Problem Statement:

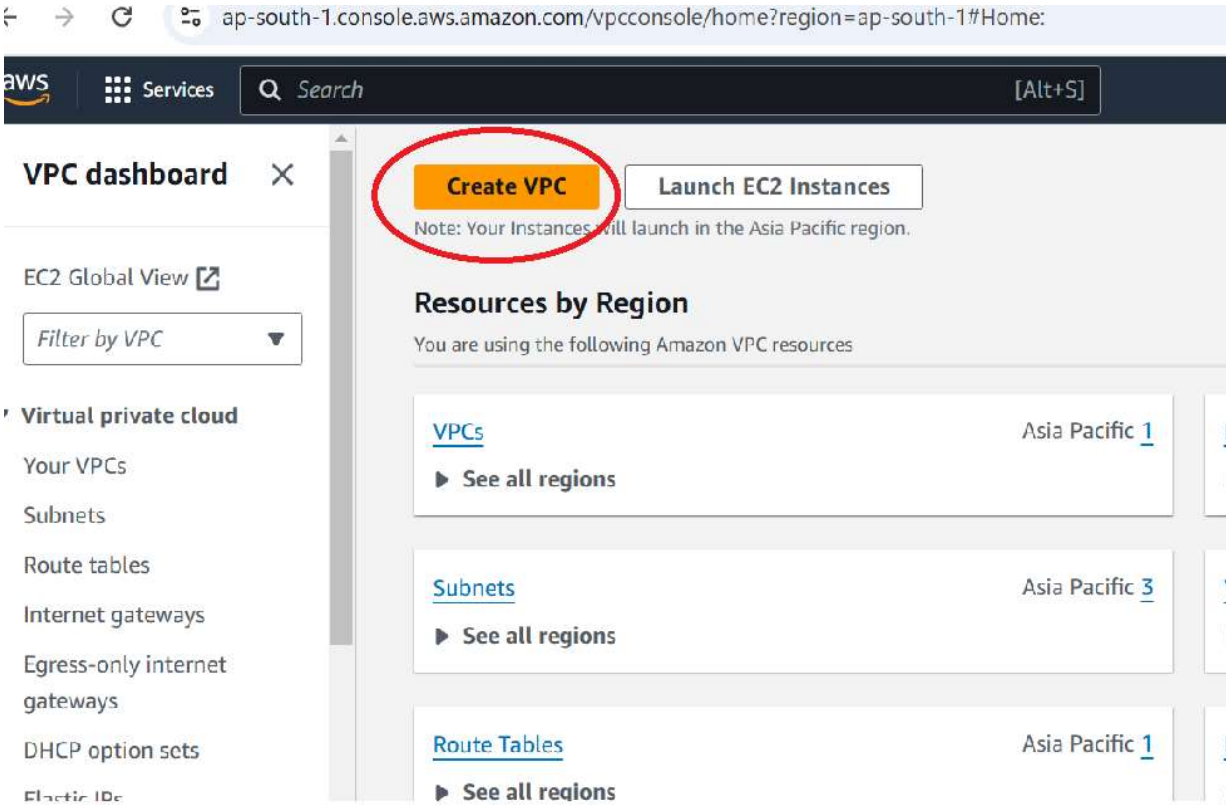
Development Network:

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.
2. 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

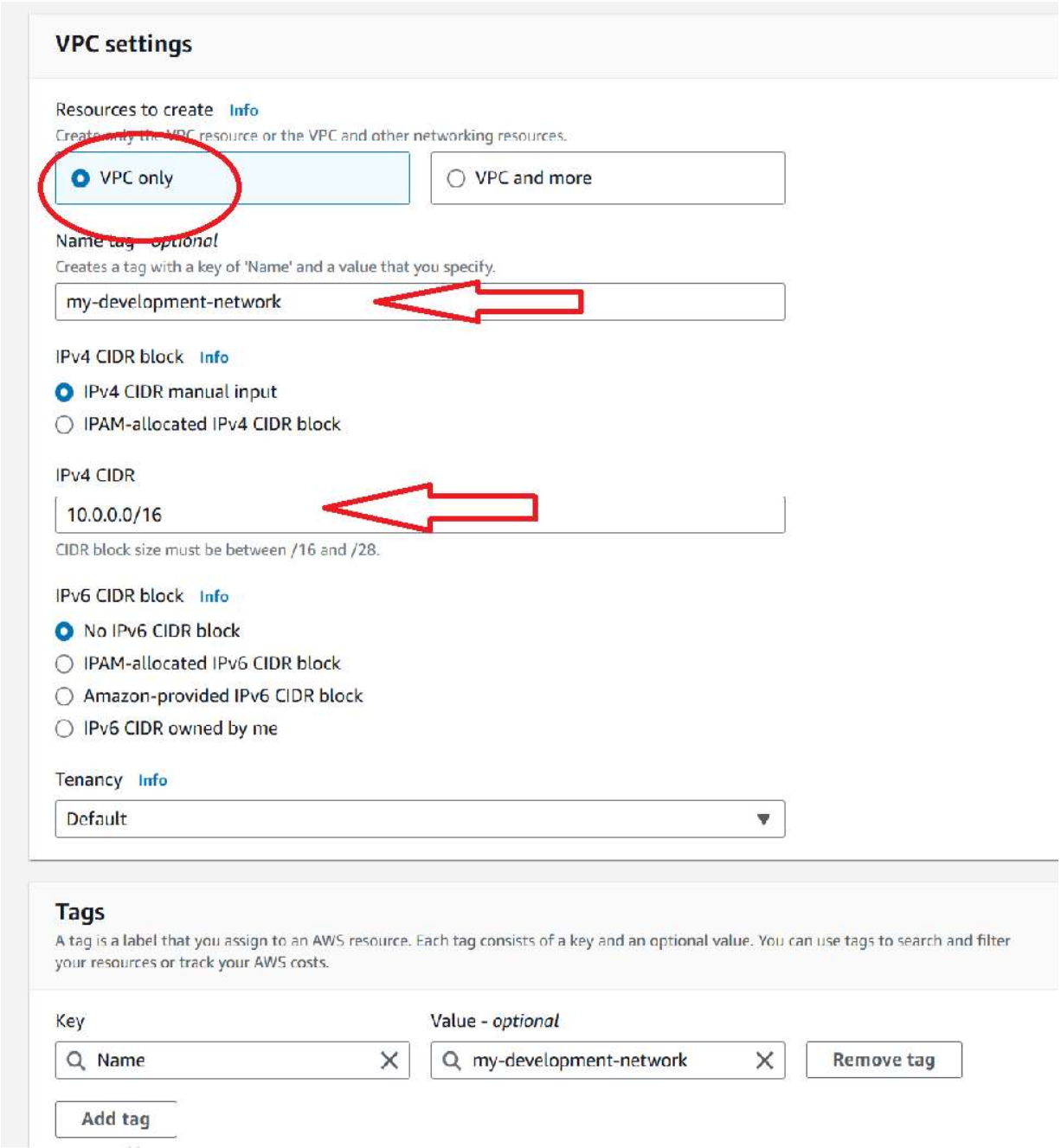
Solution:

Creating Development network.

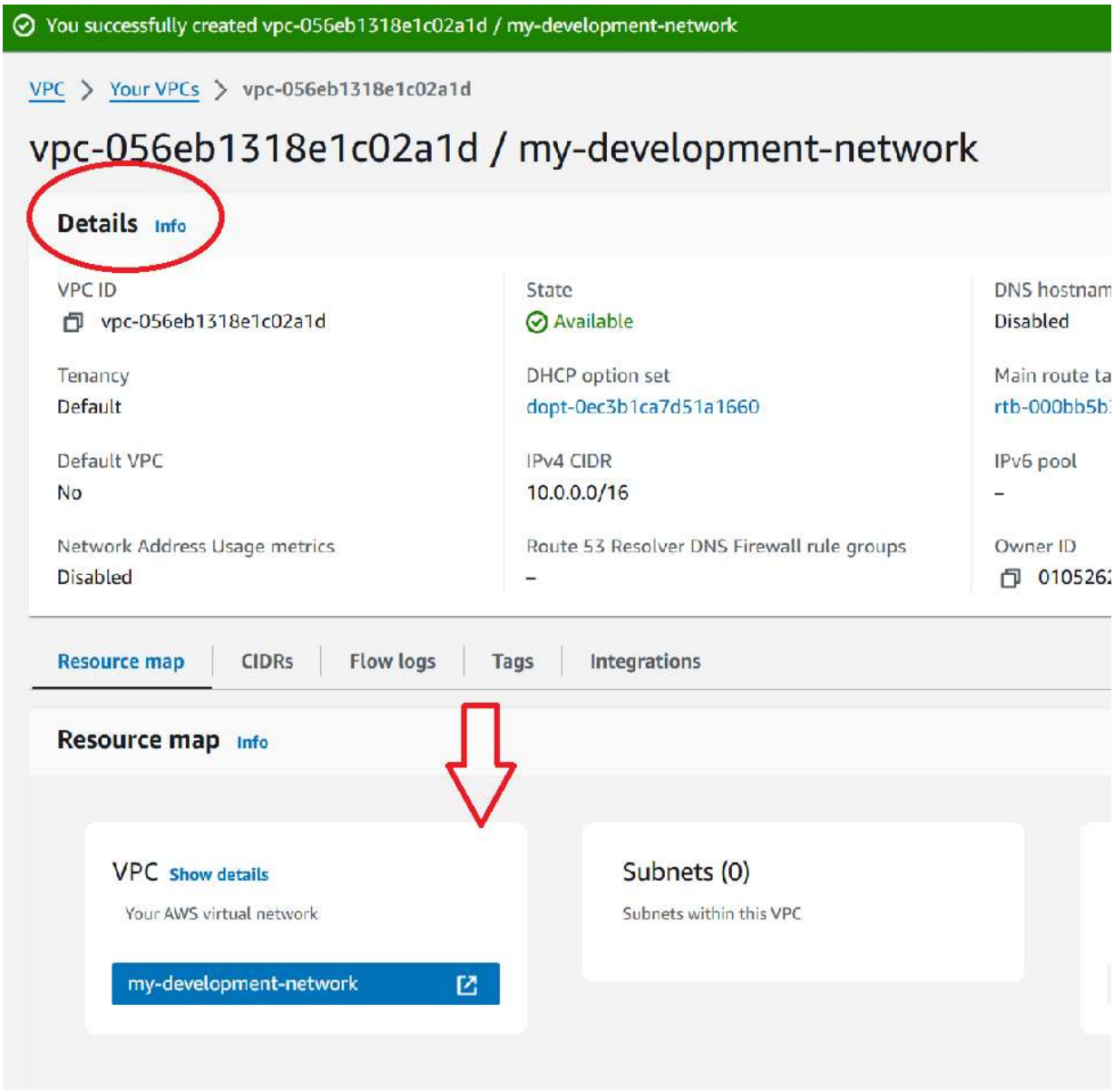
1) Go to the VPC console and create a VPC.



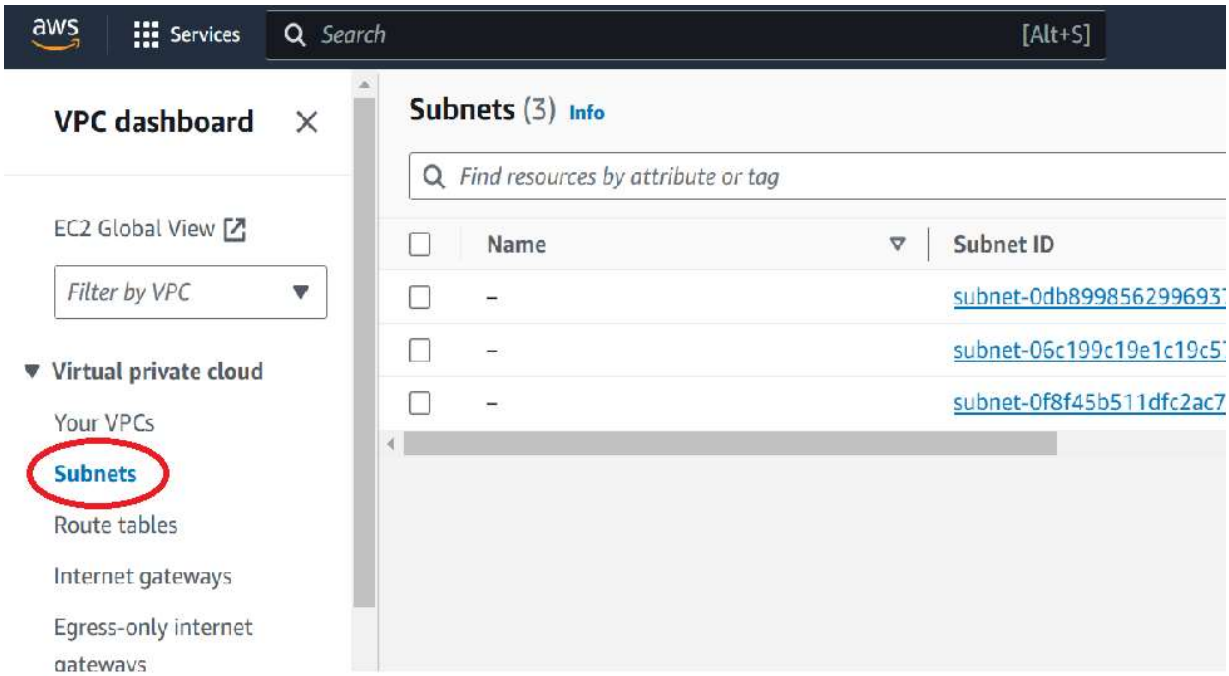
2) Under VPC settings choose VPC only, give a name for the VPC , give the CIDR and click create VPC.



3) So we can see the VPC is created. Under resource map we can see details mapping of our production VPC.



4) Now we have to create two subnets named web and db. <u>Condition: Make sure only the web subnet can send internet requests.</u> that means web subnet is public whereas db subnet is private.
5) Under VPC dashboard click subnets and create subnet.



6) Choose the VPC and give the subnet name my-web-subnet. Scroll down and click on add new subnet.

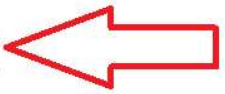
Create subnet Info

VPC

VPC ID

Create subnets in this VPC.

vpc-056eb1318e1c02a1d (my-development-network)



Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

Subnet settings

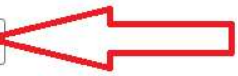
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name

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

my-web-subnet



The name can be up to 256 characters long.

Availability Zone Info

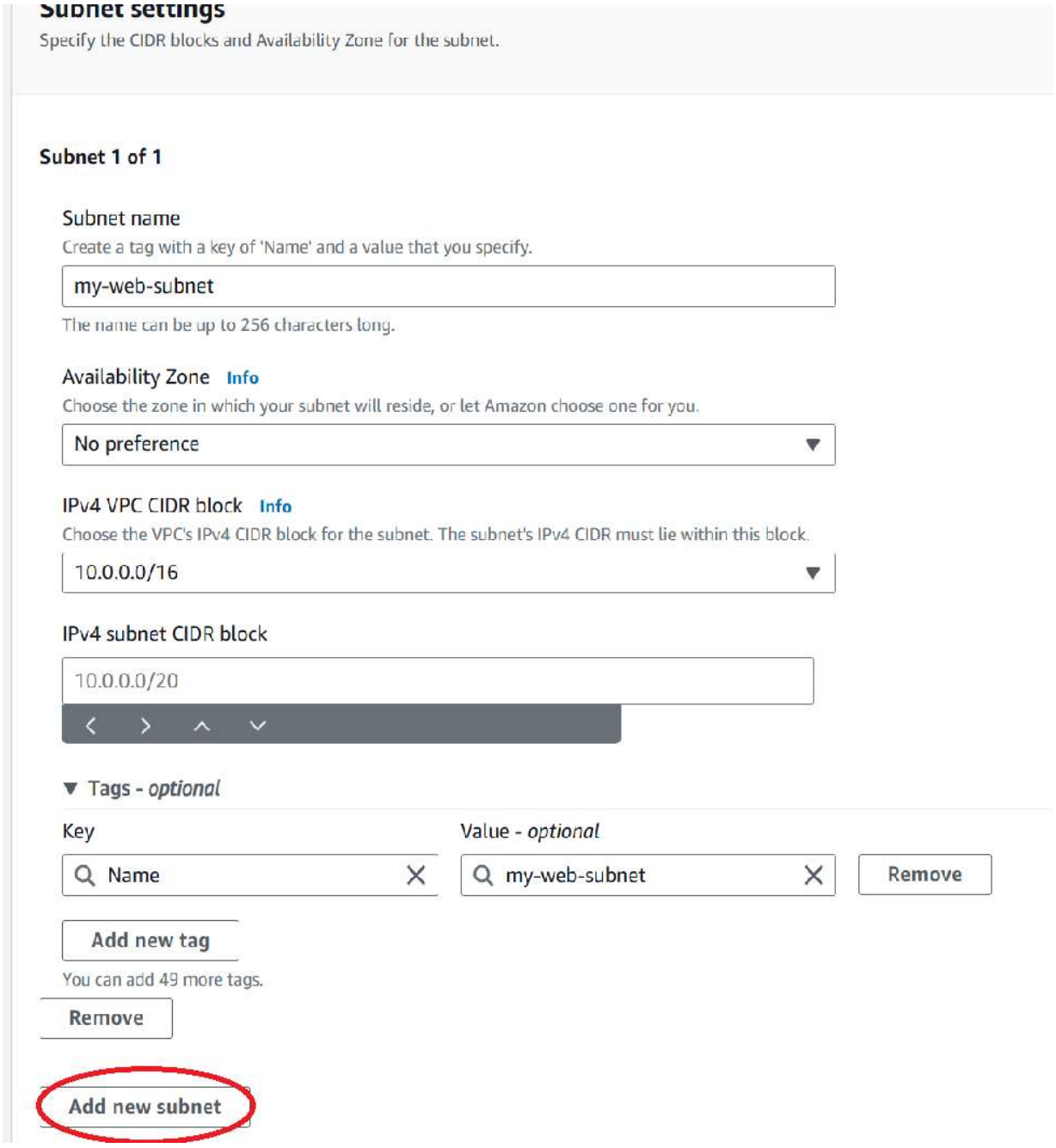
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference

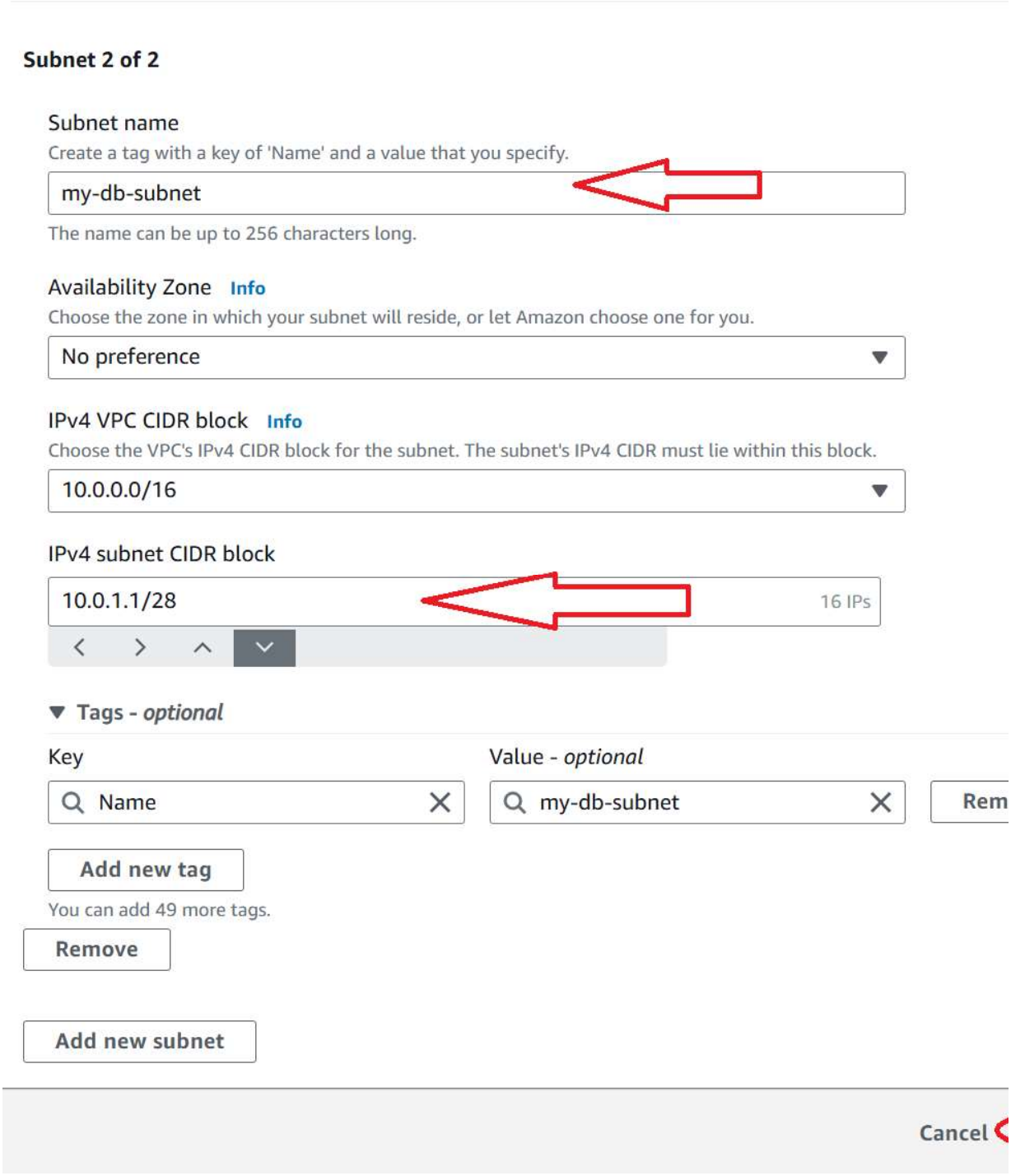
IPv4 VPC CIDR block Info

Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

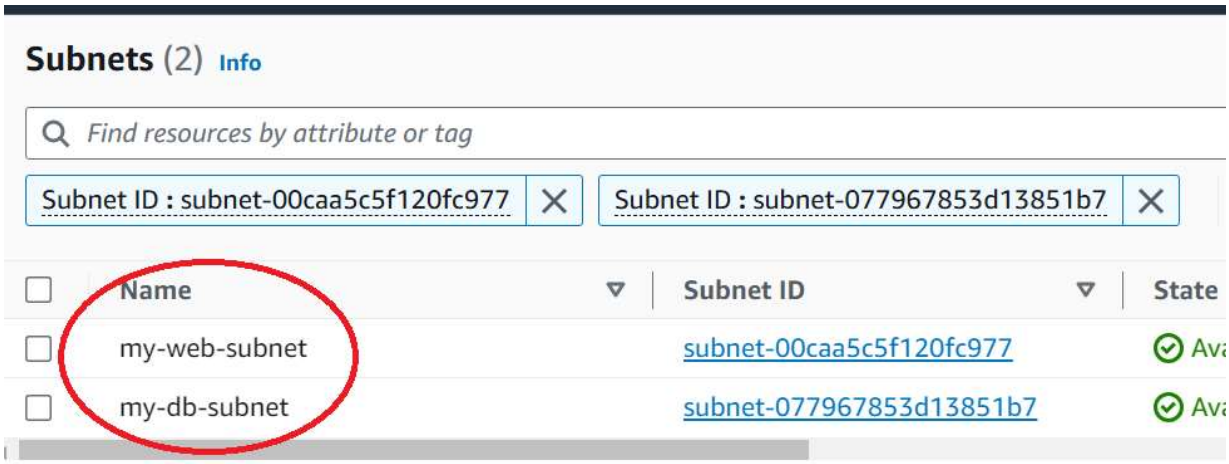
10.0.0.0/16



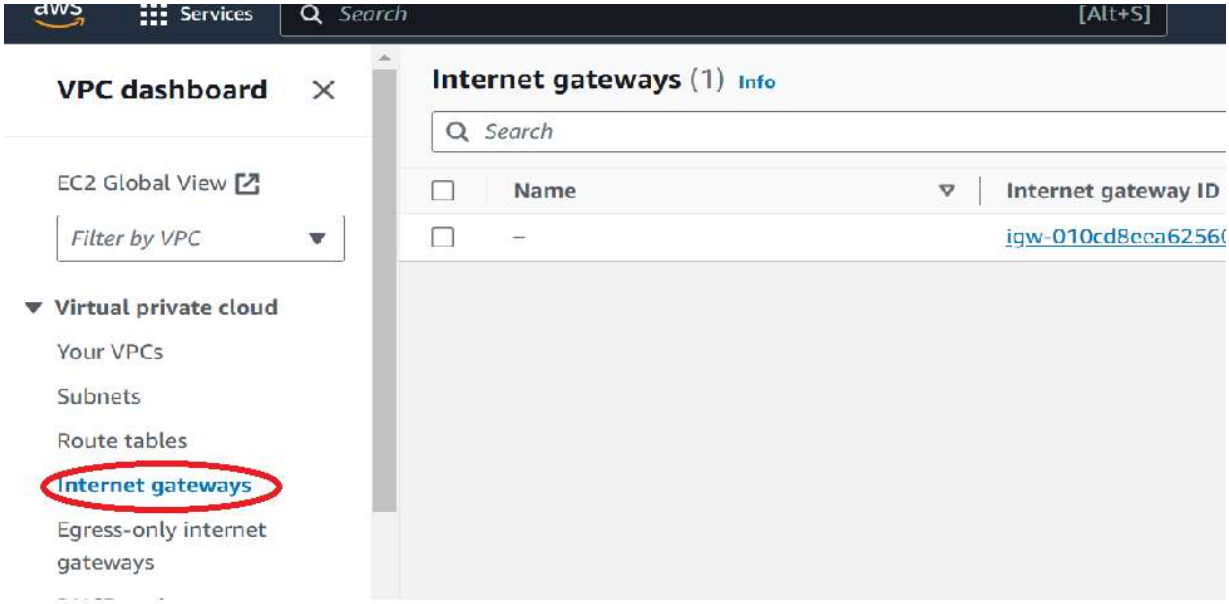
7) Give the subnet name my-db-subnet, give the CIDR value so that it doesn't overlap and click create subnet.



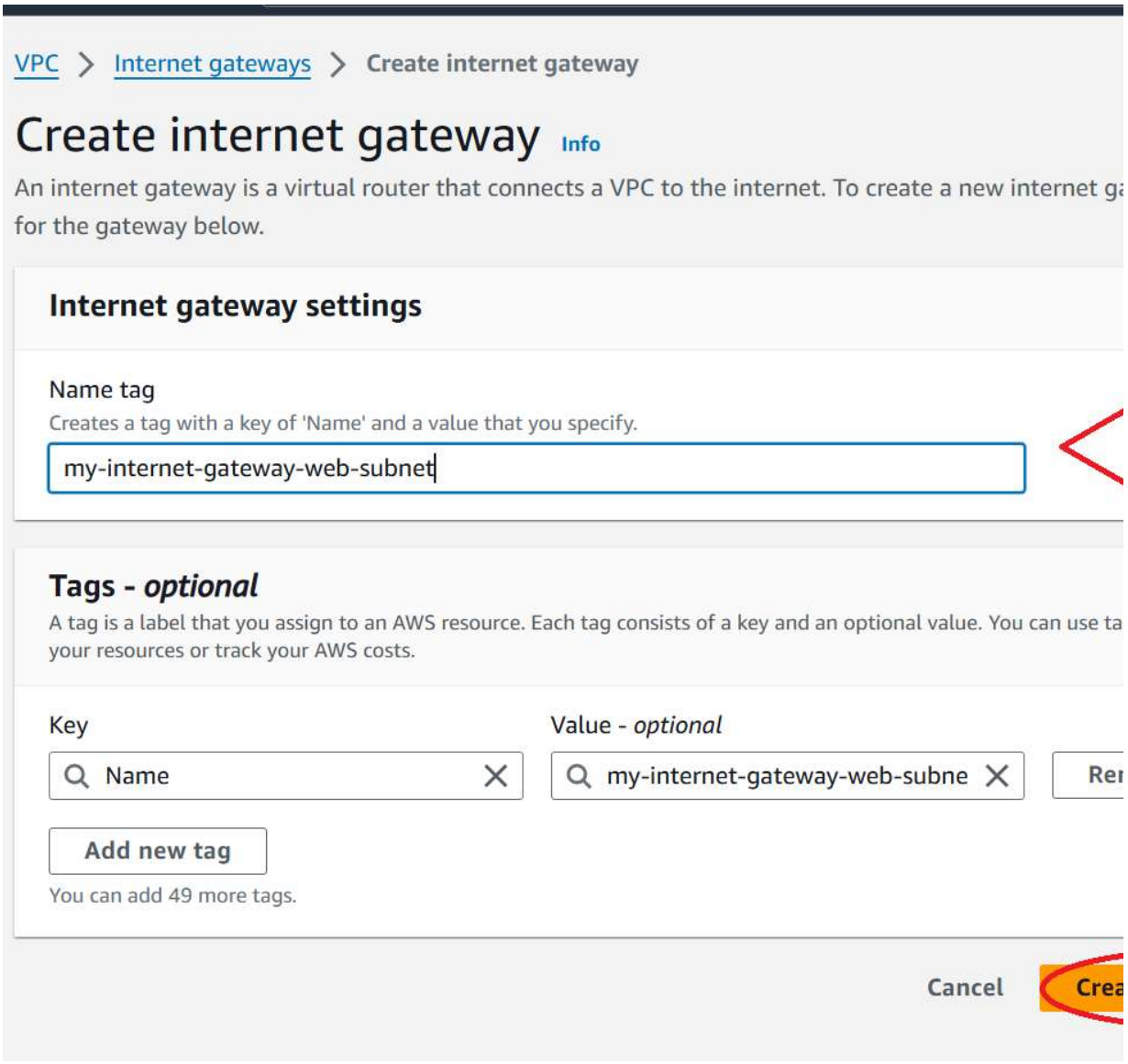
8) So my two subnets are created.



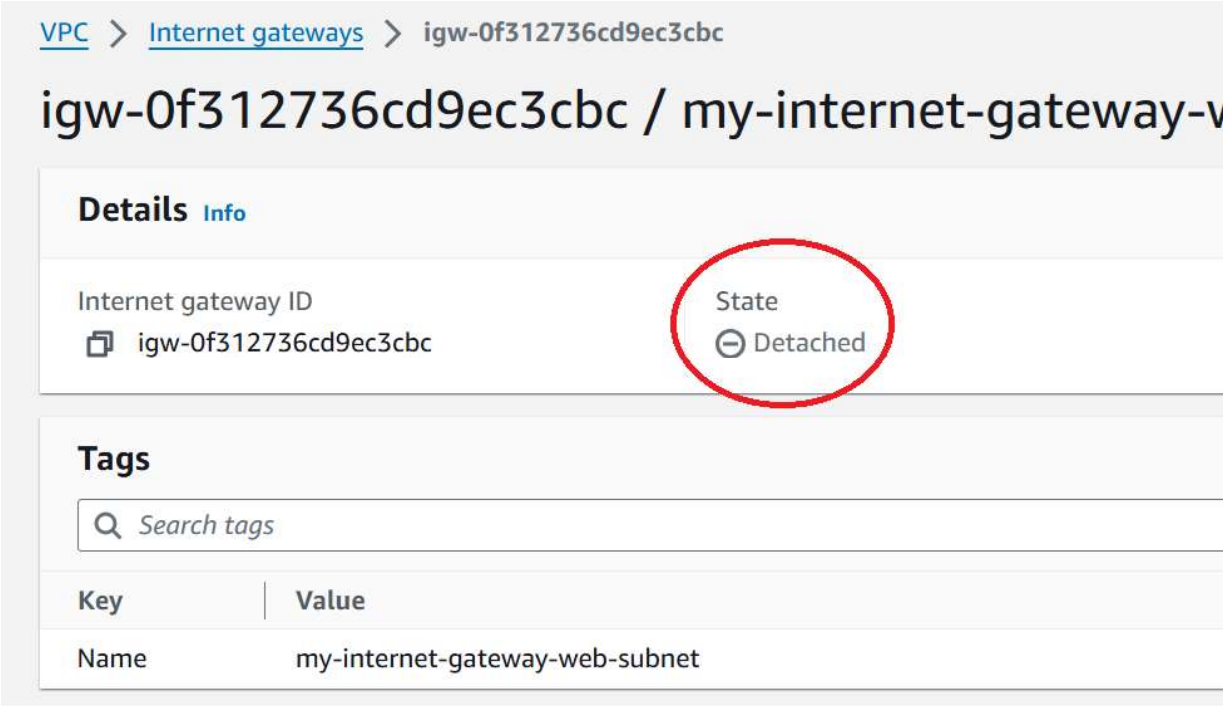
9) To give internet access to web subnet we have to create internet gateway. So under VPC dashboard click internet gateway and create internet gateway.



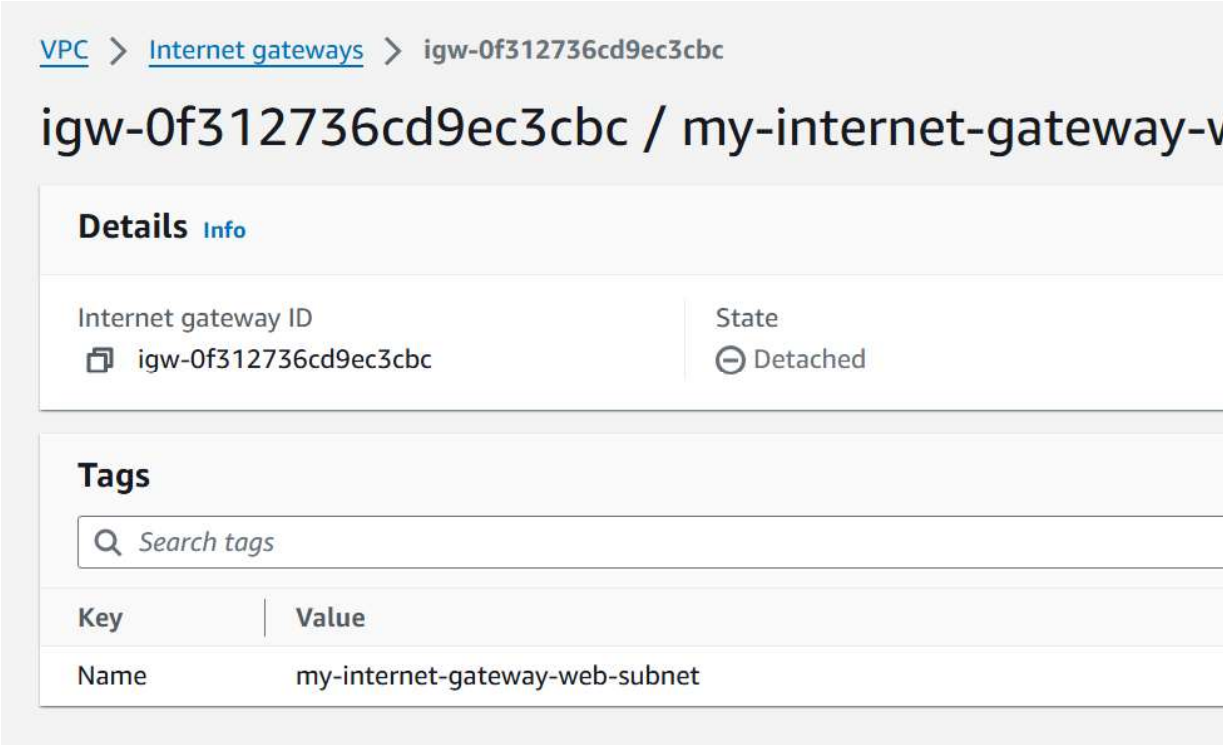
10) Give a name tag and click create internet gateway.



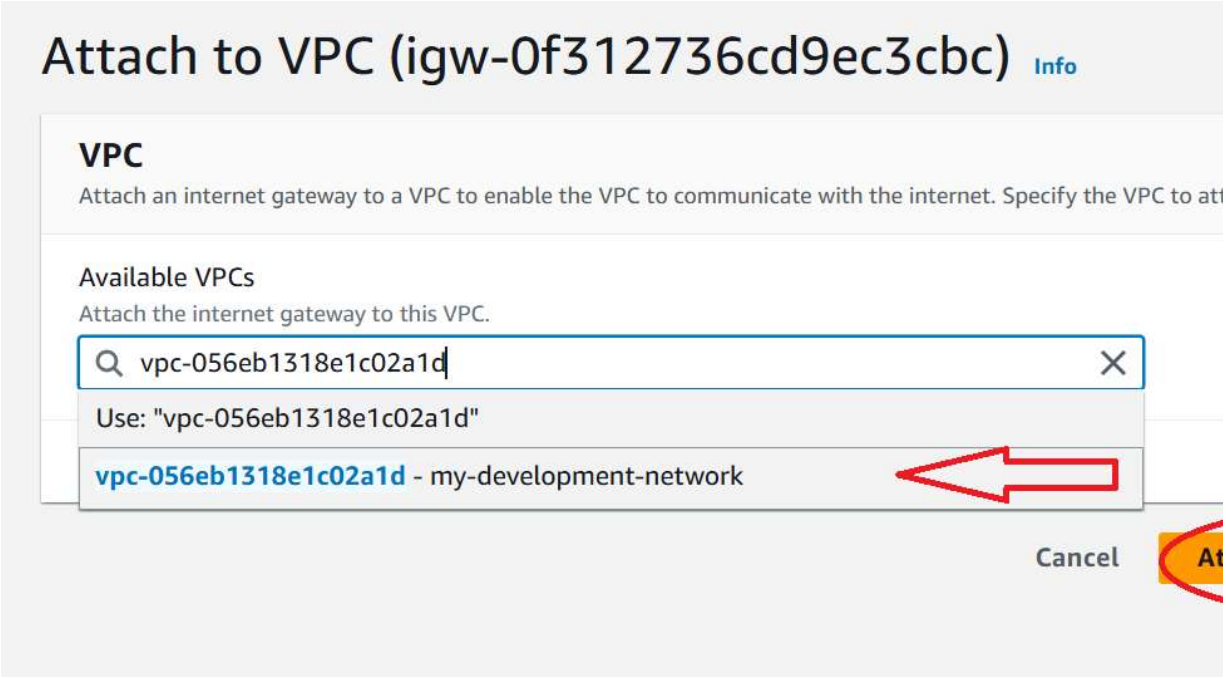
11) We can see our internet gateway is created but it is detached. So we need to attach it.



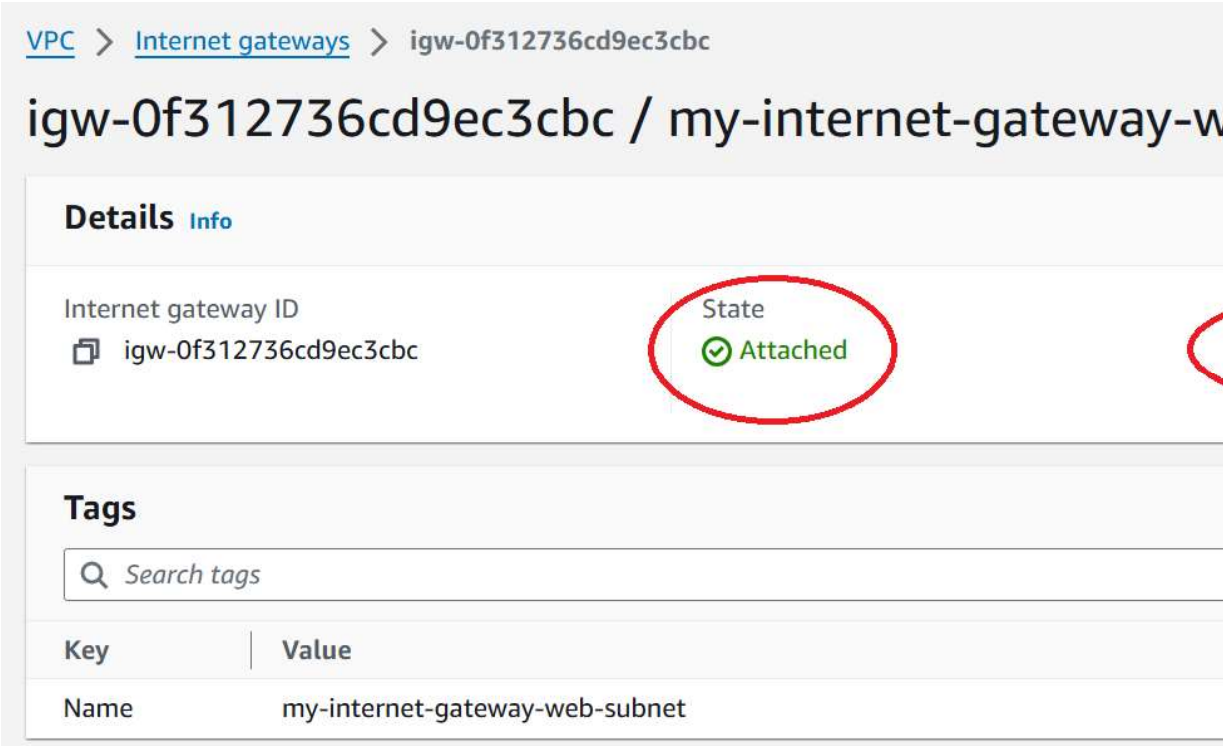
12) Click on actions and attach to VPC.



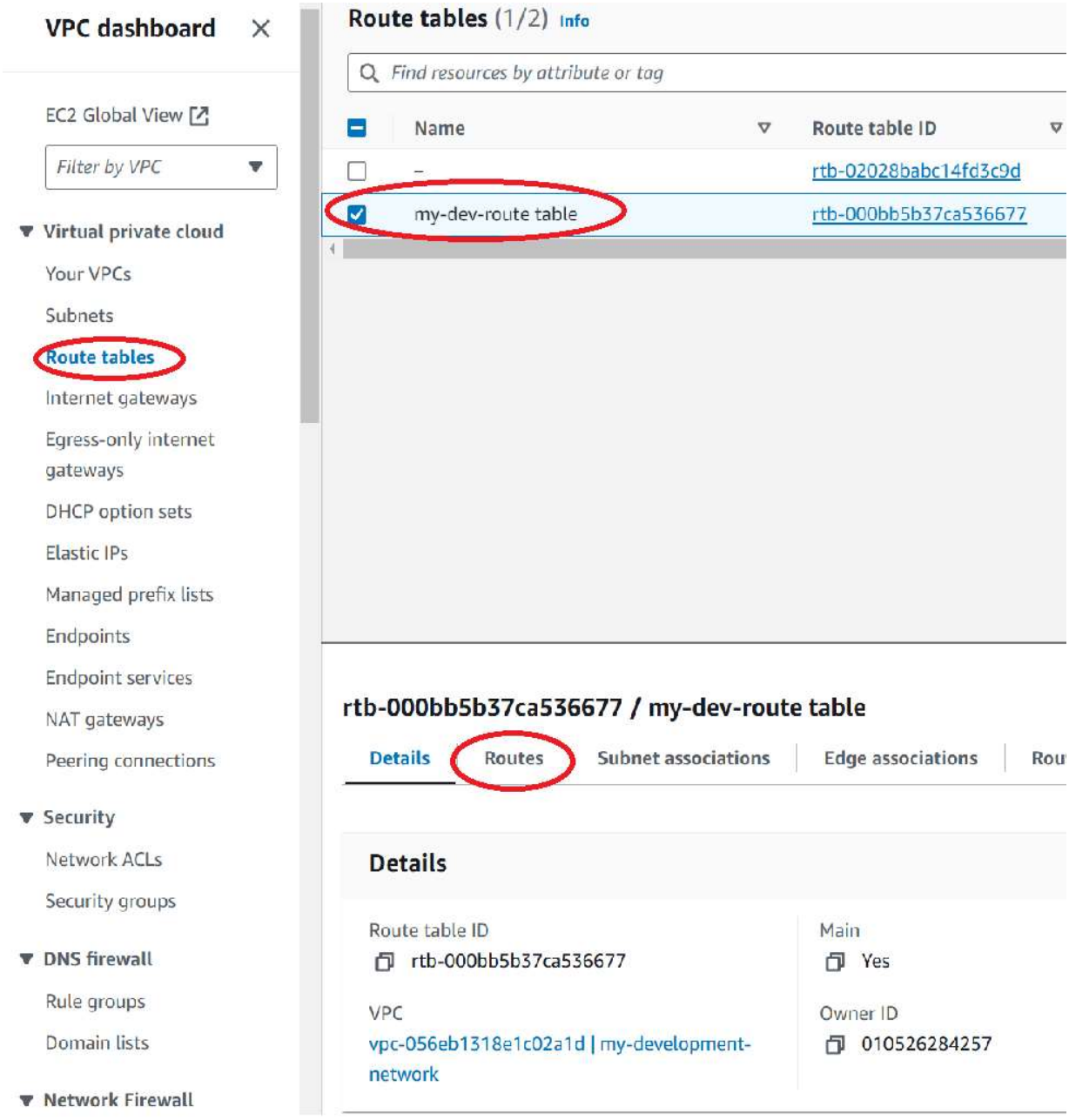
13) Choose the VPC we created and attach internet gateway.



14) Now we can see our gateway is attached and its showing our VPC ID .

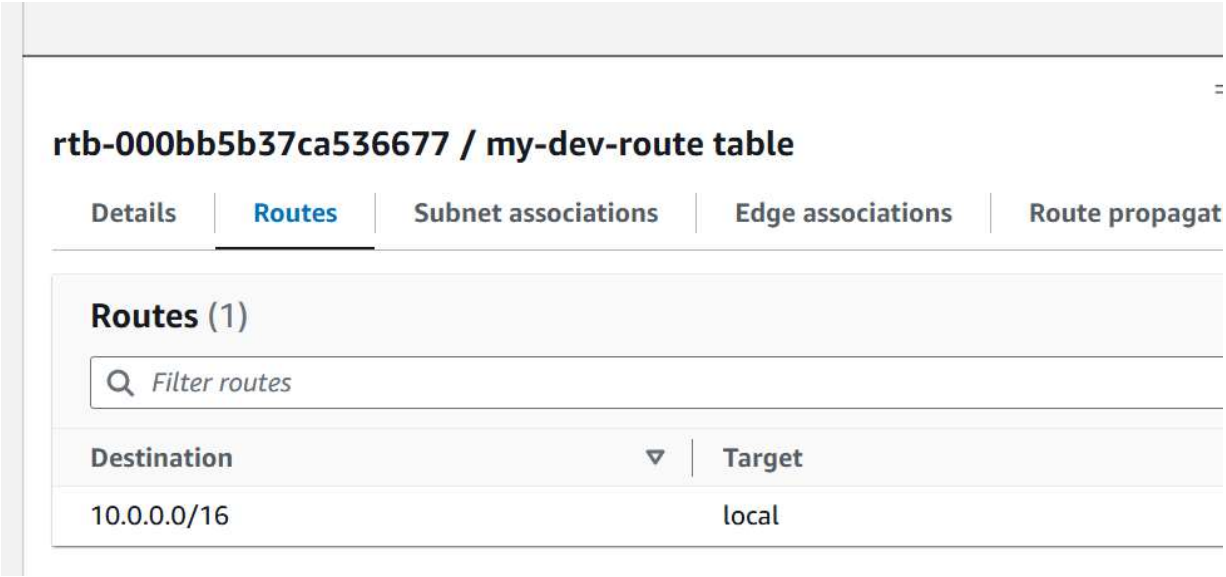


15) So under VPC dashboard click route table and we can see route table associated with our VPC . click on routes.

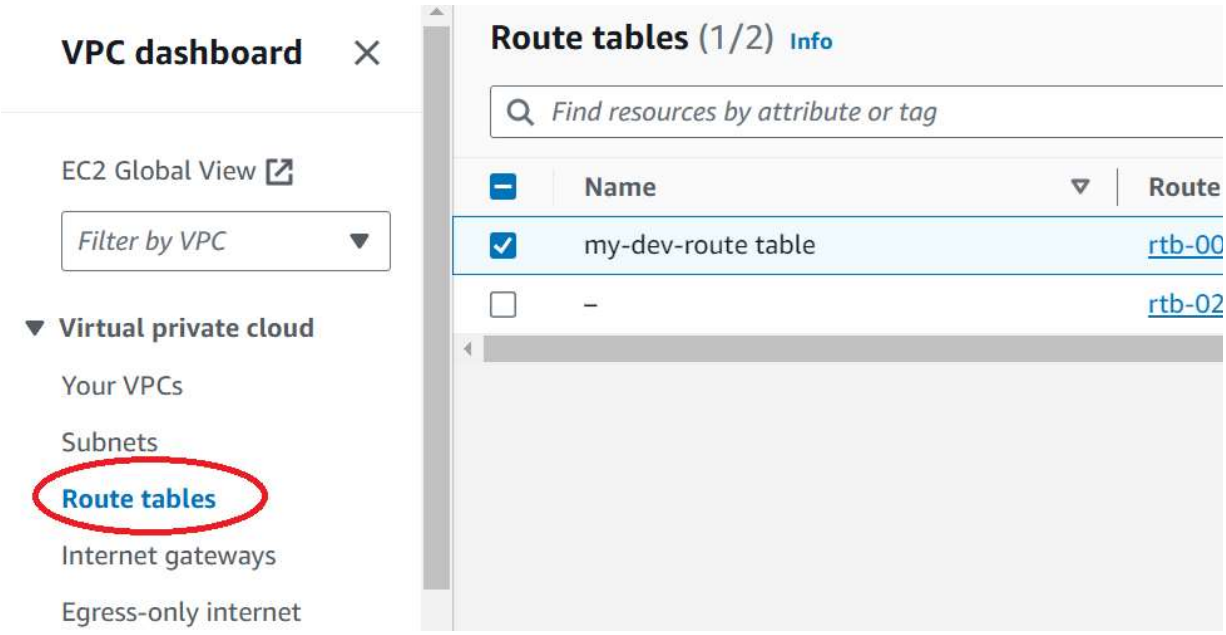


16) Here we can see only 1 route which states that our all resources inside our vpc can communicate

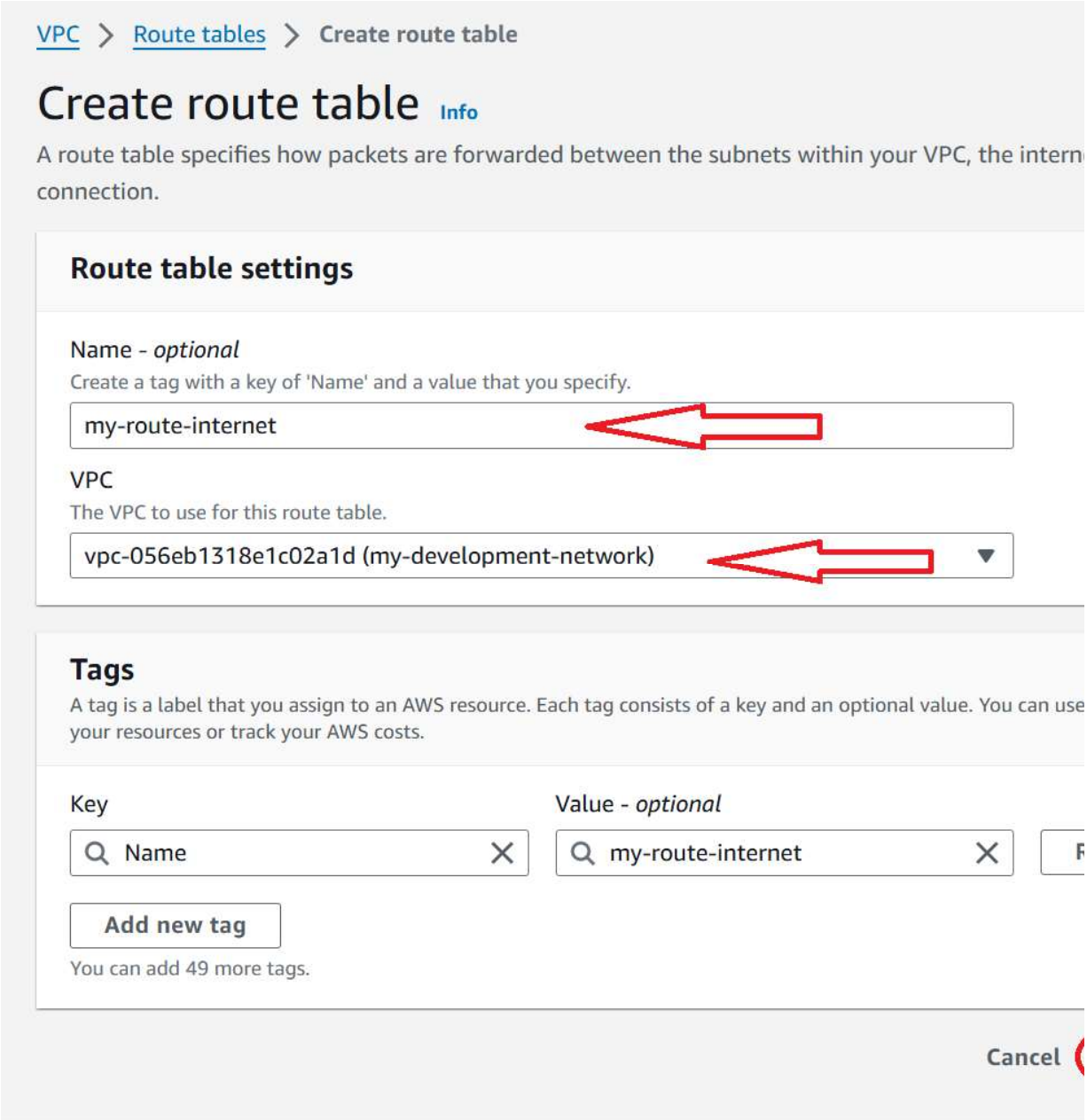
with each other but there is no route to connect to internet. So we will make a new route.



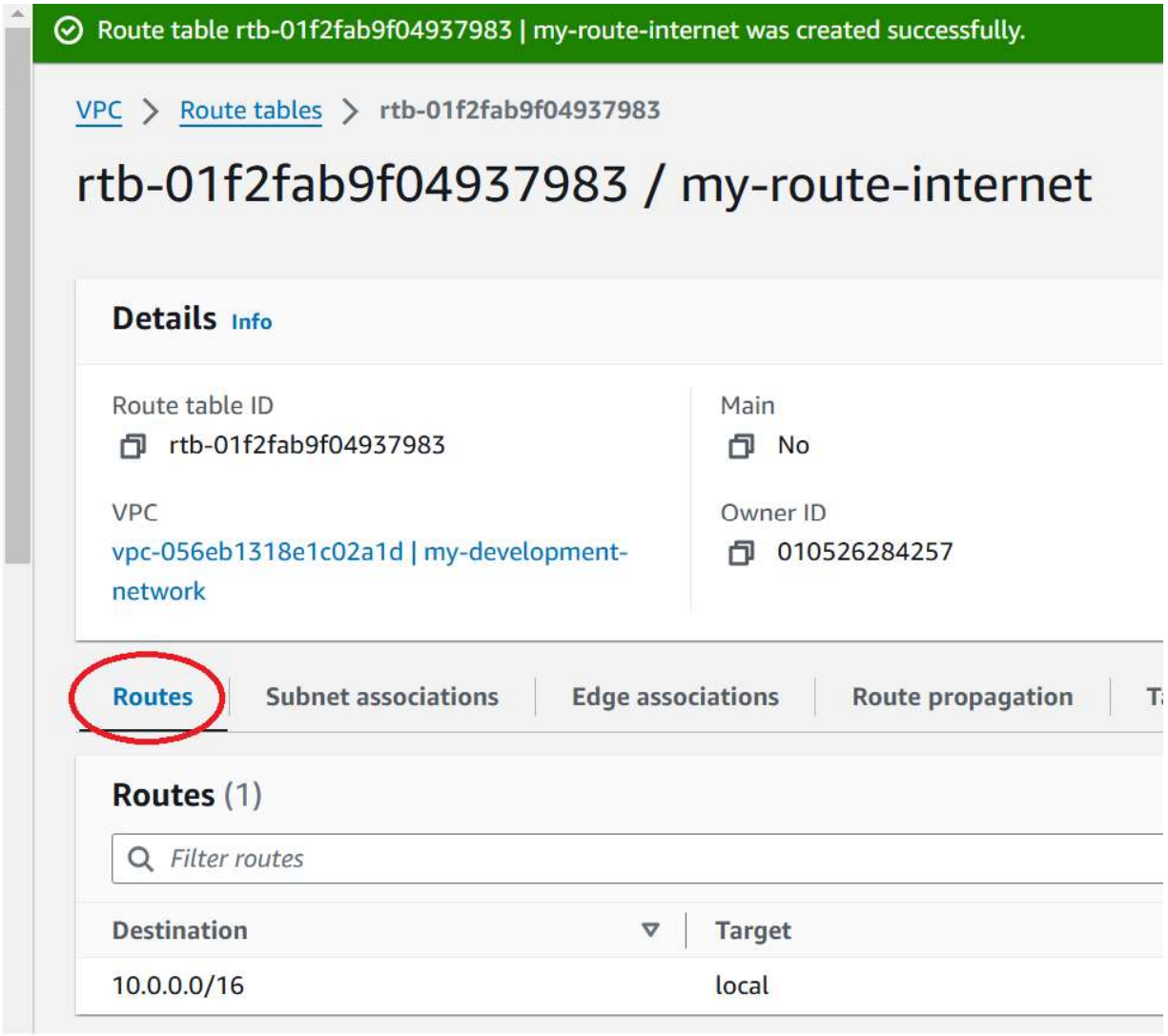
17) Under VPC dashboard click route tables and create route table.



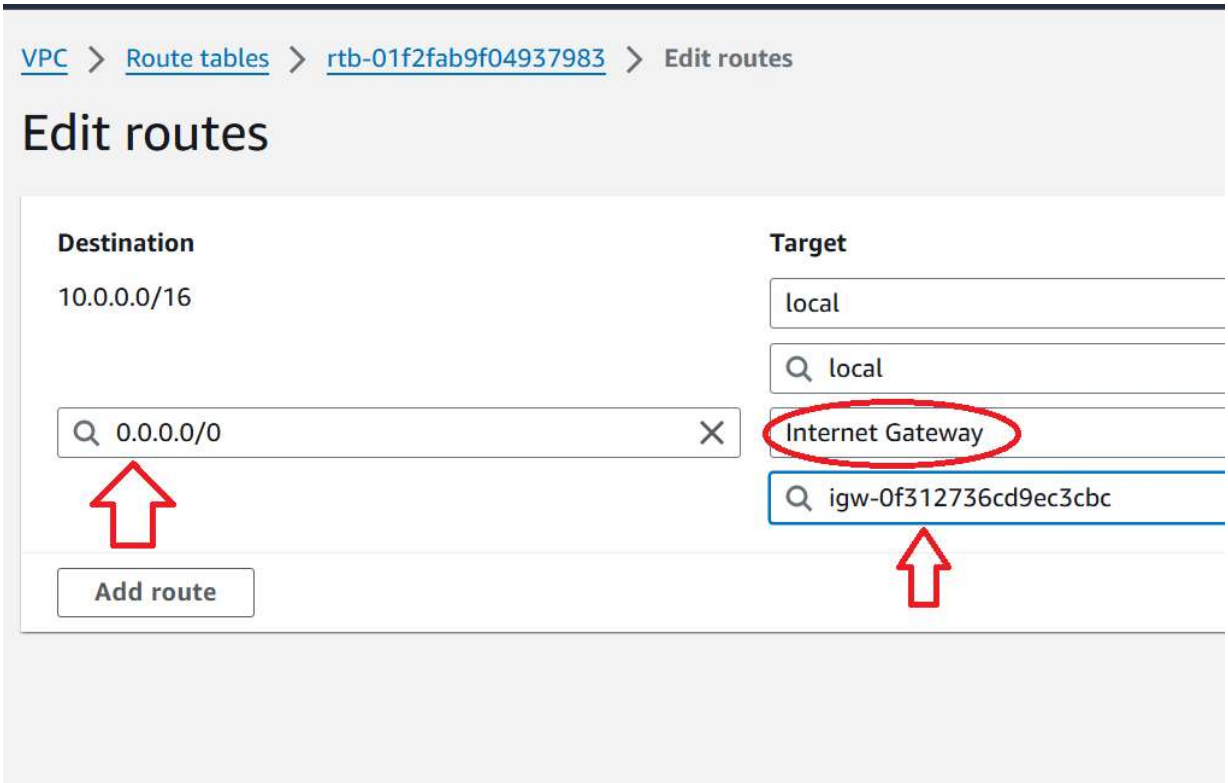
18) Give a name and choose the development VPC and create route table.



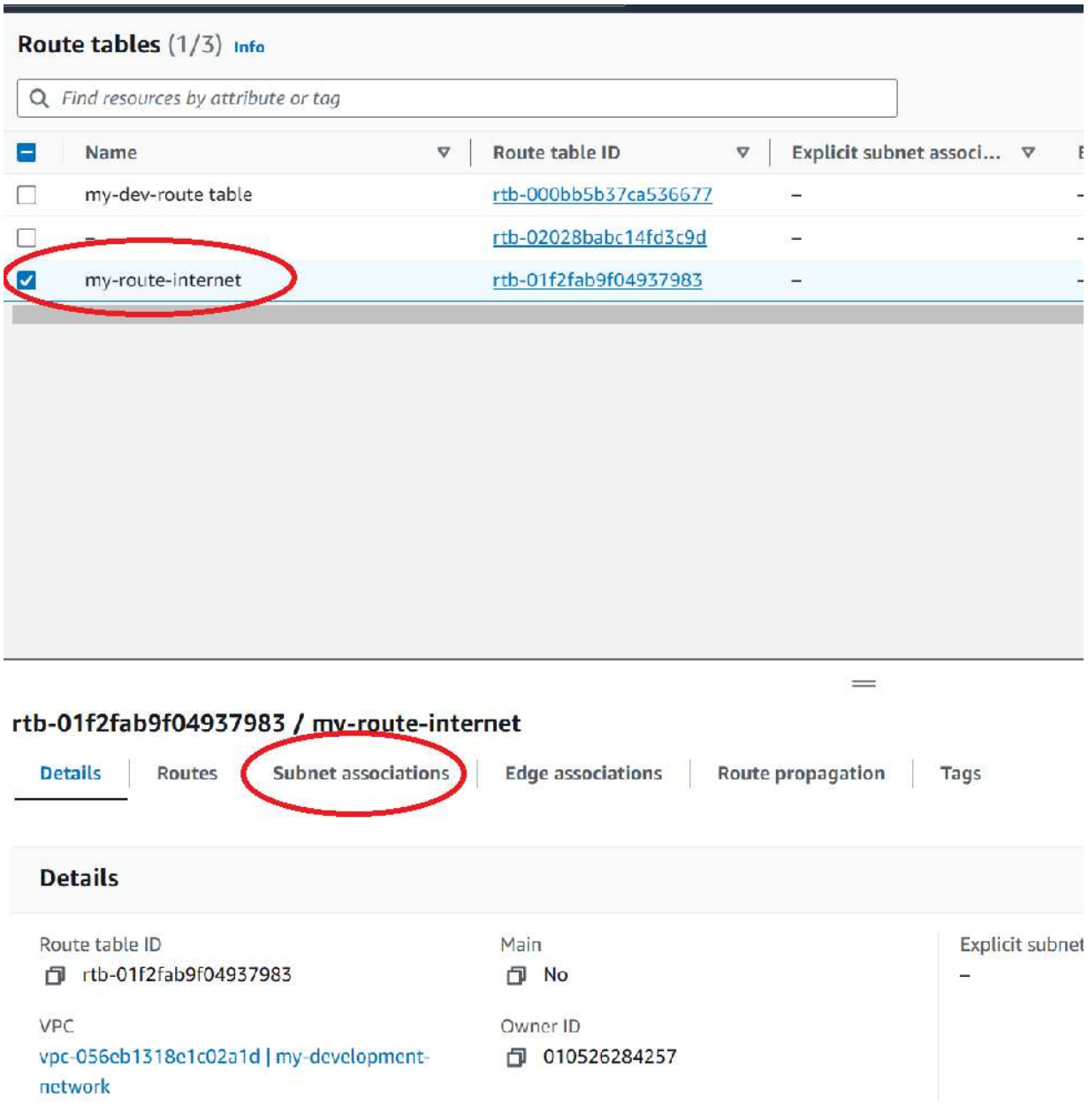
19) The route table is created . we can see there is a default route but we want route to internet. So click on edit routes.



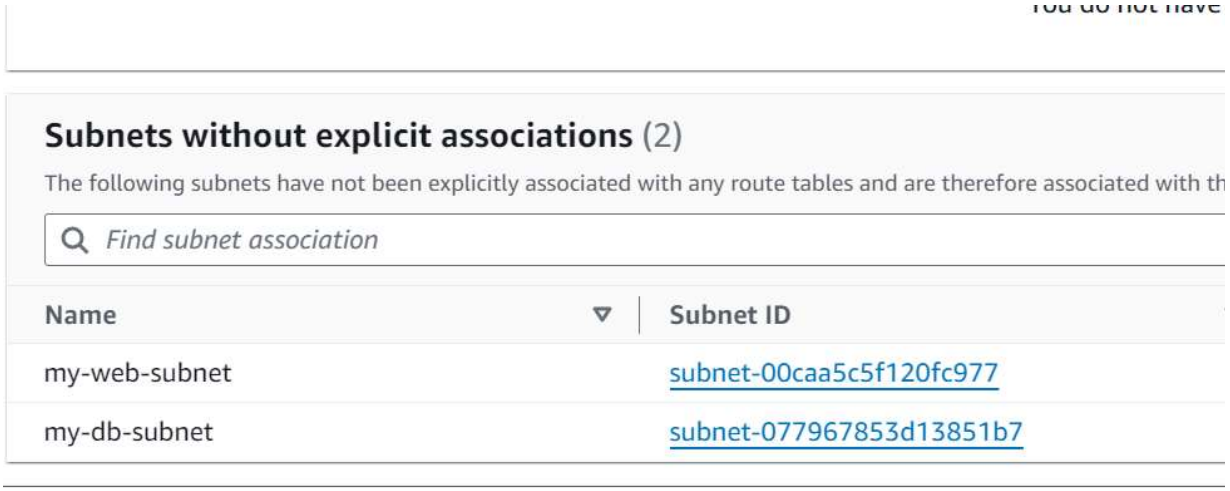
20) Under destination give 0.0.0.0/0 which is route to internet via target is our internet gateway which we created and click save changes.



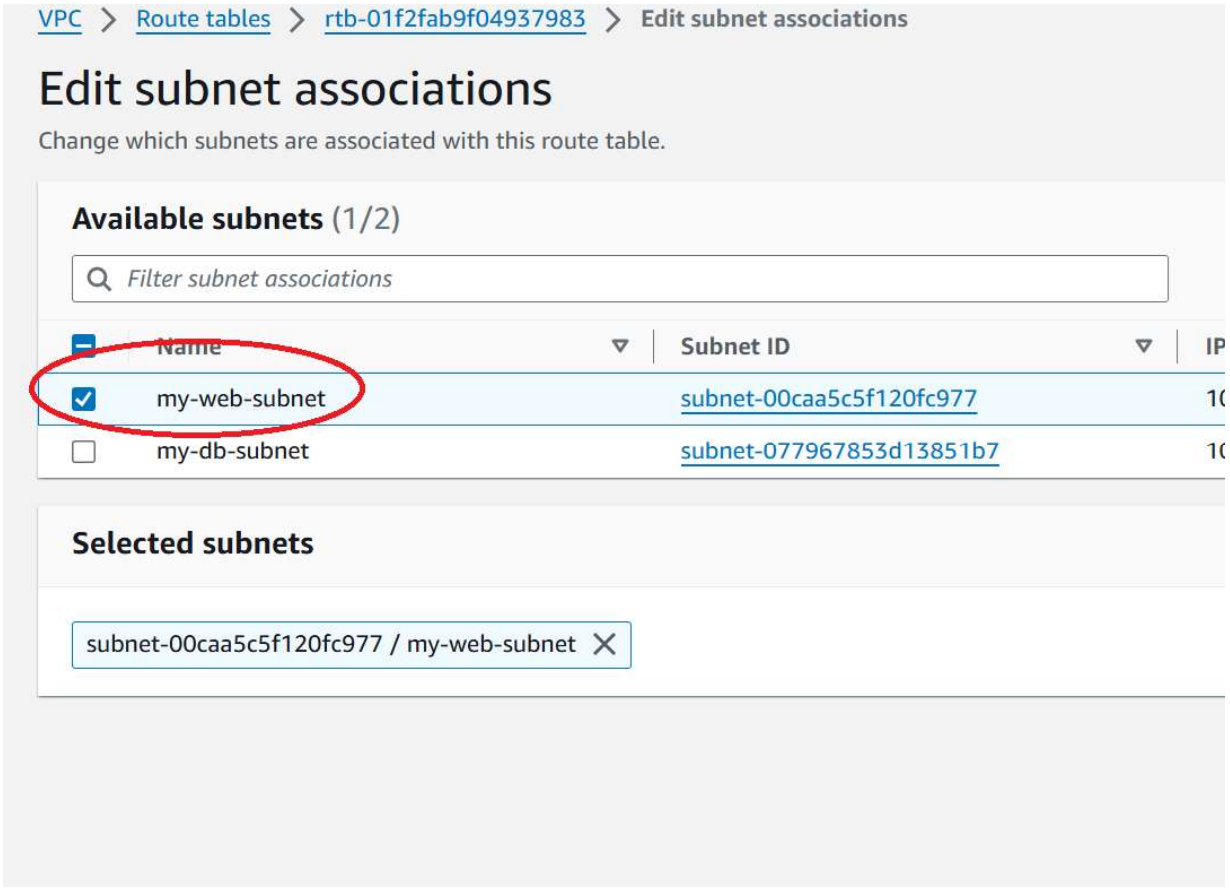
21) Now we have to associate our subnets with the route table. We have two subnets. One can access internet and the other one cant. So select the route table which can access internet and click subnet associations.



22) We can see both of our subnets available.
Click edit subnet associations.

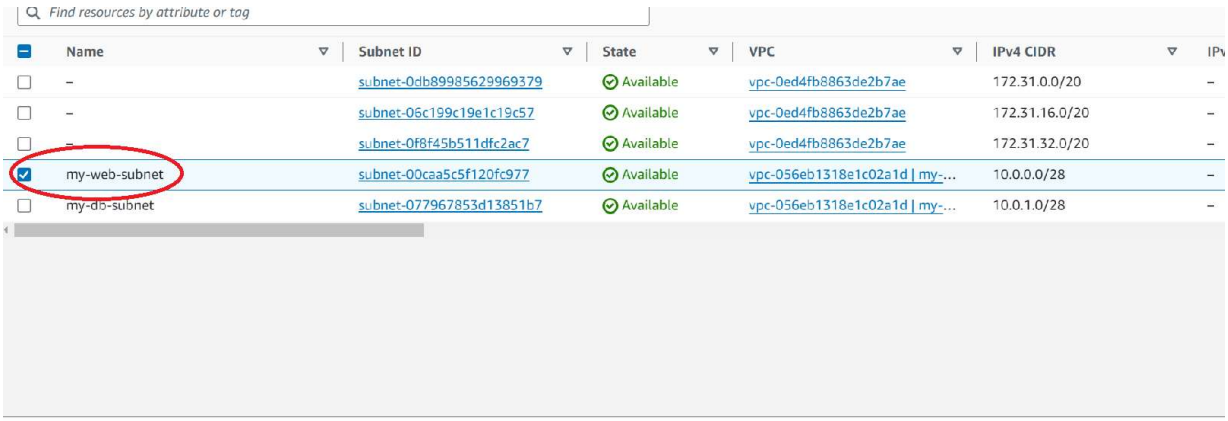


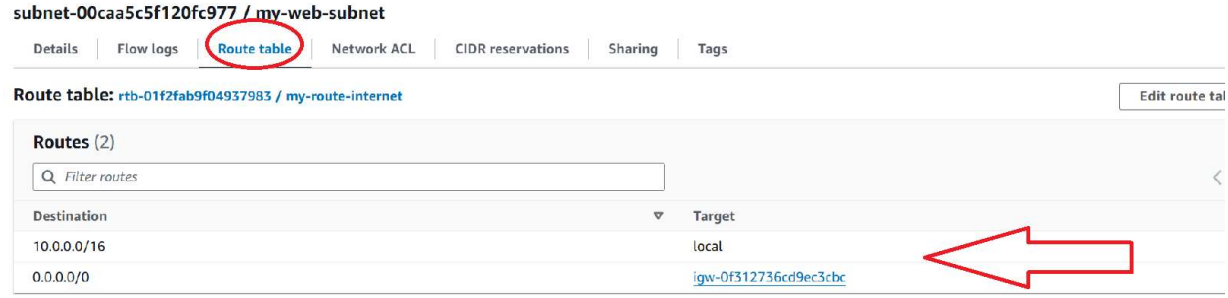
23) Choose my-web subnet and click save association.

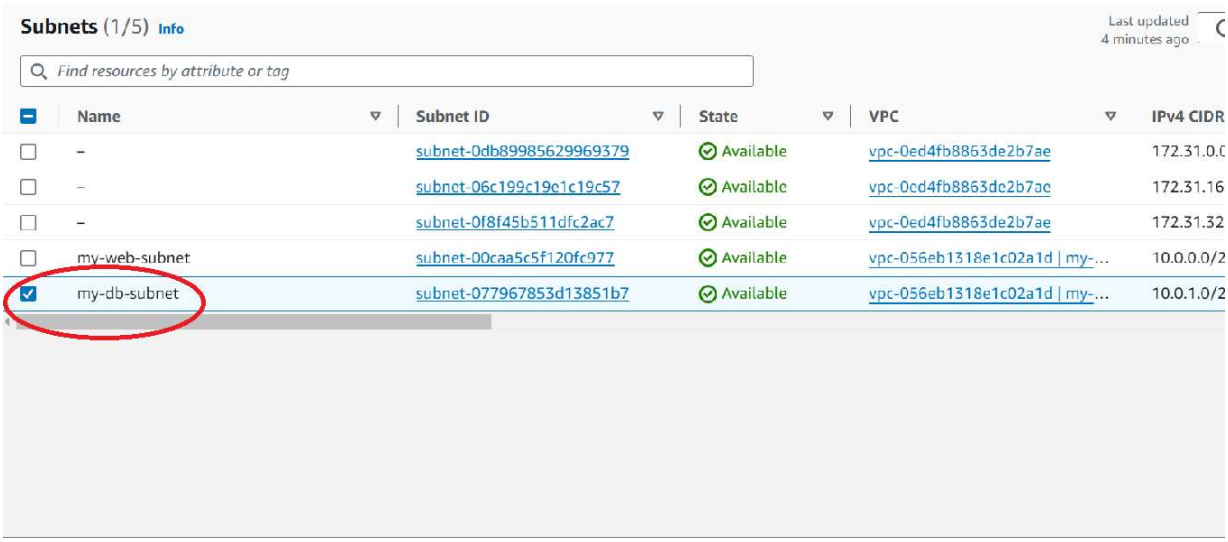


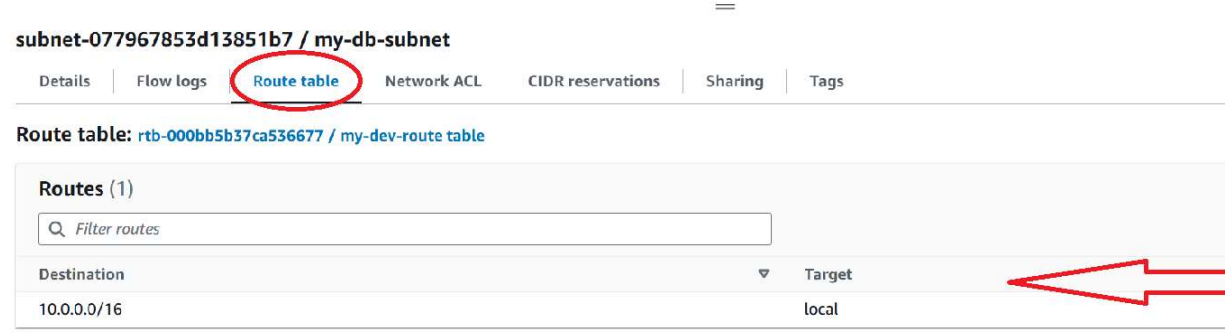
24) Similarly associate the db subnet with my-dev route table which cant access internet. So we can see that my web subnet has 2 routes associated

with it and my db subnet has 1 route associated with it.


Find resources by attribute or tag
Name
Subnet ID
State
VPC
IPv4 CIDR
– subnet-0db89985629969379 Available vpc-0ed4fb8863de2b7ae 172.31.0.0/20
– subnet-06c199c19e1c19c57 Available vpc-0ed4fb8863de2b7ae 172.31.16.0/20
– subnet-0f8f45b511dfc2ac7 Available vpc-0ed4fb8863de2b7ae 172.31.32.0/20
my-web-subnet subnet-00caa5c5f120fc977 Available vpc-056eb1318e1c02a1d | my-... 10.0.0.0/28
my-db-subnet subnet-077967853d13851b7 Available vpc-056eb1318e1c02a1d | my-... 10.0.1.0/28

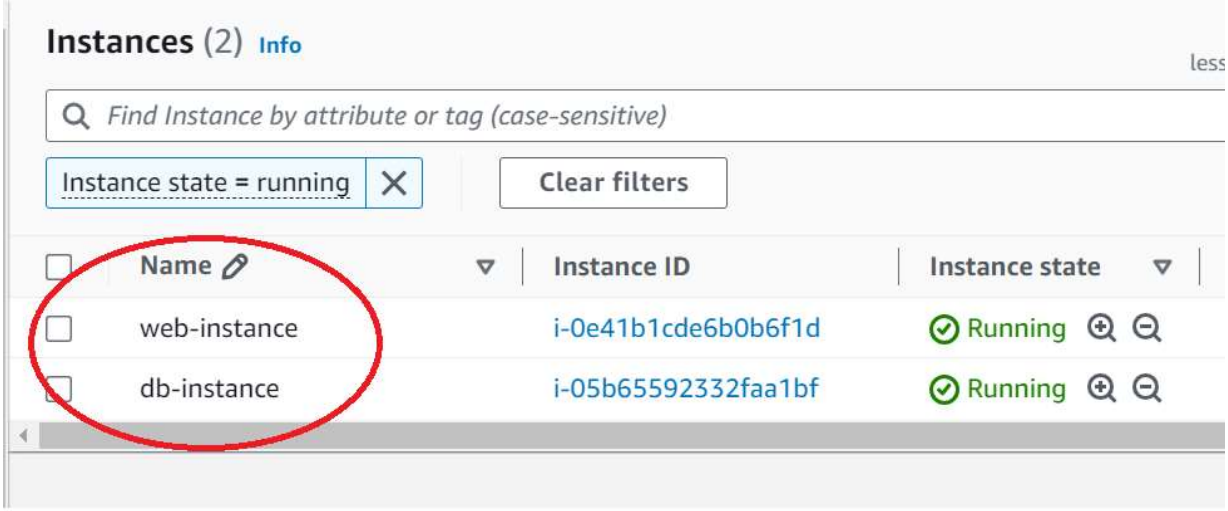

subnet-00caa5c5f120fc977 / my-web-subnet
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags
Route table: rtb-01f2fab9f04937983 / my-route-internet
Edit route tal
Routes (2)
Filter routes
Destination
Target
10.0.0.0/16 local
0.0.0.0/0 igw-0f312736cd9ec3cbc


Subnets (1/5) Info
Last updated 4 minutes ago
Find resources by attribute or tag
Name
Subnet ID
State
VPC
IPv4 CIDR
– subnet-0db89985629969379 Available vpc-0ed4fb8863de2b7ae 172.31.0.0
– subnet-06c199c19e1c19c57 Available vpc-0ed4fb8863de2b7ae 172.31.16
– subnet-0f8f45b511dfc2ac7 Available vpc-0ed4fb8863de2b7ae 172.31.32
my-web-subnet subnet-00caa5c5f120fc977 Available vpc-056eb1318e1c02a1d | my-... 10.0.0.0/2
my-db-subnet subnet-077967853d13851b7 Available vpc-056eb1318e1c02a1d | my-... 10.0.1.0/2

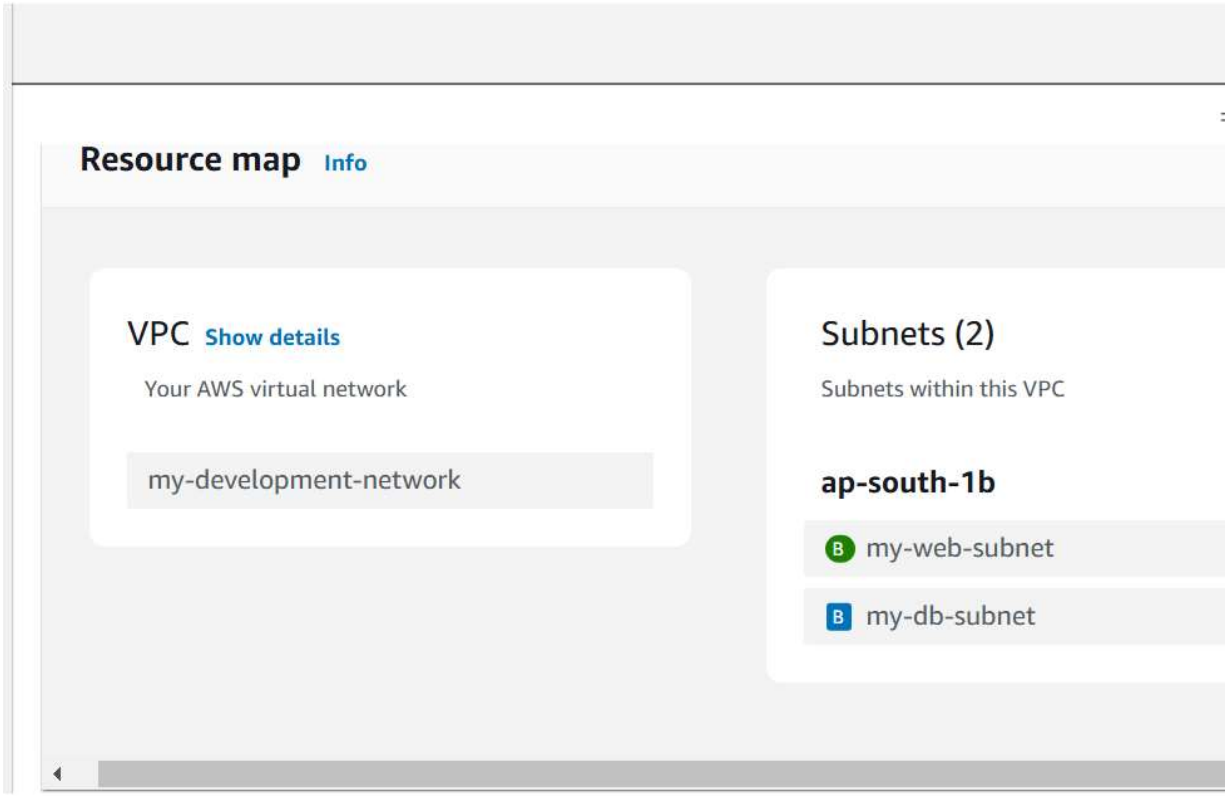

subnet-077967853d13851b7 / my-db-subnet
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags
Route table: rtb-000bb5b37ca536677 / my-dev-route table
Routes (1)
Filter routes
Destination
Target
10.0.0.0/16 local

25) We create 1 instance in each subnet.

Note:Security group rule for web-instance allow SSH and all traffic from anywhere. Security group rule for db-instance allow SSH and HTTP from the security group of the web-instance.



26) So my development VPC is created and we can see the mapping below.



Problem Statement:

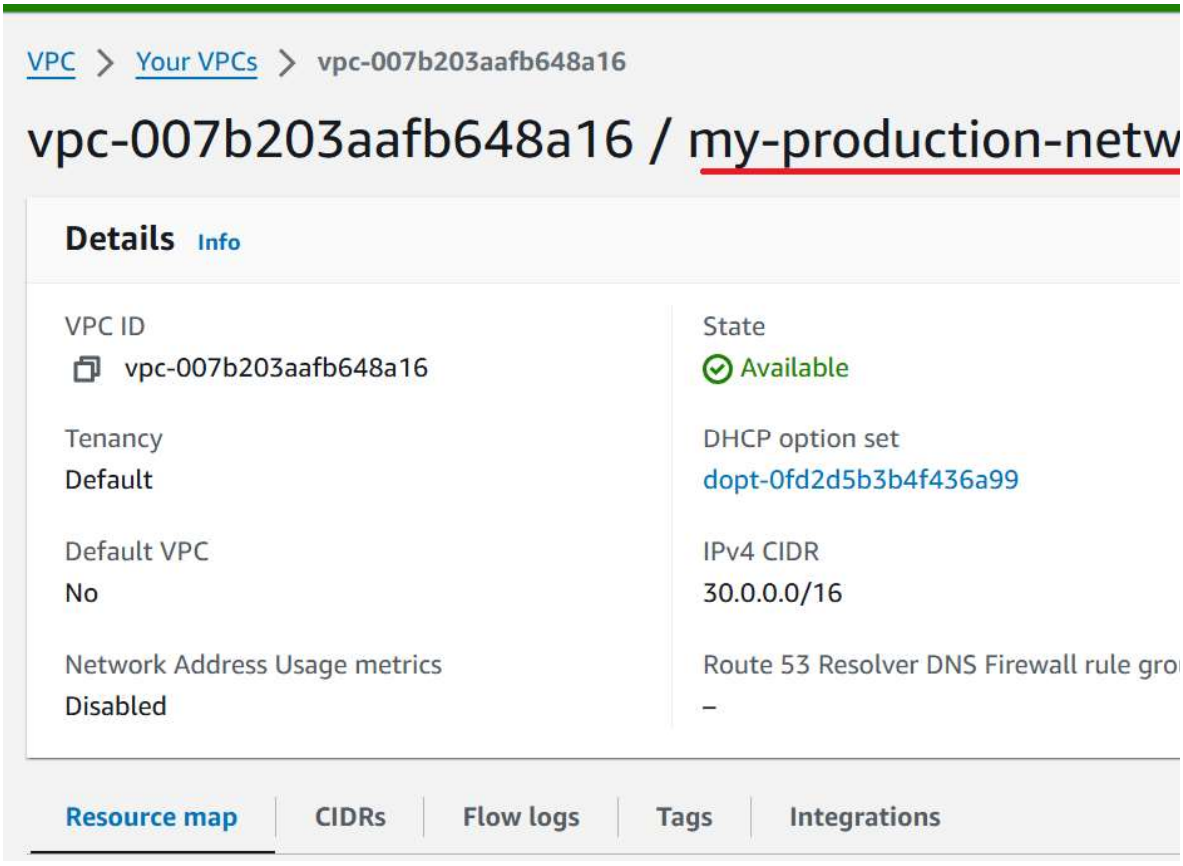
Production Network:

1. Design and build a 4-tier architecture.
2. Create 5 subnets out of which 4 should be private named app1, app2, dbcache and db and one should be public, named web.
3. Launch instances in all subnets and name them as per the subnet that they have been launched in.

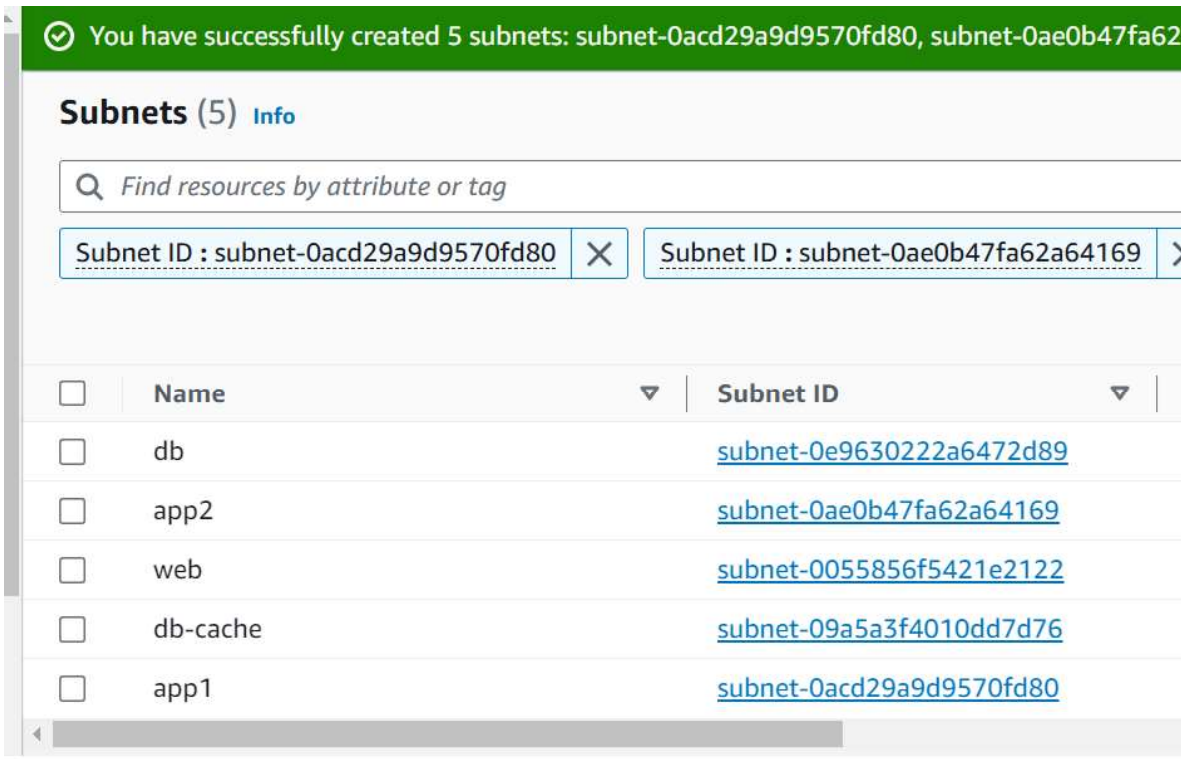
4. Allow dbcache instance and app1 subnet to send internet requests.
5. Manage security groups and NACLs.

Solution:

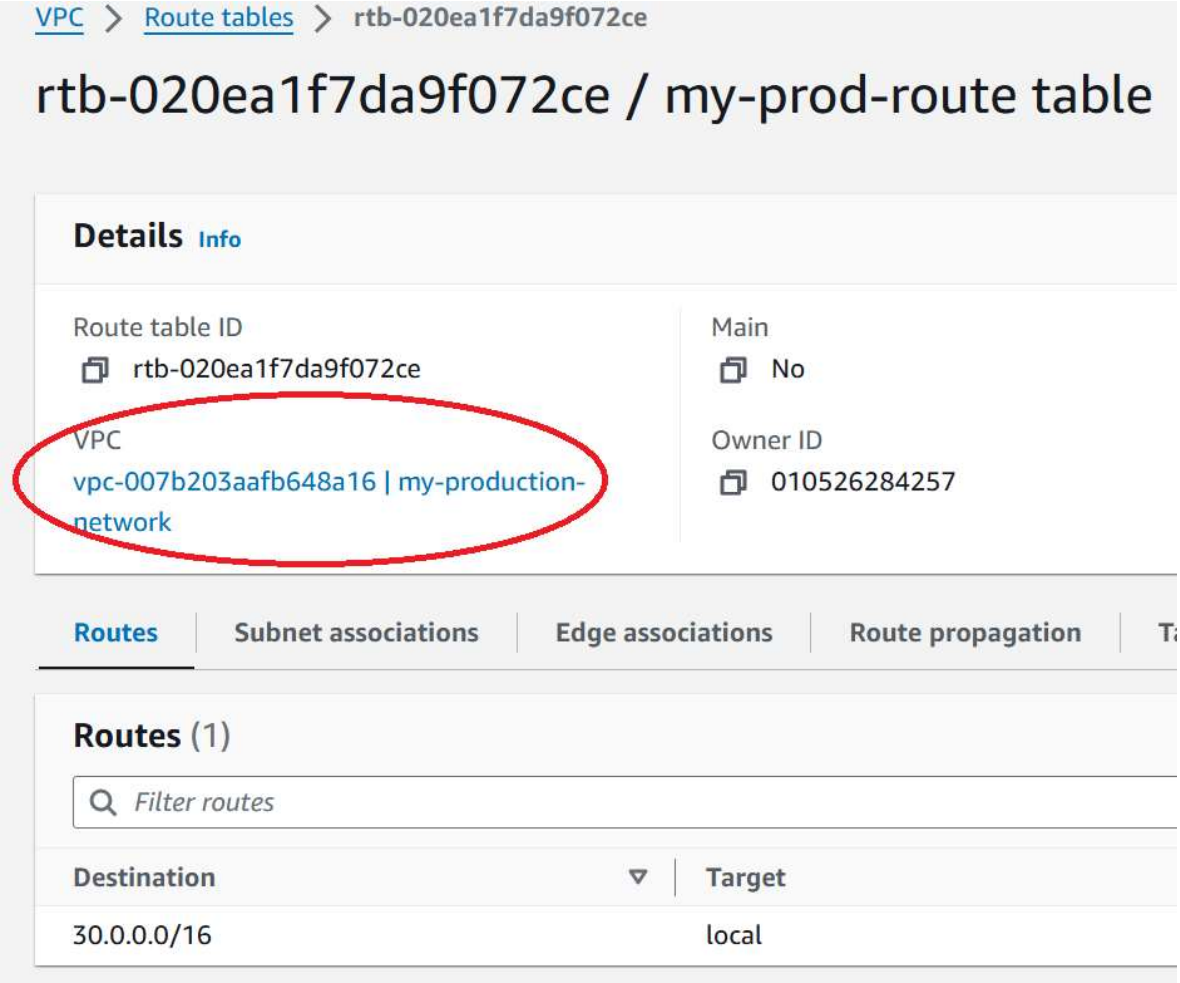
1) Create a production network VPC . The steps are same as we did in development network.



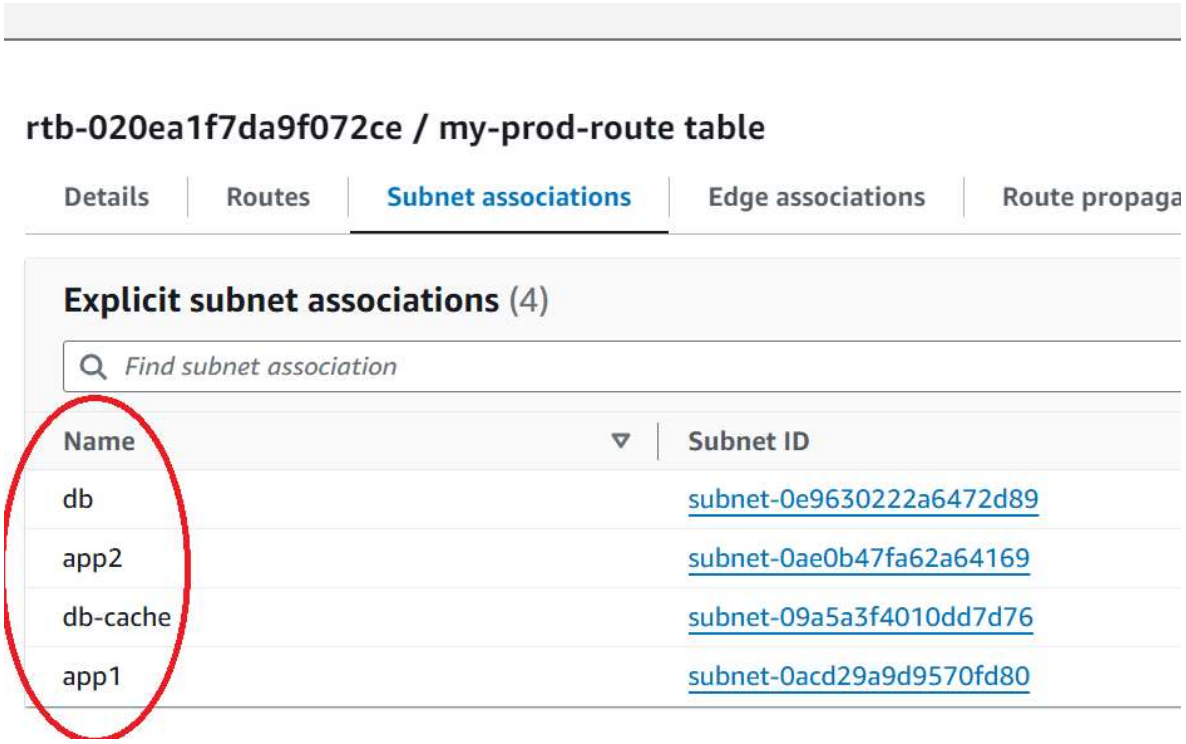
2) Create 5 subnets app1, app2, dbcache ,db and web. Make sure the CIDR do not overlap.



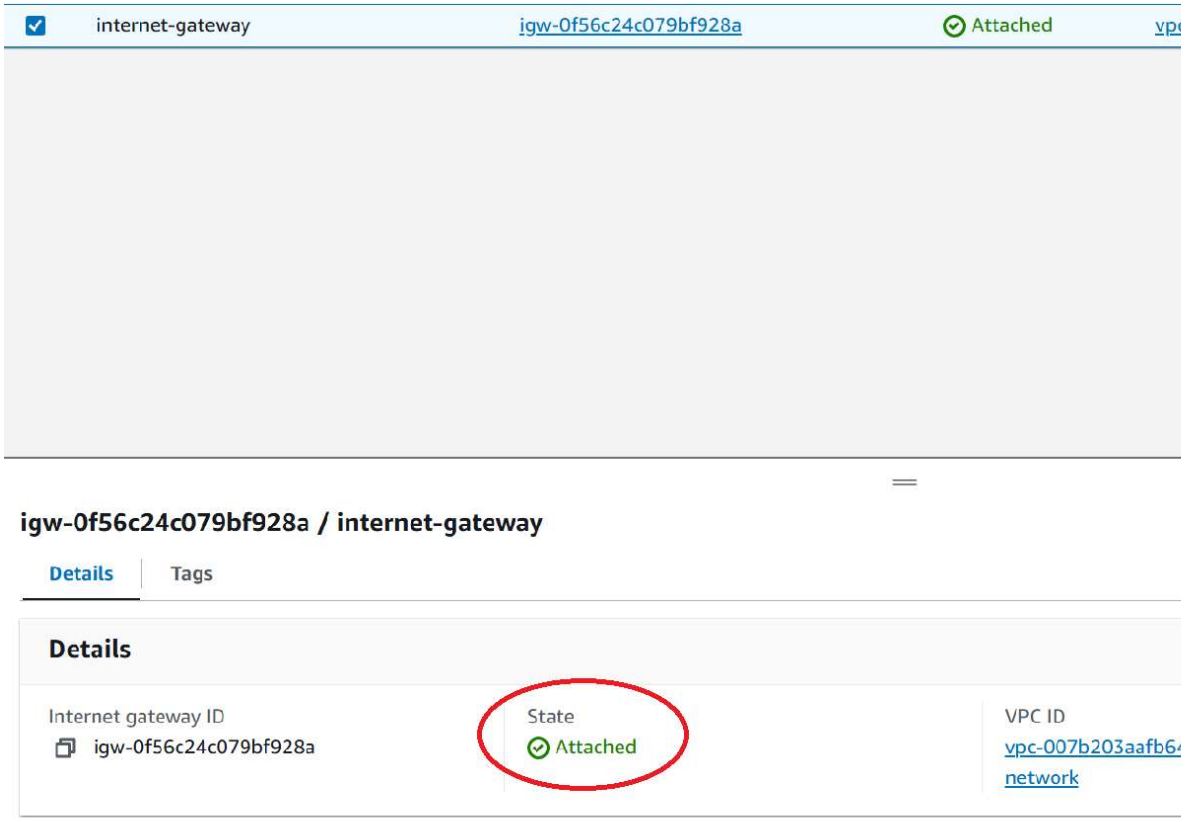
3) Create a route table and associate with production network. There is no route to internet.



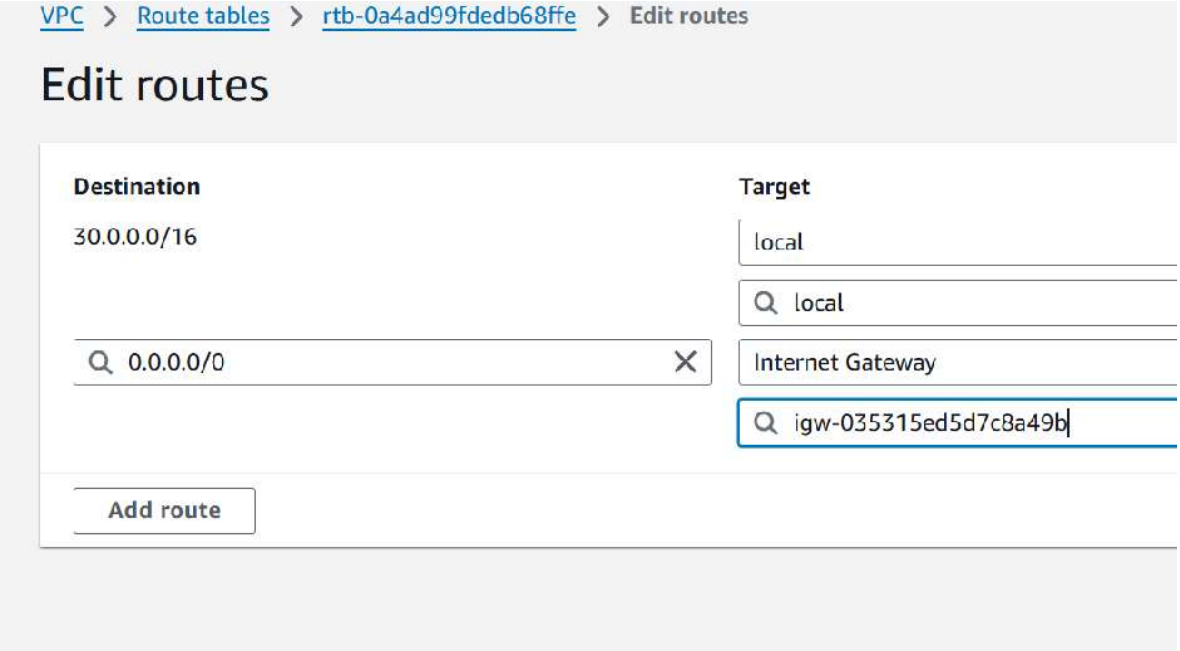
4) Under subnet association associate app1, app2, db cache and db subnets because they are private subnets.

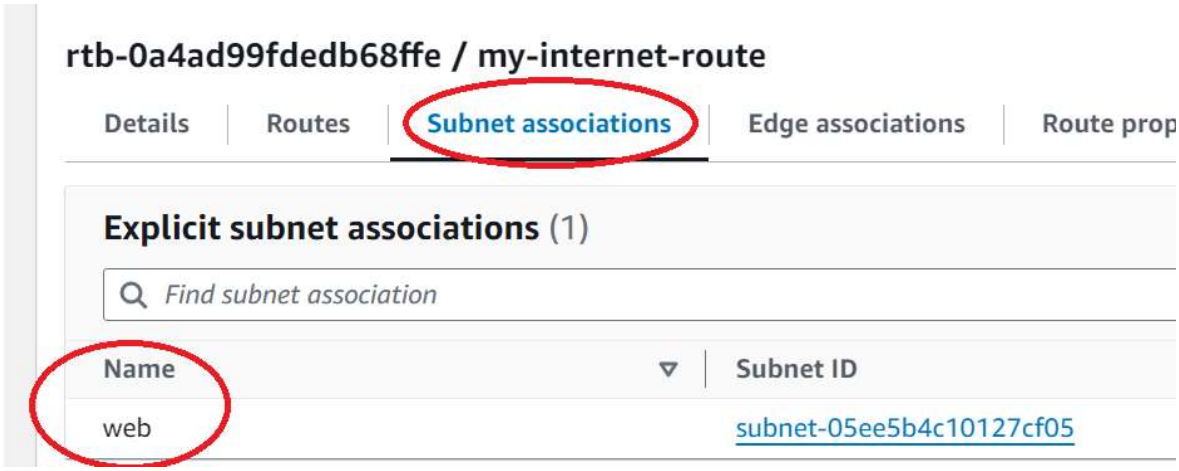


5) Create an internet gateway and attach it to production network.

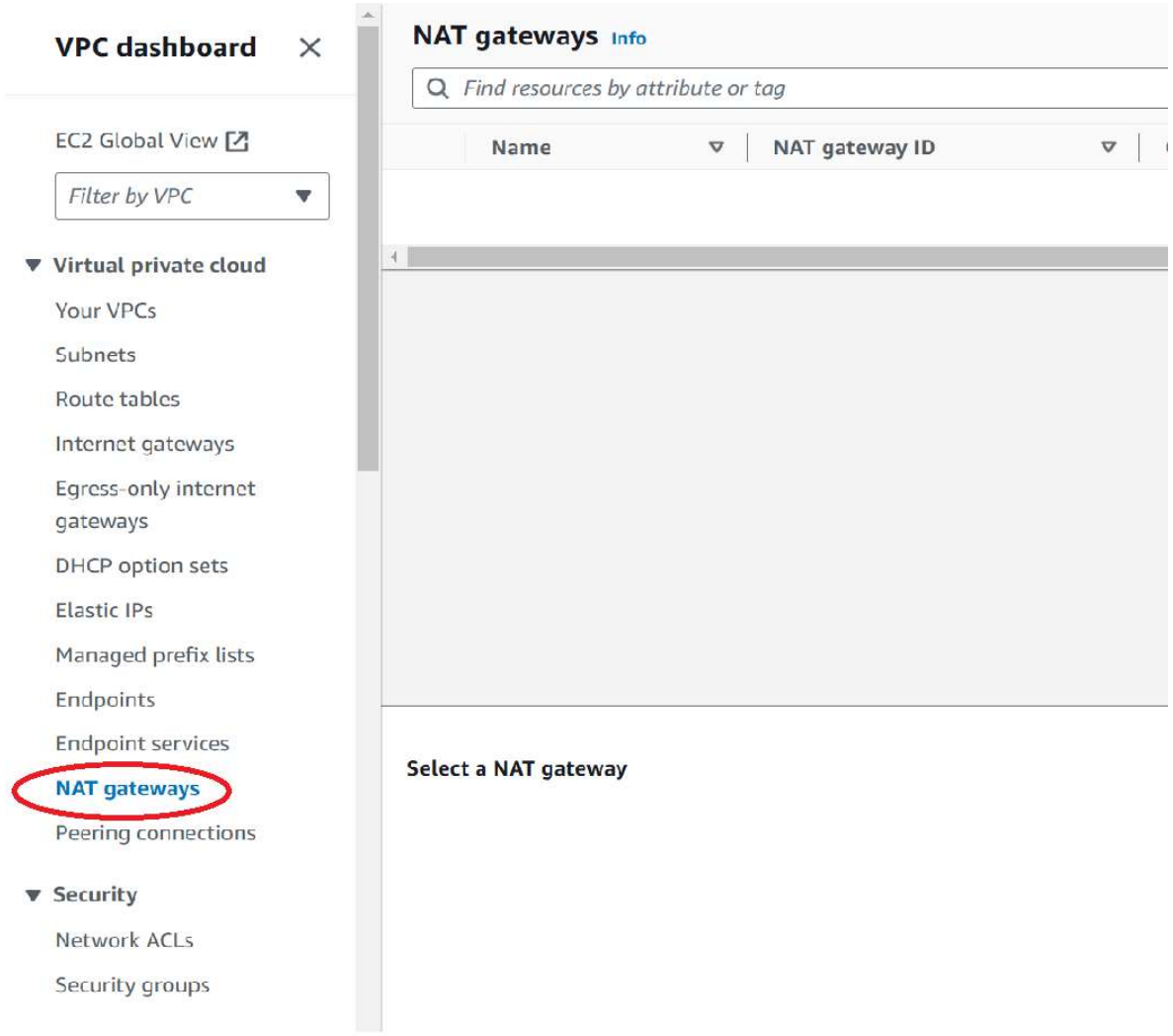


6) Create a route table with destination to internet 0.0.0.0/0 via internet gateway which we just created and attach this route table with web subnet which is public.

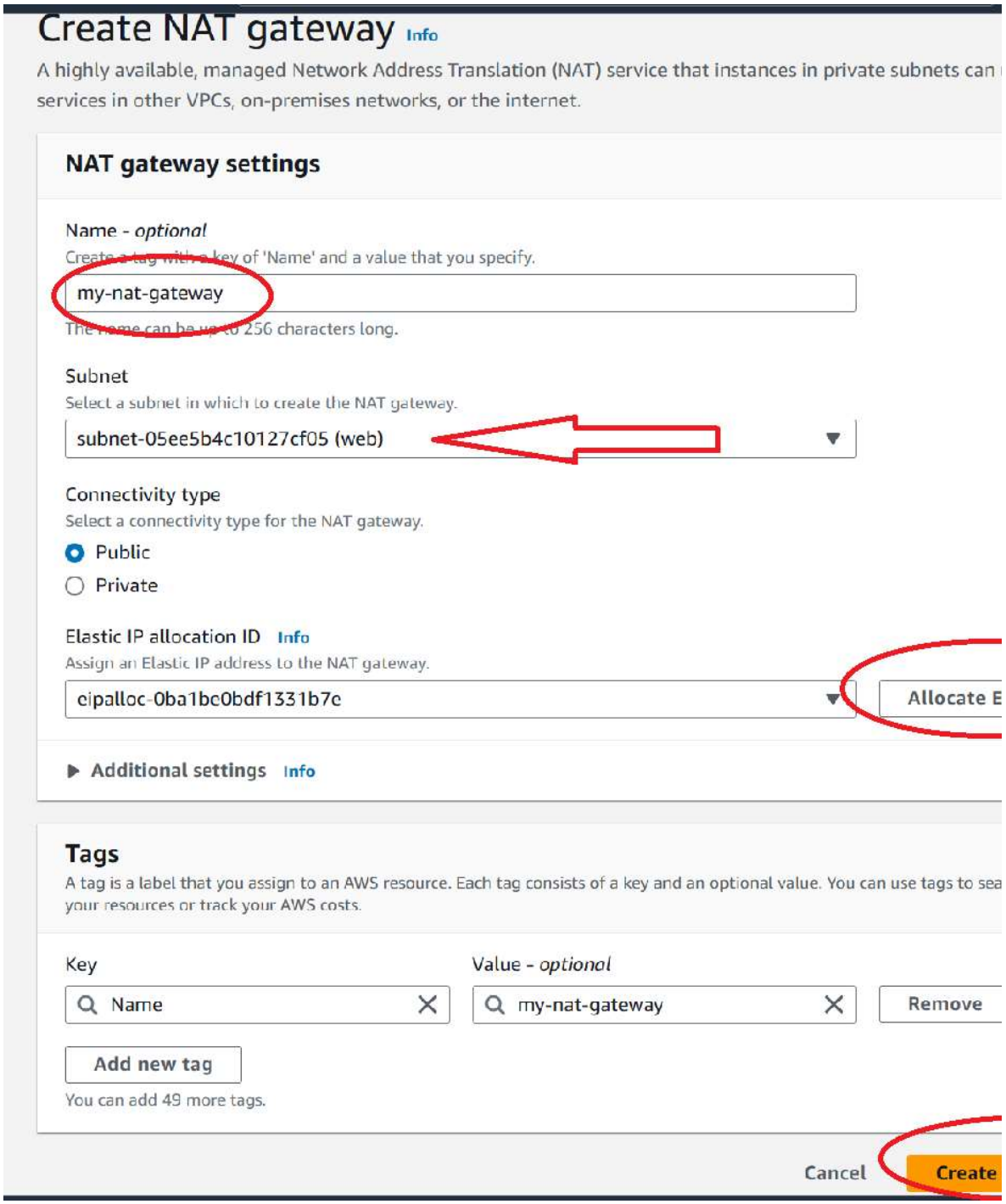




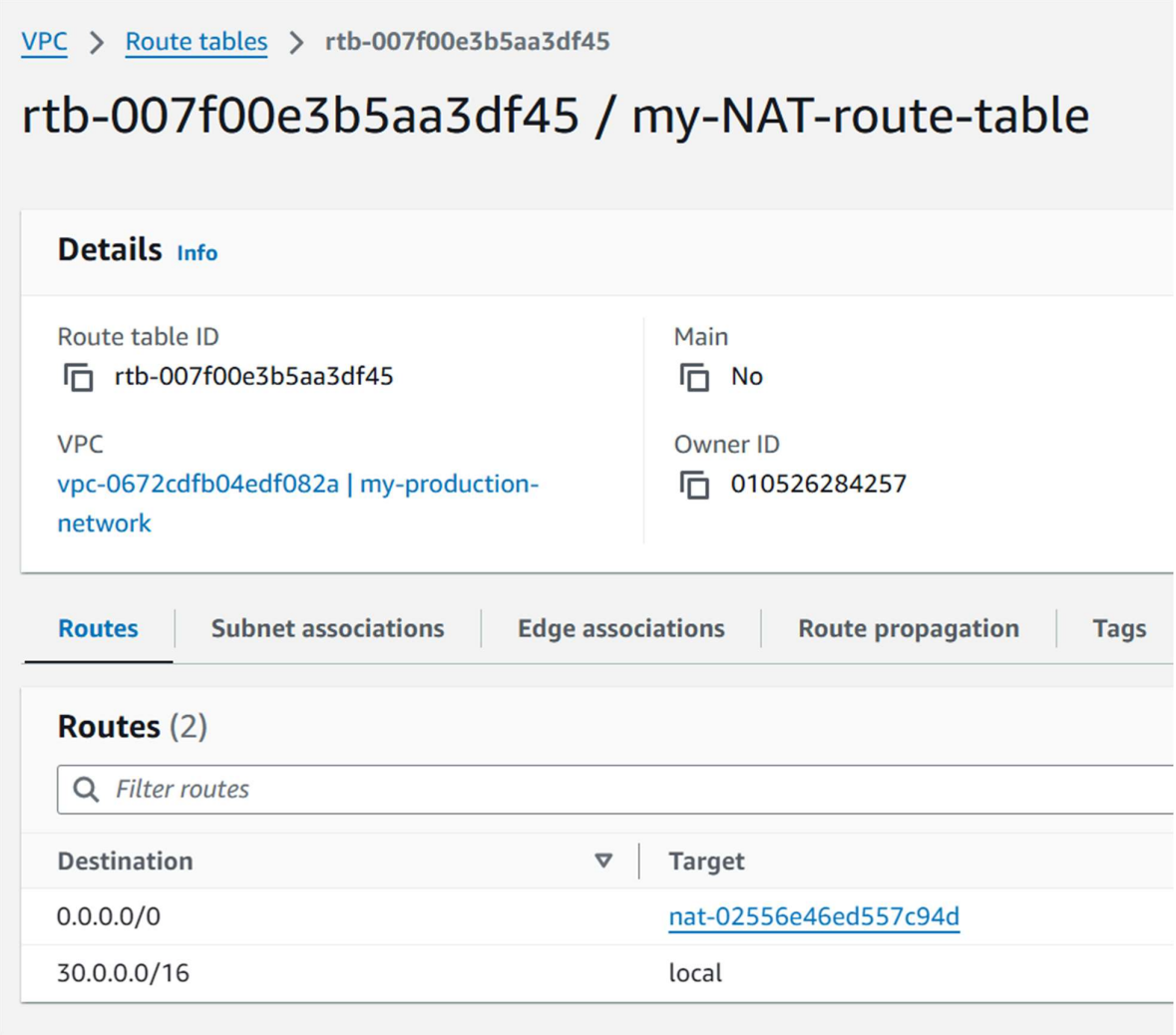
7) We have to create a NAT gateway which will allow private instance and private subnets to send internet request. Under VPC dashboard click NAT gateway and create NAT gateway.



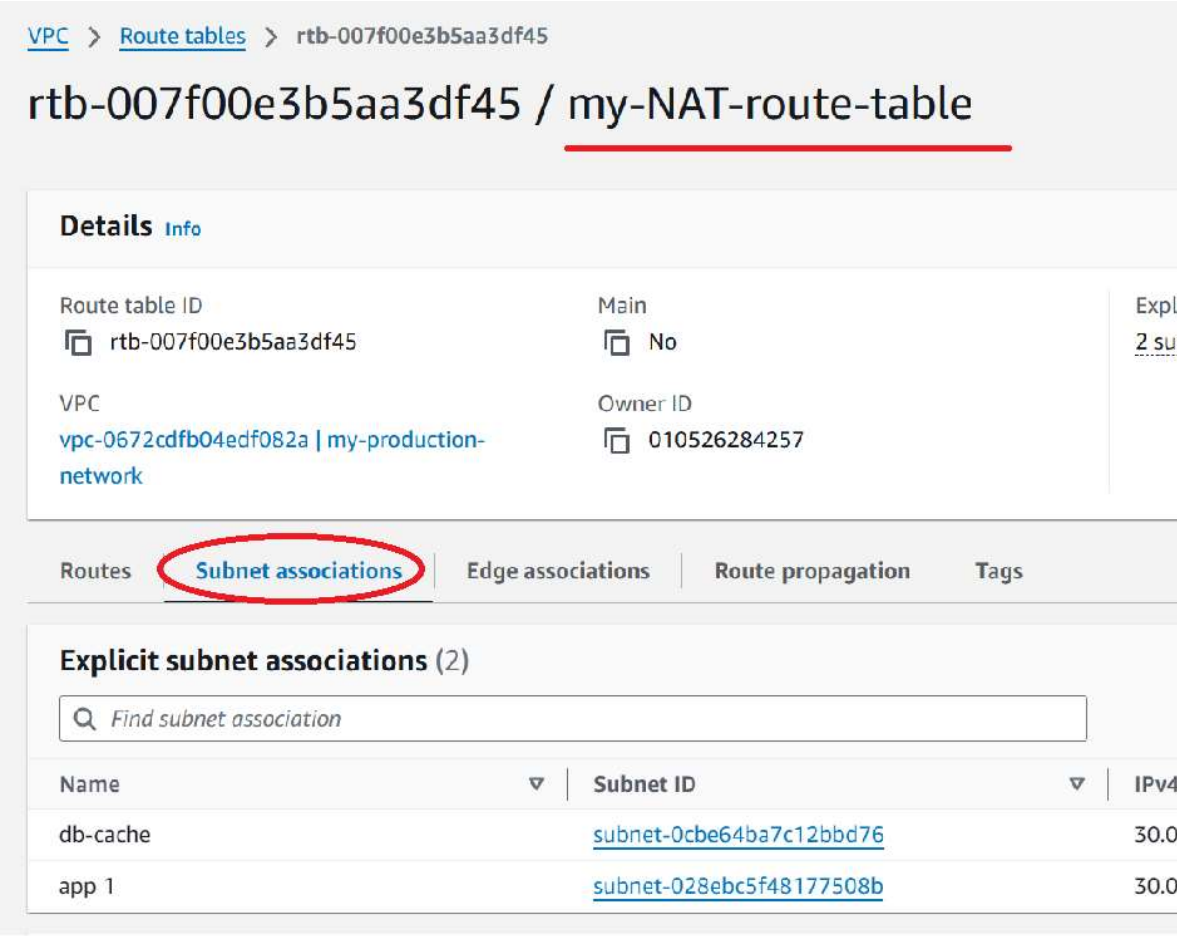
8) Give a name, choose the web Subnet (public)and allocate an elastic IP. Click Create NAT gateway.



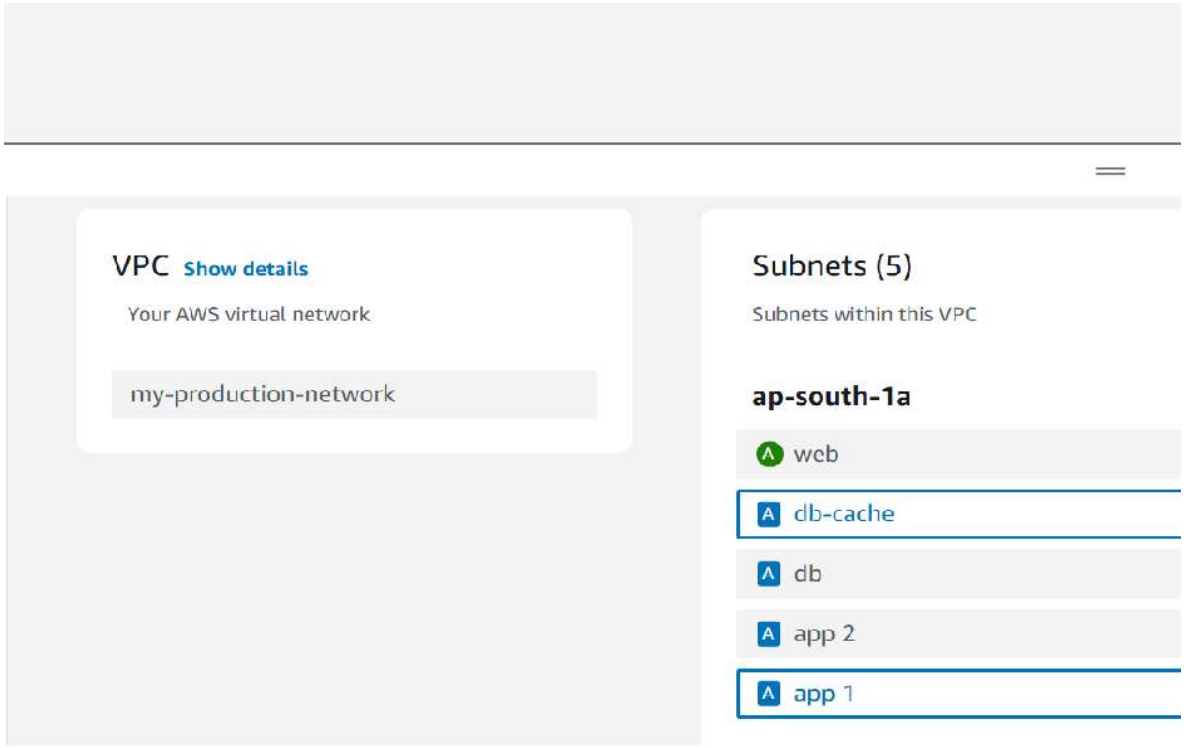
9) Create a new route table which will have internet route access via NAT gateway.



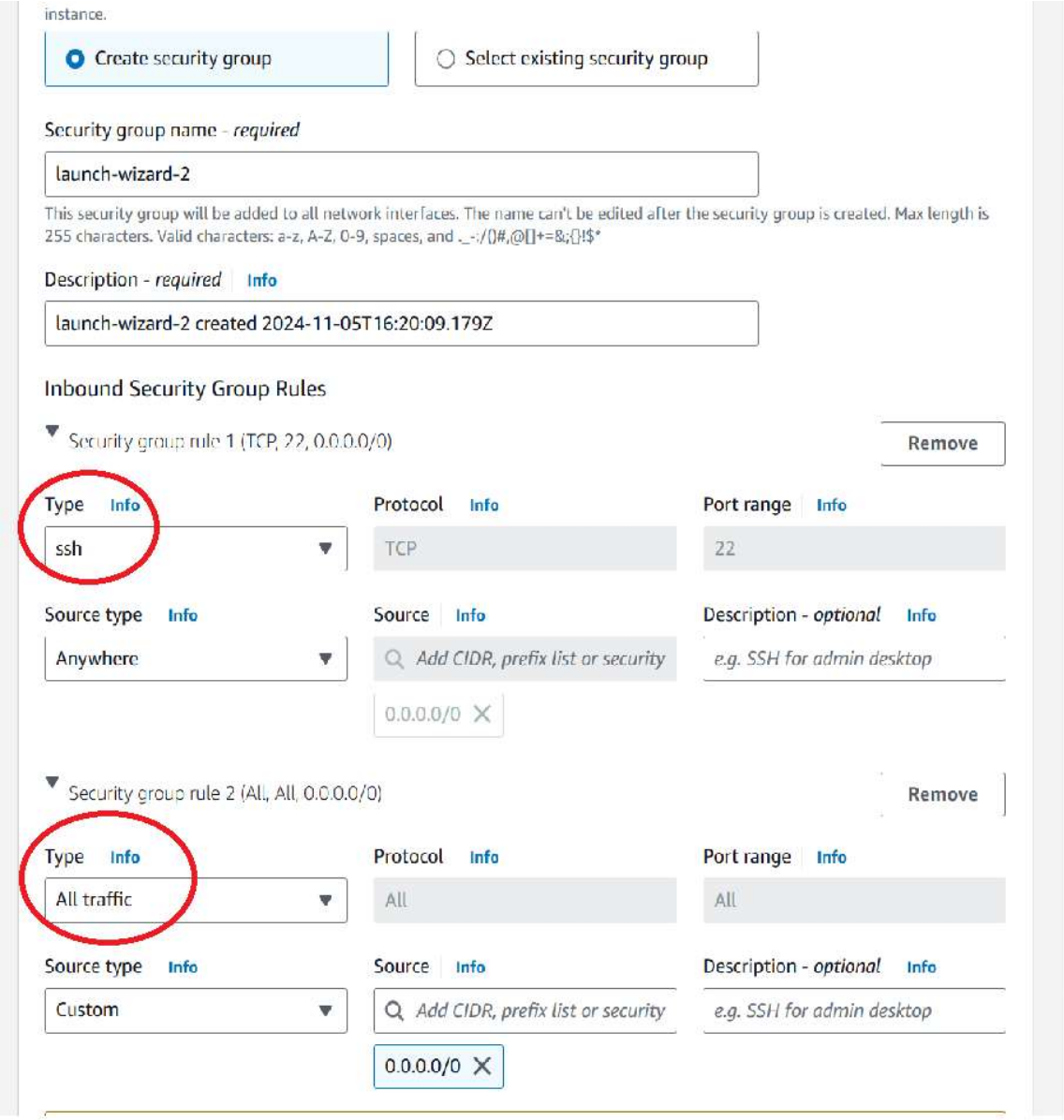
10) Associate my-NAT-route table with app 1 subnet and db-cache subnet.



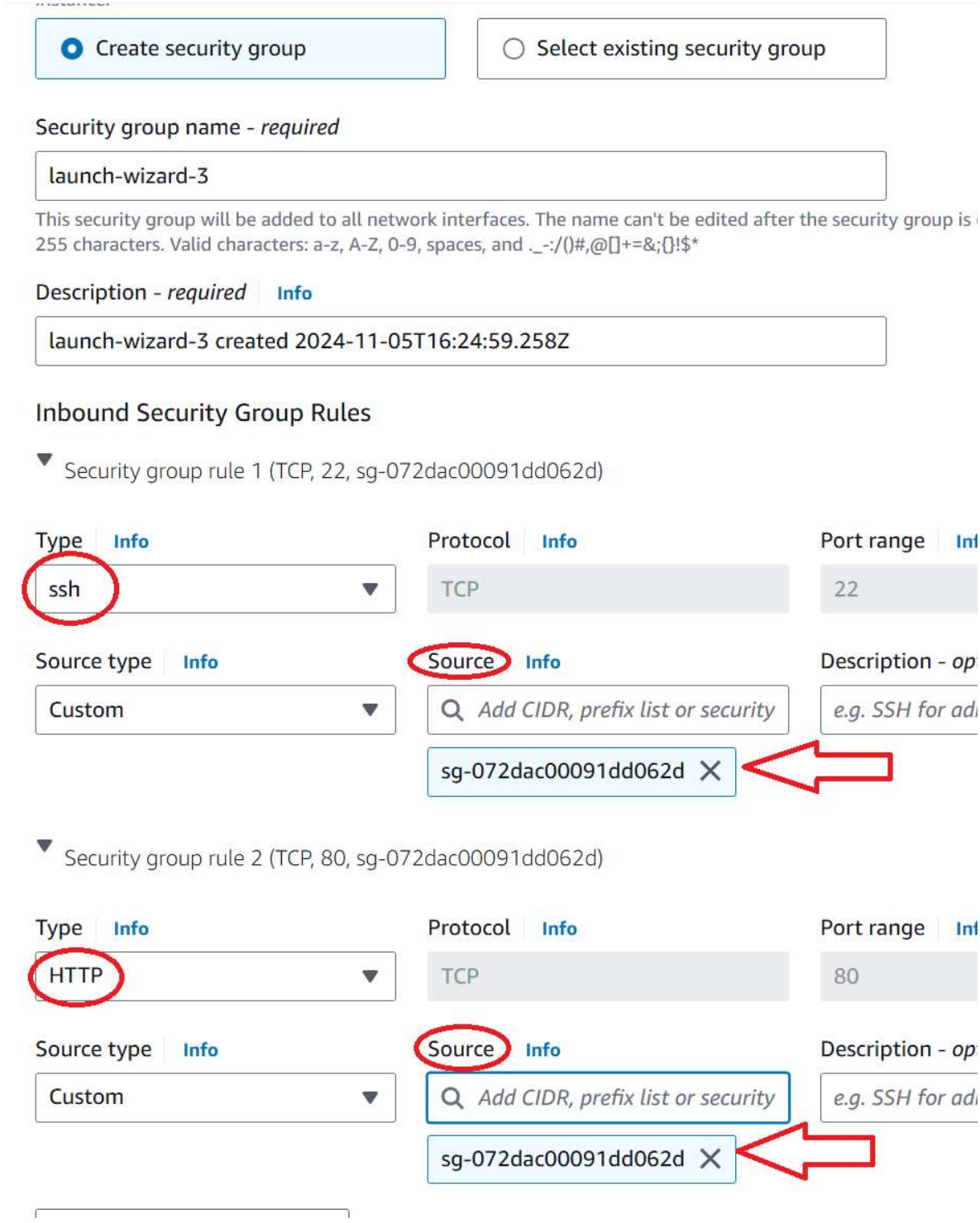
11) My production VPC is created.



12) Now we create instance in public subnet web. In security groups allow incoming rule SSH and all traffic from anywhere.



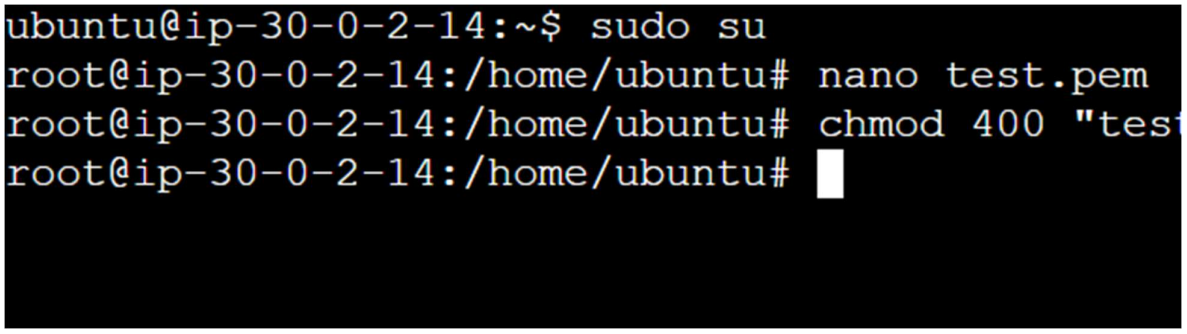
13) We create instances in private subnet with security group incoming rule allowing SSH and HTTP , source:security group of our public instance .



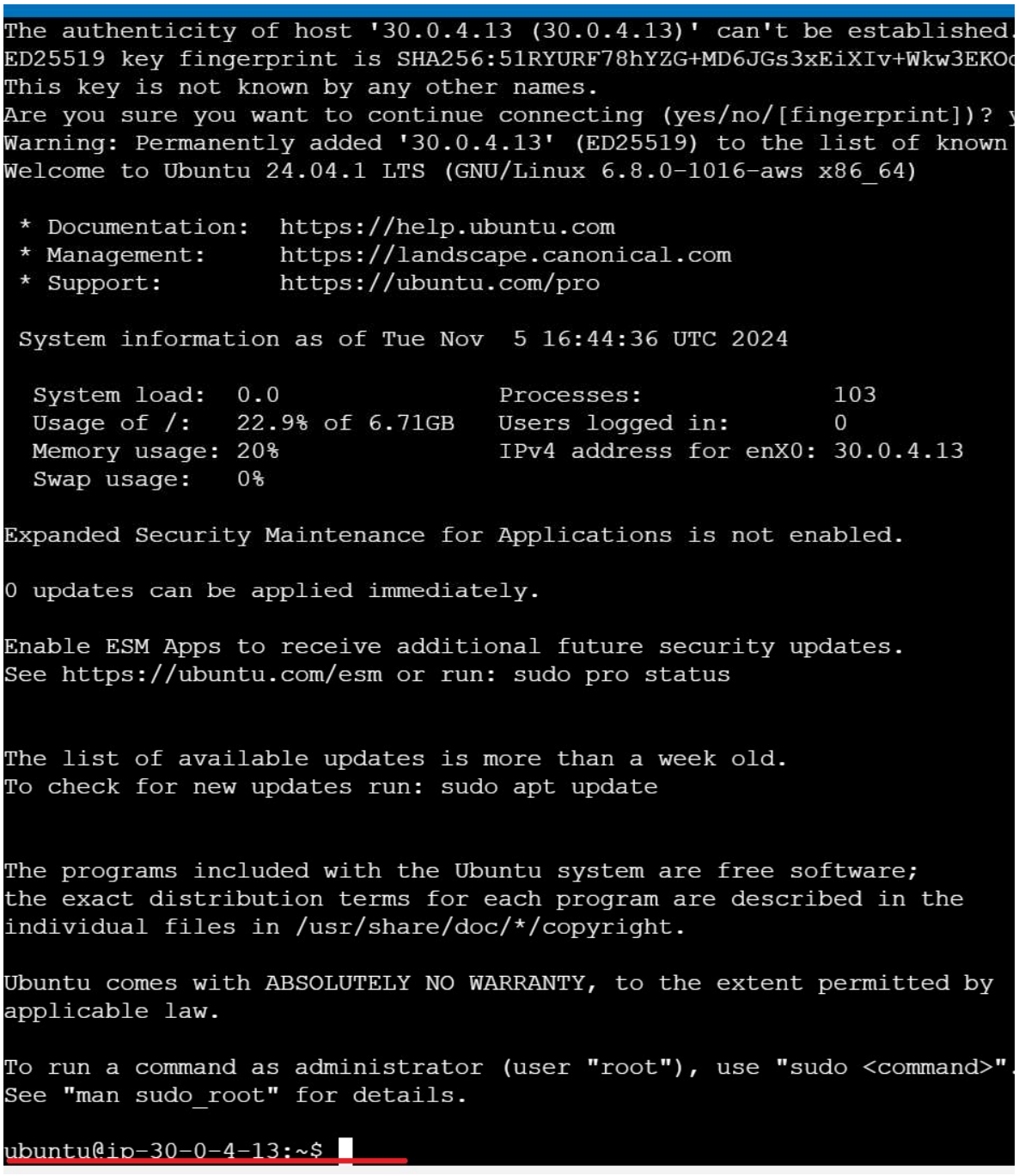
14) Now we will check whether db-cache instance is able to connect to internet via NAT gateway. So first we

connect our web instance which is public. We ping google.com and we can see that it is able to connect via internet gateway.

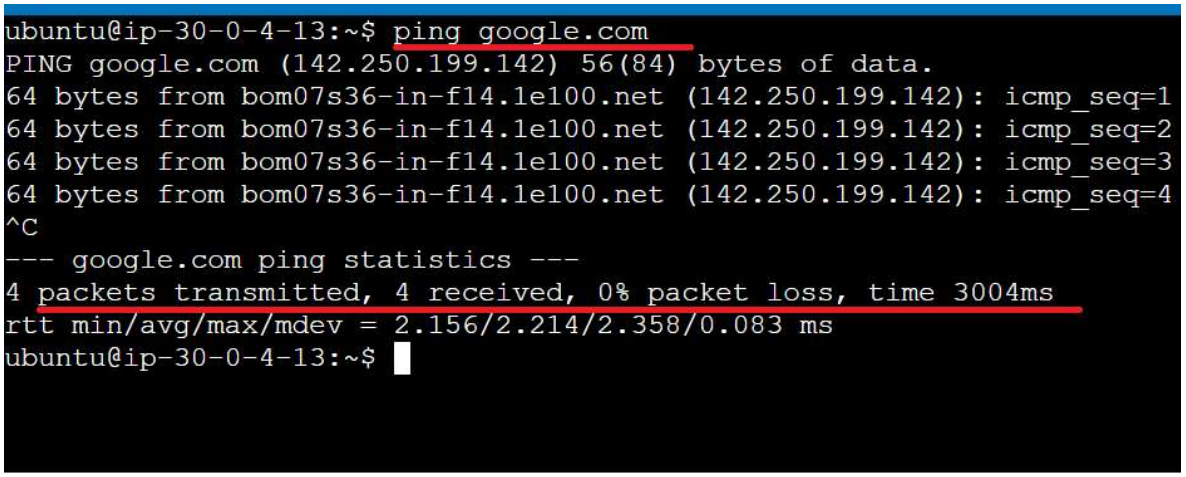
15) We will connect to our private db-cache instance using web instance as a jump server. So we locate our *.pem file , copy it , create a nano file and paste the content of *.pem file. Then change access permissions using chmod command.



16) Now we connect using SSH client and check how the ip change. We are now connected to our db-cache instance.



17) Now we try to connect to internet by pinging google.com and we are able to access internet via NAT gateway.



18) My production network working absolutely fine.

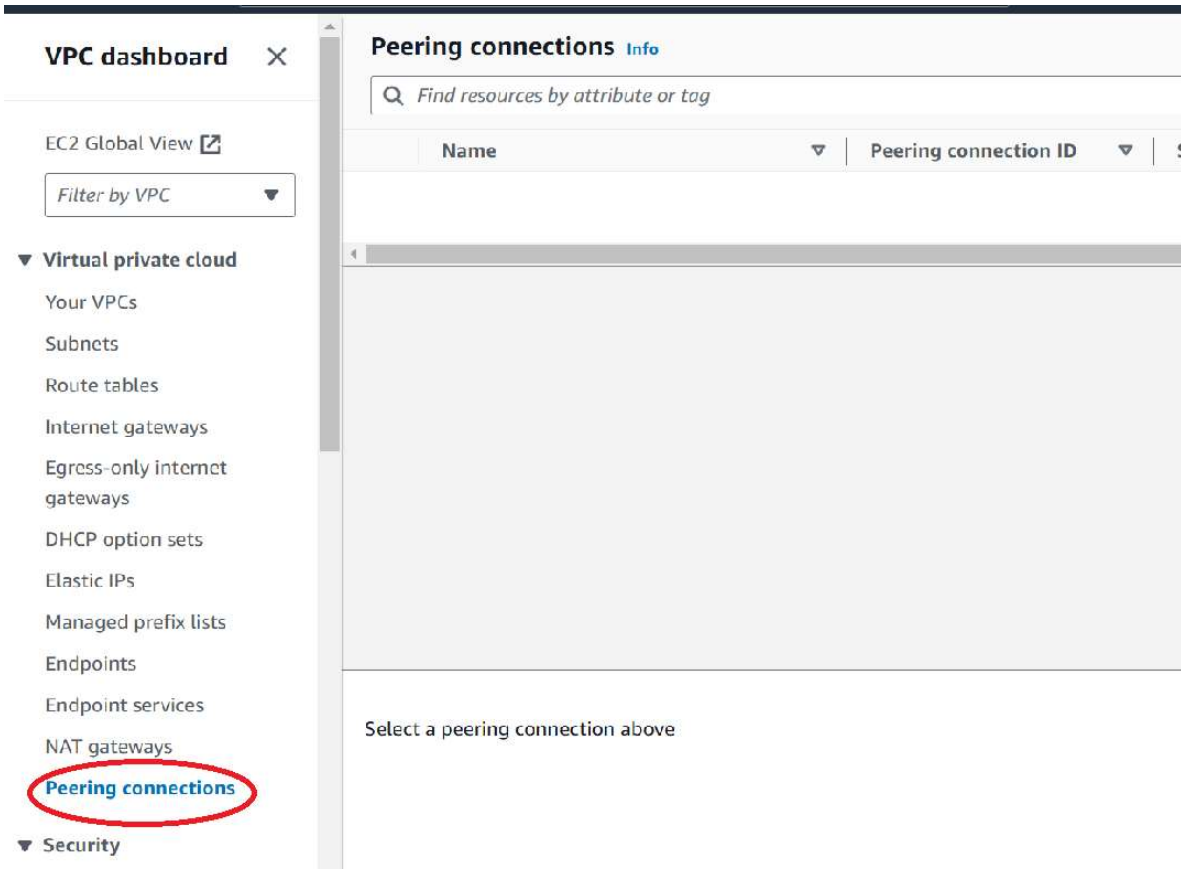
Task to be performed:

Create peering connection between production network and development network.

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

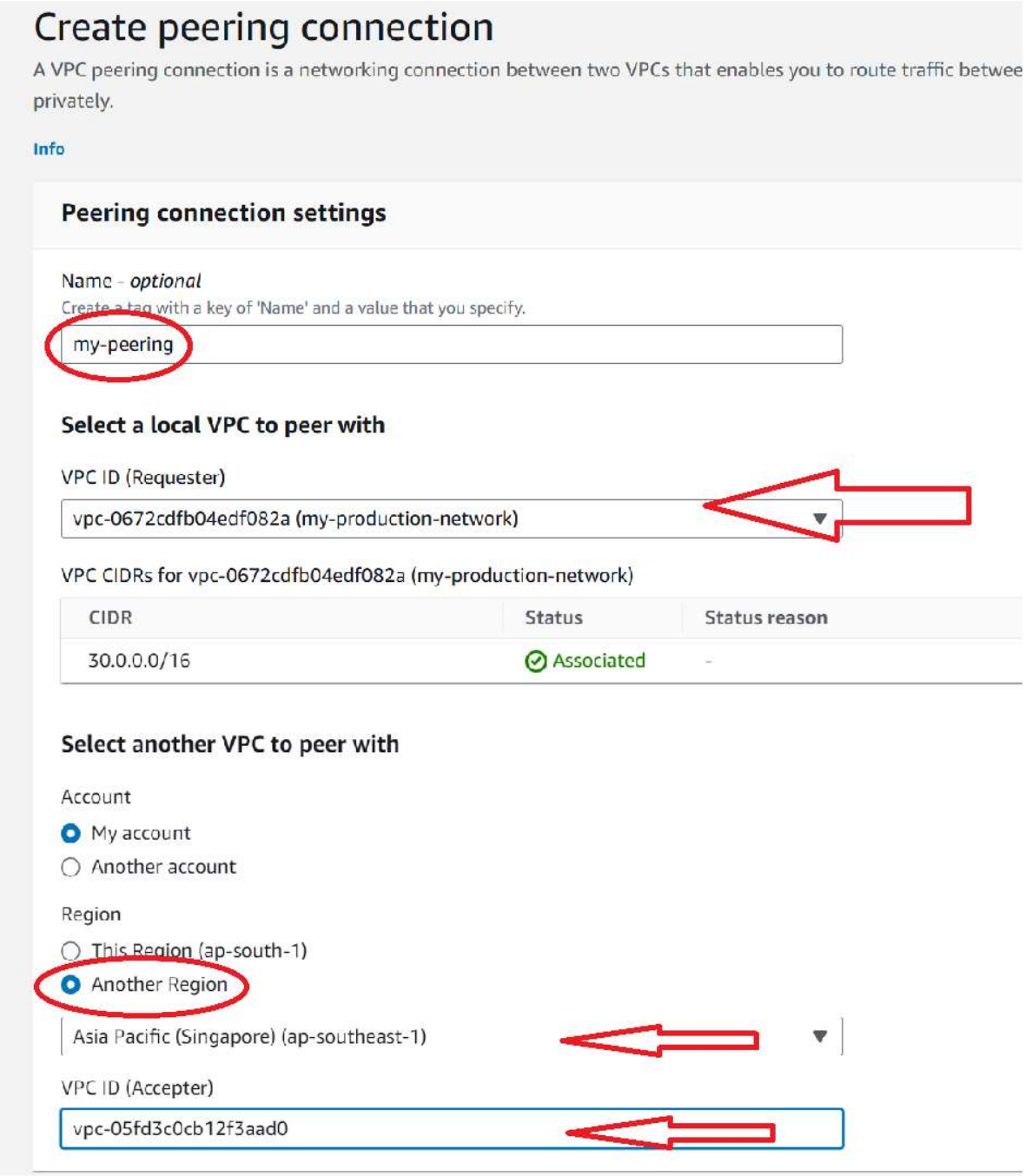
Solution:

1) VPC peering : we can initiate the peering from any network. So lets choose production network. So under VPC dashboard click peering connections and create peering connection.

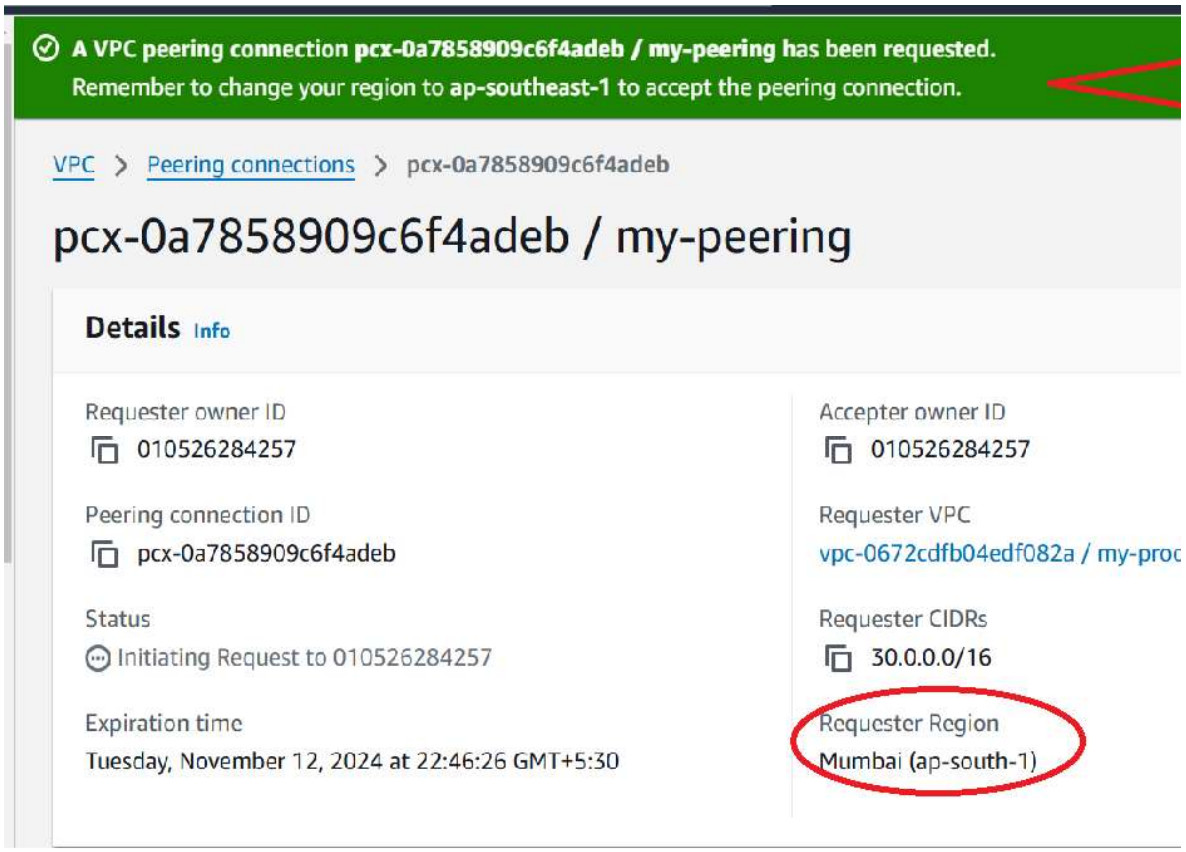


2) Give a name, select the VPC requester which is my production network , under region choose where the development

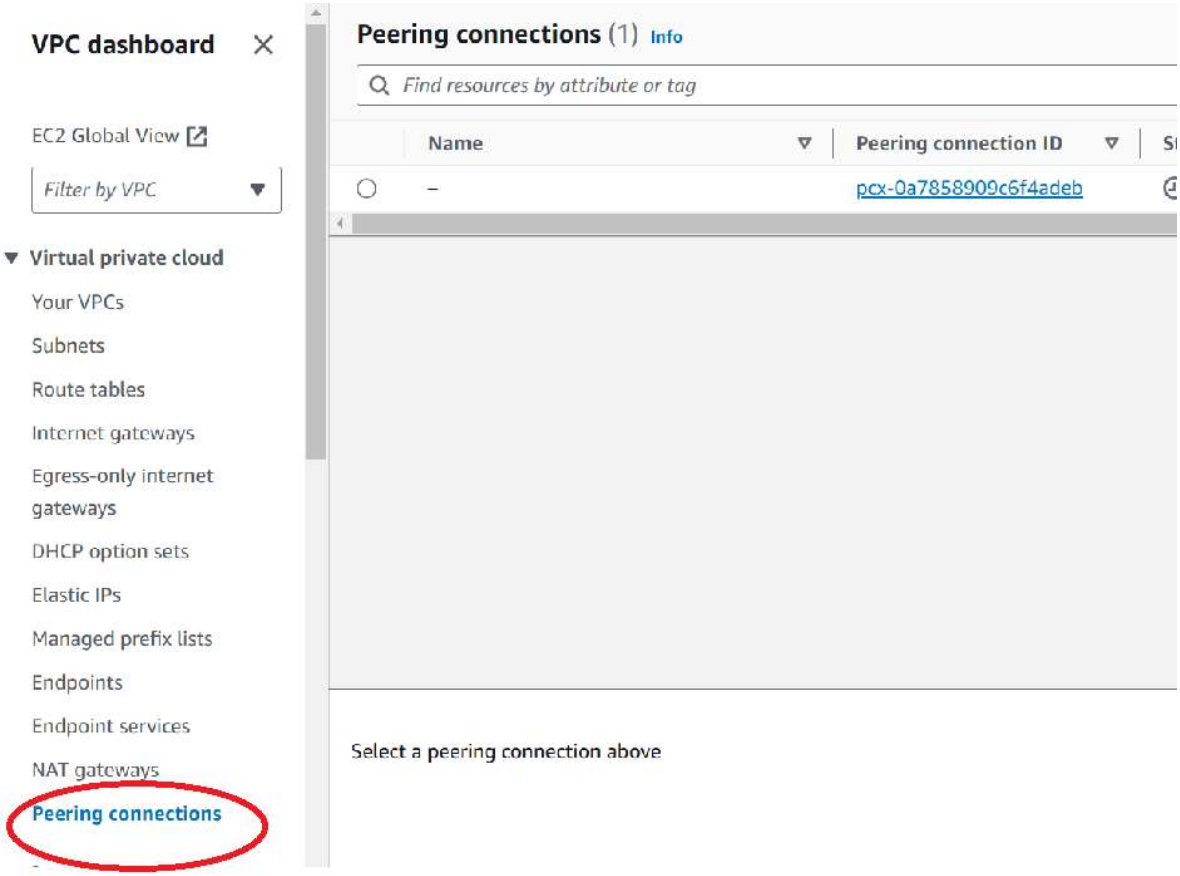
network is . my development network is in Singapore and in VPC acceptor give the VPC ID of the development network and create peering.



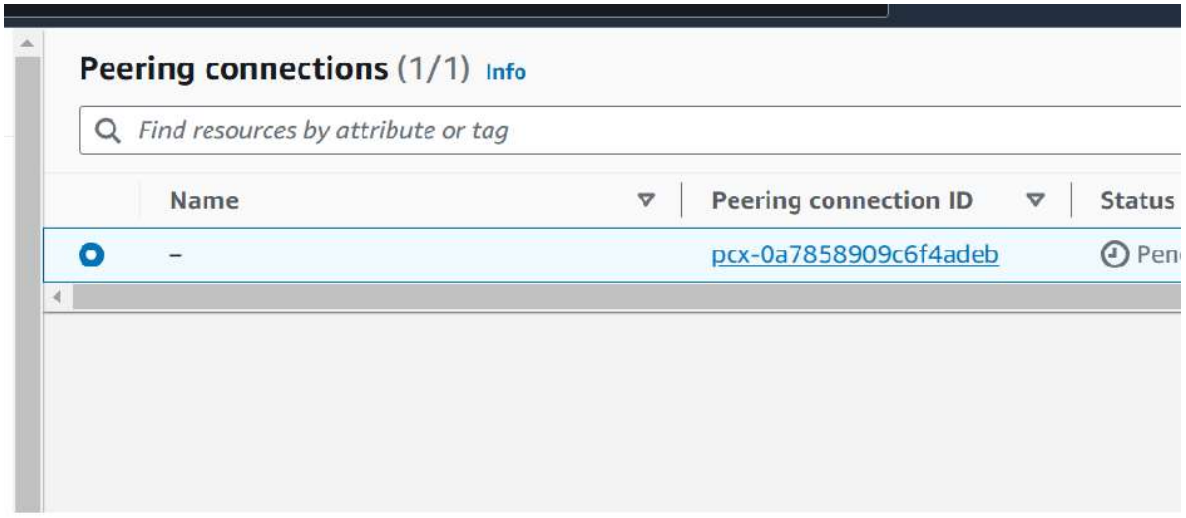
3) So a request is sent to the acceptor region i.e Singapore region.



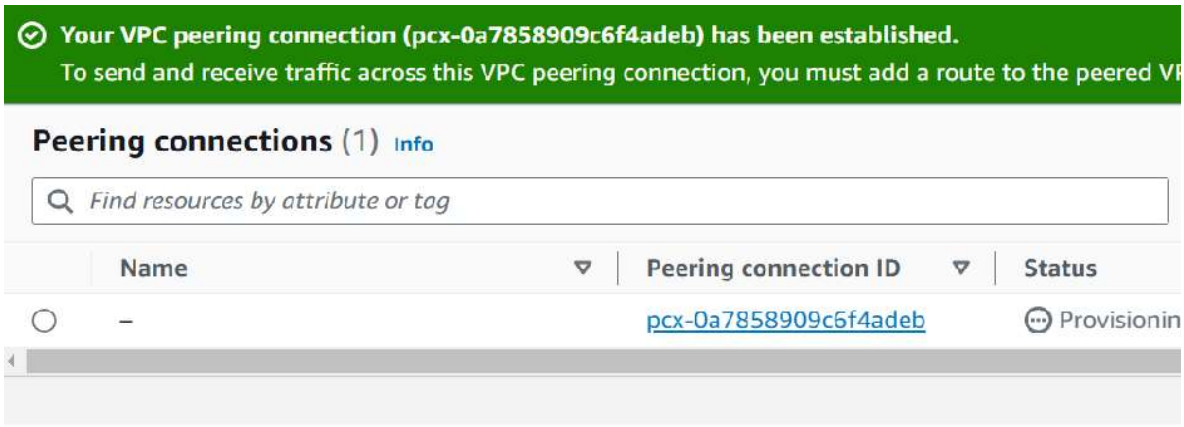
4) When we go to Singapore region and click peering connection we see the pending acceptance notification.



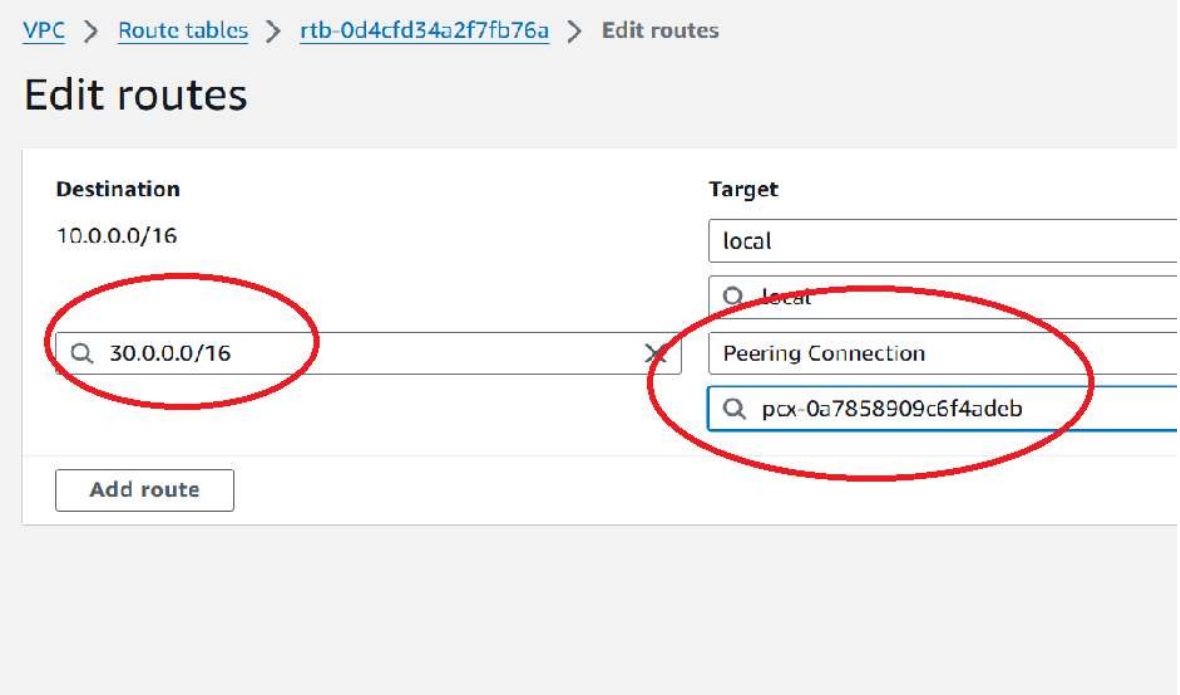
5) Go to actions and accept the request.



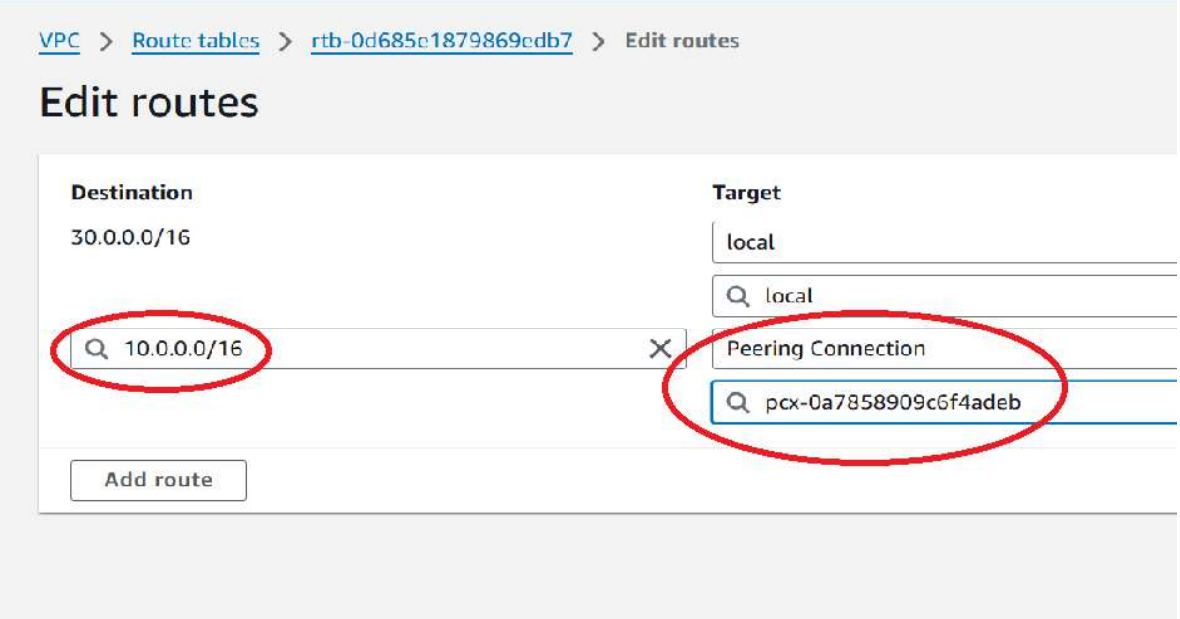
6) Finally VPC peering done.



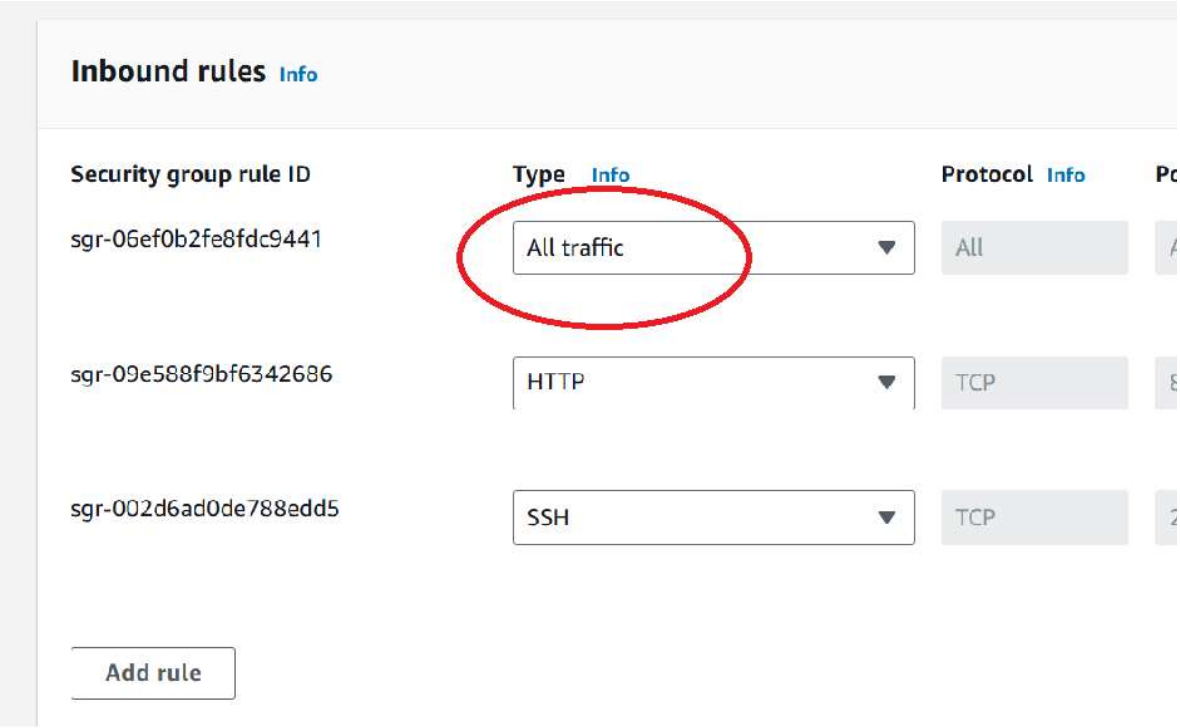
7) To Setup connection between db subnets of both production network and development network we need to modify route table first. So in development network , go to route table and add a route where destination will be the CIDR of production VPC via peering connection.

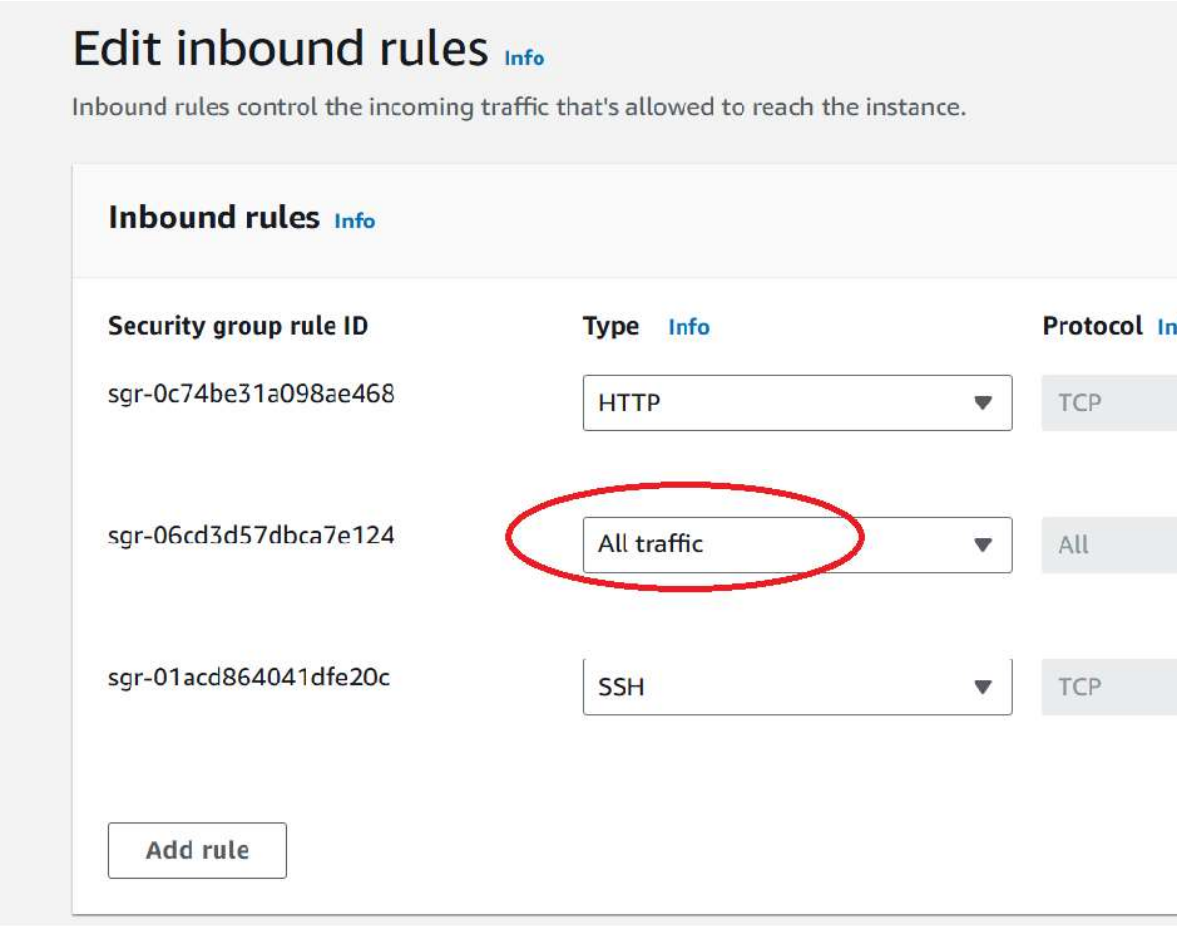


8) In the production network, go to route table and add a route where destination will be the CIDR of development VPC via peering connection.

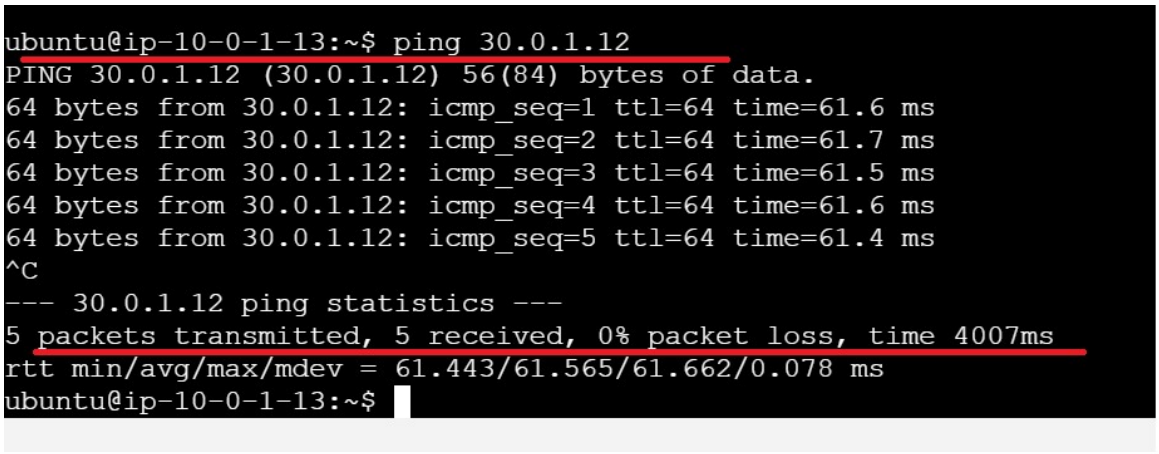


9) Now we modify the security group rules of the instances in db subnets of both the development and production VPC. In the inbound rules of db instance(prod VPC) allow all traffic from source: give the CIDR range of the development VPC and vice versa.





10) Now when we ping the db instance in production VPC from db instance in development VPC we can communicate easily. Ping (the private IP of the instance).



11) So peering connection done successfully.