



Placement Empowerment Program

Cloud Computing and DevOps Centre

Set Up a Private Network in the Cloud : Create a Virtual Private Cloud (VPC) with subnets for your instances. Configure routing for internal communication between subnets.

Name: Akshaya S

Department:CSE

Introduction

The goal of this Proof of Concept (PoC) was to set up a **Private Network in the Cloud** by creating a **Virtual Private Cloud (VPC)** in AWS, configuring **subnets**, and ensuring **internal communication** between instances within the VPC. This setup focused on isolating cloud resources in a private network, providing a secure environment for communication, and making sure that only internal traffic is allowed, without exposing resources to the public internet.

In this PoC, we created a **private subnet** where EC2 instances could communicate with each other without direct exposure to external networks.

Overview

In this PoC, we:

1. **Created a VPC** in AWS, which serves as the isolated private network.
2. **Created a private subnet** inside the VPC where EC2 instances can reside, ensuring no direct access from the public internet.
3. **Set up routing** to allow communication between the instances within the same VPC and subnet.

4. Launched **EC2 instances** in the private subnet and verified their ability to communicate internally using their private IP addresses.

The setup is designed to simulate a secure cloud environment where resources can interact securely without being exposed to external traffic.

Objective

The primary objectives of this PoC were:

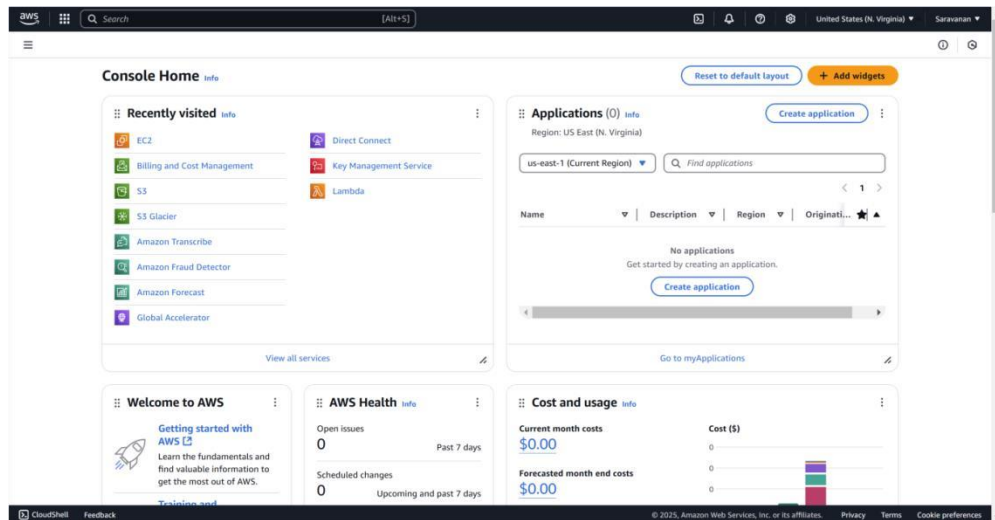
1. **Establish a Private Network:** Set up a private VPC and subnets for cloud resources to reside in, ensuring they are isolated from the public internet.
2. **Internal Communication:** Ensure that EC2 instances within the private subnet can communicate with each other using their private IPs.
3. **Security:** Maintain internal communication only within the VPC, preventing direct exposure of instances to the public internet.
4. **Simplify Management:** Organize cloud resources into subnets for easier management and scaling, with clear routing between them.

Step-by-Step Overview

Step 1:

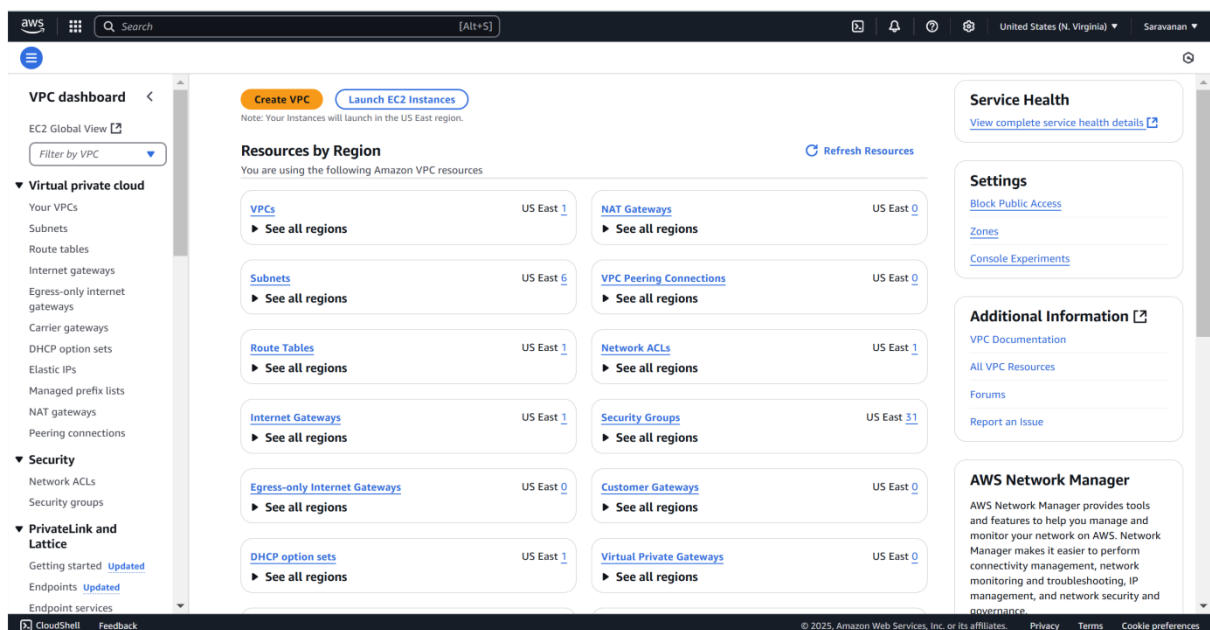
1. Go to [AWS Management Console](#).

2. Enter your username and password to log in.



Step 2:

In the **VPC Dashboard**, click the **Create VPC** button.



Step 3:

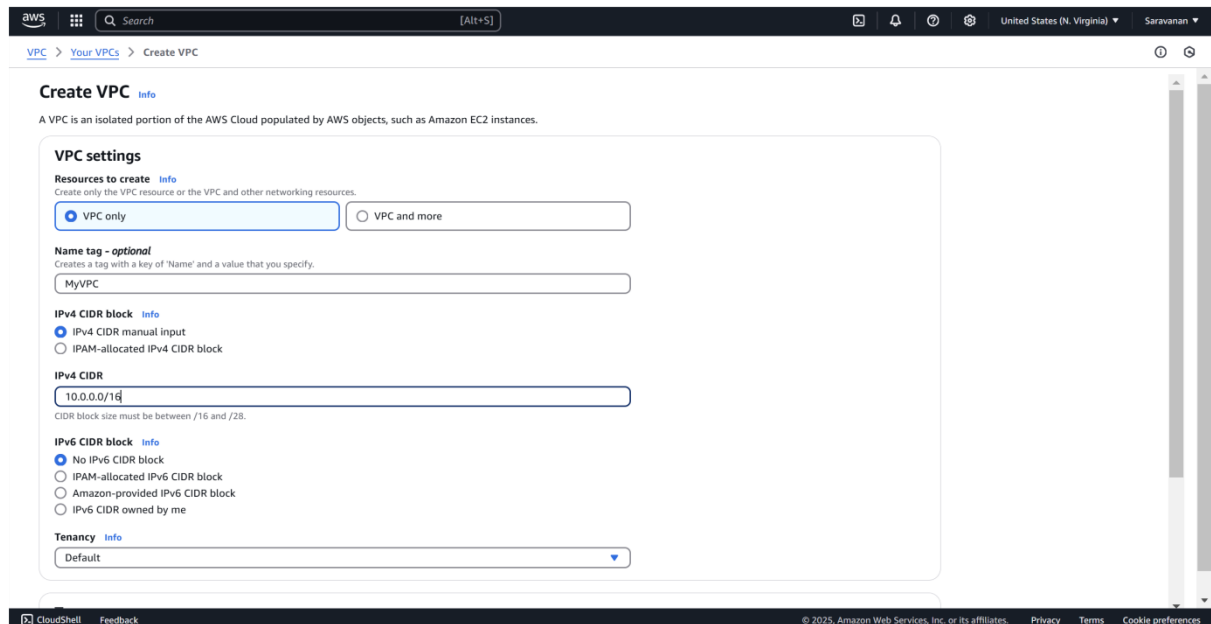
In the VPC creation wizard, select **VPC only**.

Name tag: Enter MyVPC .

IPv4 CIDR block: Enter 10.0.0.0/16 (this defines the IP range for your VPC).

Tenancy: Leave it as **Default**.

Click **Create VPC**.

The screenshot shows the AWS 'Create VPC' wizard. The 'Resources to create' section has 'VPC only' selected. The 'Name tag - optional' field contains 'MyVPC'. The 'IPv4 CIDR block' section has 'IPv4 CIDR manual input' selected, with the 'IPv4 CIDR' field containing '10.0.0.0/16'. The 'IPv6 CIDR block' section has 'No IPv6 CIDR block' selected. The 'Tenancy' dropdown is set to 'Default'. The interface includes a top navigation bar with the AWS logo, a search bar, and a right-hand menu with icons for home, notifications, settings, and a user profile. The bottom of the screen shows a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services.

Step 4:

In the **VPC Dashboard**, click on **Subnets** in the left-hand menu.

Click the **Create subnet** button.

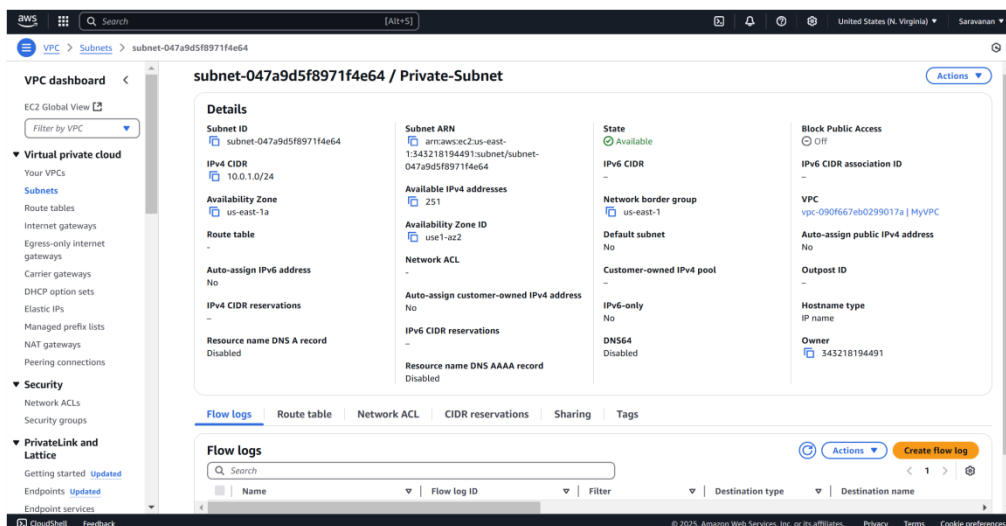
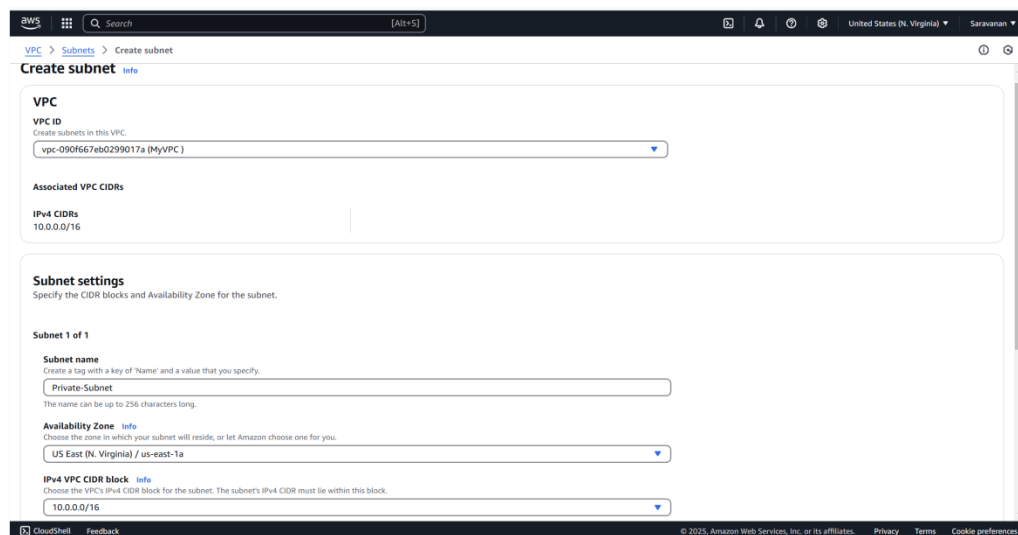
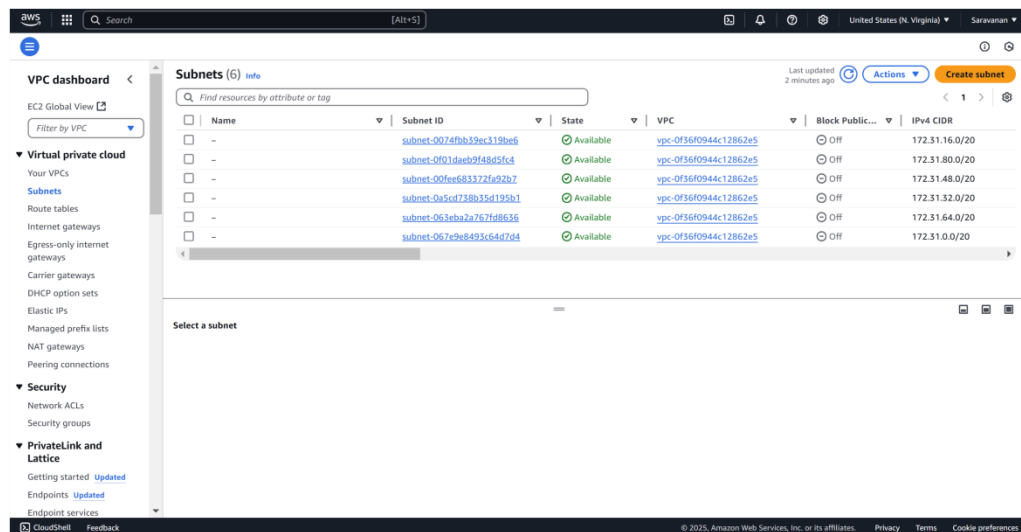
VPC: Select MyVPC (the one you just created).

Subnet name: Enter Private-Subnet.

Availability Zone: Pick any (e.g., us-east-1a or any zone from your region).

IPv4 CIDR block: Enter 10.0.1.0/24 (this is a smaller range within the VPC's IP range).

Click Create subnet.



In the **VPC Dashboard**, click on **Route Tables** in the left-hand menu. Click **Create route table**.

Step 5:

Name tag: Enter InternalRouteTable.

VPC: Select MyVPC (the one you created earlier).

Click **Create route table**.

The screenshot shows the 'Create route table' page in the AWS Management Console. The page has a dark header with the AWS logo, a search bar, and navigation links. The main content area is titled 'Create route table' with a sub-header 'Route table settings'. Under 'Name - optional', there is a text input field containing 'InternalRouteTable'. Under 'VPC', there is a dropdown menu showing 'vpc-090f667eb0299017a (MyVPC)'. Below this, the 'Tags' section shows a table with one tag: Key 'Name' and Value 'InternalRouteTable'. There are buttons for 'Add new tag' and 'Remove'. At the bottom right, there are 'Cancel' and 'Create route table' buttons.

The screenshot shows the 'Subnet associations' page for the route table 'rtb-0704f15461ee91808 / InternalRouteTable'. The page has a dark header with the AWS logo, a search bar, and navigation links. The main content area is titled 'rtb-0704f15461ee91808 / InternalRouteTable' with a sub-header 'Subnet associations'. The 'Details' section shows the route table ID, VPC, and owner ID. The 'Subnet associations' section shows a table with one row: Name 'Private-Subnet', Subnet ID 'subnet-047a9d5f8971f4e64', IPv4 CIDR '10.0.1.0/24', and IPv6 CIDR '-'. There are buttons for 'Edit subnet associations' and 'Find subnet association'.

Step 6:

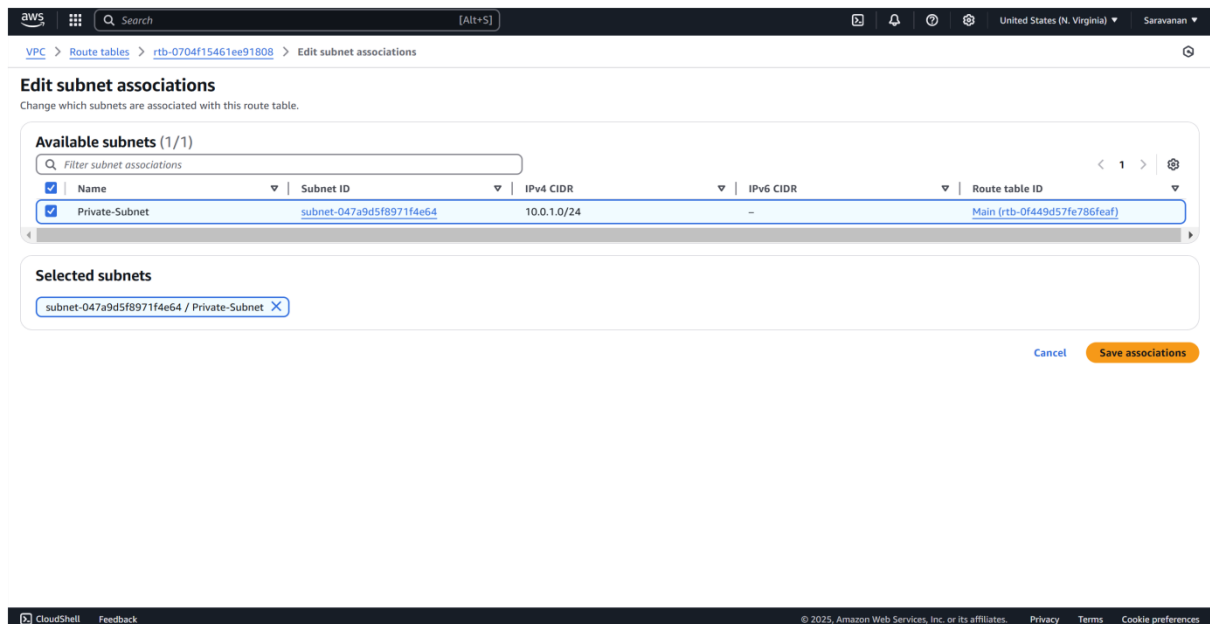
Select the InternalRouteTable you just created.

Go to the **Subnet Associations** tab (it's near the bottom).

Click **Edit subnet associations**.

Select Private-Subnet (the subnet you created earlier).

Click **Save associations**.



To launch a new EC2 instance in your private subnet, go to the EC2 Dashboard, click **Launch Instance**, and fill in the details: Name it "Private-Instance", choose an Amazon Linux 2 AMI (or another freetier eligible image), select the **t2.micro** instance type, and either choose an existing key pair or create a new one for SSH access. Under **Network settings**, select your **MyVPC** and **Private-Subnet**, and make sure

Step 7:

Auto-assign Public IP is disabled to keep it private. Leave all other settings as default, then click **Launch Instance**.

Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-0f56f0944c12862e5

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

Auto-assign public IP [Info](#)
Disable
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

We'll create a new security group called 'launch-wizard-29' with the following rules:

☒ Allow SSH traffic from [Info](#)
Helps you connect to your instance. Anywhere 0.0.0.0/0

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server.

☐ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server.

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023.6.2...[read more](#)
ami-085a6ae776d8f09c

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and

[Cancel](#) [Launch instance](#) [Preview code](#)

Instance type [>](#)

Select an instance type that meets your computing, memory, networking, or storage needs.

Pricing
Prices shown are for instances running common operating systems with no pre-installed software. Prices for instances running other operating systems are available on the [Amazon EC2 On-Demand Pricing](#) page. You can calculate your estimated costs using the [AWS Pricing Calculator](#).

Learn more [>](#)
[Amazon EC2 instance types](#)

Network settings [Info](#)

VPC [Info](#)
vpc-090f667eb0299017a (MyVPC) 10.0.0.0/16

Subnet [Info](#)
subnet-047a9d5f8971f4e64 Private-Subnet
VPC: vpc-090f667eb0299017a Owner: 34321819481
Availability Zone: us-east-1a Zone type: Availability Zone
IP addresses available: 251 CIDR: 10.0.1.0/24 [Create new subnet](#)

Auto-assign public IP [Info](#)
Disable

Firewall (security groups) [Info](#)
A 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 _/[]@.()+=&|!~*~*~*

☒ Create security group ☐ Select existing security group

Security group name - required
launch-wizard-29

Description - required [Info](#)
launch-wizard-29 created 2025-02-08T16:18:43.781Z

Inbound Security Group Rules
Security group rule 1 (TCP: 22, 0.0.0.0/0) [Remove](#)

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023.6.2...[read more](#)
ami-085a6ae776d8f09c

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and

[Cancel](#) [Launch instance](#) [Preview code](#)

Instance type [>](#)

Select an instance type that meets your computing, memory, networking, or storage needs.

Pricing
Prices shown are for instances running common operating systems with no pre-installed software. Prices for instances running other operating systems are available on the [Amazon EC2 On-Demand Pricing](#) page. You can calculate your estimated costs using the [AWS Pricing Calculator](#).

Learn more [>](#)
[Amazon EC2 instance types](#)

Step 8: Verify Internal Communication

1. Find the private IP of your instance:

Go to the **EC2 Dashboard**.

Select your instance in Private-Subnet.

Note the **Private IPv4 address** (e.g., 10.0.1.x).

2. Ping the Private IP:

If you have only one instance, you can skip this. If you have multiple instances in the private subnet, SSH into one instance and try pinging the private IP of the other instance.

