AWS VPN for access Kafka [Dev Account]

Ticket: **ECCO-991** - Getting issue details... STATUS

We want to explore an AWS VPN as a solution that could help alleviate the difficulties of performing a checkout locally. Therefore, the below steps explain how we can set a VPN to access a Kafka cluster (MKS).

Create Certificate

Since it is for our Dev account and to improve how we can perform a checkout locally, we could generate a certificate using the OpenVPN solution.

1. Clone repository

   ```
   git clone https://github.com/OpenVPN/easy-rsa.git
   ```
2. Generate a certificate with the below steps

```
#
# Go to the directory where you cloned the repository
#
cd easy-rsa
cd easyrsa3

#
# Init pki and build ca for the server and client
#
./easyrsa init-pki
./easyrsa build-ca nopass
./easyrsa build-server-full server nopass
./easyrsa build-client-full client1.domain.tld nopass

#
# Optional: Create a directory to keep the new certificates
#
mkdir ~/custom_easyrsa_folder

cp pki/ca.crt ~/custom_easyrsa_folder
cp pki/issued/server.crt ~/custom_easyrsa_folder

cp pki/private/server.key ~/custom_easyrsa_folder
cp pki/issued/client1.domain.tld.crt ~/custom_easyrsa_folder
cp pki/private/client1.domain.tld.key ~/custom_easyrsa_folder

#
# Go to the directory where you moved the certificates
#
cd ~/custom_easyrsa_folder
```

Note: we have a .crt and .key for the client and server.

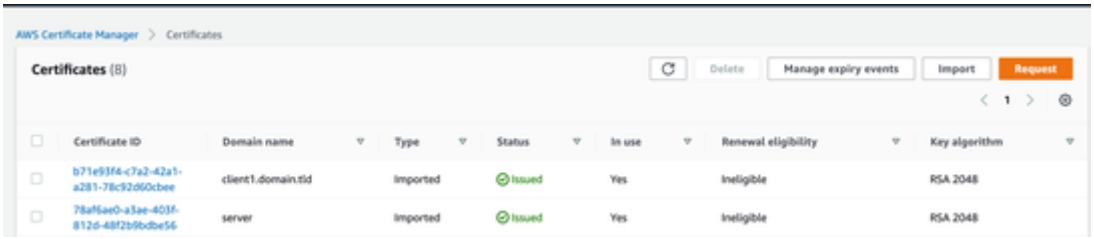
Import certificates to ACM

1. Import server certificate

```
aws acm import-certificate --certificate fileb://server.crt --
private-key fileb://server.key --certificate-chain fileb://ca.crt --
region us-east-1
```

2. Import client certificate

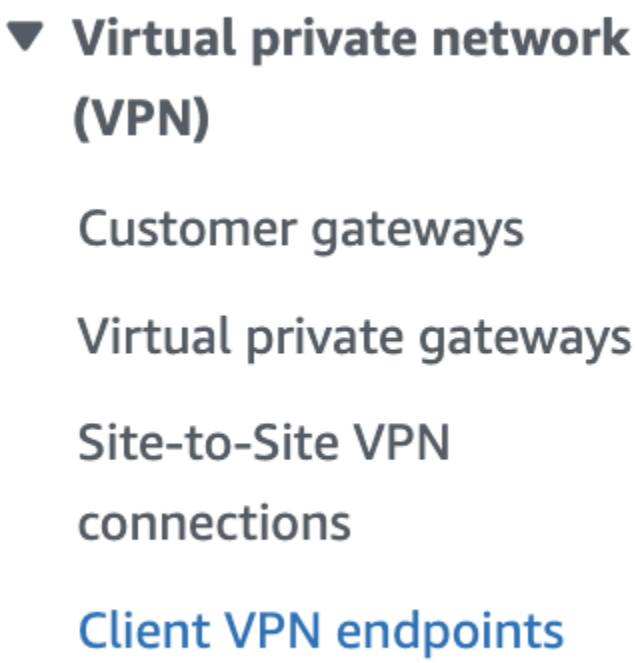
```
aws acm import-certificate --certificate fileb://client1.domain.tld.
crt --private-key fileb://client1.domain.tld.key --certificate-
chain fileb://ca.crt --region us-east-1
```



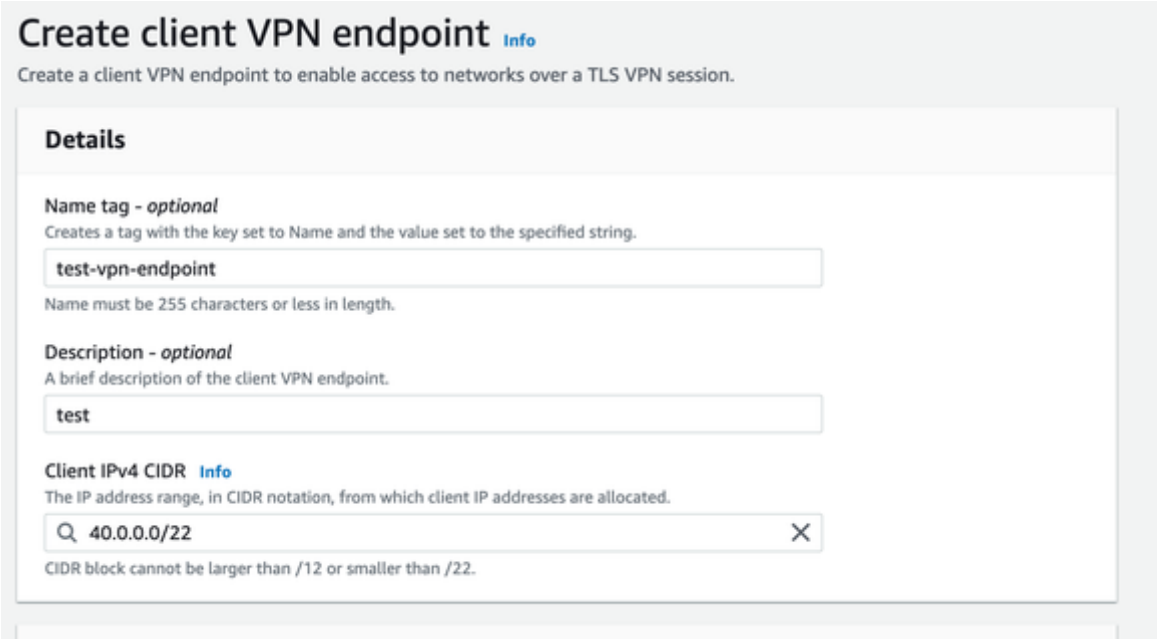
Set AWS VPN

Go to the VPC service to create the VPN

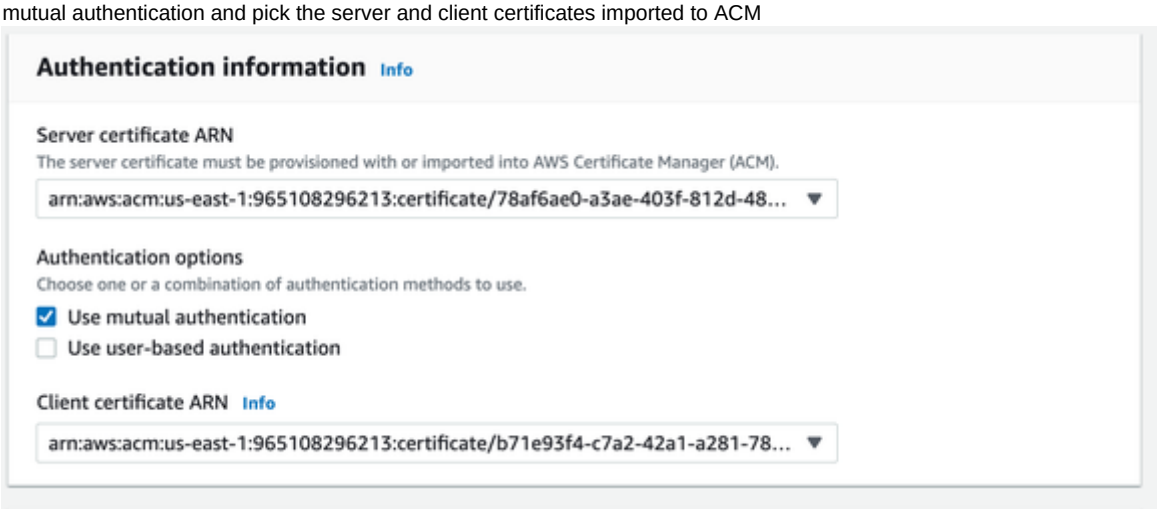
1. Go to the section Client VPN endpoints and click on create VPN endpoints



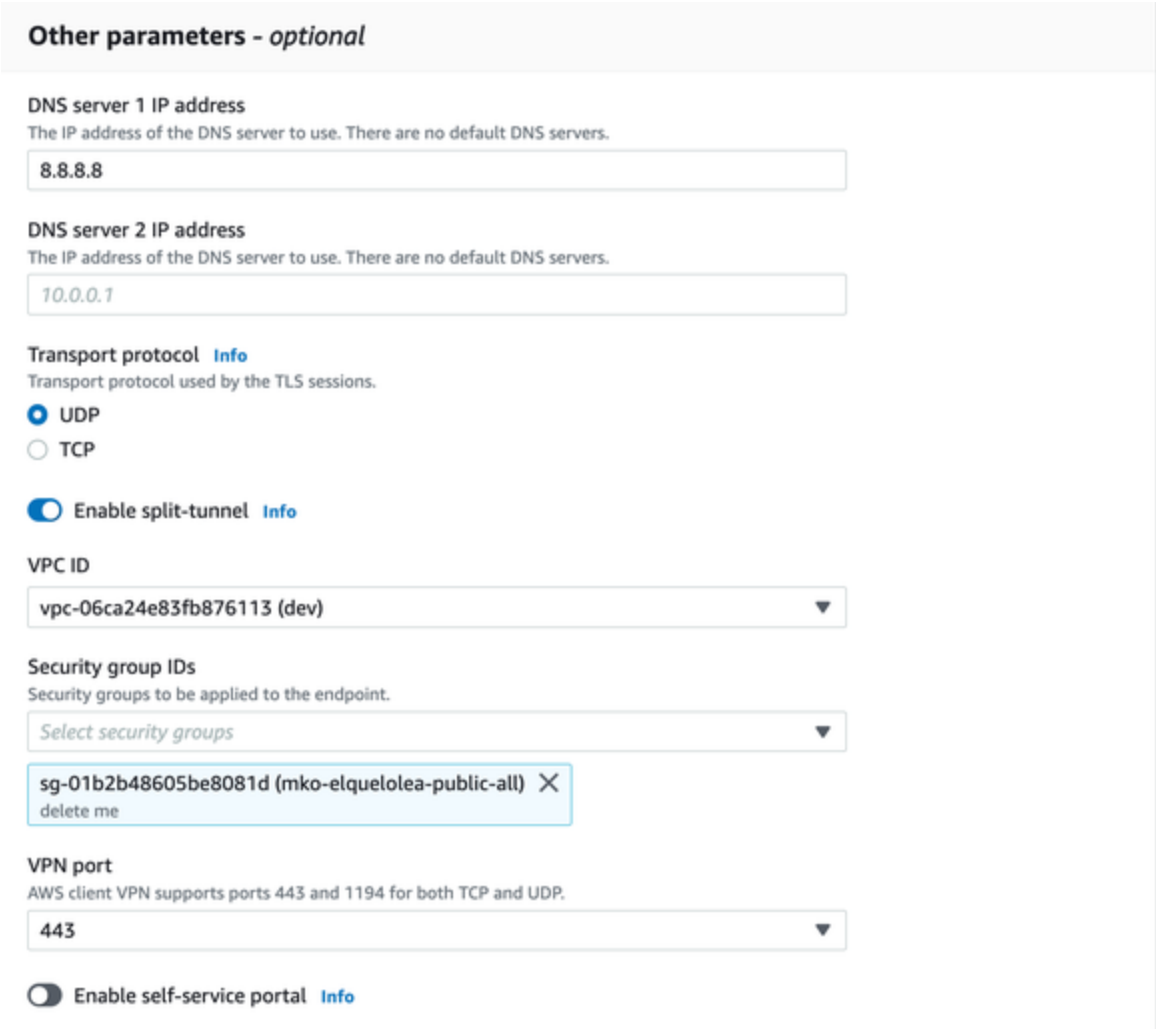
2. Set the CIDR value according to what you need
   Take a look at this web page: https://www.ipaddressguide.com/cidr



3. Authentication information
   Choose mutual authentication and pick the server and client certificates imported to ACM



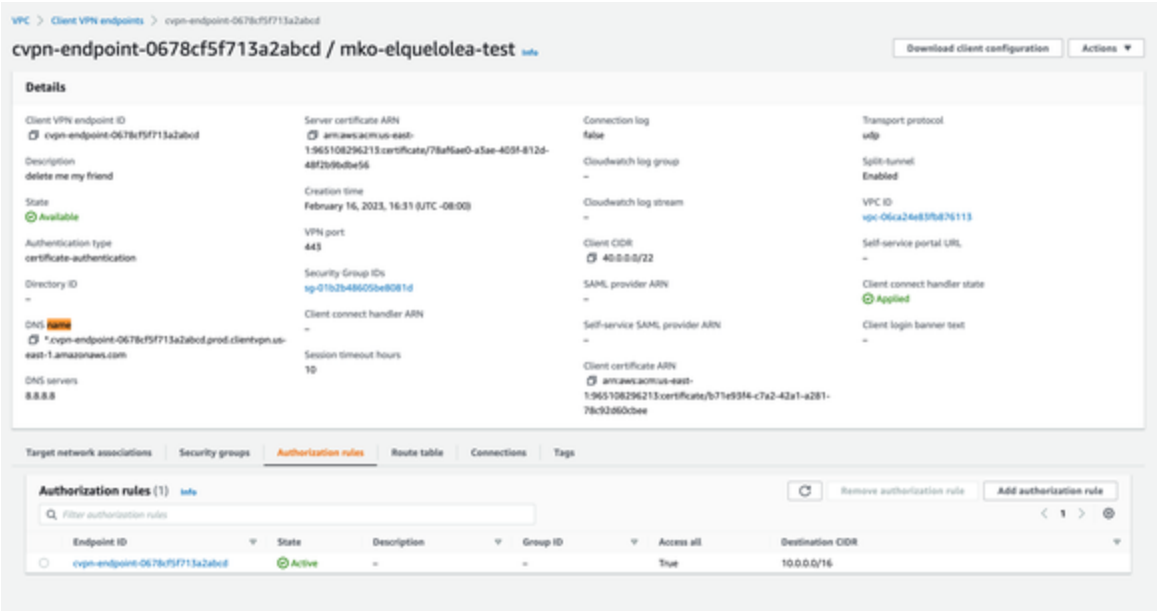
4. Not connection logging and Not client handler to avoid an extra cost
5. Other parameters
   f. Put the google DNS 8.8.8.8
   g. Check UDP option
   h. Choose the VPC ide where the Kafka cluster is running.
   i. Create a new Security group through the EC2 service
      i. Select the target network association (here, we choose the networks from where you will allow the connection)
      ii. validate if we want to enable anywhere connections or whatever you want :)



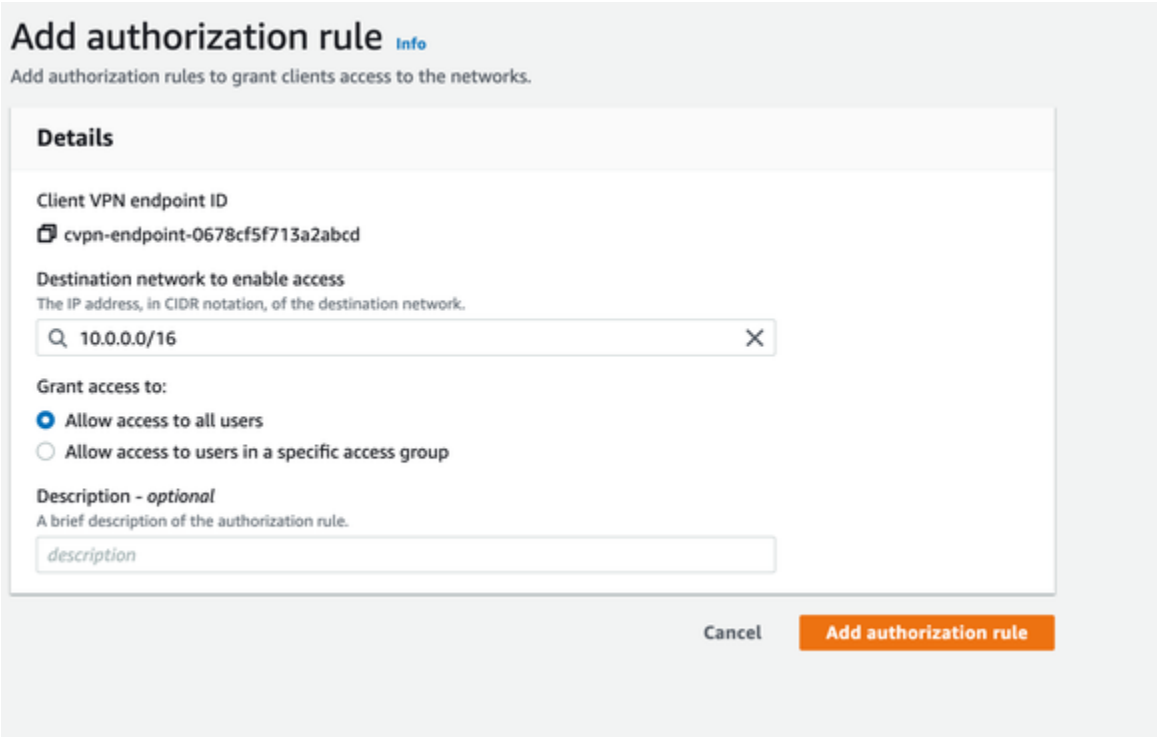
6. Add some tags and click on create VPN

Set authorization rule

1. Go to client VPN endpoints again and choose the new VPN
2. Click on the authorizations rule in the bottom section

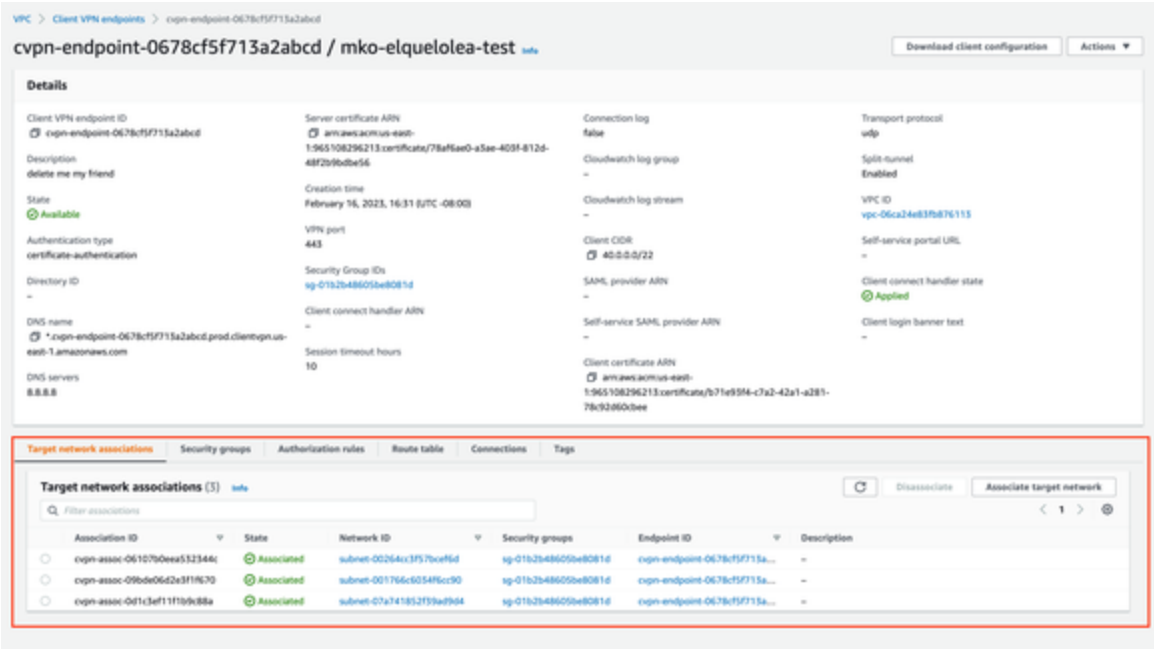


3. Add the rule to the destination



Add target network associations

1. It is the step where we need to add all the subnets associated with the Kafka cluster (you can get them by looking at the MKS service)

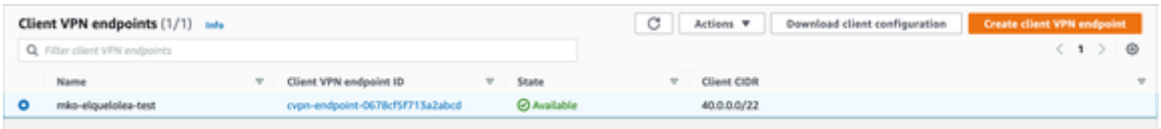


Configure AWS VPN certificate (This step is required only once when the VPN is new)

> This step is just needed if we don't have a certificate configured to use the VPN as a client. You could skip this step and use the certificate in the next step.

1. Download the client configuration file from the VPC service -> client VPN endpoint

   Button: `Download client configuration`



2. Go to the directory where you downloaded the certificate and open the file.
   e.g. `nano downloaded-client-config.ovpn`
3. Add the client cat and key created in the first step of the process
   e.g.
   a. Go to the directory where you have the certificates `custom_easyrsa_folder`
   b. Open the certificate crt `client1.domain.tld.crt`
   c. Copy the certificate into the .ovpn file between the tags `<cert> </cert>`

```
<cert>
-----BEGIN CERTIFICATE-----
HERE
-----END CERTIFICATE-----
</cert>
```

   d. Open the key file `client1.domain.tld.key`
   e. Copy `client1.domain.tld.key` -> the key into the `.ovpn` file between tags `<key> </key>`

```
<key>
-----BEGIN PRIVATE KEY-----
here
-----END PRIVATE KEY-----
</key>
```

4. Now open `.ovpn` file again and modify the property `remote`, adding a subfix value; in this case, I added the value `article`.

```
client
dev tun
proto udp
remote article.cvpn-endpoint-0678cf5f713a2abcd.prod.clientvpn.us-east-1.amazonaws.com 443
remote-random-hostname
resolv-retry infinite
nobind
remote-cert-tls server
cipher AES-256-GCM
verb 3
<ca>
```

Import the VPN certificate to your Tunnelblick

As we already have a VPN, you can use this certificate to test it.

Once you import the certificate and connect to the VPN, you can execute the below command from your terminal to ensure it works properly.

```
netstat -r
```

```
➜  k8s git:(feature/BXC-723-add-tiltfile-kafka-local-env) ✗ netstat -r
Routing tables

Internet:
Destination        Gateway            Flags        Netif Expire
default            192.168.0.1        UGScg          en0
10/16              40.0.0.129         UGSc         utun11
40.0.0.128/27      40.0.0.130         UGSc         utun11
40.0.0.130         40.0.0.130         UH           utun11
127                localhost          UCS            lo0
localhost          localhost          UH             lo0
```

```
telnet z-3.devkafka.rwtj5d.c16.kafka.us-east-1.amazonaws.com 2182
```

```
➜  k8s git:(feature/BXC-723-add-tiltfile-kafka-local-env) ✗ telnet z-3.devkafka.rwtj5d.c16.kafka.us-east-1.amazonaws.com 2182
Trying 10.0.113.204...
Connected to z-3.devkafka.rwtj5d.c16.kafka.us-east-1.amazonaws.com.
Escape character is '^]'.
```

Next steps

- Investigate how we can authenticate and authorize users in a better way
- Work on this spike ECCO-1131 - Getting issue details... STATUS
  - https://support.jumpcloud.com/support/s/article/Single-Sign-On-SSO-with-AWS-Client-VPN
  - AWS approach
  - https://www.youtube.com/watch?v=l4TpB_MV8tI&list=PL8K2AhiVhR195HTta5rYctdS_PH61PuXM&index=1

Authenticate AWS Client VPN users with federated authentication.

We implemented the approach recommended on this page, Authenticate AWS Client VPN users with AWS IAM Identity Center, since it avoids having a dependency on the IT team for the configuration and at the same time, we can take advantage of the federated authentication configured on the root account (**Engineering Production**).

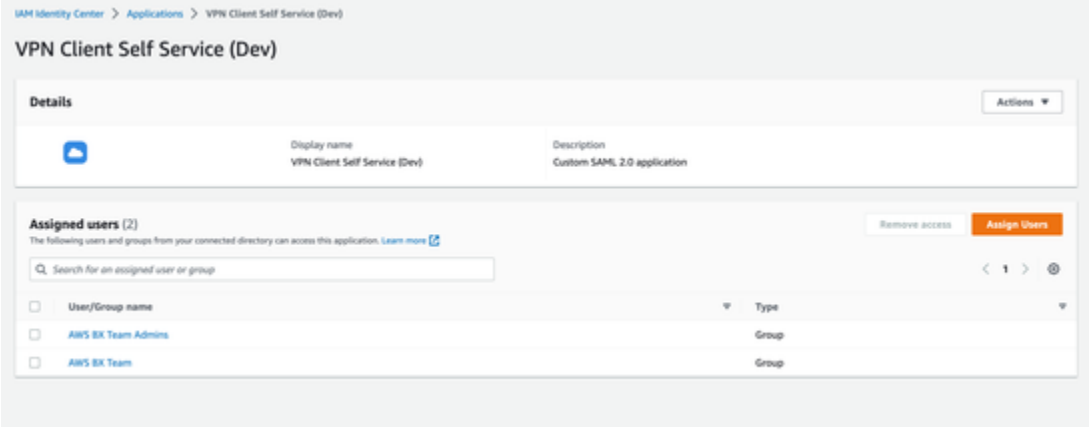
We created two SAML applications:

- AWS VPN (dev)
  - This allows defining the group of users enabled to use the VPN
  - The metadata generated by this application is used to set up an identity provider in the dev account (**Engineering Dev**)
  - Identity provider
- VPN Client Self Service (Dev)
  - This allows defining the group of users enabled to download the VPN configuration (.openvpn file)
  - The metadata generated by this application is used to set up an identity provider in the dev account (**Engineering Dev**)
  - Identity provider

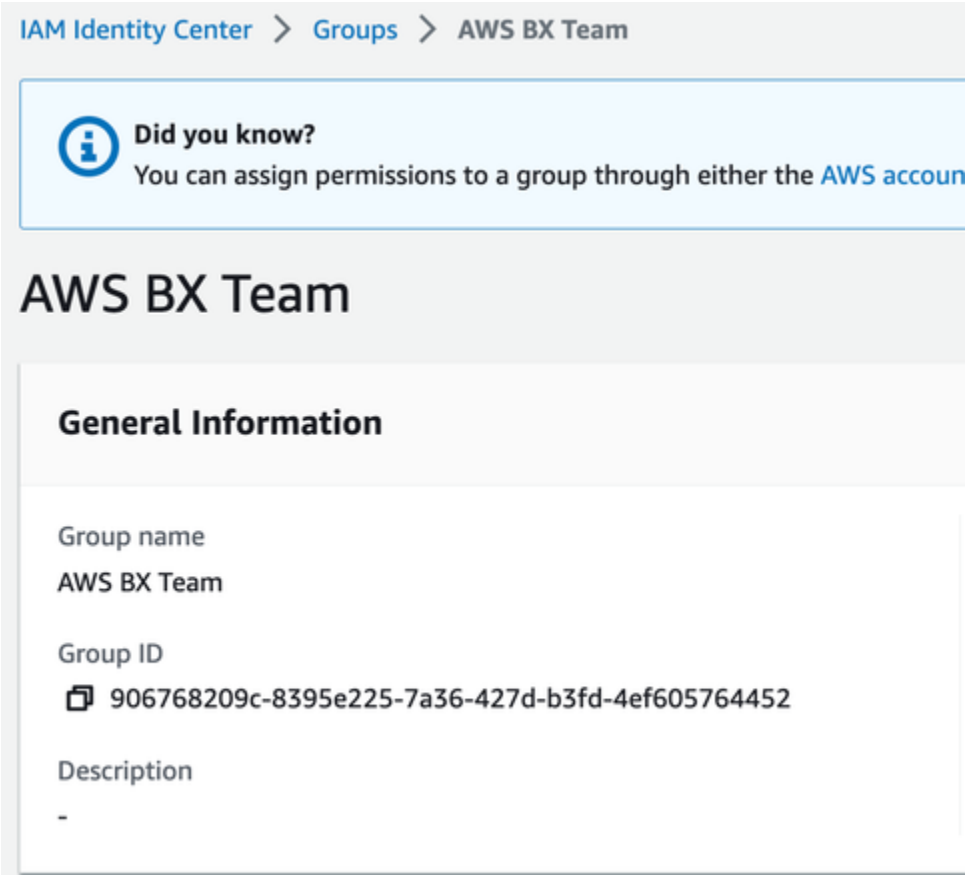
The VPN uses the identity providers created previously, and we set them when creating the VPN. e.g.

How to add or remove user groups to the VPN

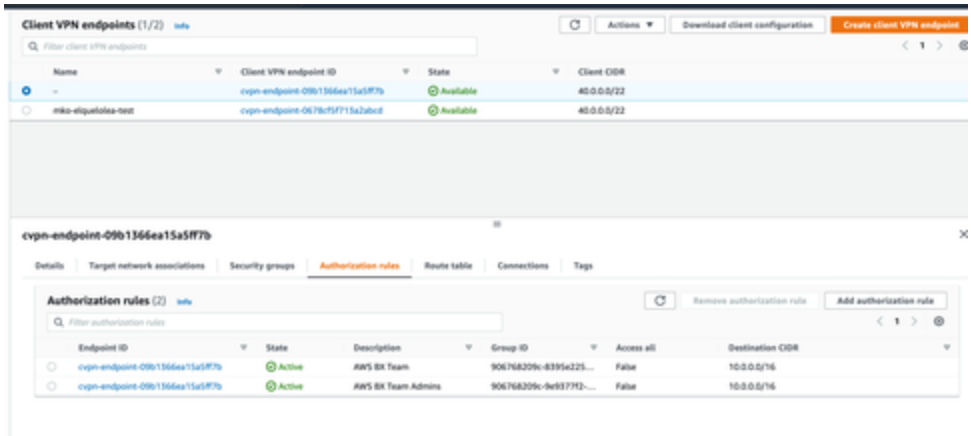
1. Go to the root account (**Engineering Production**), where the Identity center is enabled.
2. Select the applications AWS VPN (dev) and VPN Client Self Service (Dev)
   a. We have to make this change for both applications.
3. Click on **Assign Users**

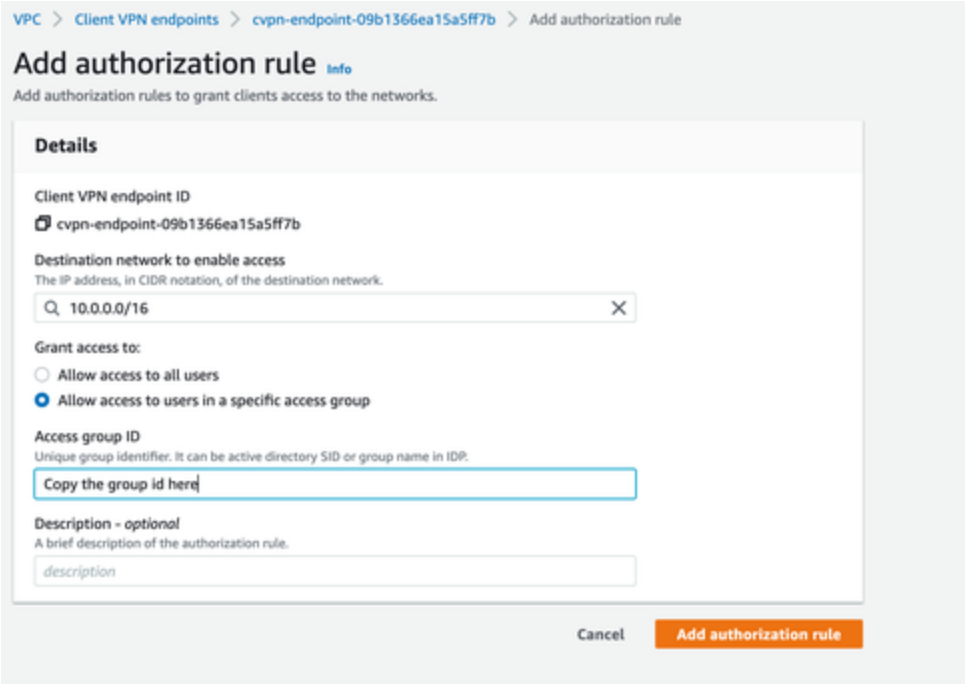


4. Select the new group that you want to add
5. Go to the **groups** option in the Identity center and copy the group id



6. Open the VPN created in the Dev account
7. Select the VPN and add the authorization rule for the new group


Client VPN endpoints (1/2) Info
Actions
Download client configuration
Create client VPN endpoint
Filter client VPN endpoints
Name
Client VPN endpoint ID
State
Client CIDR
-
cvpn-endpoint-09b1366ea15a5ff7b
Available
40.0.0.0/22
mko-eliquelolea-test
cvpn-endpoint-0678cf5f715a2abcd
Available
40.0.0.0/32
cvpn-endpoint-09b1366ea15a5ff7b
Details
Target network associations
Security groups
Authorization rules
Route table
Connections
Tags
Authorization rules (2) Info
Remove authorization rule
Add authorization rule
Filter authorization rules
Endpoint ID
State
Description
Group ID
Access all
Destination CIDR
cvpn-endpoint-09b1366ea15a5ff7b
Active
AWS BX Team
906768209c-8395e225...
False
10.0.0.0/16
cvpn-endpoint-09b1366ea15a5ff7b
Active
AWS BX Team Admins
906768209c-9e937ff2-...
False
10.0.0.0/16


VPC > Client VPN endpoints > cvpn-endpoint-09b1366ea15a5ff7b > Add authorization rule
Add authorization rule Info
Add authorization rules to grant clients access to the networks.
Details
Client VPN endpoint ID
cvpn-endpoint-09b1366ea15a5ff7b
Destination network to enable access
The IP address, in CIDR notation, of the destination network.
10.0.0.0/16
Grant access to:
Allow access to all users
Allow access to users in a specific access group
Access group ID
Unique group identifier. It can be active directory SID or group name in IDP.
Copy the group id here
Description - optional
A brief description of the authorization rule.
description
Cancel
Add authorization rule