**VPC TASK-2**

**Create one VPC, with 1 public subnet and 1 private subnet.**

**Step 1: Create a VPC**

1. Go to **AWS Console → VPC → Your VPCs → Create VPC**.
2. Choose **VPC only**.
3. Enter:
   * **Name tag**: MyVPC
   * **IPv4 CIDR block**: 10.0.0.0/16 (big enough for subnets)
   * Tenancy: Default
4. Click **Create VPC**.

**Step 2: Create Subnets**

1. Go to **Subnets → Create subnet**.
2. Choose your VPC (MyVPC).

* **Public Subnet**:
  + Name: Public-Subnet
  + Availability Zone: pick one (e.g., us-east-1a)
  + CIDR block: 10.0.1.0/24
* **Private Subnet**:
  + Name: Private-Subnet
  + Availability Zone: same or different (e.g., us-east-1b)
  + CIDR block: 10.0.2.0/24

Click **Create Subnet**.

**Step 3: Internet Gateway (for Public Subnet)**

1. Go to **Internet Gateways → Create internet gateway**.
2. Name: MyIGW.
3. Attach it to your VPC (MyVPC).

**Step 4: Route Tables**

1. Go to **Route Tables → Create route table**.
   * Name: Public-RT
   * VPC: MyVPC.
2. Select Public-RT → **Subnet Associations** → Add Public-Subnet.
3. In **Routes tab**, add:
   * Destination: 0.0.0.0/0
   * Target: Internet Gateway (MyIGW)

✅ Now Public-Subnet has internet access.

**Step 5: Private Route Table**

1. Create another route table: Private-RT.
2. Associate it with Private-Subnet.
3. For now, leave routes as **local only** (no internet).

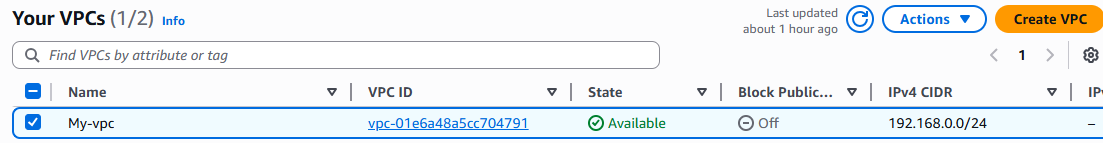
*(Later you can add NAT Gateway if you want private subnet internet access)*

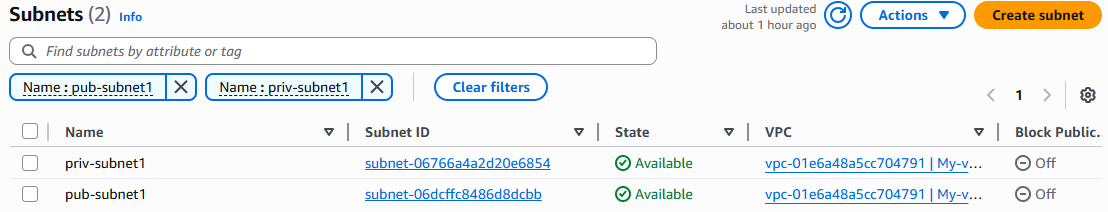
**Step 6: Auto-assign Public IP for Public Subnet**

1. Go to **Subnets → Public-Subnet → Edit subnet settings**.
2. Enable **Auto-assign public IPv4 address**.

✅ Done! You now have:

* **1 Public Subnet** → internet access via IGW.
* **1 Private Subnet** → isolated, no internet (unless you add NAT).





**2. Enable VPC peering for cross-region.**

# **Steps to Create Cross-Region VPC Peering**

### **Step 1: Prerequisites**

* You have **two VPCs** in **different regions** (for example:
  + VPC-A in us-east-1
  + VPC-B in us-west-2)
* Both must have **non-overlapping CIDR ranges** (e.g., 10.0.0.0/16 and 10.1.0.0/16).

### **Step 2: Create Peering Connection**

1. Go to **VPC Console** in **Region A (us-east-1)**.
2. **Peering Connections → Create Peering Connection**.
3. Configure:
   * **Name**: VPC-A-to-VPC-B
   * **VPC (Requester)**: Choose VPC-A
   * **VPC (Accepter)**: Enter VPC ID of VPC-B (from us-west-2).
   * **Region**: Select us-west-2.
4. Click **Create Peering Connection**.

### **Step 3: Accept Peering Connection**

1. Go to **VPC Console → us-west-2** (Region B).
2. Under **Peering Connections**, find the request.
3. Select it → **Accept request**.

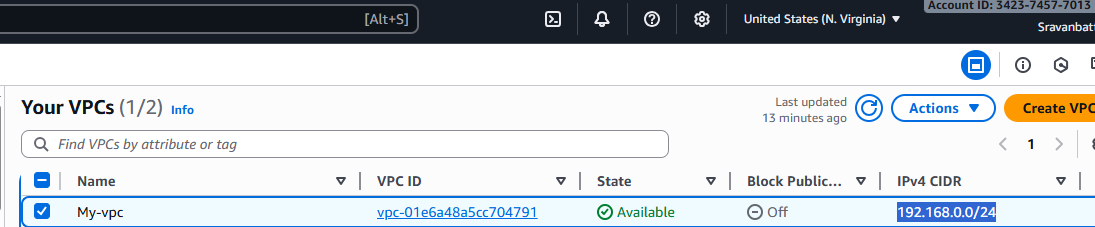
### **Step 4: Update Route Tables**

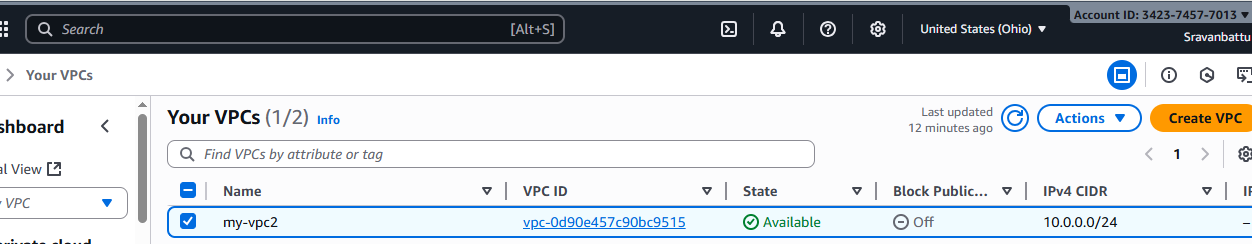
You need to update **both VPC route tables** so traffic can flow.

* **In VPC-A (us-east-1):**
  + Go to Route Tables → choose subnet’s route table.
  + Add route:
    - Destination: CIDR of VPC-B (10.1.0.0/16)
    - Target: Peering Connection ID.
* **In VPC-B (us-west-2):**
  + Add route:
    - Destination: CIDR of VPC-A (10.0.0.0/16)
    - Target: Peering Connection ID.

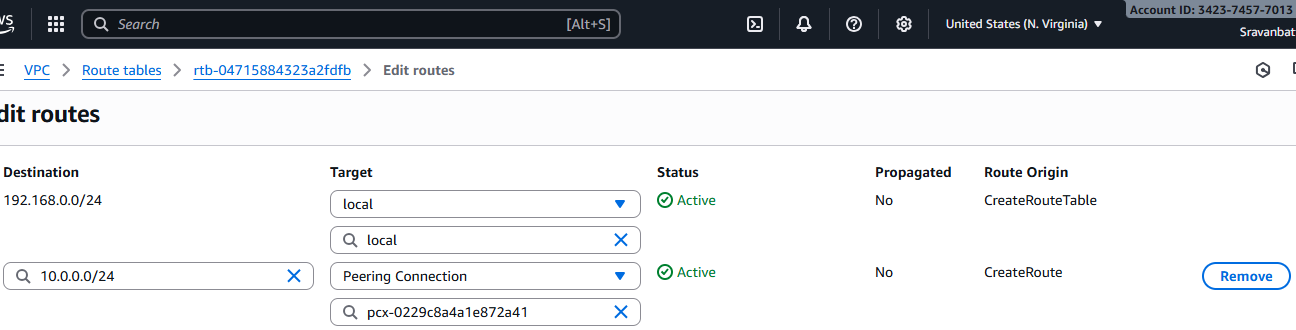
### **Step 5: Security Groups**

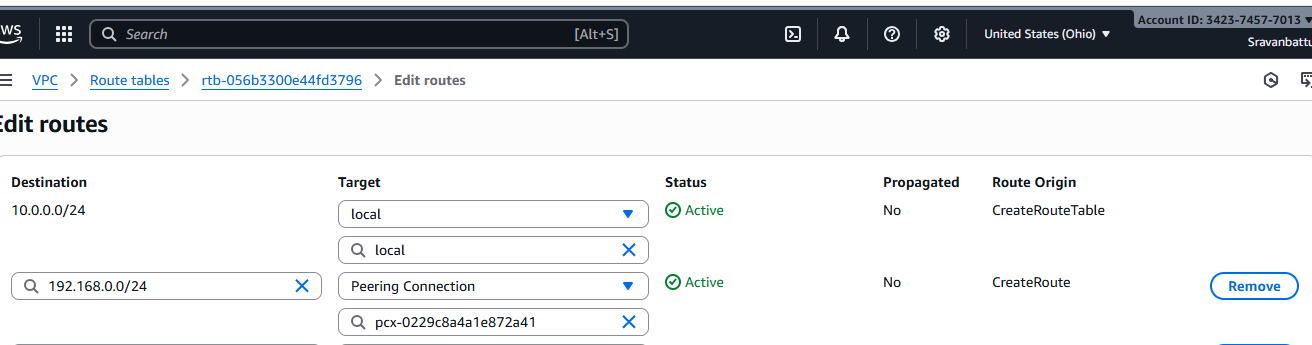
* Allow inbound traffic from the **peer VPC CIDR**.  
  Example:
  + In VPC-A SG → allow 10.1.0.0/16.
  + In VPC-B SG → allow 10.0.0.0/16

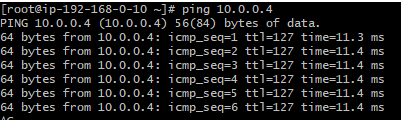




**Peering 1**



**Perring 2**

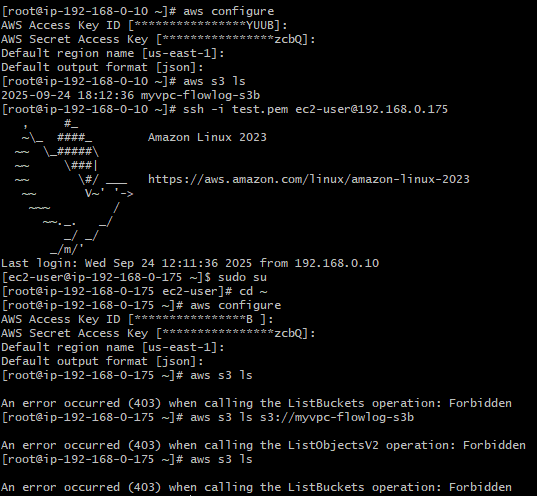


**5.Set up a VPC Endpoint.**

## Setting Up a VPC Endpoint

Here’s how to do it from the AWS Console (example: **S3 Gateway Endpoint**):

1. **Navigate:** VPC → Endpoints → Create Endpoint.
2. **Service category:** “AWS services”.
3. **Service name:** Search and select (e.g., com.amazonaws.<region>.s3).
4. **VPC:** Choose your VPC.
5. **Subnets / Route tables:**
   * If **Gateway endpoint (S3/DynamoDB)** → Select **Route Tables** (public + private as needed).
   * If **Interface endpoint** → Select the **subnets** (one per AZ ideally, e.g., your private subnet for private access).



**4.Set up a VPC Transit Gateway.**

### Architecture Overview

* **VPCs**: 4 VPCs, each with:
  + CIDR Block: 10.0.x.0/24 (where x is 1 to 4)
  + One **public subnet** in each VPC
  + One **EC2 instance** in the public subnet
* **Transit Gateway**: A central hub connecting all VPCs
* **Internet Gateway**: Attached to each VPC for internet access

### 🛠️ Step-by-Step Setup in AWS Console

#### 1. **Create 4 VPCs**

1. Navigate to the **VPC Dashboard**.
2. Click **Create VPC**.
3. For each VPC:
   * **CIDR Block**: 10.0.x.0/24 (replace x with 1 to 4)
   * **Name Tag**: VPC-x
   * **Tenancy**: Default
4. Click **Create VPC**.

#### 2. **Create Public Subnets in Each VPC**

1. In the **VPC Dashboard**, go to **Subnets**.
2. Click **Create subnet**.
3. For each VPC:
   * **VPC**: Select the corresponding VPC
   * **Subnet Name**: Public-Subnet-x
   * **Availability Zone**: Select any available zone
   * **CIDR Block**: 10.0.x.0/25 (adjust x accordingly)
4. Click **Create subnet**.

#### 3. **Launch EC2 Instances in Public Subnets**

1. Navigate to the **EC2 Dashboard**.
2. Click **Launch Instance**.
3. For each VPC:
   * **AMI**: Choose an Amazon Linux 2 AMI
   * **Instance Type**: t2.micro (eligible for the free tier)
   * **Network**: Select the corresponding VPC
   * **Subnet**: Select the corresponding public subnet
   * **Auto-assign Public IP**: Enable
   * **Security Group**: Create a new security group allowing:
     + **SSH** (port 22) from your IP
     + **ICMP** (ping) from the CIDR blocks of other VPCs
4. Launch the instance and note the **Public IP** for SSH access.

#### 4. **Create a Transit Gateway**

1. In the **VPC Dashboard**, go to **Transit Gateways**.
2. Click **Create transit gateway**.
3. Configure:
   * **Name**: My-Transit-Gateway
   * **Description**: Central hub for 4 VPCs
   * **Amazon ASN**: 64512 (default)
   * **Default Route Table Association**: Enable
   * **Default Route Table Propagation**: Enable
4. Click **Create transit gateway**.

#### 5. **Create Transit Gateway Attachments for Each VPC**

1. In the **Transit Gateway Dashboard**, select your transit gateway.
2. Go to the **Attachments** tab and click **Create attachment**.
3. For each VPC:
   * **Attachment Type**: VPC
   * **VPC**: Select the corresponding VPC
   * **Subnet**: Select the corresponding public subnet
4. Click **Create attachment** and wait for the status to become **Available**.

#### 6. **Update Route Tables**

1. **Transit Gateway Route Table**:
   * Go to the **Transit Gateway Route Tables** section.
   * Select the default route table.
   * Click **Edit routes** and add routes for each VPC's CIDR block, targeting the respective VPC attachment.
2. **VPC Route Tables**:
   * For each VPC:
     + Go to the **Route Tables** section.
     + Select the route table associated with the public subnet.
     + Click **Edit routes** and add a route for the CIDR blocks of the other VPCs, targeting the transit gateway.

#### 7. **Test Connectivity**

1. SSH into the EC2 instance in one VPC using its public IP.
2. From the terminal, try pinging the private IPs of EC2 instances in other VPCs to test connectivity.

