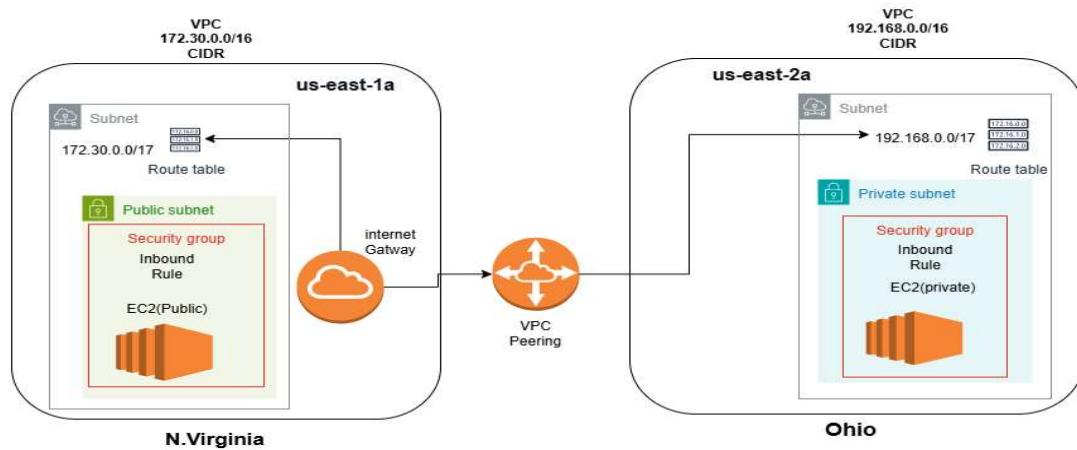


VPC Peering :-

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.virginia

The screenshot shows the AWS VPC dashboard with the following details:

- Header:** AWS logo, Search bar, United States (N. Virginia), Ganraj.
- Navigation:** EC2, IAM, VPC, Billing and Cost Management, Aurora and RDS, S3.
- Left sidebar:** VPC dashboard, EC2 Global View, 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).
- Main Content:**
 - Success Message:** You successfully deleted vpc-0898aba00d301c744 / VPC_NV.
 - Your VPCs (1):** Shows a table with one row:

Name	VPC ID	State	Block Public Access	IPv4 CIDR	IPv6 CIDR
vpc-0dbbb05820b87165c	vpc-0dbbb05820b87165c	Available	Off	172.31.0.0/16	-
 - vpc-0898aba00d301c744 / VPC_NV:** Details view with tabs: Details, Resource map, CIDs, Flow logs, Tags, Integrations. The Details tab shows:

VPC ID	State	Block Public Access	DNS hostnames
vpc-0898aba00d301c744	Available	Off	Disabled

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.

VPC_NV

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input
 IPAM-allocated IPv4 CIDR block

IPv4 CIDR
172.30.0.0/16

IPv4 block size must be between 16 and 30

VPC dashboard <

Your VPCs (2) [Info](#)

Last updated less than a minute ago ⟳ [Actions](#) Create VPC

Find VPCs by attribute or tag					
<input type="checkbox"/> Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/> -	vpc-0dbbb05820b87165c	Available	<input type="checkbox"/> Off	172.31.0.0/16	-
<input type="checkbox"/> VPC_NV	vpc-090b4aadbfcfd4bafe	Available	<input type="checkbox"/> Off	172.30.0.0/16	-

< 1 > ⟳

Virtual private cloud

- Your VPCs**
- Subnets
- Route tables

Creating Subnet in same VPC.

Create subnet [Info](#)

VPC
VPC ID
Create subnets in this VPC.
vpc-090b4aadbfcfd4bafe (VPC_NV)

Associated VPC CIDRs

IPv4 CIDRs
172.30.0.0/16

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

Subnet 1 of 1

aws | Search [Alt+S] | United States (N. Virginia) | Ganraj

EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3

VPC > Subnets > Create subnet

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
Subnet_NV
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 (N. Virginia) / us-east-1a

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.
172.30.0.0/16

IPv4 subnet CIDR block
172.30.0.0/17 32,768 IPs

Creating Internet Gateway with Same VPC.

aws | Search [Alt+S] | United States (N. Virginia) | Ganraj

EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3

VPC > Internet gateways

VPC dashboard

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways**
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists

Internet gateways (2) Info

Name	Internet gateway ID	State	VPC ID
-	igw-014b7c1da638c3747	Attached	vpc-0dbbb05820b87165c
-	igw-07bc94b08053c58d9	Detached	-

Create internet gateway

aws | Search [Alt+S] | United States (N. Virginia) | Ganraj

EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3

VPC > Internet gateways > Create internet gateway

Internet gateway successfully deleted - igw-0de89d3a3071ec824

Internet gateway settings

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

Tags - optional
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**

Q Name	X	Q NV_IGW	X	Remove
--------	---	----------	---	--------

Add new tag
You can add 49 more tags.

Create internet gateway

Need to attach to VPC (N.virginia) that we have created .

VPC dashboard < VPC > Internet gateways / igw-0ba601fa5b34dcfff / NV_IGW

Details Info

Internet gateway ID: igw-0ba601fa5b34dcfff State: Detached VPC ID: - Owner: 35135295

Tags

Name: NV_IGW

Actions ▾

- Attach to VPC
- Detach from VPC
- Manage tags
- Delete

Attach to VPC (igw-0ba601fa5b34dcfff) Info

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

Attach the internet gateway to this VPC.

vpc-0add352ddb43cfb96

Use: "vpc-0add352ddb43cfb96"

vpc-0add352ddb43cfb96 - VPC_NV

Cancel Attach internet gateway

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

VPC dashboard < VPC > Your VPCs / vpc-0add352ddb43cfb96

VPC vpc-0add352ddb43cfb96

DNS resolution: Enabled Main network ACL: adl-0760be325026fe833 IPv6 CIDR (Network border group): -

Tenancy: default Default VPC: No Network Address Usage metrics: Disabled

DHCP option set: dopt-058c9890e15e8aa56 IPv4 CIDR: 172.30.0.0/16

Route 53 Resolver DNS Firewall rule groups: -

Off Main route table: rtb-012cd9974c7b95e57 IPv6 pool: -

Disabled 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) Route network traffic to resources rtb-012cd9974c7b95e57

Network connections (1) Connections to other networks NV_IGW

VPC dashboard

vpc-0add352ddb43cfb96

DNS resolution Enabled	Tenancy default	Off	Disabled
Main network ACL acl-0760be325026fe833	Default VPC No	DHCP option set dept-058c9890e15e8aa56	Main route table rtb-012cd9974c7b95e57
IPv6 CIDR (Network border group) -	Network Address Usage metrics Disabled	IPv4 CIDR 172.30.0.0/16	IPv6 pool -
		Route 53 Resolver DNS Firewall rule groups -	Owner ID 351352951218

Resource map | CIDs | 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)
Route tables within this VPC
rtb-012cd9974c7b95e57
1 subnet association
1 route including local

Network connections (1)
Connections to other networks
NV_IGW

VPC dashboard

rtb-012cd9974c7b95e57

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)

Destination	Target	Status	Propagated
172.30.0.0/16	local	Active	No

VPC > **Route tables** > **rtb-012cd9974c7b95e57** > Edit routes

Edit routes

Destination 172.30.0.0/16	Target local	Status Active	Propagated No
	<input type="text"/> local	<input type="button" value="X"/>	

Add route

Cancel **Preview** **Save changes**

Screenshot of the AWS VPC Edit routes interface. A route is being added to a route table with a destination of 172.30.0.0/16, target set to Internet Gateway, and status set to Active. The 'Save changes' button is highlighted with a red box.

Screenshot of the AWS VPC Resource map interface. It shows the connection between a VPC (VPC_NV), Subnets (us-east-1a), Route tables (rtb-012cd9974c7b95e57), and Network connections (NV_IGW).

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

Now we will create VPC Ohio Region .

Screenshot of the AWS VPC Create VPC interface. A new VPC is being created with the name 'VPC_Ohio'. The 'IPv4 CIDR block' is set to 192.168.0.0/16. The 'Name tag - optional' field contains 'VPC_Ohio'.

aws | Search [Alt+S] | United States (Ohio) | Ganraj | EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3

VPC > Subnets > Create subnet

Create a new subnet with the settings you specify:

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 Name **Value - optional** Subnet_ohio **Add new tag**

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws | 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

aws | Search [Alt+S] | United States (Ohio) | Ganraj | EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3

VPC > Subnets > Create subnet

Create a new subnet with the settings you specify:

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 Name **Value - optional** Subnet_ohio **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 will not auto assign **IP(public)** because we have creating a private network or the private subnet which dosen't have the public access IP.

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with 'Virtual private cloud' options like Your VPCs, Subnets, Route tables, etc. The main area shows details for a specific subnet:

- Details** section:
 - Subnet ID: subnet-02c211cb51340b7e6
 - IPv4 CIDR: 192.168.0.0/17
 - Availability Zone: us-east-2a
 - Network ACL: -
 - Auto-assign customer-owned IPv4 address: No
 - IPv6 CIDR reservations: -
- Subnet ARN**: arn:aws:ec2:us-east-2:351352951218:subnet/subnet-02c211cb51340b7e6
- State**: Available
- IPv6 CIDR**: -
- VPC**: vpc-07abf75b72739b67e | VPC_Ohio
- Auto-assign public IPv4 address**: No (highlighted with a red box)
- Default subnet**: No
- Customer-owned IPv4 pool**: -
- IPv6-only**: -
- Outpost ID**: -
- Hostname type**: IP name
- Block Public Access**: Off
- IPv6 CIDR association ID**: -
- Route table**: -
- Auto-assign IPv6 address**: No
- IPv4 CIDR reservations**: -
- Resource name DNS A record**: Disabled

Now we will create **VPC Peering** : -

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

Note : We need to also connect VPC Peering CrossAccount.

The screenshot shows the AWS VPC Peering connections page. On the left, there's a sidebar with 'Virtual private cloud' options like Your VPCs, Subnets, Route tables, etc. The main area shows the 'Peering connections' section:

- Peering connections** section:
 - Actions** dropdown and **Create peering connection** button (highlighted with a red box).
 - A search bar: Find peering connections by attribute or tag.
 - A table with columns: Name, Peering connection ID, Status, Requester VPC, Acceptor VPC.
 - A message: No peering connection found.
 - A note: Select a peering connection above.

Screenshot of the AWS VPC Peering Connections creation page. The 'Name - optional' field contains 'NV_Peering'. The 'Select a local VPC to peer with' dropdown shows 'vpc-0add352ddb43cfb96 (VPC_NV)'. The 'VPC CIDRs for vpc-0add352ddb43cfb96 (VPC_NV)' table lists one entry: CIDR 172.30.0.0/16, Status Associated.

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

And create peering.

Screenshot of the AWS VPC dashboard for the Ohio Region. The VPC 'vpc-07abf75b72739b67e / VPC_Ohio' is selected. A red box highlights the 'Copied' status next to the VPC ID 'vpc-07abf75b72739b67e' in the list.

Screenshot of the AWS VPC Peering Connections creation page. The 'Region' section shows 'United States (Ohio) (us-east-2)' selected. The 'VPC ID (Acceptor)' dropdown shows 'vpc-07abf75b72739b67e'. The 'Tags' section contains a single tag 'NV_Peering'. The 'Create peering connection' button is at the bottom right.

AWS Search [Alt+S] United States (N. Virginia) Ganraj

EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3

VPC Peering connections ppx-035c75809b23269f8 NV_Peering

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

pcc-035c75809b23269f8 / NV_Peering

Details Info

Requester owner ID	351352951218	Acceptor owner ID	351352951218
Peering connection ID	pcc-035c75809b23269f8	Requester VPC	vpc-0add352ddb43cfb96 / VPC_NV
Status	Initiating Request to 351352951218	Requester CIDRs	172.30.0.0/16
Expiration time	Friday, July 11, 2025 at 19:39:27 GMT+5:30	Requester Region	N. Virginia (us-east-1)
		VPC Peering connection ARN	arn:aws:ec2:us-east-1:351352951218:vpc-peering-connection/pcc-035c75809b23269f8
		Acceptor VPC	vpc-07abf75b72739b67e
		Acceptor CIDRs	-
		Acceptor Region	Ohio (us-east-2)

DNS Route tables Tags

DNS settings Edit DNS settings

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

AWS Search [Alt+S] United States (Ohio) Ganraj

EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3

VPC Peering connections

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections**
- Route servers New

Peering connections (1/1) Info

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC
-	pcc-035c75809b23269f8	Pending acceptance	vpc-0add352ddb43cfb96	vpc-07abf75b72739b67e

pcc-035c75809b23269f8

Pending acceptance

You can accept or reject this peering connection request using the 'Actions' menu. You have until Friday, July 11, 2025 at 19:39:27 GMT+5:30 to accept or reject the request, otherwise it expires.

Details DNS Route tables Tags

AWS Search [Alt+S] United States (Ohio) Ganraj

EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3

VPC Peering connections

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections**
- Route servers New

Peering connections (1/1) Info

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC
-	pcc-035c75809b23269f8	Pending acceptance	vpc-0add352ddb43cfb96	vpc-07abf75b72739b67e

pcc-035c75809b23269f8

Pending acceptance

You can accept or reject this peering connection request using the 'Actions' menu. You have until Friday, July 11, 2025 at 19:39:27 GMT+5:30 to accept or reject the request, otherwise it expires.

Actions Create peering connection

- View details
- Accept request
- Reject request
- Edit DNS settings
- Manage tags
- Delete peering connection

Screenshot of the AWS VPC Peering connections page showing a successfully established peering connection between two VPCs.

Peering connections (1) Info

Requester CIDRs	Acceptor CIDRs	Requester owner ID	Acceptor owner ID	Requester Region	Acceptor Region
172.30.0.0/16	192.168.0.0/16	351352951218	351352951218	N. Virginia (us-east-1)	Ohio (us-east-2)

Screenshot of the AWS VPC Peering connections page showing the same successfully established peering connection.

Peering connections (1) Info

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC
-	pcx-035c75809b23269f8	Active	vpc-0add352ddb43cfb96	vpc-07abf

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

Now we need to connect route tables to each other (CrossAccount or region)

First we will go to Ohio private subnet route table.

Screenshot of the AWS VPC Your VPCs page for the Ohio VPC.

Resource map

- VPC Show details:** Your AWS virtual network (VPC_Ohio)
- Subnets (1):** Subnets within this VPC (us-east-2a, Subnet_ohio)
- Route tables (1):** Route network traffic to resources (rtb-08fe9687e9931917c)
 - 1 subnet association
 - 1 route including local
- Network connections (0):** Connections to other networks

VPC dashboard < Details info

Route table ID: rtb-08fe8687e9931917c Main Explicit subnet associations Edge associations

VPC: vpc-07abf75b72739b67e | VPC_Ohio Owner ID: 351352951218

Routes Subnet associations Edge associations Route propagation Tags

Routes (1)

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No

Both Edit routes

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

Edit routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
172.30.0.0/16	Peering Connection	-	No
pxc-035c75809b23269f8			

Add route Cancel Preview Save changes

Here we have added route tables peering with N.virginia

VPC dashboard < Details info

Route table ID: rtb-08fe8687e9931917c Main Explicit subnet associations Edge associations

VPC: vpc-07abf75b72739b67e | VPC_Ohio Owner ID: 351352951218

Routes (2)

Destination	Target	Status	Propagated
172.30.0.0/16	pxc-035c75809b23269f8	Active	No
192.168.0.0/16	local	Active	No

Both Edit routes

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

The screenshot shows the AWS VPC Route Tables page. The route table ID is rtb-012cd9974c7b95e57. It has one main entry (0.0.0.0/0) pointing to the Internet Gateway (igw-0ba601fa5b34dcff). There are no explicit subnet associations or edge associations.

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0ba601fa5b34dcff	Active	No
172.30.0.0/16	local	Active	No

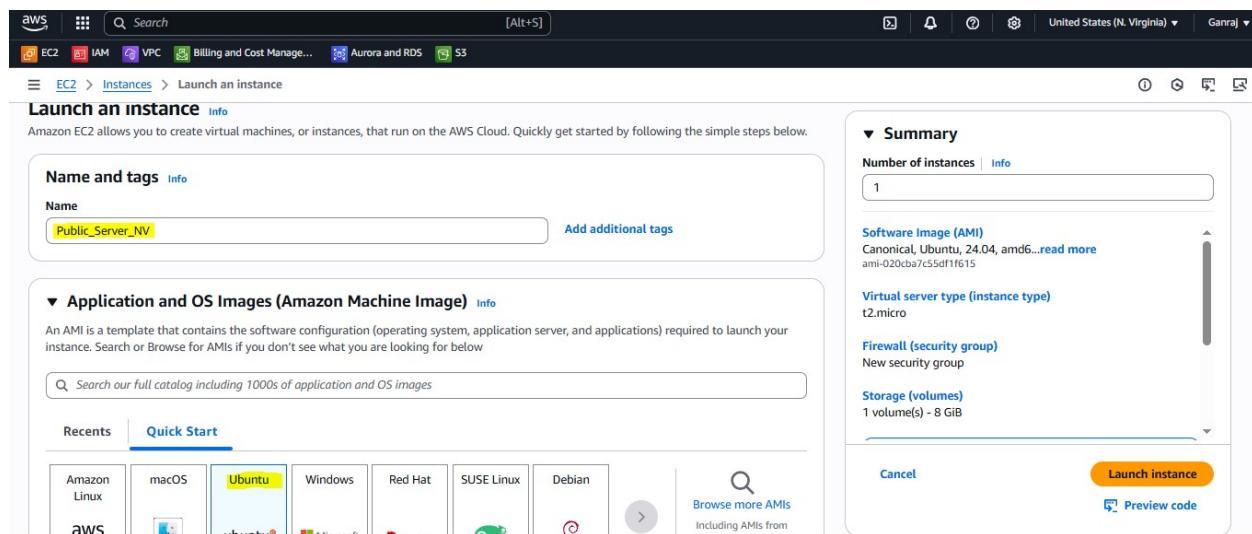
The screenshot shows the 'Edit routes' dialog box for route table rtb-012cd9974c7b95e57. A new route is being added for destination 172.30.0.0/16, targeting a Peering Connection (pcx-035c75809b23269f8). The 'Save changes' button is highlighted with a red box.

Destination	Target	Status	Propagated
172.30.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	Active	No
192.168.0.0/16	Peering Connection	-	No
	pcx-035c75809b23269f8		

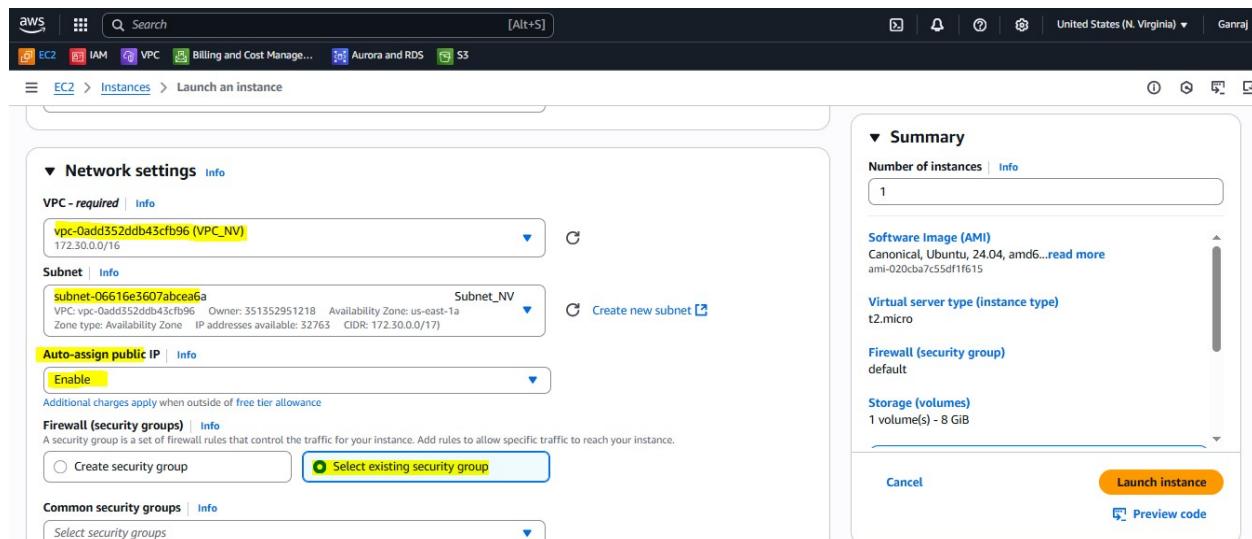
The screenshot shows the AWS VPC Route Tables page after saving the changes. The route table now has three entries: 0.0.0.0/0 pointing to the Internet Gateway, 172.30.0.0/16 pointing to local, and 192.168.0.0/16 pointing to the Peering Connection (pcx-035c75809b23269f8).

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0ba601fa5b34dcff	Active	No
172.30.0.0/16	local	Active	No
192.168.0.0/16	pcx-035c75809b23269f8	Active	No

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



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 be aware 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 done changes in security groups also.

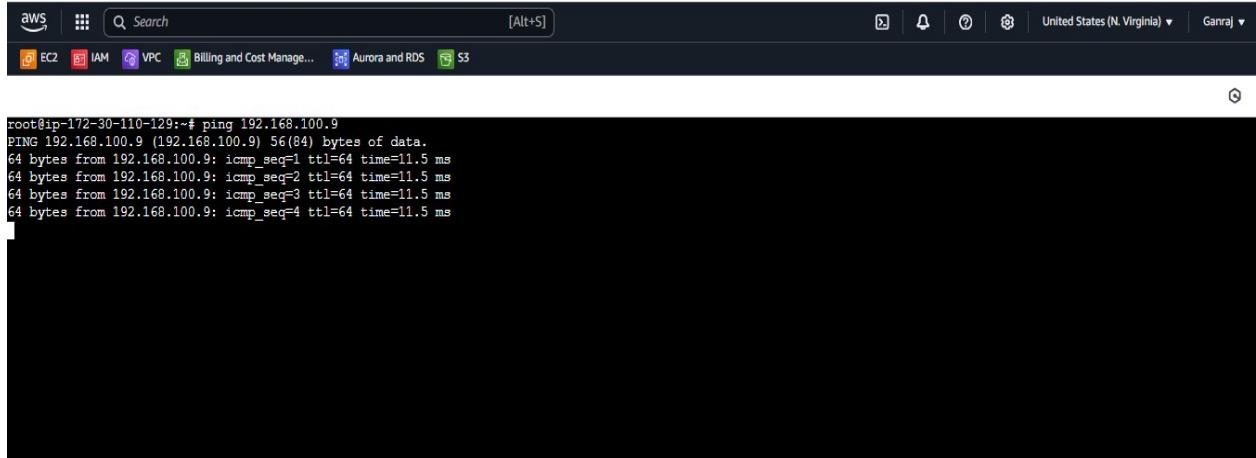
And see here we are not enabled the auto assign IP so that why server will be **remain Private**

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's . AWS sometimes not properly work so always check . Check that SSH, All traffic , ALL TCP and if need ICMP.

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

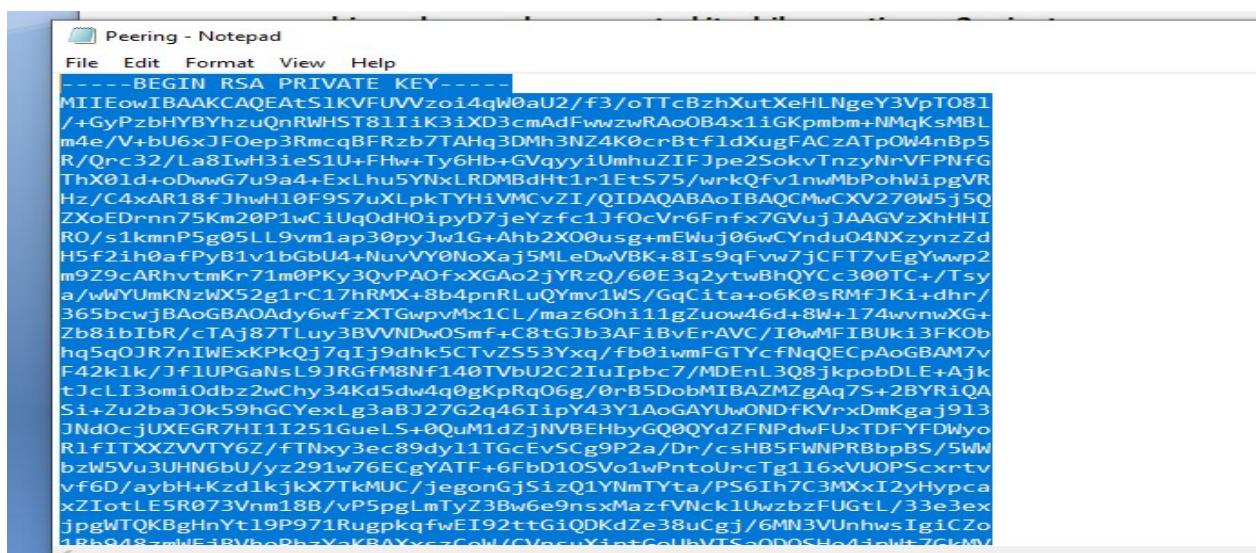


```
root@ip-172-30-110-129:~# ping 192.168.100.9
PING 192.168.100.9 (192.168.100.9) 56(84) bytes of data.
64 bytes from 192.168.100.9: icmp_seq=1 ttl=64 time=11.5 ms
64 bytes from 192.168.100.9: icmp_seq=2 ttl=64 time=11.5 ms
64 bytes from 192.168.100.9: icmp_seq=3 ttl=64 time=11.5 ms
64 bytes from 192.168.100.9: icmp_seq=4 ttl=64 time=11.5 ms
```

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

```
-----BEGIN RSA PRIVATE KEY-----  
MIIEowIBAAKCAQEAEtS1KVFUVVzoi4W0aUfF3/oTTCBzhXutXeHINgeY3VpOr081  
/+GyPzbHYBYhzuzOnRWHSST81iK3iXD3cmdfwzzRaoB4x1lGRymhm+NMqKsMBL  
m4e/V+bU6xJF0ep3RmcgBFxz7TAHQ3DMhNZ4K0crBtflDXugPACzATpW4nBp5  
R/Orc32/La8IwH3ieS1U+Fnw+Ty6Hb+GVqyyiUmhuZIIPe25okvInzyNrVFPNG  
rhxX0ld+oLwvG/u9a4+Exihu5YNxLRUM8dHt1-1et575/wkQfvlnM45phWipgVR  
Hz+4xAR18JhwH10P957uLpkTYHiVMcvzI/QIDQAABAIQCMwCXV270W5j5Q  
ZXoEdrn75km20PlwC1UgOdHOipyD7jeyfc1JFocvrfnx7/GVuJAAGVzXhHHI  
R0+siKmP5g05iL9vn1ap30pyJw1G+Abh2X0Usag+mEWu06wCYnduo4NXzynzZd  
H5f2ihoafPyb1vlGbU4+NuvVYONxKa5M1elTwBkR+81s9FvW7jCFT7veg7wp2  
m929cARhvtmKc71m0PKy3OvPAOfxXGAo2jyRzO/60E3g2ytBh0YCe300TC-/Tsy  
a/wYUmKNzWX52g1rC17RMX+8b4pnRLuQMyv1WS/GqCita+o6K0srMFJKi+dhr/  
365bcwjBaoGBAOAdy6wfz1TGWpVmxiCI/maz60h11lg2uoow46d+8W+174wvnwXG+  
zB81b1bR/cTAj87LuY3BVVNLwSmf+C8tGJb3AF1BvErAVC/10wMFIBUk13FKob  
hq5gQR7iWEKRPKQj7q1j9dhk5TvZSS3Xq/fb01wmFGTycfnqECpAcGBAM7v  
F421k/Jf1UPGaNsL3JRGfM8nf140TVBU22iulpb7/MDEn3Q8)kpobDLE+Ajk  
t;ciL13miOdbz2wChy34kd5dw4q0qxpRp06g/0xB5dokMTBzM2qAq7s+2BYR1OA  
Si+zu2baJ0k59hGCYextLg3aBzJ27c2q461Y43Y1AoGAYUwONDfKVrxmRgaj913  
JNd0cJUXEGR7H11251GuElS+0QuMidzjNVBEHbyGQ00Yd2fNPdxFUx1DFYFDWyo  
R1fITXXZVVY62/ftNky3ec89dy1TGeBvSCg9p2a/Dx/cshB5FWNPRBbpBS/5WW  
bzWSv3UHNgB0/y291w+6xCgyATF+6fbD1OSV01wntoUrcigl16xVUOFScxrtv  
vt6D/ayBH+Zzd1kjxk7TKMUC/jegonG/SizQ1YnmYtta/PS61h/C3MxxIzYhpc  
xzIotLE5R073Vm18B/v85pglmty23Bw6e9nsxMafVNcklUwzbzFUGtI/33e3ex  
JpgWTQKBgBgHnYt19P971RugpkqfwEI92ttG1QDKd2e38uCgj/6MN3VUnhwsIgjCZo  
-----END RSA PRIVATE KEY-----
```

1,29

i-053e039deeb325ebc (Public_Server_NV)
Public IPs: 44.214.142.248 Private IPs: 172.30.110.129

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	<p>-i means “identity file.”</p> <p>This tells SSH to use your private key file named peering.pem for authentication.</p>
Ubuntu@192.168.100.9	<p>This is the login user and target host:</p> <ul style="list-style-type: none"> - 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

```

aws [Alt+S] United States (N. Virginia) Ganraj
EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3

root@ip-172-30-110-129:~# ssh -i peering.pem ubuntu@192.168.100.9
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Jul  4 15:45:12 UTC 2025

System load: 0.0      Processes:          103
Usage of /: 25.6% of 6.71GB   Users logged in: 0
Memory usage: 20%      IPv4 address for enX0: 192.168.100.9
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
to check for new updates run: sudo apt update

i-053e039deeb325ebc (Public_Server_NV)
PublicIPs: 44.214.142.248 PrivateIPs: 172.30.110.129

```

Using that command we have connected to private server(192.168.100.9) from Public server(44.214.142.248) as you can see in the screenshot.

```

aws [Alt+S] United States (N. Virginia) Ganraj
EC2 IAM VPC Billing and Cost Manage... Aurora and RDS S3

root@ip-192-168-100-9:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eno0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:ff:03:ff:e9:el brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.9/17 metric 100 brd 192.168.127.255 scope global dynamic eno0
        valid_lft 346sec preferred_lft 346sec
    inet6 fe80::ff:3fe:fe3f:e9el/64 scope link
        valid_lft forever preferred_lft forever
root@ip-192-168-100-9:~#

```