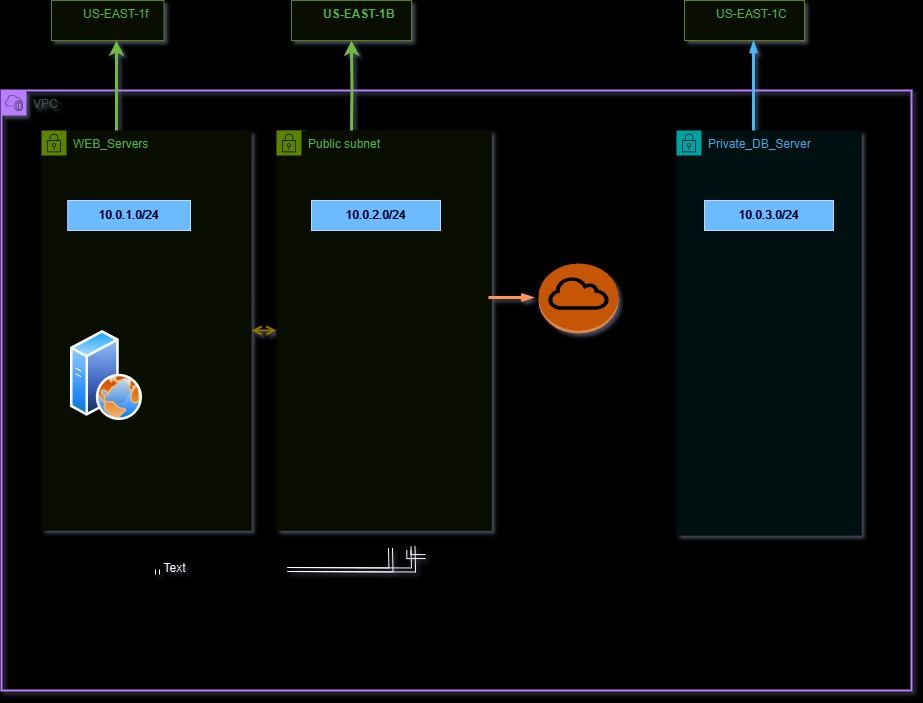
##### **Overview**

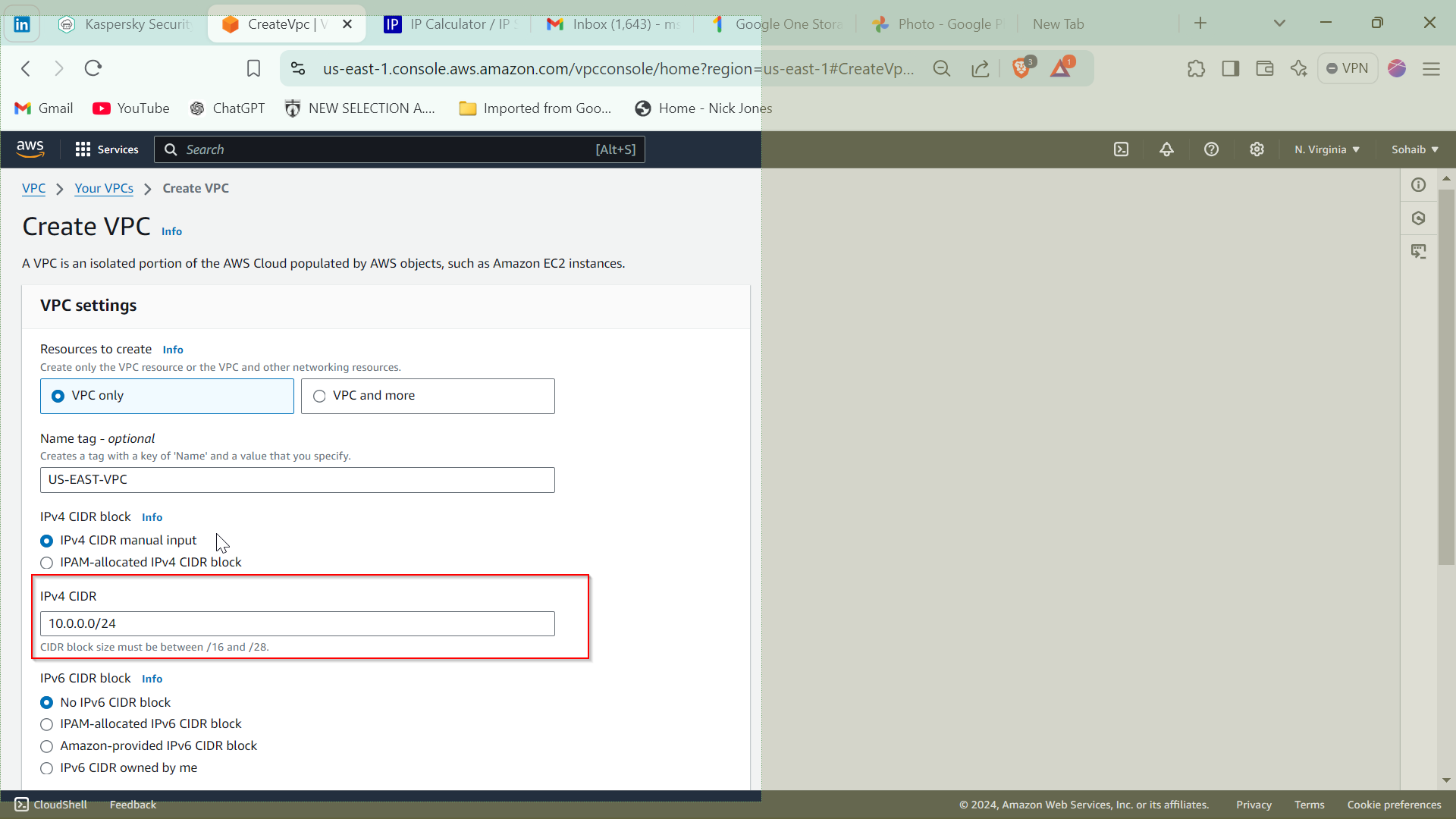
Hey, I've designed a network demonstrating high availability and maximum security. This setup is for MegaCorp, which is located in the US East region.

## **Step-by-Step Guide to Creating a VPC with Public and Private Subnets**



### **Step 1: Create a VPC**

1. **Navigate to the AWS Management Console** and go to the **VPC Dashboard**.
2. Click **Create VPC**.
3. Fill in the details:
   1. **Name**: Megacorp
   2. **IPv4 CIDR Block**: 10.0.0.0/16



1. Click **Create VPC**.

**Why This Step?**

* A **Virtual Private Cloud (VPC)** is an isolated network within AWS where you can deploy resources securely.
* Defining a **CIDR block** (10.0.0.0/16) provides a large address range, which allows you to create multiple subnets within your VPC.

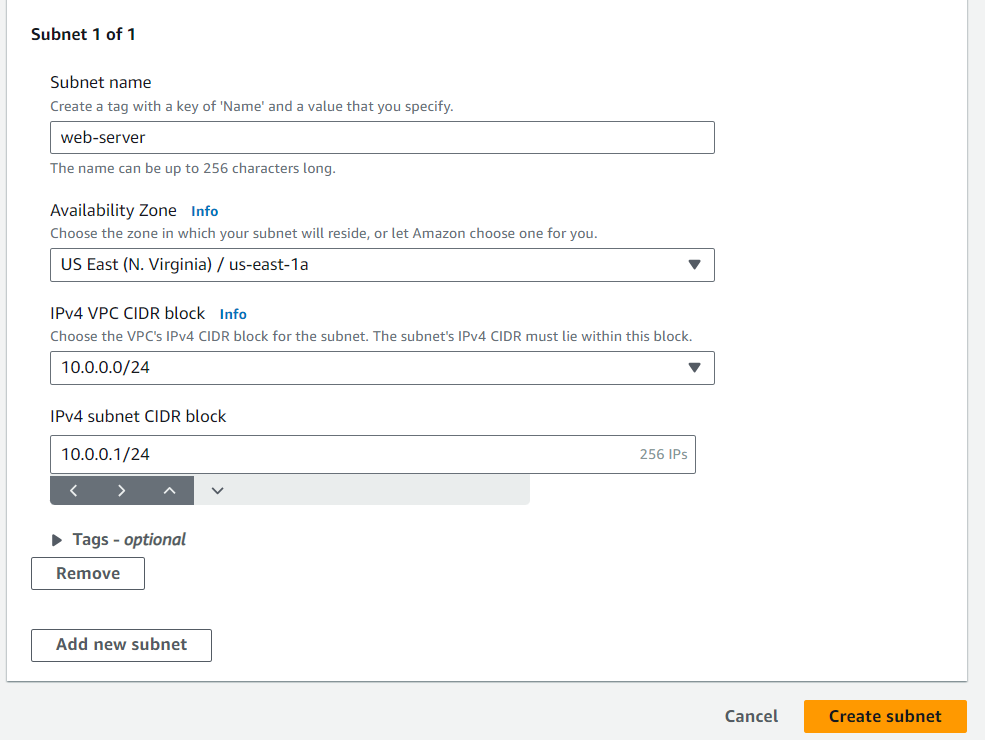
**Why This Is a Great Step:**

* You get complete control over your network architecture.
* Isolation enhances security and helps you manage resources efficiently.

### **Step 2: Create the Subnets**

#### **1. Create the Web Servers Subnet (Public Subnet)**

1. Go to **Subnets** and click **Create Subnet**.
2. Fill in the details:
   1. **Name**: WEB\_Servers
   2. **VPC**: Select megacoprp
   3. **Availability Zone**: us-east-1f
   4. **IPv4 CIDR Block**: 10.0.1.0/24
3. Click **Create Subnet**.



**Why This Step?**

* This subnet is designated for web servers that need to be accessible from the internet.
* A **public subnet** allows resources to receive public IP addresses and communicate with the internet.

**Why This Is a Great Step:**

* Placing web servers in a public subnet ensures they can handle client requests directly.
* Separating web servers from the database increases security by keeping different roles isolated.

#### **2. Create the Public Subnet**

1. Go to **Subnets** and click **Create Subnet**.
2. Fill in the details:
   1. **Name**: App\_server
   2. **VPC**: Select Megacorp
   3. **Availability Zone**: us-east-1b
   4. **IPv4 CIDR Block**: 10.0.2.0/24
3. Click **Create Subnet**.

**Why This Step?**

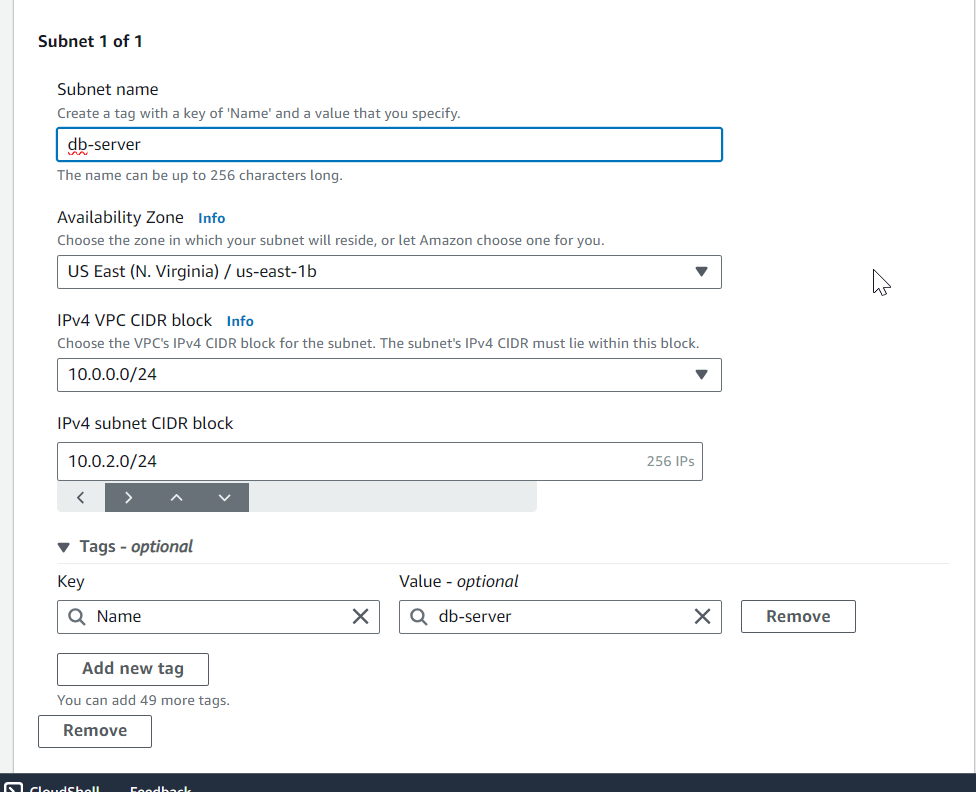
* This subnet provides an additional public-facing segment for future use, such as load balancers or additional web resources.

**Why This Is a Great Step:**

* Ensures flexibility and scalability for future deployments.
* Distributing resources across different availability zones enhances **high availability**.

#### **3. Create the Database Subnet (Private Subnet)**

1. Go to **Subnets** and click **Create Subnet**.
2. Fill in the details:
   1. **Name**: Private\_DB\_Server
   2. **VPC**: Select megacorp-vpc
   3. **Availability Zone**: us-east-1b
   4. **IPv4 CIDR Block**: 10.0.3.0/24



1. Click **Create Subnet**.

**Why This Step?**

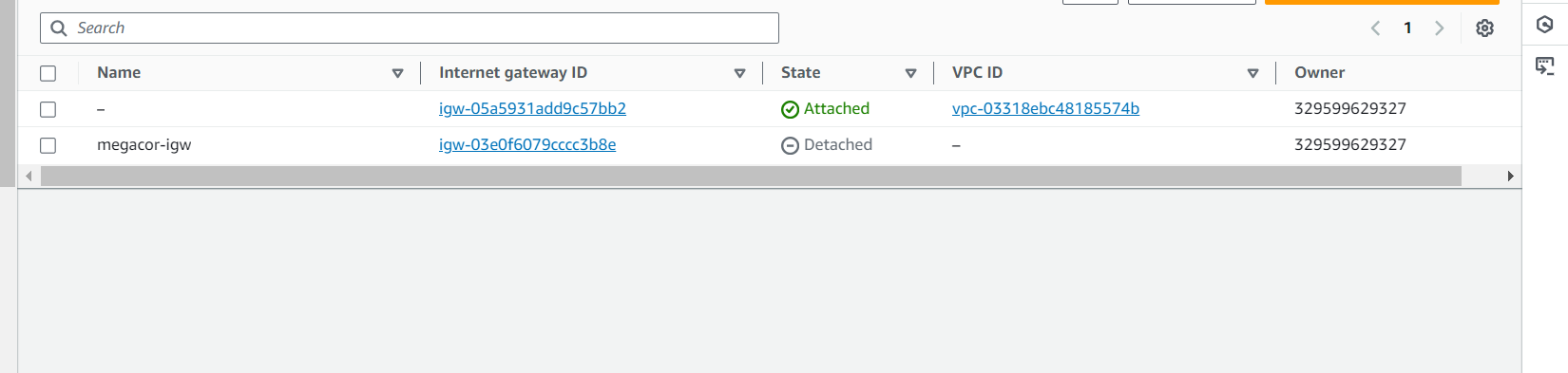
* This subnet is for database servers that should not be accessible from the internet for security reasons.
* **Private subnets** restrict public internet access, enhancing database security.

**Why This Is a Great Step:**

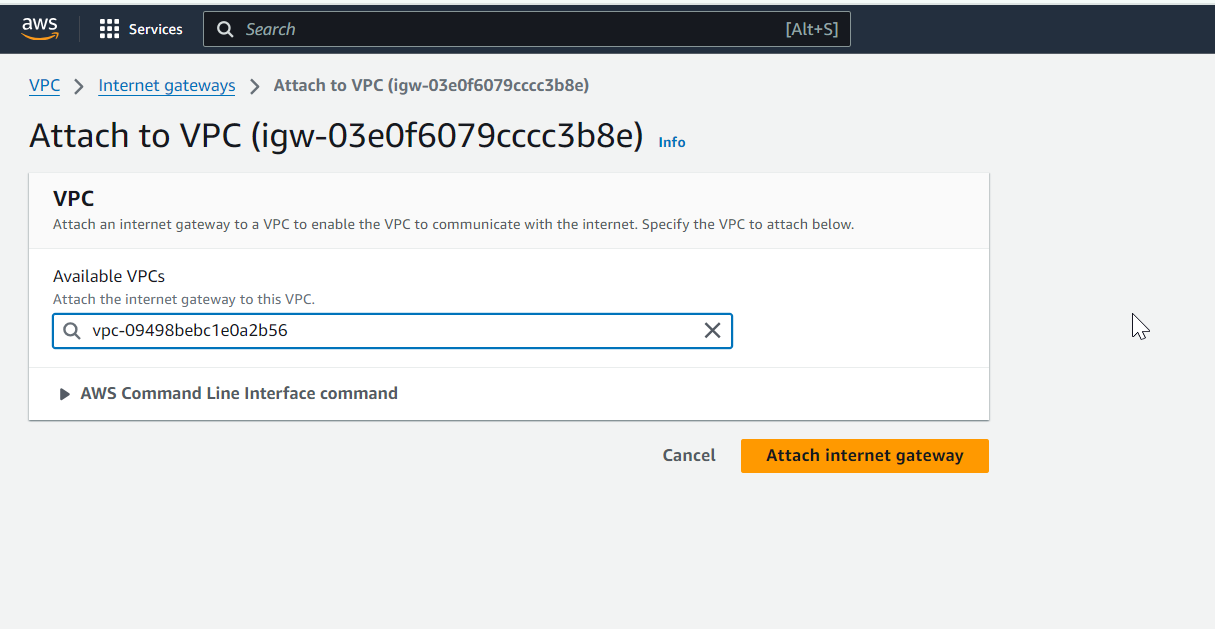
* Protects sensitive data by isolating databases from the public network.
* Follows the principle of **least privilege** by limiting exposure to the internet.

### **Step 3: Configure Internet Gateway for Public Subnets**

1. Go to **VPC Dashboard > Internet Gateways**.
2. Click **Create Internet Gateway**.
   1. **Name**: megacorp-igw



1. Attach the Internet Gateway to your VPC:
   1. Select MyIGW and click **Actions > Attach to VPC**.
   2. Choose megacorp-vpc.



**Why This Step?**

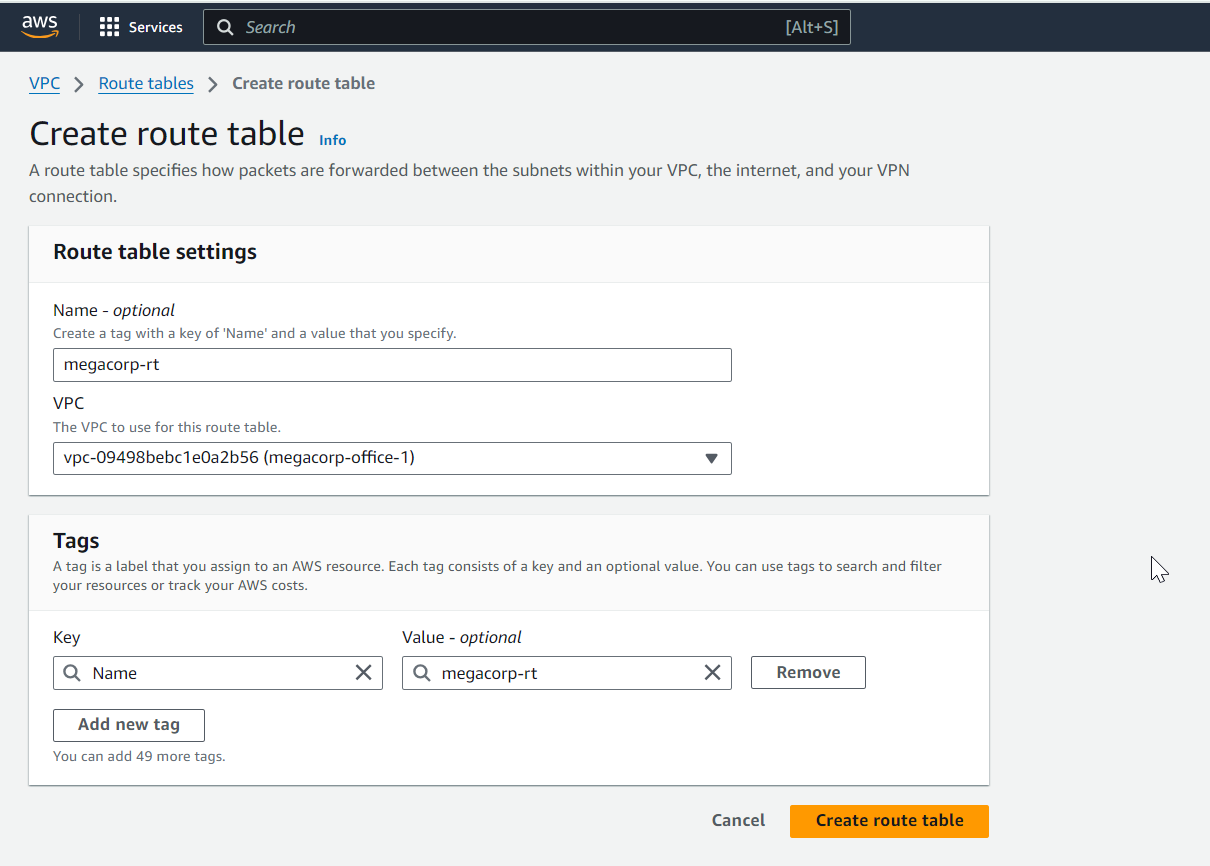
* An **Internet Gateway (IGW)** allows traffic to flow between the public subnets and the internet.

**Why This Is a Great Step:**

* Enables public-facing resources (like web servers) to serve client requests over the internet.
* Essential for creating a fully functional public subnet.

### **Step 4: Create a Route Table for Public Subnets**

1. Go to **VPC Dashboard > Route Tables**.
2. Click **Create Route Table**.
   1. **Name**: megacorp-rt
   2. **VPC**: megacop-vpc
3. Add a route to the internet:
   1. Select the route table, go to the **Routes** tab, and click **Edit Routes**.
   2. Add the following route:
      1. **Destination**: 0.0.0.0/0
      2. **Target**: MyIGW (Internet Gateway)



1. Click **Save Routes**.

**Why This Step?**

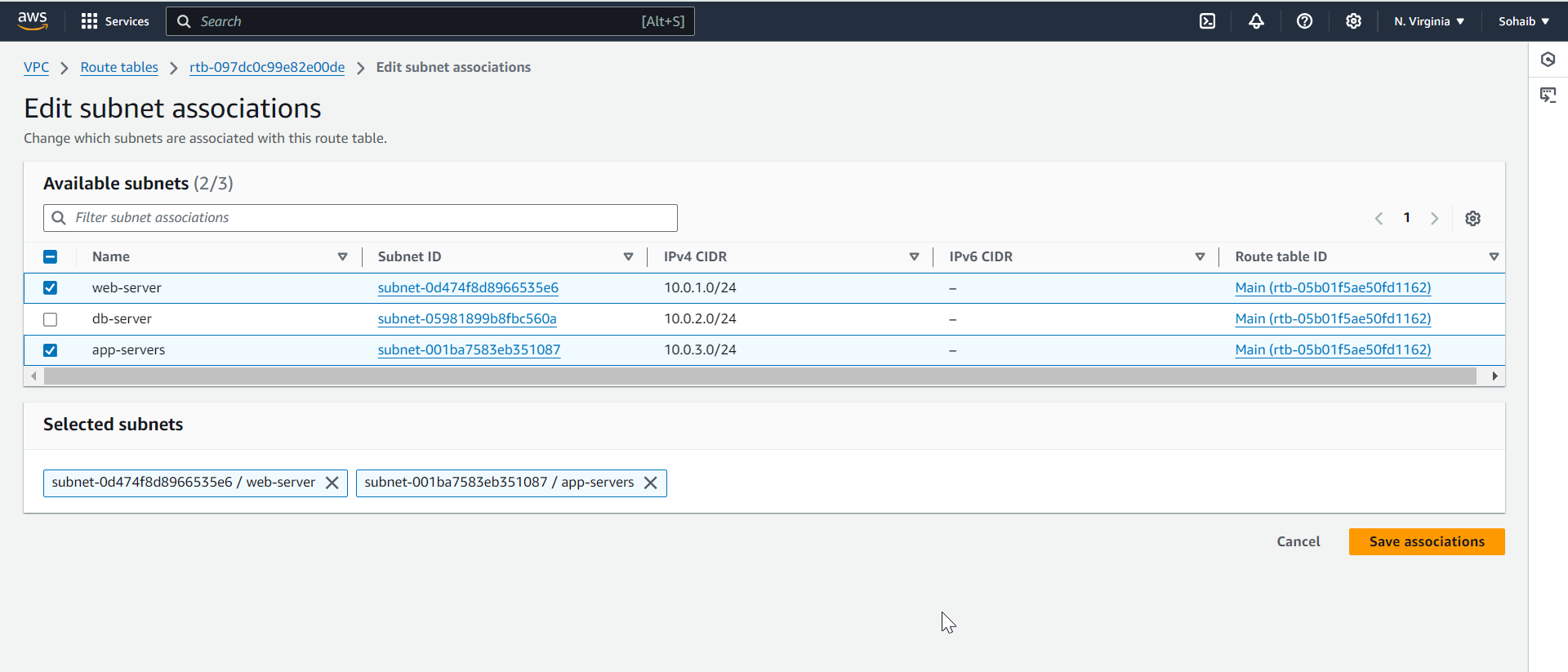
* A **route table** directs traffic from your public subnet to the internet.

**Why This Is a Great Step:**

* Ensures web servers and public resources can communicate with the internet.
* Provides clarity and control over network traffic.

### **Step 5: Associate Public Subnets with the Public Route Table**

1. Select **Public\_Route\_Table**.
2. Go to the **Subnet Associations** tab and click **Edit Subnet Associations**.
3. Select:
   1. WEB\_Servers
   2. DB\_Server
4. Click **Save**.



**Why This Step?**

* Subnets must be explicitly associated with a route table for the routes to take effect.

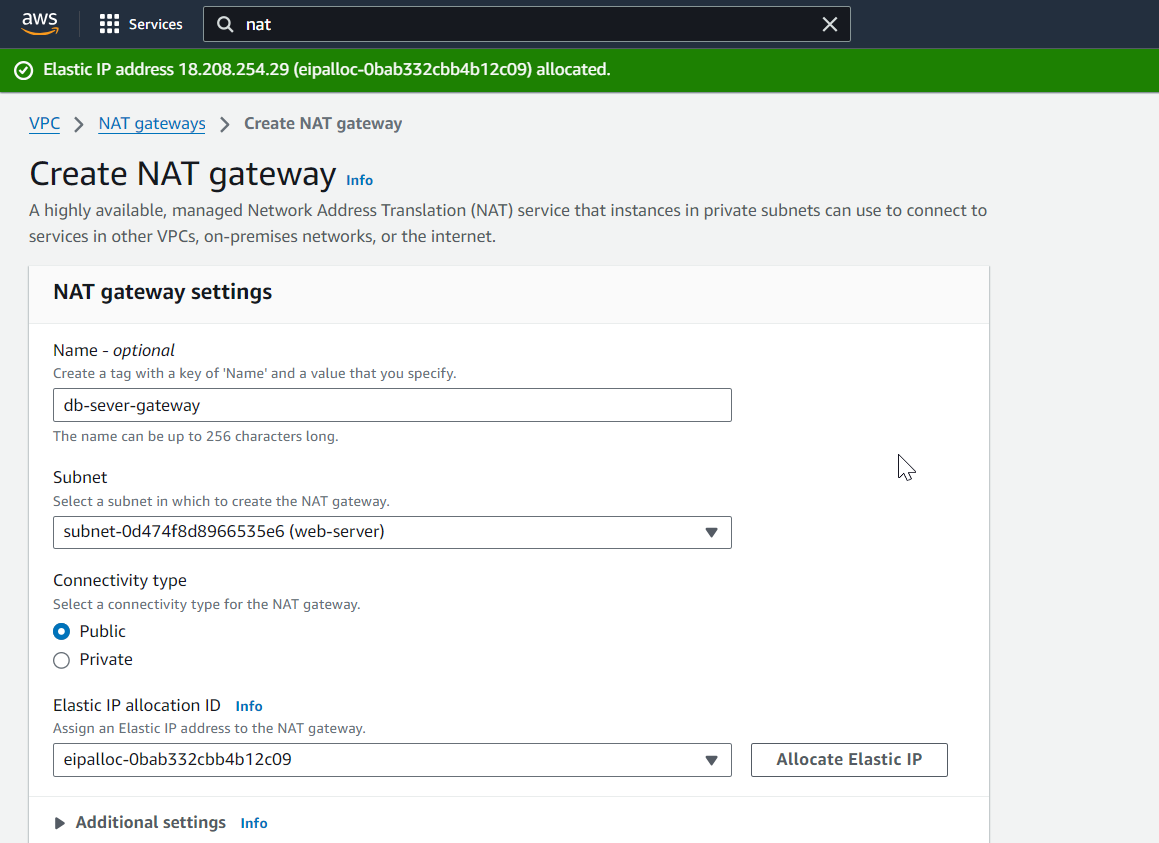
**Why This Is a Great Step:**

* Ensures that web servers in public subnets can route traffic to the internet.
* Facilitates organized and predictable network behavior.

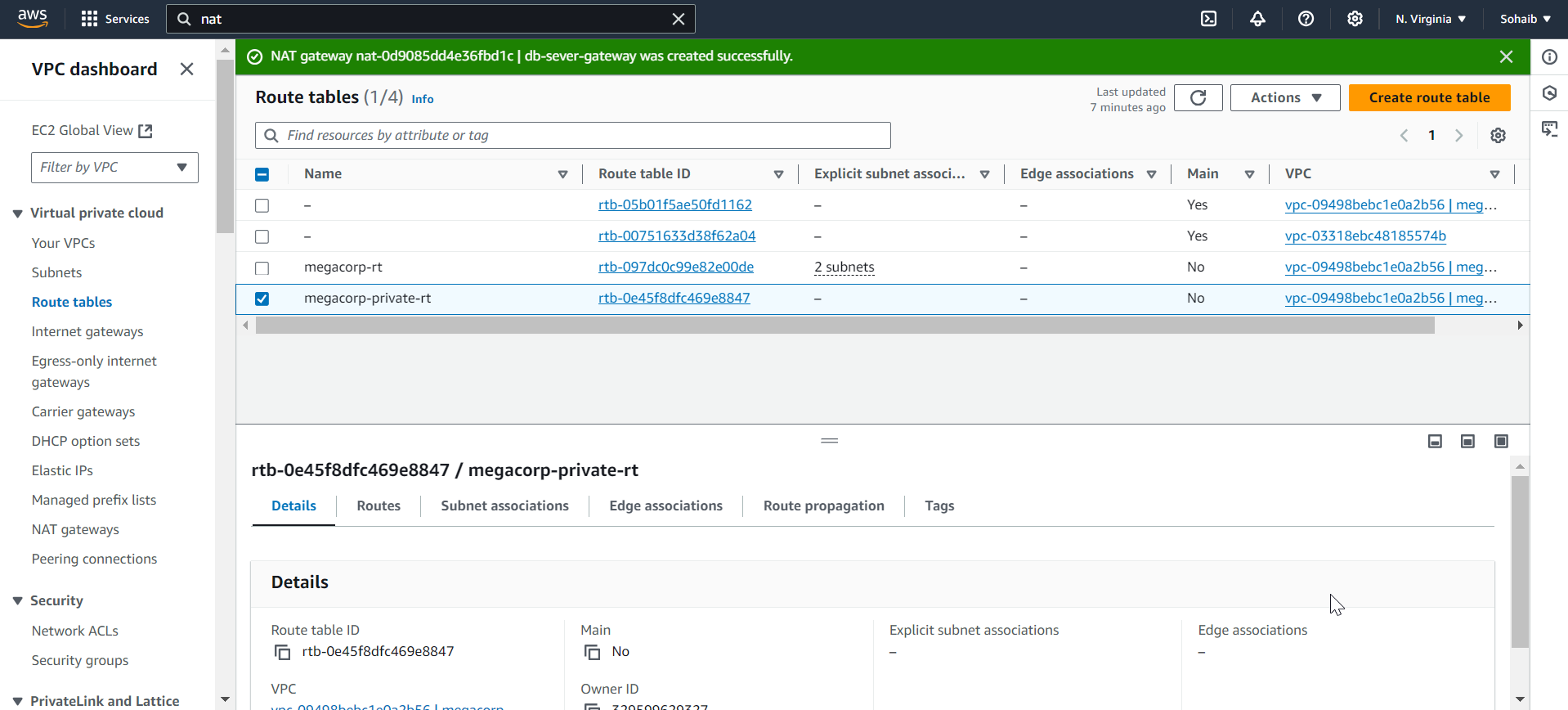
### **Step 6: Configure the Private Subnet**

Since the private subnet (Private\_DB\_Server) does not need direct internet access, it doesn't require an Internet Gateway. If you need outbound internet access (e.g., for updates), follow these steps:

1. **Create a NAT Gateway** in the public subnet.



1. **Add a route** in the private subnet’s route table to direct outbound traffic to the NAT Gateway.



**Why This Step?**

* A **NAT Gateway** allows instances in private subnets to access the internet while keeping them secure.

**Why This Is a Great Step:**

* Provides **controlled internet access** for database servers without exposing them directly.
* Balances security and functionality.

##### Step 7: Launch Resources

1. **Web Servers**: Launch instances in the WEB\_Servers subnet with public IP addresses.
2. **Database Instances**: Launch instances in the Private\_DB\_Server subnet without public IP addresses.

**Why This Step?**

* Deploying resources according to their roles ensures adherence to security best practices.

**Why This Is a Great Step:**

* Isolating web and database layers enhances security and simplifies management.
* Ensures a scalable and secure infrastructure for your application.

### **Summary of CIDR Blocks and Availability Zones**

|  |  |  |
| --- | --- | --- |
| **Subnet Name** | **CIDR Block** | **Availability Zone** |
| WEB\_Servers | 10.0.1.0/24 | us-east-1f |
| Public\_Subnet | 10.0.2.0/24 | us-east-1b |
| Private\_DB\_Server | 10.0.3.0/24 | us-east-1c |

This guide ensures a **secure, scalable, and highly available architecture** for deploying web applications and databases. Each step aligns with **best practices** for cloud security and network design.