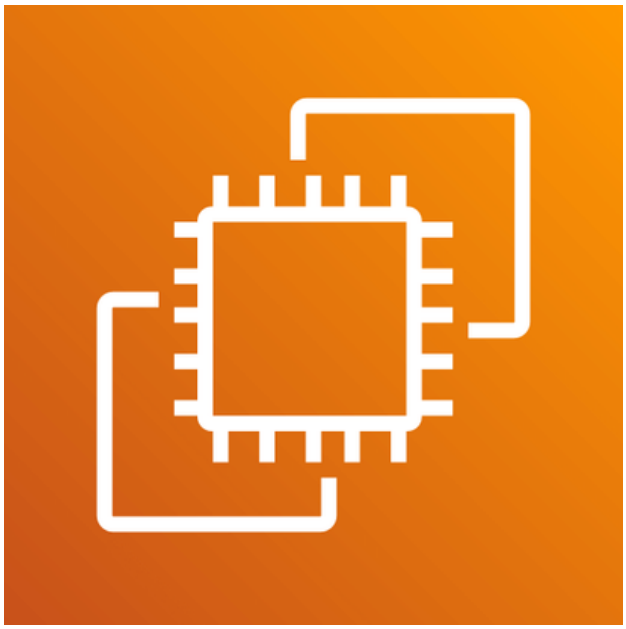


# AWS EC2

## & Security Groups

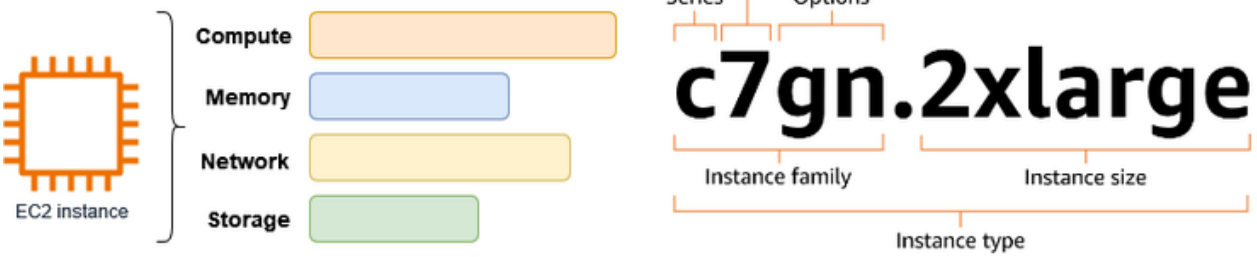
# Concept Overview:

# About  EC2:

Amazon EC2 is a cloud service that provides on-demand, scalable virtual servers.

You can quickly launch, scale up for high demand, and scale down when not needed, saving costs.

Each EC2 instance is a virtual server, and its instance type decides the amount of CPU, memory, storage, and networking it gets.



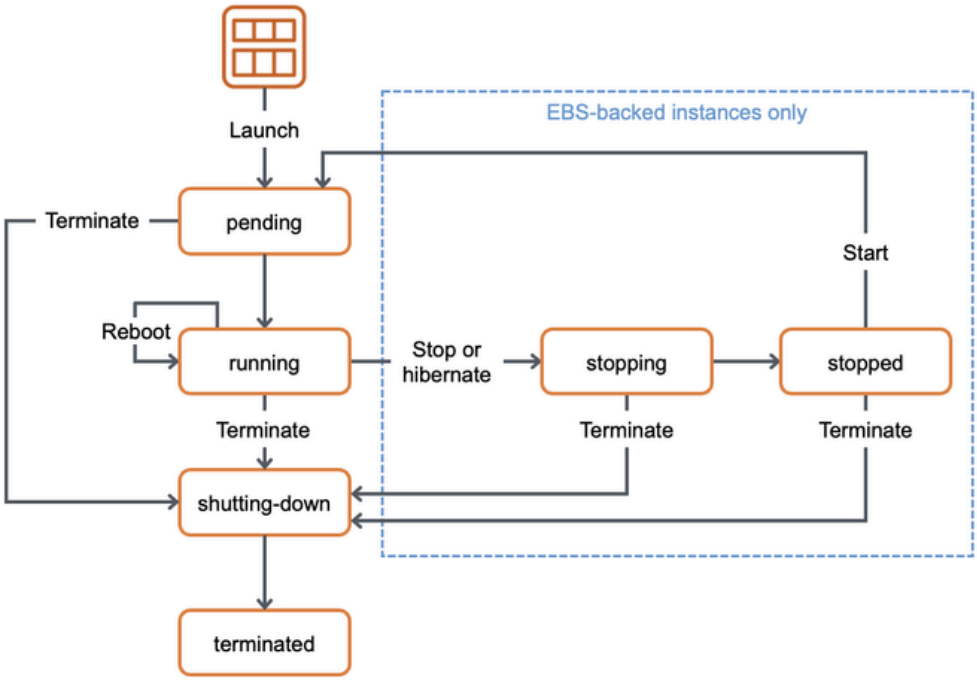
## EC2 Features:

- Instances: Virtual servers.
- Amazon Machine Images (AMIs): Preconfigured templates for your instances that package the components you need for your server (including the operating system and additional software).
- Instance types: Various configurations of CPU, memory, storage, networking capacity, and graphics hardware for your instances.
- Amazon EBS volumes: Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS).
- Instance store volumes: Storage volumes for temporary data that is deleted when you stop, hibernate, or terminate your instance.
- Key pairs: Secure login information for your instances. AWS stores the public key and you store the private key in a secure place.
- Security groups: A virtual firewall that allows you to specify the protocols, ports, and source IP ranges that can reach your instances, and the destination IP ranges to which your instances can connect.

# EC2 Lifecycle:

- Pending: The instance is being provisioned and prepared for launch.
- Running: The instance is operational and actively serving your applications.
- Stopping: The instance is undergoing a graceful shutdown process.
- Stopped: The instance is powered off, but its resources remain allocated.
- Shutting down: The instance is preparing to be terminated.
- Terminated: The instance is permanently deleted, and its resources are released.

**For better understanding, follow the diagram:**



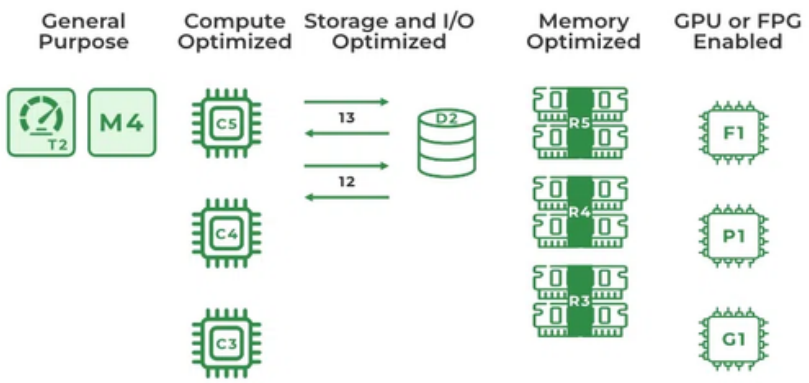
# EC2 Instance Types:

**The AWS EC2 Instance Types are as follows:**

1. General Purpose Instances
2. Compute Optimized Instances
3. Memory-Optimized Instances
4. Storage Optimized Instances
5. Accelerated Computing Instances

**Most common used instance type:**

- General Purpose (T3, T4g, M5, M6): Most popular for balanced workloads.
- Compute Optimized (C5, C6): Popular for CPU-intensive tasks.
- Memory Optimized (R5, R6): Popular for databases & big in-memory apps.




For more details about EC2 instance types, you can follow the AWS official docs.

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html

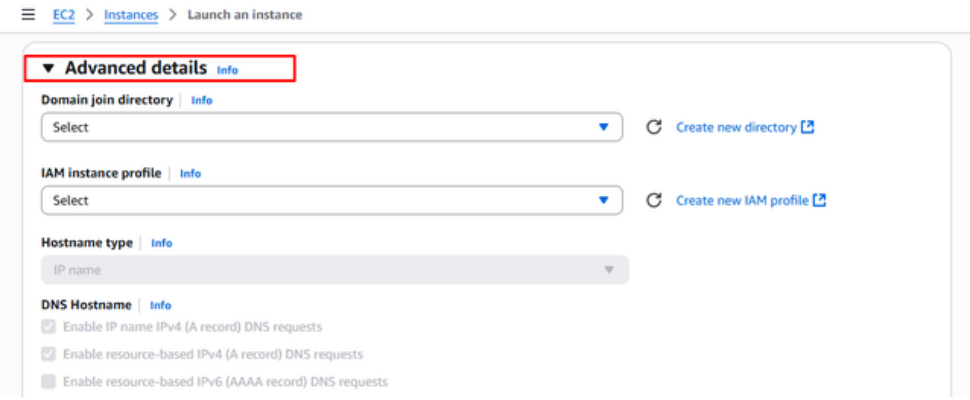
# User Data:

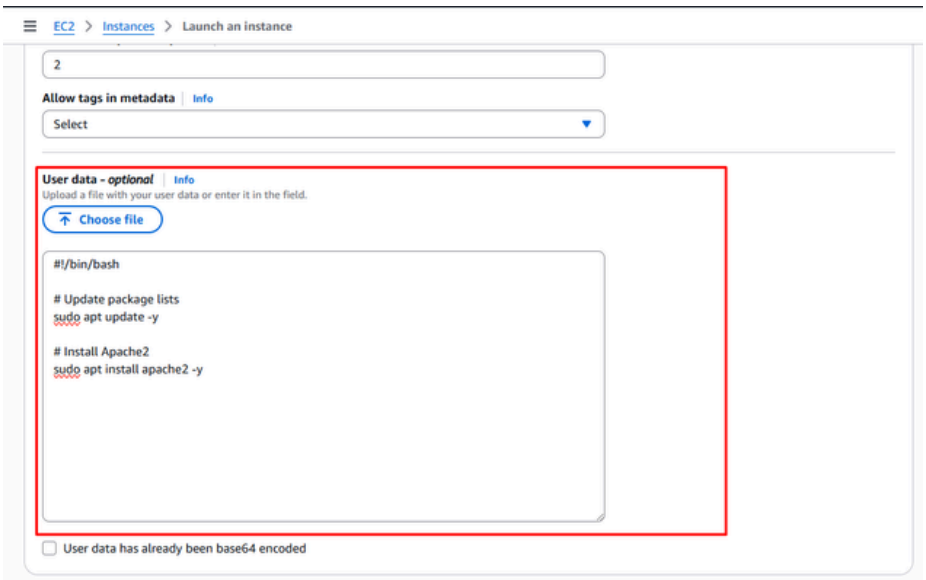
AWS user data is a feature that allows the execution of scripts or commands on an Amazon EC2 instance when it is launched for the first time.

- This script is executed with a user with root privileges
- You can install updates, packages, change configurations and anything you can do with scripts can be done.

User Data helps you to Automation of your tasks.

You find it in Advance Details tabs when you create a new instance.



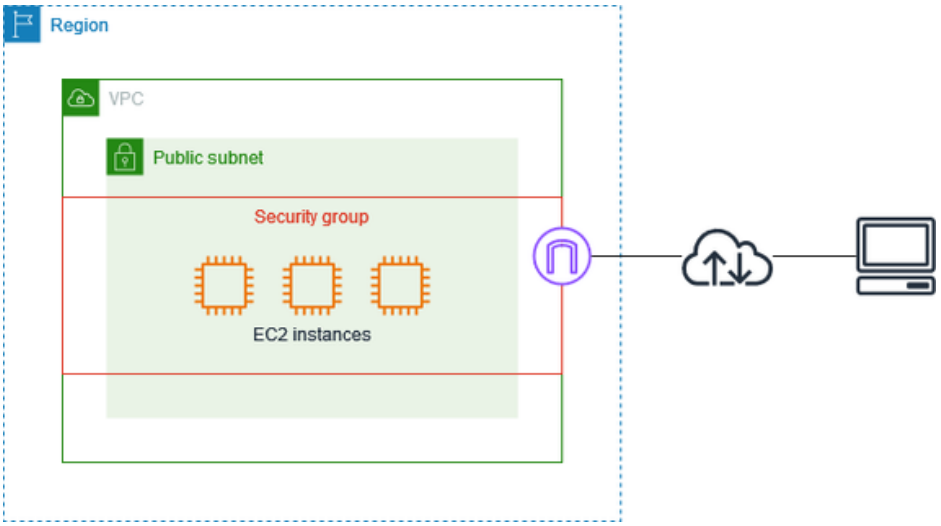



# Security Groups:

A security group acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic.

Inbound rules control the incoming traffic to your instance, and outbound rules control the outgoing traffic from your instance. When you launch an instance, you can specify one or more security groups.

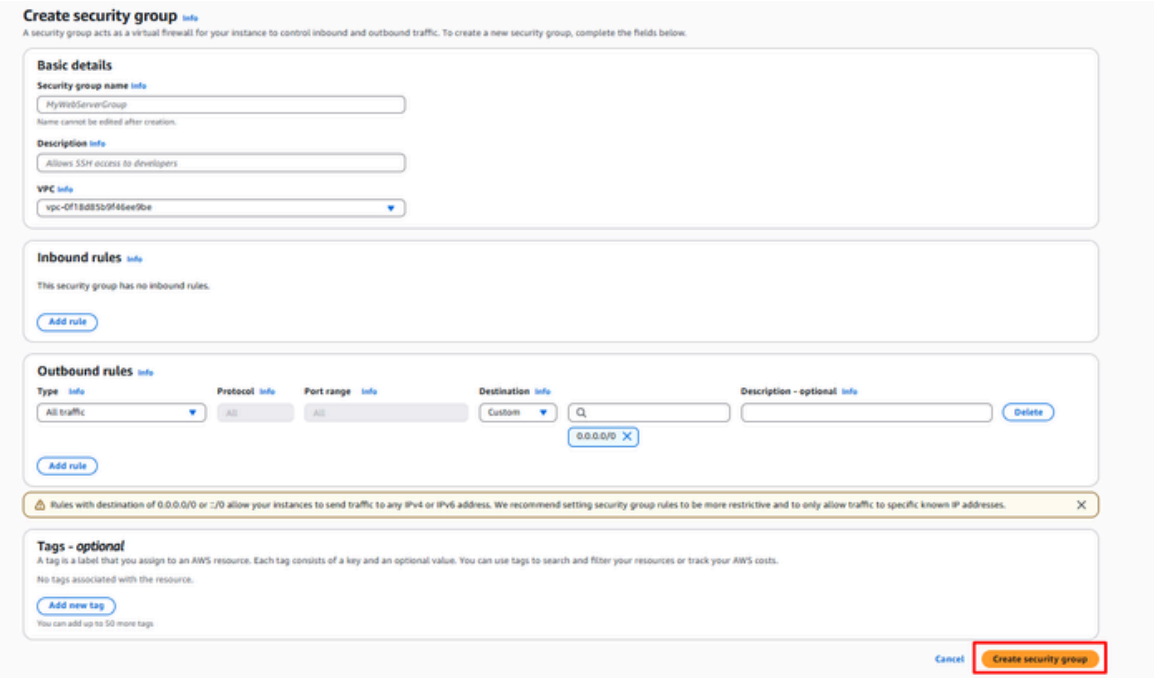
If you don't specify a security group, Amazon EC2 uses the default security group for the VPC. After you launch an instance, you can change its security groups.



# Security Groups:

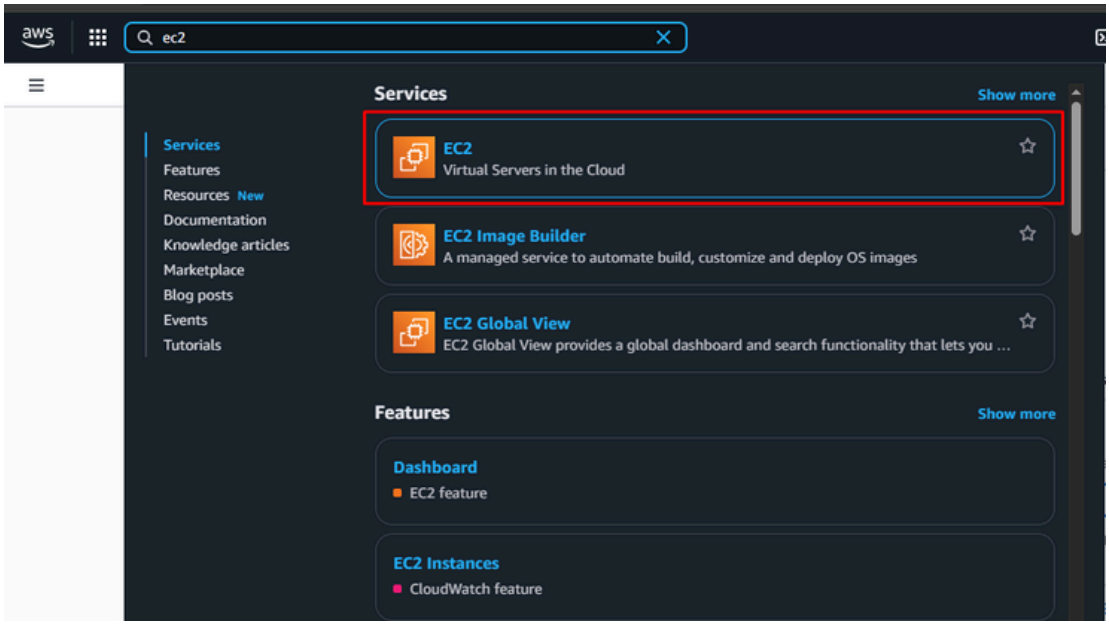
For create a security group you need to go security group section from EC2 dashboard.


Fill up the input fields based on your needs and click on Create Security Group.

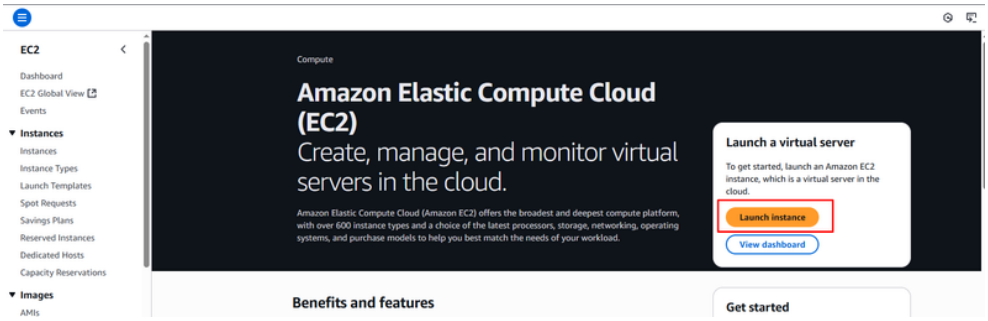


# Launch First EC2 Instance:

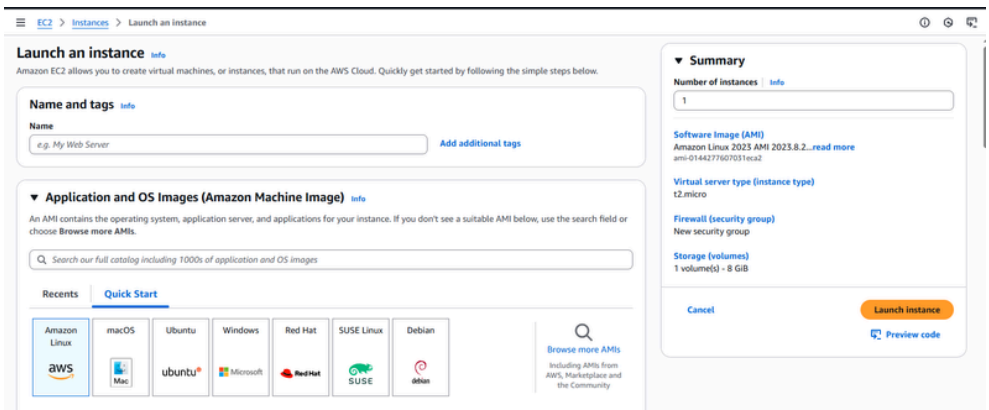
## Step: 1 - Log in into your AWS account and search for EC2 service.




## Step: 2 - Click on EC2



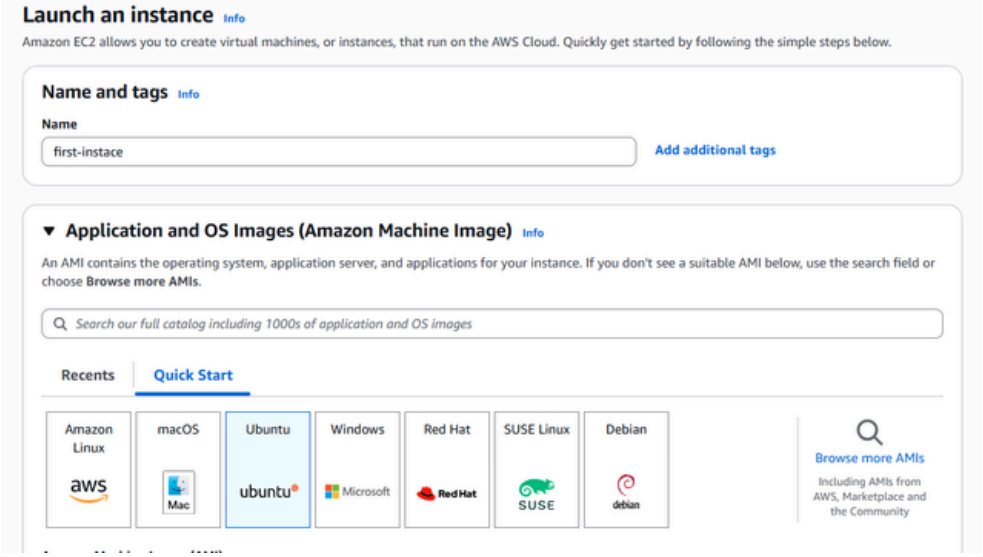

## Step: 3 - Click on Launch Instance and get this page



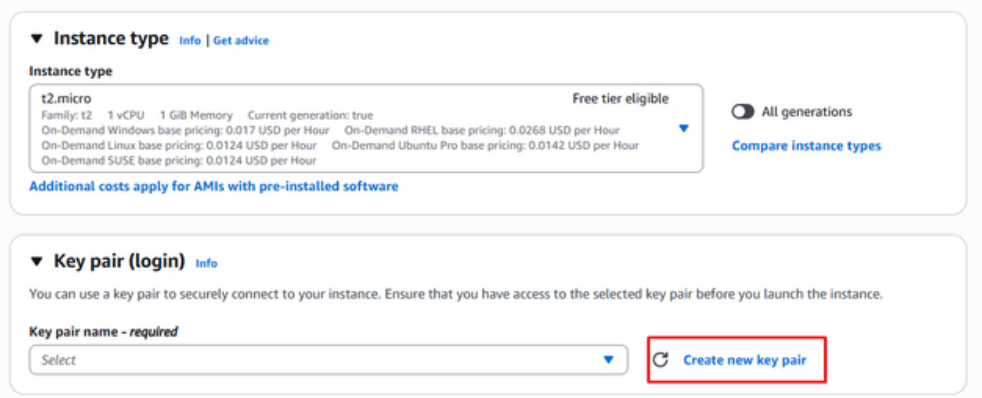
# Launch First EC2 Instance:

## Step: 4 - Now set configuration for EC2 instance

In my case, i used free tier for launch instance and create new key-pair file for connect with ssh.



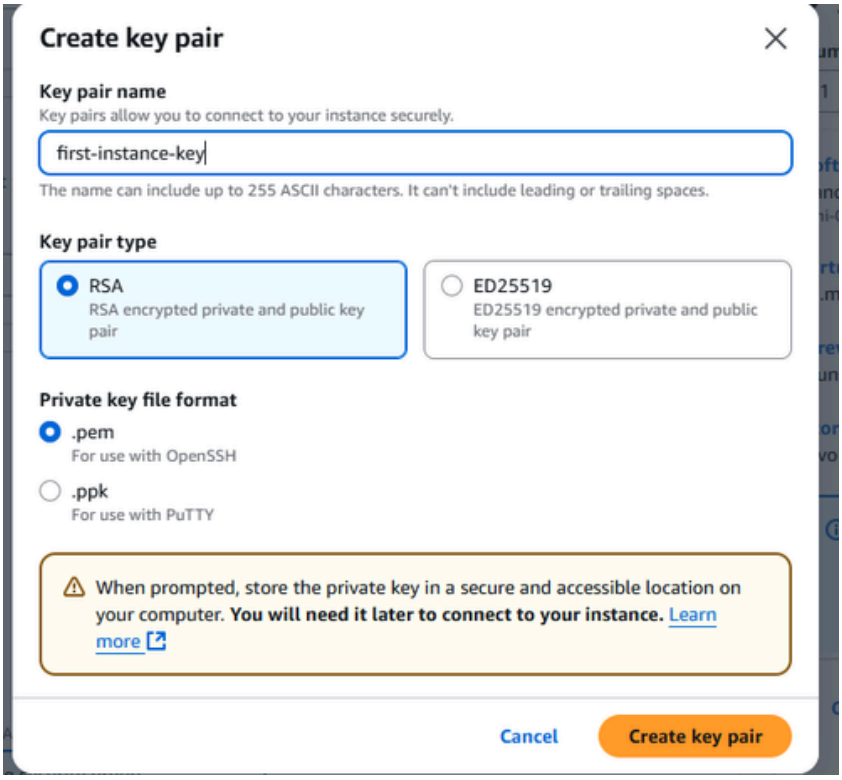

Give the name of your instance, and choose OS. I am choosing ubuntu.



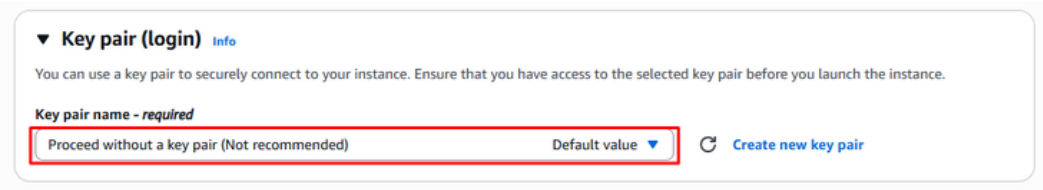

I keep all things as it is. Just create a new key pair for connect to SSH. If you need any other custom configuration you can.

# Launch First EC2 Instance:

Name your key pair if you want to create it. For an existing key pair no need naming just select previous one.

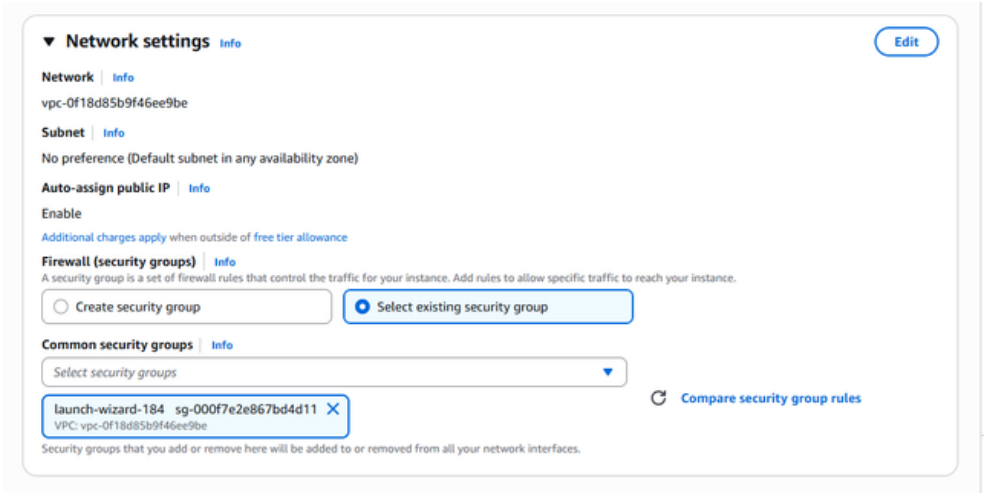



You can launch instance without key pair also.

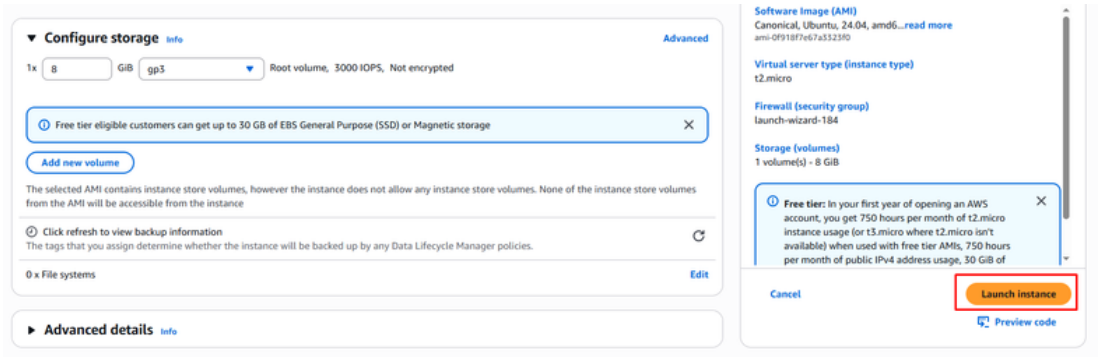


# Launch First EC2 Instance:

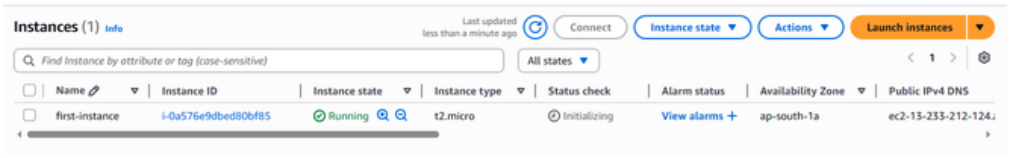
I use an existing security group. If you need, you can creeate new one.




After configure all of things, then click on Lanuch instance button and create it.

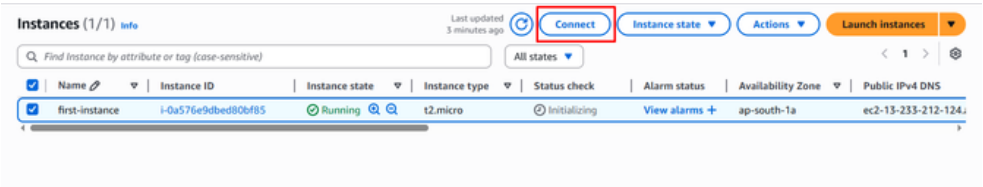



Successfully launch your instance you can see it in your EC2 instance page.

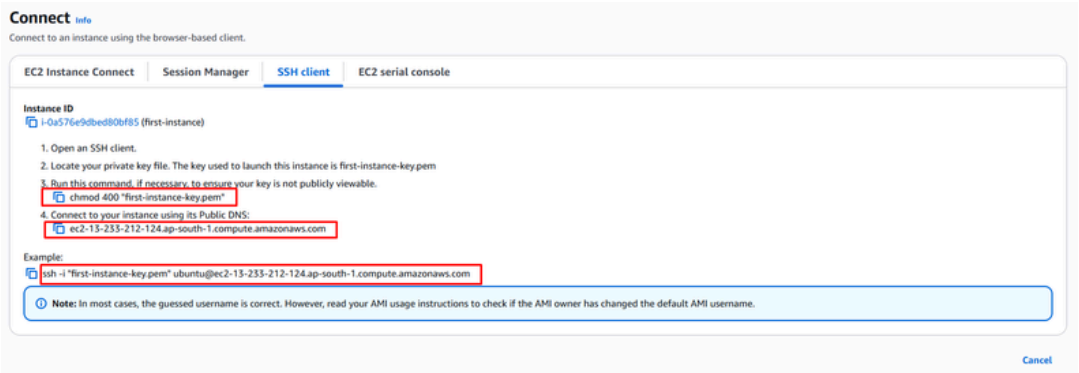


# Connect EC2 with SSH:

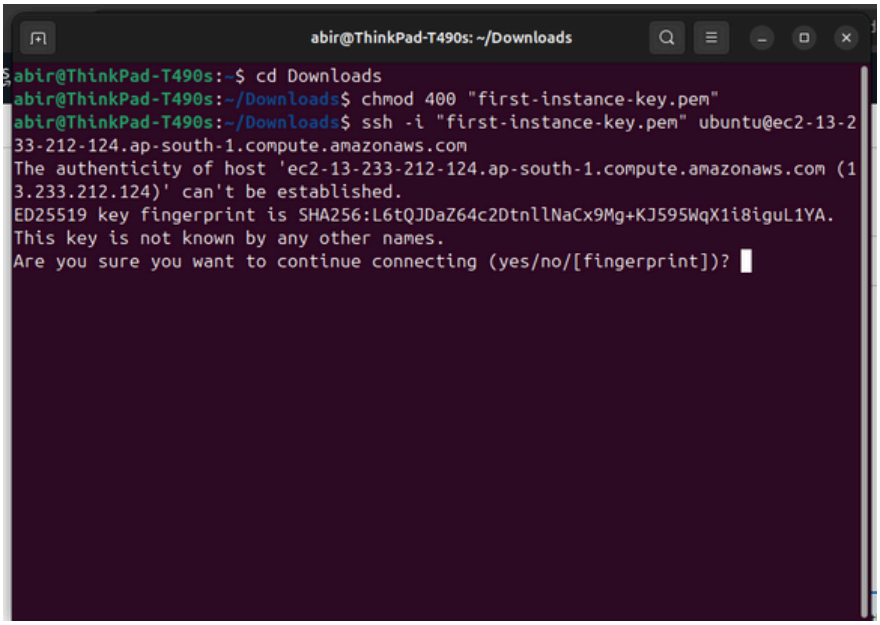
For connect your instance with SSH. First you need to connect your instance.




After connecting your instance, you need to go SSH client tab. From here, you need three info for connecting with ssh, which i marked.



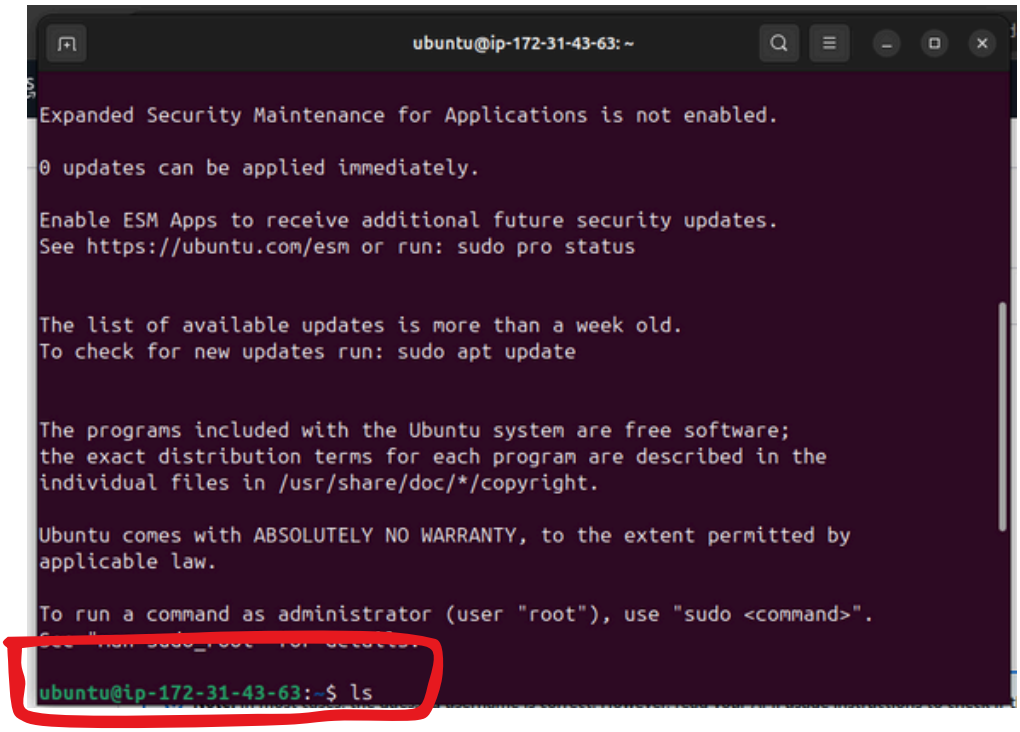

Now open you local machine terminal and connection the EC2 instance.




type yes and press enter to connect.

# Connect EC2 with SSH:

Successfully conntect EC2 with SSH.

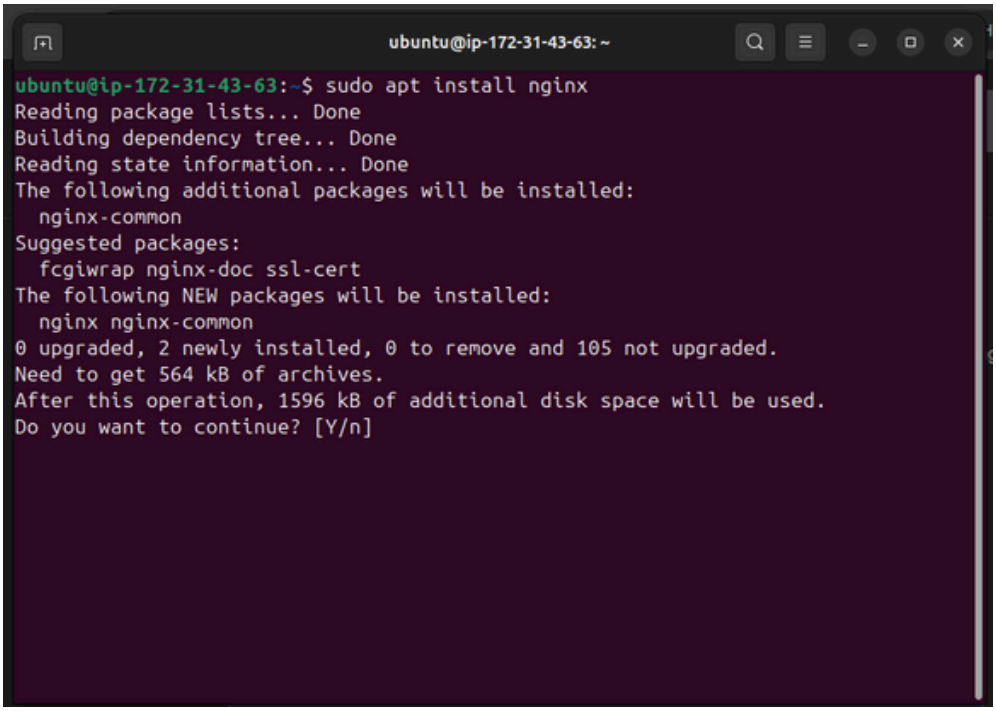



Now it’s time to configure nginx in EC2 instance.

# Configure nginx:

For configure nginx in your EC2 you need to run two one for system updates and another for install nginx.

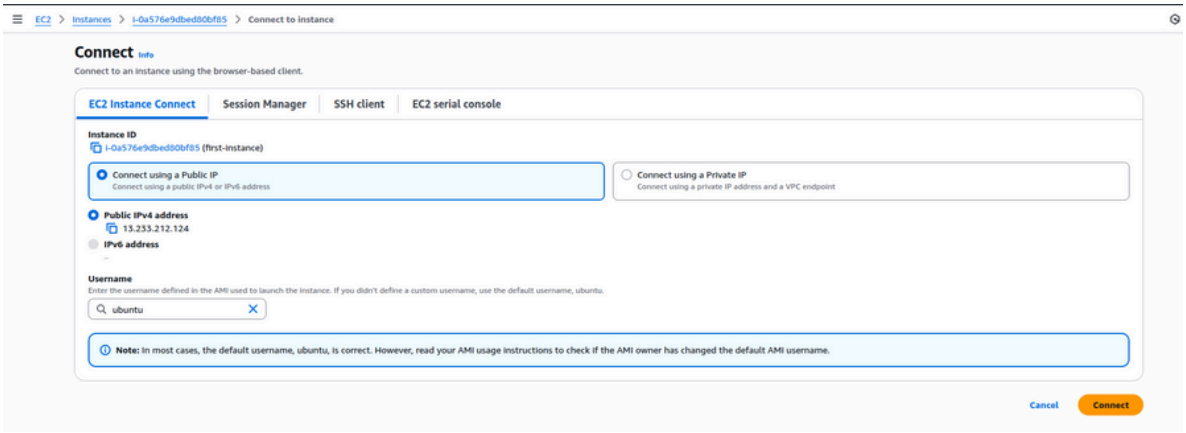
1. sudo apt update
2. sudo install nginx



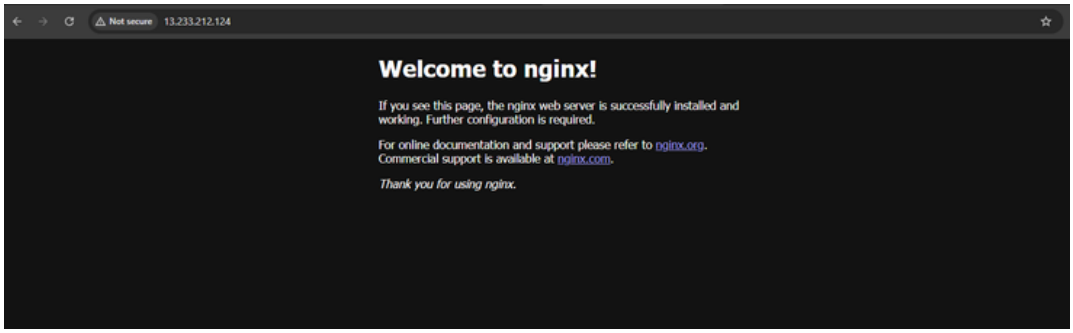

Type y and press enter for full configuration.

# Configure nginx:

It’s time to connect EC2 instance and testing.




Successfully configure the nginx.



# Thank You

## Stay Connect:

/in/alamgirweb11

/alamgirweb11

# AWS VPC

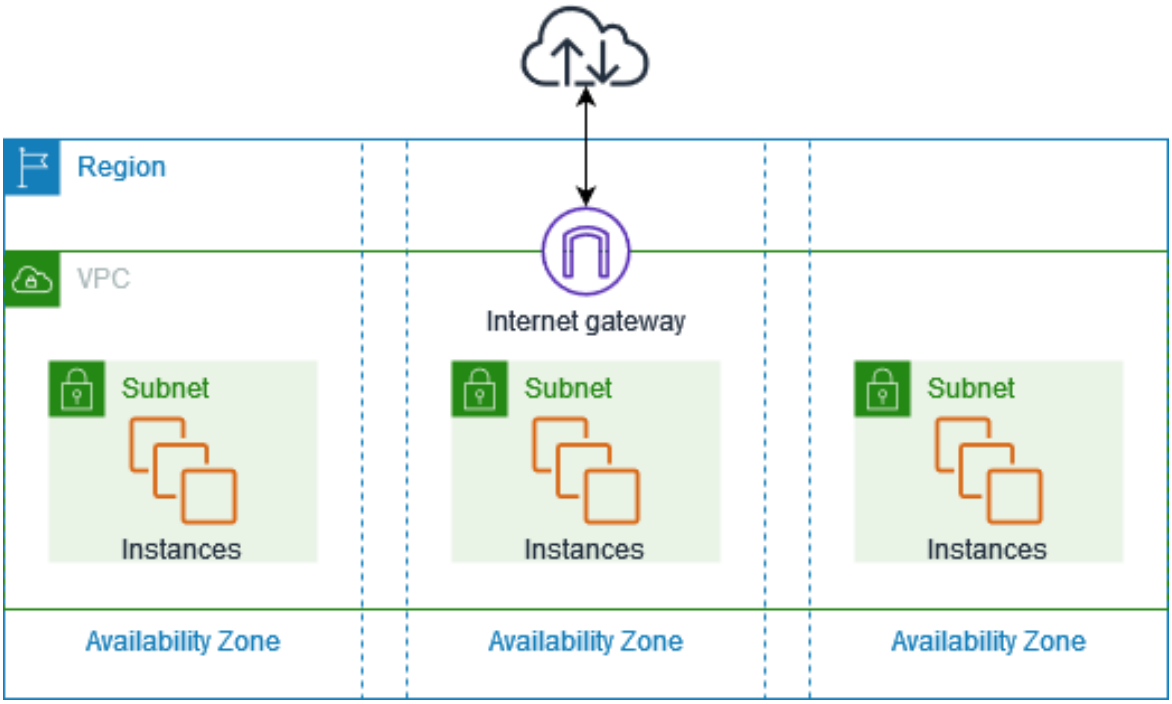
## Virtual Private Cloud

# Concept Overview:

# About VPC:

Amazon VPC (Virtual Private Cloud) is a logically isolated virtual network in AWS where you can launch and manage resources, similar to a traditional on-premises network but with the scalability of AWS.

The following diagram shows an example VPC. The VPC has one subnet in each of the Availability Zones in the Region, EC2 instances in each subnet, and an internet gateway to allow communication between the resources in your VPC and the internet.



# About VPC:

## Features of VPC:

### Virtual private clouds (VPC):

A VPC is a virtual network that closely resembles a traditional network that you'd operate in your own data center. After you create a VPC, you can add subnets.

### Subnets:

A subnet is a range of IP addresses in your VPC. A subnet must reside in a single Availability Zone. After you add subnets, you can deploy AWS resources in your VPC.

### IP addressing:

You can assign IP addressing, both IPv4 and IPv6, to your VPCs and subnets. You can also bring your public IPv4 addresses and IPv6 GUA addresses to AWS and allocate them to resources in your VPC, such as EC2 instances, NAT gateways, and Network Load Balancers.

### Routing:

Use route tables to determine where network traffic from your subnet or gateway is directed.

# About VPC:

## Features of VPC:

### Gateways and endpoints:

A gateway connects your VPC to another network. For example, use an internet gateway to connect your VPC to the internet. Use a VPC endpoint to connect to AWS services privately, without the use of an internet gateway or NAT device.

### Peering connections:

Use a VPC peering connection to route traffic between the resources in two VPCs.

### Traffic Mirroring:

Copy network traffic from network interfaces and send it to security and monitoring appliances for deep packet inspection.

### Transit gateways:

Use a transit gateway, which acts as a central hub, to route traffic between your VPCs, VPN connections, and AWS Direct Connect connections.

### VPC Flow Logs:

A flow log captures information about the IP traffic going to and from network interfaces in your VPC.

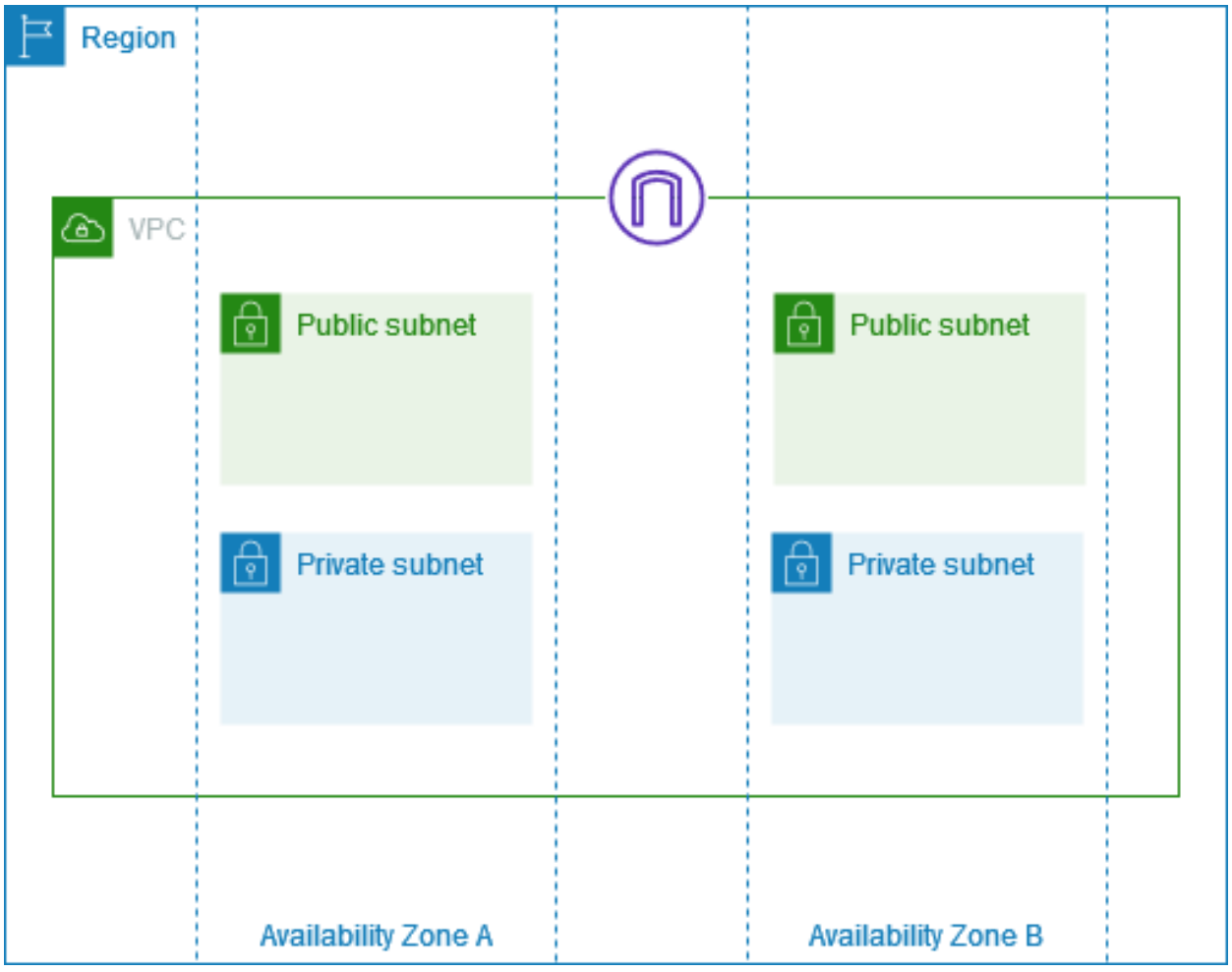
### VPN connections:

Connect your VPCs to your on-premises networks using AWS Virtual Private Network (AWS VPN).

# About Subnet:

A subnet is a range of IP addresses in your VPC. You can create AWS resources, such as EC2 instances, in specific subnets.

The following diagram shows a VPC with subnets in two Availability Zones and an internet gateway. Each Availability Zone has a public subnet and a private subnet.



# About Subnet:

**Subnet IP address range:**

When you create a subnet, you specify its IP addresses, depending on the configuration of the VPC:

- **IPv4 only:** The subnet has an IPv4 CIDR block but does not have an IPv6 CIDR block. Resources in an IPv4-only subnet must communicate over IPv4.
- **Dual stack:** The subnet has both an IPv4 CIDR block and an IPv6 CIDR block. The VPC must have both an IPv4 CIDR block and an IPv6 CIDR block. Resources in a dual-stack subnet can communicate over IPv4 and IPv6.
- **IPv6 only:** The subnet has an IPv6 CIDR block but does not have an IPv4 CIDR block. The VPC must have an IPv6 CIDR block. Resources in an IPv6-only subnet must communicate over IPv6.

**Subnet Types in VPC**

- **Public subnet:** Direct route to internet gateway → resources access internet.
- **Private subnet:** No direct internet route → uses NAT for outbound internet.
- **VPN-only subnet:** Routes traffic through Site-to-Site VPN via virtual private gateway.
- **Isolated subnet:** No external routes → access only within the same VPC.
- **EVS subnet:** Subnet created using Amazon EVS.

# About Subnet:

## Subnet routing

Each subnet must be associated with a route table, which specifies the allowed routes for outbound traffic leaving the subnet. Every subnet that you create is automatically associated with the main route table for the VPC. You can change the association, and you can change the contents of the main route table.

## Subnet settings

All subnets have a modifiable attribute that determines whether a network interface created in that subnet is assigned a public IPv4 address and, if applicable, an IPv6 address. This includes the primary network interface (for example, eth0) that's created for an instance when you launch an instance in that subnet. Regardless of the subnet attribute, you can still override this setting for a specific instance during launch.

After you create a subnet, you can modify the following settings for the subnet:

- **Auto-assign IP settings**: Enables you to configure the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.
- **Resource-based Name (RBN) settings**: Enables you to specify the hostname type for EC2 instances in this subnet and configure how DNS A and AAAA record queries are handled.

# About Subnet:

**Subnet security**

To protect your AWS resources, we recommend that you use private subnets. Use a bastion host or NAT device to provide internet access to resources, such as EC2 instances, in a private subnet.

AWS provides features that you can use to increase security for the resources in your VPC. Security groups allow inbound and outbound traffic for associated resources, such as EC2 instances. Network ACLs allow or deny inbound and outbound traffic at the subnet level. In most cases, security groups can meet your needs.

By design, each subnet must be associated with a network ACL. Every subnet that you create is automatically associated with the default network ACL for the VPC. The default network ACL allows all inbound and outbound traffic. You can update the default network ACL, or create custom network ACLs and associate them with your subnets.

You can create a flow log on your VPC or subnet to capture the traffic that flows to and from the network interfaces in your VPC or subnet. You can also create a flow log on an individual network interface.

## About Route Tables:

A route table serves as the traffic controller for your virtual private cloud (VPC). Each route table contains a set of rules, called routes, that determine where network traffic from your subnet or gateway is directed. When you create a VPC, we also create the main route table for the VPC. You can create additional route tables for your VPC, so that you have more granular control over the network paths for your VPC.

You can use route tables to specify which networks your VPC can communicate with, such as other VPCs or on-premises networks. Each route specifies a destination (CIDR block or prefix list) and a target (such as an internet gateway, NAT gateway, VPC peering connection, or VPN connection). Traffic is routed to targets based on its destination IP address. Route tables enable you to create complex networking architectures that include public subnets, private subnets, VPN-only subnets, and isolated subnets.

# About Route Tables:

## Features of Route Tables:

- **Main route table:** The route table that automatically comes with your VPC. It controls the routing for all subnets that are not explicitly associated with any other route table.
- **Custom route table:** A route table that you create for your VPC.
- **Destination:** The range of IP addresses where you want traffic to go (destination CIDR). For example, an external corporate network with the CIDR 172.16.0.0/12.
- **Target:** The gateway, network interface, or connection through which to send the destination traffic; for example, an internet gateway.
- **Local route:** A default route for communication within the VPC. If the VPC has both IPv4 and IPV6 addresses, there is a local route for IPv4 and a local route for IPv6.
- **Route table association:** The association between a route table and a subnet, internet gateway, or virtual private gateway.
- **Subnet route table:** A route table that's associated with a subnet.

# About Route Tables:

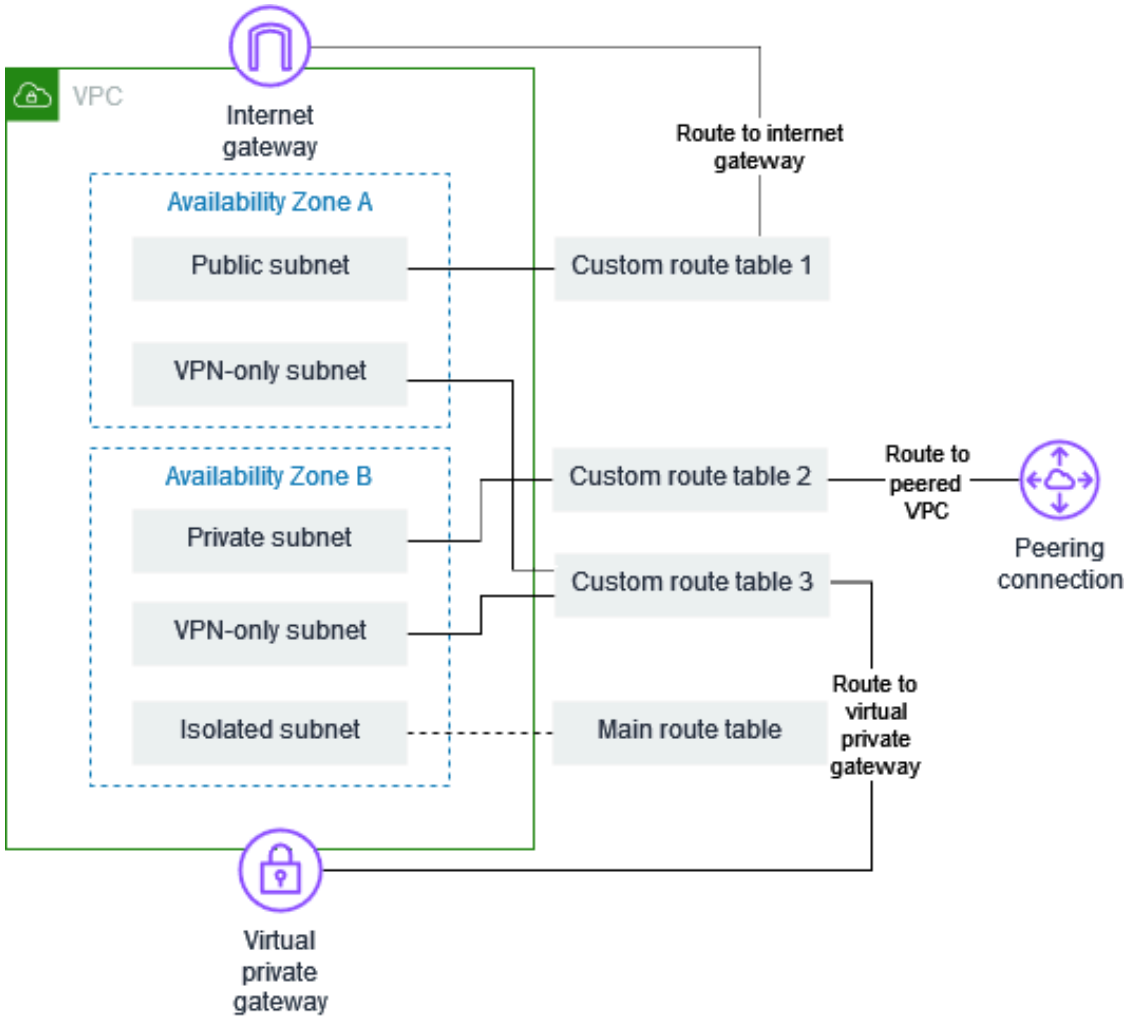
## Features of Route Tables:

- **Subnet route table:** A route table that's associated with a subnet.
- **Propagation:** If you've attached a virtual private gateway to your VPC and enable route propagation, we automatically add routes for your VPN connection to your subnet route tables. This means that you don't need to manually add or remove VPN routes.
- **Gateway route table:** A route table that's associated with an internet gateway or virtual private gateway.
- **Edge association:** A route table that you use to route inbound VPC traffic to an appliance. You associate a route table with the internet gateway or virtual private gateway, and specify the network interface of your appliance as the target for VPC traffic.
- **Transit gateway route table:** A route table that's associated with a transit gateway.
- **Local gateway route table:** A route table that's associated with an Outposts local gateway.

# About Route Tables:

## Example VPC with route tables:

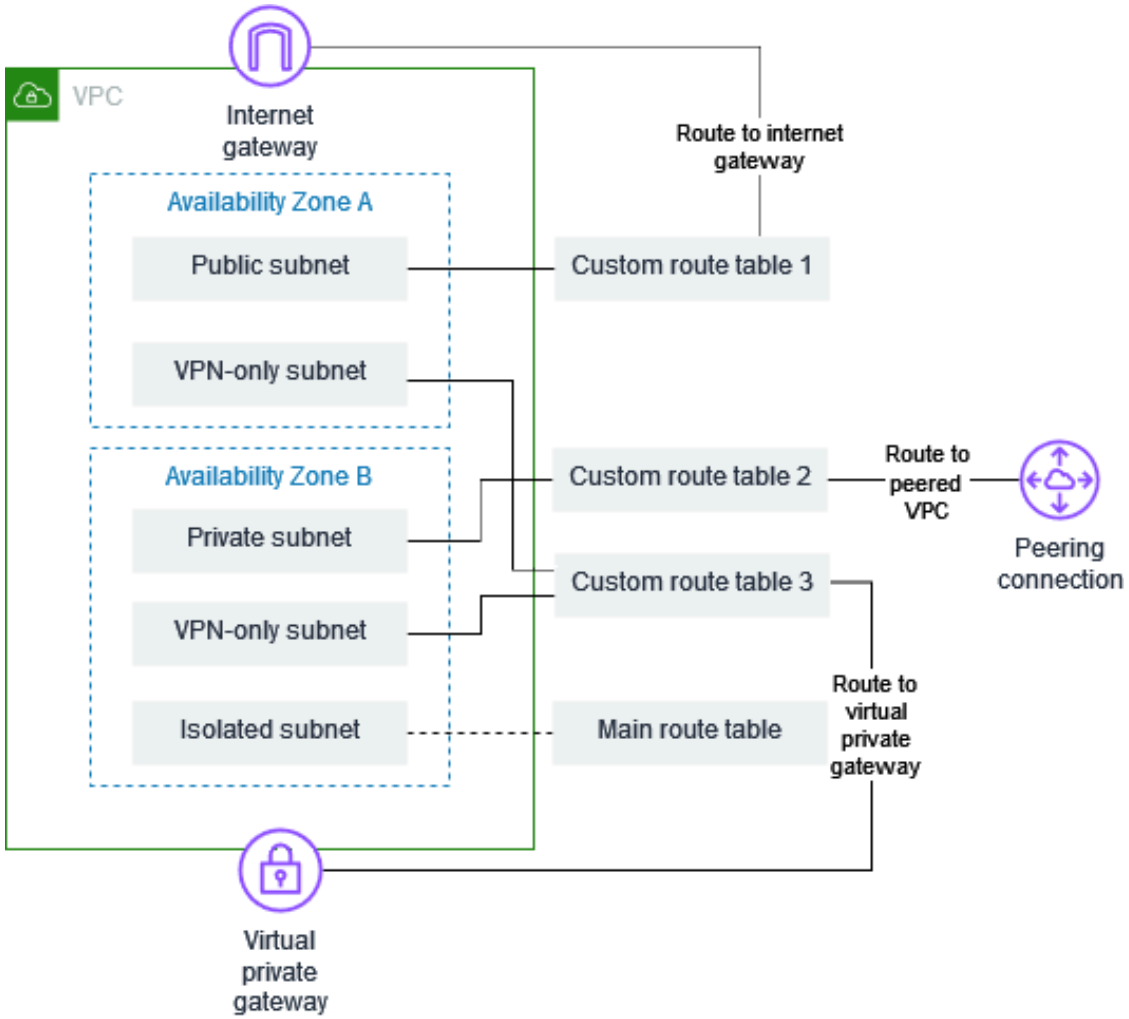
The following diagram shows a VPC with five subnets, a main route table, and three custom route tables. All four route tables have local routes. Custom route table 1 has a route to an internet gateway, and it is associated with the public subnet in Availability Zone A. Custom route table 2 has a route to a peered VPC, and it is associated with the private subnet in Availability Zone B. Custom route table 3 has a route to a virtual private gateway, and it is associated with the VPN-only subnets in both Availability Zones.



# About Route Tables:

## Example VPC with route tables:

The following diagram shows a VPC with five subnets, a main route table, and three custom route tables. All four route tables have local routes. Custom route table 1 has a route to an internet gateway, and it is associated with the public subnet in Availability Zone A. Custom route table 2 has a route to a peered VPC, and it is associated with the private subnet in Availability Zone B. Custom route table 3 has a route to a virtual private gateway, and it is associated with the VPN-only subnets in both Availability Zones.



# About Internet Gateway:

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between your VPC and the internet. It supports IPv4 and IPv6 traffic. It does not cause availability risks or bandwidth constraints on your network traffic.

An internet gateway enables resources in your public subnets (such as EC2 instances) to connect to the internet if the resource has a public IPv4 address or an IPv6 address. Similarly, resources on the internet can initiate a connection to resources in your subnet using the public IPv4 address or IPv6 address. For example, an internet gateway enables you to connect to an EC2 instance in AWS using your local computer.

An internet gateway provides a target in your VPC route tables for internet-routable traffic.

# About Internet Gateway:

## Internet gateway basics:

**Routing configuration:**

If a subnet is associated with a route table that has a route to an internet gateway, it's known as a public subnet. If a subnet is associated with a route table that does not have a route to an internet gateway, it's known as a private subnet.

In your public subnet's route table, you can specify a route for the internet gateway to all destinations not explicitly known to the route table (0.0.0.0/0 for IPv4 or ::/0 for IPv6). Alternatively, you can scope the route to a narrower range of IP addresses; for example, the public IPv4 addresses of your company's public endpoints outside of AWS, or the Elastic IP addresses of other Amazon EC2 instances outside your VPC.

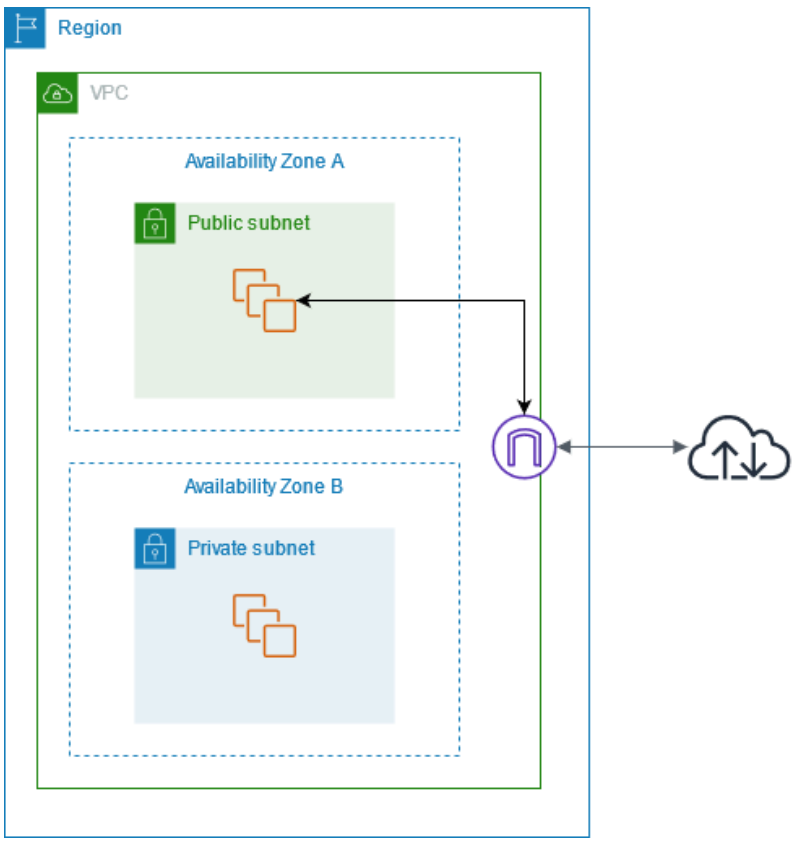
# About Internet Gateway:

## Internet gateway basics:

### Internet gateway diagram:

If a subnet is associated with a route table that has a route to an internet gateway, it's known as a public subnet. If a subnet is associated with a route table that does not have a route to an internet gateway, it's known as a private subnet.

In your public subnet's route table, you can specify a route for the internet gateway to all destinations not explicitly known to the route table (0.0.0.0/0 for IPv4 or ::/0 for IPv6). Alternatively, you can scope the route to a narrower range of IP addresses; for example, the public IPv4 addresses of your company's public endpoints outside of AWS, or the Elastic IP addresses of other Amazon EC2 instances outside your VPC.



# About NAT Gateway:

A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services can't initiate a connection with those instances.

When you create a NAT gateway, you specify one of the following connectivity types:

**Public NAT Gateway**

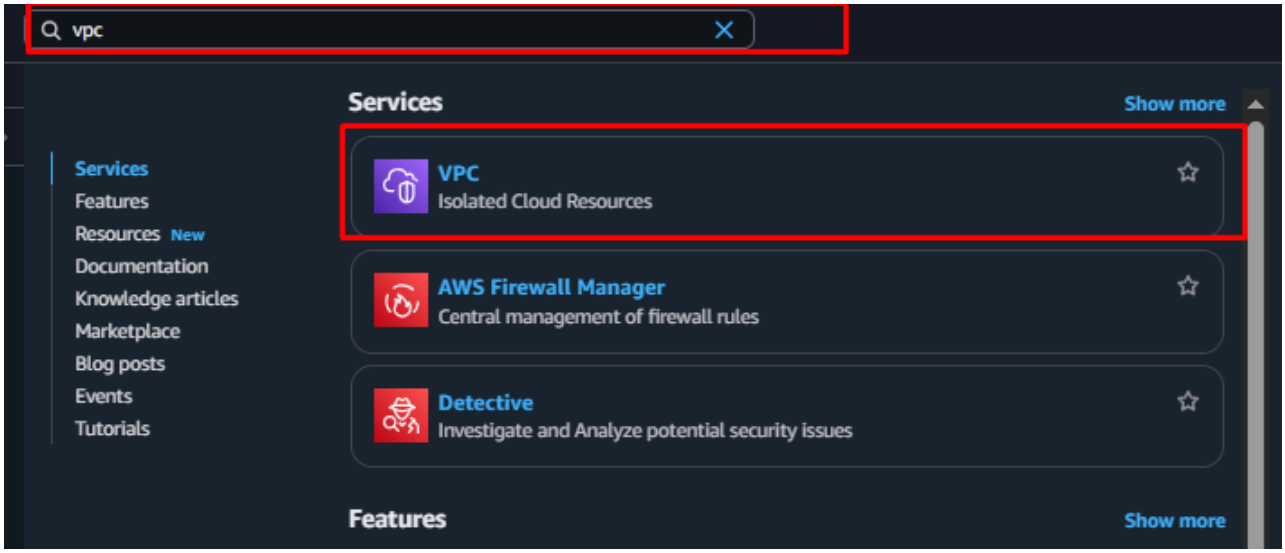
- Placed in a public subnet with an Elastic IP.
- Lets private subnet instances connect to the internet (outbound only).
- Routes through Internet Gateway (IGW), Transit Gateway, or Virtual Private Gateway.
- Cannot receive inbound internet traffic.

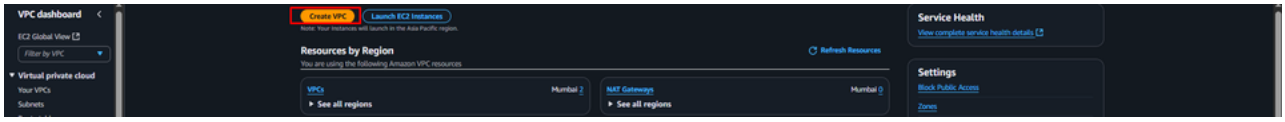
**Private NAT Gateway**

- Placed in a private subnet (no Elastic IP).
- Lets private instances connect to other VPCs or on-premises networks (outbound only).
- Routes through Transit Gateway or Virtual Private Gateway.
- Internet Gateway drops traffic if routed directly.

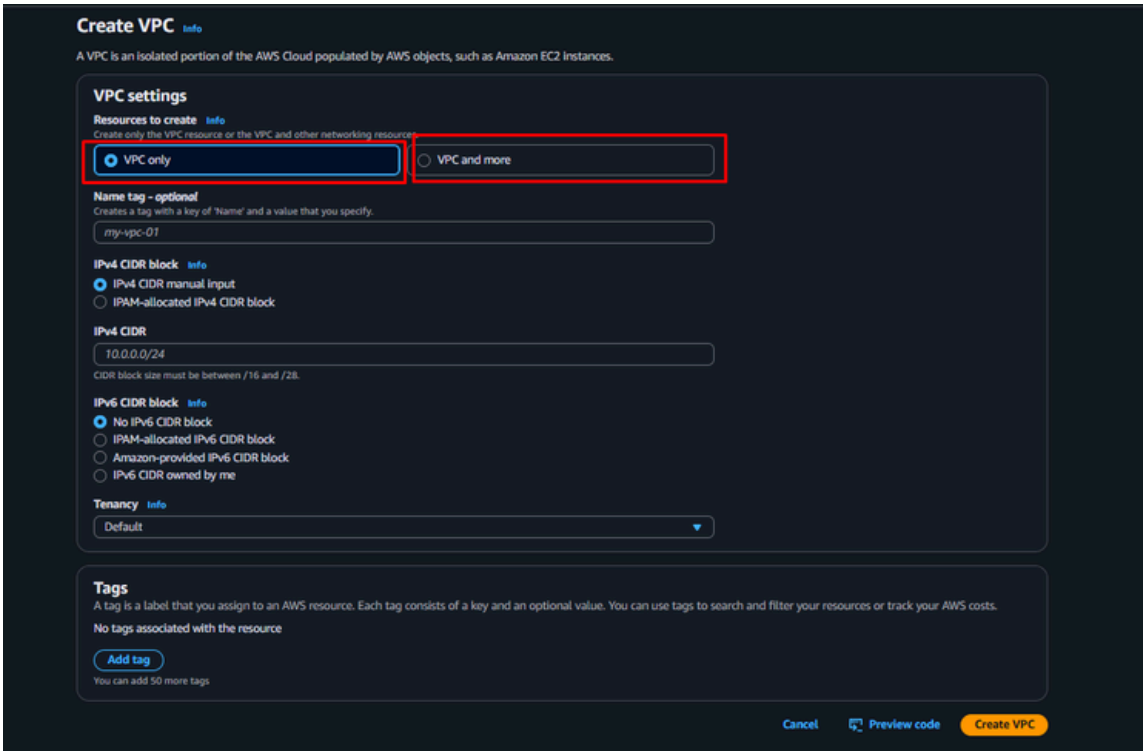
# Configure VPC:

To configure a VPC, first log in to your AWS Console and then search for VPC.



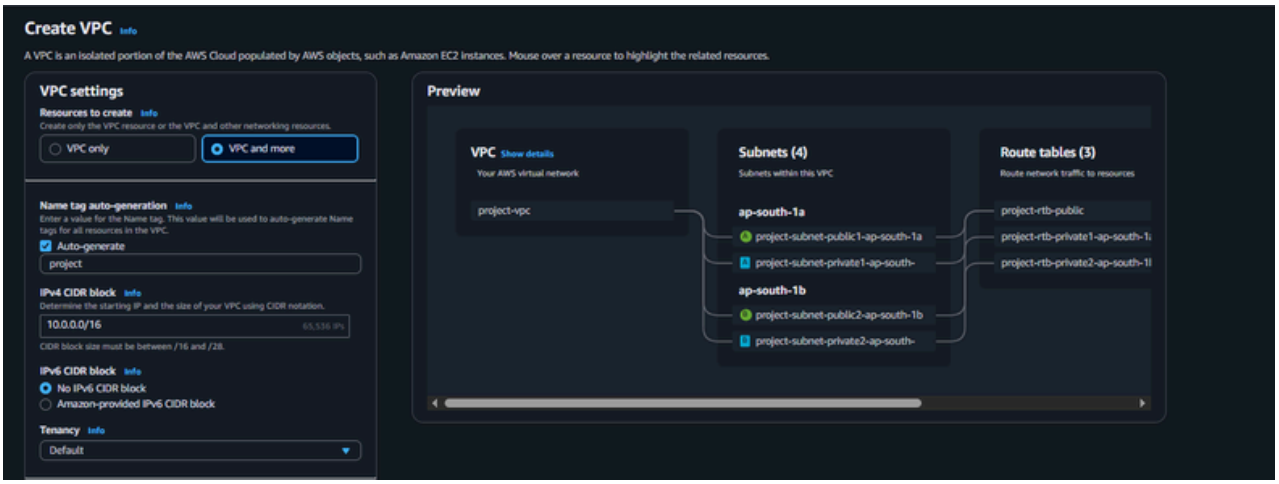




On the VPC creation page, you get two options: create a VPC manually or use the default method.

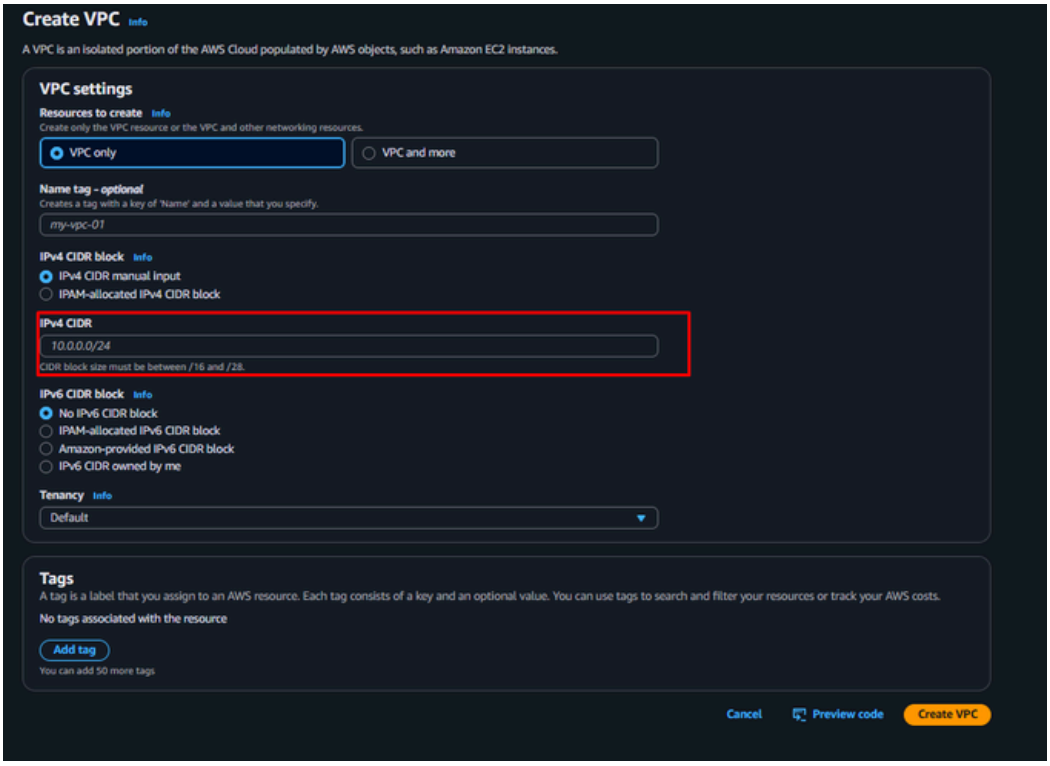


# Configure VPC:

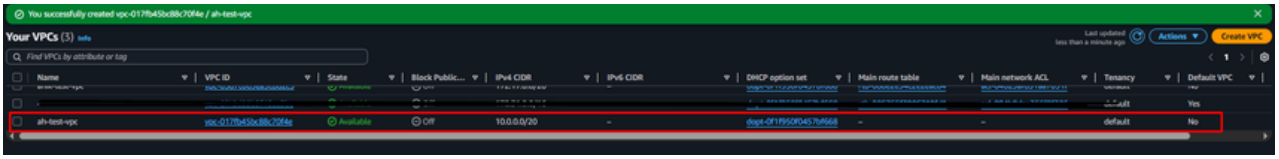
The default way looks like this, but I chose to create the VPC manually for better understanding.




For manual VPC creation, you need to set an IPv4 CIDR.

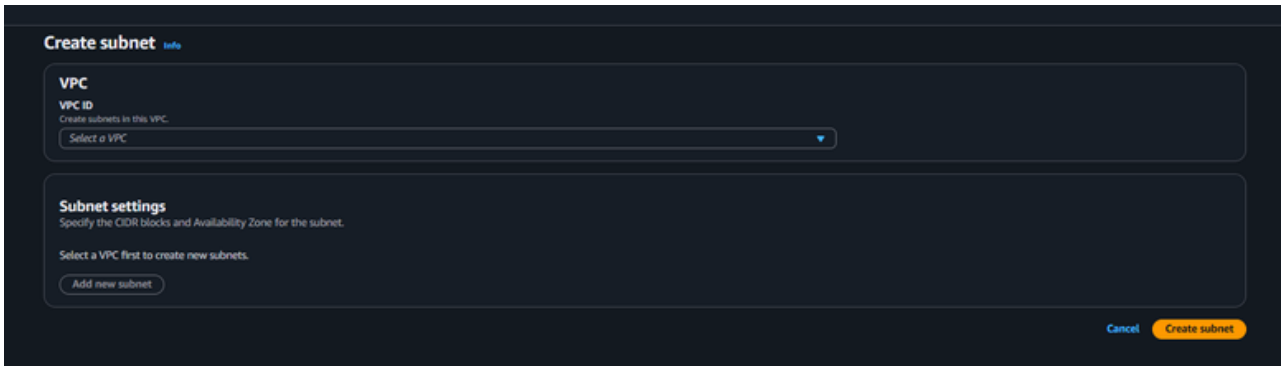



Here the VPC:

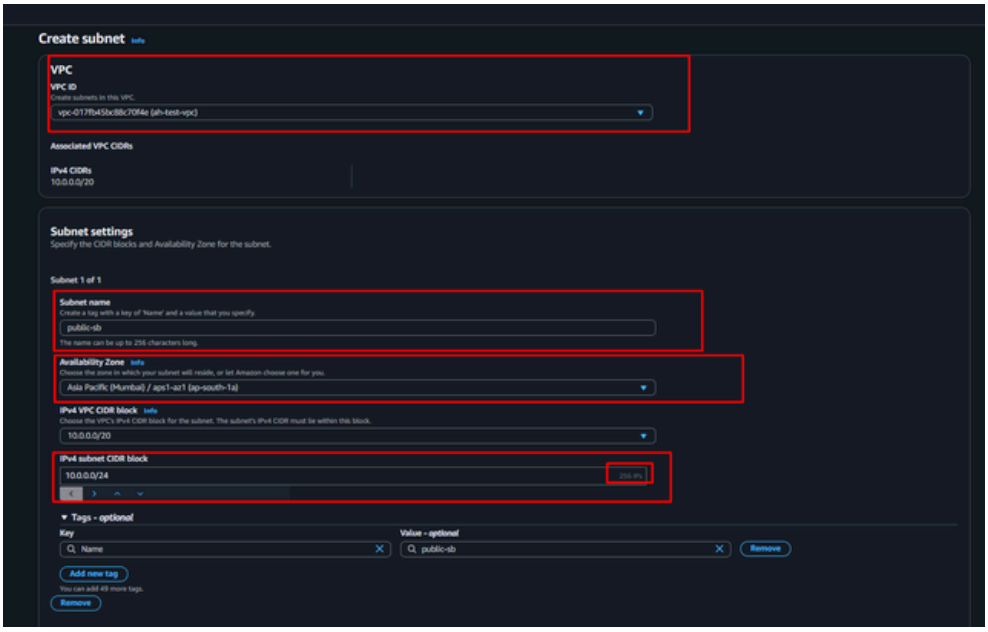


# Configure Subnet:

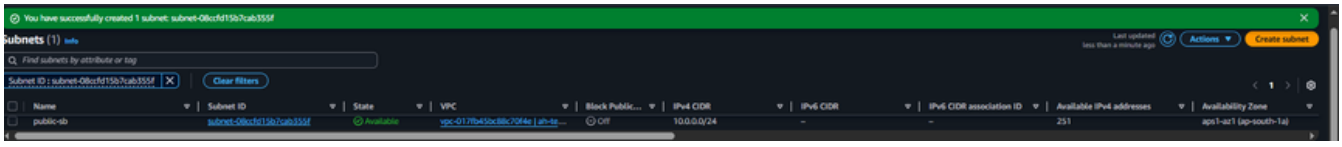
Now, configure two subnets: one Public and one Private. Make sure to select the newly created VPC.




Here the important config input fields.

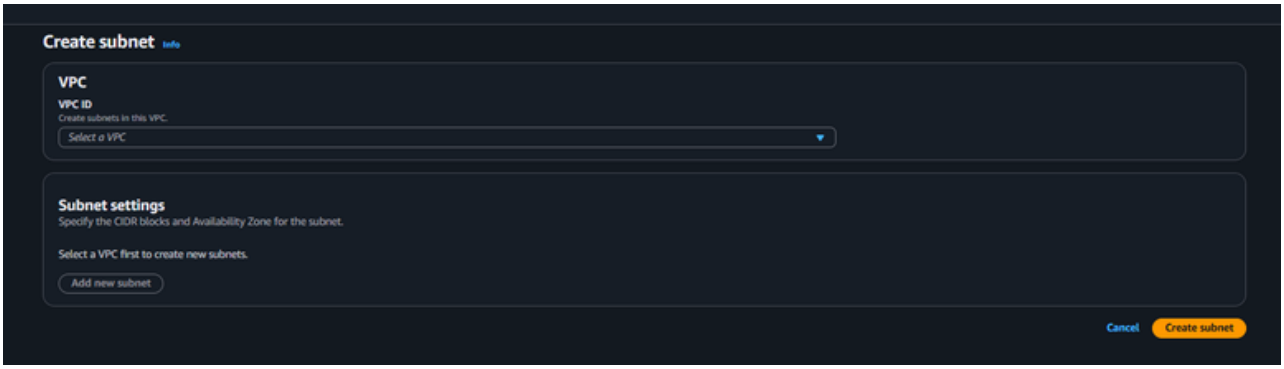



Here the Public subnet:

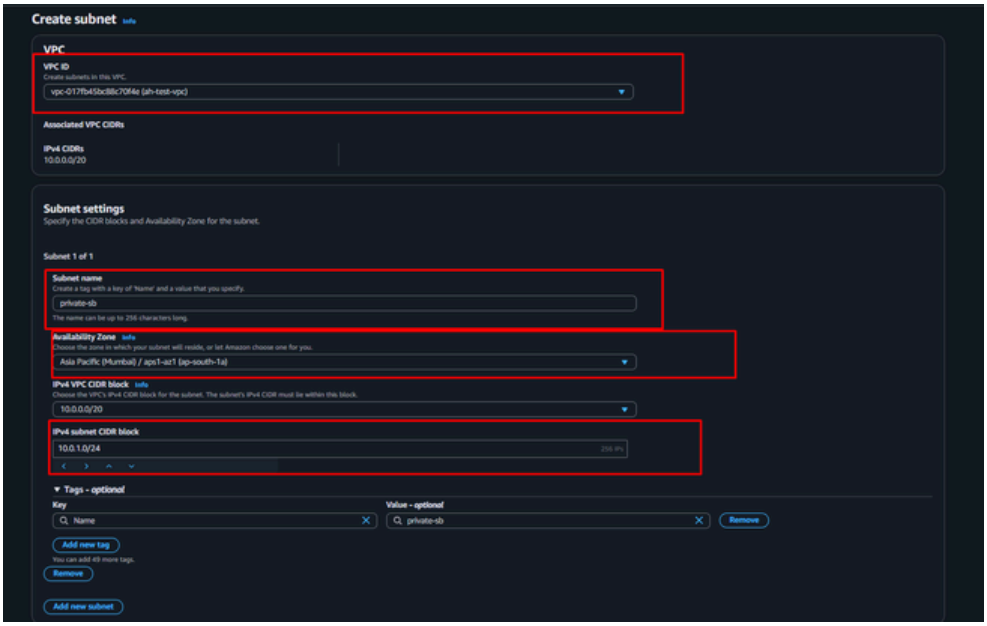


# Configure Subnet:

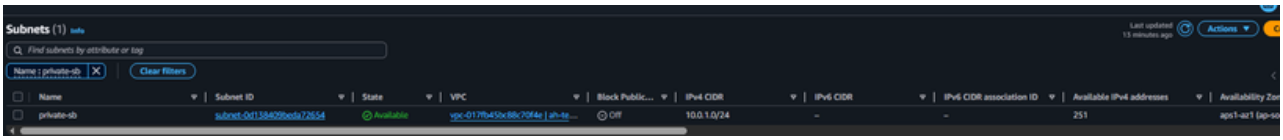
To create a Private subnet, follow the same steps as the Public subnet. The only difference is in the configuration settings you choose for it.




Here the important config input fields.



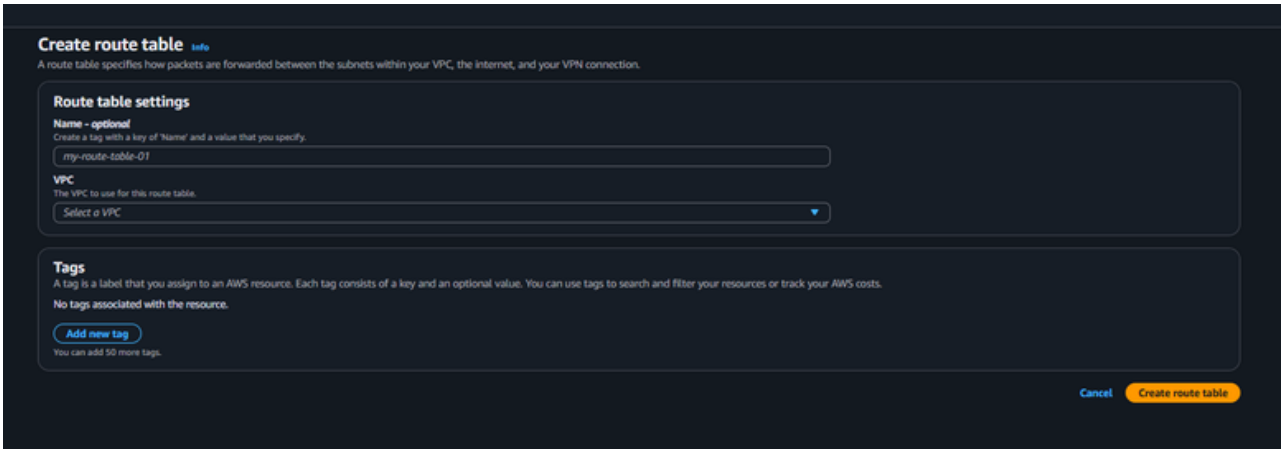

Here the Private subnet:



# Configure Route Tables:

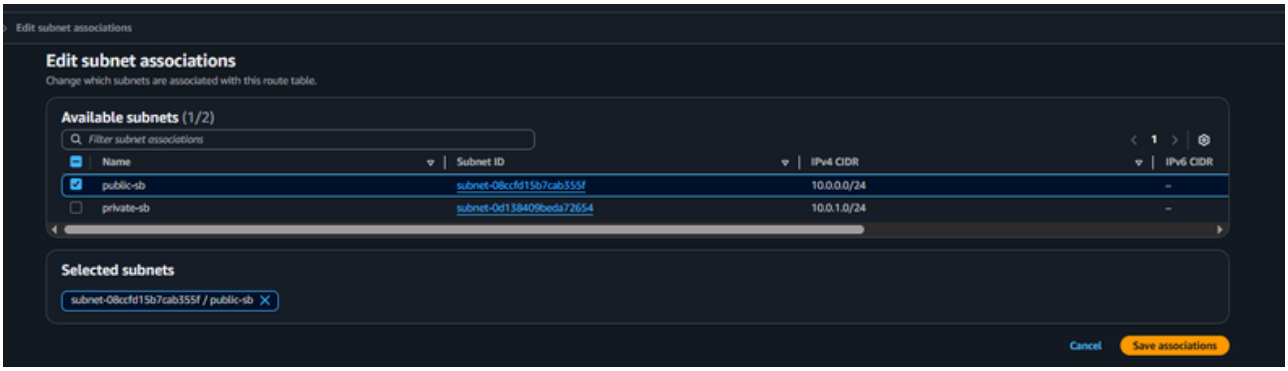
Here is the resource map showing the default Route Tables.


Now, create two new Route Tables — one for the Public Subnet and another for the Private Subnet.



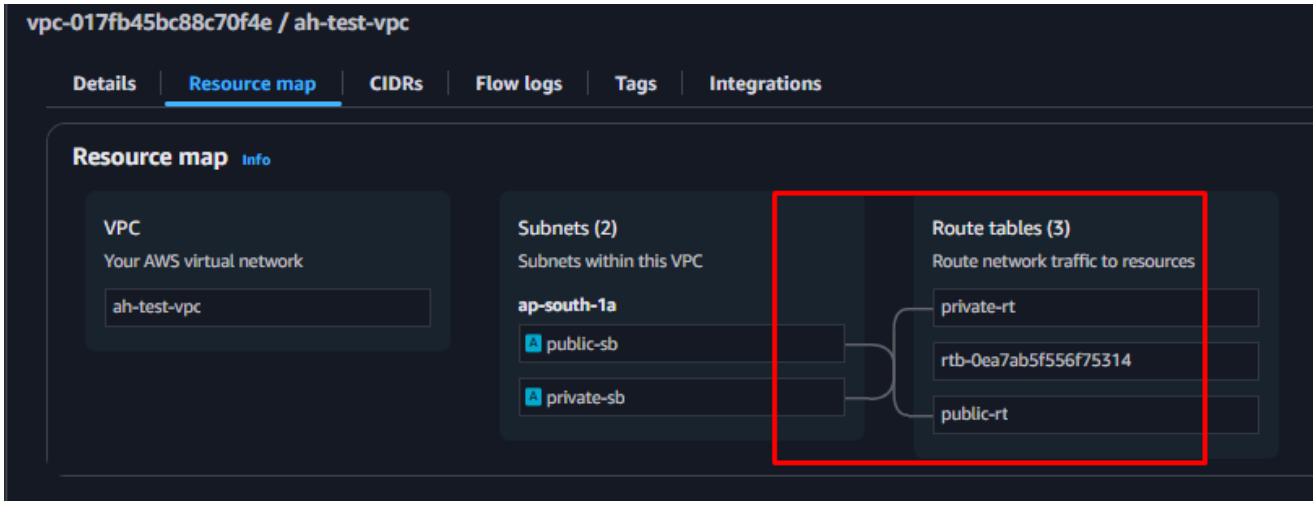

Here the two Route Tables:


Now, associate with two difference subnets.



# Configure Route Tables:

Route Tables connect with two different subnets.




Now, you can create an EC2 instance inside the newly created VPC and subnets. If you want communication between the two different subnets, you need to configure a NAT Gateway for proper peering and connectivity.

# Thank You

## Stay Connect:

/in/alamgirweb11

/alamgirweb11