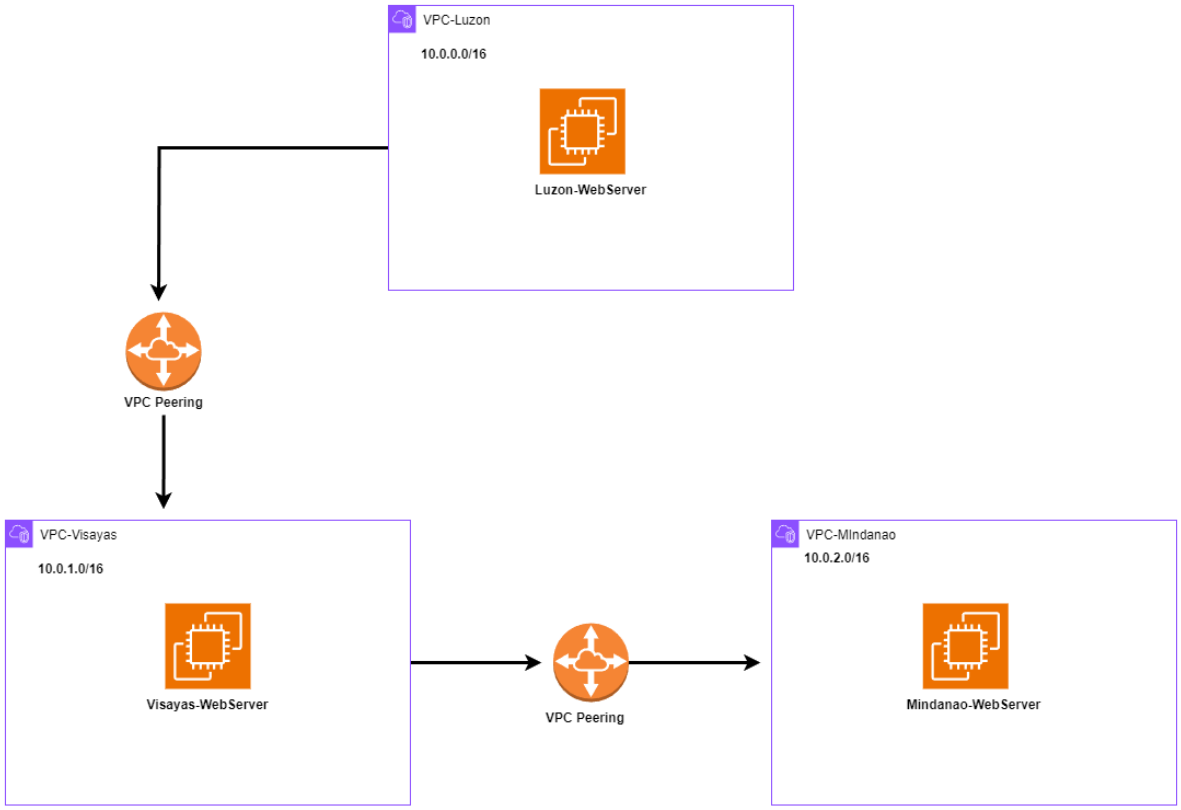
Guided Lab: Setting up VPC Peering

Description

Amazon Virtual Private Cloud (VPC) Peering allows private networking connections between different VPCs, enabling resources in separate VPCs to communicate as if they are within the same network. This lab will guide you through the process of creating and testing VPC peering connections using three distinct VPCs, each hosting an Amazon EC2 instance. This setup will help you understand the network connectivity and security configurations necessary for VPC peering.



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 a Custom Virtual Private Cloud (VPC) from scratch**
- **Creating an Amazon EC2 instance (Linux)**
- **Setting up a Web server on an EC2 instance**

Objectives

In this lab, you will:

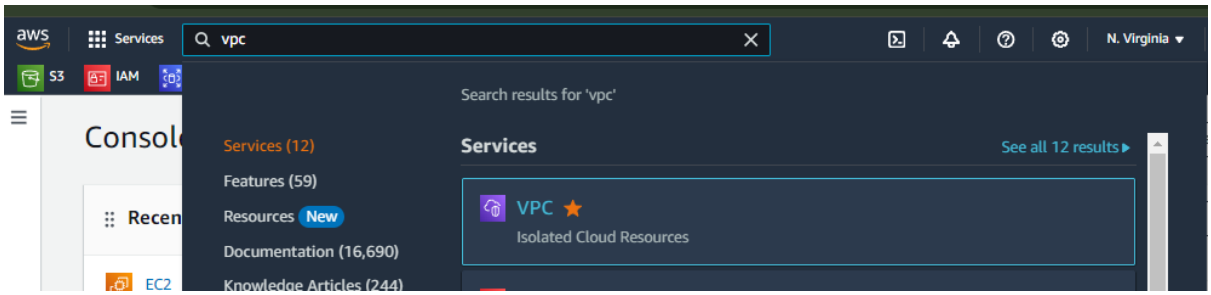
- **Learn how to configure VPC peering between multiple VPCs.**

- **Demonstrate inter-VPC communication through EC2 instances.**
- **Understand the security and routing implications of VPC peering.**

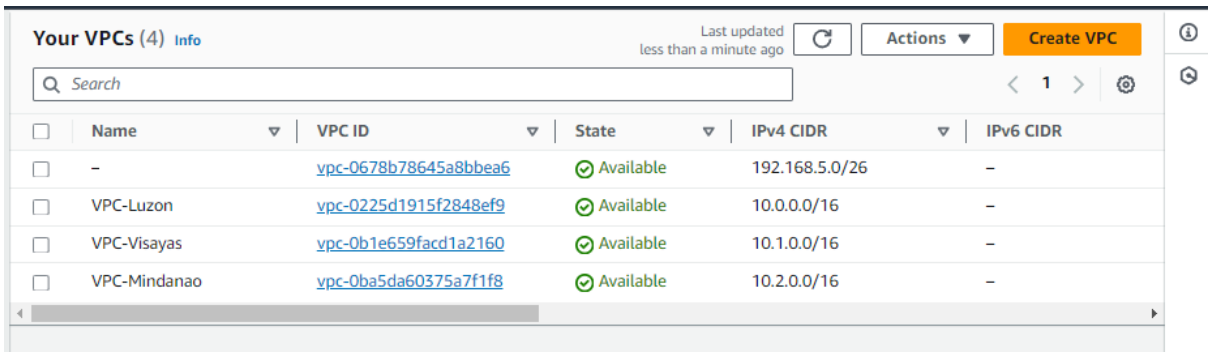
[Subscribe to access AWS PlayCloud Labs]

Lab Steps

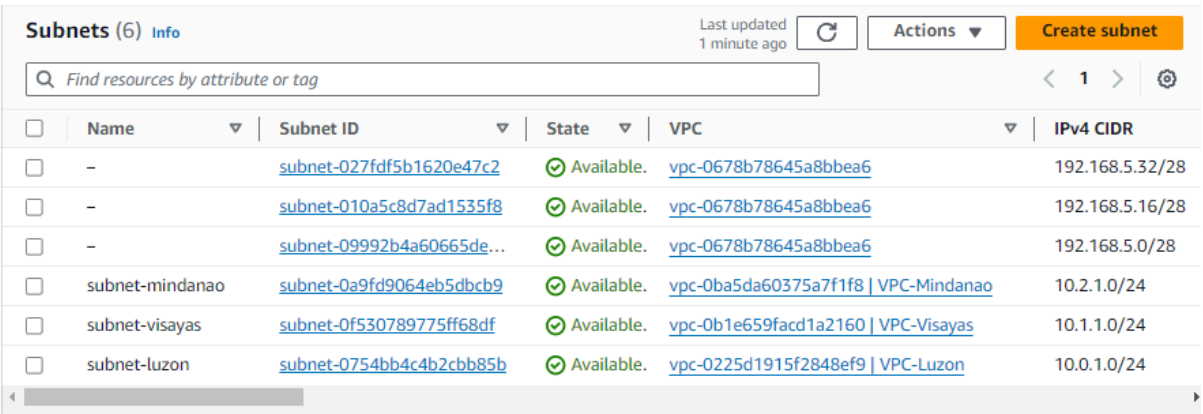
Preparing your environment



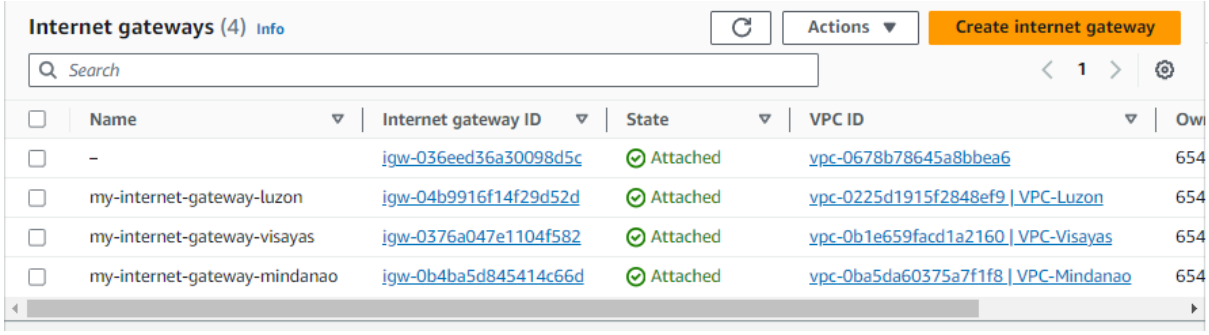
-
-
-



-
-
-



-
-
-



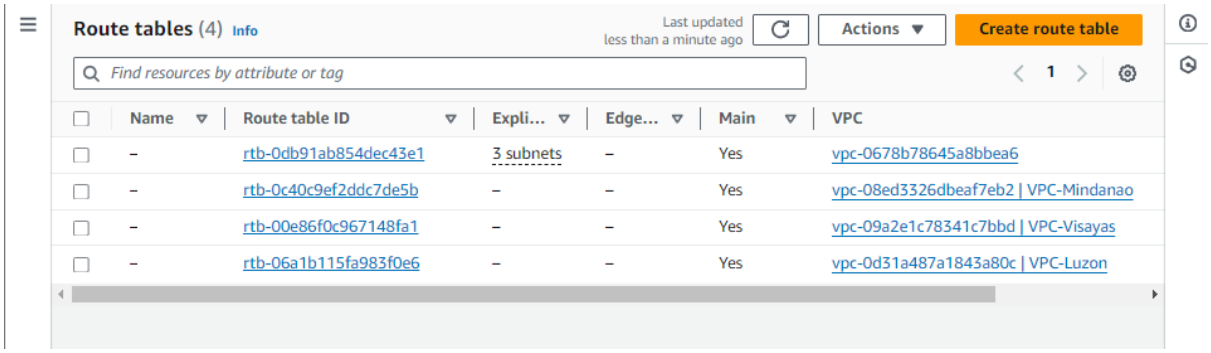
Internet gateways (4) Info

Actions | Create internet gateway

Search

1

Name	Internet gateway ID	State	VPC ID	Ow
–	igw-036eed36a30098d5c	Attached	vpc-0678b78645a8bbea6	654
my-internet-gateway-luzon	igw-04b9916f14f29d52d	Attached	vpc-0225d1915f2848ef9 | VPC-Luzon	654
my-internet-gateway-visayas	igw-0376a047e1104f582	Attached	vpc-0b1e659facd1a2160 | VPC-Visayas	654
my-internet-gateway-mindanao	igw-0b4ba5d845414c66d	Attached	vpc-0ba5da60375a7f1f8 | VPC-Mindanao	654



Route tables (4) Info

Last updated less than a minute ago

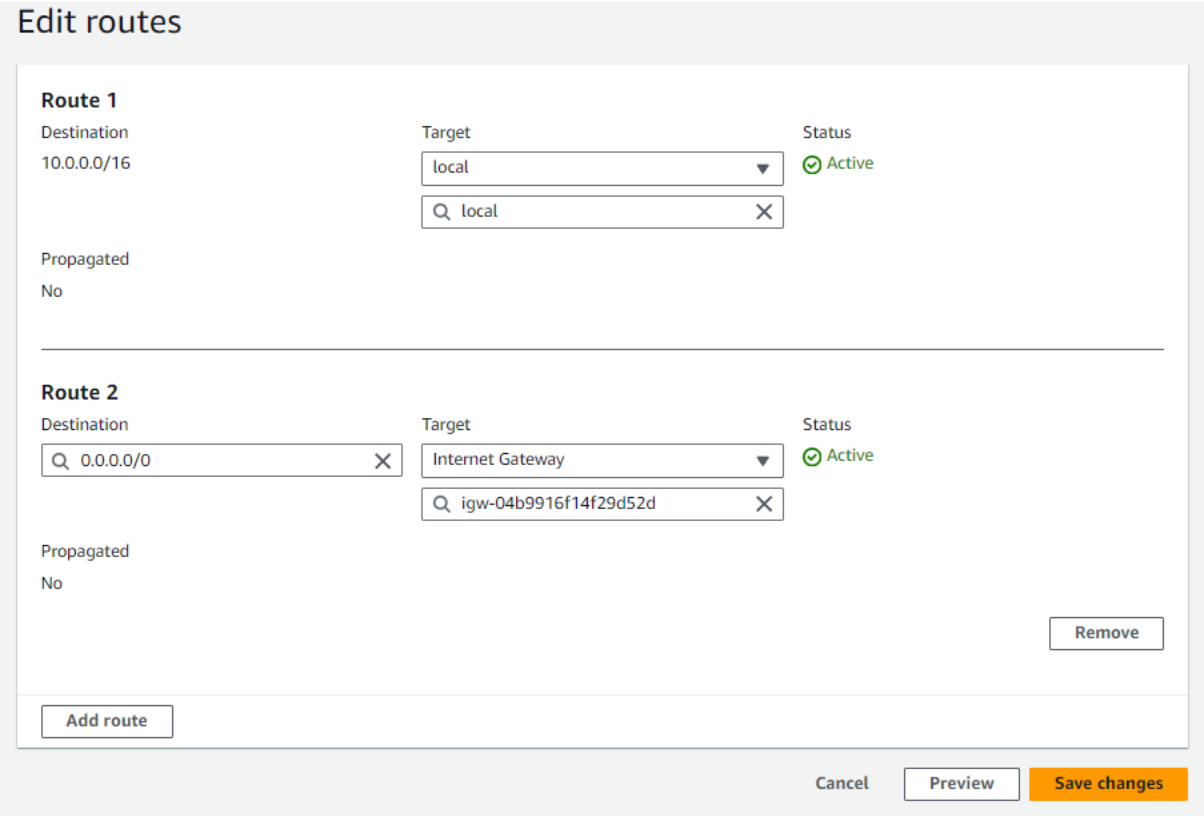
Actions | Create route table

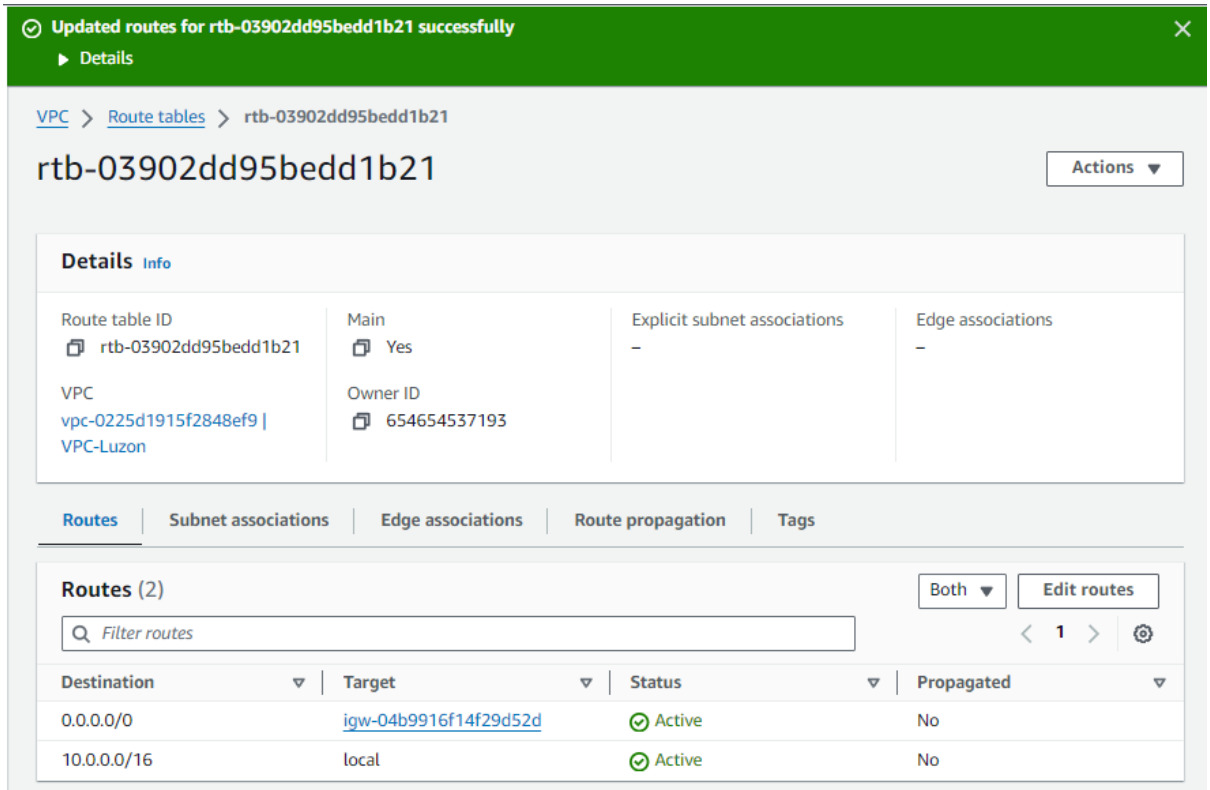
Find resources by attribute or tag

1

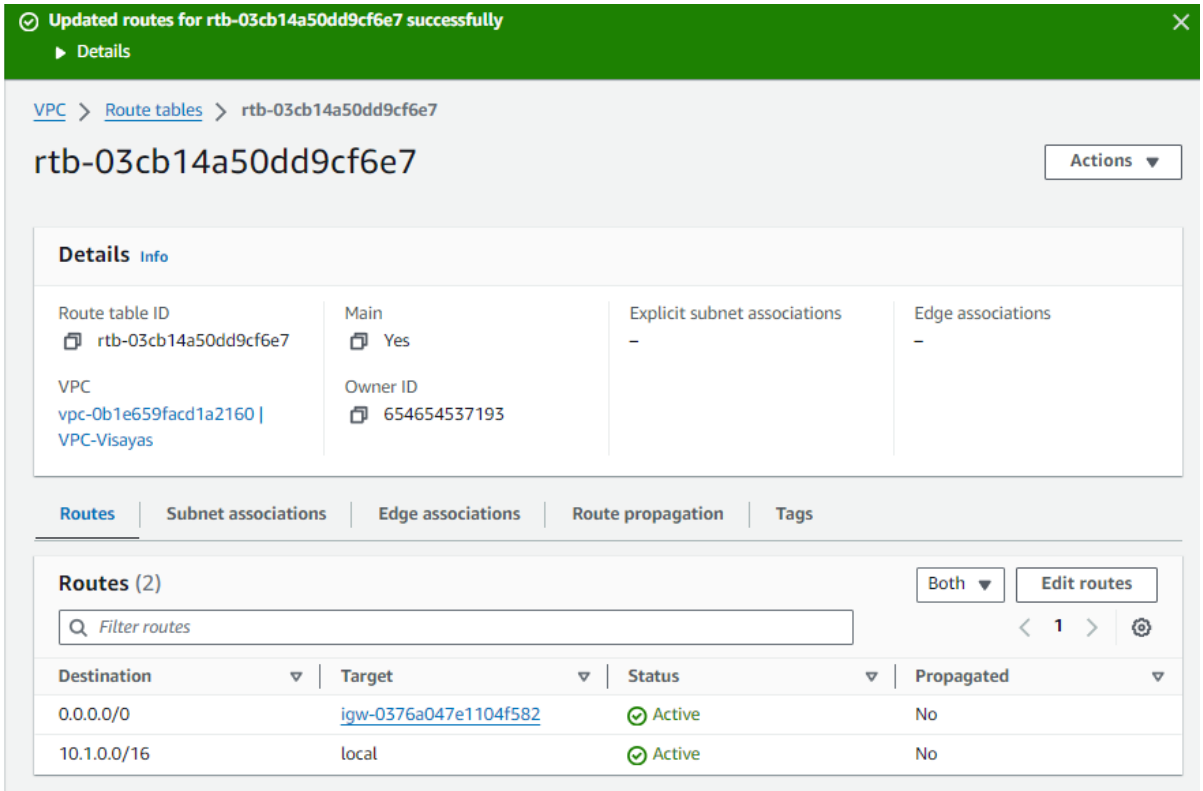
Name	Route table ID	Expli...	Edge...	Main	VPC
–	rtb-0db91ab854dec43e1	3 subnets	–	Yes	vpc-0678b78645a8bbea6
–	rtb-0c40c9ef2ddc7de5b	–	–	Yes	vpc-08ed3326dbeaf7eb2 | VPC-Mindanao
–	rtb-00e86f0c967148fa1	–	–	Yes	vpc-09a2e1c78341c7bbd | VPC-Visayas
–	rtb-06a1b115fa983f0e6	–	–	Yes	vpc-0d31a487a1843a80c | VPC-Luzon

-
  -
  -

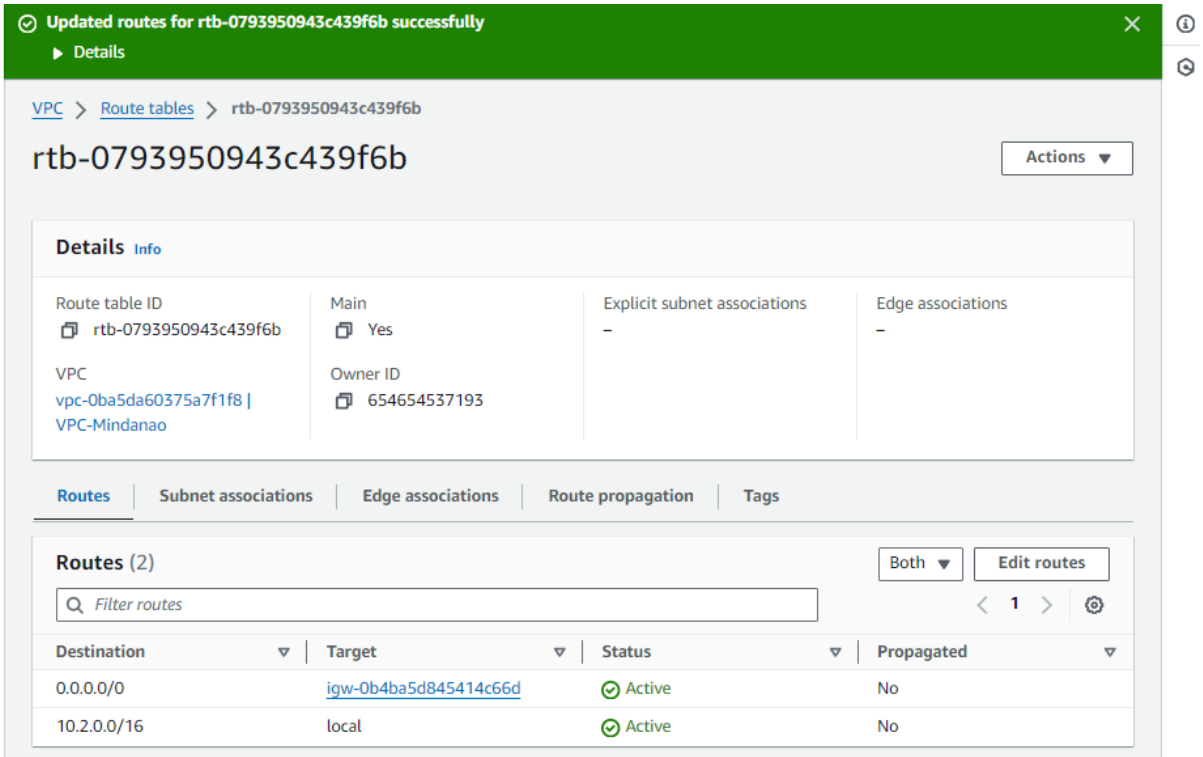

Edit routes
Route 1
Destination
10.0.0.0/16
Target
local
local
Status
Active
Propagated
No
Route 2
Destination
0.0.0.0/0
Target
Internet Gateway
igw-04b9916f14f29d52d
Status
Active
Propagated
No
Remove
Add route
Cancel
Preview
Save changes


Updated routes for rtb-03902dd95bedd1b21 successfully
Details
VPC > Route tables > rtb-03902dd95bedd1b21
rtb-03902dd95bedd1b21
Actions
Details Info
Route table ID
rtb-03902dd95bedd1b21
Main
Yes
Explicit subnet associations
–
Edge associations
–
VPC
vpc-0225d1915f2848ef9 | VPC-Luzon
Owner ID
654654537193
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (2)
Both
Edit routes
Filter routes
1
Destination
Target
Status
Propagated
0.0.0.0/0
igw-04b9916f14f29d52d
Active
No
10.0.0.0/16
local
Active
No

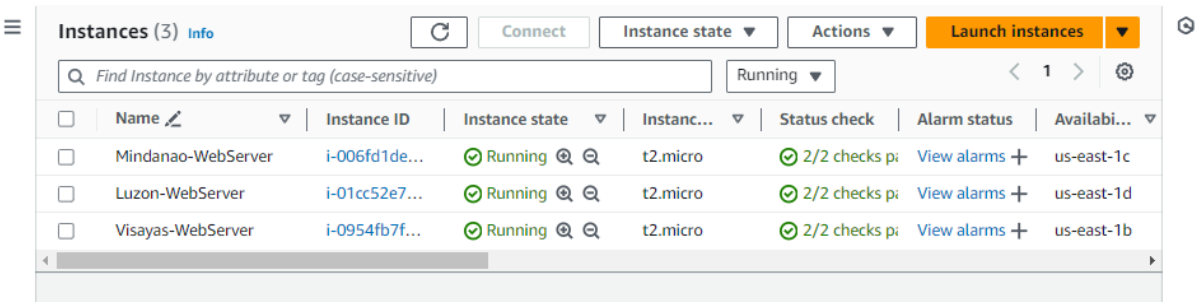
-
  -


Updated routes for rtb-03cb14a50dd9cf6e7 successfully
Details
VPC > Route tables > rtb-03cb14a50dd9cf6e7
rtb-03cb14a50dd9cf6e7
Actions
Details Info
Route table ID
rtb-03cb14a50dd9cf6e7
Main
Yes
Explicit subnet associations
–
Edge associations
–
VPC
vpc-0b1e659facd1a2160 | VPC-Visayas
Owner ID
654654537193
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (2)
Both
Edit routes
Filter routes
1
Destination
Target
Status
Propagated
0.0.0.0/0
igw-0376a047e1104f582
Active
No
10.1.0.0/16
local
Active
No

-
  -


Updated routes for rtb-0793950943c439f6b successfully
Details
VPC > Route tables > rtb-0793950943c439f6b
rtb-0793950943c439f6b
Actions
Details Info
Route table ID
rtb-0793950943c439f6b
Main
Yes
Explicit subnet associations
–
Edge associations
–
VPC
vpc-0ba5da60375a7f1f8 | VPC-Mindanao
Owner ID
654654537193
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (2)
Both
Edit routes
Filter routes
1
Destination
Target
Status
Propagated
0.0.0.0/0
igw-0b4ba5d845414c66d
Active
No
10.2.0.0/16
local
Active
No

-
  -


Instances (3) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Running
1
Name
Instance ID
Instance state
Instanc...
Status check
Alarm status
Availabi...
Mindanao-WebServer
i-006fd1de...
Running
t2.micro
2/2 checks pa
View alarms
us-east-1c
Luzon-WebServer
i-01cc52e7...
Running
t2.micro
2/2 checks pa
View alarms
us-east-1d
Visayas-WebServer
i-0954fb7f...
Running
t2.micro
2/2 checks pa
View alarms
us-east-1b

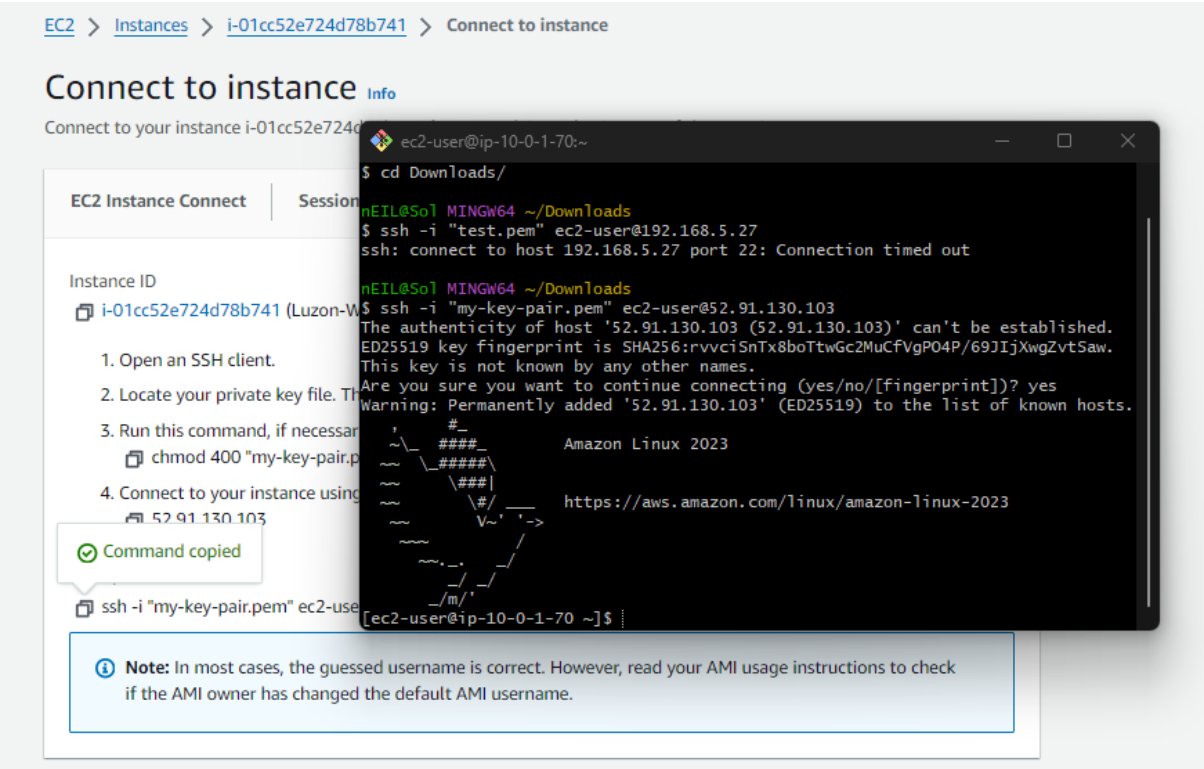
Instance Test Connection

The EC2 instances in the three VPCs are not connected yet because each VPC is isolated by default, with no routes or permissions set up to allow them to communicate with each other. Without establishing VPC peering connections and configuring the necessary routing and security settings, these instances remain isolated within their own separate networks.
To check this, follow the following steps:

---

1. Connect to Luzon-WebServer via SSH using your terminal or GitBash.

Please note that you can choose any Web-Server you want to connect with via SSH to do this step.

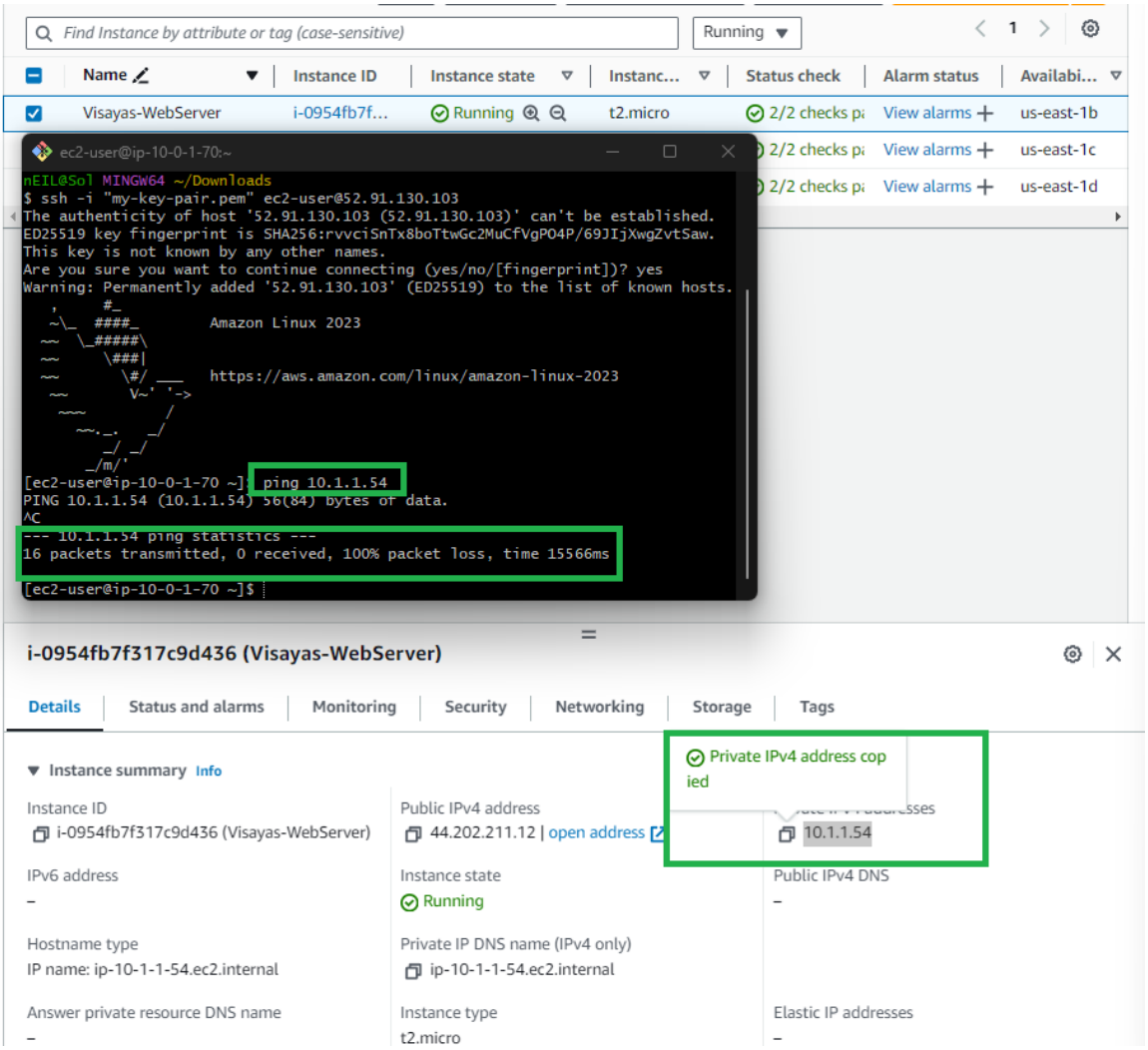


2. Ping the Visayas-WebServer's by copying their Private IP address using the command:

ping <Visayas-WebServer's- Private-IPV4-address>

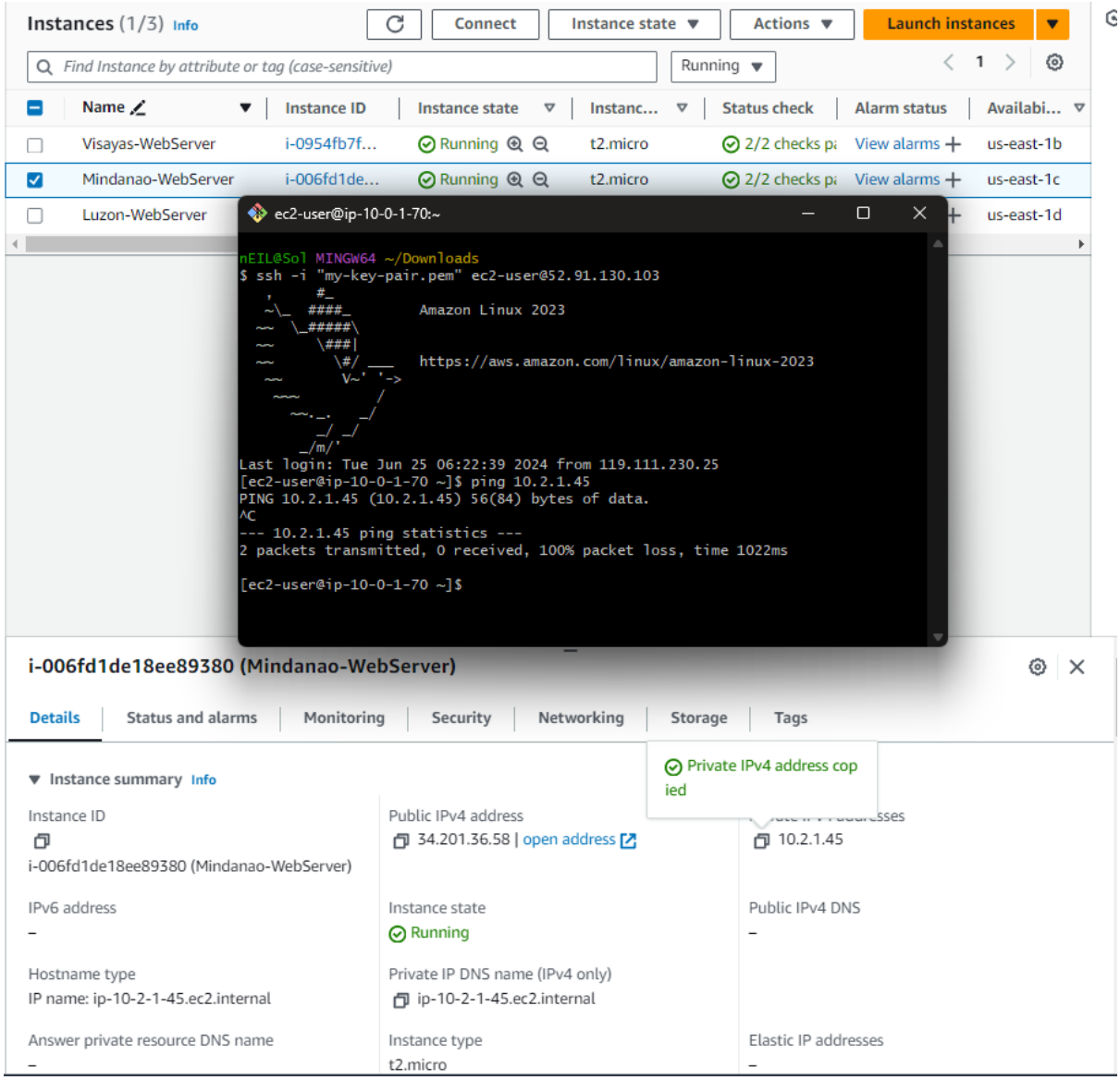
a. Hit Enter

b. In few seconds, Hit CTRL + C on your keyboard to exit the ping.



As you can observed, there are no packets received.

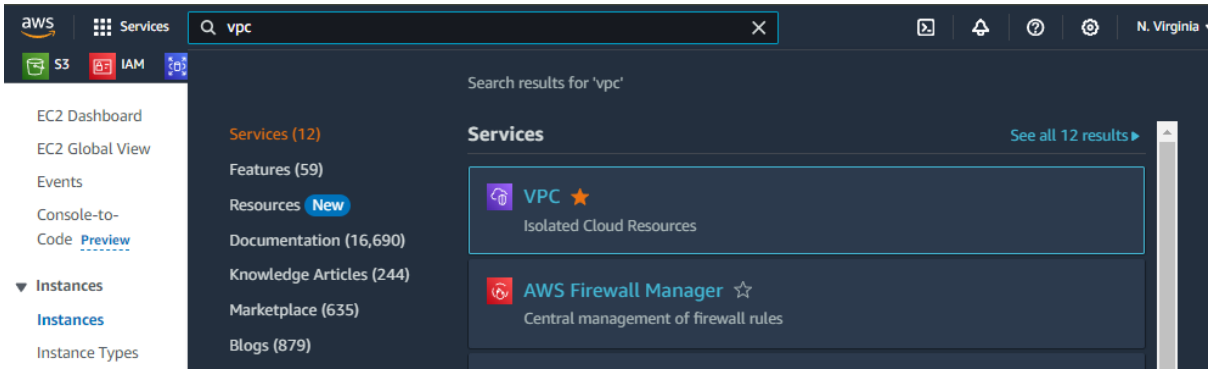
3. Try pinging the Mindanao-WebServers' private IP address this time using the *ping* command



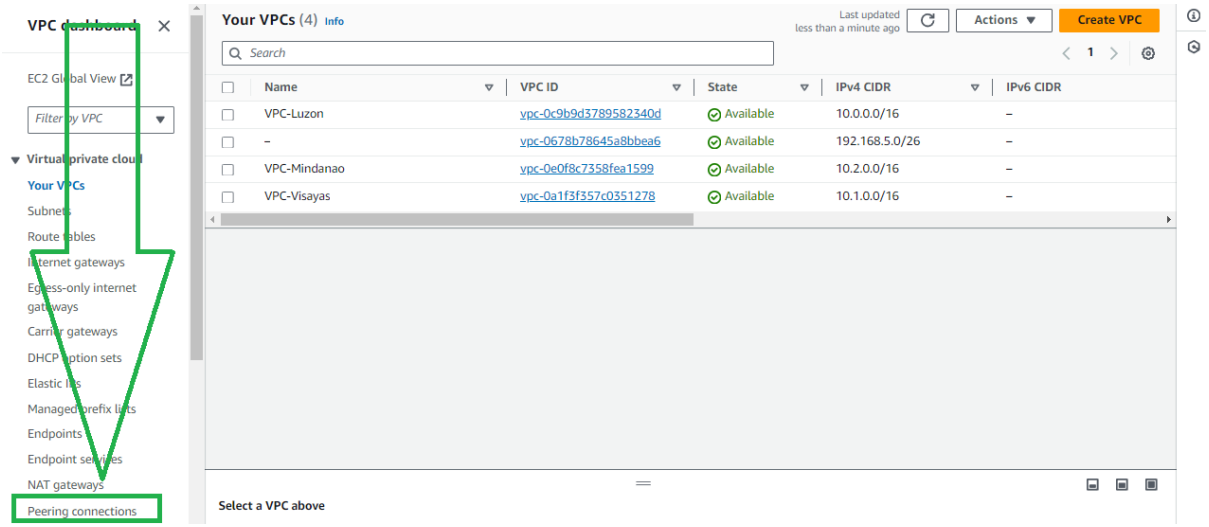
4. This confirms that these 3 web servers are not connected. To connect this servers, we need to do the following next steps…

Configuring VPC Peering Connections

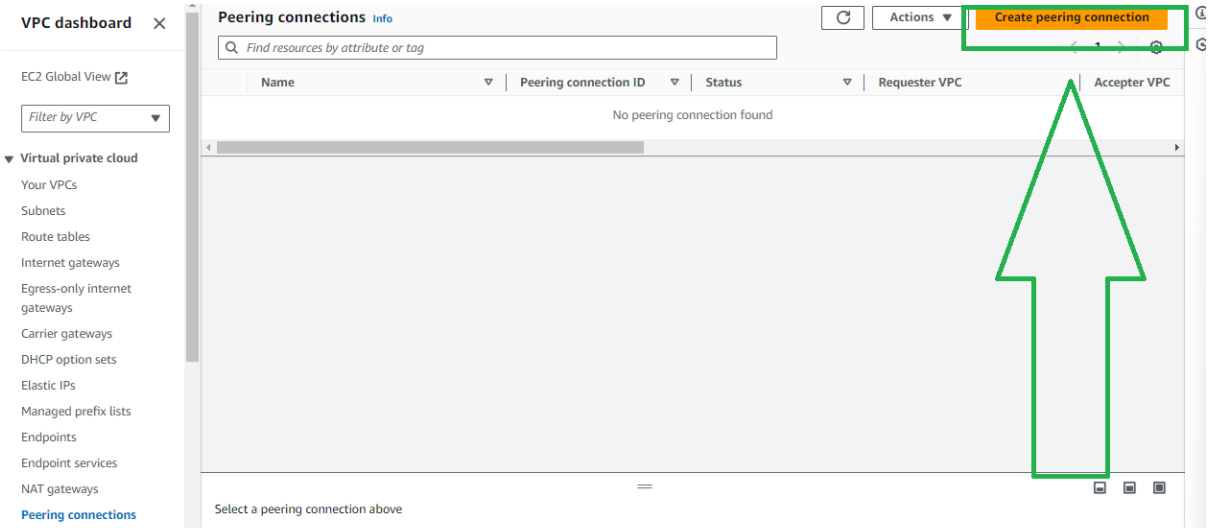
1. Navigate back to the **VPC Dashboard**



2. Search for the **Peering connections** in the **Left-side-bar**



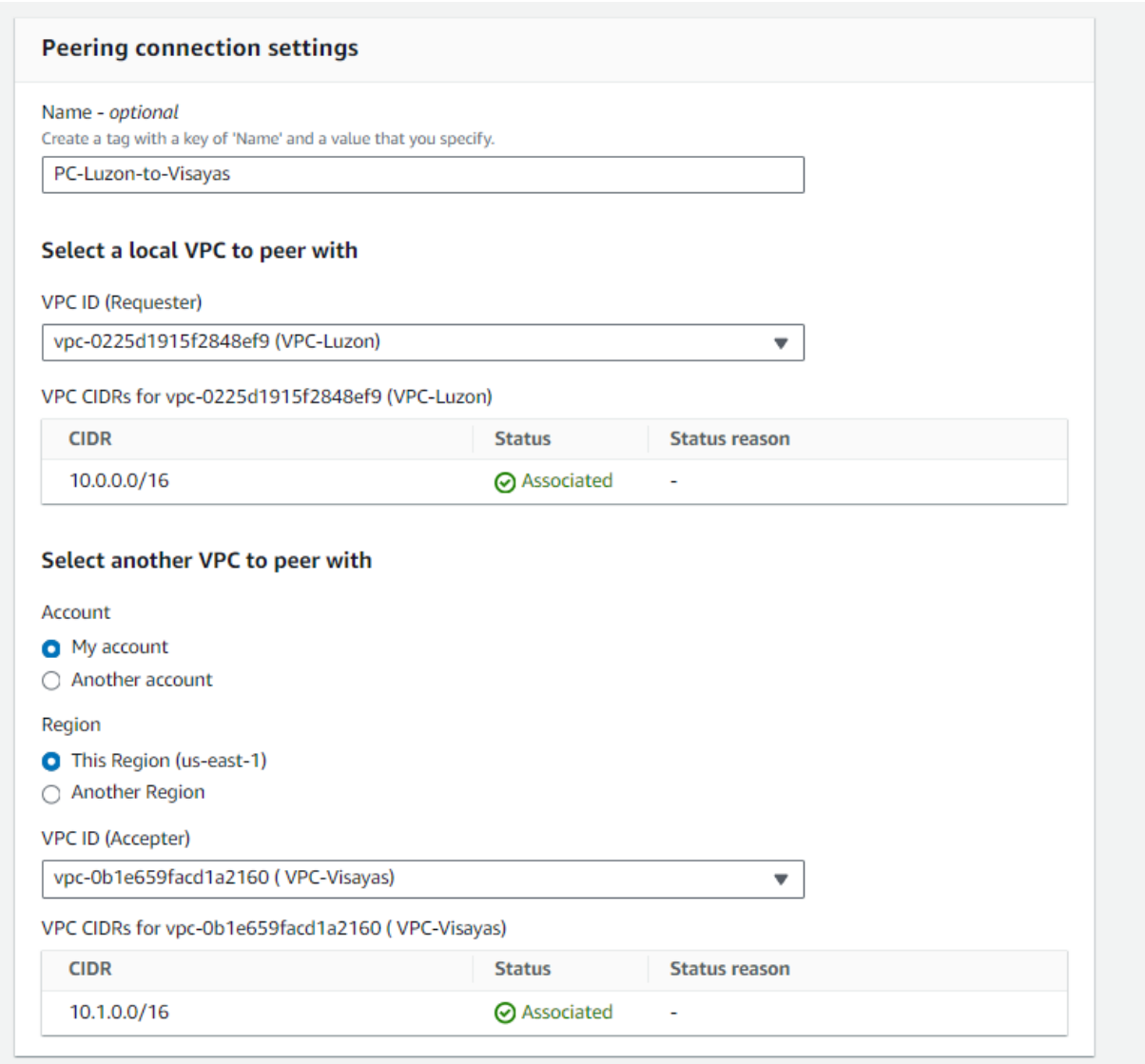
3. **Peer VPC-Luzon with VPC-Visayas** by Clicking **Create peering connection**



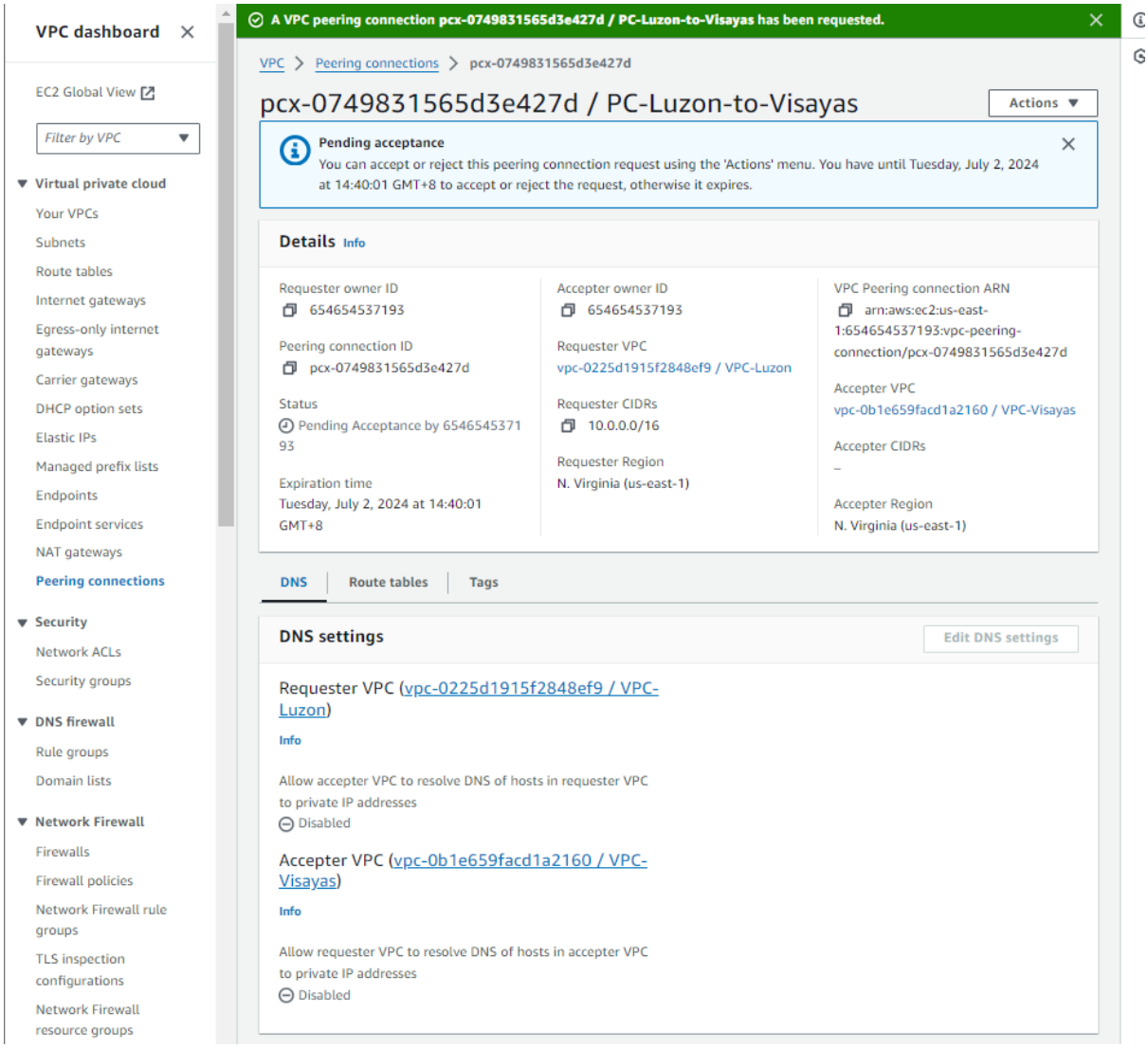
a.Name – *optional* : **PC-Luzon-to-Visayas**

b. Select a local VPC to peer with:

-
    - VPC ID (Requester) : **VPC-Luzon**
    - VPC ID (Accepter) : **VPC-Visayas**

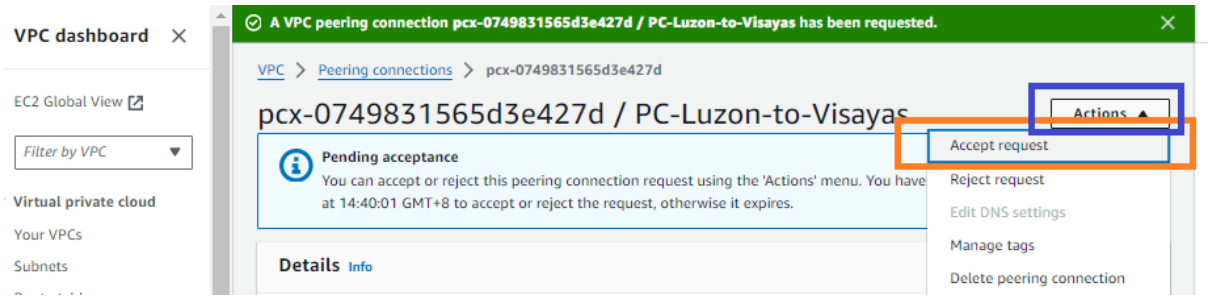


c. Click **Create peering connection**

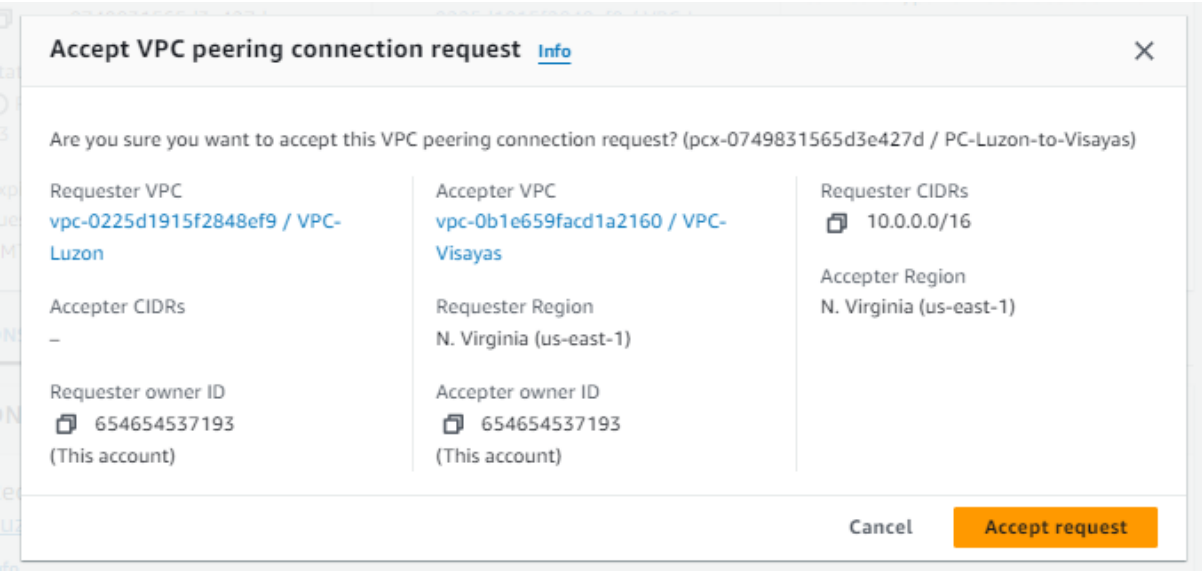


d. Accept the peering requests in each VPC's dashboard.

-
  - Click on the **Actions** and Click on **Accept request**



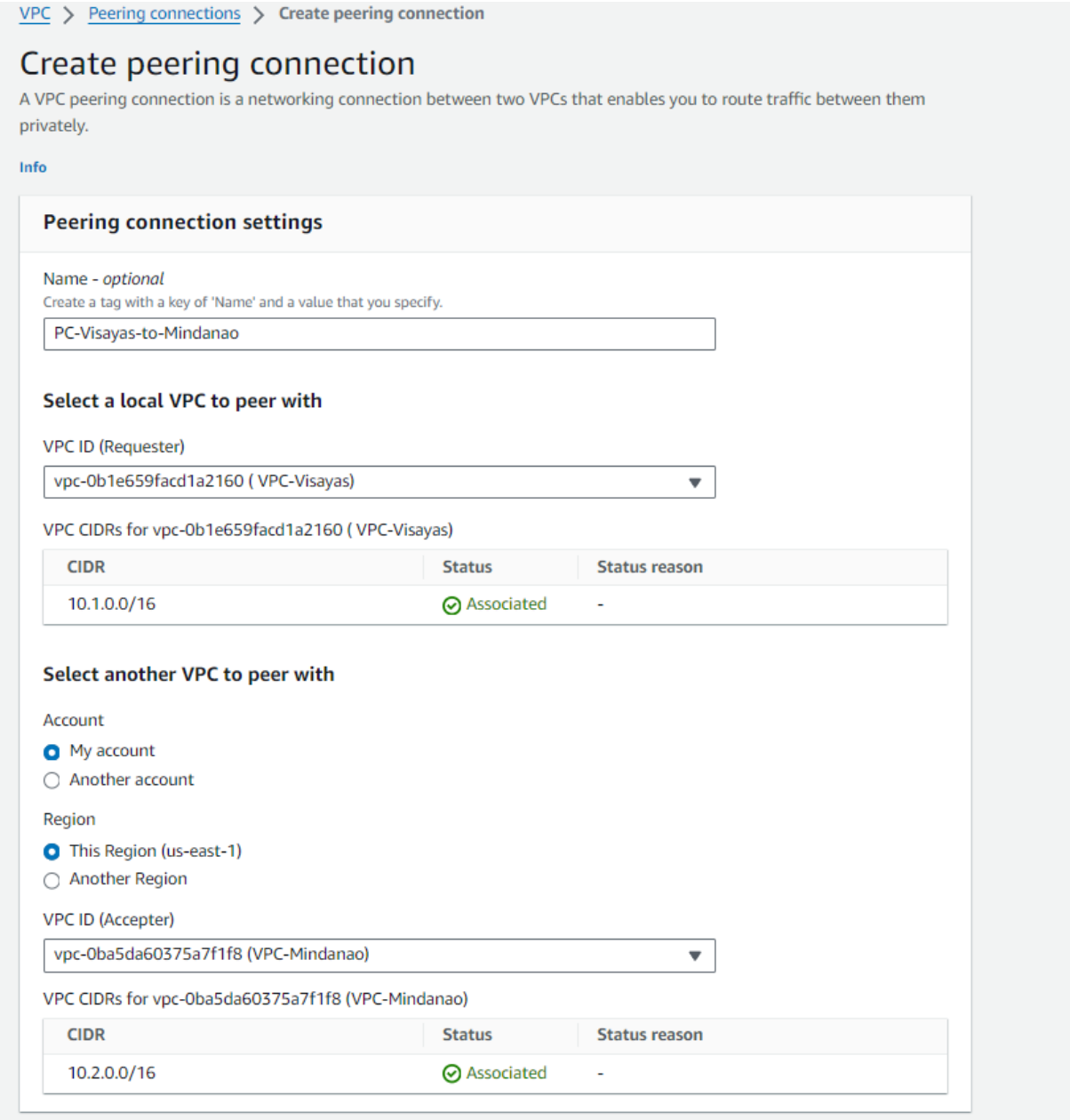
-
  - Click **Accept request**



4. **Peer VPC-Visayas with VPC-Mindanao**

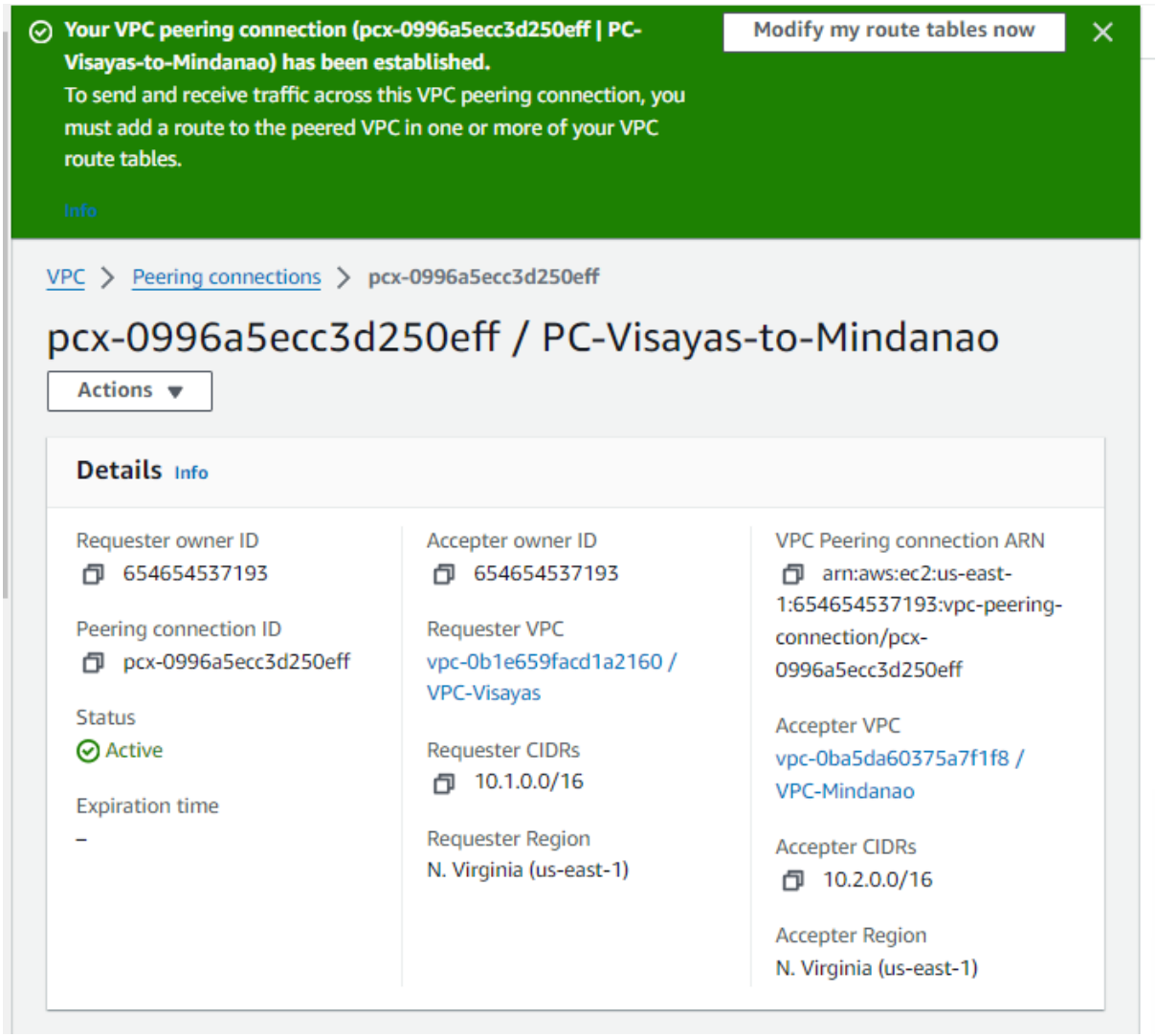
a. Repeat the same process but using the following configuration:

-
    - Name – *optional* : **PC-Visayas-to-Mindanao**
    - Select a local VPC to peer with:
        - VPC ID (Requester) : **VPC-Visayas**
        - VPC ID (Accepter) : **VPC-Mindanao**



-
    - Click **Create peering connection**

b. Accept the peering requests.

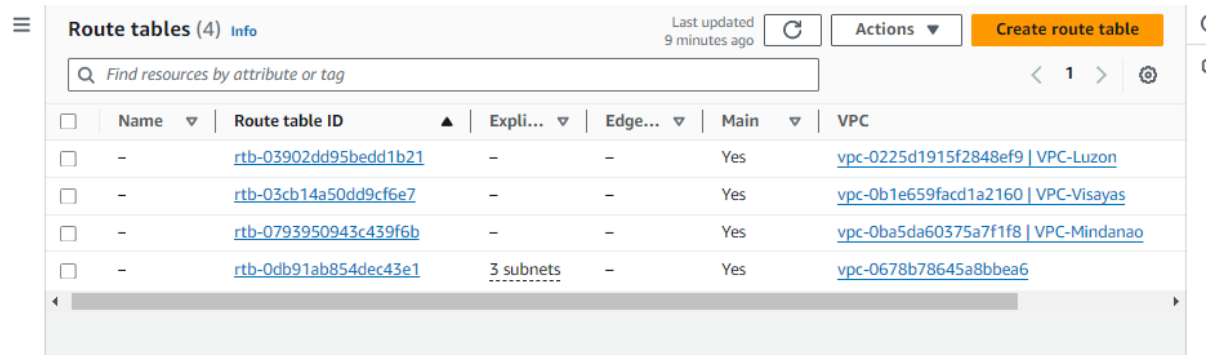


Updating Route Tables

We also need to update the route tables for the peered VPC to send and receive traffic.

---

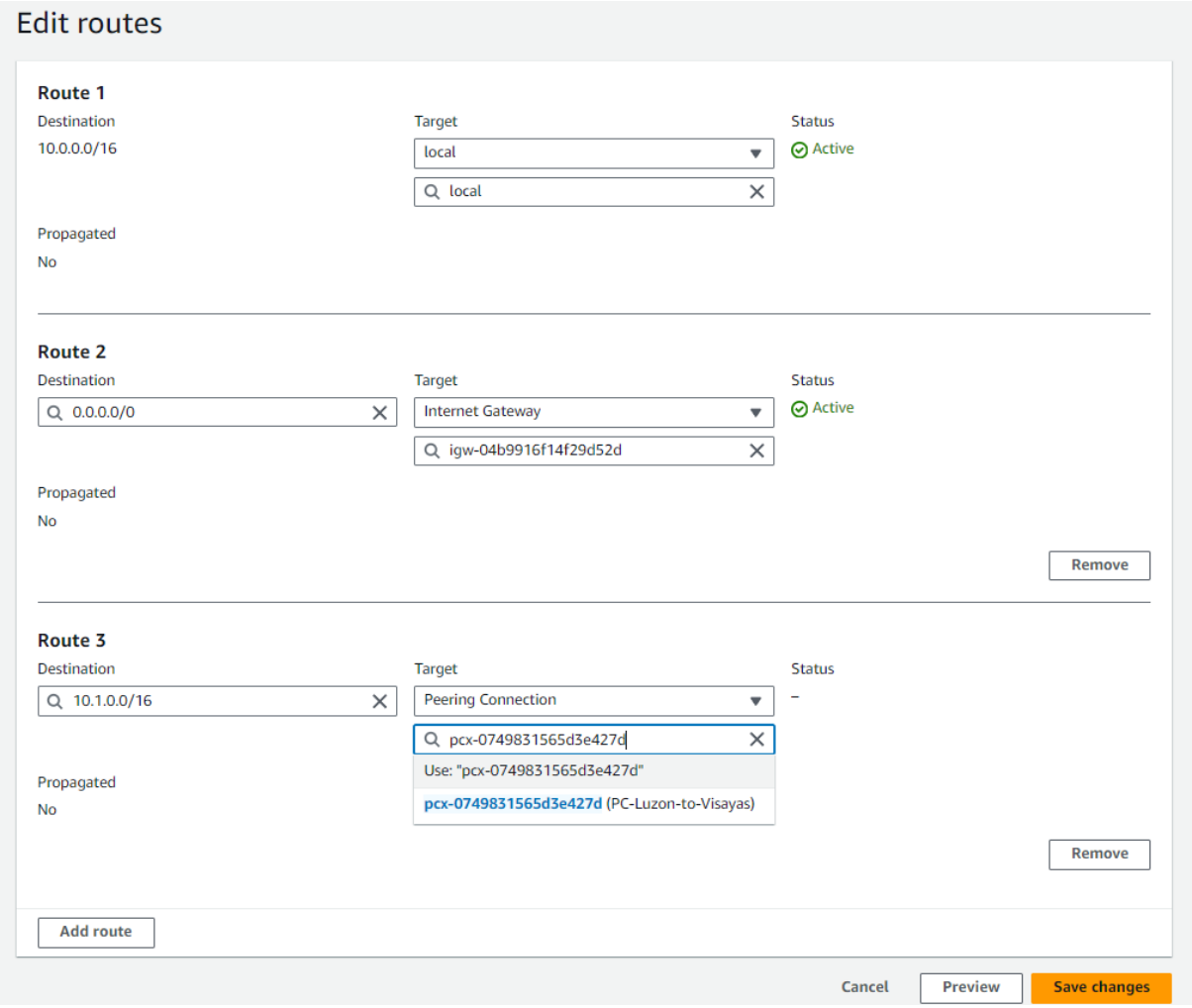
1. Navigate to the Route tables



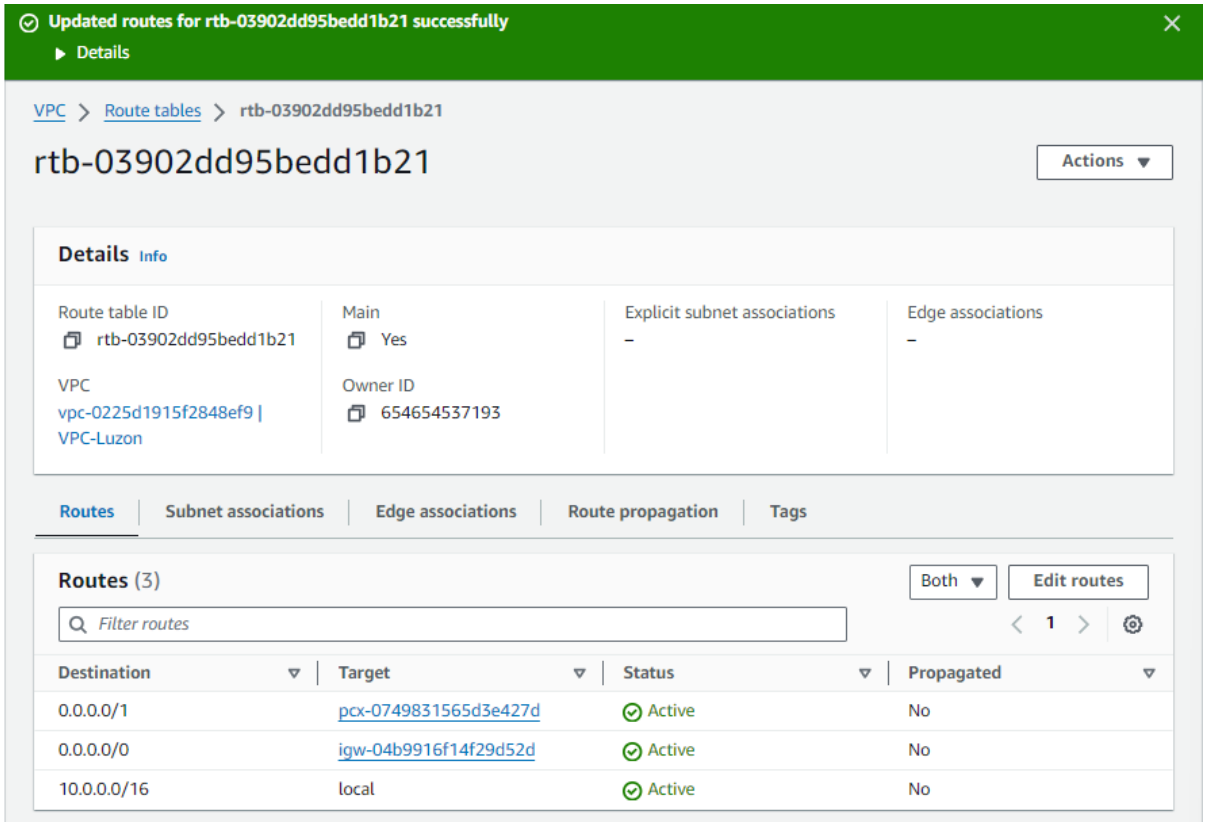
2.Add routes to direct traffic to the peered VPCs using the peering connection IDs:

a. **VPC-Luzon route table**:

-
    - Click **Edit routes**
    - Add in **Destination: *10.1.0.0/16***
        - **Target: *Peering Connection PC-Luzon-to-Visayas***

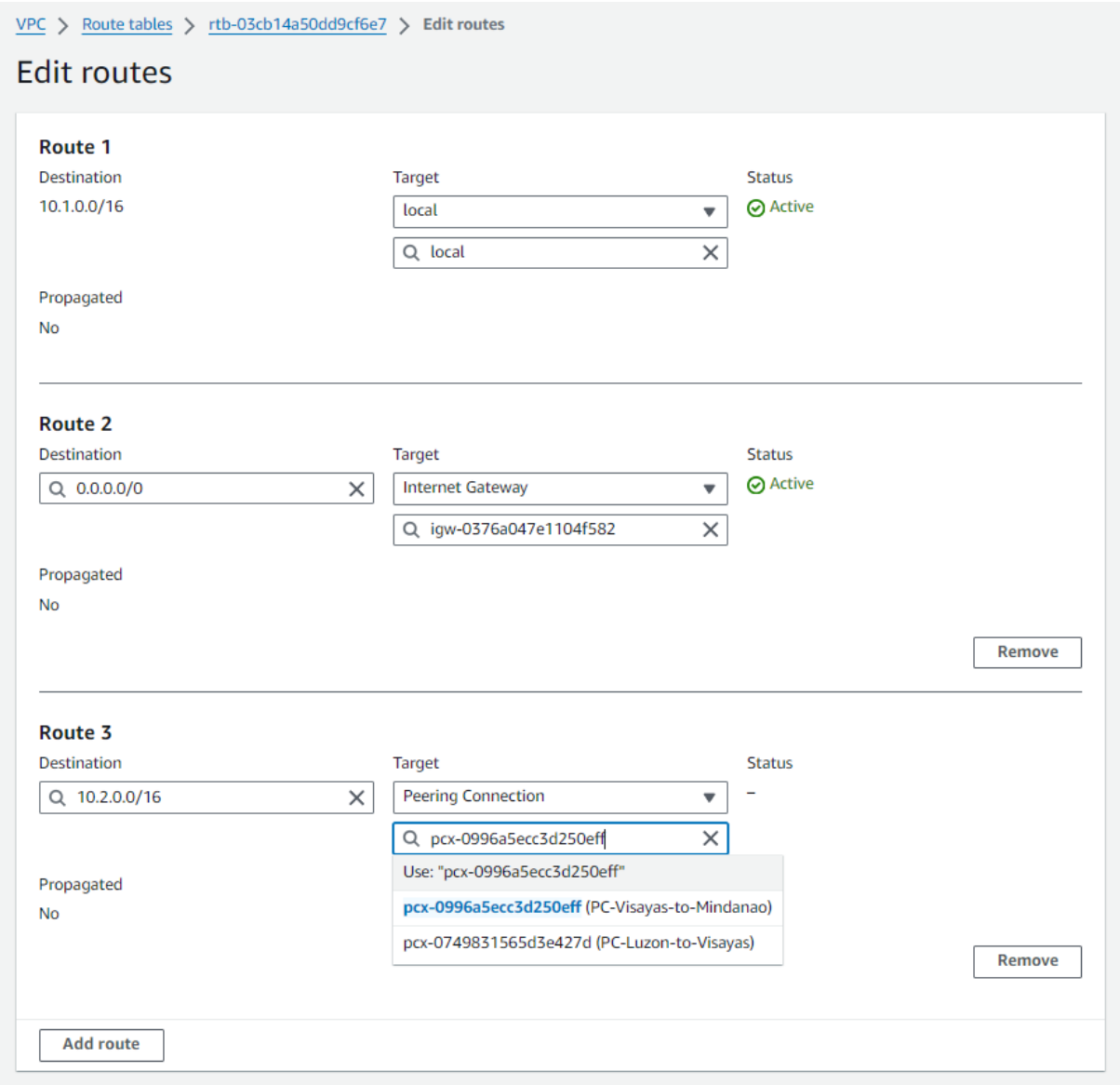


-
    - **Save changes**

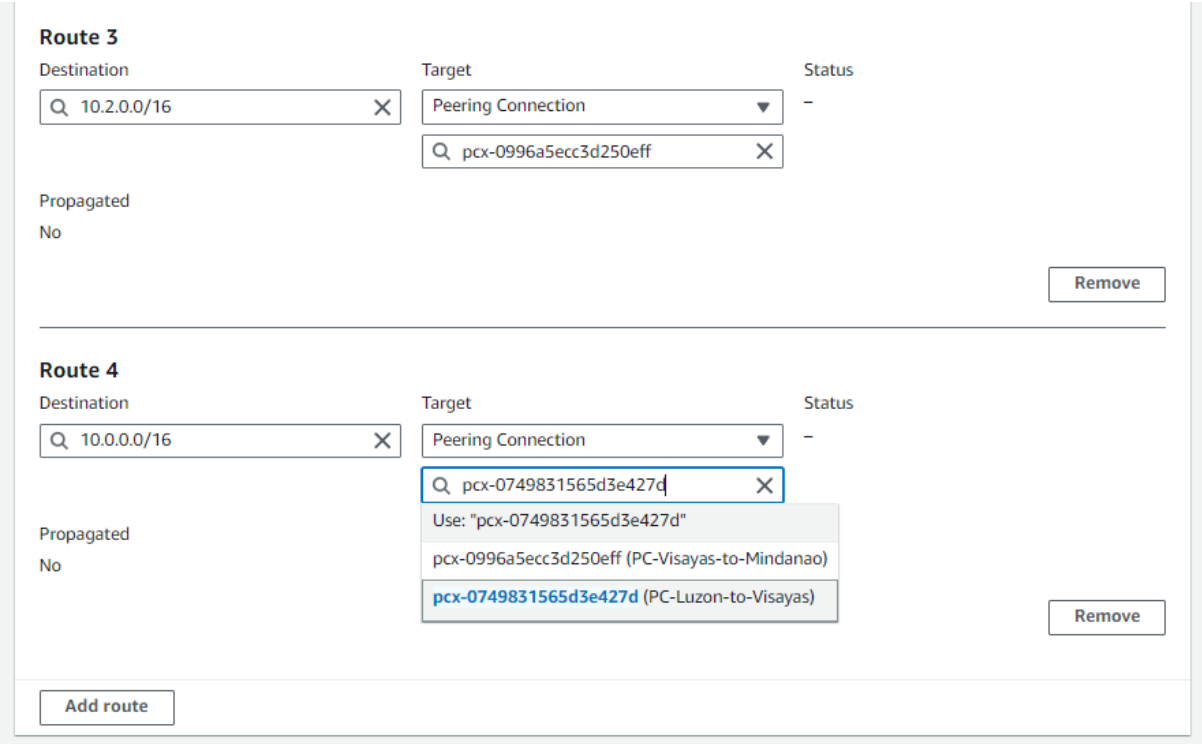


b, **VPC-Visayas route table:**

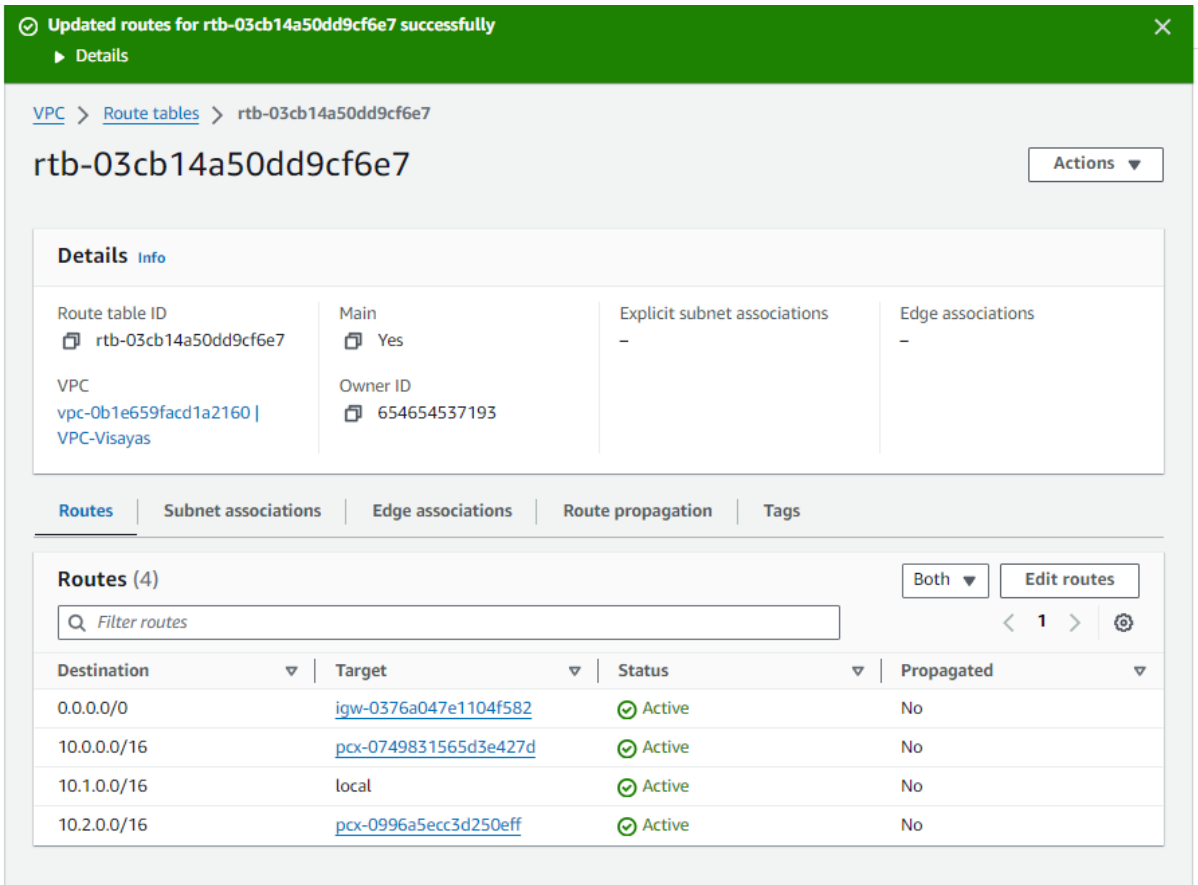
-
    - Click **Edit routes**
    - Add in **Destination : *10.2.0.0/16***
        - **Target : *Peering Connection PC-Visayas-to-Mindanao***



-
    - Add another route with the following configuration:
        - **Destination :** ***10.0.0.0/16***
        - **Target :** ***Peering Connection PC-Luzon-to-Visayas***

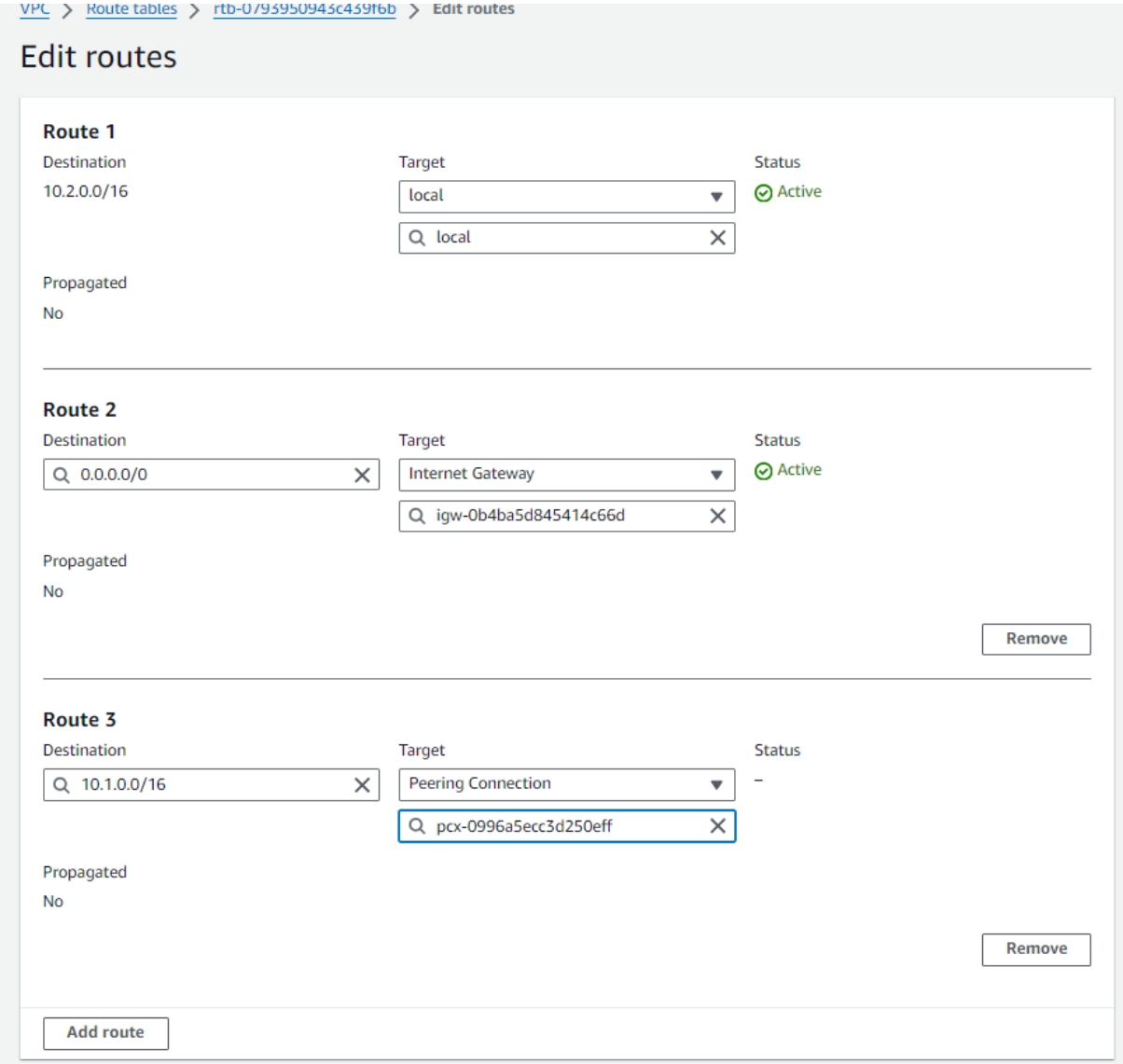


-
  - **Save changes**

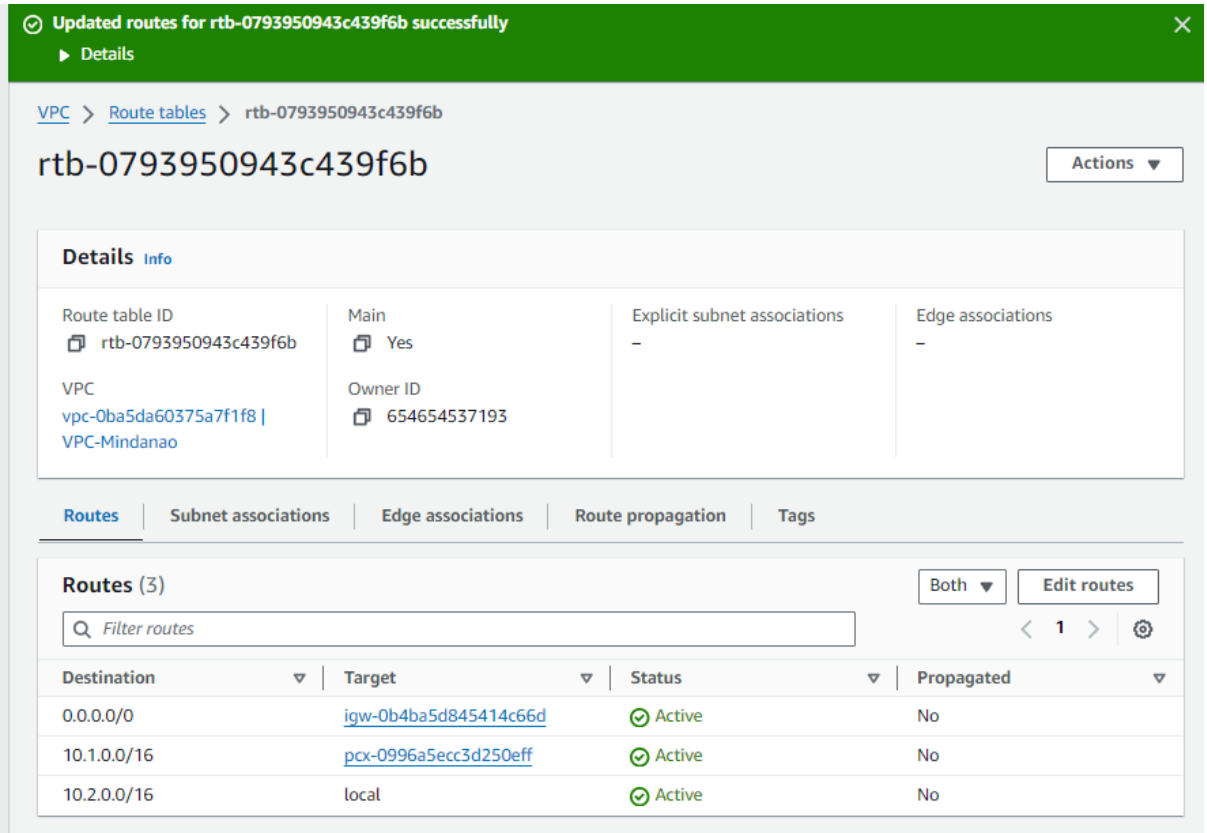


c. **VPC-Mindanao route table:**

-
    - Click **Edit routes**
    - Add in **Destination : *10.1.0.0/16***
        - **Target : *Peering Connection PC-Visayas-to-Mindanao***



-
    - Save changes

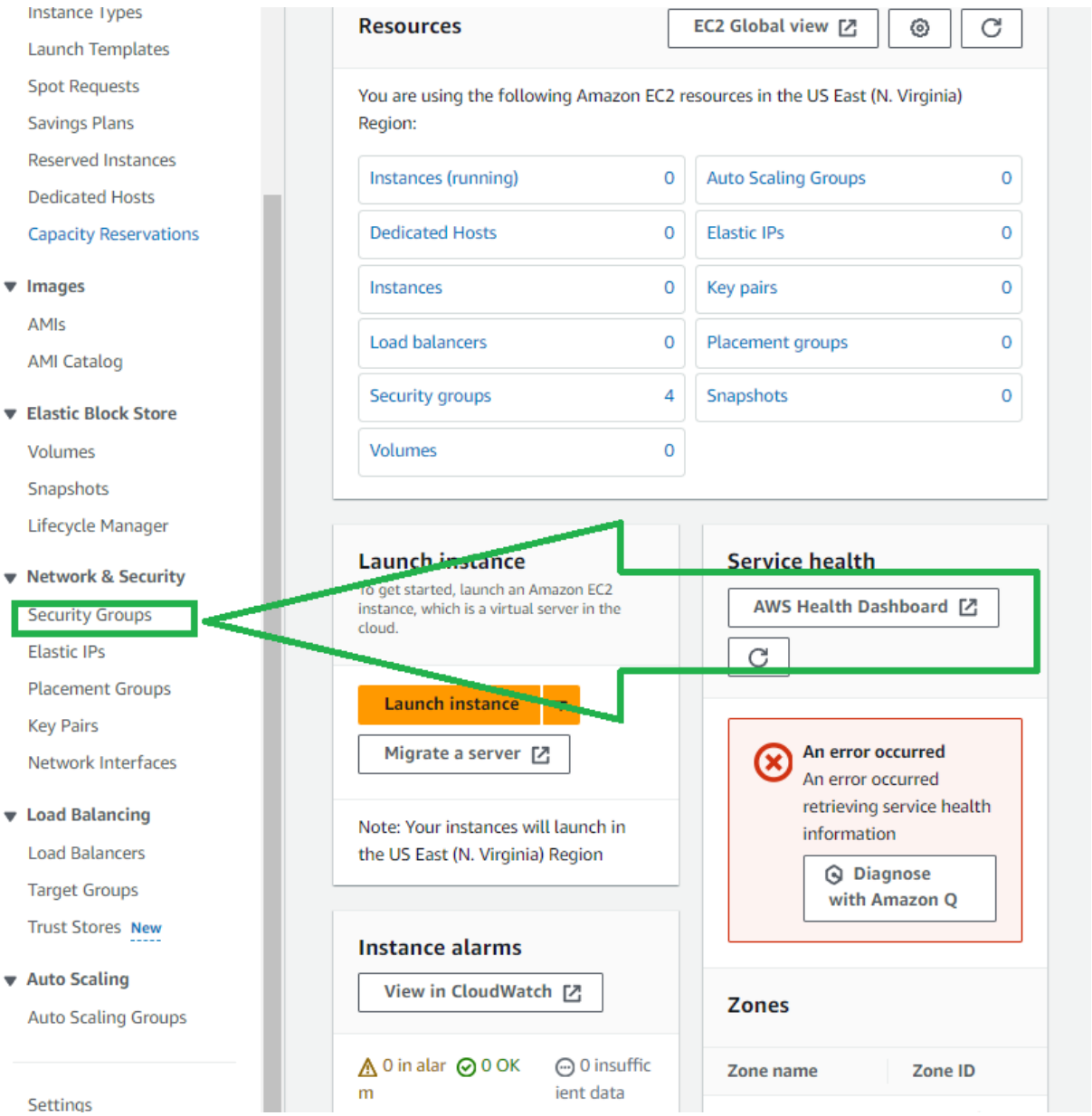


3. Lastly, we need to configure the **Security Group for Each Instances**:

Think of a security group in AWS as a bouncer at a nightclub. The security group/bouncer checks each visitor (data packet) at the door (instance). They have a list of rules that determine who can enter (inbound rules) and who can leave (outbound rules) based on specific criteria such as the visitor's appearance (IP addresses), their invitation (ports), and the type of event they're attending (protocols).

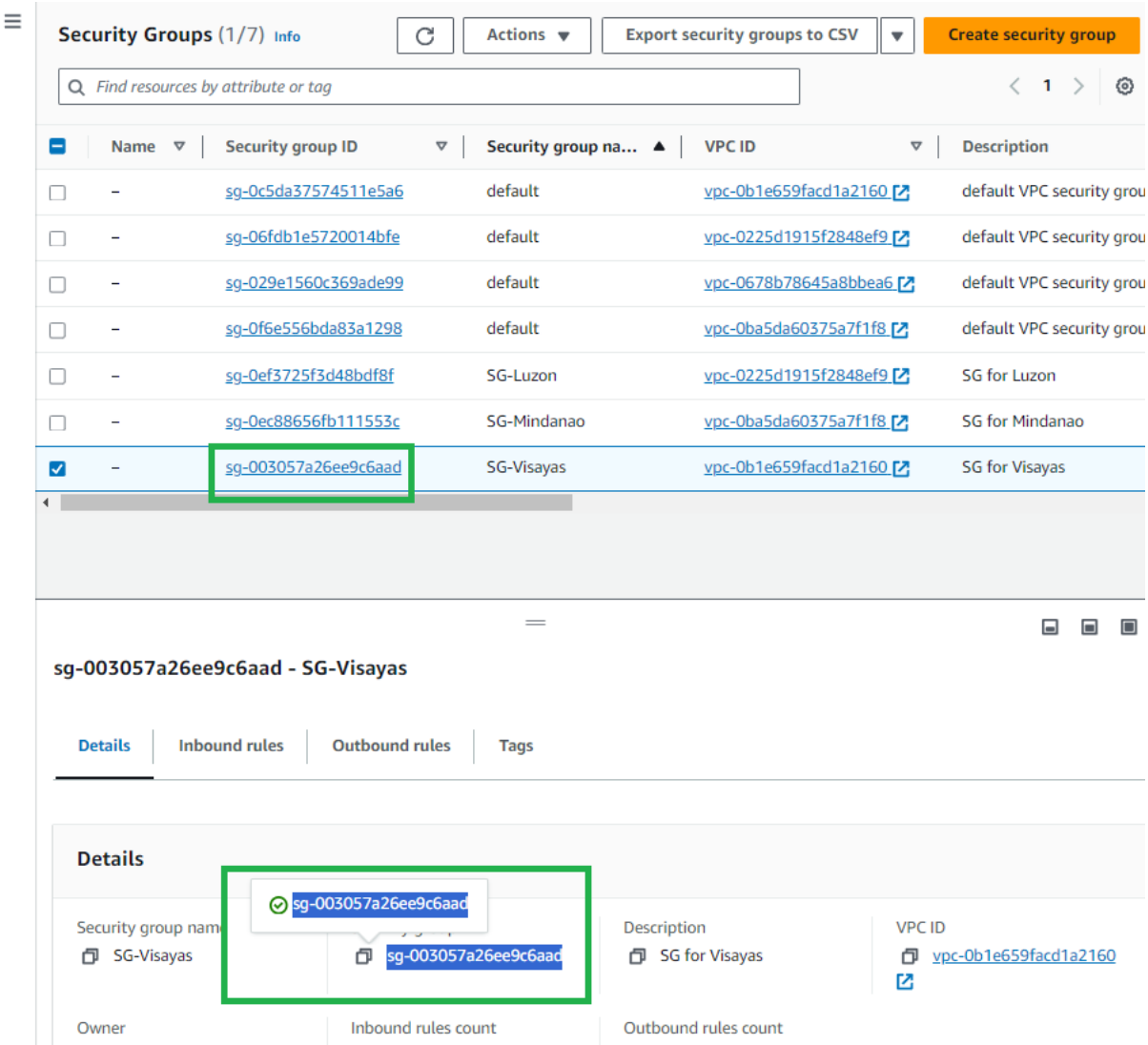
---

a. Navigate to the EC2 dashboard and look for **Security Groups** under **Network & Security** in the left-side-bar and Click on it.

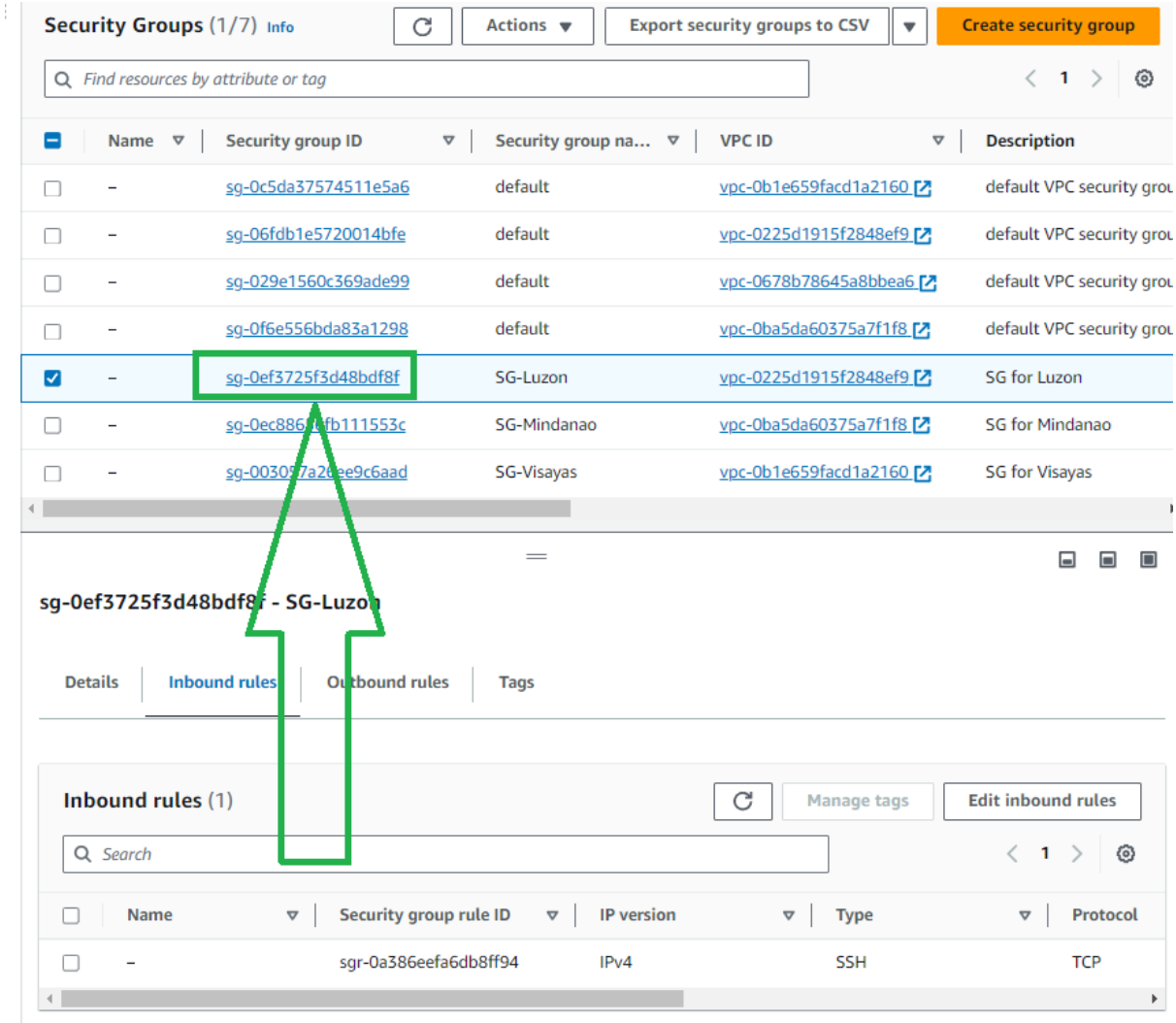


b. SG-Luzon inbound rules Configuration:

-
  - Copy the Security **Security group ID** of **SG-Visayas**

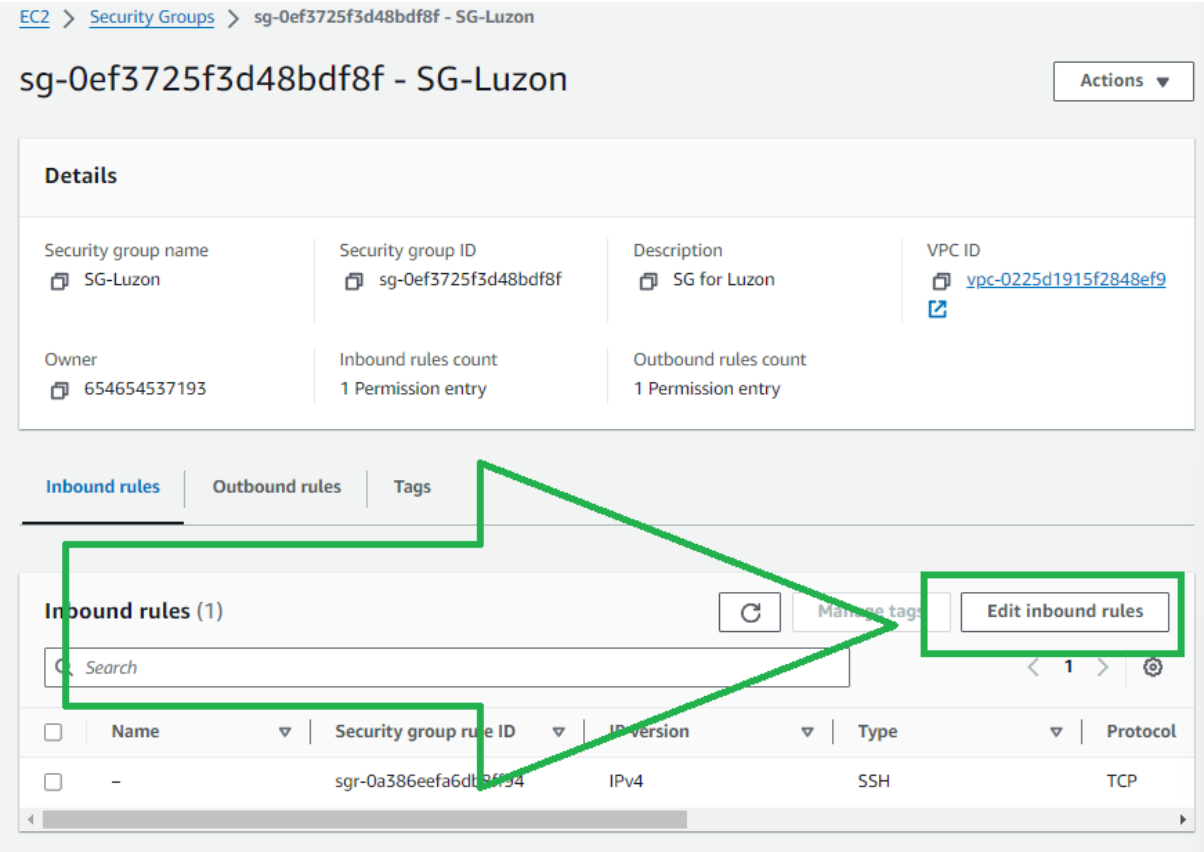


-
  - Click on the Security group ID of SG-Luzon

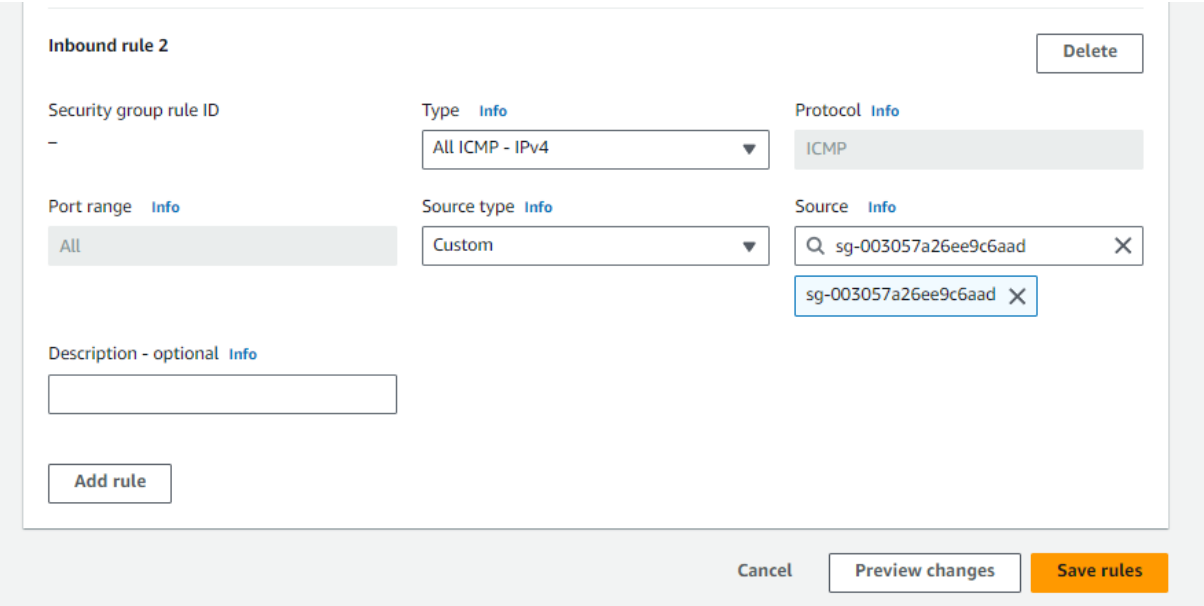


You will be redirected to the SG-Visayas Dashboard

-
  - Click **Edit inbound rules**



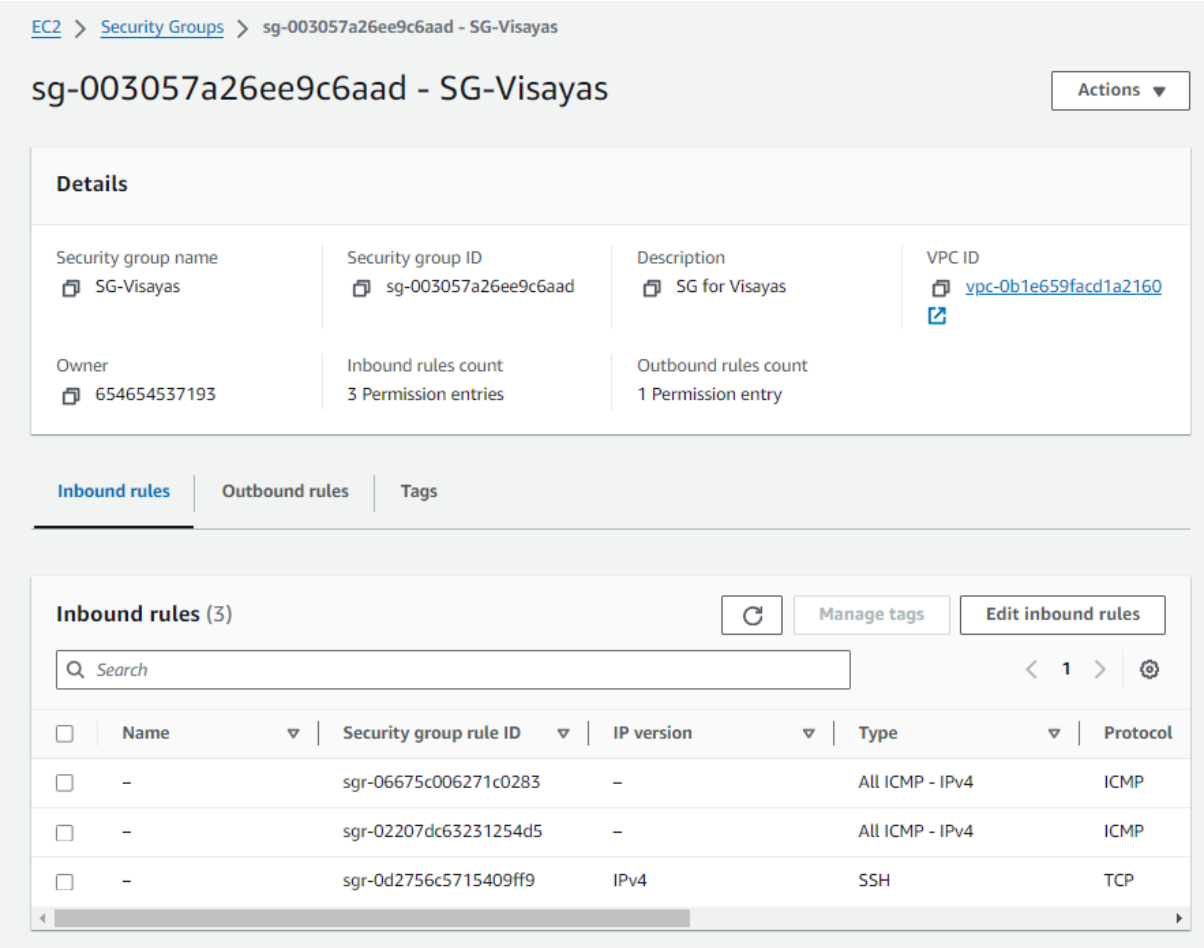
-
  - Click **Add rule** and Add following the Inbound rules configuration:
    - Type : **All ICMP – IPv4**
    - Paste the Security Group ID of SG-Visayas in the **Source**



-
  - Click **Save rules**

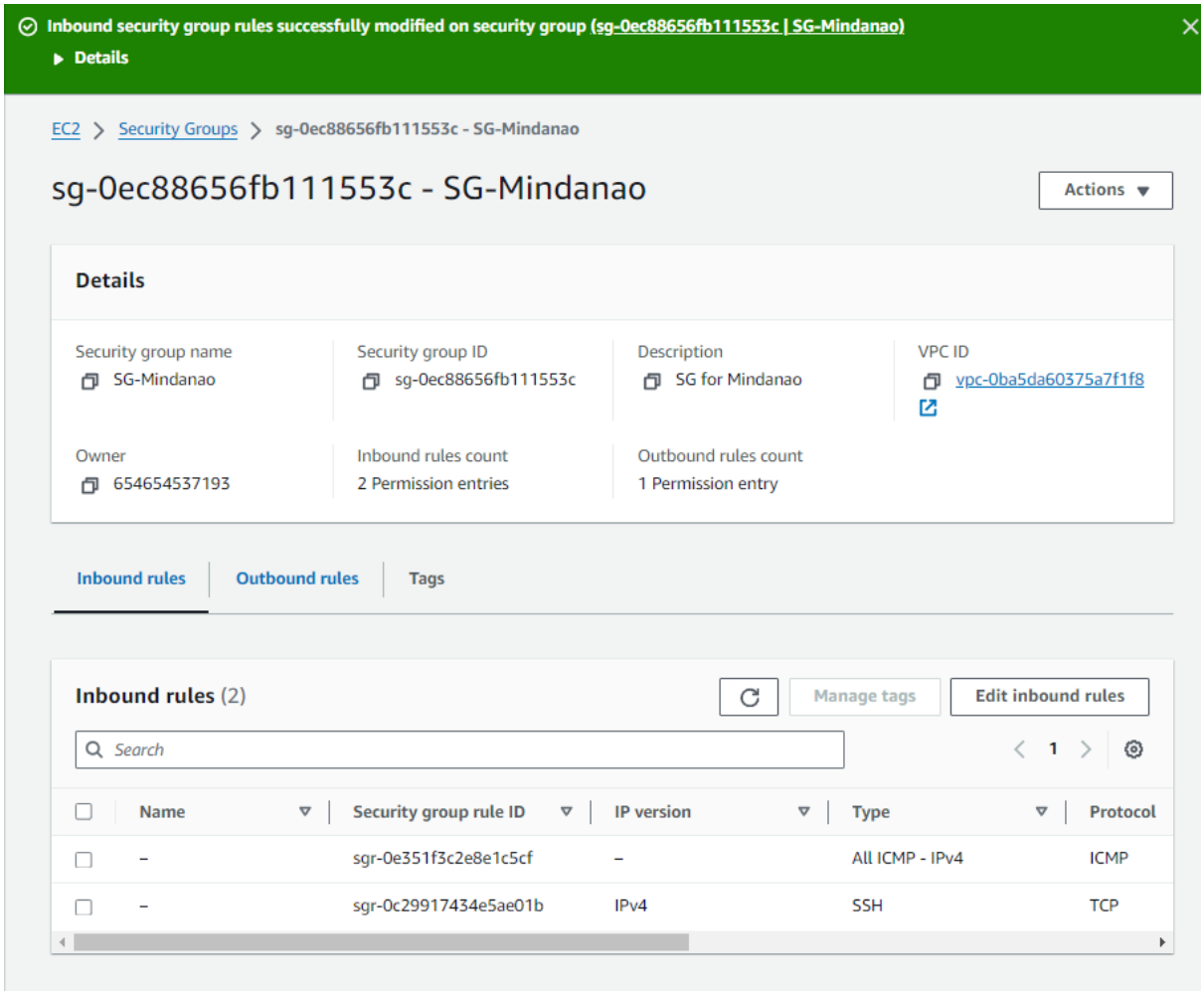
c. **SG-Visayas inbound rules Configuration**:

-
    - Follow the same process as **Step 3-b** but with the following inbound rules instead:
        - Type : **All ICMP – IPv4**
        - Paste the Security Group ID of **SG-Luzon** in the **Source**
        - Type : **All ICMP – IPv4**
        - Paste the Security Group ID of **SG-Mindanao** in the **Source**
        - Click Save



d. **SG-Mindanao inbound rules Configuration**:

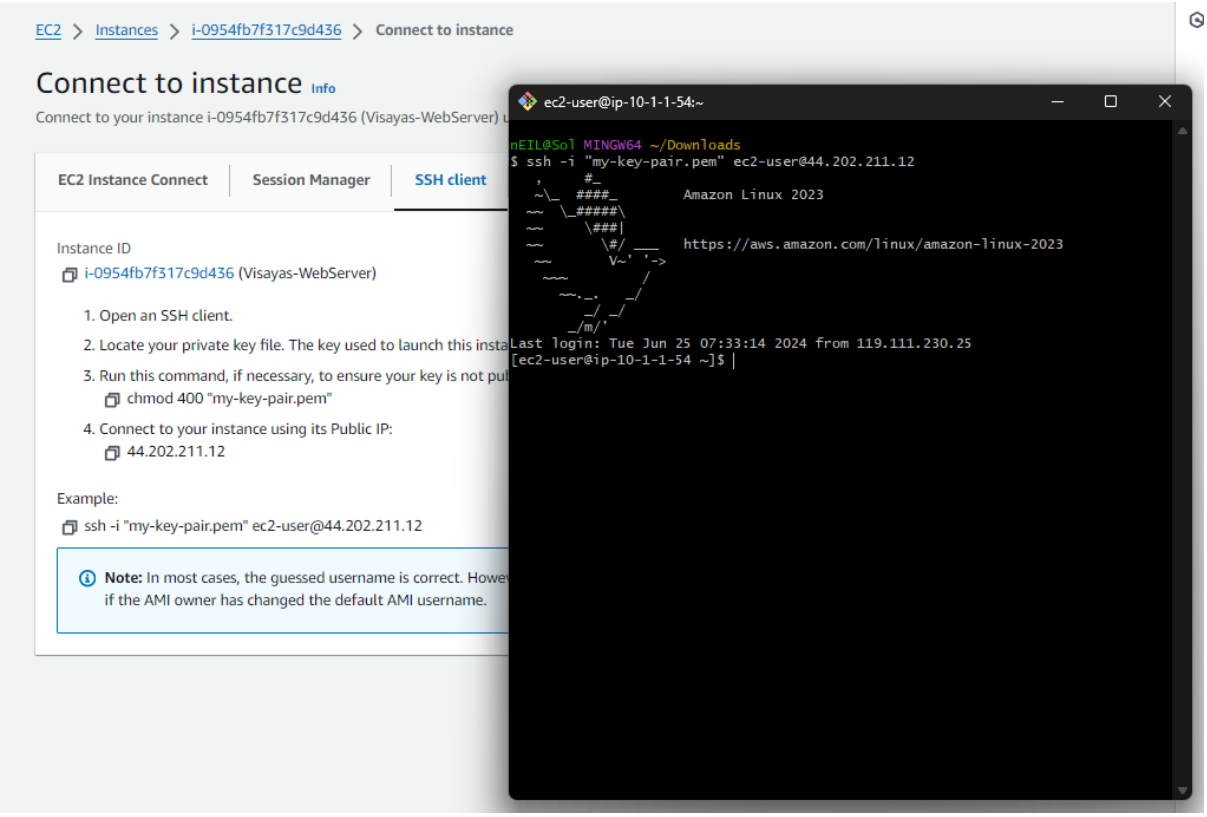
-
    - Similar to the previous step, add the following inbound rules for this SG:
        - Type : **All ICMP – IPv4**
        - Paste the Security Group ID of **SG-Visayas** in the **Source**



Testing Connectivity

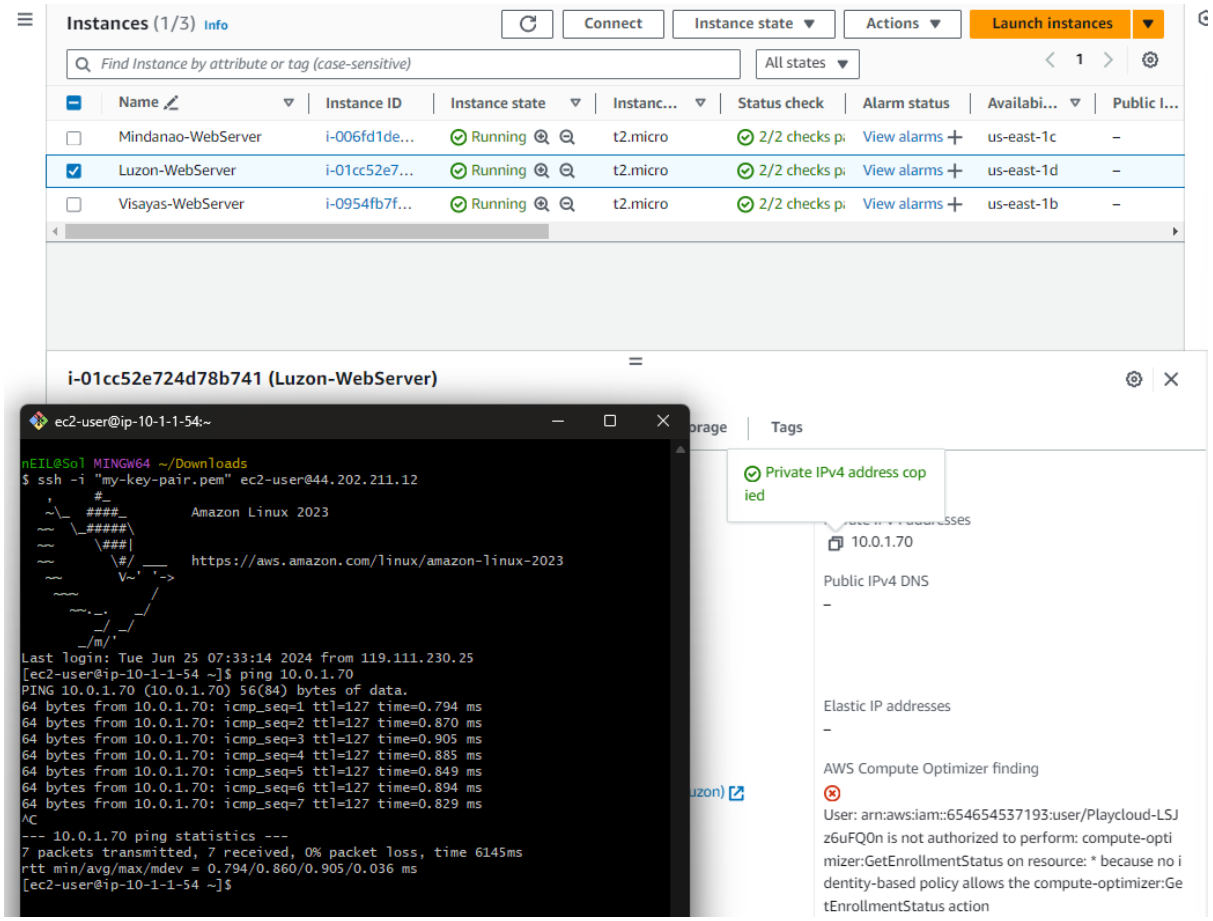
1. Similar to **Instance Test Coonnection section**, SSH into each instance from your local machine.

a. **SSH to the Visayas-WebServer:**



-
    - Lets ping the other servers Private IP address

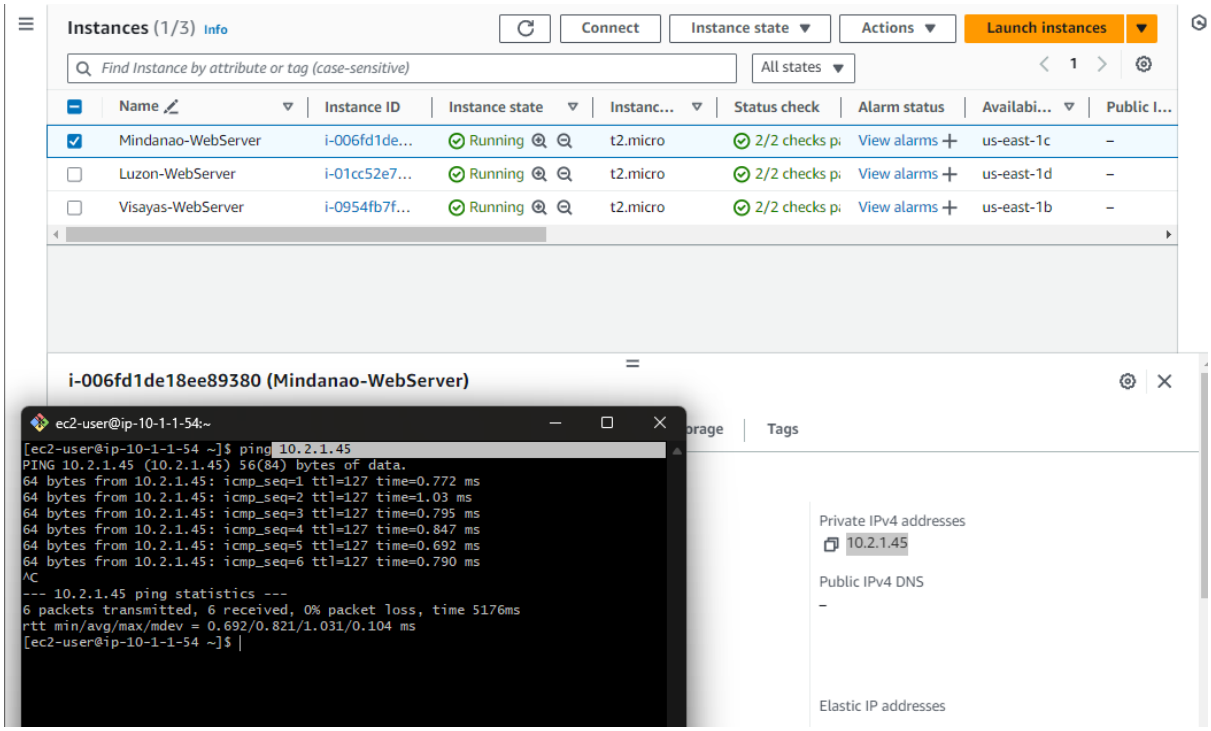
ping <Luzon-WebServer-private-IPv4-Address>



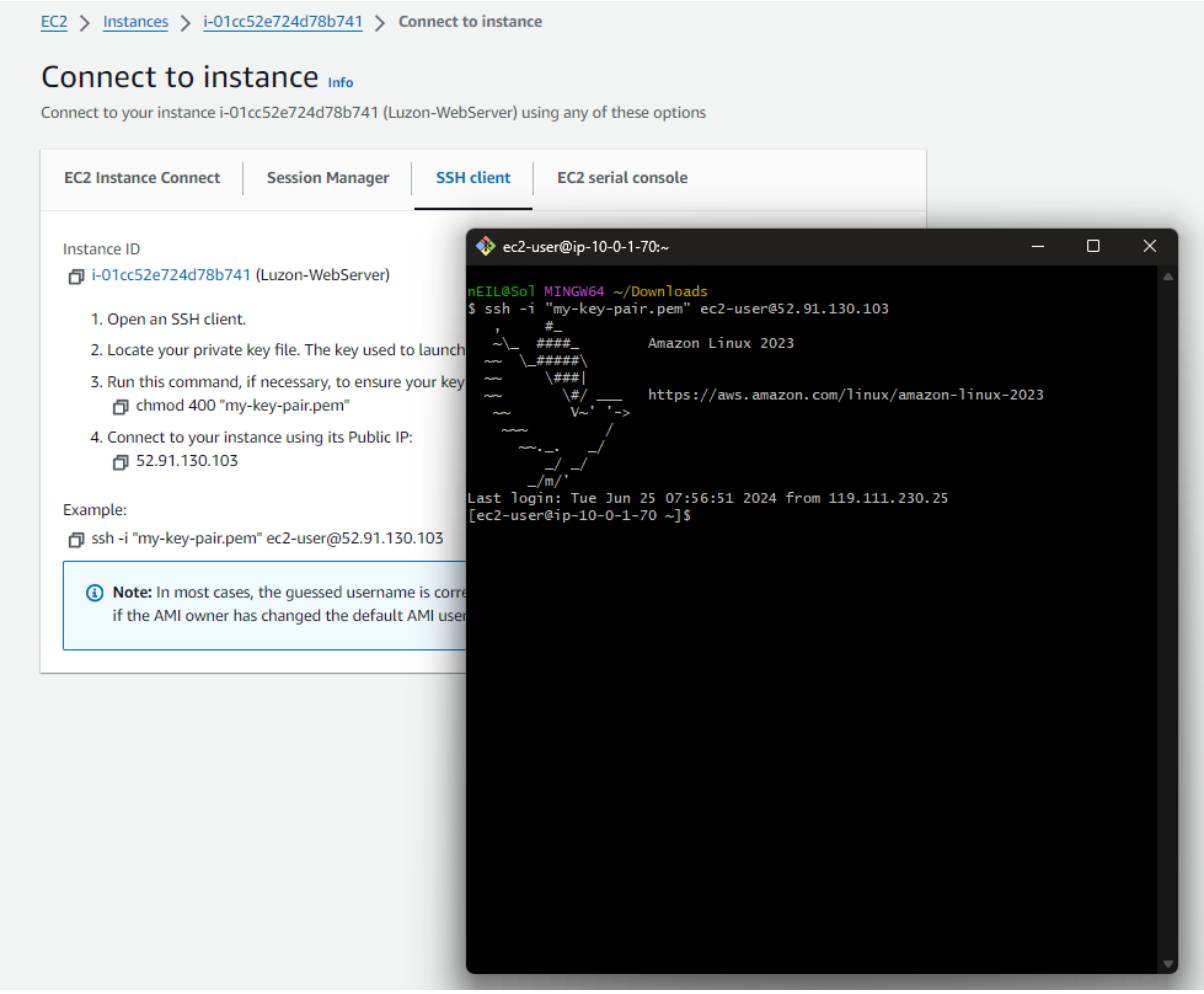
This time we can received packets from Luzon-WebServer

-
  - ping Mindanao-WebServer

ping <Mindanao-WebServer-private-IPv4-Address>



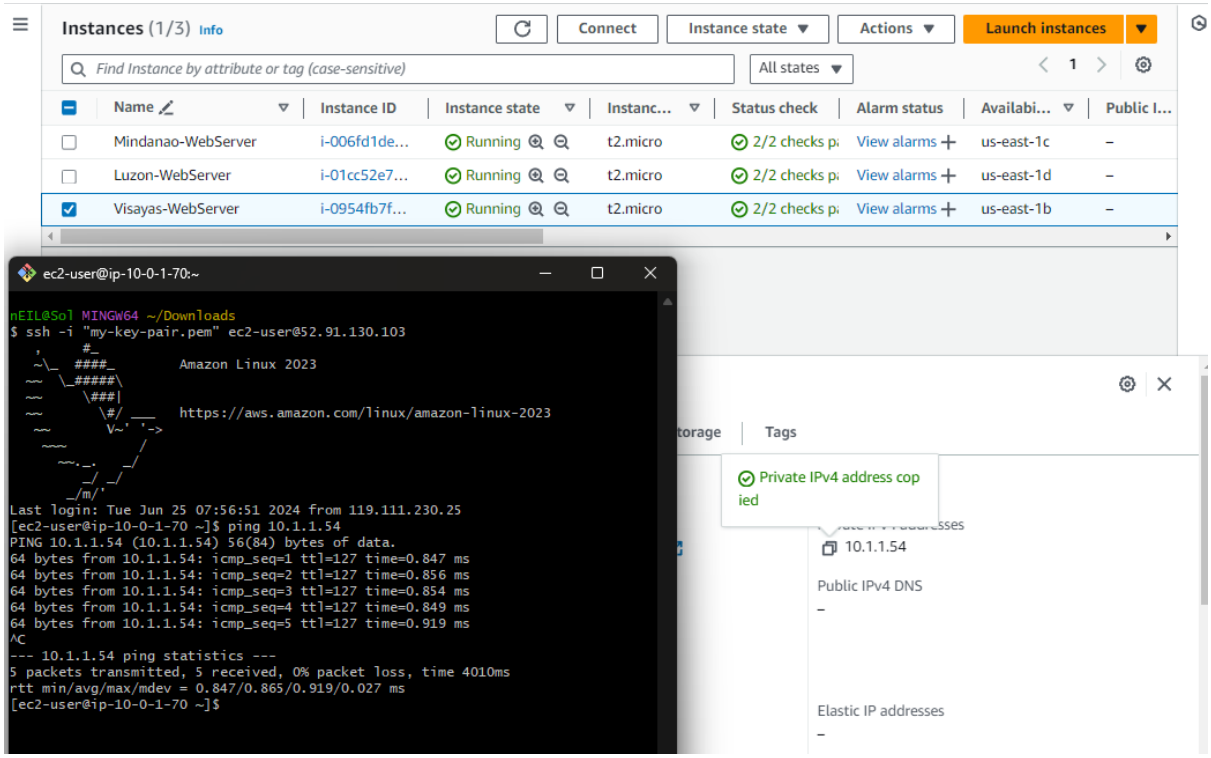
b. **SSH to Luzon-WebServer:**



•

- Lets ping the **Visayas-Webserver**

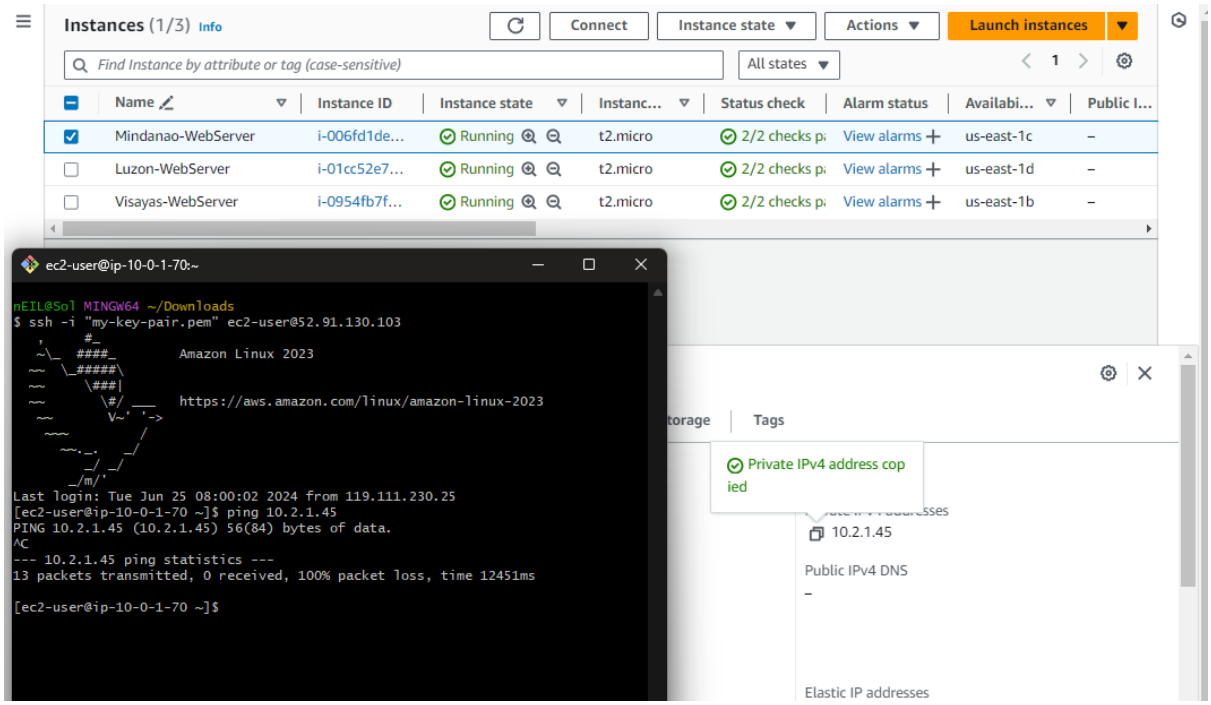
ping <Visayas-WebServer-private-IPv4-Address>



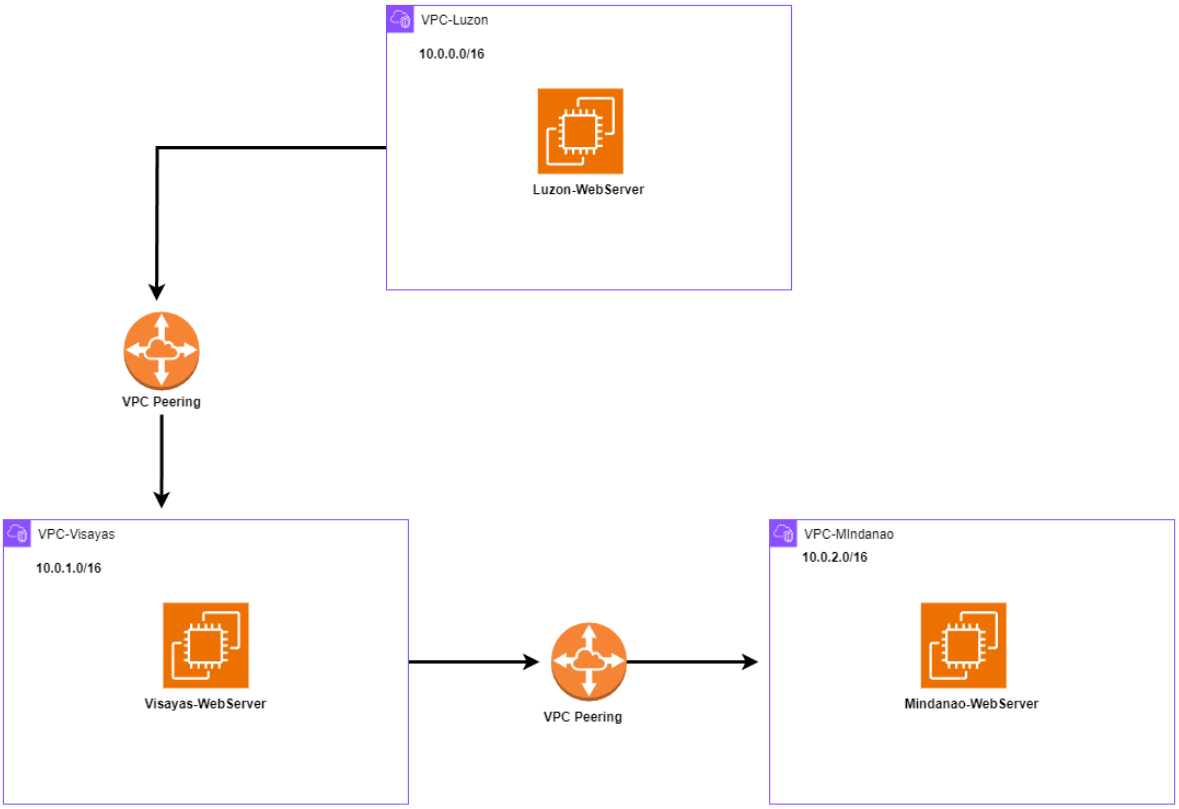
We can also received packets from Visayas-WebServer this time

-
  - Lets ping the **Mindanao-WebServer**

ping <Mindanao-WebServer-private-IPv4-Address>



We can't ping the Mindanao-WebServer. Why is that? The reason for this is because VPC Peering is **not transitive**, meaning that if VPC-Luzon is peered with VPC-Visayas, and VPC-Visayas is peered with VPC-Mindanao, VPC-Luzon cannot communicate directly with VPC-Mindanao through VPC-Visayas. Each peering connection is a one-to-one relationship that does not automatically extend beyond the two VPCs directly involved.



That's it! Congratulations! This lab has effectively demonstrated the setup and testing of VPC Peering across multiple VPCs, illustrating how resources in different VPCs can communicate securely and efficiently. Throughout the lab, you've configured and verified network connections, thereby acquiring practical skills in managing complex network architectures on AWS. This prepares you for real-world scenarios that involve interconnected VPC environments.

One critical insight from this exercise is the non-transitive nature of VPC Peering. As observed, while VPC-Luzon could communicate with VPC-Visayas and VPC-Visayas could interact with VPC-Mindanao, there was no direct communication between VPC-Luzon and VPC-Mindanao. This limitation highlights that each VPC Peering connection is an isolated, one-to-one relationship and does not inherently allow indirect routing through a third VPC. Understanding this characteristic is essential for network architects in planning and structuring AWS environments, ensuring that connectivity requirements are met efficiently without relying on transitive peering capabilities.

By internalizing the constraints and capabilities of VPC Peering, you can better design solutions that are both scalable and secure, addressing complex networking challenges with strategic configurations.