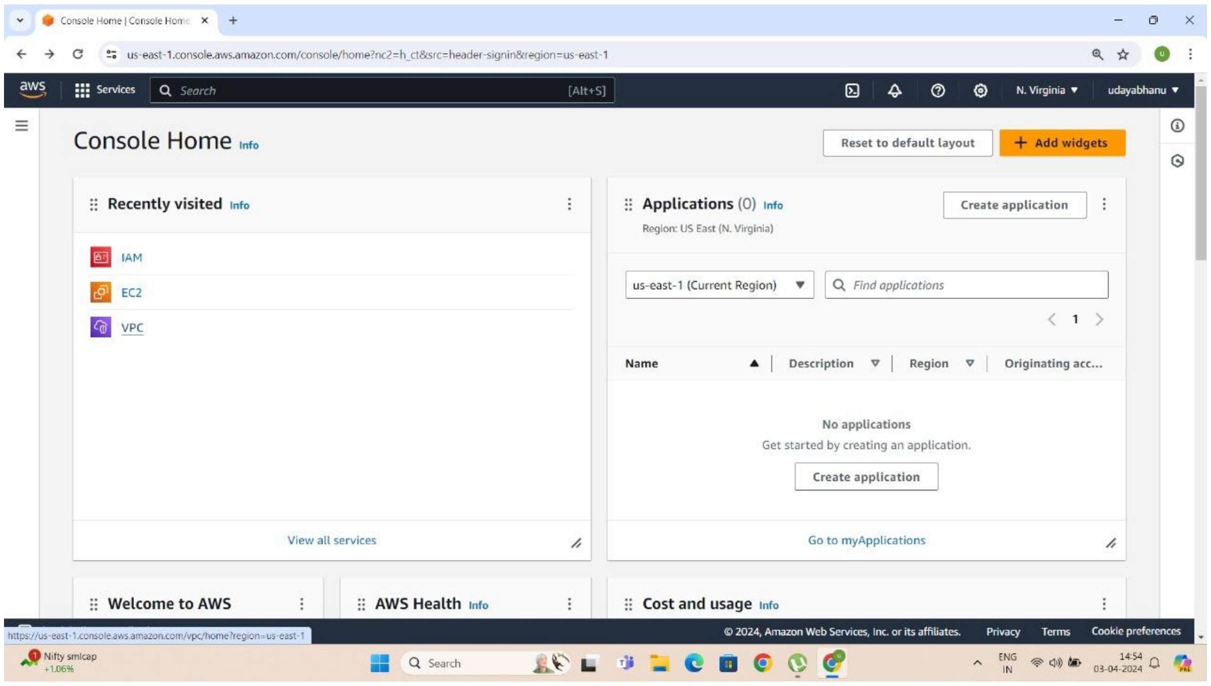
VPC's PEERING

IN DIFFERENT REGIONS

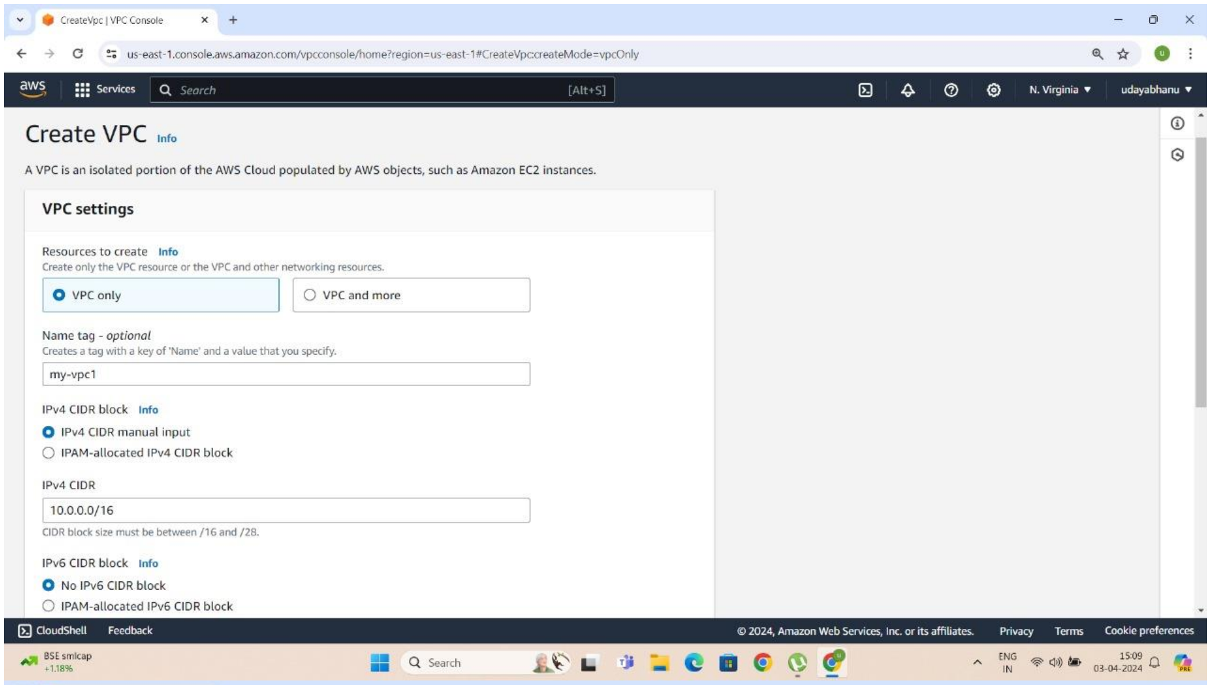
UDAYABHANU
udayabhanu072@gmail.com
Batch:122
Date:04/03/2023

Creation of two VPC'S in different regions and connecting the two vpc's [peering]



No applications
Get started by creating an application.

Step1: log into your aws management console and select desired region here we considered ‘N.Virginia’ as our first region then created ‘my-vpc1’ virtual private cloud with CIDR 10.0.0.0/16


CreateVpc | VPC Console
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateVpc:createMode=vpcOnly
Services
Search
N. Virginia
udayabhanu

Create VPC Info

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

VPC settings

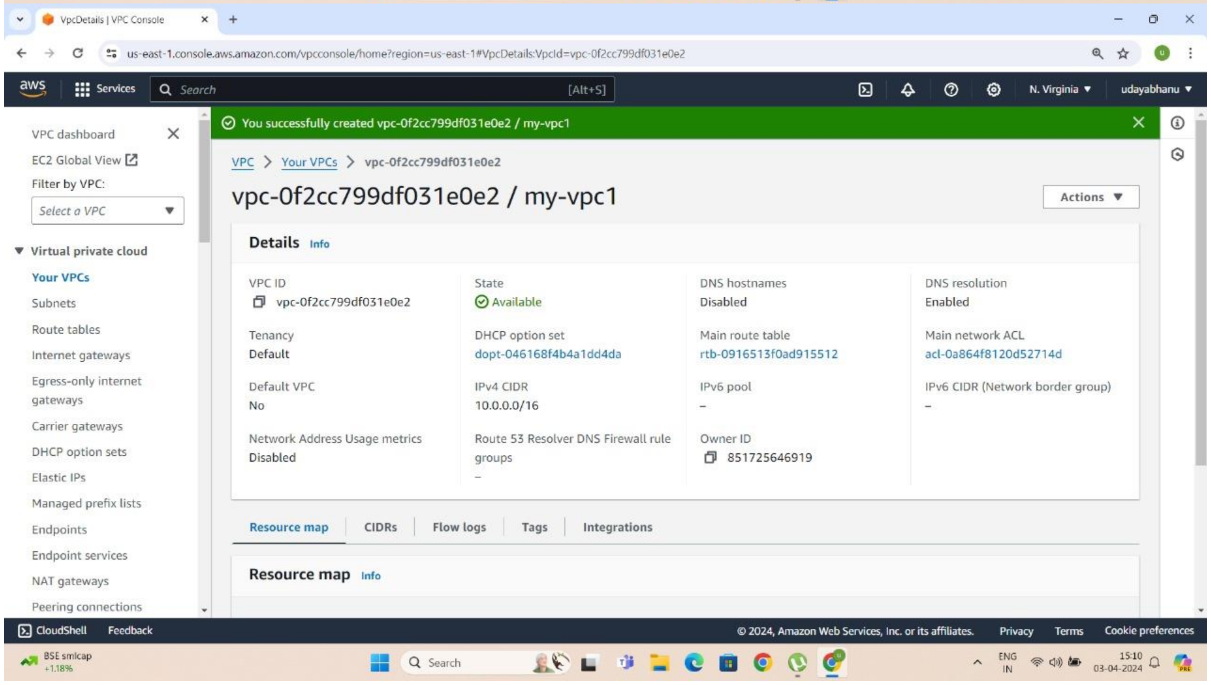
Resources to create Info
Create only the VPC resource or the VPC and other networking resources.
[✓ VPC only] VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
my-vpc1

IPv4 CIDR block Info
[✓ IPv4 CIDR manual input] IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
[✓ No IPv6 CIDR block] IPAM-allocated IPv6 CIDR block



You successfully created vpc-0f2cc799df031e0e2 / my-vpc1

VPC > Your VPCs > vpc-0f2cc799df031e0e2

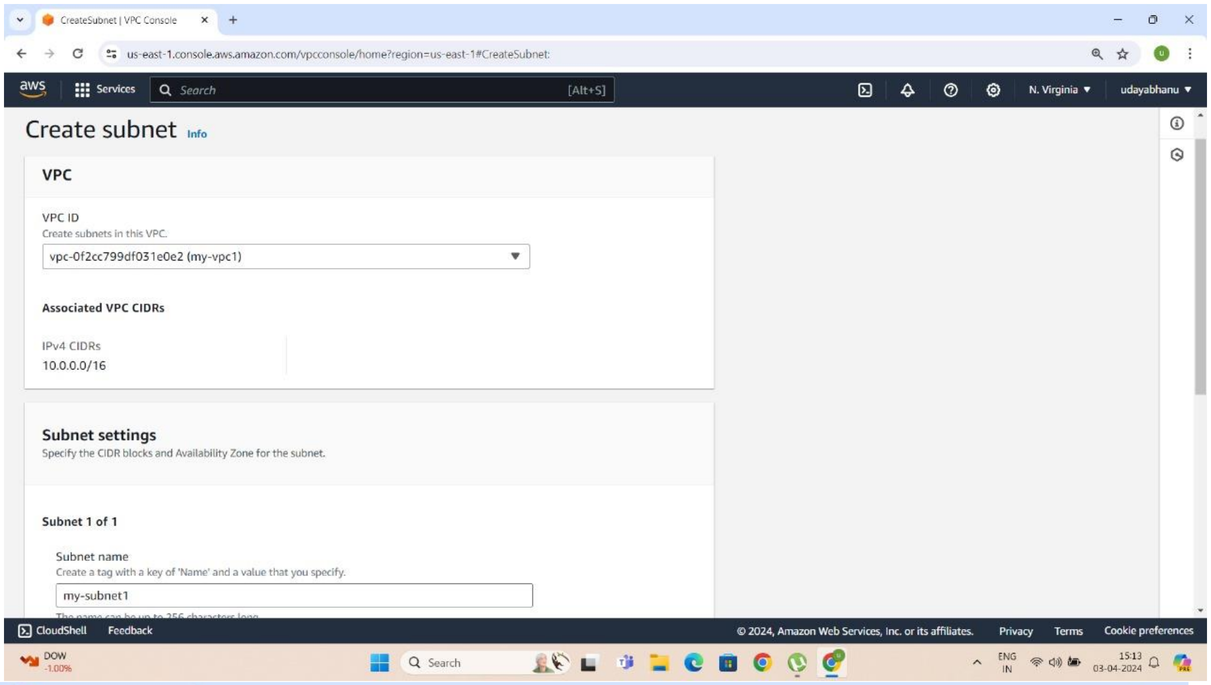
vpc-0f2cc799df031e0e2 / my-vpc1

Actions

Details Info

VPC ID vpc-0f2cc799df031e0e2	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-046168f4b4a1dd4da	Main route table rtb-0916513f0ad915512	Main network ACL acl-0a864f8120d52714d
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –	IPv6 CIDR (Network border group) –
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 851725646919	

Step2: created 'my-subn1' subnet with CIDR 10.0.0.0/24 in the us-east-1a zone



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet:

Create subnet

VPC

VPC ID
Create subnets in this VPC.
vpc-0f2cc799df031e0e2 (my-vpc1)

Associated VPC CIDRs

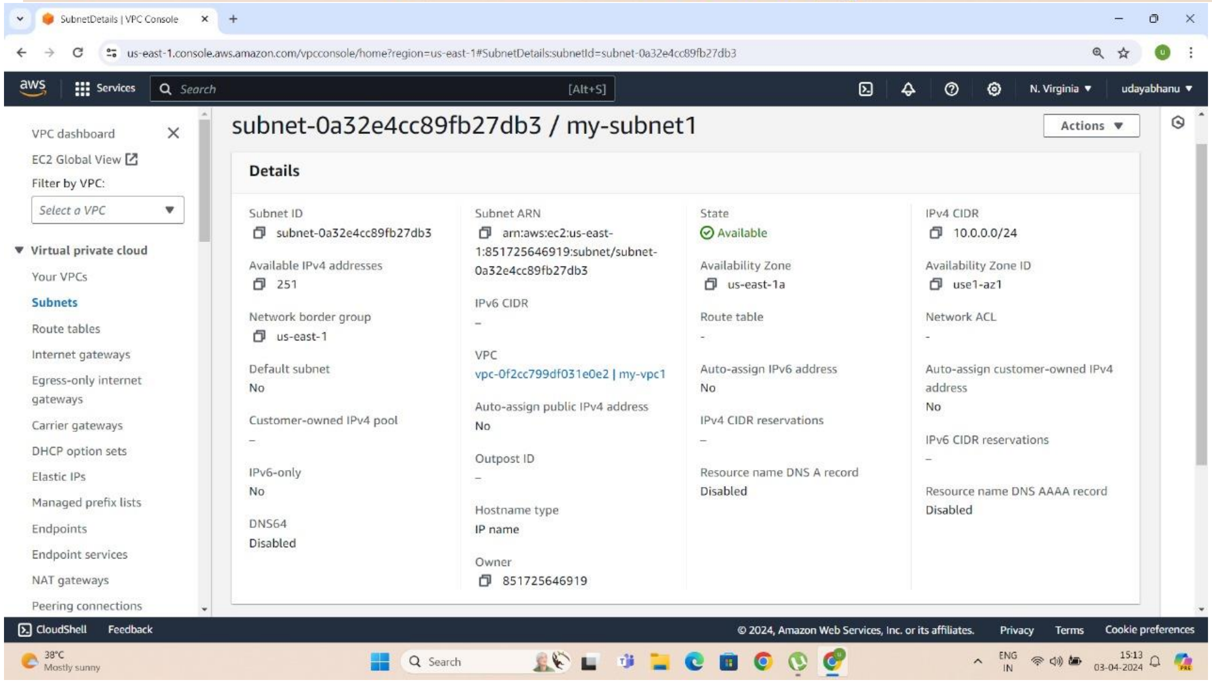
IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
my-subnet1

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



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#SubnetDetails:subnetId=subnet-0a32e4cc89fb27db3

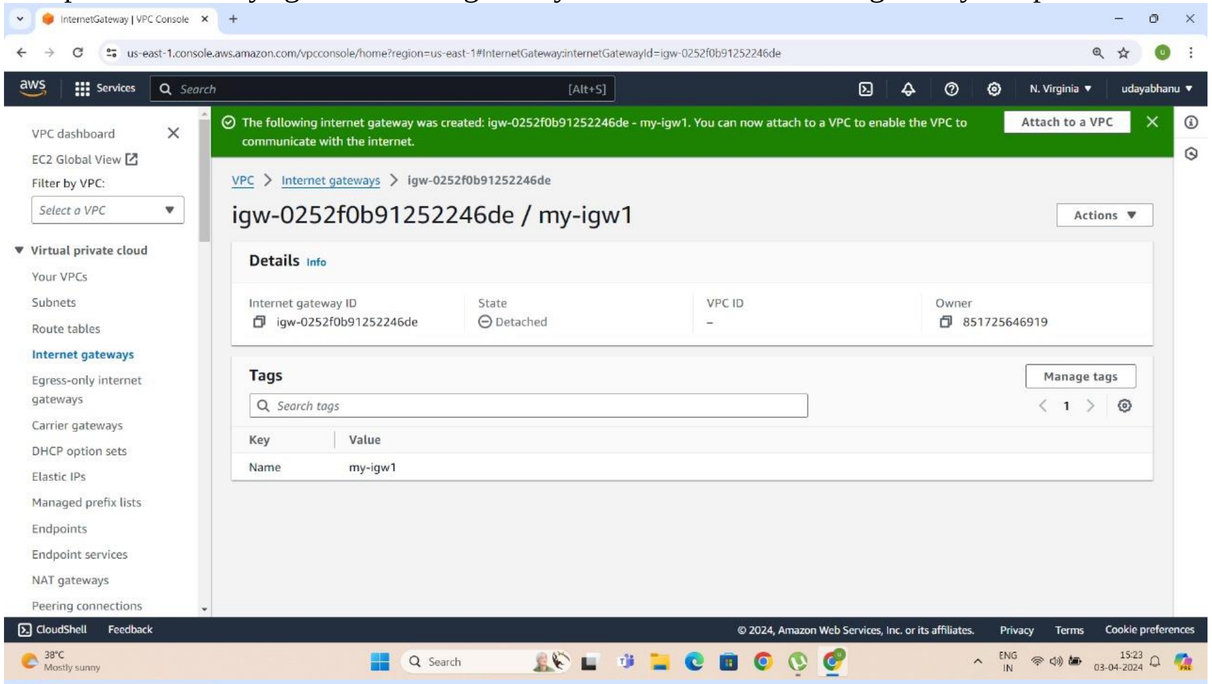
subnet-0a32e4cc89fb27db3 / my-subnet1

Details

Subnet ID: subnet-0a32e4cc89fb27db3	Subnet ARN: arn:aws:ec2:us-east-1:851725646919:subnet/subnet-0a32e4cc89fb27db3	State: Available	IPv4 CIDR: 10.0.0.0/24
Available IPv4 addresses: 251	IPv6 CIDR: –	Availability Zone: us-east-1a	Availability Zone ID: use1-az1
Network border group: us-east-1	VPC: vpc-0f2cc799df031e0e2 | my-vpc1	Route table: –	Network ACL: –
Default subnet: No	Auto-assign public IPv4 address: No	Auto-assign IPv6 address: No	Auto-assign customer-owned IPv4 address: No
Customer-owned IPv4 pool: –	Outpost ID: –	IPv4 CIDR reservations: –	IPv6 CIDR reservations: –
IPv6-only: No	Hostname type: IP name	Resource name DNS A record: Disabled	Resource name DNS AAAA record: Disabled
DNS64: Disabled	Owner: 851725646919		

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

Step3: created ‘my-igw1’ internet gateway and attach the internet gateway to vpc



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#InternetGateway:internetGatewayId=igw-0252f0b91252246de

The following internet gateway was created: igw-0252f0b91252246de - my-igw1. You can now attach to a VPC to enable the VPC to communicate with the internet.

VPC > Internet gateways > igw-0252f0b91252246de

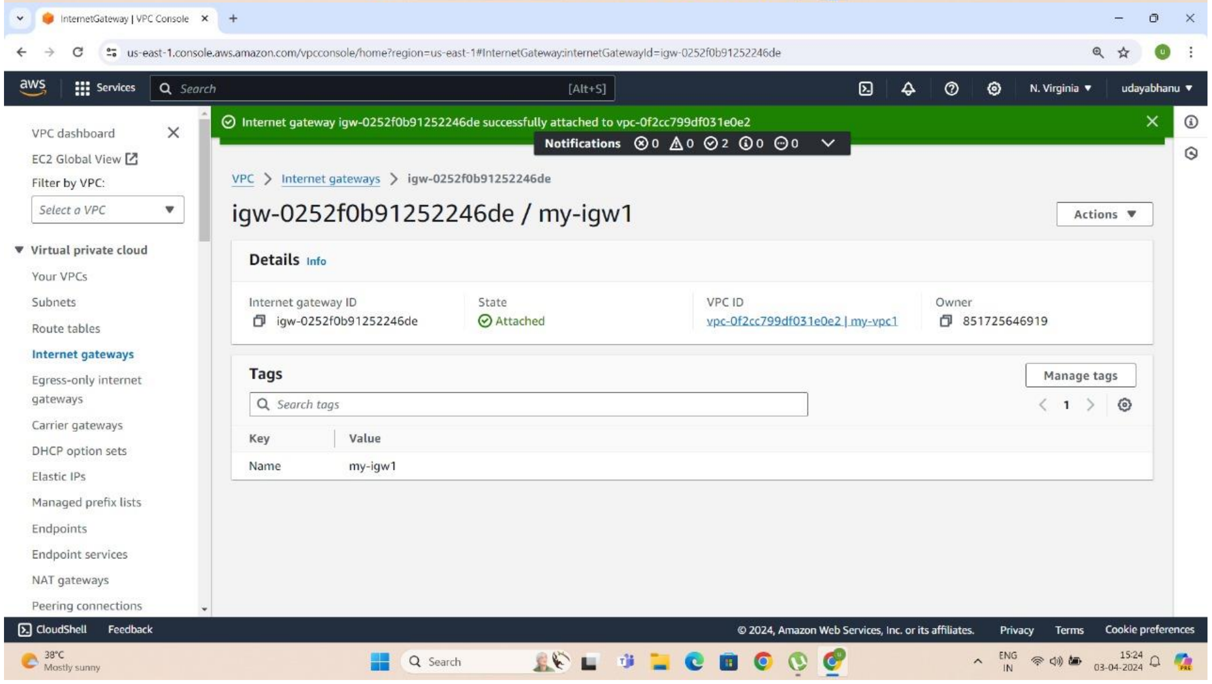
igw-0252f0b91252246de / my-igw1

Details

Internet gateway ID	State	VPC ID	Owner
igw-0252f0b91252246de	Detached	–	851725646919

Tags

Key	Value
Name	my-igw1



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#InternetGateway:internetGatewayId=igw-0252f0b91252246de

Internet gateway igw-0252f0b91252246de successfully attached to vpc-0f2cc799df031e0e2

VPC > Internet gateways > igw-0252f0b91252246de

igw-0252f0b91252246de / my-igw1

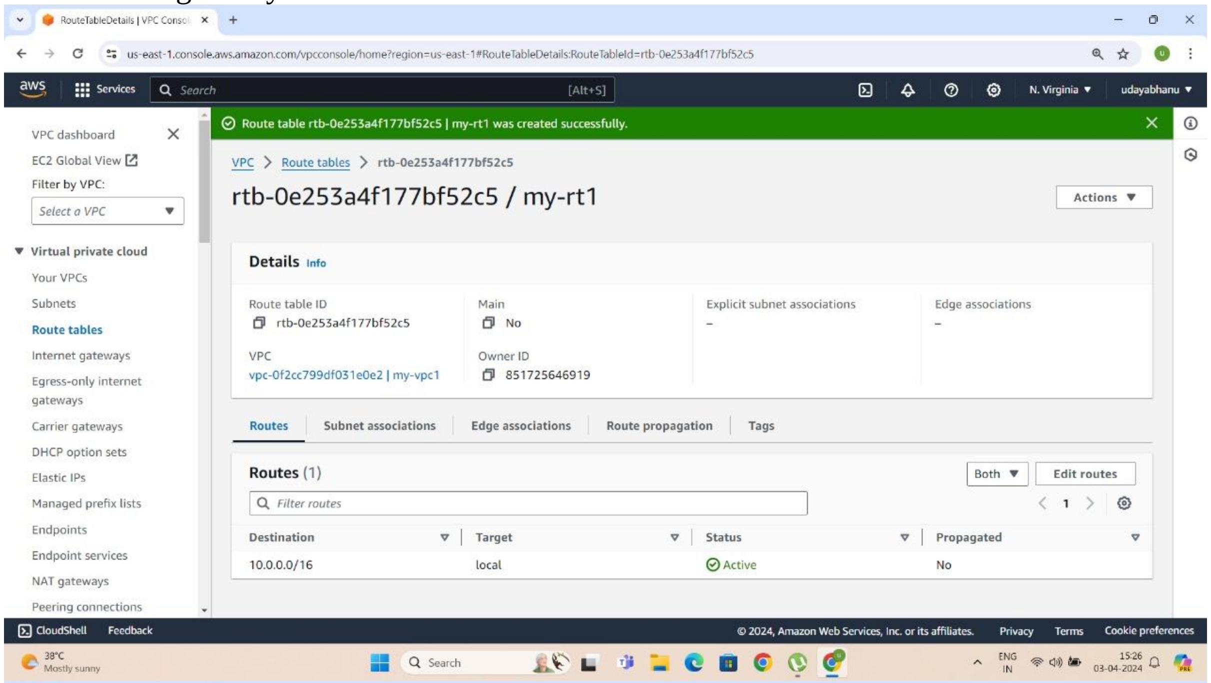
Details

Internet gateway ID	State	VPC ID	Owner
igw-0252f0b91252246de	Attached	vpc-0f2cc799df031e0e2 | my-vpc1	851725646919

Tags

Key	Value
Name	my-igw1

Step4: created 'my-rt1' route table and attached vpc and edit route section weattached the internet gateway we created earlier and associated the subnet



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-0e253a4f177bf52c5

Route table rtb-0e253a4f177bf52c5 | my-rt1 was created successfully.

VPC > Route tables > rtb-0e253a4f177bf52c5

rtb-0e253a4f177bf52c5 / my-rt1

Details Info

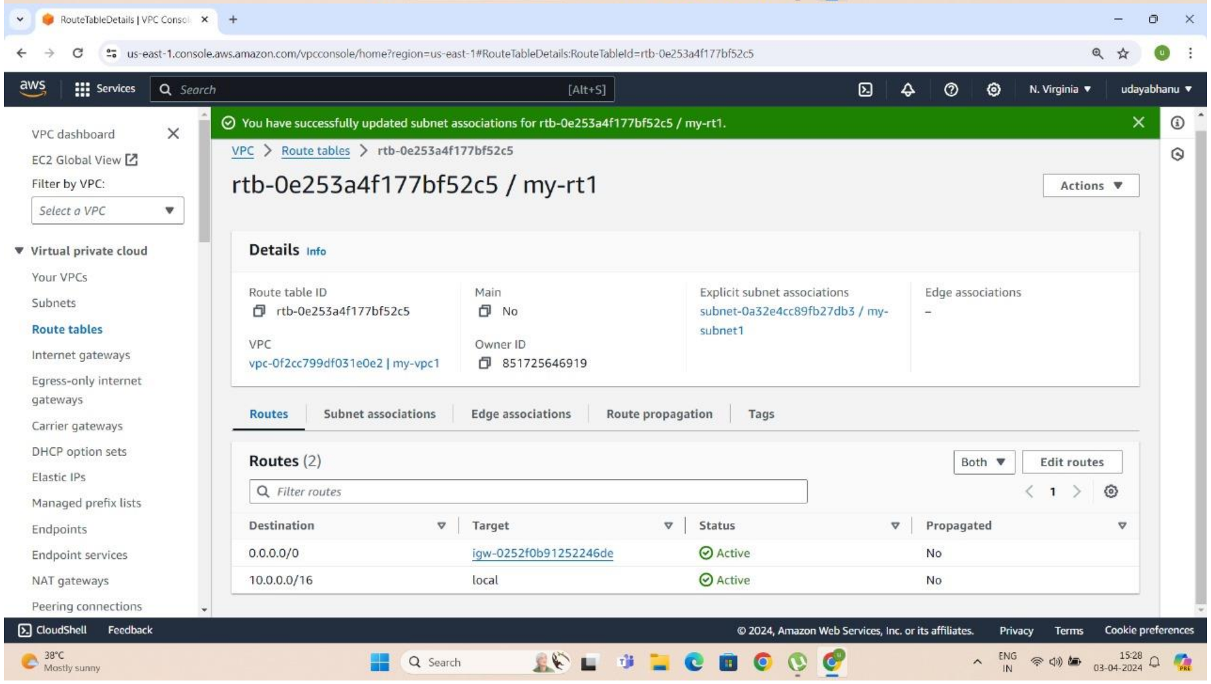
Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0e253a4f177bf52c5	No	–	–
VPC	Owner ID		
vpc-0f2cc799df031e0e2 | my-vpc1	851725646919		

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

38°C Mostly sunny 15:26 03-04-2024



us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-0e253a4f177bf52c5

You have successfully updated subnet associations for rtb-0e253a4f177bf52c5 / my-rt1.

VPC > Route tables > rtb-0e253a4f177bf52c5

rtb-0e253a4f177bf52c5 / my-rt1

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0e253a4f177bf52c5	No	subnet-0a32e4cc89fb27db3 / my-subnet1	–
VPC	Owner ID		
vpc-0f2cc799df031e0e2 | my-vpc1	851725646919		

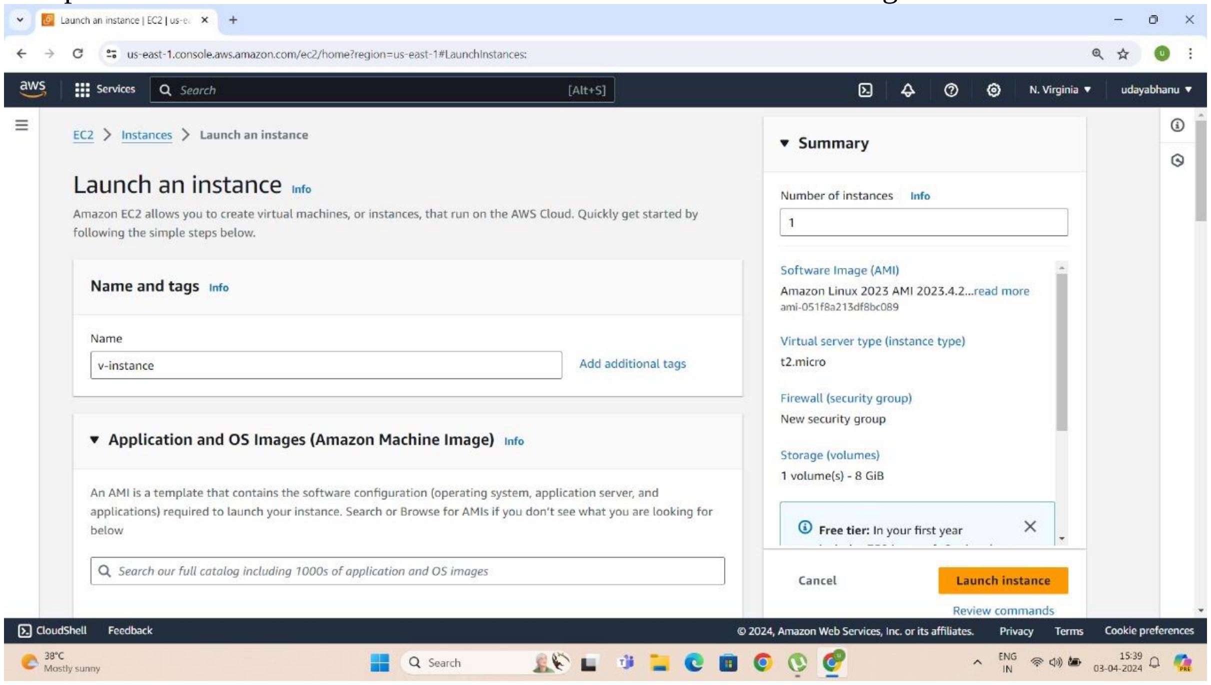
Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0252f0b91252246de	Active	No
10.0.0.0/16	local	Active	No

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

38°C Mostly sunny 15:28 03-04-2024

Step5: now create a EC2 instance here we created ‘v-instance’ using Amazon Linux


Launch an instance | EC2 | us-e
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Services
Search [Alt+S]
N. Virginia
udayabhanu
EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: v-instance

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Summary

Number of instances Info: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.4.2...read more ami-051f8a213df8bc089

Virtual server type (instance type): t2.micro

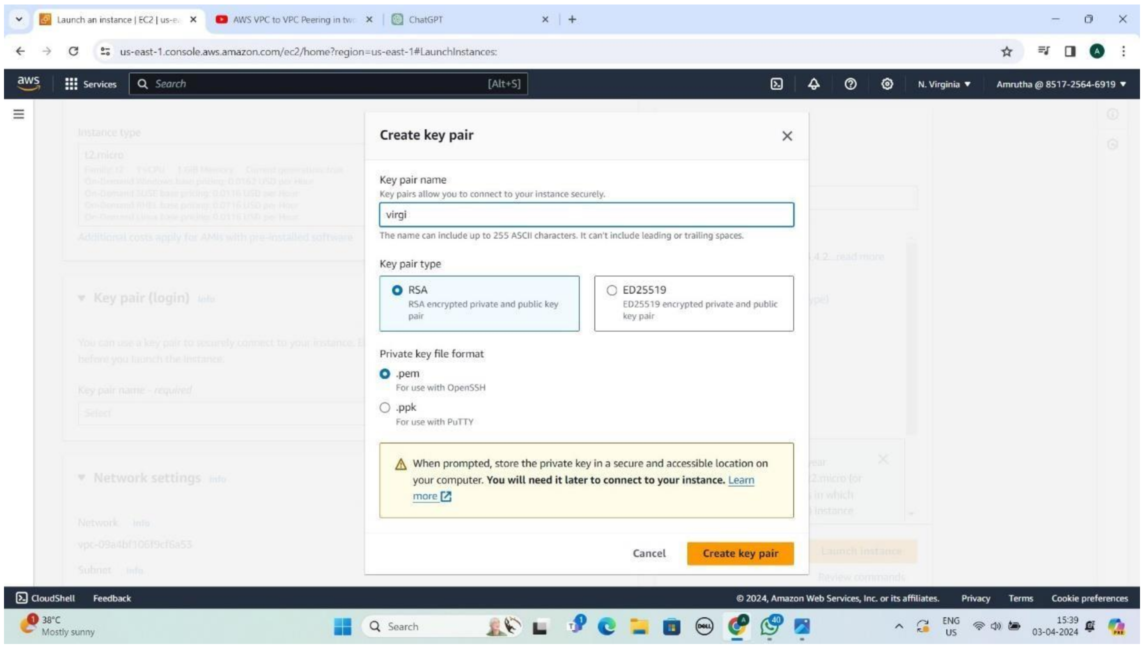
Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year

Cancel | Launch instance | Review commands

Step6: created ‘virgi’ keypair in .pem file formate


Launch an instance | EC2 | us-e
AWS VPC to VPC Peering in tw
ChatGPT
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Services
Search [Alt+S]
N. Virginia
Amrutha @ 8517-2564-6919

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

virgi

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

[✓ RSA] RSA encrypted private and public key pair

ED25519 ED25519 encrypted private and public key pair

Private key file format

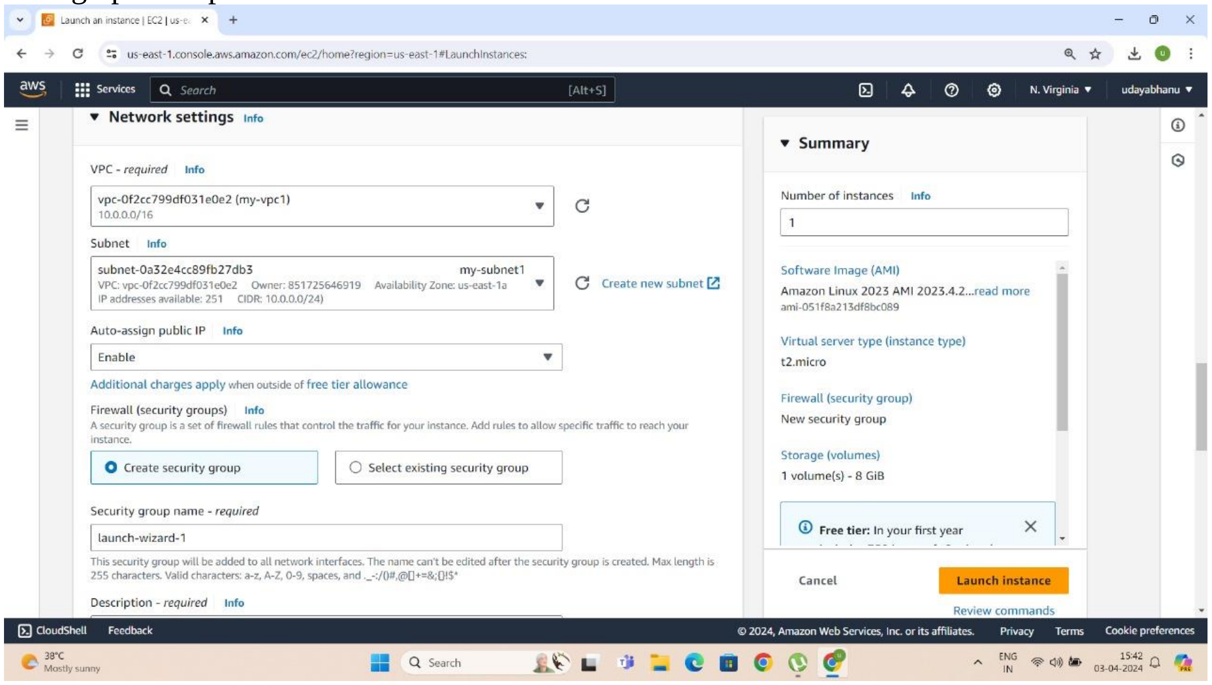
[✓ .pem] For use with OpenSSH

.ppk For use with PuTTY

When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Cancel | Create key pair

Step7:then edit network settings by selecting appropriate vpc and subnet then allow auto assign public ip then launch instance



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

Network settings

VPC - *required*
vpc-0f2cc799df031e0e2 (my-vpc1)
10.0.0.0/16

Subnet
subnet-0a32e4cc89fb27db3 my-subnet1
VPC: vpc-0f2cc799df031e0e2 Owner: 851725646919 Availability Zone: us-east-1a
IP addresses available: 251 CIDR: 10.0.0.0/24)

Auto-assign public IP
Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)
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*

Summary

Number of instances
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.4.2...read more
ami-051f8a213df8bc089

Virtual server type (instance type)
t2.micro

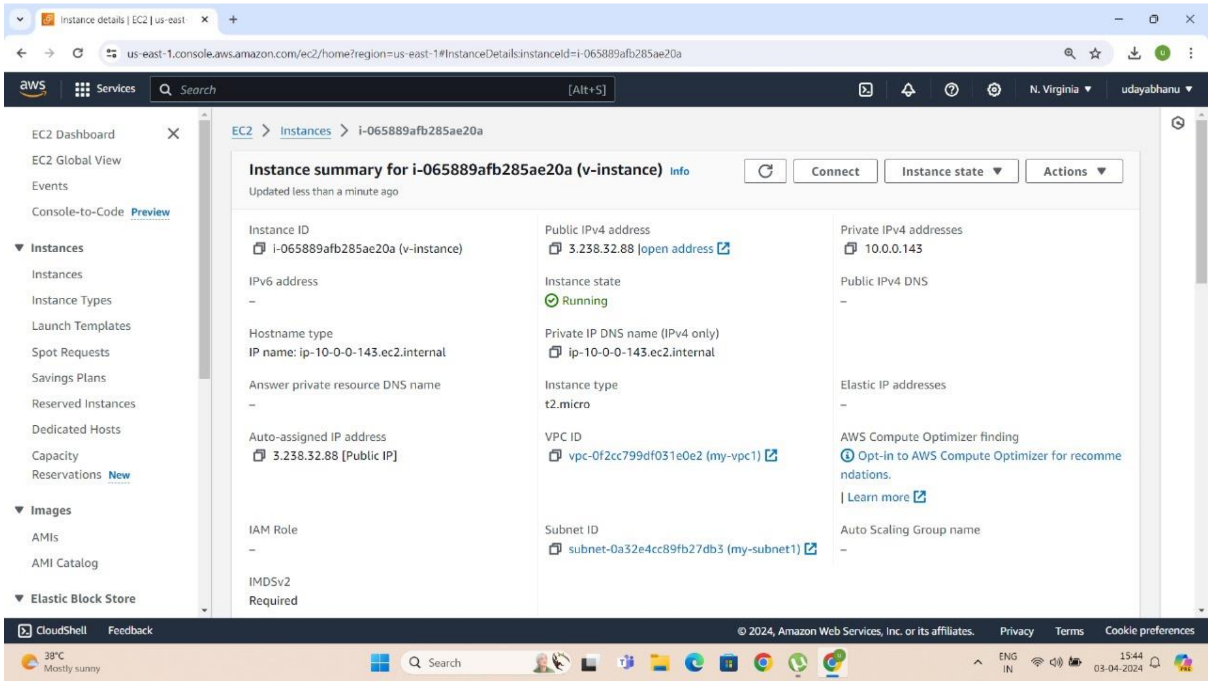
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year

Cancel | Launch instance

Review commands



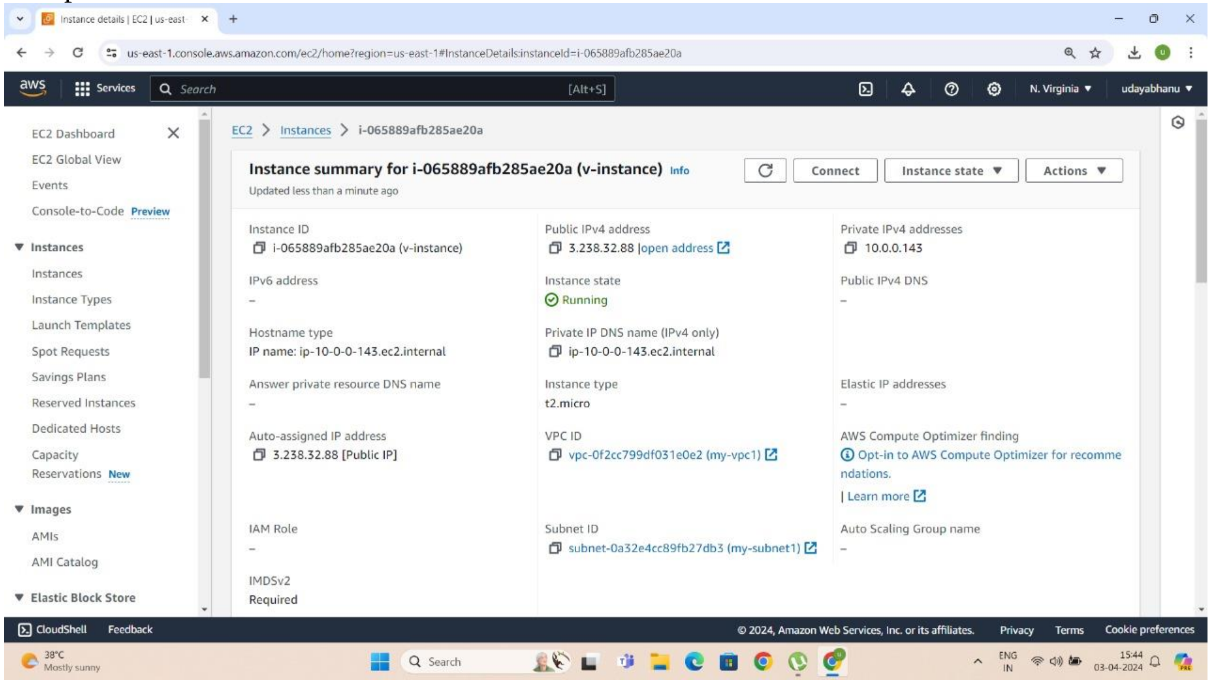
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-065889afb285ae20a

EC2 > Instances > i-065889afb285ae20a

Instance summary for i-065889afb285ae20a (v-instance)
Updated less than a minute ago

Instance ID: i-065889afb285ae20a (v-instance)
Public IPv4 address: 3.238.32.88 | open address
Private IPv4 addresses: 10.0.0.143
IPv6 address: –
Instance state: Running
Public IPv4 DNS: –
Hostname type: IP name: ip-10-0-0-143.ec2.internal
Private IP DNS name (IPv4 only): ip-10-0-0-143.ec2.internal
Answer private resource DNS name: –
Instance type: t2.micro
Elastic IP addresses: –
Auto-assigned IP address: 3.238.32.88 [Public IP]
VPC ID: vpc-0f2cc799df031e0e2 (my-vpc1)
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –
Subnet ID: subnet-0a32e4cc89fb27db3 (my-subnet1)
Auto Scaling Group name: –
IMDSv2: Required

Step8:connect to the EC2 instance



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-065889afb285ae20a

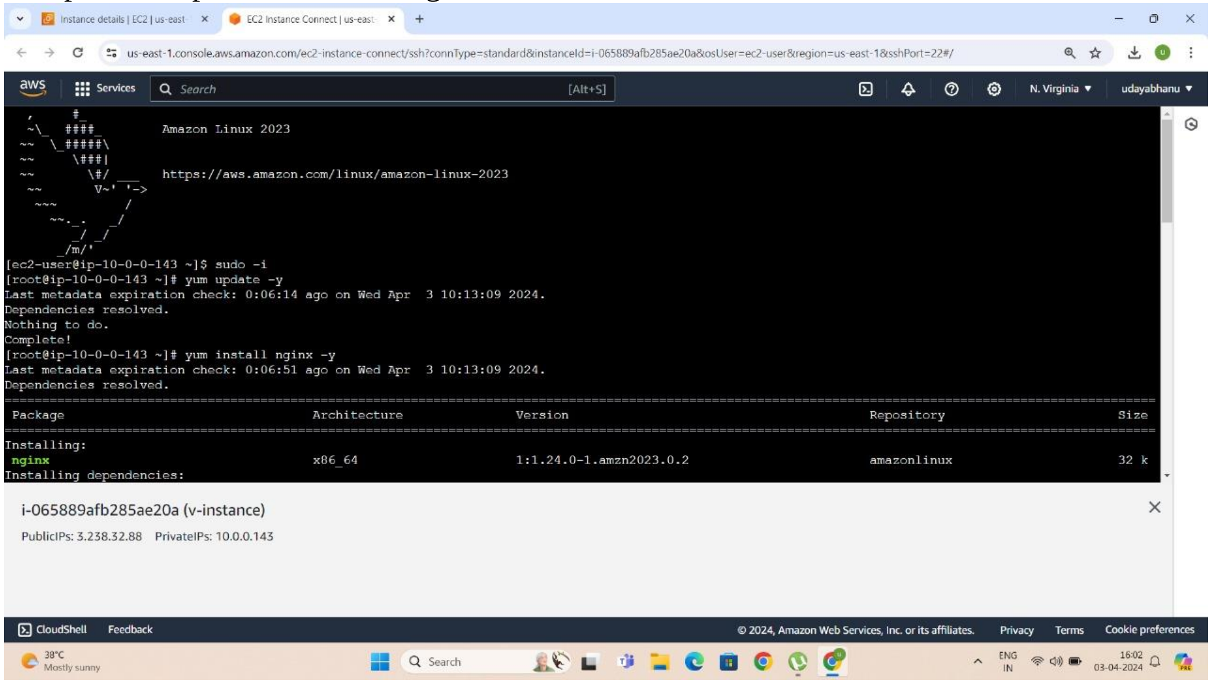
EC2 > Instances > i-065889afb285ae20a

Instance summary for i-065889afb285ae20a (v-instance) Info

Updated less than a minute ago

Instance ID	Public IPv4 address	Private IPv4 addresses
i-065889afb285ae20a (v-instance)	3.238.32.88 |open address	10.0.0.143
IPv6 address	Instance state	Public IPv4 DNS
–	Running	–
Hostname type	Private IP DNS name (IPv4 only)	
IP name: ip-10-0-0-143.ec2.internal	ip-10-0-0-143.ec2.internal	
Answer private resource DNS name	Instance type	Elastic IP addresses
–	t2.micro	–
Auto-assigned IP address	VPC ID	AWS Compute Optimizer finding
3.238.32.88 [Public IP]	vpc-0f2cc799df031e0e2 (my-vpc1)	Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role	Subnet ID	Auto Scaling Group name
–	subnet-0a32e4cc89fb27db3 (my-subnet1)	–
IMDSv2		
Required		

Step9: then update and install nginx and create a file and insert the content and save it

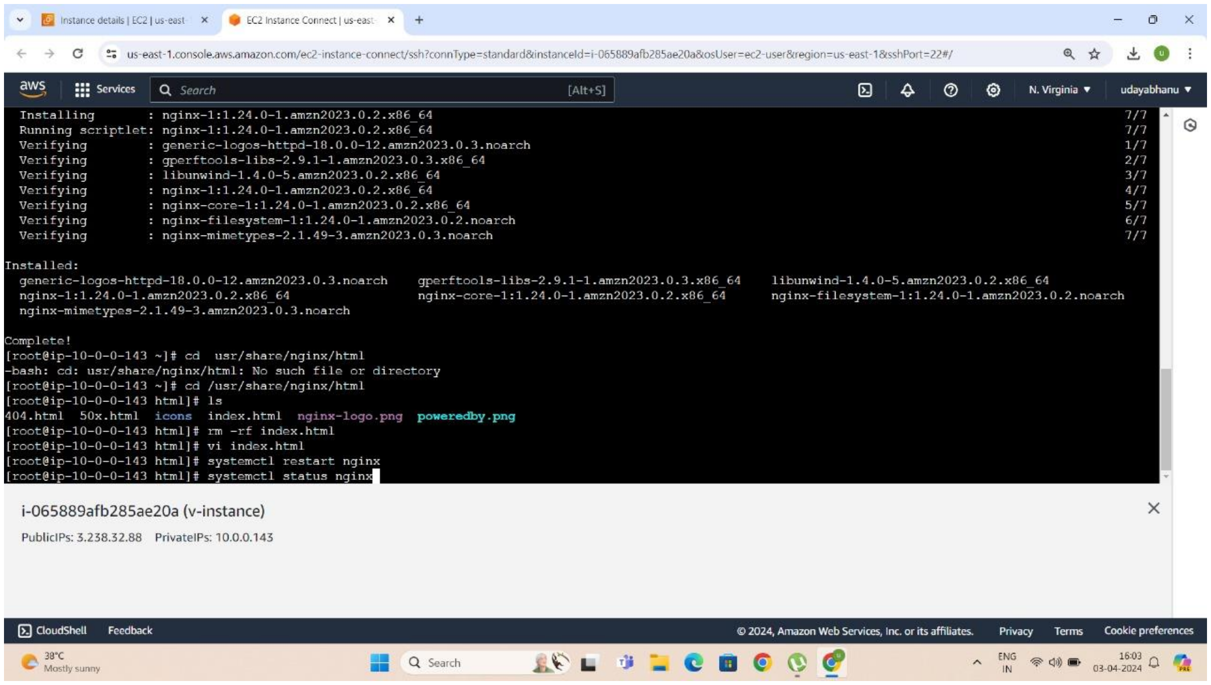


us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-065889afb285ae20a&osUser=ec2-user&region=us-east-1&sshPort=22#/

```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-0-143 ~]$ sudo -i
[root@ip-10-0-0-143 ~]# yum update -y
Last metadata expiration check: 0:06:14 ago on Wed Apr  3 10:13:09 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-0-143 ~]# yum install nginx -y
Last metadata expiration check: 0:06:51 ago on Wed Apr  3 10:13:09 2024.
Dependencies resolved.
================================================================================================================
 Package                    Architecture         Version                               Repository          Size
================================================================================================================
Installing:
 nginx                      x86_64               1:1.24.0-1.amzn2023.0.2               amazonlinux         32 k
Installing dependencies:
```

i-065889afb285ae20a (v-instance)

PublicIPs: 3.238.32.88 PrivateIPs: 10.0.0.143

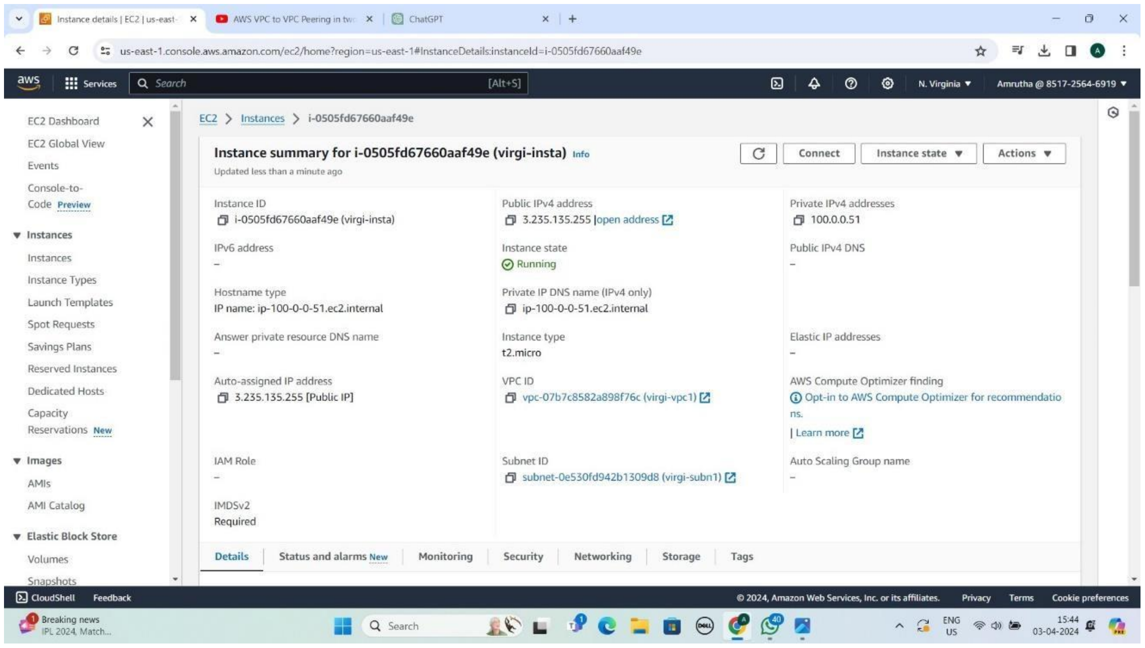


```
 Installing        : nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                                                          7/7
 Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                                                           7/7
 Verifying         : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                                             1/7
 Verifying         : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                                                   2/7
 Verifying         : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                                                         3/7
 Verifying         : nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                                                          4/7
 Verifying         : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64                                                                     5/7
 Verifying         : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch                                                               6/7
 Verifying         : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                                                  7/7

Installed:
 generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch    gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64    libunwind-1.4.0-5.amzn2023.0.2.x86_64
 nginx-1:1.24.0-1.amzn2023.0.2.x86_64                 nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64      nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
 nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-10-0-0-143 ~]# cd  usr/share/nginx/html
-bash: cd: usr/share/nginx/html: No such file or directory
[root@ip-10-0-0-143 ~]# cd /usr/share/nginx/html
[root@ip-10-0-0-143 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-10-0-0-143 html]# rm -rf index.html
[root@ip-10-0-0-143 html]# vi index.html
[root@ip-10-0-0-143 html]# systemctl restart nginx
[root@ip-10-0-0-143 html]# systemctl status nginx
```

i-065889afb285ae20a (v-instance)

PublicIPs: 3.238.32.88   PrivateIPs: 10.0.0.143

Step10: Then in the Security section click on the security groups and edit the in-bound rules by adding the all traffic type and save rules

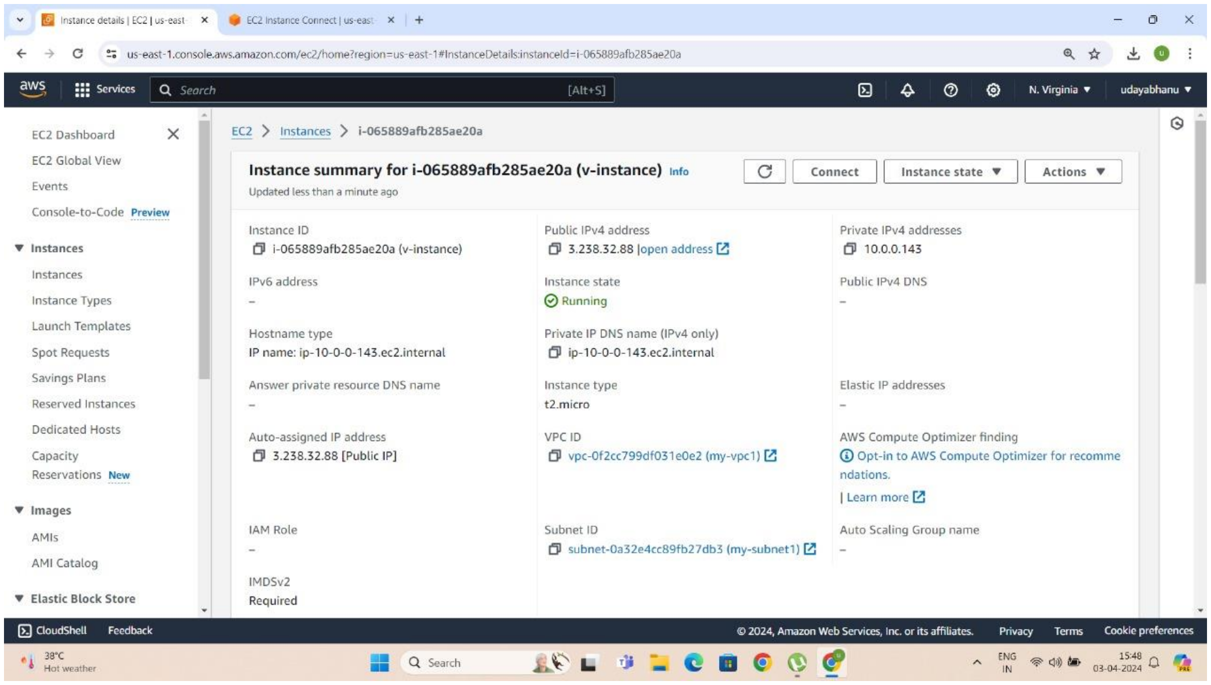


EC2 > Instances > i-0505fd67660aaf49e

Instance summary for i-0505fd67660aaf49e (virgi-insta)

Updated less than a minute ago

Instance ID: i-0505fd67660aaf49e (virgi-insta)	Public IPv4 address: 3.235.135.255 | open address	Private IPv4 addresses: 100.0.0.51
IPv6 address: –	Instance state: Running	Public IPv4 DNS: –
Hostname type: IP name: ip-100-0-0-51.ec2.internal	Private IP DNS name (IPv4 only): ip-100-0-0-51.ec2.internal	
Answer private resource DNS name: –	Instance type: t2.micro	Elastic IP addresses: –
Auto-assigned IP address: 3.235.135.255 [Public IP]	VPC ID: vpc-07b7c8582a898f76c (virgi-vpc1)	AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –	Subnet ID: subnet-0e530fd942b1309d8 (virgi-subn1)	Auto Scaling Group name: –
IMDSv2: Required		


Instance details | EC2 | us-east · EC2 Instance Connect | us-east

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-065889afb285ae20a

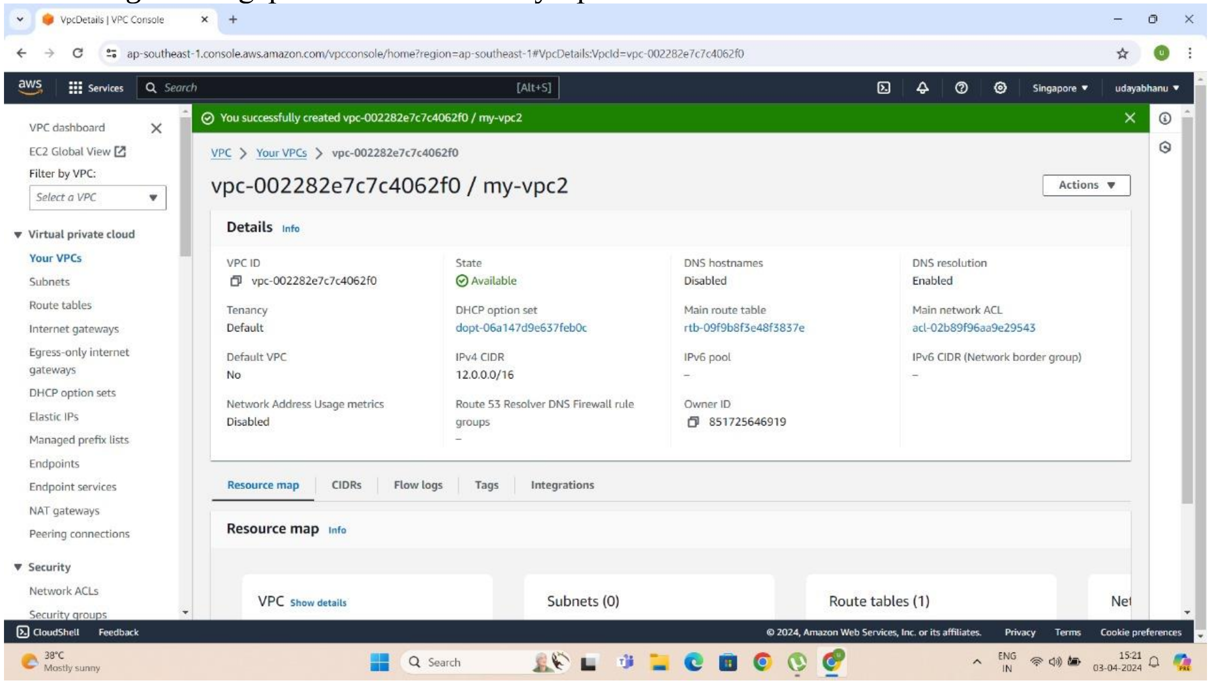
EC2 > Instances > i-065889afb285ae20a

Instance summary for i-065889afb285ae20a (v-instance) Info

Updated less than a minute ago

Instance ID	Public IPv4 address	Private IPv4 addresses
i-065889afb285ae20a (v-instance)	3.238.32.88 | open address	10.0.0.143
IPv6 address	Instance state	Public IPv4 DNS
–	Running	–
Hostname type	Private IP DNS name (IPv4 only)	
IP name: ip-10-0-0-143.ec2.internal	ip-10-0-0-143.ec2.internal	
Answer private resource DNS name	Instance type	Elastic IP addresses
–	t2.micro	–
Auto-assigned IP address	VPC ID	AWS Compute Optimizer finding
3.238.32.88 [Public IP]	vpc-0f2cc799df031e0e2 (my-vpc1)	Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role	Subnet ID	Auto Scaling Group name
–	subnet-0a32e4cc89fb27db3 (my-subnet1)	–
IMDSv2		
Required		

Step11: In the similar way we create another vpc in another region here we take our second region 'singapore' and created 'my-vpc2' with CIDR 12.0.0.0/16


ap-southeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-southeast-1#VpcDetails:VpcId=vpc-002282e7c7c4062f0

You successfully created vpc-002282e7c7c4062f0 / my-vpc2

VPC > Your VPCs > vpc-002282e7c7c4062f0

vpc-002282e7c7c4062f0 / my-vpc2

Details Info

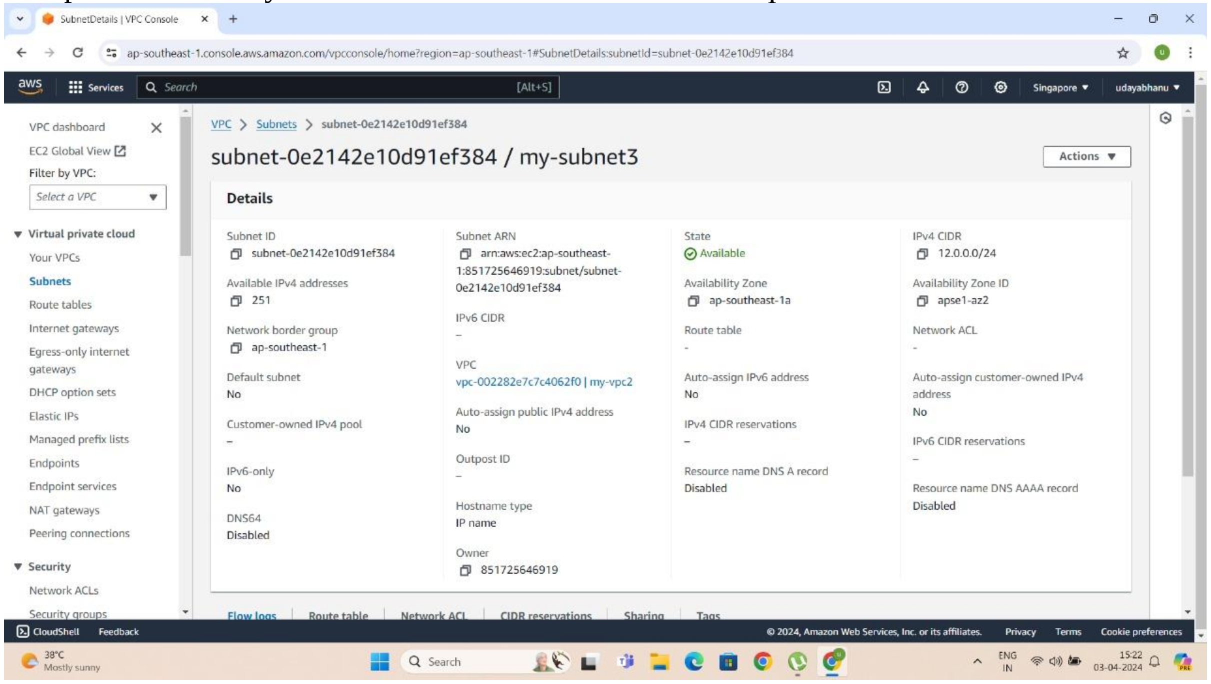
VPC ID	State	DNS hostnames	DNS resolution
vpc-002282e7c7c4062f0	Available	Disabled	Enabled
Tenancy	DHCP option set	Main route table	Main network ACL
Default	dopt-06a147d9e637feb0c	rtb-09f9b8f3e48f3837e	acl-02b89f96aa9e29543
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR (Network border group)
No	12.0.0.0/16	–	–
Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Owner ID	
Disabled	–	851725646919	

Resource map · CIDRs · Flow logs · Tags · Integrations

Resource map Info

VPC Show details · Subnets (0) · Route tables (1)

Step12: created ‘my-subnet3’ with CIDR 12.0.0.0/24 in ‘ap-southeast-1a’



ap-southeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-southeast-1#SubnetDetails:subnetId=subnet-0e2142e10d91ef384

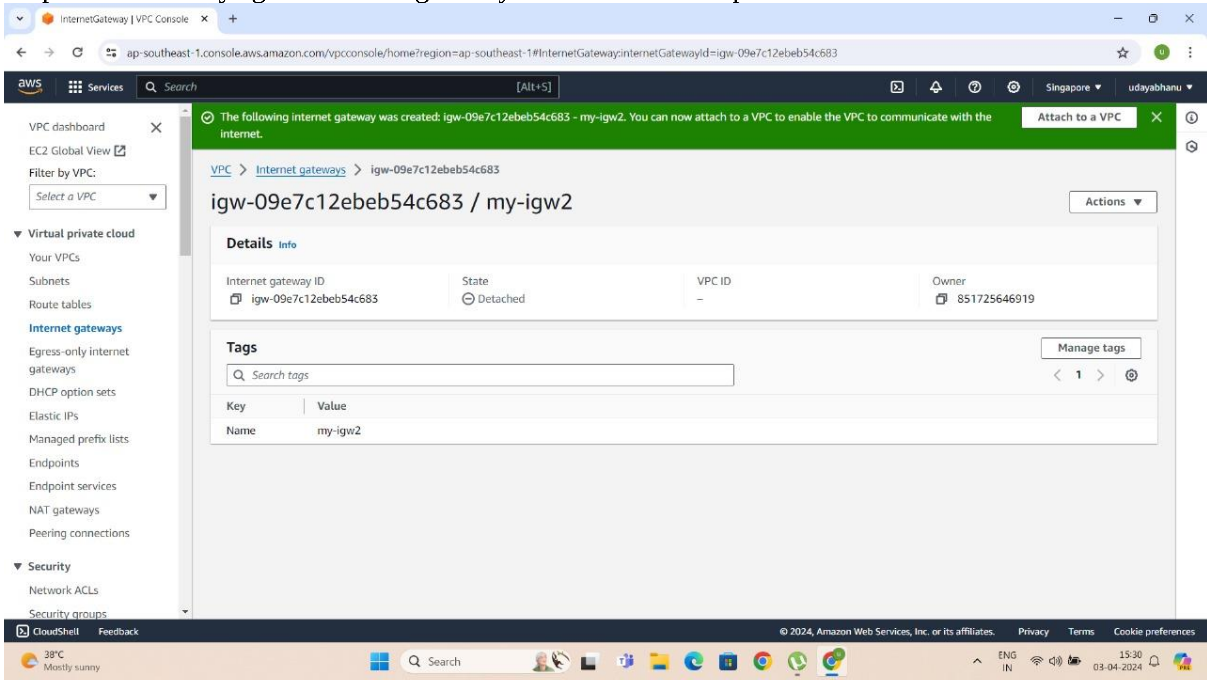
VPC > Subnets > subnet-0e2142e10d91ef384

subnet-0e2142e10d91ef384 / my-subnet3

Details

Subnet ID: subnet-0e2142e10d91ef384
Subnet ARN: arn:aws:ec2:ap-southeast-1:851725646919:subnet/subnet-0e2142e10d91ef384
State: Available
IPv4 CIDR: 12.0.0.0/24
Available IPv4 addresses: 251
IPv6 CIDR: –
Availability Zone: ap-southeast-1a
Availability Zone ID: apse1-az2
Network border group: ap-southeast-1
VPC: vpc-002282e7c7c4062f0 | my-vpc2
Route table: –
Network ACL: –
Default subnet: No
Auto-assign public IPv4 address: No
Auto-assign IPv6 address: No
Auto-assign customer-owned IPv4 address: No
Customer-owned IPv4 pool: –
Outpost ID: –
IPv4 CIDR reservations: –
IPv6 CIDR reservations: –
IPv6-only: No
Hostname type: IP name
Resource name DNS A record: Disabled
Resource name DNS AAAA record: Disabled
DNS64: Disabled
Owner: 851725646919

Step13:created ‘my-igw2’ internet gateway and attached the vpc



ap-southeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-southeast-1#InternetGateway:internetGatewayId=igw-09e7c12ebeb54c683

The following internet gateway was created: igw-09e7c12ebeb54c683 - my-igw2. You can now attach to a VPC to enable the VPC to communicate with the internet.

VPC > Internet gateways > igw-09e7c12ebeb54c683

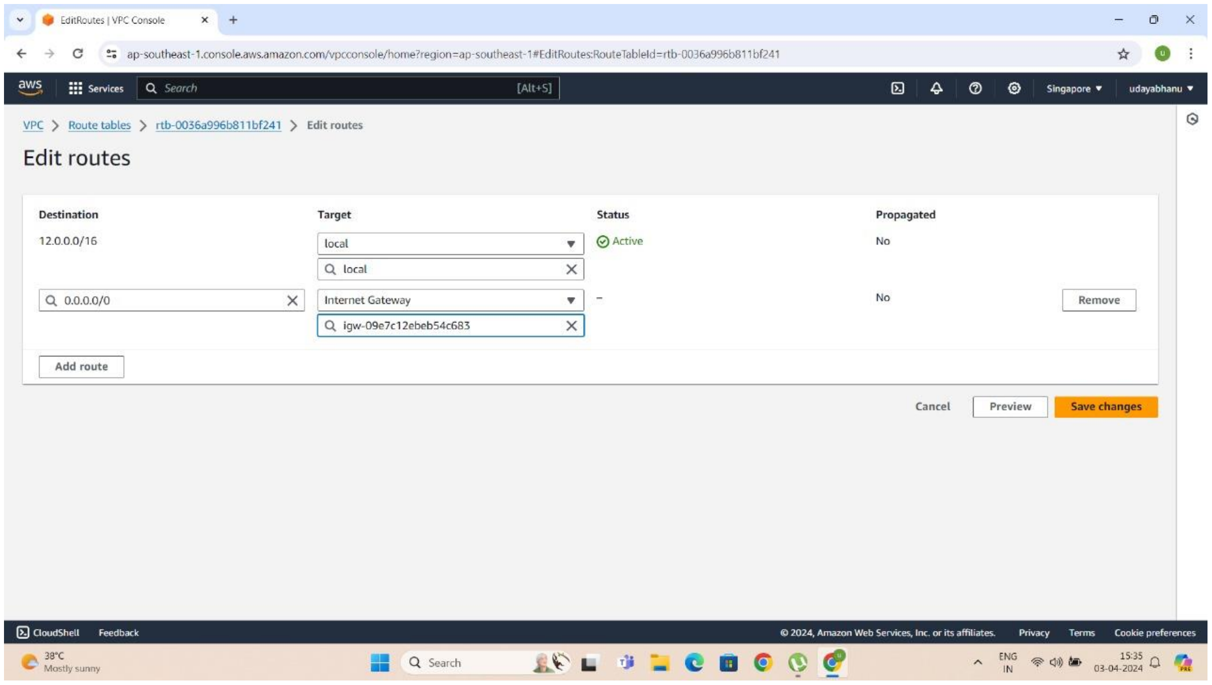
igw-09e7c12ebeb54c683 / my-igw2

Details

Internet gateway ID	State	VPC ID	Owner
igw-09e7c12ebeb54c683	Detached	–	851725646919

Tags

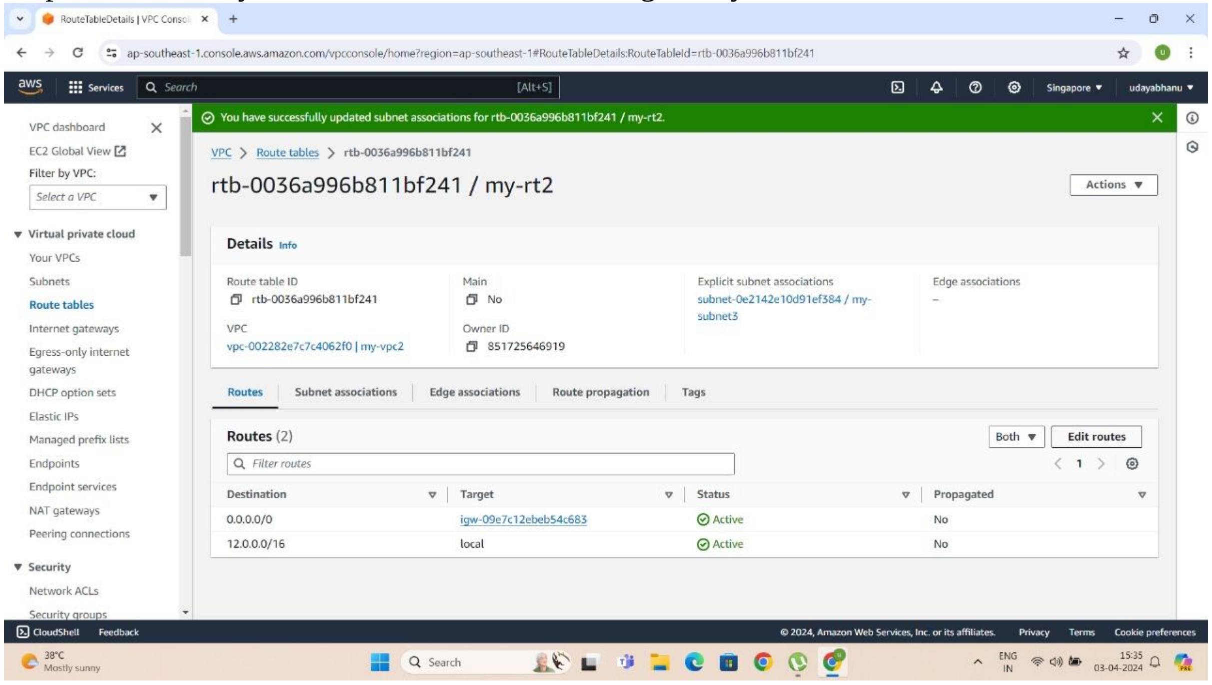
Key	Value
Name	my-igw2



Edit routes

Destination	Target	Status	Propagated
12.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway igw-09e7c12ebeb54c683	-	No

Step14:created 'my-rt2' and attached the internet gateway and associated the subnet



You have successfully updated subnet associations for rtb-0036a996b811bf241 / my-rt2.

rtb-0036a996b811bf241 / my-rt2

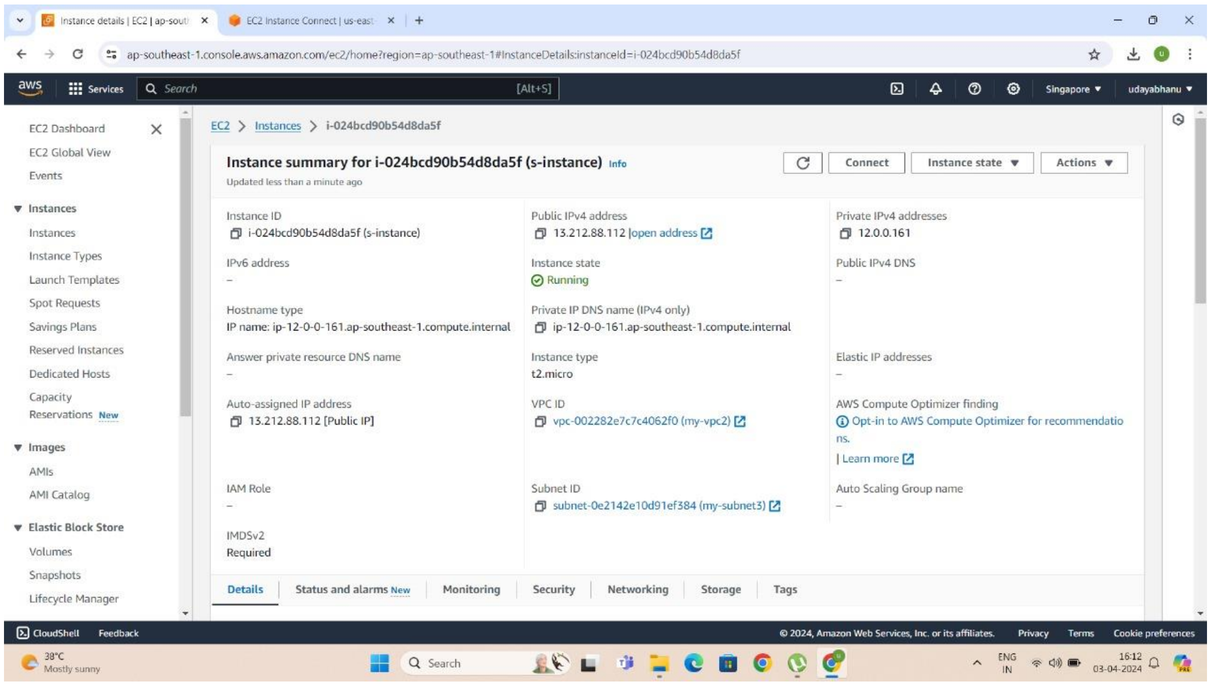
Details

Route table ID: rtb-0036a996b811bf241
Main: No
Explicit subnet associations: subnet-0e2142e10d91ef384 / my-subnet3
Edge associations: –
VPC: vpc-002282e7c7c4062f0 | my-vpc2
Owner ID: 851725646919

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-09e7c12ebeb54c683	Active	No
12.0.0.0/16	local	Active	No

Step15: now create a EC2 instance here we created ‘s-instance’ using Amazon Linux



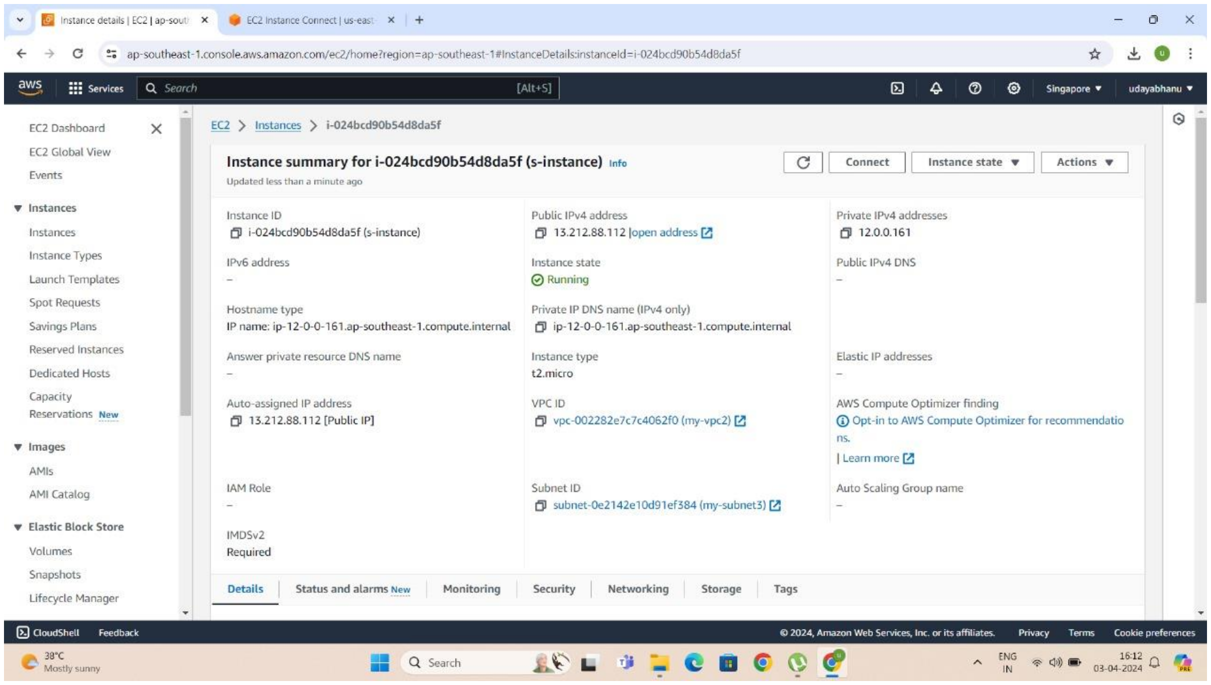
ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#InstanceDetails:instanceId=i-024bcd90b54d8da5f

EC2 > Instances > i-024bcd90b54d8da5f

Instance summary for i-024bcd90b54d8da5f (s-instance) Info
Updated less than a minute ago

Instance ID i-024bcd90b54d8da5f (s-instance)	Public IPv4 address 13.212.88.112 | open address	Private IPv4 addresses 12.0.0.161
IPv6 address –	Instance state Running	Public IPv4 DNS –
Hostname type IP name: ip-12-0-0-161.ap-southeast-1.compute.internal	Private IP DNS name (IPv4 only) ip-12-0-0-161.ap-southeast-1.compute.internal	
Answer private resource DNS name –	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 13.212.88.112 [Public IP]	VPC ID vpc-002282e7c7c4062f0 (my-vpc2)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-0e2142e10d91ef384 (my-subnet3)	Auto Scaling Group name –
IMDSv2 Required		

Step16: created a then edit network settings by selecting appropriate vpc and subnet then allow auto assign public ip then launch instance and in the Security section click on the security groups and edit the in-bound rules by adding the all traffic type and save rules



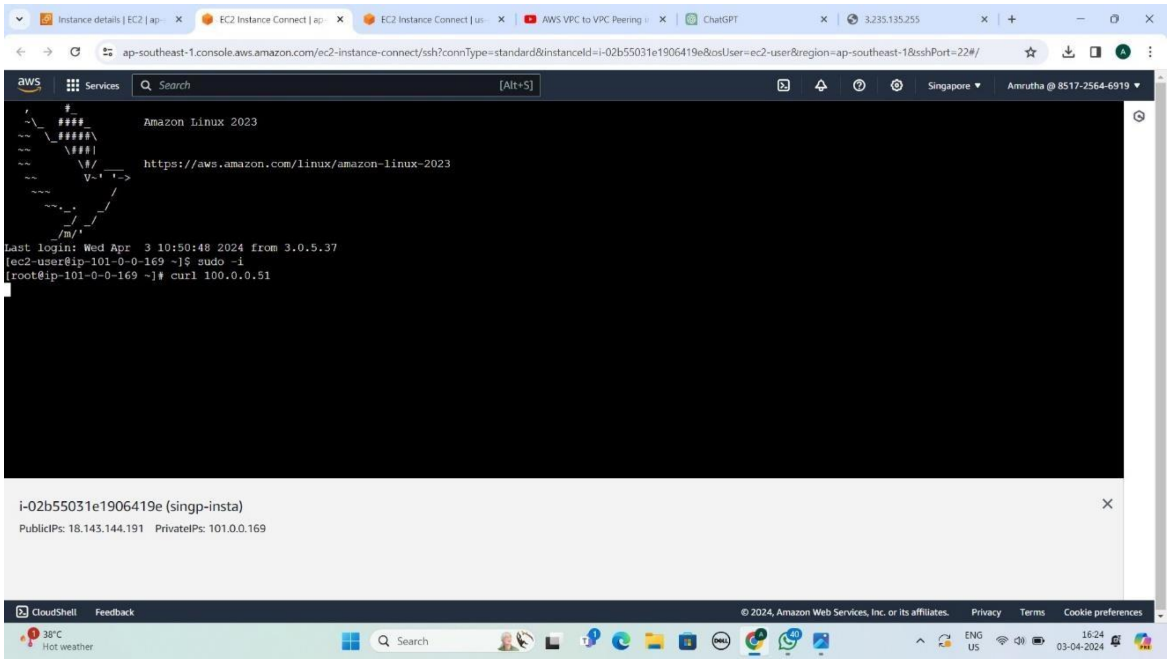
ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#InstanceDetails:instanceId=i-024bcd90b54d8da5f

EC2 > Instances > i-024bcd90b54d8da5f

Instance summary for i-024bcd90b54d8da5f (s-instance) Info
Updated less than a minute ago

Instance ID i-024bcd90b54d8da5f (s-instance); Public IPv4 address 13.212.88.112 | open address; Private IPv4 addresses 12.0.0.161; IPv6 address –; Instance state Running; Public IPv4 DNS –; Hostname type IP name: ip-12-0-0-161.ap-southeast-1.compute.internal; Private IP DNS name (IPv4 only) ip-12-0-0-161.ap-southeast-1.compute.internal; Answer private resource DNS name –; Instance type t2.micro; Elastic IP addresses –; Auto-assigned IP address 13.212.88.112 [Public IP]; VPC ID vpc-002282e7c7c4062f0 (my-vpc2); AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more; IAM Role –; Subnet ID subnet-0e2142e10d91ef384 (my-subnet3); Auto Scaling Group name –; IMDSv2 Required

Step17:connect to the instance and try to access the private ip in signapore region it gives a failure as there is no connection between them

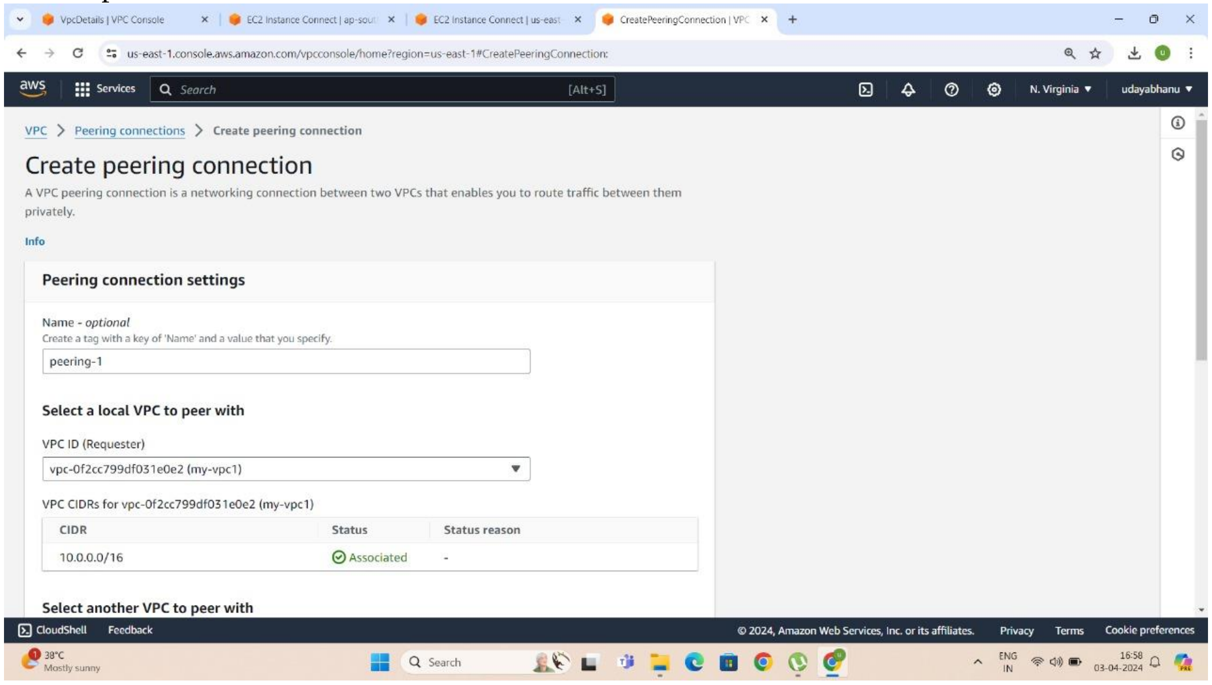


```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Apr  3 10:50:48 2024 from 3.0.5.37
[ec2-user@ip-101-0-0-169 ~]$ sudo -i
[root@ip-101-0-0-169 ~]# curl 100.0.0.51
```

i-02b55031e1906419e (singp-insta)

PublicIPs: 18.143.144.191 PrivateIPs: 101.0.0.169

Step18:create a peering connection in virgina here we created 'peering-1' and attach localvpc



VPC > Peering connections > Create peering connection

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately.

Peering connection settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
peering-1

Select a local VPC to peer with

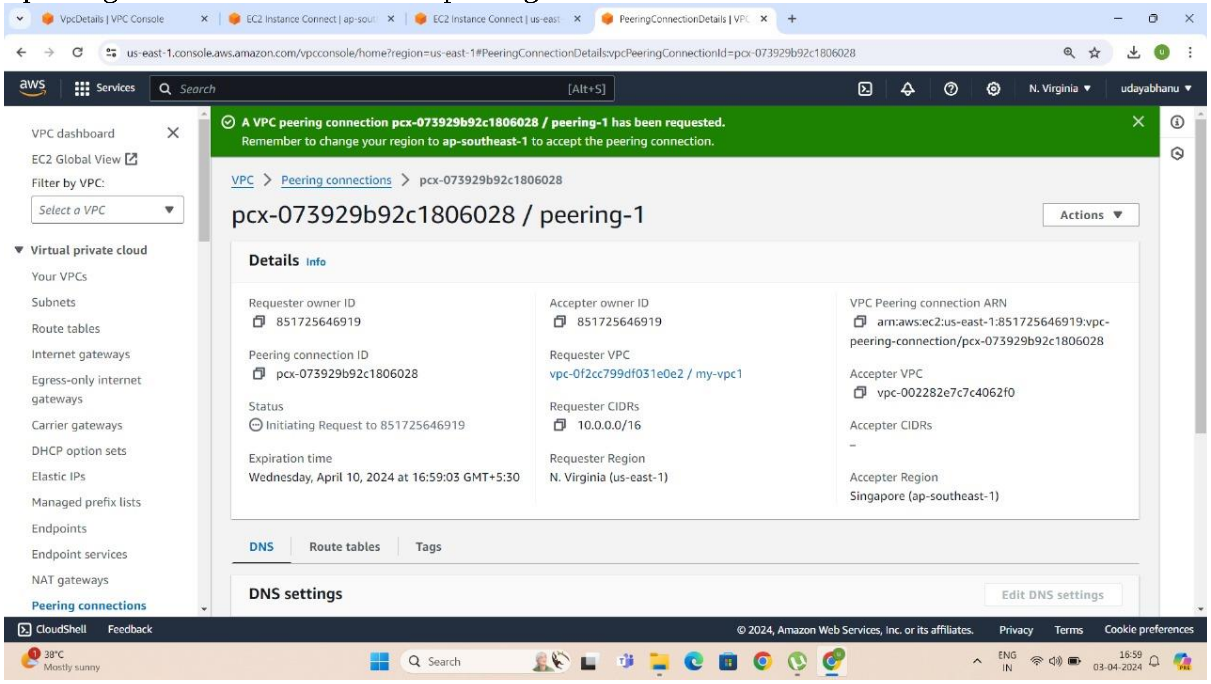
VPC ID (Requester)
vpc-0f2cc799df031e0e2 (my-vpc1)

VPC CIDRs for vpc-0f2cc799df031e0e2 (my-vpc1)

CIDR	Status	Status reason
10.0.0.0/16	✓ Associated	-

Select another VPC to peer with

Step19: now copy the vpc id of the accepter [another region vpc] and past it in the virgina peering creation field and create peering

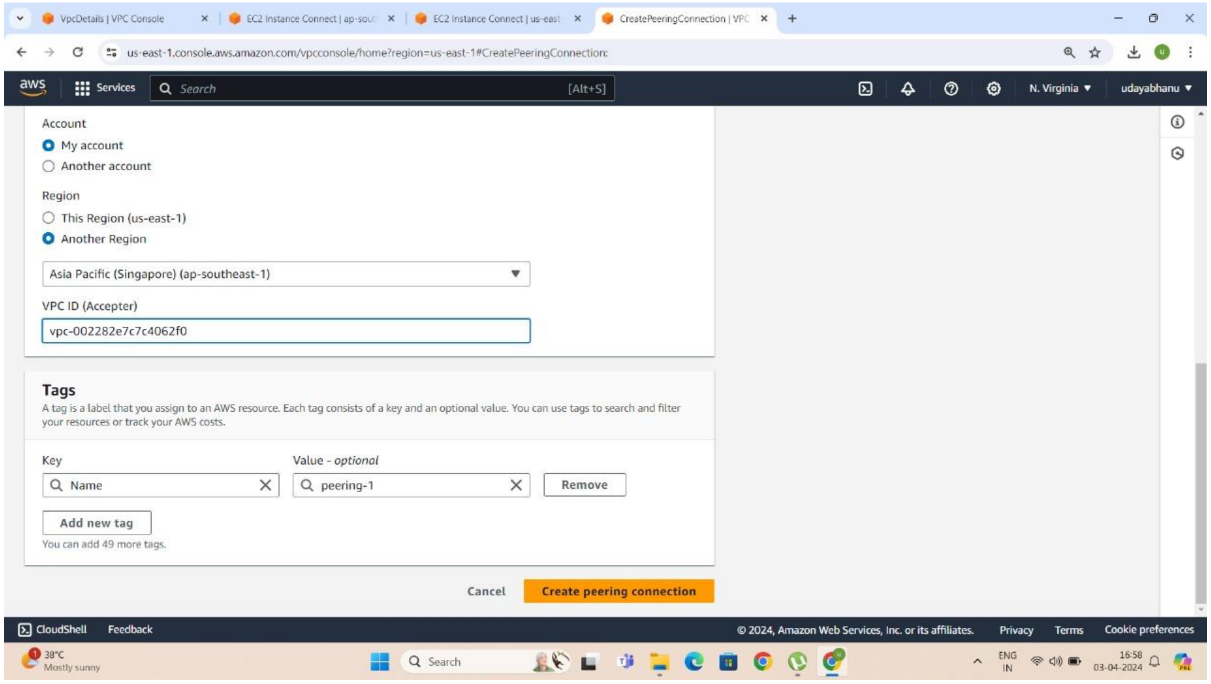

A VPC peering connection pcx-073929b92c1806028 / peering-1 has been requested.
Remember to change your region to ap-southeast-1 to accept the peering connection.

VPC > Peering connections > pcx-073929b92c1806028

pcx-073929b92c1806028 / peering-1

Details

Requester owner ID: 851725646919	Accepter owner ID: 851725646919	VPC Peering connection ARN: arn:aws:ec2:us-east-1:851725646919:vpc-peering-connection/pcx-073929b92c1806028
Peering connection ID: pcx-073929b92c1806028	Requester VPC: vpc-0f2cc799df031e0e2 / my-vpc1	Accepter VPC: vpc-002282e7c7c4062f0
Status: Initiating Request to 851725646919	Requester CIDRs: 10.0.0.0/16	Accepter CIDRs: –
Expiration time: Wednesday, April 10, 2024 at 16:59:03 GMT+5:30	Requester Region: N. Virginia (us-east-1)	Accepter Region: Singapore (ap-southeast-1)


Account
My account [✓]
Another account

Region
This Region (us-east-1)
Another Region [✓]

Asia Pacific (Singapore) (ap-southeast-1)

VPC ID (Accepter)
vpc-002282e7c7c4062f0

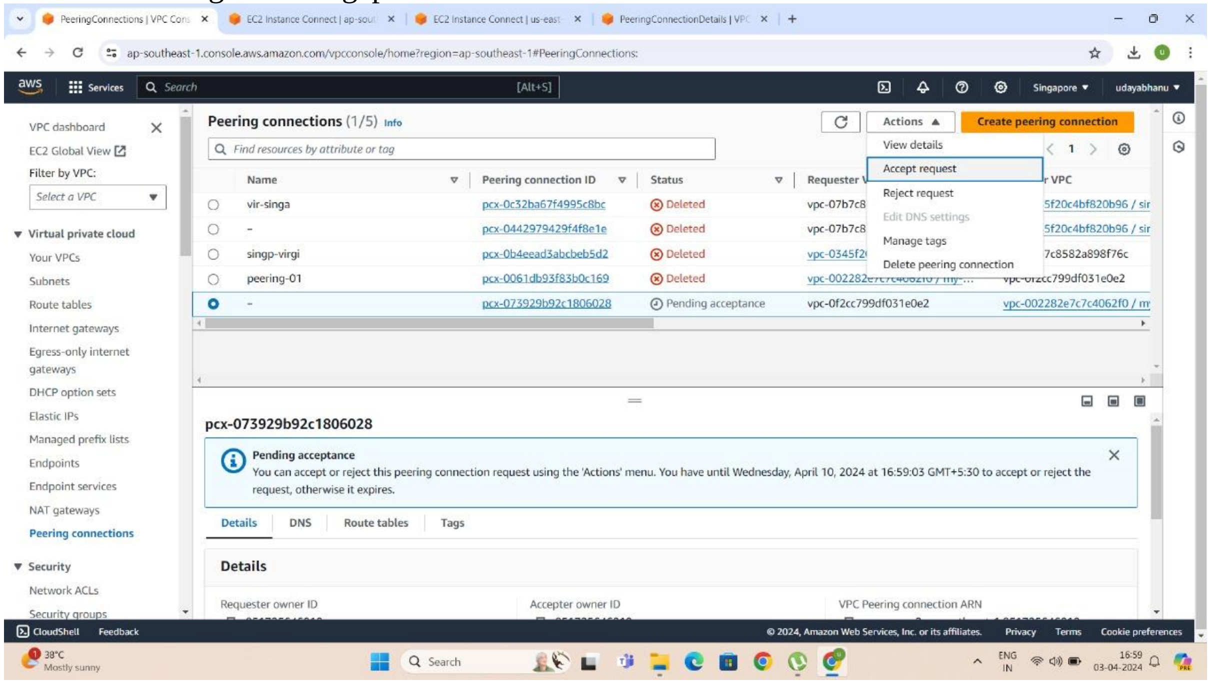
Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
Name	peering-1

You can add 49 more tags.

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreatePeeringConnection:
Remove
Add new tag
Cancel
Create peering connection

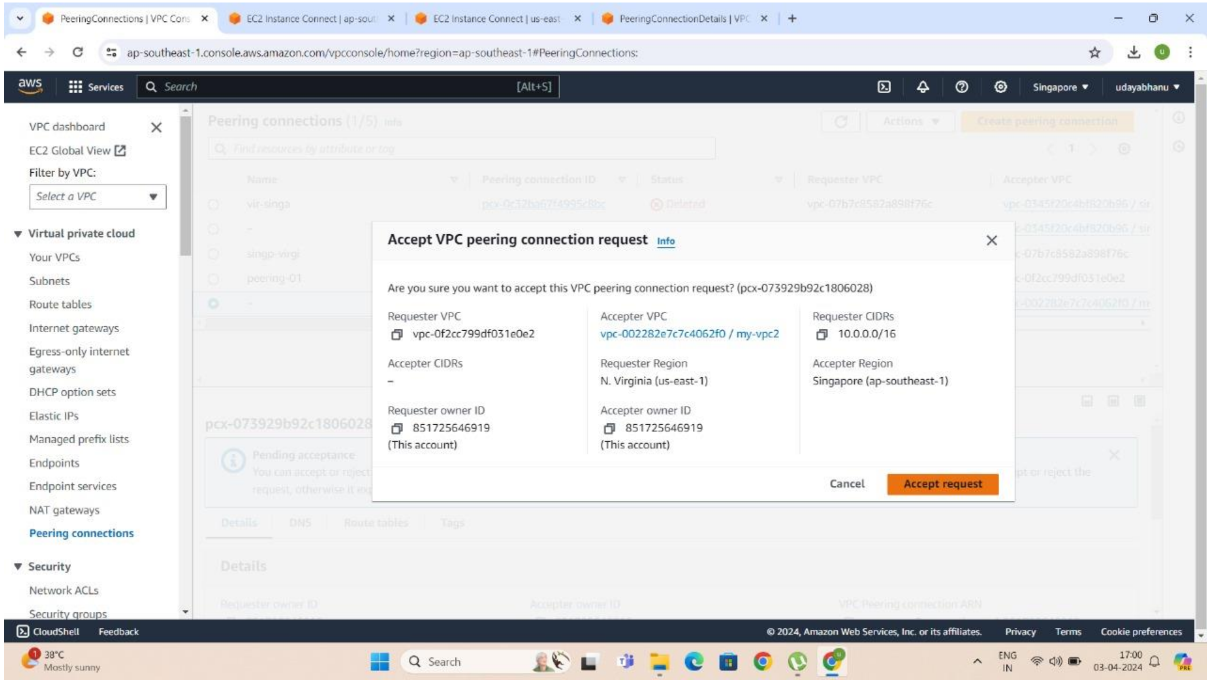
Step20: accept the peering request in the Singapore region and by clicking the edit route we link the virgina to singapore



	Name	Peering connection ID	Status	Requester VPC
○	vir-singa	pcx-0c32ba67f4995c8bc	Deleted	vpc-07b7c8
○	-	pcx-0442979429f4f8e1e	Deleted	vpc-07b7c8
○	singp-virgi	pcx-0b4eead3abcbeb5d2	Deleted	vpc-0345f2
○	peering-01	pcx-0061db93f83b0c169	Deleted	vpc-002282e7c7c4062f0 / my-...
●	-	pcx-073929b92c1806028	Pending acceptance	vpc-0f2cc799df031e0e2

pcx-073929b92c1806028

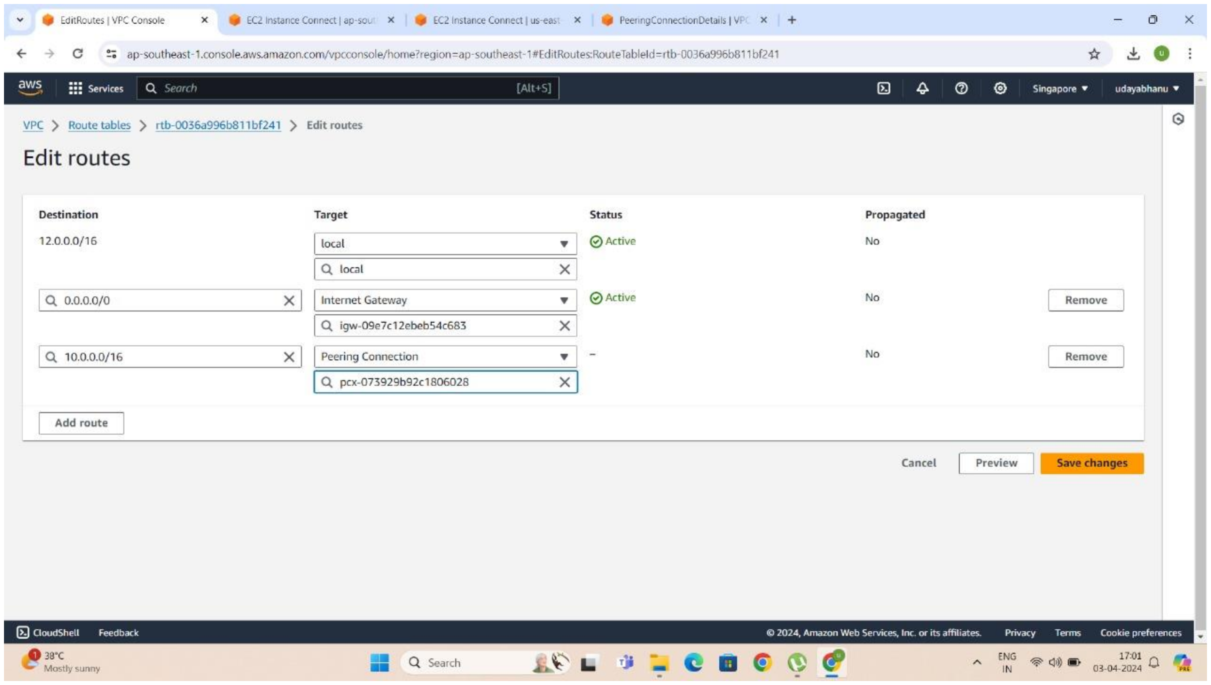
Pending acceptance
You can accept or reject this peering connection request using the 'Actions' menu. You have until Wednesday, April 10, 2024 at 16:59:03 GMT+5:30 to accept or reject the request, otherwise it expires.



Accept VPC peering connection request

Are you sure you want to accept this VPC peering connection request? (pcx-073929b92c1806028)

Requester VPC: vpc-0f2cc799df031e0e2
Accepter VPC: vpc-002282e7c7c4062f0 / my-vpc2
Requester CIDRs: 10.0.0.0/16
Accepter CIDRs: –
Requester Region: N. Virginia (us-east-1)
Accepter Region: Singapore (ap-southeast-1)
Requester owner ID: 851725646919 (This account)
Accepter owner ID: 851725646919 (This account)

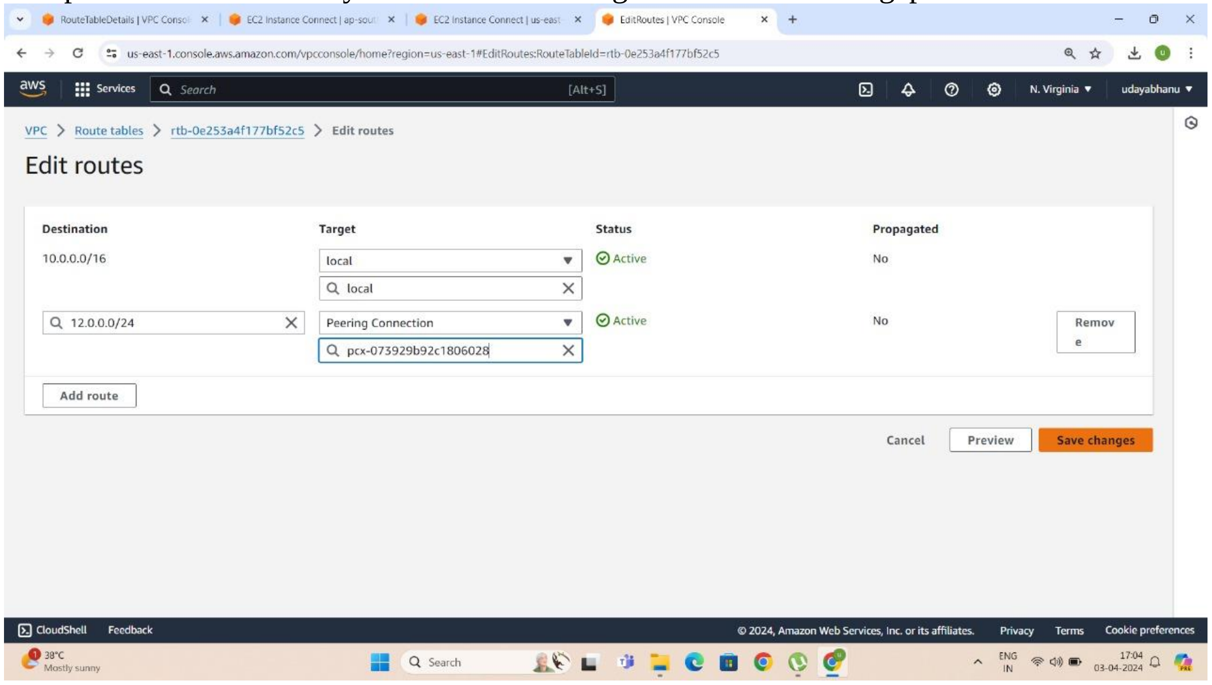

EditRoutes | VPC Console
EC2 Instance Connect | ap-sout
EC2 Instance Connect | us-east
PeeringConnectionDetails | VPC
ap-southeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-southeast-1#EditRoutes:RouteTableId=rtb-0036a996b811bf241
Services
Search
[Alt+S]
Singapore
udayabhanu
VPC > Route tables > rtb-0036a996b811bf241 > Edit routes

Edit routes

Destination	Target	Status	Propagated	
12.0.0.0/16	local / local	Active	No	
0.0.0.0/0	Internet Gateway / igw-09e7c12ebeb54c683	Active	No	Remove
10.0.0.0/16	Peering Connection / pcx-073929b92c1806028	-	No	Remove

Add route
Cancel
Preview
Save changes
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
38°C
Mostly sunny
17:01
03-04-2024

Step21: in the similar way edit the route in virgina and connect Singapore

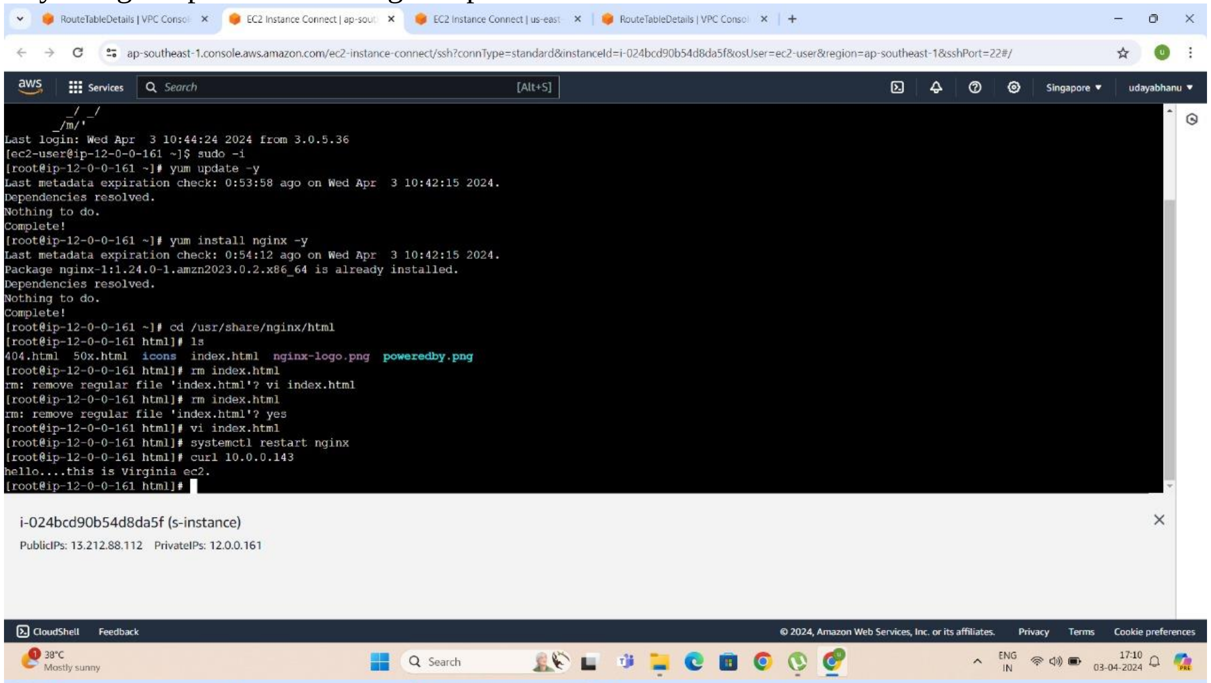

RouteTableDetails | VPC Consol
EC2 Instance Connect | ap-sout
EC2 Instance Connect | us-east
EditRoutes | VPC Console
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#EditRoutes:RouteTableId=rtb-0e253a4f177bf52c5
Services
Search
[Alt+S]
N. Virginia
udayabhanu
VPC > Route tables > rtb-0e253a4f177bf52c5 > Edit routes

Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/16	local / local	Active	No	
12.0.0.0/24	Peering Connection / pcx-073929b92c1806028	Active	No	Remove

Add route
Cancel
Preview
Save changes
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
38°C
Mostly sunny
17:04
03-04-2024

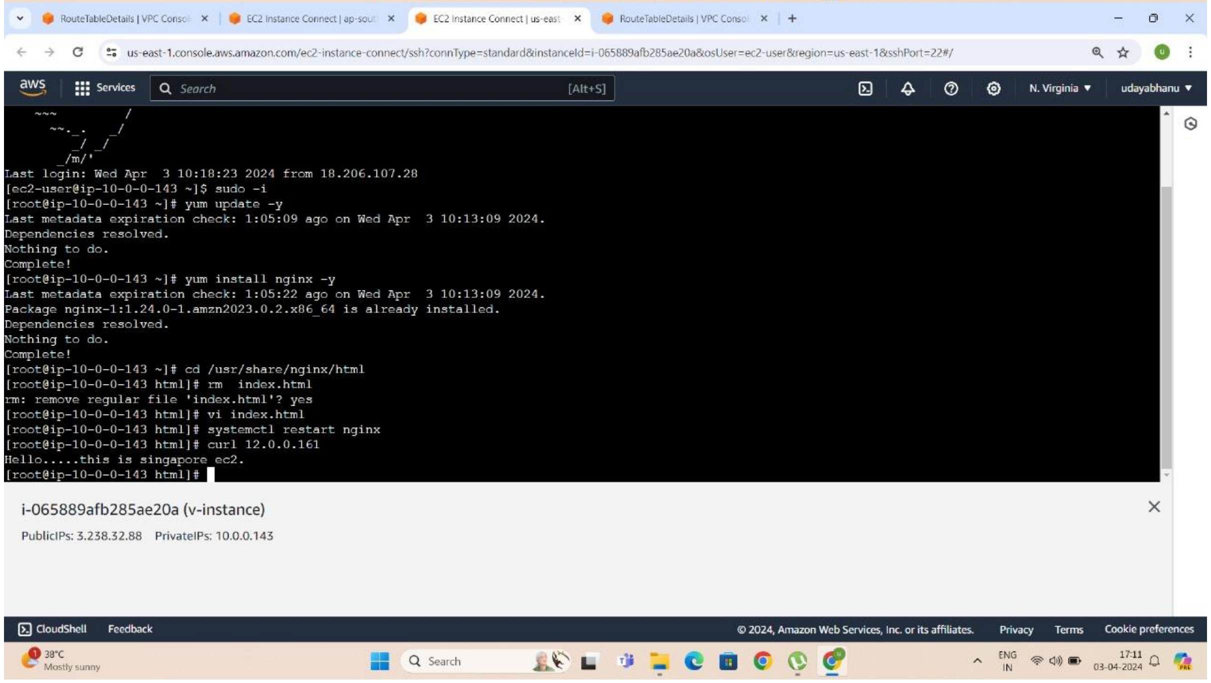
Step22: them connect to the EC2 instance in Singapore and access virgina vpc through it by using the private id of virgina vpc



```
        _/ _/
       _/m/'
Last login: Wed Apr  3 10:44:24 2024 from 3.0.5.36
[ec2-user@ip-12-0-0-161 ~]$ sudo -i
[root@ip-12-0-0-161 ~]# yum update -y
Last metadata expiration check: 0:53:58 ago on Wed Apr  3 10:42:15 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-12-0-0-161 ~]# yum install nginx -y
Last metadata expiration check: 0:54:12 ago on Wed Apr  3 10:42:15 2024.
Package nginx-1:1.24.0-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-12-0-0-161 ~]# cd /usr/share/nginx/html
[root@ip-12-0-0-161 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-12-0-0-161 html]# rm index.html
rm: remove regular file 'index.html'? vi index.html
[root@ip-12-0-0-161 html]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-12-0-0-161 html]# vi index.html
[root@ip-12-0-0-161 html]# systemctl restart nginx
[root@ip-12-0-0-161 html]# curl 10.0.0.143
hello....this is Virginia ec2.
[root@ip-12-0-0-161 html]#
```

i-024bcd90b54d8da5f (s-instance)

PublicIPs: 13.212.88.112 PrivateIPs: 12.0.0.161



```
   ~~~          /
     ~~._.    _/
        _/ _/
       _/m/'
Last login: Wed Apr  3 10:18:23 2024 from 18.206.107.28
[ec2-user@ip-10-0-0-143 ~]$ sudo -i
[root@ip-10-0-0-143 ~]# yum update -y
Last metadata expiration check: 1:05:09 ago on Wed Apr  3 10:13:09 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-0-143 ~]# yum install nginx -y
Last metadata expiration check: 1:05:22 ago on Wed Apr  3 10:13:09 2024.
Package nginx-1:1.24.0-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-0-143 ~]# cd /usr/share/nginx/html
[root@ip-10-0-0-143 html]# rm  index.html
rm: remove regular file 'index.html'? yes
[root@ip-10-0-0-143 html]# vi index.html
[root@ip-10-0-0-143 html]# systemctl restart nginx
[root@ip-10-0-0-143 html]# curl 12.0.0.161
Hello.....this is singapore ec2.
[root@ip-10-0-0-143 html]#
```

i-065889afb285ae20a (v-instance)

PublicIPs: 3.238.32.88 PrivateIPs: 10.0.0.143

Conclusion: Here we successfully connect two vpc's in different region through vpc peering and access the content in them.