**Lesson 6 Demo 1**

**Create Custom VPC (Non-Default VPC)**

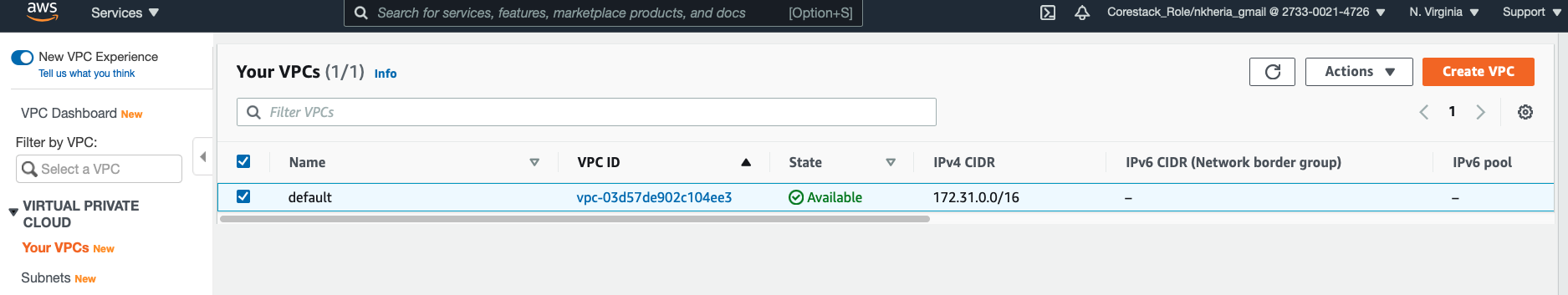


Steps to be followed:

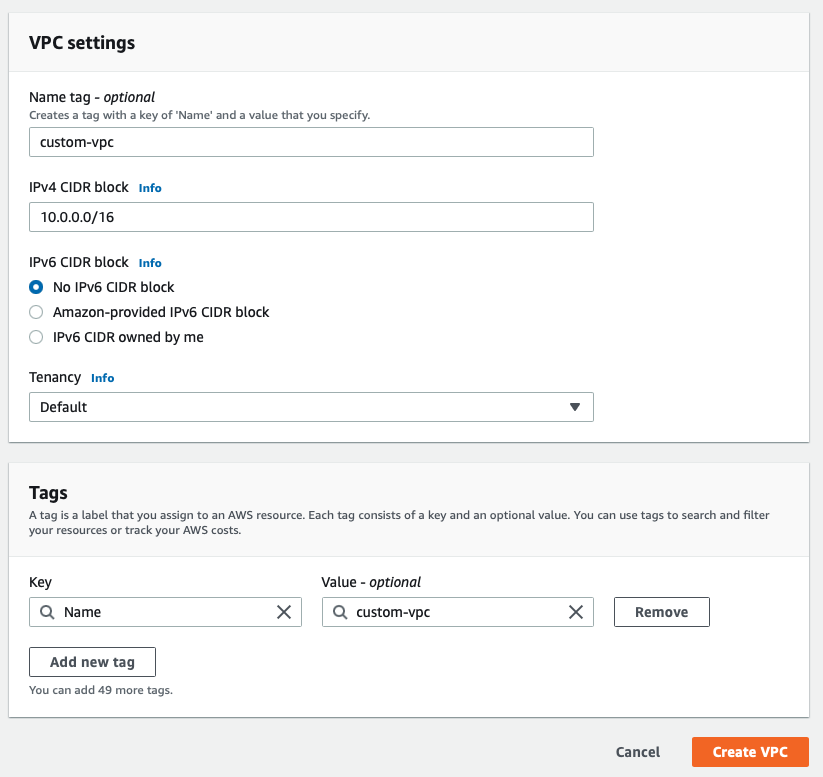
1. To create a VPC
2. Create a Subnet
3. Attach the internet gateway to your VPC
4. Create a route table and attach it your VPC
5. Associate the subnets with the route table
6. Create a network access control list
7. Create EC2 instances and attach the VPC

**Step 1: To create a VPC**

* 1. Under the AWS Management Console, search for and select VPC
  2. Select **Your VPC** in the VPC dashboard
  3. Click on Create VPC



* 1. Enter Name tag as **custom -vpc** and **10.0.0.0/16** as IPv4 CIDR Block
  2. Click on Yes, Create



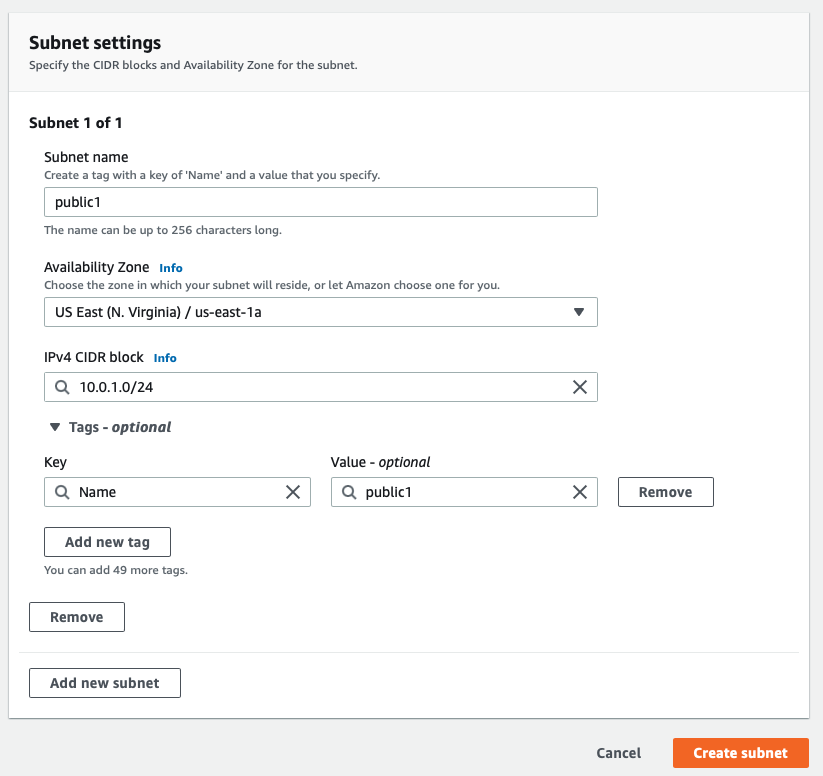
|  |
| --- |
| Note: The maximum range of CIDR block is /16 so that you get (2^16)-2 IP addresses for your VPC  For example, /20 will give you (2^(32-20))-2, i.e., (2^12)-2 addresses |

## **Target to Achieve**

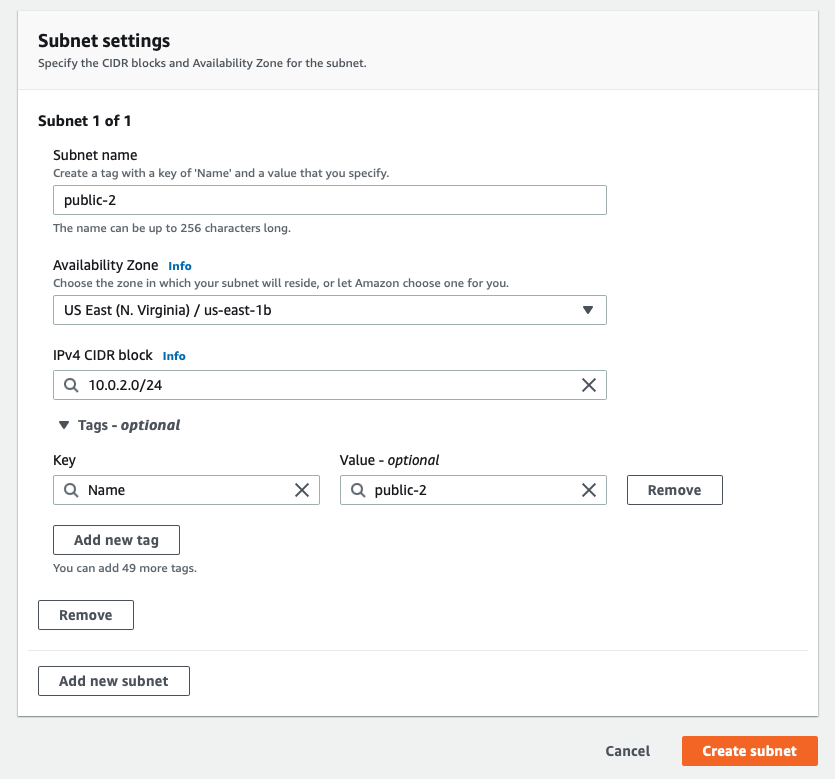
1. We will create two private and two public subnets in two Availability Zones
2. Public subnets will have instances that host web servers, and a user-facing application. Private subnet mostly contains instances that host app server, database, etc.

**Step 2: Create a Subnet**

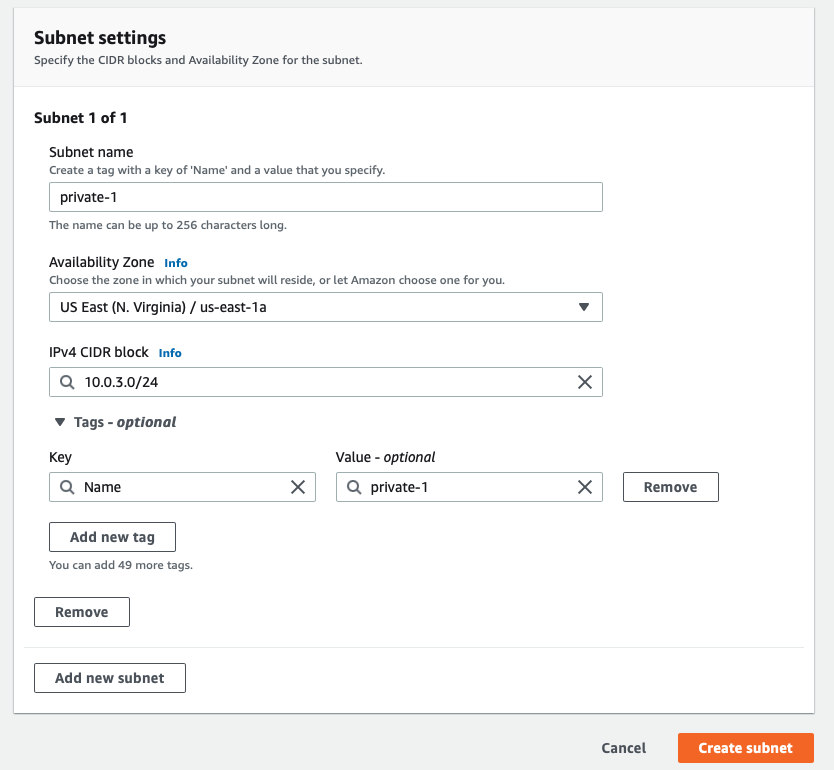
* 1. In your VPC dashboard, click on **subnets**
  2. /28 is the minimum value for any subnet
  3. Give the name for the first public subnet as **public1**
  4. Select the Availability Zone as **us-east-1a** Ipv4 CIDR block **10.0.1.0/24**



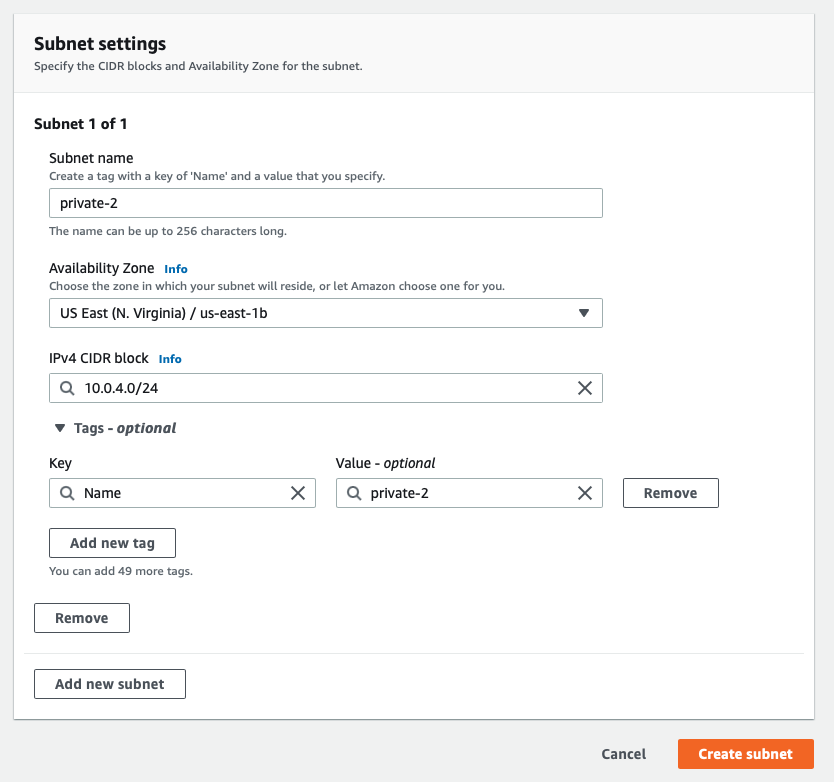
* 1. Let us create a public subnet 2 (public2) under the custom-vpc
  2. In the other Availability Zone, select the availability zone as **us-east-1b** andIPv4 CIDR block as **10.0.2.0/24**
  3. Click on Yes, Create



2.9 Similarly create private subnet 1 (private1) under custom-vpc with CIDR block **10.0.3.0/24** under availability zone **us-east-1a**

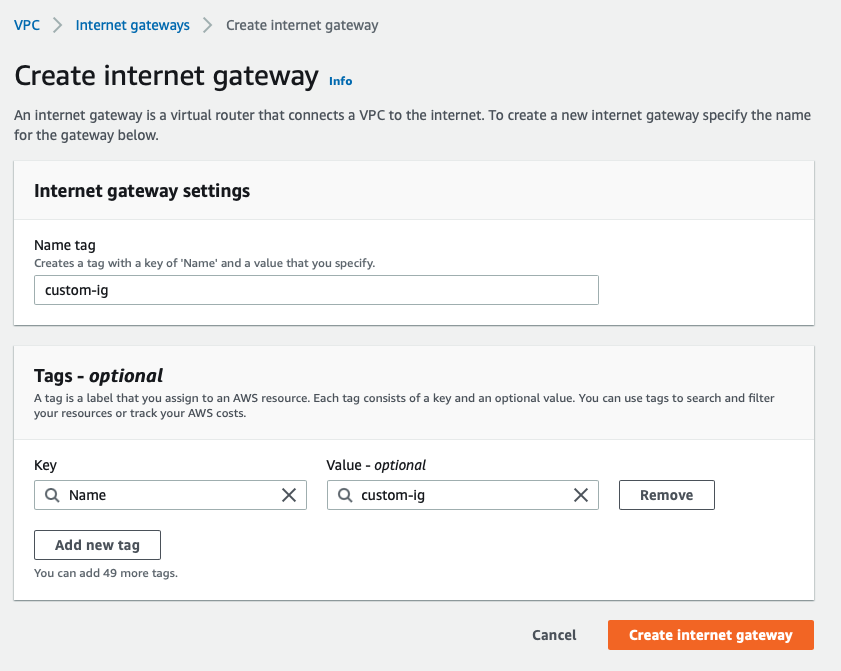


2.10 Create one more private subnet(private-2) under custom-vpc with CIDR block **10.0.3.0/24** under **us-east-1b**

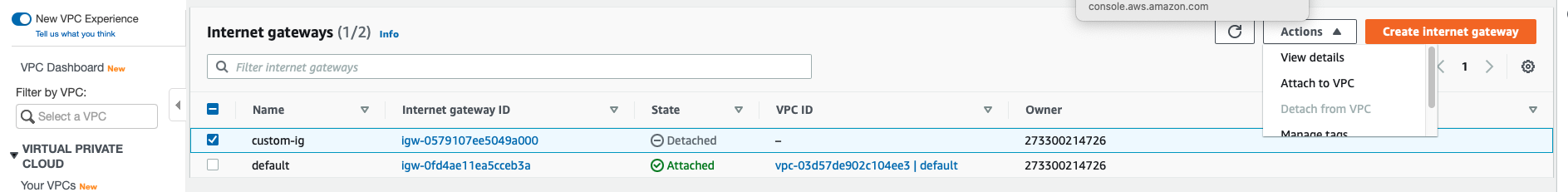


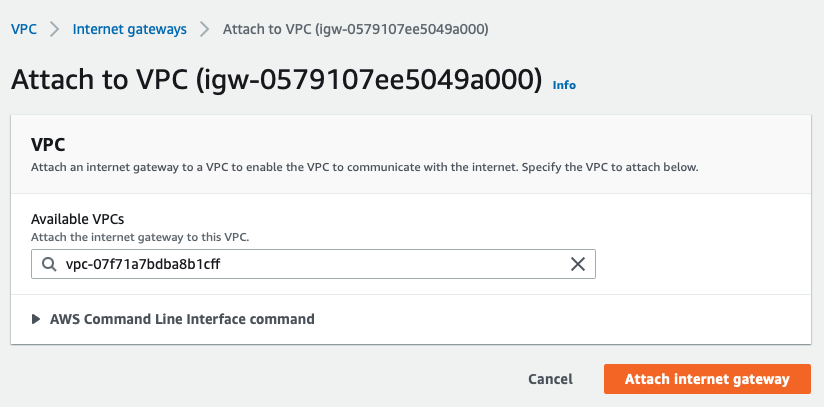
**Step 3: Attach the internet gateway to your VPC**

* 1. In your VPC dashboard, click on the **Internet gateway**
  2. Click on Create Internet gateway
  3. Give its name as **custom-ig**
  4. Click on **Create**
  5. Through internet gateway, Instances running in public subnet can be connected to the internet



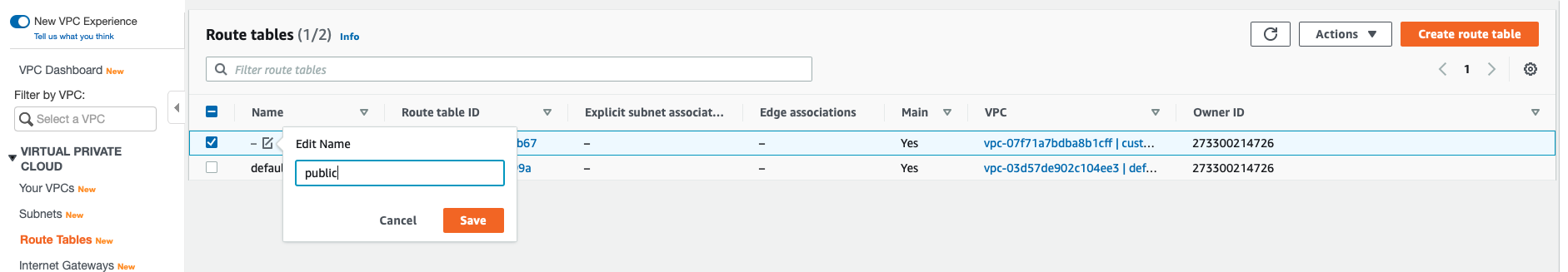
* 1. Select the VPC you have created
  2. Under actions, click on **Attach to VPC**



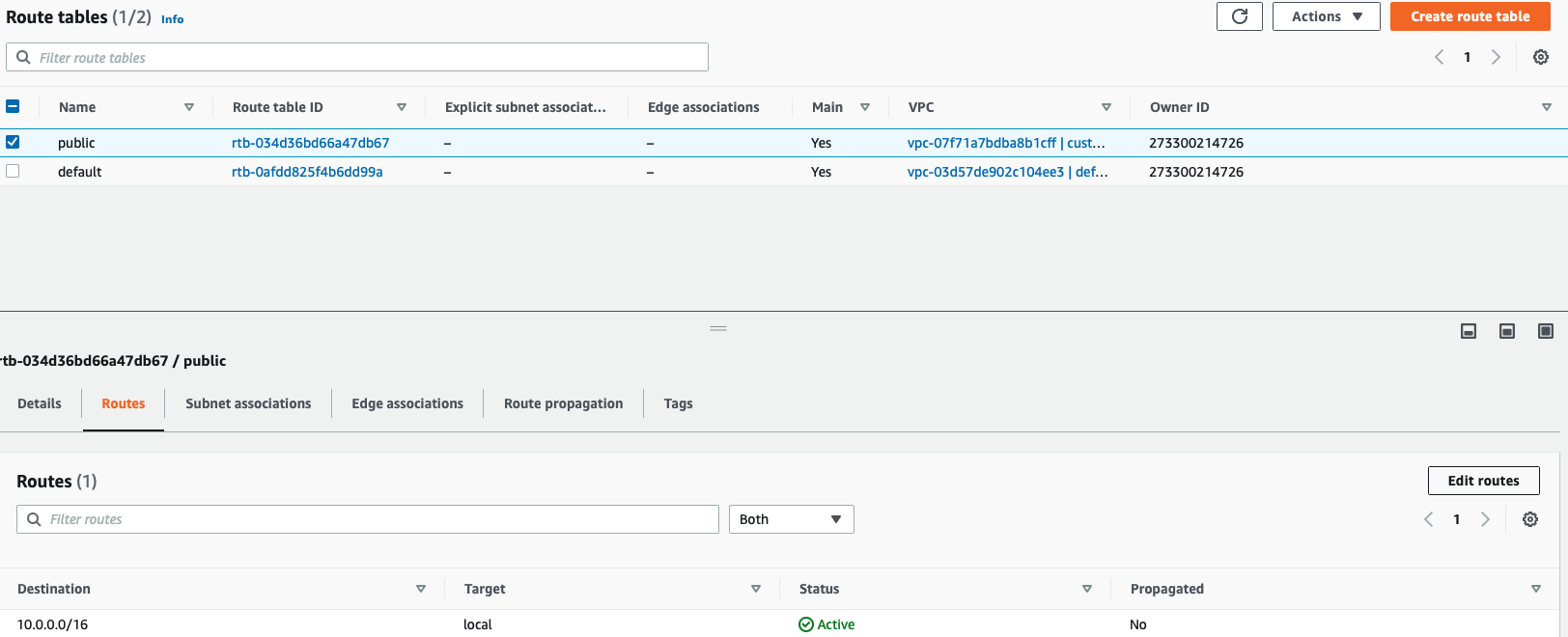


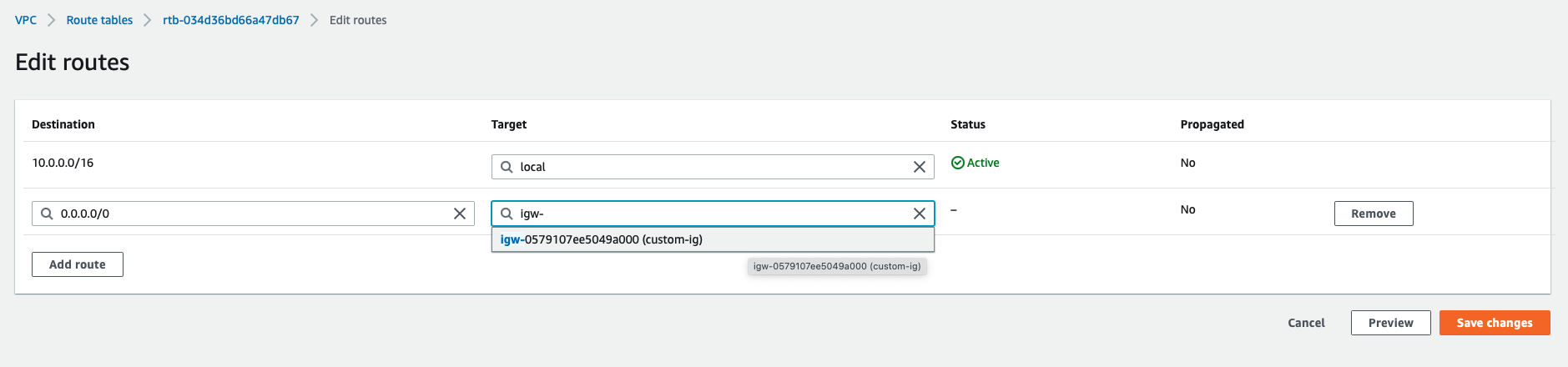
## **Step 4: Create a route table and attach it your VPC**

* 1. There will be a default route table attached to your VPC
  2. Tag the default route table as **public**

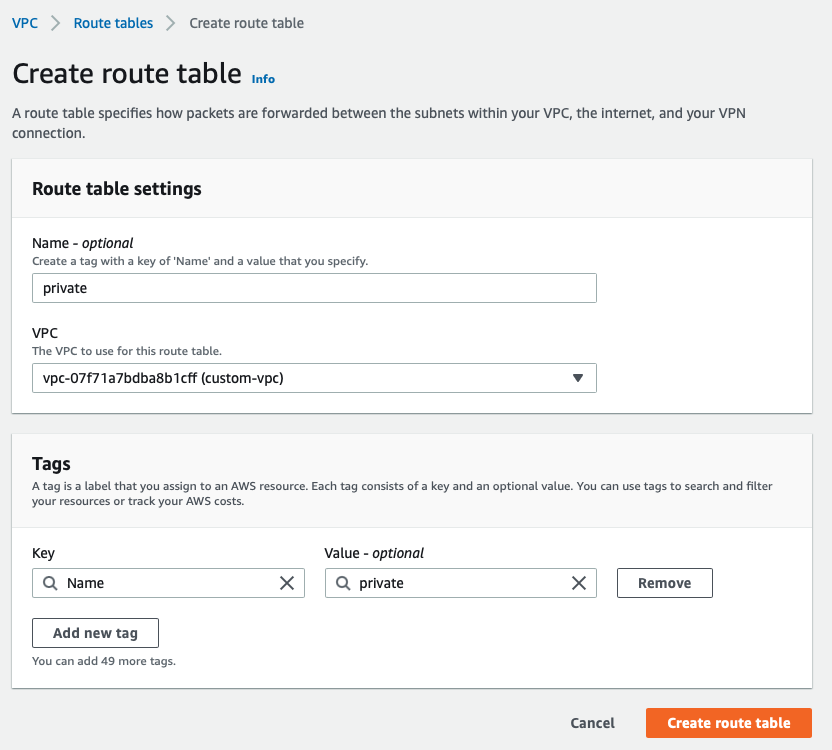
****

* 1. Select Public Route table and update Routes
  2. Type Destination **0.0.0.0/0**
  3. Attach Internet Gateway to the public route table. Click on **Save**



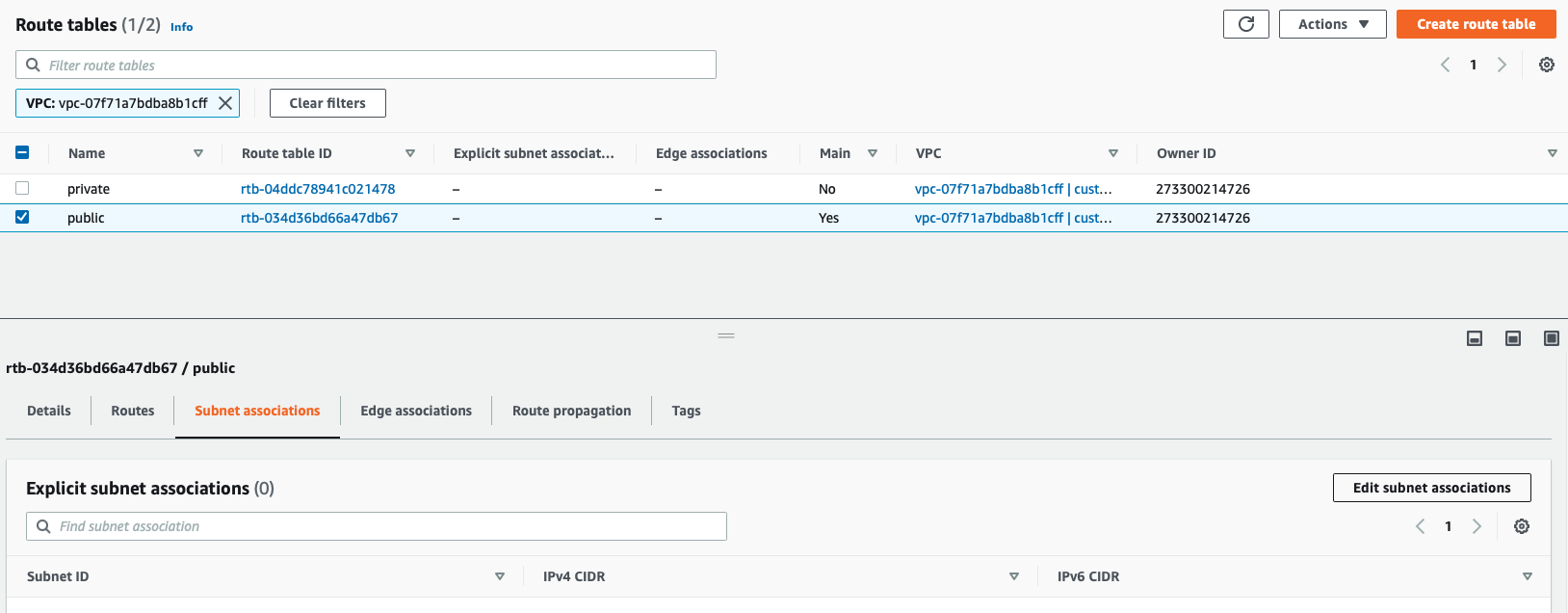


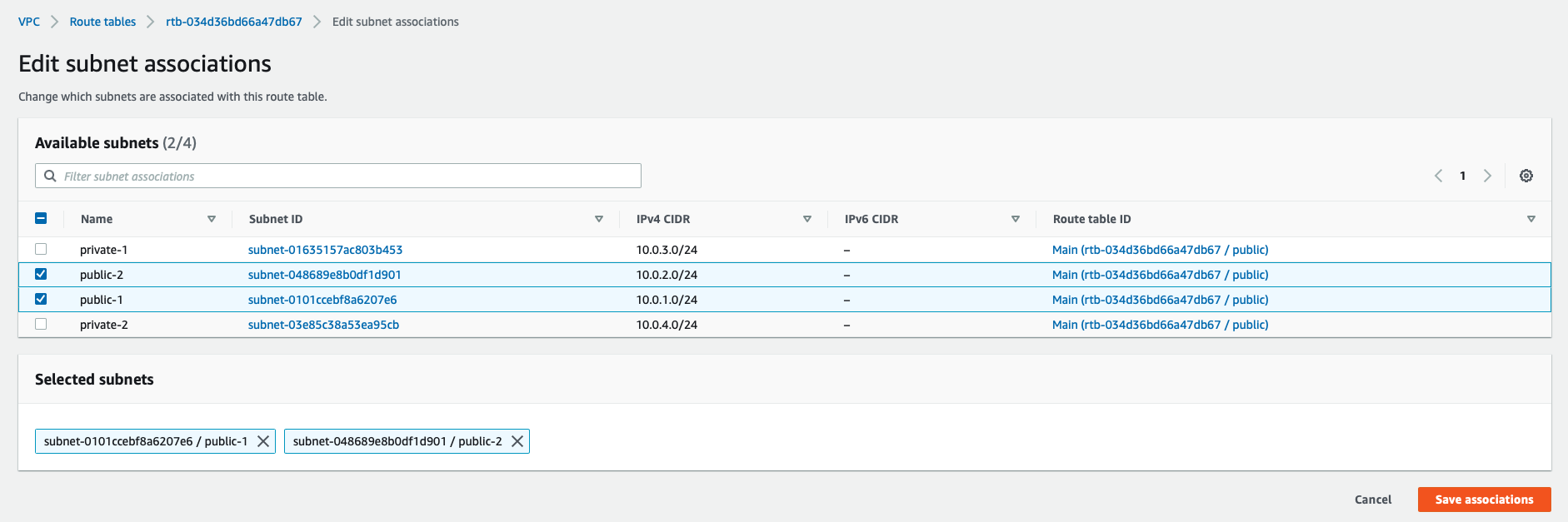
**Create a new route table for the private subnet and name it as private**



**Step 5: Associate the subnets with the route table**

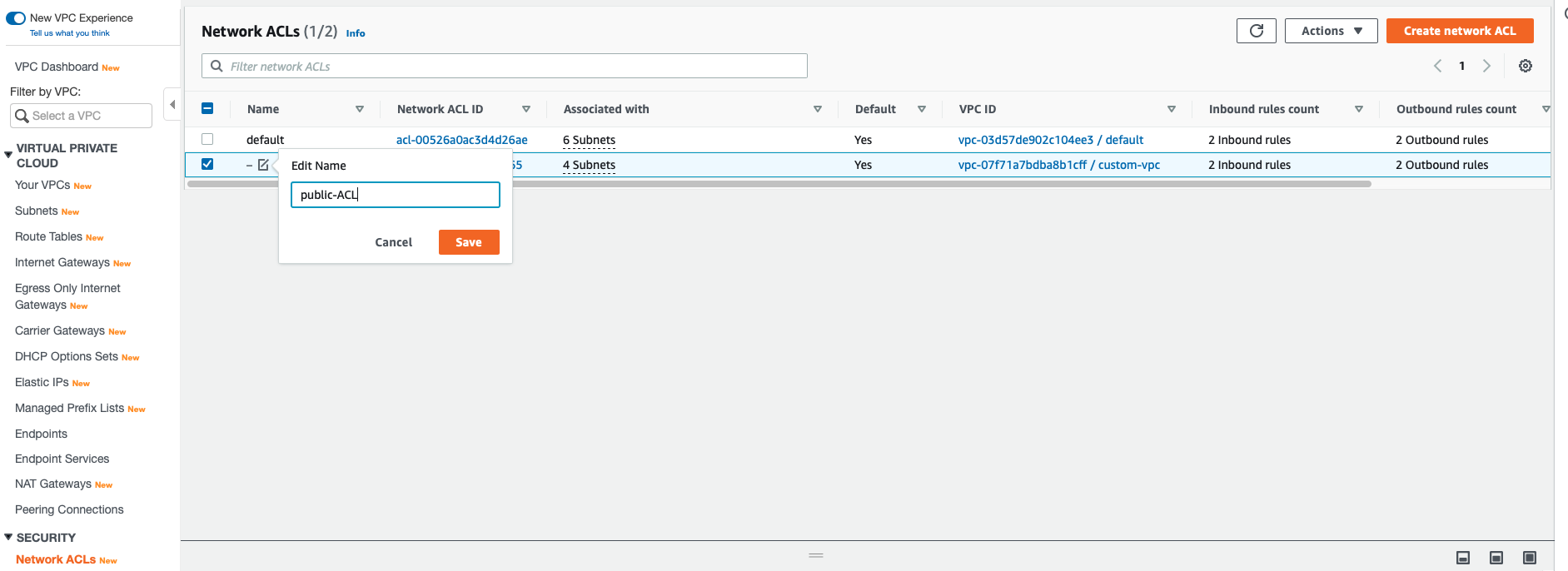
* 1. Select the public subnet and click on **Subnet Association**
  2. Update the public subnets and save it
  3. Repeat the same steps for the private route table and associate private-1 and private -2 subnet with private route table.



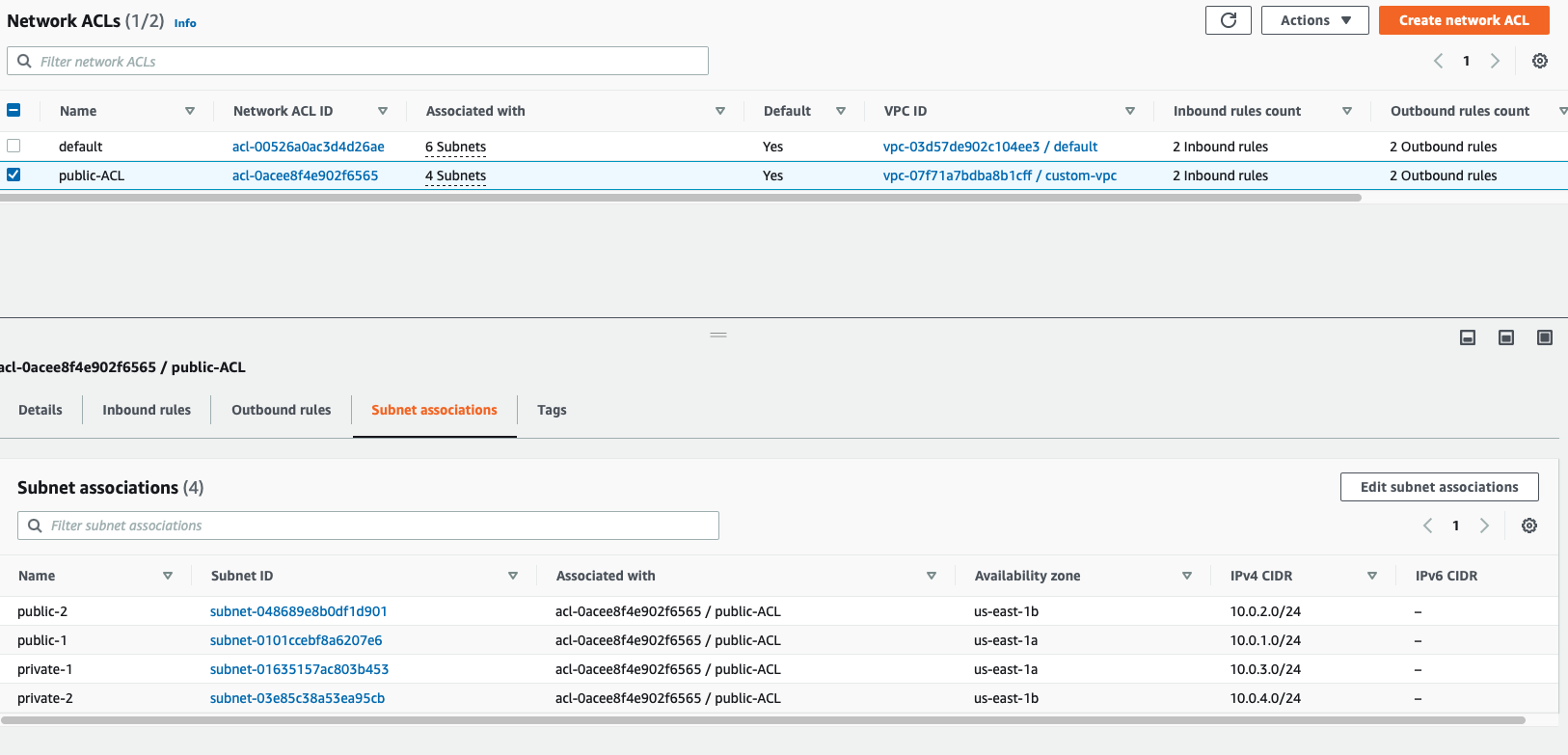


**Step 6: Create a network access control list**

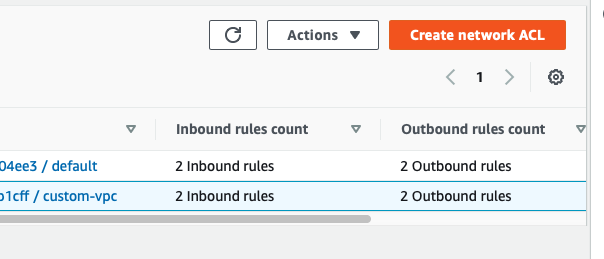
* 1. In your VPCs, select the VPC you have created Under summary, select the default ACL



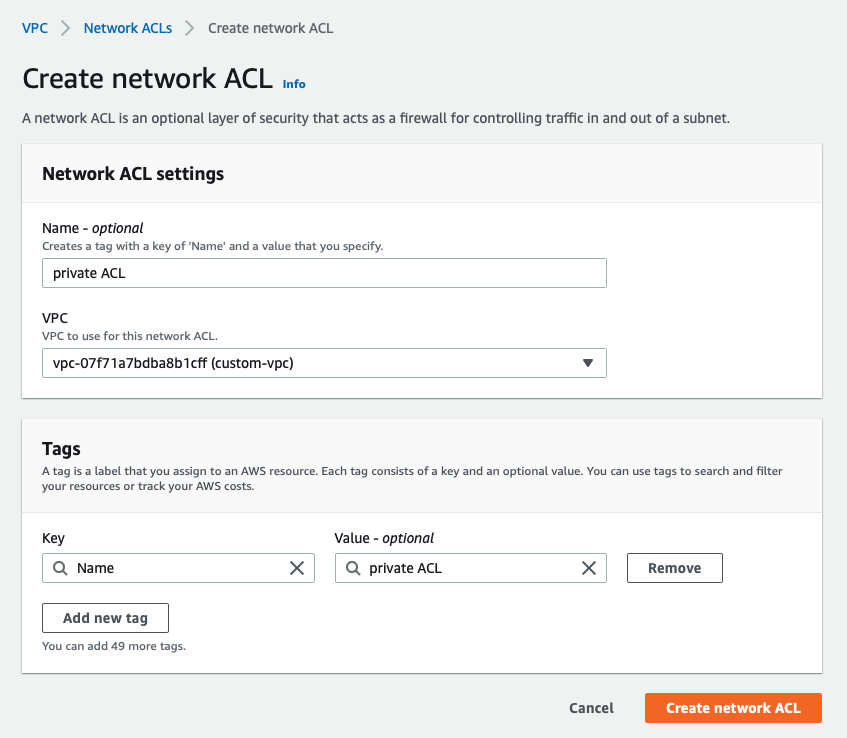
* 1. Default ACL, by default, allows all the inbound and outbound traffic
  2. In the subnet associated on, select only the public subnets



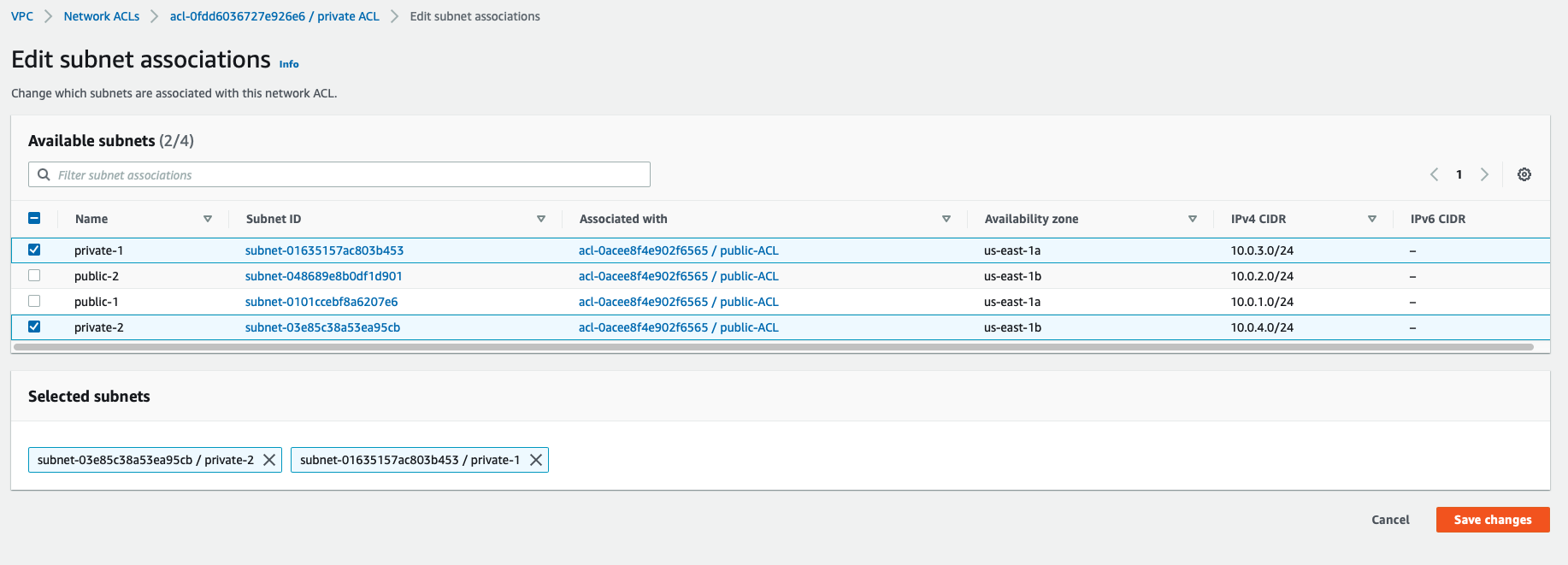
* 1. Create a custom ACL for private subnets



* 1. Select the VPC you have created and click on **Create**



* 1. Select the ACL you have created
  2. Attach private subnets with private ACL and save it.



**Step 7: Create EC2 instances and attach the VPC**

* 1. In your EC2 services, launch a Windows instance in the private subnet
  2. Under Configure on Instance Details, change the Network to the VPC you have created
  3. Choose the subnet in which you want to launch the instance
  4. Click on **Review and Launch.** Your instance will now be created in your virtual network.

