# Placement Empowerment Program

***Cloud Computing and DevOps Centre***

***Set a private network in cloud – Create a VPC with subnets for your instances. Configure routing for internal communication between subnets***

Name :JOSHUA MOSES Department: IT



**Introduction**

A **Virtual Private Cloud (VPC)** is a logically isolated segment within a cloud provider's infrastructure, offering full control over network configuration, security, and access management. A VPC facilitates the deployment of cloud resources in a secure environment while allowing controlled exposure to external networks as required. The setup involves defining subnets, configuring route tables, and implementing security mechanisms to regulate traffic flow and enforce access policies.

**Objectives**

1. VPC Creation – Establish a private cloud network with a designated IPv4 CIDR block tailored to application requirements.
2. Subnet Configuration – Architect multiple subnets (public/private) within the VPC to logically separate workloads.
3. Routing Table Configuration – Define custom route tables to control packet flow between subnets and external endpoints.
4. Security Implementation – Leverage security groups and network ACLs to restrict inbound and outbound traffic based on predefined rules.
5. High Availability – Distribute resources across multiple Availability Zones (AZs) to enhance resilience and fault tolerance.

## **Key Benefits**

* **Security** – Enforces resource isolation while enabling controlled access through security groups and ACLs.
* **Customization** – Supports flexible network topology, including **private IP addressing**, subnet segmentation, and custom routing.
* **Cost Optimization** – Reduces unnecessary data transfer costs and optimizes resource allocation.
* **Scalability** – Adapts dynamically to increasing workloads without compromising security or performance.
* **Network Control** – Provides granular control over **IP address ranges, route propagation, and access permissions**.

# Step-by-Step Overview

**Step 1: Access AWS Management Console**

1. Navigate to **AWS Management Console**.
2. Authenticate using your credentials (**username and password**).



**Step 2: VPC Configuration**

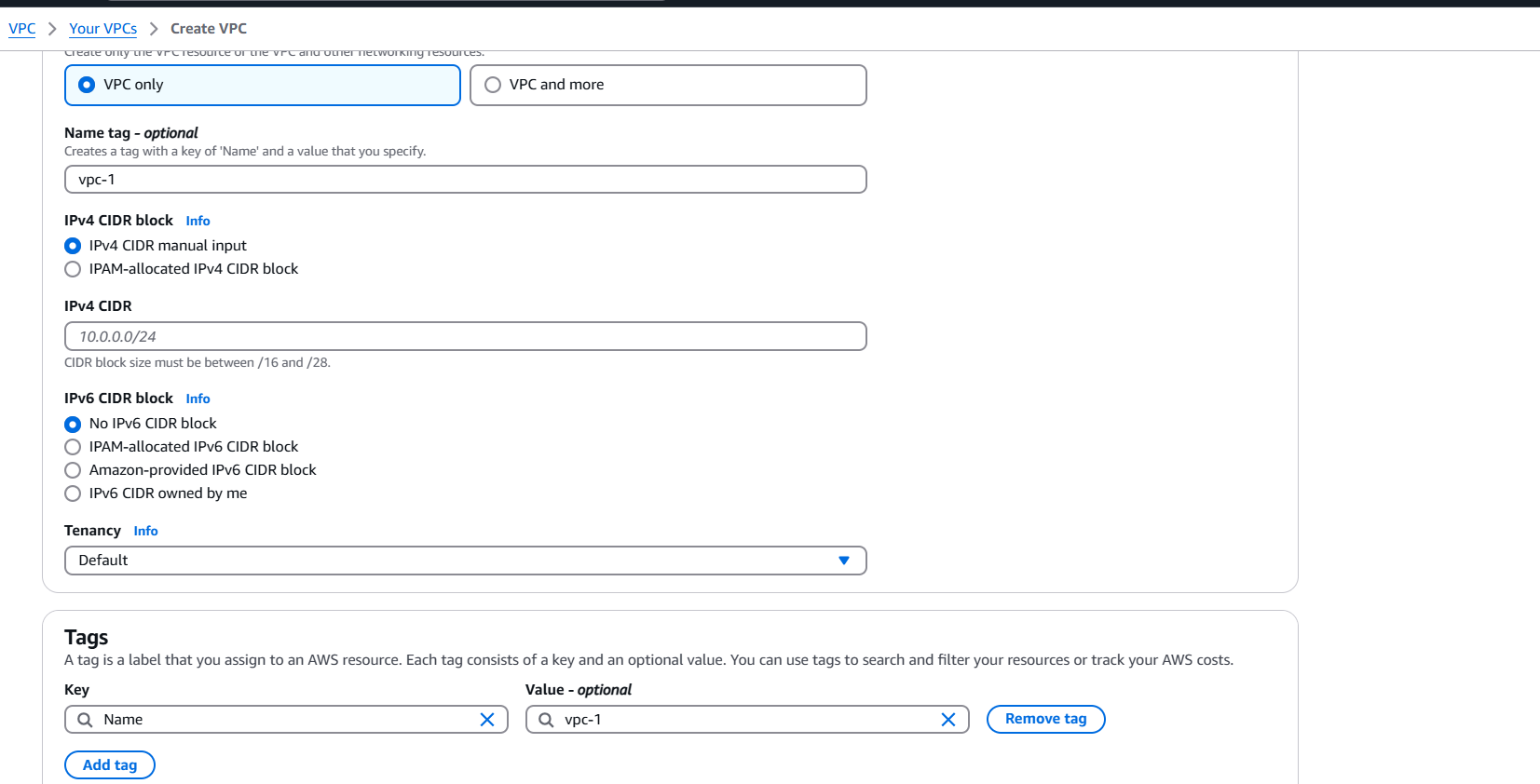
**Navigate to VPC Dashboard**

* From the **AWS Services** menu, select **VPC** to access the **VPC Management Console**.

**Create a VPC**

1. Click on **Your VPCs** → **Create VPC**.
2. Provide the following details:
   * **Name Tag:** Custom identifier (e.g., MyPrivateVPC).
   * **IPv4 CIDR Block:** 10.0.0.0/16 (provides 65,536 IP addresses).
   * **IPv6 CIDR Block:** Optional.
   * **Tenancy:** Default (for shared hardware usage).
3. Click **Create VPC**.

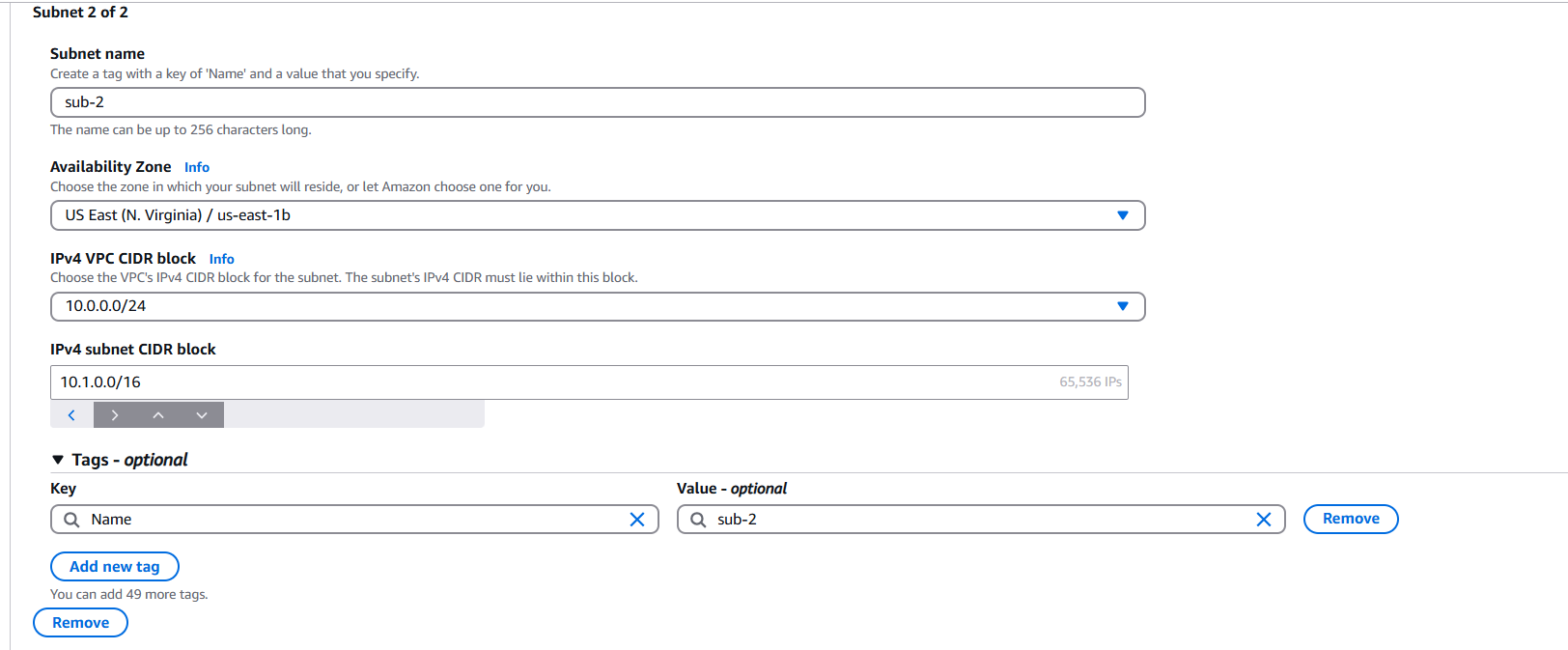


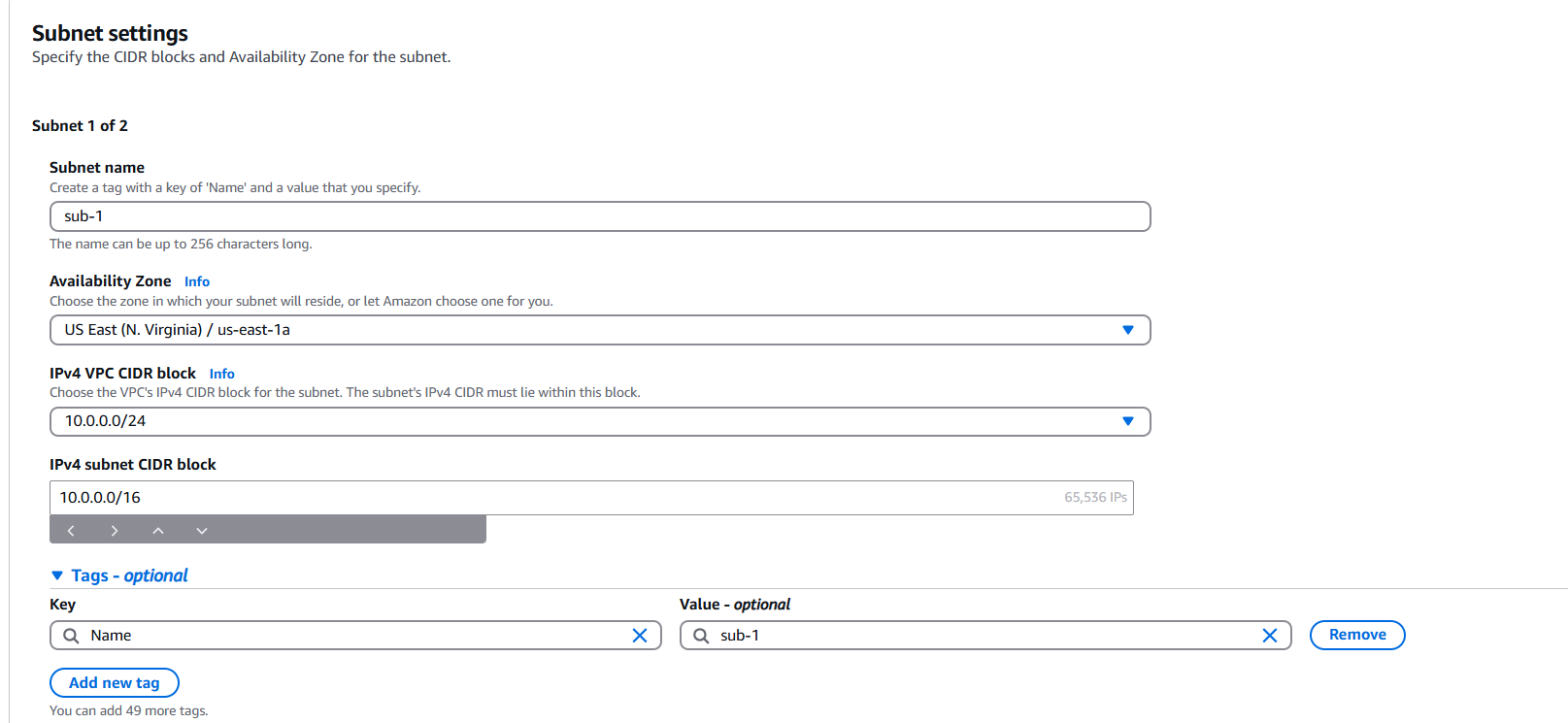


**Step 3: Subnet Creation**

**Define Subnets**

1. Navigate to **Subnets** → **Create Subnet**.
2. Select the **VPC (MyPrivateVPC)**.
3. Define the required subnets:
   * **Private Subnet A**
     + **CIDR Block:** 10.0.1.0/24
     + **Availability Zone:** us-east-1a
   * **Private Subnet B**
     + **CIDR Block:** 10.0.2.0/24
4. Click **Create Subnet**.





## **Step 4: Route Table Configuration**

## Navigate to **Route Tables** → **Create Route Table**.

## Provide the following details:

## **Name Tag:** PrivateRouteTable.

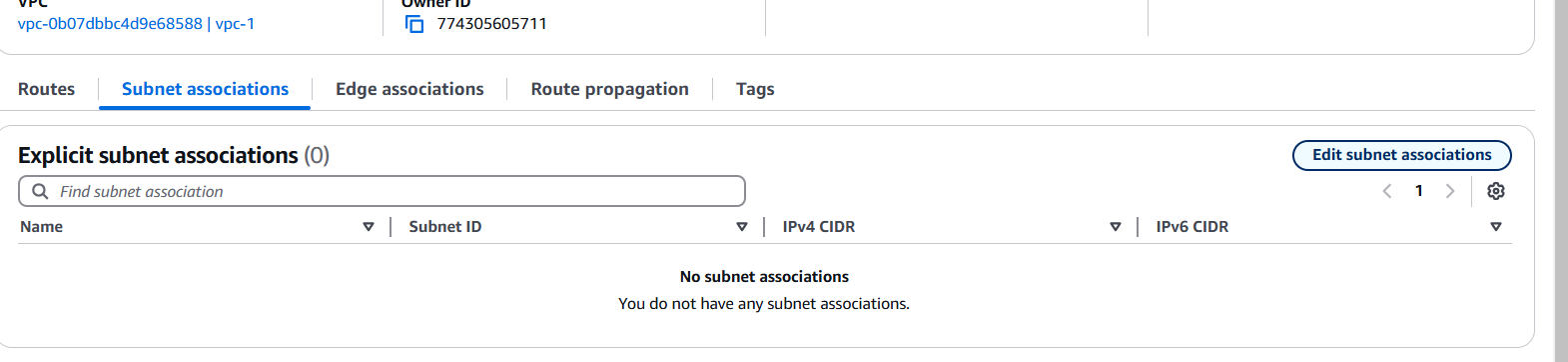
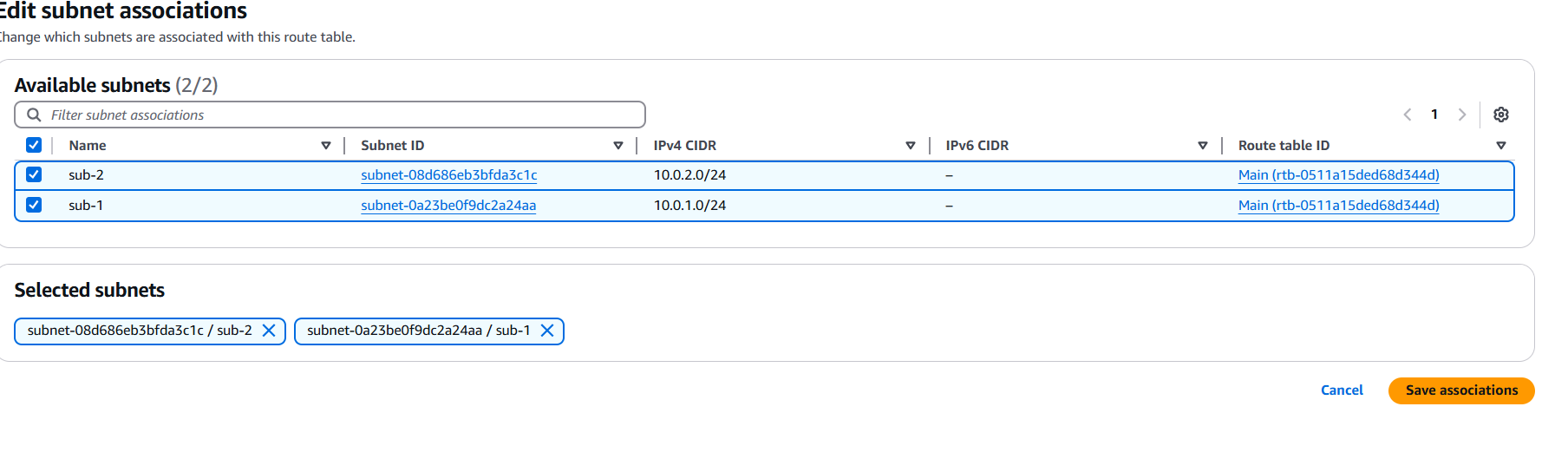
## **VPC:** MyPrivateVPC.

## Click **Create Route Table**.

## 

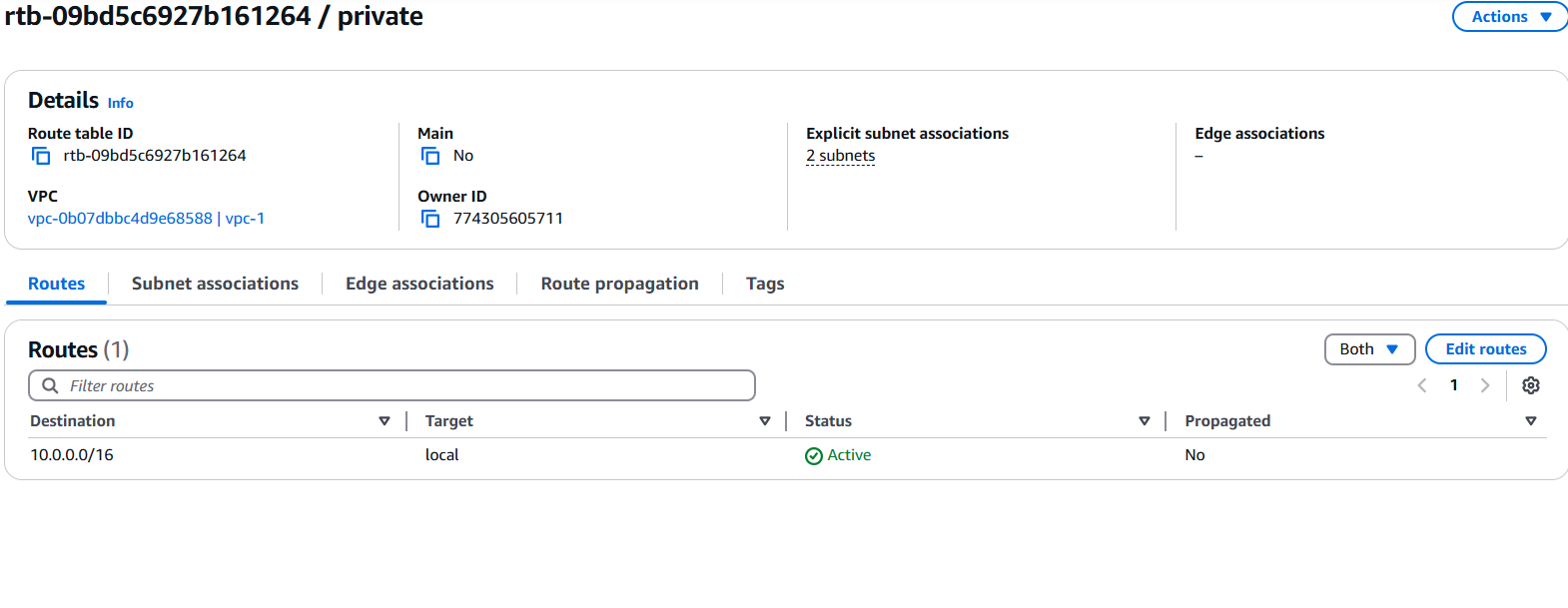
**Step 5: Associate Subnets with Route Table**

1. Navigate to **Subnet Associations** → **Edit Subnet Associations**.
2. Select **Private Subnet A** and **Private Subnet B**.
3. Click **Save Associations**.



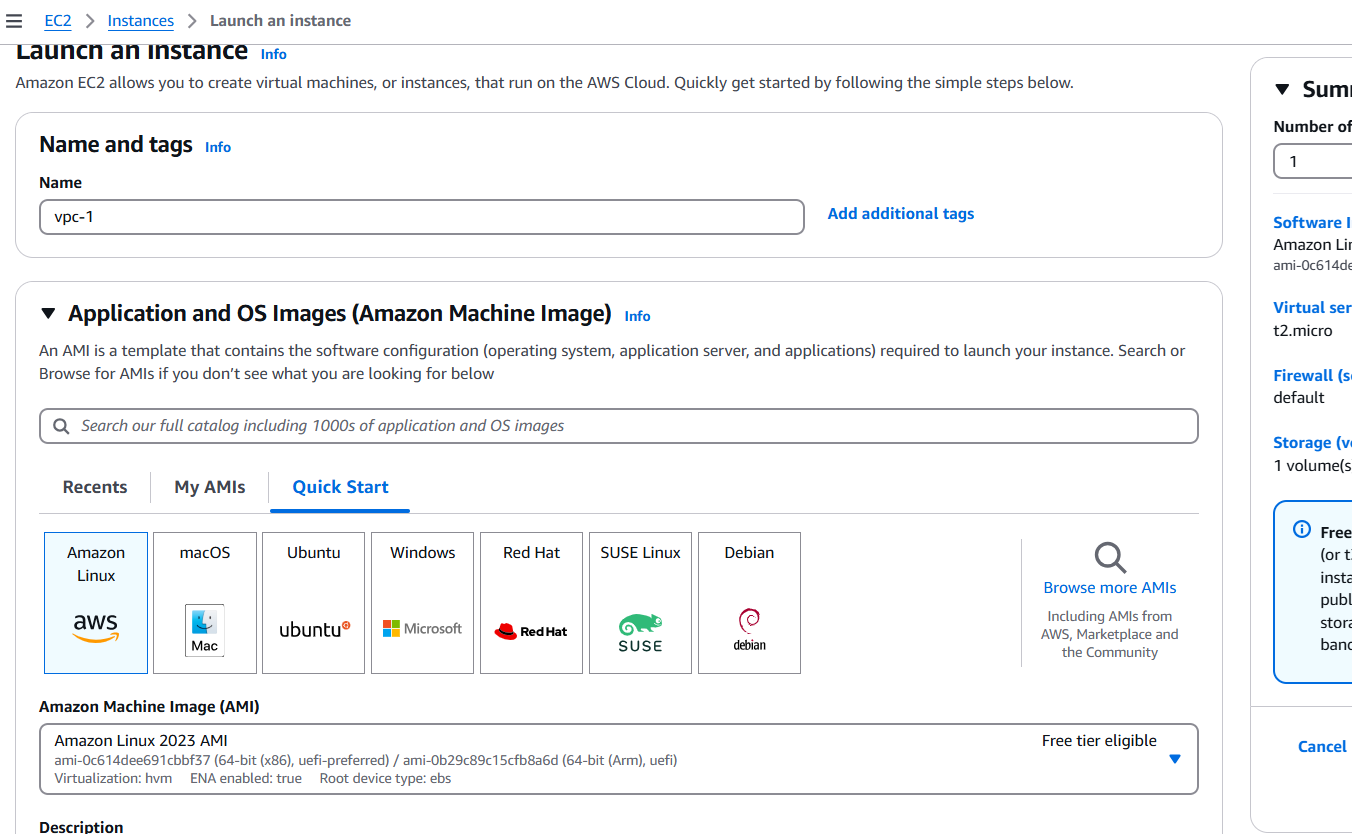
## Step 6:

Default route: 10.0.0.0/16 → local (Automatically added).



**Step 7: Launch EC2 Instances in Private Subnets**

1. Navigate to **EC2 Dashboard** → **Launch Instance**.
2. Select an **Amazon Machine Image (AMI)** (e.g., **Amazon Linux 2, Ubuntu**).
3. Choose an **Instance Type** (e.g., t2.micro).
4. Under **Network Settings**:
   * Select **MyPrivateVPC**.
   * Choose **Private Subnet A** or **Private Subnet B**.
   * Disable **Auto-assign Public IP** (to ensure instance remains private).
5. Click **Launch**.



**Step 8: Enable Internal Communication**

* Instances within the private subnets communicate internally without requiring an **Internet Gateway (IGW)**.
* If external access is needed (e.g., for software updates), set up a **NAT Gateway** in a **Public Subnet**.
* Configure **Security Groups** to:
  + Allow SSH access only from within the VPC (10.0.0.0/16).
  + Restrict inbound traffic based on workload requirements.

**Step 9: Final Validation**

Your private cloud network is now operational, ensuring secure internal communication and controlled external access.

For additional configurations such as **VPN Connectivity, Bastion Host Deployment, or NAT Gateway Setup**, further refinements can be implemented based on requirements.

**Expected Outcome**

Upon completion of these steps, you will have: A fully isolated VPC with a structured network architecture. Subnets tailored for internal workloads, with private and public segmentation. Properly configured routing policies ensuring seamless intra-VPC communication and external connectivity where necessary.

For further optimizations, consider VPC Peering, AWS Transit Gateway, or Direct Connect based on network expansion needs.