

## **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 :Kowsika V

Department: CSE

# 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

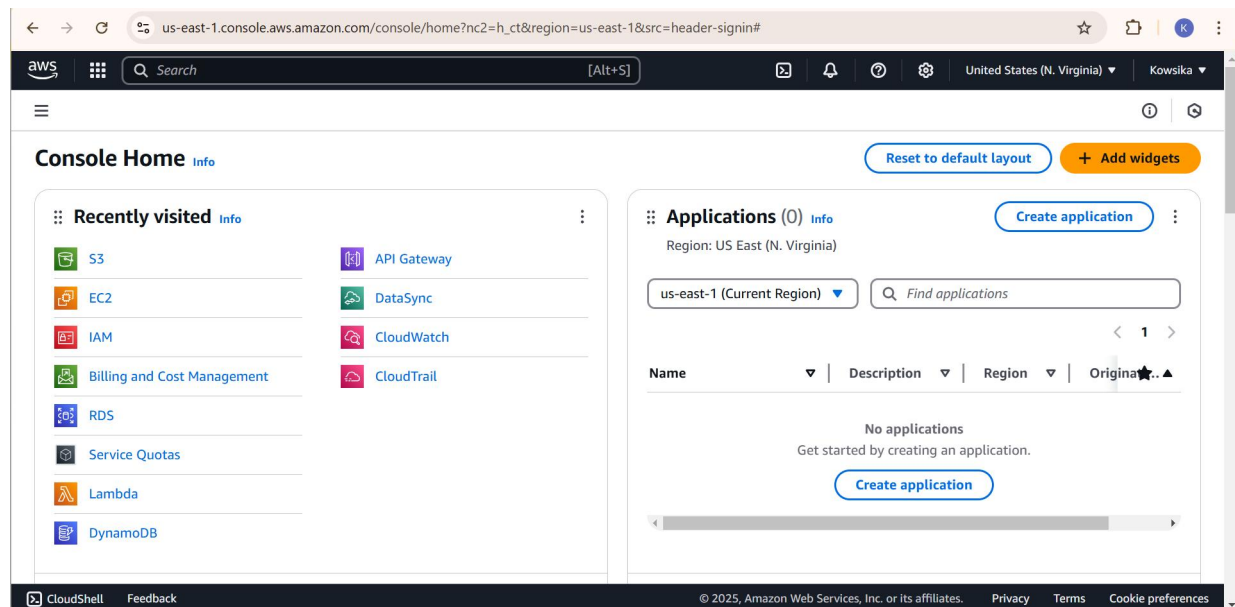
# Importance

- **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 subnetwork 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

<a href="#">VPCs</a> US East <a href="#">1</a> ▶ <a href="#">See all regions</a>	<a href="#">NAT Gateways</a> US East <a href="#">0</a> ▶ <a href="#">See all regions</a>
---	---

The screenshot shows the AWS Management Console 'Create VPC' page. The browser address bar shows 'us-east-1.console.aws.amazon.com/vpconsole/home?region=us-east-1#CreateVpc:createMode=vpcOnly'. The page has a dark header with the AWS logo, a search bar, and navigation icons. Below the header, the breadcrumb trail is 'VPC > Your VPCs > Create VPC'. The main content area has two radio buttons: 'VPC only' (selected) and 'VPC and more'. Below this is the 'Name tag - optional' section, which says 'Creates a tag with a key of 'Name' and a value that you specify.' and has a text input field containing 'my-vpc-01'. The next section is 'IPv4 CIDR block', with a link to 'Info'. It has two radio buttons: 'IPv4 CIDR manual input' (selected) and 'IPAM-allocated IPv4 CIDR block'. Below this is a text input field for the CIDR block, containing '10.0.0.0/24'. A note below the field says 'CIDR block size must be between /16 and /28.'. The next section is 'IPv6 CIDR block', with a link to 'Info'. It has four radio buttons: 'No IPv6 CIDR block' (selected), 'IPAM-allocated IPv6 CIDR block', 'Amazon-provided IPv6 CIDR block', and 'IPv6 CIDR owned by me'. At the bottom, there is a 'Tenancy' section with a link to 'Info'. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

## 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**

The screenshot shows the 'Create subnet' page in the AWS Management Console for 'Subnet 1 of 2'. The page is titled 'VPC > Subnets > Create subnet'. The 'Subnet name' field is 'sub\_01'. The 'Availability Zone' is 'US East (N. Virginia) / us-east-1a'. The 'IPv4 VPC CIDR block' is '10.0.0.0/24'. The 'IPv4 subnet CIDR block' is '10.0.0.0/28' with a range of '16 IPs'. The page includes a search bar, navigation icons, and a footer with '© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

aws [Search] [Alt+S] United States (N. Virginia) Kowsika

VPC > Subnets > Create subnet

Subnet 1 of 2

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
sub\_01  
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**  
10.0.0.0/28 16 IPs

< > ^ v

Tags - optional

The screenshot shows the 'Create subnet' page in the AWS Management Console for 'Subnet 2 of 2'. The page is titled 'VPC > Subnets > Create subnet'. The 'Subnet name' field is 'sub\_02'. The 'Availability Zone' is 'US East (N. Virginia) / us-east-1b'. The 'IPv4 VPC CIDR block' is '10.0.0.0/24'. The 'IPv4 subnet CIDR block' is '10.0.0.0/28' with a range of '16 IPs'. The page includes a search bar, navigation icons, and a footer with '© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

aws [Search] [Alt+S] United States (N. Virginia) Kowsika

VPC > Subnets > Create subnet

Subnet 2 of 2

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
sub\_02  
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**  
10.0.0.0/28 16 IPs

< > ^ v

Tags - optional

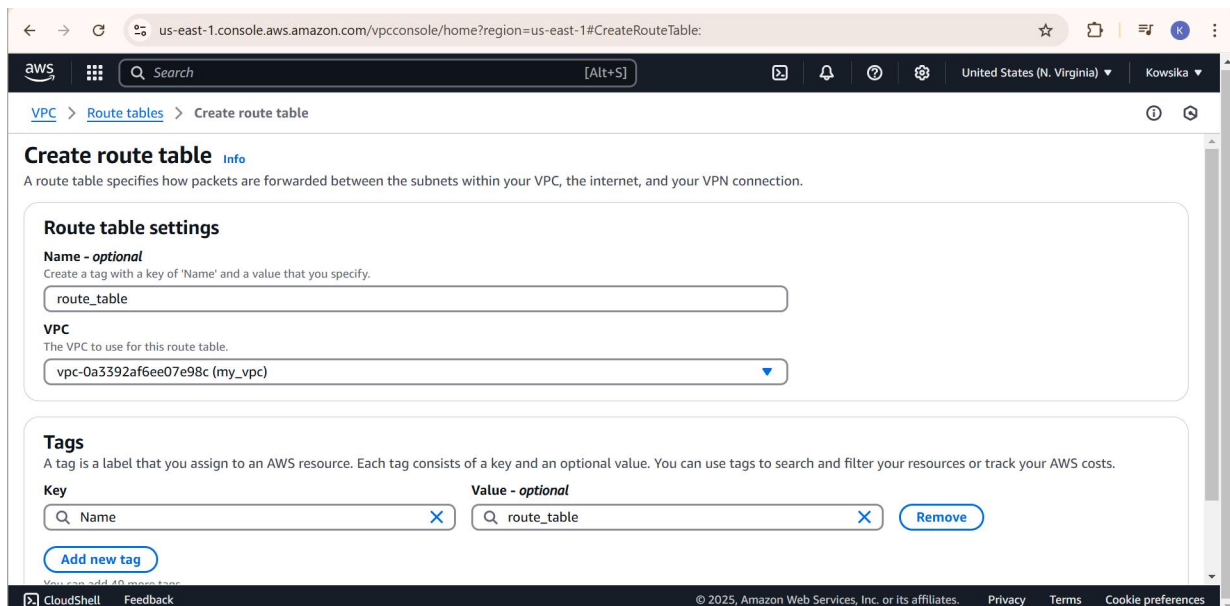
Key Value - optional

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 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 AWS Management Console interface for creating a new route table. The browser address bar indicates the URL: `us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateRouteTable:`. The console header shows the AWS logo, a search bar, and navigation icons. The breadcrumb trail is `VPC > Route tables > Create route table`. The main heading is **Create route table** with an `Info` link. Below this is a descriptive sentence: "A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection." The form is divided into two main sections: **Route table settings** and **Tags**. In the **Route table settings** section, there is a **Name - optional** field with the value `route_table` and a **VPC** dropdown menu showing `vpc-0a3392af6ee07e98c (my_vpc)`. The **Tags** section explains that a tag is a label for an AWS resource and shows a table with one tag: 

Key	Value - optional	
Name	route_table	Remove

 Below the table is an `Add new tag` button. The footer of the console includes `CloudShell`, `Feedback`, and copyright information for Amazon Web Services, Inc. (2025).

## Step 5:

### Associate the subnets:

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

aws

Search

[Alt+S]

United States (N. Virginia)

Kowsika

VPC > Route tables > rtb-07c56a490ef8df344 > Edit subnet associations

### Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/2)

Filter subnet associations

<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	sub_02	<a href="#">subnet-042fd6bd92a5e94b9</a>	10.0.0.0/28	-	<a href="#">Main (rtb-036302607763c48cd)</a>
<input checked="" type="checkbox"/>	sub_01	<a href="#">subnet-03c005f33bbb6e6ff</a>	10.0.0.16/28	-	<a href="#">Main (rtb-036302607763c48cd)</a>

Selected subnets

subnet-042fd6bd92a5e94b9 / sub\_02

subnet-03c005f33bbb6e6ff / sub\_01

Cancel

Save associations

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

### Explicit subnet associations (0)

Find subnet association

< 1 >

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
No subnet associations			
You do not have any subnet associations.			

Edit subnet associations

### Subnets without explicit associations (2)

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

Find subnet association

< 1 >

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
sub_02	<a href="#">subnet-042fd6bd92a5e94b9</a>	10.0.0.0/28	-
sub_01	<a href="#">subnet-03c005f33bbb6e6ff</a>	10.0.0.16/28	-

Edit subnet associations

Step 6:

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

Details

Info

Route table ID

rtb-07c56a490ef8df344

VPC

[vpc-0a3392af6ee07e98c](#) | my\_vpc

Main

No

Owner ID

476114141910

Explicit subnet associations

[2 subnets](#)

Edge associations

-

Routes

Subnet associations

Edge associations

Route propagation

Tags

### Routes (1)

Filter routes

Both

Edit routes

< 1 >

Destination	Target	Status	Propagated
10.0.0.0/24	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).

The screenshot displays the AWS Management Console interface for launching an EC2 instance. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:`. The console header includes the AWS logo, a search bar, and navigation icons. The main content area is titled "Launch an instance" and is divided into two columns.

**Left Column (Network Settings):**

- VPC - required:** A dropdown menu shows `vpc-0a3392af6ee07e98c (my_vpc)` with a subnet of `10.0.0.0/24`.
- Subnet:** A dropdown menu shows `subnet-042fd6bd92a5e94b9` with details: VPC: `vpc-0a3392af6ee07e98c`, Owner: `476114141910`, Availability Zone: `us-east-1b`, Zone type: `Availability Zone`, IP addresses available: `11`, CIDR: `10.0.0.0/28`. A "Create new subnet" link is available.
- Auto-assign public IP:** A dropdown menu shows `Disable`.
- Firewall (security groups):** A section with a description: "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." It contains two radio buttons: `Create security group` (selected) and `Select existing security group`.
- Security group name - required:** A text input field contains `launch-wizard-9`. A note below states: "This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and .-:/@#,%&()\*~!\*'".

**Right Column (Summary):**

- Summary:** A section with a "Number of instances" input field set to `1`.
- Software Image (AMI):** A dropdown menu shows `Amazon Linux 2023 AMI 2023.6.2...read more` with ID `ami-085ad6ae776d8f09c`.
- Virtual server type (instance type):** A dropdown menu shows `t2.micro`.
- Firewall (security group):** A dropdown menu shows `New security group`.
- Storage (volumes):** A section with a "Cancel" button and a "Launch instance" button.
- Preview code:** A link to "Preview code" is located below the "Launch instance" button.

The footer of the console shows "CloudShell", "Feedback", and copyright information: "© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

## 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.

