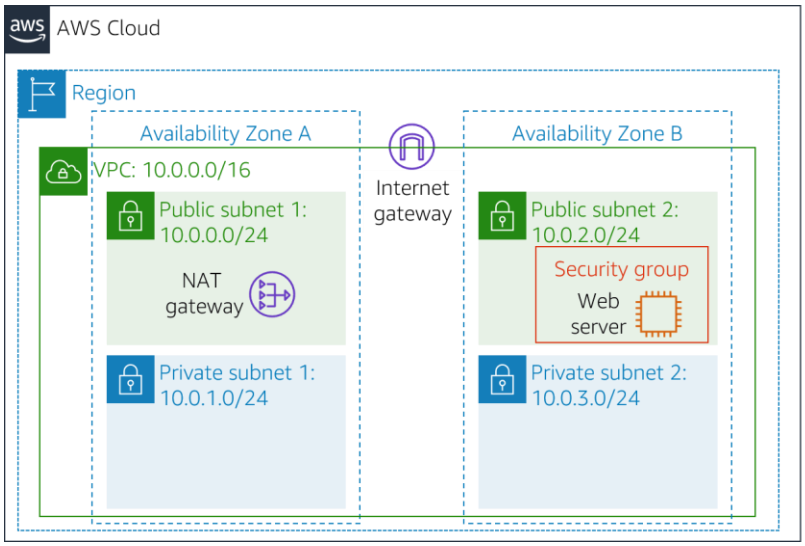
Build your VPC and Launch a Web Server

In this project, you will use Amazon Virtual Private Cloud (VPC) to create your own VPC and add additional components to produce a customized network. You will also create security groups for your EC2 instance. You will then configure and customize an EC2 instance to run a web server and launch it into the VPC.

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you defined. This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure of AWS. You can create a VPC that spans multiple Availability Zones.

NOTE THIS PROJECT IS BEING WORK IN THE NEW AWS MANAGEMENT CONSOLE

In this project you build the following infrastructure:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

Project Objectives

- Create a VPC.
- Create subnets.
- Configure a security group.
- Launch an EC2 instance into a VPC.

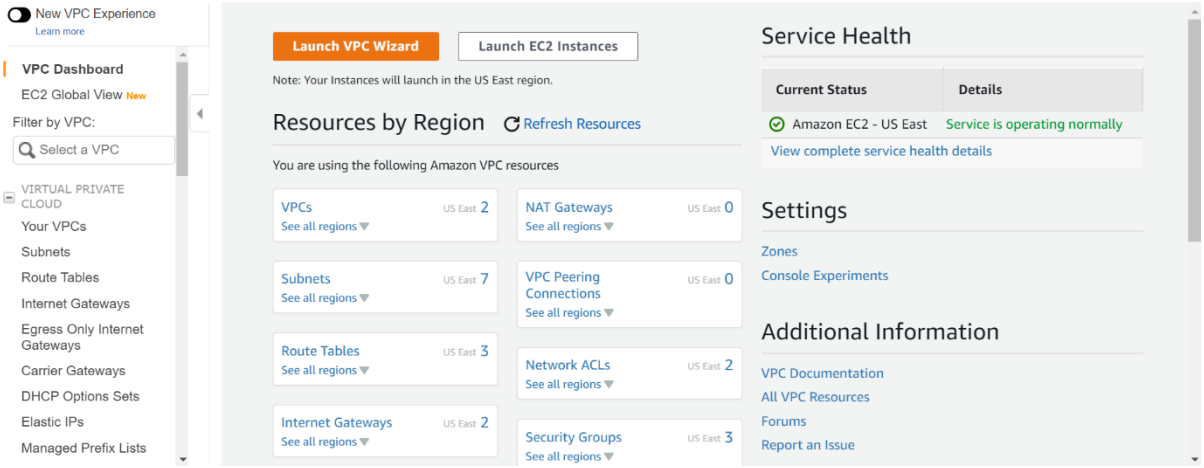
STEP 1: Create Your VPC

In this step, you will use the VPC Wizard to create a VPC an Internet Gateway and two subnets in a single Availability Zone. An **Internet gateway (IGW)** is a VPC component that allows communication between instances in your VPC and the Internet.

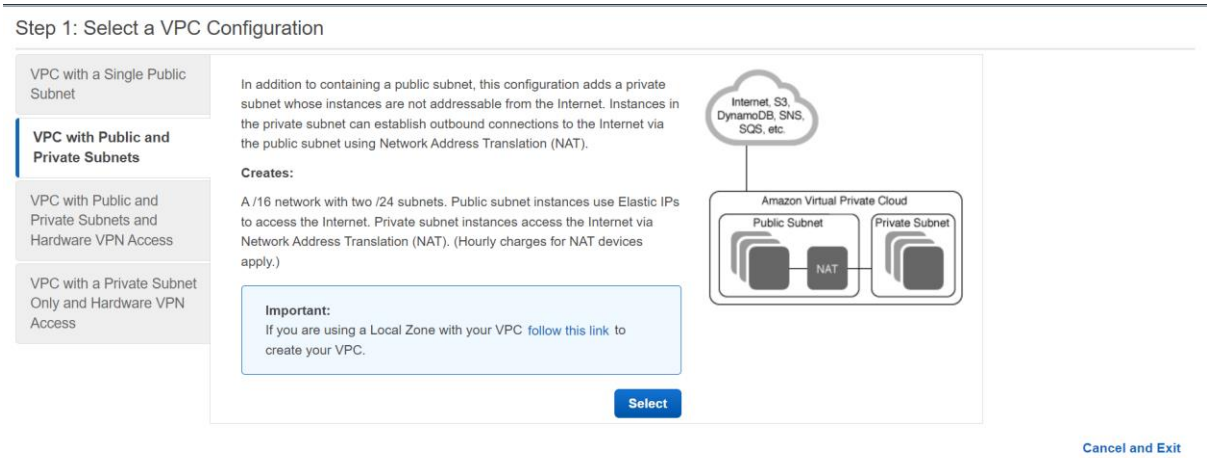
After creating a VPC, you can add **subnets**. Each subnet resides entirely within one Availability Zone and cannot span zones. If a subnet's traffic is routed to an Internet Gateway, the subnet is known as a *public subnet*. If a subnet does not have a route to the Internet gateway, the subnet is known as a *private subnet*.

The wizard will also create a *NAT Gateway*, which is used to provide internet connectivity to EC2 instances in the private subnets.

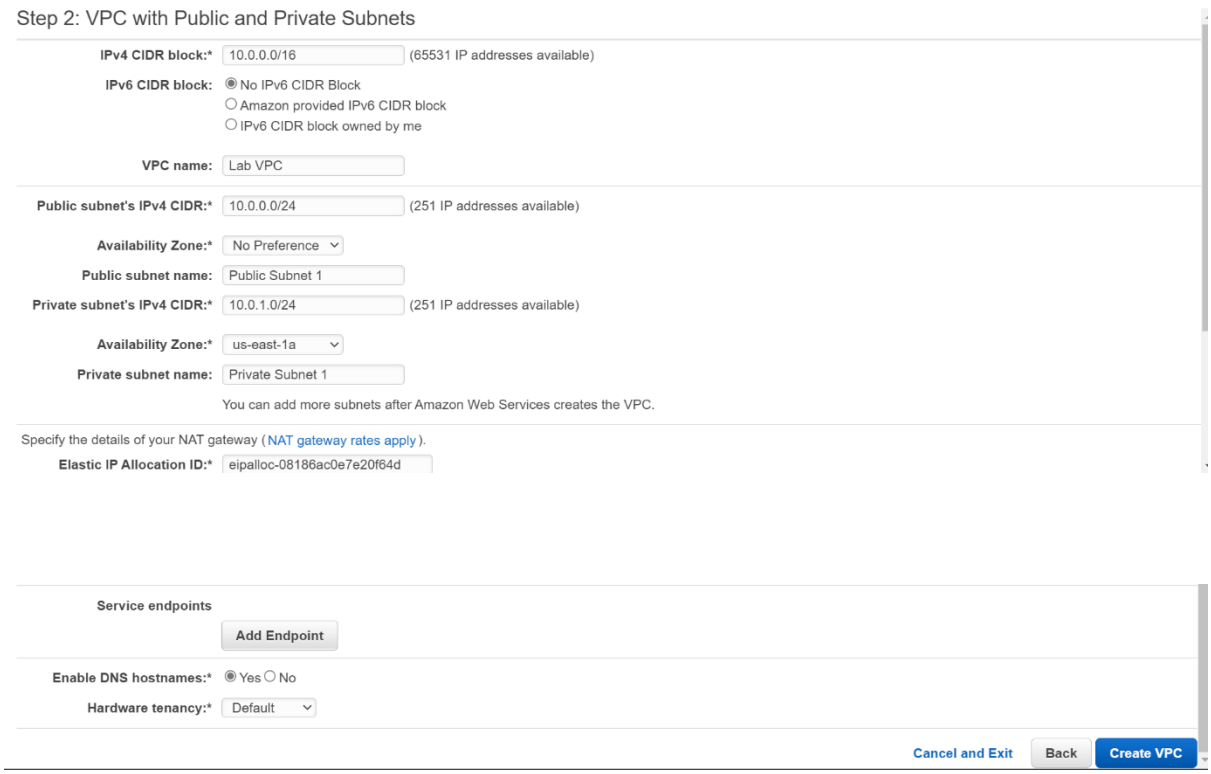
- In the **AWS Management Console**, on the **Services** menu, choose **VPC**.
- Choose **Launch VPC Wizard**
- In the left navigation pane, choose **VPC with Public and Private Subnets** (the second option).



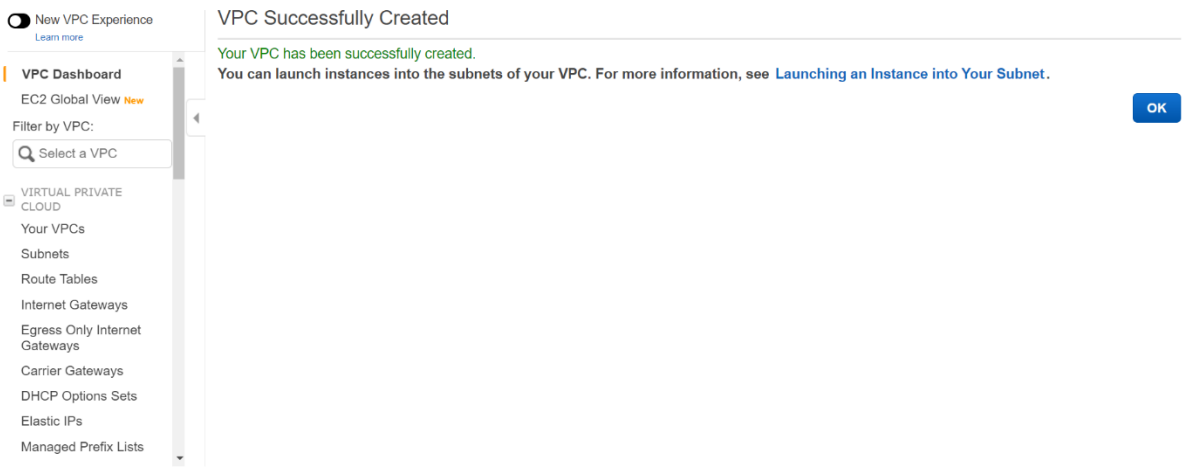
- Choose **Select** then configure:

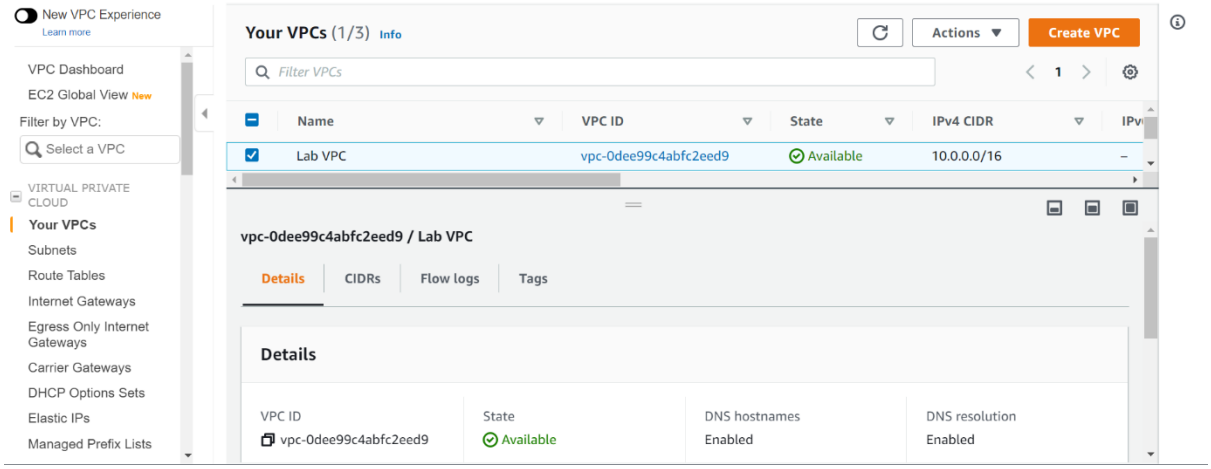


- **VPC name:** Lab VPC
- **Availability Zone:** Select the *first* Availability Zone
- **Public subnet name:** Public Subnet 1
- **Availability Zone:** Select the *first* Availability Zone (the same as used above)
- **Private subnet name:** Private Subnet 1
- **Elastic IP Allocation ID:** Choose in the box and select the displayed IP address
- Choose **Create VPC**

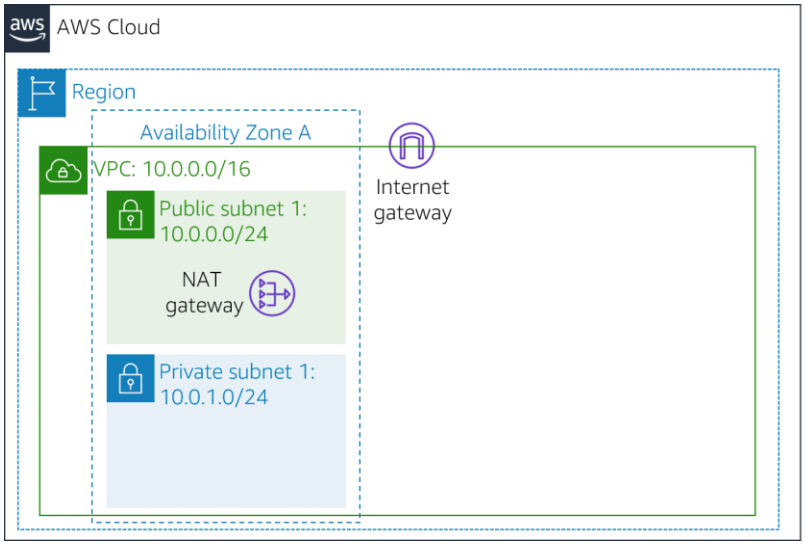


- The wizard will create your VPC.
- Once it is complete, choose **OK**





The wizard has provisioned a VPC with a public subnet and a private subnet in the same Availability Zone, together with route tables for each subnet:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

The Public Subnet has a CIDR of **10.0.0.0/24**, which means that it contains all IP addresses starting with **10.0.0.x**.
The Private Subnet has a CIDR of **10.0.1.0/24**, which means that it contains all IP addresses starting with **10.0.1.x**.

STEP 2: Create Additional Subnets

In this task, you will create two additional subnets in a second Availability Zone. This is useful for creating resources in multiple Availability Zones to provide *High Availability*.

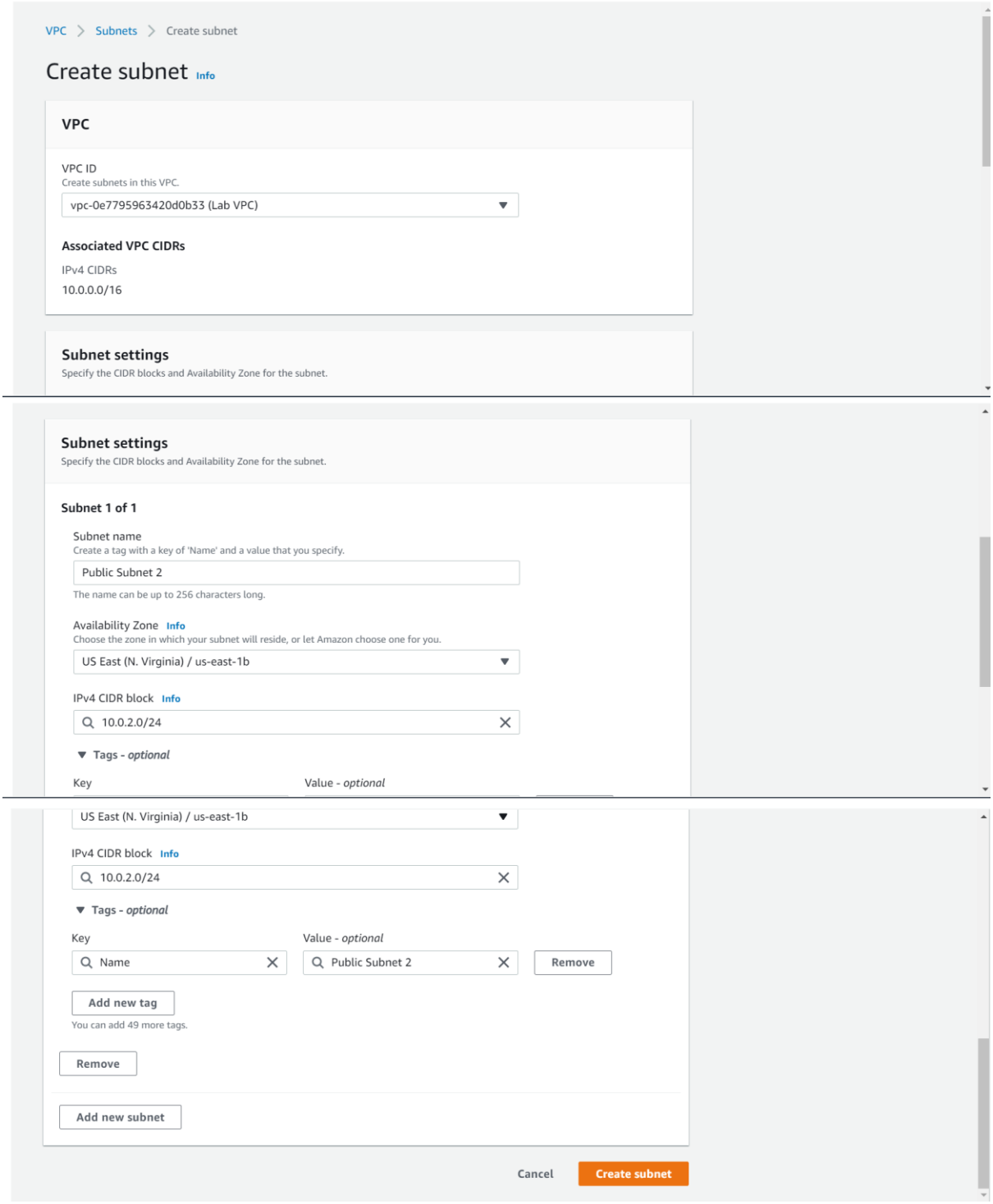
In the left navigation pane, choose Subnets.
First, you will create a second Public Subnet.

- Choose **Create subnet** then configure:
- **VPC ID:** Lab VPC

- **Subnet name:** Public Subnet 2
- **Availability Zone:** Select the *second* Availability Zone
- **IPv4 CIDR block:** 10.0.2.0/24

The subnet will have all IP addresses starting with **10.0.2.x**.

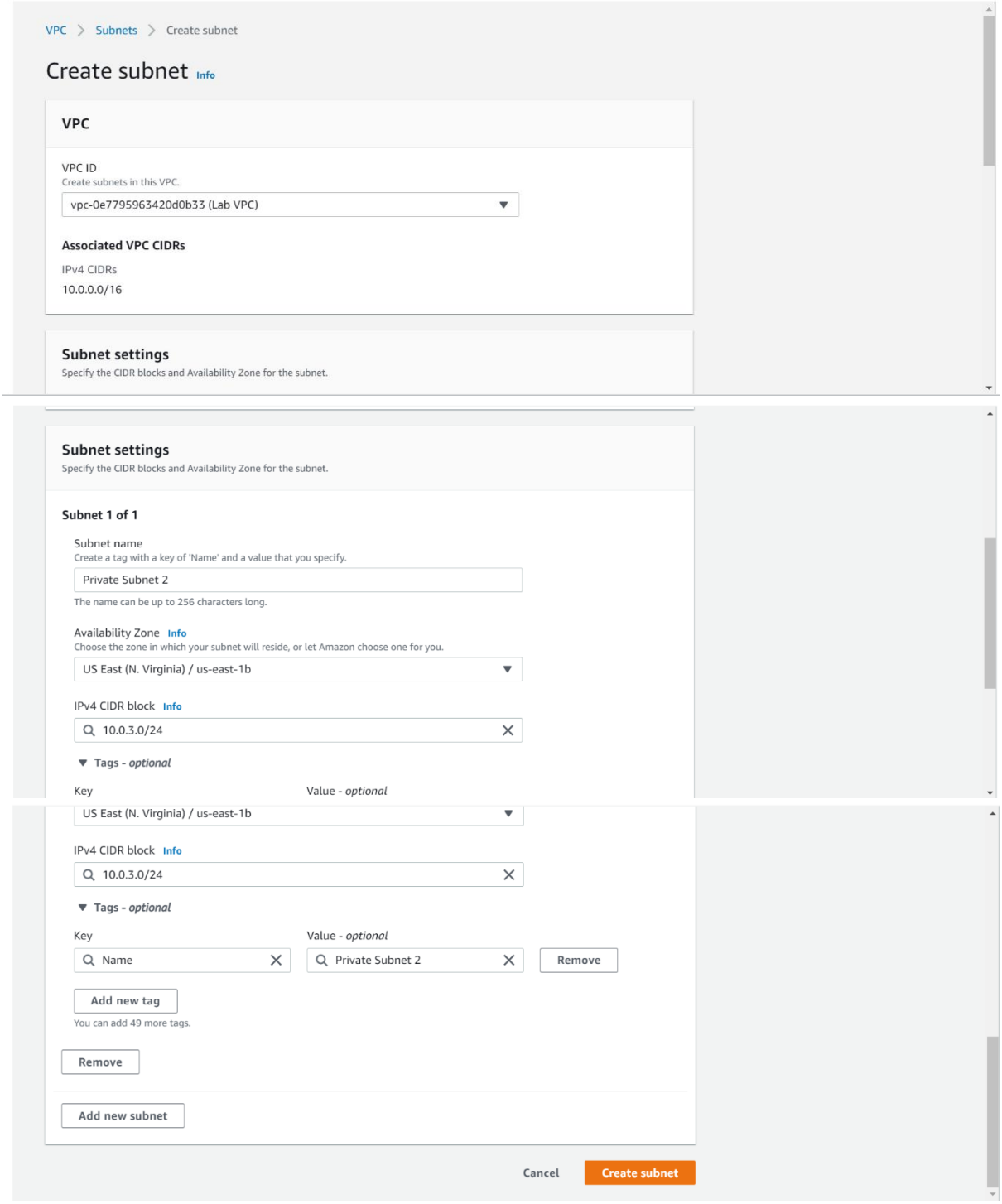
Choose **Create subnet**



You will now create a second Private Subnet.

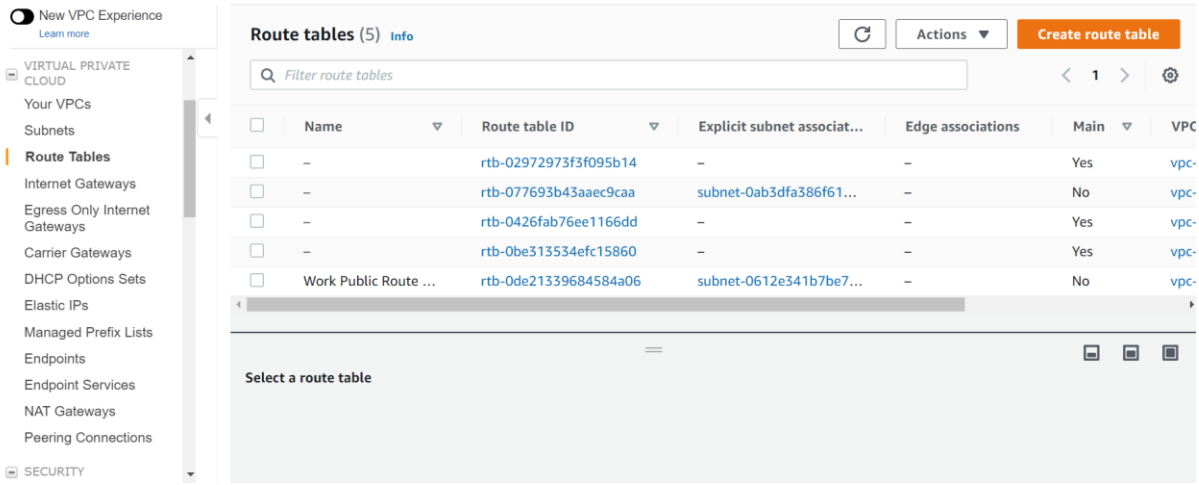
Choose **Create subnet** then configure:

- **VPC ID:** Lab VPC
- **Subnet name:** Private Subnet 2
- **Availability Zone:** Select the *second* Availability Zone
- **CIDR block:** 10.0.3.0/24
- The subnet will have all IP addresses starting with **10.0.3.x**.
- Choose **Create subnet**

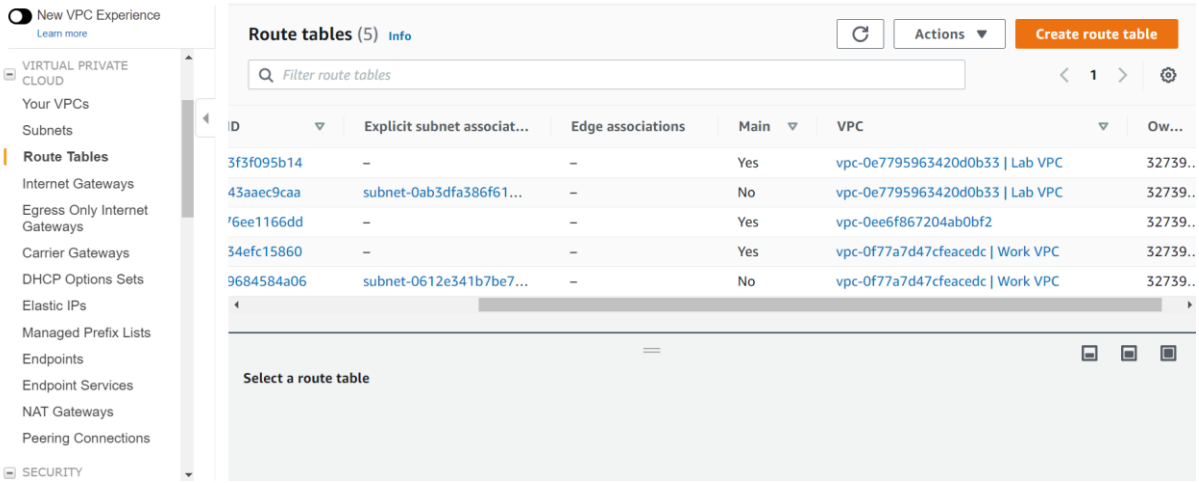


You will now configure the Private Subnets to route internet-bound traffic to the NAT Gateway so that resources in the Private Subnet are able to connect to the Internet, while still keeping the resources private. This is done by configuring a *Route Table*.
A *route table* contains a set of rules, called *routes*, that are used to determine where network traffic is directed. Each subnet in a VPC must be associated with a route table; the route table controls routing for the subnet.

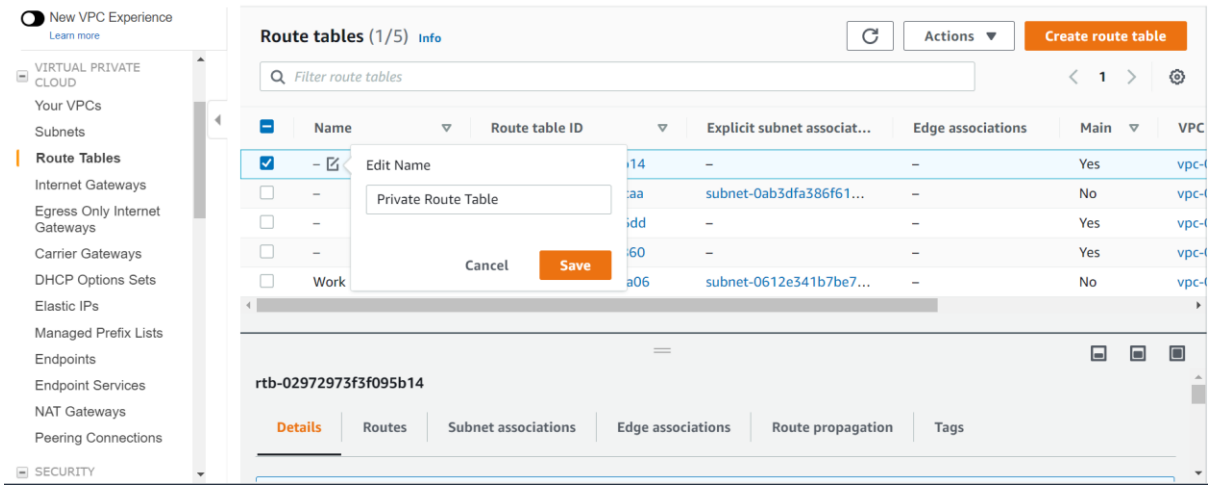
In the left navigation pane, choose **Route Tables**.



Select the route table with **Main = Yes** and **VPC = Lab VPC**. (Expand the *VPC ID* column if necessary to view the VPC name.)



In the **Name** column for this route table, choose the pencil then type Private Route Table and choose **Save**

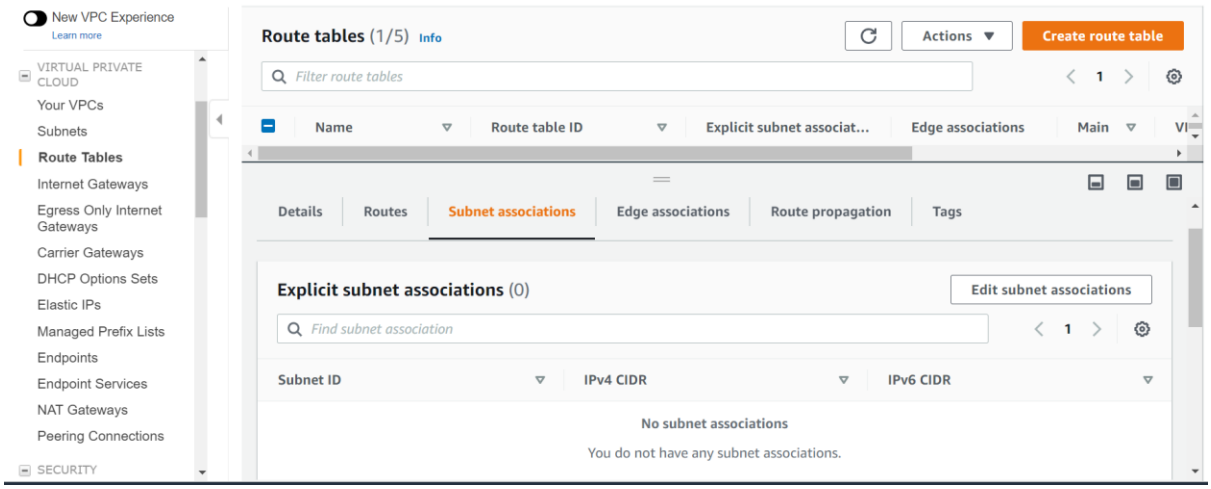


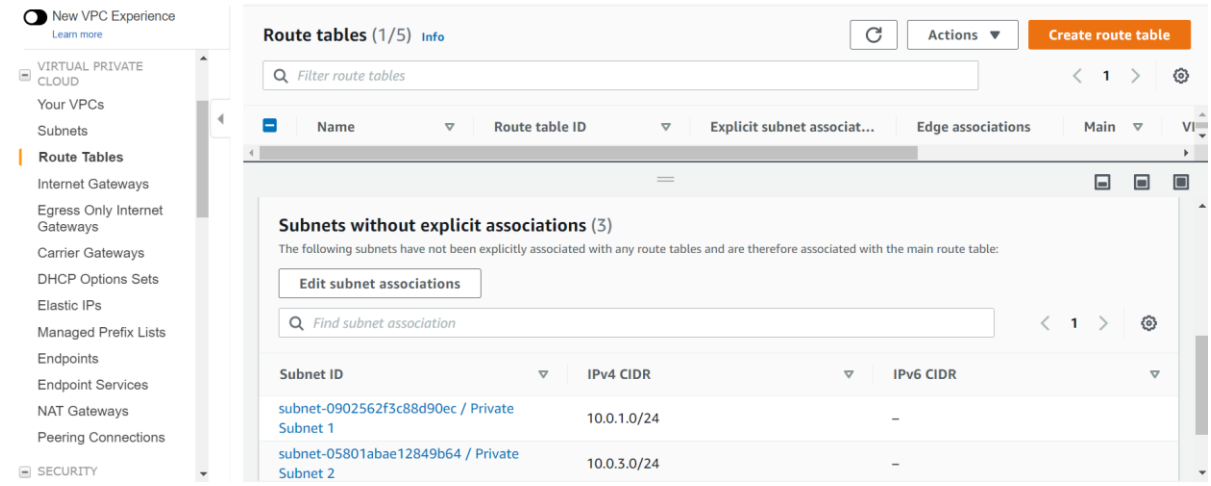
In the lower pane, choose the **Routes** tab.

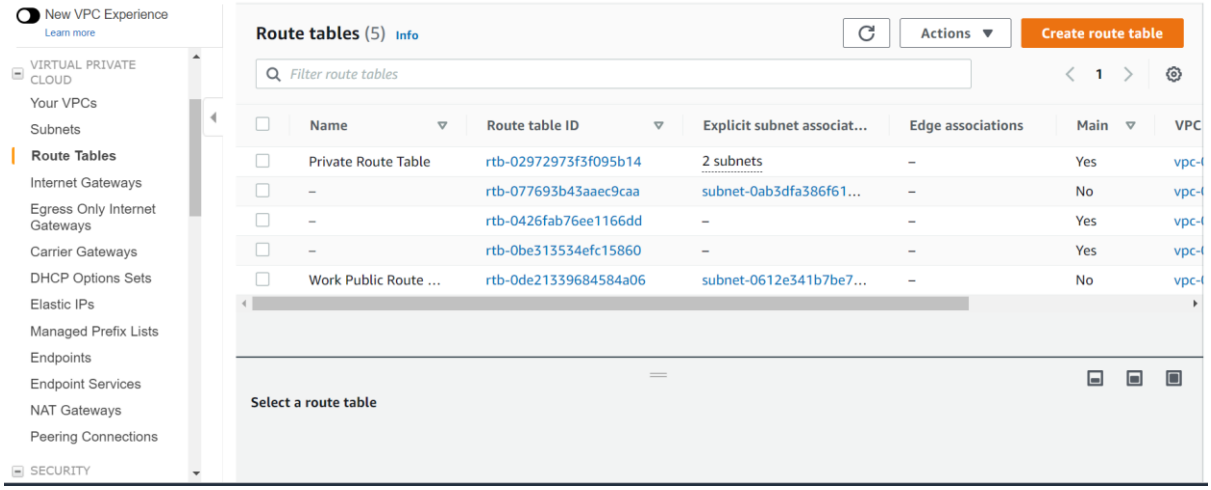
Note: This means that traffic destined for the internet (0.0.0.0/0) will be sent to the NAT Gateway. The NAT Gateway will then forward the traffic to the internet.

This route table is therefore being used to route traffic from Private Subnets. You will now add a name to the Route Table to make this easier to recognize in future.

- In the lower pane, choose the **Subnet Associations** tab.
- You will now associate this route table to the Private Subnets.
- Choose **Edit subnet associations**
- Select both **Private Subnet 1** and **Private Subnet 2**.
- You can expand the *Subnet ID* column to view the Subnet names.
- Choose **Save associations**



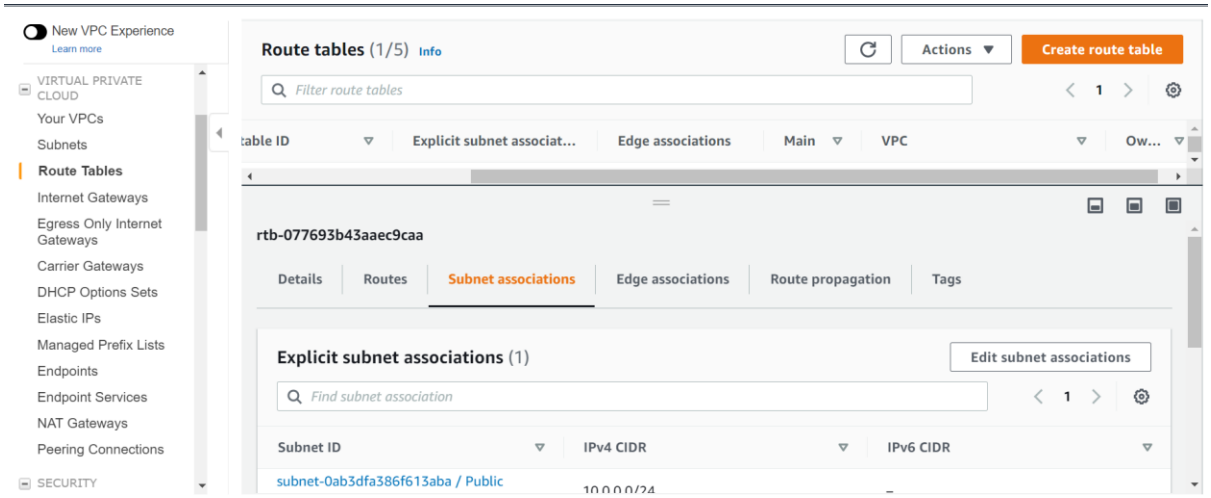


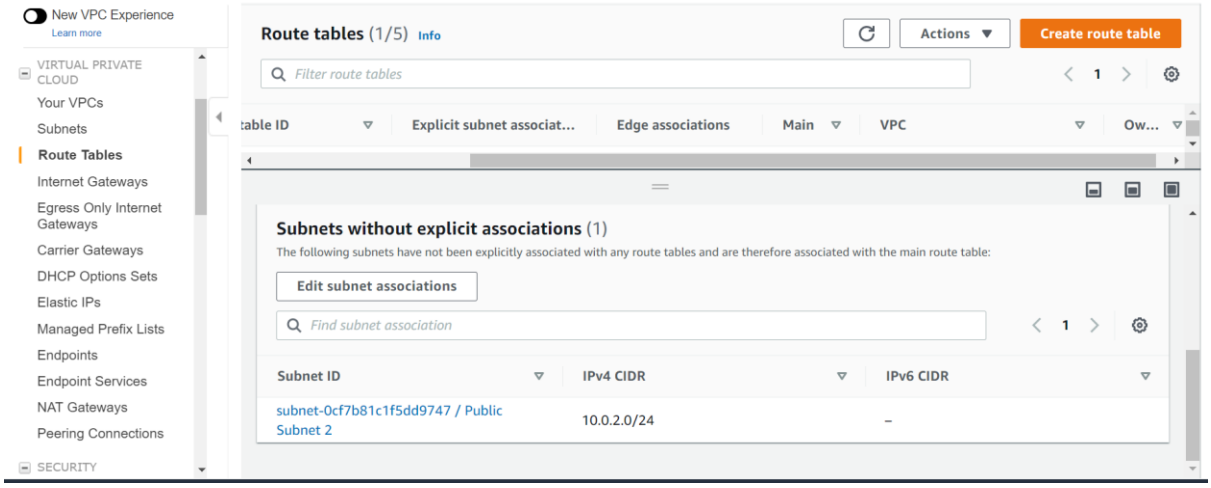


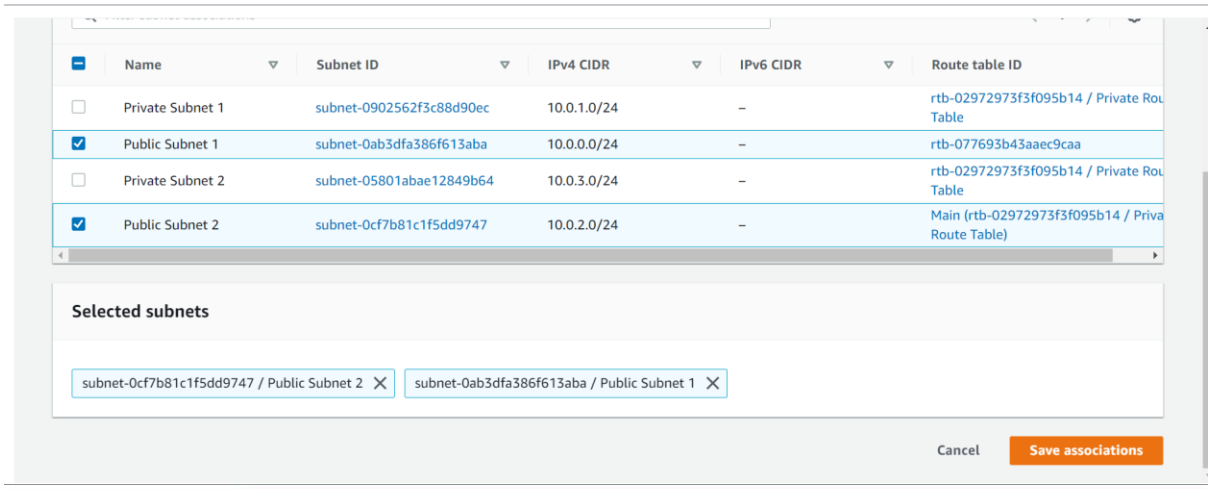
You will now configure the Route Table that is used by the Public Subnets.

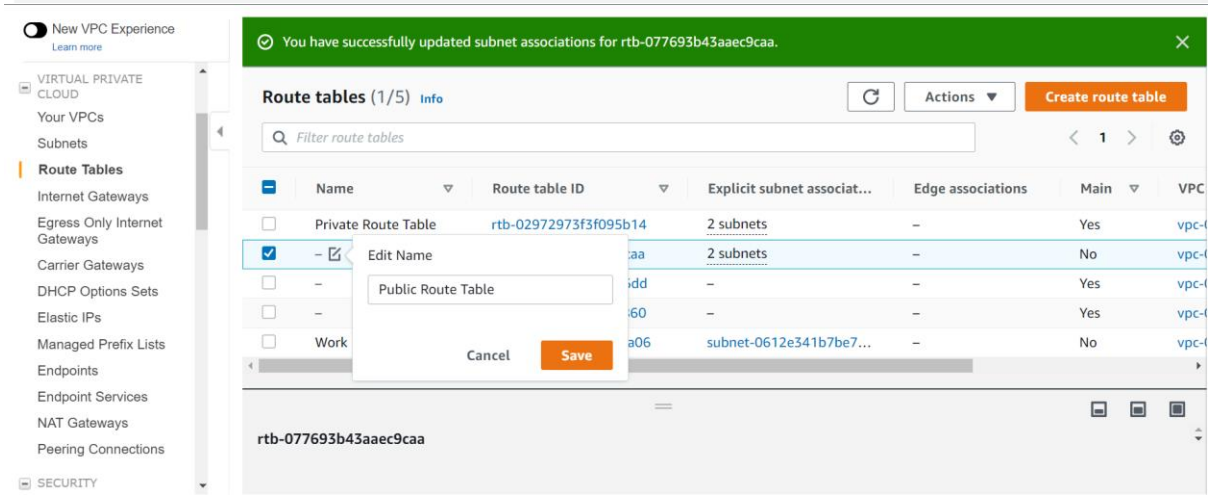
- Select the route table with **Main = No** and **VPC = Lab VPC** (and deselect any other subnets).
- In the **Name** column for this route table, choose the pencil then type Public Route Table, and choose **Save**
- In the lower pane, choose the **Routes** tab.
- You will now associate this route table to the Public Subnets.

- Choose the **Subnet Associations** tab.
- Choose **Edit subnet associations**
- Select both **Public Subnet 1** and **Public Subnet 2**.
- Choose **Save associations**

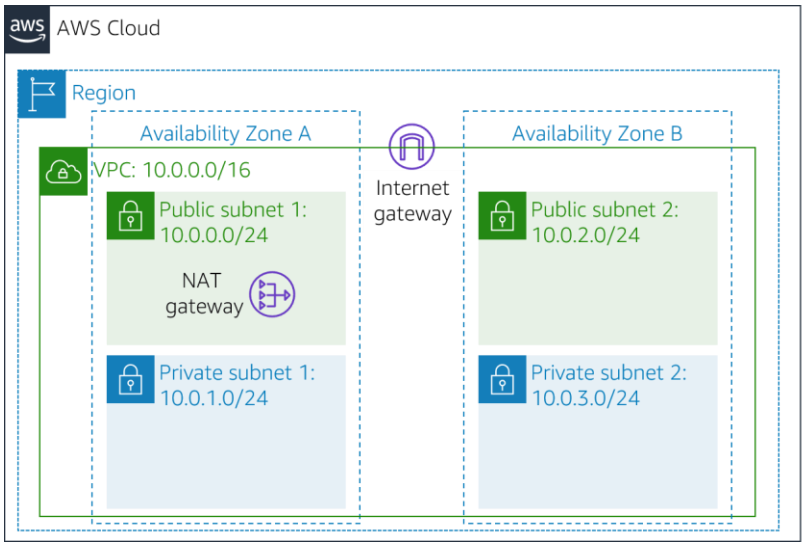








Your VPC now has public and private subnets configured in two Availability Zones:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

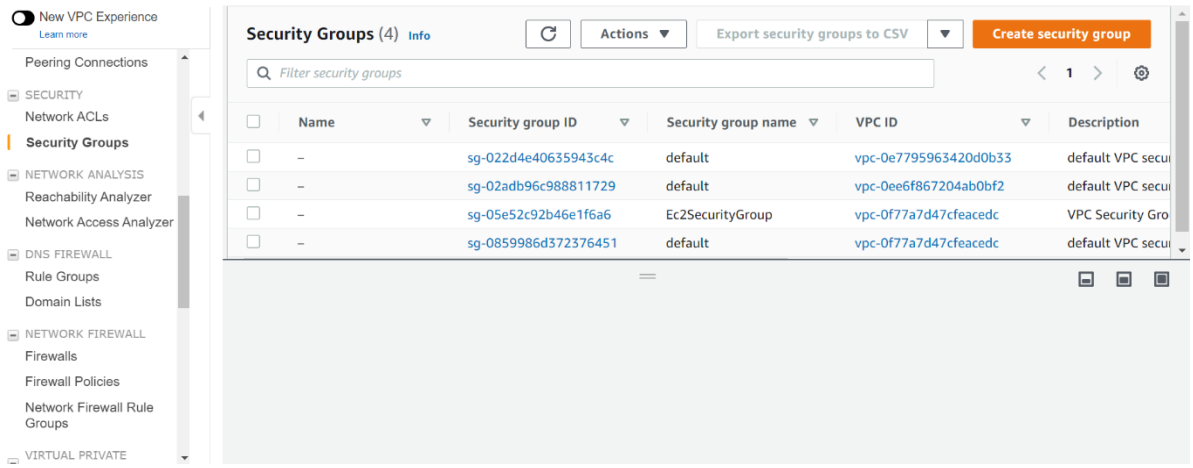
Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

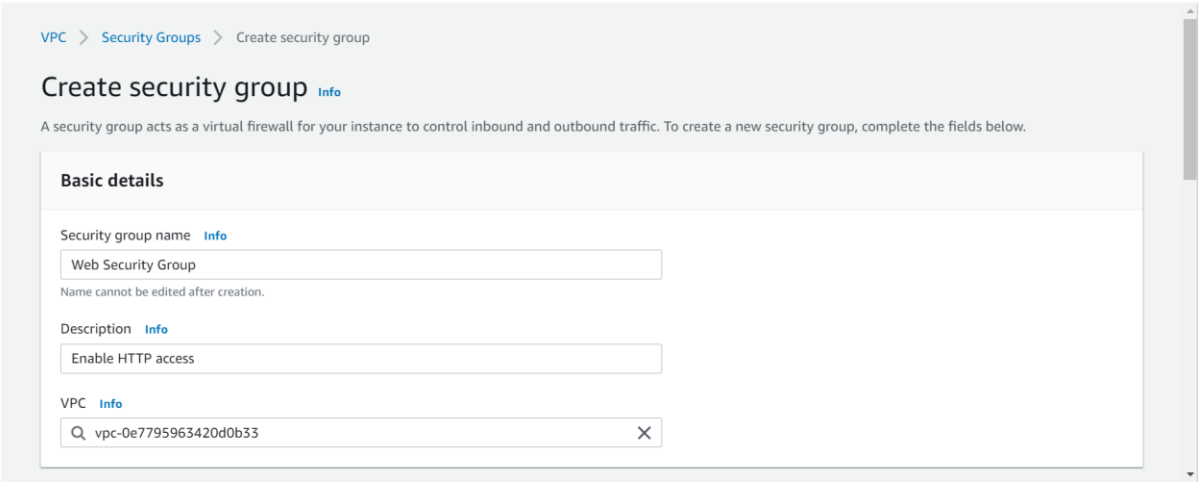
STEP 3: Create a VPC Security Group

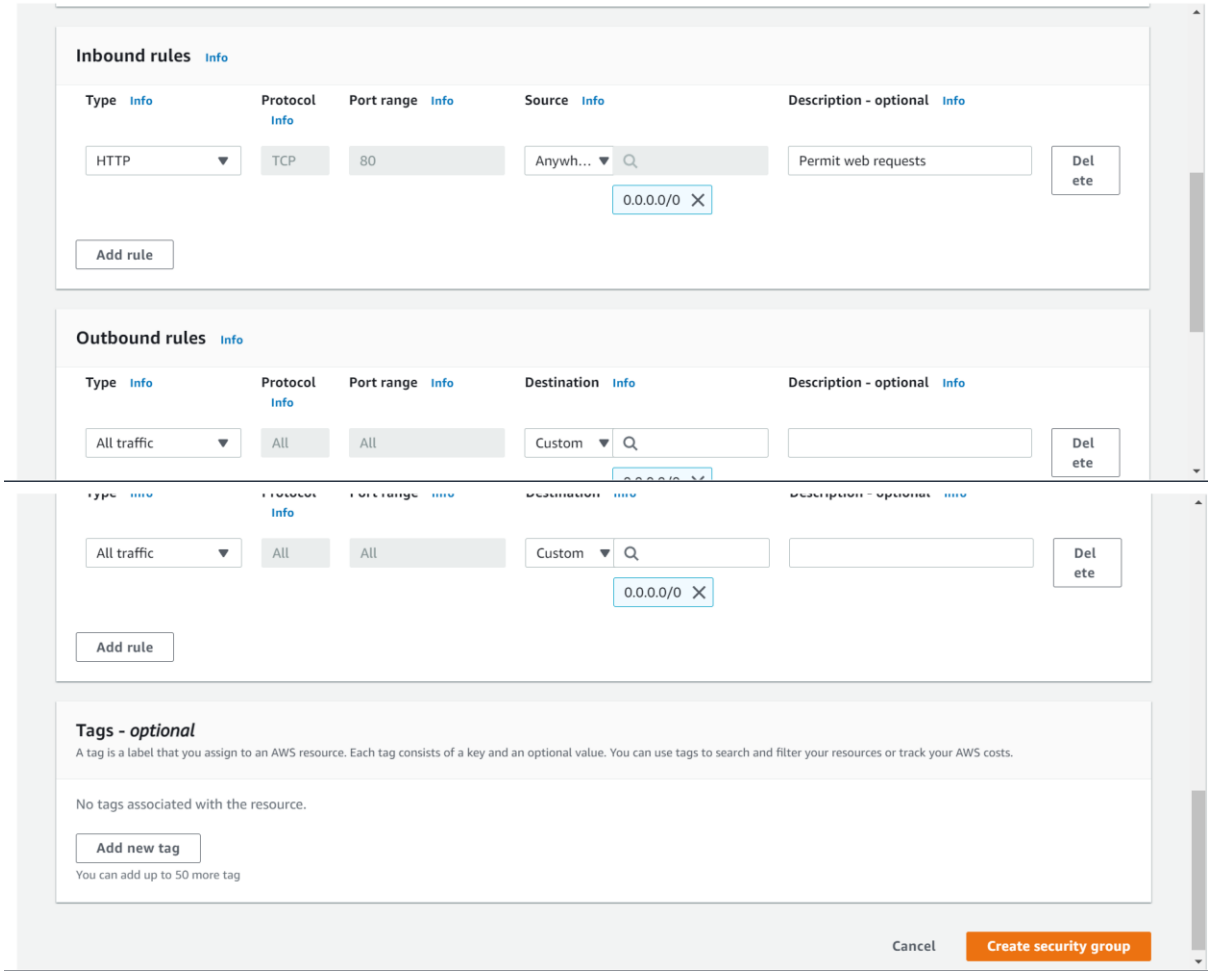
In this task, you will create a VPC security group, which acts as a virtual firewall. When you launch an instance, you associate one or more security groups with the instance. You can add rules to each security group that allow traffic to or from its associated instances.

In the left navigation pane, choose **Security Groups**.

- Choose **Create security group** and then configure:
- **Security group name:** Web Security Group
- **Description:** Enable HTTP access
- **VPC:** *Lab VPC*
- In the **Inbound rules** pane, choose **Add rule**
- Configure the following settings:
- **Type:** *HTTP*
- **Source:** *Anywhere-IPv4*
- **Description:** Permit web requests
- Scroll to the bottom of the page and choose **Create security group**





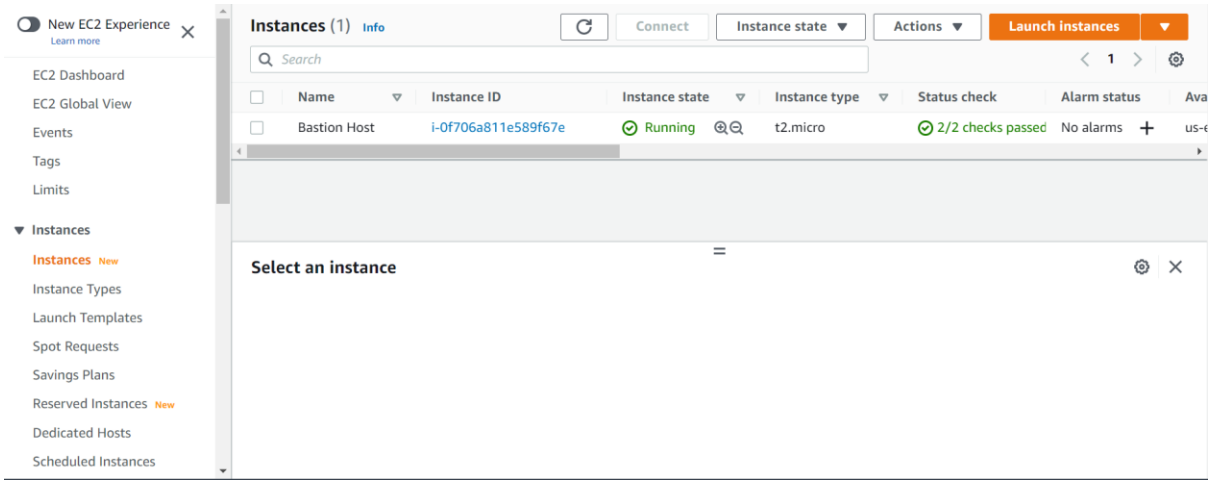


You will use this security group in the next task when launching an Amazon EC2 instance.

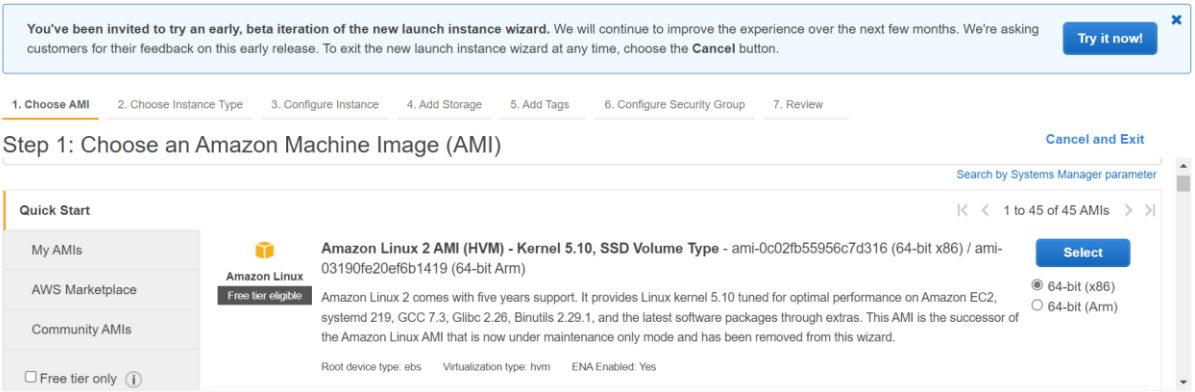
STEP 4: Launch a Web Server Instance

In this task, you will launch an Amazon EC2 instance into the new VPC. You will configure the instance to act as a web server.

- On the **Services** menu, choose **EC2**.
- Choose **Launch Instance**, and then choose **Launch Instance**

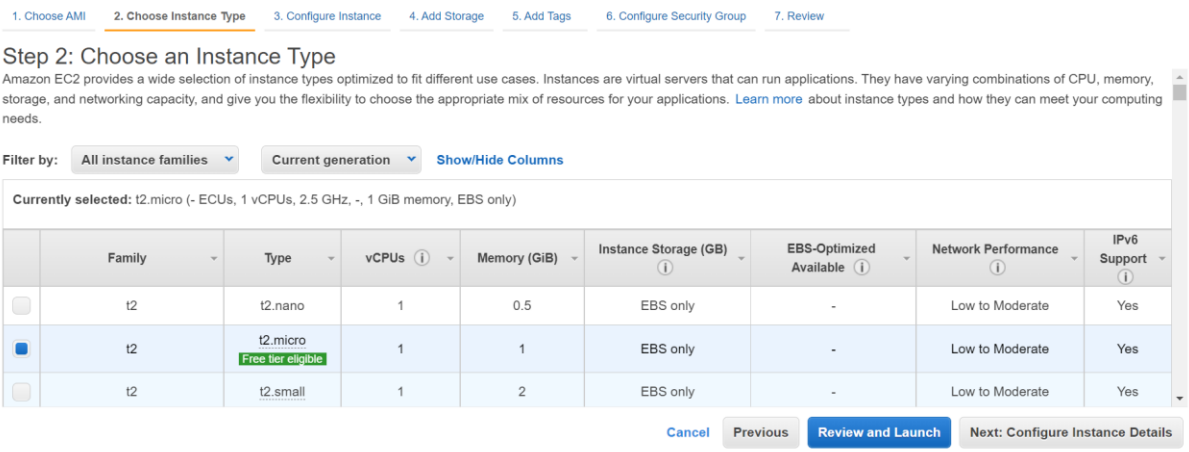


- First, you will select an *Amazon Machine Image (AMI)*, which contains the desired Operating System.
- In the row for **Amazon Linux 2** (at the top), choose **Select**



The *Instance Type* defines the hardware resources assigned to the instance.

- Select **t2.micro** (shown in the *Type* column).

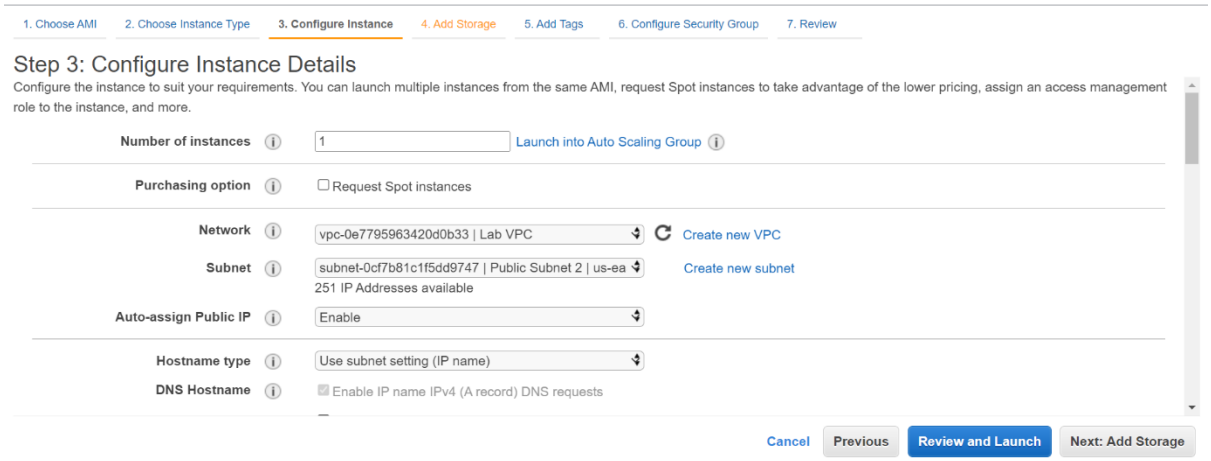


- Choose **Next: Configure Instance Details**

You will now configure the instance to launch in a Public Subnet of the new VPC.

Configure these settings:

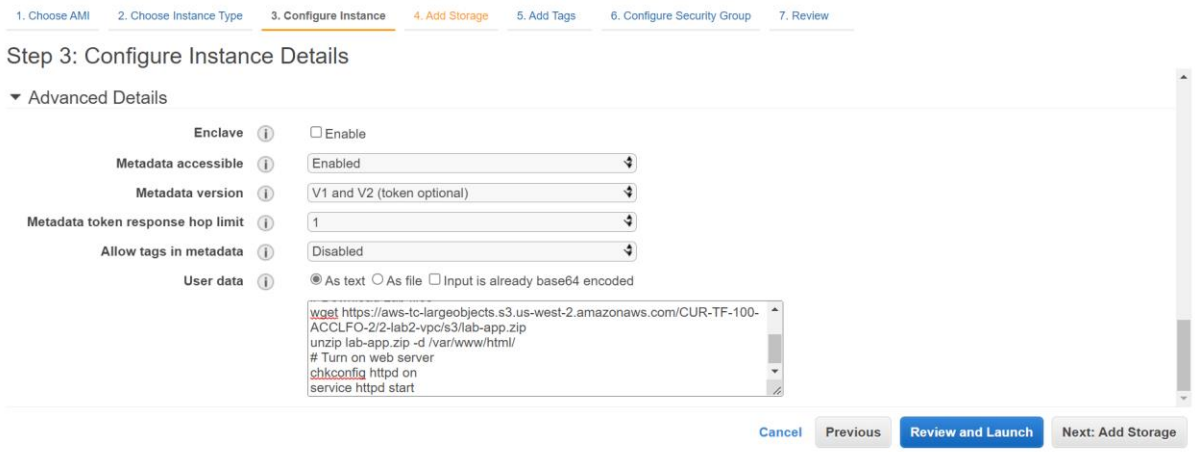
- **Network:** *Lab VPC*
- **Subnet:** *Public Subnet 2* (*not* Private!)
- **Auto-assign Public IP:** *Enable*



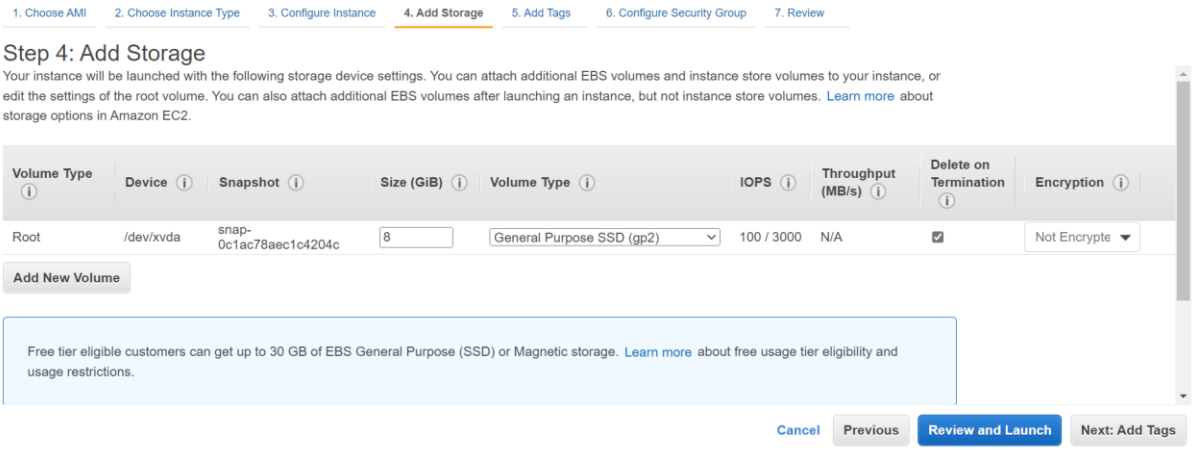
- Expand the **Advanced Details** section (at the bottom of the page).
- Copy and paste this code into the **User data** box:

```
#!/bin/bash
# Install Apache Web Server and PHP
yum install -y httpd mysql php
# Download Lab files
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-ACCLFO-2/2-lab2-vpc/s3/lab-app.zip
unzip lab-app.zip -d /var/www/html/
# Turn on web server
chkconfig httpd on
service httpd start
```

This script will be run automatically when the instance launches for the first time. The script loads and configures a PHP web application.

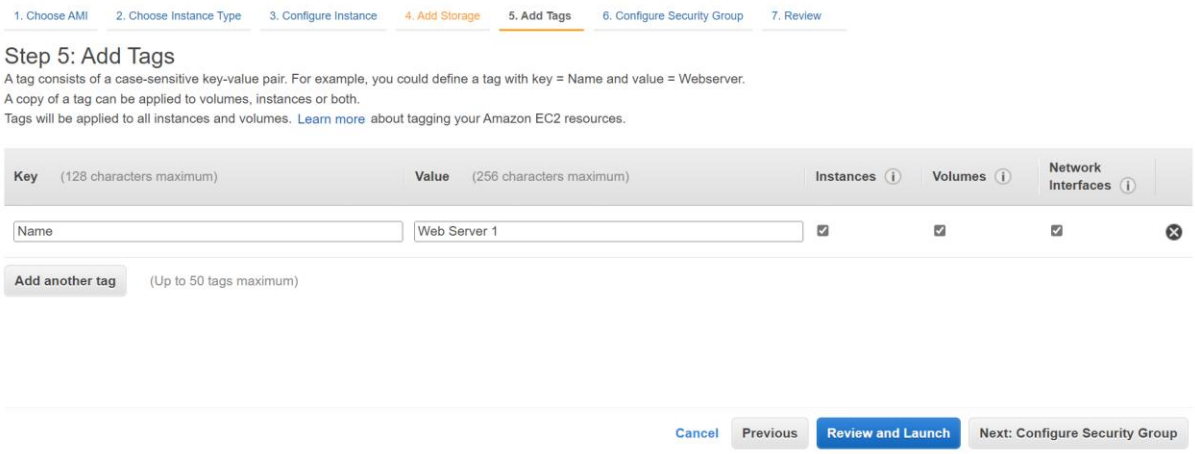


- Choose **Next: Add Storage**
- You will use the default settings for storage.



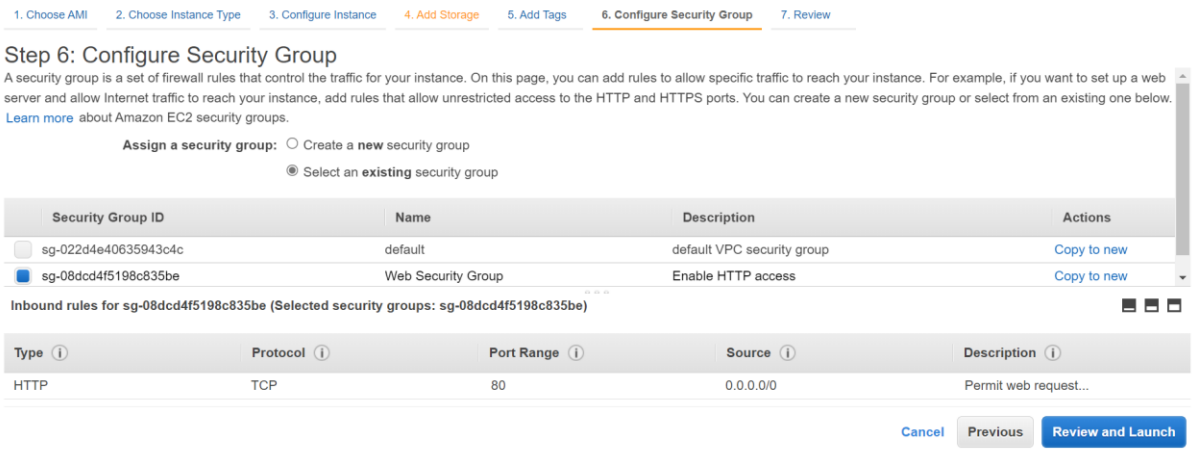
- Choose **Next: Add Tags**
- Tags can be used to identify resources. You will use a tag to assign a Name to the instance.

- Choose **Add Tag** then configure:
- **Key:** Name
- **Value:** Web Server 1



- Choose **Next: Configure Security Group**

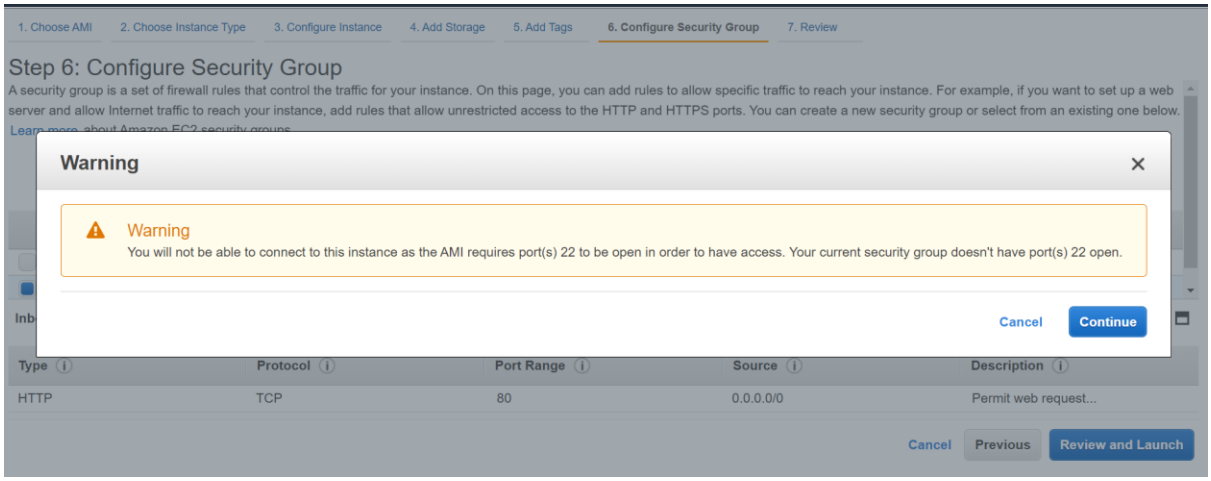
You will configure the instance to use the *Web Security Group* that you created earlier.



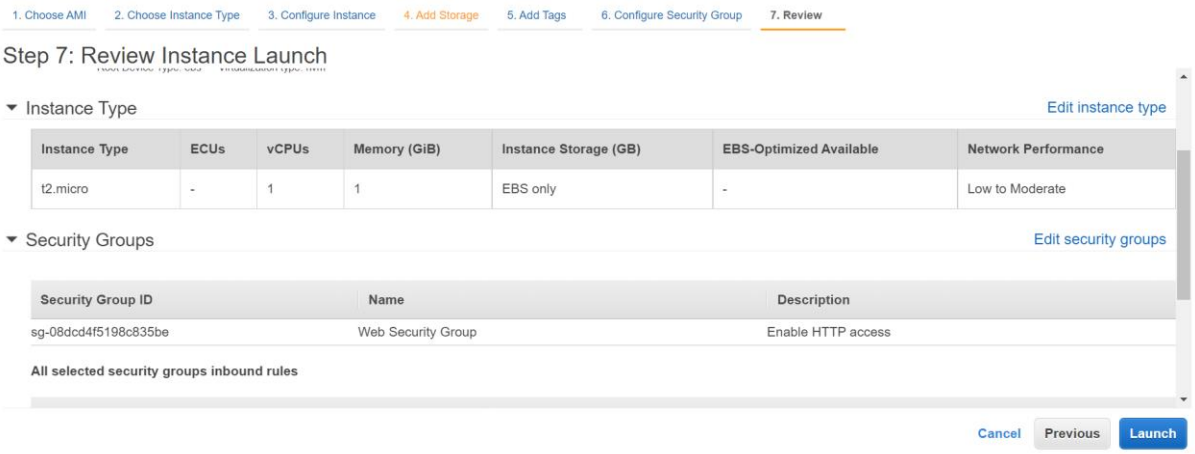
- Select **an existing security group**
- Select **Web Security Group**

This is the security group you created in the previous task. It will permit HTTP access to the instance.

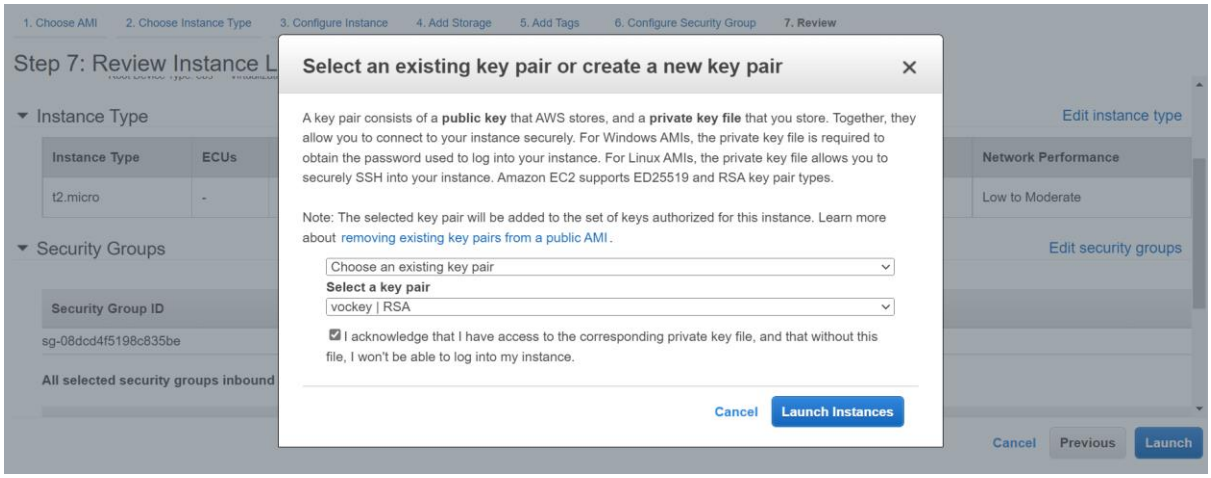
- Choose **Review and Launch**
- When prompted with a *warning* that you will not be able to connect to the instance through port 22, choose **Continue**

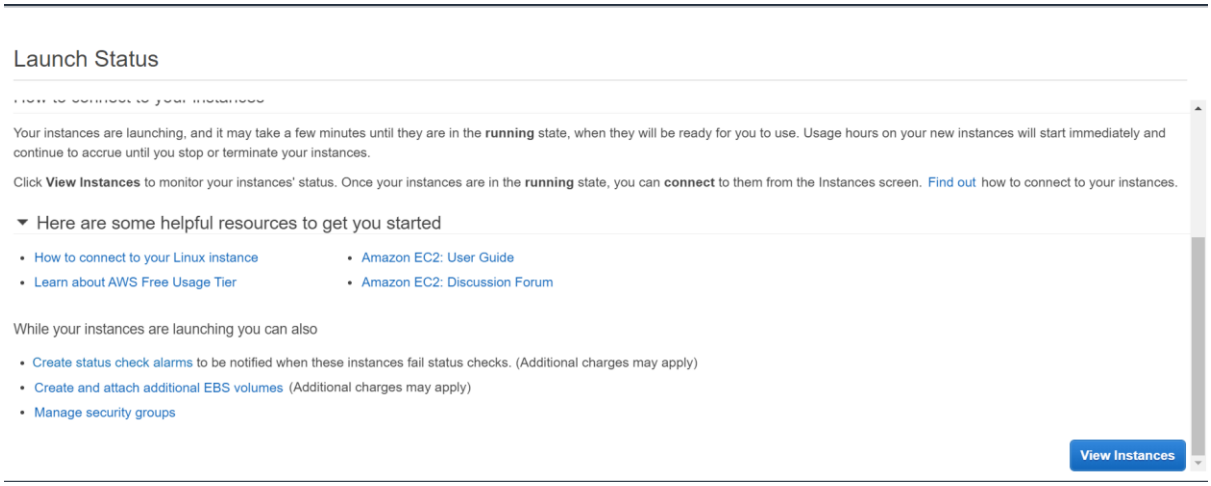


- Review the instance information and choose **Launch**

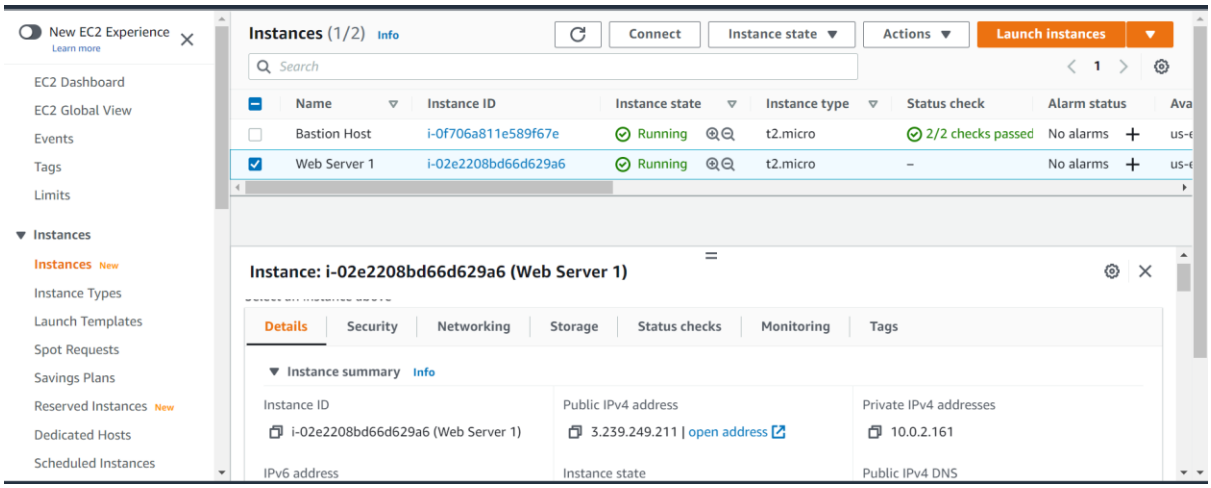


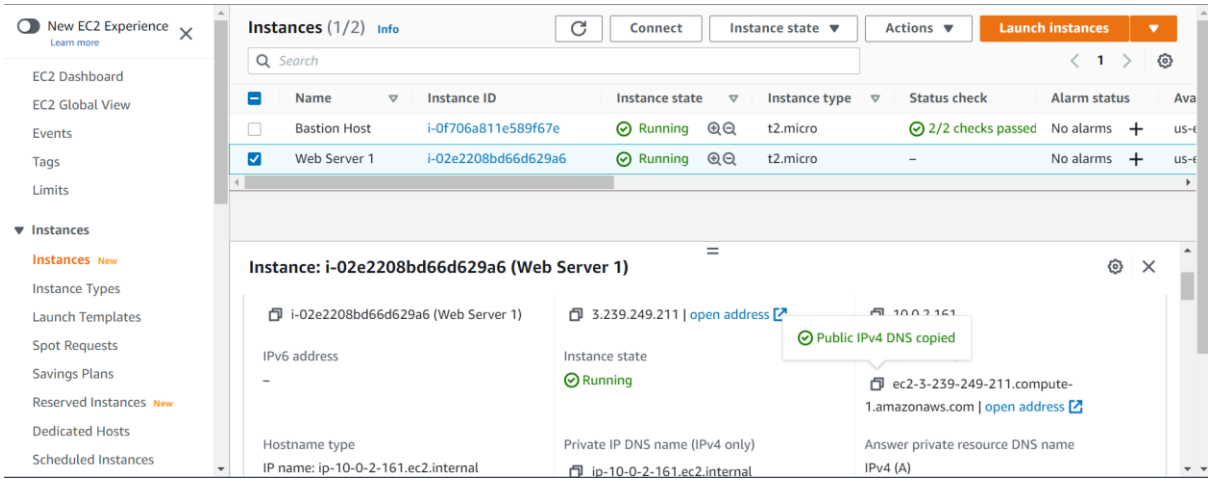
- In the **Select an existing keypair** dialog, select **I acknowledge...**.
- Choose **Launch Instances** and then choose **View Instances**

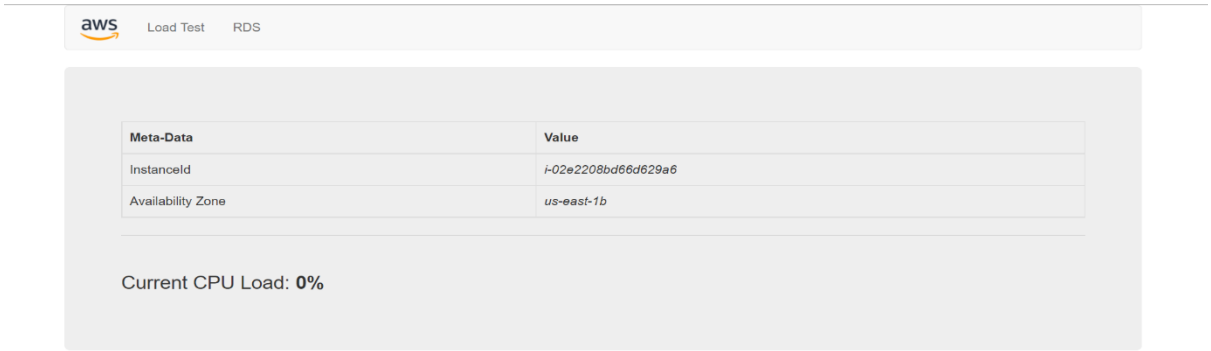




- Wait until **Web Server 1** shows *2/2 checks passed* in the **Status Checks** column.
- This may take a few minutes. Choose refresh in the top-right every 30 seconds for updates.
- You will now connect to the web server running on the EC2 instance.
- Select **Web Server 1**.
- Copy the **Public DNS (IPv4)** value shown in the **Description** tab at the bottom of the page.
- Open a new web browser tab, paste the **Public DNS** value and press Enter.

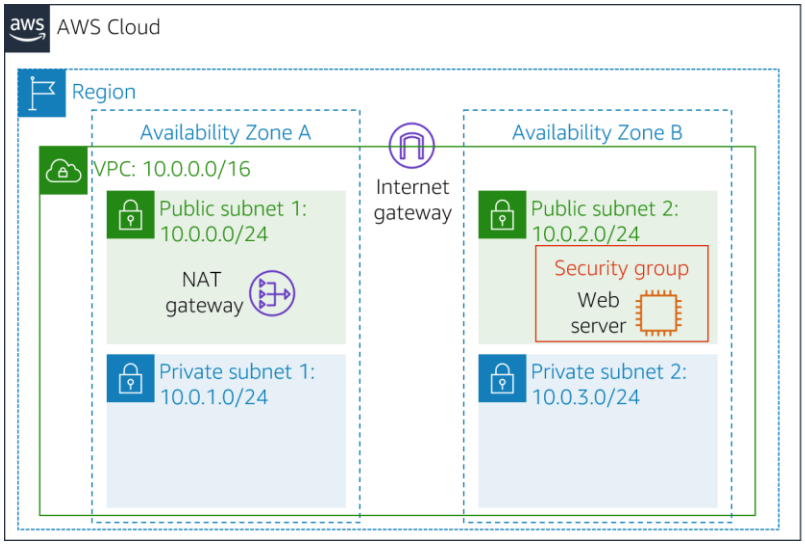






You should see a web page displaying the AWS logo and instance meta-data values.

The complete architecture you deployed is:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway