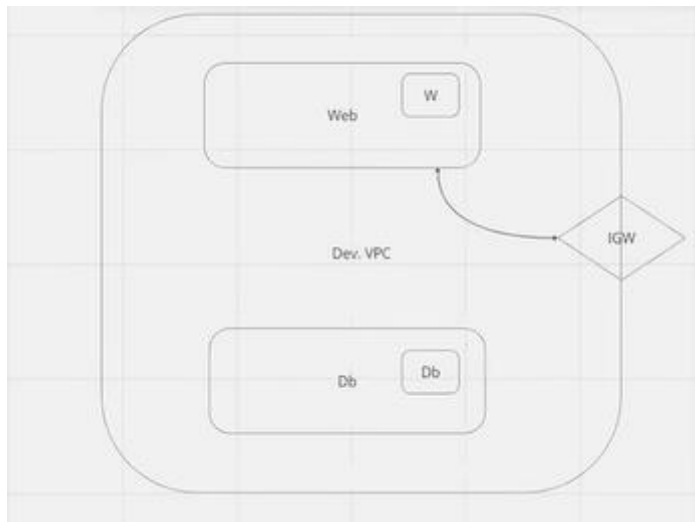


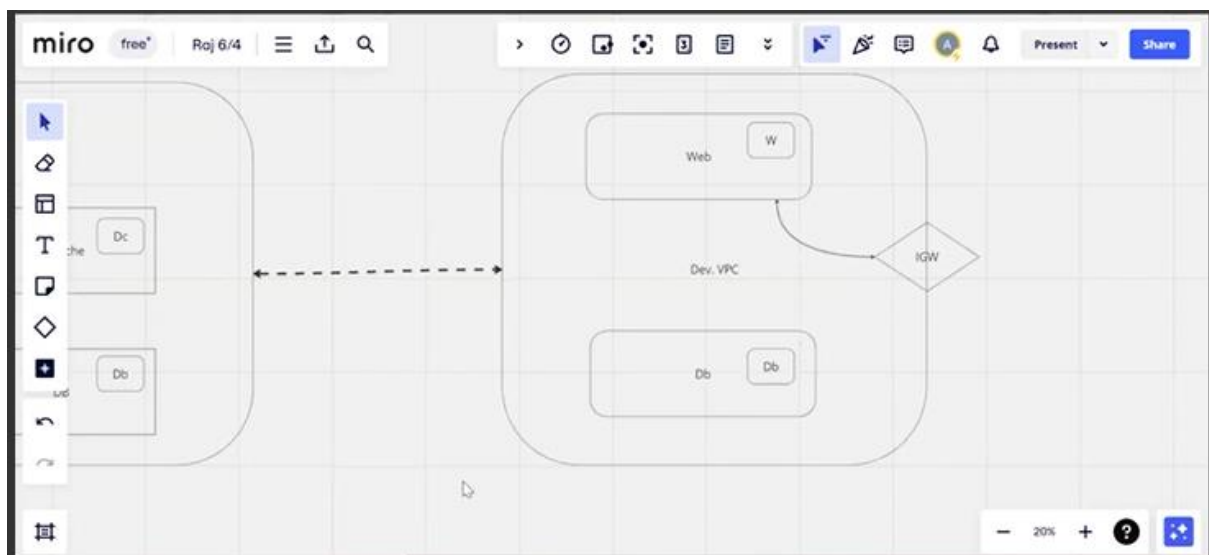
### Development Network:

1. Design and build 2-tier architecture with two subnets named web and db and launch instances in both subnets and name them as per the subnet names.

1. Make sure only the web subnet can send internet requests.



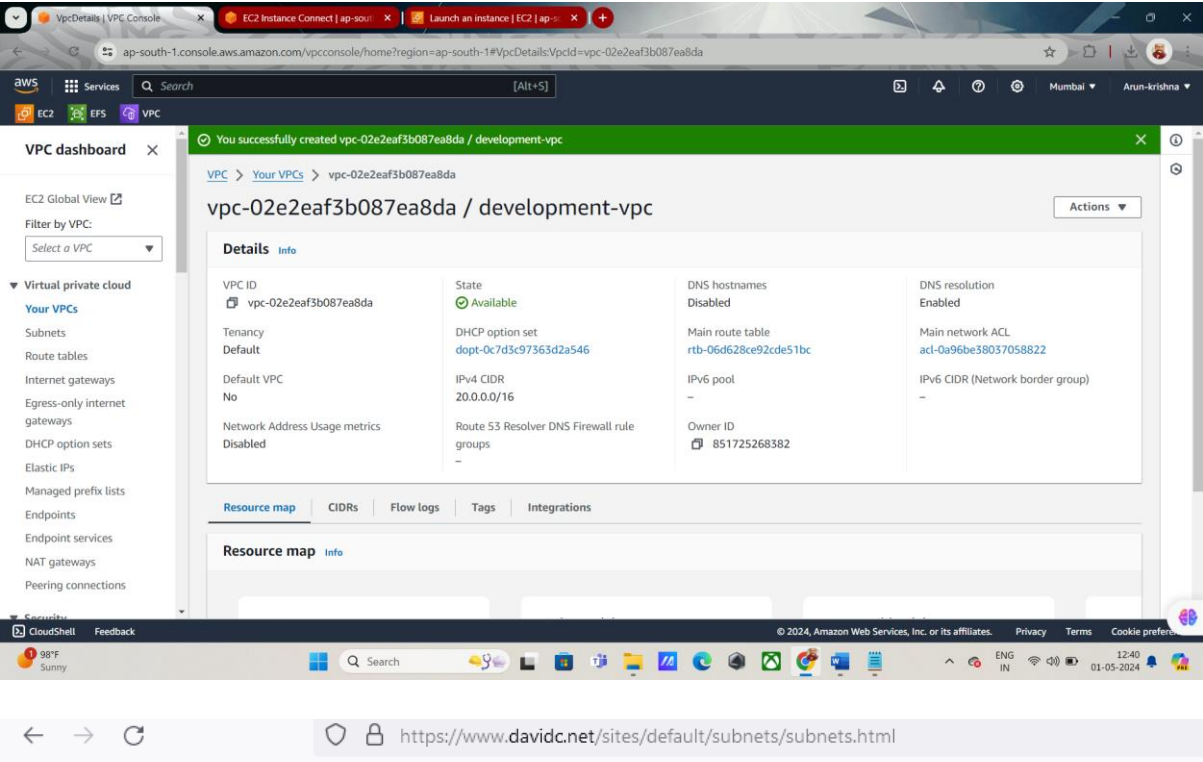
3. Create peering connection between production network and development network.



4. Setup connection between db subnets of both production network and development network respectively.

Pls add the ICMP protocol in both db subnets to get the db connectivity.

Now Go to VPC and Create VPC for the Development Network



Visual Subnet Calculator

Enter the network you wish to subnet:

Network Address: 20.0.0.0 Mask bits: 16 [Update] [Reset]

Show columns: ☒ Subnet address ☐ Netmask ☒ Range of addresses ☒ Useable IPs ☒ Hosts ☒ Divide ☒ Join

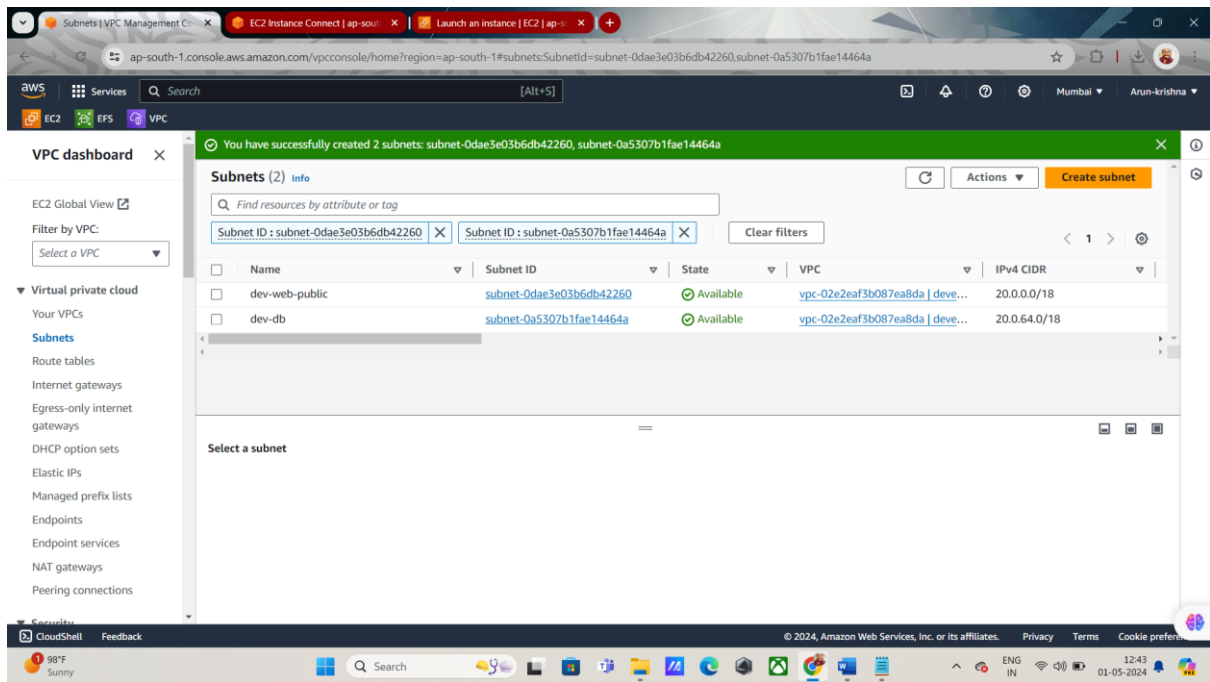
Click below to split and join subnets.

If you wish to save this subnetting for later, bookmark [this hyperlink](https://www.davidc.net/sites/default/subnets/subnets.html).

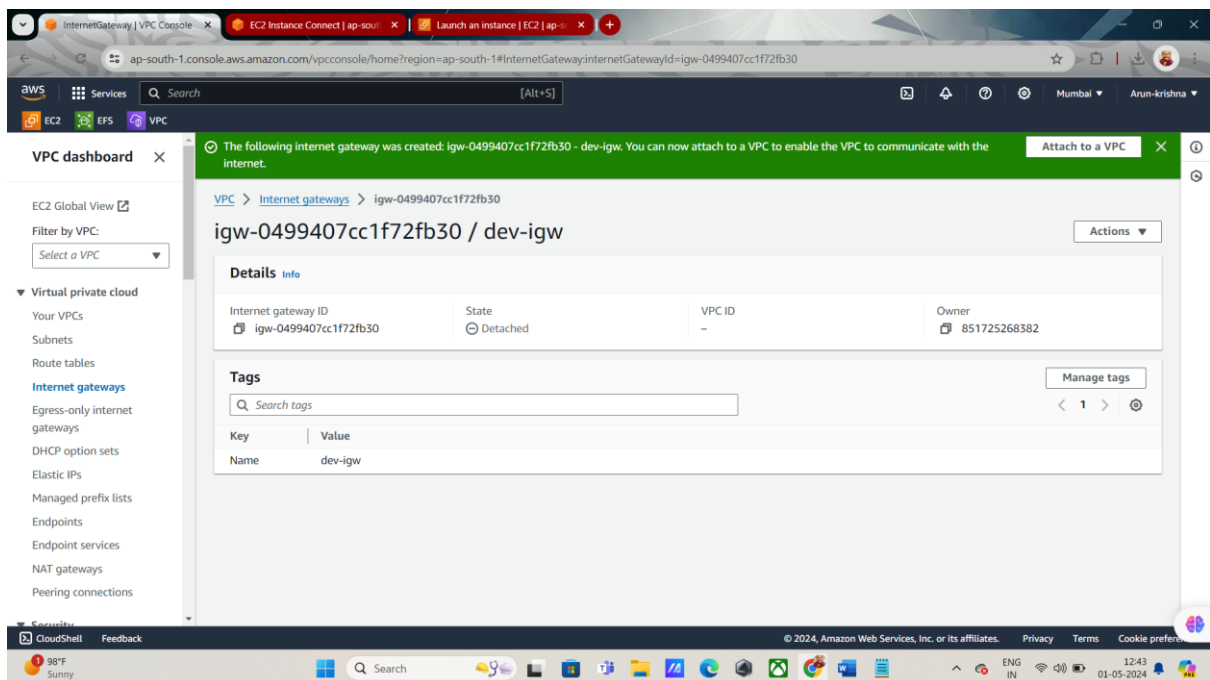
Subnet address	Range of addresses	Useable IPs	Hosts	Divide	Join
20.0.0.0/18	20.0.0.0 - 20.0.63.255	20.0.0.1 - 20.0.63.254	16382	<a href="#">Divide</a>	<a href="#">Join</a>
20.0.64.0/18	20.0.64.0 - 20.0.127.255	20.0.64.1 - 20.0.127.254	16382	<a href="#">Divide</a>	<a href="#">Join</a>
20.0.128.0/18	20.0.128.0 - 20.0.191.255	20.0.128.1 - 20.0.191.254	16382	<a href="#">Divide</a>	<a href="#">Join</a>
20.0.192.0/18	20.0.192.0 - 20.0.255.255	20.0.192.1 - 20.0.255.254	16382	<a href="#">Divide</a>	<a href="#">Join</a>

User different CIDR for the peering connection to avoid overlapping

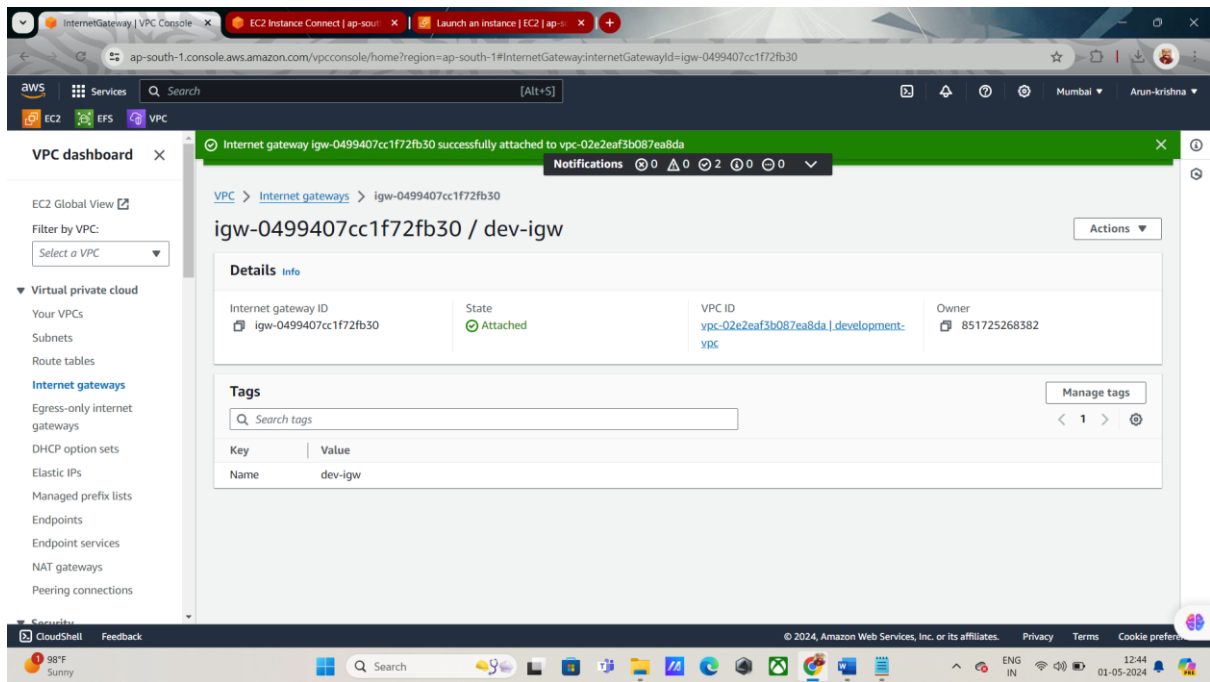
Now go to subnet and create web public subnet and db private subnet



Now create the Internet gateway



Now attach it to DEV VPC



Now we need to create the two Route table one for web and one for db private subnet

Web RT and Db Rt

<input type="checkbox"/>	-	<a href="#">rtb-06d628ce92cde51bc</a>	-	-	Yes	<a href="#">vpc-02e</a>
<input type="checkbox"/>	Web-dev-prt	<a href="#">rtb-061886204cbb974d6</a>	-	-	No	<a href="#">vpc-02e</a>
<input type="checkbox"/>	DB-Dev	<a href="#">rtb-0a1f0c35951696ae8</a>	-	-	No	<a href="#">vpc-02e</a>

at

ts

**rtb-0162122a9bde15c9d / Nat-rt**

<a href="#">Details</a>	<a href="#">Routes</a>	<a href="#">Subnet associations</a>	<a href="#">Edge associations</a>	<a href="#">Route propagation</a>	<a href="#">Tags</a>
-------------------------	------------------------	-------------------------------------	-----------------------------------	-----------------------------------	----------------------

Now go to web RT and Db Rt add the assciaotion

Now go to edit route

## Save changes for the internet connection

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditRoutes:routeTableId=rtb-0a1f0c35951696ae8

### Edit routes

Destination	Target	Status	Propagated
20.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No
	igw-0499407cc1f72fb30		

Buttons: Add route, Cancel, Preview, Save changes

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#RouteTableDetails:routeTableId=rtb-061886204cbb974d6

### rtb-061886204cbb974d6 / Web-dev-prt

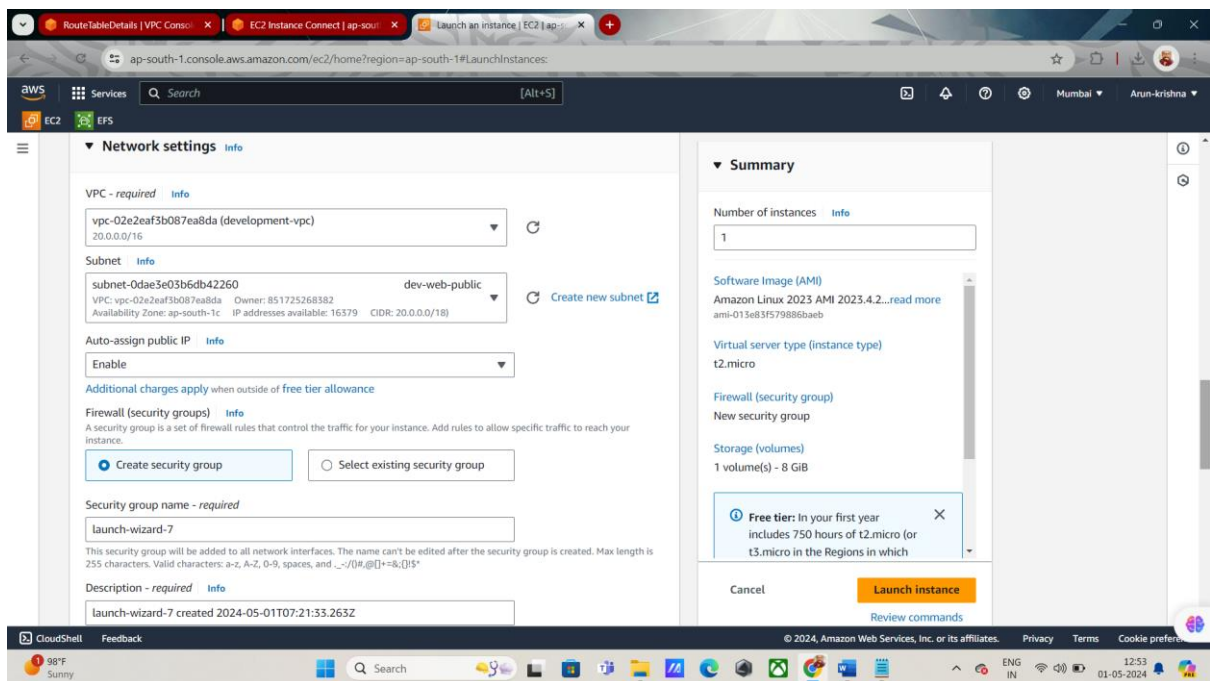
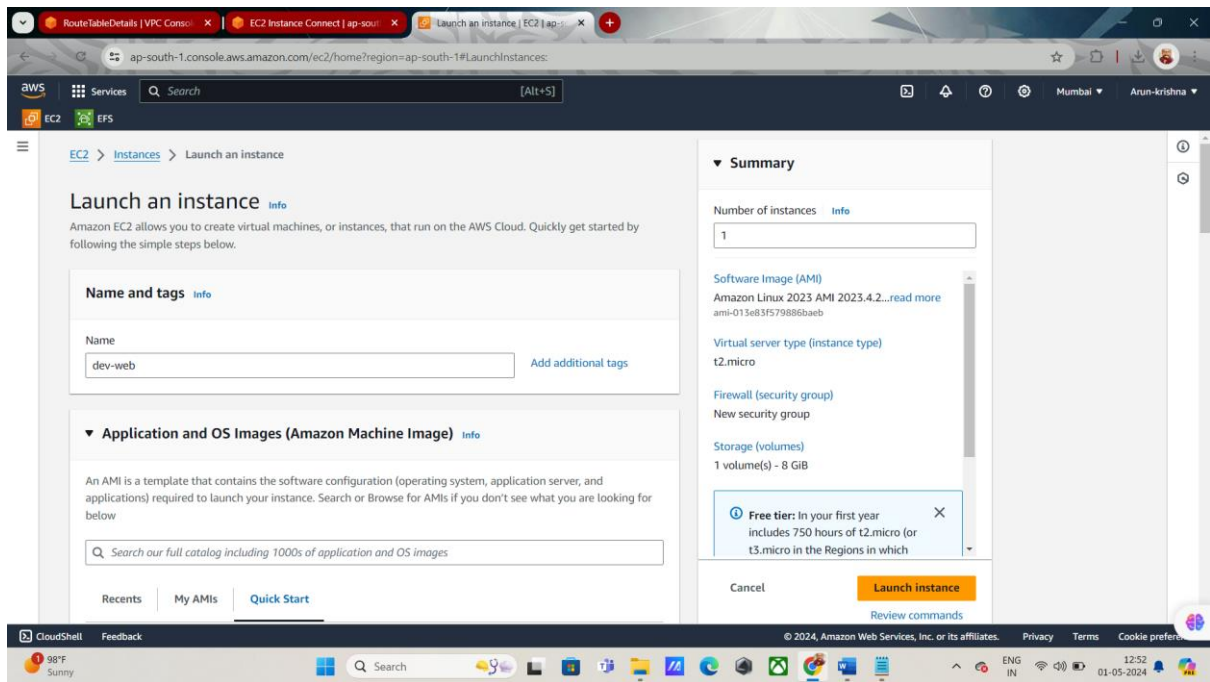
Updated routes for rtb-061886204cbb974d6 / Web-dev-prt successfully

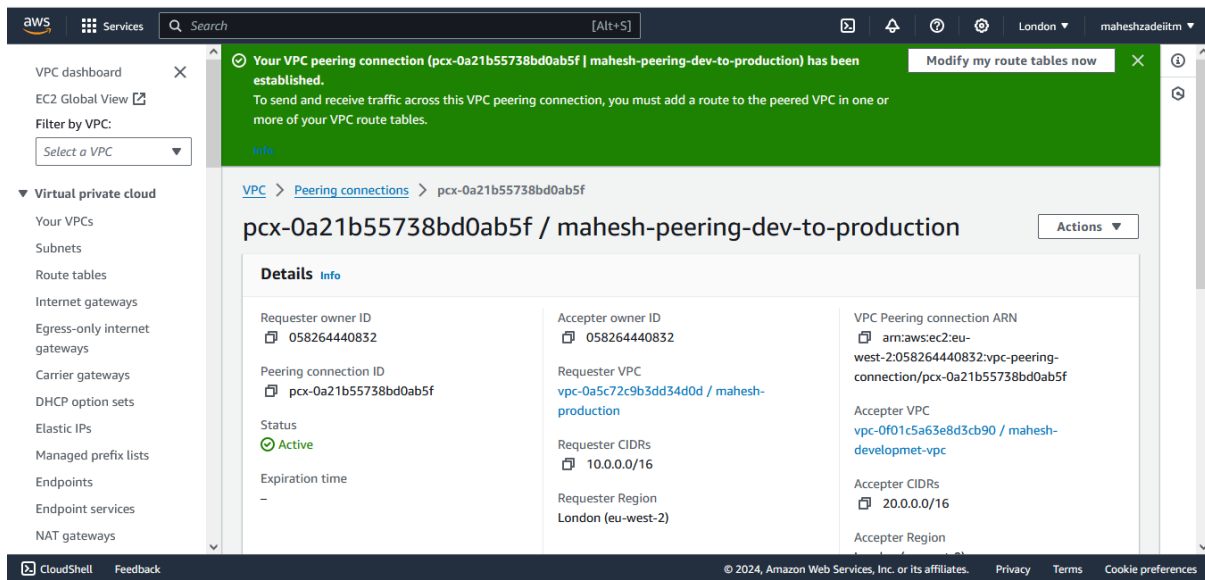
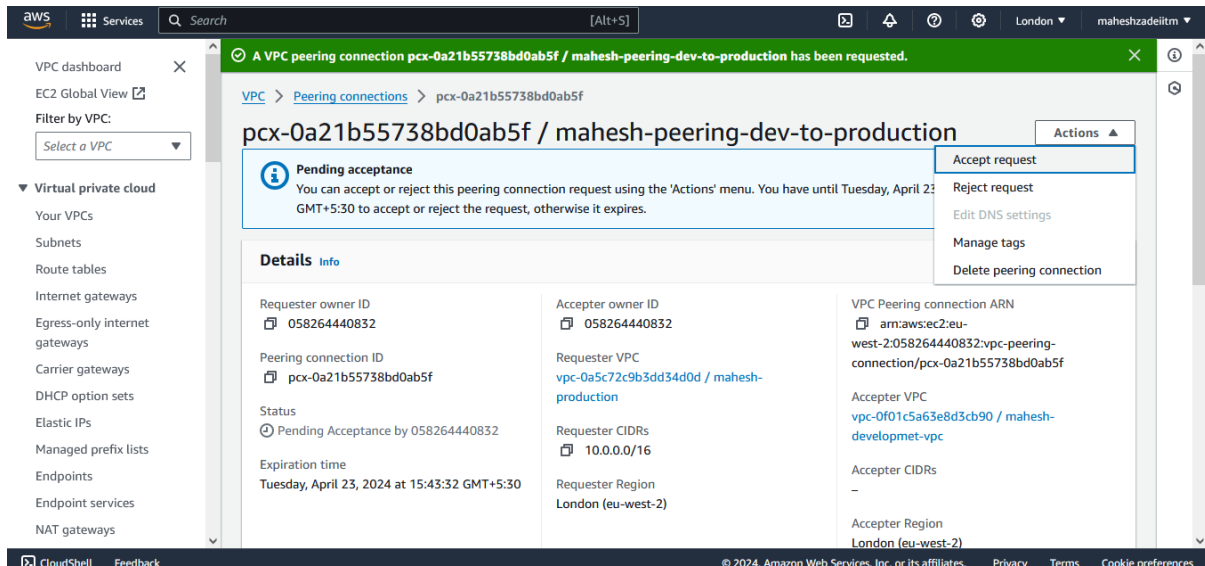
Destination	Target	Status	Propagated
0.0.0.0/0	igw-0499407cc1f72fb30	Active	No
20.0.0.0/16	local	Active	No

Buttons: Both, Edit routes

Now Launch the two EC2 Instance in web and db subnet of Dev VPC

Launch the web instance





Your VPC peering connection (pcx-0a21b55738bd0ab5f | mahesh-peering-dev-to-production) 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.

Now go to private RT of production VPC



The screenshot shows the AWS VPC console interface. On the left, there's a navigation menu with options like 'VPC dashboard', 'EC2 Global View', and 'Virtual private cloud'. The main content area displays the details of a specific route table. The route table ID is 'rtb-0c1b481aec671fb0c' and it belongs to the VPC 'vpc-0f01c5a63e8d3cb90 | mahesh-developmet-vpc'. The route table has one route with destination '20.0.0.0/16' and target 'local', which is in an 'Active' status. The console also shows tabs for 'Routes', 'Subnet associations', 'Edge associations', 'Route propagation', and 'Tags'.

Go to dev priate subnet and take the CIDR and go to Route table and add the CIDR and perring the connection only between the two private subnets

This screenshot shows the 'Edit routes' page in the AWS VPC console. It allows adding a new route to the selected route table. The 'Destination' field is set to '10.0.0.0/26'. The 'Target' dropdown menu is open, showing options like 'local' and 'Peering Connection'. The 'Status' is 'Active' and 'Propagated' is 'No'. There is a 'Remove' button next to the 'Peering Connection' option. At the bottom, there are 'Cancel', 'Preview', and 'Save changes' buttons.

The screenshot shows a success message at the top: 'Updated routes for rtb-0c1b481aec671fb0c / mahesh-RT-db-private successfully'. Below the message, the route table details are shown again, confirming the update. The route table ID is 'rtb-0c1b481aec671fb0c' and it belongs to the VPC 'vpc-0f01c5a63e8d3cb90 | mahesh-developmet-vpc'. The console also shows tabs for 'Routes', 'Subnet associations', 'Edge associations', 'Route propagation', and 'Tags'.

Similarly repeat the process for development private RT



The screenshot shows the AWS Management Console interface. On the left is a navigation sidebar with options like 'VPC dashboard', 'EC2 Global View', and 'Virtual private cloud'. The main content area displays the details for a specific route table. The title is 'rtb-0938eccc64f501cae / mahesh-db-private-RT-dev'. Below the title, there's a 'Details' section with a tabbed interface. The 'Details' tab is active, showing information such as 'Route table ID', 'Main' status, 'Explicit subnet associations', 'Edge associations', 'VPC', and 'Owner ID'. Below this, there's a 'Routes' section with a table showing one route. The route has a destination of '10.0.0.0/16', a target of 'local', and a status of 'Active'.

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

## Edit route

## Go to production subnet route table and copy the CIDR

The screenshot shows the AWS Management Console interface displaying a list of subnets. The title is 'Subnets (1/10) Info'. Below the title, there's a search bar and a table of subnets. The table has columns for 'Name', 'Subnet ID', 'State', 'VPC', and 'IPv4 CIDR'. There are five subnets listed, all with a state of 'Available'. The subnet 'mahesh-db-private' has a CIDR of '10.0.224.0/19'. Below the table, there's a section for the selected subnet 'subnet-04609aaf69ab87fbc / mahesh-db-private' with tabs for 'Details', 'Flow logs', 'Route table', 'Network ACL', 'CIDR reservations', 'Sharing', and 'Tags'.

Name	Subnet ID	State	VPC	IPv4 CIDR
mahesh-web-public	subnet-0cbcd501f88849a78	Available	vpc-0a5c72c9b3dd34d0d   mah...	10.0.0.0/18
mahesh-web-public	subnet-0a0be36588305e5a2	Available	vpc-078935ea941dd11c6	172.31.32.0/20
mahesh-db-private	subnet-04609aaf69ab87fbc	Available	vpc-0a5c72c9b3dd34d0d   mah...	10.0.224.0/19
mahesh-dbcashe-private	subnet-0aa904adb5bccd1e5	Available	vpc-0a5c72c9b3dd34d0d   mah...	10.0.192.0/19
mahesh-web-public	subnet-078cf428b0af217e2	Available	vpc-078935ea941dd11c6	172.31.16.0/20

## Copy the CIDR 10.0.224.0/19

## 10.0.0.0/16 prod CIDR

## 20.0.0.0/16 dev CID