EJ Lilagan, Kyrstn Hall

Prof. Gutta

CS446 - Cloud Computing

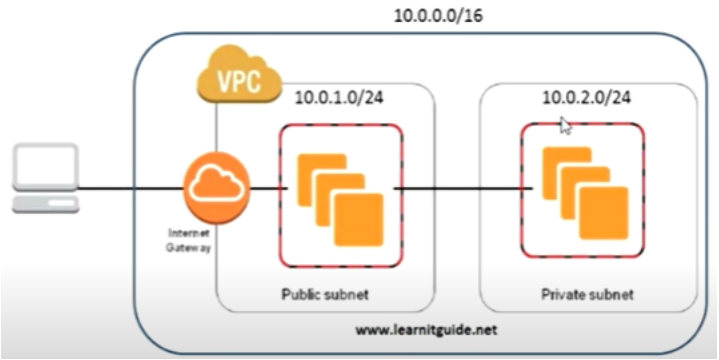
03 March 2023

AWS Virtual Private Cloud (VPC)

VPC will be successfully launched from the youtube video provided on canvas. To expand on the VPC, photos and summarizations are applied in this document. Before we show our demonstration of launching the VPC, here are its background of what VPC is/what it does. Further details that point up about VPC include the following:

- Virtual network that can be created in a public cloud given an AWS account with sufficient access
- Isolated from other virtual networks in AWS cloud based on its requirements with what has been created.
- It can specify an IP address range for the VPC by adding subnets and associating security groups and configure the route tables, in order to launch database resources, such as Amazon EC2 instances within the VPC.

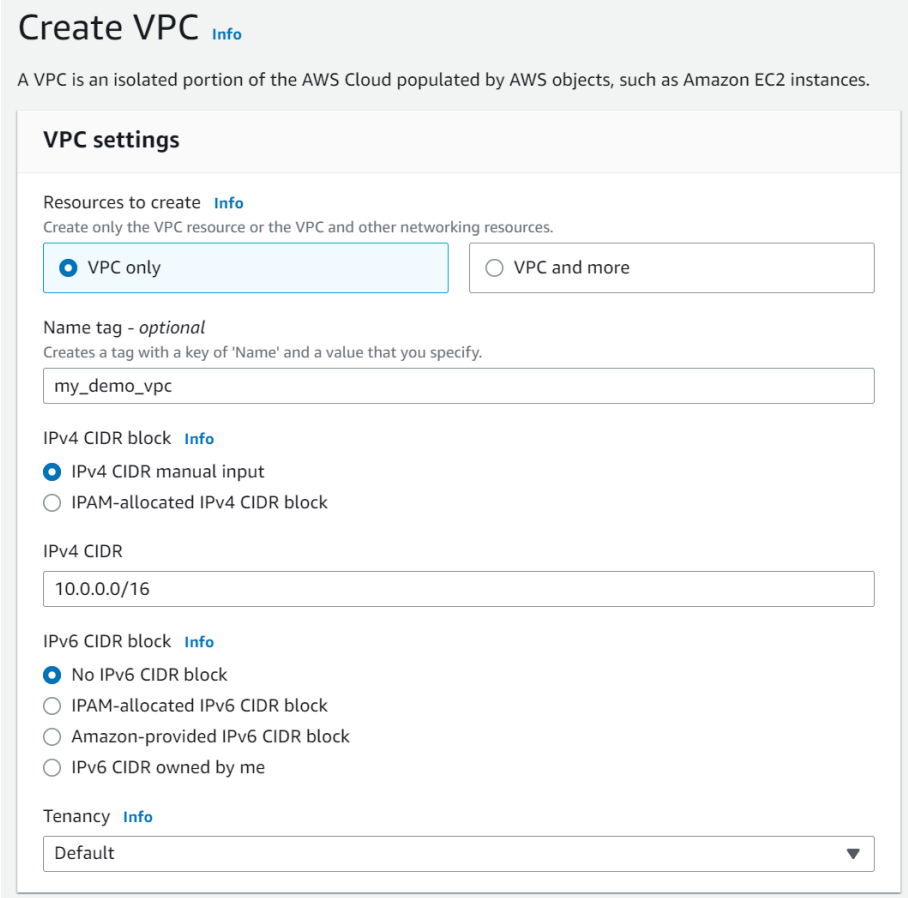
Given this photo we are trying to accomplish the following within the code



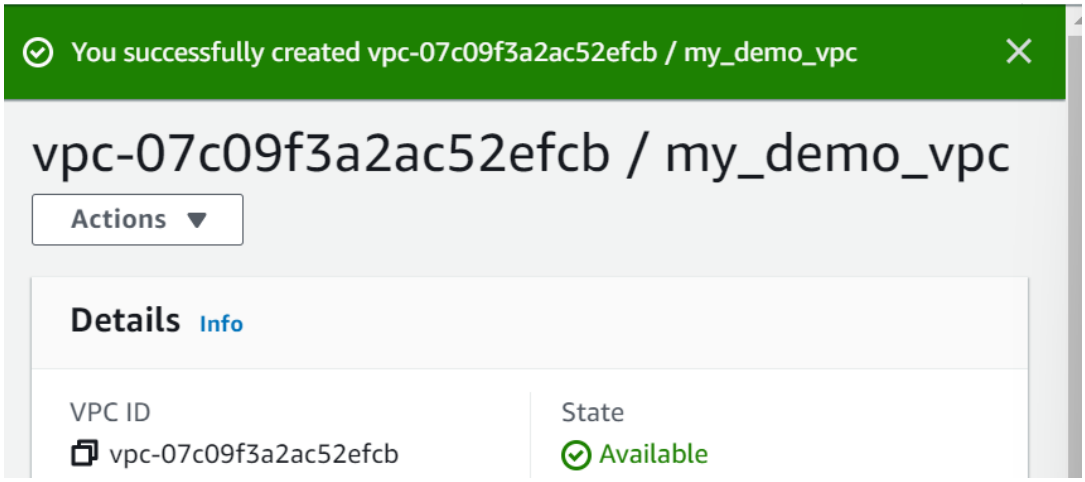
Checklist:

1. One VPC instance with IP of 10.0.0.0/16
2. Create two subnets inside the VPC
    a. Private with IP of 10.0.2.0/24 where its resource is 10.0.2
    b. Public with IP of 10.0.1.0/24 where its resource is 10.0.1
3. Public subnet must be attached to an internet gateway, which is where its resource will be obtained via the internet access.
4. On the other hand, private subnet will not have any access to the external world as its gateway is accessed from public subnet.

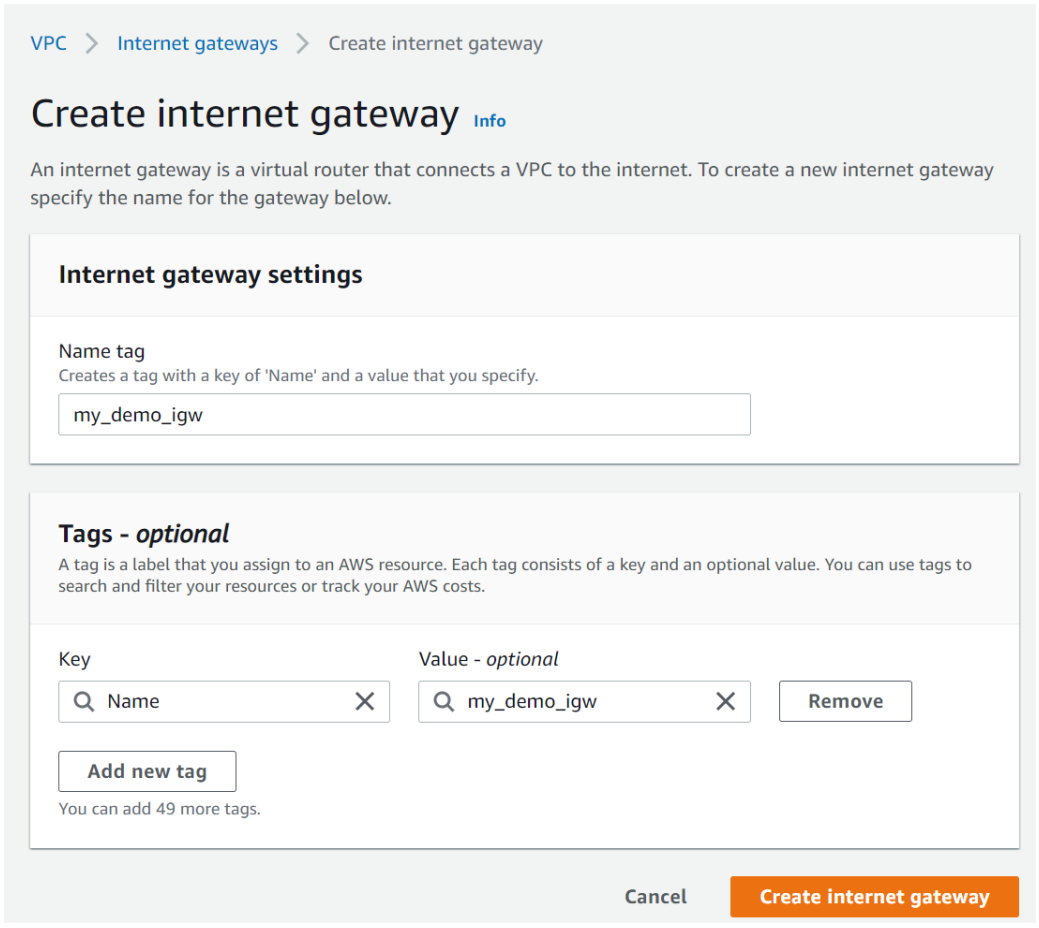
The following process starts below:



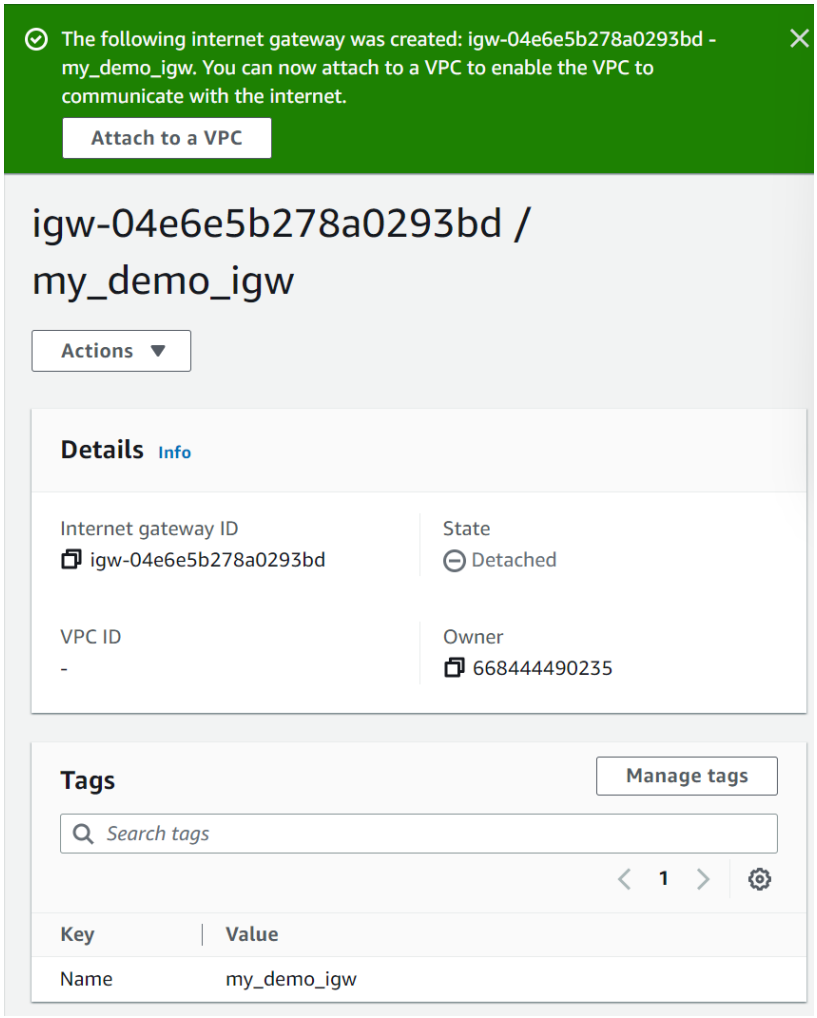
Creating a VPC given the following instructions



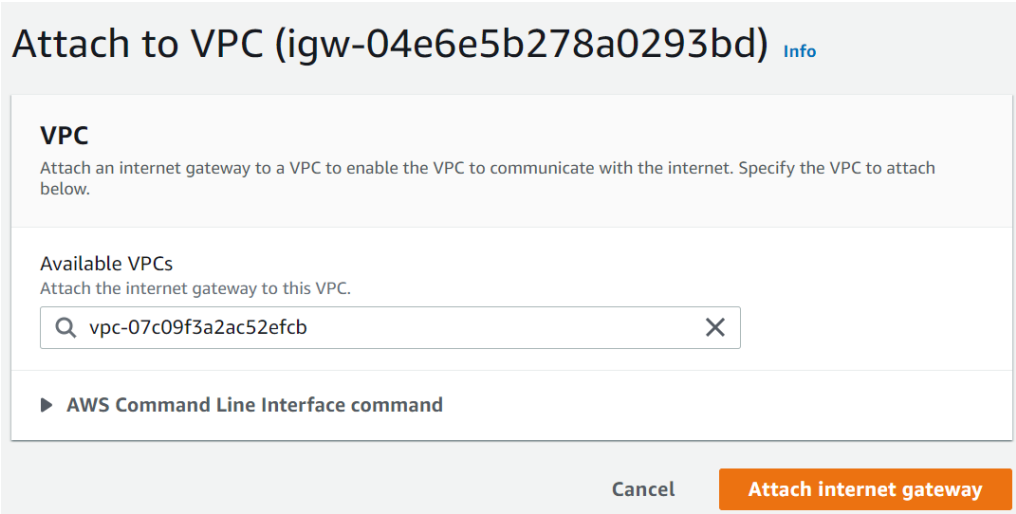
Pop up message that VPC has been created



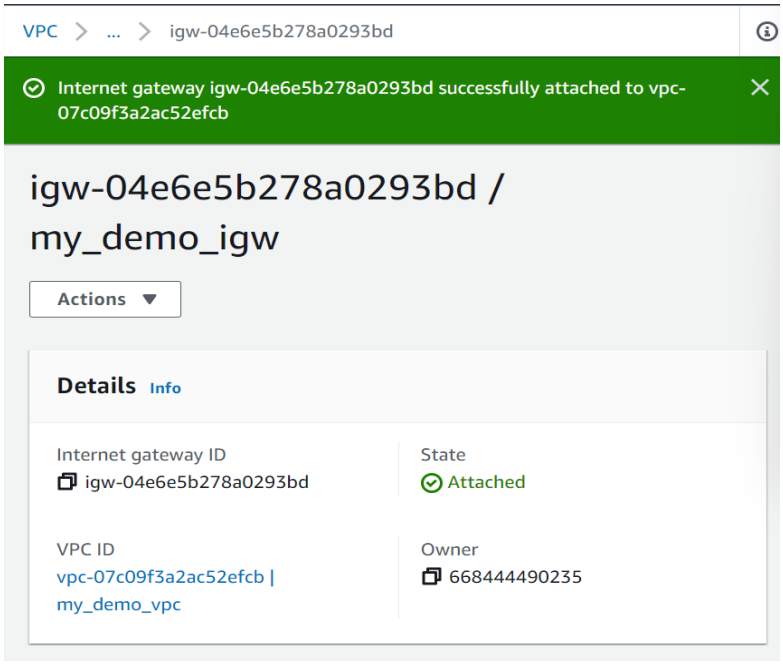
Creating an internet gateway for the public subnet



Showing the details that the internet gateway has been made without being attached to any subnets.



Selecting the VPC to attach with the internet gateway.



Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

vpc-07c09f3a2ac52efcb (my_demo_vpc)

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Public Subnet

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference

IPv4 CIDR block Info

10.0.1.0/24

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

vpc-07c09f3a2ac52efcb (my_demo_vpc)

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Private Subnet

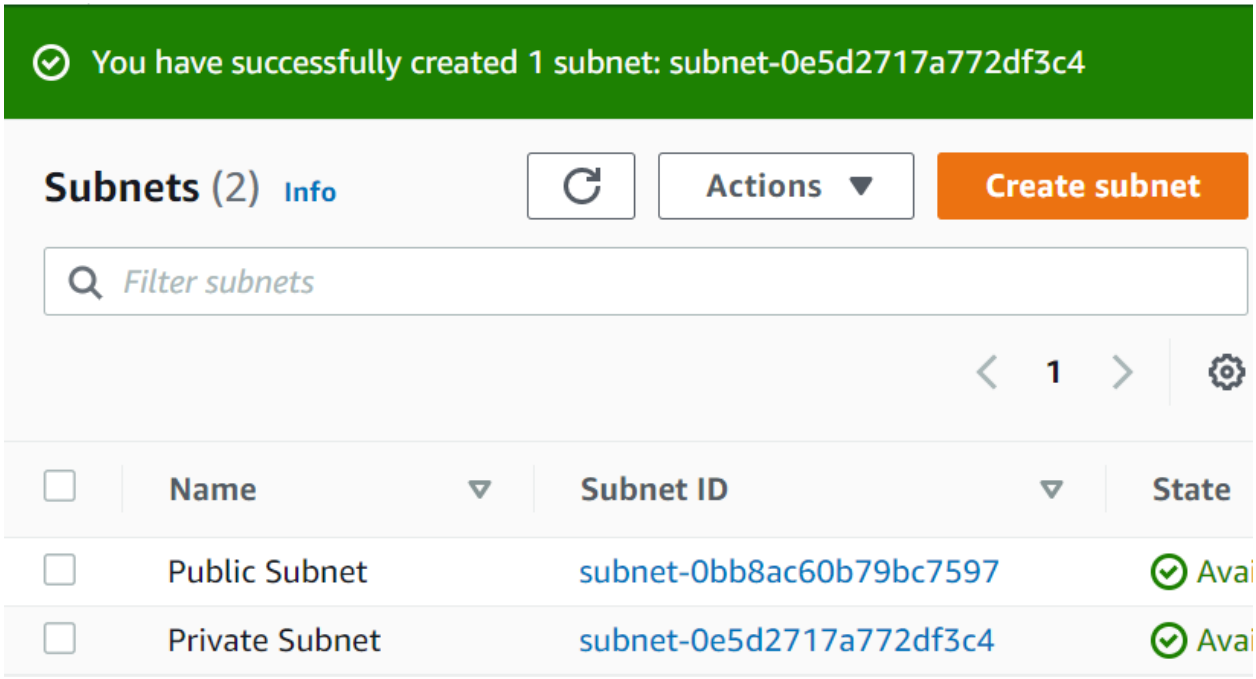
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference

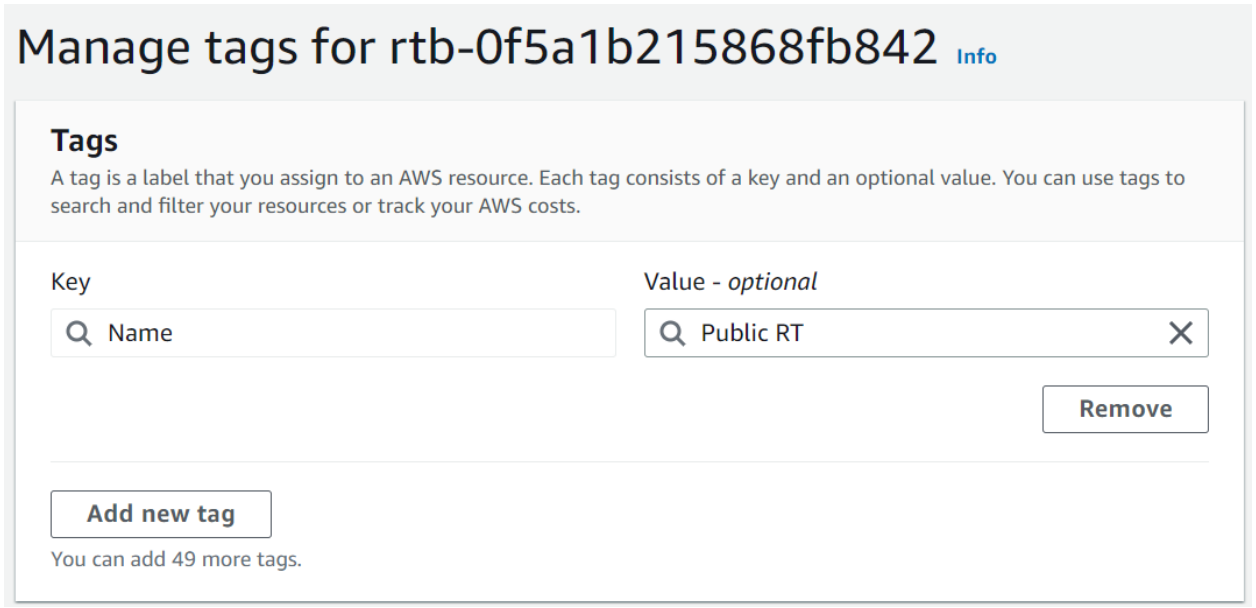
IPv4 CIDR block Info

10.0.2.0/24



Creating both public and private subnets that are created within the VPC.

– Making route tables for both subnets –



Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.

Private RT

VPC
The VPC to use for this route table.

vpc-07c09f3a2ac52efcb (my_demo_vpc)

Route table rtb-0c87bca621e206380 | Private RT was created successfully.

Route tables (2) Info

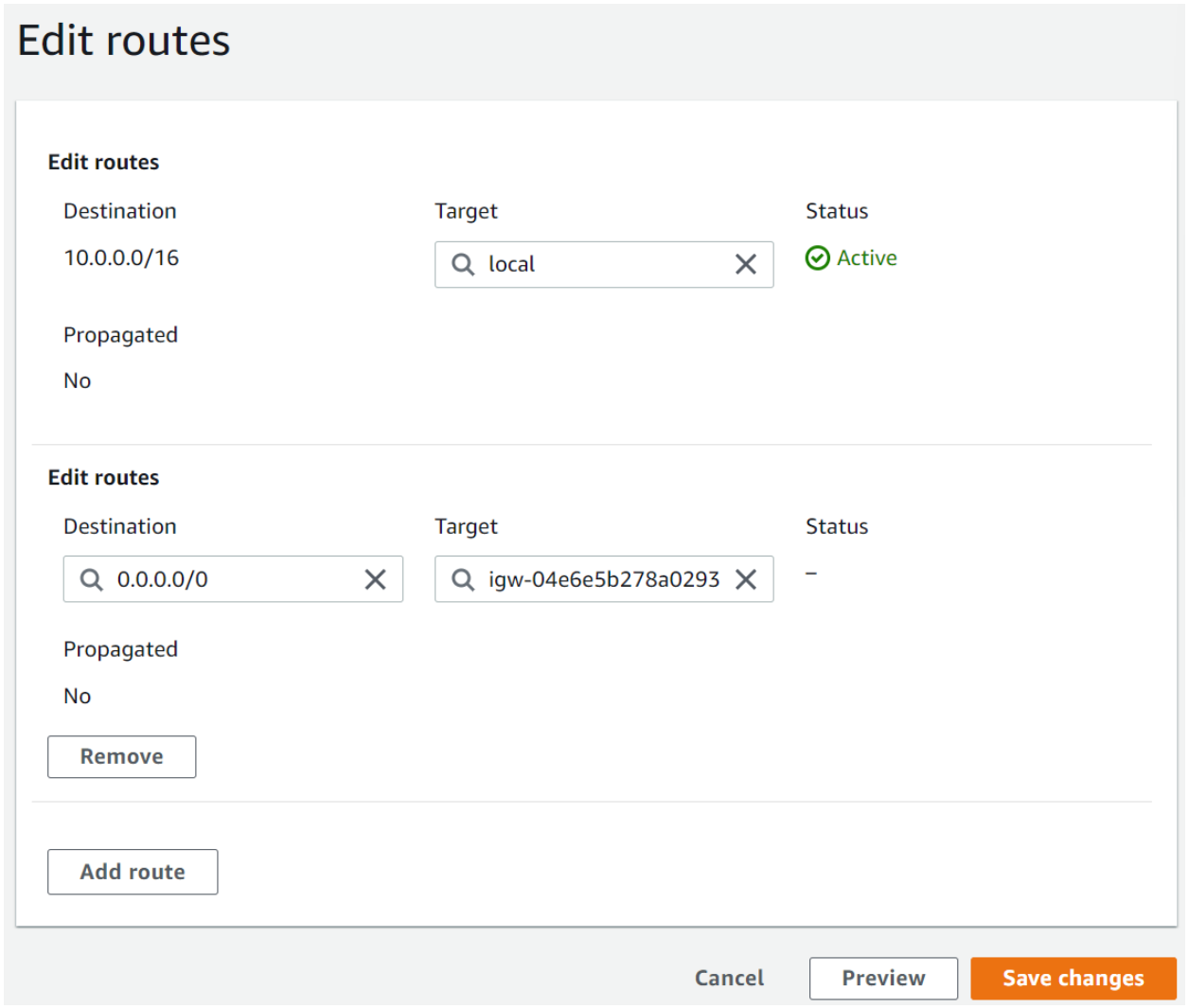
Actions

Create route table

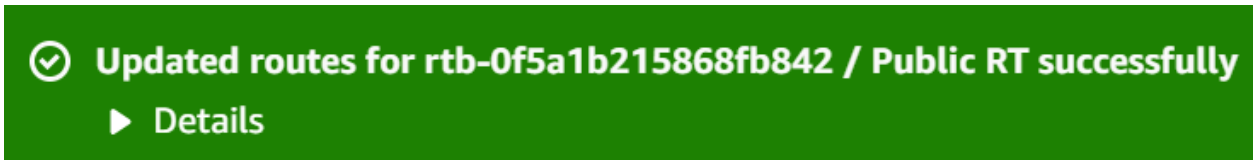
Filter route tables

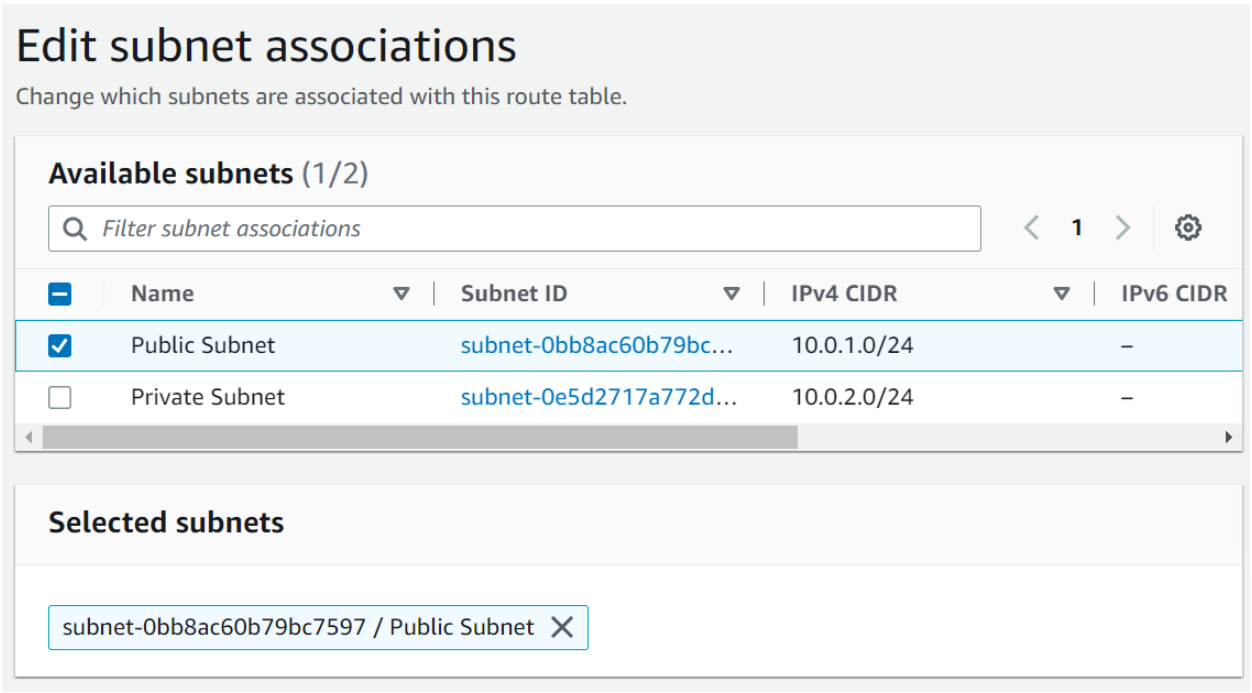
1

Name	Route table ID	Explicit subne
Private RT	rtb-0c87bca621e206380	–
Public RT	rtb-0f5a1b215868fb842	–

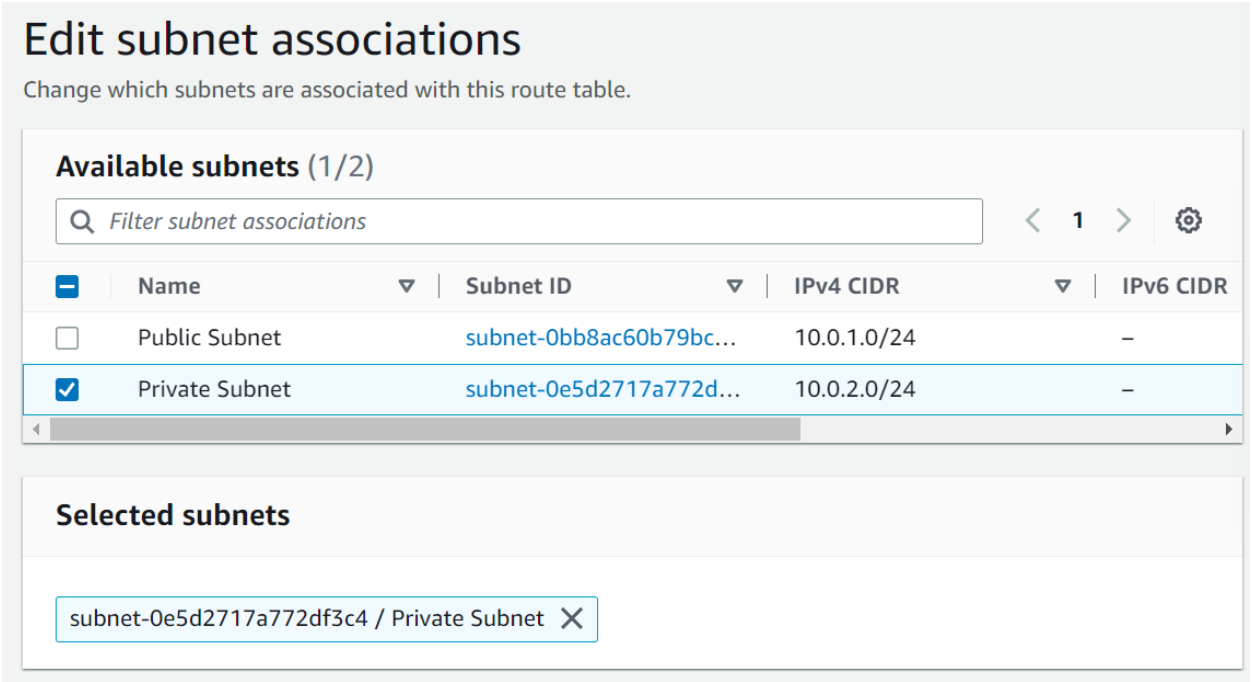


Searched target by clicking on “Internet Gateway” instead of typing “igw”

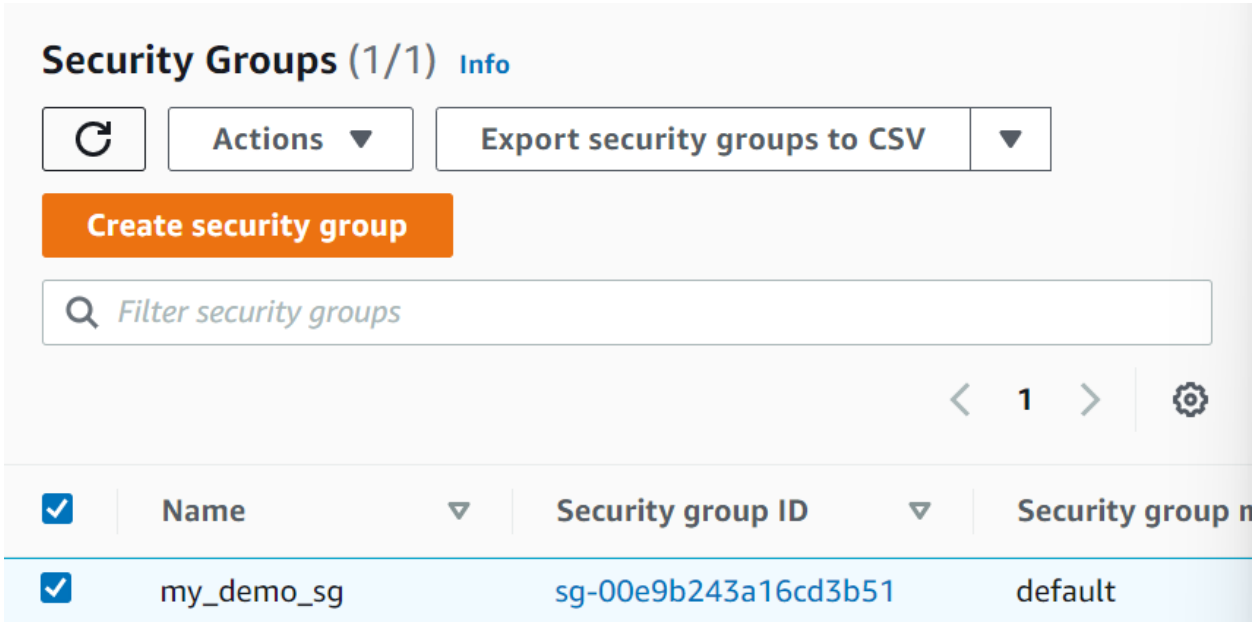




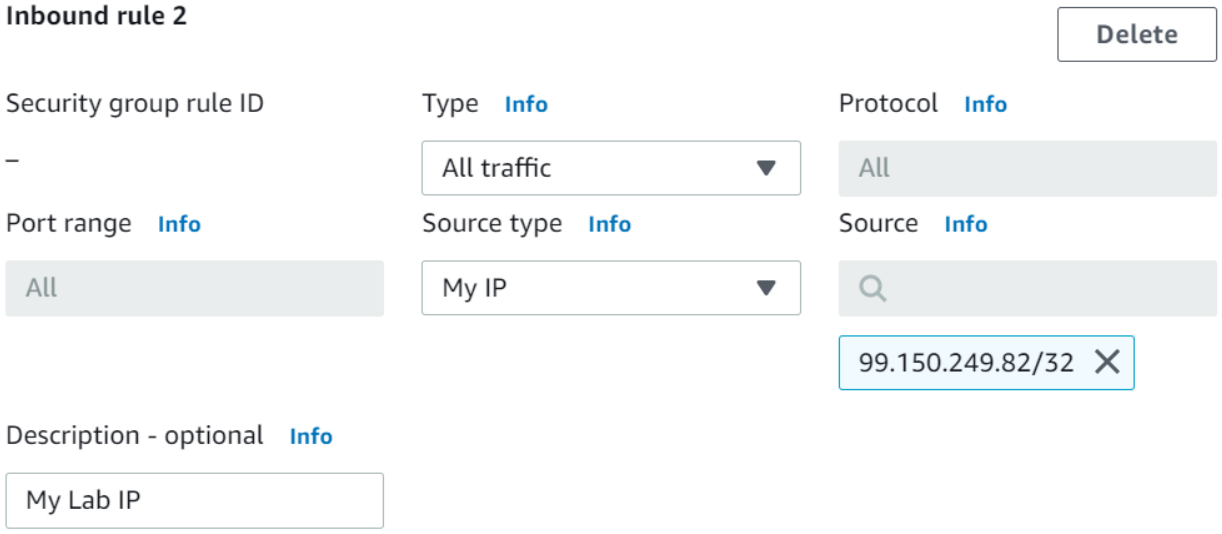
Connecting public subnet to the Public Route Table



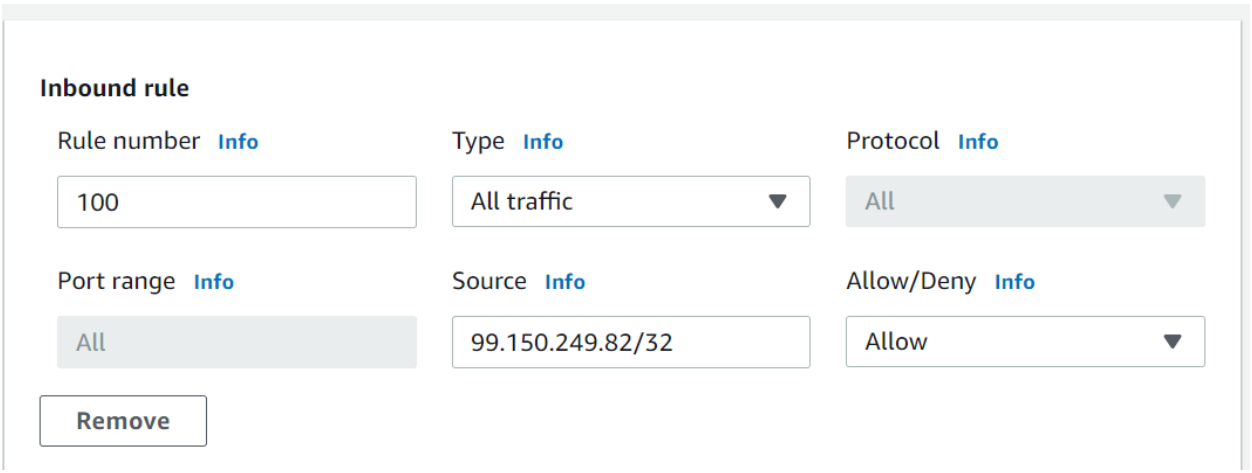
Same process again but for associating Private Subnet with Private Route Table



Create security group to protect resources from accessing the internet

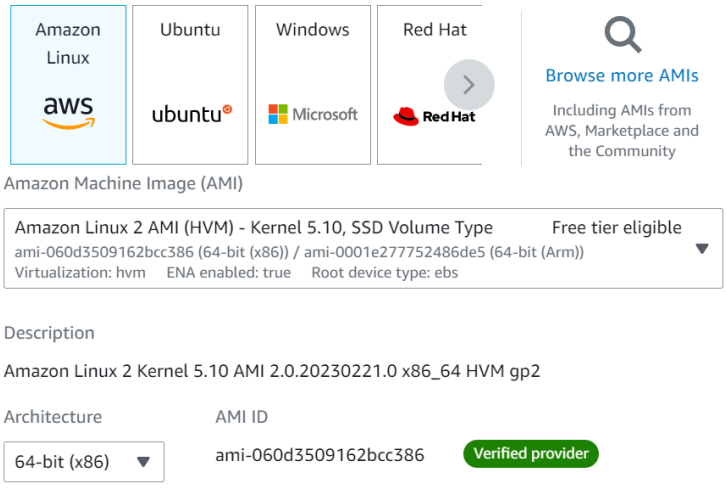


Creating an inbound rule

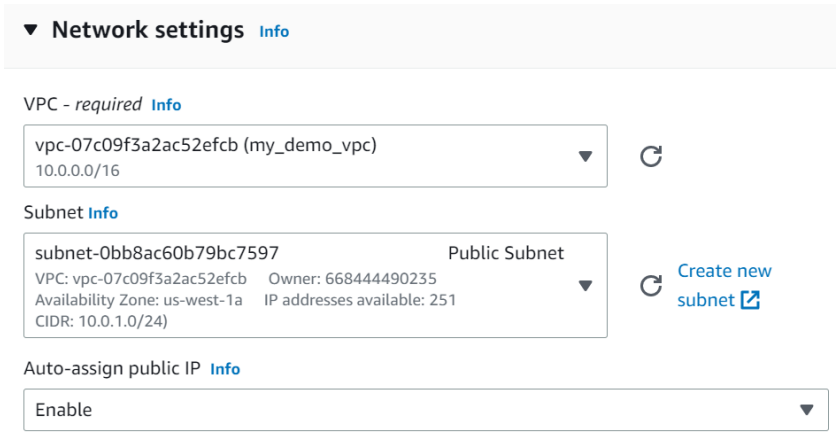


Edit inbound rule for the Network ACL's

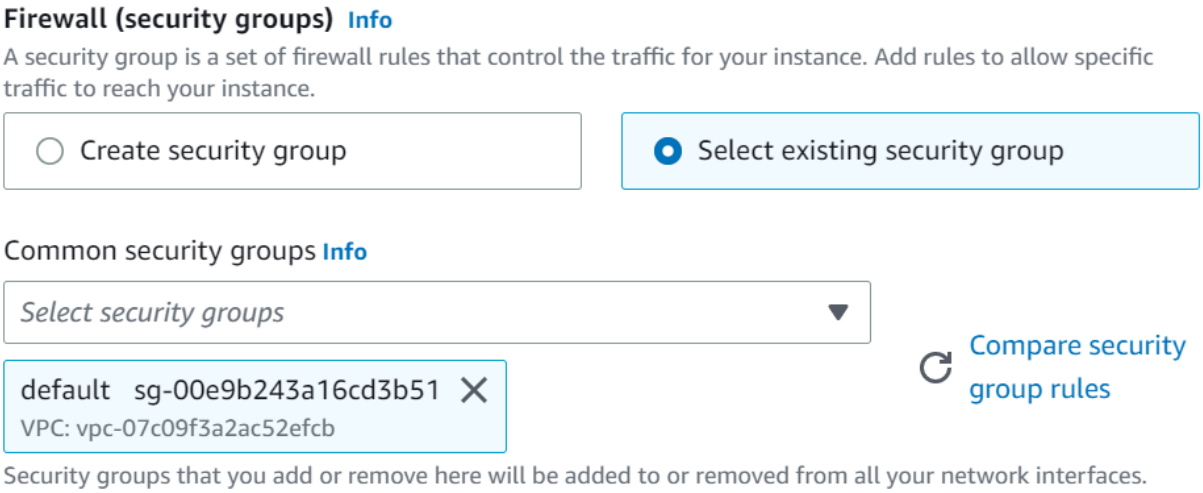
–Transition to EC2 instance–



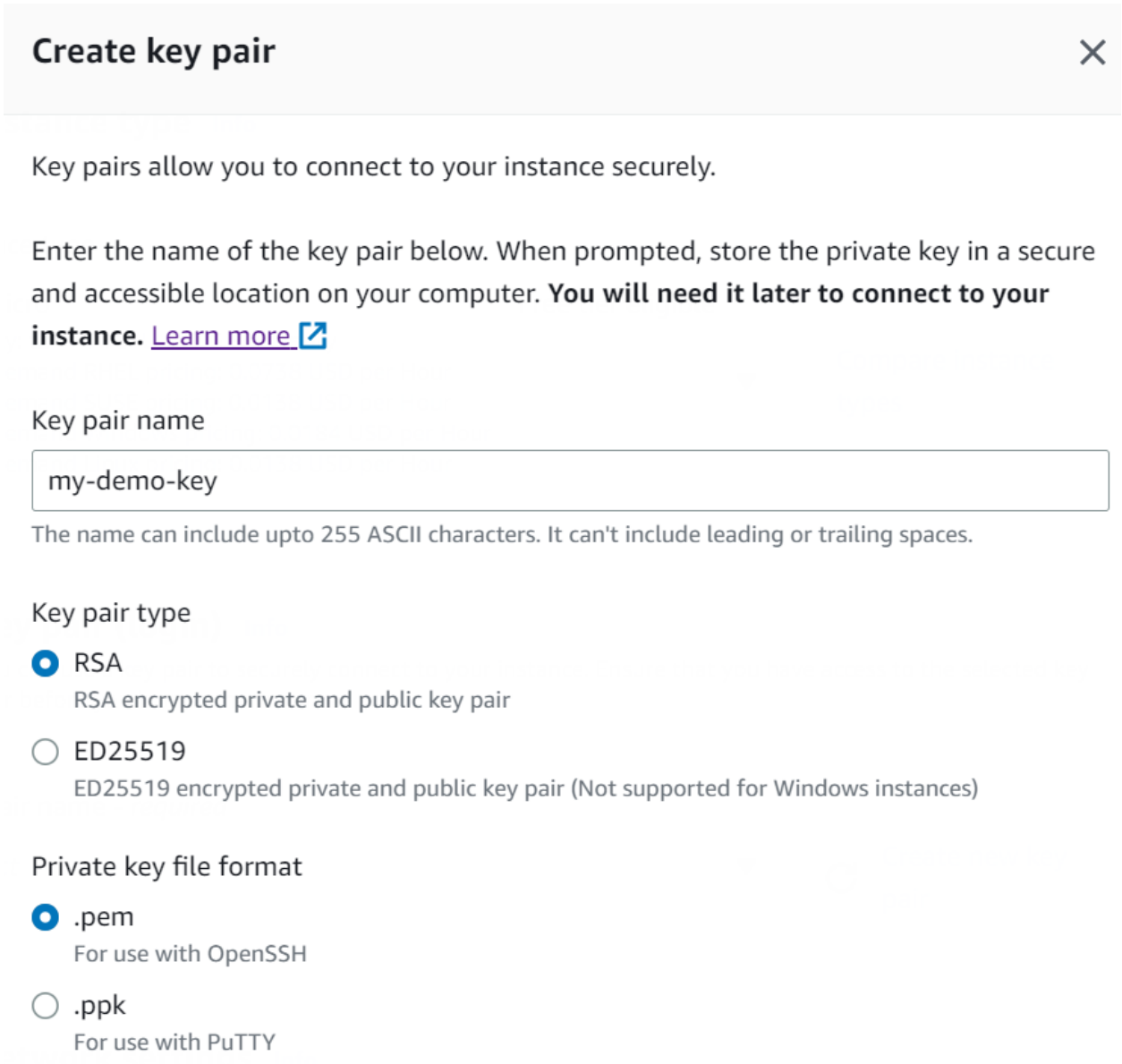
Setting AMI to Amazon Linux 2 (free tier)



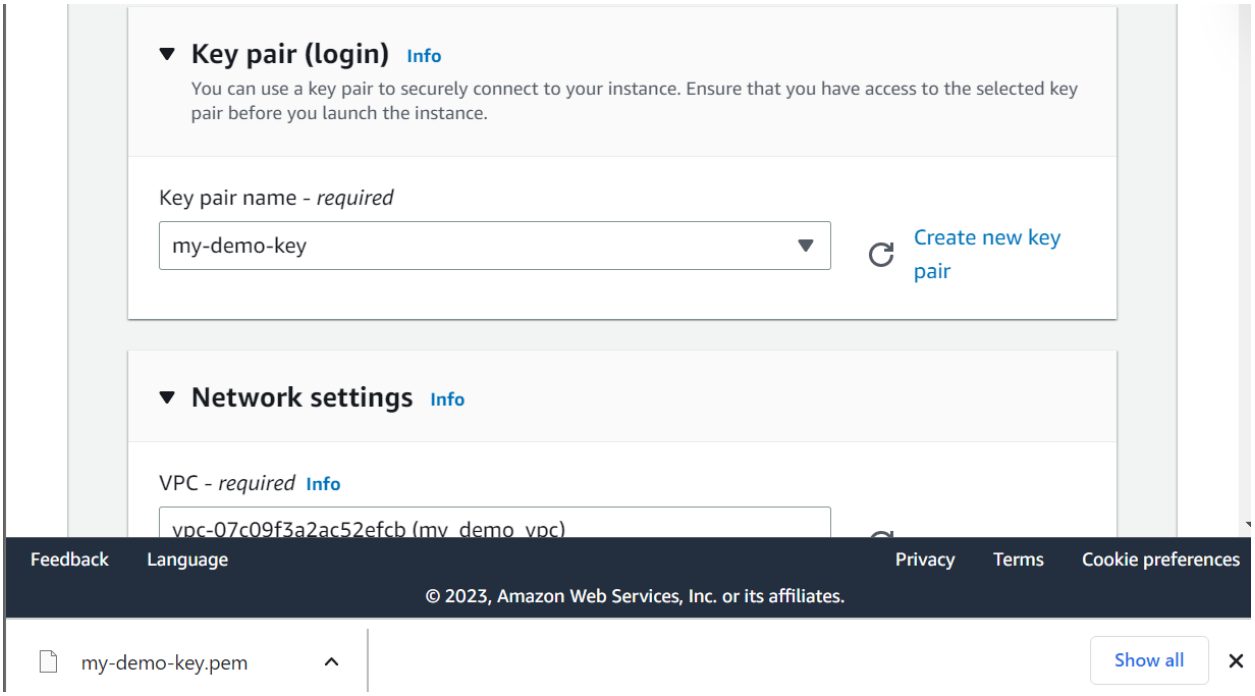
Network settings included both network and subnet and enabling the public IP



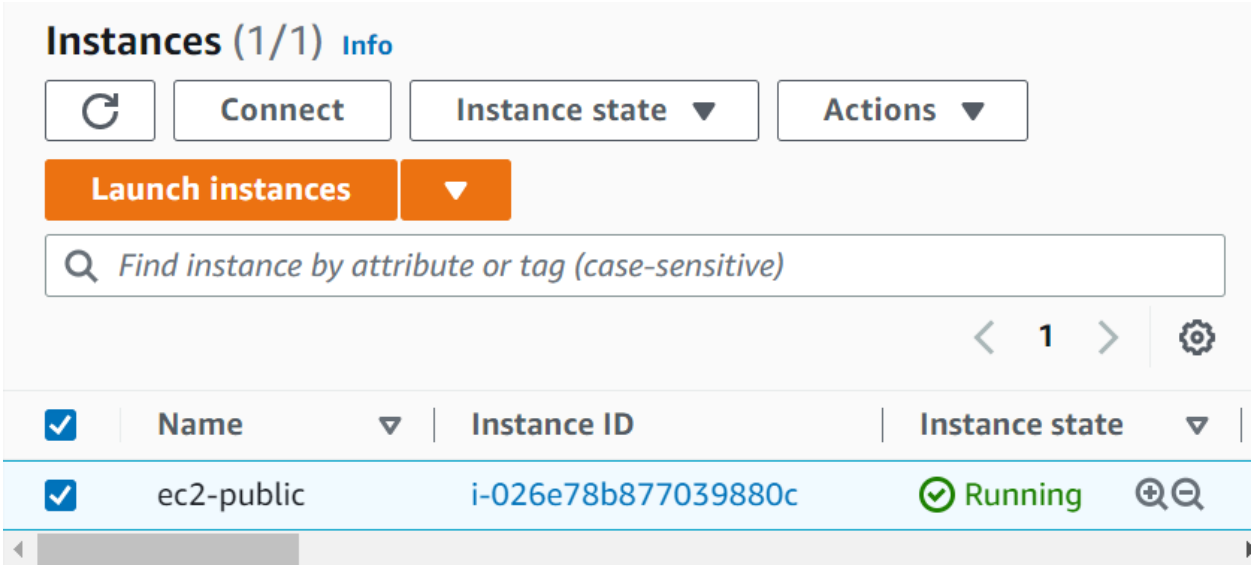
Select security group to existing, and select common security group to be the default



Creating a new key pair for using VPC

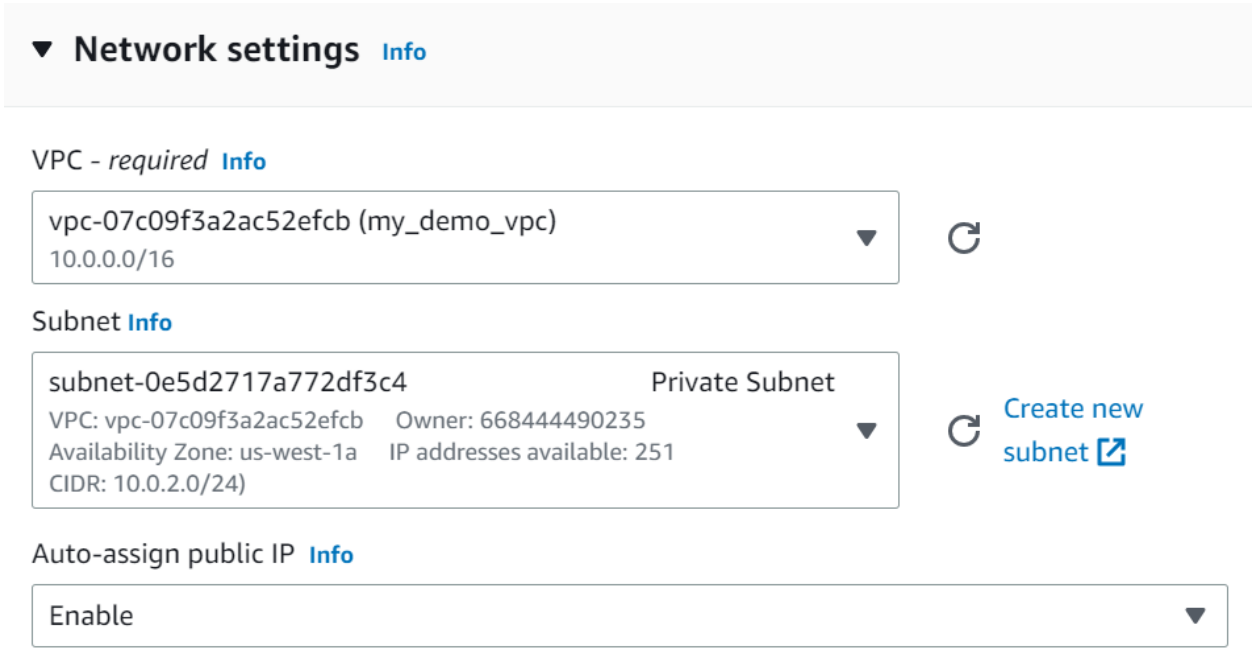


Use the download for this demo

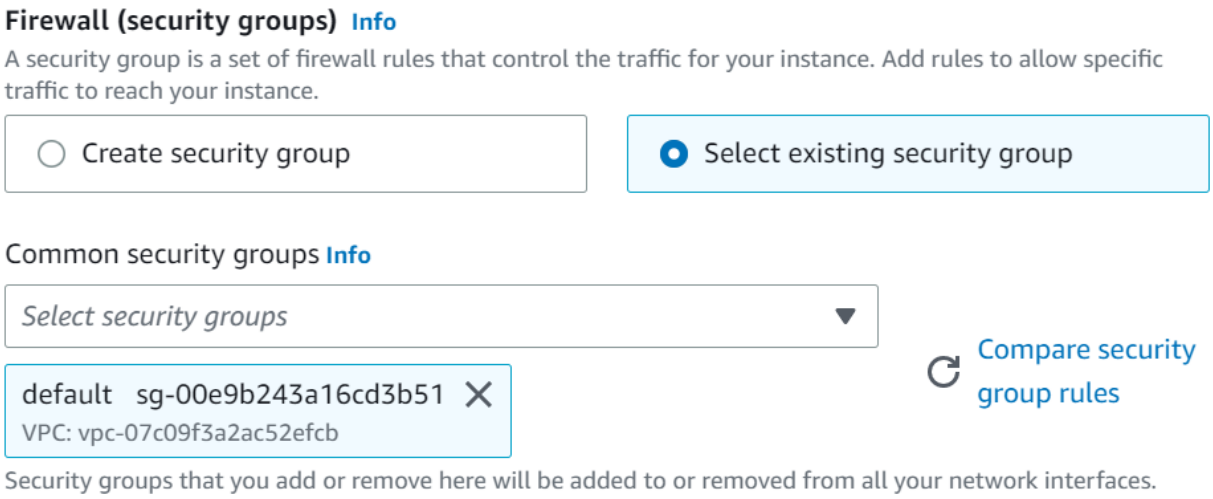


Once launched, check if on running state and attach instance to public subnet

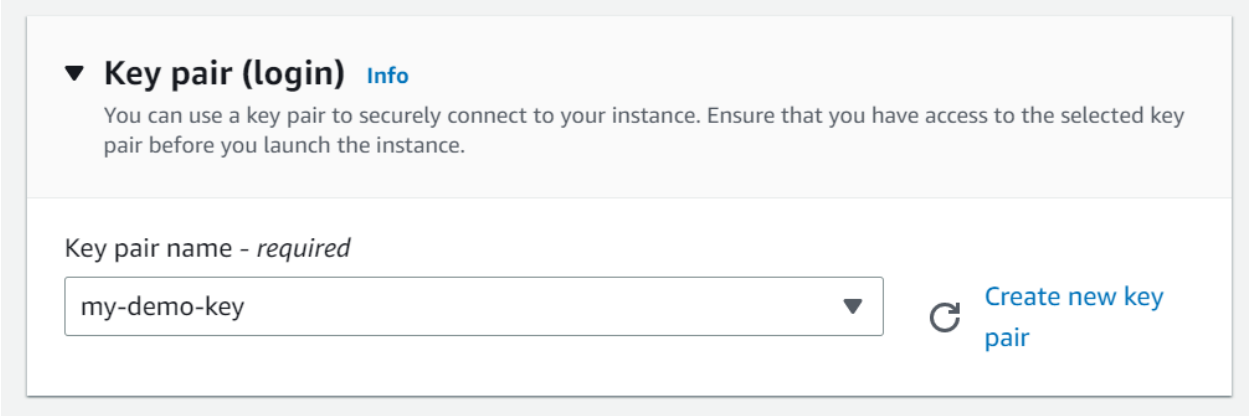
–Same process again but for private subnet–



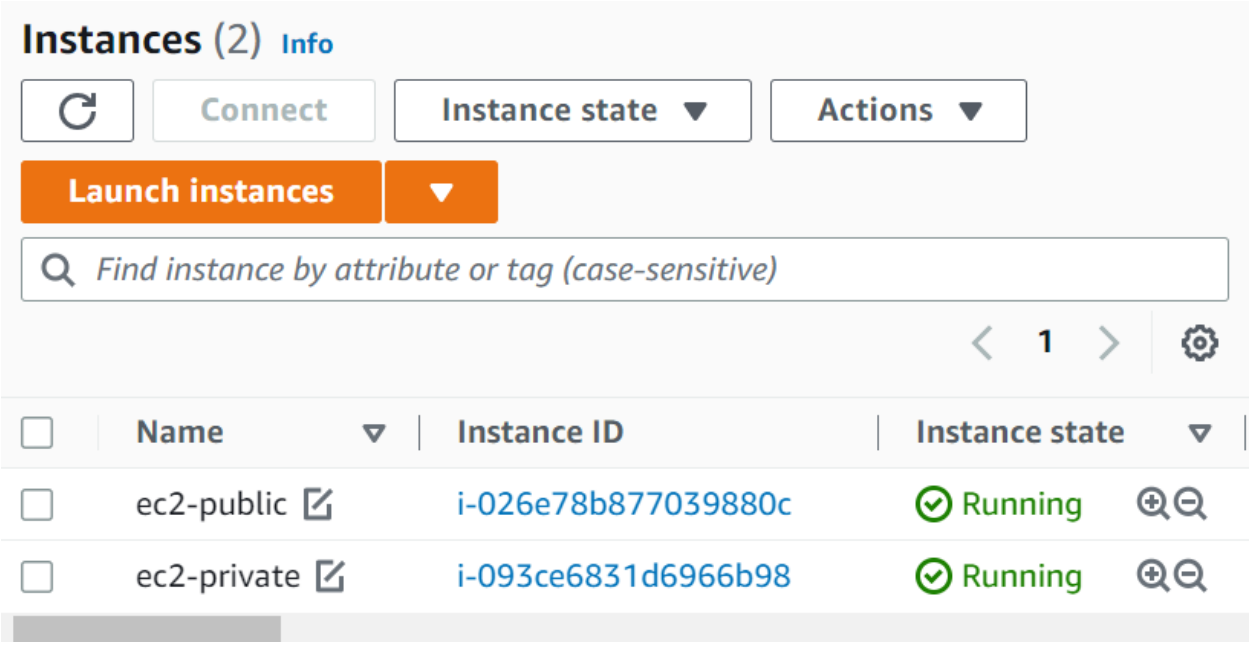
Do the same but change the subnet for the private subnet.



Same as to public subnet security configuration

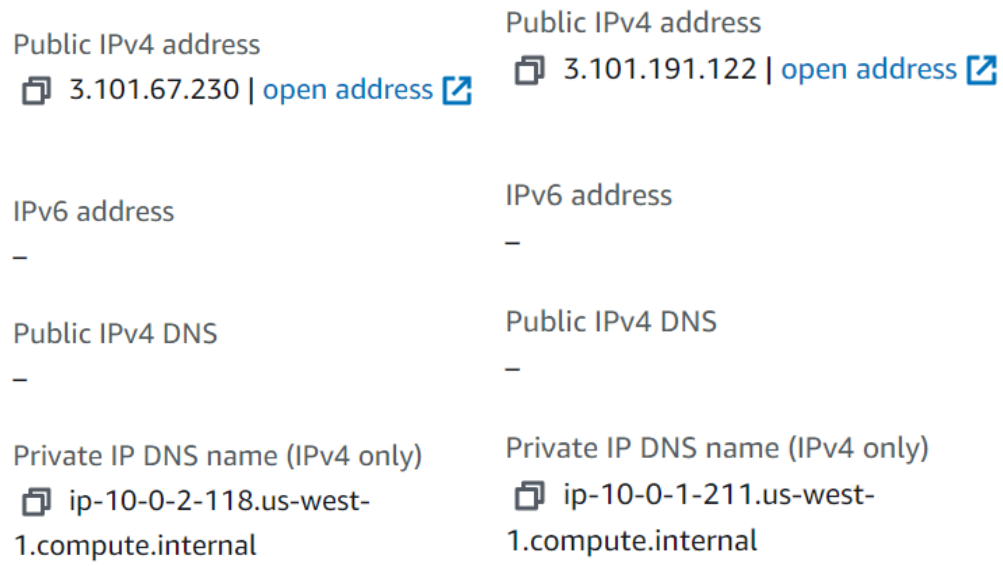


Use existing key pair that was created from public subnet instance



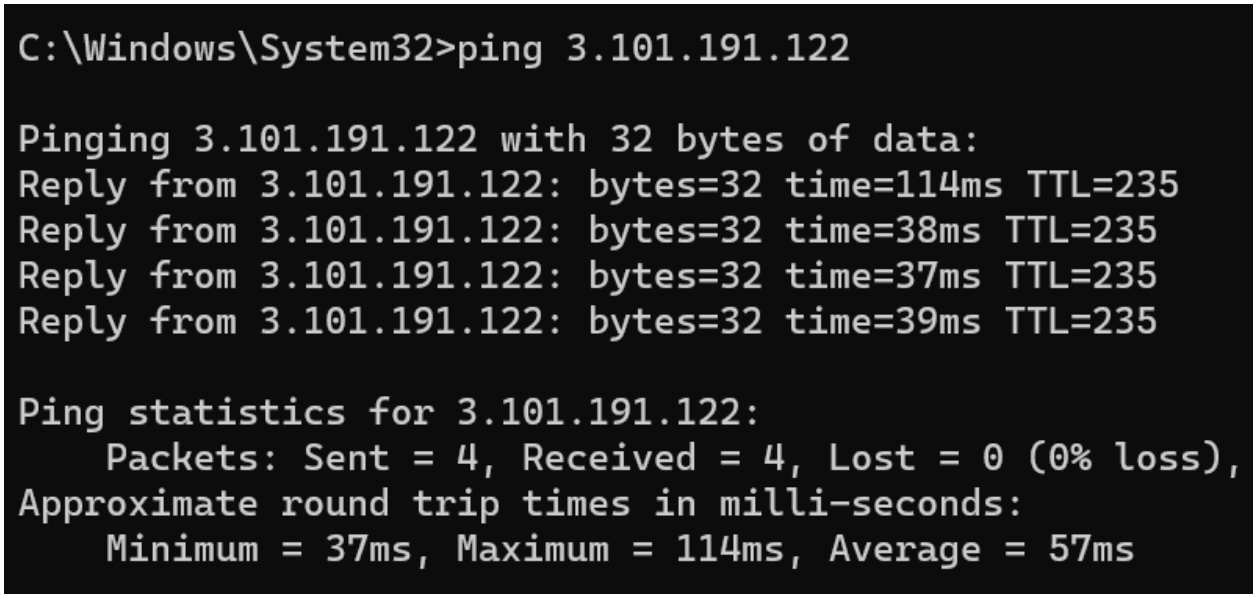
Proof to show public/private were successfully launched

Key differences between both public and private EC2 include the Public IPv4 and private IP

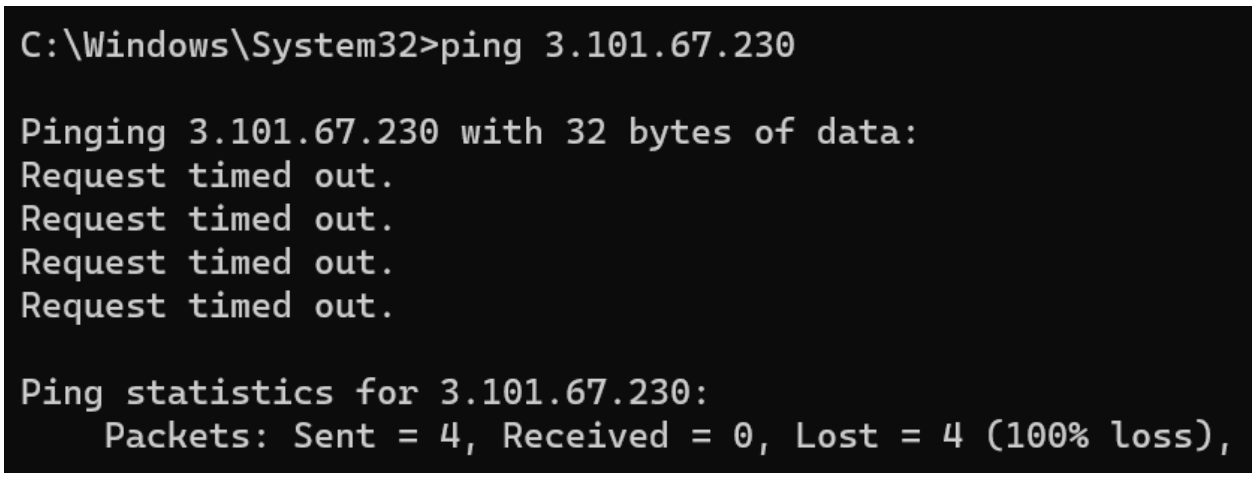



```
C:\Windows\System32>ping 3.101.191.122

Pinging 3.101.191.122 with 32 bytes of data:
Reply from 3.101.191.122: bytes=32 time=114ms TTL=235
Reply from 3.101.191.122: bytes=32 time=38ms TTL=235
Reply from 3.101.191.122: bytes=32 time=37ms TTL=235
Reply from 3.101.191.122: bytes=32 time=39ms TTL=235

Ping statistics for 3.101.191.122:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 37ms, Maximum = 114ms, Average = 57ms
```

Open command prompt and type PING.EXE to see if it is located. Had to go into my windows directory to get the ping command to work.



This photo pings the EC2 from the public EC2



This ping has been timed out due to the private subnet not being accessed to the internet gateway.

Challenges/Thoughts: Me and Kyrstn ended up having difficulties starting the demo when following the instructions on the YouTube video. Although the formats were different from current AWS and the video, it was a challenge from launching certain number of subnets/route tables along the way. What has happened was when a subnet was launched the first time, it would save but then the second time the subnet was created, it would not display the first subnet due to filters being activated, which led me to have around 4 subnets in total, and was a hassle to

know which subnet to use when attaching it to the internet gateway, thus led the both of us to restart the process from the very start of the video. A lesson that it both taught us was to not watch the video without volume that ended up being costly, and to pay attention to every detail. Another challenge was that the command ping would be difficult to run on the command prompt until we had to call PING.EXE in one of the windows 32 directory in order to run the IP address that would run the EC2 instance from the public subnet since it is connected to the internet gateway and gave results whereas the private subnet would run but not including any replies due to its resources not directly be allocated to the internet gateway.