Build a Virtual Private Cloud (VPC)

By Haroon Zaman | November 2025

Introduction

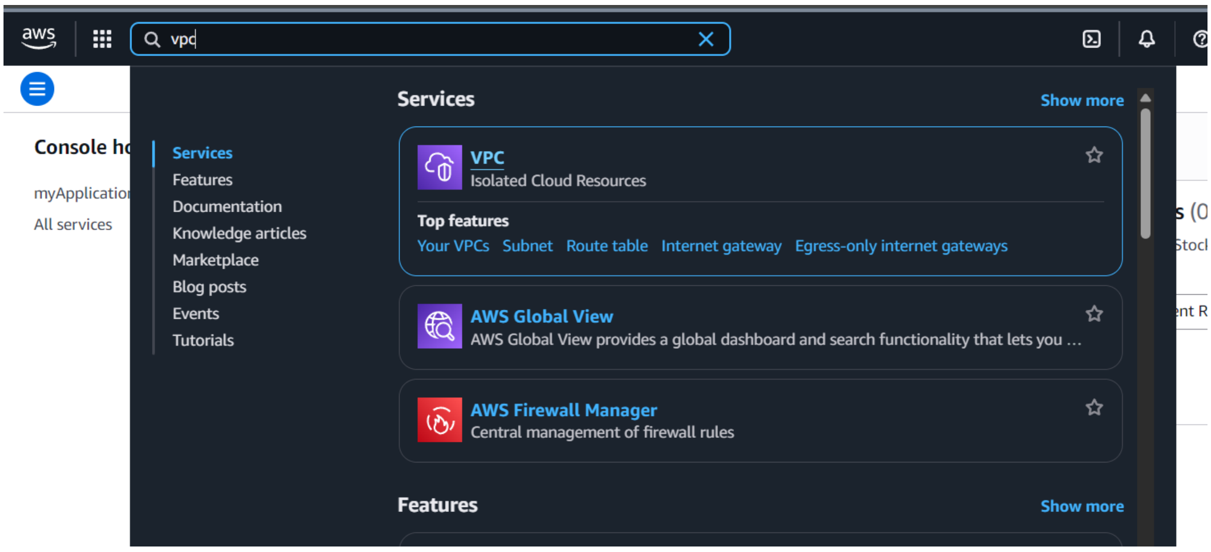
I am doing this project to learn the fundamentals of **AWS networking** and how to design a secure cloud environment. By building a VPC, I aim to understand how cloud resources communicate, how networking works in AWS, and how to create the foundation for more advanced cloud architectures.

In this project, I performed the following tasks:

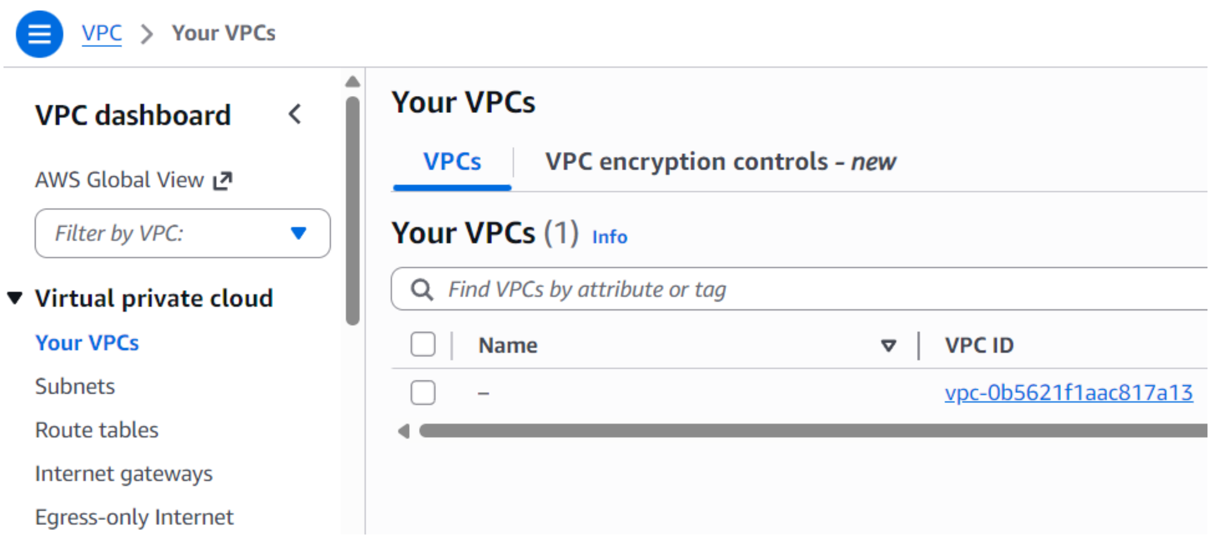
1. **Created a VPC** – I set up my own private network in AWS, defining an IPv4 CIDR block to control IP ranges and ensure proper communication between resources.
2. **Created Subnets** – I divided my VPC into subnets, which act like neighborhoods to organize resources and manage traffic flow.
3. **Created an Internet Gateway** – I attached an internet gateway to my VPC, allowing my resources to connect to and be accessed from the internet securely.

Virtual Private Clouds (VPCs)

A VPC (Virtual Private Cloud) is a private network inside AWS where we control IP ranges, subnets, routing, and security. It allows us to securely host resources like EC2 instances and databases while managing how they communicate with each other.



First, I logged in to my AWS account. I typed **VPC** in the top search bar and clicked on it to open the VPC dashboard, where I could start creating my own private network.


VPC > Your VPCs

VPC dashboard

AWS Global View

Filter by VPC:

Virtual private cloud
- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only Internet

Your VPCs

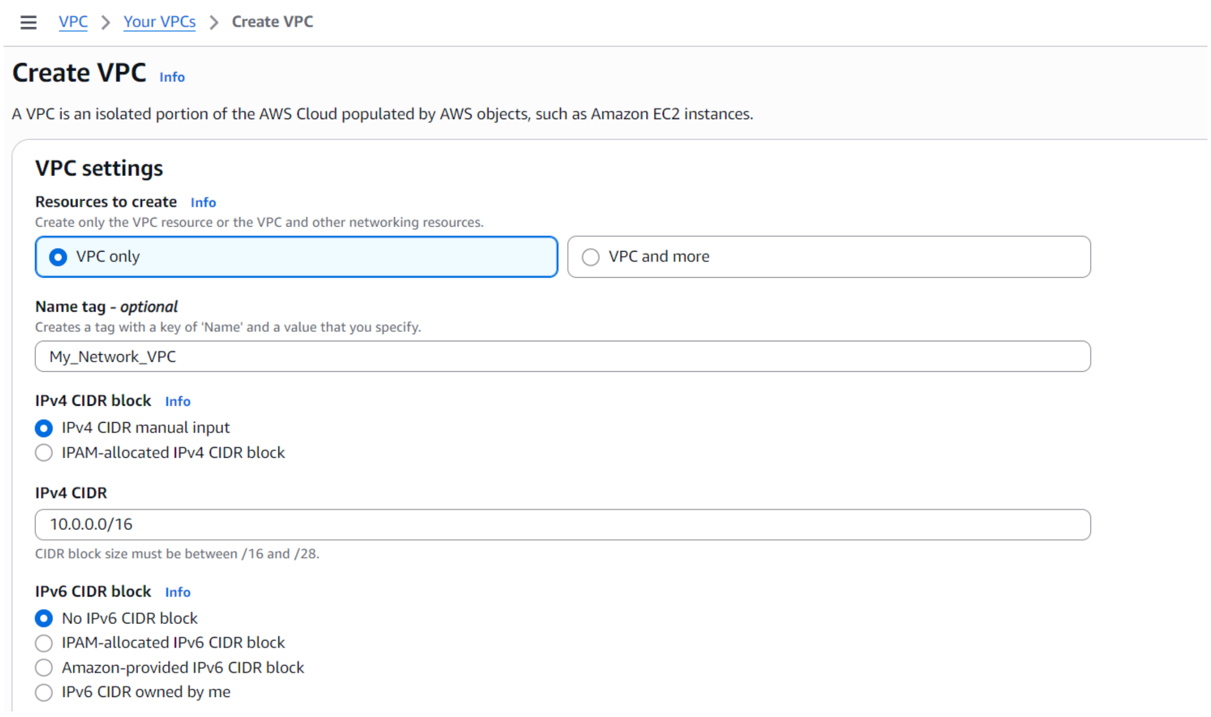
VPCs | VPC encryption controls - *new*

Your VPCs (1) Info

Find VPCs by attribute or tag

Name	VPC ID
–	vpc-0b5621f1aac817a13

In the VPC dashboard, I selected **Your VPCs**, which shows a list of VPCs in my account. There was already one **default VPC**, created by AWS for quick resource launching. Then, I clicked **Create VPC** to start making my own.


VPC > Your VPCs > Create VPC

Create VPC Info

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

VPC settings

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

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

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

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

IPv6 CIDR block Info
[✓ No IPv6 CIDR block]
IPAM-allocated IPv6 CIDR block
Amazon-provided IPv6 CIDR block
IPv6 CIDR owned by me

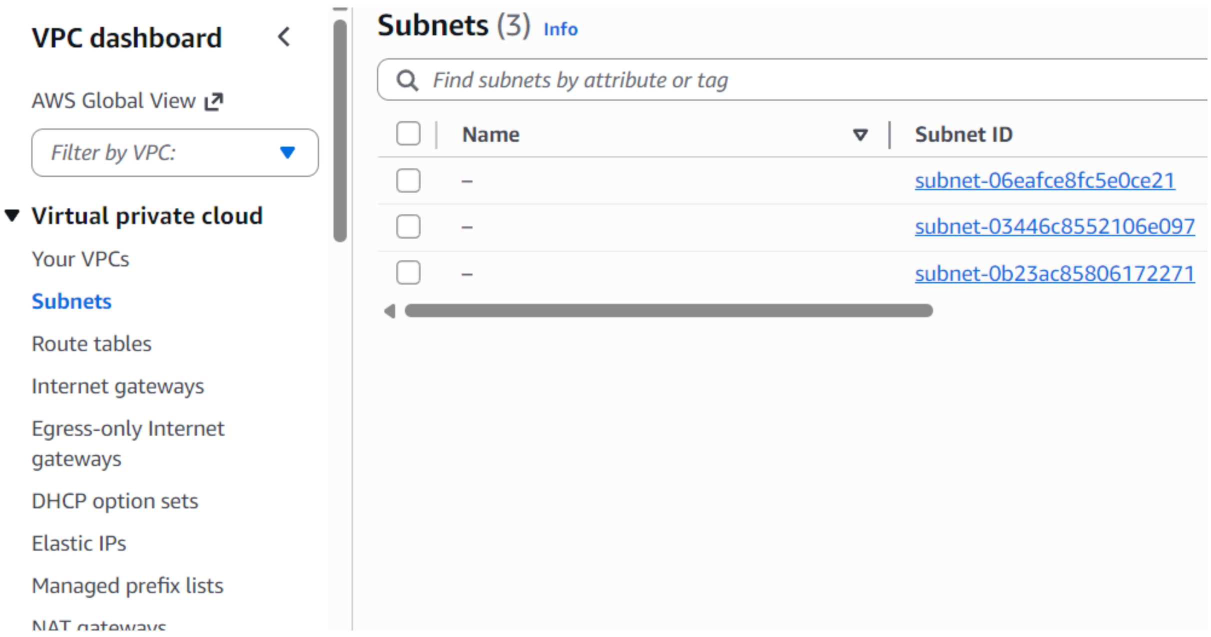
When the **Create VPC** page opened, I selected **VPC only** and named it **MY_Network_VPC**. I set the IPv4 CIDR block to **10.0.0.0/16**, giving me a large range of IP addresses for my network. I left all other settings at default and clicked **Create VPC**.

I chose /16 because it provides up to 65,536 IP addresses, enough to divide into multiple subnets for future resources.

Subnets

Subnets are smaller sections inside a VPC that organize resources and manage traffic. In my account, some subnets already existed because the default VPC includes one subnet for each Availability Zone in the Region.

To continue, I created my own subnet within **MY_Network_VPC** to plan where my resources will live and operate.

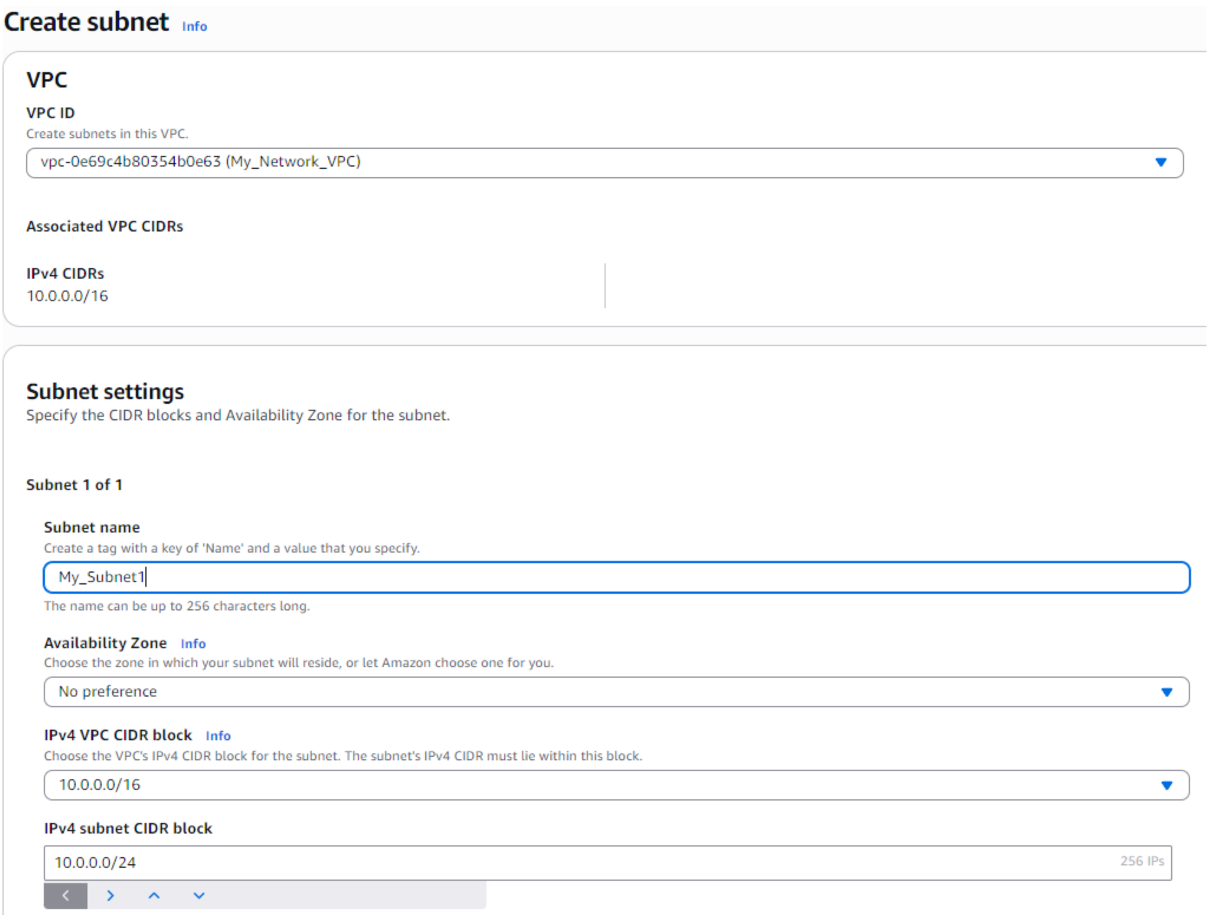


Name	Subnet ID
–	subnet-06eafce8fc5e0ce21
–	subnet-03446c8552106e097
–	subnet-0b23ac85806172271

On the VPC dashboard, I selected **Subnets** from the left panel. By default, there were already three subnets because the default VPC creates one subnet for each Availability Zone in the Region. Then, I clicked **Create Subnet** on the right to add my own.

Here is what I did to create a subnet:

- On the **Create Subnet** panel, the first step was to select **in which VPC** my subnet should be created. I chose **MY_Network_VPC** because that is the VPC I just created.
- Next, I gave a **name** to my subnet: **My_Subnet1**, so I could easily identify it later.
- For **Availability Zone**, I learned that Availability Zones (AZs) are separate data centers in a region that provide high availability and fault tolerance for resources. There were other options to choose a specific AZ, but I selected **No Preference**, which means AWS will automatically place my subnet in any AZ in the region.
- The **IPv4 VPC CIDR block** was already populated as **10.0.0.0/16**, which is the IP range of my VPC. I used /16 because it allows up to 65,536 IP addresses to divide across multiple subnets.
- I had to fill the **IPv4 Subnet CIDR block**, and I chose **10.0.0.0/24**, which creates 256 IP addresses for this subnet. I chose /24 because it provides enough addresses for the resources I plan to place here without wasting IPs.
- Finally, I left all other settings at **default** and clicked **Create Subnet** to complete the setup.



Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-0e69c4b80354b0e63 (My_Network_VPC)

Associated VPC CIDRs

IPv4 CIDRs
10.0.0.0/16

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

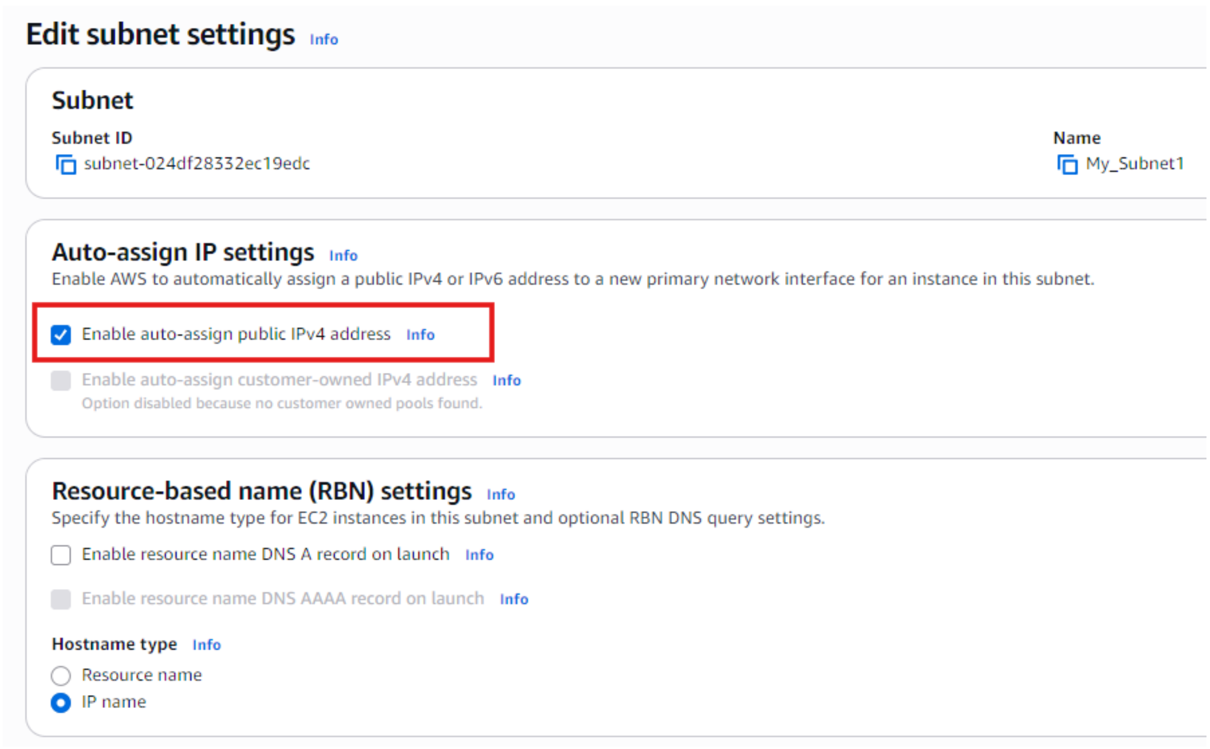
Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
My_Subnet1
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.
No preference

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.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.0.0/24 256 IPs



Edit subnet settings Info

Subnet

Subnet ID
subnet-024df28332ec19edc

Name
My_Subnet1

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.

Resource-based name (RBN) settings Info
Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

Enable resource name DNS A record on launch Info

Enable resource name DNS AAAA record on launch Info

Hostname type Info
Resource name [✓ IP name]

- After creating the subnet, I needed to **edit it**.
- I selected the subnet I had just created, **My_Subnet1**, from the list.
- On the right side, I clicked **Actions**, which opened a drop-down menu.

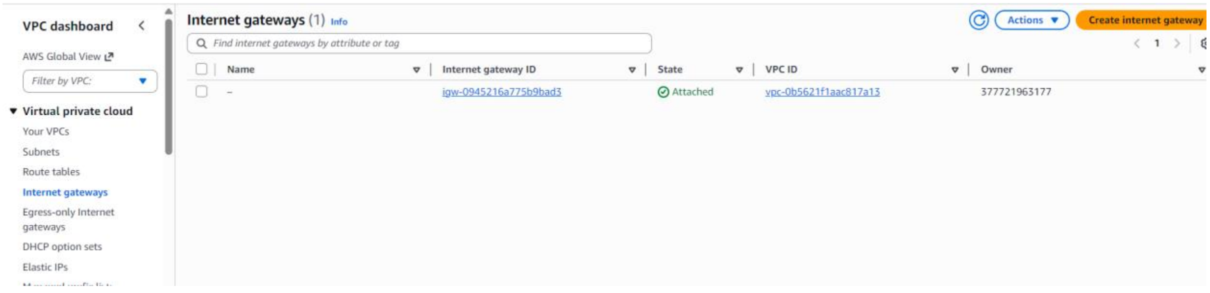
- From the drop-down, I selected **Edit Subnet** to modify its settings.
- In the **Edit Subnet** window, I selected the option **Enable auto-assign public IPv4 address** under **Auto-assign IP settings**.
- I did this because enabling it automatically gives any EC2 instance launched in this subnet a **public IP**, allowing it to access the internet without manual configuration.
- After selecting this option, I clicked **Save** to apply the changes.

Internet Gateway

Internet gateways are AWS components that let resources in a VPC communicate with the internet. They act like a bridge connecting my private network to the outside world, enabling both outbound and inbound traffic.

In this step, I will create and attach an internet gateway to **MY_Network_VPC** so my resources can access the internet.

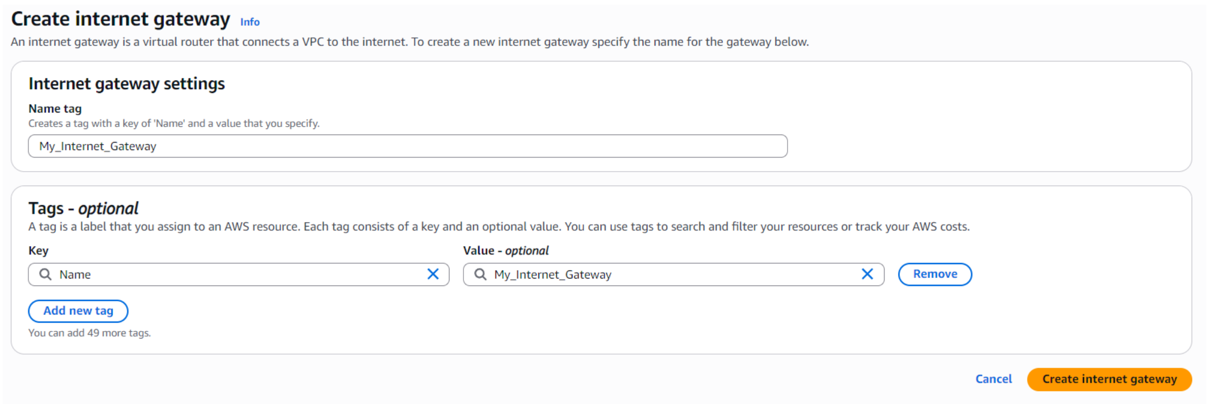
- On the VPC dashboard, I selected **Internet Gateways** from the left panel.
- I noticed there was already a **default internet gateway** created by AWS for the default VPC.
- Even so, I decided to **create my own internet gateway** for **MY_Network_VPC** to connect my custom network to the internet.



Internet gateways (1) Info

Find internet gateways by attribute or tag

Name	Internet gateway ID	State	VPC ID	Owner
–	igw-0945216a775b9bad3	Attached	vpc-0b5621f1aac817a13	377721963177



Create internet gateway Info

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.

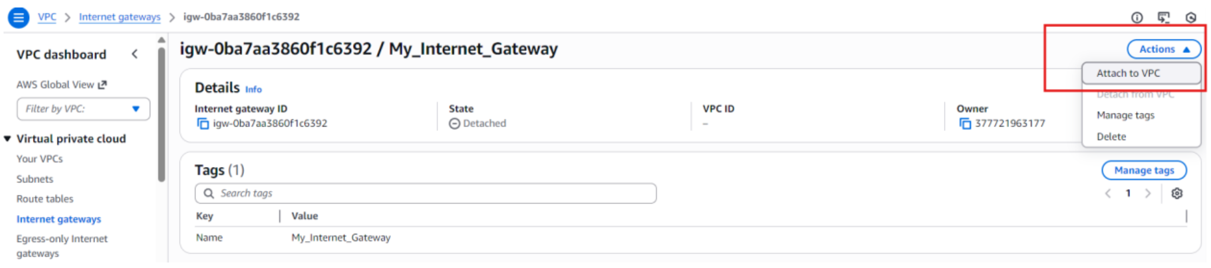
Internet gateway settings

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

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*
Name	My_Internet_Gateway

- In the **Create Internet Gateway** page, I saw that only a **name** was required.
- I entered **MY_Internet_Gateway** as the name to identify it easily later.
- After naming it, I clicked **Create Internet Gateway** to complete the creation.


VPC > Internet gateways > igw-0ba7aa3860f1c6392

igw-0ba7aa3860f1c6392 / My_Internet_Gateway

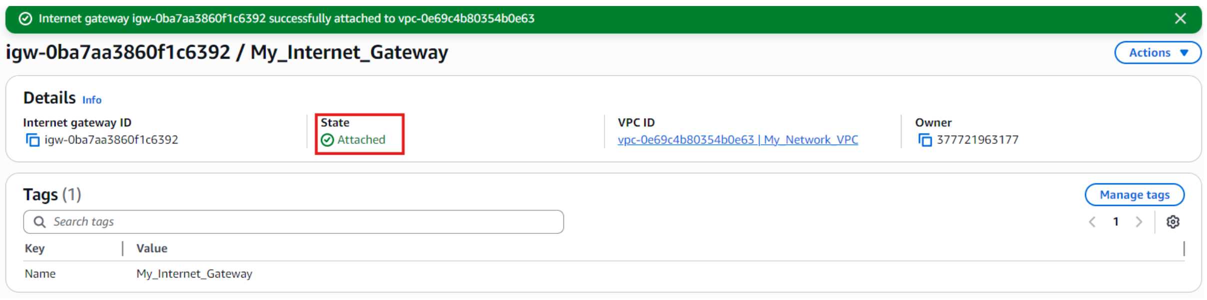
Details Info

Internet gateway ID	State	VPC ID	Owner
igw-0ba7aa3860f1c6392	Detached	–	377721963177

Tags (1)

Key	Value
Name	My_Internet_Gateway

- After creating **MY_Internet_Gateway**, I noticed it was not yet attached to my VPC.
- On the upper right corner, I clicked **Actions** and selected **Attach to VPC**.
- In the next window, I selected **MY_Network_VPC** and clicked **Save** to attach the internet gateway to my VPC.


Internet gateway igw-0ba7aa3860f1c6392 successfully attached to vpc-0e69c4b80354b0e63

igw-0ba7aa3860f1c6392 / My_Internet_Gateway

Details Info

Internet gateway ID	State	VPC ID	Owner
igw-0ba7aa3860f1c6392	Attached	vpc-0e69c4b80354b0e63 | My_Network_VPC	377721963177

Tags (1)

Key	Value
Name	My_Internet_Gateway

- After clicking **Save**, I went back to the dashboard and saw that **MY_Internet_Gateway** now showed **"Attached"** in the state field.
- This confirmed that my internet gateway was successfully connected to **MY_Network_VPC**, allowing resources in my VPC to communicate with the internet.

Conclusion

In this project, I created my own **VPC**, added a **subnet**, and attached an **internet gateway** in AWS. Through this process, I learned how to design a private network, organize resources, and enable secure internet access for cloud instances.

This project was important because it helped me understand the **foundations of AWS networking**, including IP addressing, subnets, and internet connectivity. These skills are essential for building scalable and secure cloud environments, which will be useful for more advanced cloud architectures in the future.