



VPC Peering

G

Gaurav Balpande

A VPC peering connection pcx-012008aa668d40111 / VPC 1 <=> VPC 2 has been requested.

pcx-012008aa668d40111 / VPC 1 <=> VPC 2

Pending acceptance

You can accept or reject this peering connection request using the 'Actions' menu. You have until Friday, August 15, 2025 at 22:36:04 GMT+5:30 to accept or reject the request, otherwise it expires.

Details Info

Requester owner ID	Acceptor owner ID
590183705172	590183705172
Peering connection ID	Requester VPC
pcx-012008aa668d40111	vpc-079f76d6d5a14309d / dray-project-vpc
Status	Requester CIDRs
Pending Acceptance by 590183705172	10.1.0.0/16
Expiration time	Requester Region
Friday, August 15, 2025 at 22:36:04 GMT+5:30	Mumbai (ap-south-1)
	Acceptor Peering connection ARN
	arn:aws:ec2:ap-south-1:590183705172:vpc-peering-connection/pcx-012008aa668d40111
	Acceptor VPC
	vpc-0f9cc8fed55140bd / dray2-project-vpc
	Acceptor CIDRs
	-
	Acceptor Region
	Mumbai (ap-south-1)

Actions ▾

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is the fundamental networking tool which is used for creation of private space within the AWS region. It is needed for security and to easily manage the resources.

How I used Amazon VPC in this project

in today's project i had test the connectivity between the instance in two vpc by setting VPC peering and also learn about the concept of peering

One thing I didn't expect in this project was...

One thing I never expected is that we can peer with the instance in other aws account.

This project took me...

it took near to 2.00 hr and it slightly hard.

In the first part of my project...

Step 1 - Set up my VPC

In this step we are going to create two VPC with the help of Amazon VPC wizard.

Step 2 - Create a Peering Connection

In this step we are going to create VPC peering by using VPC peering connection so that the two VPC we had created can connect with each other.

Step 3 - Update Route Tables

In this step we are going to set up a connection to connect two VPC so that they can share data.

Step 4 - Launch EC2 Instances

In this step we are going to create EC2 instance in the earlier created two VPC so that we can share data between this two and can check the connection.

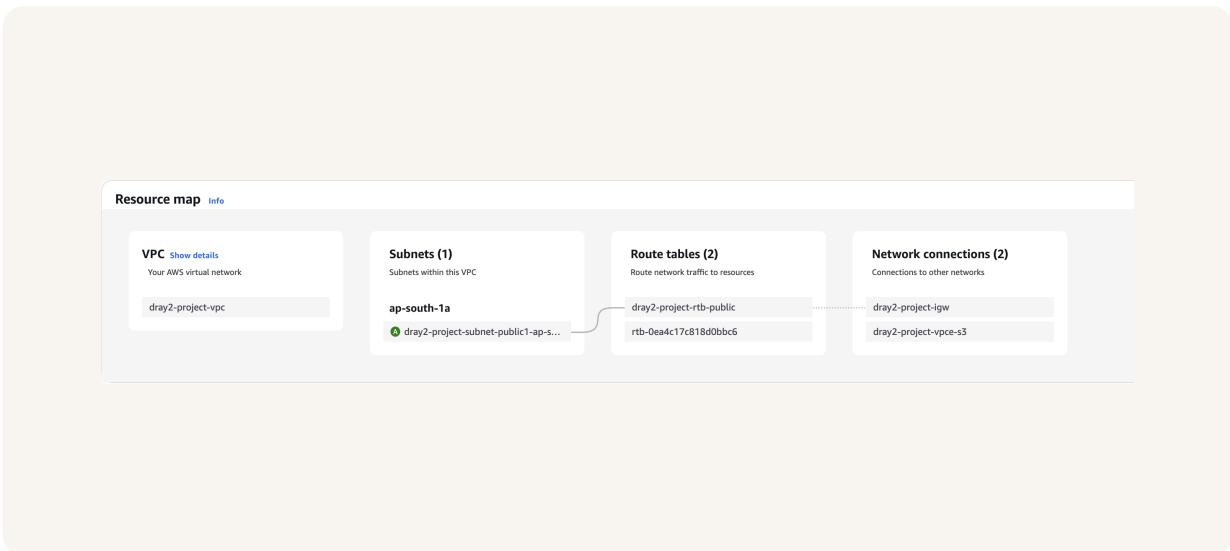
Multi-VPC Architecture

I started my project by launching two VPC and created only one subnet and that too public this is because in this project we just want to try VPC peering

The CIDR blocks for VPCs 1 and 2 are 10.1.0.0/24 and 10.2.0.0/24 They have to be unique because for peering we need to have different address so there will be no confusion.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances as we are going to connect with the help of EC2 instance connect and for that we don't need the key pair.

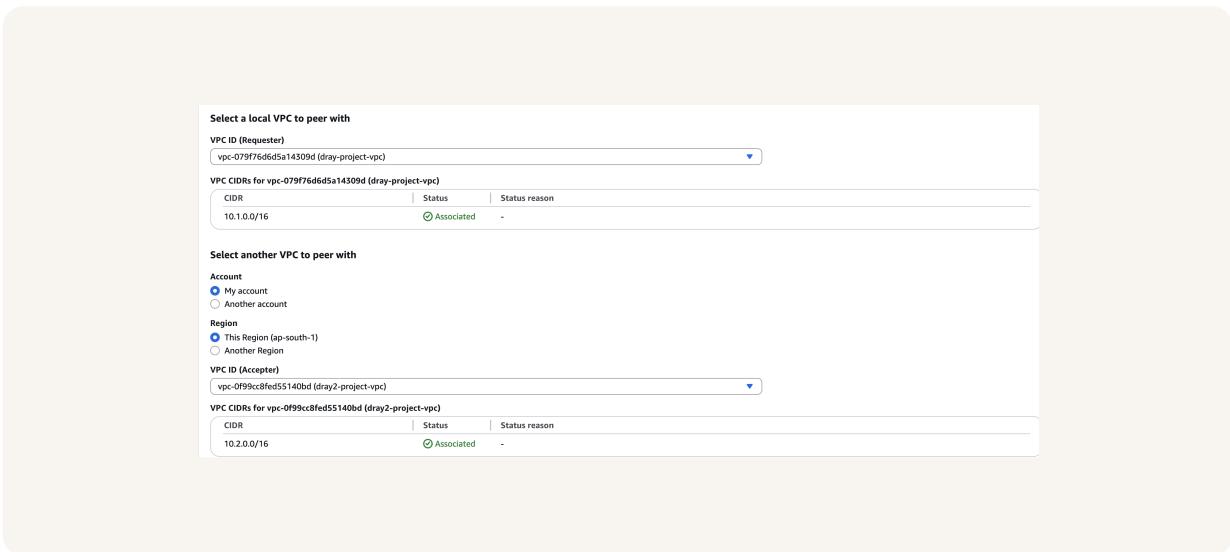


VPC Peering

A VPC peering connection is the tool in AWS for peering two resources which are in different VPC and that too without exposing to public internet.

VPCs would use peering connections to connect two resources in different VPC and that too without exposing to internet

The difference between a Requester and an Acceptor in a peering connection is the requester made the request and the accepter accept or deny the request.



Updating route tables

After accepting a peering connection, my VPCs' route tables need to be updated because the VPC will not know the exact location of the VPC peering connection and that path is provided by route table.

My VPCs' new routes have a destination of 10.2.0.0/16 and other has 10.1.0.0/16 The routes' target was pcx-012008aa668d40111.

The screenshot shows the AWS Route Tables interface for a specific VPC. A green notification bar at the top indicates that routes have been updated successfully. The main table lists two route tables: 'dray-project-rtb-public' and 'dray2-project-rtb-public'. The second route table is selected. Below the table, the 'Routes' tab is active, displaying three new routes:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-017b2a6d2ba0d46be	Active	No
10.1.0.0/16	pcx-012008aa668d40111	Active	No
10.2.0.0/16	local	Active	No



In the second part of my project...

Step 5 - Use EC2 Instance Connect

In this step we are actually going to check the connection by launching one of the instance and will try to share data to other instance to check the connection.

Step 6 - Connect to EC2 Instance 1

In this step we are again going to connect to our EC2 instance as our earlier try is failed.

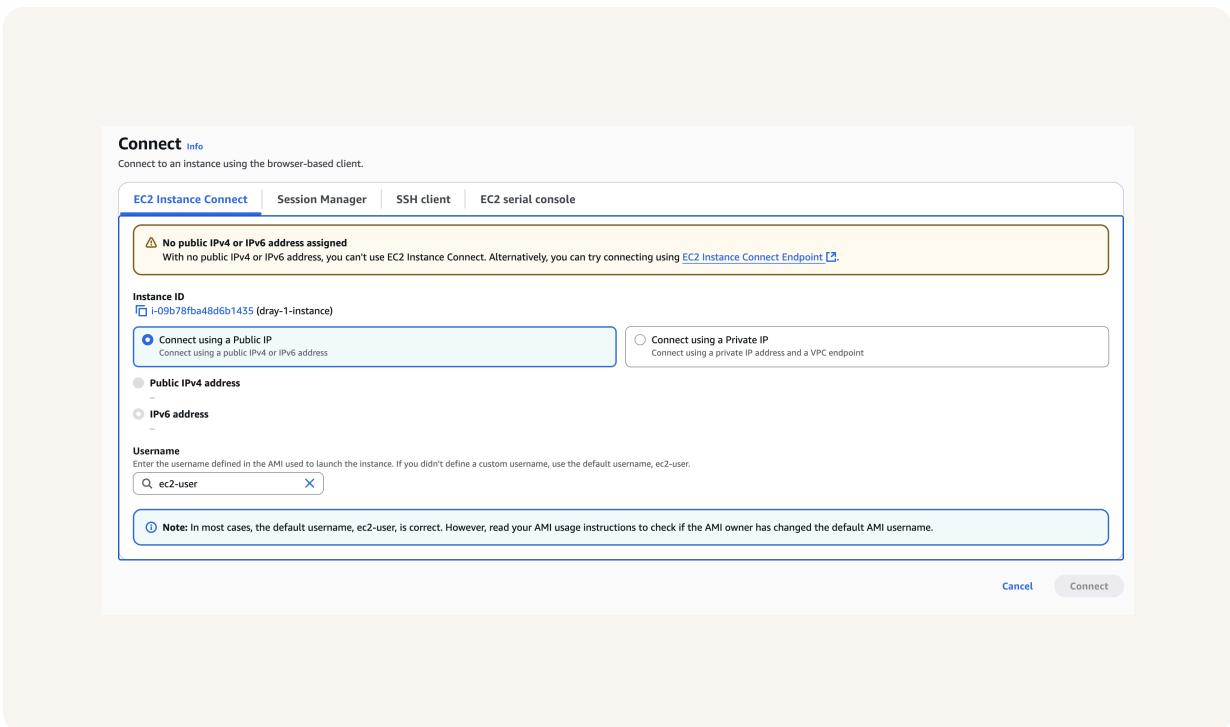
Step 7 - Test VPC Peering

In this step we will send message from one instance to other to check the connection and will try to solve various issues.

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to connect to one of our instance so that we can send message to check the connection.

I was stopped from using EC2 Instance Connect as a warning come telling that there is no public ip associated with the instance.





Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. Elastic IP addresses are static Ip which can reallocate but it is unique and can be used when we want the website to host statistically.

Associating an Elastic IP address resolved the error because it assign a public ip for the instance.

The screenshot shows the 'Allocate Elastic IP address' dialog box on the AWS EC2 console. The 'Elastic IP address settings' section is selected, showing the 'Amazon's pool of IPv4 addresses' option is chosen. A 'Network border group' dropdown is set to 'ap-south-1'. Below this, there's a note about AWS Global Accelerator and a 'Create accelerator' button. The 'Tags - optional' section indicates no tags are associated with the resource, with a 'Add new tag' button available. At the bottom right are 'Cancel' and 'Allocate' buttons. The footer includes links for CloudShell, Feedback, and legal notices.

Troubleshooting ping issues

To test VPC peering, I ran the command ping 10.2.4.85

A successful ping test would validate my VPC peering connection because it will send a continuous messages of time and other.

I had to update my second EC2 instance's security group because by default the ICMP messages are blocked I added a new rule that allow the ICMP V4 traffic.

```
[ec2-user@ip-10-1-0-70 ~]$ ping 10.2.4.85
PING 10.2.4.85 (10.2.4.85) 56(84) bytes of data.
64 bytes from 10.2.4.85: icmp_seq=154 ttl=127 time=0.188 ms
64 bytes from 10.2.4.85: icmp_seq=155 ttl=127 time=0.189 ms
64 bytes from 10.2.4.85: icmp_seq=156 ttl=127 time=0.180 ms
64 bytes from 10.2.4.85: icmp_seq=157 ttl=127 time=0.181 ms
64 bytes from 10.2.4.85: icmp_seq=158 ttl=127 time=0.196 ms
64 bytes from 10.2.4.85: icmp_seq=159 ttl=127 time=0.190 ms
64 bytes from 10.2.4.85: icmp_seq=160 ttl=127 time=0.192 ms
64 bytes from 10.2.4.85: icmp_seq=161 ttl=127 time=0.200 ms
64 bytes from 10.2.4.85: icmp_seq=162 ttl=127 time=0.188 ms
64 bytes from 10.2.4.85: icmp_seq=163 ttl=127 time=0.189 ms
64 bytes from 10.2.4.85: icmp_seq=164 ttl=127 time=0.161 ms
64 bytes from 10.2.4.85: icmp_seq=165 ttl=127 time=0.181 ms
64 bytes from 10.2.4.85: icmp_seq=166 ttl=127 time=0.187 ms
64 bytes from 10.2.4.85: icmp_seq=167 ttl=127 time=0.192 ms
64 bytes from 10.2.4.85: icmp_seq=168 ttl=127 time=0.194 ms
64 bytes from 10.2.4.85: icmp_seq=169 ttl=127 time=0.195 ms
64 bytes from 10.2.4.85: icmp_seq=170 ttl=127 time=0.165 ms
64 bytes from 10.2.4.85: icmp_seq=171 ttl=127 time=0.162 ms
64 bytes from 10.2.4.85: icmp_seq=172 ttl=127 time=0.176 ms
64 bytes from 10.2.4.85: icmp_seq=173 ttl=127 time=0.170 ms
64 bytes from 10.2.4.85: icmp_seq=174 ttl=127 time=0.173 ms
64 bytes from 10.2.4.85: icmp_seq=175 ttl=127 time=0.203 ms
64 bytes from 10.2.4.85: icmp_seq=176 ttl=127 time=0.188 ms
64 bytes from 10.2.4.85: icmp_seq=177 ttl=127 time=0.189 ms
64 bytes from 10.2.4.85: icmp_seq=178 ttl=127 time=0.186 ms
64 bytes from 10.2.4.85: icmp_seq=179 ttl=127 time=0.197 ms
64 bytes from 10.2.4.85: icmp_seq=180 ttl=127 time=0.197 ms
64 bytes from 10.2.4.85: icmp_seq=181 ttl=127 time=0.197 ms
64 bytes from 10.2.4.85: icmp_seq=182 ttl=127 time=0.192 ms
64 bytes from 10.2.4.85: icmp_seq=183 ttl=127 time=0.175 ms
64 bytes from 10.2.4.85: icmp_seq=184 ttl=127 time=0.188 ms
64 bytes from 10.2.4.85: icmp_seq=185 ttl=127 time=0.188 ms
64 bytes from 10.2.4.85: icmp_seq=186 ttl=127 time=0.179 ms
```



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

