

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: Shaon jenifer 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.

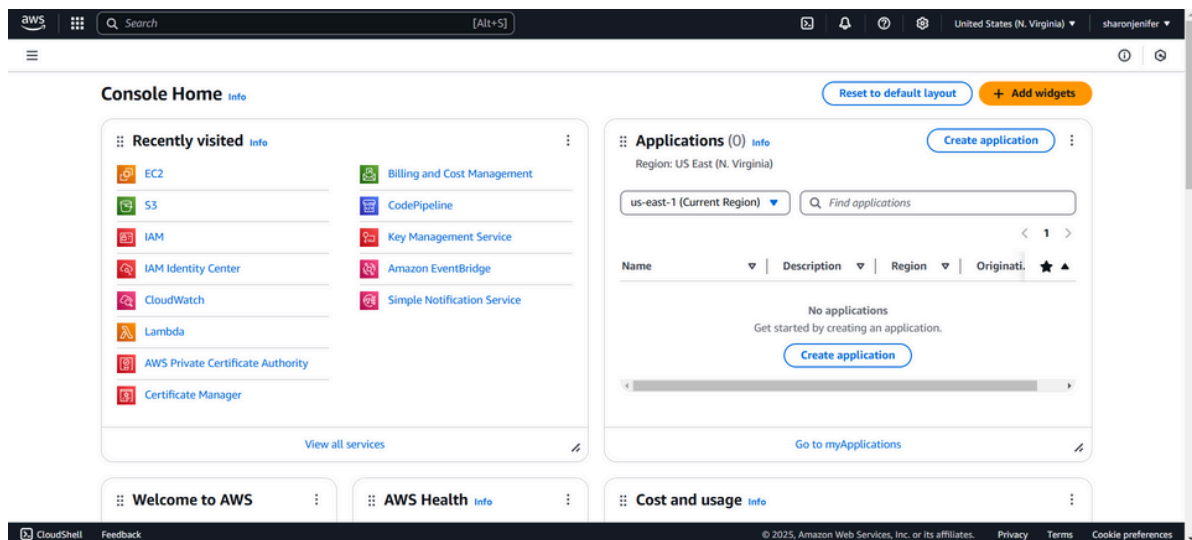
Importance

1. **Security:** By placing EC2 instances in a private subnet and ensuring that no public IP is assigned, the resources are isolated from external traffic. This is crucial for keeping sensitive data and services protected.
2. **Cost Efficiency:** Using internal communication and private subnets can help reduce costs related to public internet access and data transfer.
3. **Flexibility:** This setup provides a foundation for building more complex cloud infrastructures, such as multi-tier applications where only backend servers (databases, app servers) are private, while frontend servers may be public.

Step-by-Step Overview

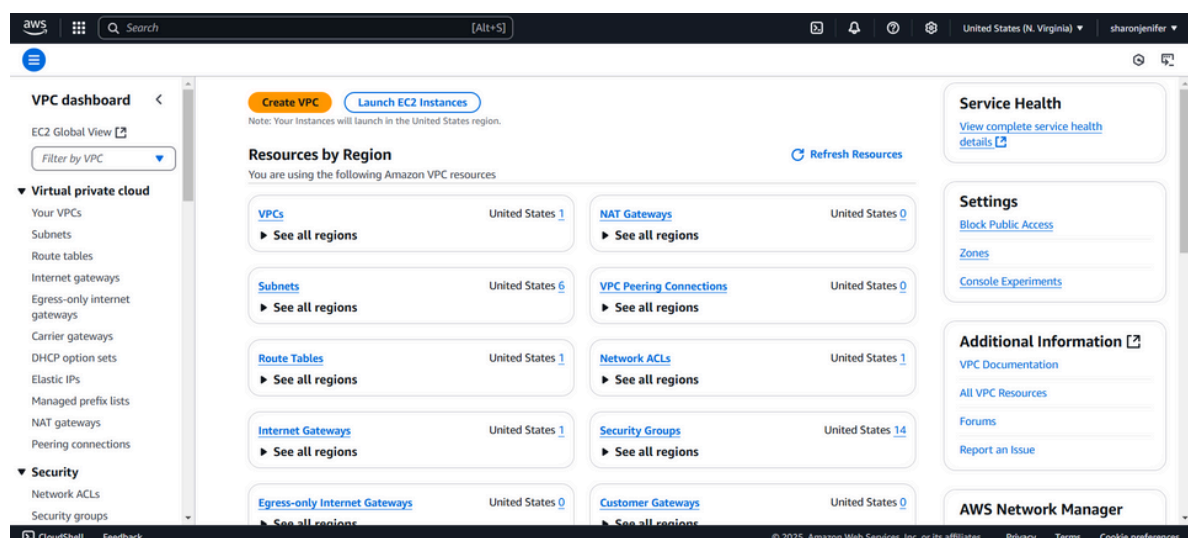
Step 1:

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:

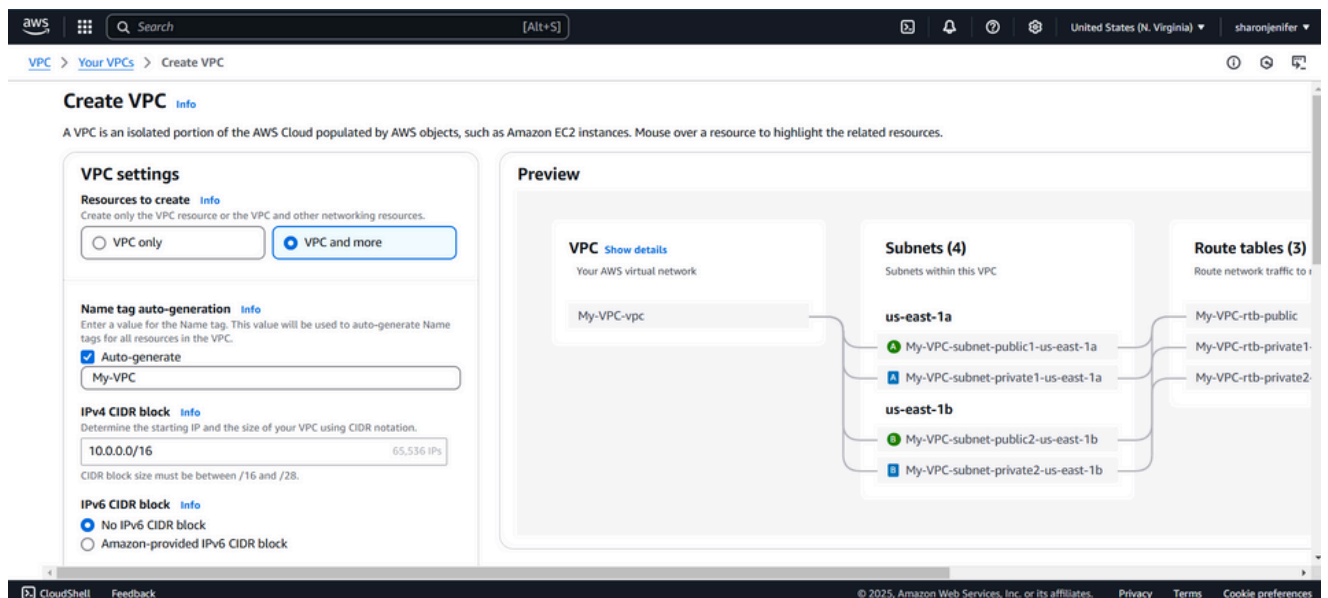
In the VPC creation wizard, select VPC only.

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.



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.

The screenshot shows the AWS VPC dashboard. A green notification banner at the top states: "You have successfully created 1 subnet: subnet-0b116329809d80403". Below this, the "Subnets (1/1)" section displays a table with one entry:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
Private-Subnet	subnet-0b116329809d80403	Available	vpc-01661d581b58879a5 My...	Off	10.0.32.0/20

Below the table, the "Details" section for "subnet-0b116329809d80403 / Private-Subnet" is shown. It includes fields for Subnet ID, Subnet ARN, State (Available), IPv4 CIDR (10.0.32.0/20), Availability Zone (us-east-1a), Available IPv4 addresses (4091), Availability Zone ID (us-east-1a), Network border group (us-east-1), and Block Public Access (Off). The VPC is identified as "vpc-01661d581b58879a5 | My-VPC-vpc".

The screenshot shows the "Create route table" wizard. The "Route table settings" section includes a "Name - optional" field with the value "InternalRouteTable" and a "VPC" dropdown menu set to "vpc-01661d581b58879a5 (My-VPC-vpc)". The "Tags" section shows a key-value pair: "Name" with the value "InternalRouteTable". At the bottom right, there are "Cancel" and "Create route table" buttons.

The screenshot shows the AWS VPC dashboard with the "Route table rtb-07bddf6b2a8ce8ca9 / InternalRouteTable" selected. A green notification banner at the top states: "Route table rtb-07bddf6b2a8ce8ca9 | InternalRouteTable was created successfully." Below this, the "Details" section shows the Route table ID (rtb-07bddf6b2a8ce8ca9), VPC (vpc-01661d581b58879a5 | My-VPC-vpc), Main (No), Owner ID (867344462005), and Explicit subnet associations (None). The "Routes (1)" section shows a table with one entry:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

Step 5:

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

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 title is 'Create route table' with an 'info' icon. Below the title 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 'Route table settings' section contains two fields: 'Name - optional' with the value 'InternalRouteTable' and 'VPC' with a dropdown menu showing 'vpc-090f667eb0299017a (MyVPC)'. The 'Tags' section shows a table with one tag: 'Name' as the key and 'InternalRouteTable' as the value. At the bottom right, there are 'Cancel' and 'Create route table' buttons.

Create route table info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

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

InternalRouteTable

VPC
The VPC to use for this route table.

vpc-090f667eb0299017a (MyVPC)

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

Name

Value - optional

InternalRouteTable

Remove

Add new tag

You can add 49 more tags.

Cancel Create route table

The screenshot shows the 'Route table details' page for 'rtb-0704f15461ee91808 / InternalRouteTable'. A green banner at the top states 'Route table rtb-0704f15461ee91808 / InternalRouteTable was created successfully.' The page has a left-hand navigation menu with categories like 'Virtual private cloud', 'Security', and 'PrivateLink and Lattice'. The main content area shows details for the route table, including its ID, VPC, and owner ID. Below this are tabs for 'Routes', 'Subnet associations', 'Edge associations', 'Route propagation', and 'Tags'. The 'Subnet associations' tab is active, showing 'Explicit subnet associations (0)' and 'Subnets without explicit associations (1)'. The 'Subnets without explicit associations' section lists one subnet: 'Private-Subnet' with ID 'subnet-047a9d5f8971f4e64' and CIDR '10.0.1.0/24'.

Route table rtb-0704f15461ee91808 / InternalRouteTable Actions

Details info

Route table ID
rtb-0704f15461ee91808

VPC
vpc-090f667eb0299017a | MyVPC

Main
No

Owner ID
343218194491

Explicit subnet associations
-

Edge associations
-

Subnet associations

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.			

Subnets without explicit associations (1) Edit subnet associations

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

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Private-Subnet	subnet-047a9d5f8971f4e64	10.0.1.0/24	-

Step 6:

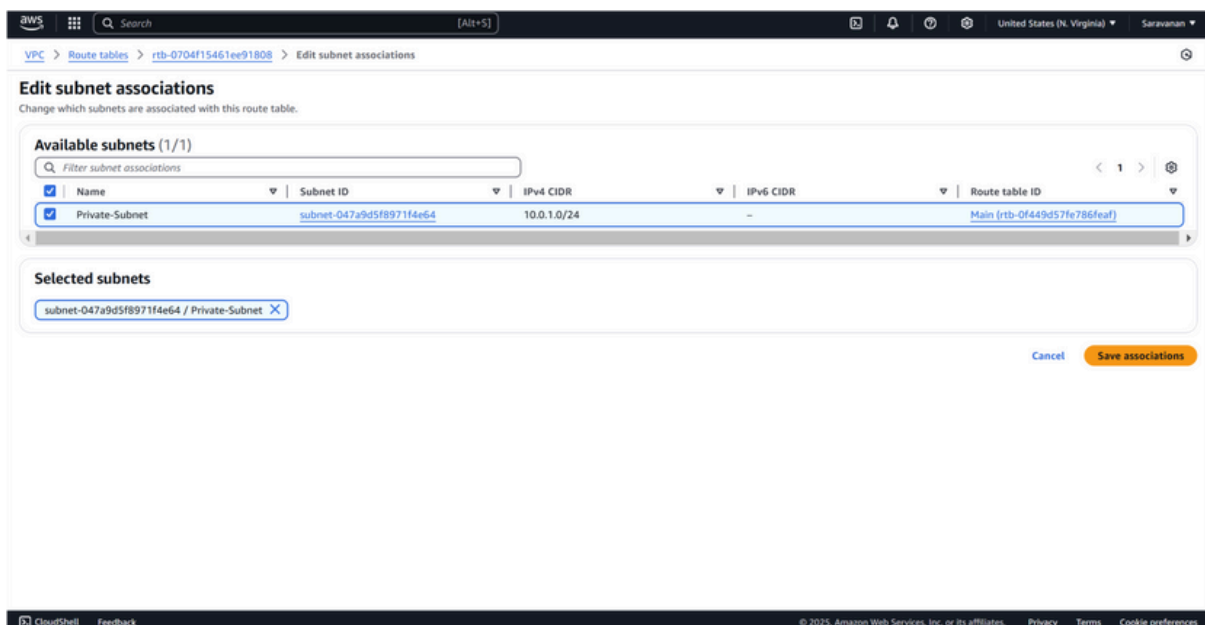
Select the InternalRouteTable you just created.

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.



Step 7:

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

EC2 > Instances > Launch an instance

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

Network [info](#)
vpc-0f36f0944c12862e5

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
Helps you connect to your instance

☐ 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 AMI 2023.6.2...[read more](#)
ami-085ad8ae775d8909c

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 [info](#)

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

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

EC2 > Instances > Launch an instance

Network settings [info](#)

VPC - required [info](#)
vpc-01661d581b58879a5 (My-VPC-vpc) ☒

Subnet [info](#)
subnet-0b116329809d80403 Private-Subnet ☒ [Create new subnet](#)

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

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 [Compare security group rules](#)

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

Advanced network configuration

Summary

Number of instances [info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...[read more](#)
ami-05b10e08d247b9d27

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 of opening an AWS account, you get 750 hours per month of

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

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

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.

Outcome

By completing this PoC of setting up a Private Network in AWS, you will:

1. Deploy a VPC with a private subnet to isolate cloud resources securely from the public internet.
2. Launch EC2 instances within the private subnet and ensure internal communication between them using private IPs.
3. Configure routing tables to enable efficient communication within the VPC while maintaining the isolation of private resources.
4. Implement security groups to allow only internal traffic between instances while restricting external access.
5. Gain practical experience in designing secure cloud architectures and foundational AWS services like VPC, EC2, and private networking.

