Guided Lab: Creating a Custom Virtual Private Cloud (VPC) from scratch

Description

Amazon Virtual Private Cloud (VPC) is a foundational component of AWS that allows you to provision a logically isolated section of the AWS Cloud. Think of a VPC as a private estate within a bustling city (the AWS Cloud). Within this estate, you have the freedom to construct various properties and infrastructure according to your specific requirements.

- **Virtual Private Cloud (VPC)**: Acts as the cornerstone of your network on AWS. Imagine it as owning a large piece of land where you can establish different areas for various purposes — residential, commercial, and recreational.
- **Subnet**: If the VPC is your large piece of land, a subnet is akin to sectioning off parts of this land into different plots. Each plot can be developed independently, with specific characteristics such as security measures and connectivity options, similar to dividing a large estate into smaller, manageable lots for specific uses.
- **Internet Gateway**: This is like the main gate of your estate that allows access to and from the public roads. It manages the traffic between your estate and the outside world, ensuring that residents can reach global destinations outside your private land.
- **Route Table**: Think of this as the map and directional signs within your estate. It guides the traffic, directing it where to go within the estate or how to exit efficiently to reach external destinations. It ensures that all traffic flows smoothly according to predefined rules, like how a traffic control system manages vehicles' movements.

In this lab, you'll gain hands-on experience with these components by setting up a custom VPC from scratch. While AWS accounts come with a default VPC, understanding how to create and configure a custom VPC is essential for tailored network solutions that fit specific security and network requirements.

Prerequisites

This labs assume you have basic knowledge of IP addressing & network subnets, and familiarity with AWS core services like EC2 (Elastic Compute Cloud).

If you find any gaps in your knowledge, consider taking the following lab:

- Creating an Amazon EC2 instance (Linux)
- Setting up a Web server on an EC2 instance
- Launching an EC2 Instance with User Data

Objectives

In this lab, you will:

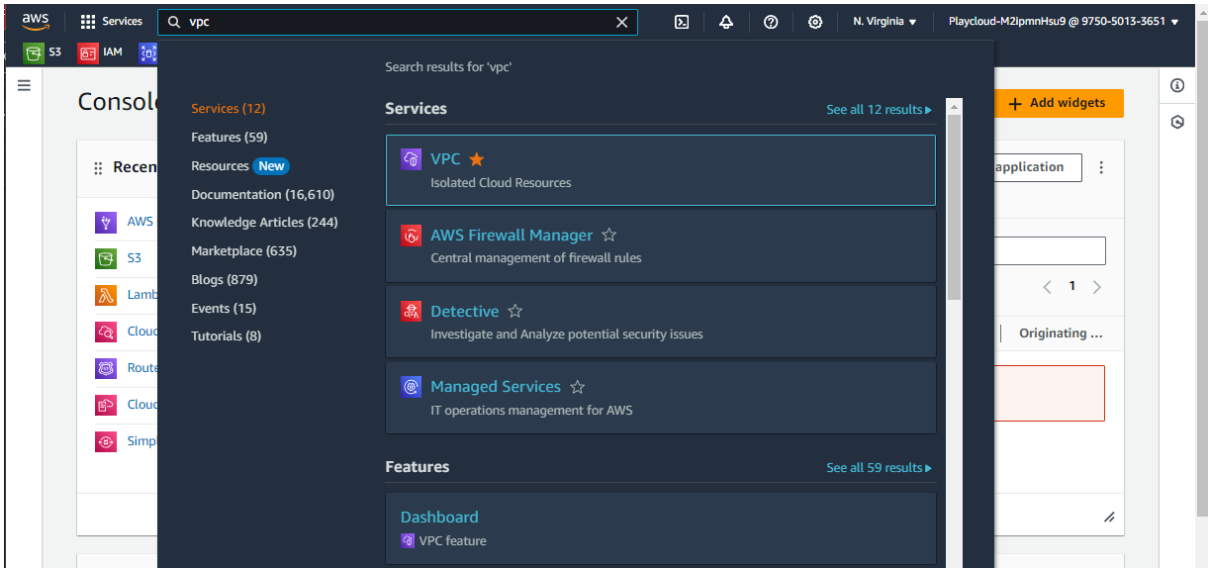
- Understand the structure and components of an Amazon VPC.
- Learn how to create a VPC, including subnets, route tables, and an internet gateway.
- Demonstrate practical skills in isolating network environments within AWS.

[Subscribe to access AWS PlayCloud Labs]

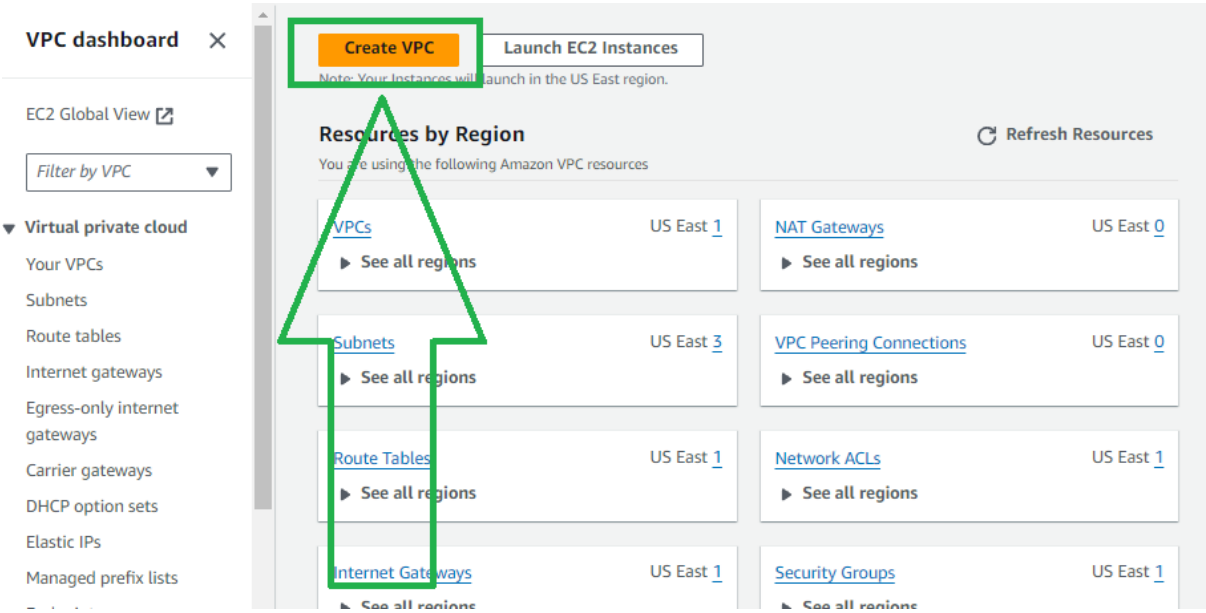
Lab Steps

Creating a Custom VPC

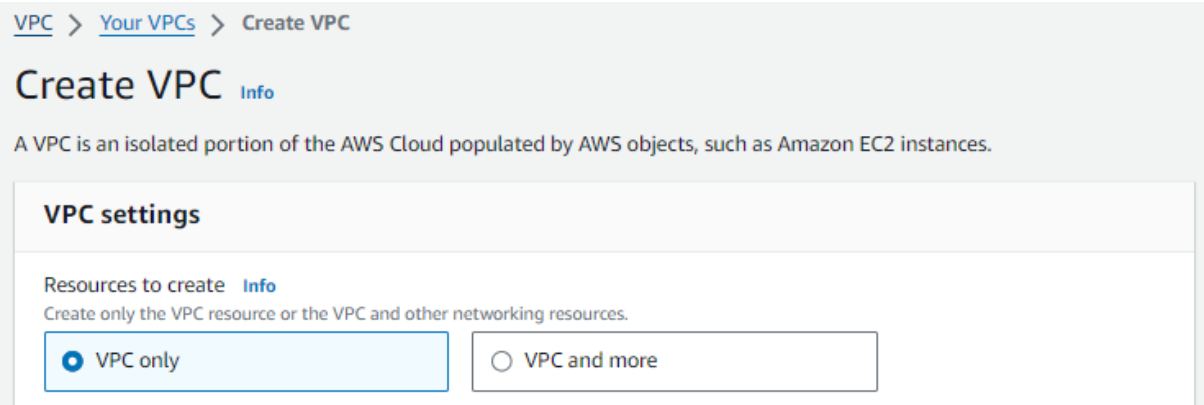
1. Navigate to the VPC dashboard.



2. Click on **"Create VPC"**



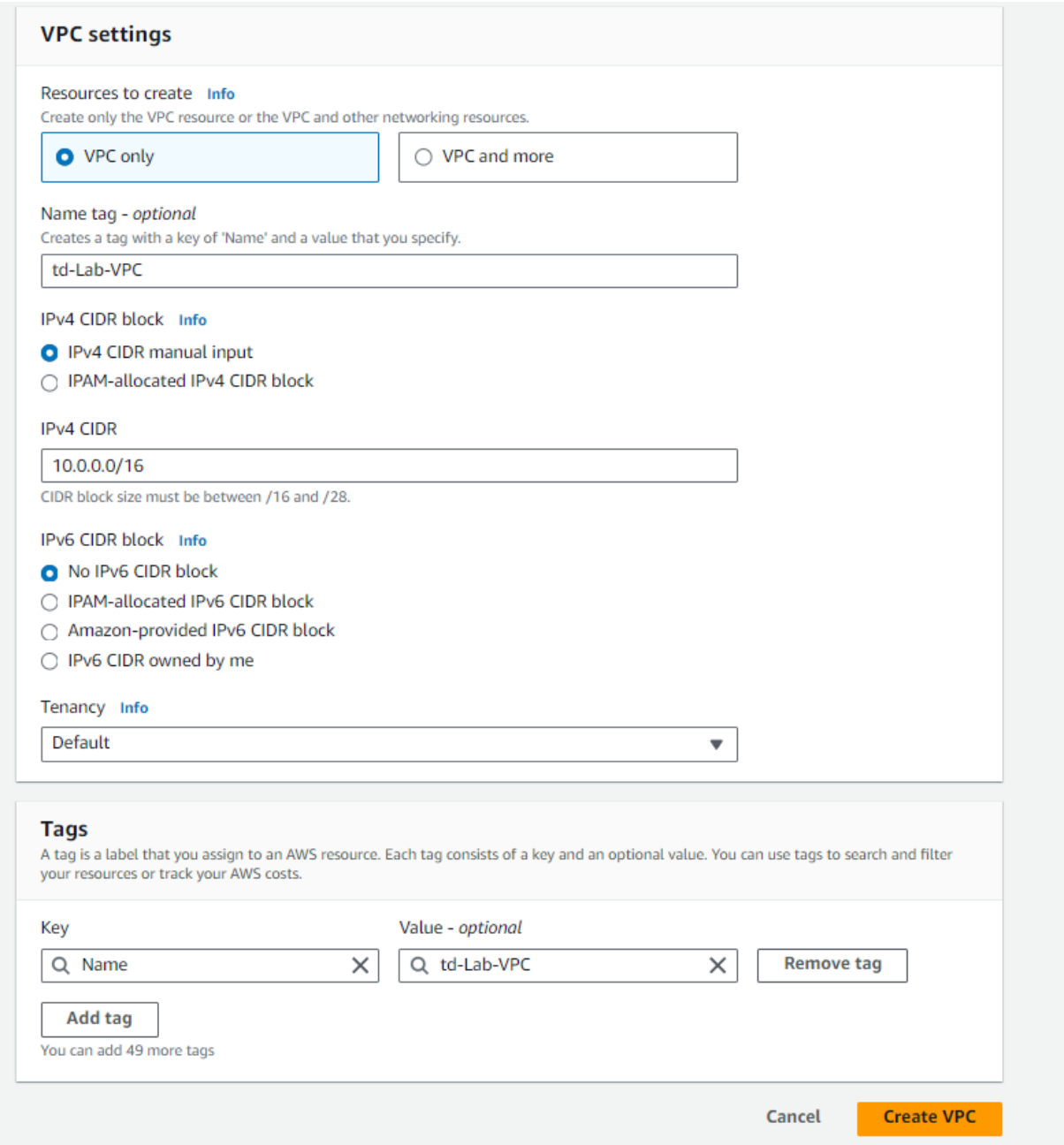
3. Let's select VPC only in the Resources to create to do this lab step by step.



Note: *VPC and more – is an option if you want to create VPC, subnets, Route Table, etc. all at once.*

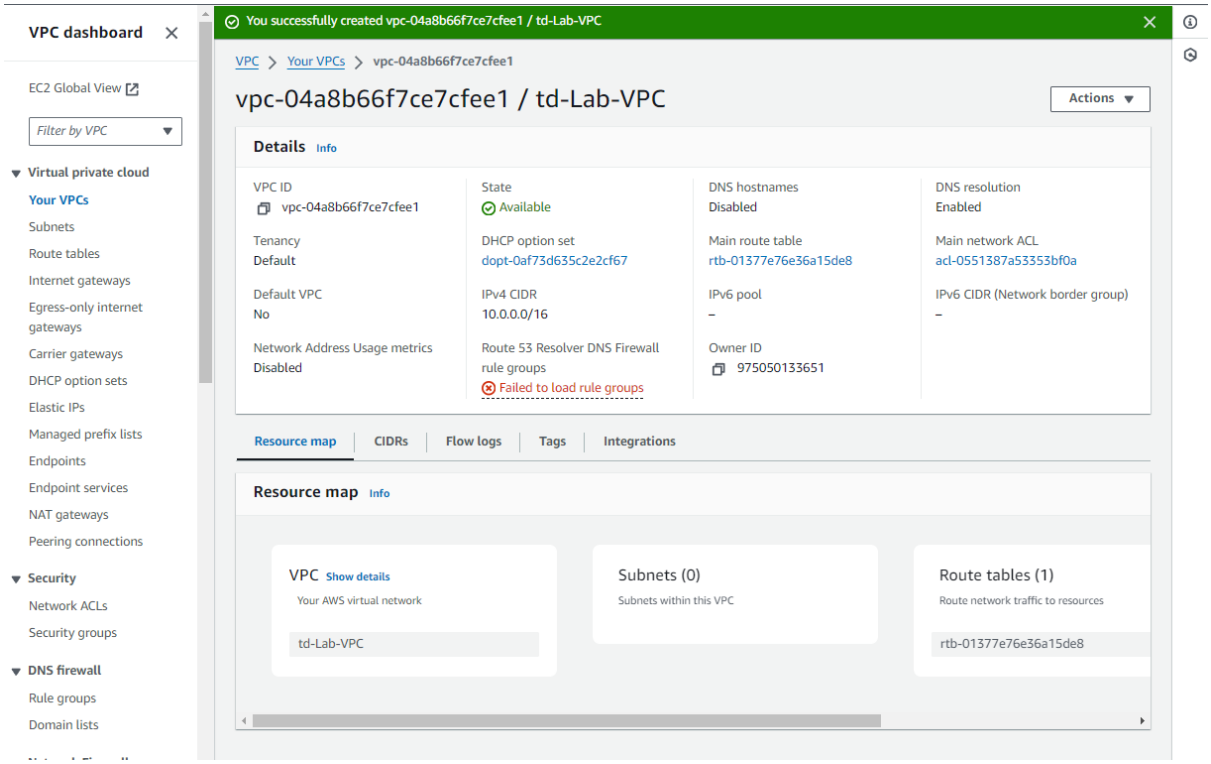
4. Enter the following details:

- Name tag: **td-Lab-VPC**
- IPv4 CIDR block: **10.0.0.0/16**



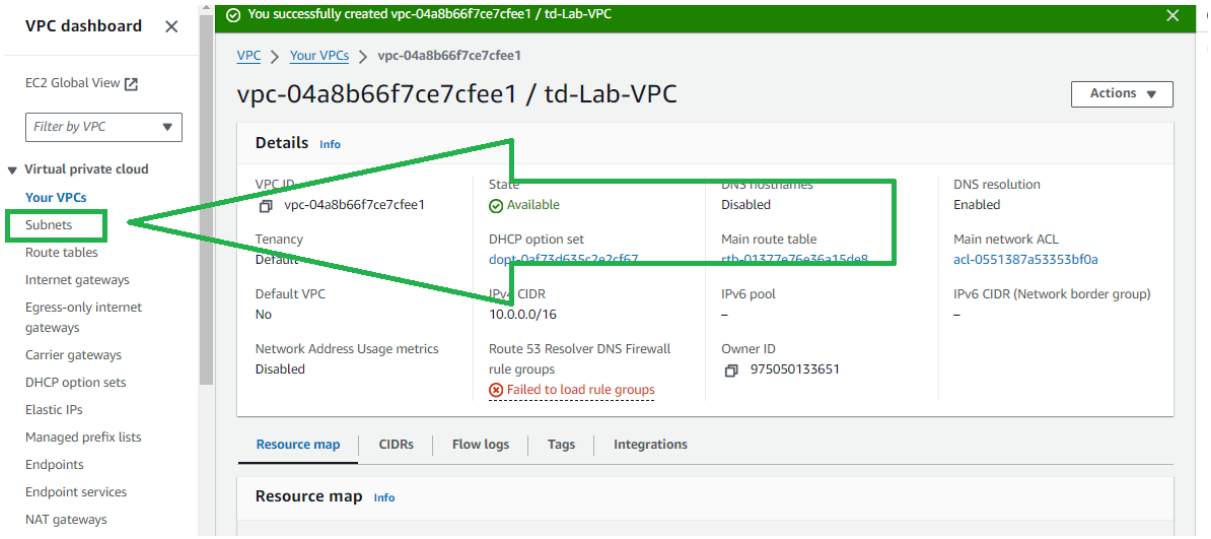
- Click on **"Create."**

5. A new VPC will be created with the specified CIDR block, establishing the network space for this lab.

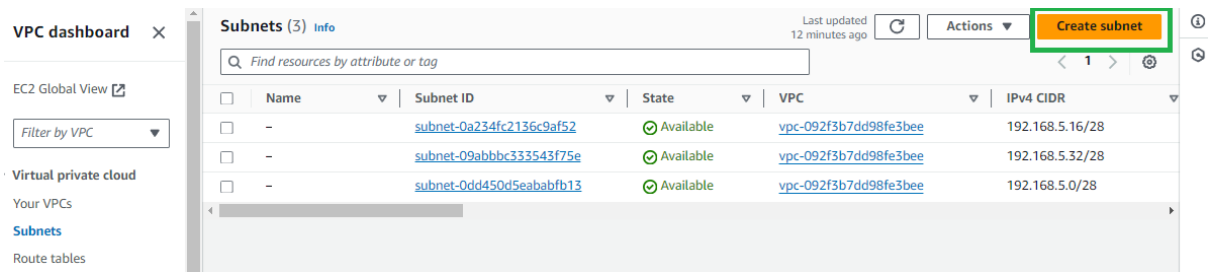


Setting Up Subnets

1. Within the VPC dashboard, select "**Subnets**"



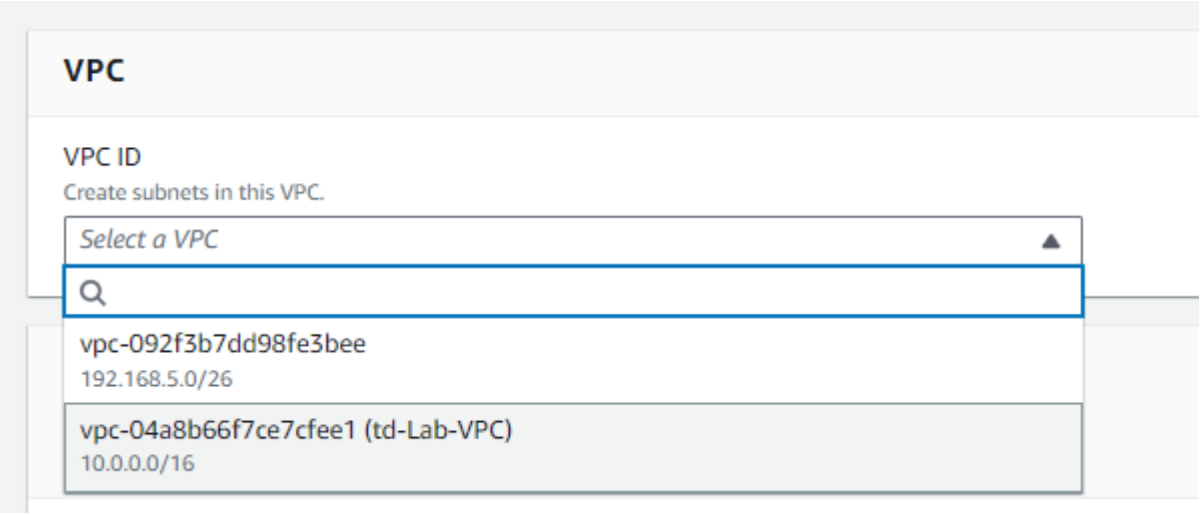
2. Then click on "**Create subnet**."



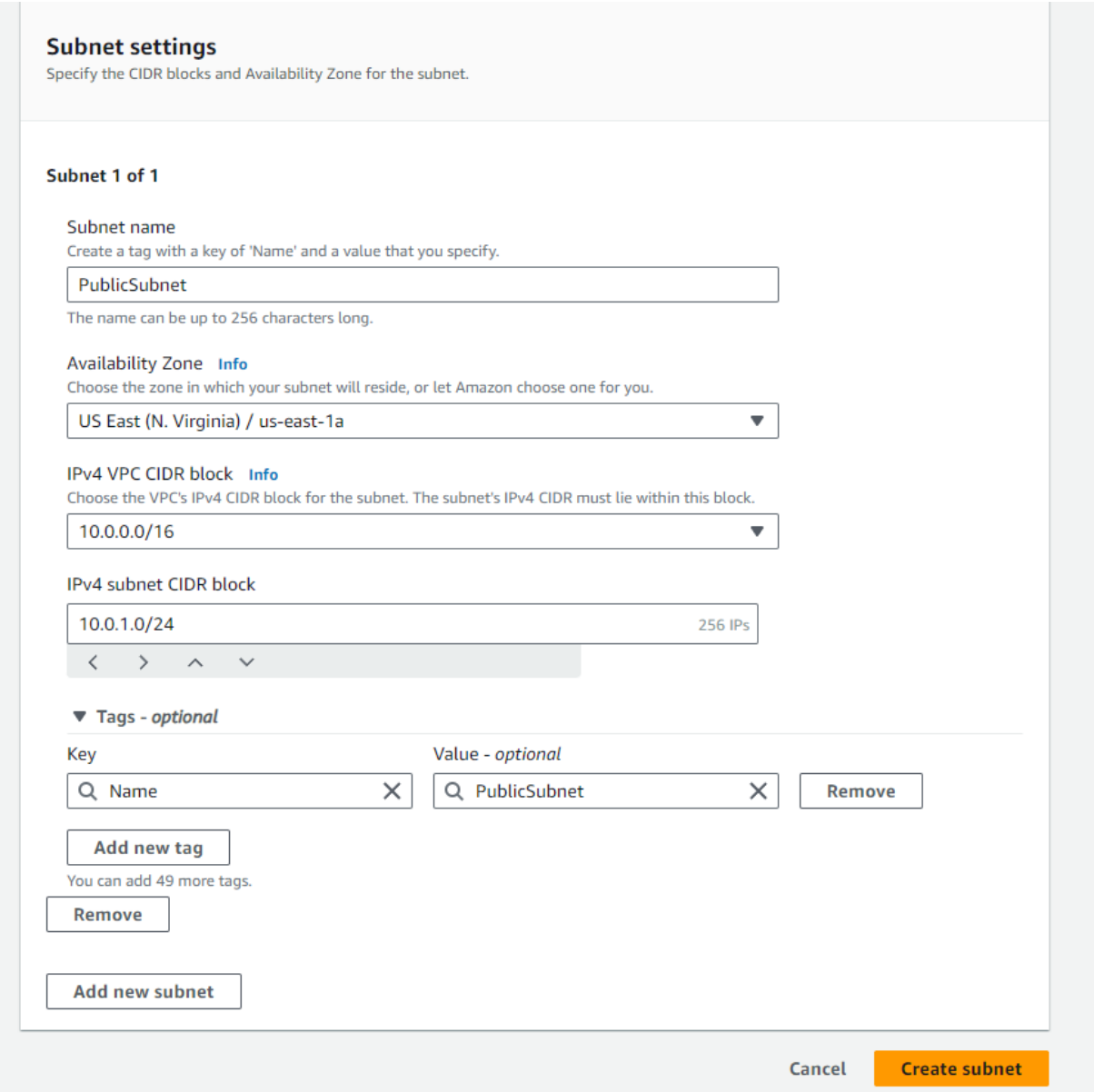
Note: *Notice that there are already default subnets which comes along with the default VPC*

3.Create subnets:

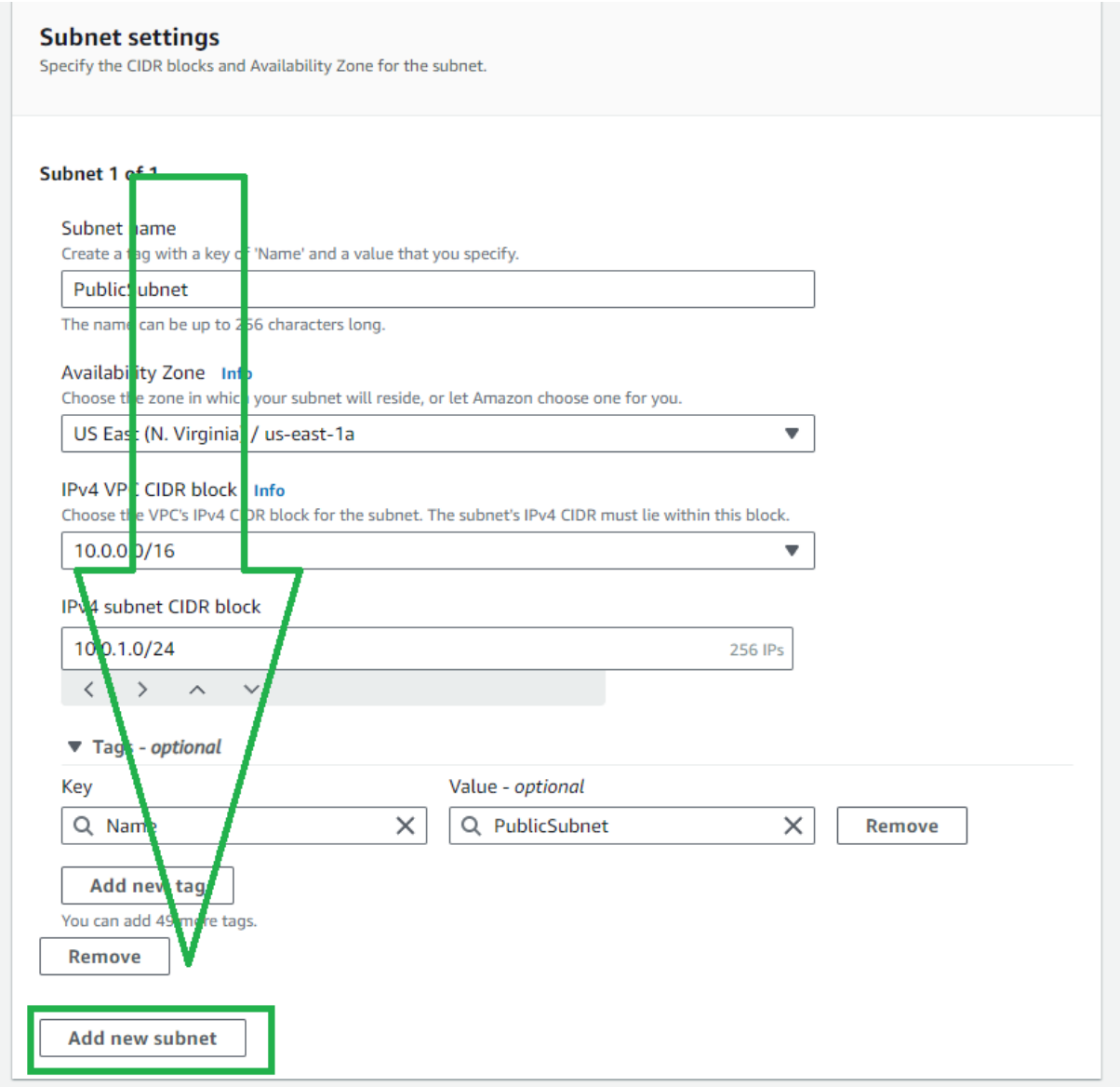
- **VPC ID**: Select **td-Lab-VPC**



- **Name tag**: PublicSubnet-1
- **Availability Zone**: US East (N. Virginia) / us-east-1a
- **IPv4 CIDR block**: 10.0.1.0/24

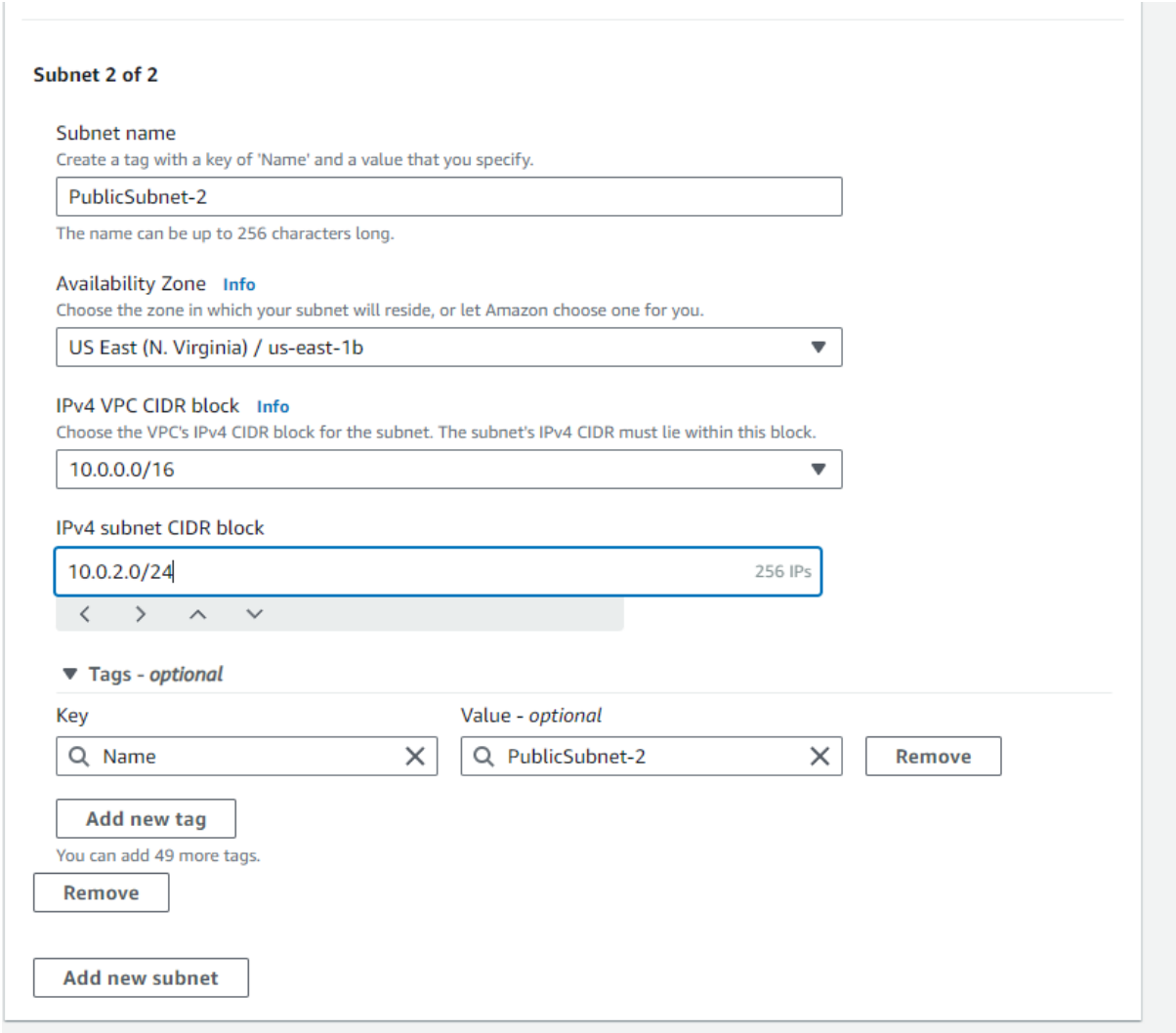


- Add another subnet by clicking the "**Add new subnet**"

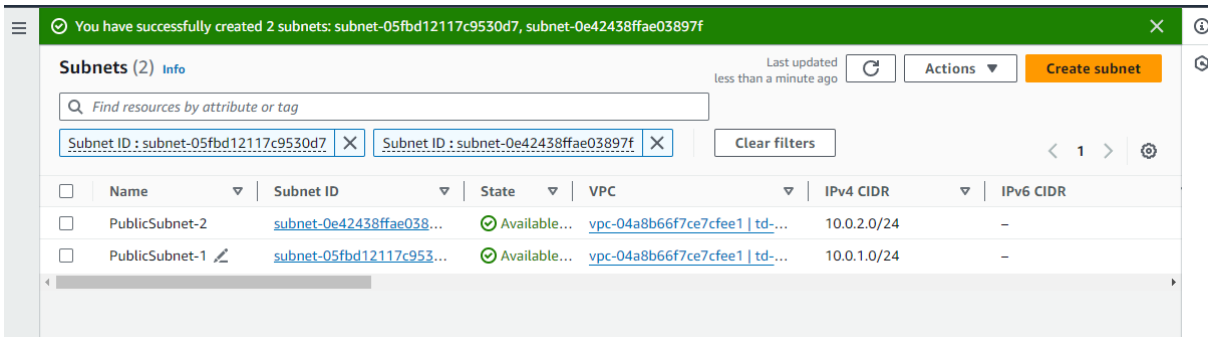


Note: *While you can create multiple subnets within a VPC, remember that the number depends on the VPC's CIDR block size and AWS quotas. Each subnet needs a unique range of IP addresses. By default, AWS allows up to 200 subnets per VPC, but this can be increased by requesting a service quota increase from AWS support. Plan your subnets wisely to accommodate future growth and manage different network environments effectively.*

- Follow the configurations below:
    - **Name tag**: PublicSubnet-2
    - **Availability Zone**: US East (N. Virginia) / us-east-1b
    - **IPv4 CIDR block**: 10.0.2.0/24

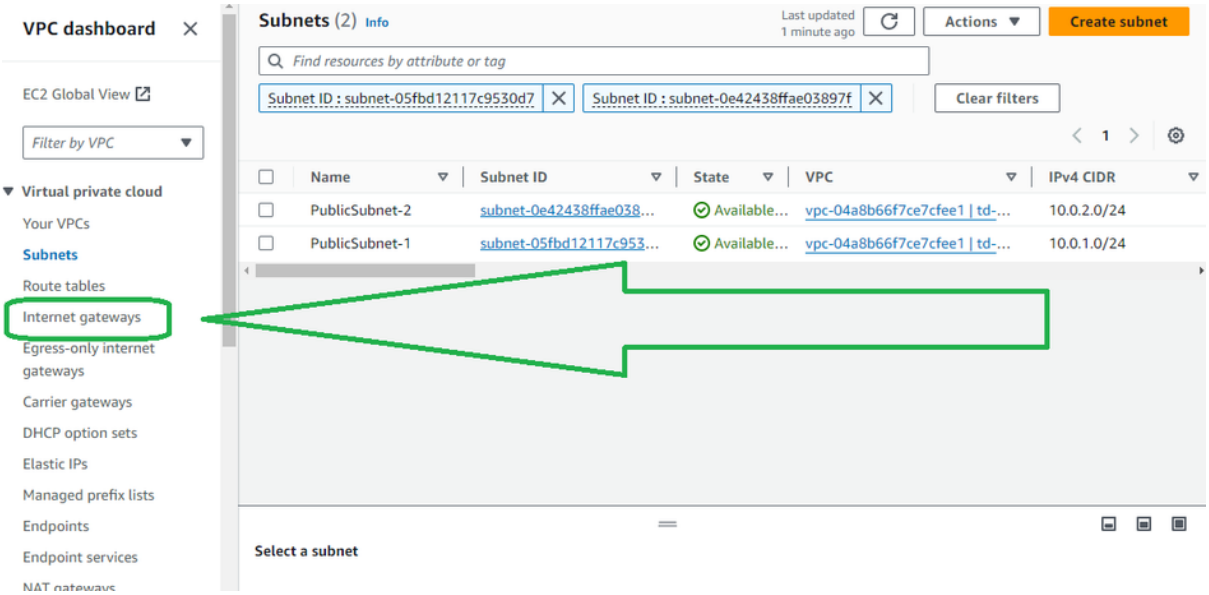


- Click on "**Create.**"



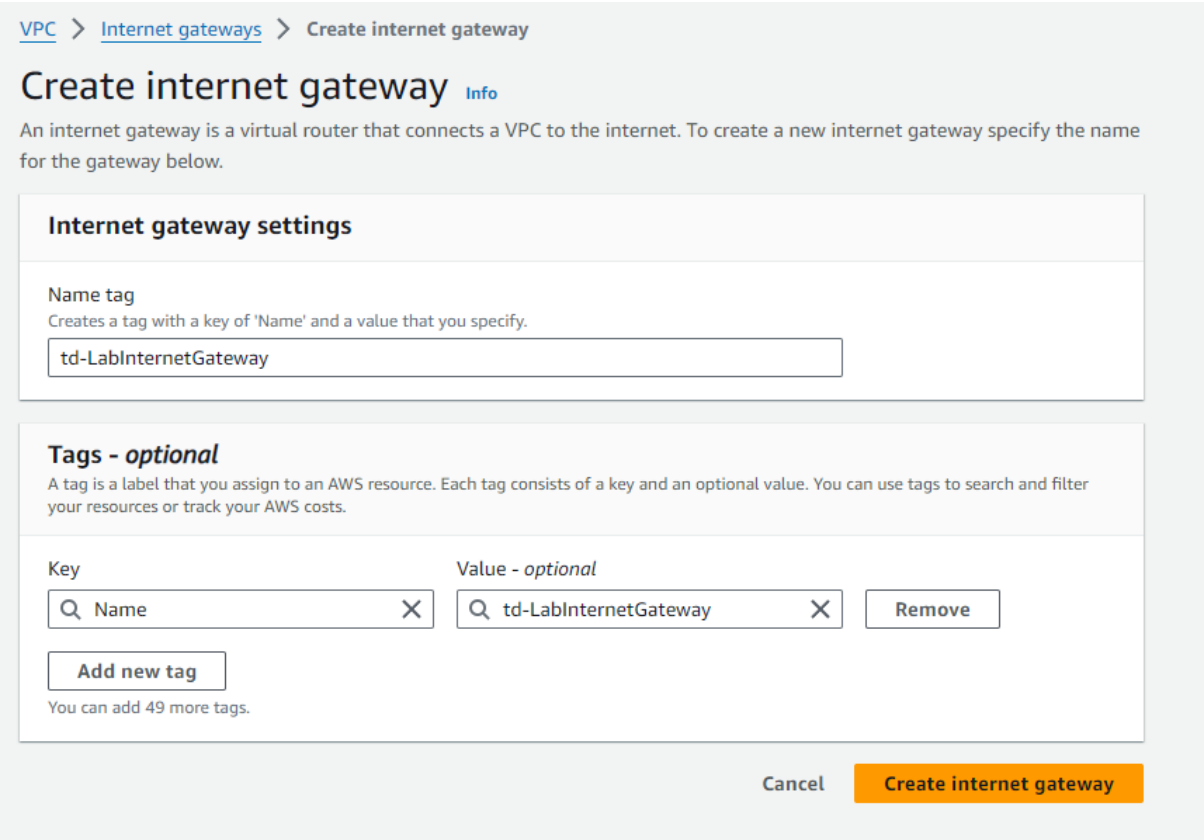
Establishing an Internet Gateway

1. In the left sidebar find and click on "**Internet Gateways**."



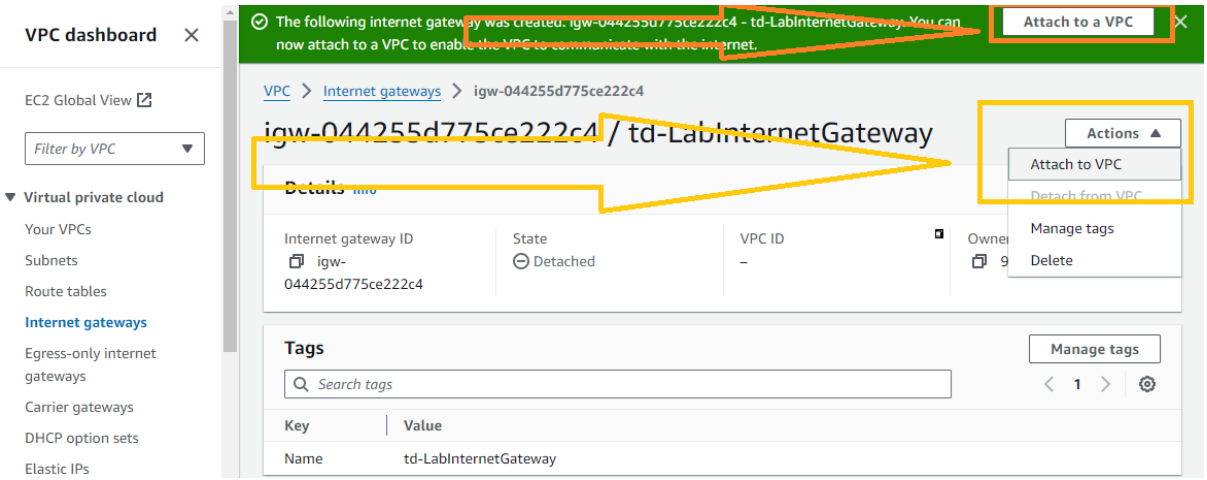
2. Click on "**Create internet gateway**"

3. Name it td-LabInternetGateway

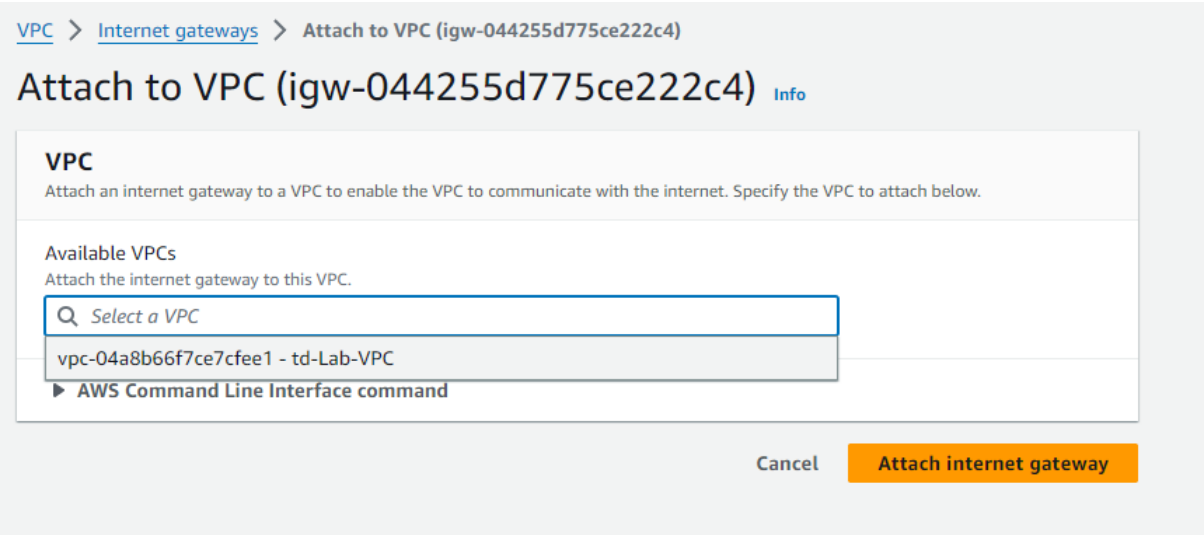


4. Click on **Create internet gateway**

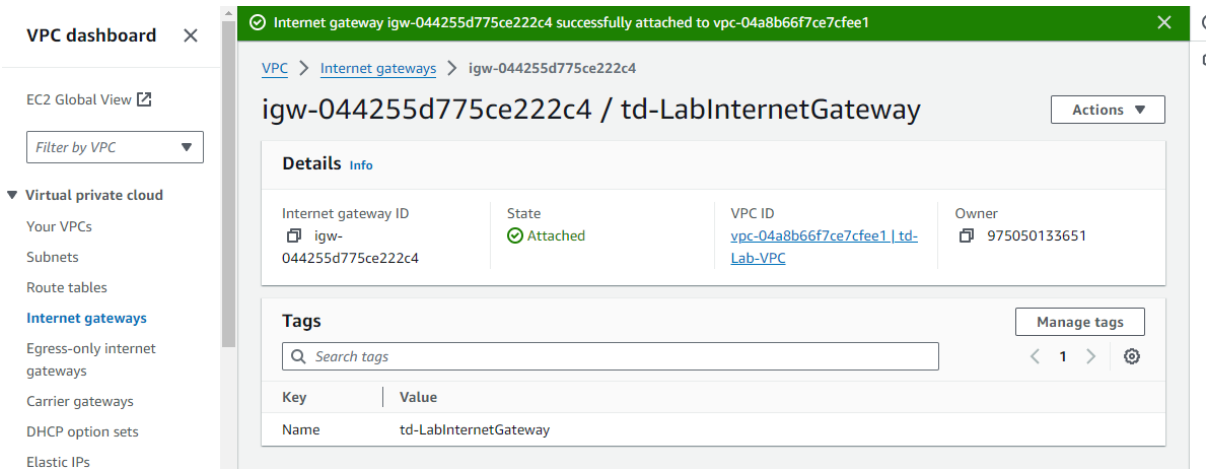
5. Choose the "**Attach to VPC**" in the **Action dropdown** or the notification that pops up after you created the **internet gateway** or in the **Actions** dropdown



6. Attach the internet gateway to td-Lab-VPC

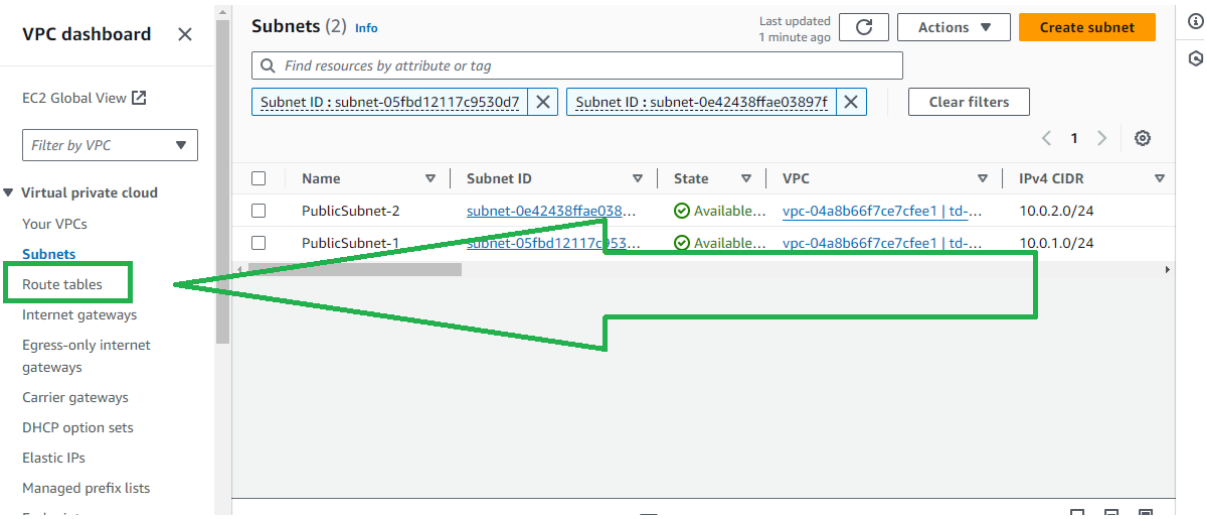


7. Click **Attach internet gateway**

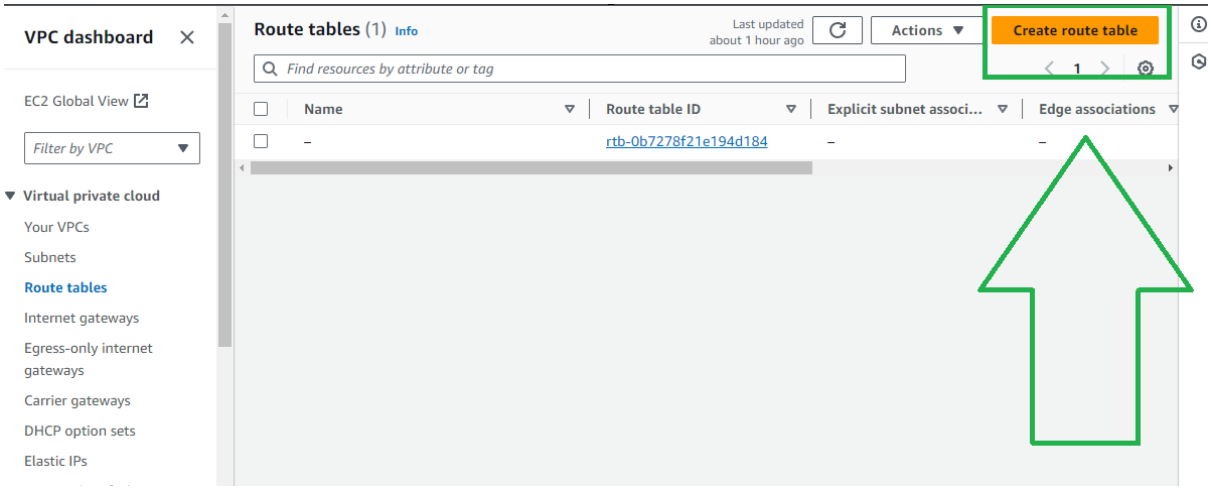


Configuring Route Tables

1. Navigate to "**Route Tables**" in the VPC dashboard

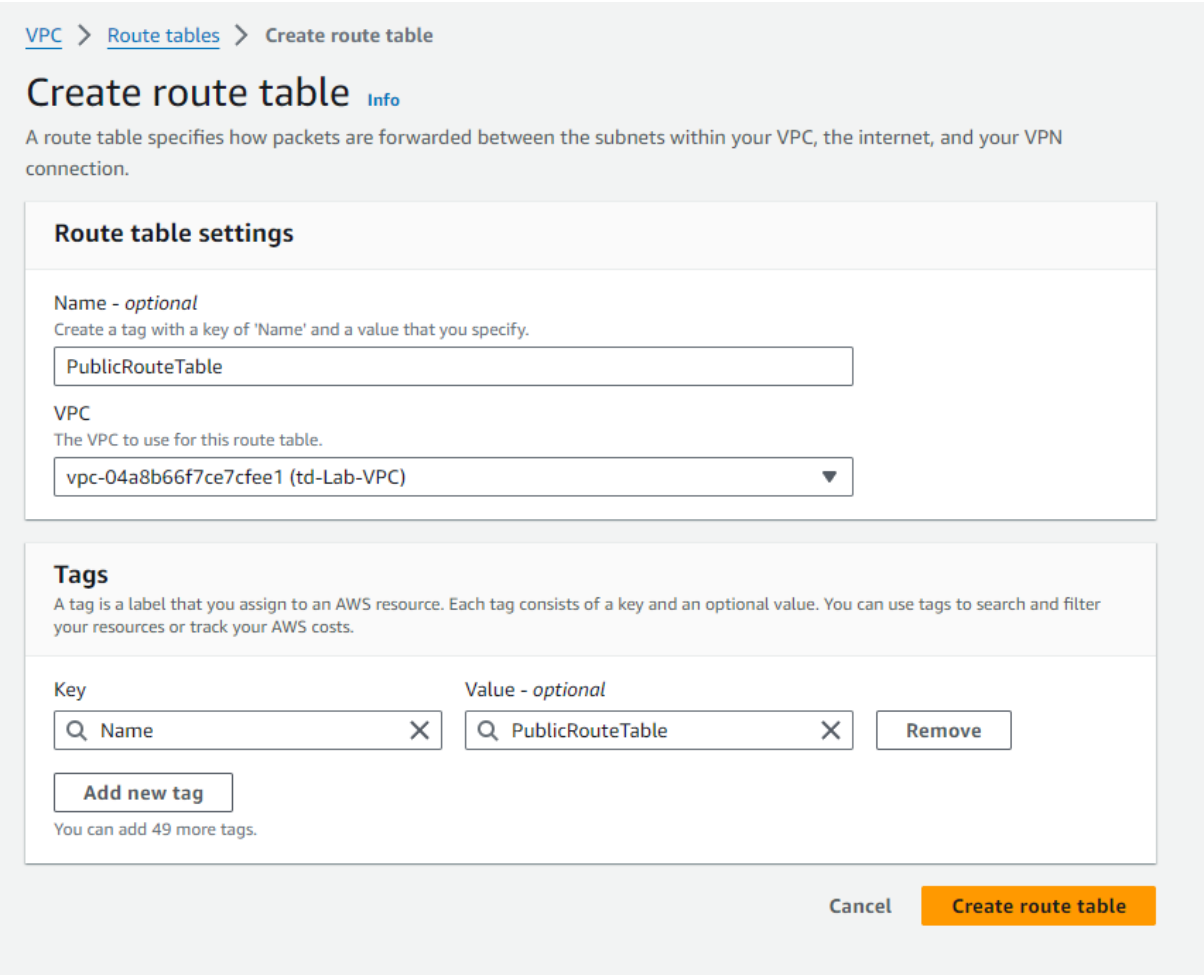


2. Click "**Create route table**."

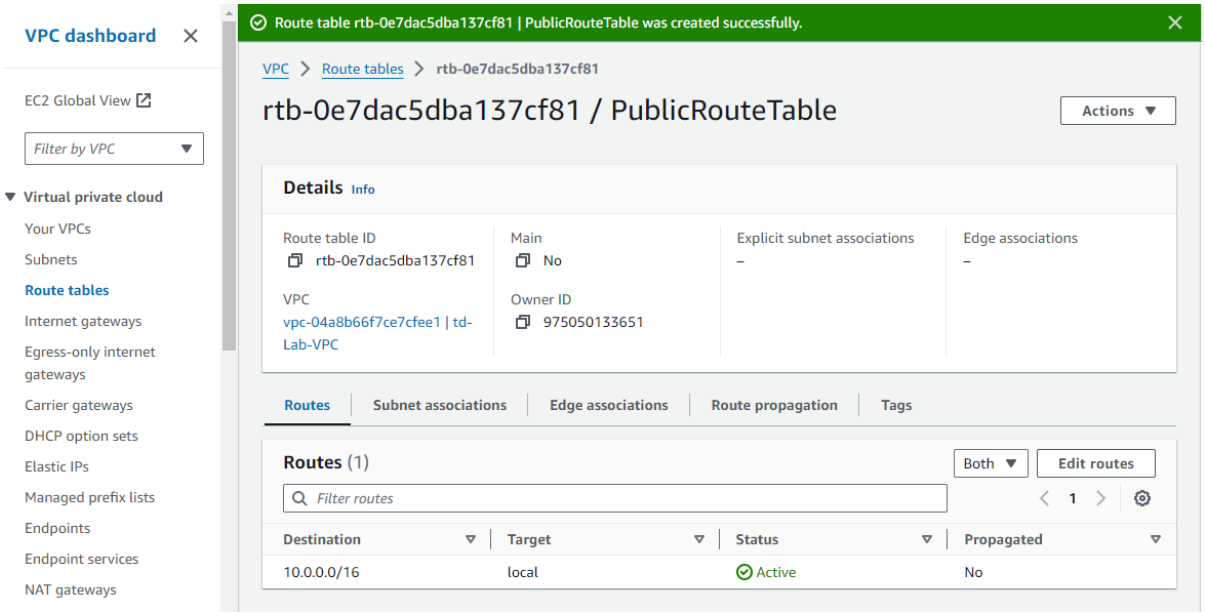


Note: *Notice that after creating a VPC, AWS automatically create route table for that VPC for you. You can either use this or reconfigure it. But for this lab we will manually create our own route table and its configurations.*

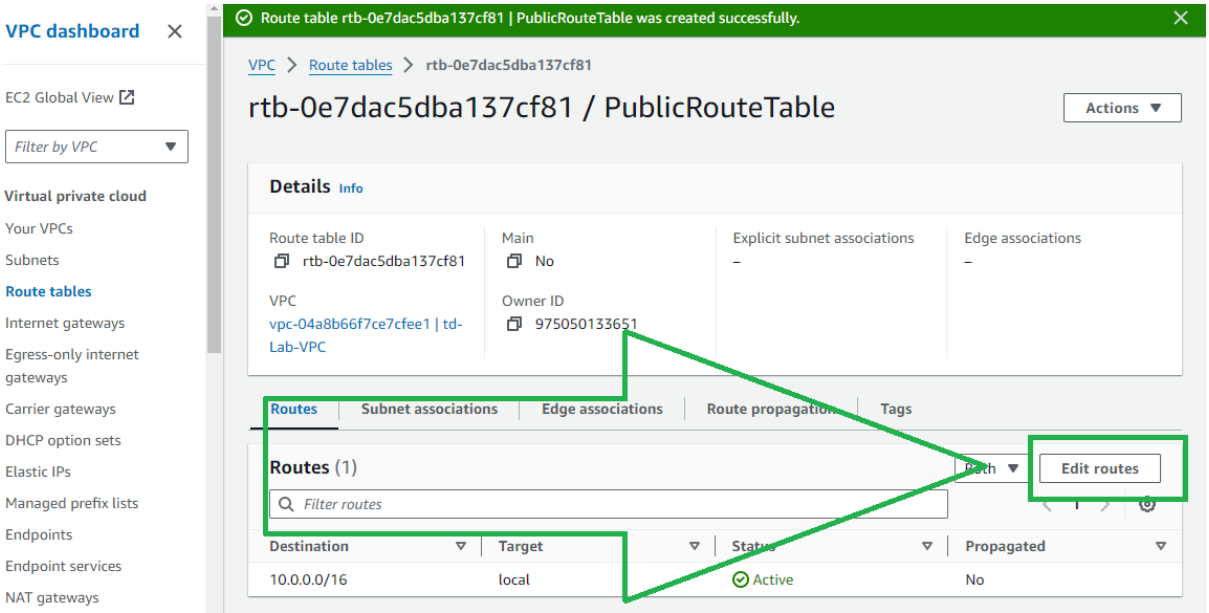
3. Name it PublicRouteTable and associate it with td-Lab-VPC.



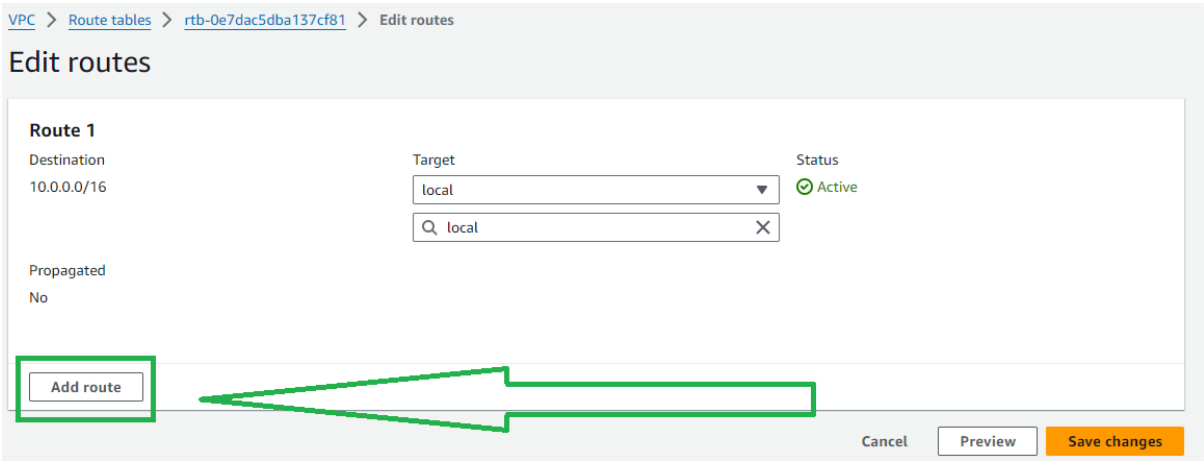
4. Click "**Create route table**"



5. Click on "**Edit routes**"

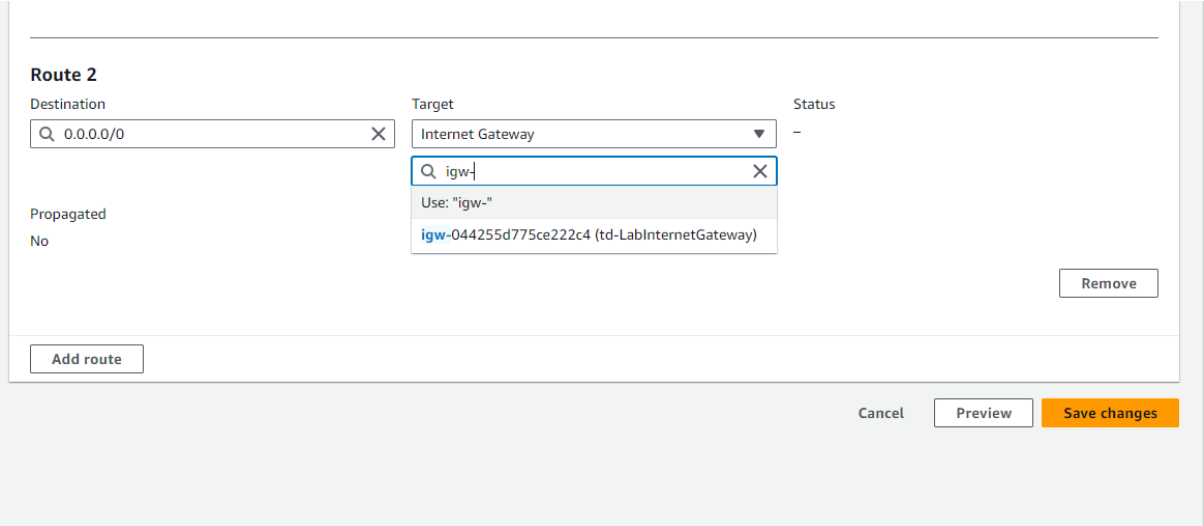


6. Click on "**Add route**"



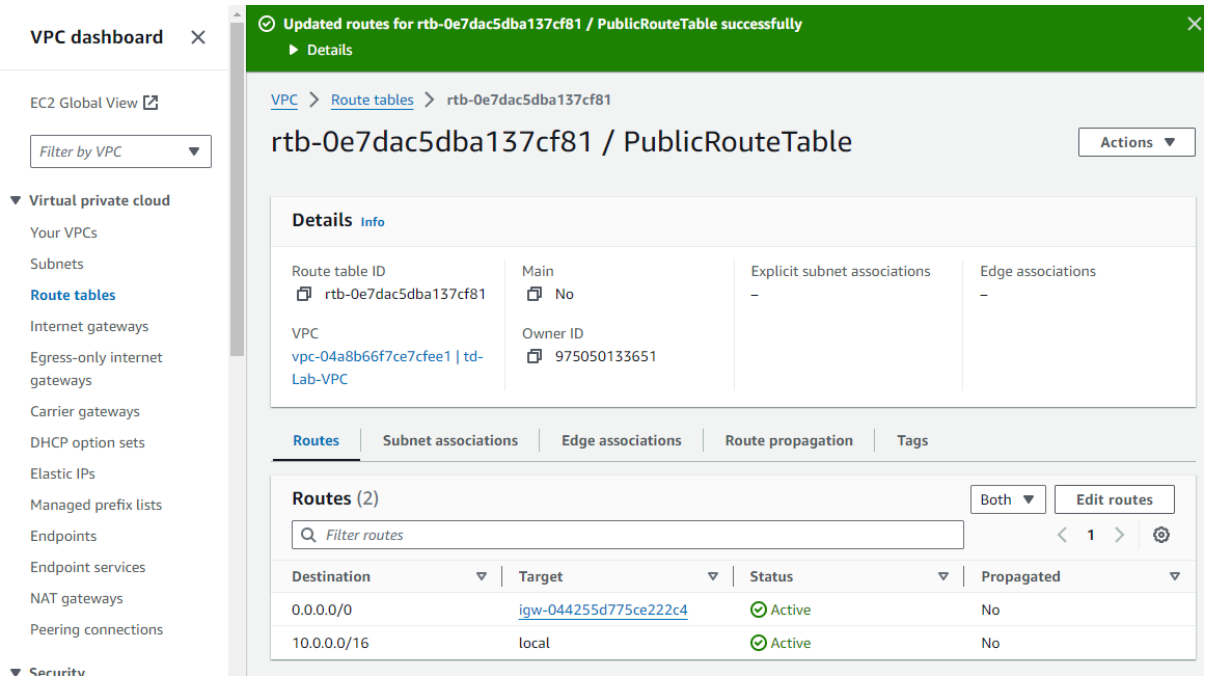
7.Add the following:

- **Destination**: 0.0.0.0/0
- **Target**: Select **Internet Gateway** and the name of the internet gateway that you created.



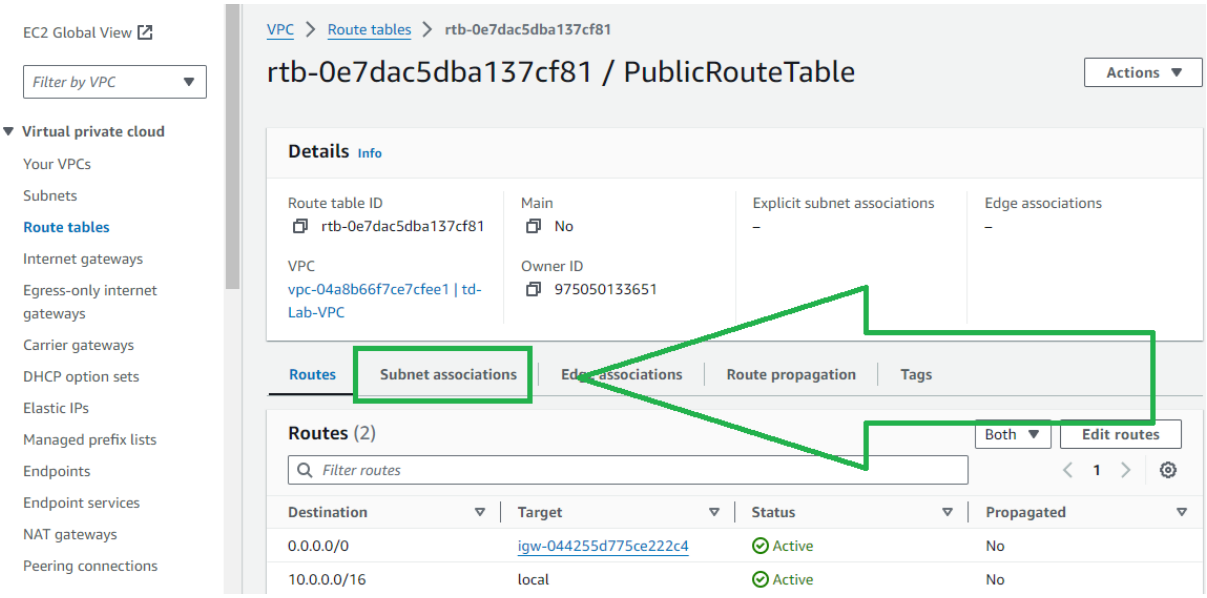
By adding this route, you ensure that any internet-bound traffic (any IP address represented by 0.0.0.0/0) to pass through an internet gateway from your VPC is correctly routed through the internet gateway, thus enabling external connectivity. This setting is particularly important for public subnets that host web servers or other resources needing to communicate with the internet.

8. Click **Save changes**

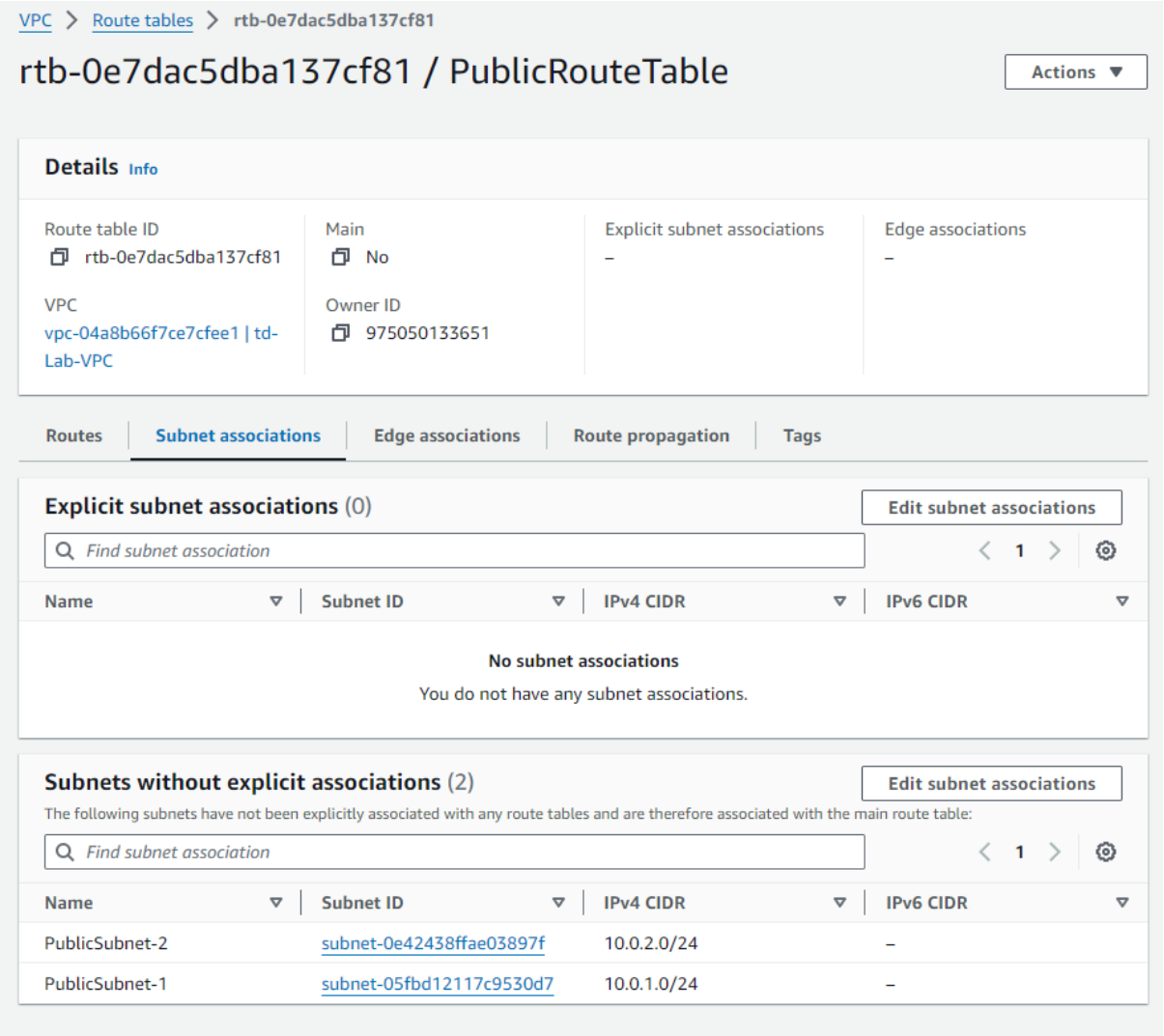


9. Associate PublicRouteTable with PublicSubnet-1 and PublicSubnet-2

- Go to the **Subnet associations tab**



- In this tab you will see:



CONCEPT:

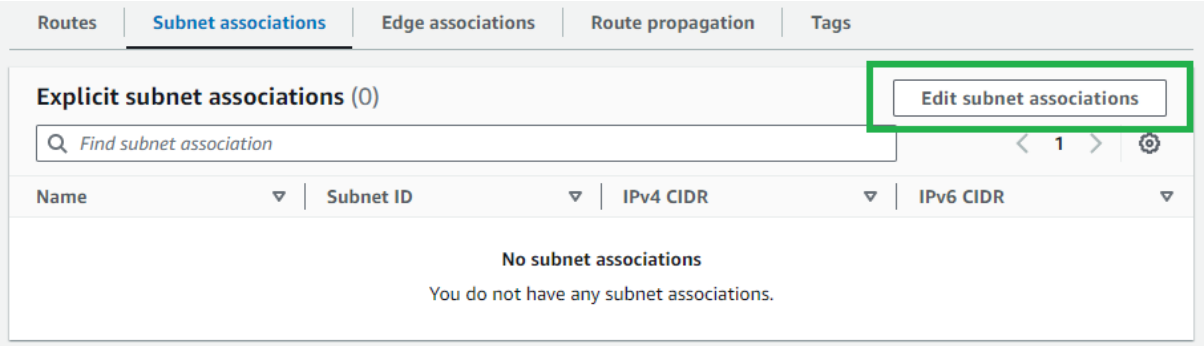
Explicit Subnet Associations
This occurs when you manually associate a subnet with a specific route table. By doing this, you ensure that all the traffic from that subnet is directed according to the routes defined in the associated route table. This is typically used to enforce specific routing policies for different parts of your network, such as distinguishing between public (internet-facing) and private (internal-only) subnets.

- **Example Usage**: You might explicitly associate a subnet with a route table that has a route to an internet gateway for public subnets, ensuring instances within these subnets can access the internet.

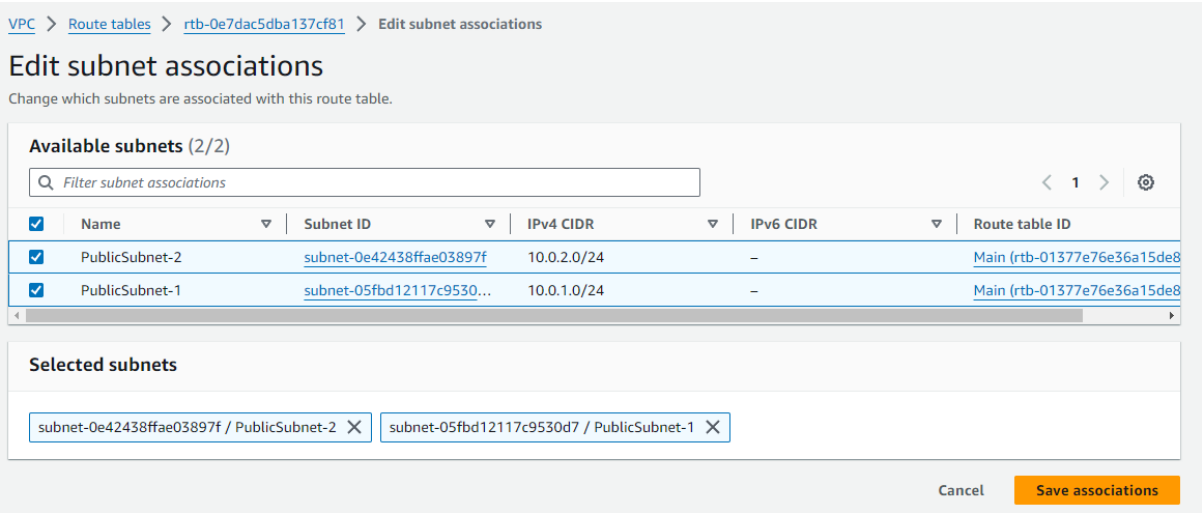
Subnets Without Explicit Associations
Any subnets in a VPC that do not have an explicit route table association will automatically be associated with the VPC's main route table. The main route table is the default route table that AWS creates with each VPC. This setup ensures that every subnet has at least basic routing capabilities, even if no custom routing has been configured.

- **Example Usage**: Private subnets that do not require direct access to the internet can be left with the default route table, which typically does not contain a route to the internet gateway. This helps in maintaining the security posture by limiting external access.

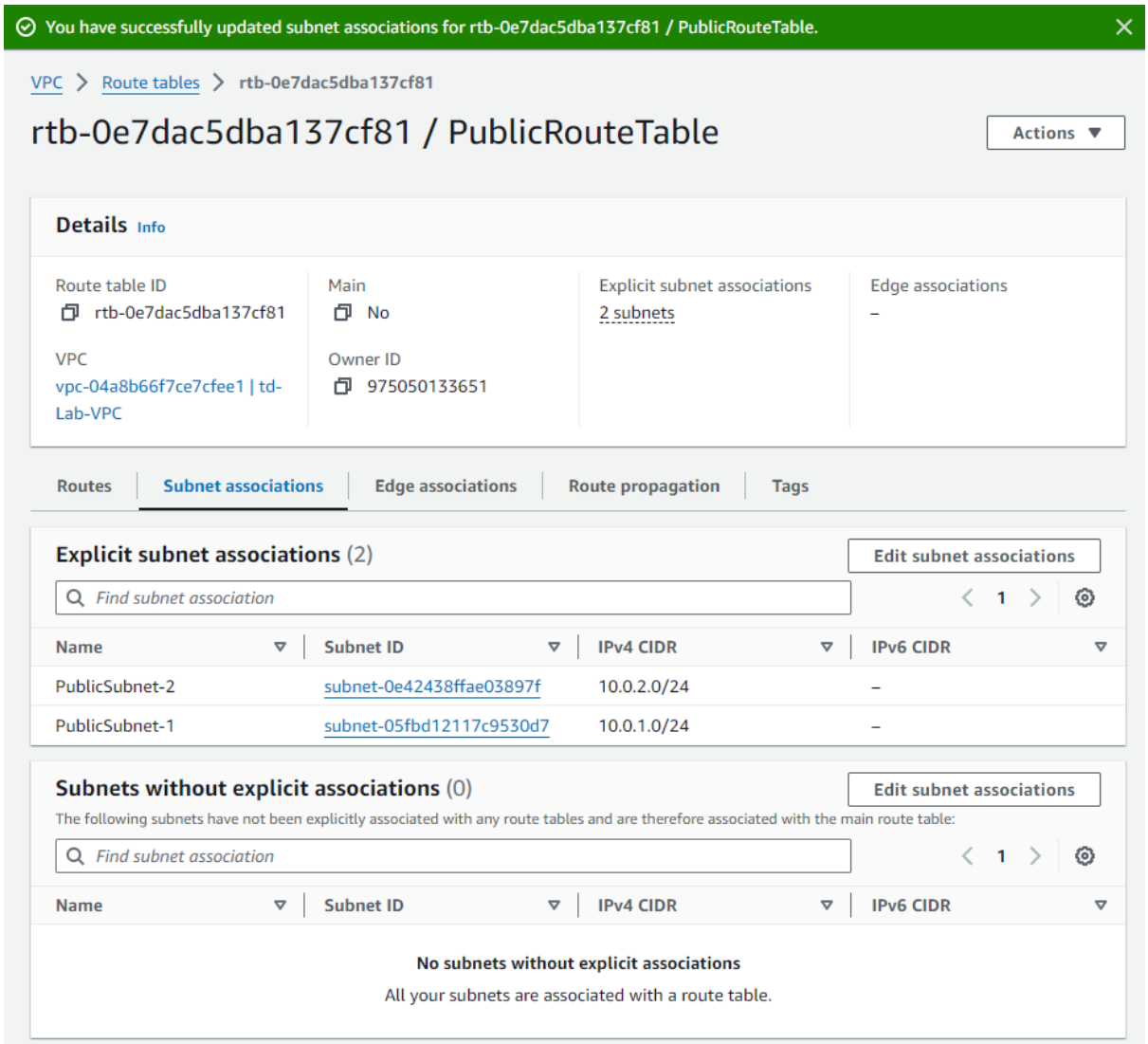
- Click on "**Edit subnet associations**" inside the **Explicit Subnet Associations**



- Select both PublicSubnet-1 and PublicSubnet-2 and click on **Save associations**

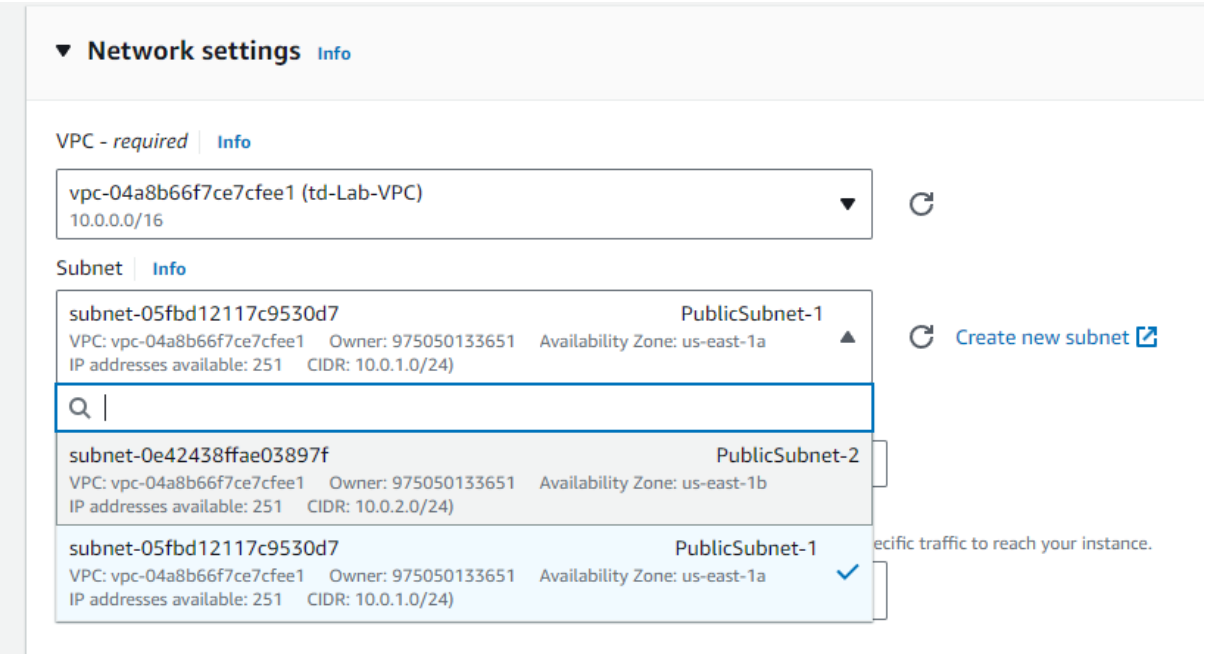


- This will be the output:



Testing Connectivity

1. Navigate to the EC2 Launch and launch an EC2 instance within PublicSubnet-1 or PublicSubnet-2.

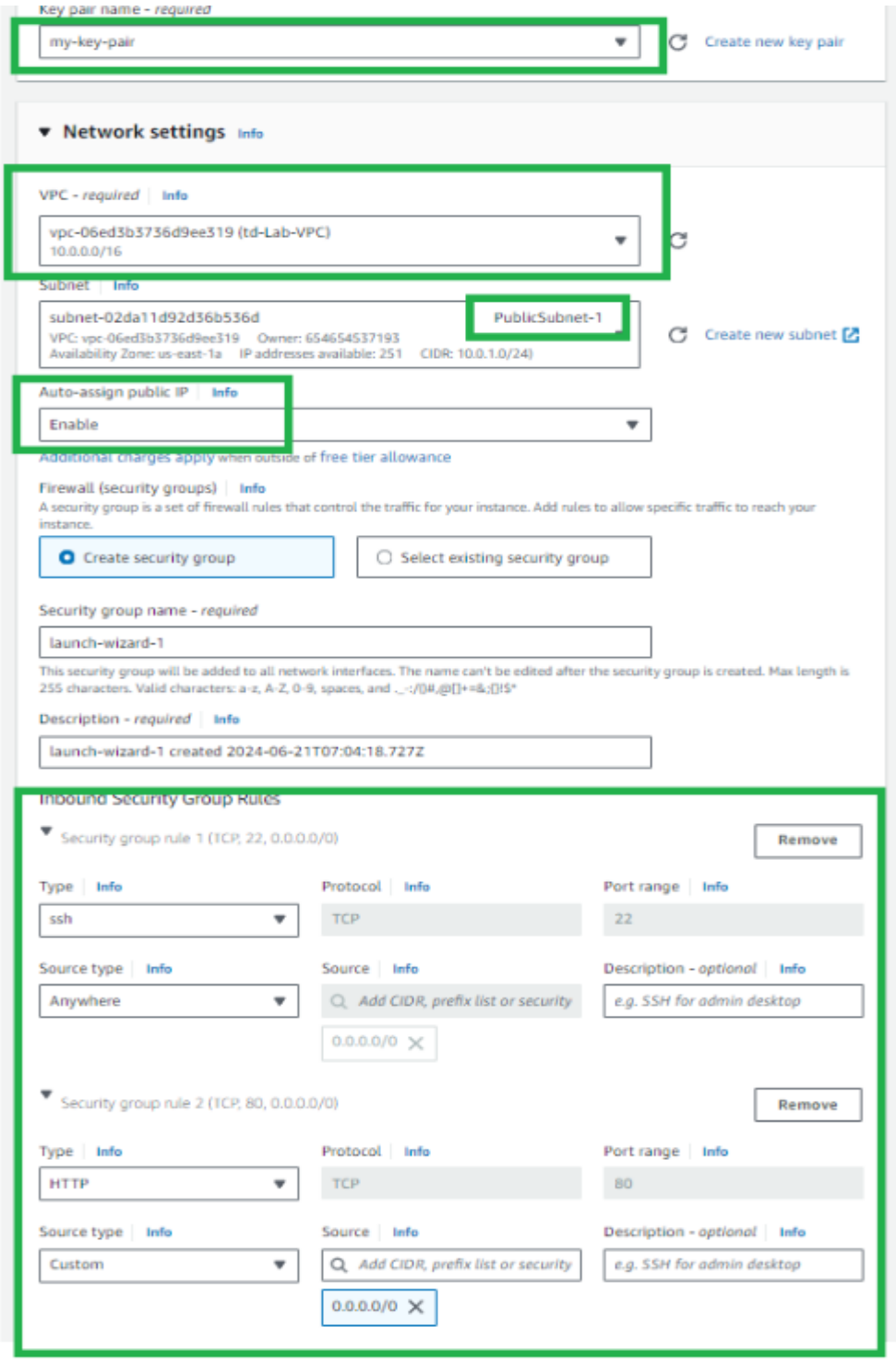


2. During setup:

- Ensure the instance's security group allows inbound ssh access (e.g., allow all inbound traffic anywhere or simply your IP for security) and **HTTP** connection using source **0.0.0.0/0**
  Note: *Usually, you need to make ssh connection only to the trusted IPs but for this testing and simplicity we can use* ***anywhere*** *which is* ***0.0.0.0/0***

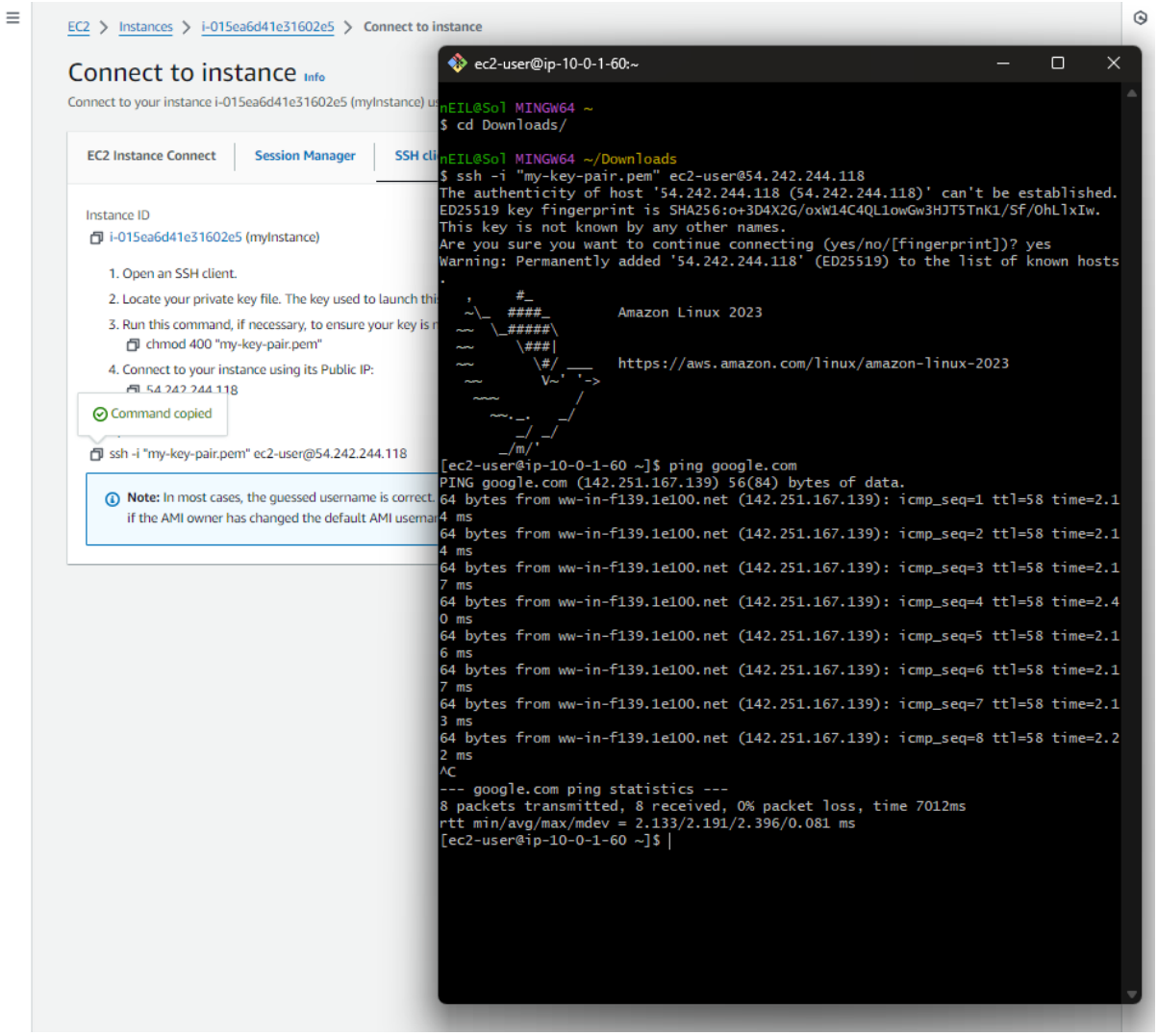
- Ensure that you **enable** Auto-assign public IP

- Created an SSH key.



- Launch your instance.

3. Once the instance is running, connect via SSH attempt to ping a public internet address to verify connectivity.



That’s it. Congratulations! You’ve successfully set up a custom Amazon VPC, configured its subnets, established an internet gateway, and managed route table associations. This practical experience has enhanced your understanding of AWS network isolation and configuration. You’ve seen firsthand how to manage traffic flow within a VPC, ensuring secure and efficient network operations. This foundation will aid in further exploration of AWS networking features, enabling you to build more robust and scalable cloud solutions. Happy learning!