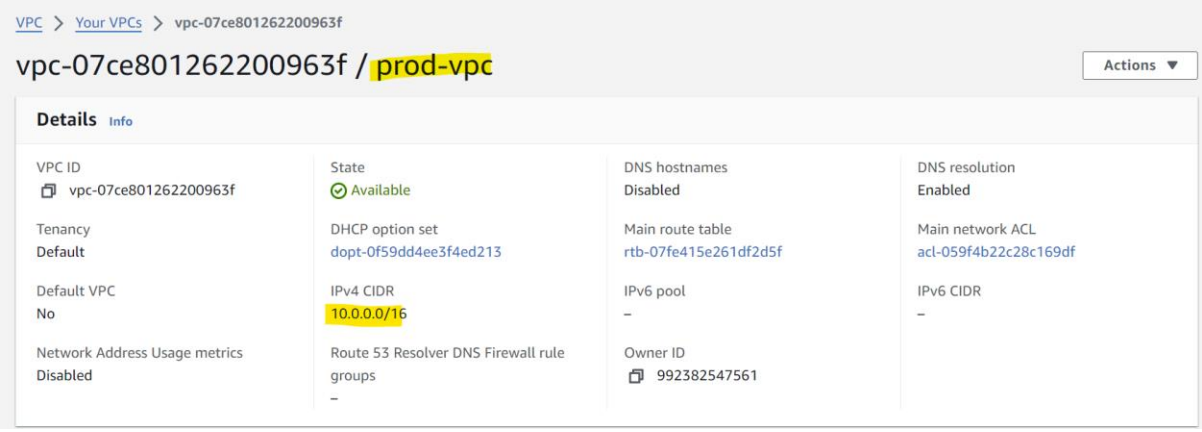
Assignment – VPC

Production Network

1) VPC Creation

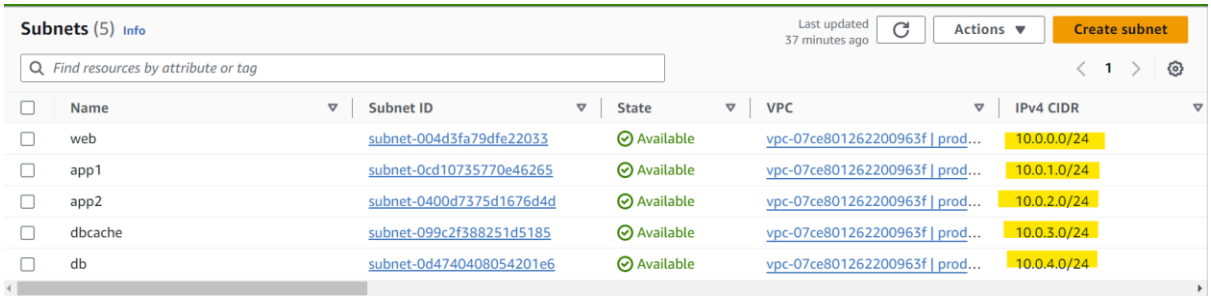
Created VPC 'prod-vpc' with CIDR 10.0.0.0/16



2) Subnet Creation

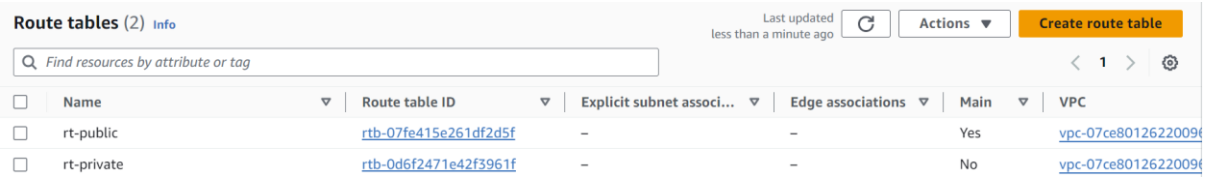
Created 5 subnets web, app1, app2, dbcache and db

CIDRs for web, app1, app2, dbcache and db are 10.0.0.0/24, 10.0.1.0/24, 10.0.2.0/24, 10.0.3.0/24 and 10.0.4.0/24 resp.



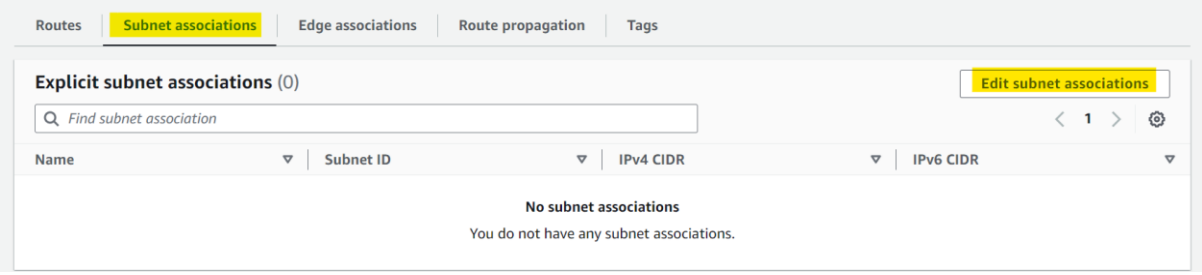
3) Route Table creation

Because there must be 1 public and 4 private subnets, 2 route tables will have to be created. Internet gateway will get added to one of the route tables with which web (public) subnet will also get associated.

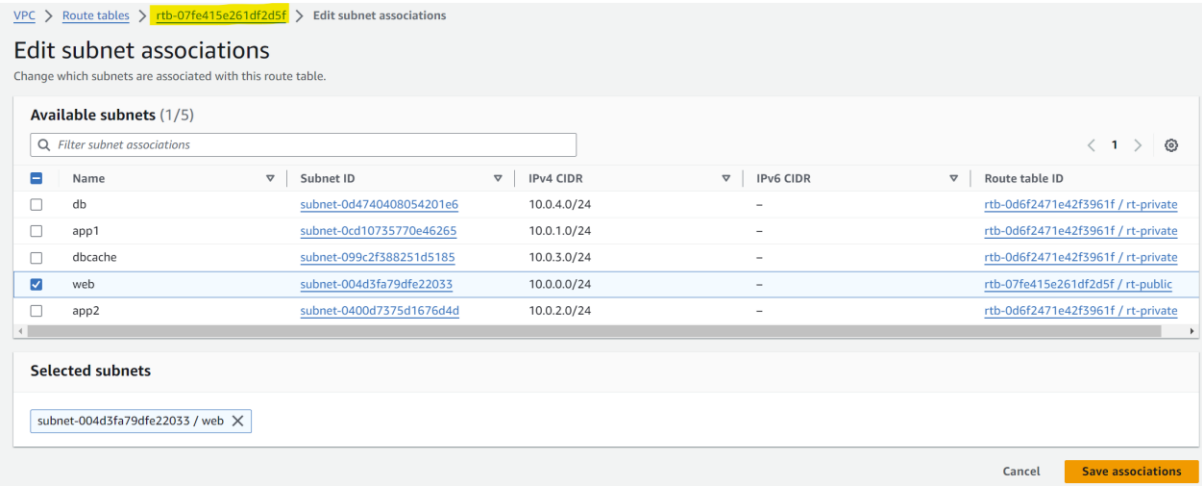


4) Associate subnets with the route table

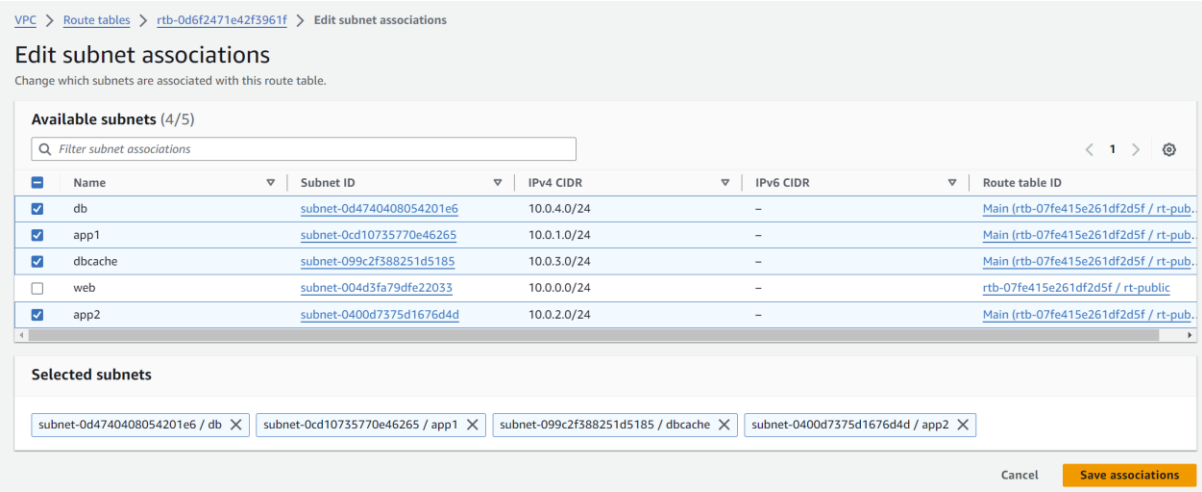
To associate subnets with route table, select the subnet and click the EDIT SUBNET ASSOCIATIONS on the SUBNET ASSOCIATION tab.



Add web subnet to rt-public subnet:



Add app1, app2, dbcache and db subnets to rt-private subnet:

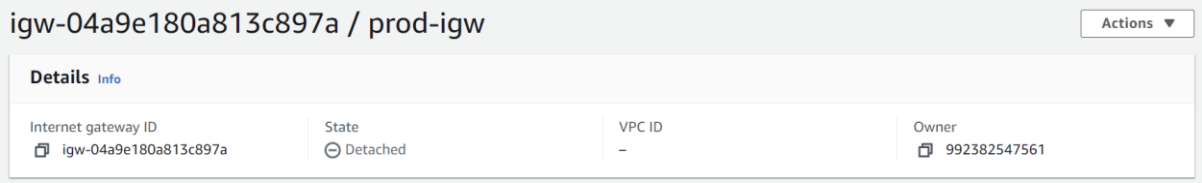


5) Make web subnet public

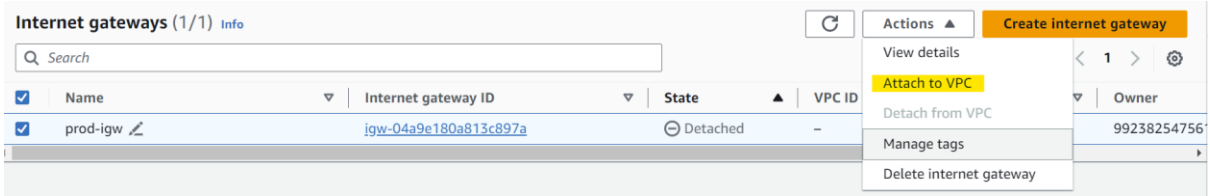
To make a subnet public, it has to be routed to an internet gateway

As we have already added the subnet to the rt-public route table, we will now create an Internet Gateway and add to the same route table to make the web subnet public.

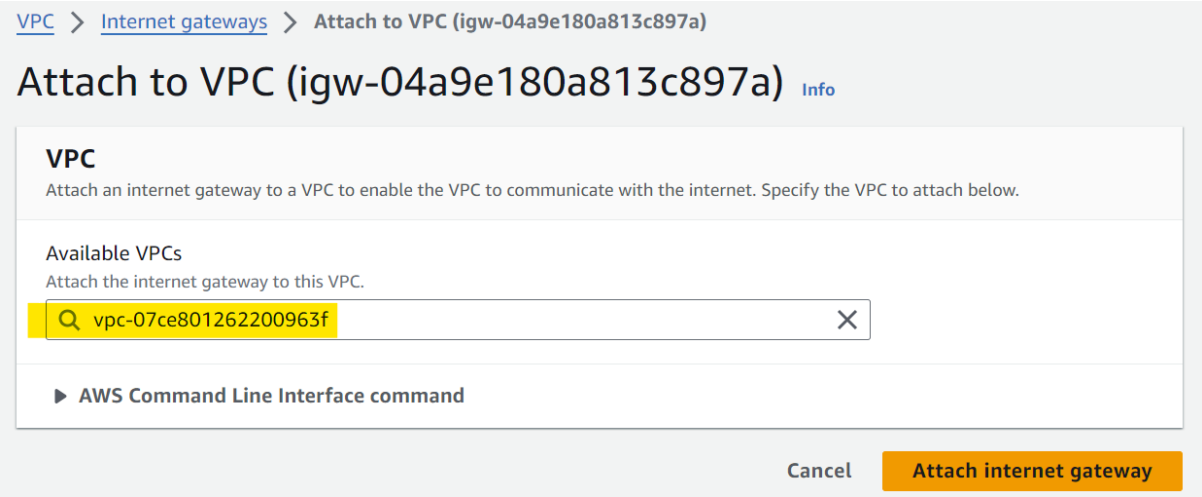
Internet Gateway creation:



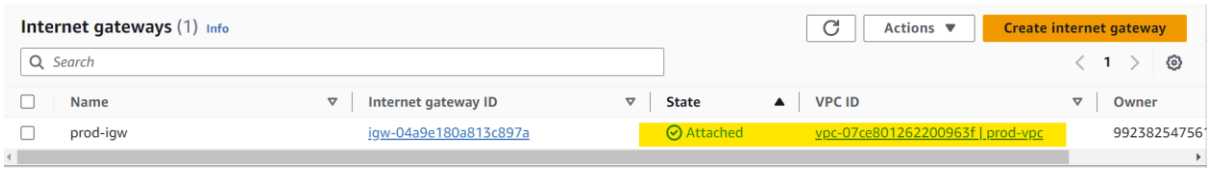
Attach the Internet Gateway to VPC:



Select option to attach VPC



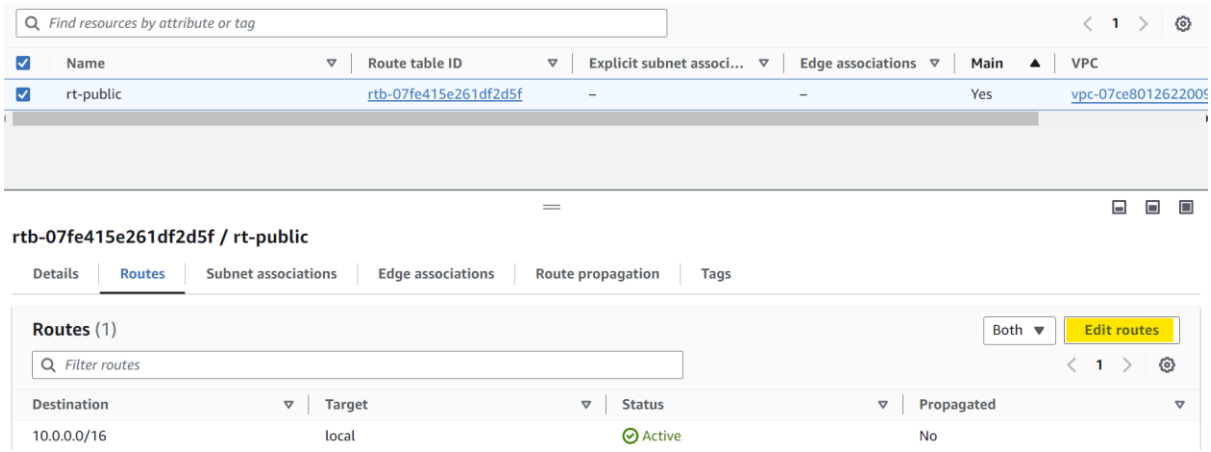
Select a VPC



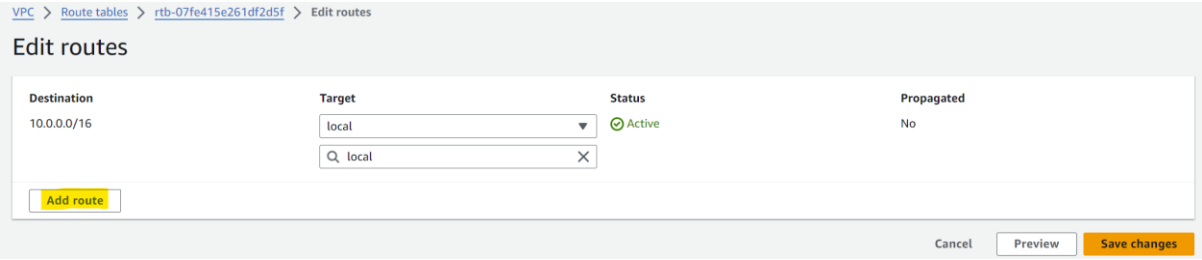
VPC attached

Add Internet Gateway to the Route Table:

Select the Route Table & go to the ROUTES tab. Select EDIT ROUTES.

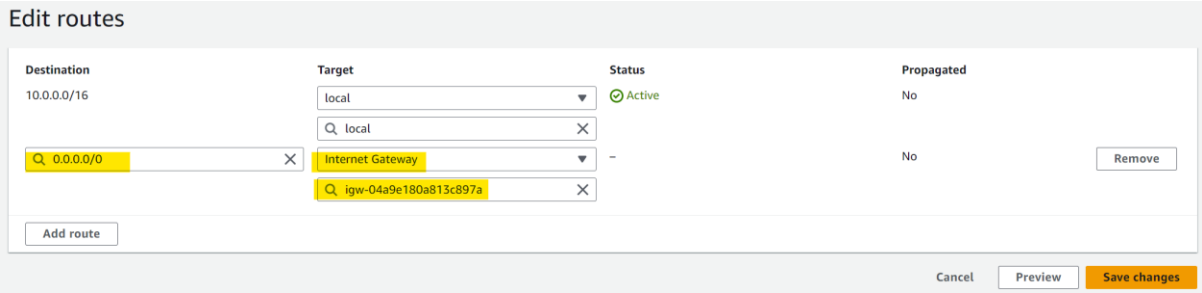


In Edit routes screen, click ADD ROUTE button to add the Internet Gateway

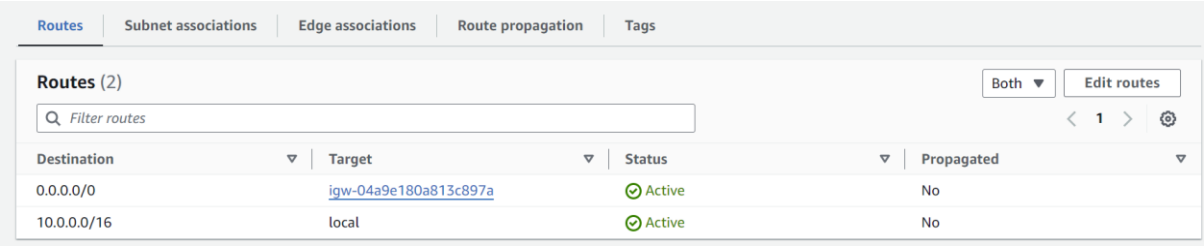


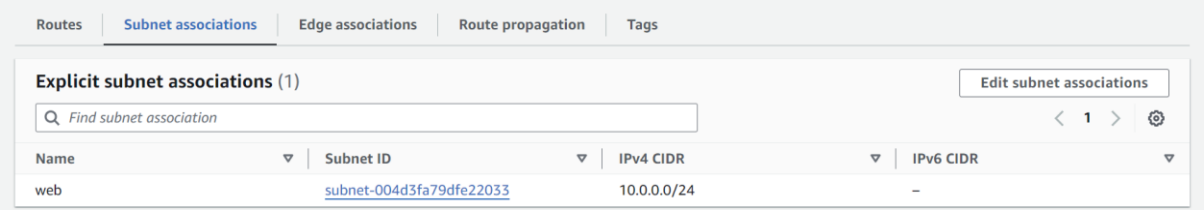
Once a new field to include a new route is added, select the destination CIDR, Target service to route and its ID

In this case, destination CIDR is '0.0.0.0/0' so that the server can communicate to any destination IP. Target routing service is Internet Gateway and the ID is the ID of internet gateway we have created.



After the internet gateway and subnets are added, the association can be verified in the respective tabs under the route table

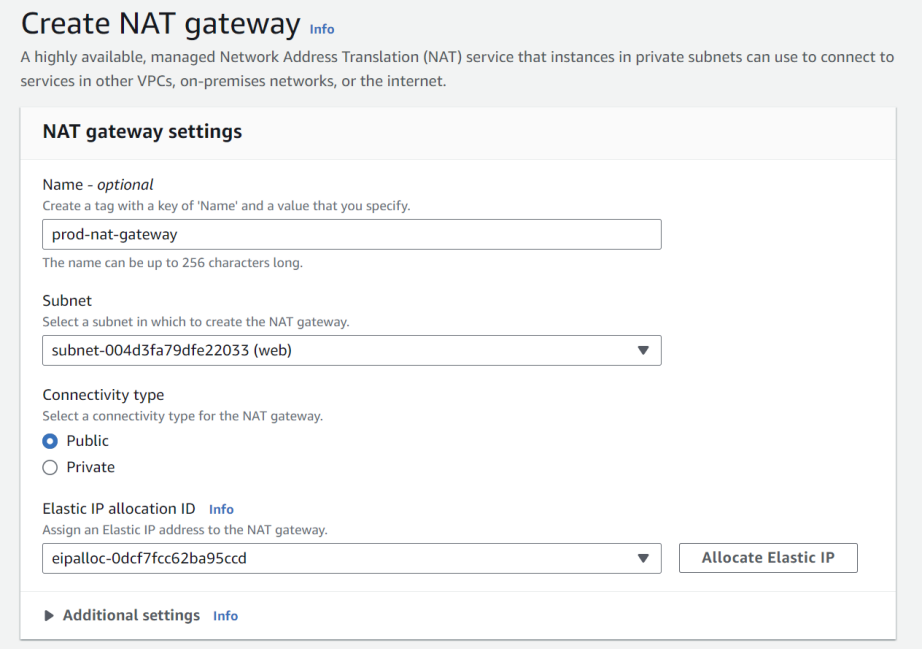




6) Add NAT gateway for private subnets

NAT gateway helps private subnets connect to the services outside VPC while keeping their IP addresses private. All inbound communications are blocked in NAT gateway.

NAT gateway creation:
Go to NAT gateway and click CREATE NAT GATEWAY



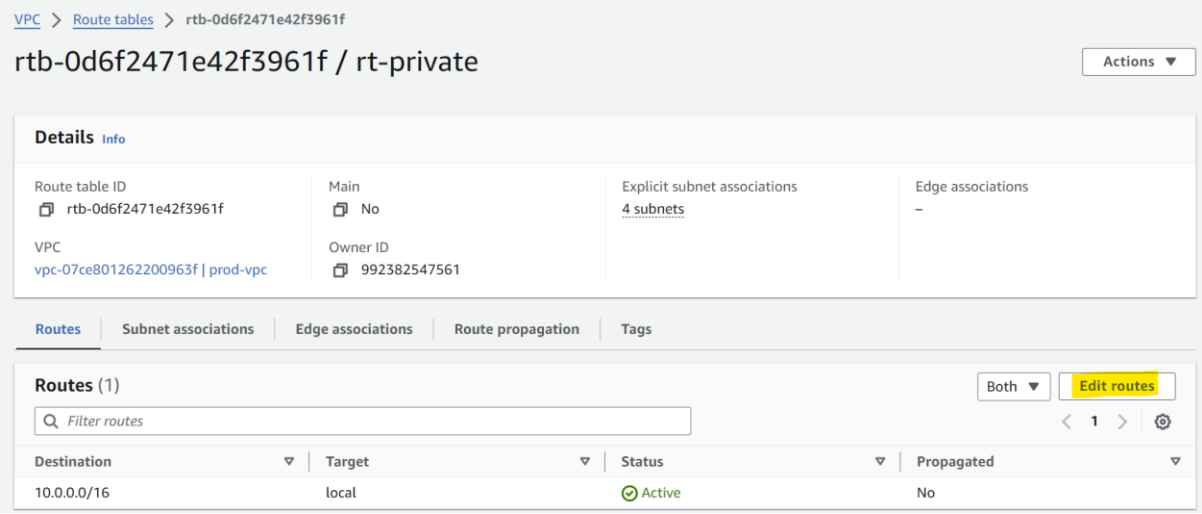
Select the VPC under which the NAT gateway must be created.

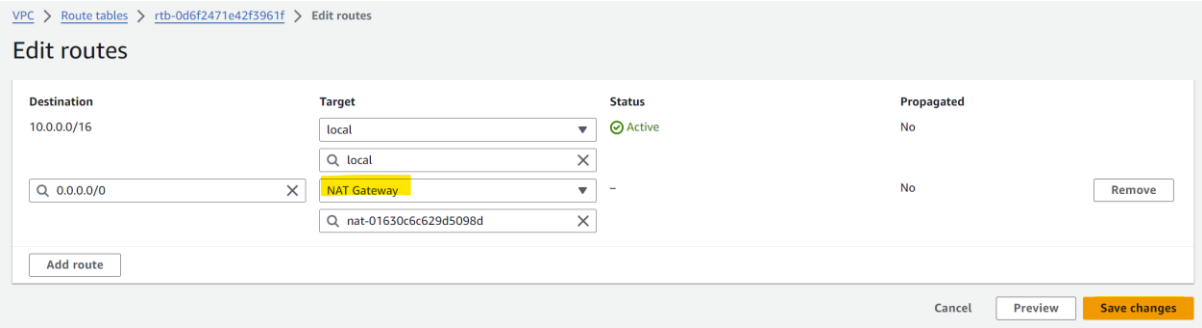
Select the public subnet because a NAT gateway will always be inside a public subnet.

Click ALLOCATE ELASTIC IP and finally click create NAT gateway

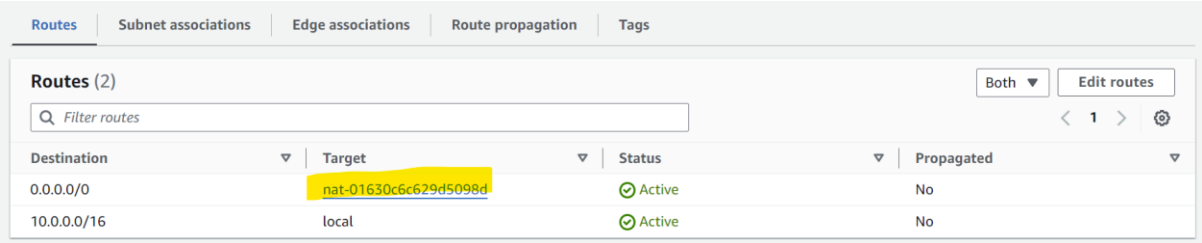
Add NAT gateway to route table:

Now the NAT gateway has to be added to route table that is associated with private subnets



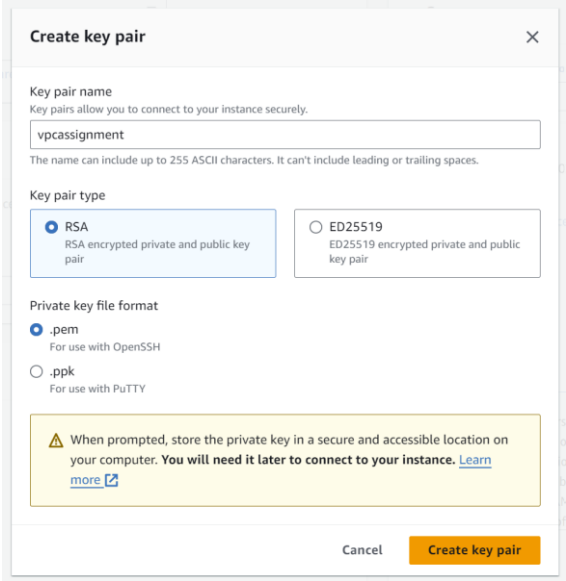


After the NAT gateway is added to the route table, we can find the link in the tab as show below



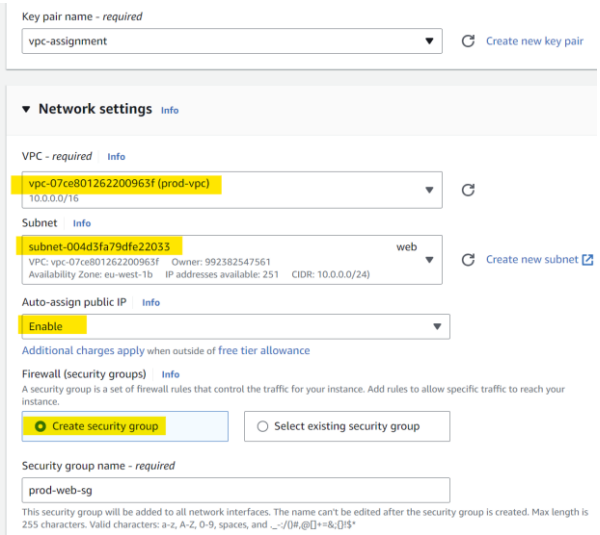
7) EC2 instances creation:

Create a key-pair:

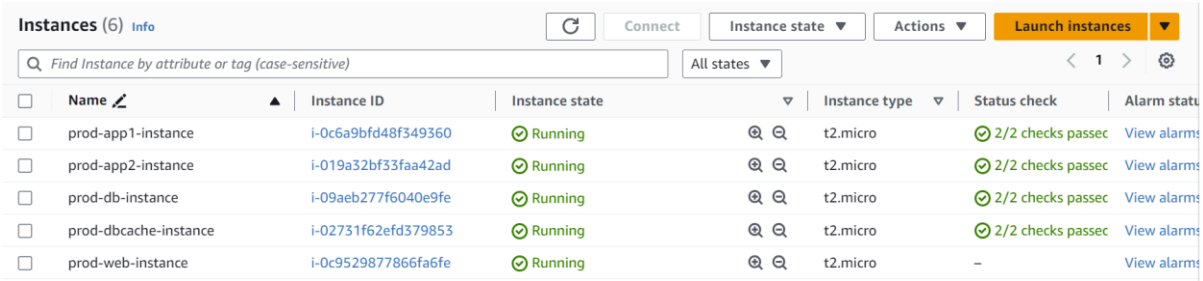


Web server instance creation:

Select the key-pair, VPC and subnet (web). As this server must communicate with services outside VPC, a public IP will be assigned. We will simultaneously create a new security group for all EC2 instances.



As the other 4 servers will be private, public IP won't be created for them keeping the rest of the steps same. The final EC2 instances will look like below



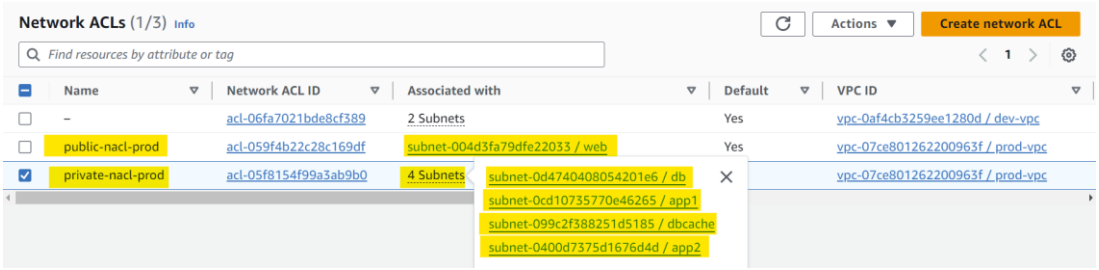
8) Configure NACL and security groups for security

NACL creation:

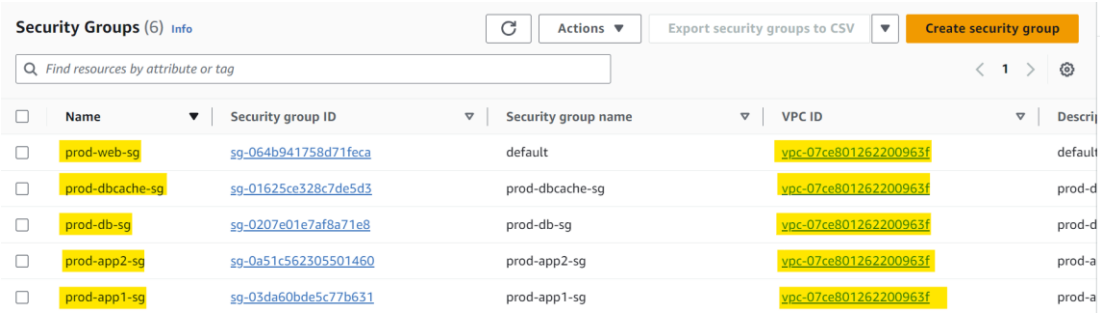
We will need 2 NACLs for private and public subnets.

A default NACL is automatically created for every VPC. We can use the same NACL out of the two

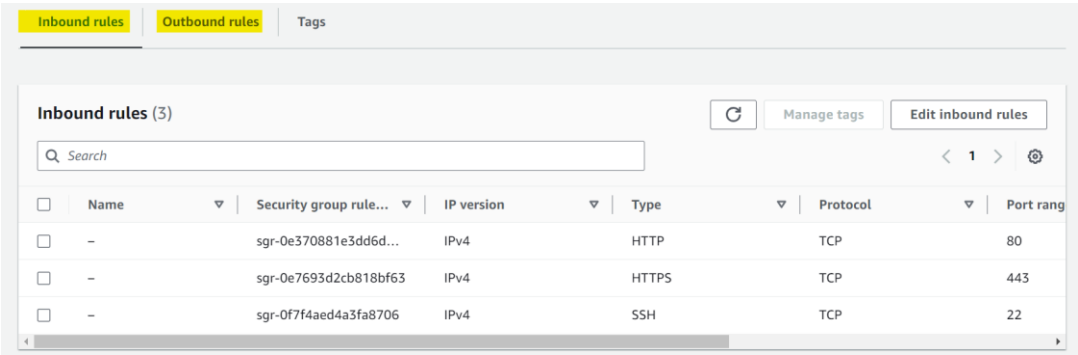
Renamed default NACL for including public subnet and created a NACL for the private subnets



Security Group creation:



Declare inbound and outbound rules of NACL and Security Groups:



Development Network

1) VPC Creation

VPC > Your VPCs > vpc-0af4cb3259ee1280d

vpc-0af4cb3259ee1280d / dev-vpc

Actions

Details Info

VPC ID vpc-0af4cb3259ee1280d	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-0f59dd4ee3f4ed213	Main route table rtb-0d9bce25cbbf50cc0 / rt-public-dev	Main network ACL acl-06fa7021bde8cf389
Default VPC No	IPv4 CIDR 107.12.0.0/16	IPv6 pool –	IPv6 CIDR –
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 992382547561	

2) Subnet Creation

Subnets (1/7) Info

Last updated 3 minutes ago | Actions | Create subnet

Find resources by attribute or tag

Name	Subnet ID	State	VPC	IPv4 CIDR
db-dev	subnet-0874576f3538f52f6	Available	vpc-0af4cb3259ee1280d | dev-...	107.12.1.0/24
web-dev	subnet-02d2a4001d5346842	Available	vpc-0af4cb3259ee1280d | dev-...	107.12.0.0/24

3) Route Table creation

Because there must be 1 public and 1 private subnet, 2 route tables will have to be created. Internet gateway will get added to one of the route tables with which web (public) subnet will also get associated.

Route tables (4) Info

Last updated about 2 hours ago | Actions | Create route table

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
rt-public-dev	rtb-0d9bce25cbbf50cc0	subnet-02d2a4001d5346...	–	Yes	vpc-0af4cb3259ee128
rt-private-dev	rtb-0d8d3fde1758584a0	subnet-0874576f3538f5...	–	No	vpc-0af4cb3259ee128

4) Associate subnets with the route table

Steps to associate subnets with route table is same as mentioned production setup.

Below is a screenshot after the association is done

Route tables (4) Info

Last updated about 2 hours ago | Actions | Create route table

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associations	Edge associations	Main	V
rt-public-dev	rtb-0d9bce25cbbf50cc0	subnet-02d2a4001d5346842 / web-dev	–	Yes	v
rt-private-dev	rtb-0d8d3fde1758584a0	subnet-0874576f3538f52f6 / db-dev	–	No	v
rt-private	rtb-0d6f2471e42f3961f	4 subnets	–	No	v

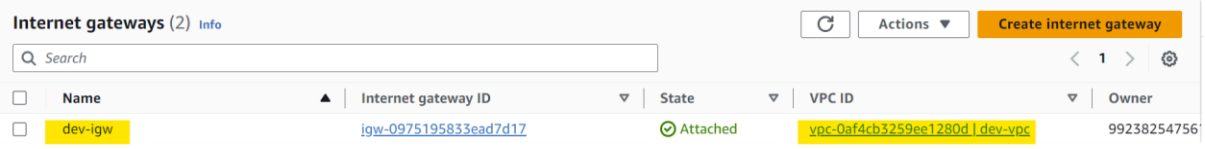
5) Make web subnet public

To make a subnet public, it has to be routed to an internet gateway

As we have already added the subnet to the rt-public-dev route table, we will now create an Internet Gateway and add to the same route table to make the web subnet public.

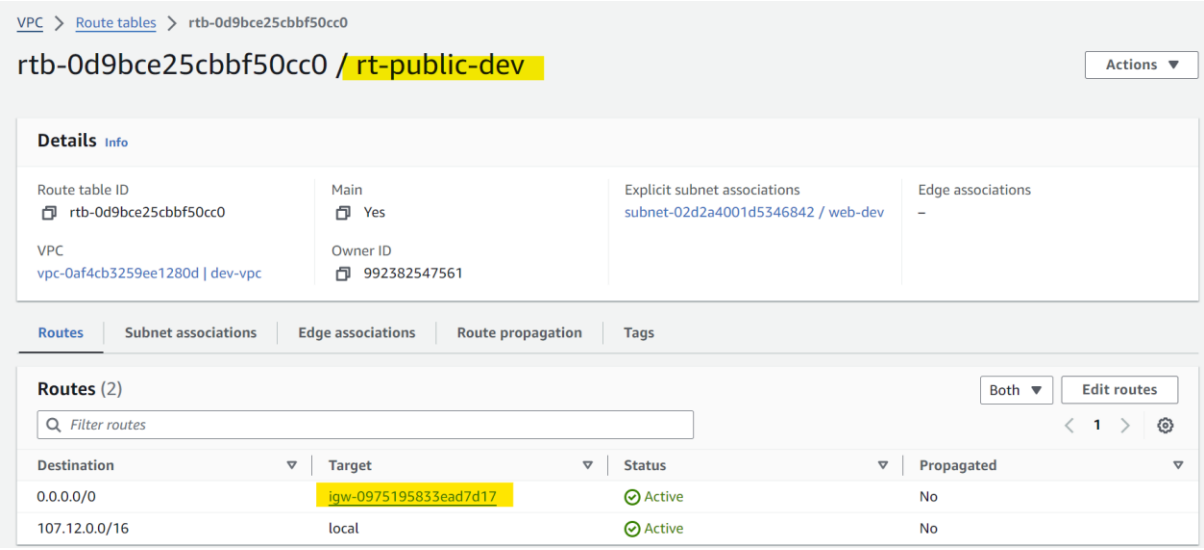
Internet Gateway creation:

Attach the Internet Gateway to VPC:



VPC attached

Add Internet Gateway to the Route Table:

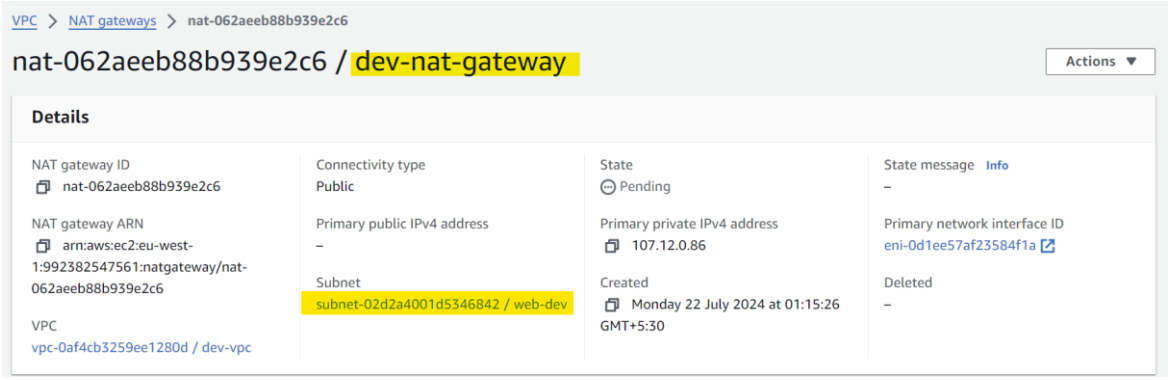


Add subnet to route table:

6) Add NAT gateway for private subnets

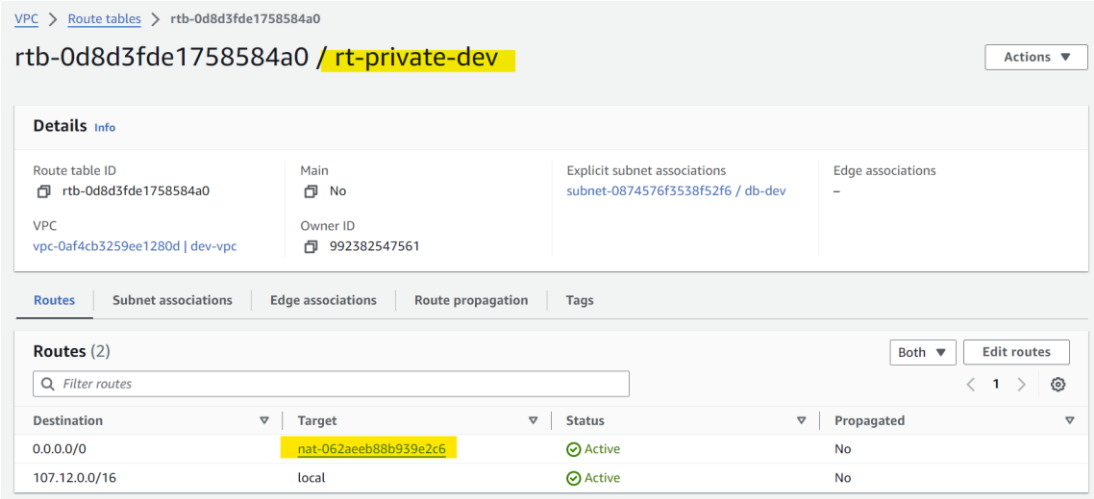
NAT gateway helps private subnets connect to the services outside VPC while keeping their IP addresses private. All inbound communications are blocked in NAT gateway.

NAT gateway creation:



Add NAT gateway to route table:

Now the NAT gateway has to be added to route table that is associated with private subnets



7) EC2 instances creation:

We can use the same key-pair for development environment.

Like production, web instance in development will also have a public IP and db instance won't as it is private.

The instance creation steps will be the same.

After creation the instances will like below

Instances (8) Info

Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm statu
dev-db-instance	i-0cb09faeb9bf5859b	Running	t2.micro	2/2 checks passec	View alarms
dev-web-instance	i-04435c07e0c65f88a	Running	t2.micro	Initializing	View alarms

8) Configure NACL and security groups for security

NACL creation:

Like production, here too we will need 2 NACLs for private and public subnets.

Network ACLs (4) Info

Actions | Create network ACL

Find resources by attribute or tag

Name	Network ACL ID	Associated with	Default	VPC ID
public-nacl-dev	acl-06fa7021bde8cf389	subnet-02d2a4001d5346842 / web-dev	Yes	vpc-0af4cb3259ee1280d / dev-vpc
private-nacl-dev	acl-0ceb0f56826a41012	subnet-0874576f3538f52f6 / db-dev	No	vpc-0af4cb3259ee1280d / dev-vpc

Security Group creation:

Security Groups (8) Info

Actions | Export security groups to CSV | Create security group

Find resources by attribute or tag

Name	Security group ID	Security group name	VPC ID	Descri
dev-db-sg	sg-0dbcdac9b337dab9d	default	vpc-0af4cb3259ee1280d	default
dev-web-sg	sg-08f517bf5fa206a11	dev-web-sg	vpc-0af4cb3259ee1280d	dev-we

Declare inbound and outbound rules of NACL and Security Groups:

Inbound rules | Outbound rules | Tags

Inbound rules (3)

Manage tags | Edit inbound rules

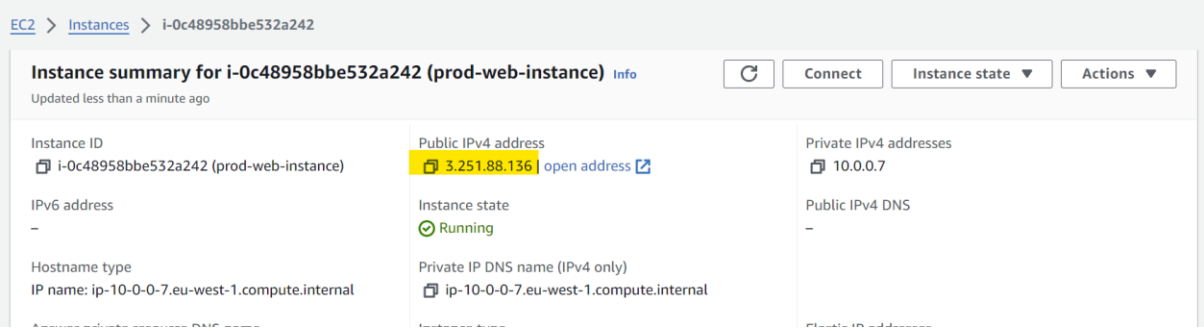
Search

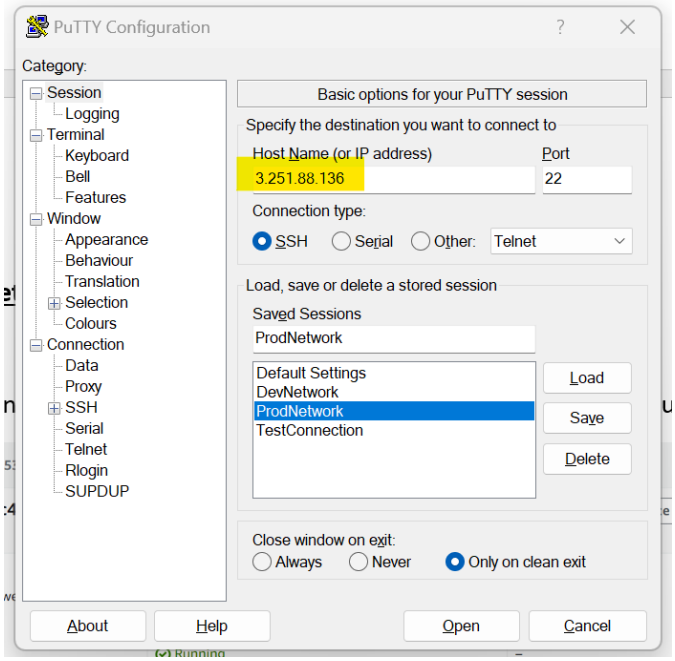
Name	Security group rule...	IP version	Type	Protocol	Port rang
–	sgr-0e370881e3dd6d...	IPv4	HTTP	TCP	80
–	sgr-0e7693d2cb818bf63	IPv4	HTTPS	TCP	443
–	sgr-0f7f4aed4a3fa8706	IPv4	SSH	TCP	22

Testing connection between private and public networks (with PuTTY)

Production:

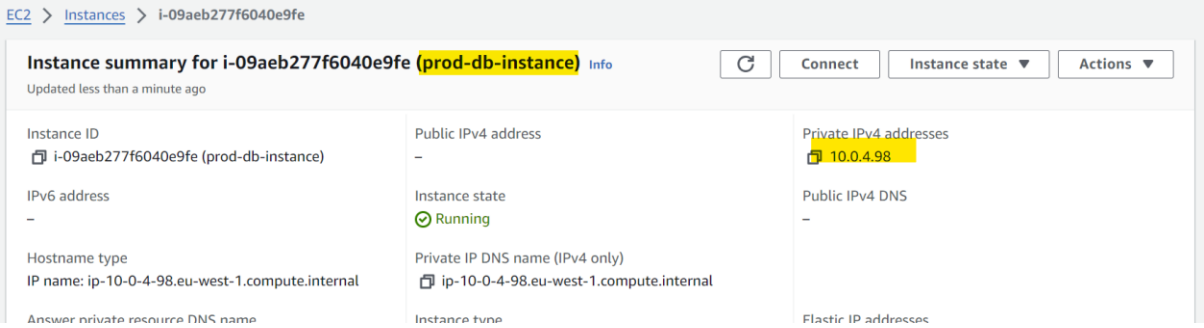
The public IP of web instance is **3.251.88.136**, and we will configure the sane in PuTTY.





Now, we will ping any of the private servers (app1, app2, db, dbcache) from the configured web instance.

Private IP of db instance is **10.0.4.98**, which we will ping to see if successful connection gets established



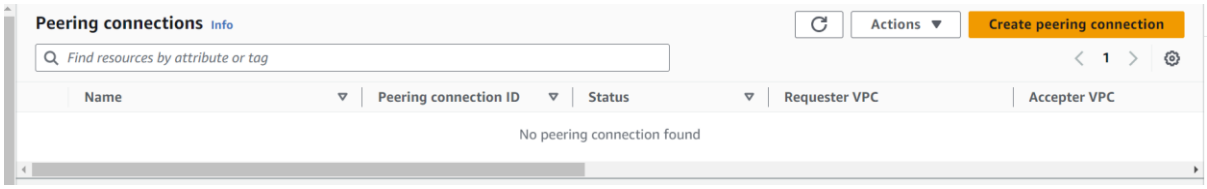
As observed in the below screenshot, the connection establishment is successful

```
ec2-user@ip-10-0-0-7:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Jul 22 08:30:24 2024 from 111.125.240.53
[ec2-user@ip-10-0-0-7 ~]$ ping 10.0.4.98
PING 10.0.4.98 (10.0.4.98) 56(84) bytes of data.
64 bytes from 10.0.4.98: icmp_seq=1 ttl=127 time=0.842 ms
64 bytes from 10.0.4.98: icmp_seq=2 ttl=127 time=0.672 ms
64 bytes from 10.0.4.98: icmp_seq=3 ttl=127 time=0.494 ms
64 bytes from 10.0.4.98: icmp_seq=4 ttl=127 time=0.462 ms
64 bytes from 10.0.4.98: icmp_seq=5 ttl=127 time=0.490 ms
64 bytes from 10.0.4.98: icmp_seq=6 ttl=127 time=0.536 ms
64 bytes from 10.0.4.98: icmp_seq=7 ttl=127 time=0.492 ms
64 bytes from 10.0.4.98: icmp_seq=8 ttl=127 time=0.538 ms
64 bytes from 10.0.4.98: icmp_seq=9 ttl=127 time=0.514 ms
^C
--- 10.0.4.98 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8357ms
rtt min/avg/max/mdev = 0.462/0.560/0.842/0.114 ms
[ec2-user@ip-10-0-0-7 ~]$
```

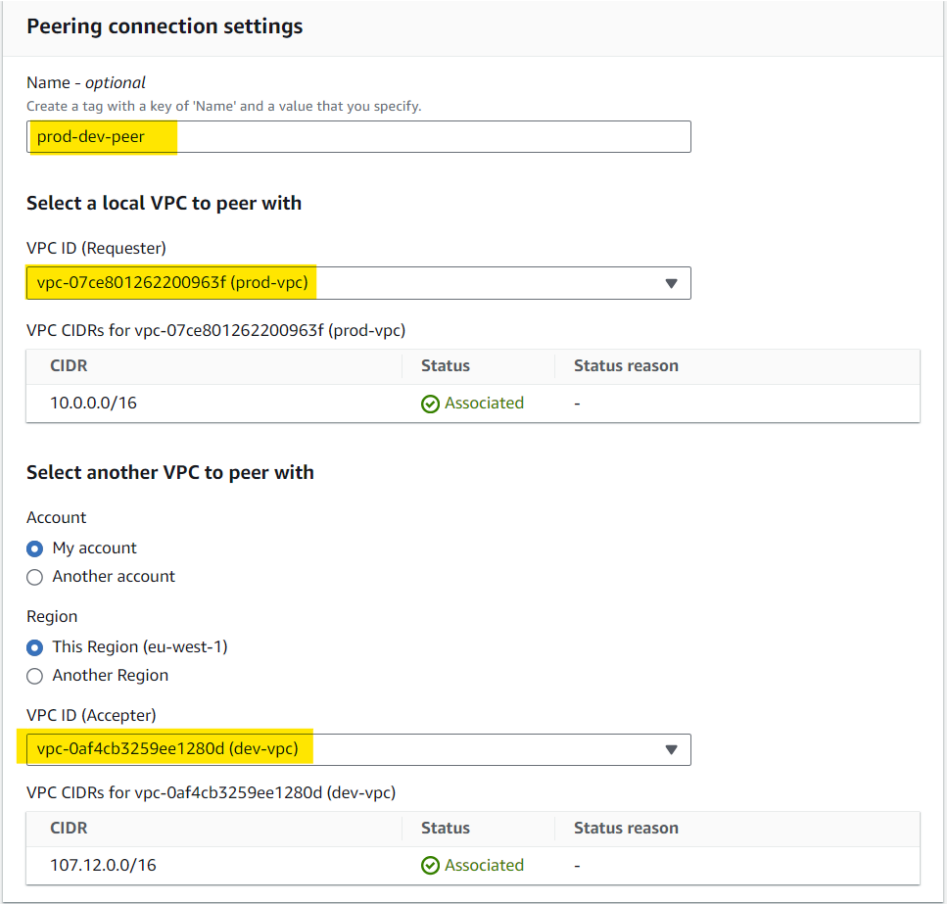
Same steps can be followed for testing connection of *development* network.

Peering production and development networks

To create a peering connection, in VPC dashboard, select 'Peering Connections' and click Create peering connection.

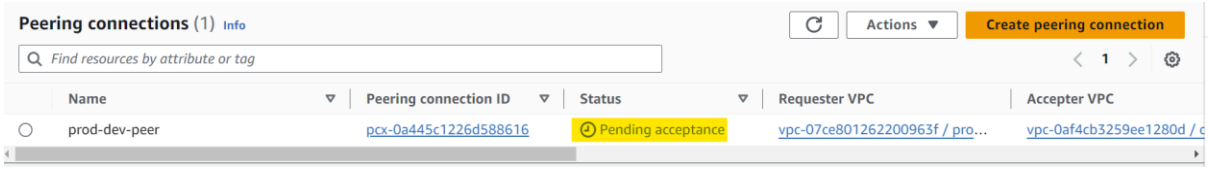


Enter peering name and select requester VPC and accepter VPC

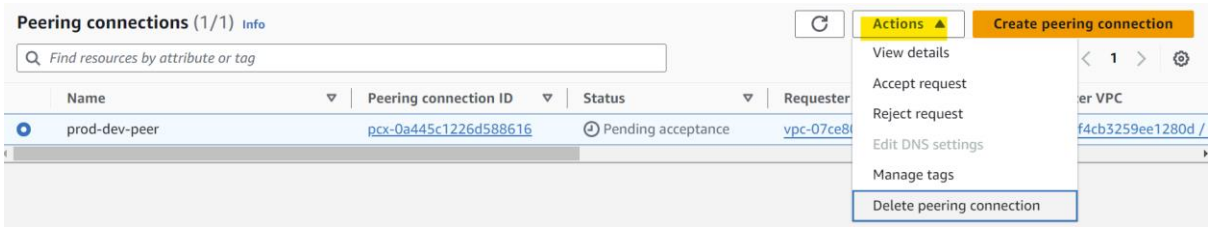


Once done, click 'Create peering connection' button at the bottom

Once peering created, the status is Pending Acceptance until user accepts the connection request

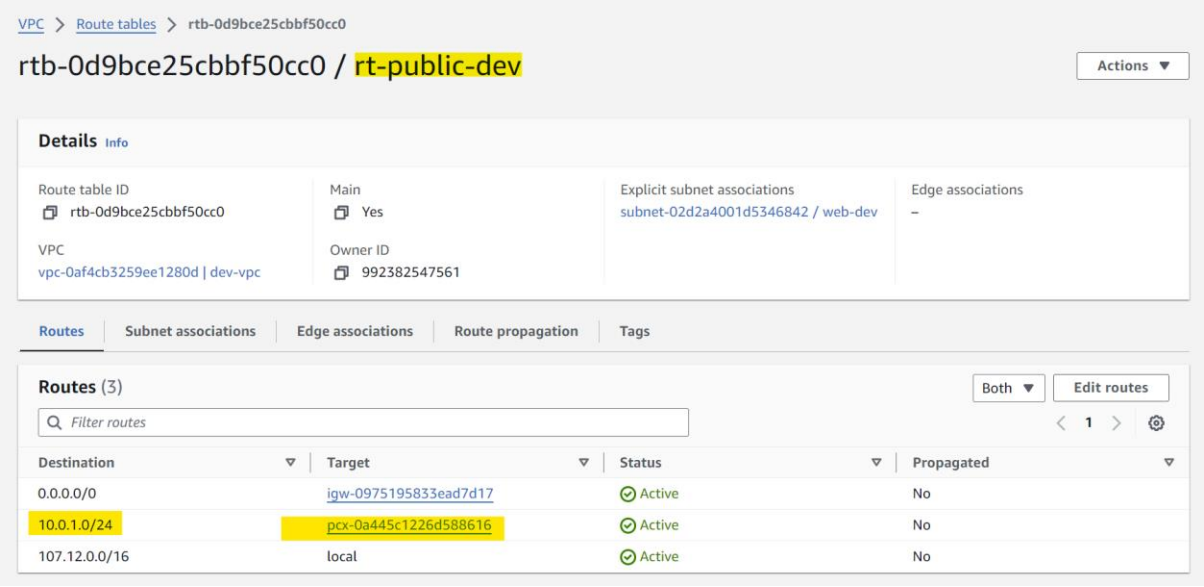


Choose the peering connection, click actions and select Accept Request. Follow the screenshot

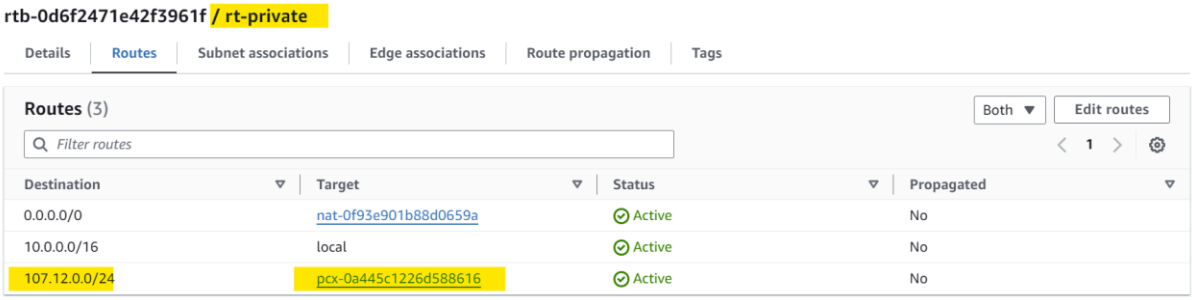


Test peering connections

To test the established peered connection we will configure the add the peering connection in public and private route tables in both production and development networks and destination CIDR of the required subnets.



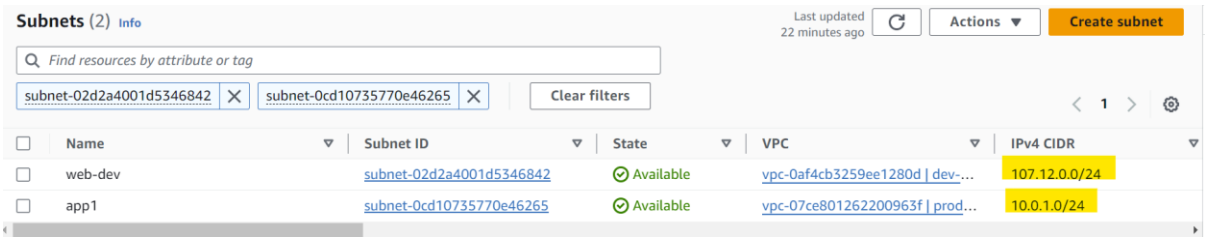
Add peering connection in development public route



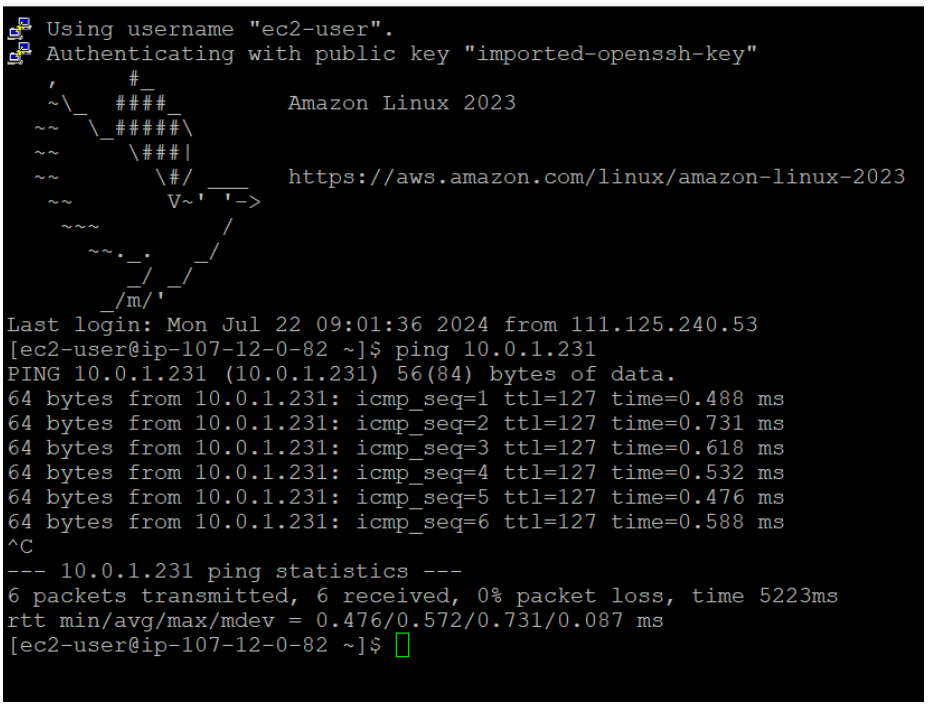
Add peering connection in production private route

Testing the connection in PuTTY

Launch PuTTY with web instance of development environment and ping app1 from production network to see if the peering was successful.



107.12.0.0/24 is CIDR of web subnet in development network and 10.0.1.0/24 is CIDR of app1 in production network. The same can be found in the screenshot below.



As the ping is receiving a response, we can conclude that peering between production and development network was established successfully.