-32-226 easyrsa3]# cp pki/private/server.key ~/custom_folder/
[root@ip-172-31-32-226 easyrsa3]# cp pki/issued/client1.domain.tld.crt ~/custom_folder
[root@ip-172-31-32-226 easyrsa3]# cp pki/private/client1.domain.tld.key ~/custom_folder/
[root@ip-172-31-32-226 easyrsa3]# cd ~/custom_folder/
[root@ip-172-31-32-226 custom_folder]# ls
ca.crt  client1.domain.tld.crt  client1.domain.tld.key  server.crt  server.key
[root@ip-172-31-32-226 custom_folder]#
```

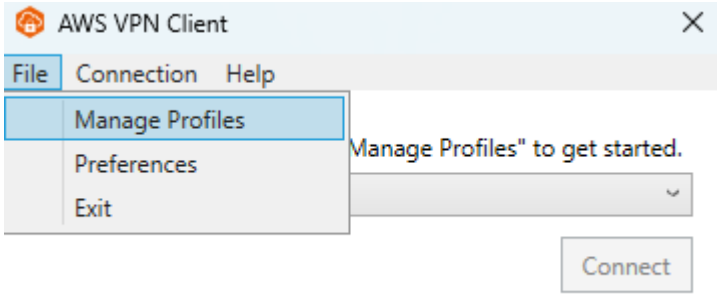
37. Also, you can use the **cat command** to view your certificate and key. Then copy them and paste them into the .opvn file using the above format.
38. After saving that you have to add one more thing in the .ovpn file you have to add a random name before cvpn as highlighted. Then just save your file.

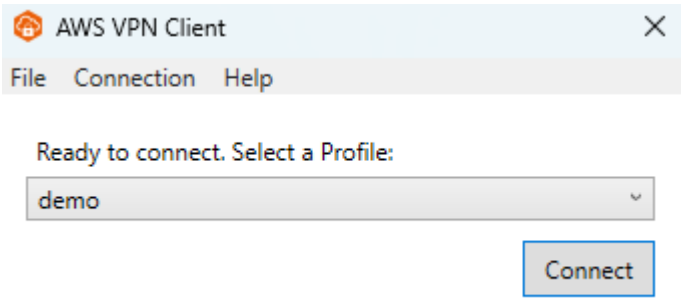
```
client
dev tun
proto udp
remote democert.cvpn-endpoint-09caff1c904eb30a0.prod.clientvpn.ap-south-1.amazonaws.com 443
remote-random-hostname
resolv-retry infinite
nobind
remote-cert-tls server
cipher AES-256-GCM
verb 3
```

39. Open your downloaded VPN and upload your file into it. You will see that it has been connected.
40. Similarly, you can do the same with the other VPN. First disconnect from this then open that.

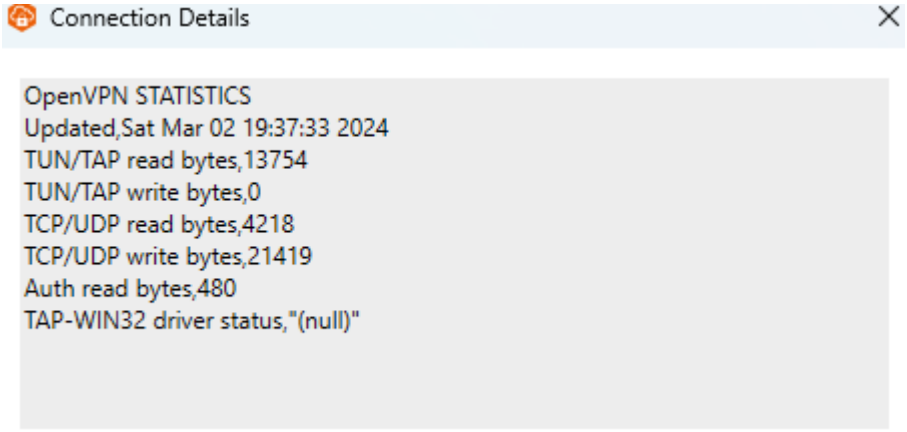


41. Now for the AWS VPN client first click on file then on manage profile, after that add profile, after that give a name then choose the ovpn file and click on connect.

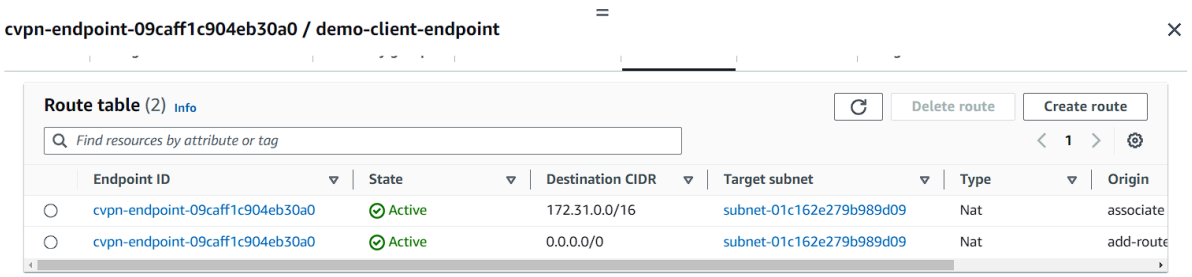




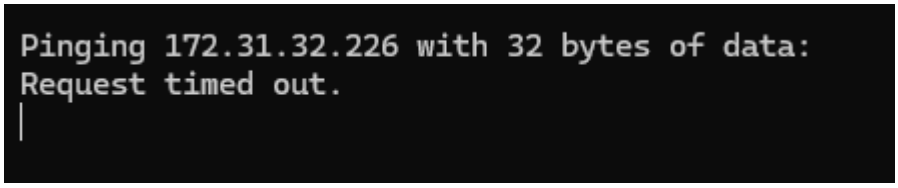
42. Below you can see that it has been connected.



43. Now it all depends on you which one you will prefer to use. One more thing, your internet may break because of the VPN.
44. For that you can do one thing go to the Client VPN endpoint in AWS Console. Then move to the route tables part in it. There you are going to create a route for 0.0.0.0/0
45. After that turn on your VPN again this time you won't get disconnected.



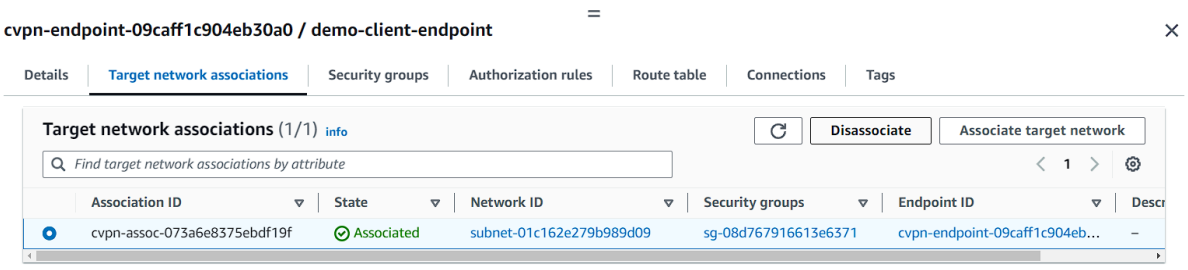
46. Now come back to EC2 and copy the private IP address of your instance and try to ping it from your local machine. If you get this type of request time out, then can go to the security group assigned to the instance and add All traffic from everywhere. Then again try to ping it.



```
Pinging 172.31.32.226 with 32 bytes of data:
Reply from 172.31.32.226: bytes=32 time=25ms TTL=126
Reply from 172.31.32.226: bytes=32 time=26ms TTL=126
Reply from 172.31.32.226: bytes=32 time=26ms TTL=126
Reply from 172.31.32.226: bytes=32 time=24ms TTL=126

Ping statistics for 172.31.32.226:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 24ms, Maximum = 26ms, Average = 25ms
```

Now for the cleanup, first disassociate your target network association from the VPN endpoint.



Then delete your VPN endpoint, after that delete your certificate and terminate your instance.