(AWS) File sharing SFTP server and connection via vpn only

Create a VPC with a /16 cidr block

aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications 'ResourceType=vpc,Tags=[{Key=Name,Value=sftp-vpc}]'

Enable DNS hostnames

aws ec2 modify-vpc-attribute --vpc-id <vpc-id> --enable-dns-hostnames

or in GUI, Go to edit VPC

A screenshot of a computer

Description automatically generated

Create 2 subnets, 1 for public use as VPN and 1 private for SFTP server

# Create private subnet for SFTP server

aws ec2 create-subnet --vpc-id <vpc-id> --cidr-block 10.0.1.0/24 --availability-zone us-east-1a

A screenshot of a computer

Description automatically generated

# Create public subnet for VPN server

aws ec2 create-subnet --vpc-id <vpc-id> --cidr-block 10.0.2.0/24 --availability-zone us-east-1aA screenshot of a computer

Description automatically generated

Create internet gateway for the public subnet

A screenshot of a computer

Description automatically generated

Attach to your VPC first by clicking the internet gateway > Actions> Attach to VPC



Create route table

A screenshot of a computer

Description automatically generated

Associate with the public subnet

A screenshot of a computer

Description automatically generated

Create a route for 0.0.0.0/0 to your created internet gateway

A screenshot of a computer

Description automatically generated

Create nat gateway inside your public subnet, click the allocate elastic ip

A screenshot of a computer

Description automatically generated

Create another route table, to associate with your private subnet

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

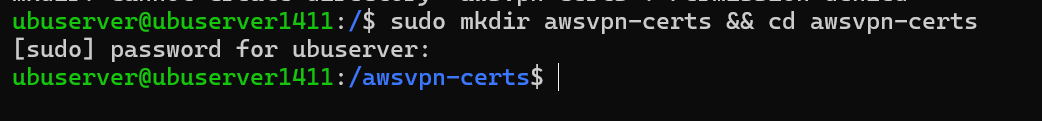
Now create a route for the nat gateway in the route table associated with your private subnet A screenshot of a computer

Description automatically generated

Next, setup client VPN endpoint

As we cannot request for private certificate from AWS cert manager, we can create using easyrsa

Create a folder for your certs then git clone easyrsa there. For this case, we use a ubuntu server VM as the cert server



Run git clone git clone https://github.com/OpenVPN/easy-rsa.git

A computer screen shot of a black background

Description automatically generated

Navigate to easyrsa3 folder

Initialize the pki using sudo

A screen shot of a computer

Description automatically generated

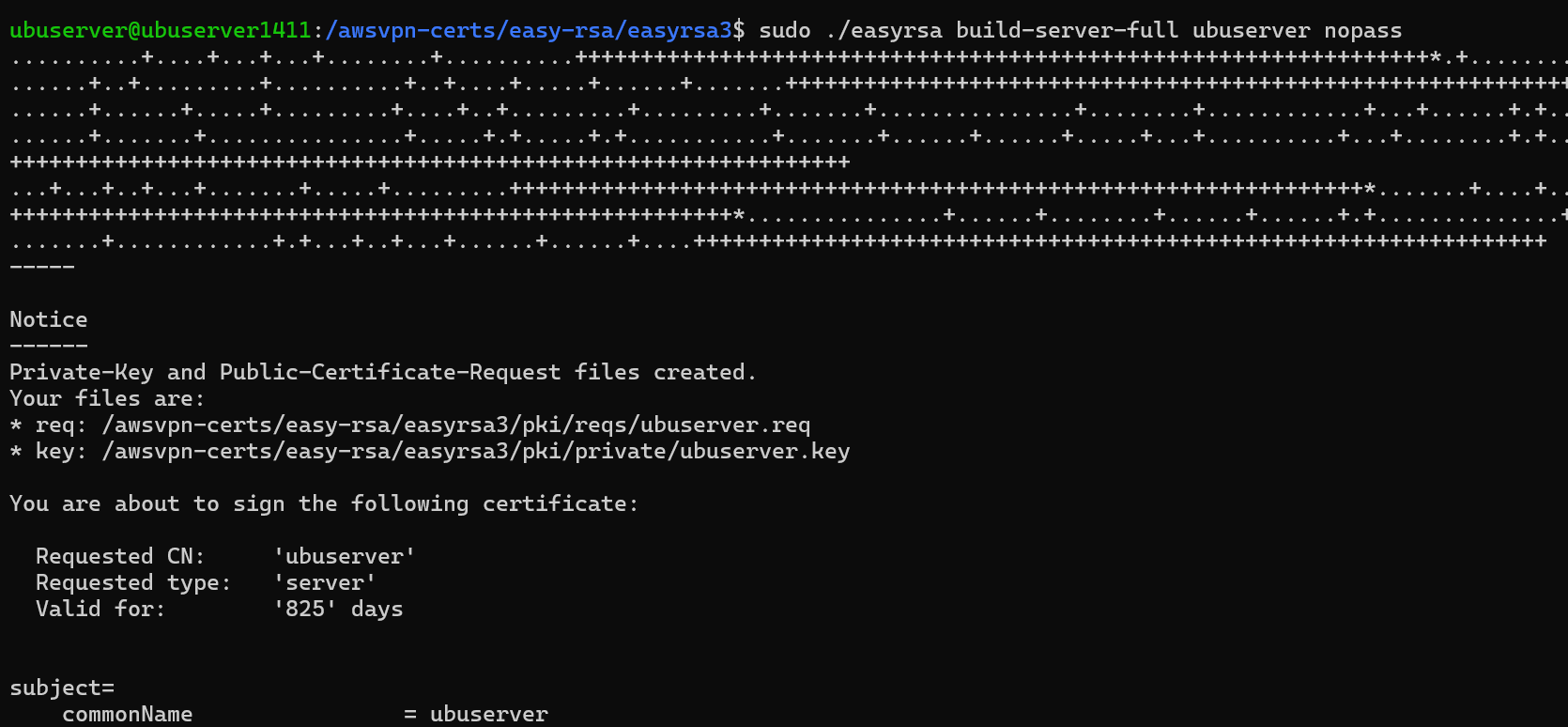
Create the Certificate Authority

A computer screen with white text

Description automatically generated

**Generate server cert and key**

Sudo ./easyrsa build-server-full server nopass (you can change the server to any name you want so e.g. ./easyrsa build-server-full ubuserver nopass)



**Generate the client cert and key**

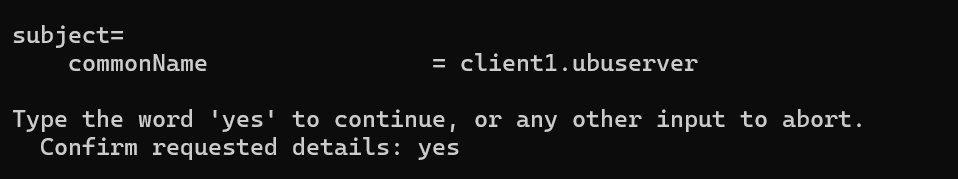
Sudo ./easyrsa build-client-full client1.domain.tld nopass

Can change domain to your hostname so client1.ubuserver nopass

A screenshot of a computer screen

Description automatically generated

If you happen to key in the wrong name, at this prompt where they ask for confirmation you can press any other key to abort the operation so that the keys won’t be created.

Then run the same command again with the desired name to create the keys.

The certificates and keys will be in these locations:

* CA Certificate: pki/ca.crt
* Server Certificate: pki/issued/ubuserver.crt
* Server Key: pki/private/ubuserver.key
* Client Certificate: pki/issued/client1.ubuserver.crt
* Client Key: pki/private/client1.ubuserver.key

Now go back to AWS Certificate Manager and select Import certificate

A screenshot of a certificate

Description automatically generated

For the Server certificate:

* Copy contents of ubuserver.crt into "Certificate body" (at pki/issued) – only copy from begin cert to end cert section for all 3.
* Copy contents of ubuserver.key into "Certificate private key" (at pki/private)
* Copy contents of ca.crt into "Certificate chain" (go to pki folder)
* Make sure for all 3 fields pasted have the header footer included





Repeat for the client certificate

* Copy contents of client1.ubuserver.crt into "Certificate body" (at pki/issued) – only copy from begin cert to end cert section for all 3.
* Copy contents of client1.ubuserver.key into "Certificate private key" (at pki/private)
* Copy contents of ca.crt into "Certificate chain" (go to pki folder)

To test if certs are valid, can run this command after you are in the easyrsa3 directory

openssl x509 -in pki/issued/ubuserver.crt -text -noout | grep "X509v3 Extended Key Usage" -A1

Type in the password for your account and showing this means its ok:

A screen shot of a computer

Description automatically generated

Import the server cert then import the client cert, now that you see the domain name after importing you can proceed to the next step of creating VPN endpoint. If you don’t see the domain name you can delete both certs and try recreating new keys from sudo ./easyrsa build-server-full vpn-server.ubuserver nopass (replace vpn-server.ubuserver with your preferred domain name)

A screenshot of a computer

Description automatically generated

Now, head to your VPC and click on client VPN endpoint then create client VPN endpoint (use a cidr range that don’t conflict with your VPC cidr). In this case as our VPC cidr is 10.0.0.0/16 we can use another private ip range which is 172.16.0.0/22. /22 will give 1022 addresses.

A screenshot of a computer

Description automatically generated

For authentication select mutual authentication and select your server cert for Server certificate ARN and client cert for the client certificate ARN

A screenshot of a computer

Description automatically generated

A screenshot of a certificate

Description automatically generated

Keep split tunnel enabled and port as default (443)

Eventually the 3 setups to take note of:

A screenshot of a computer

Description automatically generated

Now we can associate to our SFTP VPC’s private subnet. Select your vpc endpoint that you created, select Target network associations and select associate target network

A screenshot of a computer

Description automatically generated

Choose your VPC and the private subnet in it and click on associate target network

A screenshot of a computer

Description automatically generated

Now head to authorization rules and add authorization rules

A screenshot of a computer

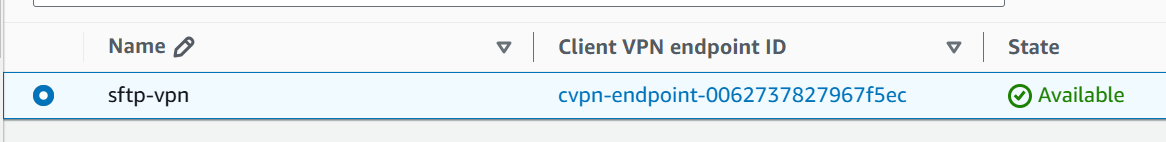
Description automatically generated

Destination network key in your vpc cidr and then add this rule

A screenshot of a computer

Description automatically generated

Once your VPN endpoint has been associated already it will show available and we can proceed with the next step of creating the SFTP server which will be hosted in AWS as well



**Creating our SFTP server**

Head to your VPC and create security group

A screenshot of a computer

Description automatically generated

Add inbound rule for your VPN ip range

A screenshot of a computer

Description automatically generated

Outbound leave as default

A screenshot of a computer

Description automatically generated

Now, launch an EC2 instance

Name: up to you

Select the Amaon Linux2023 image

Instance type: t2.micro

Create a new keypair

A screenshot of a computer

Description automatically generated

It will auto download the file. You can save somewhere securely in your local drive

A blue rectangle with black text

Description automatically generated

For network settings, select your sftp vpc, the private subnet and disable auto assign IP

A screenshot of a computer

Description automatically generated

For firewall select the security group you created for your vpn

A screenshot of a computer

Description automatically generated

Scroll down to the storage part and can leave EBS as default 8GB or increase it. But we will be using EFS anyway, so can scroll down and select create new shared file system

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The 2 check boxes means:

* Automatically create and attach security groups (this handles the EFS security automatically)
* Automatically mount shared file system (this will install efs-utils and handle mounting)

Next part, we will need to SSH to your SFTP server to create the users.

First, add a route to your private subnet route table so that you can VPN in. the destination will be your vpn network cidr and the target you can select the network interface associated with your vpn client

To find the network interface id, can click on your EC2 > select network interfaces under Network & Security and see which one matches the description

A white screen with black text

Description automatically generated

A search box with a number and a magnifying glass

Description automatically generated

Your route table should now look something like this:

* 10.0.0.0/16 → local (VPC)
* 0.0.0.0/0 → nat-gateway-id (for internet access)
* 172.16.0.0/22 → vpn-endpoint-id (for VPN access)

Head to your VPC and select client VPN end points, select the VPN endpoint you created earlier and download the client configuration

A screenshot of a computer

Description automatically generated

Next, head to <https://aws.amazon.com/vpn/client-vpn-download/> and download the AWS Client VPN for desktop. Run the installer with default settings.

Edit the downloaded ovpn client file and add in the following after the </ca> and before the reneg-sec 0

To get the cert and key:

sudo cat /awsvpn-certs/easy-rsa/easyrsa3/pki/issued/client1.ubuserver.crt

sudo cat /awsvpn-certs/easy-rsa/easyrsa3/pki/private/client1.ubuserver.key

Add these:

<cert>

-----BEGIN CERTIFICATE-----

[Contents of client1.ubuserver.crt]

-----END CERTIFICATE-----

</cert>

<key>

-----BEGIN PRIVATE KEY-----

[Contents of client1.ubuserver.key]

-----END PRIVATE KEY-----

</key>

Next, open the downloaded AWS VPN Client

Go to file > manage profiles

A screenshot of a computer

Description automatically generated

Add profile

A screenshot of a computer

Description automatically generated

Import the edited ovpn file and add profile

A screenshot of a computer error

Description automatically generated

Now you can ssh into your ec2 using ssh -i your-key.pm ec2-user@your-ec2-private-ip via the vpn profile. Normal ssh would time out in this scenario

Click on connect

A screenshot of a computer

Description automatically generated