

## **Placement Empowerment Program**

### ***Cloud Computing and DevOps Centre***

***Set a private network in cloud – Create a VPC with subnets for your instances. Configure routing for internal communication between subnets***

Name :KEERTHIKA ANANTHAN

Department: AML

# Introduction

A Virtual Private Cloud (VPC) is a secure and isolated portion of a cloud provider's infrastructure where you can deploy your resources in a controlled environment. Setting up a VPC involves creating subnets, configuring routing, and implementing security measures to manage traffic and access. This setup is essential for applications that require secure internal communication while being accessible to external networks when necessary.

## Objectives

1. **Create a VPC:** Establish a private network in the cloud that suits your application requirements.
2. **Configure Subnets:** Design and implement subnets within the VPC for different types of instances (e.g., public and private).
3. **Set Up Routing:** Configure routing tables to enable internal communication between subnets and external access as required.
4. **Implement Security:** Use security groups and network ACLs to control inbound and outbound traffic to your instances.
5. **Ensure High Availability:** Distribute resources across multiple Availability Zones to enhance resilience

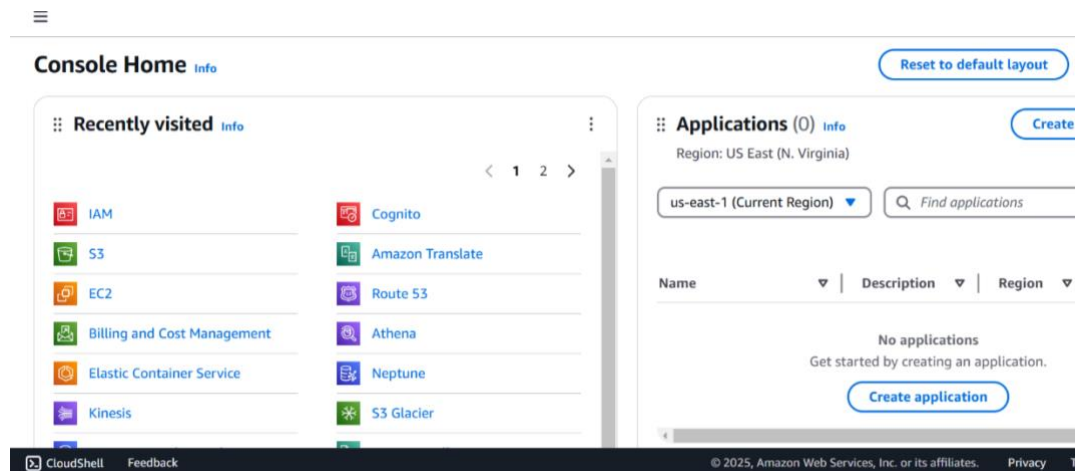
# Importance

- **Security:** A VPC allows you to maintain a secure environment, isolating your resources from public internet exposure while enabling controlled access.
- **Customization:** You can tailor the network architecture to meet specific needs, such as private IP addressing and subnet segmentation.
- **Cost Efficiency:** Efficiently using cloud resources helps in managing costs associated with data transfer and resource allocation.
- **Scalability:** Easily scale your infrastructure to accommodate growing workloads without compromising security or performance.
- **Control:** Gain complete control over the networking environment, including IP address ranges, routing, and access controls.

# Step-by-Step Overview

## Step 1:

1. Go to [AWS Management Console](#).
2. Enter your username and password to log in



## Step 2:

### Navigate to the VPC Dashboard

- In the Services menu, select "VPC" to access the VPC Dashboard.
- 

### Create a VPC

- Click on "Your VPCs" in the left menu, then click "Create VPC."
- Specify the following:
  - **Name tag:** A name for your VPC.
  - **IPv4 CIDR block:** E.g., 10.0.0.0/16 (this gives you 65,536 IP addresses).
  - **IPv6 CIDR block:** (Optional).
  - **Tenancy:** Default is usually sufficient.
- Click "Create."

[Create VPC](#)[Launch EC2 Instances](#)

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

## Resources by Region

[Refresh Resources](#)

You are using the following Amazon VPC resources

[VPCs](#)US East [1](#)[▶ See all regions](#)[NAT Gateways](#)US East [0](#)[▶ See all regions](#)[VPC](#) > [Your VPCs](#) > Create VPC

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.

### IPv4 CIDR block

[Info](#)☒ IPv4 CIDR manual input☐ IPAM-allocated IPv4 CIDR block

### IPv4 CIDR

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](#)

### 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

[Remove tag](#)[Add tag](#)

## Step 3: Create Subnets

**You need at least two private subnets for internal communication:**

**1. Go to Subnets → Click Create Subnet.**

**2. Select the VPC (MyPrivateVPC) you created earlier.**

**3. Create two subnets:**

**Subnet 1 (Private-Subnet-A)**

**IPv4 CIDR: 10.0.1.0/24**

**Availability Zone: us-east-1a (example)**

**Subnet 2 (Private-Subnet-B)**

**IPv4 CIDR: 10.0.2.0/24**

Subnet 2 of 2

Subnet name

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

sub-2

The name can be up to 256 characters long.

Availability Zone

Info

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

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

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.

10.0.0.0/24

IPv4 subnet CIDR block

10.1.0.0/16

65,536 IPs

▼ Tags - optional

Key

Q Name

Value - optional

Q sub-2

Remove

Add new tag

You can add 49 more tags.

Remove

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 2

Subnet name

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

sub-1

The name can be up to 256 characters long.

Availability Zone

Info

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

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

IPv4 VPC CIDR block

Info

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

10.0.0.0/24

IPv4 subnet CIDR block

10.0.0.0/16

65,536 IPs

▼ Tags - optional

Key

Q Name

Value - optional

Q sub-1

Remove

Add new tag

You can add 49 more tags.

## Step 4:

### Configure Route Tables for Internal Communication

1. Go to Route Tables → Click Create Route Table.
2. Name it (e.g., PrivateRouteTable).
3. Select MyPrivateVPC.
4. Click Create.

The screenshot shows the 'Create route table' page in the AWS Management Console. The breadcrumb navigation at the top reads 'VPC > Route tables > Create route table'. The page title is 'Create route table' with an 'Info' link. A descriptive text states: 'A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.'

The 'Route table settings' section contains two fields: 'Name - optional' with the value 'private' and 'VPC' with a dropdown menu showing 'vpc-0b07dbbc4d9e68588 (vpc-1)'. Below this is the 'Tags' section, which includes a description: '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.' It features a table with two columns: 'Key' and 'Value - optional'. The first row has 'Name' as the key and 'private' as the value. There are 'Add new tag' and 'Remove' buttons. At the bottom right, there are 'Cancel' and 'Create route table' buttons.

## Step 5:

### Associate the subnets:

- Go to Subnet Associations → Click Edit subnet associations.
- Select Private-Subnet-A and Private-Subnet-B.
- Click Save associations.

VPC vpc-0b07dbbc4d9e68588 | vpc-1 Owner ID 774305605711

Routes **Subnet associations** Edge associations Route propagation Tags

Explicit subnet associations (0) Edit subnet associations

Find subnet association

| Name                   | Subnet ID | IPv4 CIDR | IPv6 CIDR |
|------------------------|-----------|-----------|-----------|
| No subnet associations |           |           |           |

You do not have any subnet associations.

### Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/2) Filter subnet associations < 1 > ⚙

| <input checked="" type="checkbox"/> | Name  | Subnet ID                | IPv4 CIDR   | IPv6 CIDR | Route table ID               |
|-------------------------------------|-------|--------------------------|-------------|-----------|------------------------------|
| <input checked="" type="checkbox"/> | sub-2 | subnet-08d686eb3bfda3c1c | 10.0.2.0/24 | -         | Main (rtb-0511a15ded68d344d) |
| <input checked="" type="checkbox"/> | sub-1 | subnet-0a23be0f9dc2a24aa | 10.0.1.0/24 | -         | Main (rtb-0511a15ded68d344d) |

Selected subnets

subnet-08d686eb3bfda3c1c / sub-2 × subnet-0a23be0f9dc2a24aa / sub-1 ×

Cancel Save associations

## Step 6:

Default route: 10.0.0.0/16 → local (Automatically added).

rtb-09bd5c6927b161264 / private Actions

**Details** [Info](#)

|   |                          |   |                        |
|---|--------------------------|---|------------------------|
| Route table ID<br>rtb-09bd5c6927b161264 | Main<br>No               | Explicit subnet associations<br>2 subnets | Edge associations<br>- |
| VPC<br>vpc-0b07dbbc4d9e68588   vpc-1    | Owner ID<br>774305605711 |   |                        |

Routes **Subnet associations** Edge associations Route propagation Tags

Routes (1) Both Edit routes

Filter routes

| Destination | Target | Status | Propagated |
|-------------|--------|--------|------------|
| 10.0.0.0/16 | local  | Active | No         |

## Step 7:

### Launch Instances in Private Subnets

1. Go to EC2 Dashboard → Launch Instance.
2. Select an AMI (Amazon Linux, Ubuntu, etc.).
3. Choose an Instance Type (e.g., t2.micro).



4. Under Network settings:

Select MyPrivateVPC.

Select Private Subnet-A or Private-Subnet-B.

Disable Auto-assign Public IP (to keep it private).

EC2 > Instances > Launch an instance

### 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

vpc-1 [Add additional tags](#)

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

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

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

Recents | My AMIs | **Quick Start**

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

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

##### Amazon Machine Image (AMI)

Amazon Linux 2023 AMI  
ami-0c614dee691cbbf37 (64-bit (x86), uefi-preferred) / ami-0b29c89c15cfb8a6d (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

#### ▼ Summary

Number of instances: 1

Software: Amazon Linux

Virtual server type: t2.micro

Firewall (security group): default

Storage (volumes): 1 volume(s)

Free tier eligible (or t2.micro instances in public subnets)

[Cancel](#)

## Step 8:

Enable Internal Communication

Instances inside the private subnets can communicate without an internet gateway.

If instances need internet access (for updates, etc.), configure a NAT Gateway in a Public Subnet.

Use Security Groups to allow inbound traffic only from internal sources (e.g., allow SSH from 10.0.0.0/16).

## Step 9:

Now, your private network is set up, and instances inside can communicate securely! Let me know if you need extra configurations like VPN, Bastion Host, or NAT setup.

## Outcome

After following these steps, you will have:

- A VPC that is isolated from other networks.
- One or more subnets for your instances, with at least one public subnet that can communicate with the Internet.
- Proper routing configured for internal communication between subnets.

