Create a Custom VPC with Subnets

Step 1 – Create a VPC

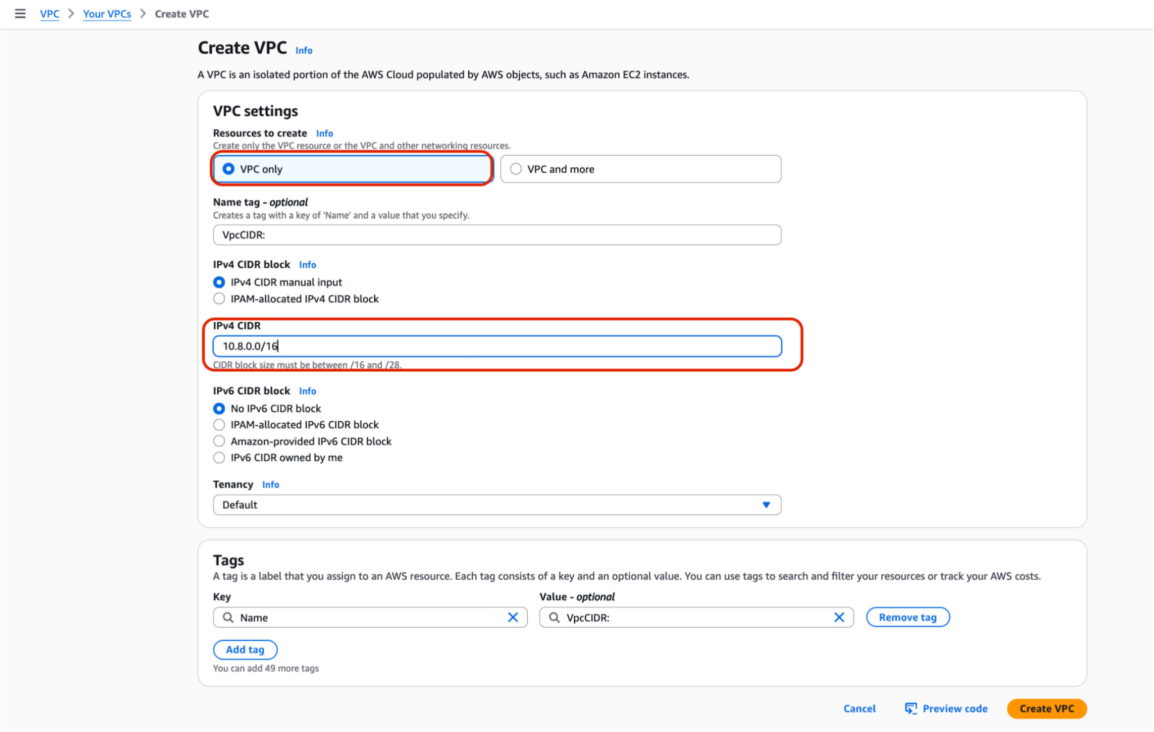
1. In the AWS console, search for **VPC**.

2. Click **Create VPC**.

3. Enter a name for your VPC.

4. For the **IPv4 CIDR block**, use `10.8.0.0/16`.

5. Keep the tenancy set to **default** (do not select dedicated, as it will cost extra).



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.
VpcCIDR:

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

IPv4 CIDR
10.8.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
● No IPv6 CIDR block
○ IPAM-allocated IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me

Tenancy Info
Default

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	VpcCIDR:

You can add 49 more tags

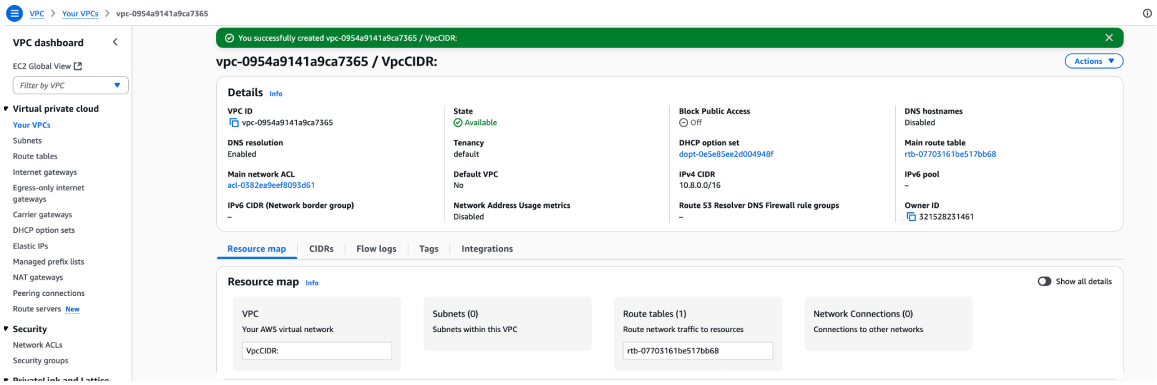
Enable DNS Hostnames

Once your VPC is created:

- Go to **Actions → Edit VPC settings**.

- Enable **DNS hostnames**.

This ensures that EC2 instances launched in the VPC automatically receive DNS hostnames.



You successfully created vpc-0954a9141a9ca7365 / VpcCIDR:

vpc-0954a9141a9ca7365 / VpcCIDR:

Details

VPC ID
vpc-0954a9141a9ca7365

State
Available

Block Public Access
Off

DNS hostnames
Disabled

DNS resolution
Enabled

Tenancy
default

DHCP option set
dopt-0e5e85ee2d004948f

Main route table
rtb-07703161be517bb68

Main network ACL
acl-0382ea9eef8093d61

Default VPC
No

IPv4 CIDR
10.8.0.0/16

IPv6 pool
–

IPv6 CIDR (Network border group)
–

Network Address Usage metrics
Disabled

Route 53 Resolver DNS Firewall rule groups
–

Owner ID
321528231461

Resource map

VPC
Your AWS virtual network
VpcCIDR:

Subnets (0)
Subnets within this VPC

Route tables (1)
Route network traffic to resources
rtb-07703161be517bb68

Network Connections (0)
Connections to other networks

Step 2 – Create Subnets

1. Go to **VPC → Subnets → Create subnet**.

2. Select the VPC you created.

3. For the first subnet:

   - **Name**: `Public-1A`

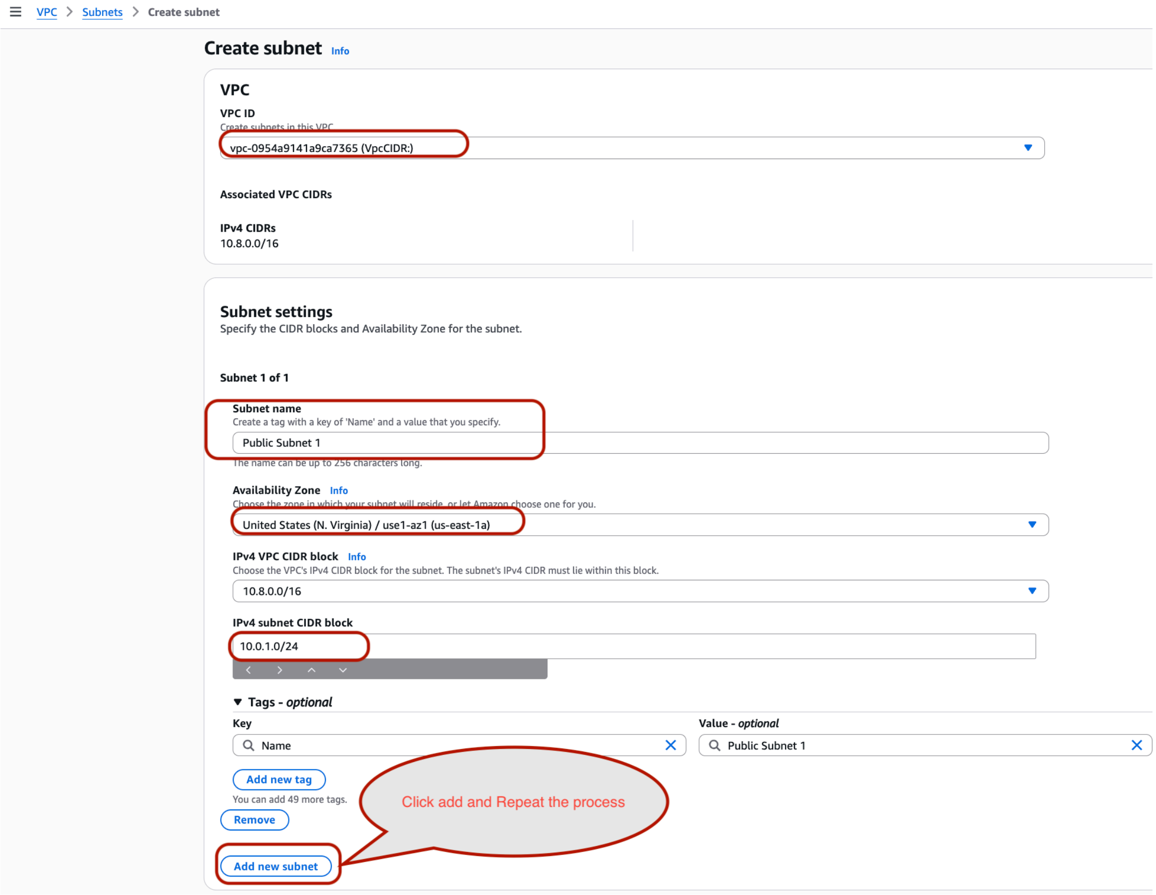
   - **Availability Zone**: `us-east-1a`

   - **IPv4 CIDR Block**: `10.0.1.0/24`

4. Repeat for:

   - **Private Subnet 1**

   - **Private Subnet 2**



Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-0954a9141a9ca7365 (VpcCIDR:)

Associated VPC CIDRs

IPv4 CIDRs
10.8.0.0/16

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.
Public 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.
United States (N. Virginia) / use1-az1 (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.
10.8.0.0/16

IPv4 subnet CIDR block
10.0.1.0/24

▼ Tags - *optional*

Key	Value - *optional*
Name	Public Subnet 1

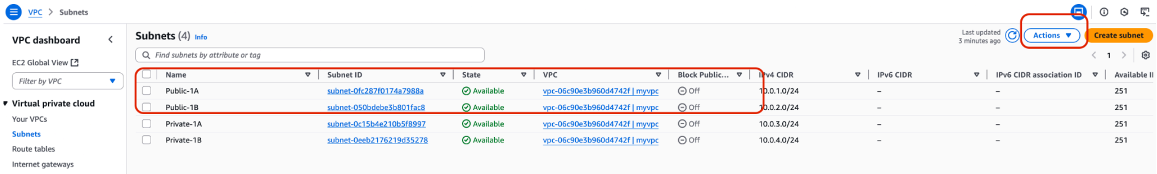
Click add and Repeat the process

Step 3 – Configure Subnets

Enable Auto-Assign IP Addresses

- Go to **Actions → Modify auto-assign IP settings**.

- Enable **Auto-assign IPv4 address** for both public subnets (`Public-1A` and `Public-1B`).

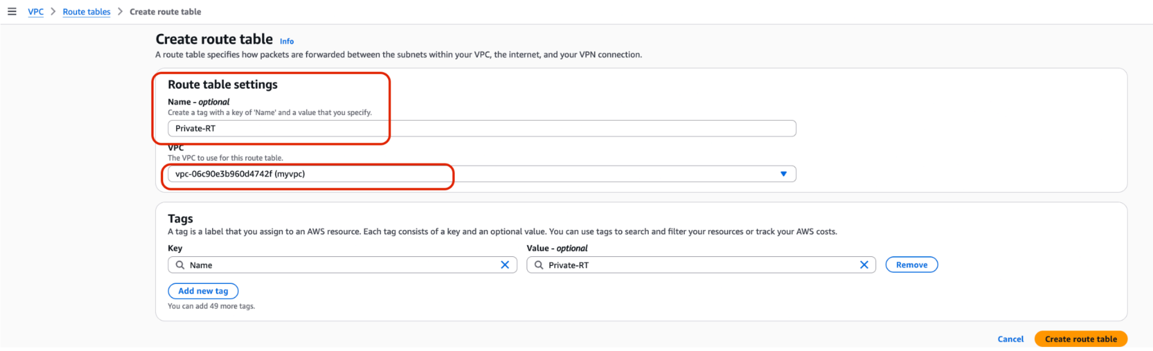


Subnets (4) Info

□	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID	Available I
□	Public-1A	subnet-0fc287f0174a7988a	Available	vpc-06c90e3b960d4742f | myvpc	Off	10.0.1.0/24	–	–	251
□	Public-1B	subnet-050bdebe3b801fac8	Available	vpc-06c90e3b960d4742f | myvpc	Off	10.0.2.0/24	–	–	251
□	Private-1A	subnet-0c15b4e210b5f8997	Available	vpc-06c90e3b960d4742f | myvpc	Off	10.0.3.0/24	–	–	251
□	Private-1B	subnet-0eeb2176219d35278	Available	vpc-06c90e3b960d4742f | myvpc	Off	10.0.4.0/24	–	–	251

1. ### Create Route Tables for Private Subnets
2. 1. Go to **Route Tables**.
3. 2. Click **Create route table** → give it a name (e.g., `Private-RT-1A`).
4. 3. Associate with your VPC.
5. 4. Repeat for `Private-RT-1B`.

6. 5. Associate each private route table with its subnet.



Create route table Info

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

Route table settings

Name - *optional*

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

Private-RT

VPC

The VPC to use for this route table.

vpc-06c90e3b960d4742f (myvpc)

Tags

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

Key	Value - *optional*
Name	Private-RT

You can add 49 more tags.

Associate Private Subnets with the Route Table

After creating the route table:

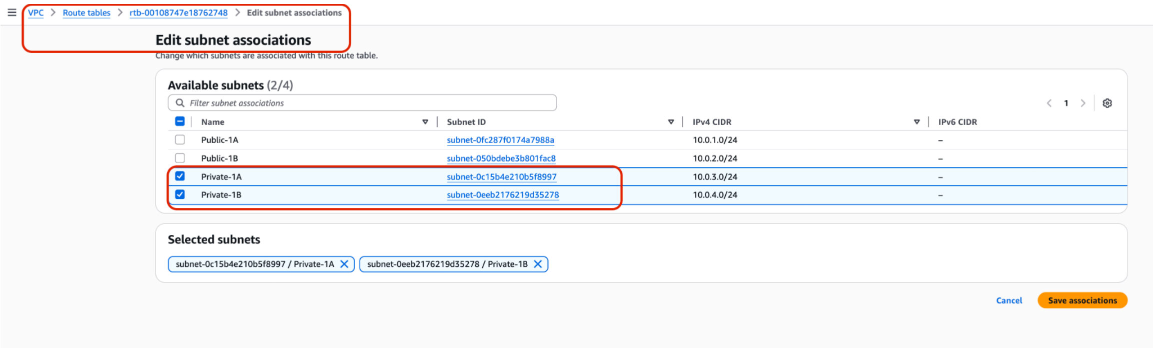
1. Open the route table and go to the Subnet associations tab.

2. Click Edit subnet associations.

3. Select the private subnets (e.g., Private-1A and Private-1B).

4. Click Save.

This links your private subnets to their dedicated route table.



VPC > Route tables > rtb-00108747e18762748 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/4)

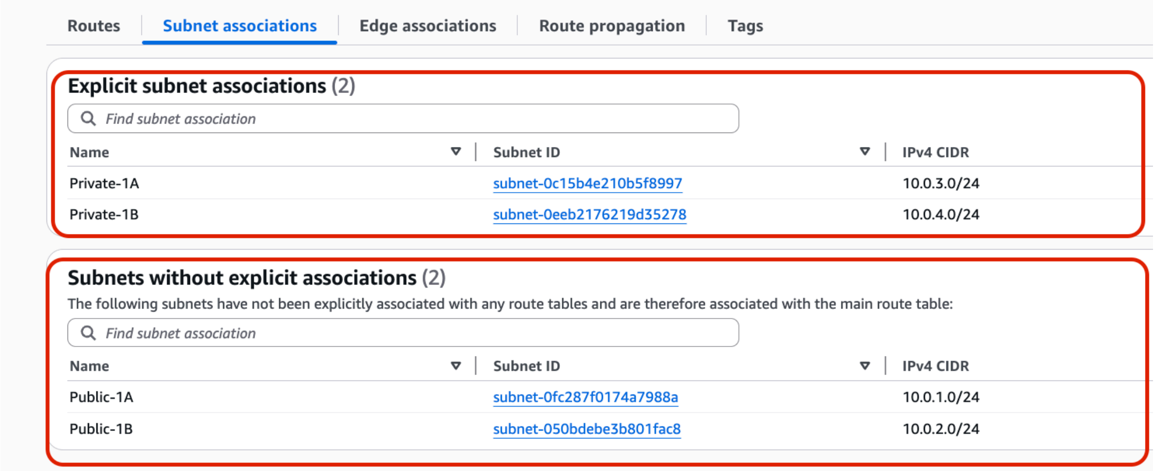
Filter subnet associations
1

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
□	Public-1A	subnet-0fc287f0174a7988a	10.0.1.0/24	–
□	Public-1B	subnet-050bdebe3b801fac8	10.0.2.0/24	–
☑	Private-1A	subnet-0c15b4e210b5f8997	10.0.3.0/24	–
☑	Private-1B	subnet-0eeb2176219d35278	10.0.4.0/24	–

Selected subnets

subnet-0c15b4e210b5f8997 / Private-1A

subnet-0eeb2176219d35278 / Private-1B



Explicit subnet associations (2)

Find subnet association

Name	Subnet ID	IPv4 CIDR
Private-1A	subnet-0c15b4e210b5f8997	10.0.3.0/24
Private-1B	subnet-0eeb2176219d35278	10.0.4.0/24

Subnets without explicit associations (2)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Find subnet association

Name	Subnet ID	IPv4 CIDR
Public-1A	subnet-0fc287f0174a7988a	10.0.1.0/24
Public-1B	subnet-050bdebe3b801fac8	10.0.2.0/24

Create and Attach an Internet Gateway

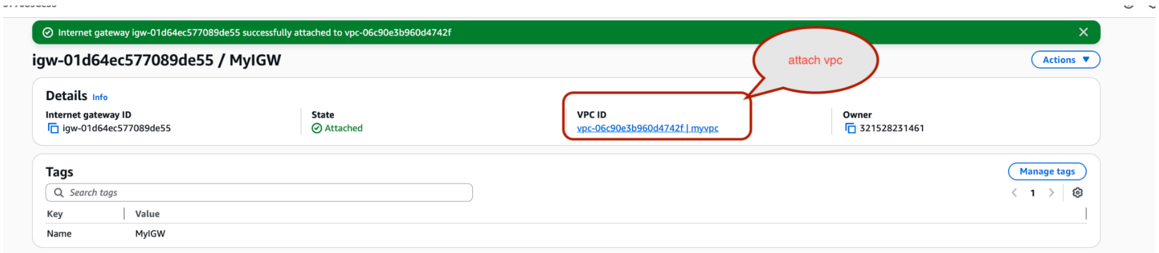
1. **In the VPC console, go to Internet Gateways.**

2. **Click Create internet gateway and give it a name.**

3. **After creation, select the internet gateway and click Attach to VPC.**

4. **Choose the VPC you created earlier and attach it.**

This allows your public subnets to access the internet.



Internet gateway igw-01d64ec577089de55 successfully attached to vpc-06c90e3b960d4742f

igw-01d64ec577089de55 / MyIGW

Details

Internet gateway ID: igw-01d64ec577089de55

State: Attached

VPC ID: vpc-06c90e3b960d4742f | myvpc

Owner: 321528231461

Tags

Key	Value
Name	MyIGW

Update Route Table for Public Subnets

1. **Open the public route table in the VPC console.**

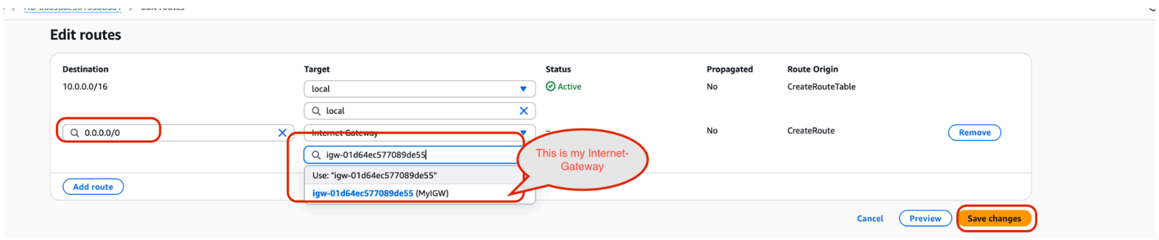
2. **Go to the Routes tab and click Edit routes.**

3. **Add a new route:**

- ○ **Destination: 0.0.0.0/0**
- ○ **Target: Select the Internet Gateway (IGW) you created.**

4. **Click Save changes.**

This ensures that resources in your public subnets can reach the internet.



Edit routes

Destination	Target	Status	Propagated	Route Origin	
10.0.0.0/16	local	Active	No	CreateRouteTable	
0.0.0.0/0	~~Internet Gateway~~	-	No	CreateRoute	Remove

local

igw-01d64ec577089de55

Use: "igw-01d64ec577089de55"

igw-01d64ec577089de55 (MyIGW)

This is my Internet-Gateway

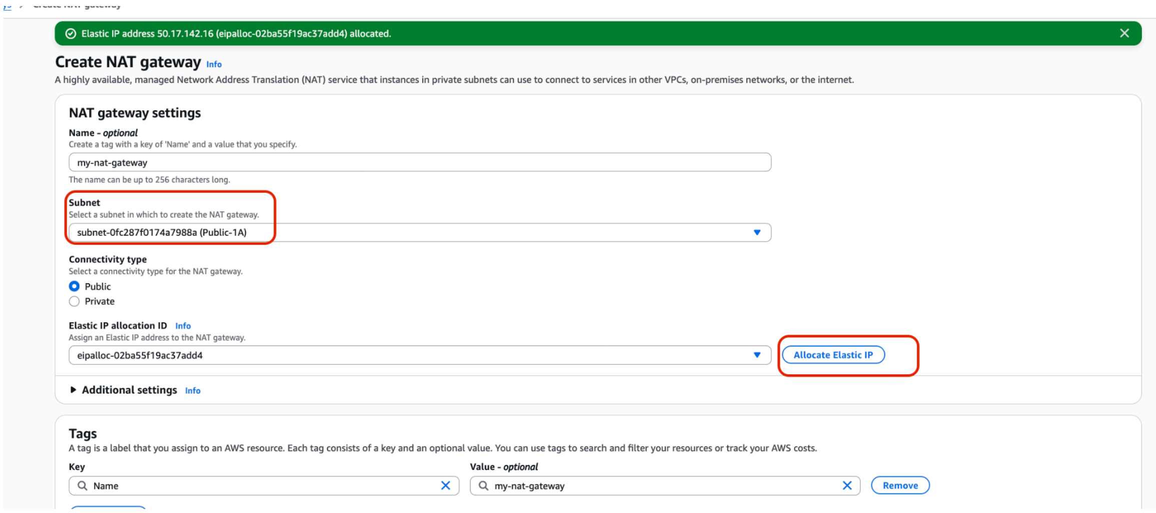
Launch Instances and Configure NAT Gateway

To test connectivity between subnets, you'll set up a NAT Gateway for your private subnets:

1. **Launch Test Instances**
   - ○ **Deploy one EC2 instance in a public subnet and another in a private subnet.**
   - ○ **This will help verify connectivity later.**
2. **Create a NAT Gateway**
   - ○ **Go to NAT Gateways in the VPC console.**
   - ○ **Click Create NAT Gateway.**
   - ○ **Place it in a public subnet.**
   - ○ **Allocate and attach an Elastic IP address.**
   - ○ **Save the configuration.**
3. **Update Private Route Table**
   - ○ **Open the route table for your private subnets (Private-RT).**

- Scroll to the Routes tab and click Edit routes.
- Add a route:
  - Destination: 0.0.0.0/0
  - Target: Select the NAT Gateway you just created (e.g., nat-07ff85de1c63c6d80).
- Save changes.

This setup ensures that instances in private subnets can access the internet through the NAT Gateway while remaining unreachable from outside.



Elastic IP address 50.17.142.16 (eipalloc-02ba55f19ac37add4) allocated.

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
my-nat-gateway
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-0fc287f0174a7988a (Public-1A)

Connectivity type
Select a connectivity type for the NAT gateway.
● Public
○ Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-02ba55f19ac37add4

Additional settings Info

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	my-nat-gateway

Create a Security Group

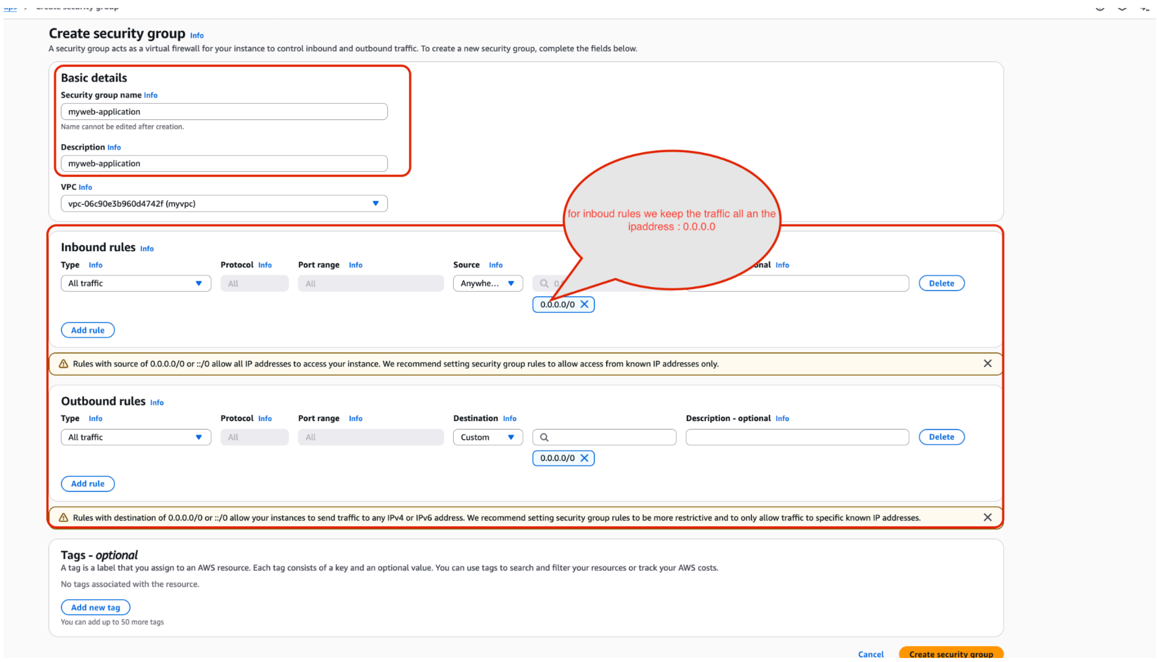
After setting up the NAT Gateway, the next step is to create a security group for your instances:

1. In the VPC or EC2 console, go to Security Groups.
2. Click Create security group.
3. Enter a name and description (e.g., Web-SG or Private-SG).
4. Select the VPC you created earlier.

5. **Configure inbound/outbound rules as needed (for example, allow SSH, HTTP, or ICMP).**

6. **Save the security group.**

You can then attach this security group to your test instances.



Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info: myweb-application

Name cannot be edited after creation.

Description Info: myweb-application

VPC Info: vpc-06c90e3b960d4742f (myvpc)

for inboud rules we keep the traffic all an the ipaddress : 0.0.0.0

Inbound rules Info

Type	Protocol	Port range	Source		…onal
All traffic	All	All	Anywhe...	0.0.0.0/0	

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Outbound rules Info

Type	Protocol	Port range	Destination		Description - optional
All traffic	All	All	Custom	0.0.0.0/0	

Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.

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.

No tags associated with the resource.

You can add up to 50 more tags

Launch Test Instances

To validate your VPC setup, launch EC2 instances in the same region where your VPC was created (for example, N. Virginia):

1. **Go to the EC2 console and click Launch instance.**

2. **Choose an Amazon Machine Image (AMI), such as Amazon Linux 2.**

3. **Select an instance type (e.g., t2.micro for testing).**

4. **Under Network settings:**

   ○ **Select the VPC you created.**

   ○ **Choose either a public subnet or a private subnet depending on the test.**

   ○ **Attach the appropriate security group.**

5. **If launching into a public subnet, enable Auto-assign Public IP.**

6. **Review and launch the instance.**

This will allow you to test connectivity between your public and private subnets, as well as internet access through the IGW (public) and NAT Gateway (private).