



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: Gopinath M

Department: ADS

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.

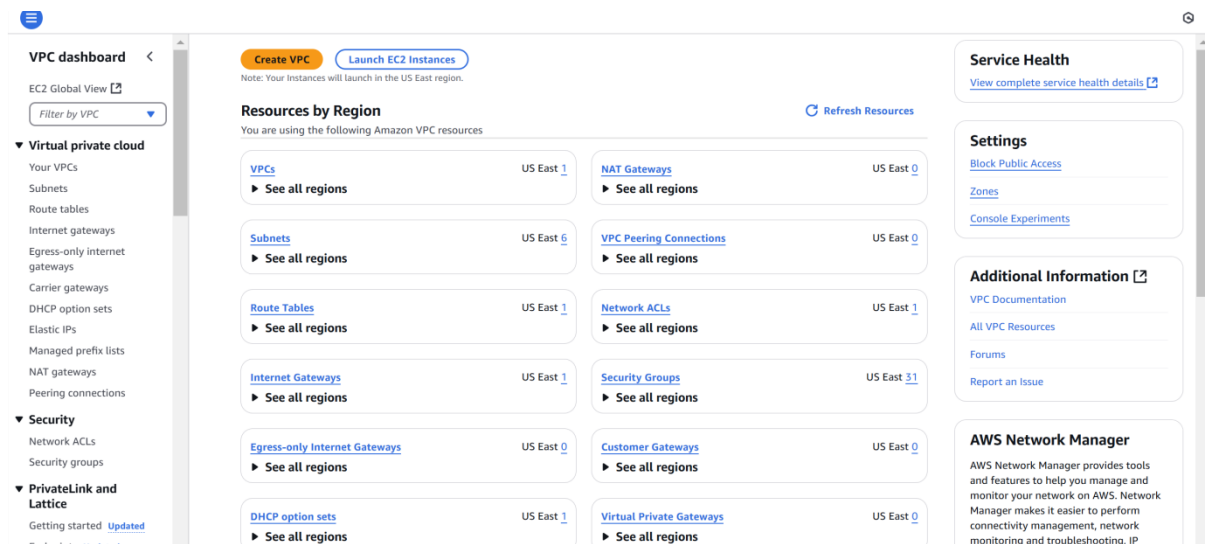
Step-by-Step Overview Step

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.

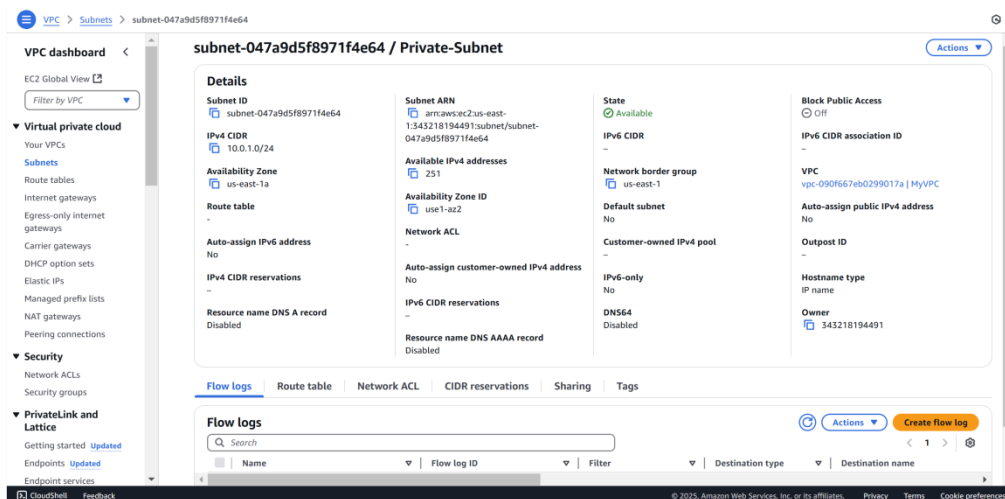
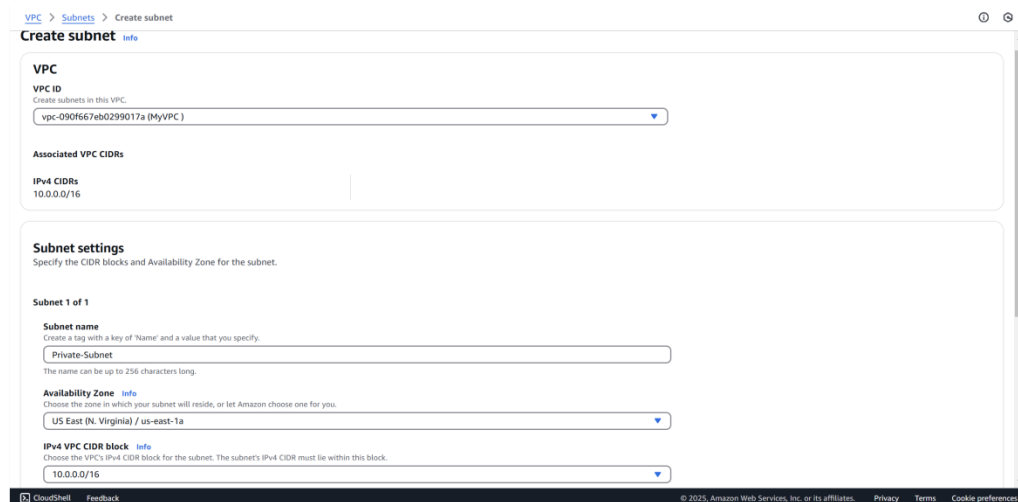
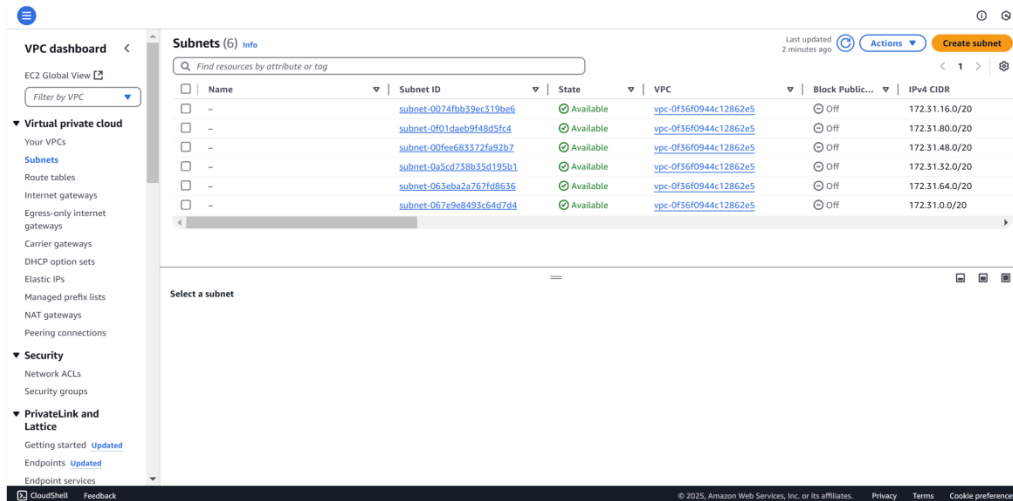
VPC: Select MyVPC (the one you just created).

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**.



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

Name tag: Enter InternalRouteTable.

Step 5:

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

Click **Create route table**.

The image shows two screenshots from the AWS Management Console. The top screenshot is the 'Create route table' wizard. It has a breadcrumb trail: VPC > Route tables > Create route table. The main heading is 'Create route table' with an 'Info' link. Below it is a note: '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 includes a 'Name - optional' field with the value 'InternalRouteTable' and a 'VPC' dropdown menu showing 'vpc-090f667eb0299017a (MyVPC)'. The 'Tags' section has a 'Key' field with 'Name' and a 'Value - optional' field with 'InternalRouteTable'. There are 'Add new tag' and 'Remove' buttons. At the bottom right are 'Cancel' and 'Create route table' buttons.

The bottom screenshot shows the details page for the route table 'rtb-0704f15461ee91808 / InternalRouteTable'. It has a breadcrumb trail: VPC > Route tables > rtb-0704f15461ee91808. A green success message at the top says 'Route table rtb-0704f15461ee91808 | InternalRouteTable was created successfully.' The page has a left sidebar with navigation links: VPC dashboard, EC2 Global View, Filter by VPC, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections), Security (Network ACLs, Security groups), and PrivateLink and Lattice (Getting started, Endpoints). The main content area has tabs: Routes, Subnet associations (selected), Edge associations, Route propagation, and Tags. The 'Subnet associations' tab shows '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'.

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**.

VPC > Route tables > rtb-0704f15461ee91808 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/1)

Filter subnet associations

<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	Private-Subnet	subnet-047a9d5f8971f4e64	10.0.1.0/24	-	Main (rtb-0f449d57fe786feaf)

Selected subnets

subnet-047a9d5f8971f4e64 / Private-Subnet

Cancel 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 **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

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](#)
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

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. 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 [Info](#)

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...[read more](#)
ami-085ad8ae776d8f09c

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 GB 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](#)

CloudWatch Feedback

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

Step 7:

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console, specifically the 'Network settings' step. The 'VPC' is set to 'vpc-090f667eb0299017a (MyVPC)' with a CIDR of '10.0.0.0/16'. The 'Subnet' is 'subnet-047a9d5f8971f4e64' (Private-Subnet) with a CIDR of '10.0.1.0/24'. The 'Auto-assign public IP' is set to 'Disable'. A new security group is being created with the name 'launch-wizard-29' and description 'launch-wizard-29 created 2025-02-08T16:18:43.781Z'. The summary on the right shows 1 instance, Amazon Linux 2023.6.2 AMI, t2.micro instance type, new security group, and 1 volume (8 GiB). A 'Free tier' notification is visible. The bottom of the console shows the 'cloudshell' tab and footer information.

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.

