Using ***AWS Cloud Formation*** to create a simple client-sid VPN.

So what is cloud Formation actually?

AWS cloud Formation is one of the many services available in aws to streamline user interaction with various other services within the AWS ecosystem. It helps the client to provision, set-up, configure the resources(Ex: EC2 instance, RDB instance etc..) and policies attached automatically with ease from a pre-defined template that defined all the said specifications, so that the client can focus on actually focus on maintaining and improving the resources than actually spend resources and time on deploying them individually.

Understanding the use-case:

For a scalable web application with a backend database, you typically use an Auto Scaling group, Elastic Load Balancing, and Amazon RDS. Manually provisioning and configuring these resources can be complex and time-consuming.

Instead, you can use a CloudFormation template to streamline this process. A template defines all your resources and their properties. When you create a CloudFormation stack from the template, CloudFormation provisions and configures the Auto Scaling group, load balancer, and database for you. This approach allows you to manage your resources as a single unit, simplifying deployment and maintenance. You can also easily delete the stack which removes all associated resources. CloudFormation enables quick replication and efficient management of your infrastructure.

How cloud formation works?

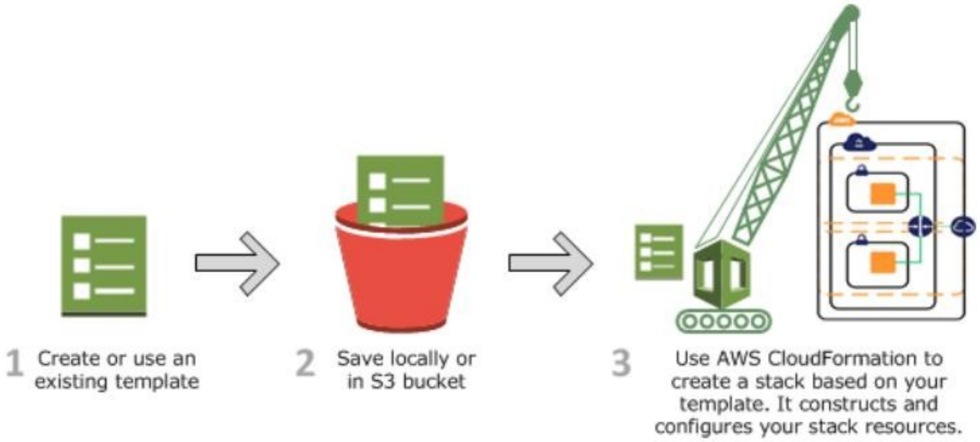
Cloud formation consists of two main key concepts:

. Templates

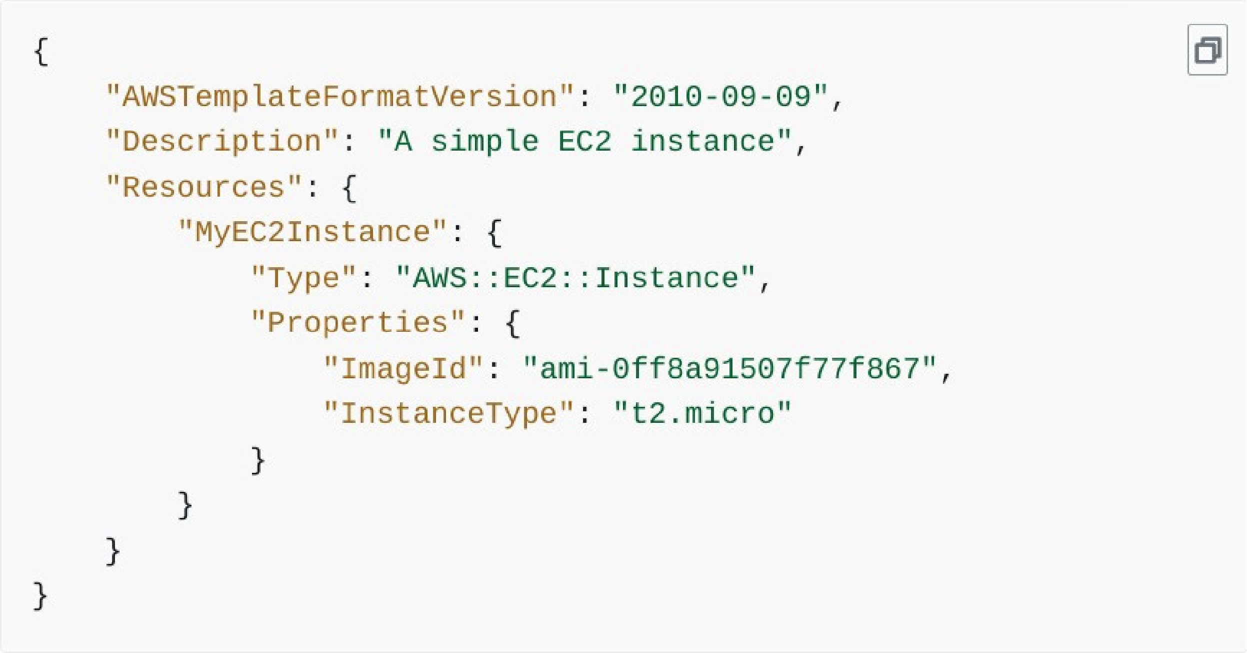
. Stacks

When creating a stack, CloudFormation makes service calls to AWS to provision and configure resources, acting within your permission limits. For instance, to create or terminate EC2 instances, you need corresponding permissions, managed via AWS IAM.

The template defines all resource actions. For example, a template specifying a t2.micro EC2 instance prompts CloudFormation to call the EC2 create instance API with that instance type.


1 Create or use an existing template
2 Save locally or in S3 bucket
3 Use AWS CloudFormation to create a stack based on your template. It constructs and configures your stack resources.

Example of how a JSON template looks like :



```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "A simple EC2 instance",
    "Resources": {
        "MyEC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId": "ami-0ff8a91507f77f867",
                "InstanceType": "t2.micro"
            }
        }
    }
}
```

Explantion:

{

"AWSTemplateFormatVersion": "2010-09-09",

. This line specifies the version of the AWS CloudFormation template format you are using "2010-09-09" is the latest version as of this writing.

"Description": "A simple EC2 instance",

. This line provides a brief description of what the CloudFormation stack will do. In this case, it describes the creation of a simple EC2 instance.

"Resources": {

This line begins the section where you define the AWS resources that CloudFormation will create

"MyEC2Instance": {

. This line specifies a logical name for the resource within the template. "MyEC2Instance" is a unique identifier for this EC2 instance in the template.

"Type": "AWS::EC2::Instance",

. This line defines the type of AWS resource to create. Here, "AWS::EC2::Instance" specifies that the resource is an EC2 instance.

"Properties": {

. This line starts the section where you define properties for the EC2 instance.

"ImageId": "ami-0ff8a91507f77f867",

. This line specifies the Amazon Machine Image (AMI) ID that the EC2 instance will use. Th AMI ID "ami-0ff8a91507f77f867" refers to a specific Amazon Linux AMI.

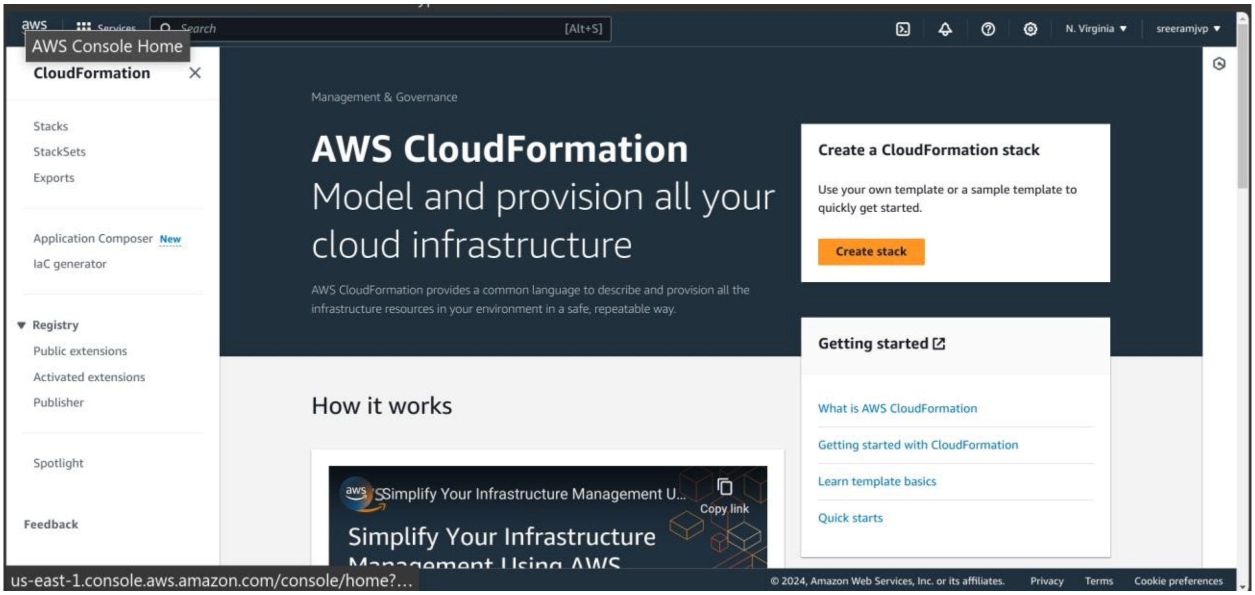
"InstanceType": "t2. micro"

. This line specifies the instance type for the EC2 instance. "t2.micro" is a small instance type suitable for low-traffic applications and testing.

Creating a client side vpn from pre-defined template:

(Step-1):

Home page of AWS cloud formation:



Management & Governance

AWS CloudFormation

Model and provision all your cloud infrastructure

AWS CloudFormation provides a common language to describe and provision all the infrastructure resources in your environment in a safe, repeatable way.

Create a CloudFormation stack

Use your own template or a sample template to quickly get started.

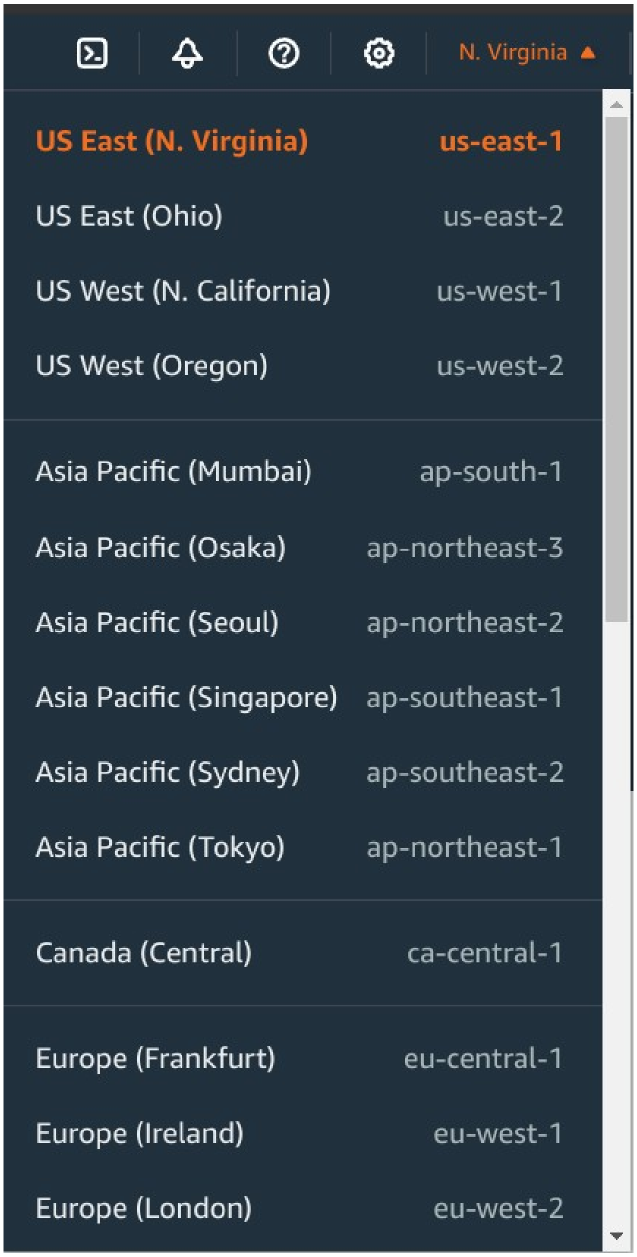
How it works

us-east-1.console.aws.amazon.com/console/home?...

© 2024, Amazon Web Services, Inc. or its affiliates.

(Step-2)

Region selection:



US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2

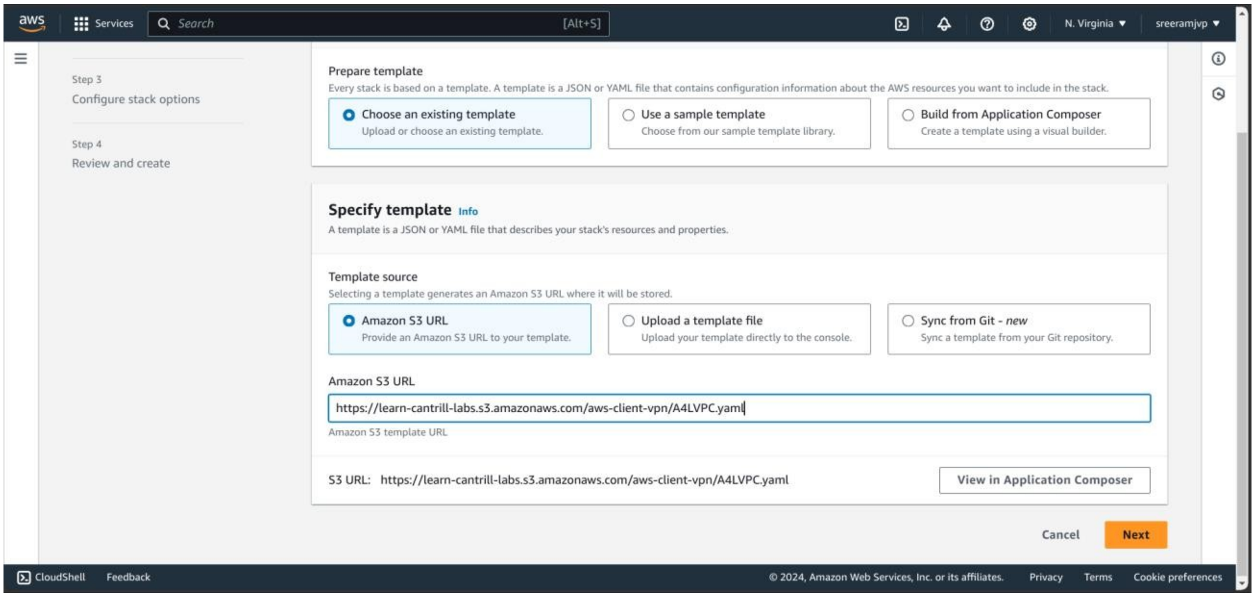
(I will be selecting us-east-1 region)

(Step-3):

Stacks are created using *Templates,* which can be done in three ways:

1. Using a pre-defined template or
2. Create a scratch template using *application composer*

We will be using a pre-defined template, loaded from an amazon S3 URL.*(s3, stands for simple storage service).*



Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

- ● Choose an existing template — Upload or choose an existing template.
- ○ Use a sample template — Choose from our sample template library.
- ○ Build from Application Composer — Create a template using a visual builder.

Specify template Info

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

- ● Amazon S3 URL — Provide an Amazon S3 URL to your template.
- ○ Upload a template file — Upload your template directly to the console.
- ○ Sync from Git - *new* — Sync a template from your Git repository.

Amazon S3 URL: https://learn-cantrill-labs.s3.amazonaws.com/aws-client-vpn/A4LVPC.yaml

Amazon S3 template URL

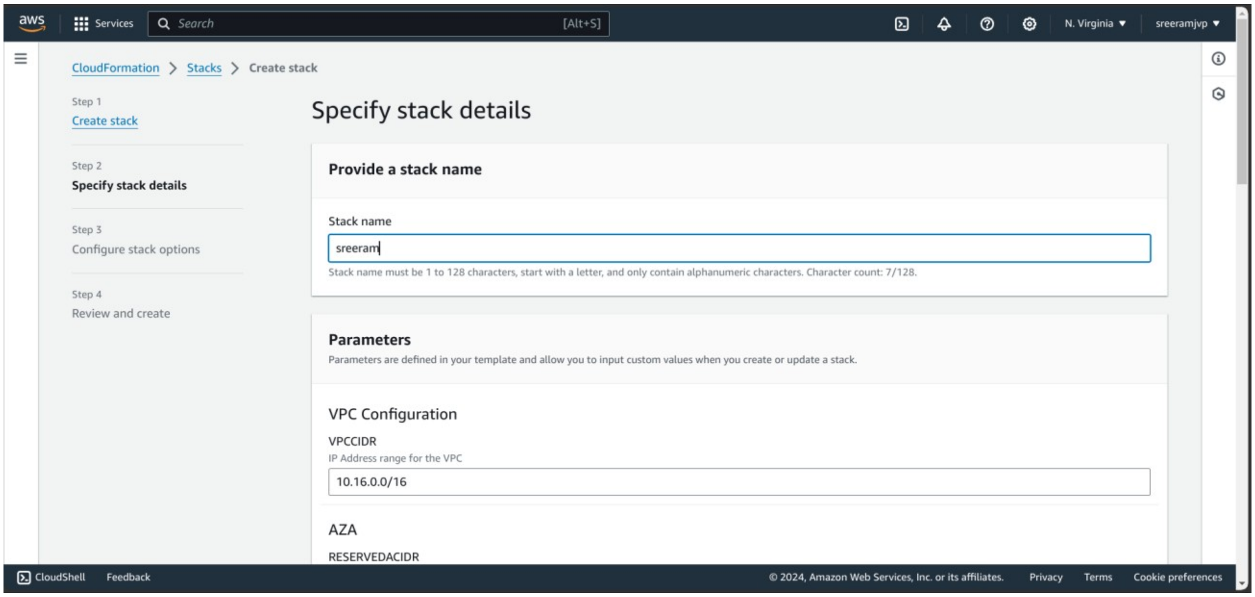
S3 URL: https://learn-cantrill-labs.s3.amazonaws.com/aws-client-vpn/A4LVPC.yaml

View in Application Composer
Cancel
Next
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

The template is loaded from https://learn-cantrill-labs.s3.amazonaws.com/aws-client-vpn/A4LVPC.yaml.

(Step-4):

Click next.



Specify stack details

Provide a stack name

Stack name: sreeram

Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 7/128.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

VPC Configuration

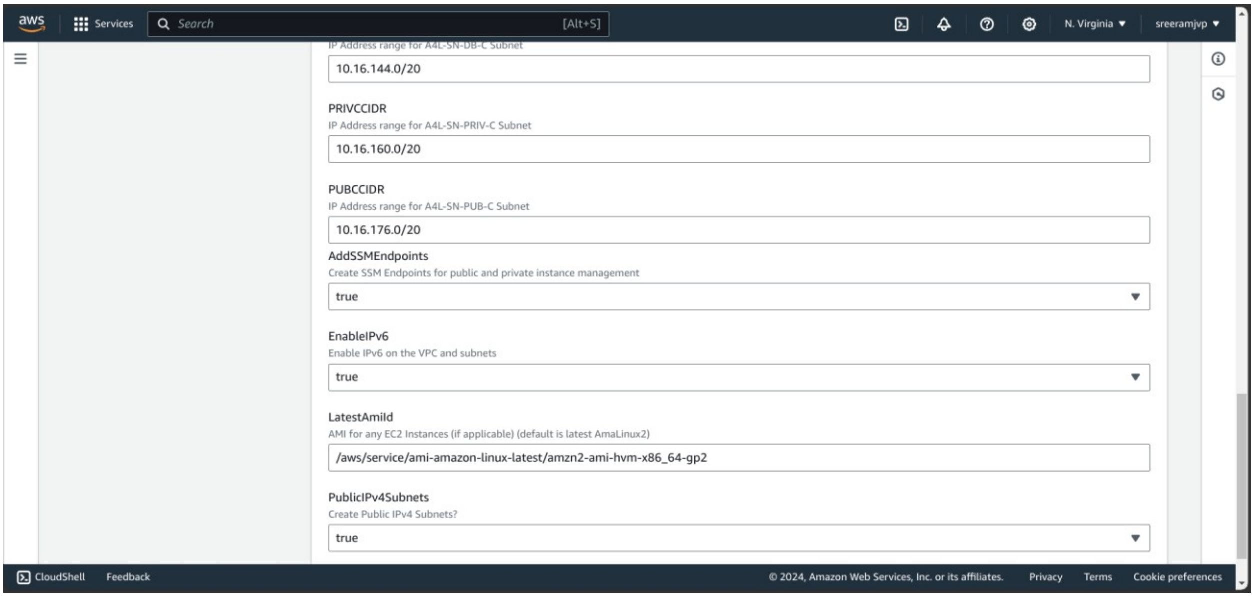
VPCCIDR

IP Address range for the VPC

10.16.0.0/16

AZA

RESERVEDACIDR



IP Address range for A4L-SN-DB-C Subnet
10.16.144.0/20

PRIVCCIDR
IP Address range for A4L-SN-PRIV-C Subnet
10.16.160.0/20

PUBCCIDR
IP Address range for A4L-SN-PUB-C Subnet
10.16.176.0/20

AddSSMEndpoints
Create SSM Endpoints for public and private instance management
true

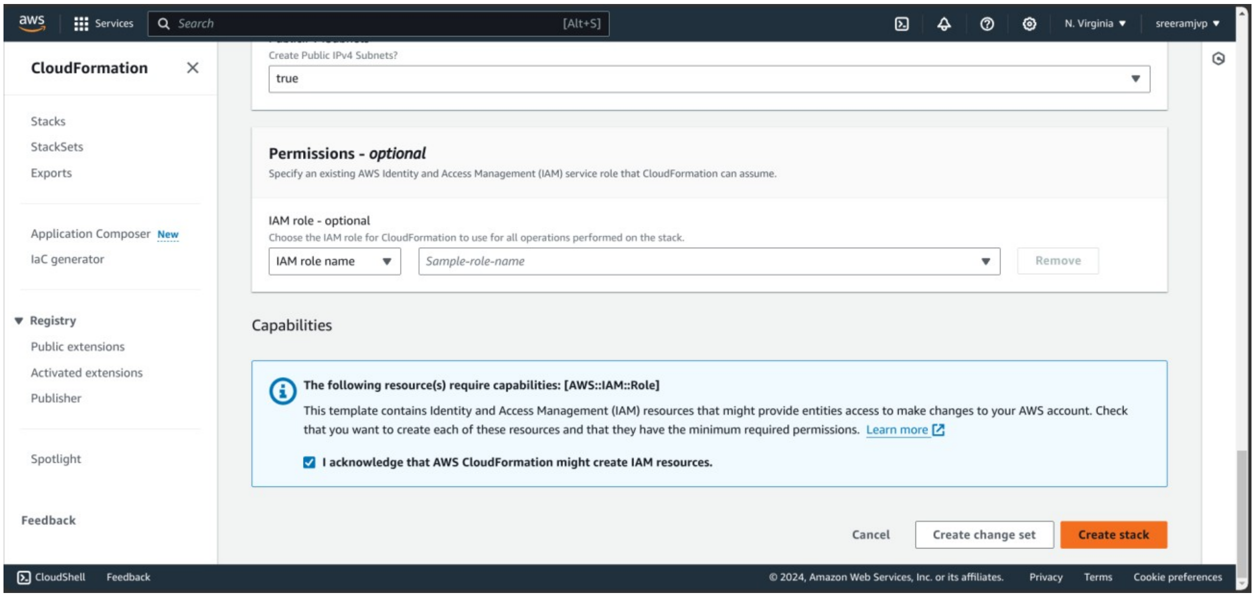
EnableIPv6
Enable IPv6 on the VPC and subnets
true

LatestAmiId
AMI for any EC2 Instances (if applicable) (default is latest AmaLinux2)
/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

PublicIPv4Subnets
Create Public IPv4 Subnets?
true

As seen, all of the required resources and policies have been defined by the sample template.

Authorizing IAM role for access and permissions:



Create Public IPv4 Subnets?
true

Permissions - *optional*
Specify an existing AWS Identity and Access Management (IAM) service role that CloudFormation can assume.

IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.
IAM role name ▼ Sample-role-name ▼ Remove

Capabilities

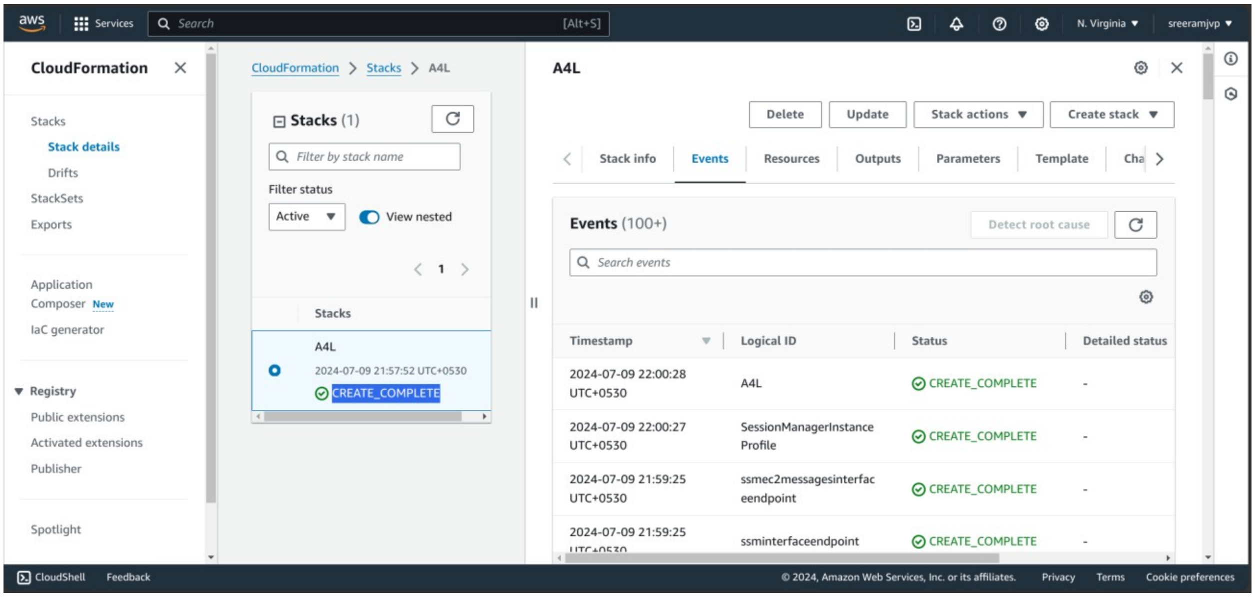
The following resource(s) require capabilities: [AWS::IAM::Role]
This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. Learn more

☑ **I acknowledge that AWS CloudFormation might create IAM resources.**

Click ***Create Stack.***

(Step-5):

Wait till the status changed from CREATE_IN_PROGRESS *to* CREATE_COMPLETE.



Stacks (1)

Filter status: Active ▼ View nested

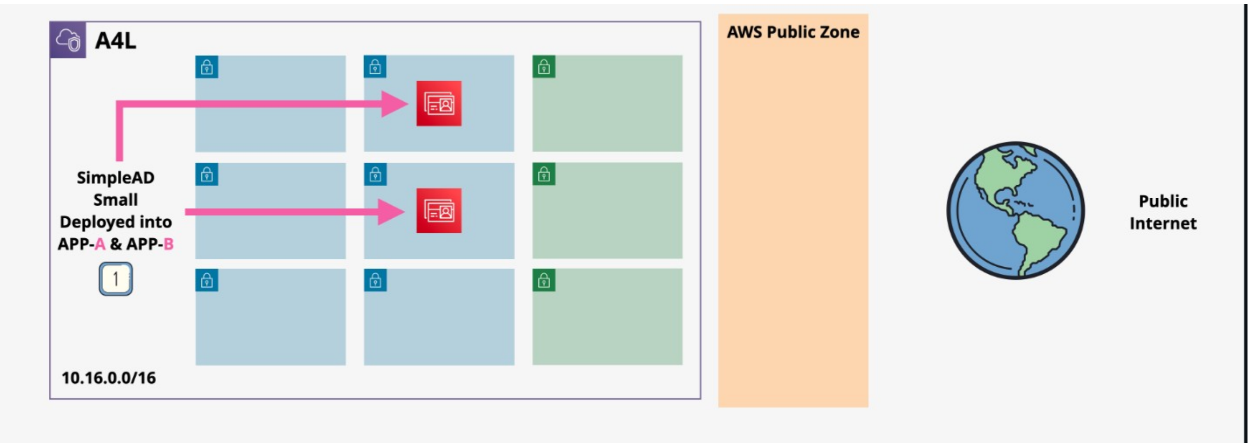
Stacks
A4L 2024-07-09 21:57:52 UTC+0530 CREATE_COMPLETE

A4L

Events (100+)

Timestamp	Logical ID	Status	Detailed status
2024-07-09 22:00:28 UTC+0530	A4L	CREATE_COMPLETE	-
2024-07-09 22:00:27 UTC+0530	SessionManagerInstanceProfile	CREATE_COMPLETE	-
2024-07-09 21:59:25 UTC+0530	ssmec2messagesinterfaceendpoint	CREATE_COMPLETE	-
2024-07-09 21:59:25 UTC+0530	ssminterfaceendpoint	CREATE_COMPLETE	-

Stage:1

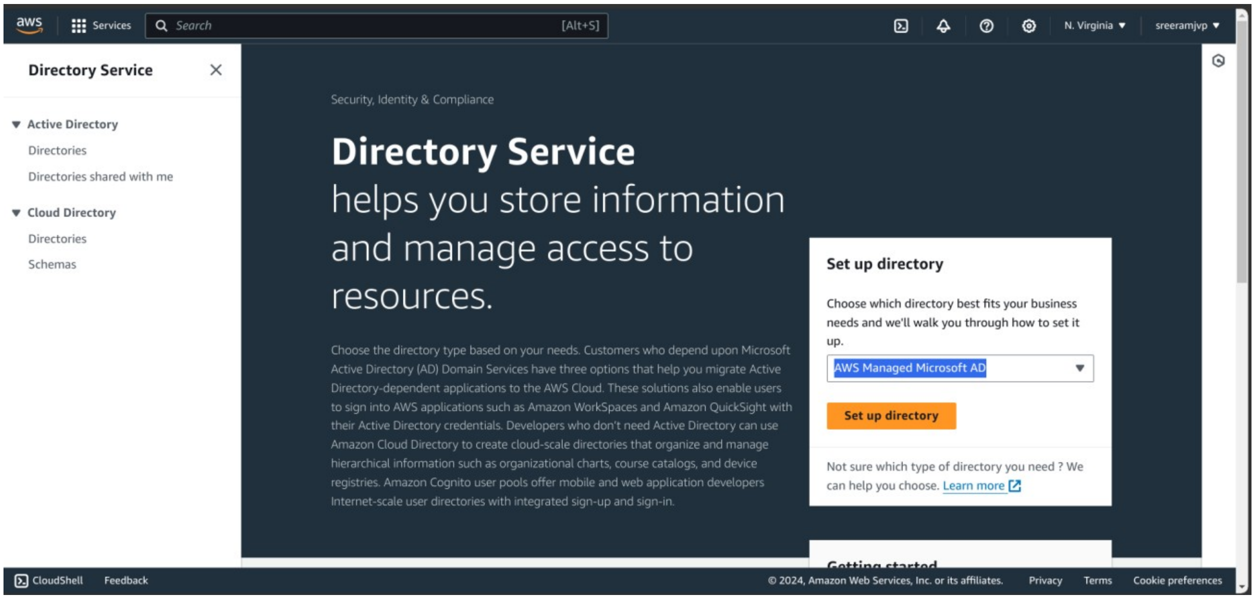


Client VPN is capable of using identity-based and/or certificate-based authentication - this demo will use Simple AD as an identity store

We will be using AWS directory service in simple AD mode. AWS Directory Service in Simple AD mode refers to a specific configuration of AWS Directory Service that provides basic Active Directory (AD) features without requiring a full Microsoft AD setup. It is designed to be simple to set up and manage, making it suitable for smaller organizations less complex use cases.

(Step-6)

Choose AWS Directory service.



Directory Service helps you store information and manage access to resources.

Choose the directory type based on your needs. Customers who depend upon Microsoft Active Directory (AD) Domain Services have three options that help you migrate Active Directory-dependent applications to the AWS Cloud. These solutions also enable users to sign into AWS applications such as Amazon WorkSpaces and Amazon QuickSight with their Active Directory credentials. Developers who don't need Active Directory can use Amazon Cloud Directory to create cloud-scale directories that organize and manage hierarchical information such as organizational charts, course catalogs, and device registries. Amazon Cognito user pools offer mobile and web application developers Internet-scale user directories with integrated sign-up and sign-in.

Set up directory

Choose which directory best fits your business needs and we'll walk you through how to set it up.

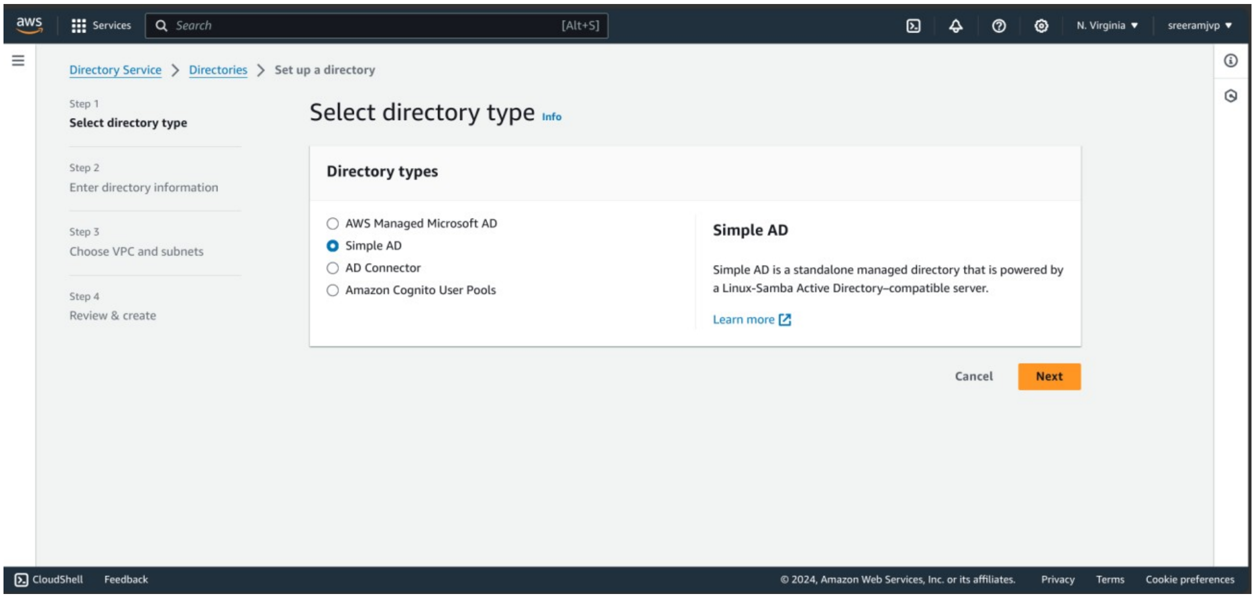
AWS Managed Microsoft AD ▾

Set up directory

Not sure which type of directory you need ? We can help you choose. Learn more

© 2024, Amazon Web Services, Inc. or its affiliates.

Select simple AD in the drop down menu:



Step 1
Select directory type

Step 2
Enter directory information

Step 3
Choose VPC and subnets

Step 4
Review & create

Select directory type

Directory types

○ AWS Managed Microsoft AD
● Simple AD
○ AD Connector
○ Amazon Cognito User Pools

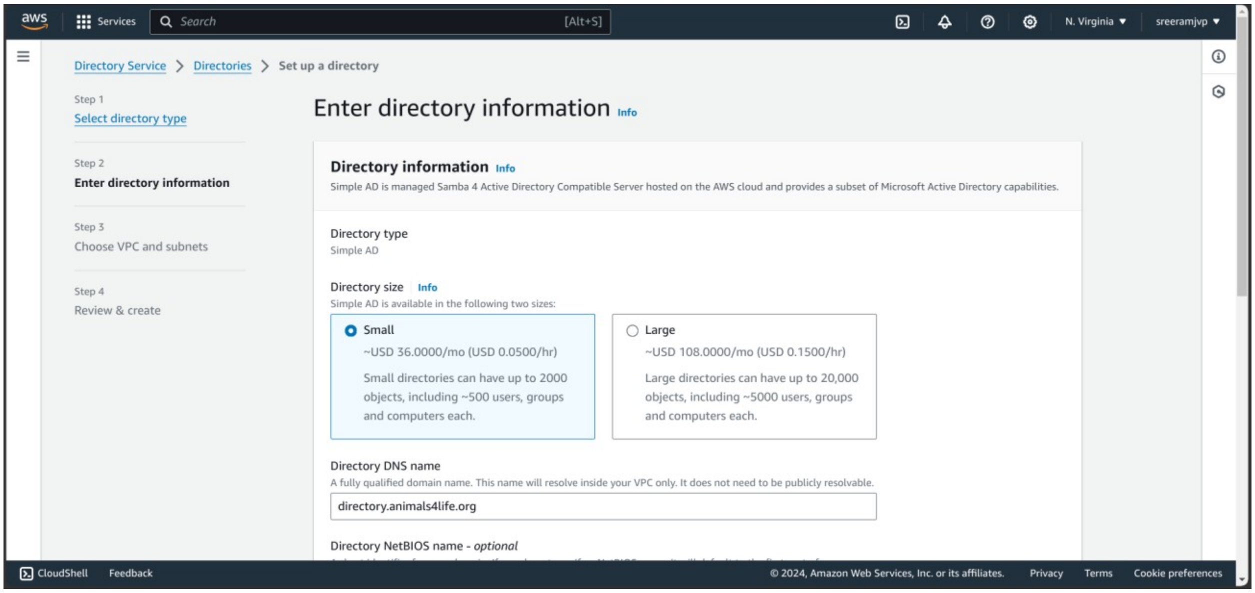
Simple AD

Simple AD is a standalone managed directory that is powered by a Linux-Samba Active Directory–compatible server.

Learn more

© 2024, Amazon Web Services, Inc. or its affiliates.

Directory information:



Directory Service > Directories > Set up a directory

Step 1
Select directory type

Step 2
Enter directory information

Step 3
Choose VPC and subnets

Step 4
Review & create

Enter directory information Info

Directory information Info

Simple AD is managed Samba 4 Active Directory Compatible Server hosted on the AWS cloud and provides a subset of Microsoft Active Directory capabilities.

Directory type
Simple AD

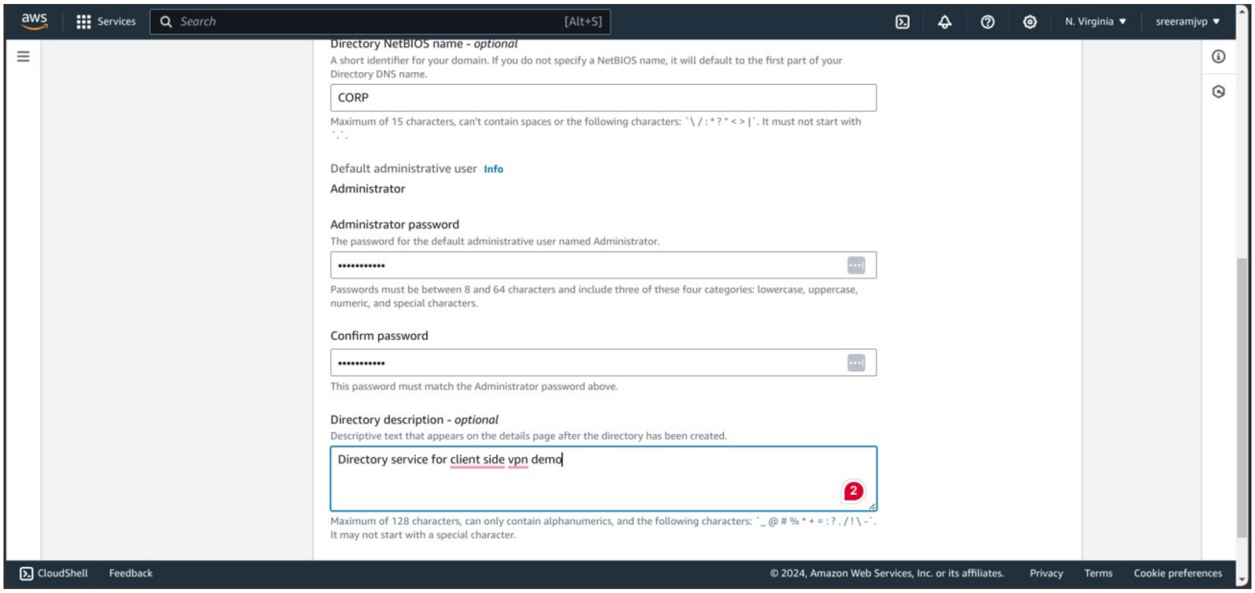
Directory size Info
Simple AD is available in the following two sizes:

● Small
~USD 36.0000/mo (USD 0.0500/hr)
Small directories can have up to 2000 objects, including ~500 users, groups and computers each.

○ Large
~USD 108.0000/mo (USD 0.1500/hr)
Large directories can have up to 20,000 objects, including ~5000 users, groups and computers each.

Directory DNS name
A fully qualified domain name. This name will resolve inside your VPC only. It does not need to be publicly resolvable.
directory.animals4life.org

Directory NetBIOS name - *optional*



Directory NetBIOS name - *optional*
A short identifier for your domain. If you do not specify a NetBIOS name, it will default to the first part of your Directory DNS name.
CORP
Maximum of 15 characters, can't contain spaces or the following characters: `\ / : * ? " < > |`. It must not start with `.`.

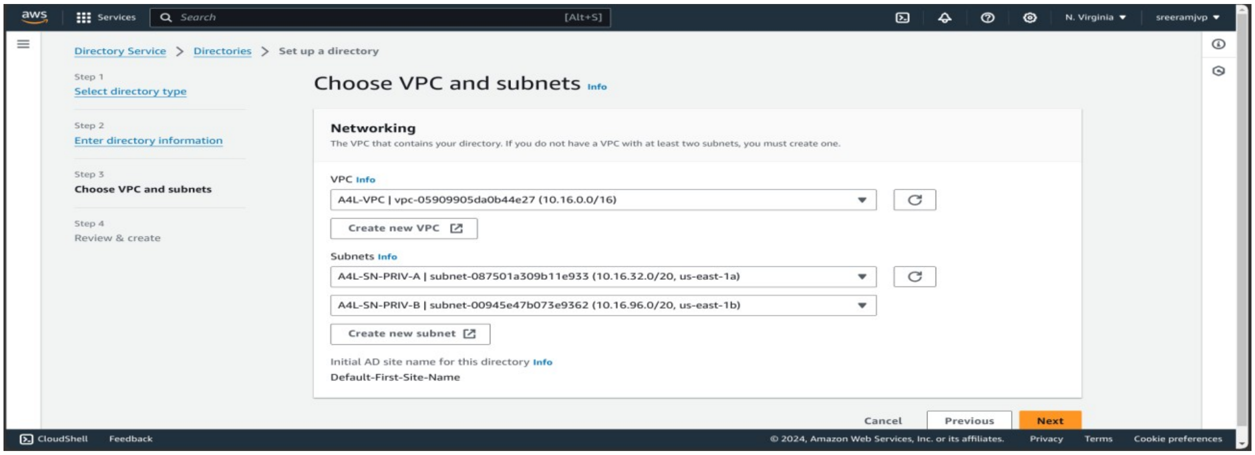
Default administrative user Info
Administrator

Administrator password
The password for the default administrative user named Administrator.
•••••••••••
Passwords must be between 8 and 64 characters and include three of these four categories: lowercase, uppercase, numeric, and special characters.

Confirm password
•••••••••••
This password must match the Administrator password above.

Directory description - *optional*
Descriptive text that appears on the details page after the directory has been created.
Directory service for client side vpn demo
Maximum of 128 characters, can only contain alphanumerics, and the following characters: `_ @ # % * + = : ? . / ! \ -`. It may not start with a special character.

Choosing vpc and subnets:



Directory Service > Directories > Set up a directory

Step 1 Select directory type

Step 2 Enter directory information

Step 3 **Choose VPC and subnets**

Step 4 Review & create

Choose VPC and subnets

Networking

The VPC that contains your directory. If you do not have a VPC with at least two subnets, you must create one.

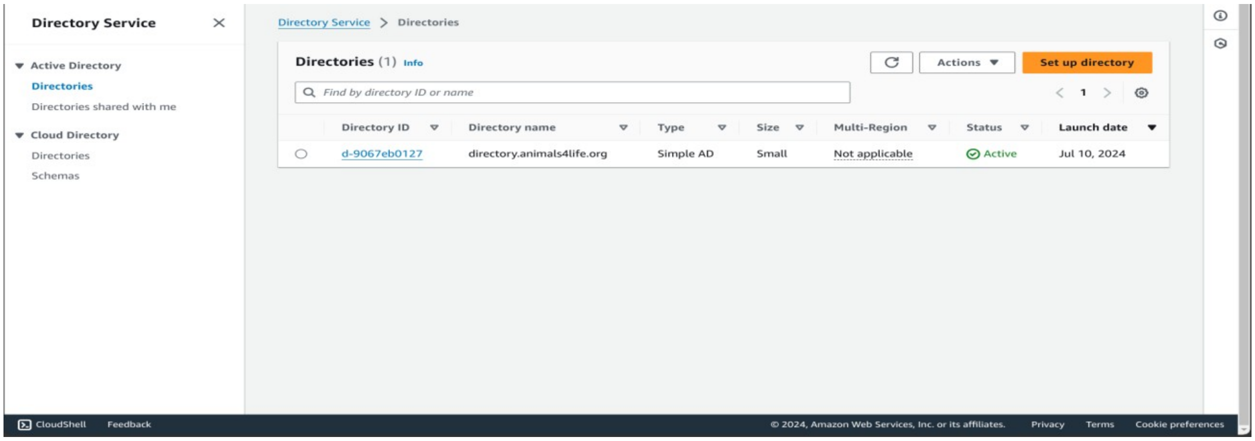
VPC: A4L-VPC | vpc-05909905da0b44e27 (10.16.0.0/16)

Subnets: A4L-SN-PRIV-A | subnet-087501a309b11e933 (10.16.32.0/20, us-east-1a)

A4L-SN-PRIV-B | subnet-00945e47b073e9362 (10.16.96.0/20, us-east-1b)

Initial AD site name for this directory

Default-First-Site-Name



Directory Service > Directories

Directories (1)

	Directory ID	Directory name	Type	Size	Multi-Region	Status	Launch date
○	d-9067eb0127	directory.animals4life.org	Simple AD	Small	Not applicable	Active	Jul 10, 2024

Creating a Server Certificate for Client VPN

Prerequisites:

- Ensure the directory is in the active state.

Authentication Method:

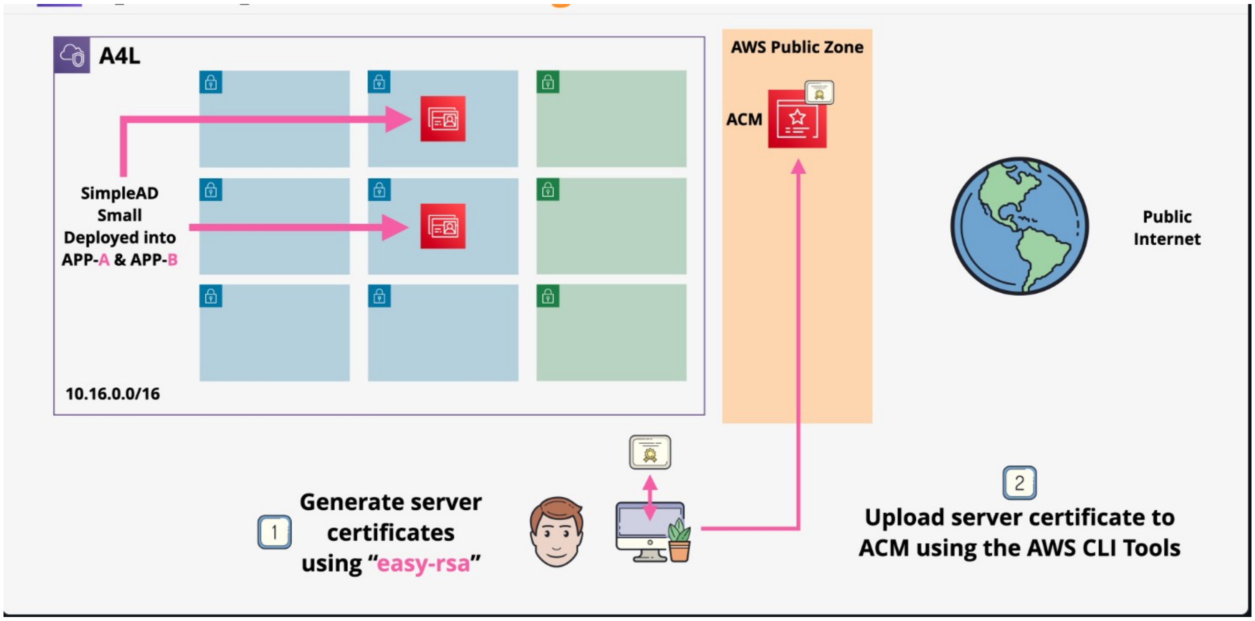
There are two ways we can implement authentication.

. Using Certificate based authentication.

- Using Identity-based Authentication.

Our choice method in the implementation is *Identity-based Authentication.*

We are at this level:



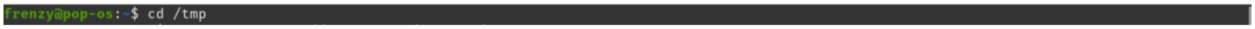
(Step-7):

" A service certificate is generated IB Access Management"

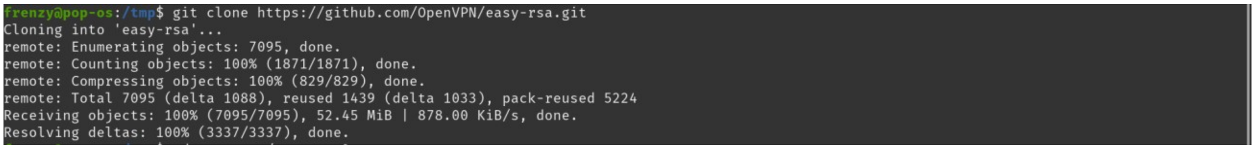
Moving to local machine......

OS used: Linux(debian)>

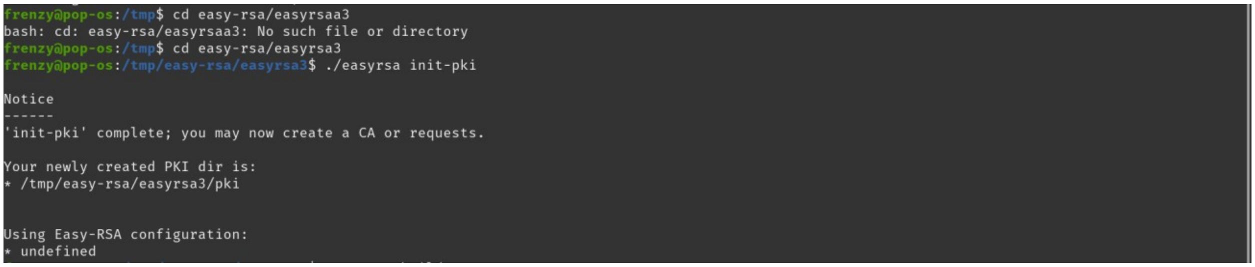
Steps:



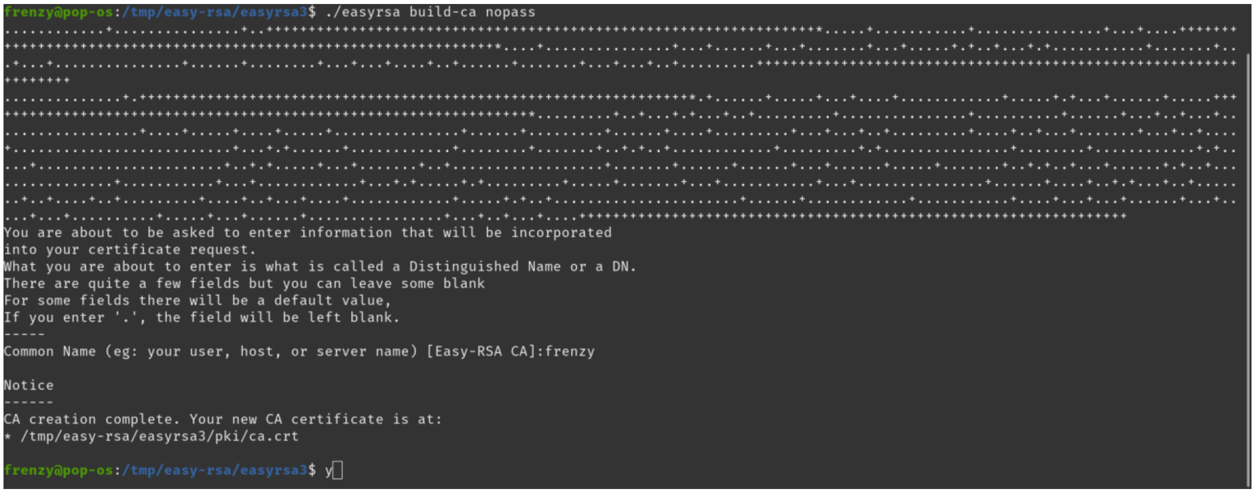
```
frenzy@pop-os:~$ cd /tmp
```



```
frenzy@pop-os:/tmp$ git clone https://github.com/OpenVPN/easy-rsa.git
Cloning into 'easy-rsa'...
remote: Enumerating objects: 7095, done.
remote: Counting objects: 100% (1871/1871), done.
remote: Compressing objects: 100% (829/829), done.
remote: Total 7095 (delta 1088), reused 1439 (delta 1033), pack-reused 5224
Receiving objects: 100% (7095/7095), 52.45 MiB | 878.00 KiB/s, done.
Resolving deltas: 100% (3337/3337), done.
```

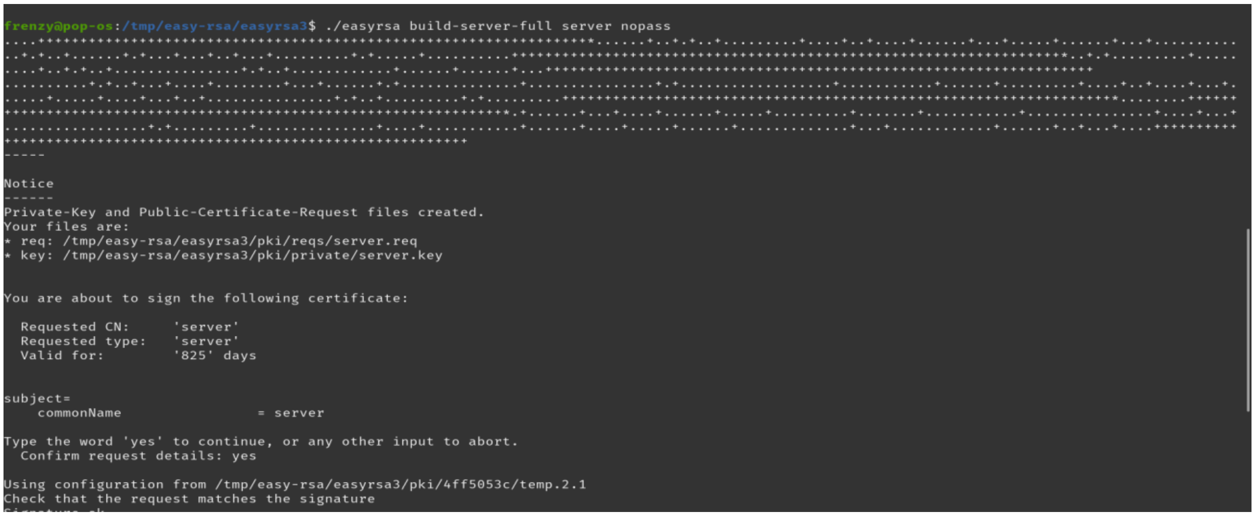


```
frenzy@pop-os:/tmp$ cd easy-rsa/easyrsaa3
bash: cd: easy-rsa/easyrsaa3: No such file or directory
frenzy@pop-os:/tmp$ cd easy-rsa/easyrsa3
frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ ./easyrsa init-pki

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

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


Using Easy-RSA configuration:
* undefined
```



```
frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ ./easyrsa build-ca nopass
............+...............+..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.....+...........+................+...+....++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*....+...............+...+......+...+........+...+......+.+..+...+.+...........+.......+..
.+...+................+......+........+...+...+....+..+......+.......+...+....+..+.........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++
.............+.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+......+.....+...+....+............+.....+.+...+......+.....+++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.........+..+...+.+...+..+..........+................+..........+......+...+..+...+..
................+....+.....+....+.....+...............+......+.........+......+....+..........+...+...+..+..........+....+..+...+......+...+..+....
+........................+...+.+......+............+........+.......+..+..+..............+.........+.+..................+........+...........+.+..
...+....................+..+.+.....+...+.......+..+..........................+.......+......+.......+...+......+.......+...+..+..+...+.......+..+..+...
............+...........+...+............+...+.+.....+.+.........+.....+......+...+............+...+.................+......+....+..+.+...+..+.....
..+..+....+..+.........+....+..+...+....+............+.....+.+..+.....................+......+.............+...........+....+...+...+.....+...+..
...+...+..........+.....+...+......+................+...+..+...+....+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:frenzy

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

frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ y
```

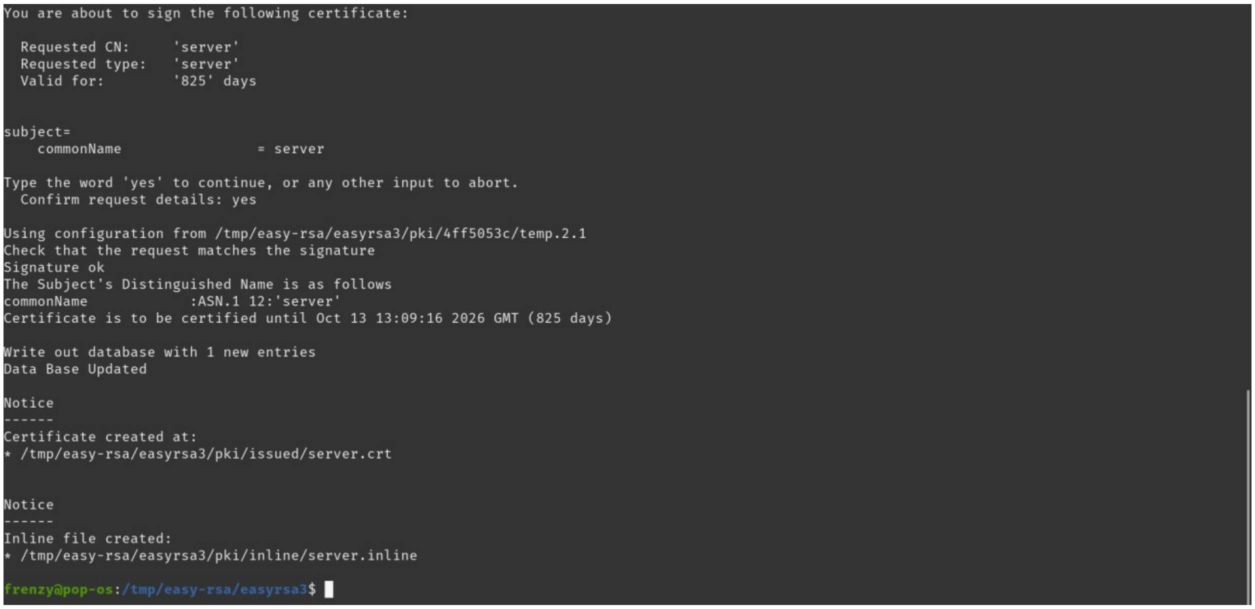
Server buiding:



```
frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ ./easyrsa build-server-full server nopass
....+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*......+..+.+..+..........+....+..+....+......+...+.....+......+...+..........
..+.+..+......+.+...+...+..+...+.........+.+.....+..........+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*..+.+.........+.....
....+..+.+..+...............+.+..+...........+......+......+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
..........+.+..+...+....+........+...+......+.+..............+...............+.+..................+...........+......+.........+....+..+....+...+.
.....+.....+....+...+..+...............+.+..+..........+.+.........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*........++++++
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+......+...+....+......+.....+.........+.......+...........+..................+....+...+
................+.+.........+...............+....+...........+......+....+.....+......+.............+...+..............+......+..+...+....+++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----

Notice
------
Private-Key and Public-Certificate-Request files created.
Your files are:
* req: /tmp/easy-rsa/easyrsa3/pki/reqs/server.req
* key: /tmp/easy-rsa/easyrsa3/pki/private/server.key


You are about to sign the following certificate:

  Requested CN:     'server'
  Requested type:   'server'
  Valid for:        '825' days


subject=
    commonName                = server

Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes

Using configuration from /tmp/easy-rsa/easyrsa3/pki/4ff5053c/temp.2.1
Check that the request matches the signature
Signature ok
```

Name: *“server”*

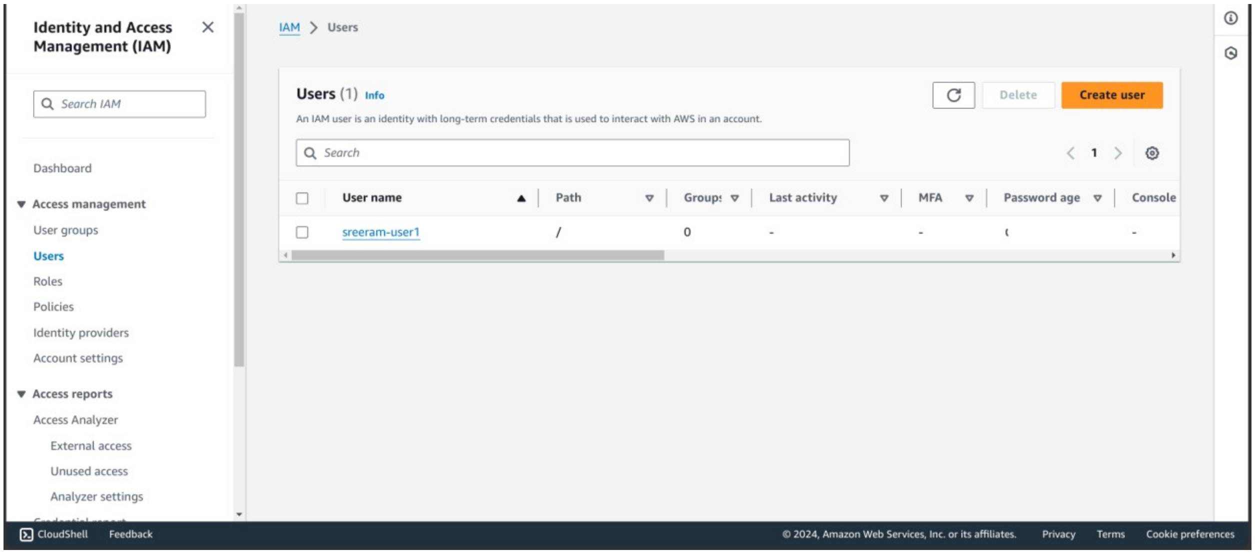
Server certificate succefully completed….



```
You are about to sign the following certificate:

  Requested CN:     'server'
  Requested type:   'server'
  Valid for:        '825' days


subject=
    commonName                = server

Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes

Using configuration from /tmp/easy-rsa/easyrsa3/pki/4ff5053c/temp.2.1
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'server'
Certificate is to be certified until Oct 13 13:09:16 2026 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Notice
------
Certificate created at:
* /tmp/easy-rsa/easyrsa3/pki/issued/server.crt


Notice
------
Inline file created:
* /tmp/easy-rsa/easyrsa3/pki/inline/server.inline

frenzy@pop-os:/tmp/easy-rsa/easyrsa3$
```

“Pki folder contains all the required certificates. All the contents in easyrsa folder are to uploaded to ACM, so that client vpn could authenticate and access.”

#Attaching a user profile to ACM



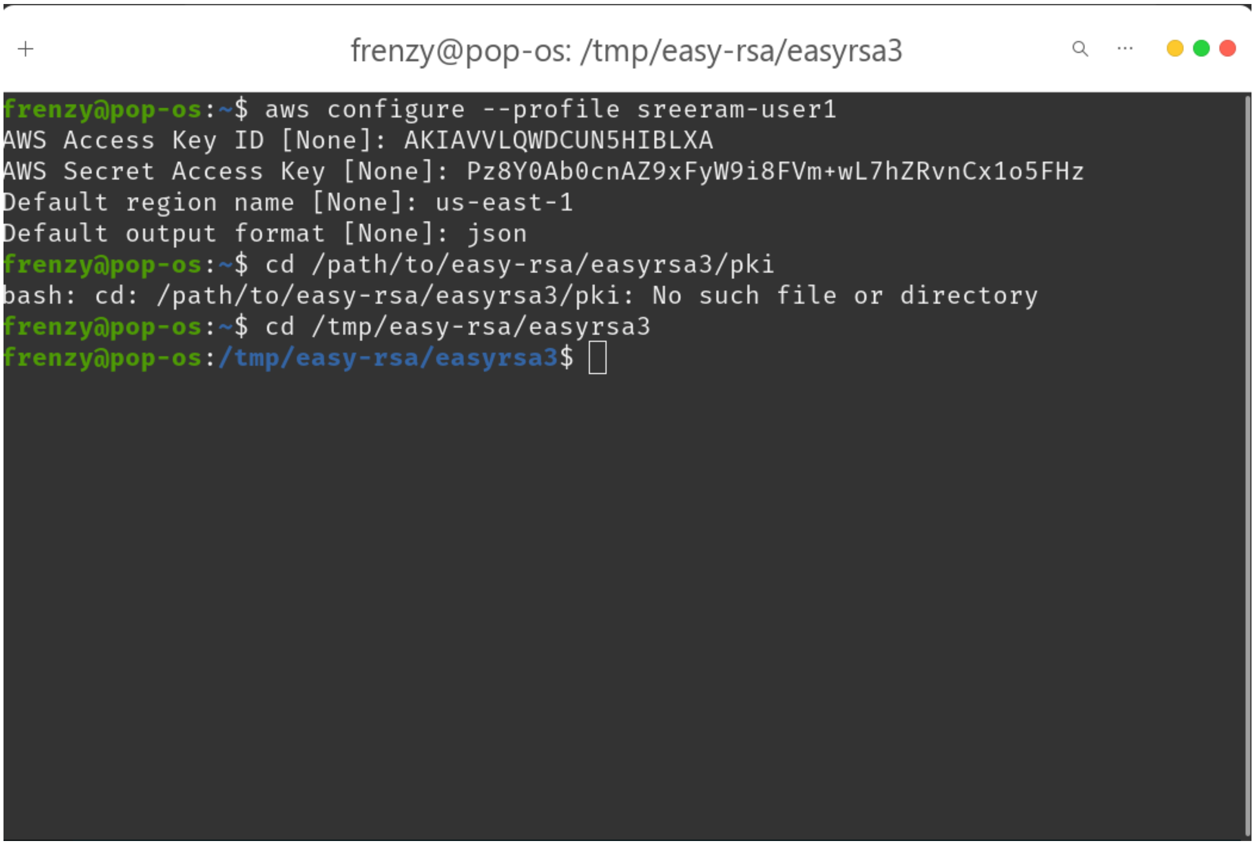
Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

	User name	Path	Groups	Last activity	MFA	Password age	Console
□	sreeram-user1	/	0	-	-	(	-

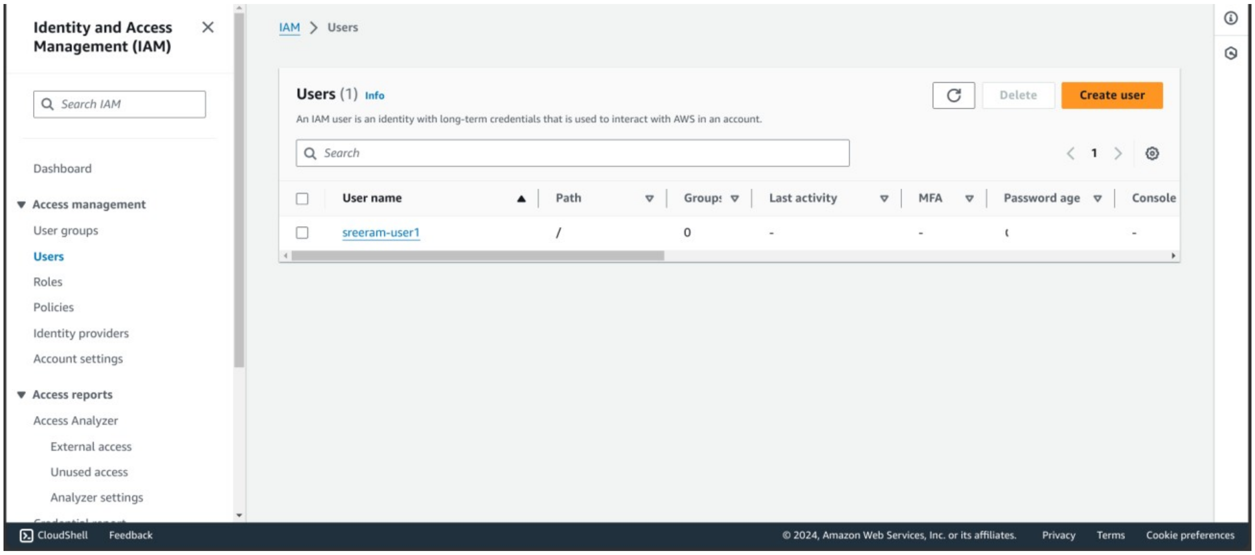
Created a user-profile using aws console, with IAM.

. Accessing user-profile ‘sreeram-user1’ using aws acm profile.



```
frenzy@pop-os:~$ aws configure --profile sreeram-user1
AWS Access Key ID [None]: AKIAVVLQWDCUN5HIBLXA
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
frenzy@pop-os:~$ cd /path/to/easy-rsa/easyrsa3/pki
bash: cd: /path/to/easy-rsa/easyrsa3/pki: No such file or directory
frenzy@pop-os:~$ cd /tmp/easy-rsa/easyrsa3
frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ 
```

Adding custom policy to allow write ACM server certificates upload:



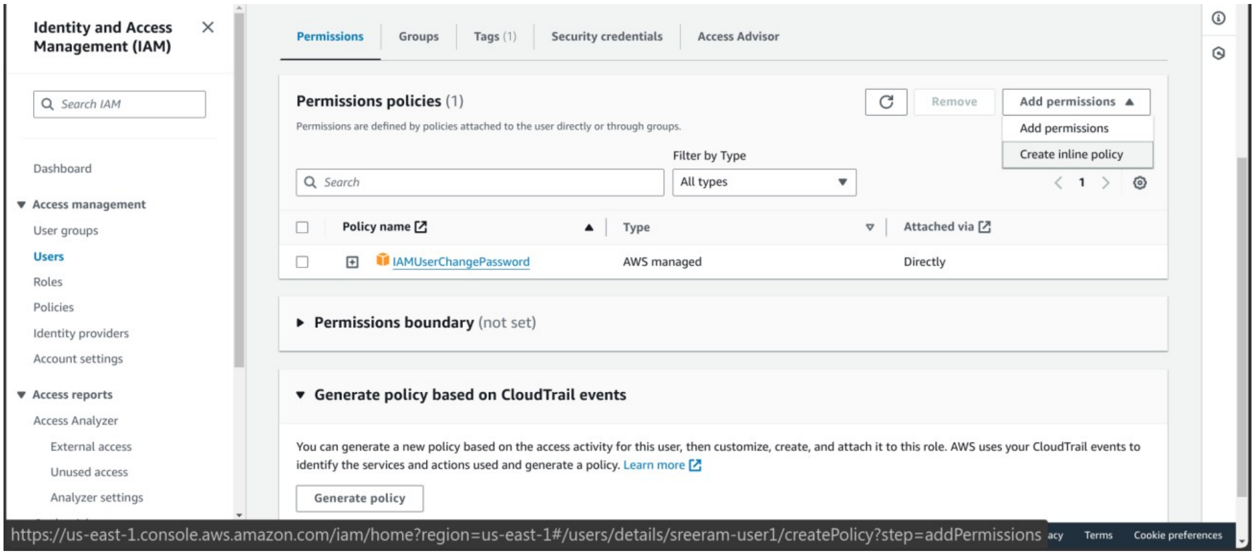
Users (1)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

	User name	Path	Group:	Last activity	MFA	Password age	Console
□	sreeram-user1	/	0	-	-	(	-

© 2024, Amazon Web Services, Inc. or its affiliates.

#Creating a new inline policy:



Permissions are defined by policies attached to the user directly or through groups.

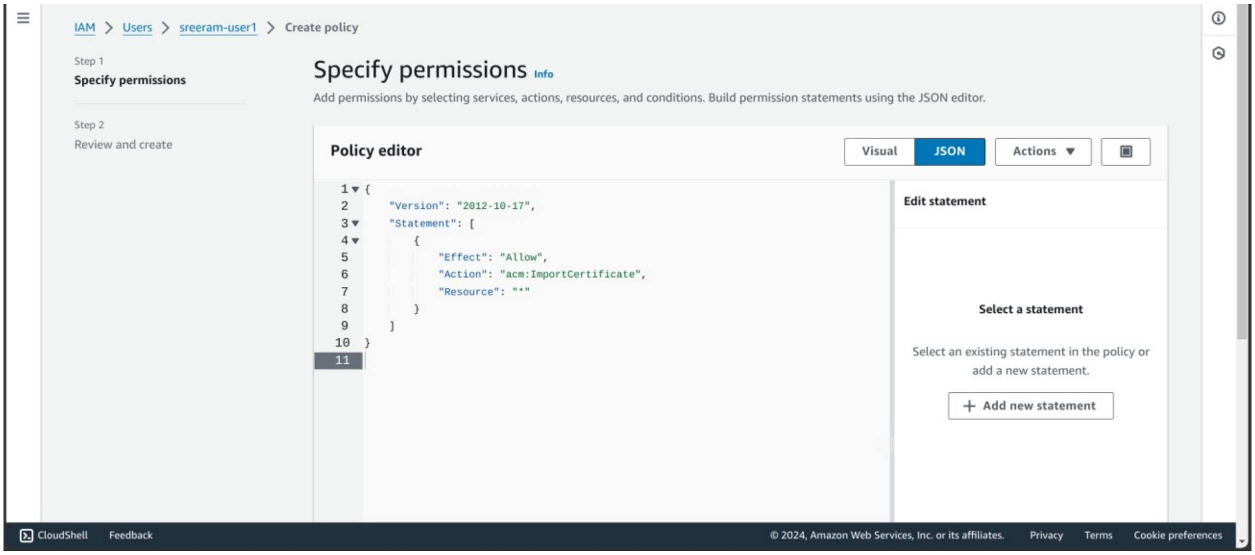
☐	Policy name	Type	Attached via
☐	IAMUserChangePassword	AWS managed	Directly

You can generate a new policy based on the access activity for this user, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/details/sreeram-user1/createPolicy?step=addPermissions

Adding policy in form of json, which would only allow AWS CLI to access and write into the simple AD directory service using ACM.

# “Sreeram-user1” is only allowed to accessed through AWS CLI, for which the access keys and secrey access keys have been generated....



IAM > Users > sreeram-user1 > Create policy

Step 1
Specify permissions

Step 2
Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

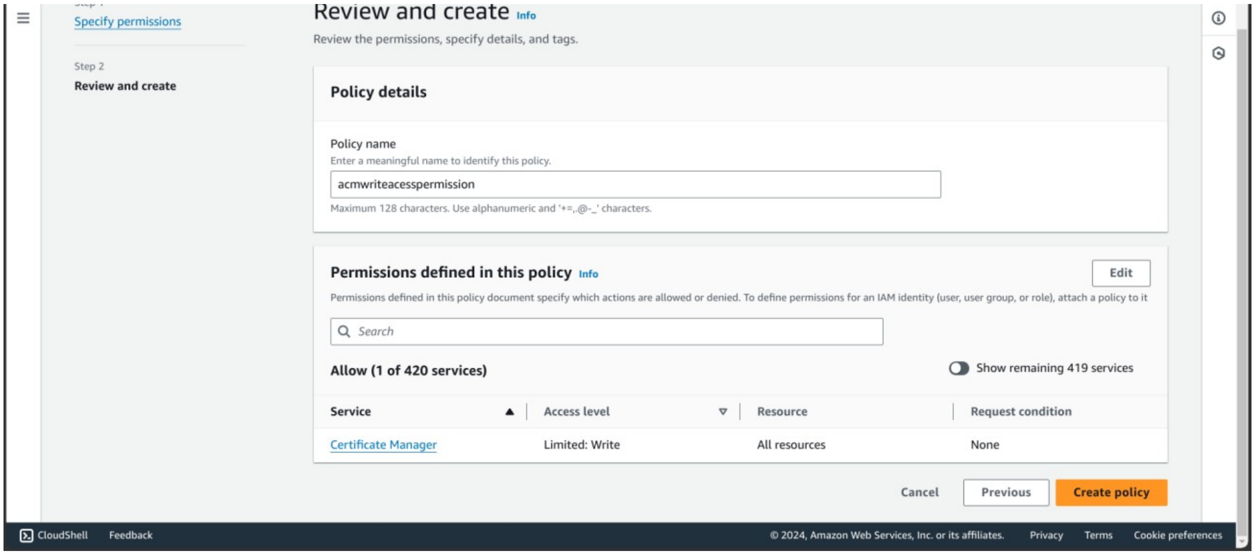
Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "acm:ImportCertificate",
            "Resource": "*"
        }
    ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.



Specify permissions

Step 2
Review and create

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.
acmwriteacesspermission
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

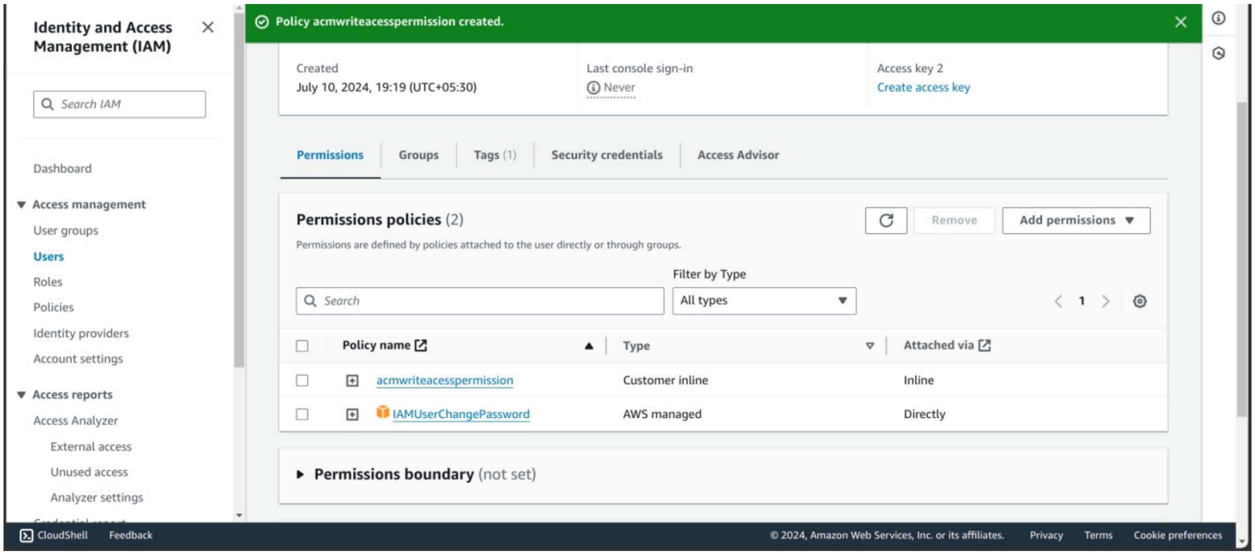
Allow (1 of 420 services)

○ Show remaining 419 services

Service	Access level	Resource	Request condition
Certificate Manager	Limited: Write	All resources	None

……………..

Policy has been successfully created…..



Policy acmwriteacesspermission created.

Created
July 10, 2024, 19:19 (UTC+05:30)

Last console sign-in
Never

Access key 2
Create access key

Permissions policies (2)

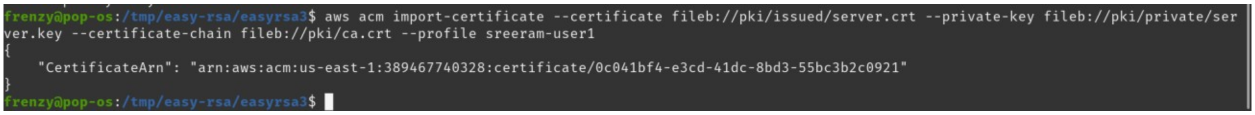
Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

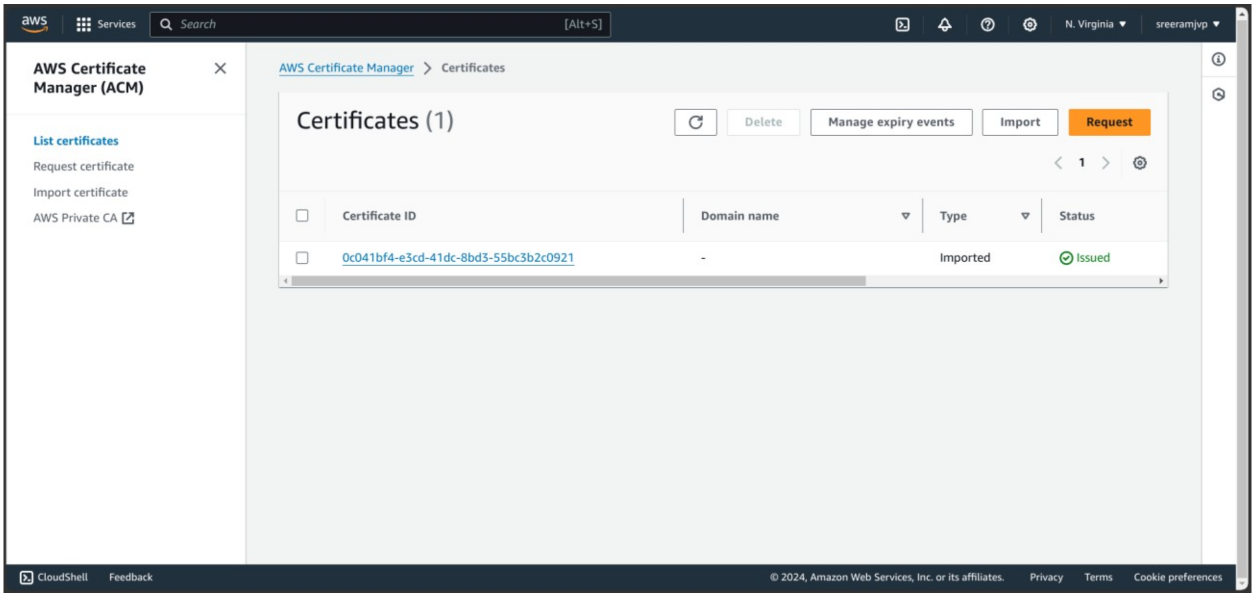
Policy name	Type	Attached via
acmwriteacesspermission	Customer inline	Inline
IAMUserChangePassword	AWS managed	Directly

Permissions boundary (not set)

Uploaded the server certificates to “sreeram-user1” profile using AWS ACM successfully...



```
frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ aws acm import-certificate --certificate fileb://pki/issued/server.crt --private-key fileb://pki/private/ser
ver.key --certificate-chain fileb://pki/ca.crt --profile sreeram-user1
{
    "CertificateArn": "arn:aws:acm:us-east-1:389467740328:certificate/0c041bf4-e3cd-41dc-8bd3-55bc3b2c0921"
}
frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ 
```

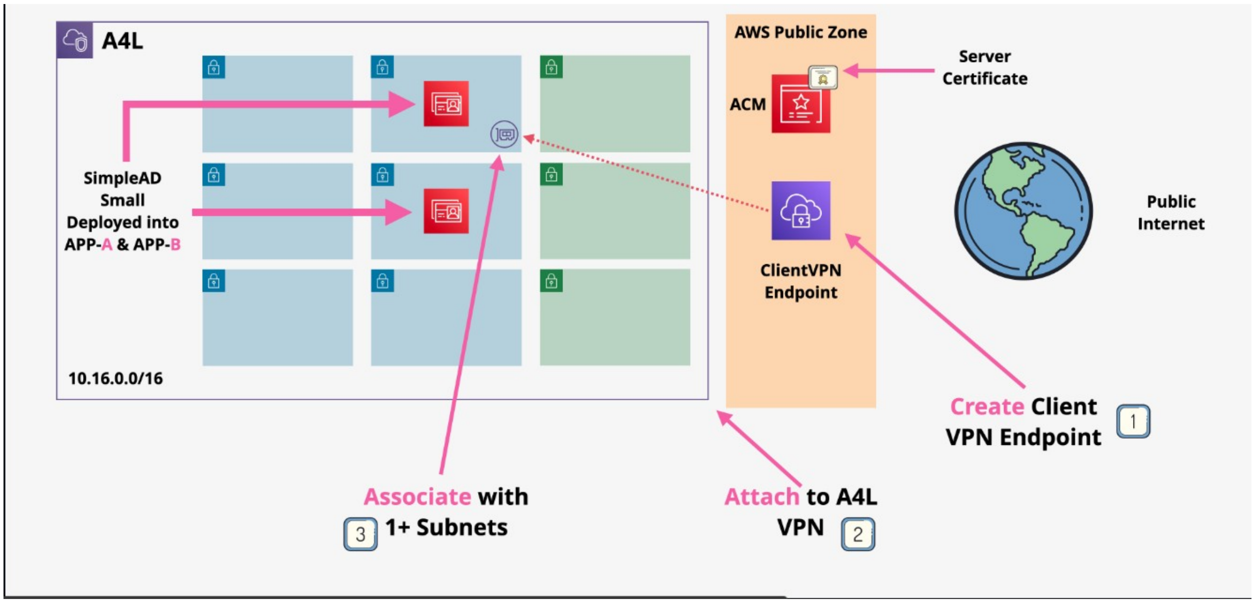


Certificates (1)

Certificate ID	Domain name	Type	Status
0c041bf4-e3cd-41dc-8bd3-55bc3b2c0921	-	Imported	Issued

(Step-8):

We are here.....

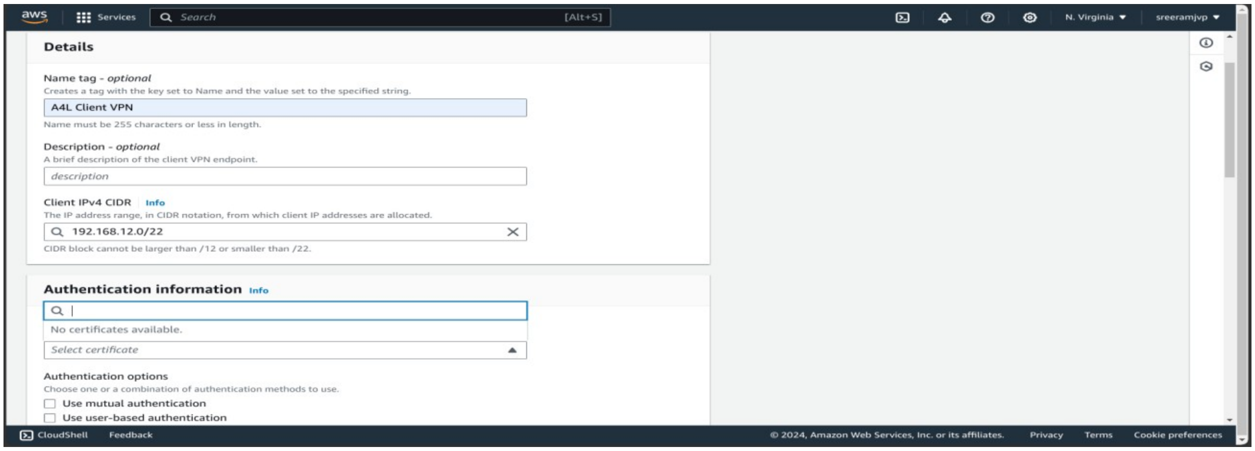


This stage involves creating a vpn end point and association using the VPC service within the aws console:

1. *Choose vpc service*
2. *Select client VPN endpoint*

Challenges faced:

As specified through the inline policy , acmwriteaccess is alone not enough for locating th server ARN certificates.



Details

Name tag - *optional*
Creates a tag with the key set to Name and the value set to the specified string.
A4L Client VPN
Name must be 255 characters or less in length.

Description - *optional*
A brief description of the client VPN endpoint.
description

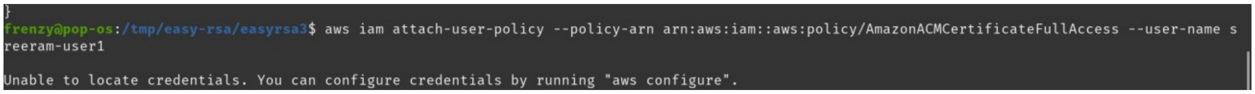
Client IPv4 CIDR Info
The IP address range, in CIDR notation, from which client IP addresses are allocated.
192.168.12.0/22
CIDR block cannot be larger than /12 or smaller than /22.

Authentication information Info

No certificates available.
Select certificate

Authentication options
Choose one or a combination of authentication methods to use.
□ Use mutual authentication
□ Use user-based authentication

Unable to update or add inline policy using aws cli:

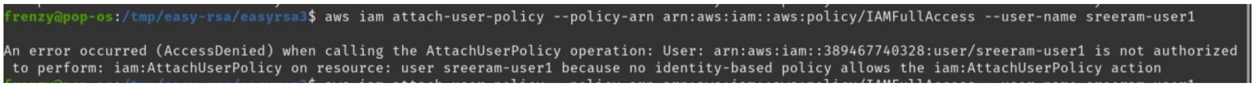


```
}
frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ aws iam attach-user-policy --policy-arn arn:aws:iam::aws:policy/AmazonACMCertificateFullAccess --user-name s
reeram-user1

Unable to locate credentials. You can configure credentials by running "aws configure".
```

Solution;

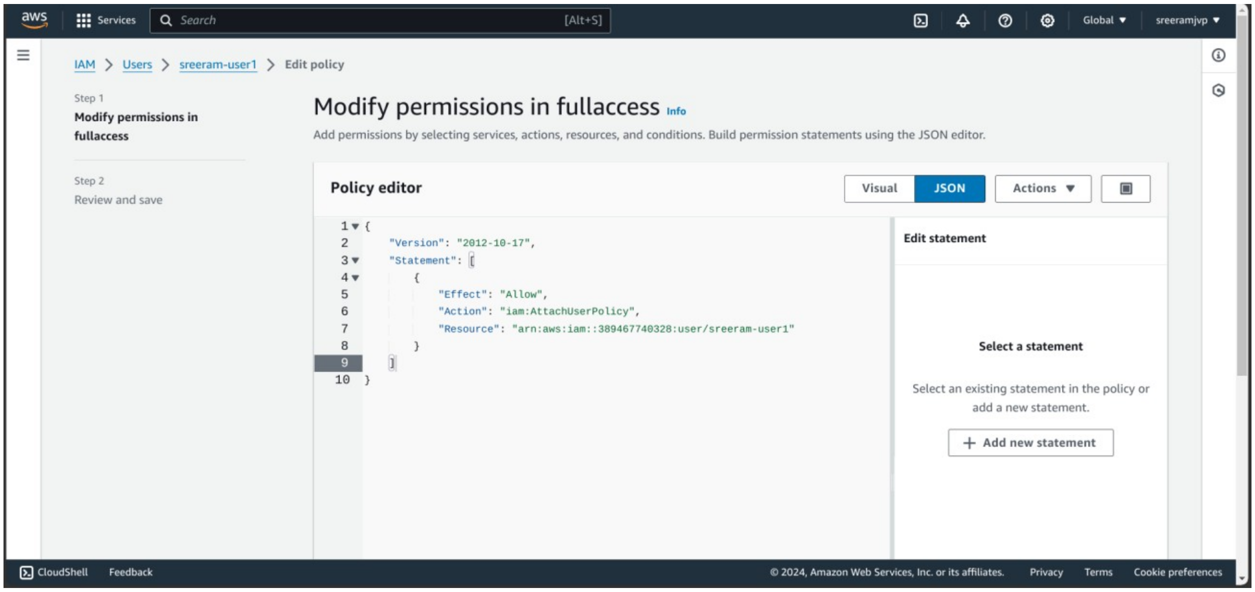
Used ‘aws configure’ to authenticate request from my local machine.

Lack of policy permissions for sreeram-user1 to manifest policy in Identity access management:



```
frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ aws iam attach-user-policy --policy-arn arn:aws:iam::aws:policy/IAMFullAccess --user-name sreeram-user1

An error occurred (AccessDenied) when calling the AttachUserPolicy operation: User: arn:aws:iam::389467740328:user/sreeram-user1 is not authorized
 to perform: iam:AttachUserPolicy on resource: user sreeram-user1 because no identity-based policy allows the iam:AttachUserPolicy action
```

To solve this, I have added an inline policy that allows attaching policy to user-1



aws
Services
Search
[Alt+S]
Global ▾
sreeramjvp ▾
IAM > Users > sreeram-user1 > Edit policy
Step 1
Modify permissions in fullaccess
Step 2
Review and save

Modify permissions in fullaccess Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:AttachUserPolicy",
            "Resource": "arn:aws:iam::389467740328:user/sreeram-user1"
        }
    ]
}
```

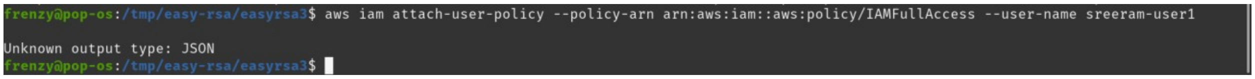
Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

On adding this policy, we are now able to attach AWS managed policy to allow fullaccess to the acm service for “sreeram-user1”:

Succes….



```
frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ aws iam attach-user-policy --policy-arn arn:aws:iam::aws:policy/IAMFullAccess --user-name sreeram-user1

Unknown output type: JSON
frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ 
```

To solve this, we need to add a new inline policy that allows acm discoverability(i.e. ALLOWFULLACCESS), to ensure its visible to the VPC service.

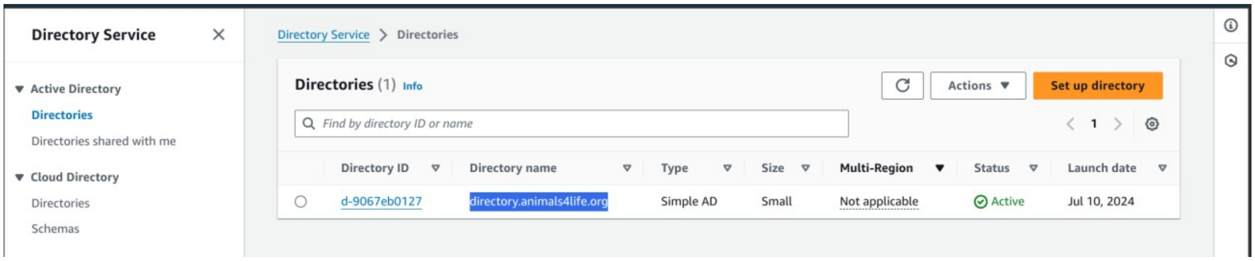
Now lets configure the cliend VPN end points…

But no so fast…

We still cant get out vpc to recognise the ACM certificate:

On further investigation, with reference to https://repost.aws/questions/QUbHwO-HGfTcWCrSc5fBZmKw/server-certificate-not-showing-for-vpn-endpoint, the main error stems from not including domain name in the server build command in easyrsa.

Our domain name, that we have created in simple AD, using Directory service, is
directory.a“nimals4life.org”

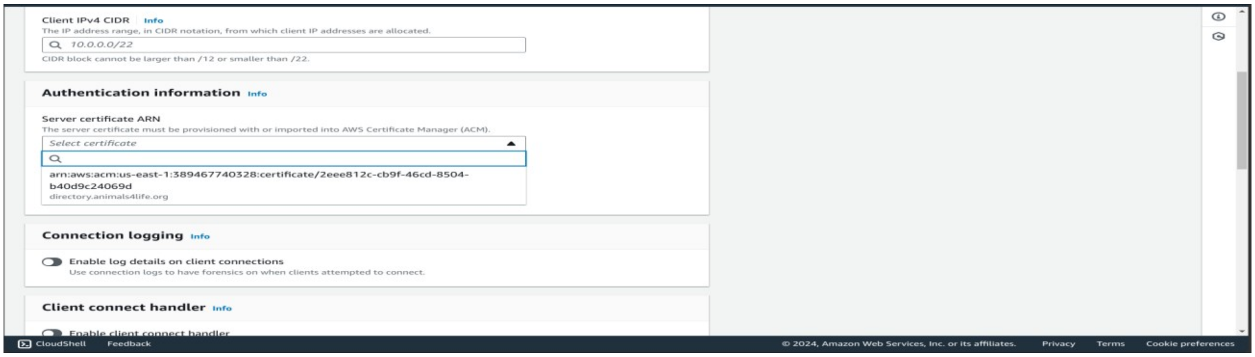


	Directory ID	Directory name	Type	Size	Multi-Region	Status	Launch date
○	d-9067eb0127	directory.animals4life.org	Simple AD	Small	Not applicable	Active	Jul 10, 2024

On repeating the steps from creating sever certificate on our local machine, we have modified the command to :

```
“frenzy@pop-os:/tmp/easy-rsa/easyrsa3$ ./easyrsa build-server-full directory.animals4life.org nopass”
```

SUCCESS!



Client IPv4 CIDR Info
The IP address range, in CIDR notation, from which client IP addresses are allocated.
10.0.0.0/22
CIDR block cannot be larger than /12 or smaller than /22.

Authentication information Info

Server certificate ARN
The server certificate must be provisioned with or imported into AWS Certificate Manager (ACM).
Select certificate
arn:aws:acm:us-east-1:389467740328:certificate/2eee812c-cb9f-46cd-8504-b40d9c24069d
directory.animals4life.org

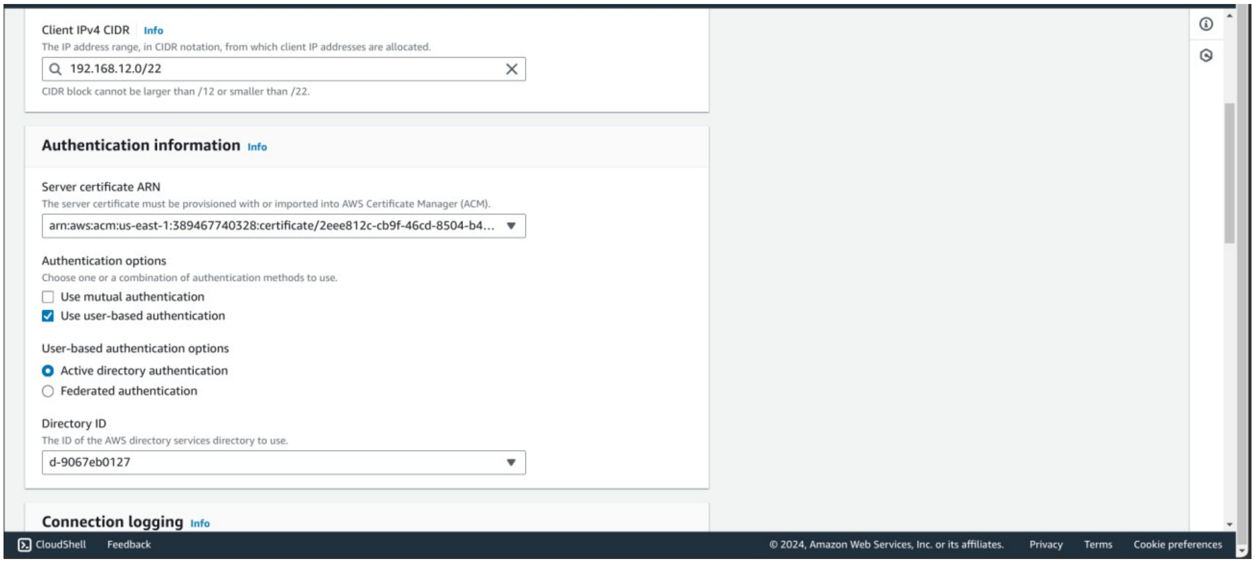
Connection logging Info

Enable log details on client connections
Use connection logs to have forensics on when clients attempted to connect.

Client connect handler Info

Enable client connect handler

Lets move to client vpn endpoint creation!



Client IPv4 CIDR Info

The IP address range, in CIDR notation, from which client IP addresses are allocated.

192.168.12.0/22

CIDR block cannot be larger than /12 or smaller than /22.

Authentication information Info

Server certificate ARN

The server certificate must be provisioned with or imported into AWS Certificate Manager (ACM).

arn:aws:acm:us-east-1:389467740328:certificate/2eee812c-cb9f-46cd-8504-b4...

Authentication options

Choose one or a combination of authentication methods to use.

☐ Use mutual authentication

☑ Use user-based authentication

User-based authentication options

● Active directory authentication

○ Federated authentication

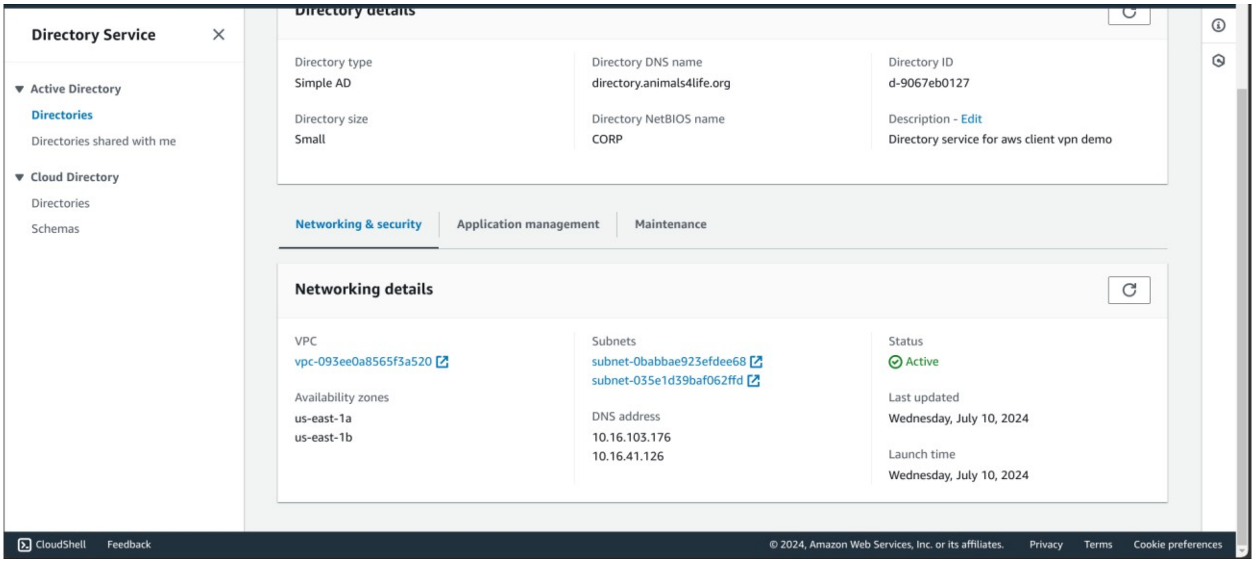
Directory ID

The ID of the AWS directory services directory to use.

d-9067eb0127

Connection logging Info

For DNS server , it is essential when a user needs to resolve any connectivity issues.
We will use simple AD service for dns addressing:



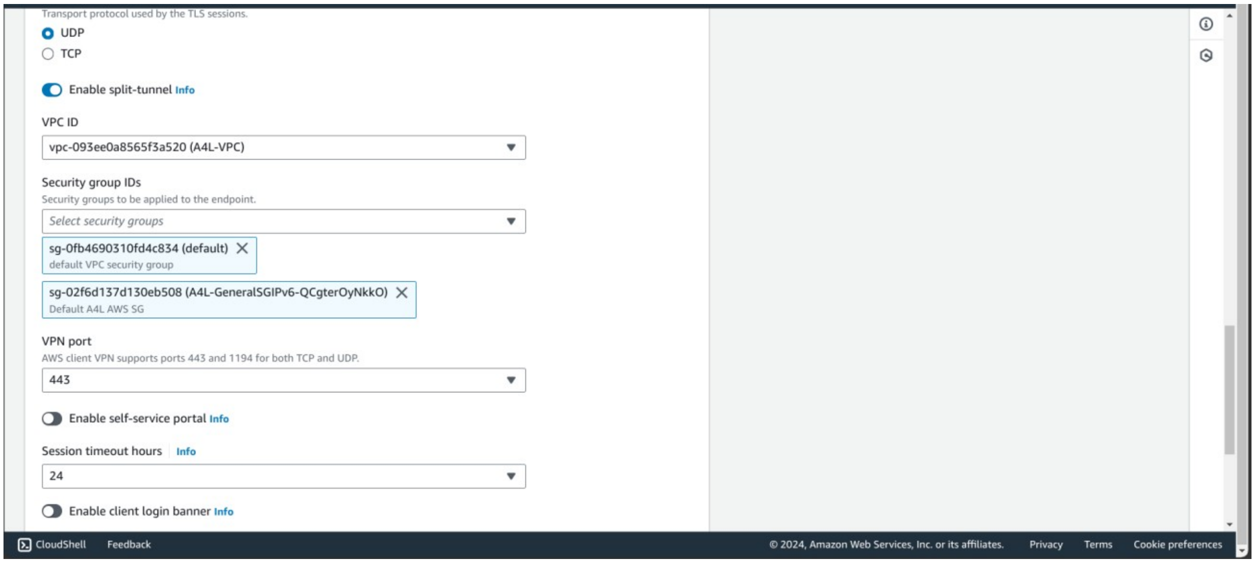
Directory details

Directory type Simple AD	Directory DNS name directory.animals4life.org	Directory ID d-9067eb0127
Directory size Small	Directory NetBIOS name CORP	Description - Edit Directory service for aws client vpn demo

Networking details

VPC vpc-093ee0a8565f3a520	Subnets subnet-0babbae923efdee68 subnet-035e1d39baf062ffd	Status Active
Availability zones us-east-1a us-east-1b	DNS address 10.16.103.176 10.16.41.126	Last updated Wednesday, July 10, 2024
		Launch time Wednesday, July 10, 2024

The dns service is provided to our client end vpn within the vpc by the ad service itself!
We would also want our user to access the internet resources without the vpn . The vpn is
configured in such a way that, it allows the connection to attach itself only to the resource
that lie inside within the VPC.
To achieve that functionality, *split tunnel* is enabled.



Transport protocol used by the TLS sessions.

● UDP

○ TCP

Enable split-tunnel Info

VPC ID

vpc-093ee0a8565f3a520 (A4L-VPC)

Security group IDs

Security groups to be applied to the endpoint.

Select security groups

sg-0fb4690310fd4c834 (default) default VPC security group

sg-02f6d137d130eb508 (A4L-GeneralSGIPv6-QCgterOyNkkO) Default A4L AWS SG

VPN port

AWS client VPN supports ports 443 and 1194 for both TCP and UDP.

443

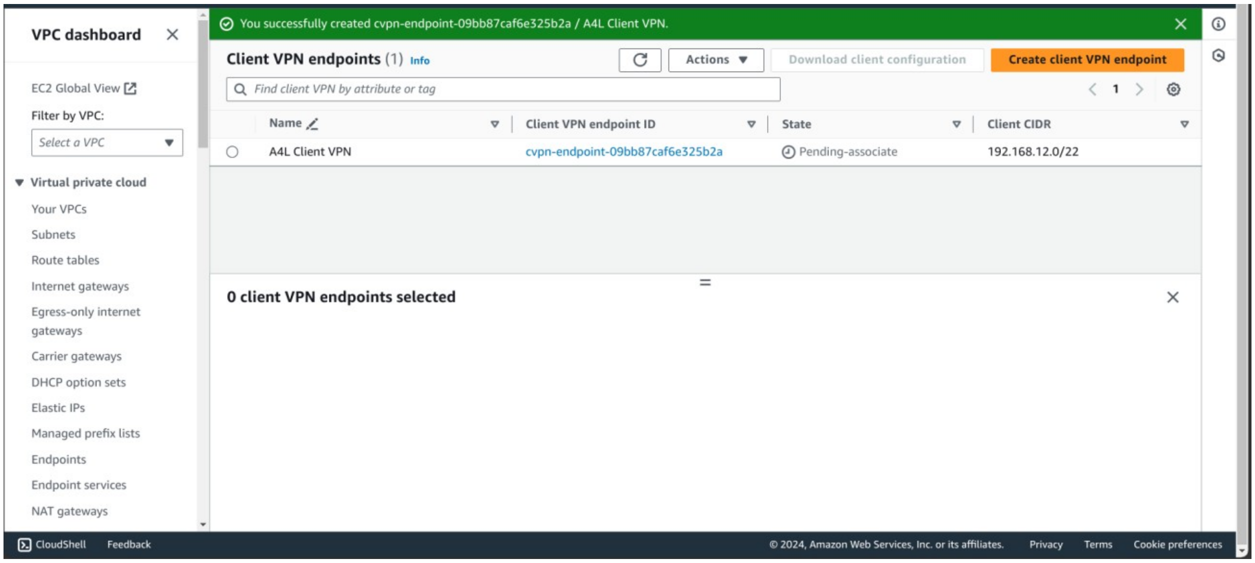
Enable self-service portal Info

Session timeout hours Info

24

Enable client login banner Info

SUCCESS!!



You successfully created cvpn-endpoint-09bb87caf6e325b2a / A4L Client VPN.

Client VPN endpoints (1) Info

Name	Client VPN endpoint ID	State	Client CIDR
A4L Client VPN	cvpn-endpoint-09bb87caf6e325b2a	Pending-associate	192.168.12.0/22

0 client VPN endpoints selected

The state remains pending association as it still needs to be attached with a subnet withir a VPC.

Client VPN Endpoint State

- **Pending-associate**: This state indicates that the Client VPN endpoint is waiting to be associated with a subnet in your VPC (Virtual Private Cloud).

Association with a Subnet

- **Subnet Association**: You need to link the Client VPN endpoint with a subnet in you VPC. This subnet will host the network interfaces for the Client VPN.

Network Interfaces

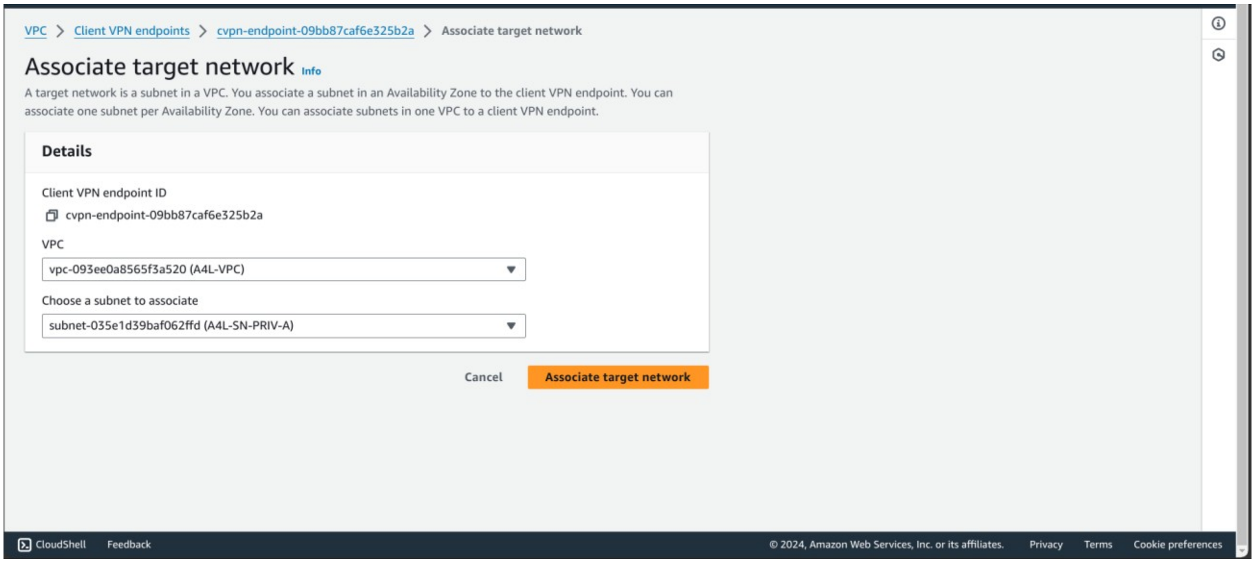
- **Client VPN Interfaces**: These are virtual network interfaces created in the associated subnet. They facilitate communication between the VPN clients and the VPC.

Traffic Flow

- **Traffic from Clients to VPC**: Traffic from the VPN clients will exit through these interfaces into the VPC.
- **Traffic from VPC to Clients**: Traffic destined for the VPN clients from within the VPC will enter through these interfaces.

(Step-9)

Creating association...



VPC > Client VPN endpoints > cvpn-endpoint-09bb87caf6e325b2a > Associate target network

Associate target network Info

A target network is a subnet in a VPC. You associate a subnet in an Availability Zone to the client VPN endpoint. You can associate one subnet per Availability Zone. You can associate subnets in one VPC to a client VPN endpoint.

Details

Client VPN endpoint ID

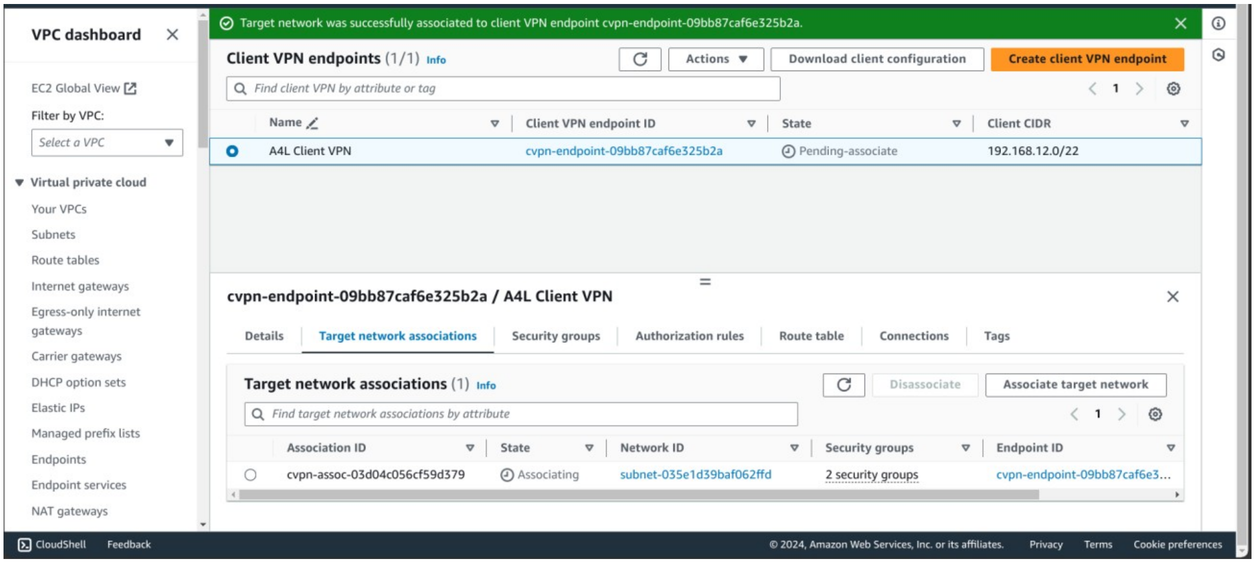
cvpn-endpoint-09bb87caf6e325b2a

VPC

vpc-093ee0a8565f3a520 (A4L-VPC)

Choose a subnet to associate

subnet-035e1d39baf062ffd (A4L-SN-PRIV-A)



Target network was successfully associated to client VPN endpoint cvpn-endpoint-09bb87caf6e325b2a.

Client VPN endpoints (1/1) Info

Actions
Download client configuration
Create client VPN endpoint
Find client VPN by attribute or tag

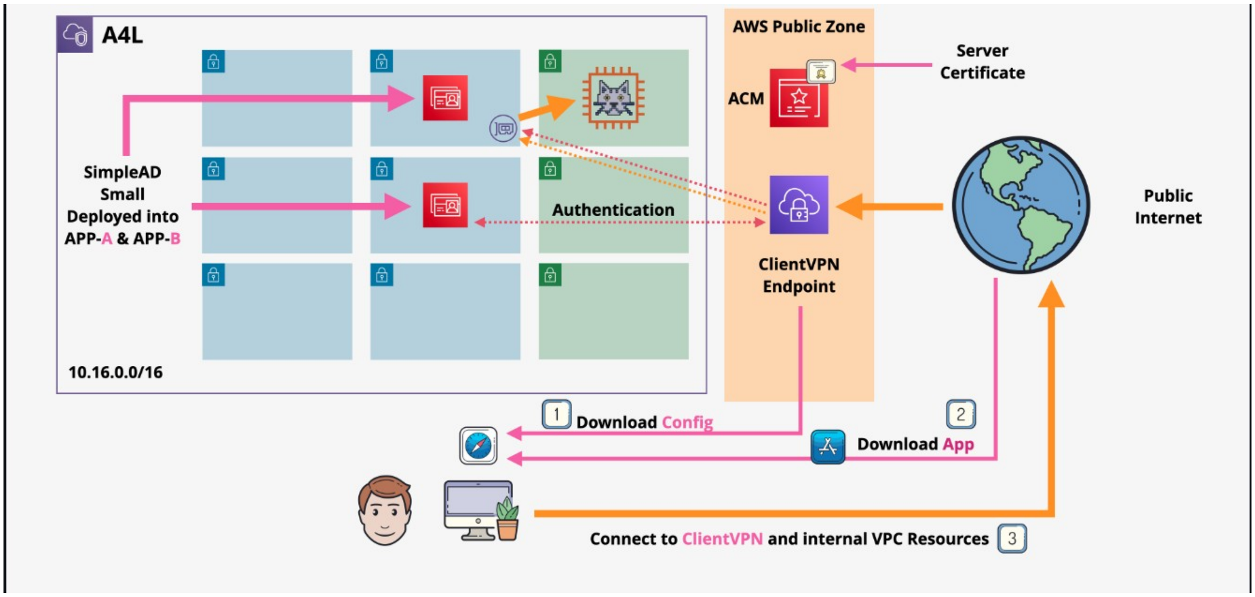
	Name	Client VPN endpoint ID	State	Client CIDR
●	A4L Client VPN	cvpn-endpoint-09bb87caf6e325b2a	Pending-associate	192.168.12.0/22

cvpn-endpoint-09bb87caf6e325b2a / A4L Client VPN

Target network associations (1) Info

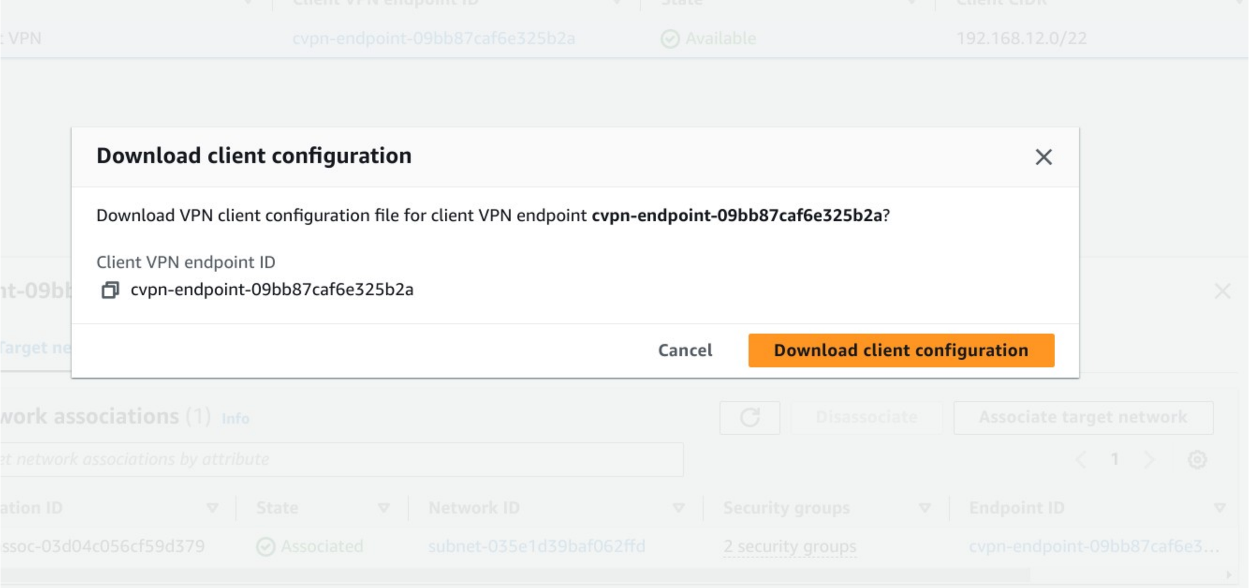
	Association ID	State	Network ID	Security groups	Endpoint ID
○	cvpn-assoc-03d04c056cf59d379	Associating	subnet-035e1d39baf062ffd	2 security groups	cvpn-endpoint-09bb87caf6e3...

We are here....



Wait for the target association status to change to “Associated”...

After that, download client configuration settings...



Download client configuration

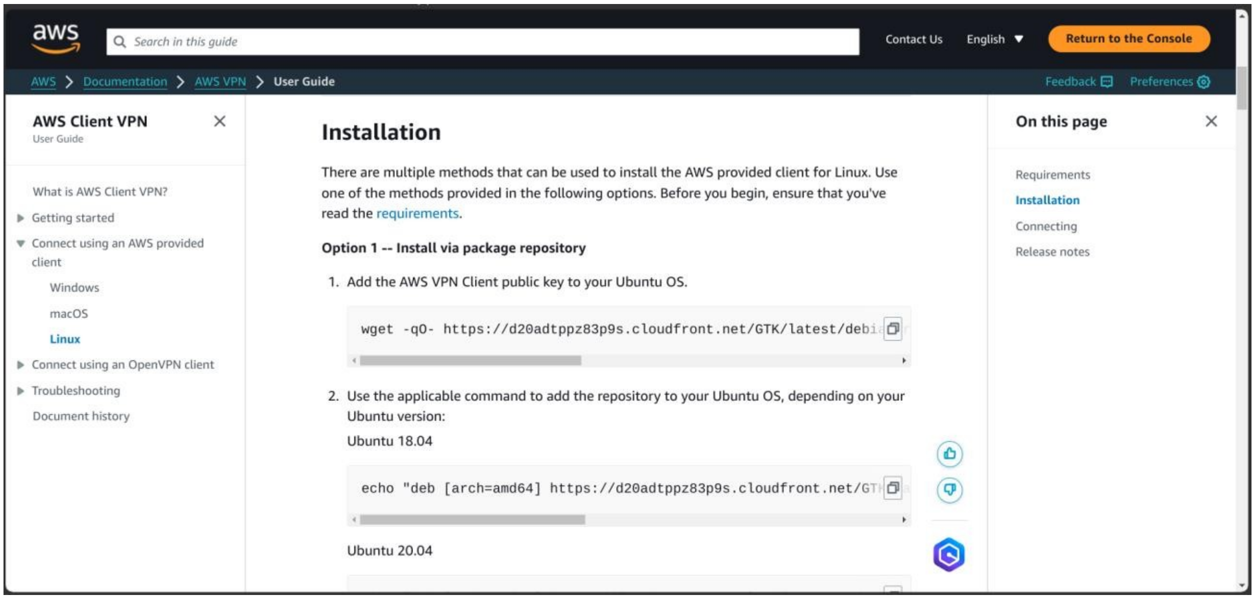
Download VPN client configuration file for client VPN endpoint **cvpn-endpoint-09bb87caf6e325b2a**?

Client VPN endpoint ID

cvpn-endpoint-09bb87caf6e325b2a

....

We are required to download the client vpn ...



Installation

There are multiple methods that can be used to install the AWS provided client for Linux. Use one of the methods provided in the following options. Before you begin, ensure that you've read the requirements.

Option 1 -- Install via package repository

1. Add the AWS VPN Client public key to your Ubuntu OS.

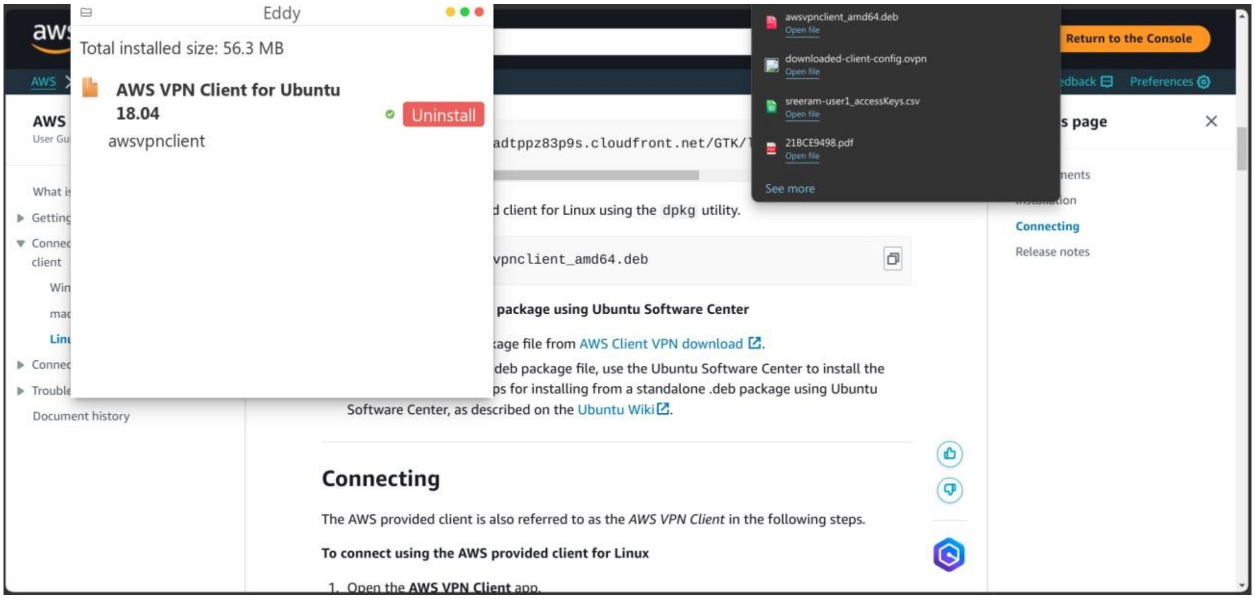
```
wget -qO- https://d20adtppz83p9s.cloudfront.net/GTK/latest/debi
```

2. Use the applicable command to add the repository to your Ubuntu OS, depending on your Ubuntu version:

   Ubuntu 18.04

```
echo "deb [arch=amd64] https://d20adtppz83p9s.cloudfront.net/GT
```

   Ubuntu 20.04



Total installed size: 56.3 MB

AWS VPN Client for Ubuntu 18.04

awsvpnclient

adtppz83p9s.cloudfront.net/GTK/l

d client for Linux using the dpkg utility.

```
vpnclient_amd64.deb
```

package using Ubuntu Software Center

kage file from AWS Client VPN download.

deb package file, use the Ubuntu Software Center to install the ps for installing from a standalone .deb package using Ubuntu Software Center, as described on the Ubuntu Wiki.

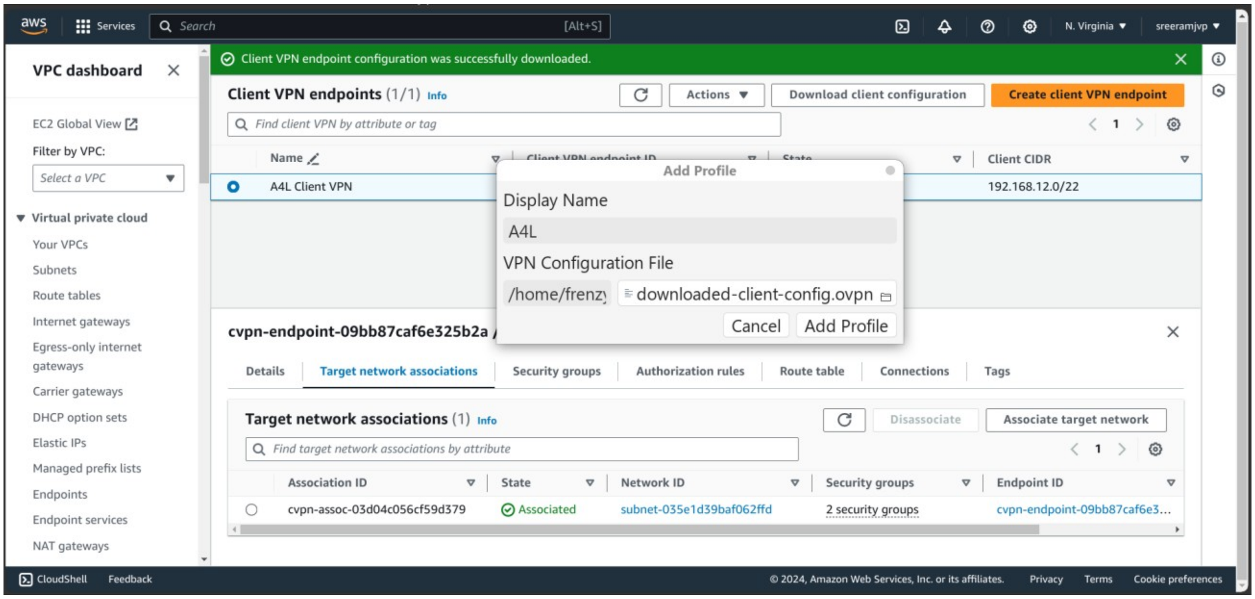
Connecting

The AWS provided client is also referred to as the *AWS VPN Client* in the following steps.

To connect using the AWS provided client for Linux

1. Open the **AWS VPN Client** app.

Associate openVPN profile to the AWS VPN CLIENT...



Client VPN endpoint configuration was successfully downloaded.

Client VPN endpoints (1/1) Info

Name	Client VPN endpoint ID	State	Client CIDR
A4L Client VPN			192.168.12.0/22

Add Profile

Display Name: A4L

VPN Configuration File: /home/frenzy downloaded-client-config.ovpn

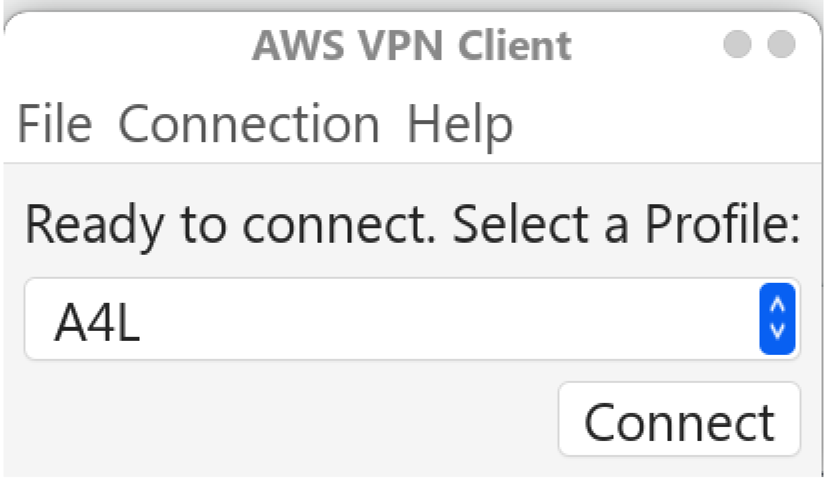
Cancel | Add Profile

cvpn-endpoint-09bb87caf6e325b2a /

Target network associations (1) Info

Association ID	State	Network ID	Security groups	Endpoint ID
cvpn-assoc-03d04c056cf59d379	Associated	subnet-035e1d39baf062ffd	2 security groups	cvpn-endpoint-09bb87caf6e3...

Click on connect...and enter the simple ad credentials that was created at the beginning..



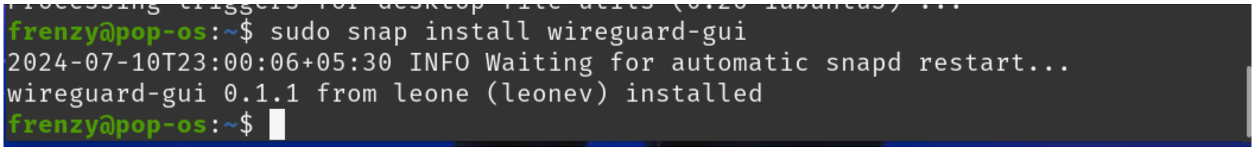
Ready to connect. Select a Profile:

A4L

Well....there seems to be another problem with the dependencies associated with the .de file. On further digging, its been found that, AWS is no longer supporting AWS client vpn fo Ubuntu version, and the client would only run on older versions of lib, which would be a no no due to security concerns.

The alternative would be to use, wireguard with gui installed using snapd, which is a proprietary software packaging by *canonical softwares.*

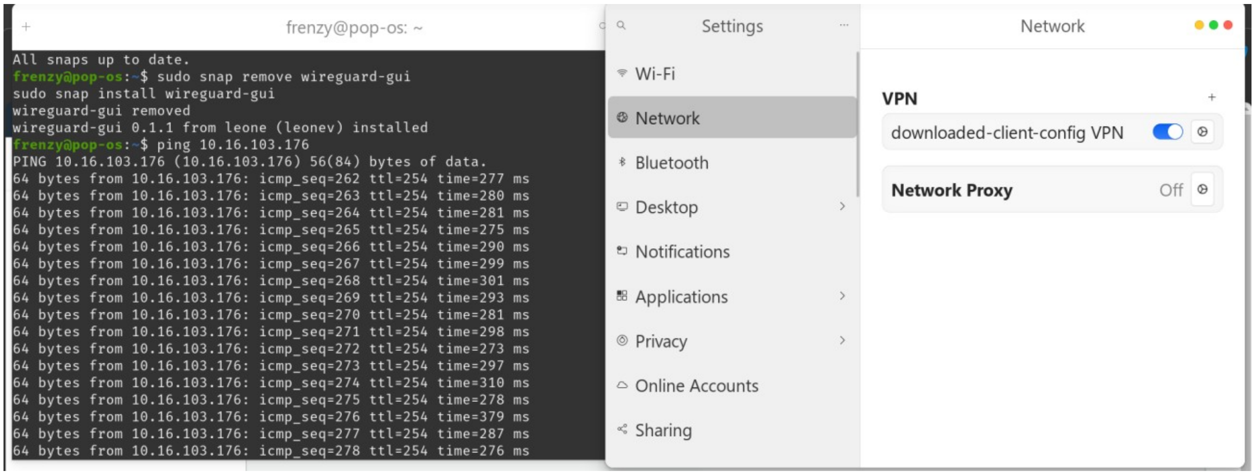
Wireguard is similar to AWS client vpn, and is supported and updated. The repository is updated .



```
frenzy@pop-os:~$ sudo snap install wireguard-gui
2024-07-10T23:00:06+05:30 INFO Waiting for automatic snapd restart...
wireguard-gui 0.1.1 from leone (leonev) installed
frenzy@pop-os:~$ 
```

After adding in user-name and password , go to the vpn client settings and select authorization roles, and approve,

SUCCESS!!!



frenzy@pop-os: ~

```
All snaps up to date.
frenzy@pop-os:~$ sudo snap remove wireguard-gui
sudo snap install wireguard-gui
wireguard-gui removed
wireguard-gui 0.1.1 from leone (leonev) installed
frenzy@pop-os:~$ ping 10.16.103.176
PING 10.16.103.176 (10.16.103.176) 56(84) bytes of data.
64 bytes from 10.16.103.176: icmp_seq=262 ttl=254 time=277 ms
64 bytes from 10.16.103.176: icmp_seq=263 ttl=254 time=280 ms
64 bytes from 10.16.103.176: icmp_seq=264 ttl=254 time=281 ms
64 bytes from 10.16.103.176: icmp_seq=265 ttl=254 time=275 ms
64 bytes from 10.16.103.176: icmp_seq=266 ttl=254 time=290 ms
64 bytes from 10.16.103.176: icmp_seq=267 ttl=254 time=299 ms
64 bytes from 10.16.103.176: icmp_seq=268 ttl=254 time=301 ms
64 bytes from 10.16.103.176: icmp_seq=269 ttl=254 time=293 ms
64 bytes from 10.16.103.176: icmp_seq=270 ttl=254 time=281 ms
64 bytes from 10.16.103.176: icmp_seq=271 ttl=254 time=298 ms
64 bytes from 10.16.103.176: icmp_seq=272 ttl=254 time=273 ms
64 bytes from 10.16.103.176: icmp_seq=273 ttl=254 time=297 ms
64 bytes from 10.16.103.176: icmp_seq=274 ttl=254 time=310 ms
64 bytes from 10.16.103.176: icmp_seq=275 ttl=254 time=278 ms
64 bytes from 10.16.103.176: icmp_seq=276 ttl=254 time=379 ms
64 bytes from 10.16.103.176: icmp_seq=277 ttl=254 time=287 ms
64 bytes from 10.16.103.176: icmp_seq=278 ttl=254 time=276 ms
```

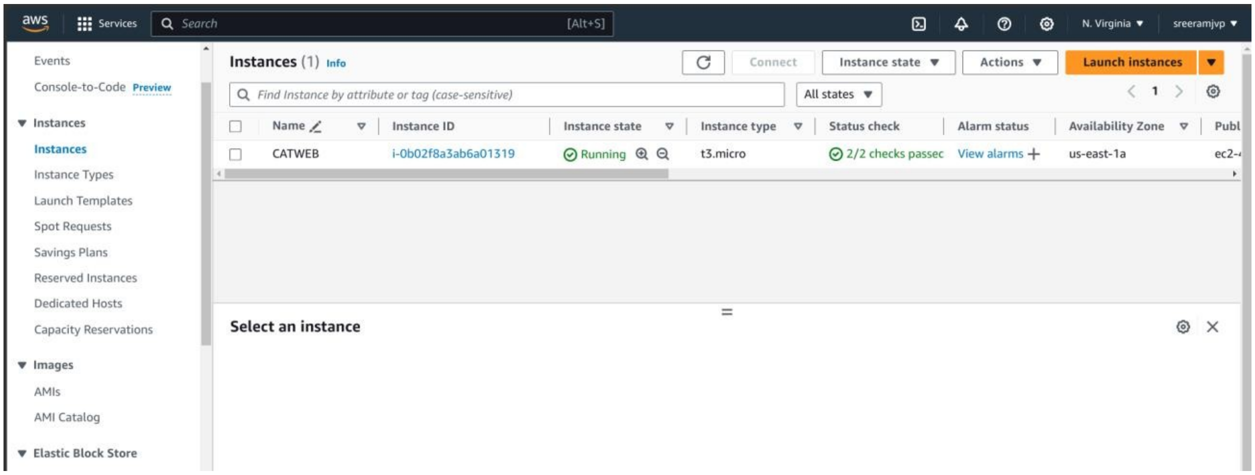
Network

VPN

downloaded-client-config VPN ●

Network Proxy Off

Another way to verify if our VPN is working is to select Ec2 instance:



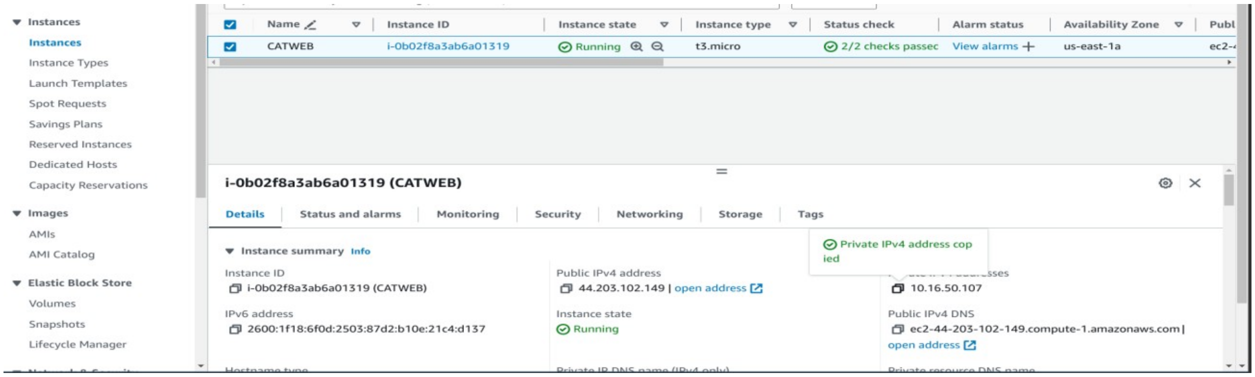
Instances (1) Info

Find Instance by attribute or tag (case-sensitive) | All states ▾

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
□	CATWEB	i-0b02f8a3ab6a01319	Running	t3.micro	2/2 checks passec	View alarms +	us-east-1a	ec2-4

Select an instance

We can see there is a an ec2 instance running hosting a static webiste. Choose the ec2 instance and copy the private IPV4 instance ip address:



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
☑	CATWEB	i-0b02f8a3ab6a01319	Running	t3.micro	2/2 checks passec	View alarms +	us-east-1a	ec2-4

i-0b02f8a3ab6a01319 (CATWEB)

Instance summary Info

Instance ID: i-0b02f8a3ab6a01319 (CATWEB)

Public IPv4 address: 44.203.102.149 | open address

Private IPv4 address copied

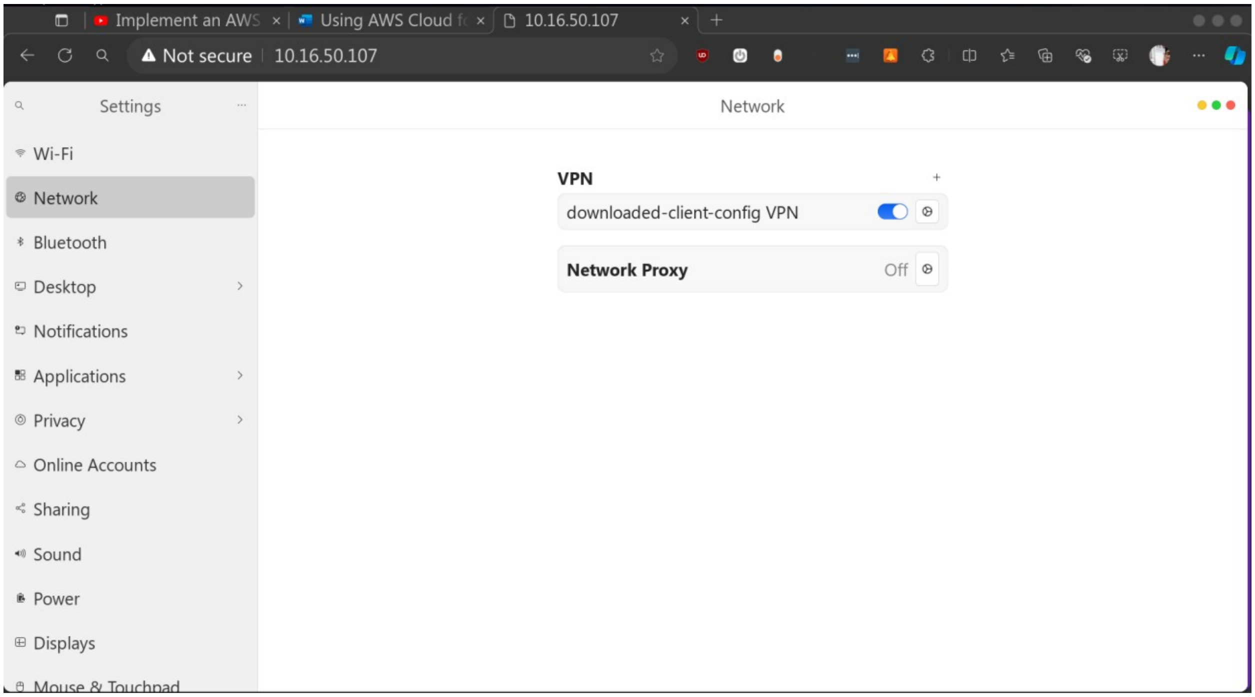
10.16.50.107

IPv6 address: 2600:1f18:6f0d:2503:87d2:b10e:21c4:d137

Instance state: Running

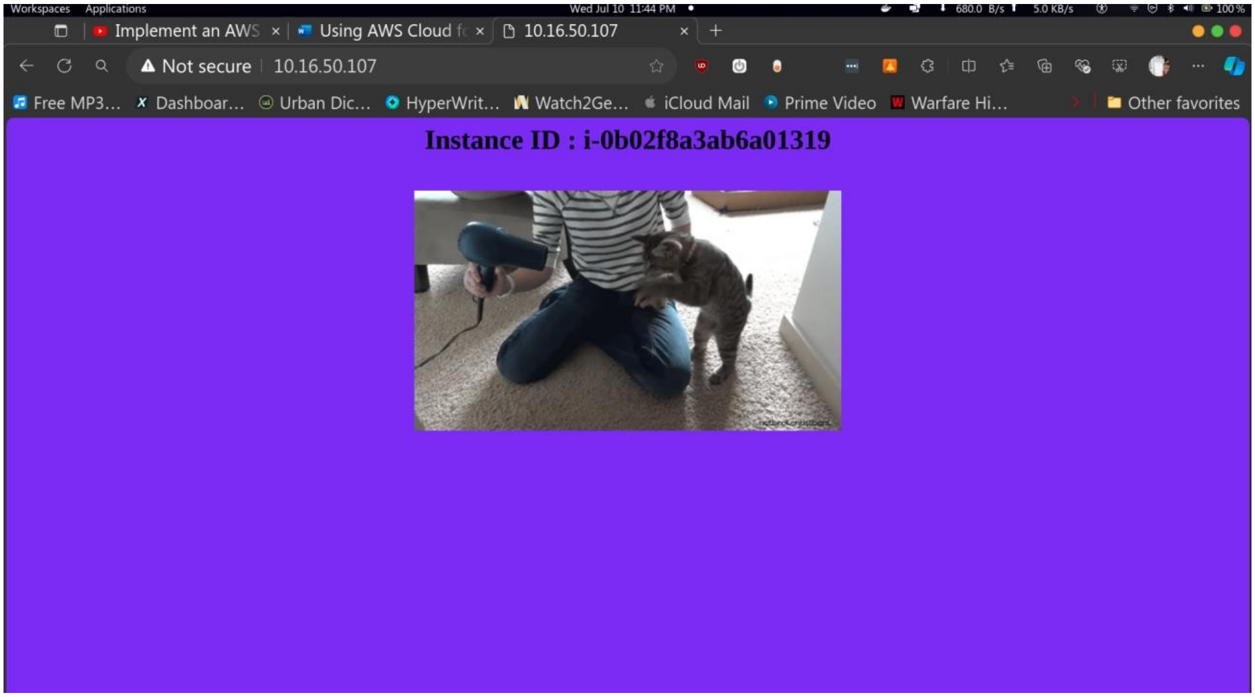
Public IPv4 DNS: ec2-44-203-102-149.compute-1.amazonaws.com | open address

VPN ON:



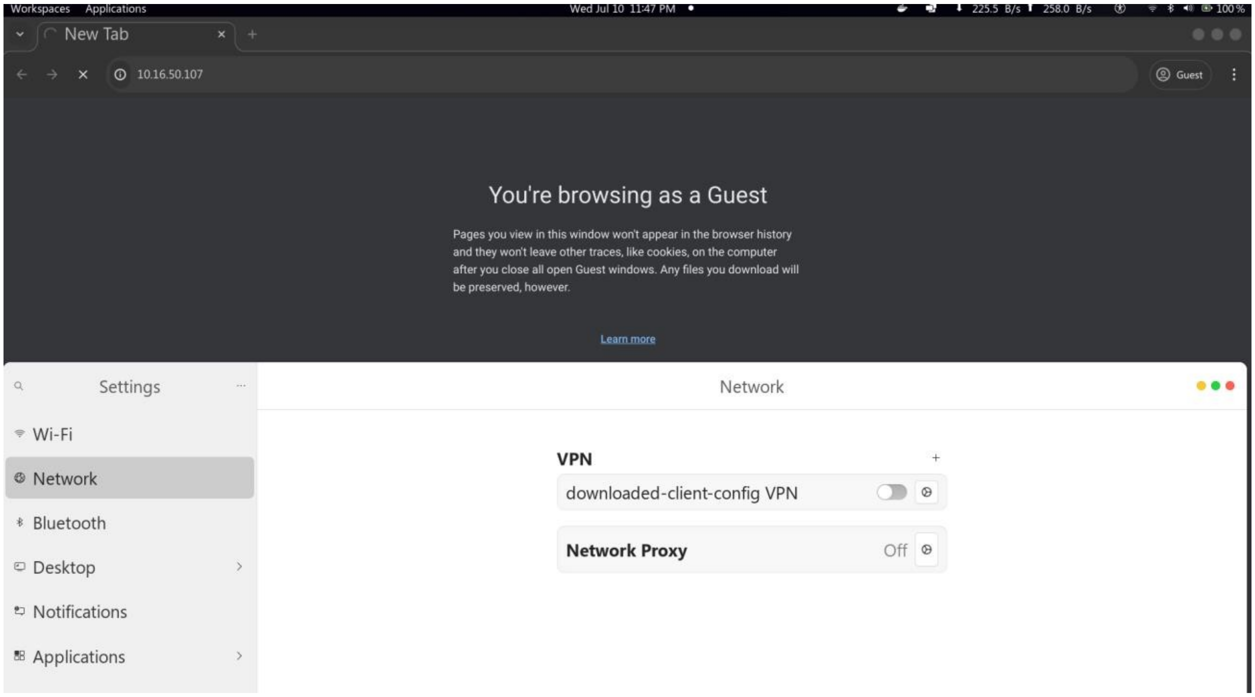
downloaded-client-config VPN ●

Network Proxy Off



Instance ID : i-0b02f8a3ab6a01319

With vpn off:



You're browsing as a Guest

Pages you view in this window won't appear in the browser history and they won't leave other traces, like cookies, on the computer after you close all open Guest windows. Any files you download will be preserved, however.

Learn more

VPN

downloaded-client-config VPN

Network Proxy Off

....

“Successfully set up a client side vpn using simple AD service, AWS cli, directory service, vpc, vpn, and verified it using an ec2 instance within the vpc network.”