VPC Peering

By Chisom Uketui

Introducing Today's Project!

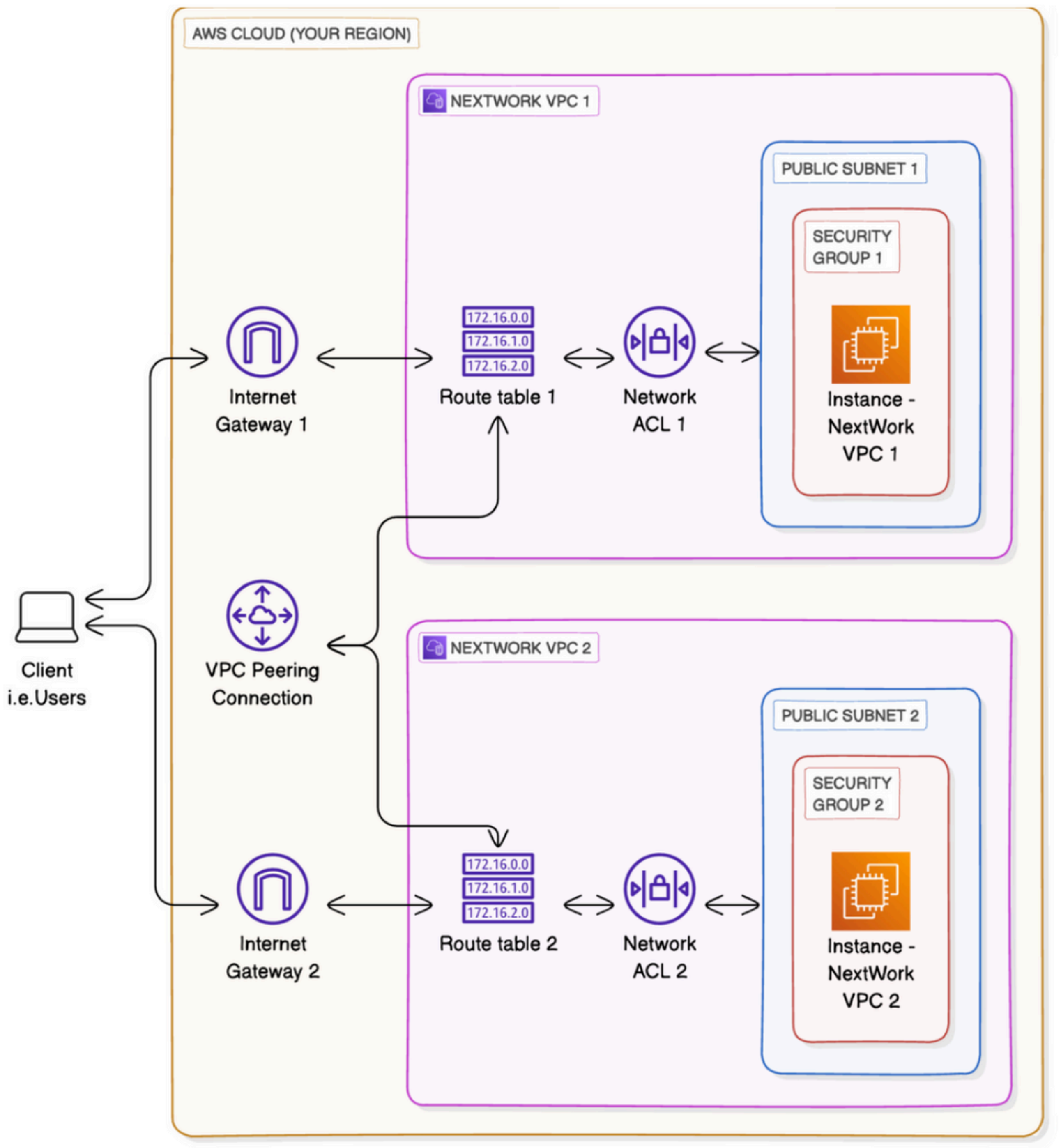
What is Amazon VPC?
Amazon Virtual Private Cloud (VPC) is a service that lets you create a logically isolated, customizable network within the AWS cloud. It functions as your private data center in the cloud, providing control over the network's virtual environment.

How I used Amazon VPC in this project
I used VPC today to set up a multi-VPC architecture (setting up 2 VPCs), created a peering connection between them, and updated my security groups to run a successful connectivity test to validate my VPC peering connection setup.

One thing I didn't expect in this project was...
I didn't expect to need a public IPv4 IP address for Instance Connect to work.
Also, I didn’t expect that Elastic IPs can assign public IPv4 addresses to resources.

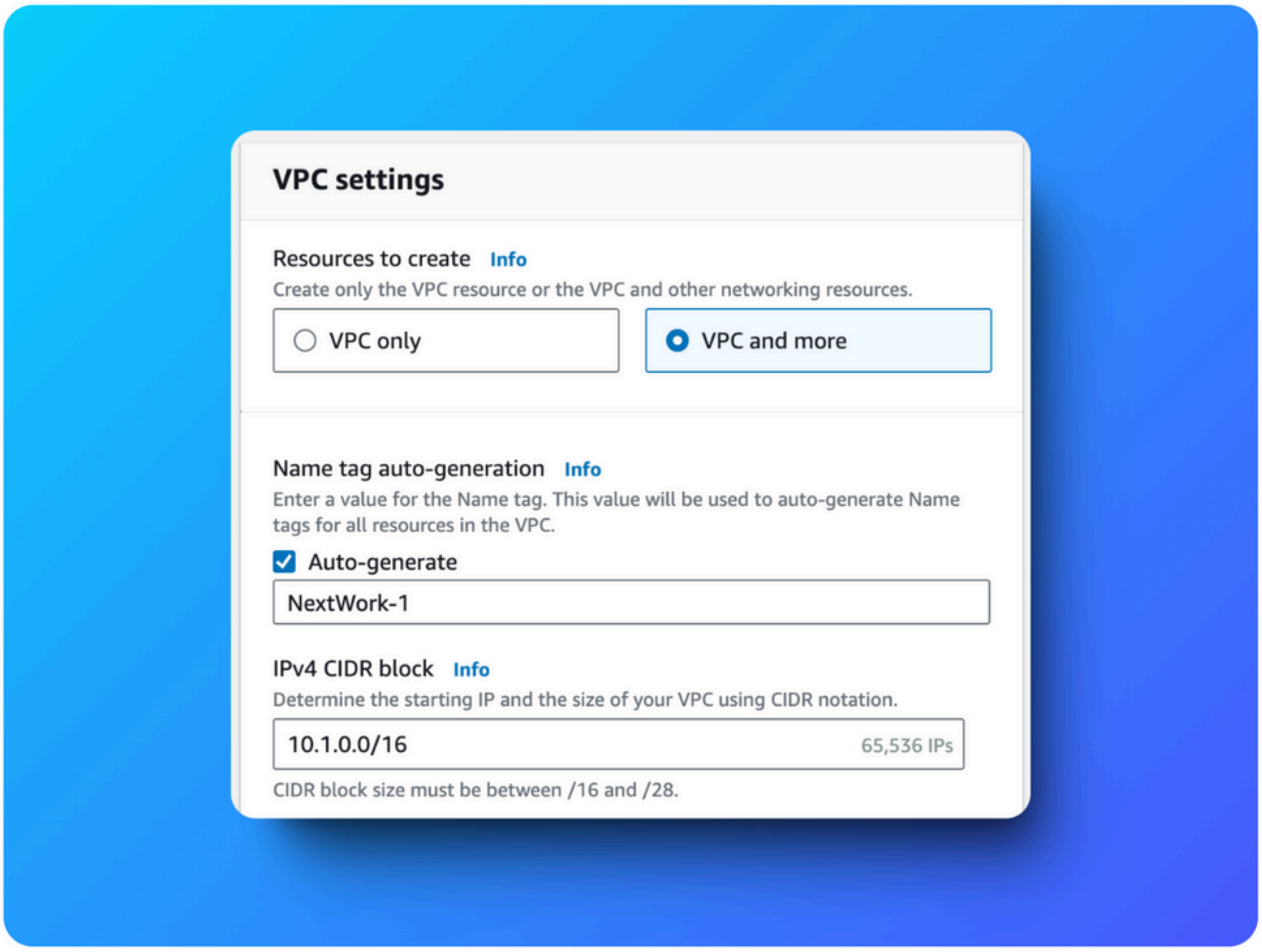
This project took me...
This project took me about an hour to complete.

Below is an overview of what I will be doing in this project.



In the first part of my project...

Step 1 – Set up my VPC
In this step, I am going to create 2 VPCs using the VPC resource map, which is a fast and easier way to create VPCs in AWS.


VPC settings

Resources to create Info

Create only the VPC resource or the VPC and other networking resources.

VPC only [✓ VPC and more]

Name tag auto-generation Info

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

✓ Auto-generate

NextWork-1

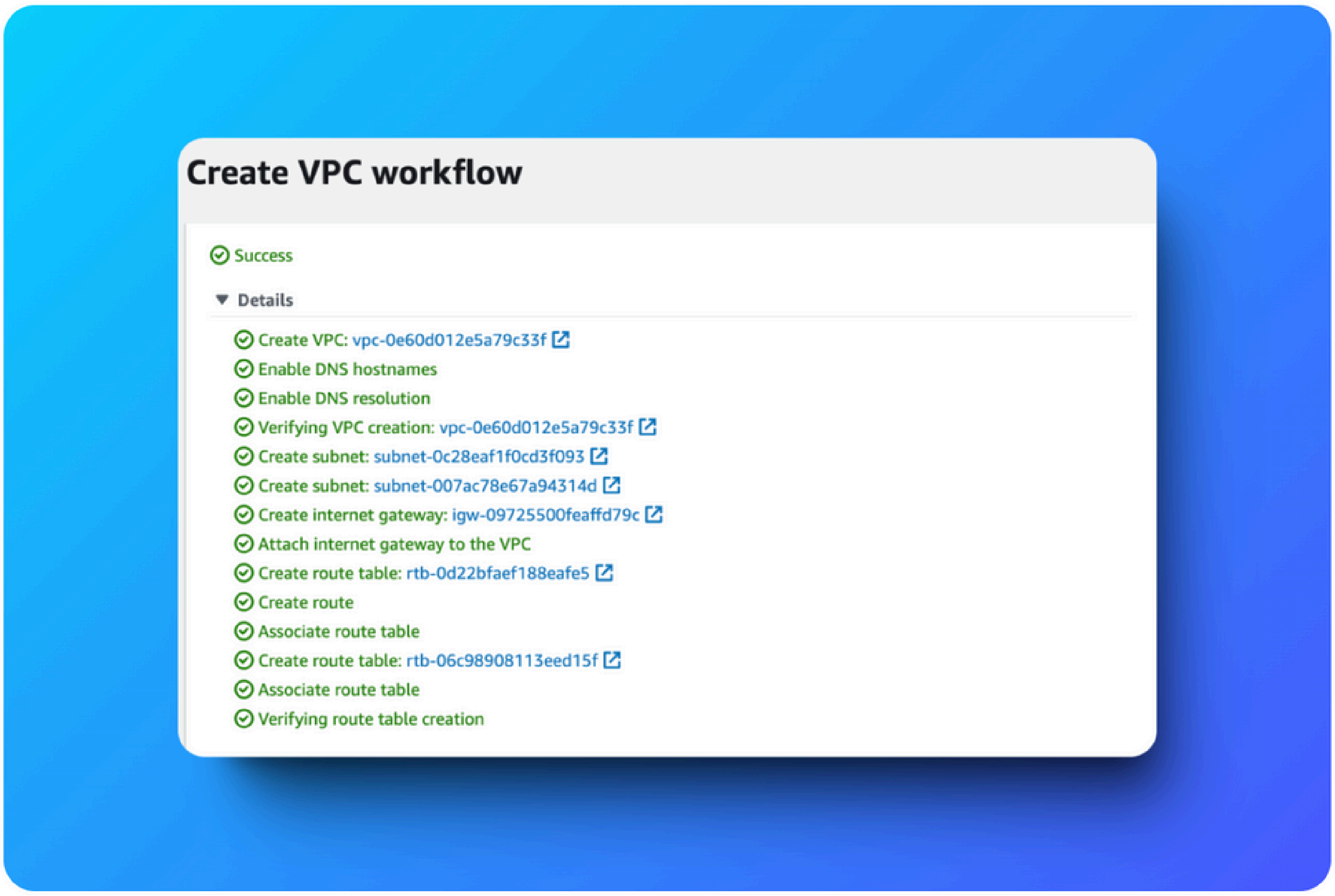
IPv4 CIDR block Info

Determine the starting IP and the size of your VPC using CIDR notation.

10.1.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

Step 1: Select VPC and more.



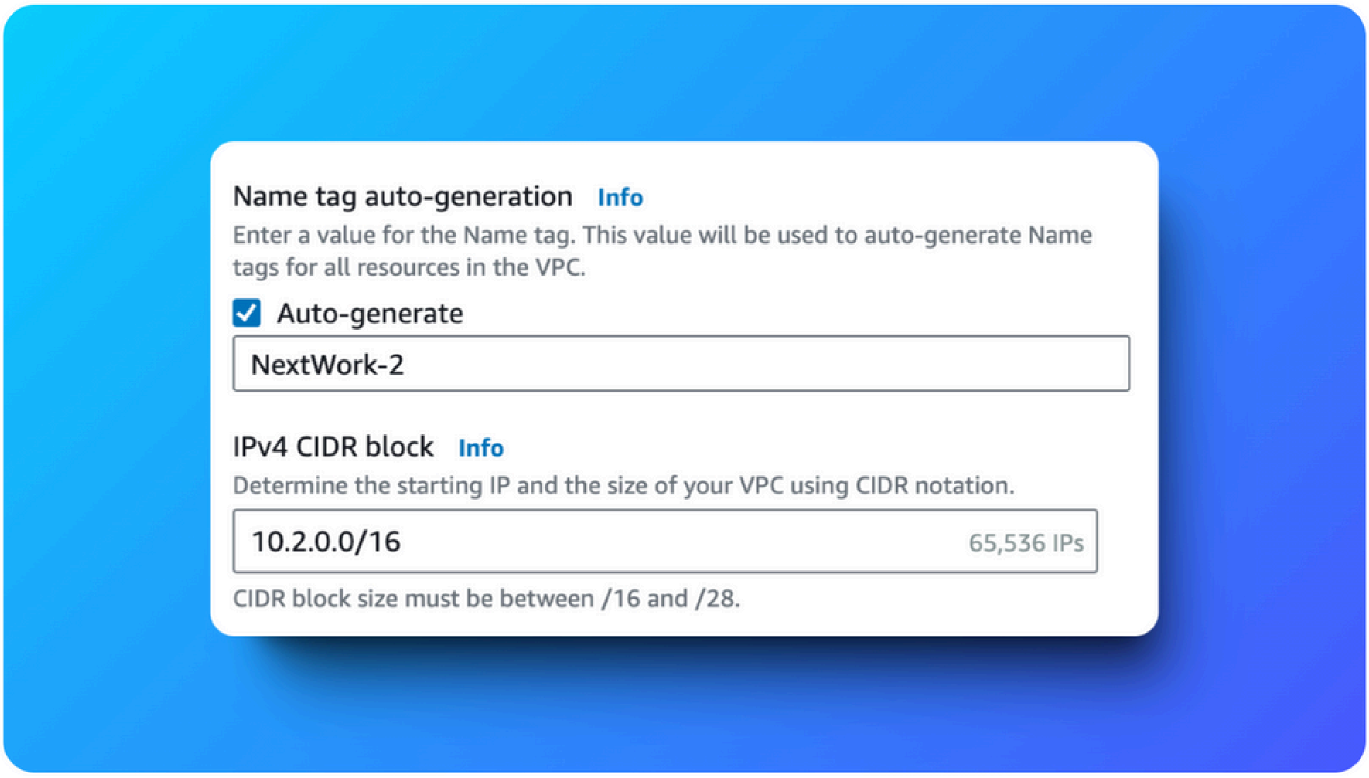
Create VPC workflow

Success

Details

- Create VPC: vpc-0e60d012e5a79c33f
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: vpc-0e60d012e5a79c33f
- Create subnet: subnet-0c28eaf1f0cd3f093
- Create subnet: subnet-007ac78e67a94314d
- Create internet gateway: igw-09725500feaffd79c
- Attach internet gateway to the VPC
- Create route table: rtb-0d22bfaef188eafe5
- Create route
- Associate route table
- Create route table: rtb-06c98908113eed15f
- Associate route table
- Verifying route table creation

VPC 2 using a different name tag and CIDR block.



Name tag auto-generation Info

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☑ Auto-generate

NextWork-2

IPv4 CIDR block Info

Determine the starting IP and the size of your VPC using CIDR notation.

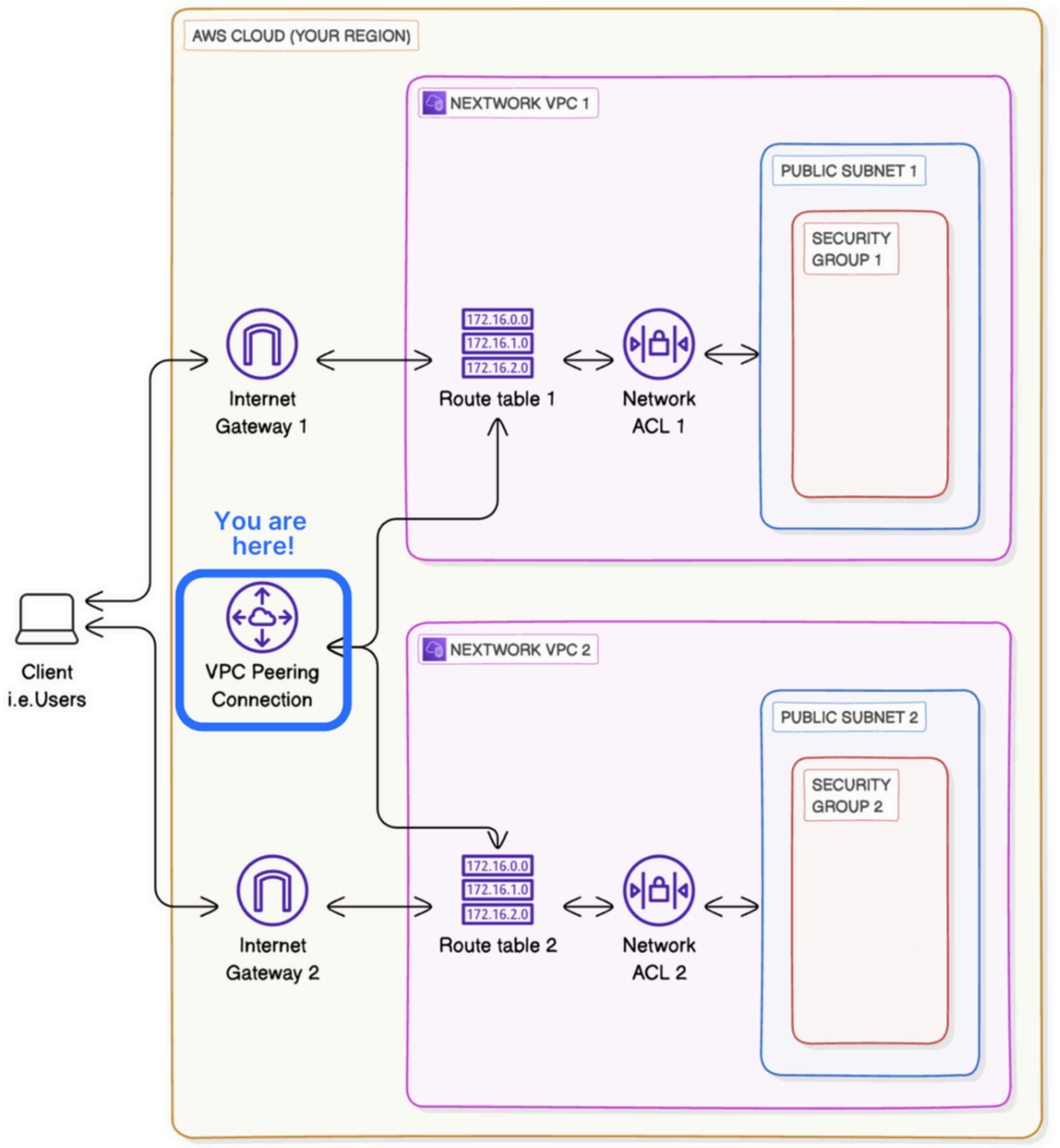
10.2.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

Step 1: Use a new name tag and CIDR block.

Step 2 – Create a Peering Connection

Here, I am setting up a VPC peering connection, which is a VPC component designed to directly connect two VPCs together.

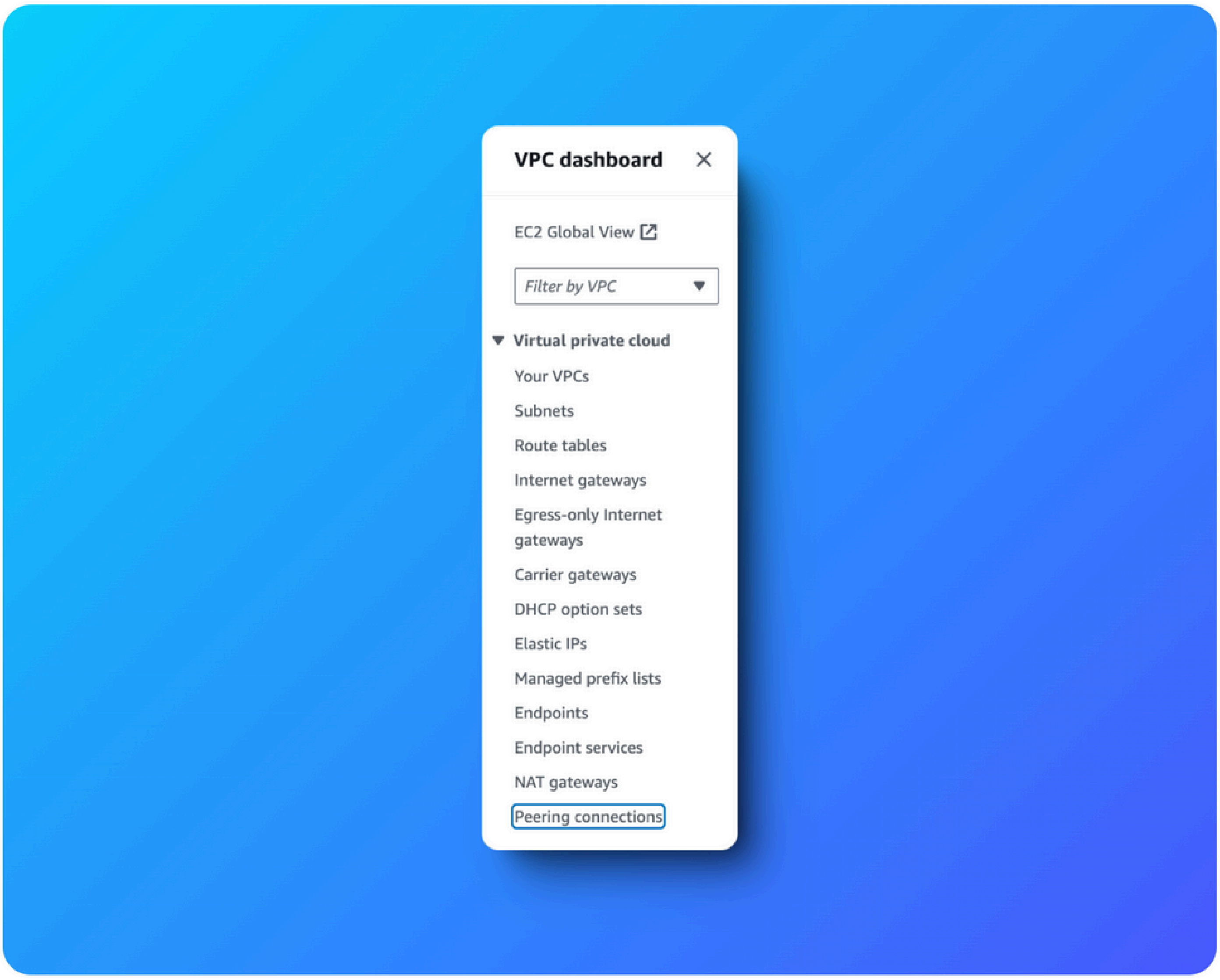


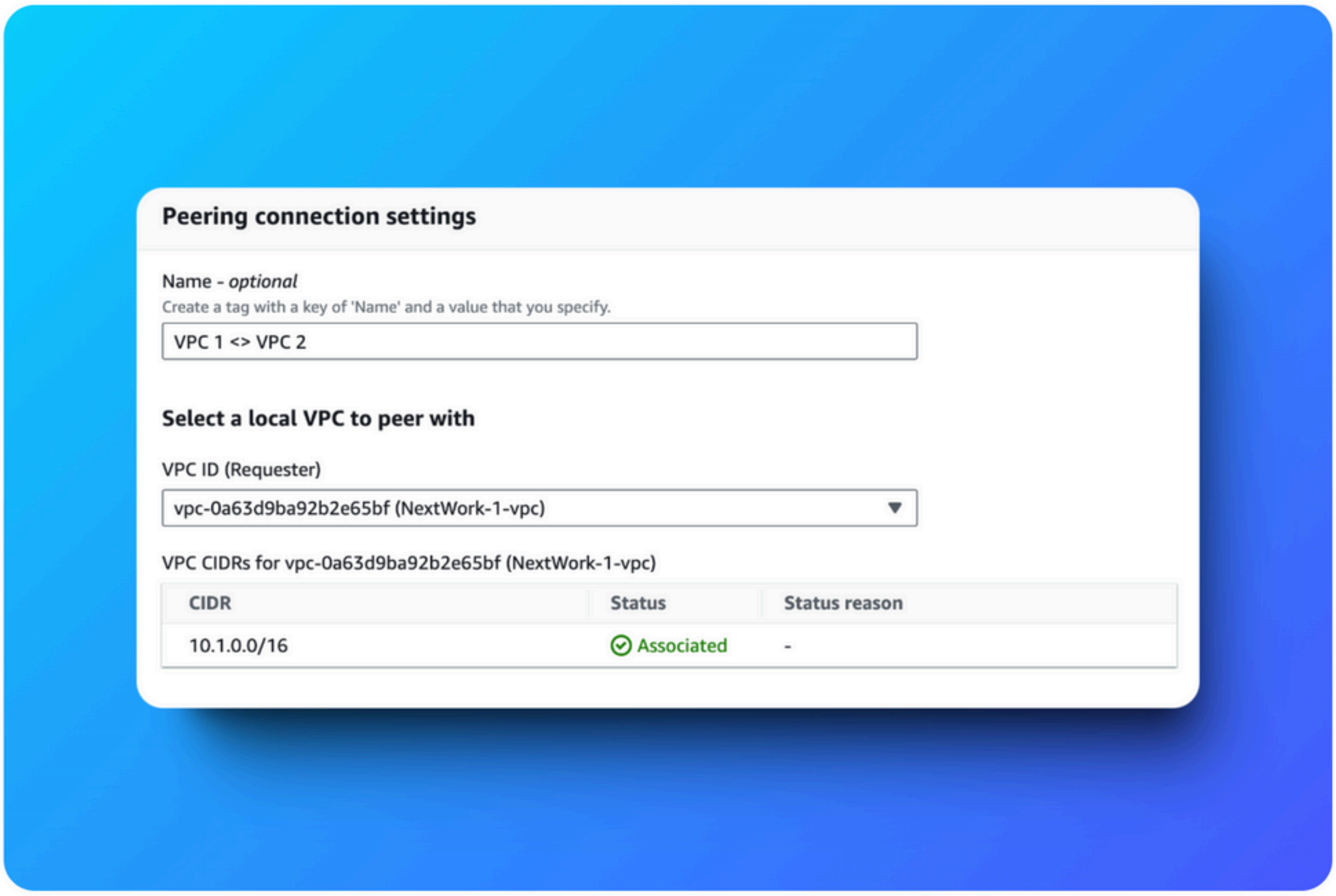
A VPC peering connection is a direct connection between two VPCs.

A peering connection lets VPCs and their resources route traffic between them using their **private** IP addresses. This means data can now be transferred between VPCs without going through the public internet.

VPCs would use peering connections to address specific networking needs in cloud environments where VPCs need to share resources, exchange data, or collaborate without exposing traffic to the public internet.

Without a peering connection, data transfers between VPCs would use resources' public address - meaning VPCs have to communicate over the public internet.





Peering connection settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
VPC 1 <> VPC 2

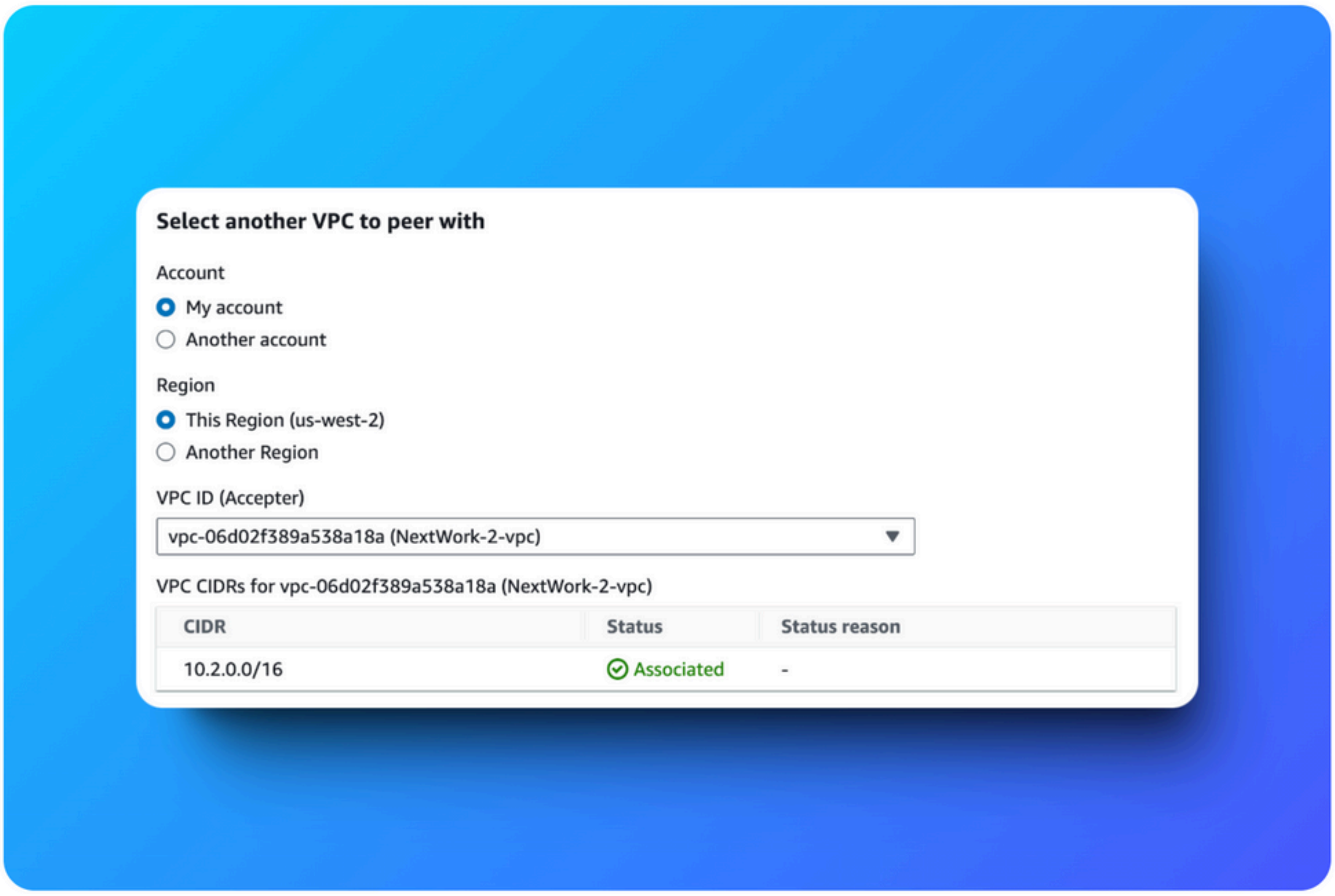
Select a local VPC to peer with

VPC ID (Requester)
vpc-0a63d9ba92b2e65bf (NextWork-1-vpc)

VPC CIDRs for vpc-0a63d9ba92b2e65bf (NextWork-1-vpc)

CIDR	Status	Status reason
10.1.0.0/16	✓ Associated	-

Step 2: Set the Requester.



Select another VPC to peer with

Account
[✓ My account]
Another account

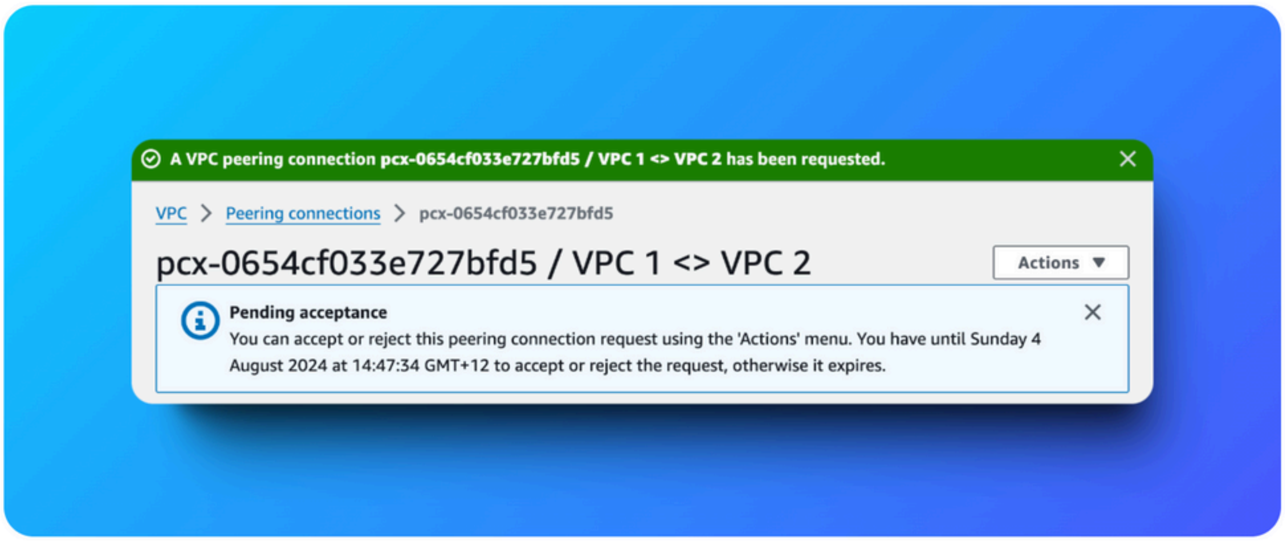
Region
[✓ This Region (us-west-2)]
Another Region

VPC ID (Accepter)
vpc-06d02f389a538a18a (NextWork-2-vpc)

VPC CIDRs for vpc-06d02f389a538a18a (NextWork-2-vpc)

CIDR	Status	Status reason
10.2.0.0/16	✓ Associated	-

Step 2: Set the Accepter.

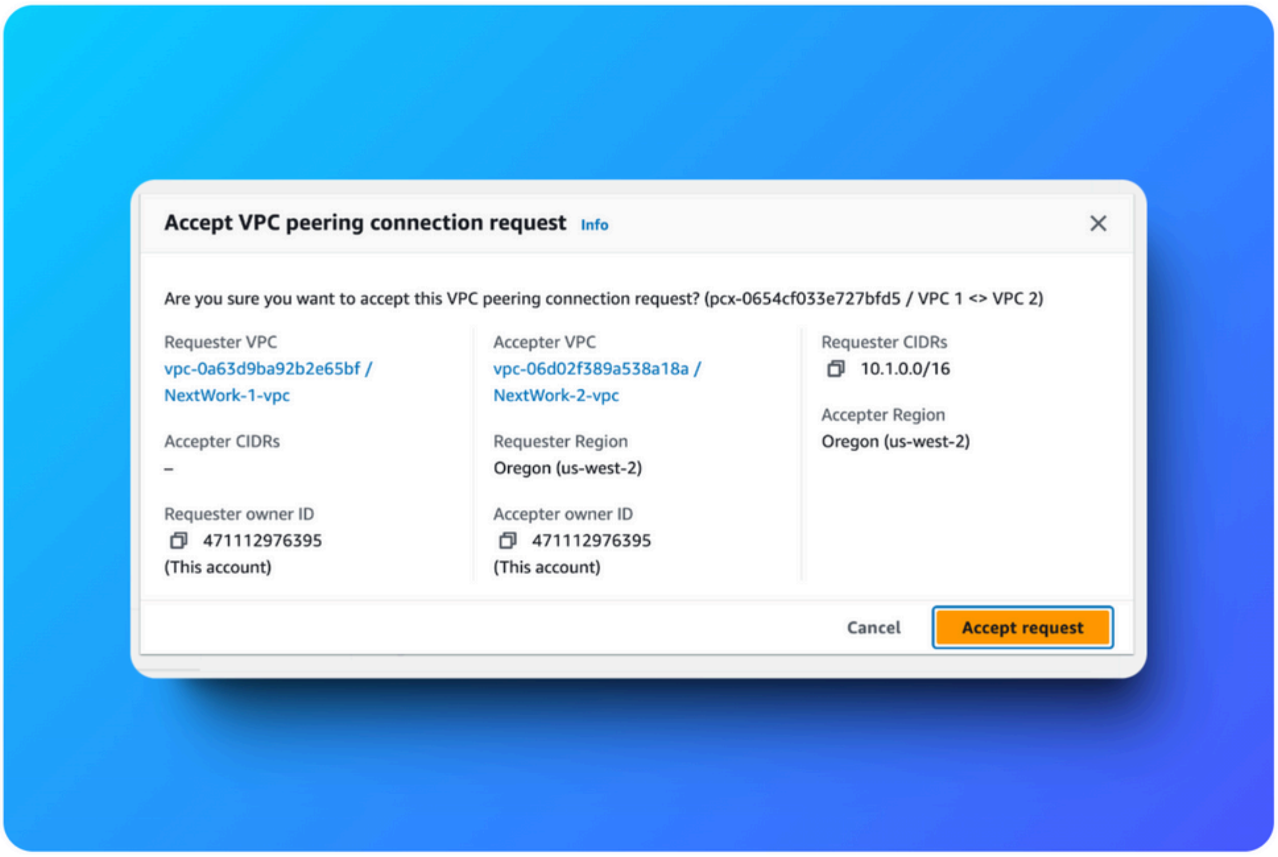

A VPC peering connection pcx-0654cf033e727bfd5 / VPC 1 <> VPC 2 has been requested.

VPC > Peering connections > pcx-0654cf033e727bfd5

pcx-0654cf033e727bfd5 / VPC 1 <> VPC 2

Pending acceptance
You can accept or reject this peering connection request using the 'Actions' menu. You have until Sunday 4 August 2024 at 14:47:34 GMT+12 to accept or reject the request, otherwise it expires.

Step 2: A peering connection is requested.


Accept VPC peering connection request Info

Are you sure you want to accept this VPC peering connection request? (pcx-0654cf033e727bfd5 / VPC 1 <> VPC 2)

Requester VPC
vpc-0a63d9ba92b2e65bf / NextWork-1-vpc

Accepter VPC
vpc-06d02f389a538a18a / NextWork-2-vpc

Requester CIDRs
10.1.0.0/16

Accepter Region
Oregon (us-west-2)

Accepter CIDRs
–

Requester Region
Oregon (us-west-2)

Requester owner ID
471112976395 (This account)

Accepter owner ID
471112976395 (This account)

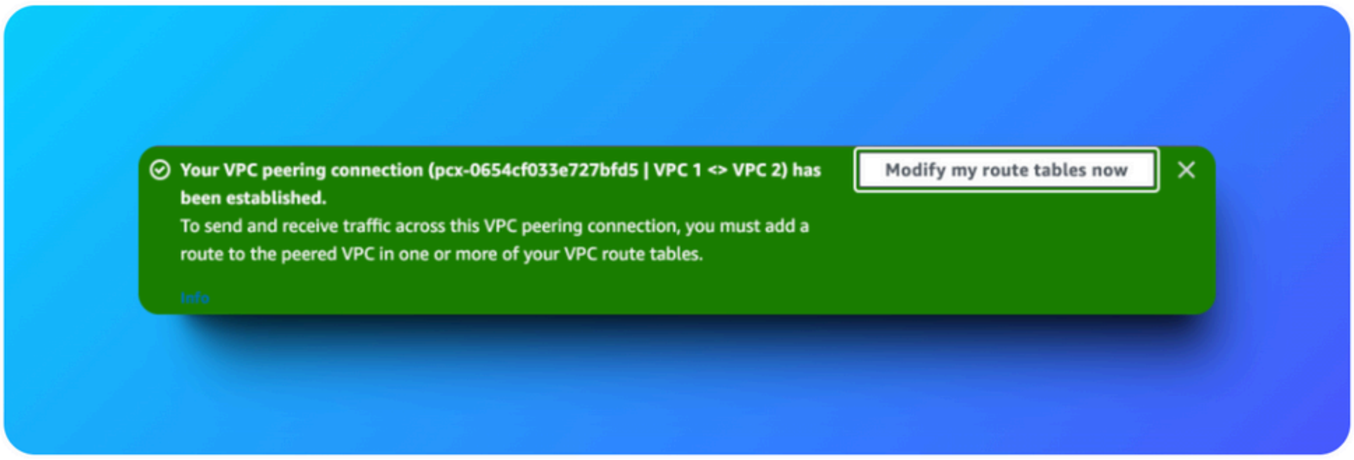
Step 2: Select Accept request here.

In VPC peering, the **Requester** is the VPC that initiates a peering connection. As the requester, they will be sending the other VPC an **invitation** to connect!

The **Accepter** is the VPC that receives a peering connection request! The Accepter can either accept or decline the invitation. This means the peering connection isn't actually made until the other VPC also agrees to it!

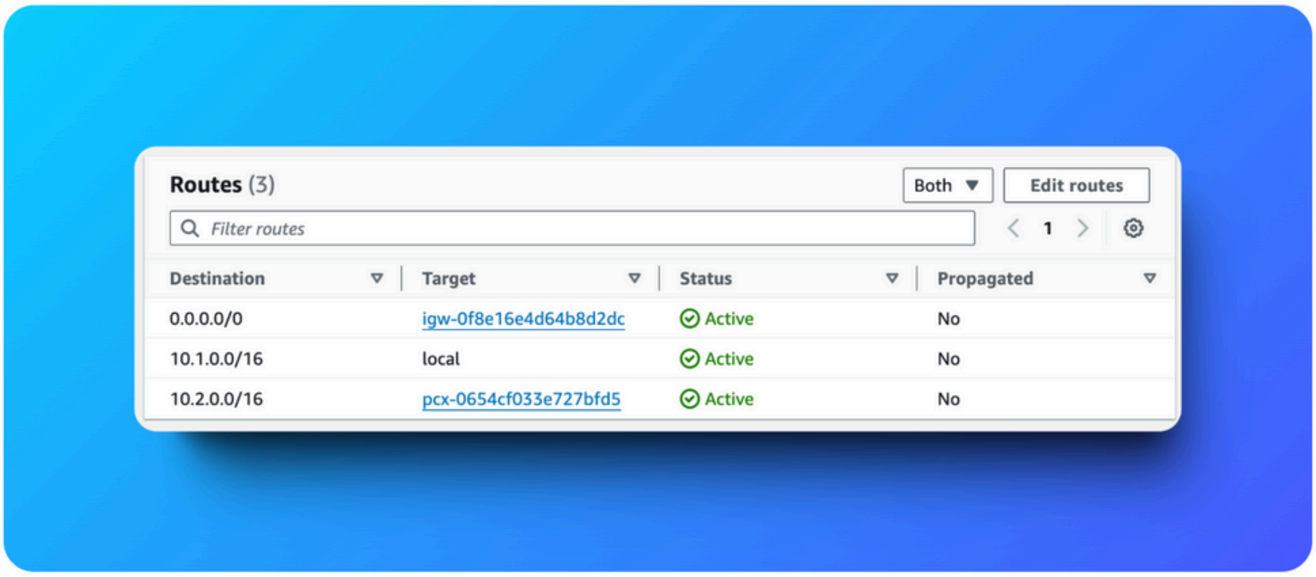
Step 3 – Update Route Tables

In this step, I am going to update the route tables to set up a way for traffic coming from VPC 1 to get to VPC 2 and vice versa.

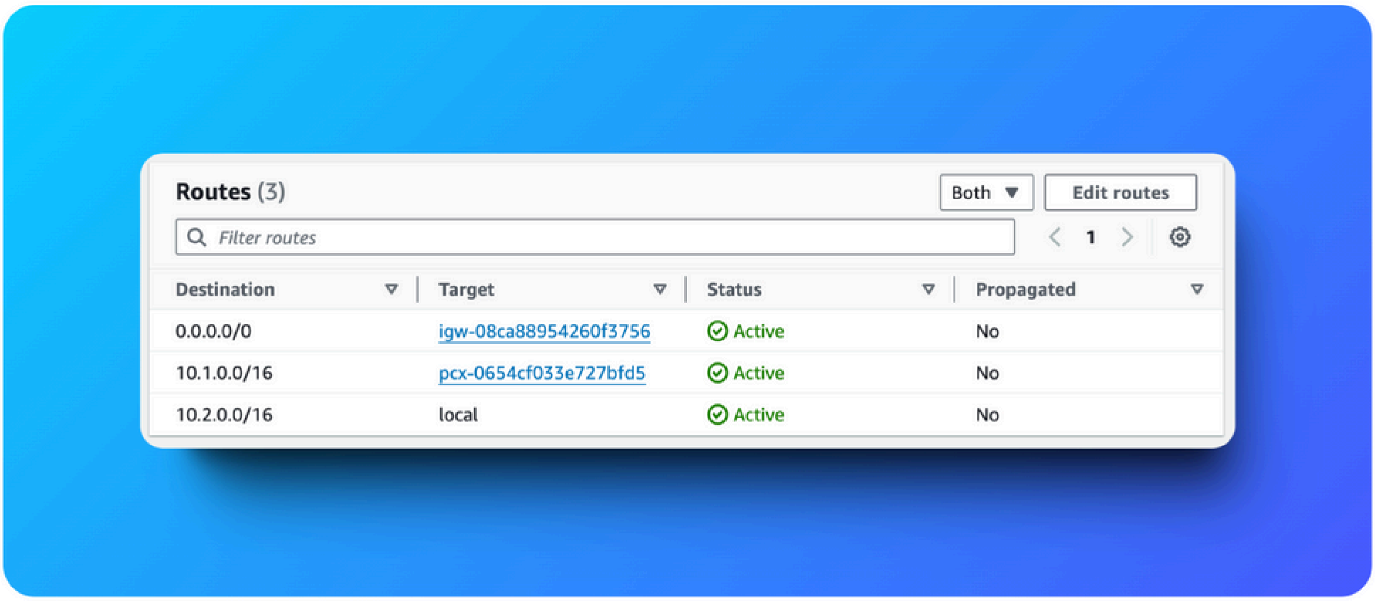

Your VPC peering connection (pcx-0654cf033e727bfd5 | VPC 1 <> VPC 2) has been established.
To send and receive traffic across this VPC peering connection, you must add a route to the peered VPC in one or more of your VPC route tables.

Step 2: Select Modify my route tables now.

Even if my peering connection has been accepted, traffic in VPC 1 won't know how to get to resources in VPC 2 without a route in my route table! I need to set up a route that directs traffic bound for VPC 2 to the peering connection I've set up.


Routes (3)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0f8e16e4d64b8d2dc	Active	No
10.1.0.0/16	local	Active	No
10.2.0.0/16	pcx-0654cf033e727bfd5	Active	No



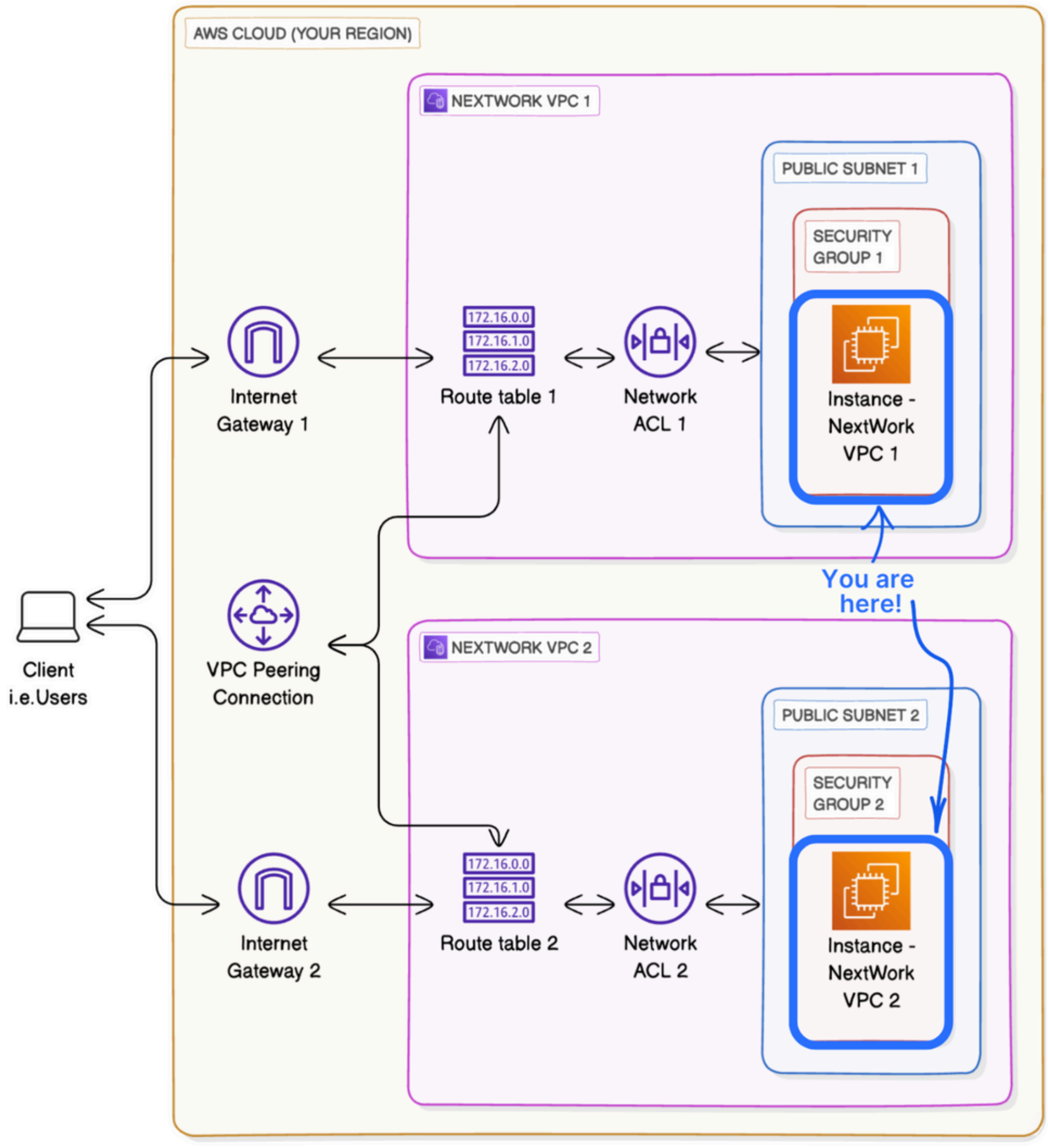
Destination	Target	Status	Propagated
0.0.0.0/0	igw-08ca88954260f3756	✓ Active	No
10.1.0.0/16	pcx-0654cf033e727bfd5	✓ Active	No
10.2.0.0/16	local	✓ Active	No

Step 3: Update VPC 2's route table too.

My VPCs' route tables need to be updated because the default route table doesn’t have a route using the peering connection yet; this needs to be set up so that resources can be directed to the peering connection when trying to reach the other VPC.

My VPCs' new routes have a destination of the other VPC's CIDR block. The routes' target was the peering connection I set up.

Step 4 – Launch EC2 Instances
I am launching EC2 instances in each of the VPCs (VPC 1 and VPC 2) so that I can directly connect with my instances later and test my VPC connection.



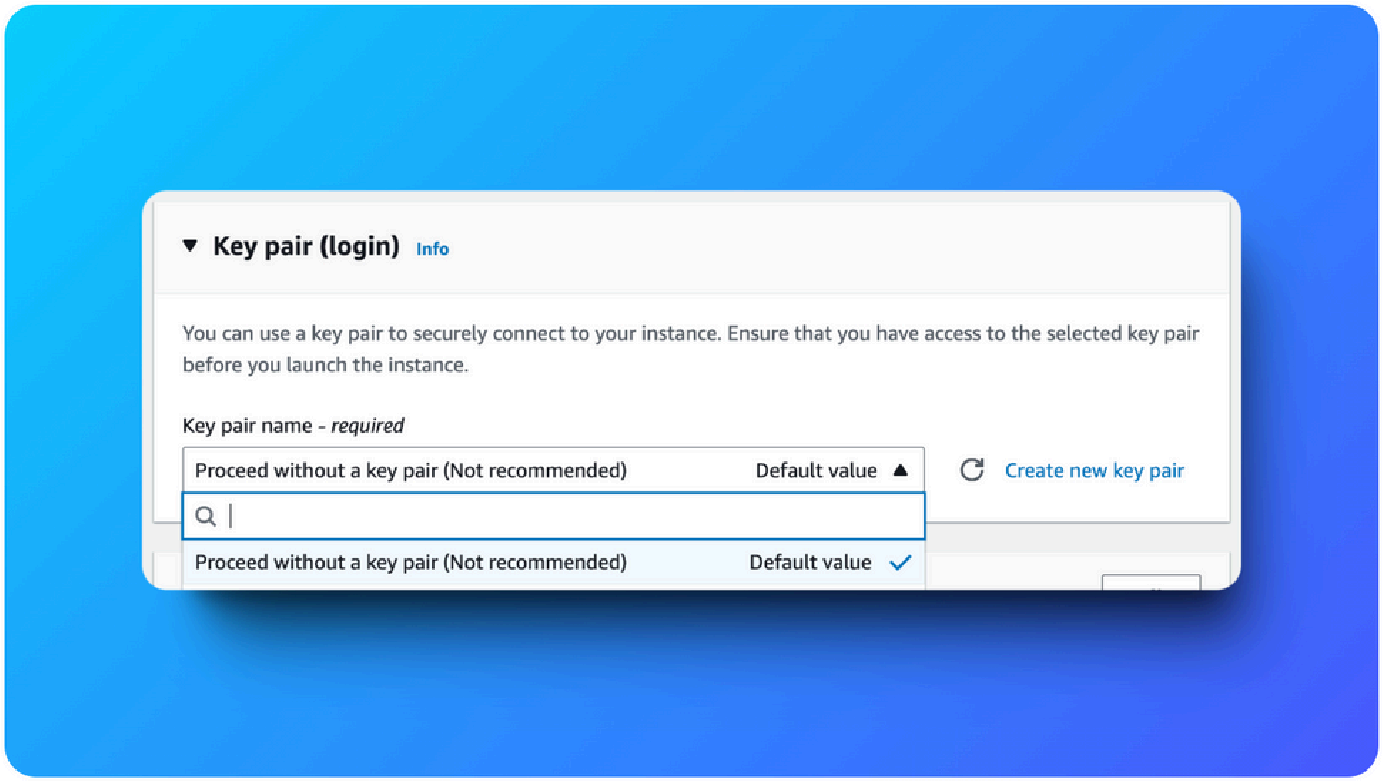
Multi-VPC Architecture

I started my project by launching two VPCs - they have unique CIDR blocks, and they each have 1 public subnet.

The CIDR blocks for VPCs 1 and 2 are 10.1.0.0/16 and 10.2.0.0/16, respectively. They have to be unique because once you set up a VPC peering connection, route tables need unique addresses for correct routing across VPCs.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances as I'm using EC2 Instance Connect to directly connect to my EC2 instance. When using this connection type to EC2, AWS actually manages a key pair for us! We don't need to manage key pairs ourselves.



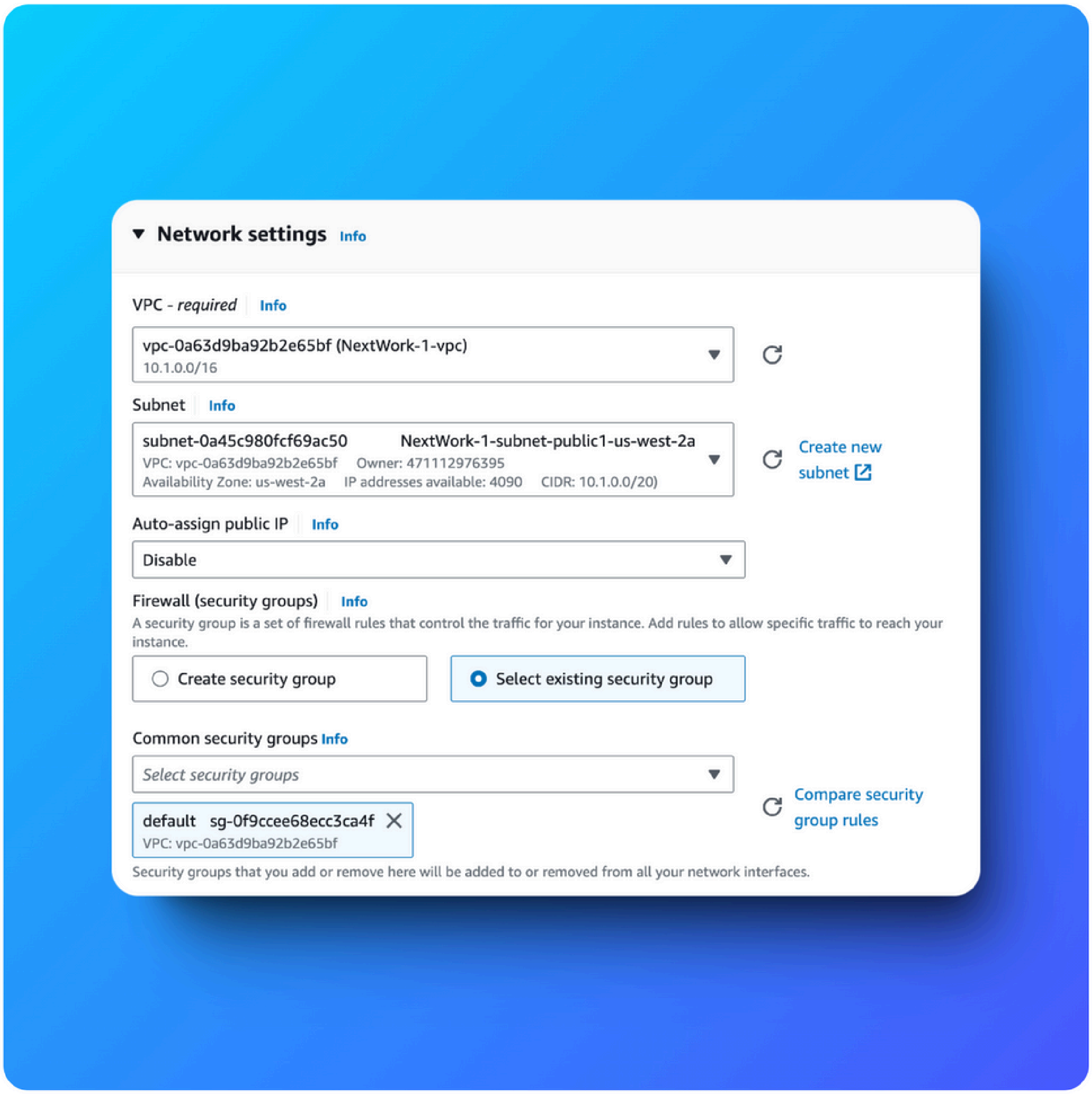
Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Proceed without a key pair (Not recommended) Default value

Proceed without a key pair (Not recommended) Default value ✓



Network settings Info

VPC - *required* Info

vpc-0a63d9ba92b2e65bf (NextWork-1-vpc)
10.1.0.0/16

Subnet Info

subnet-0a45c980fcf69ac50 NextWork-1-subnet-public1-us-west-2a
VPC: vpc-0a63d9ba92b2e65bf Owner: 471112976395
Availability Zone: us-west-2a IP addresses available: 4090 CIDR: 10.1.0.0/20)

Create new subnet

Auto-assign public IP Info

Disable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group [✓ Select existing security group]

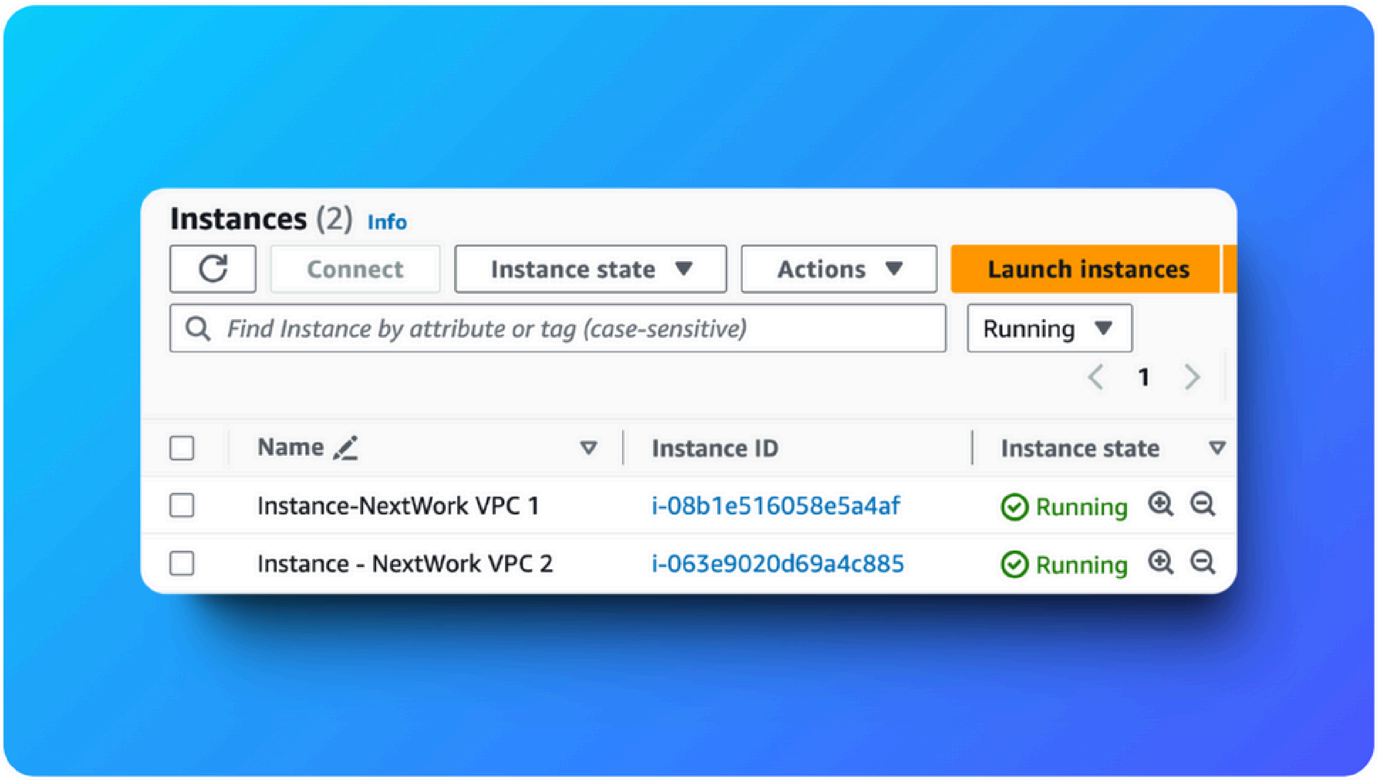
Common security groups Info

Select security groups

default sg-0f9ccee68ecc3ca4f
VPC: vpc-0a63d9ba92b2e65bf

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

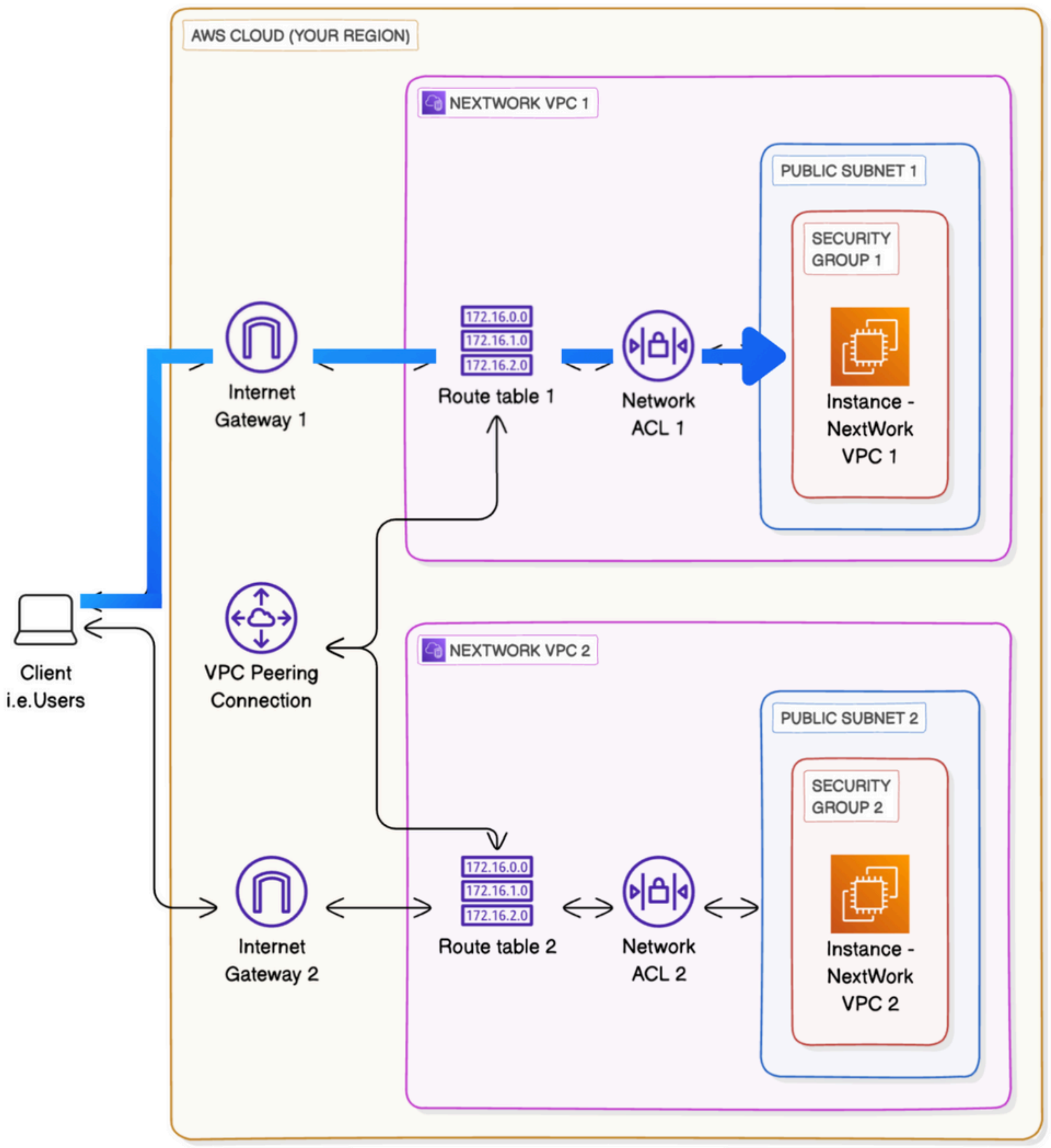


Name	Instance ID	Instance state
Instance-NextWork VPC 1	i-08b1e516058e5a4af	Running
Instance - NextWork VPC 2	i-063e9020d69a4c885	Running

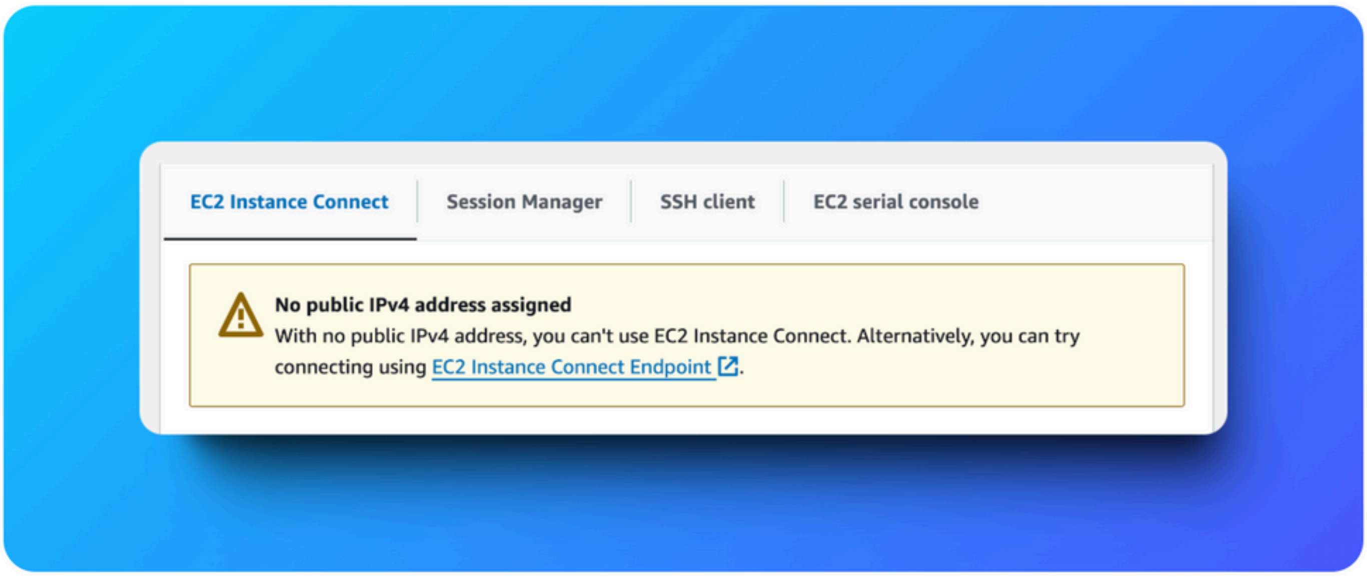
In the second part of my project...

Step 5 – Use EC2 Instance Connect

In this step, I will use EC2 Instance Connect to connect to my first EC2 instance.
I need to do this as I will be using this EC2 instance for connectivity tests later in this project.

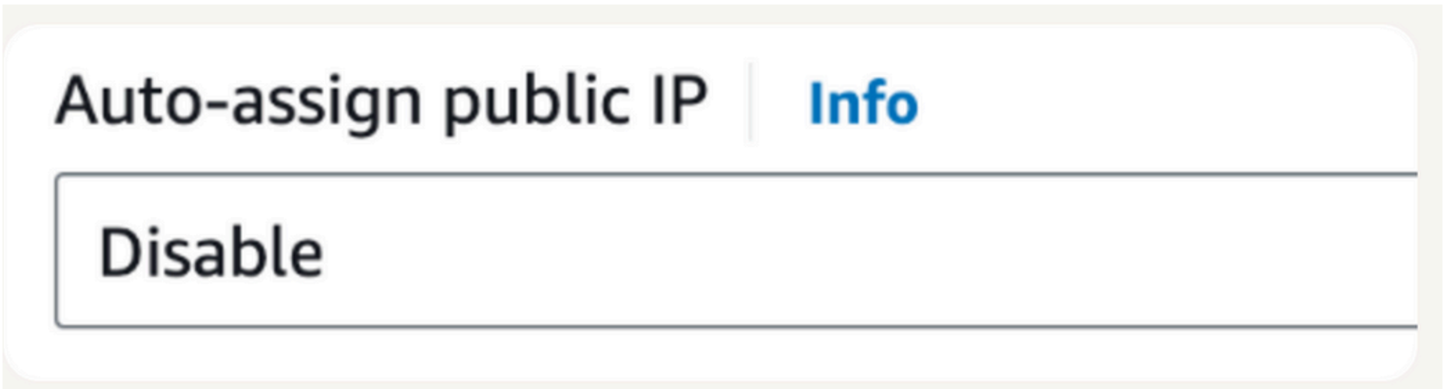


Oooooop!! An error!

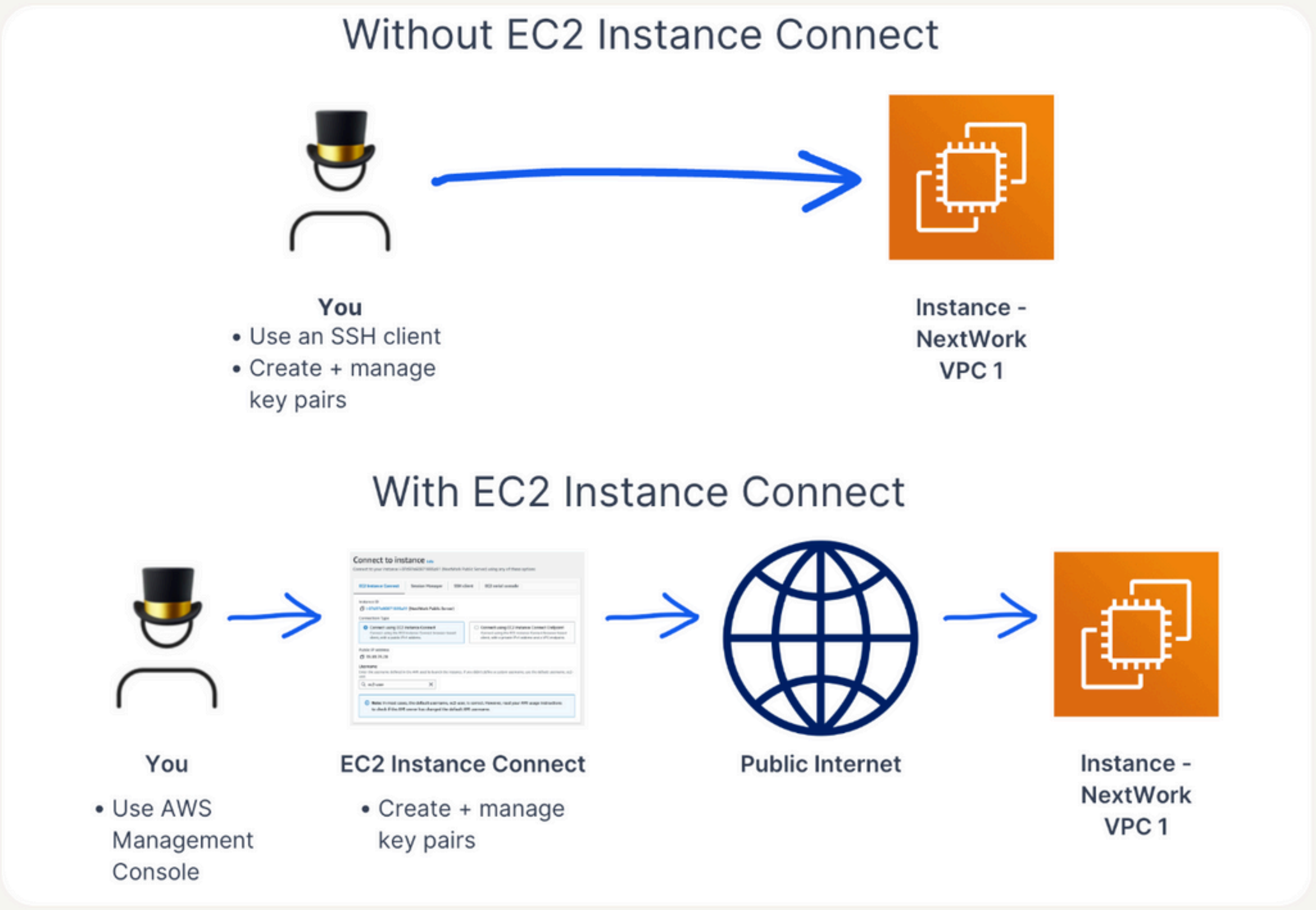


No public IPv4 address assigned
With no public IPv4 address, you can't use EC2 Instance Connect. Alternatively, you can try connecting using EC2 Instance Connect Endpoint.

I ran into this error because while creating my instance I disabled 'Auto assign public IP' botton. If you want to connect to your instance over EC2 Instance Connect, then your instance must have a public IP address and be in a public subnet. This is because using EC2 Instance Connect connects to your server **over the internet** by default.


Auto-assign public IP | Info
Disable

Step 5: We didn't assign a public IP address in the previous step 🙉

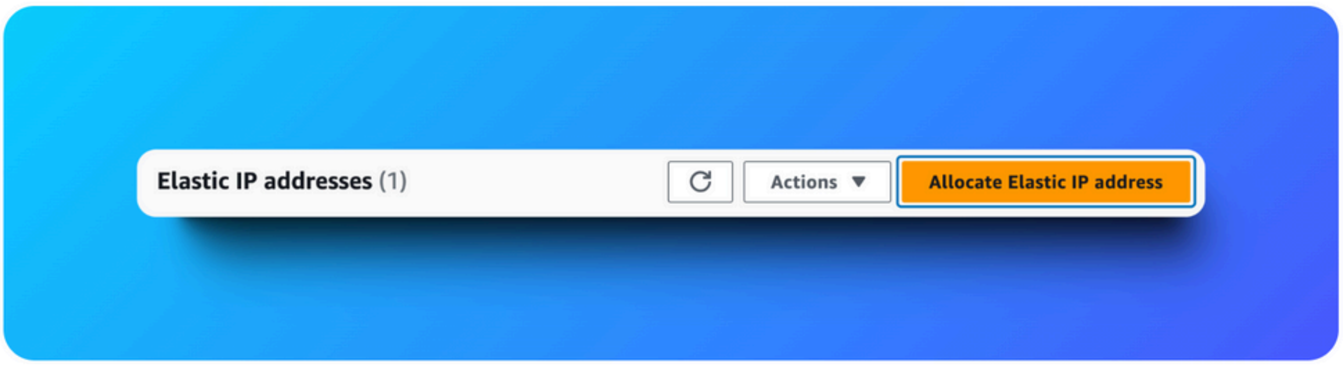


Without EC2 Instance Connect
You
• Use an SSH client
• Create + manage key pairs
Instance - NextWork VPC 1
With EC2 Instance Connect
You
• Use AWS Management Console
EC2 Instance Connect
• Create + manage key pairs
Public Internet
Instance - NextWork VPC 1

Step 5: EC2 Instance Connect works over the public internet.

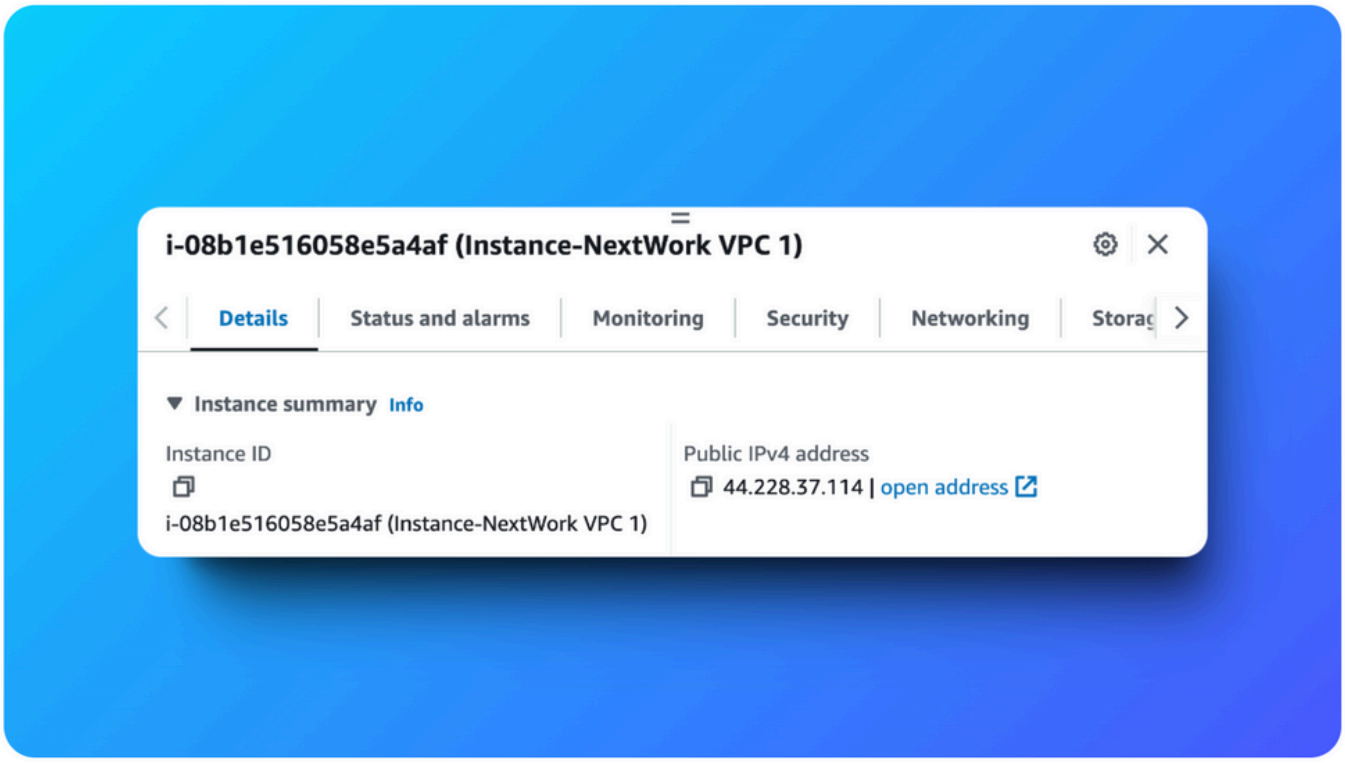
To resolve this error, I set up Elastic IP addresses. These are static, public IPv4 IPs provided by AWS to enable your resources to communicate over the internet. Unlike dynamically assigned public IP addresses, Elastic IPs are persistent.

Associating an Elastic IP address resolved the error because it gives my EC2 instance a public IP address, fulfilling the requirement for Instance Connect to work.



Step 5: Select Allocate Elastic IPs.

Nice 🙂 My EC2 now has a public IPV4 address



i-08b1e516058e5a4af (Instance-NextWork VPC 1)

Instance summary Info

Instance ID

i-08b1e516058e5a4af (Instance-NextWork VPC 1)

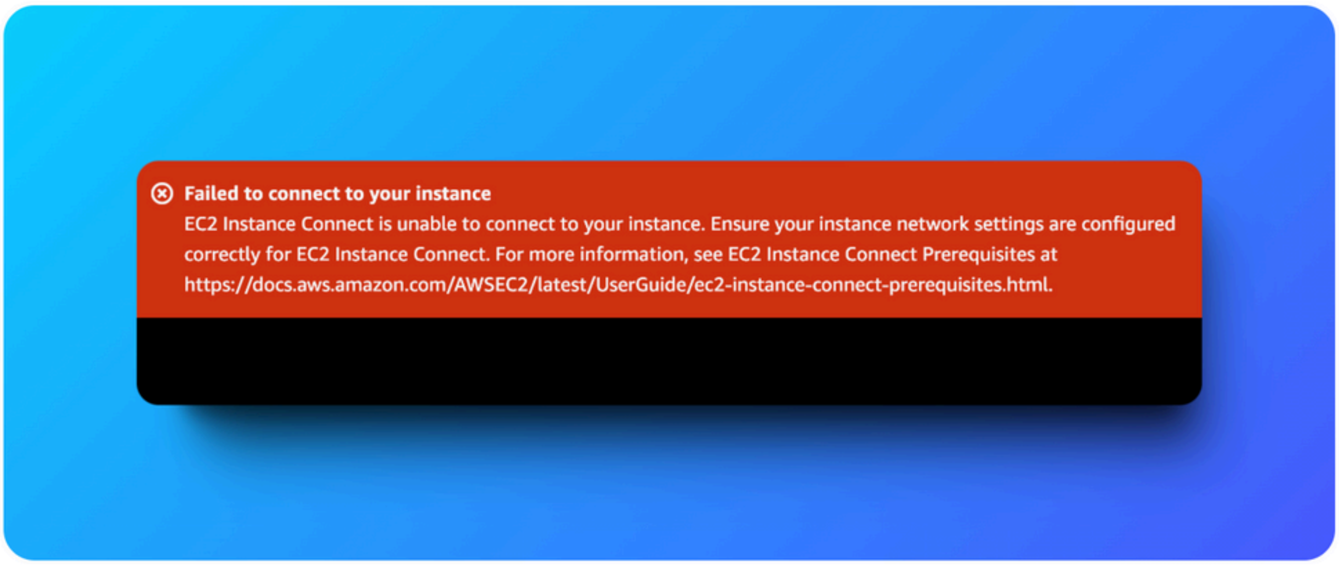
Public IPv4 address

44.228.37.114 | open address

Step 6 – Connect to EC2 Instance 1

I am re-attempting my connection to instance - Chisom VPC 1 and resolving another error preventing us from using Instance Connect to directly connect to my EC2 instance.

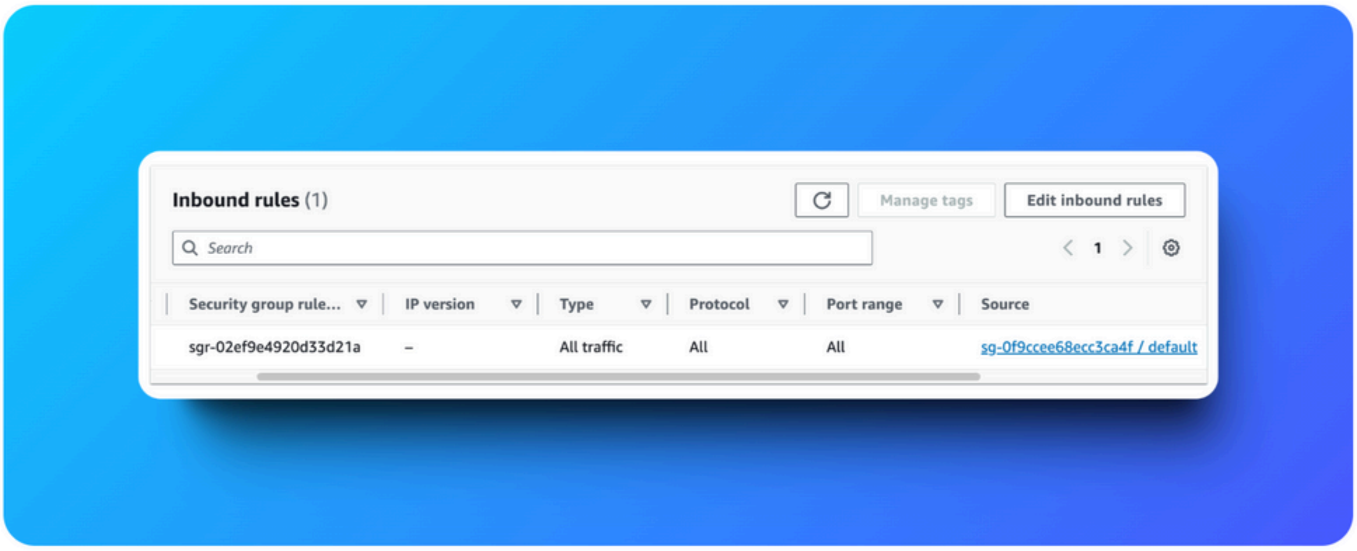
Failed to connect to my instance again 😀



Failed to connect to your instance

EC2 Instance Connect is unable to connect to your instance. Ensure your instance network settings are configured correctly for EC2 Instance Connect. For more information, see EC2 Instance Connect Prerequisites at https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-connect-prerequisites.html.

Let's trouble shoot!



Security group rule...	IP version	Type	Protocol	Port range	Source
sgr-02ef9e4920d33d21a	–	All traffic	All	All	sg-0f9ccee68ecc3ca4f / default

Step 6: Check out VPC 1's security group's inbound rules.

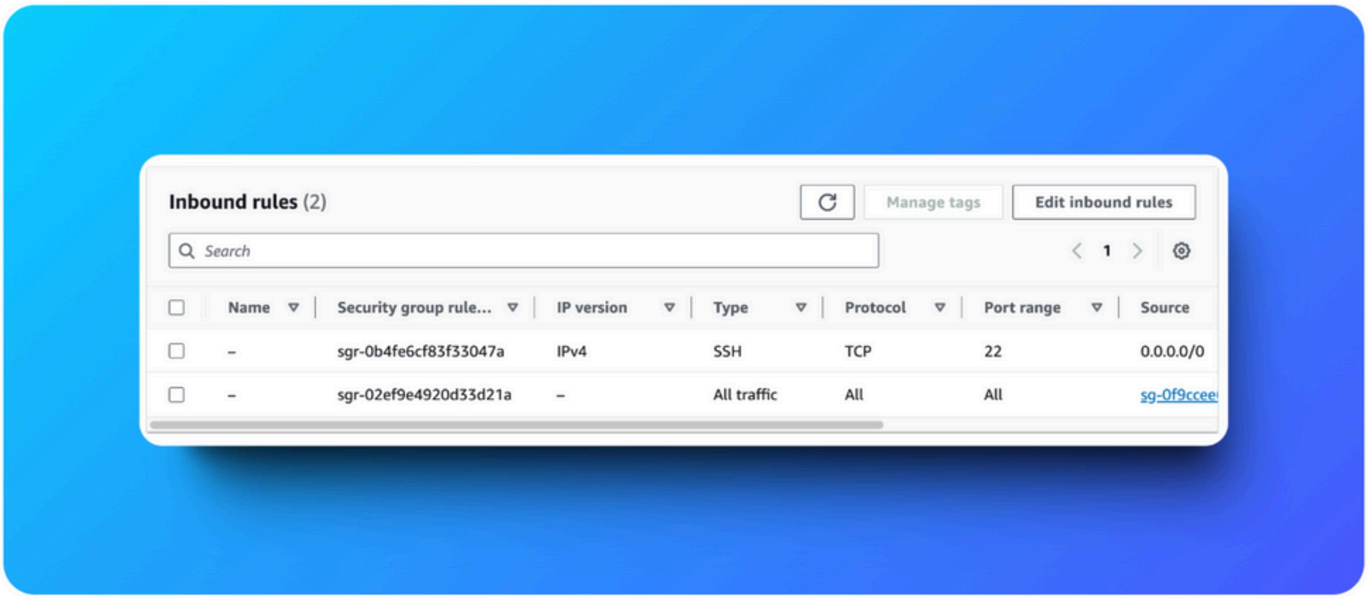
Found it!

I am trying to access Instance - VPC 1 using SSH through EC2 Instance Connect, which is trying to connect to my instance over the internet.

My default security group only allows inbound traffic from within the VPC, so traffic from the internet is being cut off!

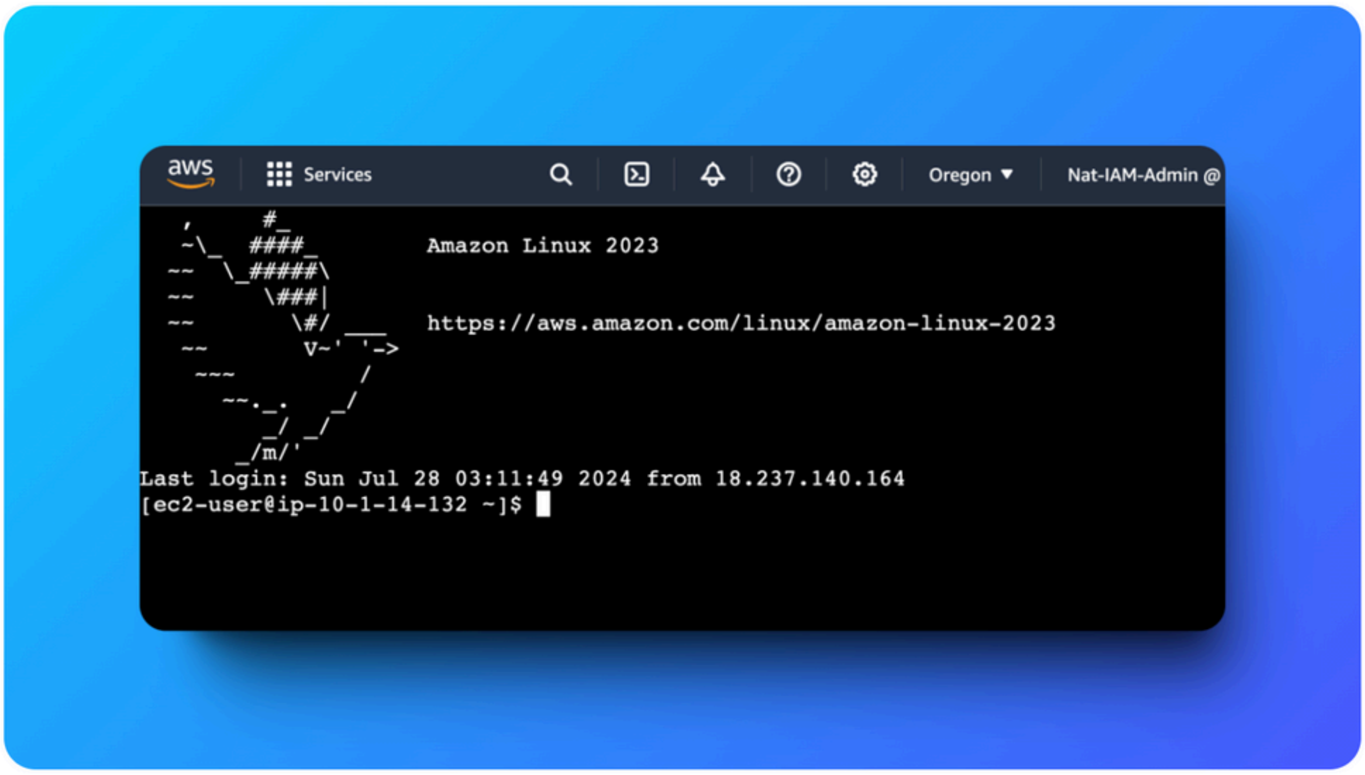
The default security group for a new VPC does not allow incoming traffic from outside of the VPC.

I have to allow inbound SSH traffic on port 22!



Name	Security group rule...	IP version	Type	Protocol	Port range	Source
–	sgr-0b4fe6cf83f33047a	IPv4	SSH	TCP	22	0.0.0.0/0
–	sgr-02ef9e4920d33d21a	–	All traffic	All	All	sg-0f9ccee

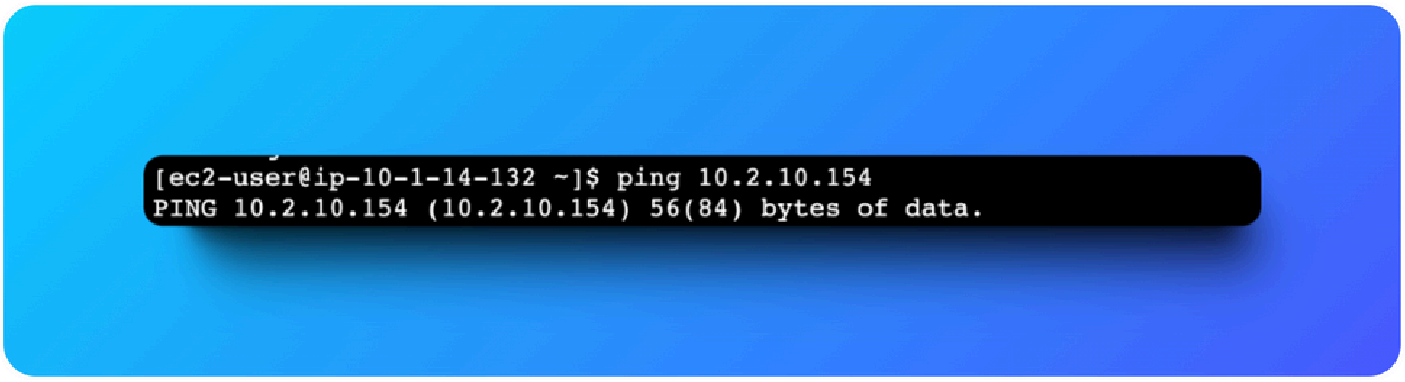
Successfully connected!



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun Jul 28 03:11:49 2024 from 18.237.140.164
[ec2-user@ip-10-1-14-132 ~]$
```

Step 7 – Test VPC Peering
In this step, I'm going to get Instance VPC1 to attempt a direct connection to Instance VPC2 to validate my peering connection is set up properly. To test VPC peering, I ran the command ping 10.2.xxxx (i.e. private IPv4 address of the other EC2 instance in VPC 2).

A successful ping test would validate my VPC peering connection because this ping test will not get any replies if the peering connection did not successfully connect the 2 VPCs.



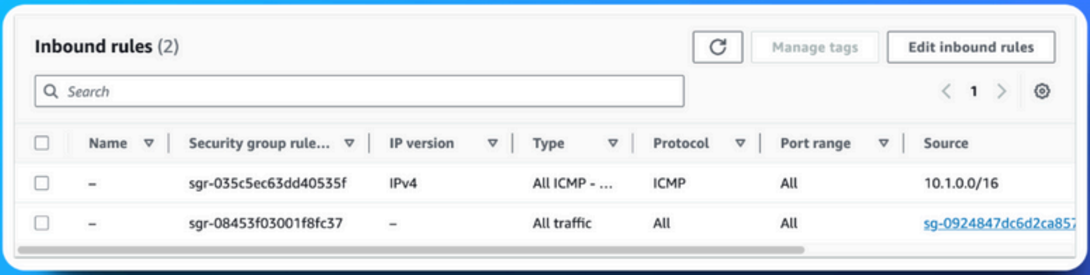
```
[ec2-user@ip-10-1-14-132 ~]$ ping 10.2.10.154
PING 10.2.10.154 (10.2.10.154) 56(84) bytes of data.
```

This single line indicates that my Instance - VPC 1 has sent out a ping message... and that's about it.

Usually, when you ping another computer successfully, you should see **several** replies back instantly. Each reply tells you how long it took for the message to go to the Instance -VPC 2 and come back.

If you don't get any replies (like my situation right now), or if the replies stop suddenly, it's usually a sign that there's a problem with the connection.

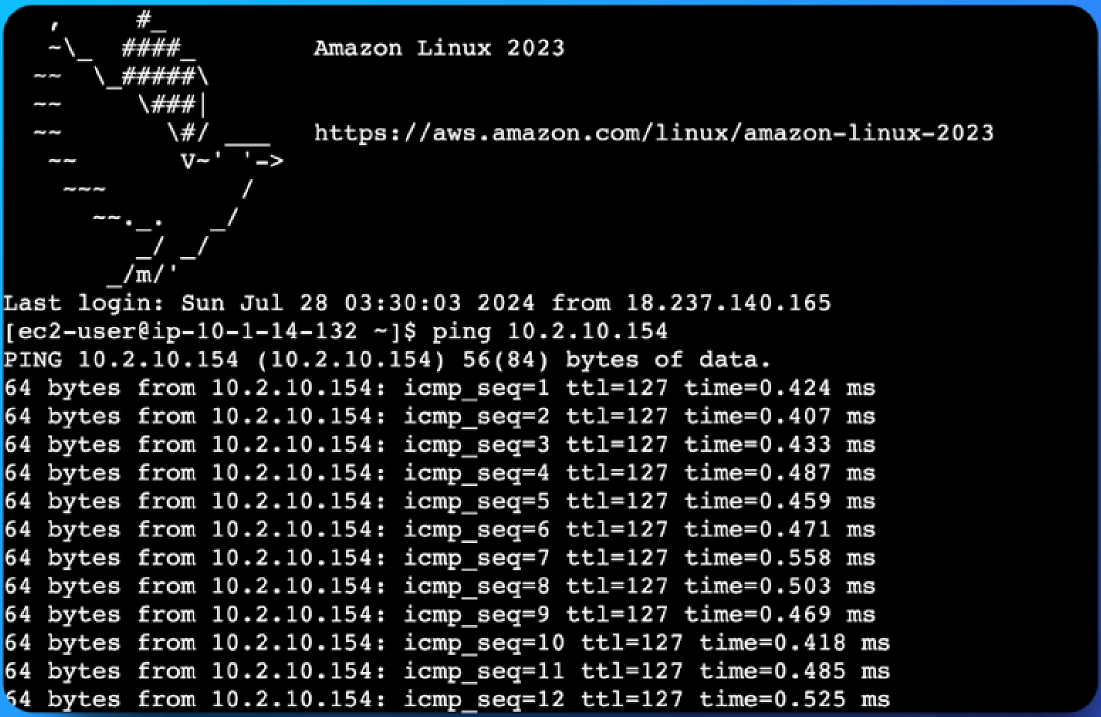
To fix this, I had to update my second EC2 instance's security group because it was not letting in ICMP traffic, which is a traffic type of a ping message. I added a new rule that allows ICMP traffic coming in from any resource in VPC 2.



Inbound rules (2)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
–	sgr-035c5ec63dd40535f	IPv4	All ICMP - ...	ICMP	All	10.1.0.0/16
–	sgr-08453f03001f8fc37	–	All traffic	All	All	sg-0924847dc6d2ca85

Running the ping command once again gave me my desired output!



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun Jul 28 03:30:03 2024 from 18.237.140.165
[ec2-user@ip-10-1-14-132 ~]$ ping 10.2.10.154
PING 10.2.10.154 (10.2.10.154) 56(84) bytes of data.
64 bytes from 10.2.10.154: icmp_seq=1 ttl=127 time=0.424 ms
64 bytes from 10.2.10.154: icmp_seq=2 ttl=127 time=0.407 ms
64 bytes from 10.2.10.154: icmp_seq=3 ttl=127 time=0.433 ms
64 bytes from 10.2.10.154: icmp_seq=4 ttl=127 time=0.487 ms
64 bytes from 10.2.10.154: icmp_seq=5 ttl=127 time=0.459 ms
64 bytes from 10.2.10.154: icmp_seq=6 ttl=127 time=0.471 ms
64 bytes from 10.2.10.154: icmp_seq=7 ttl=127 time=0.558 ms
64 bytes from 10.2.10.154: icmp_seq=8 ttl=127 time=0.503 ms
64 bytes from 10.2.10.154: icmp_seq=9 ttl=127 time=0.469 ms
64 bytes from 10.2.10.154: icmp_seq=10 ttl=127 time=0.418 ms
64 bytes from 10.2.10.154: icmp_seq=11 ttl=127 time=0.485 ms
64 bytes from 10.2.10.154: icmp_seq=12 ttl=127 time=0.525 ms
```