

AWS EC2

& Security Groups



Concept Overview:

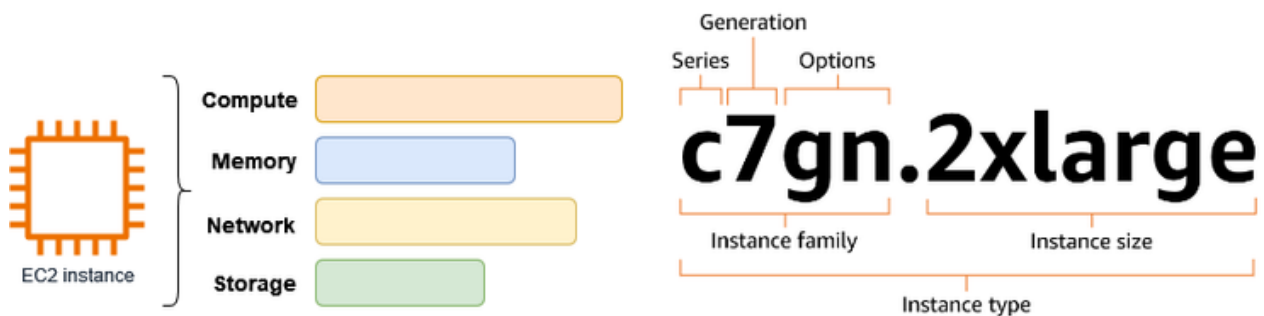
Introduction to EC2	1
EC2 Features	2
EC2 Lifecycle	3
EC2 Instance Types	4
User Data	5
Security Groups	6
Launch First EC2 Instance	7
Connect EC2 with SSH	8
Configure nginx	9

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.



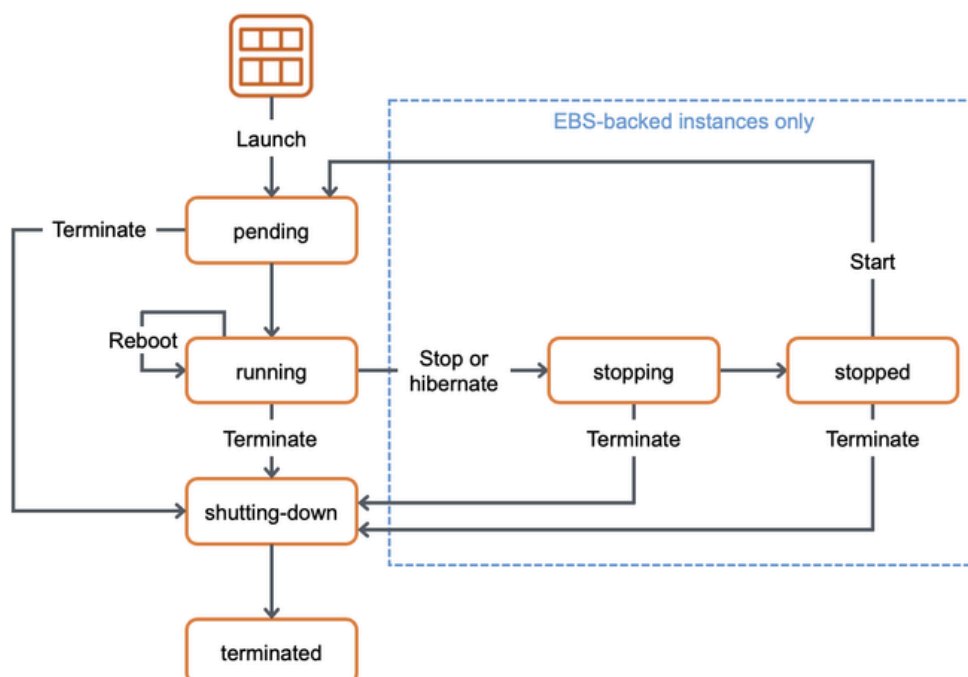
EC2 Features:

- **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:



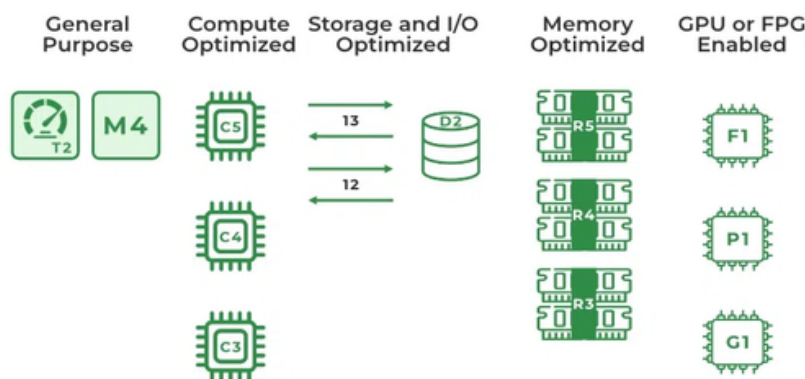
EC2 Instance Types:

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.

The image shows two screenshots of the AWS Management Console 'Launch an instance' page. The top screenshot highlights the 'Advanced details' section, which includes options for 'Domain join directory', 'IAM instance profile', 'Hostname type', and 'DNS Hostname'. The bottom screenshot highlights the 'User data - optional' section, which allows users to upload a file or enter a script. The script entered is a bash script that updates package lists and installs Apache2.

Advanced details

Domain join directory | Info
Select [Create new directory](#)

IAM instance profile | Info
Select [Create new IAM profile](#)

Hostname type | Info
IP name

DNS Hostname | Info
☒ Enable IP name IPv4 (A record) DNS requests
☒ Enable resource-based IPv4 (A record) DNS requests
☐ Enable resource-based IPv6 (AAAA record) DNS requests

User data - optional | Info
Upload a file with your user data or enter it in the field.
[Choose file](#)

```
#!/bin/bash
# Update package lists
sudo apt update -y

# Install Apache2
sudo apt install apache2 -y
```

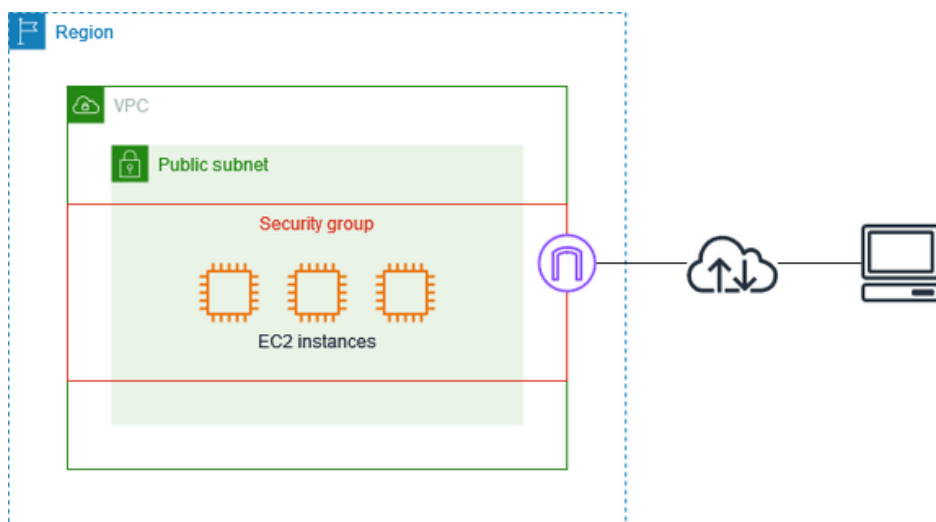
☐ User data has already been base64 encoded

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.

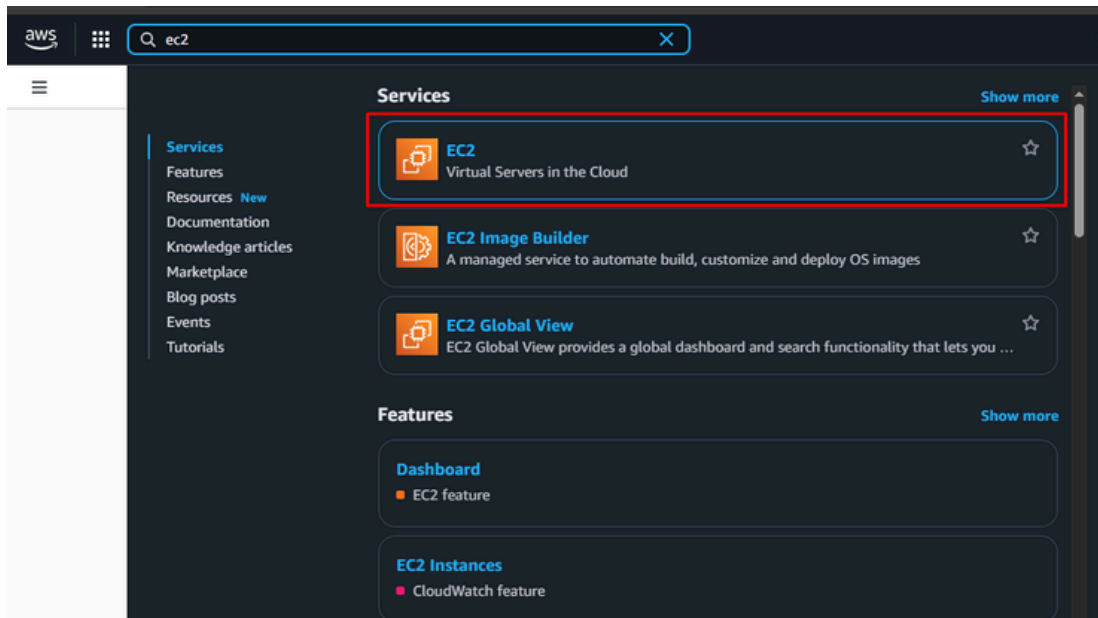


Fill up the input fields based on your needs and click on Create Security Group.

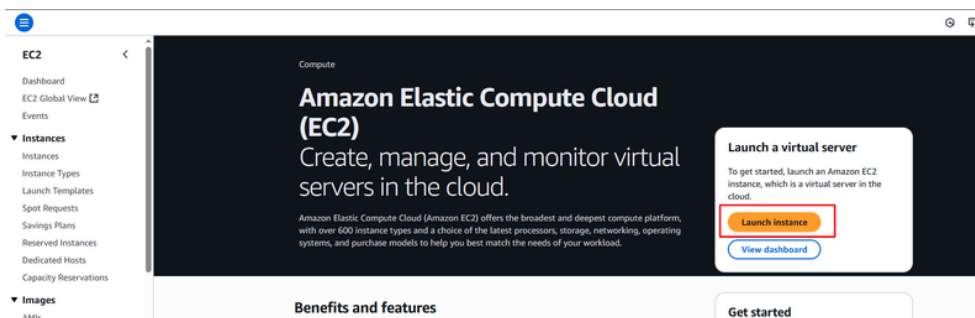
This screenshot shows the 'Create security group' form in the AWS Management Console. The form is divided into several sections: 'Basic details' with fields for 'Security group name' (pre-filled with 'MyWebServerGroup'), 'Description' (pre-filled with 'Allows SSH access to developers'), and 'VPC' (pre-filled with 'vpc-0f18d8529f46ee9be'); 'Inbound rules' with a message 'This security group has no inbound rules' and an 'Add rule' button; 'Outbound rules' with a table for defining rules, including columns for Type, Protocol, Port range, Destination, and Description - optional; and 'Tags - optional' with a message 'No tags associated with the resource' and an 'Add new tag' button. At the bottom right, there are 'Cancel' and 'Create security group' buttons, with the latter highlighted by a red box.

Launch First EC2 Instance:

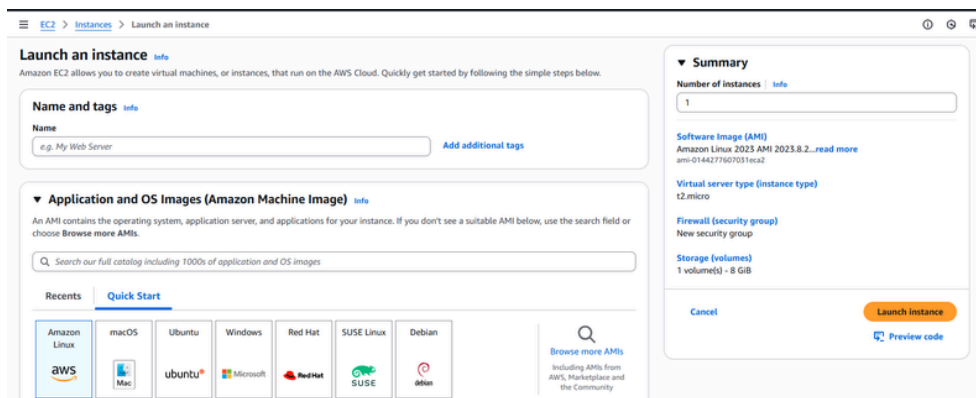
Step: 1 - Log in into your AWS account and search for EC2 service.



Step: 2 - Click on EC2



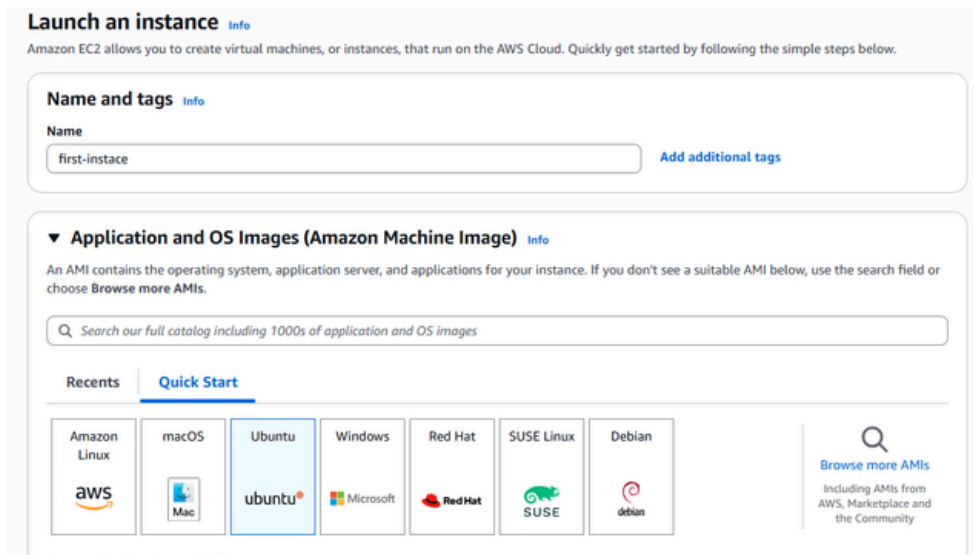
Step: 3 - Click on Launch Instance and get this page



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.



Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name
first-instance [Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Q Search our full catalog including 1000s of application and OS images

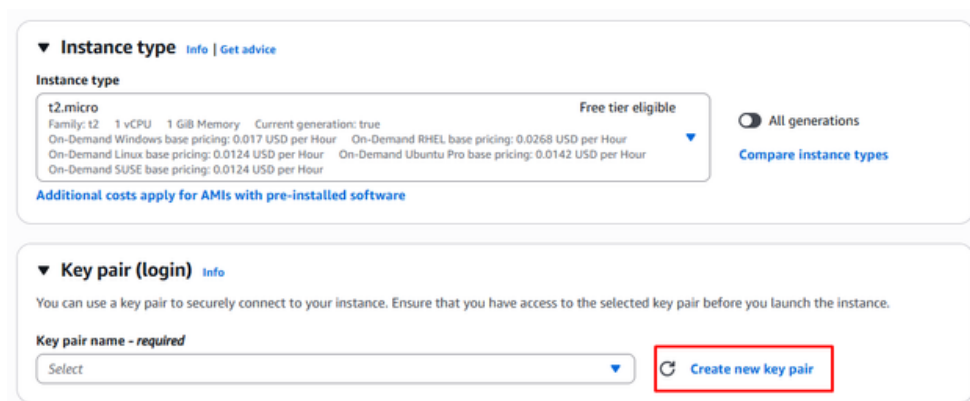
Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

aws Mac ubuntu Microsoft Red Hat SUSE debian

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

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



▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.017 USD per Hour On-Demand RHEL base pricing: 0.0268 USD per Hour

On-Demand Linux base pricing: 0.0124 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour

On-Demand SUSE base pricing: 0.0124 USD per Hour

[Additional costs apply for AMIs with pre-installed software](#)

[All generations](#)

[Compare instance types](#)

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

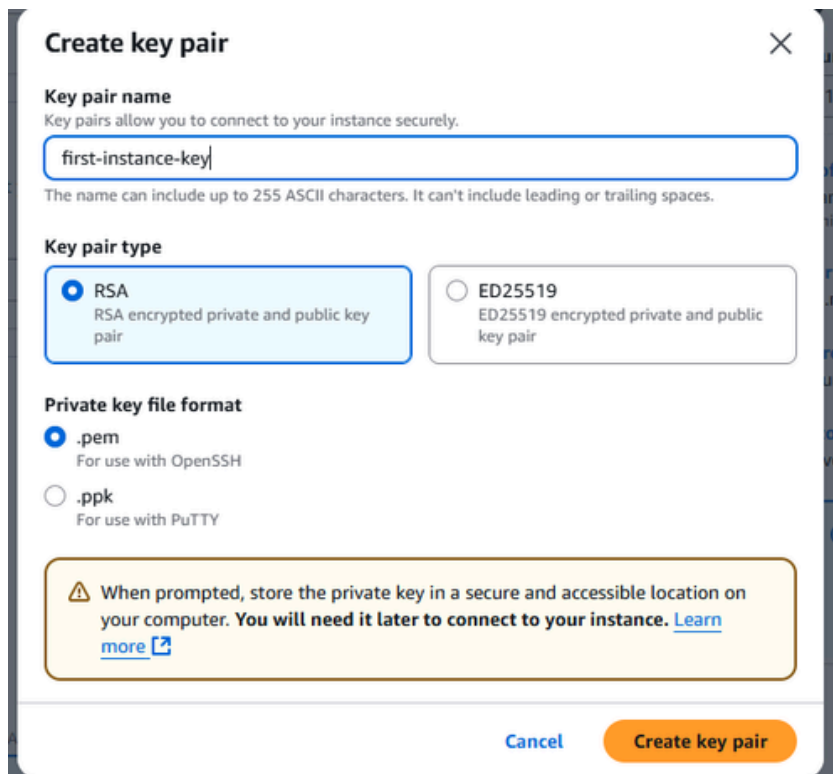
Key pair name - required

Select [Create new key pair](#)

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.



The screenshot shows the 'Create key pair' dialog box. It has a title bar with a close button (X). The main content area is divided into sections: 'Key pair name' with a text input field containing 'first-instance-key' and a note about character limits; 'Key pair type' with two radio button options: 'RSA' (selected) and 'ED25519'; 'Private key file format' with two radio button options: '.pem' (selected) and '.ppk'. At the bottom, there is a warning box with a yellow triangle icon and text about storing the private key. Below the warning box are two buttons: 'Cancel' and 'Create key pair'.

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.
first-instance-key
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ **RSA**
RSA encrypted private and public key pair

☐ **ED25519**
ED25519 encrypted private and public key pair

Private key file format

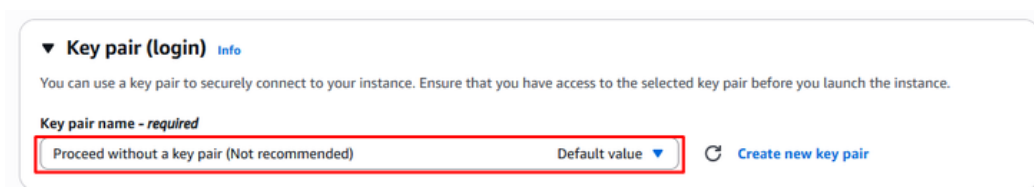
☒ **.pem**
For use with OpenSSH

☐ **.ppk**
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel Create key pair

You can launch instance without key pair also.



The screenshot shows the 'Key pair (login)' section. It has a dropdown arrow and the text 'Key pair (login)' followed by an 'Info' link. Below this is a note: 'You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.' Underneath is a label 'Key pair name - required' followed by a text input field containing 'Proceed without a key pair (Not recommended)'. To the right of the input field is a 'Default value' dropdown menu. Further right is a circular refresh icon and a link 'Create new key pair'.

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Proceed without a key pair (Not recommended) Default value ▼ ↻ [Create new key pair](#)

Launch First EC2 Instance:

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

Network settings Info Edit

Network Info
vpc-0f18d85b9f46ee9be

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance

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

Common security groups Info
Select security groups
launch-wizard-184 sg-000f7e2e867bd4d11 X
VPC: vpc-0f18d85b9f46ee9be

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

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

Configure storage Info Advanced

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

Free tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

Advanced details Info

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...read more
ami-0f918f7e67a532350

Virtual server type (instance type)
t2.micro

Firewall (security group)
launch-wizard-184

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of

Cancel Launch instance Preview code

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

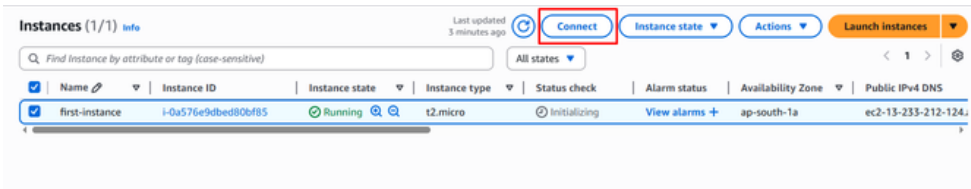
Instances (1) Info

Find Instance by attribute or tag (case-sensitive) All states

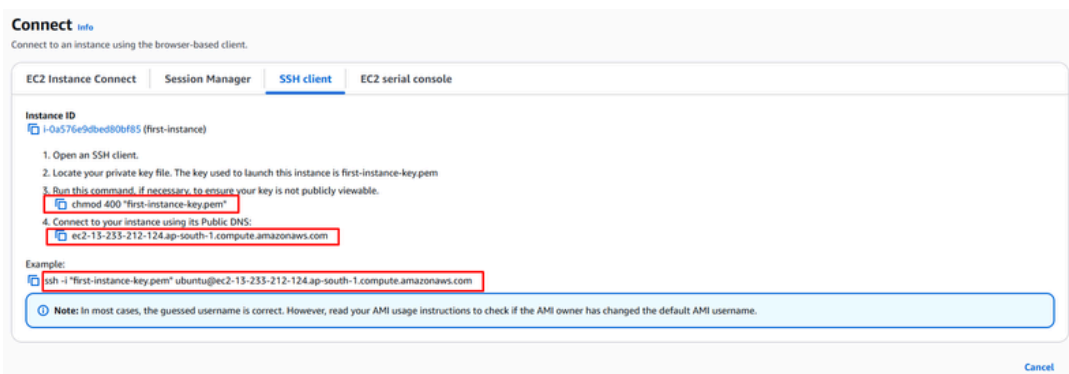
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
first-instance	i-0a576e9dbed80bf85	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-233-212-124...

Connect EC2 with SSH:

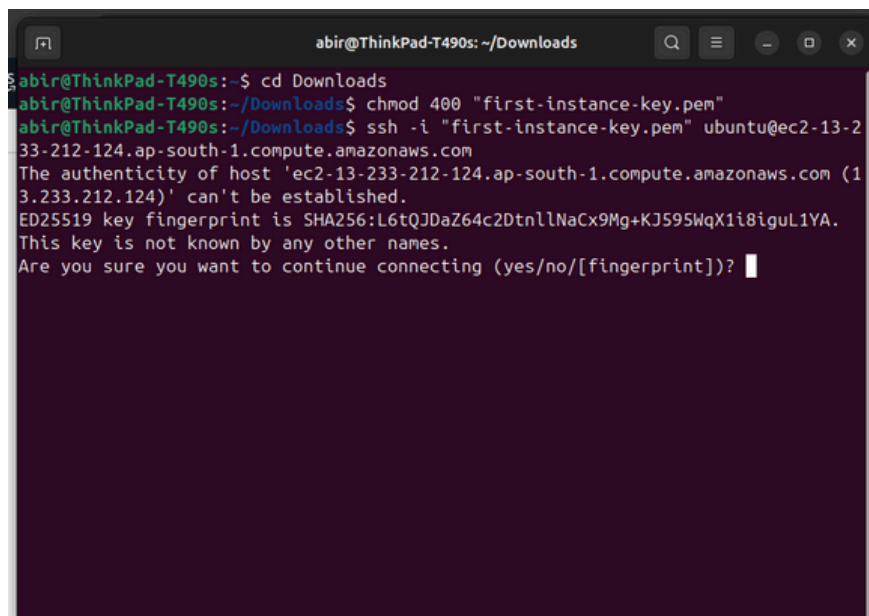
For connect your instance with SSH. First you need to connect your instance.



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



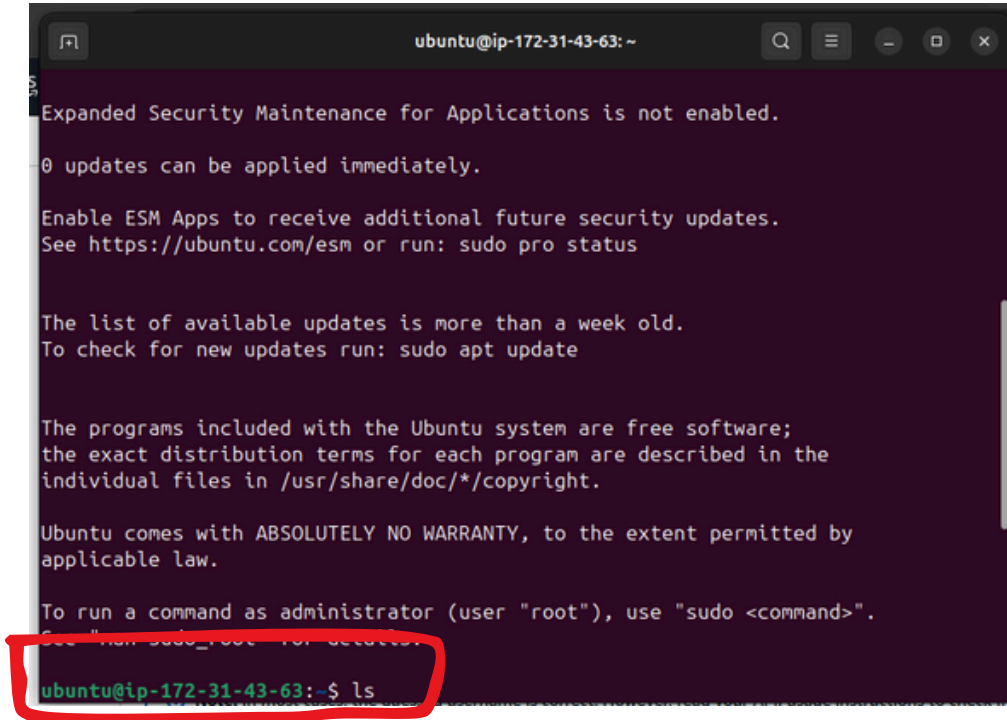
Now open you local machine terminal and connection the EC2 instance.



type yes and press enter to connect.

Connect EC2 with SSH:

Successfully connect EC2 with SSH.

A terminal window titled 'ubuntu@ip-172-31-43-63: ~' showing the output of an SSH connection to an Ubuntu instance. The terminal displays various system messages, including security maintenance status and update information. The prompt 'ubuntu@ip-172-31-43-63:~\$' is highlighted with a red rectangle.

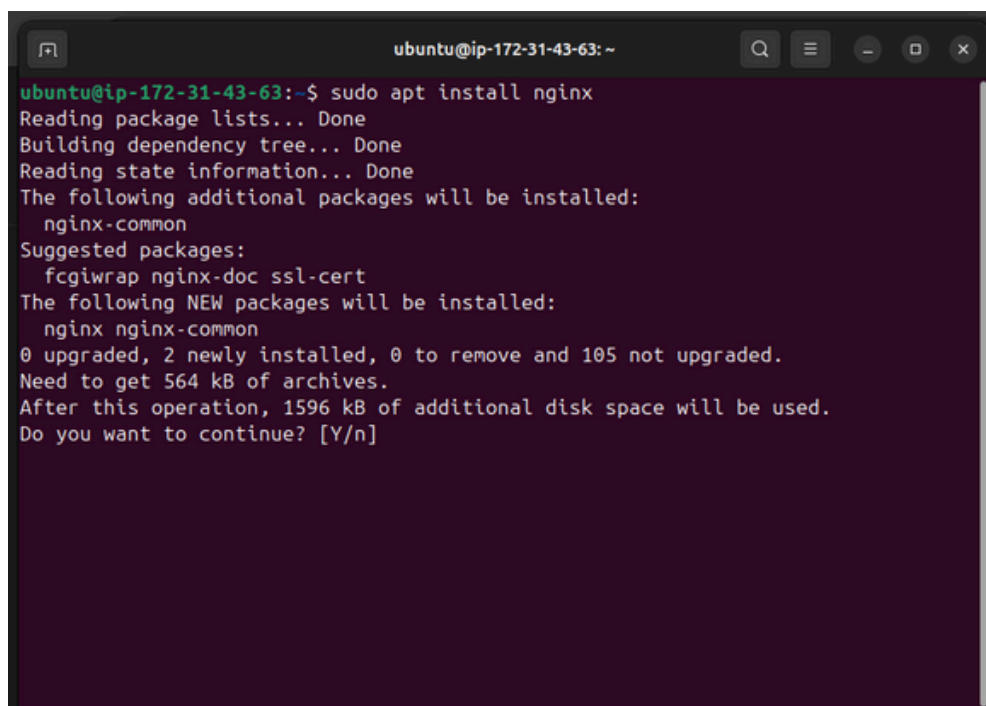
```
ubuntu@ip-172-31-43-63: ~  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-172-31-43-63:~$ ls
```

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

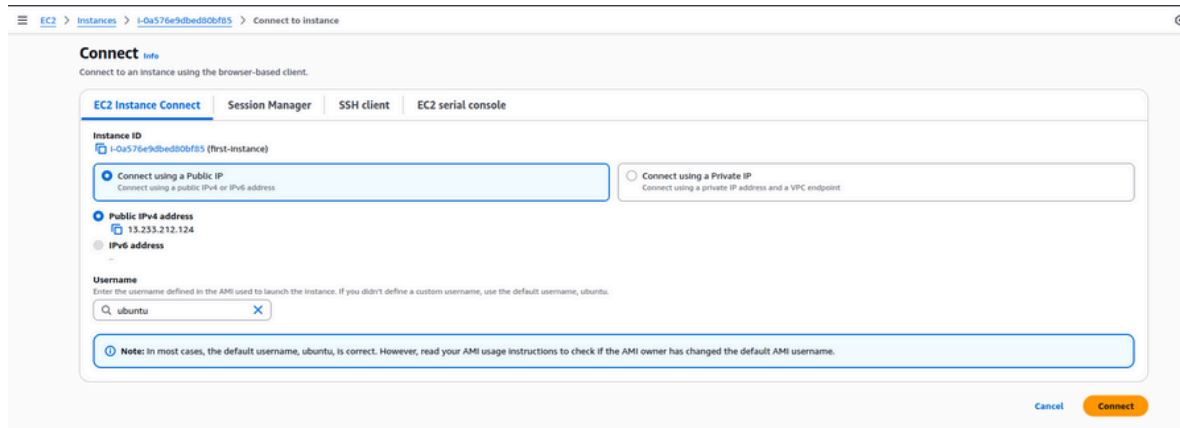
A terminal window with a dark purple background and white text. The window title is 'ubuntu@ip-172-31-43-63: ~'. The command 'sudo apt install nginx' has been entered. The output shows the package lists being read, the dependency tree being built, and state information being read. It lists additional packages to be installed (nginx-common) and suggested packages (fcgiwrap, nginx-doc, ssl-cert). It shows that two new packages (nginx and nginx-common) will be installed, requiring 564 kB of archives and 1596 kB of additional disk space. The prompt 'Do you want to continue? [Y/n]' is visible at the bottom.

```
ubuntu@ip-172-31-43-63:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 105 not upgraded.
Need to get 564 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Type y and press enter for full configuration.

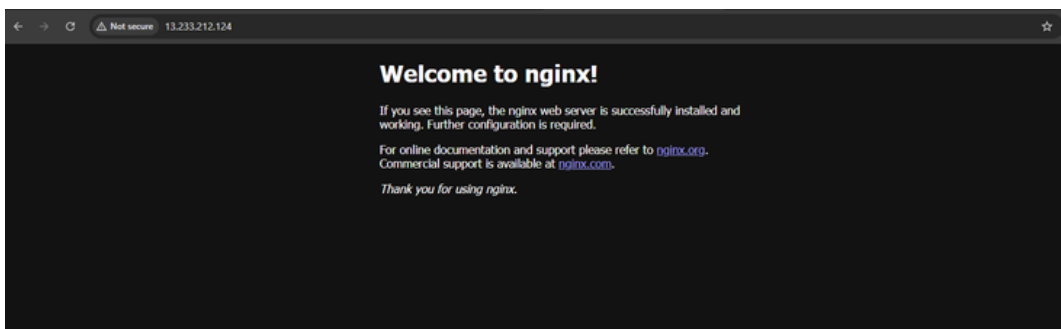
Configure nginx:

It's time to connect EC2 instance and testing.



The screenshot shows the AWS Management Console 'Connect to instance' page. The breadcrumb trail is 'EC2 > Instances > i-0a576e9dbed080bf65 > Connect to instance'. The page title is 'Connect' with an 'Info' icon. Below the title, it says 'Connect to an instance using the browser-based client.' There are four tabs: 'EC2 Instance Connect' (selected), 'Session Manager', 'SSH client', and 'EC2 serial console'. Under 'EC2 Instance Connect', there are two main options: 'Connect using a Public IP' (selected) and 'Connect using a Private IP'. The 'Public IP' option has a sub-option 'Public IPv4 address' selected, with the address '13.233.212.124' entered. Below this is a 'Username' field with 'ubuntu' entered. A note at the bottom states: 'Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' At the bottom right are 'Cancel' and 'Connect' buttons.

Successfully configure the nginx.



Thank You

Stay Connect:

/in/alamgirweb11

/alamgirweb11



AWS VPC

**Virtual Private
Cloud**



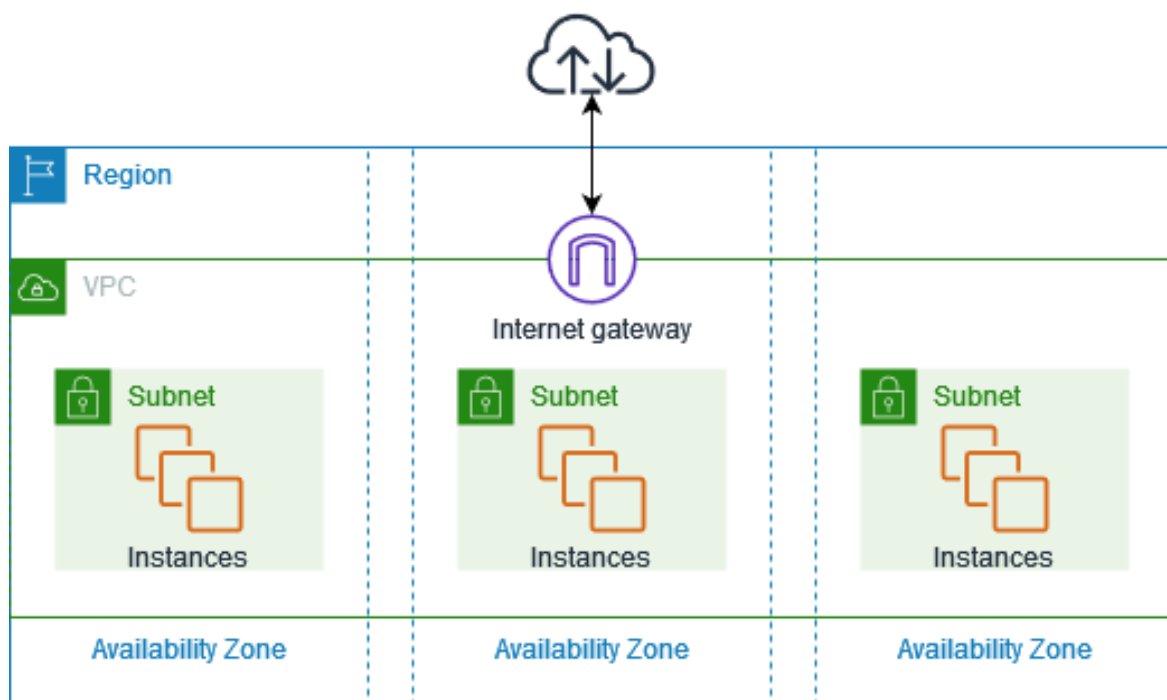
Concept Overview:

Introduction of VPC	1
Subnet	2
Route Tables	3
Internet Gateway	4
NAT Gateway	5
Configure VPC	6
Configure Subnet	7
Configure Route Tables	8

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.

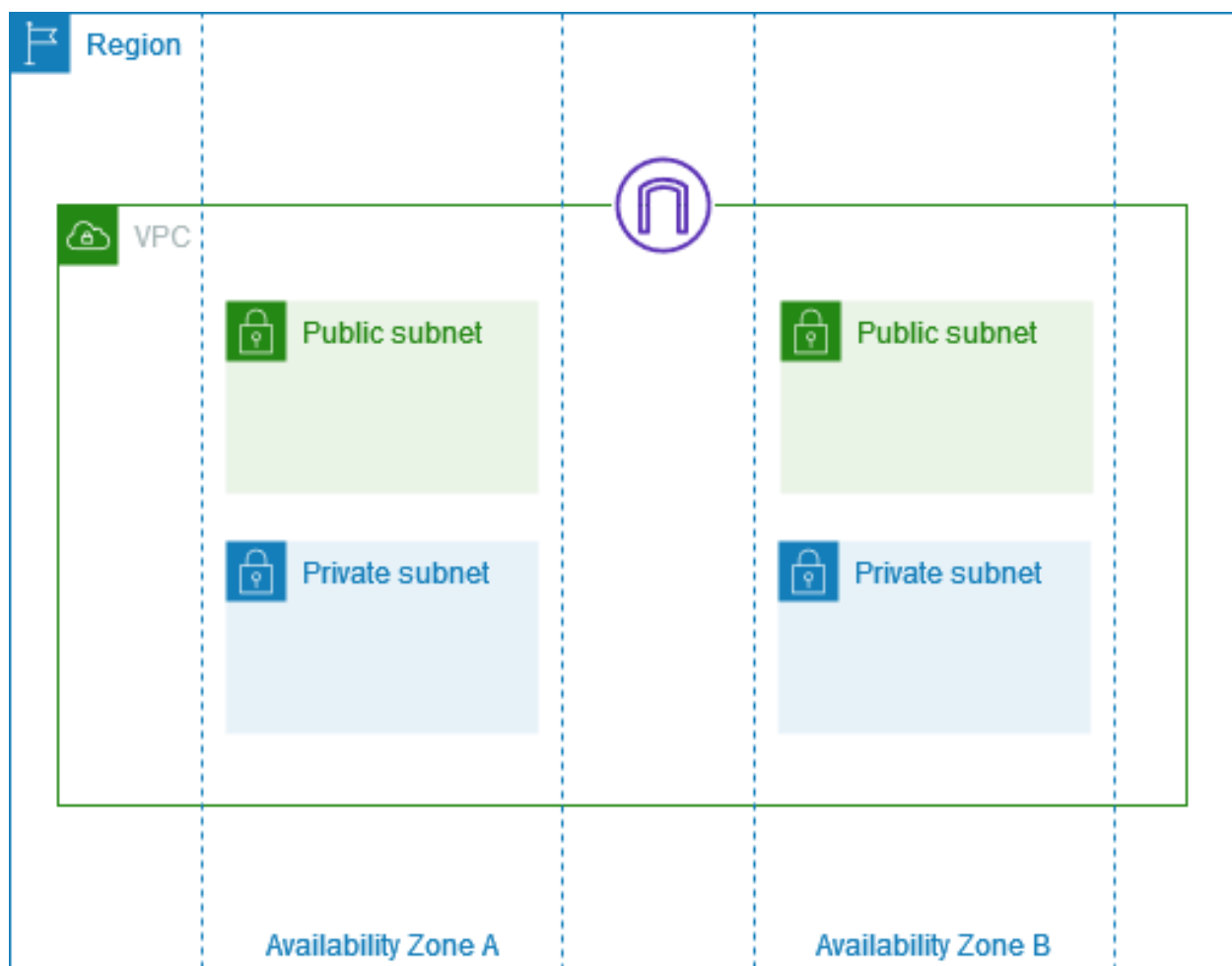
VPN connections:

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:

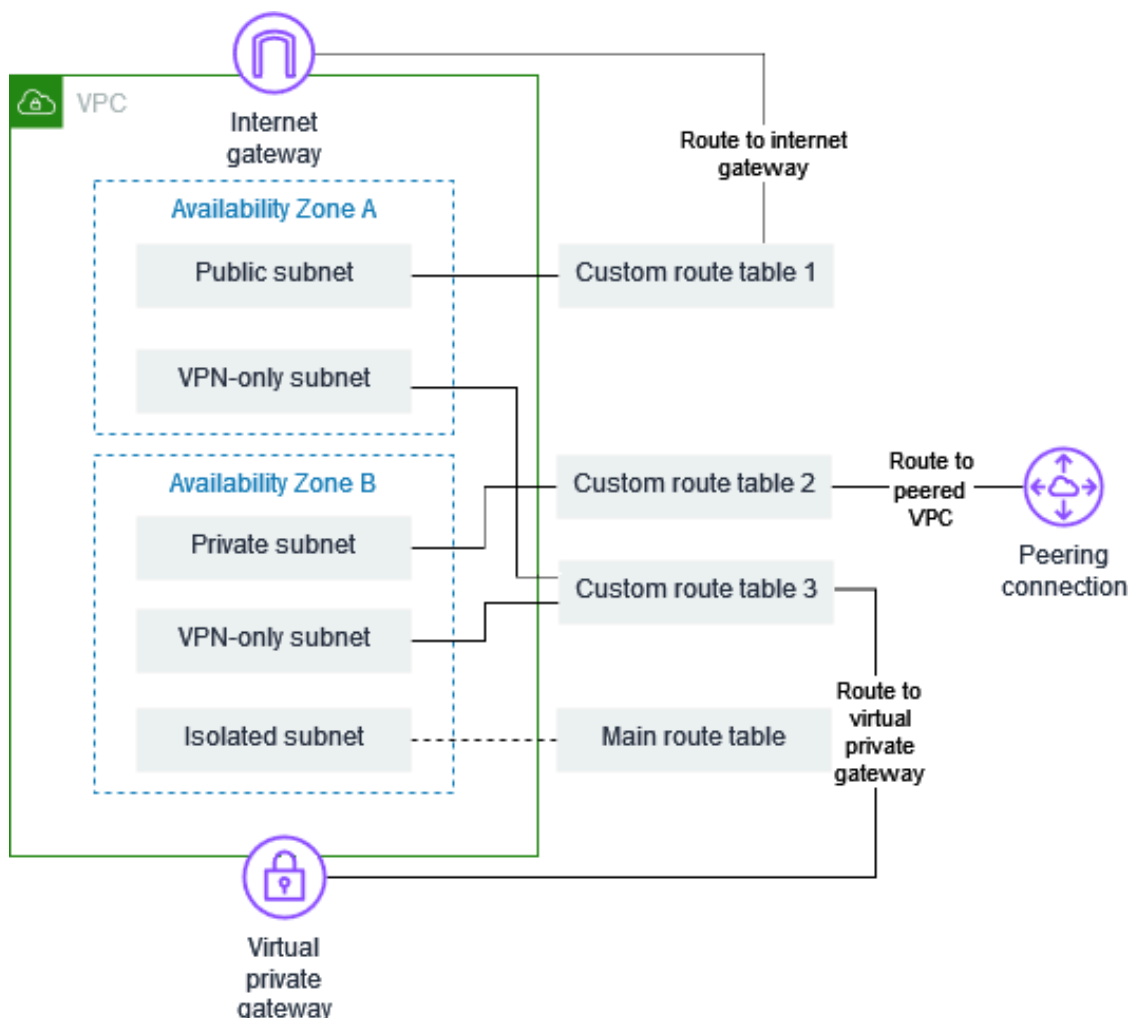
Features of Route Tables:

- **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:

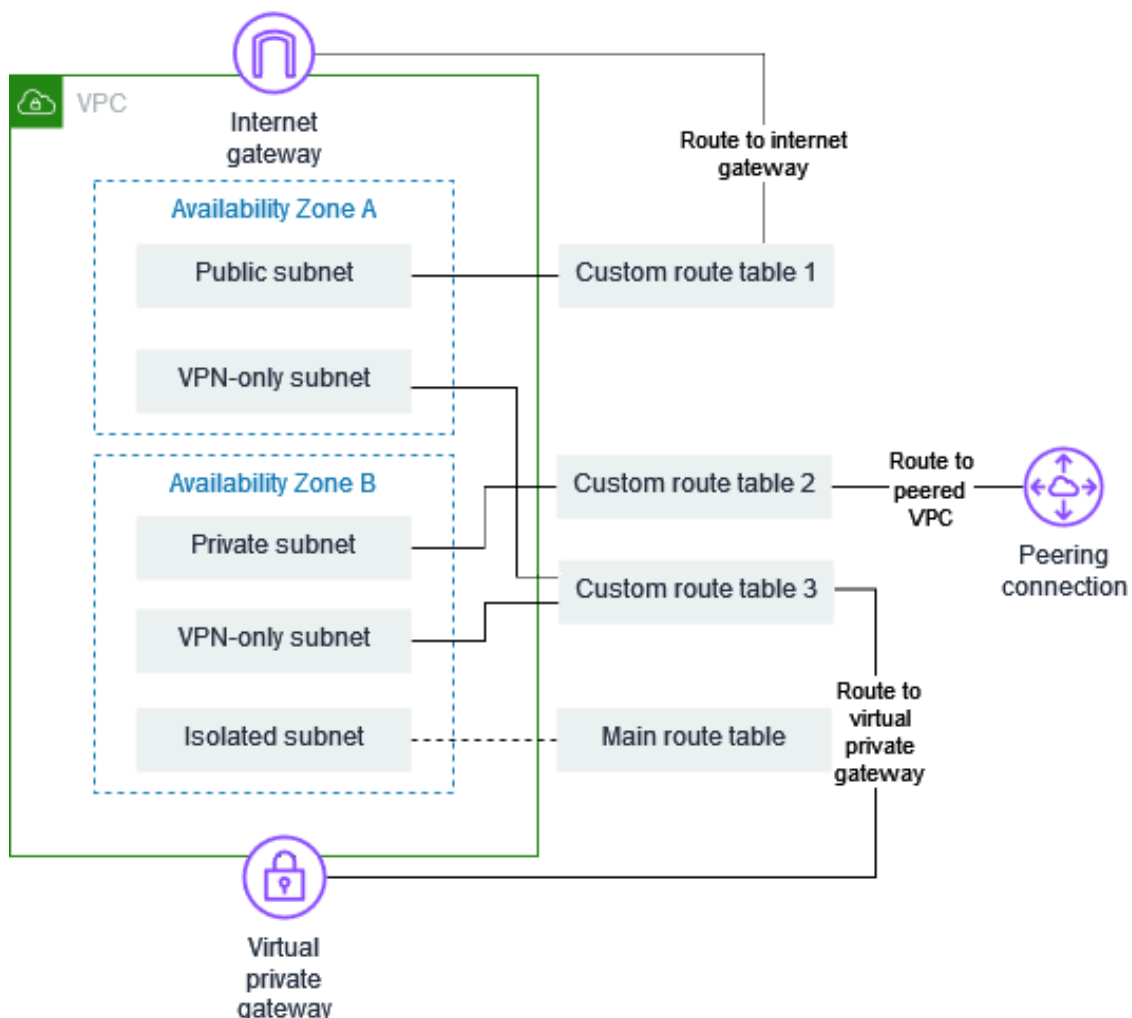
The following diagram shows a VPC with five subnets, a main route table, and three custom route tables. All four route tables have local routes. Custom route table 1 has a route to an internet gateway, and it is associated with the public subnet in Availability Zone A. Custom route table 2 has a route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

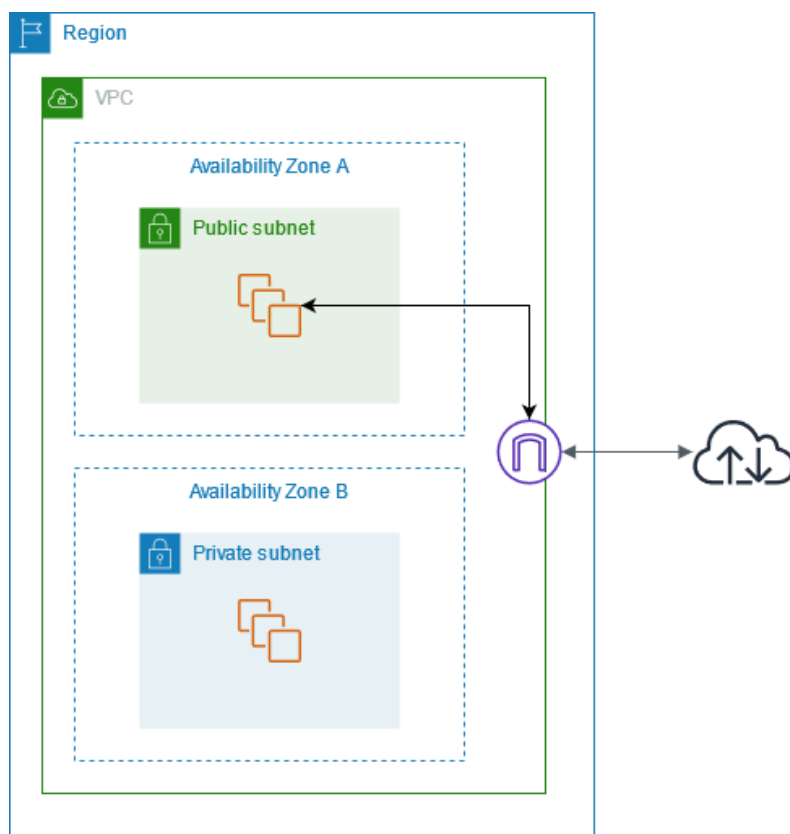
About Internet Gateway:

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

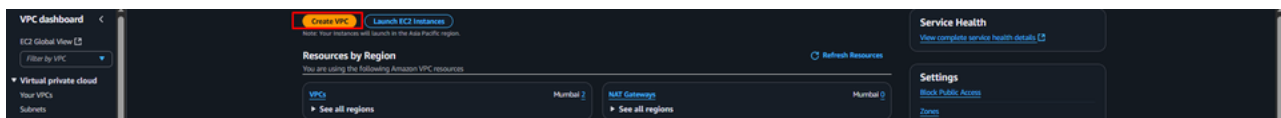
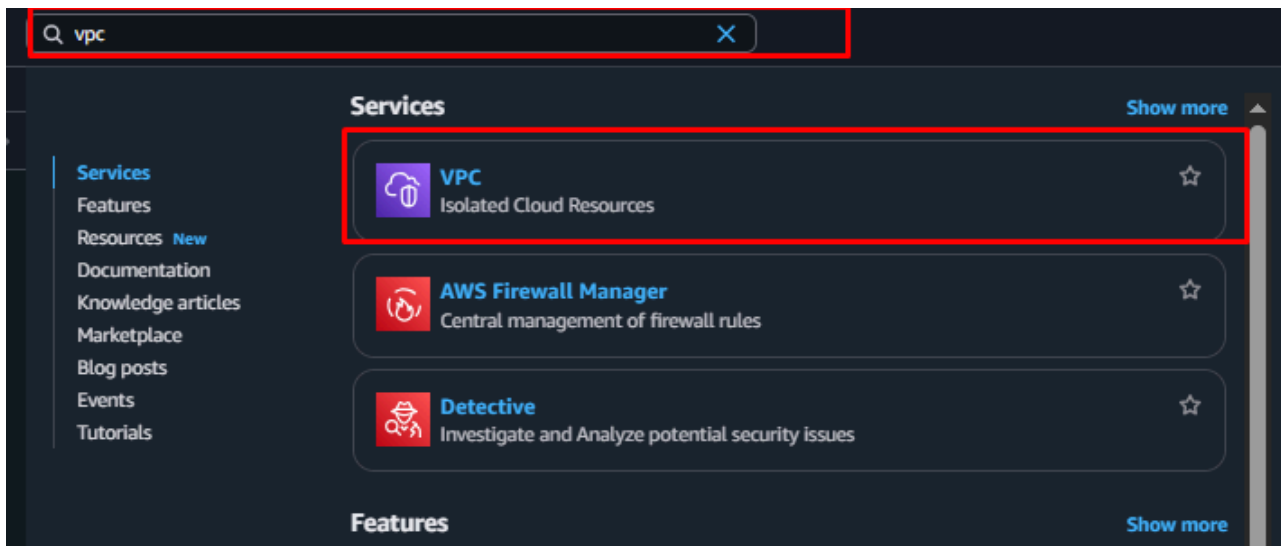
- Placed in a public subnet with an Elastic IP.
- Lets private subnet instances connect to the internet (outbound only).
- Routes through Internet Gateway (IGW), Transit Gateway, or Virtual Private Gateway.
- Cannot receive inbound internet traffic.

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.

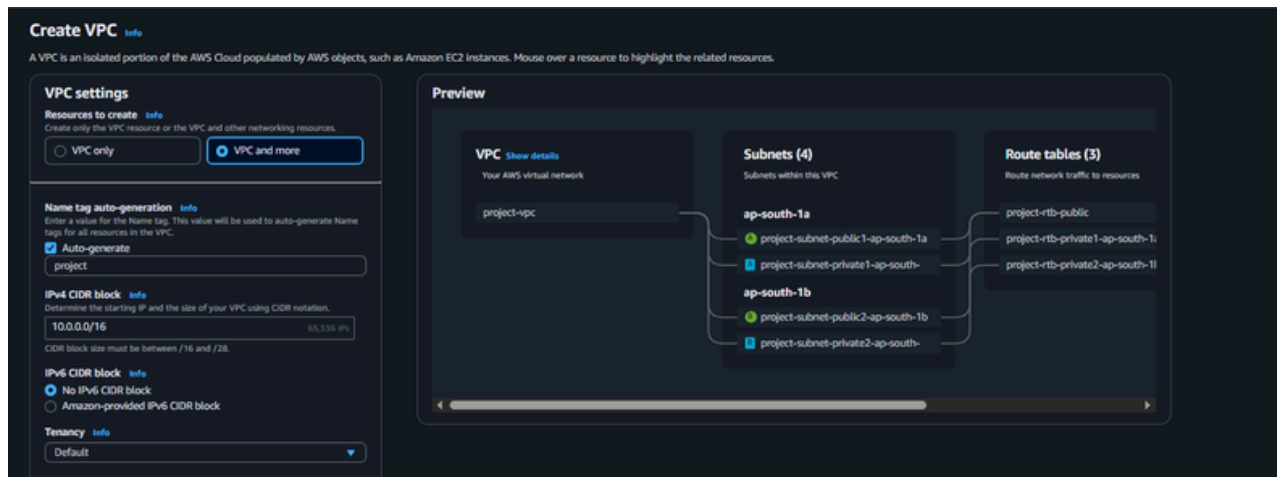


On the VPC creation page, you get two options: create a VPC manually or use the default method.

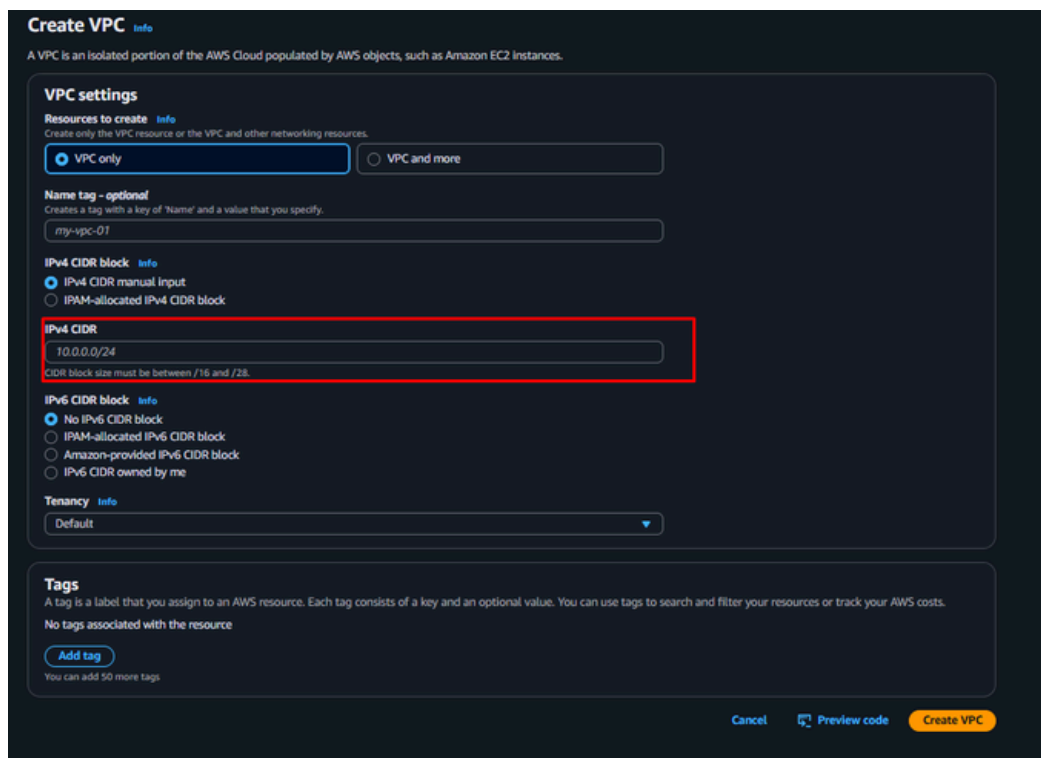
A screenshot of the AWS 'Create VPC' page. The page title is 'Create VPC'. Below the title, there is a section 'VPC settings' with a sub-section 'Resources to create'. This section has two radio buttons: 'VPC only' (selected) and 'VPC and more'. Below this, there is a 'Name tag - optional' field with the value 'my-vpc-01'. Then, there is a section for 'IPv4 CIDR block' with two radio buttons: 'IPv4 CIDR manual input' (selected) and 'IPAM-allocated IPv4 CIDR block'. Below this, there is a field for 'IPv4 CIDR' with the value '10.0.0.0/24'. Then, there is a section for 'IPv6 CIDR block' with four radio buttons: 'No IPv6 CIDR block' (selected), 'IPAM-allocated IPv6 CIDR block', 'Amazon-provided IPv6 CIDR block', and 'IPv6 CIDR owned by me'. Below this, there is a 'Tenancy' dropdown menu with the value 'Default'. At the bottom, there is a 'Tags' section with an 'Add tag' button. The 'Create VPC' button is highlighted with a red box.

Configure VPC:

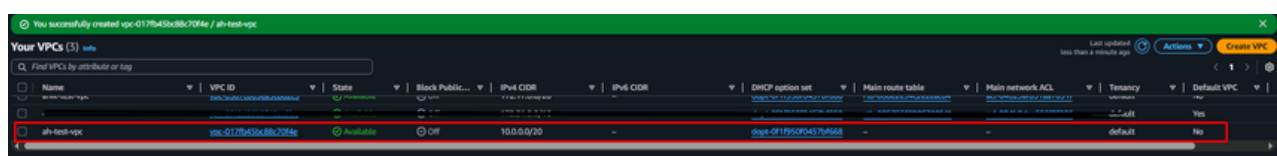
The default way looks like this, but I chose to create the VPC manually for better understanding.



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

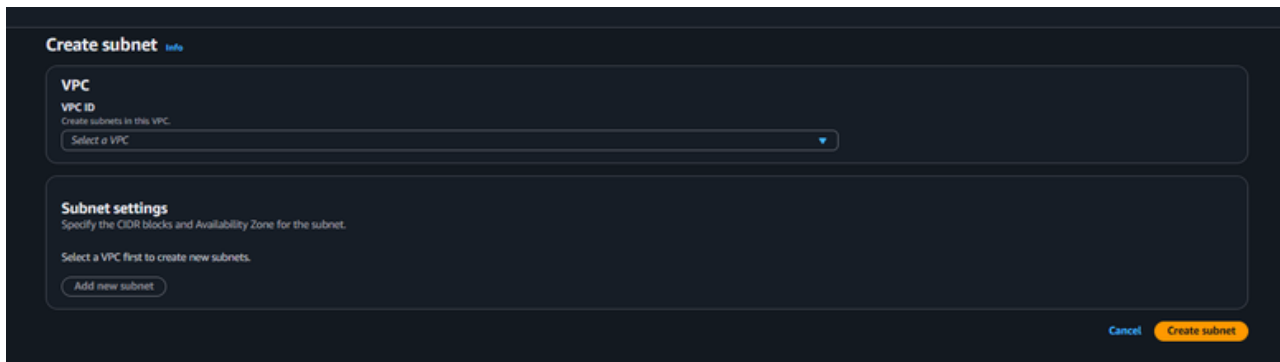


Here the VPC:

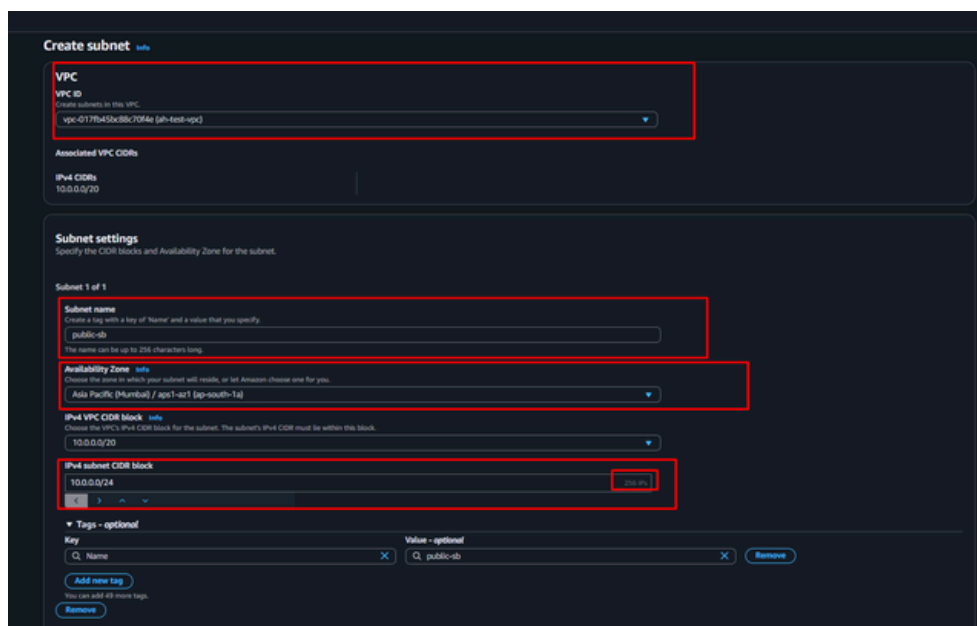


Configure Subnet:

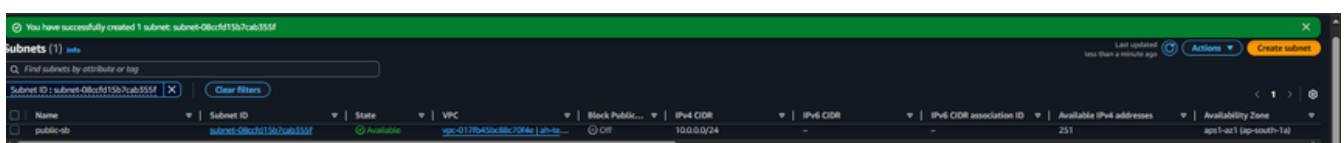
Now, configure two subnets: one Public and one Private. Make sure to select the newly created VPC.



Here the important config input fields.



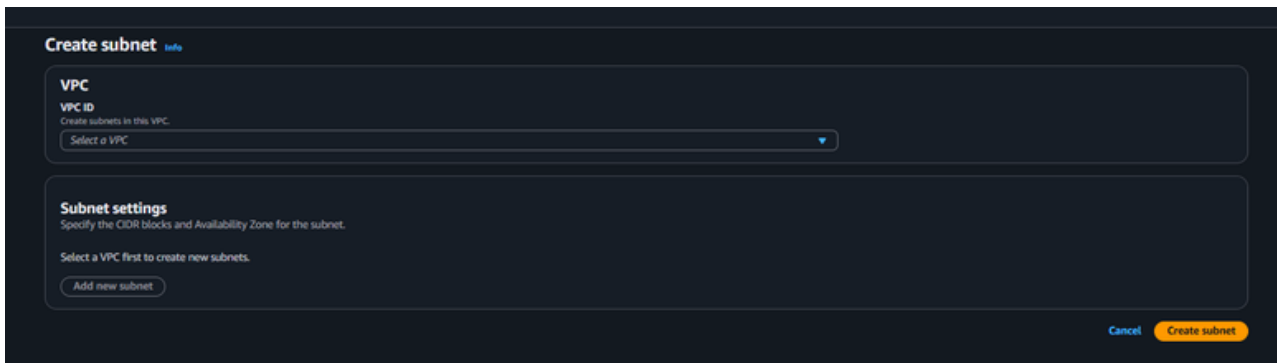
Here the Public subnet:



Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID	Available IPv4 addresses	Availability Zone
public-sb	subnet-086cfd15b7cab355f	Available	vpc-017f645b8c70f4e ph-test-vpc	Off	10.0.0/24	-	-	251	ap-south-1a

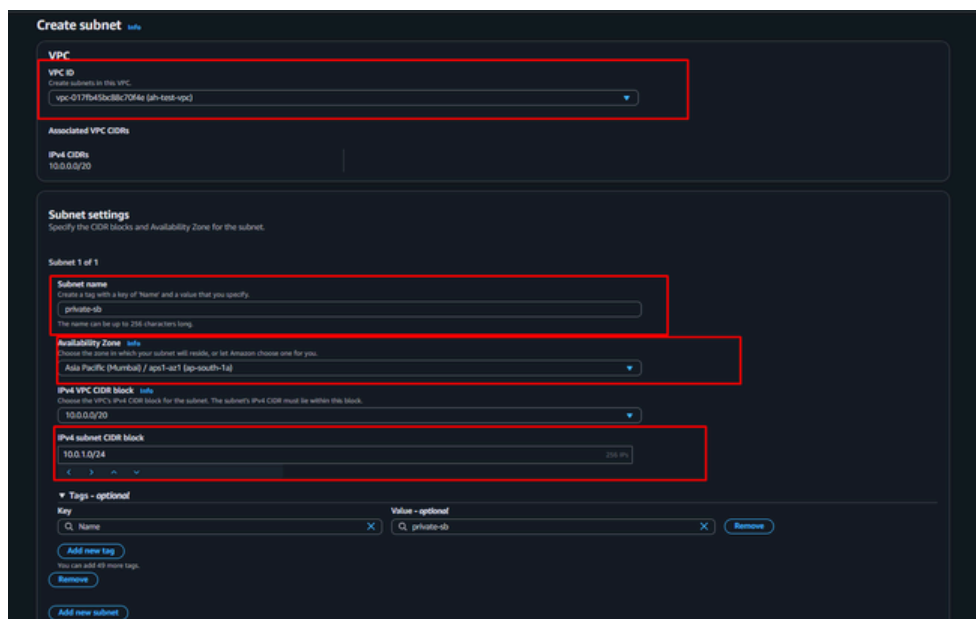
Configure Subnet:

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.



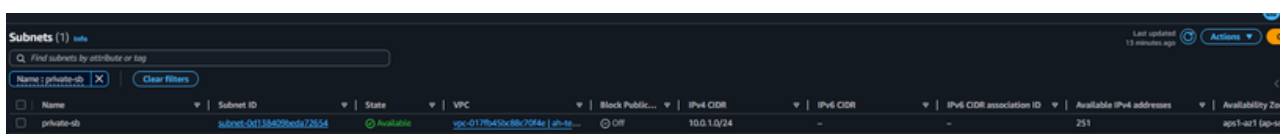
The screenshot shows the 'Create subnet' form in the AWS Management Console. It has two main sections: 'VPC' and 'Subnet settings'. In the 'VPC' section, the 'VPC ID' is set to 'vpc-017b453c8b79f4e' (ap-test-vpc). The 'Subnet settings' section is currently empty, with a note to 'Specify the CIDR blocks and Availability Zone for the subnet' and a button to 'Add new subnet'. At the bottom right, there are 'Cancel' and 'Create subnet' buttons.

Here the important config input fields.



This screenshot highlights the key configuration fields in the 'Create subnet' form with red boxes. The highlighted fields include: 'VPC ID' (vpc-017b453c8b79f4e), 'Subnet name' (private-db), 'Availability Zone' (Asia Pacific (Mumbai) / ap-south-1a), 'IPv4 VPC CIDR block' (10.0.0.0/20), and 'IPv4 subnet CIDR block' (10.0.1.0/24). The 'Tags' section is also visible at the bottom, showing a key 'Name' with value 'private-db'.

Here the Private subnet:



The screenshot shows the 'Subnets' list in the AWS Management Console. It displays a table with columns: Name, Subnet ID, State, VPC, Block Public..., IPv4 CIDR, IPv6 CIDR, IPv4 CIDR association ID, Available IPv4 addresses, and Availability Zone. The table contains one entry: 'private-db' with Subnet ID 'subnet-04138420b0a72654', State 'Available', VPC 'vpc-017b453c8b79f4e', Block Public... 'Off', IPv4 CIDR '10.0.1.0/24', IPv6 CIDR '-', IPv4 CIDR association ID '-', Available IPv4 addresses '251', and Availability Zone 'ap-south-1a'.

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv4 CIDR association ID	Available IPv4 addresses	Availability Zone
private-db	subnet-04138420b0a72654	Available	vpc-017b453c8b79f4e	Off	10.0.1.0/24	-	-	251	ap-south-1a

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.

The 'Create route table' form shows the following details:

- Route table settings**
 - Name - optional:** Create a tag with a key of 'Name' and a value that you specify. Value: `my-route-table-01`
 - VPC:** The VPC to use for this route table. Select a VPC: `ah-test-vpc`
- 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.
 - No tags associated with the resource.
 - [Add new tag](#)
 - You can add 50 more tags.

Buttons: [Cancel](#), [Create route table](#)

Here the two Route Tables:

<input type="checkbox"/>	public-rt	rtb-07663942a94dc881	-	-	No	vpc-017fb450c88c70f4e ah-te...	515966537504
<input type="checkbox"/>	private-rt	rtb-057fb709ff4c0c962	-	-	No	vpc-017fb450c88c70f4e ah-te...	515966537504

Now, associate with two difference subnets.

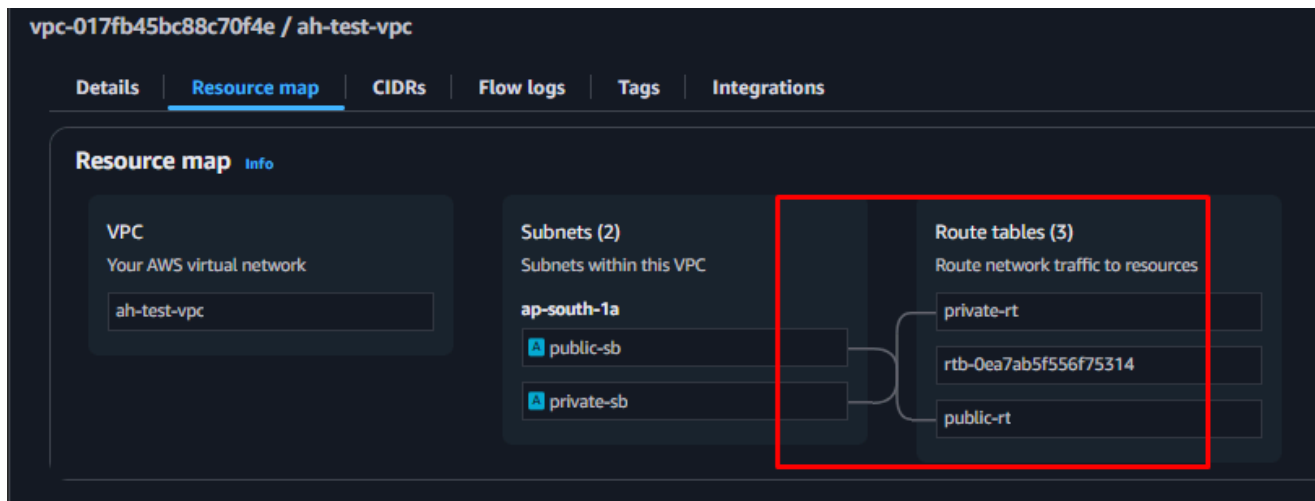
The 'Edit subnet associations' form shows the following details:

- Available subnets (1/2)**
 - Filter subnet associations:
 - Table with columns: Name, Subnet ID, IPv4 CIDR, IPv6 CIDR
 - public-sb: subnet-08ccfd15b7cab355f, 10.0.0.0/24, -
 - private-sb: subnet-0d158409bed72654, 10.0.1.0/24, -
- Selected subnets**
 - subnet-08ccfd15b7cab355f / public-sb

Buttons: [Cancel](#), [Save associations](#)

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/in/alamgirweb11](https://www.linkedin.com/in/alamgirweb11)

 [/alamgirweb11](https://github.com/alamgirweb11)