

## **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: Shana R 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.

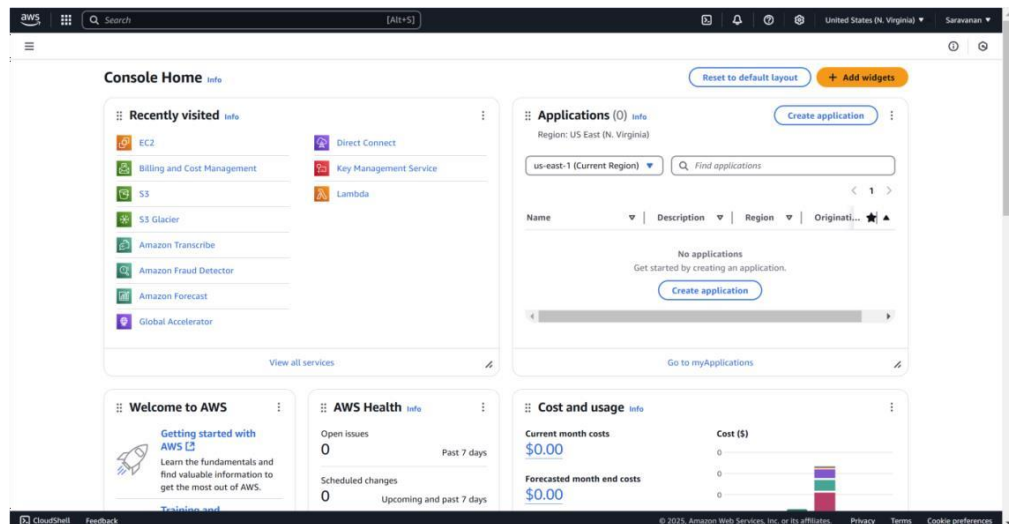
# Importance

- 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

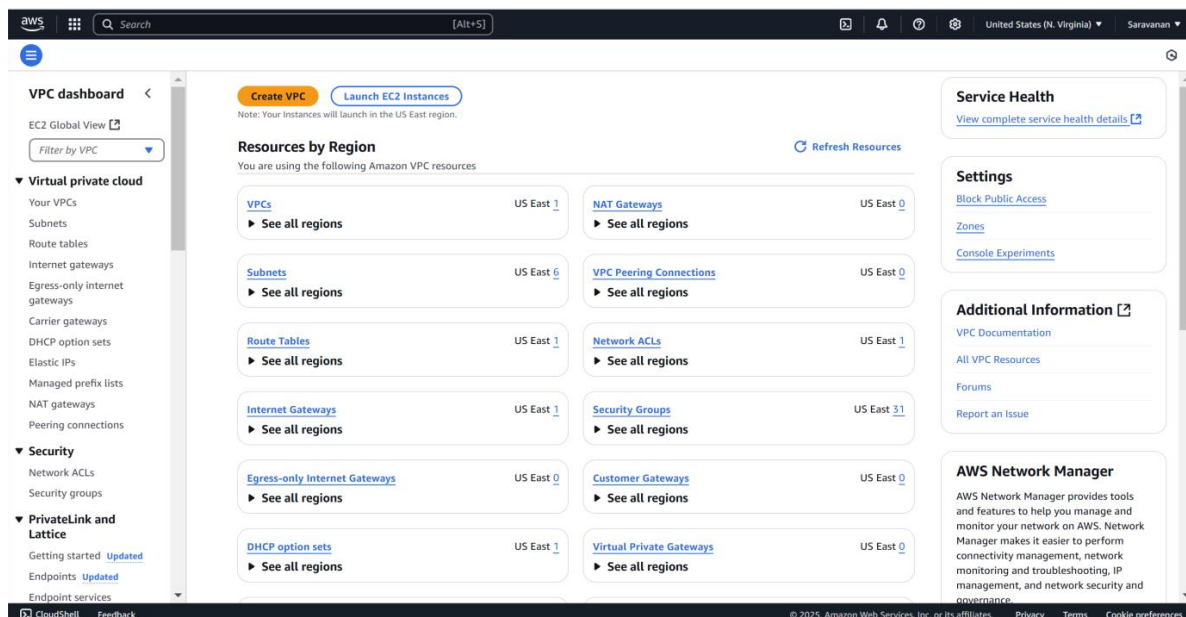
## Step 1:

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:

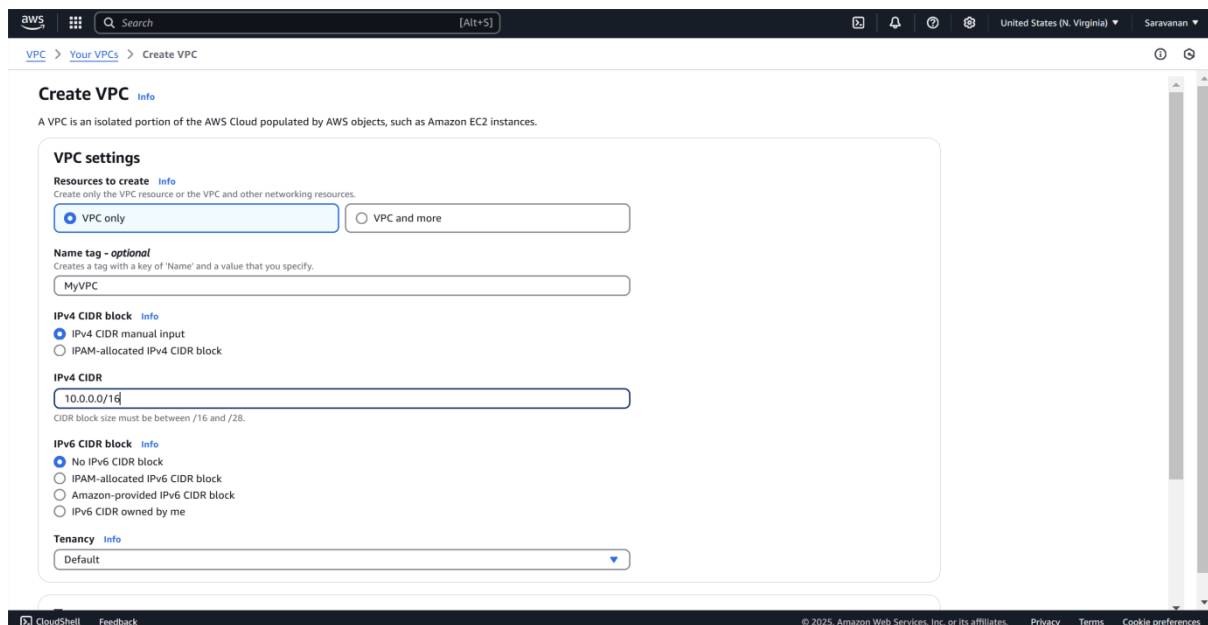
In the VPC creation wizard, select **VPC only**.

**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**.

The screenshot shows the 'Create VPC' wizard in the AWS Management Console. The page title is 'Create VPC' with an 'Info' link. Below the title, a note states: '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 bottom of the page shows the 'CloudShell' and 'Feedback' links, and the copyright notice '© 2025, Amazon Web Services, Inc. or its affiliates.' along with 'Privacy', 'Terms', and 'Cookie preferences' links.

## Step 4:

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

Click the **Create subnet** button.

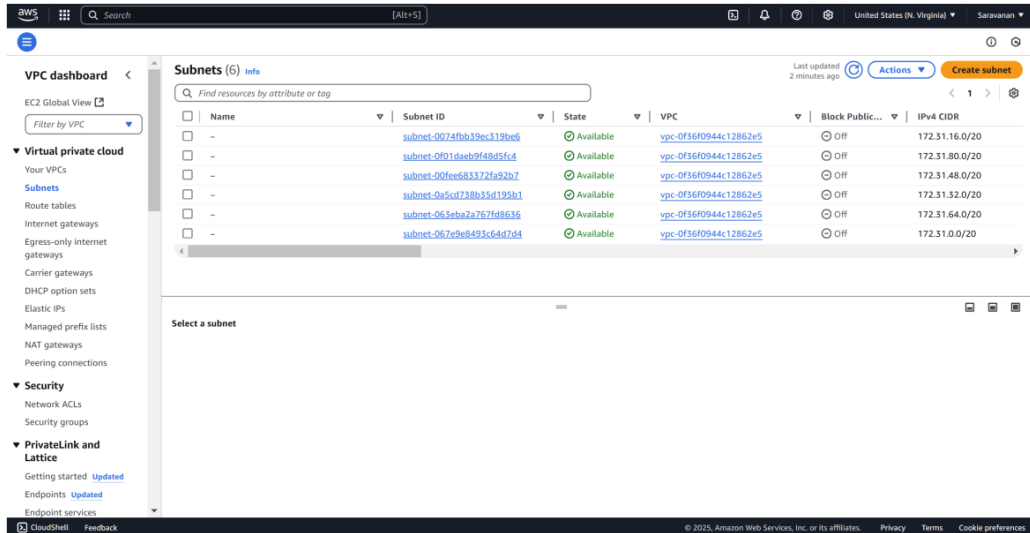
**VPC:** Select MyVPC (the one you just created).

**Subnet name:** Enter Private-Subnet.

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

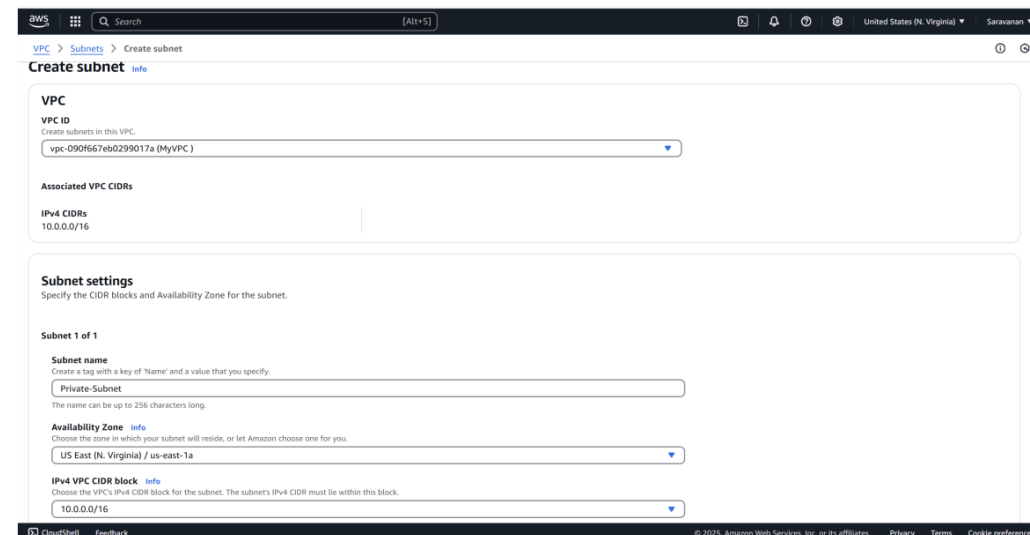
**IPv4 CIDR block:** Enter 10.0.1.0/24 (this is a smaller range within the VPC's IP range).

Click **Create subnet**.

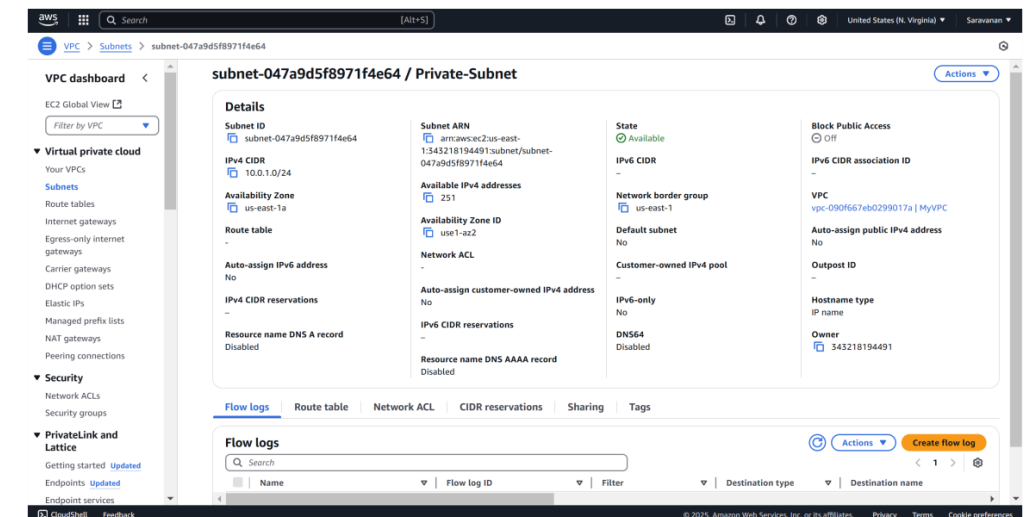


The screenshot shows the AWS VPC console's 'Subnets' page. On the left is a navigation menu with options like 'Virtual private cloud', 'Security', and 'PrivateLink and Lattice'. The main area displays a table of subnets. The table has columns for Name, Subnet ID, State, VPC, Block Public..., and IPv4 CIDR. All subnets listed are in an 'Available' state and are associated with the VPC 'vpc-0f36f0944c12862e5'. The IPv4 CIDR blocks range from 172.31.16.0/20 to 172.31.0.0/20.

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-0074fb39ec319be6	Available	vpc-0f36f0944c12862e5	Off	172.31.16.0/20
-	subnet-0f01dae99f48d5fcd	Available	vpc-0f36f0944c12862e5	Off	172.31.80.0/20
-	subnet-00fee683372fa92b7	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' wizard in the AWS VPC console. It is divided into two main sections: 'VPC' and 'Subnet settings'. In the 'VPC' section, the 'VPC ID' is set to 'vpc-090f667eb0299017a (MyVPC)'. In the 'Subnet settings' section, the 'Subnet name' is 'Private-Subnet', the 'Availability Zone' is 'US East (N. Virginia) / us-east-1a', and the 'IPv4 VPC CIDR block' is '10.0.0.0/16'.



The screenshot shows the 'Details' page for the subnet 'subnet-047a9d5f8971f4e64 / Private-Subnet'. The page is divided into two main sections: 'Details' and 'Flow logs'. The 'Details' section contains information about the subnet's configuration, including its ID, ARN, CIDR block, availability zone, and various settings like 'Block Public Access' and 'Auto-assign public IPv4 address'. The 'Flow logs' section shows a table of flow logs, with one log entry visible.

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 Management Console. The breadcrumb navigation is '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.'

**Route table settings**

- Name - optional:** A text input field containing 'InternalRouteTable'. Below it, a note says 'Create a tag with a key of 'Name' and a value that you specify.'
- VPC:** A dropdown menu showing '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**

Q Name X

**Value - optional**

Q InternalRouteTable X 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 success message at the top states: 'Route table rtb-0704f15461ee91808 | InternalRouteTable was created successfully.'

**Details**

- Route table ID:** rtb-0704f15461ee91808
- VPC:** vpc-090f667eb0299017a | MyVPC
- Main:** No
- Owner ID:** 343218194491
- Explicit subnet associations:** -
- Edge associations:** -

**Routes** Subnet associations Edge associations Route propagation Tags

**Explicit subnet associations (0)**

Find subnet association

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

Edit subnet associations

**Subnets without explicit associations (1)**

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	-

Edit subnet associations

## 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**.

The screenshot shows the AWS Management Console interface for editing subnet associations. The breadcrumb trail at the top reads: VPC > Route tables > rtb-0704f15461ee91808 > Edit subnet associations. The main heading is "Edit subnet associations" with a subtext "Change which subnets are associated with this route table." Below this, there are two sections: "Available subnets (1/1)" and "Selected subnets".

The "Available subnets" section contains a table with the following data:

<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)

The "Selected subnets" section shows a single entry: "subnet-047a9d5f8971f4e64 / Private-Subnet" with a close button (X). At the bottom right of the main content area, there are two buttons: "Cancel" and "Save associations".

The footer of the console shows "CloudShell", "Feedback", and copyright information: "© 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**.

**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  
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 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](#)

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

**VPC - required** [info](#)  
vpc-090f667eb0299017a (MyVPC)  
10.0.0.0/16

**Subnet** [info](#)  
subnet-047a9d5f8971f4e64 Private-Subnet  
VPC: vpc-090f667eb0299017a Owner: 343218194491  
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 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 \_/[!@#%&'()\*+,-.:/:;]{0,255}

**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** [>](#)

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.

## 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.

