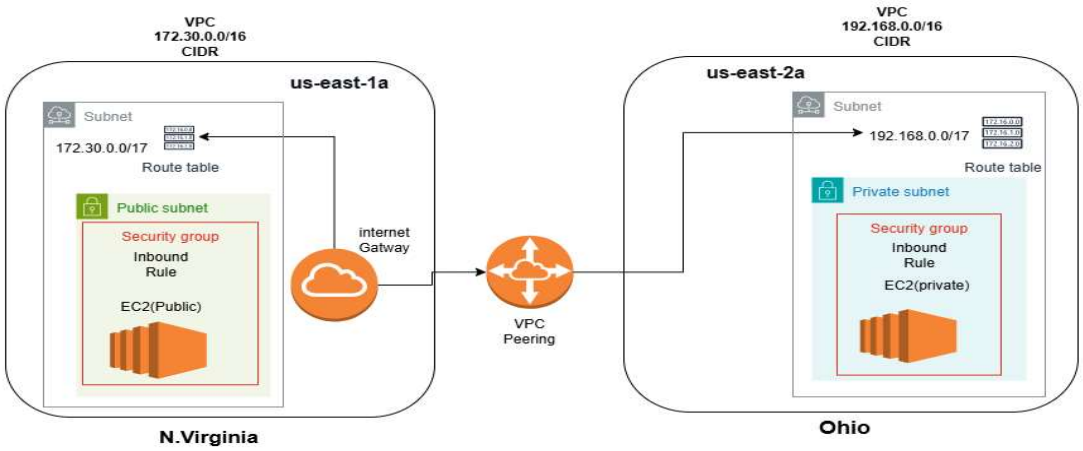
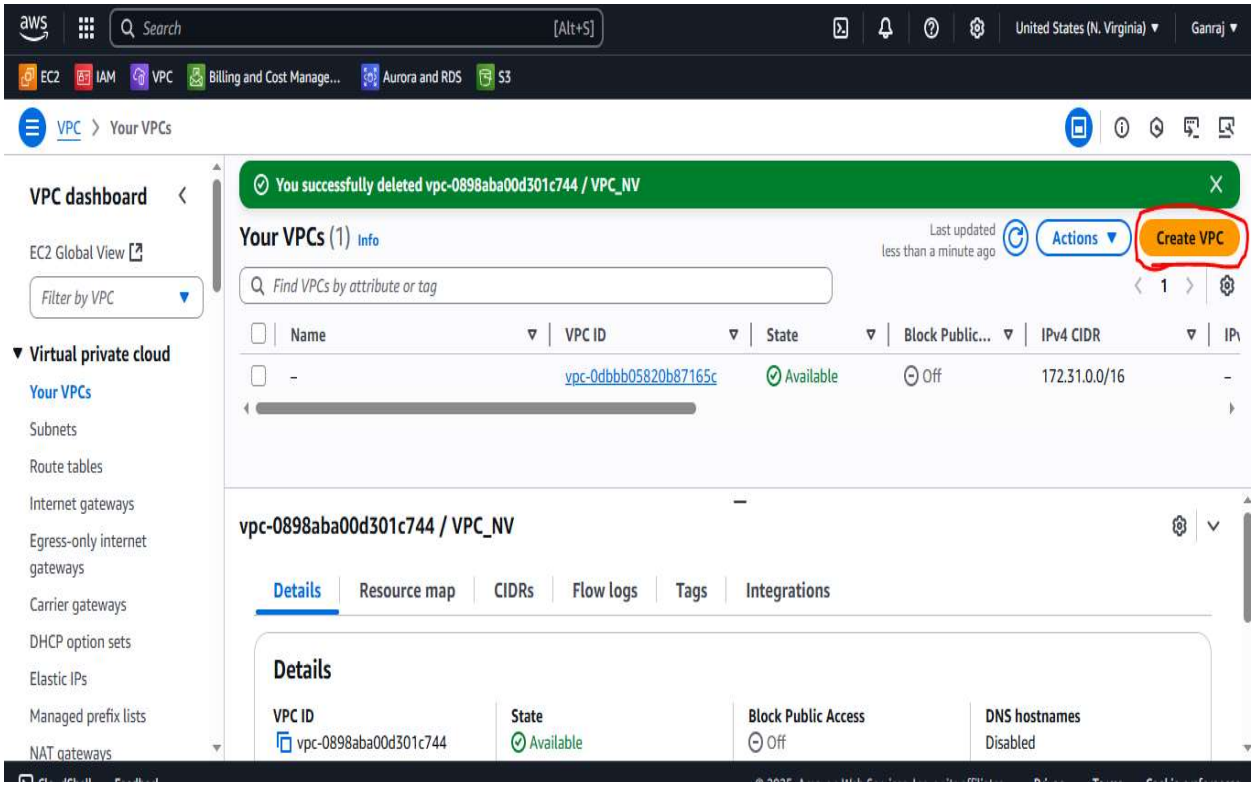
VPC Peering :-

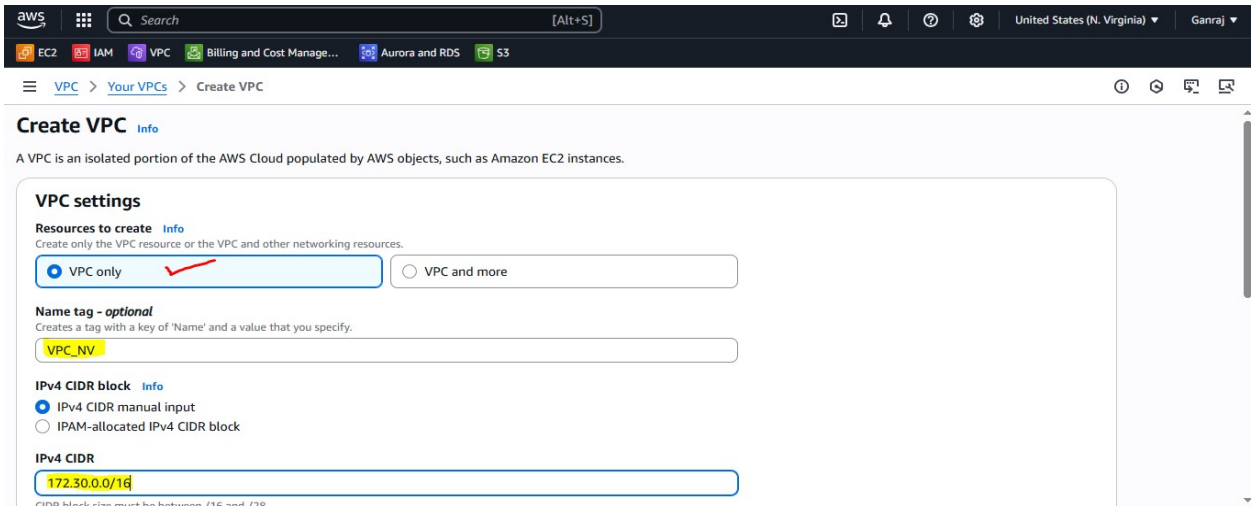
Connecting two VPCs with each other explaining concept of public server (ec2) internet giving to private server(ec2).

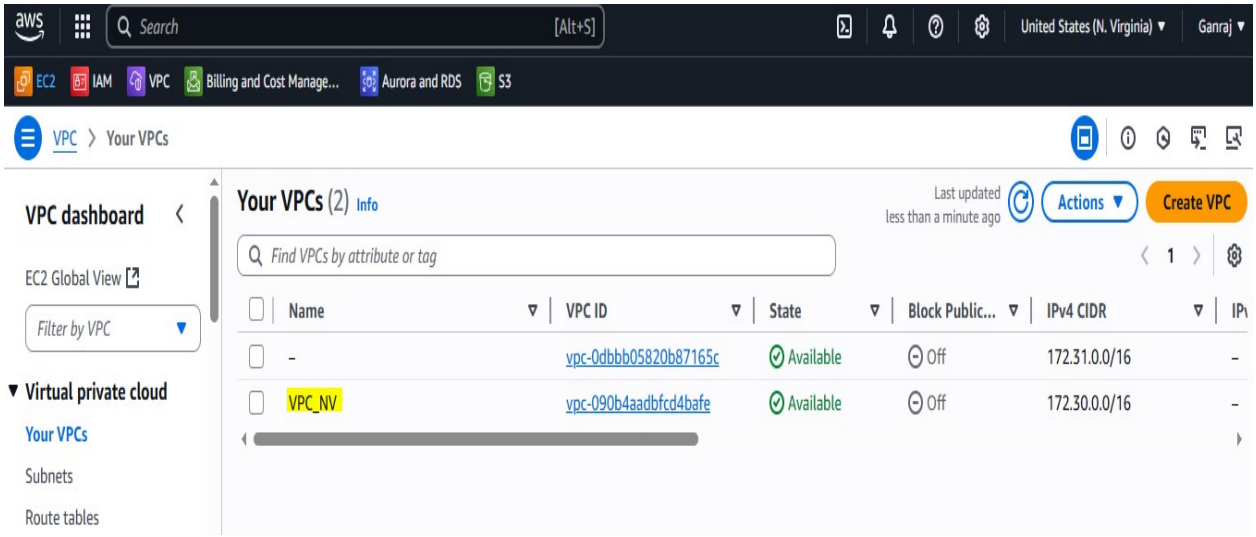


Note : VPC peering need to create in that VPC where the Internet gateway created.

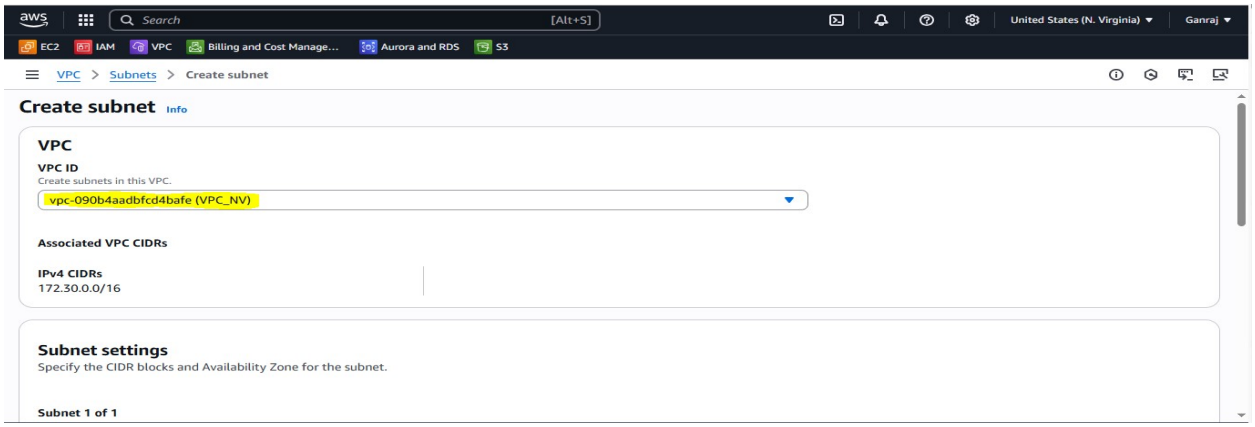
First we will create a VPC in N.viriginia

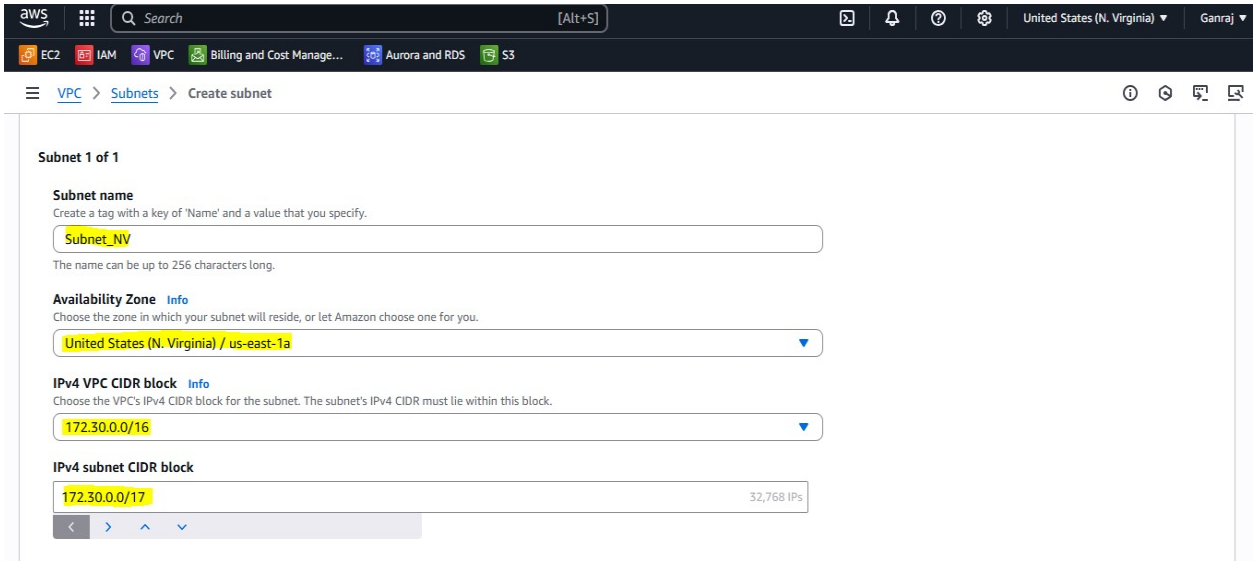




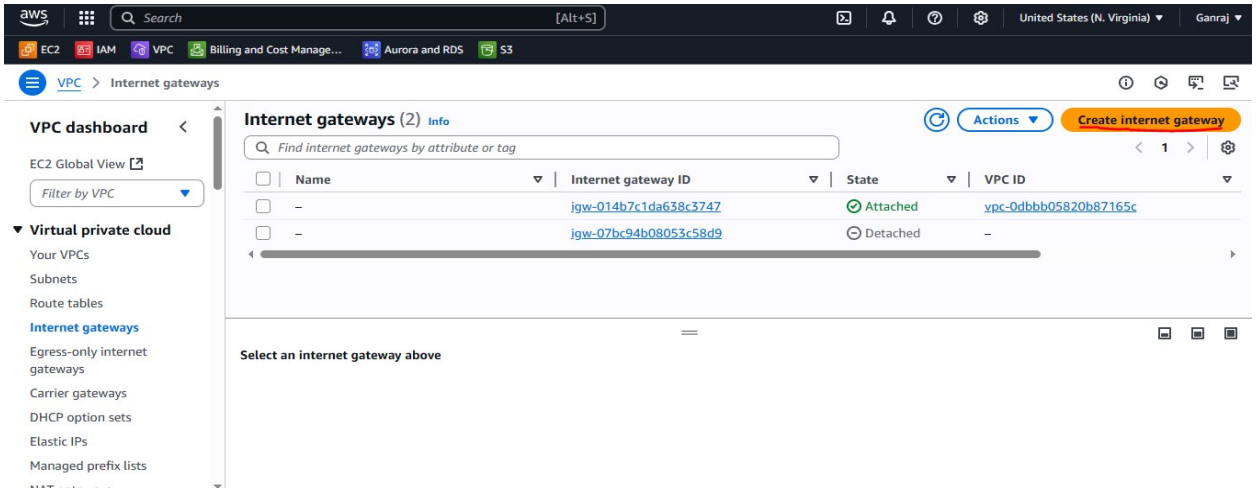


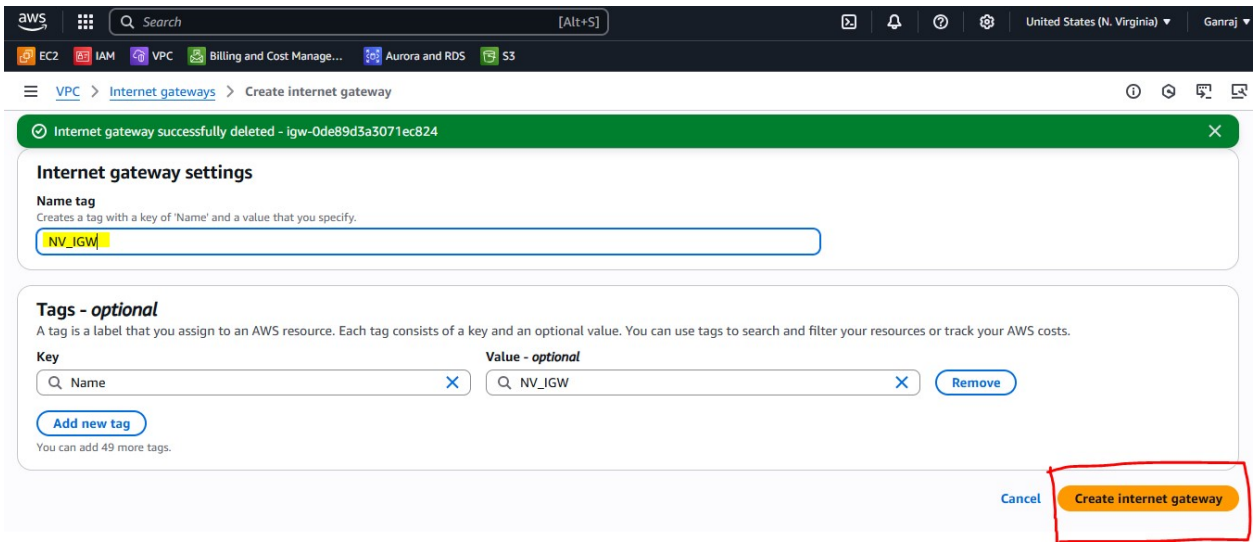
Creating Subnet in same VPC.



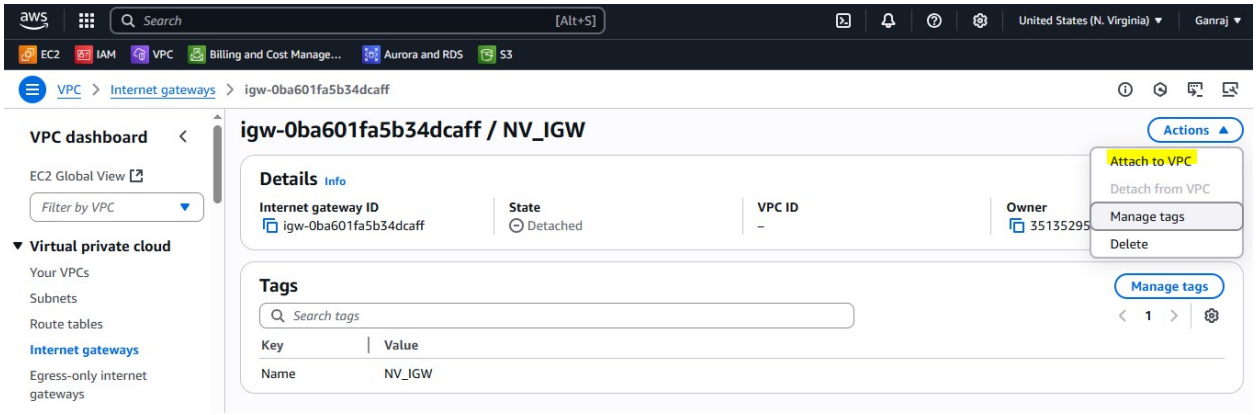


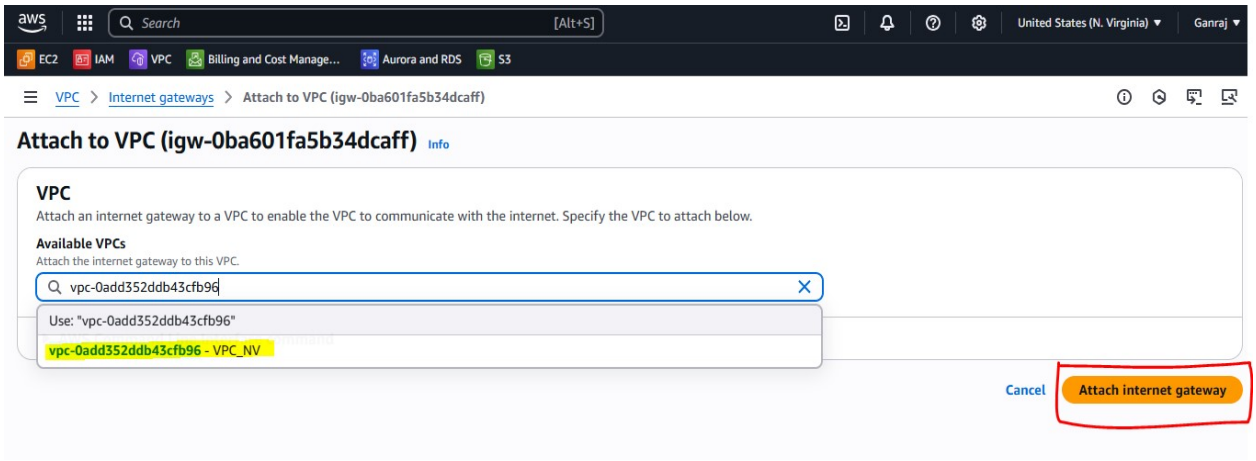
Creating Internet Gateway with Same VPC.



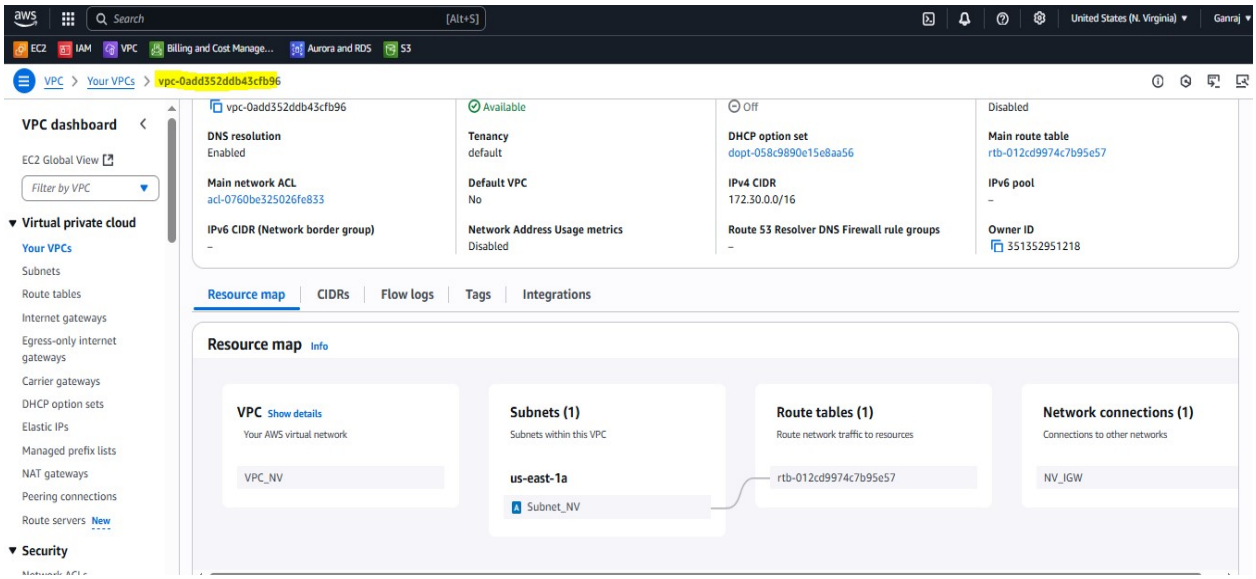


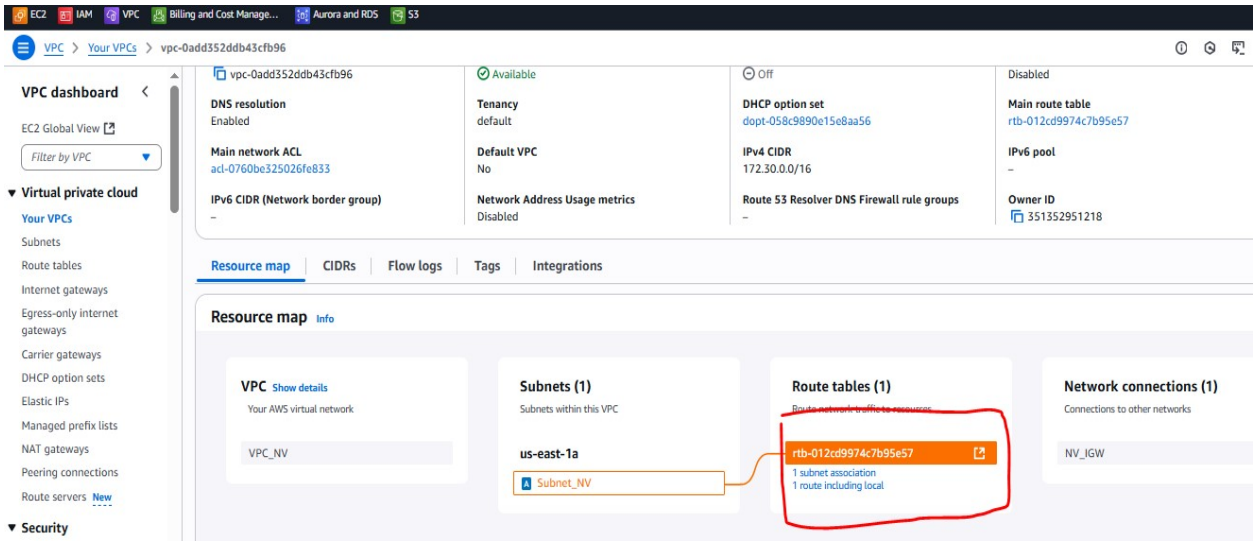
Need to attach to VPC that we have created one.

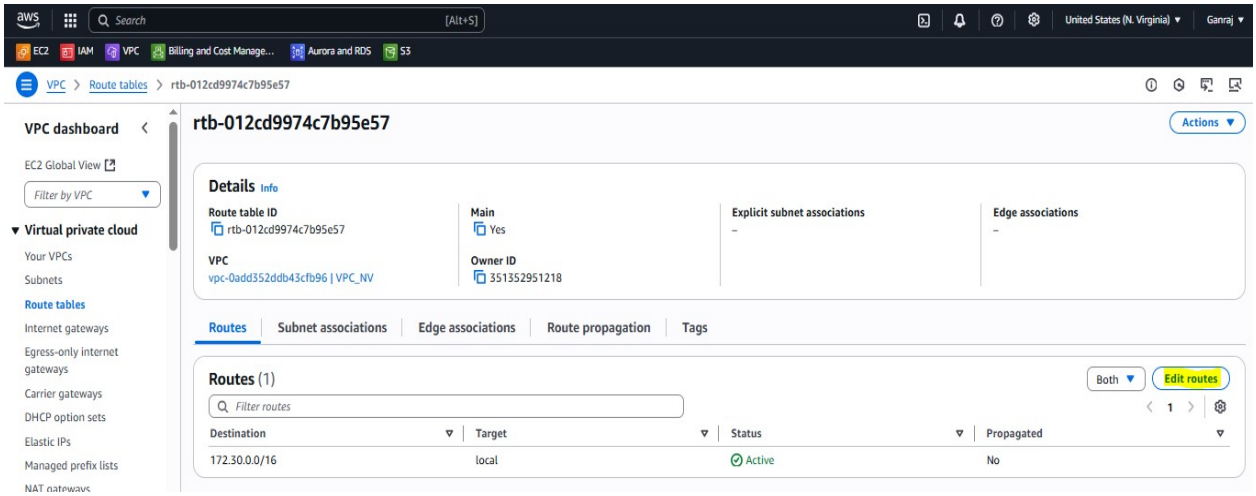


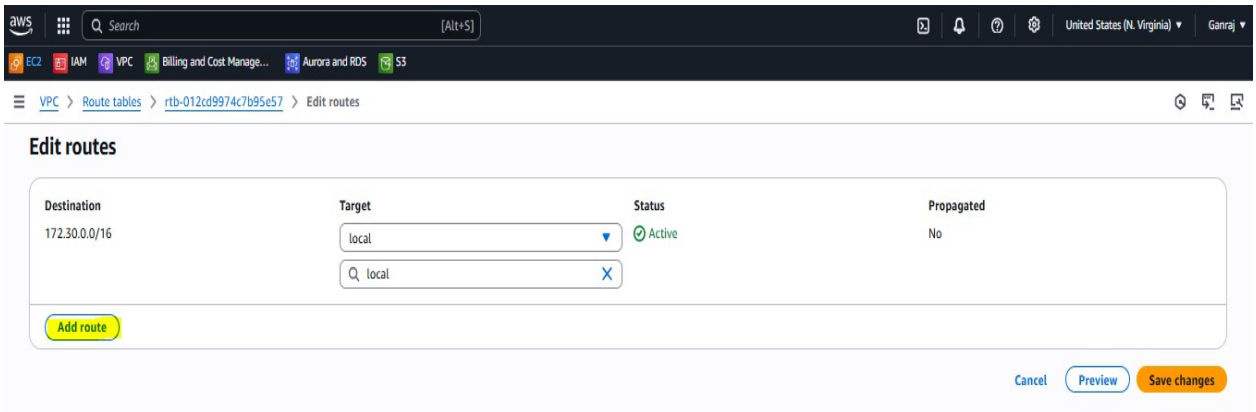


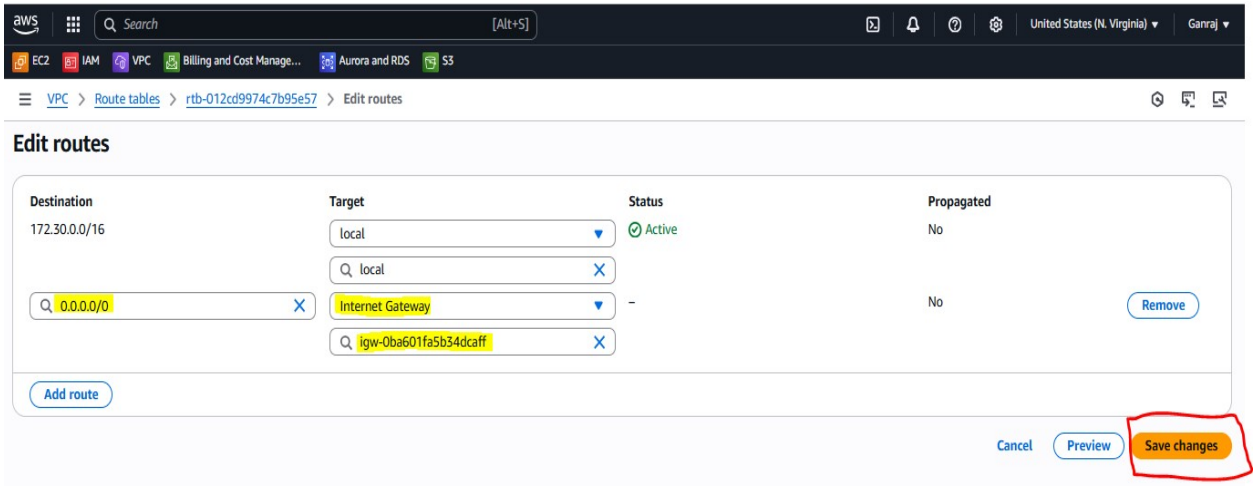
Need to connect route table to internet gateway is very important.

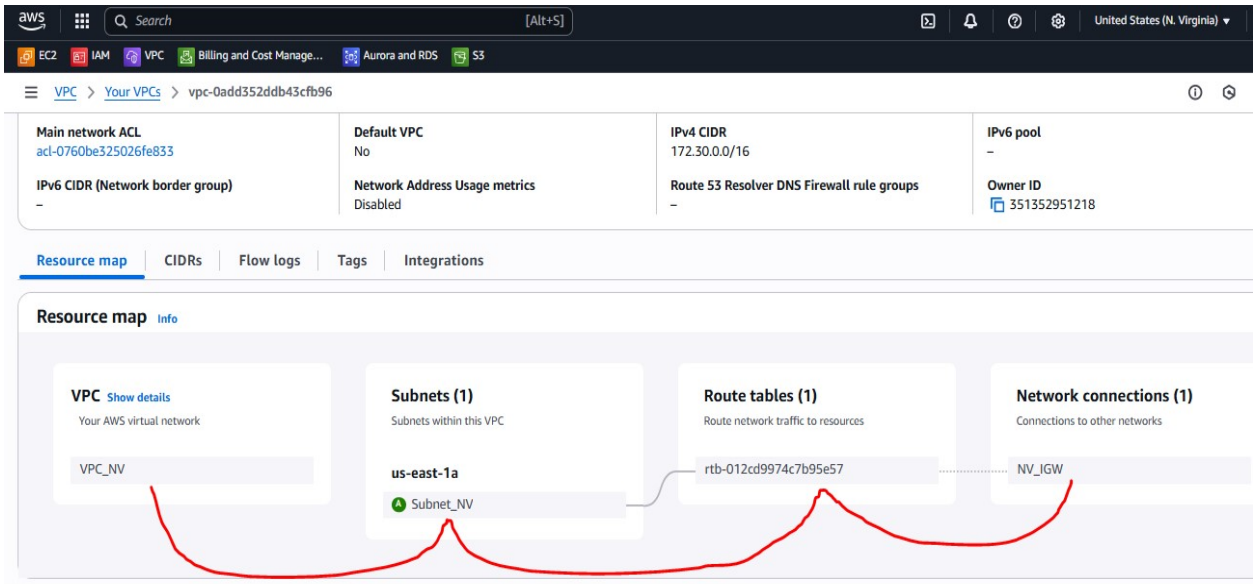



VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers New
Security
VPC > Your VPCs > vpc-0add352ddb43cfb96
vpc-0add352ddb43cfb96
Available
Off
Disabled
DNS resolution
Enabled
Tenancy
default
DHCP option set
dopt-058c9890e15e8aa56
Main route table
rtb-012cd9974c7b95e57
Main network ACL
acl-0760be325026fe833
Default VPC
No
IPv4 CIDR
172.30.0.0/16
IPv6 pool
IPv6 CIDR (Network border group)
Network Address Usage metrics
Disabled
Route 53 Resolver DNS Firewall rule groups
Owner ID
351352951218
Resource map
CIDRs
Flow logs
Tags
Integrations
Resource map Info
VPC Show details
Your AWS virtual network
VPC_NV
Subnets (1)
Subnets within this VPC
us-east-1a
Subnet_NV
Route tables (1)
rtb-012cd9974c7b95e57
1 subnet association
1 route including local
Network connections (1)
Connections to other networks
NV_IGW


Search
[Alt+S]
United States (N. Virginia)
Ganraj
VPC > Route tables > rtb-012cd9974c7b95e57
VPC dashboard
rtb-012cd9974c7b95e57
Actions
Details Info
Route table ID
rtb-012cd9974c7b95e57
Main
Yes
Explicit subnet associations
Edge associations
VPC
vpc-0add352ddb43cfb96 | VPC_NV
Owner ID
351352951218
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (1)
Filter routes
Both
Edit routes
Destination
Target
Status
Propagated
172.30.0.0/16
local
Active
No

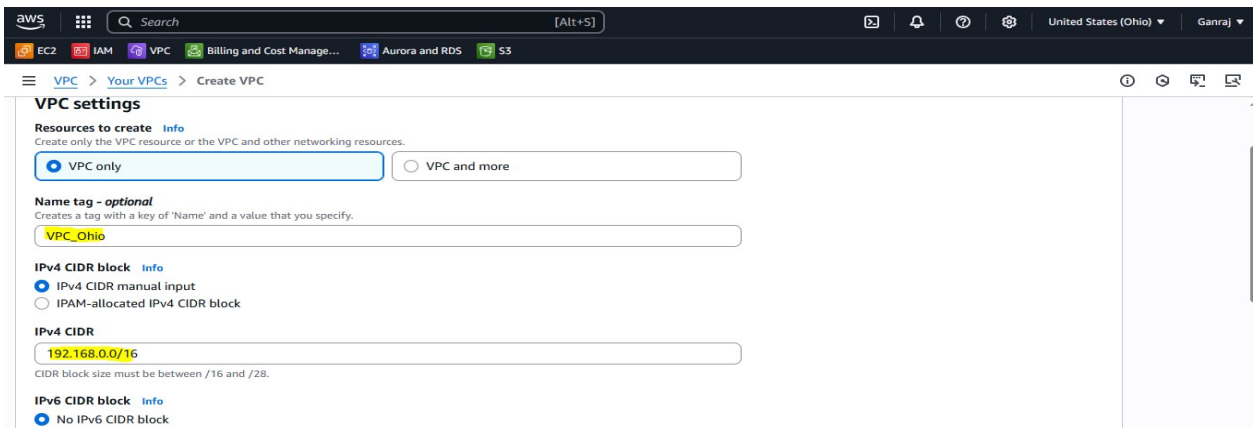

Search
[Alt+S]
United States (N. Virginia)
Ganraj
VPC > Route tables > rtb-012cd9974c7b95e57 > Edit routes
Edit routes
Destination
Target
Status
Propagated
172.30.0.0/16
local
Active
No
local
Add route
Cancel
Preview
Save changes

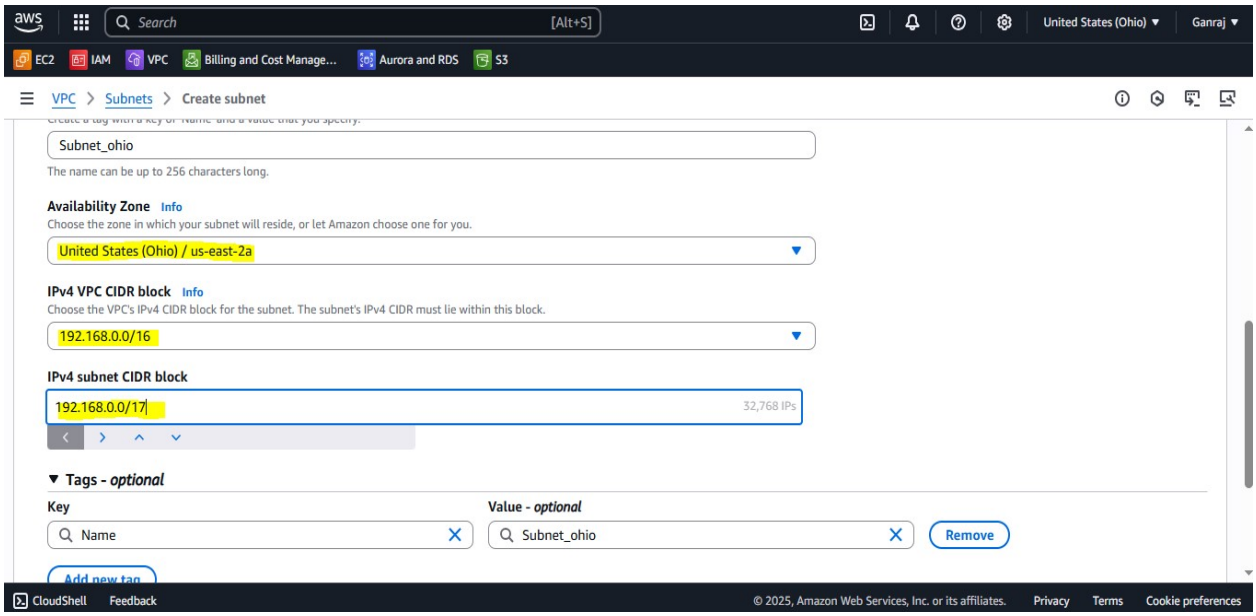


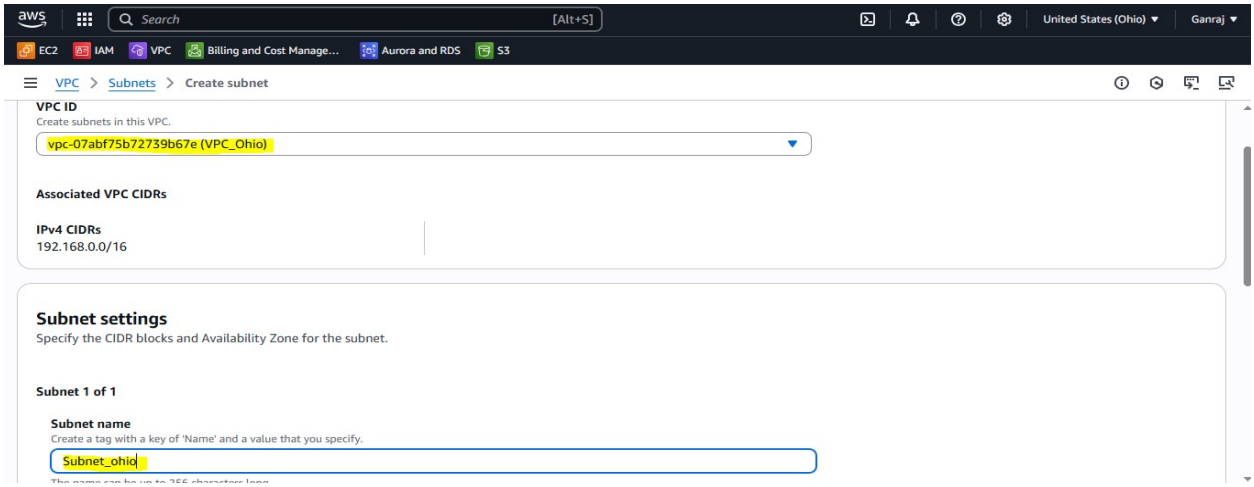


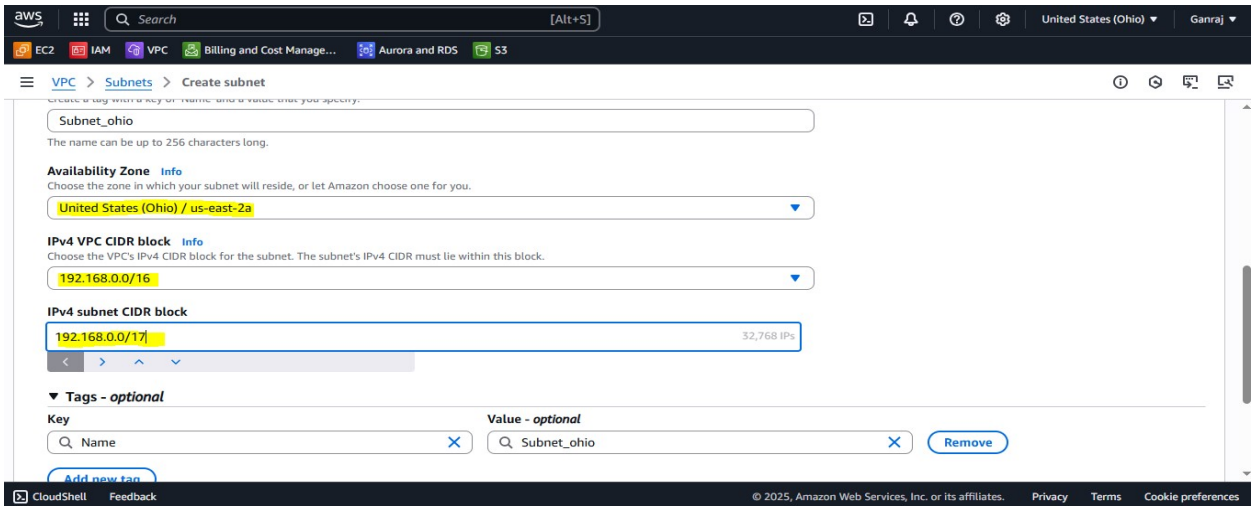
See we have made connection between internet gateway route tables and subnets.

Now we will create VPC Ohio Region .

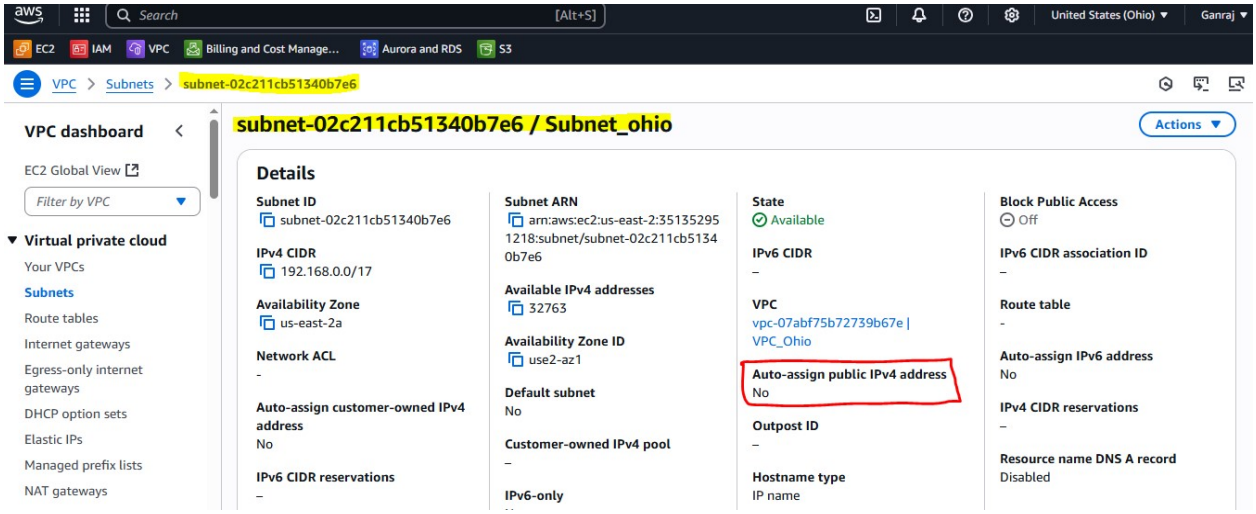



Search [Alt+S]
United States (Ohio)
Ganraj
EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3
VPC > Subnets > Create subnet
Subnet_ohio
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (Ohio) / us-east-2a
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
192.168.0.0/16
IPv4 subnet CIDR block
192.168.0.0/17
32,768 IPs
Tags - optional
Key
Value - optional
Name
Subnet_ohio
Remove
Add new tag
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


Search [Alt+S]
United States (Ohio)
Ganraj
EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3
VPC > Subnets > Create subnet
VPC ID
Create subnets in this VPC.
vpc-07abf75b72739b67e (VPC_Ohio)
Associated VPC CIDRs
IPv4 CIDRs
192.168.0.0/16
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.
Subnet_ohio


Search [Alt+S]
United States (Ohio)
Ganraj
EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3
VPC > Subnets > Create subnet
Subnet_ohio
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (Ohio) / us-east-2a
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
192.168.0.0/16
IPv4 subnet CIDR block
192.168.0.0/17
32,768 IPs
Tags - optional
Key
Value - optional
Name
Subnet_ohio
Remove
Add new tag
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

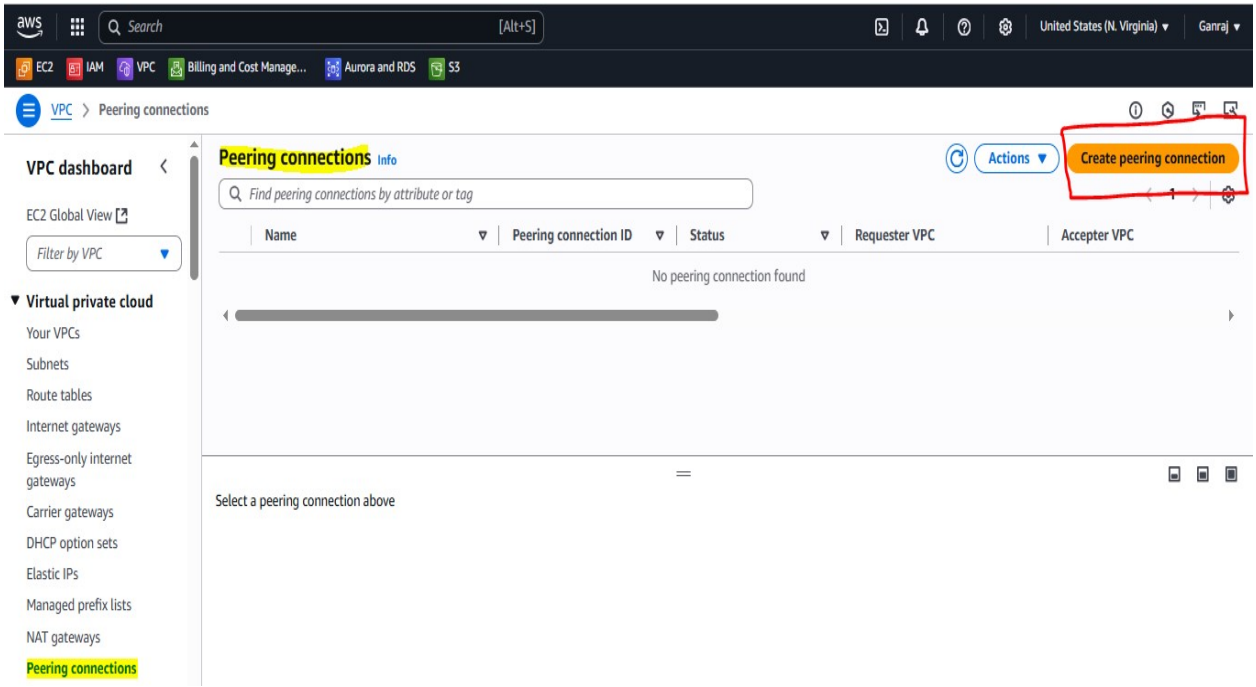
We have created subnet in ohio vpc but need to check that it not auto assign IP(public) because we creating a private network or the private subnet which dosen't have the public access IP

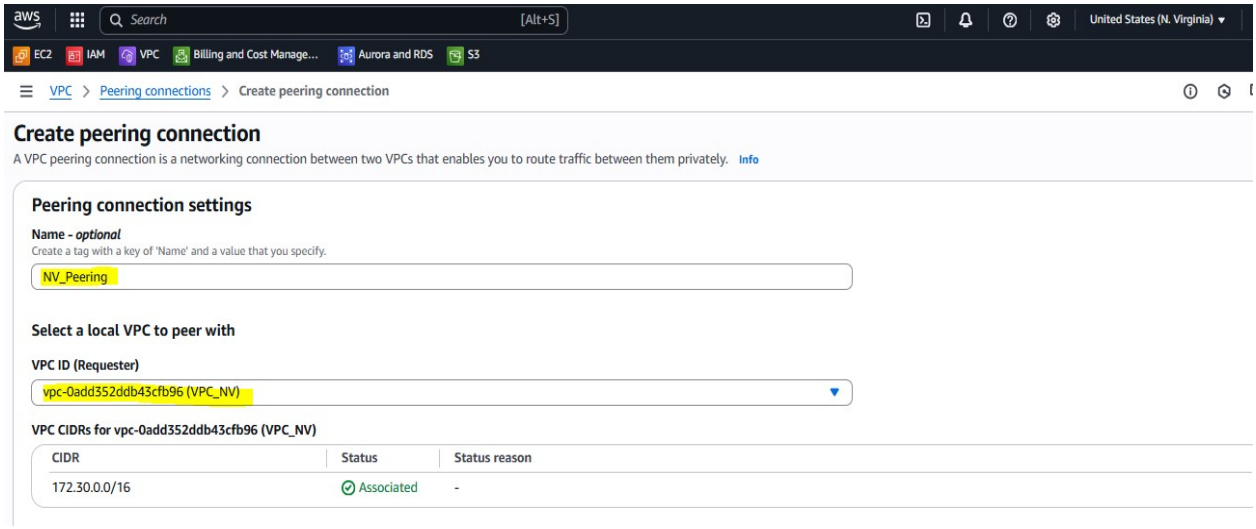


Now we will create VPC Peering : -

VPC peering created only where the internet gateway create VPC (N.virginia)

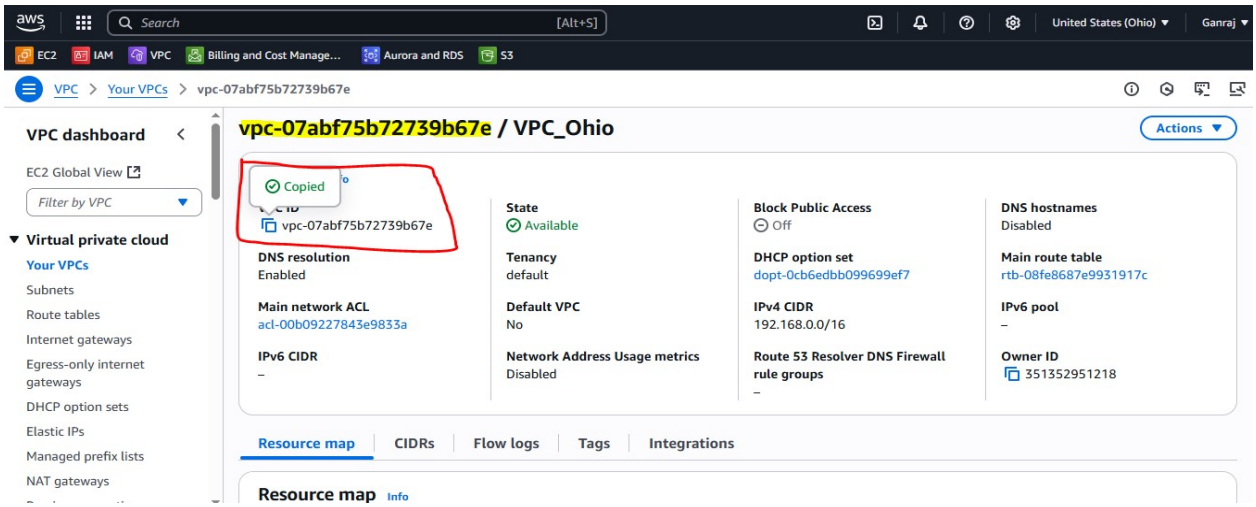
Note : We also connect VPC Peering CrossAccount also

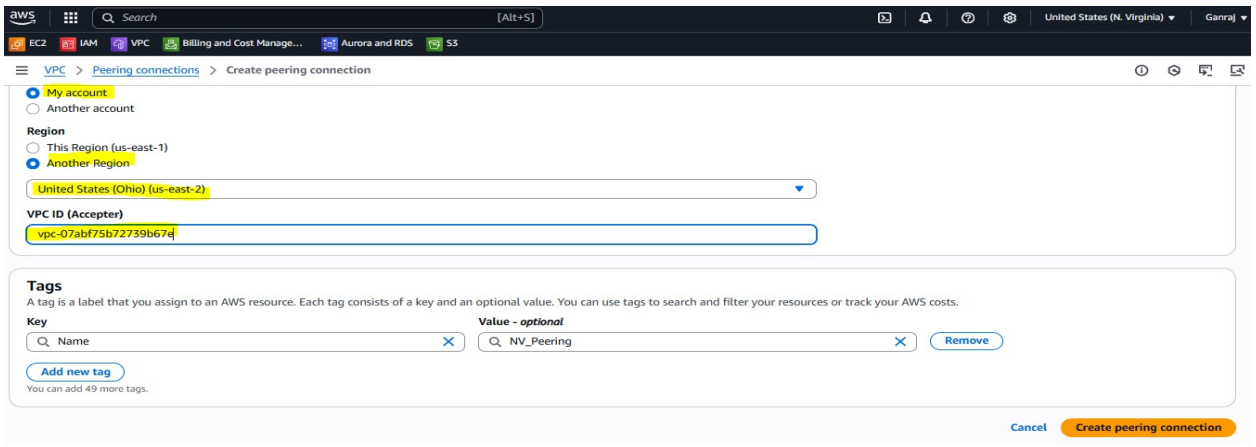


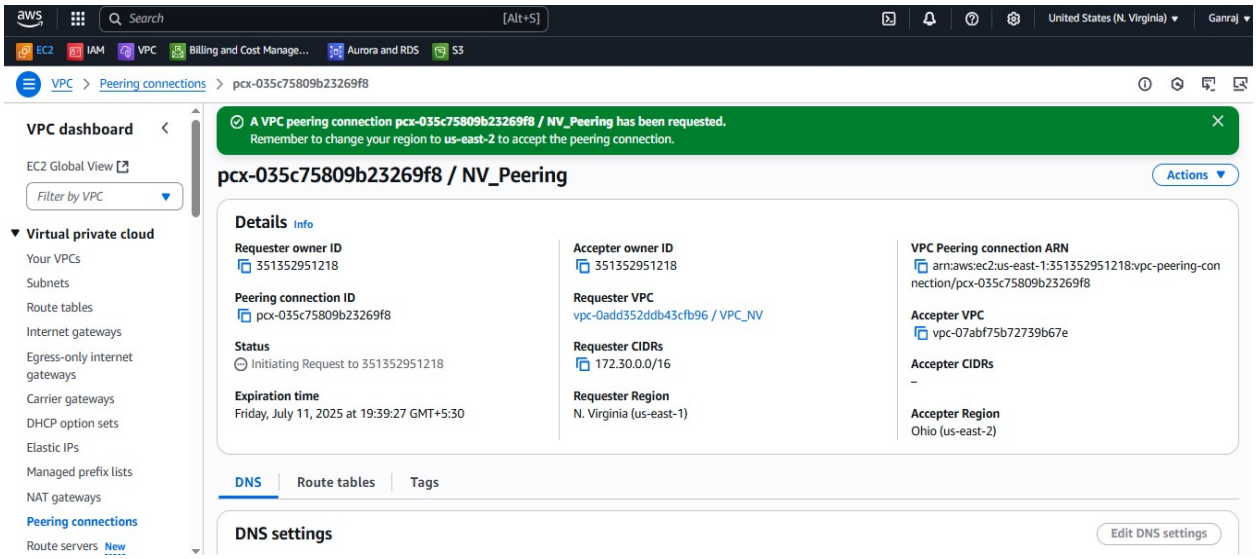


First go to Ohio Region to take VPC ID from VPC_ohio and paste it over VPC ID Acceptor

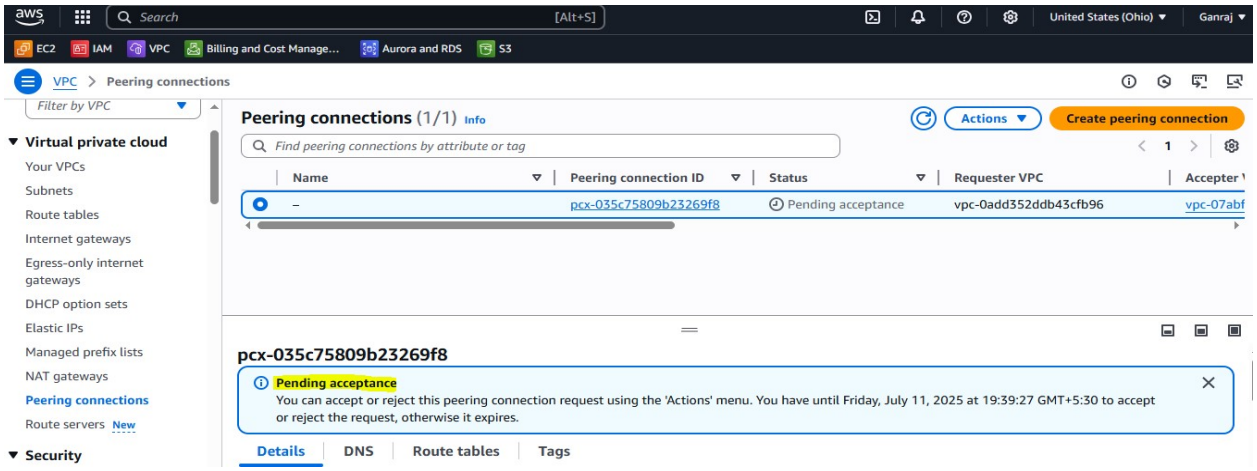
And create peering.

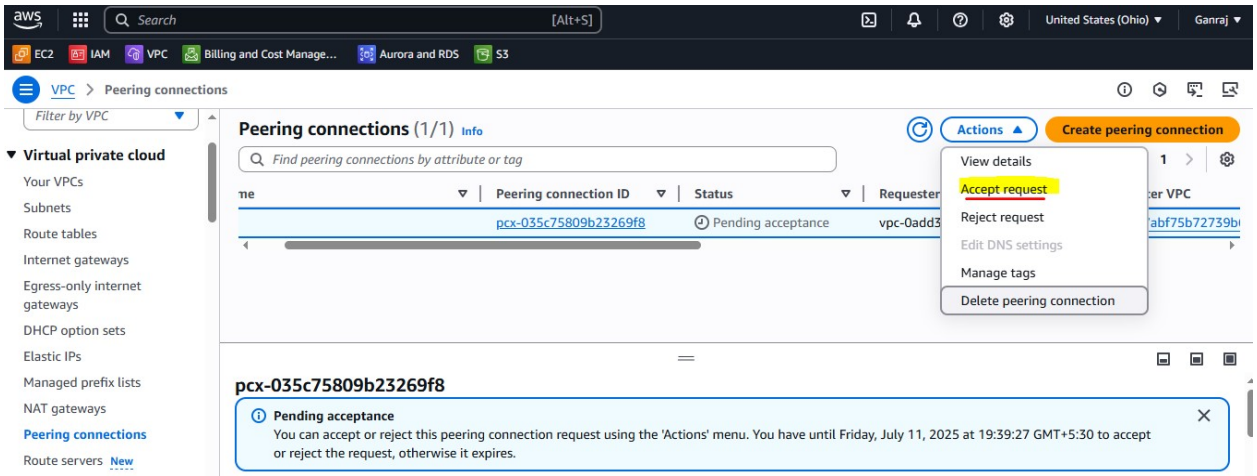


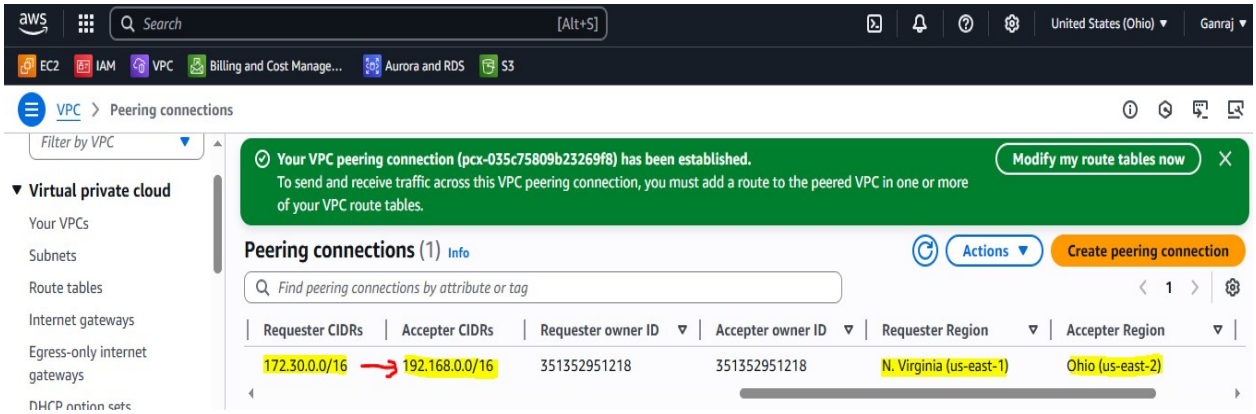


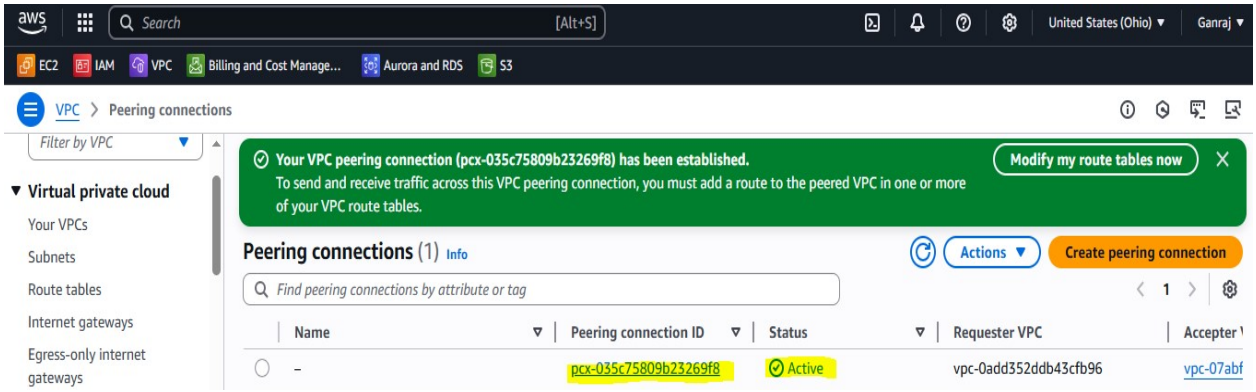


VPC Peering is created and N.Virigina VPC requested to Ohio VPC for peering Connection We need to accept that request from ohio region VPC Peering.





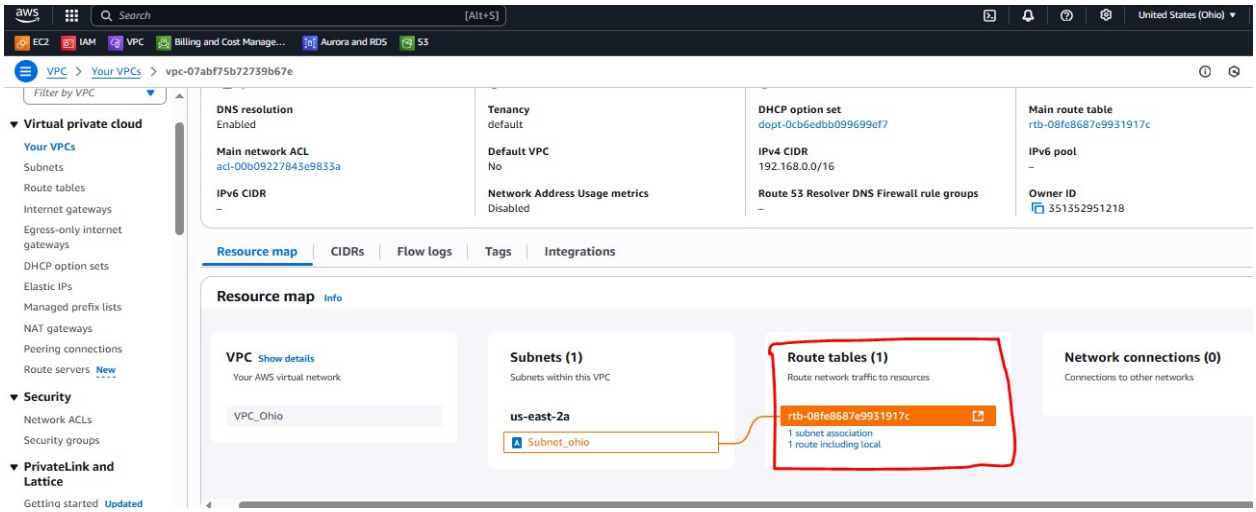


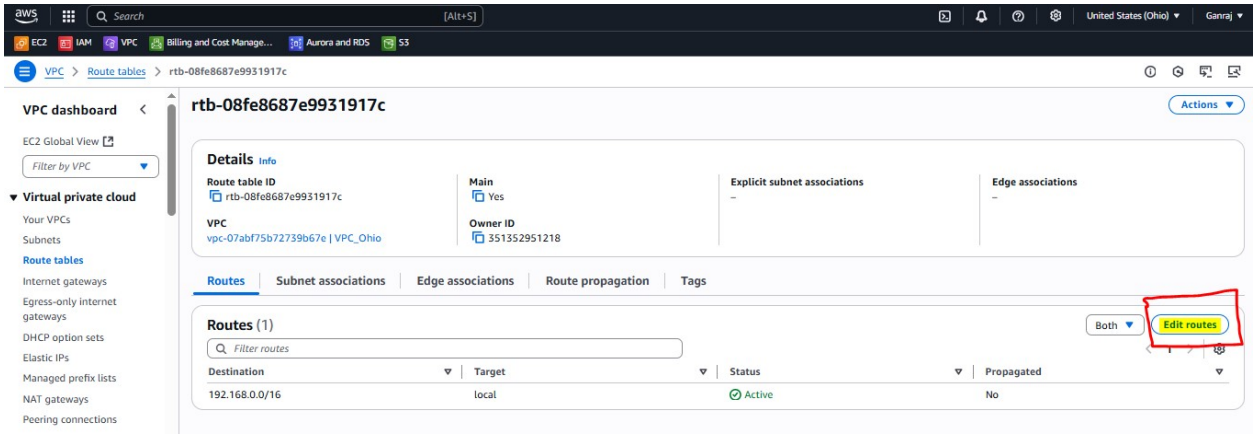


Note : we only create one VPC Peering always because its process of requester and acceptor connecting.

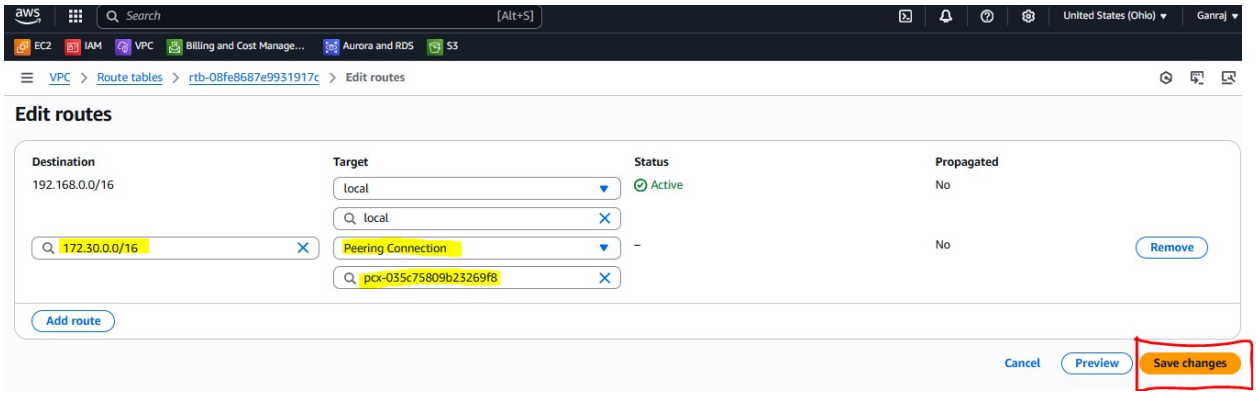
Now we have route tables need to connect each other

First we will go to Ohio private subnet route table.

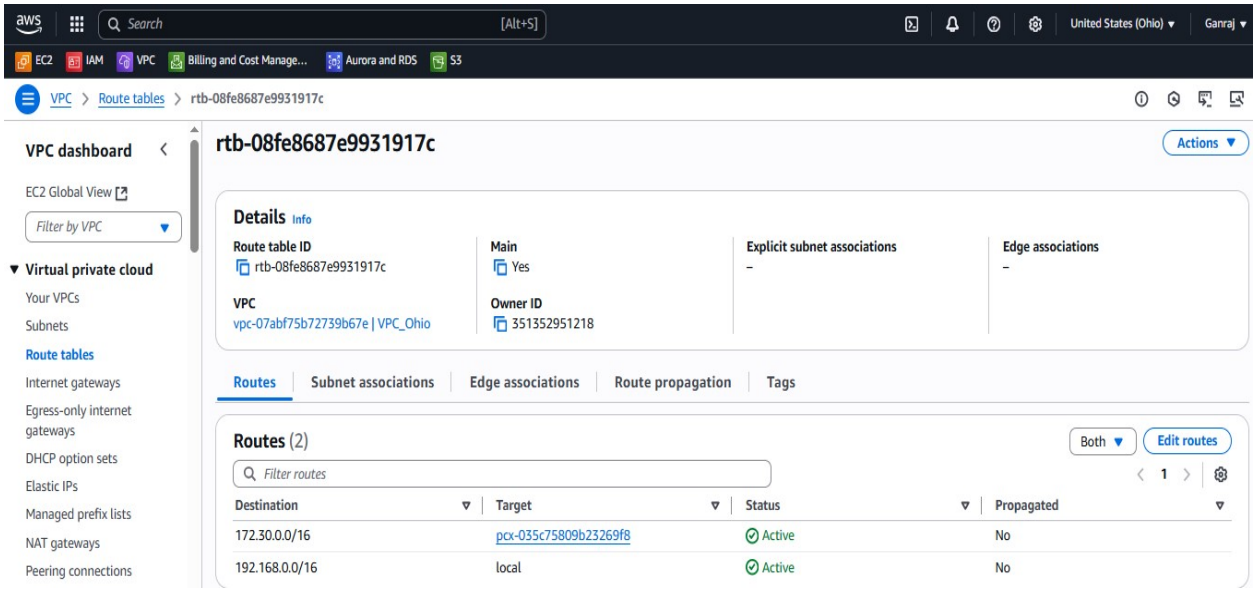




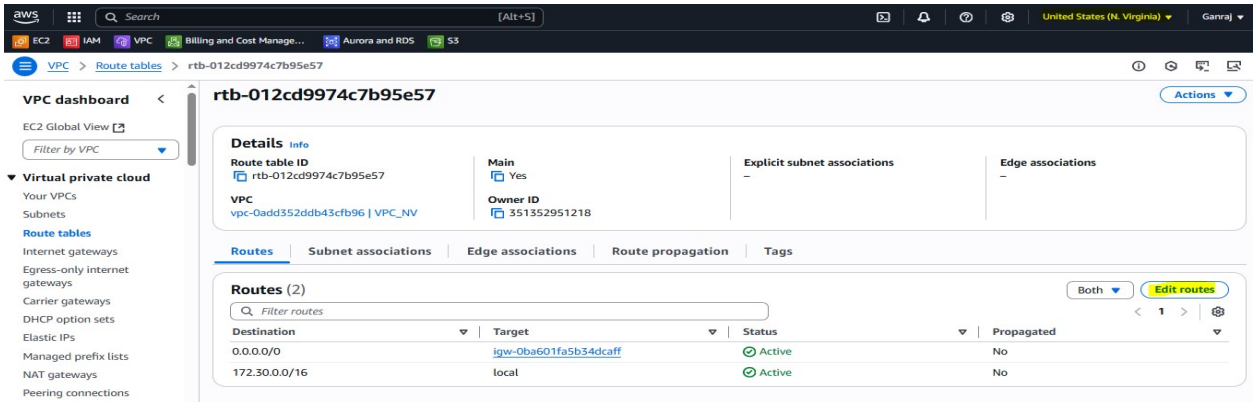
If need secure connection so putting 0.0.0.0/0 you can use direct CIDR of N.Virigina VPC 172.30.0.0/16

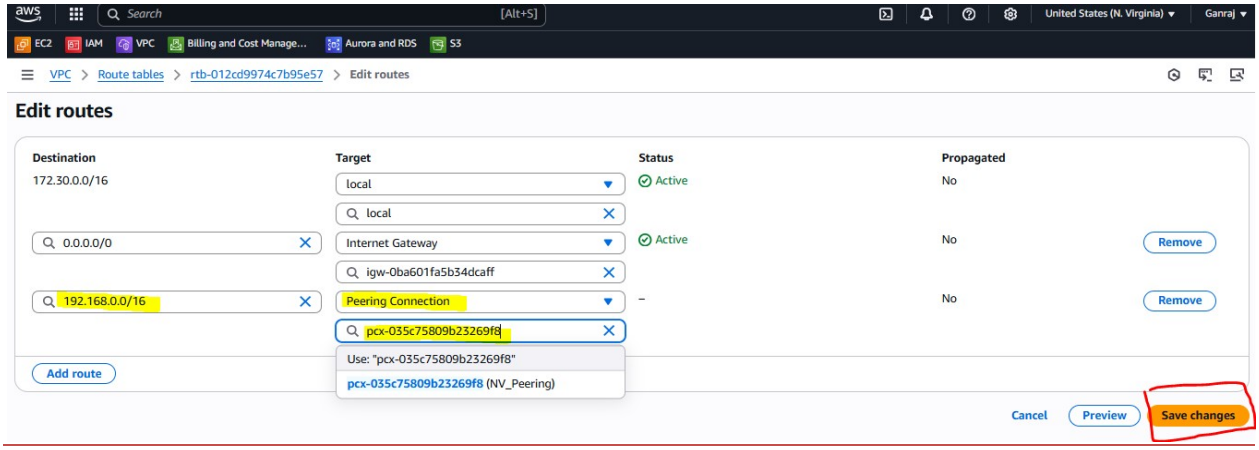


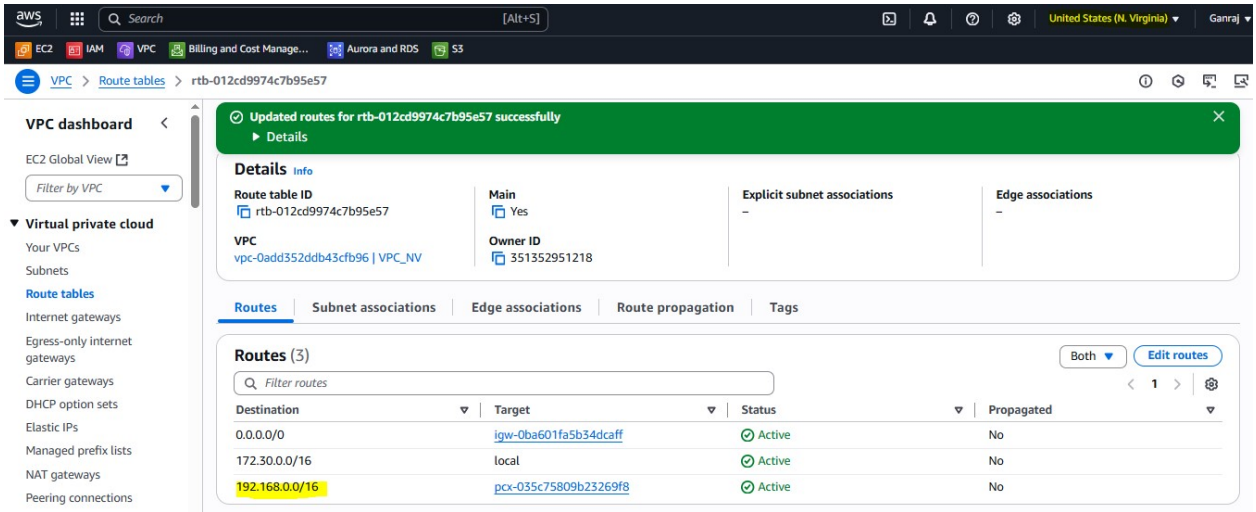
Here we have added route table peering with N.virigina



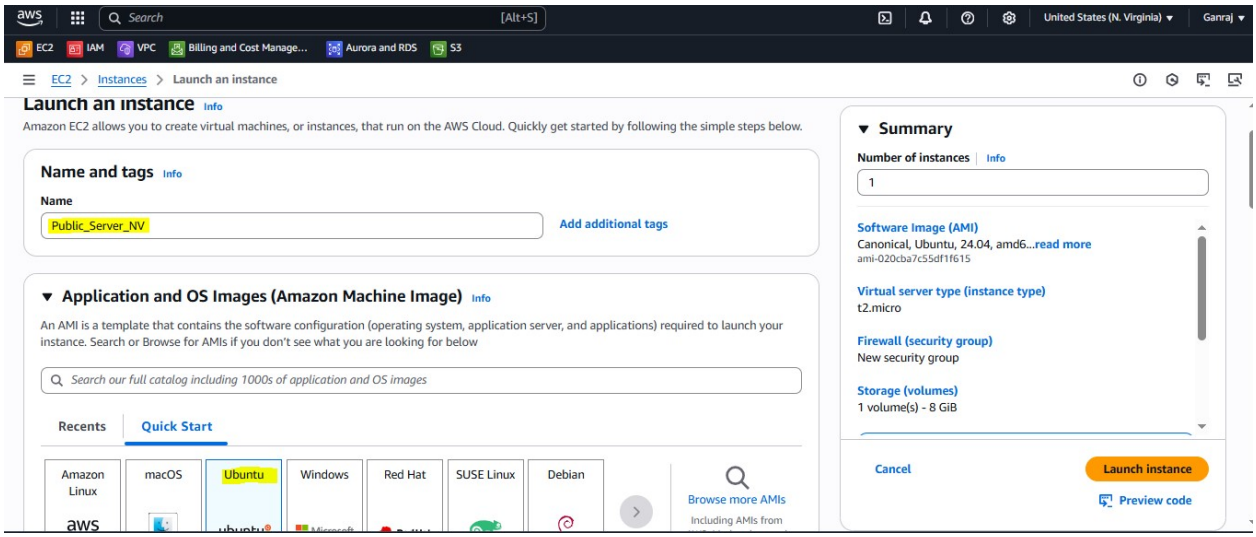
Now we will do same in N.virigina like cross connection.



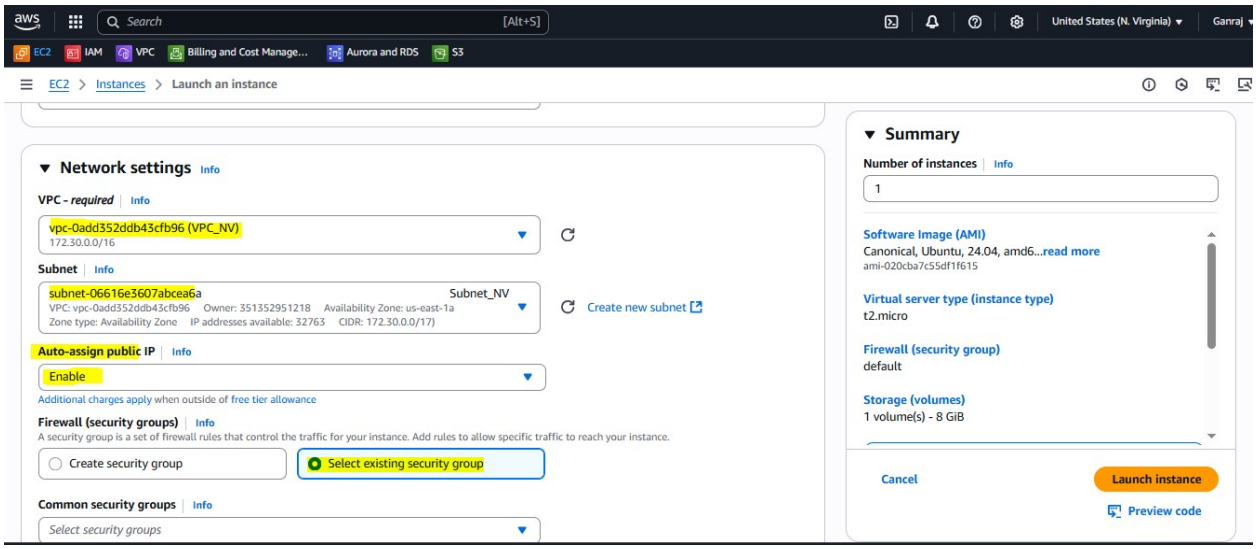




CONNECTION MADE BETWEEN TWO VPC USING PEERING CONNECTION NOW WE WE HAVE CREATE EC2'S FOR EACH VPC ONE WILL BE PUBLIC ANOTHER.

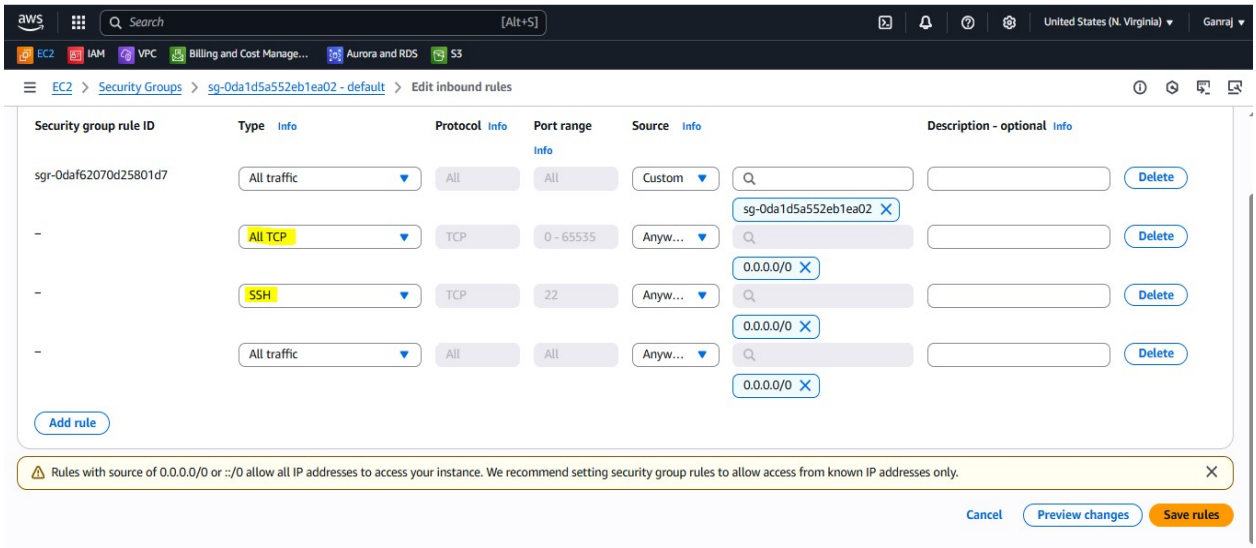


Here we have to select created VPC (VPC_NV) while creating Ec2 Instance



And then we Launch the instance and don't forget to changes in security group because here we are selected default and in the that security group not enabled the ports like SSH And HTTP or all TCP which provides connection Internet and access.

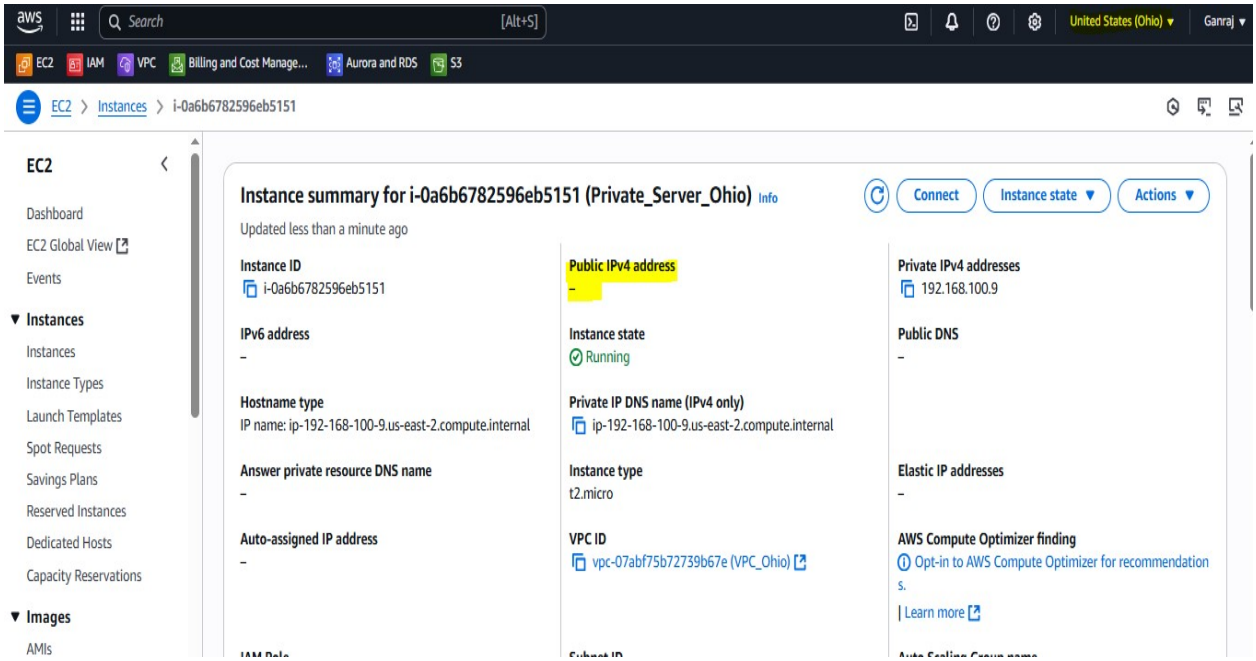
We are making changes in inbound Rule beware while changing it

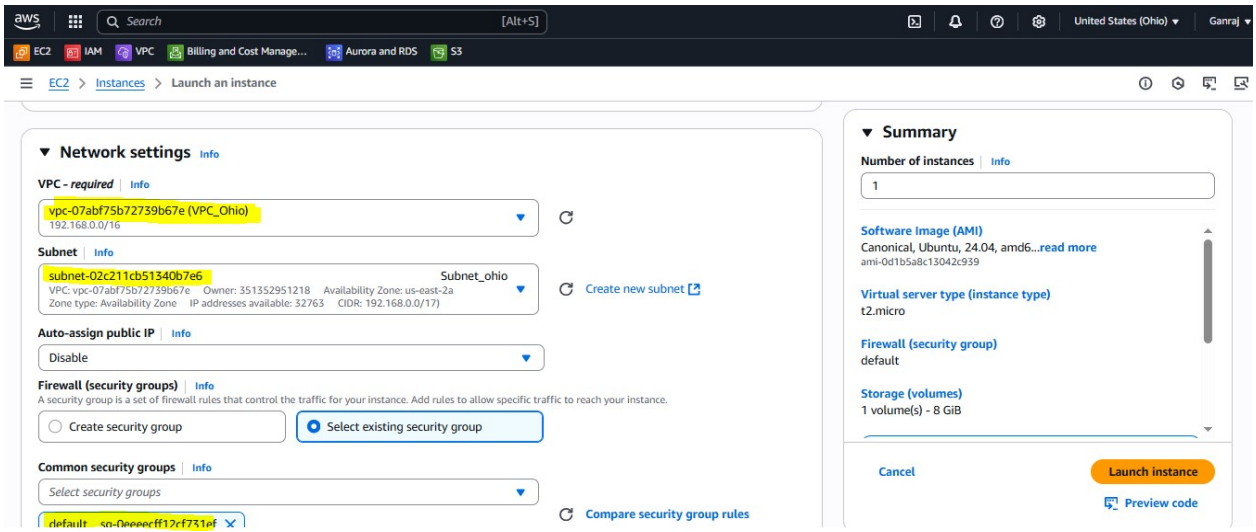


We need to create one EC2 in Ohio which is private server.

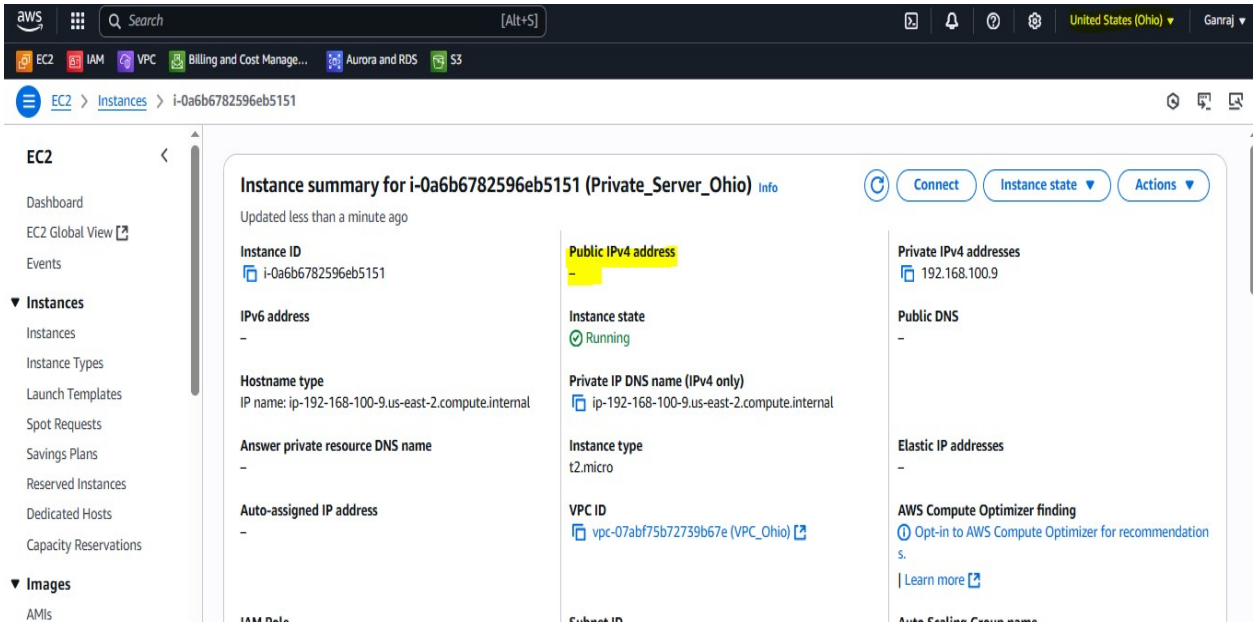
Here also we have select vpc that we have created. Need to changes in security groups also.

And see here we are not enabled the auto sign IP so this server will remain Private





here you can see this server has no Public IP.

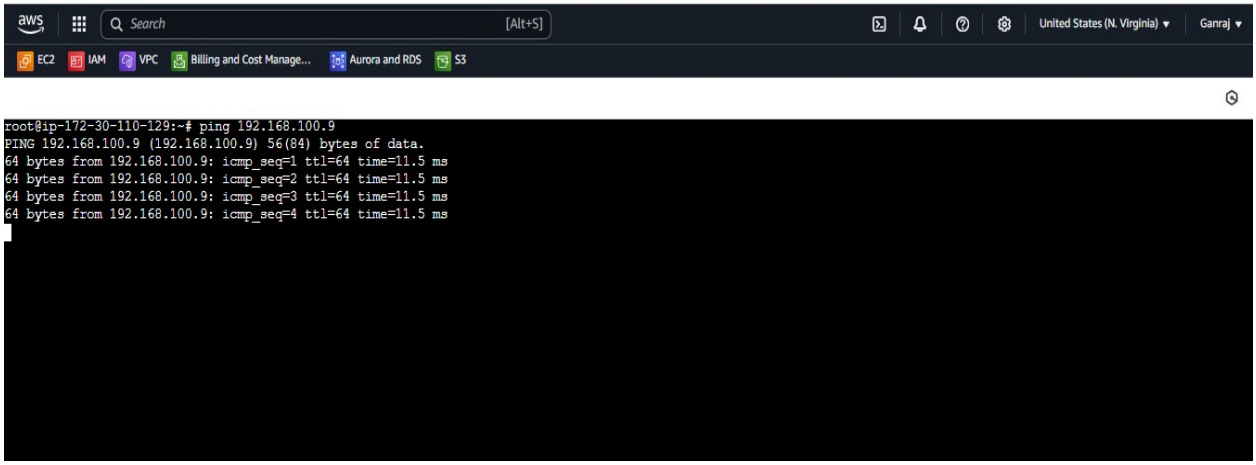


This will only get internet from public server

IMPORTANT-NOTE: Check security groups of both EC2 . AWS sometime not properly so always

Check that SSH, All traffic , ALL TCP and if need ICMP.

As you can see in the screeshot connection between Public Ec2 to Private Ec2 made.

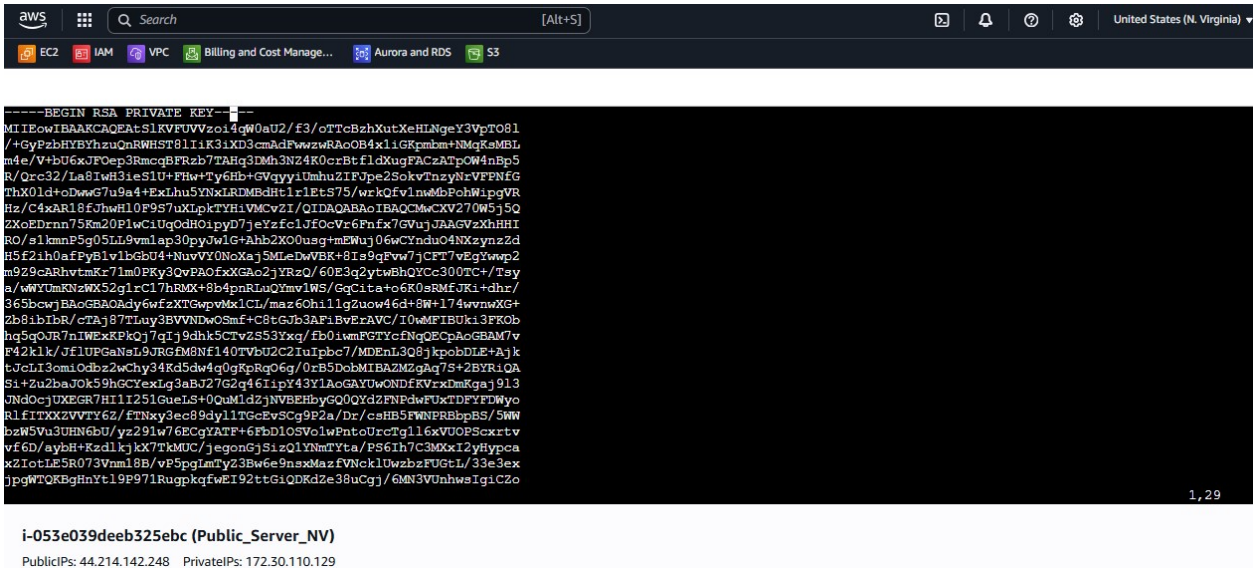


Now we will connect Private Sever (ec2) using Public Server but remember we need ssh key for that process and that ssh is store in the server.

So in that public ec2 server we first need to add ssh key which we have download on our local machine when we have created it while creating ec2 private server

First we open that key using notepad and copy it creating a pem file using vim editor and paste that ket content it.

Using vim peering.pem command opening the text editor and paste the content



And save it.

Now we will give permission to key run and access

chmod 400 peering.pem

Now will ssh command to access the private server in public server

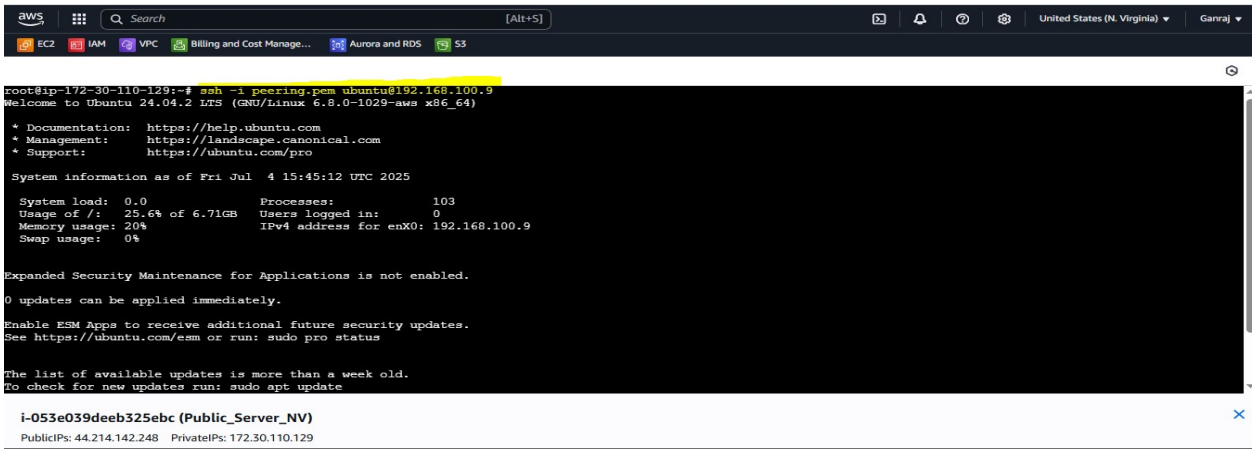
example :- ssh -i private-key.pem ec2-user@<Private-EC2-Private-IP>

ssh -i peering.pem ubuntu@192.168.100.9

Part	Meaning
`ssh`	Secure Shell command—used to log in remotely to a Linux server.
`-i peering.pem`	`-i` means "identity file." This tells SSH to use your **private key file named `peering.pem`** for authentication.
`Ubuntu@192.168.100.9`	This is the **login user** and **target host**: - `Ubuntu`: The **username** you are trying to log in as. - `192.168.100.9`: The **private IP address** of your EC2 instance inside the Ohio VPC.

OS	Username
Amazon Linux 2	ec2-user
Ubuntu	ubuntu
RHEL	ec2-user
CentOS	centos

Always remember while using names of username its case sensitive so type carefully



Using that command we have connected to private server(192.168.100.9) as you see in screenshot