VIRTUAL PRIVATE CLOUD (VPC)

A Virtual Private Cloud (VPC) in AWS is like having your own private piece of the internet. It's a virtual network that you can set up within AWS to run your applications and services. A VPC allows you to create a space where you can control and manage your own resources.

Within your VPC, you can launch various AWS resources such as virtual servers (EC2 instances), databases (RDS), and storage (S3) in an environment that is isolated from other users' resources. This isolation provides security and privacy for your applications and data.

You have control over who can enter and exit, and you can set up security groups to protect your VPC from outside incoming and outgoing traffic.

VPC needs ways to connect to the internet or other AWS services. You can set up gateways to make these connections safely.

Definition: ***AWS VPC is a virtual network environment within AWS that allows users to launch and manage AWS resources in a private, isolated section of the cloud.***

Key Components

Subnets: ***Segments of IP addresses within the VPC's CIDR block.***
Internet Gateway (IGW): ***Allows communication between instances in the VPC and the internet.***
NAT Gateway/NAT Instance: ***Provides outbound internet access for instances in private subnets.***
Route Tables: ***Direct traffic between subnets and gateways.***
Security Groups: ***Act as virtual firewalls for the Instance, controlling inbound and outbound traffic.***
Network Access Control Lists (NACLs): ***Provide subnet-level traffic control.***

Benefits

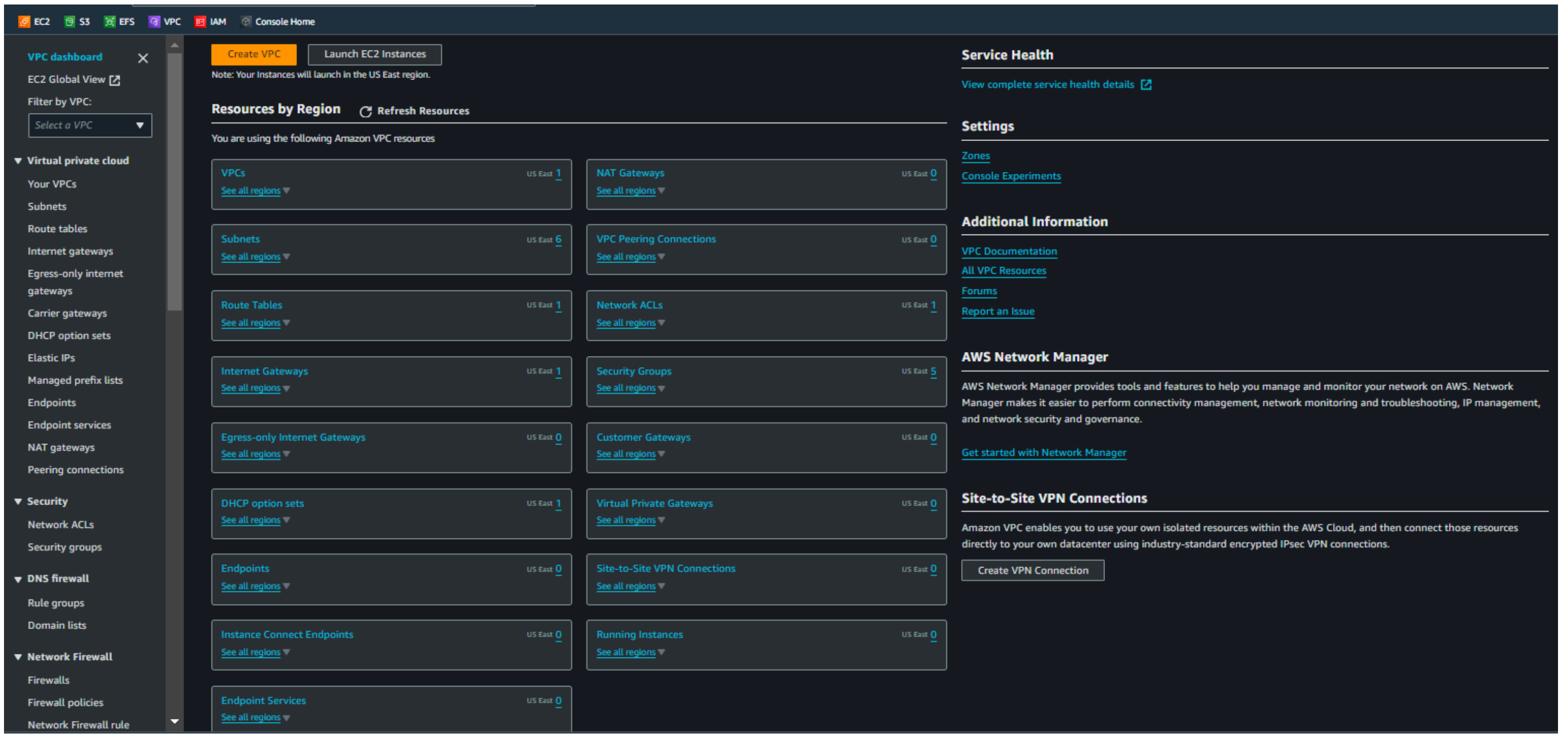
Isolation: ***Each VPC is logically isolated, providing security and privacy.***
Customization: ***Users can define their own IP address ranges, subnets, and routing tables.***
Scalability: ***Easily scale resources up or down within the VPC as needed.***
Security: ***Security Groups and NACLs provide granular control over traffic flow.***

Connectivity

Internet Access: ***Achieved through Internet Gateway or NAT Gateway (NAT For instance not have a Public IP Address)***
VPC Peering: ***Connects two VPCs privately to facilitate communication.***
VPN Connections: ***Establish encrypted connections between VPC and on-premises networks.***
VPC Endpoints: ***Privately connect VPC to AWS services without internet access.***

Now we create the VPC and launch the instance in it.

Step 1) *Search VPC in the AWS console and Select. This will take you to the VPC dashboard.*



Note: Your Instances will launch in the US East region.

Resources by Region Refresh Resources

You are using the following Amazon VPC resources

VPCs US East 1	NAT Gateways US East 0
Subnets US East 6	VPC Peering Connections US East 0
Route Tables US East 1	Network ACLs US East 1
Internet Gateways US East 1	Security Groups US East 5
Egress-only Internet Gateways US East 0	Customer Gateways US East 0
DHCP option sets US East 1	Virtual Private Gateways US East 0
Endpoints US East 0	Site-to-Site VPN Connections US East 0
Instance Connect Endpoints US East 0	Running Instances US East 0
Endpoint Services US East 0	

(each: See all regions)

Service Health

View complete service health details

Settings

Zones

Console Experiments

Additional Information

VPC Documentation

All VPC Resources

Forums

Report an Issue

AWS Network Manager

AWS Network Manager provides tools and features to help you manage and monitor your network on AWS. Network Manager makes it easier to perform connectivity management, network monitoring and troubleshooting, IP management, and network security and governance.

Get started with Network Manager

Site-to-Site VPN Connections

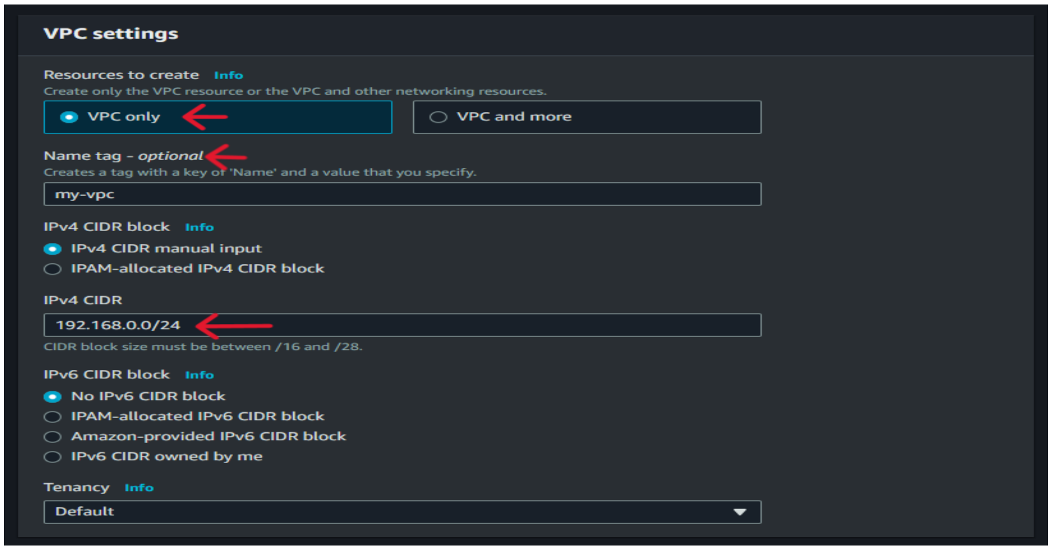
Amazon VPC enables you to use your own isolated resources within the AWS Cloud, and then connect those resources directly to your own datacenter using industry-standard encrypted IPsec VPN connections.

Create VPN Connection

Step 2) *Simply click on the Create VPC Button and select the only VPC option.*

Step 3) *Enter the VPC name that you want, Specify the IPv4 CIDR block to define the IP address range within VPC.*

In AWS we will get CIDR block size between /16 and /28



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-vpc

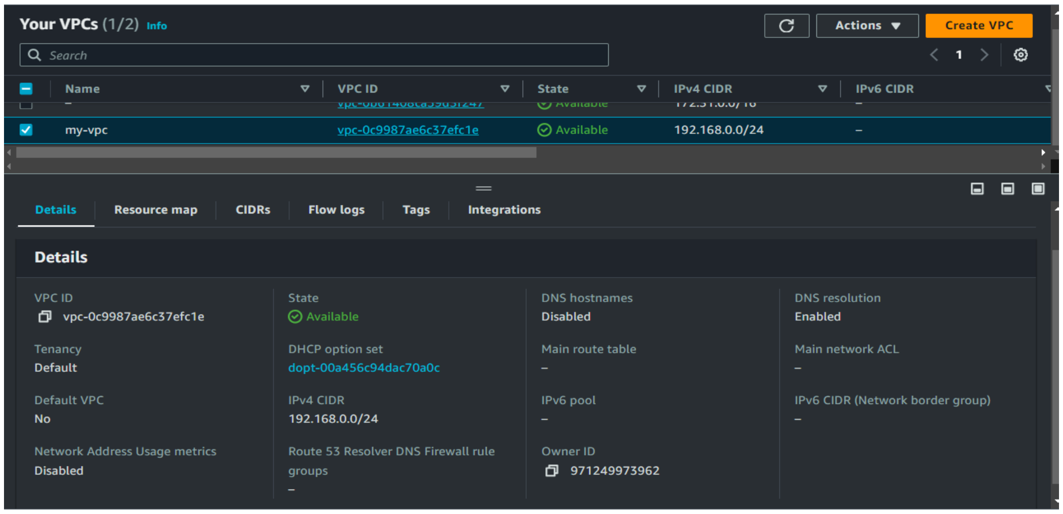
IPv4 CIDR block Info
● IPv4 CIDR manual input
○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
192.168.0.0/24
CIDR block size must be between /16 and /28.

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

Step 4) *Click on the Create VPC button. Your custom VPC will be created.*



Your VPCs (1/2) Info

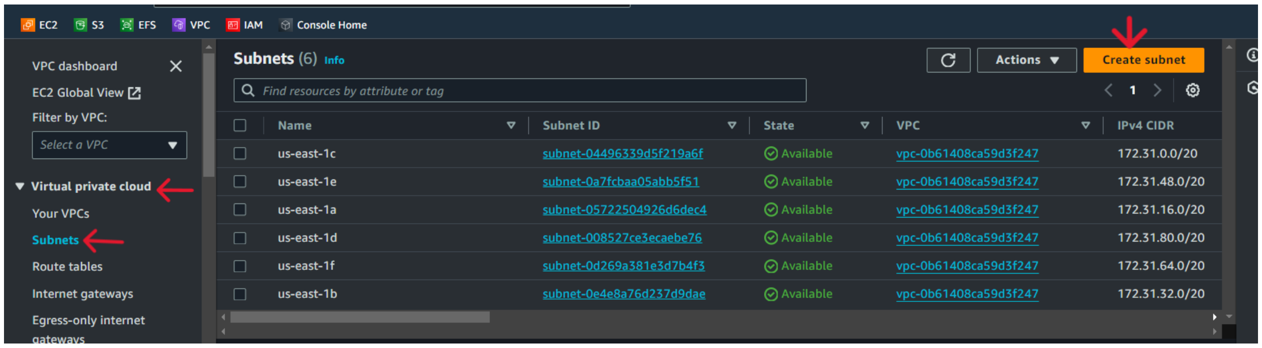
Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
my-vpc	vpc-0c9987ae6c37efc1e	Available	192.168.0.0/24	–

Details

VPC ID vpc-0c9987ae6c37efc1e	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-00a456c94dac70a0c	Main route table –	Main network ACL –
Default VPC No	IPv4 CIDR 192.168.0.0/24	IPv6 pool –	IPv6 CIDR (Network border group) –
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 971249973962	

Step 5) *Now we need to create a **subnet** within our custom VPC, Subnets are segments of the VPC's IP address range.*

- *Select the Subnet option in the VPC Dashboard Menu. Click on the Create Subnet button.*



Subnets (6) Info

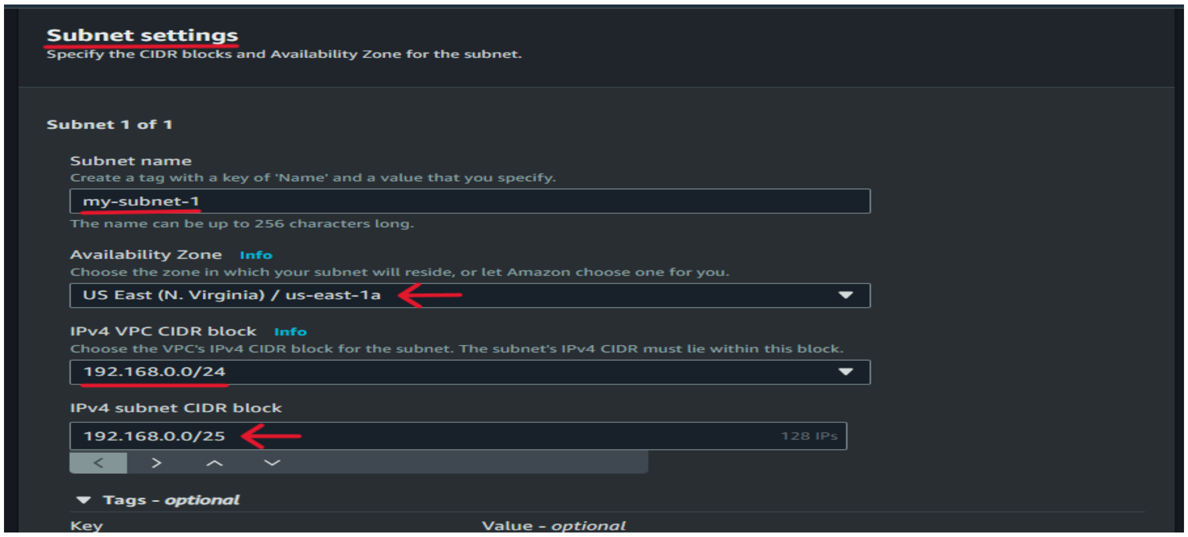
Name	Subnet ID	State	VPC	IPv4 CIDR
us-east-1c	subnet-04496339d5f219a6f	Available	vpc-0b61408ca59d3f247	172.31.0.0/20
us-east-1e	subnet-0a7fcbaa05abb5f51	Available	vpc-0b61408ca59d3f247	172.31.48.0/20
us-east-1a	subnet-05722504926d6dec4	Available	vpc-0b61408ca59d3f247	172.31.16.0/20
us-east-1d	subnet-008527ce3ecaebe76	Available	vpc-0b61408ca59d3f247	172.31.80.0/20
us-east-1f	subnet-0d269a381e3d7b4f3	Available	vpc-0b61408ca59d3f247	172.31.64.0/20
us-east-1b	subnet-0e4e8a76d237d9dae	Available	vpc-0b61408ca59d3f247	172.31.32.0/20

Step 6) *Select our custom VPC ID.*

Step 7) *Enter the Subnet name, select the availability zone, define the CIDR block, and Click the Create Subnet button.*

We specified CIDR block 192.168.0.0/24, and its IP range from 192.168.0.0 to 192.168.0.255.

There are a total of $2^{32-24} = 2^8 = 256$ IP addresses in this block



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-subnet-1
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.
US East (N. Virginia) / us-east-1a

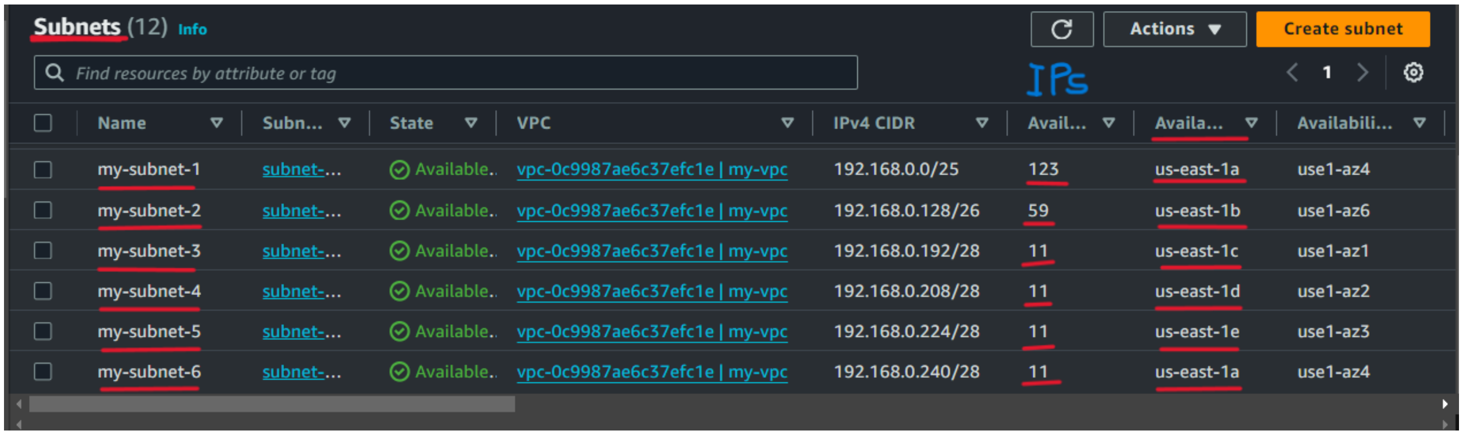
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
192.168.0.0/24

IPv4 subnet CIDR block
192.168.0.0/25 128 IPs

Tags - *optional*
Key Value - *optional*

Our first subnet's CIDR is /25, IPs = $2^{32-25} = 2^7 = 128$ IPS, so the IP address range is 192.168.0.0 to 192.168.0.127.

Now follow the above step and create more subnets in different availability zones.



Subnets (12) Info

Name	Subn...	State	VPC	IPv4 CIDR	Avail...	Availa...	Availabili...
my-subnet-1	subnet-...	Available..	vpc-0c9987ae6c37efc1e | my-vpc	192.168.0.0/25	123	us-east-1a	use1-az4
my-subnet-2	subnet-...	Available..	vpc-0c9987ae6c37efc1e | my-vpc	192.168.0.128/26	59	us-east-1b	use1-az6
my-subnet-3	subnet-...	Available..	vpc-0c9987ae6c37efc1e | my-vpc	192.168.0.192/28	11	us-east-1c	use1-az1
my-subnet-4	subnet-...	Available..	vpc-0c9987ae6c37efc1e | my-vpc	192.168.0.208/28	11	us-east-1d	use1-az2
my-subnet-5	subnet-...	Available..	vpc-0c9987ae6c37efc1e | my-vpc	192.168.0.224/28	11	us-east-1e	use1-az3
my-subnet-6	subnet-...	Available..	vpc-0c9987ae6c37efc1e | my-vpc	192.168.0.240/28	11	us-east-1a	use1-az4

*In AWS, when you create a subnet within a VPC, **AWS reserves 5 IP addresses of each subnet for its own use**. Typically, AWS reserves the following five IP addresses within each subnet:*

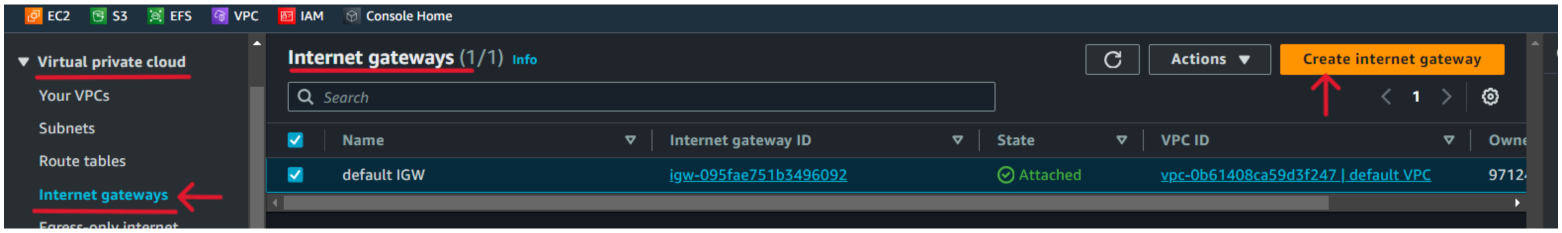
1. *The **first** IP address in a subnet is reserved for the **network address.***
2. *The **second** IP address is reserved for **DNS resolution** and other **AWS services**.*
3. *The **third** IP address is reserved for **future use by AWS**.*
4. *The **fourth** IP address is reserved for the **VPC router**. Its IP address is used as the **default gateway** for instances within the subnet to route traffic outside the VPC.*
5. *The **fifth** IP address is reserved for the **broadcast address**, although it's not used in AWS.*

Step 8) Now we need to set up an Internet Gateway to our custom VPS is crucial for

- ***Internet Access**: Enables resources within the VPC to connect to the internet.*
- ***Public Facing Services**: Allows hosting of services (e.g., web servers, APIs) accessible from the internet.*
- ***Updates and Patches**: Facilitates fetching updates and patches from external repositories.*
- ***Hybrid Cloud Configurations**: Enables communication between on-premises infrastructure and cloud resources.*
- ***External APIs and Services**: Allows accessing external APIs or services hosted on the internet.*
- ***Outbound Traffic Monitoring and Filtering**: Enables security measures such as traffic filtering, monitoring, and logging for outbound traffic.*

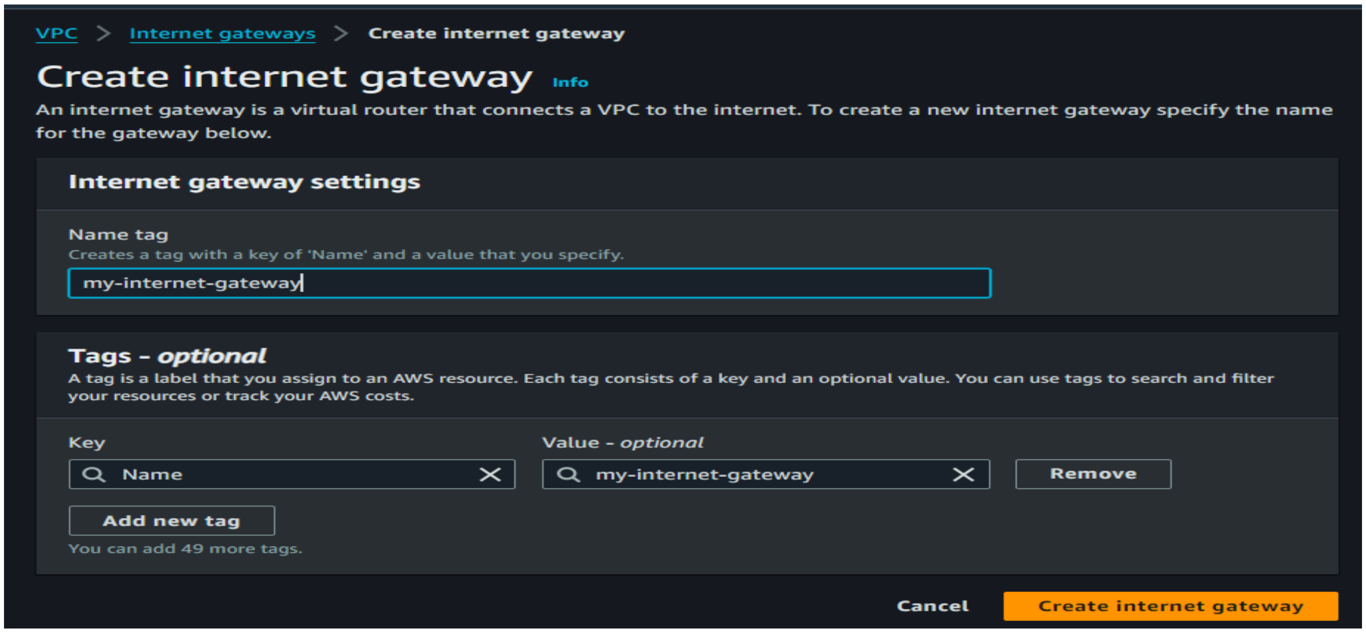
Steps to create an Internet gateway:

a) *In the VPC dashboard select the Internet Gateway option and click on the Create Internet Gateway button.*



Name	Internet gateway ID	State	VPC ID	Owne
default IGW	igw-095fae751b3496092	Attached	vpc-0b61408ca59d3f247 | default VPC	9712

b) *Name the Internet gateway, and click on the Create Internet Gateway button.*


VPC > Internet gateways > Create internet gateway
Create internet gateway Info
Internet gateway settings
Tags - optional
Add new tag
Remove
Cancel
Create internet gateway

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

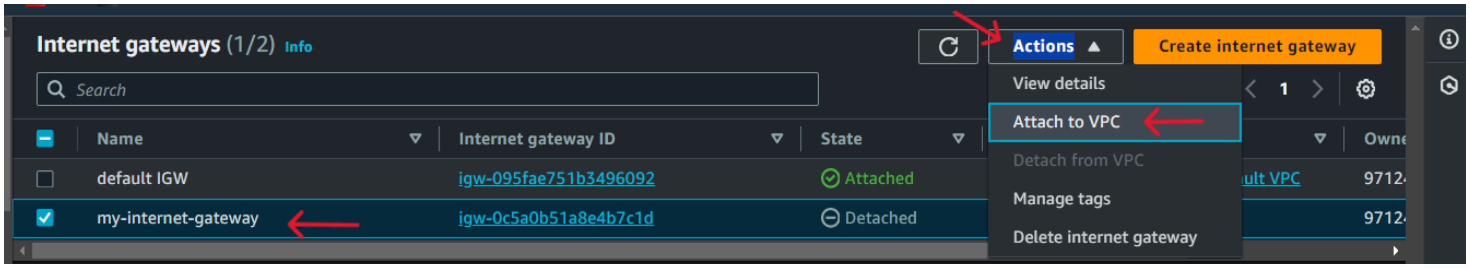
Name tag
Creates a tag with a key of 'Name' and a value that you specify.
my-internet-gateway

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: Name
Value - optional: my-internet-gateway

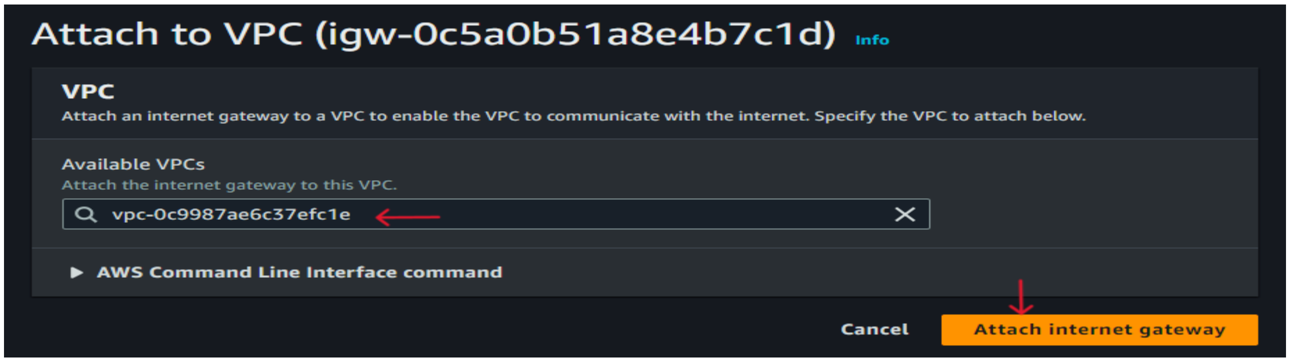
You can add 49 more tags.

c) *Select Internet Gateway, click on the action button, and select Attach to the VPC.*



Name	Internet gateway ID	State	Owne
default IGW	igw-095fae751b3496092	Attached	9712
my-internet-gateway	igw-0c5a0b51a8e4b7c1d	Detached	9712

d) *Select our custom VPC, and Attach the IGW to the VPC.*

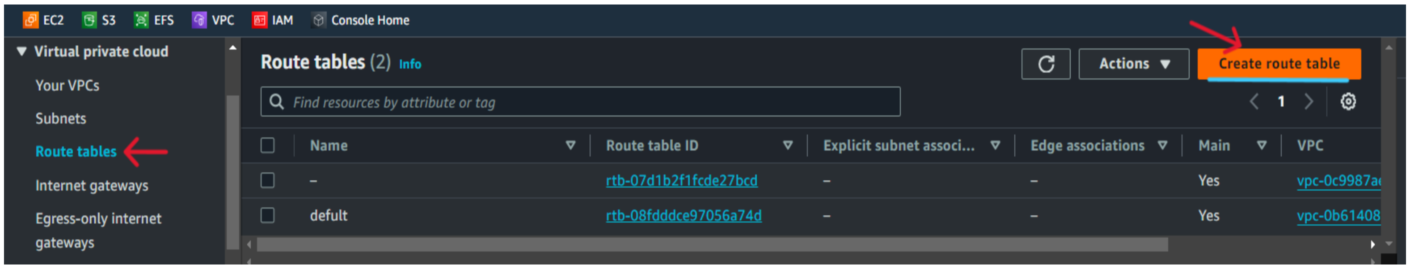

Attach to VPC (igw-0c5a0b51a8e4b7c1d) Info
VPC
AWS Command Line Interface command
Cancel
Attach internet gateway

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-0c9987ae6c37efc1e

Step 9) Default Route table is automatically created when we create VPC. So we can edit it or create a custom route table.
Creating custom route tables for a VPC is necessary to achieve granular control, define specific routing policies, enable internet and VPN connectivity, accommodate diverse networking requirements, optimize performance, and enhance security within the VPC environment.

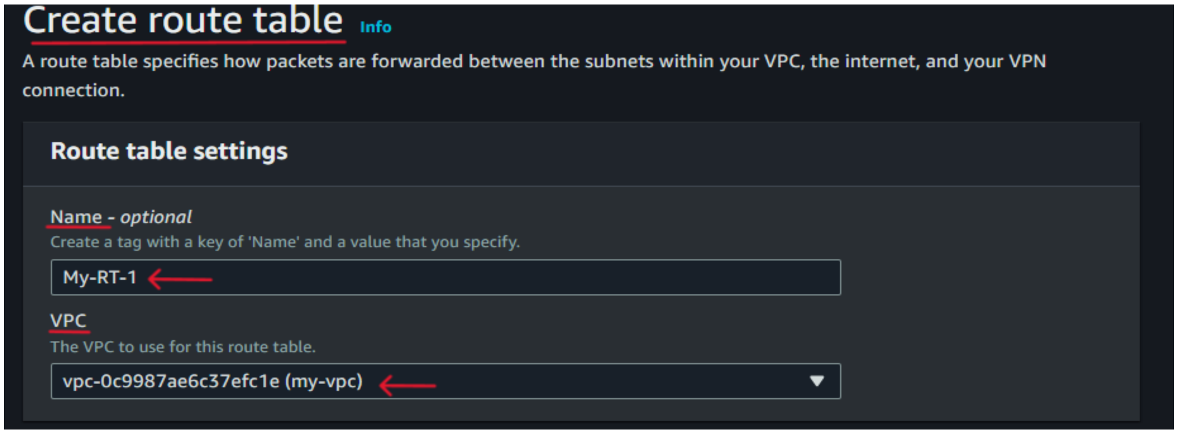
- *Select Route Table in VPC Dashboard, and click on the create route table.*



Route tables (2) Info

	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
☐	–	rtb-07d1b2f1fcde27bcd	–	–	Yes	vpc-0c9987ae
☐	defult	rtb-08fdddce97056a74d	–	–	Yes	vpc-0b61408

- *Name Route Table, select our custom VPC, and click on the Create route table button.*



Create route table Info

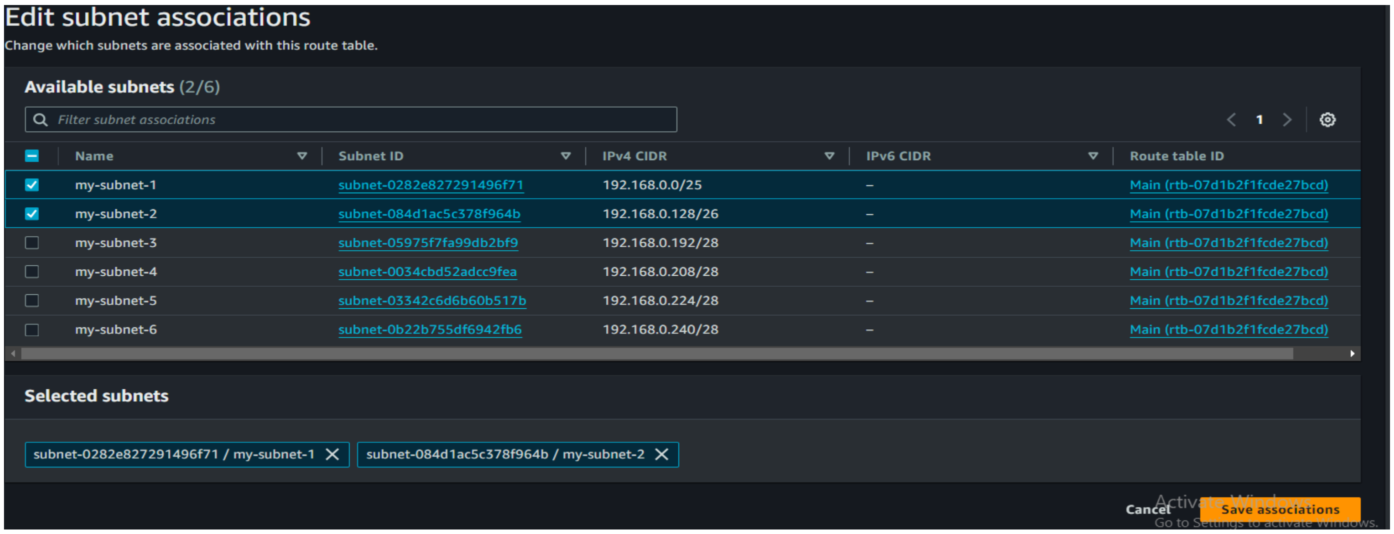
A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

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

VPC
The VPC to use for this route table.
vpc-0c9987ae6c37efc1e (my-vpc)

- *Select the custom route table, click on the action button then click the Edit Subnet association option.*
- *Select subnets that you want to associate with this custom route table and click the save association button.*



Edit subnet associations
Change which subnets are associated with this route table.

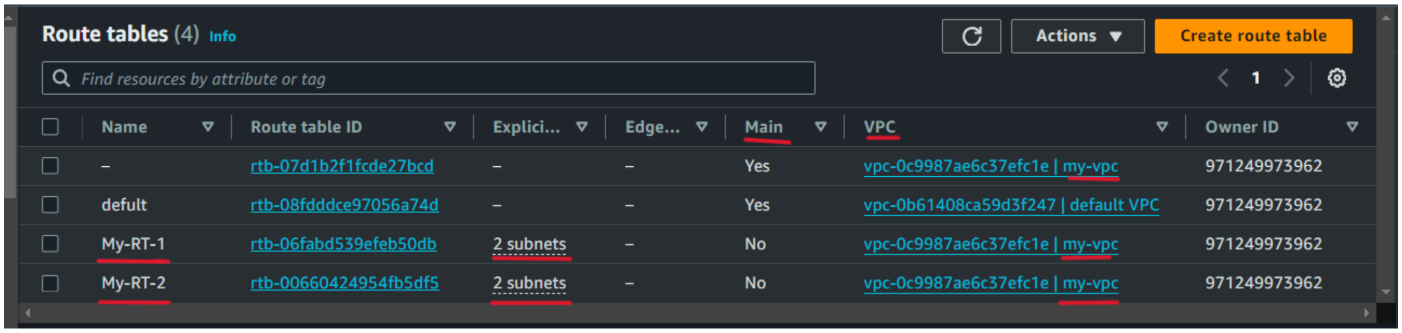
Available subnets (2/6)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
✓	my-subnet-1	subnet-0282e827291496f71	192.168.0.0/25	–	Main (rtb-07d1b2f1fcde27bcd)
✓	my-subnet-2	subnet-084d1ac5c378f964b	192.168.0.128/26	–	Main (rtb-07d1b2f1fcde27bcd)
☐	my-subnet-3	subnet-05975f7fa99db2bf9	192.168.0.192/28	–	Main (rtb-07d1b2f1fcde27bcd)
☐	my-subnet-4	subnet-0034cbd52adcc9fea	192.168.0.208/28	–	Main (rtb-07d1b2f1fcde27bcd)
☐	my-subnet-5	subnet-03342c6d6b60b517b	192.168.0.224/28	–	Main (rtb-07d1b2f1fcde27bcd)
☐	my-subnet-6	subnet-0b22b755df6942fb6	192.168.0.240/28	–	Main (rtb-07d1b2f1fcde27bcd)

Selected subnets

subnet-0282e827291496f71 / my-subnet-1
subnet-084d1ac5c378f964b / my-subnet-2

- *Now create another route table with associate different subnets.*

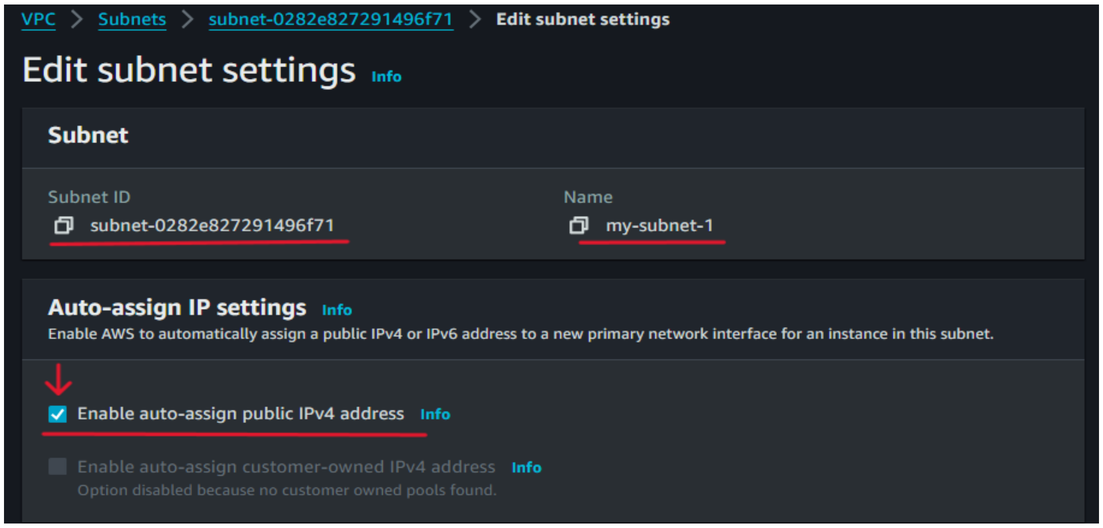


Route tables (4) Info

	Name	Route table ID	Explici...	Edge...	Main	VPC	Owner ID
☐	–	rtb-07d1b2f1fcde27bcd	–	–	Yes	vpc-0c9987ae6c37efc1e | my-vpc	971249973962
☐	defult	rtb-08fdddce97056a74d	–	–	Yes	vpc-0b61408ca59d3f247 | default VPC	971249973962
☐	My-RT-1	rtb-06fabd539efeb50db	2 subnets	–	No	vpc-0c9987ae6c37efc1e | my-vpc	971249973962
☐	My-RT-2	rtb-00660424954fb5df5	2 subnets	–	No	vpc-0c9987ae6c37efc1e | my-vpc	971249973962

Step 10) *Our all subnets are by default **disabled** to **auto-assign public IPv4 addresses** by AWS. This means that by default, instances launched in those subnets **won't receive public IP addresses**. So here we **enable** this option for subnets under our **My-RT-1 route table**. (Also there is an option to enable or disable auto-assign public IPv4 address when you launch the instance)*

- *Go to the subnets menu, select subnet, click on the action button then select the Edit subnet setting option.*
- *Check the box of Enable Auto-assign IPv4 address, and click on the save button.*



VPC > Subnets > subnet-0282e827291496f71 > Edit subnet settings

Edit subnet settings Info

Subnet

Subnet ID: subnet-0282e827291496f71
Name: my-subnet-1

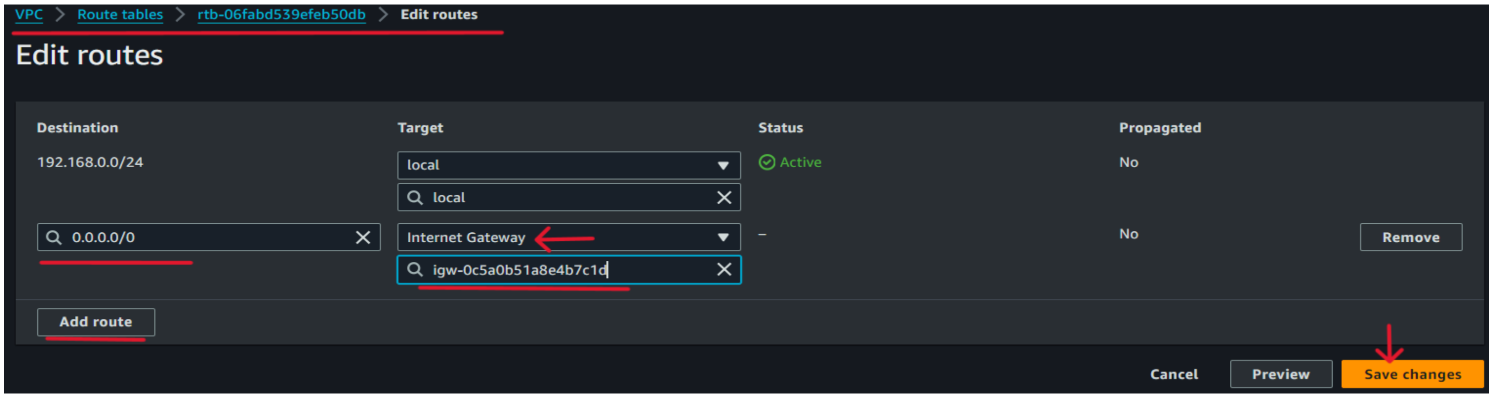
Auto-assign IP settings Info

Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

[✓] Enable auto-assign public IPv4 address Info

[ ] Enable auto-assign customer-owned IPv4 address Info
Option disabled because no customer owned pools found.

Step 11) *After creating a route table we need to edit it to add a route for internet access by setting up the target Internet gateway.*



VPC > Route tables > rtb-06fabd539efeb50db > Edit routes

Edit routes

Destination	Target	Status	Propagated
192.168.0.0/24	local	Active	No
0.0.0.0/0	Internet Gateway / igw-0c5a0b51a8e4b7c1d	-	No

Adding destination 0.0.0.0/0 represents the default route. This effectively allows resources within your VPC to access the internet. Any traffic destined for external addresses (including websites, APIs, etc.) will be routed through the Internet Gateway, enabling internet connectivity for your VPC.

Now we have created custom VPC, Subnets within custom VPC, Internet gateway and attached to VPC, then created route table and subnet associated with it, then added the route in route table by setting up target IGW with destination 0.0.0.0/0.

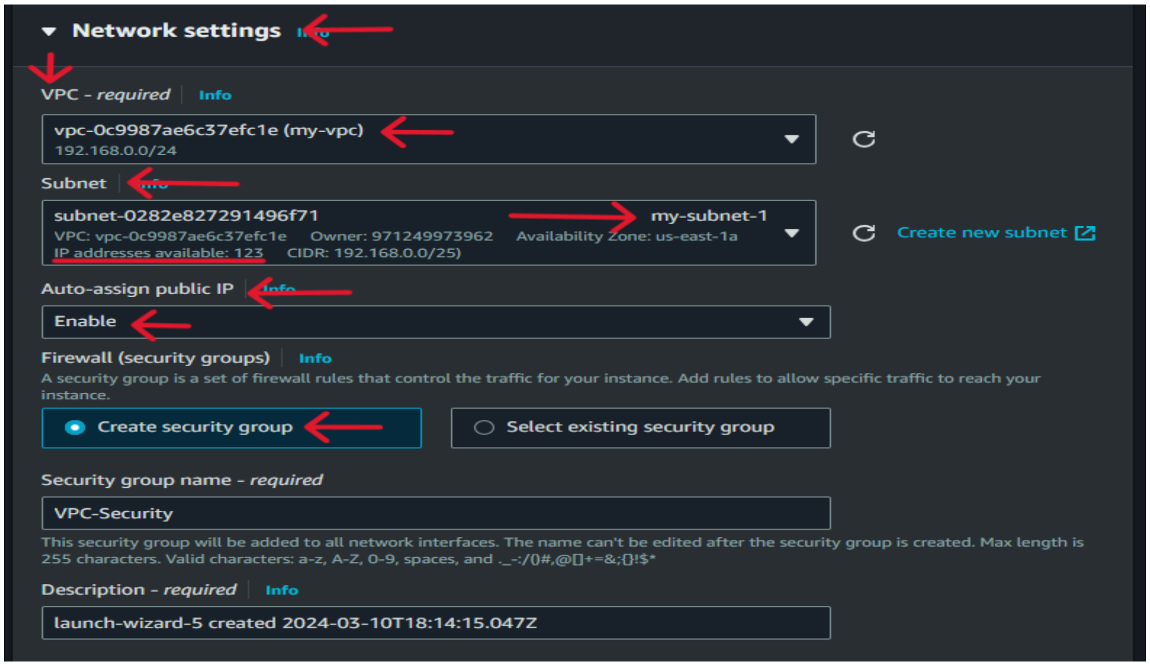
*That means we have successfully set up **basic network infrastructure** within our **custom VPC** in AWS.*

Step 12) ***Now we launch 2 Instances within our custom VPC. 1st with public IPv4 address and 2nd is without Public IPv4 address.***

Instance with Public IPv4 Address:	***Instance without Public IPv4 Address:***
• *This instance has a public IPv4 address assigned to it, allowing it to communicate directly with the internet.* • *It can be accessed from the internet using its public IP address.* • *Useful for instances that need to serve content or provide services directly to external clients over the internet.* • *Commonly used for web servers, public-facing APIs, and other services that require direct internet access.*	• *This instance does not have a public IPv4 address assigned to it, making it inaccessible directly from the internet.* • *It can still communicate with other instances within the same VPC or with resources in other VPCs connected via VPN or Direct Connect.* • *Accessible from the internet indirectly via services like **NAT Gateway or Load Balancer in the public subnet**.* • *Suitable for instances that do not need direct internet access or are part of an internal network, such as database servers, internal APIs, or backend services.*

a) *Launch 1st Instance with Public IPv4 Address*

- *At launching the Instance edit the network setting, select our custom VPC, then select my-subnet-1. Ensure that Auto-assign public IP is enabled. Create a security group allowing SSH and HTTP rules.*



Network settings Info

VPC - *required* Info
vpc-0c9987ae6c37efc1e (my-vpc)
192.168.0.0/24

Subnet Info
subnet-0282e827291496f71 my-subnet-1
VPC: vpc-0c9987ae6c37efc1e Owner: 971249973962 Availability Zone: us-east-1a
IP addresses available: 123 CIDR: 192.168.0.0/25)

Create new subnet

Auto-assign public IP Info
Enable

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.

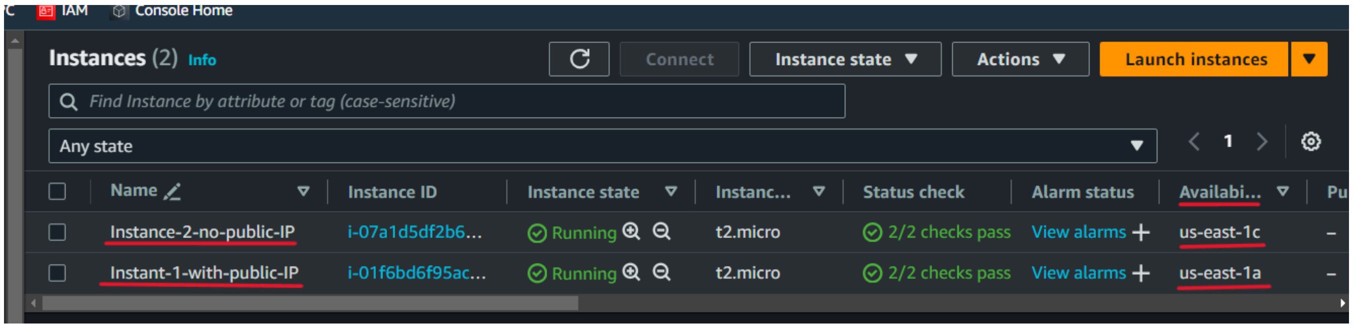
● Create security group ○ Select existing security group

Security group name - *required*
VPC-Security
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-5 created 2024-03-10T18:14:15.047Z

b) *Now launch 2nd Instance in a different subnet and disable Auto-assign Public IP Address. In the security group allow SSH and HTTP rules.*

We will see that both instances have launched.



Instances (2) Info

Name	Instance ID	Instance state	Instanc...	Status check	Alarm status	Availabi...	Pu
Instance-2-no-public-IP	i-07a1d5df2b6...	Running	t2.micro	2/2 checks pass	View alarms	us-east-1c	–
Instant-1-with-public-IP	i-01f6bd6f95ac...	Running	t2.micro	2/2 checks pass	View alarms	us-east-1a	–

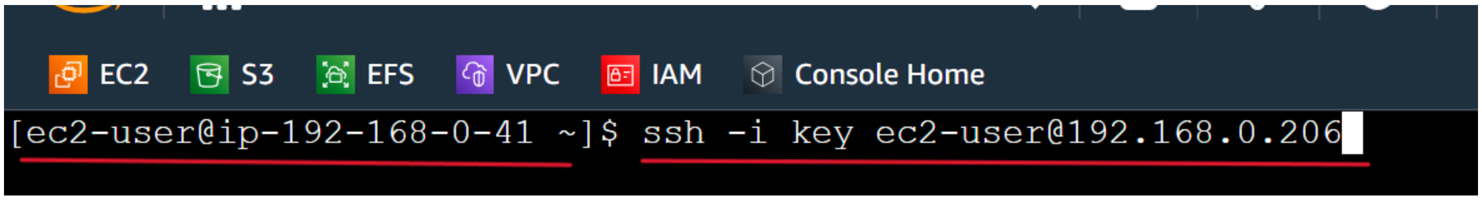
Step 13) *Now you can connect to the 1st Instance having a public IP, But the 2nd Instance does not have a Public IP we can't connect it using EC2 Connect.*

To Access 2nd Instance does not have a public by using the following Methods.

1) ***Using a Bastion Host or Jump Server***: *Connect to a separate instance with a public IP address (bastion host or jump server) in a public subnet, then use that instance to SSH into the target instance in the private subnet.*
2) ***Using Elastic IP (EIP):*** *Assign an Elastic IP (EIP) to the instance. This gives it a static public IP address, allowing direct access from the internet. However, this method is less secure because it exposes the instance directly to the internet.*
3) ***Using AWS Systems Manager Session Manager***: *Use AWS Systems Manager Session Manager, a service that provides secure shell (SSH) access to your instances without the need for a public IP address or opening inbound ports in security groups. This method requires the AWS Systems Manager agent to be installed on the instance.*
4) ***Using VPN or Direct Connect***: *Connect to your VPC using a VPN connection or AWS Direct Connect, then access the instance via its private IP address.*
5) ***Using NAT Gateway***: *Configure port forwarding on a NAT Gateway to allow SSH traffic from a specific port on the NAT Gateway's public IP address to the instance's private IP address.*
6) ***Using AWS Client VPN***: *Set up an AWS Client VPN endpoint to provide secure access to your VPC from remote clients. Once connected, access the instance via its private IP address.*

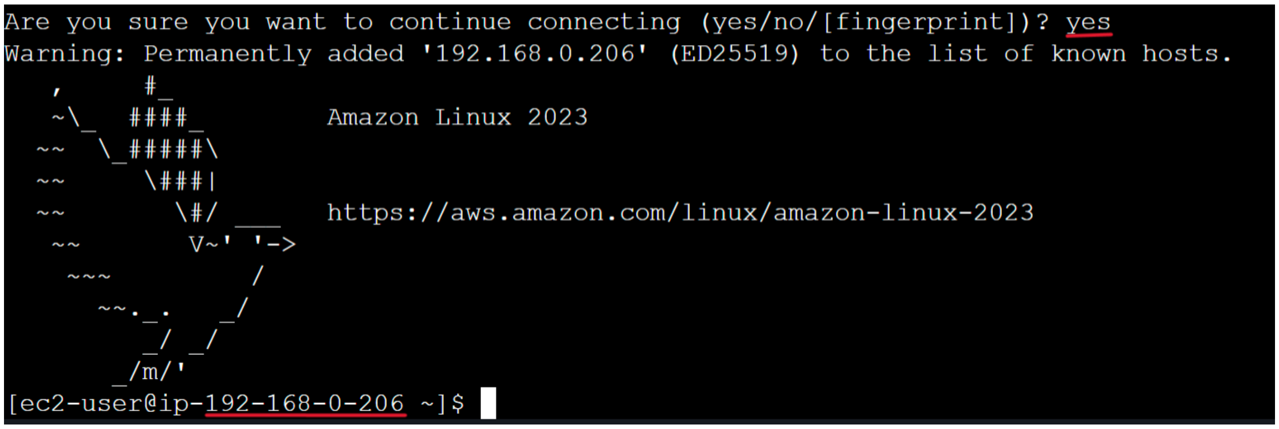
*Now we use **Bastion Host or Jump Sever** Method to access our 2nd Instance does not have a public IP address.*

- *Connect to the Instance have a public IP address.*
- *Open the private key file of the 2nd Instance in the notepad and copy all the text from it.*
- *Go to the connected instance CLI, create a new file, and paste the copied text of the key file into a new file then save it.*
- *The key is not publicly viewable is necessary. When we have created a key file in our Instance is default permission 644.*
- *Run the command **"chmod 400 key.file.name"** to set only read permission for the owner.*
- *Now copy or 2nd instance Private IPv4 address and run SSH client command on 1st instance CLI.*
- **ssh -i key.name ec2-user@192.168.0.206**



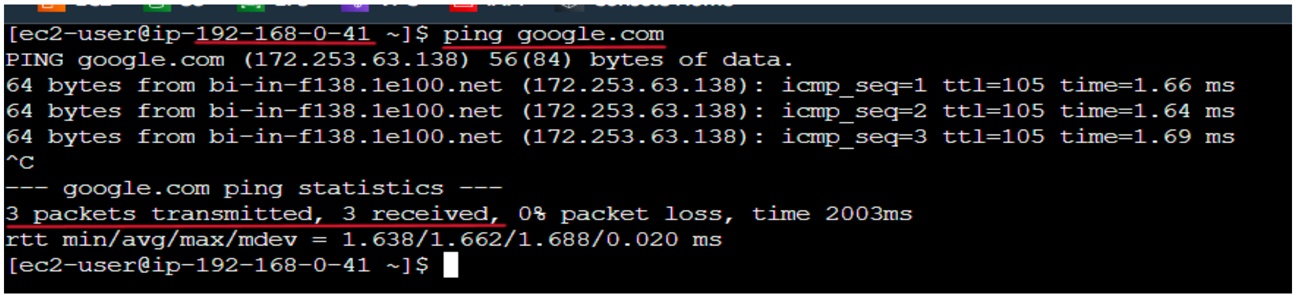
```
[ec2-user@ip-192-168-0-41 ~]$ ssh -i key ec2-user@192.168.0.206
```

192.168.0.41 is our instance having public IP and we are connecting our accessing to 192.168.0.206 Instance does not have a private IP address. Here we can see we connected to 2nd instance..



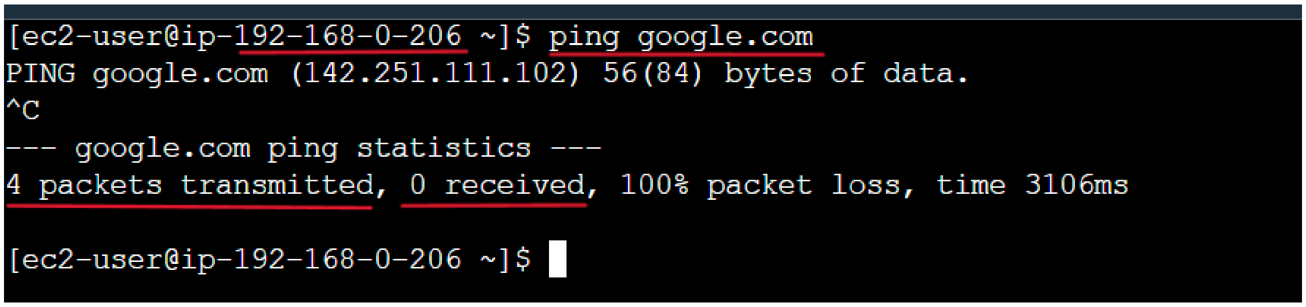
```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.206' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-192-168-0-206 ~]$
```

Stay connected with it, and open another tab to connect to 1st Instance again. Now we have access of both Instances but only 1st instance can access the internet and 2nd does not. As we can see in bellow Images.



```
[ec2-user@ip-192-168-0-41 ~]$ ping google.com
PING google.com (172.253.63.138) 56(84) bytes of data.
64 bytes from bi-in-f138.1e100.net (172.253.63.138): icmp_seq=1 ttl=105 time=1.66 ms
64 bytes from bi-in-f138.1e100.net (172.253.63.138): icmp_seq=2 ttl=105 time=1.64 ms
64 bytes from bi-in-f138.1e100.net (172.253.63.138): icmp_seq=3 ttl=105 time=1.69 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.638/1.662/1.688/0.020 ms
[ec2-user@ip-192-168-0-41 ~]$
```

In the above image, 1st instance can packets transmitted and received means accessing the internet



```
[ec2-user@ip-192-168-0-206 ~]$ ping google.com
PING google.com (142.251.111.102) 56(84) bytes of data.
^C
--- google.com ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3106ms

[ec2-user@ip-192-168-0-206 ~]$
```

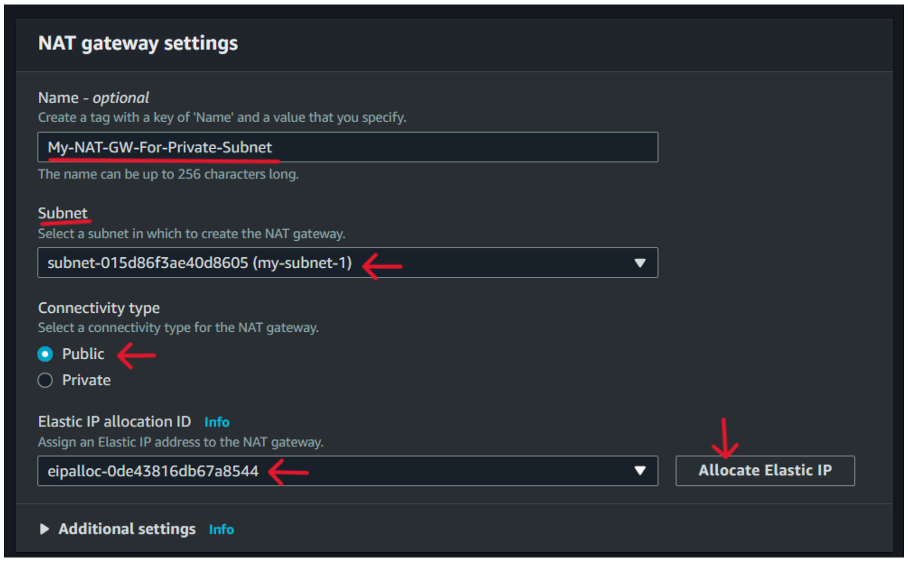
In this 2nd Instance, packets transmitted but not received means not accessing the internet.

This happens because we do not create a NAT gate for the private subnet where we created our 2nd Instance.

NAT Gateway provides a secure and scalable solution for instances in private subnets to access the internet while maintaining control over inbound traffic and ensuring the privacy and security of the network environment.

Step 14) Create a NAT Gateway.

- *In the VPC dashboard click on NAT Gateway, then click on the Create NAT gateway button.*
- *Name the NAT Gateway, select the Public subnet, and select the connectivity type as Public, (Creating a NAT Gateway in a public subnet, it uses the public subnet's routing table to route traffic to the internet. Instances in the private subnet are configured to send their outbound internet-bound traffic to the NAT Gateway)*
- *Allocate Elastic IP (This EIP provides the public IP address for the NAT Gateway. Placing the NAT Gateway in a public subnet ensures it has access to the internet and can communicate using this public IP address).*
- *Click on the Create NAT Gateway button.*


NAT gateway settings

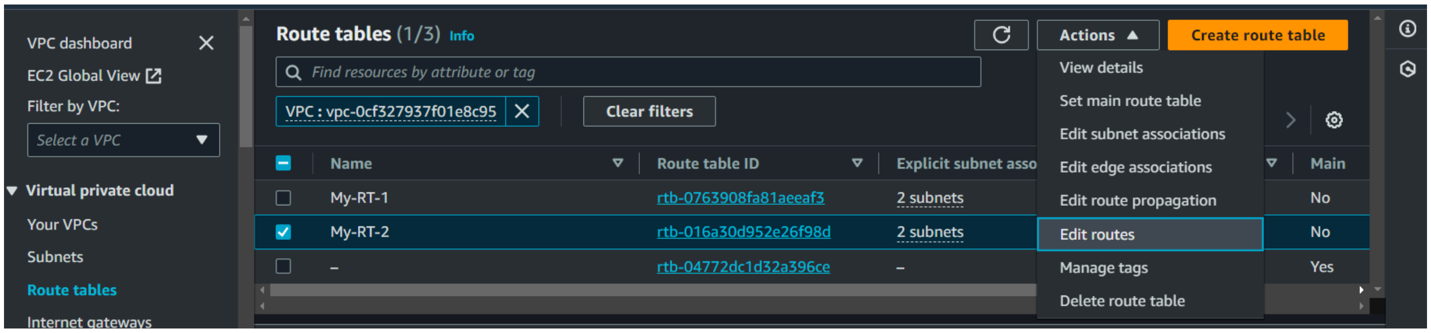
Name - optional
Create a tag with a key of 'Name' and a value that you specify.
My-NAT-GW-For-Private-Subnet
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-015d86f3ae40d8605 (my-subnet-1)

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-0de43816db67a8544

- *After creating the NAT gateway we need to add a route into the route table for internet access by setting up the target* ***NAT gateway***
- *Go to the route table, Select the route table, click on the Action button, then click on the edit route button.*

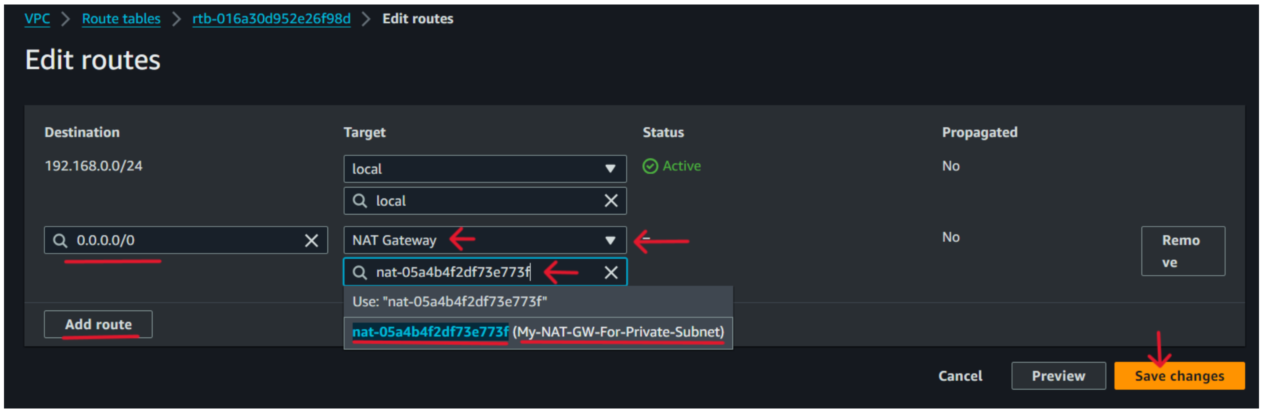


Route tables (1/3) Info

VPC : vpc-0cf327937f01e8c95

	Name	Route table ID	Explicit subnet asso	Main
☐	My-RT-1	rtb-0763908fa81aeeaf3	2 subnets	No
✓	My-RT-2	rtb-016a30d952e26f98d	2 subnets	No
☐	–	rtb-04772dc1d32a396ce	–	Yes

- *Click on the Add Route button, enter destination 0.0.0.0/0, select target NAT Gateway, then choose our NAT target, and click on Save Changes.*

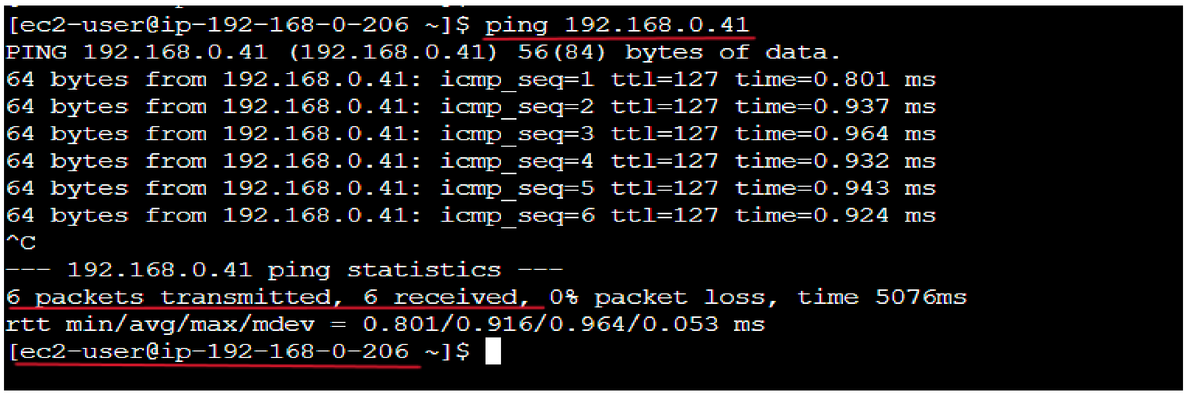

VPC > Route tables > rtb-016a30d952e26f98d > Edit routes

Edit routes

Destination	Target	Status	Propagated
192.168.0.0/24	local	Active	No
0.0.0.0/0	NAT Gateway	–	No

nat-05a4b4f2df73e773f
Use: "nat-05a4b4f2df73e773f"
nat-05a4b4f2df73e773f (My-NAT-GW-For-Private-Subnet)

Now you our 2nd Instance can access the internet. Ping google.com to verify.



```
[ec2-user@ip-192-168-0-206 ~]$ ping 192.168.0.41
PING 192.168.0.41 (192.168.0.41) 56(84) bytes of data.
64 bytes from 192.168.0.41: icmp_seq=1 ttl=127 time=0.801 ms
64 bytes from 192.168.0.41: icmp_seq=2 ttl=127 time=0.937 ms
64 bytes from 192.168.0.41: icmp_seq=3 ttl=127 time=0.964 ms
64 bytes from 192.168.0.41: icmp_seq=4 ttl=127 time=0.932 ms
64 bytes from 192.168.0.41: icmp_seq=5 ttl=127 time=0.943 ms
64 bytes from 192.168.0.41: icmp_seq=6 ttl=127 time=0.924 ms
^C
--- 192.168.0.41 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5076ms
rtt min/avg/max/mdev = 0.801/0.916/0.964/0.053 ms
[ec2-user@ip-192-168-0-206 ~]$
```

We can see it working.

***Step 15)* Add an ICMP IPv4 rule in the security groups of both Instances.**

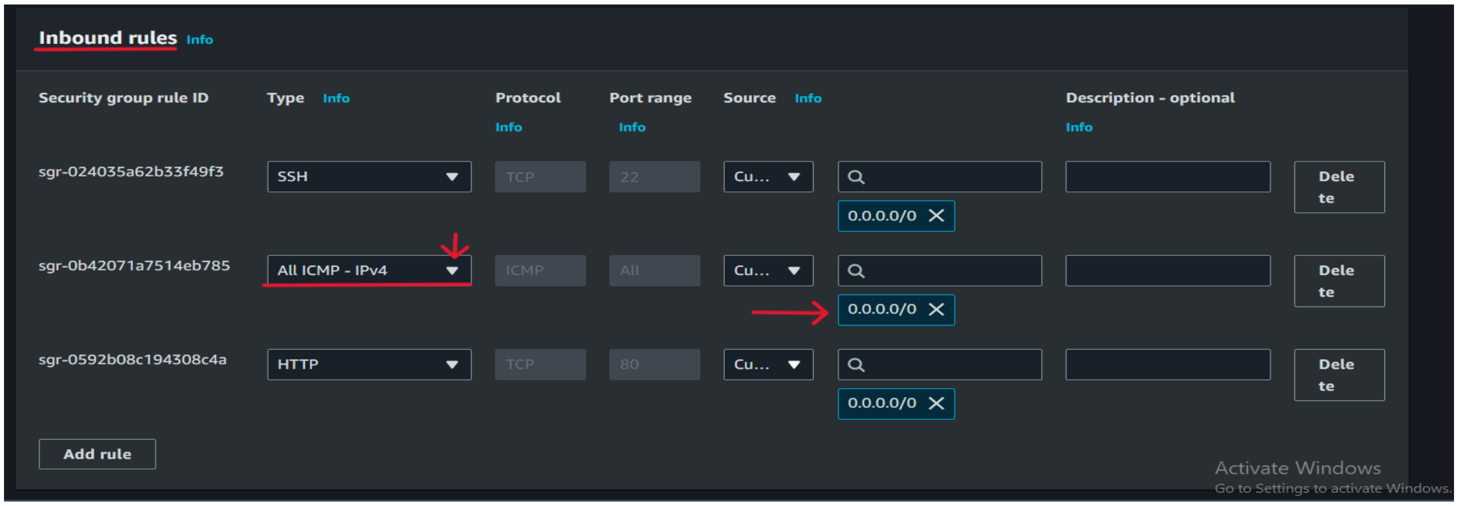
ICMP (Internet Control Message Protocol) is a network protocol used for various network diagnostics and error reporting purposes. It's commonly associated with IPv4, although it's also used in IPv6 networks.

Ping: The most common use of ICMP is for the ping command, which sends ICMP Echo Request packets to a destination host and waits for ICMP Echo Reply packets to confirm connectivity.

Firewall Considerations: ICMP traffic is often blocked by firewalls for security reasons. However, it's important to allow certain ICMP types, such as Echo Request/Reply, for network troubleshooting and monitoring.

Diagnostic Tool: ICMP is a crucial tool for network administrators for diagnosing network issues, such as determining whether a host is reachable, identifying routing problems, and troubleshooting network congestion.

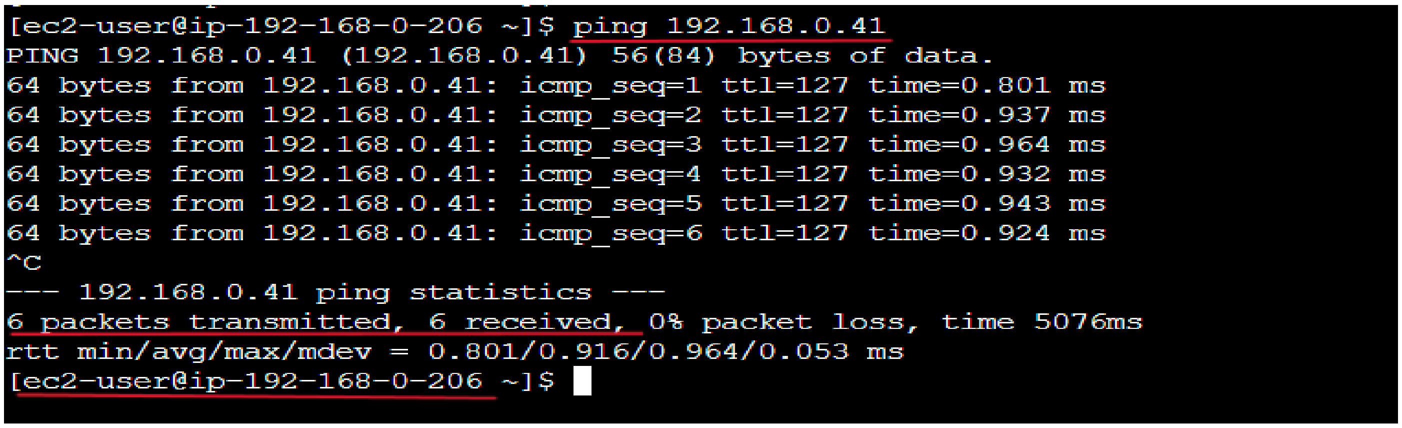
- *Edit Security rule, and add the inbound rule ICMP – Ipv4 and save the changes.*



Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-024035a62b33f49f3	SSH	TCP	22	Cu...	0.0.0.0/0		Delete
sgr-0b42071a7514eb785	All ICMP - IPv4	ICMP	All	Cu...	0.0.0.0/0		Delete
sgr-0592b08c194308c4a	HTTP	TCP	80	Cu...	0.0.0.0/0		Delete

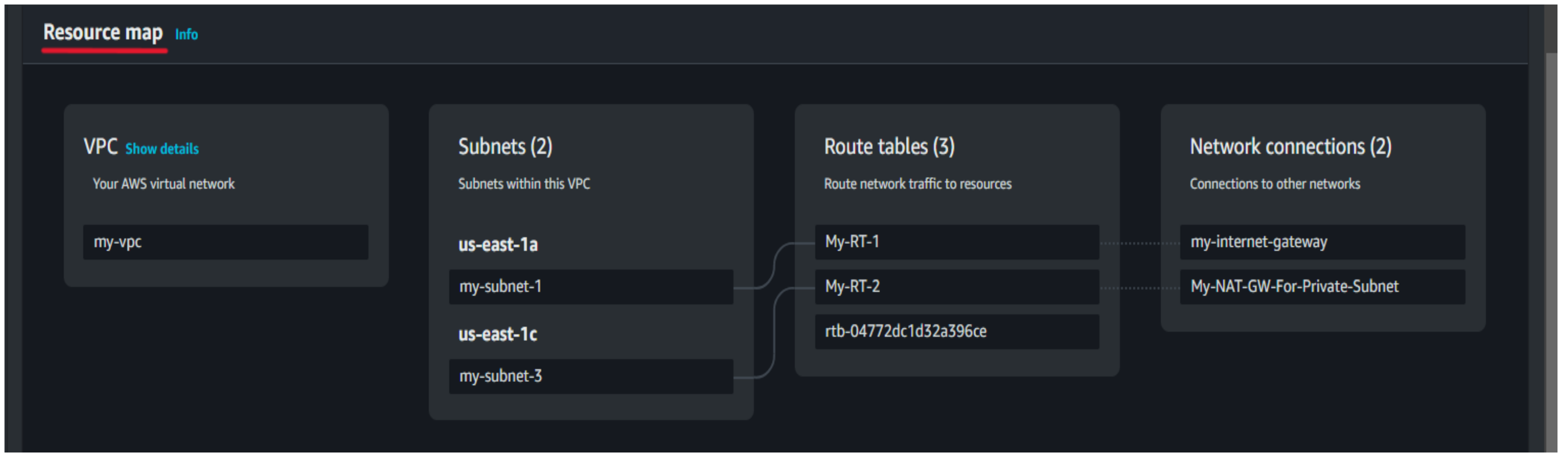
Now we ping a public Instance from a private Instance using its Public IP address.



```
[ec2-user@ip-192-168-0-206 ~]$ ping 192.168.0.41
PING 192.168.0.41 (192.168.0.41) 56(84) bytes of data.
64 bytes from 192.168.0.41: icmp_seq=1 ttl=127 time=0.801 ms
64 bytes from 192.168.0.41: icmp_seq=2 ttl=127 time=0.937 ms
64 bytes from 192.168.0.41: icmp_seq=3 ttl=127 time=0.964 ms
64 bytes from 192.168.0.41: icmp_seq=4 ttl=127 time=0.932 ms
64 bytes from 192.168.0.41: icmp_seq=5 ttl=127 time=0.943 ms
64 bytes from 192.168.0.41: icmp_seq=6 ttl=127 time=0.924 ms
^C
--- 192.168.0.41 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5076ms
rtt min/avg/max/mdev = 0.801/0.916/0.964/0.053 ms
[ec2-user@ip-192-168-0-206 ~]$
```

It is working.

Resource Map of our custom VPC


Resource map
Info
VPC
Show details
Your AWS virtual network
my-vpc
Subnets (2)
Subnets within this VPC
us-east-1a
my-subnet-1
us-east-1c
my-subnet-3
Route tables (3)
Route network traffic to resources
My-RT-1
My-RT-2
rtb-04772dc1d32a396ce
Network connections (2)
Connections to other networks
my-internet-gateway
My-NAT-GW-For-Private-Subnet