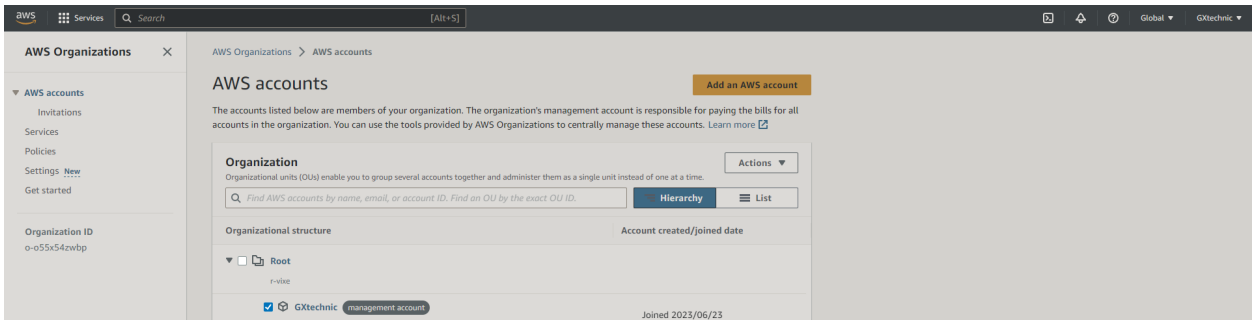
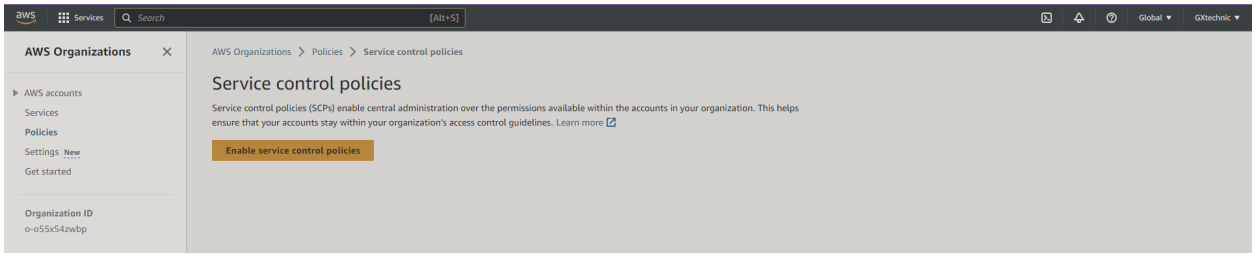
Mack Dirks
Seattle-ops-301d8

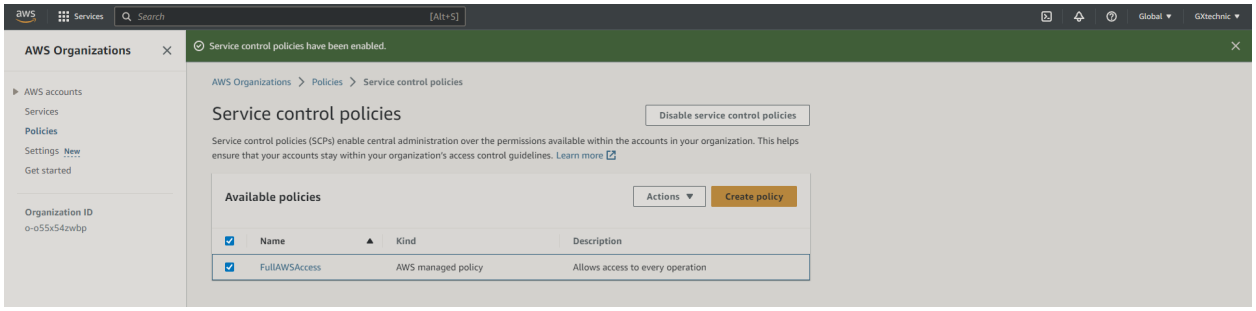
Final Project 301
GXtechnic AWS Site-to-Site VPN Solution for GreenGenius

Creating a management account on AWS for GXtechnic:

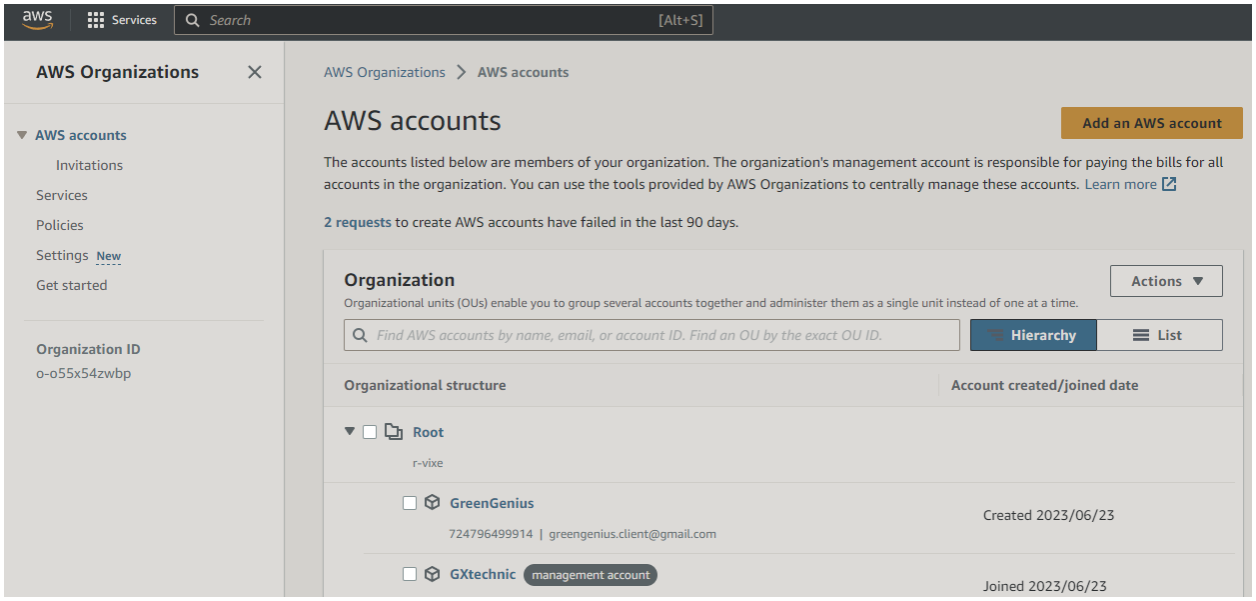


Enabling “Service Control Policies” for our organization:

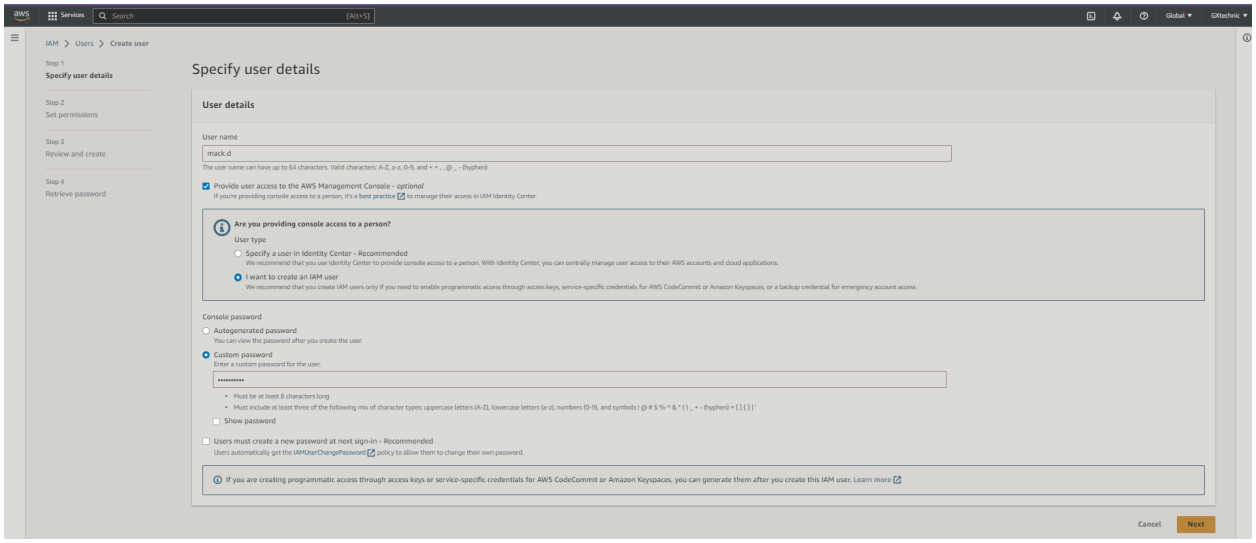




Creating an AWS account for GreenGenius:

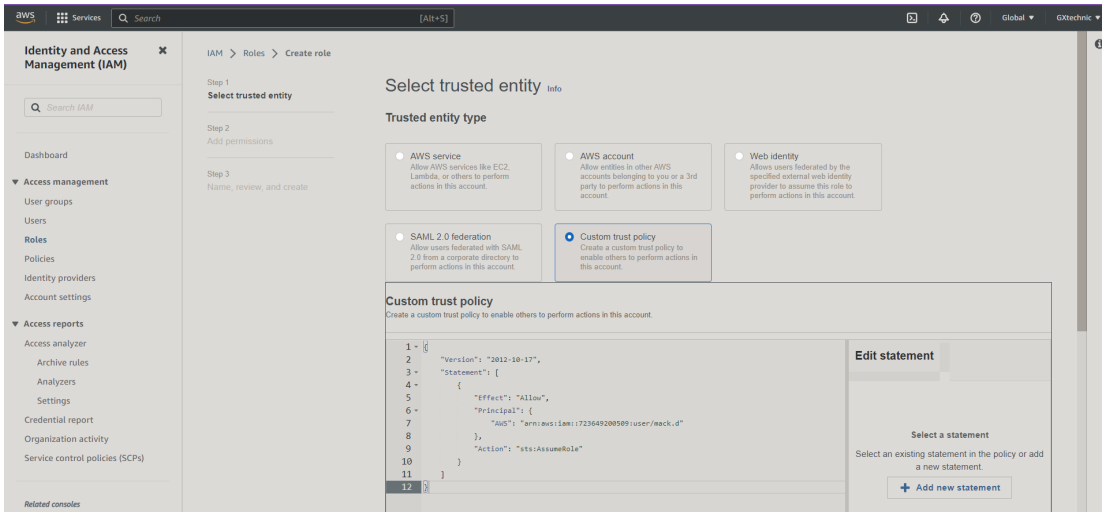


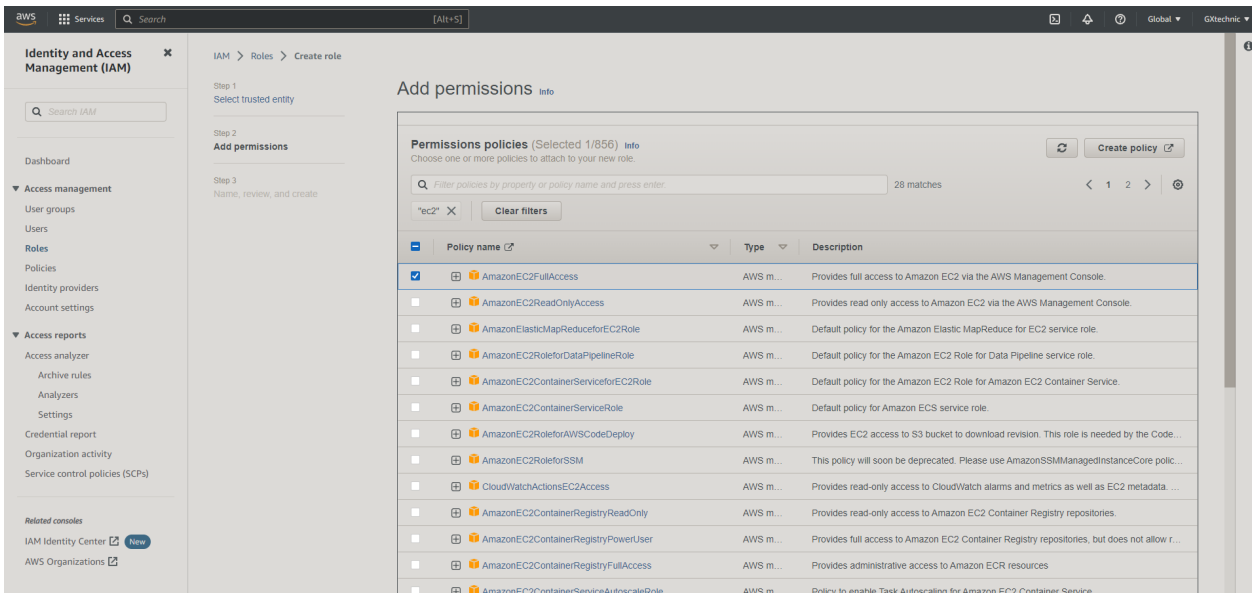
Creating an IAM user for GXtechnic in order to be able to switch users and accounts later:

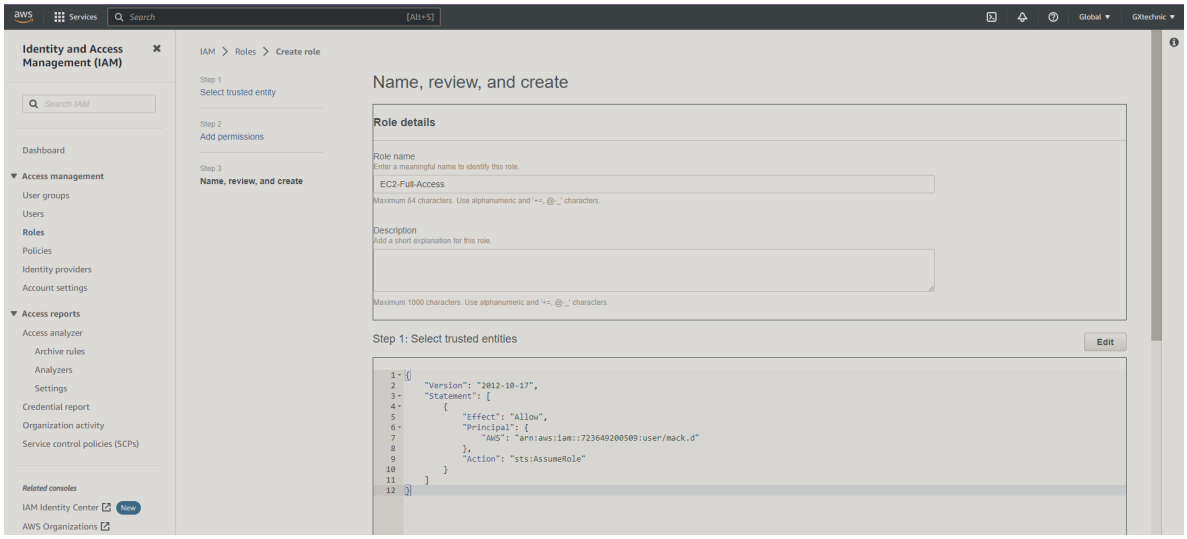


Giving IAM permissions to our newly created user (Trust Policy for the Role):

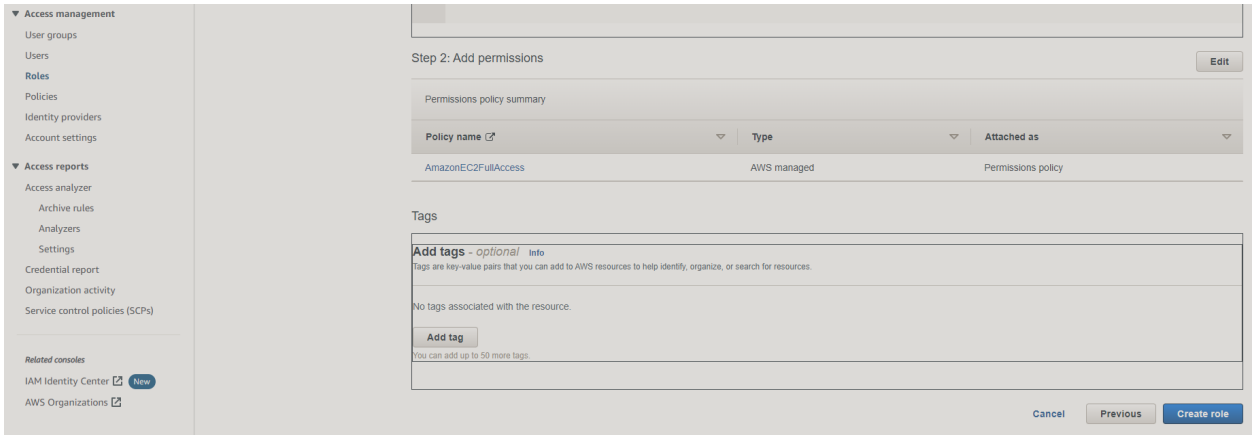
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::723649200509:user/mack.d"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```


Identity and Access Management (IAM)
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Select trusted entity Info
Trusted entity type
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.
Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Principal": {
"AWS": "arn:aws:iam::723649200509:user/mack.d"
},
"Action": "sts:AssumeRole"
}
]
}
Edit statement
Select a statement
Select an existing statement in the policy or add a new statement.
Add new statement

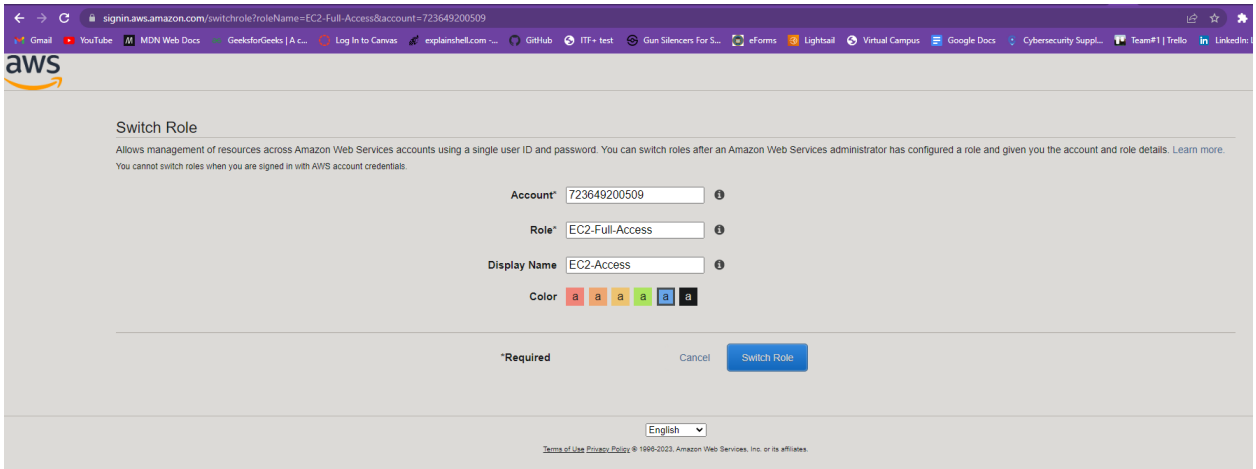

Identity and Access Management (IAM)
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Add permissions Info
Permissions policies (Selected 1/856) Info
Choose one or more policies to attach to your new role.
Create policy
Filter policies by property or policy name and press enter.
28 matches
"ec2"
Clear filters
Policy name
Type
Description
AmazonEC2FullAccess
Provides full access to Amazon EC2 via the AWS Management Console.
AmazonEC2ReadOnlyAccess
Provides read only access to Amazon EC2 via the AWS Management Console.
AmazonElasticMapReduceforEC2Role
Default policy for the Amazon Elastic MapReduce for EC2 service role.
AmazonEC2RoleforDataPipelineRole
Default policy for the Amazon EC2 Role for Data Pipeline service role.
AmazonEC2ContainerServiceforEC2Role
Default policy for the Amazon EC2 Role for Amazon EC2 Container Service.
AmazonEC2ContainerServiceRole
Default policy for Amazon ECS service role.
AmazonEC2RoleforAWSCodeDeploy
Provides EC2 access to S3 bucket to download revision. This role is needed by the Code...
AmazonEC2RoleforSSM
This policy will soon be deprecated. Please use AmazonSSMManagedInstanceCore polic...
CloudWatchActionsEC2Access
Provides read-only access to CloudWatch alarms and metrics as well as EC2 metadata. ...
AmazonEC2ContainerRegistryReadOnly
Provides read-only access to Amazon EC2 Container Registry repositories.
AmazonEC2ContainerRegistryPowerUser
Provides full access to Amazon EC2 Container Registry repositories, but does not allow r...
AmazonEC2ContainerRegistryFullAccess
Provides administrative access to Amazon ECR resources

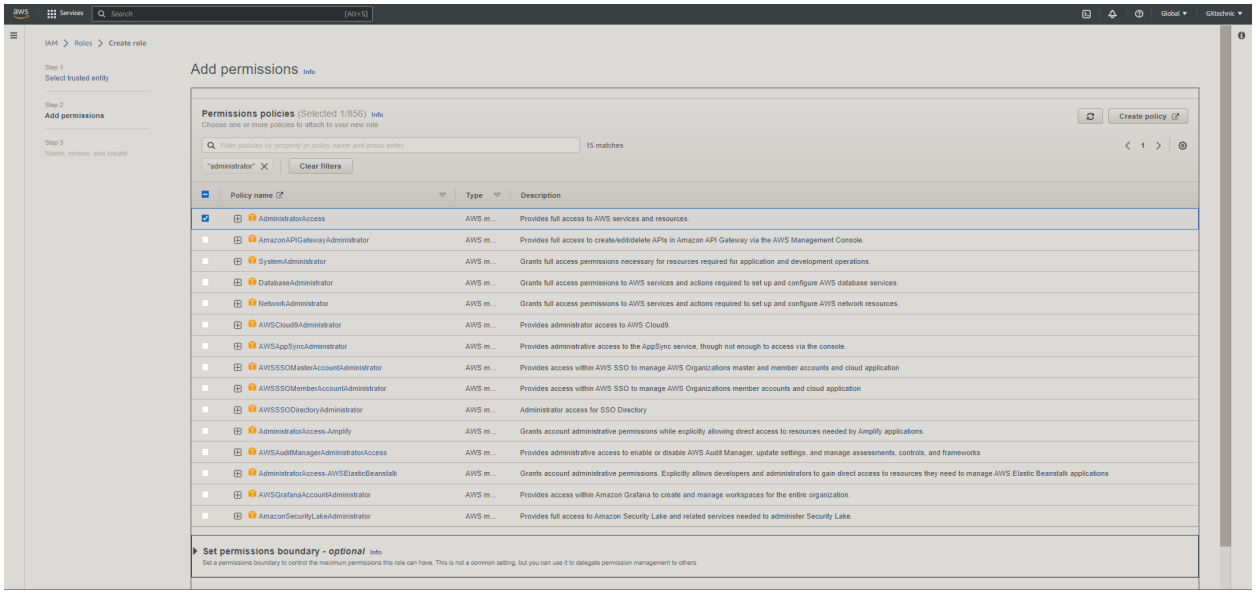

Identity and Access Management (IAM)
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
EC2-Full-Access
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.
Step 1: Select trusted entities
Edit
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Principal": {
"AWS": "arn:aws:iam::723649200509:user/mack.d"
},
"Action": "sts:AssumeRole"
}
]
}

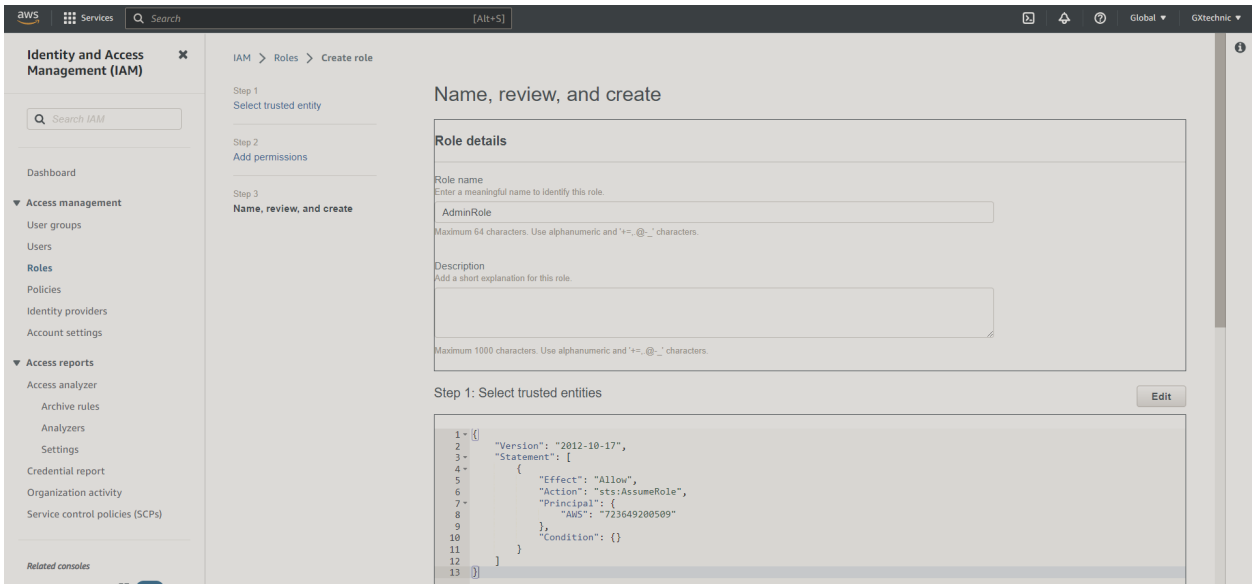
Creating the Role:

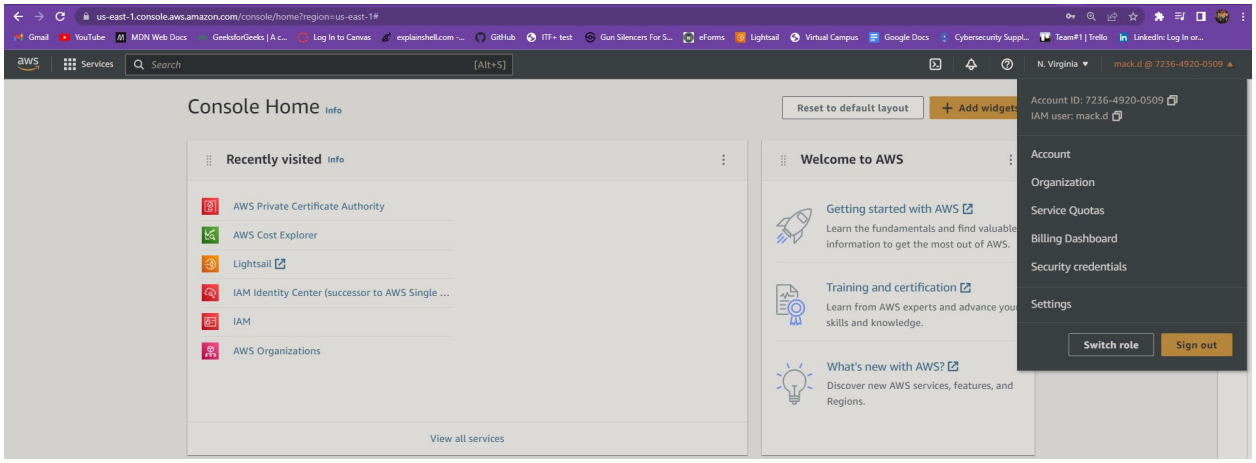


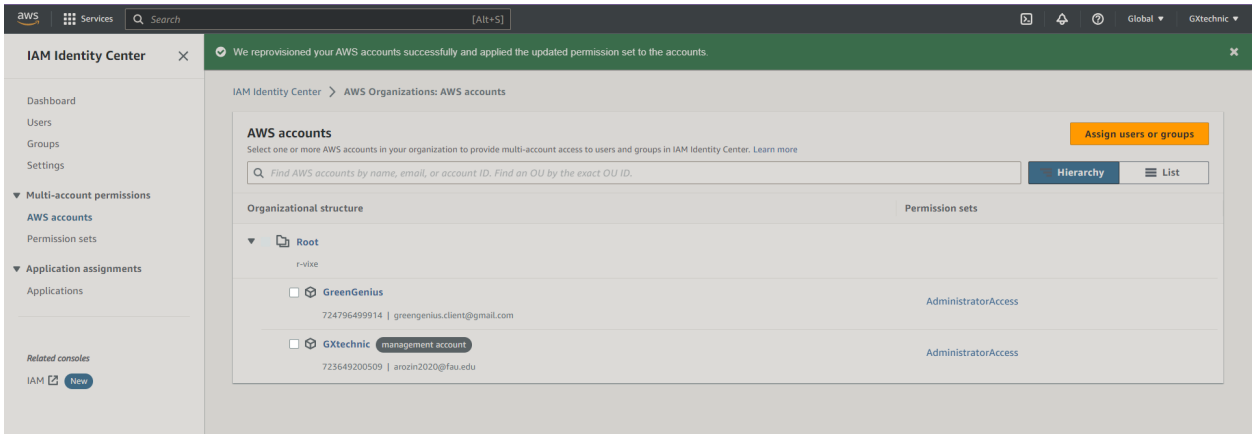
Assuming the role and giving the newly created user (mack.d) all the necessary permissions:



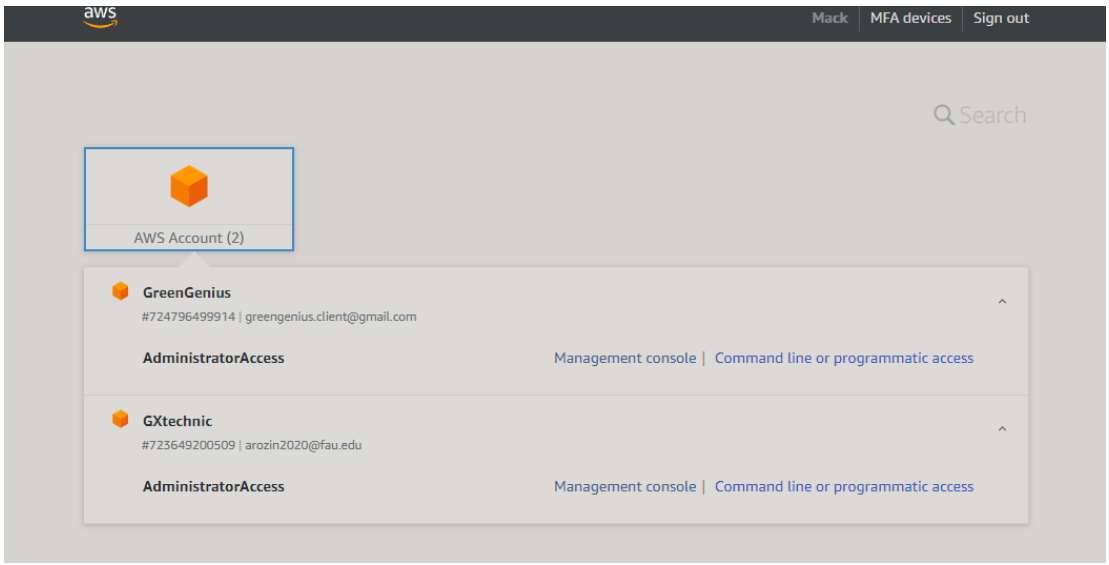

IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Add permissions Info
Permissions policies (Selected 1/856) Info
Choose one or more policies to attach to your new role.
Create policy
15 matches
"administrator"
Clear filters
Policy name
Type
Description
AdministratorAccess
AWS m...
Provides full access to AWS services and resources.
AmazonAPIGatewayAdministrator
Provides full access to create/edit/delete APIs in Amazon API Gateway via the AWS Management Console.
SystemAdministrator
Grants full access permissions necessary for resources required for application and development operations.
DatabaseAdministrator
Grants full access permissions to AWS services and actions required to set up and configure AWS database services.
NetworkAdministrator
Grants full access permissions to AWS services and actions required to set up and configure AWS network resources.
AWSCloud9Administrator
Provides administrator access to AWS Cloud9.
AWSAppSyncAdministrator
Provides administrative access to the AppSync service, though not enough to access via the console.
AWSSSOMasterAccountAdministrator
Provides access within AWS SSO to manage AWS Organizations master and member accounts and cloud application
AWSSSOMemberAccountAdministrator
Provides access within AWS SSO to manage AWS Organizations member accounts and cloud application
AWSSSODirectoryAdministrator
Administrator access for SSO Directory
AdministratorAccess-Amplify
Grants account administrative permissions while explicitly allowing direct access to resources needed by Amplify applications.
AWSAuditManagerAdministratorAccess
Provides administrative access to enable or disable AWS Audit Manager, update settings, and manage assessments, controls, and frameworks
AdministratorAccess-AWSElasticBeanstalk
Grants account administrative permissions. Explicitly allows developers and administrators to gain direct access to resources they need to manage AWS Elastic Beanstalk applications
AWSGrafanaAccountAdministrator
Provides access within Amazon Grafana to create and manage workspaces for the entire organization.
AmazonSecurityLakeAdministrator
Provides full access to Amazon Security Lake and related services needed to administer Security Lake.
Set permissions boundary - optional Info
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.


Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
Related consoles
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
AdminRole
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.
Step 1: Select trusted entities
Edit
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": "sts:AssumeRole",
"Principal": {
"AWS": "723649200509"
},
"Condition": {}
}
]
}

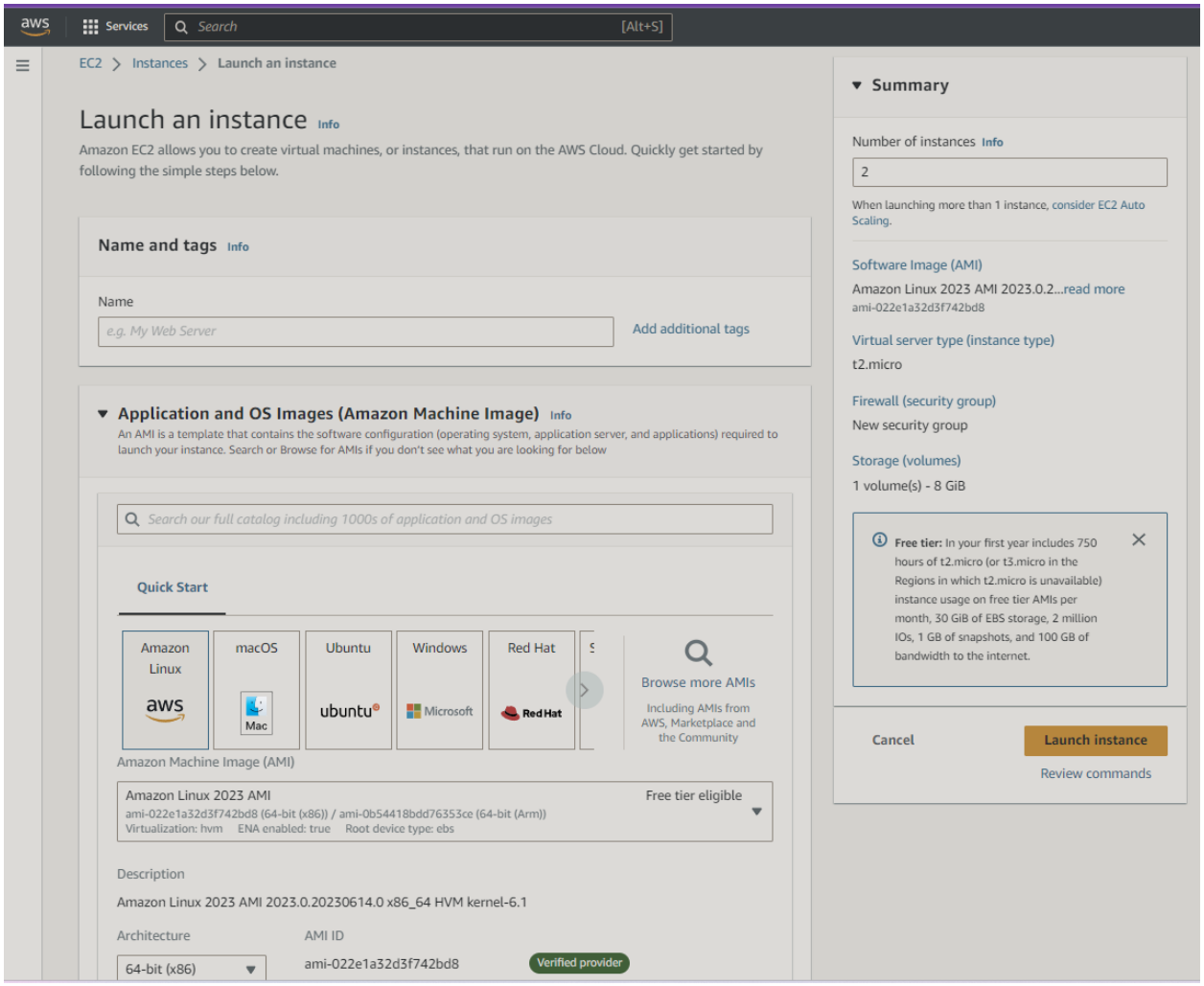

us-east-1.console.aws.amazon.com/console/home?region=us-east-1#
N. Virginia
mack.d @ 7236-4920-0509
Console Home Info
Reset to default layout
Recently visited Info
AWS Private Certificate Authority
AWS Cost Explorer
Lightsail
IAM Identity Center (successor to AWS Single ...
IAM
AWS Organizations
View all services
Welcome to AWS
Getting started with AWS
Learn the fundamentals and find valuable information to get the most out of AWS.
Training and certification
Learn from AWS experts and advance your skills and knowledge.
What's new with AWS?
Discover new AWS services, features, and Regions.
Account ID: 7236-4920-0509
IAM user: mack.d
Account
Organization
Service Quotas
Billing Dashboard
Security credentials
Settings
Switch role
Sign out



2 different accounts (for both Globex and GreenGenius) were successfully implemented and deployed into Identity and Access Management System for more secure control:



Launching 2 EC2 instances on the client side: Linux AMI (1 will play a role of an OpenSwan/Customer Gateway, and another will be their internal server)



Creating a security group with 2 rules for both instances:

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

- (●) Create security group
- ( ) Select existing security group

Security group name - *required*

launch-wizard-1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info

launch-wizard-1 created 2023-06-24T16:01:05.808Z

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 0.0.0.0/0) — Remove

Type Info	Protocol Info	Port range Info
ssh	TCP	22

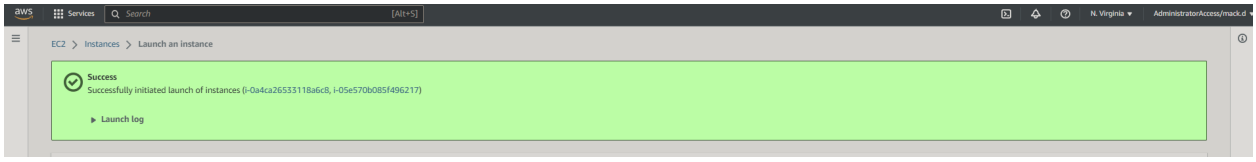
Source type Info	Source Info	Description - *optional* Info
Anywhere	Add CIDR, prefix list or security 0.0.0.0/0 ::/0	e.g. SSH for admin desktop

Security group rule 2 (ICMP, All, Multiple sources) — Remove

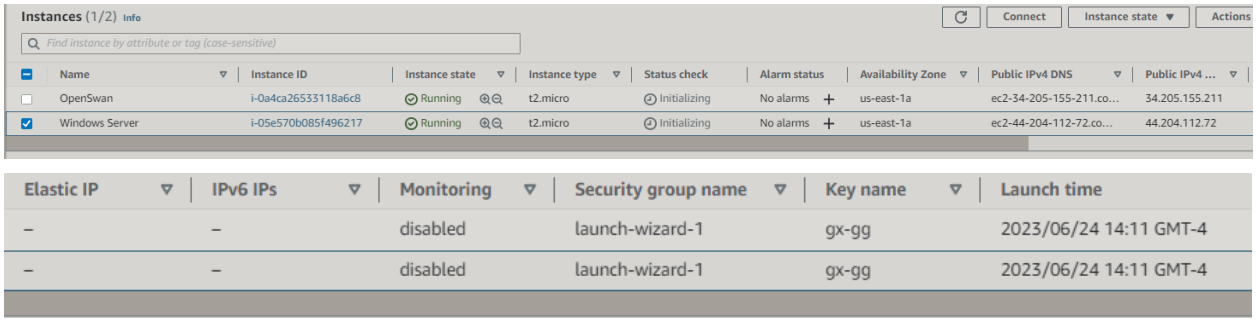
Type Info	Protocol Info	Port range Info
All ICMP - IPv4	ICMP	All

Source type Info	Source Info	Description - *optional* Info
Anywhere	Add CIDR, prefix list or security 0.0.0.0/0 ::/0	e.g. SSH for admin desktop

The security key was created. Name of the file: gx-gg.pem
After that, both client instances were launched successfully:



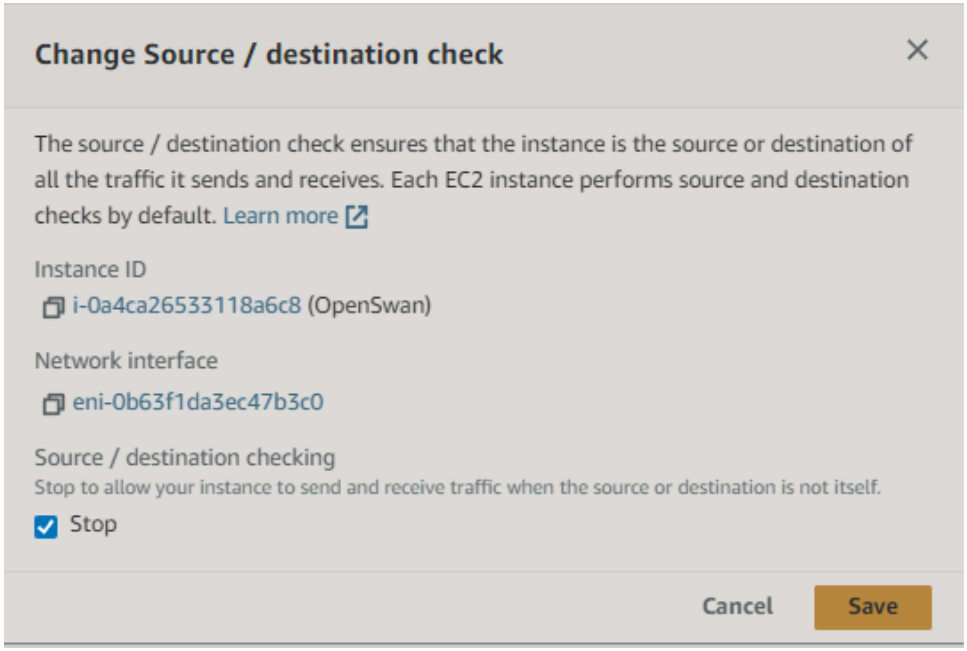
Instance 1: OpenSwan
Instance 2: Windows Server

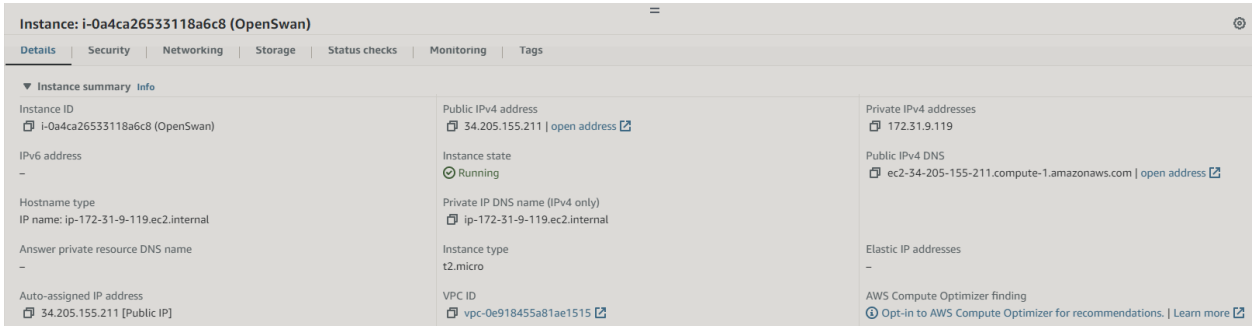


Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name	Launch time
OpenSwan	i-0a4ca26533118a6c8	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-34-205-155-211.co...	34.205.155.211	–	–	disabled	launch-wizard-1	gx-gg	2023/06/24 14:11 GMT-4
Windows Server	i-05e570b085f496217	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-44-204-112-72.co...	44.204.112.72	–	–	disabled	launch-wizard-1	gx-gg	2023/06/24 14:11 GMT-4

GreenGenius Side:
OpenSwan ID: i-0a4ca26533118a6c8 Public IPv4:34.205.155.211
Windows Server(Internal) ID: i-05e570b085f496217 Public IPv4:44.204.112.72

OpenSwan configuration:





Instance: i-0a4ca26533118a6c8 (OpenSwan)

IAM Role –	Subnet ID subnet-0fddbb08f2c895f5a	Auto Scaling Group name –
IMDSv2 Required		

▼ Instance details Info

Platform Amazon Linux (Inferred)	AMI ID ami-022e1a32d3f742bd8	Monitoring disabled
Platform details Linux/UNIX	AMI name al2023-ami-2023.0.20230614.0-kernel-6.1-x86_64	Termination protection Disabled
Stop protection Disabled	Launch time Sat Jun 24 2023 14:11:05 GMT-0400 (Eastern Daylight Time) (17 minutes)	AMI location amazon/al2023-ami-2023.0.20230614.0-kernel-6.1-x86_64

Instance: i-0a4ca26533118a6c8 (OpenSwan)

Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled
AMI Launch index 1	Key pair assigned at launch gx-gg	State transition reason –
Credit specification standard	Kernel ID –	State transition message –
Usage operation RunInstances	RAM disk ID –	Owner 724796499914
ClassicLink –	Enclaves Support –	Boot mode uefi-preferred
Current instance boot mode legacy-bios	Allow tags in instance metadata Disabled	Use RBN as guest OS hostname Disabled

Instance: i-0a4ca26533118a6c8 (OpenSwan)

Answer RBN DNS hostname IPv4 Disabled		

▼ Host and placement group Info

Host ID –	Affinity –	Placement group –
Host resource group name –	Tenancy default	Placement group ID –
Virtualization type hvm	Reservation r-032fb23f3d89bb6ba	Partition number –
Number of vCPUs 1		

▼ Capacity reservation Info

Creating an instance for GXtechnic (our managing account, which is used as an example of Globex's established network):

First, we'll create and configure our custom VPC on AWS. We'll use an us-east availability, consisting of 1 public and 1 private subnet.

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info

Create only the VPC resource or the VPC and other networking resources.

- (●) VPC only
- ( ) VPC and more

Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

GXVPC

IPv4 CIDR block Info

- (●) IPv4 CIDR manual input
- ( ) IPAM-allocated IPv4 CIDR block

IPv4 CIDR

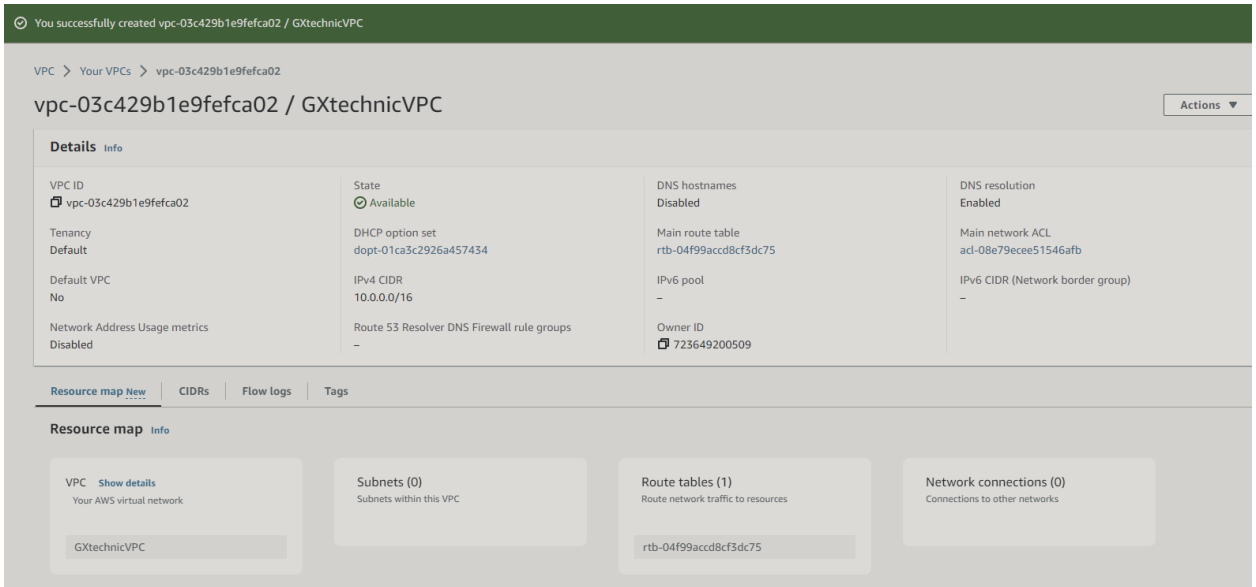
10.0.0.0/16

IPv6 CIDR block Info

- (●) No IPv6 CIDR block
- ( ) IPAM-allocated IPv6 CIDR block
- ( ) Amazon-provided IPv6 CIDR block
- ( ) IPv6 CIDR owned by me

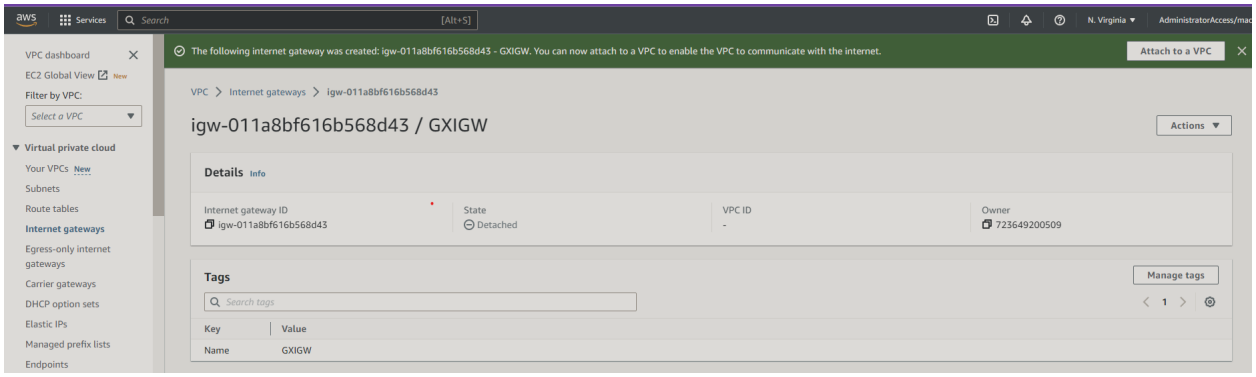
Tenancy Info

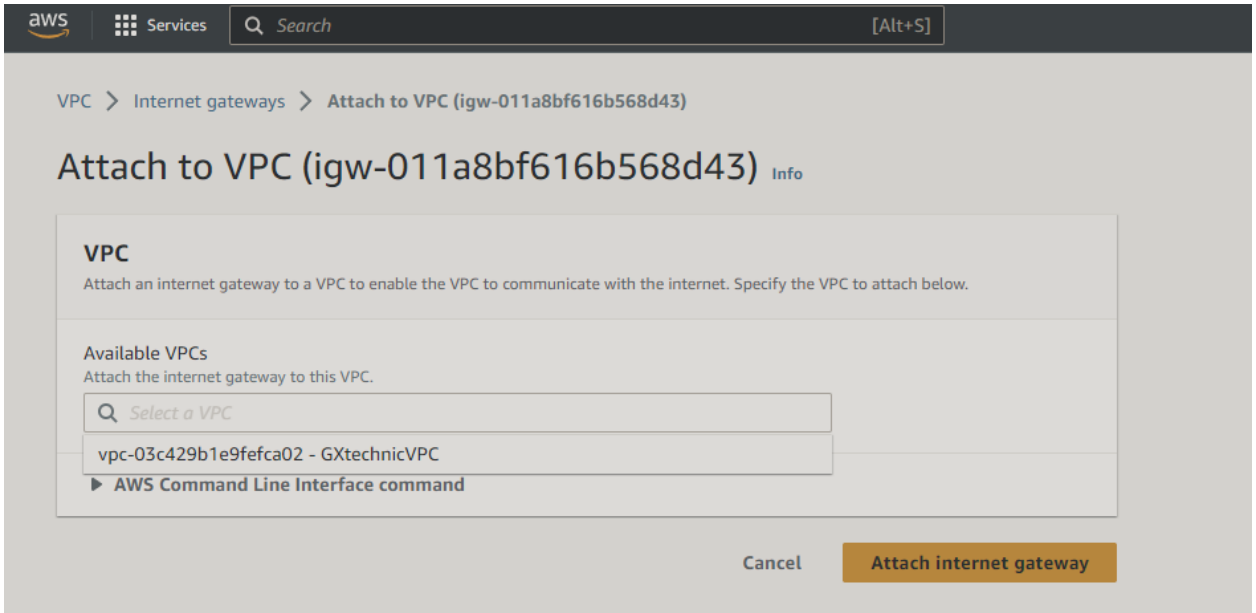
Default



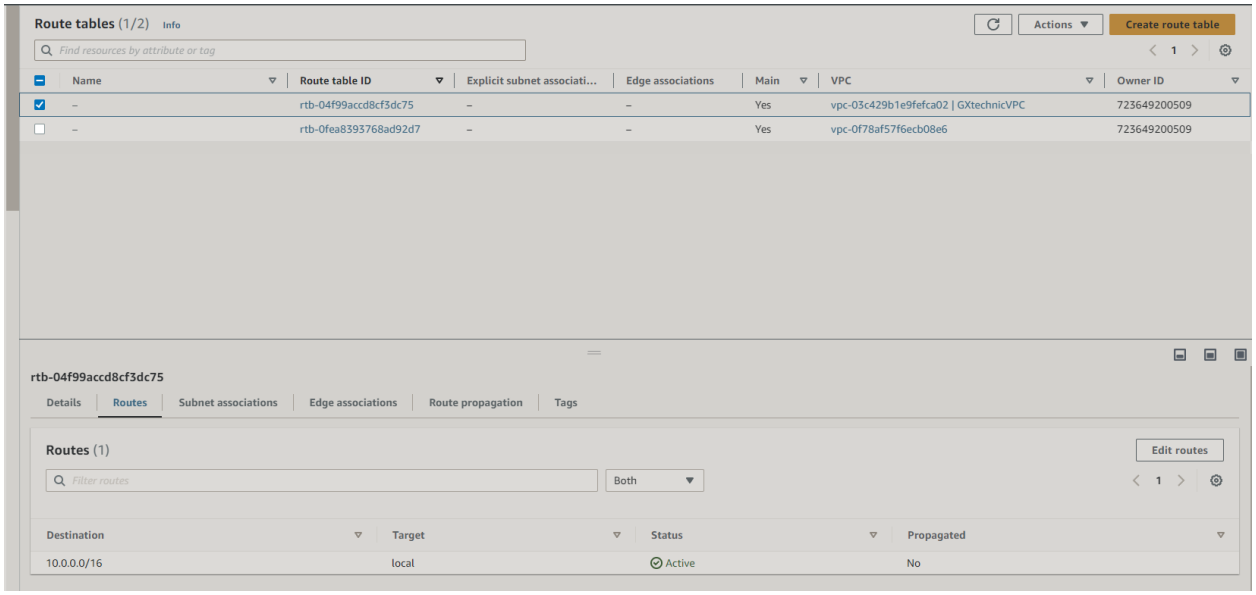
We enabled public subnets.

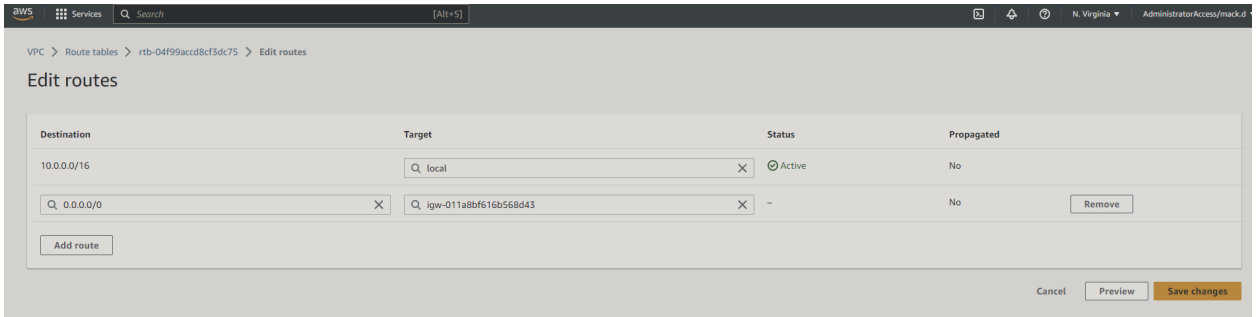
Now we're creating Internet Gateway with NAT Gateway:





Configuring Route Tables:





aws | Services | Search [Alt+S]

VPC > Route tables > Create route table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*

Create a tag with a key of 'Name' and a value that you specify.

Private-RT

VPC

The VPC to use for this route table.

vpc-03c429b1e9fefca02 (GXtechnicVPC)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	Private-RT	Remove

Add new tag

You can add 49 more tags.

Cancel | Create route table

aws | Services | Search [Alt+S] | N. Virginia | AdministratorAccess/mack.d

VPC > Route tables > rtb-014e0495bfd0f9621 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/4)

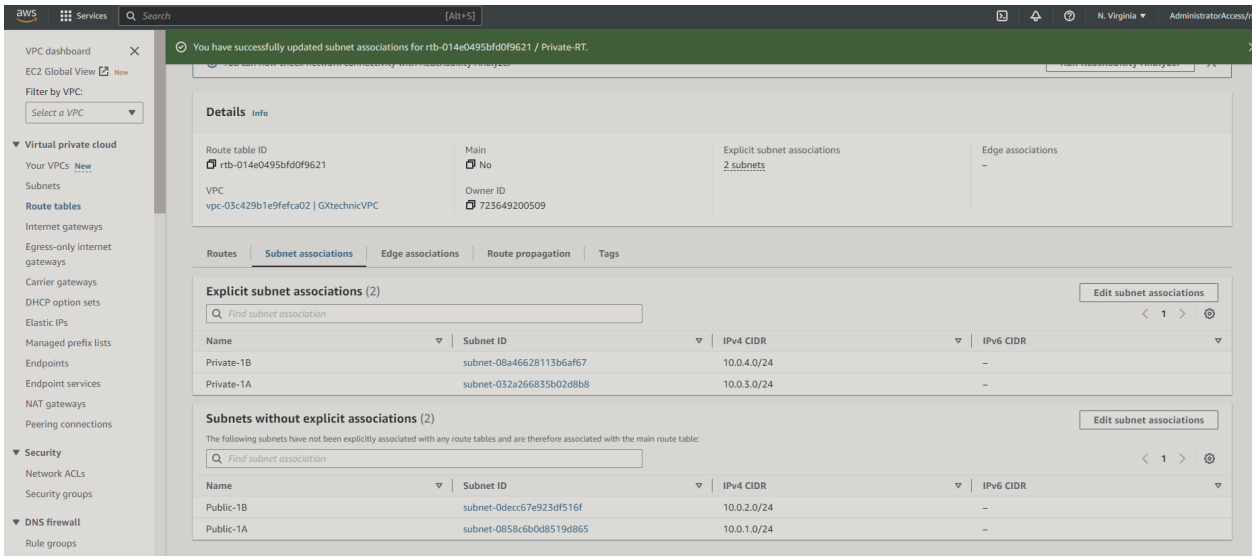
Filter subnet associations

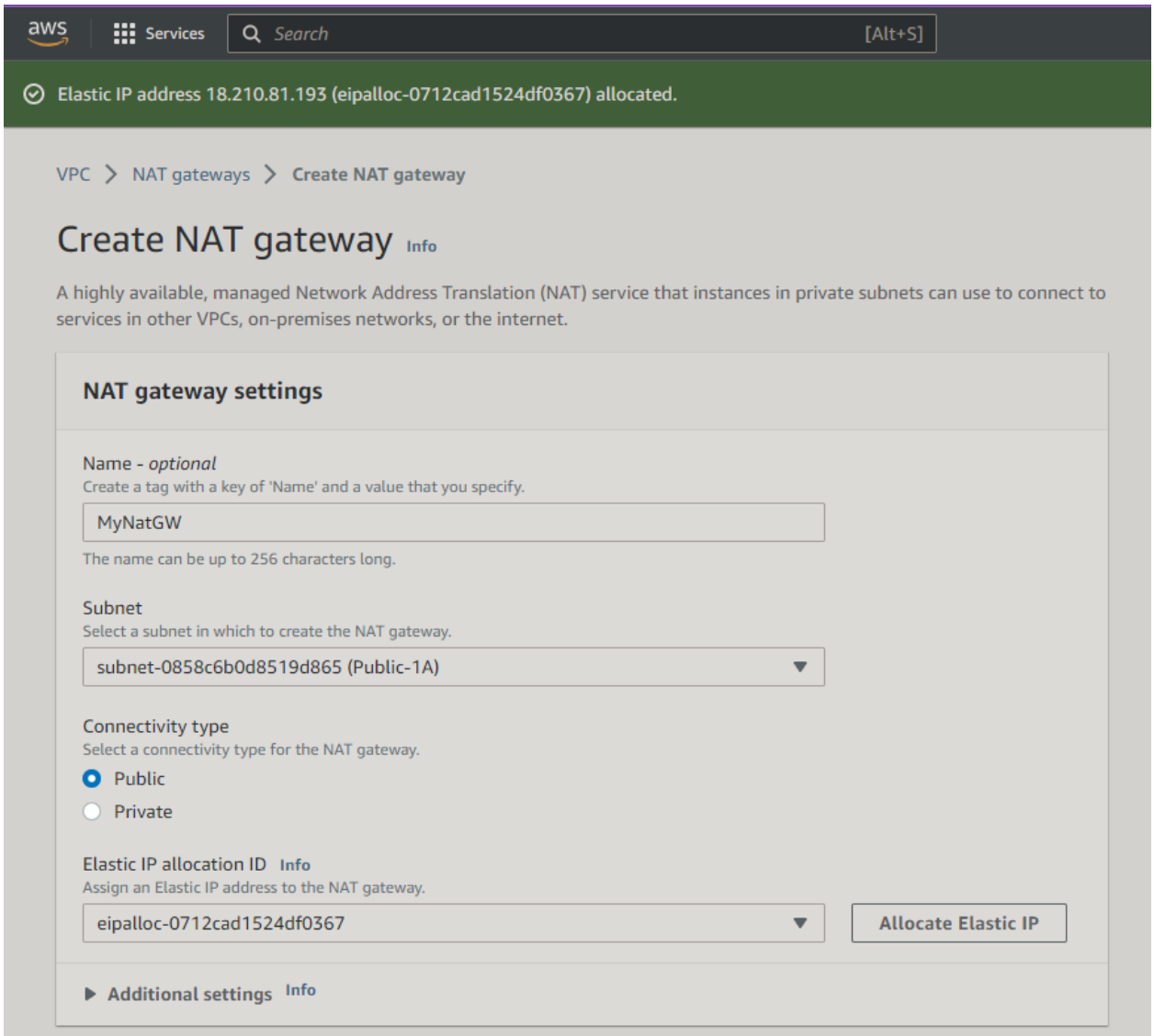
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Public-1B	subnet-0decc67e923df516f	10.0.2.0/24	–	Main (rtb-04f99accd8cf3dc75)
Private-1B	subnet-08a46628113b6af67	10.0.4.0/24	–	Main (rtb-04f99accd8cf3dc75)
Private-1A	subnet-032a266835b02d8b8	10.0.3.0/24	–	Main (rtb-04f99accd8cf3dc75)
Public-1A	subnet-0858c6b0d8519d865	10.0.1.0/24	–	Main (rtb-04f99accd8cf3dc75)

Selected subnets

subnet-032a266835b02d8b8 / Private-1A | subnet-08a46628113b6af67 / Private-1B

Cancel | Save associations


aws
Services
Search
[Alt+S]
N. Virginia
AdministratorAccess/
VPC dashboard
EC2 Global View New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs New
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
Security groups
DNS firewall
Rule groups
You have successfully updated subnet associations for rtb-014e0495bfd0f9621 / Private-RT.
Details Info
Route table ID
rtb-014e0495bfd0f9621
Main
No
Explicit subnet associations
2 subnets
Edge associations
–
VPC
vpc-03c429b1e9fefca02 | GXtechnicVPC
Owner ID
723649200509
Routes
Subnet associations
Edge associations
Route propagation
Tags
Explicit subnet associations (2)
Edit subnet associations
Find subnet association
Name
Subnet ID
IPv4 CIDR
IPv6 CIDR
Private-1B
subnet-08a46628113b6af67
10.0.4.0/24
Private-1A
subnet-032a266835b02d8b8
10.0.3.0/24
Subnets without explicit associations (2)
The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:
Public-1B
subnet-0decc67e923df516f
10.0.2.0/24
Public-1A
subnet-0858c6b0d8519d865
10.0.1.0/24


aws
Services
Search
[Alt+S]
Elastic IP address 18.210.81.193 (eipalloc-0712cad1524df0367) allocated.
VPC > NAT gateways > Create NAT gateway
Create NAT gateway Info
A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.
NAT gateway settings
Name - optional
Create a tag with a key of 'Name' and a value that you specify.
MyNatGW
The name can be up to 256 characters long.
Subnet
Select a subnet in which to create the NAT gateway.
subnet-0858c6b0d8519d865 (Public-1A)
Connectivity type
Select a connectivity type for the NAT gateway.
Public
Private
Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-0712cad1524df0367
Allocate Elastic IP
Additional settings Info

NAT gateway nat-04b962b417e87980b | MyNatGW was created successfully.

VPC > NAT gateways > nat-04b962b417e87980b

nat-04b962b417e87980b / MyNatGW

Actions

Details Info

NAT gateway ID nat-04b962b417e87980b	Connectivity type Public	State Pending	State message Info –
NAT gateway ARN arn:aws:ec2:us-east-1:723649200509:natgateway/nat-04b962b417e87980b	Primary public IPv4 address –	Primary private IPv4 address –	Primary network interface ID –
VPC vpc-03c429b1e9fefca02 / GXtechnicVPC	Subnet subnet-0858c6b0d8519d865 / Public-1A	Created Sunday, June 25, 2023 at 24:46:50 EDT	Deleted –

Secondary IPv4 addresses | Monitoring | Tags

Secondary IPv4 addresses

Edit secondary IPv4 address associations

Filter by secondary IPv4 address

1

Private IPv4 address	Network interface ID	Status	Failure message

Secondary IPv4 addresses are not available for this nat gateway.

VPC > Route tables > rtb-014e0495bfd0f9621

rtb-014e0495bfd0f9621 / Private-RT

Actions

You can now check network connectivity with Reachability Analyzer — Run Reachability Analyzer

Details Info

Route table ID rtb-014e0495bfd0f9621	Main No	Explicit subnet associations 2 subnets	Edge associations –
VPC vpc-03c429b1e9fefca02 | GXtechnicVPC	Owner ID 723649200509		

Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (1)

Edit routes

Filter routes — Both

1

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

aws — Services — Search [Alt+S] — N. Virginia — AdministratorAccess/mack.d

VPC > Route tables > rtb-014e0495bfd0f9621 > Edit routes

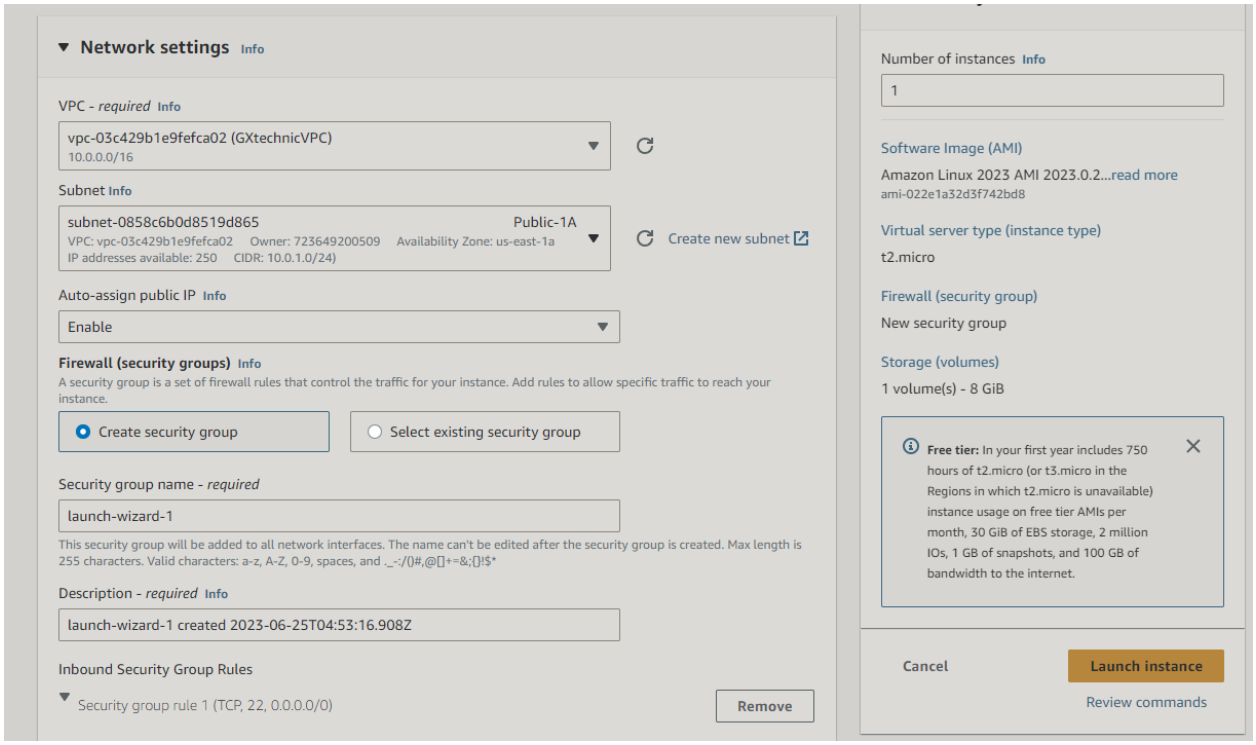
Edit routes

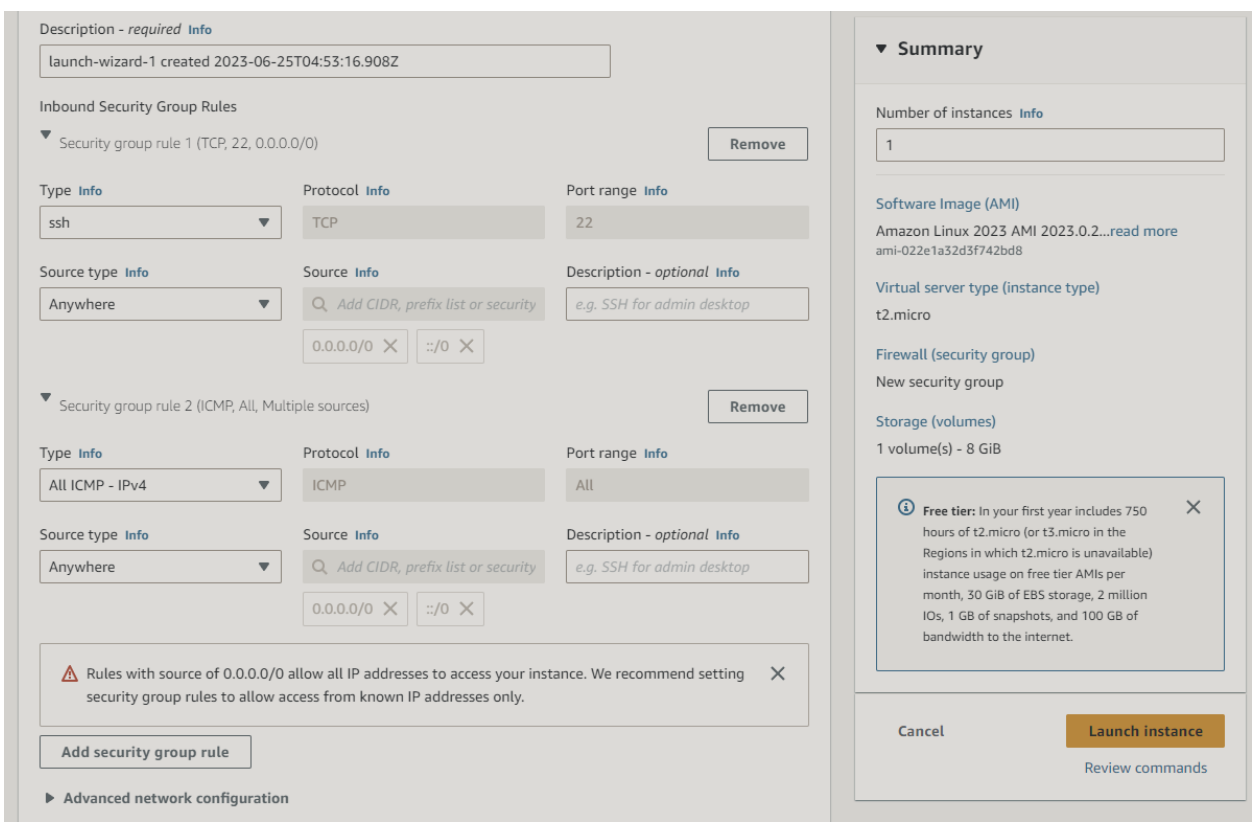
Destination	Target	Status	Propagated	
10.0.0.0/16	local	Active	No	
0.0.0.0/0	nat-04b962b417e87980b	–	No	Remove

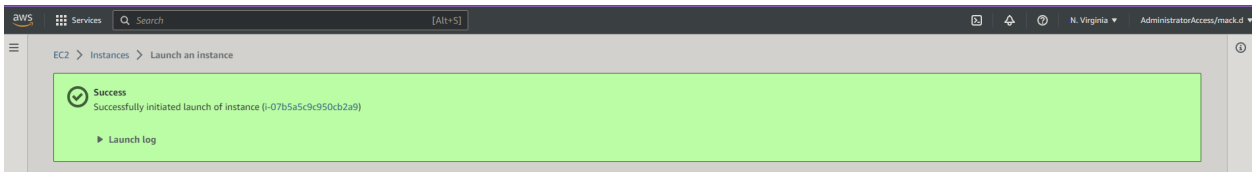
Add route

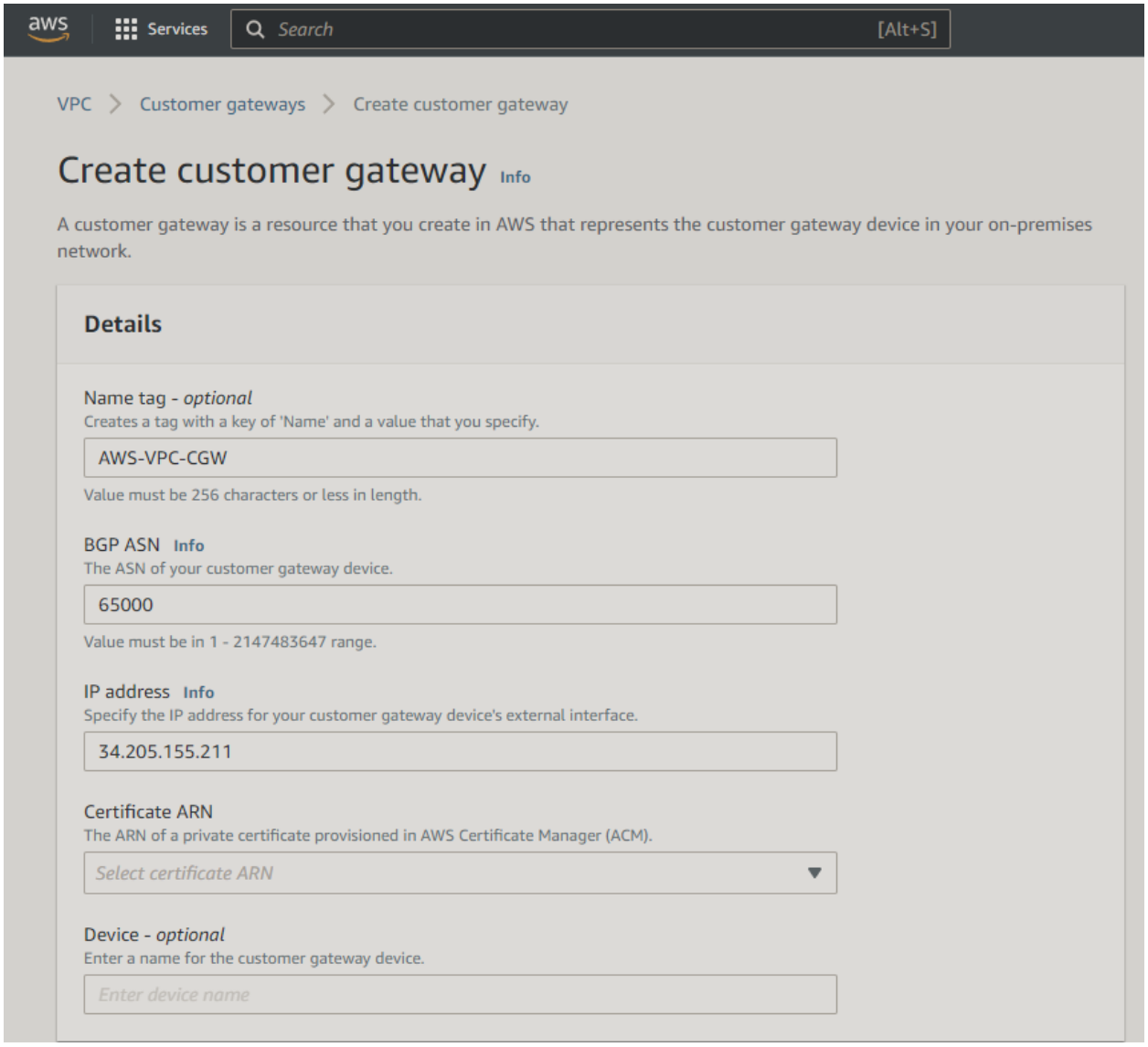
Cancel | Preview | Save changes

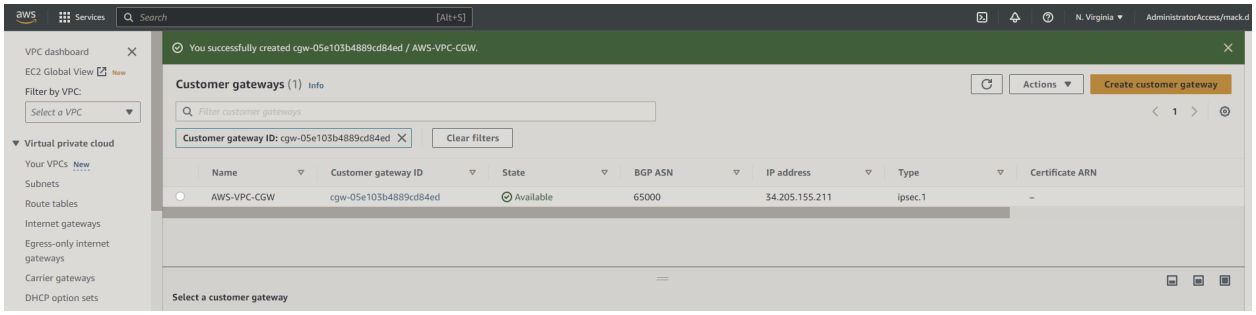
Now we can launch and configure our EC2 instance for GXtechnic:

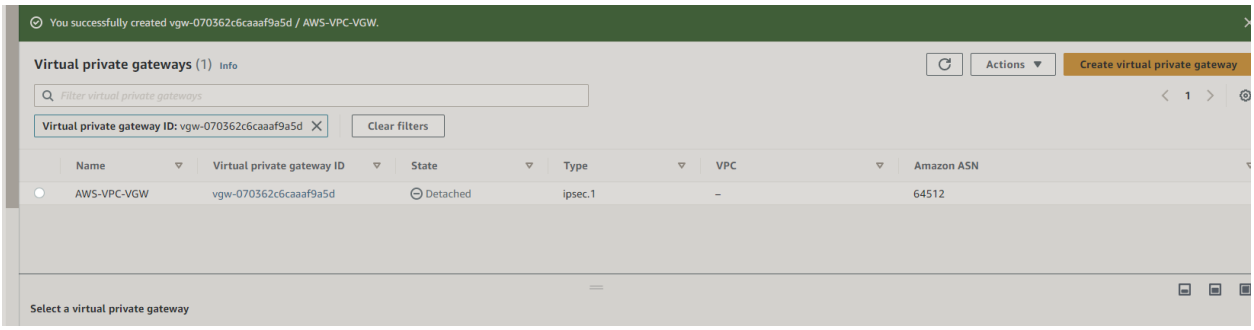




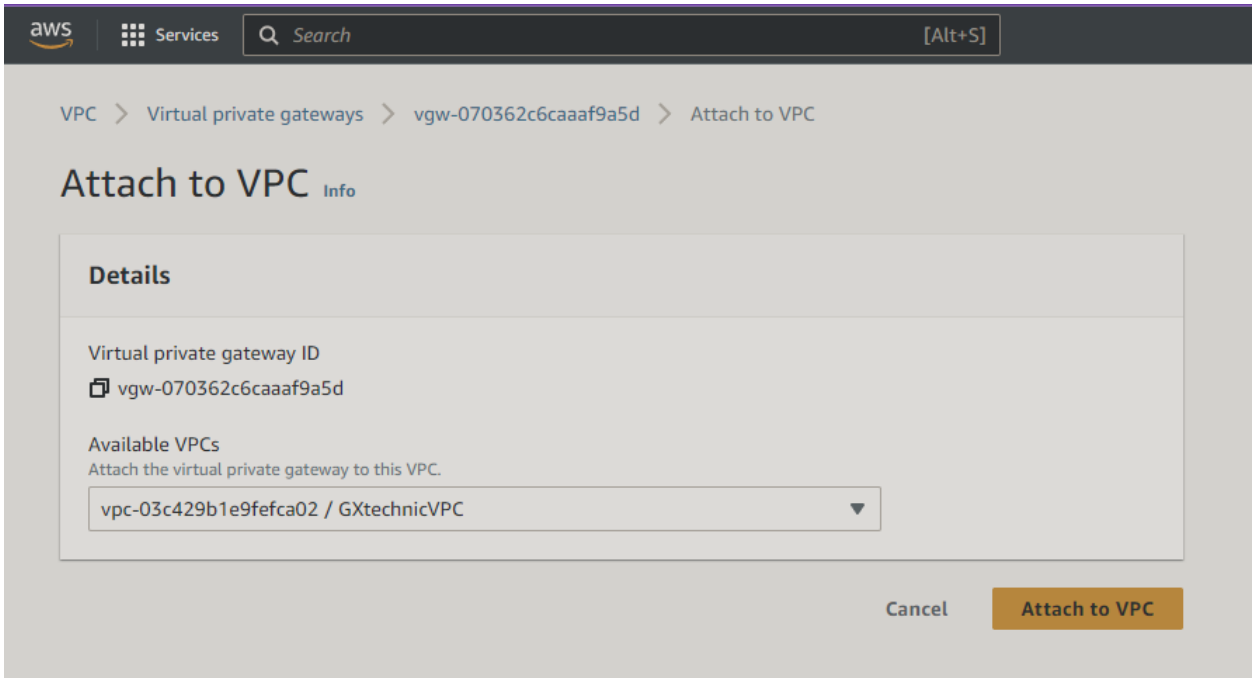

aws
Services
Search
[Alt+S]
N. Virginia
AdministratorAccess/mack.d
EC2 > Instances > Launch an instance
Success
Successfully initiated launch of instance (i-07b5a5c9c950cb2a9)
Launch log


aws
Services
Search
[Alt+S]
VPC > Customer gateways > Create customer gateway
Create customer gateway Info
A customer gateway is a resource that you create in AWS that represents the customer gateway device in your on-premises network.
Details
Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.
AWS-VPC-CGW
Value must be 256 characters or less in length.
BGP ASN Info
The ASN of your customer gateway device.
65000
Value must be in 1 - 2147483647 range.
IP address Info
Specify the IP address for your customer gateway device's external interface.
34.205.155.211
Certificate ARN
The ARN of a private certificate provisioned in AWS Certificate Manager (ACM).
Select certificate ARN
Device - optional
Enter a name for the customer gateway device.
Enter device name


aws
Services
Search
[Alt+S]
N. Virginia
AdministratorAccess/mack.d
VPC dashboard
EC2 Global View New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs New
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
You successfully created cgw-05e103b4889cd84ed / AWS-VPC-CGW.
Customer gateways (1) Info
Actions
Create customer gateway
Filter customer gateways
Customer gateway ID: cgw-05e103b4889cd84ed
Clear filters
Name
Customer gateway ID
State
BGP ASN
IP address
Type
Certificate ARN
AWS-VPC-CGW
cgw-05e103b4889cd84ed
Available
65000
34.205.155.211
ipsec.1
–
Select a customer gateway



Attaching it to our VPC:



aws | Services | Search [Alt+S]

VPC > VPN connections > Create VPN connection

Create VPN connection Info

Select the resources and additional configuration options that you want to use for the site-to-site VPN connection.

Details

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.

ON-PREM-AWS-VPN

Value must be 256 characters or less in length.

Target gateway type Info

- (●) Virtual private gateway
- ( ) Transit gateway
- ( ) Not associated

Virtual private gateway

vgw-070362c6caaaf9a5d / AWS-VPC-VGW

Customer gateway Info

- (●) Existing
- ( ) New

Customer gateway ID

cgw-05e103b4889cd84ed / AWS-VPC-CGW

Routing options Info

- ( ) Dynamic (requires BGP)
- (●) Static

You successfully created vpn-0f1b1b06d27ef3b0f / ON-PREM-AWS-VPN.

VPN connections (1) Info

Actions | Download configuration | Create VPN connection

Filter VPN connections

VPN ID: vpn-0f1b1b06d27ef3b0f | Clear filters

Name	VPN ID	State	Virtual private gateway	Transit gateway	Customer gateway	Customer gateway ad...	Inside IP ve
ON-PREM-AWS-VPN	vpn-0f1b1b06d27ef3b0f	Pending	vgw-070362c6caaaf9a5d	–	cgw-05e103b4889cd84ed	34.205.155.211	IPv4

aws Services Search [Alt+S]

VPC > Route tables > rtb-04f99accd8cf3dc75 > Edit route propagation

Edit route propagation

Route table basic details

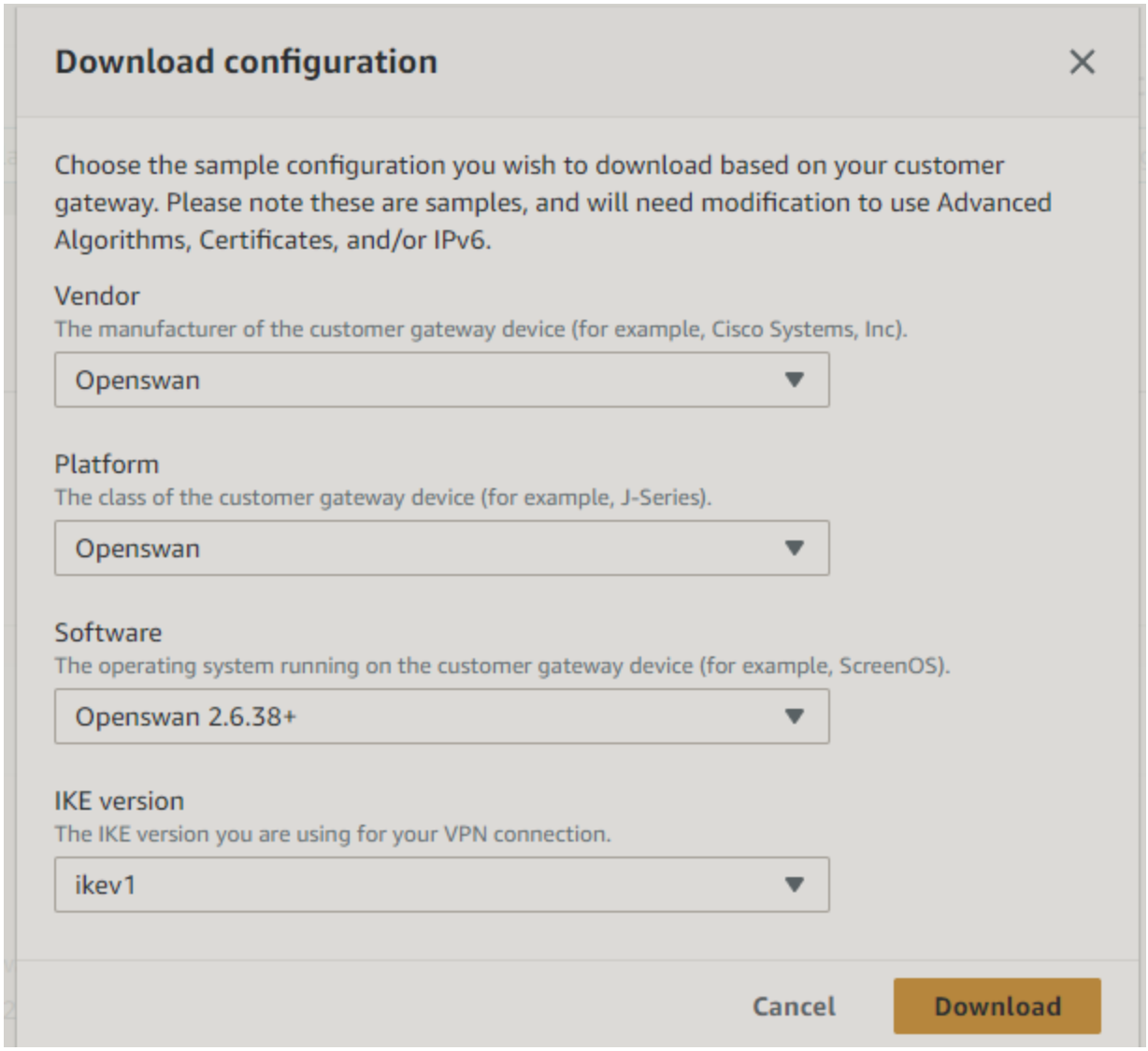
Route table ID

rtb-04f99accd8cf3dc75

Edit route propagation

Virtual Private Gateway	Propagation
vgw-070362c6caaaf9a5d / AWS-VPC-VGW	☑ Enable

Cancel Save



Autogenerated Config File:

IPSEC Tunnel #1

This configuration assumes that you already have a default openswan installation in place on the Amazon Linux operating system (but may also work with other distros as well)

1) Open /etc/sysctl.conf and ensure that its values match the following:
   net.ipv4.ip_forward = 1
   net.ipv4.conf.default.rp_filter = 0
   net.ipv4.conf.default.accept_source_route = 0

2) Apply the changes in step 1 by executing the command 'sysctl -p'

3) Open /etc/ipsec.conf and look for the line below. Ensure that the # in front of the line has been removed, then save and exit the file.

```
#include /etc/ipsec.d/*.conf
```

4) Create a new file at /etc/ipsec.d/aws.conf if doesn't already exist, and then open it. Append the following configuration to the end in the file:

#leftsubnet= is the local network behind your openswan server, and you will need to replace the <LOCAL NETWORK> below with this value (don't include the brackets). If you have multiple subnets, you can use 0.0.0.0/0 instead.

#rightsubnet= is the remote network on the other side of your VPN tunnel that you wish to have connectivity with, and you will need to replace <REMOTE NETWORK> with this value (don't include brackets).

```
conn Tunnel1
        authby=secret
        auto=start
        left=%defaultroute
        leftid=34.205.155.211
        right=3.86.75.218
        type=tunnel
        ikelifetime=8h
        keylife=1h
        phase2alg=aes128-sha1;modp1024
        ike=aes128-sha1;modp1024
        auth=esp
        keyingtries=%forever
        keyexchange=ike
        leftsubnet=<LOCAL NETWORK>
        rightsubnet=<REMOTE NETWORK>
        dpddelay=10
        dpdtimeout=30
        dpdaction=restart_by_peer
```

5) Create a new file at /etc/ipsec.d/aws.secrets if it doesn't already exist, and append this line to the file (be mindful of the spacing!):

```
34.205.155.211 3.86.75.218: PSK "rc10LuZZ2qCVt5VDTcPMB98ucprQ2IIu"
```

Establishing an SSH connection with our OpenSwan server (the one that is running an IPSec protocol on the customer's side) to apply all the necessary configurations:

```
ec2-user@ip-172-31-9-119:~

Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\macka>cd downloads

C:\Users\macka\Downloads>ssh -i "gx-gg.pem" ec2-user@ec2-34-205-155-211.compute-1.amazonaws.com
The authenticity of host 'ec2-34-205-155-211.compute-1.amazonaws.com (34.205.155.211)' can't be established.
ED25519 key fingerprint is SHA256:jIXju6Jy98PXi1pnun6nOIUe6icjUFusY7sHBU9eGVU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-205-155-211.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-9-119 ~]$
```

```
root@ip-172-31-9-119:/home/

Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\macka>cd downloads

C:\Users\macka\Downloads>ssh -i "gx-gg.pem" ec2-user@ec2-34-205-155-211.compute-1.amazonaws.com
The authenticity of host 'ec2-34-205-155-211.compute-1.amazonaws.com (34.205.155.211)' can't be established.
ED25519 key fingerprint is SHA256:jIXju6Jy98PXi1pnun6nOIUe6icjUFusY7sHBU9eGVU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-205-155-211.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-9-119 ~]$ sudo su
[root@ip-172-31-9-119 ec2-user]# nano /etc/sysctl.conf
```

```
root@ip-172-31-9-119:/home/
  GNU nano 5.8                                    /etc/sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
   net.ipv4.ip_forward = 1
   net.ipv4.conf.default.rp_filter = 0
   net.ipv4.conf.default.accept_source_route = 0

                                        [ Wrote 13 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```

Applying the changes from Step 1:

```
root@ip-172-31-9-119:/home/
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\macka>cd downloads

C:\Users\macka\Downloads>ssh -i "gx-gg.pem" ec2-user@ec2-34-205-155-211.compute-1.amazonaws.com
The authenticity of host 'ec2-34-205-155-211.compute-1.amazonaws.com (34.205.155.211)' can't be established.
ED25519 key fingerprint is SHA256:jIXju6Jy98PXi1pnun6nOIUe6icjUFusY7sHBU9eGVU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-205-155-211.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-9-119 ~]$ sudo su
[root@ip-172-31-9-119 ec2-user]# nano /etc/sysctl.conf
[root@ip-172-31-9-119 ec2-user]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.default.accept_source_route = 0
[root@ip-172-31-9-119 ec2-user]#
```

Installing OpenSwan (libreswan, which is a more modern version) before moving to Step 4:

```
[root@ip-172-31-9-119 ec2-user]# yum install telnet
Last metadata expiration check: 20:33:10 ago on Sat Jun 24 18:11:51 2023.
Package telnet-1:0.17-83.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-9-119 ec2-user]# nano /etc/ipsec.d/aws.conf
[root@ip-172-31-9-119 ec2-user]# nano /etc/ipsec.d/aws.conf
[root@ip-172-31-9-119 ec2-user]# mkdir /etc/ipsec.d/
[root@ip-172-31-9-119 ec2-user]# cd /etc/ipsec.d/
[root@ip-172-31-9-119 ipsec.d]# nano aws.conf
[root@ip-172-31-9-119 ipsec.d]# nano /etc/ipsec.d/aws.conf
[root@ip-172-31-9-119 ipsec.d]#
```

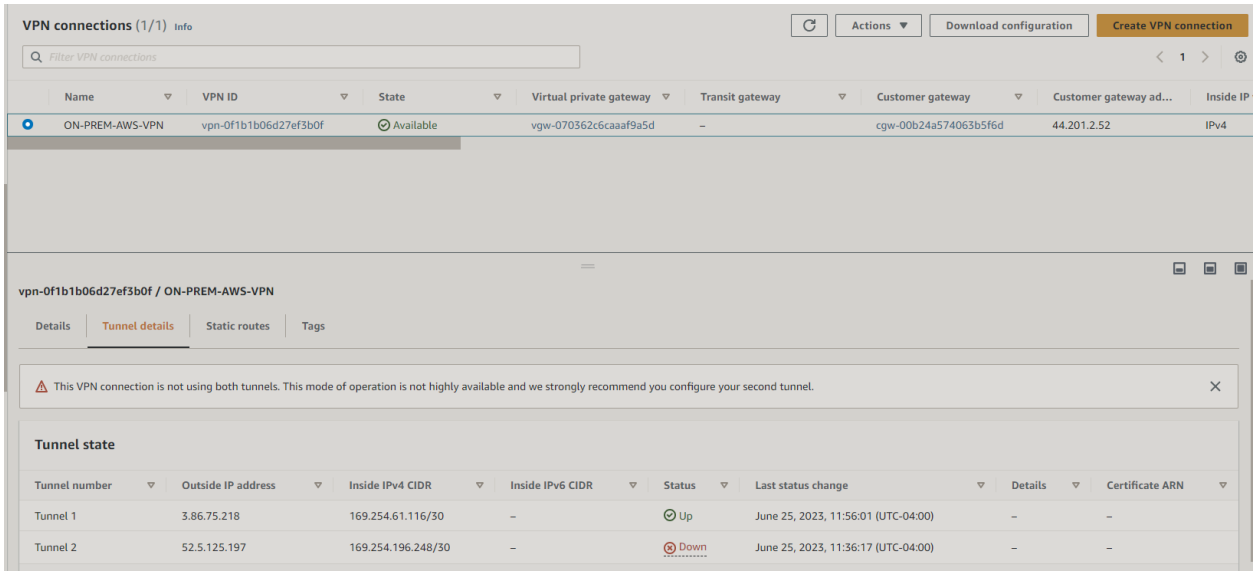
As it appeared later, the “openswan” was unavailable at the Linux AMI 2023 instance. Thus, we had to repeat the entire process for a newly created OpenSwan2 instance, which runs on AWS Linux2 AMI kernel 5.10. After that, we successfully ran the “yum install openswan -y” command.

```
root@ip-172-31-10-122:/home    Windows PowerShell

================================================================================================================
Install  1 Package (+2 Dependent packages)

Total download size: 2.3 M
Installed size: 7.2 M
Downloading packages:
(1/3): ldns-1.6.16-10.amzn2.0.3.x86_64.rpm                                          | 473 kB  00:00:00
(2/3): unbound-libs-1.7.3-15.amzn2.0.4.x86_64.rpm                                   | 485 kB  00:00:00
(3/3): libreswan-3.25-4.8.amzn2.0.1.x86_64.rpm                                      | 1.4 MB  00:00:00
----------------------------------------------------------------------------------------------------------------
Total                                                                        13 MB/s | 2.3 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : unbound-libs-1.7.3-15.amzn2.0.4.x86_64                                                      1/3
  Installing : ldns-1.6.16-10.amzn2.0.3.x86_64                                                             2/3
  Installing : libreswan-3.25-4.8.amzn2.0.1.x86_64                                                         3/3
  Verifying  : libreswan-3.25-4.8.amzn2.0.1.x86_64                                                         1/3
  Verifying  : ldns-1.6.16-10.amzn2.0.3.x86_64                                                             2/3
  Verifying  : unbound-libs-1.7.3-15.amzn2.0.4.x86_64                                                      3/3

Installed:
  libreswan.x86_64 0:3.25-4.8.amzn2.0.1

Dependency Installed:
  ldns.x86_64 0:1.6.16-10.amzn2.0.3                    unbound-libs.x86_64 0:1.7.3-15.amzn2.0.4

Complete!
[root@ip-172-31-10-122 ec2-user]#
```

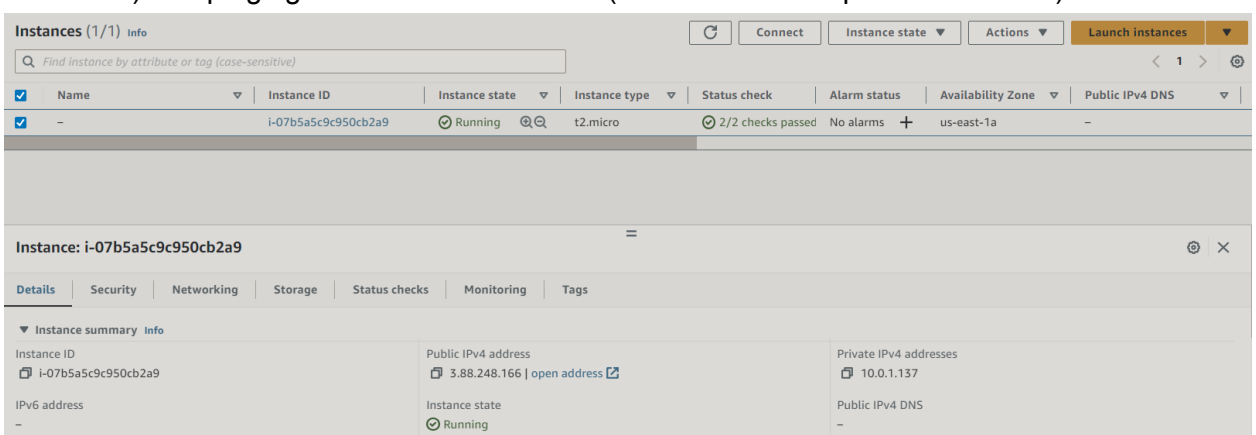
```
-----------------------------------------------------------------------------------------------------------------
IPSEC Tunnel #1
-----------------------------------------------------------------------------------------------------------------

This configuration assumes that you already have a default openswan installation in place on the Amazon Linux operating system (but may also work with other distros as well)

1) Open /etc/sysctl.conf and ensure that its values match the following:
   net.ipv4.ip_forward = 1
   net.ipv4.conf.default.rp_filter = 0
   net.ipv4.conf.default.accept_source_route = 0

2) Apply the changes in step 1 by executing the command 'sysctl -p'

3) Open /etc/ipsec.conf and look for the line below. Ensure that the # in front of the line has been removed, then save and exit the file.
    #include /etc/ipsec.d/*.conf

4) Create a new file at /etc/ipsec.d/aws.conf if doesn't already exist, and then open it. Append the following configuration to the end in the file:
 #leftsubnet= is the local network behind your openswan server, and you will need to replace the <LOCAL NETWORK> below with this value (don't include the brackets). If you
have multiple subnets, you can use 0.0.0.0/0 instead.
 #rightsubnet= is the remote network on the other side of your VPN tunnel that you wish to have connectivity with, and you will need to replace <REMOTE NETWORK> with this
value (don't include brackets).

conn Tunnel1
        authby=secret
        auto=start
        left=%defaultroute
        leftid=44.201.2.52
        right=3.86.75.218
        type=tunnel
        ikelifetime=8h
        keylife=1h
        phase2alg=aes128-sha1;modp1024
        ike=aes128-sha1;modp1024
        keyingtries=%forever
        keyexchange=ike
        leftsubnet=172.31.0.0/16
        rightsubnet=10.0.0.0/16
        dpddelay=10
        dpdtimeout=30
        dpdaction=restart_by_peer


5) Create a new file at /etc/ipsec.d/aws.secrets if it doesn't already exist, and append this line to the file (be mindful of the spacing!):
44.201.2.52 3.86.75.218: PSK "rc10LuZZ2qCVt5VDTcPMB98ucprQ2IIu"
```

```
root@ip-172-31-10-122:/home    Windows PowerShell

  Verifying  : ldns-1.6.16-10.amzn2.0.3.x86_64                                          2/3
  Verifying  : unbound-libs-1.7.3-15.amzn2.0.4.x86_64                                   3/3

Installed:
  libreswan.x86_64 0:3.25-4.8.amzn2.0.1

Dependency Installed:
  ldns.x86_64 0:1.6.16-10.amzn2.0.3                     unbound-libs.x86_64 0:1.7.3-15.amzn2.0.4

Complete!
[root@ip-172-31-10-122 ec2-user]# nano /etc/ipsec.d/aws.conf
[root@ip-172-31-10-122 ec2-user]# nano /etc/ipsec.d/aws.secrets
[root@ip-172-31-10-122 ec2-user]# systemctl start ipsec
[root@ip-172-31-10-122 ec2-user]# systemctl status ipsec
● ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec
   Loaded: loaded (/usr/lib/systemd/system/ipsec.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2023-06-25 15:54:46 UTC; 24s ago
     Docs: man:ipsec(8)
           man:pluto(8)
           man:ipsec.conf(5)
  Process: 4084 ExecStartPre=/usr/sbin/ipsec --checknflog (code=exited, status=0/SUCCESS)
  Process: 4078 ExecStartPre=/usr/sbin/ipsec --checknss (code=exited, status=0/SUCCESS)
  Process: 3563 ExecStartPre=/usr/libexec/ipsec/_stackmanager start (code=exited, status=0/SUCCESS)
  Process: 3561 ExecStartPre=/usr/libexec/ipsec/addconn --config /etc/ipsec.conf --checkconfig (code=exited, status=0/SUCCESS)
 Main PID: 4102 (pluto)
   Status: "Startup completed."
   CGroup: /system.slice/ipsec.service
           └─4102 /usr/libexec/ipsec/pluto --leak-detective --config /etc/ipsec.conf --nofork

Jun 25 15:54:46 ip-172-31-10-122.ec2.internal pluto[4102]: | setup callback for interface eth0:500 fd 15
Jun 25 15:54:46 ip-172-31-10-122.ec2.internal pluto[4102]: loading secrets from "/etc/ipsec.secrets"
Jun 25 15:54:46 ip-172-31-10-122.ec2.internal pluto[4102]: loading secrets from "/etc/ipsec.d/aws.secrets"
Jun 25 15:54:46 ip-172-31-10-122.ec2.internal pluto[4102]: "Tunnel1" #1: initiating Main Mode
Jun 25 15:54:46 ip-172-31-10-122.ec2.internal pluto[4102]: "Tunnel1" #1: STATE_MAIN_I2: sent MI2, expecting MR2
Jun 25 15:54:46 ip-172-31-10-122.ec2.internal pluto[4102]: "Tunnel1" #1: STATE_MAIN_I3: sent MI3, expecting MR3
Jun 25 15:54:46 ip-172-31-10-122.ec2.internal pluto[4102]: "Tunnel1" #1: Peer ID is ID_IPV4_ADDR: '3.86.75.218'
Jun 25 15:54:46 ip-172-31-10-122.ec2.internal pluto[4102]: "Tunnel1" #1: STATE_MAIN_I4: ISAKMP SA established {au...024}Jun 25 15:54:46 ip-
172-31-10-122.ec2.internal pluto[4102]: "Tunnel1" #2: initiating Quick Mode PSK+ENCRYPT+TUNNEL...024}Jun 25 15:54:46 ip-172-31-10-122.ec2.i
nternal pluto[4102]: "Tunnel1" #2: STATE_QUICK_I2: sent QI2, IPsec SA estab...ive}Hint: Some lines were ellipsized, use -l to show in full.

[root@ip-172-31-10-122 ec2-user]#
```

As we can see, our VPN Tunnel was properly configured, and it is Up now. For reliability purposes, an additional (redundant) Tunnel can be implemented as well:



Testing the connection:
Copying the Private IPv4 address of our instance on the managing side (GXtechnic - Globex simulation) and pinging it from the client's side (GreenGenius's OpenSwan server)



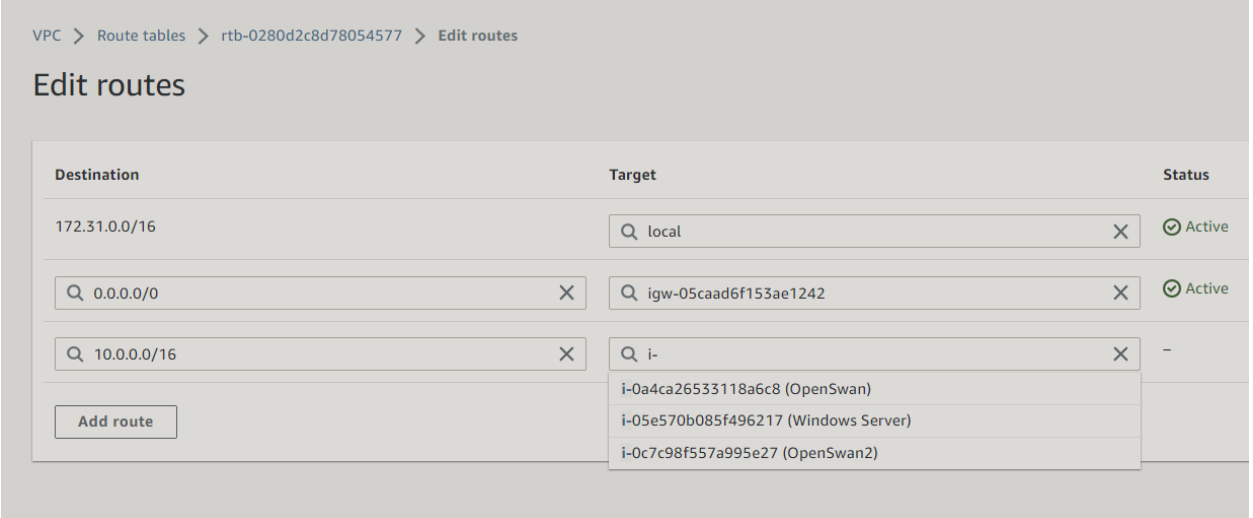
```
[root@ip-172-31-10-122 ec2-user]# ping 10.0.1.137
PING 10.0.1.137 (10.0.1.137) 56(84) bytes of data.
64 bytes from 10.0.1.137: icmp_seq=1 ttl=127 time=1.91 ms
64 bytes from 10.0.1.137: icmp_seq=2 ttl=127 time=2.05 ms
64 bytes from 10.0.1.137: icmp_seq=3 ttl=127 time=1.97 ms
64 bytes from 10.0.1.137: icmp_seq=4 ttl=127 time=1.91 ms
64 bytes from 10.0.1.137: icmp_seq=5 ttl=127 time=4.84 ms
64 bytes from 10.0.1.137: icmp_seq=6 ttl=127 time=2.07 ms
64 bytes from 10.0.1.137: icmp_seq=7 ttl=127 time=1.97 ms
64 bytes from 10.0.1.137: icmp_seq=8 ttl=127 time=2.00 ms
64 bytes from 10.0.1.137: icmp_seq=9 ttl=127 time=1.96 ms
64 bytes from 10.0.1.137: icmp_seq=10 ttl=127 time=1.98 ms
64 bytes from 10.0.1.137: icmp_seq=11 ttl=127 time=2.04 ms
64 bytes from 10.0.1.137: icmp_seq=12 ttl=127 time=2.00 ms
64 bytes from 10.0.1.137: icmp_seq=13 ttl=127 time=3.68 ms
64 bytes from 10.0.1.137: icmp_seq=14 ttl=127 time=2.10 ms
64 bytes from 10.0.1.137: icmp_seq=15 ttl=127 time=1.93 ms
64 bytes from 10.0.1.137: icmp_seq=16 ttl=127 time=1.92 ms
64 bytes from 10.0.1.137: icmp_seq=17 ttl=127 time=2.54 ms
64 bytes from 10.0.1.137: icmp_seq=18 ttl=127 time=1.98 ms
64 bytes from 10.0.1.137: icmp_seq=19 ttl=127 time=2.05 ms
64 bytes from 10.0.1.137: icmp_seq=20 ttl=127 time=2.03 ms
64 bytes from 10.0.1.137: icmp_seq=21 ttl=127 time=2.07 ms
64 bytes from 10.0.1.137: icmp_seq=22 ttl=127 time=1.88 ms
64 bytes from 10.0.1.137: icmp_seq=23 ttl=127 time=2.12 ms
64 bytes from 10.0.1.137: icmp_seq=24 ttl=127 time=1.99 ms
64 bytes from 10.0.1.137: icmp_seq=25 ttl=127 time=3.88 ms
64 bytes from 10.0.1.137: icmp_seq=26 ttl=127 time=1.90 ms
64 bytes from 10.0.1.137: icmp_seq=27 ttl=127 time=2.16 ms
64 bytes from 10.0.1.137: icmp_seq=28 ttl=127 time=2.04 ms
64 bytes from 10.0.1.137: icmp_seq=29 ttl=127 time=1.89 ms
64 bytes from 10.0.1.137: icmp_seq=30 ttl=127 time=1.95 ms
64 bytes from 10.0.1.137: icmp_seq=31 ttl=127 time=1.93 ms
64 bytes from 10.0.1.137: icmp_seq=32 ttl=127 time=2.18 ms
64 bytes from 10.0.1.137: icmp_seq=33 ttl=127 time=1.86 ms
```

As we can see, everything works properly, and we got a good response.

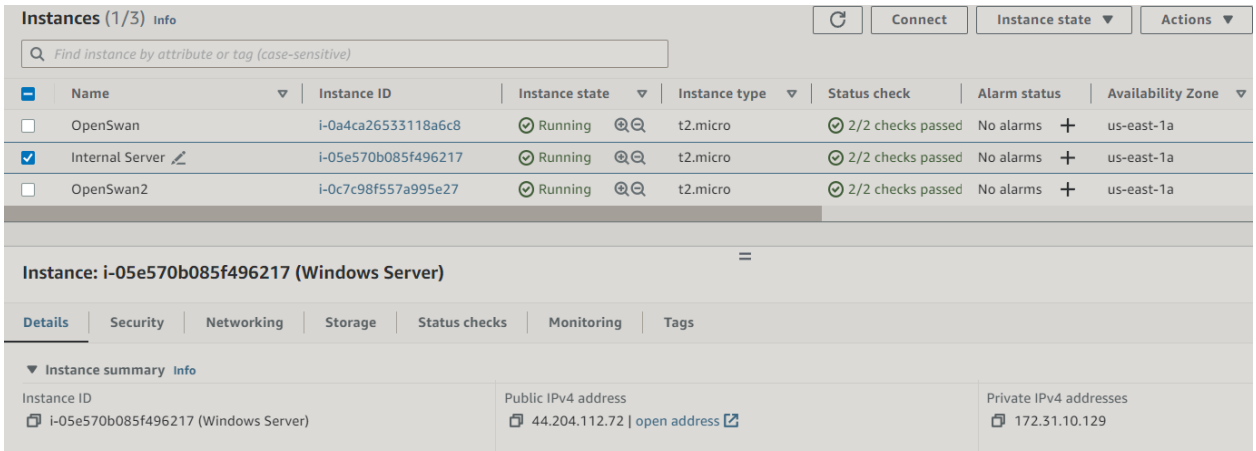
We established a good connection between GreenGenius's OpenSwan server and GXtechnic's ES2 instance/server using its private IPv4 address. Thus, we ensured it went through the configured VPN connection/tunnel.

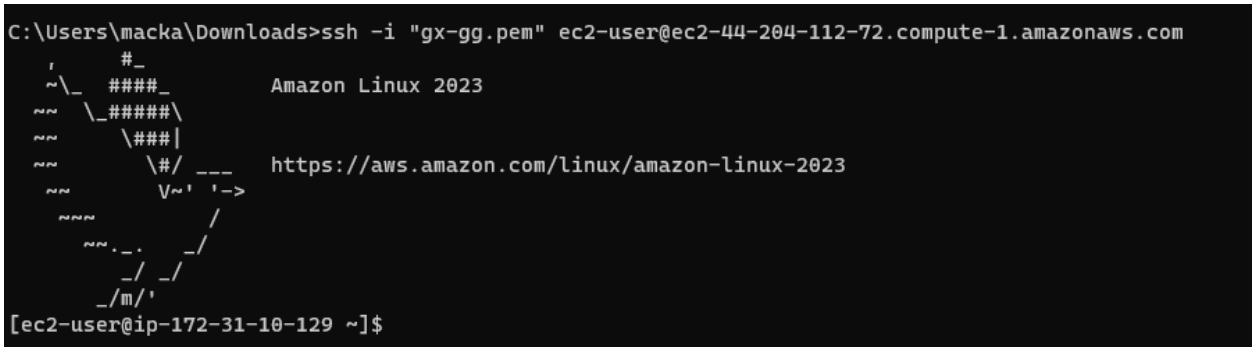
Now we need to ensure we can ping from GreenGenius's internal server to GXtechnic's internal server:

First, configuring the Routes on the client's account:
Adding a Route: 10.0.0.0/16
Choosing a Target: i-OpenSwan2



Establishing the SSH connection to GreenGenius's Internal Server using its IP address:





```
[ec2-user@ip-172-31-10-129 ~]$ ping 10.0.1.137
PING 10.0.1.137 (10.0.1.137) 56(84) bytes of data.
From 172.31.10.122 icmp_seq=1 Redirect Host(New nexthop: 172.31.0.1)
64 bytes from 10.0.1.137: icmp_seq=1 ttl=126 time=2.97 ms
From 172.31.10.122 icmp_seq=2 Redirect Host(New nexthop: 172.31.0.1)
64 bytes from 10.0.1.137: icmp_seq=2 ttl=126 time=2.51 ms
From 172.31.10.122 icmp_seq=3 Redirect Host(New nexthop: 172.31.0.1)
64 bytes from 10.0.1.137: icmp_seq=3 ttl=126 time=2.47 ms
From 172.31.10.122 icmp_seq=4 Redirect Host(New nexthop: 172.31.0.1)
64 bytes from 10.0.1.137: icmp_seq=4 ttl=126 time=2.48 ms
From 172.31.10.122 icmp_seq=5 Redirect Host(New nexthop: 172.31.0.1)
64 bytes from 10.0.1.137: icmp_seq=5 ttl=126 time=3.80 ms
From 172.31.10.122 icmp_seq=6 Redirect Host(New nexthop: 172.31.0.1)
64 bytes from 10.0.1.137: icmp_seq=6 ttl=126 time=2.38 ms
64 bytes from 10.0.1.137: icmp_seq=7 ttl=126 time=2.37 ms
From 172.31.10.122 icmp_seq=8 Redirect Host(New nexthop: 172.31.0.1)
64 bytes from 10.0.1.137: icmp_seq=8 ttl=126 time=2.42 ms
64 bytes from 10.0.1.137: icmp_seq=9 ttl=126 time=2.75 ms
64 bytes from 10.0.1.137: icmp_seq=10 ttl=126 time=4.08 ms
From 172.31.10.122 icmp_seq=11 Redirect Host(New nexthop: 172.31.0.1)
64 bytes from 10.0.1.137: icmp_seq=11 ttl=126 time=2.48 ms
64 bytes from 10.0.1.137: icmp_seq=12 ttl=126 time=2.38 ms
64 bytes from 10.0.1.137: icmp_seq=13 ttl=126 time=2.48 ms
64 bytes from 10.0.1.137: icmp_seq=14 ttl=126 time=2.98 ms
64 bytes from 10.0.1.137: icmp_seq=15 ttl=126 time=2.52 ms
64 bytes from 10.0.1.137: icmp_seq=16 ttl=126 time=2.34 ms
From 172.31.10.122 icmp_seq=17 Redirect Host(New nexthop: 172.31.0.1)
64 bytes from 10.0.1.137: icmp_seq=17 ttl=126 time=2.36 ms
64 bytes from 10.0.1.137: icmp_seq=18 ttl=126 time=2.48 ms
64 bytes from 10.0.1.137: icmp_seq=19 ttl=126 time=2.49 ms
```

As we can see, we get a response here as well.
We were able to establish an encrypted connection between GreenGenius's internal server and Globex's internal server.
Note: When the request is sent to the Globex server, it first reaches the OpenSwan, and only after it is redirected to the Globex via a secure tunnel.

Resources used to implement the AWS Site-to-Site VPN and the AWS AIM:

https://docs.aws.amazon.com/iam/index.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-console.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-service.html
https://signin.aws.amazon.com/switchrole
https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC_VPN.html
https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/aws-site-to-site-vpn.html
https://www.youtube.com/watch?v=7tTrN8WXMlg&t=237s
https://www.youtube.com/watch?v=T4U2YC4PJkY
https://www.youtube.com/watch?v=uOrq8ZUuaAQ
https://www.youtube.com/watch?v=7tTrN8WXMlg&t=187s
https://www.youtube.com/watch?v=BfE2G-fsBNU
https://www.youtube.com/watch?v=PjKvwxTTSUk&list=PLzde74P_a04cKnuXyi--fkIoY1sxztyqL
https://www.youtube.com/watch?v=FNaXYAXcSuU&list=PLzde74P_a04cKnuXyi--fkIoY1sxztyqL&index=6
https://dev.to/michael_timbs/switching-between-multiple-aws-accounts-2g1d
https://www.youtube.com/watch?v=BfE2G-fsBNU
https://www.youtube.com/watch?v=AKQ7FdEuWz4
https://www.youtube.com/watch?v=Wa0JTKhbsOY
https://github.com/xelerance/Openswan/issues/480
https://openswan.org/
https://gist.github.com/josephspurrier/ea6079a995354b39c948d2ebbdae990f
https://libreswan.org/wiki/FAQ