

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: SAMRAJ K

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.

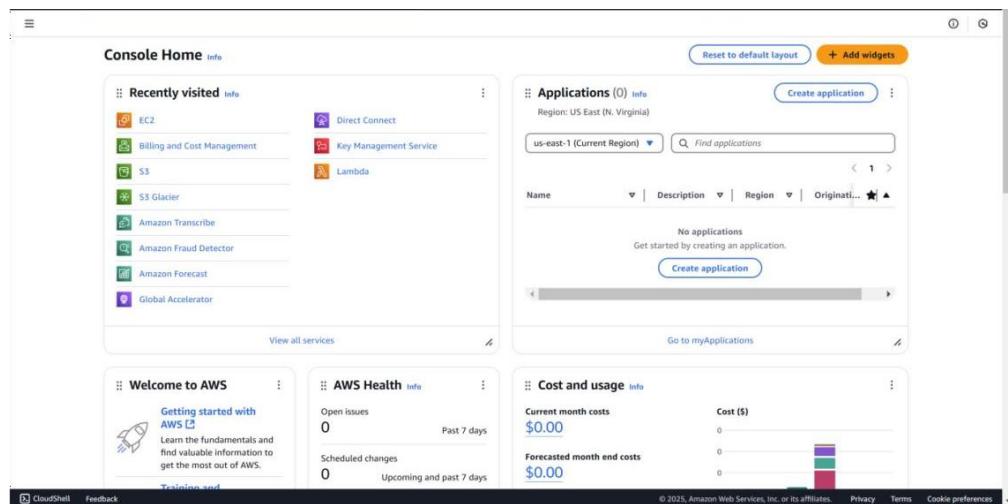
Importance

- 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

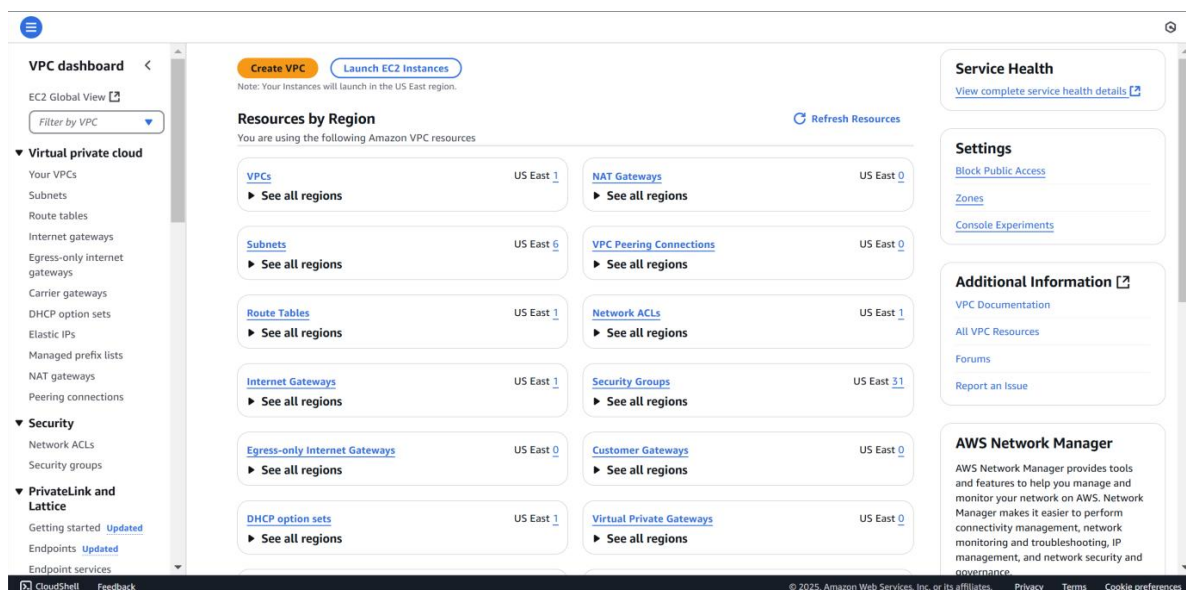
Step 1:

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:

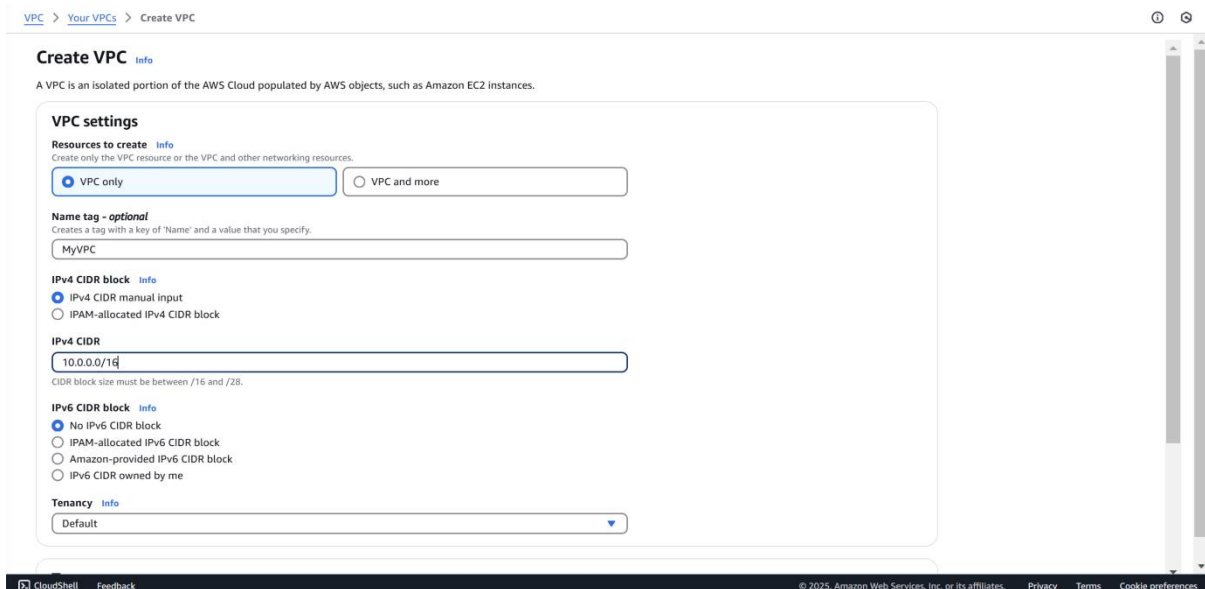
In the VPC creation wizard, select **VPC only**.

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**.

The screenshot shows the 'Create VPC' wizard in the AWS Management Console. The breadcrumb navigation at the top reads 'VPC > Your VPCs > Create VPC'. The main heading is 'Create VPC' with an 'Info' link. Below this is a descriptive sentence: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.' The 'VPC settings' section contains several fields: 'Resources to create' with radio buttons for 'VPC only' (selected) and 'VPC and more'; 'Name tag - optional' with a text input field containing 'MyVPC'; 'IPv4 CIDR block' with radio buttons for 'IPv4 CIDR manual input' (selected) and 'IPAM-allocated IPv4 CIDR block'; 'IPv4 CIDR' with a text input field containing '10.0.0.0/16' and a note 'CIDR block size must be between /16 and /28'; 'IPv6 CIDR block' with radio buttons for 'No IPv6 CIDR block' (selected), 'IPAM-allocated IPv6 CIDR block', 'Amazon-provided IPv6 CIDR block', and 'IPv6 CIDR owned by me'; and 'Tenancy' with a dropdown menu set to 'Default'. 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 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 console's 'Subnets' page. On the left, a navigation menu includes 'Virtual private cloud', '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', 'PrivateLink and Lattice', 'Getting started', 'Endpoints', and 'Endpoint services'. The main content area is titled 'Subnets (6)' and includes a search bar and a table of subnets. The table has columns for Name, Subnet ID, State, VPC, Block Public..., and IPv4 CIDR. All subnets are in an 'Available' state and are associated with the VPC 'vpc-0f36f0944c12862e5'. Below the table, there is a 'Select a subnet' section.

| Name | Subnet ID | State | VPC | Block Public... | IPv4 CIDR |
|------|--------------------------|-----------|-----------------------|-----------------|----------------|
| - | subnet-0074fbb39ec319be6 | Available | vpc-0f36f0944c12862e5 | Off | 172.31.16.0/20 |
| - | subnet-0f01daeb9f48d5fcd | Available | vpc-0f36f0944c12862e5 | Off | 172.31.80.0/20 |
| - | subnet-00fec68337f1a92b7 | Available | vpc-0f36f0944c12862e5 | Off | 172.31.48.0/20 |
| - | subnet-0a5cd738b35d195b1 | Available | vpc-0f36f0944c12862e5 | Off | 172.31.32.0/20 |
| - | subnet-063eba2a767f88636 | Available | vpc-0f36f0944c12862e5 | Off | 172.31.64.0/20 |
| - | subnet-067e9e8493c64d7d4 | Available | vpc-0f36f0944c12862e5 | Off | 172.31.0.0/20 |

The screenshot shows the 'Create subnet' page in the AWS VPC console. It includes a 'VPC' section with a dropdown menu showing 'vpc-090f667eb0299017a (MyVPC)'. Below this is the 'Associated VPC CIDRs' section, showing 'IPv4 CIDRs' as '10.0.0.0/16'. The 'Subnet settings' section is titled 'Subnet 1 of 1' and contains three main fields: 'Subnet name' (with a placeholder 'Private-Subnet'), 'Availability Zone' (set to 'us-east-1a'), and 'IPv4 VPC CIDR block' (set to '10.0.0.0/16').

The screenshot shows the details page for the subnet 'subnet-047a9d5f8971f4e64 / Private-Subnet'. The page is divided into several sections: 'Details' (Subnet ID, IPv4 CIDR, Availability Zone, Route table, Auto-assign IPv6 address, IPv4 CIDR reservations, Resource name DNS A record), 'Subnet ARN', 'Available IPv4 addresses', 'Availability Zone ID', 'Network ACL', 'Auto-assign customer-owned IPv4 address', 'IPv6 CIDR reservations', 'Resource name DNS AAAA record', 'State' (Available), 'IPv6 CIDR', 'Network border group', 'Default subnet', 'Customer-owned IPv4 pool', 'IPv6-only', 'DNS64', 'Block Public Access' (Off), 'IPv6 CIDR association ID', 'VPC', 'Auto-assign public IPv4 address', 'Outpost ID', 'Hostname type', and 'Owner'. Below these sections is a 'Flow logs' section with a search bar and a table of flow logs.

| Name | Flow log ID | Filter | Destination type | Destination name |
|------|-------------|--------|------------------|------------------|
|------|-------------|--------|------------------|------------------|

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 VPC 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 sentence 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 'InternalRouteTable' and 'VPC' with a dropdown menu showing 'vpc-090f667eb0299017a (MyVPC)'. Below this is the 'Tags' section, which includes a 'Key' field with 'Name' and a 'Value - optional' field with 'InternalRouteTable'. There are 'Add new tag' and 'Remove' buttons. At the bottom right of the form are 'Cancel' and 'Create route table' buttons.

The screenshot shows the details page for the route table 'rtb-0704f15461ee91808 / InternalRouteTable'. A green success banner at the top states: 'Route table rtb-0704f15461ee91808 | InternalRouteTable was created successfully.' The left-hand navigation menu is expanded, showing 'Virtual private cloud' with sub-items like 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', etc. The main content area has 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 IP address '10.0.1.0/24'. At the bottom, there is a footer with '© 2025, Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

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 X

Cancel Save associations

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

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
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.

☐ 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-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 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](#)

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

EC2 > Instances > Launch an instance

Network settings Info

VPC - required Info
vpc-090f667eb0299017a (MyVPC)

Subnet Info
subnet-047a9d5f897114e64 (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

Security group name - required
launch-wizard-29

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 _/!@,[@!]+&.,[]*

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](#)

Type Info **Protocol** Info **Port range** Info

Summary

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 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](#)

© 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.

